MS4610: Introduction to Data Analytics

(Jul-Nov 2020)

Project Report

'Loan Default Prediction using Machine Learning Models'

Course Instructor:

Dr. Nandan Sudarsanam

Submitted by:

Group 19

CE17B102	Dasi Manoj Kumar
CE17B110	Bennabhaktula Ritesh
CE17B116	Hrishikesh Gadekar
CE17B123	Pushpraj Singh Chouhan
CE17B127	Sai Charan Nalla
CE17B128	Santosh Kumar Mantripragada
EE17B132	Kammula Sri Hari Charan

Department of Management Studies

Indian Institute of Technology, Madras

January, 2021

1. Abstract

The dataset given for the project consisted of details of the customers who had taken loan. Based on multiple features available, our task is to predict whether a loan will go default or not.

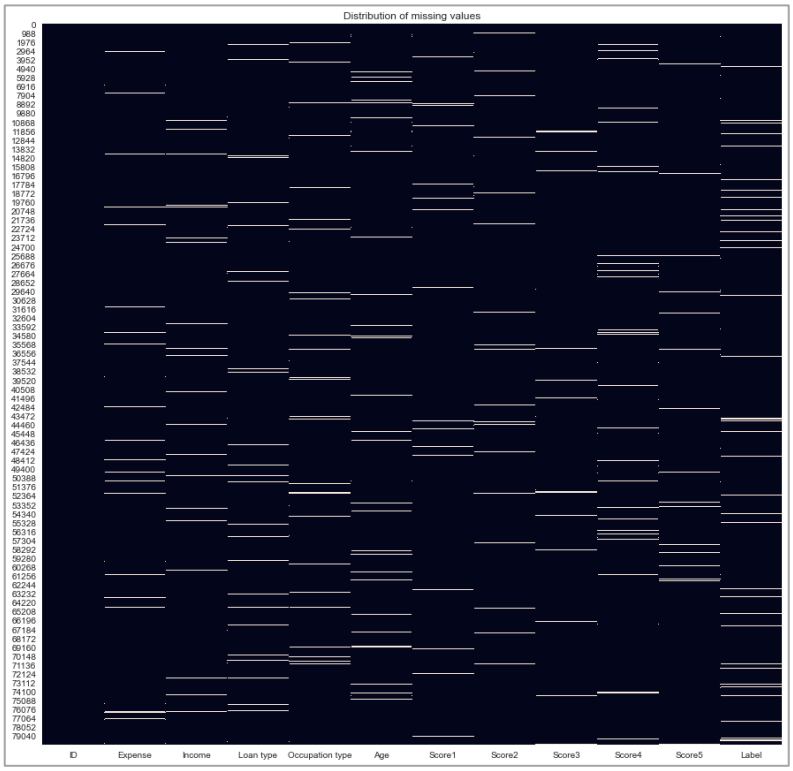
The analysis done consists of- Using various libraries from viz. Pandas, NumPy, Seaborn, SciKit Learn etc., to summarize the data, pre-processing the data to ensure the suitability of the dataset for various Machine Learning Models, visualizing the distributions in the dataset using graphical and statistical methods. Towards the end, we train various models on the dataset and assess their performance using multiple parameters such as accuracy, F1 score etc. to figure out the best model which could be used for the purpose of prediction.

We now go over each of the points in detail, starting off with the data pre-processing.

2. Data Pre-processing

2.1 Missing Values

All the feature columns in the dataset given consisted of significant proportion of missing Values as can be observed from the figure below. We use different techniques to either omit rows containing large number of missing values or fill in the missing values (imputation) with some parameter depending upon nature of the feature (numerical or categorical)



	column_name	% Missing
10	label	4.87875
0	expense	2.55500
6	score2	2.54500
4	age	2.51750
2	loan type	2.51375
9	score5	2.49750
8	score4	2.46500
1	income	2.44375
7	score3	2.44375
5	score1	2.42500
3	occupation type	2.32375

Fig. 1&2: *Heatmap showing missing values and the table with percentage of missing values in each column*

Based on the type of data present, we have broadly classified our features as follows:

- Numerical features: `['expense', 'income', 'age', 'score1', 'score2', 'score3', 'score4', 'score5']`
- Categorical features: `['loan type', 'occupation type', 'label']`

Before we proceed towards the method of imputation, here are some potential issues with the same:

- Values filled in by the method of imputation may not resonate well with the actual data and hence add noise or unnecessary variation affecting the model's performance.
- Missing data can be useful in some cases. In the field of banking, missing information from loan details can be an indicative of a customer with high risk of defaulting the loan. Thus, keeping the missing values intact can sometimes be beneficial.

The methods used for imputation are:

- **Median** of the column- For all the numerical features in our dataset, we fill in the missing places with the median value of the respective column. We did not use mean, since it is sensitive to outliers.
- **Mode** of the column- For all the categorical features in our dataset, we fill in the missing places with the mode value of the respective column.
- **Removal** of rows with more than 3 missing values- We check for the number of rows with more than 3 missing values. Since the method of imputation may not work best for more than 3 missing values, we remove (a total of 113) such rows.
- **Regression** for the 'expense' column- We found that the features 'expense' and 'score5' were perfectly correlated. So, to fill in the missing values of expense, we have used linear regression to predict the missing values of expense, with score5 as the independent variable and expense as the target/dependent variable.

2.2 Outliers

Outliers are unusual values in a dataset, that usually fall long way apart from the other observations. The boxplots shown in fig. 2 for all the features are a clear indicative of the outliers present in almost all the features of our dataset. The corresponding distributions indicate a near normal distribution of all these features with no skewness observed.

Now, it is almost always advisable to remove the outliers present in the dataset in order to ensure effectiveness of our model. The **IQR method** for removing outliers was used. However, after removing the outliers, a dip in accuracy and F1 score was observed on the training as well as the test data. This shows that the outliers also contained some important data points useful for a more accurate prediction. Hence, it was finally decided to keep the outliers intact. Another aspect of keeping the outliers relates to the context of this project. An outlier present in the details of the customer such as expenses or income maybe a sign of a fraudulent customer which has high chances of defaulting the loans. Thus, it is indeed useful to consider such observation for the modelling purpose.

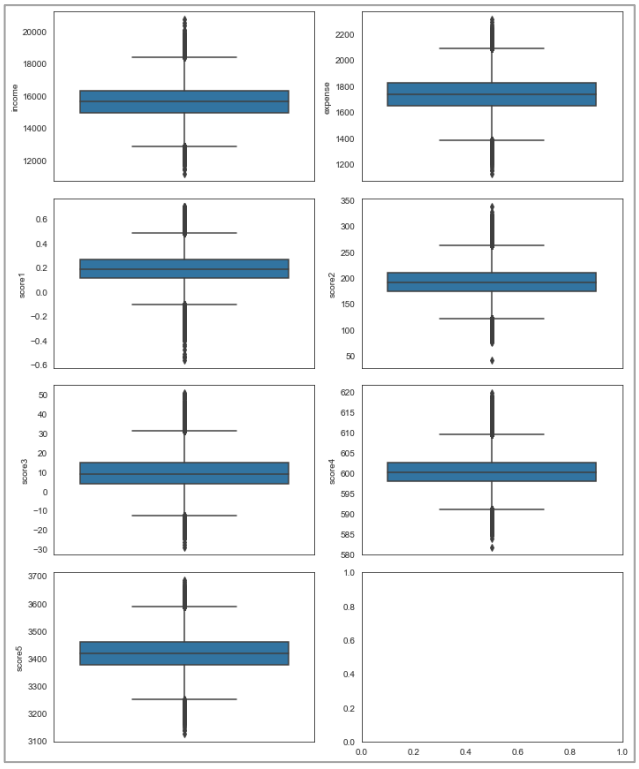


Fig. 3: *Boxplots for all the numerical features*

3. Exploratory Data Analysis (EDA)

In the Exploratory Data Analysis, we use data visualization techniques in order to gain some more clear insights and observations about the distribution of the features in our dataset. The relationship between our features helped us understand the important features to use for the final modelling and some more interesting observations to help improve prediction accuracy.

3.1 General Observations

Some of the interesting observations from the different plots (available to view in the Notebook) are mentioned below-

- More number of customers in our dataset have taken the loan type 'A' which also has relatively lesser default and relatively greater non-default loans than the loan type ‘B’.
- The Customers aged above 50 seem to prefer loan 'B' while for loan 'A', we have a greater number of the customers aged below 50.
- For both the types of loans, the average annual income of the customers with default loan is higher than the ones with non-default loans.
- For both the types of loans, the average expenses of the customers with default loan are greater than the ones with non-default loans.
- The customers with loan type ‘B’ in general are seen to have higher annual income as well expenses than those with the loan type ‘A’.

- The customers from occupation type 'X' have the highest level of average annual income as well as the expenses, followed by the occupation type ‘Y’.
- The customers with occupation type 'X' have the highest average value for all score1 to score5.
- The average score1, score2, score3 values are higher for customers with Loan type 'B' than those with the Loan type 'A'.

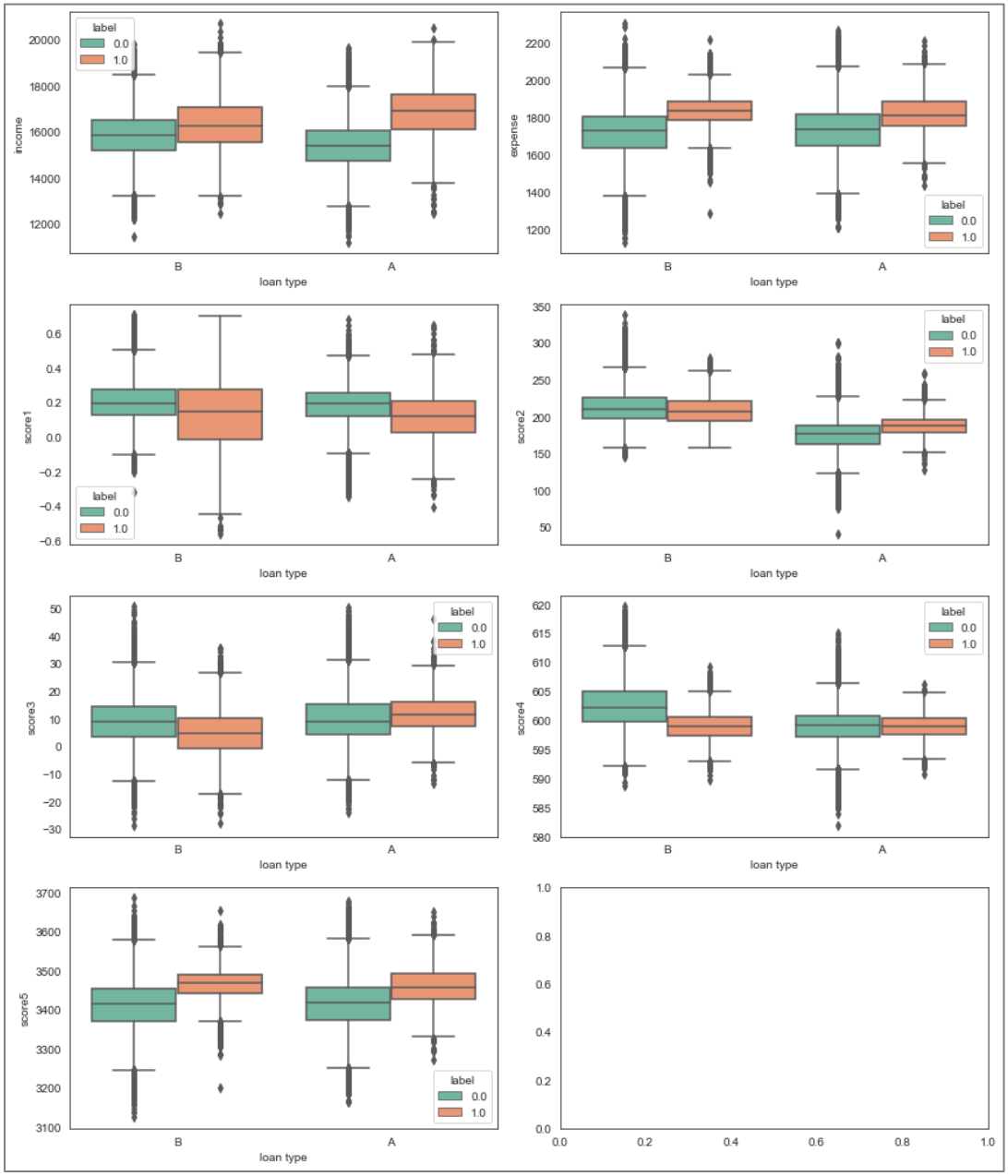


Fig. 4: *Boxplot showing variation between the categorical variable loan type (X-axis) and numerical continuous variables on the Y-axis*

3.2 Correlation

The correlation present between the variables can be detrimental to our model's performance. We find the correlation using Pandas `df.corr()` method and plot using Seaborn.

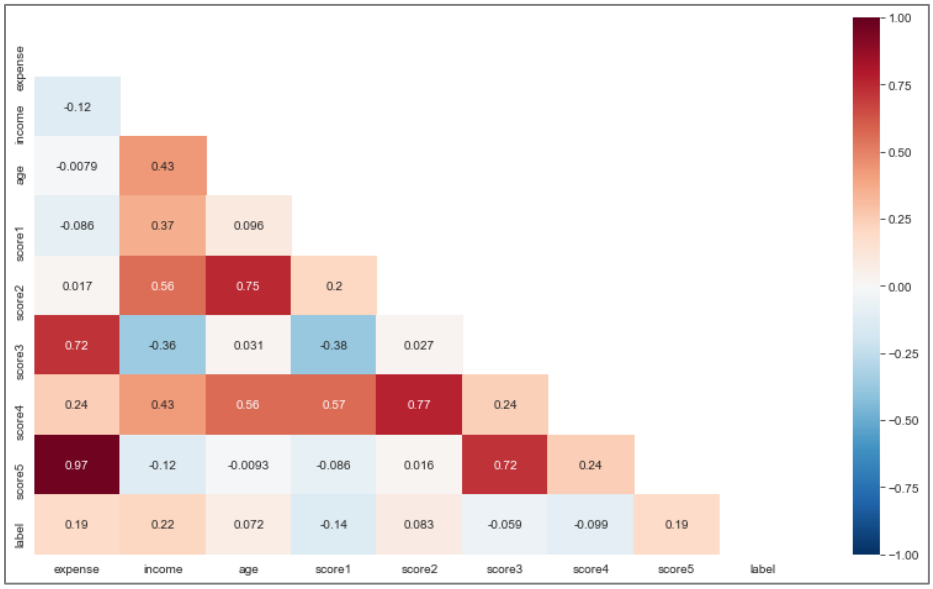


Fig. 5: *Correlation Heatmap*

As we can observe from the above heatmap, Dark Red and Blue hues indicate high (+ve and -ve respectively) correlation. There are many highly correlated variables as we had expected. For example, the feature 'score4' has a high positive correlation with several other features. The feature 'score5' has correlation value of nearly 1 with the feature 'expense'. This is in line with our previous conclusion of these features being perfectly correlated.

Based on several iterations of training models with different combinations of variables, we decided whether to omit or include the set of variables.

3.3 Variance Inflation Factor (VIF)

	feature	VIF
0	ID	4.001074
1	Expense	6994.088532
2	Income	617.157903
3	Loan type	4.894687
4	Occupation type	13.937272
5	Age	4.985370
6	Score1	10.170125
7	Score2	252.107978
8	Score3	18.106185
9	Score4	65304.971992
10	Score5	105579.478252

We have calculated the Variance Inflation Factor (VIF) for all features in the dataset, in order to see the relationship between each feature and the other features, and also as a check for multicollinearity.

Based on our observations, we have removed the 'score4' feature since it had a very high VIF value, and was also found to be highly correlated with other features. The same goes with the feature 'score5' as well.

Fig. 6: *Variance Inflation Factors for all features*

3.4 Dummy Coding

While using Machine Learning Classification models, it is required to convert all the object data types to either floats or integers. Since two of our features viz. 'loan type' and 'occupation type' had string values, we use the method of 'Dummy Coding' to convert these columns into ones with numerical float type data. Dummy coding scheme is similar to one-hot encoding. This categorical data encoding method transforms the categorical variable into a set of binary variables (also known as dummy variables. The dummy encoding is a small improvement over one-hot-encoding. Dummy encoding uses N-1 features to represent N labels/categories. We use the Pandas `pd.get_dummies()` method to do the same.

3.5 Upsampling

This technique is used to add more entries of the minority class in a dataset having imbalanced class labels. In our project we have used the resample function (imported from the sklearn.utils) to implement this technique. The default strategy is to implement the first step of the bootstrapping procedure where more entries are created by sampling from the minority class labels itself (with replacement). Using which we created a dataset containing equal proportion of class labels as it adds robustness to a model and prevents it from overfitting.

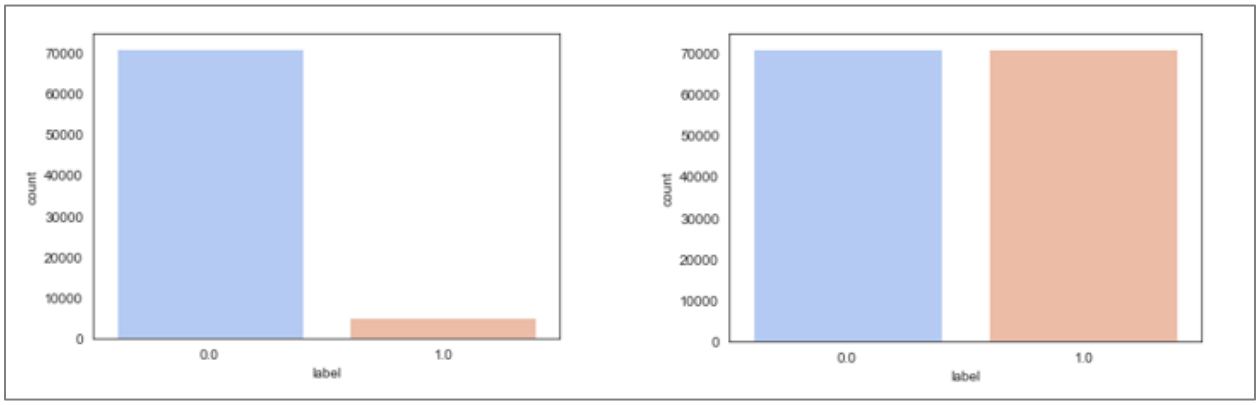


Fig. 7: *The distribution of the label feature before and after upsampling*

3.6 Standardization

The features in the dataset given are of different scales. The problem with features of different scales is that, some machine learning algorithms (Ex: KNN, K-means etc;) give higher importance to features with large magnitude, thus making those features play a more decisive role, even though they are not very significant. Hence, there is a need to make features scales similar.

Standardization refers to scaling the distribution of each feature to have a mean of zero and a standard deviation of one. It is most useful when the features have a gaussian distribution. Since all the features are approximately normal and have different scales, we have standardized all features so that each feature has zero mean and unit variance.

4 Model Building

In this stage, we tried to fit various models to the data and accessed the fit using both accuracy and f1 score to have a better understanding of the fit. As the labels in the dataset were not equally distributed, we **upsampled the minority class** so that the model does not result in overfitting (resulted in better results for most of the models). KNN, Random Forest, Decision Tree and XGBoost classifiers were found to be giving consistent results on the data. On taking the results of upsampled data into consideration, the Random Forest Classifier turned out to be the best model giving the best results in all the evaluation metrics of accuracy, f1 score and ROC-AUC score. Therefore, we decided to go ahead with the **Random Forest Classifier**.

A brief description of these models is:

Model	Description
Logistic Regression Model	A statistical method of analysis used to model the probability of a certain class or event when the given dependent variable is dichotomous. It uses a basic logistic function to model a binary dependent variable to make predictions. It can also be seen as one of the foundational tools for making decisions. Despite the huge applicability of this model, it has its own assumptions. Like there should be no, or very little multicollinearity between the predicted variables, the independent variables should be linearly related to the log odds and the sample size needs to be fairly large. While it is quite easier to implement and works well for the data that is linearly separable, it fails to predict a continuous outcome. One another drawback is that it assumes linearity between the dependent and independent variable which limits its applicability.
K-Nearest Neighbours	One of the most basic and essential algorithms used in machine learning for predictive and classification analysis. It belongs to the supervised learning domain and finds intense application in the field of pattern recognition and data mining. k-NN is a type of learning where the function is only approximated locally and all computation is deferred until function evaluation. As this algorithm takes into account the distance for classification, it can improve the accuracy to a much larger extent training data is normalized.

Naïve Bayes	A machine learning model that is used to discriminate different objects based on certain features. It is based on bayes theorem and assumes that the predictor variables used contribute independently to the probability. And that's why it is known as 'naïve' classification. The assumption where holds, helps it perform better compared to the models like logistic regression even with the lesser training data. There are three type of naïve bayes; Gaussian, Multinomial and Bernoulli. And there uses can be extended for various applications like real-time prediction, multi-class prediction, spam filtering and recommendation systems.
Random Forest	A supervised learning tool which is used both for classification and regression. A random forest consists of various decision tress on randomly selected data samples, gets prediction from these trees and votes out the best solution. Here, each individual tree is generated using an attribute selection indictor then each tree votes and the most popular is class is chosen as the final result. Alternatively, the weight concept can also be applied to a random forest to alter the impact of each individual tree. Those with low error rate are given high weight value and vice versa. This would increase the decision impact of trees with low error rate. Despite this method being highly accurate and robust, the features need to have good predictive power and the predictions need to be very less correlated to perform well.
Decision Trees	A method commonly used in data mining. It is a mathematical technique use to categorize and generalize a given set of data. The decision tree acquires data in form of a tree. Its main advantage is the ability to use different decision rules at different stages of classification. Decision trees have various advantages over other data mining techniques. It is quite simple to understand and interpret, can handle both numerical and categorical data. It can easily work on large data sets and requires very little data preparation. But on the other hand, these are very non-robust. Meaning that the final predictions can vary significantly even if little changes are made in the training sets.
XGBoost Classifier	One of the most popular machine learning classifiers these days. It works perfectly for both classification as well as regression. It is an ensemble learning method where bagging and boosting are the two most widely used ensemble learners. Both the techniques can be used with various statistical models but the most common is using these with the decision trees. XGBoost was developed to increase computational speed and optimize model performance and it very well works the same. XGBoost was written in C++ and has built in parameters for regularization and cross validation to keep the bias and variance at it minimum. These built in parameters are what gives it the advantage and leads to faster implementation.

Model	Normal (Imbalanced) Data		Upsampled Data		
	Accuracy (%)	F1 Score	Accuracy (%)	F1 Score	ROC_AUC Score
Logistic Regression	95.69	94.97	83.63	83.35	0.908
KNN (k=15) *	98.08	97.97	96.20	96.30	0.994
Naïve Bayes	95.25	94.30	81.26	81.23	0.898
Random Forest Classifier	98.23	98.14	99.71	99.71	0.999
Decision Tree Classifier	97.08	97.05	99.17	99.17	0.992
XGBoost Classifier	98.24	98.15	97.99	98.00	0.997

*Value of k was selected using cross validation
--> Detailed results for each model are shown in the Jupyter notebook

Table 1: Summary of models trained on training data.

5 Final Conclusion & Recommendations

Some of the **Recommendations** to avoid loans from going default:

a) Rolling out loan type 'A' for customers with higher annual income seems risky.
b) Focus more on customers aged above 50 which have better financial stability.
c) Ensure to carefully handle customers with higher annual income as well as occupation type 'X' since they have greater risk of defaulting the loan.
d) Customers belonging to occupation type Z and having high expenses, are more likely to default on loan. So, its advisable to focus on these people.
e) Focus more on customers whose occupation type is X and have low 'score4' value.
f) Having lesser amount of missing values would have helped for a better analysis.

6 Important Links

Here are the links to the Notebook, Dataset and Report uploaded to a GitHub Repository. Please do check them out!

- GitHub Repository Main Page Link
- Final Project Jupyter Notebook Link
