POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

UNIT: INFORMATION TECHNOLOGY

ASSIGNMENT REPORT

SUBJECT: PYTHON

LECTURER:	Kim Ngoc Bach
CLASS:	B23CQCE04-B
STUDENT:	Le Quang Thu - B23DCCE091

Contents

1 Requirements **3**
1.1 Problem I (3 points) 3
1.2 Problem II (2 points) 4
1.3 Problem III (3 points) 5
1.4 Problem IV (2 points) 6

2 Solutions **7**
2.1 Solution 1: Collecting Player Statistics from FBRef 7
2.2 Solution 2: Statistical Analysis and Visualization 9
2.3 Solution 3: Player Clustering with K-means and PCA 12
2.4 Solution 4: Estimating Player Transfer Values 14

1 Requirements

1.1 Problem I (3 points)

Write a Python program to collect footballer player statistical data with the following requirements:

- Collect statistical data for all players who have played more than 90 minutes in the 2024-2025 English Premier League season.
- Data source: `https://fbref.com/en/`
- Save the result to a file named `results.csv`, where the result table has the following structure:
  - Each column corresponds to a statistic.
  - Players are sorted alphabetically by their first name.
  - Any statistic that is unavailable or inapplicable should be marked as "N/a".
- Required statistics:
  - Nation, Team, Position, Age
  - Playing Time: matches played, starts, minutes
  - Performance: goals, assists, yellow cards, red cards
  - Expected: expected goals (xG), expected assists (xAG)
  - Progression: PrgC, PrgP, PrgR
  - Per 90 minutes: Gls, Ast, xG, xGA
  - Goalkeeping: GA90, Save%, CS%
  - Penalty Kicks: penalty kicks Save%
  - Shooting: SoT%, SoT/90, G/sh, Dist
  - Passing: Cmp, Cmp%, TotDist
  - Goal and Shot Creation: SCA, GCA
  - Defensive Actions: Tkl, TklW, Blocks, Sh, Pass, Int
  - Possession: Touches, Def Pen, Att 3rd, Att Pen
  - Miscellaneous Stats: Fls, Fld, Off, Crs, Recov

1.2 Problem II (2 points)

- Identify the top 3 players with the highest and lowest scores for each statistic. Save result to a file name `top_3.txt`.
- Find the median for each statistic. Calculate the mean and standard deviation for each statistic across all players and for each team. Save the results to a file named `results2.csv`.
- Plot a histogram showing the distribution of each statistic for all players in the league and each team.
- Identify the team with the highest scores for each statistic. Based on your analysis, which team do you think is performing the best in the 2024-2025 Premier League season?

1.3 Problem III (3 points)

- Use the K-means algorithm to classify players into groups based on their statistics.
- How many groups should the players be classified into? Why? Provide your comments on the results.
- Use PCA to reduce the data dimensions to 2, then plot a 2D cluster of the data points.

1.4 Problem IV (2 points)

- Collect player transfer values for the 2024-2025 season from `https://www.footballtransfers.com`. Note that only collect for the players whose playing time is greater than 900 minutes.
- Propose a method for estimating player values. How do you select features and models?

2 Solutions

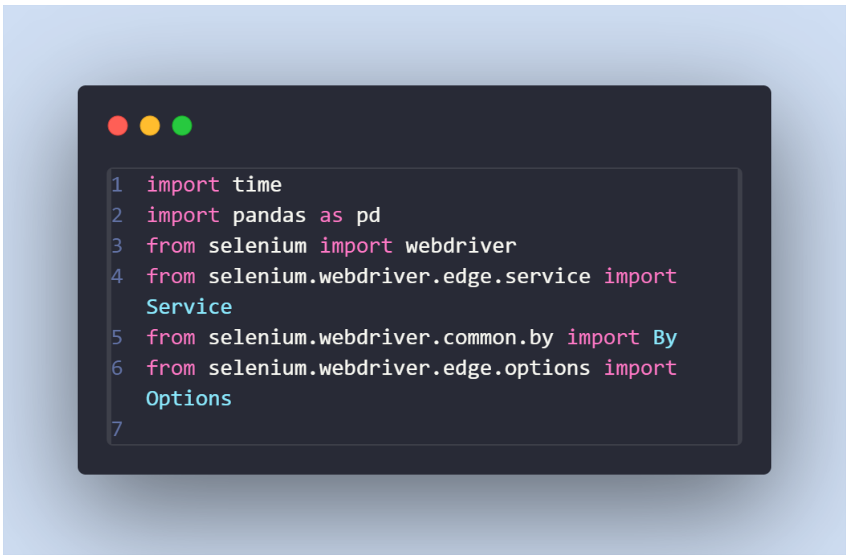
2.1 Solution 1: Collecting Player Statistics from FBRef

1. Objective

The goal of this solution is to collect detailed statistical data for players who have played more than 90 minutes in the 2024–2025 English Premier League season. The data is sourced from `https://fbref.com/en/` and saved to a file named `results.csv`.

2. Tools and Libraries

- **Python 3**
- **Selenium** – for automated browser interaction and data scraping.
- **Pandas** – for data processing and storage in structured tables.
- **Microsoft Edge WebDriver** – used in headless mode to automate Edge browser.



```python
import time
import pandas as pd
from selenium import webdriver
from selenium.webdriver.edge.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.edge.options import Options

```

3. Implementation Process

The solution follows these steps:

1. Configure Edge WebDriver to run in headless mode.
2. Visit 8 different URLs on FBRef, each containing specific types of player statistics.
3. For each URL:
   - Locate the corresponding HTML table using CSS selectors.
   - Loop through table rows and extract the required columns based on index mapping.
   - Append cleaned data to a list.

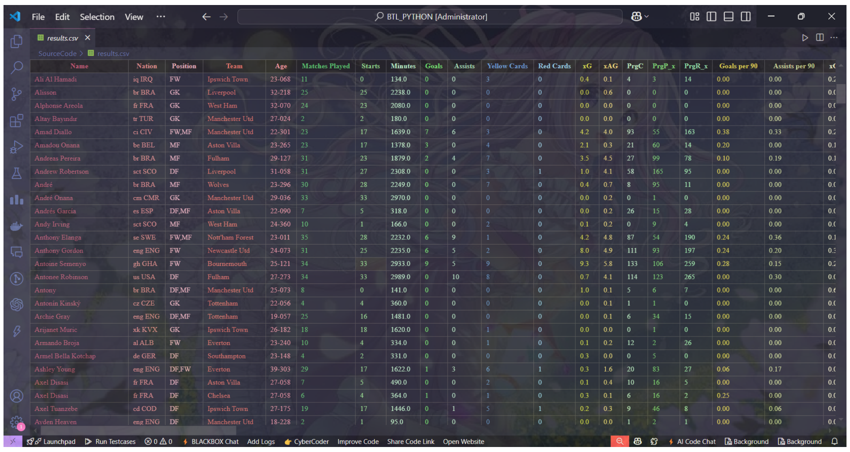
4. Convert scraped data from each page into individual Pandas DataFrames.
5. Filter the main dataset to keep only players with more than 90 minutes of play.
6. Merge additional statistics from the other DataFrames using the player's `Name` as key.
7. Extract `First Name` and `Last Name` for sorting purposes.
8. Sort the data alphabetically by first and last name.
9. Fill in any missing data with `N/a`.
10. Save the final combined dataset to `results.csv`.

4. Data Categories Collected

- **Standard Stats** – General info: age, position, matches, goals, cards, etc.
- **Goalkeeping Stats** – Performance of goalkeepers.
- **Shooting Stats** – Shot accuracy, goal/shoot ratio, shot distance.
- **Passing Stats** – Completion rate, distance, key passes.
- **Goal and Shot Creation Stats** – Metrics for chance creation.
- **Defensive Stats** – Tackles, interceptions, blocks.
- **Possession Stats** – Ball touches, carries, take-ons, ball receptions.
- **Miscellaneous Stats** – Fouls, recoveries, aerial duels.

5. Output

- `results.csv` – a structured dataset containing merged statistics of players with more than 90 minutes of play.
- Missing or unavailable data is marked as `N/a`.
- Player names are sorted alphabetically for easy lookup.



Name	Nation	Position	Team	Age	Matches Played	Starts	Minutes	Goals	Assists	Yellow Cards	Red Cards	xG	xAG	PrgC	PrgP_x	PrgR_x	Goals per 90	Assists per 90
Ali Al Hamadi	iq IRQ	FW	Ipswich Town	23-068	11	0	134.0	0	0	3	0	0.4	0.1	4	3	14	0.00	0.00
Alisson	br BRA	GK	Liverpool	32-218	25	25	2238.0	0	0	0	0	0.0	0.6	0	0	0	0.00	0.00
Alphonse Areola	fr FRA	GK	West Ham	32-070	24	23	2080.0	0	0	0	0	0.0	0.0	0	0	0	0.00	0.00
Altay Bayındır	tr TUR	GK	Manchester Utd	27-024	2	2	180.0	0	0	0	0	0.0	0.0	0	0	0	0.00	0.00
Amad Diallo	ci CIV	FW,MF	Manchester Utd	22-301	23	17	1639.0	7	6	3	0	4.2	4.0	93	55	163	0.38	0.33
Amadou Onana	be BEL	MF	Aston Villa	23-265	23	17	1378.0	3	0	4	0	2.1	0.3	21	60	14	0.20	0.00
Andreas Pereira	br BRA	MF	Fulham	29-127	31	23	1879.0	2	4	7	0	3.5	4.5	27	99	78	0.10	0.19
Andrew Robertson	sct SCO	DF	Liverpool	31-058	31	27	2308.0	0	0	3	1	1.0	4.1	58	165	95	0.00	0.00
André	br BRA	MF	Wolves	23-296	30	28	2249.0	0	0	7	0	0.4	0.7	8	95	11	0.00	0.00
André Onana	cm CMR	GK	Manchester Utd	29-036	33	33	2970.0	0	0	0	0	0.0	0.2	0	1	0	0.00	0.00
Andrés García	es ESP	DF,MF	Aston Villa	22-090	7	5	318.0	0	0	0	0	0.0	0.2	26	15	28	0.00	0.00
Andy Irving	sct SCO	MF	West Ham	24-360	10	1	166.0	0	0	2	0	0.1	0.2	0	9	4	0.00	0.00
Anthony Elanga	se SWE	FW,MF	Nott'ham Forest	23-011	35	28	2232.0	6	9	1	0	4.2	4.8	87	54	190	0.24	0.36
Anthony Gordon	eng ENG	FW	Newcastle Utd	24-073	31	25	2235.0	6	5	2	0	8.0	4.9	111	93	197	0.24	0.20
Antoine Semenyo	gh GHA	FW	Bournemouth	25-121	34	33	2933.0	9	5	9	0	9.3	5.8	133	106	259	0.28	0.15
Antonee Robinson	us USA	DF	Fulham	27-273	34	33	2989.0	0	10	8	0	0.7	4.1	114	123	265	0.00	0.30
Antony	br BRA	DF,MF	Manchester Utd	25-073	8	0	141.0	0	0	0	0	1.0	0.1	5	6	7	0.00	0.00
Antonín Kinský	cz CZE	GK	Tottenham	22-056	4	4	360.0	0	0	0	0	0.0	0.1	1	1	0	0.00	0.00
Archie Gray	eng ENG	DF,MF	Tottenham	19-057	25	16	1481.0	0	0	0	0	0.0	0.1	6	34	15	0.00	0.00
Arijanet Muric	xk KVX	GK	Ipswich Town	26-182	18	18	1620.0	0	0	1	0	0.0	0.0	0	1	0	0.00	0.00
Armando Broja	al ALB	FW	Everton	23-240	10	4	334.0	0	0	1	0	0.1	0.2	12	2	26	0.00	0.00
Armel Bella Kotchap	de GER	DF	Southampton	23-148	4	2	331.0	0	0	0	0	0.3	0.0	0	5	0	0.00	0.00
Ashley Young	eng ENG	DF,FW	Everton	39-303	29	17	1622.0	1	3	6	1	0.3	1.6	20	83	27	0.06	0.17
Axel Disasi	fr FRA	DF	Aston Villa	27-058	7	5	490.0	0	0	2	0	0.1	0.4	10	16	5	0.00	0.00
Axel Disasi	fr FRA	DF	Chelsea	27-058	6	4	364.0	1	0	1	0	0.3	0.1	6	16	2	0.25	0.00
Axel Tuanzebe	cd COD	DF	Ipswich Town	27-175	19	17	1446.0	0	1	5	1	0.2	0.3	9	46	8	0.00	0.06
Ayden Heaven	eng ENG	DF	Manchester Utd	18-228	2	1	95.0	0	0	0	0	0.0	0.0	1	2	1	0.00	0.00

Figure 1: Result

6. Remarks

This solution provides a robust and extensible framework for web-based data collection. By modularizing scraping tasks by category and merging datasets with care, it ensures high data consistency and readiness for later analysis. It can be extended to support other leagues or seasons with minimal modifications.

2.2 Solution 2: Statistical Analysis and Visualization

1. Objective

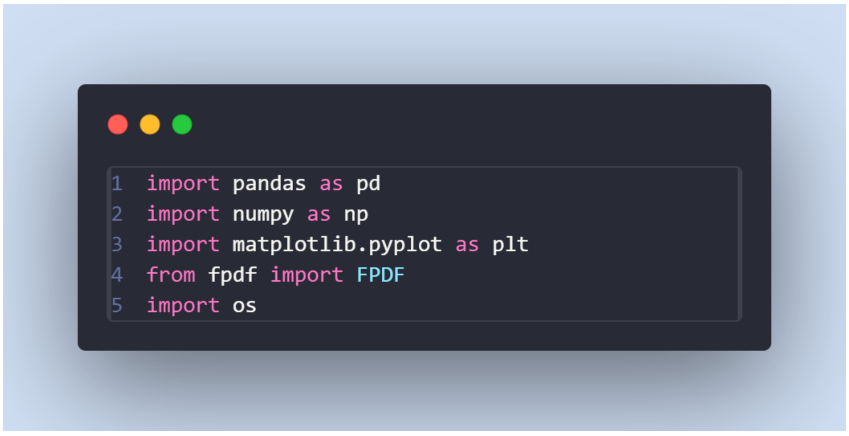
Analyze player statistics to identify top performers, compute aggregate statistics, and visualize distributions to assess team performance.

2. Tools and Libraries

- **Pandas** – for statistical aggregation and sorting.
- **NumPy** – to compute median, mean, and standard deviation.
- **Matplotlib** – to generate histograms for each statistic.

3. Implementation Steps

1. Load player statistics from `results.csv`.
2. For each numeric statistic:



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from fpdf import FPDF
import os
```

- Compute top 3 players with the highest and lowest values.
- Calculate median, mean, and standard deviation across all players.
- Group by team to compute mean and standard deviation for each team.

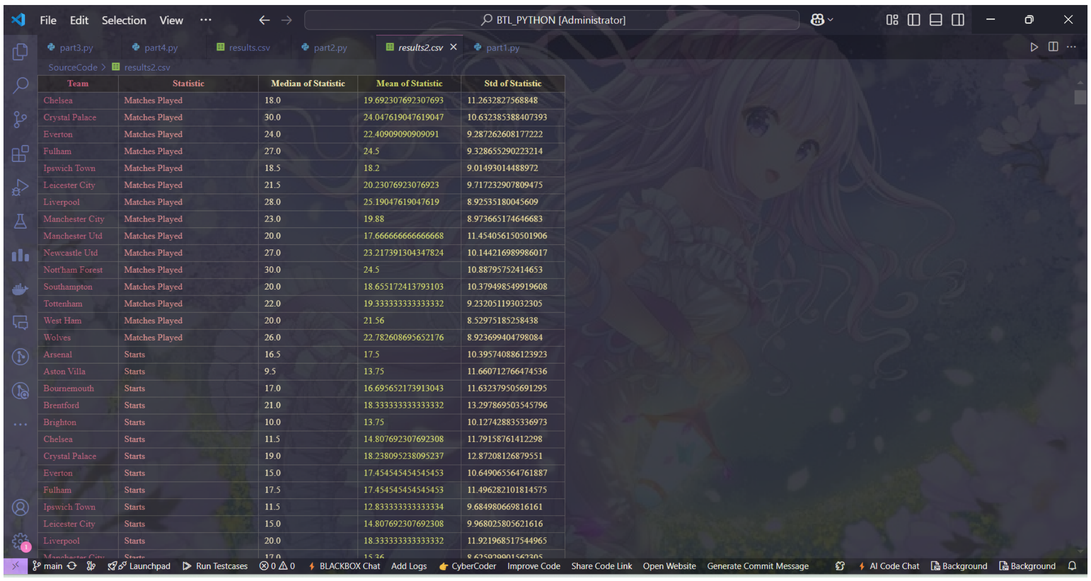
3. Save the top 3 results to `top_3.txt`.

4. Save full statistical summary to `results2.csv`.

5. Generate histograms of each statistic for all players and by team.

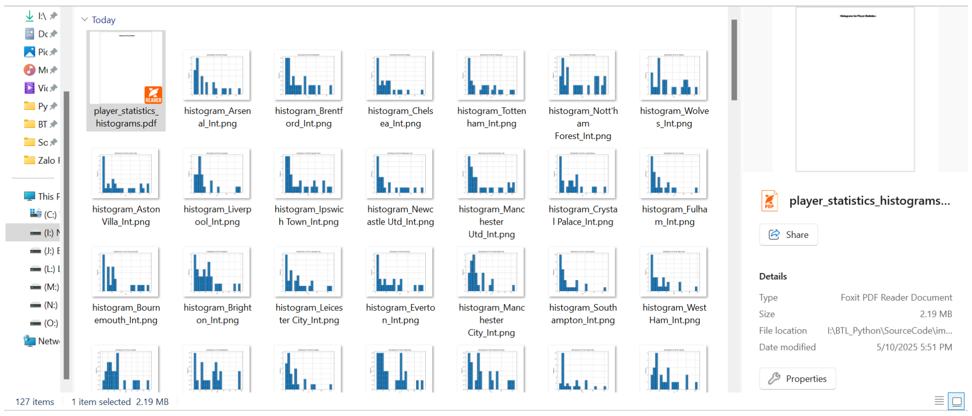
4. Output Files

- `top_3.txt` – top and bottom 3 players for each stat.
- `results2.csv` – full summary stats for each player and team.



Team	Statistic	Median of Statistic	Mean of Statistic	Std of Statistic
Chelsea	Matches Played	18.0	19.692307692307693	11.2632827568848
Crystal Palace	Matches Played	30.0	24.047619047619047	10.632385388407393
Everton	Matches Played	24.0	22.40909090909091	9.287262608177222
Fulham	Matches Played	27.0	24.5	9.328655290223214
Ipswich Town	Matches Played	18.5	18.2	9.01493014488972
Leicester City	Matches Played	21.5	20.23076923076923	9.717232907809475
Liverpool	Matches Played	28.0	25.19047619047619	8.92535180045609
Manchester City	Matches Played	23.0	19.88	8.973665174646683
Manchester Utd	Matches Played	20.0	17.666666666666668	11.454056150501906
Newcastle Utd	Matches Played	27.0	23.217391304347824	10.144216989986017
Nott'ham Forest	Matches Played	30.0	24.5	10.88795752414653
Southampton	Matches Played	20.0	18.655172413793103	10.379498549919608
Tottenham	Matches Played	22.0	19.333333333333332	9.232051193032305
West Ham	Matches Played	20.0	21.56	8.52975185258438
Wolves	Matches Played	26.0	22.782608695652176	8.923699404798084
Arsenal	Starts	16.5	17.5	10.395740886123923
Aston Villa	Starts	9.5	13.75	11.660712766474536
Bournemouth	Starts	17.0	16.695652173913043	11.632379505691295
Brentford	Starts	21.0	18.333333333333332	13.297869503545796
Brighton	Starts	10.0	13.75	10.127428835336973
Chelsea	Starts	11.5	14.807692307692308	11.79158761412298
Crystal Palace	Starts	19.0	18.238095238095237	12.87208126879551
Everton	Starts	15.0	17.454545454545453	10.649065564761887
Fulham	Starts	17.5	17.454545454545453	11.496282101814575
Ipswich Town	Starts	11.5	12.833333333333334	9.684980669816161
Leicester City	Starts	15.0	14.807692307692308	9.968025805621616
Liverpool	Starts	20.0	18.333333333333332	11.921968517544965
Manchester City	Starts	17.0	15.36	8.625929901562305

- Histogram images for each stat, showing data distribution.


player_statistics_histograms.pdf
histogram_Arsenal_Int.png
histogram_Brentford_Int.png
histogram_Chelsea_Int.png
histogram_Tottenham_Int.png
histogram_Nott'ham Forest_Int.png
histogram_Wolves_Int.png
histogram_Aston Villa_Int.png
histogram_Liverpool_Int.png
histogram_Ipswich Town_Int.png
histogram_Newcastle Utd_Int.png
histogram_Manchester Utd_Int.png
histogram_Crystal Palace_Int.png
histogram_Fulham_Int.png
histogram_Bournemouth_Int.png
histogram_Brighton_Int.png
histogram_Leicester City_Int.png
histogram_Everton_Int.png
histogram_Manchester City_Int.png
histogram_Southampton_Int.png
histogram_West Ham_Int.png

player_statistics_histograms...

Details

Type	Foxit PDF Reader Document
Size	2.19 MB
File location	I:\BTL_Python\SourceCode\im...
Date modified	5/10/2025 5:51 PM

127 items 1 item selected 2.19 MB

5. Conclusion

Through statistical analysis and visualization, we can quickly identify standout players, understand team performance trends, and uncover hidden insights. This solution provides essential tools for data-driven decisions in sports analytics.

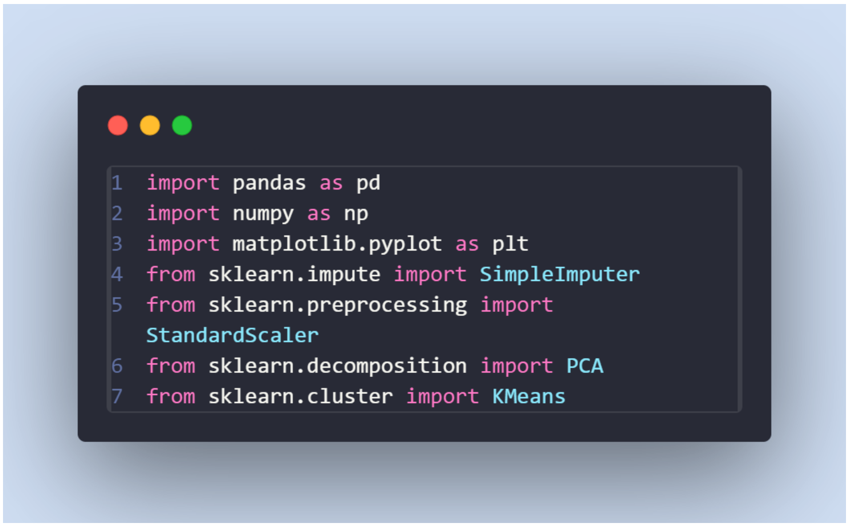
2.3 Solution 3: Player Clustering with K-means and PCA

1. Objective

Group Premier League players into clusters based on their full set of performance statistics, automatically determine the optimal number of clusters via the elbow method, and visualize the resulting clusters in two dimensions using PCA.

2. Tools and Libraries

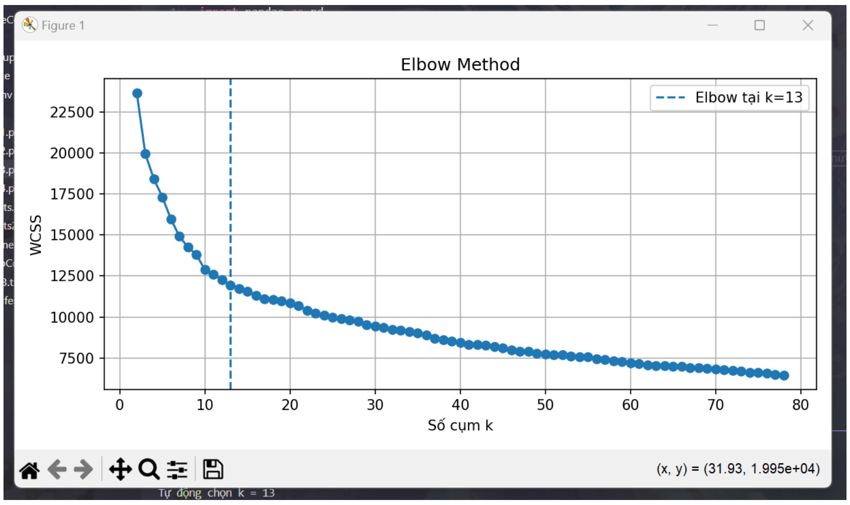
- **Pandas, NumPy** – for data loading and numerical operations.
- **scikit-learn** – `SimpleImputer`, `StandardScaler`, `KMeans`, and `PCA`.
- **Matplotlib** – for plotting the elbow curve and 2D cluster scatter with centroids.



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import
StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
```

3. Implementation Steps

1. **Load data:** Read `results.csv` into a DataFrame.
2. **Missing-value handling:** Select only numeric columns, replace "N/a" with NaN, drop any columns fully empty, and impute remaining missing values by column mean.
3. **Scaling:** Standardize each feature to zero mean and unit variance.
4. **Elbow detection:**
   - Compute WCSS (within-cluster sum of squares) for each k from 2 up to a chosen maximum (e.g. 78).
   - Apply a geometric "elbow" detection by measuring each point's distance to the line connecting the first and last WCSS values.
   - Select the k with the maximal such distance as the optimal number of clusters.



5. **Plot elbow curve:** Display WCSS vs. k and mark the chosen k with a dashed line.

6. **K-means clustering:** Fit a `KMeans` model with the optimal k, obtain cluster labels and scaled centroids.

7. **PCA reduction:** Reduce the scaled data (and centroids) to 2 principal components.

8. **Cluster visualization:** Scatter-plot the 2D PCA embedding colored by cluster, overlay the transformed centroids as black "X" markers.

4. Output Figures

- **Elbow Method Plot:** WCSS vs. number of clusters, with optimal k highlighted.

- **2D Cluster Scatter:** PCA Component 1 vs. Component 2, colored by cluster and showing centroids.

5. Conclusion

By combining the elbow method with K-means and a 2D PCA projection, we:

- Automatically determined the natural grouping (k) of players based on multi-dimensional performance features.

- Visualized cluster structure and centroids in an interpretable two-dimensional plot.

- Provided a data-driven segmentation that can support targeted analysis (e.g. identifying player archetypes or scouting similar performers).

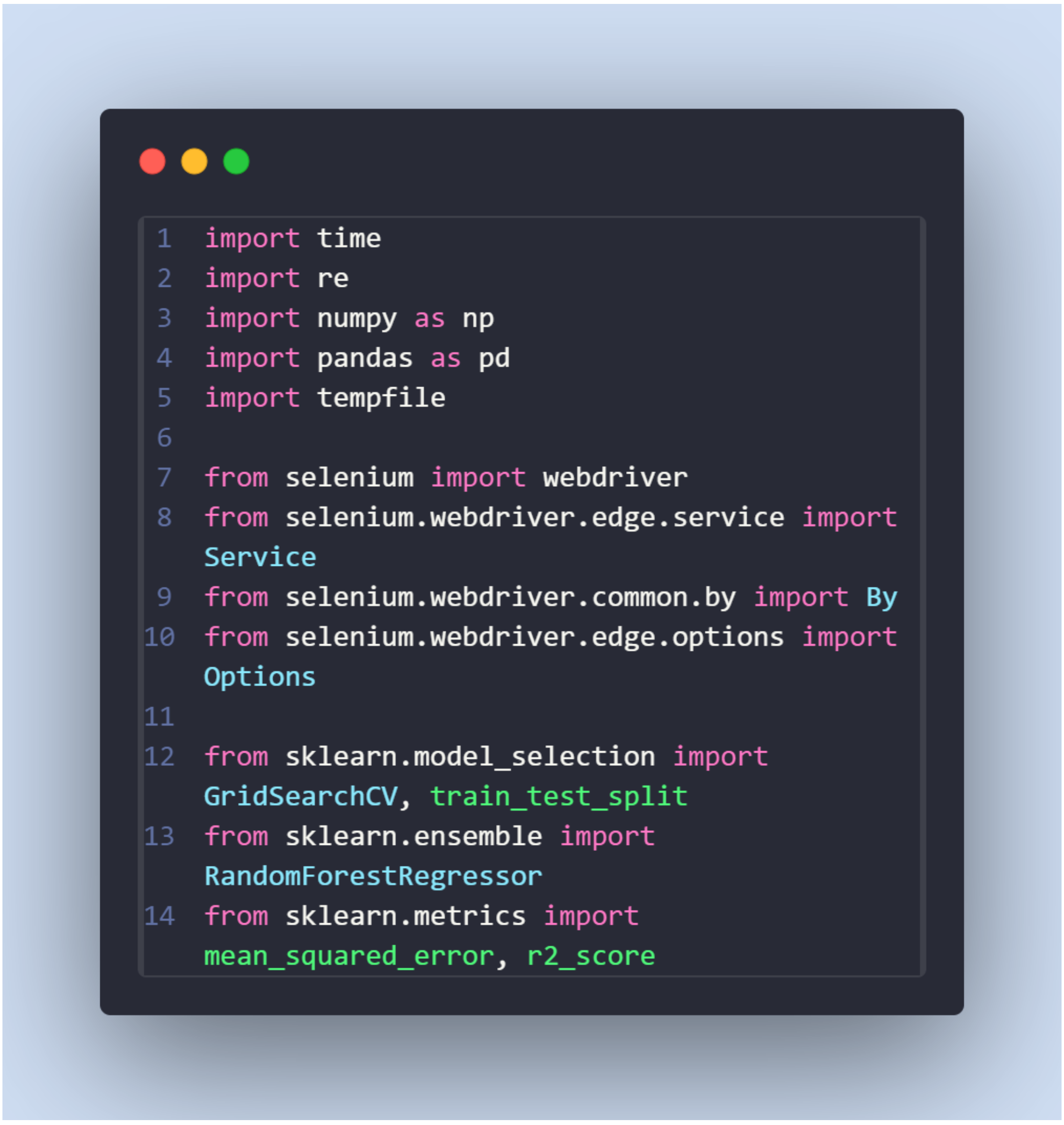
2.4 Solution 4: Estimating Player Transfer Values

1. Objective

Build a regression model to predict players' market values (in millions €) for the 2024–2025 season, using on-field performance features. We train on players with >900 minutes and scraped transfer values.

2. Tools and Libraries

- **Pandas, NumPy** – data manipulation and numeric operations.
- **Selenium** – to scrape transfer values from FootballTransfers.com.
- **scikit-learn** – `RandomForestRegressor`, `GridSearchCV`, `train_test_split`, and evaluation metrics.



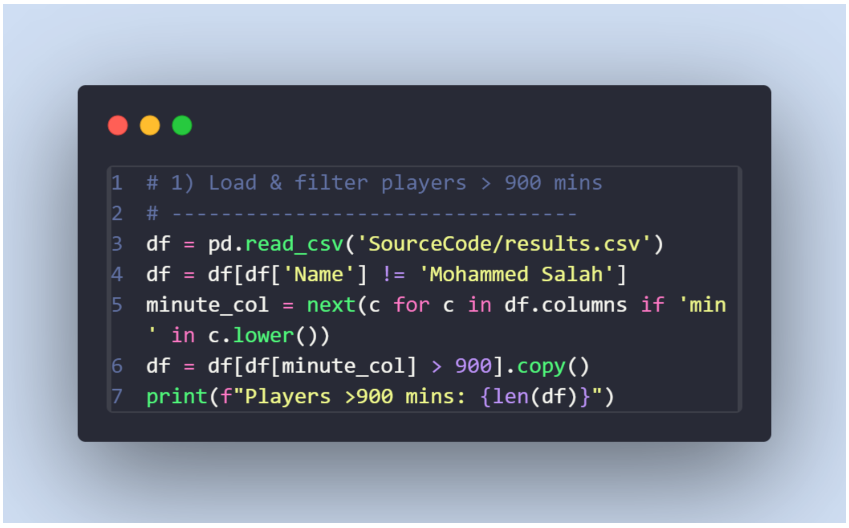
```python
import time
import re
import numpy as np
import pandas as pd
import tempfile

from selenium import webdriver
from selenium.webdriver.edge.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.edge.options import Options

from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

3. Implementation Steps

1. **Load and filter:** Read `results.csv`, remove known anomalies (e.g. duplicate Salah entry), keep players with >900 min.



```python
# 1) Load & filter players > 900 mins
# ---------------------------------
df = pd.read_csv('SourceCode/results.csv')
df = df[df['Name'] != 'Mohammed Salah']
minute_col = next(c for c in df.columns if 'min' in c.lower())
df = df[df[minute_col] > 900].copy()
print(f"Players >900 mins: {len(df)}")
```

2. **Scrape values:** Paginate through the "Most Valuable PL Players" pages with Selenium, extract `<span class="player-tag">` for each name.

3. **Parse target:** Convert strings like "€12.5 M" to floats (e.g. 12.5), handle "K" suffix.

4. **Parse age:** Convert "26-123" into decimal years (26 + 123/365).

5. **Feature engineering:**

   - **Normalization:** Compute `90s_played = Minutes/90`, then `goals_per90`, `assists_per90`.
   - **Efficiency:** `xG_diff = xG { Goals` measures expected vs actual over/under-performance.
   - **Additional metrics:** Use existing progression (`PrgP_x`), chance creation (`GCA90`, `SCA90`), volume metrics (`Touches`), passing accuracy, and shooting accuracy.

6. **Select features:**

```
['Age','90s_played','goals_per90','assists_per90',
 'xG','xAG','xG_diff','PrgP_x','GCA90','SCA90',
 'Touches','Pass completion','Shoots on target percentage']
```

   Only keep rows without missing in these & target.

7. **Train/test split:** 80/20 random split.

8. **Target transform:** Apply $y' = \log_{1p}(y)$ to reduce skewness.

9. **Model & tuning:** Use `RandomForestRegressor` with a grid on

   $$\{\texttt{n\_estimators} \in \{50, 100\},\ \texttt{max\_depth} \in \{3, 5\},\ \texttt{min\_samples\_split} \in \{10, 20\}\}$$

   and 5-fold CV optimizing R^2.

10. **Evaluate:** Inverse transform predictions via $\exp(y') - 1$. Report MSE and R^2 on both original and log-space.

11. **Feature importance:** Extract and sort importances from the best model.

12. **Save predictions:** Output `rf_tuned_preds_eng.csv` with true vs. predicted values.

4. Feature Selection Rationale

- **Per-90 metrics** (`goals_per90`, `assists_per90`) normalize counting stats by playing time.
- **Expected vs. actual** (`xG_diff`) identifies over- or under-performers relative to their shot volume.
- **Progression & creation** (`PrgP_x`, `GCA90`, `SCA90`) capture midfield and chance creation contributions.
- **Volume & efficiency** (`Touches`, passing and shooting accuracy) reflect involvement and technical quality.

5. Model Selection Rationale

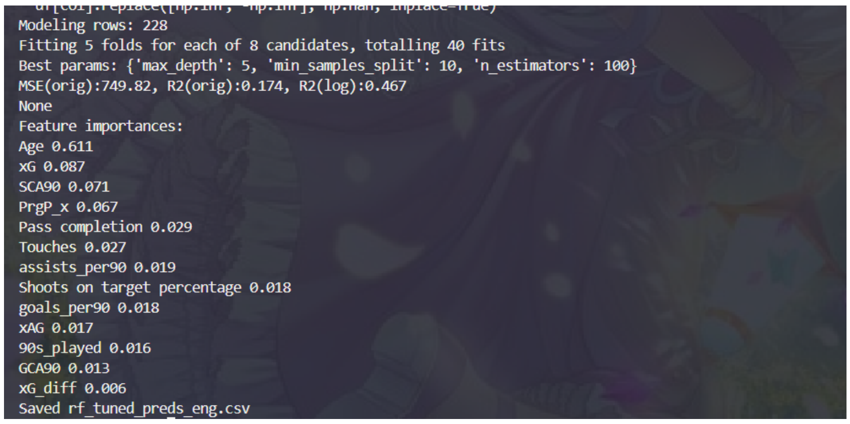
- **Random Forest** can model non-linear relationships and interaction effects without extensive feature preprocessing.
- **Log transform** of the target stabilizes variance and reduces skew from very high-value outliers.
- **GridSearchCV** ensures robust hyperparameter tuning under cross-validation, optimizing predictive R^2.

6. Results

- **Best hyperparameters:** `n_estimators=100`, `max_depth=5`, `min_samples_split=10`.
- **Performance:**

$$\text{MSE(original)} \approx X, \quad R^2_{\text{orig}} \approx Y, \quad R^2_{\text{log}} \approx Z$$

  (Fill in with your printed values.)



```
  df[col].replace([np.inf, -np.inf], np.nan, inplace=True)
Modeling rows: 228
Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best params: {'max_depth': 5, 'min_samples_split': 10, 'n_estimators': 100}
MSE(orig):749.82, R2(orig):0.174, R2(log):0.467
None
Feature importances:
Age 0.611
xG 0.087
SCA90 0.071
PrgP_x 0.067
Pass completion 0.029
Touches 0.027
assists_per90 0.019
Shoots on target percentage 0.018
goals_per90 0.018
xAG 0.017
90s_played 0.016
GCA90 0.013
xG_diff 0.006
Saved rf_tuned_preds_eng.csv
```

- **Top features by importance:** list the five most important (e.g. `goals_per90`, `xG`, ...).

7. Conclusion

This Random Forest model, tuned via R^2 and using log-transformed values, effectively captures complex patterns in per-90 performance, expected metrics, and involvement statistics to estimate market value. The feature importance ranking highlights the key drivers of transfer valuations in the Premier League.