ST4035 - Assignment 2

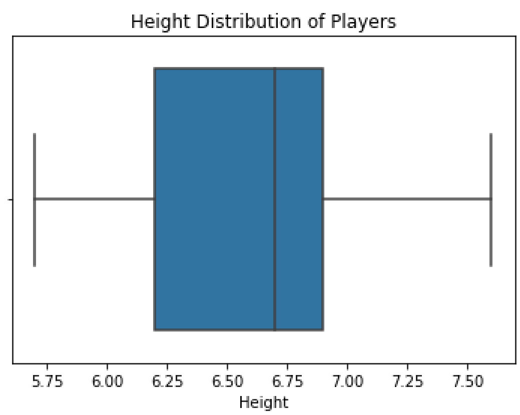
a) Descriptive Analysis

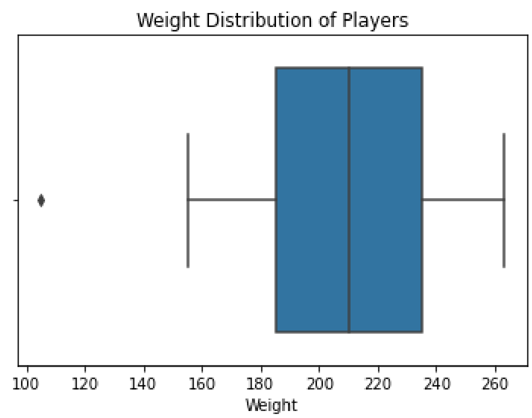
Information about variables in train dataset

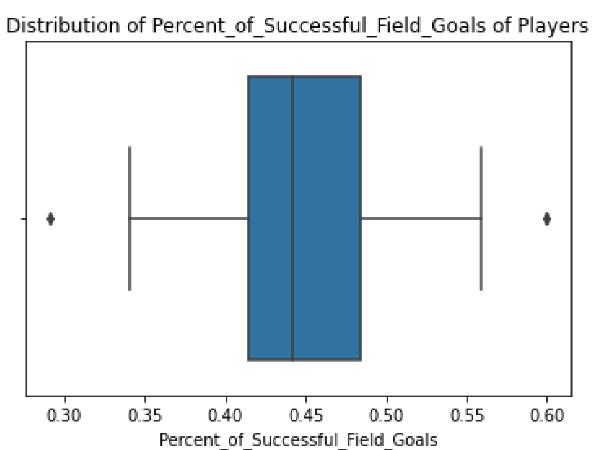
```
RangeIndex: 43 entries, 0 to 42
Data columns (total 6 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   ID                                  43 non-null     int64
 1   Height                              43 non-null     float64
 2   Weight                              43 non-null     int64
 3   Percent_of_Successful_Field_Goals   43 non-null     float64
 4   Percent_of_Successful_Free_Throws   43 non-null     float64
 5   Average_Points_Scored_Per_Game      43 non-null     float64
```

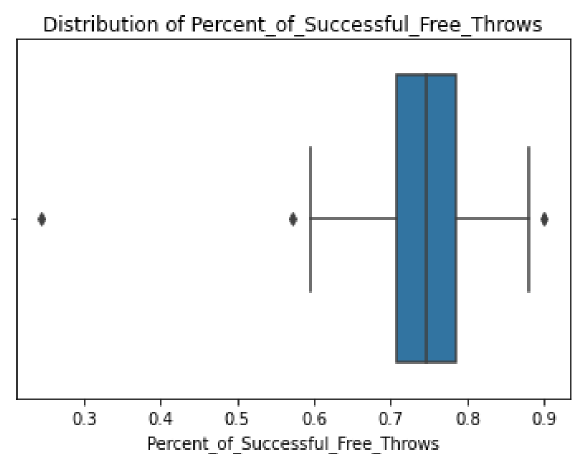
There are no any missing values (NaN) in this train dataset.

- Uni-Variate analysis



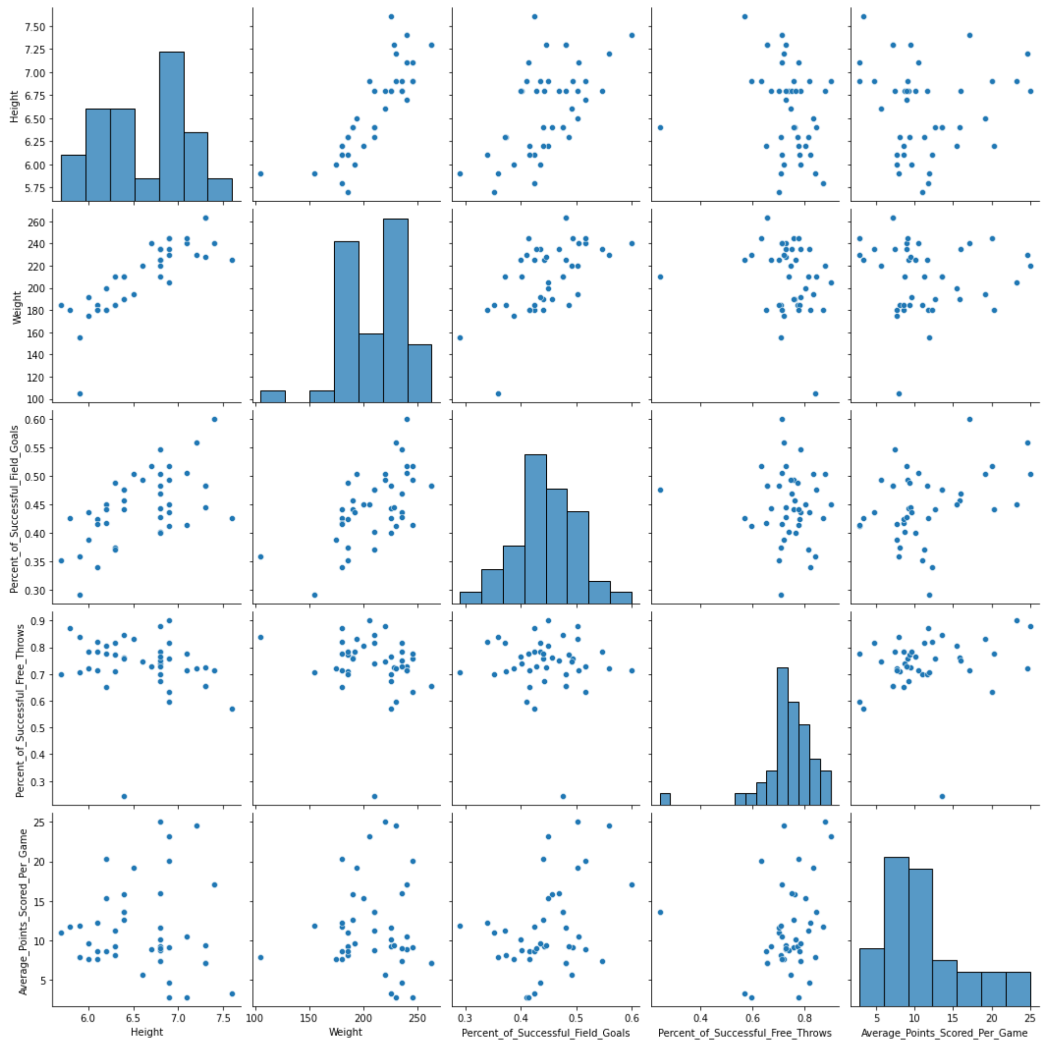






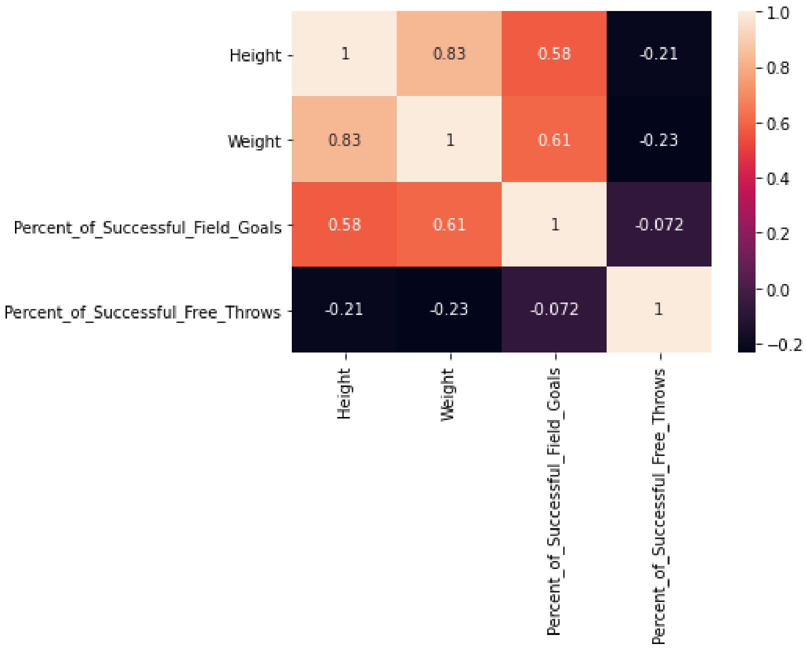
The above boxplot represents the distribution of each independent variable in the train data set. Only the height variable does not have outliers. Other tree variables have outliers. Outliers are not removed without knowing the background of the dataset. After the outlier which deviated significantly and included the percent of successful free throws variable was removed R square value of the model was increased by a considerable amount. But RMSE score of that model also increased. So that there is no point in removing the outliers without doing an investigation about that.

- Pairwise Comparison Between Independent Variables



Using this pairwise plot we can get an idea about correlations between independent variables. And from the diagonal can get an idea about the shape of the distributions of each variable. According to this plot, we can see there is a strong positive relationship between height and weight variables. And also, between Height and Percent of Successful Field Goals, weight and Percent of Successful Field Goals have a moderate positive relationship. Checking the collinearity among independent variables before fitting the regression is very important.

Below heat map was used to get more information about correlations between independent variables.



This heat map has confirmed the results that got from the above pairwise plot. Height and weight variables have a correlation with other variables. So surely these two variables have multicollinearity. When fitting the model have to consider multicollinearity.

b)

Python language and sklearn library was used to build up the model. Because Height and Weight are highly correlated there is no pointing fitting model using all variables. A dimension reduction process was used before fitting the model. Recursive feature elimination (RFECV) was imported from sklearn library and applied to select the best features for the model. As a minimum feature three are used. (Only one variable was selected for value 2 when the model reaches the null model chance of predicting the values correctly is very lower)

To feather improve the model has tried the Lasso, Ridge, and ElasticNet models in the sklearn library. (Because there may have multicollinearity, there is a high correlation between the height and weight of the player.)KFold was imported and tried to optimize the parameter alpha and the Mean Squared Error of each model was compared using test data (20% random sample of train dataset). But RMSE value given by the Kaggle was not improved than the above model.

Finally, some interaction terms were added to the date set, and using the above-mentioned feature selection procedure (minimum features =3) variables were selected (this method is used to improve the Kaggle score), and also using some transformations (log and reciprocal) models were built.

Model

X5 = -71.93798042586431 + 160.99554659(X3) + 91.82821341(X4) + -171.09647741(X3*X4)

c)

- **Predictions**

ID	X5
1	11.96388
2	11.55789
3	13.34395
4	12.9472
5	13.74795
6	11.73873
7	6.303193
8	12.5581
9	14.24419
10	11.64663
11	11.38963

- **Code**

```
import pandas as pd
from sklearn.model_selection import
train_test_split,GridSearchCV,RandomizedSearchCV,KFold
from sklearn.feature_selection import RFECV
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge, RidgeCV, Lasso, ElasticNet
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Import Dataset

```
test=pd.read_csv('test_data_id.csv')
train=pd.read_csv('train_data_id.csv')
train.head()
```

Out[80]:

	X1	X2	X3	X4	X12	X13	X14	X23	X24	X34	X123	X124	X234	X1234	X5
0	6.7	240	0.516	0.728	1608.0	3.4572	4.8776	123.840	174.720	0.375648	829.7280	1170.6240	90.15552	604.041984	8.9
1	7.3	263	0.482	0.655	1919.9	3.5186	4.7815	126.766	172.265	0.315710	925.3918	1257.5345	83.03173	606.131629	7.2
2	6.3	185	0.374	0.709	1165.5	2.3562	4.4667	69.190	131.165	0.265166	435.8970	826.3395	49.05571	309.050973	8.1
3	6.8	210	0.402	0.739	1428.0	2.7336	5.0252	84.420	155.190	0.297078	574.0560	1055.2920	62.38638	424.227384	8.7
4	5.8	180	0.425	0.872	1044.0	2.4650	5.0576	76.500	156.960	0.370600	443.7000	910.3680	66.70800	386.906400	11.8

```
#sns.pairplot(data=train, diag_kind='kde')
#sns.heatmap(train[['X2', 'X3', 'X4']].corr(),annot=True)
#plt.show()
X_train=train.iloc[:44,:14]
Y_train=train.iloc[:44,14]
```

Variable Selection procedure

```
rfe = RFECV(estimator=LinearRegression(),min_features_to_select=3)
Reg_var = rfe.fit(X_train,Y_train)
Reg_var.support_
```

Out[129]:

```
array([False, False,  True,  True, False, False, False, False, False,
        True, False, False, False, False])
X_train=train[['X3','X4','X34']]
X_test=test[['X3','X4','X34']]
```

Model Fitting

```
model=LinearRegression()
m=model.fit(X_train,(Y_train))
m.intercept_
```

Out[133]:

```
-71.93798042586431
m.coef_
```

Out[134]:

```
array([ 160.99554659,   91.82821341, -171.09647741])
m.score(X_train,(Y_train))
```

Out[135]:

```
0.20846502213621343
```

Predictions

```
y_pred=model.predict(X_test)
(y_pred)
```

Out[136]:

```
array([11.96387519, 11.55789283, 13.34395294, 12.94719841, 13.74794741,
       11.73873058,  6.30319272, 12.5580987 , 14.2441941 , 11.64663281,
       11.38963408])
```

Lasso, Ridge, ElasticNet Models

```
x=train[['X3','X4','X34']]
y=train.iloc[:,14]
x_test_data=test[['X3','X4','X34']]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state
=0)
model1=Lasso()
params={"alpha":[0.08,0.09,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,4,6,7,8,9,
10,20,50,100 ]}
cval=KFold(n_splits=10)
model3=Ridge()
params={"alpha":[0.08,0.09,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,4,6,7,8,9,
10,20,50,100 ]}
cval=KFold(n_splits=10)
model2=ElasticNet()
params={"alpha":[0.08,0.09,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,4,6,7,8,9,
10,20,50,100 ]}
cval=KFold(n_splits=10)
gsearch=GridSearchCV(model1,params,cv=cval)
esearch=GridSearchCV(model2,params,cv=cval)
rsearch=GridSearchCV(model3,params,cv=cval)
results=gsearch.fit(x_train,y_train)
results.best_params_
```

Out[192]:

```
{'alpha': 0.3}
results=esearch.fit(x_train,y_train)
results.best_params_
```

Out[193]:

```
{'alpha': 0.5}
results=rsearch.fit(x_train,y_train)
results.best_params_
```

Out[194]:

```
{'alpha': 20}
```

```
lmodel=Lasso(alpha=0.0001)
lmodel.fit(x_train,y_train)

emodel=ElasticNet(alpha=0.0001) #0.0000000000001
emodel.fit(x_train,y_train)

rmodel=Ridge(alpha=0.0001) #0.0000000000001
rmodel.fit(x_train,y_train)
C:\Users\user\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_de
scent.py:530: ConvergenceWarning: Objective did not converge. You might want
to increase the number of iterations. Duality gap: 111.75892123933562, tolera
nce: 0.0884129411764706
  model = cd_fast.enet_coordinate_descent(
```

Out[195]:

```
Ridge(alpha=0.0001)
```

Lasso Model

```
lmodel.coef_
```

Out[196]:

```
array([ 161.80636243,   97.59329005, -178.04745855])
lmodel.score(x_train,y_train)
```

Out[197]:

```
0.23422944880539365
ly_pred=(lmodel.predict(x_test))
MSE=mean_squared_error(ly_pred,y_test)
MSE
```

Out[198]:

```
37.2987995569082
(lmodel.predict(x_test_data))
```

Out[199]:

```
array([12.39040027, 11.95625791, 13.62566875, 12.97517821, 14.09639711,
       12.32044569,  6.20446293, 12.64309745, 14.21624667, 11.80604321,
       11.72569296])
```

Ridge Model

```
rmodel.coef_
```

Out[200]:

```
array([ 134.12560266,   80.10815106, -140.19310027])
rmodel.score(x_train,y_train)
```

Out[201]:

```
0.23257982398680166
ry_pred=emodel.predict(x_test)
MSE=mean_squared_error(ry_pred,y_test)
MSE
```

Out[202]:

```
36.978745286128586
rmodel.predict(x_test_data)
```

Out[207]:

```
array([12.33838562, 11.91417509, 13.70484516, 13.00238961, 14.2198792 ,
       12.18095398,  6.5483167 , 12.66679267, 14.37821053, 11.82048909,
       11.70728859])
```

ElasticNet Model

```
emodel.coef_
```

Out[208]:

```
array([38.65849254, 19.89310131, -9.99741727])
emodel.score(x_train,y_train)
```

Out[209]:

```
0.21606643535533232
ey_pred=emodel.predict(x_test)
MSE=mean_squared_error(ey_pred,y_test)
MSE
```

Out[210]:

```
36.978745286128586
emodel.predict(x_test_data)
```

Out[211]:

```
array([12.15715879, 11.76946494, 13.96087514, 13.08292974, 14.62447428,
       11.70366277,  7.75321321, 12.73823351, 14.90907878, 11.86692117,
       11.64419938])
```