RL Datix Assignment Report

Anton Price

May 27, 2025

1 Binary Classifier

The first task here was to make a binary classifier model on tabular data. My approach to this was fairly straightforward, to look at the data and do some simple EDA to see if there are any outliers and check the target distributions by each variable. From there I could do some simple feature engineering, mostly some target encoding of the character categorical variables to make them useable in a model. Then finally split the data and train a gradient boosting machine, in this case LightGBM with a training dataset and an early stopping dataset, which is essential for any boosting class of model. Finally I then output a standard report for binary classification models, that is feature importances, a confusion matrix, a classification report and ROC/PR curves.

In order to evaluate the performance of such a binary classification model the most important factor to consider is the use case and the real world implications of misclassification. In this case the model is just trying to predict the likelihood of a patient needing a follow up appointment and the data is imbalanced with the positive samples being roughly half as many as the negatives. This particular model would definitely be a failed candidate for a go live as I would want a model that would at minimum have a higher probabilty of accurately guessing the positives and probably wouldn't worry too much about the recall rate as a false positive just means the doctors will have extra time, however false negatives could lead to long waiting times for patients.

If given additional data I would implement Bayesian Optimisation for hyperparameter tuning as well as a 3rd validation fold for a true out of sample test.

2 Named Entity Recognition

The second task in this assignment was to create a named entity recognition model on the text field `discharge_note` in the data. My approach to this was to fine tune a BERT mode, to do this I cleansed the text data by making it all lower case and removing punctuation, and then generated the targets by manually going through the data. Next up was converting the data into tensors that the model could read by tokenizing it and then going over the targets to make sure the targets all aligned correctly. With the preprocessing done the next task was simply to train the model and format the outputs. When it came to training the model didn't really train particularly well at all unfortunately. I will attribute this to the total text corpus being 10 lines, which is absolutely microscopic when compared to the terabytes of text data that modern LLMs are trained on. Due to the real world impact of giving misleading diagnoses and medical notes to doctors being potentially disastrous I would strongly request much more and preferably pre-labelled data.