Home Credit Default Risk Kaggle Competition

Group Number - 16

Team :-

NagaJahnavi Dhulipalla
Shivani Chennoju
Shalini Kothuru
Bharath Chowdary Anumolu

Outline

- Team members and Photos
- Project Abstract
- Team and project meta information
- Project Description
- Neural Network
- Leakage
- Modelling Pipeline
- Four P’s
- Results and discussion of results
- Conclusion

Team members and photos - Team 16

Shalini Kothuru
skothuru@iu.edu

Bharath Chowdary Anumolu
banumolu@iu.edu

Shivani Chennoju
schennoj@iu.edu

NagaJahnavi Dhulipalla
ndhulipa@iu.edu

Project Abstract

Home Credit is a prominent developing market consumer financing specialist that has developed a platform that manages its core technology, product, and funding activities keeping in mind the local market needs. Their target market is underserved borrowers in the blue-collar and junior white-collar segments who have a steady source of income from their jobs or micro-businesses but are less likely to get loans from banks and other traditional lenders. It is vital for Financial Organizations to observe whether their loan applicants will be able to repay their loans. There are multiple parameters that need to be considered for optimally predicting if the client will be defaulting. The parameters that can be used are occupation, credit history, age, location, credit card usage, cash balance and others. We will thoroughly visualize and study these parameters before implementing them. Our goal is to make use of all the parameters to predict with the best possible efficacy which will aid financial organizations to make better decisions for their loan applicants.

To perform the Home Credit Default Risk project, we have extracted the data from the Kaggle competition 'Home Credit Risk Analysis' and performed Exploratory Data Analysis to understand and explore the data. Various visualizations were performed on most of the input features to the 'Target' variable to find the people at maximum risk. In the second phase of the project, the models built were overfitting. So, we have remodeled the old feature engineering on all the tables and added features such as AMT_DRAWING_ratio, DAYS_INSTALLMENTS_diff, and AMT_ANNUITY_ratio.

Project Abstract(Continued…….)

Data leakage was taken into consideration throughout the project to retain the efficiency of the project. Modeled a neural network model and implemented Multi_layer Perceptron with the help of Pytorch. A Tensor board was used to visualize and track the training and validation loss of the models. The Machine Learning Classifiers such as Logistic Regression, Random Forest, and XGBoost have performed better. The accuracies of our models improved compared to phase 3. Logistic Regression achieved a test accuracy of 91.1% and a test ROC_AUC score of 0.6855, whereas XGBoost has a better ROC_AUC score, which is 0.7207. We have also implemented Multi-Layer Perceptron and achieved a test accuracy of 68% and a ROC_AUC score of 0.705. A Kaggle submission was made using the best performing model XGBoost and obtained a score of 0.7255.

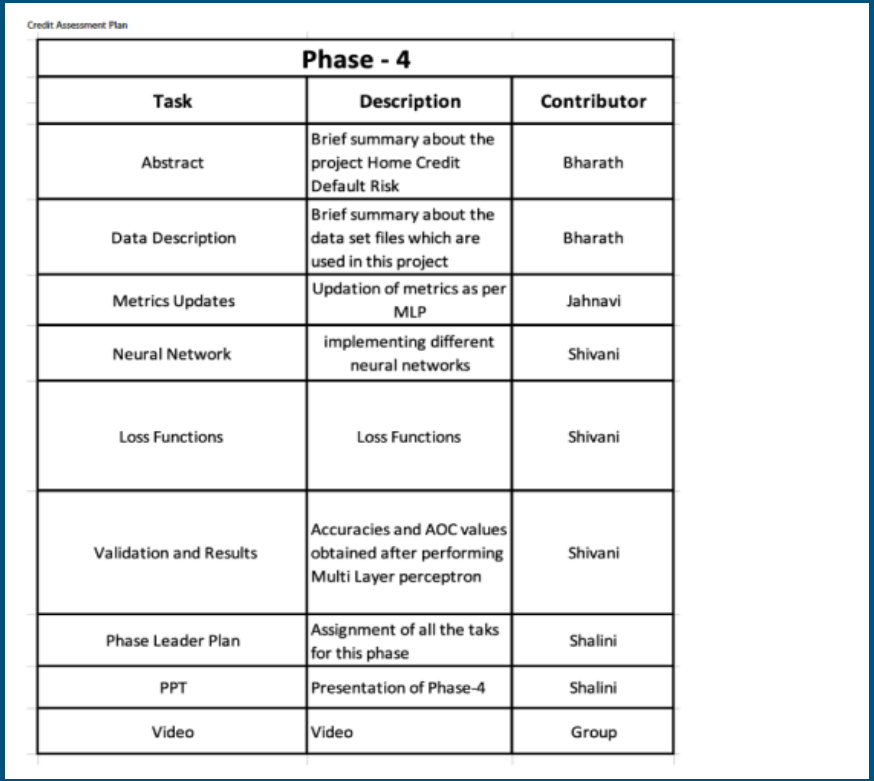
For Kaggle submission, we obtained a public score of 0.7158 and a private score of 0.7255.

Team and project meta information

Phase Leader Plan				
Phase	Phase - 1	Phase - 2	Phase - 3	Phase - 4
Phase Leader	Bharath	Jahnavi	Shivani	Shalini
Tasks	Abstract	Phase Leader Plan	Phase Leader Plan	Phase Leader Plan
	Data Description	Project Abstract	Metrics Updates	Metrics Updates
	Machine Algorithms	Project Description	Model Development	Neural Network Implementation
	Metrics	EDA	Future Engineering	Advanced model architectures
	Machine Learning Pipelines	Visual EDA	Hyperparameter tuning	Loss Functions
	Overall Proposal Appearance	Modelling pipelines	Additional future selection	Validation
	Phase Leader Plan	Results	esemble methods	Summary
	Credit Assessment Plan	Conclusion	Validation	Final Report
		Video	Video	Video
		PPT	PPT	PPT

Team and project meta information(Continued…)

Credit Assignment Plan



Credit Assessment Plan

Phase - 4		
Task	**Description**	**Contributor**
Abstract	Brief summary about the project Home Credit Default Risk	Bharath
Data Description	Brief summary about the data set files which are used in this project	Bharath
Metrics Updates	Updation of metrics as per MLP	Jahnavi
Neural Network	implementing different neural networks	Shivani
Loss Functions	Loss Functions	Shivani
Validation and Results	Accuracies and AOC values obtained after performing Multi Layer perceptron	Shivani
Phase Leader Plan	Assignment of all the taks for this phase	Shalini
PPT	Presentation of Phase-4	Shalini
Video	Video	Group

Project Description- Data Dictionary

There are seven different sources of data:

1. application_{train|test}.csv

- This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET).Static data for all applications. One row represents one loan in our data sample.

2. bureau.csv

- All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in our sample).
- For every loan in our sample, there are as many rows as number of credits the client had in Credit Bureau before the application date.

Project Description

3. bureau_balance.csv

- Monthly balances of previous credits in Credit Bureau.
- This table has one row for each month of history of every previous credit reported to Credit Bureau – i.e the table has (#loans in sample * # of relative previous credits * # of months where we have some history observable for the previous credits) rows.

4. POS_CASH_balance.csv

- Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
- This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample * # of relative previous credits * # of months in which we have some history observable for the previous credits) rows.

Project Description-Data Dictionary

5. credit_card_balance.csv

- Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
- This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample * # of relative previous credit cards * # of months where we have some history observable for the previous credit card) rows.

6. previous_application.csv

- All previous applications for Home Credit loans of clients who have loans in our sample.
- There is one row for each previous application related to loans in our data sample.

Project Description - Data Dictionary

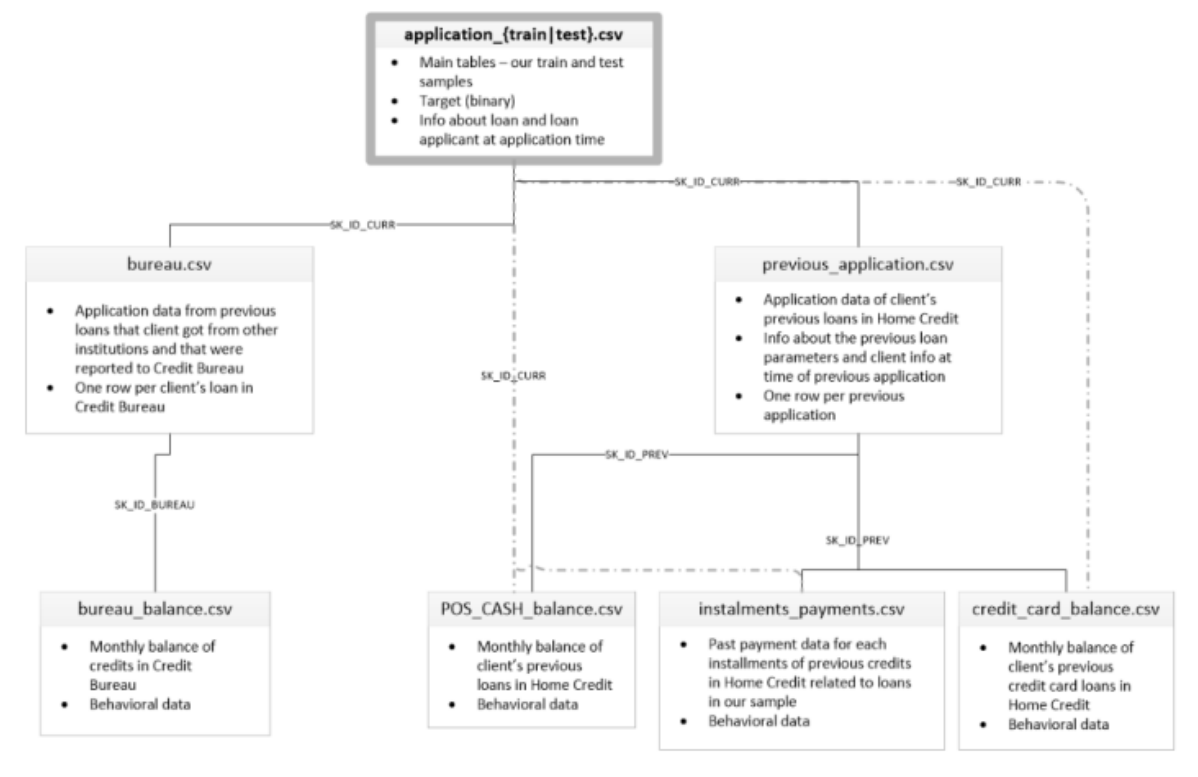
6. installments_payments.csv

- Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.
- There is a) one row for every payment that was made plus b) one row each for missed payment.
- One row is equivalent to one payment of one installment OR one installment corresponding to one payment of one previous Home Credit credit related to loans in our sample.

7. HomeCredit_columns_description.csv

- This file contains descriptions for the columns in the various data files.

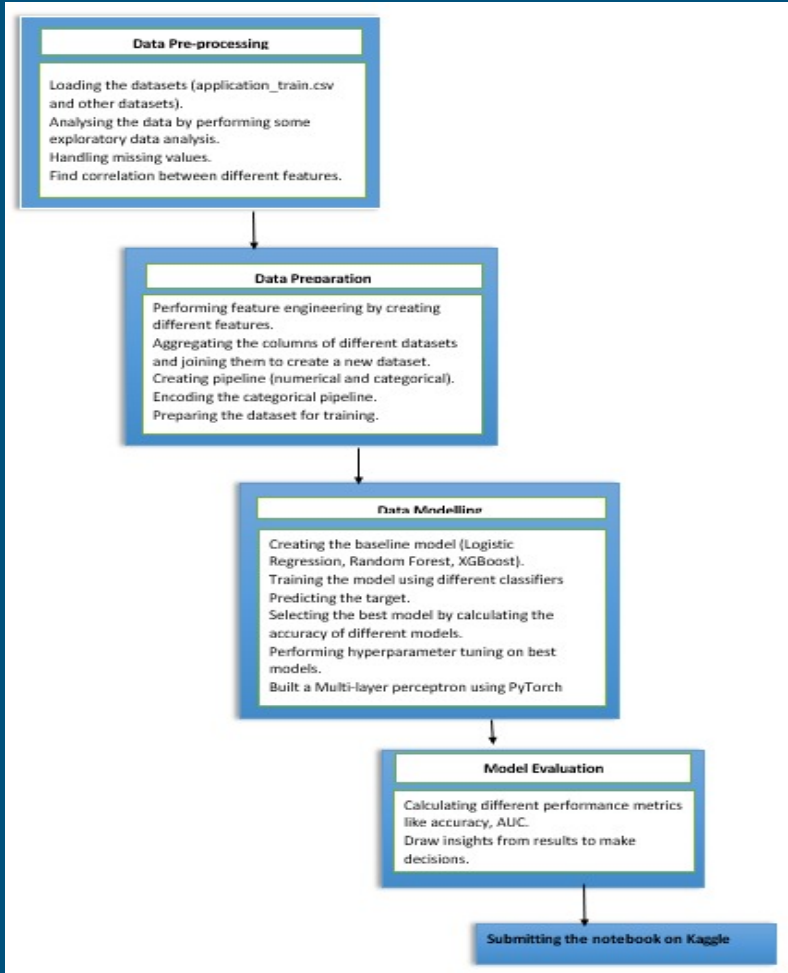
Project Description (Cont..)



Project Description - Tasks to be tackled

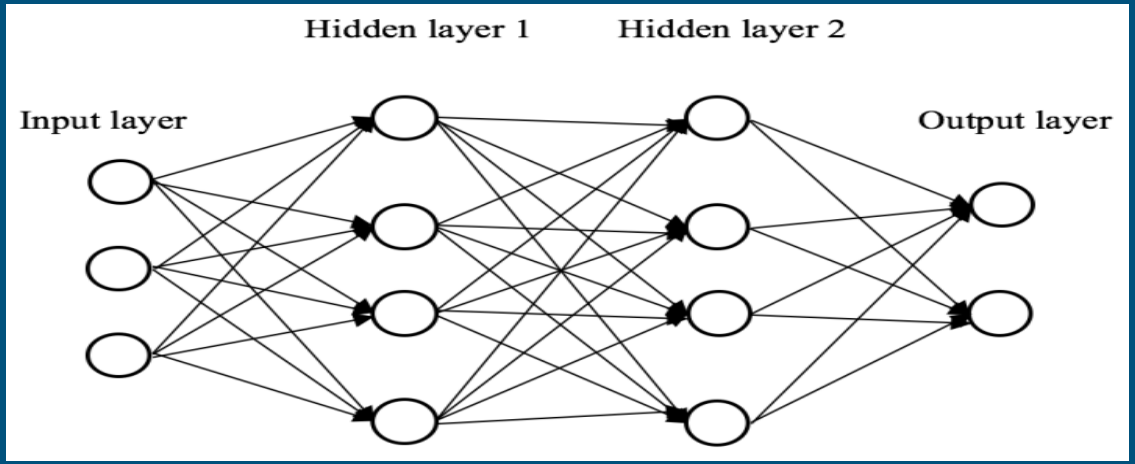
1. Building a Multi Layer Perceptron using PyTorch
2. Using Tensorboard to visualize the results of training in real-time.
3. We have increased number of features selected from 14(in Phase 3) to 50(in phase 4) and ran all the ML Classifiers.

Project Description - Workflow



Neural Networks

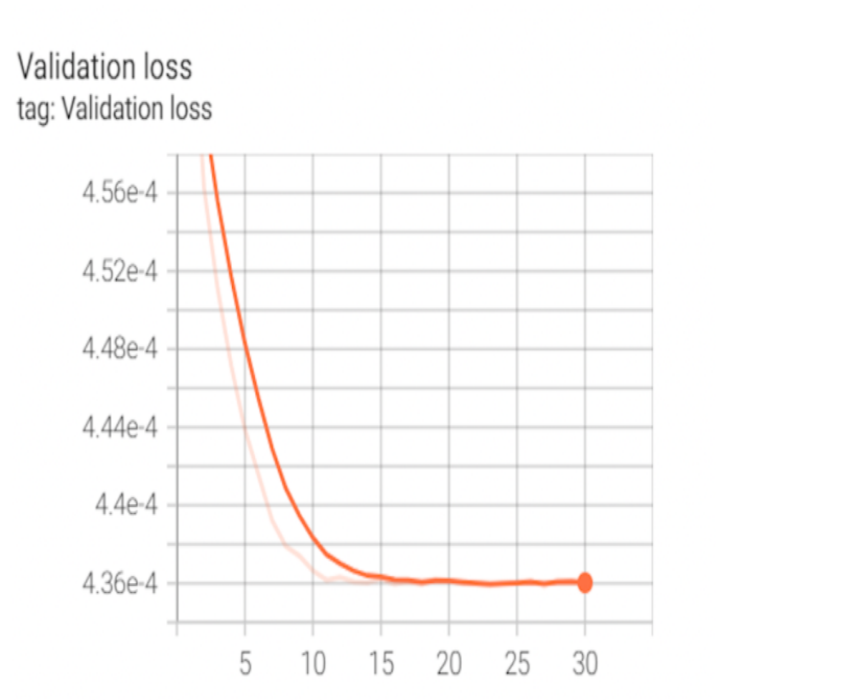
- We have built Multilayer perceptron using 3 hidden layers and [80,60,10]neurons with 40 input dimensions in the phase 4 of HCDR project.
- And also we implemented Multilayer perceptron using 2 hidden layers and 20 neurons with 166 input dimensions.
- We have acquired the AUC score - 0.705 and accuracy - 68%

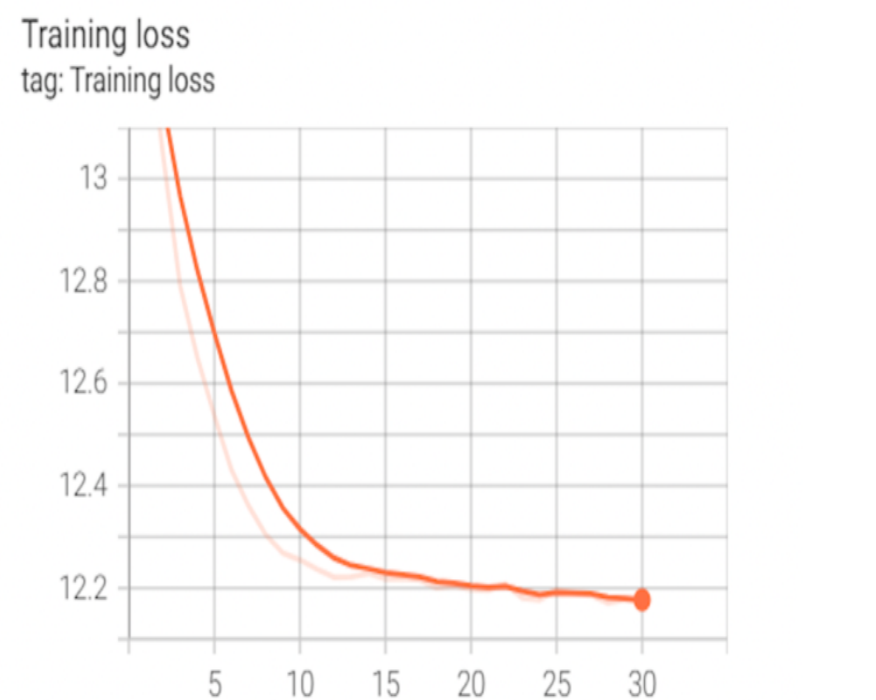


Neural Networks(Continued……)

- Neural network architecture in string form for 3 hidden layers is 80-Relu-BCE -60-Relu-BCE-10-Relu-BCE-1.
- Neural network architecture in string form for 2 hidden layers is 20-Relu-BCE -20-Relu-BCE-1.

Tensor board visualizations





Data Leakage

Data leakage (or leakage) happens when your training data contains information about the target, but similar data will not be available when the model is used for prediction. We have split the dataset into training and test data. We are removing missing values in our data by replacing them with mean. We scaled the data using StandardScaler. Because of these factors, there will not be any significant data leakage in our modeled pipelines.

Cardinal sins avoided: To avoid overfitting we have splitted our data into train and test. The test dataset is only available when we train the model on the train set and once the evaluation is done. We can observe that the accuracy for the train and test are very close which avoids the overfitting the model.

Modeling Pipelines

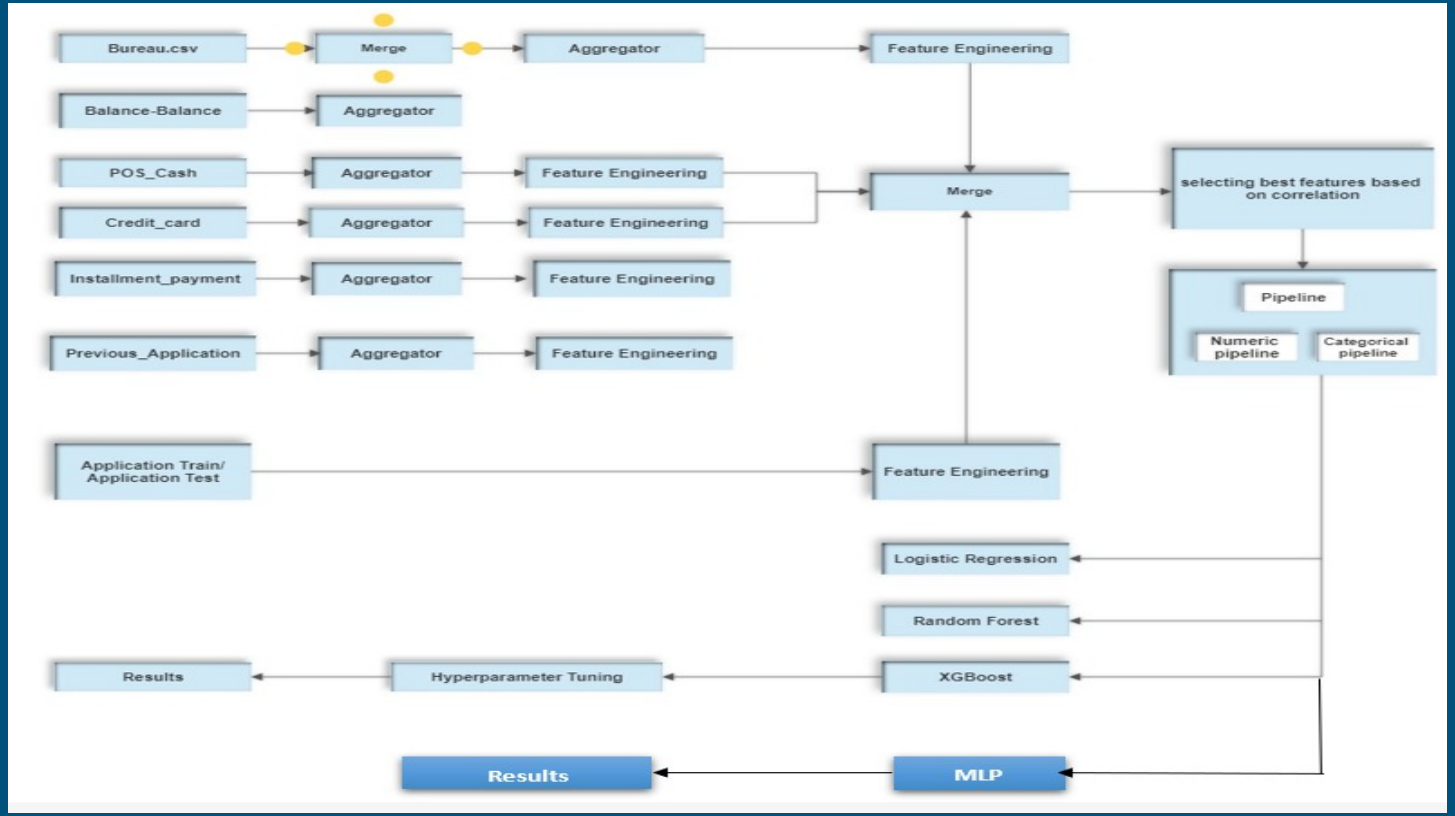
Families of input features: count of numerical features: 106(int64, float64) count of categorical features: 16(object)

The total number of input features: 122 input features.

We have trained four models:

- Logistic Classifier
- RandomForestClassifier
- XGBoost
- Multi Layer Perceptron

Modeling Pipelines



Loss functions

Loss Functions

Log Loss

Log loss is indictive of how close is the prediction probability is to the corresponding value/true value. The more the predicted probability diverges from the actual value, the higher is the log loss value.

[6]:
```
!pip install latexify-py
```

```
Requirement already satisfied: latexify-py in /Users/bharathchowdary/opt/anaconda3/lib/python3.9/site-packages (0.2.0)
Requirement already satisfied: dill>=0.3.2 in /Users/bharathchowdary/opt/anaconda3/lib/python3.9/site-packages (from latexify-py) (0.3.6)
```

[7]:
```
import latexify
@latexify.function(use_math_symbols=True)
def logLoss():
    return (-1 / N) * Sigma(_i**N) (y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i)))
logLoss
```

[7]:
$$\text{logLoss}() = \frac{-1}{N} \Sigma \binom{N}{i} (y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i)))$$

Gini Impurity

Gini Impurity is a measure of variance across the different classes

[8]:
```
import latexify
@latexify.function(use_math_symbols=True)
def Gini(node):
    return Sigma(_i**c) (p_i * (1-p_i))
Gini
```

[8]:
$$\text{Gini}(node) = \Sigma \binom{c}{i} (p_i (1 - p_i))$$

Four P’s

Past

- We performed feature Engineering.
- We have done the hyperparameter tuning on the Xgboost to get the optimal parameters.

Present

- We implemented Multi Layer Perceptron.
- We increased number of features selected from 14 to 50 features.

Four P’s

Problems

- At present, everyone are working hard to complete the project and everything went well.

Planned

- We have reached final stage of the project.

Results and discussion of results

The models were overfitting in phase 3 and had an accuracy of 1. We improved this, and the models performed better than in the previous phase after increasing the number of features to 50.

Following are the results we have got in Phase 4:

Logistic Regression
- Test accuracy - 91%
- Test ROC_AUC - 0.685

Random Forest
- Test accuracy - 92%
- Test ROC_AUC - 0.661

XGBoost
- Test accuracy - 92%
- Test ROC_AUC- 0.720

MLP
- Test accuracy - 68%
- Test ROC_AUC- 0.705

Overall, XGBoost performed better among all the models.

Conclusion

After exploring new features from EDA in phase 2. We have focused much on improving the test accuracy, additional Feature Engineering, HyperParameter Tuning, Feature selection, analysis of Feature importance, and other ensemble methods in this phase.

In Phase 3, the data was trained and tested with the help of ML Classifiers Logistic Regression, Random Forest, and XGBoost. Accuracies and areas under the ROC_AUC curve were found and compared. We have improved our Logistic Regression and Random Forest compared to our phase 2 submission. Logistic Regression has achieved an accuracy of 0.808. We have achieved this by doing feature engineering on secondary and tertiary tables. Later we chose the 50 best-correlated columns and dropped all null values. We then performed hyper-parameter tuning. XGBoost gave an accuracy of 0.996.

In the final phase: This project's main goal is to develop a Machine Learning model that can predict whether or not a loan applicant will be able to repay the loan. Without any statistical analysis, many deserving applicants with no credit history or default history get approved.

Conclusion(Continued……)

The HCDR dataset is used to train the machine learning model. Based on the history of comparable applicants in the past, it will be able to estimate whether or not an applicant would be able to repay their loan. This model would help in the screening of applications by providing statistical support resulting from numerous aspects taken into account.

In the previous phase of the project the models were overfitting the data and hence got an accuracy score of almost 1, to overcome this we added new features using the pipelines, and tables were merged considering inherent hierarchy. A neural network model, Multi-layer Perceptron was implemented using PyTorch in the final phase, and MLP was modeled with two hidden layers, twenty neurons, and 166 input dimensions. This resulted in an accuracy of 68% with an AUC score of 0.705. Out of all models implemented, **XGboost gave the best results resulting in an accuracy of 92.2 and an AUC score of 0.720**. **We have also made a Kaggle submission and obtained a score of 0.7255**.