Objective:

Our Objective is Loan Delinquency Prediction. Predicting the Loan Delinquency i.e if the loan gets repaid or not based on the features given to us like unpaid principal balance, borrower credit score, payment date, etc.

Data Preprocessing:

- There is no missing values, corresponding to any of the features.

- There are 5 categorical variables, indicating that a tree based model should work fine in this case.

- There is a huge imbalance between the classes, with 99.67% values belonging to the majority class- 0.

- Many of the features were highly correlated, to deal with this following steps were taken:

- no_of_borrowers was dropped as it was highly correlated with co-borrower_credit_score (PCC-0.996).
- monthly payments are highly correlated so months showing high correlation are clubbed by taking the sum.

Feature Engineering:

- As loan_origination_date and first_payment_date don't make a sense as a feature, so a new feature was created by taking the difference between the two to give a more problem oriented feature.

- Co-borrower credit score and borrower credit score are converted into ordinal categorical variable.

Modelling:

- 3 approaches were considered to deal with imbalanced dataset:

1.) SMOTE with ratio of minority:majoity class = 0.6, considering the huge imbalance between the datasets.

2.) SMOTE with ratio of majority:minority class = 1

3.) Random Undersampling and Oversampling with ratio of minorty:majority = 0.6

- I followed Kitchen-Sink Approach i.e. build all basic models like KNN,Naive Bayes, SVM , Logistic Regression etc and try all the ensembles models on . I used 10% of the data as test set and validated the result on that.

- In this first I tried boosting models like random forest, xgboost , lightgbm and from observation.

- I found that SMOTE model with ratio=0.6 is giving better scores. After proper hyperparameter tuning by GridSearch of cv=4, XGBoost was giving the best accuracy.

- Also to deal with the tradeoff between precision and recall, I tried to decrease the threshold to 0.45 and 0.4, but it decreased the overall F1 score and also the leaderboard score. So I proceeded with the normal threshold.