Demand of Football in European Leagues

DATA 450 Capstone

Matthew Wilcox

4/17/23

```
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn import metrics
from sklearn.linear_model import LinearRegression
```

```
total_data = pd.read_pickle('data/final_datasets/data_standardized.pkl')
print(total_data.head(10))
```

```
          home_team                away_team  home_score  away_score  \
0  tottenham_hotspur          manchester_city           0           0
1  tottenham_hotspur           wigan_athletic           0           1
2  tottenham_hotspur  wolverhampton_wanderers           3           1
3  tottenham_hotspur              everton_fc           1           1
4  tottenham_hotspur           sunderland_afc           1           1
5  tottenham_hotspur         blackburn_rovers           4           2
6  tottenham_hotspur             liverpool_fc           2           1
7  tottenham_hotspur               chelsea_fc           1           1
8  tottenham_hotspur         newcastle_united           2           0
9  tottenham_hotspur                fulham_fc           1           0

        date   time day_of_week  raw_attendance       stadium    city  ...  \
```

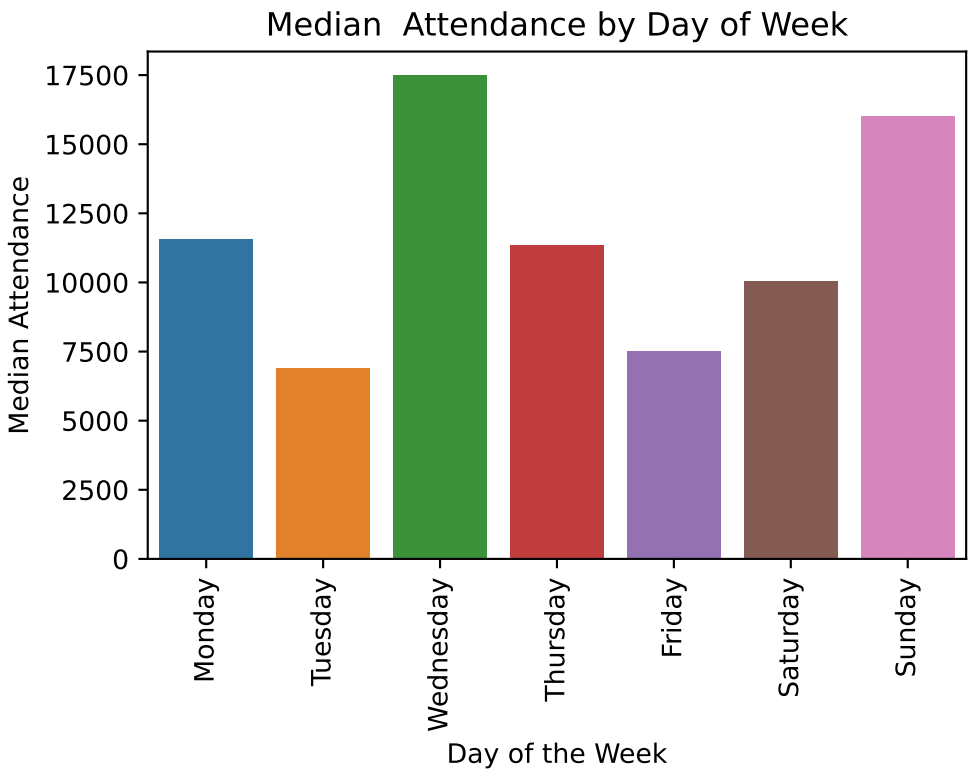
```
0  2010-08-14  12:45    Saturday           35928  Stadium News  London  ...
1  2010-08-28  15:00    Saturday           35101  Stadium News  London  ...
2  2010-09-18  15:00    Saturday           35940  Stadium News  London  ...
3  2010-10-23  12:45    Saturday           35967  Stadium News  London  ...
4  2010-09-11  20:00     Tuesday           35843  Stadium News  London  ...
5  2010-11-13  15:00    Saturday           35700  Stadium News  London  ...
6  2010-11-28  16:00      Sunday           35962  Stadium News  London  ...
7  2010-12-12  16:00      Sunday           35787  Stadium News  London  ...
8  2010-12-28  15:00     Tuesday           35927  Stadium News  London  ...
9  2011-01-01  15:00    Saturday           35603  Stadium News  London  ...

  BbMx>2.5  BbAv>2.5 BbMx<2.5 BbAv<2.5  capacity_filled           date_time  \
0     2.03      1.91     1.95     1.84         0.990189 2010-08-14 12:45:00
1     1.55      1.50     2.63     2.48         0.967396 2010-08-28 15:00:00
2     1.85      1.75     2.11     2.02         0.990519 2010-09-18 15:00:00
3     2.07      1.99     1.87     1.79         0.991263 2010-10-23 12:45:00
4     1.90      1.80     2.06     1.97         0.987846 2010-09-11 20:00:00
5     1.91      1.83     2.02     1.94         0.983905 2010-11-13 15:00:00
6     2.00      1.88     2.03     1.87         0.991126 2010-11-28 16:00:00
7     2.04      1.95     1.91     1.83         0.986303 2010-12-12 16:00:00
8     1.77      1.67     2.28     2.16         0.990161 2010-12-28 15:00:00
9     1.93      1.81     2.05     1.99         0.981231 2011-01-01 15:00:00

  season   mean_attend  std_attend standard_attend
0   2011  35892.894737  269.766338        0.130132
1   2011  35892.894737  269.766338       -2.935484
2   2011  35892.894737  269.766338        0.174615
3   2011  35892.894737  269.766338        0.274702
4   2011  35892.894737  269.766338       -0.184955
5   2011  35892.894737  269.766338       -0.715044
6   2011  35892.894737  269.766338        0.256167
7   2011  35892.894737  269.766338       -0.392542
8   2011  35892.894737  269.766338        0.126425
9   2011  35892.894737  269.766338       -1.074614

[10 rows x 42 columns]
```

```
time_df = total_data[[
    'date', 'time', 'day_of_week', 'date_time', 'raw_attendance', 'capacity_filled', 'stan
]]
```

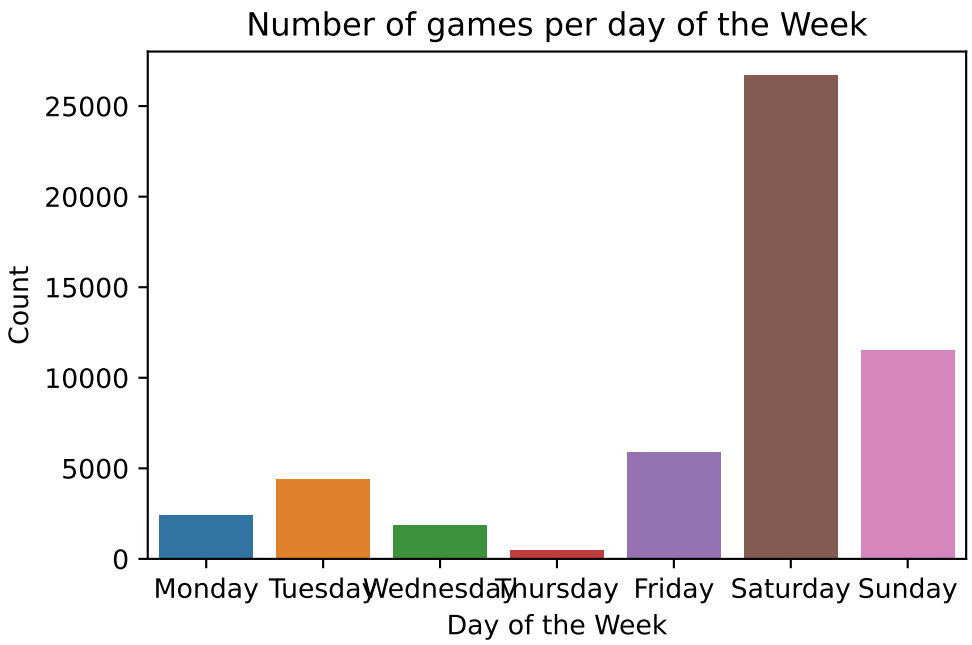
```
df_grouped_mean = time_df.groupby('day_of_week')['raw_attendance', 'capacity_filled', 'sta
df_grouped_median = time_df.groupby('day_of_week')['raw_attendance', 'capacity_filled', 's

day_categories = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sun
df_grouped_median['day_of_week'] = pd.Categorical(df_grouped_median['day_of_week'], catego
df_grouped_median.sort_values(by = 'day_of_week', inplace = True)
```

```
C:\Users\matth\AppData\Local\Temp\ipykernel_17192\26337685.py:6: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use

C:\Users\matth\AppData\Local\Temp\ipykernel_17192\26337685.py:7: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use
```

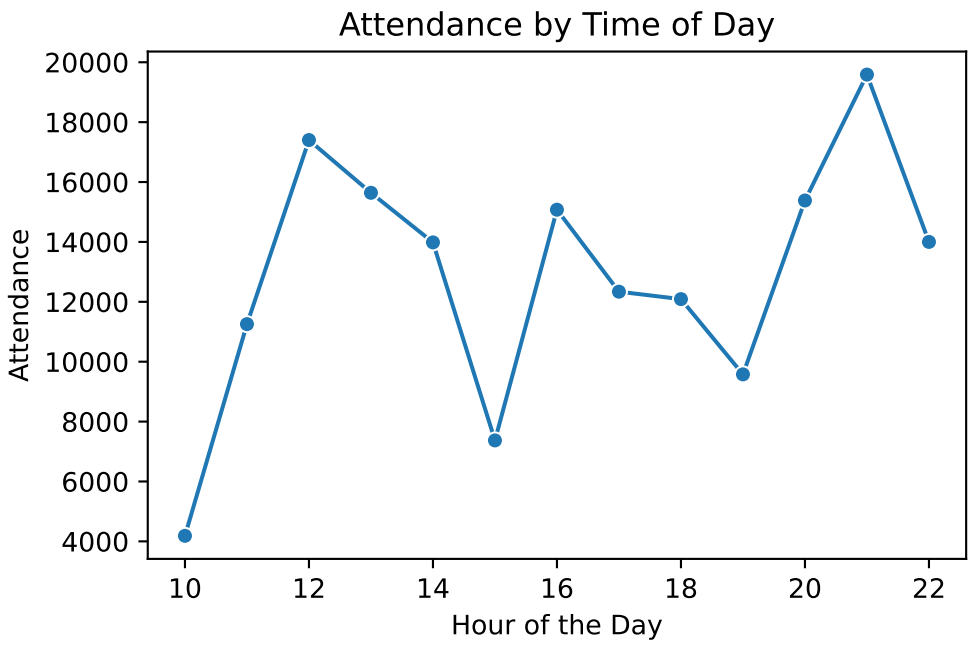
```
sns.barplot(data=df_grouped_median, x = 'day_of_week', y = 'raw_attendance').set(title ='M
plt.xticks(rotation=90)
plt.xlabel('Day of the Week')
plt.ylabel('Median Attendance')
plt.show()
```



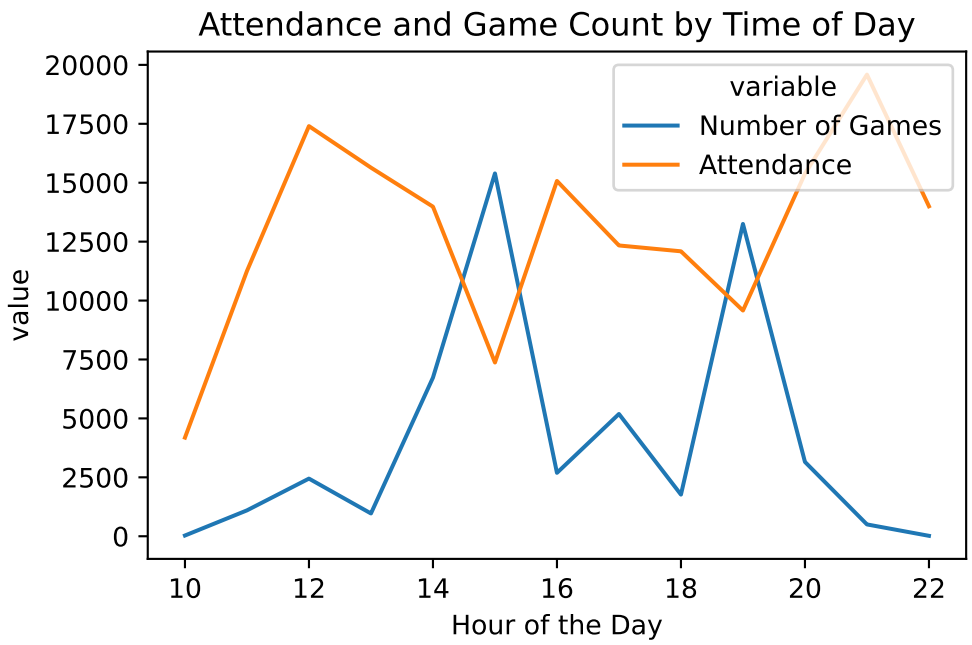

```
grouped_week_count = time_df.groupby('day_of_week').count().reset_index()

grouped_week_count['day_of_week'] = pd.Categorical(grouped_week_count['day_of_week'], cate
grouped_week_count.sort_values(by = 'day_of_week', inplace = True)


sns.barplot(data = grouped_week_count, x = 'day_of_week', y = 'date')
plt.xlabel('Day of the Week')
plt.ylabel('Count')
plt.title('Number of games per day of the Week')
plt.show()
```




```
df_grouped_mean_tod= time_df.groupby(time_df['date_time'].dt.hour).mean()
df_grouped_median_tod= time_df.groupby(time_df['date_time'].dt.hour).median()


sns.lineplot(data = df_grouped_median_tod, x = 'date_time', y = 'raw_attendance', markers
plt.title('Attendance by Time of Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Attendance')
plt.show()
```




```
df_grouped_count = time_df.groupby(time_df['date_time'].dt.hour).count()
# print(df_grouped_count)
df_grouped_count = df_grouped_count['raw_attendance'].reset_index()
df_grouped_count['count'] = df_grouped_count['raw_attendance']
df_grouped_count = df_grouped_count[['date_time', 'count']]
# df_grouped_count= df_grouped_count.rename(columns = {'date':'count'})
# print(df_grouped_count)

df_count_atted = pd.merge(df_grouped_count, df_grouped_median_tod, on = 'date_time')
df_count_atted = df_count_atted.drop(columns= ['capacity_filled'])
df_count_atted.rename( columns = {'raw_attendance': 'Attendance', 'count': "Number of Game
# print(df_count_atted)
melted_count_attend = pd.melt(df_count_atted, value_vars=['Number of Games', 'Attendance']
# print(melted_count_attend)

sns.lineplot(data = melted_count_attend, x = 'date_time', y = 'value', hue = 'variable')
plt.title('Attendance and Game Count by Time of Day')
plt.xlabel('Hour of the Day')

plt.show()
```


Attendance and Game Count by Time of Day
variable
Number of Games
Attendance
value
20000
17500
15000
12500
10000
7500
5000
2500
0
10
12
14
16
18
20
22
Hour of the Day