# Machine Learning Engineer Nanodegree

## Capstone Proposal

### Dhrubojyoti Jasu

October 16, 2018

### `Proposal`

### Domain Background

For this capstone project, I will use a English Premier League Football dataset from http://football-data.co.uk/englandm.php . This is a binary classification problem similar to the `CharityML` in Supervised Learning section of the MLND. The legal sports-betting market in the U.S. was worth an estimated USD270 million in 2017 -- with another USD2.5 billion to USD3 billion in black market betting, according to research firm Eilers & Krejcik Gaming, LLC.

Even though it is not legalized in many other countries like India, but for my own interest, I want to build a predictive model capable of predicting if the home team will win a football match. Usually betting is conducted with human instincts but now we can use some machine learning algorithm to predict the result of the future matches also.

There are reports related to sports prediction using machine learning. Some of them I have listed below:

- Using Machine Learning to Predict the Outcome of English County twenty over Cricket Matches
- How I Used Machine Learning to Predict Soccer Games for 24 Months Straight
- Predicting Football Results With Statistical Modelling

I am a regular viewer of football around the globe, my favorite club is Manchester United and I follow English Premier League religiously and now I am excited I can use my knowledge of machine learning to have some fun with data.

### Problem Statement

The problem is to use the existing dataset of EPL obtained thorugh http://football-data.co.uk/englandm.php and use it to train some supervised learning algorithms to predict the matches of EPL. I want to predict whether a home team is gonna win the match by training some supervised learning algorithms.

### Datasets & Inputs

The dataset for this project is obtained through http://football-data.co.uk/englandm.php . The datasets have been attached as separate files in the repository in `Capstone\Datasets` folder. A text file containing the description of the data also provided from the site origin itself. Columns related to betting statistics are missing but playing statistics are all present. I have downloaded for the last 18 sessions data and consolidated into a single dataset called `My_Capstone_Dataset.csv`. All datasets can be accessed from here. A detail explanation of every feature I will choose will be well documented in my final capstone project report.

**Lets preview the dataset**

In [10]:

```
#import necessary dependencies
import pandas as pd
from IPython.display import display
%matplotlib inline

# Read the data into a dataframe
data = pd.read_csv('My_Capstone_Dataset.csv')

#drop the redundant data
data.drop(['Unnamed: 0'],1 , inplace=True)
display(data.head())
```

|   | Date | HomeTeam | AwayTeam | FTHG | FTAG | FTR | HTGS | ATGS | HTGC | ATGC | ... | HTLossStreak5 | ATWinStreak3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 19/08/00 | Charlton | Man City | 4 | 0 | H | 0 | 0 | 0 | 0 | ... | 0 | 0 |
| **1** | 19/08/00 | Chelsea | West Ham | 4 | 2 | H | 0 | 0 | 0 | 0 | ... | 0 | 0 |
| **2** | 19/08/00 | Coventry | Middlesbrough | 1 | 3 | NH | 0 | 0 | 0 | 0 | ... | 0 | 0 |
| **3** | 19/08/00 | Derby | Southampton | 2 | 2 | NH | 0 | 0 | 0 | 0 | ... | 0 | 0 |
| **4** | 19/08/00 | Leeds | Everton | 2 | 0 | H | 0 | 0 | 0 | 0 | ... | 0 | 0 |

5 rows × 42 columns

**Lets explore this dataset a little**

In [6]:

```
# Total Number of Matches
n_matches = data.shape[0]

# Calculate number of features. -1 because we are saving FTR as the target variable
(win/lose/draw)
n_features = data.shape[1] - 1

# Calculate matches won by home team.
n_homewins = len(data[data.FTR == 'H'])

# Calculate win rate for home team.
win_rate = (float(n_homewins) / (n_matches)) * 100

# Print the results
print ("Total number of matches: {}".format(n_matches))
print ("Number of features: {}".format(n_features))
print ("Number of matches won by home team: {}".format(n_homewins))
print ("Win rate of home team: {:.2f}%".format(win_rate))
```

```
Total number of matches: 6080
Number of features: 42
Number of matches won by home team: 2816
Win rate of home team: 46.32%
```

## Solution Statement

I will be using `Supervised Learning` approach for this binary classification problem. I will train algorithms like `Logistic Regression`, `Random Forest`, `AdaBoost`. Specifically I want to use some **Ensemble Learning** algorithm for this problem for several reasons because multiple learners are employed to build a stronger learning algorithm. It works by choosing a base algorithm (decision trees) and iteratively improving it by accounting for the incorrectly classified examples in the training set.

## Benchmark Model

The benchmark score I will be comparing against an untuned `Logistic Regression` Model. I will train and test this model on same data on which I am going to build my tuned prediction model.

## Evaluation Metrics

I will be using `accuracy_score` & `fbeta_score` from sklearn.metrics to evaluate both the benchmark and my final model. Goal of this project is to predict win of 'Home Team' accurately. So **accuracy** as a metric to evaluate a model's performance is appropriate. However, predicting a team is not going to win is not that much important, hence, a model's ability to precisely predict the win of a 'home team' is *more important* than the model's ability to recall those teams.

$$ F_{\beta} = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\left( \beta^2 \cdot precision \right) + recall} $$

For this reason, we can use **F-beta score** as a metric which considers both `precision` & `recall`.

## Project Design

**Workflow**

- **Data Exploration**

Here I will explore some basic staistics on the data like win percentage of home team on the whole dataset. I will explain all details on relevant features and will outline which features I am going to take for training of my algorithms.

- **Data Pre-Processing**

I will pre-process the data by seperating the feature set and the target variable `FTR`. Before the data can be used by any machine learning algorithms, it should be cleaned, formatted and restructured. Pre-processing can tremendously affect the performance of machine learning algorithms, so here I will do this. Also I will shuffle & split the data into training & testing set in this data pre-processing step.

- **Evaluating performance of the model**

I will use the metrics mentioned above here to evaluate the performance of my benchmark model and several other models. Here I will design a training & predicting pipeline like we did in `Finding Donor` project. I will print `f-beta` & `accuracy` score for each model and will choose the best one w.r.t the metrics generated.

- **Tuning the parameters of the choosen model**

In this step I will do a hyper parameter tuning by using sklearn GridSearchCV method. Then I will report the final score of evaluation metrics choosen earlier. By tuning the hyper parameters I guess definitely I can achieve better performing model.

- **Prediction**

In this step I will perform prediction on my testing dataset of 2018-19 session of EPL dataset with the help of my tuned classifier.