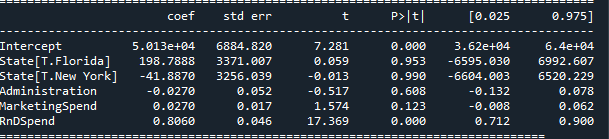
Prepare a prediction model for profit of 50\_startups data.

Do transformations for getting better predictions of profit and

make a table containing R^2 value for each prepared model.

When applying the model 1 we get the p value which is greater than 0.05 for state and administration

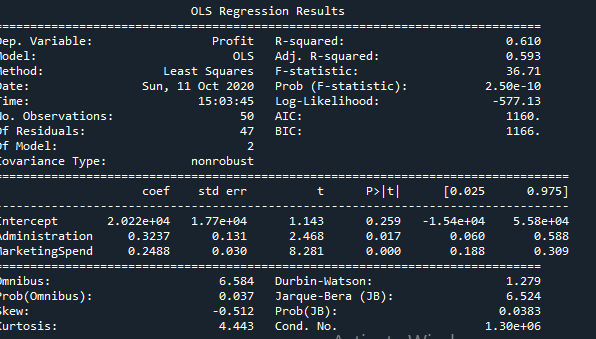


R Squared value is 0.945 when we prepared for model 1

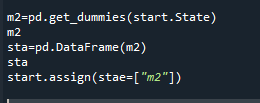
After then, I individually do a model for Administration and Marketing Spend with the target class Profit.

When I Applied this model, I get the P value less than 0.05 and Adjust R squared value is 0.593 for this model



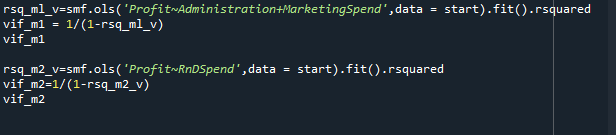


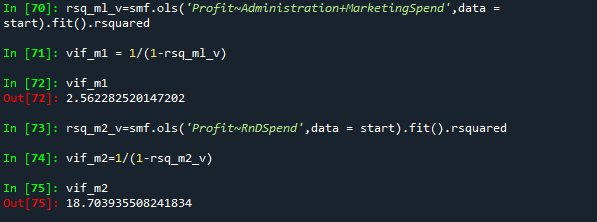
State is in Discrete Model. We cannot prepare an model with continuous and discrete values So, I create Dummy varabile and performed some transformation.



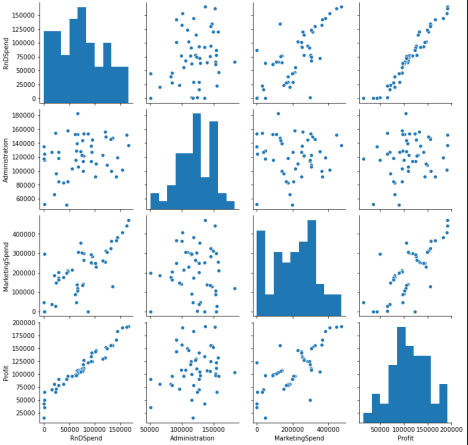
Even though preparing model for state we cannot predict it ,

Finally performing R squared values for both model so it is independent





PLOT:



FULL CODE:

# -\*- coding: utf-8 -\*-

"""

Created on Mon Oct 5 23:42:57 2020

@author: sunil

"""

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

start=pd.read\_csv("E://Excelr//excler//MLR assignment//50\_Startups.csv")

start.columns

type(start)

start.corr()

import seaborn as sns

sns.pairplot(start)

import statsmodels.formula.api as smf

ml1 = smf.ols('Profit~Administration+MarketingSpend+RnDSpend+State',data=start).fit()

ml1.params

ml1.summary()

pred = pd.DataFrame(ml1.predict(start))

pred

start["pred"]= pred

ml\_v=smf.ols('Profit~Administration+MarketingSpend',data = start).fit()

ml\_v.params

ml\_v.summary()

m2\_v=smf.ols('Profit~RnDSpend',data = start).fit()

m2\_v.params

m2\_v.summary()

m2=pd.get\_dummies(start.State)

m2

sta=pd.DataFrame(m2)

sta

start.assign(stae=["m2"])

rsq\_ml\_v=smf.ols('Profit~Administration+MarketingSpend',data = start).fit().rsquared

vif\_m1 = 1/(1-rsq\_ml\_v)

vif\_m1

rsq\_m2\_v=smf.ols('Profit~RnDSpend',data = start).fit().rsquared

vif\_m2=1/(1-rsq\_m2\_v)

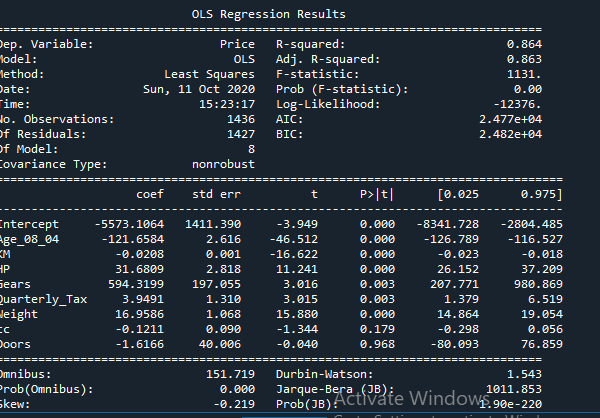
vif\_m2

2) Consider only the below columns and prepare a prediction model for predicting Price.

Model1:

In this model cc & price has the p value >0.05 so it is independent



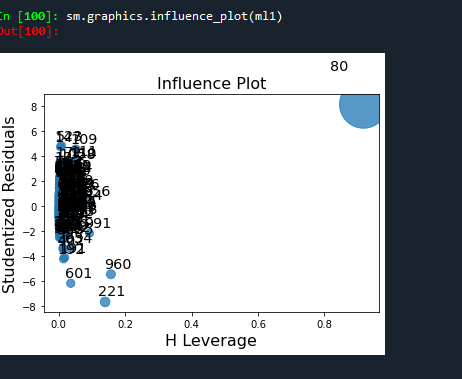


Model 2 for price and cc

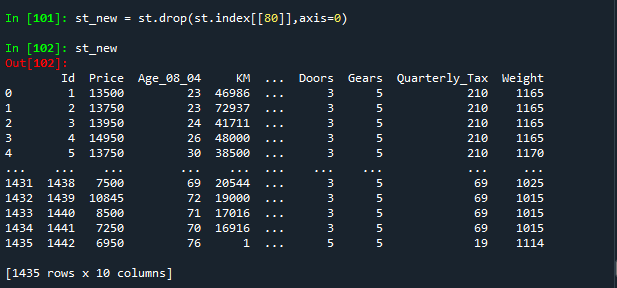


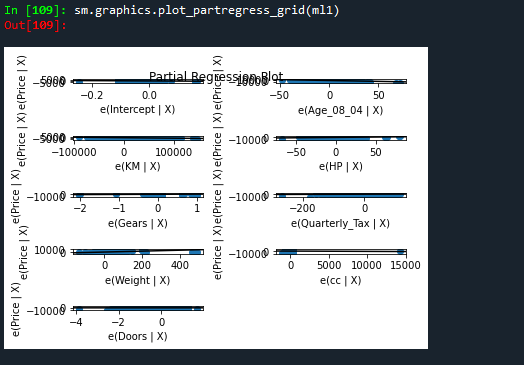
By applying this P values are less than 0.05 it is independent.

Creating an Influence plot

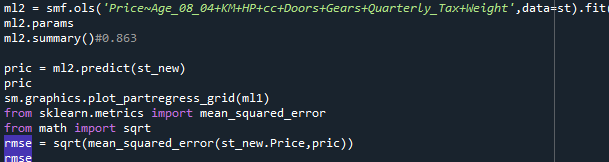


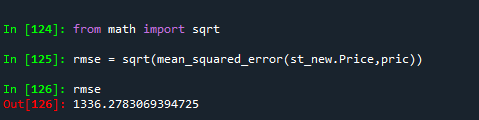
We remove 80 to get the better results.





RMSE Value:





FULL CODE:

# -\*- coding: utf-8 -\*-

"""

Created on Wed Oct 7 15:43:07 2020

@author: sunil

"""

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

st=pd.read\_csv("E://Excelr//excler//MLR assignment//ToyotaCorolla.csv")

st.columns

type(st)

st.corr()

import seaborn as sns

sns.pairplot(st)

import statsmodels.formula.api as smf

ml1 = smf.ols('Price~Age\_08\_04+KM+HP+Gears+Quarterly\_Tax+Weight',data=st).fit()

ml1.params

ml1.summary()

pred = pd.DataFrame(ml1.predict(st))

pred

st["pred"]= pred

m2 = smf.ols('Price~cc+Doors',data=st).fit()

m2.params

m2.summary()

import statsmodels.api as sm

sm.graphics.influence\_plot(ml1)

st\_new = st.drop(st.index[[80]],axis=0)

st\_new

ml2 = smf.ols('Price~Age\_08\_04+KM+HP+cc+Doors+Gears+Quarterly\_Tax+Weight',data=st).fit()

ml2.params

ml2.summary()#0.863

pric = ml2.predict(st\_new)

pric

sm.graphics.plot\_partregress\_grid(ml1)

from sklearn.metrics import mean\_squared\_error

from math import sqrt

rmse = sqrt(mean\_squared\_error(st\_new.Price,pric))

rmse