# “Loan Defaults Prediction Using Machine Learning”

**Arya Patel**

**Abstract - Bank generate majority of income using loan. To maximize their revenue, bank need to identify the credit risk. Loan default is one of the issues which is faced by banks and lead to the financial loss. It is important to manage the loan default. Traditional evaluation methods such as manual analysis of income, expanses and checking credit history used for loan approval process are not enough and prone to error. Nowadays, bank use machine learning algorithm to analysis of client profile to predict the loan defaults possibility.**

## Introduction

This kind of problems generally fall under classification problem in machine learning. My aim is to build an accurate classifier using machine learning techniques to identify chance of loan default. In this project data will be predicted on two class. “0” represent that borrower has repay the loan, whereas “1” represent borrower has defaulted on loan repayment.

To achieve desirable accuracy, I will build machine learning model on different algorithms such as Random Forest Classifier and SVM, and I will compare that output using model evaluation matrices and K-fold cross validation. I will also perform some visualization to support my outcome.

## Dataset & Analysis

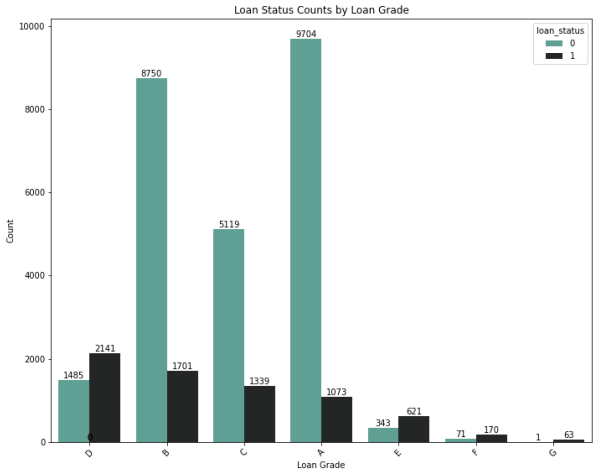
In this project I used dataset from Kaggle for credit risk analysis. Dataset has 31680 rows and 12 columns. Let’s view the each and every column of the dataset to decide what information that contain.

| Column | Information |
|---|---|
| person_age | Age of Person |
| person_income | Annual Income |
| person_home_ownership | Home Ownership |
| person_emp_length | Employment Length (in years) |
| loan_intent | Intention of Taking Loan |
| loan_grade | Category of Loan |
| loan_amnt | Loan Amount |
| loan_int_rate | Interest Rate |
| loan_status | Loan Status |
| loan_percent_income | Percent Income |
| cb_person_defult_on_file | Historical Default |
| cb_preson_cred_hist_length | Credit History Length |

Our target variable is ‘loan_status’. It has binary class ‘0’ and ‘1’ here ‘0’ indicate there is no loan default, whereas ‘1’ indicate loan default. We have total 24854 observation marked as 0, and 6457 observation marked as 1
Out of 12 columns 'person_age’, ‘person_income', ‘loan_status’, ‘cb_person_cred_hist_length’ has datatype as in64 ‘person_emp_length’, ‘loan_int_rate’, ‘loan_percent_incom’ has datatype as flot64 ‘person_home_ownership’, ‘loan_intent’, ‘loan_grade’, ‘cb_person_defult_on_file’ has datatype as object.
Let’s check all the unique value of the object data.

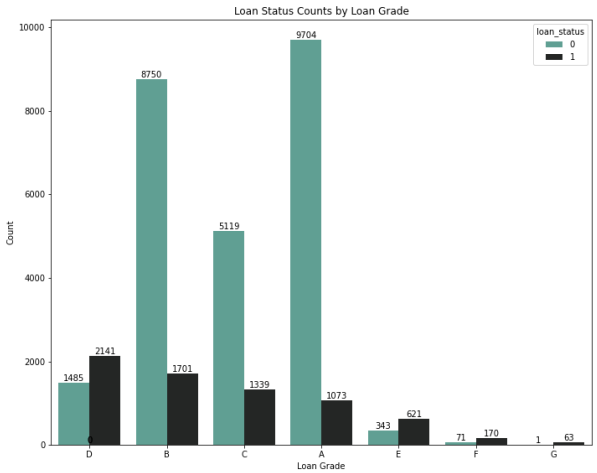
| Column | Unique Values |
|---|---|
| person_home_ownership | ‘RENT’,<br>‘OWN’,<br>‘MORTGAGE’,<br>‘OTHER’ |
| loan_intent | ‘PERSONAL’,<br>‘EDUCATION’,<br>‘MEDICAL’,<br>‘VENTURE’,<br>‘HOMEIMPROVEMENT’,<br>‘DEBTCONSOLIDATION’ |
| loan_grade | ‘A’,<br>‘B’,<br>‘C’,<br>‘D’,<br>‘E’,<br>‘F’,<br>‘G’ |
| cb_person_default_on_file | ‘Y’,<br>‘N’ |

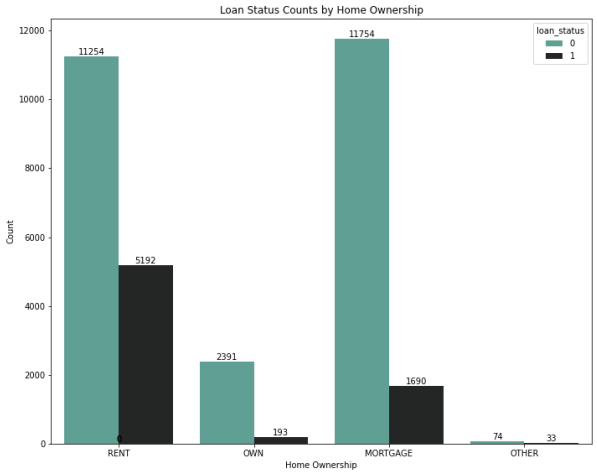
Here we can see that the person_home_ownership & loan_intent are nominal data, whereas loan_grade is ordinal data. Loan grade ‘A’ has lowest mean of interest rate, while grade ‘G’ has highest interest rate on and average. Let’s plot a bar chart to see how loan grade affects the loan status



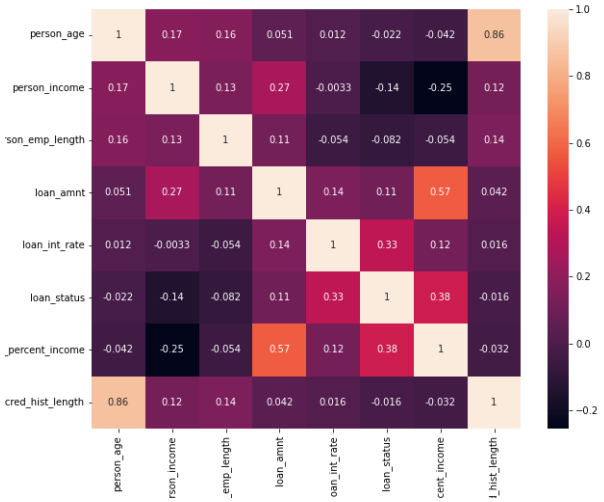

From the prima facie glance, we can observe that loan grade ‘D’ as the highest loan default at 2141, whereas loan grade ‘G’ has the lowest loan default at 63. Loan grade ‘A’ has more non- loan default compare to loan default

Let's visualize the personal_home_ownership and loan_intent columns



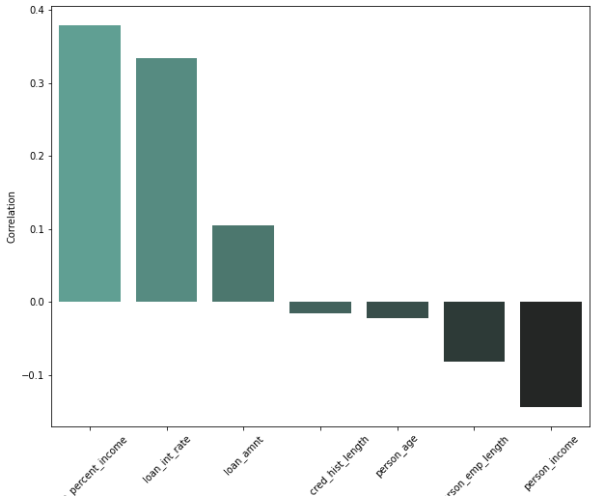




Let's plot the correlation heatmap to see the coleniarity between the columns.




In above correlation heatmap light color show the positive correlation, while dark color show the negative correlation. In this chart coliniarity between loan_status and percent_income, lone_int_rate, and loan_amnt is positive which shows that as value of thos column increases loan status is more likely to be 1.

Barplot : correlation with loan status

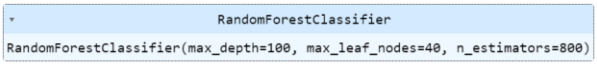


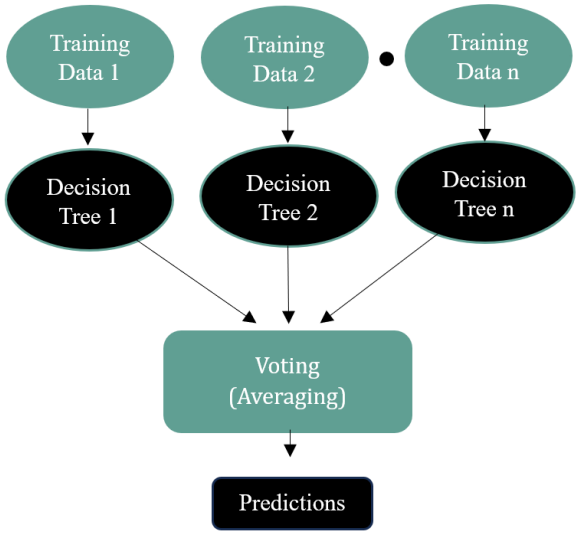

# METHOD

Our goal is to classify the non-loan default and loan default. I built Random Forest classifier and Linear SVM to classify into binary class.

## *A. Random Forest*

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset"



### B. *Support Vector Machine*

Hyperplane is created by Support Vector Machine to classify the loan default. Hyperplane that maximizes prgeometric margin with respect to labelled input data is determined by the SVM

```
LinearSVC
LinearSVC(max_iter=2000)
```

$$min_{\gamma,w,b}\frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\epsilon_i$$

$$s.t.\ y^{(i)}(w^T x^{(i)} + b) >= 1 - \epsilon_i,\ i = 1, ..., m$$

$$\epsilon_i >= 0,\ i = 1, ..., m$$

### C. *Data Splitting Strategy*

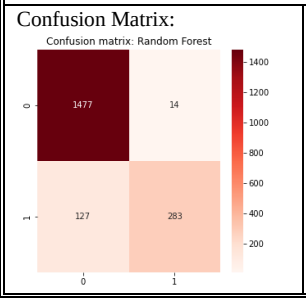
I divided my dataset into three splits – 70 percent for training purpose, 15 percent for cross-validation, and 15 percent for testing purpose. I used stratified sampling that allow me to increase randomness in train-test split

| Split | Default | Non-Default | Total |
|---|---|---|---|
| Train | 4,778 | 17,398 | 22,176 |
| CV | 1,638 | 5,965 | 7,603 |
| Test | 410 | 1,491 | 1,901 |
| Total | 6,457 | 24,854 | 31,680 |

## RESUTL & DISCUSSION

In this section, I will discuss result I achieved with the both method on train, cross-validation and test split. I calculate the model performance with the help of recall, precision, area under precision-recall curve (AUPRC), f1 score.

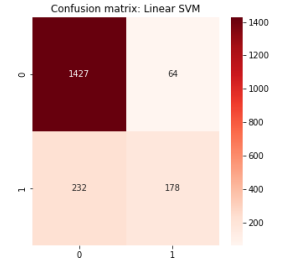
### A. *Random Forest*

| Test Data | | | |
|---|---|---|---|
| | Precision | Recall | F1-Score |
| 0 | 0.92 | 0.99 | 0.95 |
| 1 | 0.95 | 0.69 | 0.80 |
| Confusion Matrix:  | | Cross_validation Score: 0.917<br><br>Area Under Precision Recall Curve (AUPRC): 0.85 | |



The precision and Recall value of class 0 is 0.92 and 0.99 respectively. Specifically, model has high precision value for class 0 that determine that model is 0.91% correct while predicting the class 0. On the other hand, Precision and Recall value of class 1 is 0.95 and 0.69 that tells us that model is 0.95 percent correct while predicting the class 1. Recall value for class 1 indicate that it has comparatively lower recall for class 1 than class 0 which indicate that there is a room for improvement in its performance on class 1.

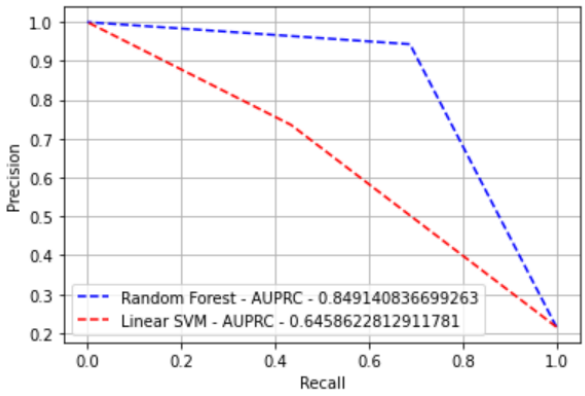
### B. *Liner SVM*

| Test Data | | | |
|---|---|---|---|
| | Precision | Recall | F1-Score |
| 0 | 0.86 | 0.96 | 0.91 |
| 1 | 0.74 | 0.43 | 0.55 |
| Confusion Matrix:  | | Cross_validation Score: 0.847<br><br>Area Under Precision Recall Curve (AUPRC): 0.64 | |



The precision and Recall value of class 0 is 0.86 and 0.96 respectively. Specifically, model has high precision value for class 0 that determine that model is 0.86% correct while predicting the class 0. On the other hand, Precision and Recall value of class 1 is 0.74 and 0.43 that tells us that model is 0.74 percent correct while predicting the class 1. Recall value for class 1 indicate that it has comparatively lower recall for class 1 than class 0 which indicate that there is a room for improvement in its performance on class 1.

### C. *Random Forest vs. Linear SVM*




Random Forest algorithm performs better than LinearSVM