# Machine Learning Capstone Project

## Definition

### Project Overview

In recent years there has been a trend in financial institutions towards greater use of models in decision making, driven in part by government regulations but manifest in all areas of management. Nowadays a high proportion of bank decisions are automated through decision models. These decision models can be a combination of statistical or machine learning based models and a set of rules. With increased use algorithmic trading, an automated electronic platform that execute trade commands which have been pre-programmed by time, price or volume, and can start without manual intervention, the need to include model based decision into the automated trading is also increasing. Government regulations like Basel III also encourage banks to use decision models, which involves modeling risk in a trade and setting a threshold to decide whether the trade should be made or not.

Many investment banks trade loan backed securities. If a loan defaults, it leads to devaluation of the securitized product. That is why banks use risk models to identify loans at risk and predict loans that might default in near future. Risk models are also used to decide on approving a loan request by a borrower.

### Problem Statement

The goal is to find a model to predict loans that are likely to default and identify attributes that contribute to bad loans, so that it can be used to deny future loan applications. Loan data needs to be analyzed to see which attributes have high correlation with loan being default. If there are attributes, whose values definitely cause a loan to default then that can be used as a rule in the decision model. The attributes that do not definitely determine if a loan will default, can be used to build a machine learning model to predict likelihood of future loans being defaulted. This can then be used to assign a risk score to future loans and a threshold can be set, so loans exceeding a certain risk score can be automatically denied.

Since we are trying to predict if a loan will default or not as opposed to predicting the amount of profit or loss that will be incurred, this is a classification problem.

The loan data used here are for consumer loans. So most loans are expected to be short term loans and missing monthly installment has heavy penalty.

### Metrics

Looking at the dataset, we can see that out of 5000 loans there are 1533 loans that have defaulted and 3467 loans that were paid. With such an imbalanced dataset, accuracy score might not be a good choice. Accuracy score measures fraction of correctly identified labels. So accuracy score will be high if we detect loans that will default and loans that will not default. For credit risk it is more important to detect loans that will default than the ones that will not. F1 score is created out of a combination of precision and recall and it is a better measure of fraction of true positives or loans that will default. So we need to look at F1 Score as the metrics to measure model performance.

## Analysis

### Data Exploration

I will be using data from Lending Club (https://www.lendingclub.com/info/download-data.action). Lending Club is the world’s largest online marketplace connecting borrowers and investors.

The file LoansImputed.csv in project folder contains complete loan data for all loans issued through the time period stated.

#### Variables in Data Set

##### Dependent Variable

* **not.fully.paid**: A binary variable. 1 means borrower defaulted and 0 means monthly payments are made on time

##### Independent Variables

* **credit.policy**: 1 if borrower meets credit underwriting criteria and 0 otherwise
* **purpose**: The reason for the loan
* **int.rate**: Annual interest rate for the loan (14% is stored as 0.14)
* **installment**: Monthly payment to be made for the loan
* **log.annual.inc**: Natural log of self-reported annual income of the borrower
* **dti**: Debt to Income ratio of the borrower
* **fico**: FICO credit score of the borrower
* **days.with.cr.line**: Number of days’ borrower has had credit line
* **revol.bal**: The borrower's revolving balance (Principal loan amount still remaining)
* **revol.util**: Amount of credit line utilized by borrower as percentage of total available credit
* **inq.last.6mths**: Borrowers credit inquiry in last 6 months
* **delinq.2yrs**: Number of times borrower was delinquent in last 2 years
* **pub.rec**: Number of derogatory public record borrower has (Bankruptcy, tax liens and judgements etc.)

Below is summary of the dataset (Panda df.describe())

credit.policy int.rate installment log.annual.inc dti

count 5000.000000 5000.000000 5000.000000 5000.000000 5000.000000

mean 0.896200 0.120816 308.325968 10.911819 12.308698

std 0.305031 0.025336 197.307080 0.598897 6.754521

min 0.000000 0.060000 15.690000 7.600902 0.000000

25% 1.000000 0.100800 163.550000 10.545341 7.067500

50% 1.000000 0.121800 260.640000 10.915088 12.300000

75% 1.000000 0.137900 407.510000 11.277203 17.652500

max 1.000000 0.216400 926.830000 14.528354 29.960000

fico days.with.cr.line revol.bal revol.util

count 5000.000000 5000.000000 5000.000000 5000.000000

mean 710.926000 4510.713433 15872.533200 46.395622

std 37.026757 2418.553606 31116.319033 29.138604

min 617.000000 180.041667 0.000000 0.000000

25% 682.000000 2790.041667 3328.500000 22.300000

50% 707.000000 4080.000000 8605.000000 45.700000

75% 737.000000 5640.281250 18155.250000 70.500000

max 827.000000 16259.041670 1207359.000000 106.500000

inq.last.6mths delinq.2yrs pub.rec not.fully.paid annualincome

count 5000.0000 5000.00000 5000.000000 5000.000000 5000.00000

mean 1.4068 0.16140 0.066800 0.306600 66260.20820

std 1.9897 0.49699 0.257587 0.461128 56864.18592

min 0.0000 0.00000 0.000000 0.000000 2000.00000

25% 0.0000 0.00000 0.000000 0.000000 38000.00000

50% 1.0000 0.00000 0.000000 0.000000 55000.00000

75% 2.0000 0.00000 0.000000 1.000000 79000.00000

max 33.0000 6.00000 3.000000 1.000000 2039784.00000

Below are first five rows of the dataset (Panda df.head())

credit.policy purpose int.rate installment log.annual.inc

0 1 debt\_consolidation 0.1496 194.02 10.714418

1 1 all\_other 0.1114 131.22 11.002100

2 1 credit\_card 0.1343 678.08 11.884489

3 1 all\_other 0.1059 32.55 10.433822

4 1 small\_business 0.1501 225.37 12.269047

dti fico days.with.cr.line revol.bal revol.util inq.last.6mths

0 4.00 667 3180.041667 3839 76.8 0

1 11.08 722 5116.000000 24220 68.6 0

2 10.15 682 4209.958333 41674 74.1 0

3 14.47 687 1110.000000 4485 36.9 1

4 6.45 677 6240.000000 56411 75.3 0

delinq.2yrs pub.rec not.fully.paid annualincome

0 0 1 1 45000

1 0 0 1 60000

2 0 0 1 145000

3 0 0 1 33990

4 0 0 1 213000

Below is summary of the dataset that contains data for loans that defaulted

credit.policy int.rate installment log.annual.inc dti

count 1533.000000 1533.000000 1533.000000 1533.000000 1533.000000

mean 0.661448 0.132452 342.785114 10.885023 13.195838

std 0.473372 0.025495 223.948527 0.666718 7.006769

min 0.000000 0.070500 15.910000 7.600902 0.000000

25% 0.000000 0.115400 168.640000 10.491274 7.830000

50% 1.000000 0.131600 287.310000 10.878047 13.340000

75% 1.000000 0.148200 491.300000 11.276633 18.830000

max 1.000000 0.216400 926.830000 13.458836 29.960000

fico days.with.cr.line revol.bal revol.util

count 1533.000000 1533.000000 1533.000000 1533.000000

mean 697.828441 4393.541259 21066.293542 52.255075

std 33.756808 2431.785491 49905.689359 29.057906

min 617.000000 180.041667 0.000000 0.000000

25% 672.000000 2759.958333 3323.000000 29.900000

50% 692.000000 4050.000000 8850.000000 53.900000

75% 717.000000 5580.041667 20616.000000 77.000000

max 822.000000 15692.000000 1207359.000000 106.500000

inq.last.6mths delinq.2yrs pub.rec not.fully.paid annualincome

count 1533.000000 1533.000000 1533.000000 1533 1533.000000

mean 2.330724 0.174821 0.091324 1 67360.671885

std 2.933480 0.520562 0.292659 0 59224.859089

min 0.000000 0.000000 0.000000 1 2000.000000

25% 0.000000 0.000000 0.000000 1 36000.000000

50% 1.000000 0.000000 0.000000 1 53000.000000

75% 3.000000 0.000000 0.000000 1 78955.000000

max 33.000000 4.000000 2.000000 1 700000.000000

Below is summary of dataset that contains data for loans that did not default

credit.policy int.rate installment log.annual.inc dti

count 3467 3467.000000 3467.000000 3467.000000 3467.000000

mean 1 0.115671 293.089201 10.923667 11.916432

std 0 0.023498 182.272593 0.566024 6.603058

min 1 0.060000 15.690000 8.342840 0.000000

25% 1 0.096300 159.920000 10.585573 6.775000

50% 1 0.116600 249.680000 10.915088 11.860000

75% 1 0.131600 394.360000 11.277203 17.120000

max 1 0.208600 914.420000 14.528354 29.420000

fico days.with.cr.line revol.bal revol.util

count 3467.000000 3467.000000 3467.000000 3467.000000

mean 716.717335 4562.523339 13576.013268 43.804753

std 36.935882 2411.216297 16685.502884 28.800678

min 627.000000 1110.000000 0.000000 0.000000

25% 687.000000 2820.000000 3343.000000 18.950000

50% 712.000000 4109.041667 8507.000000 42.100000

75% 742.000000 5669.958333 17448.500000 66.950000

max 827.000000 16259.041670 149527.000000 99.700000

inq.last.6mths delinq.2yrs pub.rec not.fully.paid

count 3467.000000 3467.000000 3467.000000 3467

mean 0.998269 0.155466 0.055956 0

std 1.166961 0.486163 0.239701 0

min 0.000000 0.000000 0.000000 0

25% 0.000000 0.000000 0.000000 0

50% 1.000000 0.000000 0.000000 0

75% 2.000000 0.000000 0.000000 0

max 8.000000 6.000000 3.000000 0

annualincome

count 3467.000000

mean 65773.617248

std 55790.362442

min 4200.000000

25% 39560.000000

50% 55000.000000

75% 79000.000000

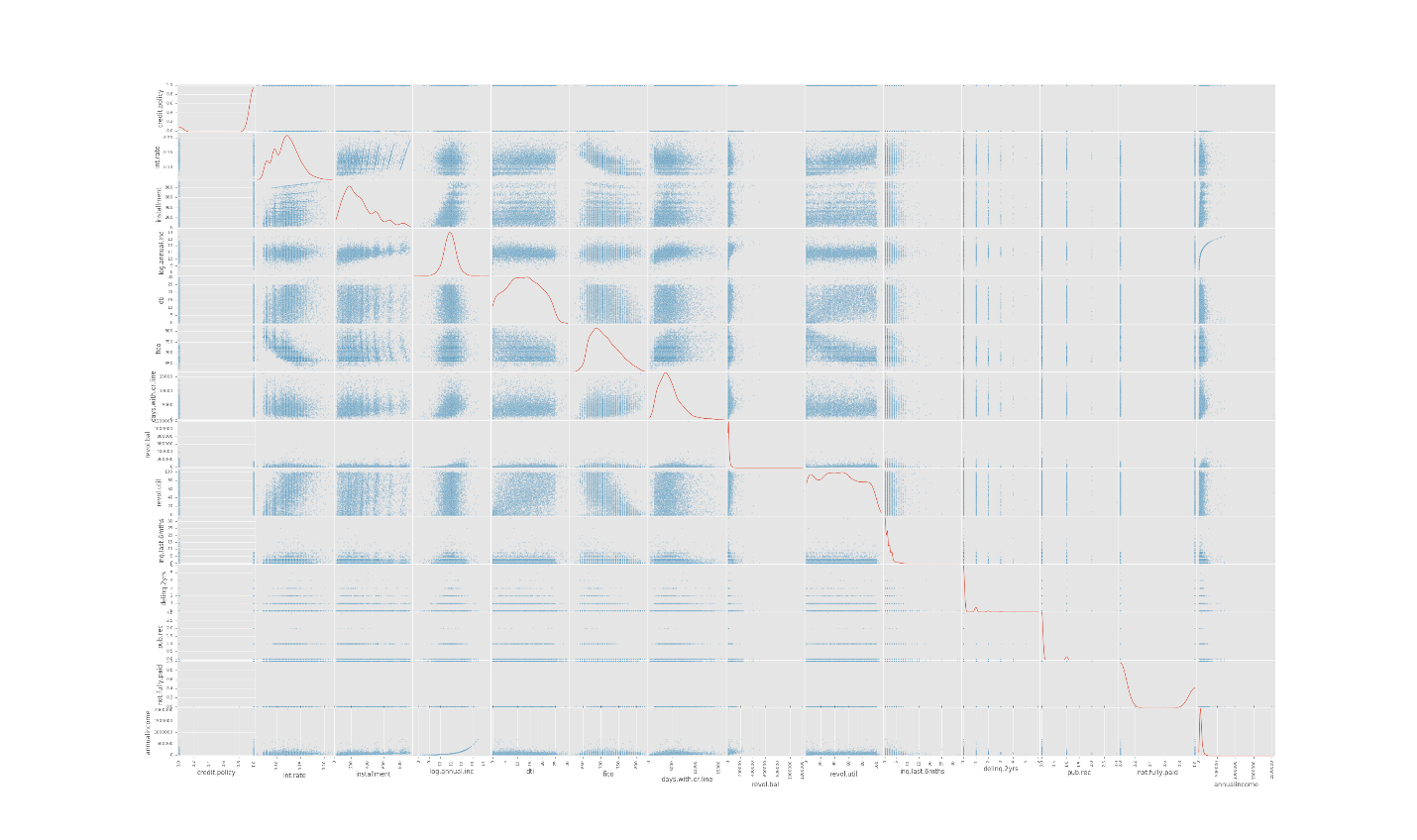
max 2039784.000000

The following observations can be made from looking at statistical summary of the data

* 100% of the accounts that are not default, had met credit underwriting criteria and out of the accounts that are defaulted 66% had met credit underwriting criteria. This means not meeting credit underwriting criteria should be used as primary reason to deny loan application.
* Mean interest rate of loans that defaulted is 13% and for loans that are current is 11%. That means loans that default tend to have higher interest rates.
* Mean installment of loans that defaulted is 342 and for current loans it is 293. This means loans that default tend to have higher installment payments.
* Mean debt to income ratio of defaulted loans is 13.2 and for current loans it is 11.9. This means loans that default tend to have higher debt to income ratio
* Mean revolving balance of default loans is 21066.3 and for current loans it is 13576. This means loans that default tend to have higher revolving balance.
* Mean credit utilization of default loans is 52.25 and for current loans it is 43.8. This means loans that default tend to have higher credit utilization
* Mean credit inquiry of default loans is 2.33 and for current loans it is 0.998. This means loans that default tend to have higher credit inquiry
* Income level, fico score, purpose, days with credit, public record and delinquent in past 2 years don’t seem to have a significant correlation with loans being default.

### Exploratory Visualization

The plot below shows correlation matrix of all the features in the dataset



From the plot it can be seen that there is strong correlation between interest rate, amount of credit line utilized and debt to income ratio and there is inverse correlation between fico score and interest rate. Strong correlation or positive correlation means both variables rise or fall together. For the loan data this means people who have utilized more credit line tend to have higher interest rate, higher debt to income ratio and lower fico score. Having correlation between variables can make them redundant in machine learning models as they do not offer any additional information.

### Algorithms and Techniques

This is a classification problem as we are trying to predict whether a loan will default or not. The dependent variable not.fully.paid is binary variable that can only be 0 or 1.

The algorithm needs to be fast as loans applied online need approval decision immediately. There is not a lot of data or time needed to train the model. So neural network would not be a good fit.

I will be using Logistic Regression, Support Vector Machine and Extra Trees Classifier to build the model and make prediction and pick the best model based on F1 score.

Logistic regression converts the dependent variable from a classification type to a probability and then uses linear regression to predict the probability of the dependent variable. Eventually a threshold is used to convert the probability of dependent variable to a classification variable by using the logic if the probability is above threshold then true otherwise false. Logistic regression can be used to solve binary classification problem. Its strength is that it can give probability of the dependent variable, so we can have a measure of likelihood of the dependent variable. Its weakness is that given a threshold, it will not be able to determine class of some dependent variable that are equal to the threshold. Another major limitation of logistic regression is that as the number of features increase, larger sample sizes are to make prediction. Since we only have 14 features, logistic regression can work for our use case. Logistic regression is also easy to interpret.

Support Vector Machine models the features as points in space and tries to form boundary lines between different classes of variables, thus separating each class of features into different planes. Its strength is in dividing features that have a clear separation. It can quickly separate different classes of features when they are distinctively apart. Its weakness is that it fails in situation when points are uniformly distributed. Since the current data can be divided into two classes, support vector machine can separate them into two planes. One issue with SVMs is having to choose a kernel function. Finding the right kernel function is not trivial in many cases. SVMs are known to have good generalization performance, but can be slow in test phase. SVMs are effective in high dimensional spaces. Even when number of features exceed the number of samples SVM can still give good results, but can have poor performance when number of features is much greater than number of samples. Unlike logistic regression, SVM does not give probability and finding probability after classifying labels using SVM requires complex k-fold algorithms, which can have poor performance. SVM is suitable for current scenario, because we only need to determine if loan will default or not and we are not required to give probability estimates. The number of features are also not that large.

Extra Trees Classifier is an ensemble algorithm. It uses an ensemble of trees and then averages the output from all the trees. It is comparably fast, works with large number of features and can give better accuracy than Logistic Regression or Support Vector Machine.

We need to apply feature scaling to the dataset before using any of these machine learning models on them. The main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage especially for SVM is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors. In case of linear kernel and the polynomial kernel, large attribute values might cause numerical problems.

### Benchmark

Credit utilization seems to be one of the primary feature that determines if a loan is likely to default. We could assume that everyone who has 50% credit utilization is going to default. Looking at the dataset 54.59% of the loans that defaulted had credit utilization of 50% or more. We can use this as benchmark to evaluate model performance.

#### Baseline F1 Calculation

Total number of loans that defaulted = 1533

Loans with more than 50% credit utilization that defaulted = TP = 837

Loans with 50% or less credit utilization that defaulted = FN = 696

Total number of loans that did not default = 3467

Loans with more than 50% credit utilization that did not default = FP = 1433

Loans with 50% or less credit utilization that did not default = TN = 2034

Precision = P = TP / (TP + FP) = 837 / (837 + 1433) = 0.3687

Recall = R = TP / (TP + FN) = 837 / (837 + 696) = 0.5459

Baseline F1 score = 2\*P\*R / (P+R) = 2 \* 0.3687 \* 0.5459 / (0.3687 + 0.5459) = 0.4401

Our objective is to find a model that can perform better than the baseline F1 score of 0.44.

## Methodology

### Data Preprocessing

There were no missing values in the dataset. Looking at the data it seems purpose is a set of categorical string values consisting of 'debt\_consolidation’, ‘all\_other’, 'credit\_card’, ‘small\_business’, ‘home\_improvement’, 'educational’, ‘major\_purchase'. This is converted to numerical factor values from 0 to 6. OneHotEncoding is then applied on purpose. Feature scaling is then performed to normalize the data. Feature scaling was done by dividing each feature with its mean. As described in algorithms and techniques section, feature scaling was necessary to avoid large features from dominating and SVM needs features to be scaled to avoid large numerical issue. Dataset is split into training and testing set with 30% or 1500 data points in testing set and rest 3500 in training set.

### Implementation

First the dependent variable is separated from independent variable to create labels and features as separate data frames. The data set is split into training and testing set. Logistic Regression, Support Vector Machine and Extra Trees Classifier algorithms are tried to see which one gives the best result. The table below shows the results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Training accuracy | Test accuracy | F1 Score | Training time (sec) | Prediction time (sec) |
| LogisticRegression | 0.8 | 0.787 | 0.53 | 0.026 | 0.0 |
| SVM | 0.81 | 0.79 | 0.517 | 2.15 | 0.132 |
| ExtraTreesClassifier | 1 | 0.78 | 0.53 | 0.047 | 0.0 |

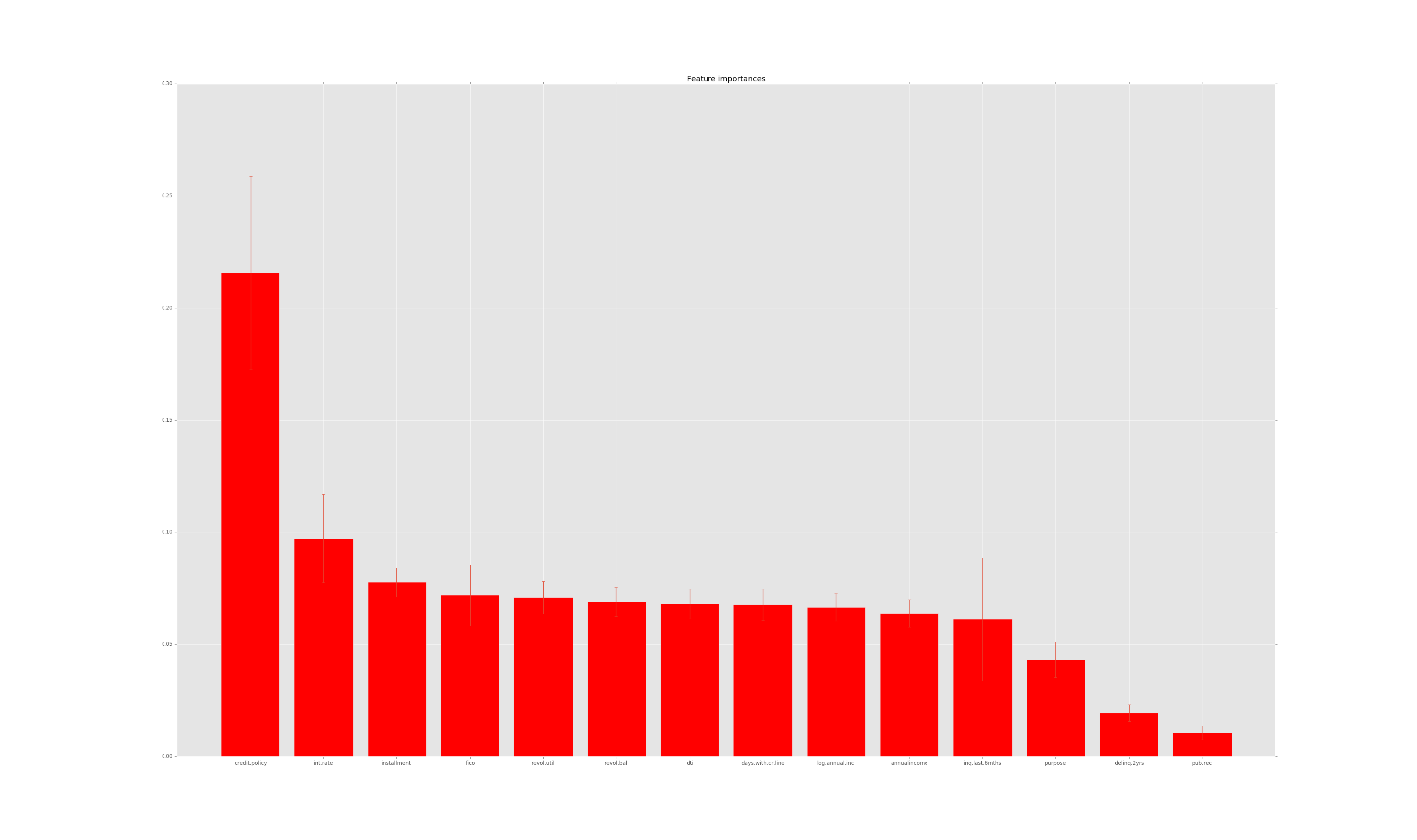
### Refinement

ExtraTreesClassifier is used as the final model as it gives the best F1 Score. The models run very fast and return results within seconds. Using GridSearchCV on ExtraTreesClassifier the final model has F1 score 0.53, which is not an improvement over the default parameter. So ExtraTreesClassifier with default parameter is the best model for our use case.

## Results

### Model Evaluation and Validation

Using ExtraTreesClassifier in sklearn, the features are ranked by importance as shown in figure below



It seems the most important features in the order of importance are:

1. credit.policy
2. int.rate
3. installment
4. fico
5. revol.util
6. revol.bal
7. dti
8. days.with.cr.line
9. log.annual.inc
10. annualincome
11. inq.last.6mths
12. purpose
13. delinq.2yrs
14. pub.rec

This is in line with what we found from data exploration.

### Justification

The final model has accuracy score of 78% and F1 score 0.53, which is more than the rough estimate of accuracy score 54.59% and F1 score 0.44

## Conclusion

This model can be used to predict loans that will be at risk. This can be used to decide whether to approve a loan or not and proactive action can be taken on existing loans that are likely to default.

In order to find a model to predict whether a loan will default or not, I used historical data of past loans where some defaulted and some didn’t. After data transformation and feature scaling, the data was split into training and testing set. LogisticRegression, Support Vector Machine and ExtraTreesClassifier algorithms were used to train a model and predict if loan will default or not. Using F1 score, it was found that ExtraTreesClassifier had the best performance. Using GridSearchCV it was found that the ExtraTreesClassifier works best with default parameters in this case. The final model could be used in an online system to take decision on loan applications.

Although implementing there were no difficulties in implementing the model and it performs better than educated guess, there are still more options that could be tried to improve the decision model even further. For example, using feature ranking, we know which features influence loans to default the most. We could try putting higher weight on the features that are more important or remove features that are not important or are correlated with other features and that can help simplify the model and make it even more accurate. This is not done in this project to save time. Another issue was the availability of similar datasets from different sources, which could be used to test how well the model generalizes to different datasets.

Another option is to use deep learning. Although deep learning would take more time to train. A model could be trained with historical data and then be used to take decision in an online system and then be updated periodically by training with new data as they become available. This is also beyond the scope for this project.