Software Technology 2024

Γαβριήλ Κοσκινάς[1], Μιχαήλ Αβαγιανός[2], Θανάσης Ράτσικας[3]

[1]Π2019211, [2]inf2021009, [3]inf2021193

Abstract

This document introduces the software implementation and architecture, the dataset that was used, the analysis, the results, and the conclusion of the aforementioned analysis. Furthermore, it explains how the data analysis was done, gives the results as well as figures for the algorithms' graphs, and concludes the analysis by referring to the input and output of each type of algorithm and the results of each algorithm. It also mentions what a software lifecycle is, and suggests the use of Agile as a methodology for software development. Moreover, it details how the software will be released to the public, how each phase will need to be adjusted for such release. Finally, it lists the contributions of each team member.

General Information

- Language: Python 3.11
- Libraries: pandas, numpy, sklearn, scipy, imblearn, matplotlib, streamlit, eaborn, pyreverse, tabulate
- Git Repository: https://github.com/maikkundev/soft-tech-project-2024
- Software Report: This document, using Typst
- Typst File Path: `<PROJECT_ROOT>/docs/report.typ`

Implementation

We used the Streamlit library to create the user interface, and the matplotlib and seaborn libraries to create the data visualization plots. The machine learning algorithms were implemented using the sklearn library, and the data was preprocessed using the pandas and numpy libraries. The software was containerized using Docker, and the UML diagrams were created using Pyreverse and Draw.io.

UML Diagrams

The UML diagrams that were created for the software show the architecture of the software, both the function and file structure respectively, and the user interaction with the software.

Architecture

The figures below show the file structure and the functions architecture of the software. Pyreverse was used to generate the UML diagrams.

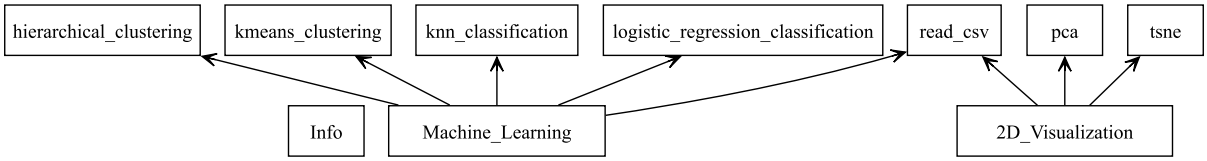

Figure 1: File Structure UML Diagram

The diagram in Figure 1 shows the file structure of the software, where there are three pages: **Info**, **Machine Learning**, and **2D Visualization**. Each page contains the files that are used for the respective tasks.

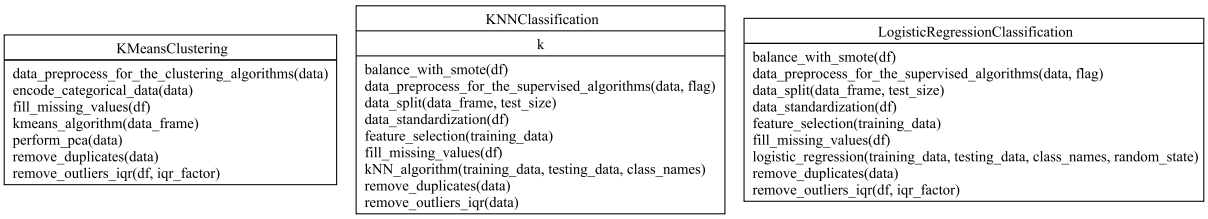

Figure 2: Functions Architecture UML Diagram (1)

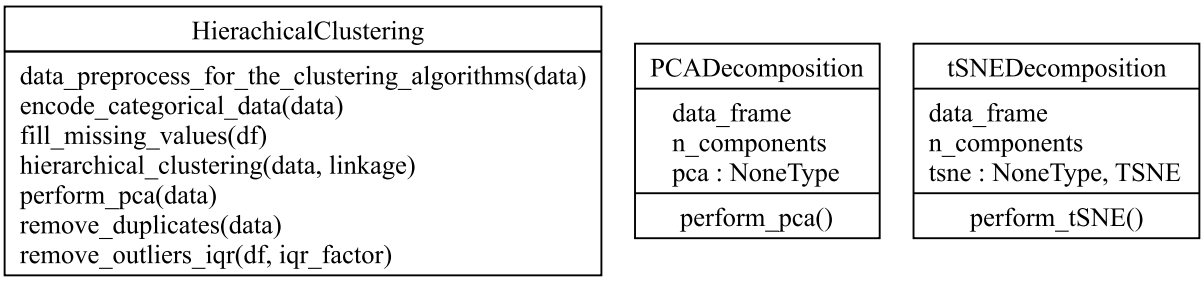

Figure 3: Functions Architecture UML Diagram (2)

The diagrams in Figure 2 and Figure 3 show all 6 classes that were used, their constructors, and the functions that were implemented in each class. The classes are **KMeansClustering**, **KNNClassification**, **LogisticRegressionClassification**, **HierarchicalClustering**, **PCADecomposition**, and **tSNEDecomposition**.

User Interaction

The figure below shows the user interaction with the software. Draw.io was used to create the UML diagram.

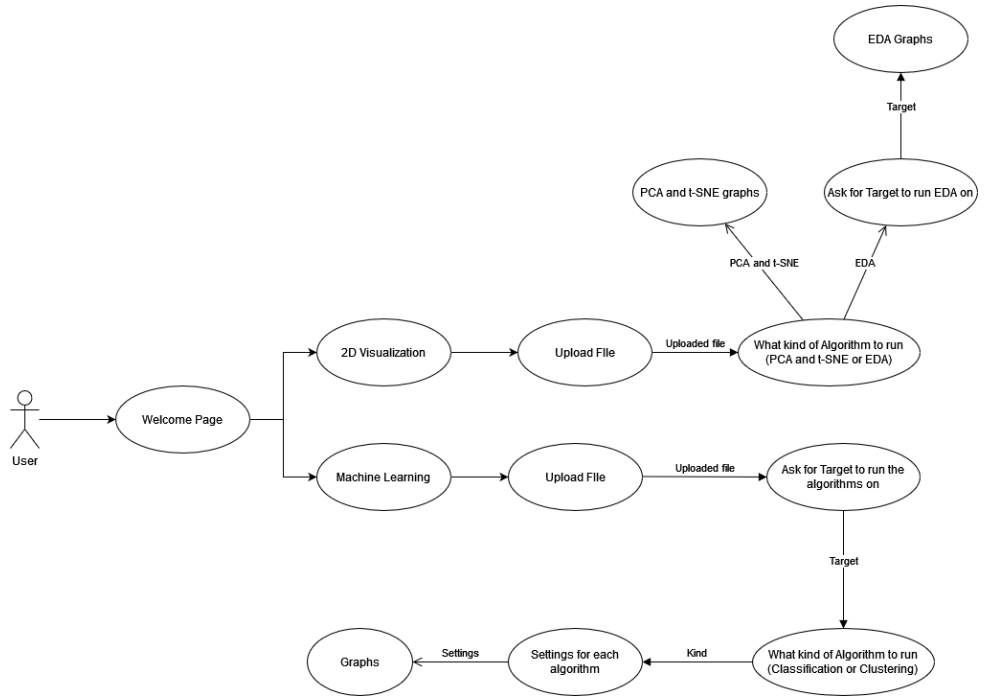

Figure 4: User Interaction UML Diagram

As seen in Figure 4, the user can interact with the software by inputting the data, and then the software will process the data and display the results.

Data

The dataset we used is from the UC Invine Machine Learning Repository, and allows us to be able to Predict Students' Dropout and Academic Success[1].

Analysis

The analysis of the dataset was done using both classification and clustering algorithms. The classification algorithms used were Logistic Regression and KNN Classification, while the clustering algorithms used were KMeans Clustering and Hierarchical Clustering.

Classification

First the data was divided into a training set, which contained 80% of the data, and a test set, which contained the remaining 20% of the data. Then, to prepare the data for analysis and modeling, the following steps took place in the following order:

1. Handling missing values: a check for missing values was performed and replaced with the mean of each attribute.
2. Removal of duplicates: the data was checked for duplicate records and any existing records were removed.
3. Categorical data coding: a check was made for variables with categorical data and they were converted into numerical values. This is necessary for machine learning algorithms that work with numeric data.
4. Removing outliers: extreme values, which could negatively affect the analysis, were identified and removed. This helped to improve the stability of the models and avoid misinterpretations.
5. Standardization: the data were standardized (standard scaling) to have a mean of 0 and a standard deviation of 1. This brought all features to the same scale, improving the performance of machine learning algorithms.
6. Feature selection: filter based feature selection based on f-statistic was applied, where the 10 most relevant and important features were selected. This helped to reduce the data dimension and improve the performance of the models.
7. SMOTE[2] (Synthetic Minority Over-sampling Technique): SMOTE technique was used for over-sampling the minority category, in order to achieve better training of machine learning algorithms.

Clustering

The data preprocessing methodology for clustering was similar to the one for classification with some minor differences. More specifically, to prepare the data for analysis and modelling, the following steps took place in the following order:

1. Handling missing values: a check for missing values was performed and replaced with the mean of each attribute.
2. Removal of duplicates: the data was checked for duplicate records and any existing records were removed.
3. Categorical Data Coding: a check for variables with categorical data was performed and converted to numeric values.

[1] M.V.Martins, D. Tolledo, J. Machado, L. M.T. Baptista, V.Realinho. (2021) "Early prediction of student's performance in higher education: a case study" Trends and Applications in Information Systems and Technologies, vol.1, in Advances in Intelligent Systems and Computing series. Springer. doi: 10.1007/978-3-030-72657-7_16

[2] Note that the SMOTE technique was only applied to the training dataset.

4. Outlier removal: extreme values that could negatively affect the analysis were identified and removed. This helped to improve the stability of the models and avoid misinterpretations.
5. Principal Component Analysis (PCA): the PCA algorithm was applied in order to reduce the dimensions of the dataset and thus obtain better results from the analysis.

Results

The results of the analysis are presented below, with the results of the classification algorithms first, followed by the results of the clustering algorithms.

Data Visualization

Below are all three data visualization algorithms that we have implemented. They are PCA, t-SNE, and EDA respectively.

PCA

PCA or Principal Component Analysis is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.[3]

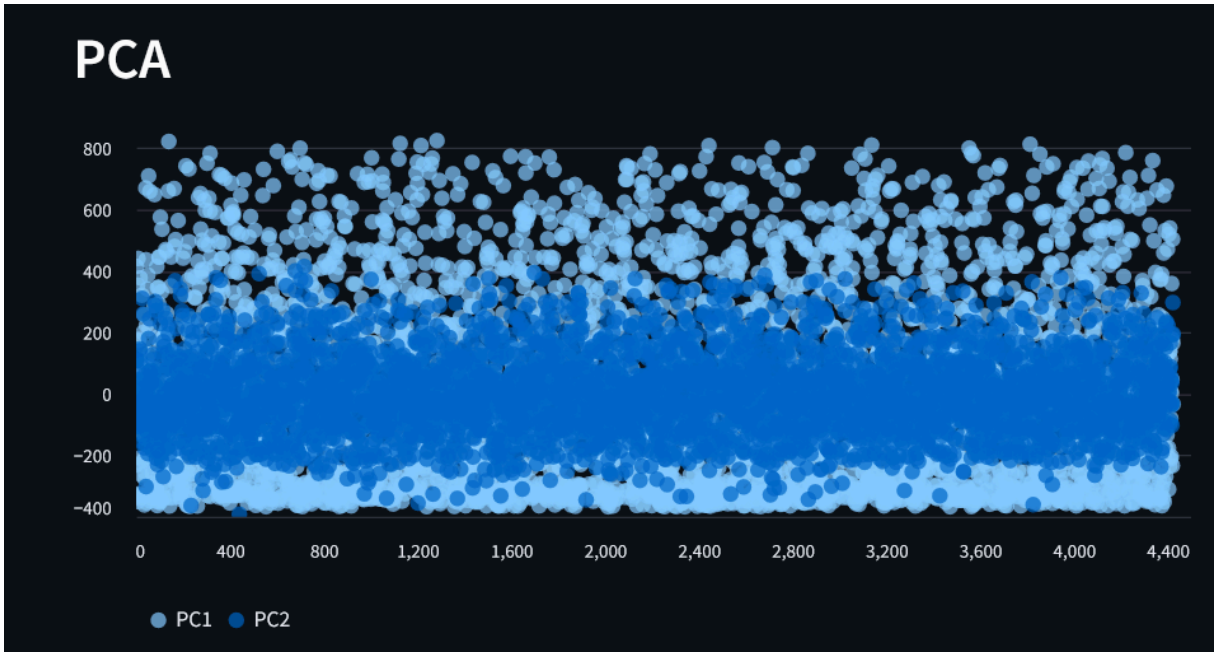

Figure 5: PCA Graph with two classes, PC1 and PC2.

As seen in Figure 5, each dot represents an instance of the data, and the colors represent the different features of the data. In this graph we can see the seperation of the data into two features. Those features can be any of the features available in the dataset.

t-SNE

t-SNE (t-distributed Stochastic Neighbor Embedding) is an unsupervised non-linear dimensionality reduction technique for data exploration and visualizing high-dimensional data.[4]

[3] What is Principal Component Analysis (PCA) and How it is Used? - Sartorius

[4] Introduction to t-SNE - Datacamp

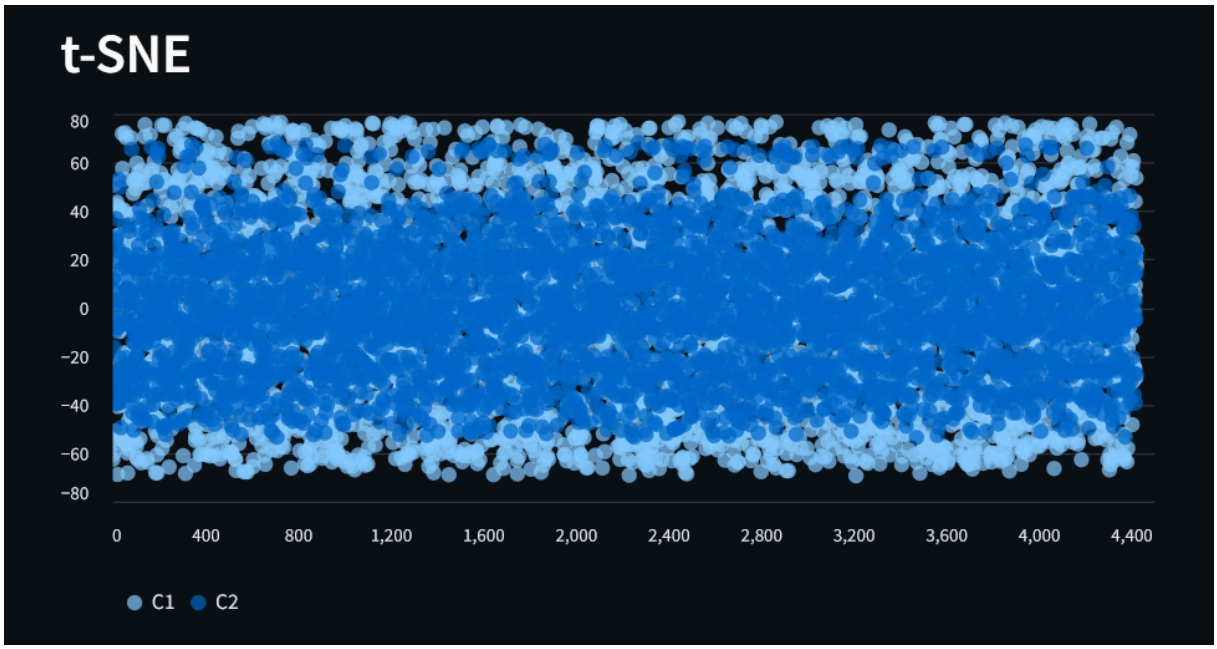

Figure 6: t-SNE Graph with two features, C1 and C2.

As seen in Figure 6, just like PCA, each dot represents an instance of the data, and the colors represent the different features of the data. In this graph we can see the seperation of the data into two features. Those features can be any of the features available in the dataset.

EDA

Exploratory Data Analysis (EDA) is an analysis approach that identifies general patterns in the data. These patterns include outliers and features of the data that might be unexpected.[5]

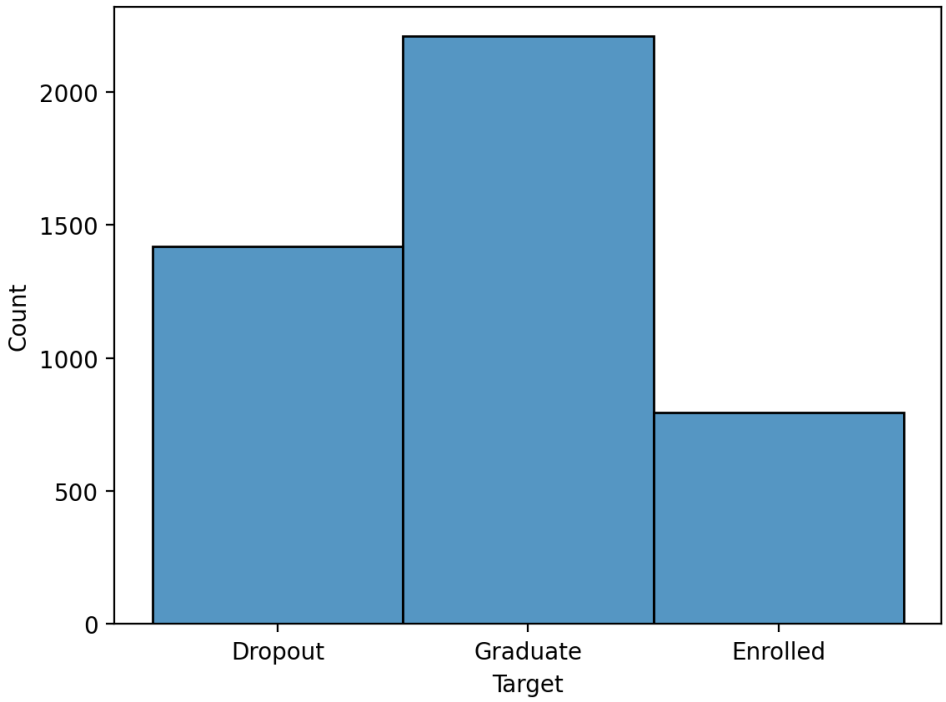

Figure 7: EDA Histogram

[5] Exploratory Data Analysis | US EPA

As seen in Figure 7, the histogram shows the distribution of the data, and each bar represents a feature of the data. It visually represents the number of instances for each feature in the dataset.

Machine Learning

We have two types of Machine Learning algorithms, Classification and Clustering. Under classification algorithms, we have Logistic Regression and KNN Classification. Under clustering algorithms, we have KMeans Clustering and Hierarchical Clustering.

Logistic Regression

In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (the coefficients in the linear combination).[6]

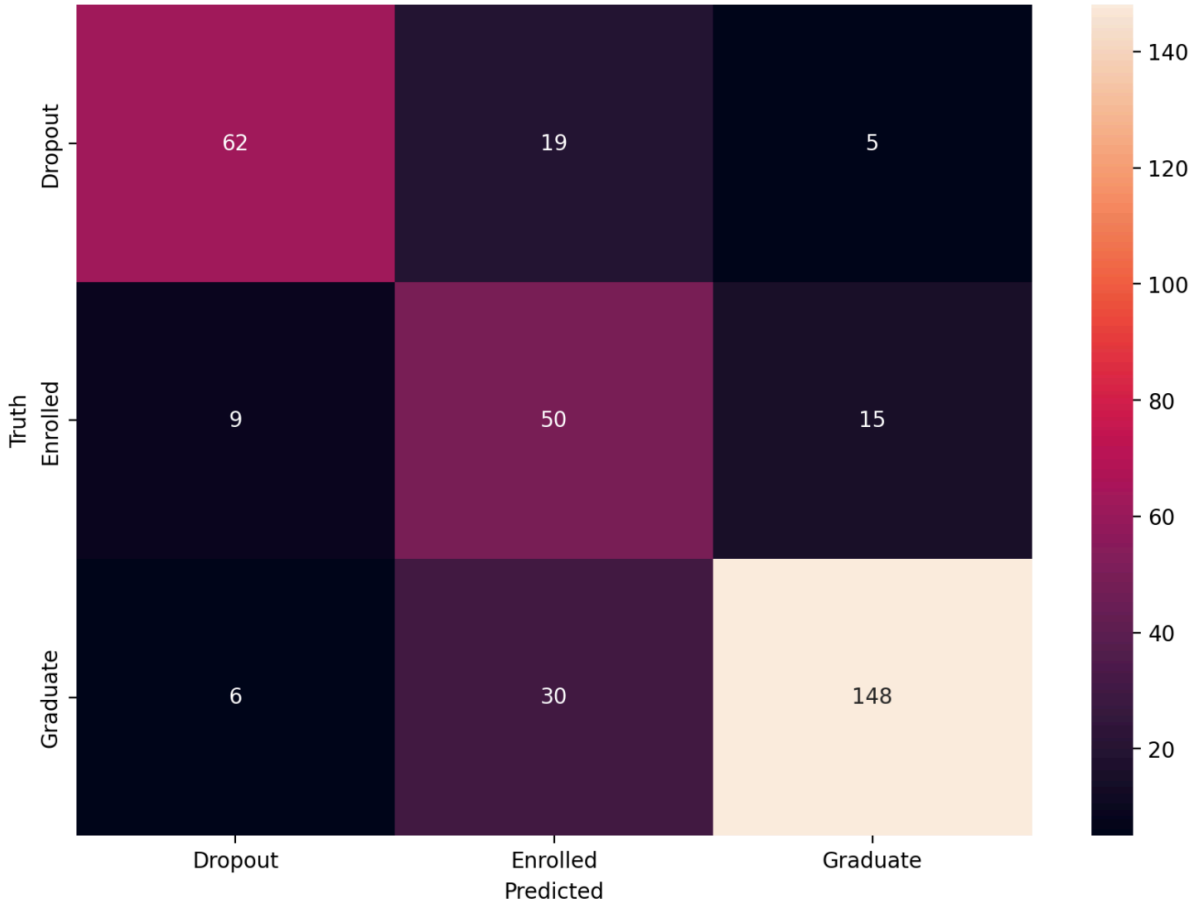

Figure 8: Logistic Regression Confusion Matrix

We have limited the test size s for $s \in (0.0, 0.5]$ .The results of running logistic regression on the dataset, with a test size of 0.15 are as follows:

- Accuracy: 75.58%
- Precision: 75.58%
- Recall: 75.58%
- F1 Score: 75.581%

[6] Logistic regression - Wikipedia

KNN Classification

The k-nearest neighbors algorithm (k-NN) is a non-parametric supervised learning method. It is used for classification and regression. In both cases, the input consists of the k closest training examples in a data set. The output depends on whether k-NN is used for classification or regression.[7]

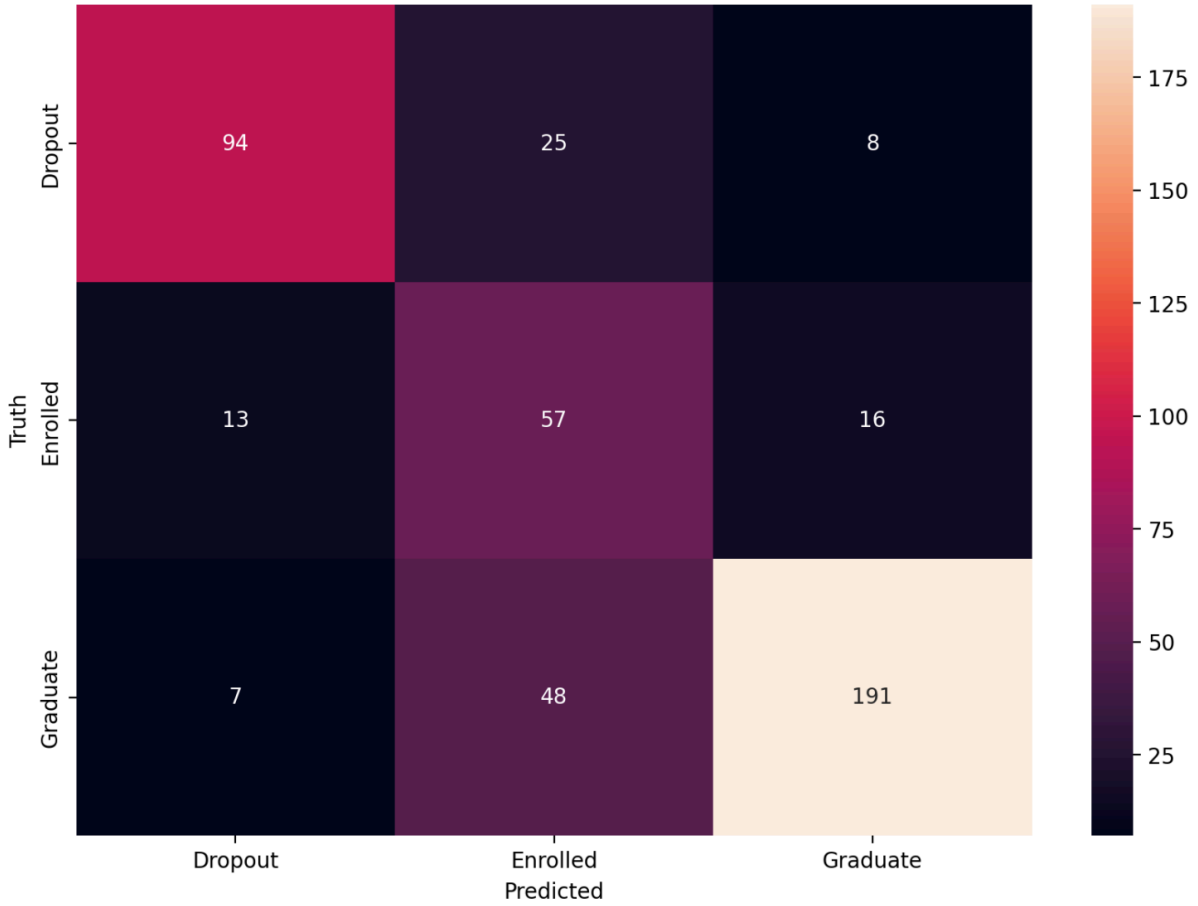

Figure 9: KNN Classification Confusion Matrix

We have limited the value of k to be 1 or more, since it would not make sense to classify 0 or negative amounts of features. The results of running KNN classification on the dataset, with $k = 73$, are as follows:

- Accuracy: 74.51%
- Precision: 74.51%
- Recall: 74.51%
- F1 Score: 74.51%

KMeans Clustering

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.[8]

[7] K-nearest neighbors algorithm - Wikipedia

[8] K-means clustering - Wikipedia

We have limited the value of k to be 2 or more, since it would not make sense to cluster 1, 0 or negative amounts of features. The result of running KMeans clustering on the dataset, with $k = 2$, is a Sillhouette score of 79.36.

Hierarchical Clustering

In data mining and statistics, hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of cluster analysis that seeks to build a hierarchy of clusters.[9]

Using the Chebyshev distance metric, the result of running Hierarchical clustering on the dataset, is a Sillhouette score of 65.33.

Conclusion

We have analyzed the dataset using both classification and clustering algorithms. The conclusion for each of the types of algorithms are listed below.

Classification

Before commenting on the results, note that different values of k were tested for the kNN algorithm and the results presented are as a function of $k = 73$. More specifically, the starting point for testing different values of k in order to find the optimal number of neighbors was the square root of the number of samples which was approximately equal to 66. The results presented above indicate that both algorithms are equally successful in achieving good classification. Also, the subject matter of our analysis does not prompt us to prefer a particular metric as would be the case if we were talking about a disease and we would be more interested in the accuracy of the negative prediction of the model, for example. Therefore, we conclude that both algorithms are equally effective.

Clustering

The results we obtained as output from our analysis when clustering the data are 65.33 silhouette score for the Hierarchical clustering algorithm and 79.36 for the K-means clustering algorithm for number of clusters equal to 2. However, knowing that our data is categorized into 3 different groups in total we will consider the result we obtained as output for number of clusters equal to 3, which is 65 silhouette score. Therefore, we conclude that both algorithms achieve equally good clustering.

Software Lifecycle

Software Lifecycle or Software Development Lifecycle (SDLC) is a process used by the software industry to design, develop and test high-quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, and reaches completion within times and cost estimates.

Agile Methodology

Agile is a type of project management methodology, mainly used for software development, where demands and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change and includes six phases: Planning, Design, Development, Testing, Deployment, and Review.

[9] Hierarchical clustering - Wikipedia

Releasing to the public

Using the Agile methodology, it requires adjustment to some of the phases mentioned above. The 6-phase cycle will be repeated for each new software iteration.

Planning

During this phase, the team will research the market and the competition to understand the needs of their customers, alongside planning the features and the timeline of the project.

Design

During the Design phase, they will create the app's design, which includes the user interface and user experience, accounting for the information they gathered through research during planning.

Development

In this phase, the team will focus on implementing the features that were planned and designed, while following certain software design principles that allow for easy implementation of new features, patching and bug fixing.

Testing

In the Testing phase, the team tests the features they implemented, and make sure they work as intended, using various testing principles such as unit testing and end-to-end testing. Unit and E2E (end-to-end) testing are automated through use of testing frameworks. Alongside them, they can incrementally test the software with real users, using A/B testing (non opt-in) and beta application clients (opt-in), to gather feedback and improve the software.

Deployment

In the Deployment phase, the team should release the software to the public, allowing for use of testing principles mentioned above, to ensure the software is working as intended.

Review

Finally, during Review, the team will gather feedback from the users, and use it to improve the software, by adding new features, patching bugs, and improving the user experience.

Contributions

- Γαβριήλ Κοσκινάς implemented data fetching, data preprocessing, data analysis using Machine Learning, as well as wrote the algorithmic Analysis and Conclusion remarks.
- Μιχαήλ Αβαγιανός created the user interface, the data visualization, the UML and modularized the code.
- Θανάσης Ράτσικας containerized the application, and wrote the majority this document.