**Supervised Learning**

**Example Log Transformation**

**Import Libraries**

import seaborn as sns

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

**Load Data Set and make a copy**

tips =sns.load\_dataset('tips')

tips1= tips

print (tips1)

**Create Box plot to check outliers**

sns.boxplot (data = tips1 , x = 'day', y = 'total\_bill' )

plt.show()

**Create dis plot**

sns.displot(tips1['total\_bill'], kde = True)

plt.show()

**Apply log Transformation to address outliers**

tips1['total\_bill'] = np.log10(tips1['total\_bill'])

print (tips1)

**Create box plot and check outlier again**

sns.boxplot (data = tips1 , x = 'day', y = 'total\_bill' )

plt.show()

**Create dis plot**

sns.displot(tips1['total\_bill'], kde = True)

plt.show()

**Save the result in .xls**

tips1.to\_excel('C:\\Noble\\Training\\DS Temporary Files\\tips.xlsx')

**Simple Linear regression –**

**Import the Libraries**

import numpy as np

import pandas as pd

import os

import matplotlib.pyplot as plt

**Load the Data Set**

os.chdir('C:\\Noble\\Training\\Top Mentor\\Training\\Data Set\\')

os.getcwd()

df1= pd.read\_csv('Salary\_Data.csv')

print (df1)

**Split the data into x and y - Independent and Dependent variable**

**Create X**

x = df1.iloc[:,:-1].values

print (x)

**Create Y**

y = df1.iloc[:,1].values

print (y)

**create the graph to check the trend**

plt.plot(x,y)

plt.show()

**Split the Data – Train Test split**

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.2)

print ('X Shape-',x.shape)

print ('Y Shape-', y.shape)

print ('X\_Train Shape-',x\_train.shape)

print ('Y\_Train Shape-',y\_train.shape)

print ('X\_Test Shape-',x\_test.shape)

print ('Y\_Test Shape-',y\_test.shape)

**Model fitting**

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

lr.fit(x\_train, y\_train)

print (lr)

**Prediction**

y\_pred= lr.predict(x\_test)

print (y\_pred)

**Display the Prediction with Difference**

df\_x\_test =pd.DataFrame(x\_test, columns = ['Experience'])

df\_y\_test= pd.DataFrame(y\_test, columns = ['Salary'])

df\_y\_test\_pred = pd.DataFrame(y\_pred, columns = ['Prediction'])

df\_diff =df\_y\_test- df\_y\_test\_pred

y\_test\_pred = pd.concat ([df\_x\_test ,df\_y\_test, df\_y\_test\_pred ],axis =1)

y\_test\_pred['Difference'] =df\_y\_test ['Salary']- df\_y\_test\_pred['Prediction']

print (y\_test\_pred)

**Accuracy of the model**

from sklearn.metrics import r2\_score

accuracy = r2\_score(y\_test,y\_pred)

print (accuracy)

**Plot Graph- Test Data**

plt.scatter (x\_test, y\_test)

plt.plot (x\_test, y\_pred,'r')

plt.show()

**y= mx +c (Coefficient and Interceptor Values)**

**Y= slope**

print ('Coefficient', lr.coef\_)

print ('Intercept', lr.intercept\_)

**Prediction with Complete Data**

y\_pred\_final= lr.predict (x)

print (y\_pred\_final)

**Final Result in Data Frame**

y\_pred\_final = pd.DataFrame(y\_pred\_final,columns= ['Prediction'])

result = pd.concat([df1,y\_pred\_final], axis =1)

result['Difference'] = result['Salary'] - result['Prediction']

print (result)

result.to\_excel("C:\\Noble\\Training\\DS Temporary Files\\Simple Regression.xlsx")

**Accuracy of the model**

from sklearn.metrics import r2\_score

accuracy = r2\_score(y,y\_pred\_final)

print (accuracy)

**Create a Graph with predicted numbers**

plt.plot(x,y)

plt.plot (x,y\_pred\_final,'red' )

plt.show()

**Prediction for new set of data**

y\_pred= lr. predict ([[12], [9.6],[8.5], [2.5]])

print (y\_pred)

**Linear Regression Prediction with Data Frame**

Import Libraries

import numpy as np

import pandas as pd

import os

import matplotlib.pyplot as plt

Change directory

os.chdir('C:\\Noble\\Training\\Top Mentor\\Training\\Data Set\\')

os.getcwd()

Load Data Set

df1= pd.read\_csv('Salary\_Data.csv')

print (df1)

Create X – Independent Variable

x = df1.iloc[:,:-1]

print (x)

Create Y – Dependent Variable

y = df1.iloc[:,1]

print (y)

Plot Graph

plt.plot (x,y)

plt.show()

Train Test Split

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.2)

print ('X Shape-',x.shape)

print ('Y Shape-', y.shape)

print ('X\_Train Shape-',x\_train.shape)

print ('Y\_Train Shape-',y\_train.shape)

print ('X\_Test Shape-',x\_test.shape)

print ('Y\_Test Shape-',y\_test.shape)

Print X\_Test

print (x\_test)

Print Y\_Test

print (y\_test)

Linear Regression

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

lr.fit(x\_train, y\_train)

print (lr)

Prediction

y\_pred= lr.predict(x\_test)

print (y\_pred)

Display X\_Test

print (x\_test)

Reset Index

x\_test=x\_test.reset\_index()

print (x\_test)

Drop Index Column

x\_test=x\_test.drop('index',axis =1)

print (x\_test)

Display Y\_Test

print (y\_test)

Reset Index

y\_test=y\_test.reset\_index()

print (y\_test)

Drop Index Column

y\_test=y\_test.drop('index',axis =1)

print (y\_test)

Prediction as Data Frame

y\_pred = pd.DataFrame (y\_pred,columns = ['Prediction'])

print (y\_pred)

Display Test , Prediction and Difference

result = pd.concat ([x\_test,y\_test,y\_pred],axis =1)

result['Difference'] = y\_test['Salary']- y\_pred['Prediction']

print (result)

Accuracy

from sklearn.metrics import r2\_score

accuracy =r2\_score(y\_test,y\_pred)

print (accuracy)

Coefficient and Intercept

print ('Coefficient', lr.coef\_)

print ('Intercept', lr.intercept\_)

Export data to excel -Full Data Set

y\_pred\_final = pd.DataFrame(lr.predict(x),columns= ['Salary Prediction'])

result = pd.concat([df1,y\_pred\_final], axis =1)

result['Difference'] = df1['Salary']-y\_pred\_final['Salary Prediction']

print (result)

result.to\_excel("C:\\Noble\\Training\\DS Temporary Files\\Simple Regression.xlsx")

Accuracy

from sklearn.metrics import r2\_score

accuracy = r2\_score(df1['Salary'],y\_pred\_final)

print (accuracy)