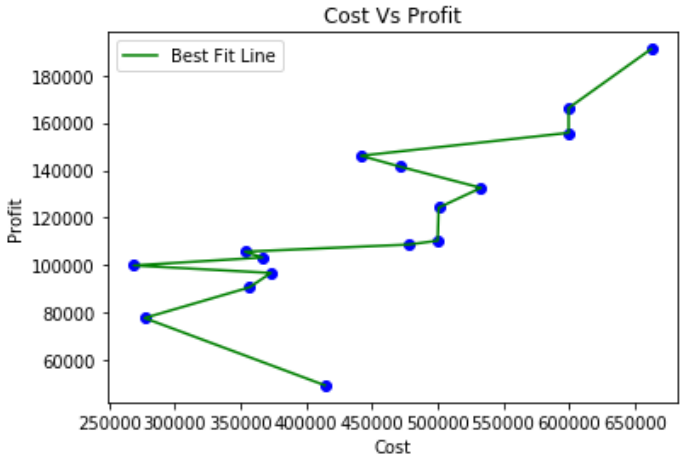
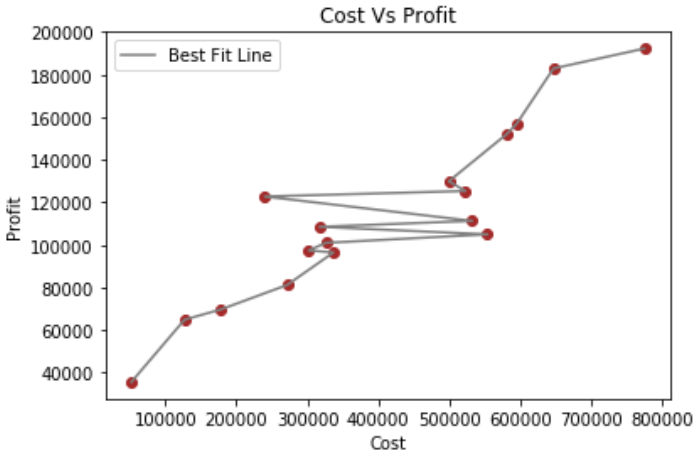
**QUESTION 01: Take 50 startups of any two countries and find out which country is going to provide best profit in future.**

```
# SARAH SALEEM AKHTER
# Using Decision Tree Regression
# 50 Startups
# Taking data of Florida and NewYork

# Importing Libraries
import matplotlib.pyplot as plt
import pandas as pd

#Importing dataset for Florida
dataset = pd.read_csv('50_Startups_Flor.csv')
v1 = dataset.iloc[:, 0:1].values #R&D Spend
v2 = dataset.iloc[:, 1:2].values #Administration
v3 = dataset.iloc[:, 2:3].values #Marketing Spend
sum = v1 + v2 + v3
profit = dataset.iloc[:, 4].values #Profit Generated

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
sum_train, sum_test, profit_train, profit_test = train_test_split(sum, profit, test_size = 0.2, random_state = 0)

#Fitting Florida's dataset by Decision Tree Regression
from sklearn.tree import DecisionTreeRegressor
RegrVaria = DecisionTreeRegressor(random_state = 0)
RegrVaria.fit (sum , profit)

# Predicting new result for Florida
Pred_Flor = RegrVaria.predict ([[8000000]])

plt.scatter(sum, profit, color = 'blue')
plt.plot(sum, RegrVaria.predict(sum), color = 'green', label = 'Best Fit Line')
plt.title('Cost Vs Profit')
plt.xlabel('Cost')
plt.ylabel('Profit')
plt.legend()
plt.show()

#Importing dataset for NewYork
dataset = pd.read_csv('50_Startups_NY.csv')
v1 = dataset.iloc[:, 0:1].values #R&D Spend
v2 = dataset.iloc[:, 1:2].values #Administration
v3 = dataset.iloc[:, 2:3].values #Marketing Spend
sum = v1 + v2 + v3
profit = dataset.iloc[:, 4].values #Profit Generated

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
sum_train, sum_test, profit_train, profit_test = train_test_split(sum, profit, test_size = 0.2, random_state = 0)

#Fitting NewYork's dataset by Decision Tree Regression
from sklearn.tree import DecisionTreeRegressor
RegrVaria = DecisionTreeRegressor(random_state = 0)
RegrVaria.fit (sum , profit)

# Predicting new result for NewYork
Pred_NY = RegrVaria.predict ([[8000000]])

plt.scatter(sum, profit, color = 'brown')
plt.plot(sum, RegrVaria.predict(sum), color = 'grey', label = 'Best Fit Line')
plt.title('Cost Vs Profit')
plt.xlabel('Cost')
plt.ylabel('Profit')
plt.legend()
plt.show()

print("Predicted Profit for FLorida: ", Pred_Flor)
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.**

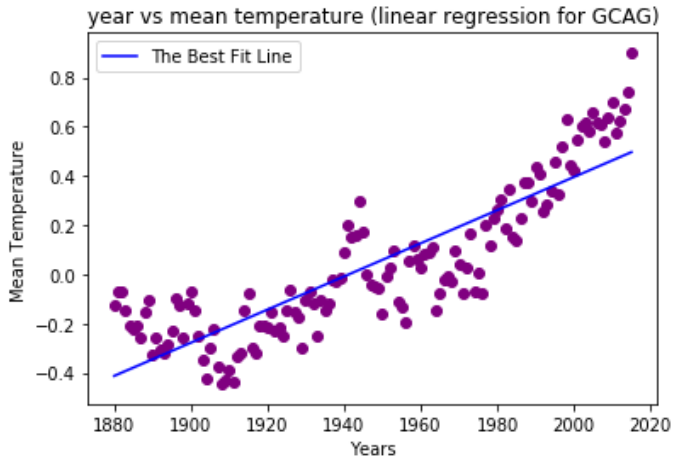
```
#SARAH SALEEM AKHTER
# Using Polynomial Regression to predict the temperature in 2016 and 2017 using the past data of both industries

#================================ FOR GCAG=====================================
# Importing the libraries and the dataset for evaluation
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('ann_temp_gcag.csv')
year = dataset.iloc[:, 1:2].values
mean = dataset.iloc[:, 2:3].values

#Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin_regr = LinearRegression()
lin_regr.fit(year, mean)

## Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
poly_regr = PolynomialFeatures(degree = 4)
year_poly = poly_regr.fit_transform(year)
poly_regr.fit(year_poly, mean)
lin_regr_2 = LinearRegression()
lin_regr_2.fit(year_poly, mean)

#Visualising the Linear Regression results
plt.scatter(year, mean, color= 'purple')
plt.plot(year, lin_regr.predict(year), color = 'blue', label = 'The Best Fit Line')
plt.title('year vs mean temperature (linear regression for GCAG)')
plt.xlabel('Years')
plt.ylabel('Mean Temperature')
plt.legend()
plt.show()
```

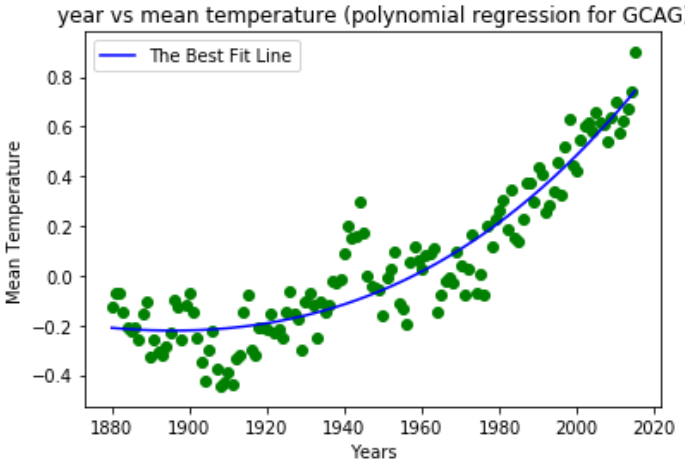
```
#Visualising the Polynomial Regression results
plt.scatter(year, mean, color= 'green')
plt.plot(year, lin_regr_2.predict(poly_regr.fit_transform(year)), color = 'blue', label = 'The Best Fit Line')
plt.title('year vs mean temperature (polynomial regression for GCAG)')
plt.xlabel('Years')
plt.ylabel('Mean Temperature')
plt.legend()
plt.show()

#Predicting a new result with Linear Regression
temp_2016 = lin_regr.predict([[2016]])
temp_2017 = lin_regr.predict([[2017]])

#Predicting a new result with Polynomial Regression
temp_poly_2016 = lin_regr_2.predict(poly_regr.fit_transform([[2016]]))
temp_poly_2017 = lin_regr_2.predict(poly_regr.fit_transform([[2017]]))

print("Temperatures for GCAG ")
print("A/c to linear regression, Temperature in 2016 will be: ", temp_2016)
print("A/c to polynomial regression, Temperature in 2016 will be: ", temp_poly_2016)
print("A/c to linear regression, Temperature in 2017 will be: ", temp_2017)
print("A/c to polynomial regression, Temperature in 2017 will be: ", temp_poly_2017)

#============================== FOR GISTEMP=====================================
# Importing the libraries and the dataset for evaluation
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('ann_temp_gistemp.csv')
year = dataset.iloc[:, 1:2].values
mean = dataset.iloc[:, 2:3].values

#Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin_regr = LinearRegression()
lin_regr.fit(year, mean)
```

```

## Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
poly_regr = PolynomialFeatures(degree = 4)
year_poly = poly_regr.fit_transform(year)
poly_regr.fit(year_poly, mean)
lin_regr_2 = LinearRegression()
lin_regr_2.fit(year_poly, mean)

#Visualising the Linear Regression results
plt.scatter(year, mean, color= 'purple')
plt.plot(year, lin_regr.predict(year), color = 'blue', label = 'The Best Fit Line')
plt.title('year vs mean temperature (linear regression for GISTEMP)')
plt.xlabel('Years')
plt.ylabel('Mean Temperature')
plt.legend()
plt.show()

#Visualising the Polynomial Regression results
plt.scatter(year, mean, color= 'green')
plt.plot(year, lin_regr_2.predict(poly_regr.fit_transform(year)), color = 'blue', label = 'The Best Fit Line')
plt.title('year vs mean temperature (polynomial regression for GISTEMP)')
plt.xlabel('Years')
plt.ylabel('Mean Temperature')
plt.legend()
plt.show()

#Predicting a new result with Linear Regression
temp2016 = lin_regr.predict([[2016]])
temp2017 = lin_regr.predict([[2017]])

#Predicting a new result with Polynomial Regression
temp_poly2016 = lin_regr_2.predict(poly_regr.fit_transform([[2016]]))
temp_poly2017 = lin_regr_2.predict(poly_regr.fit_transform([[2017]]))

```

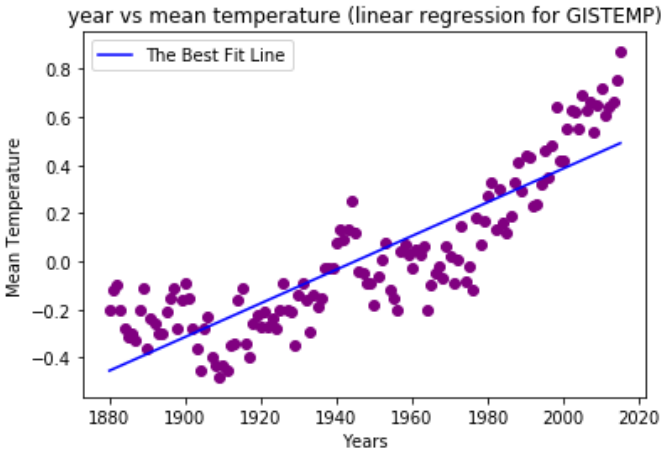
```
print("Temperatures for GISTEMP ")
print("A/c to linear regression, Temperature in 2016 will be: ", temp2016)
print("A/c to polynomial regression, Temperature in 2016 will be: ", temp_poly2016)
print("A/c to linear regression, Temperature in 2017 will be: ", temp2017)
print("A/c to polynomial regression, Temperature in 2017 will be: ", temp_poly2017)
```

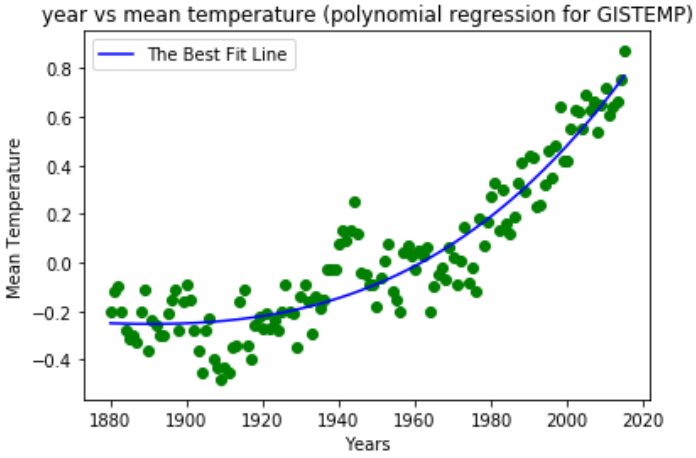
**OUTPUTS:**




```
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]]
```



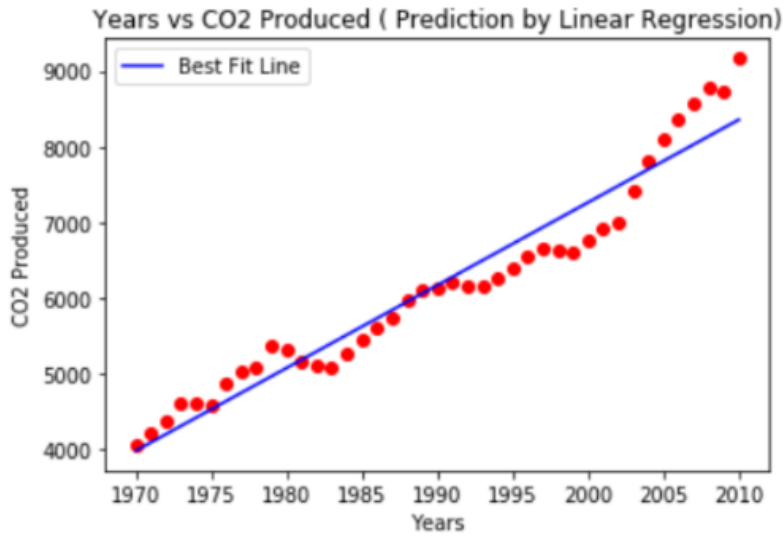



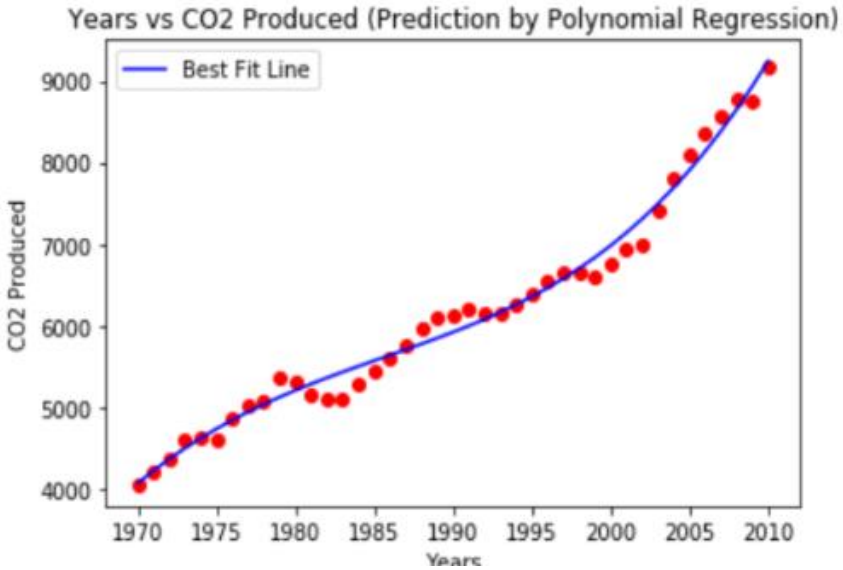

```
Temperatures for GISTEMP
A/c to linear regression, Temperature in 2016 will be:  [[0.49777778]]
A/c to polynomial regression, Temperature in 2016 will be:  [[0.78885745]]
A/c to linear regression, Temperature in 2017 will be:  [[0.50477625]]
A/c to polynomial regression, Temperature in 2017 will 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.**

```
#SARAH SALEEM AKHTER
#Using Polynomial Regression to predict the temperature of CO2

#Importing the libraries for the dataset
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('global_co2.csv')
year = dataset.iloc[:, 0:1].values
total = dataset.iloc[:, 1:2].values

#Fitting linear regression to the dataset
from sklearn.linear_model import LinearRegression
lin_regr = LinearRegression()
lin_regr.fit(year, total)

#Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
poly_regr = PolynomialFeatures(degree=5)
year_poly = poly_regr.fit_transform(year)
poly_regr.fit = (year_poly, total)
lin_regr_2 = LinearRegression()
lin_regr_2.fit(year_poly, total)

#Visualising the Linear Regression results
plt.scatter(year, total, color= 'red')
plt.plot(year, lin_regr.predict(year), color = 'blue', label = 'Best Fit Line')
plt.title('Years vs CO2 Produced (Prediction by Linear Regression)')
plt.xlabel('Years')
plt.ylabel('CO2 Produced')
plt.legend()
plt.show()

#Visualising the Polynomial Regression results
plt.scatter(year, total, color= 'red')
plt.plot(year, lin_regr_2.predict(poly_regr.fit_transform(year)), color = 'blue', label = 'Best Fit Line')
plt.title('Years vs CO2 Produced (Prediction by Polynomial Regression)')
plt.xlabel('Years')
plt.ylabel('CO2 Produced')
plt.legend()
plt.show()

#Predicting a new result with linear regression
Prod_CO2_11 = lin_regr.predict([[2011]])
Prod_CO2_12 = lin_regr.predict([[2012]])
Prod_CO2_13 = lin_regr.predict([[2013]])

#Predicting a new result with polynomial regression
Prod_CO2_poly11 = lin_regr_2.predict(poly_regr.fit_transform([[2011]]))
Prod_CO2_poly12 = lin_regr_2.predict(poly_regr.fit_transform([[2012]]))
Prod_CO2_poly13 = lin_regr_2.predict(poly_regr.fit_transform([[2013]]))


print("Prediction for Production of CO2 in 2011: ")
print("A/c to Linear Regression, CO2 produced in 2011 will be: ", Prod_CO2_11)
print("A/c to polynomial Regression, CO2 produced in 2011 will be: ", Prod_CO2_poly11)

print("Prediction for Production of CO2 in 2012: ")
print("A/c to Linear Regression, CO2 produced in 2012 will be: ", Prod_CO2_12)
print("A/c to polynomial Regression, CO2 produced in 2012 will be: ", Prod_CO2_poly12)

print("Prediction for Production of CO2 in 2013: ")
print("A/c to Linear Regression, CO2 produced in 2013 will be: ", Prod_CO2_13)
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]]
```

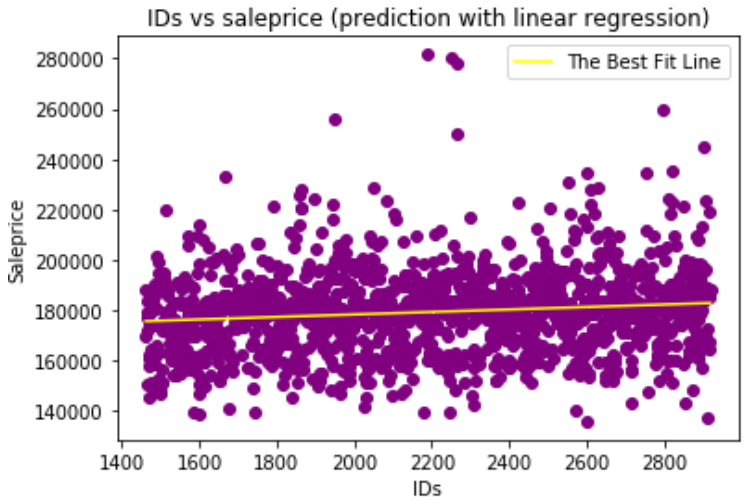
**A/c to Polynomial Regression:**

```
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.**

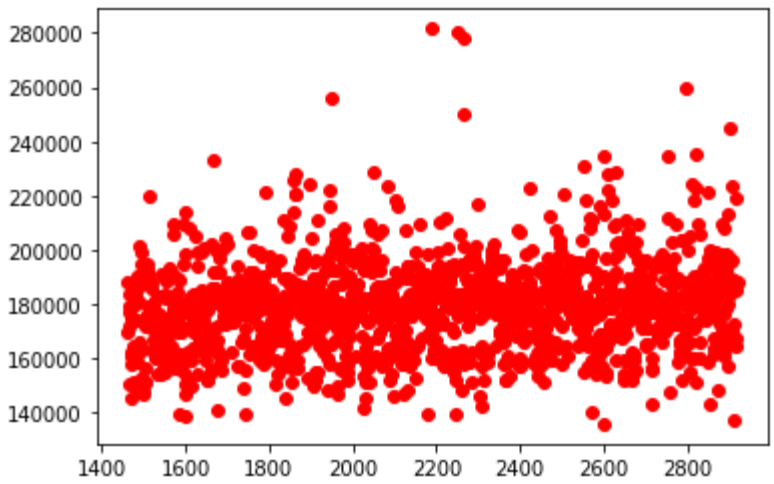
```
#SARAH SALEEM AKHTER
#Using Polynomial Regression to predict the Housing price according to the ID is assigned to every-house.

#Importing the libraries for the dataset
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('housing_price.csv')
ids = dataset.iloc[:, 0:1].values
saleprice = dataset.iloc[:, 1:2].values

#Fitting linear regression to the dataset
from sklearn.linear_model import LinearRegression
lin_regr = LinearRegression()
lin_regr.fit(ids, saleprice)

#Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
poly_regr = PolynomialFeatures(degree=5)
ids_poly = poly_regr.fit_transform(ids)
poly_regr.fit = (ids_poly, saleprice)
lin_regr_2 = LinearRegression()
lin_regr_2.fit(ids_poly, saleprice)

#Visualising the Linear Regression results
plt.scatter(ids, saleprice, color= 'purple')
plt.plot(ids, lin_regr.predict(ids), color = 'yellow', label = 'The Best Fit Line')
plt.title('IDs vs saleprice (prediction with linear regression)')
plt.xlabel('IDs')
plt.ylabel('Saleprice')
plt.legend()
plt.show()

#Visualising the Polynomial Regression results
plt.scatter(ids, saleprice, color= 'red')
plt.plot(ids, lin_regr_2.predict(poly_regr.fit_transform(ids)), color = 'blue', label = 'The Best Fit Line')
plt.title('IDs vs saleprice (prediction with polynomial regression)')
plt.xlabel('IDs')
plt.ylabel('Saleprice')
plt.legend()
plt.show()

#Predicting a new result with Linear Regression
hp2920 = lin_regr.predict([[2920]])
hp2925 = lin_regr.predict([[2925]])
hp2930 = lin_regr.predict([[2930]])


#Predicting a new result with Polynomial Regression
hp_poly2920 = lin_regr_2.predict(poly_regr.fit_transform([[2920]]))
hp_poly2925 = lin_regr_2.predict(poly_regr.fit_transform([[2925]]))
hp_poly2930 = lin_regr_2.predict(poly_regr.fit_transform([[2930]]))

print("A/c to Linear Regression: ")
print("The Housing Price for ID number 2920 will be= ", hp2920)
print("The Housing Price for ID number 2925 will be= ", hp2925)
print("The Housing Price for ID number 2930 will be= ", hp2930)

print("A/c to polynomial Regression: ")
print("The Housing Price for ID number 2920 will be= ", hp_poly2920)
print("The Housing Price for ID number 2925 will be= ", hp_poly2925)
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
```

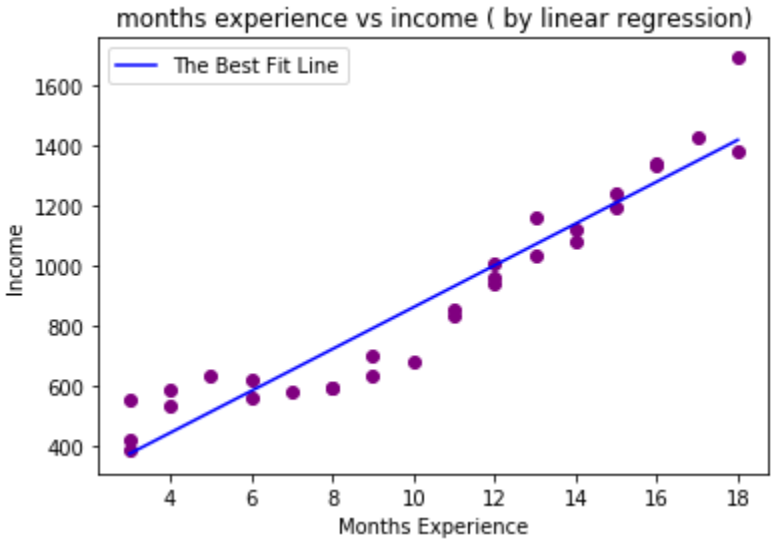


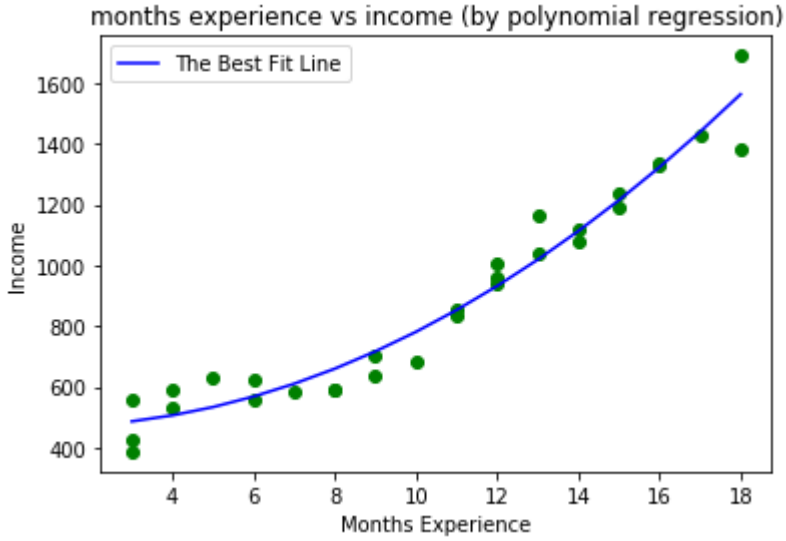
**QUESTION 05**: **Data of monthly experience and income distribution of different employs is given. Perform regression.**

```
#SARAH SALEEM AKHTER
#Using Polynomial Regression to predict the Data of monthly experience and income distribution of different employees

# Importing the libraries and the dataset for evaluation
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset = pd.read_csv('monthlyexp vs incom.csv')
months_exp = dataset.iloc[:, 0:1].values
income = dataset.iloc[:, 1:2].values

#Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin_regr = LinearRegression()
lin_regr.fit(months_exp, income)

## Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
poly_regr = PolynomialFeatures(degree = 2)
months_exp_poly = poly_regr.fit_transform(months_exp)
poly_regr.fit(months_exp_poly, income)
lin_regr_2 = LinearRegression()
lin_regr_2.fit(months_exp_poly, income)

#Visualising the Linear Regression results
plt.scatter(months_exp, income, color= 'purple')
plt.plot(months_exp, lin_regr.predict(months_exp), color = 'blue', label = 'The Best Fit Line')
plt.title('months experience vs income ( by linear regression)')
plt.xlabel('Months Experience')
plt.ylabel('Income')
plt.legend()
plt.show()
```

```

#Visualising the Polynomial Regression results
plt.scatter(months_exp, income, color= 'green')
plt.plot(months_exp, lin_regr_2.predict(poly_regr.fit_transform(months_exp)), color = 'blue', label = 'The Best Fit Line')
plt.title('months experience vs income (by polynomial regression)')
plt.xlabel('Months Experience')
plt.ylabel('Income')
plt.legend()
plt.show()

#Predicting a new result with Linear Regression
exp_19 = lin_regr.predict([[19]])
exp_20 = lin_regr.predict([[20]])
exp_21 = lin_regr.predict([[21]])

#Predicting a new result with Polynomial Regression
exp_poly19 = lin_regr_2.predict(poly_regr.fit_transform([[19]]))
exp_poly20 = lin_regr_2.predict(poly_regr.fit_transform([[20]]))
exp_poly21 = lin_regr_2.predict(poly_regr.fit_transform([[21]]))

print("By Linear Regression: ")
print("Income= ", exp_19)
print("Income= ", exp_20)
print("Income= ", exp_21)

print("By Polynomial Regression: ")
print("Income= ", exp_poly19)
print("Income= ", exp_poly20)
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]]
```