Assignment 1

CIS4930- Intro to ML

Kenneth Krause

3/3/2024

Link to Code Walkthrough: https://youtu.be/p_Bk2BPyssk

Implementation

Classification Models

The classification models were trained, per request, on the Final Board Dataset, as well as the Intermediate Board- Single Label Dataset. The goal for these models is to predict the optimal index of the next move on the board. Being that this was a classification model, the output consisted of the label of the optimal move, a whole number from 0 to 8.

Regression Models

The regression models were trained, per request, on the Intermediate Board- Multi-Label Dataset. The goal for these models is to predict the optimal next move given a board marked with player positions. Being that these are regression models, the output consisted of decimal values from 0 to 1, based on the "confidence" of the model in this position. These values were then rounded to the nearest whole number, in the case of the KNN Regressor, or the highest confidence value was taken, in the case of the MLP Regression and Linear Regression models.

Cross-validation for these models required a custom KFold implementation, since the predicted values are continuous. I was able to find a section in the textbook for this course that created a KFold validation template, and updated this to fit the needs of this assignment. The basic flow of the custom KFold validation is as follows: (1) Create shuffled KFolds. (2) Iterate through each fold of training and testing data, training model on each then assessing its score. (3) Store and average scores of all testing folds to produce single CV Score.

The Linear Regression model required special care to implement and evaluate, seeing as we needed to train a model to regress on each specific output class. The steps to implement this model were as follows: (1) Create a list to store the 9 models (1 for each output class). (2) Create, train, and store each model on 1 of the 9 outputs. (3) For cross-validation, each model was iterated over, data was split into 10 different train/test sets, model was trained on each, predictions were rounded as previously mentioned, and then the model was assessed on that dataset. Results were then averaged first by dataset per model, then by combined models.

Results

Classification Models

The classification models were able to predict with a high level of accuracy the next optimal move for each dataset, with the models trained on the Final Dataset producing the quickest, and most accurate results overall. This makes sense because the model is aware of every position on the board, and does not have to take into consideration potential next moves.

Perhaps most notable, the Linear SVM Classifier was able to predict with 98% accuracy on the Final Dataset, however dropped to around 77% for the Intermediate play dataset. To follow, The MLP Perceptron was able to produce over 90% Accuracy for each dataset. Being the most complex model, this makes sense as it can analyze relations across every datapoint for each board it was trained on. This also resulted in by far the longest training time. Somewhere in the intersection between performance and accuracy laid the KNN Classifier, who could achieve almost 100% accuracy on the Final Dataset, as it also analyzes based on the relationship between features, but could only predict with <80% accuracy on Intermediate Datasets.

Regression Models

While requiring slightly more feature engineering, the KNN and MLP Regressors were able to predict with a very high level of accuracy the optimal move given the Muli-Label dataset. With both scoring over 90% in both cross-validation and accuracy, these impressive feats display regression's ability to find patterns in complex data. The Linear Regression model received cross-validation scores in the upper 70s, however was only able to achieve <50% in the accuracy score. While this is more than complete randomness (whose score would be <=1/9), it shows possible overfitting based on the training technique provided. Additionally, the confusion matrix for the Linear Regression model provided valuable insight into its prediction strategy, the data, and why it was least accurate. This matrix showed a severe favoring towards even indices, and even more so to the middle index, 4. While puzzling at first, these results make sense in terms of the game of Tic Tac Toe, where even numbered indices are the corners (0,2,6,8) and the middle (4), giving the most amount of win opportunities to the model if these can be selected.

How to Run:

Model Analysis

Simply need to run every cell in order to display the Cross-Validation Scores, Confusion Matrices, and Accuracies.

Gameplay

Run all cells, with the last cell being the gameplay driver. When playing the game, you will first be prompted to select the model you would like to play against. From here, you will take turns with the model selecting squares by index number (1-9) until the board is full or a win condition has been met. The game driver should automatically terminate when there is a winner or the board is full and has done so with no bugs found.