FIFA 24 Player Value Prediction and Analysis Using Player Performance Metrics

-

Name	Netid
• Suneeth Kunche	ks1983
• Ram Sampreeth Budireddy	rb1424
• Amrutha Karuturi	ak2508
• Arvind Chary Padala	ap2522

TABLE OF CONTENTS

Contents	Page No
Abstract	03
Introduction	04
EDA ● Data Description ● Data PreProcessing ● Visualisation	05
Methodology ● Modelling ● Hypothesis Testing ● Model Improvements	12
Conclusion	26

ABSTRACT

This project delves into the domain of soccer player valuation by employing regression and hypothesis testing techniques on the FIFA 24 Player Stats Dataset. The primary objective is to predict player values using linear regression models, unravelling insights into the dynamics of market valuation. Additionally, the project explores the impact of various attributes on predicted values through hypothesis testing, offering a comprehensive understanding of the role-dependent dynamics in player valuation. The combination of regression analysis and hypothesis testing equips soccer clubs with valuable tools for budgeting and team optimization. We used a variety of regression approaches: the OLS model with log transformations and the OLS model with Box-Cox transformation. We checked Q-Q plots for residual normality and analysed the performance of the model on the basis of R2, adjusted R2, and Root Mean Squared Error (RMSE).

INTRODUCTION

In the context of professional soccer team management, the art of player valuation stands as a pivotal element, shaping decisions on financial resource allocation and team composition. This project centres around the FIFA 24 Player Stats Dataset, sourced from Kaggle, as a comprehensive source of information for understanding player valuation dynamics. The project unfolds with a twofold agenda: developing a reliable predictive model for player values and exploring the nuanced aspects of market valuation concerning player roles. Through the application of regression analysis and hypothesis testing, this project aspires to furnish soccer clubs with analytical tools for strategic decision-making.

Commencing with a meticulous exploration of the dataset, emphasis is placed on data cleaning and formatting to ensure the integrity of subsequent analyses. Outlier treatment and standard scaling further refine the dataset, rendering it amenable to in-depth scrutiny. Examination of collinearity and correlation patterns sheds light on the intricate relationships within player statistics, establishing a foundational understanding of the dataset.

Subsequently, the project incorporates feature selection techniques to pinpoint the most influential factors in predicting player values. This step is integral for constructing a streamlined model that encapsulates the essence of player valuation within the FIFA 24 gaming environment.

The modeling phase encompasses a spectrum of regression techniques, ranging from conventional Linear Regression to more sophisticated methods such as Ridge Regression, Lasso Regression, Random Forest Regression, and XGBoost. Each model is tailored to unveil distinct facets of the dataset, providing a comprehensive view of player valuation dynamics.

Our approach revolves around employing Linear Regression as the primary predictive model. The essence of Linear Regression lies in finding the line of best fit that minimizes the sum of squared differences between observed data and predicted values. In this context, the project seeks to uncover the linear relationships between player statistics and market values.

To enhance the predictive performance of our linear regression model, in a parallel fashion, we perform logarithmic and box-cox transformations on the dependent variable, the player value(Million) or ‘y'. These transformations serve the dual purpose of normalizing the distribution of the dependent variable and stabilizing its variance, making it more suitable for application in an OLS regression model.

In parallel with regression analysis, the project engages in hypothesis testing to discern whether player roles significantly impact predicted values. This examination adds depth to our understanding, unraveling role-dependent dynamics contributing to a player's market value.

Beyond these conventional techniques, our analysis extends to advanced models such as Ridge Regression and Lasso Regression, which introduce regularisation to mitigate overfitting and enhance model generalization. Furthermore, the exploration of Random Forest Regression and XGBoost aims to harness the power of ensemble learning, capturing intricate patterns and interactions within player statistics.

In the pursuit of empowering soccer clubs with actionable insights, this project aspires to contribute a comprehensive understanding of individual player values and the intricate role-dependent dynamics that underpin the bidding and team-building processes in the realm of FIFA 24.

Exploratory Data Analysis

Data Description:

The FIFA Football Players Dataset is a comprehensive collection of information about football (soccer) players from around the world. This dataset offers a wealth of attributes related to each player, making it a valuable resource for various analyses and insights into the realm of football, both for gaming enthusiasts and real-world sports enthusiasts

Data overview:

Attributes:

- Player: The name of the football player.
- Country: The nationality or home country of the player.
- Height: The height of the player in centimeters.
- Weight: The weight of the player in kilograms.
- Age: The age of the player.
- Club: The club to which the player is currently affiliated.
- Ball Control: Player's skill in controlling the ball.
- Dribbling: Player's dribbling ability.
- Marking: Player's marking skill.
- Slide Tackle: Player's ability to perform slide tackles.
- Stand Tackle: Player's ability to perform standing tackles.
- Aggression: Player's aggression level.
- Reactions: Player's reaction time.
- Attacking Position: Player's positioning for attacking plays.

- Interceptions: Player's skill in intercepting passes.
- Vision: Player's vision on the field.
- Composure: Player's composure under pressure.
- Crossing: Player's ability to deliver crosses.
- Short Pass: Player's short passing accuracy.
- Long Pass: Player's ability in long passing.
- Acceleration: Player's acceleration on the field.
- Stamina: Player's stamina level.
- Strength: Player's physical strength.
- Balance: Player's balance while playing.
- Sprint Speed: Player's speed in sprints.
- Agility: Player's agility in maneuvering.
- Jumping: Player's jumping ability.
- Heading: Player's heading skills.
- Shot Power: Player's power in shooting.
- Finishing: Player's finishing skills.
- Long Shots: Player's ability to make long-range shots.
- Curve: Player's ability to curve the ball.
- Free Kick Accuracy: Player's accuracy in free-kick situations.
- Penalties: Player's penalty-taking skills.
- Volleys: Player's volleying skills.
- Goalkeeper Positioning: Goalkeeper's positioning attribute (specific to goalkeepers).
- Goalkeeper Diving: Goalkeeper's diving ability (specific to goalkeepers).
- Goalkeeper Handling: Goalkeeper's ball-handling skill (specific to goalkeepers).
- Goalkeeper Kicking: Goalkeeper's kicking ability (specific to goalkeepers).
- Goalkeeper Reflexes: Goalkeeper's reflexes (specific to goalkeepers).
- Value: The estimated value of the player

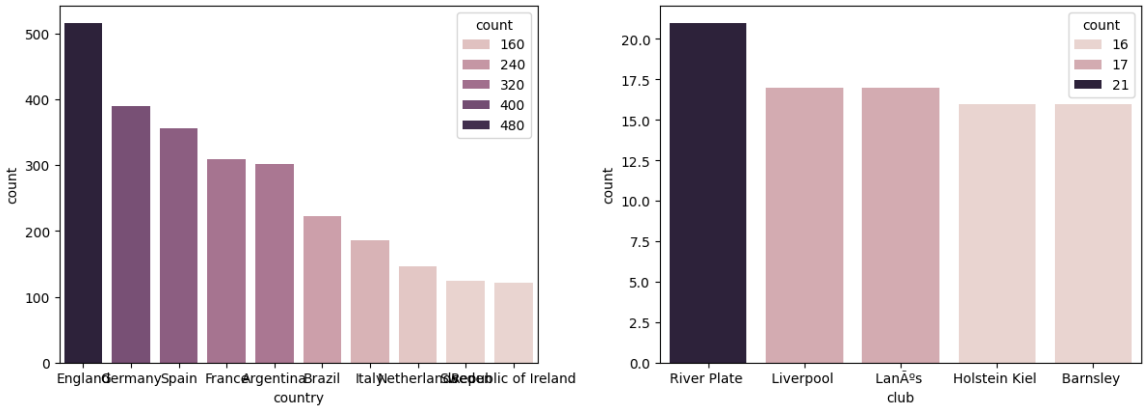
Data Preprocessing:

Data Cleaning:

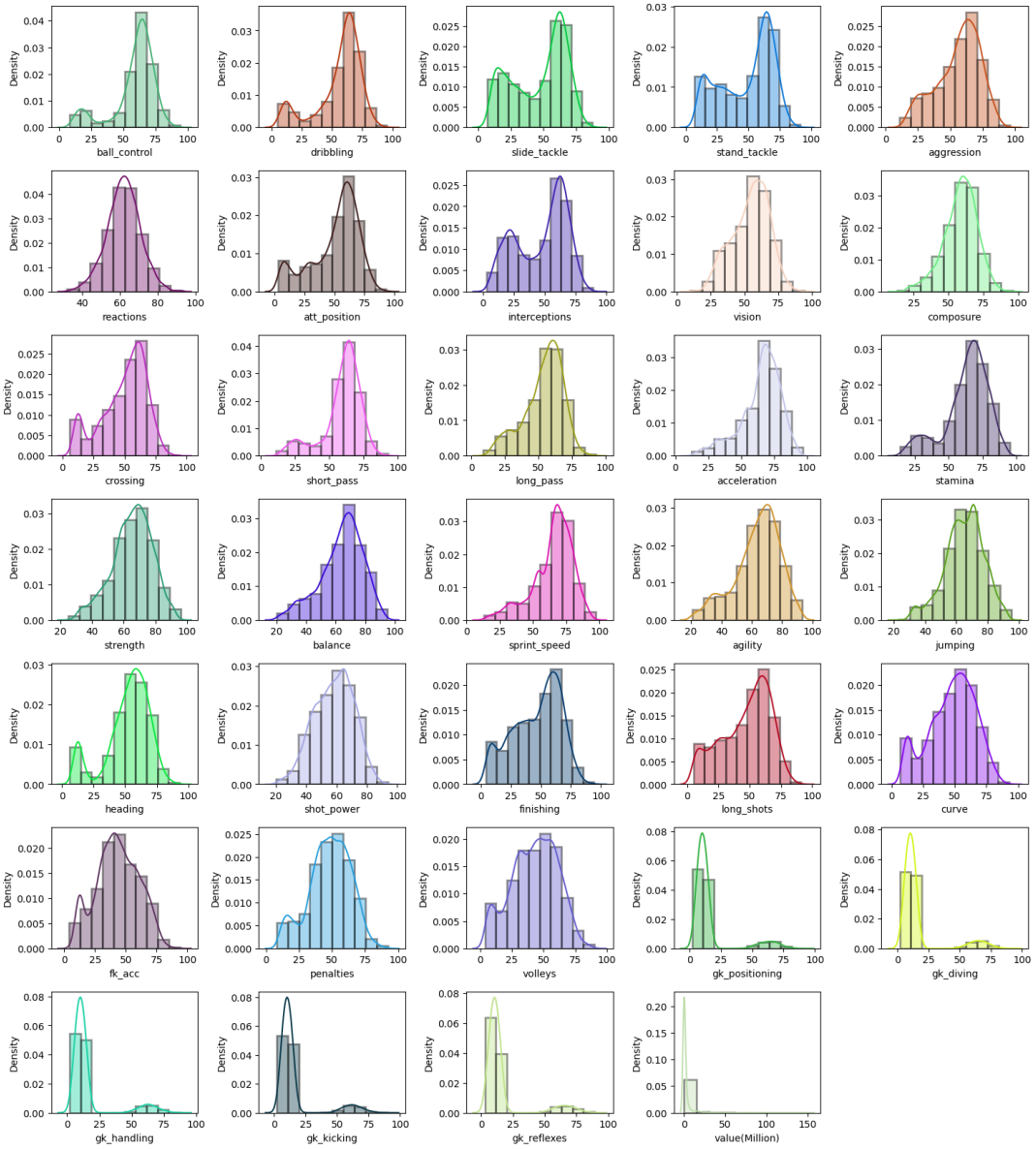
- **Value column conversion:** The Player value column in the data is in string format, so we removed the unnecessary characters and converted it into float. Further, we scaled the player value to millions to make visualizations simpler.
- **Null value treatment:** More than 90% of the column has missing values, so we have decided to drop the particular column as replacing the missing values might result in reduced accuracy.
- **Outlier treatment:** we defined a function, remove_outliers, which takes a DataFrame and a z-score threshold as parameters. Utilizing the z-score method from the scipy.stats module, the function identifies and removes outliers by filtering rows where the absolute z-scores in all columns exceed the specified threshold. The resulting DataFrame, named df_no_outliers,

contains data points with z-scores within the defined limits, mitigating the impact of extreme values on subsequent analyses.

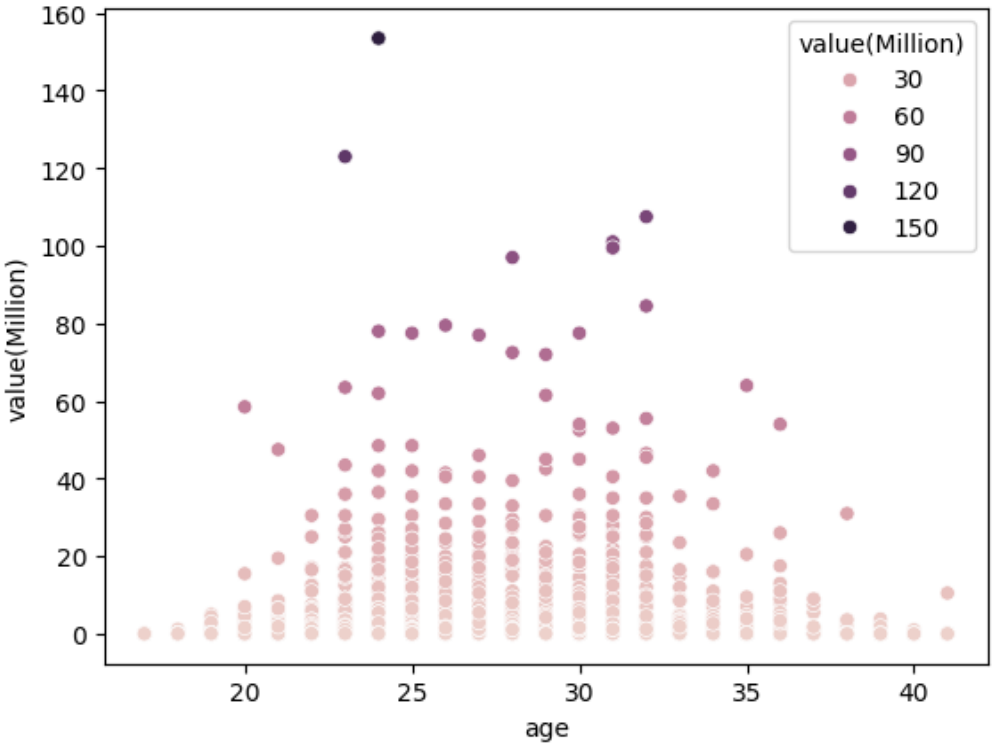
Top countries and clubs based on number of players:



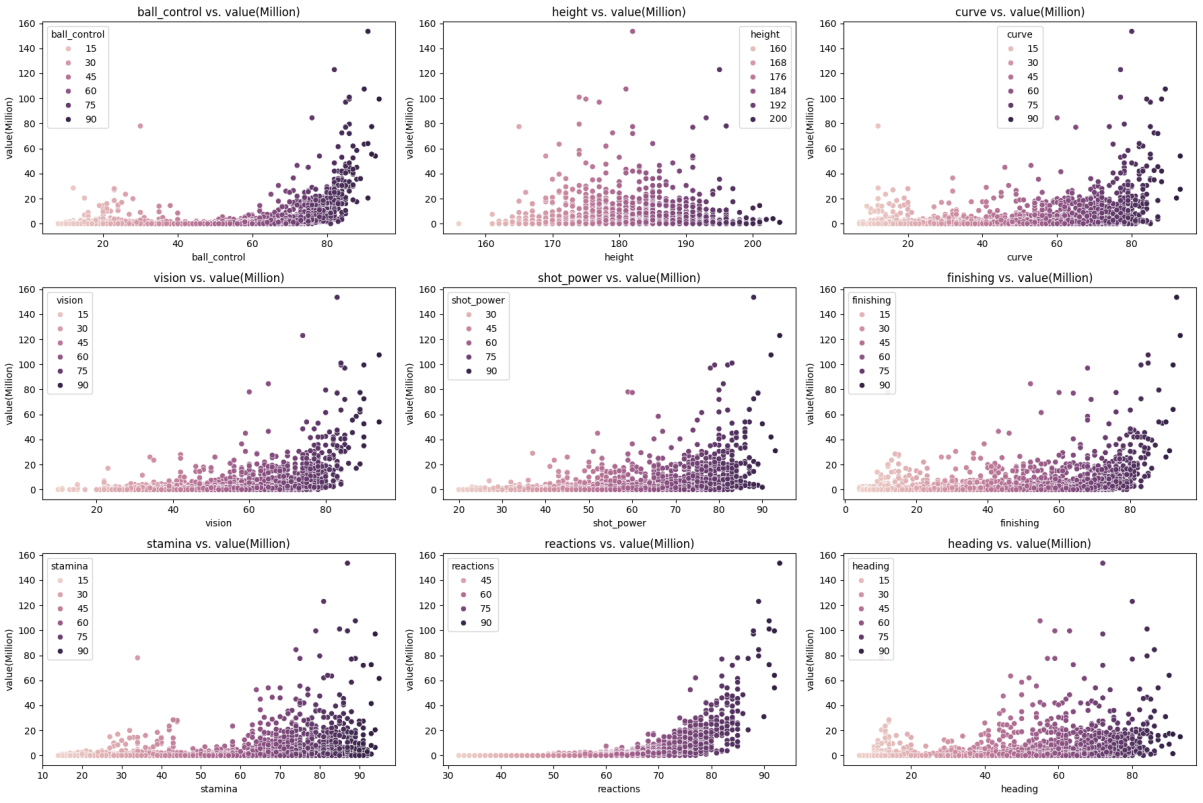
Distributions of all the numerical attributes in the data:



Scatterplot of Player ‘age’ vs. ‘Value(Million):

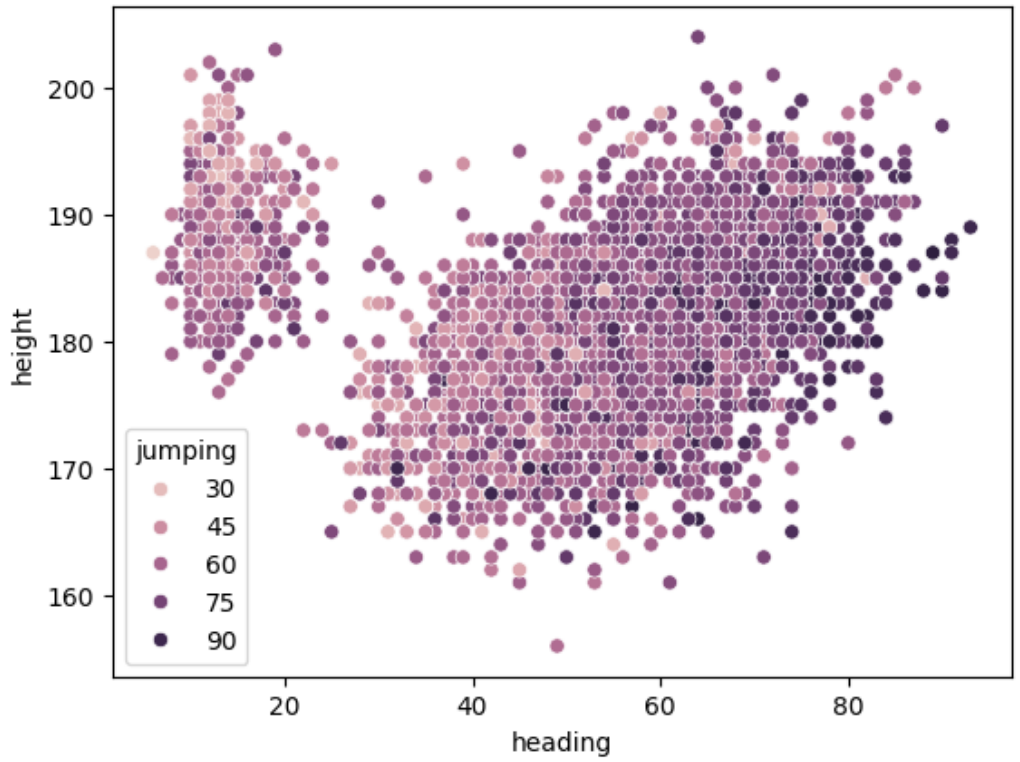


Scatterplots of Some other important features against the player value:

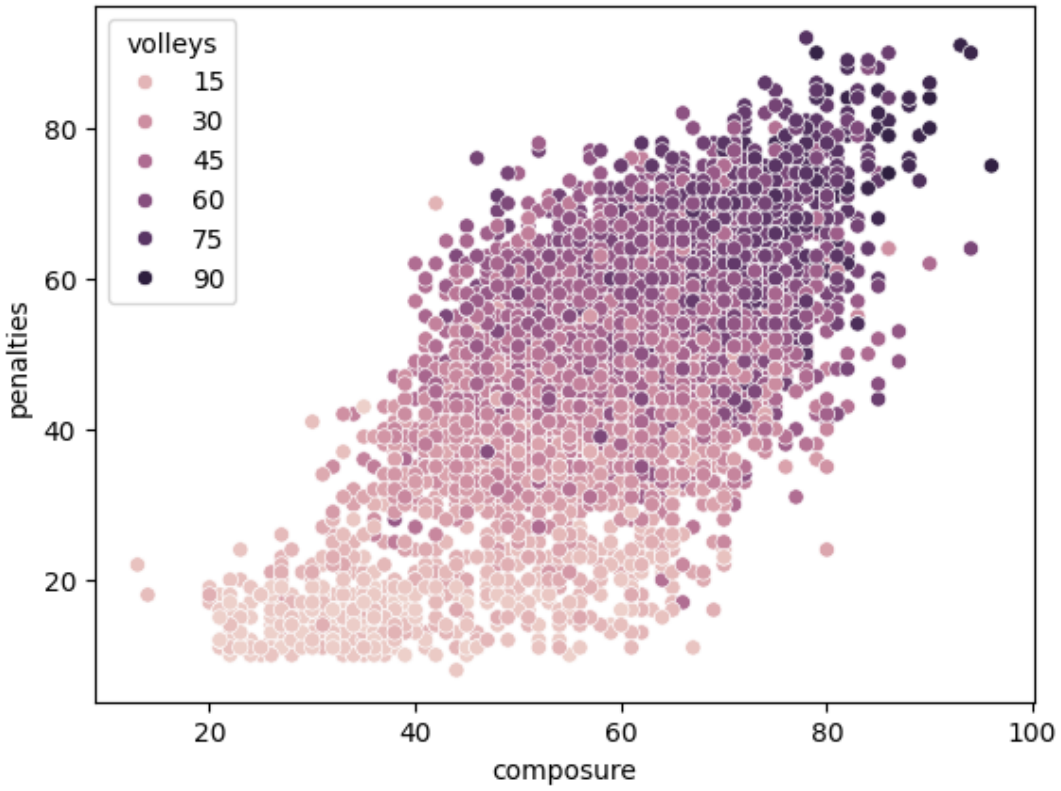


Multivariate plots:

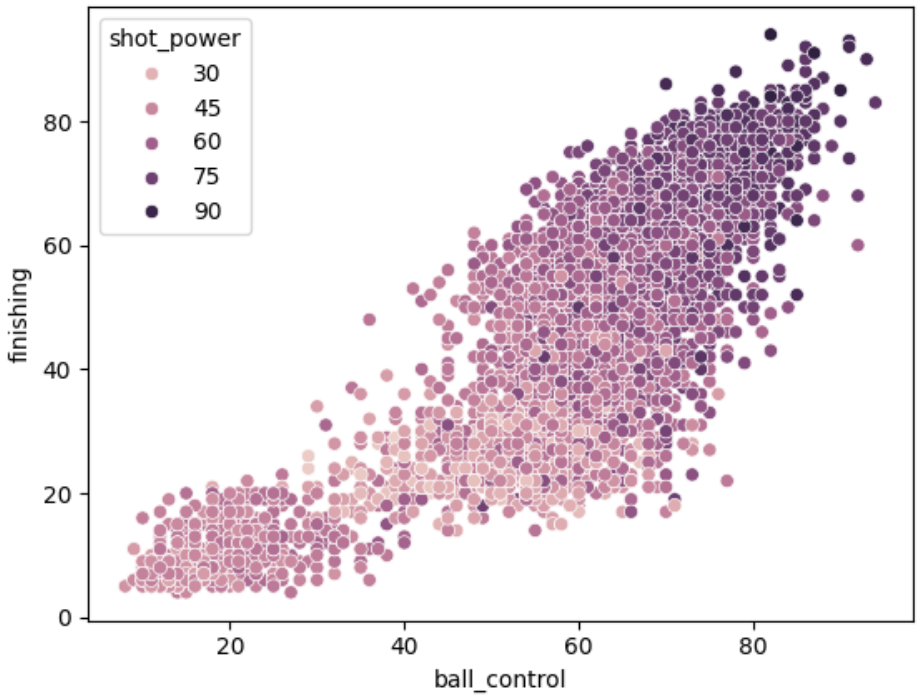
Heading vs. Height vs. Jumping



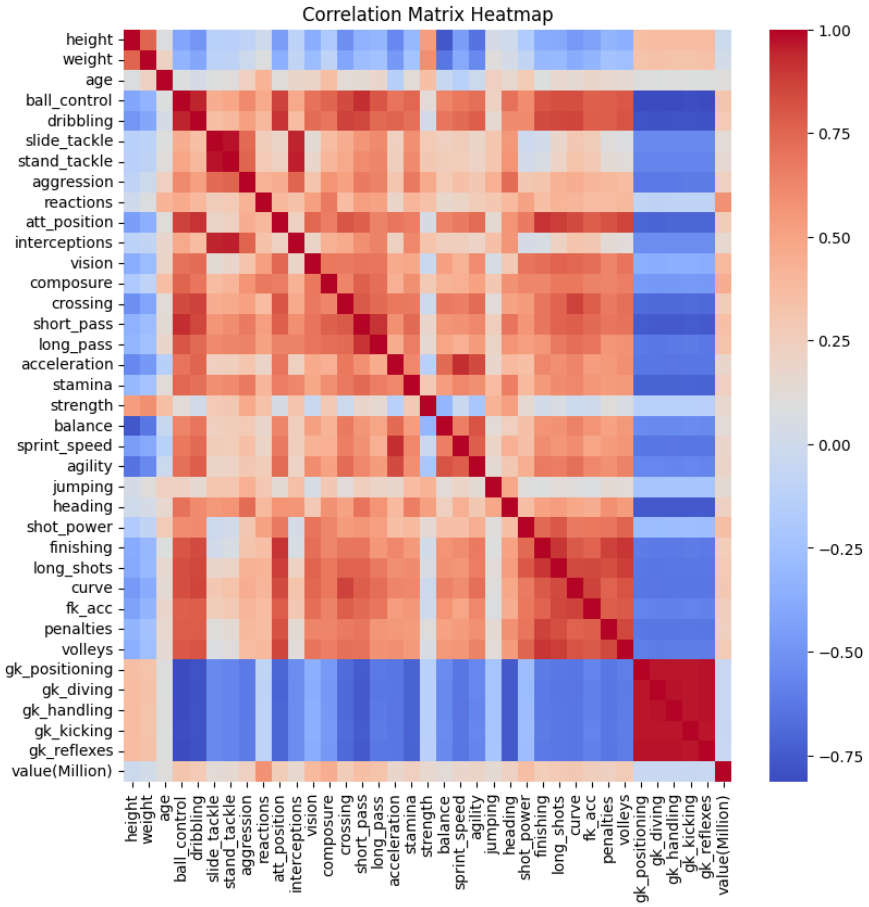
Composure vs. penalties vs. volleys



Ball Control vs. Finishing vs. Shot power



Correlation heatmap



METHODOLOGY

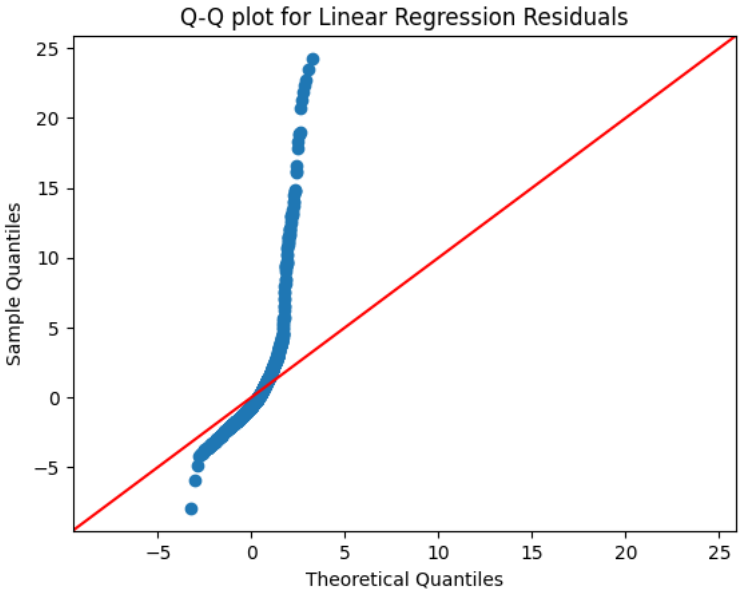
Modelling:

Linear regression (sci-kit-learn)

Model Overview:

The model is built using a set of features represented by the coefficients. Each coefficient indicates the change in the dependent variable for a one-unit change in the corresponding independent variable, assuming other variables are held constant. Negative coefficients suggest a negative impact on the dependent variable, while positive coefficients suggest a positive impact.

Model Evaluation:
Root Mean Squared Error (RMSE): The RMSE value of approximately 3.23 indicates the average magnitude of errors between the predicted and actual player values. A lower RMSE suggests a better fit of the model, but the interpretation depends on the scale of the target variable.
R-squared (R2) Score: With an R2 score of around 0.39, the model explains about 39% of the variance in the player values. While not exceptionally high, this indicates a moderate level of predictive power. It's important to consider domain-specific factors that may contribute to the inherent variability in player values



- Compared to the linear regression model(scikit learn), The OLS model provides additional statistical information about the model, such as p-values for each coefficient, confidence intervals, and more.
- The OLS model incorporates statistical tests to evaluate the significance of each coefficient

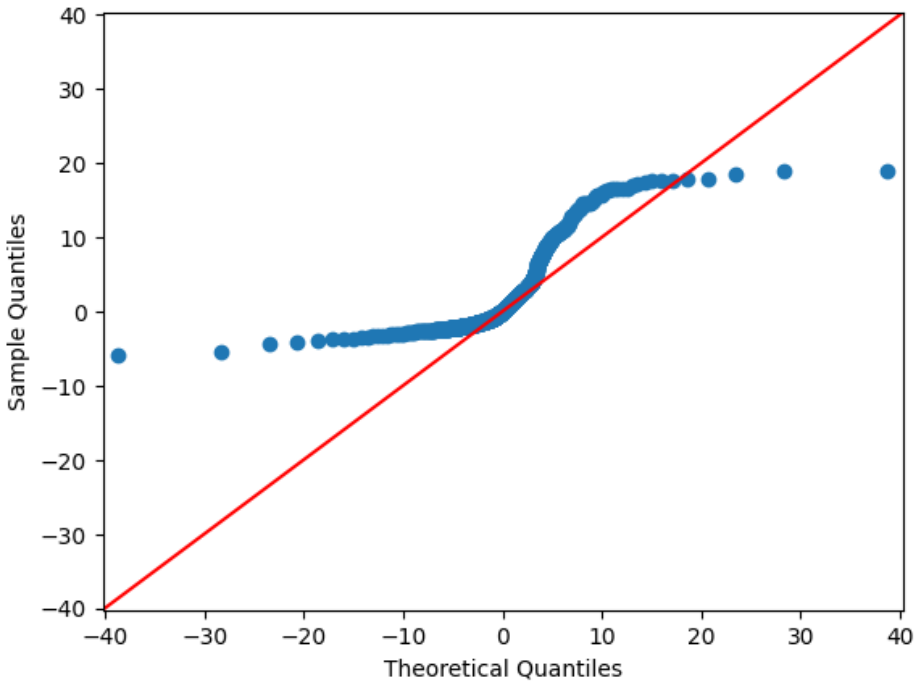
Ordinary Least Squares Estimation:

Model Overview:

- The Ordinary Least Squares (OLS) model provides an overview of the relationships between various player attributes and their market values.
- The model's R-squared value of approximately 0.489 indicates that the features included explain about 48.9% of the variance in player values. The Adj. R-squared adjusts for the number of predictors in the model, offering a more accurate measure of goodness of fit.
- The F-statistic of 148.2 with a p-value close to zero suggests that the model is statistically significant.

Model Evaluation:

- The root mean squared error (RMSE) of approximately 3.22 suggests the average magnitude of errors between the predicted and actual player values, which is similar to the previous Linear Regression model's performance.
- The Durbin-Watson statistic of approximately 1.955 suggests that there might be some autocorrelation present in the residuals. Further investigation or adjustment may be considered.



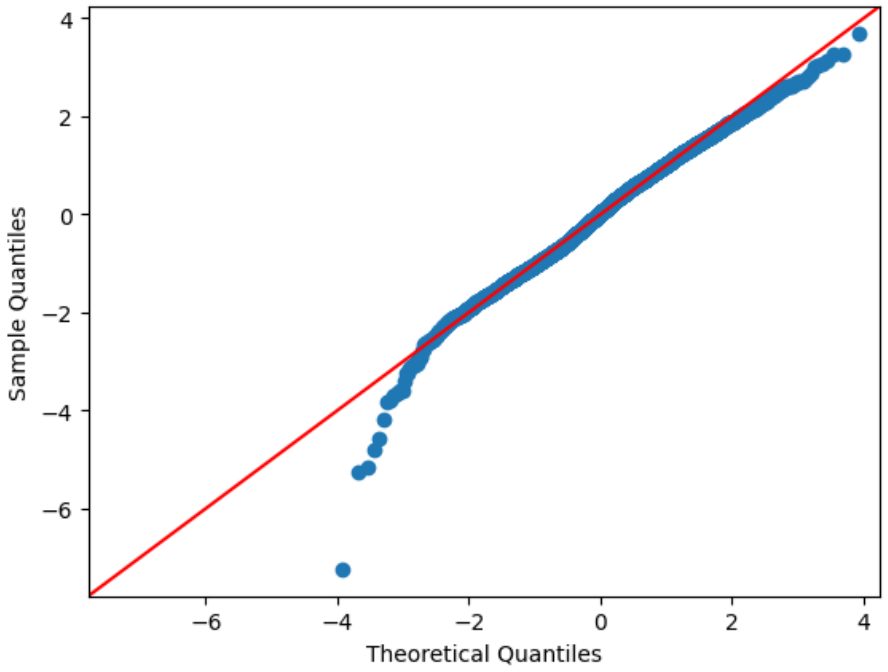
OLS with Logarithmic Transformation:

Model Overview:

The application of a logarithmic transformation to the dependent variable ('value(Million)') significantly improves the model's explanatory power. The R-squared value increases substantially to approximately 0.825, indicating that the logarithmic transformation explains a larger proportion of the variance in player values.

Model Evaluation:

The root mean squared error (RMSE) of approximately 4.72 is higher than the previous models. This indicates that, while the logarithmic transformation improves the R-squared value, it may result in larger errors when predicting the original player values.



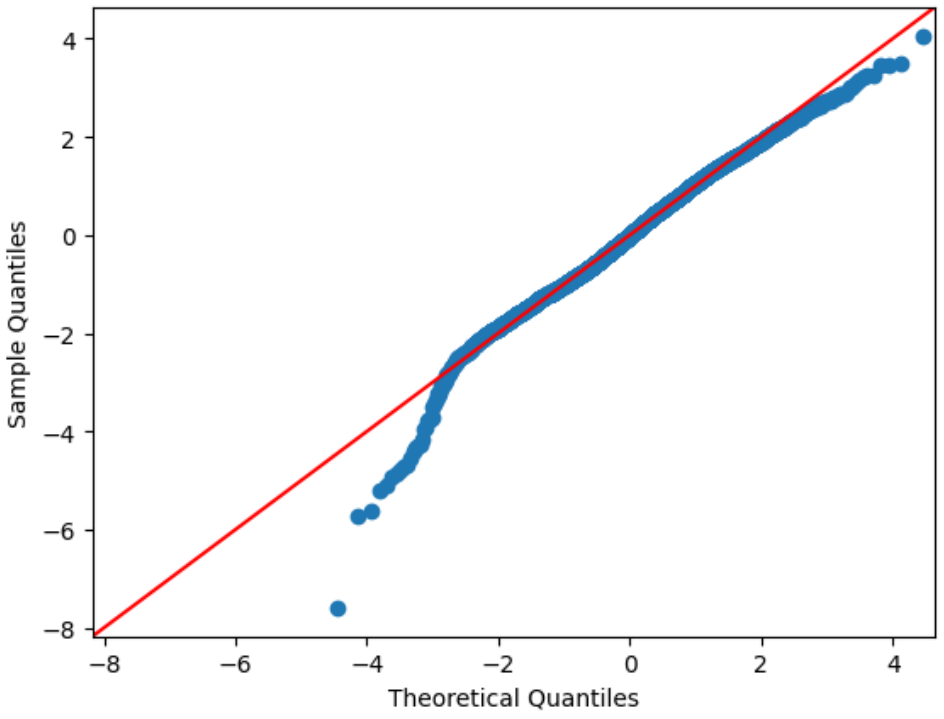
OLS with Box-Cox Transformation:

Model Overview:

The application of the Box-Cox transformation to the dependent variable ('value(Million)') further enhances the model's explanatory power. The R-squared value increases to approximately 0.839, indicating that the Box-Cox transformation explains a substantial proportion of the variance in player values.

Model Evaluation:

The root mean squared error (RMSE) of approximately 5.04 is higher than the original model, indicating that the Box-Cox transformation may introduce larger errors when predicting the original player values.



Ridge

Model Overview:

Ridge regression puts constraints on the coefficients (w). The penalty term (λ) regularises the coefficients such that if they take large values it shrinks them by penalising the optimization function. By shrinking the coefficients Ridge regression reduces the model complexity and multi-collinearity.

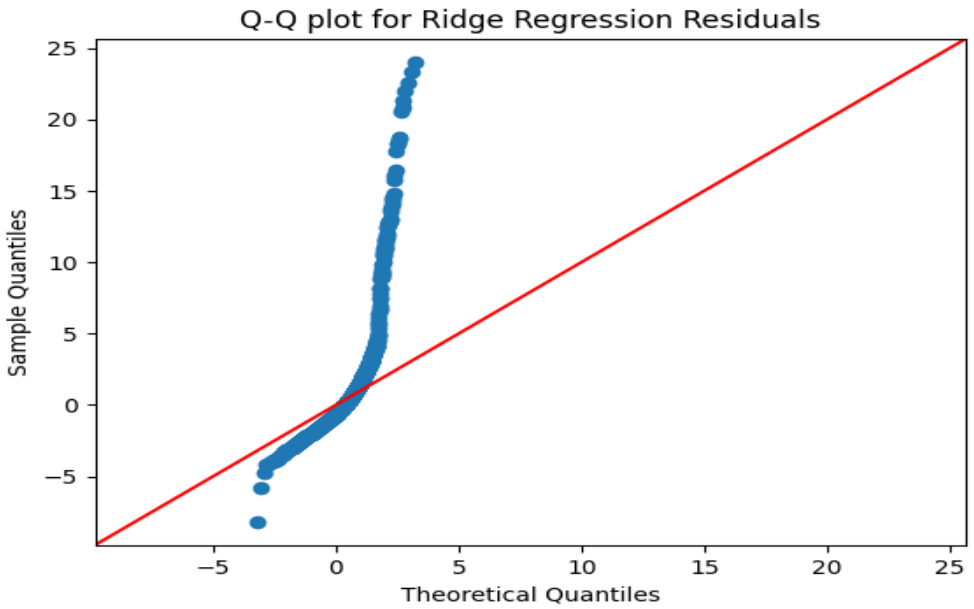
Model Evaluation:

- The Ridge RMSE (Root Mean Squared Error) is approximately 3.20, indicating the predictive performance of the Ridge Regression model.
- The R-squared value of around 0.40 suggests that the model explains 40% of the variance in player values

r-square as follows:

```
ridge_rmse = np.sqrt(mean_squared_error(y_test, ridge.predict(X_test)))
print("Ridge RMSE:", ridge_rmse)
print("r2_score", r2_score(y_test,ridge.predict(X_test)))
```

```
Ridge RMSE: 3.2049928647223513
r2_score 0.4001729372757731
```



Lasso

Model Overview:

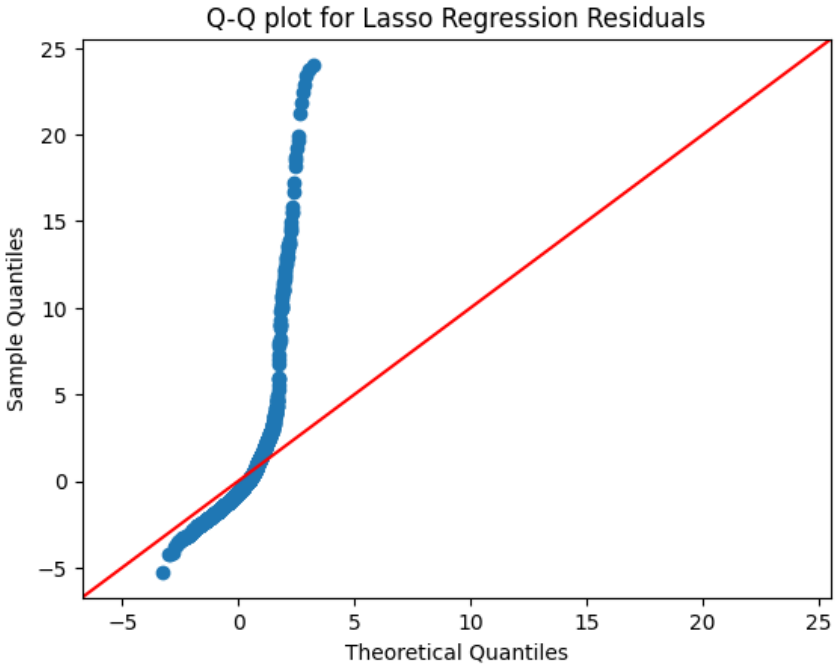
LASSO regression, also known as L1 regularisation, is a popular technique used in statistical modeling and machine learning to estimate the relationships between variables and make predictions. LASSO stands for Least Absolute Shrinkage and Selection Operator.

Model Evaluation:

- The Lasso RMSE (Root Mean Squared Error) is approximately 3.28, indicating the predictive performance of the Lasso Regression model.
- The R-squared value of around 0.37 suggests that the model explains 37% of the variance in player values.

```
lasso_rmse = np.sqrt(mean_squared_error(y_test, lasso.predict(X_test)))
print("Lasso RMSE:", lasso_rmse)
print("r2_score", r2_score(y_test,lasso.predict(X_test)))
```

```
Lasso RMSE: 3.2830299287585074
r2_score 0.37060743844397737
```



Hypothesis Testing:

1) **Model with the possible features of the striker:**

Model: value ~ dribbling + crossing + att_position + reactions + short_pass + acceleration + sprint_speed + agility + heading + shot_power + finishing + long_shots + volleys

```
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -259.83746    8.08881 -32.123  < 2e-16 ***
dribbling       0.11214    0.19703   0.569 0.569274
crossing        0.13549    0.10889   1.244 0.213440
att_position   -0.35498    0.14704  -2.414 0.015803 *
ball_control   -0.04521    0.24108  -0.188 0.851241
reactions       3.95218    0.12797  30.883  < 2e-16 ***
short_pass      0.37841    0.18050   2.096 0.036085 *
acceleration    0.12310    0.17726   0.694 0.487413
sprint_speed    0.28020    0.15609   1.795 0.072679 .
agility        -0.25003    0.12783  -1.956 0.050515 .
heading        -0.39468    0.08391  -4.704 2.62e-06 ***
shot_power      0.48674    0.13289   3.663 0.000252 ***
finishing       0.50637    0.14845   3.411 0.000651 ***
long_shots     -0.66480    0.14417  -4.611 4.09e-06 ***
volleys         0.29835    0.12708   2.348 0.018926 *
```

ANOVA TEST:

By considering the above summary of the model let

H0: βdribbling=βcrossing=βball_control=βacceleration= 0 Vs

H1: At Least one of βdribbling, βcrossing, βball_control, βacceleration ≠ 0

```
Analysis of Variance Table

Model 1: value ~ att_position + reactions + short_pass + sprint_speed +
    agility + heading + shot_power + finishing + long_shots +
    volleys
Model 2: value ~ dribbling + crossing + ball_control + att_position +
    reactions + short_pass + acceleration + sprint_speed + agility +
    heading + shot_power + finishing + long_shots + volleys
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1   5671 24295077
2   5667 24281312  4     13766 0.8032 0.5229
```

p-value is large => H0(restricted model) not rejected => dribbling, crossing, ball_control, acceleration do not have significant effect on the value

2) **Model with the possible features of the midfielder:**

Model : value ~ vision + ball_control + crossing + dribbling + reactions + long_pass + stamina + sprint_speed + agility + short_pass + finishing + long_shots

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.551e+02  7.415e+00 -34.408  < 2e-16 ***
vision        7.145e-01  1.216e-01   5.878 4.39e-09 ***
ball_control -1.183e-01  2.323e-01  -0.509 0.610690
crossing      1.186e-01  1.075e-01   1.103 0.270278
dribbling    -1.462e-01  1.900e-01  -0.770 0.441591
reactions     3.917e+00  1.256e-01  31.201  < 2e-16 ***
long_pass     5.147e-03  1.644e-01   0.031 0.975030
stamina      -3.483e-01  8.951e-02  -3.891 0.000101 ***
sprint_speed  4.629e-01  1.053e-01   4.395 1.13e-05 ***
agility      -1.544e-01  1.114e-01  -1.386 0.165914
short_pass    1.626e-01  2.388e-01   0.681 0.496016
finishing     3.964e-01  1.260e-01   3.145 0.001668 **
long_shots   -4.447e-01  1.286e-01  -3.459 0.000547 ***
```

ANOVA TEST:

By considering the above summary of the model let

H0: βdribbling=βcrossing=βball_control=βlong_pass=βagility=βshort_pass= 0 Vs

H1: Atleast one of βdribbling, βcrossing, βball_control, βlong_pass, βshort_pass, βagility ≠ 0

```
Analysis of Variance Table

Model 1: value ~ att_position + reactions + short_pass + sprint_speed +
    agility + heading + shot_power + finishing + long_shots +
    volleys
Model 2: value ~ dribbling + crossing + ball_control + att_position +
    reactions + short_pass + acceleration + sprint_speed + agility +
    heading + shot_power + finishing + long_shots + volleys
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1   5671 24295077
2   5667 24281312  4     13766 0.8032 0.5229
```

p-value is large => H0(restricted model) not rejected => dribbling, crossing, ball_control, long_pass,short_pass, agility do not have significant effect on the value.

3)Model with the possible features of defender:

Model: value~stamina + strength + aggression + sprint_speed + slide_tackle + stand_tackle + interceptions + reactions + crossing + sprint_speed + jumping + heading + short_pass

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -2.540e+02  8.344e+00 -30.442  < 2e-16 ***
stamina       -2.296e-01  1.009e-01  -2.276 0.022864 *
strength       8.599e-02  9.781e-02   0.879 0.379350
aggression     7.196e-03  1.017e-01   0.071 0.943604
sprint_speed   2.979e-01  8.657e-02   3.441 0.000583 ***
slide_tackle  -2.225e-01  2.123e-01  -1.048 0.294671
stand_tackle   4.487e-01  2.268e-01   1.979 0.047918 *
interceptions -5.141e-01  1.567e-01  -3.281 0.001041 **
reactions      4.171e+00  1.292e-01  32.293  < 2e-16 ***
crossing       1.274e-01  9.363e-02   1.361 0.173606
jumping       -5.082e-02  8.799e-02  -0.578 0.563624
heading       -2.218e-01  9.447e-02  -2.348 0.018920 *
short_pass     4.861e-01  1.419e-01   3.426 0.000618 ***
---
```

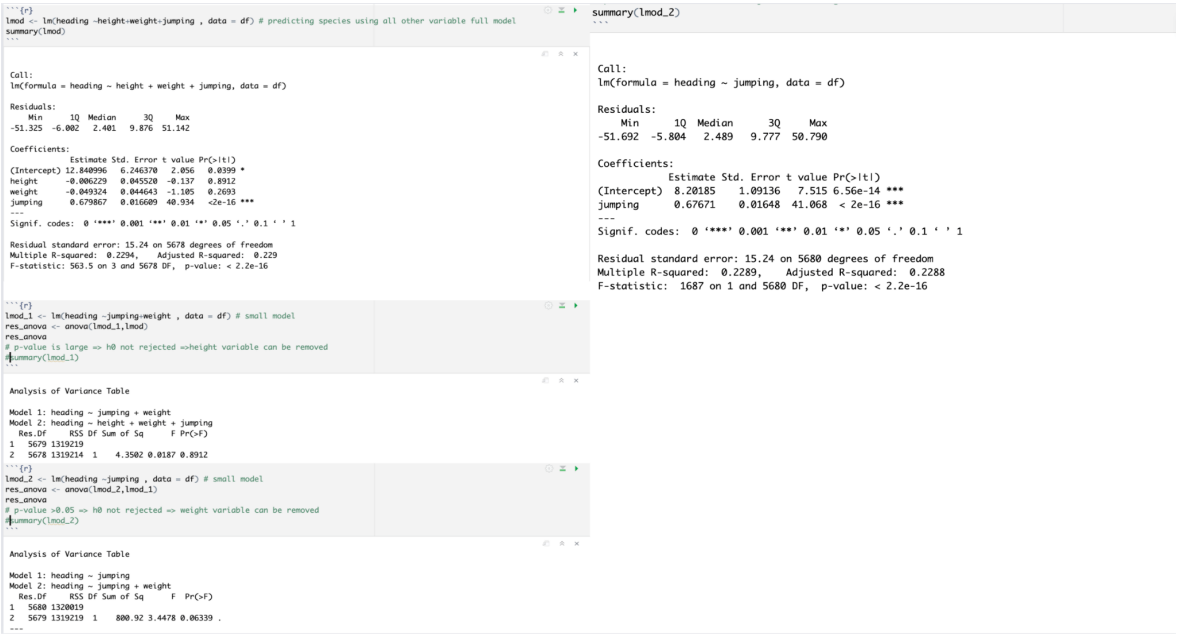
ANOVA test:
By considering the above summary of the model let
H0:βstrength=βaggression=βslide_tackle=βcrossing=βjumping= 0 Vs
H1: At Least one of βstrength, βaggression, βslide_tackle, βcrossing, βjumping ≠ 0

```
Analysis of Variance Table

Model 1: value ~ stamina + sprint_speed + stand_tackle + interceptions +
    reactions + sprint_speed + heading + short_pass
Model 2: value ~ stamina + strength + aggression + sprint_speed + slide_tackle +
    stand_tackle + interceptions + reactions + crossing + sprint_speed +
    jumping + heading + short_pass
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1   5674 24343326
2   5669 24326429  5     16896 0.7875 0.5585
```

p-values is large => H0(restricted model) not rejected =>strength, aggression, slide_tackle, crossing, jumping do not significantly affect the value.

4)Model: heading ~height, weight, jumping, let:
H0:βheight=βweight= 0 Vs H1: At Least one of βheight, βweight ≠ 0



```
Call:
lm(formula = heading ~ height + weight + jumping, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-51.325  -6.002   2.401   9.876  51.142

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.840996   6.246370   2.056   0.0399 *
height      -0.006229   0.045520  -0.137   0.8912
weight      -0.049324   0.044643  -1.105   0.2693
jumping      0.679867   0.016609  40.934   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.24 on 5678 degrees of freedom
Multiple R-squared:  0.2294,    Adjusted R-squared:  0.229
F-statistic: 563.5 on 3 and 5678 DF,  p-value: < 2.2e-16
```

```
lmod_1 <- lm(heading ~jumping+weight , data = df) # small model
res_anova <- anova(lmod_1,lmod)
res_anova
# p-value is large => h0 not rejected =>height variable can be removed
#summary(lmod_1)
```

```
Analysis of Variance Table

Model 1: heading ~ jumping + weight
Model 2: heading ~ height + weight + jumping
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1   5679 1319219
2   5678 1319214  1    4.3502 0.0187 0.8912
```

```
lmod_2 <- lm(heading ~jumping , data = df) # small model
res_anova <- anova(lmod_2,lmod_1)
res_anova
# p-value >0.05 => h0 not rejected => weight variable can be removed
#summary(lmod_2)
```

```
Analysis of Variance Table

Model 1: heading ~ jumping
Model 2: heading ~ jumping + weight
  Res.Df     RSS Df Sum of Sq      F  Pr(>F)
1   5680 1320019
2   5679 1319219  1    800.92 3.4478 0.06339 .
---
```

```
summary(lmod_2)
```

```
Call:
lm(formula = heading ~ jumping, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-51.692  -5.804   2.489   9.777  50.790

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.20185    1.09136   7.515 6.56e-14 ***
jumping      0.67671    0.01648  41.068  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.24 on 5680 degrees of freedom
Multiple R-squared:  0.2289,    Adjusted R-squared:  0.2288
F-statistic:  1687 on 1 and 5680 DF,  p-value: < 2.2e-16
```

p-value is large => H0(restricted model) not rejected =>height and weight do not have significant effect on heading.

EFFECT OF PLAYER FEATURES ON PLAYER VALUE :

feature	dribbling	vision	att_postiion	ball_control	reactions
p-value	1.109e-10	3.463e-11	0.07341	0.211	<2.2e-16
conclusion	Fail to reject H0: Dribbling not important factor	Reject H0: Vision is an important factor	Close call	Fail to reject H0: Ball_control not important factor	Reject H0: reaction is an important factor

feature	dribbling	vision	crossing	ball_control	curve	
p-value	1.109e-10	<2.2e-16	0.2278	6.754e-15	0.1143	
conclusion	Reject H0: dribbling is an important factor	Reject H0: Vision is an important factor	Fail to reject H0: Crossing not important factor	Reject H0: ball_control is an important factor	Fail to reject H0: Curve not important factor	

feature	gk_positioning	gk_diving	gk_handling	gk_kicking	gk_reflexes
p-value	0.3026	0.5526	0.5833	0.6377	0.9482
conclusion	Fail to reject H0: dribbling not important factor	Fail to reject H0: vision not important factor	Fail to reject H0: crossing not important factor	Fail to reject H0: ball_control not important factor	Fail to reject H0: Curve not important factor

feature	acceleration	sprint_speed	slide_tackle	stand_tackle	interceptions
p-value	0.06621	0.005248	3.659e-13	0.002187	1.021e-11
conclusion	Close call	Reject H0: sprint_speed is an important factor	Reject H0: Slide_tackle is an important factor	Reject H0: Stand_tackle is an important factor	Reject H0: Interceptions is an important factor

Model Improvements:

Utilising Ensembling methods:

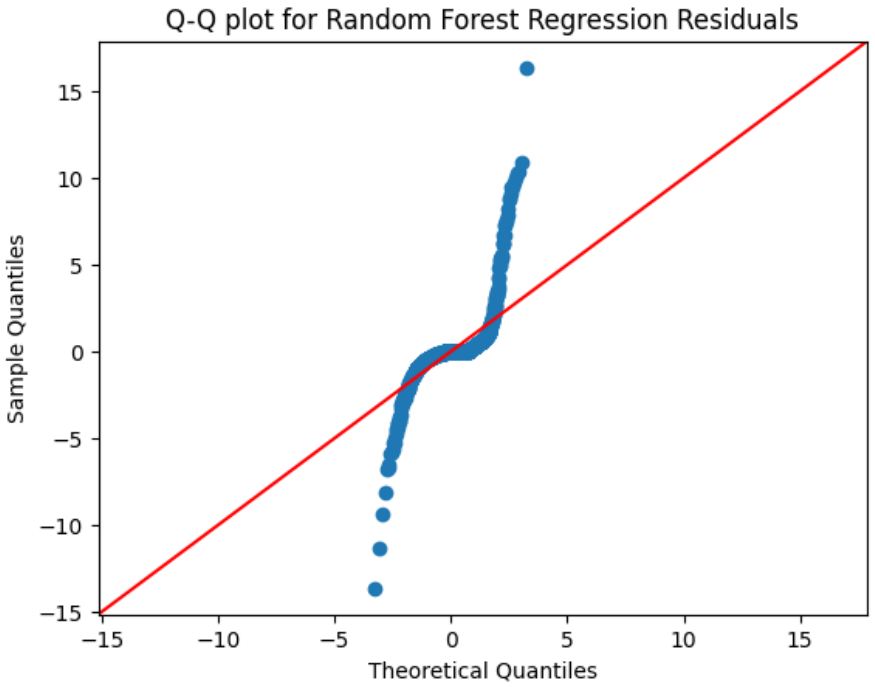
Random forest

Model Overview:

Random forests typically perform better than other models for the following reasons: Random forests solve the problem of overfitting because they combine the output of multiple decision trees to come up with a final prediction.

Model Evaluation:

- The RandomForestRegressor achieves a remarkably low RMSE (Root Mean Squared Error) of approximately 1.52, indicating strong predictive performance.
- The R-squared value of approximately 0.87 suggests that the model explains 87% of the variance in player values, showcasing a high level of explanatory power.



XGBoost:

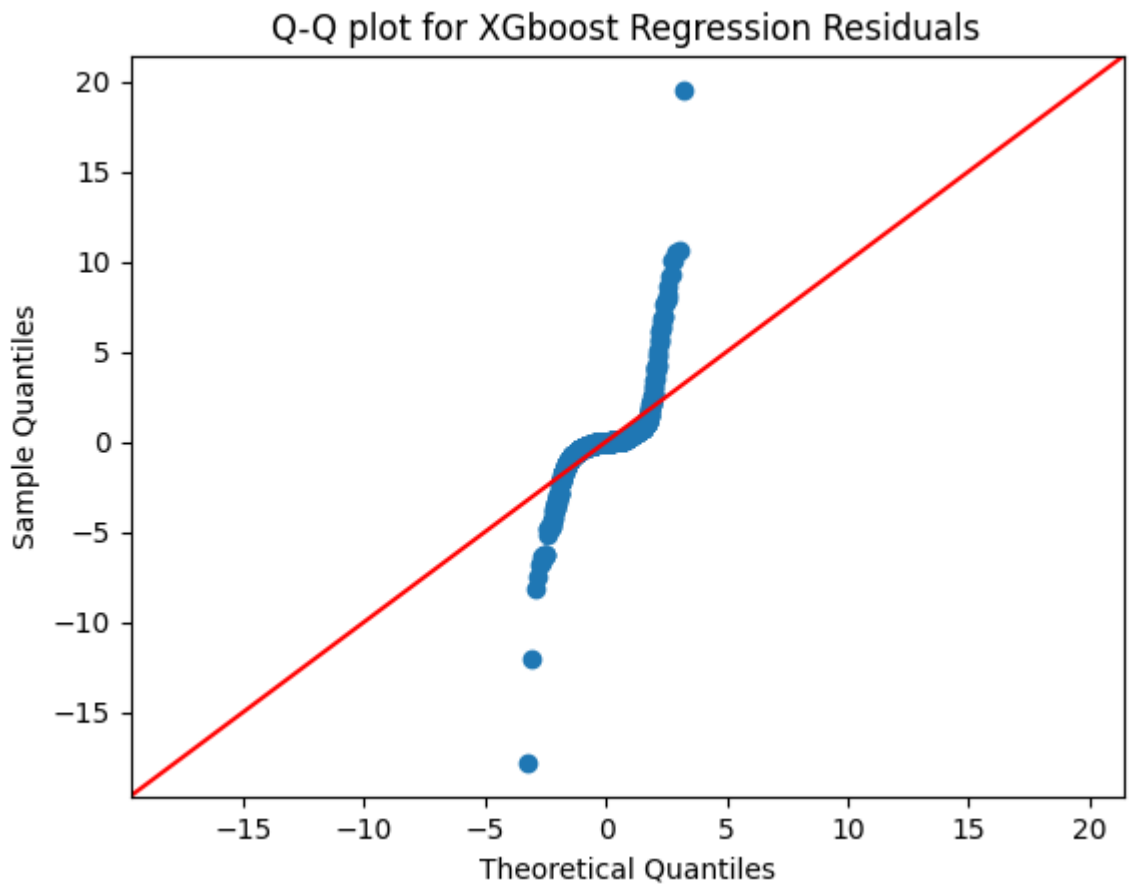
Model Overview:

XGBoost is a robust machine-learning algorithm that can help you understand your data and make better decisions. XGBoost is an implementation of gradient-boosting decision trees. Here, we use XGBoost to Optimise mean square error.

Model Evaluation:

```
xgb_rmse = np.sqrt(mean_squared_error(y_test, xgb_model.predict(X_test)))
print("xgbregressor rmse:",xgb_rmse)
print("r2_score", r2_score(y_test,xgb_model.predict(X_test)))
```

```
xgbregressor rmse: 1.5255279598790699
r2_score 0.8641022501782685
```



Feature Selection with Ridge regularisation to deal with high Collinearity:

Model Overview:

Here we are using Ridge for feature selection because it reduces multicollinearity. Using Ridge for Pipeline model evaluation.A pipeline is a linear sequence of data preparation options, modelling operations, and prediction transform operations. It allows the sequence of steps to be specified, evaluated, and used as an atomic unit. After feature selection we predicted the training data set and test data set residual plot and y vs hat{y} with root mean square error, explained variance score and R-square values.

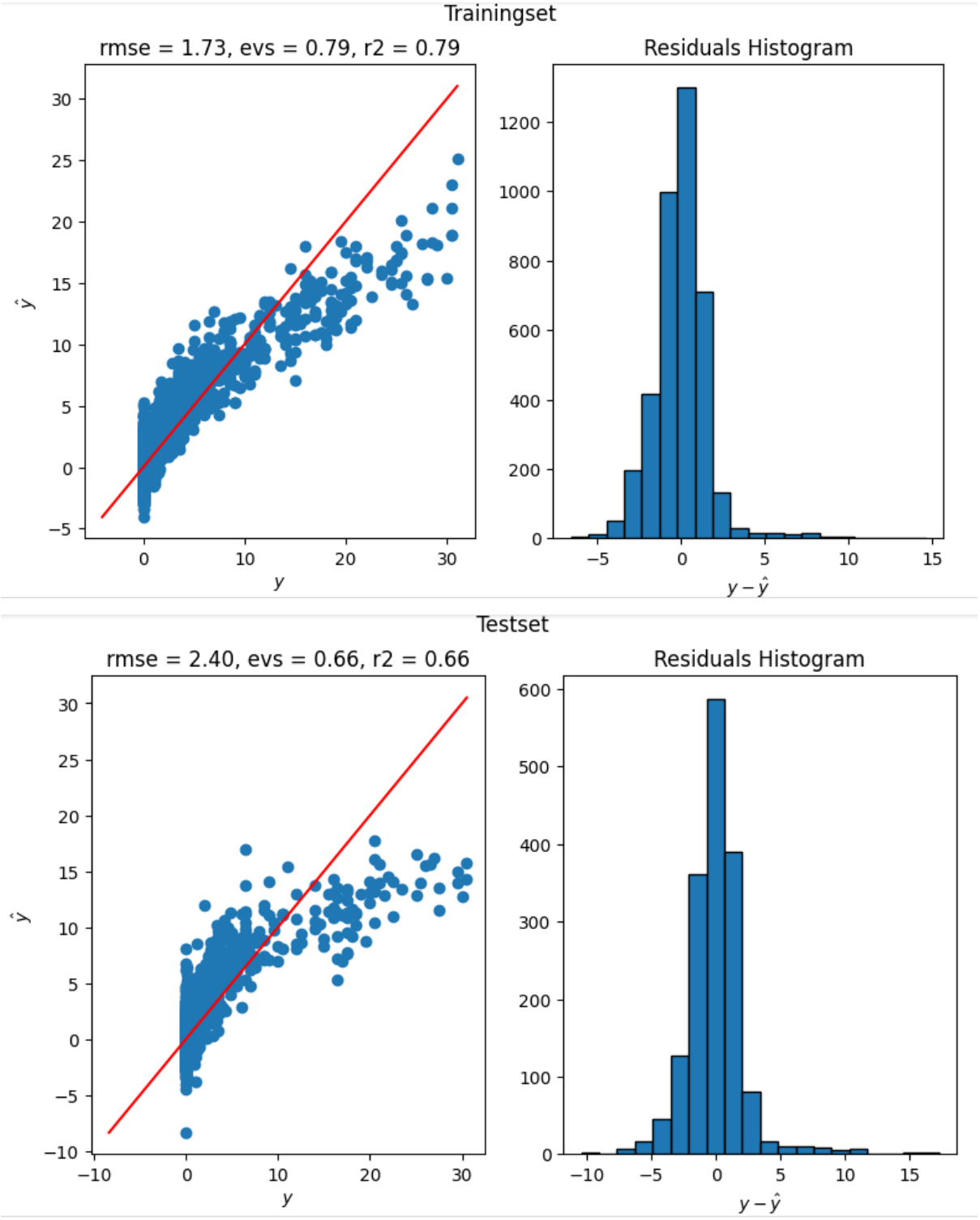
Model Evaluation:

Scatter plot (y vs Predicted y):

- This plot is a scatter diagram where the actual observed values (y) are plotted against the predicted values\ hat{y} from your regression model.
- Each point in the scatter plot represents an observation in your dataset, and its position is determined by the actual value of the target variable (y) and the predicted value hat{y}.
- The red line in the plot typically represents the line where y equals hat{y} perfectly. Deviations from this line indicate differences between the actual and predicted values.

Residuals Histogram (y - \hat{y}:

- The residuals are the differences between the actual values (y) and the predicted values\ hat{y} from the regression model. The histogram of residuals shows the distribution of these differences.
- The histogram provides insights into the distribution of errors made by the model. A residual is essentially the error of the model for each data point.



In summary, both the scatter plot of y vs hat{y} and the residuals histogram are valuable tools for understanding how well a regression model fits the data. They help identify patterns, trends, and potential issues in the model's predictions, guiding further analysis and model improvement efforts.

XGBoost using ridge feature selection:

Model Overview:

We are using the XGBoost model with ridge feature selection to reduce multicollinearity. We get root mean squared error and R- squared as follows:

Model Evaluation:

```
xgbs_rmse = np.sqrt(mean_squared_error(ys_test, xgb_sel.predict(Xs_test)))
print("xgbregressor rmse:",xgbs_rmse)
print("r2_score", r2_score(ys_test,xgb_sel.predict(Xs_test)))
```

```
xgbregressor rmse: 1.671625262274219
r2_score 0.8368264456548775
```

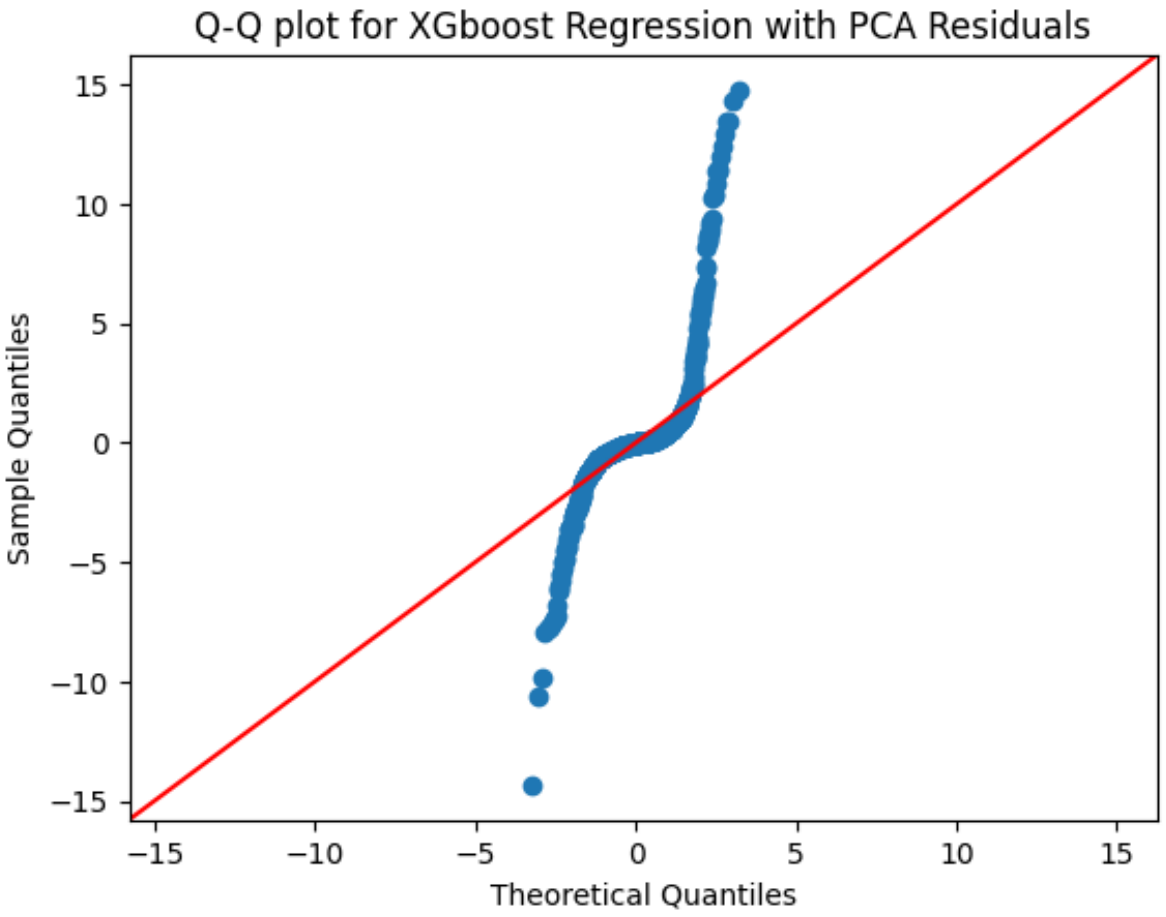
Feature selection based on the significance of attributes from hypothesis testing:

Model Overview:

XGBoost, chosen as the modelling algorithm, exhibits notable advantages over traditional linear models due to its ability to handle complex relationships and interactions. To address multicollinearity, features were meticulously selected using hypothesis testing for significance. This approach aids in mitigating collinearity issues by retaining only the most relevant features, thereby enhancing the model's interpretability.

Model Evaluation:

For the first model, the XGBoost regressor demonstrates robust predictive performance. The Root Mean Squared Error (RMSE) is calculated to be 1.634, reflecting the model's accuracy in predicting player values. The R-squared value of approximately 0.844 signifies that the model accounts for 84.4% of the variance in player values, highlighting its substantial explanatory capability. These results underscore the effectiveness of feature selection in managing multicollinearity while maintaining strong predictive accuracy.



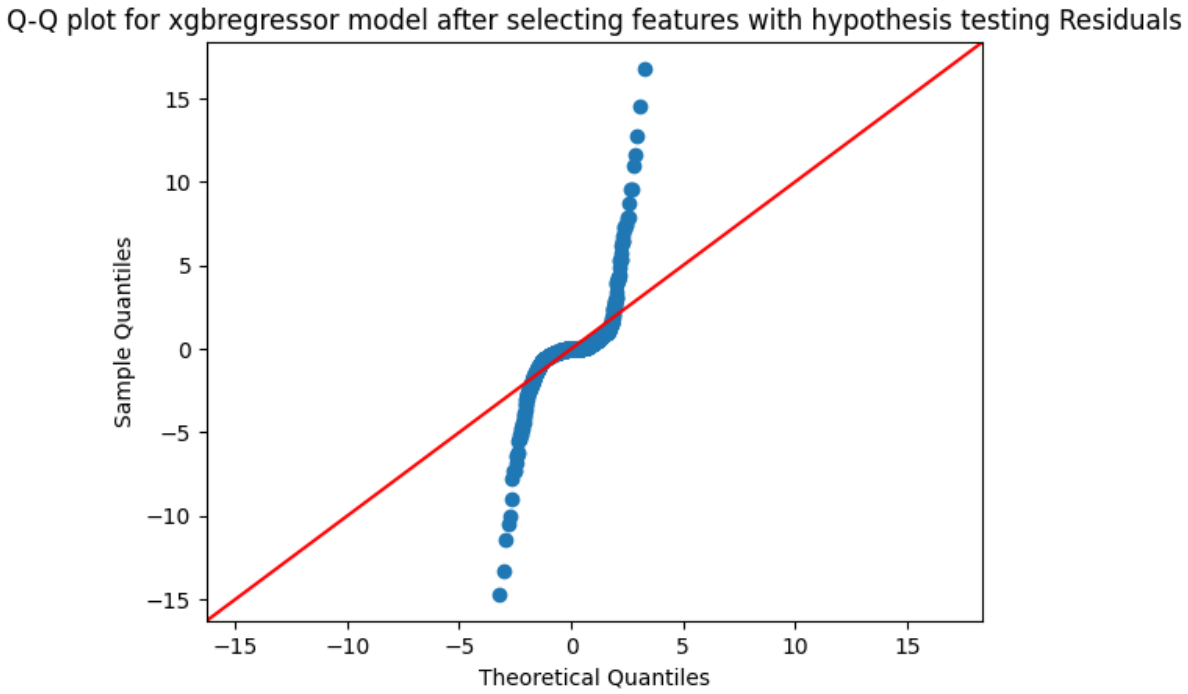
Feature reduction using Principal Component Analysis to deal with Multicollinearity:

Model Overview:

Utilising XGBoost as the chosen model, an effective strategy was implemented to address multicollinearity concerns. Feature reduction was achieved through Principal Component Analysis (PCA), condensing the original set of features into 10 components. This technique helps alleviate multicollinearity by transforming the features into a set of orthogonal variables, maintaining the essential information while minimising inter-feature correlations.

Model Evaluation:

In the second model, XGBoost, coupled with PCA for feature reduction, demonstrates commendable performance. The Root Mean Squared Error (RMSE) is calculated at 1.878, reflecting the model's accuracy in predicting player values. The R-squared value, approximately 0.794, indicates that the model accounts for 79.4% of the variance in player values, underscoring its substantial explanatory power. The successful combination of XGBoost with PCA showcases an effective approach to managing multicollinearity while maintaining competitive predictive accuracy.



CONCLUSION

Results:

Model	RMSE	R^2
Linear-Regression(scikit-learn)	3.233079991126887	0.3896136375378023
Ordinary Least Squares Estimation	3.222857346243236	0.489
OLS with log Transformation	4.723467923706853	0.825
OLS with box-cox Transformation	5.0443790758369955	0.839
Ridge Regression	3.2049928647223513	0.4001729372757731
Lasso Regression	3.2830299287585074	0.3706074384439773
Random Forest Regressor	1.5152210320565866	0.865932379285945
XGboost Regressor	1.5255279598790699	0.8641022501782685
XGboost with Feature Selection	1.671625262274219	0.8368264456548775
XGboost with PCA	1.8776395523021796	0.7941284319905643
XGboost with features from hypothesis testing	1.634440755713893	0.8440051400707622

Our project explored various regression models to predict football player values. Traditional linear regression methods, including Ordinary Least Squares (OLS) and transformations, offered limited performance. Regularised techniques like Ridge and Lasso showed marginal improvements. The Random Forest Regressor excelled with an impressive RMSE of 1.52 and an R-squared value of 0.87. The XGBoost Regressor further enhanced performance with an RMSE of 1.53 and an R-squared value of 0.86.

Feature selection techniques, especially with XGBoost and hypothesis testing, proved effective in maintaining accuracy while improving interpretability. However, addressing multicollinearity with Principal Component Analysis (PCA) and XGBoost resulted in a trade-off, reducing features to 10 principal components but slightly lowering the R-squared value to 0.79.

In conclusion, Random Forest and XGBoost Regressors are robust choices that outperform traditional linear methods. Careful consideration of feature selection techniques is crucial, balancing simplicity and explanatory power.

References:

1) https://www.kaggle.com/datasets/rehandl23/fifa-24-player-stats-dataset/code
2) https://www.fifaindex.com/players/
3) https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/
4) https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/