Machine Learning Engineer Nanodegree

Capstone Project

Shubhendra Kumar
December 15, 2018

I. Definition

Project Overview

The project background is based on the analysis performed on potential customers or candidates in financial industry especially by investing firms and banks. Since banks are those institutions which generate most of their revenue by giving out loans hence it is very important to analyze potential customers which can contribute to generating targeted revenue in the banking industry. Since the fundamental background of customers differ from each other, hence it becomes of utmost importance to leverage machine learning to analyze financial history and background of the customers to generate meaningful and predictive insights from their data.

Using machine learning, we can create Models that could predict or classify whether the customer will be interested in utilizing services and products offered by banks and hence efforts could be directed to engaging those customers and on the other hand predicted, less interested customers can be targeted with much better schemes or service rates that they couldn't refuse to the offers made to them.

This article https://www.stoodnt.com/blog/scopes-of-big-data-data-science-in-the-banking-finance-fintech-sector/ illustrates the impact of data science in the functionality of financial services and institutions such as corporate banks.

Also this research paper https://www.researchgate.net/publication/327011881_Predicting_Credit_Worthiness_of_Bank_Customer_with_Machine_Learning_Over_Cloud on predicting credit worthiness of a bank customer user data science and machine learning to generate insights and profits for a bank. While lenders have been relying on analytics to provide automated loans, they had only the customer's history with the bank to fall back on.

Financial history of the customer such as age, income, months employed, years employed etc. will be used to build up a classification model, that could learn from the provided data and make predictions thereby classifying those customers that are at a high chance of

signing an E-Loan when offered by the financial institution/bank. The dataset has been obtained/downloaded from Super Data Science Machine Learning Practical (https://www.superdatascience.com/machine-learning-practical/) Module 5. The dataset has been included as "financial_data.csv".

Problem Statement

The problem statement is to classify customers of a financial institution so as to find that they would be interested in signing E-loan services offered or not by a financial bank or a firm, keeping in mind the financial history of customers. To process each of the customer's financial background manually, would not simply suffice, hence this problem requires a supervised machine learning model that can learn from the existing data and make predictions for the customers. The financial parameters such as their age, years of employment, debt etc. could be utilized as input features for the learning model. A classification supervised learning model shall be developed to classify customers to segregate them into 2 categories: those who will sign the E-loan and those who wouldn't.

The final model shall be evaluated on accuracy and f1_score majorly as evaluating metrics and will then be ready for operating upon new incoming customer's data.

Metrics

The model shall be evaluated on the basis of accuracy metric where

- Accuracy = $\frac{\text{(true positives + true negatives)}}{\text{(true positives + true negatives + false positives +false negatives).}}$

More the evaluated accuracy, more is the success of the classification model which means that the model more accurately classifies the customers into the categories of signing or not signing the E-loan. The classification report for the model shall also generate the evaluation of precision and recall scores too. Also we shall evaluate f1 score because f1-score mathematically represents the balance between precision and recall scores as

- F1-score = $\frac{\text{(2 * precision * recall)}}{\text{(precision + recall)}}$

i.e. the harmonic mean of precision and recall scores. Precision of the model as well as the recall score of the model should be well maintained, hence we choose 'f1_score' as the evaluation metric after 'accuracy_score' metric. Since f1_score is the harmonic mean of precision and recall scores, therefore f1_score holds the final importance after accuracy score.

High Precision score: The number of customers that did not sign the E-loan according to model, but in real, did sign the E-loan, are very less.

High Recall score: The number of customers that did sign the E-loan according to model, but in real, did not sign the E-loan, are very less.

II. Analysis

Data Exploration

The dataset has been obtained/downloaded from Super Data Science Machine Learning Practical (https://www.superdatascience.com/machine-learning-practical/) Module 5. The dataset is a real time dataset but the identity of the users has been anonymized. The dataset contains the financial history of the customers that will be used to build up a classification model to find whether the user will be interested in signing an E-Loan or not when offered by a financial institution.

The parameters of the datasets are:

1.) **entry_id** = User's entry id/ unique identification
2.) **age** = user's/customer's age
3.) **pay_schedule** = how often the applicant gets payed
4.) **home_owner** = owns a home or not
5.) **income** = monthly income
6.) **years_employed** = years passed since users started to do job
7.) **months_employed** = months employed after the previous job year completion
8.) **current_address_year** = years living in the same house till today
9.) **personal_account_m** = months for which user has personal account after the latest year completion
10.) **personal_account_y** = years for which user has had account
11.) **has_debt** = has pending debt
12.) **amount_requested** = amount requested by user from the financial institution
13.) **risk_score, risk_score_2, risk_score_3, risk_score_4, risk_score_5** = risk score attached with the customer which signifies the risk percentage that the user shall be able to return money or not within the time allotted.
14.) **ext_quality_score, ext_quality_score_2** = external quality score of a customer
15.) **inquiries_last_month** = inquiries made in the last month by the user
16.) **e_signed** = signed an e-loan when offered by the financial institution.

These shall be the input parameters for the classification model.
The dataset contains a total of 17908 rows.
The no of customers that didn't sign the E-loan = 8270 and those who signed = 9640.
Therefore the dataset is pretty much balanced and this is also beneficial for our model as it contains sufficient classes to learn from where in this case it will be a classification model.

DATA SAMPLE :

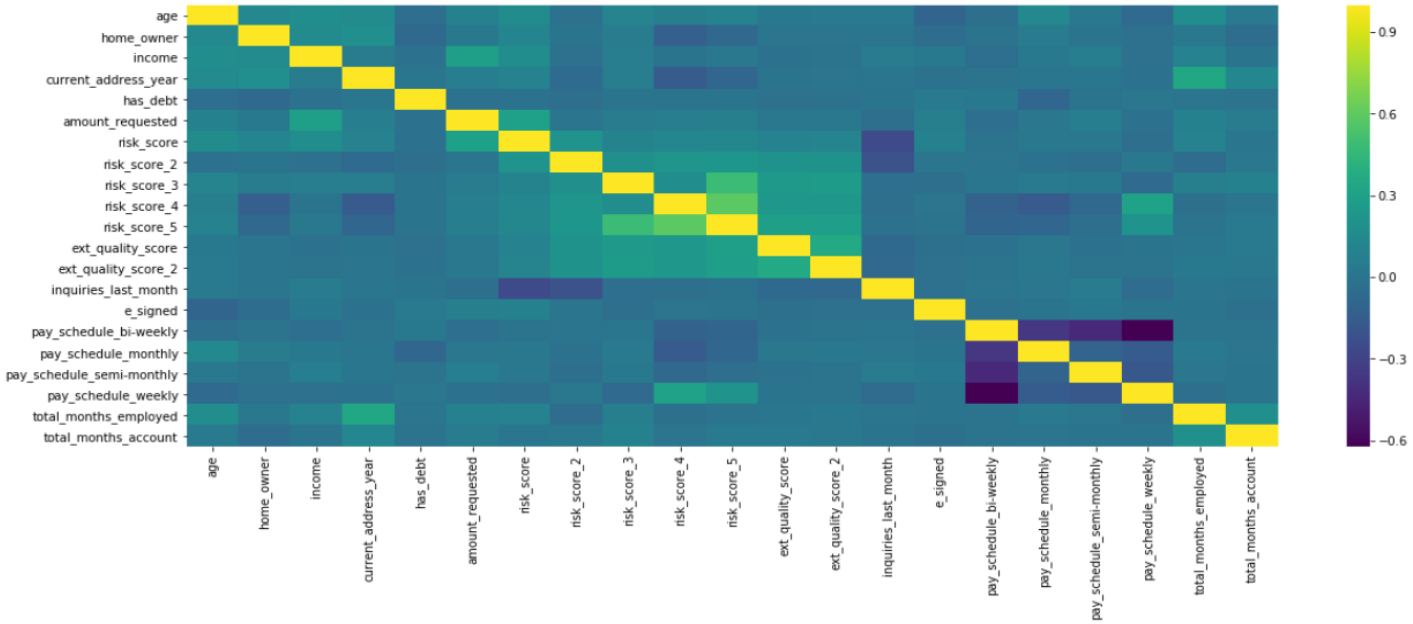
A	B	C	D	E	F	G	H	I
entry_id	age	pay_schedule	home_owner	income	months_employed	years_employed	current_address_year	personal_account_m
7629673	40	bi-weekly	1	3135	0	3	3	6
3560428	61	weekly	0	3180	0	6	3	2
6934997	23	weekly	0	1540	6	0	0	7
5682812	40	bi-weekly	0	5230	0	6	1	2
5335819	33	semi-monthly	0	3590	0	5	2	2
8492423	21	weekly	0	2303	0	5	8	2
7948313	26	bi-weekly	0	2795	0	4	4	1
4297036	43	bi-weekly	0	5000	0	2	1	1
6493191	32	semi-monthly	0	5260	3	0	3	1
8908605	51	bi-weekly	1	3055	0	6	11	4

J	K	L	M	N	O	P
personal_account_y	has_debt	amount_requested	risk_score	risk_score_2	risk_score_3	risk_score_4
2	1	550	36200	0.737398319	0.903517238	0.4877125
7	1	600	30150	0.738510084	0.881026665	0.713423437
1	1	450	34550	0.642993277	0.76655369	0.595017969
7	1	700	42150	0.665223529	0.960832286	0.767828125
8	1	1100	53850	0.617361345	0.857559706	0.613486719
7	1	600	74850	0.677109244	0.758765039	0.495609375
6	1	800	50800	0.738054622	0.873204345	0.666436719
2	1	1100	69100	0.798303361	0.841746723	0.401971094
4	1	1150	64050	0.652428571	0.802433112	0.593815625
2	1	600	59750	0.624665546	0.968564823	0.50991875
3	1	400	61700	0.659736134	0.937286766	0.852323438

Q	R	S	T	U
risk_score_5	ext_quality_score	ext_quality_score_2	inquiries_last_month	e_signed
0.515976695	0.580918	0.380918	10	1
0.826401956	0.73072	0.63072	9	0
0.762283825	0.531712	0.531712	7	0
0.778830826	0.792552	0.592552	8	1
0.665523488	0.744634	0.744634	12	0
0.664761939	0.592556	0.492556	6	1
0.700392155	0.58413	0.68413	14	1
0.568787109	0.525905	0.725905	5	1
0.560388681	0.569459	0.369459	3	1
0.749624146	0.758607	0.758607	5	1
0.7856976	0.632466	0.732466	7	0

Distribution plot for each of the relevant input features have been calculated and the comparisons have been made. Again a correlation heatplot has been generated. This heatplot displays the correlation factor (pearson's correlation) of each relevant feature with the other possible feature combination. Correlation intensity determines the depth at which a feature is dependent on other feature i.e a feature shows increase or decrease in magnitude on increasing or decreasing the other input feature. Correlation varies between -1 and +1. For example with increase in income feature, loan amount requested feature also shows an increase, i.e positive correlation.

CORRELATION PLOT :



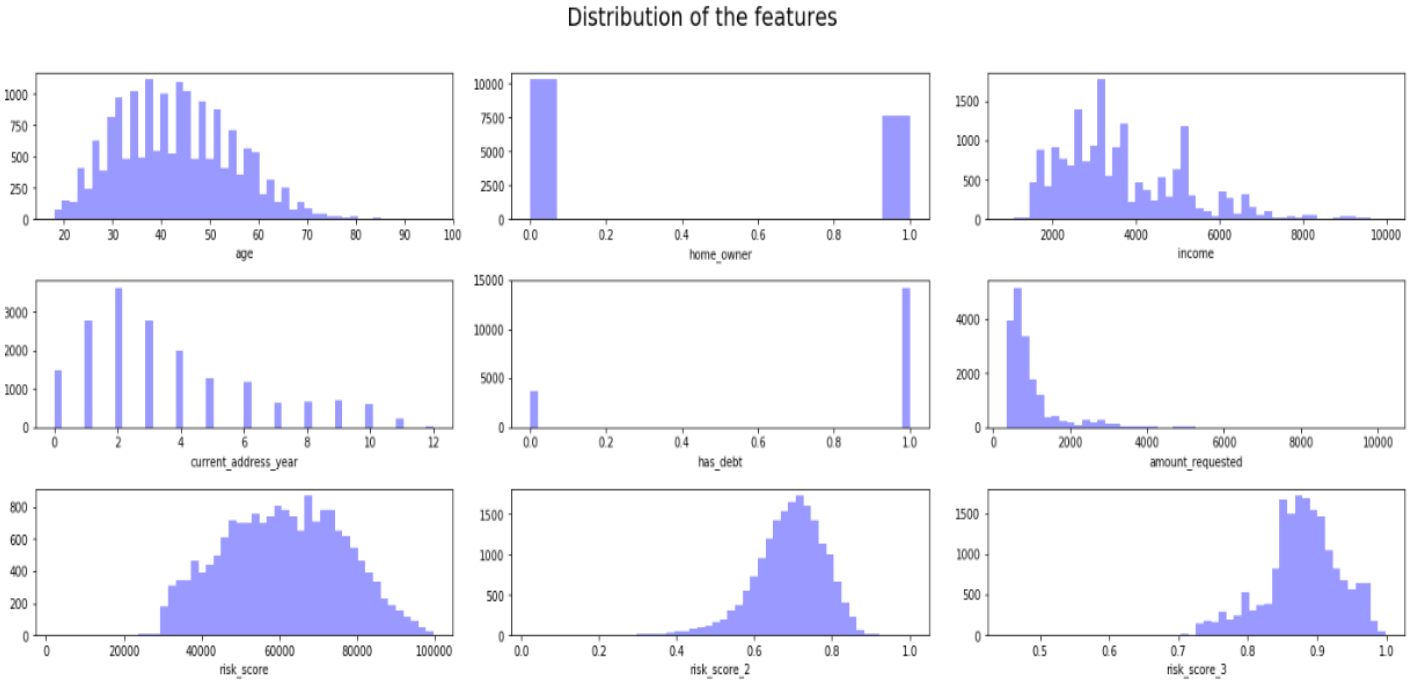
The dataset has been described using python's pandas library which displays statistical values for the input features.

DATASET DESCRIPTION :

	age	home_owner	income	current_address_year	has_debt	amount_requested
count	17908.000000	17908.000000	17908.000000	17908.000000	17908.000000	17908.000000
mean	43.015412	0.425173	3657.214653	3.584711	0.795399	950.446449
std	11.873107	0.494383	1504.890063	2.751937	0.403421	698.543683
min	18.000000	0.000000	905.000000	0.000000	0.000000	350.000000
25%	34.000000	0.000000	2580.000000	2.000000	1.000000	600.000000
50%	42.000000	0.000000	3260.000000	3.000000	1.000000	700.000000
75%	51.000000	1.000000	4670.000000	5.000000	1.000000	1100.000000
max	96.000000	1.000000	9985.000000	12.000000	1.000000	10200.000000

Exploratory Visualization

The data distribution of the input features have been provided. This indicates the probability distribution of the input features data.

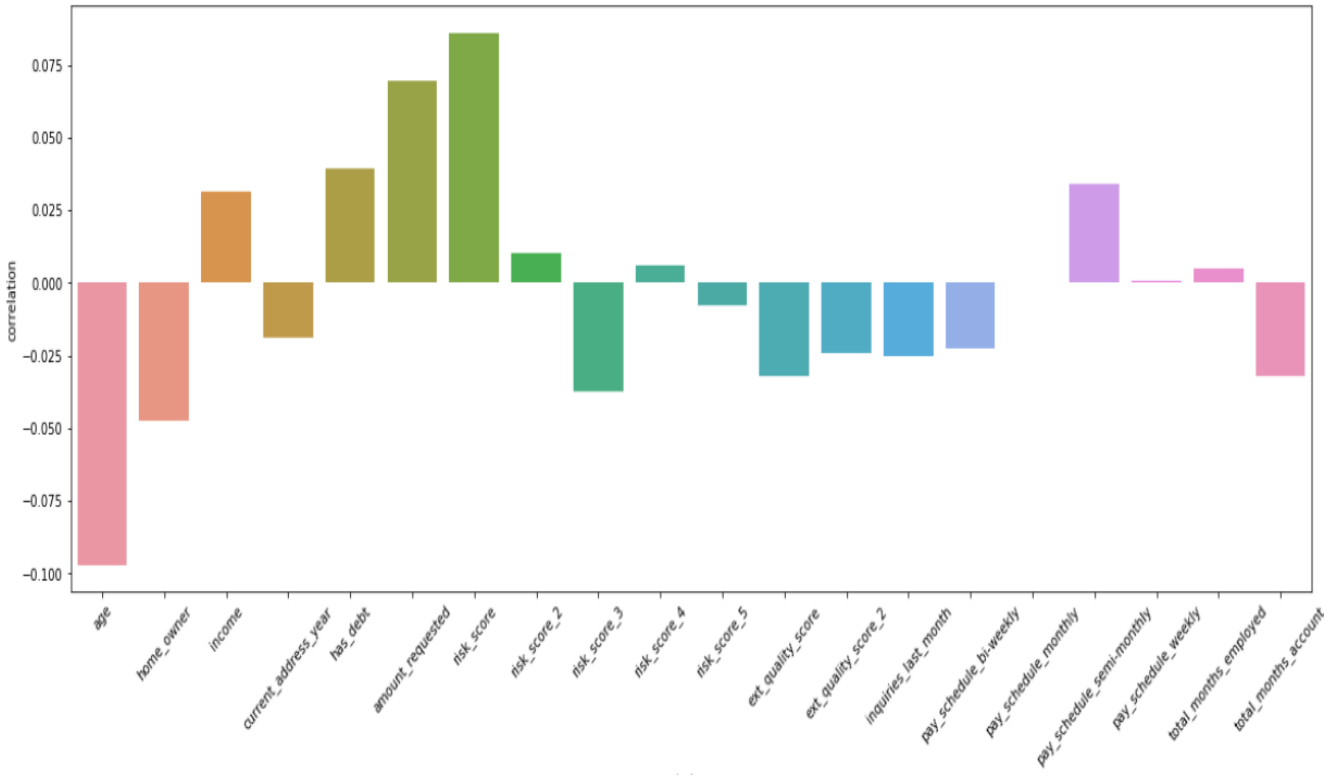


The data distribution for the entire input features have been plotted using histogram plot in python. This reveals information such as :

- The amount requested ranges between being greater than 200 dollars and shows a high distribution maximum around 1000 dollars and few over 2000 dollars with very less distribution near 4000 dollars.
- Age distribution is maximum in the range 20 ad 50 at the time when a person may require facilities for education loan, home loan, car loan etc.

Correlation between 'E-signed' feature and other dependent input features reveals :

- More the 'amount requested', more is the positive correlation with the signing of the E-loan.
- 'Age' is strongly and negatively correlated with the E-signing of the loan.
- 'Risk score' is strongly and correlated positively with the E signing of the loan.



Algorithms and Techniques

At most two classification algorithms: Support Vector Machines and Logistic Regression shall be selected and fed to the input features and the model shall be trained with the training data. The model that gives the best metrics (accuracy) on the test data, shall be selected and will be hyper-tuned to enhance its performance. At first Logistic Regression is used as a benchmark model and then for actually developing the solution, Support Vector Machine classifier algorithm is used.

1.) Logistic Regression :

Logistic regression uses an equation as the representation, very much like linear regression. Input values (x) are combined linearly using weights or coefficient values (referred to as the

Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value. Logistic regression models the probability of the default class.

Logistic Regression model is fitted with the training data and the made to perform predictions on the test dataset. Also the training time and prediction time has been noted and observed so as to compare it with the respective metrics of the SVM model.

2.) Support Vector Machine Classifier :

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

SVM model is also fitted with the training data and allowed to predict on the test data. The training time and the prediction time are also measured and compared with the benchmark model i.e. Logistic Regression model. Since the SVM model offers much dynamic hyper-tuning of the hyper-parameters, hence this will be our core prime solution classifier model.

The initial parameters of the SVM model are :

- C=1, gamma='auto_deprecated', kernel='rbf', degree=3

- **C :** C affects the trade-off between complexity and proportion of nonseparable samples and must be selected by the user." or more specifically the trade-off between errors on training data set and margin maximization. A smaller C will allow more errors and margin errors and usually produce a larger margin. When C goes to Infinity, svm becomes a hard-margin. Note that the value of C is chosen by users based on training experiments.

- **gamma :** Intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

- **Kernel :** kernel parameters defines whether the available data can be classified linearly or not.

Benchmark

Logistic Regression model has been used as a benchmark model.

Logistic Regression model provides :

- Accuracy_score = 57.069 %
- Precision_score = 58.018 %
- F1-score = 64.328 %
- Recall_score = 72.17 %

The above metrics were obtained after the logistic regression model was fitted with the training set and made to predict on the test dataset. The above result was obtained with the training data that constitutes 75% of the total dataset.

We observe that the SVM model provides with the more improved metrics of the same type in the very first attempt signifying the scope of improvement of the hyper-parameters tuning to enhance the model performance.

III. Methodology

Data Preprocessing

Since the dataset contains no null values or the values entirely of different datatypes, than as expected from the input feature, so the dataset is concluded to be clean.

The dataset has been divided into train and test set where train set constitutes 75% of the total dataset and test dataset represents 25% of the total dataset.

For the implementation of the SVM model, feature scaling is required hence Standard Scaling has been used for performing feature scaling on the training and the test dataset. The idea behind standard scaling is that it will transform your data such that its distribution will have a mean value 0 and standard deviation of 1. Given the distribution of

the data, each value in the dataset will have the sample mean value subtracted, and then divided by the standard deviation of the whole dataset.

Standard scaling :

$$z = \frac{(\text{observation value} - \text{mean})}{(\text{standard deviation})}$$

This completes the data preprocessing steps and no further preprocessing is required for model development.

Implementation

The implementation is done in two stages :

- Model Training stage
- Model Testing stage

Steps :

1. Split the dataset into 75% training and 25% test dataset.
2. Performing feature scaling using Standard scaling
3. Creating a model instance
4. Fitting the classifier model with the training dataset
5. Observing the time taken to train the dataset
6. Noting the predictions on the test set by the classifier
7. Prepare a classification report highlighting the accuracy, precision and recall score along with the F1-score.
8. Developing a visual representation of the confusion matrix for observing the classifier performance class wise.

The process was simple and straightforward with no complexities involving except the fact that the SVM model took thrice as more time to train than as compared to the logistic regression model.

Refinement

An initial solution has been found using the SVM model classifier with the following used hyper parameters value :

- C=1, gamma='auto_deprecated', kernel='rbf', degree=3

The final classification report for the SVM classifier is as follows :

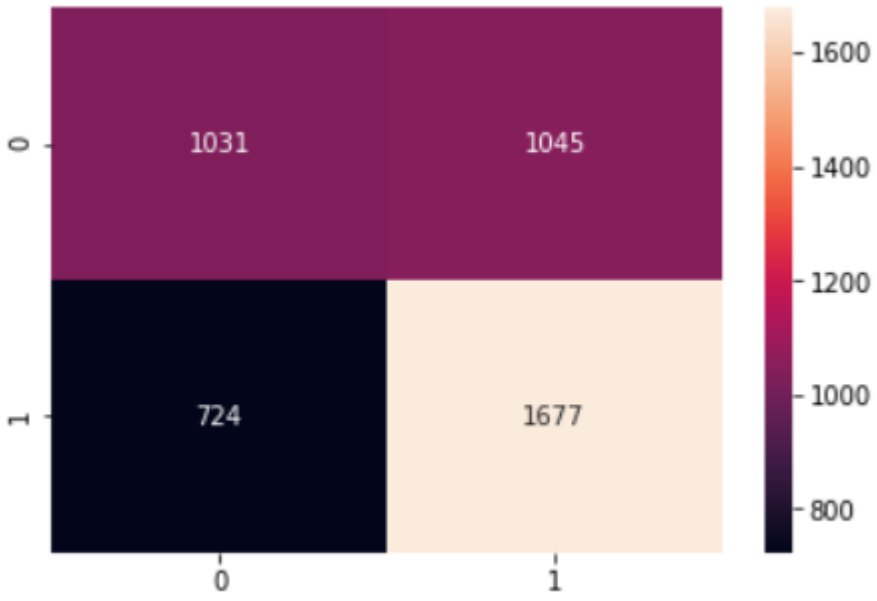
	precision	recall	f1-score	support
0	0.59	0.50	0.54	2076
1	0.62	0.70	0.65	2401
micro avg	0.60	0.60	0.60	4477
macro avg	0.60	0.60	0.60	4477
weighted avg	0.60	0.60	0.60	4477

ACCURACY METRICS FOR THE SVM AND THE SVM TUNED CLASSIFIER :

- Svm model = Accuracy : 60.48 %
- Svm tuned model = Accuracy : 60.60 %

The results of the evaluated model has been obtained using classification report and the confusion matrix.

CONFUSION MATRIX :



The model has been improved using Grid search cv and k-fold cross validation where times of validation stages = 10.

Grid Search CV :
Parameter grid => 'C' = [10,100], 'gamma' = [0.01, 0.001], 'kernel' = ['linear', 'rbf']

The possible number of combination that can be made for the hyper tuned model :

2*2*2= 8 combinations. Hence 8 combination are tried for each validation stage where every combination is tested and the accuracy which is the scoring metric, is computed and the mean accuracy noted. After all the combinations have been tried, then the combination that resulted in the best scoring parameter (accuracy), is selected as the best parameter for the hyper-tuned model.

We found that the combination :

- {'C': 10, 'gamma': 'auto deprecated', 'kernel': 'rbf'}

is the best parameter which when used to develop the SVM model, results an accuracy of 60.58 % than as compared to the initial default SVM model prototype that gave an accuracy of 60.48 %.

IV. Results

Model Evaluation and Validation

- The svm model at first is evaluated using the k-fold cross validation and later on is tuned using 'grid search cv'. The accuracy obtained at each of the cross validation stages is noted which is found to have very less standard deviation and good mean accuracy, hence showing that the model has a good bias-variance tradeoff, neither over fitting nor under fitting.
- The model has been tested with several inputs and the model performs optimally on each input test features.
- [0.59598214, 0.60193452, 0.61830357, 0.59717051, 0.62844378,
  0.60312733, 0.60089352, 0.60685034, 0.61997019, 0.585693].
  These are the accuracies of the cross validation stage at number of validation stages set to 10. The above accuracy array has a standard deviation of ~0.01.
- Since the model has been cross validated and it performed with accuracies at each validating stage with least standard deviation, the model is robust and the results from the model can be trusted.

- The classification report for the final evaluated model :

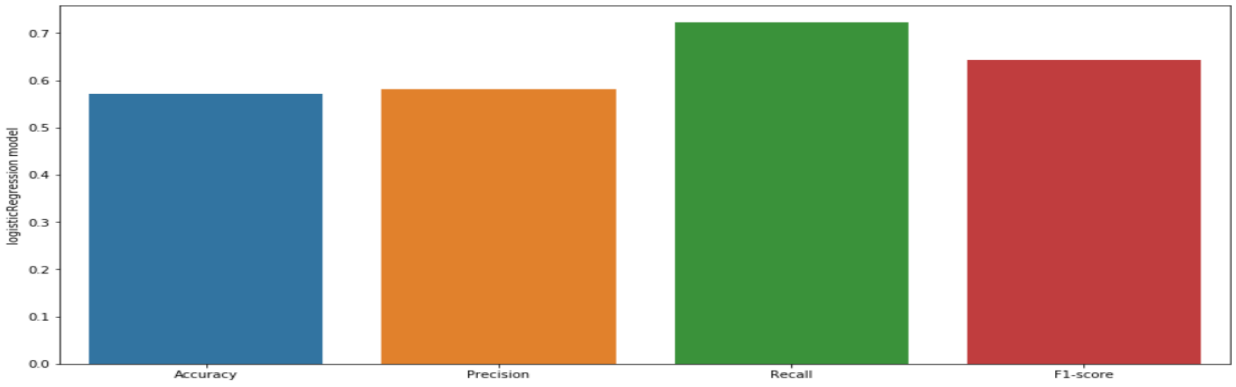
	precision	recall	f1-score	support
0	0.59	0.50	0.54	2076
1	0.62	0.70	0.65	2401
micro avg	0.60	0.60	0.60	4477
macro avg	0.60	0.60	0.60	4477
weighted avg	0.60	0.60	0.60	4477

- The final parameters of the evaluated model (hypertuned) :

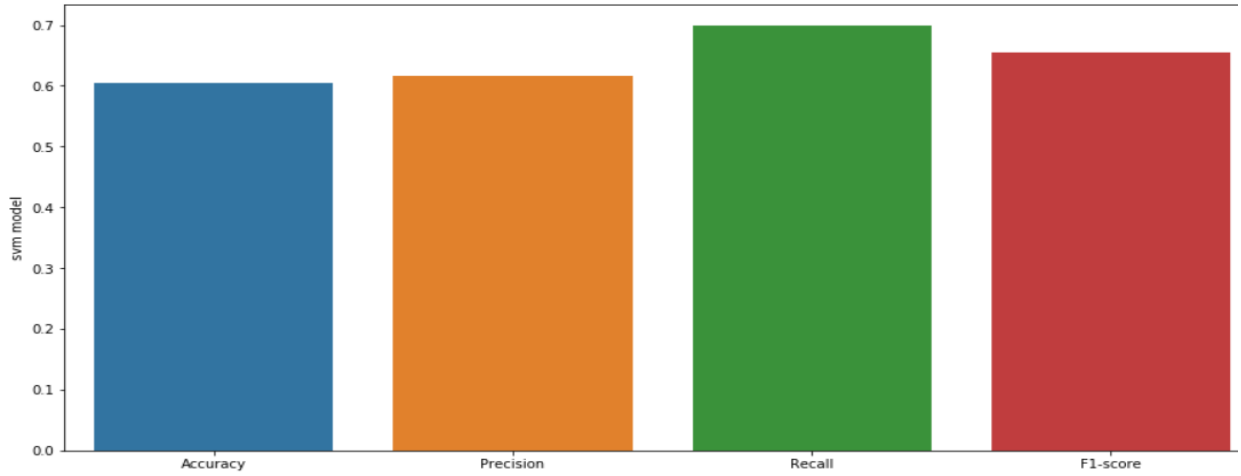
```
SVC(C=1,
    cache_size=200,
    class_weight=None,
    coef0=0.0,
    decision_function_shape='ovr',
    degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1,
    probability=False,
    random_state=None,
    shrinking=True,
    tol=0.001,
    verbose=False)
```

Justification

Results of the logistic regression model :



Results of the final svm model :

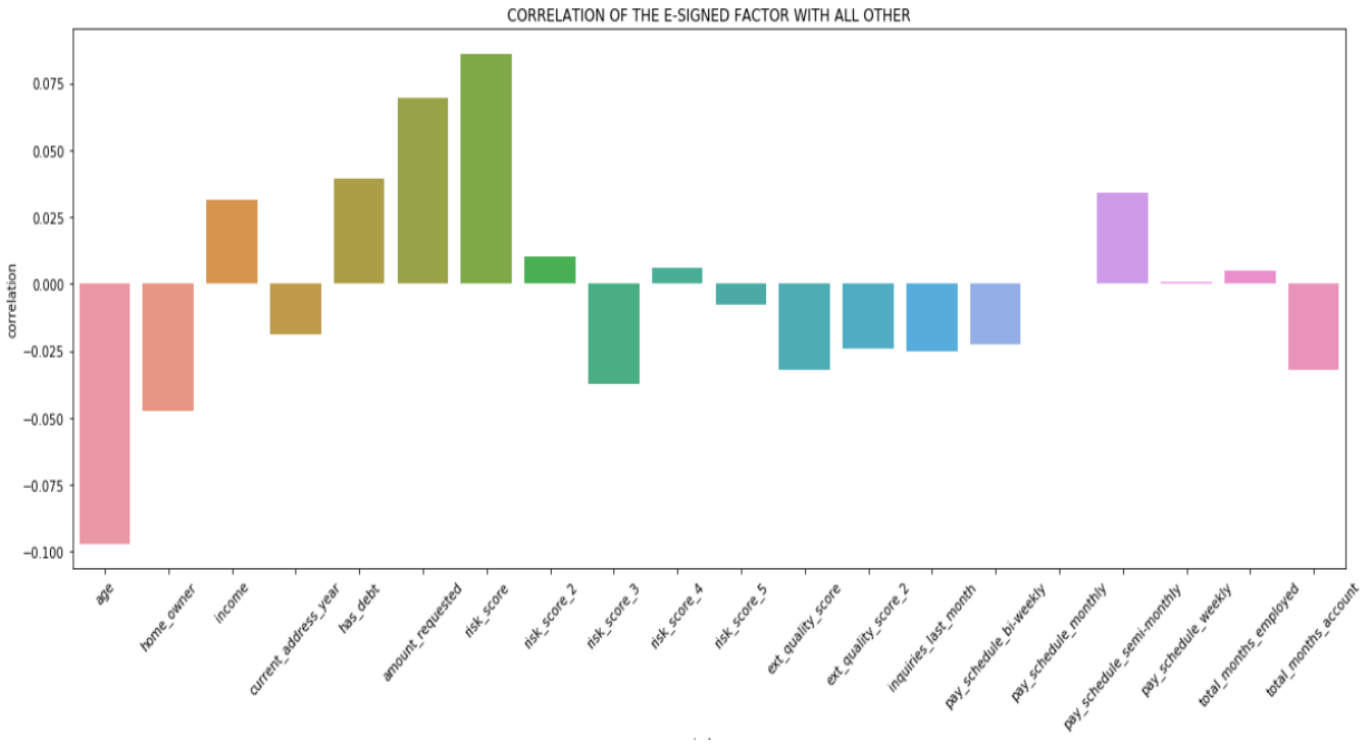


The results of the svm model is far better than the logistic regression model which was the benchmark model. The benchmark model outperforms the svm model only for the case of Recall score which implies the number of people to whom the model predicts 1 (Will sign E-loan) but actually didn't sign the loan are very less in the logistic regression model i.e. the number of false negatives are less as predicted in case of the benchmark model. Else for precision, f1_score and accuracy, the svm tuned model outperforms the logistic regression.

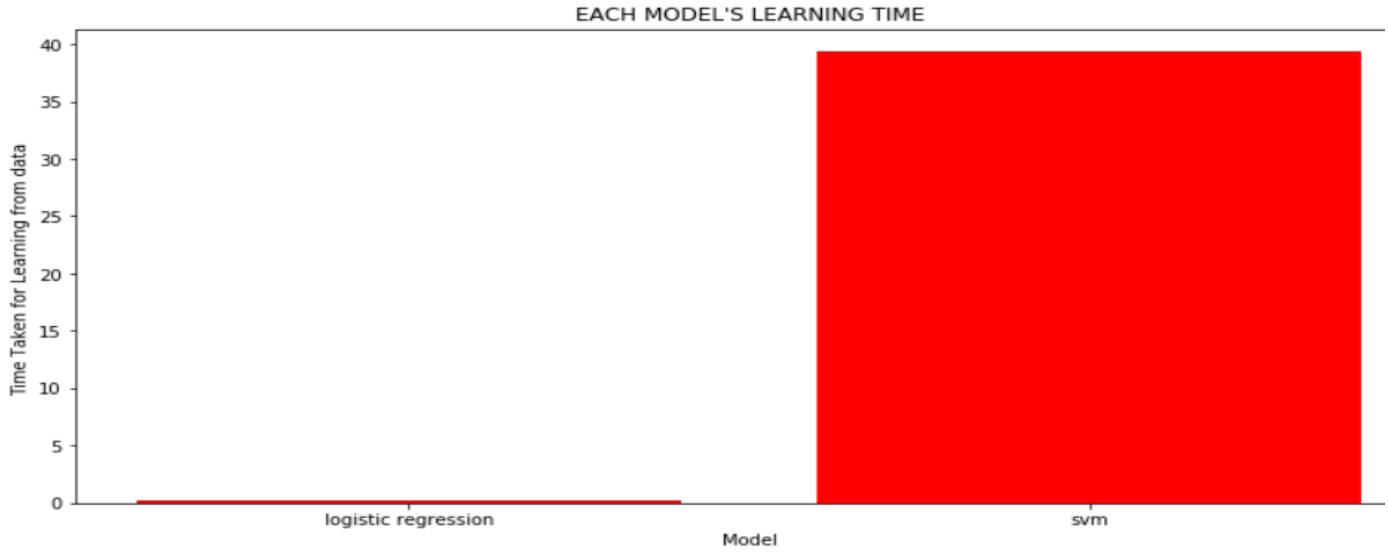
The final solution which is the hyper-parameter tuned svm model is significant enough to solve the problem of predicting the customer's E-signing a loan based on his financial history.

V. Conclusion

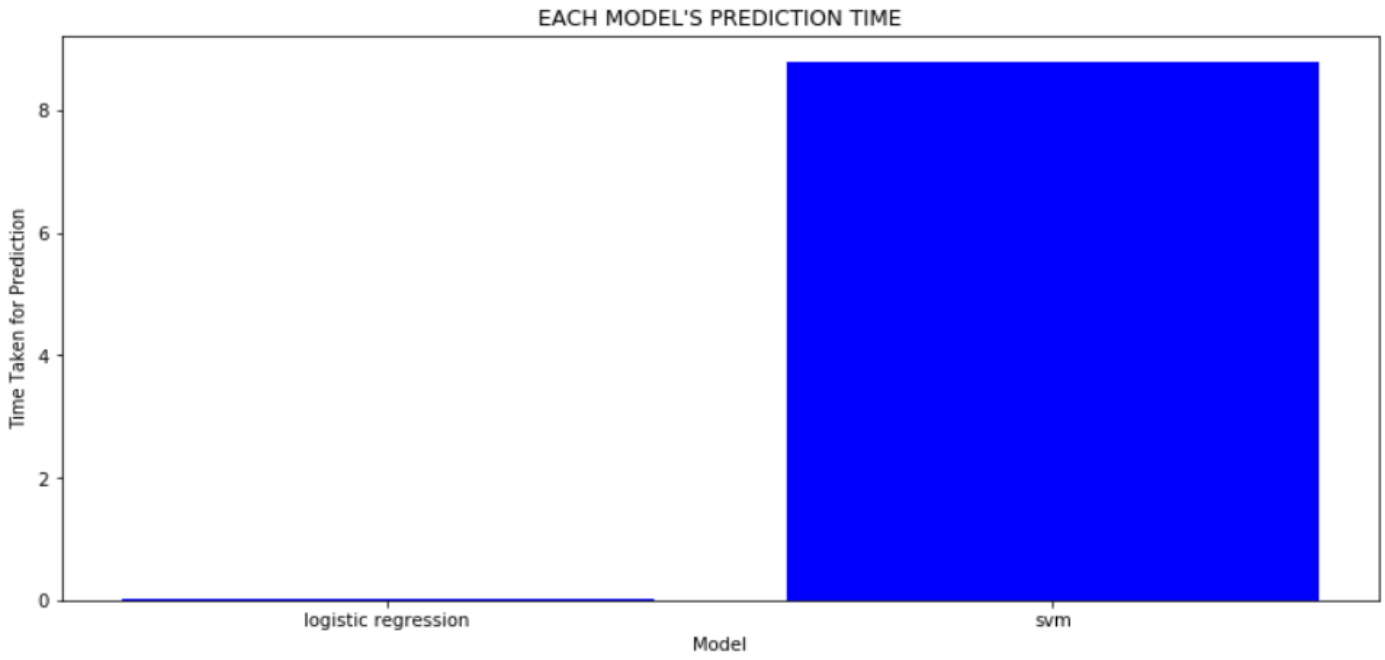
Free-Form Visualization



The above barplot displays the correlation of E-signed feature with the other input features. This is one of the most important plot that determines which input features are actually strongly correlated with the E signing factor for the loan. Factors like ‘pay_schedule’ monthly resembles no correlation at all with the E signing. Factors like ‘age’ and ‘amount_requested’, ‘has_debt’ are strongly correlated with the E-signing factor. Majority features are negatively correlated with the loan E-signing factor.



The above plot displays the time taken for learning from training data by both the models. The benchmark model takes much less time than as compared to the SVM model.



The above bar plot demonstrates that the SVM model takes much more time than the benchmark model to make predictions on the test data. Hence this reveals that svm models in general takes more time to train and predict that usual other Machine learning algorithms. Also, SVM tends to over fit a lot on smaller datasets.

Reflection

The project shall start by incorporating the dataset into the IPython notebook. At the first stage, the data shall be tried for cleaning where presence of any possible outliers or Null values shall be removed or rectified. Then Exploratory analysis shall be performed which would include several statistical visualization plots to understand the data and features more clearly. This phase shall be followed by selecting appropriate features for our models and then developing two of the popular classification models i.e. logistic regression and SVM for evaluation on the test set.

1. As per the process, we preprocessed the data after exploratory analysis and during data preprocessing, feature scaling was performed to normalize all the statistical observations in our dataset.

2. Two data models for classification were selected i.e. Logistic Regression and Support Vector Machine classifier. Logistic regression model is suited for binary classification hence used a benchmark model.
3. The data was split into training (75%) and testing (25%) dataset size. Both the models were fitted with the training dataset and the learning time for both the models was recorded.
4. The models were made to perform prediction on the test dataset with the prediction time being also recorded for each model.

The interesting part about developing models was the fact that although SVM model gave more accuracy and outperformed the benchmark model in all the scoring metrics, yet it was heavy on computation for learning and predicting for data. Hence when the dataset is heavy, also other algorithms should be tried that could reduce the computation time for learning and prediction. Also logistic regression trained and predicted itself efficiently thereby reducing down the number of heavy computation.

The difficult part was performing grid search using k-fold cross validation on the training dataset for the svm model. Since it had already been noticed of the much more time and computation invested for learning and predicting, therefore grid search for the svm model, increased the computation and it took a lot of time (more than 40 minutes) to complete the grid search for all the possible combinations of the parameter list provided.

The final solution is very much optimized and is a better option for performing on new data for deducing inferences and predictions. Since the size of the dataset was hugh in our case, hence svm, even though took more time to learn and train itself, gave good results on evaluating metrics than as compared to other benchmark models.

Improvement

- Final improvement could be made by using ensemble learning methods in machine learning by developing a model that is itself a combination of 2 or more machine learning algorithms.
- Using ensemble in random forest classifier might have yielded out a better model though with less training time and less prediction time as against SVM classifier model.