# Credit Classification Report

UCD - Carl Somers (19422704)

01/05/23

# I. Executive Summary

The German Credit dataset is a widely used dataset in the machine learning community, consisting of 798 entries with 20 categorical and continuous attributes. The objective of the dataset is to predict whether a person applying for a bank loan is a good or bad credit risk. The dataset is imbalanced, with more than double the number of good credit observations. Exploratory data analysis shows that variables such as duration, amount, age, and savings status are critical predictor variables. Therefore, we conclude that both customer characteristics and loan specifications are important for this task.

To prepare the data for modelling, we use one-hot and binary encoding for feature engineering. To simplify the model, we apply two feature selection techniques: Chi-Squared testing and Recursive Feature Elimination. Accounting for the data imbalance we test both SMOTE and Random Oversampling techniques. We then evaluate a broad range of classification models and find that the final Random Forest model has an accuracy of 79% and a more crucial metric, macro F1-score, of 66% on out-of-sample data.

# II. Data

This is a dataset was supplied by DataSoc for the Data Link 2023 competition. From initial inspection this is a modified version of the classic German Credit dataset, which is common across machine learning literature.

The dataset contains 798 entries with 20 categorical and numerical attributes. These entries outline persons who have applied for a bank loan, or credit. These people are then classified by the bank as either good or bad credit based on their applications attributes, this is our target or dependent variable. The dataset is described in the below table:

| ***Variable Name*** | ***Description*** | ***Data Type*** |
|---|---|---|
| **checking_status** | Status of existing checking account | Categorical (ordinal) |
| **duration** | Duration in months | Continuous |
| **credit_history** | Credit history | Categorical (ordinal) |
| **purpose** | Purpose of credit | Categorical (nominal) |
| **credit_amount** | Credit amount in euros | Continuous |
| **savings_status** | Savings account/bonds status | Categorical (ordinal) |
| **employment** | Present employment since | Categorical (ordinal) |

| | | |
|---|---|---|
| **installment_commitment** | Instalment rate in percentage of disposable income | Discrete |
| **personal_status** | Personal status and sex | Categorical (nominal) |
| **other_parties** | Other debtors / guarantors | Categorical (nominal) |
| **residence_since** | Present residence since | Continuous |
| **property_magnitude** | Property | Categorical (ordinal) |
| **age** | Age in years | Continuous |
| **other_payment_plans** | Other instalment plans | Categorical (nominal) |
| **housing** | Housing | Categorical (nominal) |
| **existing_credits** | Number of existing credits at this bank | Discrete |
| **job** | Job | Categorical (nominal) |
| **num_dependents** | Number of dependents | Discrete |
| **own_telephone** | Has a telephone | Categorical (binary, nominal) |
| **foreign_worker** | Is a foreign worker | Categorical (binary, nominal) |
| **class** | Creditability (target variable: 1 = good, 0 = bad) | Categorical (binary, nominal) |

Furthermore, we have no missing variables in our dataset. We employ a 90/10 train/test split to ensure that we do not introduce bias into our modelling. All EDA and variable selection is performed on the training/development set (n = 638).

## III. Exploratory Data Analysis

First, we can split the *personal_status* variable into both *sex* and *relationship_status* variables. These may prove to be more useful variables for exploratory analysis.

**Target Variable**

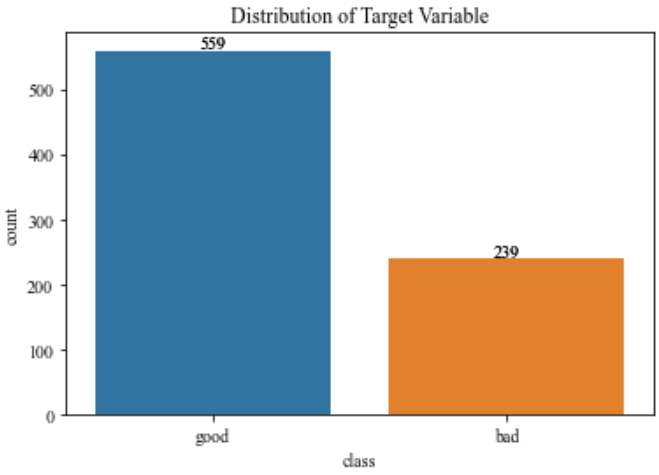


Figure 1 – Target classification count plot

Here we can immediately observe that this is an imbalanced dataset. Here we have more than double the number of good credit observations. This is not ideal for a classification variable as it may result in a biased learning model.

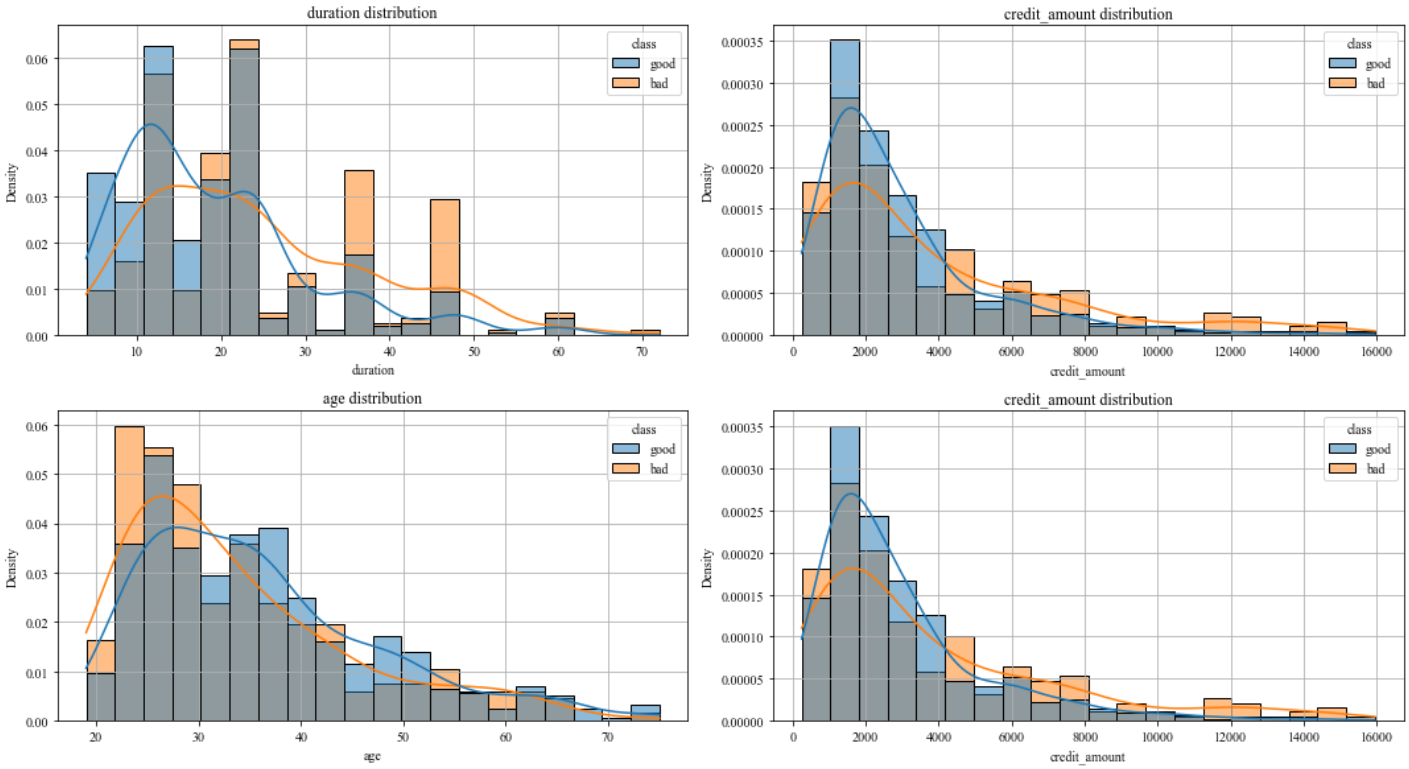


Figure 2 - Distribution plots of continuous variables

We can immediately note the differences between the distributions of the classes across our continuous variables. This highlights that these variables may be useful elements to add to our model.

We can see that duration has a disproportionate distribution across good and bad credit applications. This too is clear in age. Bad credit is likely to come from those who take on longer duration loans, but is this impacted by loan amount? Is there a difference in age group with paying back loans, regardless of amount?

We will observe the two hypotheses in more detail: 1) Duration and loan amount are important predictors for credit class; 2) Age results in a lower likelihood of default (bad credit).


Figure 3 – Scatterplot of Loan Duration on Loan Amount Divided by Class

Here we can confirm our previous findings that bad credit customers both tend to take out longer loan durations and also much larger loan amounts than their good credit equivalents. However, we must also observe if age is a contributing factor given that larger purchases such as a house come at mid-years (20-50).

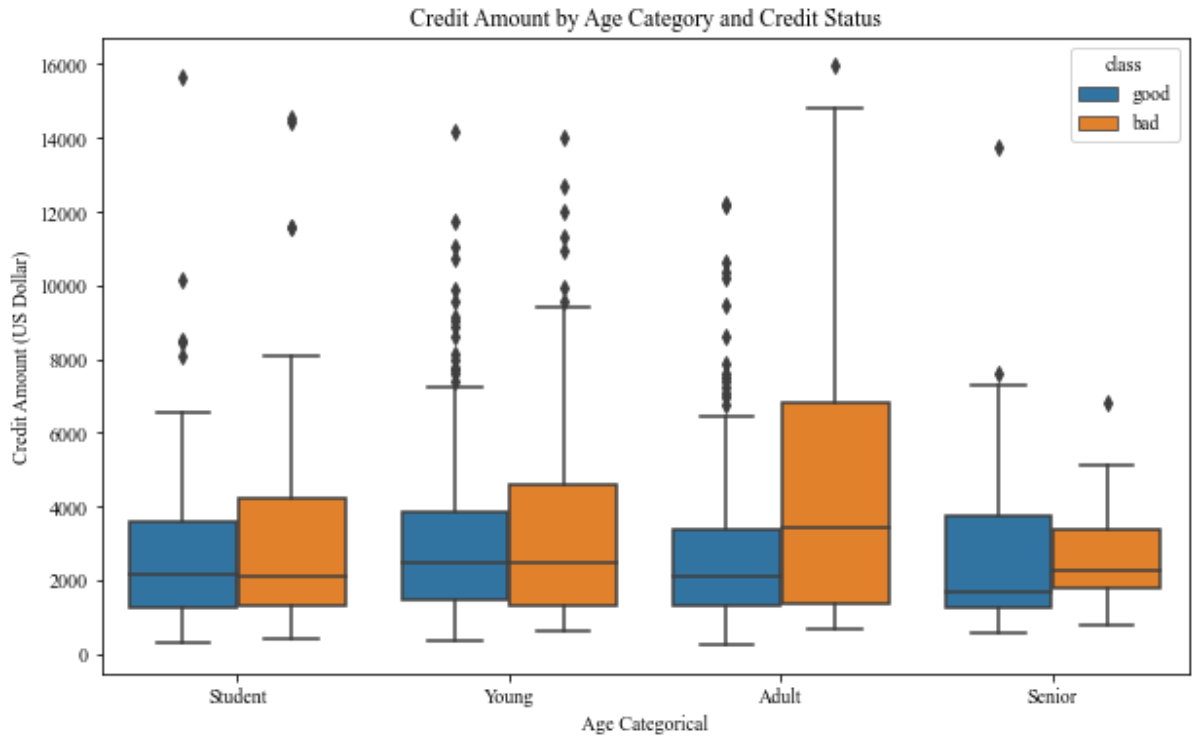


Figure 4 – Breaking Credit Class by Age and Credit Amount

From this we can see that Adults are in fact the most likely to default. This is mostly due to the much larger loans taken out by this age category. These loans include mortgages. Those who take out smaller loans are once again shown to have a better track record of good credit.

Now let's look at our categorical features. Immediately one stands out – savings account. If someone has no savings, it is very unlikely they will be able to cover the loan, and this may result in a default therefore bad credit.

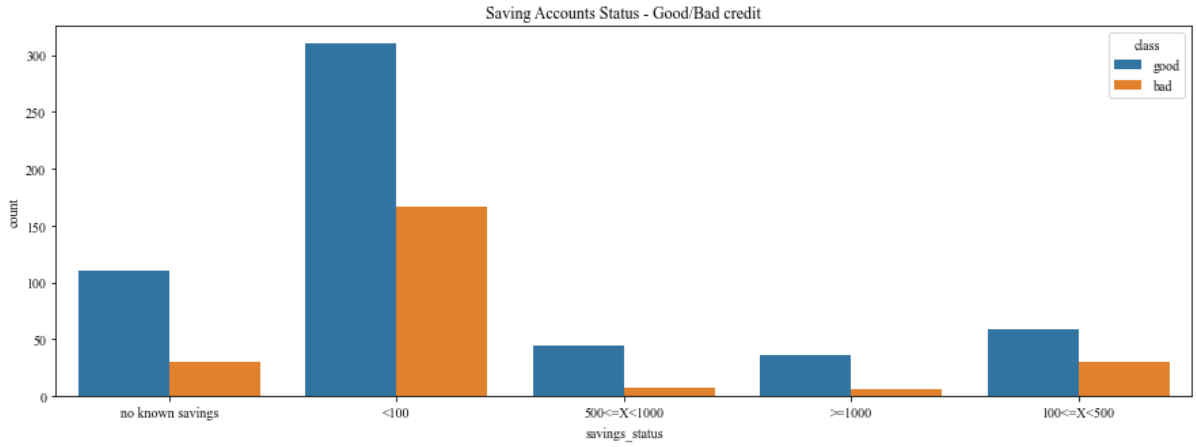


Figure 5 – Savings Status Counts by Class

Here it is difficult to see whether or not savings has a direct impact on credit classification, it seems that both are spread proportionately through our sample. However, a larger number of people with <100 in savings and those in the next category between 100 and 500 are more likely to be bad credit in comparison to their counterparts. However, could this be due to credit amount?

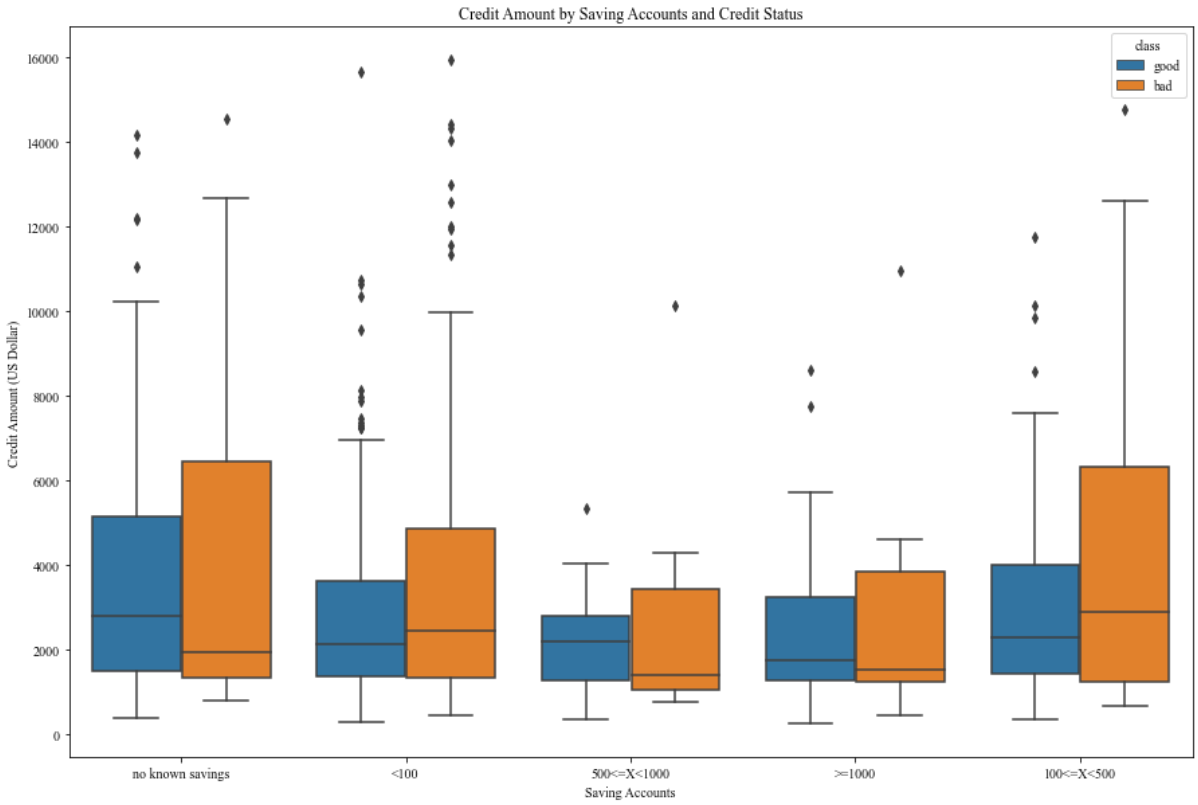


Figure 6 – Credit Amount by Savings and classification

This presents us with interesting information, those who are bad credit and have <100 or between 100 and 500 take out larger loans then their peers. For the lower end of savers the loans are extremely large, this disparity is not seen within those who have >500 in their savings accounts.

# IV. Feature Engineering

**Basic Encoding**

First, we must find a way to employ our categorical variables in our model. One way to do this is through categorical (label) encoding. This can be extremely useful for feature engineering and for ordinal variables, such as *checking_status* or *credit_history*. Note, this is suboptimal for unordered variables however it does prove useful in a clean correlation plot.

For non-ordinal variables (ones without a clear ordering) we employ one-hot encoding. This allows us to turn each variable categorical into several binary features, this prevents our models from inferring an order. For example, our housing variable can be split into “own”, “for free” or “rent”.

Below is our correlation matrix of the top 10 feature correlations to our target class variable.

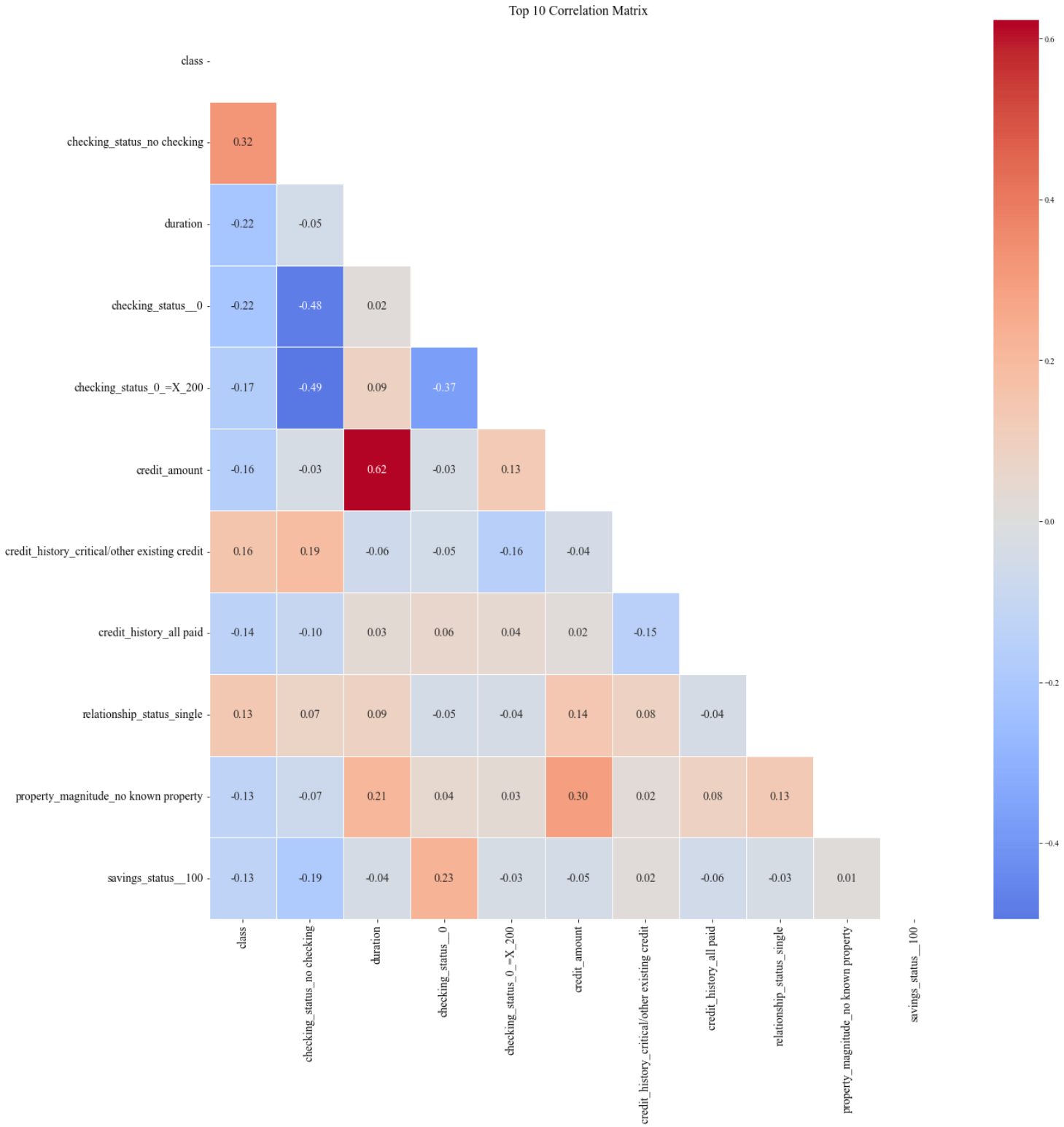

Figure 7 – Correlation plot top 10 correlated with target.

From this correlation plot, along with our previous EDA we can confirm the importance of duration, and no checking account as predictor variables for our classification task. This further reinforces our exploratory analysis. While this is a useful exercise to examine some of the top 10 key-elements, it would not be robust to select features based on an arbitrary correlation coefficient cut-off point, we therefore will employ more vigorous (automated) selection criteria.

## V. Feature Selection

We will employ 2 forms of feature selection: statistical (Chi-squared) testing and recursive (random forest feature elimination). We hope by doing this we can add robustness to our selection, reducing our model complexity without dropping features that have potential to reap predictive power.

**Chi-Squared Feature Selection**

Chi-Squared feature selection is a statistical technique used for selecting the most informative features in a dataset with categorical or discrete variables. It measures the dependency between each feature and the target variable using the Chi-Squared test statistic. The formula for the Chi-Squared test statistic for a feature f and target variable y is:

$$\mathcal{X}^2(\mathrm{f},\mathrm{y}) = \sum \frac{(\mathrm{O_{i,j}} - \mathrm{E_{i,j}})^2}{\mathrm{E_{i,j}}}$$

where Oij is the observed frequency of the joint distribution of feature f and target y, and E_ij is the expected frequency under the assumption of independence. The Chi-Squared test statistic measures the deviation of the observed frequencies from the expected frequencies and follows a Chi-Squared distribution. We select the features with a p-value less than 0.05, indicating that they are statistically significant at a 5% significance level. This whittles our 59 features into just 25, significantly reducing the model complexity.

However, to ensure the accuracy of this feature selection method we confirm using Recursive Feature Elimination (RFE).

**Random Forest Recursive Feature Elimination**

Random Forest Recursive Feature Elimination (RF-RFE) is a feature selection technique that uses the Random Forest algorithm to eliminate less important features from a dataset. It

iteratively removes the feature with the lowest importance score, calculated using metrics such as Mean Decrease Impurity (MDI), Mean Decrease Accuracy (MDA), or Permutation Importance (PI), until a desired number of features is obtained, or model performance starts to degrade.

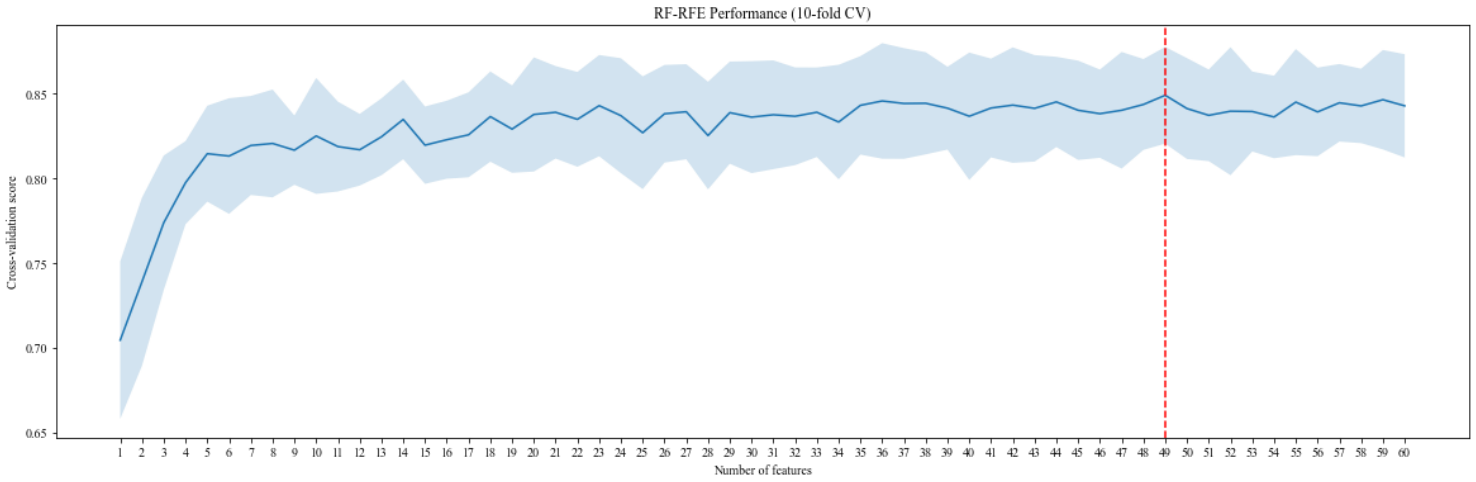


Figure 8 – RF-RFE Feature Selection on Encoded Variables

This method provides us with 2 valuable insights: 1) it suggests that the top 6 features contribute disproportionately to the performance; 2) out of 59 encoded features we will only drop 10 leaving us with 49 features.

We finally combine all the selected features from the resulting in a final variable selection of K = 49. Therefore, we have dropped 8 variables deemed unimportant.

# VI. Modelling

For our modelling we will also split our data into a development and validation set at a 90/10 split. This is to aid us in model selection, which we will further fine-tune.

## 6.1 Data imbalance

Generally, data imbalance often occurs in the credit risk classification due to the huge differences of the number of good borrowers and bad borrowers. SMOTE (Synthetic Minority Oversampling Technique) is one of the most widely used approach to address this problem. SMOTE utilises a k-nearest neighbour algorithm to create synthetic data. As our data is mixed with categorical and continuous data, we use ENC-SMOTE. However, ROS (Random Oversampling) is also a common, yet more naïve approach. This works by randomly selecting examples from the minority class, with replacement, and adding them to

the training dataset. Dealing with data imbalances is generally neglected in credit classification, however, we will test the use of both in our analysis.

### 6.2 Evaluation Metrics

To evaluate our models, in both training and testing we will employ the following evaluation metrics.

1. Accuracy: The proportion of correctly classified instances to the total number of instances. It is computed as:

$$\textbf{accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

where *TP* is the number of true positives, *TN* is the number of true negatives, *FP* is the number of false positives, and *FN* is the number of false negatives.

2. Precision: The proportion of correctly classified positive instances to the total number of instances predicted as positive. It is computed as:

$$\textbf{precision} = TP / (TP + FP)$$

3. Recall: The proportion of correctly classified positive instances to the total number of positive instances in the dataset. It is computed as:

$$\textbf{recall} = TP / (TP + FN)$$

4. F1-score: A harmonic mean of precision and recall, with equal weight assigned to both metrics. It is computed as:

$$\textbf{F1} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

It is clear in our classification task that false positives (incorrectly stating good credit when bad) will be a costly mistake for the loan providers, this is mostly evaluated by precision. However, false negatives will also result in both unhappy customers and opportunity costs. To balance the trade-off between these we will observe/minimise the F1-score for our final model tuning.

### 6.3 Model Training

In order to select the best possible model we observe a wide range of classification models.

**Linear Models:**

- Logistic Regression: A simple linear model that uses a logistic function to model the probability of a binary outcome.

**Tree-Based Models:**

- Random Forest: A decision tree-based ensemble model that randomly selects subsets of features and samples to create a set of decision trees, and then aggregates their predictions to make the final prediction.
- AdaBoost: Another decision tree-based ensemble model that assigns weights to misclassified samples to improve performance.
- XGBoost: A gradient boosting decision tree-based model that uses a gradient descent algorithm to minimize the loss function.
- CatBoost: A gradient boosting decision tree-based model that can handle categorical features directly.
- LightGBM: Another gradient boosting decision tree-based model that is designed to be faster and more memory-efficient than other gradient boosting frameworks.

**Neural Network-Based Models:**

- MLPClassifier: A multi-layer perceptron neural network model that is often used for simple classification tasks.

You may notice the wide range of tree-based models – this is because they benefit from accuracy in classification tasks and explain-ability.

We first must observe our training results based on the evaluation criteria in 5.2. This will allow us to determine which models may prove useful for hyper-parameter tuning. We will fit and evaluate these using the basic unbalanced and the balanced datasets. Evaluation is done using 10-fold cross evaluation.

| | **accuracy** | **precision** | **recall** | **f1_score** | **ROC AUC** |
|---|---|---|---|---|---|
| **(ros) Random Forest** | 0.88 | 0.88 | 0.88 | 0.88 | 0.95 |
| **(ros) LightGBM** | 0.87 | 0.87 | 0.8672 | 0.87 | 0.94 |
| **(ros) XGBoost** | 0.86 | 0.87 | 0.8638 | 0.86 | 0.94 |

| | | | | | |
|---|---|---|---|---|---|
| **(ros) CatBoost** | 0.86 | 0.87 | 0.8634 | 0.86 | 0.94 |
| **(ros) Neural Network** | 0.85 | 0.85 | 0.8460 | 0.85 | 0.90 |

Table 2 – Top 5 Models in CV Model Evaluation on Training Set

This training evaluation has two implications: 1) tree-based models are top performing in our classification task (on training set); 2) Random Oversampling has far outperformed both SMOTE and unbalanced datasets. Models trained on unbalanced data had poor performance in classifying bad credit applications.

We therefore will select Random Forest, XGBoost and CatBoost for further hyperparameter tunings and test evaluation. We select CatBoost as it has potential from prior research, and in application on imperfectly balanced datasets (such as ours despite resampling efforts).

## VII. Evaluation

We employ Optuna to parameter tune our selected models. This employs a grid-search across a range of parameters in order to optimise our accuracy evaluation – training on our development set, scoring on our validation set (90/10 split).

Once we have tuned our model parameters for our Random Forest, XGBoost and Catboost models we can score each model individually on our held-out test dataset.

| **Model** | **Precision** | **Recall** | **F1-score** | **Accuracy** |
|---|---|---|---|---|
| **Random Forest** | 0.75 | 0.64 | 0.66 | 0.79 |
| **XGBoost** | 0.64 | 0.59 | 0.60 | 0.74 |
| **CatBoost** | 0.67 | 0.59 | 0.59 | 0.75 |

Table 3 – Out-of-Sample Testing with (unweighted) Macro Averages

Individually we note a disparity in comparison to our training evaluation, heavily dropping in out-of-sample testing. This means that our models were overfit, this is likely due to the resampling measures taken. The models were too specified around the (limited) bad credit participants from our training set which were not generalised to the full test set.

It is important we measure across a wide range of metrics, so we understand the difficulties beneath our models. For example, we observe the macro averages to count both positive and negative classes. Our best performing model based on this evaluation is the Random Forest.

## VIII. Conclusion

One of the biggest challenges in credit classification is the significant imbalance between good and bad credit customers. However, this is often a trial left unaccounted. To address this problem, we utilized various re-sampling techniques along with a wide range of models and metrics. Our objective was to accurately classify the applicants by leveraging historical data.

From our exploratory analysis we can observe the contrast in distribution between the two credit types. We found that both customer characteristics like age and savings along with loan specifications like duration and amount are key considerations to a customer's creditworthiness. We used this knowledge to apply more robust variable selection techniques (RFE and Chi-squared testing) to ensure that we can properly account for all observed relationships within the dataset.

After evaluating several models and metrics, we found that using randomly oversampled training data and fine-tuning a Random Forest model produced the most accurate credit classifications. We selected macro-F1 score as our primary evaluation metric to ensure that both positive and negative classifications are given equal importance.

For future research, it would be valuable to explore the use of neural networks on small, imbalanced datasets for classification tasks. Additionally, novel feature engineering approaches could potentially further improve model performance.

# IX. Appendix

Table 1 - Excluded Features

| **Feature** |
| --- |
| foreign_worker |
| purpose_domestic appliance |
| purpose_other |
| purpose_repairs |
| purpose_retraining |
| other_parties_co applicant |
| other_parties_guarantor |
| other_payment_plans_stores |
| job_unemp/unskilled non res |
| relationship_status_mar/wid |