Exploring the Link Between Physical Attributes and Player Prices in FC24

Name, akyrkjeeide@bellarmine.edu

I. INTRODUCTION

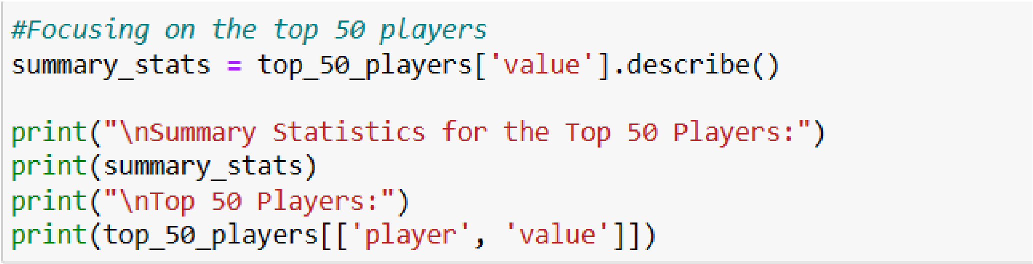
I discovered the FC24 Players Stats dataset particularly intriguing, given my personal engagement with the game. It's interesting to have the ability to delve into various correlations between different variables and see if there are any correlations. Moreover, the dataset encompasses abundant categorical data, providing valuable insights for in-depth analyses of diverse coefficients and regression aspects.

II. BACKGROUND

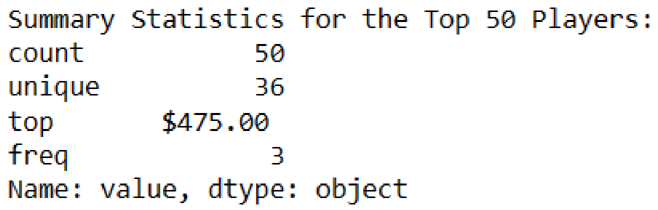
The FIFA Football Players Dataset presents an extensive compilation of details concerning football players globally. With a plethora of attributes for each player, this dataset proves invaluable for diverse analyses and gaining insights into the world of football. It serves as a valuable resource for gaming enthusiasts and real-world sports fans alike.

III. EXPLORATORY ANALYSIS

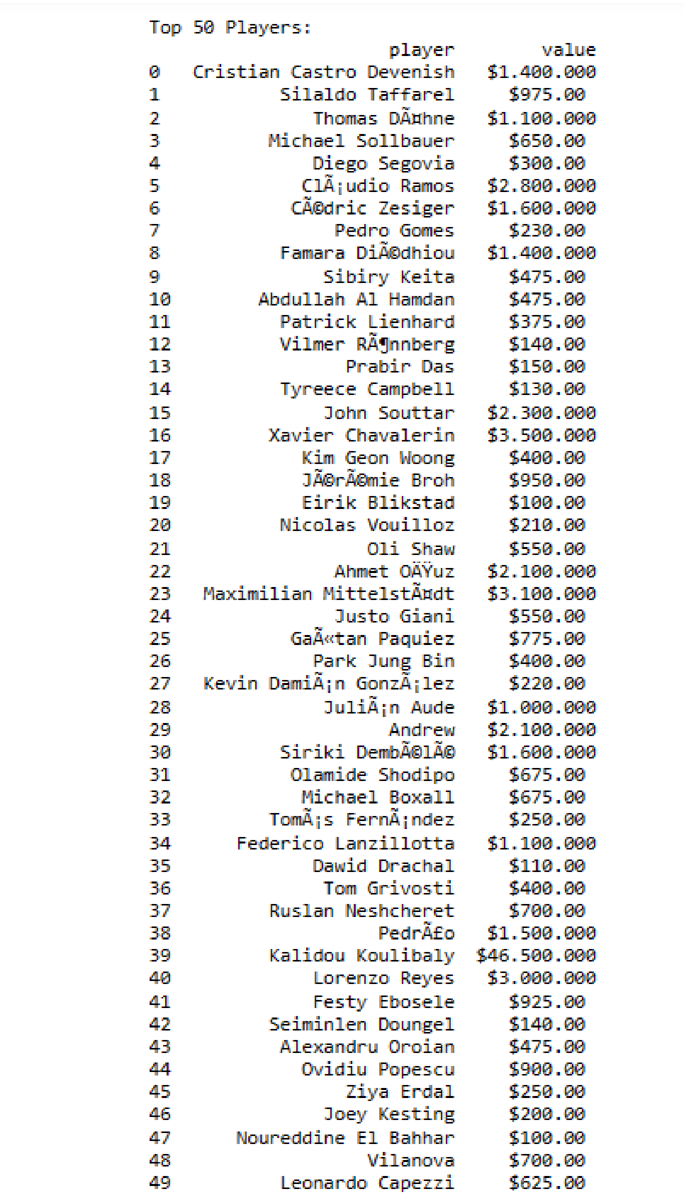
Summary of the top 50 players:



```python
#Focusing on the top 50 players
summary_stats = top_50_players['value'].describe()

print("\nSummary Statistics for the Top 50 Players:")
print(summary_stats)
print("\nTop 50 Players:")
print(top_50_players[['player', 'value']])
```

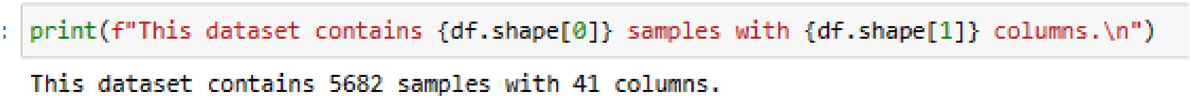


```
Summary Statistics for the Top 50 Players:
count          50
unique         36
top       $475.00
freq            3
Name: value, dtype: object
```



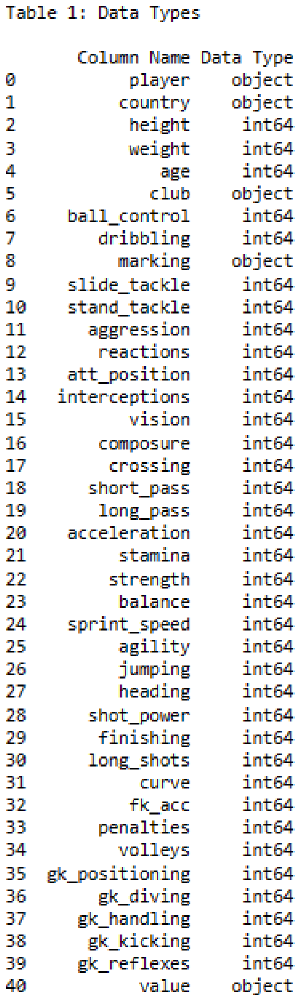
```
Top 50 Players:
                      player        value
0   Cristian Castro Devenish   $1.400.000
1           Silaldo Taffarel     $975.00
2              Thomas DÃ¤hne   $1.100.000
3          Michael Sollbauer     $650.00
4              Diego Segovia     $300.00
5             ClÃ¡udio Ramos   $2.800.000
6            CÃ©dric Zesiger   $1.600.000
7                Pedro Gomes     $230.00
8           Famara DiÃ©dhiou   $1.400.000
9               Sibiry Keita     $475.00
10        Abdullah Al Hamdan     $475.00
11          Patrick Lienhard     $375.00
12          Vilmer RÃ¶nnberg     $140.00
13                Prabir Das     $150.00
14          Tyreece Campbell     $130.00
15              John Souttar   $2.300.000
16         Xavier Chavalerin   $3.500.000
17            Kim Geon Woong     $400.00
18            JÃ©rÃ©mie Broh     $950.00
19            Eirik Blikstad     $100.00
20          Nicolas Vouilloz     $210.00
21                  Oli Shaw     $550.00
22               Ahmet OÄŸuz   $2.100.000
23   Maximilian MittelstÃ¤dt   $3.100.000
24               Justo Giani     $550.00
25           GaÃ«tan Paquiez     $775.00
26             Park Jung Bin     $400.00
27   Kevin DamiÃ¡n GonzÃ¡lez     $220.00
28              JuliÃ¡n Aude   $1.000.000
29                    Andrew   $2.100.000
30          Siriki DembÃ©lÃ©   $1.600.000
31           Olamide Shodipo     $675.00
32            Michael Boxall     $675.00
33         TomÃ¡s FernÃ¡ndez     $250.00
34      Federico Lanzillotta   $1.100.000
35             Dawid Drachal     $110.00
36              Tom Grivosti     $400.00
37         Ruslan Neshcheret     $700.00
38                   PedrÃ£o   $1.500.000
39         Kalidou Koulibaly  $46.500.000
40             Lorenzo Reyes   $3.000.000
41             Festy Ebosele     $925.00
42         Seiminlen Doungel     $140.00
43          Alexandru Oroian     $475.00
44            Ovidiu Popescu     $900.00
45                Ziya Erdal     $250.00
46              Joey Kesting     $200.00
47      Noureddine El Bahhar     $100.00
48                  Vilanova     $700.00
49          Leonardo Capezzi     $625.00
```

Table 1: Data Types



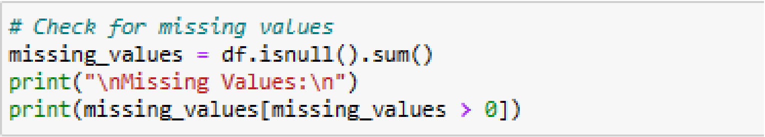
```
print(f"This dataset contains {df.shape[0]} samples with {df.shape[1]} columns.\n")
```

```
This dataset contains 5682 samples with 41 columns.
```

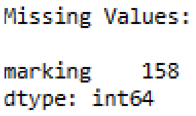


```
Table 1: Data Types

      Column Name Data Type
0          player    object
1         country    object
2          height     int64
3          weight     int64
4             age     int64
5            club    object
6    ball_control     int64
7       dribbling     int64
8         marking    object
9     slide_tackle     int64
10    stand_tackle     int64
11      aggression     int64
12       reactions     int64
13    att_position     int64
14   interceptions     int64
15          vision     int64
16       composure     int64
17        crossing     int64
18      short_pass     int64
19       long_pass     int64
20    acceleration     int64
21         stamina     int64
22        strength     int64
23         balance     int64
24    sprint_speed     int64
25         agility     int64
26         jumping     int64
27         heading     int64
28      shot_power     int64
29       finishing     int64
30      long_shots     int64
31           curve     int64
32          fk_acc     int64
33       penalties     int64
34         volleys     int64
35  gk_positioning     int64
36       gk_diving     int64
37     gk_handling     int64
38      gk_kicking     int64
39     gk_reflexes     int64
40           value    object
```

Checking for missing values:



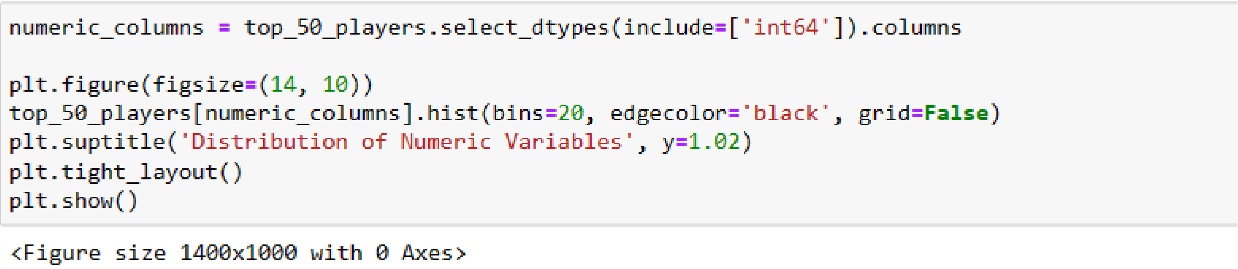
```
# Check for missing values
missing_values = df.isnull().sum()
print("\nMissing Values:\n")
print(missing_values[missing_values > 0])
```



```
Missing Values:

marking    158
dtype: int64
```

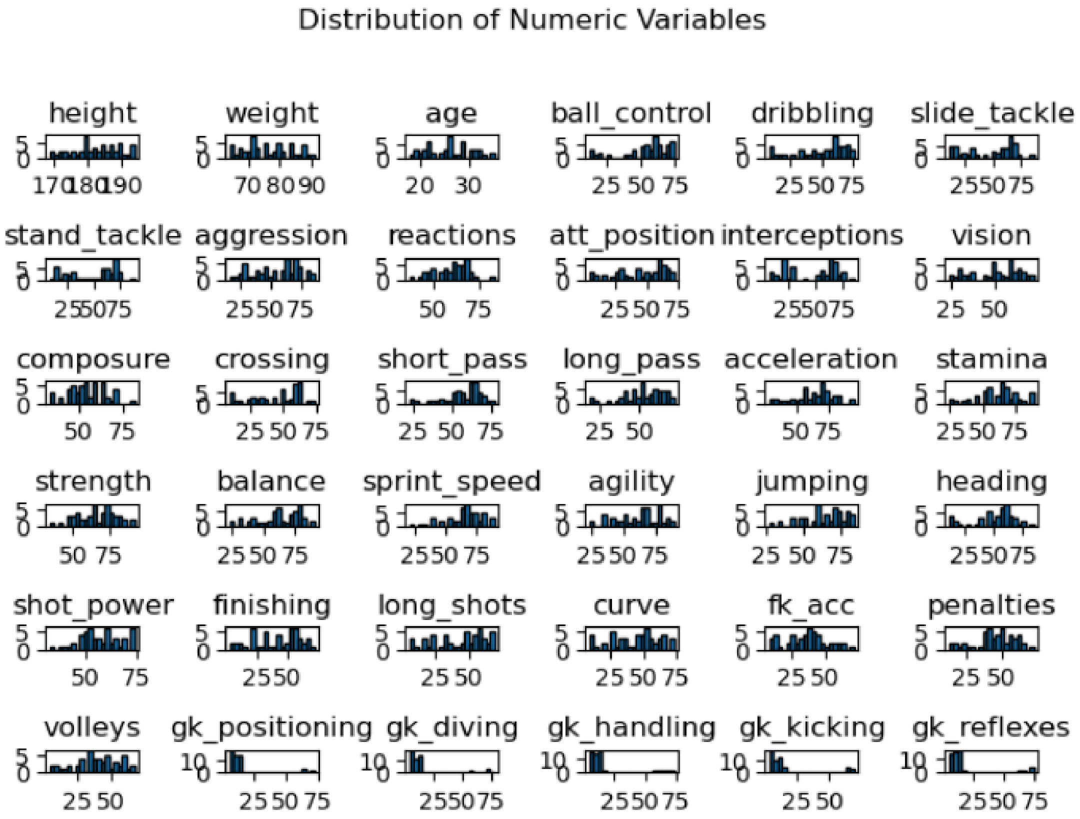
Plotting unusual statistics or distributions:

By consolidating the distribution of every numerical variable into an overall representation, I've created a holistic view that facilitates a comprehensive examination of the dataset. This consolidated approach is particularly beneficial for discerning potential outliers, identifying skewed distributions, and gaining a broader understanding of the overall dataset dynamics. This method allows for more efficient detection of any data points that might be misleading or unusual, offering a concise yet insightful overview of the entire numerical feature space:

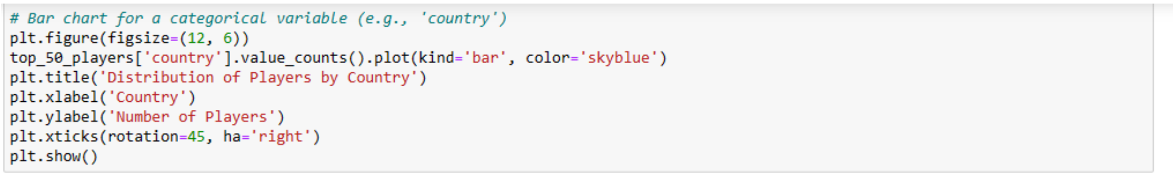


```
numeric_columns = top_50_players.select_dtypes(include=['int64']).columns

plt.figure(figsize=(14, 10))
top_50_players[numeric_columns].hist(bins=20, edgecolor='black', grid=False)
plt.suptitle('Distribution of Numeric Variables', y=1.02)
plt.tight_layout()
plt.show()
```

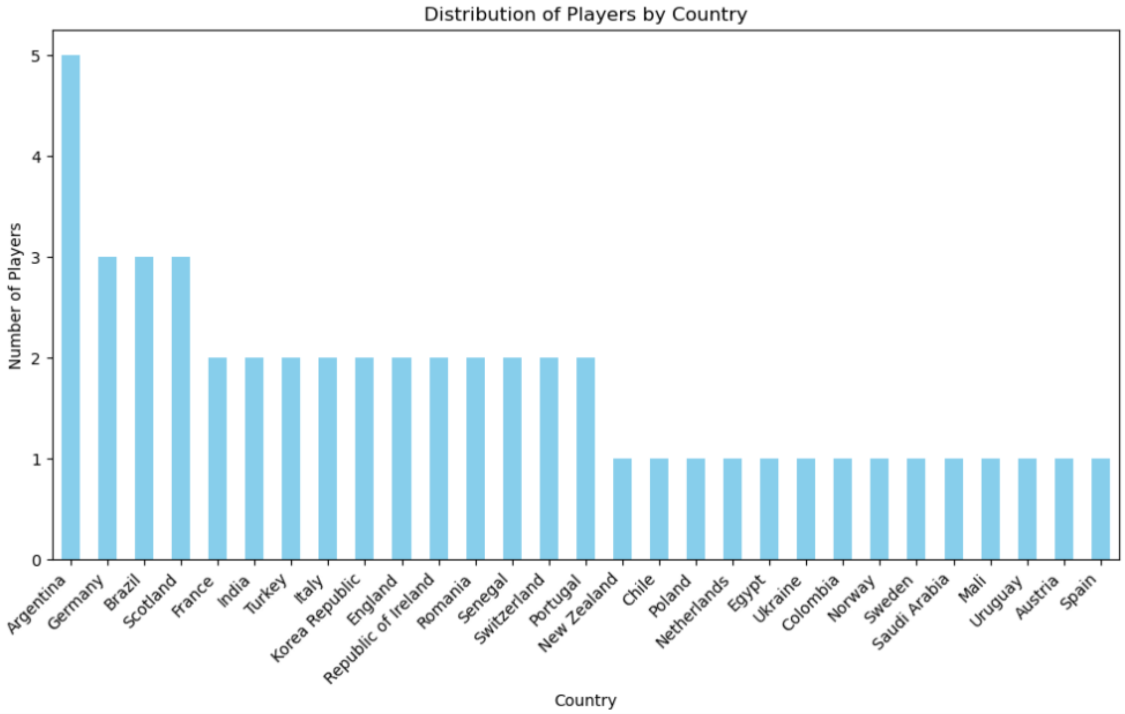
```
<Figure size 1400x1000 with 0 Axes>
```



Analyzing key player attributes—such as ball control, dribbling, aggression, short and long passes, sprint speed, and stamina—reveals a noteworthy trend. These attributes display left-skewed distributions, indicating that, on average, a significant proportion of players possess lower values in these skills. In contrast, a smaller subset of players demonstrates exceptional proficiency with higher attribute values. This statistical asymmetry prompts further investigation into the concentration of players with moderate to lower skill levels, inviting consideration of potential implications for player performance and team dynamics.



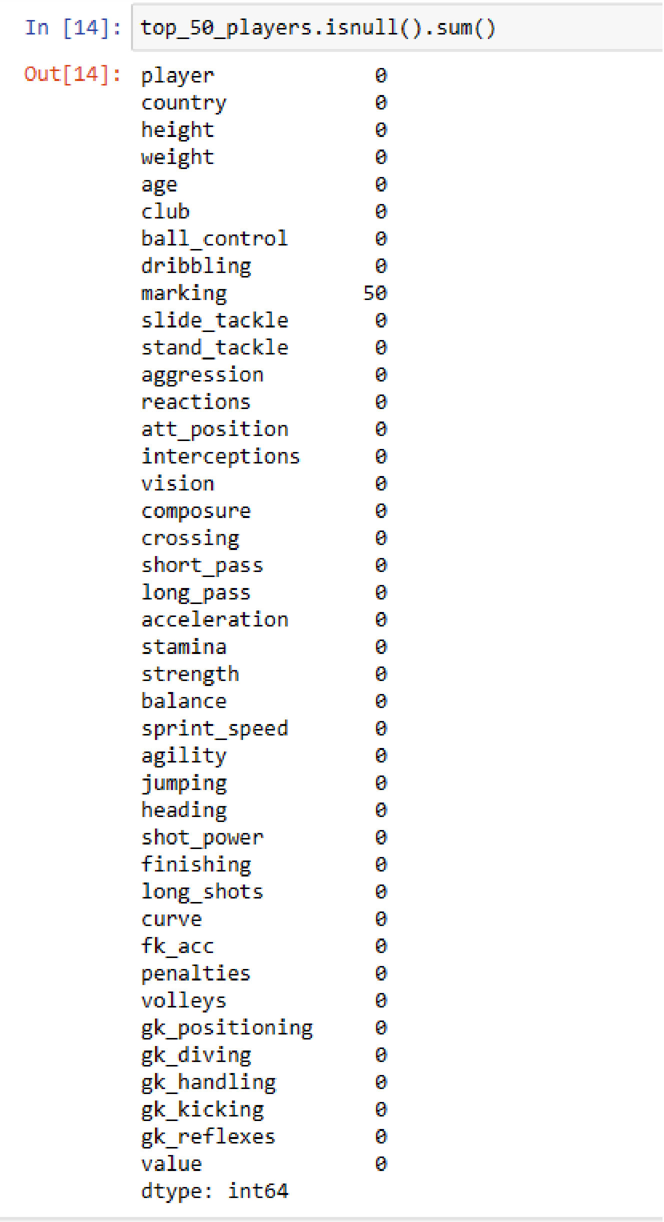
```
# Bar chart for a categorical variable (e.g., 'country')
plt.figure(figsize=(12, 6))
top_50_players['country'].value_counts().plot(kind='bar', color='skyblue')
plt.title('Distribution of Players by Country')
plt.xlabel('Country')
plt.ylabel('Number of Players')
plt.xticks(rotation=45, ha='right')
plt.show()
```



Exploring the origin of players within the top 50 was intriguing, especially when examining the distribution across countries. The data reveals a distinct right skew, indicating that a few countries contribute significantly more players to the top 50 compared to others. The right skewness suggests a concentration of players from specific countries with higher representation, potentially implying that these nations possess a more substantial talent pool or are more dominant in the realm of football; however, this would require further investigation into the countries contributing the most players providing insights into global football dynamics, talent development, and regional football prowess.

IV. METHODS

A. Data Preparation



```
In [14]: top_50_players.isnull().sum()
```

```
Out[14]: player             0
         country            0
         height             0
         weight             0
         age                0
         club               0
         ball_control       0
         dribbling          0
         marking           50
         slide_tackle       0
         stand_tackle       0
         aggression         0
         reactions          0
         att_position       0
         interceptions      0
         vision             0
         composure          0
         crossing           0
         short_pass         0
         long_pass          0
         acceleration       0
         stamina            0
         strength           0
         balance            0
         sprint_speed       0
         agility            0
         jumping            0
         heading            0
         shot_power         0
         finishing          0
         long_shots         0
         curve              0
         fk_acc             0
         penalties          0
         volleys            0
         gk_positioning     0
         gk_diving          0
         gk_handling        0
         gk_kicking         0
         gk_reflexes        0
         value              0
         dtype: int64
```

To evaluate necessary data preparations, the code top_50_players.isnull().sum() is checking for missing values in each column of the DataFrame top_50_players and then calculating the sum of these missing values for each column. It provides a quick summary of the count of missing values in each column. This information is crucial for data preparation before linear regression for the following reasons:

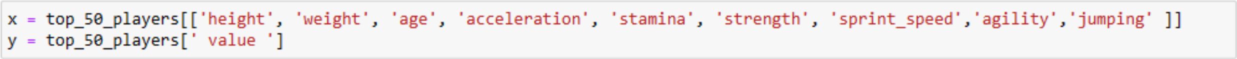
1. **Data Quality Assessment:** Identifying missing values helps assess the quality of the dataset. If a significant number of values are missing in a particular column, it might impact the reliability of the linear regression model.
2. **Handling Missing Values:** Understanding which columns have missing values allows for informed decisions on how to handle them. Techniques such as imputation or removal of rows/columns with missing values can be applied based on the context of the data.
3. **Input Validity:** Linear regression assumes that the input features are complete and valid. Checking for missing values ensures that the variables intended for regression are appropriately populated.

From looking at the output one can see that the data looks prepared except for marking, which would need to be cleaned; however, since I will not be utilizing this variable in my analysis the data is ready.

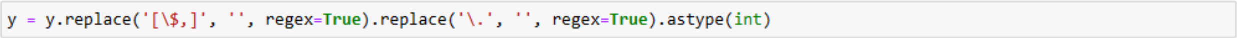
B. Experimental Design

Creating independent and dependent variables to predict

Physical Attributes Predicting Performance:



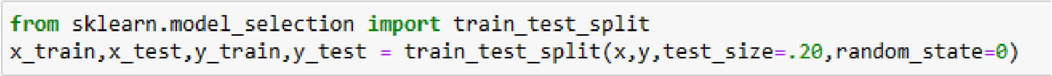
```
x = top_50_players[['height', 'weight', 'age', 'acceleration', 'stamina', 'strength', 'sprint_speed','agility','jumping' ]]
y = top_50_players[' value ']
```



```
y = y.replace('[\$,]', '', regex=True).replace('\.', '', regex=True).astype(int)
```

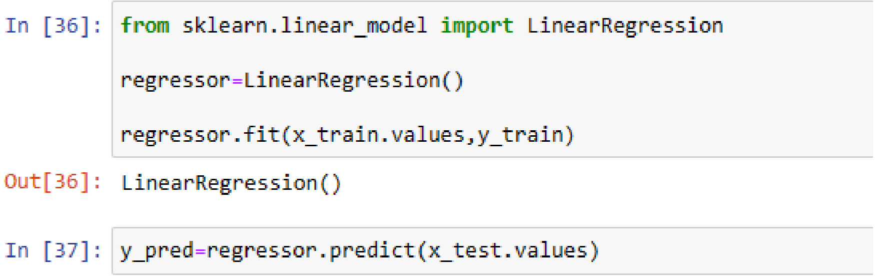
Creating ‘value’ to numeric data (removing dollar sign etc.)

Splitting the dataset into Training set and Test set (20% test size)



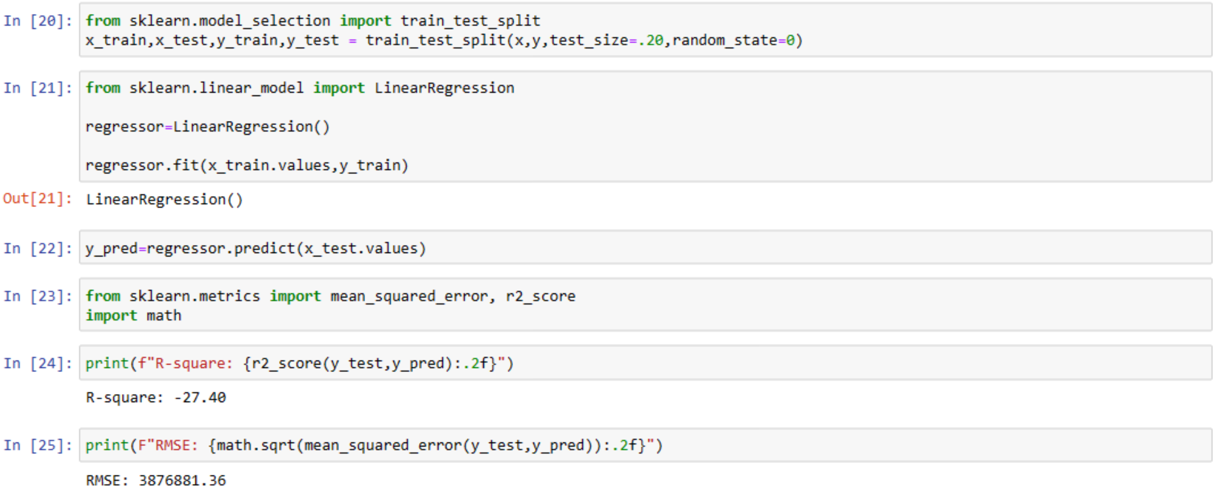
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.20,random_state=0)
```

Training the Linear Regression model



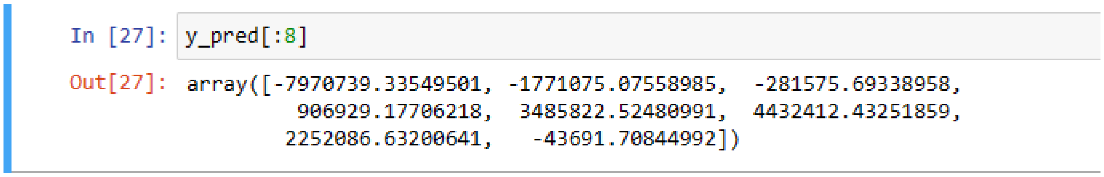
```
In [36]: from sklearn.linear_model import LinearRegression

         regressor=LinearRegression()

         regressor.fit(x_train.values,y_train)

Out[36]: LinearRegression()

In [37]: y_pred=regressor.predict(x_test.values)
```

Results



```
In [20]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.20,random_state=0)

In [21]: from sklearn.linear_model import LinearRegression

         regressor=LinearRegression()

         regressor.fit(x_train.values,y_train)

Out[21]: LinearRegression()

In [22]: y_pred=regressor.predict(x_test.values)

In [23]: from sklearn.metrics import mean_squared_error, r2_score
         import math

In [24]: print(f"R-square: {r2_score(y_test,y_pred):.2f}")

         R-square: -27.40

In [25]: print(F"RMSE: {math.sqrt(mean_squared_error(y_test,y_pred)):.2f}")

         RMSE: 3876881.36
```

Prediction



```
In [27]: y_pred[:8]

Out[27]: array([-7970739.33549501, -1771075.07558985,  -281575.69338958,
                  906929.17706218,  3485822.52480991,  4432412.43251859,
                 2252086.63200641,   -43691.70844992])
```

C. *Tools Used*

For this analysis, I employed Python v3.5.2 within the Anaconda 4.3.22 environment on an HP computer. These tools were chosen for their versatility, ease of use, and strong community support. The accompanying libraries, including Pandas, NumPy, Matplotlib, Seaborn and Scikit-Learn selected for their effectiveness in data manipulation, analysis, and machine learning tasks. The Anaconda environment facilitated seamless integration and package management, ensuring a cohesive and efficient workflow. The choice of these tools was driven by the previous teachings of utilization shown by Prof. Sarkar throughout the course, offering a robust foundation for successful data preparation and model implementation.

V. RESULTS

A. *Classification Measures/ Accuracy measure*

Test size 20%
R squared value: -27.40
RMSE: 3876881.36

Test size 15%
R squared value: -41.95
RMSE: 3742037.54

Test size 10%
R squared value: -72.90
RMSE: 4184662.27

B. *Discussion of Results*

R-squared (Rsquared) Value:

The R-squared value represents the proportion of the variance in the dependent variable (player value) that is explained by the independent variables (physical attributes). A negative R-squared suggests that the model is performing worse than a simple average prediction. It might indicate that the chosen independent variables of physical attributes do not have a linear relationship with the dependent variable, value, or that the model is not suitable for the data.

Root Mean Squared Error (RMSE):

RMSE measures the average magnitude of the errors between predicted and actual values. The high RMSE value for each test size indicates substantial discrepancies between predicted and actual values. It suggests that the model's predictions deviate significantly from the actual player values.

Predicted Values (y_pred[:8]):

The predicted values for the first eight instances show a wide range, with some negative values. Negative values in the context of player values are not meaningful, and this suggests that the model might be making unrealistic predictions. It does the same for every test size decrease.

Interpretation and Possible Issues:

The negative R-squared and large RMSE indicate that the linear regression model, using the selected physical attributes, is not effectively capturing the variation in player values and as we are decreasing the test size the R squared value increases. It might be that the relationship between these attributes and player value is not adequately modelled by a linear equation. Non-linear relationships or interactions between attributes could be at play.

The negative predicted values raise concerns about the model's appropriateness for the data. Negative values for player values are not realistic and could signal issues such as overfitting, outliers, or an inadequate choice of features. When evaluating the distribution of each attributes there was a lot of skewness which is reflecting the model fit.

C. Problems Encountered & Limitations of Implementation

During the course of this project, I encountered several challenges that required attention. Firstly, I faced difficulties in converting my dependent variable, "value," into a numerical format by eliminating symbols like dollar signs and commas. This caused errors during the training and testing phases of the linear regression model, leading to confusion in interpreting the R-squared results.

Moreover, I realized that my initial enthusiasm for the dataset, fueled by my personal engagement with the FC24 game, might have overshadowed a critical consideration – the regression logic inherent in the game. While FIFA assigns attribute values to players, this process includes an element of randomness, particularly based on player positions. For example, a highly valued defender like Kalidou Koulibaly ($46,500) might exhibit lower values in physical attributes such as speed, acceleration, and jumping, yet command a high price due to exceptional defensive performance. In the game's logic, high-paced defenders are intentionally rare to maintain a balance between defensive and offensive capabilities.

I believe this inherent game design strategy is a primary factor contributing to the poor fit and predictions observed in the linear regression model. The model struggles to capture the nuanced valuation system of FC24, where certain attributes hold more weight in determining a player's market value, leading to discrepancies in the predicted outcomes.

D. Improvements/Future Work

I suspect there could be connections between certain variables in the FC24 video game dataset. To enhance the model, I'm considering revisiting feature selection, exploring alternative attributes, or combinations that might better explain player prices. Additionally, expanding the dataset is on the table for a more robust analysis. Though I anticipate the linear regression model's R-squared value might still not be ideal, I expect improvement from the current results.

VI. CONCLUSION

In conclusion, the classification measures, particularly the R-squared values and Root Mean Squared Error (RMSE), reveal significant challenges in the linear regression model's performance. The negative R-squared values across varying test sizes indicate a poor fit, suggesting that the chosen physical attributes do not adequately explain the variation in player values. The escalating R-squared negativity with decreasing test size implies an increasing disparity between predictions and actual values.

The high RMSE values underscore substantial discrepancies in predictions, indicating a significant deviation from actual player values. Negative predicted values further raise concerns about the model's realism and suitability for the data, hinting at potential issues like overfitting, outliers, or inappropriate feature selection.

Encountered challenges, such as difficulties in converting the dependent variable and the game's inherent regression logic, highlight limitations in the dataset's representativeness. Future work involves exploring alternative attributes, revisiting feature selection, and expanding the dataset for a more comprehensive analysis, aiming for an improved linear regression model fit.