UNIVERSITY COLLEGE LONDON
Faculty of Engineering Sciences

Department of Computer Science

COMP0036: Group Assignment

BEAT THE BOOKIE

Matthew Caldwell & Dariush Hosseini

Overview

- Assignment Release Date: Friday 4th November 2022
- **Assignment Hand-in Date: Monday 19th December 2022 at 4pm (GMT)**
- Weighting: 40% of module total
- Submission Format: A zip file containing:
  - a PDF file of the written report
  - a Jupyter Notebook .ipynb file of the code
  - a CSV file containing your predictions for the matches given in the test data

Please note that this is a group assignment. As a first step, please arrange yourself into groups of 5 or 6 and then register your group on the Group Choice section of the module Moodle page.

Assignment Description

This assignment prompts you 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 football match information over the past 10+ years.

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 14th January 2023.

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 and validation 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).

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!

You will primarily be assessed not so much on the final predictive accuracy which you obtain, but rather on your approach when attempting this task, and on the level of understanding which you display. 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. Most of all, please reflect upon what you have learned during the term, and seek to apply machine learning in a way that is appropriate for this task.

The assignment submission will take the form of a zip file containing:

1. A PDF file containing a report of the task.
2. A Jupyter notebook containing the **Python** source code of your approach as well as (brief) in-line documentation.
3. A CSV file containing your predictions for the matches given in the test data.

Data Description

The data is available online via the module's Moodle page under the 'Group Assessment' link, and comprises three files: `epl-training.csv`, `epl-test.csv`, and `sample-submission.csv`, 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-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 Started

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 `https://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.
- All source code should be written in Python.

Submission Format & Structure

The assignment submission will take the form of a zip file containing:

1. A **PDF file** containing a **report** of the task (this should be at most 10 A4 sides in length, including references). This PDF should be prepared using the LaTeX files included on the module Moodle page under the 'Group Assessment' link, which provide a format similar in style to "preprint" publications such as arXiv.
2. A **Jupyter notebook .ipynb file** containing the **Python source code** of your approach as well as (brief) in-line documentation. The notebook should include an analysis of the performance of your classifier on the data from the test set file.
3. A CSV file containing your predictions for the matches given in the test data.

The PDF report should adopt 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 Transformation & Exploration**

   Any transformations that you apply to the data prior to training. Also, any exploration of the data that you performed such as visualisation, feature selection, etc.

3. **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.

4. **Model Training & Validation**

   This contains a breakdown of how your model was trained and evaluated.

5. **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.

6. **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.

7. **Conclusion**

   This is the section where you consider your findings and suggest avenues for future research.

The Notebook should adopt the following structure:

1. **Introduction**

   A brief précis of the equivalent section in your report.

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 & Exploration**

   Code for the equivalent section in your report, together with in-line documentation of that code.

4. **Methodology Overview**

   Code for the equivalent section in your report, together with in-line documentation of that code.

5. **Model Training & Validation**

   Code for the equivalent section in your report, together with in-line documentation of that code.

6. **Results**

   Code for the equivalent section in your report, together with in-line documentation of that code.

7. **Final Predictions on Test Set**

   Code for the equivalent section in your report, together with in-line documentation of that code.

Note:

- Your notebook need only contain brief in-line documentation, while the PDF should contain a more detailed description.
- You will be assessed primarily on the contents of your PDF report. The notebook and CSV file are required so that we can check that your results are replicable.
- Keep in mind that your notebook should be written in such a way that we 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 available online via the module's Moodle page under the 'Group Assessment' link.
The mark weighting for each section is as follows:

- **Methodology (15%)**

  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?

- **Evaluation Strategy (15%)**

  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?

- **Presentation of Results (15%)**

  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?

- **Interest of Approach (40%)**

  How interesting and novel is your approach (regardless of predictive accuracy)? Have you used any 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?

- **Format, structure, referencing, and clarity of writing/code (15%)**

  Are your final notebook and report 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.