Case Study Report: CCB Risk Modelling

JPMC CCB Risk Modelling Mentorship | Unnati Singh

Problem Statement:

For a financial organization, accurately assessing the risk profile of loan applicants is crucial to avoid losses from unworthy customers and opportunity costs from declining worthy ones. This assessment must be unbiased, adhering to the Fair Housing Act (FHA) and the Equal Credit Opportunity Act (ECOA), which prohibit discriminatory lending practices based on race, religion, national origin, gender, age, etc. A dataset of 42,000 loan accounts from 2007 to 2011, containing demographic, credit bureau attributes, and loan statuses, is available for analysis. The goal is to establish screening criteria based on relevant qualitative and quantitative customer attributes to approve or decline future applications, ensuring fair lending practices. The decision-making process must also be evaluated for potential unintentional bias to prevent discrimination against sensitive attributes.

Data Overview:

Dataset Description:

The given dataset consists of data of 42,534 customers who applied for loans in between 2007 to 2011. It includes various demographic and credit bureau attributes along with the current loan status of each account. The dataset helps in understanding the risk profile of customers and ensuring fair lending practices.

Variable Description:

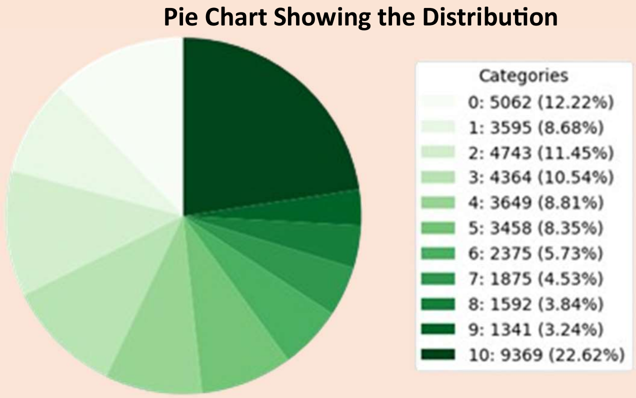
The dataset contains the following variables:

NAME	DESCRIPTION
id	A unique identifier for each loan
member_id	A unique identifier for each borrower
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.
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.
term	Term of the loan. Values are in months and can be either 36 or 60.
installment	The monthly payment owed by the borrower if the loan originates.
emp_title	The job title supplied by the Borrower when applying for the loan.
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.

home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
annual_inc	The self-reported annual income provided by the borrower during registration.
verification_status	Indicates if income was verified, not verified, or if the income source was verified
issue_d	The month which the loan was funded
loan_status	Current status of the loan
pymnt_plan	Indicates if a payment plan has been put in place for the loan
desc	Loan description provided by the borrower
purpose	A category provided by the borrower for the loan request.
title	The loan title provided by the borrower
addr_state	The state provided by the borrower in the loan application
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.
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
mths_since_last_delinq	The number of months since the borrower's last delinquency.
mths_since_last_record	The number of months since the last public record.
open_acc	The number of open credit lines in the borrower's credit file.
pub_rec	Number of derogatory public records
revol_bal	Total credit revolving balance
total_acc	The total number of credit lines currently 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
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_prncp	Principal received to date
total_rec_int	Interest received to date
total_rec_late_fee	Late fees received to date
recoveries	post charge off gross recovery
collection_recovery_fee	post charge off collection fee
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
pub_rec_bankruptcies	Number of public record bankruptcies
interest_rate	Interest Rate on the loan
revol_utilization	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
number_bc_gt_75	number of bankcard accounts > 75% of limit.
fico_score	Origination Fico score or the fico score at the point of loan origination
lti	Loan value to income ratio
month_since_oldest_tl	Month since the oldest trade line for the borrower
race_name	Race of the borrower
gender	Gender of the borrower

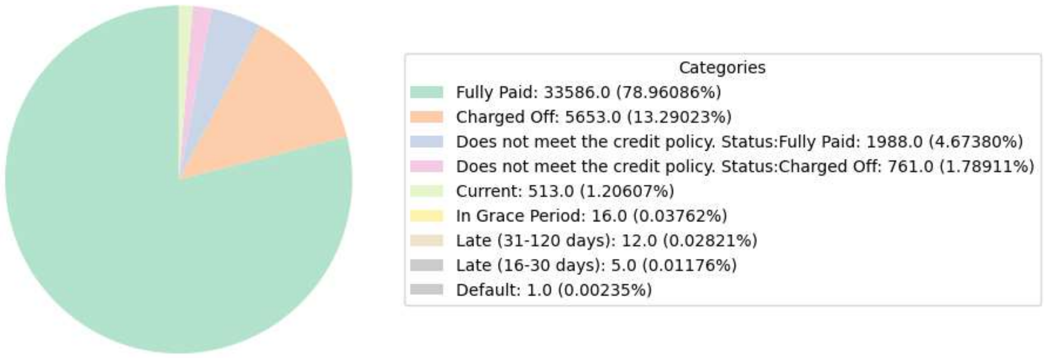
Data Statistics:

Summary of important continuous variables:

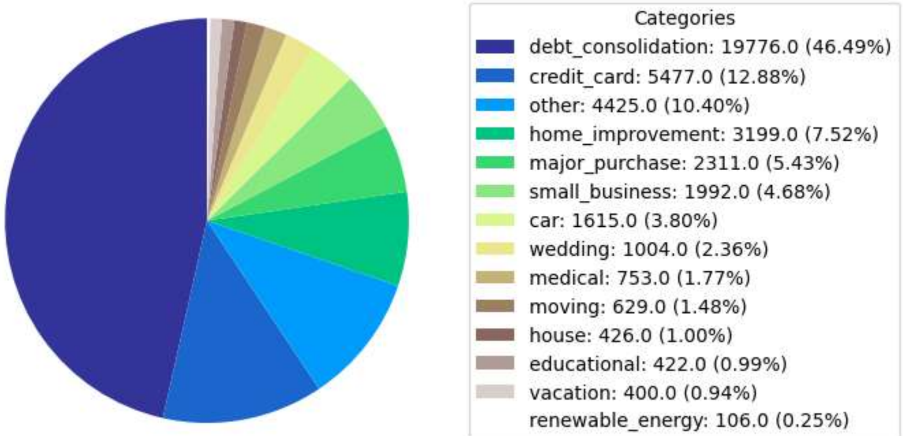
Loan Amount	• Mean: 11,089.72 • Median: 9,700 • Mode: 10,000 • Maximum: 35,000 • Minimum: 500
Employment Length	• Mean: 4.92 • Median: 4.0 • Mode: 10.0 • Maximum: 10.0 • Minimum: 0.0 
Annual Income	• Mean: 69136.56 • Median: 59000 • Mode: 60000 • Maximum: 6000000 • Minimum: 1896
FICO Score	• Mean: 723.07 • Median: 719 • Mode: 704 • Maximum: 829 • Minimum: 619
LTI	• Mean: 0.1861 • Median: 0.1628 • Mode: 0.2 • Maximum: 1.3375 • Minimum: 0.000789
DTI	• Mean: 13.37 • Median: 13.47 • Mode: 0.0 • Maximum: 29.99 • Minimum: 0.0

Summary of important categorical variables:

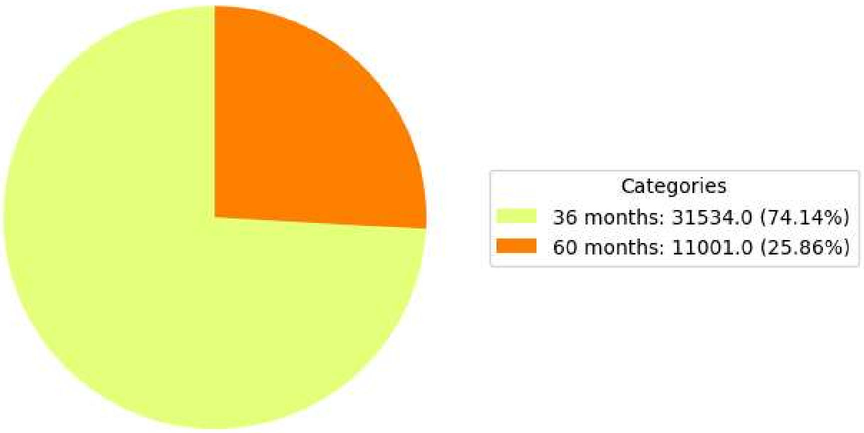
Pie Chart showing the distribution of loan status in the given dataset

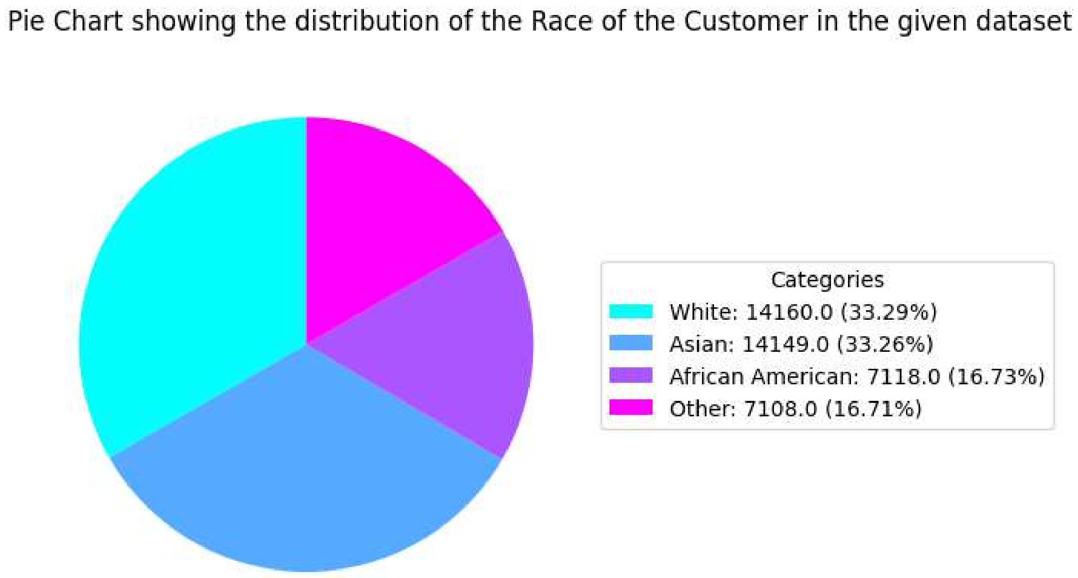


Pie Chart showing the distribution of the purpose of the loan taken in the given dataset

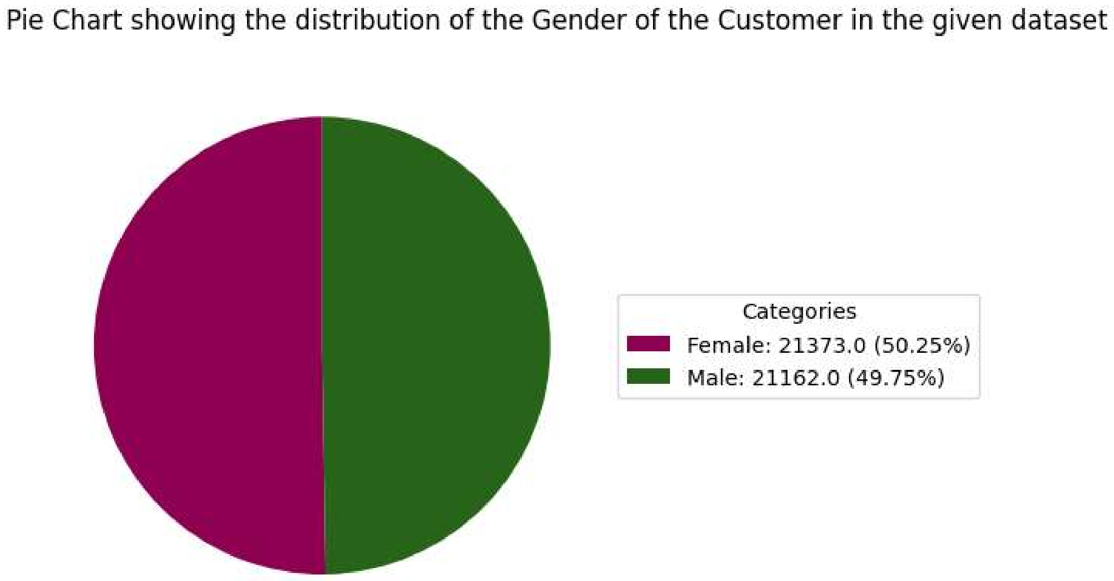


Pie Chart showing the distribution of the term of the loan taken in the given dataset





Pie Chart showing the distribution of the Race of the Customer in the given dataset



Pie Chart showing the distribution of the Gender of the Customer in the given dataset

Exploratory Data Analysis:

In this section, we try to observe the distribution of data and perform appropriate pre-processing. We also observe the variation of several factors over loan status and try to see if there is any notable correlation data before starting the modelling.

The unique values in each column are:

```
For column id: No of unique values = 42535
For column member_id: No of unique values = 42535
For column loan_amnt: No of unique values = 898
For column funded_amnt: No of unique values = 1051
For column funded_amnt_inv: No of unique values = 9240
For column term: No of unique values = 2
For column installment: No of unique values = 16459
For column emp_title: No of unique values = 30447
For column emp_length: No of unique values = 12
For column home_ownership: No of unique values = 5
For column annual_inc: No of unique values = 5598
For column verification_status: No of unique values = 3
For column issue_d: No of unique values = 55
For column loan_status: No of unique values = 9
For column pymnt_plan: No of unique values = 2
For column desc: No of unique values = 28948
For column purpose: No of unique values = 14
For column title: No of unique values = 20964
For column addr_state: No of unique values = 50
For column dti: No of unique values = 2894
For column delinq_2yrs: No of unique values = 13
For column inq_last_6mths: No of unique values = 30
For column mths_since_last_delinq: No of unique values = 96
For column mths_since_last_record: No of unique values = 117
For column open_acc: No of unique values = 45
For column pub_rec: No of unique values = 7
For column revol_bal: No of unique values = 22709
For column total_acc: No of unique values = 84
For column out_prncp: No of unique values = 547
For column out_prncp_inv: No of unique values = 548
For column total_pymnt: No of unique values = 40579
For column total_pymnt_inv: No of unique values = 40108
For column total_rec_prncp: No of unique values = 8214
For column total_rec_int: No of unique values = 37533
For column total_rec_late_fee: No of unique values = 1562
For column recoveries: No of unique values = 4530
For column collection_recovery_fee: No of unique values = 2857
For column collections_12_mths_ex_med: No of unique values = 2
For column pub_rec_bankruptcies: No of unique values = 4
For column interest_rate: No of unique values = 394
For column revol_utilization: No of unique values = 1099
For column number_bc_gt_75: No of unique values = 4
For column fico_score: No of unique values = 42
For column lti: No of unique values = 13345
For column month_since_oldest_tl: No of unique values = 457
For column race_name: No of unique values = 4
For column gender: No of unique values = 2
```

Also, the number of null values in each columns are:

```
id                                0
member_id                         0
loan_amnt                         0
funded_amnt                       0
funded_amnt_inv                   0
term                              0
installment                       0
emp_title                      2626
emp_length                     1112
home_ownership                    0
annual_inc                        4
verification_status               0
issue_d                           0
loan_status                       0
pymnt_plan                        0
desc                          13521
purpose                           0
title                            13
addr_state                        0
dti                               0
delinq_2yrs                      29
inq_last_6mths                   29
mths_since_last_delinq        26926
mths_since_last_record        38884
open_acc                         29
pub_rec                          29
revol_bal                         0
total_acc                        29
out_prncp                         0
out_prncp_inv                     0
```

total_pymnt	0
total_pymnt_inv	0
total_rec_prncp	0
total_rec_int	0
total_rec_late_fee	0
recoveries	0
collection_recovery_fee	0
collections_12_mths_ex_med	145
pub_rec_bankruptcies	1365
interest_rate	0
revol_utilization	90
number_bc_gt_75	0
fico_score	0
lti	4
month_since_oldest_tl	29
race_name	0
gender	0

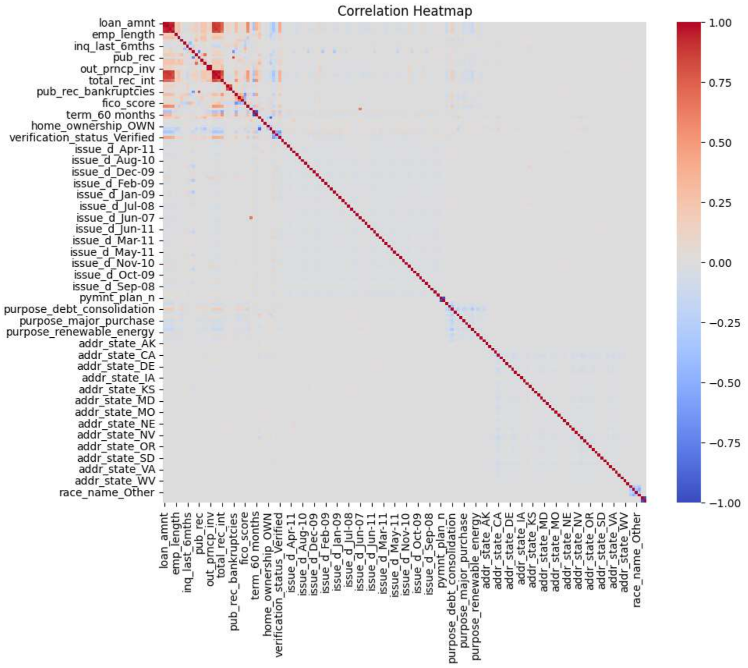
To make sure that our model performs correctly on the dataset, it would be ideal to fill in the missing values.

However, first, we drop the columns “emp_title”, “desc”, and “title”. These columns have been dropped as they are categorical and contain too many unique values. Keeping them as data for our model may cause overfitting and these columns will just end up complicating our data without adding any useful information to it.

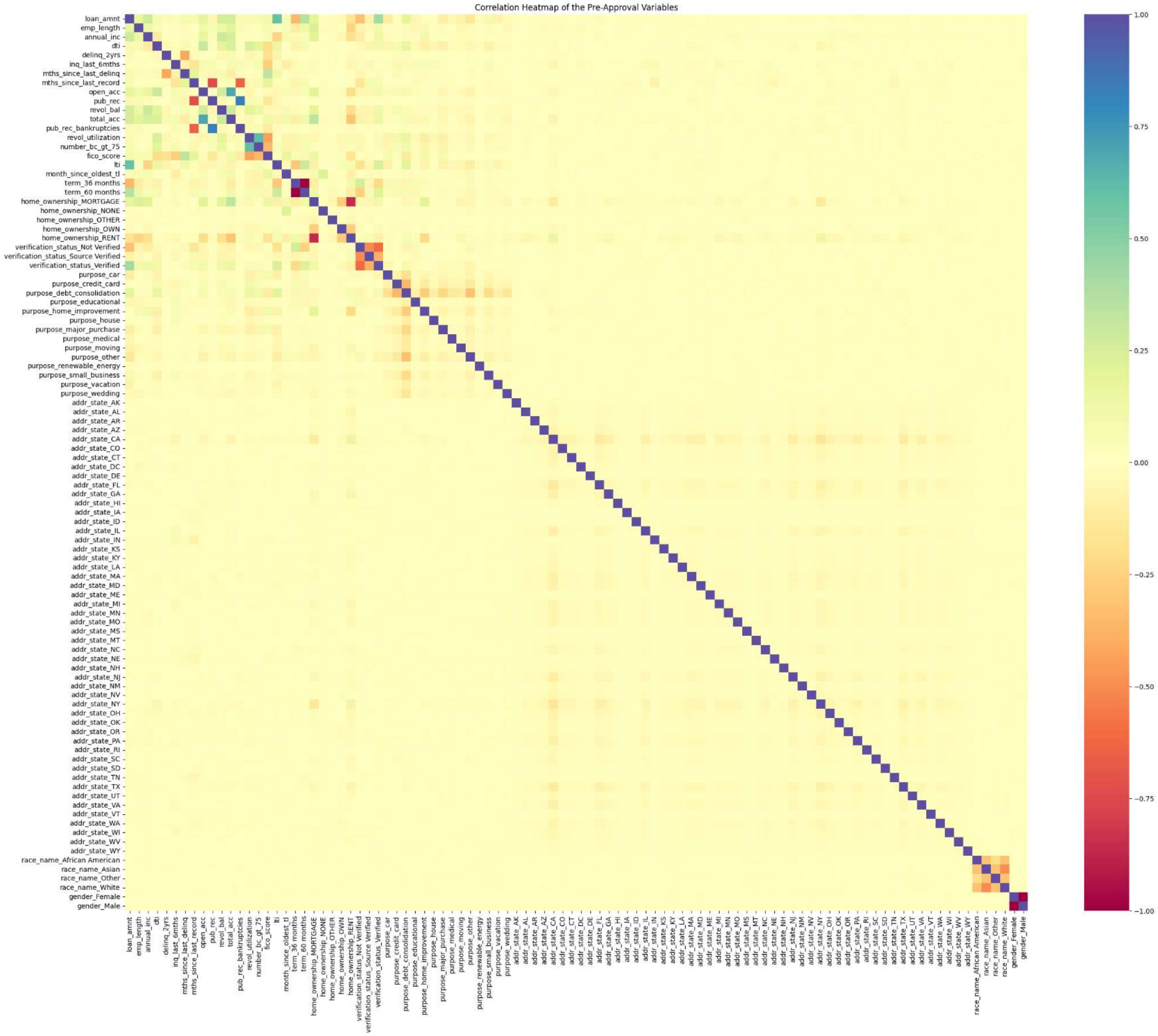
Besides these, we also remove the columns “collections_12_mths_ex_med” as the column either has its value as 0 or is null, and thus would not contribute to the model at all.

Next, we try to study the relationships between the various variables given in the dataset and see how the loan status varies with them.

The correlation between the variables of the entire dataset can be seen as:



From the above heatmap, we see that there is very little meaningful correlation between the given variables. To get a clearer picture, we can observe the correlation between the pre-approval variables which will be used to train the main model.

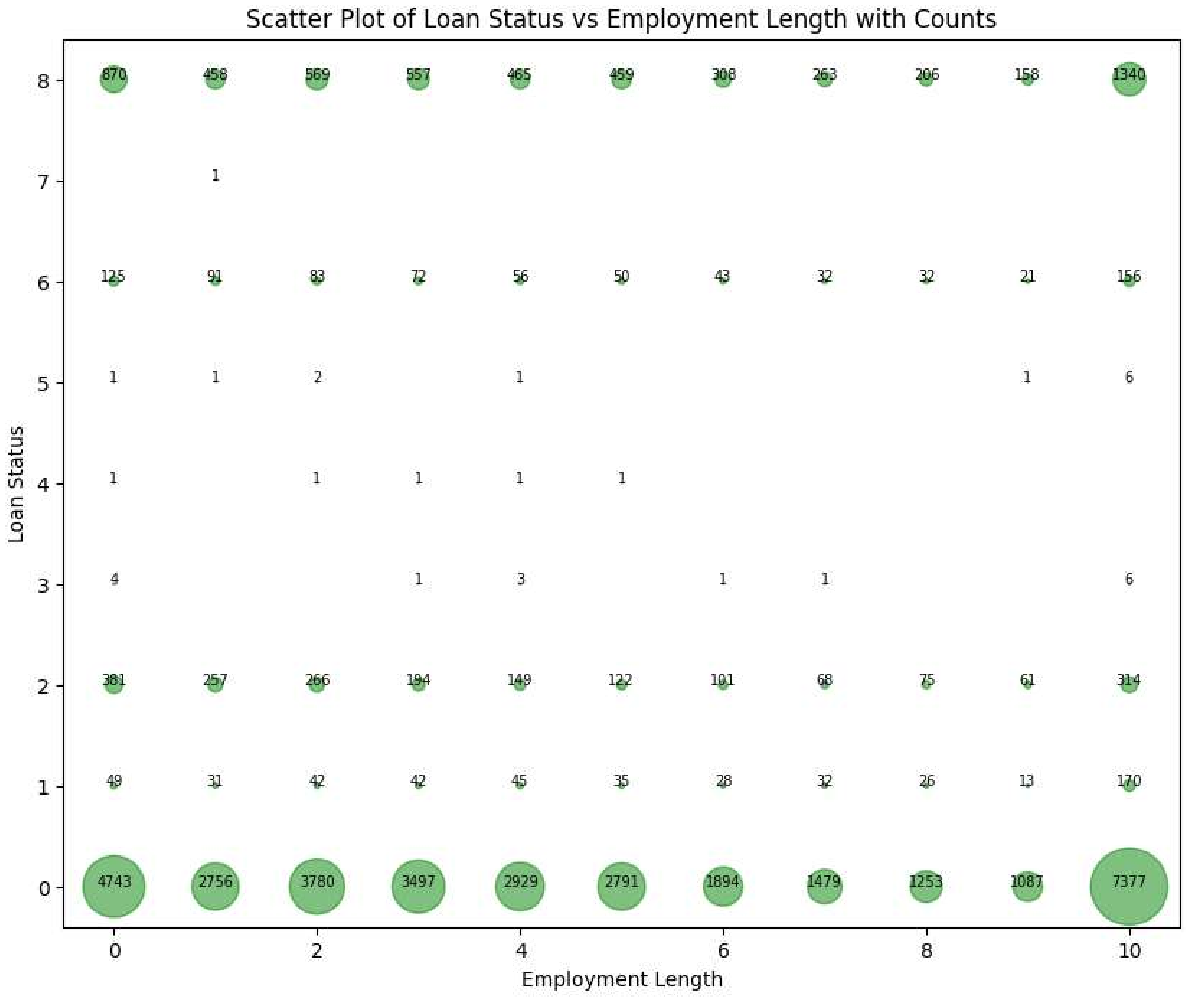


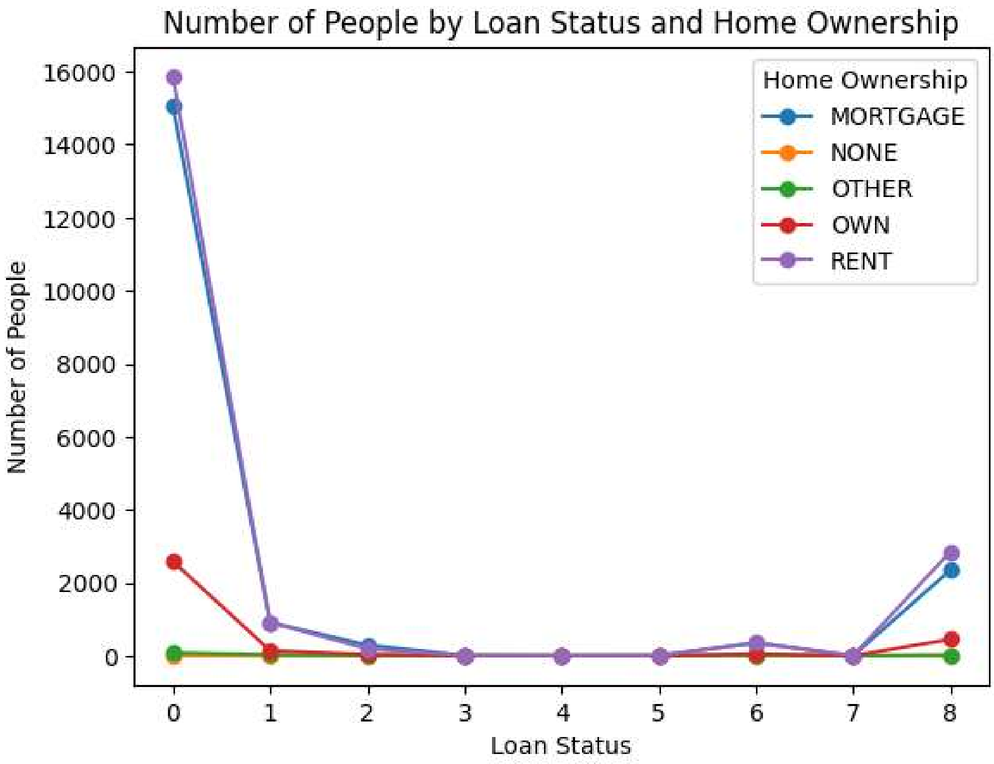
Thus, we see that the majority of the features are uncorrelated, which suggests that they may add new, independent information to the model.

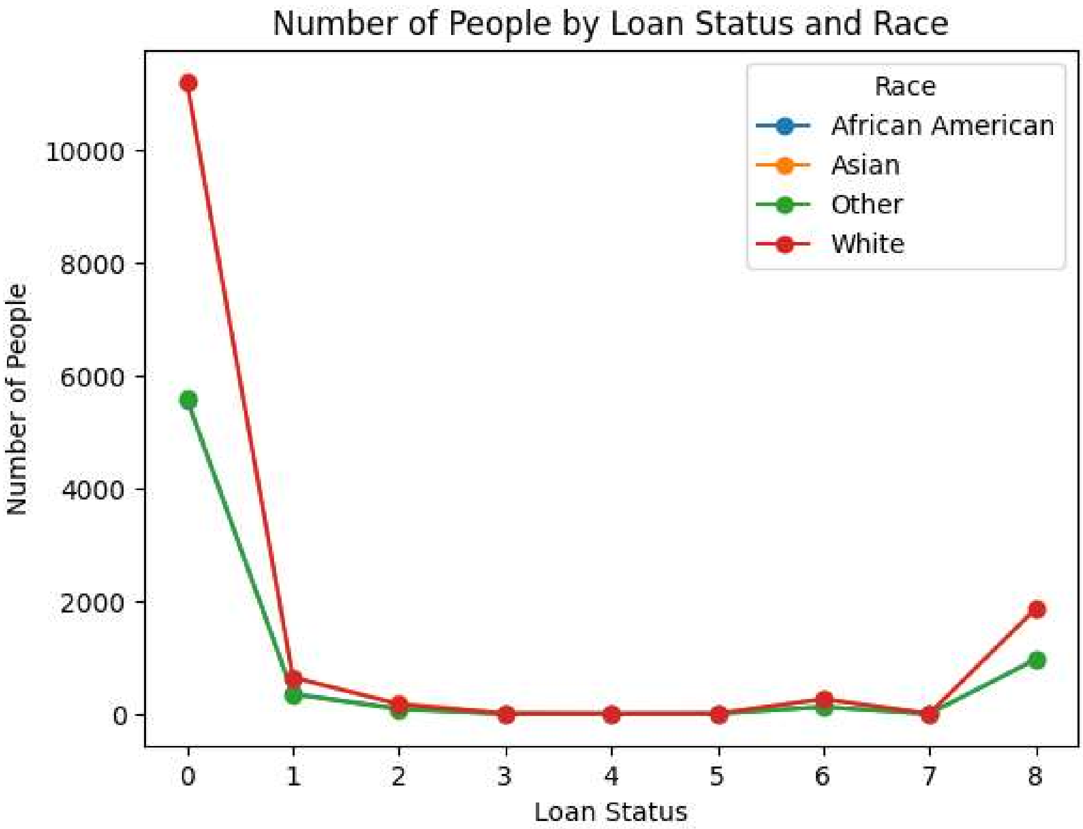
Also, we can observe the distribution of loan status over various variables:
In all the plots attached below, the loan status is as follows:

0: Fully Paid
1: Does not meet the credit policy. Status: Fully Paid
2: Current
3: In Grace Period
4: Late (16-30 days)
5: Late (31-120 days)
6: Does not meet the credit policy. Status: Charged Off
7: Default
8: Charged Off

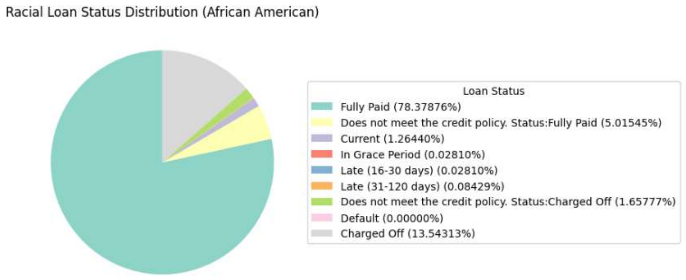
Thus, a higher number indicates a higher risk associated with the customer.

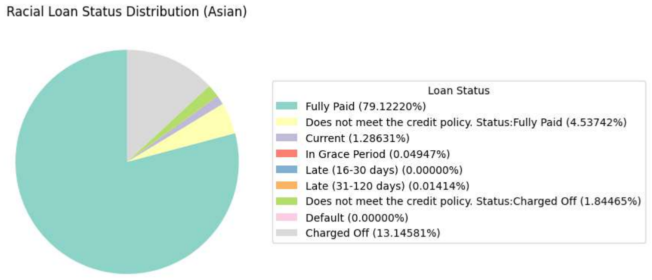


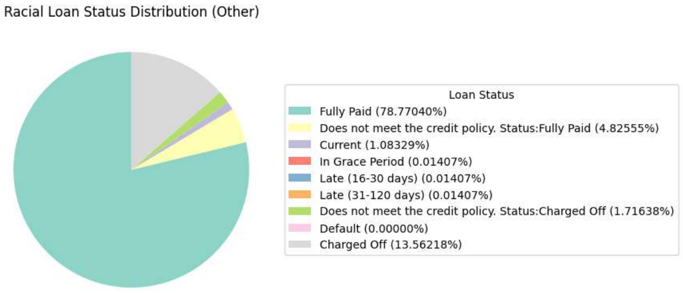


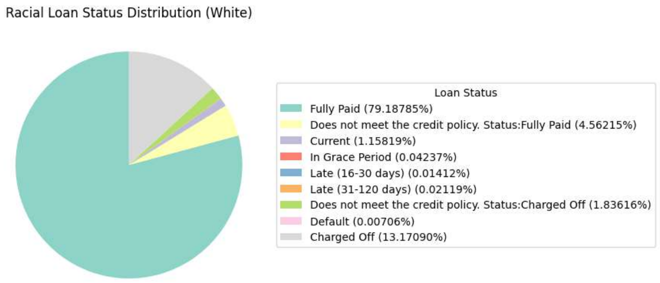


In the above plot, we see only two lines despite there being four races. This is because the distribution of Asian people follows the distribution of White people very closely. Similarly, the distribution of African Americans is very similar to the distribution of people from other races and gets overshadowed by its plot. To understand the percentage distribution of the loan status by percentage, we can take a look at the pie charts:



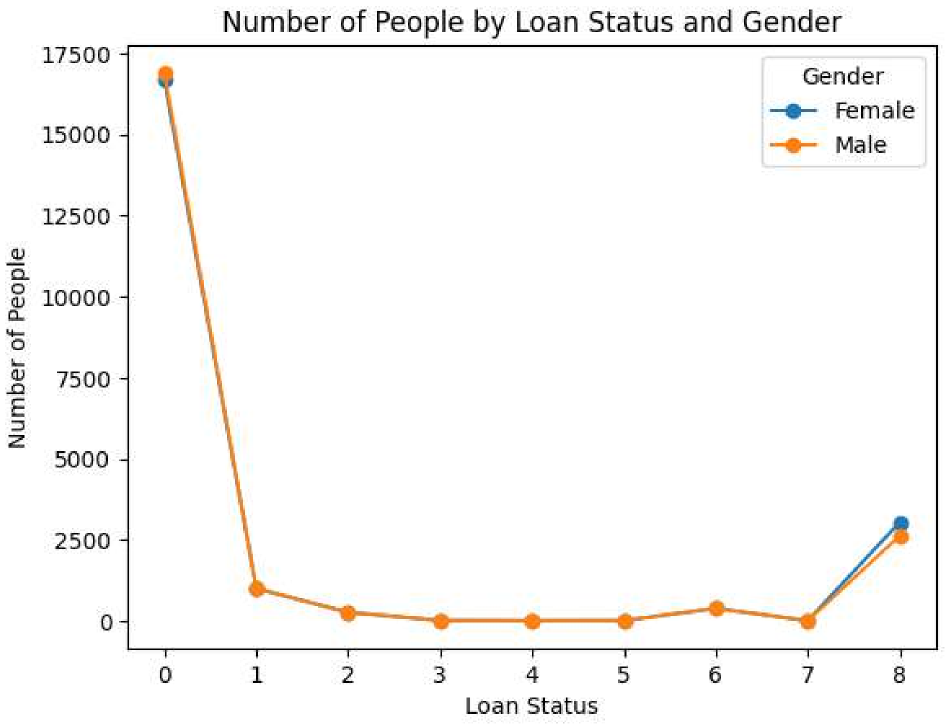






Thus, the distribution of loan status is nearly constant for all races.

We can do a similar study for gender as well.



Thus, we can say that both the genders in the dataset are extremely close with regards to loan status.

Data Pre-Processing:

To make sure that all the models fit properly on the data, appropriate transformations had to be applied.

The loan dataset underwent several preprocessing steps to prepare it for analysis. Initially, the dataset was read using 'latin1' encoding to handle any special characters. Unnecessary columns such as 'emp_title', 'title', 'desc', and 'collections_12_mths_ex_med' were dropped to streamline the dataset. Employment length categories, originally in textual format, were replaced with corresponding numerical values for uniformity. Missing values in 'emp_length' were filled with 0, while missing values in 'annual_inc' were substituted with the column's mean. A new ratio, 'Division' (loan amount divided by annual income), was calculated and used to fill missing values in 'lti', after which 'Division' was dropped. Specific strategies were employed to fill missing values in other columns: the maximum value multiplied by 5 was used for 'mths_since_last_delinq', 'mths_since_last_record', and 'month_since_oldest_tl'; and zeros were used for 'pub_rec_bankruptcies', 'delinq_2yrs', 'inq_last_6mths', 'open_acc', 'pub_rec', 'total_acc', and 'revol_utilization'. These steps ensured a complete and consistent dataset for subsequent analysis.

The output variable and input columns had to be defined and scaled appropriately. The categorical columns had to be one-hot encoded to make sure that the models understood them.

Creating the output column:

The output variable **'y'** was modelled taking into account several loan post-approval features. The post approval features that were identified were 'funded_amnt', 'funded_amnt_inv', 'issue_d', 'loan_status', 'pymnt_plan', 'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee', 'recoveries', 'collection_recovery_fee', 'interest_rate', and 'installment'.

Then, a new column called 'total_amnt_to_be_paid' was created by multiplying the installment with the term of the loan. It was then divided by 'total_pymnt' to emulate the total time lapsed from the date of issuing the loan.

Four ratios were created out of these to aid with refining of the output variable which was primarily based on 'loan_status'. The loan statuses 'Fully Paid', and 'Does not meet the credit policy. Status: Fully Paid' were automatically assigned to be approved as the output. On the other hand, the statuses 'Late (31-120 days)', 'Does not meet the credit policy. Status: Charged Off', 'Default', 'Charged Off' were assigned to be rejected. The ones in the middle left out were first refined based on the ratios created.

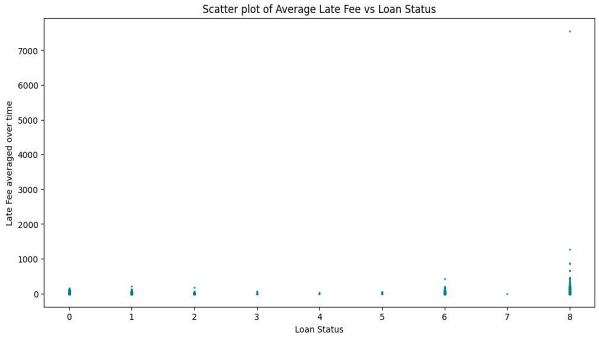
The ratios were:

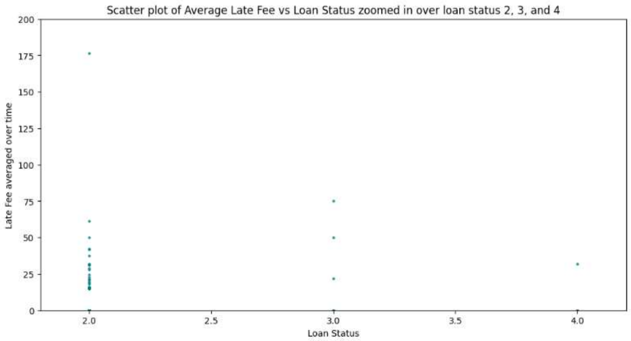
1. $r1 = \frac{total\ late\ fee\ received}{time\ elapsed}$

   r1 was intended to represent the average late fee over time. A higher average late fee is typically associated with financial instability and indicates a higher risk.

   The boundaries for different statuses were as follows:

   - Current: If r1 > 100, then the customer was assigned to be 'good' else, 'bad' was assigned.
   - In grace period: If r1 > 75, then the customer was assigned to be 'good' else, 'bad' was assigned.
   - Late (16 – 31 days): If r1 > 50, then the customer was assigned to be 'good' else, 'bad' was assigned.





2. $r2 = \frac{recovery}{total\ payment}$

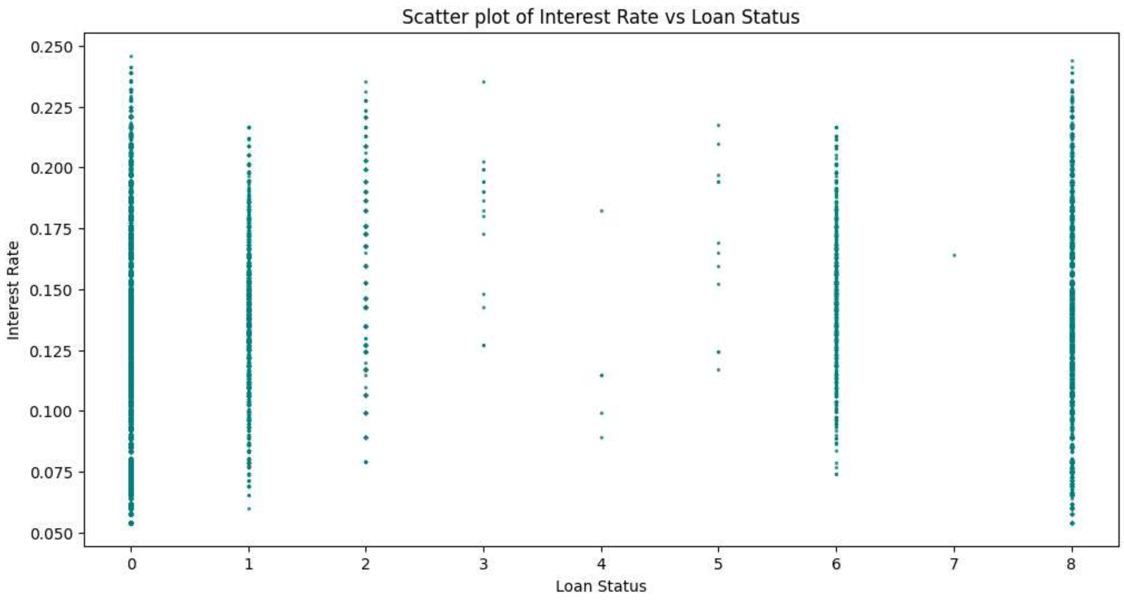
   This column was created to see how much of the total amount had to be recovered, and how much was actually paid. However, this was not used as it was non-zero only when the loan was charged off, and those customers had already been assigned as bad.

3. $r3 = \frac{interest\ received}{tota\ \ payment\ received}$

   This column was intended to check if a customer is paying much higher interest than others. A higher interest would denote that the customer was already considered a bit risky but still was given loan. However, this column was not used as the interest is typically higher for initial instalments, which would make this ratio biased towards the customers in the later stages of their loan. Instead of this ratio, column 'interest_rate' was used to create a decision boundary.

   The boundaries for different statuses were as follows:

   - Current: If interest rate < 0.25, then the customer was assigned to be 'good', else 'bad' was assigned.
   - In grace period: If interest rate < 0.175, then the customer was assigned to be 'good', else 'bad' was assigned.
   - Late (16 – 31 days): If interest rate < 0.15, then the customer was assigned to be 'good', else 'bad' was assigned.



4. $r4 = \frac{funded\ amou\quad by\ investors}{funded\ amount}$

   This ratio signifies the trust of investors in the particular customer. A very low ratio can be seen as a potentially high risk customer.

   This column was only used to refine the customers in Current as the other columns already had this ratio to be > 0.9 for all people

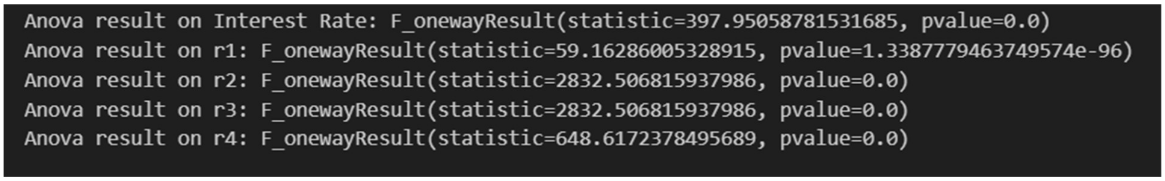
   The boundary assigned was:

   - Current: If r4 > 0.6, then the customer was assigned to be 'good' else, 'bad' was assigned.

All these methods were used to get the output column `'y'`.

We performed ANOVA test on these variables to make sure that these parameters are affecting the loan status of an individual.

To perform the test, our null hypothesis is that the means of all the loan statuses for each of the above variables are equal and any difference is just a coincidence. The results of the tests are:



```
Anova result on Interest Rate: F_onewayResult(statistic=397.95058781531685, pvalue=0.0)
Anova result on r1: F_onewayResult(statistic=59.16286005328915, pvalue=1.3387779463749574e-96)
Anova result on r2: F_onewayResult(statistic=2832.506815937986, pvalue=0.0)
Anova result on r3: F_onewayResult(statistic=2832.506815937986, pvalue=0.0)
Anova result on r4: F_onewayResult(statistic=648.6172378495689, pvalue=0.0)
```

Given the very low p-value, we can reject the null hypothesis that all group means are equal. This means that there are statistically significant differences between the means of the different groups.

The results indicate a significant difference between the categories of loan status.

Creating the input variables:

To create the input variables, all the output columns were dropped.

According to the **Equal Credit Opportunity Act (ECOA),** creditors must not consider certain factors when making a credit determination. These factors include race, colour, religion, national origin, sex (including sexual orientation and gender identity), marital status, age, whether all (or part) of a person’s income comes from public assistance, whether the applicant has in good faith acted on one of their rights under the federal credit laws (like if you exercised your right to dispute errors in your credit report).

Taking into account all these, some sensitive columns were identified and dropped. These were ‘gender’, ‘address_state’, ‘emp_title’, and ‘race_name’.

From the remaining columns, the categorical variables were one-hot encoded. Also, the continuous variables were scaled, but the scaled version was used only for logistic regression. Scaling has no effect on the output of tree-based methods and hence they were used on dataset without any scaling.

Creating the Train and Test sets:

The dataset was divided into train and test sets with the test size being 0.1

It was observed that there was significant class imbalance in the dataset with “good” customers accounting to more that 85% of the data. Thus, to handle class imbalance, SMOTE was used to generate synthetic data, which then made both the labels occupy 50% of the train set, effectively increasing the length of the train set.

Modelling and Evaluation:

Four models were used in fitting the data, the models being Logistic Regression, Decision Tree, Random Forest, and XGBoost.

The hyperparameters for Decision Tree, Random Forest, and XGBoost were tuned by employing random search.

As a first attempt, the models were fitted on the entire dataset, without removing the output or the sensitive variables. The ‘y’ in this case was solely based on loan output. Any loan status above class 5 was taken to be “good” and the rest were “bad”.

The results can be seen as:

MODEL	RESULTS
Logistic Regression	Classification report for Train Set:

	precision	recall	F1-score	support
0	1	1	1	32484
1	1	1	1	32484

Accuracy			1	64968
Macro avg	1	1	1	64968
Weighted avg	1	1	1	64968

Classification report for Test Set:

	precision	recall	F1-score	support
0	1.00	0.97	0.98	635
1	0.99	1	1	3619
Accuracy			1	4254
Macro avg	1	0.98	0.99	4254
Weighted avg	1	1	1	4254

Decision Tree

Classification report for Train Set:

	precision	recall	F1-score	support
0	1	1	1	32484
1	1	1	1	32484
Accuracy			1	64968
Macro avg	1	1	1	64968
Weighted avg	1	1	1	64968

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.99	0.98	0.99	635
1	1	1	1	3619
Accuracy			1	4254
Macro avg	0.99	0.99	0.99	4254
Weighted avg	1	1	1	4254

Random Forest

Classification report for Train Set:

	precision	recall	F1-score	support
0	1	1	1	32484
1	1	1	1	32484
Accuracy			1	64968
Macro avg	1	1	1	64968
Weighted avg	1	1	1	64968

Classification report for Test Set:

	precision	recall	F1-score	support
0	1	0.97	0.98	635
1	0.99	1	1	3619
Accuracy			1	4254
Macro avg	1	0.98	0.99	4254
Weighted avg	1	1	1	4254

XGBoost

Classification report for Train Set:

	precision	recall	F1-score	support
0	1	1	1	32484
1	1	1	1	32484

Accuracy			1	64968
Macro avg	1	1	1	64968
Weighted avg	1	1	1	64968

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.99	0.98	0.99	635
1	1	1	1	3619
Accuracy			1	4254
Macro avg	0.99	0.99	0.99	4254
Weighted avg	1	1	1	4254

Then, the sensitive columns were dropped and a similar analysis was performed, although this time only with XGBoost model.
The results can be seen as:

Classification report for Train Set:

	precision	recall	F1-score	support
0	1	1	1	32484
1	1	1	1	32484
Accuracy			1	64968
Macro avg	1	1	1	64968
Weighted avg	1	1	1	64968

Classification report for Test Set:

	precision	recall	F1-score	support
0	1	0.99	0.99	635
1	1	1	1	3619
Accuracy			1	4254
Macro avg	1	0.99	1	4254
Weighted avg	1	1	1	4254

The models seem to be very accurate. However, this is because the post-loan approval columns were not removed from the input data. Including these columns led to data leakage, where information from the future (post-loan approval) influences the model during training, resulting in overly optimistic performance metrics that will not generalise to real-world data. When approving new loans, a lot of information which has been used for training will not be available.

Thus, we try training models again, this time after dropping the columns which can be available only after the loan is approved. Here the output variable ‘y’ is also the one we obtained after refining.

The results can be seen as:

MODEL RESULTS

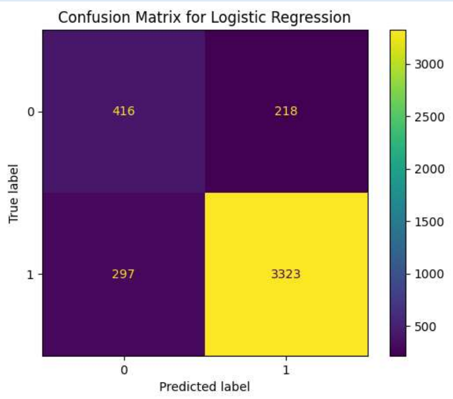
Logistic Regression

Classification report for Train Set:

	precision	recall	F1-score	support
0	0.91	0.89	0.90	32488
1	0.89	0.91	0.90	32488
Accuracy			0.90	64976
Macro avg	0.90	0.90	0.90	64976
Weighted avg	0.90	0.90	0.90	64976

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.58	0.66	0.62	634
1	0.94	0.92	0.93	3620
Accuracy			0.88	4254
Macro avg	0.76	0.79	0.77	4254
Weighted avg	0.89	0.88	0.88	4254


Confusion Matrix for Logistic Regression

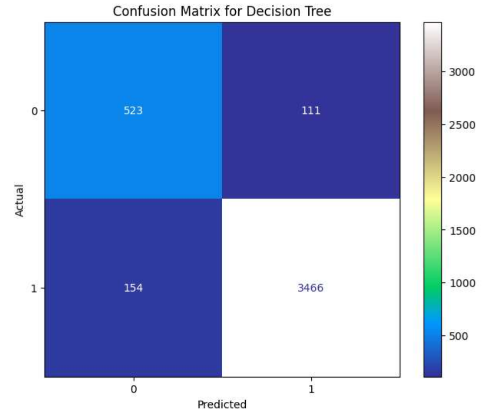
Decision Tree

Classification report for Train Set:

	precision	recall	F1-score	support
0	0.91	0.89	0.90	32488
1	0.89	0.91	0.90	32488
Accuracy			0.90	64976
Macro avg	0.90	0.90	0.90	64976
Weighted avg	0.90	0.90	0.90	64976

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.58	0.66	0.62	634
1	0.94	0.92	0.93	3620
Accuracy			0.88	4254
Macro avg	0.76	0.79	0.77	4254
Weighted avg	0.89	0.88	0.88	4254



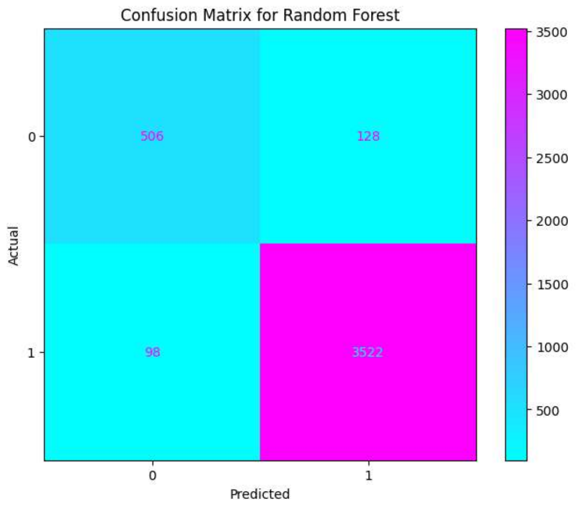
Random Forest

Classification report for Train Set:

	precision	recall	F1-score	support
0	0.98	0.96	0.97	32488
1	0.97	0.98	0.97	32488
Accuracy			0.97	64976
Macro avg	0.97	0.97	0.97	64976
Weighted avg	0.97	0.97	0.97	64976

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.84	0.80	0.82	634
1	0.96	0.97	0.97	3620
Accuracy			0.95	4254
Macro avg	0.90	0.89	0.89	4254
Weighted avg	0.95	0.95	0.95	4254



XGBoost

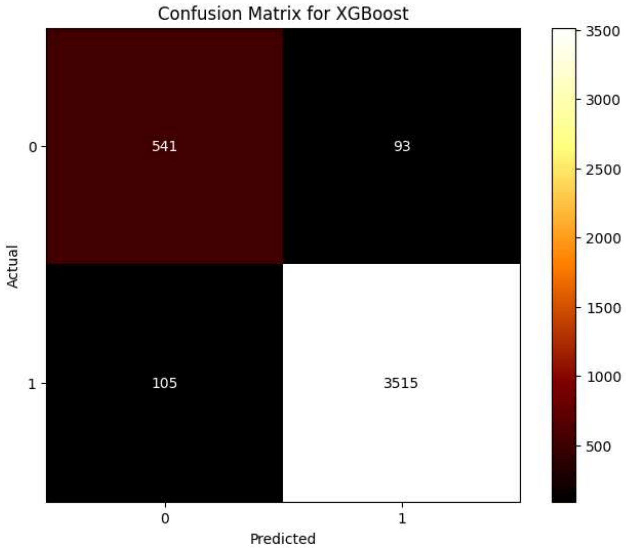
Classification report for Train Set:

	precision	recall	F1-score	support
0	0.98	0.96	0.97	32488
1	0.96	0.98	0.97	32488
Accuracy			0.95	64976

Macro avg	0.91	0.91	0.91	64976
Weighted avg	0.95	0.95	0.95	64976

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.84	0.85	0.85	634
1	0.97	0.97	0.97	3620
Accuracy			0.95	4254
Macro avg	0.91	0.91	0.91	4254
Weighted avg	0.95	0.95	0.95	4254



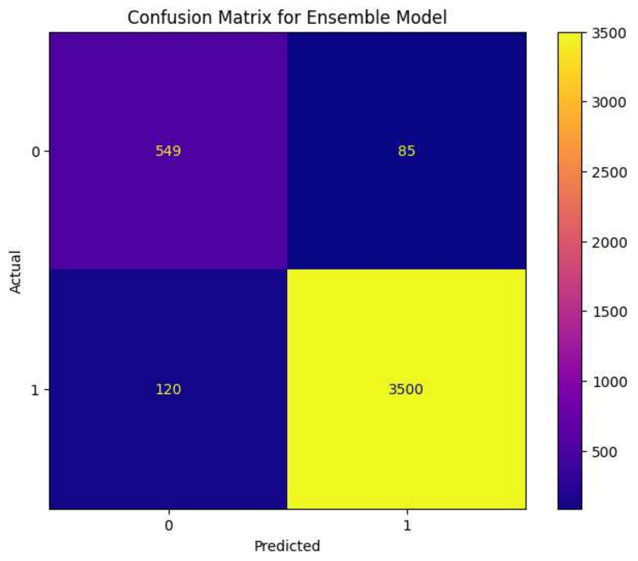
Ensemble Model

Classification report for Train Set:

	precision	recall	F1-score	support
0	0.97	0.98	0.98	32488
1	0.98	0.97	0.98	32488
Accuracy			0.98	64976
Macro avg	0.98	0.98	0.98	64976
Weighted avg	0.98	0.98	0.98	64976

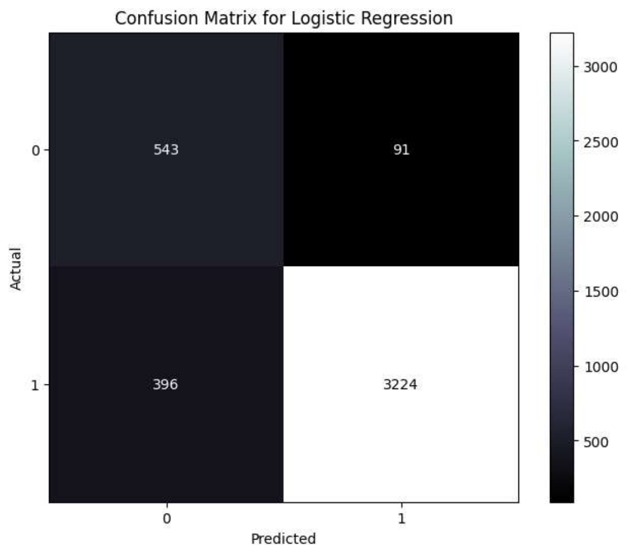
Classification report for Test Set:

	precision	recall	F1-score	support
0	0.82	0.87	0.84	634
1	0.98	0.97	0.97	3620
Accuracy			0.95	4254
Macro avg	0.90	0.92	0.91	4254
Weighted avg	0.95	0.95	0.95	4254

Then, we perform a similar analysis with the sensitive columns removed. The results for this modelling can be seen as:

<table>
<tr><th>MODEL</th><th>RESULTS</th></tr>
<tr><td>Logistic Regression</td><td>
Classification report for Train Set:
<table>
<tr><th></th><th>precision</th><th>recall</th><th>F1-score</th><th>support</th></tr>
<tr><td>0</td><td>0.88</td><td>0.89</td><td>0.88</td><td>32488</td></tr>
<tr><td>1</td><td>0.89</td><td>0.88</td><td>0.88</td><td>32488</td></tr>
<tr><td>Accuracy</td><td></td><td></td><td>0.88</td><td>64976</td></tr>
<tr><td>Macro avg</td><td>0.88</td><td>0.88</td><td>0.88</td><td>64976</td></tr>
<tr><td>Weighted avg</td><td>0.88</td><td>0.88</td><td>0.88</td><td>64976</td></tr>
</table>
Classification report for Test Set:
<table>
<tr><th></th><th>precision</th><th>recall</th><th>F1-score</th><th>support</th></tr>
<tr><td>0</td><td>0.58</td><td>0.86</td><td>0.69</td><td>634</td></tr>
<tr><td>1</td><td>0.97</td><td>0.89</td><td>0.93</td><td>3620</td></tr>
<tr><td>Accuracy</td><td></td><td></td><td>0.88</td><td>4254</td></tr>
<tr><td>Macro avg</td><td>0.78</td><td>0.87</td><td>0.81</td><td>4254</td></tr>
<tr><td>Weighted avg</td><td>0.91</td><td>0.89</td><td>0.89</td><td>4254</td></tr>
</table>
</td></tr>
</table>



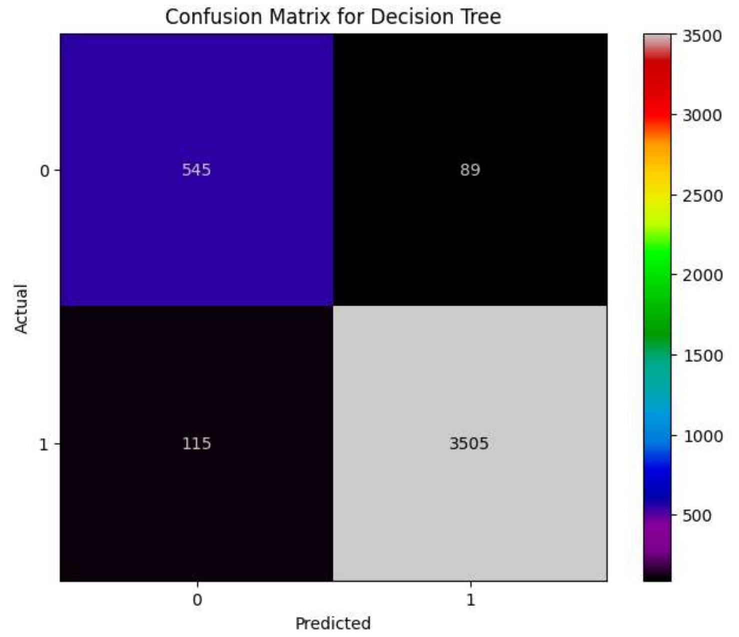
Decision Tree

Classification report for Train Set:

	precision	recall	F1-score	support
0	0.77	0.85	0.81	634
1	0.97	0.95	0.96	3620
Accuracy			0.94	4254
Macro avg	0.87	0.90	0.89	4254
Weighted avg	0.94	0.94	0.94	4254

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.97	0.97	0.97	32488
1	0.97	0.97	0.97	32488
Accuracy			0.97	64976
Macro avg	0.97	0.97	0.97	64976
Weighted avg	0.97	0.97	0.97	64976


Confusion Matrix for Decision Tree

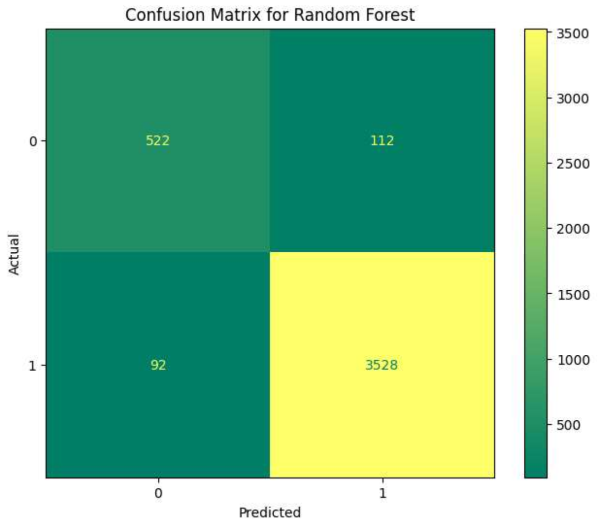
Random Forest

Classification report for Train Set:

	precision	recall	F1-score	support
0	0.85	0.82	0.84	634
1	0.97	0.97	0.97	3620
Accuracy			0.95	4254
Macro avg	0.91	0.90	0.90	4254
Weighted avg	0.95	0.95	0.95	4254

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.99	0.99	0.99	32488
1	0.99	0.99	0.99	32488
Accuracy			0.99	64976
Macro avg	0.99	0.99	0.99	64976
Weighted avg	0.99	0.99	0.99	64976



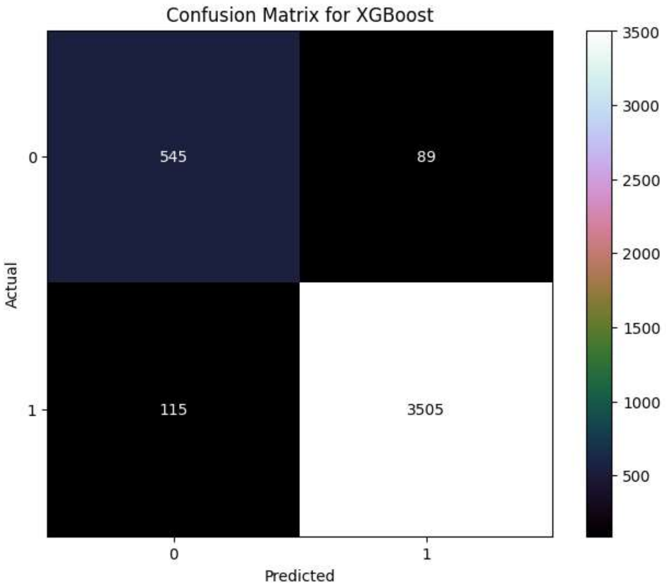
XGBoost

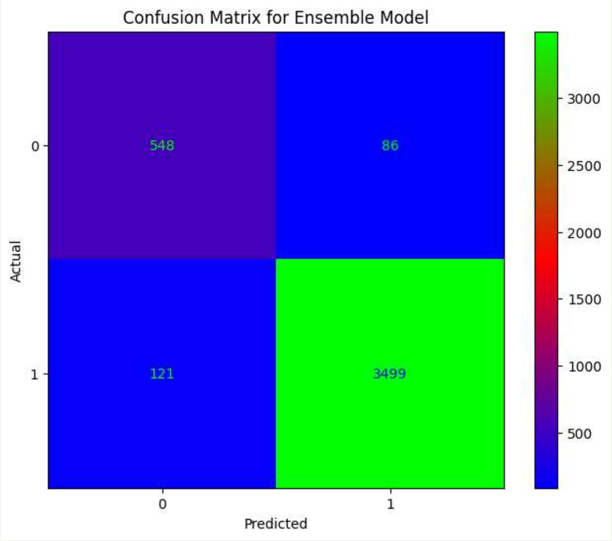
Classification report for Train Set:

	precision	recall	F1-score	support
0	0.98	0.96	0.97	32488
1	0.96	0.98	0.97	32488
Accuracy			0.95	64976
Macro avg	0.91	0.91	0.91	64976
Weighted avg	0.95	0.95	0.95	64976

Classification report for Test Set:

	precision	recall	F1-score	support
0	0.83	0.86	0.84	634
1	0.98	0.97	0.97	3620
Accuracy			0.95	4254
Macro avg	0.90	0.91	0.91	4254
Weighted avg	0.95	0.95	0.95	4254



<table>
<tr>
<td>Ensemble Model</td>
<td>
Classification report for Train Set:
<table>
<tr><th></th><th>precision</th><th>recall</th><th>F1-score</th><th>support</th></tr>
<tr><td>0</td><td>0.99</td><td>0.99</td><td>0.99</td><td>634</td></tr>
<tr><td>1</td><td>0.99</td><td>0.99</td><td>0.99</td><td>3620</td></tr>
<tr><td>Accuracy</td><td></td><td>0.99</td><td>0.99</td><td>4254</td></tr>
<tr><td>Macro avg</td><td>0.99</td><td>0.99</td><td>0.99</td><td>4254</td></tr>
<tr><td>Weighted avg</td><td>0.99</td><td>0.99</td><td>0.99</td><td>4254</td></tr>
</table>
Classification report for Test Set:
<table>
<tr><th></th><th>precision</th><th>recall</th><th>F1-score</th><th>support</th></tr>
<tr><td>0</td><td>0.82</td><td>0.86</td><td>0.84</td><td>32488</td></tr>
<tr><td>1</td><td>0.98</td><td>0.97</td><td>0.97</td><td>32488</td></tr>
<tr><td>Accuracy</td><td></td><td></td><td>0.88</td><td>64976</td></tr>
<tr><td>Macro avg</td><td>0.90</td><td>0.92</td><td>0.95</td><td>64976</td></tr>
<tr><td>Weighted avg</td><td>0.95</td><td>0.95</td><td>0.95</td><td>64976</td></tr>
</table>


</td>
</tr>
</table>

Thus, the model performs a lot worse than our previous model. This is just because now there is no data leakage and the loans are being approved (or rejected) based only on the pre-loan approval data, and is much more representative of real-world data than the previous attempt.

We observe that the precision and recall values in the test set for "bad" customers are low. This is because the original dataset was imbalanced towards the "good" customers. The data has been augmented through synthetic generation, but it is certainly not as good as actual data. However, in terms of loss, it is more imperative that lesser number of "bad" customers are given the loan, and out model achieves this objective very satisfactorily.

While sensitive columns like gender, race and address state have been removed to prevent bias, it is important to keep checking our model to make sure that it does not have an inherent bias through the other columns.

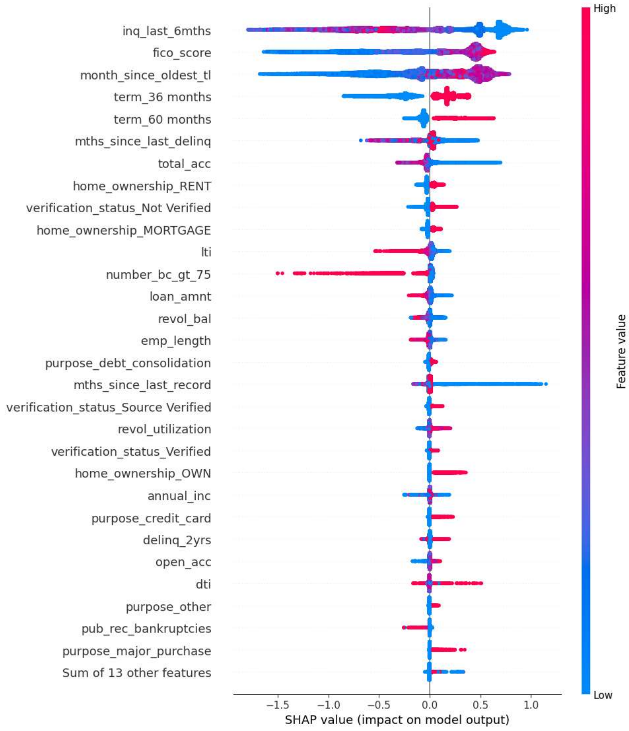
This can be done through regular testing of the output to make sure that the model is not getting biased towards a certain demographic. As data keeps getting added to a model, it is important that we perform regular sanity checks and make sure that the model is performing satisfactorily.

Understanding the Model Output:

According to the **Equal Credit Opportunity Act (ECOA)**, the customers have a right to seek information about the reason their application were rejected in case it happens. To make sure we are in compliance with the model, we make use of SHAP to understand the model output.

SHAP (SHapley Additive exPlanations) is a powerful tool used in machine learning to interpret and explain model predictions. It applies concepts from cooperative game theory to assign each feature's contribution to the prediction outcome. SHAP values provide a clear understanding of how individual features impact predictions, offering insights into the model's decision-making process. By using SHAP, we ensure transparency and compliance with regulations like the Equal Credit Opportunity Act (ECOA), as it enables us to provide customers with clear explanations regarding why their applications were accepted or rejected. This transparency not only enhances trust but also helps us validate and improve our models by identifying factors that significantly influence their decisions.

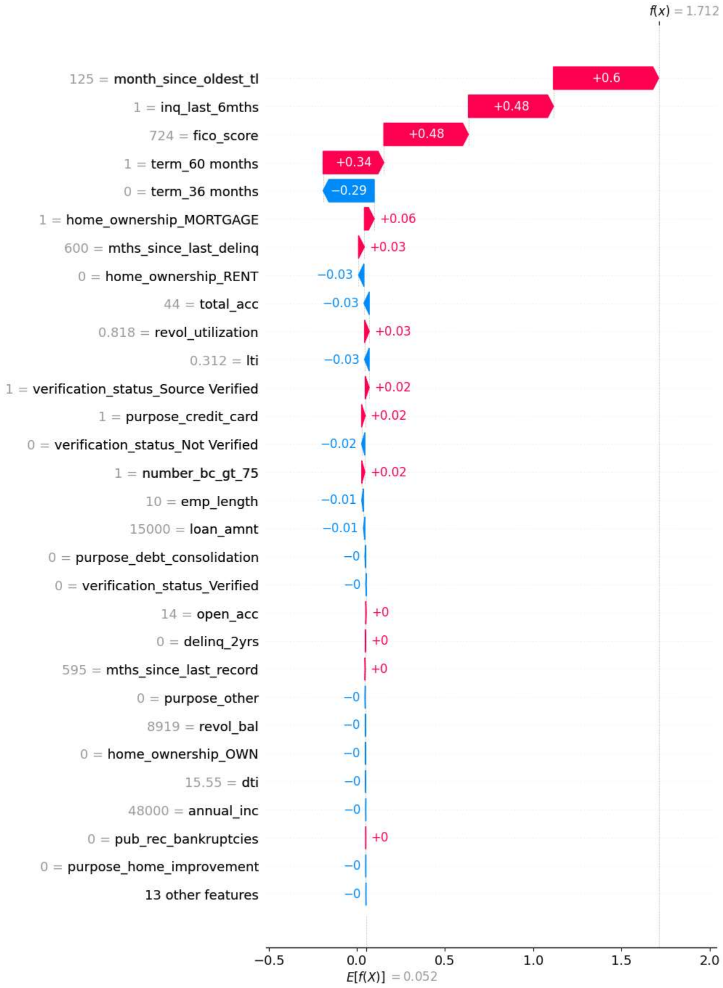
However, SHAP cannot work on ensemble models, hence for our analysis, we fit it on the final XGBoost model.
The values for the dataset can be seen as:



This SHAP summary plot illustrates the impact of various features on XGBoost predictions for loan outcome. Each dot represents a SHAP value for a feature in a specific observation, with the y-axis listing the features and the x-axis showing the SHAP values (impact on the model's output). Colours indicate feature values, where red represents high values and blue represents low values. For example, a high 'fico_score' (red) positively impacts the model's output, indicating lower risk, whereas more recent inquiries ('inq_last_6mths') negatively impact the output, indicating higher risk. The plot highlights key features like 'fico_score', 'month_since_oldest_ti', and 'term_36_months' as significant contributors to the model's predictions.

There is a base value given for each shap model. Then, the shap values of each output is added to the base value to get the final SHAP value. This SHAP value is then fed into logistic function to get the final probability of class 1.

The shap value for an individual applicant can be seen by a waterfall plot:



In this case, the model's base value is 0.052. The applicant's SHAP value is 1.712. On plugging it in logistic equation, the probability of this person being a good customer comes out to be 0.15 and their loan application is rejected.

Review Questions

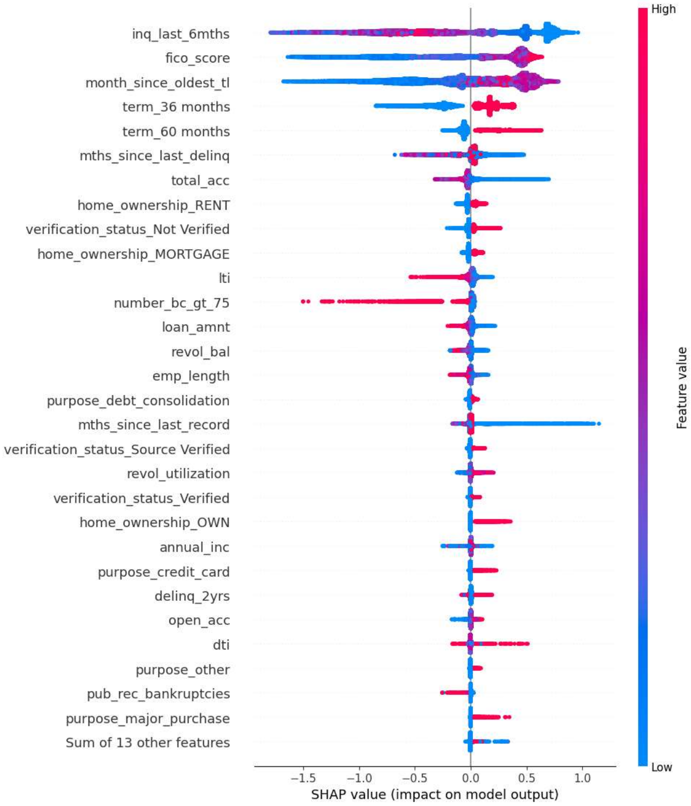
Q. As a decision maker, looking into these given booked accounts, how would you want to take a decision to approve/decline the same applications again in future? What would be your screening criterion(s), if any?

Ans: The screening criterion would be a data-driven process that analyses the account holder's past performance and reputation against the different loans issued as well as will consider their current financial position. The screening criterions would include the following information:

- The amount of loan applied for.
- The total amount committed to that loan at that point in time and the loan term
- The monthly payment owed by the borrower if the loan originates.
- Employment details: Job title and Employment Length
- The home ownership status provided by the borrower during registration (Rent, Own etc.)
- The self-reported annual income provided by the borrower during registration.
- A category provided by the borrower for the loan request and loan description
- The number of 30+ days past-due incidences of delinquency in the borrower's credit file in the past
- The number of inquiries in past (excluding auto and mortgage inquiries)
- Number of derogatory public records
- Remaining outstanding principal for total amount funded
- Payments received to date for total amount funded
- Principal, interest, and late fee received to date
- Number of public record bankruptcies
- Origination Fico score or the fico score at the point of loan origination
- Loan value to income ratio

Q. Please provide data driven rationale to substantiate the appropriateness of the characteristics you have identified in the screening logic.

The SHAP values can be used to analyse the importance of the features in the model.



As we can see in this graph, most of the variables used as input are significantly affecting the model's performance. Hence, we can say that they are important variables in deciding the outcome of a loan application.

Q. Given that you have a decision engine, which helps you to take a call based on the given data, how would you identify if your process has an unintentional bias and discriminate your customers based on some of the sensitive attributes, available in this dataset.

Ans: To identify unintentional bias and potential discrimination in a decision engine, especially regarding sensitive attributes such as race, gender, age, etc., we need a systematic and analytical. Here is how we could analyse the bias:

1. Collect Data:
   - Gather data on loan applications, including sensitive attributes and whether each application was approved or denied.
2. Statistical Analysis:
   - Calculate the approval rates for different races and genders.
   - Use visual tools to compare these rates and identify any disparities.
3. Statistical Tests:
   - Applying Chi-Square test to see if the distribution of the outcomes is independent of the sensitive information
   - T-test/ANOVA for comparing the means of a continuous outcome variable across different groups defined by a sensitive attribute.