Multiple Linear Regression

Accessing the data

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [11]:
```
ipl = pd.read_csv('https://raw.githubusercontent.com/Foridur3210/IPL-Dataset-Player-price-prediction/master/IPL%20IMB381IPL2013.c
```

In [12]:
```
ipl
```

Out[12]:

	Sl.NO.	PLAYER NAME	AGE	COUNTRY	TEAM	PLAYING ROLE	T-RUNS	T-WKTS	ODI-RUNS-S	ODI-SR-B	...	SR-B	SIXERS	RUNS-C	WKTS	AVE-BL	ECON	SR-BL	AUCTION YEAR
0	1	Abdulla, YA	2	SA	KXIP	Allrounder	0	0	0	0.00	...	0.00	0	307	15	20.47	8.90	13.93	2009
1	2	Abdur Razzak	2	BAN	RCB	Bowler	214	18	657	71.41	...	0.00	0	29	0	0.00	14.50	0.00	2008
2	3	Agarkar, AB	2	IND	KKR	Bowler	571	58	1269	80.62	...	121.01	5	1059	29	36.52	8.81	24.90	2008
3	4	Ashwin, R	1	IND	CSK	Bowler	284	31	241	84.56	...	76.32	0	1125	49	22.96	6.23	22.14	2011
4	5	Badrinath, S	2	IND	CSK	Batsman	63	0	79	45.93	...	120.71	28	0	0	0.00	0.00	0.00	2011
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
125	126	Yadav, AS	2	IND	DC	Batsman	0	0	0	0.00	...	125.64	2	0	0	0.00	0.00	0.00	2010
126	127	Younis Khan	2	PAK	RR	Batsman	6398	7	6814	75.78	...	42.85	0	0	0	0.00	0.00	0.00	2008
127	128	Yuvraj Singh	2	IND	KXIP+	Batsman	1775	9	8051	87.58	...	131.88	67	569	23	24.74	7.02	21.13	2011
128	129	Zaheer Khan	2	IND	MI+	Bowler	1114	288	790	73.55	...	91.67	1	1783	65	27.43	7.75	21.26	2008
129	130	Zoysa, DNT	2	SL	DC	Bowler	288	64	343	95.81	...	122.22	0	99	2	49.50	9.00	33.00	2008

130 rows × 26 columns

In [13]:
```
ipl.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 130 entries, 0 to 129
Data columns (total 26 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Sl.NO.         130 non-null    int64  
 1   PLAYER NAME    130 non-null    object 
 2   AGE            130 non-null    int64  
 3   COUNTRY        130 non-null    object 
 4   TEAM           130 non-null    object 
 5   PLAYING ROLE   130 non-null    object 
 6   T-RUNS         130 non-null    int64  
 7   T-WKTS         130 non-null    int64  
 8   ODI-RUNS-S     130 non-null    int64  
 9   ODI-SR-B       130 non-null    float64
 10  ODI-WKTS       130 non-null    int64  
 11  ODI-SR-BL      130 non-null    float64
 12  CAPTAINCY EXP  130 non-null    int64  
 13  RUNS-S         130 non-null    int64  
 14  HS             130 non-null    int64  
 15  AVE            130 non-null    float64
 16  SR-B           130 non-null    float64
 17  SIXERS         130 non-null    int64  
 18  RUNS-C         130 non-null    int64  
 19  WKTS           130 non-null    int64  
 20  AVE-BL         130 non-null    float64
 21  ECON           130 non-null    float64
 22  SR-BL          130 non-null    float64
 23  AUCTION YEAR   130 non-null    int64  
 24  BASE PRICE     130 non-null    int64  
 25  SOLD PRICE     130 non-null    int64  
dtypes: float64(7), int64(15), object(4)
memory usage: 26.5+ KB
```

Data Preprocessing

In [14]:
```
ipl.iloc[0:10, 0:15]
```

Out[14]:

	Sl.NO.	PLAYER NAME	AGE	COUNTRY	TEAM	PLAYING ROLE	T-RUNS	T-WKTS	ODI-RUNS-S	ODI-SR-B	ODI-WKTS	ODI-SR-BL	CAPTAINCY EXP	RUNS-S	HS
0	1	Abdulla, YA	2	SA	KXIP	Allrounder	0	0	0	0.00	0	0.0	0	0	0
1	2	Abdur Razzak	2	BAN	RCB	Bowler	214	18	657	71.41	185	37.6	0	0	0
2	3	Agarkar, AB	2	IND	KKR	Bowler	571	58	1269	80.62	288	32.9	0	167	39
3	4	Ashwin, R	1	IND	CSK	Bowler	284	31	241	84.56	51	36.8	0	58	11
4	5	Badrinath, S	2	IND	CSK	Batsman	63	0	79	45.93	0	0.0	0	1317	71
5	6	Bailey, GJ	2	AUS	CSK	Batsman	0	0	172	72.26	0	0.0	1	63	48
6	7	Balaji, L	2	IND	CSK+	Bowler	51	27	120	78.94	34	42.5	0	26	15
7	8	Bollinger, DE	2	AUS	CSK	Bowler	54	50	50	92.59	62	31.3	0	21	16
8	9	Botha, J	2	SA	RR	Allrounder	83	17	609	85.77	72	53.0	1	335	67
9	10	Boucher, MV	2	SA	RCB+	W. Keeper	5515	1	4686	84.76	0	0.0	1	394	50

In [15]:
```
ipl.iloc[0:10, 15:]
```

Out[15]:

	AVE	SR-B	SIXERS	RUNS-C	WKTS	AVE-BL	ECON	SR-BL	AUCTION YEAR	BASE PRICE	SOLD PRICE
0	0.00	0.00	0	307	15	20.47	8.90	13.93	2009	50000	50000
1	0.00	0.00	0	29	0	0.00	14.50	0.00	2008	50000	50000
2	18.56	121.01	5	1059	29	36.52	8.81	24.90	2008	200000	350000
3	5.80	76.32	0	1125	49	22.96	6.23	22.14	2011	100000	850000
4	32.93	120.71	28	0	0	0.00	0.00	0.00	2011	100000	800000
5	21.00	95.45	0	0	0	0.00	0.00	0.00	2009	50000	50000
6	4.33	72.22	1	1342	52	25.81	7.98	19.40	2011	100000	500000
7	21.00	165.88	1	693	37	18.73	7.22	15.57	2011	200000	700000
8	30.45	114.73	3	610	19	32.11	6.85	28.11	2011	200000	950000
9	28.14	127.51	13	0	0	0.00	0.00	0.00	2008	200000	450000

In [16]:
```
# Target

y = ipl['SOLD PRICE']
y
```

Out[16]:
```
0        50000
1        50000
2       350000
3       850000
4       800000
         ...  
125     750000
126     225000
127    1800000
128     450000
129     110000
Name: SOLD PRICE, Length: 130, dtype: int64
```

In [17]:
```
# Features

X = ipl.drop(['SOLD PRICE'], axis = 1)
```

In [22]:
```
# Drop irrelevant columns

X_1 = X.drop(['Sl.NO.', 'PLAYER NAME', 'TEAM'], axis = 1)
```

In [23]:
```
X_1.head()
```

Out[23]:

	AGE	COUNTRY	PLAYING ROLE	T-RUNS	T-WKTS	ODI-RUNS-S	ODI-SR-B	ODI-WKTS	ODI-SR-BL	CAPTAINCY EXP	...	AVE	SR-B	SIXERS	RUNS-C	WKTS	AVE-BL	ECON	SR-BL	AUCTIO YEA
0	2	SA	Allrounder	0	0	0	0.00	0	0.0	0	...	0.00	0.00	0	307	15	20.47	8.90	13.93	20
1	2	BAN	Bowler	214	18	657	71.41	185	37.6	0	...	0.00	0.00	0	29	0	0.00	14.50	0.00	20
2	2	IND	Bowler	571	58	1269	80.62	288	32.9	0	...	18.56	121.01	5	1059	29	36.52	8.81	24.90	20
3	1	IND	Bowler	284	31	241	84.56	51	36.8	0	...	5.80	76.32	0	1125	49	22.96	6.23	22.14	20
4	2	IND	Batsman	63	0	79	45.93	0	0.0	0	...	32.93	120.71	28	0	0	0.00	0.00	0.00	20

5 rows × 22 columns

In [24]:
```
X['AGE'].unique()
```

Out[24]:
```
array([2, 1, 3], dtype=int64)
```

In [25]:
```
X['COUNTRY']
```

Out[25]:
```
0       SA
1      BAN
2      IND
3      IND
4      IND
      ... 
125    IND
126    PAK
127    IND
128    IND
129     SL
Name: COUNTRY, Length: 130, dtype: object
```

In [26]:
```
X['PLAYING ROLE'].unique()
```

Out[26]:
```
array(['Allrounder', 'Bowler', 'Batsman', 'W. Keeper'], dtype=object)
```

In [27]:
```
X['CAPTAINCY EXP'].unique()
```

Out[27]:
```
array([0, 1], dtype=int64)
```

In [28]:
```
# Convert categorical variables to numeric using One hot encoding

X_1 = pd.get_dummies(X_1, columns = ['AGE', 'COUNTRY', 'PLAYING ROLE', 'CAPTAINCY EXP'])
```

In [29]:
```
X_1
```

Out[29]:

	T-RUNS	T-WKTS	ODI-RUNS-S	ODI-SR-B	ODI-WKTS	ODI-SR-BL	RUNS-S	HS	AVE	SR-B	...	COUNTRY_SA	COUNTRY_SL	COUNTRY_WI	COUNTRY_ZIM	PLAYING ROLE_Allrounder
0	0	0	0	0.00	0	0.0	0	0	0.00	0.00	...	1	0	0	0	1
1	214	18	657	71.41	185	37.6	0	0	0.00	0.00	...	0	0	0	0	0
2	571	58	1269	80.62	288	32.9	167	39	18.56	121.01	...	0	0	0	0	0
3	284	31	241	84.56	51	36.8	58	11	5.80	76.32	...	0	0	0	0	0
4	63	0	79	45.93	0	0.0	1317	71	32.93	120.71	...	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
125	0	0	0	0.00	0	0.0	49	16	9.80	125.64	...	0	0	0	0	0
126	6398	7	6814	75.78	3	86.6	3	3	3.00	42.85	...	0	0	0	0	0
127	1775	9	8051	87.58	109	44.3	1237	66	26.32	131.88	...	0	0	0	0	0
128	1114	288	790	73.55	278	35.4	99	23	9.90	91.67	...	0	0	0	0	0
129	288	64	343	95.81	108	39.4	11	10	11.00	122.22	...	0	1	0	0	0

130 rows × 37 columns

In [30]:
```
y
```

Out[30]:
```
0        50000
1        50000
2       350000
3       850000
4       800000
          ...
125     750000
126     225000
127    1800000
128     450000
129     110000
Name: SOLD PRICE, Length: 130, dtype: int64
```

In [31]:
```
import statsmodels.api as sm
X_1 = sm.add_constant(X_1)
X_1
```

```
C:\Users\Urvi Sharma\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWarning: In a future version of pandas
all arguments of concat except for the argument 'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```

Out[31]:

	const	T-RUNS	T-WKTS	ODI-RUNS-S	ODI-SR-B	ODI-WKTS	ODI-SR-BL	RUNS-S	HS	AVE	...	COUNTRY_SA	COUNTRY_SL	COUNTRY_WI	COUNTRY_ZIM	PLAYING ROLE_Allrounder	F
0	1.0	0	0	0	0.00	0	0.0	0	0	0.00	...	1	0	0	0	1	
1	1.0	214	18	657	71.41	185	37.6	0	0	0.00	...	0	0	0	0	0	
2	1.0	571	58	1269	80.62	288	32.9	167	39	18.56	...	0	0	0	0	0	
3	1.0	284	31	241	84.56	51	36.8	58	11	5.80	...	0	0	0	0	0	
4	1.0	63	0	79	45.93	0	0.0	1317	71	32.93	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
125	1.0	0	0	0	0.00	0	0.0	49	16	9.80	...	0	0	0	0	0	
126	1.0	6398	7	6814	75.78	3	86.6	3	3	3.00	...	0	0	0	0	0	
127	1.0	1775	9	8051	87.58	109	44.3	1237	66	26.32	...	0	0	0	0	0	
128	1.0	1114	288	790	73.55	278	35.4	99	23	9.90	...	0	0	0	0	0	
129	1.0	288	64	343	95.81	108	39.4	11	10	11.00	...	0	1	0	0	0	

130 rows × 38 columns

In [32]:
```
X_1.shape
```

Out[32]:
```
(130, 38)
```

Splitting data to train and test

In [33]:
```python
from sklearn.model_selection import train_test_split

X_train_1, X_test1, y_train_1, y_test_1 = train_test_split(X_1, y, test_size = 0.2, random_state = 10)
```

In [35]:
```python
X_train_1.shape, X_test1.shape, y_train_1.shape, y_test_1.shape
```

Out[35]:
```
((104, 38), (26, 38), (104,), (26,))
```

In [36]:
```python
X_train_1
```

Out[36]:

	const	T-RUNS	T-WKTS	ODI-RUNS-S	ODI-SR-B	ODI-WKTS	ODI-SR-BL	RUNS-S	HS	AVE	...	COUNTRY_SA	COUNTRY_SL	COUNTRY_WI	COUNTRY_ZIM	PLAYING ROLE_Allrounder
19	1.0	654	11	2536	84.00	25	47.6	978	74	36.22	...	1	0	0	0	0
14	1.0	0	0	69	56.09	0	0.0	1540	95	31.43	...	0	0	0	0	0
91	1.0	9382	0	10472	75.75	0	0.0	1567	94	27.98	...	0	1	0	0	0
35	1.0	503	0	575	87.51	1	66.0	1006	73	31.44	...	0	0	0	0	0
20	1.0	380	157	73	45.62	60	35.6	4	3	4.00	...	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
64	1.0	392	43	5	27.77	19	40.1	186	31	10.94	...	0	0	0	0	0
15	1.0	3509	0	6773	88.19	1	12.0	1782	70	37.13	...	0	0	0	0	0
100	1.0	537	1	1587	70.40	1	42.0	40	23	20.00	...	0	1	0	0	0
125	1.0	0	0	0	0.00	0	0.0	49	16	9.80	...	0	0	0	0	0
9	1.0	5515	1	4686	84.76	0	0.0	394	50	28.14	...	1	0	0	0	0

104 rows × 38 columns

Building the model

In [38]:
```python
mlr_1 = sm.OLS(y_train_1, X_train_1) # OLS Ordinary least squares
```

In [39]:
```python
mlr_1 = mlr_1.fit()
```

In [40]:
```
mlr_1.params
```

Out[40]:
```
const                        -4.030794e+07
T-RUNS                       -3.642910e+01
T-WKTS                       -7.924946e+02
ODI-RUNS-S                    1.524333e+01
ODI-SR-B                     -1.061064e+03
ODI-WKTS                      1.649076e+03
ODI-SR-BL                    -1.044786e+03
RUNS-S                        1.805545e+02
HS                           -2.881482e+03
AVE                           5.848201e+03
SR-B                         -6.373365e+01
SIXERS                        3.016505e+03
RUNS-C                        1.745518e+02
WKTS                         -1.364873e+03
AVE-BL                        1.169297e+04
ECON                         -3.327271e+03
SR-BL                        -1.669414e+04
AUCTION YEAR                  4.406899e+04
BASE PRICE                    1.888119e+00
AGE_1                        -1.329211e+07
AGE_2                        -1.347979e+07
AGE_3                        -1.353603e+07
COUNTRY_AUS                  -4.453859e+06
COUNTRY_BAN                   9.993025e-08
COUNTRY_ENG                  -4.916729e+06
COUNTRY_IND                  -4.303342e+06
COUNTRY_NZ                   -4.374145e+06
COUNTRY_PAK                  -4.496219e+06
COUNTRY_SA                   -4.395567e+06
COUNTRY_SL                   -4.486193e+06
COUNTRY_WI                   -4.386283e+06
COUNTRY_ZIM                  -4.495603e+06
PLAYING ROLE_Allrounder      -1.005057e+07
PLAYING ROLE_Batsman         -1.001859e+07
PLAYING ROLE_Bowler          -1.009737e+07
PLAYING ROLE_W. Keeper       -1.014141e+07
CAPTAINCY EXP_0              -2.023362e+07
CAPTAINCY EXP_1              -2.007432e+07
dtype: float64
```

Diagnosing the model

In [41]:
```
mlr_1.summary2()
```

Out[41]:

Model:	OLS	Adj. R-squared:	0.503
Dependent Variable:	SOLD PRICE	AIC:	2941.3368
Date:	2022-10-06 16:58	BIC:	3028.6017
No. Observations:	104	Log-Likelihood:	-1437.7
Df Model:	32	F-statistic:	4.257
Df Residuals:	71	Prob (F-statistic):	1.92e-07
R-squared:	0.657	Scale:	8.7185e+10

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	-40307940.3172	24745537.1103	-1.6289	0.1078	-89649139.9119	9033259.2775
T-RUNS	-36.4291	26.8420	-1.3572	0.1790	-89.9505	17.0923
T-WKTS	-792.4946	566.6974	-1.3984	0.1663	-1922.4571	337.4678
ODI-RUNS-S	15.2433	28.6606	0.5319	0.5965	-41.9042	72.3909
ODI-SR-B	-1061.0636	1450.3045	-0.7316	0.4668	-3952.8887	1830.7615
ODI-WKTS	1649.0764	742.0599	2.2223	0.0295	169.4510	3128.7018
ODI-SR-BL	-1044.7855	1686.6499	-0.6194	0.5376	-4407.8700	2318.2989
RUNS-S	180.5545	163.9192	1.1015	0.2744	-146.2911	507.4002
HS	-2881.4824	2458.9831	-1.1718	0.2452	-7784.5555	2021.5907
AVE	5848.2011	7729.2484	0.7566	0.4518	-9563.4826	21259.8847
SR-B	-63.7337	1172.4877	-0.0544	0.9568	-2401.6078	2274.1405
SIXERS	3016.5046	3549.7471	0.8498	0.3983	-4061.4900	10094.4993
RUNS-C	174.5518	249.3836	0.6999	0.4863	-322.7049	671.8085
WKTS	-1364.8732	6016.5118	-0.2269	0.8212	-13361.4571	10631.7107
AVE-BL	11692.9681	9725.7685	1.2023	0.2333	-7699.6634	31085.5997
ECON	-3327.2705	9459.2777	-0.3517	0.7261	-22188.5345	15533.9935
SR-BL	-16694.1377	13373.3174	-1.2483	0.2160	-43359.7753	9971.4998
AUCTION YEAR	44068.9943	27027.4095	1.6305	0.1074	-9822.1297	97960.1183
BASE PRICE	1.8881	0.5338	3.5374	0.0007	0.8238	2.9524
AGE_1	-13292113.4053	8254614.0242	-1.6103	0.1118	-29751346.2895	3167119.4789
AGE_2	-13479794.7546	8248893.0973	-1.6341	0.1067	-29927620.4346	2968030.9254
AGE_3	-13536032.1574	8242925.8285	-1.6421	0.1050	-29971959.4414	2899895.1266
COUNTRY_AUS	-4453859.3284	2771303.9943	-1.6071	0.1125	-9979682.5470	1071963.8902
COUNTRY_BAN	0.0000	0.0000	1.6302	0.1075	-0.0000	0.0000
COUNTRY_ENG	-4916729.2323	2814659.3130	-1.7468	0.0850	-10529000.5010	695542.0365
COUNTRY_IND	-4303342.2864	2779116.7278	-1.5485	0.1260	-9844743.6532	1238059.0803
COUNTRY_NZ	-4374144.5913	2745138.4899	-1.5934	0.1155	-9847795.2760	1099506.0933
COUNTRY_PAK	-4496219.2405	2721433.5914	-1.6522	0.1029	-9922603.7001	930165.2191
COUNTRY_SA	-4395566.8905	2763070.5940	-1.5908	0.1161	-9904973.1751	1113839.3941
COUNTRY_SL	-4486192.5787	2731080.1550	-1.6426	0.1049	-9931811.7397	959426.5823
COUNTRY_WI	-4386283.1747	2747032.4467	-1.5967	0.1148	-9863710.3019	1091143.9526
COUNTRY_ZIM	-4495602.9945	2750548.5599	-1.6344	0.1066	-9980041.0523	988835.0633
PLAYING ROLE_Allrounder	-10050574.2972	6186416.1335	-1.6246	0.1087	-22385937.7147	2284789.1204
PLAYING ROLE_Batsman	-10018590.2432	6186804.6479	-1.6193	0.1098	-22354728.3364	2317547.8501
PLAYING ROLE_Bowler	-10097368.6031	6199035.9371	-1.6289	0.1078	-22457895.1943	2263157.9882
PLAYING ROLE_W. Keeper	-10141407.1739	6176128.9522	-1.6420	0.1050	-22456258.5345	2173444.1867
CAPTAINCY EXP_0	-20233622.9568	12374362.3734	-1.6351	0.1064	-44907400.7375	4440154.8238
CAPTAINCY EXP_1	-20074317.3605	12371417.5760	-1.6226	0.1091	-44742223.3819	4593588.6610

Omnibus:	11.448	Durbin-Watson:	2.154
Prob(Omnibus):	0.003	Jarque-Bera (JB):	12.071
Skew:	0.705	Prob(JB):	0.002
Kurtosis:	3.893	Condition No.:	11985550242486654

Note:

Only ODI_WKTS and BASE PRICE are relevant features.

Multicollinearity

In [42]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [44]:
```
def var_inf_factor(data): #objective - to create a datafram; 1st column -> features, 2nd column -> corres values
    vif = pd.DataFrame()
    vif['Feature'] = data.columns
    vif['VIF_Value'] = [variance_inflation_factor(data.values, i) for i in range(data.shape[1])]
    print(vif)
```

In [45]:
```
var_inf_factor(X_1)
```

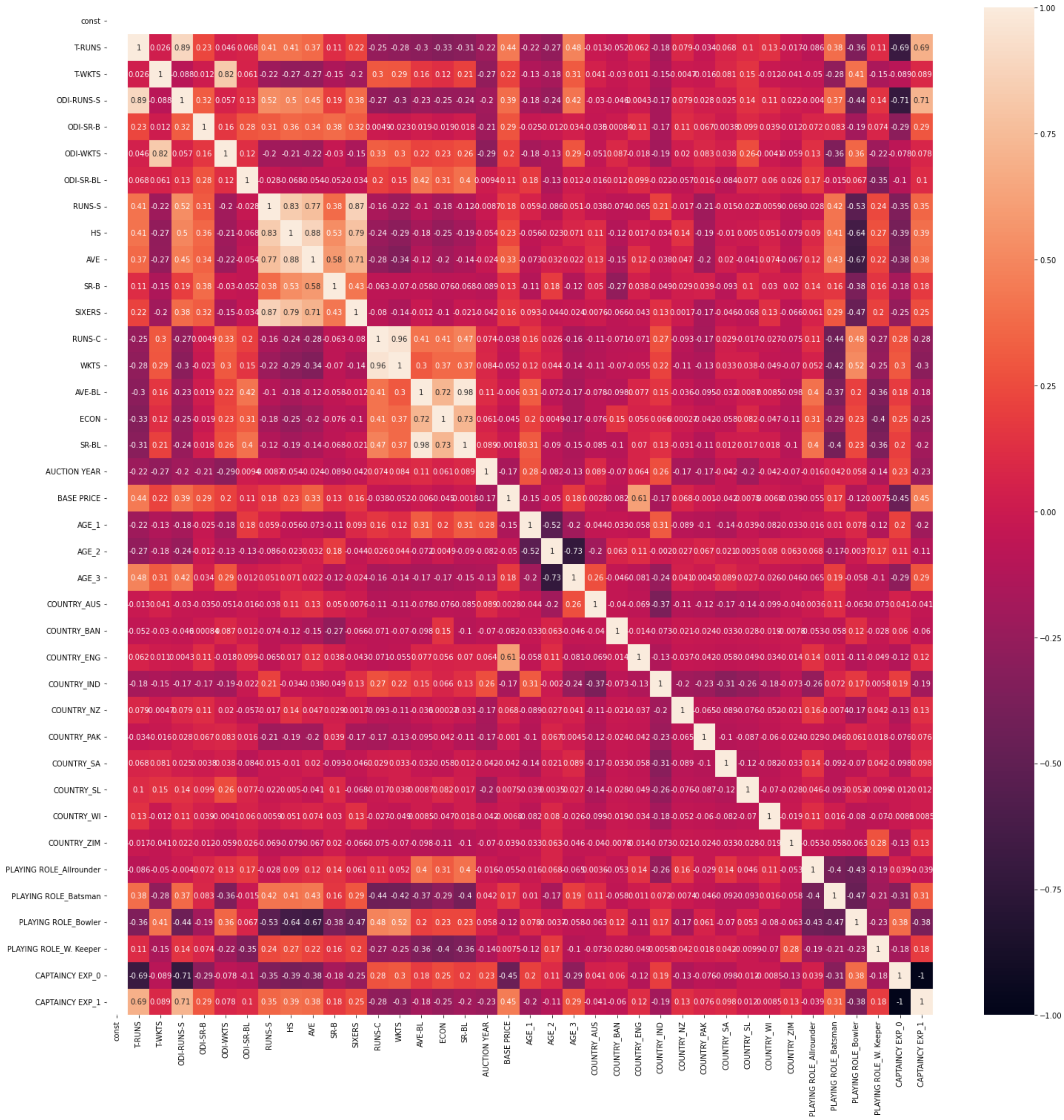
```
                   Feature  VIF_Value
0                    const   0.000000
1                   T-RUNS   9.233542
2                   T-WKTS   6.522453
3               ODI-RUNS-S  11.067128
4                 ODI-SR-B   1.703841
5                 ODI-WKTS   7.048664
6                ODI-SR-BL   1.707550
7                   RUNS-S   9.948044
8                       HS   8.602278
9                      AVE   7.467939
10                    SR-B   2.293498
11                  SIXERS   6.425581
12                  RUNS-C  22.310115
13                    WKTS  20.896087
14                  AVE-BL  45.182628
15                    ECON   2.981483
16                   SR-BL  45.596075
17            AUCTION YEAR   1.508571
18              BASE PRICE   3.347050
19                   AGE_1        inf
20                   AGE_2        inf
21                   AGE_3        inf
22             COUNTRY_AUS        inf
23             COUNTRY_BAN        inf
24             COUNTRY_ENG        inf
25             COUNTRY_IND        inf
26              COUNTRY_NZ        inf
27             COUNTRY_PAK        inf
28              COUNTRY_SA        inf
29              COUNTRY_SL        inf
30              COUNTRY_WI        inf
31             COUNTRY_ZIM        inf
32  PLAYING ROLE_Allrounder        inf
33     PLAYING ROLE_Batsman        inf
34      PLAYING ROLE_Bowler        inf
35    PLAYING ROLE_W. Keeper        inf
36          CAPTAINCY EXP_0        inf
37          CAPTAINCY EXP_1        inf

C:\Users\Urvi Sharma\anaconda3\lib\site-packages\statsmodels\regression\linear_model.py:1715: RuntimeWarning: divide by zero en
countered in double_scalars
  return 1 - self.ssr/self.centered_tss
C:\Users\Urvi Sharma\anaconda3\lib\site-packages\statsmodels\stats\outliers_influence.py:193: RuntimeWarning: divide by zero en
countered in double_scalars
  vif = 1. / (1. - r_squared_i)
```

consider variables with vif_value>4 and check it's correlation with other variables using heatmap

In [46]:
```
plt.figure(figsize = (25, 25))
sns.heatmap(X_1.corr(), annot = True)
```

Out[46]: <AxesSubplot:>



Note:

```
T-RUNS <==> ODI-RUNS-SCORE

T-WKTS <==> ODI-WKTS

ODI-RUNS-S <==> CAPTIANCY_EXP_1
```

In [ ]: