Project Report
Data Analytics and Visualization
Loan Default Prediction by Group 8

Aparna Subha

Ashwini Gore

Simi Adelore

ABSTRACT

Defaulting on a loan happens when repayments aren't made for a certain period. Defaulting will drastically reduce your credit score, impact your ability to receive future credit, and can lead to the seizure of personal property (vehicle). The probability of default, sometimes abbreviated as POD or PD, expresses the likelihood the borrower will not maintain the financial capability to make scheduled debt payments. There are several major variables to consider: the financial health of the borrower; the severity of the consequences of a default for the borrower and the creditor; the size of the credit extension; historical trends in default rates; and a variety of macroeconomic considerations. We are trying to identify these factors and predict the probability of a borrower defaulting on the loan with the demographic and financial information present in the dataset so that financial institutions providing vehicle loans can make better and smarter decisions based on a customer portfolio. The variables we would consider for the prediction are loan default (Payment default in the first EMI on due date), asset cost (cost of the asset), age (customer age), employment type (employment type: salaried, self-employment and unemployed), credit history length (Time since first loan).

Data Description

The dataset which was used for this project was downloaded from Kaggle. The dataset has about 41 features and 233154 instances. It includes details about the cost of assets, employment status of the individuals, different age groups, the amount disbursed for loans, loan default status, etc. The dataset is a mix of qualitative and quantitative variables. Let's see the feature descriptions:

FEATURES	DESCRIPTION
Uniqueid	Identifier for customers
Loan_Default	Payment default in the first EMI on due date
Disbursed_Amount	Amount of Loan disbursed
Asset_Cost	Cost of the Asset
Ltv	Loan to Value of the asset
Current_Pincode	Current pin code of the customer
Date.Of.Birth	Date of birth of the customer
Employment.Type	Employment Type of the customer (Salaried/Self Employed)
Disbursaldate	Date of disbursement
Aadhar_Flag	if Aadhar was shared by the customer then flagged as 1
Perform_Cns.Score	Bureau Score
Perform_Cns.Score.Description	Bureau score description
Pri.No.Of.Accts	count of total loans taken by the customer at the time of disbursement
Pri.Active.Accts	count of active loans taken by the customer at the time of disbursement
Pri.Overdue.Accts	count of default accounts at the time of disbursement
Pri.Current.Balance	total Principal outstanding amount of the active loans
Pri.Sanctioned.Amount	total amount that was sanctioned for all the loans
Pri.Disbursed.Amount	total amount that was disbursed for all the loans
Sec.No.Of.Accts	count of total loans taken by the customer at the time of disbursement
Sec.Active.Accts	count of active loans taken by the customer at the time of disbursement
Sec.Overdue.Accts	count of default accounts at the time of disbursement
Sec.Current.Balance	total Principal outstanding amount of the active loans
Sec.Sanctioned.Amount	total amount that was sanctioned for all the loans
Sec.Disbursed.Amount	total amount that was disbursed for all the loans
Primary.Instal.Amt	EMI Amount of the primary loan

Sec.Instal.Amt	EMI Amount of the secondary loan
New.Accts.In.Last.Six.Months	New loans taken by the customer in the last six months before the disbursement
Delinquent.Accts.In.Last.Six.Months	Loans defaulted in the last 6 months
Average.Acct.Age	Average loan tenure
Credit.History.Length	Time since first loan
No.Of_Inquiries	Enquires done by the customer for loans
Year Of Birth	Birth year of the customer
Age	Age of the customer

Tools and Techniques:

We used Python for data cleaning and analysis mainly using libraries Pandas and Scikit Learn

For visualizing the data we have used Tableau

Data Cleaning:

The first thing we did was to clean the data set. There was only one column (Employee type) which had 7661 missing values, we also had to convert the data of births to ages, drop columns not needed, convert some columns into categorical data and encode some others.

Treating Missing Values:

Since we have a significant number of null values (7661) in the Employment Type column, we assumed that because there were just two options, unemployed members had to put NA, so we created another category with the null values.

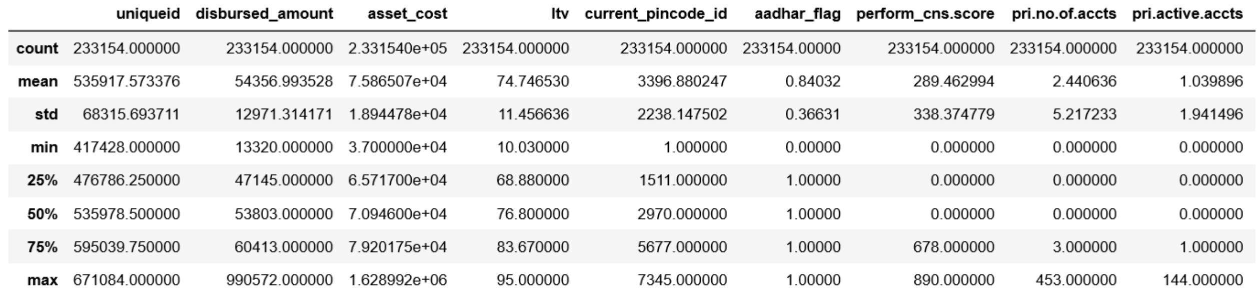
The rest of the cleaning:

- We had to transform the Date of birth with this format; 1984-01-01 to their ages for better visualizations.
- We converted credit.history.length and average.acct.age from objects to integers
- We also converted loan default and employment type to categories.
- We encoded the Perform cns score descriptions to no-history, very low, low, medium, high, and very high.

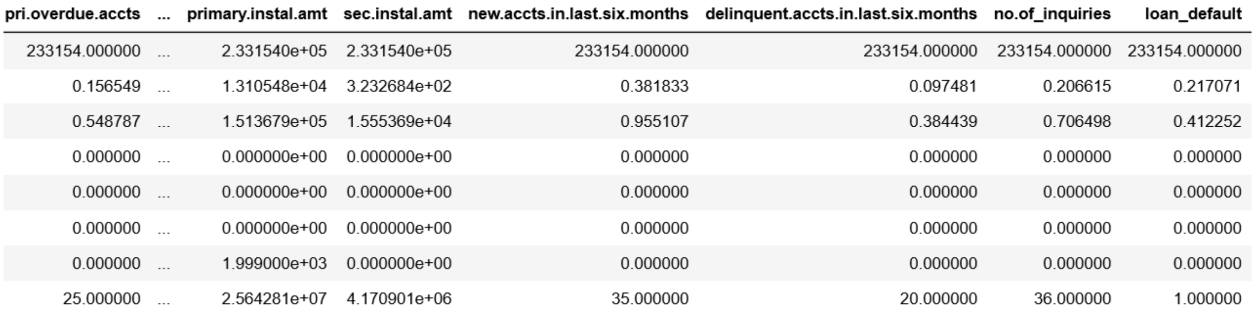
- Lastly, we dropped unnecessary columns like pan_flag, voterid_flag, driving_flag, 'passport_flag,mobileno_avl_flag,employee_code_id,branch_id,supplier_id, manufacturer_id,state_id.
- We also checked for duplicates and found none.
- We also checked for outliers, and we removed them

After cleaning, we were left with 33 columns and 213636 instances.

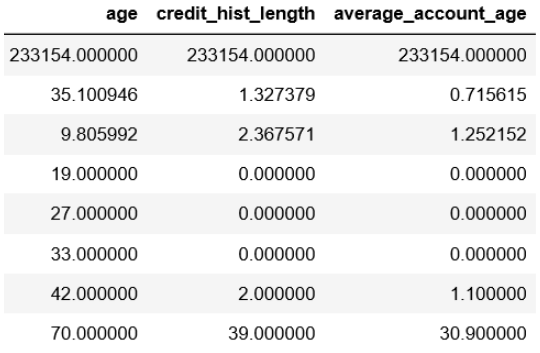
Descriptive Analysis:



	uniqueid	disbursed_amount	asset_cost	ltv	current_pincode_id	aadhar_flag	perform_cns.score	pri.no.of.accts	pri.active.accts
count	233154.000000	233154.000000	2.331540e+05	233154.000000	233154.000000	233154.00000	233154.000000	233154.000000	233154.000000
mean	535917.573376	54356.993528	7.586507e+04	74.746530	3396.880247	0.84032	289.462994	2.440636	1.039896
std	68315.693711	12971.314171	1.894478e+04	11.456636	2238.147502	0.36631	338.374779	5.217233	1.941496
min	417428.000000	13320.000000	3.700000e+04	10.030000	1.000000	0.00000	0.000000	0.000000	0.000000
25%	476786.250000	47145.000000	6.571700e+04	68.880000	1511.000000	1.00000	0.000000	0.000000	0.000000
50%	535978.500000	53803.000000	7.094600e+04	76.800000	2970.000000	1.00000	0.000000	0.000000	0.000000
75%	595039.750000	60413.000000	7.920175e+04	83.670000	5677.000000	1.00000	678.000000	3.000000	1.000000
max	671084.000000	990572.000000	1.628992e+06	95.000000	7345.000000	1.00000	890.000000	453.000000	144.000000



pri.overdue.accts	...	primary.instal.amt	sec.instal.amt	new.accts.in.last.six.months	delinquent.accts.in.last.six.months	no.of_inquiries	loan_default
233154.000000	...	2.331540e+05	2.331540e+05	233154.000000	233154.000000	233154.000000	233154.000000
0.156549	...	1.310548e+04	3.232684e+02	0.381833	0.097481	0.206615	0.217071
0.548787	...	1.513679e+05	1.555369e+04	0.955107	0.384439	0.706498	0.412252
0.000000	...	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000
0.000000	...	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000
0.000000	...	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000
0.000000	...	1.999000e+03	0.000000e+00	0.000000	0.000000	0.000000	0.000000
25.000000	...	2.564281e+07	4.170901e+06	35.000000	20.000000	36.000000	1.000000

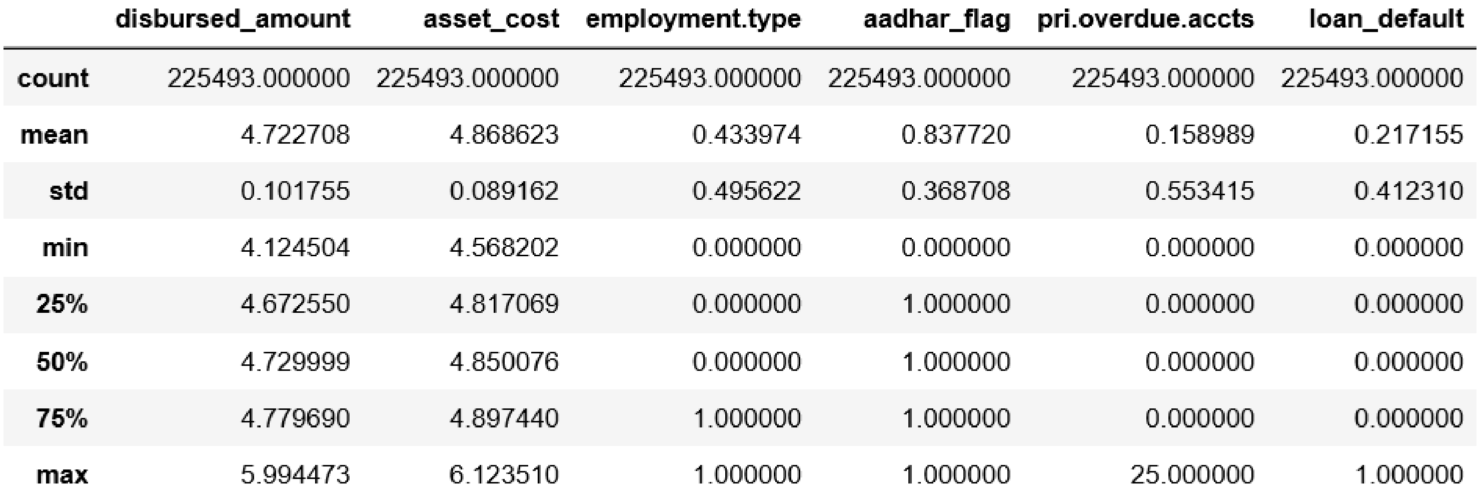


age	credit_hist_length	average_account_age
233154.000000	233154.000000	233154.000000
35.100946	1.327379	0.715615
9.805992	2.367571	1.252152
19.000000	0.000000	0.000000
27.000000	0.000000	0.000000
33.000000	0.000000	0.000000
42.000000	2.000000	1.100000
70.000000	39.000000	30.900000

Above is the descriptive analysis of our data before transformation or preprocessing. From the table we can see the:

- The maximum amount the bank disbursed as loan is 990572 US dollars, and the minimum was 13320 US dollars.
- The costs of assets for which the loans were made ranged from 3700 to 1628992 US dollars.
- The average age of people who took loans where in their late twenties. The ages ranged from 19 to 70.

The below is the descriptive analysis of our data after feature selection and data transformation (log).

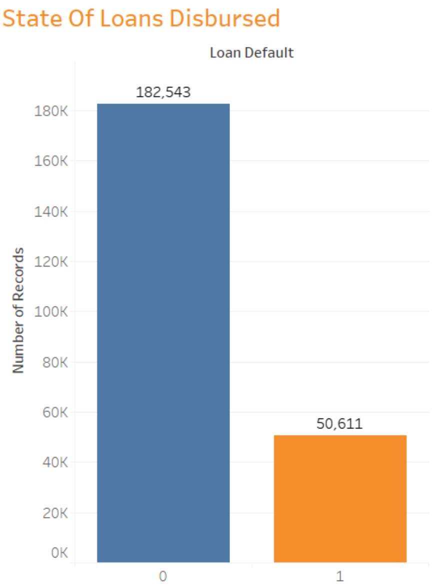


	disbursed_amount	asset_cost	employment.type	aadhar_flag	pri.overdue.accts	loan_default
count	225493.000000	225493.000000	225493.000000	225493.000000	225493.000000	225493.000000
mean	4.722708	4.868623	0.433974	0.837720	0.158989	0.217155
std	0.101755	0.089162	0.495622	0.368708	0.553415	0.412310
min	4.124504	4.568202	0.000000	0.000000	0.000000	0.000000
25%	4.672550	4.817069	0.000000	1.000000	0.000000	0.000000
50%	4.729999	4.850076	0.000000	1.000000	0.000000	0.000000
75%	4.779690	4.897440	1.000000	1.000000	0.000000	0.000000
max	5.994473	6.123510	1.000000	1.000000	25.000000	1.000000

Exploratory Analysis and Visualization:

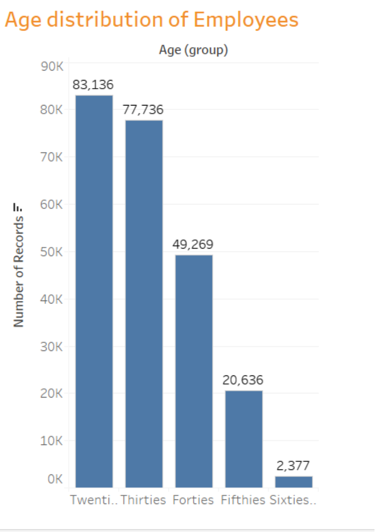
Using visuals, we tried to make sense of the data with boxplots, charts, and scatter plots for columns of concern and to examine the relationships between features.

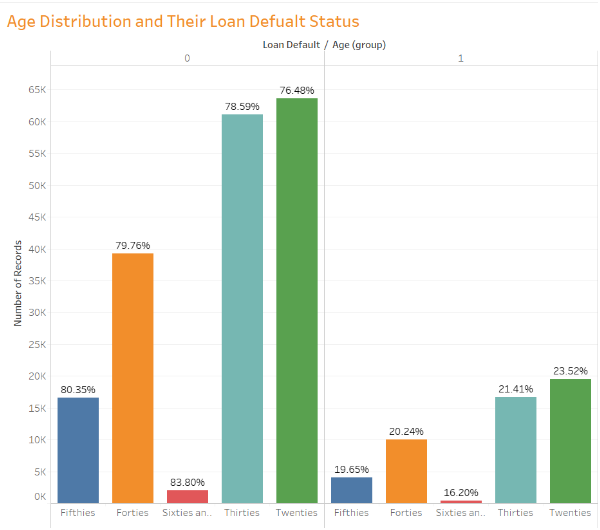
Loan Default Count:



- Status of loans given. The number of defaulters appears to be much less (< 50%) than those who repay their loans. The data is not balanced.

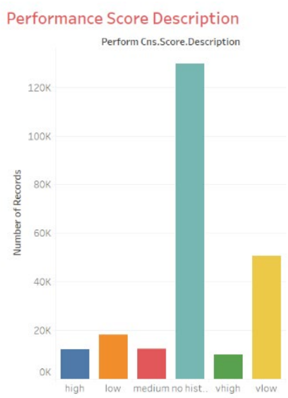
Age Distribution:

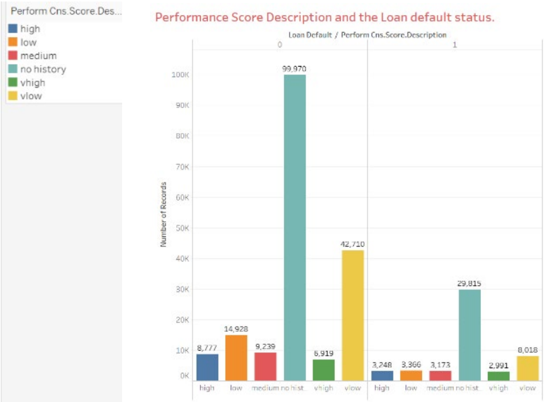




- Employees in the twenty-age range take more assets loans than others. It seems like the older people got, the less loans they took.
- The percentage of people who paid back over the total number of people in each group was calculated, and we found that as the age ranges went up, the number of defaulters lessened, making those in their twenties the must defaulters on loans.
- It is not a surprise that those in their sixties and up don’t take as many loans as individuals between 20 and 40, our guess is they don’t have much expenses anymore. Between the age of 20-40 most people get married and have kids so there is a lot they need money for.

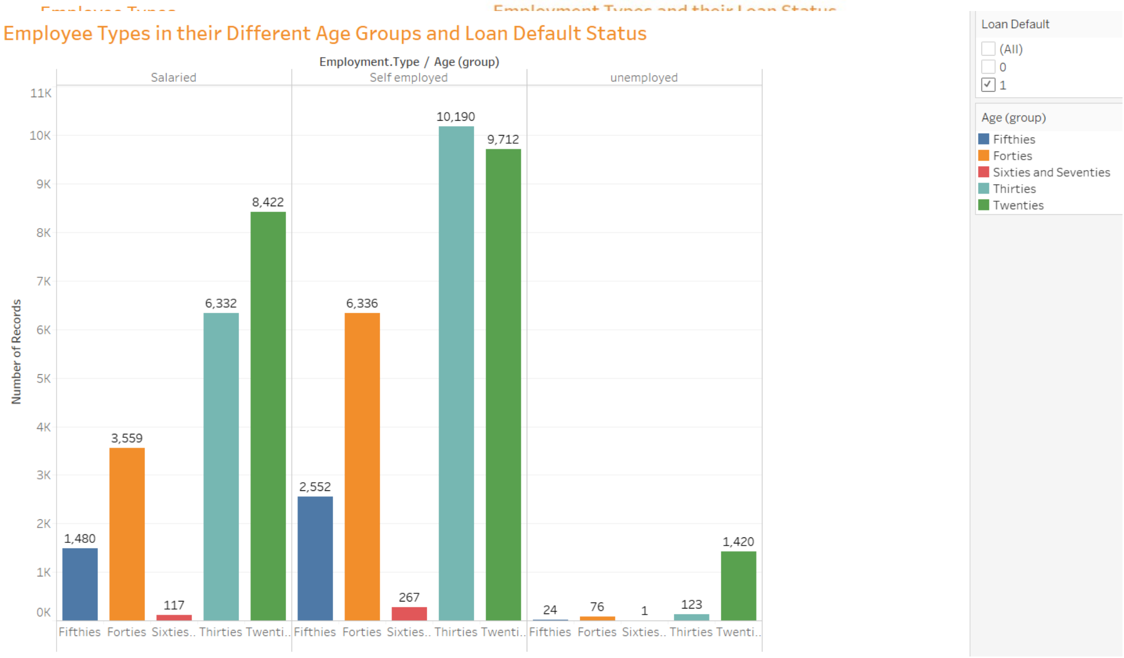
Credit Score:





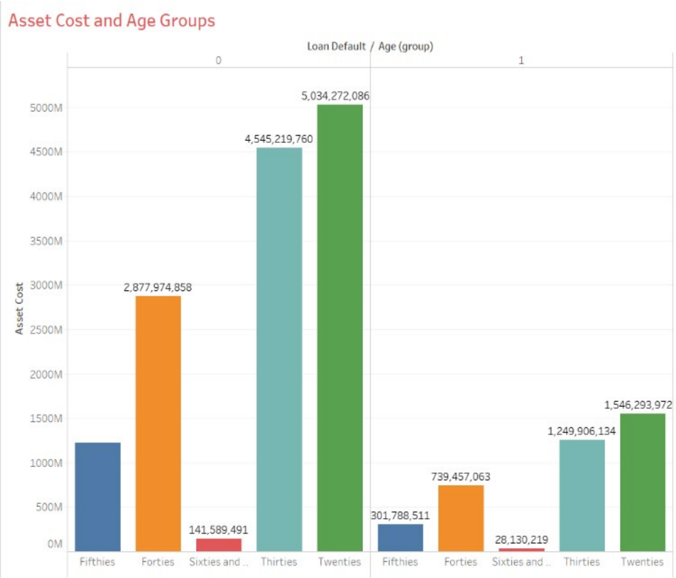
- After encoding the performance score, we found that most of the lenders had no history records with the banks.
- From the visual, we can see that the most defaulters had either no history or very low-risk scores. The individuals with no history had very bad credit score and could be considered as very high-risk individual so it makes perfect sense that they have the most defaults but for people who were classified as very low risk, which means they had scores ranging from 700-890, it doesn’t make sense that they are next in line for the most defaulters. This could either be a problem with the data (because it’s not balanced) or it really happened.

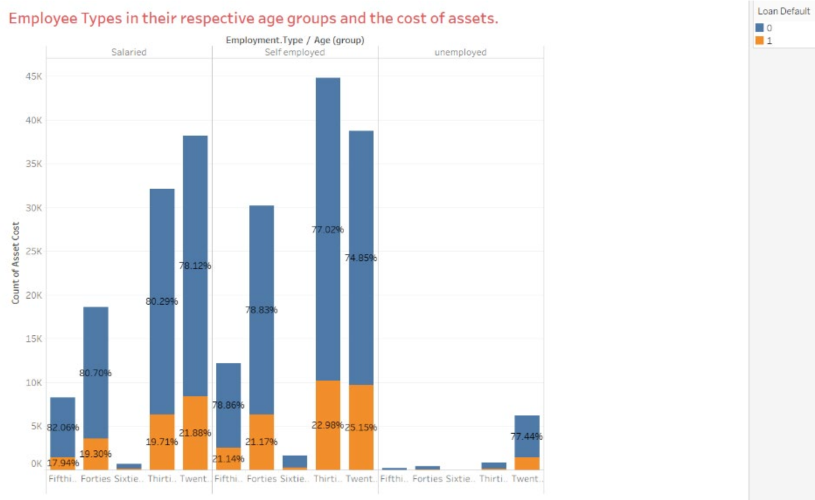
Employee Type:



- In this dataset we have more people who are Self-employed.
- After calculating the percentage of defaulters and non-defaulters by their employment types, we observed that the must defaulters are the self-employed individuals followed by unemployed folks.
- Most of these self-employed defaulters are in their thirties. Most of the Salaried and unemployed people are in their twenties

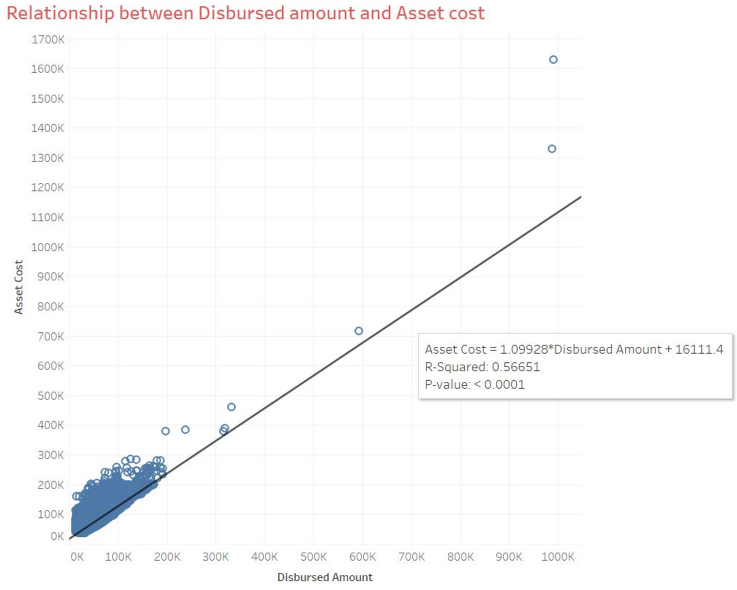
Assets Cost:



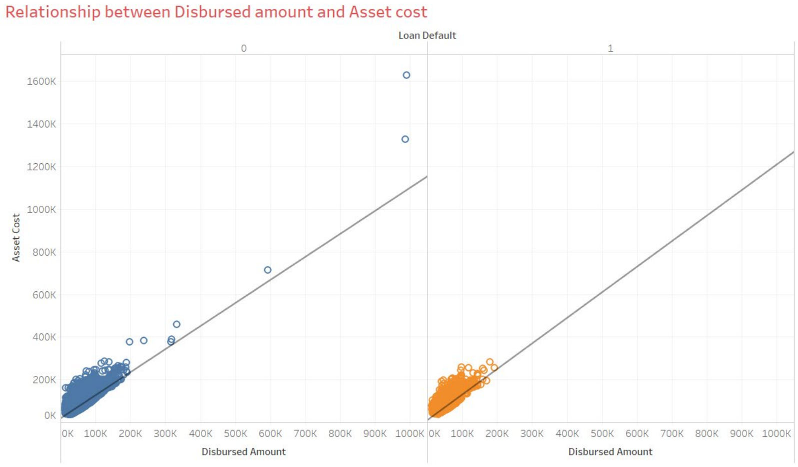


- Individuals in their twenties had the most asset costs (approximately 5billon US dollars) and were mostly salaried and self-employed.
- We calculated the percentage of loan defaulters and non-defaulters by their age groups. We observed that unemployed individuals in their fifties had the most non defaulters (with 86.55% of them who paid back their loans), It is possible that this happened in real life and it could be a mistake in the data since we classified the null values we had as unemployed. What makes more is if they didn't take so much in loans to begin with i.e. the asset for which they took the loans was not so expensive they could not payback.
- We can also see that the percentage of people in each age group and salary type who defaulted on loans were almost the same, they ranged from 16% to 26% approximately.

Disbursed Amount:



- From the scatter plot shows us a positive relationship between disbursed amount and asset cost. The higher your asset cost the more the bank loans you. We also observed that this is not the case all the time as some who had asset costs between 100k and 200k were loaned as low as 10k to 80k. We can also identify outliers in the data.



Relationship between Disbursed amount and Asset cost

Feature Selection

The data has 33 features after the initial cleaning. In Order to prevent overfitting and removal of multiple correlated features that would give repetitive information we needed to select the models that would contribute most to the predictive power of the model and eliminate the rest

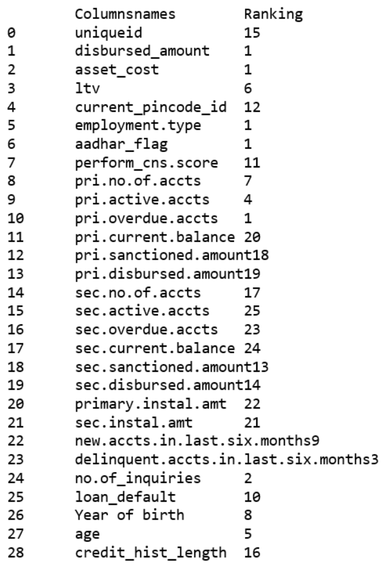
We tried two methods for feature selection

1. Backward Elimination Method

Backward elimination method removes the least important features in the dataset in a stepwise manner. The backward elimination has selected 17 features. We need to filter the number of features to reduce the complexity of the model. We can employ a feature selection method to rank the features according to importance. This didn't really help choose the best features because the p-values are close, so we tried the Recursive Feature Elimination Method.

2. Recursive Feature Elimination Method

We did a feature selection by ranking them using the recursive feature elimination method. Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coef_attribute or through a feature_importances_ attribute. Then, the least important features are pruned from the current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached. The base classifier used was logistic regression



```
        Columnsnames        Ranking
0       uniqueid            15
1       disbursed_amount    1
2       asset_cost          1
3       ltv                 6
4       current_pincode_id  12
5       employment.type     1
6       aadhar_flag         1
7       perform_cns.score   11
8       pri.no.of.accts     7
9       pri.active.accts    4
10      pri.overdue.accts   1
11      pri.current.balance 20
12      pri.sanctioned.amount18
13      pri.disbursed.amount19
14      sec.no.of.accts     17
15      sec.active.accts    25
16      sec.overdue.accts   23
17      sec.current.balance 24
18      sec.sanctioned.amount13
19      sec.disbursed.amount14
20      primary.instal.amt  22
21      sec.instal.amt      21
22      new.accts.in.last.six.months9
23      delinquent.accts.in.last.six.months3
24      no.of_inquiries     2
25      loan_default        10
26      Year of birth       8
27      age                 5
28      credit_hist_length  16
```

We pick the top 5 features so any feature with a score of 1 was selected. The features include;

1. disbursed_amount
2. asset_cost
3. employment.type
4. aadhar_flag
5. pri.overdue.accts

Analysis.

The analysis of the project was the most interesting part. Because we had unbalanced data, we tried to method to balance the data and compared the model accuracy with the unbalanced data using Logistic regression and Random forest classifier.

LOGISTIC REGRESSION

Logistic regression with the Unbalanced data:

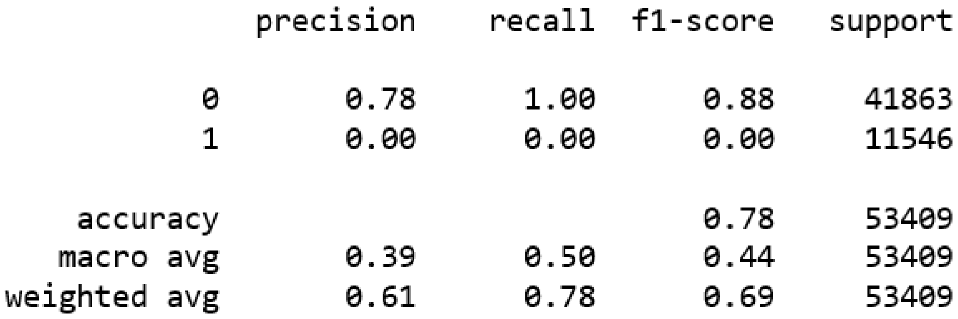
1- no default	167544
2- default	46092

After we noticed that the data, we had was not balanced the first thing we did was to experiment with the imbalanced data using the logistic regression. We had a 78% accuracy after we noticed that the data, we had was not balanced the first thing we did was to experiment with the imbalanced data using logistic regression. We had a 78% accuracy, but the data was not generalized just like we expected.

Confusion Matrix

41863- True Positive	0- False Negative
11546- False Positive	0- True Negative

From the confusion matrix, we can see the '1'- default categories were not represented at all so we balanced the data by upsampling and downsampling.



```
              precision    recall  f1-score   support

           0       0.78      1.00      0.88     41863
           1       0.00      0.00      0.00     11546

    accuracy                           0.78     53409
   macro avg       0.39      0.50      0.44     53409
weighted avg       0.61      0.78      0.69     53409
```

From the diagram above we can see the ability of the logistic regression to precisely label the defaulters and the non-defaulters the recall, f1 score, and the number of actual occurrences of each class in the dataset. We can see that the precision for the non-default class is much higher than that of the default, the recall (the ability of the model to find all positives) for the default was 0 and non-default was 1(perfect score) but the accuracy was 78%.

Precision = TP/TP+FP - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Recall = TP/TP+FN - Recall is the ratio of correctly predicted positive observations to all observations in the actual class.

F1 Score = 2*(Recall * Precision) / (Recall + Precision) - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

AUC -ROC Curve: A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as probability of false alarm and can be calculated as (1 − specificity).

The AUC can be used to compare the performance of two or more classifiers. A single threshold can be selected and the classifiers' performance at that point compared, or the overall performance can be compared by considering the AUC.

The AUC-ROC metric for logistic regression of unbalanced data was 0.59 which is not a good score

Balancing Data:

Balancing the data using upsampling the minority class:

0-no default	167544
1-default	46092

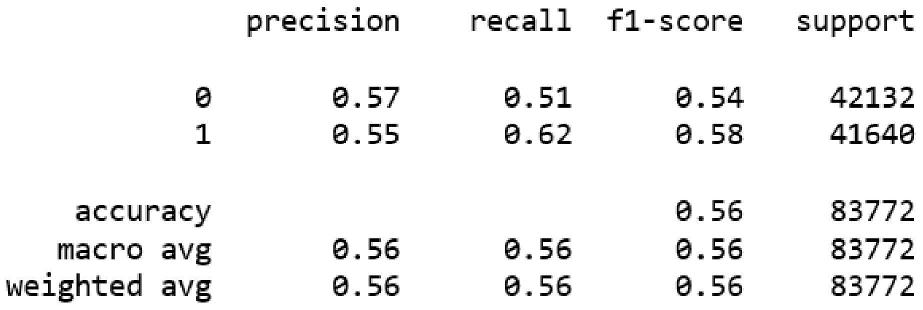
Upsampling the minority class, is a method matches the minority class which is the default in this case to match the non-default majority. We used the Logistic regression to fit the data and predict the results. We got a 56% accuracy.

1- no default	167544
2- default	167544

Confusion Matrix

21329 - True Positive	20803- False Negative
15981- False Positive	25659- True Negative

From the confusion matrix, we can see that just like the accuracy score implied the accurate predictions and false predictions are almost the same but both classes were well represented, but we can note that the model got more rights/accurate(True Positive, True Negative) than wrong (False Negative, False Positive). This accuracy should be better if we include more explanatory variables.



```
              precision    recall  f1-score   support

           0       0.57      0.51      0.54     42132
           1       0.55      0.62      0.58     41640

    accuracy                           0.56     83772
   macro avg       0.56      0.56      0.56     83772
weighted avg       0.56      0.56      0.56     83772
```

After balancing the data, although the accuracy of the model was 56% the report of precision for each class, recall, f1-score and the data represented is much better than the logistic regression with unbalanced data.

ROC Score: 0.59

Balancing the data using downsampling majority class:

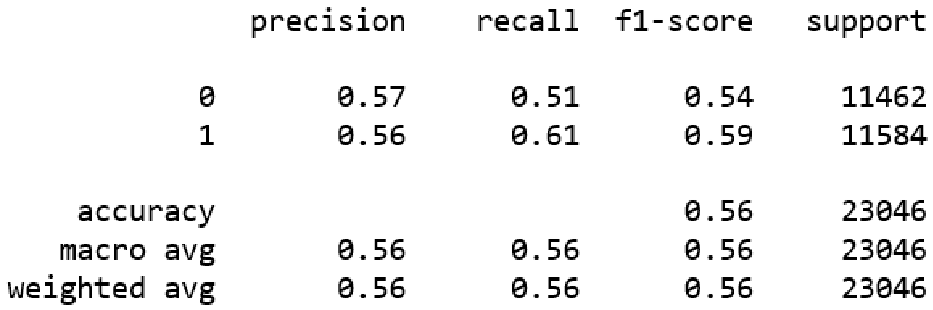
Downsampling the majority class, is a method matches the majority class which is the non-default in this case to match the default minority. We got a 56% accuracy.

1- no default	46092
2- default	46092

Confusion Matrix:

5833 - True Positive	5629- False Negative
4464- False Positive	7120- True Negative

Just like was noted in the up-sampling method, balancing the data didn't improve the model accuracy but if more explanatory variables are added that might change or if another model is used. But we can see that the model did better predicting the rights than wrongs.



```
              precision    recall  f1-score   support

           0       0.57      0.51      0.54     11462
           1       0.56      0.61      0.59     11584

    accuracy                           0.56     23046
   macro avg       0.56      0.56      0.56     23046
weighted avg       0.56      0.56      0.56     23046
```

For the downsampling method, the precision for defaulters was better by 0.01 than the up-sampling method, the recall and f1-scores are almost the same.

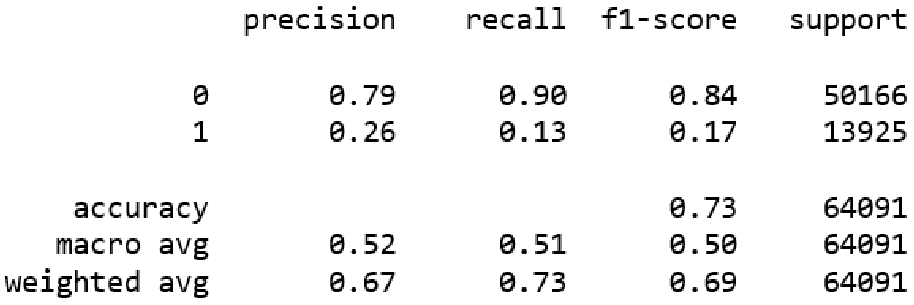
ROC: 0.59

RANDOM FOREST CLASSIFIER

Random Forest Classifier with Unbalanced Data:

The second model we tried was the random forest classifier which gave us a 72.8% accuracy, but it accurately classified the True Negatives (defaulters) badly, you can see that in the table below. The good thing is that each class was represented in this model as represented by the ROC score

44984 - True Positive	5182- False Negative
12122- False Positive	1803- True Negative



```
              precision    recall  f1-score   support

           0       0.79      0.90      0.84     50166
           1       0.26      0.13      0.17     13925

    accuracy                           0.73     64091
   macro avg       0.52      0.51      0.50     64091
weighted avg       0.67      0.73      0.69     64091
```

ROC Score :0.9890

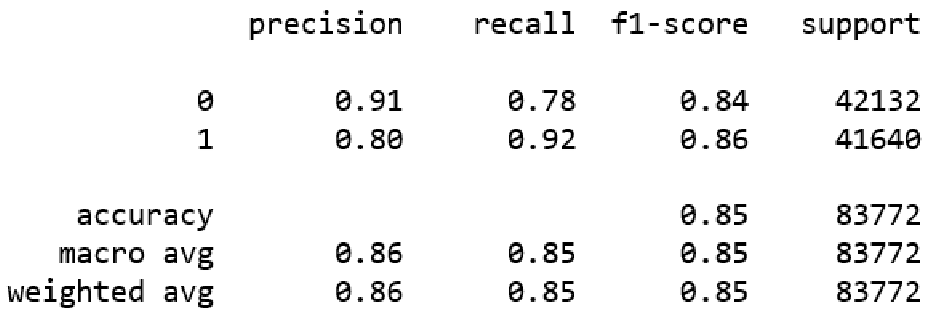
Balancing the Data:

Up Sampling:

32671 - True Positive	9461 - False Negative

3259 - False Positive	38381- True Negative

With the random forest classifier, the upsampling performed best with an 84% accuracy and the True positives and True Negatives in the confusion matrix are the best so far.



```
              precision    recall  f1-score   support

           0       0.91      0.78      0.84     42132
           1       0.80      0.92      0.86     41640

    accuracy                           0.85     83772
   macro avg       0.86      0.85      0.85     83772
weighted avg       0.86      0.85      0.85     83772
```

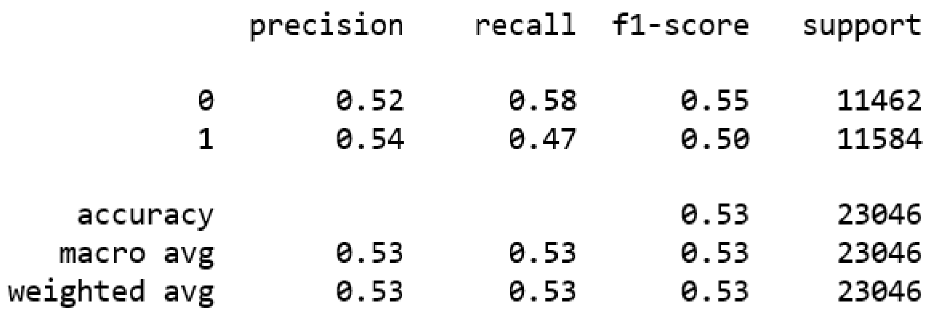
The precision for the default class is 0.85 and for the non-default is 0.91. The recall - ratio of all the model classified right for the default class was 0.92 and for the non-default class it was a 0.78

ROC SCORE: 0.997922665856164

Both classes are well represented

Down Sampling:

6707 - True Positive	4755 - False Negative
6083 - False Positive	5501- True Negative



```
              precision    recall  f1-score   support

           0       0.52      0.58      0.55     11462
           1       0.54      0.47      0.50     11584

    accuracy                           0.53     23046
   macro avg       0.53      0.53      0.53     23046
weighted avg       0.53      0.53      0.53     23046
```

The downsampling accuracy score of 53% different from the up-sampling method. The recall - ratio of all the model classified right for the default class was 0.47 and for the non-default class, it was 0.58.

ROC_AUC Score :0.989080693963751

Both classes are well represented

Conclusion:

Proposed Model: The best model this project is the Random Forest classifier up-sampling with gives an 85% accuracy.

Challenges: The major challenge we had during the analysis was the problem of unbalanced data.

Business objective accomplished: This model can be used for predicting whether an applicant is at risk of defaulting or not by credit agencies and banks

Future work : Going forward we can try more ensemble learning methods that are not sensitive to a class imbalance in the data and can get a better prediction or accuracy score.

References:

1. Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures

   https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#:~:targetText=High%20precision%20relates%20to%20the,precision%20which%20is%20pretty%20good.&targetText=Recall%20(Sensitivity)%20%2D%20Recall%20is,observations%20in%20actual%20class%20%2D%20yes.&targetText=Accuracy%20works%20best%20if%20false%20positives%20and%20false%20negatives%20have%20similar%20cost

2. What Factors are Taken Into Account to Quantify Credit Risk?

   https://www.investopedia.com/ask/answers/022415/what-factors-are-taken-account-quantify-credit-risk.asp

3. Sklearn.feature_selection.RFE

   https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html