# Tamas Flesch Thesis - Linear regression step

## LMJU - UpGrad - DS

### Fifa 23 Ultimate Team player price prediction based on the player's attributes

Table of contents



## 1. Read data

### Imports

In [1]:
```
# Suppressing Warnings
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Importing Pandas and NumPy
import pandas as pd, numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE

import statsmodels.api as sm
```

In [3]:
```
pd.options.display.max_columns = None
pd.options.display.max_rows = None
```

In [4]:
```
# Importing all datasets
futbin_data = pd.read_csv("futbin.csv")
futbin_data.head()
```

Out[4]:

| | Name | Rating | Price | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | De / S |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Pelé | 98 | 3270000.0 | 5 | 4 | 95.0 | 96 | 93 | 96 | |
| **1** | Lionel Messi | 98 | 4350000.0 | 4 | 4 | 93.0 | 98 | 97 | 99 | |
| **2** | Lionel Messi | 98 | 4640000.0 | 4 | 4 | 94.0 | 97 | 96 | 99 | |
| **3** | Karim Benzema | 97 | 1850000.0 | 4 | 5 | 92.0 | 97 | 90 | 94 | |
| **4** | Kylian Mbappé | 97 | 9750000.0 | 5 | 4 | 99.0 | 96 | 88 | 98 | |

In [5]:
```
futbin_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6397 entries, 0 to 6396
Columns: 105 entries, Name to Alt_Pos_3_missing
dtypes: float64(2), int64(102), object(1)
memory usage: 5.1+ MB
```

In [6]:
```
len(futbin_data[futbin_data.Price > 100000])
```

Out[6]:
```
153
```

In [7]:
```
153/6397*100
```

Out[7]:
```
2.3917461309989054
```

It seems to me that the Price are a little bit strange variable, there are some outliers in the data.

We have only 153 players in the database which have a the Price greater than 100.000. Which is almost 2.5%.

I will create two models, one for the 'average' players and one for the extra expensive players. It will be interesting to compare what are the important features for the two groups.

In [8]:
```
average_players = futbin_data[futbin_data.Price < 100000]
```

In [9]:
```
average_players.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6243 entries, 30 to 6396
Columns: 105 entries, Name to Alt_Pos_3_missing
dtypes: float64(2), int64(102), object(1)
memory usage: 5.0+ MB
```

In [10]:
```
star_players = futbin_data[futbin_data.Price >= 100000]
```

In [11]:
```
star_players.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 154 entries, 0 to 566
Columns: 105 entries, Name to Alt_Pos_3_missing
dtypes: float64(2), int64(102), object(1)
memory usage: 127.5+ KB
```

## 2. Price distribution analysis

In [12]:
```
futbin_data.Price.describe()
```

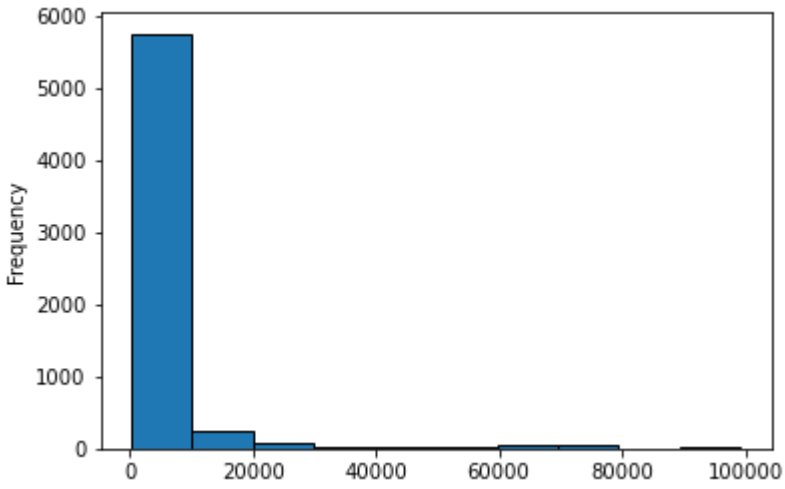
Out[12]:
```
count    6.397000e+03
mean     2.557498e+04
std      3.280786e+05
min      2.000000e+02
25%      2.000000e+02
50%      3.500000e+02
75%      1.000000e+03
max      1.500000e+07
Name: Price, dtype: float64
```

In [13]:
```
len(futbin_data[futbin_data.Price == 0])
```

Out[13]:
```
0
```

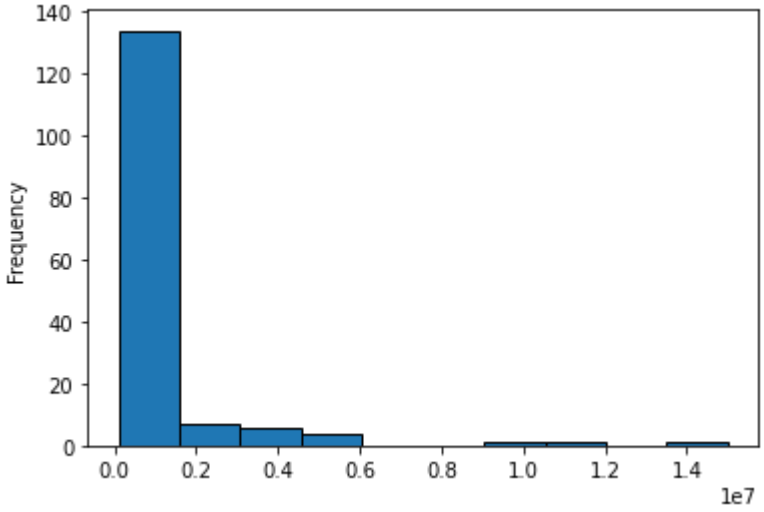
In [14]:
```
#plot distribution of values in price column
average_players['Price'].plot(kind='hist', edgecolor='black')
```

Out[14]:
```
<AxesSubplot:ylabel='Frequency'>
```




In [15]:
```
star_players['Price'].plot(kind='hist', edgecolor='black')
```

Out[15]:
```
<AxesSubplot:ylabel='Frequency'>
```




## 3. Test-Train Split

### average

In [16]:
```
# Putting feature variable to X
X_avg = average_players.drop(['Name','Price'], axis=1)

X_avg.head()
```

Out[16]:

| | Rating | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | Defense / Speed | Physical / Positioning | I |
|---|---|---|---|---|---|---|---|---|---|---|
| **30** | 94 | 1 | 4 | 92.0 | 91 | 94 | 96 | 53 | 91 | |
| **46** | 93 | 4 | 5 | 90.0 | 90 | 81 | 85 | 43 | 84 | |
| **51** | 93 | 3 | 5 | 87.0 | 82 | 93 | 92 | 92 | 93 | |
| **54** | 93 | 5 | 4 | 94.0 | 90 | 90 | 94 | 69 | 78 | |
| **56** | 93 | 1 | 3 | 89.0 | 95 | 95 | 94 | 73 | 90 | |

In [17]:
```
# Putting response variable to y
y_avg = average_players['Price']

y_avg.head()
```

Out[17]:
```
30    61500.0
46    75000.0
51    70000.0
54    80000.0
56    38500.0
Name: Price, dtype: float64
```

In [18]:
```
# Splitting the average data into train and test
X_avg_train, X_avg_test, y_avg_train, y_avg_test = train_test_split(X_avg, y_avg, trai
```

## star

In [19]:
```
# Putting feature variable to X
X_star = star_players.drop(['Name','Price'], axis=1)

X_star.head()
```

Out[19]:

| | Rating | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | Defense / Speed | Physical / Positioning | P |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 98 | 5 | 4 | 95.0 | 96 | 93 | 96 | 60 | 76 | |
| **1** | 98 | 4 | 4 | 93.0 | 98 | 97 | 99 | 40 | 77 | |
| **2** | 98 | 4 | 4 | 94.0 | 97 | 96 | 99 | 40 | 79 | |
| **3** | 97 | 4 | 5 | 92.0 | 97 | 90 | 94 | 45 | 90 | |
| **4** | 97 | 5 | 4 | 99.0 | 96 | 88 | 98 | 44 | 87 | |

In [20]:
```
# Putting response variable to y
y_star = star_players['Price']

y_star.head()
```

Out[20]:
```
0    3270000.0
1    4350000.0
2    4640000.0
3    1850000.0
4    9750000.0
Name: Price, dtype: float64
```

In [21]:
```
# Splitting the star data into train and test
X_star_train, X_star_test, y_star_train, y_star_test = train_test_split(X_star, y_star
```

## full data

In [22]:
```
# Putting feature variable to X
X_full = futbin_data.drop(['Name','Price'], axis=1)

X_full.head()
```

Out[22]:

| | Rating | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | Defense / Speed | Physical / Positioning | P |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 98 | 5 | 4 | 95.0 | 96 | 93 | 96 | 60 | 76 | |
| **1** | 98 | 4 | 4 | 93.0 | 98 | 97 | 99 | 40 | 77 | |
| **2** | 98 | 4 | 4 | 94.0 | 97 | 96 | 99 | 40 | 79 | |
| **3** | 97 | 4 | 5 | 92.0 | 97 | 90 | 94 | 45 | 90 | |
| **4** | 97 | 5 | 4 | 99.0 | 96 | 88 | 98 | 44 | 87 | |

In [23]:
```
# Putting response variable to y
y_full = futbin_data['Price']

y_full.head()
```

Out[23]:
```
0    3270000.0
1    4350000.0
2    4640000.0
3    1850000.0
4    9750000.0
Name: Price, dtype: float64
```

In [24]:
```
# Splitting the full data into train and test
X_full_train, X_full_test, y_full_train, y_full_test = train_test_split(X_full, y_full
```

## 4. Feature Scaling

In [152…
```
# columns to be scaled
scale_columns = ['Rating','Skills_Star','Weak_Foot_Star', 'Pace / Diving', 'Shooting /
                 'Dribbling / Reflexes', 'Defense / Speed', 'Physical / Positioning', '
                 'Ingame_Stats', 'Height_in_cm', 'BodyType_Weight', 'Alt_Pos_Count']
```

### average player scaling

In [26]:
```
scaler = StandardScaler()

X_avg_train[scale_columns] = scaler.fit_transform(X_avg_train[scale_columns])

X_avg_train.head()
```

Out[26]:

| | Rating | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | Defense / Speed | F Po |
|---|---|---|---|---|---|---|---|---|---|
| **5917** | -1.314890 | -1.508137 | -1.420183 | -1.336225 | -2.100084 | -2.308421 | -3.184101 | 0.375084 | - |
| **3239** | -0.116787 | 0.738101 | -0.002555 | -0.906434 | 0.326950 | 0.437884 | 0.159459 | 0.728813 | |
| **4563** | -0.476218 | 0.738101 | -0.002555 | 0.898689 | 0.124697 | 0.071710 | 0.063929 | -1.275649 | - |
| **1641** | 0.482264 | 1.486847 | -1.420183 | 0.812730 | 0.326950 | 0.529427 | 1.401353 | -1.157740 | - |
| **3696** | -0.236598 | 0.738101 | 1.415074 | 0.640814 | 0.461785 | 0.437884 | 0.159459 | 0.139265 | - |

## star player scaling

In [27]:
```
#scaler = StandardScaler()

X_star_train[scale_columns] = scaler.fit_transform(X_star_train[scale_columns])

X_star_train.head()
```

Out[27]:

| | Rating | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | Defense / Speed | Ph Posi |
|---|---|---|---|---|---|---|---|---|---|
| **160** | -0.463246 | -0.763577 | -1.413577 | 0.225810 | -0.026655 | -0.005039 | -1.267805 | 0.958672 | 0 |
| **306** | -1.149744 | 1.153157 | 1.282081 | 0.035573 | 0.196880 | 0.149898 | 0.427195 | -1.119155 | -2 |
| **12** | 1.252998 | 0.194790 | -0.065748 | -0.344902 | 0.420415 | 1.079516 | 0.992195 | 1.194789 | 1 |
| **16** | 1.252998 | 1.153157 | -0.065748 | -0.535140 | -1.889445 | -1.244530 | -2.397806 | 1.525352 | 0 |
| **63** | 0.223251 | 0.194790 | 1.282081 | -2.817990 | 0.494927 | 1.389389 | -0.137805 | 0.061429 | -0 |

## full data scaling

In [28]:
```
#scaler = StandardScaler()

X_full_train[scale_columns] = scaler.fit_transform(X_full_train[scale_columns])

X_full_train.head()
```

Out[28]:

| | Rating | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | Defense / Speed | F Po |
|---|---|---|---|---|---|---|---|---|---|
| **2556** | 0.051639 | -0.049719 | -0.037041 | -0.172038 | 0.471123 | 0.353793 | 0.187485 | 0.811246 | |
| **5208** | -0.831731 | -0.049719 | -0.037041 | -2.784516 | -1.943855 | -1.902212 | -2.439653 | 0.343847 | |
| **2567** | 0.051639 | -0.049719 | -1.420548 | -0.172038 | -1.291159 | 0.180254 | 0.006303 | 0.869671 | |
| **3412** | -0.169203 | 0.687708 | 1.346465 | 0.586423 | 0.601662 | -0.427132 | 0.096894 | -1.350474 | - |
| **2673** | 0.051639 | -0.049719 | -0.037041 | 0.249329 | 0.666932 | 0.353793 | 0.368666 | -0.941500 | - |

## 5. Feature Selection Using RFE

In [29]:
```
logreg = LogisticRegression()
```

### average player rfe

In [30]:
```
rfe_avg = RFE(estimator=logreg, n_features_to_select=15)             # running RFE wit
rfe_avg = rfe_avg.fit(X_avg_train, y_avg_train)
```

In [31]:
```
rfe_avg.support_
```

Out[31]:
```
array([ True, False, False,  True,  True,  True, False,  True,  True,
        True, False, False, False, False,  True, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False,  True, False,  True,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False,  True, False,  True, False, False, False, False, False,
        True, False, False, False, False,  True, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False,  True, False, False, False, False, False, False, False,
       False, False, False, False])
```

In [32]:
```
list(zip(X_avg_train.columns, rfe_avg.support_, rfe_avg.ranking_))
```

Out[32]:
```
[('Rating', True, 1),
 ('Skills_Star', False, 13),
 ('Weak_Foot_Star', False, 10),
 ('Pace / Diving', True, 1),
 ('Shooting / Handling', True, 1),
 ('Passing / Kicking', True, 1),
 ('Dribbling / Reflexes', False, 5),
 ('Defense / Speed', True, 1),
 ('Physical / Positioning', True, 1),
 ('Popularity', True, 1),
 ('Base_Stats', False, 34),
 ('Ingame_Stats', False, 11),
 ('Height_in_cm', False, 7),
 ('BodyType_Weight', False, 4),
 ('Club_Hero', True, 1),
 ('Alt_Pos_Count', False, 16),
 ('Main_Position_CB', False, 31),
 ('Main_Position_CDM', False, 46),
 ('Main_Position_CF', False, 58),
 ('Main_Position_CM', False, 20),
 ('Main_Position_GK', False, 70),
 ('Main_Position_LB', False, 33),
 ('Main_Position_LM', False, 42),
 ('Main_Position_LW', False, 39),
 ('Main_Position_LWB', False, 68),
 ('Main_Position_RB', False, 38),
 ('Main_Position_RM', False, 29),
 ('Main_Position_RW', False, 53),
 ('Main_Position_RWB', False, 67),
 ('Main_Position_ST', False, 26),
 ('Run_Style_Explosive', False, 3),
 ('Run_Style_Lengthy', False, 19),
 ('Attack_Workrate_L', False, 55),
 ('Attack_Workrate_M', True, 1),
 ('Defense_Workrate_L', False, 17),
 ('Defense_Workrate_M', True, 1),
 ('BodyType_Text_CR7', False, 86),
 ('BodyType_Text_Courtois', False, 84),
 ('BodyType_Text_High & Average', False, 14),
 ('BodyType_Text_High & Average+', False, 32),
 ('BodyType_Text_High & Lean', False, 81),
 ('BodyType_Text_High & Stocky', False, 52),
 ('BodyType_Text_Lean', False, 8),
 ('BodyType_Text_Messi', False, 87),
 ('BodyType_Text_R9', False, 88),
 ('BodyType_Text_Ronaldinho', False, 89),
 ('BodyType_Text_Salah', False, 83),
 ('BodyType_Text_Shaqiri', False, 85),
 ('BodyType_Text_Short & Lean', False, 56),
 ('BodyType_Text_Short & Lean-', False, 47),
 ('BodyType_Text_Short and Balanced', False, 72),
 ('BodyType_Text_Stocky', False, 40),
 ('BodyType_Text_Unique', False, 2),
 ('League_Cat_Icons', False, 18),
 ('League_Cat_LaLiga Santander', False, 30),
 ('League_Cat_Ligue 1', True, 1),
 ('League_Cat_Major League Soccer', False, 28),
 ('League_Cat_Other', True, 1),
 ('League_Cat_Premier League', False, 9),
 ('League_Cat_Serie A TIM', False, 25),
```

```
('Nation_Cat_England', False, 22),
('Nation_Cat_France', False, 24),
('Nation_Cat_Germany', False, 21),
('Nation_Cat_Other', True, 1),
('Nation_Cat_Spain', False, 23),
('Alt_Pos_1_CB', False, 54),
('Alt_Pos_1_CDM', False, 50),
('Alt_Pos_1_CF', False, 15),
('Alt_Pos_1_CM', True, 1),
('Alt_Pos_1_LB', False, 64),
('Alt_Pos_1_LM', False, 35),
('Alt_Pos_1_LW', False, 65),
('Alt_Pos_1_LWB', False, 43),
('Alt_Pos_1_RB', False, 41),
('Alt_Pos_1_RM', False, 27),
('Alt_Pos_1_RW', False, 37),
('Alt_Pos_1_RWB', False, 49),
('Alt_Pos_1_ST', False, 60),
('Alt_Pos_1_missing', False, 12),
('Alt_Pos_2_CB', False, 61),
('Alt_Pos_2_CDM', False, 74),
('Alt_Pos_2_CF', False, 51),
('Alt_Pos_2_CM', False, 62),
('Alt_Pos_2_LB', False, 71),
('Alt_Pos_2_LM', False, 48),
('Alt_Pos_2_LW', False, 36),
('Alt_Pos_2_LWB', False, 66),
('Alt_Pos_2_RB', False, 77),
('Alt_Pos_2_RM', False, 59),
('Alt_Pos_2_RW', False, 44),
('Alt_Pos_2_ST', False, 45),
('Alt_Pos_2_missing', True, 1),
('Alt_Pos_3_CDM', False, 76),
('Alt_Pos_3_CF', False, 73),
('Alt_Pos_3_CM', False, 80),
('Alt_Pos_3_LB', False, 82),
('Alt_Pos_3_LM', False, 75),
('Alt_Pos_3_LW', False, 57),
('Alt_Pos_3_LWB', False, 79),
('Alt_Pos_3_RM', False, 78),
('Alt_Pos_3_RW', False, 63),
('Alt_Pos_3_ST', False, 69),
('Alt_Pos_3_missing', False, 6)]
```

In [33]:
```
col_avg = X_avg_train.columns[rfe_avg.support_]
```

In [34]:
```
X_avg_train.columns[~rfe_avg.support_]
```

Out[34]:
```
Index(['Skills_Star', 'Weak_Foot_Star', 'Dribbling / Reflexes', 'Base_Stats',
       'Ingame_Stats', 'Height_in_cm', 'BodyType_Weight', 'Alt_Pos_Count',
       'Main_Position_CB', 'Main_Position_CDM', 'Main_Position_CF',
       'Main_Position_CM', 'Main_Position_GK', 'Main_Position_LB',
       'Main_Position_LM', 'Main_Position_LW', 'Main_Position_LWB',
       'Main_Position_RB', 'Main_Position_RM', 'Main_Position_RW',
       'Main_Position_RWB', 'Main_Position_ST', 'Run_Style_Explosive',
       'Run_Style_Lengthy', 'Attack_Workrate_L', 'Defense_Workrate_L',
       'BodyType_Text_CR7', 'BodyType_Text_Courtois',
       'BodyType_Text_High & Average', 'BodyType_Text_High & Average+',
       'BodyType_Text_High & Lean', 'BodyType_Text_High & Stocky',
       'BodyType_Text_Lean', 'BodyType_Text_Messi', 'BodyType_Text_R9',
       'BodyType_Text_Ronaldinho', 'BodyType_Text_Salah',
       'BodyType_Text_Shaqiri', 'BodyType_Text_Short & Lean',
       'BodyType_Text_Short & Lean-', 'BodyType_Text_Short and Balanced',
       'BodyType_Text_Stocky', 'BodyType_Text_Unique', 'League_Cat_Icons',
       'League_Cat_LaLiga Santander', 'League_Cat_Major League Soccer',
       'League_Cat_Premier League', 'League_Cat_Serie A TIM',
       'Nation_Cat_England', 'Nation_Cat_France', 'Nation_Cat_Germany',
       'Nation_Cat_Spain', 'Alt_Pos_1_CB', 'Alt_Pos_1_CDM', 'Alt_Pos_1_CF',
       'Alt_Pos_1_LB', 'Alt_Pos_1_LM', 'Alt_Pos_1_LW', 'Alt_Pos_1_LWB',
       'Alt_Pos_1_RB', 'Alt_Pos_1_RM', 'Alt_Pos_1_RW', 'Alt_Pos_1_RWB',
       'Alt_Pos_1_ST', 'Alt_Pos_1_missing', 'Alt_Pos_2_CB', 'Alt_Pos_2_CDM',
       'Alt_Pos_2_CF', 'Alt_Pos_2_CM', 'Alt_Pos_2_LB', 'Alt_Pos_2_LM',
       'Alt_Pos_2_LW', 'Alt_Pos_2_LWB', 'Alt_Pos_2_RB', 'Alt_Pos_2_RM',
       'Alt_Pos_2_RW', 'Alt_Pos_2_ST', 'Alt_Pos_3_CDM', 'Alt_Pos_3_CF',
       'Alt_Pos_3_CM', 'Alt_Pos_3_LB', 'Alt_Pos_3_LM', 'Alt_Pos_3_LW',
       'Alt_Pos_3_LWB', 'Alt_Pos_3_RM', 'Alt_Pos_3_RW', 'Alt_Pos_3_ST',
       'Alt_Pos_3_missing'],
      dtype='object')
```

In [35]:
```
col_avg
```

Out[35]:
```
Index(['Rating', 'Pace / Diving', 'Shooting / Handling', 'Passing / Kicking',
       'Defense / Speed', 'Physical / Positioning', 'Popularity', 'Club_Hero',
       'Attack_Workrate_M', 'Defense_Workrate_M', 'League_Cat_Ligue 1',
       'League_Cat_Other', 'Nation_Cat_Other', 'Alt_Pos_1_CM',
       'Alt_Pos_2_missing'],
      dtype='object')
```

## star player rfe

In [36]:
```
rfe_star = RFE(estimator=logreg, n_features_to_select=15)             # running RFE wi
rfe_star = rfe_star.fit(X_star_train, y_star_train)
```

In [37]:
```
rfe_star.support_
```

Out[37]:
```
array([ True,  True,  True,  True,  True,  True, False,  True,  True,
        True,  True, False,  True,  True, False,  True, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False,  True,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False])
```

In [38]:
```
list(zip(X_star_train.columns, rfe_star.support_, rfe_star.ranking_))
```

Out[38]:
```
[('Rating', True, 1),
 ('Skills_Star', True, 1),
 ('Weak_Foot_Star', True, 1),
 ('Pace / Diving', True, 1),
 ('Shooting / Handling', True, 1),
 ('Passing / Kicking', True, 1),
 ('Dribbling / Reflexes', False, 4),
 ('Defense / Speed', True, 1),
 ('Physical / Positioning', True, 1),
 ('Popularity', True, 1),
 ('Base_Stats', True, 1),
 ('Ingame_Stats', False, 9),
 ('Height_in_cm', True, 1),
 ('BodyType_Weight', True, 1),
 ('Club_Hero', False, 2),
 ('Alt_Pos_Count', True, 1),
 ('Main_Position_CB', False, 46),
 ('Main_Position_CDM', False, 64),
 ('Main_Position_CF', False, 18),
 ('Main_Position_CM', False, 22),
 ('Main_Position_GK', False, 74),
 ('Main_Position_LB', False, 24),
 ('Main_Position_LM', False, 61),
 ('Main_Position_LW', False, 23),
 ('Main_Position_LWB', False, 57),
 ('Main_Position_RB', False, 48),
 ('Main_Position_RM', False, 60),
 ('Main_Position_RW', False, 37),
 ('Main_Position_RWB', False, 75),
 ('Main_Position_ST', False, 13),
 ('Run_Style_Explosive', False, 12),
 ('Run_Style_Lengthy', False, 42),
 ('Attack_Workrate_L', False, 50),
 ('Attack_Workrate_M', False, 7),
 ('Defense_Workrate_L', False, 15),
 ('Defense_Workrate_M', True, 1),
 ('BodyType_Text_CR7', False, 78),
 ('BodyType_Text_Courtois', False, 80),
 ('BodyType_Text_High & Average', False, 26),
 ('BodyType_Text_High & Average+', False, 31),
 ('BodyType_Text_High & Lean', False, 58),
 ('BodyType_Text_High & Stocky', False, 83),
 ('BodyType_Text_Lean', False, 10),
 ('BodyType_Text_Messi', False, 59),
 ('BodyType_Text_R9', False, 55),
 ('BodyType_Text_Ronaldinho', False, 53),
 ('BodyType_Text_Salah', False, 65),
 ('BodyType_Text_Shaqiri', False, 81),
 ('BodyType_Text_Short & Lean', False, 69),
 ('BodyType_Text_Short & Lean-', False, 56),
 ('BodyType_Text_Short and Balanced', False, 88),
 ('BodyType_Text_Stocky', False, 47),
 ('BodyType_Text_Unique', False, 3),
 ('League_Cat_Icons', False, 6),
 ('League_Cat_LaLiga Santander', False, 14),
 ('League_Cat_Ligue 1', False, 19),
 ('League_Cat_Major League Soccer', False, 54),
 ('League_Cat_Other', False, 35),
 ('League_Cat_Premier League', False, 5),
 ('League_Cat_Serie A TIM', False, 21),
```

```
('Nation_Cat_England', False, 41),
('Nation_Cat_France', False, 16),
('Nation_Cat_Germany', False, 25),
('Nation_Cat_Other', True, 1),
('Nation_Cat_Spain', False, 30),
('Alt_Pos_1_CB', False, 72),
('Alt_Pos_1_CDM', False, 33),
('Alt_Pos_1_CF', False, 8),
('Alt_Pos_1_CM', False, 38),
('Alt_Pos_1_LB', False, 52),
('Alt_Pos_1_LM', False, 17),
('Alt_Pos_1_LW', False, 87),
('Alt_Pos_1_LWB', False, 36),
('Alt_Pos_1_RB', False, 39),
('Alt_Pos_1_RM', False, 20),
('Alt_Pos_1_RW', False, 86),
('Alt_Pos_1_RWB', False, 43),
('Alt_Pos_1_ST', False, 27),
('Alt_Pos_1_missing', False, 29),
('Alt_Pos_2_CB', False, 68),
('Alt_Pos_2_CDM', False, 76),
('Alt_Pos_2_CF', False, 51),
('Alt_Pos_2_CM', False, 49),
('Alt_Pos_2_LB', False, 89),
('Alt_Pos_2_LM', False, 45),
('Alt_Pos_2_LW', False, 34),
('Alt_Pos_2_LWB', False, 73),
('Alt_Pos_2_RB', False, 71),
('Alt_Pos_2_RM', False, 66),
('Alt_Pos_2_RW', False, 40),
('Alt_Pos_2_ST', False, 32),
('Alt_Pos_2_missing', False, 11),
('Alt_Pos_3_CDM', False, 85),
('Alt_Pos_3_CF', False, 63),
('Alt_Pos_3_CM', False, 77),
('Alt_Pos_3_LB', False, 79),
('Alt_Pos_3_LM', False, 70),
('Alt_Pos_3_LW', False, 44),
('Alt_Pos_3_LWB', False, 82),
('Alt_Pos_3_RM', False, 84),
('Alt_Pos_3_RW', False, 67),
('Alt_Pos_3_ST', False, 62),
('Alt_Pos_3_missing', False, 28)]
```

In [39]:
```
col_star = X_star_train.columns[rfe_star.support_]
```

In [40]:
```
X_star_train.columns[~rfe_star.support_]
```

Out[40]:
```
Index(['Dribbling / Reflexes', 'Ingame_Stats', 'Club_Hero', 'Main_Position_CB',
       'Main_Position_CDM', 'Main_Position_CF', 'Main_Position_CM',
       'Main_Position_GK', 'Main_Position_LB', 'Main_Position_LM',
       'Main_Position_LW', 'Main_Position_LWB', 'Main_Position_RB',
       'Main_Position_RM', 'Main_Position_RW', 'Main_Position_RWB',
       'Main_Position_ST', 'Run_Style_Explosive', 'Run_Style_Lengthy',
       'Attack_Workrate_L', 'Attack_Workrate_M', 'Defense_Workrate_L',
       'BodyType_Text_CR7', 'BodyType_Text_Courtois',
       'BodyType_Text_High & Average', 'BodyType_Text_High & Average+',
       'BodyType_Text_High & Lean', 'BodyType_Text_High & Stocky',
       'BodyType_Text_Lean', 'BodyType_Text_Messi', 'BodyType_Text_R9',
       'BodyType_Text_Ronaldinho', 'BodyType_Text_Salah',
       'BodyType_Text_Shaqiri', 'BodyType_Text_Short & Lean',
       'BodyType_Text_Short & Lean-', 'BodyType_Text_Short and Balanced',
       'BodyType_Text_Stocky', 'BodyType_Text_Unique', 'League_Cat_Icons',
       'League_Cat_LaLiga Santander', 'League_Cat_Ligue 1',
       'League_Cat_Major League Soccer', 'League_Cat_Other',
       'League_Cat_Premier League', 'League_Cat_Serie A TIM',
       'Nation_Cat_England', 'Nation_Cat_France', 'Nation_Cat_Germany',
       'Nation_Cat_Spain', 'Alt_Pos_1_CB', 'Alt_Pos_1_CDM', 'Alt_Pos_1_CF',
       'Alt_Pos_1_CM', 'Alt_Pos_1_LB', 'Alt_Pos_1_LM', 'Alt_Pos_1_LW',
       'Alt_Pos_1_LWB', 'Alt_Pos_1_RB', 'Alt_Pos_1_RM', 'Alt_Pos_1_RW',
       'Alt_Pos_1_RWB', 'Alt_Pos_1_ST', 'Alt_Pos_1_missing', 'Alt_Pos_2_CB',
       'Alt_Pos_2_CDM', 'Alt_Pos_2_CF', 'Alt_Pos_2_CM', 'Alt_Pos_2_LB',
       'Alt_Pos_2_LM', 'Alt_Pos_2_LW', 'Alt_Pos_2_LWB', 'Alt_Pos_2_RB',
       'Alt_Pos_2_RM', 'Alt_Pos_2_RW', 'Alt_Pos_2_ST', 'Alt_Pos_2_missing',
       'Alt_Pos_3_CDM', 'Alt_Pos_3_CF', 'Alt_Pos_3_CM', 'Alt_Pos_3_LB',
       'Alt_Pos_3_LM', 'Alt_Pos_3_LW', 'Alt_Pos_3_LWB', 'Alt_Pos_3_RM',
       'Alt_Pos_3_RW', 'Alt_Pos_3_ST', 'Alt_Pos_3_missing'],
      dtype='object')
```

In [41]:
```
col_star
```

Out[41]:
```
Index(['Rating', 'Skills_Star', 'Weak_Foot_Star', 'Pace / Diving',
       'Shooting / Handling', 'Passing / Kicking', 'Defense / Speed',
       'Physical / Positioning', 'Popularity', 'Base_Stats', 'Height_in_cm',
       'BodyType_Weight', 'Alt_Pos_Count', 'Defense_Workrate_M',
       'Nation_Cat_Other'],
      dtype='object')
```

## full data

In [93]:
```
rfe_full = RFE(estimator=logreg, n_features_to_select=15)             # running RFE wi
rfe_full = rfe_star.fit(X_full_train, y_full_train)
```

it is a very resource heavy operation, took almost 19 mins to run on a pc

In [94]:
```
rfe_full.support_
```

Out[94]:
```
array([ True, False,  True,  True,  True,  True, False,  True,  True,
        True, False, False,  True, False,  True,  True, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False,  True,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False,  True, False,
       False,  True, False, False, False, False, False, False, False,
        True, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False])
```

In [95]:
```
list(zip(X_full_train.columns, rfe_full.support_, rfe_full.ranking_))
```

Out[95]:
```
[('Rating', True, 1),
 ('Skills_Star', False, 8),
 ('Weak_Foot_Star', True, 1),
 ('Pace / Diving', True, 1),
 ('Shooting / Handling', True, 1),
 ('Passing / Kicking', True, 1),
 ('Dribbling / Reflexes', False, 6),
 ('Defense / Speed', True, 1),
 ('Physical / Positioning', True, 1),
 ('Popularity', True, 1),
 ('Base_Stats', False, 31),
 ('Ingame_Stats', False, 14),
 ('Height_in_cm', True, 1),
 ('BodyType_Weight', False, 3),
 ('Club_Hero', True, 1),
 ('Alt_Pos_Count', True, 1),
 ('Main_Position_CB', False, 36),
 ('Main_Position_CDM', False, 40),
 ('Main_Position_CF', False, 52),
 ('Main_Position_CM', False, 22),
 ('Main_Position_GK', False, 69),
 ('Main_Position_LB', False, 34),
 ('Main_Position_LM', False, 46),
 ('Main_Position_LW', False, 37),
 ('Main_Position_LWB', False, 68),
 ('Main_Position_RB', False, 47),
 ('Main_Position_RM', False, 32),
 ('Main_Position_RW', False, 49),
 ('Main_Position_RWB', False, 65),
 ('Main_Position_ST', False, 15),
 ('Run_Style_Explosive', False, 12),
 ('Run_Style_Lengthy', False, 20),
 ('Attack_Workrate_L', False, 53),
 ('Attack_Workrate_M', False, 2),
 ('Defense_Workrate_L', False, 26),
 ('Defense_Workrate_M', True, 1),
 ('BodyType_Text_CR7', False, 89),
 ('BodyType_Text_Courtois', False, 86),
 ('BodyType_Text_High & Average', False, 18),
 ('BodyType_Text_High & Average+', False, 35),
 ('BodyType_Text_High & Lean', False, 80),
 ('BodyType_Text_High & Stocky', False, 55),
 ('BodyType_Text_Lean', False, 11),
 ('BodyType_Text_Messi', False, 82),
 ('BodyType_Text_R9', False, 85),
 ('BodyType_Text_Ronaldinho', False, 84),
 ('BodyType_Text_Salah', False, 81),
 ('BodyType_Text_Shaqiri', False, 88),
 ('BodyType_Text_Short & Lean', False, 56),
 ('BodyType_Text_Short & Lean-', False, 44),
 ('BodyType_Text_Short and Balanced', False, 70),
 ('BodyType_Text_Stocky', False, 45),
 ('BodyType_Text_Unique', True, 1),
 ('League_Cat_Icons', False, 9),
 ('League_Cat_LaLiga Santander', False, 28),
 ('League_Cat_Ligue 1', True, 1),
 ('League_Cat_Major League Soccer', False, 33),
 ('League_Cat_Other', False, 4),
 ('League_Cat_Premier League', False, 5),
 ('League_Cat_Serie A TIM', False, 21),
```

```
('Nation_Cat_England', False, 19),
('Nation_Cat_France', False, 17),
('Nation_Cat_Germany', False, 25),
('Nation_Cat_Other', True, 1),
('Nation_Cat_Spain', False, 24),
('Alt_Pos_1_CB', False, 59),
('Alt_Pos_1_CDM', False, 39),
('Alt_Pos_1_CF', False, 27),
('Alt_Pos_1_CM', False, 13),
('Alt_Pos_1_LB', False, 58),
('Alt_Pos_1_LM', False, 29),
('Alt_Pos_1_LW', False, 66),
('Alt_Pos_1_LWB', False, 42),
('Alt_Pos_1_RB', False, 41),
('Alt_Pos_1_RM', False, 23),
('Alt_Pos_1_RW', False, 43),
('Alt_Pos_1_RWB', False, 54),
('Alt_Pos_1_ST', False, 60),
('Alt_Pos_1_missing', False, 16),
('Alt_Pos_2_CB', False, 63),
('Alt_Pos_2_CDM', False, 75),
('Alt_Pos_2_CF', False, 51),
('Alt_Pos_2_CM', False, 61),
('Alt_Pos_2_LB', False, 73),
('Alt_Pos_2_LM', False, 50),
('Alt_Pos_2_LW', False, 30),
('Alt_Pos_2_LWB', False, 72),
('Alt_Pos_2_RB', False, 79),
('Alt_Pos_2_RM', False, 62),
('Alt_Pos_2_RW', False, 38),
('Alt_Pos_2_ST', False, 48),
('Alt_Pos_2_missing', False, 10),
('Alt_Pos_3_CDM', False, 76),
('Alt_Pos_3_CF', False, 71),
('Alt_Pos_3_CM', False, 83),
('Alt_Pos_3_LB', False, 87),
('Alt_Pos_3_LM', False, 74),
('Alt_Pos_3_LW', False, 57),
('Alt_Pos_3_LWB', False, 77),
('Alt_Pos_3_RM', False, 78),
('Alt_Pos_3_RW', False, 67),
('Alt_Pos_3_ST', False, 64),
('Alt_Pos_3_missing', False, 7)]
```

In [96]:
```
col_full = X_full_train.columns[rfe_full.support_]
```

In [97]:
```
X_full_train.columns[~rfe_full.support_]
```

Out[97]:
```
Index(['Skills_Star', 'Dribbling / Reflexes', 'Base_Stats', 'Ingame_Stats',
       'BodyType_Weight', 'Main_Position_CB', 'Main_Position_CDM',
       'Main_Position_CF', 'Main_Position_CM', 'Main_Position_GK',
       'Main_Position_LB', 'Main_Position_LM', 'Main_Position_LW',
       'Main_Position_LWB', 'Main_Position_RB', 'Main_Position_RM',
       'Main_Position_RW', 'Main_Position_RWB', 'Main_Position_ST',
       'Run_Style_Explosive', 'Run_Style_Lengthy', 'Attack_Workrate_L',
       'Attack_Workrate_M', 'Defense_Workrate_L', 'BodyType_Text_CR7',
       'BodyType_Text_Courtois', 'BodyType_Text_High & Average',
       'BodyType_Text_High & Average+', 'BodyType_Text_High & Lean',
       'BodyType_Text_High & Stocky', 'BodyType_Text_Lean',
       'BodyType_Text_Messi', 'BodyType_Text_R9', 'BodyType_Text_Ronaldinho',
       'BodyType_Text_Salah', 'BodyType_Text_Shaqiri',
       'BodyType_Text_Short & Lean', 'BodyType_Text_Short & Lean-',
       'BodyType_Text_Short and Balanced', 'BodyType_Text_Stocky',
       'League_Cat_Icons', 'League_Cat_LaLiga Santander',
       'League_Cat_Major League Soccer', 'League_Cat_Other',
       'League_Cat_Premier League', 'League_Cat_Serie A TIM',
       'Nation_Cat_England', 'Nation_Cat_France', 'Nation_Cat_Germany',
       'Nation_Cat_Spain', 'Alt_Pos_1_CB', 'Alt_Pos_1_CDM', 'Alt_Pos_1_CF',
       'Alt_Pos_1_CM', 'Alt_Pos_1_LB', 'Alt_Pos_1_LM', 'Alt_Pos_1_LW',
       'Alt_Pos_1_LWB', 'Alt_Pos_1_RB', 'Alt_Pos_1_RM', 'Alt_Pos_1_RW',
       'Alt_Pos_1_RWB', 'Alt_Pos_1_ST', 'Alt_Pos_1_missing', 'Alt_Pos_2_CB',
       'Alt_Pos_2_CDM', 'Alt_Pos_2_CF', 'Alt_Pos_2_CM', 'Alt_Pos_2_LB',
       'Alt_Pos_2_LM', 'Alt_Pos_2_LW', 'Alt_Pos_2_LWB', 'Alt_Pos_2_RB',
       'Alt_Pos_2_RM', 'Alt_Pos_2_RW', 'Alt_Pos_2_ST', 'Alt_Pos_2_missing',
       'Alt_Pos_3_CDM', 'Alt_Pos_3_CF', 'Alt_Pos_3_CM', 'Alt_Pos_3_LB',
       'Alt_Pos_3_LM', 'Alt_Pos_3_LW', 'Alt_Pos_3_LWB', 'Alt_Pos_3_RM',
       'Alt_Pos_3_RW', 'Alt_Pos_3_ST', 'Alt_Pos_3_missing'],
      dtype='object')
```

In [98]:
```
col_full
```

Out[98]:
```
Index(['Rating', 'Weak_Foot_Star', 'Pace / Diving', 'Shooting / Handling',
       'Passing / Kicking', 'Defense / Speed', 'Physical / Positioning',
       'Popularity', 'Height_in_cm', 'Club_Hero', 'Alt_Pos_Count',
       'Defense_Workrate_M', 'BodyType_Text_Unique', 'League_Cat_Ligue 1',
       'Nation_Cat_Other'],
      dtype='object')
```

## After RFE

As a first step, I run an RFE, automatic feature selection, in the dataset, we have 103 columns (105, minus the name and the price). From these I wanted to select the 15 most important ones.

In [42]:
```
col_avg
```

Out[42]:
```
Index(['Rating', 'Pace / Diving', 'Shooting / Handling', 'Passing / Kicking',
       'Defense / Speed', 'Physical / Positioning', 'Popularity', 'Club_Hero',
       'Attack_Workrate_M', 'Defense_Workrate_M', 'League_Cat_Ligue 1',
       'League_Cat_Other', 'Nation_Cat_Other', 'Alt_Pos_1_CM',
       'Alt_Pos_2_missing'],
      dtype='object')
```

In [43]:
```
col_star
```

Out[43]:
```
Index(['Rating', 'Skills_Star', 'Weak_Foot_Star', 'Pace / Diving',
       'Shooting / Handling', 'Passing / Kicking', 'Defense / Speed',
       'Physical / Positioning', 'Popularity', 'Base_Stats', 'Height_in_cm',
       'BodyType_Weight', 'Alt_Pos_Count', 'Defense_Workrate_M',
       'Nation_Cat_Other'],
      dtype='object')
```

In [100…
```
col_full
```

Out[100]:
```
Index(['Rating', 'Weak_Foot_Star', 'Pace / Diving', 'Shooting / Handling',
       'Passing / Kicking', 'Defense / Speed', 'Physical / Positioning',
       'Popularity', 'Height_in_cm', 'Club_Hero', 'Alt_Pos_Count',
       'Defense_Workrate_M', 'BodyType_Text_Unique', 'League_Cat_Ligue 1',
       'Nation_Cat_Other'],
      dtype='object')
```

## 6. Building model using statsmodel, for the detailed statistics

### first I build a model for the average players, and then optimizing it by removeing the least significant features (iterativly)

In [44]:
```
def linear_model_avg(columns):
    # Creating X_train dataframe with RFE selected variables
    rfe = X_avg_train[columns]
    # Adding a constant variable
    rfe = sm.add_constant(rfe)
    # Running the linear model
    lm = sm.OLS(y_avg_train,rfe).fit()
    #Let's see the summary of our linear model
    print(lm.summary())
```

https://medium.com/swlh/interpreting-linear-regression-through-statsmodels-summary-4796d359035a

In [45]:
```
#linear_model_avg(col_avg)
```

Nation_Cat_Other is the least significant feature (0.770 P>|t|). -> drop the feature

In [46]:
```
col_avg = col_avg.drop('Nation_Cat_Other')
```

In [47]:
```
#col_avg
```

In [48]:
```
#linear_model_avg(col_avg)
```

In [49]:
```
col_avg = col_avg.drop('Defense / Speed')
```

In [50]:
```
#linear_model_avg(col_avg)
```

In [51]:
```
col_avg = col_avg.drop('Shooting / Handling')
```

In [52]:
```
#linear_model_avg(col_avg)
```

In [53]:
```
col_avg = col_avg.drop('Alt_Pos_1_CM')
```

In [54]:
```
#linear_model_avg(col_avg)
```

In [55]:
```
col_avg = col_avg.drop('Alt_Pos_2_missing')
```

In [56]:
```
#linear_model_avg(col_avg)
```

In [57]:
```
col_avg = col_avg.drop('Defense_Workrate_M')
```

In [58]:
```
#linear_model_avg(col_avg)
```

In [59]:
```
col_avg = col_avg.drop('League_Cat_Other')
```

In [60]:
```
#linear_model_avg(col_avg)
```

In [61]:
```
col_avg = col_avg.drop('Physical / Positioning')
```

In [62]:
```
#linear_model_avg(col_avg)
```

In [63]:
```
col_avg = col_avg.drop('Passing / Kicking')
```

keep dropping the least significant column from the table

In [64]:
```
linear_model_avg(col_avg)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.591
Model:                            OLS   Adj. R-squared:                  0.591
Method:                 Least Squares   F-statistic:                     1201.
Date:                Tue, 15 Aug 2023   Prob (F-statistic):               0.00
Time:                        11:10:20   Log-Likelihood:                -51357.
No. Observations:                4994   AIC:                         1.027e+05
Df Residuals:                    4987   BIC:                         1.028e+05
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       2238.0164    168.093     13.314      0.000    1908.479    2567.554
Rating      2963.0985    119.764     24.741      0.000    2728.309    3197.888
Pace / Diving  798.1322    111.530      7.156      0.000     579.484    1016.780
Popularity  2684.7737    107.148     25.057      0.000    2474.717    2894.831
Club_Hero    4.12e+04    806.121     51.113      0.000    3.96e+04    4.28e+04
Attack_Workrate_M   636.7730    215.181      2.959      0.003     214.924    1058.621
League_Cat_Ligue 1  1566.8349    539.872      2.902      0.004     508.449    2625.221
==============================================================================
Omnibus:                     4223.690   Durbin-Watson:                   2.020
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           315598.807
Skew:                           3.599   Prob(JB):                         0.00
Kurtosis:                      41.274   Cond. No.                         10.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

## let's try the same for the star players

In [65]:
```
def linear_model_star(columns):
    # Creating X_train dataframe with RFE selected variables
    rfe = X_star_train[columns]
    # Adding a constant variable
    rfe = sm.add_constant(rfe)
    # Running the linear model
    lm = sm.OLS(y_star_train,rfe).fit()
    #Let's see the summary of our linear model
    print(lm.summary())
```

In [66]:
```
#linear_model_star(col_star)
```

In [67]:
```
col_star = col_star.drop('Physical / Positioning')
```

In [68]:
```
#linear_model_star(col_star)
```

In [69]:
```
col_star = col_star.drop('BodyType_Weight')
```

In [70]:
```
#linear_model_star(col_star)
```

In [71]:
```
col_star = col_star.drop('Alt_Pos_Count')
```

In [72]:
```
#linear_model_star(col_star)
```

In [73]:
```
col_star = col_star.drop('Pace / Diving')
```

In [74]:
```
#linear_model_star(col_star)
```

In [75]:
```
col_star = col_star.drop('Weak_Foot_Star')
```

In [76]:
```
#linear_model_star(col_star)
```

In [77]:
```
col_star = col_star.drop('Nation_Cat_Other')
```

In [78]:
```
#linear_model_star(col_star)
```

In [79]:
```
col_star = col_star.drop('Height_in_cm')
```

In [80]:
```
#linear_model_star(col_star)
```

In [81]:
```
col_star = col_star.drop('Defense_Workrate_M')
```

In [82]:
```
#linear_model_star(col_star)
```

In [83]:
```
col_star = col_star.drop('Popularity')
```

In [84]:
```
#linear_model_star(col_star)
```

In [85]:
```
col_star = col_star.drop('Shooting / Handling')
```

In [86]:
```
#linear_model_star(col_star)
```

In [87]:
```
col_star = col_star.drop('Base_Stats')
```

In [88]:
```
#linear_model_star(col_star)
```

In [89]:
```
col_star = col_star.drop('Defense / Speed')
```

In [90]:
```
#linear_model_star(col_star)
```

In [91]:
```
col_star = col_star.drop('Passing / Kicking')
```

In [92]:
```
linear_model_star(col_star)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.274
Model:                            OLS   Adj. R-squared:                  0.262
Method:                 Least Squares   F-statistic:                     22.67
Date:                Tue, 15 Aug 2023   Prob (F-statistic):           4.46e-09
Time:                        11:10:22   Log-Likelihood:                -1941.7
No. Observations:                 123   AIC:                             3889.
Df Residuals:                     120   BIC:                             3898.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        9.507e+05   1.59e+05      5.996      0.000    6.37e+05    1.26e+06
Rating        8.82e+05   1.63e+05      5.411      0.000    5.59e+05     1.2e+06
Skills_Star  4.307e+05   1.63e+05      2.642      0.009    1.08e+05    7.53e+05
==============================================================================
Omnibus:                      145.810   Durbin-Watson:                   2.003
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3419.406
Skew:                           4.310   Prob(JB):                         0.00
Kurtosis:                      27.350   Cond. No.                         1.27
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
```

## checking the same on the original database

In [134…
```
def linear_model_full(columns):
    # Creating X_train dataframe with RFE selected variables
    rfe = X_full_train[columns]
    # Adding a constant variable
    rfe = sm.add_constant(rfe)
    # Running the linear model
    lm = sm.OLS(y_full_train,rfe).fit()
    #Let's see the summary of our linear model
    print(lm.summary())
```

In [107…
```
#linear_model_full(col_full)
```

In [106…
```
col_full = col_full.drop('Pace / Diving')
```

In [110…
```
#linear_model_full(col_full)
```

In [109…
```
col_full = col_full.drop('Height_in_cm')
```

In [113…
```
#linear_model_full(col_full)
```

In [112…
```
col_full = col_full.drop('Nation_Cat_Other')
```

In [116…
```
#linear_model_full(col_full)
```

In [115…
```
col_full = col_full.drop('Defense_Workrate_M')
```

In [119…
```
#linear_model_full(col_full)
```

In [118…
```
col_full = col_full.drop('Rating')
```

In [122…
```
#linear_model_full(col_full)
```

In [121…
```
col_full = col_full.drop('Alt_Pos_Count')
```

In [125…
```
#linear_model_full(col_full)
```

In [126…
```
col_full = col_full.drop('Shooting / Handling')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Input In [126], in <cell line: 1>()
----> 1 col_full = col_full.drop('Shooting / Handling')

File C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py:6644, in
Index.drop(self, labels, errors)
   6642 if mask.any():
   6643     if errors != "ignore":
-> 6644         raise KeyError(f"{list(labels[mask])} not found in axis")
   6645     indexer = indexer[~mask]
   6646 return self.delete(indexer)

KeyError: "['Shooting / Handling'] not found in axis"
```

In [129…
```
#linear_model_full(col_full)
```

In [128…
```
col_full = col_full.drop('Passing / Kicking')
```

In [172…
```
linear_model_full(col_full)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.103
Model:                            OLS   Adj. R-squared:                  0.102
Method:                 Least Squares   F-statistic:                     84.05
Date:                Tue, 15 Aug 2023   Prob (F-statistic):          4.10e-116
Time:                        17:22:35   Log-Likelihood:                -72195.
No. Observations:                5117   AIC:                         1.444e+05
Df Residuals:                    5109   BIC:                         1.445e+05
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
                               coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
const                     8585.8419   4767.239      1.801      0.072    -759.989    1.7
9e+04
Weak_Foot_Star            1.292e+04   4709.215      2.743      0.006    3686.113    2.2
2e+04
Defense / Speed          -1.585e+04   5519.660     -2.871      0.004   -2.67e+04   -502
7.140
Physical / Positioning    1.461e+04   5750.312      2.540      0.011    3332.707    2.5
9e+04
Popularity                5.908e+04   4926.639     11.992      0.000    4.94e+04    6.8
7e+04
Club_Hero                 3.325e+05   2.96e+04     11.249      0.000    2.75e+05     3.
9e+05
BodyType_Text_Unique      2.104e+05   3.24e+04      6.496      0.000    1.47e+05    2.7
4e+05
League_Cat_Ligue 1        9.868e+04   2.32e+04      4.245      0.000    5.31e+04    1.4
4e+05
==============================================================================
Omnibus:                    13219.783   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        250455743.874
Skew:                          29.352   Prob(JB):                         0.00
Kurtosis:                    1085.245   Cond. No.                         9.31
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
```

In [ ]:
```
making predictions with the final model
```

In [ ]:
```
model evaluation
```

## 7. Residual Analysis of the train data (avg)

So, now to check if the error terms are also normally distributed (which is infact, one of the major assumptions of linear regression), let us plot the histogram of the error terms and see what it looks like.

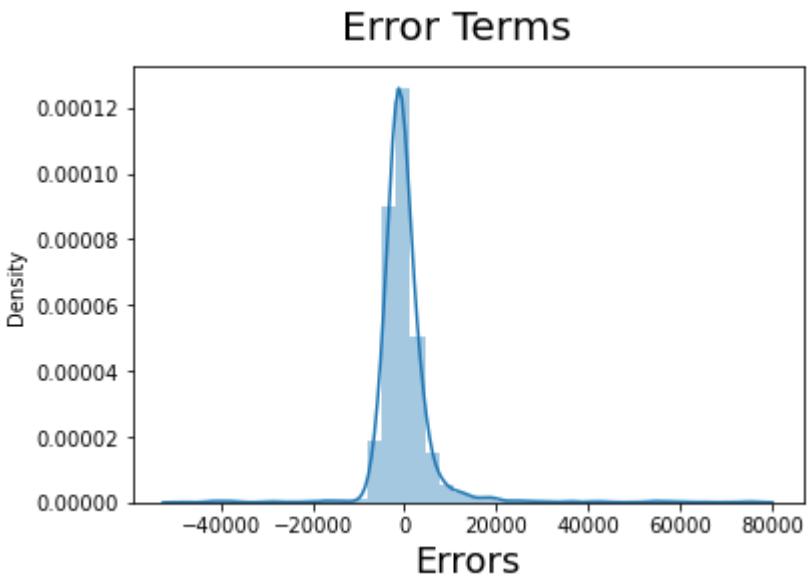
In [173…
```
rfe = X_avg_train[col_avg]
# Adding a constant variable
rfe = sm.add_constant(rfe)
```

```
# Running the linear model
lm = sm.OLS(y_avg_train,rfe).fit()
```

In [174…
```
y_avg_train_price = lm.predict(rfe)
```

In [177…
```
# Plot the histogram of the error terms
fig = plt.figure()
sns.distplot((y_avg_train - y_avg_train_price), bins = 40)
fig.suptitle('Error Terms', fontsize = 20)                  # Plot heading
plt.xlabel('Errors', fontsize = 18)                         # X-label
```

Out[177]:
```
Text(0.5, 0, 'Errors')
```




## 8. Making Predictions Using the Final Model

In [178…
```
X_avg_test[scale_columns] = scaler.transform(X_avg_test[scale_columns])

X_avg_test.head()
```

Out[178]:

| | Rating | Skills_Star | Weak_Foot_Star | Pace / Diving | Shooting / Handling | Passing / Kicking | Dribbling / Reflexes | Defense / Speed | Po |
|---|---|---|---|---|---|---|---|---|---|
| **5306** | -0.831731 | -0.049719 | 1.346465 | 0.080782 | -1.030080 | -0.253593 | -0.809016 | 0.226997 | - |
| **2583** | 0.051639 | 0.687708 | -0.037041 | 0.417876 | 0.536393 | 0.614101 | 0.368666 | 0.928095 | |
| **3561** | -0.279625 | -1.524573 | -0.037041 | -0.509132 | 0.797471 | 0.180254 | 0.187485 | 0.168572 | - |
| **353** | 2.039222 | 0.687708 | 1.346465 | 1.850525 | 1.972326 | 1.915642 | 1.636940 | -0.590951 | |
| **5499** | -0.942152 | 0.687708 | -0.037041 | -0.424859 | 0.079505 | -1.555134 | -0.627834 | -1.759448 | - |

In [179…
```
# Adding constant variable to test dataframe
rfe = X_avg_test[col_avg]
rfe = sm.add_constant(rfe)
```

In [180…
```
# Making predictions

y_avg_pred = lm.predict(rfe)
```

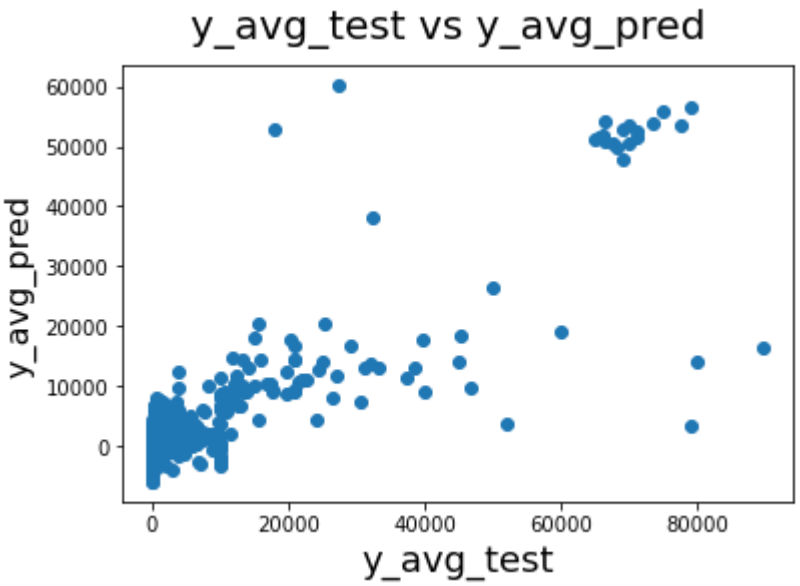
## 9. Model Evaluation

Let's now plot the graph for actual versus predicted values.

In [181…
```
# Plotting y_avg_test and y_avg_pred to understand the spread

fig = plt.figure()
plt.scatter(y_avg_test, y_avg_pred)
fig.suptitle('y_avg_test vs y_avg_pred', fontsize = 20)              # Plot heading
plt.xlabel('y_avg_test', fontsize = 18)                             # X-Label
plt.ylabel('y_avg_pred', fontsize = 16)
```

Out[181]:
```
Text(0, 0.5, 'y_avg_pred')
```




In [166…
```
from sklearn.metrics import r2_score
```

In [182…
```
r2_score(y_avg_test, y_avg_pred)
```

Out[182]:
```
0.6515615396141419
```