Capstone Project I
Lending Club loan status prediction

Synopsis
Lending Club is a US peer-to-peer lending company, which operates an online lending platform that enables borrowers to obtain a loan, and investors to purchase notes backed by payments made on loans. Lending Club is the world's largest peer-to-peer lending platform. The company claims that $15.98 billion in loans had been originated through its platform up to December 31, 2015.

To find good borrowers is key for the investors in Lending Club to make profit. How to find a good borrowers? In other word, how to predict the loan status according to the information of the loan applicants? A predictive model of loan status can help the investors to make a wise decision for the loan applicants and decrease the risk of bad loans.

In this report, we attempt to predict the risk of the loan being default based on the past loan data. We obtained data from Lending Club’s website. We also get external data resource for the unemployment rate in United States by zip codes from Kaggle. The assumption is if a borrowers from a zip code area with high unemployment rate he is more likely to have a bad loan. The investors can make a judgement according to the zip code of the borrowers for they correlate with Unemployment rate.

We use loan data from year 2015 and perform data wrangling, explanatory data analysis and statistical hypothesis testing. We also split the data set into training and test data sets and apply tree based methods to build predictive models for the loan status. We found that, among multiple machine learning algorithms that we tried, Gradient boosting method provided a reasonable trade-off performance, and a higher return than the naive loan picking strategy can be achieved.

Target variable
The target variable in the project is loan status. There are 7 types of loan status. Among them 'Current', 'Fully Paid' and 'in grace period' mean that the lenders are paying the loan on time or have already paid the loan off. So these three status are called 'good loan'. And the rest four status mean the lenders have delayed the payment more than 15 days or even have no ability to make any payment at all. These four status are called 'bad loan'.

Data wrangling

a. Data wrangling of unemployment rate data sets 2015

1. Get unemployment rate in different zip code in US
Data resources:
a. https://www.kaggle.com/jayrav13/unemployment-by-county-us
b. https://www.gaslampmedia.com/download-zip-code-latitude-longitude-city-state-county-csv/
c. https://github.com/liyepeng/Spring-Board-Data-Science-Track/blob/master/Lending%20Club%20project/Three_state_unemploy.csv

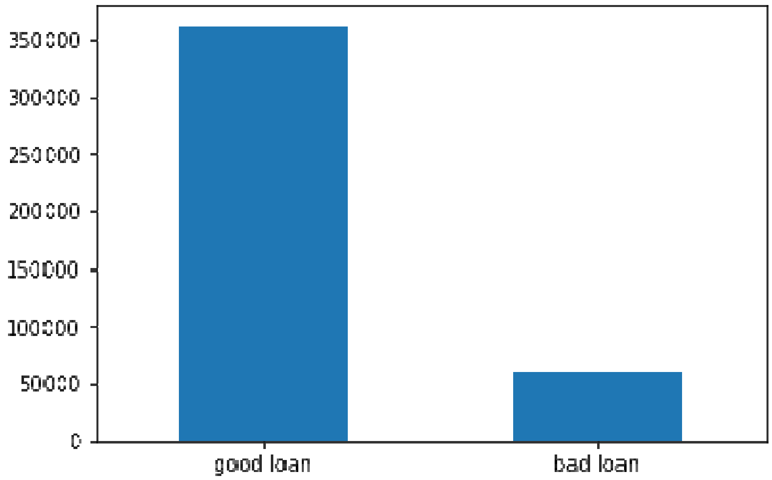
2. Data cleaning:
Data 'a' is about unemployment rate in counties of USA in 2015. Cleaning job is to change states name to abbreviation, and add missing states unemployment information (3 states) from data 'c'.
Data 'c' is the information of counties and corresponding zip codes. This data set contains 62 states among them there are associate states, military base. And the some county names are different from that in data a. Keeping 50 states information and making the counties' name match in both data a and data c are the majority of the cleaning work.
The last step is to combine data 'a' and data b together. One county may has several zip code (XXX00 format) and one zip code may correspond with several county.

b. Data wrangling for lending club data set

Data resources: https://www.lendingclub.com/info/download-data.action
1. Filling blank with 'NaN'. Because the tree based algorithm doesn't matter the missing values I keep these values in original status.
2. Deleting blank columns, deleting post loan and hard ship variables.
3. My target variable is loan status and try to keep the variables it help to make the decision for the loan.
4. Data format wrangling: get rid of '%', change the strings to lower case, check if there are duplicate records.
5. Feature engineering: There are more than 100000 categories for 'emp_title' variable. I choose top ten titles and create 10 dummy variables so it can be used in EDA and model building.
6. Concatenating unemployment rate in different zip code file with lending club data set on zip code. My aim is to check if the unemployment rate has connection with loan status.
7. There are some outliers in the part of the numerical variables. The way to deal with them is depending on what kind of machine learning algorithms I plan to use.

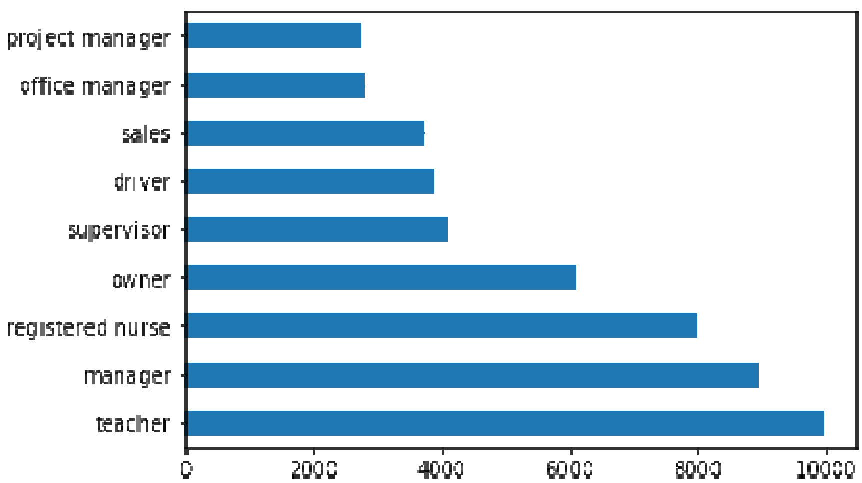
Explanatory Data Analysis

How many good loans and bad loans are there?



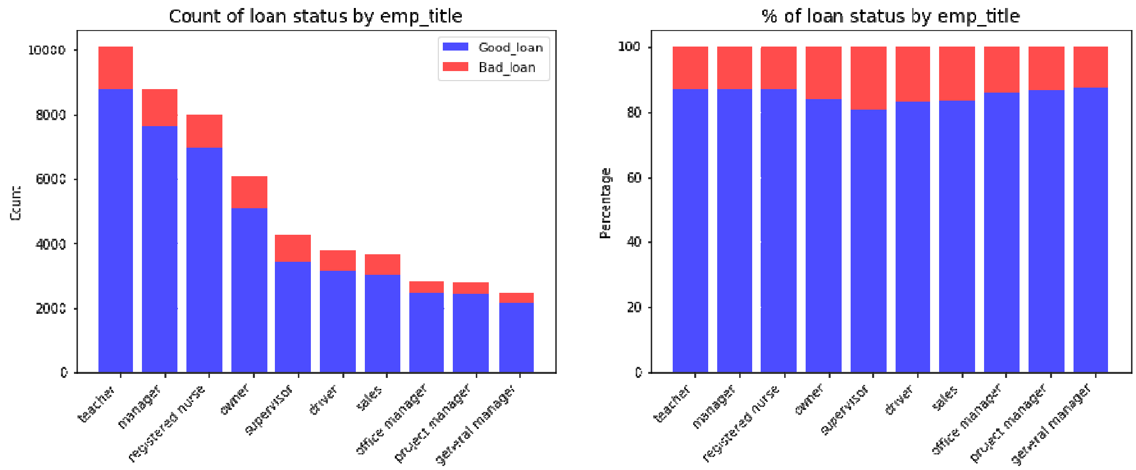
Among all the loans about 20% is bad loans and 80% is good loans.

The loan amount and employment title



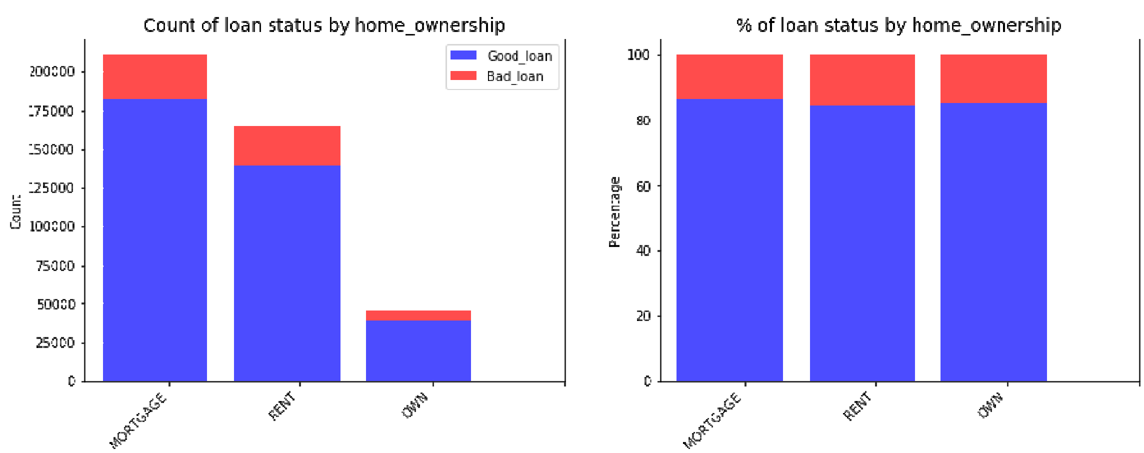
Teacher is the occupation which has the largest number of loans. And project manager has lowest number of loans.

The bad loan percentage for different occupations



Although teachers have largest number of loans they have relatively low bad loan percentage as project managers.

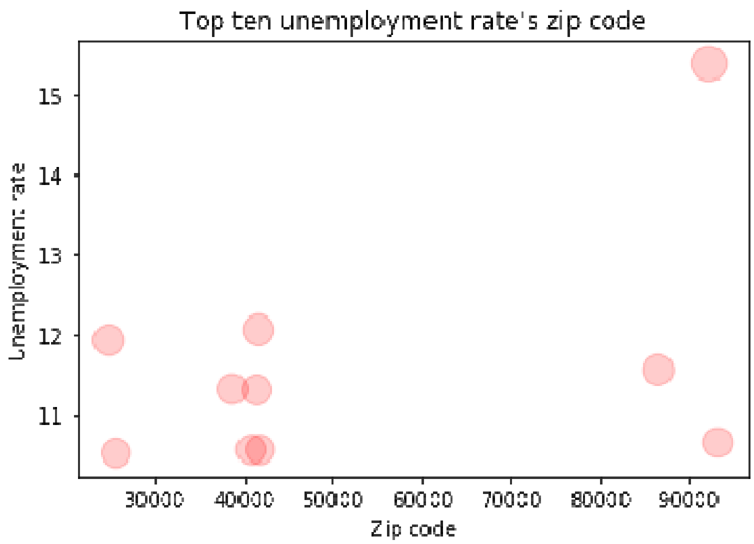
The relationship between loan status and home ownership



The people who have mortgage have largest number of loans while their bad loan percentage relatively low compare to the renters/home owners.

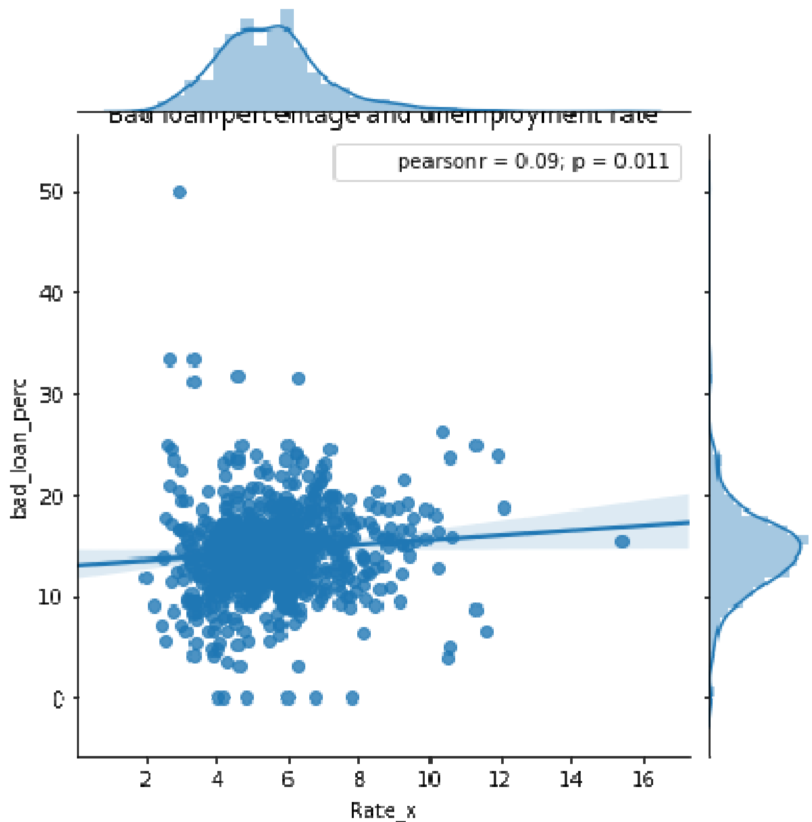
The relationship of zip code, unemployment rate and bad loan percentage

What is the mean unemployment rate in zip codes for bad/good loans?



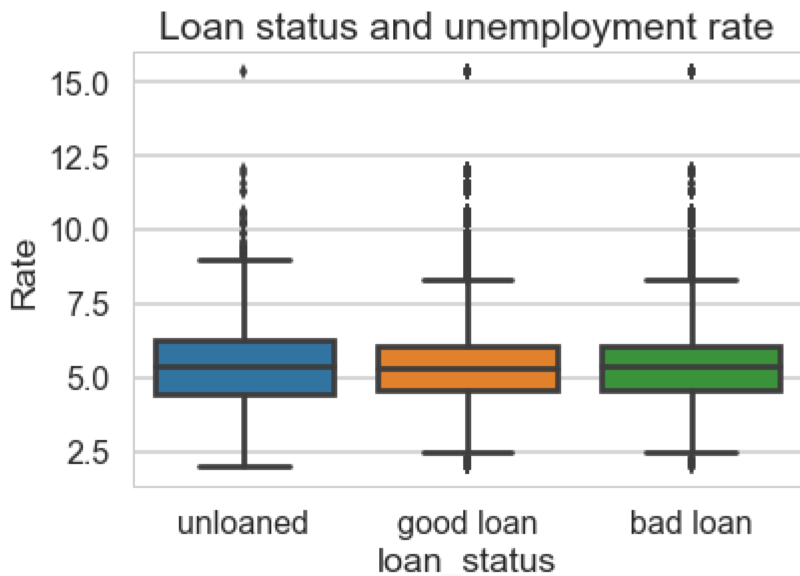
The top ten unemployment rates locate in three zip codes area(20000, 40000, 90000).States CA and AZ's zip codes are around 90000, KY is around 40000, WV and MS are around 20000.

Will the loan lenders in the five states (CA, AZ, KY, WV and MS) which have high unemployment rates also have high bad loan percentage?



From the above plot we can see that the higher the unemployment rate is the higher the bad loan percentage. So we should pay attention to the loan applicants from the five states (CA, AZ, KY, WV and MS) which have high unemployment rates.

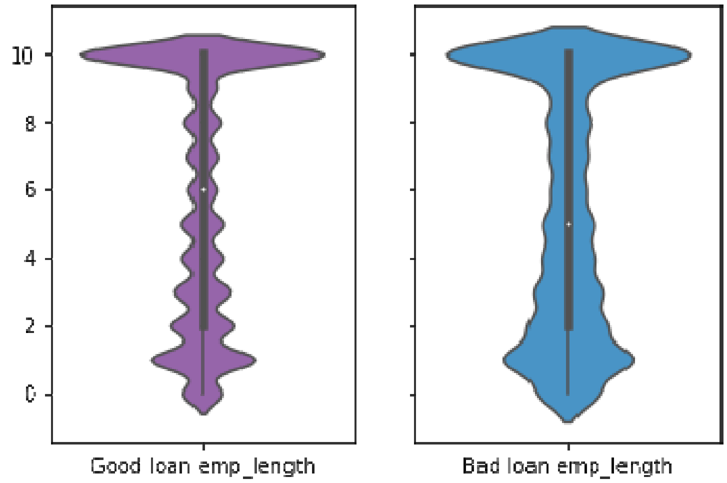
What is the relationship between reported unemployment rate (by applicants in employment status) and average unemployment rate? What about with respect to bad/good loans? In other words, are people that are accepted for a loan (or pay back a loan or don't pay back a loan) typically more unemployed than the rest of the population?



From the above figure we can see the mean unemployment rate for good loan lenders is lower than that of unloaded population and bad loan lenders. The mean calculation shows the mean unemployment rate for unloaded population is lower than that of bad loan lenders.

In other words, people that are accepted for a loan and pay back a loan on time typically more easily employed than the rest of the population. While people that are accepted for a loan and don't pay back a loan typically more likely to be unemployed than the rest of the population.

The relationship of employment length and bad loan percentage

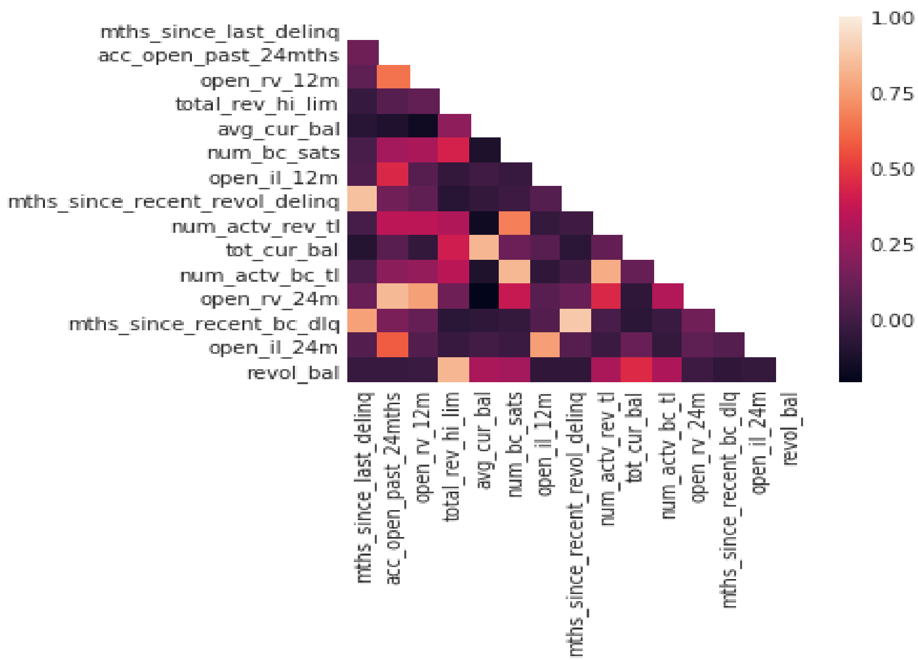


The median employment length of good loan lenders is higher than that of the bad loan lenders. Longer employment length means better financial condition and also the ability for paying the debt on time.

Statistical Analysis report

1. Correlation analysis
There are more than 30 pairs of numeric variables highly correlate and the correlation coefficient are over 0.75.

Correlation hot map for the selected variables:



2. Chi square test for the loan status and employment title
Hypothesis
H_0: In the population, variable 'loan_status' and variable 'emp_title' are independent.
H_1: In the population, variable 'loan_status' and variable 'emp_title' are dependent.

Test statistics
chi_squared_stat: 269.3   Critical value: 21.0260698175   P value: 0.0

Conclusion:
We reject H_0 and consider variable 'loan_status' and variable 'emp_title are dependent. In other words loan status depends on what kind of employment title of the borrowers.

3. Chi square test for loan status and home ownership
Hypothesis
H_0: In the population, variable 'loan_status' and variable 'home_ownership' are independent.
H_1: In the population, variable 'loan_status' and variable 'home_ownership' are dependent.

Test Statistics
chi_squared_stat: 1692.8   Critical value: 11.0704976935   P value: 0.0

Conclusion:
We reject H_0 and consider variable'loan_status' and variable 'home_ownership' are dependent. It means loan status is dependent on whether borrowers are renting a home or have purchased house.

4. Kruskal-Wallis H-test for the median of employment length of different loan status
The Kruskal-Wallis H-test tests the null hypothesis that the population median of all of the groups are equal. It is a non-parametric version of ANOVA. The test works on 2 or more independent samples, which may have different sizes. Note that rejecting the null hypothesis does not indicate which of the groups differs. Post-hoc comparisons between groups are required to determine which groups are different.

Hypothesis:
H_0: The population median of employment length in good loan borrowers and bad loan borrowers are equal.
H_1: The population median of employment length in good loan borrowers and bad loan borrowers are not equal.

Test statistics:
Kruskal Wallis H-test test: H-statistic: 189435635.4   P-Value: 0.0

Conclusion: We reject H_0 and consider the population median of employment length in good loan borrowers and bad loan borrowers are equal. The median of good loan borrowers' employment length is higher than that of bad loan borrowers.

5. Two sample t-test for the mean of unemployment rate in good loan lenders and bad loan lenders

Hypothesis:

H_0: The population mean of unemployment rate of good loan borrowers and bad loan borrowers are equal.
H_1: The population mean of unemployment rate of good loan borrowers and bad loan borrowers are not equal.

Test statistics:
ttest_ind: t = -9.58285   p = 9.71499e-22

Conclusions:
We reject H_0 and consider the mean of unemployment rate of good loan lenders is lower than that of the bad loan lenders.

Model building

Decision Tree

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees.

Decision tree is a method **Easy to Understand**, **Useful in Data exploration** and **Less data cleaning required**. But it is easy to overfitting. What are the key parameters of tree modeling and how can we avoid over-fitting in decision trees? Setting constraints on tree size and Tree pruning are two ways to avoid the over-fitting of decision trees.

In this project using cross validation we tuned **Maximum depth of tree (vertical depth)** which is used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample.

For the hyper parameter **Minimum samples leaves** which defines the minimum samples (or observations) required in a terminal node or leaf. Used to control over-fitting similar to min_samples_split. Generally lower values should be chosen for imbalanced class problems because the regions in which the minority class will be in majority will be very small. In this project we choose 50 as the value of **Minimum samples leaves.**

Hyper parameter tuning for max depth

```
[(3, 0.85669108733167099), (4, 0.85671483517706548), (5,
0.85673066729330494), (6, 0.85673066729330494), (7,
0.85663567777590433), (8, 0.85623987914651389), (9,
0.85583221575084278)]
```

From the scores we obtained from the tuning process max_depth equaling 5 or 6 is the best choice.

Model score using ‘gini index’ and ‘entropy’

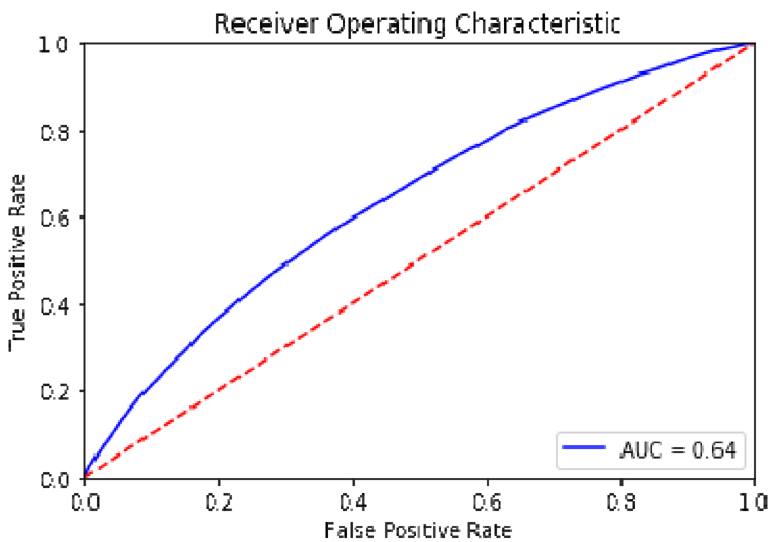
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=6, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=50, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=0, splitter='best')
```

```
AUC - ROC gini : 0.636293062904
```

```
AUC - ROC entropy : 0.637417236881
```

The score of the model using 'entropy' is a little bit higher than that using 'gini index'.

ROC_AUC Curve of Decision tree model



Random Forest

Random forest is an ensemble tool which takes a subset of observations and a subset of variables to build a decision trees. It builds multiple such decision tree and amalgamate them together to get a more accurate and stable prediction.

There are primarily 3 features which can be tuned to improve the predictive power of the model:

1. max_features:

These are the maximum number of features Random Forest is allowed to try in individual tree. Increasing max_features generally improves the performance of the model as at each node now we have a higher number of options to be considered. However, this is not

necessarily true as this decreases the diversity of individual tree which is the USP of random forest. But, for sure, you decrease the speed of algorithm by increasing the max_features.
2. n_estimators:
This is the number of trees you want to build before taking the maximum voting or averages of predictions. Higher number of trees give you better performance but makes your code slower. You should choose as high value as your processor can handle because this makes your predictions stronger and more stable.
3. min_sample_leaf:
If you have built a decision tree before, you can appreciate the importance of minimum sample leaf size. Leaf is the end node of a decision tree. A smaller leaf makes the model more prone to capturing noise in train data. Generally I prefer a minimum leaf size of more than 50.

Features which will make the model training easier:
1. n_jobs:
This parameter tells the engine how many processors are allowed to use. A value of "-1" means there is no restriction whereas a value of "1" means it can only use one processor.
2. oob_score:
This is a random forest cross validation method. It is very similar to leave one out validation technique, however, this is so much faster. This method simply tags every observation used in different tress. And then it finds out a maximum vote score for every observation based on only trees which did not use this particular observation to train itself.

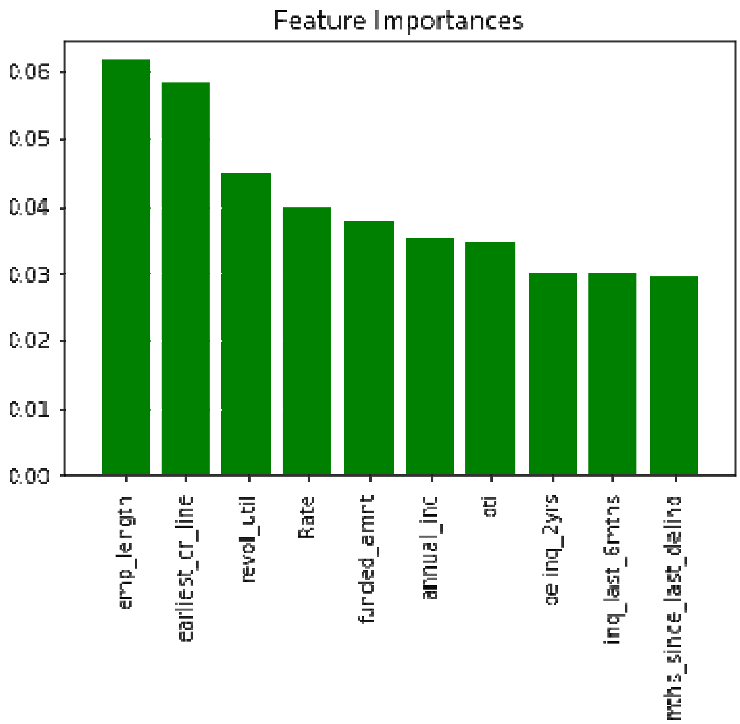
From the above rules we choose the following Random forest model:

```
Forest = RandomForestClassifier (n_estimators=1000, oob_score = True, random_state=0,
          n_jobs=-1, max_features = "auto", min_samples_leaf = 50)
```

Compare the feature importance

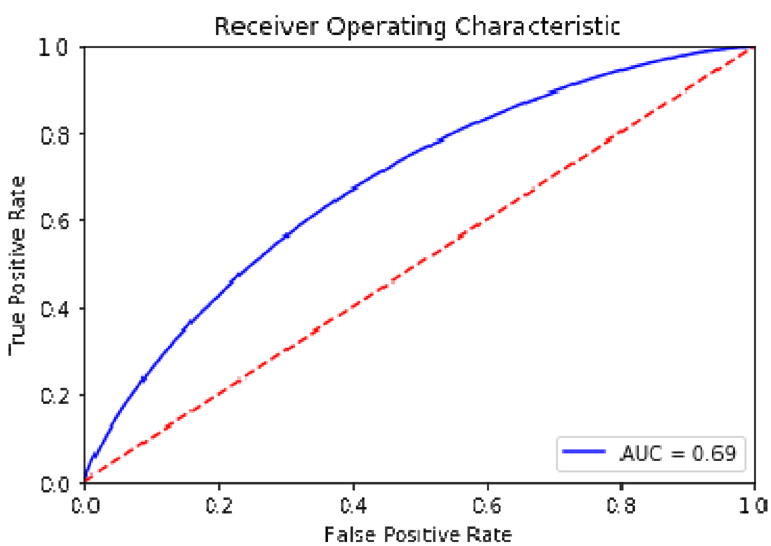
```
 1) emp_length                      0.061432
 2) earliest_cr_line                0.058367
 3) revol_util                      0.044979
 4) Rate                            0.040004
 5) funded_amnt                     0.037900
 6) annual_inc                      0.035266
 7) dti                             0.034621
 8) delinq_2yrs                     0.030147
 9) inq_last_6mths                  0.030045
10) mths_since_last_delinq          0.029514
```

Emp_length and unemployment rate variables are among the top five important features. We discussed these two features in EDA part and found them be important to the loan status. So the modeling building part agrees with the EDA part on these two variables.



Score the random forest model and plot roc curve

```
AUC - ROC :  0.685429646528
```



Gradient Boosting Method

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Let's consider the important GBM parameters used to improve model performance in Python:

1. learning_rate

This determines the impact of each tree on the final outcome. GBM works by starting with an initial estimate which is updated using the output of each tree. The learning parameter controls the magnitude of this change in the estimates.

Lower values are generally preferred as they make the model robust to the specific characteristics of tree and thus allowing it to generalize well. Lower values would require higher number of trees to model all the relations and will be computationally expensive.

2. n_estimators

The number of sequential trees to be modeled. Though GBM is fairly robust at higher number of trees but it can still over fit at a point. Hence, this should be tuned using CV for a particular learning rate.

3. subsample

The fraction of observations to be selected for each tree. Selection is done by random sampling. Values slightly less than 1 make the model robust by reducing the variance. Typical values ~0.8 generally work fine but can be fine-tuned further.

Tuning the learning rate

```
learning rate 0.0001 mean scores: 0.856691087231
learning rate 0.001 mean scores: 0.856691087231
learning rate 0.005 mean scores: 0.856691087231
learning rate 0.01 mean scores: 0.856691087231
learning rate 0.05 mean scores: 0.856885026801
learning rate 0.1 mean scores: 0.85705917701
```

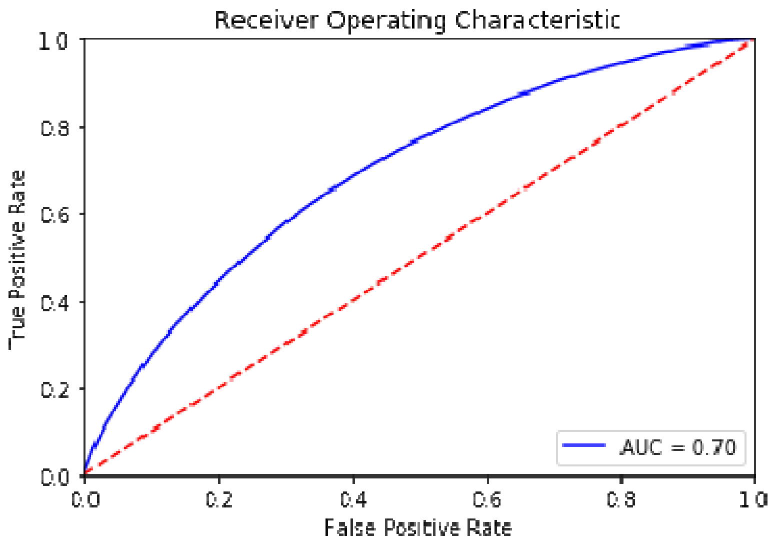
Learning rate decides the number of trees. The lower the learning rate the more of trees needed. Learning rate 0.1 performs better than other smaller rates.

Final models of GBM:

```
GradientBoostingClassifier(criterion='friedman_mse', init=None,
learning_rate=0.2, loss='deviance', max_depth=2, max_features=None,
max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=200, presort='auto',
random_state=None, subsample=1.0, verbose=0, warm_start=False)
```

Scoring the GBM model

```
AUC - ROC : 0.695163508282
```



GBM model gets a score of 0.70 and a little bit improvement than the random Forest.

To predict the loan status we tried Decision Tree, Random Forest and Gradient Boost method. We tuned hyper parameters and used cross validation. Finally we increase the ROC_AUC score from 0.64 to 0.70. Gradient Boost method proved to be the best method for the predictive model.

There is still room for the improvement of model performance, however. In the EDA and statistical part we analyzed four features which are employment length, unemployment rate, employment title and home ownership. They all have impacts on the bad loan ratio and statistical testing shows the impact has statistical significance. But from the feather importance analysis we can only see the employment length and unemployment rate are among the top ten important features. The employment title's importance is zero. It is worth to dig deeper for the feature "employment title". For example we can explore more than 10 employment titles and see if it will increase the importance of the feature. Or we can connect employment title with employment length and create a new variable, because the income for a teacher working for ten years will have great difference with a manager working for ten years.