Universidade de Lisboa

Instituto Superior Técnico

Bologna Master Degree in Mechanical Engineering

Advanced Automation

Data Analysis – Loan Approval

Faculty:

João Miguel da Costa Sousa (Senior Lecturer)

Bernardo Marreiros Firme

Diogo Miguel Ferreira de Oliveira

Authors:

Ricardo Simões, 110917

João Marreiros, 110932

Francisco Carvalho, 111000

Group: 10

8th January 2024

1. Project Overview

The main goal of this project is to study and develop a predictive model for loan approval based on historical data, using various machine learning algorithms such as Logistic Regression, Support Vector Machines (SVM) Classification, Random Forest Classification, Gradient Boosting Classification, and Neural Networks Classification.

The project aims to analyse a dataset that includes information about loan applicants, such as the number of dependents, education level, annual income, loan amount, loan term and other relevant factors. Various machine learning techniques will be used to train models that can accurately predict whether a loan application is likely to be approved or denied.

2. Data and Preprocessing

The selected dataset for this project contains the following parameters:

- **loan_id:** Unique identifier for each loan;
- **no_of_dependents:** Number of dependents of the applicant;
- **education:** The applicant's level of education;
- **self_employed:** Indicates whether the applicant is self-employed;
- **income_annum:** The applicant's annual income;
- **loan_amount:** Amount of the loan requested;
- **loan_term:** Loan term in months;
- **cibil_score:** Applicant's CIBIL credit score (Credit Information Bureau (India) Limited Score);
- **residential_assets_value:** Value of the applicant's residential assets;
- **commercial_assets_value:** Value of the applicant's commercial assets;
- **luxury_assets_value:** Value of the applicant's luxury assets;
- **bank_asset_value:** Value of the applicant's banking assets;
- **loan_status:** Current status of the loan (approved, rejected).

2.1. Cleaning Data

Cleaning data is a key practice in data science that ensures data quality by identifying and correcting errors, inconsistencies and redundancies in datasets.

The dataset utilised in this project had already been pre-processed, so it was clean. The only detail we corrected was a space that appeared before the parameter names and to the data of some columns. This correction will prevent any future confusion.

For example: [' Approved' ' Rejected'] → ['Approved' 'Rejected'].

2.2. Analysing Data

A heatmap was used to analyse the correlation between predictors and loan status. This visual representation of data uses colours to represent values. In Figure 1 we can see that the variable with the greatest influence on loan status is the CIBIL Score.

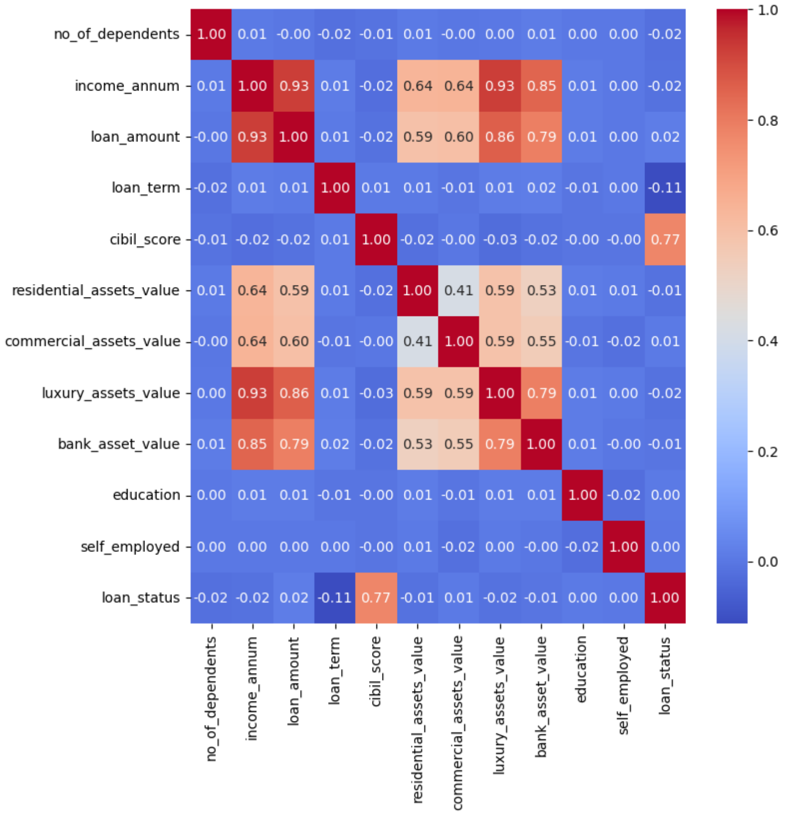


Figure 1 - Heatmap

2.3.Standardizing and Splitting Data

The data was separated into input and output, with the loan status being the output and all other features being the input, serving as predictors. Then the data was split into training and test sets in an 80/20 ratio. It is important to note that the training set will be divided into training and validation when appropriate.

As the output is binary, it is only required to standardise the input. By creating a MinMaxScaler, we can ensure that all input data is of the same scale.

Once the training set is divided into training and validation sets, these are also standardised before use. It is also relevant to note that not only in this case, but in the whole project, the randomness seed used was 42 (random_state = 42).

3. Creation and Training of the Models

3.1.Creation

As previously stated, this project will use the following models:

- Logistic Regression
- Support Vector Machines (SVM) Classification
- Random Forest Classification
- Gradient Boosting Classification
- Neural Networks

All these models were created under similar circumstances, except for SVM, where the ‘probability’ parameter had to be set to True for study purposes, and Neural Networks, where the ‘max_iter’ parameter was set to 1000 to facilitate and reduce study time.

3.2.K-Fold Cross-Validation (ROC curves & mean AUC)

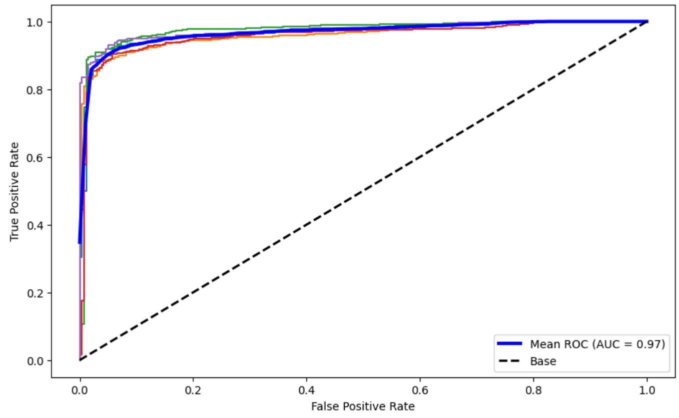
As in many projects of machine learning, the first training method used was k-fold cross-validation. The models were trained and evaluated using ROC curves and the mean AUC. To achieve a balance between robust evaluation of the model and computational efficiency, 5 folds were used in the k-fold cross-validation.

Before training, a choice had to be made between the classes StratifiedKFold and K-Fold for cross-validation, considering the distribution of 'loan_status' (Table 1). Due to the significant difference, the class StratifiedKFold was used.

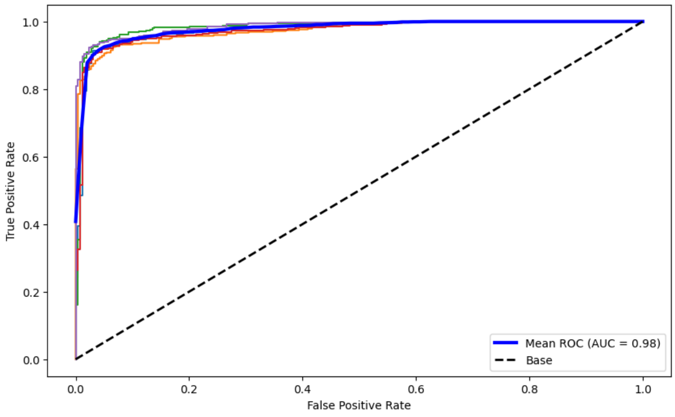
Table 1 - Loan Status Count

Loan Status	
Approved	2656
Rejected	1613

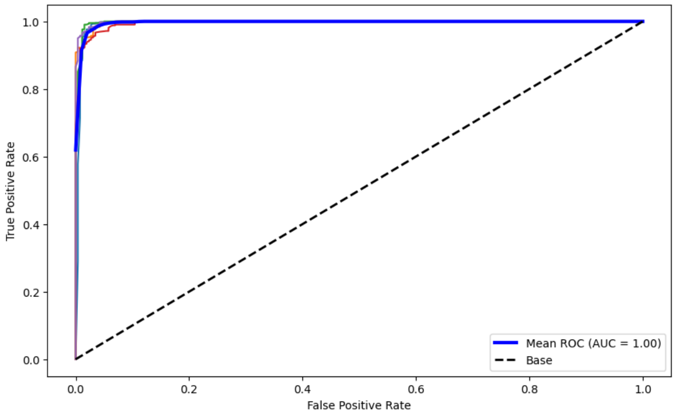
Subsequently, we implemented a cross-validation method accompanied by ROC curves and AUC of each fold and their mean values to assess the performance of the models.



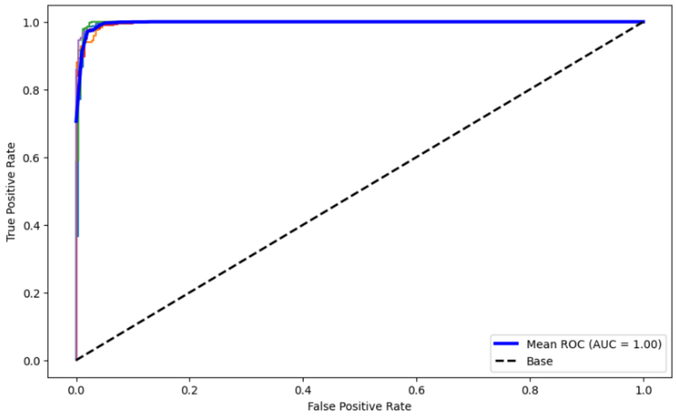
Graph 1 - ROC curve for Logistic Regression



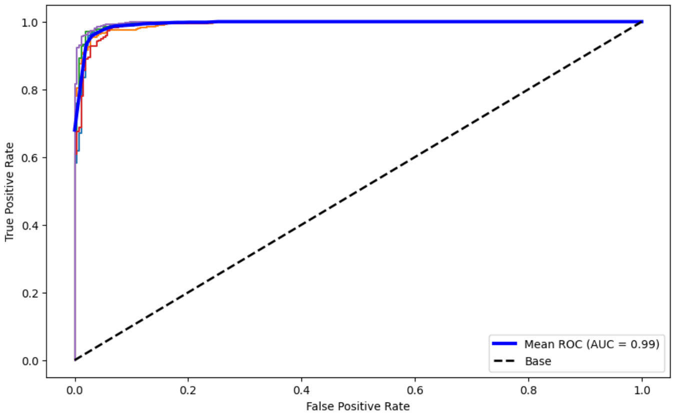
Graph 2 - ROC curve for SVM Classification



Graph 3 - ROC curve for Random Forest Classification



Graph 4 - ROC curve for Gradient Boosting Classification



Graph 1 - ROC curve for Neural Networks Classification

The above graphs indicate that the Random Forest and Gradient Boosting models have performed the best so far, with a mean AUC of 1.00. The Neural Networks, SVM, and Logistic Regression models follow closely behind in that order.

Based on these results, it can be stated that all the models are well suited to the data and are able to distinguish the classes effectively.

3.3.Best subset selection

Best subset selection for logistic regression is a fundamental approach to building predictive models, aimed at identifying the most effective combination of explanatory variables to optimise model performance. It explores all possible combinations of available features and evaluates performance metrics such as accuracy, precision, recall, F1-score, and AUC through cross-validation techniques, dividing the training set into train and validation sets.

The optimal features for the best subset were determined by iterating from 1 to 11 features, resulting in the following selection: 'income_annum', 'loan_amount', 'cibil_score', 'residential_assets_value', 'commercial_assets_value', 'bank_asset_value', 'self_employed' and these were selected by the model with the best AUC of 0.9178.

Lastly, we filtered the training data to these features, standardised it and trained the model with it.

3.4.Grid Search Cross-Validation and Forward Stepwise Selection

The remaining models were trained by performing grid search cross-validation followed by forward stepwise selection with the hyperparameters found. In this approach, the grid parameters were limited to the two most important ones in each model, so there weren't too many hyperparameters restricting the models. The second step of this training process was based on incrementing the features by relevance until they were all added and then selecting the combination with the highest AUC score.

Support Vector Machines (SVM) Classification

The Grid Search technique applied to find the best hyperparameters for the SVM model used for grid parameters is as follows:

- C (0.1 to 100) - regularization parameter;

- kernel type (linear, poly and rbf) - transformation applied to the input data

The best hyperparameters discovered were {'C': 10, 'kernel': 'rbf'}, resulting in a Mean AUC of 0.9870. As the 'C' value is small, the model can have a more flexible decision boundary. The Radial Basis Function (RBF) transformation is effective in capturing complex, non-linear relationships between features.

Based on the results obtained from grid selection, forward stepwise selection can then be applied. The features that yield the best performance to the case is only the 'cibil_score' with a mean AUC of 0.9590.

Random Forest

In this case, a maximum depth of up to 50 and a number of estimators between 50 and 500 were chosen where:

- n_estimators - number of decision trees that will be built in the forest;
- max_depth - maximum depth allowed for each decision tree in the Random Forest.

The best hyperparameters discovered were {'max_depth': 20, 'n_estimators': 450}, resulting in a mean AUC of 0.9979. 450 is a high number of decision trees which will improve the model's performance. This low max_depth, also known as 'shallow tree', is a less complex tree and may capture simpler relationships in the data. It helps prevent overfitting and promotes better generalization to unseen examples.

Using the Forward stepwise selection with the best parameters obtained in the grid search as described above, we have that the parameter that yield the best response to the problem is 'cibil_score', with a mean AUC of 0.9571.

Gradient Boosting

The performance of the model was evaluated using the mean AUC of the folds, considering a learning rate range of 0.01 to 0.5 and a number of estimators range of 50 to 500, where:

- n_estimators - number of weak learners (typically decision trees) that will be sequentially added to the ensemble;
- learning_rate - controls the contribution of each weak learner to the overall ensemble.

The best hyperparameters discovered were {'learning_rate': 0.2, 'n_estimators': 50}, resulting in a Mean AUC of 0.9979. A low number of estimators reduces the risk of overfitting.

A low learning rate means that the contribution of each tree to the final prediction is scaled down, making the training process more cautious and preventing rapid adjustments to the model. A low learning rate and low hyperparameters can be helpful in situations where computational resources are limited.

Using the same procedure as described above, the Best Features obtained was only the 'cibil_score', with a Mean AUC of 0.9595.

Neural Networks

For this case, the following grid parameters were used: learning rate init from 0.001 to 0.1 and hidden layer sizes of [(50,), (100,), (50, 50), (100, 50, 25)], where:

- learning_rate_init - size of the steps taken during the optimization process;
- hidden_layer_sizes - number of neurons or nodes in each hidden layer of the neural network.

The best hyperparameters discovered were {'hidden_layer_sizes': (50,), 'learning_rate_init': 0.1}, resulting in a Mean AUC of 0.9953. A high initial learning rate can lead to faster convergence during training, as the model adjusts. A small hidden layer size can act as a form of implicit regularization, helping prevent overfitting.

Using the same procedure as described above, the Best Features obtained were 'cibil_score', 'commercial_assets_value', 'loan_amount', with a Mean AUC of 0.9536.

3.5.Final Train

During the final training phase, the entire training set is used to ensure that the model is well adapted to all available patterns.

4. Test and Compare

4.1.Test

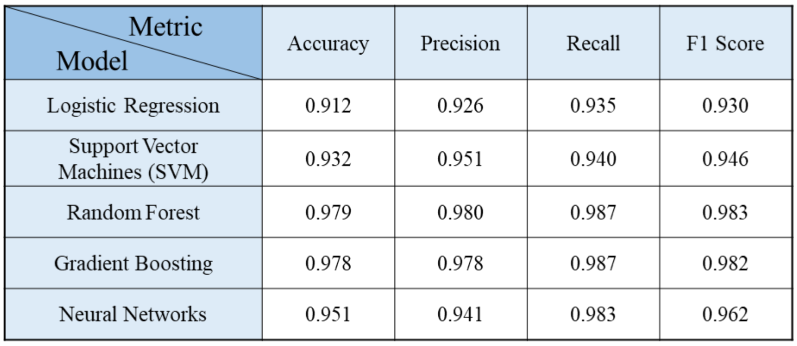
The testing phase evaluates the previously trained models against independent test sets. Predictions for each model are generated and stored in corresponding variables.

4.2.Performance Metrics

The performance metrics provide an objective measure of the model's effectiveness in a specific task, enabling informed choices during development.

- **Accuracy** refers to the proportion of correctly classified instances to the total instances.
- **Precision** indicates the proportion of true positive instances to the total instances classified as positive.
- **Recall** represents the proportion of true positive instances to the total instances positive.
- The **F1 Score** is a metric that combines precision and recall into a single measure. It is calculated as the harmonic mean between the two.

Table 2 – Performance metrics for each model



Metric Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.912	0.926	0.935	0.930
Support Vector Machines (SVM)	0.932	0.951	0.940	0.946
Random Forest	0.979	0.980	0.987	0.983
Gradient Boosting	0.978	0.978	0.987	0.982
Neural Networks	0.951	0.941	0.983	0.962

- The **Random Forest** model demonstrates exceptional performance, achieving an overall accuracy of 97.9%. Its high precision, recall, and F1 Score indicate a robust ability to make accurate predictions and deal with diverse situations.
- The results of the **Gradient Boosting** model are very similar to those of the Random Forest model, with an accuracy of 97.8%. This indicates that both models perform comparably, although Gradient Boosting is in every metric, except recall, slightly worse than Random Forest.
- **Support Vector Machines (SVM)** demonstrate a good balance between precision and recall, achieving values of 93.2% and 94.0% respectively. This model can be a solid choice when seeking a balanced combination of accurate predictions and the ability to recall positive instances.
- The **Neural Networks** model has an accuracy of 95.1% and a recall of 98.3%. These results suggest that the model is effective at identifying positive instances, although its precision is slightly lower compared to other models.
- The **Logistic Regression** model demonstrates consistency, achieving an accuracy of 91.2%, precision of 92.6%, recall of 93.5%, and F1 Score of 93.0%. It is particularly noteworthy for its interpretability, making it a solid choice when comprehensibility is crucial. However, the Random Forest and Gradient Boosting models slightly outperform Logistic Regression in all metrics, indicating an overall superior classification capacity.

Based on the results and analyses presented in the loan approval prediction study, the **Random Forest** model appears to be the **most appropriate** choice due to its robust ability to make accurate predictions and handle diverse situations. While Gradient Boosting also demonstrated comparable results, Random Forest slightly outperforms it in terms of accuracy. This decision is supported by the importance of obtaining accurate predictions when dealing with loan approval decisions, where confidence in the model is crucial.

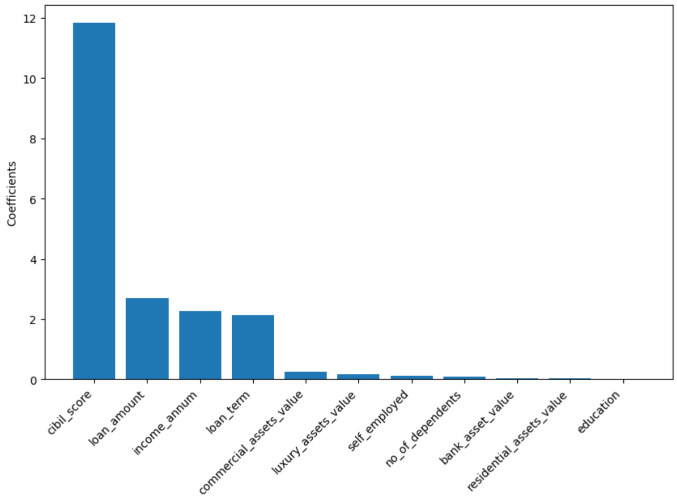
The selection of Random Forest as the best-performing model for a given dataset conveys valuable insights into the nature of the data and the problem at hand. This preference suggests that the underlying relationships within the data are likely non-linear and intricate, as Random Forests excel at capturing complex patterns.

Additionally, Random Forests demonstrate effectiveness in managing high-dimensional feature spaces, making them suitable for datasets with numerous variables. The ensemble nature of Random Forests provides a means of reducing overfitting and improving generalization, especially in situations where the dataset size is moderate.

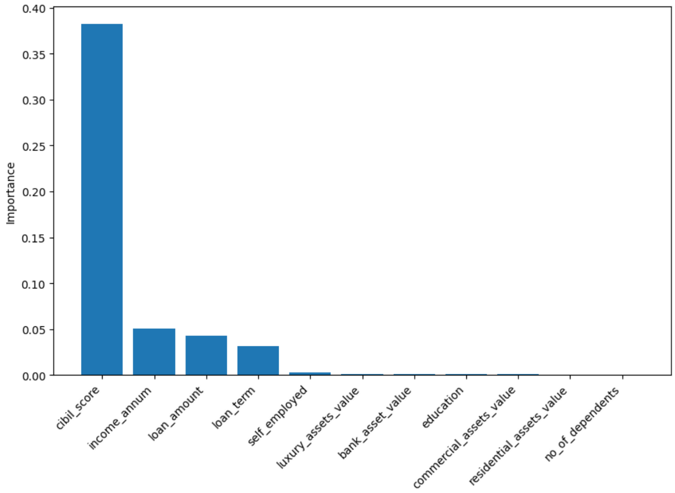
5. Feature importance

Feature importance is a metric used to evaluate the relative contribution of each feature. The graphs below show that the CIBIL score has a substantial impact on loan assignment, with an importance score of 0.8 for the best model (Random Forests Classification). This suggests that creditworthiness, as represented by CIBIL scores, is a dominant factor influencing the model's outcomes. It’s important to note that this dataset is from India, so it does not reflect the same conclusions to Portugal, however, in comparison, it’s curious to know that in Portugal this feature would not matter as much as it does here, about 60% less.

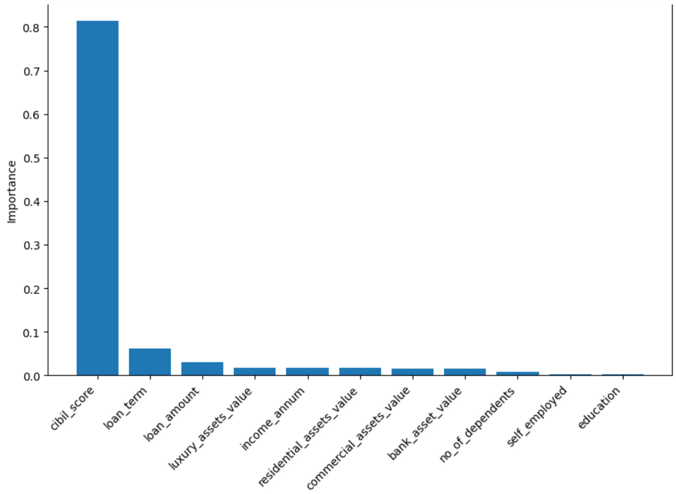
Conversely, the features 'loan_term', 'loan_amount', and 'income_amount' have lower importance scores, below 0.1 for the best model. In general, the top four most important features, including CIBIL score, are loan amount, income amount and loan term.



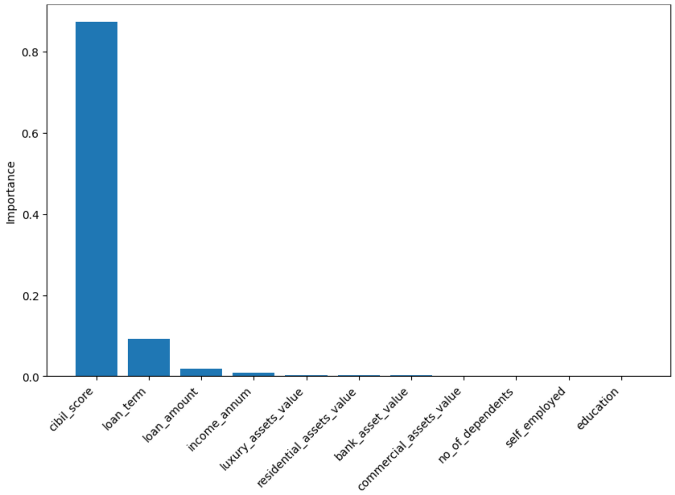
Graph 6 – Feature importance for Logistic Regression



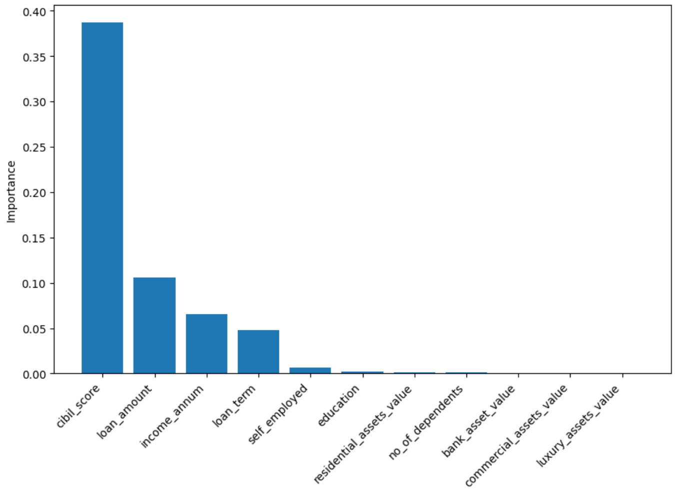
Graph 7 - Feature importance for SVM Classification



Graph 8 – Feature importance for Random Forests Classification



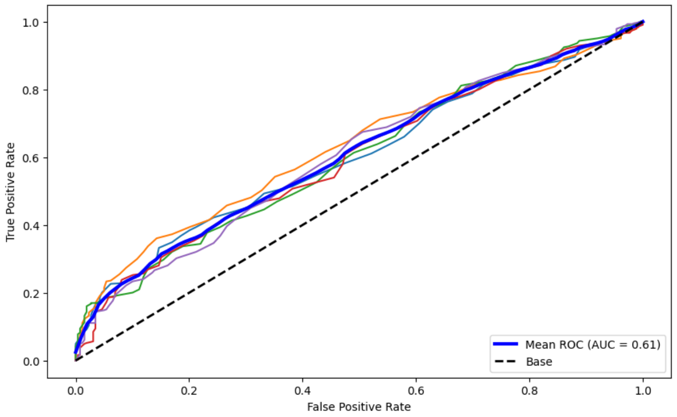
Graph 9 – Feature importance for Gradient Boosting Classification



Graph 10 – Feature importance for Neural Networks Classification

6. Extra Study

As the CIBIL score dominates the dataset, it is interesting to explore what the study would look like without this parameter. This study was repeated only for Random Forest Classification in order to avoid making it too exhaustive.



Graph 11 – ROC curve for Random Forest Classification without CIBIL score

As anticipated, after k-fold cross-validation, the mean AUC drops significantly (40%) without the main feature, in this instance to a value of 0.61. By retracing all the previous steps, we obtain the following values:

Table 3 – Extra Study Hyperparameters

Best Hyperparameters	With CIBIL Score	Without CIBIL Score
max_depth	20	30
n_estimators	450	500

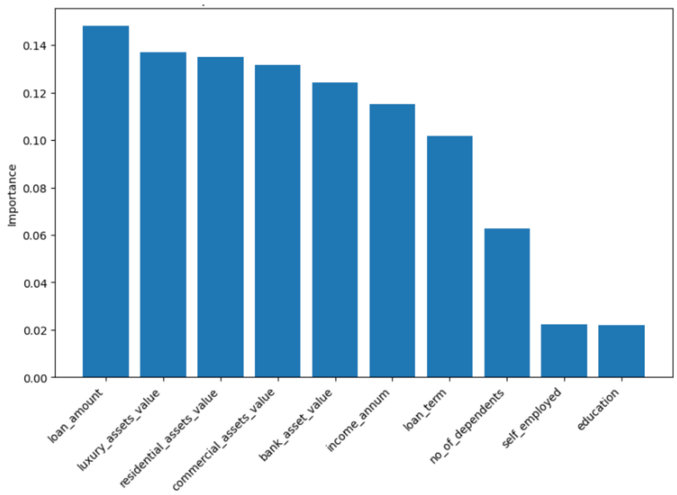
This means that the number of trees and the depth of each tree are both increased in this new model.

By applying forward stepwise selection again, we obtained a set of features that includes the following: 'loan_amount', 'loan_term' and ‘luxury_assets_value’ and these were selected by the model with the best AUC of 0.5551. Following the final training and testing, these different performance metrics were obtained:

Table 4 – Extra Study Performance Metrics

Random Forest	Accuracy	Precision	Recall	F1 Score
With CIBIL Score	0.979	0.980	0.987	0.983
Without CIBIL Score	0.528	0.634	0.789	0.703

In the graph below there is a clear difference in the distribution of the features in their importance. To conclude from this study, even though the main feature is gone, most of the other features have the same importance, however this change impacts heavily the performance of this extra model.



Graph 12 – New Study Feature importance

7. Dashboard

7.1. Loan approval dashboard

Page one assesses loan approvals and explores the demographic characteristics of individuals who accepted loans and self-employed participants. Users interact with sliders to dynamically manipulate these variables to observe their impact on loan approval metrics, keeping in mind that more specific values, more bias the data will be.

7.2. Key visualisation features

The second page presents a graph for each of the four most influential features. Users can dynamically modify these graphs by selecting from the four most dominant variables: loan term (months), loan amount, income amount and CIBIL score. The CIBIL score emerges as the most dominant variable, significantly influencing the graphs shown. In addition, a bar chart illustrates the number of borrowers per CIBIL score rating (Excellent, Good, Average and Poor). This CIBIL score aligns with the Forbes Advisor assessment presented in Table 5.

Table 5 - CIBIL Score Ranking by Forbes

Rating	CIBIL Range
Poor	300-499
Average	500-649
Good	650-749
Excellent	750-900

7.3. Model selection and evaluation

On the final page, users select from predictive models, including Gradient Boosting, Logistic Regression, Neural Network Classification, Random Forest Classification and Support Vector Machines (SVM). This page provides comprehensive insights, showing the Receiver Operating Characteristic (ROC) curve, feature importance, accuracy, precision, recall and F1 score for the selected model.

Contributions: Ricardo: main focus on dashboard development with support in the other areas; João: Main focus on writing the report and analysing the results with support in the other areas; Francisco: main focus on coding and analysing the results with support in the other areas.