Experiment-3

Title: Build a regression model using python on given data set by

a) Prepare the data for the ML Model.
b) Splitting Training data and Test data.
c) Evaluate the model (intercept and slope).
d) Visualize the training set and testing set using Matplotlib, Seaborn.
e) Predicting the test set result.
f) Compare actual output values with predicted values.

Context: Simple Linear Regression is one of the regression models available, using which we can predict the values of given test data using the training data. Here the outputs of training data are linearly dependent upon the inputs of it. We train the model using the training data and calculate the slope and the intercept of the best fit regression line and plot the line on a graph, this regression line gives us the values of the test outputs.

Dataset Description(For Linear Regression): The following is a list of most expensive association football transfers, which details the highest transfer fees ever paid for players, as well as transfers which set new world transfer records.

This data contains top 50 transfers based on paid fees.

Columns

- Rank: Rank of transfer
- Origin : Origin Country of Player
- Player : Player Name
- From(Country) : From which country the player transferred?
- From(Club) : From which Club the player transferred?
- To(Country) : To which country the player transferred?
- To(Club) : To which Club the player transferred?
- Position : Position of The player
- Fees(In million) : How much fee were paid for the transfer as in million.
- Year : In which year the transfer occured?
- Born : When was the player born?

Dataset Description(For Multiple Linear Regression): Using the advertising dataset given in ISLR I have analysed the relationship between 'TV advertising' and 'sales' using a simple multiple linear regression model.

Dataset Description(For Polynomial Regression): This dataset contains 62 rows and 2 coulmns consisting of Year and GDP respectively. Using this I have analysed the relationship between ‘Year’ and ‘GDP’ using a simple polynomial regression model.

Code & Output(For Linear Regression):

#Importing all the necessary libraries

```
import numpy as np
import seaborn as sns
import pandas as pd
```

```
import matplotlib.pyplot as plt
from sklearn import datasets, metrics
import io
from google.colab import files
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

uploaded=files.upload()
```

```
<IPython.core.display.HTML object>

Saving football_players.csv to football_players (1).csv
```

```
pv=pd.read_csv(io.BytesIO(uploaded['football_players.csv']),encoding='lati
n1',index_col=0)
pv
```

Rank	Origin	Player	From(Country)
1	Brazil	Neymar	Spain
2	France	Kylian Mbappé	France
3	Brazil	Philippe Coutinho	England
4	Portugal	João Félix	Portugal
5	France	Antoine Griezmann	Spain
6	England	Jack Grealish	England
7	France	Paul Pogba	Italy
7	France	Ousmane Dembélé	Germany
9	Wakes	Gareth Bale	England
10	Portugal	Cristiano Ronaldo	Spain
11	Belgium	Eden Hazard	England
12	Portugal	Cristiano Ronaldo	England
13	Argentina	Gonzalo Higuaín	Italy
14	England	Harry Maguire	England
15	Belgium	Romelu Lukaku	England
16	England	Jadon Sancho	Germany
17	Netherlands	Virgil van Dijk	England
18	Uruguay	Luis Suárez	England
19	Spain	Kepa Arrizabalaga	Spain
20	Ivory Coast	Nicolas Pépé	France
21	France	Lucas Hernandez	Spain
22	Belgium	Romelu Lukaku	England
23	Germany	Kai Havertz	Germany
24	France	Zinedine Zidane	Italy
25	Argentina	Ángel Di María	Spain
26	Netherlands	Matthijs de Ligt	Netherlands
27	Netherlands	Frenkie de Jong	Netherlands
28	Colombia	James Rodríguez	France
29	Belgium	Kevin De Bruyne	Germany
30	Brazil	Arthur	Spain
31	Spain	Rodri	Spain
32	Nigeria	Victor Osimhen	France
33	France	Thomas Lemar	France
34	Sweden	Sweden Zlatan Ibrahimovic	Italy
35	Portugal	Rúben Dias	Portugal
36	Algeria	Riyad Mahrez	England

37	Brazil	Kaká	Italy	
38	Spain	Álvaro Morata	Spain	
39	France	Aymeric Laporte	Spain	
40	Portugal	João Cancelo	Italy	
41	Uruguay	Edinson Cavani	Italy	
42	United States	Christian Pulisic	Germany	
42	Argentina	Ángel Di María	England	
44	Gabon	Pierre-Emerick Aubameyang	Germany	
45	England	Raheem Sterling	England	
45	Brazil	Alisson	Italy	
47	Portugal	Luís Figo	Spain	
48	Brazil	Oscar	England	
49	Serbia	Luka Jovic	Germany	
50	Morocco	Achraf Hakimi	Italy	
51	Guinea	Naby Keïta	Germany	
52	France	Tanguy Ndombele	France	
53	Colombia	Radamel Falcao	Spain	
54	Bosnia and Herzegovina	Miralem Pjanic	Italy	

Rank	From(Club)	To(Country)	To(Club)	Position
1	Barcelona	France	Paris Saint-Germain	Forward
2	Monaco	France	Paris Saint-Germain	Forward
3	Liverpool	Spain	Barcelona	Midfielder
4	Benfica	Spain	Atlético Madrid	Forward
5	Atlético Madrid	Spain	Barcelona	Forward
6	Aston Villa	England	Manchester City	Midfielder
7	Juventus	England	Manchester United	Midfielder
7	Borussia Dortmund	Spain	Barcelona	Forward
9	Tottenham Hotspur	Spain	Real Madrid	Forward
10	Real Madrid	Italy	Juventus	Forward
11	Chelsea	Spain	Real Madrid	Forward
12	Manchester United	Spain	Real Madrid	Forward
13	Napoli	Italy	Juventus	Striker
14	Leicester City	England	Manchester United	Defender
15	Everton	England	Manchester United	Striker
16	Borussia Dortmund	England	Manchester United	Midfielder
17	Southampton	England	Liverpool	Defender
18	Liverpool	Spain	Barcelona	Striker
19	Athletic Bilbao	England	Chelsea	Goalkeeper
20	Lille	England	Arsenal	Forward
21	Atlético Madrid	Germany	Bayern Munich	Defender
22	Manchester United	Italy	Internazionale	Striker
23	Bayer Leverkusen	England	Chelsea	Midfielder
24	Juventus	Spain	Real Madrid	Midfielder
25	Real Madrid	England	Manchester United	Midfielder
26	Ajax	Italy	Juventus	Defender
27	Ajax	Spain	Barcelona	Midfielder
28	Monaco	Spain	Real Madrid	Midfielder
29	VfL Wolfsburg	England	Manchester City	Midfielder
30	Barcelona	Italy	Juventus	Midfielder
31	Atlético Madrid	England	Manchester City	Midfielder
32	Lille	Italy	Napoli	Forward
33	Monaco	Spain	Atlético Madrid	Midfielder

34	Internazionale	Spain	Barcelona	Striker
35	Benfica	England	Manchester City	Defender
36	Leicester City	England	Manchester City	Forward
37	Milan	Spain	Real Madrid	Midfielder
38	Real Madrid	England	Chelsea	Striker
39	Athletic Bilbao	England	Manchester City	Defender
40	Juventus	England	Manchester City	Defender
41	Napoli	France	Paris Saint-Germain	Striker
42	Borussia Dortmund	England	Chelsea	Forward
42	Manchester United	France	Paris Saint-Germain	Midfielder
44	Borussia Dortmund	England	Arsenal	Striker
45	Liverpool	England	Manchester City	Forward
45	Roma	England	Liverpool	Goalkeeper
47	Barcelona	Spain	Real Madrid	Midfielder
48	Chelsea	China	Shanghai SIPG	Midfielder
49	Eintracht Frankfurt	Spain	Real Madrid	Striker
50	Internazionale	France	Paris Saint-Germain	Defender
51	RB Leipzig	England	Liverpool	Midfielder
52	Lyon	England	Tottenham Hotspur	Midfielder
53	Atlético Madrid	France	Monaco	Striker
54	Juventus	Spain	Barcelona	Midfielder

Rank	Fees(In Million)	Year
1	222.0	2017
2	180.0	2018
3	145.0	2018
4	126.0	2019
5	120.0	2019
6	117.0	2021
7	105.0	2016
7	105.0	2017
9	100.0	2013
10	100.0	2018
11	100.0	2019
12	94.0	2009
13	90.0	2016
14	87.0	2019
15	85.0	2017
16	85.0	2021
17	84.5	2018
18	82.3	2014
19	80.0	2018
20	80.0	2019
21	80.0	2019
22	80.0	2019
23	77.0	2020
24	76.0	2001
25	75.6	2014
26	75.0	2019
27	75.0	2019
28	75.0	2014
29	75.0	2015
30	72.0	2020

```
31                70.0  2019
32                70.0  2020
33                70.0  2018
34                69.5  2009
35                68.0  2020
36                67.8  2018
37                67.0  2009
38                65.5  2017
39                65.2  2018
40                65.0  2019
41                64.5  2013
42                64.0  2019
42                64.0  2015
44                63.7  2018
45                62.5  2015
45                62.5  2018
47                62.0  2000
48                61.0  2017
49                60.0  2019
50                60.0  2021
51                60.0  2018
52                60.0  2019
53                60.0  2013
54                60.0  2020
```

```
pv.head()
```

```
        Origin             Player From(Country)       From(Club)
To(Country)  \
Rank
1       Brazil             Neymar         Spain        Barcelona
France
2       France      Kylian Mbappé        France           Monaco
France
3       Brazil  Philippe Coutinho       England        Liverpool
Spain
4     Portugal         João Félix      Portugal          Benfica
Spain
5       France  Antoine Griezmann         Spain  Atlético Madrid
Spain

                 To(Club)    Position  Fees(In Million)  Year
Rank
1     Paris Saint-Germain     Forward             222.0  2017
2     Paris Saint-Germain     Forward             180.0  2018
3               Barcelona  Midfielder             145.0  2018
4         Atlético Madrid     Forward             126.0  2019
5               Barcelona     Forward             120.0  2019
```

```
pv.tail()
```

```
                     Origin           Player From(Country)
From(Club)  \
Rank
```

```
50                   Morocco    Achraf Hakimi         Italy
Internazionale
51                    Guinea       Naby Keïta       Germany       RB
Leipzig
52                    France  Tanguy Ndombele        France
Lyon
53                  Colombia   Radamel Falcao         Spain  Atlético
Madrid
54    Bosnia and Herzegovina   Miralem Pjanic         Italy
Juventus

     To(Country)             To(Club)    Position  Fees(In Million)  Year
Rank
50        France  Paris Saint-Germain    Defender              60.0  2021
51       England            Liverpool  Midfielder              60.0  2018
52       England    Tottenham Hotspur  Midfielder              60.0  2019
53        France               Monaco     Striker              60.0  2013
54         Spain            Barcelona  Midfielder              60.0  2020
```

```
pv.isnull().sum()
```

```
Origin              0
Player              0
From(Country)       0
From(Club)          0
To(Country)         0
To(Club)            0
Position            0
Fees(In Million)    0
Year                0
dtype: int64
```

```
pv["From(Country)"].unique()
```

```
array(['Spain', 'France', 'England', 'Portugal', 'Italy', 'Germany',
       'Netherlands'], dtype=object)
```

```
pv["From(Club)"].unique()
```

```
array(['Barcelona', 'Monaco', 'Liverpool', 'Benfica', 'Atlético Madrid',
       'Aston Villa', 'Juventus', 'Borussia Dortmund',
       'Tottenham Hotspur', 'Real Madrid', 'Chelsea', 'Manchester United',
       'Napoli', 'Leicester City', 'Everton', 'Southampton',
       'Athletic Bilbao', 'Lille', 'Bayer Leverkusen', 'Ajax',
       'VfL Wolfsburg', 'Internazionale', 'Milan', 'Roma',
       'Eintracht Frankfurt', 'RB Leipzig', 'Lyon'], dtype=object)
```

```
pv["To(Country)"].unique()
```

```
array(['France', 'Spain', 'England', 'Italy', 'Germany', 'China'],
      dtype=object)
```

```
pv["To(Club)"].unique()
```

```
array(['Paris Saint-Germain', 'Barcelona', 'Atlético Madrid',
       'Manchester City', 'Manchester United', 'Real Madrid', 'Juventus',
```

```
       'Liverpool', 'Chelsea', 'Arsenal', 'Bayern Munich',
       'Internazionale', 'Napoli', 'Shanghai SIPG', 'Tottenham Hotspur',
       'Monaco'], dtype=object)
```

```
pv["Position"].unique()
```

```
array(['Forward', 'Midfielder', 'Striker', 'Defender', 'Goalkeeper'],
      dtype=object)
```

```
pv["Fees(In Million)"].unique()
```

```
array([222. , 180. , 145. , 126. , 120. , 117. , 105. , 100. ,  94. ,
        90. ,  87. ,  85. ,  84.5,  82.3,  80. ,  77. ,  76. ,  75.6,
        75. ,  72. ,  70. ,  69.5,  68. ,  67.8,  67. ,  65.5,  65.2,
        65. ,  64.5,  64. ,  63.7,  62.5,  62. ,  61. ,  60. ])
```

```
pv["Year"].unique()
```

```
array([2017, 2018, 2019, 2021, 2016, 2013, 2009, 2014, 2020, 2001, 2015,
       2000])
```

```
pv["Fees(In Million)"].mean()
```

```
83.15925925925927
```

```
pv["Fees(In Million)"].median()
```

```
75.0
```

```
pv["Fees(In Million)"].mode()
```

```
0    60.0
dtype: float64
```

```
pv["Year"].mean()
```

```
2016.6296296296296
```

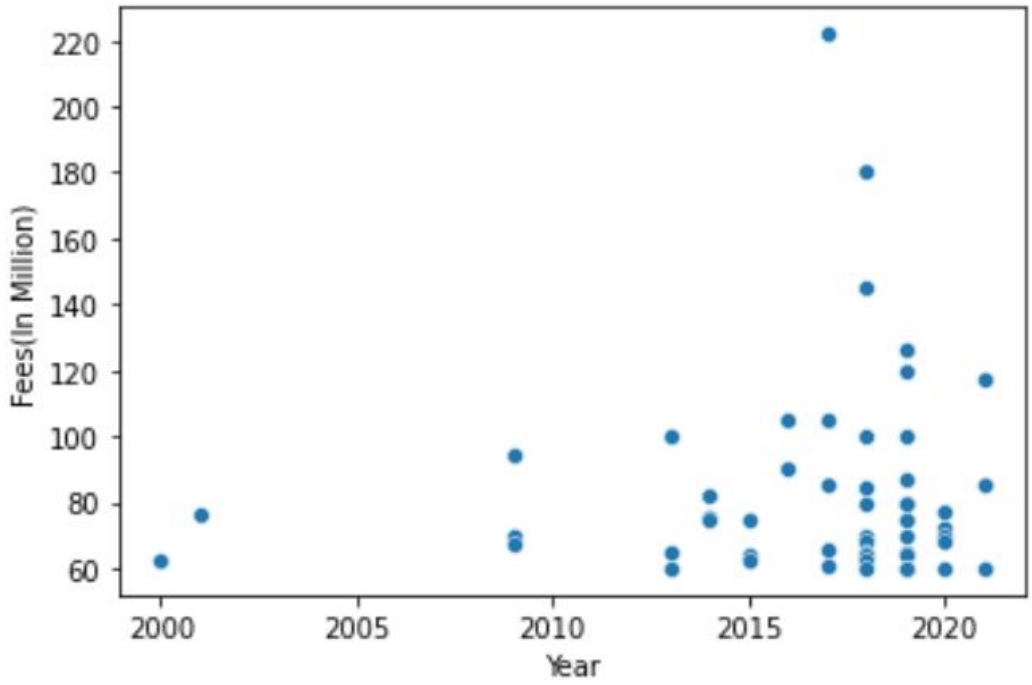
```
pv["Year"].median()
```

```
2018.0
```

```
pv["Year"].mode()
```

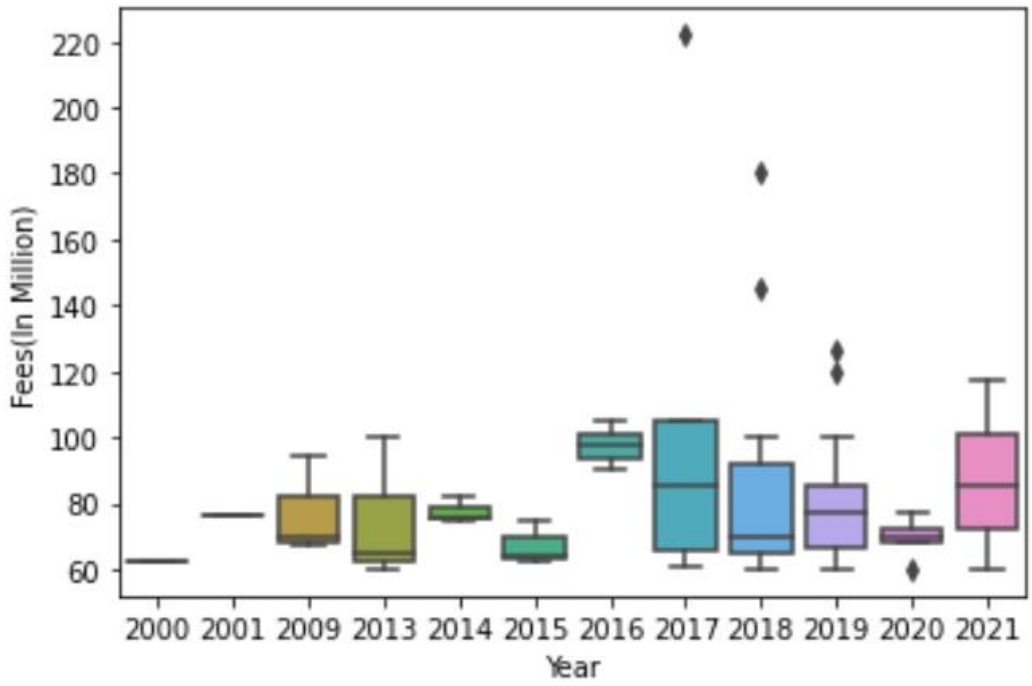
```
0    2019
dtype: int64
```

```
plt = sns.scatterplot(x="Year",y="Fees(In Million)",data=pv)
```




```
sns.boxplot(x="Year",y="Fees(In Million)",data=pv)
```

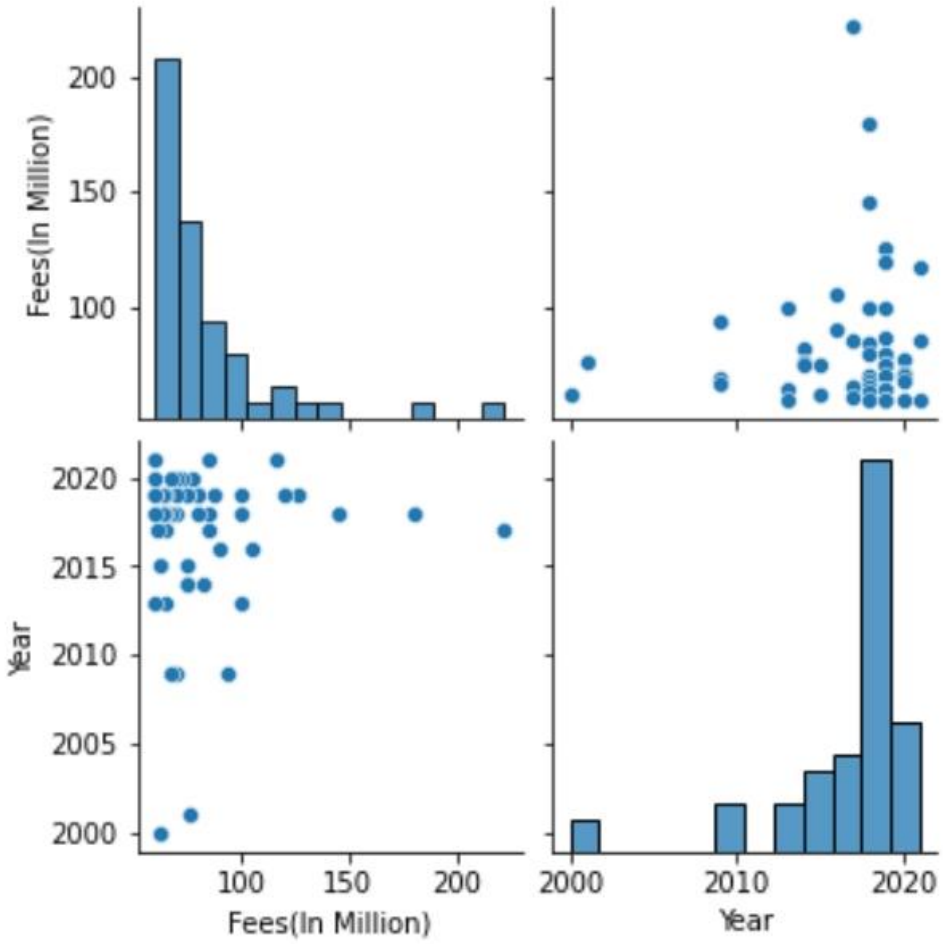
```
<matplotlib.axes._subplots.AxesSubplot at 0x7faa697e2280>
```




```
pv = pv[~pv.index.duplicated()]
```

```
sns.pairplot(data=pv,kind="scatter")
```

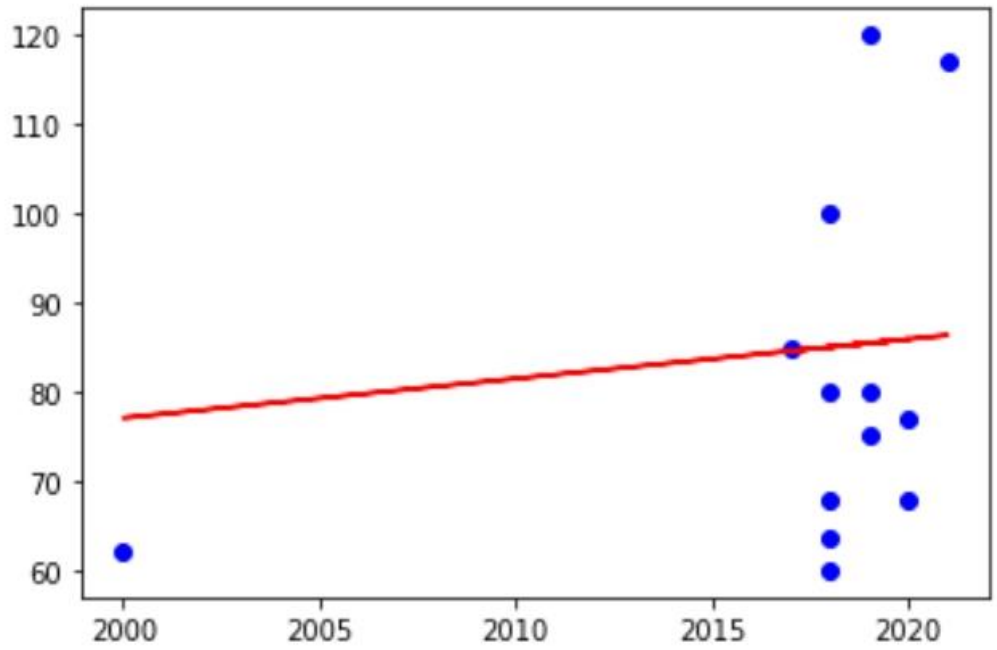
```
<seaborn.axisgrid.PairGrid at 0x7faa69628df0>
```




```
a=np.array(pv["Year"]).reshape(-1,1)
b=np.array(pv["Fees(In Million)"]).reshape(-1,1)
a_train, a_test, b_train, b_test = train_test_split(a, b, test_size=0.25)
r = LinearRegression()
r.fit(a_train, b_train)
print(r.score(a_test, b_test))
b_predict=r.predict(a_test)
plt.scatter(a_test,b_test,color='b')
plt.plot(a_test,b_predict,color="red")
```

```
0.03558560379192899
[[85.01441101]
 [86.33970372]
 [85.45617525]
 [85.89793949]
 [85.01441101]
 [85.01441101]
 [77.06265475]
 [85.45617525]
 [84.57264677]
 [85.01441101]
 [85.45617525]
```

```
 [85.89793949]
 [85.01441101]]
```




```
from sklearn.metrics import
mean_absolute_error,mean_squared_error,r2_score
mean_squared_error(b_test,b_predict)
```

```
1136.7936132571908
```

```
mean_absolute_error(b_test,b_predict)
```

```
19.873905838389508
```

Code & Output(For Multiple Linear Regression):

Importing all the necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

Importing the dataset

```
apy=pd.read_csv('advertising.csv')
data=apy.copy()
data
```

```
        TV  Radio  Newspaper  Sales
0    230.1   37.8       69.2   22.1
1     44.5   39.3       45.1   10.4
```

```
2     17.2   45.9       69.3   12.0
3    151.5   41.3       58.5   16.5
4    180.8   10.8       58.4   17.9
..     ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4

[200 rows x 4 columns]
```

```
data.columns
```

```
Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')
```

```
data.isnull().sum()
```

```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

Exploratory Data Analysis

```
data=data.dropna()
data.isnull().sum()
```

```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 7.8 KB
```

```
num=data.select_dtypes(exclude=[object])
num.corr(method='pearson')
num.corr()
```

```
                TV     Radio  Newspaper     Sales
TV        1.000000  0.054809   0.056648  0.901208
Radio     0.054809  1.000000   0.354104  0.349631
```

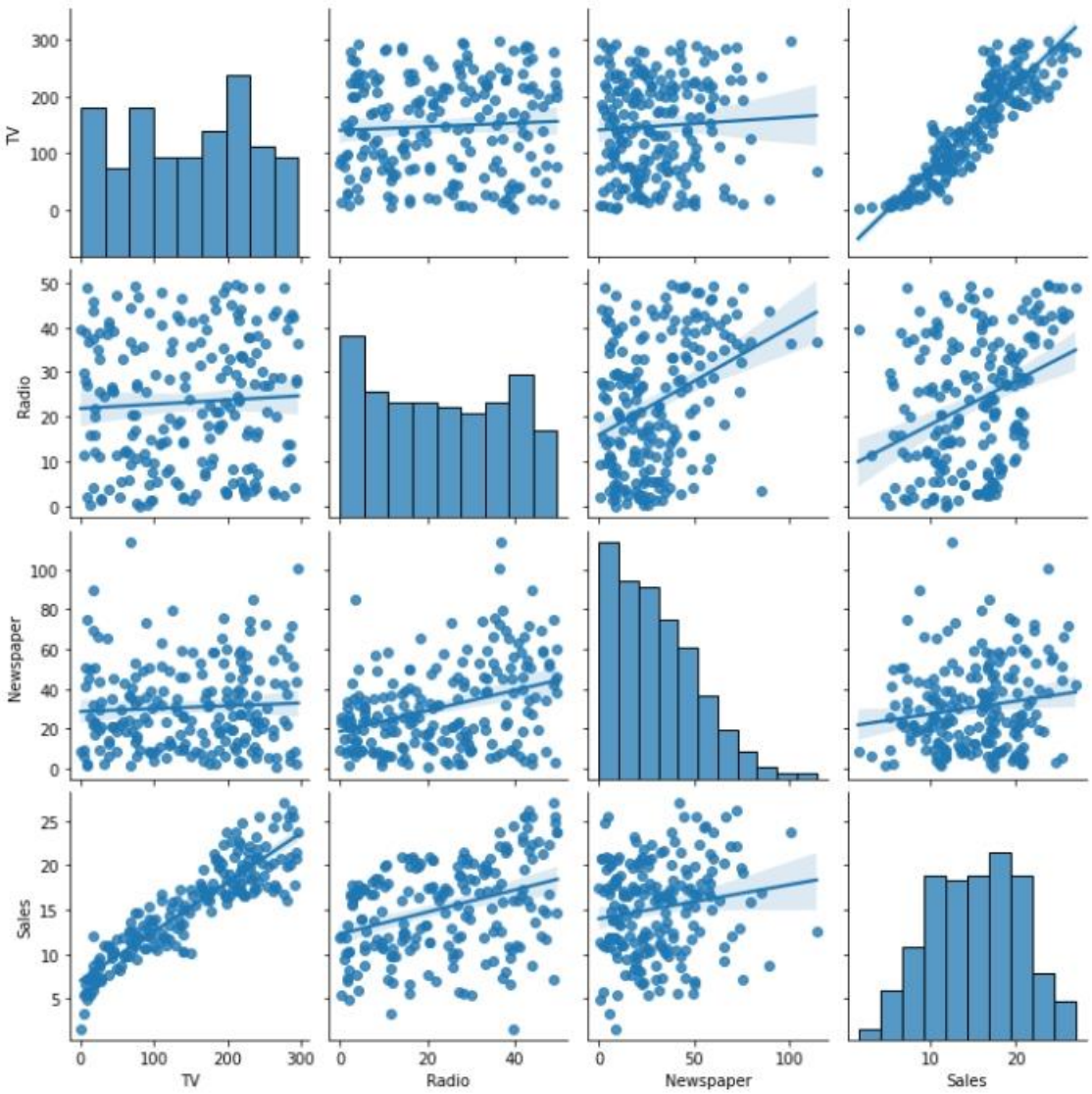
```
Newspaper  0.056648  0.354104   1.000000  0.157960
Sales      0.901208  0.349631   0.157960  1.000000
```

```
x=data.iloc[:,0:-1]
y=data.iloc[:,-1]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

Data Visualisation

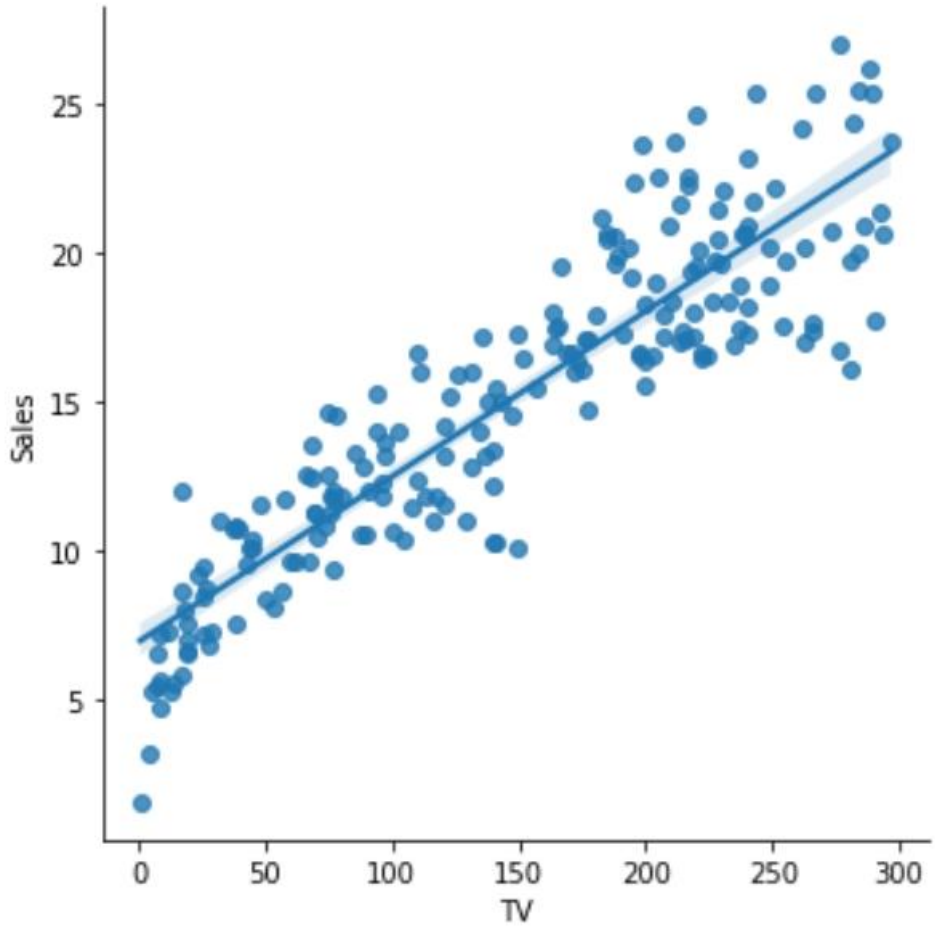
```
sns.pairplot(data=data,kind='reg')
```

```
<seaborn.axisgrid.PairGrid at 0x7f13ca56c880>
```



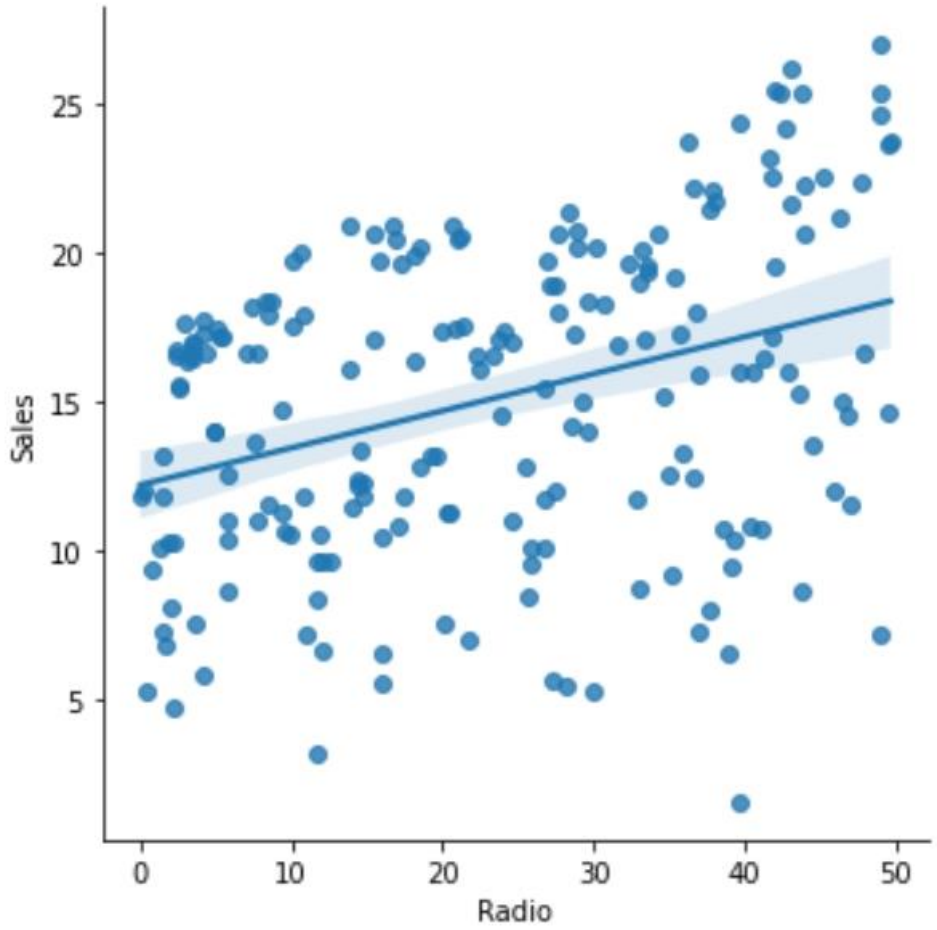

```
sns.lmplot(x='TV',y='Sales',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f13ca5ac9d0>
```



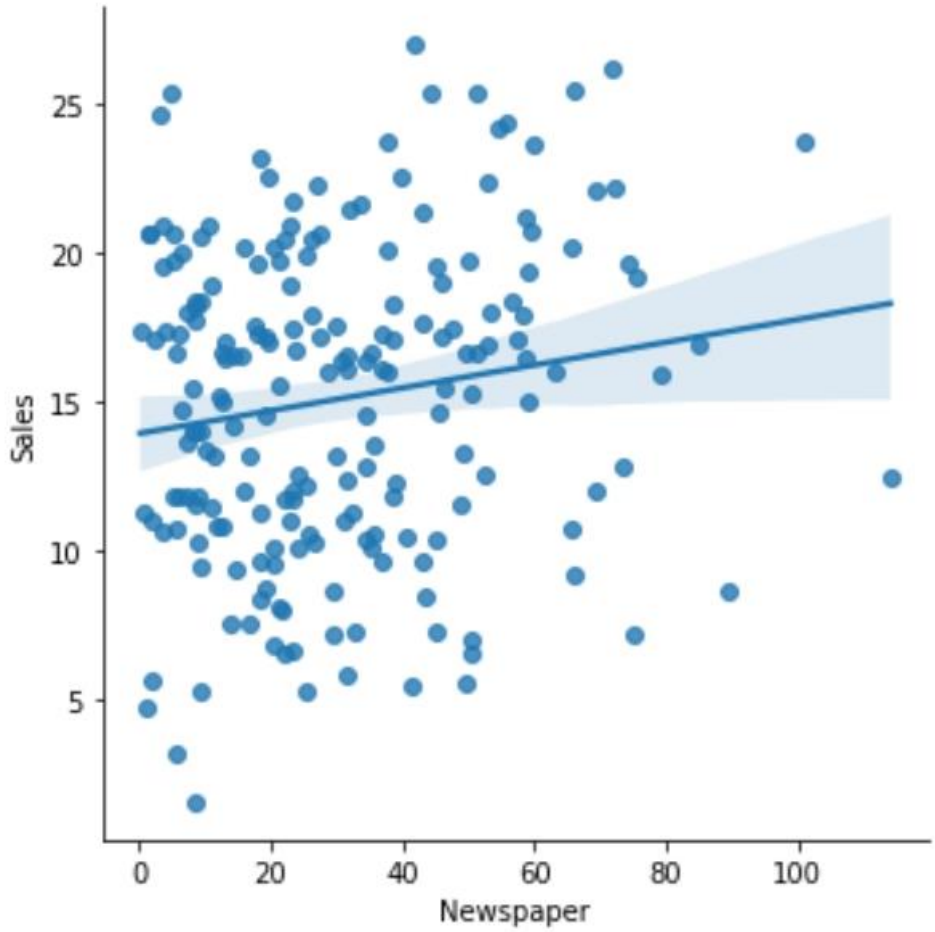

```
sns.lmplot(x='Radio',y='Sales',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f13c76b4250>
```




```
sns.lmplot(x='Newspaper',y='Sales',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f13dcd9d0a0>
```



Linear Regression Model

```
lgr=LinearRegression()
l1=lgr.fit(x_train,y_train)
pred=lgr.predict(x_test)
```

Checking the accuracy

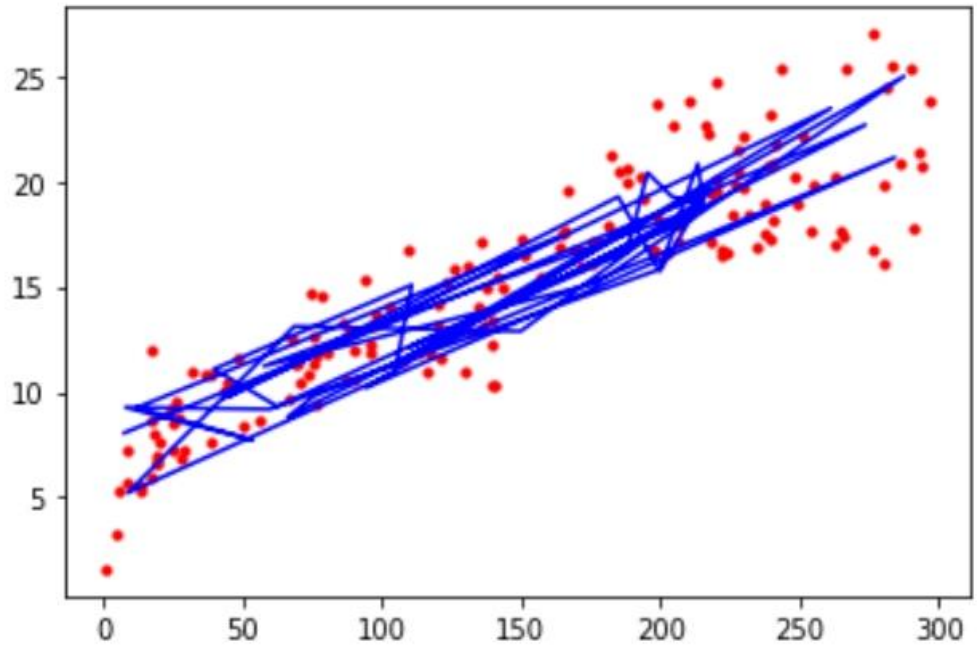
```
r1=l1.score(x_test,y_test)
r2=l1.score(x_train,y_train)
print(r1,r2)
```

```
0.906255073962913 0.9013103082057977
```

Visualising the Regression line

```
plt.scatter(x_train['TV'],y_train,color='red',s=10)
plt.plot(x_test['TV'],pred,color='blue')
```

```
[<matplotlib.lines.Line2D at 0x7f13c27721f0>]
```




```
mse=metrics.mean_squared_error(y_test,pred)
rmse=np.sqrt(mse)
mae=metrics.mean_absolute_error(y_test,pred)
print(mse)
print(rmse)
print(mae)
```

```
2.6047209330198813
1.6139147849313114
1.2478046140985302
```

```
l1.coef_
```

```
array([0.05434053, 0.10682914, 0.00318254])
```

```
l1.intercept_
```

```
4.536203134599694
```

Code & Output(For Polynomial Regression):

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import io
from google.colab import files
uploaded=files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving GDP.csv to GDP.csv
```

```
df=pd.read_csv(io.BytesIO(uploaded["GDP.csv"]))
df
```

```
    Year      GDP
0   2021  3173.40
1   2020  2667.69
2   2019  2831.55
3   2018  2702.93
4   2017  2651.47
..   ...      ...
57  1964    56.48
58  1963    48.42
59  1962    42.16
60  1961    39.23
61  1960    37.03

[62 rows x 2 columns]
```

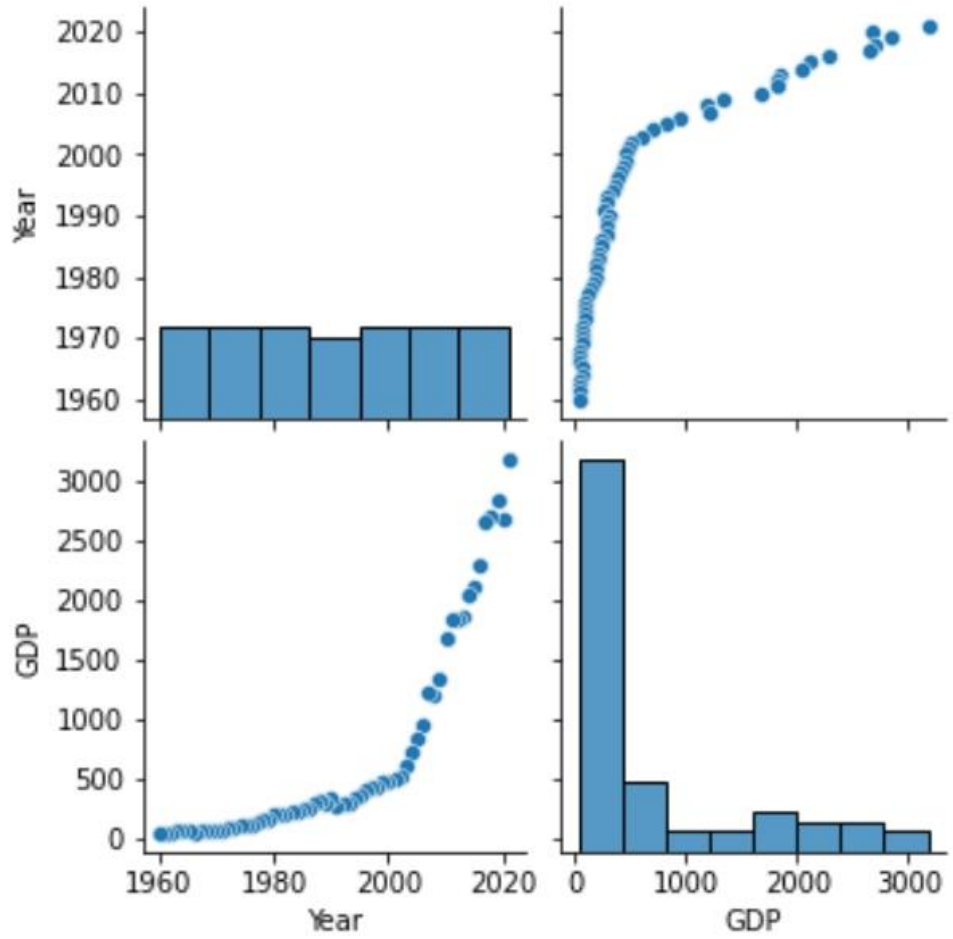
```
x=df["Year"].values.reshape(-1,1)
y=df["GDP"].values.reshape(-1,1)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Year    62 non-null     int64  
 1   GDP     62 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 1.1 KB
```

```
sb.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f9990ffdf10>
```



Training Linear reg model with the dataset

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x,y)
```

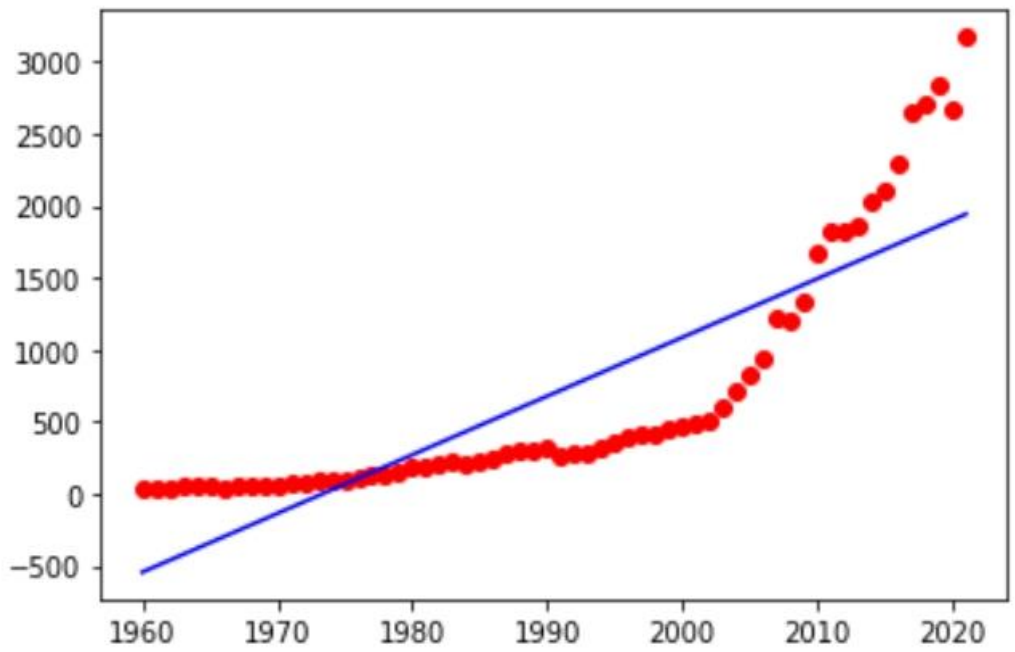
```
LinearRegression()
```

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=4)
x_poly=poly_reg.fit_transform(x)
lin_reg=LinearRegression()
lin_reg.fit(x_poly,y)
```

```
LinearRegression()
```

```
lr = LinearRegression()
lr.fit(x,y)
```

```
plt.scatter(x,y,color="red")
plt.plot(x,lr.predict(x),color="blue")
plt.show()
```




```
plt.scatter(x,y,color="red")
plt.plot(x,lin_reg.predict(poly_reg.fit_transform(x)),color="blue")
plt.title("GDp vs ")
plt.xlabel("position level")
plt.ylabel("salary")
plt.show()
```