COPS Summer of Code Report

M Gokulan
Roll No: 24155043
Department: Mining Engineering

Project Title:
Neural Network Implementation on Medical Appointment No-Show Dataset

Abstract

This report discusses the implementation and comparison of two models: one built as a neural network from scratch, and the other using the PyTorch framework.

Project Approach

- Performed data preprocessing, including label encoding and feature scaling, to prepare the dataset for training.
- Built a neural network from scratch, but it failed to learn—producing a constant loss of 0.250 across all epochs.
- Suspected class imbalance as the cause and applied class weights, but observed no improvement.
- Switched to a PyTorch implementation, which showed some learning but didn't yield strong results.
- Tried initializing weights and biases with zeros in the scratch implementation, but it still didn't work; others faced the same issue.
- Focused on improving the PyTorch model and explored solutions beyond class imbalance.

- Realized that feature engineering could boost performance and added meaningful features:
  - Days between scheduled date and appointment date
  - Total number of appointments per patient ID
  - No-show rate per patient
- These new features significantly improved the model's performance.
- I did not use the same number of epochs for both models, as I experimented with different values and observed the results each time. The best performance was observed when the number of epochs was around 75. Therefore, comparing convergence time and speed is not meaningful.

Model Comparison

```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.90      0.85     17642
           1       0.23      0.12      0.15      4464

    accuracy                           0.74     22106
   macro avg       0.52      0.51      0.50     22106
weighted avg       0.69      0.74      0.71     22106

F1 Score: 0.1540532455877954
ROC AUC: 0.44500260101444133
Confusion Matrix:
 [[15935  1707]
 [ 3949   515]]
```

Figure 1: From Scratch Implementation

```
classification report              precision    recall  f1-score   support

          0       0.98      0.88      0.93     17642
          1       0.66      0.91      0.76      4464

    accuracy                           0.89     22106
   macro avg       0.82      0.90      0.84     22106
weighted avg       0.91      0.89      0.89     22106

f1 Score 0.7643252368001501
ROC AUC Score 0.8962319549226573
Confusion Matrix
 [[15518  2124]
 [  389  4075]]
```

Figure 2: PyTorch Implementation

I did not use the same number of epochs for both models, as I experimented with different values and observed the results each time. I found that the best performance occurred when the number of epochs was around 75. So there will be no meaning in comparing the convergence time and speed.