11/09/2018

GROUP PROJECT BY CODE NINJAS

CREDIT WORTHY

Submitted towards the partial fulfillment of the criteria for award of Genpact Data Science Prodegree by Imarticus.

Predict loan worthiness of the applicants for XYZ Corp loan passing.

Submitted By: Code Ninjas

Shalini Ubbey

Sumit Damania

Swati Garg

Vaibhav Pise

DSP 14, July 2018

ABSTRACT

Everyday a large number of people make application for loans, for a variety of purposes. But all these applicants are not reliable and everyone cannot be approved. Every year, we read about a number of cases where people do not repay bulk of the loan amount to the banks due to which they suffer huge losses. The risk associated with making a decision on loan approval is immense. So the idea of this project is to gather loan data from multiple data sources and use data mining algorithms on this data to extract important information and predict if a customer would be able to repay his loan or not. In other words, predict if the customer would be a defaulter or not.

ACKNOWLEDGEMENT

We would like to thank Miss. Nikita Tandel, for giving us the opportunity to do Data Science Pro Degree’s final group project under her guidance. We are extremely thankful to her for giving us her invaluable timely feedback after the completion of every milestone and helping us throughout the course to be able to complete this project with success. We would also like to thank Mr. Arun Upadhyay, our course administrator, for explaining us how the report should be written and giving us this opportunity to work in a professional environment.

TABLE OF CONTENTS

INTRODUCTION

1.1 PROJECT BACKGROUND

In today's time people are becoming more and more dependent on acquiring loans, be it education loan, housing loan, car loan, business loans etc. from the financial institutions like banks and credit unions. In some cases, people undergo sudden financial crisis while some try to scam money out of the banks. The consequences of such scenarios are late payments or missing payments, defaulting or in the worst-case scenario not being able to pay back those bulk amount to the banks. Assessing the risk, which is involved in a loan application, is one of the most important concerns of the banks hence most of the banks use their own credit scoring and risk assessment techniques in order to analyze the loan application and to make decisions on credit approval.

1.2 GOAL OF THE PROJECT

Our aim is to help XYZ Corp to set loan passing criteria, grant loan to worthy applicants and avoid risk of default. The primary goal of this project is to extract patterns from a common loan approved dataset, and then build a model based on these extracted patterns, in order to predict the likely loan defaulters by using classification data mining algorithms. The historical data of the customers like their age, income, loan amount, employment length etc. will be used in order to do the analysis. Later on, some analysis will also be done to find the most relevant attributes, i.e., the factors that affect the prediction result the most.

1.3 DATA INTRODUCTION

The dataset given contains complete loan data for all loans issued by XYZ Corp. through 2007-2015that comprises of 855970 observations and 73variables (parameters to be considered to make prediction). **'Default_ind'** is 'Y variable' used to draw conclusion. Data provided was split as per Issue date (issue_d) into Train data (Ranging from June 2007 - May 2015, used to train model to study trends) and Test data (Ranging from June 2015 - Dec 2015, used to make predictions on basis of trends studied from Train Data analysis).Conclusion is drawn from the accuracy of the model and confusion matrix created to predict loan worthiness.

MODELS APPLIED AND MOTIVATION:

2.1 Logistic Regression:

With logistic regression, outputs have a nice probabilistic interpretation for the set of predictor variables, and the algorithm can be regularized to avoid over fitting. Hence, we choose to build logistic regression classifier. However, the results were not that great. 'Type 1 Error' increased drastically when threshold was tuned for 'Type 2 Error' reduction. Tuning the model by using up sampling and Ada boosting also were not very effective in giving good balance of Accuracy, Type 1 and Type 2 errors.

Train Data Observations before Up sampling:

```
Out[56]:
0    552822
1     46156
Name: default_ind, dtype: int64
```

Train Data Observations after Up sampling:

```
In [31]: train_upsampled.default_ind.value_counts()
Out[31]:
1    552529
0    495510
Name: default_ind, dtype: int64
```

```
[[247949    399]
 [    66    245]]

Classification report :
             precision    recall  f1-score   support

          0       1.00      1.00      1.00    248348
          1       0.38      0.79      0.51       311

avg / total       1.00      1.00      1.00    248659

accuracy of the model :  0.998129969154545
```

2.2 Random Forest Classification:

As the given data is skewed, we considered using Random forest model for the predictor variable subset of features in each of its decision trees (for RandomForestClassifier= 25 and random_state = 10).Thereby reducing the bias of the model. The final output will be the mode of the outputs of all its decision trees which has better results than decision tree. As, decision tree might possibly over fit. Random forest gave us a better output with 99.83% Accuracy, keeping Type 2 error=6 and Type 1 error=395:

```
[[247953    395]
 [     6    305]]

Classification report :
             precision    recall  f1-score   support

          0       1.00      1.00      1.00    248348
          1       0.44      0.98      0.60       311

avg / total       1.00      1.00      1.00    248659

accuracy of the model :  0.998387349744027
```

MODEL CREATION PROCESS

3.1. Transforming Data

3.1.1 Packages used

- import numpy as np
- import pandas as pd
- import seaborn as sns
- import matplotlib.pyplot as plt
- from sklearn import preprocessing
- from sklearn.utils import resample
- from sklearn.utils import resample
- from sklearn.linear_model import LogisticRegression
- from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
- from sklearn import metrics

3.1.2 Treating missing data

- Data not only has null values, also has date variables which need data formatting to make them ready to process.
- Calculated percentage of null values per variable and dropped the variables with more than 50% null values. Remaining variables were filled with mean.

```
df_missing=df.isnull().sum().reset_index()
df_missing.columns=['Col_Name','Num_of_MV']
df_missing=df_missing[df_missing['Num_of_MV']>0]
df_missing=df_missing.sort_values(by='Num_of_MV',ascending=False)
df_missing['Percentage']=(df_missing['Num_of_MV']/len(df))*100
df_missing=df_missing.reset_index()
Max_Missing=df_missing.iloc[0:21,1].values
df=df.drop(Max_Missing,axis=1)
```

Out[3]:

	index	Col_Name	Num_of_MV	Percentage
0	52	dti_joint	855529	99.948596
1	51	annual_inc_joint	855527	99.948363
2	53	verification_status_joint	855527	99.948363
3	63	il_util	844360	98.643759
4	61	mths_since_rcnt_il	843035	98.488964
5	71	inq_last_12m	842681	98.447607
6	60	open_il_24m	842681	98.447607
7	59	open_il_12m	842681	98.447607
8	58	open_il_6m	842681	98.447607
9	57	open_acc_6m	842681	98.447607
10	64	open_rv_12m	842681	98.447607
11	65	open_rv_24m	842681	98.447607
12	62	total_bal_il	842681	98.447607
13	66	max_bal_bc	842681	98.447607
14	67	all_util	842681	98.447607
15	69	inq_fi	842681	98.447607
16	70	total_cu_tl	842681	98.447607
17	17	desc	734157	85.769111
18	27	mths_since_last_record	724785	84.674211
19	48	mths_since_last_major_derog	642830	75.099682
20	26	mths_since_last_delinq	439812	51.381767
21	45	next_pymnt_d	252971	29.553757
22	68	total_rev_hi_lim	67313	7.863953
23	56	tot_cur_bal	67313	7.863953
24	55	tot_coll_amt	67313	7.863953
25	10	emp_title	49443	5.776261
26	11	emp_length	43061	5.030673
27	43	last_pymnt_d	8862	1.035318
28	31	revol_util	446	0.052105
29	47	collections_12_mths_ex_med	56	0.006542
30	46	last_credit_pull_d	50	0.005841
31	19	title	33	0.003855

- Backfilled the remaining variables with mean value.
- Dropped observations with null values who were not defaulters.
- Label encoded variables to transform non-numerical labels to numerical labels.
- Date variables were specially filled with mode values after converting <month><Year> to Datetime format:

```
df['issue_d']=pd.to_datetime(df['issue_d'])
```

3.1.3Random Forest Classification:

Predictor Variables:

On below 33 given features and additional 3 Dummy Variables (derived from date variables) we have performed classification technique by using Random Forest Model:

LoanStatNew	Description
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
addr_state	The state provided by the borrower in the loan application
annual_inc	The self-reported annual income provided by the borrower during registration.
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
collection_recovery_fee	post charge off collection fee
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
Dti	A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested loan, divided by the borrower’s self-reported monthly income.
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
grade	XYZ corp. assigned loan grade
home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
initial_list_status	The initial listing status of the loan. Possible values are – W, F
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
int_rate	Interest Rate on the loan
issue_d	The month which the loan was funded
last_pymnt_amnt	Last total payment amount received
open_acc	The number of open credit lines in the borrower's credit file.
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
pub_rec	Number of derogatory public records
purpose	A category provided by the borrower for the loan request.
pymnt_plan	Indicates if a payment plan has been put in place for the loan
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
tot_coll_amt	Total collection amounts ever owed
tot_cur_bal	Total current balance of all accounts
total_acc	The total number of credit lines currently in the borrower's credit file
total_pymnt_inv	Payments received to date for portion of total amount funded by investors
total_rec_int	Interest received to date
total_rec_late_fee	Late fees received to date
total_rev_hi_lim	Total revolving high credit/credit limit
verification_status	Was the income source verified

Target Variables:

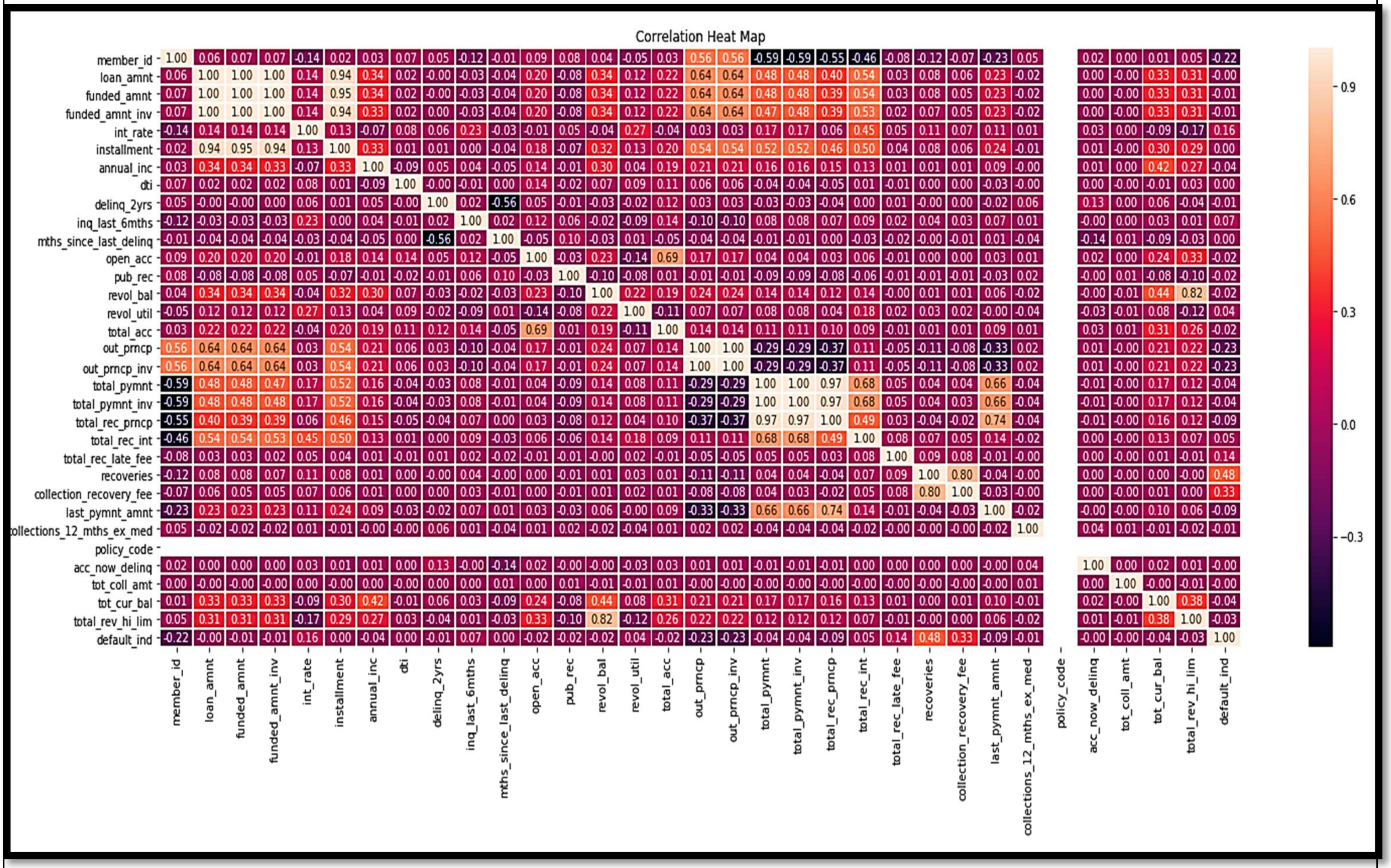
The target variable in our dataset is **'default_ind'** which shows the status of the loan. It is a Dichotomous variable with 2 values – 0 and 1. '0' stands for 'No Default' and '1' stands for 'Default'.

Data Standardization:

Data Standardization is done to normalize numerical data to reduce data redundancy.

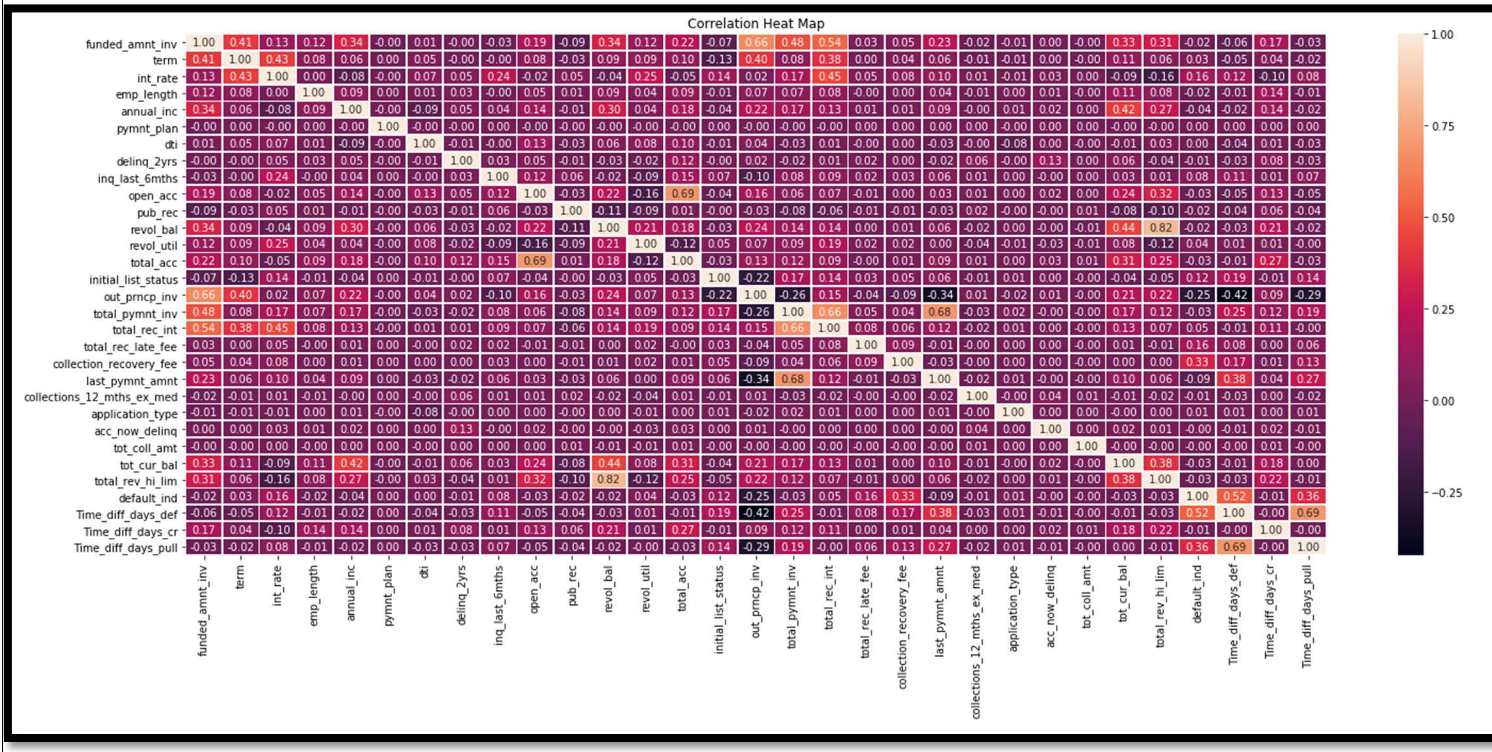
Data Visualization:

Data visualization was an important contributor variable picking, we used Heat Map to view data correlation.



Hence, we dropped the variables that displayed high correlation with each other. This could be read by looking at the color in the heat map and the numerical values in the cells. Lighter the color higher is the correlation.

- ‘Loan_amnt, funded_amnt, funded_amnt_inv, installment’ displayed correlation. Hence, we dropped ‘Loan_amnt, funded_amnt, installment’.
- ‘Total_rec_prncp, total_pymnt, total_pymnt_inv’ displayed correlation. Hence, we dropped ‘Total_rec_prncp, total_pymnt’.
- 'Out_prncp, out_prncp_inv’ displayed correlation. Hence, we dropped'out_prncp’.
- ‘Recoveries, collection_recovery_fee’ displayed correlation. Hence, we dropped‘Recoveries’.
- 'Policy_code’ had uniform value of ‘1’ in all observations. Hence, dropped 'policy_code’.



Ran random forest model for the predictor variables for RandomForestClassifier= 25 and random_state = 10. The final output was of **Accuracy=99.83%**; with **False negative=6** and **False positive=395**.

Following is the Confusion Matrix, Classification report and Accuracy display:

```
[[247953    395]
 [     6    305]]

Classification report :
             precision    recall  f1-score   support

          0       1.00      1.00      1.00    248348
          1       0.44      0.98      0.60       311

avg / total       1.00      1.00      1.00    248659

accuracy of the model :  0.998387349744027
```

RESULT INTERPRETATION

4.1. COMPARATIVE RESULT:

Logistic Regression Output at 0.98 Threshold

```
[[247473    875]
 [    30    281]]

accuracy of the model :  0.9963604776018563
Classification report :
             precision    recall  f1-score   support

          0       1.00      1.00      1.00    248348
          1       0.24      0.90      0.38       311

avg / total       1.00      1.00      1.00    248659
```

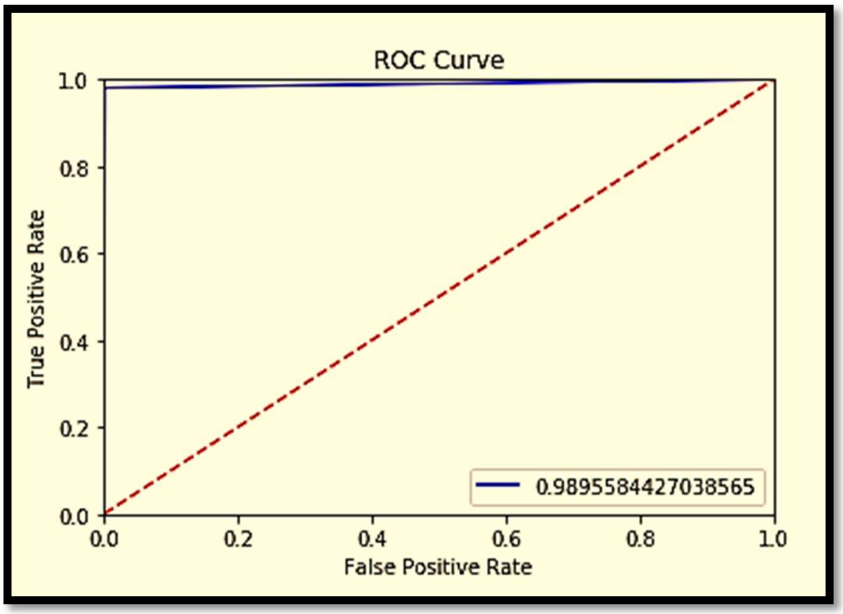
Random Forest Output for 25 Decision Trees

```
[[247953    395]
 [     6    305]]

Classification report :
             precision    recall  f1-score   support

          0       1.00      1.00      1.00    248348
          1       0.44      0.98      0.60       311

avg / total       1.00      1.00      1.00    248659

accuracy of the model :  0.998387349744027
```

4.2 ROC (Receiver Operating Characteristic):

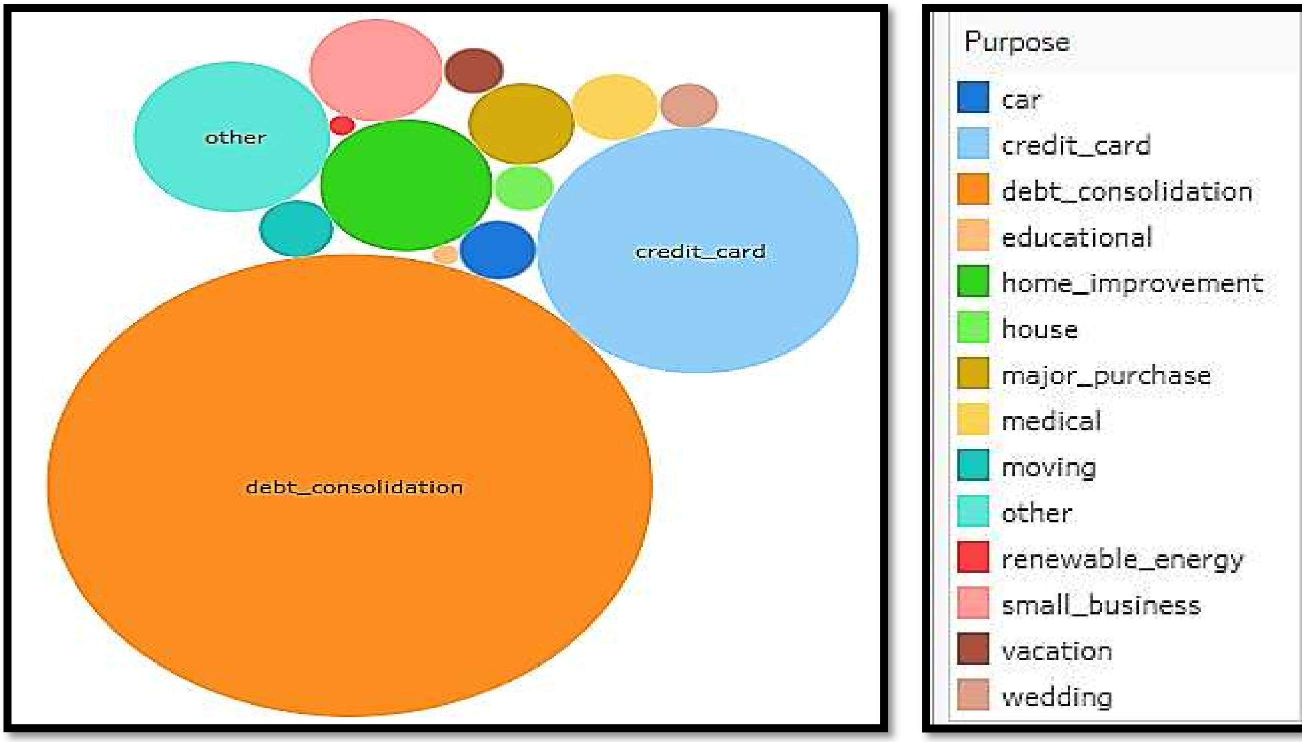
The ROC Curve is the visual output of the accuracy of the model, it gives the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points. Each point on ROC Curve represents a sensitivity/ specificity pair corresponding to a particular decision threshold.

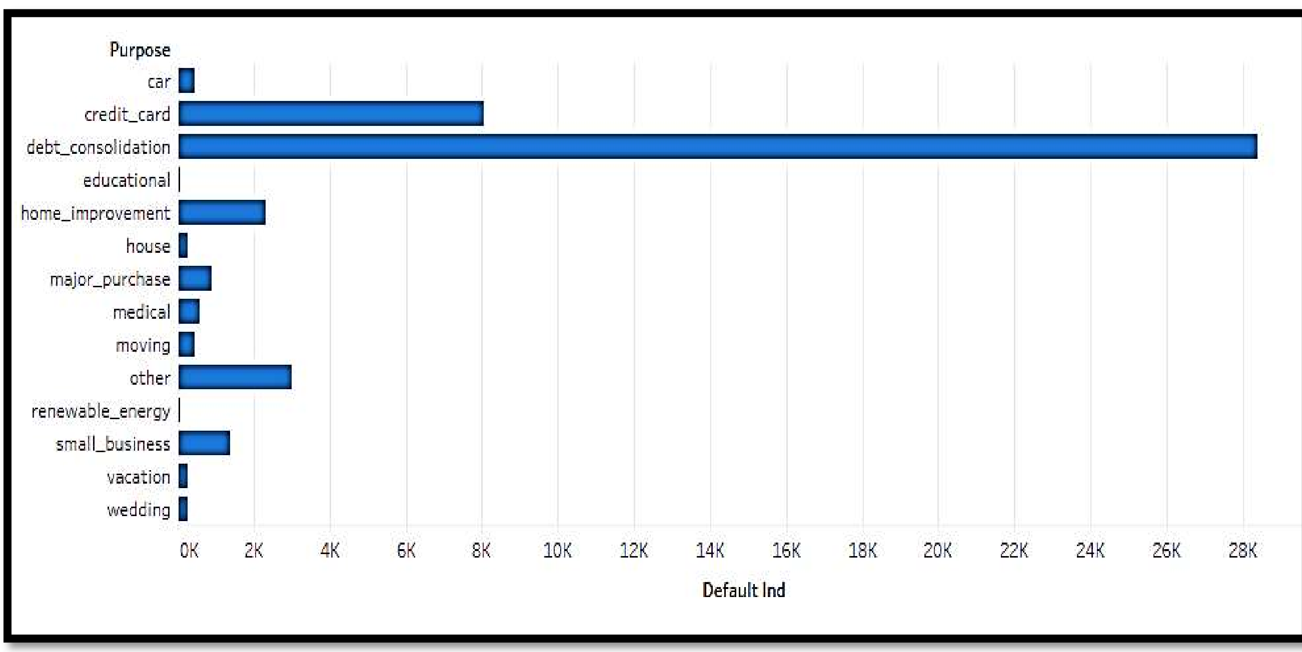
More the area covered in the ROC Curve, better the model is.



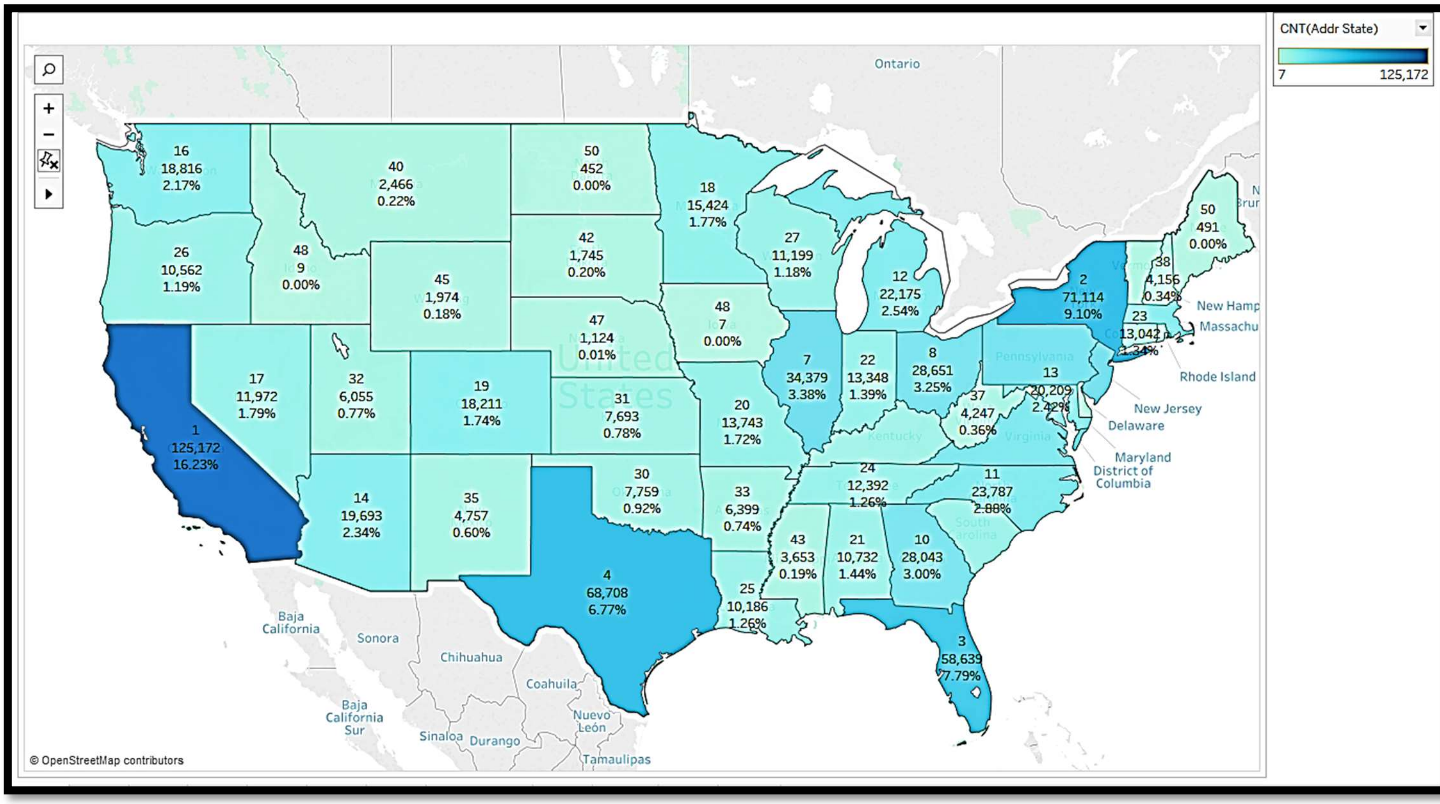
PERSPECTIVE ANALYSIS:

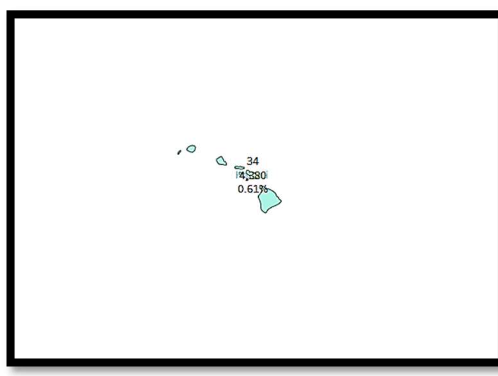
- On Basis of loan purpose we can infer that applicants taking loan for 'debt_consolidation' are the highest. However, that of the 'educational' loan are lowest.



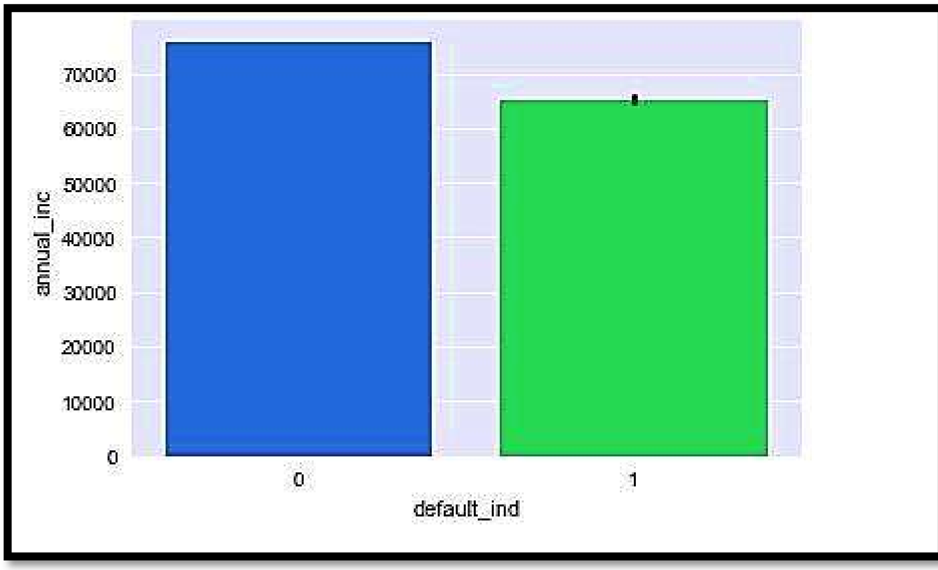


- On Basis of address state we can infer that majority of applicants are from California and percentage of default is also highest.





- On Basis of just annual income (annual_inc) we can infer that the annual income does not influence the applicant to default. There are other factors influencing the default as well. Hence, this model would be helpful in keeping all influencers in check for loan passing criteria.



FUTURE WORK:

- Time Series Analysis can be done using the Loan data of several years, for prediction of the approximate time, when the client can default.
- Future analysis can be done on predicting the approximate Interest rates that the loan applicant is expected to get as per his profile if his loan is approved. This can be useful for loan applicants, since some banks approve loans, but give very high interest rates to the customer. It would give the customers a rough insight regarding the interest rates that they should be getting for their profile and it will make sure they don’t end up paying much more amount in interest to the bank.
- An application can be built, which will take various inputs from the user like, Employment Length, Salary, Age, marital status, SSN, address, loan amount, loan duration etc. and give a prediction of whether their loan application can be approved by the banks or not based on their inputs along with an approximate interest rates.

REFERENCES:

- http://budgeting.thenest.com/mean-loan-goes-underwriting-23201.html
- http://www.investopedia.com/ (a great source to find meanings of BFSI terminology and jargon)
- https://www.google.com/