Vehicle Loan Default Prediction

Objective:

To build a model that is capable of predicting whether a person will default on his/her first EMI payment of vehicle loan.

Client/End User:

The end user Larsen & Toubro Financial Services (LTFS) will have a deeper insight on decision making on sanctioning loan for a particular applicant. Such as the result given by the model will give more confidence when deciding.

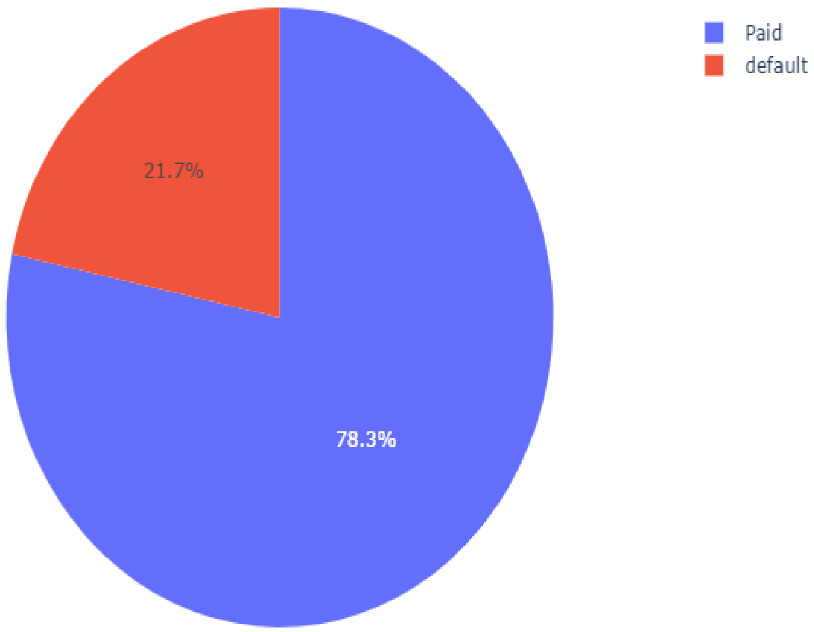
Data:

We use LTFS's own data which they used to conduct a competition in 2019.The dataset contains about 53000 entries with features such as loan amount, asset cost, customer employment status etc. The dataset contains 40 independent columns with 1 dependent column which 0 or 1 as value. (0-Paid, 1-Default).

Workflow outline:

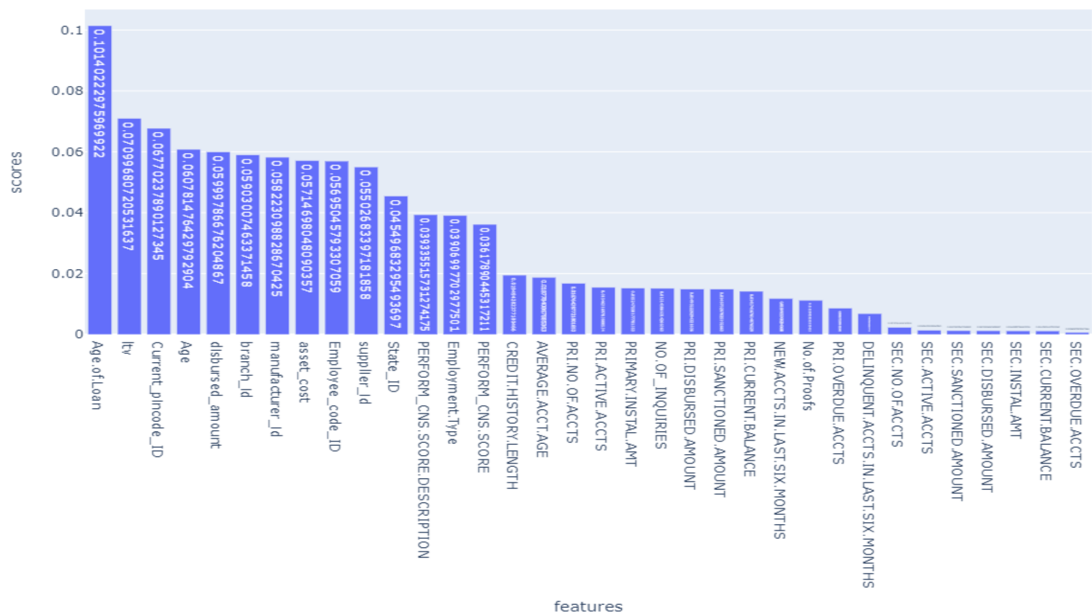
- **Data Wrangling**:
    - Transformed data type for columns like DOB, DisbursalDate into proper date strings.
    - Handled missing values in Employment Type column. NaN were replaced with 'unknown' string.
    - Employment type, PERFORM_CNS.SCORE.DESCRIPTION columns had string values which were mapped to integers.
    - AVERAGE.ACCT.AGE,CREDIT.HISTORY.LENGTH columns had values like '2 yrs 4months',replaced them with float values like 2.4 with a custom function yrscalc.
    - lables column was created out of loan_default column for easy human interpretation.

- **Feature Engineering/Selection:**
    - Proofs columns were combined (summed up) to create a new feature called No.of.Proofs.
    - The dataset faced imbalance class issue with about 78%- Paid and 22% Default class. To handle it SMOTE from imblearn was used to overcome it.

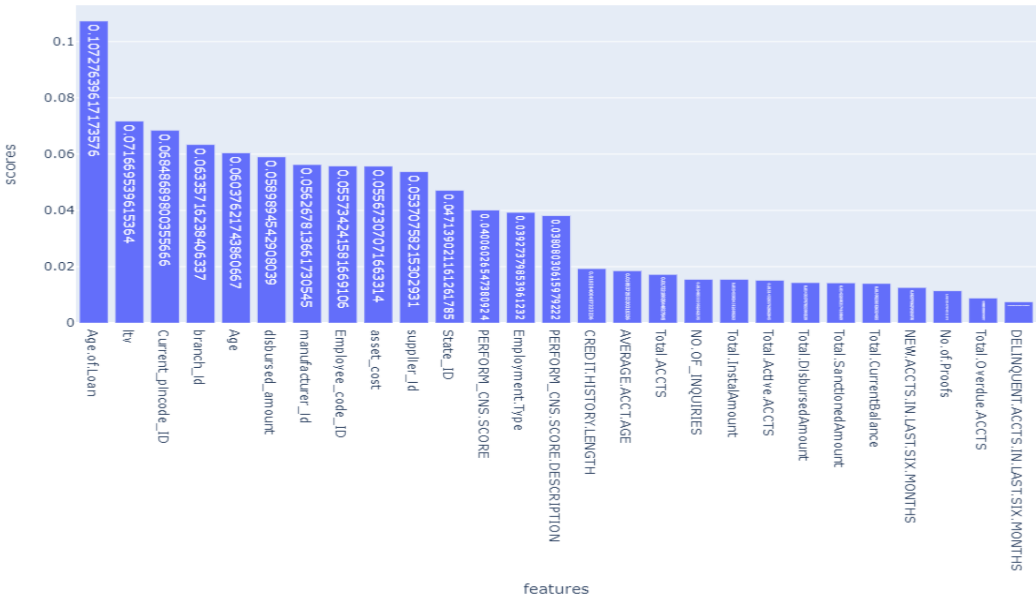


- Using ExtraTressClassifer to obtain feature importance scores it was found that secondary account related features did not have much significance and hence primary and secondary account features were clubbed.

Before clubbing



After clubbing



Final Score list

	features	scores
18	Age.of.Loan	0.107276
2	ltv	0.0716695
6	Current_pincode_ID	0.0684869
3	branch_id	0.0633572
17	Age	0.0603762
0	disbursed_amount	0.0589895
5	manufacturer_id	0.0562678
9	Employee_code_ID	0.0557342
1	asset_cost	0.0556731
4	supplier_id	0.0537076
8	State_ID	0.047139
10	PERFORM_CNS.SCORE	0.0400603
7	Employment.Type	0.0392738
11	PERFORM_CNS.SCORE.DESCRIPTION	0.0380803
15	CREDIT.HISTORY.LENGTH	0.019284
14	AVERAGE.ACCT.AGE	0.0185373
20	Total.ACCTS	0.0172219
16	NO.OF_INQUIRIES	0.0154851
26	Total.InstalAmount	0.015455
21	Total.Active.ACCTS	0.0151433
25	Total.DisbursedAmount	0.0143058
24	Total.SanctionedAmount	0.0142269
23	Total.CurrentBalance	0.0139923
12	NEW.ACCTS.IN.LAST.SIX.MONTHS	0.0125794
19	No.of.Proofs	0.0113715
22	Total.Overdue.ACCTS	0.00883256
13	DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS	0.00747327

- **Model building**
  - Using GridSearchCV, the models – RandomForestClassifier, GradientBoostingClassifier were tuned and the best scores were obtained.
  - Tuned parameters for Gradient Boosting are :
    - loss
    - learning_rate
    - n_estimators

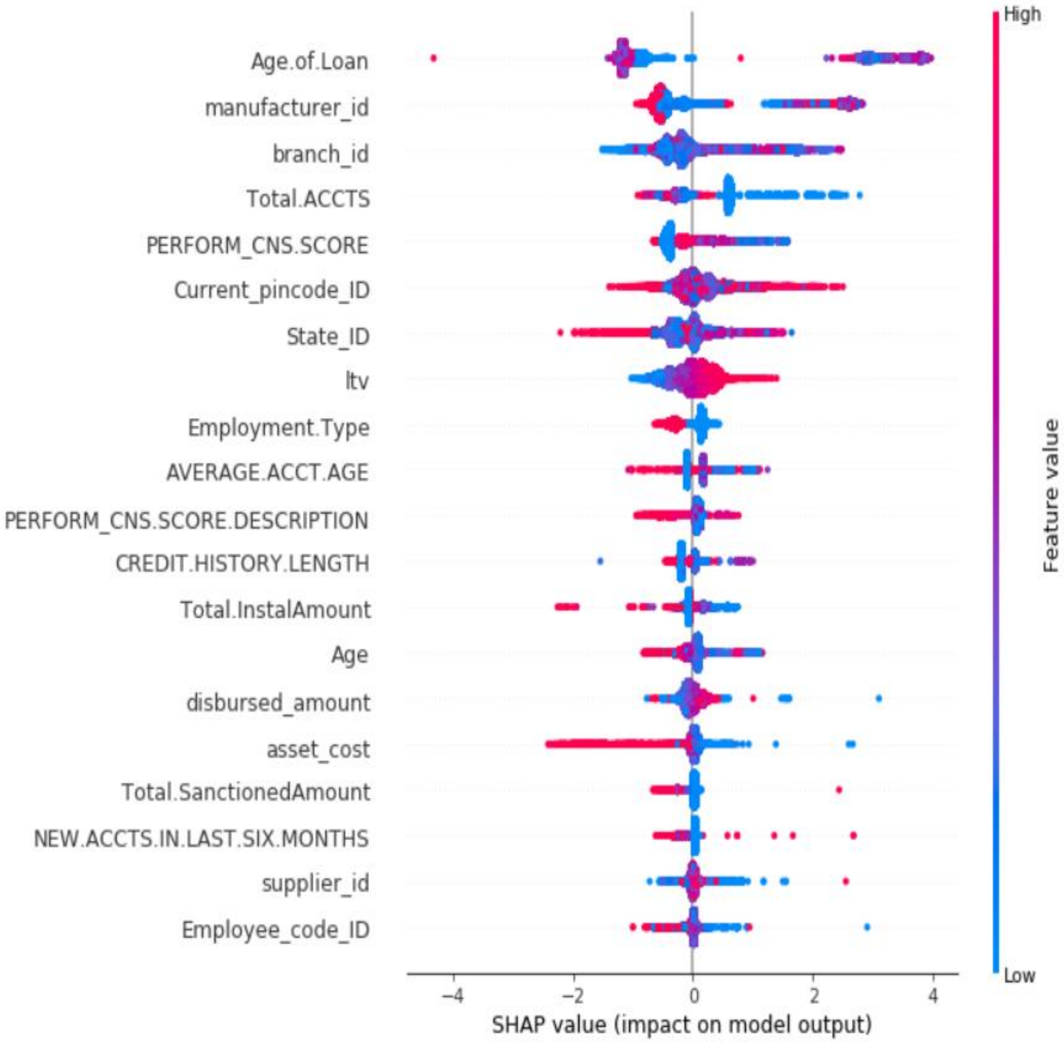
    max_depth here was dropped due to computational limits.
  - Tuned parameters for Random Forest are :
    - n_estimators
    - max_depth
    - criterion
  - precision and recall are the metrics upon which the model is evaluated.
  - A custom function was created that had the code for GridSearch which returns best params as well best best score.
  - Obtained best params for Gradient Boosting:
    - loss : deviance
    - learning_rate : 0.5
    - n_estimators : 150
    - best score : 0.902
  - Obtained best params for Random Forest:
    - n_estimators : 650
    - max_depth : 15
    - criterion : gini
    - best score : 0.823
  - With the above result Gradient boosting was chosen for its significantly better performance over Random forest and tuned accordingly.
  - Below is the classification report generated after prediction:

	precision	recall	f1-score	support
Paid	0.77	0.96	0.86	45745
Default	0.95	0.72	0.82	45745
accuracy			0.84	91490
macro avg	0.86	0.84	0.84	91490
weighted avg	0.86	0.84	0.84	91490

  - As we can see the model obtained is fairly good in predicting the loan defaults. Further tuning and exploration will definitely lead to better model.

- **Model Interpretation:**
    - While we work towards achieving a high performing model, its equally important to understand why/how the model achieved the desired output/performance.
    - We use SHAP library to do a basic analysis of how each feature contributed towards the model performance.
    - Summary plot from SHAP is used to get an overall feature impact towards the model.

Summary Plot



    - We will look at 5 features here:

1.Age.of.Loan - This one has 2 extremes where certain high and low values either decrease or increase prediction. A manual investiga -tion is needed to find the cause.

2.manufacturer_id - Here mainly high values decrease prediction while low and mid-ranged values increase prediction.

3.branch_id -Lower values either tend to decrease prediction or have a small positive effect while high and mid-ranged values increase prediction.

4.Total.ACCTS - Lower values highly tend to increase the prediction of the model.

5.PERFORM_CNS_SCORE - Certain low values and most high values decrease prediction while most lower values tend to increase prediction.