LOAN PREDICTION ANALYSIS
Team-25

TA: Christopher
Team Members: Derrie Susan Varghese
Sri Naga Chandra Vivek Garimella
Varun Jagadeesh

1. Problem Description

Banks and other housing loan agencies follow a rigorous process to decide whether a candidate is eligible for a loan. To ensure loan defaulting does not happen too often, a lot of factors such as their credit history, income, purpose of loan etc. are to be taken into consideration in the process of loan approval. In order to solve this problem, we make use of the mortgage loan application data for the year 2018 to predict whether an applicant is eligible for the loan applied or not.

This is a binary classification problem and our analytical strategy is to compare the performance of classification algorithms namely Adaboost, XGBoost, Logistic Regression, Naive Bayes, Random Forest and Linear Discriminant Analysis on the given dataset. These Classification algorithms will help us to analyze a candidate and evaluate his/her eligibility for the requested loan.

2. Dataset:

The dataset which we have selected is the National Level Loan Dataset which is collected by The Federal Financial Institutions Examination Council (FFIEC) under the Home Mortgage Disclosure Act (HMDA) for the year 2018. The dataset consists of about 15119651 data points and 99 features such as Income, loan_type, loan_amount, loan_term, property_value, etc. As the dataset is huge, we have sampled the data multiple times with different percent of the original data and evaluate. Figure-1 shows the snapshot of our dataset.

```
dataset.head()
```

	activity_year	lei	derived_msa_md	state_code	county_code	census_tract	conforming_loan_limit	derived_loan_product_type	derived_c
0	2018	RVDPPPGHCGZ40J4VQ731	17980	GA	13215	1.3215e+10	C	VA:First Lien	
1	2018	B4TYDEB6GKMZO031MB27	15804	NJ	34007	3.40076e+10	C	Conventional:First Lien	
2	2018	5493002QI2ILHHZH8D20	12420	TX	48453	4.8453e+10	C	FHA:First Lien	
3	2018	549300BRJZYHYKT4BJ84	35084	NJ	34027	3.4027e+10	C	Conventional:First Lien	
4	2018	B4TYDEB6GKMZO031MB27	29460	FL	12105	1.2105e+10	C	Conventional:Subordinate Lien	

5 rows × 99 columns

Figure-1 snapshot of dataset.

2.1 Exploratory Data Analysis

Performed Exploratory Data Analysis to find key insights from the data. From Figure 2 we can see that applicants with income to debt ratio greater than 60% faced highest number of rejects. Figure 3 shows that applicants with age greater than 74 had highest rejection rate whereas applicants of age group 25-34 had highest approval rate. Figure 4 shows that loan applied for the purpose of home improvement had highest rejection rate whereas applied for the purpose of home purchase had highest approval rate. Figure 5 infers that credit history and debt to income ratio are main causes for rejection of loan applications.

Figure -2

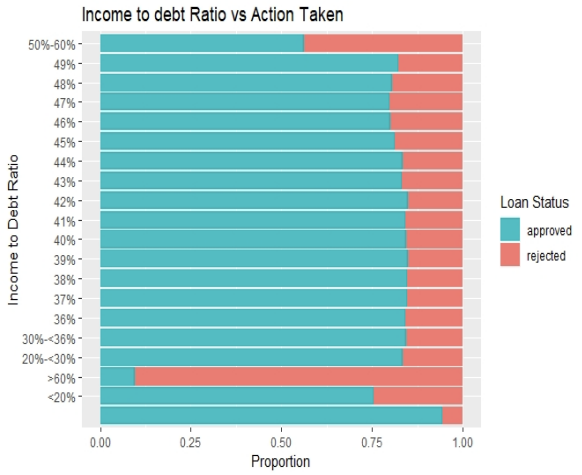


Figure-3

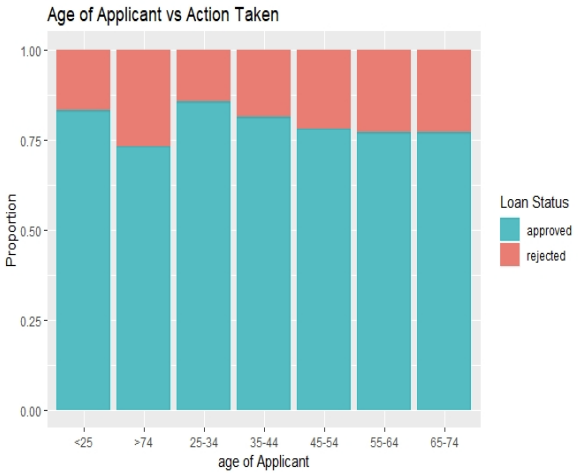


Figure-4

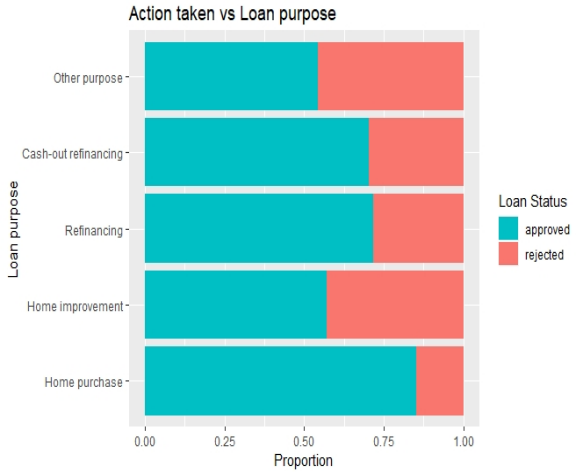
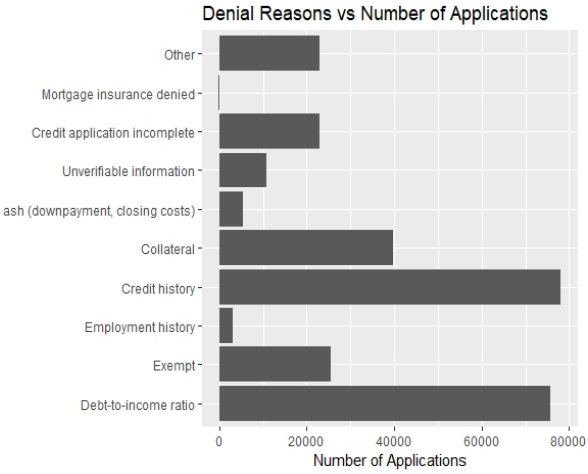


Figure-5



3. Approach and Methodology:

Our target variable is *action_taken* and it contains eight values, among which "Loan originated" and "Application approved but not accepted" represented those applications which were approved and "Application denied" & "Pre-Approval request denied" denoted those that were rejected. We have considered application approved (1) and denied (0) as the binary values and dropped the rows which contained the rest of the four values in *action_taken.* The overall workflow of the project is as shown in Figure -6

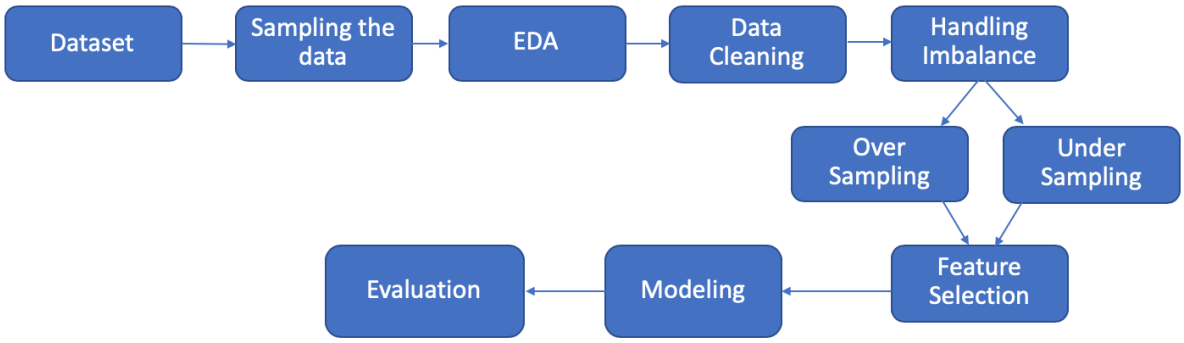


Figure -6 Overall Workflow

3.1 Data Preprocessing:

Data Cleaning: There were 30 columns with more than 90% missing values. These columns were dropped because these features cannot be used for modelling since majority of the observation were null values.

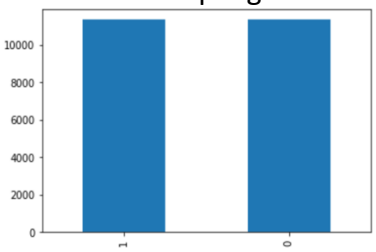
The dataset contains many features such as *interest_rate*, *denial_reason* etc. which were dependent on our target variable,ie. these observations had values only if the loan was either approved or denied. Also, these features were rated with high importance using Random Forest and thus it was necessary to remove these columns. Finally, we ended up having 48 features in the dataset. Additionally, as the dataset contained a wide range of values, it was standardized.

Encoding categorical variable: One hot encoding was performed for multiple categorical variables so that the dataset will be better processed. Classification algorithms give better results when the categorical features are encoded.

Handling Imbalance: It is extremely important to handle the imbalance in the dataset before training the model, otherwise incorrect results are obtained during the modeling process. From Figure-7 we can see that the dataset is highly imbalanced and thus we performed both

under sampling and oversampling techniques on the dataset and we trained all the classification models using both datasets to evaluate which is performing better with respect to our dataset.

Under Sampling



Figure–8

Original Distribution

```
dataset['action_taken'].value_counts()
```

```
1    1678167
0      11337
Name: action_taken, dtype: int64
```

Figure-7

Over Sampling

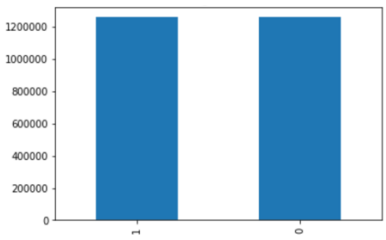


Figure-9

Under sampling: This is one way to handle imbalance in the dataset in which we remove some of the majority class and balance it to the minority class such that both are even. We randomly select samples from the data set and form a new sample so that both the minority and majority class will have the same number of observations. Figure-8 shows the dataset once under sampling is performed, we can see that it contains an equal number of loans approved and rejected observations.

Oversampling: This is another way to handle imbalance in the dataset in which we increase the number of minority class records such that both minority and majority class contain equal number of observations. To implement oversampling SMOTE (Synthetic Minority Oversampling Technique) algorithm was performed. SMOTE algorithm does not randomly duplicate observations, instead it takes the samples of the minority class and generates new synthetic examples that combines the features of the target case with features of its neighbors using KNN. Figure-9 shows the balanced data set once oversampling was done using SMOTE.

Feature Selection: This is one of the core concepts in machine learning which hugely impacts the performance of model. There is a chance of overfitting if features selection is not done properly and this can affect the accuracy of the model.
There are multiple techniques to select features. The features selection technique which we have used in this project are:

1. Feature selection using Lasso Regression
2. Feature importance using Random Forest Classifier

Once we got the feature importance from Lasso and Random Forest classifier, we compared some of the top features from both the methods and we could see that most of the top features were similar. We also used these top features which were generated from these algorithms to

see how our model will perform. Figure-10 and 11 shows the list of important features which we obtained from random forest algorithm and Lasso regression.

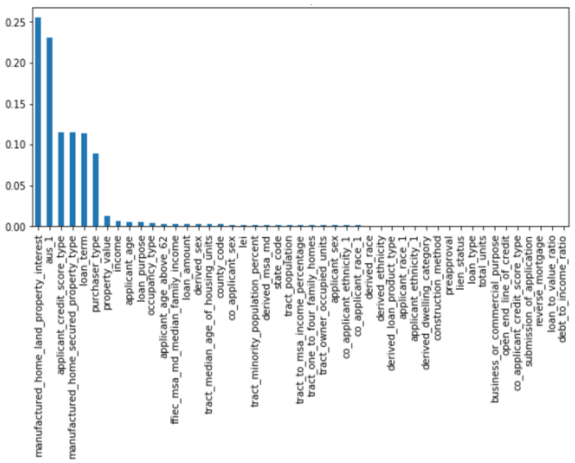


Figure-10

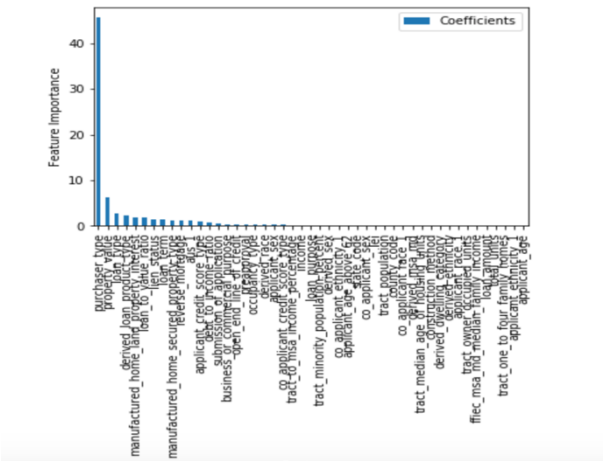


Figure-11

3.2 Modeling:

Prediction of eligibility of a candidate is a binary classification problem. Before building the classification models, we split the data into 75% training data and 25% test data. The following six classification models were trained and evaluated:

3.2.1 Logistic Regression

Logistic Regression is one of the basic classification models which uses a logistic function to predict the probability of a certain class. This is one of the most popular models which is used on categorical data, especially for binary response data.

3.2.2 Random Forest

Random Forest is an ensemble machine learning method that models by creating multiple decision trees and class with the majority votes becomes the model's prediction. Random Forest is one of the top performing classification models in terms of accuracy and robustness.

3.2.3 Naïve Bayes

Naive Bayes is yet another common classifier based on Bayes theorem and considers all features to be independent and every feature is given equal weights while predicting the probability of each class.

3.2.4 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis is a dimensionality reduction technique which is implemented for classification problems. It is an efficient way to model the differences or separation in classes.

3.2.5 XGBoost

XGBoost is a modeling which has recently been dominating in the field of machine learning. It is an implementation of Gradient with increased speed and performance. XGBoost uses parallelization of tree structures while building the model.

3.2.6 AdaBoost

AdaBoost is a boosting algorithm which converts weak learner into a strong learner by giving more emphasis on misclassified samples. It is commonly used with Decision tree as the weak learner. Adaboost is known for its high performance.

4. Results and Evaluation:

Confusion matrix:

Confusion matrix is a table used for classification which shows the number of true negatives, true positives, false negatives and false positives. The confusion matrices of all our models are shown below

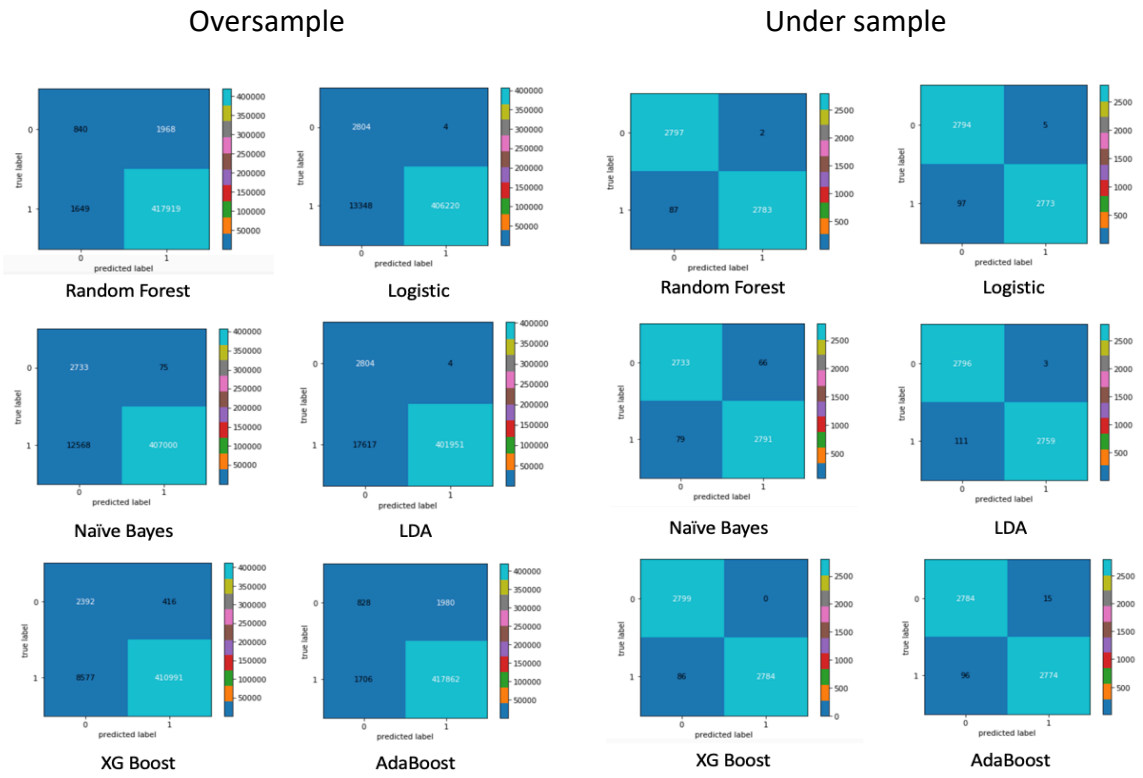


Figure – 12 Confusion Matrix for different models

Accuracy:
Accuracy is the ratio of correct predictions to the total number of observations. Even though it is a very important metric, it is not a good metric when it comes to imbalanced data. The accuracy scores were high for both under sampled and oversampled data because of the high percentage of true positives while predicting on the test data.

Model	Oversampling Accuracy	Under sampling Accuracy
Ada Boost	99.110981	98.041982
Random Forest	99.112165	98.430058
LDA	95.807763	97.989063
Logistic Regression	96.812792	98.200741
Naïve Bayes	96.963369	97.442229
XGBoost	97.812138	98.482977

Table-1

Accuracy = TP+TN/(TP+FP+FN+TN)

Precision:
Precision is defined as the proportion of the number of true positives to the sum of true positives and false positives. Precision takes false positive into consideration; thus, it is a very important metric when it comes to predicting whether a candidate is eligible for loan approval because it is important that our model does not wrongly misclassify candidates to be eligible for a loan when they actually are not. If the precision of the model is too low, then the model will approve ineligible candidates as eligible, which will be a huge loss for the loaning agency.

From Table-2, we can see that the under sampled data is showing higher precision for both minority and majority class. This is the case because the oversampled data uses synthetic examples to increase the proportion of minority class and these data points might not be a good representation of the actual data. Random Forest is performing the best in terms of precision of minority class (97.69%) and majority class for the under sampled data (99%), followed by Naive Bayes. When the number of estimators were increased, the precision increased. The values reported are for 100 estimators.

	Random Forest		Logistic Regression		Naïve Bayes		LDA		XG Boost		ADABoost	
	0	1	0	1	0	1	0	1	0	1	0	1
Over Sampling	0.337	0.995	0.173	0.999	0.179	0.999	0.137	0.999	0.218	0.998	0.326	0.995
Under Sampling	0.979	0.999	0.966	0.998	0.951	0.976	0.961	0.998	0.97	1	0.966	0.994

Table-2 Precision

Recall:

Recall is the ratio of the number of true positives to the sum of true positives and false negatives. Recall is the ability of the model to find all the data points of relevance in a dataset. From Table-3 we can see that, almost all the models are doing considerably well in terms of recall in minority classes for over sampled data, except for Random Forest (29.9%) and Adaboost(29.4%). Whereas, Logistic Regression and LDA are doing best in this case. In under sampled data XG Boost (100%) and Random Forest(99%) is performing best with a recall of for minority class

	Random Forest		Logistic Regression		Naïve Bayes		LDA		XG Boost		ADABoost	
	0	1	0	1	0	1	0	1	0	1	0	1
Over Sampling	0.299	0.996	0.998	0.968	0.973	0.970	0.998	0.958	0.851	0.979	0.294	0.995
Under Sampling	0.999	0.969	0.998	0.966	0.976	0.972	0.998	0.961	1.0	0.970	0.964	0.966

Table-3 Recall

ROC and Precision Recall curves:

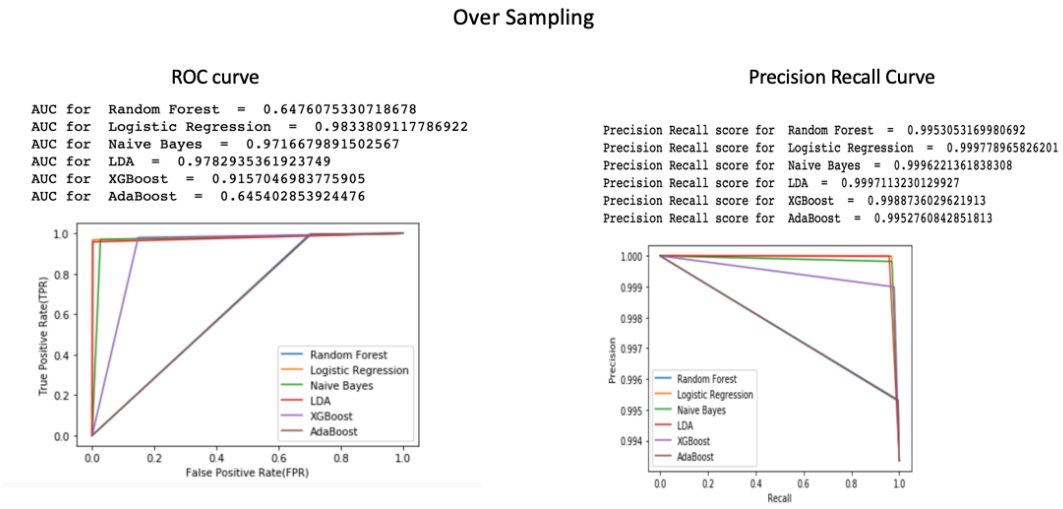


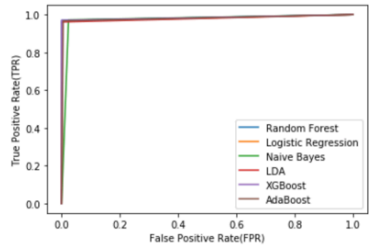
Figure- 13

ROC curves give us information on true positive rate (TPR) and false positive rate (FPR). ROC curve is not a good measure for evaluation for unbalanced dataset because it FPR and TPR are maximized since minority class have few observations. Hence, we have used precision-recall curves and calculated average precision-recall score for each of the models. We can see that all models have a good precision-recall score. From Figure-13 Adaboost, XGBoost, LDA and Naive Bayes have a high precision-recall score of 99% in oversampled data. From Figure-14 XGBoost and Random Forest performed well in under sampled data with 98%.

Under Sampling

ROC curve

```
AUC for  Random Forest  =  0.9844859351211794
AUC for  Logistic Regression  =  0.9822078691618336
AUC for  Naive Bayes  =  0.9744470088247047
AUC for  LDA  =  0.9801261152253231
AUC for  XGBoost  =  0.9850174216027874
AUC for  AdaBoost  =  0.9805957329210409
```



Precision Recall Curve

```
Precision Recall score for  Random Forest  =  0.9824390544435836
Precision Recall score for  Logistic Regression  =  0.9782384078629561
Precision Recall score for  Naive Bayes  =  0.9712738298785305
Precision Recall score for  LDA  =  0.9736868646029523
Precision Recall score for  XGBoost  =  0.9824390544435836
Precision Recall score for  AdaBoost  =  0.9691492545909338
```

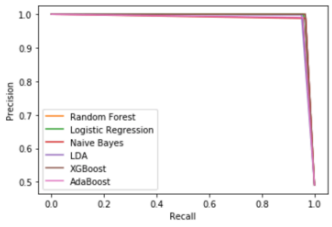


Figure-14

We trained models such as Random Forest, Logistic Regression and Ada Boost by using some of the top features that we got from Random Forest feature importance and Lasso regression. We were able to see an improvement in performance of the model in terms of precision and recall. ADA boost precision increased to 0.74 and random forest precision increased to 0.36 for oversampling data for minority class. Hence, we can say that better feature selection will give use good results.

5. Conclusion and Future work

In this project, we were trying to predict the eligibility of candidates for approval of a loan. While building models, domain knowledge on the loan application deemed to be very necessary; there were varied features in the dataset, and it was important to understand the meaning of each feature and obtain the correlation of these features with the target variable. Different evaluation metrics had to be used since the dataset was highly imbalanced.

In the future, while handling imbalanced data we can resample the data with different ratios of majority and minority class. For example, instead of 1:1 ratio where (approved: rejected) we can go with 1:2 or 1:3 and check how each of the model performs. Also, sampling is not the only method to tackle imbalance problem because by doing oversampling, we generate synthetic data and will compromise the model when we predict on the test data. Whereas, in down sampling we are losing out on a lot of observations and the model won't have enough data to train. Therefore, we can try k-cross validation to tackle this problem. Additionally, missing values in the data can be handled by different methods such as use mode for a categorical feature and regression imputation for other numerical variables.

6. References:

Link to Dataset:

https://ffiec.cfpb.gov/data-publication/snapshot-national-loan-level-dataset/2018

Link to Code:

https://github.com/varun-jagadeesh/Loan-Prediction-Analysis

https://www.ffiec.gov/hmda/pdf/2013guide.pdf
https://files.consumerfinance.gov/f/documents/cfpb_2018-mortgage-market-activity-trends_report.pdf
https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
https://jair.org/index.php/jair/article/view/10302