

```
In [2]: from config import password
import pandas as pd
import psycopg2
from sqlalchemy import create_engine
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker

In [3]: psycopg2.connect(f"dbname=postgres user=postgres host=localhost password={password} port=5432")

Out[3]: <connection object at 0x000001DDF6BCD370; dsn: 'user=postgres password=xxx dbname=postgres host=localhost port=5432', closed: 0>

In [4]: engine=create_engine(f"postgresql+psycopg2://postgres:{password}@localhost:5432/postgres")
conn=engine.connect()
conn

Out[4]: <sqlalchemy.engine.base.Connection at 0x1ddf38567f0>
```

ETL Project

In this project, I analyzed the Premier League Soccer players' 2020-2021 salary and goals per season to determine which offensive players ('forwards') were overpaid.

Note, each soccer team is made up of a combination of forwards, midfielders, defenders, and keepers. This study only focuses on forwards, whose main task is scoring goals. Additionally, this study does not consider factors such as total playing time, assists, dribbles, and defensive contributions, among other metrics that would provide a more complete assessment of a player's value and overall contribution to the team.

Extraction

Two tables were extracted from the Premier League webpage for the 2020-2021 soccer season to obtain players' goals stats and players' salaries.

I saved the data as csv files in resources as 'PLGoalsWeb' and 'PLSalaryWeb.'

Transformation

I noticed that there were differences in players' names spelling within the two tables. Given the small sample size, I manually normalized the names across both files. Additionally, I removed the currency symbols from the 'PLSalaryWeb' file for ease of use; and added a column called 'position' to the same. I also deleted data related to penalties from the 'PLGoalsWeb' file's 'Goals' column. Lastly, I selected and saved needed and formatted columns as "PLGoals.csv" and "PLSalary.csv."

Load

In PostgreSQL, I created a database ("postgres") and two tables, "plgoals" and "pksalary." I imported the formatted csv files into the tables/database.

Analysis

For analysis and data visualization purposes, I utilized Pandas to import SQL queries from PostgreSQL PGAdmin.

I. Position of the League's Highest Paid Player

Prior to determining the cost efficiency of forwards, which is the focus of this project, I was interested in finding out the position of the highest paid player in the league. Such an analysis would confirm the assumption that a forward is usually the highest paid player in the league by virtue of his main task which is to score goals.

Firstly, I joined the 'plgoals' and 'pksalary' tables. The merged tables would reflect the columns needed to define the main variable ('costpergoal').

```
In [5]: costpergoal=pd.read_sql('SELECT plgoals.player, goals, avgvalue, position FROM plgoals  
print(costpergoal.head(10))
```

	player	goals	avgvalue	position
0	Harry Kane	23	10400000	Forward
1	Mohamed Salah	22	10400000	Forward
2	Bruno Fernandes	18	8022857	Midfielder
3	Heung-Min Son	17	7280000	Forward
4	Javier Chicharito Hernandez	16	7280000	Forward
5	Jamie Vardy	15	7280000	Forward
6	Alexandre Lacazette	13	9466673	Forward
7	Ilkay Gundogan	13	7280000	Midfielder
8	Marcus Rashford	11	10400000	Forward
9	Gareth Bale	11	31200000	Forward

Secondly, I filtered the data to obtain top highest paid player per position, as displayed below.

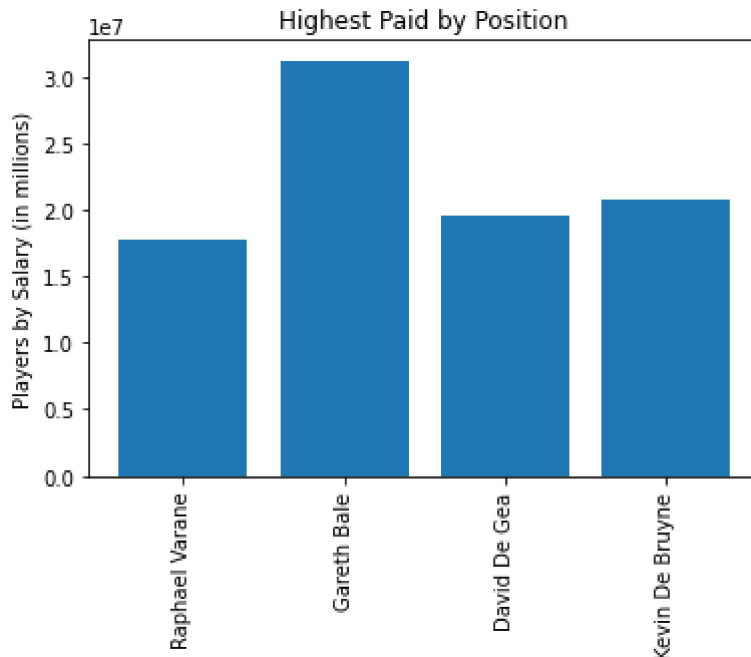
```
In [6]: highestpaid=costpergoal.loc[costpergoal.groupby('position')['avgvalue'].nlargest(1).ind  
highestpaid
```

```
Out[6]:
```

	player	goals	avgvalue	position
80	Raphael Varane	0	17680000	Defender
9	Gareth Bale	11	31200000	Forward
88	David De Gea	0	19500000	Keeper
28	Kevin De Bruyne	6	20800000	Midfielder

Lastly, I created a bar chart that compares the highest paid players per position, including forwards, midfielders, defenders, and keepers. The bar chart reflected the filtered result and confirmed the assumption by illustrating that Gareth Bale, a forward, is indeed the league's highest paid player.

```
In [7]: x=highestpaid['player']
y=highestpaid['avgvalue']
plt.bar(x,y)
plt.title('Highest Paid by Position')
plt.ylabel('Players by Salary (in millions)')
plt.xticks(rotation=90)
plt.show();
```



II. Top Ten Least Cost-Efficient Forwards

To determine the top ten least cost-efficient offensive players in the league, I joined the 'plgoals' and 'pksalary' tables. The merged table included the following columns: player, goals, avgvalue, position. I created an additional column ('costpergoal') by dividing the player's annual salary ('avgvalue') by the total numbers of goals scored during the season ('goals'). Thereafter, I filtered the table to only obtain results relevant to forwards ('position'). To ensure the top ten least cost-efficient forwards were displayed first, I ordered the data by position (i.e., to display forwards only) and by total number of goals not equal to 0. Any division by 0 is undefined and would result in an error in PostgreSQL.

```
In [8]: costpergoal=pd.read_sql("SELECT plgoals.player, goals, avgvalue, position, (avgvalue/go
print(costpergoal.head(10))
```

	player	goals	avgvalue	position	costpergoal
0	Romelu Lukaku	1	16900000	Forward	16900000
1	Steven Bergwijn	1	3800000	Forward	3800000
2	Darren Bent	2	6826000	Forward	3413000
3	Darren Bent	2	6826000	Forward	3413000
4	Anthony Martial	4	13000000	Forward	3250000
5	Callum Hudson-Odoi	2	6240000	Forward	3120000
6	Gareth Bale	11	31200000	Forward	2836363
7	Timo Werner	6	14144000	Forward	2357333
8	Gabriel Martinelli	2	4680000	Forward	2340000
9	Christian Pulisic	4	7150000	Forward	1787500

As displayed above, the initially displayed top ten players appeared on duplicate rows in some

cases. Additionally, some well-known and historically strongly performing forwards appeared to have a small number of total goals. Both cases were results of players being traded from or to another league during the season. To account for such discrepancies, only the first instance of each player was kept, and other duplicate rows were dropped. The first instance was sufficient for the analysis because salary data remained constant across duplicate instances. Likewise, the query was filtered to only apply to forwards who had scored more than one goal during the season.

```
In [9]: costpergoal=costpergoal.drop(costpergoal.index[[0,1]])
```

```
In [10]: costpergoal=costpergoal.drop_duplicates(subset='player', keep='first').head(10)
costpergoal
```

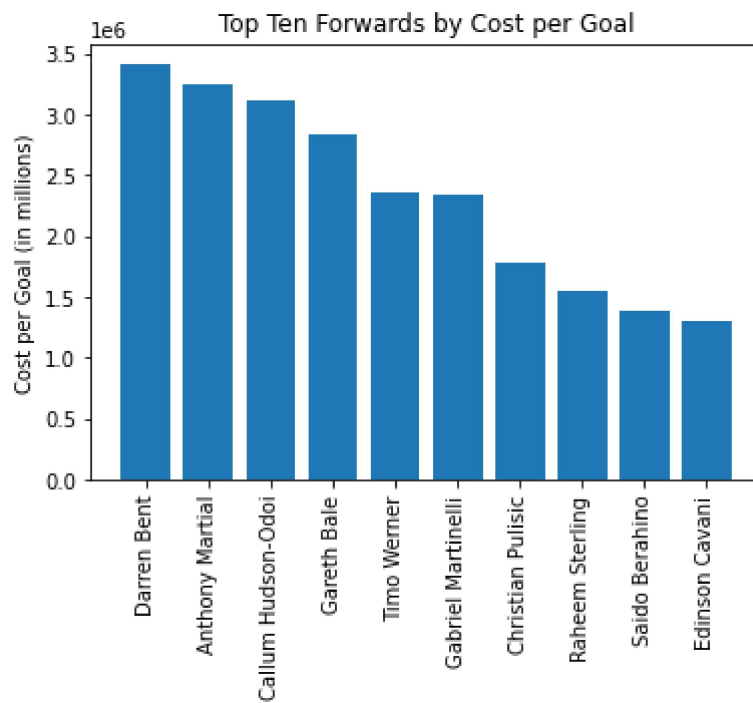
```
Out[10]:
```

	player	goals	avgvalue	position	costpergoal
2	Darren Bent	2	6826000	Forward	3413000
4	Anthony Martial	4	13000000	Forward	3250000
5	Callum Hudson-Odoi	2	6240000	Forward	3120000
6	Gareth Bale	11	31200000	Forward	2836363
7	Timo Werner	6	14144000	Forward	2357333
8	Gabriel Martinelli	2	4680000	Forward	2340000
9	Christian Pulisic	4	7150000	Forward	1787500
10	Raheem Sterling	10	15600000	Forward	1560000
11	Saido Berahino	3	4160000	Forward	1386666
12	Edinson Cavani	10	13000000	Forward	1300000

Per the chart below, the top ten overpaid offensive players ('forwards') in the 2020-2021 Premier League season were: Darren Bent, Anthony Martial, Callum Hudson-Odoi, Gareth Bale, Timo Werner, Gabriel Martinelli, Christian Pulisic, Raheem Sterling, Saido Berahino, and Edinson Cavani. These ten players are overpaid because their performance (i.e., goal scoring stats) does not justify the annual salary they receive.

Furthermore, when compared to the chart outlining the highest paid player per position, the chart displaying the top ten forwards by cost per goal provides an interesting finding. Gareth Bale who is the league's most expensive player is not the least cost-efficient forward. Although his performance does not justify his annual salary, he still performs better than three well-known and expensive forwards.

```
In [11]: x=costpergoal['player']
y=costpergoal['costpergoal']
plt.bar(x,y)
plt.title('Top Ten Forwards by Cost per Goal')
plt.ylabel('Cost per Goal (in millions)')
plt.xticks(rotation=90)
plt.show();
```



Conclusion

This project primarily sought to determine which offensive players ('forwards') were overpaid in the Premier League 2020-2021 season. Accordingly, the project provided results of the top ten least cost-efficient forward in the league.