PROJECT REPORT

Python Project: Bank Lending

Submitted towards the partial fulfillment of the criteria for award of PGA by Imarticus

Submitted By:

G BHAVANI SANKAR (IL014402)
M SAI KUMAR (IL014833)

Course and Batch: DSP-Batch26 [Oct'19-May'20]

Abstract

Keywords

*Disclaimer: *Data shared by the customer is confidential and sensitive; it should not be used for any purposes apart from capstone project submission for PGA. The Name and demographic details of the enterprise is kept confidential as per their owners' request and binding.*

Acknowledgements

We are using this opportunity to express my gratitude to everyone who supported us throughout the course of this group project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, we were fortunate to have **Manish Singh** as our mentor. He has readily shared his immense knowledge in data analytics and guides us in a manner that the outcome resulted in enhancing our data skills.

We wish to thank, all the faculties, as this project utilized knowledge gained from every course that formed the DSP program.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Date: May 17, 2020

Place: Bangalore

Member 1: **M Sai Kumar**

Member 2 : **G Bhavani Sankar**

Certificate of Completion

I hereby certify that the project titled “Bank Lending” was undertaken and completed under my supervision by **M Sai Kumar** and **G Bhavani Sankar** from the batch of PGA (May 2020)

Mentor: **Manish Singh**

Date: May 17, 2020

Place – Bangalore

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Title & Objective of the study

- The objective of our project is to predict whether a loan will default or not based on objective financial data only and whether investors should lend to a customer or not. Data from 2007-2015 will be used because most of the loans from that period have already been repaid or defaulted on.

1.2 Need of the Study

- In today's world, obtaining loans from financial institutions has become a very common phenomenon. Every day many people apply for loans, for a variety of purposes. But not all the applicants are reliable, and not everyone can be approved. Every year, there are cases where people do not repay the bulk of the loan amount to the bank which results in huge financial loss.
- The risk associated with making a decision on a loan approval is immense. Hence, the idea of this project is to gather loan data from the Lending Club website and use machine learning techniques on this data to extract important information and predict if a customer would be able to repay the loan or not. In other words, the goal is to predict if the customer would be a defaulter or not.

1.3 Business Model of Enterprise

- Financial lending is a way to borrow without using a traditional bank or credit union. For applicants with a good credit score (often a FICO credit score higher than 720), P2P loan rates can be surprisingly low. With less-than-perfect credit, an applicant still has a decent shot at being approved for an affordable loan with online lenders like XYZ corporation..

- Financial loans are loans made by individuals and investors – as opposed to loans that come from a bank. People with extra funds offer to lend that money to others (individuals and businesses) in need of cash. A P2P service (such as a website) matches lenders and borrowers so that the process is relatively easy for all involved.

1.4 Data Sources

- The provided dataset corresponds to all loans issued to individuals in the past from 2007-2015. The dataset has 855969 observations and 73 features. The data contains the indicator of default, payment information, credit history, etc. Customers under 'current' status have been considered as non-defaulters in the dataset. We have also been provided with a Data dictionary that best describes the features.

- The dataset has quite a lot of missing values and the figures can be considered as ground truth, but lots of columns are irrelevant, very sparse or non informative. Moreover, the dataset is unbalanced, with approximately 6% of loans considered as defaulted.

1.5 Tools & Techniques

Tools: Python 3.7.2-Jupyter Notebook

Techniques: Logistic regression, Random Forest Classifier, Decision Tree, KNN

1.6 Infrastructure Challenges

- The impact of economic downturn on the behaviors of borrowers as well as lenders.
- Mode of calculation of default probability

CHAPTER 2: DATA PREPARATION AND UNDERSTANDING

- One of the first steps we engaged in was to outline the sequence of steps that we will be following for our project. Each of these steps are elaborated below:

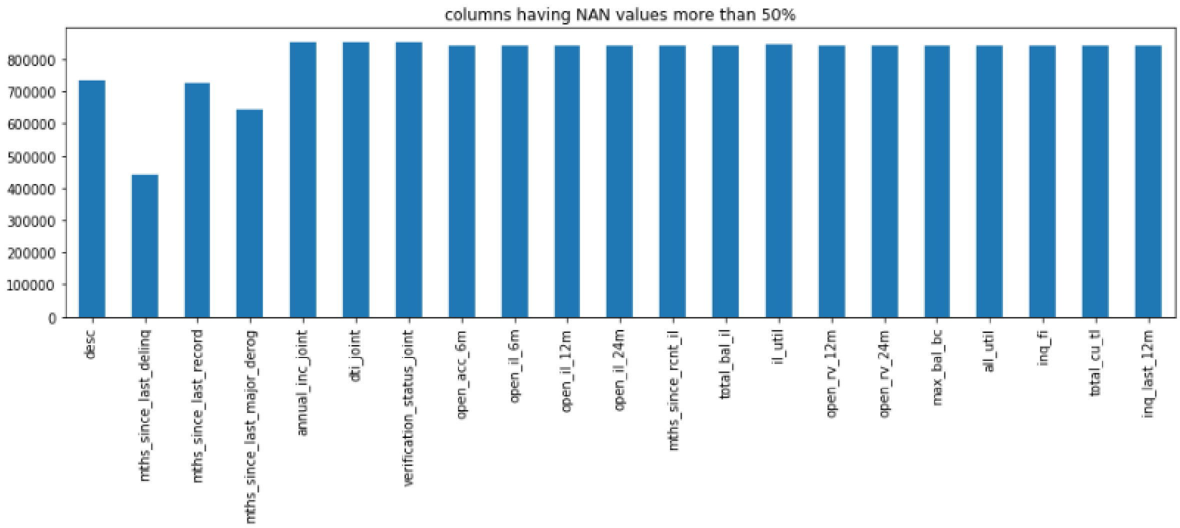
2.1 Phase I – Data Extraction and Cleaning:

Missing Value Analysis and Treatment

- In our dataset our target shows that 94% have not defaulted and 6% are defaulters or charged off. So this is clearly an unbalanced dataset.

- The first issue was to know if the columns were filled with useful information or were mostly empty. Data exploration uncovered many empty or almost empty columns which were removed from the dataset because it would prove a difficult task to go back and try to answer for each data point that did not seem necessary at the time of the loan application.

- Our dataset has 855969 rows × 73 features including the target out of which 32 have missing values or NAN. Below we will look at a plot and get some insights.



<u>Insights</u>:

- So, we can see from the above plot that there are 20+ columns in the dataset where all the values are NA.
- As we can see there are 855969 observations & 73 columns in the dataset, it will be very difficult to look at each column one by one & find the NA or missing values. So let's find out all columns where missing values are more than certain percentage, let's say 50%. We will remove those columns as it is not feasible to impute missing values for those columns.

- Out of 73 observations we only kept 51. We removed about 22 observations that had more than 50% missing values since it will not make any sense in further exploration.

- We need to especially pay close attention to data leakage, which can cause the model to over fit. This is because the model would be also learning from features that wouldn't be available when we're using it make predictions on future loans

- Some irrelevant columns Unique ID's such as "id", "member_id" because they did not provide any useful information about the customer. As last 2 digits of zip code is masked 'xx', we can remove that as well.

- We still had five more features with null values (collections_12_mths_ex_med, revol_util ,tot_coll_amt, tot_cur_bal, total_rev_hi_lim) which we have imputed using fillna method using the appropriate statistic.

Feature Extraction

Decide on a Target Column

- Now, let's decide on the appropriate column to use as a target column for modeling – keep in mind the main goal is predict who will pay off a loan and who will default. We learned from the description of columns in the preview data frame that Default_ind is the only field in the main dataset that describe a loan status, so let's use this column as the target column.

Transformation

- We have transformed emp_length and grade to integer values using transformation technique so that they provide some information to our model.

2.2 Phase II - Feature Engineering

Casting continues variables to numeric:

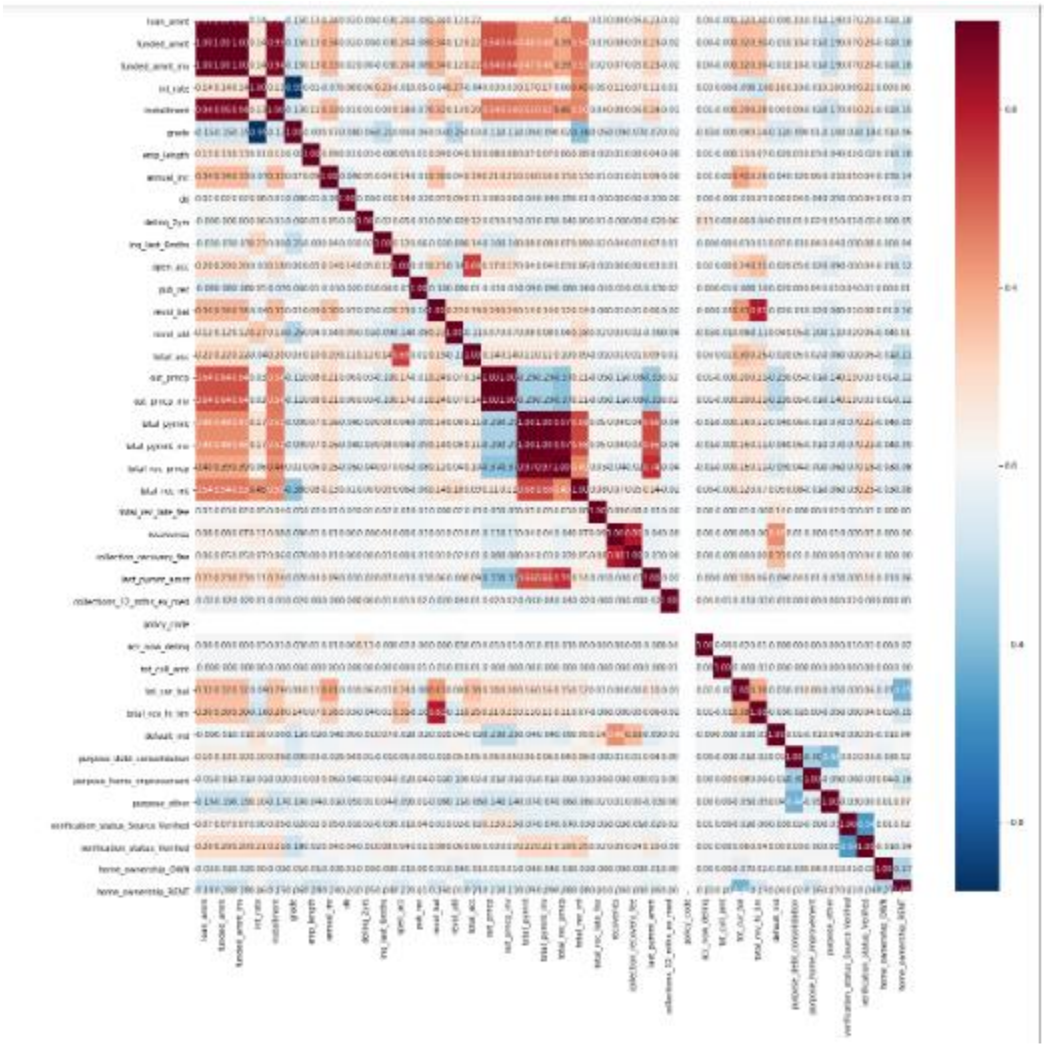
- We have Cast all continues variables that are necessary for our analysis to numeric so that we can find a correlation between them.

Mapping:

- We are mapping the issue date from "Jun-2015" to "Dec-2015" as Test for the ease of splitting our data to test set

Correlation:

- Finding the correlation between variables. We will now look at the correlation structure between our variables that we selected above. This will tell us about any dependencies reduce the dimensionality between different variables and help us a little bit more.



Insights: It is clear from the Heatmap that how 'loan_amnt', 'funded_amnt' & 'funded_amnt_inv' are closely interrelated. So we can take any one column out of them for our analysis. Also , 'total_pymnt', 'total_pymnt_inv' are highly correlated.

Feature Scaling:

We've have scaled the data so that each column has a mean of zero and unit standard deviation. We have scaled the training set and test set as well so as to reproduce the same results.

2.3 Data Dictionary:

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.
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers 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
desc	Loan description provided by the borrower
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.
dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested loan, divided by the co-borrowers' combined self-reported monthly income
earliest_cr_line	The month the borrower's earliest reported credit line was opened
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.
emp_title	The job title supplied by the Borrower when applying for the loan.
funded_amnt	The total amount committed to that loan at that point in time.
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.
id	A unique assigned ID for the loan listing.

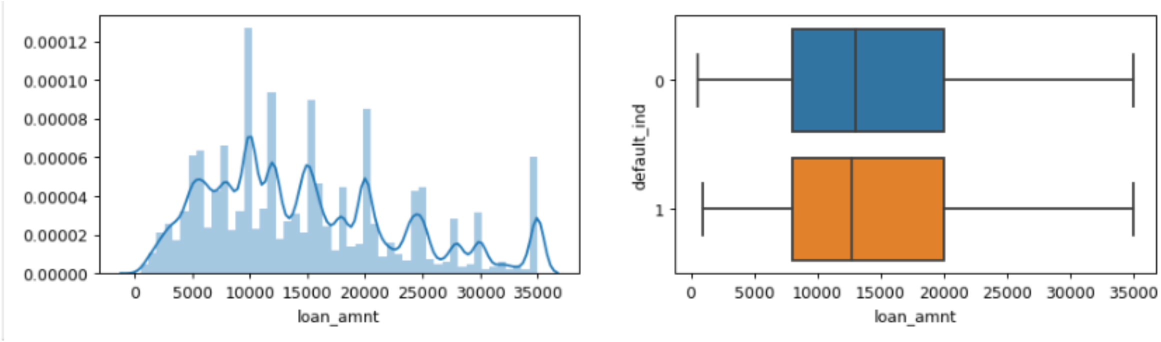
initial_list_status	The initial listing status of the loan. Possible values are – W, F
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
int_rate	Interest Rate on the loan
issue_d	The month which the loan was funded
last_credit_pull_d	The most recent month XYZ corp. pulled credit for this loan
last_pymnt_amnt	Last total payment amount received
last_pymnt_d	Last month payment was received
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
loan status	Current status of the loan
member_id	A unique Id for the borrower member.
mths_since_last_delinq	The number of months since the borrower's last delinquency.
mths_since_last_major_derog	Months since most recent 90-day or worse rating
mths_since_last_record	The number of months since the last public record.
next_pymnt_d	Next scheduled payment date
open_acc	The number of open credit lines in the borrower's credit file.
out_prncp	Remaining outstanding principal for total amount funded
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
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
recoveries	post charge off gross recovery
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.
sub_grade	XYZ assigned assigned loan subgrade
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
title	The loan title provided by the borrower
total_acc	The total number of credit lines currently in the borrower's credit file
total_pymnt	Payments received to date for total amount funded

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_rec_prncp	Principal received to date
verified_status_joint	Indicates if the co-borrowers' joint income was verified by XYZ corp., not verified, or if the income source was verified
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.
open_acc_6m	Number of open trades in last 6 months
open_il_6m	Number of currently active installment trades
open_il_12m	Number of installment accounts opened in past 12 months
open_il_24m	Number of installment accounts opened in past 24 months
mths_since_rcnt_il	Months since most recent installment accounts opened
total_bal_il	Total current balance of all installment accounts
il_util	Ratio of total current balance to high credit/credit limit on all install acct
open_rv_12m	Number of revolving trades opened in past 12 months
open_rv_24m	Number of revolving trades opened in past 24 months
max_bal_bc	Maximum current balance owed on all revolving accounts
all_util	Balance to credit limit on all trades
total_rev_hi_lim	Total revolving high credit/credit limit
inq_fi	Number of personal finance inquiries
total_cu_tl	Number of finance trades
inq_last_12m	Number of credit inquiries in past 12 months
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
tot_coll_amt	Total collection amounts ever owed
tot_cur_bal	Total current balance of all accounts
verification_status	Was the income source verified

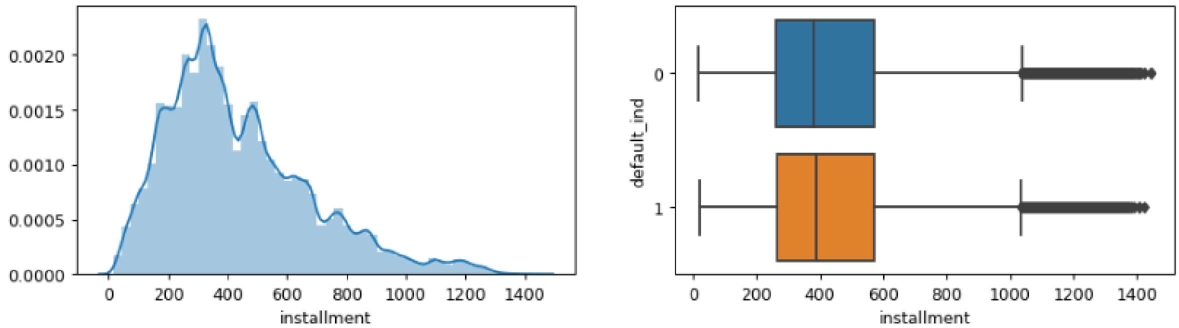
2.4 Exploratory Data Analysis:

Univariate Analysis:

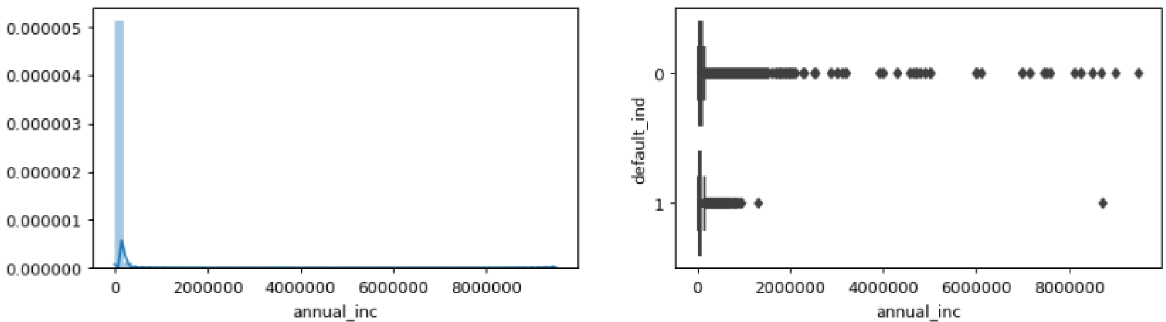
1. **Loan Amount**



2. **Installment**

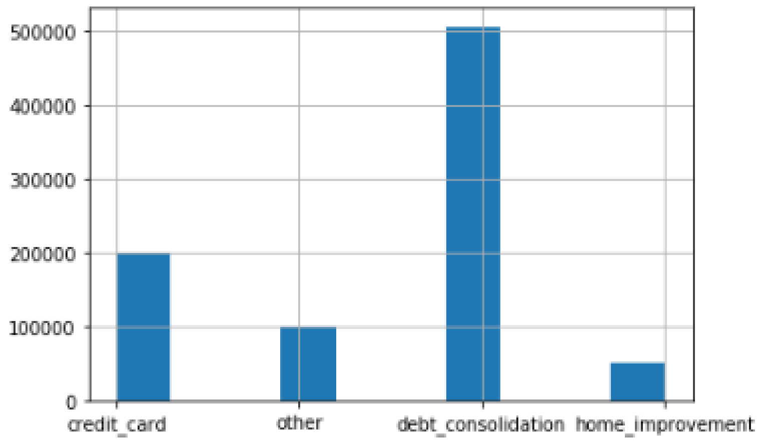


3. **Annual Income**

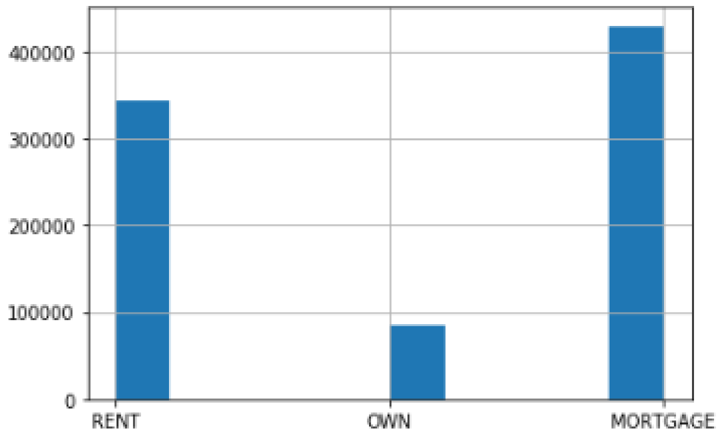


Categorical Variables:

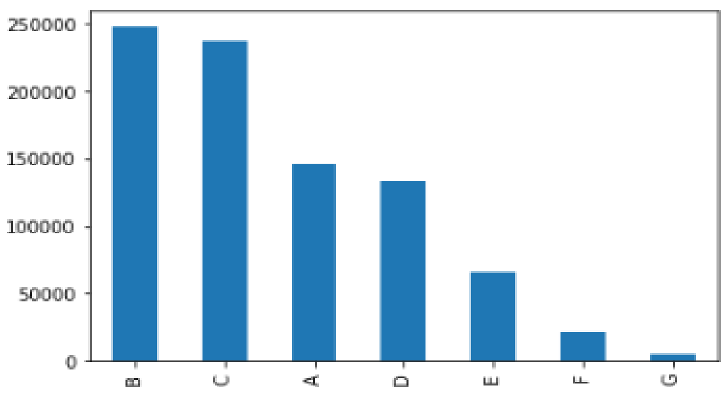
4. Purpose



5. Home Ownership



6. Grade



CHAPTER 3: FITTING MODELS TO DATA

We have used the below Models for our classification:

3.1 Logistic regression:

- Logistic Regression, despite its name, is a Linear Model for classification rather than Regression. Logistic Regression is also known in the literature as Logit Regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a **Logistic Function**.
- This implementation can fit binary, One-vs-Rest, or multinomial logistic regression with optional $\ell 1$, $\ell 2$ or Elastic-Net regularization. Note that regularization is applied by default.

3.2 Random forest Classifier:

- A Random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

3.3 Decision Tree Algorithm:

- Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems. Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data (training data).
- Decision Tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

3.4 KNN

- K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. K-Nearest Neighbours (KNN) algorithm uses 'feature similarity' to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

3.1 Logistic Regression Model

- Our goal is to have a good balance between accuracy and precision.

3.1.1 First Logistic Regression Model:

1) Logistic Regression with solver='saga'

- With this solver we got a precision of 0.78. We are tuning the parameters in the next model to find the best accuracy score and Precision.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	256680
1	0.78	0.68	0.73	311
accuracy			1.00	256991
macro avg	0.89	0.84	0.86	256991
weighted avg	1.00	1.00	1.00	256991

3.1.2 Second Logistic Regression Model:

2) Logistic Regression with solver='lbfgs'

- With this solver we are getting an precision of about 0.84. Below is the report.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	256680
1	0.84	0.80	0.82	311
accuracy			1.00	256991
macro avg	0.92	0.90	0.91	256991
weighted avg	1.00	1.00	1.00	256991

3.1.3 Third Logistic Regression Model:

3) Logistic Regression with solver="liblinear":

- With this solver we are getting good scores. The LIBLINEAR solver is often the best choice. Below is the report.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	256680
1	0.86	0.80	0.83	311
accuracy			1.00	256991
macro avg	0.93	0.90	0.91	256991
weighted avg	1.00	1.00	1.00	256991

Accuracy Score and Confusion Matrix:

```
0.999603098941208
[[256641     39]
 [    63    248]]
```

- Comparing all the above Models looks like the **THIRD** Model has a good Accuracy and Precision score. Since our data is imbalanced we are getting results that have huge variation between the models. Various sampling techniques can be used in order to balance the data and make predictions but since we have limited time we have not applied the sampling techniques.

3.2 Random Forest

- We applied Random Forest on the Training data set to validate if any further improvement of the model can be performed post Logistic regression. Below were the parameters which were applied for Random Forest:

1) Model1 with n_estimators=100:

- In this Model we are getting low accuracy score of 37%. Below is the report.

```
              precision    recall  f1-score   support

           0       1.00      0.37      0.55    256680
           1       0.00      1.00      0.00       311

    accuracy                           0.38    256991
   macro avg       0.50      0.69      0.27    256991
weighted avg       1.00      0.38      0.54    256991
```

```
print(accuracy_score(Y_test,prediction))
```

```
0.37565128739916964
```

2) Model2 with n_estimators=200

In this Model our Accuracy decreased by 1% which is 36%.

```
              precision    recall  f1-score   support

           0       1.00      0.37      0.54    256680
           1       0.00      1.00      0.00       311

    accuracy                           0.37    256991
   macro avg       0.50      0.68      0.27    256991
weighted avg       1.00      0.37      0.54    256991

[[ 94382 162298]
 [     1    310]]
0.3684642652855476
```

3.3 Decision Tree

- In this Model we are getting very low accuracy score of 28%. Below is the report.

```
              precision    recall  f1-score   support

           0       1.00      0.28      0.44    256680
           1       0.00      1.00      0.00       311

    accuracy                           0.28    256991
   macro avg       0.50      0.64      0.22    256991
weighted avg       1.00      0.28      0.44    256991
```

```
print(accuracy_score(Y_test,predictions))
```

```
0.2828425898183205
```

3.4 K-Nearest Neighbors

- In this Model we are getting good accuracy score and decent precision of 61%. Below is the report.

```
0.9990661151557837
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    256680
           1       0.62      0.61      0.61       311

    accuracy                           1.00    256991
   macro avg       0.81      0.80      0.81    256991
weighted avg       1.00      1.00      1.00    256991
```

CHAPTER 4: KEY FINDINGS

- Significant Variables identified in linear models are also used in Random forest
- Below table provides a snapshot of the various models which the business can choose from based on the pros and cons of each model.

S.No	Model	Accuracy	Precision
1	Logistic(Model3)	99%	86%
2	KNN	99%	61%
3	Random Forest	37%	0.19%
4	Decision Tree	28%	0.16%

- The Accuracy Score and Precision suggest that the Logistic Regression is the best model for prediction on this dataset.

CHAPTER 5: RECOMMENDATIONS AND CONCLUSION:

- We have successfully built a Machine Learning Algorithm to predict the probability of defaulters
- Also, we might want to look on other techniques or variables to improve the prediction power of the algorithm. One of the drawbacks is just the limited number of people who defaulted on their loan in the 8 years of data (2007-2015) present on the dataset.
- We can use an updated data frame which consist next 3 years values (2015-2018) and see how many of the current loans were paid off or defaulted or even charged off. Then these new data points can be used for predicting them or even used to train the model again to improve its accuracy.
- Since we had a lot of categorical data, we cannot apply PCA for dimensionality reduction. Because of this, we can try some different type of variable selection method like 'MULTIPLE CORRESPONDENCE ANALYSIS' to reduce the dimensionality and select the most important variables from the columns.

CHAPTER 6: REFERENCES

- www.kaggle.com
- https://www.tutorialspoint.com/machine_learning_with_python/index.htm
- https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/