

**UNIVERSITI TUNKU ABDUL RAHMAN**

**LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE**

**UECM3993 PREDICTIVE MODELLING**

**GROUP ASSIGNMENT**

**JANUARY 2024 TRIMESTER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Student ID** | **Programme** | **Contribution** | **Signature** |
| 1. | Ding Yuan Xing | | | | | 2106071 | AM | 100%/7 |  |
| 2. | Gan Min Liang | | | | | 2102657 | AS | 100%/7 |  |
| 3. | Hor Huai Liang | | | | | 2101687 | AS | 100%/7 |  |
| 4. | Lee Kim Lan | | | | | 2104745 | AM | 100%/7 |  |
| 5. | Phuah Jun Wei | | | | | 2102824 | AS | 100%/7 |  |
| 6. | Ryan Lee Yung Hong | | | | | 2102195 | AM | 100%/7 |  |
| 7. | Tan Wan Xuen | | | | | 2207214 | AM | 100%/7 |  |

**Table of Content**

|  |  |
| --- | --- |
| Title | Page |
| **1.** [**Introduction**](#_Introduction)  [**2. Data Preparation**](#_2._Data_Preparation_1)  **2.1 Data exploration**  **2.2 Data preprocessing**  [**3. Model Creation**](#_3._Model_Creation)  **3.1 Supervised models**  **3.1.1 Logistic Regression**  **3.1.2 k-Nearest Neighbours**  **3.1.3 Naïve Bayes**  **3.1.4 Decision Trees (CART)**  **3.1.5 Random Forest**  **3.2 Unsupervised models**  **3.2.1 PCA**  **3.2.2 K-modes Clustering**  **3.2.3 Hierarchical Clustering**  **3.2.4 Mean-Shift Clustering**  [**4. Conclusion**](#_4.0_Conclusion)  **4.1 Supervised Machine Learning**  **4.2 Unsupervised Machine Learning**  **4.3 Overall Conclusion**  **Reference** | 3 - 5  6 - 12  13 - 32  33 - 35  36 |

# **1. Introduction**

The Higher Evaluation Students Performance Evaluation Dataset is a dataset collected from the Faculty of Engineering and Faculty of Education Science students in 2019. The dataset consists of collection of data related to student’s performance in higher education. The dataset is accessible via this link: [https://archive.ics.uci.edu/dataset/856/higher+education+students+  
 performance+evaluation](https://archive.ics.uci.edu/dataset/856/higher+education+students+performance+evaluation). Student’s academic performance is greatly influenced by their time spent in education. It is imperative for educators, politicians, and researchers to comprehend the myriad aspects that influence student’s academic achievement. Institutions can improve the quality of education they offer by identifying patterns, trends, and opportunities for development through the analysis of student performance data. This dataset contains attributes pertaining to personal questions, family related questions, and educational habits. There 31 features in this dataset which are:

|  |  |  |
| --- | --- | --- |
| Column | Variable name | Description |
| 1 | Student Age | 1: 18-21, 2: 22-25, 3: above 26 |
| 2 | Sex | 1: female, 2: male |
| 3 | Graduated high-school type | 1: private, 2: state, 3: other |
| 4 | Scholarship type | 1: None, 2: 25%, 3: 50%, 4: 75%, 5: Full |
| 5 | Additional work | 1: Yes, 2: No |
| 6 | Regular artistic or sports activity | 1: Yes, 2: No |
| 7 | Do you have a partner | 1: Yes, 2: No |
| 8 | Total salary if available | 1: USD 135-200, 2: USD 201-270, 3: USD 271-340, 4: USD 341-410, 5: above 410 |
| 9 | Transportation to the university | 1: Bus, 2: Private car/taxi, 3: bicycle, 4: Other |
| 10 | Accommodation type in Cyprus | 1: rental, 2: dormitory, 3: with family, 4: Other |
| 11 | Mothersâ€™ education | 1: primary school, 2: secondary school, 3: high school, 4: university, 5: MSc., 6: Ph.D. |
| 12 | Fathersâ€™ education | 1: primary school, 2: secondary school, 3: high school, 4: university, 5: MSc., 6: Ph.D. |
| 13 | Number of sisters/brothers (if available) | 1: 1, 2:, 2, 3: 3, 4: 4, 5: 5 or above |
| 14 | Parental status | 1: married, 2: divorced, 3: died - one of them or both |
| 15 | Mothersâ€™ occupation | 1: retired, 2: housewife, 3: government officer, 4: private sector employee, 5: self-employment, 6: other |
| 16 | Fathersâ€™ occupation | 1: retired, 2: government officer, 3: private sector employee, 4: self-employment, 5: other |
| 17 | Weekly study hours | 1: None, 2: <5 hours, 3: 6-10 hours, 4: 11-20 hours, 5: more than 20 hours |
| 18 | Reading frequency (non-scientific books/journals) | 1: None, 2: Sometimes, 3: Often |
| 19 | Reading frequency (scientific books/journals | 1: None, 2: Sometimes, 3: Often |
| 20 | Attendance to the seminars/conferences related to the department | 1: Yes, 2: No |
| 21 | Impact of your projects/activities on your success | 1: positive, 2: negative, 3: neutral |
| 22 | Attendance to classes | 1: always, 2: sometimes, 3: never |
| 23 | Preparation to midterm exams 1 | 1: alone, 2: with friends, 3: not applicable |
| 24 | Preparation to midterm exams 2 | 1: closest date to the exam, 2: regularly during the semester, 3: never |
| 25 | Taking notes in classes | 1: never, 2: sometimes, 3: always |
| 26 | Listening in classes | 1: never, 2: sometimes, 3: always |
| 27 | Discussion improves my interest and success in the course | 1: never, 2: sometimes, 3: always |
| 28 | Flip-classroom | 1: not useful, 2: useful, 3: not applicable |
| 29 | Cumulative grade point average in the last semester (/4.00) | 1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49 |
| 30 | Expected Cumulative grade point average in the graduation (/4.00) | 1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49 |
| 31 | Course ID | Course id |
| 32 | OUTPUT Grade | 0: Fail, 1: DD, 2: DC, 3: CC, 4: CB, 5: BB, 6: BA, 7: AA |

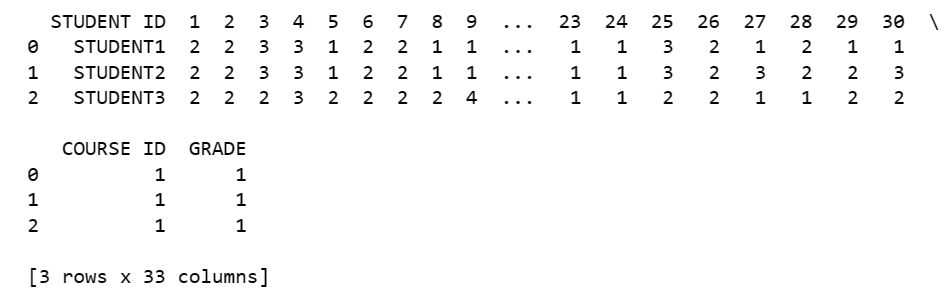
The purpose of this dataset is to predict student’s end-of-term performances using predictive models. Moreover, this dataset can help to determine how each feature is correlated to the student’s end-of-term performances.

# **2. Data Preparation**

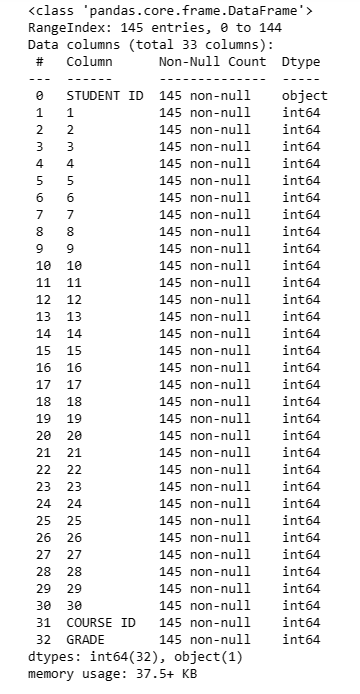
Data preparation involves three critical stages: data exploration, data splitting, and data preprocessing.

2.1 Data Exploration

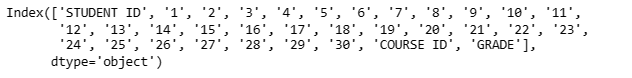
The 'Higher Education Students Performance Evaluation Dataset.csv' dataset is imported using the pandas library. Upon examination of the first three rows, it is evident that the dataset consists of categorical data.

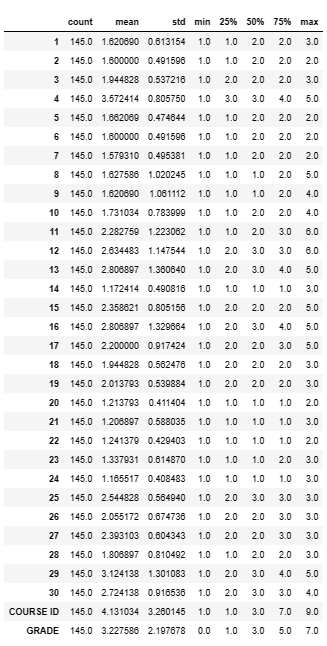


Additionally, the dataset's information is inspected to determine the data type of each column. It is observed that each column contains 145 entries.

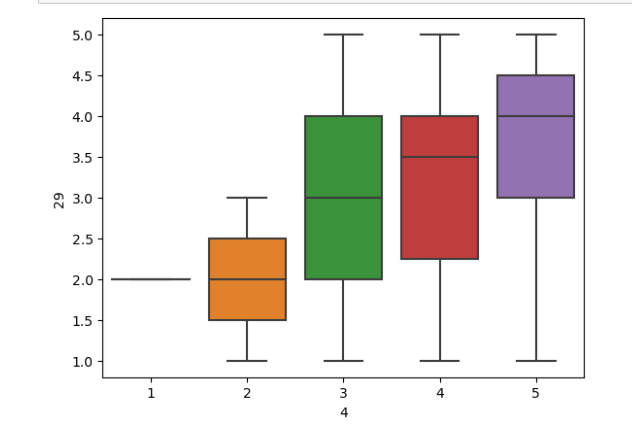


The descriptive statistic of the dataset is displayed to reveal insights into the data's distribution. The shape of dataset is determined as (145, 33). To verify alignment, the column names are checked and printed alongside the rows.





Next, in the data visualisation phase of data preparation, graphical representations are utilized to interpret dataset characteristics effectively. The graph is generated to provide insights into the dataset. The first graph plots column ‘29’ (Cumulative Grade Point Average-CGPA) against column ‘4’ (Scholarship Type). The boxplot does not contain any outliers and illustrates the distribution of CGPA scores across different scholarship categories. Based on the features explained earlier in the introduction, the students with higher scholarship award are capable to obtain a better result as it can noticed by the increasing of median CGPA values in each box. This is because the students need to score a higher CGPA to maintain as a scholarship holder.

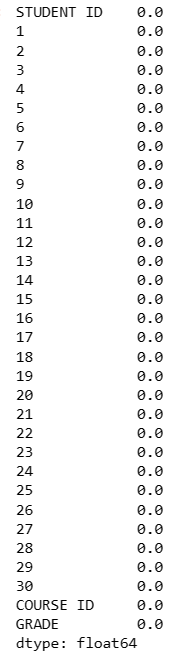


2.2 Data Preprocessing

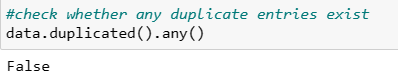
After completing various steps of Exploratory Data Analysis (EDA), which involve examining data distribution and visualizing data, several additional processes such as handling missing values, removing duplicates, treating outliers, conducting bivariate analysis, and encoding categorical variables still need to be addressed during the data preprocessing stage.

When considering the question of why EDA should be performed, it becomes evident that EDA plays a crucial role in identifying and resolving data quality issues, allowing for effective testing of assumptions and hypotheses about the data. Furthermore, EDA provides a comprehensive understanding of the data, facilitating clear communication and presentation of findings and insights to stakeholders. Additionally, EDA helps in generating new features, selecting the most relevant features, and determining the appropriate Machine Learning techniques to apply.

Continuing with the data preparation process, the existence of missing values in the dataset is checked. As depicted in the figure below, no missing values are observed.



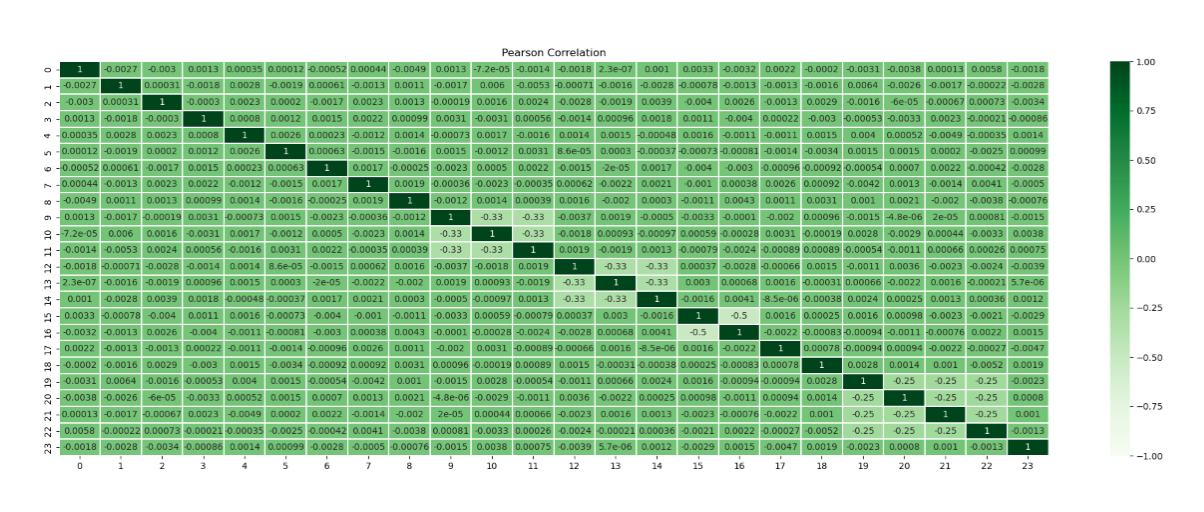
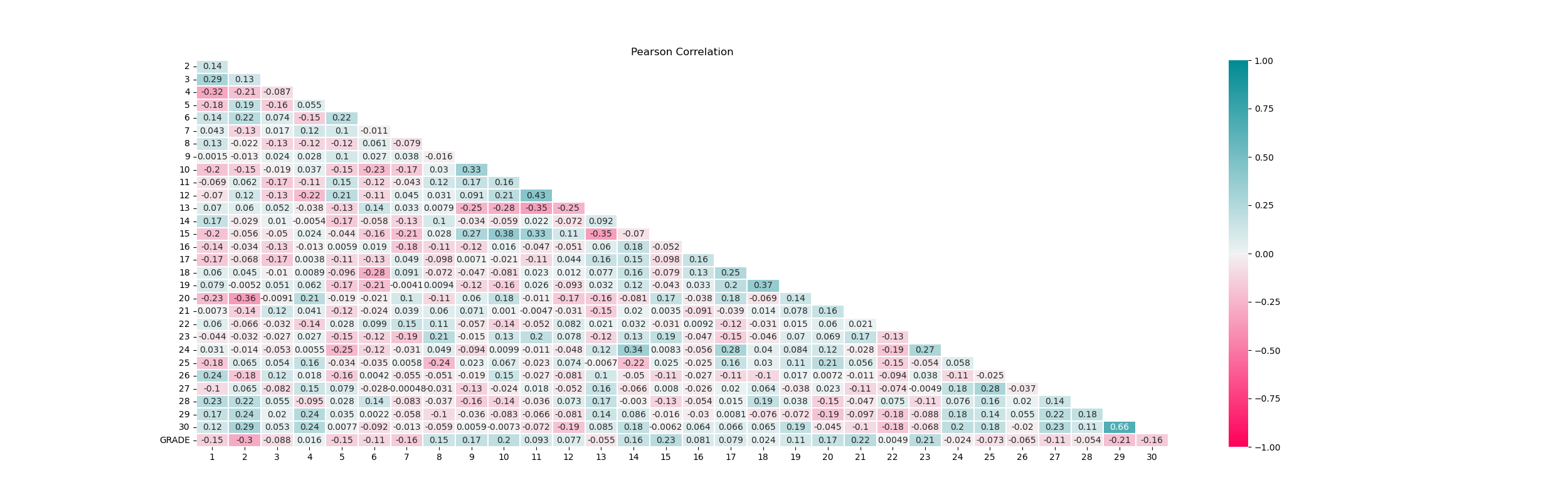
Duplicate entries and rows are also examined. If any duplicated rows are found, they are eliminated to prevent bias in the analysis. There are no duplicated values or rows present in the dataset.

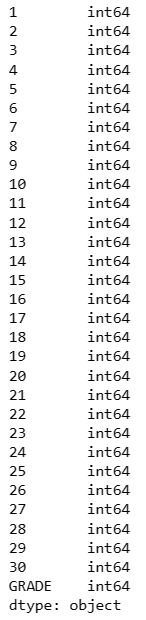
.



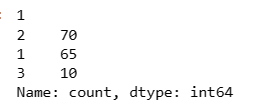
To enhance the clarity of the data information, the unused column ‘STUDENT ID’ and ‘COURSE ID’ are omitted.



Besides that, we cannot determine outliers since all the data is categorical. The correlation among all variables is assessed, and a heatmap ranging from –1 to 1 is generated. The majority of the heatmap values are negative and proximal to zero. It illustrates a negative correlation and denotes an inverse association between the variables.Moreover, the masked correlation matrix is utilized to remove redundancies in the heatmap by displaying only the lower triangular portion. This masking procedure reveals the same negative correlation pattern observed in the heatmap.  
The data type of each column is listed out explicitly.



Additionally, exploratory data analysis encompasses the identification of unique values, the determination of the frequency of each category within a column, and the calculation of their respective counts. These aspects are depicted in the figures below. Specifically, the unique values and their counts within each category in the columns are identified, providing insights into the composition of the categorical data. Furthermore, the count of unique values is determined, such as in column '1', which represents student age.

Next, the dataset is facing multiclass problem since there are 7 categories of output from the ‘GRADE’ column. To overcome this problem, category 0 is determined as ‘Fail’ while categories 1, 2, 3, 4, 5, 6 and 7 are identified as ‘Pass’. Then, the original ‘GRADE’ column is replaced by the specified values, which are ‘Pass’ and ‘Fail’. To encode the categorical variables, ‘Pass’ is converted to ‘0’ while ‘Fail’ is converted to ‘1’.

2.2 Data Splitting

The original dataset is commonly subdivided into two smaller sets, train and test sets, during the data splitting process to enhance model performance. This division addresses overfitting, where the model excessively tailors itself to the training data and results in inaccurate predictions during evaluation. Typically, the train set comprises the larger proportion of the two datasets, such as an 80% train set and a 20% test set. The table below shows the respective purposes of these smaller sets.

|  |  |
| --- | --- |
| **Datasets** | **Purpose** |
| Train | * Construct the potential models that appear to make practical sense |
| Test | * Give accurate model estimations * Provide final measure of the model’s performance on new data before deploying the model into production |

Typically, the output variable is excluded from the model evaluation process to prevent it from influencing the training phase. This separation ensures that the model does not learn from the output variable and thus avoids biased or inaccurate predictions on new data. In the 'Higher Education Students Performance Evaluation Dataset.csv' dataset, comprising 30 independent variables and an output variable named 'GRADE', the output variable is isolated prior to splitting the data into train and test sets. Subsequently, the data is divided into train and test sets, with 80% allocated to the train data and 20% to the test data. Finally, the sample sizes of the train and test sets are printed to verify the split. 

However, upon the model creation, we noticed that the kappa values are invalid. Hence, to enhance the model performance, we performed stratified k-fold cross validation to handle the class imbalance problem. Basically, the dataset randomly partitioned into k folds in a way that each fold contains the similar proportion of samples from each class as the original data (Szeghalmy and Fazekas, 2023). For our dataset, 5 folds are used as the number of folds.

Thus, we applied the holdout method and stratified cross-validation method to compare these two approaches for achieving better model performance in each supervised model.

# **3.0 Model Creation**

3.1 Supervised models

Predictive models like k-Nearest Neighbor (kNN), logistic regression, decision trees, random forest and naive bayes are used.

When dealing with imbalanced datasets, accuracy might not be the best metric to evaluate the model's performance. This is because accuracy can be misleading in the presence of class imbalance. Hence, metrics like precision, recall, F1-score and kappa scores will be considered to provide a better view of model's performance. Also, GridSearchCV method is not employed to optimize the model. This is because GridSearch CV results in decreased performance. After performing hyperparameter tuning using GridSearchCV, both f1-score and recall drop, hence the model is not able to accurately classify the instances. When the model is trained on a small dataset, it may result in overfitting and lead to poor generalization performance on unseen data.

3.1.1 Logistic Regression

Logistic regression is a classification model which known for its simplicity in implementation and its ability to yield robust results, particularly when dealing with linearly separable classes (Edgar & Manz, 2017). In scenarios involving binary classification tasks, such as predicting pass or fail, logistic regression often offers a more suitable fit. Compared to polynomial and linear regression, logistic regression presents a simpler and more effective solution for addressing binary and linear classification problems. Moreover, logistic regression makes it convenient to interpret the results and in addition to being fast.

**Holdout Method:**

A screenshot of a test

Description automatically generated

In the test set, the logistic regression model exhibits a precision of 86% for class 0 which indicate a high proportion of correctly identified non-default instances, and a recall of 100%, which suggest that all non-default instances are accurately identified. However, for class 1, the precision is perfect at 100%, but the recall is very low, hence the model misses a big portion of actual default instances. This F1-scores of 93% for class 0 and 0% for class 1 illustrate the trade-off between precision and recall. Furthermore, accuracy may not provide a comprehensive understanding, especially in imbalanced datasets.

A black text on a white background

Description automatically generated

Through the confusion matrix above, model correctly predicted 25 instances as belonging to class 0. However, the model incorrectly predicted 4 instances as belonging to class 0 when they actually belonged to class 1, resulting in false positives. Therefore, it has high recall for positive class but poor specificity.



The kappa value of 0.0 obtained suggests a low level of agreement between the model’s prediction and the actual labels. Therefore, the holdout method performs poorly due to the imbalanced data.

**K-fold Cross-Validation Method:**

A number with numbers and symbols

Description automatically generated with medium confidence

After applying k-fold cross-validation method, the kappa value of 0.1205 suggests slight agreement between the predicted and actual classes. Hence, k-fold cross-validation provides a more reliable estimate of the model's performance.

3.1.2 kNN

The k-Nearest Neighbours (kNN) algorithm is a non-parametric, supervised learning classifier and regression. The kNN classifier is suitable for handling and predicting categorical response, while the kNN regression is suitable for handling and predicting quantitative response. kNN algorithms are simple, effective, makes no assumptions about the relationship between the data distribution and fast to train. However, it has a slow classification phase and inability to provide mathematical reasoning on how the features are related to the class. kNN model searches for 𝑘 similar inputs compared to given new data from existing dataset and predicts the response for the new data by considering only the responses of the 𝑘 similar inputs, where 𝑘 is a parameter determined from observing the existing dataset. Classification of the new data is based on the distance between the new data to each data in the dataset using Minkowski distance.

**kNN-model:**

We set our k-values from range of 1-10. Then we plot the accuracy score against k-values and the error rate against k-values.

A graph of a graph with numbers and lines

Description automatically generated with medium confidence

A line graph with numbers and points

Description automatically generated

The best k-value for this model is k=1 and the classification report of the model when k=1 is shown below

A screenshot of a computer

Description automatically generated

In the test set, the kNN model exhibits a precision of 86% for class 0 which indicate a high proportion of correctly identified non-default instances, and a recall of 100%, which suggest that all non-default instances are accurately identified. However, for class 1, the precision is very low at 0%, so as the recall, hence the model does not handle the class well at all. This F1-scores of 93% for class 0 and 0% for class 1 illustrate the trade-off between precision and recall. Furthermore, accuracy may not provide a comprehensive understanding, especially in imbalanced datasets.

A black and white text

Description automatically generated

Through the confusion matrix above, model correctly predicted 25 instances as belonging to class 0. However, the model incorrectly predicted 4 instances as belonging to class 0 when they actually belonged to class 1, resulting in false positives. Therefore, it has high recall for positive class but poor specificity.



The kappa value of 0.0 obtained suggests a low level of agreement between the model’s prediction and the actual labels. Therefore, the kNN model performs poorly due to the imbalanced data.

**Stratified kfolds kNN:**

We try to improve the model performance by performing stratified k-folds kNN. We set the number of splits to be 5. After training and testing the stratified kNN model, we obtained the results below.

A graph of accuracy versus number of neighbors

Description automatically generated

A graph with red dots

Description automatically generated

The above graph shows the mean error rates of the 5 folds against the number of neighbors k.

A screenshot of a test

Description automatically generated

The best k-values obtained is 3. The stratified kNN model exhibits a precision of 96% for class 0 which indicate a high proportion of correctly identified non-default instances, and a recall of 93%, which suggest that most non-default instances are accurately identified. For class 1, the precision increased from 0% to 33%, and the recall increased to 50%, the model is able the detect the class quite well, but still includes observations of other classes. This F1-scores of 96% for class 0 and 40% for class 1 illustrate the trade-off between precision and recall.

A black and white text

Description automatically generated

Through the confusion matrix above, model correctly predicted 25 instances as belonging to class 0 and predicted 2 wrong instances that were supposed to be in class 0, resulting in false negative. The model correctly predicted 1 instance and incorrectly predicted 1 instance too as they should be belonging to class 1, resulting in false positives. Therefore, it has high recall for positive class but poor specificity.



The kappa value of 0.3459 obtained suggests a relatively poor level of agreement between the model’s prediction and the actual labels.

**Best kNN model:**

After performing few adjustments to the kNN model, we choose stratified kNN as the most suitable kNN model for this dataset. We can see from the result above, with the first kNN model having kappa value of 0 which suggests no level of agreement between the model’s prediction and the actual labels. While for stratified kNN, even though the kappa value is not that high with only 0.3459, but it is still higher than 0 kappa value. Thus, we choose stratified kNN result as the suitable kNN model for the comparison to other models in the conclusion part.

3.1.3 Naïve Bayes

Apart from that, Naive Bayes classifiers are commonly utilized in various applications such as text classification, sentiment analysis, and recommendation systems. There exist four primary types of Naive Bayes Classifiers: Optimal Naive Bayes, Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes. We choose to use Gaussian Naïve in this context.

**Holdout Method:**

A screenshot of a computer screen

Description automatically generated

From the classification report of the test set, the model exhibits high precision and recall for class 1. However, its performance on class 0 is compromised which results in a lower F1-score which is 0.00 for class 1 compared to class 0. For class 0, the precision is 0.86, and the recall is 1.00, while the F1-score is 0.93, thus indicating a relatively good balance between precision and recall. Based on these observations, class 0 demonstrates well-modelled performance. Conversely, for class 1, both the precision and recall is 0.0.

A black text on a white background

Description automatically generated

In this case, the model correctly classified 25 instances of class. However, the model incorrectly predicted 4 instances of class 1 as class 0. A kappa value of 0.0 suggests the model's performance is equivalent to randomly guessing the labels by using holdout method in this model.

In Naive Bayes, it has been observed that hyperparameter tuning does not enhance classification performance because of the limited number of hyperparameters and the simplicity of Naive Bayes classifiers. The number of hyperparameters to be tuned is determined by the number of classes in the dataset. Therefore, it can be concluded that hyperparameter tuning is unnecessary and ineffective for improving the Naive Bayes model (Tokuç, 2021).

**K-fold Cross Validation Method:**

A number and a number

Description automatically generated with medium confidence

After using k-fold cross-validation method in training Naïve Bayes model, the average kappa values of 0.0787 obtained shows that only slight agreement between the predicted and actual classifications. Therefore, the kappa values suggest only slight agreement between predicted and actual classifications, with slightly better performance observed in the k-fold cross validation method.

3.1.4 Decision Trees (CART)

Moreover, decision trees algorithm is used to fit the dataset. In accordance with Javapoint (n.d.), while it can be applied to both classification and regression problems, decision tree is a supervised learning technique that is most often employed to solve classification problems. It is a graphical tool that shows all the options for solving a problem or making a decision given certain parameters.

**Holdout Method:**

A screenshot of a computer

Description automatically generated

According to the classification report above, the precision of class 0 is 0.86, indicating that when the model predicts a data point as belonging to class 0, it is correct approximately 86% of the time. For class 1, the precision is 1.00, indicating that when the model predicts a data point as belonging to class 1, it is correct 100% of the time. However, this precision value should be interpreted with caution, considering the low recall for class 1 which is 0.00, meaning the model fails to identify any instances of class 1. Then, the f1-score of class 0 is 0.93, indicating a good balance between precision and recall while f1-score of 0.00 reflects the poor performance in correctly classifying instances of class 1. There is an accuracy of 0.86, indicating 86% of the predictions are correct.

A black text on a white background

Description automatically generated

The model produces zero true positives and four false negatives since it accurately predicts every occurrence of class 0 but fails to forecast any instances of class 1. This is demonstrated by the confusion matrix. It is clear that the model's performance is highly constrained because its kappa value of 0.0 indicates that there is no agreement that goes beyond chance.

**K-fold Cross Validation Method:**

****

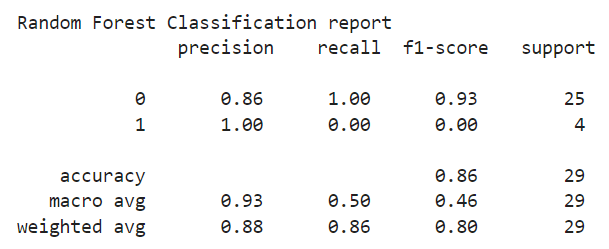
The model’s performance over several folds in k-fold cross-validation is indicated by the average accuracy of 0.9034 and average kappa value of 0.093. Given the high average accuracy, it can be concluded that the model predicts the class labels for around 90% of the dataset's cases accurately on average. Nonetheless, the model's performance is just marginally better than chance, as indicated by the low average kappa value. This could be the result of problems like overfitting or class imbalance.

In conclusion, while the decision tree model demonstrates high accuracy in some evaluations, its inability to predict instances of class 1, as observed in the holdout method, indicates a significant limitation. The discrepancy between accuracy and kappa values suggests the presence of underlying issues such as class imbalance or overfitting.

3.1.5 Random Forest

In machine learning, random forest is a potent ensemble learning method that is frequently applied to regression and classification problems. During training, it builds a large number of decision trees, from which it produces the mean prediction (regression) or the mode of the classes (classification). In order to improve generalisation performance and strengthen the model's resistance to overfitting, each decision tree is trained on a subset of the dataset and employs a random selection of features at each node. Random forest reduces the possibility of bias and volatility seen in individual trees by combining the predictions of several trees. This produces a reliable and accurate predictive model appropriate for a range of applications in both academia and business.

**Holdout Method:**



The model's performance is evaluated in the random forest classification report, which shows that while the precision for class 0 was high (93%) it was misleadingly perfect (class 1) since there were not enough cases found. class 0 has outstanding recall (100%), however class 1 has no recall (0%). The balance of precision and recall, the f1-score, is notably high (93%) for class 0 but non-existent for class 1. Even with its 86% accuracy rate, the model has trouble correctly identifying occurrences of class 1, which leaves us with an inadequate picture of how well it performs.

A black text on a white background

Description automatically generated

The confusion matrix summarises the true positive, false positive, true negative, and false negative predictions to show how well the random forest model performed. Here, every instance is expected to be class 0, which results in 25 accurate positive predictions for that class and 4 inaccurate negative predictions for class 1. The kappa value, which is computed to be 0.0, indicates the degree of agreement between the model's predictions and the actual labels that goes above what would be predicted by chance. This suggests that the model's predictions do not match the actual labels well, exposing potential limitations or problems with the model's performance that require more research and improvement. It also shows that nothing about this arrangement is more random than that.

**K-fold Cross Validation Method:**

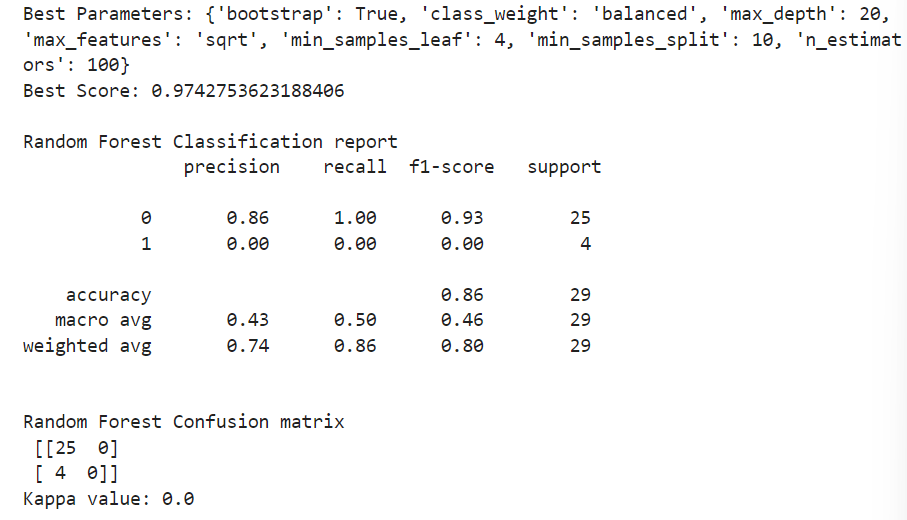
A number and number in black and white

Description automatically generated with medium confidence

The average accuracy of 94.48% indicates that, on average, the random forest model accurately predicts about 94.48% of instances across different folds in k-fold cross-validation. However, the average kappa value of 0.0 suggests that there is no agreement beyond chance between the model's predictions and the actual labels. This indicates that, despite the high accuracy, the model's performance doesn't surpass random guessing. Potential factors contributing to this mismatch could include imbalanced classes, suboptimal feature selection, or overfitting, necessitating further investigation and potential refinement of the model to enhance its predictive capability.

The random forest model exhibits great accuracy in both the holdout method and k-fold cross validation, with an accuracy of 93% in the holdout method and an average of roughly 94.48% over k-fold cross-validation. Nevertheless, the kappa score constantly stays at 0.0 in both evaluation techniques, even with this apparent high accuracy. This implies that nothing in the agreement transcends chance between the model’s predictions and what would be predicted by pure chance. As a result, even while the model can predict class labels with a fair degree of accuracy, its ability to reliably discriminate between classes is restricted.

**Random Forest with GridSearch CV:**



**Random Forest using resampling:**

A screenshot of a computer screen

Description automatically generated

After that, we apply some other methods such as hyperparameter tuning to tweak model performance for optimal results (AWS, n.d.) and resampling method by Synthetic Minority Oversampling Technique (SMOTE) for oversampling imbalanced classification datasets (Jason, 2021). Nevertheless, on the classification report of both of them, it is clear that the f1-score does not increase and kappa value remains at 0.0 which indicating low performance of random forest model on this dataset.

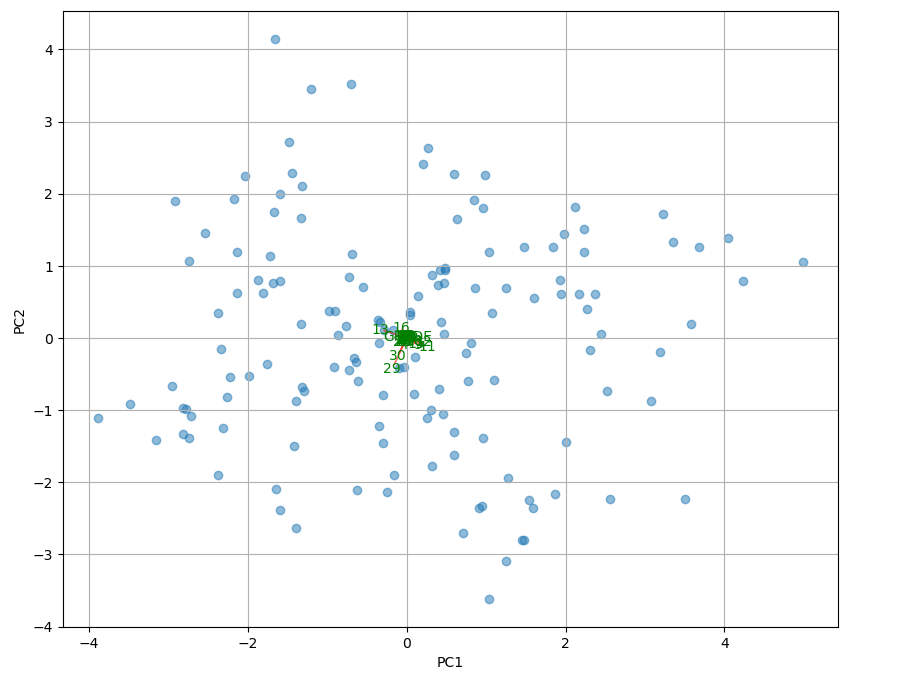
3.2 Unsupervised models

3.2.1 PCA

To further understand the dataset, our first unsupervised learning model used is the principal component analysis also known as PCA method. PCA method is a statistical technique to analyse complex data by reducing its dimensionality to explain the variability in the dataset. PCA is a linear method and is limited because of its dimensionality reducing mapping from a high-dimensional space to a low-dimensional space such as 2D.

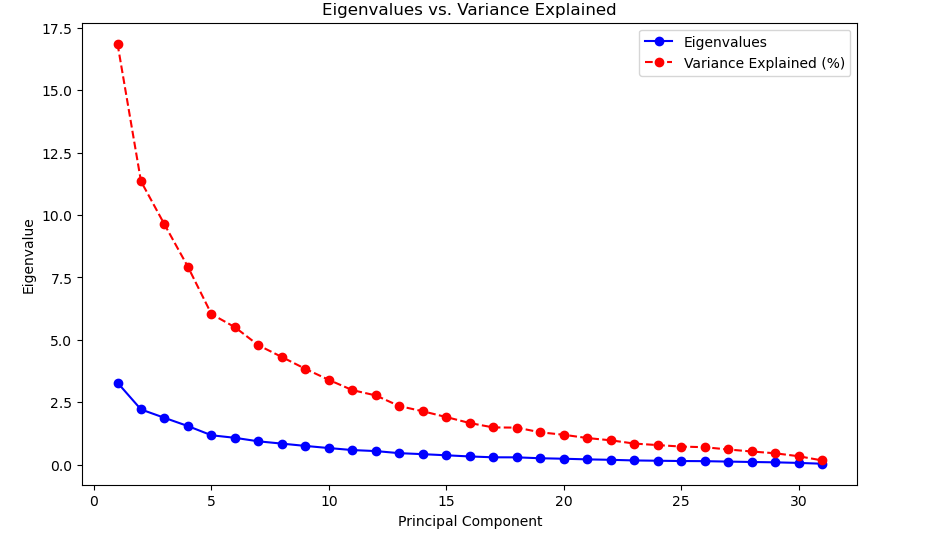
After performing PCA analysis on our dataset, there are two charts that is suitable to represent the results of the analysis. (biplot and screeplot)

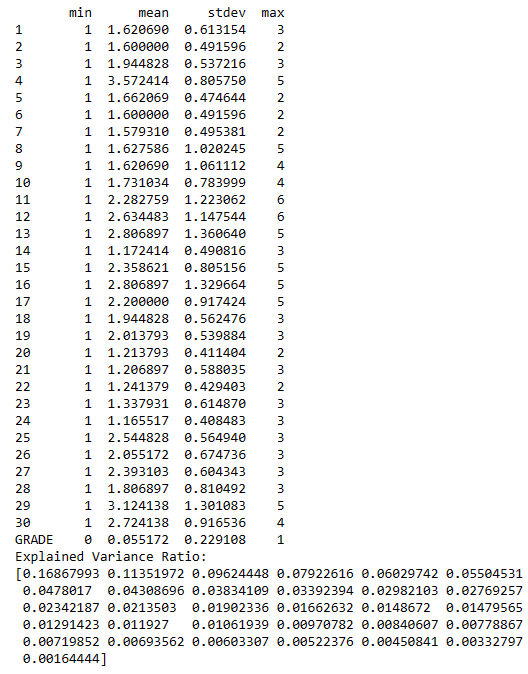
**Biplot**



The biplot above shows a very balanced spread along the principal components. The first two principal components are being chosen to undergo a biplot analysis since they capture a large proportion of the variance of the dataset while the first seven principal components form most of the variance for the dataset. The angles of the vectors provide insights for us to understand the biplot. According to this biplot, the variables “X16” and “X29” are negatively correlated. Small angles generated between the vectors indicate positive correlation while large angles between the vectors indicate negative correlation. The length of the vectors could explain the magnitude of the correlation between the vectors.

Scree Graph



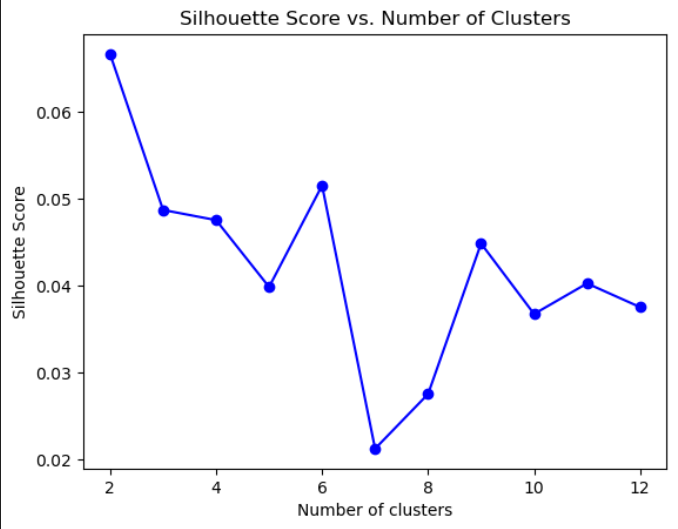


According to the scree graph, the first 5 principal components contributed the majority proportion for the variance of the dataset while the others significance are very small. While the elbow point for this distribution occurred at the 5th principal components. The curve then flattens out after the 5th principal component providing lesser and lesser eigenvalues for the dataset which indicates lesser significance. A total of 31 principal components are being used for this analysis. There is a downward trend being reflected in the chart which implies that the subsequent components are contributing decreasing proportion of the variance.

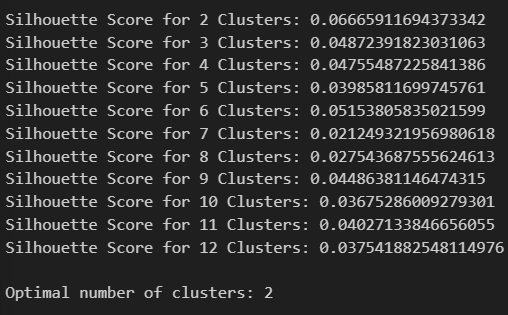
3.2.2 K-modes Clustering

K-means, K-modes and K-prototypes are clustering algorithms in unsupervised machine learning. K-means algorithm clusters data points together based on their distance to the centroid of their clusters. K-means clustering is best at handling low dimensional numerical data. K-modes is best at handling categorical data. While K-prototypes can handle a mixture of both categorical and numerical data. We use the K-modes clustering method for our dataset because all the data in our dataset is categorical data.

Silhouette Score Plot:



The graph shows the silhouette score against the numbers of clusters. The silhouette score measures how similar an object is to its own cluster compared to other clusters. The silhouette scores range from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters. According to the graph, the highest silhouette score when the number of clusters is 2.



This indicates that the optimal number of clusters is 2. After we identify the optimal number of clusters, we use n\_clusters = 2 to fit a new K-modes model.



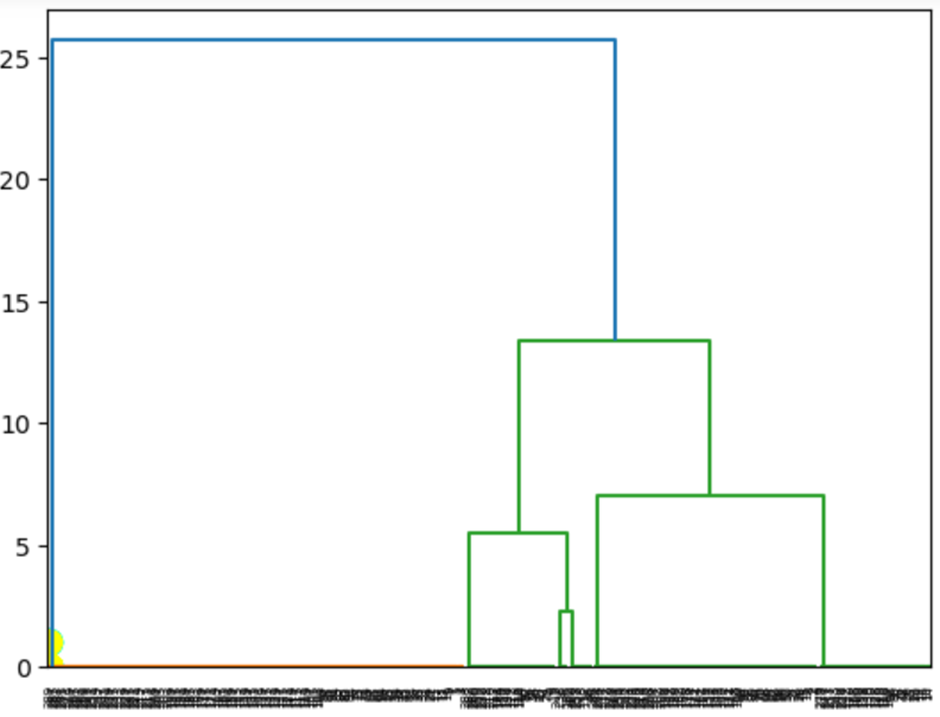
This is the cluster centroids of the optimal number of clusters = 2, showing the mode of the features within each cluster.

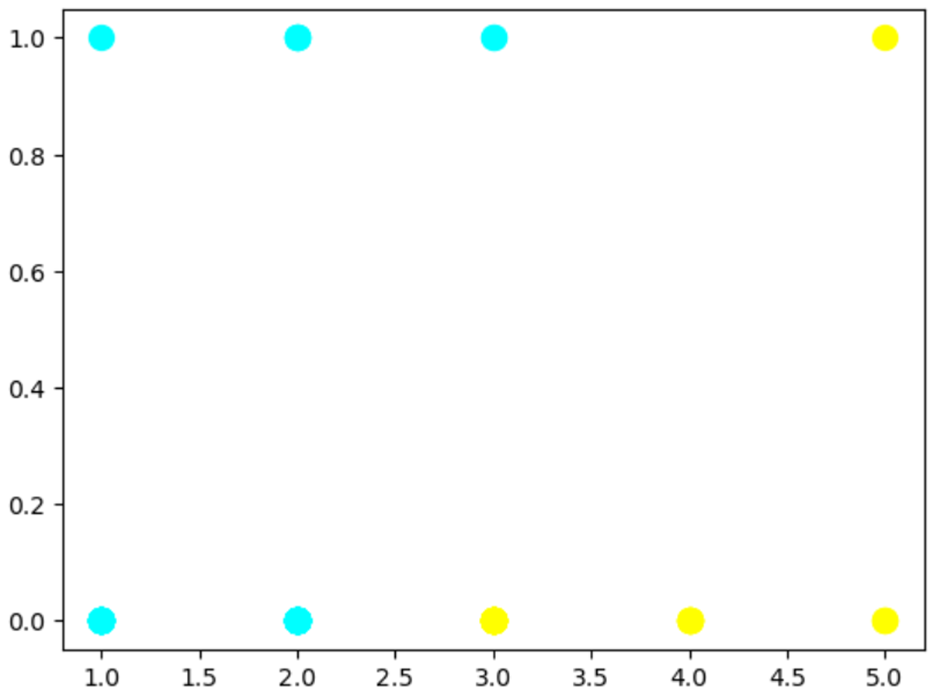


We gain a higher silhouette score for optimal clustering. Silhouette score is close to 0, it can be said that the distance between clusters is not significant.

3.2.3 Hierarchical Clustering

Hierarchical clustering is separating data into groups based on some measure of similarity, finding a way to measure how they're alike and different, and further narrowing down the data. We obtain dendrogram and scatter plot.

 The dendrogram is a visualization that shows the arrangement of the clusters produced by hierarchical clustering. Vertical lines represent clusters that are being joined or formed. The height of the line indicates the distance at which the clusters were joined, reflecting cluster similarity. The taller the line, the more dissimilar the clusters are. Horizontal lines represent the clusters themselves. Different colours represent different cluster groups.

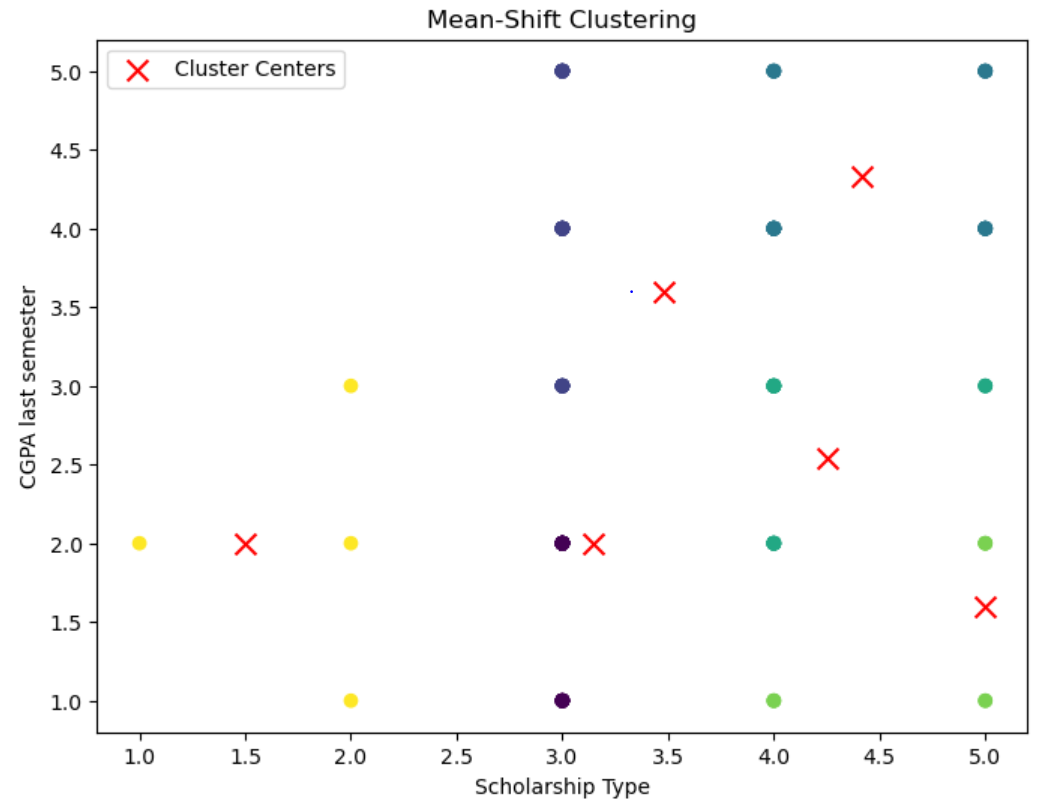


The scatter plot visualizes the clusters formed by the Agglomerative Hierarchical Clustering. The AgglomerativeClustering class from sklearn has been used to fit the model to the data and predict the clusters. Each color on the plot represents a different cluster identified by the algorithm. Each point on the plot represents an observation from the dataset. The position of the point is determined by its feature values, and its color corresponds to the cluster it belongs to.

Interpreting these together, the dendrogram is used to determine the number of clusters and the distances between them, while the scatter plot is used to visualize the distribution and grouping of the data points in the formed clusters. However, there appears to be a mismatch between the number of clusters chosen for the analysis (2, as per the AgglomerativeClustering setup) and the number of clusters visualized (4, as per the scatter plot). This needs to be reconciled to ensure consistency in the analysis.

3.2.4 Mean-Shift Clustering

Mean Shift Clustering is a centroid-based algorithm which works by determining the centroids which is the mean of the points of the given data (scikit-learn, 2007). Mean shift clustering is non-parametric and unsupervised. It works by moving the mean of the data point iteratively to its most condensed state and the mean point of the data set could be noticed easily (Yufeng, 2022).



According to the graph generated, there are 6 clusters formed by sklearn. The bandwidth estimated by sklearn in 1.069. There different colour of points plotted on the graph represents different set of clusters. The red cross represents the cluster centers. The points in the graph is widely spread and a specific dense area is not noticeable.

# **4.0 Conclusion**

4.1 Supervised Machine Learning

By using evaluation metrics, which is a numerical measurement, the model’s performance and efficiency can be evaluated (Mankad, 2020). They are helpful in comparing various models or algorithms by providing insights into the effectiveness of models. Based on the trained supervised Machine Learning models, the metric applied to compare and evaluate the models is kappa value. Kappa coefficient, also named Cohen’s Kappa Score, utilized to measure how well the agreement between two sets of observations (Khan, 2022). Also, it is helpful in examining the performance of classification models, especially the datasets with imbalanced classes. The table below shows the kappa value of each model obtained by different methods.

|  |  |  |
| --- | --- | --- |
| **Hold-Out Method:** |  |  |
|  | **Accuracy** | **Kappa score** |
| Logistic Regression | 0.86 | 0 |
| kNN | 0.86 | 0 |
| Naïve Bayes | 0.86 | 0 |
| Decision Tree | 0.86 | 0 |
| Random Forest | 0.86 | 0 |
| Random Forest with hyperparameter tuning | 0.86 | 0 |

|  |  |  |
| --- | --- | --- |
| **K-fold Cross Validation Method:** |  |  |
|  | **Average Accuracy** | **Average Kappa** |
| Logistic Regression | 0.9379 | 0.1205 |
| kNN | 0.9 | 0.3459 |
| Naïve Bayes | 0.7172 | 0.0787 |
| Decision Tree | 0.9034 | 0.0931 |
| Random Forest | 0.9448 | 0 |

From the observations, for the hold-out method, it was noticed that all models achieved the same accuracy of 0.86 and a kappa score of 0. Hence, the agreement beyond random chance which is measured by kappa is negligible. Therefore, hold-out method which invovles splitting the dataset into a training set and a testing set without further iterations may not provide a robust estimate of model performance.

In the K-fold Cross Validation Method, the accuracy varied across models, with Random Forest achieving the highest average accuracy of 0.9448 and Naïve Bayes achieving the lowest average accuracy of 0.7172. For the average kappa score, Random Forest model demonstrates the poorest performance while k-Nearest Neighbors reveals the best performance. It is indicated by their kappa value, respectively 0.0 and 0.0953. The performance of models can also be arranged under below sequence:

kNN > Logistic Regression > Decision Trees > Naive Bayes > Random Forest

However, there is an interesting phenomenon observed as the kappa value of Random Forest remains 0. Although SMOTE and hyperparameter tuning are carried out, we noticed that the kappa value has no changes. Therefore, it can be said that Random Forest has very low model performance neglected any method used.

Since the kappa value for all the models using the hold-out method is extremely low, we would not consider using them to evaluate the model performance. In conclusion, based on the models trained using k-fold cross validation, the model with the best performance is k-Nearest Neighbors.

4.2 Unsupervised Machine Learning

In K-modes clustering we evaluate the model using silhouette scores. A silhouette score below 0.25 indicates bad clustering. The silhouette score of the optimal number of clusters is 0.09978, which suggests the distance between clusters is not significant. Thus, based on the silhouette score, we can say that the K-modes algorithm is not good at clustering the dataset. However, silhouette score is not the only metric when evaluating the K-modes algorithm. Factors such as the number of clusters, the size and shape of clusters and domain knowledge of the data can affect the outcome. In conclusion, K-modes algorithm is not that suitable for our dataset. Further understanding and deep learning of the data is required to improve the performance of the K-modes model.

On the other hand, principal component analysis (PCA) is also being used to understand the dataset by presenting it in a 2-dimensional space. While we could determine the most important principal components through the scree graph there are also limitations for this method. The results presented by PCA are difficult to interpret as it is very messy, and the principal components are all stacked together. Due to high dimensionality of our dataset, forcing it into a 2D representation would affect the results of the biplot as this dataset require 31 principal components. Therefore, it is concluded that PCA is not suitable for this dataset.

Moreover, the analysis using hierarchical clustering with an intended number of clusters (2) showed a mismatch with the visualized clusters (4) in the scatter plot. To reconcile this, we may need to reconsider the number of clusters based on the data's natural structure and possibly adjust the algorithm parameters for a more accurate representation of the clustering results.

Furthermore, Mean Shift Clustering is used to determine the number of cluster because it is different from k-mean clustering which needed to calculate silhouette score to obtain the number of clusters. Mean shift clustering is useful for a large number of data which has small distance between each other as the mean of each cluster can be easily displayed in the graph. As this dataset has big distance between each other, the clusters are difficult to be differentiated and separated. Therefore, mean shift clustering is not suitable for this dataset.

4.3 Overall Conclusion

By comparing both supervised and unsupervised learning algorithms we used, the result shows that all of the unsupervised learning algorithms that we applied is not perform well on this dataset. Therefore, we can conclude that the best predictive model for this dataset is k-Nearest Neighbors.

REFERENCE

Amazon Web Services, Inc. (n.d.). *What is Hyperparameter Tuning? - Hyperparameter*  *T* *uning Methods Explained - AWS*. [online] Available at: [https://aws.amazon.com/what-](https://aws.amazon.com/what-is/hyperparameter-tuning/#:~:text=Hyperparameters%20directly%20control%20model%20structure) is/hyperparameter- t u n i n g / # : ~:text=Hyperparameters%20directly%20control%20model%20structure.

‌Brownlee, J. (2020). *SMOTE for Imbalanced Classification with Python*. [online] Machine Learning Mastery. Available at: [https://machinelearningmastery.com/smote-](https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/) oversampling-for-imbalanced-classification/.

javaTpoint (2021). *Machine Learning Decision Tree Classification Algorithm - Javatpoint*. [online] www.javatpoint.com. Available at: [https://www.javatpoint.com/machine-](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm) l earning-decision-tree-classification-algorithm.

‌

Khan, M.B. (2022). *Cohen’s Kappa Score*. [online] Medium. Available at: [h](https://bootcamp.uxdesign.cc/cohens-kappa-score-33a0710b2fe0#:~:text=Cohen) t t ps://bootcamp.uxdesign.cc/cohens-kappa-score-33a0710b2fe0#:~:text=Cohen

Mankad, S. (2020). *Evaluation Metrics in Machine Learning*. [online] Analytics Vidhya. A vailable at: [https://www.analyticsvidhya.com/blog/2020/11/a-tour-of-evaluation-](https://www.analyticsvidhya.com/blog/2020/11/a-tour-of-evaluation-metrics-for-machine-learning/) m etrics-for-machine-learning/.

‌Szeghalmy, S. and Fazekas, A. (2023). A Comparative Study of the Use of Stratified Cross- Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced L earning. *Sensors*, 23(4), p.2333. doi:https://doi.org/10.3390/s23042333.

Scikit-learn. (2007). Sklearn.cluster.MeanShift. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html#:~:text=Mean%20shift%20clustering%20aims%20to,points%20within%20a%20given%20region.

‌

Yufeng. (2022). Understanding Mean Shift Clustering and Implementation with Python.