Machine Learning Engineer Nanodegree

Capstone Proposal

Paul Salaberria

August 21st, 2017

Proposal

Domain Background

During the last decades, football has become the most popular sport in Europe. The British Premier League, Italian Calcio, German Bundesliga, French League 1 and the Spanish La Liga are some of the most important leagues. There are millions of watchers across the globe, including markets like China or India, which leads to numerous business opportunities such as TV rights, merchandising, betting, and many more.

I am especially interested in the bettings. Multiple betting companies operate both online and offline, giving their users an opportunity to become rich. For this to happen, however, there will also be users who lose all of their betting money. I would love to beat the betting systems by predicting the results of the upcoming games. For this project I am choosing the Spanish league because it is, in my opinion, the best league in the world. I could not find any existing research on Spanish league predictions, which makes my research even more interesting.

Nowadays, we have access to very rich information about football teams, players, and several other sensors. Humans are not capable of processing all this information without the help of computers. This is where machine learning comes into play. Once we have collected data from many inputs, we can train a model to generate predictions. Classification algorithms and neural networks are two of the options in this regard.

Problem Statement

Many different predictions can be made for a football match. The number of goals per team, the top scorer of the game, who will receive a yellow card, etcetera. However, I would like to start with a simple prediction, which will predict the winner of the match. It will be a win/lose/draw classification problem. In order to know if it is a clear win, an obvious draw, or a evident lose, I want to get the probability of an instance being in each of the classes. Many classifiers in scikit-learn offer a ‘predict_proba’ function which returns exactly what I am looking for. Naive Bayes or SVMs should work for the given use case.

Most of the features in the final dataset will be numerical.

Datasets and Inputs

“The ultimate Soccer database for data analysis and machine learning” is the name of the dataset used for this project. It’s available in Kaggle: https://www.kaggle.com/hugomathien/soccer.

The chosen dataset was originally created by Hugo Mathien, a Kaggle user. He used a crawler (https://github.com/hugomathien/football-data-collection/tree/master/footballData) to collect data about teams and players using an external API provided by EA Sports Fifa.

The dataset contains the following:

- +25,000 matches
- +10,000 players
- 11 European Countries with their lead championship
- Seasons 2008 to 2016
- Players and Teams' attributes* sourced from EA Sports' FIFA video game series, including the weekly updates
- Team line up with squad formation (X, Y coordinates)
- Betting odds from up to 10 providers
- Detailed match events (goal types, possession, corner, cross, fouls, cards etc...) for +10,000 matches

The dataset is a combination of tables included in a sqlite database.

My target variable will be ‘home team result’ in the format of win/draw/lose, or 1/2/3 if the algorithm only accepts numeric values. The target column will be populated with a function that compares home team goals against away team goals. I will also analyze the distribution of the target variable to find out if it’s a balanced dataset.

If the model created using the current dataset gives good results, I will try to add teams and games for seasons 2016/2017 and 2017/2018, and try to predict future games.

Solution Statement

First, a useful dataset with relevant columns needs to be created. The final dataset should contain one row per game. Due to the richness of the source dataset, we should be able to add hundreds of features. This will increase the training time for the algorithm, and therefore we may start with a smaller number of features, and add more as we go along. We could also look into techniques such as PCA to combine correlated features. The feature_selection module in sklearn could be also worth checking.

We can define the problem as a multi-class classification problem with 6 possible outputs: Win, draw or lose for the home and away teams.

SVMs and MultinomialNB (http://stuartlacy.co.uk/bayesianfootball-27062017) are two algorithms that try to solve this kind of problems. Using these algorithms in scikit-learn we can create models that will give us the probabilities for each class.

Benchmark Model

The bookmakers get the predictions right 55% of the time in the Spanish league. We also know that the percentage of home victories adds up to 40-45%. If we bet constantly for home victory we will get an accuracy of 40-45%. Therefore, our goal is to get an accuracy higher than 55% so we can win against betting systems. Anything lower than 45% should be considered as a failure.

Evaluation Metrics

The metric used for evaluation will be the accuracy of the model for guessing new game results. The benchmark model will be generated based on the average prediction accuracy of the bookmakers. This will be compared then to the results of my model.

If we have an imbalance dataset, we should use the following performance measures for gaining more insights about the accuracy of the model (https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/) instead of the clasi accuracy metric:

- **Confusion Matrix:** A breakdown of predictions into a table showing correct predictions (the diagonal) and the types of incorrect predictions made (what classes incorrect predictions were assigned).
- **Precision:** A measure of a classifiers exactness.
- **Recall:** A measure of a classifiers completeness
- **F1 Score (or F-score):** A weighted average of precision and recall.

Project Design

I will start exploring the data and generating some basic statistics.

A useful dataset needs to be created with relevant columns. In the beginning I will focus on previous match results. Additional columns such as victory strike before the game, goals in

favor/against until date in season, or victories/draws/loses until date in season will be added with the help of some python scripts I will need to develop.

Once I have the data ready, I will train different models with cross validation, and choose the most accurate one for further analysis. I will train models using Naive Bayes (MultinomialNB) and SVMs (LinearSVC or SVC). In order to optimize the models I will use GridSearch with different parameters. I will also convert home and team into a binary feature, for example RealMadrid_home or RealMadrid_away.

If the results are not as good as expected, I will need to add more features to the dataset. One option is adding team and player stats. I am sure this will help increasing accuracy.

If the results I get from training a model in the previous step, I will try to add new seasons (2016-2017 and 2017-2018) to the dataset. The model will be train again with newly added data, and predictions will be run against the games in the latest season. If the schedule is up to date we should be able to predict upcoming games, and make some cash playing against betting systems.