Practical-3

AIM: Implementation of Simple Linear Regression using dataset3.xlsx, and Multiple Linear Regression and Polynomial Regression using dataset4.xlsx. Perform data preprocessing, model training, prediction, and evaluation for each regression type. Visualize the results and compare the performance of different regression techniques on the given datasets.

Simple Linear Regression

```
import pandas as pd import
numpy as np
from sklearn.model_selection import train_test_split
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
```

```
data = pd.read_excel('/content/drive/MyDrive/sem5/MACHINE LEARNING/Dataset/dataset3.xlsx')  x=data.iloc[:,0:-1]
y=data.iloc[:,-1]
```

```
y=data.iloc[:,-1]
```

```
x=data.iloc[:,0:-1]
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=1/3,random_state=0,)
x_train
```

	YearsExperience
5	2.9
16	5.1
8	3.2
14	4.5
23	8.2
20	6.8
1	1.3
29	10.5
6	3.0
4	2.2

```
y_test
```

```
array([ 37731, 122391,  57081,  63218, 116969, 109431, 112635,  55794,
        83088, 101302])
```

```
y_train
```

```
array([ 56642,  66029,  64445,  61111, 113812,  91738,  46205, 121872,
        60150,  39891,  81363,  93940,  57189,  54445, 105582,  43525,
        39343,  98273,  67938,  56957])
```

```
x_test
```

```
array([[ 1.5],
       [10.3],
       [ 4.1],
       [ 3.9],
       [ 9.5],
       [ 8.7],
       [ 9.6],
       [ 4. ],
```

```
from sklearn.linear_model import LinearRegression lr = LinearRegression()
lr.fit(x_train,y_train)
```

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
lr.predict(x_test.iloc[0].values.reshape(1,-1))
```

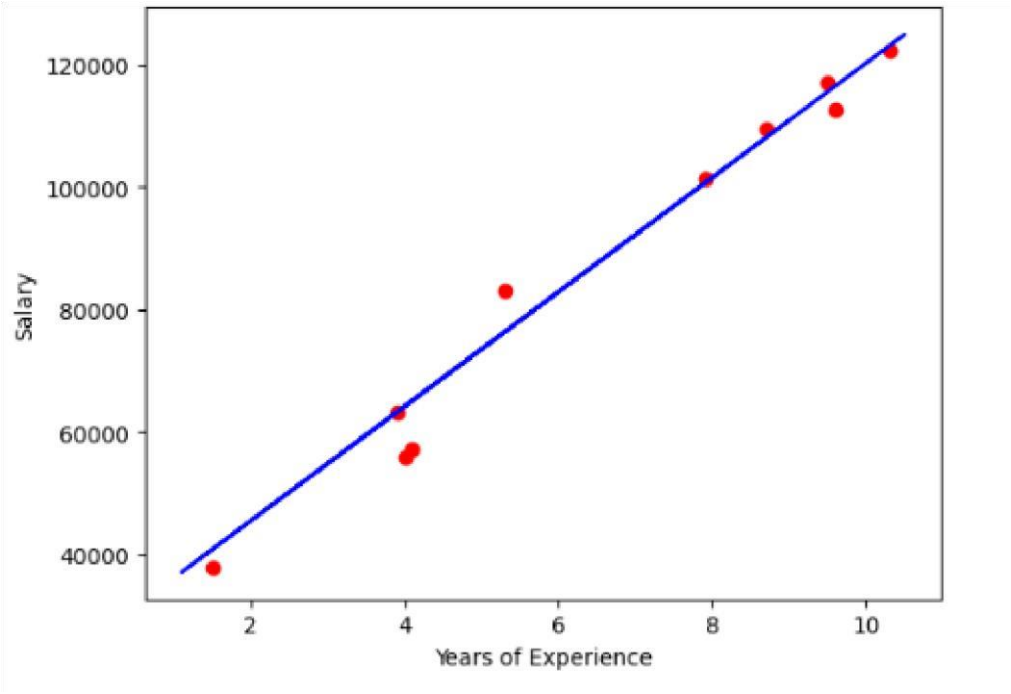
```
lr.predict(x_test.iloc[0].values.reshape(1,-1))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn,
  warnings.warn(
array([40835.10590871])
```

```
lr.predict(x_test.iloc[1].values.reshape(1,-1))
```

```
/usr/local/lib/python3.10/di
  warnings.warn(
array([123079.39940819])
```

```
import matplotlib.pyplot as plt
plt.scatter(x_test,y_test,color='red')
plt.plot(x_train,lr.predict(x_train),color='blue')
plt.xlabel('Years of Experience')  plt.ylabel('Salary')
plt.show()
```




```
m=lr.coef_
c=lr.intercept_
```

```
m*10+c
```

```
array([120275.61667525])
```

Multiple Linear Regression

```
import pandas as pd
data = pd.read_excel('/content/drive/MyDrive/sem5/MACHINE
LEARNING/Dataset/dataset4.xlsx') data.info()
```

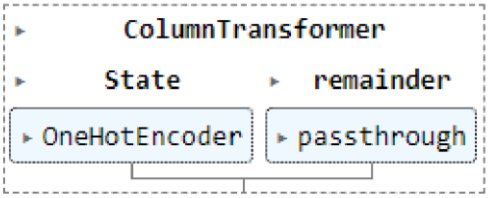
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   R&D Spend        50 non-null     float64
 1   Administration   50 non-null     float64
 2   Marketing Spend  50 non-null     float64
 3   State            50 non-null     object 
 4   Profit           50 non-null     float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

```
data['State'].isnull().sum() data
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77

```
x=data.iloc[:,:-1].values
y=data.iloc[:,-1].values
```

```
from sklearn.preprocessing import LabelEncoder,OneHotEncoder from
sklearn.compose import ColumnTransformer
ct =ColumnTransformer([("State",OneHotEncoder(),[3])],remainder="passthrough") ct
```




```
x = ct.fit_transform(x) x
```

```
array([[0.0, 0.0, 1.0, 165349.2, 136897.8, 471784.1],
       [1.0, 0.0, 0.0, 162597.7, 151377.59, 443898.53],
       [0.0, 1.0, 0.0, 153441.51, 101145.55, 407934.54],
       [0.0, 0.0, 1.0, 144372.41, 118671.85, 383199.62],
       [0.0, 1.0, 0.0, 142107.34, 91391.77, 366168.42],
       [0.0, 0.0, 1.0, 131876.9, 99814.71, 362861.36],
       [1.0, 0.0, 0.0, 134615.46, 147198.87, 127716.82],
       [0.0, 1.0, 0.0, 130298.13, 145530.06, 323876.68],
       [0.0, 0.0, 1.0, 120542.52, 148718.95, 311613.29],
       [1.0, 0.0, 0.0, 123334.88, 108679.17, 304981.62],
       [0.0, 1.0, 0.0, 101913.08, 110594.11, 229160.95],
       [1.0, 0.0, 0.0, 100671.96, 91790.61, 249744.55],
       [0.0, 1.0, 0.0, 93863.75, 127320.38, 249839.44],
       [1.0, 0.0, 0.0, 91992.39, 135495.07, 252664.93],
       [0.0, 1.0, 0.0, 119943.24, 156547.42, 256512.92],
       [0.0, 0.0, 1.0, 114523.61, 122616.84, 261776.23],
       [1.0, 0.0, 0.0, 78013.11, 121597.55, 264346.06],
       [0.0, 0.0, 1.0, 94657.16, 145077.58, 282574.31],
       [0.0, 1.0, 0.0, 91749.16, 114175.79, 294919.57],
       [0.0, 0.0, 1.0, 86419.7, 153514.11, 0.0],
       [1.0, 0.0, 0.0, 76253.86, 113867.3, 298664.47],
```

```
x=x[:,1:]
x
```

```
array([], shape=(10, 0), dtype=int64)
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x=sc.fit_transform(x) x
```

```
[ 1.39326109e+00,  1.02723599e-01,  1.16918609e+00,
  7.32787791e-01],
[-7.17740563e-01,  6.00657792e-03,  5.18495648e-02,
  7.62375876e-01],
[-7.17740563e-01, -1.36200724e-01, -5.62211268e-01,
  7.74348908e-01],
[ 1.39326109e+00,  7.31146008e-02, -7.95469167e-01,
 -5.81939297e-01],
[-7.17740563e-01, -1.99311688e-01,  6.56489139e-01,
 -6.03516725e-01],
[-7.17740563e-01,  3.53702028e-02,  8.21717916e-01,
 -6.35835495e-01],
[ 1.39326109e+00, -3.55189938e-02,  2.35068543e-01,
  1.17427116e+00],
[-7.17740563e-01, -1.68792717e-01,  2.21014050e+00,
 -7.67189437e-01],
[ 1.39326109e+00, -1.78608540e-01,  1.14245677e+00,
 -8.58133663e-01],
[-7.17740563e-01, -2.58074369e-01, -2.05628659e-01,
 -9.90357166e-01],
[ 1.39326109e+00, -2.76958231e-01,  1.13055391e+00,
 -1.01441945e+00],
[-7.17740563e-01, -2.26948675e-01,  2.83923813e-01,
 -1.36244978e+00]
```

```
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,random_state=0) from
sklearn.linear_model import LinearRegression lr=LinearRegression()
lr.fit(x_train,y_train)

lr.intercept_,lr.coef_
```

```
(111282.72816245264,
 array([  498.44578651, 35202.28572876,   882.97858029,  4326.03154922]))
```

```
lr.score(x_train,y_train)
```

```
0.9501020786438475
```

```
lr.score(x_test,y_test)
```

```
0.9367033175940501
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((40, 4), (10, 4), (40,), (10,))
```

```
import pickle with open('model.pkl','wb')
as f:
pickle.dump(lr,f)
```

□ Polynomial Regression

```
import pandas as pd import matplotlib.pyplot as plt  data
=pd.read_csv('/content/drive/MyDrive/sem5/MACHINE
LEARNING/Dataset/dataset5.csv') data
```

	Position	Level	Salary
0	Business Analyst	1	45000
1	Junior Consultant	2	50000
2	Senior Consultant	3	60000
3	Manager	4	80000
4	Country Manager	5	110000
5	Region Manager	6	150000
6	Partner	7	200000
7	Senior Partner	8	300000
8	C-level	9	500000
9	CEO	10	1000000

```
x=data.iloc[:,1:2].values
y=data.iloc[:,-1].values
```

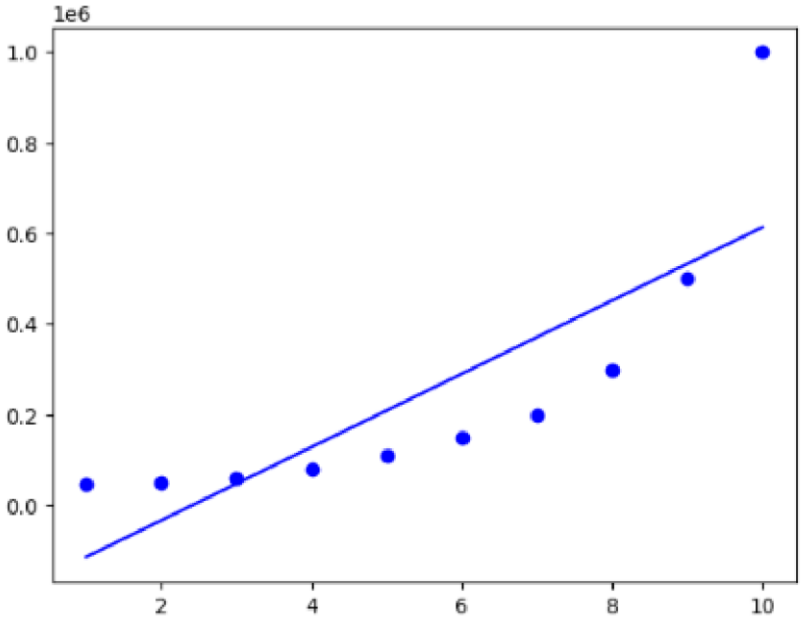
```
x
```

```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [ 9],
       [10]])
```

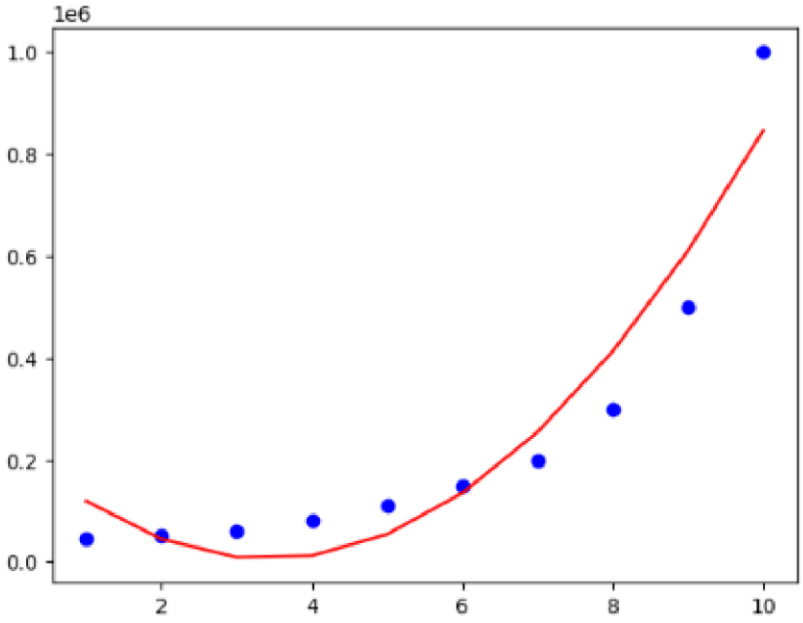
```
y
```

```
array([  45000,   50000,   60000,   80000,  110000,  150000,  200000,
        300000,  500000, 1000000])
```

```
from sklearn.linear_model import LinearRegression lr=LinearRegression()  lr.fit(x,y)
lr.score(x,y) lr.coef_ lr.intercept_
plt.plot(x,lr.predict(x),color='blue')
plt.scatter(x,y,color='blue')  plt.show()
```




```
from sklearn.preprocessing import PolynomialFeatures
pf=PolynomialFeatures(degree=2)
x_ploy=pf.fit_transform(x) lr2=LinearRegression()
lr2.fit(x_ploy,y) lr2.score(x_ploy,y) lr2.coef_,lr2.intercept_
plt.plot(x,lr2.predict(x_ploy),color='red')
plt.scatter(x,y,color='blue')  plt.show()
```


1e6
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10