```python
import pandas as pd
file_path = '/players_15.csv'
fifa_data = pd.read_csv(file_path)
print(fifa_data.head())
```

```
   sofifa_id         short_name                            long_name
age  \
0     158023           L. Messi       Lionel Andrés Messi Cuccittini
27
1      20801  Cristiano Ronaldo  Cristiano Ronaldo dos Santos Aveiro
29
2       9014          A. Robben                         Arjen Robben
30
3      41236     Z. Ibrahimović                   Zlatan Ibrahimović
32
4     167495           M. Neuer                         Manuel Neuer
28

          dob  height_cm  weight_kg  nationality                 club
\
0  24/06/1987        169         67    Argentina         FC Barcelona

1  05/02/1985        185         80     Portugal          Real Madrid

2  23/01/1984        180         80  Netherlands    FC Bayern München

3  03/10/1981        195         95       Sweden  Paris Saint-Germain

4  27/03/1986        193         92      Germany    FC Bayern München


   overall  potential  value_eur  wage_eur preferred_foot  \
0       93         95          0         0           Left
1       92         92          0         0          Right
2       90         90          0         0           Left
3       90         90          0         0          Right
4       90         90          0         0          Right

   international_reputation  weak_foot  skill_moves
0                         5          3            4
1                         5          4            5
2                         5          2            4
3                         5          4            4
4                         5          4            1
```

```python
# Feature classification --#manoj R  pes2ug23cs328
feature_classification = {
    "Nominal": ["short_name", "long_name", "nationality", "club",
"preferred_foot"],
    "Ordinal": ["international_reputation", "weak_foot",
"skill_moves"],
    "Interval": [],
    "Ratio": ["age", "height_cm", "weight_kg", "overall", "potential",
"value_eur", "wage_eur"]
}
print(feature_classification)
```

```
{'Nominal': ['short_name', 'long_name', 'nationality', 'club',
'preferred_foot'], 'Ordinal': ['international_reputation',
'weak_foot', 'skill_moves'], 'Interval': [], 'Ratio': ['age',
'height_cm', 'weight_kg', 'overall', 'potential', 'value_eur',
'wage_eur']}
```

```
# Data quality issues:- #manoj R  pes2ug23cs328
missing_values = fifa_data.isnull().sum()
print("Missing Values:\n", missing_values)

duplicates = fifa_data.duplicated().sum()
print("Duplicate Rows:", duplicates)
```

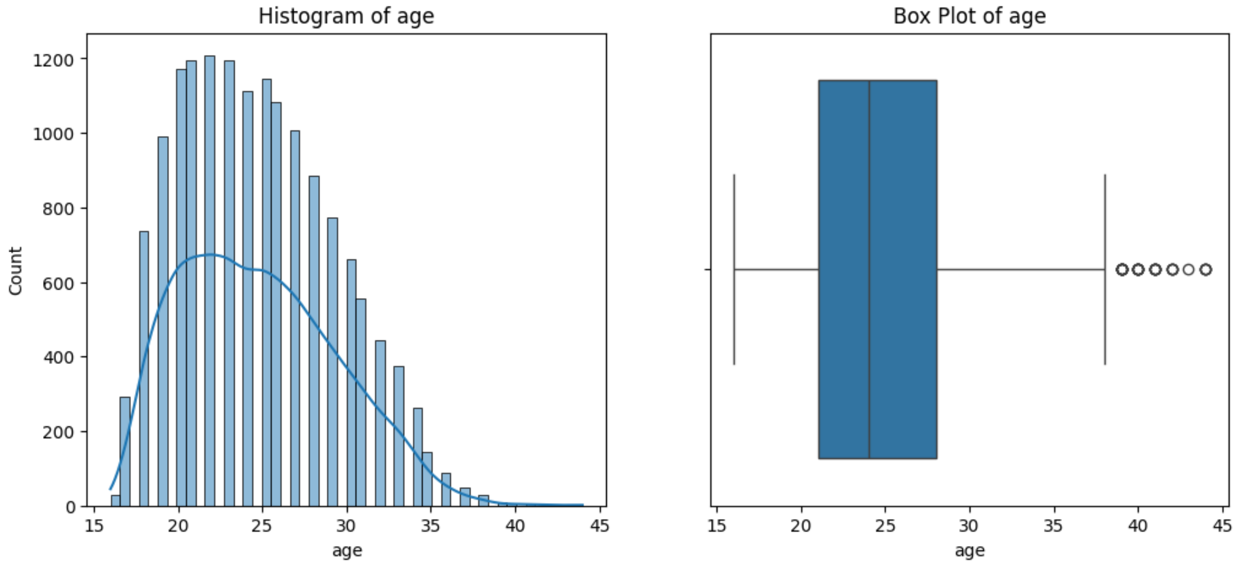
```
Missing Values:
 sofifa_id                   0
short_name                  0
long_name                   0
age                         0
dob                         0
height_cm                   0
weight_kg                   0
nationality                 0
club                        0
overall                     0
potential                   0
value_eur                   0
wage_eur                    0
preferred_foot              0
international_reputation    0
weak_foot                   0
skill_moves                 0
dtype: int64
Duplicate Rows: 0
```

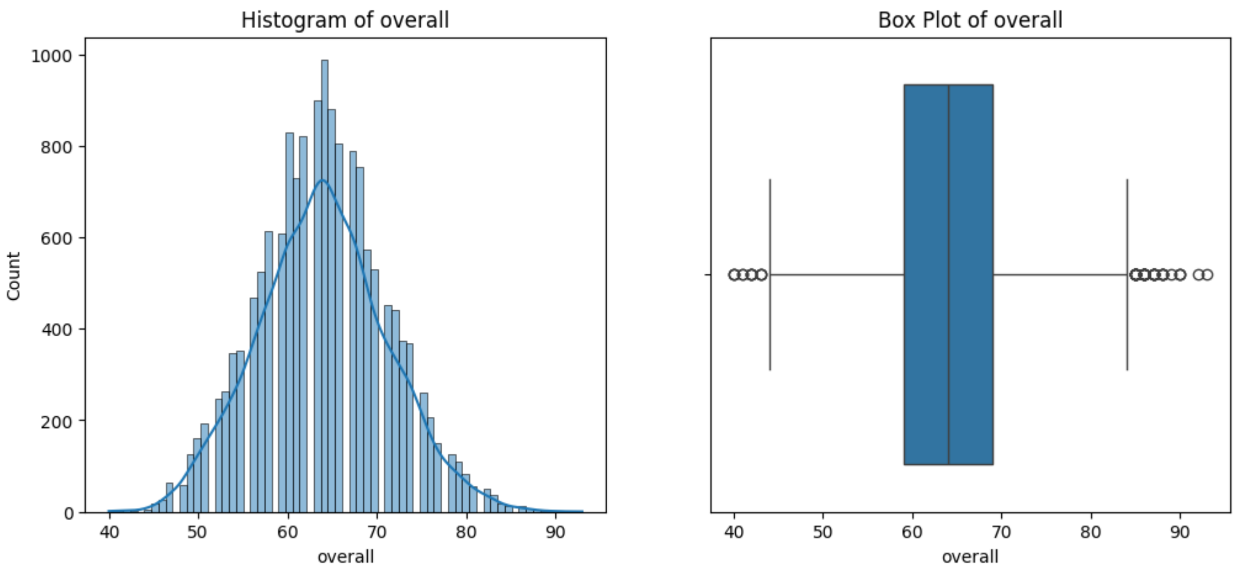
```
# Summary statistics  --Narendra babu pes2ug24cs815
summary_statistics = fifa_data.describe()
print("Summary Statistics:\n", summary_statistics)

range_values = fifa_data.max(numeric_only=True) -
fifa_data.min(numeric_only=True)
print("Range Values:\n", range_values)
```

```
Summary Statistics:
            sofifa_id           age     height_cm     weight_kg
```

```
overall  \
count   15465.000000  15465.000000  15465.000000  15465.000000
15465.000000
mean   189298.588425     24.763272    181.093631     75.482703
63.948594
std     39648.820272      4.624565      6.635182      6.907243
7.208610
min         2.000000     16.000000    155.000000     50.000000
40.000000
25%    178043.000000     21.000000    176.000000     70.000000
59.000000
50%    200844.000000     24.000000    181.000000     75.000000
64.000000
75%    214326.000000     28.000000    186.000000     80.000000
69.000000
max    225562.000000     44.000000    203.000000    110.000000
93.000000

          potential  value_eur  wage_eur  international_reputation  \
count  15465.000000    15465.0   15465.0              15465.000000
mean      68.483091        0.0       0.0                  1.126350
std        6.611708        0.0       0.0                  0.401362
min       40.000000        0.0       0.0                  1.000000
25%       64.000000        0.0       0.0                  1.000000
50%       68.000000        0.0       0.0                  1.000000
75%       73.000000        0.0       0.0                  1.000000
max       95.000000        0.0       0.0                  5.000000

          weak_foot   skill_moves
count  15465.000000  15465.000000
mean       2.932363      2.267055
std        0.652270      0.719035
min        1.000000      1.000000
25%        3.000000      2.000000
50%        3.000000      2.000000
75%        3.000000      3.000000
max        5.000000      5.000000
Range Values:
 sofifa_id                   225560
age                             28
height_cm                       48
weight_kg                       60
overall                         53
potential                       55
value_eur                        0
wage_eur                         0
international_reputation         4
weak_foot                        4
```

```
skill_moves                      4
dtype: int64

import matplotlib.pyplot as plt
import seaborn as sns

# --Narendra babu pes2ug24cs815
for column in ['age', 'overall']:
    plt.figure(figsize=(12, 5))

    # Histogram
    plt.subplot(1, 2, 1)
    sns.histplot(fifa_data[column], kde=True)
    plt.title(f'Histogram of {column}')

    # Box Plot
    plt.subplot(1, 2, 2)
    sns.boxplot(x=fifa_data[column])
    plt.title(f'Box Plot of {column}')

    plt.show()
```

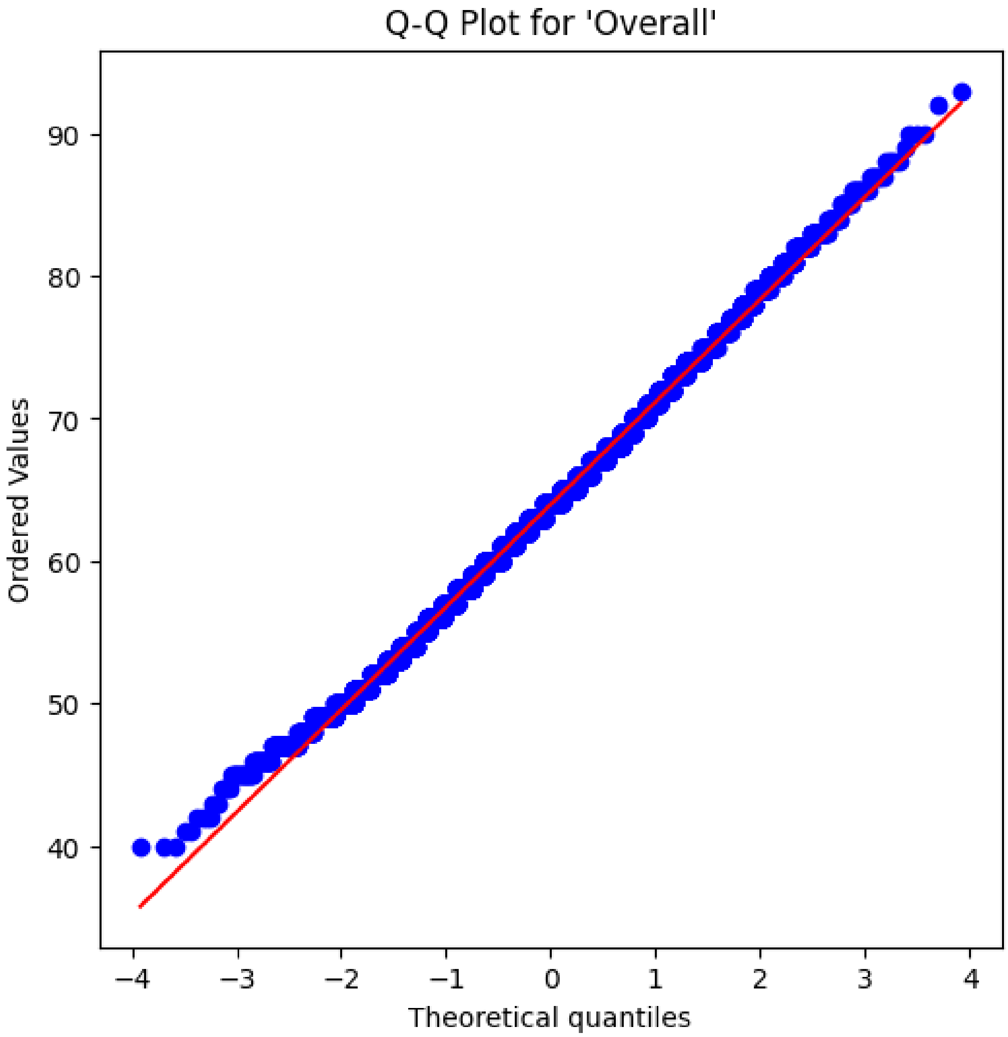




```
# Outer Handling --Narendra babu pes2ug24cs815
def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    outliers = df[(df[column] < (Q1 - 1.5 * IQR)) | (df[column] > (Q3
+ 1.5 * IQR))]
    return outliers

age_outliers = detect_outliers(fifa_data, 'age')
overall_outliers = detect_outliers(fifa_data, 'overall')
print("Age Outliers:", len(age_outliers))
print("Overall Outliers:", len(overall_outliers))
```

```
Age Outliers: 31
Overall Outliers: 56
```

```
import scipy.stats as stats

# Q-Q plot --Rohan A pes2ug24cs819
plt.figure(figsize=(6, 6))
stats.probplot(fifa_data['overall'], dist="norm", plot=plt)
plt.title("Q-Q Plot for 'Overall'")
plt.show()
```

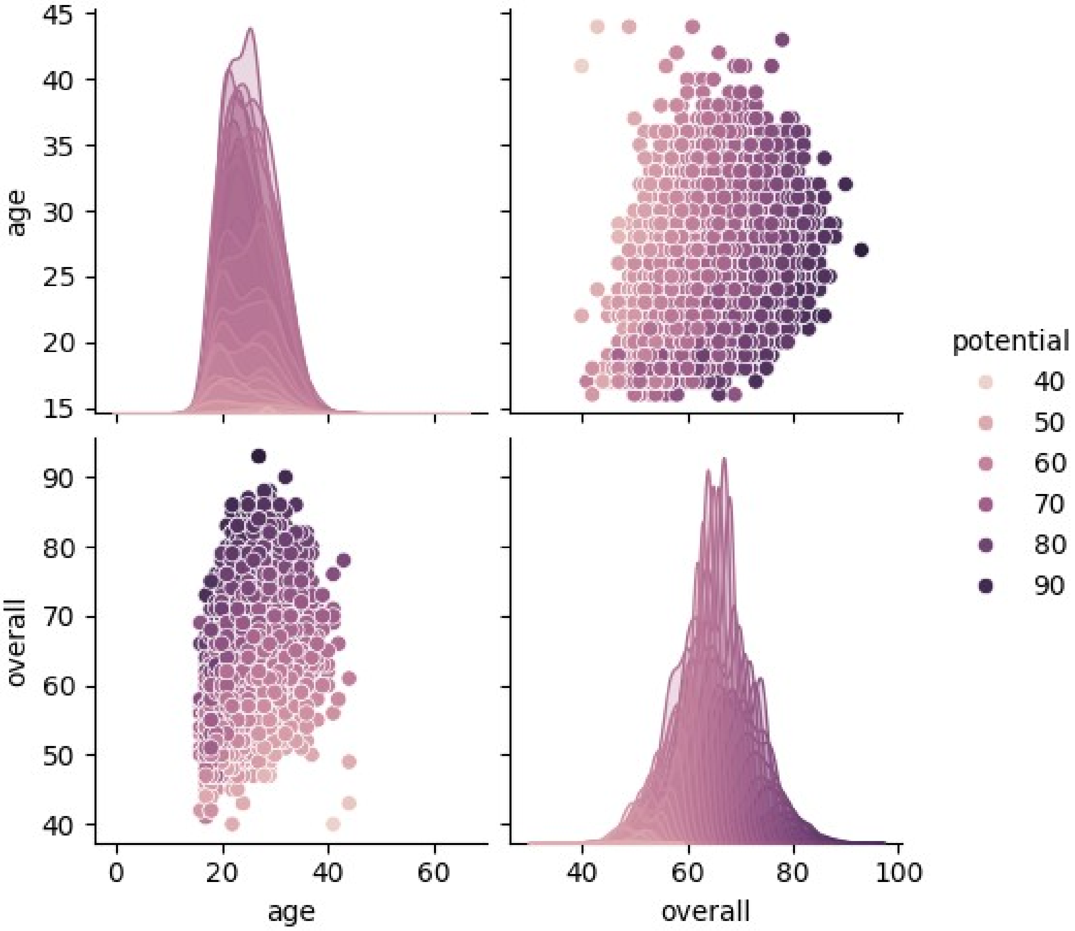


```
# Correlation analysis --Rohan A pes2ug24cs819
numeric_data = fifa_data.select_dtypes(include=['float64', 'int64'])
age_correlation = numeric_data.corr()
['age'].sort_values(ascending=False)
print("Correlation with Age:\n", age_correlation)
overall_correlation = numeric_data.corr()
['overall'].sort_values(ascending=False)
print("Strongest Correlation with Overall:\n", overall_correlation)
```

```
Correlation with Age:
 age                         1.000000
overall                     0.436108
international_reputation    0.281662
weight_kg                   0.211907
weak_foot                   0.085481
height_cm                   0.084419
skill_moves                -0.002690
potential                  -0.071597
```

```
sofifa_id                  -0.699146
value_eur                        NaN
wage_eur                         NaN
Name: age, dtype: float64
Strongest Correlation with Overall:
 overall                     1.000000
potential                   0.805234
international_reputation    0.524089
age                         0.436108
skill_moves                 0.275949
weak_foot                   0.227190
weight_kg                   0.124379
height_cm                   0.050320
sofifa_id                  -0.388582
value_eur                        NaN
wage_eur                         NaN
Name: overall, dtype: float64
```

```
#  pair plot --Rohan A pes2ug24cs819
sample_data = fifa_data.sample(n=10000, random_state=42)
sns.pairplot(sample_data, vars=['age', 'overall'], hue='potential')
plt.show()
```



```
from scipy.stats import mannwhitneyu

# Hypothesis testing  --manoj R  pes2ug23cs328
age_group_1 = fifa_data[fifa_data['age'] <= 25]['overall']
age_group_2 = fifa_data[fifa_data['age'] > 25]['overall']
stat, p_value = mannwhitneyu(age_group_1, age_group_2,
alternative='two-sided')
print("Mann-Whitney U Test")
print("Statistic:", stat)
print("P-value:", p_value)
```

```
Mann-Whitney U Test
Statistic: 16689159.0
P-value: 0.0
```

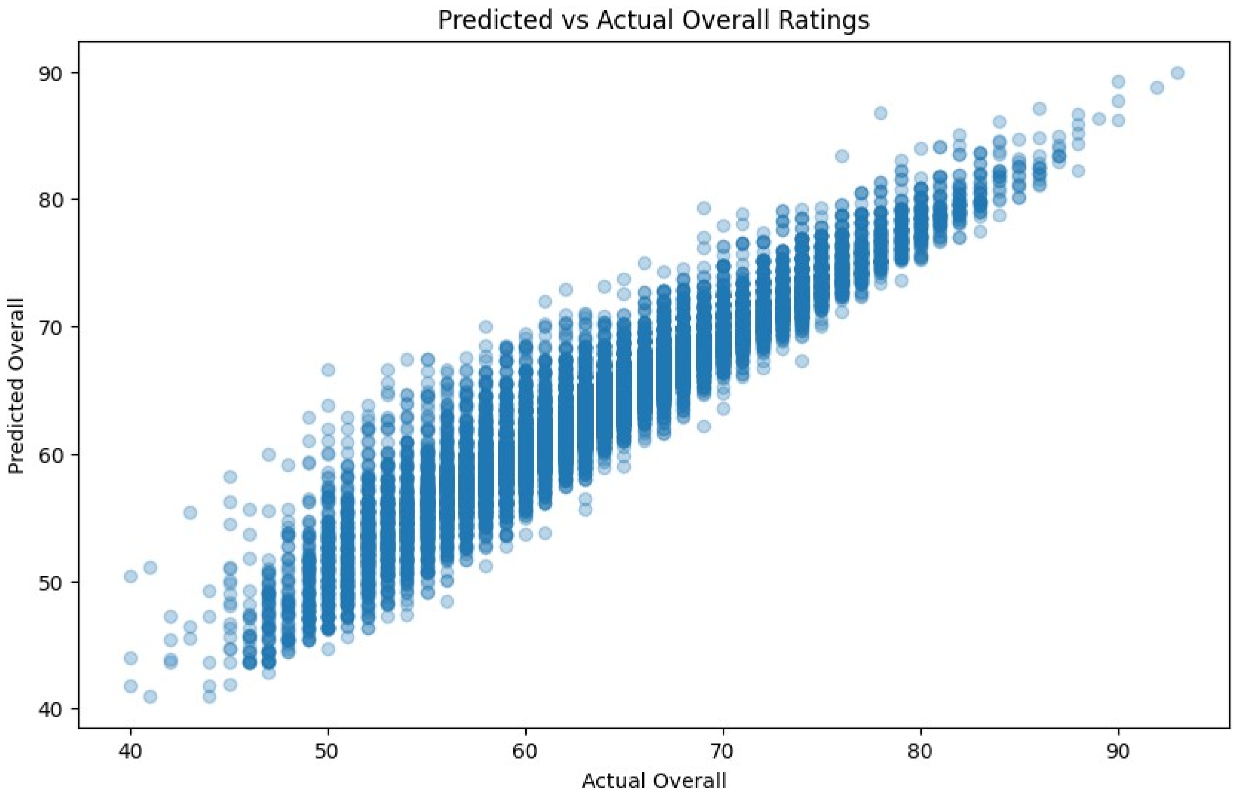
```
import numpy as np

# Margin of error --manoj R  pes2ug23cs328
n = len(fifa_data['overall'])
std_dev = fifa_data['overall'].std()
z_score = 1.96
```

```
margin_of_error = z_score * (std_dev / np.sqrt(n))
print("Margin of Error:", margin_of_error)
```

```
Margin of Error: 0.11361420191764746
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

#Linear Regression Analysis  --Narendra babu pes2ug24cs815
X = fifa_data[['age', 'potential']]
y = fifa_data['overall']

model = LinearRegression()
model.fit(X, y)

y_pred = model.predict(X)

mse = mean_squared_error(y, y_pred)
rmse = np.sqrt(mse)
print("MSE:", mse)
print("RMSE:", rmse)

plt.figure(figsize=(10, 6))
plt.scatter(y, y_pred, alpha=0.3)
plt.xlabel('Actual Overall')
plt.ylabel('Predicted Overall')
plt.title('Predicted vs Actual Overall Ratings')
plt.show()
```

```
MSE: 5.536027215732434
RMSE: 2.352876370685981
```



```
# Feature Engineering  ----Rohan A pes2ug24cs819
fifa_data['performance_consistency'] = fifa_data['overall'] /
fifa_data['potential']
fifa_data['experience_level'] = fifa_data['age'] *
fifa_data['international_reputation']

print(fifa_data[['age', 'overall', 'potential',
'performance_consistency', 'experience_level']].head())
```

```
   age  overall  potential  performance_consistency  experience_level
0   27       93         95                 0.978947               135
1   29       92         92                 1.000000               145
2   30       90         90                 1.000000               150
3   32       90         90                 1.000000               160
4   28       90         90                 1.000000               140
```