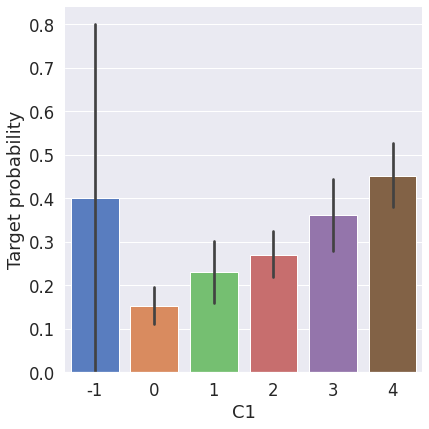
**Data Mining COMP5009/COMP3009**

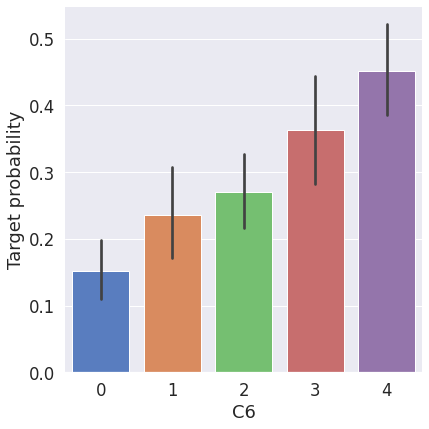
**1. Introduction**

Here, we were to perform a predictive analytics task given data containing a total of 1100 samples. The first 1000 samples had already been categorised into two classes (0 and 1), with an objective to predict the class labels of the last 100 samples associated with IDs from 1001 to 1100.

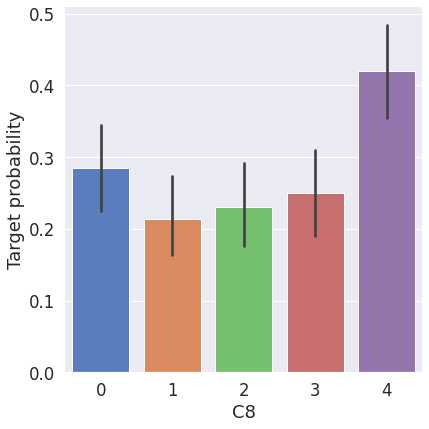
**1.1 Data exploration**

****

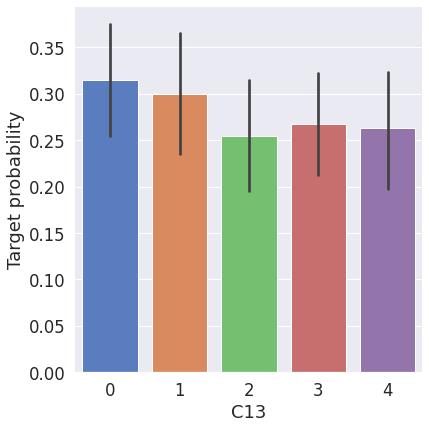
Above graph shows a somewhat positive relation between the C1 feature with the probability/class

Below graph shows a positive relation between the C6 feature with the probability/class****

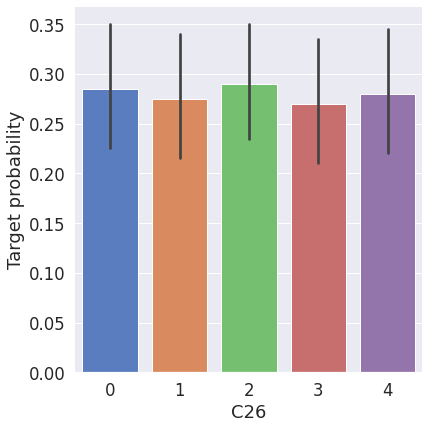
Below graph shows a non-existent relation between the C8 feature with the probability/class

****

Below graph shows a somewhat negative relation between the C13 feature with the probability/class

****

Below graph shows a non-existent relation between the C26 feature with the probability/class



**2. Data Preparation**

The very first task, was to ensure that the data was of good quality for the classification process; this included getting rid of irrelevant data, removing duplicates, standardizing the features, solving for the outliers, dummy-encoding the data, solving for missing values where necessary,

**2.1 Irrelevant attributes**

A number of features we found to be irrelevant in the classification study including columns; C28 and C29 which only less than 0.5% entries out of the required 1100 entries. This negligible participation rendered both columns irrelevant in the study. High correlation was also found between features C5 and C16, owing to their similar entries, rendering C16 column useless given that no additional information was being derived from it.

**2.2 Missing values**

Here, a number of columns including C1 and C2 had a few missing values. After doing a correlation heatmap, none of the variables had a significant correlation with these features, hence the need to replace the null values with one of the central tendency measures. However, given the skewness in the datasets, the median was the better candidate for the same, since it’s more robust to outliers unlike the mean.

**2.3 Duplicates**

While checking for the duplicates, we found columns C15, C17, C21 and C22 with the maximum number of duplicates. Infact, the columns had one entry being duplicated across all the rows for characters 1, F, T and 0 respectively. This rendered all these columns useless in the classification task, as no information was actually being gotten.

**2.4 Data types**

For ease of processing data by our classifiers during the training process, data encoding is of

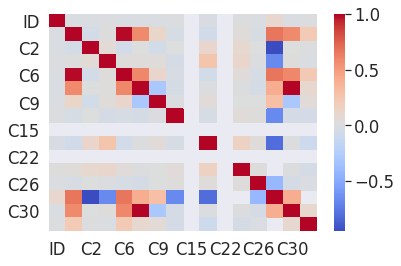
significance in ensuring this goal is achieved and better results are achieved. Here, we applied dummy variables to the categorical data to get numerical binary data of 1 if the right class and 0 elsewhere. This was applied for columns C3, C5, C10, C14, C18, C19, C23, C25, C27, C31 and C32. The same was also applied for column C12 to quantify the yes/no entries into 1/0 binary system (where yes-1 and no-0). This comes in handy during the training, as our classifiers don’t work with strings or non-numerical data.

**2.5 Scaling and standardization**

From the description of the datasets, it’s evident that the features; C1, C6, C8, C13, C20, C26 and C30 are non-normal as characterized by the very high standard deviation values. This prompts the need for rescaling the datasets to ensure classification isn’t impacted by the high volatile data. We applied the standard and robust scalers. Normalizing these features was necessary to avoid any bias as a result of variation in the scaling units hence contribution to the analysis by a variable.

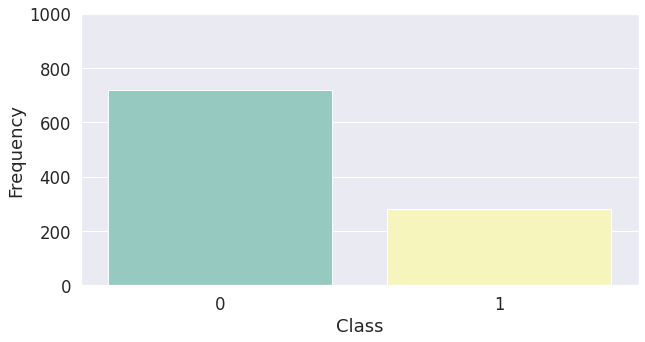
**2.6 Feature/Attribute Selection**

From the correlation matrix, none of the features happened to be highly correlated to another feature, hence retained all the features other than the ones already gotten rid of. This process was to ensure no case of multicollinearity that would induce bias into our classification process.



**2.7 Data Imbalance**

As shown by the graph below, there is evidently an imbalance between the two classes under which the variables are being grouped with a 72% of the population belonging to class ‘0’ and the remaining 23% in class ‘1’. This automatically has an effect on the classification, as the features are better trained to correctly classify an item of class ‘0’ as opposed to the other class ‘1’. This prompted the need for a resampling. We did both an undersampling and SMOTE oversampling techniques to check which would perform better than its counterpart.



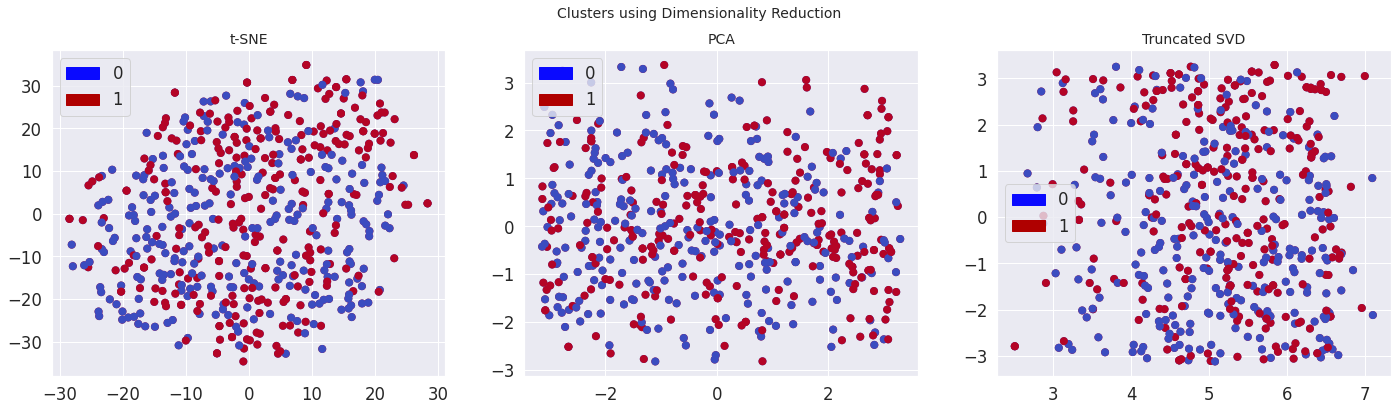
**2.8 Training-test splits**

Although we are splitting the data when implementing Random UnderSampling or OverSampling techniques, we want to test our models on the original testing set not on the testing set created by either of these techniques. We implemented the train\_test\_split model from scikit learn and the stratified shuffle split to randomize the splits. The outcome is a more balanced datasets for both classes as shown below;



**3. Data Classification**

We applied four simple classifiers including; the logistic regression, the SVM classifier, decision trees and the k-Nearest Neighbours. After the undersampling technique is effected, we see that the t-SNE algorithm can still pretty accurately cluster the 0 and 1 classes in the dataset. Although the subsample is pretty small, the t-SNE algorithm is able to detect clusters pretty accurately in every scenario (we shuffle the dataset before running t-SNE). This indicates that further predictive models will perform pretty well in separating default from non-default cases. This is further illustrated in the representation below;



**4. Prediction and Findings**

**4.1 Accuracy Scores**

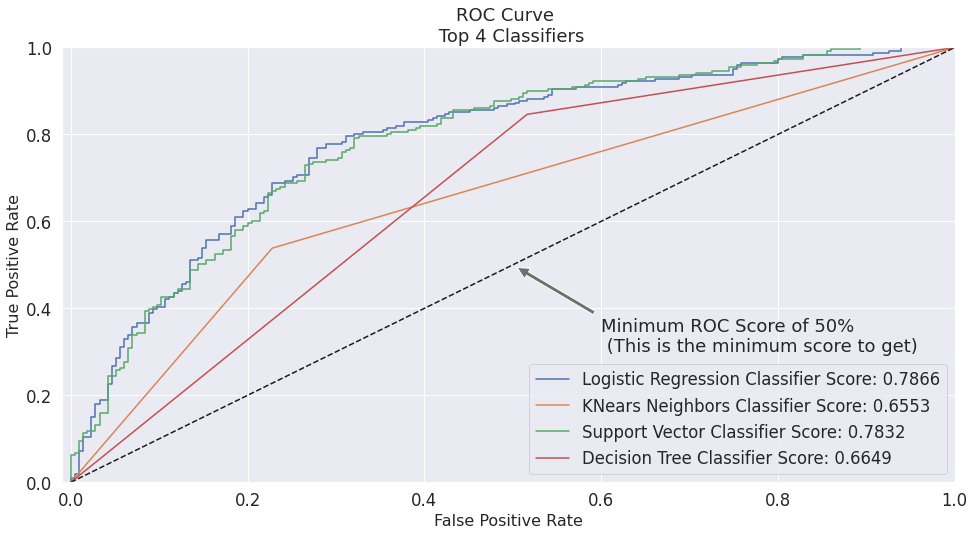
We found the highest accuracy score with the Logistic regression classifier with a 74.0% score, followed by the SVC at 73.0%, then kNN and decision tree classifier both at 64.0%.

**4.2 Cross-Validation Scores**

Similarly, we found the highest cross-validation score with the Logistic regression classifier at 74.31% score, followed by the SVC at 73.62%, then the decision tree classifier at 66.74% and finally, the kNN with a 65.37% score.

**4.3 ROC\_AUC Scores**

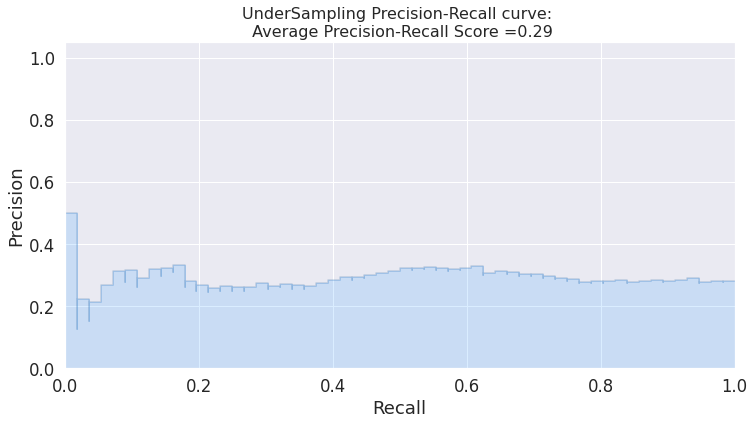
Here, we found the highest ROC-AUC score with the Logistic regression classifier at 0.7866 score, followed by the SVC at 0.7832, then the decision tree classifier at 0.6649 and lastly, the kNN classifier at 0.6553. The same is shown graphically below;



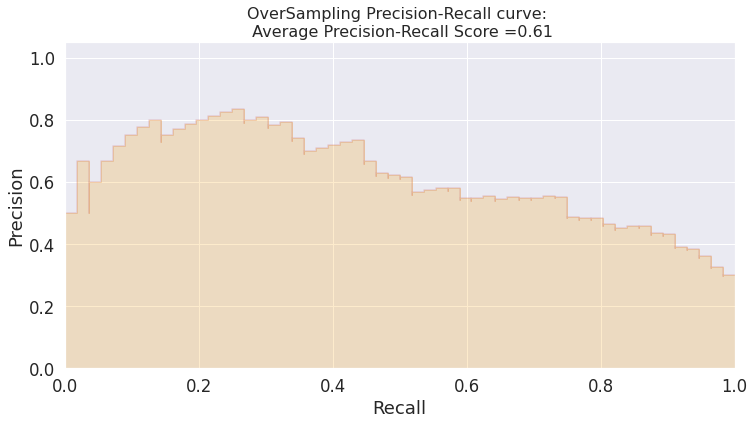
**4.4 Oversampling Versus Undersampling**

Here, we checked the performance of both resampling techniques in terms of precision and recall; the SMOTE technique achieving a higher score at 0.61 as opposed to its counterpart at a mere 0.29 as shown below;

**i) Undersampling technique**



**ii) SMOTE technique**

****

**Technique Score**

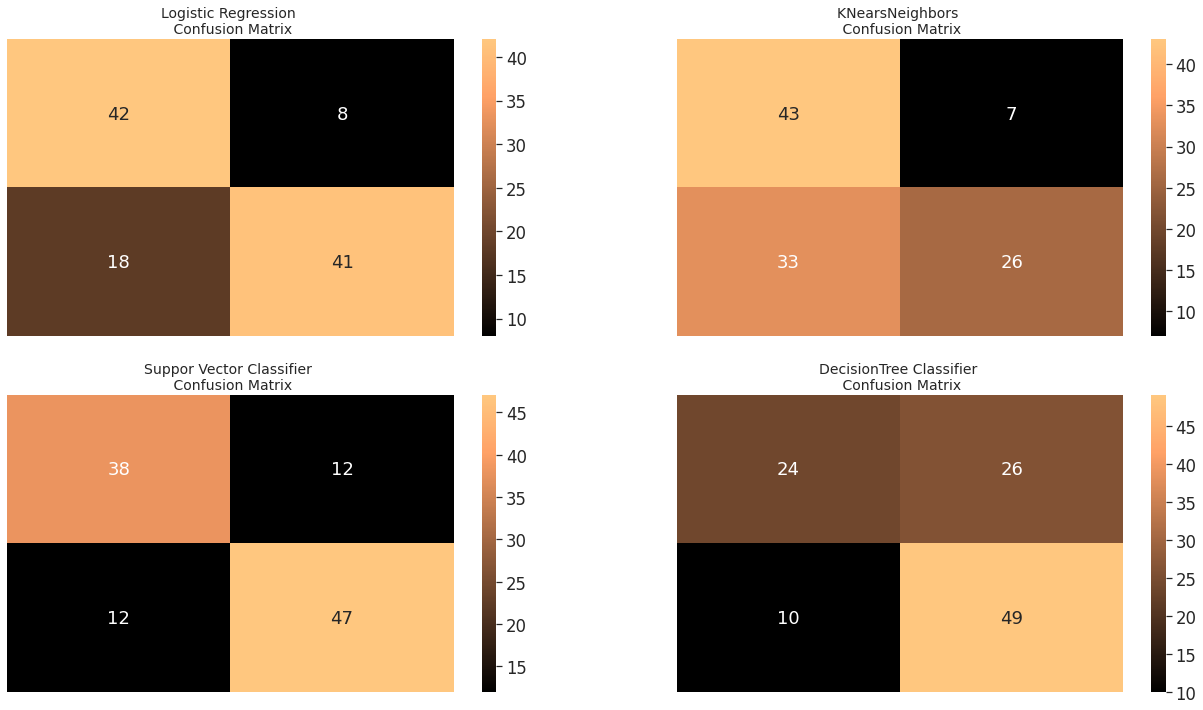
0 Random UnderSampling 0.761468

1 Oversampling (SMOTE) 0.750000

**Top 2 Classifiers;**

We evaluated the final performance of the classification models in the random undersampling subset.The models that performed the best were logistic regression and support vector classifier (SVM) as seen from above score results.

i) Confusion Matrix



From the confusion matrices above; it’s further evident that the logistic regression and SVC classifiers did good by the high numbers of true positives (42 and 38 respectively) and true negatives (41 and 47) respectively.

ii) Classification Report

Logistic Regression:

precision recall f1-score support

0 0.70 0.84 0.76 50

1 0.84 0.69 0.76 59

accuracy 0.76 109

macro avg 0.77 0.77 0.76 109

weighted avg 0.77 0.76 0.76 109

The f1-score for classes 0 and 1 are at 0.76 for both classes for the logistic classifier.

Support Vector Classifier:

precision recall f1-score support

0 0.76 0.76 0.76 50

1 0.80 0.80 0.80 59

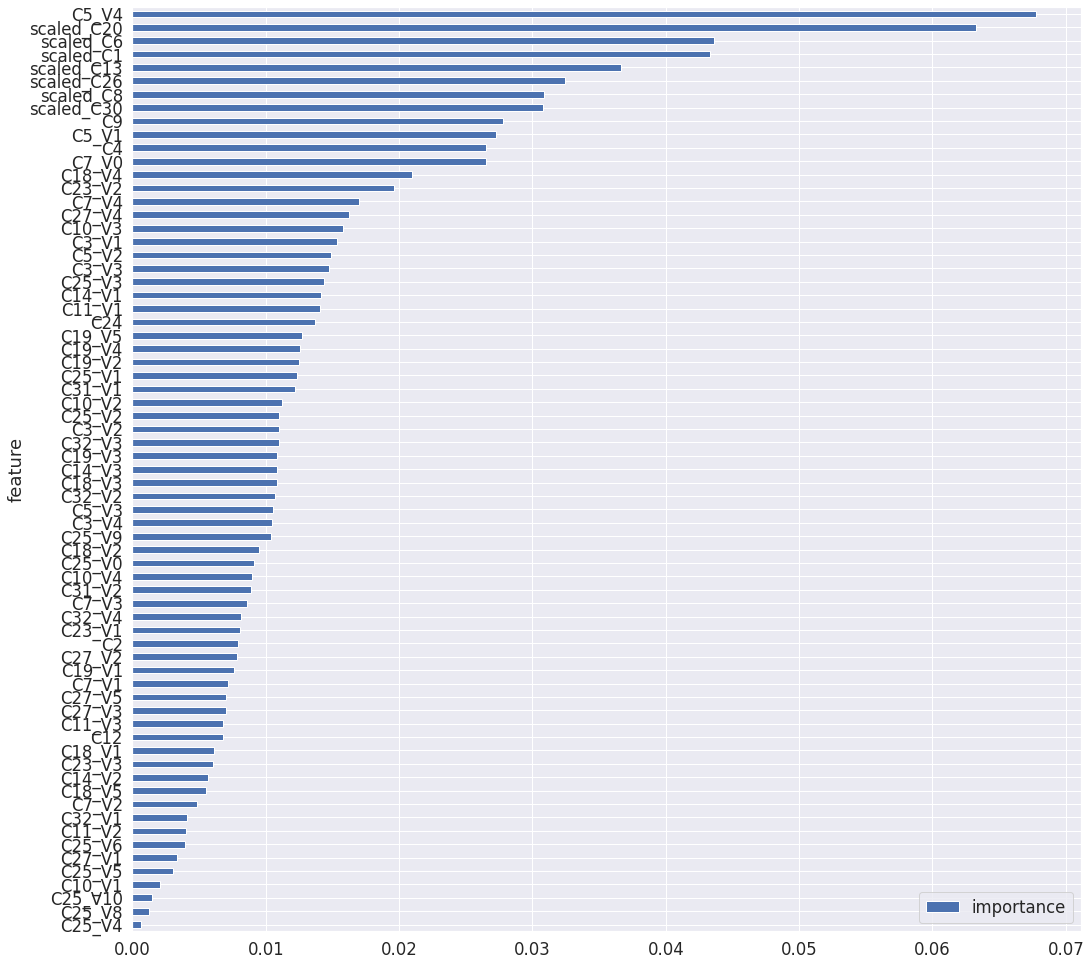
accuracy 0.78 109

macro avg 0.78 0.78 0.78 109

weighted avg 0.78 0.78 0.78 109

The f1-score for classes 0 and 1 are at 0.76 and 0.80 for classes 0 and 1 for the SVC.

**Feature importance;**



**Conclusion**

From the above findings and discussions, it is evident that the logistic and SVC classifiers did a pretty good job at predicting the classes for the test datasets with the most important features in the study comprising of C1, C5, C8, C13, C20, C26 and C30.