Tyler Arndt
COMP 479
Homework 3
10/16/2019

For this assignment, I used a dataset called “nba_logreg.csv” from https://data.world/exercises/logistic-regression-exercise-1. This file contains the data of 1,340 current and former NBA players and includes 20 features relating to each player’s rookie year statistics. The overall goal is to determine whether a particular NBA player is likely to last 5+ years in the league, solely based on their rookie statistics. Ultimately, the most effective evaluation metric of this dataset is the accuracy of these predictions.

In the pre-processing phase I only used what I considered the most important features: games played, minutes per game, points per game, rebounds per game, assists per game, steals per game, blocks per game, and turnovers per game. Next, I filled the missing values with the average of that feature’s data, and then used the Scikit-Learn train_test_split() method to distribute the dataset into training (70%), development (15%), and test (15%) datasets. Finally, I scaled the features of these datasets using standardization.

I used the Sciki-Learn logistic regression algorithm to run on the development dataset. Using the default parameters, I was able to correctly predict the target values of the development set 67.66% of the time (Figure 01). In order to find the optimal inverse-regularization parameter, I generated a for-loop that ran the algorithm under different parameters and plotted their respective error rates (Figure 02). I found that values larger than the default parameter (1) produce the same number of errors, but any inverse-regularization parameter less than that value would show an increase in the number of incorrect predictions.

For my implementation of the k-nearest neighbors algorithm, I aimed to avoid the ‘curse of dimensionality’ by using only two features: minutes per game and points per game. Once again, I used a for-loop to determine the best ‘neighbor’ value to use in my k-nearest neighbors algorithm implementation. By plotting the error rate against the algorithm’s ‘neighbor’ value, I determined that a value of 50 was the best fit for my development data (Figure 03).

When comparing the optimized logistic-regression and k-nearest neighbors algorithms on the testing data, I was able to achieve 73.13% accuracy with the logistic-regression implementation and 69.15% accuracy with the k-nearest-neighbors implementation (Figure 04).

After testing different algorithms with multiple classifier parameters, the highest accuracy I was able to achieve was 73.13% indicating that there is a moderate to strong correlation between an NBA player’s rookie statistics and whether he is on an active roster 5 years later. That said, alternative algorithms should be run on the dataset to try and increase the overall accuracy of the results.

```
QUESTION 1 - run Scikit-learn log-regression with default params on development set:
Mistakes: 65
Predictions: 201
Accuracy: 67.66169154228857%
```

Figure 01

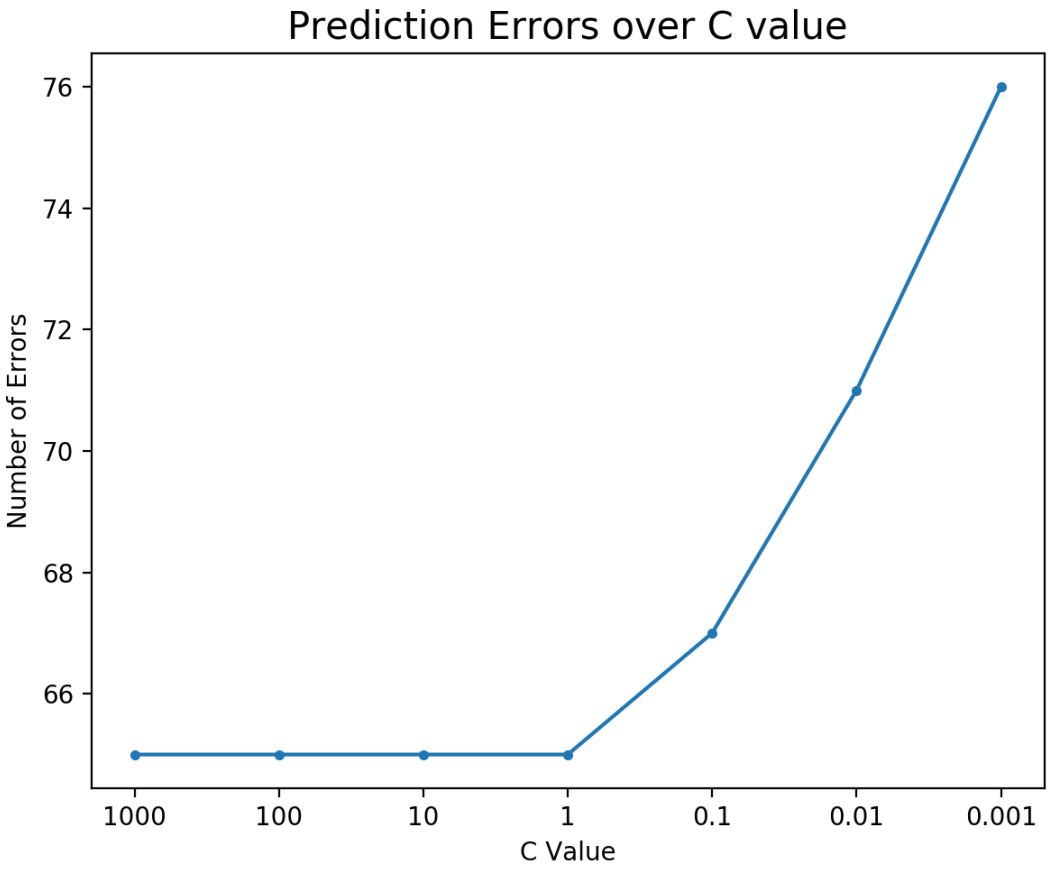

Figure 02

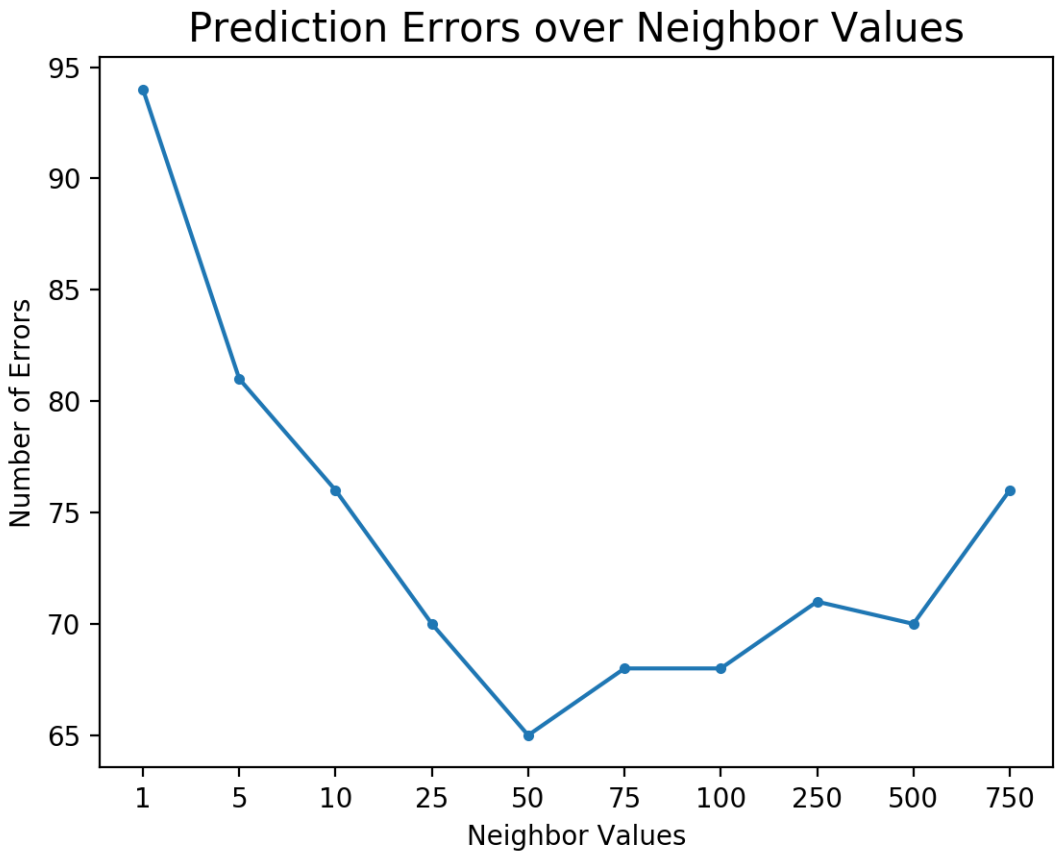

Figure 03

```
Question 4: Logistic Regression Results:
Mistakes: 54
Predictions: 201
Accuracy: 73.134328358208 96%

Question 4: K Nearest Neighbor Results:
Mistakes: 62
Predictions: 201
Accuracy: 69.15422885572139%
```

Figure 04