MID PROJECT CHECK- IN
MSDS 696- PRACTICUM 2
REGIS UNIVERSITY

ANDREW LUJAN

CONTENTS

RESEARCH PROBLEM

WITH THE PREMIER LEAGUE BECOMING MORE COMPETITIVE WITH MORE MONEY BEING INTRODUCED TO THE LEAGUE VIA NEW OWNERSHIP GROUPS IT'S IMPORTANT TO LEVERAGE DATA WHEN BUILDING A TEAM. USING DATA, I WANT TO CREATE A MODEL THAT CAN PREDICT A MATCH-WINNER BASED ON MATCH DAY DATA AND OTHER FEATURES GENERATED FROM PREVIOUS RESULTS.

RESEARCH QUESTION

- What kind of match day data is important to predicting a match-winner?
- What's the age profile of a championship squad?

DATA

Data will be pulled from the following sources:

Match-prediction: http://football-data.co.uk/englandm.php

Age-Profile : FB Ref: https://fbref.com/en/comps/9/Premier-League-Stats

PROPOSED METHODOLOGY

- DATA IMPORT
- DATA CLEANING/FEATURE ENGINEERING
- JOINING DATASETS
- EDA
  - CORRELATION/HEATMAPPING
  - DISTRIBUTION EXPLORATION
- MODELING/MACHINE LEARNING
  - PREDICTING MATCH WINNERS USING LOGISTIC REGRESSION, SUPPORT VECTOR MACHINE, RANDOM FOREST.
  - PERFORMANCE OPTIMIZATION
- SQUAD AGE PROFILE
  - DATA VISUALIZATION

READJUSTED TIMELINE

- Weeks 1-2: Proposal formation/Readjustment
- Week 3: Data Import, Feature Engineering
- Week 4: Joining Datasets/EDA
- Week 5: Model-building
- Week 6: Optimization
- Week 7: Squad Analysis/Data Visualization
- Week 8: Presentation creation

DATA IMPORT

```
[3]: # Reading in datasets

location = "/Users/drewsdesktop/Desktop/Data Science/Regis Classes/MSDS 696- Final Practicum/Datasets/"

prem_2011 = pd.read_csv(location + "11_12.csv")
prem_2012 = pd.read_csv(location + "12_13.csv")
prem_2013 = pd.read_csv(location + "13_14.csv")
prem_2014 = pd.read_csv(location + "14_15.csv")
prem_2015 = pd.read_csv(location + "15_16.csv")
prem_2016 = pd.read_csv(location + "16_17.csv")
prem_2017 = pd.read_csv(location + "17_18.csv")
prem_2018 = pd.read_csv(location + "18_19.csv")
prem_2019 = pd.read_csv(location + "19_20.csv")
prem_2020 = pd.read_csv(location + "20_21.csv")
prem_2021 = pd.read_csv(location + "21_22.csv")

[4]: prem_2017.tail()
```

FEATURE ENGINEERING

```
# Get goals conceded by team and Matchweek
def get_goals_conceded(playing_stat):
    # Create a dictionary with team names as a key
    teams = {}
    for i in playing_stat.groupby('HomeTeam').mean().T.columns:
        teams[i] = []

    # Value that goes with the key is a list with the match location
    for i in range(len(playing_stat)):
        ATGC = playing_stat.iloc[i]['FTHG']
        HTGC = playing_stat.iloc[i]['FTAG']
        teams[playing_stat.iloc[i].HomeTeam].append(HTGC)
        teams[playing_stat.iloc[i].AwayTeam].append(ATGC)

    # Create a dataframe for goals conceded
    # Rows will represent teams and Columns will represent the matchweek
    GoalsConceded = pd.DataFrame(data=teams, index = [i for i in range(1,39)]).T
    GoalsConceded[0] = 0
    # Aggregate to get uptil that point
    for i in range(2,39):
        GoalsConceded[i] = GoalsConceded[i] + GoalsConceded[i-1]
    return GoalsConceded
```

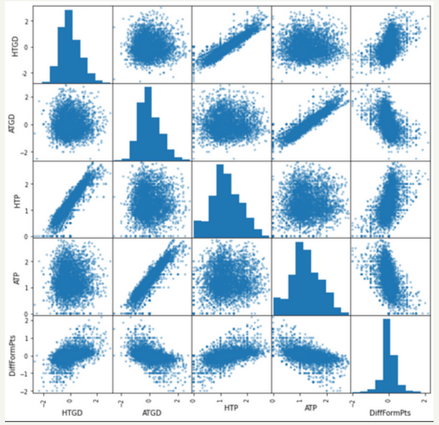
DATASET AGGREGATION

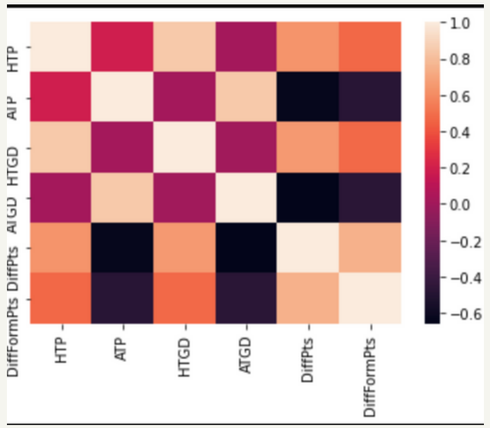
Aggregating all of the information into one table.

Next, I need to collect all of the statistical information that we've created and aggregate it into one table.

```
[30]: game_stats = pd.concat([game_stats1,
                              game_stats2,
                              game_stats3,
                              game_stats4,
                              game_stats5,
                              game_stats6,
                              game_stats7,
                             game_stats8,
                              game_stats9,
                              game_stats10],
                               ignore_index=True)
```

EDA: SCATTER MATRIX/HEATMAPS





MODEL BUILDING

```python
# Initialize Logistic Regression
clf_A = LogisticRegression(random_state = 42)

#Initialize SVM
clf_B = SVC(random_state = 912, kernel='rbf')

#Initalize Random Forest
clf_C = RandomForestClassifier(random_state = 82)

train_predict(clf_A, X_train, y_train, X_test, y_test)
print('')
train_predict(clf_B, X_train, y_train, X_test, y_test)
print('')
train_predict(clf_C, X_train, y_train, X_test, y_test)
print('')
```

PERFORMANCE EVALUATION

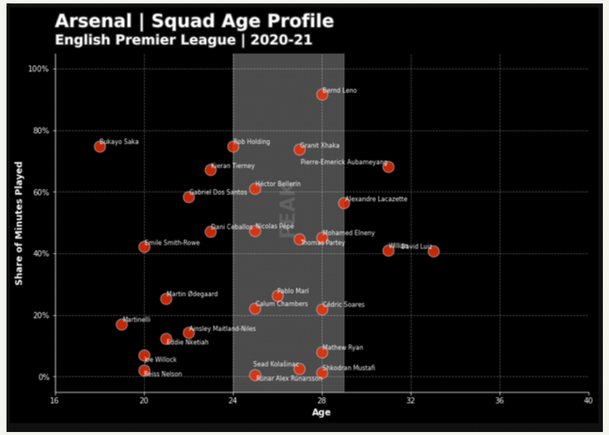
```python
model = SVC(random_state = 912, kernel='rbf')
model.fit(X_train, y_train)
predicted = model.predict(X_test)
report = classification_report(y_test, predicted)
print(report)
```

```
              precision    recall  f1-score   support

           H       0.62      0.48      0.54       112
          NH       0.64      0.76      0.70       138

    accuracy                           0.64       250
   macro avg       0.63      0.62      0.62       250
weighted avg       0.63      0.64      0.63       250
```

AGE PROFILE


Arsenal | Squad Age Profile
English Premier League | 2020-21
Share of Minutes Played
100%
80%
60%
40%
20%
0%
16
20
24
28
32
36
40
Age
Bernd Leno
Bukayo Saka
Rob Holding
Granit Xhaka
Pierre-Emerick Aubameyang
Kieran Tierney
Héctor Bellerín
Gabriel Dos Santos
Alexandre Lacazette
Dani Ceballos
Nicolas Pépé
Mohamed Elneny
Emile Smith-Rowe
Thomas Partey
Martin Ødegaard
Pablo Marí
Calum Chambers
Cédric Soares
Martinelli
Ainsley Maitland-Niles
Eddie Nketiah
Matthew Ryan
Ben Willock
Sead Kolasinac
Shkodran Mustafi
Miss Nelson
Runar Alex Rúnarsson

CONCLUSION

THE RESEARCH QUESTIONS WE WANTED TO ANSWER WERE:

- WHICH MATCH DAY DATA POINTS WERE MOST IMPORTANT IN PREDICTING A MATCH-WINNER?
- WHAT IS THE AGE PROFILE OF A PREMIER LEAGUE CHAMPION?

THE EXPLORATORY DATA ANALYSIS SHOWED THAT DIFFPTS, DIFFFORM PTS, HTGD, HTP WERE THE MOST VALUABLE DATA POINTS IN PREDICTING A MATCH-WINNER.

MANCHESTER CITY THE CHAMPIONS OF THE 2020-2021 SEASON HAD SMALLER SQUAD, AND THE PLAYERS WHO PLAYED THE MOST MINUTES WERE IN THE AGE RANGE OF 24-29. THEY DIDN'T HAVE MUCH YOUTH.

CONTACT ME

ANDREW LUJAN

DREWLUJAN33@GMAIL.COM

HTTPS://WWW.LINKEDIN.COM/IN/ANDREW-LUJAN-81B95B182/

HTTPS://GITHUB.COM/DREWSKY33