Università di Pisa

Distributed Analysis and Mining of a Bank Loan Status Dataset

Data Preparation, Data Understanding and Classification in a Hadoop/Spark Computing Framework

Ferri Lorenzo (607828)
email: l.ferri11@studenti.unipi.it

Ilic Ema (602796)
email: ema.ilic9@gmail.com

Pappolla Roberta (534109)
email: r.pappolla@studenti.unipi.it

Distributed Data Analysis and Mining (687AA), Academic Year 2020/2021

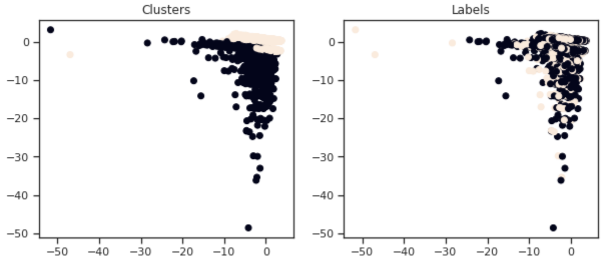
Contents

1 Context

The dataset was taken from the Kaggle website. The goal of the Analysis is to predict whether the person in question will default on paying his debt or not, based on the features of the debtor/loan combination available in the dataset. Even though the datasets seemed clean at a first glance, after the careful analysis it was clear that some serious data preparation was needed due to numerous missing values, outliers and errors. This will be tackled in the second chapter (Data Preparation, Clustering and Feature Engineering) of this report. The subsequent chapter (Data Understanding and Dimensionality Reduction), on the other hand, discusses the final, cleaned dataset which was used as an input for the Machine Learning methods (Chapter 4).

2 Data Preparation, Clustering and Features Engineering

All the main changes to the dataset were made with RDD’s and are well documented in the notebook ’1_DataPreparation_Clustering_FeaturesEngineering’. They mainly concern modifications to the attributes in order to make them more consistent, e.g. by changing the currency denomination of the money quantities from Rubles to Euros and by adjusting some attributes values to make them numeric rather than strings (e.g. values ’10+ years’ or ’<1 year’ in ’Years_In_Current_Job’). Some errors, such as the erroneous ’0’s at the end of the ’Credit_Score’ quantities, ’99,999,999’ value of ’Current_Loan_Amount’ and the duplicate rows have been corrected as well. After these improvements the dataset was characterized by the uniqueness of the ID’s (none of ’Loan_ID’ and ’Customer_ID’ were duplicated), and they were dropped. The only thing left to do was to fill the missing values. The following strategy were chosen: Attributes with few Missing Values have been filled with the Median of some given sensible grouping sets; the others have been filled againg with the Median of some grouping sets and by adding to these groupings the Label of the best clustering we found. The best clustering was found fitting models on the Standardised Numerical Features of the entire dataset and searching among three different models and several K’s iterations. This Clustering turned out to be a "Gaussian Mixture Model" with 2 Clusters, achieving a Silhouette Score of 0.778. The resulting plot of both the clusters and the dataset output class (for comparison purposes) is displayed below (the dataset dimensionalities are reduced with the principal component analysis).



Further features were engineered from the existing ones: ’Installment_Rate’, ’Debt_Income_Rate’, and finally
’Credit_Problems_Perc’, that was dropped later on after having observed a too high correlation with the variable ’Number_of_Credit_Problems’ from which it derives.
The final dataset was exported to the HDFS in the "Parquet" (columnar store) format.

3 Data Understanding and Dimensionality Reduction

3.1 Variable Interpretations and Types

Even though all the definitions can be found in the notebook, we provide the definitions of the most prominent variables here.
’Loan_Status’: The label in output: Whether the Loan has been paid off or not(categorical, class variable).
’Credit_Score’: the lower the Credit Score, the more likely the debtor is to default (numerical continuous).
’Current_Credit_Balance’: the amount of money that a client of a financial institution has in his or her account (numerical continuous).
’Current_Loan_Amount’: Loan Amount is the amount the borrower promises to repay, as set forth in the loan contract (numerical continuous).
’Maximum_Open_Credit’: An open credit is a financial arrangement between a lender and a borrower that allows the latter to access credit repeatedly up to a specific maximum limit (numerical continuous).
’Number_of_Credit_Problems’: Credit problems include lack of enough credit history, denied credit application, fraud and identity theft (numerical discrete).
’Tax_Liens’: A tax lien is a lien imposed by law upon a property to secure the payment of taxes. A tax lien may be imposed for delinquent taxes owed on real property or personal property, or as a result of failure to pay income taxes or other taxes (numerical discrete).
’Term’: Time the debtor has to pay off the debt. Categorical variable taking on only two values: Short or Long Term (categorical).
’Installment_Rate’: the percentage of the Monthly

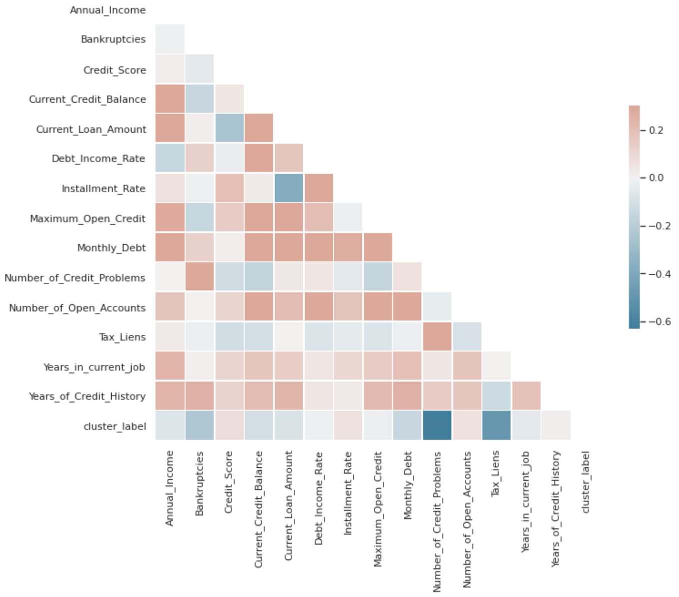
Debt over the overall Loan Amount;
'Debt_Income_Rate': the percentage of the Annual Debt over the Annual Income;

3.2 Statistics and Exploarative Analysis

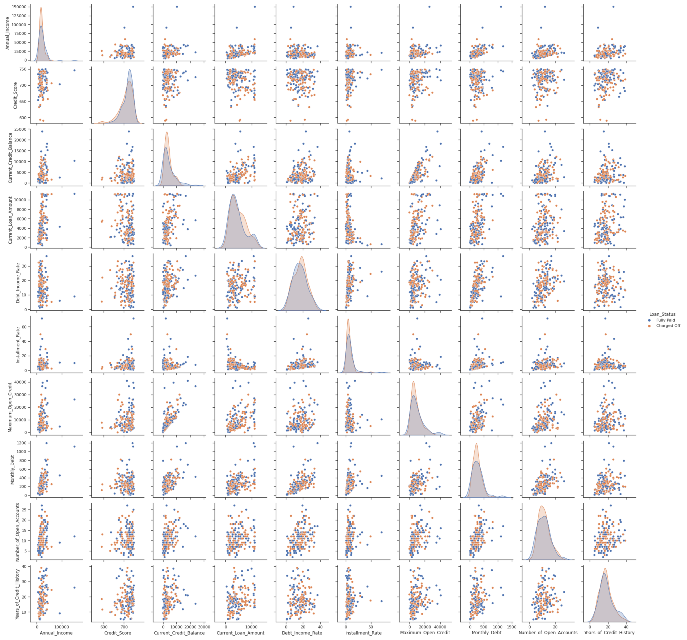
The clean training set contained 72997 values, of which 50783 (70%) rows had the dependent binary variable 'Loan_Status' as 'Fully Paid' and 22214 (30%) values as 'Charged Off'. Thus, the dataset was not considered imbalanced and no further balancing methods were used.

3.2.1 Numerical Variables

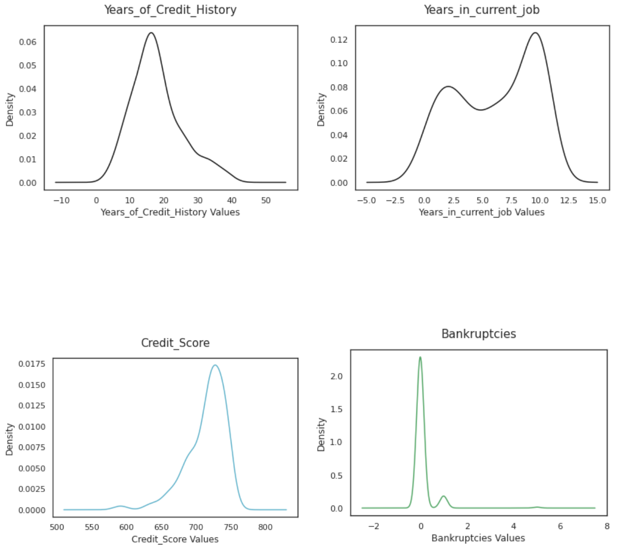
The statistical information relevant for each of the numerical values, such as average, median, number of distinct values, minimum and maximum value can be observed in the spark dataframe tables in notebook '2_Data_Understanding_&_Statistics'. For the sake of visualizing a Big Data scenario, Spark Dataframe was sampled to approximately 0.3% and the output classes were balanced to approximately 50:50. In figure below a correlation heatmap matrix can be observed displaying the correlations between different attributes. As no pair of attributes had a particularly high correlation, there was no need for eliminating any attributes.



Another interesting visualization is this scattermatrix, representing the sampled plot of the most prominent numerical attributes.

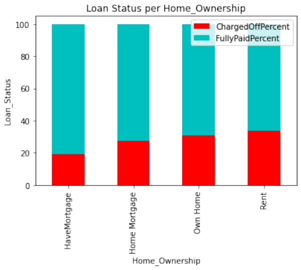


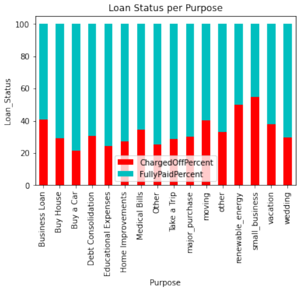
Additionally, four prominent density plots were chosen and included in the report, but for the sake of the conciseness of the report, only two will be discussed. Namely, we can observe how both 'Years_in_Current_Job' and 'Bankruptcies' have a bimodal distribution. From the small sample of only 178 values, most debtors had no previous bankruptcies, whereas a very small number had one bankruptcy, and an even smaller number of debtors had five bankruptcies. In this case, these are considered outliers. The peak at the Years_in_Current_Job equal to 10 is explained by the fact that in the original dataset, this attribute used to be a categorical string, and as mentioned in data preparation, all of the '10+ years' values were mapped to a number 10. Thus, one could say that the true mean is the one at approximately 2 years.



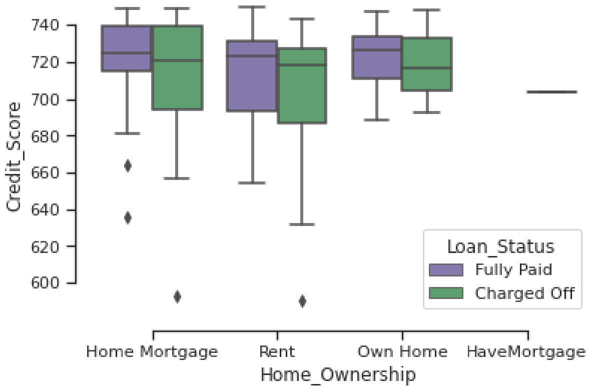
3.2.2 Categorical Variables

The figures below are worth noting because they were obtained from the whole dataset using the Spark SQL Querys rather than the sampled dataset. Different types of home ownership and purposes can be observed with respect to the class variable. These plots showed the percentage of the 'Charged Off' values to be higher for loans taken by small businesses, and lower when taken for paying off weddings of buying a car. Thus, these are less risky purposes, and more likely to be approved by the bank. On the other hand, Home_Ownership did not turn out to be a variable which discriminates a lot, even though small preference can be observed for those who have mortgage rather than does who rent.



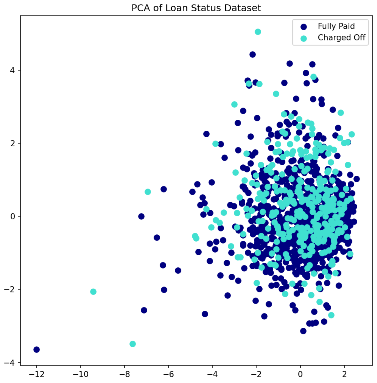


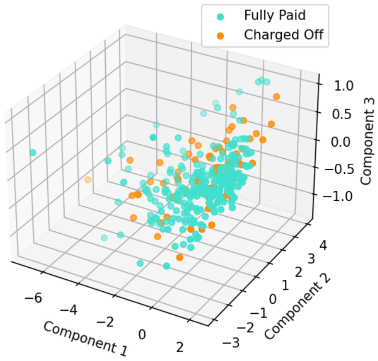
The boxplots below go more in-depth on the distribution of Credit Score for different Home Ownership categories with respect to the Loan Status variable, as well as its outliers.



3.3 Dimensionality Reduction with PCA

Dimensionality Reduction method called Principal Component Analysis was applied in order to see whether the Dataset can be reduced to k components preserving the highest variance without significant loss in accuracy when a classifier is applied. The results of 2-dimensional and 3-dimensional PCA analysis are displayed in figures below. Note that the plots portrayed here are different than the ones in the notebook: Namely, the plots in this report do not contain the numerical attributes with categorical-looking values (e.g. 'Years_in_current_job', 'Tax_Liens'...) which distort the visualizations.





The results in terms of accuracy were impressive. Namely, it was seen that the number of dimensions can be cut down to a half (from the original 14 down to 7) at the cost of just one percent of accuracy! As much of a revelation that this is, it was decided that the further analysis will be continued with the non-reduced dataset.

4 Classification and Regression

A function was defined with the scope of preparing the dataset in different ways for Machine Learning analysis. This function, when needed, encodes categorical variables, standardizes the dataset, vectorizes numerical features and encodes the dependent (class) variable. These transformations have been always applied to the training set and test set, after having fitted the transformations Pipeline itself only on the entire Dataframe. This fact is important in order to make transformed values (eg. scaled values) coherent between each other in both the training and test sets.

4.1 Classification for predicting 'Loan_Status'

The classification was conducted in ML for the most part, even though also MLlib was tested. Grid search 3-fold Cross Validation was used in order to find the right parameters for all of the classifiers. Parallelism hyperparameter in the grid search function was set to 10, which means that 10 models will be trained and evaluated simultaneously each time to make better use of cluster resources. Finally, below each model we can observe the hyperparameters chosen as the best by the grid search function by maximizing the Area Under the Curve metric. All the results can be observed in the '3_Classification_Regression' notebook.

For the Numeric classifiers (Logistic Regression and Support Vector Machines), only the numeric columns were isolated, vectorized and standardized, in order to be used for training and testing the models.

For mixed Categorical and Numerical Classifiers (Naive Bayes, Decision Tree and Random Forest) all of the categorical features were encoded and the resulting numeric features (including obviously the original numeric ones) were vectorized together.

4.1.1 Results

Classifier	Accuracy	Precision	Recall	F1
Logistic Regression (ML)	0.698	0.654	0.698	0.592
Support Vector Machines	0.696	0.484	0.696	0.571
Naive Bayes Classifier	0.436	0.617	0.436	0.427
Decision Tree Classifier	0.6	0.603	0.6	0.602
Random Forest Classifier	0.701	0.666	0.701	0.606
Logistic Regression (MLlib)	0.587	0.645	0.575	0.605

Some of the more prominent hyperparameter settings relevant for each of the classifiers are stated below:

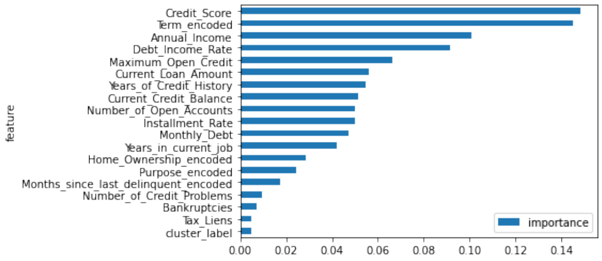
Logistic Regression: Elastic Net Mixing Parameter: 0, Regularization Parameter: 0.01;

Naive Bayes Classifier: smoothing parameter: 0.7

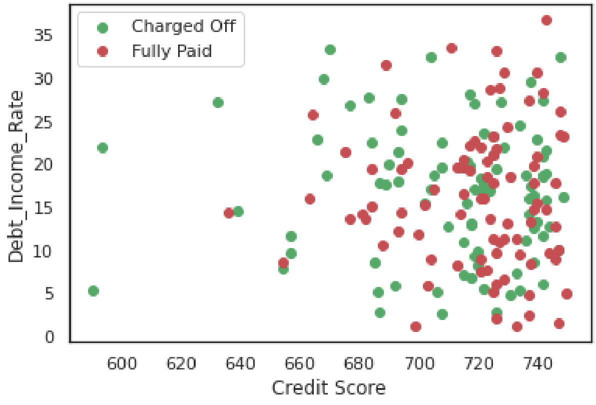
Decision Tree Classifier: Maximum Depth of the Tree: 30, Impurity measure: Entropy;

Random Forest Classifier: Maximum depth: 10, Number of trees to train: 30.

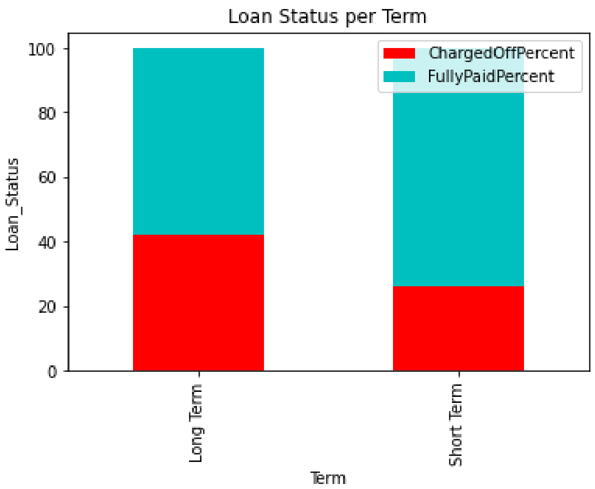
As can be seen from the above table, the classifier which obtained the best results for all the four measures is the Random Forest Ensemble classifier. Thus, the composition of all the trees in the model can be found in the Jupyter notebook, and the feature importance analysis was conducted for this classifier. On the plot below, it is visible that 'Credit_Score' discriminates the 'Loan_Status' variable the best, followed by 'Term' and 'Annual_Income'. This is unsurprising as 'Credit_Score' is ultimately the measure which serves to decide whether the person gets the loan or not.



It is interesting to observe how the 'Debt_Income_Rate', a feature which was additionally engineered and 'Credit_Score' discriminate between the values of 'Loan_Status'. For higher 'Debt_Income_Rate' and Credit_Score, there are less 'Charged_Off' values. More importantly, rows with lower 'Credit_Score' values are more likely to take on 'Charged_Off' value of output.



As 'Term' was shown to be the second variable which discriminates the class in output the best, a graphical representation of the result follows: Long term debts have a higher relative rate of 'Charged Off' values, meaning that the long term debts are riskier.



Finally, as Random Forest came out as the best model, the next thing that was done on it was re-fitting the model on the PCA-reduced dataset. Namely, the best RF model was inserted in the pipeline with PCA, and 3-fold grid-search was applied to find out whether the dataset can be reduced without a significant trade-off in terms of relevant metrics. The results showed that the dataset can be reduced from 19 dimensions (numerical + encoded categorical values) down to 16 with absolutely no loss in terms of AUC. Fitting the model and calculating the remaining metrics on the reduced dataset, the results were confirmed: Accuracy, Precision, Recall and F1 changed only by the fourth decimal. Just as before with the Logistic Regression, PCA proved to be a successful tool in dealing with big data. Thus, the best model chosen to predict the 'Loan_Status' variable in this Machine Learning analysis is Random Forest reduced to 16 columns using PCA.

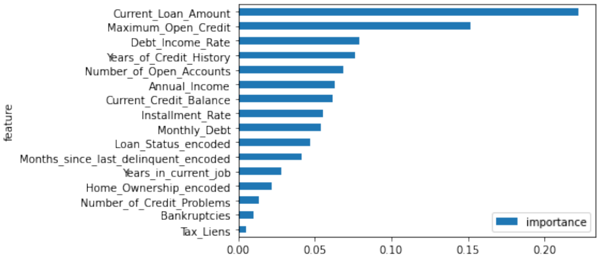
4.2 Linear and Random Forest Regression for predicting 'Credit_Score'

Since 'Credit_Score' was found to be the most important attribute in predicting the output label 'Loan_Status', a predictive analysis was performed on its values. The features used in the regression were only those which directly concern either the client, or the client's bank history. The features relevant to the nature of the debt ('Term', 'Purpose') were excluded. The models used were the 'Linear Regression' and the 'Random Forest Regressor', with two different Data Preparation steps: for the Linear Regression only the scaled numerical features were used, whereas for the Random Forest Regressor both numerical and encoded categorical features were used, but without the scaling step. Again, Grid Search was used in order to find the best parameters for the models. Here are the best parameters:

'Linear Regression': loss function type to be optimized in the algorithm: 'huber', regularization entity, 0.5;

'Random Forest Regressor' : number of trees of the Ensamble method: 30, maximum depth of the trees: 10.

Feature Importance analysis showed the most important feature to be 'Current_Loan_Amount', with an importance of 22.2% for the prediction of Credit_Score. All the importances are given in the following figure.



However, none of the regressors showed impressive results in terms of the relevant metrics: Namely, Linear Regressor and Random Forest Regressor obtained an R squared of -0.02 and 0.15, respectively. Thus, the conclusion is that a good regression model cannot be obtained with this dataset.