Text

Description automatically generated

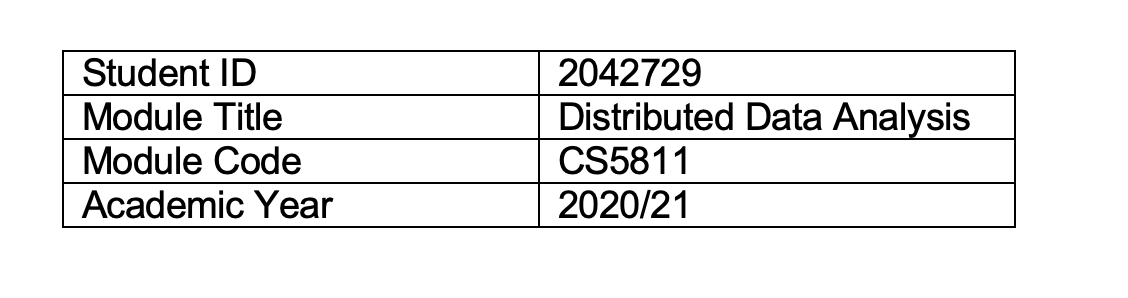


Table of Contents

[1. Data description and Research question 3](#_Toc70073886)

[1.1. Data Description 3](#_Toc70073887)

[1.2. Research question 3](#_Toc70073888)

[2. Data preparation and cleaning 3](#_Toc70073889)

[2.1. Merging Datasets 3](#_Toc70073890)

[2.2. Feature Selection 4](#_Toc70073891)

[2.3. Handling Missing values 5](#_Toc70073892)

[2.4. Type conversion 6](#_Toc70073893)

[2.5. Handling Outliers 6](#_Toc70073894)

[2.6. Data Preparation for PCA using Pyspark 8](#_Toc70073895)

[2.7. Sampling for Supervised Machine Learning 8](#_Toc70073897)

[3. Exploratory data analysis 8](#_Toc70073898)

[3.1. Preliminary data exploration 9](#_Toc70073899)

[3.2. Principle Component Analysis 10](#_Toc70073900)

[4. Machine learning Implementation 11](#_Toc70073901)

[4.1. Random Forest Baseline Model 12](#_Toc70073902)

[4.2. Random Forest Parameter Tuned Model 12](#_Toc70073903)

[5. Performance evaluation and comparison of methods 13](#_Toc70073904)

[5.1. Confusion Matrix 13](#_Toc70073905)

[5.2. ROC – Curve 14](#_Toc70073906)

[6. Discussion of the findings 14](#_Toc70073907)

[7. Data Management Plan and Author Contribution statement 14](#_Toc70073908)

[8. References 15](#_Toc70073909)

[9. Appendix 15](#_Toc70073910)

# 1. Data description and Research question

## 1.1. Data Description

The data set used for the project is obtained from Home Credit company which is a non-banking financial institution based on Netherlands (Kaggle.com, 2018). The master dataset produced for the machine learning process consists of meta data from three different data sets. Firstly, the application\_data.csv consists static data for all applications with each row representing one loan and further information about client demographic, geographic and transactional information at the time of application. Secondly bureau\_data.csv consists of information about client's previous credits at other financial institutions with each row corresponding to client's past loan behaviour. Lastly the previous\_application.csv data which has client's previous application information within Home Credit company. The detailed description of each feature is presented in Table 1,2 and 3 in appendix.

|  |  |  |
| --- | --- | --- |
| Data set | Instances | Features |
| application\_data.csv | 307511 | 122 |
| bureau\_data.csv | 1716428 | 5 |
| previous\_applications.csv | 1670214 | 4 |

## 1.2. Research question

The aim of the project is to build a machine learning model that can classify bad applications from good loan applications i.e., whether the client has defaulted on regular loan repayments. Doing so can help Home credit company to estimate the likelihood of risk associated with new client’s loan application given the applicant’s past loan history, demographic, geographic and transactional information therefore helping the company in accepting or rejecting the loan application.

# 2. Data preparation and cleaning

Data preparation and cleaning is primarily divided into two sections where the first part involves merging the datasets, type conversion, handling missing values, outliers and preliminary feature selection. The second part involves preparing data for unsupervised machine learning (PCA) by encoding features to numerical values which is done using HPCI techniques, followed by sampling the data for model building.

## 2.1. Merging Datasets

The application\_data.csv is the main data set where each instance represent information about current loan application with unique identifier being loan ID. However, the other datasets from bureau and previous application have information about past credit records of the applicants with different unique identifiers, which means each application in our main dataset can have multiple records in previous and bureau datasets. Hence this requires grouping the “bureau” and “previous application” datasets by Loan ID and counting the number of past applications clients had for joining datasets. Furthermore, the categorical features present in each dataset such as “credit\_currency” and “credit\_type” are one hot encoded and counted for merging. The inner joining of the datasets results in 18% loss in instances which might be because the client doesn’t have any past credit history or data missing in secondary datasets. Since the research question is reliant on understanding client risk given past behaviour, these instances will be ignored for further analysis.

## 2.2. Feature Selection

The resultant dataset from the inner join produces high dimensional data with 161 features and 249507 instances. Having these many features can lead to curse of dimensionality by increasing computation costs, overfitting of models and thereby decreasing the performance of the model (Xu, 2018). To handle this issue five different approaches were used which are as follows:

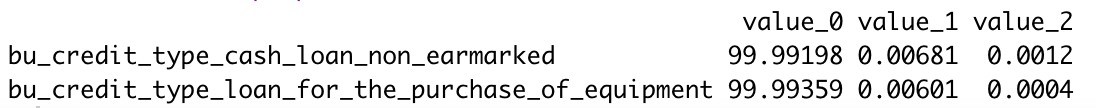
i) The dataset consists of 48 features with missing data above 40%, all of them consists information about the property of the client such as living area, number of floors and elevators etc., However these features do not exhibit clear pattern with either target feature or the feature that describes whether client owns a house or not. Considering imputing these features pose risk to performance of machine learning, these features are dropped.

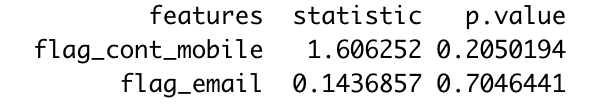
ii) There are two features which has only one distinct value, so these features are dropped.

iii) The third approach is to build a contingency table of features with two distinct values with respect to the target feature and drop the features with less than 0.05% target feature proportion. A snapshot for two example features is presented below and complete table can be found in appendix (Table 4).



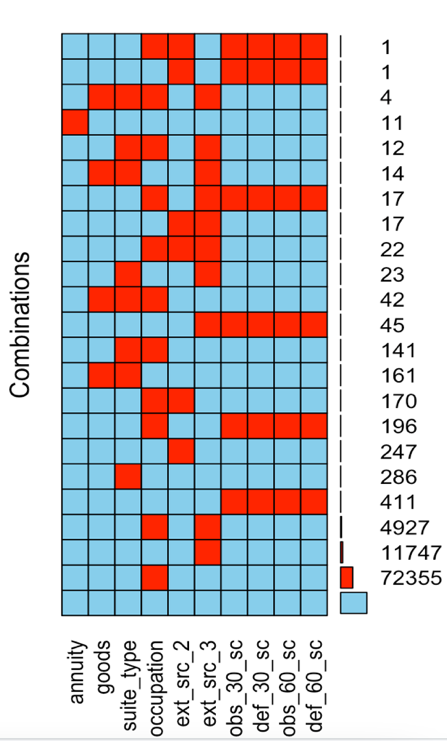
iv) The fourth approach involves identifying features with three distinct values and estimating proportion of numerical levels within each feature and dropping features with 99.99% domination of only one category level. The two features which are dropped are presented below:



v) The final approach involves conducting a Chi-Square test of independence for all the categorical features present in the dataset with respect to the target variable i.e., whether the client has defaulted on payments or not. The null hypothesis(H0) is that there is no relationship between the features. Therefore, any feature that test at significance level of less than 0.05, we can reject null hypothesis and suggest that they are dependent on target variable and can be used for the model building. We’ve identified 6 features that pose no relation with target variable hence they are dropped. A sample test for two rejected features is presented below and complete table can be found in appendix (Table 5).

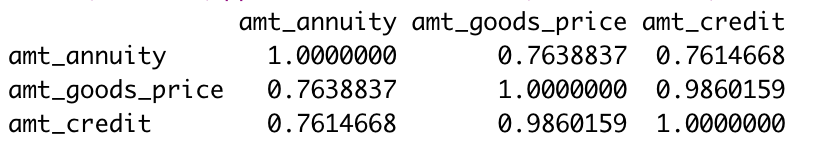
## 2.3. Handling Missing values

The feature selection from the previous section has reduced the dimensions of the dataset from 161 features to 89 features which are considered for the next stages of the project. The following section involves identifying missing data patterns through “VIM” package and take necessary steps to either discard or impute the values based on case-by-case analysis.

i) Figure.1 below represents the missing data patterns in the dataset for the 9 features identified. We can clearly see that the features "obs\_30\_sc", "def\_30\_sc","obs\_60\_sc" and "def\_60\_sc" showing clear pattern of missingness. There are 411 instances where there are missing values for all these four features. Since the data is Missing at Random (MAR) and imputing can increase risk of false machine learning outcomes the chosen approach is to discard the instances which results in 0.0.2% loss of instance in the dataset.

ii) The dataset contains two categorical features namely “occupation\_type” and “suite\_type” (who accompanied client while applying loan) have missing values for 683 and 77887 instances respectively. The missing values in occupation type have high default rate of 25% from overall default rate and removing the instances can impact model performance therefore the missing values are replaced with “Occupation\_unknown”. Similarly, the missing values in “suite\_type” are replaced with “Suite\_unknown”.

Figure 1:Missing data patterns

iii) The features “amt\_annuity” and “amt\_goods” convey transactional information by the clients the first being monthly amount payable by the client and the second being price of the goods for which loan is taken. These features are highly correlated with each other and the feature “amt\_credit” as shown below. Hence the preferred approach is to build a linear regression model on the non-missing data and impute the predicted values to the missing values. The regression models are built such that to predict one feature the other two are used as explanatory features. Both the models are significant with R-squared value of 0.60 and 0.97 respectively. The complete model diagnostics for both models in the are presented in the appendix (Table 6 & 7).

iv) The features “ext\_src\_2” and “ext\_src\_3” are normalised scores from external data sources and have missing values of 0.001% and 6.73% of total instances respectively. Since the missing values in “ext\_src\_2” are very minute they are replaced with mean value of the feature. For the feature “ext\_src\_3” the chosen approach is to build a KNN model on sub-sample of 30,000 randomly selected instances and testing Root mean square error on 30,000 different random instances. The RMSE value for the model is 0.82 which is significant and therefore the model is applied on the missing data and imputed with predicted values.

v) The features “organization\_type” and “code\_gender” the missing values are represented with a factor level namely “XNA”. The 44687 instances where “organization\_type” is missing are replaced with “organisatization\_unknown” and 4 instances where gender is missing are removed from dataset.

vi) Finally, the feature “own\_car\_age” has missing instances of 65%, however the missingness appears to be at not at random (MNAR) and dependent on feature “flag\_own\_car”. All the missing values in “own\_car\_age” represent clients who do not own a car. Hence the values are replaced with “0”.

## 2.4. Type conversion

The feature “organisation\_type” has 58 factor levels within them and most of the levels belong to subgroups within each level. For example, there are factor levels of Industry types 1 to 13 and similarly trade types 1 to 7. Since R’s “randomForest” package cannot handle factors with more than 32 levels, these levels are grouped together. For instance, all the Industry types from 1 to 13 are grouped together as just “industry”.

## 2.5. Handling Outliers

The chosen plan for outlier treatment is to use “Validate” package in R and address the data quality issues by constructing logical rules based on initial data exploration and meta data provided. The instances violated the logical rules are examined case-by-case and necessary approach is taken to handle them accordingly. The code for each logical rule is provided in attached R file.

i) The features “days\_birth”, “days\_employed”, “days\_registration”, “days\_id\_publish”, “days\_last\_phone\_change” gives information about days before the application date in negative values. For example, if “days\_registration” is -1000 then it says client changed registration 1000 days before application. According to the validation rule these days cannot be greater than 0 but “days\_employed” feature has 44583 instances where the value is “365243”. After inspection it is found that these instances belong to occupation type “unknown” which implies these people are unemployed hence the values are replaced with “0”.

ii) The second logical rule identifies the instances where client own’s a car, but the car age is represented as “0”. However, these instances do not belong to the subset where we’ve replaced “0” for missing values of who don’t a car in previous section. As the client’s car age is represented in years starting from 1 suggests that these instances belong to client who owns a new car but haven’t passed a year. So, the values are replaced with 0.5 value. Another data quality issue for the “own\_car\_age” is there are three instances where car age is greater than 65 years and these values are replaced with median value of the feature.

iii) The features “cnt\_fam\_members” and “cnt\_children” convey information about the number of family members and children the client has respectively. The logical rule extracts instances where client have more than 15 children and 20 family members. The two instances where the rule is violated appears to be implausible because the client age for these instances are less than 30 years so these are removed.

iv) The features “obs\_30\_cnt\_social\_circle” and “def\_30\_cnt\_social\_circle” gives information about the client background check done by the company to identify number of individuals in client social surroundings are likely to default or defaulted that have 30 days past due period. There is one instance where these values are far extreme to the rest of the instances, so the instance is removed.

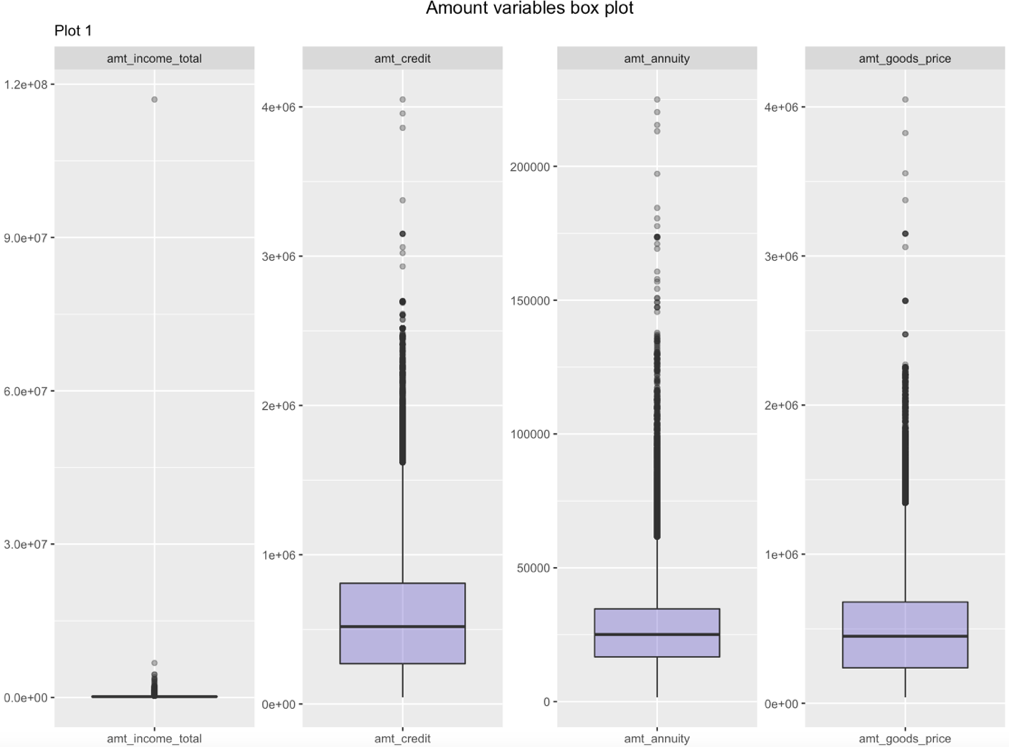
v) The Figure.2 below describes the box plot for all the four amount features present in the dataset. We can clearly see that the values are skewed towards the upper bound values. The desired approach to handle these outliers is to extract all the instances where instances lie above 99.995% quantile value and examine the instances case-by-case basis and remove them if they seem implausible. For the amount “amt\_income\_total” feature there are 5 instances where the value is greater than 2 Million and their occupation belongs to laborers, drivers and the values are not consistent with other amount features so the instances are removed. The instances above 99.995% quantile values for the features “amt\_goods\_price” “amt\_annuity”, “amt\_credit” are examined and the instances seem plausible therefore they are not removed.

Figure 2: Amount variables box plot for outliers.

vi) The feature “amt\_req\_credit\_bureau\_qrt” is the number of enquiries made

to credit bureau by the client 3 months prior to the application. There is an instance where the value is “261” whereas the maximum value for the feature excluding this instance is “8” hence the value is replaced with enquiries made to credit bureau by the client 1-month prior application feature (“amt\_req\_credit\_bureau\_mon”).

The outlier treatment resulted in removal of 13 instances which is roughly 0.005% loss of instances from the missing values handled dataset from previous section.

## 2.6. Data Preparation for PCA using Pyspark

### Diagram, text Description automatically generatedThe following section involves preparing the dataset for the requirements of unsupervised machine learning task PCA. The data requirements for the PCA requires converting all the factor levels to scaled numerical values which in turn leads to increase in dimensions of the dataset. Since there are several stages involved in this process the chosen approach is to implement “HPCI” technique of “Resilient Distributed Datasets (RDDs)” using “PySpark”.

The plan with in “Pypark” is to perform different data manipulation stages using spark API’s and execute all the stages using pipeline where the result is a spark dataframe that consists of scaled features and from there, we can then perform principal component analysis by selecting optimal number of components. The different stages in execution are presented in flow chart below (Fig.3):

After performing all the stages of the data preparation in spark the result is a dense vector that has 184 scaled predictor features with all the factor levels represented as a single feature with 248823 instances. The resultant sample of first row is presented in the appendix refer Table-8. The performed data manipulation techniques resulted in significant reduction of execution time by almost 62%, the same steps took 18 seconds in R whereas the spark executed within 6.7 seconds. Since R can handle the factor levels for supervised machine learning the following dataset will not be used for final model prediction, the dataset for the sampling and final machine learning prediction will be without one-hot encoded and unscaled values that is produced from before section with 89 features and 248823 instances.

Figure 3: Flow chart for data preparation processes.

## 2.7. Sampling for Supervised Machine Learning

The dataset consists of imbalanced target proportion i.e., clients who defaulted on regular repayments. The overall default rate in the dataset is 7.8% and therefore the chosen approach is to firstly randomly subsample the data with 70% being training data and 30% of testing data and use under sampling technique on the training data. Since the oversampling and SMOTE techniques are computationally expensive for the machine learning purpose and increase the volume of instances, under sampling is selected even though it can lead to potential loss of data. The random under-sampling on the training data is performed using “ROSE” package in R to randomly down sample the instances of majority-class (0) to match minority-class (1) such that the resultant training data consists of 50% of each target class.

# 3. Exploratory data analysis

The following section is focussed on exploring data by plotting some graphs followed by conducting principal component analysis using pyspark. For the graphical representation only few features that seems to be relevant to the research question are used and complete graphical analysis can be found in attached R file. Furthermore, the visual exploration and unsupervised learning models are conducted on non-sampled data.

## 3.1. Preliminary data exploration

Chart

Description automatically generatedi) The Fig 4. Demonstrates the kernel density plot for the features “ext\_source\_2” and “ext\_source\_3” which are normalised scores from external sources with values ranging between 0 and 1 colour coded by the target feature. Since the default rate is minor, kernel density plot is chosen to avoid overlapping problem faced by histograms. From the figure it is clear that for the feature “ext\_source\_3” the values for the non-defaulters are skewed towards left whereas the defaulters are showing the opposite pattern. However, for the “ext\_source\_2” we can that clear peak towards the right suggesting that greater the “ext\_source\_2” the clients are less likely to default, conversely the defaulters are almost showing a pattern of uniformity. Overall, the plot gives a clear distinguishing between target feature which can be useful for machine learning process.

Figure 4: Distribution of External source Features

Chart, treemap chart

Description automatically generatedii) The fig 5 is plotted from subset of the dataset in which clients had defaulted. The aim of the plot is to understand the patterns on client’s default frequency by hour and day of the week on which they applied loan. From the heatmap it can be observed that there is high frequency of defaulters between 10 A.M and 1P.M especially on Tuesdays, Wednesdays and Thursdays.

However, the same heatmap plotted on the subset on non-defaulters showed similar pattern the reason being peak timings, yet the plot gives useful information in

prioritising resources for the company to minimise the risk of false applications during these timings.

Figure 5: Defaulters frequency by week and hour of the day

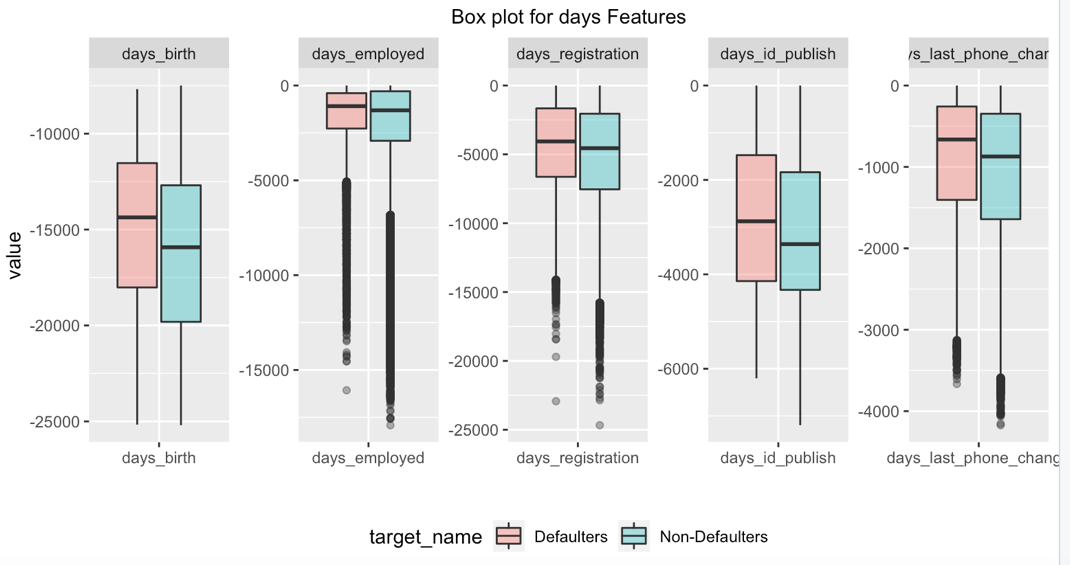
iii) The fig 6 displays the box plot for all the “days” features present in the dataset colour coded by target feature. The days are represented in negative values starting from the day before application date. For all the “days” features the median value for defaulters tends to be lower than it is for non-defaulters. For instance, “days\_birth” feature depicts younger people are more likely to default than older people. Similarly, the more recent client changed their identity document (“days\_id\_publish”) they are more likely to default. Moreover, the values for the features “days\_employed”, the inter quartile range falls between less than a year and 8 years with a maximum value of 49 years suggesting values heavily skewed towards greater values.

Figure 6: Box plot for days features colour coded by target feature

## 3.2. Principal Component Analysis

The following section is focussed on implementing PCA using “Pyspark” HPCI technique and the relevant ipython notebook is attached along with the report. Since we have 184 scaled predictor features from data preparation stage for PCA, there are 16836 possible pair-wise combinations between the features which can become highly complex to make sense of interrelationships. Henceforth the chosen approach is to implement PCA to 184 correlated features to reduce to uncorrelated composite features to retain as much information from original features (Kabacoff, 2015).

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedThe principal component analysis conducted resulted in 184 transformed scores (Table.8, appendix) for the original features and loadings for each component for feature. The next step is to select optimal number of principle components that capture as much information as possible. The plot 8 below demonstrates proportion of explained variance-PEV (left) and cumulative value of PEV (right) plotted against the principal component number. From the scree plot it can be seen that principal components below 22 are accounted for almost 25% of explained variance with elbow bend at component 13 and the explained variance decreases from thereafter as the components increase. However cumulative for the explained variance plot suggests that 109 components as optimal at 80% threshold limit hence 109 components are selected as optimal.

Figure 8: Cumulative value of PEV (right)

Figure 7:Scree plot (left)

Since it is difficult to interpret the 248823 instances with 184 features on a biplot, the approach is to generate a heatmap with loading for first 10 components along side features and explore the first 3 PCs by loadings. The complete heatmap can be found in appendix (Table.10)

i) **PC1** accounts for 3.5% of the total explained variance, From the heatmap (Table.10 appendix) the PC1 has high positive loadings for the features “occupation\_drivers”,” days\_employed”, “education\_incomplete\_higher” and negative loadings for “flag\_emp\_phone”,” income pensioner”,” cnt\_children”,” days\_birth”. Hence, we can say that PC1 is capturing information about the old clients that belong to lower education category with no formal employment history. The linear combination of this PC can be represented below, it can be interpreted as PC1 increases by a multiple of 0.13 for every unit decrease in scaled “days\_employed” feature since days are represented in negative values in dataset:

* PC1 = 0.28 \* (occupation drivers) + 0.13 \* (days\_employed) +0.07\* (education\_Incomplete higher) + .............. - 0.18 \* (income\_Pensioner) - 0.36 \* (flag\_emp\_phone) - 0.14 \* (cnt\_children) -0.27 \* (days\_birth)

ii) **PC2** accounts for the 3.09% of explained variance and the loadings positively correlates (Table.10 appendix) with features merged from bureau dataset and negatively correlates with “days\_employed” and “days\_last\_phone\_change” and “family\_type\_single”. It therefore appears to be capturing information about reliable clients with active credit history from the bureau with long employment history and are married for long time. The information captured by PC2 is quite converse to the first principal component where the information captured about clients with low education and employment and together, they account for 6% of total explained variance. Hence it appears to be the employment, education and age factor that is distinguishing between the components. The linear combination PC2 with identified features and weights are presented below:

* PC2 = 0.34 \* (bu\_prev\_app\_count) + 0.34 \* (bu\_credit\_currency\_currency\_1) + ............ +

-0.05 \* (ext\_source\_3) -0.06 \*(days\_employed) -0.08 \* (days\_last\_phone\_change)

iii) **PC3** accounts for 2.2% of explained variance and the loadings positively correlates with “amt\_goods\_price”, “amt\_credit”, “amt\_annuity”, “bu\_prev\_app\_count” and negatively correlates with “hc\_prev\_app\_count”, “amt\_req\_credit\_bureau\_year”. From the loadings it can be observed that the PC3 capturing information about new clients to the “Home credit” company but has active credit history from the bureau, furthermore it captures information about consistency in amount features present in the dataset. The linear combination PC3 with features and weights are presented below:

* PC3 = 0.21 \* (amt\_goods\_price) + 0.20 \* (amt\_credit) + 0.18\* (bu\_prev\_app\_count) + 0.17 \* (amt\_annuity) +......... -0.23 \* (amt\_req\_credit\_bureau\_year) -0.31\* (hc\_prev\_app\_count).

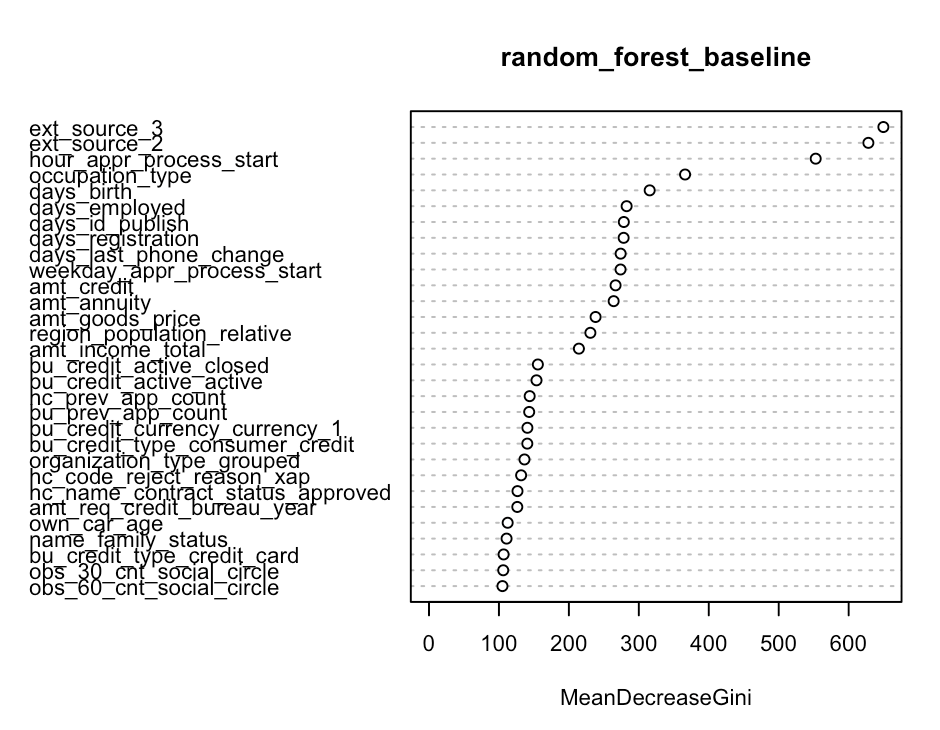
The PCA conducted has few limitations, firstly the objective to achieve optimal components significantly differ from the scree plot to the cumulative PEV threshold plot. The scree plot suggests 13 components as optimal by visual method at the same time 80% cumulative threshold suggest 109 as optimal number. Since the chosen method is 80% threshold, the objective to reduce dimensions from 184 features is not achieved and we are left with 109 components which is still a large number to interpret. Secondly with 109 components there still possible 5886 pairwise relations between principal components which is significantly large number to interpret.

# 4. Machine learning Implementation

The following section is focussed on deploying random forest classifier model on under sampled training data by using two different approaches. Firstly, to implement random forest classifier on all the features present in the dataset with default parameters as a baseline model. Secondly, the baseline model is fine-tuned with parameters. Then the performance of these two models is compared and evaluated in next section. The dataset used for the following machine learning is without one hot encoded features and unscaled values resulting dataset that consists of 88 features removing unique id feature.

## 4.1. Random Forest Baseline Model

The baseline random forest model is built on all the 87 predictor features and binary target feature with 50 trees and default “mtry” value. The “mtry” value which specifies number of random features to be selected for each tree is 9 for the model, by default R chooses the square root of number of predictor features, in this case Sqrt (87) = 9. The Out of Bag (OOB) error performed on 2/3rd of training data is 34.77% for 50 trees (James, Witten, Hastie and Tibshirani, 2013). The complete model output is presented in the appendix Table.11.

The (Fig.8) depicts feature importance for the model with mean decrease in Gini value, from the figure “ext\_source\_3”, “ext\_source\_2” and “hour\_appr\_process\_start” has highest purity in the node split with mean decrease Gini 649, 628 and 552

respectively. However, the node purity significantly drops after the first four features by 56% and it drops by 84% for 30th feature in the figure 8. This gives significant evidence that the top 10 features represented in the figure extremely contribute to splitting criterion for the purity in the nodes. Hence the next step is to tune the baseline random forest model by selecting optimal “mtry” value with minimum OOB error and optimal number of trees on the top features represented in the figure 8.

Figure 8: Feature Importance.

## 4.2. Random Forest Parameter Tuned Model

The parameter tuning is based on building model on top 30 features with mean decrease

value and to find optimal number of “mtry” and “trees”.

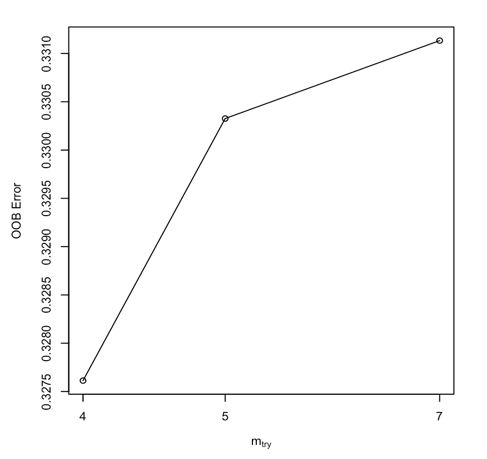
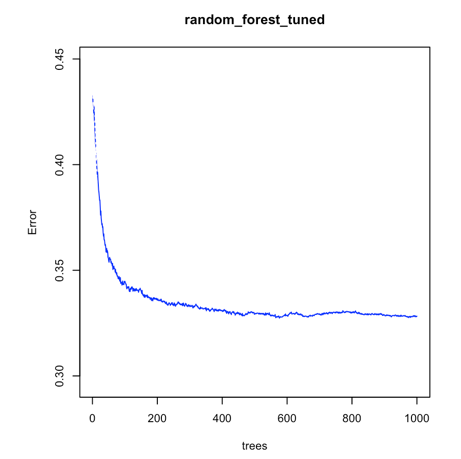
To find the optimal number of random features to be considered for each tree in the random forest, the plan is to train the model with 1000 trees by inflating mtry with a magnitude of 1.5 at each iteration till it reaches the minimum OOB error. The output of the tuning is presented in Fig.9, we can see that the model captured minimum “mtry” value at 4 with OOB error of 0.3282. The second step is to find number of trees to be considered. The fig. 10 describes the OOB error plotted against the 1000 trees within random forest. The aim is to find a value on x-axis where the error stabilizes, at around 700 trees the OOB starts to stabilise. The next step is to train the model on “mtry” value of 4 and 700 trees parameters. The output of model (Appendix – Table.12) trained on tuned parameters resulted in decrease of OOB error by 1.71%. Considering the fact, the number of predictors used the model performed better than the baseline model. An attempt to interpret random forest by extracting rules using “inTree” package in R made by 700 trees are presented below. There are 6005 rules generated and a sample is presented in the appendix Table 13.

Figure 10: Find optimal trees.

Figure 9: Find optimal "mtry" value.

There is 20% default rate when the “ext\_source\_3” is less than 0.48, client employment less than 9 years and the price of the goods for which is loan is taken is less €677,441. This rule has an error rate of 29% i.e., out of all instances that satisfy the condition the mis-classification rate is 29%. Similarly, the default rate is 16% with an error rate of 33% when client age is less than 40 years, employment less than 5 years and amount goods price in between €182,250 and €677,441. When the client is highly skilled employer with active past loan history from the bureau the non-default rate is 15% with 35% error rate. However, these target feature proportion measured is sensitive to interpretability because these rules are produced at each iteration and can differ when applying to whole data (Deng, 2018) and can change with changes in parameters. Moreover, the complexity within these 6005 rules generated makes random forests hard to interpret.

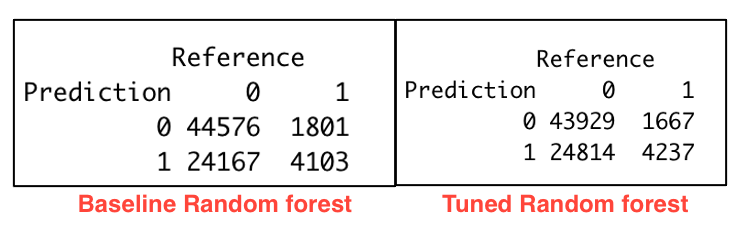
The above machine learning implementation has few limitations. Firstly, there is huge amount of information loss (approx. 80%) from the random under-sampling technique from the majority class which could be critical in establishing node purity and given the instances are picked at random it is hard to separate good from bad instances during sampling. Secondly reducing “mtry” value decreases both predictive power, correlation between trees and vice versa, the optimal range for which this value fall can be quite wide which can significantly impact random forest outcome. Finally, these ensemble algorithms are black box models which can be hard to interpret at the cost of predictive power.

# 5. Performance evaluation and comparison of methods

The following section evaluates the performance of the two machine learning models implemented in the previous section using confusion matrix, ROC – curve and few performance metrics.

## 5.1. Confusion Matrix

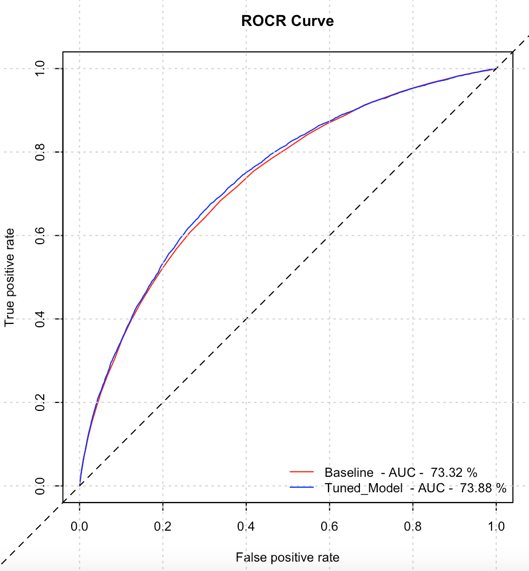
The confusion matrix presented in Fig.10 gives a clear depiction of basic performance of the model.

****The true negatives in the confusion matrix are represented as “1” i.e., defaulters that are predicted correctly by the model. The baseline predicted 4103 defaulters correctly, whereas the tuned model predicted 4237 instances out of 5904 total defaulters.

The overall accuracy of the baseline model is better than the tuned model with 65.22% over 64.52% respectively. However, accuracy can be misleading especially with imbalanced datasets because accuracy can be 100% if model predicts only one class correctly in the testing data.

Figure 10: Confusion Matrix.

From the research question perspective, the best model is that maximises the precision by minimising the type I error and maximising recall by minimising the type II error. To quantify this issue F1 score gives a way to combine precision and recall into a single metric that capture both the requirements for an imbalanced classification problem. The F1 score for the baseline model and tuned model is 0.76 and 0.77 respectively (Table 14 & 15 appendix). The tuned model slightly outperformed the baseline model but not significant enough. However, one of the limitations of threshold metrics is that they assume the same class distribution for training and testing datasets which is not the case in our machine learning implementation (Brownlee, 2021).



## 5.2. ROC – Curve

A receiver operating characteristic curve is plot between true positive rate and false positive rate, Area Under the Curve (AUC) is a portion of the area below curve of model which can quantify performance (Fawcett, 2006). The red line in our ROCR plot (Fig 11) illustrates baseline random forest classifier for our model, blue line represents tuned random forest classifier model, and the black line represents model being at random. From the ROC curve the objective is to maximize TPR (correctly classifying good applications) and minimising FPR (incorrectly classifying all bad applications). The tuned model is slightly performing better than baseline model with area under curve of 73.88%.

# 6. Discussion of the findings

Figure 11: ROC curve

Credit default prediction is one of the primary risks faced by financial institutions around the world.

The projects aim to implement a random forest machine learning to solve the problem in two different ways. The findings and limitations for the implementation are discussed below.

The performance from the machine learning is not as expected, the difference between baseline and tuned model turned out to be very minimal and is significantly impacted by the under-sampling technique. The plan to implement SMOTE technique became extremely computational to deploy because of the volume of the data and the random under sampling technique resulted in loss of huge amount of the data. Secondly the features selected from secondary datasets are very less, those features that are categorical and describe about the type of applications client had and their rejection status are used and most of the other features are neglected because of the complications in grouping and merging numerical features at Loan ID level which could have potentially impacted the performance of the machine learning model.

Moreover, the external source features showed very highly significance at the feature importance from random forest and dropping one of them (“ext\_source\_1”) during missing value treatment due to high missing percentage could have affected the result of the machine learning. Even though the principal component analysis using pyspark resulted in useful information extracted from the components but created challenges in choosing appropriate components. The interpretation of components became difficult using traditional methods because of large number of one hot encoded features. Finally, it is observed that choosing traditional methods for performance evaluation can lead to misleading especially when testing on imbalanced datasets and require business context to chose appropriate metric hence F1-score is chosen to compare performance.

# 7. Data Management Plan and Author Contribution statement

The whole assignment is contributed by the Student ID – 2042729 and the reasons are discussed with the group tutor in the meetings. The relevant data management plan file is attached with the following document.

# 8. References

* Brownlee, J., 2020. Tour of Evaluation Metrics for Imbalanced Classification. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/> [Accessed 3 April 2021].
* Deng, H., 2018. Interpreting tree ensembles with inTrees. International Journal of Data Science and Analytics, 7(4), pp.277-287.
* Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters, 27(8), pp.861-874. \*
* James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An introduction to statistical learning.
* Kabacoff, R., 2015. R in action. Shelter Island, NY: Manning Publications.
* Kaggle.com. 2018. Home Credit Default Risk | Kaggle. [online] Available at: <https://www.kaggle.com/c/home-credit-default-risk/data> [Accessed 27 March 2021].
* Xu, C., 2018. Why is Dimensionality Reduction so Important? [online] Medium. Available at: <https://medium.com/@cxu24/why-dimensionality-reduction-is-important-dd60b5611543> [Accessed 27 March 2021].

# 9. Appendix

Table 1: Meta data of Application data

|  |  |
| --- | --- |
| Applications\_data.csv | |
| Row | Description |
| SK\_ID\_CURR | ID of loan in our sample |
| TARGET | Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y instalments of the loan in our sample, 0 - all other cases) |
| NAME\_CONTRACT\_TYPE | Identification if loan is cash or revolving |
| CODE\_GENDER | Gender of the client |
| FLAG\_OWN\_CAR | Flag if the client owns a car |
| FLAG\_OWN\_REALTY | Flag if client owns a house or flat |
| CNT\_CHILDREN | Number of children the client has |
| AMT\_INCOME\_TOTAL | Income of the client |
| AMT\_CREDIT | Credit amount of the loan |
| AMT\_ANNUITY | Loan annuity |
| AMT\_GOODS\_PRICE | For consumer loans it is the price of the goods for which the loan is given |
| NAME\_TYPE\_SUITE | Who was accompanying client when he was applying for the loan |
| NAME\_INCOME\_TYPE | Clients’ income type (businessman, working, maternity leave) |
| NAME\_EDUCATION\_TYPE | Level of highest education the client achieved |
| NAME\_FAMILY\_STATUS | Family status of the client |
| NAME\_HOUSING\_TYPE | What is the housing situation of the client (renting, living with parents? ...) |
| REGION\_POPULATION\_RELATIVE | Normalized population of region where client lives (higher number means the client lives in more populated region) |
| DAYS\_BIRTH | Client's age in days at the time of application |
| DAYS\_EMPLOYED | How many days before the application the person started current employment |
| DAYS\_REGISTRATION | How many days before the application did client change his registration |
| DAYS\_ID\_PUBLISH | How many days before the application did client change the identity document with which he applied for the loan |
| FLAG\_MOBIL | Did client provide mobile phone (1=YES, 0=NO) |
| FLAG\_EMP\_PHONE | Did client provide work phone (1=YES, 0=NO) |
| FLAG\_WORK\_PHONE | Did client provide home phone (1=YES, 0=NO) |
| FLAG\_CONT\_MOBILE | Was mobile phone reachable (1=YES, 0=NO) |
| FLAG\_PHONE | Did client provide home phone (1=YES, 0=NO) |
| FLAG\_EMAIL | Did client provide email (1=YES, 0=NO) |
| OCCUPATION\_TYPE | What kind of occupation does the client have |
| CNT\_FAM\_MEMBERS | How many family members does client have |
| REGION\_RATING\_CLIENT | Our rating of the region where client lives (1,2,3) |
| REGION\_RATING\_CLIENT\_W\_CITY | Our rating of the region where client lives with taking city into account (1,2,3) |
| WEEKDAY\_APPR\_PROCESS\_START | On which day of the week did the client apply for the loan |
| HOUR\_APPR\_PROCESS\_START | Approximately at what hour did the client apply for the loan |
| REG\_REGION\_NOT\_LIVE\_REGION | Flag if client's permanent address does not match contact address (1=different, 0=same, at region level) |
| REG\_REGION\_NOT\_WORK\_REGION | Flag if client's permanent address does not match work address (1=different, 0=same, at region level) |
| LIVE\_REGION\_NOT\_WORK\_REGION | Flag if client's contact address does not match work address (1=different, 0=same, at region level) |
| REG\_CITY\_NOT\_LIVE\_CITY | Flag if client's permanent address does not match contact address (1=different, 0=same, at city level) |
| REG\_CITY\_NOT\_WORK\_CITY | Flag if client's permanent address does not match work address (1=different, 0=same, at city level) |
| LIVE\_CITY\_NOT\_WORK\_CITY | Flag if client's contact address does not match work address (1=different, 0=same, at city level) |
| ORGANIZATION\_TYPE | Type of organization where client works |
| EXT\_SOURCE\_2 | Normalized score from external data source |
| EXT\_SOURCE\_3 | Normalized score from external data source |
| OBS\_30\_CNT\_SOCIAL\_CIRCLE | How many observations of client's social surroundings with observable 30 DPD (days past due) default |
| DEF\_30\_CNT\_SOCIAL\_CIRCLE | How many observations of client's social surroundings defaulted on 30 DPD (days past due) |
| OBS\_60\_CNT\_SOCIAL\_CIRCLE | How many observations of client's social surroundings with observable 60 DPD (days past due) default |
| DEF\_60\_CNT\_SOCIAL\_CIRCLE | How many observations of client's social surroundings defaulted on 60 (days past due) DPD |
| DAYS\_LAST\_PHONE\_CHANGE | How many days before application did client change phone |
| FLAG\_DOCUMENT\_2 | Did client provide document 2 |
| FLAG\_DOCUMENT\_3 | Did client provide document 3 |
| FLAG\_DOCUMENT\_4 | Did client provide document 4 |
| FLAG\_DOCUMENT\_5 | Did client provide document 5 |
| FLAG\_DOCUMENT\_6 | Did client provide document 6 |
| FLAG\_DOCUMENT\_7 | Did client provide document 7 |
| FLAG\_DOCUMENT\_8 | Did client provide document 8 |
| FLAG\_DOCUMENT\_9 | Did client provide document 9 |
| FLAG\_DOCUMENT\_10 | Did client provide document 10 |
| FLAG\_DOCUMENT\_11 | Did client provide document 11 |
| FLAG\_DOCUMENT\_12 | Did client provide document 12 |
| FLAG\_DOCUMENT\_13 | Did client provide document 13 |
| FLAG\_DOCUMENT\_14 | Did client provide document 14 |
| FLAG\_DOCUMENT\_15 | Did client provide document 15 |
| FLAG\_DOCUMENT\_16 | Did client provide document 16 |
| FLAG\_DOCUMENT\_17 | Did client provide document 17 |
| FLAG\_DOCUMENT\_18 | Did client provide document 18 |
| FLAG\_DOCUMENT\_19 | Did client provide document 19 |
| FLAG\_DOCUMENT\_20 | Did client provide document 20 |
| FLAG\_DOCUMENT\_21 | Did client provide document 21 |
| AMT\_REQ\_CREDIT\_BUREAU\_HOUR | Number of enquiries to Credit Bureau about the client one hour before application |
| AMT\_REQ\_CREDIT\_BUREAU\_DAY | Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application) |
| AMT\_REQ\_CREDIT\_BUREAU\_WEEK | Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application) |
| AMT\_REQ\_CREDIT\_BUREAU\_MON | Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application) |
| AMT\_REQ\_CREDIT\_BUREAU\_QRT | Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application) |
| AMT\_REQ\_CREDIT\_BUREAU\_YEAR | Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application) |

Table 2: Meta data of bureau data

|  |  |
| --- | --- |
| Bureau\_data.csv | |
| Row | Description |
| SK\_BUREAU\_ID | Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application) |
| CREDIT\_ACTIVE | Status of the Credit Bureau (CB) reported credits |
| CREDIT\_CURRENCY | Recoded currency of the Credit Bureau credit |
| CREDIT\_TYPE | Type of Credit Bureau credit (Car, cash,) |

Table 3: Meta data of previous\_applications.csv

|  |  |
| --- | --- |
| Previous\_applications.csv | |
| Row | Description |
| SK\_ID\_PREV | ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loan applications in Home Credit, previous application could, but not necessarily have to lead to credit) |
| NAME\_CONTRACT\_STATUS | Contract status (approved, cancelled, ...) of previous application |
| CODE\_REJECT\_REASON | Why was the previous application rejected |

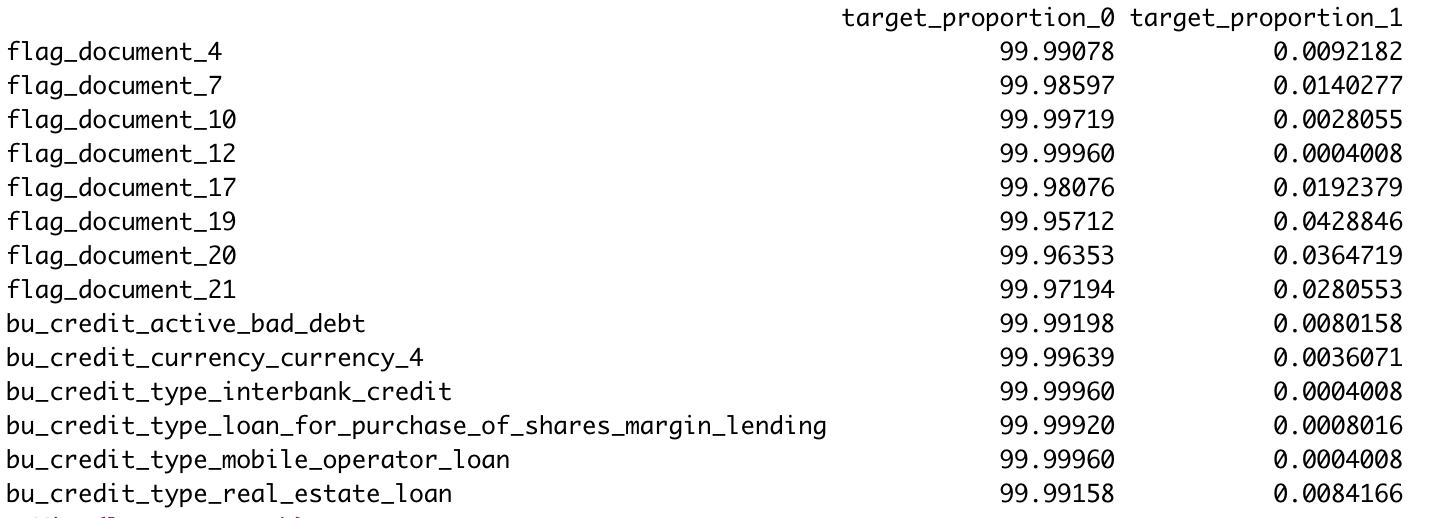
Table 4: Contingency table for features with two distinct values with target feature

Table 5: Chi square test table with features above p-value of 0.05.

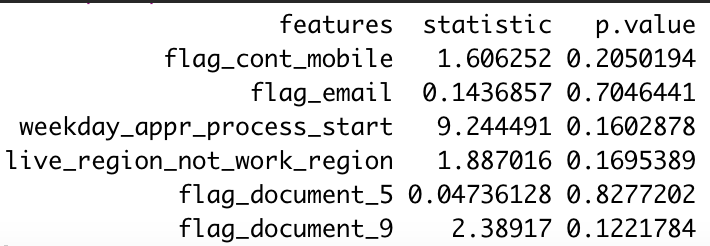


Table 6: Linear model for amount annuity

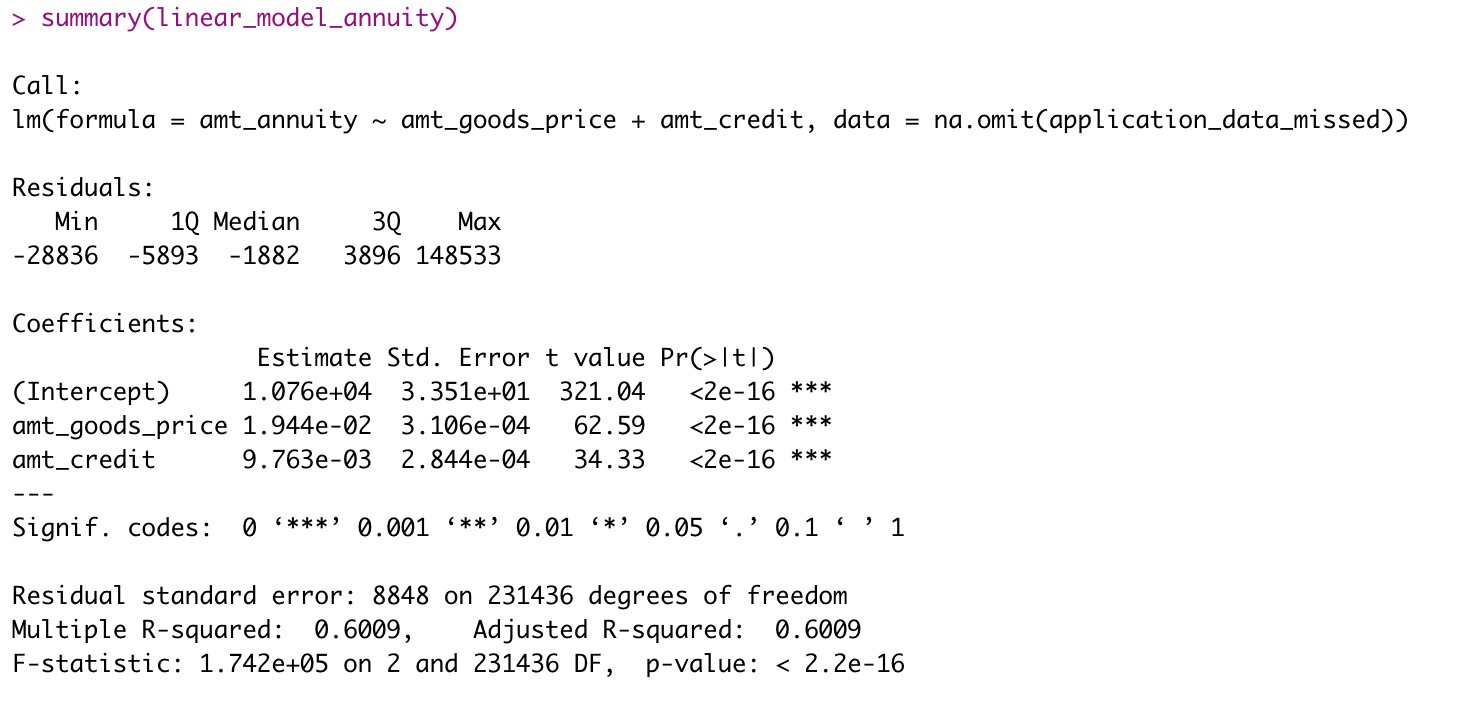


Table 7:Linear model for amount annuity goods

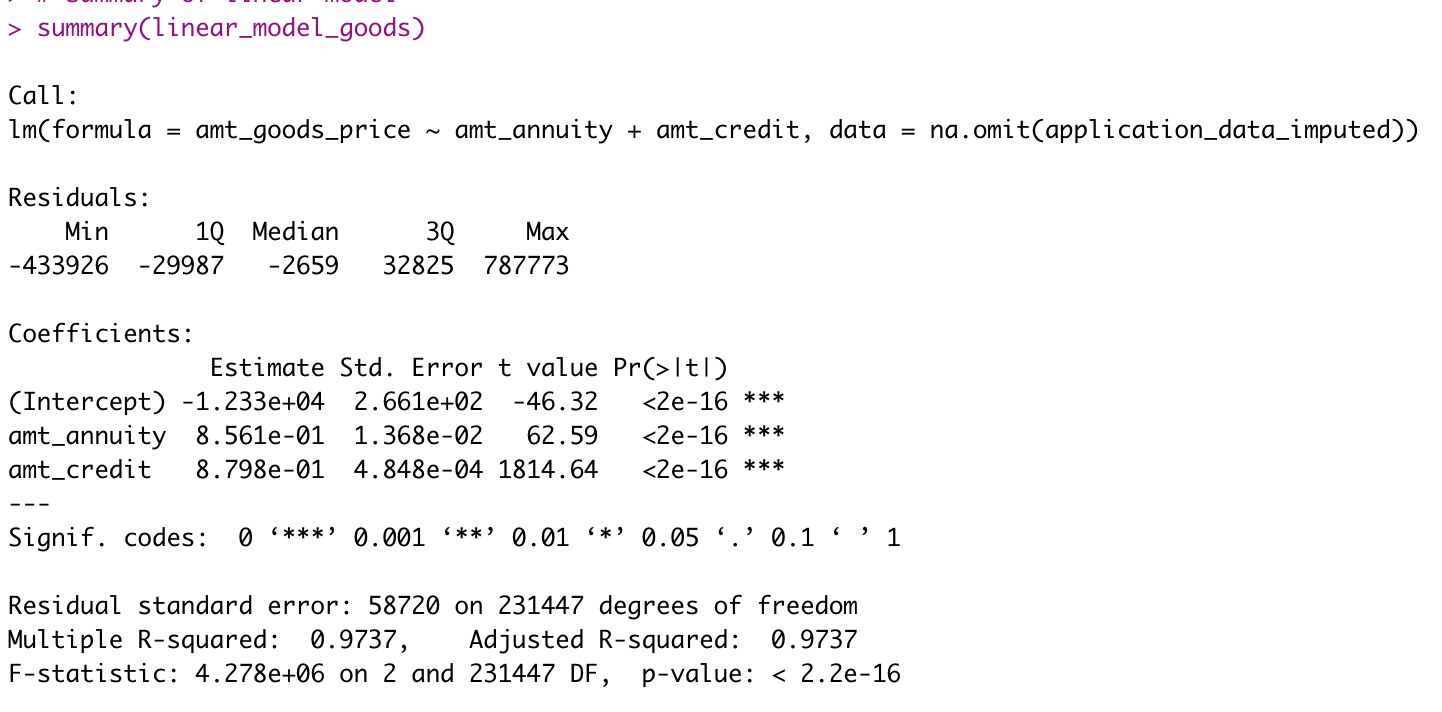


Table 8: first row of scaled data obtained from pyspark.

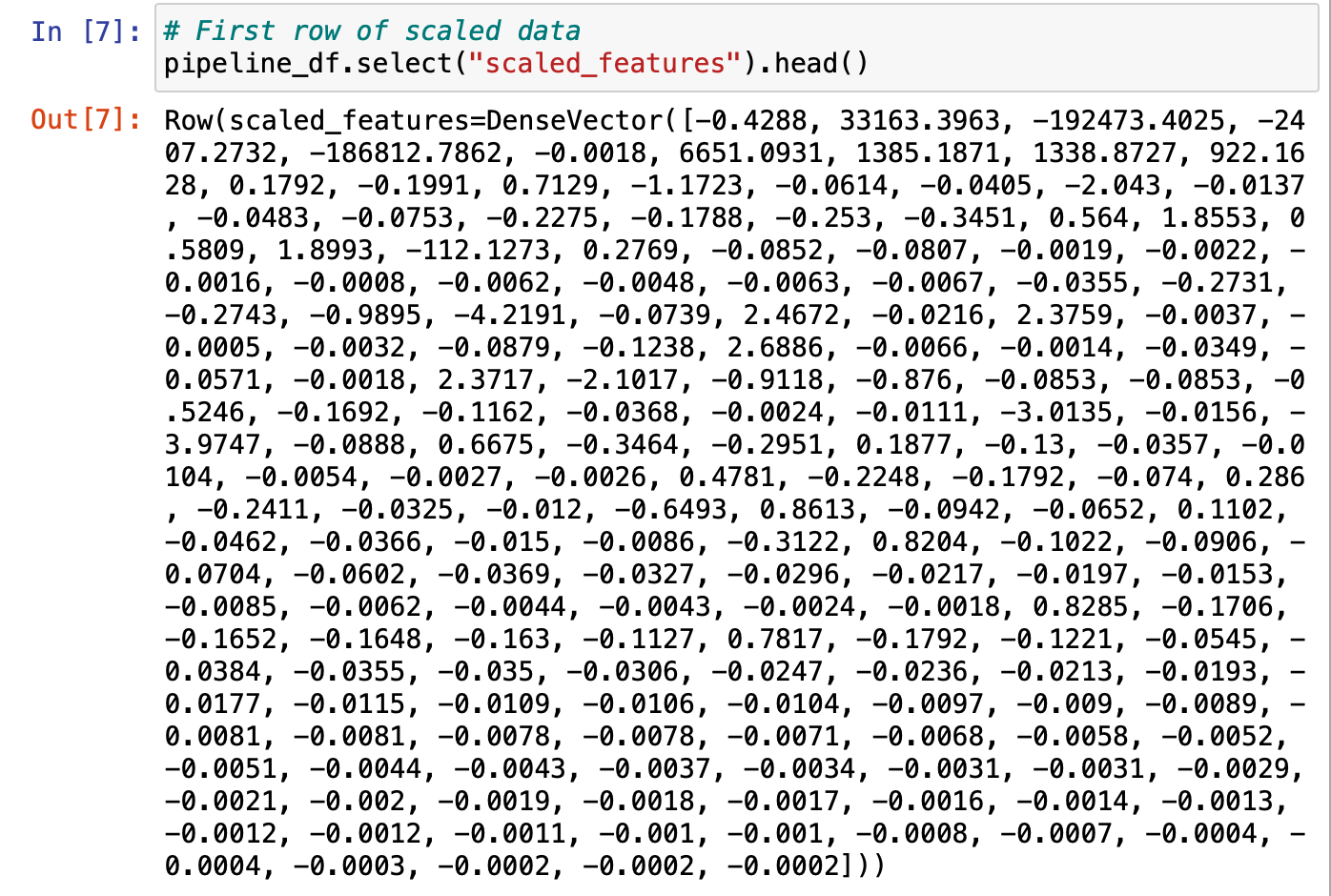


Table 9: Scores from the PCA

Text, letter

Description automatically generated

Table 10: Loadings for first 10 principal components

A close-up of a graph

Description automatically generated with low confidence

Table 11: Random forest Baseline output.

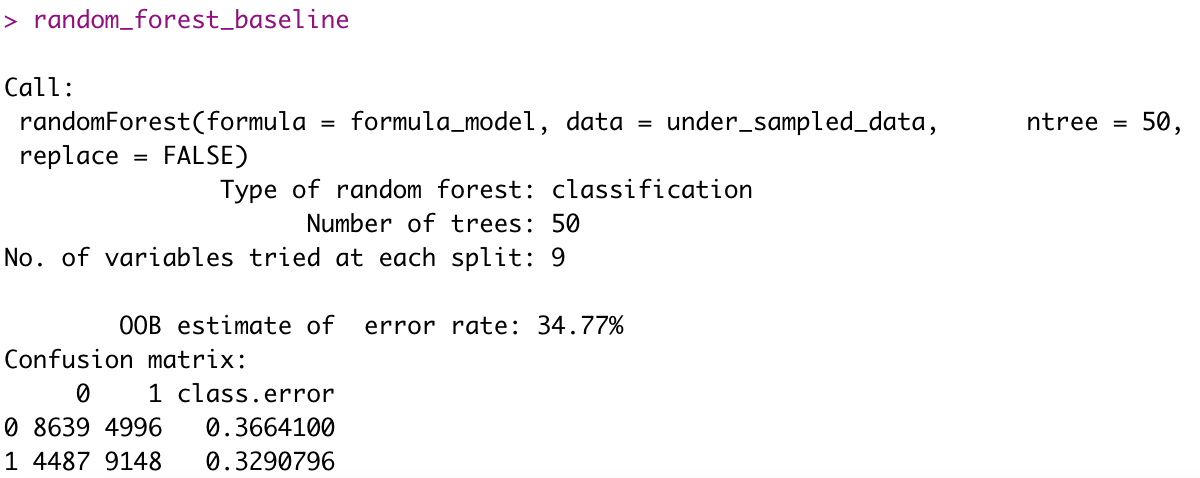


Table 12:: Random forest Tuned output

Text, letter

Description automatically generated

Table 13: Sample rules generated from tuned random forest model.

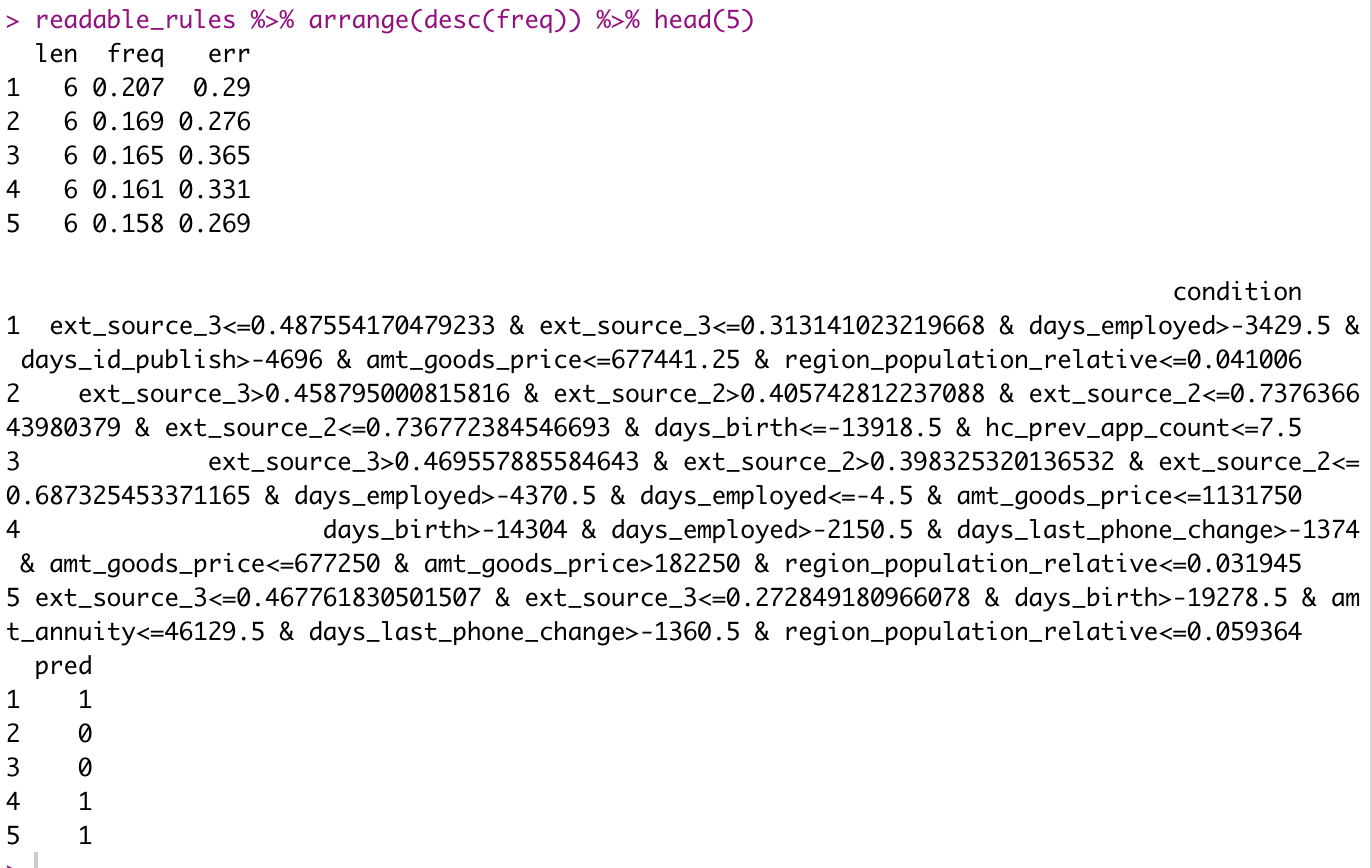


Table 14: Performance metrics for baseline model

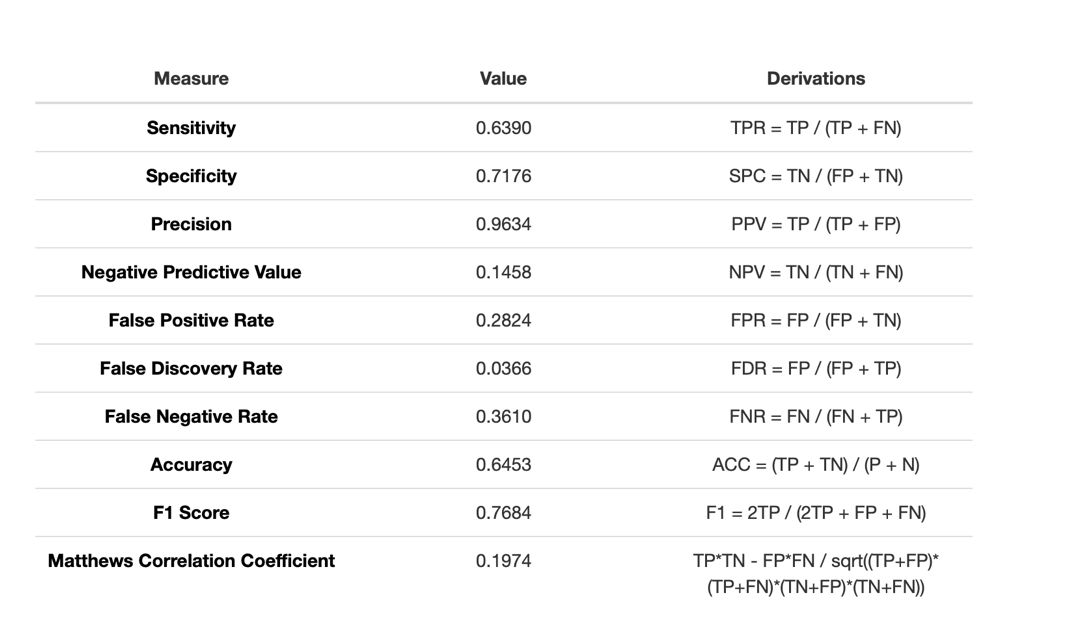


Table 15:Performance metrics for tuned model

