Multiple Regression Analysis on NBA Team Statistics To Predict Future Team Success

Nivethan Iruthayanathan

Student Number: 219580100

COSC 3117A

Introduction:

This study took NBA data ranging from 2003-2021 and looked for possible predictors of and analyzed the relationship in the advanced statistical categories of basketball to the number of wins a team has that season.

Background Information:

The NBA is one of the few sports with a variety of countable statistics which has led to the rise of some of the most comprehensive advanced statistics in recent years. The implications of these stats are broad but more relevant than ever with the legalization of sports betting in Ontario. This study conducted a multiple linear regression analysis using several python frameworks to try and determine the more correlated advanced statistics to team wins per season. The data includes stats from 2003 - 2022 to maintain consistency of data as there have been several changes to the league: the NBA added a team to the league in 2003, rule changes increased the pace of play, and the revolution of three points per game.

Statistical Categories (all averages):
Age = Age of team
MOV = Margin of victory (how much the team wins by when they win)
SOS = Strength of season (strength of opposing teams X amount they matchup against them)
FTr = Free throw rating
3PAr = Three pointer rating
eFG% = Effective Field Goal percentage (field goal percentage adjusted for threes being worth more)
TOV% = Turnover percentage
ORB% = Offensive rebound percentage
FT/FGA = Free throw attempts/ field goal attempts

Methods:

In order to complete the multiple regression analysis, the largest amount of consistent data. Advanced stats were acquired and formatted from https://www.basketball-reference.com/leagues/. Csv was processed using the pandas library, it was then trained on the multiple regression model from the scikit learn library on python.
From here several different combinations of statistics were utilized to limit the multicollinearity as it can skew the results of multiple regression analyses which is further discussed in the discussions sections. Using the Pandas library and a tool called Seaborn, scatter plots were used to visualize the individual relationships between each statistical category and the number

of wins an NBA team has that season. *The code implementation can be seen at the end of the appendix.*

Data:

mean_squared_error : 11.132553765005614
mean_absolute_error : 2.706219184307642

```
                    OLS Regression Results
==============================================================================
Dep. Variable:                  W   R-squared:                   0.933
Model:                        OLS   Adj. R-squared:              0.931
Method:             Least Squares   F-statistic:                 380.3

                               Prob (F-statistic):         5.95e-259
Time:                    23:03:19   Log-Likelihood:            -1237.9
No. Observations:             480   AIC:                         2512.
Df Residuals:                 462   BIC:                         2587.
Df Model:                      17
Covariance Type:        nonrobust
==============================================================================
```

	coef	std err	t	P>|t|	[0.025	0.975]
const	47.2093	54.191	0.871	0.384	-59.282	153.700
Age	0.3994	0.116	3.449	0.001	0.172	0.627
MOV	-0.6474	0.816	-0.793	0.428	-2.251	0.956
SOS	0.0810	0.489	0.166	0.869	-0.880	1.042
ORtg	1.5495	1.120	1.383	0.167	-0.652	3.751
DRtg	-2.3318	0.975	-2.393	0.017	-4.247	-0.417
Pace	-0.0157	0.067	-0.234	0.815	-0.147	0.116
FTr	-23.5452	67.199	-0.350	0.726	-155.598	108.508
3PAr	1.0316	5.388	0.191	0.848	-9.557	11.620
TS%	126.8742	319.858	0.397	0.692	-501.682	755.430
eFG%	98.1897	269.327	0.365	0.716	-431.068	627.448
TOV%	-2.1310	1.166	-1.827	0.068	-4.423	0.161
ORB%	0.7150	0.446	1.605	0.109	-0.161	1.590
FT/FGA	50.5696	136.657	0.370	0.712	-217.976	319.116
D_eFG%	-116.2263	84.389	-1.377	0.169	-282.059	49.607
D_TOV%	0.8352	0.752	1.111	0.267	-0.642	2.313
D_RB%	0.1240	0.275	0.450	0.653	-0.417	0.665
D_FT/FGA	-28.3443	17.589	-1.611	0.108	-62.909	6.221

```
==============================================================================
```

```
Omnibus:                      1.610   Durbin-Watson:                  1.759
Prob(Omnibus):                0.447   Jarque-Bera (JB):               1.622
Skew:                        -0.140   Prob(JB):                       0.444
Kurtosis:                     2.950   Cond. No.                    5.79e+05
==============================================================================
```

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.79e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

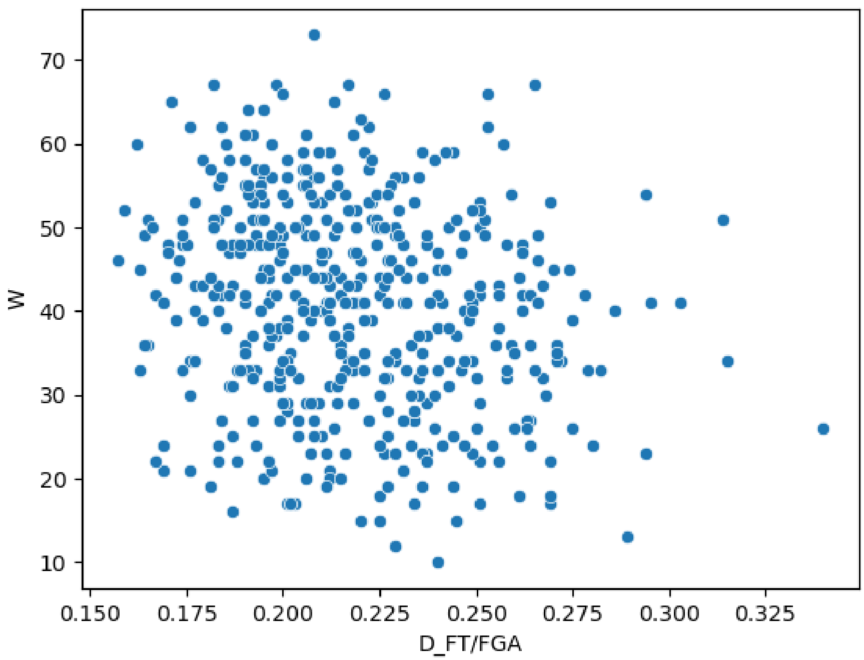
Discussion:

Some notable statistics will have to be discussed first. The t and P>|t| provide the t-value and the two-tailed p-value respectively. This would be a two-tailed test and the p-value of each. P values below 0.05 can be considered statistically significant values individually. Age, Defensive rating, and turnover percentage seem to be the most individually significant in relation to overall wins a team has because of their p values.

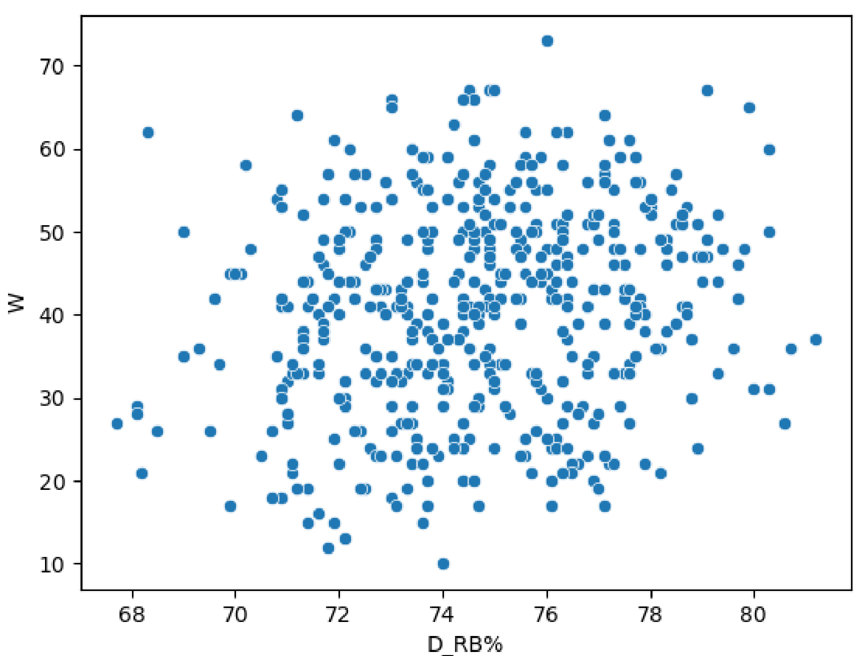
Furthermore, the Prb(F-stat) being so close to zero (below 0.001) is an indication that as a group of parameters, they are meaningful for win prediction and the null hypothesis can be rejected.

With more time and access to data, more parameters and combinations of parameters can be observed to make better predictions and can be used for real-life applications such as sports betting.

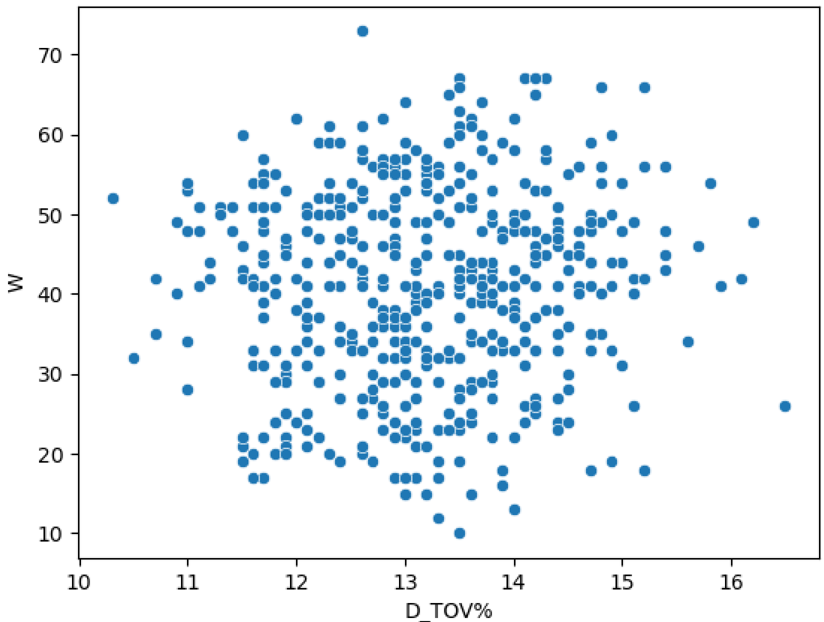
Appendix:



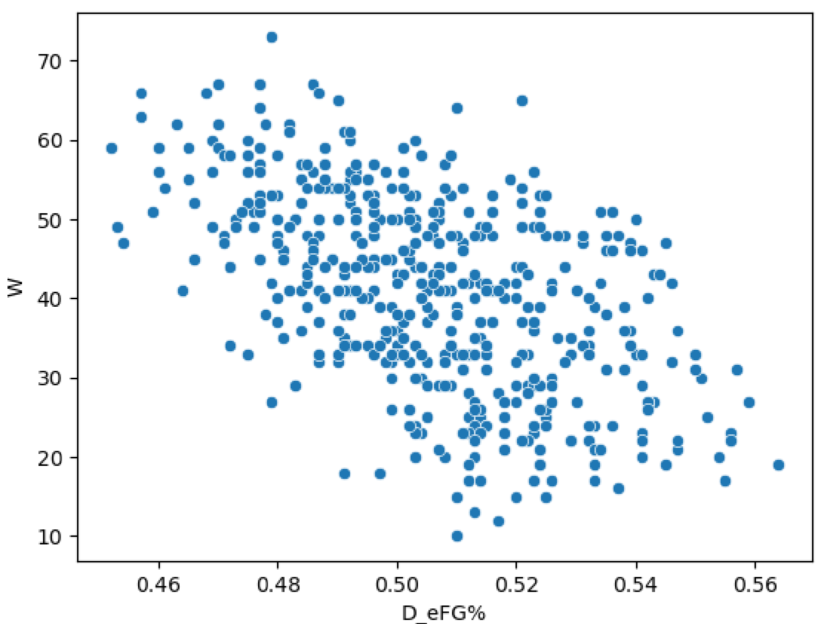
70
60
50
40
30
20
10
W
0.150
0.175
0.200
0.225
0.250
0.275
0.300
0.325
D_FT/FGA



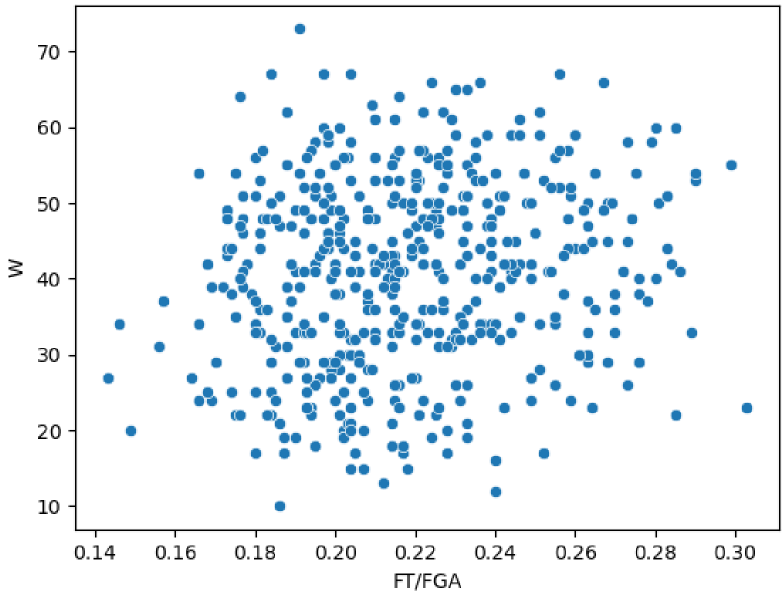
70
60
50
40
30
20
10
W
68
70
72
74
76
78
80
D_RB%

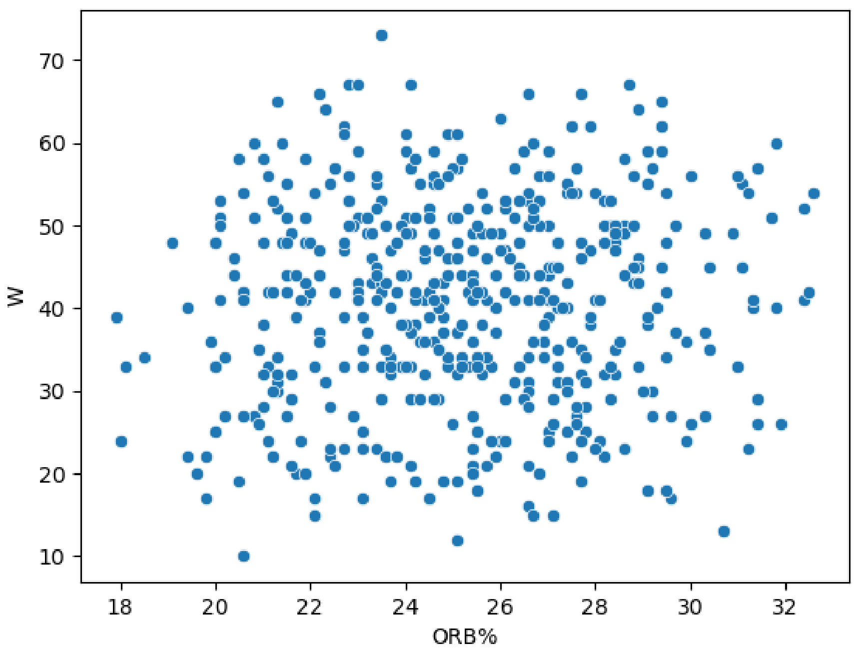


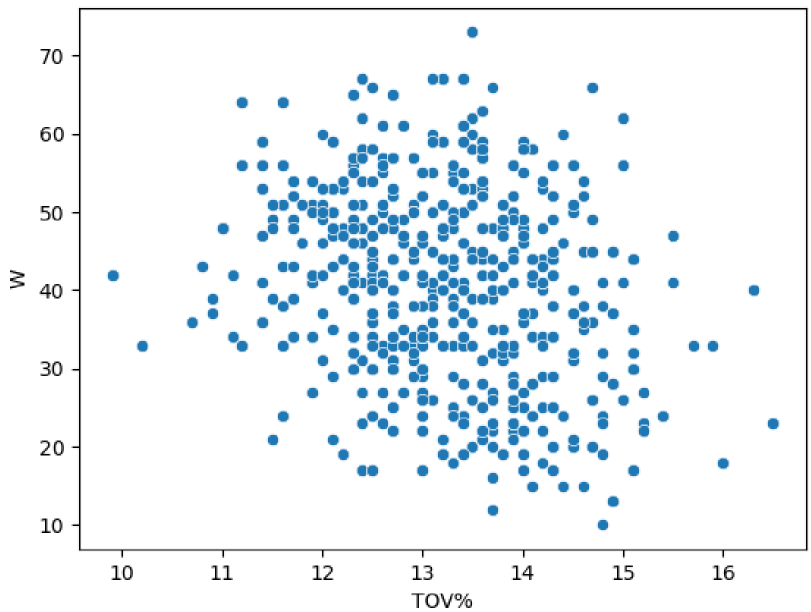
70
60
50
40
30
20
10
W
10
11
12
13
14
15
16
D_TOV%

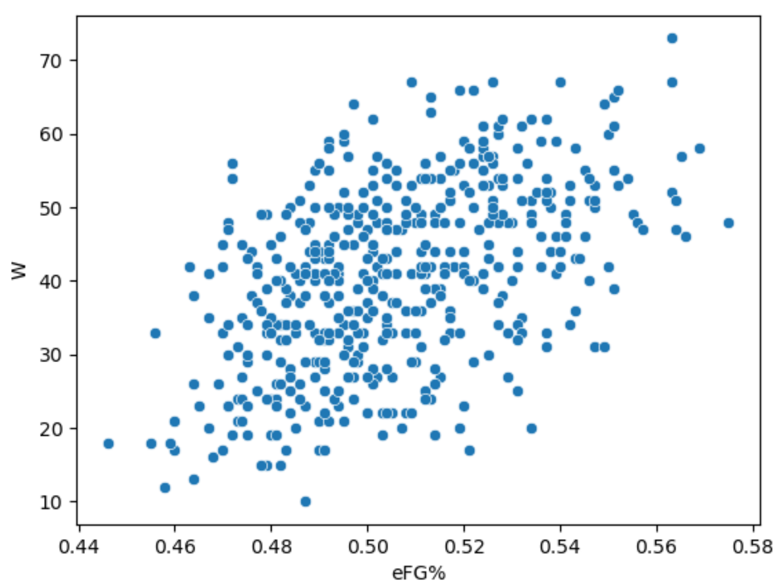


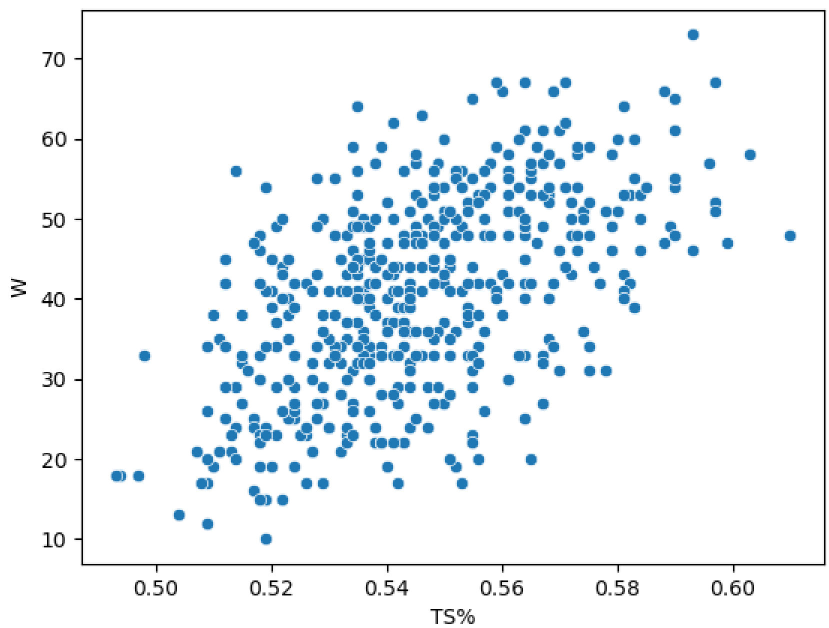
70
60
50
40
30
20
10
W
0.46
0.48
0.50
0.52
0.54
0.56
D_eFG%


W
FT/FGA
0.14 0.16 0.18 0.20 0.22 0.24 0.26 0.28 0.30
10 20 30 40 50 60 70

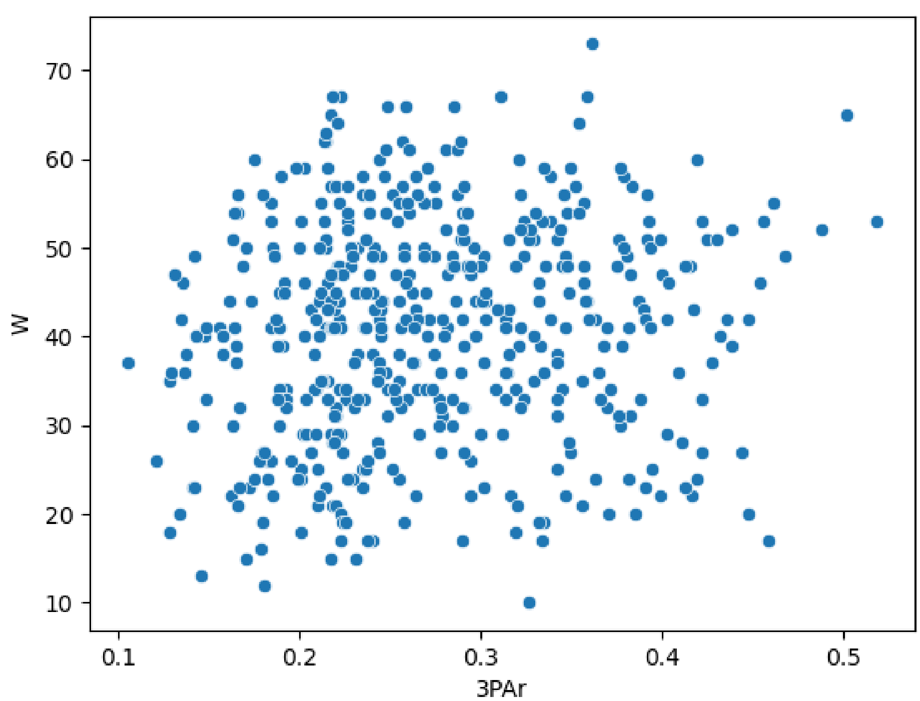

W
ORB%
18 20 22 24 26 28 30 32
10 20 30 40 50 60 70


W
TOV%
10
20
30
40
50
60
70
10
11
12
13
14
15
16

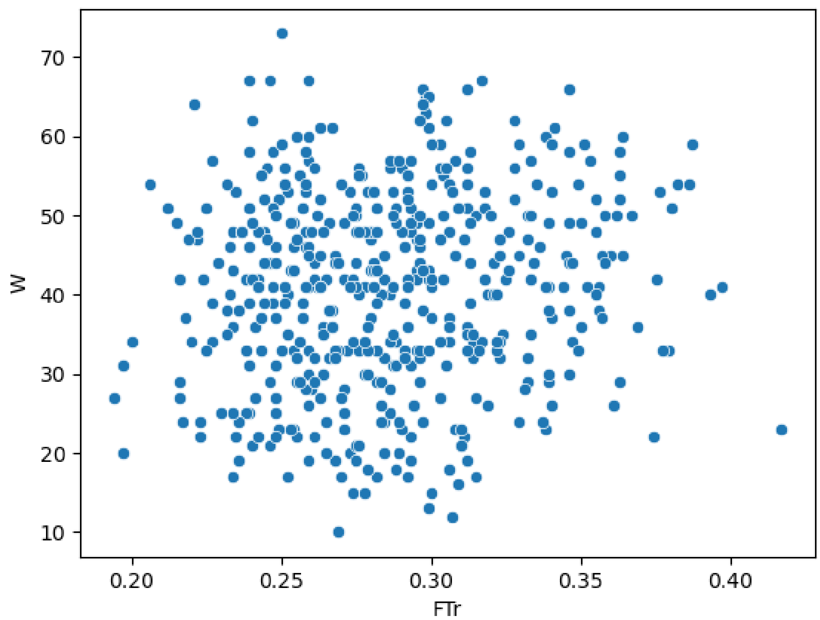

W
eFG%
10
20
30
40
50
60
70
0.44
0.46
0.48
0.50
0.52
0.54
0.56
0.58



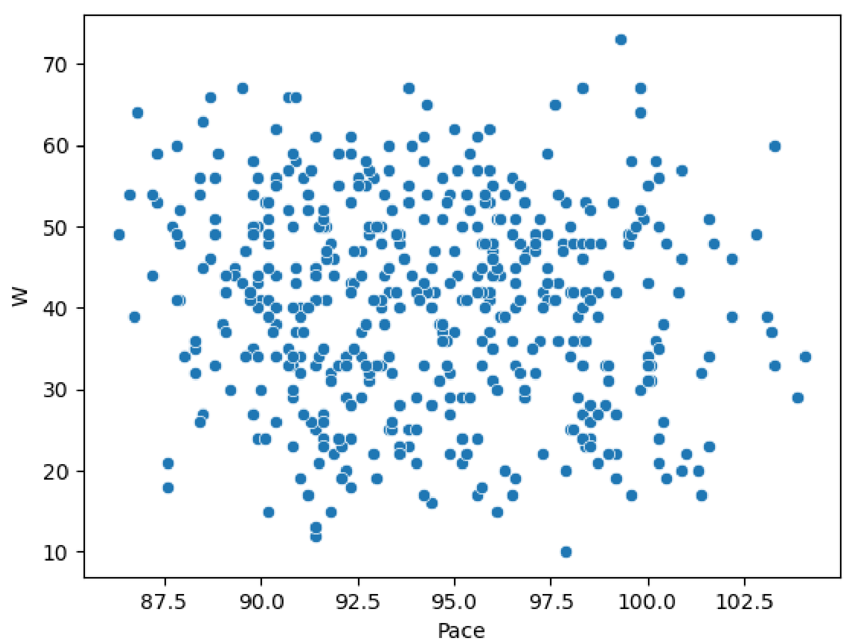
70
60
50
40
30
20
10
W
0.50
0.52
0.54
0.56
0.58
0.60
TS%



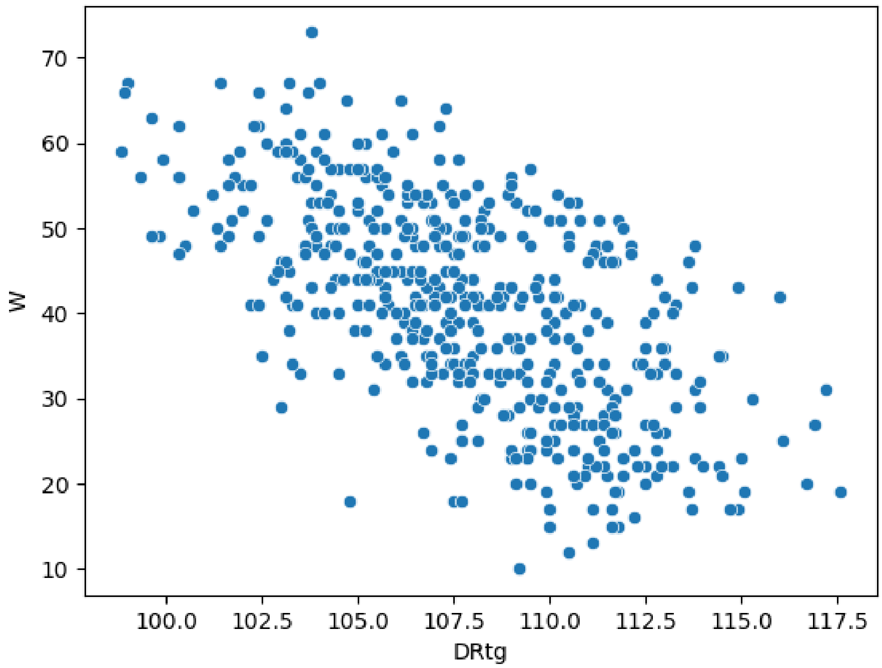
70
60
50
40
30
20
10
W
0.1
0.2
0.3
0.4
0.5
3PAr

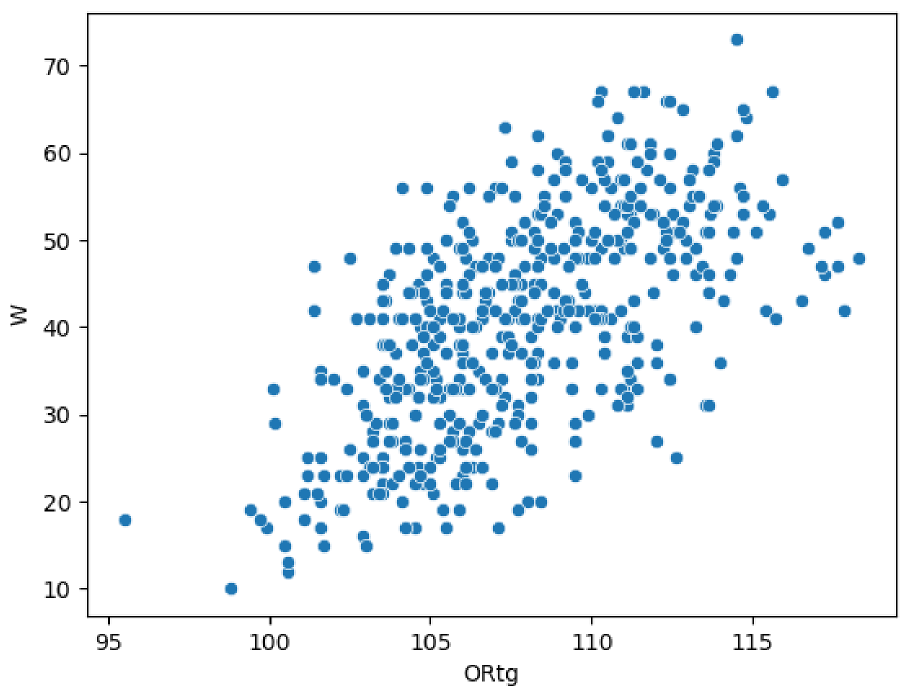


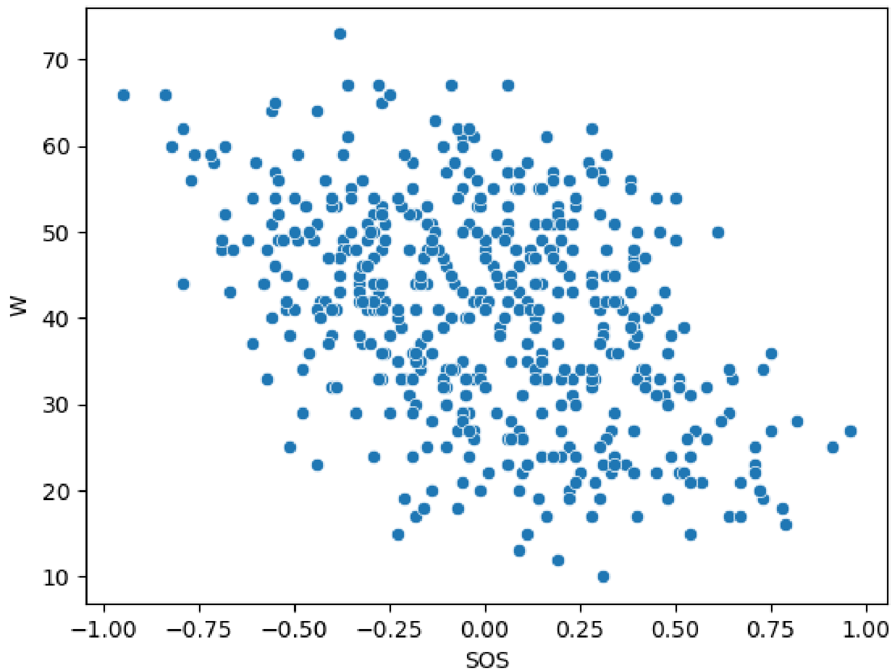
70
60
50
40
30
20
10
W
0.20
0.25
0.30
0.35
0.40
FTr

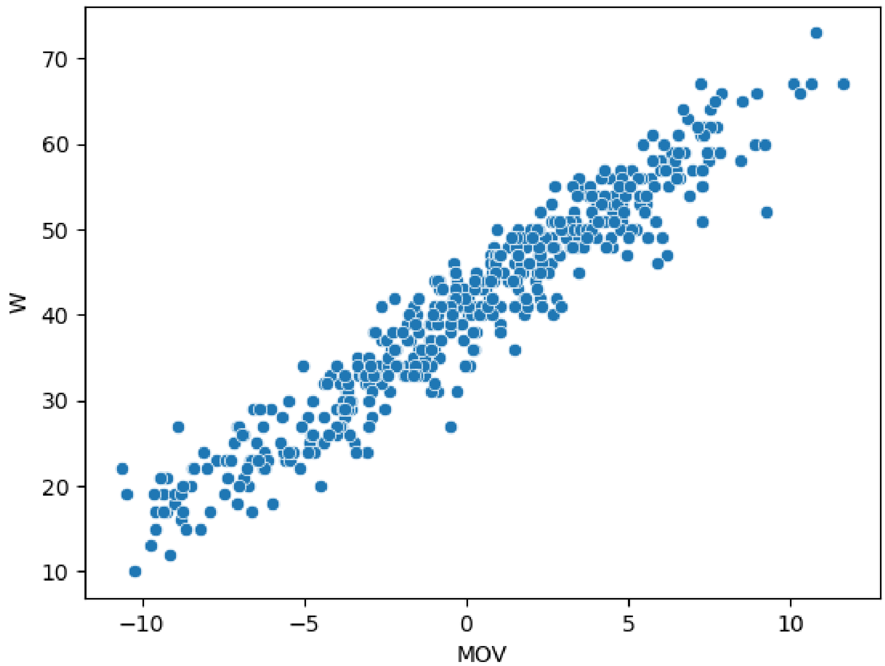


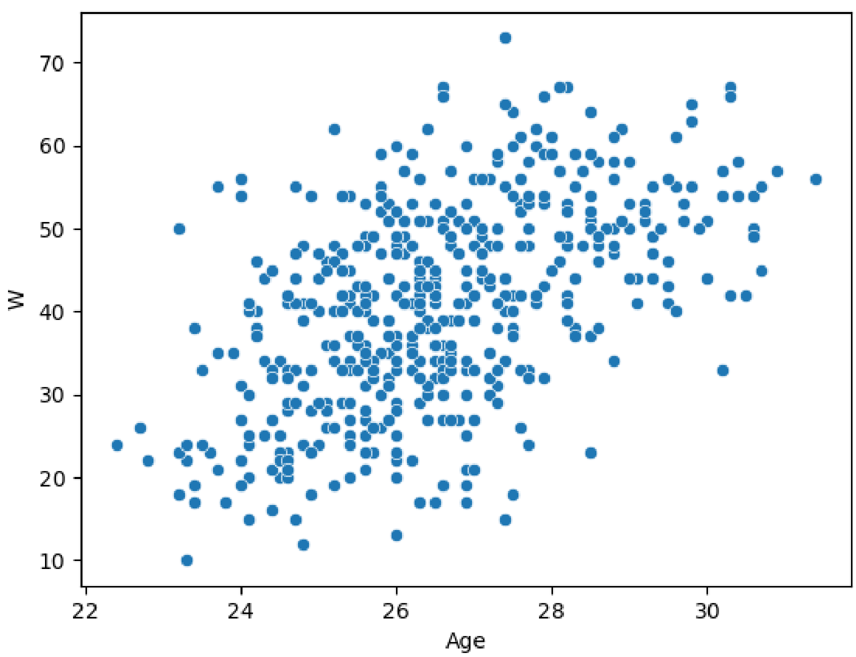
70
60
50
40
30
20
10
W
87.5
90.0
92.5
95.0
97.5
100.0
102.5
Pace


W
DRtg
100.0
102.5
105.0
107.5
110.0
112.5
115.0
117.5
10
20
30
40
50
60
70


W
ORtg
95
100
105
110
115
10
20
30
40
50
60
70


70
60
50
40
30
20
10
W
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
SOS





Code implementation:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.feature_selection import f_regression


df = pd.read_csv("MultipleRegressionProject/BallTeamStats.csv")
# df.drop( 'Rk' , inplace = True,axis=1)
print(df.head())

# print(df.head())
# print(df.columns)


X = df[['Age', 'MOV', 'SOS', 'ORtg', 'DRtg', 'Pace',
'FTr', '3PAr', 'TS%', 'eFG%', 'TOV%', 'ORB%', 'FT/FGA',
 'D_eFG%', 'D_TOV%', 'D_RB%', 'D_FT/FGA']]
y = df['W']

arrPar = ['Age', 'MOV', 'SOS', 'ORtg', 'DRtg', 'Pace',
'FTr', '3PAr', 'TS%', 'eFG%', 'TOV%', 'ORB%', 'FT/FGA',
 'D_eFG%', 'D_TOV%', 'D_RB%', 'D_FT/FGA']

for i in arrPar:
 sns.scatterplot(x=i, y='W', data=df)
```

```
plt.show()



X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.3, random_state=None)

model = LinearRegression()

model.fit(X_train,y_train)

predictions = model.predict(X_test)

finalPredict =
model.predict([[27.5,7.5,-0.56,114.8,107.3,99.8,0.221,0.354,0.581,0.549,11.6,22.3,0.17
6,0.51,13,77.1,0.195,]])
print(finalPredict)

print(
 'mean_squared_error : ', mean_squared_error(y_test, predictions))
print(
 'mean_absolute_error : ', mean_absolute_error(y_test, predictions))


import statsmodels.api as ssm #for detail description of linear coefficients,
intercepts, deviations, and many more

X=ssm.add_constant(X)        #to add constant value in the model
model= ssm.OLS(y,X).fit()         #fitting the model
print(model.summary())
```