CAPSTONE PROJECT

E-Signing a loan based on Financial History

Abstract

When an applicant applies for a loan, lending companies evaluate them by analysing the financial history of their applicant. Most of the time parameters which are assessed by the lending company prior to calculating e-signing a loan involves complex analysing techniques.

A machine learning model is developed to predict whether an applicant's likelihood of e-signing a loan based on the financial data acquired through intermediary marketplace.

M M Ranush Wickramarathne

mmoveen@gmail.com | ranushw@uom.lk

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

Lending companies/ banks work by analyzing the financial history of their loan applicants and choosing whether the applicant is too risk or not prior to the loan approval. If the applicant is satisfactory on the financial status, company/ bank determines terms of the loan.

In order to acquire these potential applicants, companies can identify them through their websites often tracking them with the advertisement campaigns. Other than that, lending companies/ banks partner up with peer-to-peer lending marketplaces. Those marketplaces provide financial details and calculated details of the applicant.

This project targets on predicting potential customers who are getting their electronic signature (e-sign) phase passed considering the leads that intermediate marketplaces provide. The customers who predicted to be passing e-signing can be advertised with relevant information and approached easily.

DATASET FOR THE MODEL

Because the applicants arrived through a intermediate marketplace, we have access to applicants financial data before the onboarding process for lending company begins. This data includes personal information like age, and time employed, as well as other financial metrics. Dataset includes columns where it utilizes these financial data points to create risk scores based on many different risk factors.

Also, the dataset consists of scores from algorithms, built by the finance and engineering in marketplace where data are taken. Furthermore, the marketplace itself provides with their own lead quality scores. Model will leverage both sets of scores, as well as small list of personal/financial features to predict if the user is likely to respond to our current onboarding process.

The dataset is available on Kaggle with example notebooks shared by users. [1]

Following table represents the column names of the dataset which is taken to develop the model and a description of each column.

Table 1 - Dataset Column Description

Column Name	Description
Entry_id	Unique ID provided by the marketplace
Age	Age of the applicant
Pay_schedule	The frequency of the applicant getting paid (weekly, bi-weekly, monthly, semi-monthly)
Home_owner	Whether the applicant own a house or not
Income	Income of the applicant
Years_employed	Years the applicant has been employed
Months_employed	Number of Months the applicant has been employed (represents the months which hasn't completed a full year)
Current_address_year	Number of years spent on the current address that applicant has provided

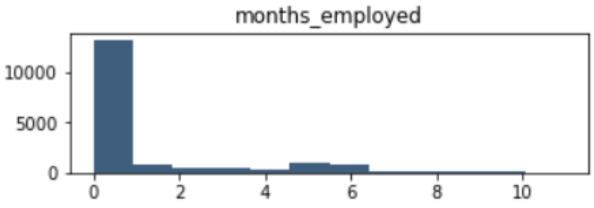
Column Name	Description
personal_account_m	Number of months (which isn't completing a full year) spent after creating the current bank account
personal_account_y	Number of years spent after creating the current bank account
has_debt	Whether the applicant has existing debt with financial institutes
amount_requested	Amount the user has applied as for the loan
risk_score (1-5)	Calculated risk score
ext_quality_score (1,2)	Calculated quality score
inquiries_last_month	Inquiries with the marketplace in the last month
E_signed	Whether the applicant was e-signed and able to onboard on next processes

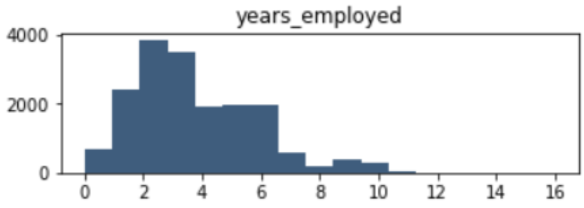
METHODOLOGY

The following section describes about the development process of the machine learning model. The model development is done on **Jupyter kernel running Python 3.8.**

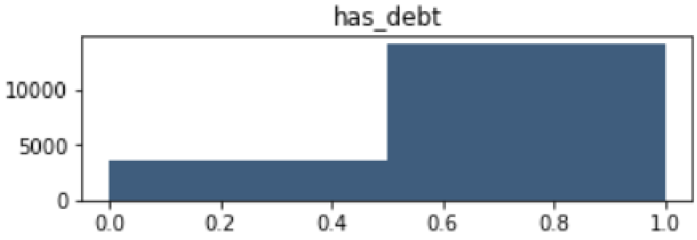
Exploratory Data Analysis

Dataset is checked for 'Null' values in the beginning. Since the dataset is completed without null or missing values a histogram plot is used to determine the distribution of data in the dataset. Following histogram plots are taken from the Jupyter notebook



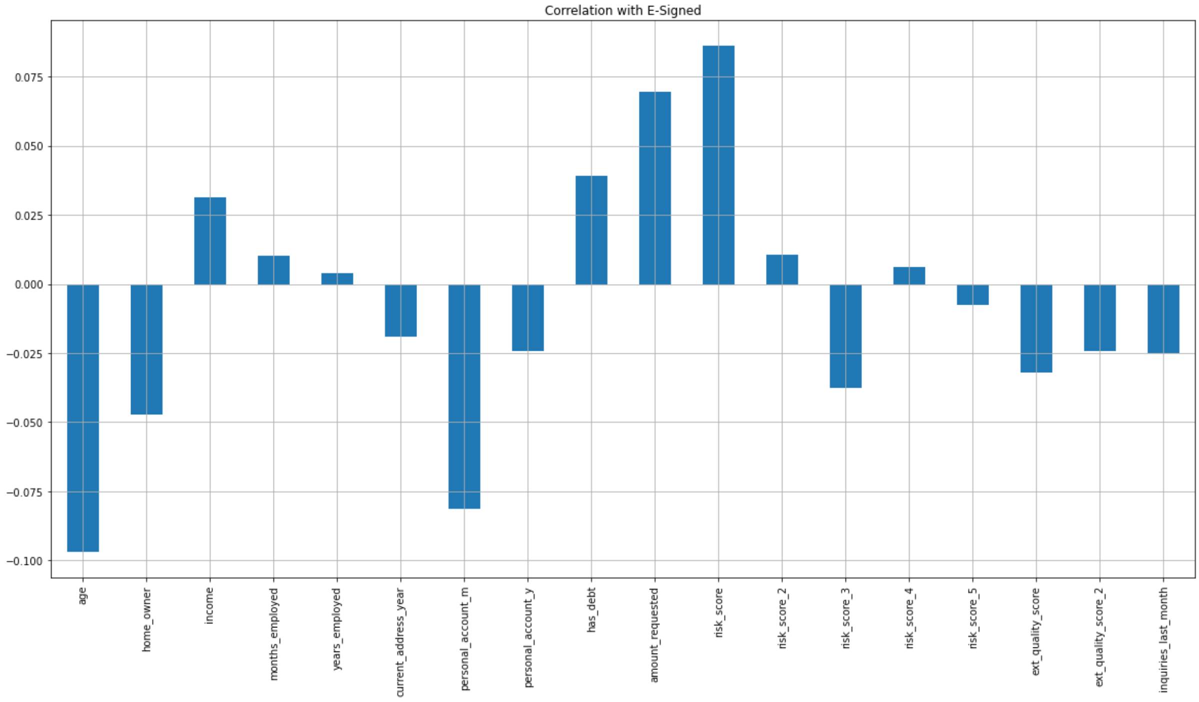


As the above histogram plot clearly showing that 'months_employed' column is mostly including values which is equal to '1', above column tends towards least important feature for the outcome.



The above histogram plot represents the column name of 'has_debt' which is having a high unbalance in the sample dataset. Other than those two columns most of the features has regular data distribution across each value it represents.

In order to understand more about the feature importance, a correlation diagram with dependent variable (e_signed) was plotted and illustrated in the following figure.



As the above correlation plot represents, it can be seen that which features of the dataset plays a vital role in determining the dependent variable. Column *age, personal account existence in months, amount requested* and *risk score 1* are having either positive or negative correlation with the dependent variable.

Feature Engineering

As the above plots illustrates, *personal account* m (months) can be joined together with *personal account y* (years) since having months calculated from the beginning it could also be a important feature for predictions. Also, the *months employed* can be dropped from the table as it doesn't play a huge role and contains same value on the dataset.

One Hot Encoding

One hot encoding is done for the only categorical variable available on the dataset which is "*pay_schedule". Get* dummies method has been used to categorized four categories.

After the one hot encoding has been done, the multi-collinearity issue was arising. As the author of [2] suggests that multi-collinearity issue (dummy variable trap) can be eliminated with removing one of the categorical column created by the dummy variable function. In order to determine which column to remove, the column with least number of occurrences has been chosen since it will be allowing weights not to be biased. Column with "*pay_schedule_monthly"* is removed after analyzing.

Model Building

The above problem is to define one binary output, which is a perfect example for a classification type model. The models for the classification were initially built with the basic regression models and then moved towards Ensemble models (e.g., Random Forest).

A model with **Support Vector machine** was also developed and the training time for the model with **linear kernel** was higher and interrupted in the process by user. (Notebook cell 37)

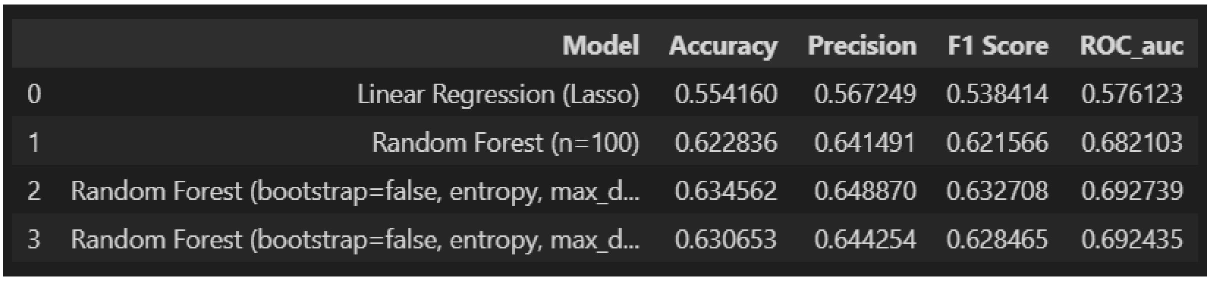
Random Forest model was initially built with basic hyper-parameters (n_estimators = 100) and afterwards it was associated with **Grid Search** method considering the ***criterion*** *as 'entropy' and 'gini'*. The hyper-parameters elected for the two grid searches are included in the following table.

Table 2 - Grid Search Hyper-Parameters

Grid Search Criterion	Hyper-Parameters
Entropy	'bootstrap': [True, False], 'criterion': ['entropy'], 'max_depth': [3, None], 'min_samples_split': [2, 5, 10], 'n_estimators': [100, 500]
Gini	'bootstrap': [True, False], 'criterion': ['gini'], 'max_depth': [3, None], 'min_samples_split': [2, 5, 10], 'n_estimators': [400, 600]}, pre_dispatch=2, verbose=3)

RESULTS

The results for the above classification-based model is evaluated with **accuracy, precision, F1 score** and **ROC_auc**. The results from all four models built for the solution is shown in the following figure which is exported from Jupyter notebook.



	Model	Accuracy	Precision	F1 Score	ROC_auc
0	Linear Regression (Lasso)	0.554160	0.567249	0.538414	0.576123
1	Random Forest (n=100)	0.622836	0.641491	0.621566	0.682103
2	Random Forest (bootstrap=false, entropy, max_d...	0.634562	0.648870	0.632708	0.692739
3	Random Forest (bootstrap=false, entropy, max_d...	0.630653	0.644254	0.628465	0.692435

As the above figure shows that the best accuracy and precision can be seen with the INDEX = 2 model. Then the specific model was saved for future evaluations.

Predicting on Sample Data

After the model is imported to the Notebook, relevant functions for data pre-processing and adding encoded data to the data-frame was developed. These functions are useful when the deployment of the model needs to be done via a webpage developed for easy and convenient interaction.

CONCLUSION

Since the above classification has been done with simple regression-based models and moved towards Random Forest based models, it can be clearly seen that the accuracy has been increased. Even though the accuracy improvement is shown with model being complexed, the accuracy is less than 65% which makes the model developed for this purpose still providing false predictions on input data.

DISCUSSION

The main issue of the models developed above is the less accuracy of the predictions. This could be solved by backward elimination method by dropping more least important features and re-training the best model.

Also, the accuracy of the above models can still be improved with moving towards Deep Neural Network (DNN). The hidden layers and other input and output parameters could lead towards a better predicting model for e-signing loan based on financial aspects of applicants.

REFERENCES

[1] "Kaggle," [Online]. Available: https://www.kaggle.com/aniruddhachoudhury/esigning-of-loan-based-on-financial-history/version/1.

[2] K. K. Mahto, "One-Hot-Encoding, Multicollinearity and the Dummy Variable Trap," July 2019. [Online]. Available: https://towardsdatascience.com/one-hot-encoding-multicollinearity-and-the-dummy-variable-trap-b5840be3c41a.