Marianopolis College

Kobe Bryant Shot Predictor:

Predicting the Outcome of Shots Using Extra Trees and Extracting Relevant Career Statistics Out of Raw Data

Karl Michel Koerich

Programming Techniques and Applications

Robert D. Vincent

18 May 2018

Introduction

This paper aims to explain in a simple and direct manner the basic functionalities and programming techniques used in the final project for Programming Techniques and Applications. The project was inspired by one of Kaggle's 2016 competitions. "Kobe Bryant Shot Selection: Which shots did Kobe sink?" was a public competition in which participants had to predict whether 5000 of Kobe's shots went in or not. Participants were provided with a dataset containing the circumstances of every shot in Kobe's career. By using this dataset and by applying Machine Learning concepts seen in class (especially Lab 6), the final project's program successfully predicts approximately 65% of Kobe's shots. In addition to that, it uses the dataset to extract relevant statistics about Kobe's career and provides them to the user.

Mini-Manual for Users

After downloading "Final.zip" and extracting its content, the user should open "RunMe.py" and run it. The user is initially faced with a 3-options menu:

1. Train
2. Statistics
3. Exit

The user then inputs its decision (1, 2 or 3) and presses enter, leading to one of the following:

Train – This option will train the program to be able to make predictions. It will take a while for it to run since the dataset contain 30697 shots. Once it finishes running, the user will see the system's percentage of successful predictions and a confusion matrix. Finally, the initial menu reappears. Although it is unlikely that the user will select the option 1 a second

time, this option is kept in the menu in case the user wants to see if there are differences in percentages between trainings.

Statistics – A table with Kobe Bryant's career statistics is printed. It contains the ratio successes/attempts for 2PT and 3PT shots, points he scored per team, the games he scored the most and the least points. After all this is printed the initial menu reappears.

Exit – The program finishes running.

Design Guide for Programmers

The main code is found under "RunMe.py" and it is fairly simple. First, it imports *extra_trees* which is the classifier used for this project, and *data_item* which is the format for the data objects. Secondly, it opens "kobe_data.csv" and starts iterating over the lines of the dataset. The program only iterates over the data once to void unnecessary iterations. While it is iterating, it extracts the relevant information to compute statistics and then transforms each line into *data_item*s in order to set up the dataset that will later be used for training. Once the iteration is over, the program gives the user the option of choosing what he or she wants to do.

Setting up the dataset means transforming into integer some values that cannot be directly converted to integer. The program assigns numerical values to strings and stores the number/string relationship into the list *possible_values* (to facilitate, the numerical values of a string is its index inside the sub lists in *possible_values)*. Also, 3 features that do not impact on the predictions are ignored when creating the *data_item*s: *loc_x*, *lox_y* and *shot_id*.

Once the data is set, the program is ready to ask the user for its input. If the user chooses to "Train," the function *train_data()* is called and it initially split the dataset 5 times into training and testing folds to perform 5-fold random sub-sampling cross validation. For

each fold, an Extra Tree classifier is created; it is trained with *data_fold* and then tested with *test_fold.* While it is performing the tests, the code checks if the predictions made are true negatives, false negatives, false positives or true positives. The files for the classifier were taken from Lab 6 of this course; furthermore, the internal functionalities of Extra Trees are not going to be explained in this paper since it is quite advanced for level of this course and beyond of what it expected for the students to know.

It takes a fairly large amount of time to train the classifier since the database is large and the techniques being applied on this project are immature. So after a long time waiting for the trainings to be done, it simply formats and prints the results for the user in a confusion table.

Finally, if the user chooses "Statistics," the previous data that was collected during iteration is simply formatted and printed, and if the user chooses "Exit," the program finishes.

Conclusion

This paper explained the basic functionalities and programming techniques of the final project, which was inspired by Kaggle's 2016 competition "Kobe Bryant Shot Selection: Which shots did Kobe sink?" The program uses Extra Trees to predict if Kobe's shot were on target or not. Although it was really slow to train using this dataset, Extra Trees provided the right prediction approximately 65% of the times. Several attempts were made to try to make the training process faster but none were significantly successful. In addition to predicting shots, the program was successful on abstracting raw data from the database and transforming it in meaningful and readable statistics about Kobe's career.