"Beat the Bookie"

Practical Application of Machine Learning to a Real World Problem

COMP312P Assignment 2
Dr. Simon Julier (s.julier@ucl.ac.uk) &
Dr. Dariush Hosseini (dariush.hosseini.09@ucl.ac.uk)

Overview

Assignment Release Date: Monday 30th October, 2017
Assignment Submission Date: Friday 12th January, 2018
Weighting: 60% of module total
Final Submission Format: zip file containing Jupyter Notebook and CSV file of predictions

Assignment Description

This assignment will get you attempting to predict the outcome of English Premier League (EPL) football matches by training suitable machine learning algorithms on historic results data. You are provided with training data consisting of match information of 1,794 football matches over the past 10+ years. These matches are only those involving the current EPL teams. Your task is to use this data, along with any other data sources that you think might help, to build a machine learning model that can predict the outcome of the matches being played in the EPL on the weekend of the 20th January 2018.

You will need to build a model that predicts the value of the FTR feature which can take the values {H, D, A} indicating a home win, draw, or away win. You can use the training data along with a suitable evaluation method (i.e. splitting the training data into training, validation, and test sets) to train and validate your model. It might also be useful to bring in extra data to help improve the accuracy of your model (such as match information for games not included in the training data, manager information, player stats, distance needed to travel for the away team, etc). As an example, this extra information can be used to add prior probabilities of winning for each team. Please note: you **cannot** use betting odds as extra features for your model.

The bookies tend to get the results correct around 53% of the time, so do not be surprised if you cannot get higher predictive accuracy than this. This is effectively the "gold standard", if you can get close to this value then that is fantastic; if not, don't worry! With this in mind, I would use the bookies' odds as a measure of how well your model performs; the closer your estimated odds are to those of the bookies, the better. It is worth pointing out that bookies have a tendency to add `overhead' to their odds to protect their profits. This means that if you convert the odds to probabilities, the sum of a bookies' probability estimates for a given match will be more than 1. The overhead added to the total probability for each match is around 0.1.

The primary purpose of this assignment is not the final predictive accuracy you obtain but rather your approach to attempting this problem and the level of understanding that you show. Be creative and ask questions of this data; for example, you might want to investigate whether the referee influences the outcome and so add that as a prior into your model. You will probably want to engineer your own features for input to your classifier. Many existing approaches to football prediction use this approach so do a search of the literature to find inspiration.

The assignment submission will take the form of a Jupyter notebook (containing the source code of your approach as well as in-line documentation forming the write-up) and a CSV file containing your predictions for the matches given in the test data (along with any extra data sources you have used).

Data Description

The data is available online via the course's Moodle page. There are three files on the course webpage: epl-training.csv, epl-test.csv, and sample-submission.csv. Each of these are described below.

epl-training.csv This file contains the data that you are to train and evaluate your model on. It consists of historic match information of all the teams currently in the premier league. The features for each match are as follows:

- Date: The date that the match took place
- HomeTeam: The team playing at home
- AwayTeam: The team playing away
- FTHG: The goals scored by the home team at full time
- FTAG: The goals scored by the away team at full time
- FTR: Full time result (This is what you are predicting)
- HTHG: The goals scored by the home team at half time
- HTAG: The goals scored by the away team at half time
- HTR: The result at half time
- Referee: The name of the referee officiating the match
- HS: Total number of shots on goal by the home team
- AS: Total number of shots on goal by the away team
- HST: Total number of shots on target by the home team
- AST: Total number of shots on target by the away team
- HF: Total number of fouls committed by the home team
- AF: Total number of fouls committed by the away team
- HC: Total number of corners by the home team
- AC: Total number of corners by the away team
- HY: Total number of yellow cards received by the home team
- AY: Total number of yellow cards received by the away team
- HR: Total number of red cards received by the home team
- AR: Total number of red cards received by the away team

epl-bookies.csv This file contains the odds for each of the games in the training data from 6 bookmakers. This data **cannot** be used in your final model as it will heavily bias your results. However, you can compare you model against the predictions obtain by the bookies. This could be particularly useful if you are using a classification approach that outputs a probability for each of the 3 possible results as it will allow you to compare your predicted odds with those of the bookies (just remember about the bookies' overhead).

epl-test.csv This file contains the data that will be used to perform the final predictions which form part of your submission. Note that this file does not have all of the features that are in the training file (obviously because information regarding the games themselves are not available). This means that you may want to engineer features or prior probabilities using your training data.

sample-submission.csv This file shows you what format your final predictions that form part of the overall assignment submission should be in.

Getting Help

You are encouraged to use the assignment discussion forum to help when you get stuck. Please check the forum regularly and try and answer each other's questions. I will be checking the forum as often as I can and will be answering any questions that have remained unanswered.

Some points to help you get started:

- You might want to use the features not available in the test data to build prior probability estimates for each team or team pairing that you can then include as priors in your final model. You could also draw in other data sources to help improve these prior probability estimates.

- You can obtain up to date match data in the same format as the training data from http://www.football-data.co.uk
- You will probably want to convert the categorical features into multiple binary features (e.g. if a categorical feature, f, can take the values {A, B, C}, then you would introduce the binary features f_A, f_B, f_C such that if f takes the value of A, then f_A=1, f_B=0, f_C=0). This will enable you to use many of the existing machine learning algorithms with the data.
- You might want to engineer your own set of features. As a simple example, you could use the training data to work out for each team how many shots they have had in the past *k* number of matches. You could then take these two statistics for the home team and away team and train your classifier on them. Then, for each of the teams in the testing set you could compute the same statistics and make predictions on them.
- This assignment shares many similarities to the March Mania Kaggle competitions where the task is to predict the outcome of the NCAA basketball tournament. If you are stuck, have a look at how people have approached that problem but **do** reference any work that you have taken inspiration from.
- There are lots of papers available online that detail different approaches to this problem. It is worth spending some time at the start of the project doing background research and getting a feel for the data.
- You **can** use existing libraries such as scikit-learn to provide implementations of key algorithms. I do not expect you to write your own versions of individual algorithms.
- You **do not** have to use Azure Machine Learning as your platform for working on this project. If you prefer, you can use a local install of Jupyter.
- All source code should be written in Python.

Notebook Submission Format and Structure

A notebook should follow the following structure:

1. Introduction
   A brief description of your approach to the problem and the results that you have obtained on the training data.

2. Data Import
   This section is how you import the data into the notebook. It should be written in such a way that I can modify it to run on my own machine by simply changing the location of the training data and any additional data sources that you have used.

3. Data Transformation and Exploration
   Any transformations that you apply to the data prior to training. Also, any exploration of the data that you performed such as visualization, feature selection, etc.

4. Methodology Overview
   Start by describing in broad terms your methodology. Include any background reading you may have done and a step by step description of how you have trained and evaluated your model. Describe any additional data sources that you have used. If you had attempted different approaches prior to landing on your final methodology, then describe those approaches here.

5. Model training/validation
   This contains a breakdown of how your model was trained and evaluated.

6. Results
   Here you show the results that you obtain using your model on the training data. If you have multiple variations or approaches, this is where you compare them.

7. Final predictions on test set
   This is the section where you perform your final predictions on the test set using the model that you have trained in the previous section.

Keep in mind that your notebook should be written in such a way that I can modify the location of the data and then step through your notebook to obtain the same results as you have submitted.

Marking Guidelines

All reports will be marked against the marking rubric, which is downloadable from the course's Moodle page. The mark weighting for each section is as follows:

1. Methodology (40%)
   How well is the methodology described? How appropriate is it to the task at hand? Have any extra data sources been used and if so are they useful? Have you done more than just apply a classifier to the training data?

2. Evaluation Strategy (30%)
   Has a suitable evaluation strategy been used so as to avoid any possible bias? If your methodology contains multiple parameters, how have the final parameter values been chosen? Have you used any form of cross validation?

3. Presentation of Results (10%)
   Have you presented results on the training data? Are the results presented appropriate and displayed in an easy to interpret manner? Do they reveal any extra insights about how your model performs?

4. Interest of Approach (10%)
   How interesting and novel is your approach (regardless of predictive accuracy)? Have you used extra data sources, or transformed the training data, in an interesting way? Have you done something that is beyond simply using a standard classifier on the training data?

5. Format, structure, referencing, and clarity of writing/code (10%)
   Is your final notebook well laid out and does the write-up follow a clear structure? Have you included any references to show background research/reading? Is your writing free from spelling, punctuation, and grammatical errors and is your code well commented?

For a more detailed breakdown of what constitutes a good (and bad) mark for each of these sections please refer to the marking rubric.

Submission and Feedback

The deadline for submission is **Midday on the 12th January 2018**. You will submit a zip file containing your Python notebook as a `.ipynb` file along with your predictions on the test data as a csv (the format of which is described above). Please also include in the zip file any extra data that you have used to build your model.

The reports will be marked and feedback given by February 19th 2018.