Intro To Data Science Final Project

Stephanie Ajah; Ernest N. Frimpong

Executive Summary

In this project, we predicted user's movie ratings. Using the training dataset, we computed the best rating estimate on the test dataset. We used random forest and a multi layer neural networks with 'relu' as the activation for the three layers and linear as the output layer activation. The random forest model was chosen as it produced a better result, and on the leaderboard, we achieved an rmse of 91. Below, we summarize the data pre-processing section, detail the models used and compare their results.

Data Pre-Processing

We started the data preprocessing by checking for duplicates on the training data and we had none. We then proceeded to concatenate the datasets into a 'full_df' to allow all the data preprocessing steps to be carried out on both the training and test set concurrently. While exploring the columns, we noticed the title column contained the movie names and the year, so we decided to split that into their own separate columns. We also checked for null values on the newly generated year column, and we had 17. We manually searched for the years the corresponding movies were made, and we filled it in, by creating a dictionary and using a '.map' function to map the movie names to their corresponding year. This ensures that the dataset had no null values. After this, we dropped unnecessary columns like 'title' and 'timestamp'.

To convert our categorical columns to numerical columns we decided to use vector embeddings. The categorical columns were 'moviename' and 'genres'. We chose vector embeddings because we wanted to be able to capture the relationships between the tokens that using one-hot encoding will not capture. We used a vector size of 15 as we felt this will both yield to good results and also not be computationally expensive. We set the window to 5 and min-count to 1 in order to limit the chances of overfitting.

After embedding, we had 32 columns and 100836 rows. Following the embedding, we went ahead to split the dataframe back up to 'train_data' and 'test_data'.

Modelling & Model Tuning

Model One

We started by splitting the dataset into training and validation sets using an 80-20 ratio and trained RandomForestRegressor on our dataset and we got a training MSE of 0.758 and a validation MSE of 0.932. We obtained predictions on the test data and put these predictions in the leaderboard and we got an RMSE of 91.

Model Two

While 91 was a good score on the leaderboard, we wanted to explore neural networks as according to our research, it could also have a massive impact on the model. We decided to try a multi-layer neural network with relu as the activation for the three layers. Adam (Adaptive Moment Estimation) was the chosen optimizer for our model as it is widely used for its efficiency and effectiveness in handling sparse gradients and noisy data.

After training for 20 epochs, we got 1.101 as the training MSE and 1.080 as the validation MSE. This was disappointing as we hoped it'd be an improvement from our random forest model. However, this was very insightful and educational to try and learn from.

MODEL COMPARISON

Given the results of both models, it was fair to say that the random forest resgressor was a better and more efficient choice, so we went with the RandomForestRegressor as the final model choice.