Methodology (Fantasy Football)

Akhil Gupta, IIT Roorkee
akhilg.iitr@gmail.com

Problem Description: Data of players playing in the English Premier League for the past 3 years had been given. Their names, teams, and the points that they secured at the end of each game week were provided. Using that, it was asked to forecast their scores for the season 2016-17, and with a fixed budget of 100 Million, build a dream team of 15 players. Their cost for the season 2016-17 was provided for this.

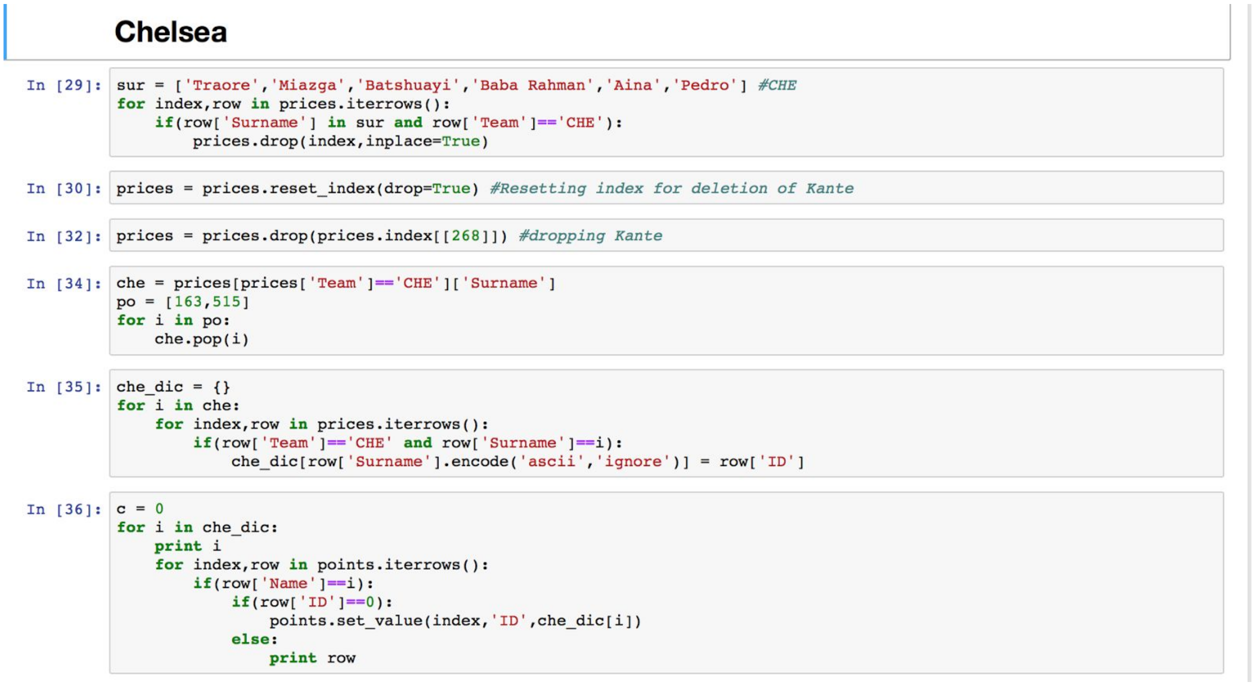
Solution Approach: As the scores for the previous 3 years were present in substantial amount, classical approach of Time Series Forecasting was adopted for prediction of scores, which was followed by use of Linear Programming to maximise the total points within a fixed budget, and other constraints on players in each position.

Assumptions:

- ❏ Scores have been forecasted for only those players whose historical data was available, and who are playing in season 2016-17.
- ❏ 2015 data was simulated from the actual data of 2013 and 2014, and it could be clearly seen that the number of matches played have been kept same. Following that trend only, it was assumed that the player would play as many matches as he played in the last season which he played in Premier League. (It could be any of the 3 years)
- ❏ More weightage was given to the data of 2013 and 2014, as data for 2015 was found to be inflated to some extent.

Step-Wise Approach:

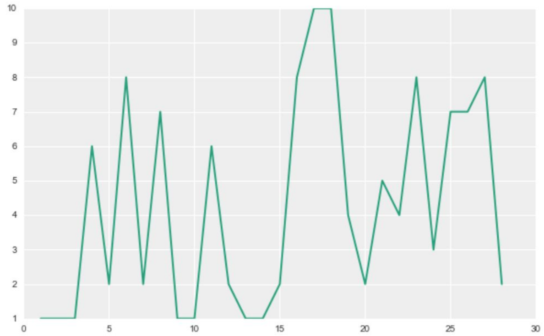
- **Data Preprocessing -** First, there were some missing values in the first names of some players. They were filled with '-' to avoid any errors in further analysis due to encounter of a null character. Cost was scaled down to millions by dividing by 1,000,000.
  Data from the three different sheets was merged into one. All this was done after importing the data into *Pandas DataFrame in Python*.
- **Data Integration -** There was no column for merging the price_data with the historical_points_data. So, for each team, the players who have no historical data were separated out by giving the ones who have historical data, a unique 'ID'. This ID was then, used for mapping the points data to the price data. Out of 584 players, only 326 players had historical data available. Forecasting of scores for the 258 players wasn't possible. Once all the data was present in one DataFrame, predictive modeling was done for each player.

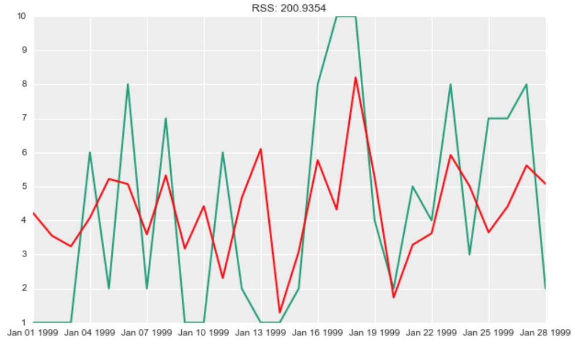


Chelsea

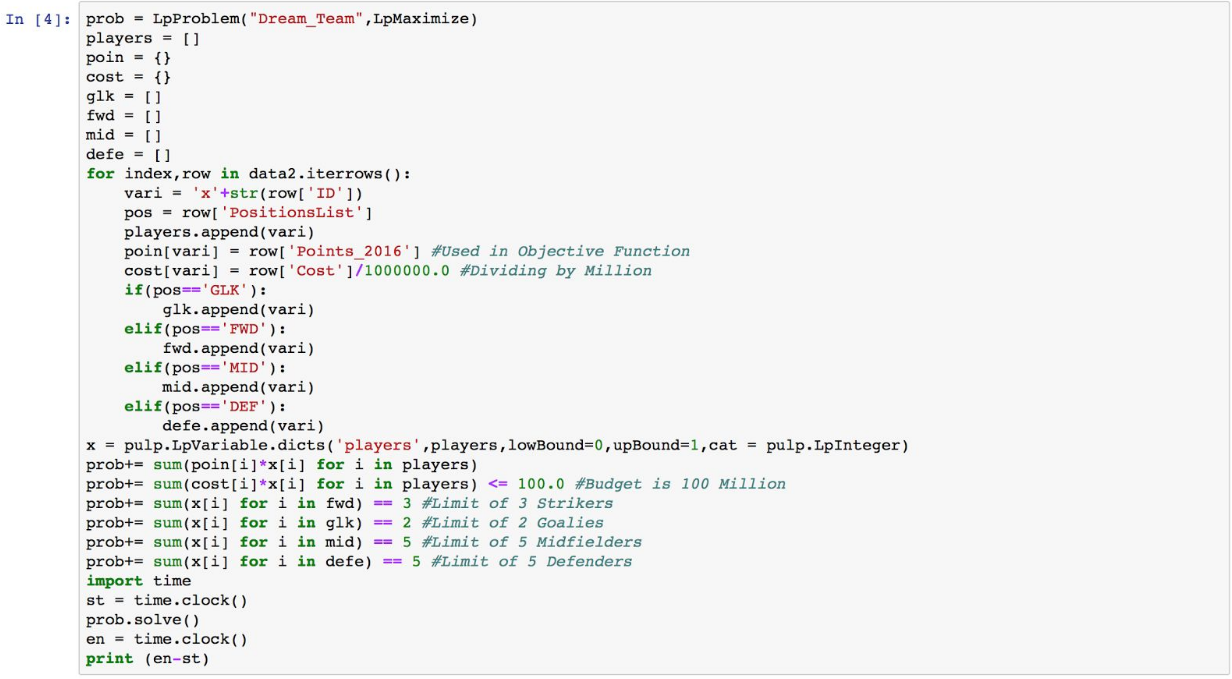
```
In [29]: sur = ['Traore','Miazga','Batshuayi','Baba Rahman','Aina','Pedro'] #CHE
         for index,row in prices.iterrows():
             if(row['Surname'] in sur and row['Team']=='CHE'):
                 prices.drop(index,inplace=True)

In [30]: prices = prices.reset_index(drop=True) #Resetting index for deletion of Kante

In [32]: prices = prices.drop(prices.index[[268]]) #dropping Kante

In [34]: che = prices[prices['Team']=='CHE']['Surname']
         po = [163,515]
         for i in po:
             che.pop(i)

In [35]: che_dic = {}
         for i in che:
             for index,row in prices.iterrows():
                 if(row['Team']=='CHE' and row['Surname']==i):
                     che_dic[row['Surname'].encode('ascii','ignore')] = row['ID']

In [36]: c = 0
         for i in che_dic:
             print i
             for index,row in points.iterrows():
                 if(row['Name']==i):
                     if(row['ID']==0):
                         points.set_value(index,'ID',che_dic[i])
                     else:
                         print row
```

- **Time-Series Modeling:** Points for players were analysed with respect to time using plots, and it was observed that there were seasonal trends in their performances, and some high/low random peaks. Best method of dealing with such time series is ARIMA (Autoregressive Integrated Moving Average) method. It takes into account, both the trend and the seasonality of the time series.
It is important to choose the AR and MA coefficients wisely in ARIMA analysis. For the same, ACF and PACF plots i.e. Autocorrelation Function and Partial Autocorrelation Function were used. A threshold is defined, and the value where the curve meets the threshold is set as the value of coefficient. ACF is used for MA and PACF is used for AR. Using these values, predictions for the next matches that the player will play were made. All this was done using the statsmodels library in Python. As Fig. 3 shows, the red line corresponds to the time-series according to ARIMA, whereas, green line refers to the actual scores. Red one tries to sync with the green one. Predictions are made using ARIMA.predict() which forecasts the future (using 'red' line).





- **Linear Programming:** Once the points for season 2016-17 were obtained for all the 326 players whose historical data was available, it was desired to find a dream team consisting of 2 goalkeepers, 5 defenders, 5 midfielders and 3 strikers, which **maximises the total points**, in a budget of 100 Million Pounds. The pulp library in Python was used for this purpose.



```
In [4]: prob = LpProblem("Dream_Team",LpMaximize)
        players = []
        poin = {}
        cost = {}
        glk = []
        fwd = []
        mid = []
        defe = []
        for index,row in data2.iterrows():
            vari = 'x'+str(row['ID'])
            pos = row['PositionsList']
            players.append(vari)
            poin[vari] = row['Points_2016'] #Used in Objective Function
            cost[vari] = row['Cost']/1000000.0 #Dividing by Million
            if(pos=='GLK'):
                glk.append(vari)
            elif(pos=='FWD'):
                fwd.append(vari)
            elif(pos=='MID'):
                mid.append(vari)
            elif(pos=='DEF'):
                defe.append(vari)
        x = pulp.LpVariable.dicts('players',players,lowBound=0,upBound=1,cat = pulp.LpInteger)
        prob+= sum(poin[i]*x[i] for i in players)
        prob+= sum(cost[i]*x[i] for i in players) <= 100.0 #Budget is 100 Million
        prob+= sum(x[i] for i in fwd) == 3 #Limit of 3 Strikers
        prob+= sum(x[i] for i in glk) == 2 #Limit of 2 Goalies
        prob+= sum(x[i] for i in mid) == 5 #Limit of 5 Midfielders
        prob+= sum(x[i] for i in defe) == 5 #Limit of 5 Defenders
        import time
        st = time.clock()
        prob.solve()
        en = time.clock()
        print (en-st)
```

- **New Features:** As we had only feature i.e. total points to compare players, some new ones were made.
    - ❑ **PPM(Points per match)-** This feature tells the average points a player can be expected to score in a match. The higher, the better.
    - ❑ **CPP(Cost per point)-** This feature helps in finding the most valuable players i.e. which are cheaper and give higher points. The lower, the better.
    - ❑ **CPI(Cost-point index)-** It is a combination of PPM and CPP, wherein, both are first scaled down to lie between 0 and 1, and then, added by giving weightage of 0.5 to each. This was calculated for only those players who are expected to play more than 20 matches. It was important to define a threshold there.

DreamTeam.png: This infographic illustrates the dream team which was obtained after solving the LPP. It gives the maximum score possible within the budget which comes out to be **3718 points**.
Just for the sake of illustration, a formation of 4-4-2 has been shown on the football field. The 11 are chosen from 15 by simply using PPM, i.e. the higher the player scores, the better he is.
Age, height and weight were found out from Wikipedia, just to get a rough idea of an average individual in the dream team. The home teams of the players were also listed to know the countries which are producing this talent. Majority of the players are from England and Belgium.

Most valuable players were found out using the CPI index, and the ones for the long race were the ones who are expected to play the most number of matches.
A player’s loyalty can be seen by the number of years he has been playing for the current team. Average period came out to be around 5 years which definitely shows, the more you play for the same team, the better you are expected to perform for them.
Under each player, player's name, team and PPM index are mentioned.

FPL Insights.png: In this infographic, top 3 of each position (apart from the dream team) have been listed as the players to watch out for. They are expected to do wonders for their team and they were chosen on the basis of CPI index, which takes into account both the cost and the performance. Also, boxplots between Position and CPI, and Team and CPI were plotted to show some interesting insights. On an average, goalkeepers do good, followed by defenders and the maximum variation can be seen among the forwards.
Similarly, for the teams, Hull City gives a low CPI whereas, Chelsea and Arsenal contribute significantly to the higher CPI bracket.

Softwares used:

- ➔ Pandas (py)
- ➔ Seaborn (py)
- ➔ Matplotlib (py)
- ➔ Numpy (py)
- ➔ Scistats (py)
- ➔ Pulp (py)
- ➔ Jupyter Notebook (py shell)
- ➔ Canva - for making infographic