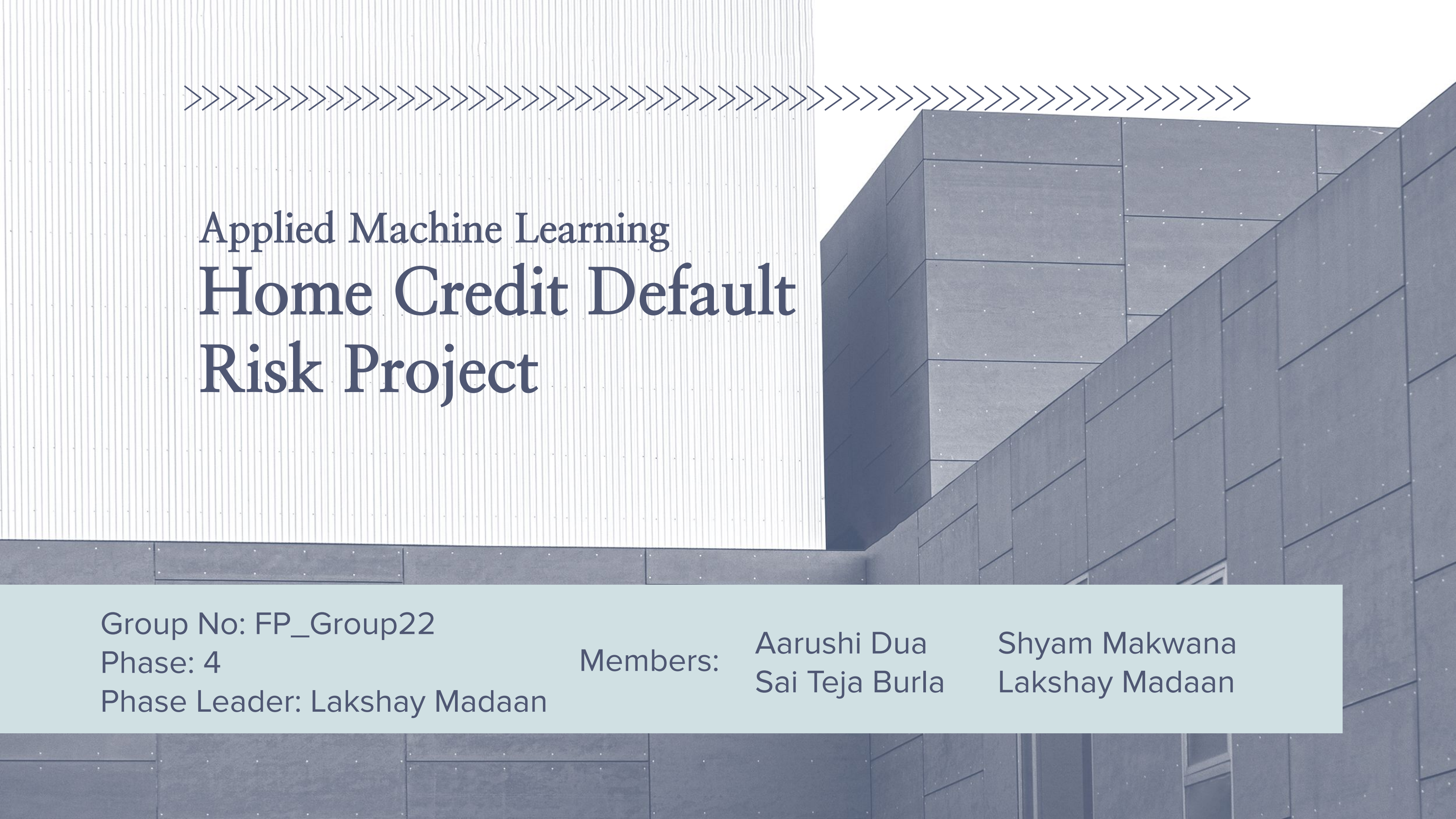


Applied Machine Learning

Home Credit Default Risk Project

Group No: FP_Group22
Phase: 4
Phase Leader: Lakshay Madaan

Members:

Aarushi Dua
Sai Teja Burla
Shyam Makwana
Lakshay Madaan

01 Team Names and Photo

02 Project Description

03 Phase 1, 2, 3

04 ML Pipeline Overview

05 Phase 4

06 TensorBoard Visualization

07 Results & Discussion

08 Conclusion and Next Steps

Team Names and Photo

Project Description

- The objective of the HCDR project is to assess a borrower's capacity to repay a loan, based on the features in the dataset
- To accomplish this objective, we will examine multiple factors of the applicant’s profile in addition to their credit history
- During this phase, our aim was to implement Multilayer Perceptron architectures to analyze our data and evaluate the performance based on the cross-entropy loss function. We utilized the ReLU activation function with the Adam optimizer and performed 1000 epochs using 4 hidden layers and 2 output features.
- Additionally, we utilized TensorBoard to visualize our accuracy and loss. As a result, we achieved a test ROC of 0.76 and a Kaggle submission public score of 0.728.

Phase 1, 2, and 3

Phase 1

It was all about understanding the given datasets and finding out that it is a classification problem. Making a decision on what machine learning models can be used to get good results and keeping track of loss functions for the same. Checking all the performance metrics that could be used to check how good our model is.

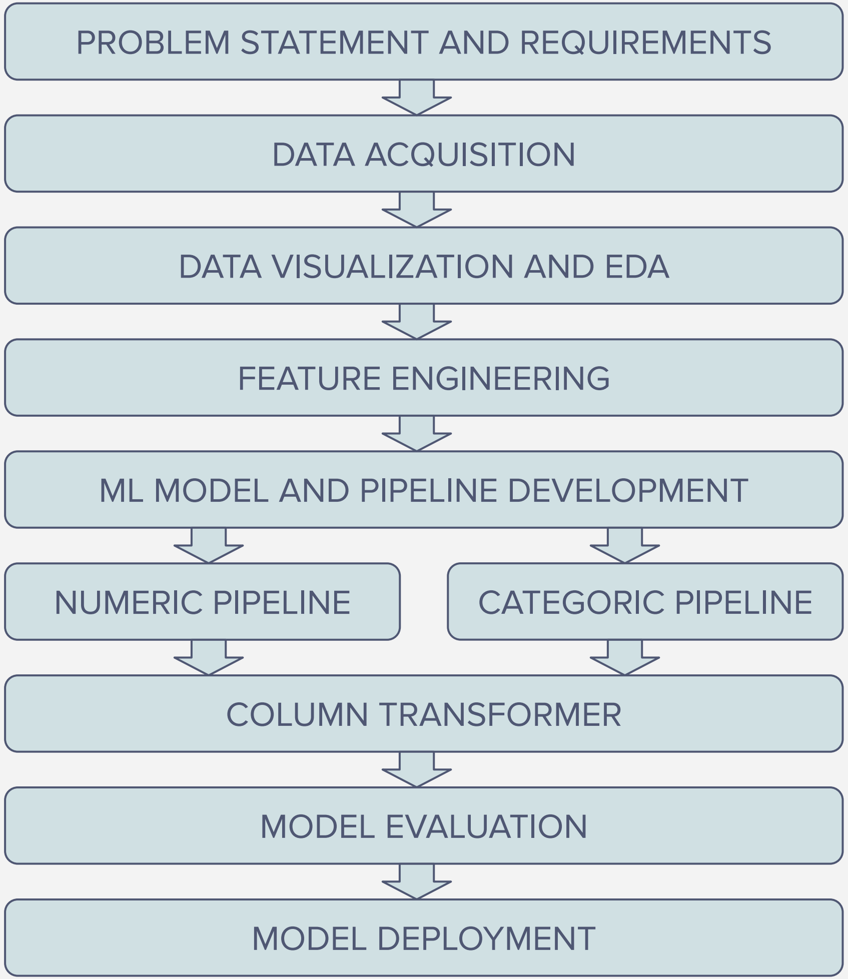
Phase 2

It was about detailed analysis of the dataset and its features with the help of Exploratory Data Analysis and making Baseline models for the problem. The models that we implemented are: Logistic Regression, Decision Tree, Random Forest and Gaussian Naive Bayes.

Phase 3

We applied all the preprocessing techniques on the merged dataset. We performed feature engineering techniques like removing columns with more than 60% missing values, removing columns with threshold more than certain value, checking for feature importance, adding additional rows based on the knowledge of the given dataset. We made use of Hyperparameter Tuning to improve the quality and accuracy of the previous baseline model and chose the best set of parameters using GridSearchCV.

ML Pipeline Overview



PROBLEM STATEMENT AND REQUIREMENTS

DATA ACQUISITION

DATA VISUALIZATION AND EDA

FEATURE ENGINEERING

ML MODEL AND PIPELINE DEVELOPMENT

NUMERIC PIPELINE

CATEGORIC PIPELINE

COLUMN TRANSFORMER

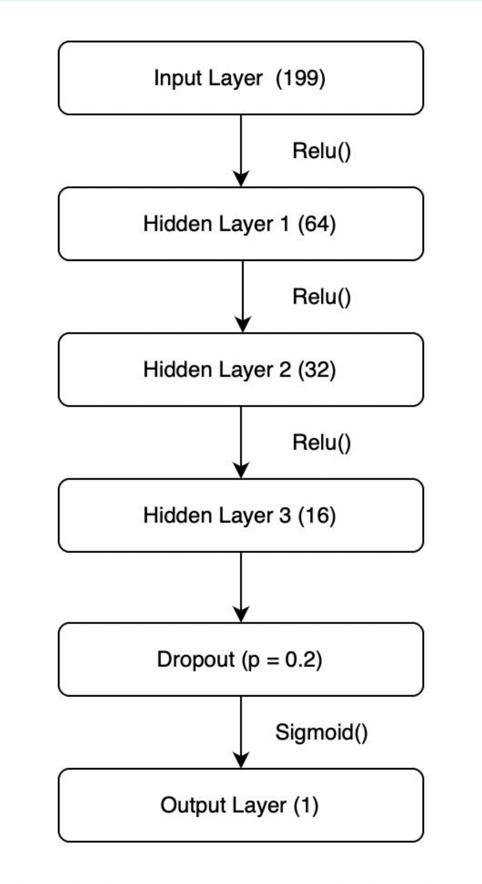
MODEL EVALUATION

MODEL DEPLOYMENT

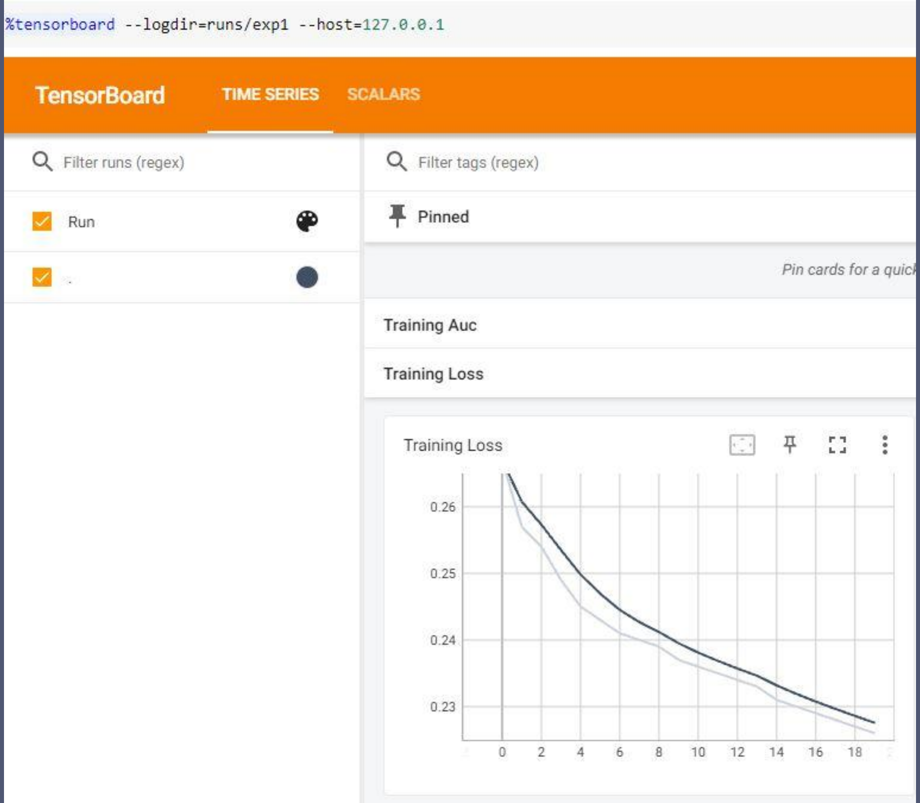
Phase 4

The following steps were performed as a part of phase 4

Our objective in this stage was to apply Multilayer Perceptron structures to examine our data and measure the effectiveness based on the cross-entropy loss function. We applied the ReLU activation function with the Adam optimizer and executed 1000 epochs utilizing 4 hidden layers and 2 output features. Moreover, we employed TensorBoard to display our accuracy and loss graphically. Consequently, we obtained a test ROC of 0.76, resulting in a public score of 0.728 for our Kaggle submission.

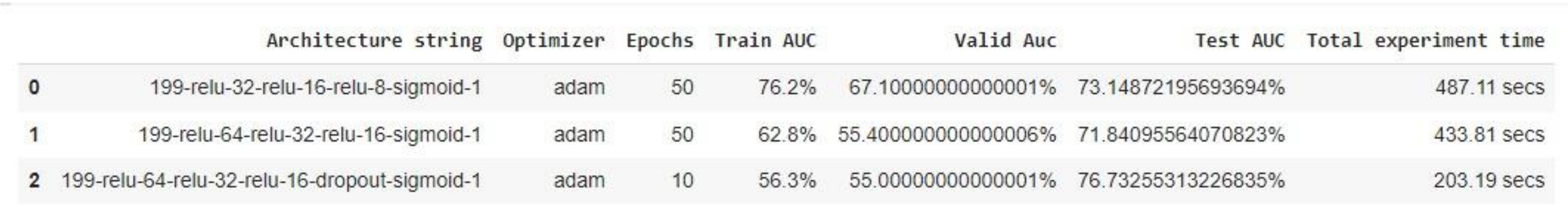


TensorBoard Visualization

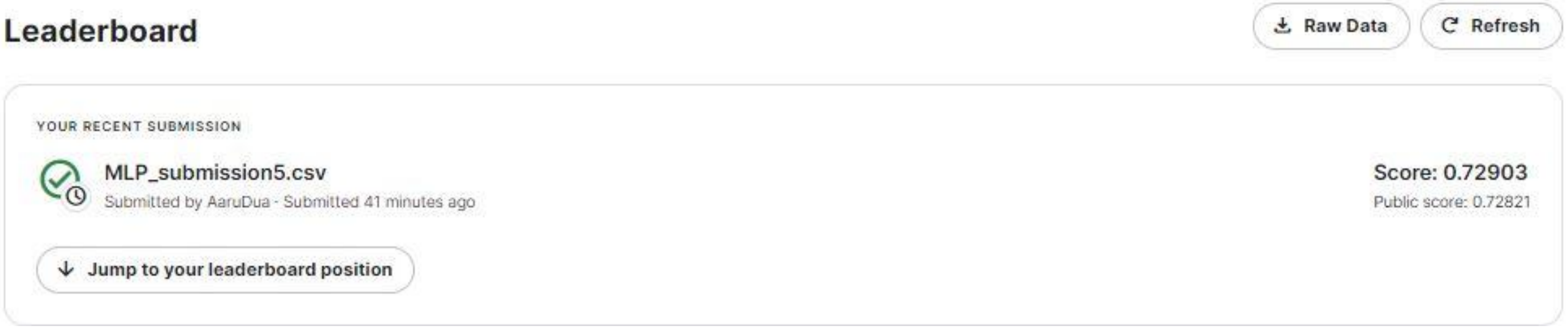


```
%tensorboard --logdir=runs/exp1 --host=127.0.0.1
```

Results & Discussion



	Architecture string	Optimizer	Epochs	Train AUC	Valid Auc	Test AUC	Total experiment time
0	199-relu-32-relu-16-relu-8-sigmoid-1	adam	50	76.2%	67.10000000000001%	73.1487219569369 4%	487.11 secs
1	199-relu-64-relu-32-relu-16-sigmoid-1	adam	50	62.8%	55.400000000000006%	71.84095564070823%	433.81 secs
2	199-relu-64-relu-32-relu-16-dropout-sigmoid-1	adam	10	56.3%	55.00000000000001%	76.73255313226835%	203.19 secs



Leaderboard

YOUR RECENT SUBMISSION

MLP_submission5.csv

Submitted by AaruDua · Submitted 41 minutes ago

Score: 0.72903

Public score: 0.72821

Conclusion and Next Steps

- Implemented Multilayer Perceptron(MLP) model using PyTorch
- MLP model gave an AUC score of 0.76
- Logistic Regression gave test AUC score of 0.74, Random Forest gave test AUC score of 0.72, XGBoost gave test AUC 0.77
- Non-deep learning models performed better than MLP model
- XGBoost gave the highest AUC score of 0.77
- Model can be improved by using embeddings in the deep learning models or advanced classification models like boosting models

Thank You!!!