Machine Learning Project Report

Milestone I

In the first part of the project, we cleaned the data, which was quite challenging and time-consuming. Each data point was considered to have 10 features: the 5 nearest podcasts and 5 user favorite categories. After preprocessing, we removed 1 of the 5 nearest podcasts and 1 of the 5 favorite categories. In the end, our samples had 8 features and 1 label, which was either 0 or 1. For more details, you can check the milestone 1 report in the milestone 1 folder.

Milestone II

After data preprocessing, we trained five models: K-Nearest Neighbors (KNN), Random Forest, Linear Regression, Lasso Regression, and Decision Tree. Since the problem involves binary classification with highly imbalanced classes, we set our evaluation metric to recall because we want our model to accurately identify the positive class (ones). As expected, the regression models did not perform well. Here is the performance of each model:

model	Recall Score	Training Time	Prediction Time
Decision Tree	0.2	3.22(s)	0.02(s)
Linear Regression	0	0.14(s)	0.02(s)
Random Forest	0.011	10.54(s)	0.2(s)
Lasso Regression	0	0.11(s)	0.003(s)
KNN	0.0046	1.38(s)	30.96(s)

Table 1: Models Performance

Milestone III

In the first part, we utilized the same dataset as in Milestone 2 and employed a neural network for the task. The problem involves binary classification with highly imbalanced classes. Initially, training the neural network on the raw data resulted in predictions exclusively of zeros. Given that class one labels constitute about 1% of the total, predicting all labels as zero would yield an accuracy of approximately 99%.

To enhance the model's sensitivity in predicting class one, we pursued two approaches: first, by assigning different weights to the classes, and second, by employing oversampling. The results are detailed below.

Evaluation Results		
baseline	**Class weights**	**Oversampling**
loss : 0.030 tp : 0 fp : 0 tn : 198533 fn : 1395 accuracy : 0.99 precision : 0.0 recall : 0.0 auc : 0.91 prc : 0.060	loss : 0.34 tp : 1291 fp : 36452 tn : 162081 fn : 104 accuracy : 0.82 precision : 0.034 recall : 0.92 auc : 0.93 prc : 0.061	loss : 0.34 tp : 1282 fp : 34748 tn : 163785 fn : 113 accuracy : 0.82 precision : 0.035 recall : 0.92 auc : 0.93 prc : 0.061

In the next section, we utilized the StellarGraph library to implement a GraphSAGE model for link prediction tasks. Initially, we constructed a bipartite graph using NetworkX, where one group represents app users and the other represents podcasts based on follower data. We incorporated node features derived from user datasets, converting the NetworkX graph to a format compatible with StellarGraph. The GraphSAGE model was configured with specific layer sizes and trained using TensorFlow/Keras. Finally, we evaluated the trained model on the test data, achieving an accuracy of 79.83%. This process illustrates how StellarGraph enables efficient construction and training of graph-based machine learning models. Next, we trained a Graph Convolutional Network (GCN) model after creating a suitable dataset from the initial data. The model achieved a performance of around 97% accuracy and a 62% recall score, which is very good in comparison to classical models.