



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

MASTER THESIS

Diellor Hoxhaj

**Techniques Applicable to the Analysis
of Educational Data**

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis: doc. RNDr. Iveta Mrázová, CSc.

Study programme: Computer Science

Study branch: Artificial Intelligence

Prague 2023

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date

Author's signature

I would like to thank my supervisor, doc. RNDr. Iveta Mrázová, for her guidance, advice, and detailed feedback throughout the preparation of my thesis. I would also like to thank my family and friends for their support.

Title: Techniques Applicable to the Analysis of Educational Data

Author: Diellor Hoxhaj

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: doc. RNDr. Iveta Mrázová, CSc., Department of Theoretical Computer Science and Mathematical Logic

Abstract:

Education plays a crucial role in shaping society. Graduates from higher education institutions have higher incomes. In 2023, U.S. employees aged 25 and over with high school diplomas made \$853 per week, while employees with master's degrees made \$1661 per week. Higher education is also connected with unemployment rates, which can affect economic growth, poverty, and societal well-being. Nonetheless, challenges such as racial disparities in graduation rates and low retention rates remain crucial in education. This thesis addresses these issues by focusing on techniques applicable to analyzing educational datasets. These techniques are also used to understand and analyze the importance of education in the political landscape.

Experimental results are divided into three parts. In the first part of the experiments, a dataset containing educational institutions in the U.S. Data was collected using Python. Using clustering techniques on this dataset, six distinct clusters were identified. The analysis of clusters highlights that universities with high tuition fees in the U.S., a small number of students, and diverse ethnic populations have higher graduation rates for all ethnic groups. The analysis further identified that universities in areas with high median household incomes and high tuition fees have better graduation rates.

In the second part of the experiments, social network techniques were used to analyze the influence of the educational background of members of parliament on the structure of the UK parliament. Key members were identified using centrality measures, with one of them, Keir Starmer, who became a prime minister in recent election. Universities like Oxford and Cambridge were the most frequently attended institutions among Members of Parliament.

In the third experiment, an early prediction model is used to assess student success. Key contributors identified are the interactions with the online learning environment and course domains (STEM or Social Sciences). The best-performing model, random forest, has an accuracy of 70%. This thesis demonstrates the application of data mining techniques to address some of the critical issues in the higher education domain.

Keywords: data mining classification vizualization social networks decision trees clustering centrality measures community detection

Contents

Introduction	5
1 Machine learning methods applicable to educational data	6
1.1 Overview of machine learning methods	6
1.1.1 Types of learning	6
1.1.2 Supervised learning	6
1.1.3 Unsupervised learning	7
1.1.4 Machine learning tasks	7
1.1.5 Classification vs regression tasks	8
1.1.6 Performance of the model	8
1.2 Cluster analysis	8
1.2.1 K - means	9
1.2.2 K-medoids	9
1.2.3 Self organizing maps	10
1.2.4 Silhouette score	13
1.3 Social network analysis	13
1.3.1 Centrality measures	14
1.3.2 Community detection	15
1.3.3 The Kernighan-Lin algorithm	16
1.3.4 Louvain algorithm	16
1.4 Decision trees	17
1.4.1 ID3 algorithm	19
1.4.2 CART algorithm	20
1.4.3 C4.5 algorithm	22
1.4.4 Decision tree algorithm – ID3 solved example	24
1.4.5 Other filter models	29
1.5 Ensemble methods	30
1.5.1 Bagging	31
1.5.2 Random forests	32
1.5.3 Boosting	32
1.6 Educational system	34
1.6.1 Educational system in UK	34
1.6.2 Educational system in the US	35
2 Experimental results	38
2.1 Cluster analysis on graduation Rates in U.S. colleges	38
2.1.1 Introduction	38
2.1.2 Data scraping	38
2.1.3 Scraping graduation rates	40
2.1.4 Scraping tuition fees	40
2.1.5 Scraping latitude and longitude	41
2.1.6 Scraping county	42
2.1.7 Scraping median household income	43
2.1.8 Merging and preprocessing	43
2.1.9 Data analysis	44

2.1.10	Cluster analysis - experiment setup	50
2.1.11	Clustering with K-means	50
2.1.12	Cluster visualization and interpretation	52
2.1.13	Changes in universities across the 2020, 2021 and 2022	65
2.1.14	Conclusions on the analysis of graduation rates in U.S. colleges	66
2.2	Social network analysis	67
2.2.1	Background on united kingdom parliament	67
2.2.2	UK parliament analysis	68
2.2.3	Graph structure	69
2.2.4	Community detection in UK Parliament	72
2.2.5	Kosovo parliament analysis	72
2.2.6	Graph structure	73
2.2.7	Background on Kosovo parliament structure	73
2.2.8	Community detection in Kosovo Parliament	75
2.3	Experimental results: OULAD dataset	76
2.3.1	Understanding the data	76
2.3.2	Data analysis	83
2.3.3	Introduction to experiments	91
2.3.4	Experiment 1 - Random forest	94
2.3.5	Experiment 1 - AdaBoost	94
2.3.6	Experiment 1 - CART	95
2.3.7	Experiment 1 - C4.5	98
2.3.8	Experiment 2 - Experimental results	101
2.3.9	Experiment 2 - Random forests	101
2.3.10	Experiment 2 - AdaBoost	101
2.3.11	Experiment 2 - CART	102
2.3.12	Experiment 2 - C4.5	102
2.3.13	Experiment 3 - Experimental results	104
	Conclusion	108
2.4	US graduation rates analysis	108
2.4.1	Relationships and trends within the clusters	108
2.5	Impact of education on the parliament analysis	109
2.5.1	Parliament of Kosovo Analysis	111
2.6	Prediction of student success/failure in a virtual learning environment	112
2.7	Future work	113
	Bibliography	115
	List of Figures	118
	List of Tables	120
	List of Abbreviations	121

A Attachments	122
A.1 User Documentation	122
A.1.1 Graduation Rates in U.S. Experiments	122
A.1.2 Predicting Student Performance Experiments	122
A.1.3 Social Networks Analysis	123
A.1.4 Additional Modifications	123

Introduction

With the advancements in information technologies seen in the past decade, a massive amount of data is generated daily. Data mining is a process of finding useful information by exploring large datasets [1]. Data mining methods can be useful in various domains, including education. This thesis uses data mining techniques to address the current issues in the education realm, such as student retention and racial disparities in graduation rates. Additionally, it investigates the influence of educational background on members of parliament and the role of education in the political landscape.

A significant challenge in education is the differences in graduation rates among different ethnic groups. Black and Hispanic students are less likely to graduate than Asian and White students in the U.S. [2]. Using data mining techniques, this research analyzes universities with different graduation rates across different ethnic groups by clustering these institutions to determine trends and relationships. Universities will be clustered based on their graduation rates, tuition fees, and median household income of the county (region) where the university is located.

Clustering techniques such as SOM, K-Means, and K-medoids will be used for the analysis. This analysis will help understand the characteristics of universities that perform well in all ethnic groups and also investigate how features like tuition fees and the median household income of the county (region) where the university is located can impact graduation rates.

This research focuses on finding which universities have the highest graduation rates among all ethnic groups, analyzing their characteristics, and determining universities where specific ethnic groups perform their best. This helps in better understanding the problem of disparities among graduation rates of different ethnic groups.

The role of government is crucial in societies. The government regulates a country or community by providing rules, making political choices, and delivering public services [3]. In the second part of the experiments, social network analysis methods are used to find the most influential members in the parliament and determine the role of education in the parliament's structure.

With the recent pandemic and the current improvements in information technology, virtual learning environments (VLEs) and online university degrees have increased rapidly. This increase in the online learning environment generates large amounts of data. This data can be leveraged to develop models predicting students' performance. These models can help identify at-risk students as early as possible, helping professors make a timely intervention. Additionally, this data can be used to understand the factors that influence students' success. While the current work in this research area focuses on developing at-risk models using virtual learning environments, the models are not interpretable. Interpretable models such as decision trees are going to be used to address this issue.

This thesis can help improve university student retention rates by developing these early prediction models and better understanding the factors contributing to students' success.

The first chapter of the thesis describes the machine learning techniques appli-

cable to educational datasets. This chapter includes techniques such as decision trees, ensemble methods, and clustering techniques. The second part of the thesis provides experimental results, which are further divided into three subsections, each addressing different issues in the educational domain. In the first subsection, Cluster analysis on graduation rates in U.S. colleges results are explained. In the second subsection of the experiments, a social network analysis on the parliaments of the U.K. and Kosovo is conducted. In the third part, experimental results of predictive algorithms on open-source educational datasets are shown. The final chapter of this thesis covers the research findings. Additionally, the attachments section has detailed instructions on how to run the Python programs used in the experiments.

1. Machine learning methods applicable to educational data

1.1 Overview of machine learning methods

There is a massive amount of data generated daily. There are approximately one trillion web pages on the internet, and each second, there is an hour-long video uploaded on streaming platforms like Youtube [4]. In order to get valuable insights from this data, data mining and machine learning algorithms can be applied. Data mining is a process of finding useful information by exploring large datasets [1]. Data mining can also be viewed as a process of discovering "gems" or valuable information in some large dataset. Various domains, such as e-commerce, social media, medicine, transportation services, streaming platforms and research use data mining techniques to get new insights and make predictions [5].

For example, different search engine corporations can analyze and store the user's interactions in their system, to improve their search engine results and streaming platforms can analyze watching history, including the types of videos being watched, the time that is spend on each video with a specific content, and various interaction data to predict and recommend other videos that users might be interested in.

Machine learning is an area of research that intersects with other areas like statistics, artificial intelligence, and computer science. Machine learning is also known as statistical learning. This field uses algorithmic approaches to build complex statistical models from the data. These models are essentially functions that can be used to predict future data [6, 7]. For example, consider the problem of predicting the current housing prices from features like the area of the house, the number of bathrooms, and the number of rooms. In this case, machine learning algorithms such as linear regression is applied to build this model. This model learns (is trained) from an existing real estate sales dataset where the prices of houses are known, after training, it can be used to predict the prices of other houses where the price is unknown.

1.1.1 Types of learning

Based on the presence or absence of the information about the real outcomes of the data, learning can be supervised, unsupervised [8].

1.1.2 Supervised learning

In supervised learning, the dataset is a set of examples represented by $S = \{(x_i, y_i)\}_{i=1}^N$, where N denotes the total number of examples. Each pair (x_i, y_i) contains a feature vector of i -th example in the set S , denoted as \mathbf{x}_i and its corresponding label y_i . The feature vectors can be grouped and represented as a matrix $X \in \mathbb{R}^{N \times D}$, where each row \mathbf{x}_i of X corresponds to an example in the dataset S and it consists of D features, where each feature $x_i^{(j)}$ for $j = 1, 2, \dots, D$

represents a specific characteristic about the example. Therefore, the matrix X consists of rows (feature vectors) $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(D)})$ [8].

For example, in a dataset representing houses, each row in the matrix X corresponds to a specific house, where $x_i^{(1)}$ could represent the number of rooms in the house, $x_i^{(2)}$ might be the area of the house, and $x_i^{(3)}$ whether it has a balcony or not. Each column in X must represent the same type of information for all houses in the dataset.

The label y_i can take different values depending on the machine learning problem being addressed. In classification tasks, y_i can belong to a finite set of classes, represented as $S = \{1, 2, \dots, C\}$, where C is the number of classes. For example, the possible classes in a spam filter application would be $S = \{0, 1\}$, where 0 might be assigned for spam emails and 1 for non-spam emails. In multiclass classification problems, depending on the specific task, more than two possible labels can be assigned to each example in the dataset.

A supervised learning algorithm aims to use an algorithmic approach to develop a model, which essentially is a function f , that given a vector or an example \mathbf{x}_i it generates an output that helps determine the corresponding label of the input vector. The model aims to learn the mapping as shown in Equation 1.1:

$$y_i = f(\mathbf{x}_i) \tag{1.1}$$

Examples of such algorithms include linear regression, support vector machines, CART algorithm, ID3 algorithm, naïve Bayes, etc.

1.1.3 Unsupervised learning

In unsupervised learning, the dataset, represented by $S = \{x_i\}_{i=1}^N$, contains of examples without their corresponding labels. Each example in set S consists of only the input feature vector \mathbf{x} . The dataset provides no information on the real outcomes for each vector \mathbf{x} . These algorithms allow us to extract useful patterns from input data [6].

For example, these algorithms can be used to find groups of similar customers based on their purchase history or other specific characteristics that they might have in common. Examples of such algorithms include k-means clustering, k-nearest neighbors, hierarchical clustering, etc.

1.1.4 Machine learning tasks

In machine learning, models are trained using data instead of relying on predefined if-else rules typical of traditional programming. This method is effective for complicated tasks such as face detection or spam filtering. For example, in face detection, a machine learning model can learn to identify faces by analyzing a set of example images and figuring out the characteristics of a face by itself. This approach simplifies solving complex problems, making machine learning methods valuable in various applications [6].

Other machine learning applications include web page ranking, recognizing characters, computer vision-related tasks, speech recognition, etc. The set of tasks that machine learning algorithms can solve is large, as seen in the applications.

These problems usually belong to either classification tasks or regression tasks [7]. Other tasks include generative models, reinforcement learning tasks, etc.

1.1.5 Classification vs regression tasks

The classification model aims to categorize data into distinct classes. In machine learning, a model learns from labeled examples and then can be used to predict labels for new examples whose labels are unknown. The model either directly predicts a label for each new example or provides a probability from which a label can be inferred. If there are two options for classifying the data (like "spam" or "not spam"), it is a binary classification task. If there are three or more options, it is called multiclass classification [8].

Contrary to classification tasks that predict a class for a given input vector, regression predicts a real number. Examples of regression tasks include estimating a person's height from their T-shirt size, annual income from their education level or a person's age based on different attributes.

1.1.6 Performance of the model

To evaluate the performance and the effectiveness of a machine learning algorithm, we use quantitative measures denoted by p where $p \in \mathbb{R}$. In classification tasks, the measure p defines the accuracy of the algorithm, which means the ratio of input data points that the model classified correctly. In regression tasks, other metrics such as means squared error, mean absolute value, or root mean squared error, is used to estimate the performance of the model [7].

1.2 Cluster analysis

Various tasks demand the classification of data points into intuitively comparable categories. The splitting of a large number of patterns into a smaller number of categories helps in better understanding the data for a wide range of data mining applications. A simple, non-formal definition of clustering involves dividing or categorizing a set of patterns so that they are comparable and similar within each group [9].

Let m be a smaller integer than the total number of data points n . The cluster problem involves identifying m clusters (subsets), denoted as subsets $I_1, I_2, I_3, \dots, I_m$, from datapoints of a set X . Datapoints assigned to the same cluster are similar, whereas those in different clusters are considered dissimilar [10]. The goal of the clustering problem is to find the best way to divide the set X into subsets $I_1, I_2, I_3, \dots, I_m$ such that certain criteria are met. A function determines the best way to divide the set, often referred to as the objective function, which computes how desirable different groupings are.

Objective functions and two popular Clustering Algorithms, such as K-means and K-medoids, are described in the following section.

1.2.1 K - means

K-means clustering, developed by MacQueen in 1967 [11], is a popular method for grouping data into clusters. This method runs iteratively until the best arrangement of data points is found in clusters. Initially, K has to be set. K decides the number of clusters in which the data points will be grouped.

At the start, the algorithm picks K random points from the data and assigns them as the initial centers of these clusters. Next, each data point in the set is assigned to the closest cluster center using Euclidean distance, which is a method that measures the distance between two data points. After all points are assigned to their respective cluster, the algorithm recalculates the data points in the center of K clusters by calculating the average (or mean) of all data points within each cluster. These cluster centers are mean vectors created by averaging the values of the original data points in each cluster. After the centers have been updated, the algorithm again reassigns each data point to the closest cluster center, and the centers are updated again. This process of reassignment and center updating continues until the clusters do not change between iterations or until the maximum number of iterations is reached.

Given a dataset $X = \{x_i\}$ where $i = 1, 2, \dots, n$ and each x_i is a d-dimensional data point of size n, the goal is to divide X into k clusters $C = \{C_j\}$ where $j = 1, 2, \dots, k$. Each cluster C_k is formed to minimize the sum of the squared distances between the data points and the cluster centroid μ_k . The Equation 1.2 shows the cost function for a single cluster C_k :

$$J(C_k) = \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \quad (1.2)$$

The goal of the K-means algorithm is to minimize this total sum of squared errors for all clusters as shown in Equation 1.3:

$$J(C) = \sum_{k=1}^K \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \quad (1.3)$$

The Algorithm 1 shows the pseudocode for the K-means algorithm.

Algorithm 1 K-means Algorithm

- 1: Initialize K centroids randomly
 - 2: **repeat**
 - 3: **for** each data point **do**
 - 4: Assign the point to the nearest centroid
 - 5: **end for**
 - 6: **for** each centroid **do**
 - 7: Update centroid to the mean of assigned data points
 - 8: **end for**
 - 9: **until** Convergence
-

1.2.2 K-medoids

K-medoids clustering, proposed by Kaufman and Rousseev in 1987 [12], is a variant of K-means in which real data points, known as medoids, become cluster

representatives. A medoid is a data point in a cluster with the lowest total dissimilarity to all other data points. This means that the medoid is the data point with the lowest sum of dissimilarities (such as distances) to all other points in the cluster. Therefore, it is the most centrally situated data point in the cluster. Unlike cluster centers in K-means, which may not correlate to actual data points, medoids are always actual data points from the dataset.

The main advantage of K-medoids over K-means is that using K-means, cluster centers can be distorted by outliers (because it uses mean to compute cluster centers and mean is sensitive to outliers). In these cases, the cluster centers may be positioned in an empty zone that does not represent the majority of the data points in that cluster. This distortion may cause partial merges between different clusters, which is not desired. Handling outliers beforehand or using other techniques like K-medians can help resolve this issue.

Initially, K medoids are initialized randomly or using a predefined strategy. Next, using a selected dissimilarity measure, data points are assigned to the cluster whose medoid is closest to them. This process is repeated by assigning data points to clusters and updating medoids until there are no changes in medoids. The objective is to maximize the distance between clusters while minimizing the overall dissimilarity within clusters. The Euclidean distance metric can calculate the distance between data points and medoids.

The Euclidean distance between two points $A = (x_1, x_2, \dots, x_n)$ and $B = (y_1, y_2, \dots, y_n)$ in \mathbb{R}^n is given in Equation 1.4:

$$\left(\sum_{i=1}^n (x_i - y_i)^2 \right)^{1/2} \quad (1.4)$$

The cost function which is minimized is given in Equation 1.5:

$$\text{Cost}(C^1, \dots, C^k, \mathbf{z}_1, \dots, \mathbf{z}_k) = \sum_{j=1}^k \sum_{i \in C^j} d(\mathbf{x}_i, \mathbf{z}_j) \quad (1.5)$$

Where:

- k : The number of clusters (medoids).
- C^j : The set of data points in the j -th cluster.
- \mathbf{z}_j : The medoid of the j -th cluster.
- $d(\mathbf{x}_i, \mathbf{z}_j)$: The dissimilarity (distance) between the data point \mathbf{x}_i and the medoid \mathbf{z}_j . This dissimilarity measure d can be an Euclidean distance measure, but other measures can be used as well.

The Algorithm 2 shows the pseudocode for k-medoids algorithm.

1.2.3 Self organizing maps

The Self-Organizing Map (SOM) model is an innovative model inspired by biological processes. SOM is typically used for the visualization of high-dimensional data. SOM works by transforming nonlinear relationships between high-dimensional data onto a lower-dimensional space, typically a 2D grid.

Algorithm 2 K-Medoids Clustering Algorithm

```
1: function KMEDOIDS(Dataset  $X$ , number of clusters  $k$ )
2:   Randomly select  $k$  data points from  $X$  to serve as initial medoids  $m_k$ 
3:   repeat
4:     Assign each non-medoid data point to the nearest medoid
5:     Calculate the total distance  $TD_i$  between medoids  $m_i$  and non-medoids
         $e_j$ 
6:     for each medoid  $m_i$  do
7:       Identify the non-medoid  $e_j$  that minimizes the total distance  $TD$ 
8:       Calculate  $TD(e_j \rightarrow m_i)$ 
9:       if  $TD(e_j \rightarrow m_i)$  is less than the current  $TD_i$  then
10:        Swap  $m_i$  with  $e_j$ 
11:      end if
12:    end for
13:  until no  $m_i$  changes
14:  return  $K$ , a set of  $k$  clusters
15: end function
```

The 2D grid maintains the topographical order of the data in the original space, making this model helpful in analyzing and better understanding the patterns, clusters, and relationships within the data.

The SOM model has a fixed number of neurons or processing units, and they are connected in a grid of neurons in hexagonal or rectangular shapes. Given an input vector \mathbf{x} , which represents a single high-dimensional data point, SOM finds the i -th unit with the closest weight vector using a distance measure such as Euclidean distance. To find the i -th neuron with the closest weight vector, the following formula is shown in Equation 1.6:

$$i(\mathbf{x}) = \arg \min_j \|\mathbf{x} - \mathbf{w}_j\| \quad (1.6)$$

In this equation:

- \mathbf{x} is the input vector representing a single high-dimensional data point.
- \mathbf{w}_j is the weight vector associated with the j -th neuron, which has the same dimensionality as \mathbf{x}

After finding the winning neuron using this formula, the weights of the neighbors of the winning neuron i are also updated. For each unit j in the neighborhood $N(i)$, the weights \mathbf{w}_j are updated. To update the weights of the neighbors of the winning neuron i , the lateral distance d_{ij} between the winner unit i and unit j is used and defined by the Equation 1.7:

$$h_{ij}(d_{ij}) = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right) \quad (1.7)$$

This is the Gaussian neighborhood function, where the values of σ can be modified. The weight update is performed using the Equation in 1.8:

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \Delta\mathbf{w}_j \quad (1.8)$$

Algorithm 3 Self-Organizing Map Training

Data: training data \mathbf{X} ; Grid MAP \mathbf{W} , with dimensions k and l ; the maximal number of iterations m ; the neighborhood function h

Result: SOM vectors \mathbf{W}

function SOM(\mathbf{X}, k, l, h, m)

 // Set the initial value of the learning rate η

$\eta \leftarrow \text{someInitialValue}$

 // Initialize all $\mathbf{w} \in \mathbf{W}$ to random values

for each $\mathbf{w}_i \in \mathbf{W}$ **do**

$\mathbf{w}_i \leftarrow \text{randomVec}(N)$

end for

for $n \leftarrow 1$ to m **do**

for each $\mathbf{x} \in \mathbf{X}$ **do**

 // Find the neuron with the closest weight vector to \mathbf{x}

$c \leftarrow \arg \min_j \|\mathbf{x} - \mathbf{w}_j\|$

 // Update the weights of every neuron in \mathbf{W}

for each $\mathbf{w}_j \in \mathbf{W}$ **do**

$\mathbf{w}_j \leftarrow \mathbf{w}_j + \eta \cdot h_{cj} \cdot (\mathbf{x} - \mathbf{w}_j)$

end for

end for

 // Optionally reduce the learning rate η

$\eta \leftarrow \text{updateEta}(\eta)$

end for

return \mathbf{W}

end function

Where:

$$\Delta \mathbf{w}_j = \eta \cdot h_{ij} \cdot (\mathbf{x} - \mathbf{w}_j) \quad (1.9)$$

In Equation 1.9 η is the learning rate, h_{ij} is the neighborhood function, \mathbf{x} is the input vector, and \mathbf{w}_j is the weight vector of the j -th unit. The training algorithm of SOM is shown in the pseudocode in Algorithm 3.

1.2.4 Silhouette score

Silhouette score is a method of evaluating the cluster's quality by measuring how similar data points (objects) are in their cluster compared to others. This evaluation metric can be used to choose the 'right' number of clusters [13].

Given three clusters, A, B, and C. To calculate the silhouette score for a cluster, first, the average dissimilarity of a data point i to all the other data points within the same cluster is computed using the Equation 1.10:

$$a(i) = \frac{1}{|A| - 1} \sum_{j \in A, j \neq i} d(i, j) \quad (1.10)$$

Then the average similarity of data point i to all the data points in cluster B and C is calculated using the formulas in Equation 1.12 and Equation 1.11:

$$d(i, B) = \frac{1}{|B|} \sum_{j \in B} d(i, j) \quad (1.11)$$

$$d(i, C) = \frac{1}{|C|} \sum_{j \in C} d(i, j) \quad (1.12)$$

Then, the $b(i)$ is determined such that the minimum average dissimilarity of i is selected (i.e., the nearest different clusters from i) using the Equation 1.12:

$$b(i) = \min(d(i, B), d(i, C)) \quad (1.13)$$

The formula for computing the silhouette score of $s(i)$ is:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (1.14)$$

This score in Equation 1.14 is the silhouette score. To get the overall quality of the created clusters, the average of all the silhouette scores $s(i)$ is computed. The resulting value will range from -1 to 1, where higher values indicate better clustering.

1.3 Social network analysis

Social networks provide a framework for understanding social systems by focusing on the relationships between the entities that form the system. These entities within the system are also known as nodes [14]. Nodes have properties known as 'attributes' that distinguish them. These characteristics can be descriptive, such

as expertise or skills, or numerical, such as income level. In network analysis, relationships between nodes have unique features known as links or ties, describing how the entities represented by nodes are connected.

In social networks, the nodes research usually represent individuals, such as humans or other social species, or organizations, such as schools, governments, private companies, etc. In network analysis, one important concept is that ties (or links) are connected through common nodes. For instance, the $X \rightarrow Y$ connection shares a node with the $Y \rightarrow Z$ link, resulting in chains or pathways. These interconnected links create a graph, which is also called a network. Graphs are usually used to represent networks in social network analysis. A graph G is a pair (V, E) where V is a set of vertices, also known as nodes, and E is the set of edges, sometimes referred to as links. Two vertices are considered adjacent when an edge connects them, indicating a connection or a relationship.

In undirected graphs, the degree of a node shows the total number of edges attached to the node. For example, if there is an edge between A and B and another edge links A to C , these edges are both connected to A , giving A a degree of two. Graphs can be directed or undirected. Relationships with a clear direction, like "reports to" or "sends a message to" is represented by directed graphs.

Undirected graphs are used in social network analysis to represent mutual or bidirectional relationships between individuals. For example, consider the relationship of individuals "attending the same university." In this case, the connection between two individuals lacks direction. It simply shows that both individuals share the exact attributes. The direction of the relationship is not important.

Social network analysis also includes centrality measures, which measure node importance, and community detection algorithms identify clusters within the network. These topics are further discussed in the following chapters.

1.3.1 Centrality measures

In undirected graphs, degree centrality measures a node's connections in a network [15]. For example, for a given node i , the degree centrality represents the total count of its edges. To normalize the degree of a node such that it falls between 0 and 1, the degree of a node i in a graph g is divided by $n - 1$, which is the maximum possible number of connections a node can have in this network. The Equation 1.15 is used to compute the normalized degree centrality, where $d_i(g)$ is the degree of node i and g is the graph, n is the total number of nodes in graph g .

$$c_i^{\text{deg}}(g) = \frac{d_i(g)}{n - 1} \quad (1.15)$$

The degree centrality shows how well-connected the individual is or how 'popular' is the node within the network. However, it does not provide any insight into the node's position in the network.

Closeness centrality focuses on the sum of distances between one node and all the other nodes in the network. And it is defined by the equation in 1.16:

$$\sum_j \rho_g(i, j) \quad (1.16)$$

The higher the distance, the lower the centrality, and the lower the distance, the closer the node is to other nodes in the network (the higher the centrality). To normalize this sum of distances such that the measurement ranges from 0 to 1, with 0 indicating a low centrality and one a high centrality, the Equation in 1.17 is used:

$$c_{\text{cls},i}(g) = \frac{n - 1}{\sum_{j \neq i} \rho_g(i, j)} \quad (1.17)$$

Where:

- $c_{\text{cls},i}(g)$ is the normalized closeness centrality of node i of graph g .
- n is the total number of nodes in the network.
- $\sum_{j \neq i} \rho_g(i, j)$ is the sum of distances from node i to all other nodes j .

Betweenness centrality measures the importance of a node by computing how well it connects other nodes in the network. For a node i , this measure considers all shortest paths between pairs of nodes in the graph, which include node i (i.e., paths that pass through node i). So, the central or most important nodes serve as bridges in the network and help transmit information. The Equation in 1.18 for a node i computes the ratio between the number of shortest paths between j and k which include node i ($p_{jk}(i)$) and the total number of shortest paths between j and k (p_{jk}):

$$c_i^{\text{bet}} = \sum_{(j,k), j \neq i, k \neq i} \frac{p_{jk}(i)}{p_{jk}} \quad (1.18)$$

To make sure that the final measure is normalized, the Equation in 1.19 is used:

$$C'_B(i) = \frac{2}{(n - 1)(n - 2)} \sum_{j < k} \frac{p_{jk}(i)}{p_{jk}} \quad (1.19)$$

Where n is the total number of nodes in the graph.

1.3.2 Community detection

Finding communities is one of the main tasks in social network research [16]. Depending on the context, social network communities are sometimes called clusters or groups. The clusters in the network are identified by focusing on the nodes that interact with each other more often than with other nodes in other clusters. This technique makes analyzing and solving problems easier from a group perspective (i.e., the focus is on the clusters, not individual nodes). One of the oldest community detection algorithms, The Kernighan Lin [17], is described in the following section to help understand how community detection algorithms work.

1.3.3 The Kernighan-Lin algorithm

This algorithm is a partitioning algorithm that focuses on minimizing the sum of edge weights between different partitions of the graph, Furthermore, it has a constraint where the sizes of the partitions are equal [17, 18].

The objective function is shown in Equation 1.20. The $A(V_i, V_j)$ is the sum of all edge weights between vertices of set V_i and V_j . Moreover, the function is minimized by considering the constraint that $|V_1| = |V_2| \dots |V_k|$.

$$KLObj(V_1, \dots, V_k) = \sum_{i \neq j} A(V_i, V_j) \quad \text{subject to} \quad |V_1| = |V_2| = \dots = |V_k| \quad (1.20)$$

Initially, the graph is split into two sets, A and B , so their size is equal. In each iteration, the algorithm computes the gain value, which measures the total reduction of the sum of edge weights if vertex v would have been swapped with another vertex from the other set. Edge weights represent a graph's cost or importance of two connected vertices. It does this greedily for each possible swap between two sets until it identifies the swap with the highest reduction in edge cut (i.e., minimizing the objective function as shown in Equation 1.20). After the swap, the vertex being considered will not be rechecked in the next iteration. After the swap, the gains are calculated again to reflect the swap, and the procedure continues until no further improvements are made in swapping the nodes. The algorithm has a complexity of $O(|E| \log |E|)$ per iteration. Where $|E|$ is the number of edges.

1.3.4 Louvain algorithm

This algorithm has two phases [19]. In the first phase, given a weighted network with N nodes, the algorithm assigns a community for each node. For each node i in the network, the gain of modularity is computed by moving the i -th node into the community of its neighbor j .

The change in modularity ΔQ when node i is moved into community C is defined as:

$$\Delta Q = \left[\frac{\Sigma_{\text{in}} + 2k_{i,\text{in}}}{2m} - \left(\frac{\Sigma_{\text{tot}} + k_i}{2m} \right)^2 \right] - \left[\frac{\Sigma_{\text{in}}}{2m} - \left(\frac{\Sigma_{\text{tot}}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (1.21)$$

Where Σ_{in} is the sum of the weights of each edge inside a community C , Σ_{tot} is the sum of the weights of all edges within the community and towards nodes in C , k_i is the sum of the weights of all edges connected to node i , $k_{i,\text{in}}$ is the sum of the weights of the edges from i to nodes in C , m is the sum of the weights of all edges in the network.

For each node i in the network, the change in modularity is computed by moving i to its neighbor's j community. If the gain (changes in Q) is positive as computed using the Equation 1.21 node i moves to the community of j , which has the highest gain. If there is no positive gain, it stays in its original community. This process loops sequentially for all nodes in the network until no more gain in modularity can be achieved.

After completing the first phase, each detected community becomes a node in a new network. The weight of an edge between two new nodes (communities)

is the sum of the weights of nodes in the original community. Edges within the same community in the original network create self-loops; the weight is again the sum of all the edge weights within the same community. These two phases are repeated on each newly created network until no improvement in modularity is possible. The algorithm, in practice, usually runs in $O(n \log n)$ time, where most of the computation is done in the first phase. In worst cases, the algorithm runs in $O(n^2)$ time.

1.4 Decision trees

A decision tree is a type of machine learning model that is capable of solving both classification and regression tasks [5]. When a decision tree is used to solve regression tasks, it is referred to as a regression tree, and when it is constructed for classification tasks, it is called a classification tree. An example of its application is classifying emails as either spam or not.

To train such decision tree for this example, a dataset containing various examples of emails that have already been classified as spam or not spam is necessary. This training dataset is used to construct the decision tree, which can then be used to determine if an email is spam or not based on the learned rules and paths developed during the training process.

Classification trees find applications in multiple practical domains such as marketing, finance, engineering, and medicine. An example of a classification tree is illustrated in Figure 1.1

The tree in Figure 1.1 provides a structured framework that banks can use to decide whether to approve a loan application. When clients apply for a loan, they fill in an application form with their personal information, and the bank uses this information to decide whether or not to give them the loan. The application form includes data like credit history, loan amount, applicant's and co-applicants income, etc. Credit history is usually a value between 0 and 1, where 0 means that the applicant has a poor credit history and 1 means that the applicant has a good credit history.

Based on this information, the decision tree classifies the applicants into one of the following classes: Accepted, which means that the loan should be accepted. Rejected, which means that the loan should be rejected.

Banks can make informed decisions about who is more likely to repay the loans using this decision tree and minimize their financial losses. For example, when a new applicant applies for a loan, the bank will ask questions using this decision tree to evaluate the client's application. Starting from the root node of the tree, the questions will be: What is the applicant's credit history score? If it is less than 0.5, the process moves to the left child node; otherwise, it moves to the right child node. The next node checks if the loan amount is less than or equal to 547 (which is the loan amount in thousands) If true, it checks if CoapplicantIncome is less than or equal to 662. If this is true, the tree predicts a certain class (i.e., a decision on loan approval). Similarly, by following the paths in the tree, the bank can decide on the loan application.

Many researchers believe that decision trees are so popular and used because they are self-explanatory. You can follow a certain decision in the tree and understand why such decisions are made. The training phase of the decision tree

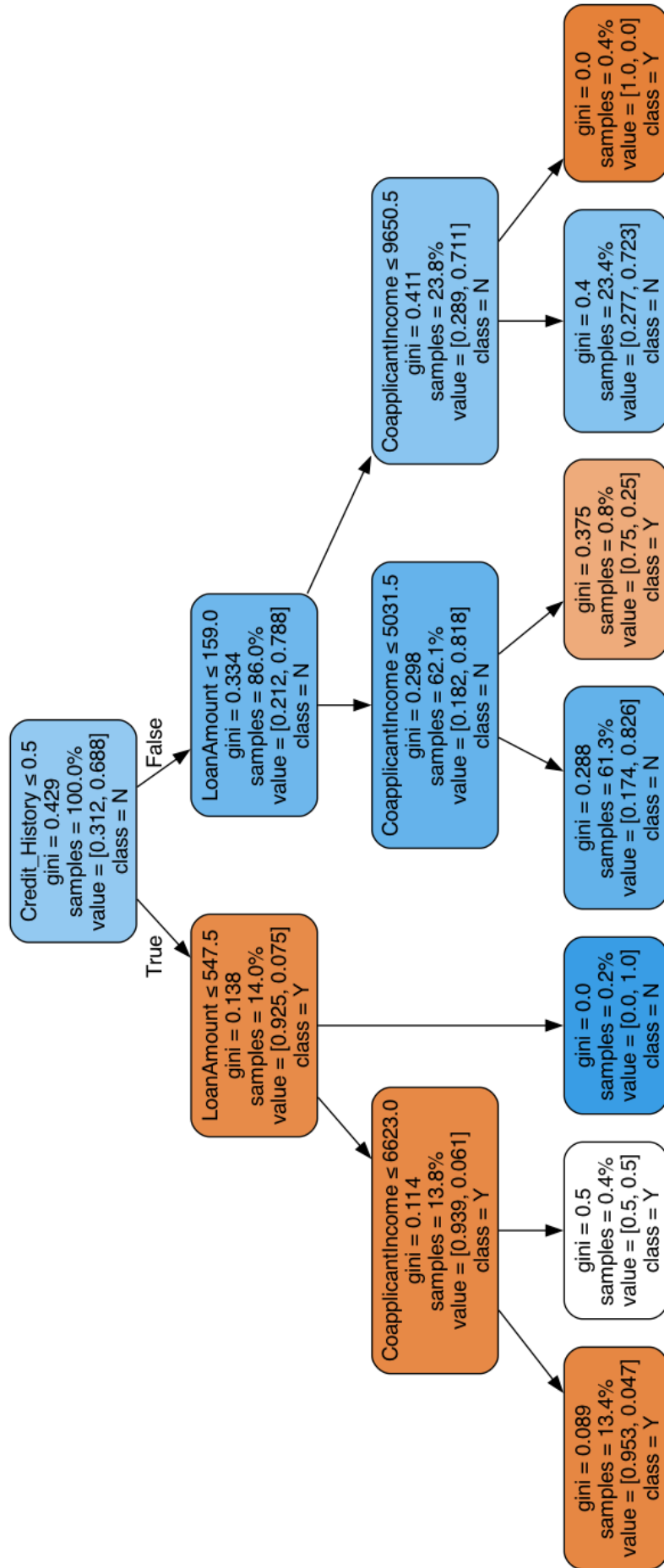


Figure 1.1: Decision tree classifier for loan applications.

algorithm involves building the tree. Decision trees are trained through a recursive approach that creates subsets from the initial training dataset [20].

In general, the construction of a decision tree algorithm starts at the root node, which is the topmost node of the tree. At this point, the entire dataset is considered. The algorithm selects the best feature to split the data. This selection can be based on measurements such as gini impurity or information gain, which measures how well a feature can separate the data into classes. Based on the selected feature, the dataset is split into subsets based on each unique value of that feature. Each subset represents a branch of the tree. For example, if the best feature for splitting is Credit History, and we split it into ≤ 0.5 and > 0.5 , then two branches are created from the root node.

This process of selecting a feature and splitting it continues recursively for each branch until either a stopping criterion is met or all the data points in a branch belong to the same class. After the tree is constructed, it can classify new data. Starting from the root, a data point follows the path through the tree based on its feature values until it reaches a leaf node. The label associated with this leaf node will be the prediction or the label for this new, unlabeled data point.”

1.4.1 ID3 algorithm

The ID3 algorithm is a machine learning algorithm that builds a decision tree from a given training set. The resulting tree predicts the class for new data points (from the test set) not used during the training phase. ID3 constructs a tree recursively by computing the information gain (IG) at each step. The feature with the highest IG is chosen as the decision node, and the dataset is then divided into subsets based on that feature. Child nodes are created for each subset, and the process is repeated recursively until all data points in the child nodes belong to the same class.

The nodes where predictions are made are called leaf nodes, these nodes usually contain elements that belong to one class. Leaf nodes are assigned a label. After constructing the decision tree, these nodes can be used to predict the class for the data points in a test set. [21].

First, it is important to understand and measure the entropy, which is used to calculate the ”purity” in a given dataset, in order to calculate IG. For a binary classification problem with two classes, 0 and 1. If all data points in the set belong to the same class, the entropy for that dataset is zero. If the data points of the dataset are equally represented by their corresponding classes, then the entropy is one. The entropy of a dataset can be calculated by the Equation 1.22

$$\text{Entropy}(S) = - \sum_{i=1}^C p_i \cdot \log_2 p_i \quad (1.22)$$

Where S is the training set where each data point x_i has a corresponding label y_i , which represents the class to which each x_i belongs to.

Information gain is used to determine which feature, at each step, best splits the dataset into subsets such that the subsets after the split are more homogenous (pure) and have lower entropy. IG is calculated using the following Equation (1.23)

$$IG(S, A) = \text{Entropy}(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \quad (1.23)$$

The Eq. 1.23 is defined as follows:

- $\sum_{v \in V}$ where each v represents a unique value from the set of all unique values V that feature A can take. Here, v is a specific value of feature A .
- S_{val} is the subset of S based on the values of A
- $|S_{val}|$ is the total number of items in S_{val}
- $|S|$ is the total number of items in S

The pseudocode of ID3 algorithm is given in Algorithm 4

Algorithm 4 ID3 Algorithm

```

1: function ID3(examples, target_attribute, attributes)
2:   Create a new root node
3:   if all examples have the same value for the target_attribute then
4:     return root node labeled with the target_attribute value
5:   else if attributes is empty or all examples have the same value for
   attributes then
6:     return root node labeled with the most common value of the
   target_attribute in examples
7:   else
8:      $A \leftarrow \arg \max_{A \in \text{attributes}} IG(\text{examples}, A)$  1.23
9:     Set the root's decision attribute to  $A$ 
10:    for each value  $v_i$  of  $A$  do
11:      Create a new branch for the rule  $A = v_i$ 
12:       $\text{examples}_{v_i} \leftarrow$  subset of examples where  $A = v_i$ 
13:      if  $\text{examples}_{v_i}$  is empty then
14:        Add a leaf node with the most common value of the
   target_attribute in examples
15:      else
16:        Add the subtree ID3( $\text{examples}_{v_i}$ , target_attribute,  $\text{attributes} \setminus$ 
    $\{A\}$ ) to the branch for the rule  $A = v_i$ 
17:      end if
18:    end for
19:  end if
20:  return root
21: end function
22:  $T \leftarrow$  ID3(examples, target_attribute, attributes)

```

1.4.2 CART algorithm

CART is one of the most well-known algorithms for decision trees. CART stands for Classification and Regression Trees, first published by Olshen, Breiman, Stone, and Friedman in 1984 [22].

The CART algorithm can process both categorical and numerical features and be used for classification and regression tasks. During the training phase (creation of the decision tree), the algorithm creates a binary tree, where each node splits the training data into two branches (subsets) based on the selected feature. The CART algorithm's data-splitting process is iterative and starts at the root node.

The training data is initially divided into two subsets (children), and then each child is further divided into sub-nodes. If there is no stopping condition explicitly defined. In that case, the algorithm will naturally stop splitting the node into further child nodes until the node achieves a pure state means all data points in that node belong to the same class, or there is only one data point in it.

The problem with this approach is that the trees grow large, and they usually overfit the training data. Pre-pruning techniques can be used to overcome this issue. Some of these techniques involve settings like specifying the maximum depth of the tree, or specifying a minimum for the number of data points in a node, or setting a maximum number of nodes a tree can reach [23, 24]. The trees that are built using the CART algorithm are then pruned using techniques such as cost-complexity pruning which is discussed further in this section.

Given a training dataset \mathbf{X} with training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where \mathbf{x}_i is the feature vector and y_i represents its corresponding label which belongs to the set $\mathcal{S} = \{\text{class1}, \text{class2}, \dots, \text{class } c\}$ [25]. The dataset S is split into two subsets S_l and S_r when selecting the j -th feature from feature vector \mathbf{x}_i and a splitting value t . The Gini index plays a crucial role in evaluating the split point which is calculated using Equation 1.24:

$$\text{Gini}(S, j, t) = \frac{|S_l|}{|S|} \text{Gini}(S_l) + \frac{|S_r|}{|S|} \text{Gini}(S_r) \quad (1.24)$$

The gini impurity measure for dataset S is calculated using the Equation 1.25

$$\text{Gini}(S) = 1 - \sum_{i=1}^c \left(\frac{|S^i|}{|S|} \right)^2 \quad (1.25)$$

where S^i is the number of samples in S belonging to class i , and $|S|$ denotes the total number of samples in the dataset S . The CART algorithm finds the best split by minimizing the gini index. The pseudocode for the CART algorithm is presented in Algorithm 5.

Cost complexity pruning

The algorithm has two parts [26, 22]. First, from the initial tree (T_0 - the tree before pruning), the algorithm generates other subtrees ($T_1, T_2, T_3, \dots, T_k$) by pruning branches from the initial tree (T_0). T_0 is the original tree, and T_k is the most pruned tree in the generated sequence.

Each subtree (T_{k+1}) is created by pruning (replacing subtrees with the appropriate leaf nodes) from the previous tree (T_k). In each step, the algorithm selects the sub-tree for pruning in the tree T_k , which minimizes the increase in error rate for each node removed. For selecting the subtree which minimizes the error rate, the Equation 1.26 is used:

$$\alpha = \frac{\epsilon(\text{pruned}(T, t), S) - \epsilon(T, S)}{|\text{leaves}(T)| - |\text{leaves}(\text{pruned}(T, t))|} \quad (1.26)$$

where $\epsilon(T, S)$ is the evaluation of the tree T over the sample S (the error rate). The sample S is the validation set used in the pruning process to evaluate the error rate of the tree. $|\text{leaves}(T)|$ is the number of leaf nodes in T , $\text{pruned}(T, t)$ represents a pruned tree created by replacing the node t in T with a leaf node, meaning the subtree with its root node t is replaced with a leaf node which has the prediction of the majority class in the node t (in case of classification).

The α value is computed for each possible subtree in T_k , and the one that has the lowest α value will be pruned. The result of this pruning will be the tree T_{k+1} .

In the second part, from the set of trees generated in the first part ($T_1, T_2, T_3, \dots, T_k$), the tree that has the lowest generalization error on a validation set is selected as the final pruned tree.

Algorithm 5 CART Algorithm

```

1: Input: Training dataset  $D$ , Feature set  $F$ 
2: Output: Decision tree  $T$ 
3: Notation:  $(j^*, v^*)$  represents the splitting rule, the  $v^*$ -th value of the  $j^*$ -th feature
4: function CART( $D, F$ )
5:   Create a node  $N$ 
6:   if stopping criterion is satisfied then
7:     return  $N$  labeled with the majority class in  $D$ 
8:   else
9:     for each feature  $j \in F$  do
10:      for each unique value  $v$  of feature  $j$  do
11:        Compute the Gini index  $Gini(D, j, v)$ 
12:      end for
13:    end for
14:    Select the best split  $(j^*, v^*) = \arg \min Gini(D, j, v)$ 
15:    Label  $N$  with  $(j^*, v^*)$ 
16:    Split  $D$  into  $D_l$  and  $D_r$  according to  $(j^*, v^*)$ 
17:     $N.left \leftarrow \text{CART}(D_l, F)$ 
18:     $N.right \leftarrow \text{CART}(D_r, F)$ 
19:  end if
20:  return  $N$ 
21: end function
22:  $T \leftarrow \text{CART}(D, F)$ 

```

1.4.3 C4.5 algorithm

This algorithm is an improvement of the ID3 algorithm, introduced by the same author [27, 26]. The training phase is similar to ID3, where the tree is constructed recursively using a top-down approach. The algorithm stops when all data points belong to the same class or a stopping condition is met.

Some improvements and characteristics of this algorithm include processing both numeric and categorical attributes. Additionally, the same attribute can be selected multiple times as a splitting condition with different threshold values. The algorithm uses a post-pruning technique called error-based pruning. Additionally, the algorithm uses “gain ratio” as a splitting criterion (i.e., select the best splitting rule), which is an improvement over the ID3 algorithm and is defined in Equation 1.27.

$$\text{GainRatio}(a) = \frac{\text{InformationGain}(a)}{\text{Entropy}(a)} \quad (1.27)$$

The formula above computes the Gain Ratio for a specific attribute, denoted by a . The gain ratio is a normalization of the Information Gain. It improves the splitting conditioning by penalizing features with high information gain just because they have many unique possible values. These unique values split the data into many small, homogeneous groups, which results in high information gain. By normalizing the information gain as shown in the Equation 1.27, this modification leads to better feature selection and improved accuracy.

Pruning in C4.5

Pessimistic error pruning is a technique used for post-pruning decision trees to reduce complexity by pruning subtrees while minimizing the increase in error. This is a bottom-up approach, starting from leaf nodes and computing the pessimistic error of the subtree in each step, both with and without pruning [27, 28, 23].

The pessimistic error of a subtree without pruning is defined as follows: For each non-leaf node (internal node) v in the tree, the pessimistic error is computed for the subtree with the root node v and for the pruned node v (i.e., the subtree is replaced with a leaf node v itself). The formula that computes the pessimistic error without pruning is defined in Equation 1.28

$$E_{\text{pess}}(T(v)) = \sum_{l \in L(v)} E(l) + \frac{|L(v)|}{2} \quad (1.28)$$

Where all the errors of leaf nodes of the subtree $T(v)$ are summed up and a constant value of $\frac{|L(v)|}{2}$ is added, where $|L(v)|$ is the number of leaf nodes rooted at v . The pessimistic error with pruning the subtree is computed using the Equation 1.29:

$$E_{\text{pess}}(v) = E(v) + \frac{1}{2} \quad (1.29)$$

This equation computes the error of node v (as if it were a leaf node), and it adds a constant value of $\frac{1}{2}$. These errors are computed for all nodes, and in each internal node (non-leaf nodes), the subtree is pruned if $E_{\text{pess}}(v) \leq E_{\text{pess}}(T(v))$. This means that if the pessimistic error after pruning the subtree is smaller or equal to the error before pruning, then the pruning of the subtree is conducted. If not, the subtree is not pruned. The computational complexity is, at worst, linear.

The C4.5 algorithm uses a modified version of this pruning algorithm, which ensures that the subtree is pruned if the error after pruning is within one standard error of the original tree without pruning. The formula that computes the

modified pessimistic error is defined in Equation 1.30:

$$\epsilon'(T, S) = \epsilon(T, S) + \frac{2|S|}{|\text{leaves}(T)|} \quad (1.30)$$

Where $\epsilon'(T, S)$ is the error of subtree T and the samples S . $|\text{leaves}(T)|$ is the number of leaves the subtree has. The modified Equation 1.31, which decides if the subtree is pruned, is defined as follows:

$$\epsilon'(\text{pruned}(T, t), S) \leq \epsilon'(T, S) + \frac{|S|\epsilon'(T, S) \cdot (1 - \epsilon'(T, S))}{|\text{leaves}(T)|} \quad (1.31)$$

Where $\epsilon'(\text{pruned}(T, t), S)$ represents the pessimistic error after pruning subtree T and replacing it with leaf node t , $E(T, S)$ is the error rate of the subtree before pruning with set S . The extra margin of error is added, which ensures the error after pruning is within one standard error from the original tree before pruning.

1.4.4 Decision tree algorithm – ID3 solved example

To better understand the training phase of the algorithm, a dataset created only for illustration purposes in Figure 1.2 is used. As we explained earlier, decision tree algorithms recursively decide the best attribute and create decision nodes based on the selected attribute. To select the best attribute in each step, the

Example	Attribute 1	Attribute 2	Attribute 3	Label
1	TRUE	Sunny	Morning	NO
2	TRUE	Sunny	Morning	YES
3	FALSE	Sunny	Evening	NO
4	FALSE	Sunny	Evening	NO
5	FALSE	Rainy	Evening	YES
6	FALSE	Rainy	Morning	YES
7	TRUE	Sunny	Morning	YES
8	TRUE	Rainy	Evening	YES
9	FALSE	Rainy	Morning	YES
10	FALSE	Rainy	Morning	YES

Figure 1.2: Dataset for decision tree example

algorithm computes the information gain for each attribute and selects the one with the highest information gain using Equation 1.23:

The calculation of information gain for attribute one, two, and three are shown below:

$$S = [7+, 3-]$$

$$\text{Entropy}(S) = -\frac{7}{10} \log_2 \frac{7}{10} - \frac{3}{10} \log_2 \frac{3}{10} = 0.8813$$

$$S_{\text{True}} = [3+, 1-]$$

$$\text{Entropy}(S_{\text{True}}) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0.8113$$

$$S_{\text{False}} = [4+, 2-]$$

$$\text{Entropy}(S_{\text{False}}) = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.9182$$

$$\text{Gain}(S, \text{Attribute 1}) = \text{Entropy}(S) - \sum_{v \in \{\text{True}, \text{False}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S, \text{Attribute 1}) = \text{Entropy}(S) - \frac{4}{10} \text{Entropy}(S_{\text{True}}) - \frac{6}{10} \text{Entropy}(S_{\text{False}})$$

$$\text{Gain}(S, \text{Attribute 1}) = 0.8813 - \frac{4}{10} \times 0.8113 - \frac{6}{10} \times 0.9182 = 0.00586$$

Attribute 2

$$S = [7+, 3-]$$

$$\text{Entropy}(S) = -\frac{7}{10} \log_2 \frac{7}{10} - \frac{3}{10} \log_2 \frac{3}{10} = 0.8813$$

$$S_{\text{Sunny}} = [2+, 3-]$$

$$\text{Entropy}(S_{\text{Sunny}}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.9709$$

$$S_{\text{Rainy}} = [5+, 0-]$$

$$\text{Entropy}(S_{\text{Rainy}}) = 0.0$$

$$\text{Gain}(S, \text{Attribute 2}) = \text{Entropy}(S) - \sum_{v \in \{\text{Sunny}, \text{Rainy}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S, \text{Attribute 2}) = \text{Entropy}(S) - \frac{5}{10} \text{Entropy}(S_{\text{Sunny}}) - \frac{5}{10} \text{Entropy}(S_{\text{Rainy}})$$

$$\text{Gain}(S, \text{Attribute 2}) = 0.8813 - \frac{5}{10} \times 0.9709 - \frac{5}{10} \times 0 = 0.48545$$

Attribute 3

$$S = [7+, 3-]$$
$$\text{Entropy}(S) = -\frac{7}{10} \log_2 \frac{7}{10} - \frac{3}{10} \log_2 \frac{3}{10} = 0.8813$$

$$S_{\text{Morning}} = [5+, 1-]$$
$$\text{Entropy}(S_{\text{Morning}}) = -\frac{5}{6} \log_2 \frac{5}{6} - \frac{1}{6} \log_2 \frac{1}{6} = 0.6500$$

$$S_{\text{Evening}} = [2+, 2-]$$
$$\text{Entropy}(S_{\text{Evening}}) = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1$$

$$\text{Gain}(S, \text{Attribute 3}) = \text{Entropy}(S) - \sum_{v \in \{\text{Morning}, \text{Evening}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S, \text{Attribute 3}) = \text{Entropy}(S) - \frac{7}{10} \text{Entropy}(S_{\text{Morning}}) - \frac{3}{10} \text{Entropy}(S_{\text{Evening}})$$

$$\text{Gain}(S, \text{Attribute 3}) = 0.8813 - \frac{7}{10} \times 0.6500 - \frac{3}{10} \times 1 = 0.0913$$

Comparison of gains

$$\text{Gain}(S, \text{Attribute 1}) = 0.00586$$

$$\text{Gain}(S, \text{Attribute 2}) = 0.48545 \quad (\text{Maximum Gain})$$

$$\text{Gain}(S, \text{Attribute 3}) = 0.0913$$

Since attribute one has the highest information gain among the three attributes, it is selected as the root node.

Attribute 2 (second split)

$$S = [2+, 3-]$$
$$\text{Entropy}(S_{\text{attribute2}}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.9709$$

$$S_{\text{True}} = [2+, 1-]$$
$$\text{Entropy}(S_{\text{True}}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183$$

$$S_{\text{False}} = [2+, 0-]$$
$$\text{Entropy}(S_{\text{False}}) = 0.0$$

$$\text{Gain}(S, \text{Attribute 2}) = \text{Entropy}(S_{\text{Attribute2}}) - \sum_{v \in \{\text{True}, \text{False}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S, \text{Attribute 2}) = \text{Entropy}(S_{\text{Attribute2}}) - \frac{3}{5} \text{Entropy}(S_{\text{True}}) - \frac{2}{5} \text{Entropy}(S_{\text{False}})$$

$$\text{Gain}(S, \text{Attribute 2}) = 0.9709 - \frac{3}{5} \times 0.9183 - \frac{2}{5} \times 0.0 = 0.28217$$

Attribute 3 (Second Split)

$$S = [2+, 3-]$$

$$\text{Entropy}(S_{\text{Attribute2}}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.9709$$

$$S_{\text{Morning}} = [2+, 1-]$$

$$\text{Entropy}(S_{\text{Morning}}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183$$

$$S_{\text{Evening}} = [0+, 2-]$$

$$\text{Entropy}(S_{\text{Evening}}) = 0.0$$

$$\text{Gain}(S, \text{Attribute 3}) = \text{Entropy}(S_{\text{Attribute2}}) - \sum_{v \in \{\text{Morning}, \text{Evening}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S, \text{Atrb. 3}) = \text{Entropy}(S_{\text{Attribute2}}) - \frac{3}{5} \text{Entropy}(S_{\text{Morning}}) - \frac{2}{5} \text{Entropy}(S_{\text{Evening}})$$

$$\text{Gain}(S, \text{Attribute 3}) = 0.9709 - \frac{3}{5} \times 0.9183 - \frac{2}{5} \times 0 = 0.41992$$

Comparison of Gains (second split)

$$\text{Gain}(S_{\text{Attribute 1}}, \text{Attribute 2}) = 0.28217$$

$$\text{Gain}(S_{\text{Attribute 3}}, \text{Attribute 3}) = 0.41992 \quad (\text{Maximum Gain})$$

Attribute 1 (third split)

$$S_{\text{True}} = [2+, 1-]$$

The entropy for this subset is:

$$\text{Entropy}(S_{\text{True}}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183$$

Since all remaining examples have the value TRUE for Attribute 1, there are no possible splits. The algorithm will stop here, and the resulting decision tree is shown in Figure 1.3.

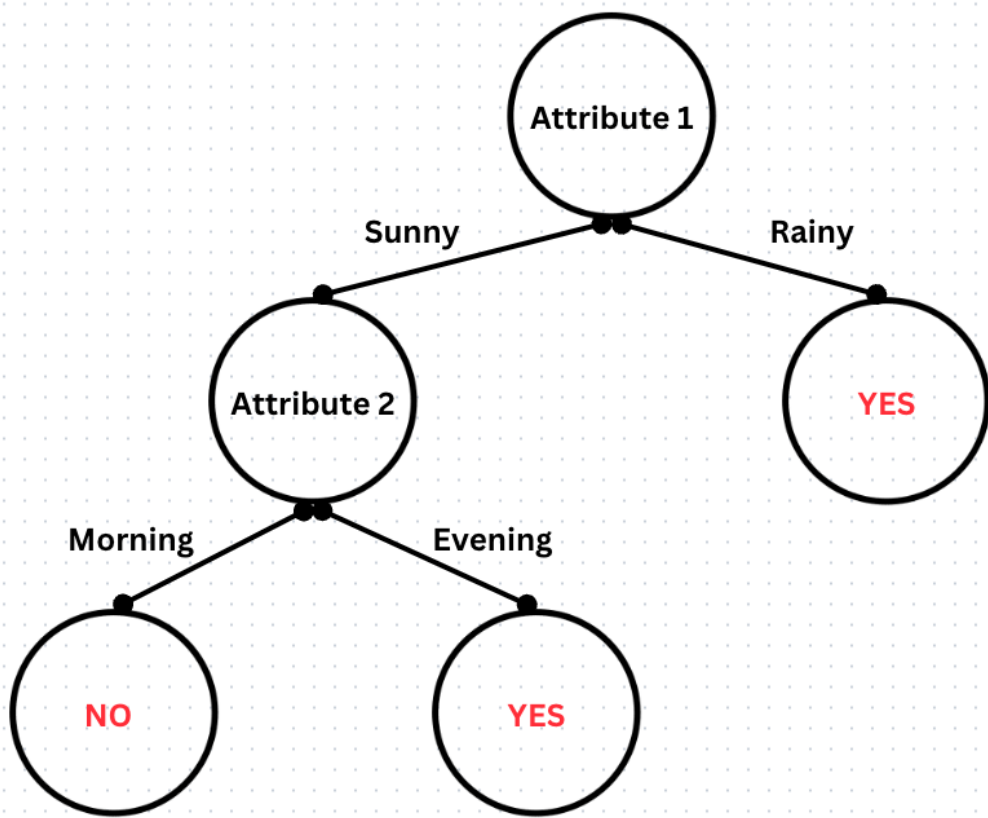


Figure 1.3: Decision tree created by ID3 algorithm using a simple dataset.

1.4.5 Other filter models

Fisher's Score

The fisher score is a technique used to measure how well a numeric feature can help differentiate between different categories or classes in a given dataset. The higher the fisher score, the better the feature is at separating these classes [9].

For example, if we want to measure how well a feature like "length_of_email" can differentiate between spam and non-spam emails, we can use the Fisher score to measure this. Fisher's score can be calculated using the Equation 1.32:

$$F = \frac{\sum_{j=1}^k p_j (\mu_j - \mu)^2}{\sum_{j=1}^k p_j \sigma_j^2} \quad (1.32)$$

Where:

- μ_j and σ_j : Represent the average (mean) and spread (standard deviation) of the data points for a specific class, respectively. An example of a class would be the "spam" class.
- p_j : Represents the probability of data points belonging to a specific class.
- μ : Is the average (mean) of the feature that is being considered.
- k : Represents the total number of classes within the target variable.

The numerator measures how well a feature can differentiate between classes. The denominator measures how much the data for each class is spread out.

Fisher's linear discriminant

Fisher's linear discriminant method goal is to determine one or more vectors in the feature space that maximize the separation between classes while minimizing the variation (i.e., the spread of data within each class). This is done by projecting the data into these vectors where the classes are the most separated, which then can be used for classification tasks [9].

For a binary classification problem, the goal is to determine the direction \mathbf{W} , which can be done by maximizing fishers score $FS(\mathbf{W})$, which is a measure of how well the classes are separated along a given direction \mathbf{W} . This involves using the means and the covariances within each class as described in the Equation 1.33:

$$FS(\mathbf{W}) = \frac{[\mathbf{W} \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)]^2}{\mathbf{W}^T (p_0 \Sigma_0 + p_1 \Sigma_1) \mathbf{W}} \quad (1.33)$$

Where $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ represent the row vectors of the average values of the samples in the two classes. Σ_0 and Σ_1 are the $d \times d$ covariance matrices for the two classes. p_0 and p_1 represent the fraction of the total samples in each class.

To find the optimal direction \mathbf{W}^* , we need to maximize the $FS(\mathbf{W})$ function. To find the optimal \mathbf{W}^* that maximizes the Fisher score, the following Equation is used 1.34

$$\mathbf{W}^* \propto (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)(p_0 \Sigma_0 + p_1 \Sigma_1)^{-1} \quad (1.34)$$

1.5 Ensemble methods

Different models can result in different predictions for the same test instance. This difference may be caused by the unique characteristics of each classifier, errors in the data, or the model's sensitivity to small changes in the training data [9]. To overcome these issues, ensemble methods can be used, which combine the outcomes of multiple classifiers, resulting in better prediction accuracy.

Algorithm 6 Ensemble Classifier Algorithm

```
EnsemblePredictions(Training Data:  $D$ , Base Algorithms:  $A_1, A_2, \dots, A_r$ ,  
Test Data:  $T$ )  
   $j = 1$   
  while termination criteria is not met do  
    Select an algorithm  $Q_j$  from  $A_1, A_2, \dots, A_r$   
    Create a modified training dataset  $D_j$  based on  $D$   
    Apply the selected algorithm  $Q_j$  to  $D_j$  to create a model  $M_j$   
     $j = j + 1$   
  end while  
  for each test instance  $t \in T$  do  
    Combine predictions from all models  $M_1, M_2, \dots, M_{j-1}$  to predict the label  
    for  $t$   
  end for
```

The Algorithm 6 represents the general form of ensemble algorithms. The idea behind this generic algorithm is to iteratively select a classification algorithm from the set of available algorithms. This set can be a set of different classification algorithms selected in each step (such as Decision Trees, SVM, etc.), or it can have the same classification algorithm applied to different subsets of the training data. So, in each iteration, a new model is trained using an algorithm selected from a set of base algorithms.

After all the models are trained, the model predicts the test instances by combining the predictions from all the models. There are several ways to combine these predictions. One way is to use the most frequent prediction from all the models as the final prediction.

How does ensemble analysis work

To understand why ensemble techniques work, it is important to understand the main types of errors that are made by classifiers. There are three types of errors.

Bias Error occurs when models make assumptions about the data. The model has a high bias if the classifier makes mistakes consistently over a particular set of test instances, even if it is trained on different subsets of the training data. For instance, if the true decision boundary that perfectly classifies the data is linear, using a trained decision tree model for classification on this data will always result in bias since the model is unable to approximate the true decision boundary. This error inherited in the model is known as bias.

Variance error occurs when a classifier behaves differently for different training data sets. For example, sometimes, there is not much training data available, and the classifier needs more data to learn the underlying patterns, which results in a

high variance error. Otherwise, with more data available for training, the classifier could improve and perform better. In other cases, for example, when training decision trees, different subsets of the training data can result in different trees. Resulting in different predictions for the same test instance. If this difference is high, it means that the model has high variance since it is sensitive to changes of the training data.

Variance is related to the concept of overfitting, when the model has high variance, it means that the model makes correct predictions on the training data, but it is not able to make correct predictions when predicting on test data.

Noise is the error that happens because of incorrect assignments (labels) in the training data, which can happen because of human error or inaccurate measurements, etc. There is not much that can be done to fix the noise since it is more related to the data quality.

When training a model, the choice of parameters plays a crucial role in determining the variance and bias of the model. For instance, if pruning the decision tree leads to a better model (i.e., performs better on the test set), the model's variance is reduced. However, this decision also increases the model's bias, underscoring the importance of the choice of parameters in model training. The bias is more significant since the model makes assumptions about the decision boundary's simplicity. So, the choice of parameters is usually a trade-off between variance and bias.

Ensemble methods can also be used to reduce variance and bias. For example, when using a combination of multiple trees to make a prediction, consider each tree has a probability 0.8 of making a correct prediction for a test instance. Assuming the trees are independent, the probability of correct prediction for the ensemble classifier, using a voting strategy will be around 90 %. Which is improved, and variance is reduced. This can be calculated using the following Equation 1.35:

$$(0.8^3 + \binom{3}{2} \cdot 0.8^2 \cdot 0.2) \cdot 100 \quad (1.35)$$

1.5.1 Bagging

Bagging is a technique that aims to reduce the variance of a model. This is achieved by training multiple models on different subsets of the training data. The principle behind bagging is that if the variance of the prediction is σ^2 , by averaging k independent classifiers that work on different subsets of the training data with replacement, the variance will be reduced to $\frac{\sigma^2}{k}$ [9].

In order to create such classifiers, bagging uses a technique called bootstrapping. In bootstrapping, a new set of training data is created by randomly selecting data points from the original training data. The size of this generated or sampled set is close to the original dataset. Since it is sampled with replacement, the set might contain duplicates. A new bootstrapped dataset is created for each classifier, and these sets will be similar to each other. After each independent classifier's training process is completed, the classification of a test instance is decided by the majority vote of all the classifiers. This approach will result in a reduced variance of the model.

It is important to understand that bagging does not reduce bias, so when cre-

ating a model using the bagging technique, it is important to focus on parameters and methods that reduce the bias since the variance will be handled by bagging.

1.5.2 Random forests

One of the most popular classification models used with bagging is decision trees. The problem with using decision trees with bagging is that the trees created using bootstrapping will be similar and highly correlated. This will limit the error reduction induced using bagging techniques [9].

To overcome this issue, a randomized decision tree model is used for each classifier in the ensemble. This model is called random forest. Using random forests, the trees will be less correlated. Random forests are usually more accurate than using bagging with decision trees.

For each node, random forest selects a subset of attributes randomly and then splits based on the best attributes from this subset. Using a large number of attributes in this subset will result in highly correlated trees and better accuracy. On the other hand, the smaller the number of attributes in this subset, the less correlated the trees will be, but the accuracy may be compromised. The goal is to find a good trade-off between tree correlation and model accuracy by selecting the best number of attributes in this subset. The formula that is used to get the best number of attributes is $m = \log_2(d) + 1$ where d is the number of total attributes available, and m is the number of attributes in the subset.

Random Forests are used to reduce the variance. However, the bias may be increased because the number of attributes that can be used to split the data is limited (reduced). This can be a problem if the number of attributes is small. In real-world applications, random forests perform better than bagging, and they are similar to boosting in terms of performance. Random forests are also effective in handling outliers and noise in the data.

1.5.3 Boosting

Boosting is a technique that aims to reduce the bias of the model. Each data point used for the training process has a corresponding weight in boosting. The base classifiers are trained iteratively. Each of them has a set of weights assigned from the previous iteration. The weights are updated iteratively based on the model's performance on the training data. So, weights in each iteration result from the previous iteration (excluding the first iteration). The focus is on the misclassified data points.

In each iteration, the data points that have been misclassified will be given a higher weight for the next iteration. The idea behind this approach is that it assumes that the errors in these particular data points result from the model's bias. By assigning them a higher weight, the assumption is that the upcoming model will correct these mistakes in the next iteration. The most popular algorithm that uses a Boosting technique is AdaBoost.

The Algorithm 7, represents AdaBoost algorithm for a binary classification task. The algorithm initializes weights for each data point in the training set. The initial weights are set to $1/n$ where n represents the number of total data points in the training set. All data points will have the same weight initially.

Algorithm 7 AdaBoost Algorithm

```
1: procedure ADABOOST( $D$  (Data),  $A$  (Base Classifier),  $T$  (Number of Iterations))
2:   Initialize  $t \leftarrow 0$ 
3:    $n \leftarrow$  number of data points in  $D$ 
4:   Initialize  $W_1(i) = 1/n$  for each data point  $i$  in  $D$ 
5:   repeat
6:      $t \leftarrow t + 1$ 
7:     Determine weighted error rate  $\varepsilon_t$  using classifier  $A$  on  $D$  with weights  $W_t(\cdot)$ 
8:      $\alpha_t \leftarrow \frac{1}{2} \ln \left( \frac{1-\varepsilon_t}{\varepsilon_t} \right)$ 
9:     for each instance  $X_i$  in  $D$  do
10:      if  $X_i$  is misclassified then
11:         $W_{t+1}(i) \leftarrow W_t(i) \cdot e^{\alpha_t}$ 
12:      else
13:         $W_{t+1}(i) \leftarrow W_t(i) \cdot e^{-\alpha_t}$ 
14:      end if
15:    end for
16:    for each instance  $X_i$  in  $D$  do
17:      Normalize  $W_{t+1}(i) \leftarrow \frac{W_{t+1}(i)}{\sum_{j=1}^n W_{t+1}(j)}$ 
18:    end for
19:  until ( $t \geq T$ ) OR ( $\varepsilon_t = 0$ ) OR ( $\varepsilon_t \geq 0.5$ )
20:  Classify test instances using the ensemble of classifiers with weights  $\alpha_t$ 
21: end procedure
```

In each iteration, in case of misclassification of an $i - th$ data point, its corresponding weight will be increased using the Equation 1.36:

$$W_{t+1}(i) \leftarrow W_t(i) \cdot e^{\alpha_t} \quad (1.36)$$

Otherwise, in case of correct classification, its corresponding weight will be decreased using the Equation 1.37:

$$W_{t+1}(i) \leftarrow W_t(i) \cdot e^{-\alpha_t} \quad (1.37)$$

The α_t is the value used for updating the weights that are calculated in each iteration, using the Equation 1.38:

$$\alpha_t \leftarrow \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \quad (1.38)$$

where ε_t is the error rate of the ada boost classifier in iteration t .

The algorithm stops when the error on the training data is 0 ($\varepsilon_t = 0$), or the classifier's accuracy is less than a random classifier ($\varepsilon_t \geq 0.5$).

Boosting is affected by noise in the data. The assumption of the model is that the errors occur because of the bias of the model and not incorrectly labeled examples.

1.6 Educational system

1.6.1 Educational system in UK

Early Education In the UK, pre-schools are free, and includes kids beginning at the age of three and continuing until five years old [29, 30]. The goal is to help children improve their language and social skills. The study is 15 hours, 38 weeks a year.

Primary education includes age ranges for children under five years old (nursery / early education), infants from ages five to seven or eight, and juniors for children from eleven up to twelve years old. In Scotland, northern infants and juniors go to the same school (there is no differentiation in schools).

In Wales, Northern Ireland, and England, the transition from primary to secondary school occurs at age eleven, while in Scotland, it occurs at age 12. In England, some people go to middle schools before secondary schools, which are from ages eight to fourteen and are categorized as primary or secondary based on age. Primary school aims to provide a basic understanding of literacy, the basics of mathematics and science, and other courses.

In England, secondary education includes comprehensive schools where all students are accepted regardless of academic performance. This ensures that all children in the local area have access to education. Grammar schools have selective measures based on students' performance and are state-funded.

Another type of school is Academies, which are publicly funded but independent schools. They are free to change the curriculum, the pay on staff can be flexible and set by the school, the school can set the length of terms, and they are independent of the local government. Academic schools aim to create or transform existing schools into better-performing ones. Wales accepts students aged

eleven and beyond for secondary school. In Scotland, secondary schools are of a comprehensive type and last for six years. In some areas, there are two-year and four-year schools, too.

In Northern Ireland, secondary education consists of five obligatory years and two extra years if students want to get from GCSE/ Level 2 courses to Level 3. In most schools in England, Wales, and Northern Ireland, students get the GCSE (General Certificate of Secondary Education) qualification. In Scotland, the students take the Standard Grades qualification in general, but they can also further continue their education to get a National Qualification (NQ) Higher Grade, which takes another extra year of secondary education.

Further Education (FE) This type of school can be attended after completing compulsory education (primary and secondary education). Students over the age of sixteen can enter this type of school. It can involve basic courses to more highly specialized education. These schools mainly teach basics in literacy or other fundamental skills, A-Levels, which are studied over two years and are required for university and specialized courses (such as dancing collages, agriculture, IT, business, and engineering) and then attain a BTEC degree.

Higher Education (HE) Higher education in the UK consists of a bachelor's degree, which students can begin after finishing their Secondary School or A level (i.e., further education degrees). Full-time Bachelor's degrees take three years to complete. They are associated with 360 UK credits or 180 ECTS credits. Depending on the institution, there can be four-year programs and part-time studies. Some examples of bachelor's degrees are Bachelor of Arts (BA), Bachelor of Law(LLB), Bachelor of Education (B.Ed), and Bachelor of Science (BSC). These degrees are first-level qualifications.

Master's degree program can vary based on the institution. The degrees can broadly be categorized into taught master degrees (PGT) where there are regular classes and research. Common degrees are Master of Arts (MA), Master of Science (MS), and specialized degrees such as MEng (Master of Engineering), MFA (Master of Fine Arts), LLM(Master of Laws), etc. Research masters (PGR) degrees are one to two years, common degrees such as MPhil (Master of Philosophy) can be attained, which is a research-based degree that leads to further studies and opens a path to a PhD for students. Another type of Master's Degree is an MBA, which is related to management.

To attain a doctorate, students usually have to finish their master's (although it is not required in some subjects). To get a doctoral degree, students must research a specific field of study and write a thesis. PhD/DPhil degree is the highest degree that can be attained.

Other certificates and diplomas include Higher National Certificate (HNC) and Higher National Diploma (HND). Usually, it takes one year to study at a higher educational institution. Certificates like HE/Dip HE are equivalent to two years of studying a particular field, and postgraduate certificates PG Cert/PG Dip are equivalent to two years of studying a particular field.

1.6.2 Educational system in the US

The education system in the US is decentralized [31]. The states administer public schools, and the local governments remain the primary entities that administer

elementary and secondary education in the US. The national-level government does not have any control over the educational institutions. This decentralization is rooted in early history, where local governments shaped education based on their needs, priorities, and values.

Early childhood education Early education in the US mainly includes nursery schools (pre-schools) for children from 2 to 4, preparing them for kindergarten. And kindergartens are for children from ages 4-5. Other institutions include day-care centers, prekindergarten, and pre-schools. For low-income families, there is a federally funded program, Head Start, which offers free educational services for three and 4-year-old children. This ensures that children will enter kindergarten with the essential language skills.

Elementary and secondary school lasts for twelve years, including six to eight years of elementary school and four to six years of secondary school. Secondary school is divided into two levels: junior high school and the last four years, known as high school. Students often graduate high school by the age of 17 or 18. After high school, students may enter a two-year community college or a 4-year college or university. Depending on the state, obligatory attendance in schools varies; for example, by law, the required schooling ends by age 16 in 30 states. Some other states require schooling until the age of 17, and some until 18. Public schools are free and supported by taxpayers.

In the US, Colleges and universities offer undergraduate degrees, including bachelor's and associate's degrees, and graduate degrees, which include master's and doctorate degrees. Bachelor's degrees take four or more years of full-time study to complete. Community colleges provide associate degrees, which normally take two years of full-time study.

A master's degree requires one to two years after earning a bachelor's degree. A doctorate (Ph.D or equivalent) takes five to seven years after completing a bachelor's degree. However, this can vary depending on the school and topic of study.

Community college's role varies based on students' interests. Some use it as a transition from high school to university, while others might pursue technical training, short certificates, and diplomas in various academic fields.

In the US, there are also schools that are for-profit institutions referred to as proprietary schools. They can offer various training in specialized fields, such as business, administration, and computer technology, as well as degrees in several subject areas. They can offer undergraduate degrees, graduate degrees, and certificates, which can be obtained quickly, depending on the specialization or training.

Adult education is mainly for students 16 years and older who did not complete high school or for mature adults interested in additional education to improve their skills for personal reasons.

Other important education-related institutions include special education. This type of education is focused on students with disabilities. Most of these students attend regular public schools and attend classes with support and specific hours with qualified teachers. Gifted students are also specially treated with special programs in local public schools. The US separates religion from government. Therefore, schools can not teach religion. However, there are private religious schools where parents can send or homeschool their children.

The US offers a variety of institutions, such as public institutions, which state governments fund. Private universities and colleges, for-profit institutions owned by private owners, and vocational schools designed for specialized training help students work on entry-level jobs in professions that do not require an undergraduate degree.

2. Experimental results

2.1 Cluster analysis on graduation Rates in U.S. colleges

2.1.1 Introduction

This study aimed to analyze how graduation rates of different ethnic groups vary across U.S. colleges over three years. Along with graduation rates, other information such as tuition fees of universities, median household income of the county to in which they belong were also scrapped.

2.1.2 Data scraping

The dataset includes the names of four-year colleges and the annual graduation rates for each ethnic group over the three years. The ethnic groups present in the dataset are:

1. Us Nonresident
2. Hispanic / Latino
3. American / Indian or Alaska Native
4. Asian
5. Black or African American
6. Native Hawaiian or other pacific islander
7. White
8. Two or more races.

Several python scripts that utilize libraries such as selenium to scrape dynamic data from the web were used to scrape this data. The following flowchart in Figure 2.1 describes the process of scraping the data. To get the graduation rates and tuition fees, the website of the National Center For Educational Statistics <https://nces.ed.gov/> was used. This site requires a university id for data retrieval from any institution. For instance, each institution, such as Berkeley University, has its unique I.D. A document found at <https://secondnature.org/wp-content/uploads/Workbookv7-Sheet1.pdf> listed 3485 institution I.D.s. This document was used as an input for the scraping process, as described in Figure 2.1.

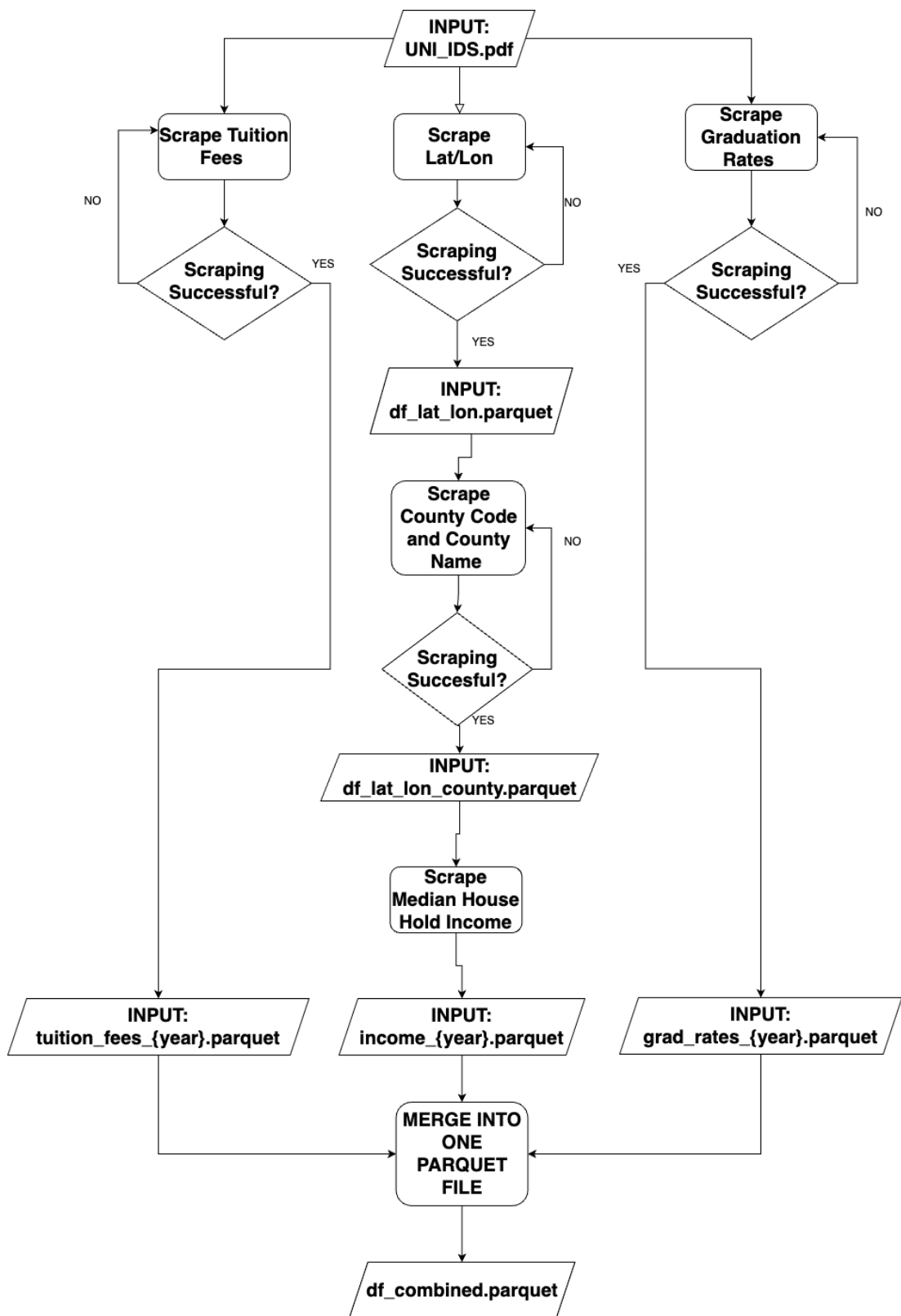


Figure 2.1: Flowchart of the scraping process

2.1.3 Scraping graduation rates

The main logic of scraping graduation rates from <https://nces.ed.gov/> is outlined in the code snippet in Figure 2.2. This Figure 2.2 shows the primary loop, which iterates through all university I.D.s and executes the scrape and preprocess grad rates method. This method saves the results in the file system for each university that is successfully scraped.

```
# Iterating through each element of UNI_IDS.pdf (converted to dataframe) input file
for index, row in initial_df_grad_rates.iterrows():
    try:
        # calling scrape_and_preprocess_grad_rates for each University ID of the input file
        processed_df = scrape_and_preprocess_grad_rates(row['IPEDS'])
        # Checking if data was successfully scraped (by checking if it's not empty)
        if not processed_df.empty:
            # Adding the IPEDS (University ID), and College Name to the graduation rates that were scraped.
            processed_df['IPEDS'] = row['IPEDS']
            processed_df['College Name'] = row['College Name']

            # Updating the dataframe in each step, with the current University that is scraped.
            save_to_parquet(processed_df, parquet_path)
```

Figure 2.2: Code Snippet for scraping graduation rates, main method

```
# Initializes a web browsing session, opens the web browser.
browser = initialize_browser()
try:
    # This variable is used to store the html tables in which the graduation rates are located in this web structure
    divs = None
    # This is the main URL from which graduation rates are being stored based on the University ID and Year
    url = f'https://nces.ed.gov/ipeds/datacenter/FacsimileView.aspx?surveyNumber=8&unitId={ipeds_value}&year={year}'
    # This method .get(url) navigates the browser to this URL
    browser.get(url)
    # The sleep method is used to wait until the page is loaded.
    time.sleep(7)

    # Condition to handle differences in webpage structure depending on the year.
    if year == 2019:
        # Selecting elements based on they're css class assigned.
        divs = browser.find_elements(By.CSS_SELECTOR, '.sc')
    else:
        # Selecting elements based on they're css class assigned.
        divs = browser.find_elements(By.CSS_SELECTOR, '.print-c.print-separator')
```

Figure 2.3: Code Snippet for scraping graduation rates, internal logic

In Figure 2.3, a code snippet from scrape and preprocess grad rates method is presented, with comments explaining the procedure for scraping university graduation rates. This snippet has the complete link where the graduation rates for each university and year can be accessed. It describes the main procedure for selecting HTML elements that contain the graduation table data. The script then processes these div elements on the page that holds the graduation rates and returns a data frame containing the graduation rates for each ethnic group available on this site. If an error occurs, it returns an empty data frame handled by the main method.

2.1.4 Scraping tuition fees

The data for tuition fees for each university was scraped from the same website “National Center for Educational Statistics” (<https://nces.ed.gov/>). The

```

# Initializes a web browsing session, opens the web browser.
browser = initialize_browser()
try:
    # This is the main URL from which graduation rates are being stored based on the University ID and Year
    url = f'https://nces.ed.gov/ipeds/datacenter/FacsimileView.aspx?surveyNumber=1&unitID={ipeds_value}&year={year}'
    # This method .get(url) navigates the browser to this URL
    browser.get(url)
    # The sleep method is used to wait until the page is loaded.
    time.sleep(7)
    #This line stores all tables in the page by their ID element. This is the table that contains Tuition Fees
    table = browser.find_element(By.ID, 'pricingTableAcademic')
    #This line stores all the rows from the table retrieved in previous step
    rows = table.find_elements(By.TAG_NAME, 'tr')

    # This list of data is used to create the dataframe after the tuition fees are successfully extracted
    data = []
    for row in rows:
        cells = row.find_elements(By.TAG_NAME, 'td') + row.find_elements(By.TAG_NAME, 'th')
        row_data = []
        for cell in cells:
            colspan = cell.get_attribute('colspan')
            # if colspan is None, it means there's only one column in the table
            if colspan is None:
                row_data.append(cell.text)
            else:
                # if colspan is NOT None, it means that the university has three different types of Tuition Fees
                for i in range(int(colspan)):
                    row_data.append(cell.text)
        data.append(row_data)

```

Figure 2.4: Code Snippet for scraping tuition fees

main logic of scraping tuition fees is the same as for graduation rates (see Figure 2.2). To scrape tuition fees, the method `scrape_and_preprocess_tuition` is used, described in the code snippet in Figure 2.4.

The script is similar to the graduation rates scraping method, here, the tables are extracted by their I.D. in their web structure, and then the tuition fees are extracted for each table row depending on the number of columns. Some universities provide only one tuition fee, while others have three types of fees depending on the student's residency. These are the following types of tuition fees available on the website:

1. In-district tuition and fees
2. In-state tuition and fees
3. Out-of-state tuition and fees

After extracting the data from the web, the script continues to create a data frame from the extracted tuition fees.

2.1.5 Scraping latitude and longitude

The geopy library is used to scrape each university's latitude and longitude coordinates. This library retrieves the latitude and longitude using the "College Name" variable extracted from the input file, `UNI_IDS.pdf`.

The code snippet in Figure 2.5 shows the main logic for scraping each university's latitude and longitude coordinates. The script uses the geopy library to retrieve the latitude and longitude of each university. The script then saves the results in a CSV file for each successfully scraped university.

The code snippet in Figure 2.5 shows that the `scrape_geo_data` method takes a data frame that is converted from the input `UNI_IDS.pdf` as its input. For each row in this data frame, it passes the College Name to the `.getcode` method,

```

def scrape_geo_data(dataframe):
    if not {'College Name'}.issubset(dataframe.columns):
        raise ValueError("DataFrame must contain 'College Name' column.")

    # Initialize the geolocator object
    geolocator = Photon(user_agent="measurements")

    # Add 'latitude' and 'longitude' columns to the input dataframe and initialize them as empty
    dataframe['latitude'] = np.nan
    dataframe['longitude'] = np.nan

    # For each university in UNI_IDS.pdf (converted to dataframe data)
    for index, row in dataframe.iterrows():
        # Check if the 'College Name' in the row is not a string
        if not isinstance(row['College Name'], str):
            # Print a message that this College cannot be processed since it is not a string
            print(f"Skipping row {index} due to invalid data type.")
            continue

        try:
            address = geolocator.geocode(row['College Name'])
            # Check if an address object was returned by the geocoder
            if address:
                # If an address is found, update the empty 'latitude' and 'longitude' values for this row (College N
                dataframe.at[index, 'latitude'] = address.latitude
                dataframe.at[index, 'longitude'] = address.longitude
                continue
            except (GeocoderTimedOut, GeocoderUnavailable) as e:
                # Print an error message if the geocoding failed for this college name
                print(f"Geopy timeout or unavailable for {row['College Name']}: {e}")

```

Figure 2.5: Code Snippet for scraping latitude and longitude

which returns the address object. The method extracts the latitude and longitude from the address object if this address exists. As a result, another file named `df_lat_lon.parquet` is created, which is used as an input for scraping the median household income as shown in flowchart in Figure 2.1.

2.1.6 Scraping county

Scraping county information code is similar to scraping latitude and longitude. The major differences are shown in the county scrape snippet code in Figure 2.6.

```

dataframe['county_name'] = pd.NaT
dataframe['county_code'] = pd.NaT

for index, row in dataframe.iterrows():
    latitude = row['latitude']
    longitude = row['longitude']

    if pd.notna(latitude) and pd.notna(longitude):
        url = f"https://geo.fcc.gov/api/census/block/find?latitude={latitude}&longitude={longitude}&format=json"

        try:
            response = requests.get(url, timeout=10)
            response.raise_for_status()
            data = response.json()

            if 'County' in data:
                county_name = data['County'].get('name', pd.NaT)
                county_code = data['County'].get('FIPS', pd.NaT)

                dataframe.at[index, 'county_name'] = county_name
                dataframe.at[index, 'county_code'] = county_code

```

Figure 2.6: Code Snippet for scraping county name and county code

As the script outlines, the initial steps remain the same. The input data frame, generated by the script referred to as `latlonscript.py`, is named `df_lat_lon_county.parquet`. This parquet file is read from the local machine, and the following steps of the script are similar to `scrape_lat_lon.py`, as described

in Figure 2.5. The primary distinction lies in the URL provided by `https://geo.fcc.gov/api/`.

After successfully scraping the county name and county code, the script saves this data in a file named `df_lat_lon_county.parquet`. This file is then used in the next step as described in the flowchart in Figure 2.1 as input for scraping median household income as shown in Figure 2.7.

2.1.7 Scraping median household income

In the code snippet in Figure 2.7, the main logic that scrapes the median household income from `countyhealthrankings.org` is shown.

```
url = f'https://www.countyhealthrankings.org/explore-health-rankings/county-health-rankings-model/health-fa
browser.get(url)

# WebDriverWait method ensures that the page is loaded by waiting 15 seconds
income_element = WebDriverWait(browser, 15).until(
    # Search inside a table row for a span element called Median household income
    # Get the second table data (td) in that row that was found. It gets the value in the second column.
    EC.presence_of_element_located(By.XPATH, "//tr[td/span[text()='Median household income']/td[2]"))
)
# store the value that was retrieved in median_income variable, this values is also returned.
median_income = income_element.text
```

Figure 2.7: Code Snippet for scraping Median Household Income

In this script, the input data frame is the output of the previous script as described in Figure 2.6, which generates `"df_lat_lon_county.parquet"`. This file associates each latitude-longitude pair with a specific county. For each latitude-longitude pair, the script scrapes the median household income. The resulting data frame contains columns for College Name, IPEDS (University ID), and Median Household Income for each row that was scraped from `"df_lat_lon_county.parquet"`.

2.1.8 Merging and preprocessing

The preprocessing and merging part includes:

- Transform the graduation rates from the file `grad_rates.parquet` for each ethnic group by converting the number of students who graduated into decimal graduation rates. This is done by dividing the number of graduates by the total student enrollment.
- Create a single column titled “Tuition Fee,” which consolidates each institution’s tuition fee data from the following columns: ‘Out-of-state tuition and fees,’ ‘Tuition and fees,’ ‘In-state tuition and fees,’ and ‘In-district tuition and fees.’ For each institution, select the tuition fee in the following order: first select ‘Out-of-state tuition and fees,’ if not available, then ‘Tuition and fees’; if neither is available, the institution is removed from this data frame.
- Replace the dollar sign in the Median Household Income values with a space and convert these values to integers.

```

df = pd.DataFrame(df_combined)

ethnicities = ['us_nonresident', 'hispanic/latino', 'american_indian_or_alaska_native', 'asian',
              'black_or_african_american', 'native_hawaiian_or_other_pacific_islander',
              'white', 'two_or_more_races', 'race_and_ethnicity_unknown']

for ethnicity in ethnicities:
    men_column = f"{ethnicity}_men"
    women_column = f"{ethnicity}_woman"
    df[ethnicity + '_students'] = df[men_column] + df[women_column]
    df.drop(columns=[men_column, women_column], inplace=True)

df.drop(columns=['total_men_women'], inplace=True)
df_combined = df.copy()

```

Figure 2.8: Code Snippet for merging genders into total students

Merge the preprocessed graduation rates, tuition fees, and median household of each year into one final data frame called `combined_df`. The merging and preprocessing script takes the tuition fee, graduation rates, and median household income parquet files for three different years as input. Its output is a data frame called `df_combined`, as specified in the flowchart in Figure 2.1.

The resulting dataset consists of 3,975 data points and 35 features. Each row represents a 4-year institution, including information on graduation rates, tuition fees, and median household income for a specific year, which can be either 2020, 2021, or 2022.

This final preprocessed and merged dataset consists of 3 different subsets of features:

- Graduation rates of 8 different ethnic groups.
- Demographics include for each ethnic group, the number of men registered in that year, the number of women registered in that year, the total number of men and women in each university, and the total number of students (both men and women).
- Tuition Fees.
- Median Household income.

Before performing the cluster analysis, the demographic features of this dataset were processed further by removing the column representing the total number of men and women. This column was redundant due to its correlation with the individual counts (columns) of total men and women. Also, data for each ethnic group was merged for both genders to include the total number of students registered within each group.

The columns for total men and total women were retained to preserve information regarding the gender distribution of the institutions. The code snippet that processes the demographic features as described is shown in Figure 2.8.

2.1.9 Data analysis

After preprocessing the demographic features, the `df_combined` data frame is updated with the feature changes. This data frame is used to perform cluster

analysis.

After preprocessing and transforming some of the features of the raw data. The final dataset has 3,975 data points with 25 features. The following plots show the distribution of the number of students in each ethnic group Figure 2.10, the distribution of total men and total women (Figure 2.11), median household income (Figure 2.12), and tuition fees (Figure 2.9). These plots use box plots to visualize the distributions for the `df_combined` data frame, which contains merged information from all three years, and it is the final dataset used for Cluster Analysis.

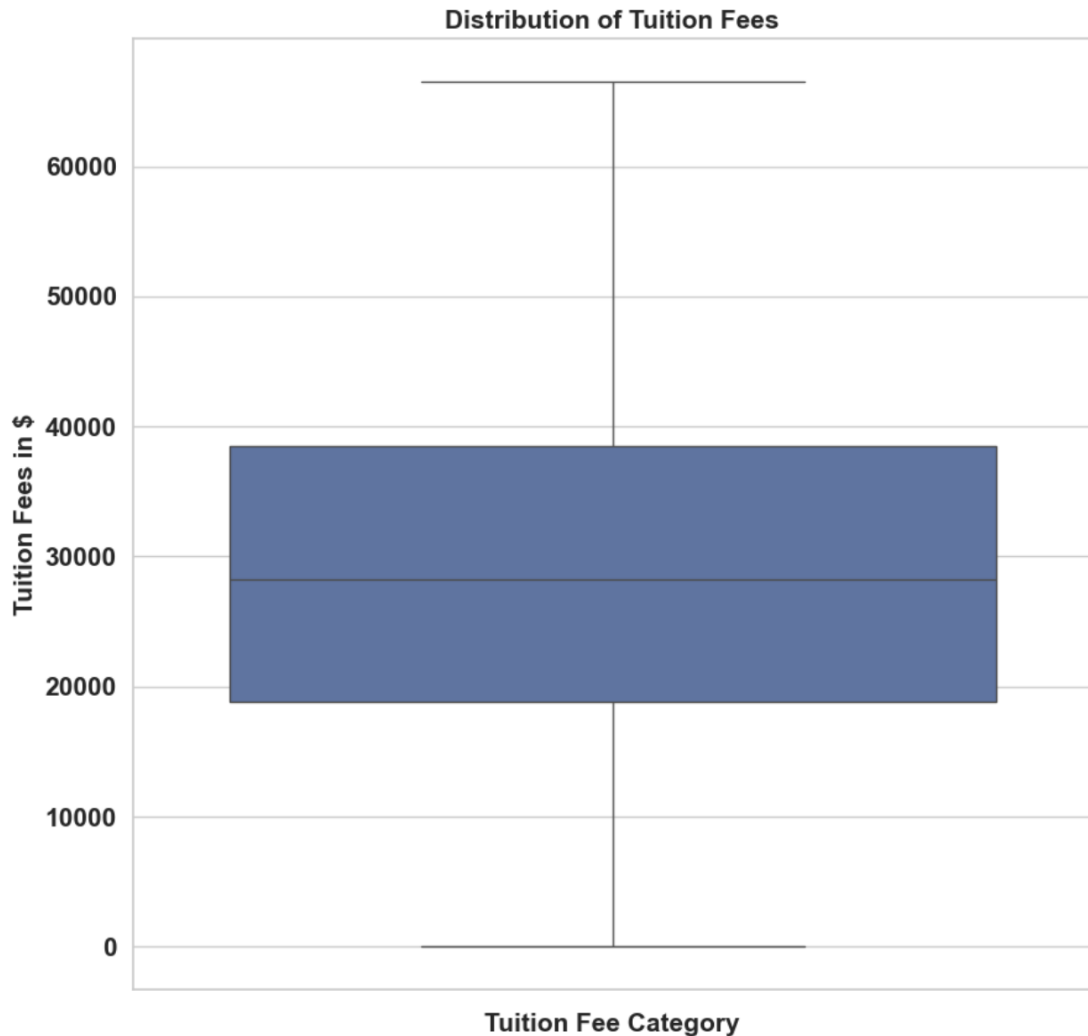


Figure 2.9: Tuition Fees distribution of all 4-year institutions

From Figure 2.10, we can observe that the number of American Indian or Alaska Native students in all institutions in the dataset across three years is really low. The same issue is also for Native Hawaiian or other Pacific Islander students, as shown in Figure 2.10. After this observation, these two features were removed since they do not provide much information about this ethnic group, and they are not going to be used as part of cluster analysis. This plot also clearly shows that most of the colleges in this dataset are populated by white

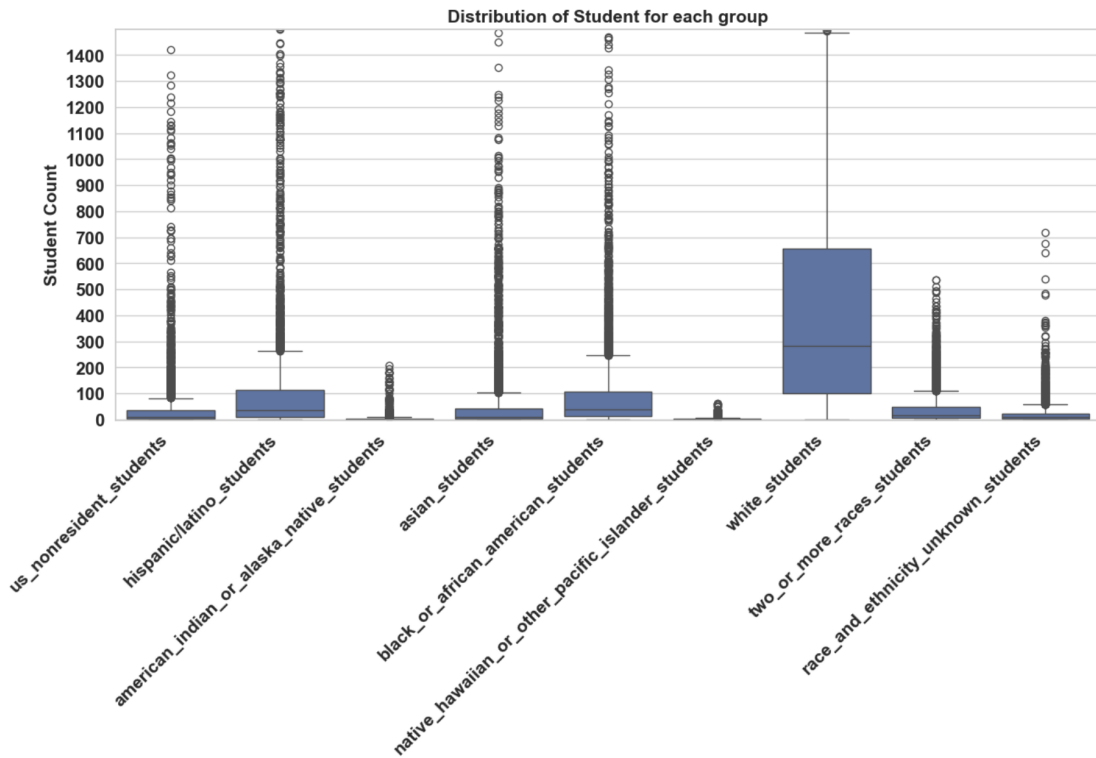


Figure 2.10: Students distribution of all 4-year institutions

students. However, some institutions have a large number of each ethnic group, as shown in box plots as outliers.

Figure 2.13 shows the graduation rates for each ethnic group over three consecutive years (2020, 2021, and 2022). These figures show that there are not many changes in graduation rates across the three years for different ethnicities.

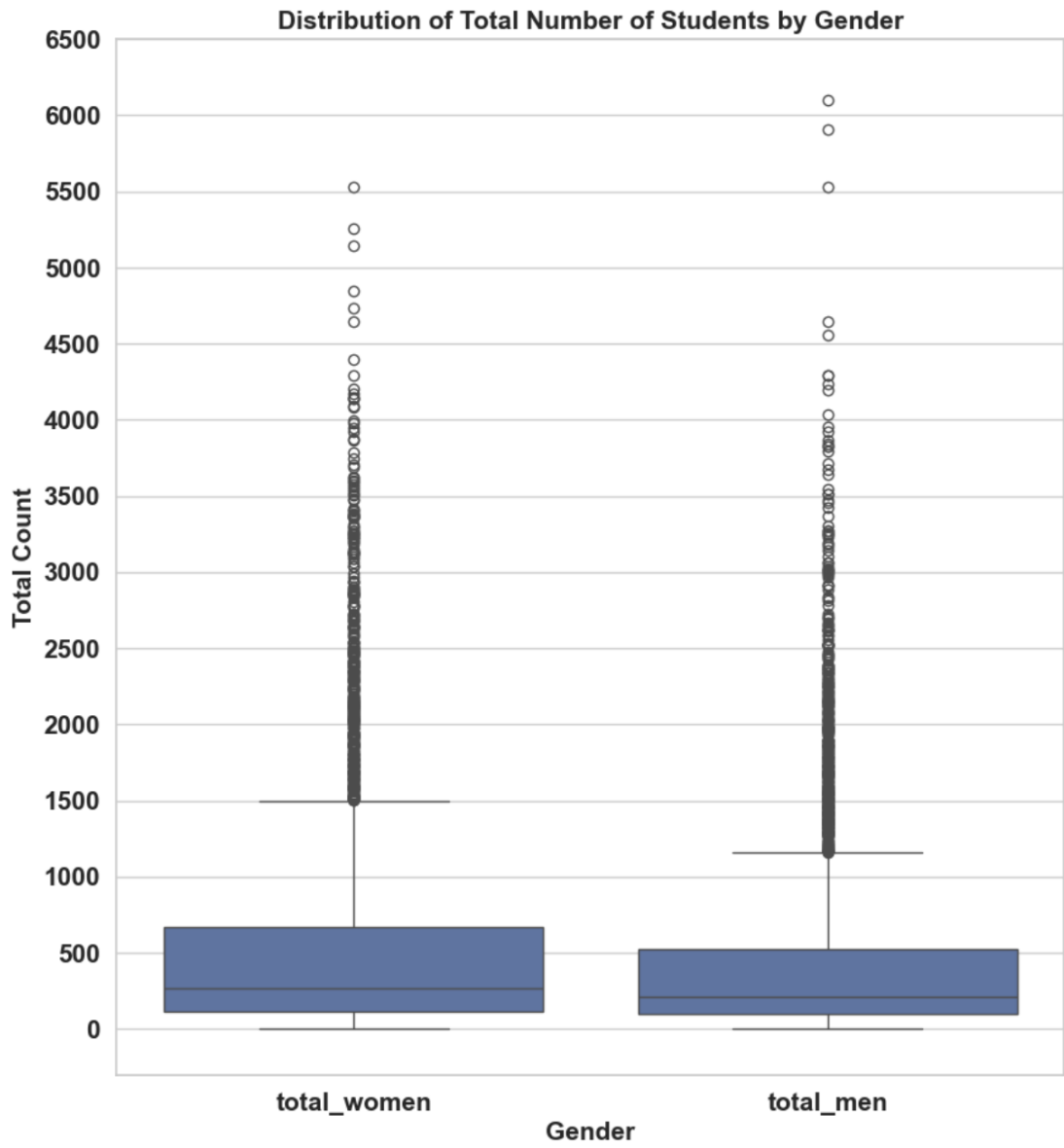


Figure 2.11: Gender distribution of 4-year institutions

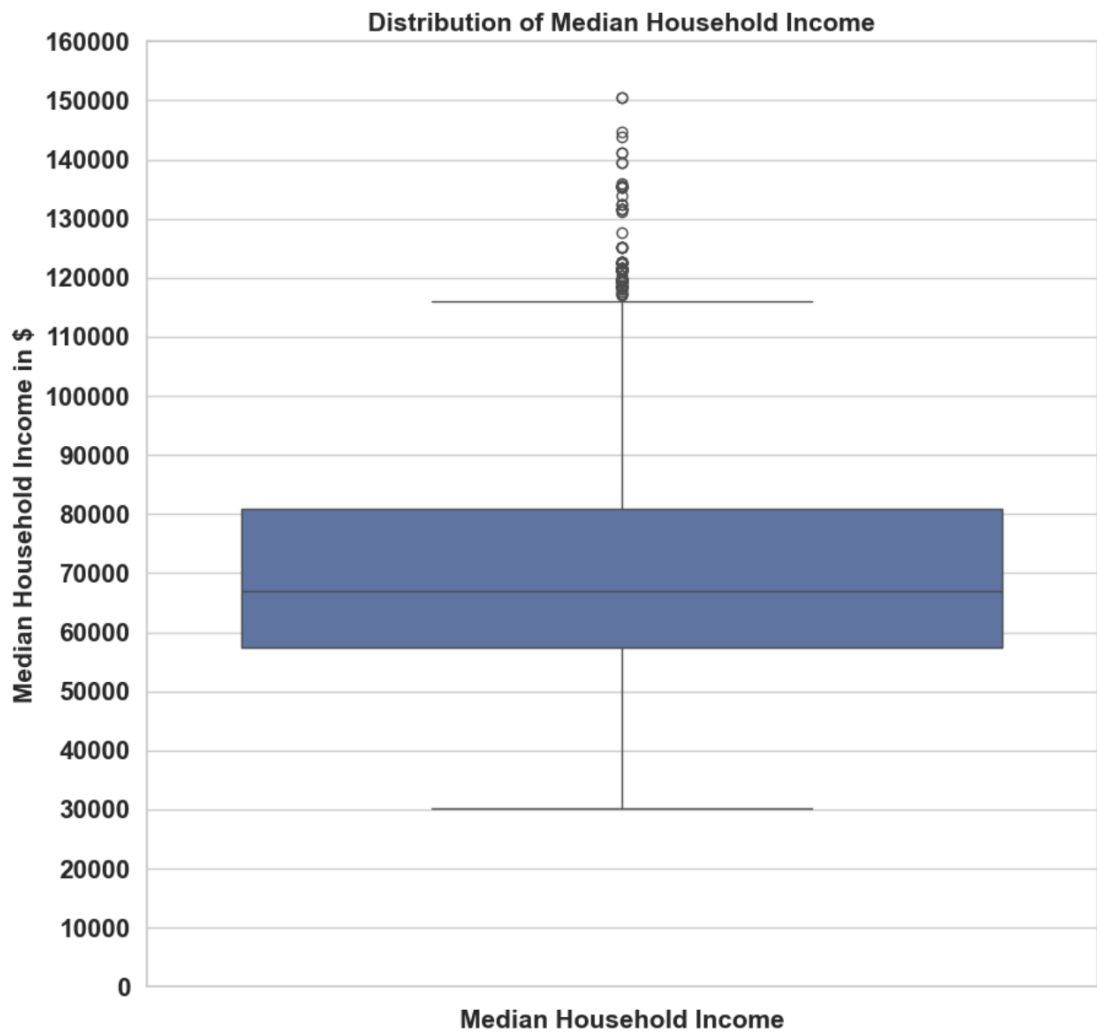


Figure 2.12: Tuition Fees distribution of 4-year institutions across three years

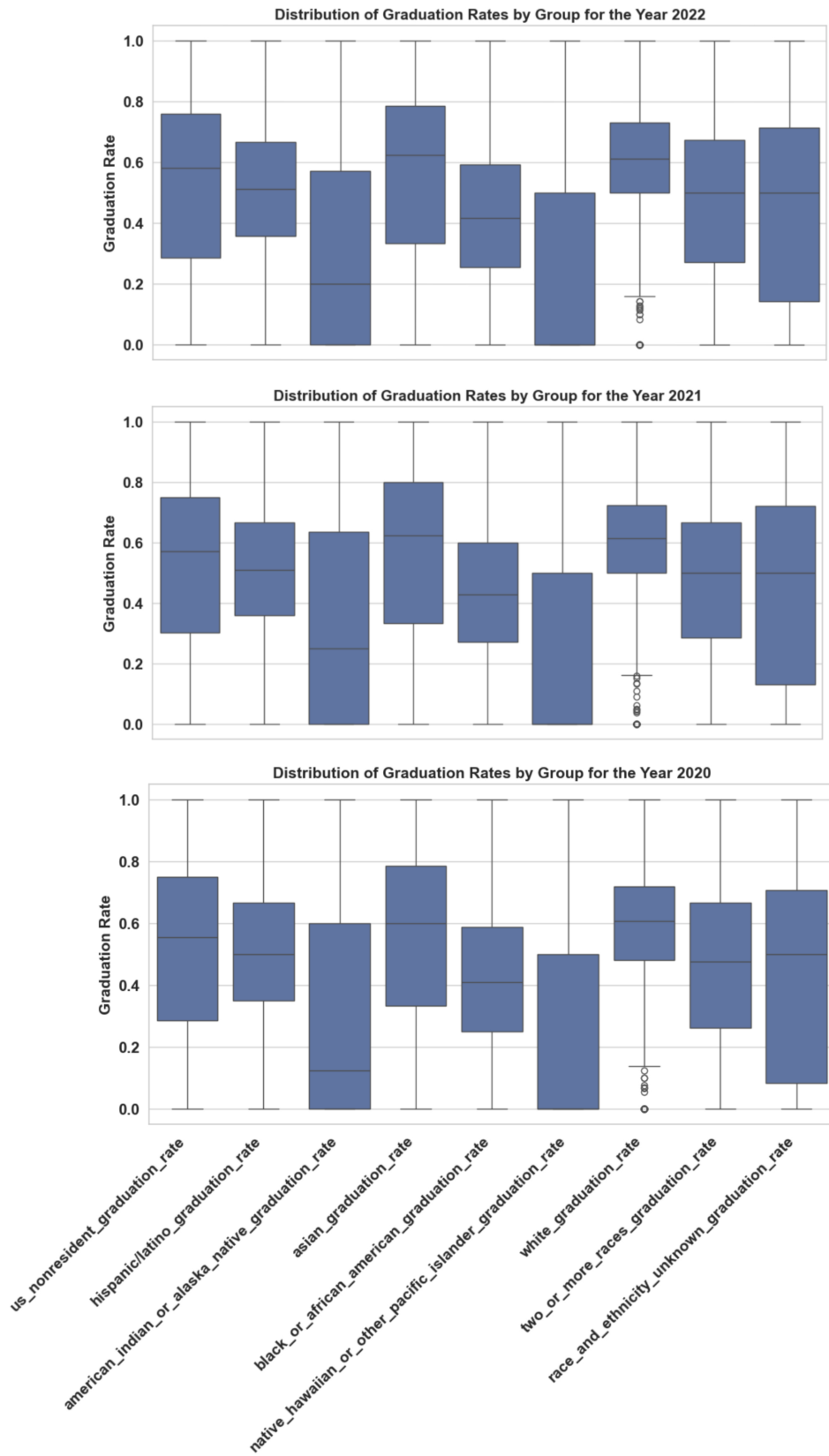


Figure 2.13: Graduation Rates of each ethnic group in 4-year institutions across all years

The figures below show the universities with the highest number of students in each respective ethnicity and show the overall campus demographics of other ethnic groups. These figures are available online in the appendix of my thesis on GitHub:

- Top 5 White Students
- Top 5 American Indian Students
- Top 5 Hawaiian Students
- Top 5 Hispanic/Latino Students
- Top 5 Black/African American Students
- Top 5 Two or More Races Students
- Top 5 Asian Students
- Top 5 Nonresidents

2.1.10 Cluster analysis - experiment setup

To train the SOM using the `df_combined` dataset. The goal was to minimize the *quantization_error* and *topographic_error* and then use the trained SOM to cluster the neurons based on their weight vectors.

A random selection of data points from the dataset was used for initialization of the weights, and the initial weight vectors of neurons were assigned from the feature vector of the actual data points.

For finding the best parameters of the model which minimize the *quantization_error* and *topographic_error*, different values of sigma and learning rate were used. The script below represents the values that were used for 500 trials, and those with the lowest topographical error and quantization error were selected as best. This range of values was selected after experimentally testing different configurations, which resulted in the best-performing SOMs.

```
space = {  
  'sigma': hp.uniform('sigma', 0.4, 10),  
  'learning_rate': hp.uniform('learning_rate', 0.0001, 1)  
}
```

No. of neurons used was 1225. Number of iterations is 500,000. The resulting SOM's, clustered using K-means is shown in Figure 2.14. The final configuration of the SOM model's error values were: Quantization Error: 0.35 and Topographic Error: 0.05.

2.1.11 Clustering with K-means

For clustering the self organizing map, silhouette analysis was performed to find the optimal clustering number. The clustering methods were applied to the weights of the neurons. In Figure 2.15, a visualization of silhouette score across different number K is presented.

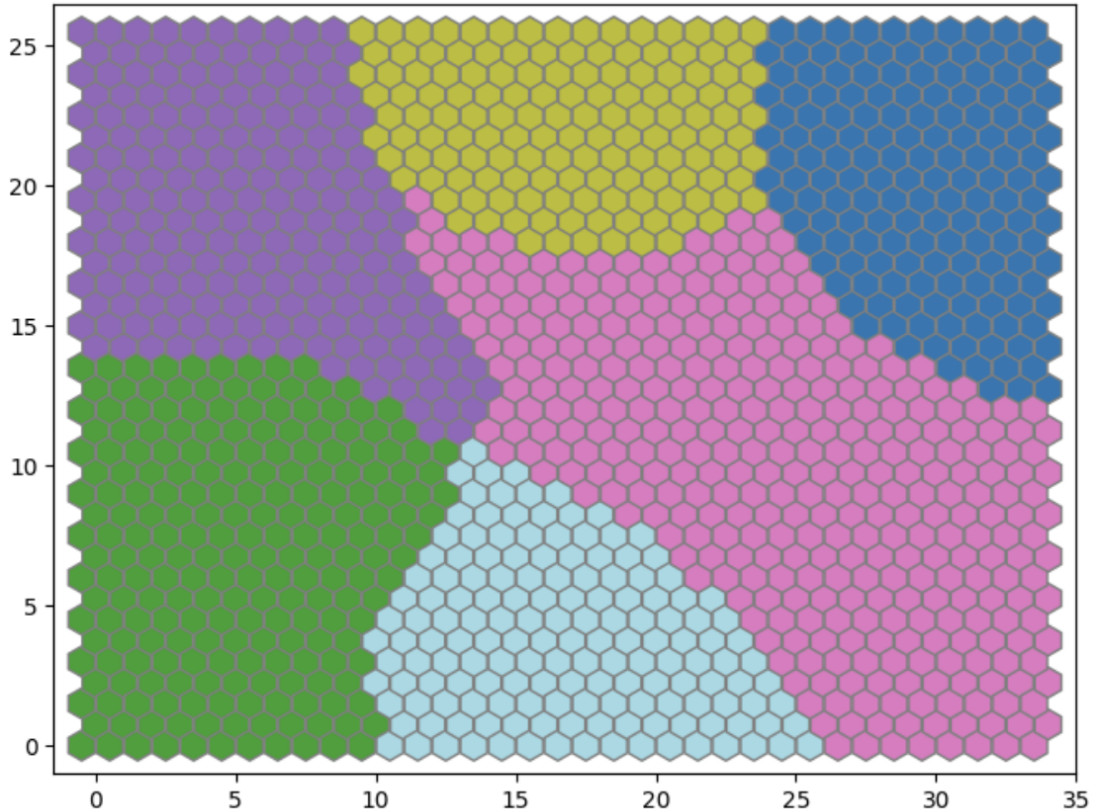


Figure 2.14: K-means clustering on SOM

Silhouette Score

- A score of 1 indicates that the clusters are clearly defined and separate from each other.
- A score of 0 indicates overlapping clusters.
- A score close to -1 indicates that clusters are not defined and the data is not separable.

After testing with different clustering techniques, including K-medoids, and using different configurations of Self-Organizing Maps (SOMs), the highest silhouette score for a configuration of six clusters, was around 0.35. The silhouette analysis for different values of k is shown in Figure 2.15. While the silhouette score is slightly higher for $k = 2$, $k = 6$ was chosen for several reasons. First, having only two clusters does not provide enough details for this analysis, potentially missing essential patterns in the data or oversimplifying the clustering. Choosing $k = 6$ allows for a more detailed and nuanced clustering, which can reveal more specific patterns and insights. Furthermore, the silhouette score for $k = 6$ is still competitive and it is not significantly lower than the score for $k = 2$. This score shows some structure in the data, and points are closer to their cluster center. However, the clusters are not well-defined and not entirely distinctive.

This score might suggest that the data may not be naturally grouped into well-defined clusters based on this current set of features. In future work, including

more features regarding these institutions might improve the clustering. The results of SOM clustering are shown in Figure 2.14.

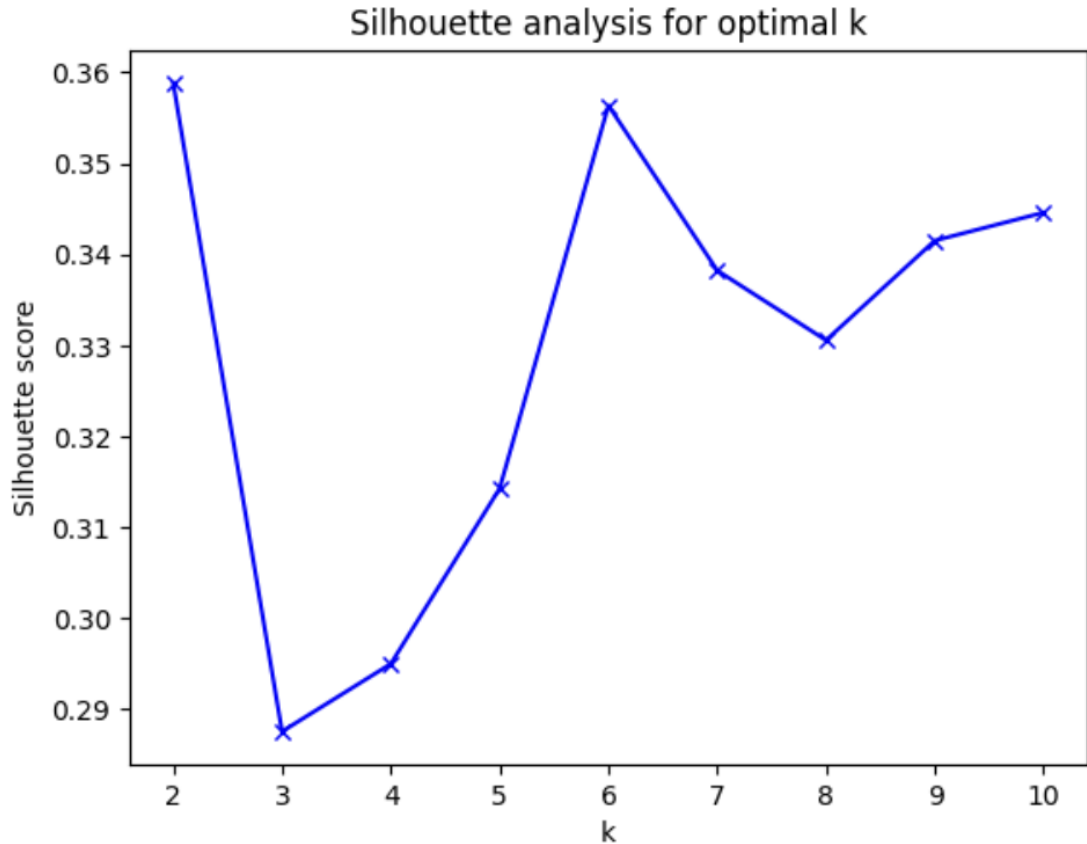


Figure 2.15: Silhouette Score Analysis

2.1.12 Cluster visualization and interpretation

The visualizations of the clusters in this section show the distribution of each feature across each cluster. For each feature, box plots will describe how data points are distributed across the clusters, highlighting the differences between these groups related to each feature. This visualization aims to understand the clusters further and check for any relationships between different ethnic groups within institutions.

Cluster One

- Second highest numbers of women students compared to other clusters, with student counts ranging from 0 to approximately 3,500. The median number of women students in this cluster is about 400. The cluster is shown in Figure 2.16.
- Second highest number of male students, with student counts ranging from 0 to approximately 3000. The median number of male students in this cluster is about 320. The cluster is shown in Figure 2.16.



Figure 2.16: Population of students, tuition fees and median household income across K means clusters

- Locations in regions where the median household income is relatively high compared to other clusters, ranging from approximately \$42,000 to \$150,000, with a median of about \$75,000. The cluster is shown in Figure 2.16.
- Highest tuition fees compared to other clusters, starting from \$0 to approximately \$66,000, with a median of \$53,000. The cluster is shown in Figure 2.16.
- Graduation rates for various ethnic groups are as follows:
 - The median graduation rate for the 'White' ethnic group within this cluster is 83%. The median number of students of this ethnic group within this cluster is about 470. The cluster is shown in Figure 2.18.
 - The median graduation rate for the 'Black or African American' ethnic group within this cluster is 88%. The median number of students of this ethnic group within this cluster is around 40. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Asian' ethnic group within this cluster is 86%. The median number of students of this ethnic group within this cluster is about 50. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Hispanic or Latino' ethnic group within this cluster is 80%. The median number of students of this ethnic group within this cluster is 70. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Two or more races' ethnic group within this cluster is 82%. The median number of students of this ethnic group within this cluster is 31. The cluster is shown in Figure 2.18.
 - The median graduation rate for the 'US-Nonresident' ethnic group within this cluster is 82%. The median number of students of this ethnic group within this cluster is 45. The cluster is shown in Figure 2.17.

Cluster Two

- Lowest numbers of women students compared to other clusters, with student counts ranging from a minimum of 0 to a maximum of about 1,100. The median number of women students in this cluster is about 70. The cluster is shown in Figure 2.16.
- Lowest numbers of male students compared to other clusters, with student counts ranging from a minimum of 0 to a maximum of approximately 800. The median number of male students in this cluster is about 70. The cluster is shown in Figure 2.16.
- These institutions are located in regions where the median household income is relatively lower compared to other clusters, with a range from \$30,200 to



Figure 2.17: Graduation Rates of ethnic groups across K means clusters

about \$130,000. The median household income is approximately \$64,000. The cluster is shown in Figure 2.16.

- Low tuition fees start from a minimum of approximately \$2,000 to a maximum of about \$66,000. The median tuition fee stands at approximately \$17,000. The cluster is shown in Figure 2.16.
- Graduation rates for various ethnic groups are as follows:
 - The median graduation rate for the 'White' ethnic group within this cluster is 41%. The median number of students of this ethnic group within this cluster is around 40 students. The cluster is shown in Figure 2.18.
 - The median graduation rate for the 'Black or African American' ethnic group within this cluster is 2%. The median number of students of this ethnic group within this cluster is 13. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Asian' ethnic group within this cluster is 0%. The median number of students of this ethnic group within this cluster is 0. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Hispanic or Latino' ethnic group within this cluster is 25%. The median number of students of this ethnic group within this cluster is 5. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'US-Nonresidents' ethnic group within this cluster is 0%. The median number of students of this ethnic group within this cluster is 0. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Two or more races' ethnic group within this cluster is 0%. The median number of students of this ethnic group within this cluster is 2. The cluster is shown in Figure 2.18.

Cluster Three

- The median number of women students in this cluster is about 240, with student counts ranging from 0 to approximately 1,500. The cluster is shown in Figure 2.16.
- The median number of male students in this cluster is 210, with student counts ranging from 0 to about 1230. The cluster is shown in Figure 2.16.
- Locations in regions where the median household income ranges from approximately \$34.0k to \$120.0k, with a median of about \$60.0k. The cluster is shown in Figure 2.16.
- Mid-range tuition fees compared to other clusters, starting from approximately \$4,000 to \$51,000k, with a median of about \$20,000k. The cluster is shown in Figure 2.16.

- Graduation rates for various ethnic groups are as follows:
 - The median graduation rate for the 'White' ethnic group within this cluster is 50%. The median number of students of this ethnic group within this cluster is about 200. The cluster is shown in Figure 2.18.
 - The median graduation rate for the 'Black or African American' ethnic group within this cluster is 30%. The median number of students of this ethnic group within this cluster is about 50. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Asian' ethnic group within this cluster is 40%. The median number of students of this ethnic group within this cluster is 6. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Hispanic or Latino' ethnic group within this cluster is 40%. The median number of students of this ethnic group within this cluster is about 30. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Two or more races' ethnic group within this cluster is 33%. The median number of students of this ethnic group within this cluster is 15. The cluster is shown in Figure 2.18
 - The median graduation rate for the 'US-Nonresident' ethnic group within this cluster is 60%. The median number of students of this ethnic group within this cluster is 10. The cluster is shown in Figure 2.17.

Cluster Four

- The median number of women students in this cluster is about 260, with student counts ranging from 0 to approximately 1,400. The cluster is shown in Figure 2.16.
- The median number of male students in this cluster is 200, with student counts ranging from 0 to about 1100. The cluster is shown in Figure 2.16.
- Locations in regions where the median household income ranges from approximately \$40.0k to \$150.0k, with a median of about \$70.0k. The cluster is shown in Figure 2.16.
- Second highest tuition fees compared to other clusters, starting from approximately \$5,000 to \$59,000k, with a median of about \$34,000k. The cluster is shown in Figure 2.16.
- Graduation rates for various ethnic groups are as follows:
 - The median graduation rate for the 'White' ethnic group within this cluster is 65%. The median number of students of this ethnic group within this cluster is about 300. The cluster is shown in Figure 2.18.

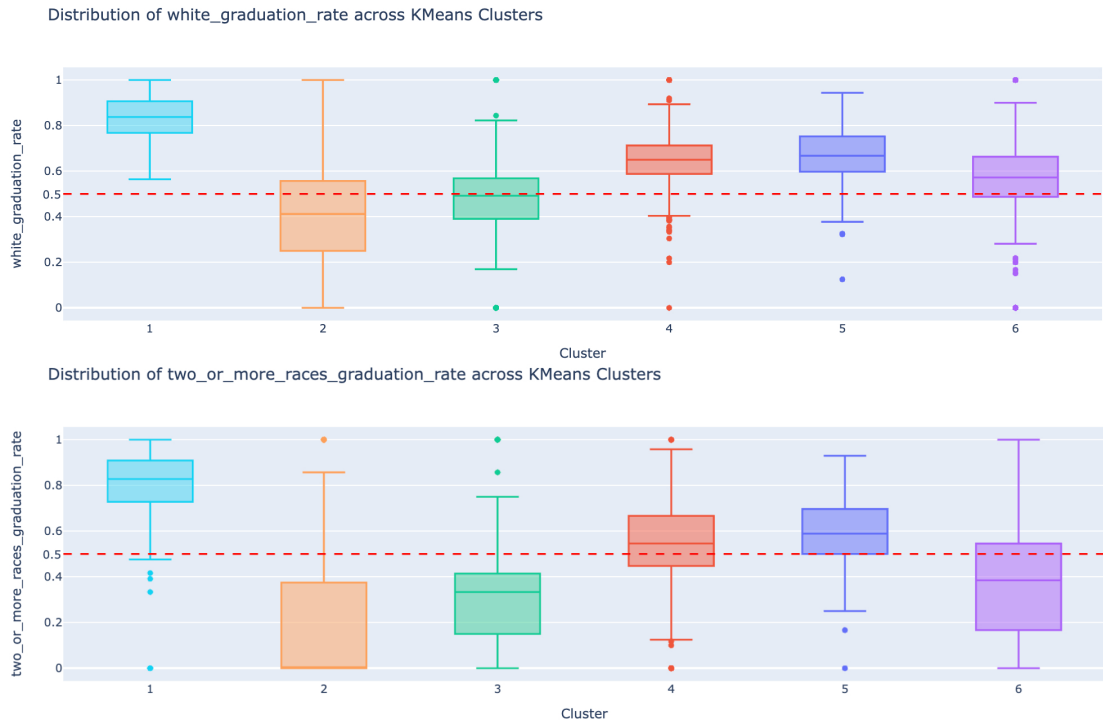


Figure 2.18: Graduation Rates of ethnic groups across K means clusters

- The median graduation rate for the 'Black or African American' ethnic group within this cluster is 45%. The median number of students of this ethnic group within this cluster is about 33. The cluster is shown in Figure 2.17.
- The median graduation rate for the 'Asian' ethnic group within this cluster is 66%. The median number of students of this ethnic group within this cluster is 10. The cluster is shown in Figure 2.17.
- The median graduation rate for the 'Hispanic or Latino' ethnic group within this cluster is 50%. The median number of students of this ethnic group within this cluster is about 40. The cluster is shown in Figure 2.17.
- The median graduation rate for the 'Two or more races' ethnic group within this cluster is 54%. The median number of students of this ethnic group within this cluster is 15. The cluster is shown in Figure 2.18
- The median graduation rate for the 'US-Nonresident' ethnic group within this cluster is 65%. The median number of students of this ethnic group within this cluster is 10. The cluster is shown in Figure 2.17.

Cluster Five

- Highest numbers of women students compared to other clusters, with student counts ranging from a minimum of around 250 to a maximum of 5,500.

The median number of women students in this cluster is around 1,800. The cluster is shown in Figure 2.16.

- Highest numbers of male students compared to other clusters, with student counts ranging from a minimum of about 200 to a maximum of approximately 6,000. The median number of male students in this cluster is about 1,400. The cluster is shown in Figure 2.16.
- These institutions are located in regions where the median household income is lower to mid-range compared to other clusters, with a range from \$39,700 to \$144,600. The median household income is about \$65,000. The cluster is shown in Figure 2.16.
- Mid-range tuition fees starting from a minimum of \$0 to a maximum of approximately \$55,000. The median tuition fee is about \$25,000. The cluster is shown in Figure 2.16.
- Graduation rates for various ethnic groups are as follows:
 - The median graduation rate for the 'White' ethnic group within this cluster is 66%. The median number of students of this ethnic group within this cluster is about 1900. The cluster is shown in Figure 2.18.
 - The median graduation rate for the 'Black or African American' ethnic group within this cluster is 56%. The median number of students of this ethnic group within this cluster is 230. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Asian' ethnic group within this cluster is 71%. The median number of students of this ethnic group within this cluster is about 140. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Hispanic or Latino' ethnic group within this cluster is 59%. The median number of students of this ethnic group within this cluster is about 140. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'US-Nonresidents' ethnic group within this cluster is 66%. The median number of students of this ethnic group within this cluster is about 70. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Two or more races' ethnic group within this cluster is 58%. The median number of students of this ethnic group within this cluster is approximately 130. The cluster is shown in Figure 2.18.

Cluster Six

- The median number of women students in this cluster is approximately 180, with student counts ranging from 0 to approximately 1,500. The cluster is shown in Figure 2.16.

- The median number of male students in this cluster is about 150, with student counts ranging from 0 to approximately 1,100. The cluster is shown in Figure 2.16.
- Locations in regions where the median household income are low compared to other clusters, ranging from approximately \$34.0k to \$135.0k, with a median of \$56.0k. The cluster is shown in Figure 2.16.
- Mid-range tuition fees compared to other clusters, starting from approximately \$800 to \$58,000k, with a median of about \$26,000k. The cluster is shown in Figure 2.16.
- Graduation rates for various ethnic groups are as follows:
 - The median graduation rate for the 'White' ethnic group within this cluster is 57%. The median number of students of this ethnic group within this cluster is about 180. The cluster is shown in Figure 2.18.
 - The median graduation rate for the 'Black or African American' ethnic group within this cluster is 33%. The median number of students of this ethnic group within this cluster is about 30. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Asian' ethnic group within this cluster is 71%. The median number of students of this ethnic group within this cluster is 5. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Hispanic or Latino' ethnic group within this cluster is 19%. The median number of students of this ethnic group within this cluster is approximately 20. The cluster is shown in Figure 2.17.
 - The median graduation rate for the 'Two or more races' ethnic group within this cluster is 38%. The median number of students of this ethnic group within this cluster is approximately 10. The cluster is shown in Figure 2.18.
 - The median graduation rate for the 'US-Nonresidents' ethnic group within this cluster is 0%. The median number of students of this ethnic group within this cluster is 1. The cluster is shown in Figure 2.17.

Figures 2.19, 2.20, 2.21, 2.22, 2.23, and 2.24 show the institutions with the highest graduation rates for each ethnic group. The selected institutions had a minimum of 20 students per ethnic group. Figures 2.25 and 2.26 show the randomly selected institutions from clusters 1, 2, 3, 4, 5 and 6.

College Name	US Non-Residents Graduation Rates	No. Of Students
Middlebury College	0.981	54
Williams College	0.979	47
Yale University	0.972	145
Princeton University	0.971	139
University of Chicago	0.964	197
Tufts University	0.963	136
Harvard University	0.958	190
University of Pennsylvania	0.951	266
Stanford University	0.948	172
Davidson College	0.941	34

Figure 2.19: Top 10 universities with the highest graduation rates for U.S. non-resident students.

College Name	Two or More Races Graduation Rates	No. Of Students
Middlebury College	1.000	28
University of Pennsylvania	0.989	89
University of Chicago	0.986	71
Harvard University	0.984	126
Princeton University	0.980	49
Yale University	0.977	88
Brown University	0.976	126
University of Pennsylvania	0.975	122
Williams College	0.975	40
Emory University	0.973	73

Figure 2.20: Top 10 universities with the highest graduation rates for students of two or more races.

College Name	Black or African Graduation Rates	No. Of Students
Harvard University	0.992	130
Brown University	0.980	99
Santa Clara University	0.979	48
Princeton University	0.972	107
University of Pennsylvania	0.964	165
Yale University	0.954	87
Davidson College	0.946	37
Williams College	0.943	35
Swarthmore College	0.939	33
Tufts University	0.938	64

Figure 2.21: Top 10 universities with the highest graduation rates for Black or African American students.

College Name	Hispanic / Latino Graduation Rates	No. Of Students
Cooper Union	1.000	21
Princeton University	0.986	141
Harvard University	0.984	188
Davidson College	0.973	37
Northwestern University	0.970	271
Grinnell College	0.969	32
University of Notre Dame	0.968	221
Williams College	0.958	71
University of Virginia	0.954	237
Brown University	0.951	182

Figure 2.22: Top 10 universities with the highest graduation rates for Hispanic students.

College Name	Asian Graduation Rates	No. Of Students
Yale University	0.988	251
Princeton University	0.986	280
Northwestern University	0.985	342
Harvard University	0.981	361
University of Pennsylvania	0.977	476
Middlebury College	0.977	43
University of Virginia	0.975	520
Stanford University	0.971	375
Williams College	0.971	68
Massachusetts Institute of Technology	0.966	264

Figure 2.23: Top 10 universities with the highest graduation rates for Asian students.

College Name	White Graduation Rates	No. Of Students
Princeton University	0.986	587
Harvard University	0.975	644
Massachusetts Institute of Technology	0.974	388
University of Notre Dame	0.971	1384
Northwestern University	0.971	927
University of Pennsylvania	0.971	1020
Yale University	0.967	614
Brown University	0.965	695
University of Chicago	0.965	649
Dartmouth College	0.962	559

Figure 2.24: Top 10 universities with the highest graduation rates for White students.

Random colleges from cluster 1

College Name	Median Household Income	Tuition Fee
Franklin & Marshall College	75600	63406
State University of New York at Binghamton	60500	28203
Southern Methodist University	82500	61980
Pacific University	90200	50070
Washington & Jefferson College	70800	50192
University of Rochester	66000	59378
Tufts University	118500	65222
Connecticut College	78600	63005
University of North Carolina at Chapel Hill	79800	36891
Mount Holyoke College	73900	54618

Random colleges from cluster 2

College Name	Median Household Income	Tuition Fee
United Talmudical Seminary	75600	15300
Texas Wesleyan University	71400	34412
Morehouse College	82800	29468
San Diego Christian College	87100	33312
Charleston Southern University	80000	29990
Oak Hills Christian College	82800	17334
The Baptist College of Florida	79500	12450
Turtle Mountain Community College	47400	2338
Ner Israel Rabbinical College	80000	12400
Mississippi Valley State University	35500	7414

Random colleges from cluster 3

College Name	Median Household Income	Tuition Fee
Saint John Vianney College Seminary	58900	23100
Arkansas Tech University	47300	13236
Louisiana State University in Shreveport	43200	20314
Emporia State University	51400	14918
Southeastern Louisiana University	54900	20851
Southern Illinois University Carbondale	44800	15416
Pikeville College	41800	23150
Bloomfield College	65500	30680
Martin Luther College	65500	16420
University of New England	68000	38750

Figure 2.25: Randomly selected institutions from cluster 1,2 and 3

Random colleges from cluster 4

College Name	Median Household Income	Tuition Fee
Minnesota State University-Mankato	68600	18200
Daemen College	63000	31250
Martin Luther College	66200	16910
Morningside College	66800	36610
North Central College	99500	42206
Widener University	83900	51639
Massachusetts Maritime Academy	76300	26807
Otterbein College	66100	33674
Ohio Dominican University	66100	33380
Central Connecticut State University	75100	24994

Random colleges from cluster 5

College Name	Median Household Income	Tuition Fee
University of Kansas	58700	28035
Temple University	52900	30602
Louisiana Tech University	40400	16806
Baylor University	65800	49246
Boise State University	87700	25701
Campus	66300	31785
University of Wisconsin-Madison	74800	38630
Auburn University	54200	31986
State University of New York at Buffalo	59600	28196
Eastern Michigan University	77400	15700

Random colleges from cluster 6

College Name	Median Household Income	Tuition Fee
Saint Joseph Seminary College	68600	22810
University of the Cumberland	43600	9875
Voorhees College	38000	12630
Paul Smith's College	55000	31115
College of Mount Saint Vincent	70600	40980
Salem College	56200	31016
California College San Diego	87100	16600
Norfolk State University	57000	20790
University of Maine at Farmington	49300	20282
Emmaus Bible College	66000	19250

Figure 2.26: Randomly selected institutions from cluster 3,4 and 6

2.1.13 Changes in universities across the 2020, 2021 and 2022

College Name	MHI 2020	MHI 2021	MHI 2022	Changes (MHI) in %	TF 2020	TF 2021	TF 2022	Changes in (TF) in %
Catawba College	85200	89700	93800	10.1%	31436	32380	32868	4.6%
Goshen College	66800	62200	63800	-4.5%	35230	35940	36660	4.1%
Pillar College	65500	66200	74700	14.0%	22756	22756	23668	4.0%
Ave Maria University	81900	75800	81800	-0.1%	23188	24610	26354	13.7%
Cumberland University	77800	82500	88500	13.8%	25386	25412	26400	4.0%
Atlanta Christian College	71500	82800	89800	25.6%	21850	21850	22300	2.1%
Dallas Christian College	65800	63500	70900	7.8%	19086	19236	20580	7.8%
Tarleton State University	72100	71400	76300	5.8%	17448	17582	17640	1.1%
Montana State University - Northern	45900	46300	53600	16.8%	18665	18665	18665	0.0%
Valley City State University	60800	65600	69100	13.7%	12532	12969	13422	7.1%
Coker College	57100	56700	67700	18.6%	31524	31524	31764	0.8%

Figure 2.27: The institutions with the most changes and the variations in tuition fees and median household income over three years.

After training SOM, the goal was to identify universities that have shown significant changes in their SOM positions over three years, which could indicate significant changes in their features. This was achieved by writing a script that calculates the total Euclidean distance each university has moved on the SOM from 2020 to 2021 and then to 2022. A more considerable total distance indicates more changes.

After selecting ten universities that showed the most significant changes over three years based on the total Euclidean distance, these universities were grouped by year, and the median values were computed for each feature across three years. The median values highlighted that Median Household Income and Tuition Fees changed the most over three years in these ten universities.

The Median Household Income in these regions has changed significantly, especially from 2021 to 2022. There are changes in graduation rates and number of students in each ethnic group, but no significant changes were observed.

In Figure 2.27, the institutions with the most changes are shown, and how tuition fees and median household income have changed over three years is described.

Based on the Figure 2.27, we could see that Coker College has the most changes in median household income after observing the data for three years at this university. It was shown that this slight increase of 0.7% in tuition fees and the median household income of 18% increase affected graduation rates where the non resident student's (international students) graduation rate in 2020 was 69%. In 2021, it was 60%, and in 2022, it dropped to 30%, which is a regression in the performance of students. Black or African American graduation rates also dropped from 57% in 2020 to 36% in 2022. Two or more races went from 35% to 25%, and white graduation rates from 53% to 48%. In this case, the increase of 18% in median household income and increase of tuition fees of 0.7% did not improve the graduation rates; in contrast, they decreased.

Another institution (Montana State University-Northern) with a 16% increase in median household income from 2020 to 2022 and no increase in tuition fees did

not show any improvement in graduation rates or significant decrease. This university, mostly populated with white students, their graduation rates decreased slightly from 44% in 2020 to 37% in 2022. Another college (Atlanta Christian College), with a high increase in median household income (by 25%) and a tuition fee increase of 2.1%, showed slight improvements in graduation rates of Hispanic / Latino students (from 38% in 2020 to 47% in 2022, and for white people, it did not change, staying at 33% in three years.

Other graduation rates did not change significantly (i.e., the changes were lower than 10%). Based on these top 10 universities with the most changes, it cannot be concluded that changes in tuition fees and median household income improved the graduation rates. As we can see, in some cases, graduation rates decreased even though tuition fees and median household income increased.

2.1.14 Conclusions on the analysis of graduation rates in U.S. colleges

The study aimed to analyze the graduation rates across different ethnic groups in U.S. colleges over three years by including additional data such as tuition fees, median household income of the county (area) where the university is located. Data was scraped from the National Center for Educational Statistics website using python scripts, libraries like selenium for scraping dynamic content, and geocoders for geographical information.

The goal of training the Self-Organizing Map was to find if these universities can be clustered, perform cluster analysis by checking each feature across clusters that were created, and observe changes in the same institutions across three years. The cluster analysis highlighted that the institutions with higher tuition fees in regions with high median household incomes have higher graduation rates for all ethnic groups. This shows that demographic and economic factors influence graduation rates. After extracting the highest-performing universities from cluster 1, we could observe that Harvard, Princeton, and Yale University are consistently in the top 10 highest-performing universities for all ethnic groups that were analyzed, this is noticeable from the Figures 2.20, 2.21, 2.22, 2.23, and 2.24. Additionally, the population of men and women in this cluster is smaller compared to cluster 5, which is characterized by institutions with a high population and more affordable tuition fees. These institutions, where students tend to succeed and have higher graduation rates, have around 400 female and about 320 male students.

Cluster 5, on the other hand, has moderate to high-performing institutions among all ethnic groups. Based on plots, we can observe that 50% of the institutions have graduation rates above 50% in all ethnic groups except Black or African, and the median graduation rate is higher than 50% in all ethnic groups. The difference between Cluster 5 and Cluster 1 is that Cluster 1 has graduation rates with a median of around 80 - 90% graduation rates for all ethnic groups, which indicates the highest-performing institutions. Cluster 5 is characterized by universities with lower tuition fees compared to Cluster 1 and located in areas with lower median household incomes compared to Cluster 1.

Cluster two has institutions with the lowest graduation rates among all ethnic groups, These institutions are located in regions with lower median household

income and lower tuition fees.

Cluster four is another example of institutions with a lower number of students and high tuition fees having high graduation rates. This analysis shows that the institutions with higher tuition fees have higher graduation rates among all ethnic groups. Additionally, institutions in regions with higher median household income have higher graduation rates.

The resulting dataset consists of 3,975 data points and 35 features. Each row represents a 4-year institution, including information on graduation rates, tuition fees, and median household income for a specific year, which can be either 2020, 2021, or 2022.

This final preprocessed and merged dataset consists of three different subsets of features: graduation rates of 8 different ethnic groups; demographics such as for each ethnic group, the number of men registered in that year, the number of women registered in that year and also the total number of men and women in each university, the total number of students (both men and women); tuition fees; and median household income of the county where the university is located.

This dataset can support further analysis and experimentation by other researchers. It offers an opportunity to experiment with different data mining techniques and enhance the current analysis.

2.2 Social network analysis

The goal of social network analysis was to understand and research the role that education plays in parliament. The parliaments used for analysis are the UK Parliament, a fully established Parliament of England, formed in 1707 [32], and the youngest democracy in Europe, the Republic of Kosovo's Parliament, was established in 2008 after declaring independence from Serbia.

2.2.1 Background on united kingdom parliament

The monarchy of England and Britain has had 63 monarchs in power over more than 1200 years [33]. Today, United Kingdom operates as a constitutional monarchy and parliamentary democracy. The country is run by a monarch, King Charles the Third, and has a parliamentary system that handles the executive functions of the parliament [34].

The monarch has a largely ceremonial role, where the real political power mainly relies on the Parliament and the Prime Minister. The main duties of the monarch include the opening and dissolution of the Parliament, the appointment of the Prime Minister, and the formality of approving legislation.

The United Kingdom has a hereditary monarchy, meaning that the throne is passed from one family member to another upon death or abdication of the incumbent (this means that if the person next in line decides not to inherit the throne, the other one in line takes the throne).

The UK Parliament consists of the House of Commons (650 Members of the Parliament) and the House of Lords (787 members as of 24th April 2024) [35]. The House of Commons serves as the primary legislative body. Its members are elected in general elections, and the party or the coalition with the majority in

the House of Commons forms the government. Usually, the leader of the party or coalition becomes the Prime Minister.

The House of Commons holds most of the legislative power by creating / voting on laws, controlling finances, and reviewing government policy [36]. The House of Lords is considered the upper house, comprised of hereditary peers, bishops, and life peers. All members of the House of Lords are appointed, not elected. Life peers are appointed by the monarch, usually with the prime minister's advice, and they serve for their lifetime. The House of Lords acts as a revising chamber for legislation proposed by the House of Commons. However, The House of Lords cannot block legislation; they can only delay it [37].

The UK government has also given special powers to Northern Ireland, Wales, and Scotland. Each of these countries has its parliament or assembly, and certain decisions are made at the national level, especially in certain core areas such as education, health, and transportation [38].

2.2.2 UK parliament analysis

To get the data that is used for the analysis, the information on the members of parliament, such as name, alma mater, and party membership information was scraped. To get the information on the current active members of parliament, such as names and the parties to which they belong, this site was used: <https://www.parallelparliament.co.uk/MPs>. To get the universities they attended, each member's Wikipedia pages were used (e.g., link https://en.wikipedia.org/wiki/Bim_Afolami). After successfully scraping the data, the dataset was cleaned and it is shown in Figure 2.28.

	MP Name	MP Party	Alma Mater
0	Bim Afolami	Conservative	Oxford
1	Adam Afriyie	Conservative	Wye College
2	Nickie Aiken	Conservative	University of Exeter
3	Peter Aldous	Conservative	University of Reading
4	Stuart Andrew	Conservative	Ysgol David Hughes
...
543	Neale Harvey	Alba Party	City University, London
544	Kenny MacAskill	Alba Party	University of Edinburgh
545	Claire Hanna	Social Democratic & Labour Party	Open University, Queen's University Belfast
546	Caroline Lucas	Green Party	University of Exeter, University of Kansas
547	Stephen Farry	Alliance	Queen's University Belfast

548 rows × 3 columns

Figure 2.28: MP Names and Associated Universities

2.2.3 Graph structure

Each member of parliament could have attended multiple universities. Each node represents a member's name, and nodes are connected if they share at least one university.

The Networkx library was used to create the graph, and compute centrality measures. Gravis library was used to visualize the communities. In Figure 2.29 the centrality measure results are shown. Using Networkx library, top 10 members with highest degree centrality, betweenness closeness centrality, and eigenvector centrality were calculated. From these 10 members, 7 of them were consistent in all three measures (Degree Centrality, Closeness Centrality, and Eigenvector Centrality).

From the centrality measure, we could observe that:

1. John Glen, Matt Hancock, and Tanmanjeet Singh Dhesi have the highest values in degree centrality, which means that they have attended universities shared by many other members of the parliament, indicating an influential educational background in the context of the parliament.
2. Closeness Centrality - The average distance from a given starting node to all other nodes in the network. John Glen, Matt Hancock have the high measures of closeness centrality. These are central people who can reach other nodes "in a short distance". They are close to all other people in the network. This could signify a central role in the network, where they can influence and reach more people.
3. Eigenvector Centrality - John Glen, Matt Hancock, and Tanmanjeet Singh Dhesi have the higher centrality measures, meaning not only are they connected with many other people in the graph, but they also have strong ties with other important members of parliament. The people and connections they have good connections since they are highly connected with other members of parliament.

Conclusions of centralities in the UK parliament

The centrality measures show that some of the most important members are Matt Hancock, John Glen, and Tanmanjeet Singh Dhesi.

From the network graph in Figure 2.31 and from the centrality measures in Figure 2.29, we can observe that John Glen is an important member who connects members of two most important communities in the network (Oxford and Cambridge). He also went to King's College London university, making him an important node in the network.

Name	Degree Centrality	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality
John Glen	0.30	0.036	0.43	0.11
Matt Hancock	0.30	0.036	0.43	0.11
Tanmanjeet Singh Dhesi	0.29	0.029	0.42	0.11
Dominic Raab	0.27	0.017	0.42	0.11
Anneliese Dodds	0.23	0.021	0.40	0.10
Ed Davey	0.23	0.066	0.43	0.10
Keir Starmer	0.20	0.065	0.41	0.10
Average (Members)	0.26	0.039	0.42	0.111
Average (All Members)	0.06	0.0029	0.27	0.021

Figure 2.29: Centralities in the UK Parliament

Tanmanjeet Singh Dhesi and Matt Hancock are also a central member of the network, connecting the two biggest communities in the network, members that went to Oxford with members that went to Cambridge, as shown in Figure 2.31.

Additionally, Keir Starmer and Ed Davey show high betweenness centrality, meaning they are important in connecting different communities in the network. Keir Starmer serves as a bridge between members that went to Oxford and University of Leeds, as shown in Figure 2.31.

Roles of some of the most important members based on the centrality measures:

- John Glen was named Paymaster General and Minister for the Cabinet Office on November 13, 2023. He was formerly Chief Secretary of the Treasury from October 25, 2022 to November 13, 2023. More information can be found at <https://www.gov.uk/government/people/alex-burghart#announcements>. John Glen went to Oxford, Cambridge and King's College London.
- Tanmanjeet Singh Dhesi has been the Chief Financial Officer (CFO) of BPDTS since 1 April 2019. His role primarily involves financial management, planning, performance, and managing risks. More information at <https://www.gov.uk/government/people/mal-singh>. Singh Dhesi went to University College London, Keble College, Oxford, Fitzwilliam College, Cambridge
- Matt Hancock held the Secretary of State for Health and Social Care position from July 9, 2018, to June 26, 2021. Before this role, he was the Secretary of State for Digital, Culture, Media, and Sport in 2018. He went to University of Oxford, Cambridge.
- Keir Starmer became the leader of the Labour Party in 2020. In recent election in 2024, he became a Prime Minister of the UK. He went to University of Leeds and Oxford.

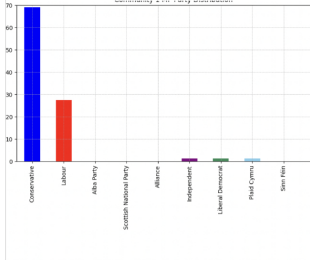
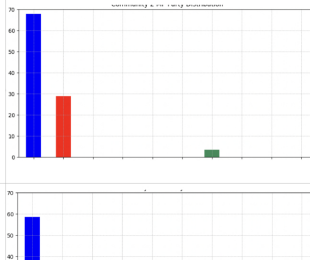
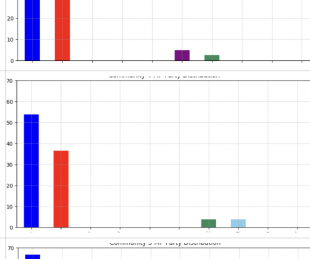
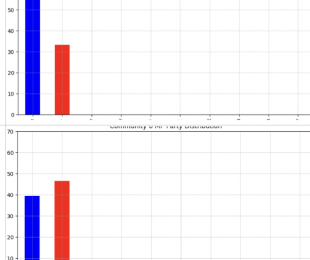
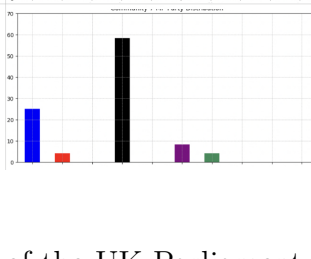


Communities	Description	Most common universities	Average age of memebes	Members and party membership (in %)
Community 1 (84 members, 59 Male, 25 Female)	Approximately 70% of the members in this community belong to the Conservative Party. 83 out of 84 members in this community went to Oxford and London School of Economics.	Oxford (83), London School Of Economics (5)	The average age in Community 1 is 53 with 95% confidence interval of (50, 55).	
Community 2 (59 members, 40 Male, 19 Female)	Approximately 70% of the members in this community belong to the Conservative Party, and around 30% are from Labour Party.	Cambridge (56) Oxford(4)	The average age in community 2 is 55 with 95% confidence interval of (52, 58).	
Community 3 (82 Members) 53 Male, 29 Female	Most of the members in this community belong to either Conservative or Labour Party.	University of London (29), DurhamUniversity(9), University of Sussex (10), University of Birmingham(10)	The average age in community 3 is 53 with 95% confidence interval of (50, 55).	
Community 4 (52 Members) 19 Female, 33 Male	In this community, approximately 36% of the members are from the Labour Party, and around 50% from the Conservative Party. We can observe more members of the Labour Party compared to the first and second community.	King's College London (13), University of Exeter (11), University College London(9), Aberystwyth University (9)	The average age in community 4 is 51 with 95% confidence interval of (48, 54).	
Community 5 (18 members) 11 Male, 7 Female	This community has only members of the Conservative Party and Labour Party, it is more balanced in terms of gender.	London School of Economics (16), Brunel University (3)	The average age in community 5 is 56 with 95% confidence interval of (49, 63).	
Community 6 (71 members), 39 Male, 32 Female	This community has more members of the Labour Party with around 45%.	University of Leeds(14), University of Hull (11), University of Salford (7), Open University (7)	The average age in community 6 is 52.96 with 95% confidence interval of (49, 55).	
Community 7 (24 members) 11 Female, 13 Male	This community has a small number of Labour Party and Independent members, but it is dominated (approximately 60%) by members from the Scottish National Party. The most common university where most of the members went is the University of Glasgow, which is also located in Scotland.	University of Glasgow (13) University of Stirling (6),	The average age in community 7 is 50 with 95% confidence interval of (45, 55).	

Figure 2.30: Community detection results and analysis of the UK Parliament

2.2.4 Community detection in UK Parliament

The Louvain algorithm was used for community detection analysis. Twenty-nine communities were detected. The communities were formed based on the members and the universities they shared. For further analysis, only communities with more than 15 members were considered. Seven communities were created after filtering the communities with less than 15 members, as shown in Figure 2.30.

To better understand the communities, another attribute, ‘Age,’ was scraped from the Wikipedia page of each community member. For each member, the party membership was also scraped from the [https://www.parliament.co.uk/MPs](https://www.parliament.parliament.co.uk/MPs) site. The genders of members of parliament were also considered as part of the analysis for each community.

The goal was to find if educational background can influence political structure in the UK parliament.

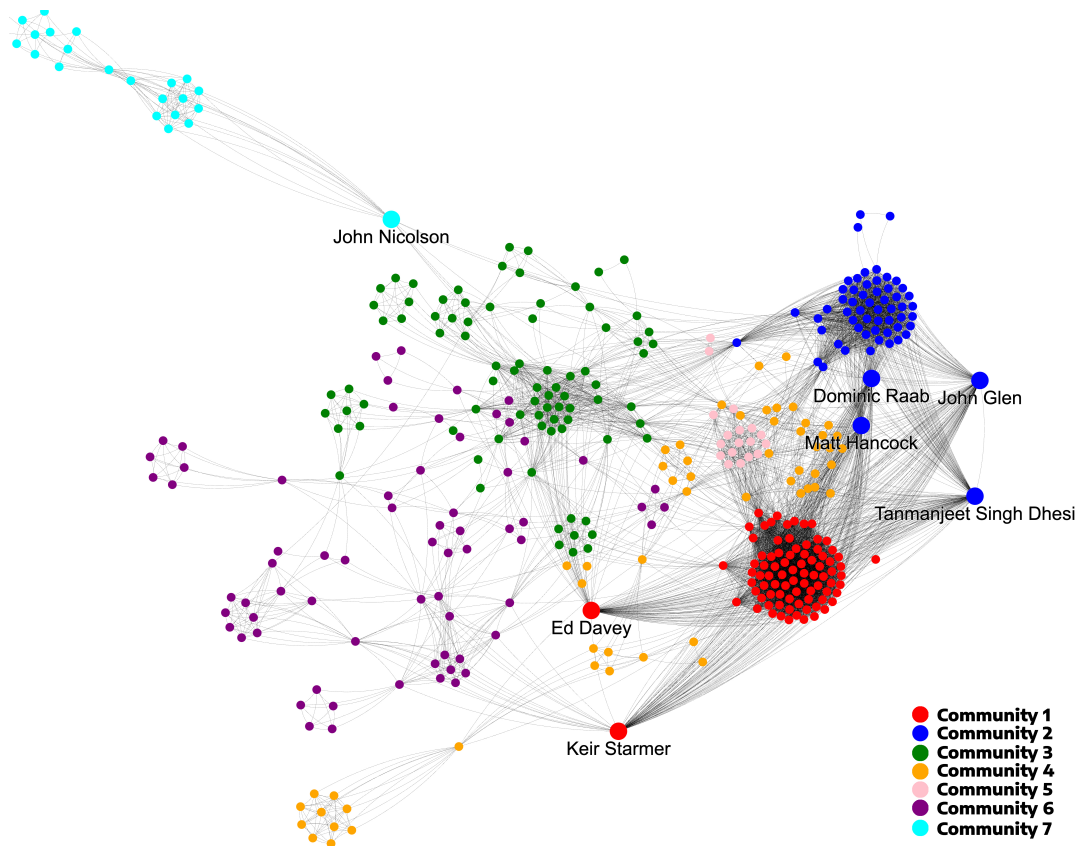


Figure 2.31: Communities in the UK Parliament

2.2.5 Kosovo parliament analysis

To get the data used for the analysis, members of parliament and their domain of study were extracted from the official government’s site: <https://www.kuvendikosoves.org/eng/deputyprofile/>. The dataset that is used to create the graphs as shown in Figure 2.32.

	Name	Party	Edukimi
0	Abelard Tahiri	PDK	Shkenca Politike
1	Adelina Grainca	LVV	Public Policy,International Relations,Strategi...
2	Adnan Rrustemi	LVV	Shkenca Politike
3	Ariana Matoshi	LVV	Arts
4	Agim Veliu	LDK	Law
...
76	Valentina Bunjaku-Rexhepi	LDK	Math
77	Valon Ramadani	LVV	Diplomacy
78	Vendenis Lahu	LVV	Medicine
79	Visar Korenica	LVV	International Business
80	Yllza Hoti	LVV	Economy, MBA

81 rows × 3 columns

Figure 2.32: The members of parliament in Kosovo

2.2.6 Graph structure

Each member of parliament is connected with another member if they have the same domain of study (e.g., if two members went to medicine, they would be connected). The graph in Figure 2.34 shows the members of parliament and their connections to other members who share the same domain of study.

The Networkx library was used to create the graph, and compute centrality measures such as Degree Centrality, Closeness Centrality, and Eigenvector Centrality. Figure 2.33 shows the centrality measure results.

Based on centrality measures: Yllza Hoti seems to be a key figure based on centrality measures. She is a central figure in this network with the most direct connections. Also, she is positioned so that she could connect to different parts of the network and other influential individuals.

2.2.7 Background on Kosovo parliament structure

Kosovo is the youngest Republic in the western Balkans and Europe, declaring its independence in 2008. The government of Kosovo is formed by the executive branch led by the President and the Prime Minister. The Assembly is the legislative body of the Republic of Kosovo, which includes members of parliament and government officials like the Prime Minister, the President, and ministers.

The Assembly of Kosovo elects the President. At least two presidential candidates must vote, and in the first two voting rounds, one must receive two-thirds or 80 votes from the Assembly to be elected President. If the first two rounds

Name	Degree Centrality	Betweenness Centrality	Closeness Centrality	Eigenvector Centrality
Yllza Hoti	0.211	0.0135	0.211	3.22
Blerta Deliu	0.134	0.0090	0.136	4.94
Hisen Berisha	0.192	0.0067	0.192	3.66
Armend Muja	0.173	0.0	0.178	3.14
Duda Balje	0.173	0.0	0.178	3.14
Agim Veliu	0.173	0.0	0.174	3.62
Average (Members)	0.176	0.0049	0.178	1.59
Average (All Members)	0.102	0.000655	0.112	6.17

Figure 2.33: Centrality measures for members of parliament in Kosovo

fail, the third vote requires fewer votes, where the majority of 61 out of the 120 votes is enough to elect the President. The term of a Presidency is five years, and there is a right for reelection for one additional term[39].

The President has to be a figure of unification who is neutral and facilitates dialogue between parties when there is disagreement on important topics. Kosovo has elected two female Presidents since its declaration of independence in 2008, demonstrating its high approval of female politicians. The role of the President is mostly ceremonial; however, the President serves as the commander of the armed forces of Kosovo and leads the country's diplomatic representation.

The Prime Minister is appointed by the President from a recommendation by the majority party or coalition in the Assembly of Kosovo. The Prime Minister presents his cabinet, including the Ministers, and the Assembly votes. The Prime Minister needs a majority of 61 votes to be elected and form the government.

The majority party in the Assembly always proposes the Head of the Assembly. If there is a coalition between parties, both the Prime Minister and the Head of the Assembly can be part of a non-majority party that is part of the coalition. The Assembly of the Republic of Kosovo is elected for a 4-year term [40], likewise, the Prime Minister; however, the term may end sooner if the government decides to disband. Democratic elections are held to choose the Members of the Assembly.

There are 120 seats in the Kosovo Assembly that are decided by the voters directly. Of the 120 seats, at least ten are reserved for Kosovar Serbs, and ten others are reserved for other minorities (Bosniaks, Turkish Roma, and other communities). The Assembly also has a gender quota to ensure a fair representation of women in the country's policy-making. At least one-third of the seats of the Assembly are reserved for women. Every municipality in Kosovo is led by a mayor and has its municipal Assembly, elected every four years or sooner if the mayor resigns. Municipal elections are generally held at different times than national elections.

For the national elections, one can vote for one party or coalition and five members of that part/coalition to become Members of the Assembly. For the local municipal elections, one can vote for one municipal mayor candidate and one candidate from any party to become a Member of the Municipal Assembly.

2.2.8 Community detection in Kosovo Parliament

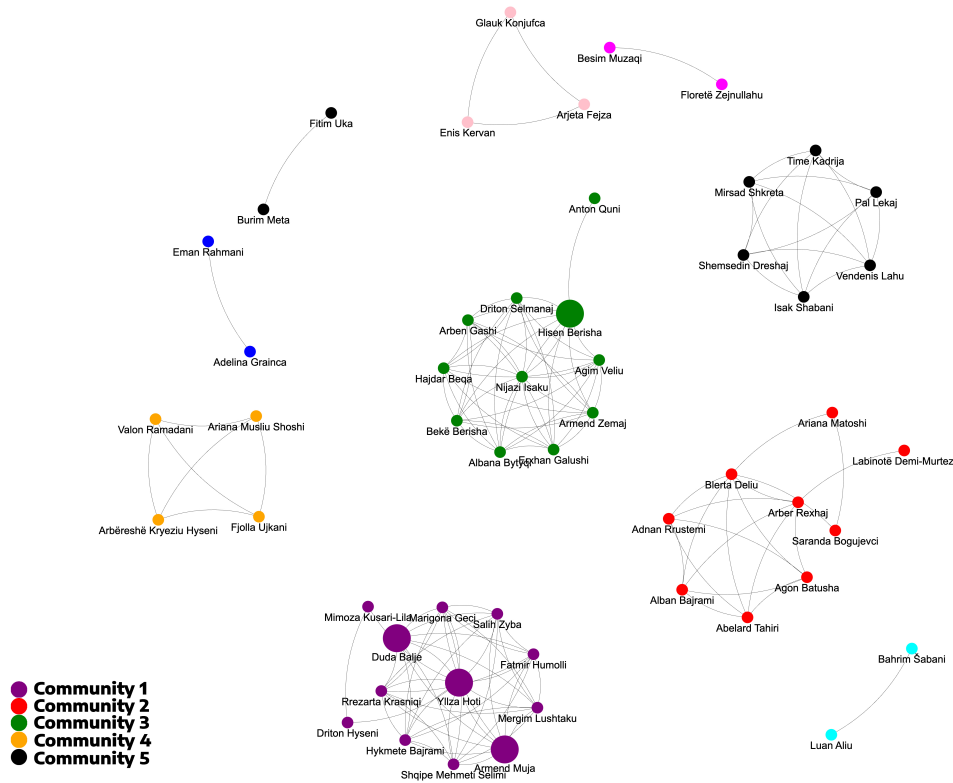


Figure 2.34: Network graph for parliament of Kosovo

The Louvain Algorithm was used to detect communities, and five communities were found. The tables below show each community, with their members, parties to which they belong, and their specialization (domain of study). To get more insights into communities that were formed, more information such as “age” about each member of communities was manually extracted from the same official website of deputies (<https://www.kuvendikosoves.org/eng/deputyprofile/>).

Communities 1, 2, and 4 are more similar, having a large ratio of members from the LVV party. These communities have close to equal representation of male and female members. Gender distribution in communities 3 and 5 is disproportionately male. Community 3 has ten males and one female, whereas Community 5 has five males and one female member. Community 3 has over 45% members from the LDK party, whereas Community 5 has 50% members from AAK. These two communities are more diverse in terms of party affiliation, while communities 1, 2, and 4 more closely reflect the LVV Party. The results of communities are shown in Figure 2.35.

Communities	Description	Most common universities	Average age of memebes	Members and party membership (in %)
Community 1 (7 Female, 5 Male)	The majority of the members in this community are from LVV party, and they studied economy.	10 out of 12 members studied economy. The other two members have a degree in MBA.	The average age in Community 1 is 51, with 95% confidence interval of (43, 59)	
Community 2 (5 Male, 4 Female)	This community has members of LVV and PDK, and most of the members here share a background on political sciences.	6 out of 9 members went to Political Sciences. The other two members went to Arts and one member studied Albanian Literature.	The average age in Community 2 is 41, with 95% confidence interval of (38, 45)	
Community 3 (10 Male, 1 Female)	This community is mixed, meaning the members are spread throughout different parties but share a common domain of study: LAW.	10 out of 11 went to LAW. Two members share the army too.	The average age in Community 3 is 51, with 95% confidence interval of (44, 58)	
Community 4 (3 Female, 1 Male)	This community has members from LVV and PDK, all members finished diplomacy.	All four members finished diplomacy, and two of them have diplomacy and public administration in common.	The average age in Community 3 is 32, with 95% confidence interval of (25, 39)	
Community 5 (5 Male, 1 Female)	This community is again mixed, where all the members went to medicine.	All members went to a medicine.	The average age in Community 3 is 56, with 95% confidence interval of (48, 63)	

Figure 2.35: The community detection results for Kosovo parliament

2.3 Experimental results: OULAD dataset

2.3.1 Understanding the data

Introduction

The educational dataset of Open University courses in the UK is used in this thesis. The dataset contains information about students and their performance at this university.

The Open University (OU) is a public research institution with the largest number of students in the UK. Most undergraduate students at this university are in the UK, and the courses are primarily online (off-campus). The university offers both undergraduate and postgraduate programs, and they can also be taken online globally [41].

This dataset combines students' details with clicks in the Virtual Learning Environment (VLE). This data could be helpful in analyzing students' online behavior while studying. The dataset includes data on seven courses involving 32,593 data points (students), their exam scores, and a record of how they interact with the VLE, given by the summaries of their clicks (a total of 10,655,280 data points). This dataset is available for free at https://analyse.kmi.open.ac.uk/open_dataset.

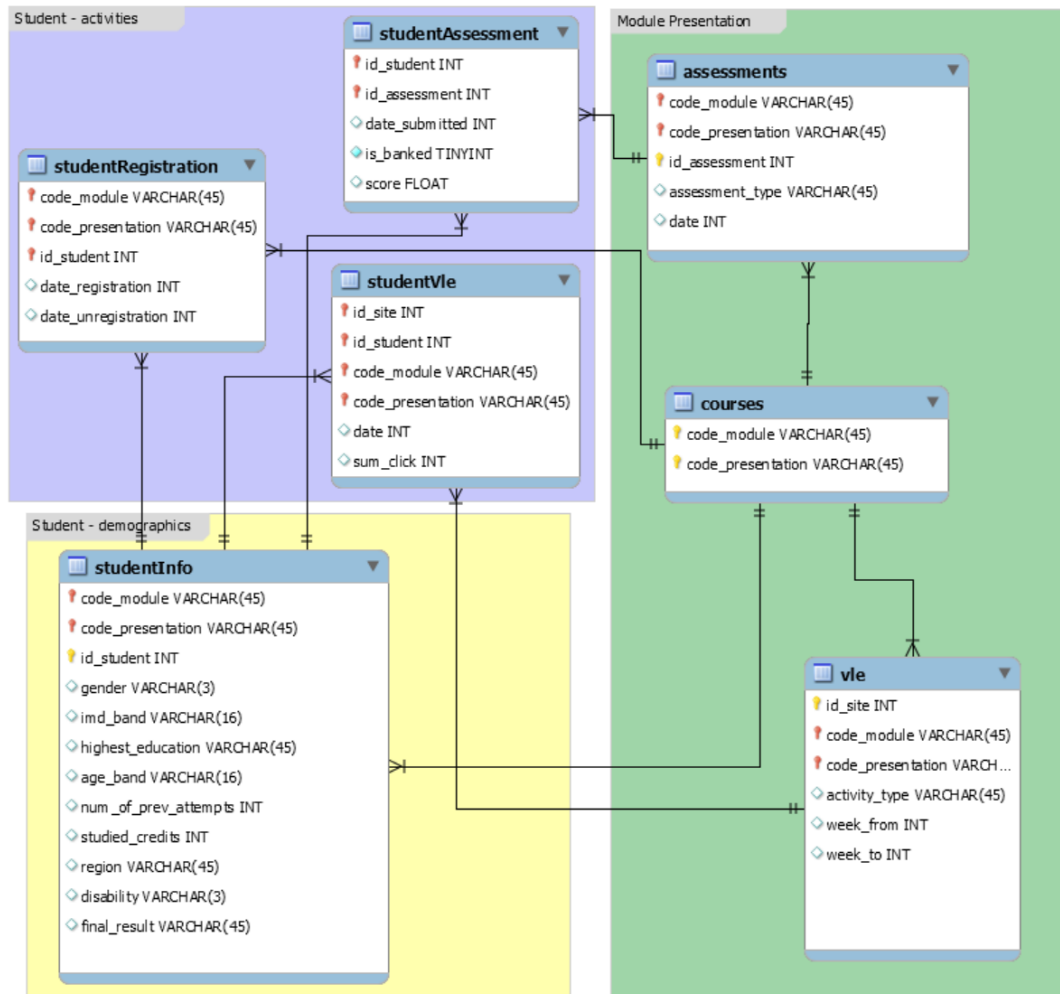


Figure 2.36: Schema of the dataset

Data overview

The OULAD dataset includes CSV format tables with student information from 2013 and 2014. The given tables are joined with each other using identifiers on specific columns. The dataset provides details on students' demographic backgrounds and course registrations. It covers course data, student evaluations, and their engagement with the Virtual Learning Environment (VLE) in seven courses. Course sessions begin in February and October, and they are described by "B" (if they started in February) and "J" (for starting in October). Figure 2.37 shows how a module (course) is structured throughout the year (time).

The course materials and content are available through VLE (Virtual Learning Environment) a few weeks before the course starts. Students are given various assignments and quizzes throughout the course, with a final exam at the end of each course.

Figure 2.36 shows the attributes of each table and how each table is linked with the other tables in the dataset using the identifiers. For example, the studentInfo table from the schema can be joined with studentRegistration using the student ID as the identifier. Merging the tables results in a combined table containing

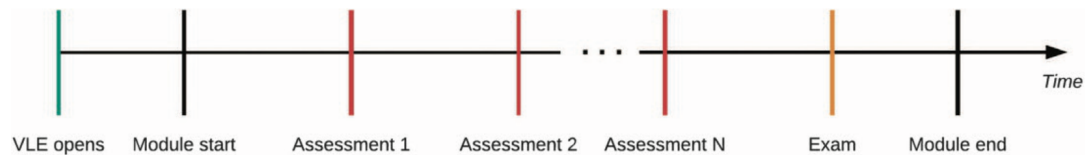


Figure 2.37: Module/Course structure over time

student information and their registration details.

Data records

This section provides a detailed description of each table in the dataset based on documentation on this dataset, which can be found at: https://analyse.kmi.open.ac.uk/open_dataset.

Courses Table

	code_module	code_presentation	module_presentation_length
0	AAA	2013J	268
1	AAA	2014J	269
2	BBB	2013B	240
3	BBB	2013J	268
4	BBB	2014B	234
5	BBB	2014J	262
6	CCC	2014B	241

Figure 2.38: Student's courses table

The 'courses.csv' file lists all the possible courses given by the code module column and the year they were presented with the column code presentation. Courses table is shown in Figure 2.38.

Assessments table

In 'assessments.csv,' details of each assessment are stored for each course presented in a specific year. The columns in assessments.csv include assessment ID, type of assessment, date or the deadline, and the assessment weight, which, if it

	code_module	code_presentation	id_assessment	assessment_type	date	weight
0	AAA	2013J	1756	TMA	215.0	30.0
1	AAA	2014J	1760	TMA	117.0	20.0
2	BBB	2013B	14988	TMA	159.0	18.0
3	BBB	2013J	14998	TMA	96.0	18.0
4	BBB	2014B	15018	CMA	152.0	1.0
5	BBB	2014J	15025	Exam	NaN	100.0
6	CCC	2014B	24286	CMA	18.0	2.0

Figure 2.39: Student's assessment types and weights

is an exam, has a 100%, and for other assessments, they should add up to 100%. Assesments table is shown in Figure 2.39.

VLE table

	id_site	code_module	code_presentation	activity_type	week_from	week_to
0	547035	AAA	2013J	resource	NaN	NaN
1	877064	AAA	2014J	oucontent	NaN	NaN
2	543204	BBB	2013B	resource	NaN	NaN
3	704097	BBB	2013J	resource	NaN	NaN
4	768523	BBB	2014B	resource	NaN	NaN
5	913626	BBB	2014J	resource	NaN	NaN
6	730101	CCC	2014B	resource	NaN	NaN

Figure 2.40: Table of materials available in the VLE

Information about the materials in the VLE is found in 'vle.csv.' This includes the site (identifier of each material available), module, and presentation codes, which provide information about the course and the type of activity. Materials in VLE table is shown in Figure 2.40.

Student information table

The 'studentInfo.csv' file contains demographic and academic data about students, covering details like module and presentation codes, student ID, gender, region, education level, and other personal data. Student information table is shown in Figure 2.41.

	code_module	code_presentation	id_student	gender	region	highest_education	imd_band	age_band	num_of_prev_attempts	studied_credits	disability	final_result
0	AAA	2013J	2694424	M	East Anglian Region	A Level or Equivalent	70-80%	35-55	0	120	N	Pass
1	AAA	2014J	402727	F	East Anglian Region	A Level or Equivalent	60-70%	0-35	0	60	Y	Pass
2	BBB	2013B	476457	F	North Region	Lower Than A Level	20-30%	0-35	1	120	N	Withdrawn
3	BBB	2013J	338878	F	West Midlands Region	A Level or Equivalent	80-90%	35-55	2	60	N	Withdrawn
4	BBB	2014B	607436	F	London Region	Lower Than A Level	20-30%	35-55	0	180	Y	Withdrawn
5	BBB	2014J	606861	F	Yorkshire Region	A Level or Equivalent	0-10%	0-35	0	60	N	Pass
6	CCC	2014B	632134	M	Yorkshire Region	A Level or Equivalent	30-40%	35-55	0	60	N	Withdrawn

Figure 2.41: Student information table

Student registration table

	code_module	code_presentation	id_student	date_registration	date_unregistration
0	AAA	2013J	57506	-103.0	NaN
1	AAA	2014J	375260	-122.0	NaN
2	BBB	2013B	2366070	-21.0	31.0
3	BBB	2013J	491758	-115.0	-22.0
4	BBB	2014B	2652209	-24.0	NaN
5	BBB	2014J	695084	-11.0	NaN
6	CCC	2014B	424112	-17.0	NaN

Figure 2.42: Student's registration table

The 'studentRegistration.csv' provides details regarding student registration for each module presentation, including module and presentation codes, student IDs, and dates of registration and unregistration. Student Registration table is shown in Figure 2.42.

Student assessment table

The 'studentAssessment.csv' records the student assessment results, including student IDs, submission dates, assessment status, and scores. Student Assessment table is shown in Figure 2.43.

Student VLE table

Lastly, 'studentVle.csv' tracks student interactions with VLE materials, including module and presentation codes, student and material IDs, the date of interaction, and the number of clicks. This table is shown in Figure 2.44.

	id_assessment	id_student	date_submitted	is_banked	score
0	1752	11391	18	0	78.0
1	1752	28400	22	0	70.0
2	1752	31604	17	0	72.0
3	1752	32885	26	0	69.0
4	1752	38053	19	0	79.0

Figure 2.43: Student assessments table

	code_module	code_presentation	id_student	id_site	date	sum_click
0	AAA	2013J	331358	546657	37	11
1	AAA	2014J	2273119	877237	1	1
2	BBB	2013B	276717	542864	90	15
3	BBB	2013J	585779	703737	115	2
4	BBB	2014B	614745	768376	30	7
5	BBB	2014J	474408	913504	90	2
6	CCC	2014B	1095916	729671	111	4

Figure 2.44: Students and their total number of clicks in specific materials in the VLE

The data science lifecycle

The data science lifecycle in Figure 2.45 was followed for the following experiments. The first step was defining the problem or the initial goals we wanted to achieve. Since the main area of interest was the analysis of educational datasets, several online and open-source datasets were analyzed, including articles and papers related to them, and the Open Learning Analytics Dataset (OULAD) was chosen for this thesis.

This dataset was chosen for analysis since it offers a large set of features such as demographic information, course details, assessments, and the interactions students have in a virtual learning environment based on the number of times they click on the online materials. The dataset also has many students and their accurate labels (i.e., whether they passed or failed the courses they registered for). This variety in features and a large number of students makes it a suitable dataset

2.3.2 Data analysis

The following sections describe the main findings of the data analysis. Detailed visualizations for each feature can be accessed online in the appendix of my thesis on GitHub:

- Gender Distribution Pie Chart - Majority male (54.8%), females (45.2%).
- Education Level - Most students have 'A Level or Equivalent' (43.09%) or 'Lower Than A Level' (40.37%). A smaller percentage have 'HE Qualification' (14.51%), and very few have 'No Formal Quals' (1.06%) or 'Post Graduate Qualification' (0.96%).
- Student's Distribution Across Regions - Highest in Scotland (10.57%), lowest in Ireland (3.63%).
- Age Band Distribution - Majority under 35 (70.4%), 35-55 (28.9%), above 55 (0.63%).
- Number of Previous Attempts - Majority on first attempt (87.2%).
- Disability Distribution - Declared disability (9.71%), no disability (90.3%).
- Studied Credits Distribution - Peak at 0-100 credits, decreases as credits increase.
- Enrollment of Students in Courses - Highest in 'BBB' course (8000 students), lowest in 'AAA' (700 students).
- Total Student's Clicks on Specific Materials Available - Total clicks by students on VLE materials.

Comparison of age groups and their final results

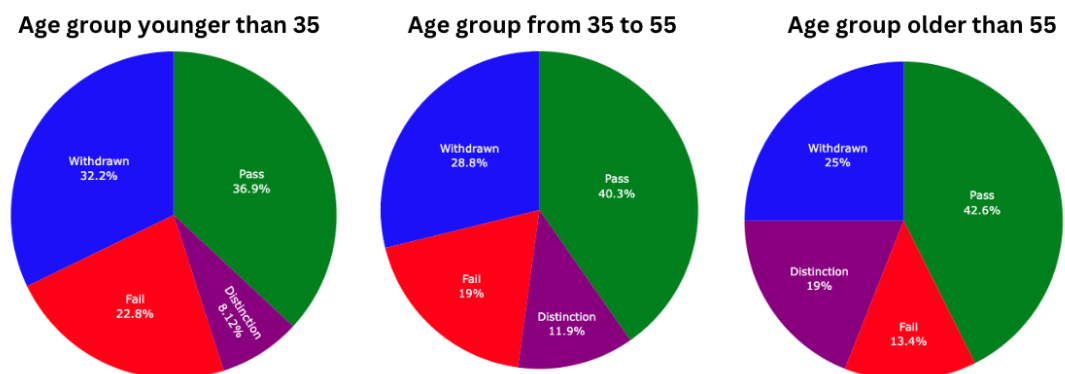


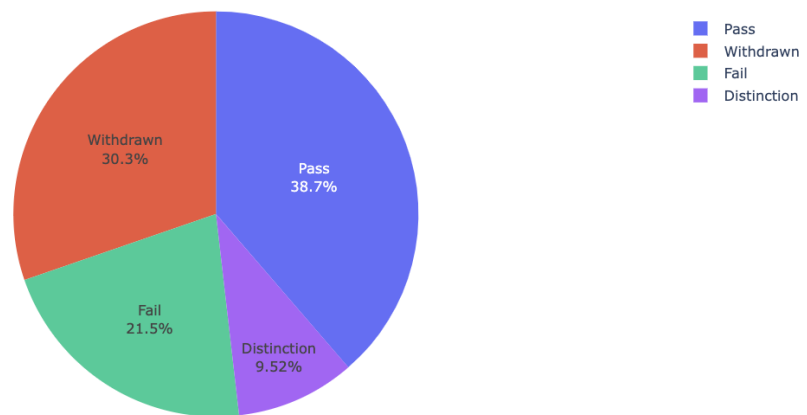
Figure 2.46: Comparison of Age Groups and their final results

Figure 2.46 shows the comparison of different age groups and their final results, with the first plot showing students younger than 35 years old, the second plot

showing students between 35 and 55 years old, and the last plot showing the final results of students over 55 years old. As we can see, there is not much difference in passing and failing rates across different age groups.

Comparison of student's gender and their final results

Final Result Distribution for Disability Status: N



Final Result Distribution for Disability Status: Y

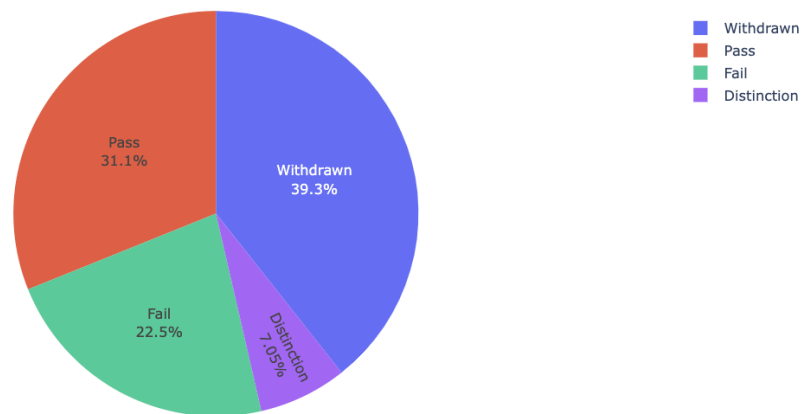


Figure 2.47: Comparison of student's gender and their final results

Figure 2.47 shows how students' gender relates to their final results. As we can see, there's not much difference in their final results.

Comparison of credits enrolled and their final Results

Figure 2.48 shows that students with more credits enrolled have a higher chance of withdrawing.

Distribution of Studied Credits by Final Result

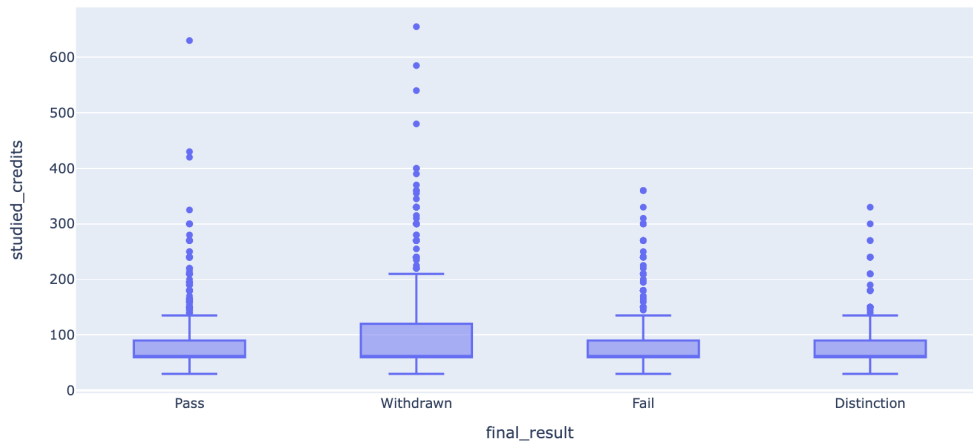


Figure 2.48: Comparison of credits enrolled and their final results

Comparison of Students IMD band and their final results

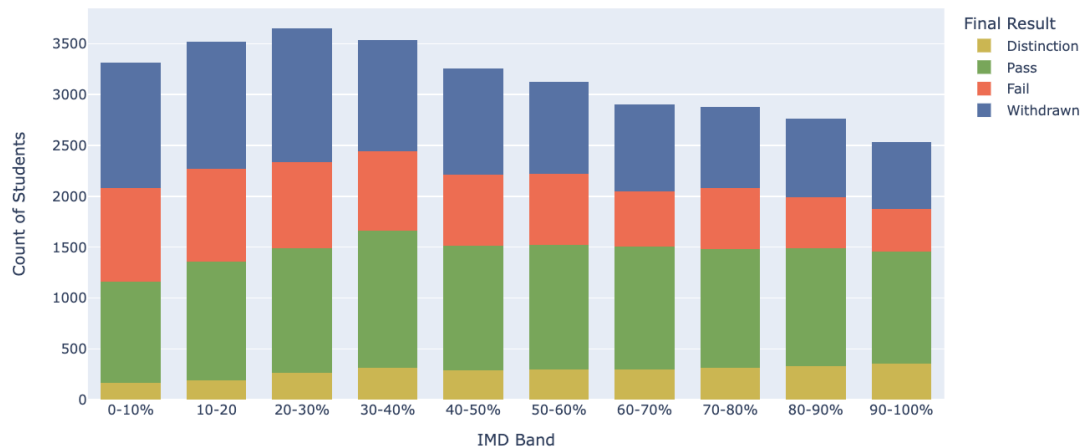


Figure 2.49: Comparison of students IMD band and their final results

Figure 2.49 shows that the fail rate decreases as the IMD band increases (Deprivation index (IMD) which is a measure of deprivation, for more info, visit this link [IMD index](#). This could mean that students with better economic conditions might have a higher chance of passing the courses.

Comparison of each student’s highest education level group and their final results

Figure 2.50 shows that withdrawing is the highest among students without formal qualifications. Postgraduates have the lowest fail rate compared to other levels of education.

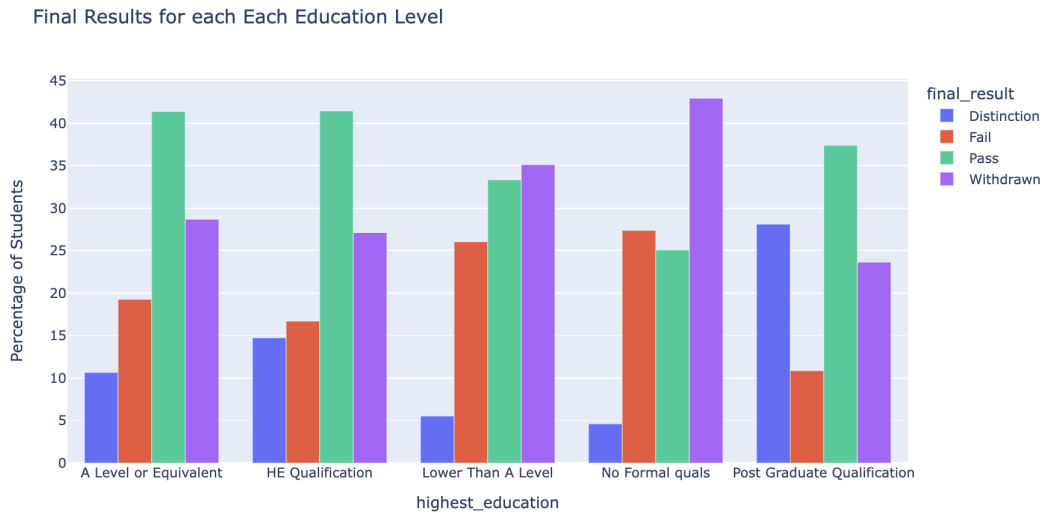


Figure 2.50: Comparison of each student’s highest education level group and their final results

Comparison of student’s total clicks and their final results

Figure 2.51 shows that the more students clicked on Virtual Learning Environment (VLE) materials (meaning they were actively participating in the courses), the higher their chances of passing the courses. Additionally, the students labeled with distinction (higher than just a pass) have the highest number of clicks.

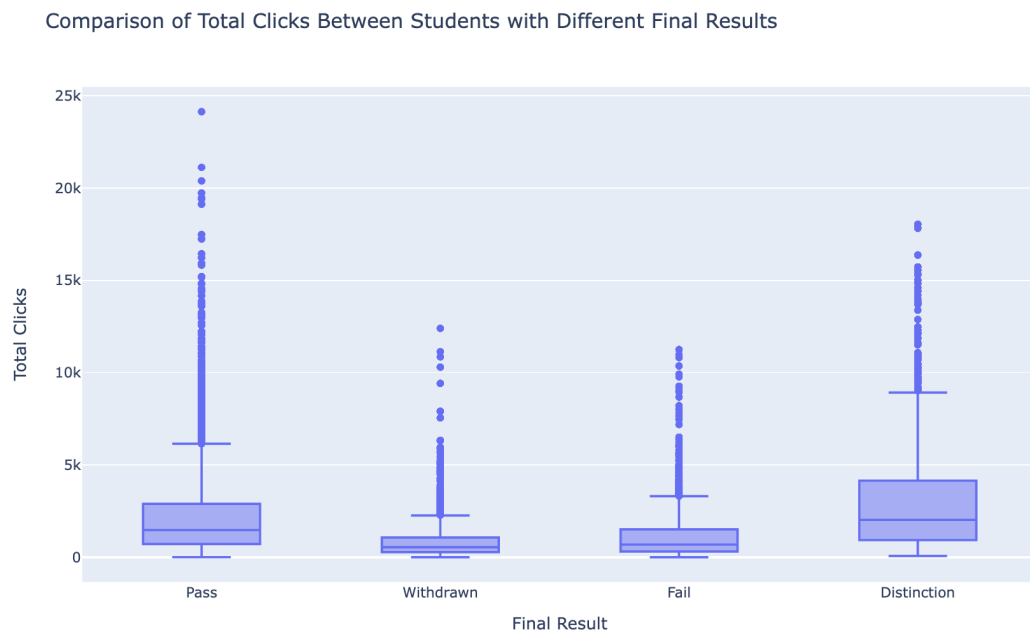


Figure 2.51: Comparison of student’s total clicks and their final results

Student's clicks in different age groups

Figure 2.52 shows that students tend to click less on the online materials as the age increases.

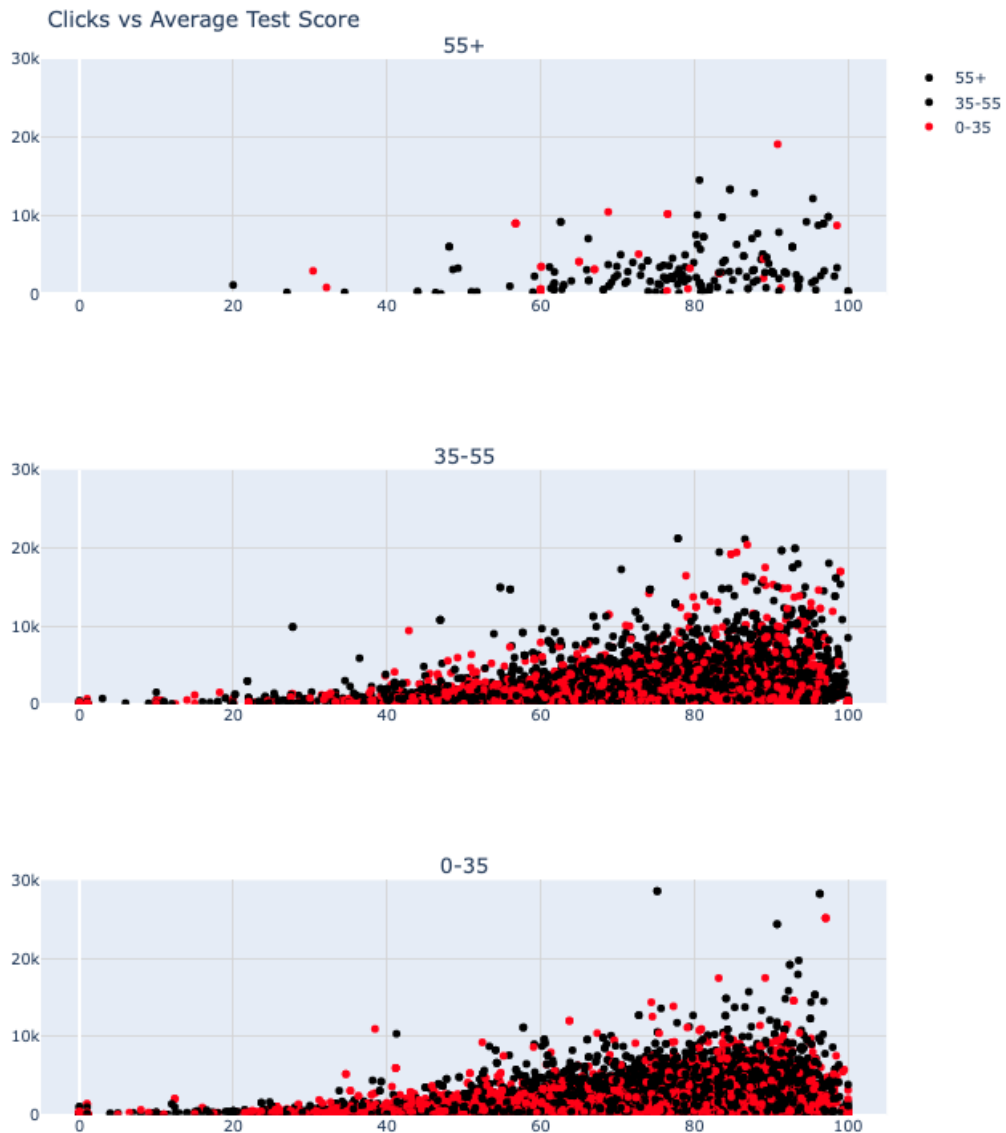


Figure 2.52: Student's clicks in different age groups

Total student's clicks and their average score on assessments

Figure 2.53 shows that the average score on all the assessments is higher for students who clicked more on the online materials. Again, this indicates a relationship between participation and performance.

Correlation matrix of numerical features

Figure 2.54 shows the correlation between numerical features. As we can see, a positive correlation exists between students' scores and their early clicks and

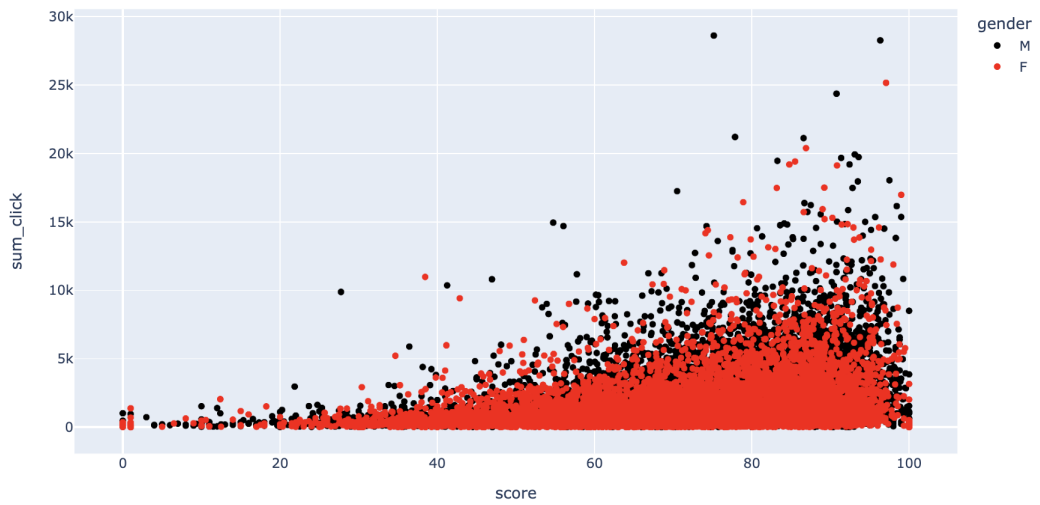


Figure 2.53: Total student's clicks and their average score on assessments

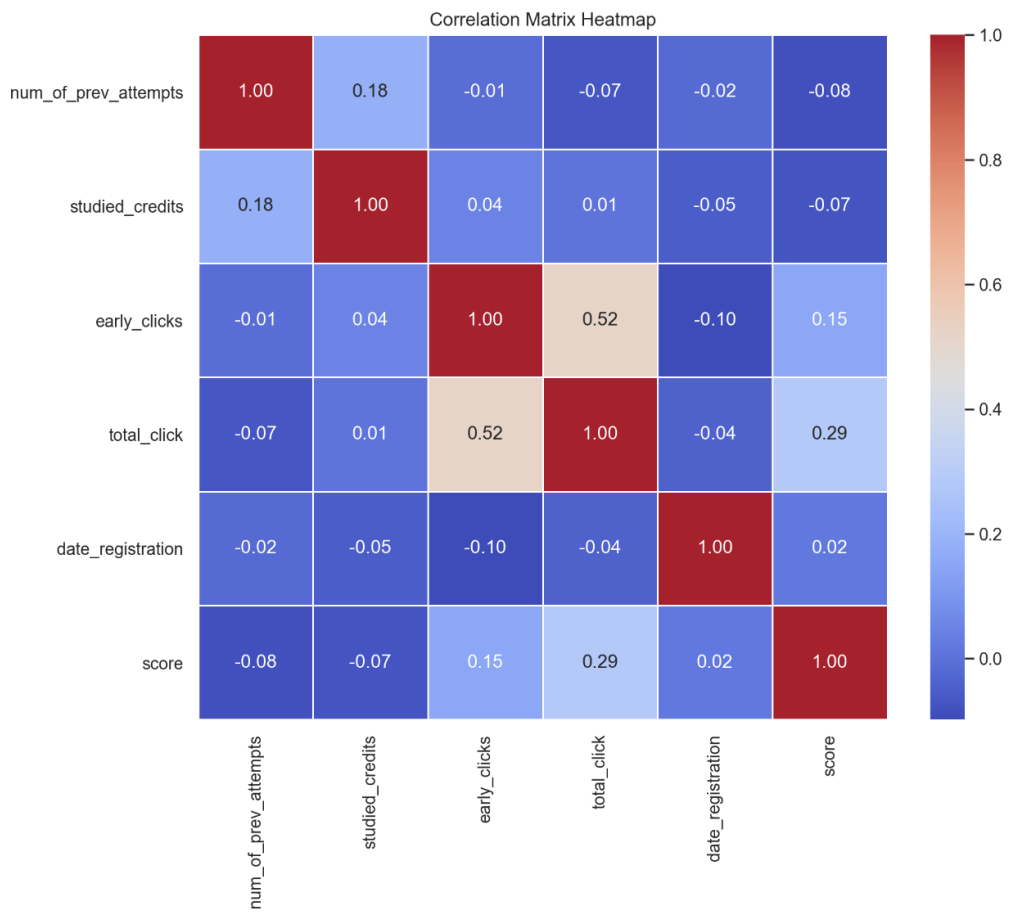


Figure 2.54: Correlation Matrix of numerical features

total clicks. This indicates that the clicks are important in the score of a student. Also, a correlation of registration date positively correlates with a score, but since the value is very small, it is not significant.

Experiments and goals

After understanding the data, the problem statement was defined, with the goals and experiments that could be done using this data. Experiments that could be done using the data:

1. Developing a predictive model that predicts students' final results for any given course. This experiment can be helpful in determining which are the most important features that contribute to students' success. These features can be helpful in various ways, for example, the teachers can have a better understanding of what is important for students to succeed.
2. Developing a predictive model for predicting students' success using only demographic features. This could be helpful in understanding if demographic features alone can predict students' success.
3. Creating a model that can be used to predict if a student will pass or fail a course before the end of the module. This way, the teachers will be able to identify which students are at risk of failing earlier in the course, and they can intervene in time. This could increase the chances of students passing the course.

Continuing on the data science lifecycle 2.45, the next step was to prepare the data for the experiments.

Data Preparation

The first step in data preparation was to check the data quality. This involves checking for missing values, inconsistencies in the data, duplicates, etc.

The issues that were found on the dataset are listed below:

1. There were students who were unregistered, but their final result was recorded as pass or fail. Nine of them were identified and removed from the dataset.
2. When merging the tables (CSVs) using identifiers, some students were present in the student info table, but their clicks for specific materials were missing. Also, some of the student's scores were missing. From 32593 students, after merging with studentVle, studentRegistration using inner join (which drops the students if they do not exist in both tables) 29228 students remained. After merging with the students' assessments table, 26331 students remained.
3. The documentation defines that the exams weight 100%. However, in the course 'CCC,' the weight of the exams is 200%, which is wrong. This was modified to 100%. Course 'GGG' has the correct exam values, however the TMA and CMA assessments which must sum up to 100 while the exam weight is fixed at 100 are missing, they are imputed directly with TMA being 100.

Handling missing data

To find the missing data, each table was checked for missing values. The column 'IMD BAND' in the student info table has missing values, with 3.4% of data points missing. The missing values will be imputed using a specific 'region's most frequent 'IMD BAND' value. This will be done by region because different regions might have different 'IMD BAND' values, as they represent the students' socio-economic status.

In the student registration table, the column data_registration has 0.1% of missing values, which are going to be imputed with the median value of the whole column.

The VLE table, as shown in Figure 2.40, has 82.4% of missing values in the week from and week to columns. Since there are too many missing data points for week from and week to, which represent the period in which materials were supposed to be used, these columns will be dropped and not used for our final dataset.

The code for imputing the missing values in the 'IMD BAND' column is shown below:

```
#Iterate on regions
for region in student_info_enhanced['region'].unique():
    # Find the mode
    most_frequent_imd_band = student_info_enhanced[student_info_enhanced['region'] == region]['imd_band'].mode()[0]

    # Check for nan
    if pd.notna(most_frequent_imd_band):
        mask = (student_info_enhanced['imd_band'].isnull()) & (student_info_enhanced['region'] == region)
        # Update missing rows
        student_info_enhanced.loc[mask, 'imd_band'] = most_frequent_imd_band
```

Figure 2.55: Code on imputing missing values on 'IMD BAND' column

Figure 2.55 shows a code on how the missing values were handled for the 'IMD BAND' column. First, we loop through each region, and for each region, we find the mode (the most frequent IMD BAND value) Then, we find all rows in the region where the 'IMD BAND' is missing and replace it with the region's most frequent 'IMD BAND' value. A similar procedure replaced the missing values in the 'registration' column. Instead of mode, we used Median for that since it is a numerical feature.

Handling inconsistencies in the data

To fix inconsistencies that were mentioned before, such as: Some students were unregistered, but their final results were recorded as passing or failing. The following code 2.56 was used to fix this issue.

In this Figure 2.56 first all the unregistered students are selected, and then after merging with the student info table (to get their final results), we select all the students whose final result is not 'withdrawn'.

This way, we find all students who were unregistered but had different incorrect labels assigned. Following this, we precisely identify their indexes and remove them from the original table, ensuring the accuracy by manipulating the data.

```

## GET all unregistered students
unregistered_students = studentRegistration[studentRegistration['date_unregistration'].notnull()]\
[['id_student', 'code_module', 'code_presentation', 'date_unregistration']]

# Merge the unregistered students registration table with student info ,to get their final results
merged_data = unregistered_students.merge(studentInfo, on=['id_student', 'code_module', 'code_presentation'])

# Filter out the students whose final result is not marked as "Withdrawn"
inconsistencies = merged_data[merged_data['final_result'] != 'Withdrawn']

# Indices of the students with incorrect final results
incorrect_indices = inconsistencies.index

# Displaying the rows with incorrect final results
print(inconsistencies.index)
student_info_enhanced.drop([719, 724, 869, 4961, 5010, 5293, 7853, 8001, 8340], inplace=True)

```

Figure 2.56: Code on removing wrongly labeled students

Another inconsistency that was found was regarding the weights of student’s assessments. Based on the documentation the exams should have a weight of 100%, and the other assessments should add up to 100%. However, this is not the case for some courses, where course encoded as ”CCC” course ”CCC” has exams that are weighted 200%, and the course ”GGG” has missing weights for assessment type TMA and CMA; they must be handled to sum up to 100.

In order to fix these issues and have everything consistent with the documentation, we assume that the values on weights of the exam with 200% are wrong, so they are replaced with 100%. For missing values of weights of other assessments (TMA and CMA) in the ”GGG” course, the weights are input such that they add up to 100%. To fix this, the following code 2.57 was used to fix these inconsistencies in weights assesments of ”GGG” and ”CCC” course.

```

assessments['weight'] = np.where(
    (assessments['code_module'] == 'CCC') & (assessments['assessment_type'] == 'Exam'),
    assessments['weight'] / 2,
    assessments['weight']
)

assessments['weight'] = np.where(
    (assessments['code_module'] == 'GGG') & (assessments['assessment_type'] == 'TMA'),
    100 / 3,
    assessments['weight']
)

```

Figure 2.57: Code on fixing wrongly labeled weights of assessments

2.3.3 Introduction to experiments

After preprocessing the data and fixing the inconsistencies, the final data has 36 features and 21663 data points shown in Table 2.1.

Three experiments were done using these features, each with a different goal and subset of the features presented above.

- Experiment 1 - Includes 35 features and 21663 data points. The goal is to find the most predictive features that contribute to students’ success (pass/fail).

Feature	Feature	Feature
1. <i>code_module</i>	13. <i>date_registration</i>	25. <i>ouelluminate_clicks</i>
2. <i>gender</i>	14. <i>score</i>	26. <i>ouwiki_clicks</i>
3. <i>region</i>	15. <i>dataplus_clicks</i>	27. <i>page_clicks</i>
4. <i>highest_education</i>	16. <i>dualpane_clicks</i>	28. <i>questionnaire_clicks</i>
5. <i>imd_band</i>	17. <i>externalquiz_clicks</i>	29. <i>quiz_clicks</i>
6. <i>age_band</i>	18. <i>folder_clicks</i>	30. <i>repeatactivity_clicks</i>
7. <i>num_of_prev_attempts</i>	19. <i>forumnng_clicks</i>	31. <i>resource_clicks</i>
8. <i>studied_credits</i>	20. <i>glossary_clicks</i>	32. <i>sharedsubpage_clicks</i>
9. <i>disability</i>	21. <i>homepage_clicks</i>	33. <i>subpage_clicks</i>
10. <i>final_result</i>	22. <i>htmlactivity_clicks</i>	34. <i>url_clicks</i>
11. <i>early_clicks</i>	23. <i>oucollaborate_clicks</i>	35. <i>Domain</i>
12. <i>total_click</i>	24. <i>oucontent_clicks</i>	

Table 2.1: List of Features

- Experiment 2 - Includes only demographic features; the goal is to find out if, by only using students' demographic information, we could predict their success.
- Experiment 3 - Includes all the features. The goal is to optimize and develop a model that could predict students at risk before the end of the course.

The distribution of the feature 'Final Result', which can either be Pass or Fail, is shown in Figure 2.58:

From Figure 2.58, we can observe that the dataset is imbalanced, meaning more students passed the course than students who failed the course. This means that the dataset is imbalanced. Evaluation metrics such as f1-score are used to better understand the model's performance on imbalanced datasets.

Decision tree implementations such as CART, Random Forest, AdaBoost, and C4.5 were used for each experiment. For CART, Random Forest, and Adaboost implementations, the scikit-learn library was used, and for the C4.5 implementation, the Weka library was used.

The final results of each classification algorithm on the test data are shown in Figure 2.59: The original documentation does not provide much information about the types of clicks already in the feature set. In this study [43], a description of each type of click was provided.

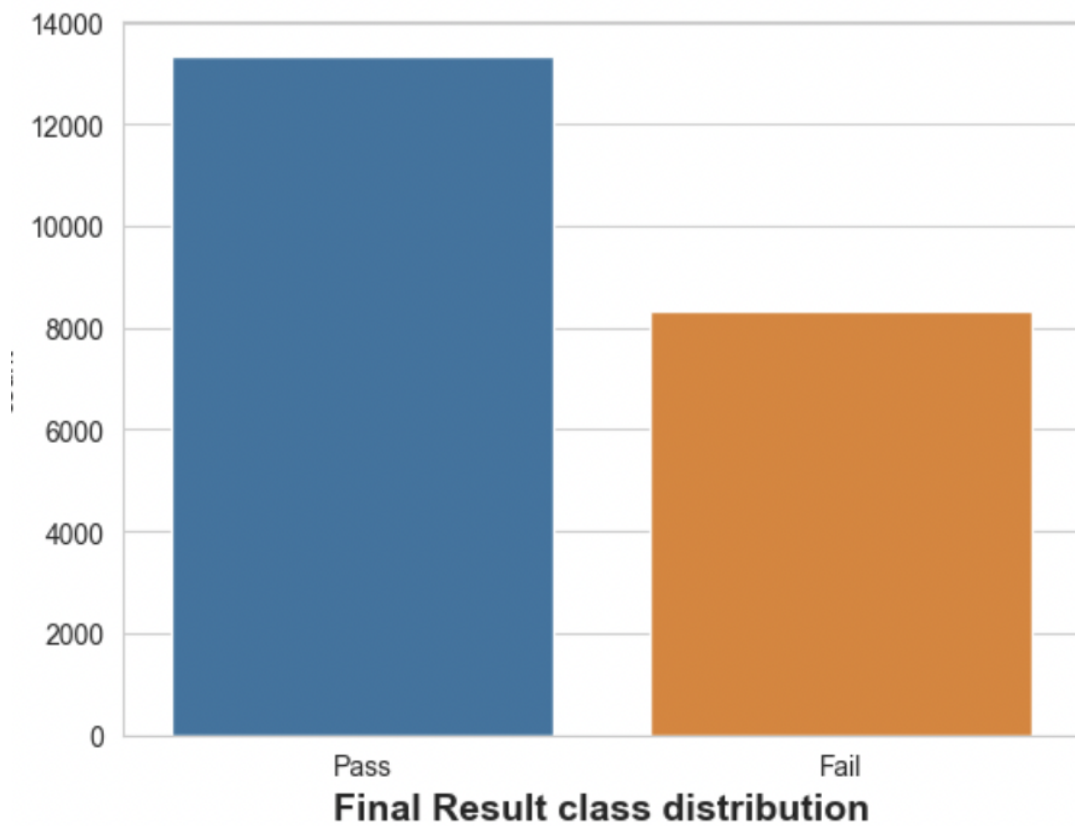


Figure 2.58: Distribution of the ‘Final Result’ feature.

Method	F1-Score	Precision	Recall	Accuracy	# Features
Ex.1 - Random Forest	0.86	0.86	0.86	0.86	35
Ex.1 - Ada Boost	0.84	0.84	0.84	0.84	35
Ex.1 - CART (No Pruning)	0.80	0.80	0.80	0.80	35
Ex.1 - CART (Post Pruning)	0.79	0.78	0.79	0.79	35
Ex.1 - C4.5	0.83	0.83	0.83	0.83	35

Figure 2.59: Final results of each classification algorithm on the test data.

2.3.4 Experiment 1 - Random forest

Using the final dataset with features introduced in Table 2.1, the original data frame was modified to have an extra categorical feature, which has its corresponding domain for each course (i.e. if it is a STEM course or Social Science). The code snippet in Figure 2.60 shows the modification, where the method gets the original data and creates an extra column, 'Domain,' based on the domain to which each course belongs.

Then the data frame was further processed where numerical features were normalized using the StandardScaler method of the scikit-learn library, categorical features such as 'code_module', 'gender', 'region', 'disability', and 'Domain' were transformed using the OneHotEncoder technique, and features like 'imd_band', 'highest_education', 'age_band' were transformed using the ordinal encoding technique. After transforming the features and adding the Domain feature, a randomized grid search CV method was used to find the best parameters of the Random Forest Algorithm such that the f1 score is maximized.

The classification results are shown in Figure 2.59.

The most important features identified by the random forest classifier are shown in Figure 2.61:

```
def determine_domain(code_module):  
    if code_module in ['CCC', 'DDD', 'EEE', 'FFF']:  
        return 'STEM'  
    else:  
        return 'Social Sciences'  
  
df['Domain'] = df['code_module'].apply(determine_domain)
```

Figure 2.60: Code modification for adding the 'Domain' feature.

Interpretation

From the plot shown in Figure 2.61, we can observe that the most important features are the total clicks that the student has with the learning environment during the semester. Other important features are related to the types of clicks that the students were mostly interacting with, and from this plot, we can see that quiz clicks are important. This measures the student's activity (number of clicks) in quizzes, and another important feature aside from clicks shown in the top ten most important features is if the Domain is STEM or Social Sciences. These are the most important features that were extracted using Random Forest.

2.3.5 Experiment 1 - AdaBoost

Before using AdaBoost, similar to Random Forest, another column, 'Domain' was added to the original data, and the features were scaled or transformed in the same approach as in the Random Forest experiment. Similarly to random forest

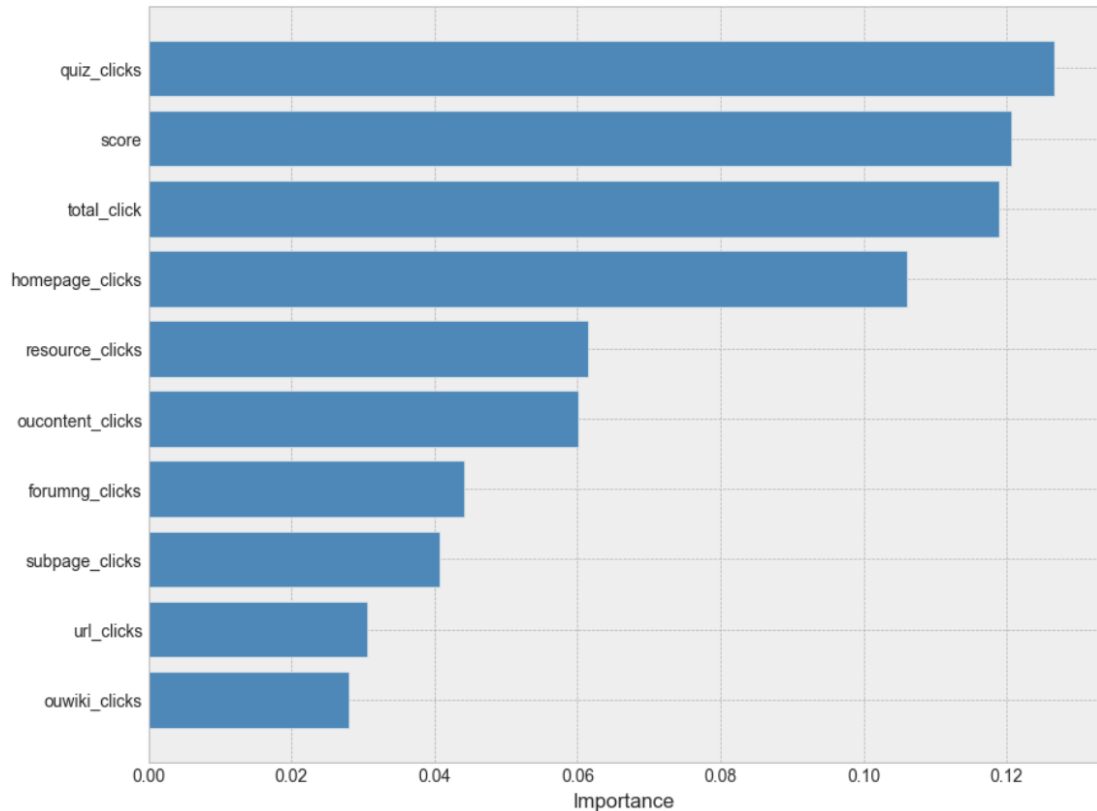


Figure 2.61: Most important features identified by the random forest classifier.

implementation, in AdaBoost, the goal was to maximize the f1 score, and techniques such as randomized grid search CV were used to find the best parameters for the model. The classification results are shown in Figure 2.59.

The most important features identified by the AdaBoost classifier are shown in Figure 2.62:

Interpretation

From the plot shown in Figure 2.62, we can observe that the most important features are the scores from students' assessments, similar to feature importances of Random Forest in Figure 2.61, quiz clicks are also significant in predicting success or failure from the AdaBoost model. Other features in top 10 are related to clicks in the virtual learning environment. Additionally, one specific course (FFF course) seems to be an important feature as well.

2.3.6 Experiment 1 - CART

Similarly to the previous experiments Random Forests and Ada Boost, first, the new Domain feature was added to the original data, and then categorical variables were transformed using one-hot encoding or ordinal encoding, similar to AdaBoost and Randomforest. In this experiment, numerical features were not scaled using any method. The original values of each click were retained. This helps in interpreting the resulting tree.

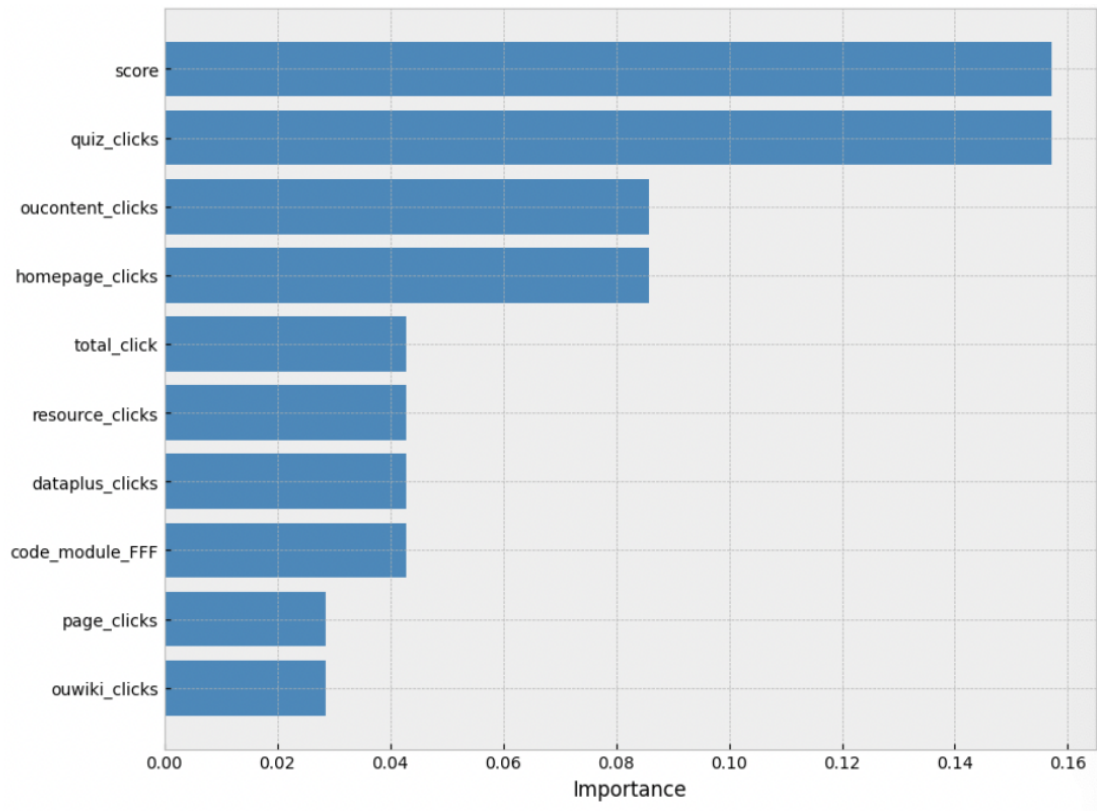


Figure 2.62: Most important features identified by the AdaBoost classifier.

After transforming the features and adding the Domain feature, a randomized grid search CV method was used to find the best parameters of the RandomForest algorithm such that the f1 score is maximized.

After finding the best parameters for the CART algorithm. The tree was built and then using post pruning techniques such as cost complexity pruning, the tree was pruned resulting in a simpler model which is easy to understand its decisions on passing or failing students.

The classification results of the CART algorithm before pruning and after are shown in Figure 2.59. The resulting tree after pruning is shown in Figure 2.63:

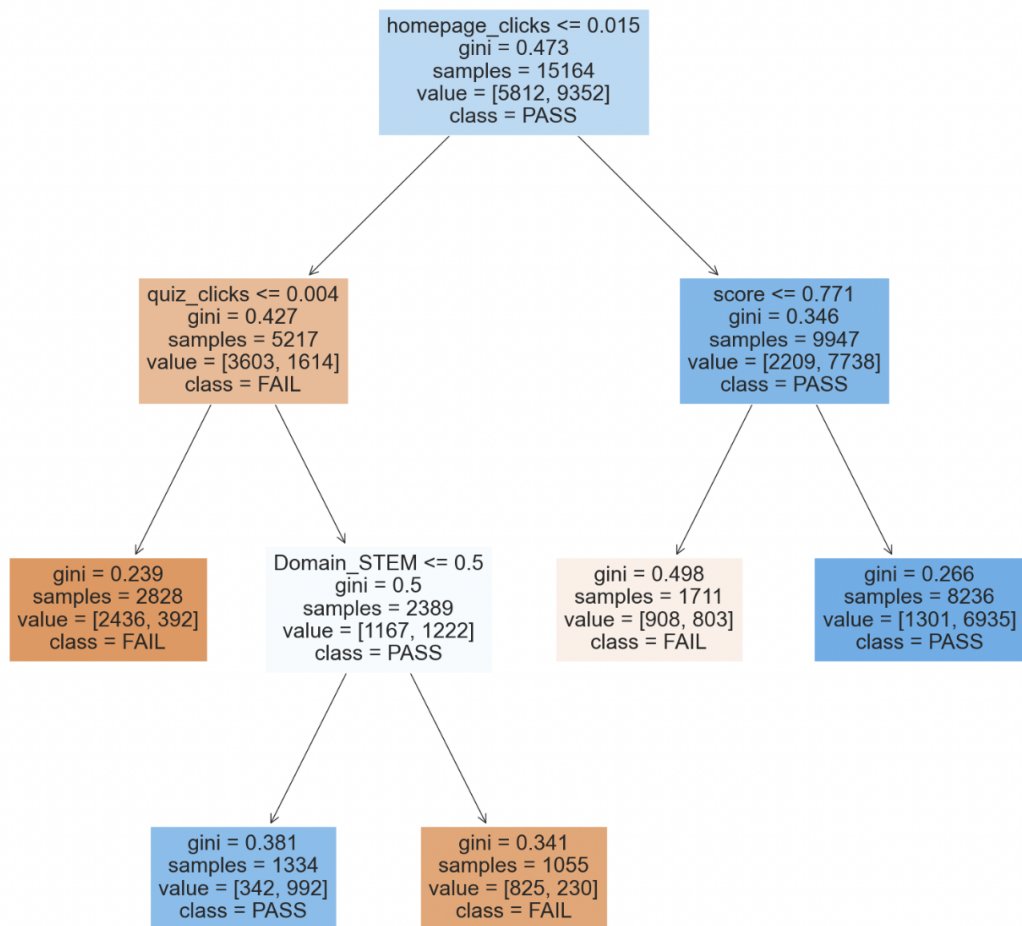


Figure 2.63: The resulting tree from the CART algorithm after pruning.

Interpretation of tree

From the tree, in Figure 2.63, we could extract rules, such as if the number of homepage clicks is less than 111 and the number of clicks on quiz-related material is less than 48, the student will fail. Similarly, we could observe how students will pass using this model. If the homepage clicks are less than 111, the quiz clicks are higher or equal to 49, and if the course is not of a STEM domain, then the student will pass; if the domain is STEM, the student will fail.

Here are all the decision rules from this tree:

```
|--- homepage_clicks <= 111.50
|   |--- quiz_clicks <= 48.50
|   |   |--- class: 0 (Fail)
|   |--- quiz_clicks > 48.50
|   |   |--- Domain_STEM <= 0.50
|   |   |   |--- class: 1 (Pass)
|   |   |--- Domain_STEM > 0.50
|   |   |   |--- class: 0 (Fail)
|--- homepage_clicks > 111.50
|   |--- score <= 65.26
|   |   |--- class: 0 (Fail)
|   |--- score > 65.26
|   |   |--- class: 1 (Fail)
```

Feature Importances: Based on this decision tree, we could observe that the most important features that the tree used to predict students' pass or fail were homepage clicks, quiz clicks, student assessment scores, and whether the course is a STEM course or a Social Sciences course.

2.3.7 Experiment 1 - C4.5

For C4.5 implementation, Weka was used as an open-source library. Documentation to Weka library: <https://fracpete.github.io/python-weka-wrapper3/> In this experiment, the preprocessing steps are the same as in the CART algorithm, where the numerical features have their original values, categorical features are encoded similarly to AdaBoost and Random Forest experiments, and the Domain feature was added for each course.

After transforming the features and adding the Domain feature, different confidence factors of post-pruning techniques were tested such that the C4.5 algorithm results in the highest f1 score value while keeping the tree small and interpretable.

The classification results of the C4.5 algorithm are shown in Figure 2.59. The resulting tree from the C4.5 implementation is shown in Figure 2.64.

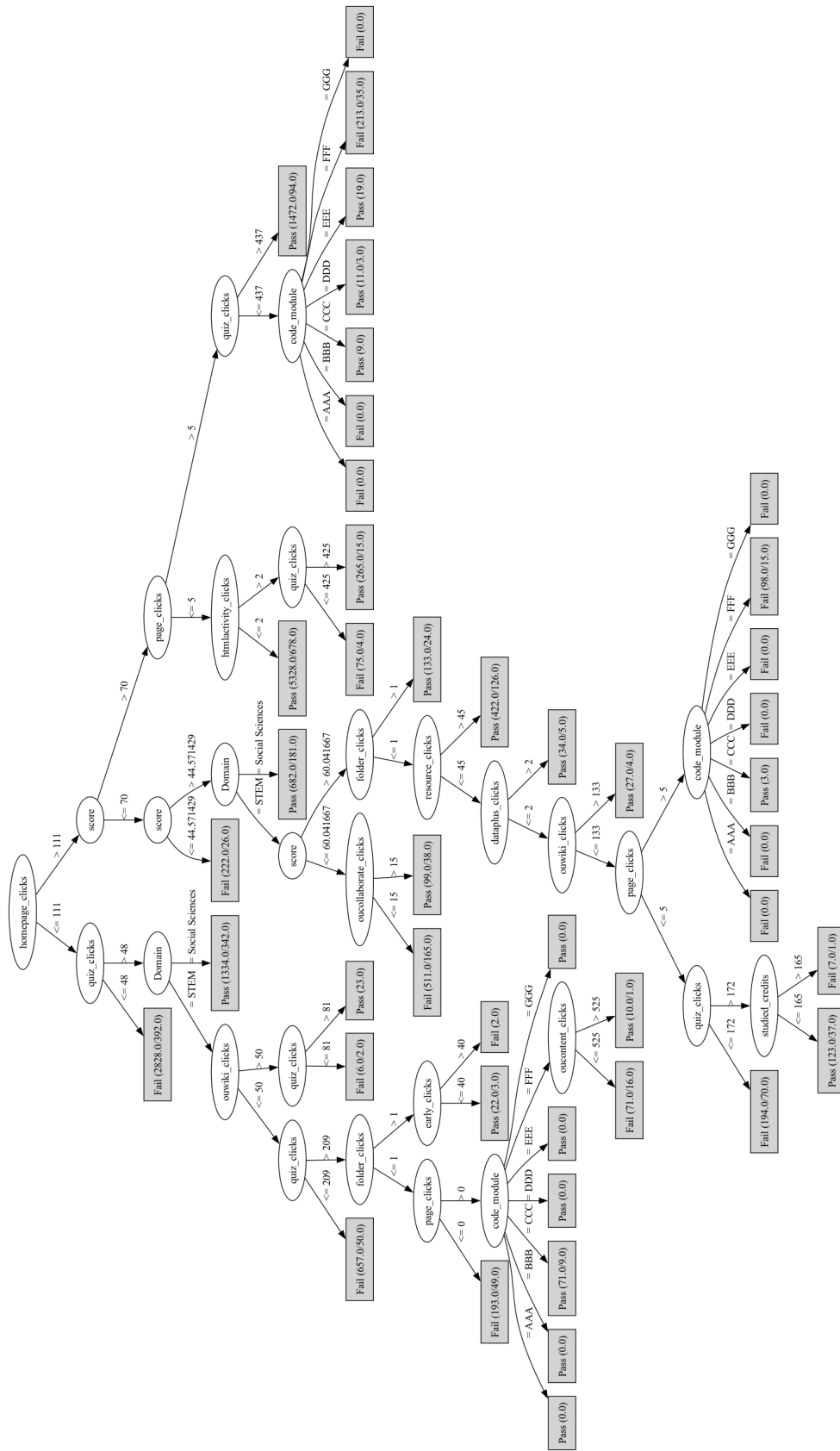


Figure 2.64: The resulting tree from the C4.5 implementation.

Interpretation of tree

From the tree in Figure 2.64, the root node is homepage clicks, and the split is similar to CART implementation at 111 clicks on the homepage. The tree splits into quiz clicks and scores as shown in Figure 2.64. Each leaf node has its label or the prediction assigned to each test sample if it were to be tested against each rule. In our example, it can be either Pass or Fail. The values in brackets, for example, the leaf node after the split on quiz clicks, indicate that 2828 samples from the training set are correctly classified as “Fail” in this case, while the other 392 samples are misclassified. These also help determine how accurate the prediction is at each leaf node.

Feature importances: Based on this decision tree, we can observe that the most important features that the tree uses to predict students’ pass or fail are similar to the CART algorithm: homepage clicks, quiz clicks, student assessment scores, and whether the course is a STEM course or a Social Sciences course.

Comparisons of results

From figure 2.59, we can see that Random Forests performs the best out of other models in predicting students’ pass or fail using the test data. Using the CART algorithm, the first algorithm that does not use post-pruning techniques performs slightly better than the algorithm that uses pruning. However, by pruning the tree, the number of nodes was reduced from 177 in the first CART model with no pruning to 9 nodes with the CART implementation using post-pruning. This decrease in model performance helps create more generalized, interpretable tree models.

Conclusions in experiment 1

The goal of the first experiment was to find out which of all 35 features, as shown in Table 2.1, are the most important in predicting students’ failure or success. Using different decision tree implementations, it was found that the most important features consistently in each implementation are Total clicks, Quiz clicks, scores on students’ assessments during the semester, and whether the domain is STEM or Social Sciences. If the students engage more with the learning environment, attend their quizzes given by the professors, and score high on assessments, they are more likely to pass. Also, students tend to succeed more in courses related to social sciences than in STEM.

Compared to other recent papers on the same topic, a recent paper, “Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models” [44], was used for comparison.

From the classification results, compared to this paper [44], which uses the same dataset, the results of the CART algorithm are similar, with an accuracy of 0.80 in this study and 0.83 in the paper.

In this study, the AdaBoost classifier has an f1-score of 0.84 and an accuracy of 0.84 while the study [44] that compared their result using the AdaBoost classifier has an accuracy of 0.79 and $f1 = 0.79$. This is an improvement in predicting the pass/fail of a student using this dataset.

Additionally, the tree was visualized in these experiments using implementations of CART and C4.5. In the visualization of a pruned tree, the decisions are visualized. They can be easily interpreted, which makes the full usage of the advantages of decision trees, which is the interpretability of a model.

2.3.8 Experiment 2 - Experimental results

In the second experiment, the goal was to use only demographic features such as: 'highest_education', 'imd_band', 'age_band', 'disability', 'final_result', and 'studied_credits'. The goal was to try and predict students' success in any of the seven courses available using only these features. The dataset is the same as in experiment one, the only difference is in features since only a subset of features is considered here.

Algorithms that were used for classification were CART, C4.5, Random Forests, and Adaboost. The results of the classification algorithms are presented in Figure 2.65:

Method	F1-Score	Precision	Recall	Accuracy	# Features
Ex.1 - Random Forest	0.58	0.59	0.58	0.58	6
Ex.1 - Ada Boost	0.59	0.59	0.58	0.58	6
Ex.1 - CART (No Pruning)	0.58	0.59	0.58	0.58	6
Ex.1 - CART (Post Pruning)	0.58	0.59	0.58	0.58	6
Ex.1 - C4.5	0.59	0.61	0.63	0.63	6

Figure 2.65: Results of the classification algorithms for demographic features.

The evaluation metrics show that the model did not perform well. The accuracy is 0.63, and the f1 score is 0.59.

Before training the model using the algorithms above, the features were transformed so that studied credits (numerical feature) were scaled using Standard Scaler, categorical features such as disability were transformed using one-hot encoding, and imd_band, highest education, and age_band were transformed using the ordinal encoding technique.

2.3.9 Experiment 2 - Random forests

Figure 2.65 shows the classification results on test data using Random Forests. Using the feature_importances method from skit-learn (more on: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html) after training the model, the most important features were extracted and shown in Figure 2.66:

Feature Importances: Based on random forest, the most important feature determining if students pass or fail their courses is their highest education (before registering).

2.3.10 Experiment 2 - AdaBoost

The classification results on test data by using AdaBoost are shown in Figure 2.65. Using the feature_importances the method, which is part of scit

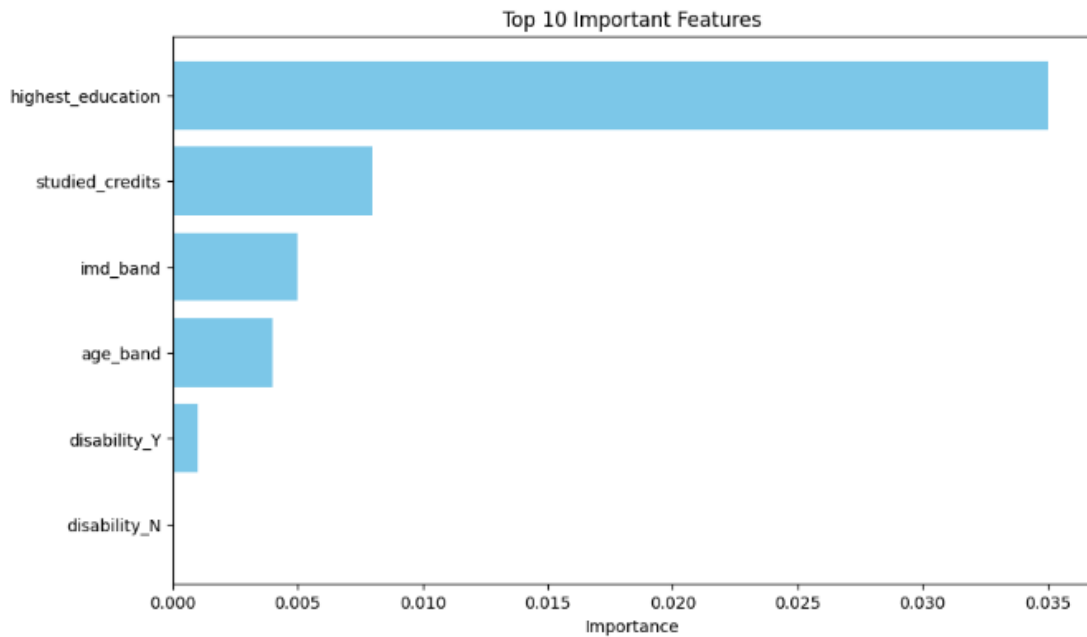


Figure 2.66: Random Forest feature importances for demographic data.

learn library, more info at https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html#sklearn.ensemble.AdaBoostClassifier.feature_importances. After training the model, the most important features were extracted and shown in Figure 2.67:

From Figure 2.67, we could observe that the most important feature extracted from the AdaBoost model was the highest education.

2.3.11 Experiment 2 - CART

The classification results on test data by using CART are shown in Figure 2.65.

After pruning the tree, in Figure 2.68, it is shown that the tree considers one feature as a root node. If a student's education is higher or equal to two (meaning the student is 'A Level or Equivalent' or higher), it passes; otherwise, it fails. The way that the encoding is done is that for all possible education levels in the highest education column (i.e. 'No Formal quals', 'Lower Than A Level', 'A Level or Equivalent', 'HE Qualification', 'Post Graduate Qualification') each of these leaves is transformed into numbers for example 'No Formal quals' maps to 0, Lower than A level maps to 1 and so on, this way each level has its number and they are ordinal, starting from lowest (no formal education) to the highest with post-graduate qualification.

2.3.12 Experiment 2 - C4.5

The classification results on test data by using C4.5 are shown in Figure 2.65. As we can see, the C4.5 best predicts students' success using only demographic features. Figure 2.69 shows the resulting tree after training the model.

As we can observe from the tree, the highest education is the most important

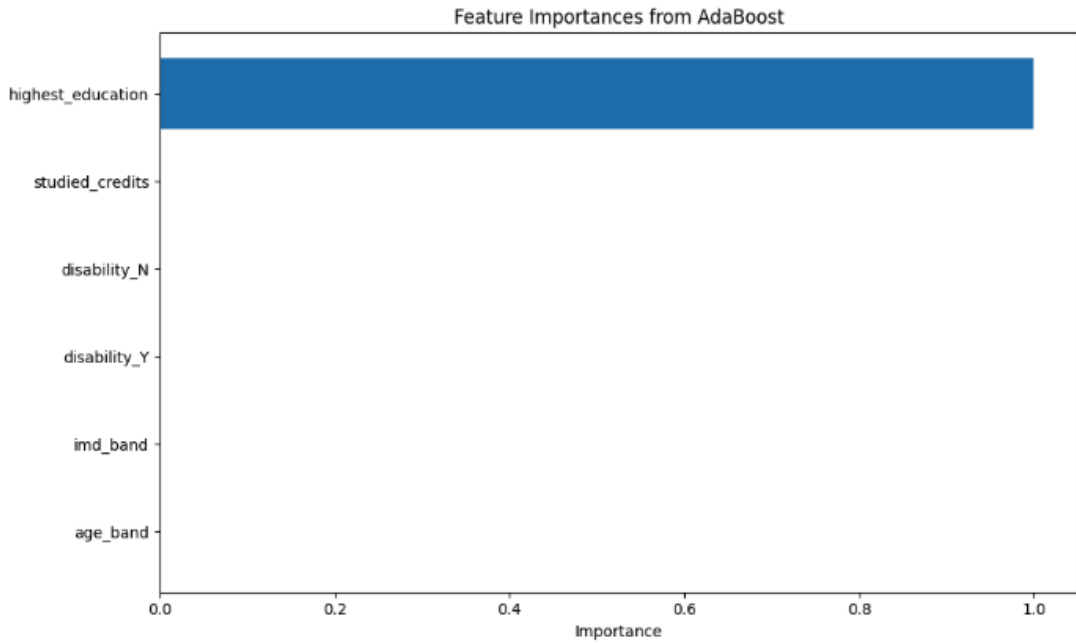


Figure 2.67: AdaBoost feature importances for demographic data.

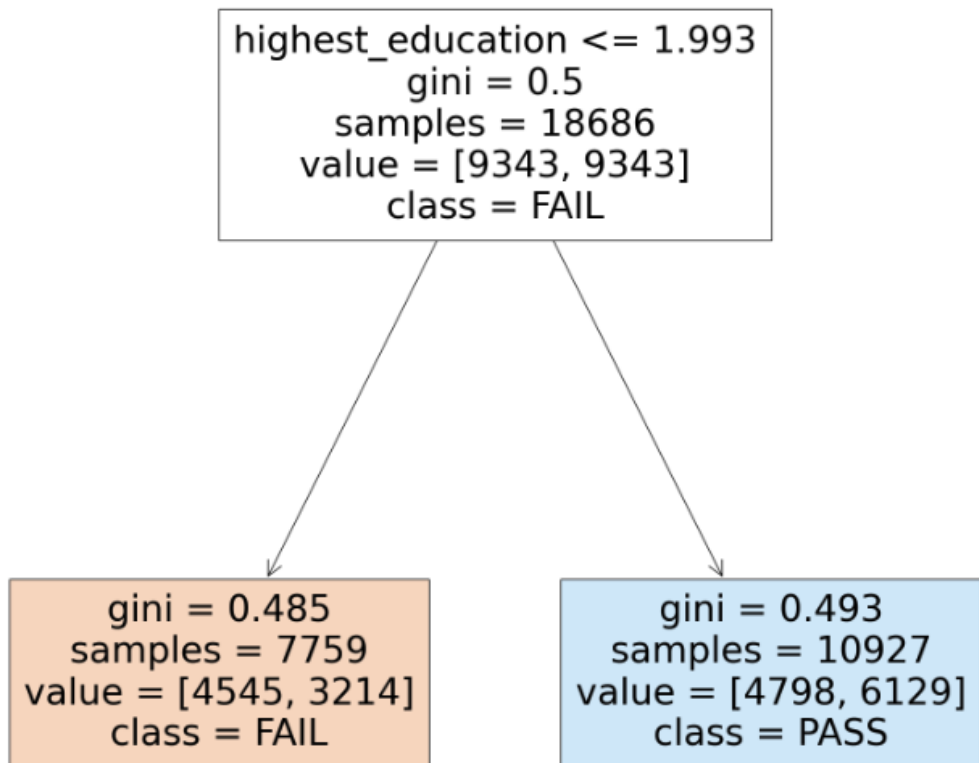


Figure 2.68: CART decision tree for demographic data.

feature, and the tree makes decisions for students' passing or failure. Other features are also used to make the right decision based on the tree in Figure 2.69.

Conclusions experiment 2

From the classification results in all algorithms in experiment two, we could conclude that it is not straightforward and simple to make decisions on student's pass or failure using only demographic features such as 'highest_education', 'imd_band', 'age_band', 'disability', 'final_result', 'studied_credits'. The C4.5 algorithm created a decision tree that can predict student success only using these features but with a low accuracy of 63% and an F1 score of 59%. It is still better than random classifiers but leaves room for improvement.

Additionally, the visualization of trees in C4.5 and CART shows how higher education plays an important role in students' success. In contrast, students with no prior/formal education have higher chances of failing their courses.

2.3.13 Experiment 3 - Experimental results

In this experiment, the goal was to create a model that would predict student's success or failure as early as possible, such that the professors or students can, in advance, see the possibility of failure and potentially improve. The course length is around 255 days. The model predicts student failure or success in the following days: 80, 120, 140, and 200. These days were chosen randomly to create four quarters and predict students' success/failure.

The table in Figure 2.70 shows the model's results for each quarter. Random Forest has the best results out of all the other implementations. As we can observe from the results, the more information about students during the semester, the better the model's performance will be.

The features that were used in this model are the same as the features used in the experiment one, shown in Table 2.1, the the difference is that the model was trained in a time series way where, depending on days, the number of clicks was reduced to the clicks up upon that day.

From the Random forest, the most important features to predict students' failure or success as early as possible are shown in Figure 2.71. As we can see, the mean score and whether the domain is STEM or Social Sciences are the most important features.

Random forest model: Out of 6499 students, 1941 were predicted to be failing (at risk). Out of these, 1941 were predicted as failing, and 1381 were correct. Using AdaBoost, 2072 were predicted to be failing (at risk) out of 6499 students. Out of these, 2072 was predicted as failing, and 1416 were correct. Using CART implementations, 2566 were predicted to be failing (at risk) out of 6499 students. Of these, 2566 were predicted as failing, and 1410 were correct. Using C4.5, out of 6499 students, 1941 were predicted to be failing (at risk). Of these, 1941 was predicted to fail, and 1381 were correct.

Compared to previous work [45] the results are similar, with both experiments having an accuracy of around 70% in the first quarter and around 80% in the last quarter using Random Forests.

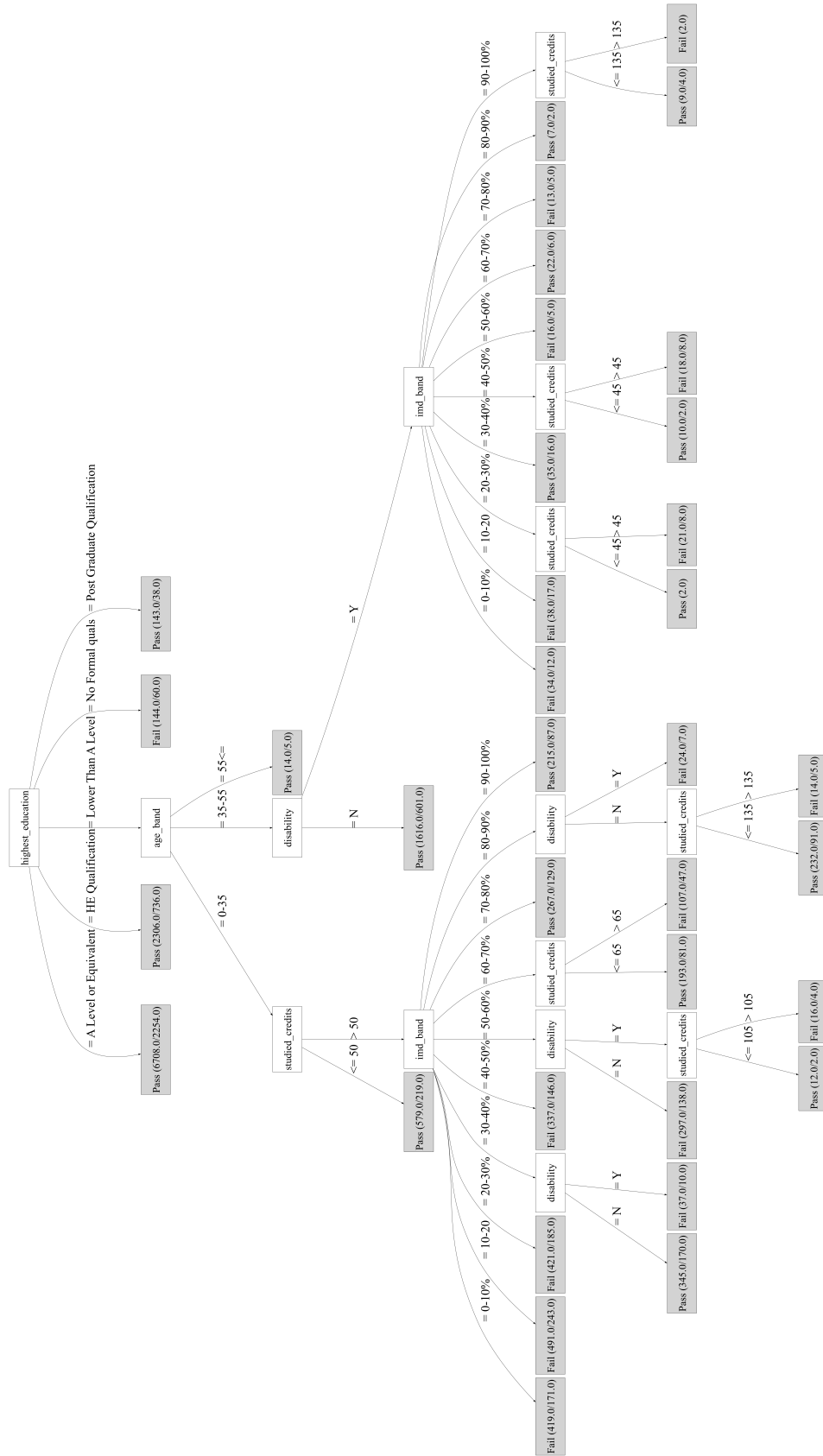


Figure 2.69: C4.5 decision tree for demographic data.

Method	Q1 - F1-score	Q1 - Accuracy	Q2 - F1-score	Q2 - Accuracy	Q3 - F1-score	Q3 - Accuracy	Q4 - F1-score	Q4 - Accuracy
Ex.3 - Random Forest	0.73	0.73	0.76	0.76	0.78	0.78	0.82	0.82
Ex.3 - AdaBoost	0.72	0.72	0.74	0.74	0.75	0.75	0.79	0.79
Ex.3 - CART (No Pruning)	0.65	0.65	0.68	0.68	0.69	0.69	0.75	0.75
Ex.3 - CART (Post Pruning)	0.67	0.67	0.68	0.68	0.67	0.67	0.75	0.74
Ex.3 - C4.5	0.68	0.68	0.71	0.72	0.73	0.73	0.78	0.78

Figure 2.70: model's predictions of student success/failure in quarters (time)

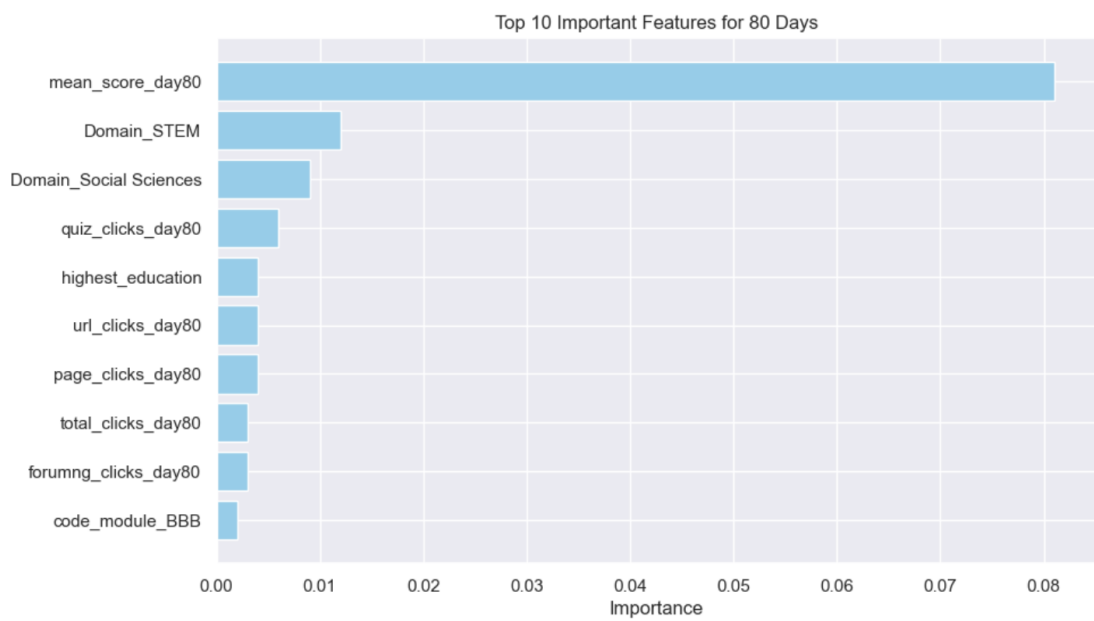


Figure 2.71: Time series analysis of Random Forest feature importances.

Conclusion

This thesis aimed to address important issues in the educational domain. Challenges such as racial disparities in graduation rates and low retention rates in universities remain crucial in education. This thesis addresses these by using data mining techniques. Additionally, social network methods were used to understand the importance and impact of education on the parliament.

Conclusions and results that address these challenges are presented in the following subsections.

2.4 US graduation rates analysis

The goal was to use clustering techniques to understand and research the graduation rates of various ethnic groups in four-year institutions.

Using clustering techniques, the institutions were grouped into six distinctive clusters. The main questions we wanted to answer from the cluster analysis:

1. Define the main characteristics of each cluster and find potential relationships between clusters.
2. Identify which universities changed the most over three years regarding all their features, such as Graduation Rates, Tuition Fees, etc.

Initially, the data of four-year educational institutions in the US were successfully scraped from various sources, resulting in around 4000 data points, which include graduation rates of seven ethnic groups, international students, the median household income of the county where the university is located, and the tuition fee.

Additionally, the created dataset can support further analysis and experimentation by other researchers.

2.4.1 Relationships and trends within the clusters

Using cluster analysis, we identified six distinctive clusters of institutions. High-performing institutions are grouped in cluster 4. A high student body and moderate graduation rate correspond to large universities in cluster 2. Low graduation rates, small number of students, low tuition fees, and median household areas include institutions in cluster 3 and clusters 1,5,6 with different performance over various groups with medium to low tuition fees and universities located in areas with varying median household income.

From cluster 1 (High Performing Institutions), the overall population of students is low, with a median male population of 320 with a range from 0 to 3016 men and female of 400 students ranging from 0 to 3514 female students. Tuition Fees for these institutions are the highest compared to all other clusters, with the median value of tuition fees of around 55.000 thousand dollars a year ranging from 0 to 66,490 thousand dollars a year. The median household income of the county where universities are located is high. The ethnic groups are diverse, with students across various ethnicities.

After extracting the highest-performing universities of each ethnic group from cluster one, we could observe the most popular and prestigious universities in the US belong to this cluster, with institutions such as Yale, Princeton, and Harvard consistently being the highest-performing institutions across all ethnic groups.

In cluster 5, institutions have lower tuition fees and the highest number of total students compared to cluster one, with a median of around 1500 men ranging from 273 to 6103 and 1800 women ranging from 259 to 5528 and a moderate performance in terms of graduation rates. These institutions are more affordable and have a large number of students. Institutions in cluster 2 have very low graduation rates. It has a small student population, and the median household income of the county where these universities are located is also the smallest. Clusters 3,4 and 6 are moderate to low-performing groups with different graduation rates across groups and various median household incomes and tuition fees. Some examples of institutions from each clusters are shown in Figures 2.25 and 2.26. After identifying these clusters as described above, the main characteristics found that universities with high tuition fees have a relationship with higher graduation rates and are located in counties with high median household income. Institutions with a large number of students have lower tuition fees, and their performance across all ethnic groups is moderate. A balanced number of students from each ethnic group in institutions might indicate better performance, as shown by the population of cluster one.

Universities with highest changes over three years

After training a SOM model, the model was used to observe the universities over three years. The universities with the highest changes in position on the map were considered for further analysis. To find the universities that changed the most across the years, the total Euclidean distance of each university's position on the SOM across the years was computed, where higher distances may indicate significant changes in universities.

From the top ten universities that were identified, the median household income of the counties (regions) where the universities were located, and tuition fees changed the most.

No direct relationship (dependency) was found between graduation rates and changes in tuition fees and the median household income of the counties (regions) where the universities are located across the years.

2.5 Impact of education on the parliament analysis

The second experiment aimed to use social network techniques to research the influence and importance of education on the structure of the parliament. The areas of interest were to find the most influential members in the parliament and find if the educational background can influence the political structure of the parliament.

Additionally, the results of these findings were compared between the two parliaments, focusing on the difference between a young democracy like Kosova

and an established parliament like the UK.

This analysis was conducted using 548 out of 650 members of the UK Parliament elected as of the 2019 constitution (ELECTION?). In recent elections held on Thursday, 4 July 2024, the structure of the parliament changed significantly. Labour Party won 411 seats, gaining 209 compared to 2019. The Conservative Party won 121 seats, losing 244 from their 2019 total of 365 seats. The Liberal Democrats increased the number of their seats by 61, reaching 72. The Scottish National Party won nine seats, down from 48 in 2019 [46].

Even though the analysis is based on the previous structure of parliament, using centrality measures as shown in the table in Figure 2.29, it was possible to identify the most important people in the parliament, where Keir Starmer became prime minister in the next parliament. Ed Davey, Leader of the Liberal Democrats, won 61 more seats compared to 2019. These important members are also visualized in the network graph in Figure 2.31. Aside from Matt Hancock and Dominic Raab, all the other key members shown in 2.29 were reelected in the 2024 elections.

From the graph in Figure 2.31 and by investigating each community using results in Figure 2.30, it was found Oxford and Cambridge Universities are the most frequently attended institutions among the members of parliament. The community results are shown in Figure 2.30, showing that Community One and Community 2 are created around prestigious universities such as Oxford and Cambridge. These communities are more traditional, and the average age in these communities is older compared to other communities. These two communities have the highest male member population, with the community one having more than twice as many males as females, with 70% male members.

Overall, the average ages across communities range from 50 to 56. Members in community four and community 7 are younger than those in other communities, with community 7 having the youngest average age of 50 with a 95% confidence interval of (45,55). Most of the communities have a narrow confidence interval, suggesting that the ages of the members of the parliaments are consistent and similar to each other. Community 5 has a wider confidence interval compared to other communities, suggesting that even though this has the highest average age compared to other communities, the ages vary and are more different from each other.

Community one has more than twice as many males as females. This community has the highest male percentage compared to other communities. Community two also has more male members, whereas male members comprise about two-thirds of the community, with 69% males and 31% females. Community 3 has a male majority, with 65% male members, slightly lower compared to communities one and two. Community 4 has a similar distribution as Community 3. Community 5 is more balanced than previous communities, with 61% Male and 39% Female members. Community 6 and 7 have close to even representations of male and female members, with Community 6 having 55% Male and 45% Female members, and Community 7 having 54% Male and 46% Female members.

The most gender equal communities are communities six and seven. There is a trend toward more equal representations of men and women in communities 3,4,5,6, whereas the first two communities have significantly higher ratios of male members. These communities (3,4,5 and 6) also have a higher ratio of members

from the Labour Party compared to the first two communities, with more than 30% of Labour Party members represented in Communities 3,4,5 and more than 45% of members in Community 6. This trend could indicate that female members prefer joining the Labour party. Also, the most equally represented communities (6,7) have the lowest ratio of Conservative Party members.

Community 7 has the highest number of Scottish National party members and the universities that the members attended are both in Scotland. This community's most common universities are the University of Glasgow and the University of Stirling.

In the second experiment of the thesis, the most important members of the parliament were effectively identified using social network techniques. Nevertheless, from the community detection results, the study did not show that a member of parliament's educational background affects their choice of political party or viewpoint. Although it might seem plausible that education could influence the member's political views, the obtained results did not support this hypothesis.

2.5.1 Parliament of Kosovo Analysis

For the parliament of Kosovo, the educational background of 81 out of the 120 members were considered for analysis. These members were elected in 2021. Five major communities were identified using Lovain algorithm for community detection. From the graph in Figure 2.34 and by investigating each community using results in Figure 2.35, it was found that the most common domain of studies of members of Parliament in Kosovo is Economy and Law.

The two most significant communities are Community One, where 10 out of 12 members studied economics, and the second biggest community, Community Three, where 10 out of 11 members went to LAW. The average age of communities ranges from 32 to 56. Community 4, with four members, has the youngest members with an average of 32 years old, and all four members (3 female and one male) finished a degree related to diplomacy. The oldest members are in community five, with five male members and one female member. They all went to medicine.

Community 1, 2, and 4 are more similar to each other, with a high ratio of members from the LVV party. The gender distribution in these communities is also closer to equal representation of male and female members.

The gender distribution in communities 3 and 5 is skewed towards more male members. Community 3 has ten males and one female, and Community 5 has five male members and one female member. Community 3 has more than 45 % of its members from the LDK party, and in Community 5, 50% of the members are from AAK. These two communities are more mixed in terms of having members from different parties, while communities 1, 2, and 4 are more representative of the LVV party.

The most influential members, as determined by centrality measures, are defined in the table in Figure 2.33.

Comparison of two parliaments

Communities 1, 2, and 4 in the Kosovo parliament are similar to most UK parliament communities (except community 7) in terms of the ratio of party mem-

bership members. However, communities 3 and 5 are different from the UK parliament communities. From the figures 2.34 and 2.31, we can observe that communities in the UK parliament are more interconnected, while communities in the Kosovo parliament are more isolated. This highlights the importance of educational ties, which may lead to better collaboration among parliament members.

The UK's parliament tends to have more older members. In the Kosovo parliament, the gender distribution in the first and second communities is notably balanced, providing a reassuring sign of societal progress. However, the other three communities show a skewed gender distribution, with either a male or female majority. The UK parliament and the remaining communities in Kosovo have more male members, except for community 4 in Kosovo, which shows a more balanced representation of male and female members.

Comparing the gender structure of communities in both the UK and Kosovo Parliament

The parliament of the UK, with its already established democracy, has 650 members of parliament, excluding the House of Lords. The government of Kosovo has 120 members of parliament. From the tables as shown in Figure 2.30 and 2.35, we could see the age distribution in both parliaments is similar, predominantly middle to older age, which might indicate that the members are experienced in their respective fields.

There are currently 238 female members in the House of Lords and 226 female Members of Parliament in the House of Commons. Women make up 29% of MPs in the House of Lords and 35% in the House of Commons. Overall, 32% of the members in both the Commons and Lords are female [47]. In the Kosovo parliament, as of the 2021 election, women constitute up to 36 percent of the members [48], which is similar to the House of Commons in the UK (35%). This indicates that both governments are progressing in gender equality in their political systems compared to the global average of women in parliaments, which is 27% [49] as of 2024. However, there is still room for improvement to achieve better and more equal representation of both genders.

2.6 Prediction of student success/failure in a virtual learning environment

This experiment aimed to use data mining techniques and analyze educational datasets available online. The dataset used is the Open University Learning Analytics Dataset (OULAD), which has a comprehensive set of features on student demographic data and click data in a virtual learning environment.

The focus of the analysis was divided into three parts. Part one aimed to use interpretable models like decision trees to predict students' success and failure. It was found that while there is lots of research using this dataset, the community lacks interpretability of the model, and we provided a final decision tree 2.63 in order to better understand the most important features that are contributing to the student's success.

Based on the decision tree implementation, the most important features that can be used to predict student success were clicks recorded on the homepage of the learning environment, the number of times students clicked (interacted) with the quizzes that were given in their virtual learning environment if the domain of the course was a STEM course or social sciences (students tend to succeed more in courses related to Social Sciences compared to the STEM domain), and their scores on their assessments during the semester. Random Forests performed the best with an accuracy 86% and an F1-score of 86%.

For comparison to other recent papers on the same topic, a recent paper, "Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models," [44], was used for comparison. From the classification results, compared to this paper[44], which uses the same dataset, the results of the CART algorithm are the same; both CART models have a similar accuracy of 0.83 in this study and 0.80 in the paper.

In this thesis, the AdaBoost classifier has an F1-score of 0.84 and an accuracy of 0.84, while the paper [44] the results using the AdaBoost classifier have an accuracy of 0.79 and $F1 = 0.79$. This shows an improvement in predicting the pass/fail of a student using this dataset compared to recent work.

The second part of the analysis focused on predicting students' success or failure by only using demographic features such as the deprivation index (IMD), which is a measure of deprivation; for more info, visit this link <https://www.gov.uk/government/statistics/english-indices-ofdeprivation-2019>, age, disability, studied credits and highest education. The C4.5 algorithm performed best, with an F1-score of 59% and an accuracy of 63% on both train and test data. From the decision tree 2.68, we could see that the most important feature was the highest education of students, indicating the importance of student's education level in passing or failing the courses. The experiment's accuracy and F1 scores were low when only demographic features were used to predict students' success or failure. This indicates that we cannot accurately predict students' success or failure using only demographic features.

In the third part of this experiment, the goal was to create a model that would predict students' success or failure as early as possible so that the professors or students could see the possibility of failure and potentially improve. Random Forest had the best results out of all the other implementations. Compared to previous work [45], the results are similar, with both this experiment having an accuracy of around 70% in the first quarter and an accuracy of around 80% in the last quarter using Random Forests.

2.7 Future work

In social network analysis, graph neural networks could have been used as well. They might improve the analysis by providing insight into the existing network structure. Other governments in the Balkan region can be used to compare with Kosovo's government, focusing on their similarities and differences.

In analyzing graduation rates in the US, other visualization techniques, such as tSNE, can be beneficial in considering other visualization techniques that visualize high-dimensional data in 2D space. In analyzing graduation rates in the US, other countries that provide access to graduation rates of different ethnicities can be

scraped and provide a comparison with the existing analysis.

To analyze the alliances in parliament, LLM can be used to find relationships between members of parliament, which can be used to create weighted graphs based on these interactions.

The thesis results suggest that diverse institutions with various ethnic groups and higher tuition fees tend to succeed more. These results can help improve universities in Kosova by supporting diversity.

Bibliography

- [1] David J. Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. MIT Press, Cambridge, Massachusetts, 2001.
- [2] D. Shapiro, A. Dundar, F. Huie, P. Wakhungu, X. Yuan, A. Nathan, and A. Hwang, Y. Completing college: A national view of student attainment rates by race and ethnicity – fall 2010 cohort (signature report no. 12b). Technical report, National Student Clearinghouse Research Center, Herndon, VA, April 2017.
- [3] John J. Kirlin. What government must do well: Creating value for society. *Journal of Public Administration Research and Theory*, 6(1):161–185, 01 1996.
- [4] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.
- [5] Lior Rokach and Oded Maimon. *Data Mining With Decision Trees: Theory and Applications*. World Scientific Publishing Co., Inc., Singapore, 2014.
- [6] A.C. Müller and S. Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O’Reilly Media, Incorporated, 2018.
- [7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.
- [8] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.
- [9] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer, USA, 2015.
- [10] Bernard S Duran and Patricia L Odell. *Cluster Analysis: A Survey*, volume 100. Springer Science & Business Media, 2013.
- [11] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [12] Leonard Kaufmann and Peter Rousseeuw. Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*, pages 405–416, 01 1987.
- [13] Peter Rousseeuw. Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *comput. appl. math.* 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987.
- [14] Stephen P. Borgatti, Martin G. Everett, and Jeffrey C. Johnson. *Analyzing Social Networks*. SAGE Publications Ltd, London, 2013.
- [15] Francis Bloch, Matthew O. Jackson, and Pietro Tebaldi. Centrality measures in networks. 2023.

- [16] Lei Tang and Huan Liu. *Community Detection and Mining in Social Media*. Morgan & Claypool, 2010.
- [17] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.
- [18] Charu C. Aggarwal, editor. *Social Network Data Analytics*. Springer, New York, 2011.
- [19] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 2008.
- [20] Soteris Kalogirou and Adel Mellit. *Handbook of Artificial Intelligence Techniques in Photovoltaic Systems*. Academic Press, London, 2022.
- [21] K. Jearanaitanakij. Classifying continuous data set by id3 algorithm. In *2005 5th International Conference on Information Communications & Signal Processing*. IEEE, 2005.
- [22] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Taylor & Francis, 1984.
- [23] Charu C. Aggarwal, editor. *Data Classification: Algorithms and Applications*. Chapman and Hall/CRC, Boca Raton, 2014.
- [24] Xindong Wu and Vipin Kumar. *The Top Ten Algorithms in Data Mining*. CRC Press, United States, 2009.
- [25] Vf-cart: A communication-efficient vertical federated framework for the cart algorithm. *Journal of King Saud University - Computer and Information Sciences*, 35(1):237–249, 2023.
- [26] Lior Rokach and Oded Maimon. *Data Mining with Decision Trees: Theory and Applications*. World Scientific Publishing Co., Inc, River Edge, NJ, USA, 2008.
- [27] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [28] Lukasz Gadomer and Zenon Sosnowski. Pruning trees in c-fuzzy random forest. *Soft Computing*, 25:1–19, 02 2021.
- [29] UniPage. Education system in the uk. https://www.unipPage.net/en/education_system_uk, 2024. Accessed: 2024-06-02.
- [30] UK Government. Education system in the uk, 2012. Accessed: 2024-06-02.
- [31] U.S. Department of Education, International Affairs Staff. Education in the united states: A brief overview. U.S. Department of Education, Washington, D.C., 2005. First published September 2003. Revised September 2005.
- [32] Parliament and the crown. <https://www.britannica.com/event/Act-of-Union-Great-Britain-1707>. Accessed: 2024-04-30.

- [33] Kings and queens of britain. <https://www.historic-uk.com/HistoryUK/KingsQueensofBritain/>. Accessed: 2024-04-30.
- [34] Parliament and the crown. <https://www.parliament.uk/about/how/role/rerelations-with-other-institutions/parliament-crown/>. Accessed: 2024-04-30.
- [35] Members of the uk parliament. <https://members.parliament.uk/>. Accessed: 2024-04-30.
- [36] Parliament and government. <https://www.parliament.uk/about/how/role/rerelations-with-other-institutions/parliament-government/>. Accessed: 2024-04-30.
- [37] Parliament acts. <https://www.parliament.uk/about/how/laws/parliamentacts/>. Accessed: 2024-04-30.
- [38] Devolved parliaments and assemblies. <https://www.parliament.uk/about/how/role/rerelations-with-other-institutions/devolved/>. Accessed: 2024-04-30.
- [39] Kosovo government and society. <https://www.britannica.com/place/Kosovo/Government-and-society>. Accessed: 2024-04-30.
- [40] Role and competencies of the assembly. <https://www.kuvendikosoves.org/eng/about-the-assembly/role-and-competencies-of-the-assembly/>. Accessed: 2024-04-30.
- [41] Jakub Kuzilek, Martin Hlosta, and Zdenek Zdrazil. Open university learning analytics dataset. *Scientific Data*, 4:170171, nov 2017.
- [42] Cristian Mihaescu and Paul Popescu. Review on publicly available datasets for educational data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11, 02 2021.
- [43] Early prediction of learners at risk in self-paced education: A neural network approach. Scientific Figure on ResearchGate. Accessed: 2024-07-05.
- [44] Muhammad et al. Adnan. Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models. 2022.
- [45] Ali Al-Zawqari, Dries Peumans, and Gerd Vandersteen. A flexible feature selection approach for predicting students' academic performance in online courses. *Computers and Education: Artificial Intelligence*, 3:100103, 2022.
- [46] General election results. Commons Library Research. Accessed: 2024-07-05.
- [47] Membership of the uk parliament. <https://commonslibrary.parliament.uk/research-briefings/sn01250/>. Accessed: 2024-04-30.
- [48] Kosovo government and society. https://pdf.usaid.gov/pdf_docs/PA00ZF2R.pdf. Accessed: 2024-04-30.
- [49] Women in national parliaments. https://data.ipu.org/women-averages/?date_year=2024&date_month=04. Accessed: 2024-04-30.

List of Figures

1.1	Decision tree classifier for loan applications.	18
1.2	Dataset for decision tree example	24
1.3	Decision tree created by ID3 algorithm using a simple dataset.	28
2.1	Flowchart of the scraping process	39
2.2	Code Snippet for scraping graduation rates, main method	40
2.3	Code Snippet for scraping graduation rates, internal logic	40
2.4	Code Snippet for scraping tuition fees	41
2.5	Code Snippet for scraping latitude and longitude	42
2.6	Code Snippet for scraping county name and county code	42
2.7	Code Snippet for scraping Median Household Income	43
2.8	Code Snippet for merging genders into total students	44
2.9	Tuition Fees distribution of all 4-year institutions	45
2.10	Students distribution of all 4-year institutions	46
2.11	Gender distribution of 4-year institutions	47
2.12	Tuition Fees distribution of 4-year institutions across three years	48
2.13	Graduation Rates of each ethnic group in 4-year institutions across all years	49
2.14	K-means clustering on SOM	51
2.15	Silhouette Score Analysis	52
2.16	Population of students, tuition fees and median household income across K means clusters	53
2.17	Graduation Rates of ethnic groups across K means clusters	55
2.18	Graduation Rates of ethnic groups across K means clusters	58
2.19	Top 10 universities with the highest graduation rates for U.S. non-resident students.	61
2.20	Top 10 universities with the highest graduation rates for students of two or more races.	61
2.21	Top 10 universities with the highest graduation rates for Black or African American students.	61
2.22	Top 10 universities with the highest graduation rates for Hispanic students.	62
2.23	Top 10 universities with the highest graduation rates for Asian students.	62
2.24	Top 10 universities with the highest graduation rates for White students.	62
2.25	Randomly selected institutions from cluster 1,2 and 3	63
2.26	Randomly selected institutions from cluster 3,4 and 6	64
2.27	The institutions with the most changes and the variations in tuition fees and median household income over three years.	65
2.28	MP Names and Associated Universities	68
2.29	Centralities in the UK Parliament	70
2.30	Community detection results and analysis of the UK Parliament	71
2.31	Communities in the UK Parliament	72
2.32	The members of parliament in Kosovo	73

2.33	Centrality measures for members of parliament in Kosovo	74
2.34	Network graph for parliament of Kosovo	75
2.35	The community detection results for Kosovo parliament	76
2.36	Schema of the dataset	77
2.37	Module/Course structure over time	78
2.38	Student's courses table	78
2.39	Student's assessment types and weights	79
2.40	Table of materials available in the VLE	79
2.41	Student information table	80
2.42	Student's registration table	80
2.43	Student assessments table	81
2.44	Students and their total number of clicks in specific materials in the VLE	81
2.45	Data Science Lifecycle	82
2.46	Comparison of Age Groups and their final results	83
2.47	Comparison of student's gender and their final results	84
2.48	Comparison of credits enrolled and their final results	85
2.49	Comparison of students IMD band and their final results	85
2.50	Comparison of each student's highest education level group and their final results	86
2.51	Comparison of student's total clicks and their final results	86
2.52	Student's clicks in different age groups	87
2.53	Total student's clicks and their average score on assessments	88
2.54	Correlation Matrix of numerical features	88
2.55	Code on imputing missing values on 'IMD BAND' column	90
2.56	Code on removing wrongly labeled students	91
2.57	Code on fixing wrongly labeled weights of assessments	91
2.58	Distribution of the 'Final Result' feature.	93
2.59	Final results of each classification algorithm on the test data.	93
2.60	Code modification for adding the 'Domain' feature.	94
2.61	Most important features identified by the random forest classifier.	95
2.62	Most important features identified by the AdaBoost classifier.	96
2.63	The resulting tree from the CART algorithm after pruning.	97
2.64	The resulting tree from the C4.5 implementation.	99
2.65	Results of the classification algorithms for demographic features.	101
2.66	Random Forest feature importances for demographic data.	102
2.67	AdaBoost feature importances for demographic data.	103
2.68	CART decision tree for demographic data.	103
2.69	C4.5 decision tree for demographic data.	105
2.70	model's predictions of student success/failure in quarters (time)	106
2.71	Time series analysis of Random Forest feature importances.	107

List of Tables

2.1 List of Features	92
--------------------------------	----

List of Abbreviations

A. Attachments

A.1 User Documentation

This user documentation describes how to attain the experimental results of this thesis. It is divided into three parts, each guiding through commands for running the programs.

The first part describes the programs that produce the results of experiments related to U.S. graduation rates. The second part explains the experiments on predicting student success or failure. The third part briefly describes how to run Jupyter notebooks for social network experiments.

A.1.1 Graduation Rates in U.S. Experiments

The following command generates results of the experiments related to graduation rates in the U.S.

```
python som_experiments.py --input df_combined.parquet
--som_size 35 --sigma 8.676 --learning_rate 0.719
--iterations 500000 --n_clusters 6
```

This command trains the SOM model and generates the parquet file with cluster assignments on the model weight vectors. It will output the following files:

- `som_plot.png`: the model's U-matrix
- `som.p`: SOM model with trained weights
- `df_final_with_clusters.parquet`: contains universities, graduation rates, and cluster assignments for each university. This data frame was used to analyze and visualize the clusters.

For detecting the universities with the highest changes across the years, the following command can be used:

```
python3 uni_movements.py som.p df_combined.parquet
```

This command reads the som model, and the input of final dataframe that is used for clustering. It outputs the universities with the highest cumulative euclidean distance across the years. The output of this dataframe was manually analyzed to find write the results in the thesis.

A.1.2 Predicting Student Performance Experiments

This part of the thesis has three experiments. The commands below can be used to reproduce the results in the thesis.

First Experiment

```
python3 classification_trees.py
```

This experiment uses the corresponding `config.txt` file. It returns models' accuracy, precision, recall, f1-score, and pruned tree images for understanding the most important features.

Second Experiment

```
python3 classification_demographics_trees.py
```

This experiment uses only demographic features, and the config file is `config-demographics.txt`. It can be run the same as the First Experiment, and it also returns the same metrics.

Third Experiment

```
python3 classification_intime.py
```

This command runs the third experiment and provides the precision, recall, accuracy, and f1-score.

A.1.3 Social Networks Analysis

To obtain the results from this part of the experiment, the Jupyter notebooks located in the `Social Networks Analysis` folder must be run.

- `socialnetworks-uk-analysis.ipynb`
- `Social Networks - KS - Final.ipynb`

`Social Networks - UK - Final - Scraping.ipynb` is used to scrape the data that is used in the analysis part (in the notebook `socialnetworks-uk-analysis.ipynb`). Each cell should be run in order in the notebook.

A.1.4 Additional Modifications

Graduation Rates experiments, creating a SOM model, with different parameters

The following scripts can be run independently to use different parameters for the model or to find the best parameters.

find_parameters.py This script is used to find the best parameters for the SOM model. You provide the input `df_combined.parquet` as arguments, and the space (or possible values) of the learning rate and sigma you want to find. The results are print statements of the algorithm performance based on error metrics.

```
python find_parameters.py
--input df_combined.parquet
--som_size 35
--iterations 500000
--max_evals 100
--min_sigma 1
--max_sigma 10
--min_learning_rate 0.0001
--max_learning_rate 5
```

train_som.py This script takes `df_combined.parquet` as input and the parameters for the model. It outputs the `som.p` model and an image of the SOM grid.

```
python train_som.py
--input df_combined.parquet
--som_size 35 --sigma 8.676
--learning_rate 0.719
--iterations 500000
```

cluster_asg.py This script inputs the `som.p` model and outputs a data frame with all universities clustered using the K-means algorithm and the silhouette score.

```
python cluster_asg.py
--input df_combined.parquet
--som_model som.p
--n_clusters 6
```

Scraping the Data

The scripts that scrape the tuition fees, graduation rates, county information, and median household income are in a folder called `scraping`.

Scraping Graduation Rates To scrape graduation rates for each year (2020, 2021, 2022), the following scripts in the `Scraping` folder are used:

- `Grad_rates_2020.py`
- `Grad_rates_2021.py`
- `Grad_rates_2022.py`

The scripts can be run independently, and the only input they need is the `Workbookv7-Sheet1.csv` Excel sheet, which is part of the `Scraping` folder.

Given that the `Workbookv7-Sheet1.csv` is in the same folder with the `grad_rates` Python scripts, by using the command `python3 grad_rates_2020.py`, the script starts executing and scraping the data. After the script finishes running, a parquet file called `grad_rates_2020.parquet` is generated in the same folder. The same procedure for each `grad_rates_{year}` file can be followed.

To scrape graduation rates for other years, the method `scrape_and_preprocess_grad_rates` must be modified based on different HTML structures of the page <https://nces.ed.gov/ipeds/datacenter/FacsimileView.aspx?surveyNumber=8&unitId=110635&year={year}>.

Scraping Tuition Fees The following scripts are used to scrape tuition fees:

- `Tuition_fees_2020.py`
- `Tuition_fees_2021.py`
- `Tuition_fees_2022.py`

The scripts can be run independently, and the only input they need is the `Workbookv7-Sheet1.xlsx` Excel sheet, which is part of the **Scraping** folder. The scripts can be executed similarly to `grad_rates` by using the `tuition_fees_2020.py` command. As a result, a parquet file called `tuition_fees_2020.parquet` is generated in the same folder.

To scrape tuition fees for other years, the method `scrape_and_preprocess_tuition_fees` must be modified based on different HTML structures of the page https://nces.ed.gov/ipeds/datacenter/FacsimileView.aspx?surveyNumber=1&unitId={ipeds_value}&year={year}. The `ipeds_values` can be found in the `Workbookv7-Sheet1.csv` sheet.

Scraping Median Household Income To scrape median household income information, the following scripts must be run in the exact order:

- `Geo_scrape.py` - Its input is `Workbookv7-Sheet1.csv` (meaning they have to be in the same folder), and it outputs a parquet file: `df_lat_lon_final.parquet`.
- `County_scrape.py` - Its input is the `df_lat_lon_final.parquet`. By running `python3 county_scrape.py` the output is `df_lat_lon_county.parquet`, which is saved under the folder `test_data`.

To scrape the median household income for each year, the following scripts are used:

- `County_2020.py`
- `County_2021.py`
- `County_2022.py`

The `County_{year}.py` scripts take as input `df_lat_lon_county.parquet` and can be run independently.

Their outputs are `median_household_income_{year}.parquet`. Merging the `tuition_fees_{year}.parquet`, `grad_rates_{year}.parquet`, and `median_household_income_{year}.parquet` is done in the script called `process_merge.py`, which is located in the folder `preprocess_and_merge`.

The script can be run using this command:

```
python process_merge.py
--income_2020 median_household_income_2020.parquet
--income_2021 median_household_income_2021.parquet
--income_2022 median_household_income_2022.parquet
--grad_rates_2020 grad_rates_2020.parquet
--grad_rates_2021 grad_rates_2021.parquet
--grad_rates_2022 grad_rates_2022.parquet
--tuition_fees_2020 tuition_fees_2020.parquet
--tuition_fees_2021 tuition_fees_2021.parquet
--tuition_fees_2022 tuition_fees_2022.parquet
```

This specifies the path of each parquet generated from the scraping part. The resulting parquet (data frame) is called `df_combined.parquet`. This file is used as input in the Cluster Analysis part of the experiments.

Decision trees for loan application

The `LoanApplicationDecisionTree.ipynb` notebook was used to generate the decision tree for the loan application example in the chapter where decision trees are explained. The dataset used in this notebook can be found online on Kaggle: <https://www.kaggle.com/datasets/ninzaami/loan-predication>, and in the local folder under the name `LoanApprovalPrediction.csv`.

The `LoanApplicationDecisionTree.ipynb` notebook was used to generate the decision tree for the loan application example in the chapter where decision trees are explained. The dataset used in this notebook can be found online on Kaggle: <https://www.kaggle.com/datasets/ninzaami/loan-predication>, and in the local folder under the name `LoanApprovalPrediction.csv`.

Data preprocess and cleaning

The `data_preprocessing_oulad.ipynb` notebook was used to clean and preprocess the data before using it for predictive analysis.