STAT5003_14_Credit_Card_Analysis

STAT5003 group project 14

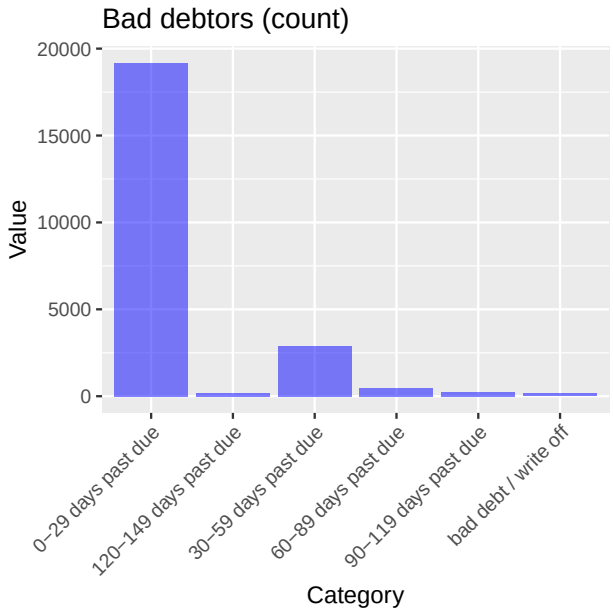
2023-10-23

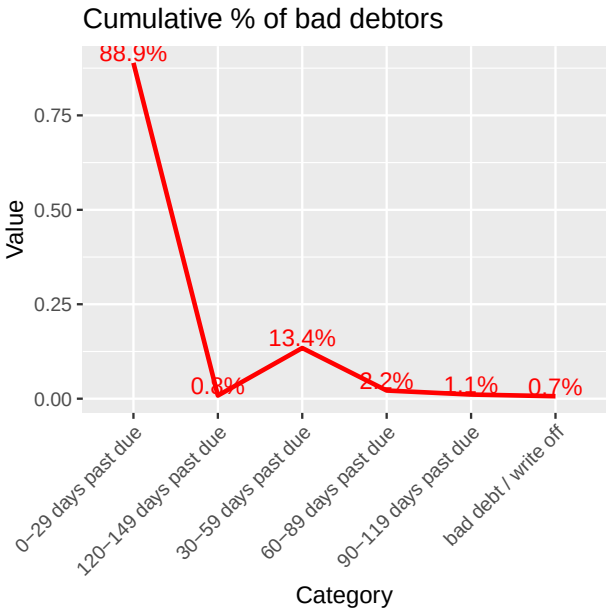
OVERVIEW

- **Problem statement:** Commercial banks need to continually observe the impact of customer's work and life stability on credit card defaults, which is costly to maintain (Li et. al., 2019). [Traditional] regression functions on demographic variables and credit card features yield limited explanatory power compared to other factors such as attitude variables and personality variables (Wang et. al, 2011). Previous attempts at applying machine learning to classify card applicants into different credit limits has achieved acceptable results (82% predictive accuracy of credit card defaulters) (Leong et. all., 2019).
- **Goal:** The goal of this exercise is to apply machine learning methods to define 'bad debtors' based on their credit history and predict credit defaulters based on a limited set of data available at point of application. This exercise also includes 'length_of_relationship' as a factor to test it's relevancy against demographic data received at point of application. Dataset can be accessed here.
- **The dataset is split into two files**: **credit_record** contains 46 thousand unique customers IDs which tracks their credit status overtime. Customers are classified into different loan status groups (e.g. 'paid off that month', '1-29 days past due', etc.).**application_record** contains collected information at the initial credit card application, such as ID, gender, occupation, number of children, marital status, etc. Both files are connected by the feature **ID**.

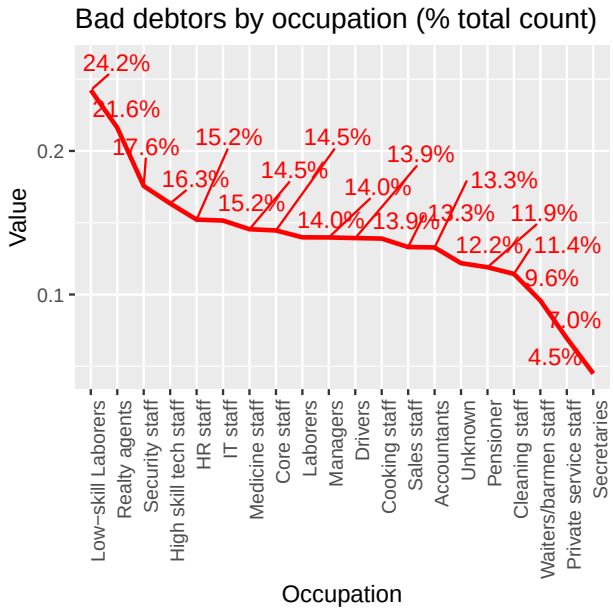
FEATURE ENGINEERING

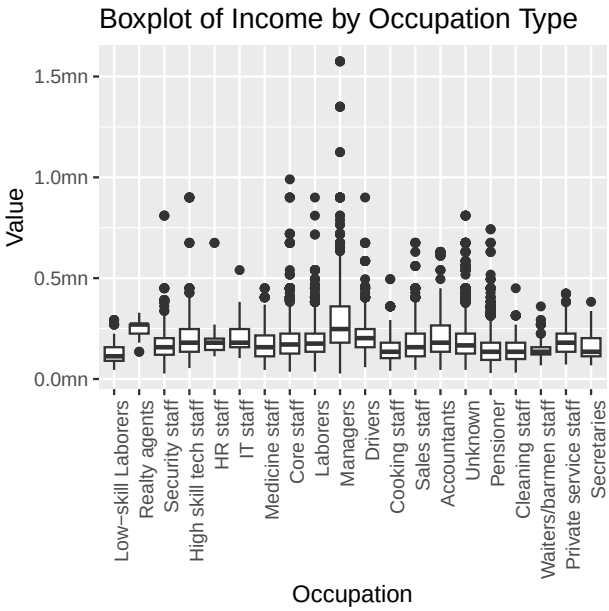
- **Data analysis and cleaning:** the dataset was screened for nulls, blanks, and duplicates and further trimmed to only include IDs with 20+ months banking history. 88.9% of debtors has more than 0 days of delinquency which could very well be an honest mistake and not representative of their ability to service their payments. Thus we defined **30+ days of delinquency as our target feature, denoted as class 1. The resulting dataset contained 21,575 customers of which 2,895 customers (c.13%) are in class 1**.



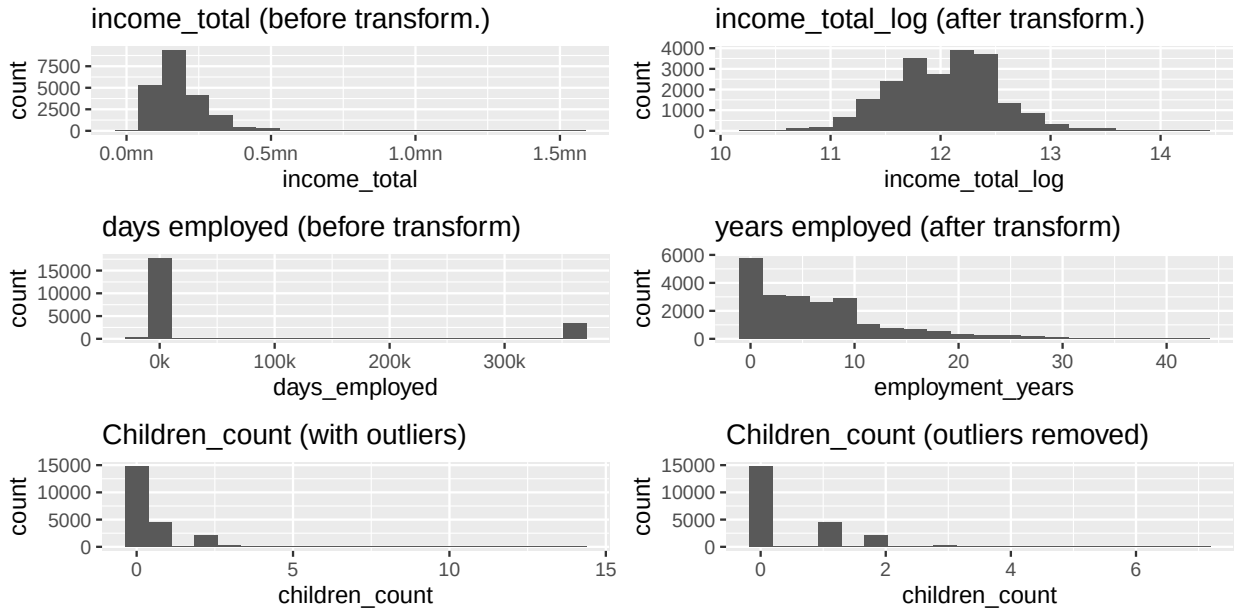


- Delinquency rate by occupation type shown below. We can see that delinquency rate does not necessarily increase with lower wages and some occupation has large variability in total income.

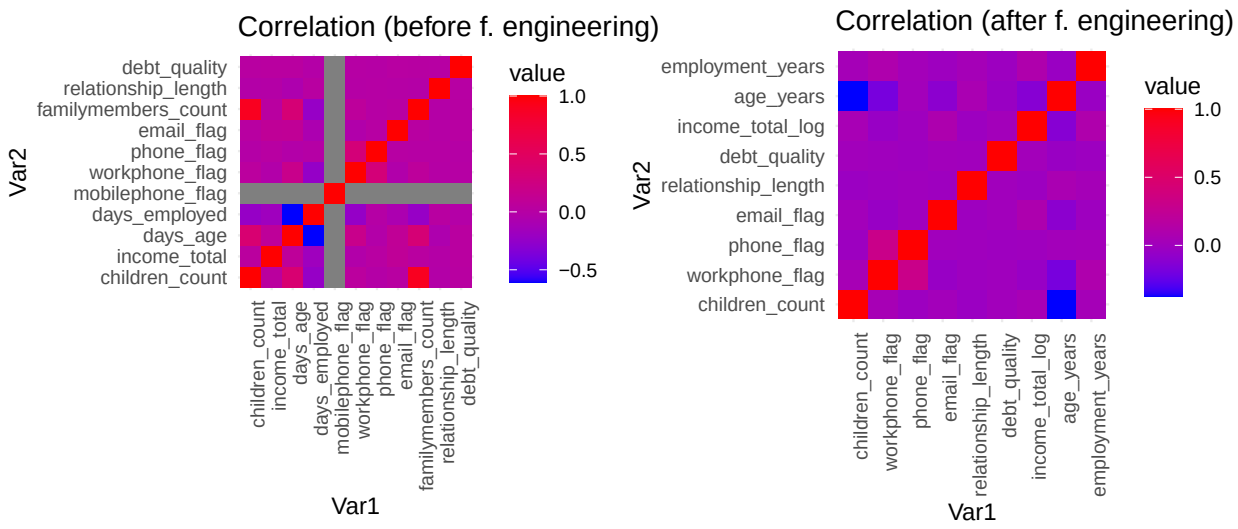




- Selected feature distributions shown below to show the impact of data cleaning/transformation.



- Correlation plots shown below to show the impact of data cleaning/transformation. Improvement in correlation between features can be seen by fewer red boxes on the right.



PRE-PROCESSING

Pre-processing used for this analysis are described below:

- **Min-max scaling:** applied to numerical data (excluding the target) to remove bias from large values.
- **One-hot-encode:** applied to categorical data (excluding the target) to create binary representation of each categorical value. This is done as some machine learning models require numerical input.

- **Synthetic Minority Over-sampling Technique (SMOTE):** applied to the training set to prevent bias towards majority class due to data imbalance. SMOTE works by creating synthetic data points between randomly-chosen k-nearest neighbor at random distances in the feature space. In practice, SMOTE should only be used on training data to prevent leakage.
- **UpSampling:** applied to the training set to address significant imbalances between classes. upSample operates by increasing the number of instances of the minority class through duplication of random instances.
- **Principal Component Analysis (PCA)**: applied to the training set to reduce data dimensionality used in Support Vector Machine (SVM). As discussed below, SVM works by building a decision boundary/hyperplane that best separates features. This process can be overly complicated on overly complex dataset.

CLASSIFICATION METHODS

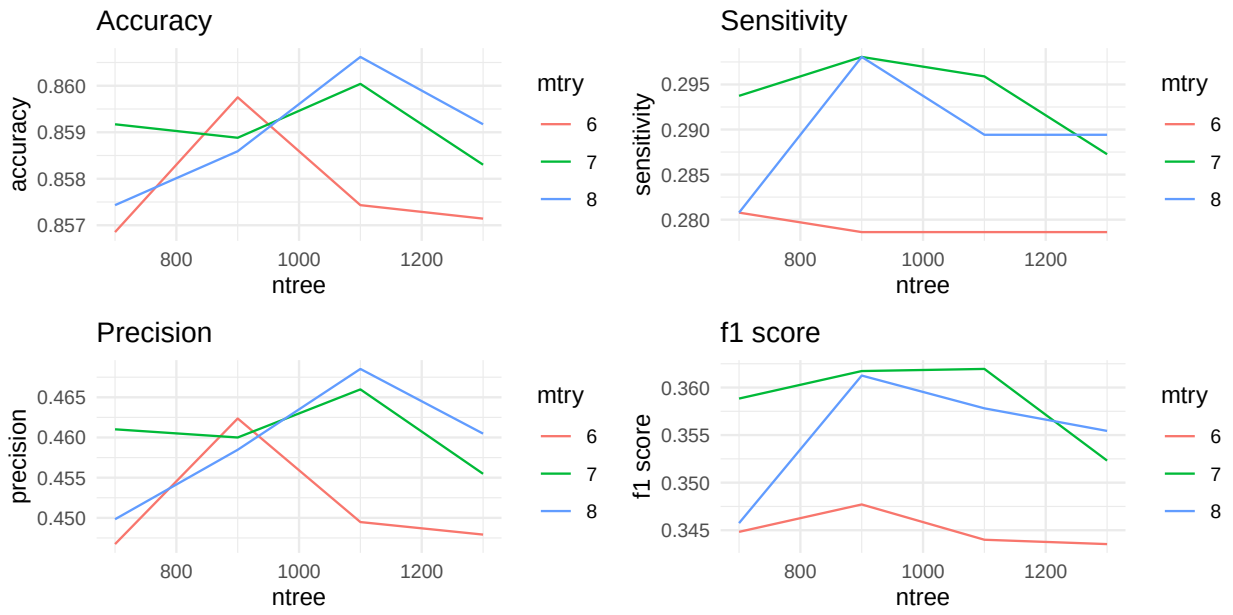
Five algorithms are used in this exercise and discussed below. Please see results on testing scores in the 'RESULTS' section.

1. Random Forest

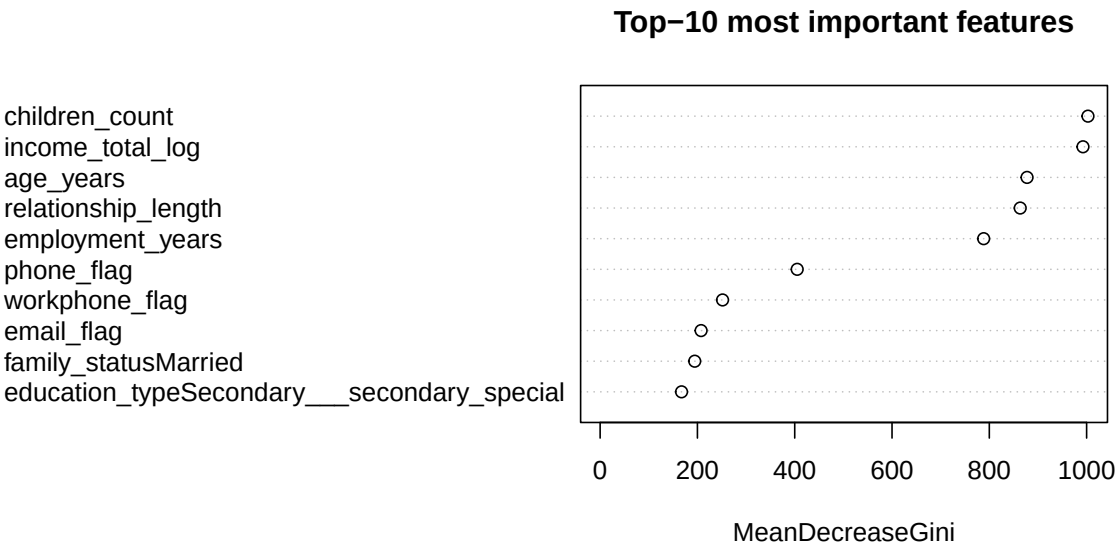
- **Description:** Random forest is an ensemble method which combines multiple decision trees to create predictions. Random forest is known for its robustness against overfitting as it takes random features in each of the individual decision trees and takes an average/vote of these trees for a combined result.
- **Design:** 1. We applied min-max scaling, one-hot-encoding and SMOTE to the dataset. 2. Grid search was applied on **'ntree'** and **'mtry'** to identify the model with the best **f1-score**, which gives a balance between precision (minimizing false positives) and recall (minimizing false negatives). 3. We also identified most important features to understand what drives bad debt. The best model was tested against the test dataset which is presented in the 'classification performance evaluation' section.
- **The best model parameters by accuracy score shown below:**

```
## Best model parameters (maximizing f1 score):
## - ntree:  1100
## - mtry:  7
##
## Training scores for the best model:
## - accuracy:  0.8600406
## - precision:  0.4659864
## - sensitivity:  0.2958963
## - specificity:  0.9474565
## - balanced accuracy:  0.6216764
## - f1 score:  0.3619551
```

- Grid search results shown below. Although maximum accuracy is high (0.86), sensitivity and precision scores are low, suggesting that the model has high counts of false negatives and false positives.



- Top-10 most important features (by Gini impurity) shown below. We can see that the top-5 features drive the prediction of our random forest model.



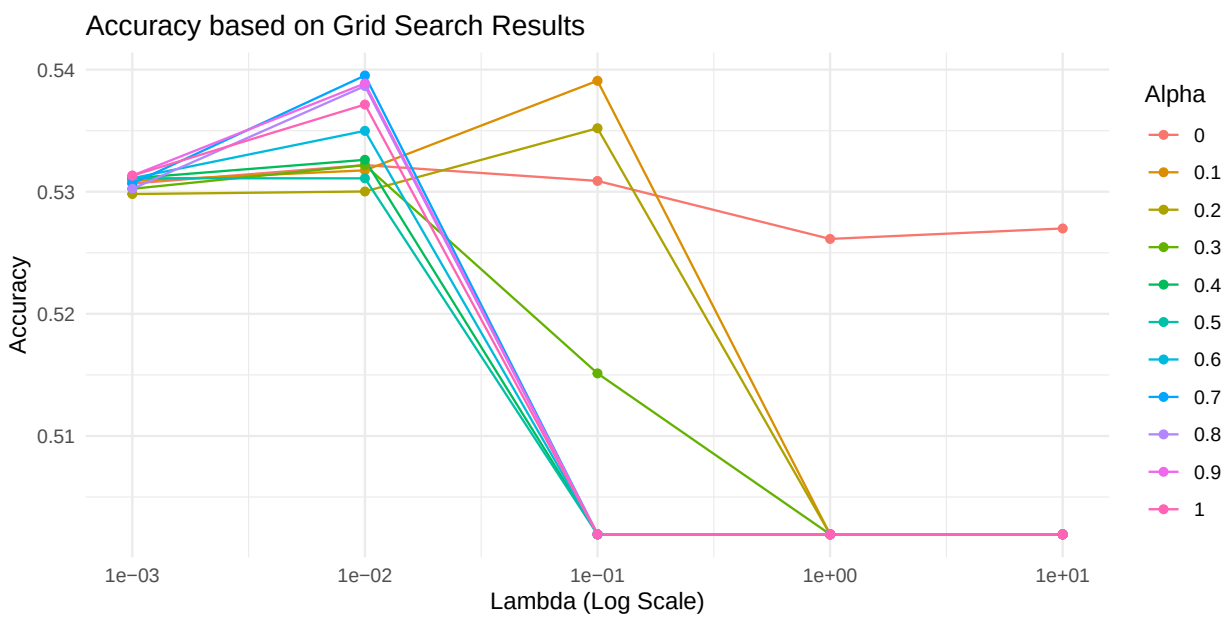
2. Logistic Regression

- **Description:** Logistic Regression is a statistical model used for predicting the probability of occurrence of an event by fitting data to a logistic curve. It is particularly adept for binary classification problems. One of the inherent strengths of logistic regression is its ability to estimate the probability of an event, allowing for a nuanced decision-making process. When combined with regularization techniques such as L1 (Lasso) and L2 (Ridge), the model can mitigate overfitting, especially in scenarios with a large feature set.

- **Design:** (i)Hyperparameter Tuning with Cross-Validation: A comprehensive grid search, coupled with 5-fold cross-validation, was conducted over hyperparameters alpha (indicating the type of regularization) and lambda (denoting the regularization strength). Cross-validation ensures a robust estimation of model performance across different subsets of the training data, preventing overfitting and providing a more generalized model.(ii) Model Training and Evaluation: The logistic regression model was trained using the best hyperparameters derived from the grid search. Performance metrics, including accuracy, precision, sensitivity, specificity, and F1 score, were evaluated on both the training and test datasets to understand the model's predictive power comprehensively.(iii) Rationale for Metric Selection: Accuracy was chosen as the primary metric for hyperparameter tuning and model evaluation due to several factors:The preprocessing steps ensured a balanced representation of both classes, making accuracy a relevant and reliable performance measure. Logistic regression's output of probabilities necessitates a threshold for classification. With a balanced dataset, accuracy provides a clear and direct measure of the model's correctness in predictions. Among various metrics, accuracy is intuitive, straightforward to interpret, and computationally efficient, making it a practical choice for evaluating model performance.
- **The best model parameters by f1 score shown below:**

```
## Training scores for best model (by accuracy score) as follow:
## - accuracy:  0.5514039
## - precision:  0.5445352
## - sensitivity:  0.6071119
## - specificity:  0.4961274
## - balanced accuracy:  0.5516196
## - f1:  0.5741234
```

- Accuracy scores for different alpha and lambda values shown below:



3. Linear discriminant analysis (LDA)

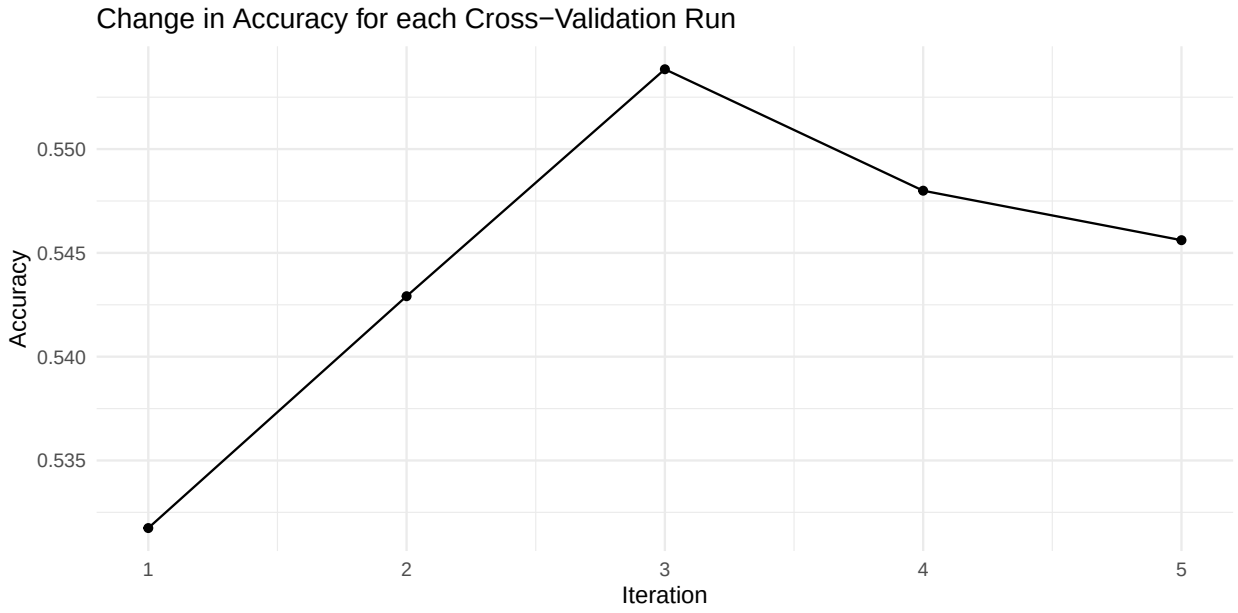
- **Description:** LDA is a dimensionality reduction technique that works by finding the linear combination of features that best separates multiple classes in data. It is very interpretable, and very robust

against overfitting since it best estimates the discriminant boundaries using covariance information. LDA does not require any hyperparameter tuning, since the lda() function in the MASS package tunes parameters itself, using the covariance matrices of the input data. This means that for this technique, only cross validation has been done to ensure robustness.

- **Design:** (i) we applied min-max scaling, one-hot-encoding and SMOTE to the dataset, (ii) lda() was run with 5 fold cross-validation, (iii) model with best accuracy score is chosen since : 1) The SMOTE pre processing done means that the classes are not too imbalanced, hence accuracy becomes a relevant metric , 2) LDA works in such a way that it uses correlation matrices to find the best results, hence accuracy is generally the more useful metric than sensitivity or precision since this mitigates a lot of the imbalance and 3) accuracy is the easiest metric to interpret and it is the most computationally efficient to calculate. (iv)the best model was tested against the test dataset which is presented in the 'classification performance evaluation' section.
- **The best model parameters by accuracy score shown below:**

```
## Accuracy on run 1 : 0.531744312026002
## Accuracy on run 2 : 0.542912873862159
## Accuracy on run 3 : 0.553846153846154
## Accuracy on run 4 : 0.547995666305525
## Accuracy on run 5 : 0.545612134344529
```

- Accuracy scores shown below:

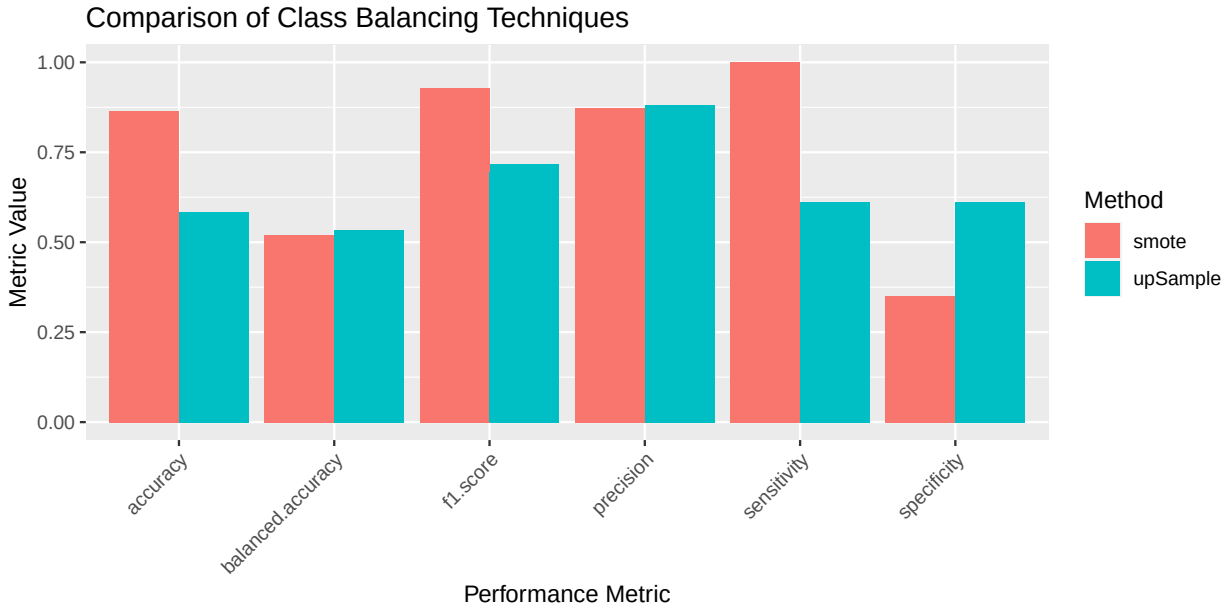


4. AdaBoost

- **Description:** Ada is an ensemble learning algorithm that builds a strong classifier by combining multiple weak learners, in this case decision trees. Ada works by focusing on instances of the dataset that are most difficult to classify by adaptively adjusting weights during successive training rounds.
- **Design:** (i) applied mix-max scaling, one-hot encoding (ii) data was oversampled using upSample and SMOTE (iii) grid search was run on iterator, loss, maxdepth values as well as the two oversampling techniques (iv) the results, encompassing accuracy, precision, sensitivity and F1-score, were compiled into overall.results.df

- **The best model parameters by accuracy score shown below:**

```
## The AdaBoost model is optimized for maximum balanced accuracy.
##
## Maximum balanced accuracy on validation set: 0.5318473 with the following parameters:
## - loss function:  huber
## - max depth:  5
## - iterations:  60
## - balancing technique:  upSample
##
## Other validation scores for the best model:
## - accuracy:  0.5575196
## - precision:  0.8790867
## - sensitivity:  0.5669344
## - f1 score:  0.6893184
```



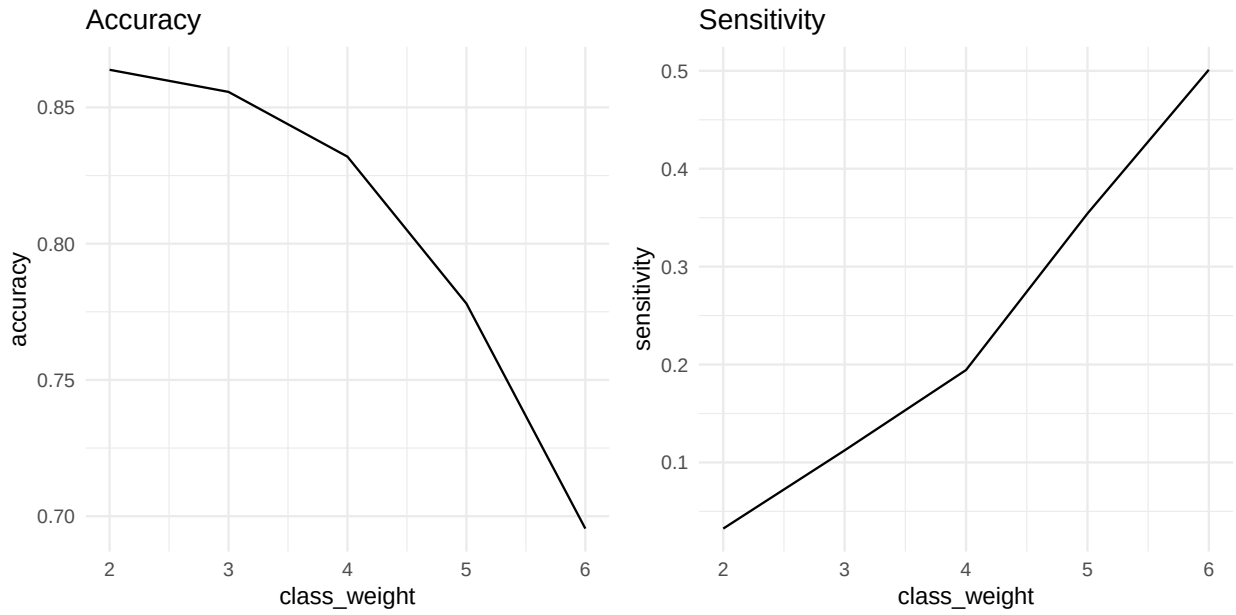
5. Support vector machine

- **Description:** Support Vector Machines (SVM) are a set of supervised learning methods used for classification, regression, and outliers detection. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other. It is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.
- **Design:** (i) we applied min-max scaling and one-hot-encoding to the dataset, (ii) Principal Component Analysis (PCA) is a commonly used data preprocessing technique, especially when dealing with high-dimensional data algorithms like Support Vector Machines (SVM). we use PCA to preprocess both train and test data, (iii) the grid search technique was applied on **'class.weights'** to find the **best f1 score**. F1 score was chosen to achieve a good balance between precision (minimizing false positives) and recall (minimizing false negatives). Accuracy was not maximised as the dataset is imbalanced, (iv) we tested the best model against the test dataset which is presented in the 'classification performance evaluation' section.

- **The best model parameters by f1 score shown below:**
- Plots of results shown below:

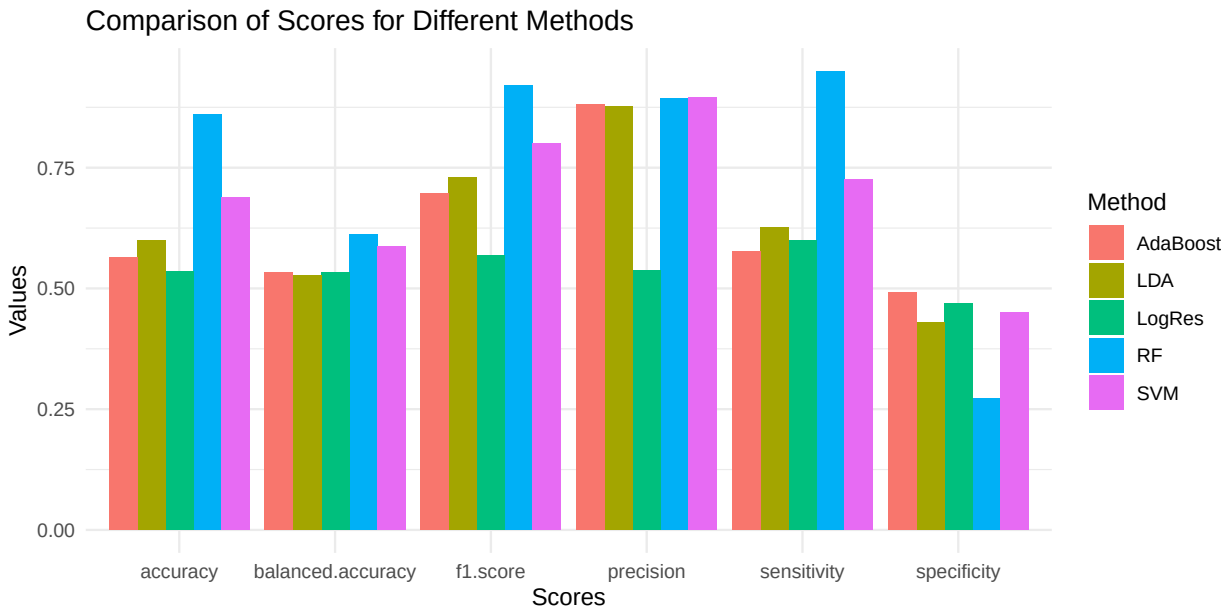
```
## The SVM model is optimized for maximum f1-score:
## - best class weights:  1 6
## Training scores for the best model:
## - accuracy:  0.6954506
## - precision:  0.2205323
## - sensitivity:  0.5010799
## - specificity:  0.7255689
## - balanced accuracy:  0.6133244
## - f1 score:  0.3062706
```

- Plots of results shown below:



RESULTS

Test results are displayed below:



Since our dataset is imbalanced, with class 1 accounting for only c.16% of data points, the team has chosen measures which focuses on maximizing true positives and minimizing false positives and false negatives: + **precision** (TP / (TP+FP)): high precision implies low proportion of false positives. All models aside from logistic regression performed relatively well (87%+) which means they were able to correctly identify actual bad debtors without being 'too strict' whereby not too many good debtors were mistakenly flagged as bad debtors. + **sensitivity** (TP/(TP+FN)): high sensitivity implies low proportion of false negatives. Random forest and support vector machine scored higher (99%+) than the rest which means they were able to correctly identify almost all bad debtors in the test set. + It is worth noting that all models scored lower in **specificity** (TN/(TN+FP)), a high specificity score implies that the model performs well the model can differentiate good debtor from the rest

Final model recommendation: We recommend Random Forest (RF) for this analysis, though it still has its limitations, as shown by its specificity score (0.41, caused by moderately high false positives). This implies that the RF model is 'too strict' : where some good debtors are also flagged as bad debtors. This limitation may be caused by data imbalance which has been minimized by SMOTE but still presents some bias. We are comfortable with this flaw as having false positives carry lesser economic impact than having false negatives (e.g., lost revenue instead of bad debt on the banking book).

DISCUSSION

Conclusion: [.]

Improvements: To improve model performance, additional data may be needed (e.g. higher volume of customer data and expanding features collected at application). Other useful features that may be collected include information on wealth level, history of indebtedness at other financial institutions, criminal records, etc. If we had a greater amount of resources, perhaps a deep learning model could have been trained on a significantl large enough dataset to ensure better results, and also ensure that it 'learns' from the data such that it can handle outliers on its own and be used in multiple use cases and more extensively.

REFERENCES

Li, Y., Li, Y., & Li, Y. (2019). What factors are influencing credit card customer's default behavior in China? A study based on survival analysis. Physica A: Statistical Mechanics and Its Applications, 526,

120861. https://doi.org/10.1016/j.physa.2019.04.097

Leong, O. J., & Jayabalan, M. (2019). A Comparative Study on Credit Card Default Risk Predictive Model. Journal of Computational and Theoretical Nanoscience, 16(8), 3591–3595. https://doi.org/10.1166/jctn.2019.8330

Wang, L., Lu, W., & Malhotra, N. K. (2011). Demographics, attitude, personality and credit card features correlate with credit card debt: A view from China. Journal of Economic Psychology, 32(1), 179–193. https://doi.org/10.1016/j.joep.2010.11.006

CONTRIBUTION STATEMENT

All members contribute equally on all aspect of this project.