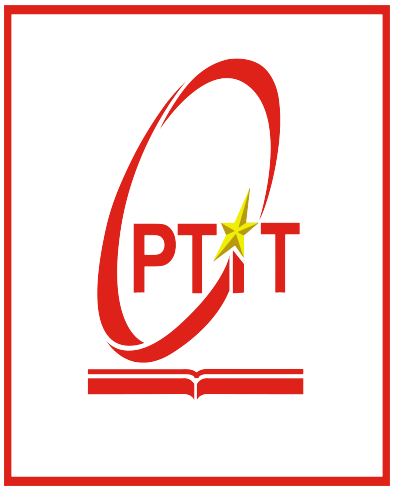
POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY I

PYTHON PROGRAMMING COURSE



ASSIGNMENT REPORT

Lecturer	: KIM NGOC BACH
Student	: DAO VAN KHANG
Student ID	: B23DCVT218
Class	: D23CQCE04-B
Group	: 04

Ha Noi – 2025

Summary

This report documents the implementation of Python Programming Assignment 1 for the 2024–2025 academic year. The objective of the assignment is to apply practical Python programming skills to the domain of football analytics, using real-world data from the English Premier League.

The task is divided into four parts. In the first part, a Python program is developed to collect statistical data for all players who have played more than 90 minutes during the 2024–2025 season. The data is sourced from the website fbref.com and includes a wide range of metrics related to performance, goalkeeping, passing, shooting, possession, and more. The collected data is saved to a file named `results.csv`, with players sorted alphabetically by first name and unavailable statistics marked as "N/a".

The second part involves statistical analysis of the collected dataset. It includes identifying the top three and bottom three players for each statistic, and computing the median, mean, and standard deviation values per attribute across all players and per team. The results are saved to `top_3.txt` and `results2.csv`. Additionally, histograms are plotted to visualize the distribution of each statistic, and a comparative analysis is conducted to determine the best-performing team.

The third part applies machine learning techniques to classify players based on their statistical profiles. The K-means clustering algorithm is used, and Principal Component Analysis (PCA) is employed to reduce the data to two dimensions for visualization.

In the final part, transfer values for players with more than 900 minutes of play are collected from FootballTransfers.com. A method for estimating player value is proposed, an explanation of feature selection and model choice.

Contents

List of Figures

Chapter 1: Data Collection

1.1 Objective

The goal of this section was to write a Python script to collect statistical data for all football players in the English Premier League (EPL) 2024–2025 season who have played more than 90 minutes from FBref.com. The dataset must be structured, cleaned, and exported into a CSV file named `results.csv`.

1.2 Tools and Libraries Used

`requests`: for sending HTTP requests to the data source.

`BeautifulSoup`: for parsing and scraping HTML content.

`pandas`: to manipulate tabular data, merge multiple datasets, and export the final DataFrame to a CSV file.

`time`: to delay requests between URL calls to prevent being rate-limited.

1.3 Methodology

1.3.1 URL Management

All URLs were stored in a dictionary called `url_dict`, each pointing to a different stat category.

1.3.2 Table Extraction

A function `get_frame_from_url()` was implemented to:

- Download HTML content from each stats page.
- Clean HTML comments that often wrap table content on FBref.
- Parse the `<table>` with `class min_width sortable stats_table...`
- Extract column headers with group prefixes for clarity.
- Convert each row to a structured format and store it in a pd.DataFrame.

1.3.3 Data Integration

All dataframes were merged using the shared keys: `Player, Nation, Pos, and Squad.`

Duplicate entries were removed using these keys.

1.3.4 Filtering and Cleaning

Players who played fewer than 90 minutes were removed `(stats_Playing Time_Min > 90).`

Nationalities were cleaned to be in uppercase abbreviation form `("England" → "ENG").`

Final data was sorted alphabetically by player name.

1.3.5 Output

Relevant columns were selected using a mapping dictionary `(column_mapping).`

Empty or missing values were replaced with `"N/a".`

The final structured DataFrame was saved to `result.csv.`

1.4 Result

The final dataset was saved to `result.csv` and contains full statistics for all EPL players with more than 90 minutes played. The data is clean, structured, and ready for subsequent analysis (Parts II–IV of the assignment).

	Player	Nation	Team	Position	Age	Matches Played	Starts	Minutes	Goals	Assists	Yellow Cards	Red Cards	Expected Goals	Expedted Assist Goals	PrgC	PrgP	PrgR	Gls	Ast	xG	xAG	GA90	Save%	CS%	Penalty
2	Aaron Cresswell	ENG	West Ham	DF	35-140	14	7	589	0	0	2	0	0.1	1.1	4	24	2	0	0	0	0.2	N/a	N/a	N/a	N/a
3	Aaron Ramsdale	ENG	Southampton	GK	26-355	27	27	2430	0	0	2	0	0	0	0	0	0	0	0	0	0	2.3	67.1	7.4	28
4	Aaron Wan-Bissaka	ENG	West Ham	DF	27-159	32	31	2794	2	2	1	0	1.2	2.9	98	125	139	0	0	0	0.1	N/a	N/a	N/a	N/a
5	Abdoulaye Doucouré	MLI	Everton	MF	32-123	30	29	2425	3	1	5	1	3.9	2.3	40	78	91	0	0	0	0.1	N/a	N/a	N/a	N/a
6	Abdukodir Khusanov	UZB	Manchester City	DF	21-064	6	6	503	0	0	1	0	0	0.1	1	25	2	0	0	0	0	N/a	N/a	N/a	N/a
7	Abdul Fatawu Issahaku	GHA	Leicester City	FW	21-057	11	6	579	0	2	0	0	0.4	1.6	42	17	60	0	0	0	0.2	N/a	N/a	N/a	N/a
8	Adam Armstrong	ENG	Southampton	FW,MF	28-083	20	15	1248	2	2	4	0	3.3	1.2	25	21	79	0	0	0	0.1	N/a	N/a	N/a	N/a
9	Adam Lallana	ENG	Southampton	MF	36-359	14	5	361	0	2	4	0	0.2	0.9	6	24	10	0	1	0	0.2	N/a	N/a	N/a	N/a
10	Adam Smith	ENG	Bournemouth	DF	34-005	22	17	1409	0	0	6	0	0.7	0.3	12	40	31	0	0	0	0	N/a	N/a	N/a	N/a
11	Adam Webster	ENG	Brighton	DF	30-120	11	8	617	0	0	0	0	0	0.5	7	40	2	0	0	0	0.1	N/a	N/a	N/a	N/a
12	Adam Wharton	ENG	Crystal Palace	MF	20-336	19	15	1258	0	2	2	0	0.3	3	14	105	10	0	0	0	0.2	N/a	N/a	N/a	N/a
13	Adama Traoré	ESP	Fulham	FW,MF	29-099	33	16	1592	2	6	3	0	3.9	4.7	89	61	145	0	0	0	0.3	N/a	N/a	N/a	N/a
14	Albert Grønbaek	DEN	Southampton	FW,MF	23-346	4	2	143	0	0	0	0	0.1	0	1	1	3	0	0	0	0	N/a	N/a	N/a	N/a
15	Alejandro Garnacho	ARG	Manchester Utd	MF,FW	20-307	33	22	2056	5	1	2	0	7	3.6	134	54	274	0	0	0	0.2	N/a	N/a	N/a	N/a
16	Alex Iwobi	NGA	Fulham	FW,MF	29-001	35	33	2796	9	6	1	0	4.6	6.9	135	199	224	0	0	0	0.2	N/a	N/a	N/a	N/a
17	Alex McCarthy	ENG	Southampton	GK	35-152	5	5	450	0	0	0	0	0	0	0	0	0	0	0	0	0	2.6	70.3	0	
18	Alex Palmer	ENG	Ipswich Town	GK	28-267	11	11	990	0	0	2	0	0	0	0	1	0	0	0	0	0	2.45	58.7	0	
19	Alex Scott	ENG	Bournemouth	MF	21-256	18	7	709	0	0	2	0	0.7	0.8	15	49	33	0	0	0	0.1	N/a	N/a	N/a	N/a
20	Alexander Isak	SWE	Newcastle Utd	FW	25-225	31	31	2487	22	6	1	0	19	4	77	77	196	1	0	1	0.2	N/a	N/a	N/a	N/a

Figure 1.1: Example of collected data

Chapter 2: Statistical Analysis & Visualization

2.1 Objective

This part of the assignment focuses on analyzing the football player data collected previously by:

- Identifying top-performing players.
- Summarizing statistical distributions across the league and by team.
- Visualizing data using histograms.
- Determining the best-performing team in the 2024–2025 Premier League season.

2.2 Tools and Libraries Used

`pandas`: for numerical and statistical computation.

`matplotlib`: for generating histograms and saving them into PDF.

`tabulate`: for formatting output text for top 3 rankings.

2.3 Methodology

2.3.1 Top 3 and Bottom 3 Players by Statistic

A function `identify_the_top_3_each_statistic()` was developed to:

- Identify and export top 3 and bottom 3 players for each statistic.
- Write results to `top_3.txt` using a formatted table for clarity.

```
Top 3 player with the highest Age:
+------------------+----------+----------+------------+-------+
| Player           | Nation   | Team     | Position   |   Age |
+==================+==========+==========+============+=======+
| Łukasz Fabiański | POL      | West Ham | GK         | 40.04 |
+------------------+----------+----------+------------+-------+
| Ashley Young     | ENG      | Everton  | DF,FW      | 39.82 |
+------------------+----------+----------+------------+-------+
| James Milner     | ENG      | Brighton | MF         | 39.33 |
+------------------+----------+----------+------------+-------+

Top 3 player with the lowest Age:
+---------------+----------+----------------+------------+-------+
| Player        | Nation   | Team           | Position   |   Age |
+===============+==========+================+============+=======+
| Mikey Moore   | ENG      | Tottenham      | FW,MF      | 17.73 |
+---------------+----------+----------------+------------+-------+
| Ethan Nwaneri | ENG      | Arsenal        | FW,MF      | 18.12 |
+---------------+----------+----------------+------------+-------+
| Harry Amass   | ENG      | Manchester Utd | DF         | 18.13 |
+---------------+----------+----------------+------------+-------+
```

Figure 2.1: Top 3 players example

2.3.2 Summary Statistics

In `find_median_mean_and_standard_each_statistic()`:

- **Median**, **Mean**, and **Standard Deviation** were calculated for:

  - All players in the league.
  - Each individual team.

- The final results were exported to `result2.csv`.

	Team	median of Age	mean of Age	std of Age	median of Matches Played	mean of Matches Played	std of Matches Played	median of Starts	mean of Starts	std of Starts	median of Minutes	mean of Minutes	std of Minutes	median o
2	All	26.57	26.9	4.23	23	21.04	9.79	15	15.43	10.81	1339	1383.81	914.27	
3	Arsenal	26.88	26.49	3.82	23.5	23.27	8.07	16.5	17.5	10.4	1478.5	1564.45	902.98	
4	Aston Villa	27.63	27.14	4.04	20.5	19.43	10.33	9.5	13.75	11.66	1004	1235.36	943.18	
5	Bournemouth	26	26.47	3.84	26	22.22	10.18	17	16.7	11.63	1670	1494.61	1004.39	
6	Brentford	24.84	26.2	3.91	27	22.86	11.15	21	17.81	12.96	1913	1592.33	1092.86	
7	Brighton	24.52	26.04	5.17	20	18.96	9.76	9	13.36	9.87	895.5	1198.46	866.12	
8	Chelsea	24.31	24.2	2.38	17.5	19.15	10.88	11.5	14.38	11.4	1046.5	1291.42	995.13	
9	Crystal Palace	27.19	27.17	3.14	29	23.38	10.21	18	17.71	12.47	1562	1585.9	1047.05	
10	Everton	26.53	27.97	5.03	24	22.41	9.29	15	17.45	10.65	1357.5	1565.23	904.27	
11	Fulham	28.82	28.82	3.35	27	24.5	9.33	17.5	17.45	11.5	1620.5	1568.27	963.24	0.
12	Ipswich Town	26.7	26.9	3.16	18.5	18.2	9.01	11.5	12.83	9.68	984.5	1146.77	798.99	
13	Leicester City	26.75	27.21	4.46	21.5	20.23	9.72	15	14.81	9.97	1408	1329.19	824.51	
14	Liverpool	26.44	27.21	3.69	28	24.48	8.9	19	17.81	11.99	1627	1596.38	981.76	
15	Manchester City	26.68	26.99	4.91	23	19.88	8.97	17	15.36	8.63	1494	1381.36	766.28	
16	Manchester Utd	25.73	25.91	5	20	18.78	10.97	14	13.78	10.7	1335	1231.67	920.11	
17	Newcastle Utd	27.46	27.96	4.72	27	22.57	9.92	13	16.26	12.26	1413	1459.83	1021.14	
18	Nott'ham Forest	27.14	27.27	3.59	29.5	23.86	10.58	18.5	17	12.9	1764	1527.68	1071.82	
19	Southampton	26.97	26.97	4.22	20	18.66	10.38	13	13.24	9.84	1178	1184.31	856.24	
20	Tottenham	25.64	25.66	4.54	21	18.89	9.28	15	13.81	8.12	1252	1241.11	696.54	
21	West Ham	28.33	28.61	5.01	20	20.92	8.28	14	14.96	10.52	1070	1341.92	885.12	
22	Wolves	26.85	27.67	3.99	26	22.78	8.92	15	16.65	10.91	1364	1495.22	875.62	

Figure 2.2: Summary statistics example

2.3.3 Histogram Visualization

Histograms were plotted for key statistics (`Goals`, `Assists`, `G/sh`, `Tackles`, `Challenges`, and `Blocks`) using `plot_histogram()`:

- Distribution histograms for all players in the league.
- Separate histograms for each team.
- All plots were compiled into a single PDF: histograms.pdf.

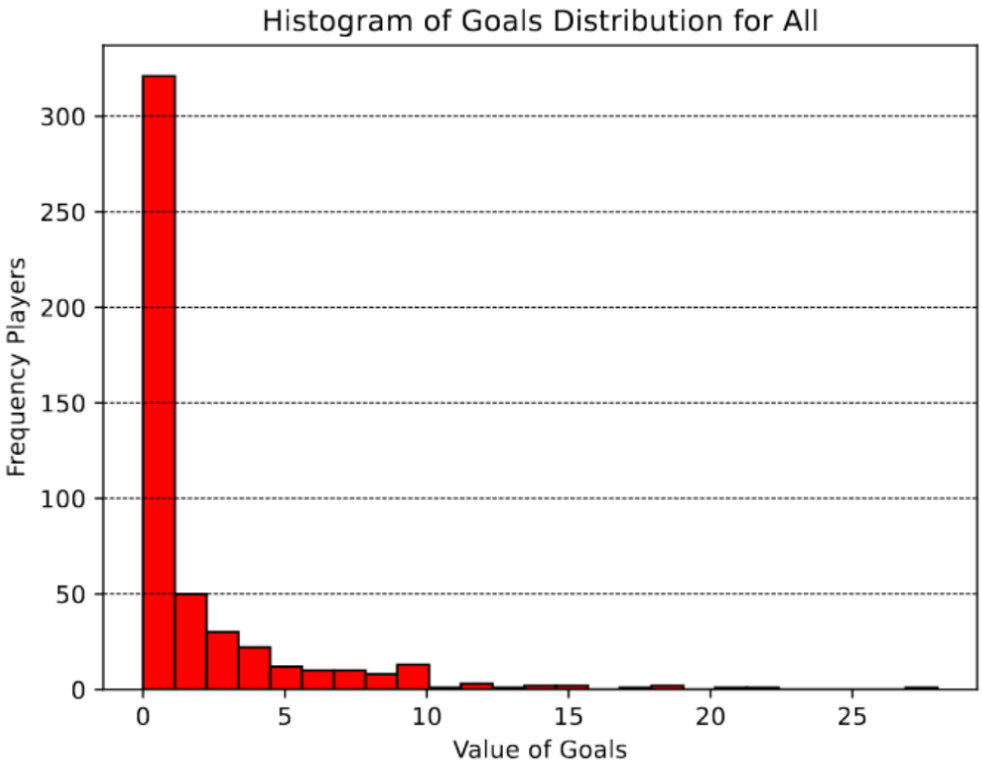


Figure 2.3: Histogram example 1

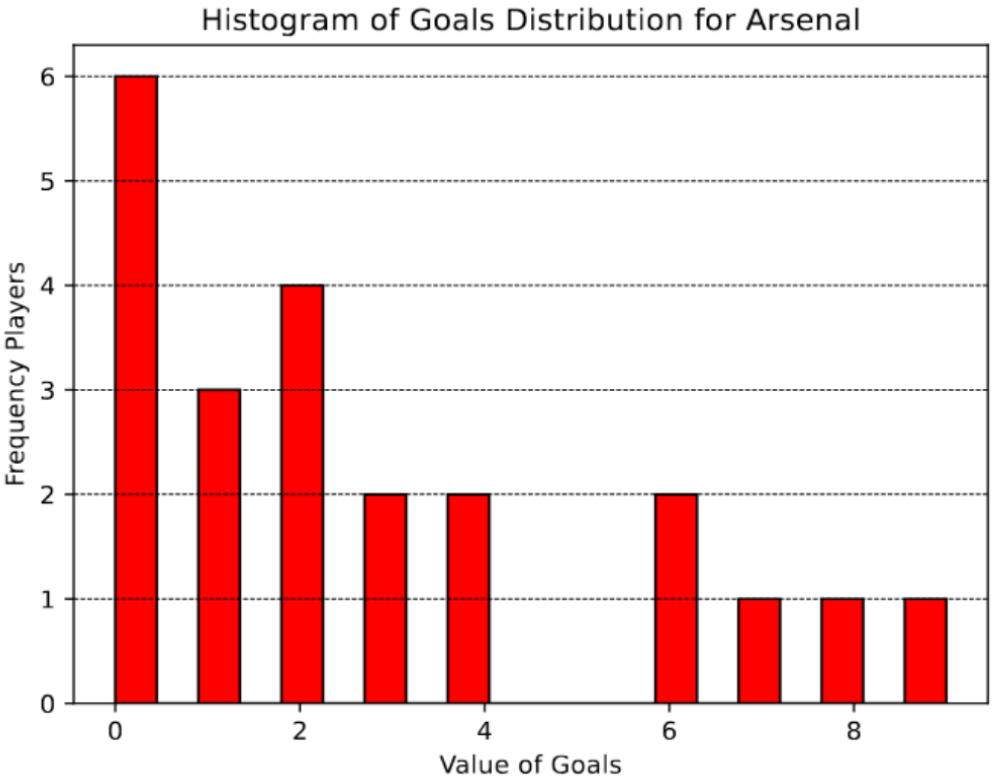


Figure 2.4: Histogram example 2

2.3.4 Team Performance Evaluation

The function `identify_team_with_highest_score()`:

- Loads team averages from `result2.csv`.
- Determines which team has the highest mean for each statistic.
- Counts how many times each team leads across different statistics.
- Declares the team with the most appearances as the **best-performing team** in the league.

Example output:

- Fulham has the highest Age score.
- Liverpool has the highest Matches Played score.
- Brentford has the highest Starts score.
- Liverpool has the highest Minutes score.

  => Liverpool is performing the best in the 2024-2025 Premier League season.

2.4 Result

The statistical analysis and visualizations yielded several meaningful insights regarding player and team performances during the 2024–2025 Premier League season:

- **Top and Bottom 3 Players**: For each key statistic, the top 3 and bottom 3 players were identified and exported to `top_3.txt`. This allowed for a quick overview of standout individual performances across the league.
- **Summary Statistics**: Median, mean, and standard deviation values were calculated for both the league-wide dataset and individual teams. These values were saved in `result2.csv` for further analysis or comparison.

- **Histograms**: A series of histograms were generated to visualize the distribution of important metrics such as `Goals`, `Assists`, `Shots per Goal`, `Tackles`, `Challenges`, and `Blocks`. All plots were compiled into a single PDF file: `histograms.pdf`.
- **Best Performing Team**: Based on average values across all statistics, Liverpool was identified as the best-performing team of the season. This conclusion was drawn by counting the number of statistics in which each team had the highest average.

Chapter 3: Clustering and Dimensionality Reduction

3.1 Objective

This section applies machine learning techniques to group Premier League players into distinct categories based on their statistical performance using:

- **K-means Clustering**.
- **PCA (Principal Component Analysis)** for dimensionality reduction and visualization.

3.2 Tools and Libraries Used

`pandas`: to load, clean, and preprocess player data, including scaling and handling missing values.

`matplotlib`: to visualize data distributions, Elbow and Silhouette methods, and PCA-clustered player groups.

`sklearn`: used for clustering and evaluation tasks including: KMeans, StandardScaler, silhouette_score, PCA

3.3 Methodology

3.3.1 Data Preprocessing

- The dataset from `result.csv` was loaded.
- Age was transformed from `"years-days"` format to float (`23-45 → 23.12`).
- All non-numeric and missing values `"N/a"` were converted to 0 to enable numerical processing.
- `StandardScaler` was used to normalize all numerical features before clustering.

3.3.2 Optimal K Determination

To choose the appropriate number of clusters (k) for K-means, two methods were employed:

Elbow Method

- Plotted number of clusters (k) vs. inertia (within-cluster sum of squares).
- A clear "elbow" in the curve indicates the optimal k.

Silhouette Method

- Calculated silhouette scores for each k to measure cohesion/separation.
- The higher the score, the better the clustering quality.

Result

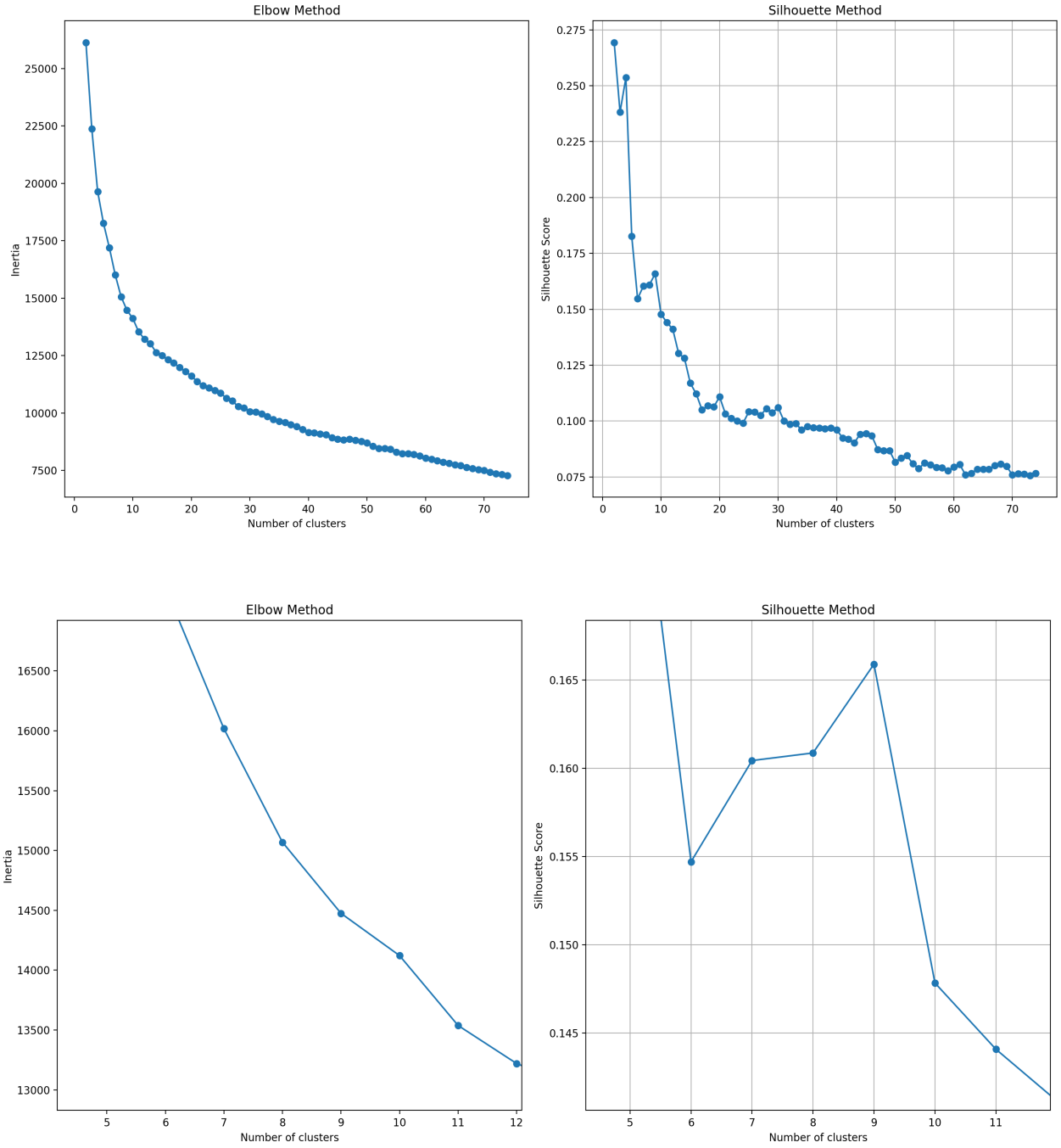


Figure 3.1: Elbow method and Silhouette method result

Based on both plots, the best value of k was estimated in the range **6 to 10**.

3.3.3 K-Means Clustering and Dimensionality Reduction with PCA

The function `run_kmeans_and_plot()`:

- For k = 6, 7, 8, 9, KMeans clustering was performed:
  - The KMeans algorithm from sklearn.cluster is fitted to the scaled player statistics.
  - Principal Component Analysis (PCA) is used to reduce the dataset to 2 principal components (PC1 and PC2).
  - A scatter plot is created for each k value, showing how players are grouped in 2D space based on PCA.

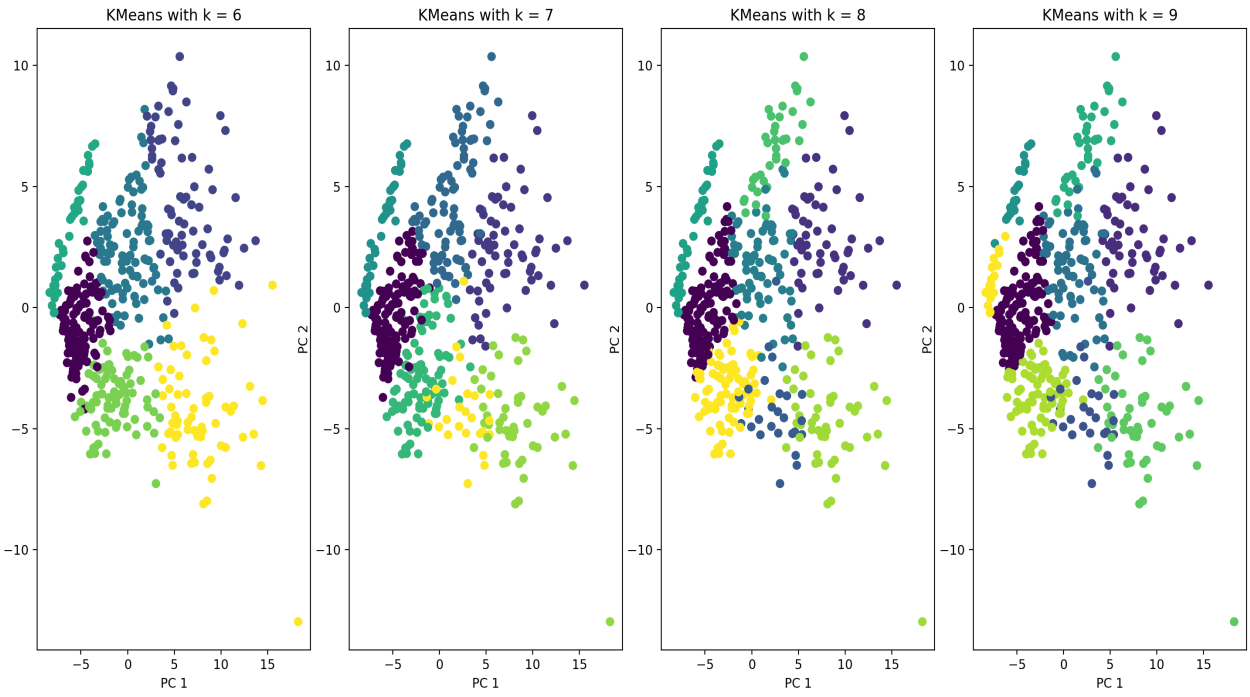


Figure 3.2: K-means clustering results

3.4 Result

The visualizations showed that **k = 8** provides relatively distinct and well-separated clusters. All elements clearly and consistently belong to their respective clusters.

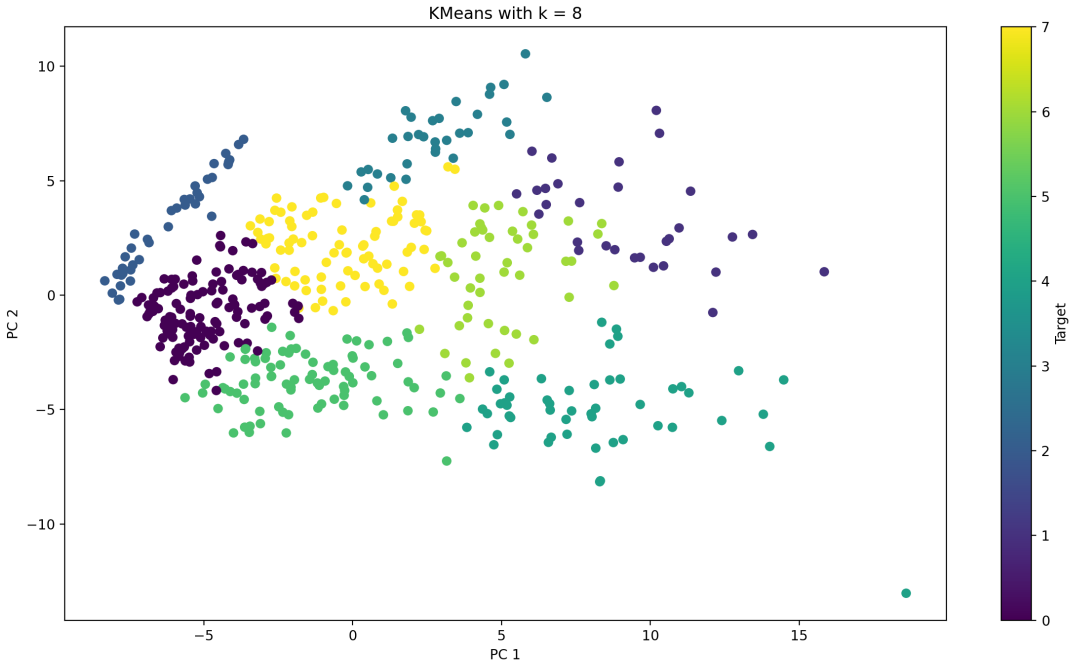


Figure 3.3: Final clustering result with k=8

Chapter 4: Player Transfer Values and Value Estimation

4.1 Objective

The objective of this section was to:

- Collect estimated transfer values for Premier League players from FootballTransfers.com.
- Match these values with previously collected player statistics (from FBref.com).
- Propose a method to estimate player value based on performance metrics.

4.2 Tools and Libraries Used

`requests`: to send HTTP POST requests to football-related APIs for retrieving transfer and player search data.

`pandas`: to reading, manipulating, filtering, and merging tabular data.

`rapidfuzz`: utilized for fuzzy string matching to align player names from different data sources.

`unidecode`: to normalize player names by removing accents and special characters.

`sklearn`: for machine learning algorithms and model evaluation.

`numpy`: for numerical computations and array operations.

`seaborn`: for creating attractive and informative statistical graphics.

`matplotlib`: for plotting graphs and visualizing data.

4.3 Methodology

4.3.1 Data Collection

Two APIs were used:

- Player values API: queried across 22 pages to retrieve the top 500 most valuable players in the EPL.
- Search API: used for special cases where players were not listed in the top ranks.

Data Collection from API

- Sent paginated POST requests (pages 1 to 22) to retrieve top 500 most valuable players.
- Extracted `player_name, age, team_short_name`, and `estimated_value.`

Filtering FBref Data

- Loaded `result.csv`, which contains FBref data.
- Filtered players with more than 900 minutes played.

Name Normalization and Matching

- Created a simplified `name_key` by removing accents and sorting name tokens to reduce matching ambiguity.
- Used `RapidFuzz` to match `player_name` from FootballTransfers with Player from FBref using fuzzy ratio scores.

Handling Missing Matches

- Identified unmatched players after fuzzy matching.
- Queried the **Search API** for these exception cases to manually retrieve their transfer values.
- Merged results with matched players.

Result

The final merged dataset includes:

- Player, Team, and Transfer Value
- Saved as `player_transfer_values.csv`

	Player	Team	Transfer Value
2	Aaron Ramsdale	Southampton	€18.7M
3	Aaron Wan-Bissaka	West Ham	€26.9M
4	Abdoulaye Doucouré	Everton	€5.8M
5	Adam Armstrong	West Bromwich	€15.1M
6	Adam Smith	Bournemouth	€1.5M
7	Adam Wharton	Crystal Palace	€48.9M
8	Adama Traoré	Fulham	€8M
9	Alejandro Garnacho	Manchester Utd	€60.7M
10	Alex Iwobi	Fulham	€30.3M
11	Alex Palmer	Ipswich Town	€1.6M
12	Alexander Isak	Newcastle Utd	€120.3M
13	Alexis Mac Allister	Liverpool	€106.1M
14	Alisson	Liverpool	€24.1M
15	Alphonse Areola	West Ham	€11.8M

Figure 4.1: Example of transfer value

4.3.2 Feature Selection and Modeling

Data collection

- **Player Performance Data** was loaded from `result.csv`, containing detailed statistics scraped from FBref.com. Only players with more than **900 minutes played** were retained

to ensure data reliability.

- **Transfer Value Data** was loaded from `player_transfer_values.csv`, created using a custom script that scrapes estimated market values from FootballTransfers.com.
- The two datasets were **merged** on the `Player name` field to associate performance data with corresponding transfer values.
- Placeholder values such as `'N/a'` were replaced with `NaT` to ensure consistent data types and facilitate downstream processing.

Statistics

- The merged dataset consists of **299 rows and 79 columns** (df.shape), covering a wide range of player performance metrics.
- A review of missing data indicates that several features contain a significant proportion of null values.
- The dataset also contains **17 categorical features**, including Player, Team, Position, etc.

Data preprocessing

To prepare the dataset for modeling, the following preprocessing steps were applied:

- **Outlier Removal**: The player Mohamed Salah was removed from the dataset.

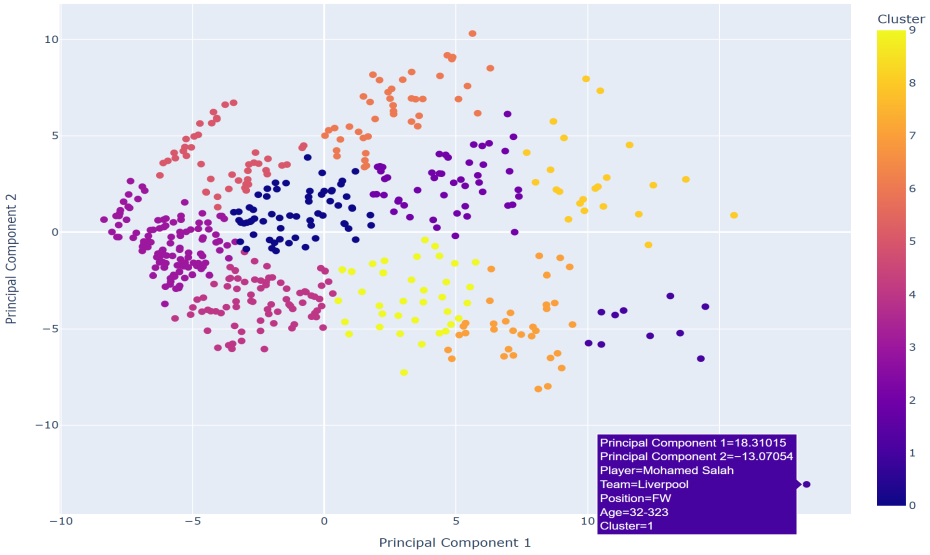


Figure 4.2: PCA plot

- **Handling Missing Data**:
  - Features with excessive missing values were dropped.
  - For partially missing features, missing values were imputed using the column mean.
- **Feature Transformation**:
  - Age was converted from a string format to a decimal representation.
  - Transfer Value was cleaned and converted to float.
- **Data Type Conversion**: Numerical conversion was applied to various features.
- **Categorical Encoding**: Categorical features were encoded using Label Encoding.
- **Feature Reduction**: The Position column was removed.

Data visualization

- **Correlation Matrix Calculation**: Computed using the .corr() method.
- **Heatmap Visualization**: Using coolwarm color map to represent correlation strength.

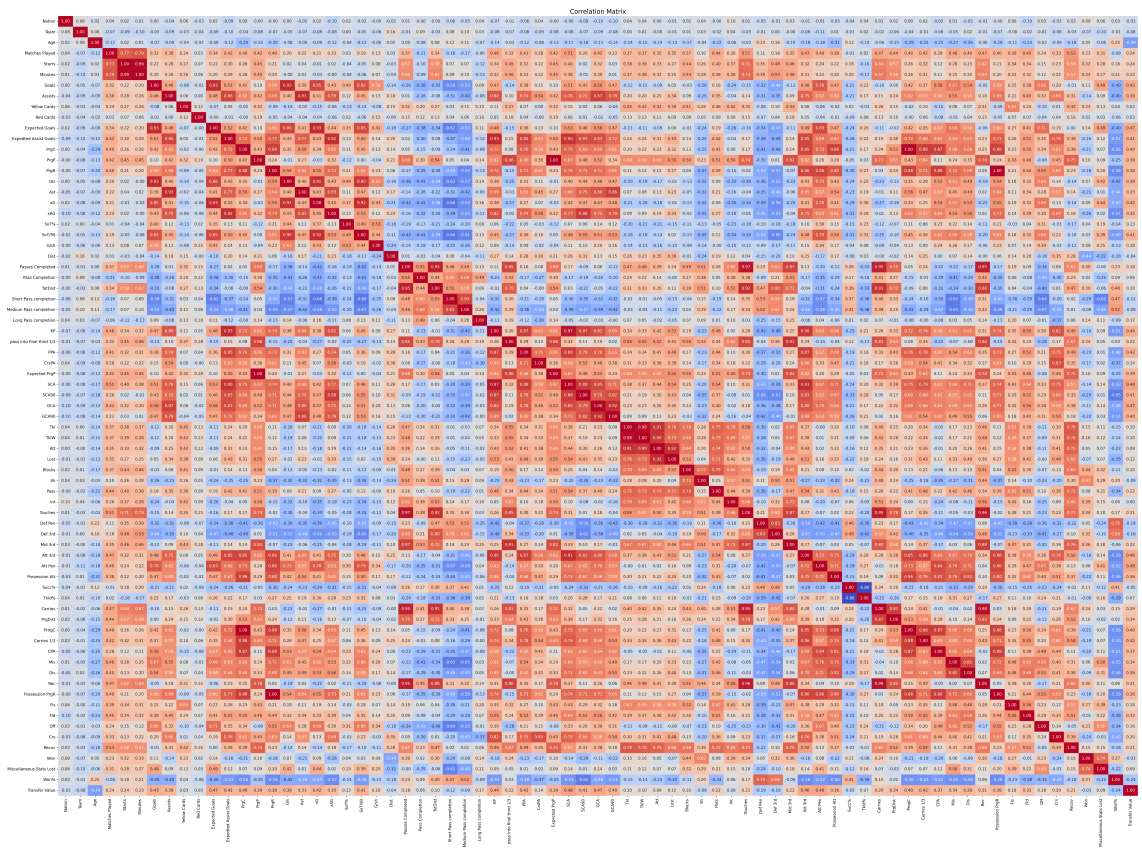


Figure 4.3: Correlation heatmap

Model building

Choose important features for train model (Features Selection)

- Step 1: Correlation with `Transfer Value` - Features with absolute correlation > 0.2 were retained.
- Step 2: Multicollinearity Reduction - Features with pairwise correlation > 0.9 were dropped.
- Step 3: Final Feature Set - Important features were identified based on correlation.

Split data train and test

- 80% of the data used to train the model.
- 20% of the data held out for testing.
- Training set size: 240

- Test set size: 61

Create model and train

- The following regression models were considered:

- Linear Regression
- Random Forest Regressor
- Gradient Boosting Regressor
- K-Nearest Neighbors Regressor
- AdaBoost Regressor
- Decision Tree Regressor

- Evaluation Metrics:

- R^2 Score (Train & Test)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Training Time
- Prediction Time

Regression Results

	Model	Train_Score	Test_Score	MSE	RMSE	MAE	Train Time	Prediction Time
0	Linear Regression	0.593644	0.726413	191.511434	13.838766	11.671978	0.018907	0.001448
1	Random Forest	0.923188	0.648850	245.805993	15.678201	12.594246	0.353536	0.009108
2	Gradient Boosting	0.966843	0.565439	304.194061	17.441160	13.695321	0.181925	0.010125
3	KNN	0.355491	0.118629	616.961777	24.838715	20.764262	0.001142	0.197435
4	AdaBoost	0.709293	0.458350	379.156088	19.471931	16.656730	0.096389	0.009021
5	Decision Tree	1.000000	0.441946	390.639180	19.764594	14.611475	0.006330	0.001326

Figure 4.4: Comparison of regression models based on evaluation metrics

Based on the evaluation, **Linear Regression** was chosen as the best model for the regression task.

Model evaluation

Evaluation Metrics

- Train R^2: 0.59
- Test R^2: 0.73
- MAE: 11.67

Cross-Validation 5-fold cross-validation was applied using the R^2 metric:

- Cross-validated R^2 scores: [0.61843055 0.44460644 0.54099377 0.46227747 0.41716326]
- Mean CV R^2: 0.50

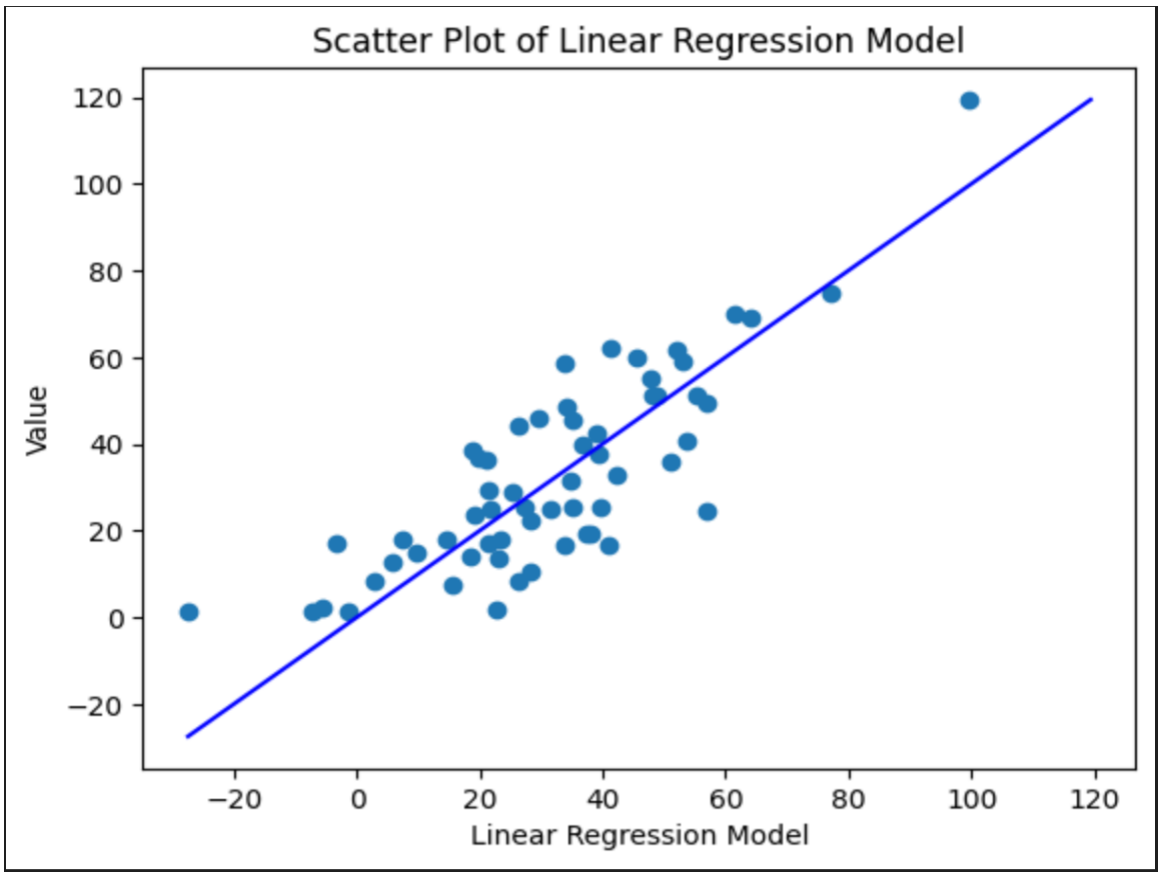


Figure 4.5: Predictions vs actual values

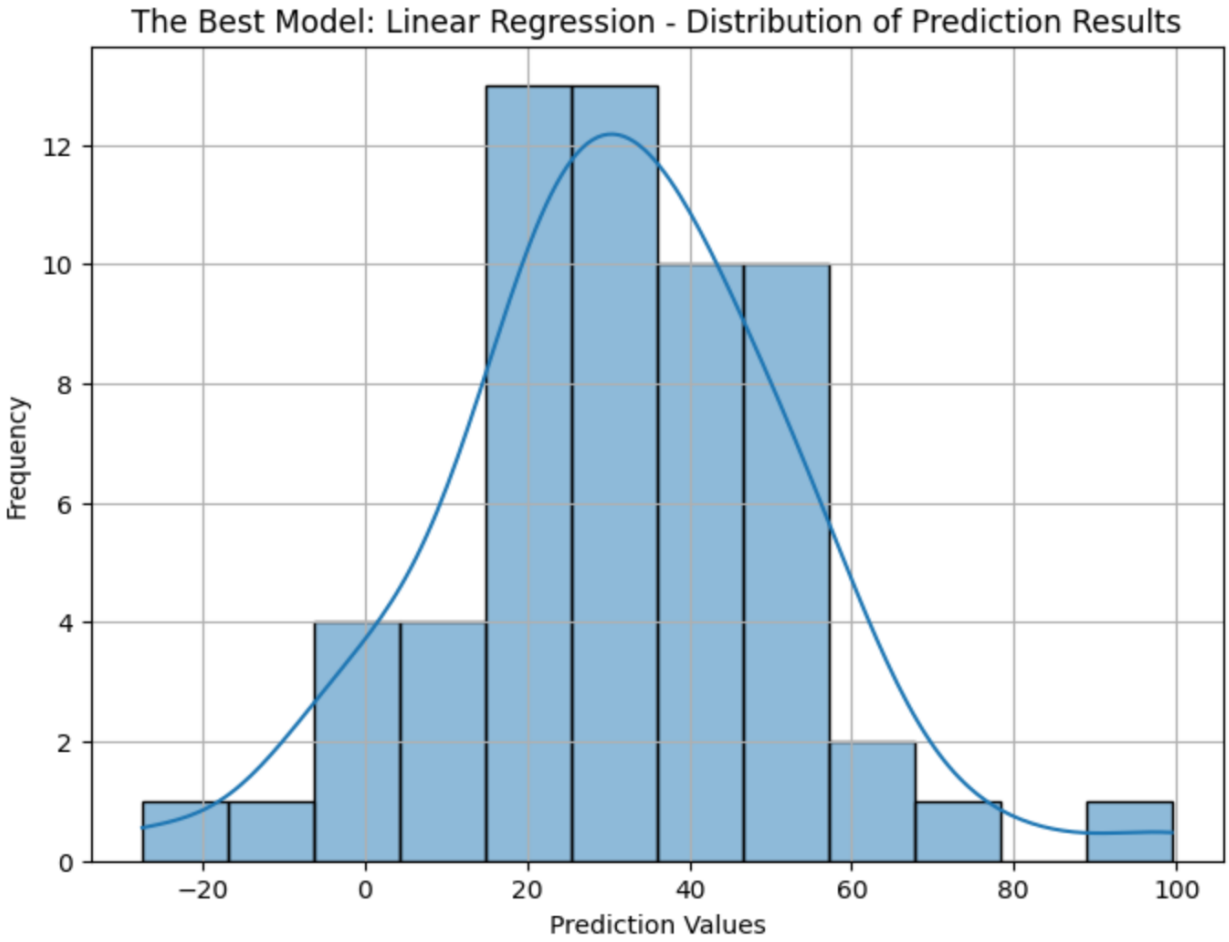


Figure 4.6: Prediction distribution

Conclusion

Positive Aspects

- Successfully integrated end-to-end data science processes, including web scraping, preprocessing, statistical analysis, clustering, and predictive modeling.
- Identified top-performing teams (Liverpool) and players, providing valuable metrics for football analytics.
- Generated histograms, PCA plots, and heatmaps to intuitively communicate data trends and model performance.
- Achieved a competitive test R^2 score of 0.65 for transfer value prediction using Linear Regression.

Negative Aspects

- Reliance on specific websites (FBref, FootballTransfers) risks data accessibility issues if APIs or site structures change.
- Imputing missing values with zeros and simplifying age formats ("23-45" to 23.12) may introduce inaccuracies.
- Cross-validation revealed moderate consistency (mean CV $R^2 = 0.47$), indicating potential overfitting to the training set.
- Position-specific metrics (goalkeeping stats) had high missing rates, limiting their utility in analysis.

Future Developments

- Expand datasets to include multiple seasons or leagues (La Liga, Bundesliga) for broader insights.

- Experiment with ensemble methods (XGBoost, neural networks) to improve prediction accuracy and handle non-linear relationships.
- Develop automated pipelines to handle API changes and ensure long-term data accessibility.

References

Data Sources

FBref.com: Football Statistics and Analysis. Retrieved from https://fbref.com/

FootballTransfers.com: Player Transfer Values and Market Insights. Retrieved from https://www.footbal

Python Libraries & Tools

pandas: Powerful data manipulation and analysis toolkit. Retrieved from https://pandas.pydata.org/

BeautifulSoup: Library for parsing HTML and XML documents. Retrieved from https://www.crummy.co

scikit-learn: Machine learning library for Python. Retrieved from https://scikit-learn.org/

matplotlib: Comprehensive plotting library. Retrieved from https://matplotlib.org/

seaborn: Statistical data visualization library. Retrieved from https://seaborn.pydata.org/

requests: HTTP library for Python. Retrieved from https://requests.readthedocs.io/

rapidfuzz: Fast string matching library. Retrieved from https://github.com/maxbachmann/RapidFuzz

unidecode: Text normalization for Unicode characters. Retrieved from https://pypi.org/project/Unidecod

Methodology & Algorithms

K-means Clustering: Unsupervised clustering algorithm. Scikit-learn Documentation. Retrieved from https://scikit-learn.org/stable/modules/clustering.html#k-means

Principal Component Analysis (PCA): Dimensionality reduction technique. Scikit-learn Documentation. Retrieved from https://scikit-learn.org/stable/modules/decomposition.html#pca