Fantasy Premier League Project Report General Assembly-Data Science DAT3

-Prathmesh Mengane

08

**Fall**

* **Problem statement and hypothesis**

I’m developing a Model to predict the total Fantasy Points scored by a player in the Fantasy Premier League Soccer (FPL) each week.  
 FPL is an online fantasy game for English Premier Soccer League. In this league a user selects a total of 15 players (11 starters + 4 substitutions) from the given pool of 624 players. Each Player has been assigned a value based on his current form and each user has been given a budget of £100m to select the 15 players in his team.

The points scored by each player depend on upon various attributes of the player in that particular game and each carries different weightage. For example a forward player gets rewarded 4 points for every goal he scores and 3 points for every assist. The total points scored in each game by each player are a cumulative sum of all this attributes.   
(Reference: Appendix B explaining how the event score is determined)

My hypothesis is that given the available player data every week, I can predict points scored for the coming week with enough accuracy, that I would personally use my model as a tool for player selection.

More Information on Fantasy Premier League can be found here: <http://fantasy.premierleague.com/rules/>

* **Description of your data set and how it was obtained**

I am extracting the data for each player from the official API of the FPL site (<http://fantasy.premierleague.com/web/api/elements/266/>).   
The API has data in the JSON format. I used the *requests* method learned in the class to grab data from the API and then feed it into a Dictionary.

* **Description of any pre-processing steps you took**

I created *for loop* to extract data from the API for all the 624 players and saved it an empty list. I used a random sleep parameter, to avoid being blocked by the API. Then I wrote the contents of the List into a CSV file (dict\_output1.csv) using *csv.DictWriter*. Then I read the data from the CSV file into a Panda’s Data Frame.

I decided to narrow down the scope of the project by only focusing for the players playing in Forward positions instead of goalkeepers, defenders and midfielders. Hence, I filtered the data from the original CSV containing all the players using condition *players\_df.type\_name==’Forward’* .

I then feed that data into a new CSV (players\_updated.csv) containing only the forward players, which constituted to 107 players as compared to previous 624 players and it still has same 58 columns.

I feed that CSV into a new Pandas Data Frame, containing only the Forwards.

I later realized from data exploration that some of the players never played any minutes and hence weren’t contributing to the model. Thus, I decided to remove those players by sorting on criteria   
*forwards\_df.minutes> 0*

I then feed that data into a new CSV (regular\_forwards.csv) .This new data set if further implemented, by reading the CSV into a pandas dataframe called regular\_forwards\_df.

* **What you learned from exploring the data, including visualizations**

I’ve narrowed down my data set to only contain data for Forwards. I observe that some of the players don’t have any significant data related to them because they haven’t been playing most of the weeks. (These are termed as “fringe players” in soccer, the one’s are used as backup if the star players get injured and hence they don’t get to play much throughout the season.)   
I wondered if they were acting as anomalies and hence making the model weak. I removed this players “*total\_minutes*” played condition as explained above.

Other than that most of my data seemed to be clean in nature. It doesn’t have any missing data or Na data for the features I’m concentrating on.

I explored the data further within the pandas data frame by reading the head and tail values to see if the data was consistent. I observed the data types of the columns I was interested in to see if I needed to alter any of them , which wasn’t the case.

* **How you chose which features to use in your analysis**

I have total of 58 columns/features in my data set. I initially started by eliminating the obvious ones, which don’t provide any insight to my model/prediction like “*photo*”, “*team\_name*”, “*current\_fixture*” etc.

I then eliminated the variables which I knew had direct relationship with the Y variable. Meaning the variables, which are used to calculate the “*event\_total*” score for a particular player. These variables are “*goals\_scored*”, “assists”.  
(Reference: Appendix B explaining how the event score is determined)

I tried plotting some scatter plots to observe any relationship between remaining features and the Y.

Eventually using my domain knowledge & learning from data exploration, I started to create a model with 6 features.  
(Reference: Appendix A explaining those 6 features).

Given the multi co-linearity graph I learned that there is co-linearity between “value\_form” and “form” & also between “value\_season” and “form”. Thus I made two interaction variables between these variables.

*regular\_forwards\_df['interaction\_term1'] = regular\_forwards\_df.value\_form \* regular\_forwards\_df.form*

*regular\_forwards\_df['interaction\_term2'] = regular\_forwards\_df.value\_season \* regular\_forwards\_df.form*

* **Details of your modeling process, including how you selected your models and validated them**

Because I’m predicting a continuous variable as my output, I decided to use Linear Regression Model for my project.

I initially created a linear regression model from the Forward players data-frame using 6 variables

forwards\_model = smf.ols(formula='event\_total ~ selected\_by +   
value\_form + value\_season + form + ea\_index + bps', data=forwards\_df).fit()

I then got an error regarding multicollinearity. Thus, I created a Scatter Matrix and a Correlation Matrix plot for the 6 variables.

I found that there was a high correlation between “ea\_index” and “bps”. Eventually by testing the model I realized that bps was worsening my prediction metrics, hence I decide to drop it. (Reference: Appendix C showing Correlation Matrix Plot)

I then applied the “train\_test\_split” module to divide my data into train and test set. But then I got ValueError when I try to run my model on the train set. After exploring this error, I realized it was due to the “train\_test\_split” module. Hence I decided to use a different method using numpy’s “randn” function to create my test and train set.

I then decided to perform Cross Validation over five folds using Linear Regression model. I made the model using the 6 input variables I identified including the interaction terms and set “event\_total” as my y variable.

I used Root Mean Squared Error (RMSE) as my evaluation metric. The RMSE gave the average over 5 folds, of the overall error of the predictions against the actual values.

I fine-tuned the model then to get the RMSE as lower as possible.

I also tried Random Forest Classifier Model over the same data. And performed Cross Validation over 5 folds. But I realized that I got better result using Linear Regression, and hence I decided to use that for my final model.

I had total of 4 sets of game week data containing player attributes for all those four game weeks i.e Nov data, Dec 1 data, Dec 4 data and Dec 9 data.

So I first fitted my model using the November data, thus using the Nov data set as my train set. I then Predicted for the remaining 3 game weeks of December, thus treating them as Test data.

The RMSE for those 3 game weeks were :   
Dec 1 week RMSE = 2.3746  
Dec 4 week RMSE = 2.8351  
Dec 9 week RMSE = 3.0285

* **Your challenges and successes**

My biggest challenge initially was feature selection. I had large number of X variables/features to select from ie 58 features. I’ve then narrowed down some of them based on my domain knowledge. I then used some techniques learned like using p value, studying linear regression results for each individual variable, to narrow down the variables.

Some of the machine learning algorithms were new to me so I wasn’t initially sure which would be a best fit for my data set. So I study them all and tried few of them to compare the results of the models.   
  
There have been few successes along the way, I am proud of learning how to get a large data set from a public API without being recognized as a bot and being denied access.   
And how to store the large data set into a dictionary and data frames and then to perform functions on top of that.

Eventually, after fine-tuning my model I was happy that I could bring down by RMSE within 2.x range.

* **Possible extensions or business applications of your project**

Once I’ve the model working fairly accurately for the Forwards, then I plan on extending the project in the future for the other player types (defenders, midfielders and goal-keepers). If I can predict for all the types then I predict an entire team of players expected to do well in next weeks round of matches.

Overall the Online Fantasy Sports market is a growing industry. It’s a

huge market especially In Europe and Asia for English Premier League Soccer. . With more than 2.5 million registered users for FPL, the fantasy league is a million dollar (if not billion) industry. This can be a good guide for users to pick their teams with the highest total score potential.

* **Conclusions and key learnings**

I was able to create a data science model using linear regression to predict the total points scored for a forward player with some level of accuracy. The RMSE was in the 2.x range, and I think if I can further fine-tune my model I can get RMSE further down.

Throughout the process of this project I learned the entire Data Science workflow. I was proud of learning the workflow and implementing it using a real world data set and being able to create a prediction model on top of it.

The steps of obtaining data and data cleaning were time consuming but I had some experience with it, which helped. The steps of Data Exploration/Data Mining using python was were intriguing to learn and implement. And the process of Model Creation & using cross validation for Model Evaluation were very fascinating & fun to learn and implement.

**Appendix A:**

Variable Names Explanation.

|  |  |
| --- | --- |
| Variable | Explanation |
| event\_total (Y variable) | Total points scored by that player in that week (event) |
| selected\_by | The % of users who has that player in that team |
| value\_form | Custom index denoting value of player compared to current form of the player |
| value\_season | Custom index term denoting value of the player compared to the form of the player based on all the previous rounds so far |
| form | Custom index term denoting the current form of the player |
| ea\_index | Index created by the EA org. and used for ranking/comparing players, and its also changes based on the current performances of the player |
| bps | Index denoting the ranking of players based on the bonus points scored by the players |
| now\_cost | The cost of the player for that game week. This variable is affected by the form of the player. |

**Appendix B:**

Scoring Rules



**Appendix C:**

Correlation Matrix denoting correlation between ea\_index and bps.

