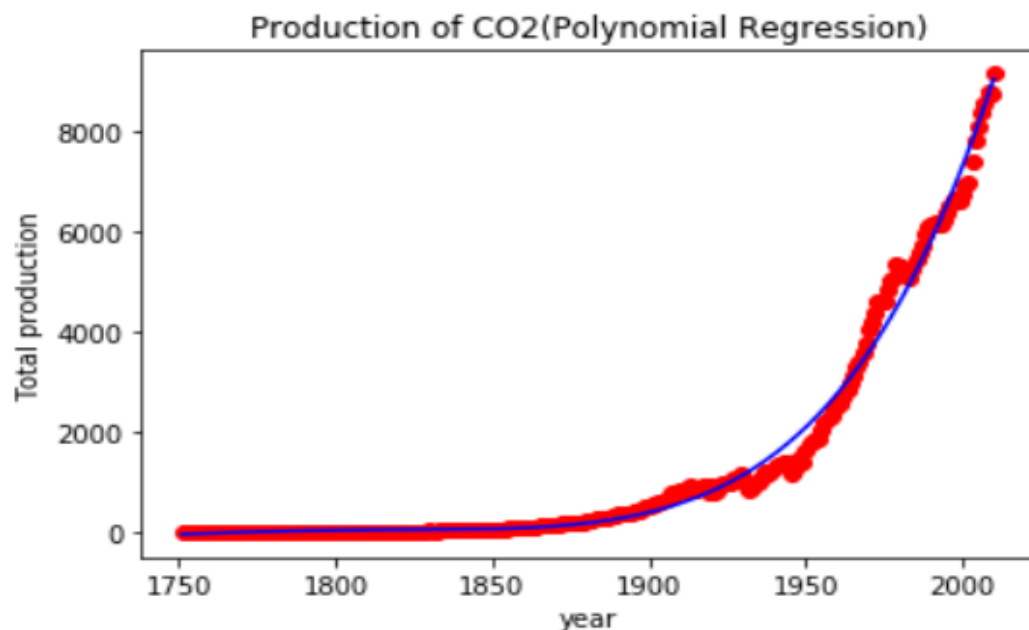


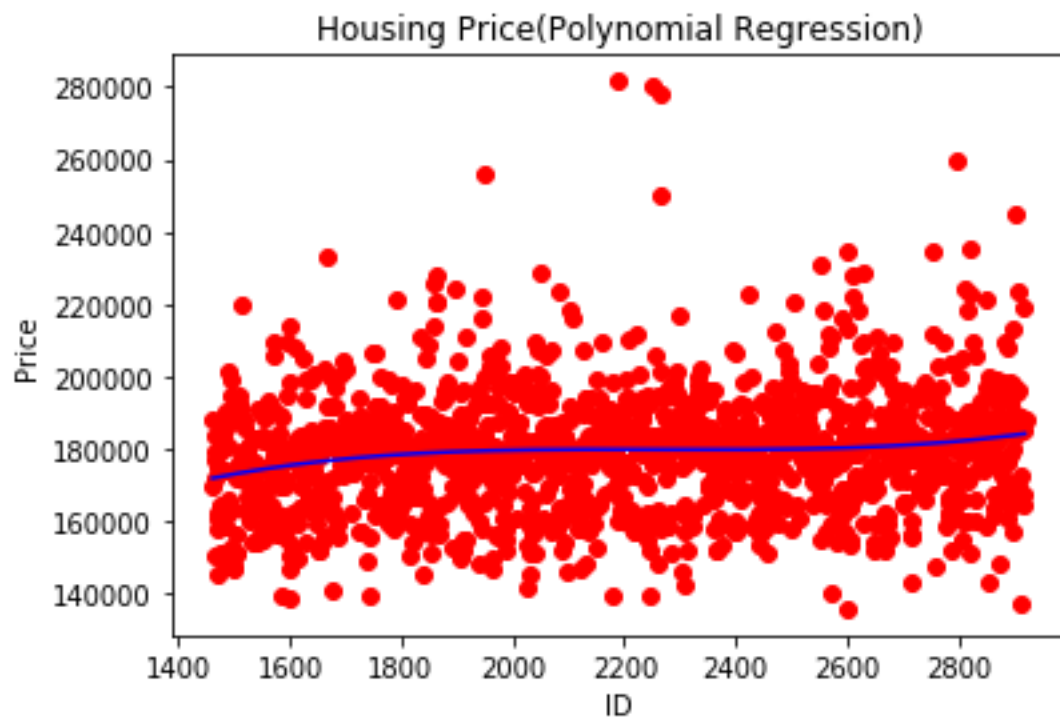
Data of global production of CO₂ of a place is given between 1970s to 2010. Predict the CO₂ production for the years 2011, 2012 and 2013 using the old data set.

```
1 # Importing the libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 # Importing the dataset
6 dataset = pd.read_csv('global_co2.csv')
7 X = dataset.iloc[:, 0:1].values
8 y = dataset.iloc[:, 1].values
9 # Splitting the dataset into the Training set and Test set
10 """from sklearn.cross_validation import train_test_split
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)"""
12 # Fitting Linear Regression to the dataset
13 from sklearn.linear_model import LinearRegression
14 lin_reg = LinearRegression()
15 lin_reg.fit(X, y)
16 # Fitting Polynomial Regression to the dataset
17 from sklearn.preprocessing import PolynomialFeatures
18 poly_reg = PolynomialFeatures(degree = 10)
19 X_poly = poly_reg.fit_transform(X)
20 poly_reg.fit(X_poly, y)
21 lin_reg_2 = LinearRegression()
22 lin_reg_2.fit(X_poly, y)
23 # Visualising the Polynomial Regression results (for higher resolution and smoother curve)
24 X_grid = np.arange(min(X), max(X), 0.1)
25 X_grid = X_grid.reshape((len(X_grid), 1))
26 plt.scatter(X, y, color = 'red')
27 plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
28 plt.title('Production of CO2(Polynomial Regression)')
29 plt.xlabel('year')
30 plt.ylabel('Total production')
31 plt.show()
32 # Predicting a new result with Linear Regression
33 lin_reg.predict([[2013]])
34 # Predicting a new result with Polynomial Regression
35 lin_reg_2.predict(poly_reg.fit_transform([[2013]]))
```



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 # Importing the libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 # Importing the dataset
6 dataset = pd.read_csv('housing price.csv')
7 X = dataset.iloc[:, 0:1].values
8 y = dataset.iloc[:, 1].values
9 # Splitting the dataset into the Training set and Test set
10 """from sklearn.cross_validation import train_test_split
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)"""
12 # Fitting Linear Regression to the dataset
13 from sklearn.linear_model import LinearRegression
14 lin_reg = LinearRegression()
15 lin_reg.fit(X, y)
16 # Fitting Polynomial Regression to the dataset
17 from sklearn.preprocessing import PolynomialFeatures
18 poly_reg = PolynomialFeatures(degree = 3)
19 X_poly = poly_reg.fit_transform(X)
20 poly_reg.fit(X_poly, y)
21 lin_reg_2 = LinearRegression()
22 lin_reg_2.fit(X_poly, y)
23 # Visualising the Polynomial Regression results (for higher resolution and smoother curve)
24 X_grid = np.arange(min(X), max(X), 0.1)
25 X_grid = X_grid.reshape((len(X_grid), 1))
26 plt.scatter(X, y, color = 'red')
27 plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
28 plt.title('Housing Price(Polynomial Regression)')
29 plt.xlabel('ID')
30 plt.ylabel('Price')
31 plt.show()
32 # Predicting a new result with Polynomial Regression
33 lin_reg_2.predict(poly_reg.fit_transform([[2950]]))
```



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

```
1 # Importing the libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 # Importing the dataset
6 dataset = pd.read_csv('monthlyexp vs incom.csv')
7 X = dataset.iloc[:, 0:1].values
8 y = dataset.iloc[:, 1].values
9 # Splitting the dataset into the Training set and Test set
10 """from sklearn.cross_validation import train_test_split
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)"""
12 # Fitting Linear Regression to the dataset
13 from sklearn.linear_model import LinearRegression
14 lin_reg = LinearRegression()
15 lin_reg.fit(X, y)
16 # Fitting Polynomial Regression to the dataset
17 from sklearn.preprocessing import PolynomialFeatures
18 poly_reg = PolynomialFeatures(degree = 10)
19 X_poly = poly_reg.fit_transform(X)
20 poly_reg.fit(X_poly, y)
21 lin_reg_2 = LinearRegression()
22 lin_reg_2.fit(X_poly, y)
23 # Visualising the Polynomial Regression results (for higher resolution and smoother curve)
24 X_grid = np.arange(min(X), max(X), 0.1)
25 X_grid = X_grid.reshape((len(X_grid), 1))
26 plt.scatter(X, y, color = 'red')
27 plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
28 plt.title('Experience Vs Income Graph(Polynomial Regression)')
29 plt.xlabel('Experience')
30 plt.ylabel('Income')
31 plt.show()
32 # Predicting a new result with Linear Regression
33 lin_reg.predict([[10]])
34 # Predicting a new result with Polynomial Regression
35 lin_reg_2.predict(poly_reg.fit_transform([[10]]))
```

