

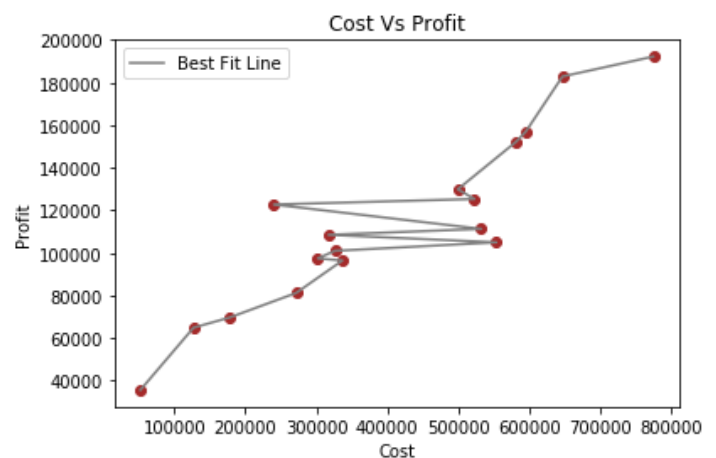
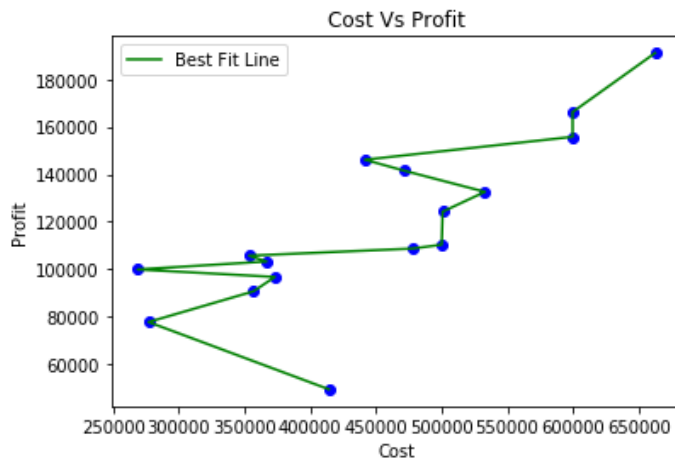
QUESTION 01: Take 50 startups of any two countries and find out which country is going to provide best profit in future.

```

1 # SARAH SALEEM AKHTER
2 # Using Decision Tree Regression
3 # 50 Startups
4 # Taking data of Florida and NewYork
5
6 # Importing Libraries |
7 import matplotlib.pyplot as plt
8 import pandas as pd
9
10 #Importing dataset for Florida
11 dataset = pd.read_csv('50_Startups_Flor.csv')
12 v1 = dataset.iloc[:, 0:1].values #R&D Spend
13 v2 = dataset.iloc[:, 1:2].values #Administration
14 v3 = dataset.iloc[:, 2:3].values #Marketing Spend
15 sum = v1 + v2 + v3
16 profit = dataset.iloc[:, 4].values #Profit Generated
17
18 # Splitting the dataset into the Training set and Test set
19 from sklearn.model_selection import train_test_split
20 sum_train, sum_test, profit_train, profit_test = train_test_split(sum, profit, test_size = 0.2, random_state = 0)
21
22 #Fitting Florida's dataset by Decision Tree Regression
23 from sklearn.tree import DecisionTreeRegressor
24 RegrVaria = DecisionTreeRegressor(random_state = 0)
25 RegrVaria.fit (sum , profit)
26
27 # Predicting new result for Florida
28 Pred_Flor = RegrVaria.predict ([[8000000]])
29
30 plt.scatter(sum, profit, color = 'blue')
31 plt.plot(sum, RegrVaria.predict(sum), color = 'green', label = 'Best Fit Line')
32 plt.title('Cost Vs Profit')
33 plt.xlabel('Cost')
34 plt.ylabel('Profit')
35 plt.legend()
36 plt.show()
37
38 |
39 #Importing dataset for NewYork
40 dataset = pd.read_csv('50_Startups_NY.csv')
41 v1 = dataset.iloc[:, 0:1].values #R&D Spend
42 v2 = dataset.iloc[:, 1:2].values #Administration
43 v3 = dataset.iloc[:, 2:3].values #Marketing Spend
44 sum = v1 + v2 + v3
45 profit = dataset.iloc[:, 4].values #Profit Generated
46
47 # Splitting the dataset into the Training set and Test set
48 from sklearn.model_selection import train_test_split
49 sum_train, sum_test, profit_train, profit_test = train_test_split(sum, profit, test_size = 0.2, random_state = 0)
50
51 #Fitting NewYork's dataset by Decision Tree Regression
52 from sklearn.tree import DecisionTreeRegressor
53 RegrVaria = DecisionTreeRegressor(random_state = 0)
54 RegrVaria.fit (sum , profit)
55
56 # Predicting new result for NewYork
57 Pred_NY = RegrVaria.predict ([[8000000]])
58
59 plt.scatter(sum, profit, color = 'brown')
60 plt.plot(sum, RegrVaria.predict(sum), color = 'grey', label = 'Best Fit Line')
61 plt.title('Cost Vs Profit')
62 plt.xlabel('Cost')
63 plt.ylabel('Profit')
64 plt.legend()
65 plt.show()
66
67 print("Predicted Profit for FLorida: ", Pred_Flor)
68 print("Predicted Profit for NewYork: ", Pred_NY)

```

OUTPUTS:



Predicted Profit for FLorida: [191050.39]

Predicted Profit for NewYork: [192261.83]

Therefore, as shown in the graph, NewYork would provide better results in future than Florida.

QUESTION 02: Annual temperature between two industries is given. Predict the temperature in 2016 and 2017 using the past data of both country.

```
1 #SARAH SALEEM AKHTER
2 # Using Polynomial Regression to predict the temperature in 2016 and 2017 using the past data of both industries
3
4 #===== FOR GCAG=====
5 # Importing the Libraries and the dataset for evaluation
6 import numpy as np
7 import matplotlib.pyplot as plt
8 import pandas as pd
9
10 dataset = pd.read_csv('ann_temp_gcag.csv')
11 year = dataset.iloc[:, 1:2].values
12 mean = dataset.iloc[:, 2:3].values
13
14 #Fitting Linear Regression to the dataset
15 from sklearn.linear_model import LinearRegression
16 lin_regr = LinearRegression()
17 lin_regr.fit(year, mean)
18
19 ## Fitting Polynomial Regression to the dataset
20 from sklearn.preprocessing import PolynomialFeatures
21 poly_regr = PolynomialFeatures(degree = 4)
22 year_poly = poly_regr.fit_transform(year)
23 poly_regr.fit(year_poly, mean)
24 lin_regr_2 = LinearRegression()
25 lin_regr_2.fit(year_poly, mean)
26
27 #Visualising the Linear Regression results
28 plt.scatter(year, mean, color= 'purple')
29 plt.plot(year, lin_regr.predict(year), color = 'blue', label = 'The Best Fit Line')
30 plt.title('year vs mean temperature (linear regression for GCAG)')
31 plt.xlabel('Years')
32 plt.ylabel('Mean Temperature')
33 plt.legend()
34 plt.show()
35
```

```

36 #Visualising the Polynomial Regression results
37 plt.scatter(year, mean, color= 'green')
38 plt.plot(year, lin_regr_2.predict(poly_regr.fit_transform(year)), color = 'blue', label = 'The Best Fit Line')
39 plt.title('year vs mean temperature (polynomial regression for GCAG)')
40 plt.xlabel('Years')
41 plt.ylabel('Mean Temperature')
42 plt.legend()
43 plt.show()
44
45 #Predicting a new result with Linear Regression
46 temp_2016 = lin_regr.predict([[2016]])
47 temp_2017 = lin_regr.predict([[2017]])
48
49 #Predicting a new result with Polynomial Regression
50 temp_poly_2016 = lin_regr_2.predict(poly_regr.fit_transform([[2016]]))
51 temp_poly_2017 = lin_regr_2.predict(poly_regr.fit_transform([[2017]]))
52
53 print("Temperatures for GCAG ")
54 print("A/c to linear regression, Temperature in 2016 will be: ", temp_2016)
55 print("A/c to polynomial regression, Temperature in 2016 will be: ", temp_poly_2016)
56 print("A/c to linear regression, Temperature in 2017 will be: ", temp_2017)
57 print("A/c to polynomial regression, Temperature in 2017 will be: ", temp_poly_2017)
58
59 #===== FOR GISTEMP=====
60 # Importing the libraries and the dataset for evaluation
61 import numpy as np
62 import matplotlib.pyplot as plt
63 import pandas as pd
64
65 dataset = pd.read_csv('ann_temp_gistemp.csv')
66 year = dataset.iloc[:, 1:2].values
67 mean = dataset.iloc[:, 2:3].values
68
69 #Fitting Linear Regression to the dataset
70 from sklearn.linear_model import LinearRegression
71 lin_regr = LinearRegression()
72 lin_regr.fit(year, mean)

```

```

73
74 ## Fitting Polynomial Regression to the dataset
75 from sklearn.preprocessing import PolynomialFeatures
76 poly_regr = PolynomialFeatures(degree = 4)
77 year_poly = poly_regr.fit_transform(year)
78 poly_regr.fit(year_poly, mean)
79 lin_regr_2 = LinearRegression()
80 lin_regr_2.fit(year_poly, mean)
81
82 #Visualising the Linear Regression results
83 plt.scatter(year, mean, color= 'purple')
84 plt.plot(year, lin_regr.predict(year), color = 'blue', label = 'The Best Fit Line')
85 plt.title('year vs mean temperature (linear regression for GISTEMP)')
86 plt.xlabel('Years')
87 plt.ylabel('Mean Temperature')
88 plt.legend()
89 plt.show()
90
91 #Visualising the Polynomial Regression results
92 plt.scatter(year, mean, color= 'green')
93 plt.plot(year, lin_regr_2.predict(poly_regr.fit_transform(year)), color = 'blue', label = 'The Best Fit Line')
94 plt.title('year vs mean temperature (polynomial regression for GISTEMP)')
95 plt.xlabel('Years')
96 plt.ylabel('Mean Temperature')
97 plt.legend()
98 plt.show()
99
100 #Predicting a new result with Linear Regression
101 temp2016 = lin_regr.predict([[2016]])
102 temp2017 = lin_regr.predict([[2017]])
103
104 #Predicting a new result with Polynomial Regression
105 temp_poly2016 = lin_regr_2.predict(poly_regr.fit_transform([[2016]]))
106 temp_poly2017 = lin_regr_2.predict(poly_regr.fit_transform([[2017]]))
107

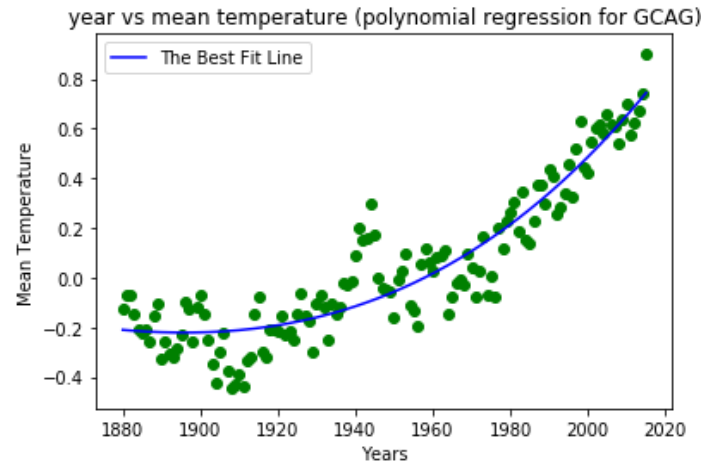
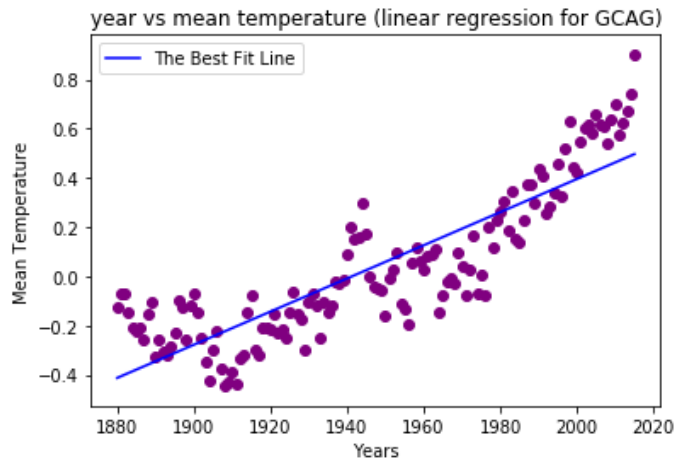
```

```

108 print("Temperatures for GISTEMP ")
109 print("A/c to linear regression, Temperature in 2016 will be: ", temp2016)
110 print("A/c to polynomial regression, Temperature in 2016 will be: ", temp_poly2016)
111 print("A/c to linear regression, Temperature in 2017 will be: ", temp2017)
112 print("A/c to polynomial regression, Temperature in 2017 will be: ", temp_poly2017)
'''

```

OUTPUTS:



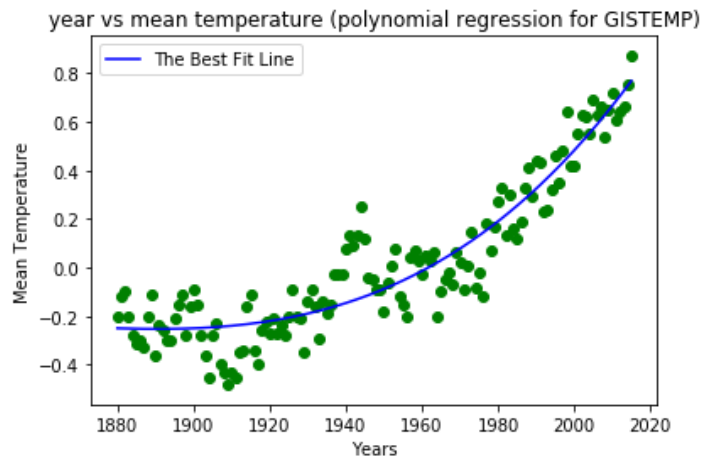
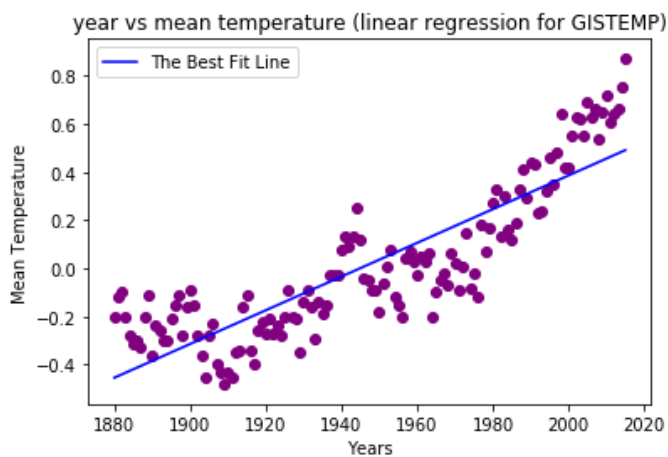
Temperatures for GCAG

A/c to linear regression, Temperature in 2016 will be: `[[0.50298425]]`

A/c to polynomial regression, Temperature in 2016 will be: `[[0.76231028]]`

A/c to linear regression, Temperature in 2017 will be: `[[0.50972011]]`

A/c to polynomial regression, Temperature in 2017 will be: `[[0.78149969]]`



Temperatures for GISTEMP

A/c to linear regression, Temperature in 2016 will be: `[[0.49777778]]`

A/c to polynomial regression, Temperature in 2016 will be: `[[0.78885745]]`

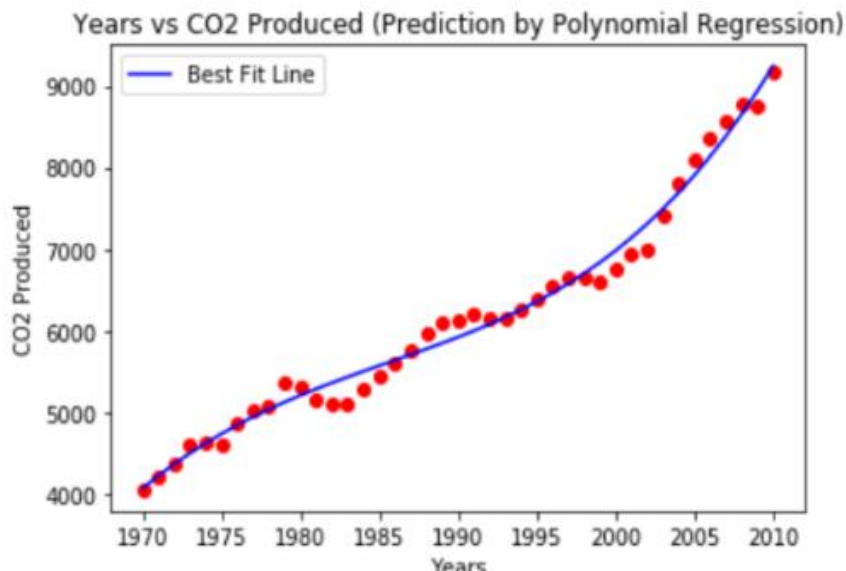
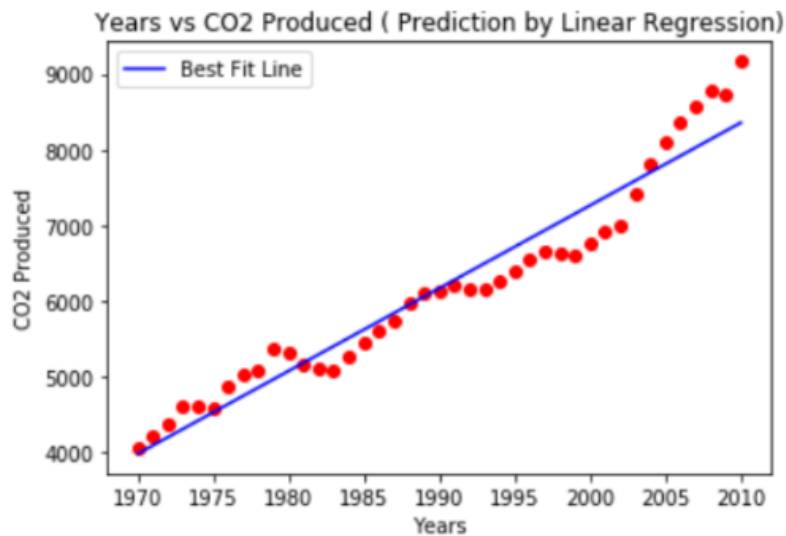
A/c to linear regression, Temperature in 2017 will be: `[[0.50477625]]`

A/c to polynomial regression, Temperature in 2017 will be: `[[0.81039365]]`

QUESTION 03: Data of global production of CO2 of a place is given between 1970s to 2010. Predict the CO2 production for the years 2011, 2012 and 2013 using the old data set.

```
1 #SARAH SALEEM AKHTER
2 #Using Polynomial Regression to predict the temperature of CO2
3
4 #Importing the libraries for the dataset
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import pandas as pd
8
9 dataset = pd.read_csv('global_co2.csv')
10 year = dataset.iloc[:, 0:1].values
11 total = dataset.iloc[:, 1:2].values
12
13 #Fitting linear regression to the dataset
14 from sklearn.linear_model import LinearRegression
15 lin_regr = LinearRegression()
16 lin_regr.fit(year, total)
17
18 #Fitting Polynomial Regression to the dataset
19 from sklearn.preprocessing import PolynomialFeatures
20 poly_regr = PolynomialFeatures(degree=5)
21 year_poly = poly_regr.fit_transform(year)
22 poly_regr.fit = (year_poly, total)
23 lin_regr_2 = LinearRegression()
24 lin_regr_2.fit(year_poly, total)
25
26 #Visualising the Linear Regression results
27 plt.scatter(year, total, color= 'red')
28 plt.plot(year, lin_regr.predict(year), color = 'blue', label = 'Best Fit Line')
29 plt.title('Years vs CO2 Produced (Prediction by Linear Regression)')
30 plt.xlabel('Years')
31 plt.ylabel('CO2 Produced')
32 plt.legend()
33 plt.show()
34
35 #Visualising the Polynomial Regression results
36 plt.scatter(year, total, color= 'red')
37 plt.plot(year, lin_regr_2.predict(poly_regr.fit_transform(year)), color = 'blue', label = 'Best Fit Line')
38 plt.title('Years vs CO2 Produced (Prediction by Polynomial Regression)')
39 plt.xlabel('Years')
40 plt.ylabel('CO2 Produced')
41 plt.legend()
42 plt.show()
43
44 #Predicting a new result with linear regression
45 Prod_CO2_11 = lin_regr.predict([[2011]])
46 Prod_CO2_12 = lin_regr.predict([[2012]])
47 Prod_CO2_13 = lin_regr.predict([[2013]])
48
49 #Predicting a new result with polynomial regression
50 Prod_CO2_poly11 = lin_regr_2.predict(poly_regr.fit_transform([[2011]]))
51 Prod_CO2_poly12 = lin_regr_2.predict(poly_regr.fit_transform([[2012]]))
52 Prod_CO2_poly13 = lin_regr_2.predict(poly_regr.fit_transform([[2013]]))
53
54
55 print("Prediction for Production of CO2 in 2011: ")
56 print("A/c to Linear Regression, CO2 produced in 2011 will be: ", Prod_CO2_11)
57 print("A/c to polynomial Regression, CO2 produced in 2011 will be: ", Prod_CO2_poly11)
58
59 print("Prediction for Production of CO2 in 2012: ")
60 print("A/c to Linear Regression, CO2 produced in 2012 will be: ", Prod_CO2_12)
61 print("A/c to polynomial Regression, CO2 produced in 2012 will be: ", Prod_CO2_poly12)
62
63 print("Prediction for Production of CO2 in 2013: ")
64 print("A/c to Linear Regression, CO2 produced in 2013 will be: ", Prod_CO2_13)
65 print("A/c to polynomial Regression, CO2 produced in 2013 will be: ", Prod_CO2_poly13)
```


OUTPUTS:



A/c to Linear Regression:

The Carbon dioxide production in 2011 will be:

[[4494.86418176]]

The Carbon dioxide production in 2012 will be:

[[4518.55824859]]

The Carbon dioxide production in 2013 will be:

[[4542.25231541]]

A/c to Polynomial Regression:

The Carbon dioxide production in 2011 will be:

[[4494.86418176]]

The Carbon dioxide production in 2012 will be:

[[4518.55824859]]

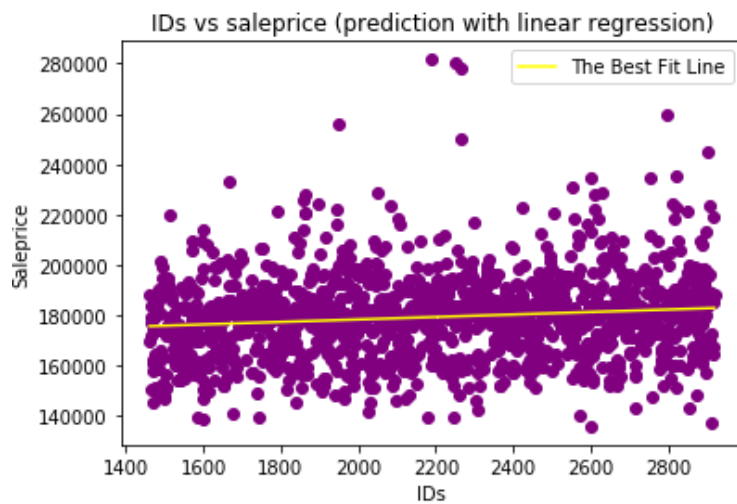
The Carbon dioxide production in 2013 will be:

[[4542.25231541]]

QUESTION 04: Housing price according to the ID is assigned to every-house. Perform future analysis where when ID is inserted the housing price is displayed.

```
1 #SARAH SALEEM AKHTER
2 #Using Polynomial Regression to predict the Housing price according to the ID is assigned to every-house.
3
4 #Importing the libraries for the dataset
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import pandas as pd
8
9 dataset = pd.read_csv('housing_price.csv')
10 ids = dataset.iloc[:, 0:1].values
11 saleprice = dataset.iloc[:, 1:2].values
12
13 #Fitting linear regression to the dataset
14 from sklearn.linear_model import LinearRegression
15 lin_regr = LinearRegression()
16 lin_regr.fit(ids, saleprice)
17
18 #Fitting Polynomial Regression to the dataset
19 from sklearn.preprocessing import PolynomialFeatures
20 poly_regr = PolynomialFeatures(degree=5)
21 ids_poly = poly_regr.fit_transform(ids)
22 poly_regr.fit = (ids_poly, saleprice)
23 lin_regr_2 = LinearRegression()
24 lin_regr_2.fit(ids_poly, saleprice)
25
26 #Visualising the Linear Regression results
27 plt.scatter(ids, saleprice, color= 'purple')
28 plt.plot(ids, lin_regr.predict(ids), color = 'yellow', label = 'The Best Fit Line')
29 plt.title('IDs vs saleprice (prediction with linear regression)')
30 plt.xlabel('IDs')
31 plt.ylabel('Saleprice')
32 plt.legend()
33 plt.show()
34
35 #Visualising the Polynomial Regression results
36 plt.scatter(ids, saleprice, color= 'red')
37 plt.plot(ids, lin_regr_2.predict(poly_regr.fit_transform(ids)), color = 'blue', label = 'The Best Fit Line')
38 plt.title('IDs vs saleprice (prediction with polynomial regression)')
39 plt.xlabel('IDs')
40 plt.ylabel('Saleprice')
41 plt.legend()
42 plt.show()
43
44 #Predicting a new result with Linear Regression
45 hp2920 = lin_regr.predict([[2920]])
46 hp2925 = lin_regr.predict([[2925]])
47 hp2930 = lin_regr.predict([[2930]])
48
49
50 #Predicting a new result with Polynomial Regression
51 hp_poly2920 = lin_regr_2.predict(poly_regr.fit_transform([[2920]]))
52 hp_poly2925 = lin_regr_2.predict(poly_regr.fit_transform([[2925]]))
53 hp_poly2930 = lin_regr_2.predict(poly_regr.fit_transform([[2930]]))
54
55 print("A/c to Linear Regression: ")
56 print("The Housing Price for ID number 2920 will be= ", hp2920)
57 print("The Housing Price for ID number 2925 will be= ", hp2925)
58 print("The Housing Price for ID number 2930 will be= ", hp2930)
59
60 print("A/c to polynomial Regression: ")
61 print("The Housing Price for ID number 2920 will be= ", hp_poly2920)
62 print("The Housing Price for ID number 2925 will be= ", hp_poly2925)
63 print("The Housing Price for ID number 2930 will be= ", hp_poly2930)
```

OUTPUTS:



Traceback (most recent call last):

```
File "<ipython-input-2-32a2fd0927c2>", line 1, in <module>
    runfile('C:/Users/Sarah/Desktop/intro to python/Part 3 - Artificial Neural Networks (ANN)/housing price.py', wdir='C:/Users/Sarah/Desktop/intro to python/Part 3 - Artificial Neural Networks (ANN)')

File "C:\ProgramData\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 827, in runfile
    execfile(filename, namespace)

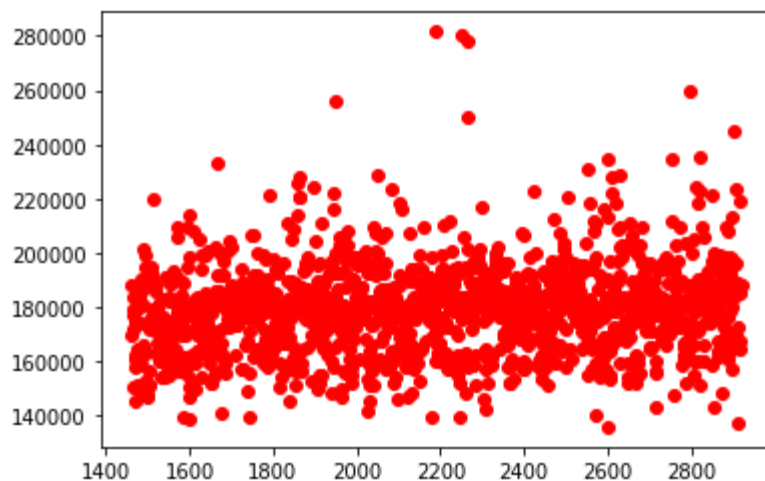
File "C:\ProgramData\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/Sarah/Desktop/intro to python/Part 3 - Artificial Neural Networks (ANN)/housing price.py", line 37, in <module>
    plt.plot(ids, lin_regr_2.predict(poly_regr.fit_transform(ids)), color = 'blue', label = 'The Best Fit Line')

File "C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py", line 553, in fit_transform
    return self.fit(X, **fit_params).transform(X)
```

TypeError: 'tuple' object is not callable

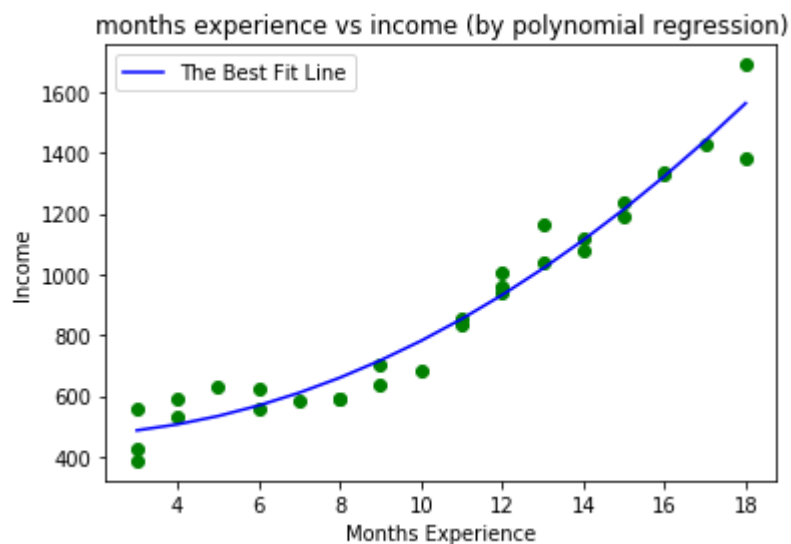
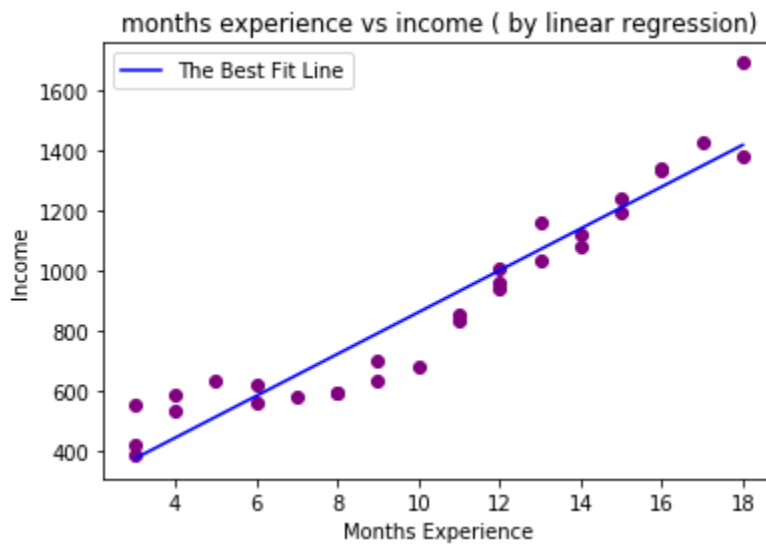
TypeError: 'tuple' object is not callable



QUESTION 05: Data of monthly experience and income distribution of different employs is given.
Perform regression.

```
1 #SARAH SALEEM AKHTER
2 #Using Polynomial Regression to predict the Data of monthly experience and income distribution of different employees
3
4 # Importing the libraries and the dataset for evaluation
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import pandas as pd
8
9 dataset = pd.read_csv('monthlyexp vs incom.csv')
10 months_exp = dataset.iloc[:, 0:1].values
11 income = dataset.iloc[:, 1:2].values
12
13 #Fitting Linear Regression to the dataset
14 from sklearn.linear_model import LinearRegression
15 lin_regr = LinearRegression()
16 lin_regr.fit(months_exp, income)
17
18 ## Fitting Polynomial Regression to the dataset
19 from sklearn.preprocessing import PolynomialFeatures
20 poly_regr = PolynomialFeatures(degree = 2)
21 months_exp_poly = poly_regr.fit_transform(months_exp)
22 poly_regr.fit(months_exp_poly, income)
23 lin_regr_2 = LinearRegression()
24 lin_regr_2.fit(months_exp_poly, income)
25
26 #Visualising the Linear Regression results
27 plt.scatter(months_exp, income, color= 'purple')
28 plt.plot(months_exp, lin_regr.predict(months_exp), color = 'blue', label = 'The Best Fit Line')
29 plt.title('months experience vs income ( by linear regression)')
30 plt.xlabel('Months Experience')
31 plt.ylabel('Income')
32 plt.legend()
33 plt.show()
34
35 #Visualising the Polynomial Regression results
36 plt.scatter(months_exp, income, color= 'green')
37 plt.plot(months_exp, lin_regr_2.predict(poly_regr.fit_transform(months_exp)), color = 'blue', label = 'The Best Fit Line')
38 plt.title('months experience vs income (by polynomial regression)')
39 plt.xlabel('Months Experience')
40 plt.ylabel('Income')
41 plt.legend()
42 plt.show()
43
44 #Predicting a new result with Linear Regression
45 exp_19 = lin_regr.predict([[19]])
46 exp_20 = lin_regr.predict([[20]])
47 exp_21 = lin_regr.predict([[21]])
48
49 #Predicting a new result with Polynomial Regression
50 exp_poly19 = lin_regr_2.predict(poly_regr.fit_transform([[19]]))
51 exp_poly20 = lin_regr_2.predict(poly_regr.fit_transform([[20]]))
52 exp_poly21 = lin_regr_2.predict(poly_regr.fit_transform([[21]]))
53
54 print("By Linear Regression: ")
55 print("Income= ", exp_19)
56 print("Income= ", exp_20)
57 print("Income= ", exp_21)
58
59 print("By Polynomial Regression: ")
60 print("Income= ", exp_poly19)
61 print("Income= ", exp_poly20)
62 print("Income= ", exp_poly21)
```

OUTPUTS:



By Linear Regression:

Income= [[1486.2554603]]

Income= [[1555.62625026]]

Income= [[1624.99704021]]

By Polynomial Regression:

Income= [[1693.95330533]]

Income= [[1832.49031114]]

Income= [[1978.4483702]]