##### 2) Salary\_hike -> Build a prediction model for Salary\_hike

##### ------------------------------------------------------------

##### Build a simple linear regression model by performing EDA and do necessary transformations and select the best model using R or Python.

Answer :- Using Python.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

salaryhike = pd.read\_csv('/home/sushil/Documents/Assingment/SLR-ASS4/Salary\_Data.csv')

salaryhike

salaryhike.columns

salaryhike

x1 = plt.hist(salaryhike.YearsExperience)

x = plt.boxplot(salaryhike.YearsExperience)

y1 = plt.hist(salaryhike.Salary)

y = plt.boxplot(salaryhike.Salary)

salaryhike.isnull()

salaryhike.isnull().sum()

plt.plot(salaryhike.YearsExperience,salary\_hike.Salary,"bo");plt.xlabel("YearsExperience");plt.ylabel("Salary")

salaryhike.Salary.corr(salaryhike.YearsExperience)

# For preparing linear regression model

import statsmodels.formula.api as smf

model = smf.ols("Salary~YearsExperience",data = salaryhike).fit()

model.params

model.summary()

print(model.conf\_int(0.05))

pred = model.predict(salaryhike.iloc[:,0])

pred

plt.scatter(x = salaryhike['YearsExperience'],y = salaryhike['Salary'],color ='red');plt.plot(salaryhike['YearsExperience'],pred,color = 'black');plt.xlabel('Experience');plt.ylabel('Salary')

pred.corr(salaryhike.Salary)

# Transforming varaible for accuracy

model2 = smf.ols('Salary~np.log(YearsExperience)',data = salaryhike).fit()

model2.params

model2.summary()

print(model2.conf\_int(0.01))

pred2 = model2.predict(pd.DataFrame(salaryhike.iloc[:,0]))

pred2

pred2.corr(salaryhike.Salary)

plt.scatter(x = salaryhike['YearsExperience'],y = salaryhike['Salary'],color ='green');plt.plot(salaryhike['YearsExperience'],pred2,color = 'black');plt.xlabel('Experience');plt.ylabel('Salary')

Exponential transformation

model3 = smf.ols('np.log(Salary)~YearsExperience',data = salaryhike).fit()

model3.params

model3.summary()

print(model3.conf\_int(0.01))

pred\_log = model3.predict(pd.DataFrame(salaryhike.iloc[:,0]))

pred\_log

pred3 = np.exp(pred\_log)

pred3

pred3.corr(salaryhike.Salary)

plt.scatter(x = salaryhike['YearsExperience'],y = salaryhike['Salary'],color ='pink');plt.plot(salaryhike['YearsExperience'],pred3,color = 'black');plt.xlabel('Experience');plt.ylabel('Salary')

resid = model3.resid\_pearson

resid

plt.plot(model3.resid\_pearson,'o');plt.axhline(y=0,color='green');plt.xlabel('Observation number');plt.ylabel('Standarized Residual')

Predicted vs Actual value

plt.scatter(x=pred3,y=salaryhike.Salary);plt.xlabel('Predicted');plt.ylabel('Actual')

Quadratic model

salaryhike['YearsExperience\_sq'] = salaryhike.YearsExperience\*salaryhike.YearsExperience

model\_quad = smf.ols('np.log(Salary)~YearsExperience+YearsExperience\_sq',data = salaryhike).fit()

model\_quad.params

model\_quad.summary()

pred\_quad = model\_quad.predict(salaryhike)

pred\_quad

model\_quad.conf\_int(0.05)

plt.scatter(salaryhike.YearsExperience,salaryhike.Salary,c='b');plt.plot(salaryhike.YearsExperience,pred\_quad,"r")

plt.scatter(np.arange(30),model\_quad.resid\_pearson);plt.axhline(y=0,color = 'red')

plt.hist(model\_quad.resid\_pearson)

##### Result : Model1 is best model out of all model which is uesd to predict the salary hike in this linear regression model.

##### 1) Delivery\_time -> Predict delivery time using sorting time

##### ------------------------------------------------------------

##### Build a simple linear regression model by performing EDA and do necessary transformations and select the best model using R or Python.

Answer :- Using Python.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

delivery =pd.read\_csv('/home/sushil/Documents/Assingment/SLR-ASS4/delivery\_time.csv')

delivery

delivery.columns

delivery

delivery.columns = ['Del\_time','Sor\_time']

delivery

x1 = plt.hist(delivery.Sor\_time)

x = plt.boxplot(delivery.Sor\_time)

y1 = plt.hist(delivery.Del\_time)

y = plt.boxplot(delivery.Sor\_time)

delivery.isnull()

delivery.isnull().sum()

plt.plot(delivery.Sor\_time,delivery.Del\_time,"bo");plt.xlabel("Sor\_time");plt.ylabel("Del\_time")

delivery.Del\_time.corr(delivery.Sor\_time)

# For preparing linear regression model

import statsmodels.formula.api as smf

model = smf.ols("Del\_time~Sor\_time",data = delivery).fit()

model.params

model.summary()

print(model.conf\_int(0.05))

pred = model.predict(delivery.iloc[:,0])

pred

plt.scatter(x = delivery['Sor\_time'],y = delivery['Del\_time'],color ='red');plt.plot(delivery['Sor\_time'],pred,color = 'black');plt.xlabel('Sor\_time');plt.ylabel('Del\_time')

pred.corr(delivery.Del\_time)

# Transforming varaible for accuracy

model2 = smf.ols('Del\_time~np.log(Sor\_time)',data = delivery).fit()

model2.params

model2.summary()

print(model2.conf\_int(0.01))

pred2 = model2.predict(pd.DataFrame(delivery.iloc[:,0]))

pred2

pred2.corr(salaryhike.Salary)

plt.scatter(x = delivery['Sor\_time'],y = delivery['Del\_time'],color ='green');plt.plot(delivery['Sor\_time'],pred2,color = 'black');plt.xlabel('Sortingtime');plt.ylabel('Deliveringtime')

# Exponential transformation

model3 = smf.ols('np.log(Del\_time)~Sor\_time',data = delivery).fit()

model3.params

model3.summary()

print(model3.conf\_int(0.01))

pred\_log = model3.predict(pd.DataFrame(delivery.iloc[:,0]))

pred\_log

pred3 = np.exp(pred\_log)

pred3

plt.scatter(x = delivery['Sor\_time'],y = delivery['Del\_time'],color ='pink');plt.plot(delivery['Sor\_time'],pred3,color = 'black');plt.xlabel('Sortingtime');plt.ylabel('Deliverytime')

resid = model3.resid\_pearson

resid

plt.plot(model3.resid\_pearson,'o');plt.axhline(y=0,color='green');plt.xlabel('Observation number');plt.ylabel('Standarized Residual')

# Predicted vs Actual value

plt.scatter(x=pred3,y=delivery.Del\_time);plt.xlabel('Predicted');plt.ylabel('Actual')

# Quadratic model

delivery['Sor\_time\_sq'] = delivery.Sor\_time\*delivery.Sor\_time

model\_quad = smf.ols('np.log(Del\_time)~Sor\_time+Sor\_time\_sq',data = delivery).fit()

model\_quad.params

model\_quad.summary()

pred\_quad = model\_quad.predict(delivery)

pred\_quad

model\_quad.conf\_int(0.05)

plt.scatter(delivery.Sor\_time,delivery.Del\_time,c='b');plt.plot(delivery.Sor\_time,pred\_quad,"r")

plt.scatter(np.arange(30),model\_quad.resid\_pearson);plt.axhline(y=0,color = 'red')

plt.hist(model\_quad.resid\_pearson)

##### Result : Model3 is best model out of all model which is uesd to predict the delivery time in this linear regression model.