TITLE

Exploring the Relationship Between Soccer Player Wages and Performance Attributes: A Study Using FIFA 19 Player Dataset

ABSTRACT

This project aims to explore the relationship between player attributes and wages in the FIFA 19 videogame. We used a dataset of over 18,000 players and conducted statistical tests such as z-test and chi-square test to analyze the data. Our findings indicate that there are significant differences in wages between different positions and skill levels. These results can be used by soccer clubs to make informed decisions when it comes to player recruitment and salary negotiations.

INTRODUCTION

The world of soccer has grown significantly over the past few decades, with more than 250 million players in over 200 countries. With such a vast number of players, it has become increasingly important to analyze data related to player attributes, performances, and salaries. The purpose of this project is to investigate the differences in player performance and salaries based on various attributes such as footedness and player position. We will explore different aspects of the data using statistical methods to draw conclusions and provide insights for decision-makers in the soccer industry.

We will compare the mean overall rating of left-footed players and right-footed players using a z-test to determine if there is a significant difference in performance between these two groups. Then, we will perform an ANOVA to test if there is a significant difference in the mean overall rating across different player positions. This analysis could help teams identify which positions they should focus on when recruiting or training players. Finally, we will perform the analysis of categorical data using the chi-square test of independence to determine if there is any association between player position and salary. This analysis could be used to identify any disparities in salaries across different player positions and take steps to address them.

The results of this project could be useful for soccer teams, agents, and other decision-makers in the industry. By gaining insights into player performance and salaries, teams can make more informed decisions about which players to acquire and how much to pay them. Moreover, these insights could help identify any areas where improvements can be made to create a more fair and equitable industry for all players.

DATA DESCRIPTION

The FIFA 19 Player Dataset is a comprehensive collection of data on professional football players, including their personal information, physical attributes, performance statistics, and market value. This dataset is compiled from EA Sports' popular video game FIFA 19, which features accurate and detailed information on thousands of players from around the world. The link to the dataset is the following- https://www.kaggle.com/datasets/chaitanyahivlekar/fifa-19-player-dataset

The dataset contains information on over 18,000 players, including their names, ages, nationalities, club teams, positions, and overall ratings. Each player is assigned a unique identifier, which is used to track their performance and market value over time. The dataset also includes detailed physical attributes such as height, weight, preferred foot, and skill moves, as well as more subjective attributes like work rate, weak foot ability, and international reputation. In addition to personal and physical attributes, the dataset contains a wide range of performance statistics for each player, including their goals, assists, appearances, and clean sheets. These statistics are broken down by season, allowing for detailed analysis of a player's performance over time. The dataset also includes information on a player's value and wage, which are key indicators of their marketability and earning potential.

One of the key features of this dataset is its extensive coverage of players from around the world. The dataset includes players from over 50 countries, representing a diverse range of football cultures and styles of play. This makes it an ideal resource for studying global trends in football, such as the rise of South American players in European leagues or the dominance of European teams in international competitions.

Another important feature of the dataset is its ability to track changes in player performance and market value over time. By analyzing trends in player statistics and market value, researchers can gain insights into the factors that influence player success and identify potential opportunities for investment and development.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
from scipy.stats import f_oneway
from scipy.stats import f
from scipy.stats import chi2_contingency
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import Lasso
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```python
from sklearn.preprocessing import PolynomialFeatures



fifa19 = pd.read_csv(r"C:\Users\Salman\Desktop\MA 541\Project\FIFA19.csv")

print(fifa19.head())

print(fifa19.info())

print(fifa19.describe())
```

```
   Unnamed: 0              Name  Age Nationality  Overall  Potential  \
0           0          L. Messi   31   Argentina       94         94   
1           1  Cristiano Ronaldo   33    Portugal       94         94   
2           2         Neymar Jr   26      Brazil       92         93   
3           3            De Gea   27       Spain       91         93   
4           4      K. De Bruyne   27     Belgium       91         92   

                  Club    Value    Wage Preferred Foot  ...  StandingTackle  \
0         FC Barcelona  €110.5M  565000           Left  ...              28   
1             Juventus     €77M  405000          Right  ...              31   
2  Paris Saint-Germain  €118.5M  290000          Right  ...              24   
3    Manchester United     €72M  260000          Right  ...              21   
4      Manchester City    €102M  355000          Right  ...              58   

   SlidingTackle  GKDiving GKHandling GKKicking  GKPositioning GKReflexes  \
0             26         6         11        15             14          8   
1             23         7         11        15             14         11   
2             33         9          9        15             15         11   
3             13        90         85        87             88         94   
4             51        15         13         5             10         13   

  Release Clause             League           Speciality  
0        €226.5M   LALIGA SANTANDER     Complete Forward  
1        €127.1M       SERIE A TIM     Distance Shooter  
2        €228.1M  LIGUE 1 CONFORAMA     Complete Forward  
3        €138.6M     PREMIER LEAGUE           Goalkeeper  
4        €196.4M     PREMIER LEAGUE  Complete Midfielder  

[5 rows x 58 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18147 entries, 0 to 18146
Data columns (total 58 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Unnamed: 0                18147 non-null  int64 
 1   Name                      18147 non-null  object
 2   Age                       18147 non-null  int64 
 3   Nationality               18147 non-null  object
 4   Overall                   18147 non-null  int64 
 5   Potential                 18147 non-null  int64 
 6   Club                      18147 non-null  object
 7   Value                     18147 non-null  object
 8   Wage                      18147 non-null  int64 
 9   Preferred Foot            18147 non-null  object
 10  International Reputation  18147 non-null  int64 
 11  Weak Foot                 18147 non-null  int64 
 12  Skill Moves               18147 non-null  int64 
 13  Work Rate                 18147 non-null  object
 14  Position                  18147 non-null  object
 15  Jersey Number             18147 non-null  int64 
 16  Joined                    16654 non-null  object
 17  Loaned From               18147 non-null  object
 18  Contract Valid Until      18147 non-null  object
 19  Height                    18147 non-null  object
 20  Weight                    18147 non-null  object
 21  Crossing                  18147 non-null  int64 
 22  Finishing                 18147 non-null  int64 
 23  HeadingAccuracy           18147 non-null  int64 
 24  ShortPassing              18147 non-null  int64 
 25  Volleys                   18147 non-null  int64 
```

```
 26  Dribbling                 18147 non-null  int64 
 27  Curve                     18147 non-null  int64 
 28  FKAccuracy                18147 non-null  int64 
 29  LongPassing               18147 non-null  int64 
 30  BallControl               18147 non-null  int64 
 31  Acceleration              18147 non-null  int64 
 32  SprintSpeed               18147 non-null  int64 
 33  Agility                   18147 non-null  int64 
 34  Reactions                 18147 non-null  int64 
 35  Balance                   18147 non-null  int64 
 36  ShotPower                 18147 non-null  int64 
 37  Jumping                   18147 non-null  int64 
 38  Stamina                   18147 non-null  int64 
 39  Strength                  18147 non-null  int64 
 40  LongShots                 18147 non-null  int64 
 41  Aggression                18147 non-null  int64 
 42  Interceptions             18147 non-null  int64 
 43  Positioning               18147 non-null  int64 
 44  Vision                    18147 non-null  int64 
 45  Penalties                 18147 non-null  int64 
 46  Composure                 18147 non-null  int64 
 47  Marking                   18147 non-null  int64 
 48  StandingTackle            18147 non-null  int64 
 49  SlidingTackle             18147 non-null  int64 
 50  GKDiving                  18147 non-null  int64 
 51  GKHandling                18147 non-null  int64 
 52  GKKicking                 18147 non-null  int64 
 53  GKPositioning             18147 non-null  int64 
 54  GKReflexes                18147 non-null  int64 
 55  Release Clause            18147 non-null  object
 56  League                    18147 non-null  object
 57  Speciality                18147 non-null  object
dtypes: int64(43), object(15)
memory usage: 8.0+ MB
None
         Unnamed: 0           Age       Overall     Potential           Wage  \
count  18147.000000  18147.000000  18147.000000  18147.000000   18147.000000   
mean    9089.239599     25.121122     66.253926     71.324076    9759.023530   
std     5257.923360      4.669796      6.913320      6.132286   22030.250349   
min        0.000000     16.000000     46.000000     48.000000       0.000000   
25%     4536.500000     21.000000     62.000000     67.000000    1000.000000   
50%     9076.000000     25.000000     66.000000     71.000000    3000.000000   
75%    13662.500000     28.000000     71.000000     75.000000    9000.000000   
max    18206.000000     45.000000     94.000000     95.000000  565000.000000   

       International Reputation     Weak Foot   Skill Moves  Jersey Number  \
count              18147.000000  18147.000000  18147.000000   18147.000000   
mean                   1.113297      2.947154      2.361492      19.546096   
std                    0.394150      0.660498      0.756274      15.947765   
min                    1.000000      1.000000      1.000000       1.000000   
25%                    1.000000      3.000000      2.000000       8.000000   
50%                    1.000000      3.000000      2.000000      17.000000   
75%                    1.000000      3.000000      3.000000      26.000000   
max                    5.000000      5.000000      5.000000      99.000000   

           Crossing  ...     Penalties     Composure       Marking  \
count  18147.000000  ...  18147.000000  18147.000000  18147.000000   
mean      49.738414  ...     48.546371     58.651127     47.286053   
std       18.364255  ...     15.703113     11.437138     19.900450   
min        5.000000  ...      5.000000      3.000000      3.000000   
```

```
25%        38.000000  ...      39.000000     51.000000     30.000000
50%        54.000000  ...      49.000000     60.000000     53.000000
75%        64.000000  ...      60.000000     67.000000     64.000000
max        93.000000  ...      92.000000     96.000000     94.000000

       StandingTackle  SlidingTackle      GKDiving    GKHandling  \
count    18147.000000   18147.000000  18147.000000  18147.000000
mean        47.701879      45.666336     16.616906     16.393839
std         21.663630      21.287961     17.698612     16.909971
min          2.000000       3.000000      1.000000      1.000000
25%         27.000000      24.000000      8.000000      8.000000
50%         55.000000      52.000000     11.000000     11.000000
75%         66.000000      64.000000     14.000000     14.000000
max         93.000000      91.000000     90.000000     92.000000

          GKKicking  GKPositioning    GKReflexes
count  18147.000000   18147.000000  18147.000000
mean      16.233041      16.389651     16.712019
std       16.504103      17.037031     17.957521
min        1.000000       1.000000      1.000000
25%        8.000000       8.000000      8.000000
50%       11.000000      11.000000     11.000000
75%       14.000000      14.000000     14.000000
max       91.000000      90.000000     94.000000

[8 rows x 43 columns]
```

In [2]:

```
fifa19.isnull().sum()
```

Out[2]:
```
Unnamed: 0                   0
Name                         0
Age                          0
Nationality                  0
Overall                      0
Potential                    0
Club                         0
Value                        0
Wage                         0
Preferred Foot               0
International Reputation     0
Weak Foot                    0
Skill Moves                  0
Work Rate                    0
Position                     0
Jersey Number                0
Joined                    1493
Loaned From                  0
Contract Valid Until         0
Height                       0
Weight                       0
Crossing                     0
Finishing                    0
HeadingAccuracy              0
ShortPassing                 0
Volleys                      0
Dribbling                    0
Curve                        0
FKAccuracy                   0
LongPassing                  0
BallControl                  0
Acceleration                 0
SprintSpeed                  0
Agility                      0
Reactions                    0
Balance                      0
ShotPower                    0
Jumping                      0
Stamina                      0
Strength                     0
LongShots                    0
Aggression                   0
Interceptions                0
Positioning                  0
Vision                       0
Penalties                    0
Composure                    0
Marking                      0
StandingTackle               0
SlidingTackle                0
GKDiving                     0
GKHandling                   0
GKKicking                    0
GKPositioning                0
GKReflexes                   0
Release Clause               0
League                       0
Speciality                   0
dtype: int64
```

4.1 Comparing Two Samples-

We will compare the mean wages (salary) of the left-footed and right-footed players in the game, to check whether there is any significant difference between the two groups. We will perform this comparison using the z-test as the number of samples in this dataset is very large.

The results of this test can provide insights into whether there is any difference between the wages of left-footed and right-footed soccer players. If the results of the z-test suggest asignificant difference in the mean wages of left-footed and right-footed players, this may indicate that one group of players is more highly valued or in greater demand than the other. As a result, teams, agents, and other stakeholders in the soccer industry may adjust their strategies for recruitment, scouting, and player development accordingly.

In [3]:

```
# Defining the hypotheses-
# H0: There is no significant difference in the mean wages of the left-footed and righ
# H1: Not H0.

left_footed = fifa19[fifa19["Preferred Foot"] == "Left"]
right_footed = fifa19[fifa19["Preferred Foot"] == "Right"]

mean_left = left_footed["Wage"].mean()
mean_right = right_footed["Wage"].mean()
std_left = left_footed["Wage"].std()
std_right = right_footed["Wage"].std()
n_left = left_footed["Wage"].count()
n_right = right_footed["Wage"].count()
print('Mean wages of left-footed players: ', mean_left)
print('Mean wages of right-footed players: ', mean_right)

z = (mean_left - mean_right) / ((std_left**2 / n_left) + (std_right**2 / n_right))**0.
print("Test statistic: ", z)

p_value = norm.sf(abs(z))*2
print("P-value: ", p_value, '\n')

alpha = 0.05

if p_value < alpha:
    print("Reject the null hypothesis. There is a significant difference between the m
else:
    print("Fail to reject the null hypothesis. There is no significant difference betw
```

```
Mean wages of left-footed players:  10353.290567830838
Mean wages of right-footed players:  9579.566652317406
Test statistic:  1.8981027408159028
P-value:  0.05768254919771028

Fail to reject the null hypothesis. There is no significant difference between the me
an wages of left-footed and right-footed players.
```

4.2 The Analysis of Variance-

We will perform ANOVA to test if there is a significant difference in the mean overall rating across the different player positions. (ST- Attacker, CM- Midfielder, CB- Defender, GK- Goalkeeper)

It is a statistical technique used to compare the means of two or more groups of data. ANOVA helps us determine if there is a significant difference between the groups, or if any observed differences are likely due to random sampling variation.

In [4]:

```
# Defining the hypotheses-
# H0: There is no significant difference in the mean overall rating across different p
# H1: Not H0.

forward_players = fifa19[fifa19['Position'] == 'ST']
midfielder_players = fifa19[fifa19['Position'] == 'CM']
defender_players = fifa19[fifa19['Position'] == 'CB']
goalkeeper_players = fifa19[fifa19['Position'] == 'GK']

alpha = 0.05

f_statistic, p_value = f_oneway(forward_players['Overall'], midfielder_players['Overal
print('F-statistic: ', f_statistic)

df_between = 3
df_within = fifa19.shape[0] - df_between*4
critical_value = f.ppf(1-alpha, df_between, df_within)
print('Critical value: ', critical_value)
```

```
F-statistic:  19.2403771172369
Critical value:  2.6053987903176643
```

In [5]:

```
if f_statistic > critical_value:
    print("Reject the null hypothesis. There is a significant difference in the mean c
else:
    print("Fail to reject the null hypothesis. There is no significant difference in t
```

```
Reject the null hypothesis. There is a significant difference in the mean overall rat
ing across different player positions.
```

4.3 The Analysis of Categorical Data-

We will perform the categorical data analysis using the chi-square test of independence. This test is used to determine if there is a significant association between two categorical variables. The two variables are "Position" and "Salary" (categorical, after being converted to a categorical variable based on quartiles). We want to test whether there is a significant association between these two variables.

In [6]:

```
# Defining the hypotheses-
# H0: There is no relationship between the two categorical variables being compared.
# H1: Not H0.

fifa19['Salary Category'] = pd.qcut(fifa19['Wage'], q=4, labels=['Low', 'Medium', 'Hig

for quartile in fifa19['Salary Category'].unique():
    min_wage = fifa19[fifa19['Salary Category'] == quartile]['Wage'].min()
```

```
    max_wage = fifa19[fifa19['Salary Category'] == quartile]['Wage'].max()
    print(f"The range of wages in the {quartile} quartile is from €{min_wage} to €{max

print('\n')

contingency_table = pd.crosstab(fifa19['Position'], fifa19['Salary Category'])
contingency_table = contingency_table.replace(np.nan, 0)
contingency_table = contingency_table.replace(np.inf, 0)
print(contingency_table)

chi2, p_value, dof, expected = chi2_contingency(contingency_table)
print("\nChi-square value: ", chi2)
print("p-value: ", p_value, '\n')

if p_value < 0.05:
    print("Reject the null hypothesis. There is a significant relationship between the
else:
    print("Fail to reject the null hypothesis. We do not have enough evidence to rejec
```

```
The range of wages in the Very High quartile is from €10000 to €565000.
The range of wages in the Low quartile is from €0 to €1000.
The range of wages in the High quartile is from €4000 to €9000.
The range of wages in the Medium quartile is from €2000 to €3000.


Salary Category  Low  Medium  High  Very High
Position
CAM              271     213   223        251
CB               580     510   355        333
CDM              266     240   226        216
CF                25      18     8         23
CM               515     364   246        269
GK               874     463   355        333
LAM                0       3     4         14
LB               377     372   291        282
LCB              118     159   170        201
LCM               58     106    95        136
LDM               55      54    59         75
LF                 1       4     3          7
LM               267     275   273        280
LS                21      50    49         87
LW                89     103    89        100
LWB               23      23    12         20
RAM                2       2     2         15
RB               360     370   283        278
RCB              117     171   173        201
RCM               55     104   109        123
RDM               48      54    59         87
RF                 0       3     4          9
RM               298     287   264        275
RS                21      50    62         70
RW                85      93    88        104
RWB               28      23    16         20
ST               548     551   525        528

Chi-square value:  769.6711307097052
p-value:  2.7226385414426825e-114

Reject the null hypothesis. There is a significant relationship between the two categ
orical variables being compared.
```

This means that the data provides strong evidence to support the claim that there is a relationship between a player's position and their salary category. Specifically, it suggests that certain positions tend to have higher salaries than others.
In the real world, this finding could be useful in a variety of ways. For example, it could be used by clubs and managers during decision-making when it comes to player acquisition, contract negotiations and squad building. It could also be of interest to agents and representatives who are negotiating contracts for players.

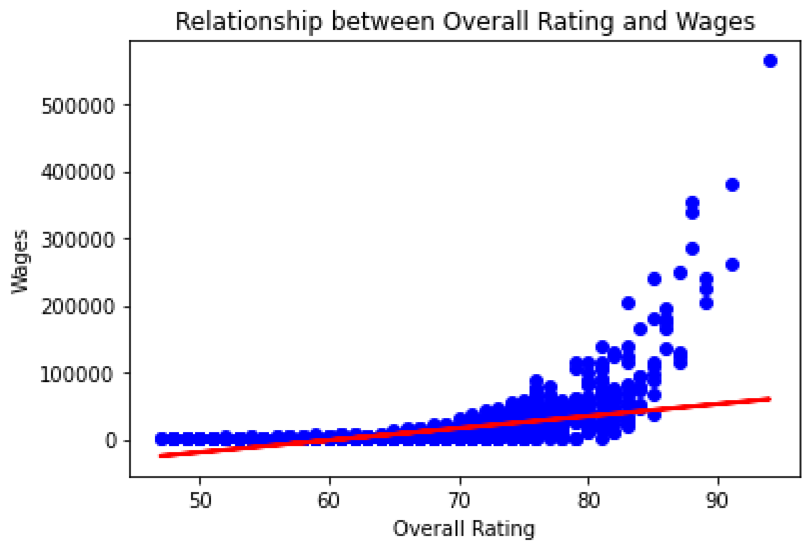
4.4 Linear Regression-

We will perform a linear regression to find the relationship between a player's rating and their wage. Linear regression is a statistical technique used to model the relationship between two variables by fitting a linear equation to the observed data. It assumes that there is a linear relationship between the independent variable and the dependent variable. The goal of linear regression is to find the best-fitting straight line through the data, which can be used to predict the value.

In this specific example, we will try to use the overall ratings of the players in order to predict their wages.

In [7]:
```
fifa19 = fifa19.dropna()

X = fifa19['Overall'].values.reshape(-1, 1) # independent variable
y = fifa19['Wage'].values.reshape(-1, 1) # dependent variable

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean squared error: ", mse)

r2 = r2_score(y_test, y_pred)
print('R-squared:', r2)

plt.scatter(X_test, y_test, color='blue')
plt.plot(X_test, y_pred, color='red', linewidth=2)
plt.title('Relationship between Overall Rating and Wages')
plt.xlabel('Overall Rating')
plt.ylabel('Wages')
plt.show()
```

```
Mean squared error:  427652443.59555185
R-squared: 0.2972323505316462
```



In [13]:
```
fifa19 = fifa19[["Overall", "Wage"]]

fifa19["Wage"] = fifa19["Wage"].astype(float)
print(fifa19.head())
```

```
   Overall      Wage
0       94  565000.0
1       94  405000.0
2       92  290000.0
3       91  260000.0
4       91  355000.0
```

An R-squared value of 0.297 suggests that the linear regression model explains around 29.7% of the variance in wages based on ratings. This indicates that there are likely other factors that influence wages as well, and the model is not able to capture all of the variation in the data.

4.5 Resampling Methods-

Cross-validation is a statistical technique used to evaluate how well a machine learning model generalizes to new, unseen data. It involves partitioning a dataset into subsets, or folds, where one fold is used as the testing set and the remaining folds are used as the training set. This process is repeated multiple times, with different folds being used as the testing set each time. The performance of the model is then averaged over all the iterations to provide an estimate of the model's accuracy.

We first split the dataset into features and target variables. We then split the dataset into training and testing sets using an 80-20 split. We create a Linear Regression model and then perform 10-fold cross-validation on the training set using Scikit-Learn.

In [8]:
```
fifa19 = pd.read_csv('fifa19.csv')

def value_to_float(x):
    if 'K' in x:
        return float(x.replace('€', '').replace('K', '')) * 1000
    elif 'M' in x:
```

```
        return float(x.replace('€', '').replace('M', '')) * 1000000
    else:
        return float(x.replace('€', ''))

fifa19['Value'] = fifa19['Value'].apply(value_to_float)
X = fifa19[['Overall', 'Potential', 'International Reputation', 'Skill Moves', 'Jersey
y = fifa19['Value']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

model = LinearRegression()
alphas = [0.001, 0.01, 0.1, 1, 10, 100, 1000]

scores = cross_val_score(model, X_train, y_train, cv=10, scoring='neg_mean_squared_err
if np.isnan(scores).any():
    print("Warning: NaN values in scores array.")

avg_score = -1 * scores.mean()
print("Average MSE score from 10-fold cross-validation: ", avg_score)

model.fit(X_train, y_train)
y_pred = model.predict(X_test)
test_mse = mean_squared_error(y_test, y_pred)
print("Test MSE: ", test_mse)
```

```
Average MSE score from 10-fold cross-validation:  12432875227166.402
Test MSE:  11741755484862.246
```

The average MSE score represents the average mean squared error (MSE) across 10 rounds of cross-validation. The average MSE score from cross-validation gives an estimate of how well the model is likely to perform on new, unseen data. The result obatined is quite high, which suggests that the model may not be fitting the data very well.

4.6 Linear Model Selection and Regularization-

Linear model selection and regularization refer to a set of techniques used to improve the performance of linear regression models by selecting the most relevant features and reducing overfitting. Linear model selection addresses this issue by identifying the most relevant variables to include in the model. Regularization, on the other hand, is a technique that reduces overfitting by imposing a penalty on the magnitude of the model coefficients. The two most common forms of regularization are Lasso and Ridge.

Lasso regularization is a technique used to prevent overfitting in a regression model. It helps the model to focus on the most important features and avoid being too complex, which can be especially useful when dealing with high-dimensional datasets with many features.

In [9]:

```
X = fifa19[["Wage", "Potential"]]
y = fifa19["Overall"]

scaler = StandardScaler()
X = scaler.fit_transform(X)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

alpha = 0.1
model = Lasso(alpha=alpha).fit(X_train, y_train)

score = model.score(X_test, y_test)
print(f"R^2 score: {score}")
```

```
R^2 score: 0.5122648918555313
```

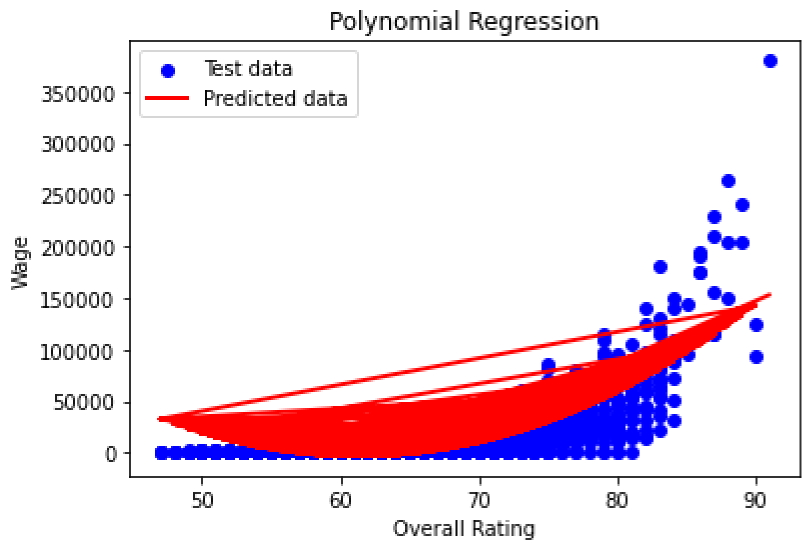
This R^2 score means that the model explains about 51% of the variance in the target variablem i.e. "Overall". Basically, the model is able to capture some of the underlying relationship between "Wage", "Potential", and "Overall", but there is still a lot of variation that the model cannot account for.

4.7 Moving Beyond Linearity-

In this section we will perform a polynomial regression to assess the relationship between a player's overall rating and their wage. Polynomial regression is a type of regression analysis in which the relationship between the independent variable x and the dependent variable y is modeled as an n-th degree polynomial function. The goal of polynomial regression is to find the best-fitting polynomial curve that describes the relationship between x and y in the data.

In [10]:
```
X = fifa19['Overall'].values.reshape(-1, 1)
y = fifa19['Wage'].values.reshape(-1, 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

poly_degree = 2
poly_features = PolynomialFeatures(degree=poly_degree)
X_poly_train = poly_features.fit_transform(X_train)
poly_model = LinearRegression()
poly_model.fit(X_poly_train, y_train)
X_poly_test = poly_features.transform(X_test)
y_pred = poly_model.predict(X_poly_test)

rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
print('Root-mean-squared-error:', rmse)
print('R-squared:', r2)

plt.scatter(X_test, y_test, color='blue', label='Test data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted data')
plt.title('Polynomial Regression')
plt.xlabel('Overall Rating')
plt.ylabel('Wage')
plt.legend()
plt.show()
```

```
Root-mean-squared-error: 12083.993325441523
R-squared: 0.6346305915393289
```



The RMSE shows that on average, the predicted wages from the model differ from the true wages by around €12,084, which isn't too bad of a prediction. The R-squared value indicates that the model explains around 63.5% of the variance in the wages. This model is much more accurate and true when compared to the linear regression model which had an R-squared value of 29.7%, that is the polynomial regression model has more than twice the R-squared value of the other.

These results suggest that there is a moderate positive relationship between a player's overall rating and their wage, with higher-rated players generally earning higher wages.

CONCLUSION

Based on the project's findings, it is clear that there is a significant relationship between a player's wages and their performance attributes in soccer. The z-test results suggest that there isn't a significant difference in the wages of left-footed and right-footed players, which could have had implications for team selection strategies and negotiation of player contracts. The categorical data analysis reveals that there is a significant association between a player's position and salary. This information could be used by managers to make more informed decisions about player recruitment and salary negotiation.

The use of cross-validation and Lasso regularization techniques in the machine learning model ensures that the model is pretty reliable and not overfitting the data too much. This could lead to more accurate predictions and insights for teams and player agents.

The linear regression results suggest that a player's rating has a positive relationship with their wage. But the polynomial regression results suggest that a player's overall rating has a much more positive relationship with their wage. This information could be used by teams to justify higher wages for players with higher overall ratings, as well as by player agents to negotiate better contracts for their clients.

Finally, this project's findings have significant implications for the soccer industry, providing valuable insights into the relationship between player wages and performance attributes. These insights could be used to inform talent recruitment, player contract negotiation, and team selection strategies, ultimately leading to more successful outcomes for teams and players as well.

REFERENCES

1. https://www.kirenz.com/post/2019-08-12-python-lasso-regression-auto/
2. https://towardsdatascience.com/chi-square-test-with-python-d8ba98117626
3. https://www.reneshbedre.com/blog/anova.html
4. Rice, John. A. (2006). Mathematical Statistics and Data Analysis.