
TECHNOCOLABS DATA SCIENCE INTERNSHIP

PROJECT REPORT

PROJECT TITLE:

LENDING CLUB'S LOAN APPROVAL OPTIMIZATION BY REDUCING THE DEFAULTER'S PERCENTAGE

AIM:

Performing various analysis on lending club's loan applicant's historical data and identifying the key terms which helps the organisation to reduce their financial risk.

INTRODUCTION:

Lending Club founded with the idea that bringing borrowers and investors together can help everybody succeed. This platform helps borrowers take control of their debt and empowers everyone to reach their financial goals. Generally, lending companies makes profits from the various interest rate which they charge from the customers. Customers with less risk, they charge them low interest rate and customers with high risk they charge them high interest rate. In order to make more profit out of high interest rate, they provide loans to many risky customers who's really not eligible for getting loan and due to this, most of the risky customers will not be repaying their principal loan amount and interest amount properly. sometimes this will make the company go in loss. So, in this project we are going to design a model which will optimize the loan approval by reducing the defaulter customers percentage using historical data of the Lending Club's loan applicants and will be creating the web application for the ease of end user.

OVERVIEW:

- Data Segmentation and Data Cleaning
- Exploratory Data Analysis using python's data visualisation libraries
- Feature Engineering and Feature Selection

- Training the data with various machine learning classification algorithm
- And deploying the best model in cloud

DATASET:

The dataset we have considered is directly downloaded from the Lending Club's website. There are two different kind of a files, one is loan accepted applicants and another one is loan rejected applicants. The data is having the timeline from 2007 to 2018. The accepted dataset is having around 2.26 million of records and 151 Features and Rejected dataset is having around 27.65 million of record and 9 features.

Original format of the dataset: csv

Original dataset link: [dataset](#)

I. ACCEPTED DATASET:

1 acc_now_delinq :

The number of accounts on which the borrower is now delinquent.

2 acc_open_past_24mths :

Number of trades opened in past 24 months.

3 addr_state :

The state provided by the borrower in the loan application

4 all_util :

Balance to credit limit on all trades

5 annual_inc :

The self-reported annual income provided by the borrower during registration.

6 annual_inc_joint :

The combined self-reported annual income provided by the co-borrowers during registration

7 application_type :

Indicates whether the loan is an individual application or a joint application with two co-borrowers

8 avg_cur_bal :

Average current balance of all accounts

9 bc_open_to_buy :

Total open to buy on revolving bankcards.

10 bc_util :

Ratio of total current balance to high credit/credit limit for all bankcard accounts.

11 chargeoff_within_12_mths :

Number of charge-offs within 12 months

12 collection_recovery_fee :

post charge off collection fee

13 collections_12_mths_ex_med :

Number of collections in 12 months excluding medical collections

14 delinq_2yrs :

The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years

15 delinq_amnt :

The past-due amount owed for the accounts on which the borrower is now delinquent.

16 desc :

Loan description provided by the borrower

17 dti :

A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.

18 dti_joint :

A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income

19 earliest_cr_line :

The month the borrower's earliest reported credit line was opened

20 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.

21 emp_title :

The job title supplied by the Borrower when applying for the loan.*

22 fico_range_high :

The upper boundary range the borrower's FICO at loan origination belongs to.

23 fico_range_low :

The lower boundary range the borrower's FICO at loan origination belongs to.

24 funded_amnt :

The total amount committed to that loan at that point in time.

25 funded_amnt_inv :

The total amount committed by investors for that loan at that point in time.

26 grade :

LC assigned loan grade

27 home_ownership :

The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER

28 id :

A unique LC assigned ID for the loan listing.

29 il_util :

Ratio of total current balance to high credit/credit limit on all install acct

30 initial_list_status :

The initial listing status of the loan. Possible values are – W, F

31 inq_fi :

Number of personal finance inquiries

32 inq_last_12m :

Number of credit inquiries in past 12 months

33 inq_last_6mths :

The number of inquiries in past 6 months (excluding auto and mortgage inquiries)

34 installment :

The monthly payment owed by the borrower if the loan originates.

35 int_rate :

Interest Rate on the loan

36 issue_d :

The month which the loan was funded

37 last_credit_pull_d :

The most recent month LC pulled credit for this loan

38 last_fico_range_high :

The upper boundary range the borrower's last FICO pulled belongs to.

39 last_fico_range_low :

The lower boundary range the borrower's last FICO pulled belongs to.

40 last_pymnt_amnt :

Last total payment amount received

41 last_pymnt_d :

Last month payment was received

42 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.

43 loan_status :

Current status of the loan

44 max_bal_bc :

Maximum current balance owed on all revolving accounts

45 member_id :

A unique LC assigned Id for the borrower member.

46 mo_sin_old_il_acct :

Months since oldest bank installment account opened

47 mo_sin_old_rev_tl_op :

Months since oldest revolving account opened

48 mo_sin_rcnt_rev_tl_op :

Months since most recent revolving account opened

49 mo_sin_rcnt_tl :

Months since most recent account opened

50 mort_acc :

Number of mortgage accounts.

51 mths_since_last_delinq :

The number of months since the borrower's last delinquency.

52 mths_since_last_major_derog :

Months since most recent 90-day or worse rating

53 mths_since_last_record :

The number of months since the last public record.

54 mths_since_rcnt_il :

Months since most recent installment accounts opened

55 mths_since_recent_bc :

Months since most recent bankcard account opened.

56 mths_since_recent_bc_dlq :

Months since most recent bankcard delinquency

57 mths_since_recent_inq :

Months since most recent inquiry.

58 mths_since_recent_revol_delinq :

Months since most recent revolving delinquency.

59 next_pymnt_d :

Next scheduled payment date

60 num_accts_ever_120_pd :

Number of accounts ever 120 or more days past due

61 num_actv_bc_tl :

Number of currently active bankcard accounts

62 num_actv_rev_tl :

Number of currently active revolving trades

63 num_bc_sats :

Number of satisfactory bankcard accounts

64 num_bc_tl :

Number of bankcard accounts

65 num_il_tl :

Number of installment accounts

66 num_op_rev_tl :

Number of open revolving accounts

67 num_rev_accts :

Number of revolving accounts

68 num_rev_tl_bal_gt_0 :

Number of revolving trades with balance >0

69 num_sats :

Number of satisfactory accounts

70 num_tl_120dpd_2m :

Number of accounts currently 120 days past due (updated in past 2 months)

71 num_tl_30dpd :

Number of accounts currently 30 days past due (updated in past 2 months)

72 num_tl_90g_dpd_24m :

Number of accounts 90 or more days past due in last 24 months

73 num_tl_op_past_12m :

Number of accounts opened in past 12 months

74 open_acc :

The number of open credit lines in the borrower's credit file.

75 open_acc_6m :

Number of open trades in last 6 months

76 open_il_12m :

Number of installment accounts opened in past 12 months

77 open_il_24m :

Number of installment accounts opened in past 24 months

78 open_act_il :

Number of currently active installment trades

79 open_rv_12m :

Number of revolving trades opened in past 12 months

80 open_rv_24m :

Number of revolving trades opened in past 24 months

81 out_prncp :

Remaining outstanding principal for total amount funded

82 out_prncp_inv :

Remaining outstanding principal for portion of total amount funded by investors

83 pct_tl_nvr_dlq :

Percent of trades never delinquent

84 percent_bc_gt_75 :

Percentage of all bankcard accounts > 75% of limit.

85 policy_code :

publicly available policy_code=1

new products not publicly available policy_code=2

86 pub_rec :

Number of derogatory public records

87 pub_rec_bankruptcies :

Number of public record bankruptcies

88 purpose :

A category provided by the borrower for the loan request.

89 pymnt_plan :

Indicates if a payment plan has been put in place for the loan

90 recoveries :

post charge off gross recovery

91 revol_bal :

Total credit revolving balance

92 revol_util :

Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.

93 sub_grade :

LC assigned loan subgrade

94 tax_liens :

Number of tax liens

95 term :

The number of payments on the loan. Values are in months and can be either 36 or 60.

96 title :

The loan title provided by the borrower

97 tot_coll_amt :

Total collection amounts ever owed

98 tot_cur_bal :

Total current balance of all accounts

99 tot_hi_cred_lim :

Total high credit/credit limit

100 total_acc :

The total number of credit lines currently in the borrower's credit file

101 total_bal_ex_mort :

Total credit balance excluding mortgage

102 total_bal_il :

Total current balance of all installment accounts

103 total_bc_limit :

Total bankcard high credit/credit limit

104 total_cu_tl :

Number of finance trades

105 total_il_high_credit_limit :

Total installment high credit/credit limit

106 total_pymnt :

Payments received to date for total amount funded

107 total_pymnt_inv :

Payments received to date for portion of total amount funded by investors

108 total_rec_int :

Interest received to date

109 total_rec_late_fee :

Late fees received to date

110 total_rec_prncp :

Principal received to date

111 total_rev_hi_lim :

Total revolving high credit/credit limit

112 url :

URL for the LC page with listing data.

113 verification_status :

Indicates if income was verified by LC, not verified, or if the income source was verified

114 verified_status_joint :

Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified

115 zip_code :

The first 3 numbers of the zip code provided by the borrower in the loan application.

116 revol_bal_joint :

Sum of revolving credit balance of the co-borrowers, net of duplicate balances

117 sec_app_fico_range_low :

FICO range (high) for the secondary applicant

118 sec_app_fico_range_high :

FICO range (low) for the secondary applicant

119 sec_app_earliest_cr_line :

Earliest credit line at time of application for the secondary applicant

120 sec_app_inq_last_6mths :

Credit inquiries in the last 6 months at time of application for the secondary applicant

121 sec_app_mort_acc :

Number of mortgage accounts at time of application for the secondary applicant

122 sec_app_open_acc :

Number of open trades at time of application for the secondary applicant

123 sec_app_revol_util :

Ratio of total current balance to high credit/credit limit for all revolving accounts

124 sec_app_open_act_il :

Number of currently active installment trades at time of application for the secondary applicant

125 sec_app_num_rev_accts :

Number of revolving accounts at time of application for the secondary applicant

126 sec_app_chargeoff_within_12_mths :

Number of charge-offs within last 12 months at time of application for the secondary applicant

127 sec_app_collections_12_mths_ex_med :

Number of collections within last 12 months excluding medical collections at time of application for the secondary applicant

128 sec_app_mths_since_last_major_derog :

Months since most recent 90-day or worse rating at time of application for the secondary applicant

129 hardship_flag :

Flags whether or not the borrower is on a hardship plan

130 hardship_type :

Describes the hardship plan offering

131 hardship_reason :

Describes the reason the hardship plan was offered

132 hardship_status :

Describes if the hardship plan is active, pending, canceled, completed, or broken

133 deferral_term :

Amount of months that the borrower is expected to pay less than the contractual monthly payment amount due to a hardship plan

134 hardship_amount :

The interest payment that the borrower has committed to make each month while they are on a hardship plan

135 hardship_start_date :

The start date of the hardship plan period

136 hardship_end_date :

The end date of the hardship plan period

137 payment_plan_start_date :

The day the first hardship plan payment is due. For example, if a borrower has a hardship plan period of 3 months, the start date is the start of the three-month period in which the borrower is allowed to make interest-only payments.

138 hardship_length :

The number of months the borrower will make smaller payments than normally obligated due to a hardship plan

139 hardship_dpd :

Account days past due as of the hardship plan start date

140 hardship_loan_status :

Loan Status as of the hardship plan start date

141 orig_projected_additional_accrued_interest :

The original projected additional interest amount that will accrue for the given hardship payment plan as of the Hardship Start Date. This field will be null if the borrower has broken their hardship payment plan.

142 hardship_payoff_balance_amount :

The payoff balance amount as of the hardship plan start date

143 hardship_last_payment_amount :

The last payment amount as of the hardship plan start date

144 disbursement_method :

The method by which the borrower receives their loan. Possible values are: CASH, DIRECT_PAY

145 debt_settlement_flag :

Flags whether or not the borrower, who has charged-off, is working with a debt-settlement company.

146 debt_settlement_flag_date :

The most recent date that the Debt_Settlement_Flag has been set

147 settlement_status :

The status of the borrower's settlement plan. Possible values are: COMPLETE, ACTIVE, BROKEN, CANCELLED, DENIED, DRAFT

148 settlement_date :

The date that the borrower agrees to the settlement plan

149 settlement_amount :

The loan amount that the borrower has agreed to settle for

150 settlement_percentage :

The settlement amount as a percentage of the payoff balance amount on the loan

151 settlement_term :

The number of months that the borrower will be on the settlement plan

152 nan :

nan

153 nan :

* Employer Title replaces Employer Name for all loans listed after 9/23/2013

II. REJECTED DATASET:

1 Amount Requested :

The total amount requested by the borrower

2 Application Date :

The date which the borrower applied

3 Loan Title :

The loan title provided by the borrower

4 Risk_Score :

For applications prior to November 5, 2013 the risk score is the borrower's FICO score. For applications after November 5, 2013 the risk score is the borrower's Vantage score.

5 Debt-To-Income Ratio :

A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.

6 Zip Code :

The first 3 numbers of the zip code provided by the borrower in the loan application.

7 State :

The state provided by the borrower in the loan application

8 Employment 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.

9 Policy Code :

publicly available policy_code = 1

new products not publicly available policy_code = 2

DATA SEGMENTATION AND DATA CLEANING:

(i) The accepted dataset contains 2.26 Million records, so we have dropped the features which contains more than 25% of missing values.

(ii) Since the dataset is very huge, most of the features which will not help our analysis has been dropped and some important features which will help our analysis has given below.

➤ From Accepted Dataset:

Loan Status, Loan Amount, Interest Rate, Annual Income, Debt-to-Income (DTI), Instalment, Fico Score (Risk Score), Delinquent Amount, Employment Title, Term, Grade, Sub-grade, Home Ownership.

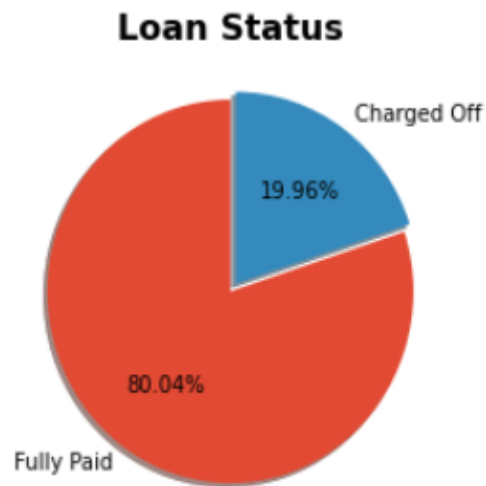
➤ From Rejected Dataset:

Amount Requested, Debt-to-Income Ratio, Fico Score (Risk Score)

(iii) Outliers detected and removed from the above features

(iv) Dependent variable:

Loan Status is the dependent variable (Target Variable), In that apart from Fully paid and Charged off customers, all other categories records have dropped. So, it is a binary classification problem.

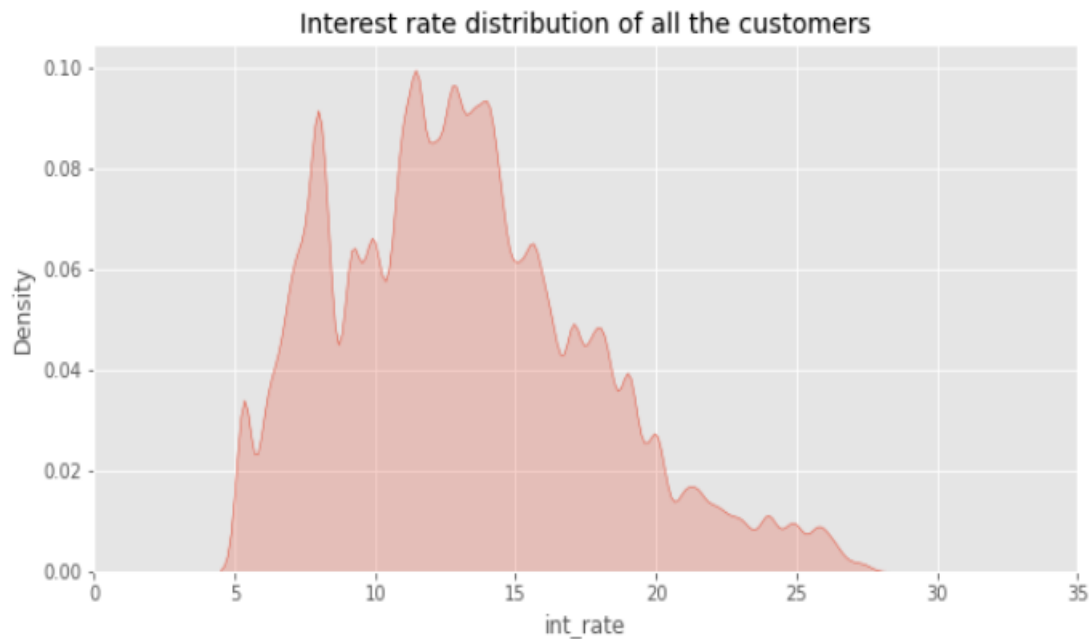


EXPLORATORY DATA ANALYSIS:

1. Interest Rate:

General loan principal is a good trust worthy customer will get loan easily with less interest rate for his credibility (Also his FICO Score, debt to income ratio both are well maintained in an appropriate level). Then a customer with the less Risk score and high DTI ratio are always risky customers and they always get charged for more interest rate.

Interest rate ranging from 5.31% to 27.79%. beyond this limit is considered as an outlier (based on Z-Score) and also these high interests will cause financial risk to Lending Club.



2) ANNUAL INCOME:

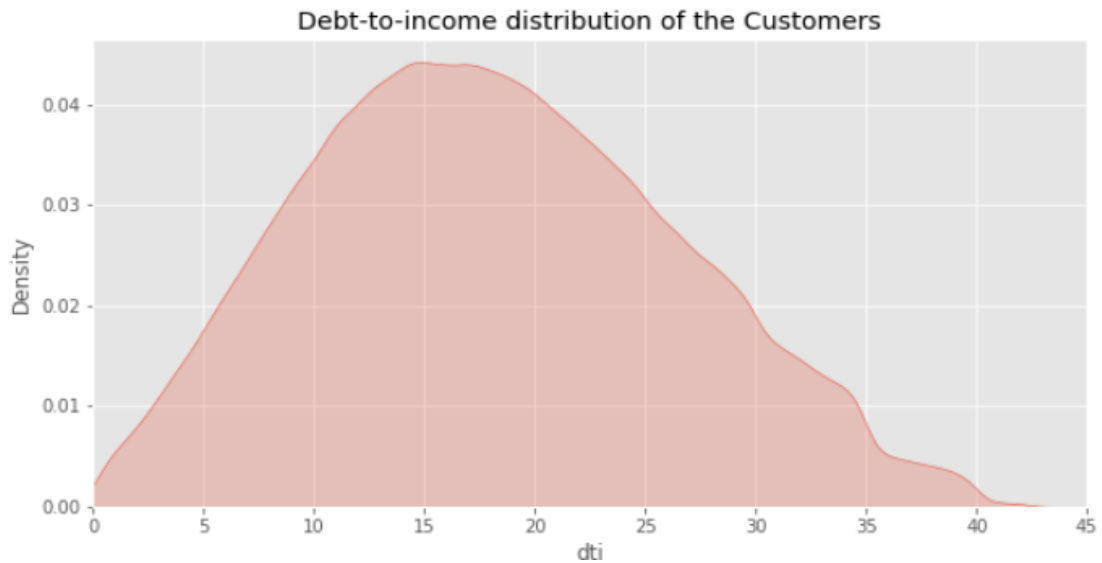
Based on annual income summary, the mean income of the US employees is 65000 Dollars and 25% and 75% is respectively 45760 and 90000 Dollars. Based on Z-Score beyond 286800 is considered as a Outlier and removed.

3) Debt-to-Income (DTI) Ratio:

The debt-to-income (DTI) ratio is the percentage of your gross monthly income that goes to paying your monthly debt payments and is used by lenders to determine your borrowing risk.

DTI= (Total of Monthly Debt Payments / Gross Monthly Income)

Generally, Debt to income ratio lower than 18% Considered Excellent. 18% to 35% is considered very good percentage. most of the customers are Defaulters when considering DTI ratio is more than 42.36.

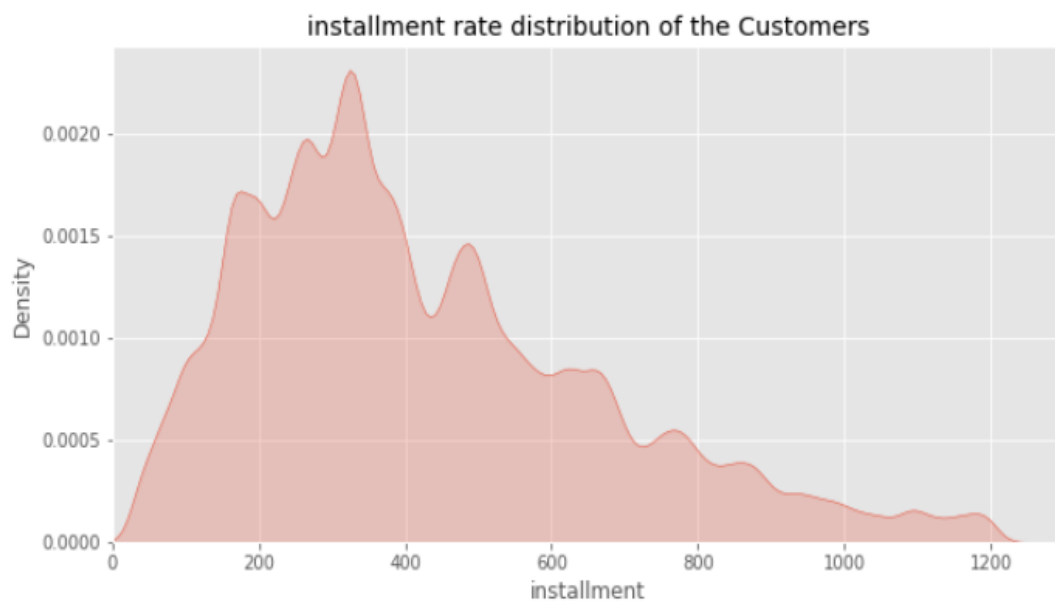


4) INSTALMENT:

An instalment loan is a type of agreement or contract involving a loan that is repaid over time with a set number of scheduled payments.

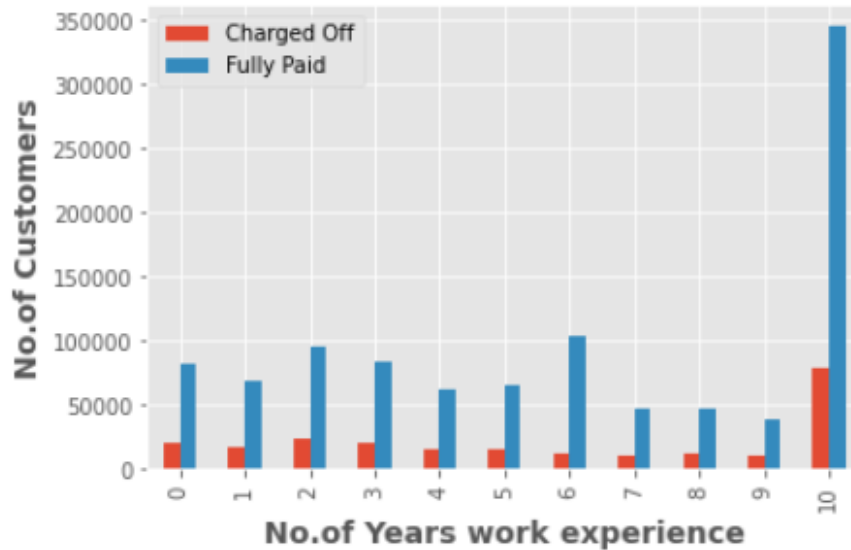
So, customers who pay more than 1206.41 Dollars as monthly instalment, finding difficulty in re-paying loan and causes company loss.

Mean value of the installment : 424.21



5) EMPLOYMENT LENGTH:

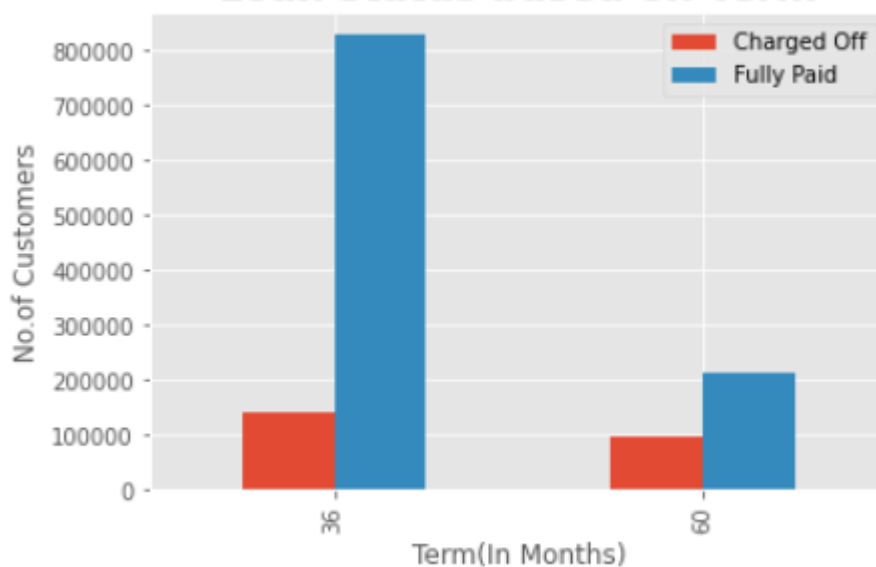
Loan status based on customer's work experience



6) TERM:

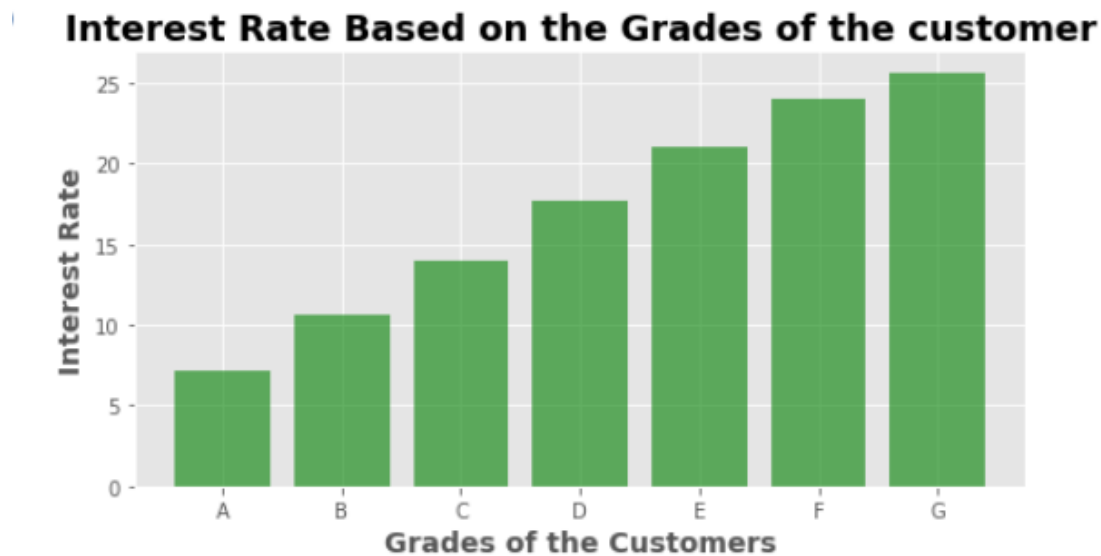
A loan term is the length of time it will take for a loan to be completely paid off when the borrower is making regular payments. The time it takes to eliminate the debt is a loan's term.

Loan status based on Term



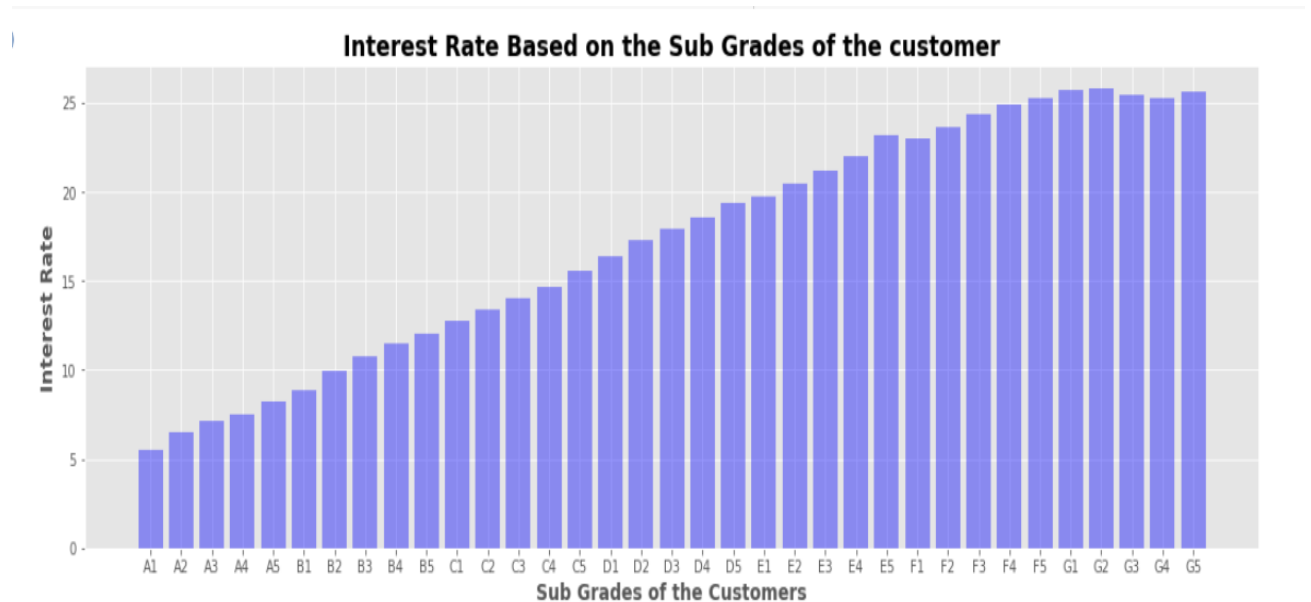
7) GRADE:

Loan grading is used to assign a quality score to a loan based on the credit history of the borrower, quality of the collateral, and the likelihood of the repayment. It is the result of a formula that takes into account not only credit score, but also a combination of several indicators of credit risk from the credit report.



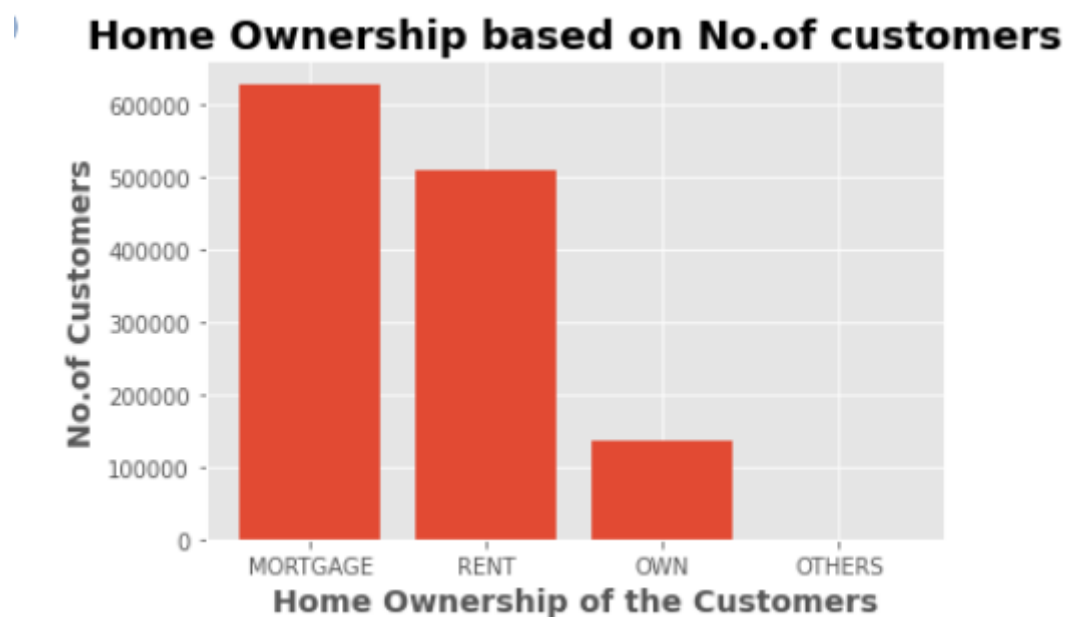
8) SUB-GRADE:

Customers Grades are further classified into Subgrades in order to classify them customers in better way.

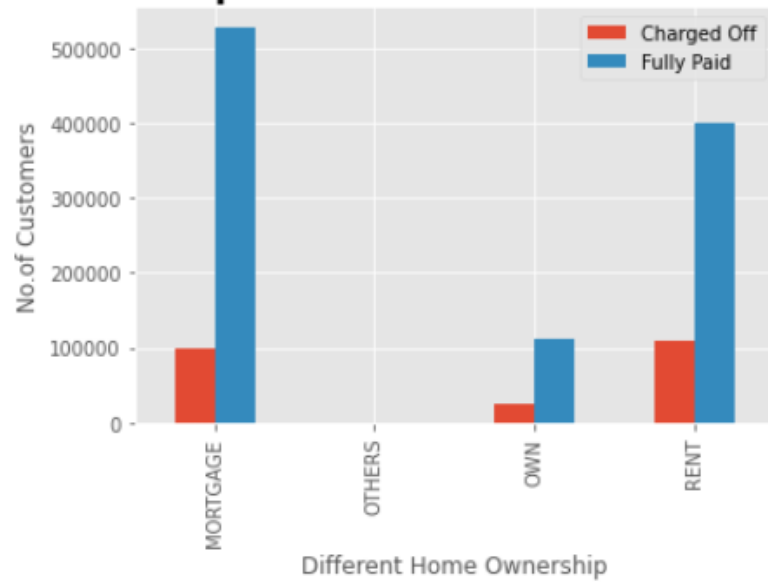


9) HOME OWNERSHIP:

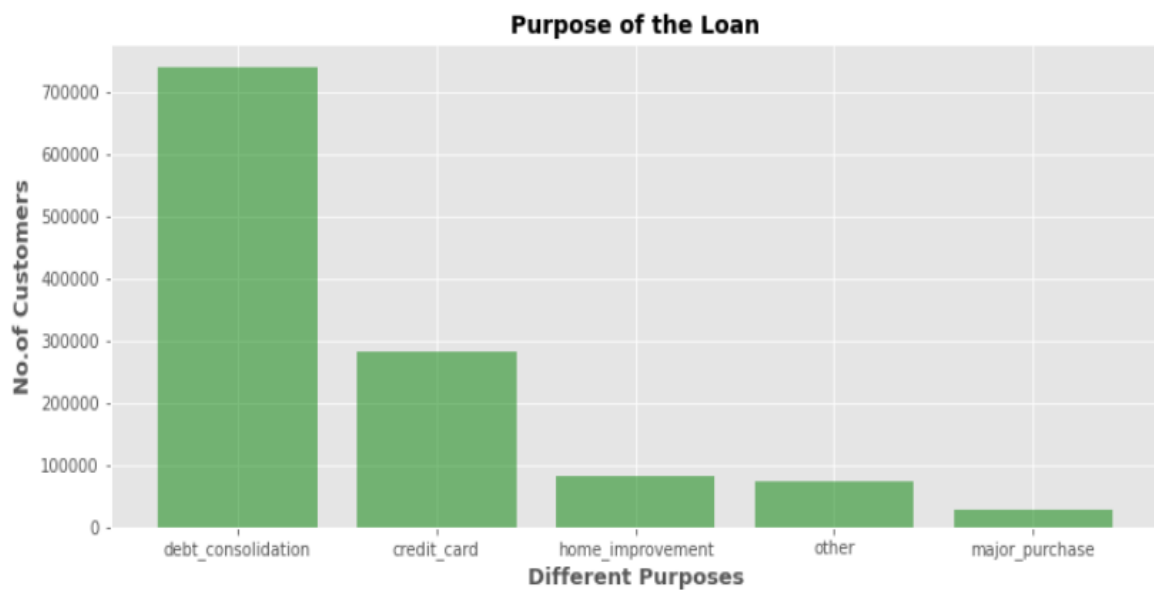
The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER



Home Ownership of the Customers based on loan status



10) PURPOSE OF THE LOAN:



KEY FEATURES RANGE FOR REDUCING FINANCIAL RISK OF LENDING CLUB IN FUTURE:

Loan Amount	\$1000 to \$40000
Interest Rate	5.31% - 27.79%
Annual Income	Up to \$286800
Debt-To-Income (DTI) Ratio	Up to 42.36
Fico Score (Risk Score)	627 to 847.5
Instalment	Not More than 1206.41
Public Derogatory	Note More than 3

DATA MODELING:

So, after the exploratory data analysis we have combined both accepted and rejected datasets on their respective columns. Here, we have used the Machine learning algorithms on the combined dataset to build a classification model that will generate the output that landing club should provide the loan to the borrower or not. In this step we have divided the data into train, validation and test as 80%,10%,10% respectively. In this process we have used many algorithms and applied some hyperparameter tuning so that our algorithms can do better.

The algorithms which we have tried are:

1. Decision tree
2. Random forest
3. Naïve bayes
4. Neural networks
5. Logistic regression
6. KNN (k-nearest neighbors)

In this project our aim is to try out all these algorithms and will proceed with the best one whichever having good accuracy score.

DECISION TREE

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node. It can be used for any of two regression or classification problem.

Tuned Hyperparameters:

min_sample_split: *int or float, default=2*

- The minimum number of samples required to split an internal node: After hyperparameter tuning selected value is '11'.

max_features: *int, float or {"auto", "sqrt", "log2"}, default=None*

- The number of features to consider when looking for the best split:
Selected: $\text{Max_features} = \log_2(\text{num_features})$

max_depth: *int, default=None*

- The maximum depth of the tree. After tuning we have selected value '6'.

criterion: *{"gini", "entropy"}, default = "gini"*

- The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain. Here we have chosen the 'entropy'.

Accuracy of the model:

```
from sklearn.tree import DecisionTreeClassifier

Model_dt = DecisionTreeClassifier(min_samples_split = 11, max_features = 'log2', max_depth = 11, criterion = 'entropy')
Model_dt.fit(X_train, y_train)
print("Accuracy score of training data {} %".format(Model_dt.score(X_train, y_train)*100))
print("Accuracy score of testing data {} %".format(Model_dt.score(X_test, y_test)*100))
```

```
Accuracy score of training data 88.80267318922908 %:
Accuracy score of testing data 88.80215745775058 %
```

The accuracy obtained from Decision tree model is 88.802%

RANDOM FOREST

Random forest is a supervised machine learning algorithm. The forest it builds is an ensemble of decision trees. So here this algorithm builds many decision trees and merge them together to get more accurate result. This algorithm searches for the best feature to split the data on each node.

Tuned Hyperparameters:

n_estimators: *int, default = 100*

- This parameter represents the number of trees should be build in the forest. But after hyperparameter tuning the value for this parameter we have found is 500.

max_depth: *int, default = None*

- The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

max_features: *int, default = 'auto'*

- The maximum features to consider at each split. Here we have used '*None*'. So, to consider all the features during split.

Bootstrap: *bool, default=True*

- Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree. Here, we have set this to '*True*'.

Accuracy of the model:

```
[ ] from sklearn.ensemble import RandomForestClassifier

Model_RF = RandomForestClassifier(n_estimators=500, max_features=None, max_depth=6, bootstrap=True)
Model_RF.fit(X_train, y_train)
print("Accuracy score of training data {} %:".format(Model_RF.score(X_train, y_train)*100))
print("Accuracy score of testing data {} %".format(Model_RF.score(X_test, y_test)*100))

Accuracy score of training data 88.80056252276505 %:
Accuracy score of testing data 88.80577573438954 %
```

The accuracy obtained from Random forest model is 88.805%

DATA NORMALIZATION

Here before going on the Naïve bayes and Neural networks we have normalised the dataset so the features which are having larger value can't able to dominate in the process of prediction and normalization is also used to overcome the computation load for the algorithms. So, for the normalization purpose we have used the Standardization technique because it doesn't affect the outliers much. Standardization is a scaling technique where the values are centred around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. standardization does not have a bounding range. So, even if data have outliers, they will not be affected by standardization.

NAIVE BAYES

Naïve Bayes is a probabilistic machine learning algorithm based on the Bayes theorem, used in a wide variety of classification tasks. The crux of the classifier is based on the Bayes theorem.

Bayes theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Tuned Hyperparameters:

Priors: *array-like of shape (n_classes)*

- Prior probabilities of the classes. If specified the priors are not adjusted according to the data.

var_smoothing: *float, default=1e-9*

- Portion of the largest variance of all features that is added to variances for calculation stability.
- {'var_smoothing': 0.1} turned out to be the best hyperparameters.

Accuracy of the model:

```
from sklearn.metrics import accuracy_score

#df_test_scaled = std.transform(X_test_scaled)
#y_prediction = model.predict_classes(X_test_scaled)
print("\n\nThe Train Accuracy of the model is: {} %".format(gs_NB.score(X_trainval_scaled, y_trainval) * 100.))
print("\n\nThe Test Accuracy of the model is: {} %".format(gs_NB.score(X_test_scaled, y_test) * 100.))
```

The Train Accuracy of the model is: 81.28381589199178 %

The Test Accuracy of the model is: 81.27155889830713 %

The accuracy obtained from Naïve bayes model is 81.27%

NEURAL NETWORK

Neural network is based on a connected units or nodes. A neuron in a neural network is a mathematical function which collects the data or output from previous layer and classifies the data and give output. This algorithm mimics the operation of the human brain to recognize the patterns in the data.

Hyperband method for hyperparameter tuning:

It is based on the idea that when the hyperparameters give us poor results, we can quickly spot it, so it makes no sense to continue training.

Here, implementation of Hyperband trains multiple models for a small number of epochs (*hyperband_iterations* = 2). After that, it picks the best performing models and continues training them for a few more epochs (*max_epochs* = 5). The cycle of picking the best models and training them a little bit more continues until we get the, best model.

Tunned Hyperparameters:

1. How many number of hidden layers Neural network should have?
2. How many number of neurons model should have in each hidden layer?
3. Learning rate

1. '*number_of_layers*': 6,
2. '*unit_0*': 34,
3. '*unit_1*': 34,

4. *'unit_2': 66,*
5. *'unit_3': 2,*
6. *'unit_4': 2,*
7. *'unit_5': 66,*
8. *'learning_rate': 0.01,*
9. *'tuner/epochs': 5*


'number_of_layers': 6 number of layers plus two extra layers. One is with linear activation function and second is output layer.

Accuracy of the model:

```
from sklearn.metrics import accuracy_score

'''standardize the test data before prediction
and predict the result.'''

df_test_scaled = std.transform(X_test)
y_prediction = model.predict_classes(df_test_scaled)
print("\n\nThe Test Accuracy of the model is: {} %".format(accuracy_score(y_test, y_prediction) * 100.))
```

 /usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential.py:450: UserWarning: `model.predict_classes()` is deprecated and will be removed in a future version. Please use `model.predict` instead.

The Test Accuracy of the model is: 88.77900048726126 %

The accuracy obtained from Neural network is 88.779%

LOGISTIC REGRESSION

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.

In regression analysis, **logistic regression** is estimating the parameters of a logistic model (a form of binary regression).

Logistic regression does not really have any critical hyperparameters to tune.

Solver: Algorithm to use in the optimization problem.

➤ **solver** = lbfgs

penalty: Used to specify the norm used in the penalization

➤ **penalty = l2**

The C parameter controls the penalty strength, which can also be effective.

➤ **C = 0.01**

n_jobs: Number of CPU cores used when parallelizing over classes. -1 means using all processors

➤ **n_jobs = -1**

Accuracy of the model:

```
cv = StratifiedKFold(n_splits=10, random_state=1)
logistic_R_rs = LogisticRegression(C= 0.01, penalty= 'l2', solver='lbfgs', n_jobs=-1)
logistic_R_rs.fit(X_train_robust, y_train)
print("Accuracy score of training data {} %".format(logistic_R_rs.score(X_train_robust, y_train)*100))
print("Accuracy score of testing data {} %".format(logistic_R_rs.score(X_test_robust, y_test)*100))
```

/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_split.py:296: FutureWarning: Setting a random_state has no effect since shuffle is on by default. Please set shuffle=False to disable this warning.
FutureWarning
Accuracy score of training data 85.7356335979506 %:
Accuracy score of testing data 85.785720833072 %

The accuracy obtained from Logistic Regression model is 85.73%

KNN (k-nearest neighbors)

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

Tuned Hyperparameters:

n_neighbors: *int, default=5*

➤ Number of neighbors to use by default for k-neighbors queries.

Accuracy of the model:

```
knn = KNeighborsClassifier(n_neighbors = 22)
knn.fit(X_train_stand, y_train)
y_pred_stand = knn.predict(X_test_stand)
print("train score:", knn.score(X_train_stand, y_train))
print("test score:", knn.score(X_test_stand, y_test))
print("Accuracy score :", accuracy_score(y_test, y_pred_stand))
```

train score: 0.8824726638540529
test score: 0.879161621180909
Accuracy score : 0.879161621180909

The accuracy obtained from k-nearest neighbour model is 87.91%
So, after applying all the three algorithms on this dataset we have found that 'Random forest' is having highest accuracy score. Which is 88.80%.

- **DEPLOYMENT AND WEB APPLICATION:**

What is mean by model deployment?

Deploying a **machine learning model**, known as **model deployment**, simply means to integrate a **machine learning model** and integrate it into an existing production environment (1) where it can take in an input and return an output.

The Lending Club's Loan Approval Optimisation Machine Learning Model is trained and tested using Random Forest Algorithm with 88.80% accuracy. And using flask framework, HTML, CSS and python all the necessary files have created along with Procfile and requirement.txt.

Using Heroku, Platform as a service we have successfully deployed our model in the online web server and provided ease to end users.

Deployed Online url: <https://lending-club-web-api.herokuapp.com/>

Using Flask Framework Deployed Web App picture:



The screenshot displays a web application titled "Lending Club's Loan Approval Prediction". The interface features a dark blue background with white text. There are four input fields stacked vertically, each with a light blue border and placeholder text: "Amount Requested", "Risk Score", "Debt-To-Income Ratio", and "Employment Length". Below these fields is a prominent blue button with the text "Predict" in white.

We have deployed web app using Streamlit and results are below:

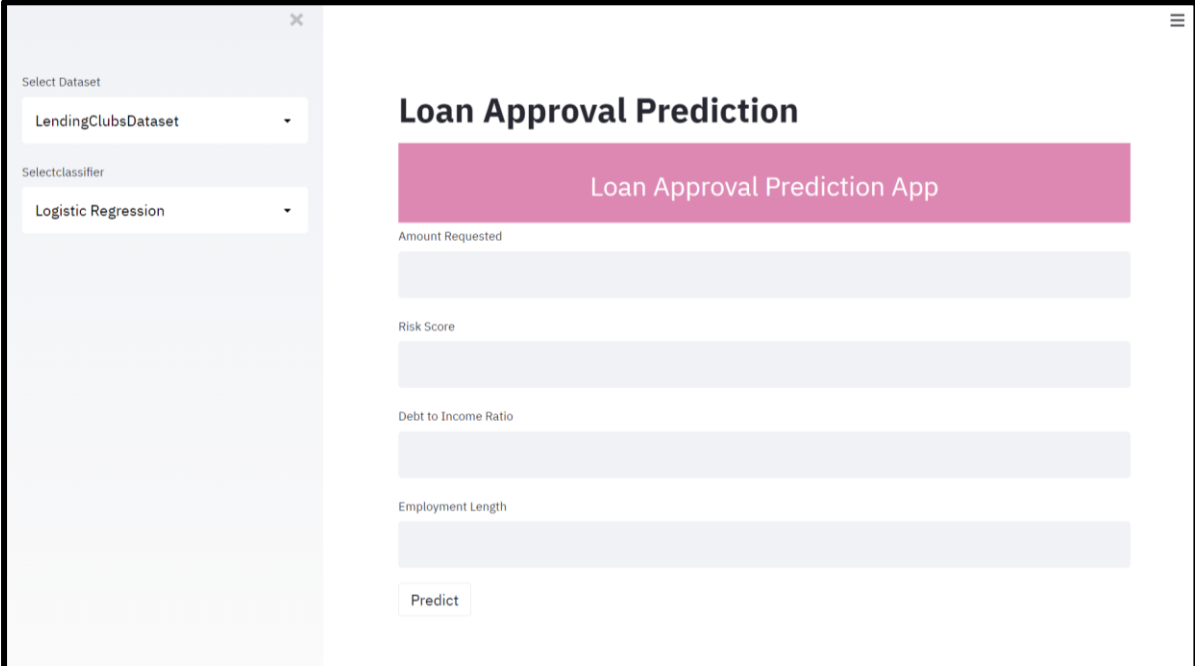
A few of the advantages of using Streamlit tools like Dash and Flask: It embraces Python scripting; No HTML knowledge is needed! Less code is needed to create a beautiful application.

Therefore, we have deployed a similar web app on Heroku using Streamlit framework where we have used the considered features like, 'Amount Requested', 'Risk Score', 'Debt to Income Ratio' and 'Employment length'.

In this web app along with various features we also have a side bar which contains 2 Drop Down lists in which the first allows us to choose the Dataset and the second one allows us to choose the Machine Learning Algorithms through which we want the Output.

Deployed Online url: <https://predictingloan.herokuapp.com/>

Using Streamlit Framework Deployed Web App picture:



The screenshot displays a web application titled "Loan Approval Prediction". On the left, there is a sidebar with two dropdown menus: "Select Dataset" (currently set to "LendingClubsDataset") and "Select classifier" (currently set to "Logistic Regression"). The main content area features a pink header bar with the text "Loan Approval Prediction App". Below this, there are four input fields labeled "Amount Requested", "Risk Score", "Debt to Income Ratio", and "Employment Length". At the bottom of the main area is a "Predict" button. The interface is clean and modern, with a light gray background and a dark gray sidebar.

TEAM MEMBERS:

Sampathkumar S

Muppa Chinmai Ram Naga Prasad

Lakshita sehgal

S Manasa

Debhasih shah

Rekha

Abdul Wasi Lone

Suneha Ghosh

Mukund Sojitra

Priyanshi Gupta

Anshal Aggarwal

Riyansh Buktar

Team Leader name:

Sampathkumar S

Rekha

Mukund Sojitra

Coordinator name: *Mr. Yasin shah*