Ahbab Ashraf

>>> Project 3

> Problem

For my project, I wanted to predict the popularity of a song based on its musical qualities.

To accomplish this, I used the Spotify Tracks Dataset. It contains nearly 120k samples in the original set, but my implementation removes repeated songs to handle just over 80k samples. Each sample has 19 features, but I dropped all string and binary (having only 2 unique values) features to arrive at 11 for my model.

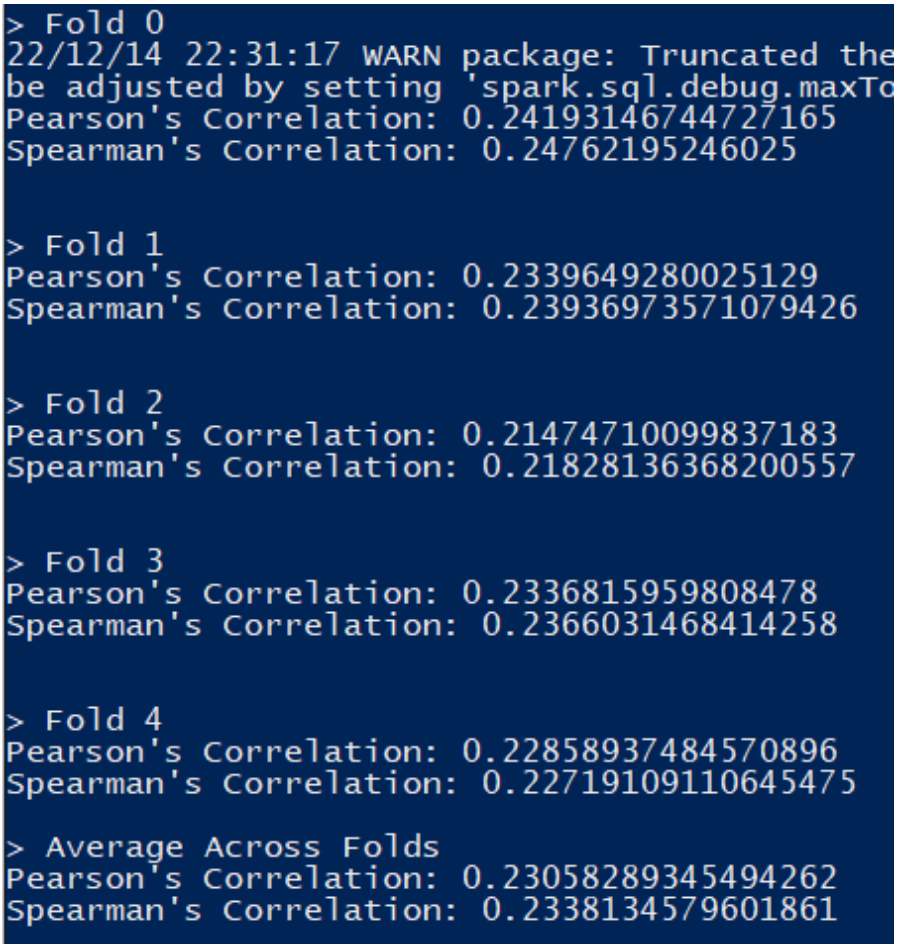
The feature names are as follows: duration_ms, danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, and tempo. The target label is popularity, which is skewed such that there are a disproportionate amount of very low popularity songs.

This is a regression problem, as I am trying to predict a continuous value from my features. Moreover, every feature that I consider is continuous. This means that a Linear Regression algorithm makes the most sense for my case.

> Training

For the training procedure, I tried to compensate for the uneven label distribution. I did so by splitting the dataset into 3 groups based on song popularity. From these I created 5 data splits with a roughly equal amount of low, medium, and high popularity songs. This attempts to make each split more representative of the dataset, minimizing the risk of oversampling for one level of popularity.

With these splits I then performed 5-fold cross-validation, choosing one split as testing data and unifying the rest as training data. This was done once for each split, resulting in 5 folds. In each fold I fit the model before making predictions and printing metrics. The results are as follows:



> Discussion

Unfortunately, the metrics for my model are consistently low. This means that not much insight can be gained, as it cannot predict popularity accurately. There are some factors I have considered which may affect performance.

Firstly, my implementation only accounts for the numerical features available in my dataset. There are 5 string features and 3 binary features which I drop from my data. Perhaps accounting for those as well will result in more accurate predictions.

It could be that this problem requires a more sophisticated implementation with multiple models. Each model could be trained on a specific range of the popularity label. The predictions from such models could then be fed as features to a higher level model. Said model would output the final prediction.

Finally, it is possible that the dataset is simply too noisy to gain accurate results. Even cutting down from 19 features in the original dataset to 11 in my implementation, it is possible that some features have no correlation with the target label. Perhaps an improved implementation would have weights for different features, or simply use fewer features.