```
In [1]: import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         from sklearn.linear_model import LinearRegression
         from sklearn.preprocessing import PolynomialFeatures
         import numpy as np
```

```
In [2]:  df=pd.read_csv("canada_per_capita_income.csv")
         df.head()
```

Out[2]:

	year	per capita income (US$)
0	1970	3399.299037
1	1971	3768.297935
2	1972	4251.175484
3	1973	4804.463248
4	1974	5576.514583

```
In [3]:  # check linear relationship between Year and per capita income (US$) column
         sns.scatterplot(data=df,x="year",y="per capita income (US$)")
         plt.show()
```
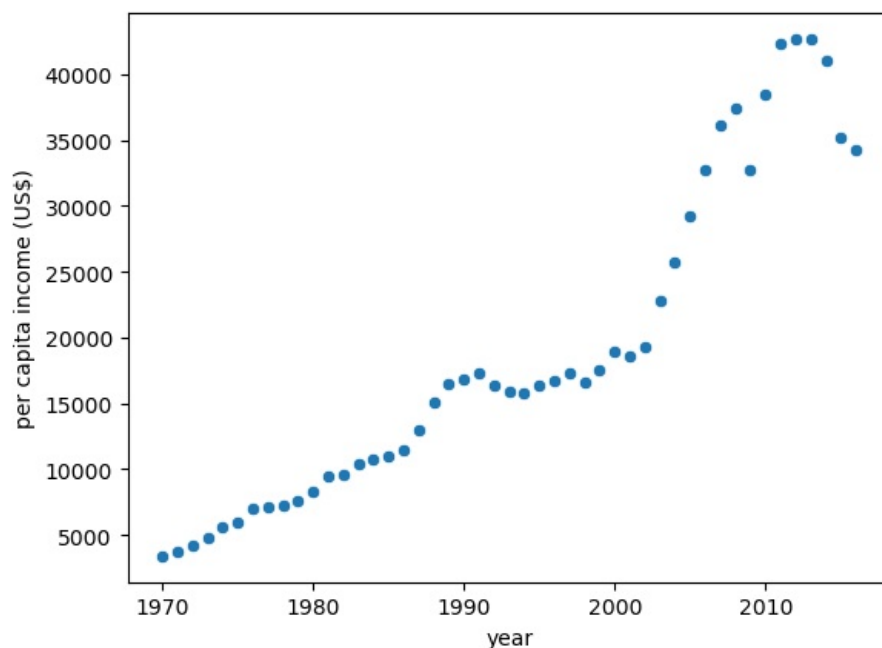


```
In [4]:  # extract dependent and independent variable
         x=df[["year"]]
         y=df[["per capita income (US$)"]]
```

```
In [5]:  plf=PolynomialFeatures(degree=2)
         plf.fit(x)
         x=plf.transform(x)
```

```
In [6]:  # split data into training data and testing data
         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=30)
```
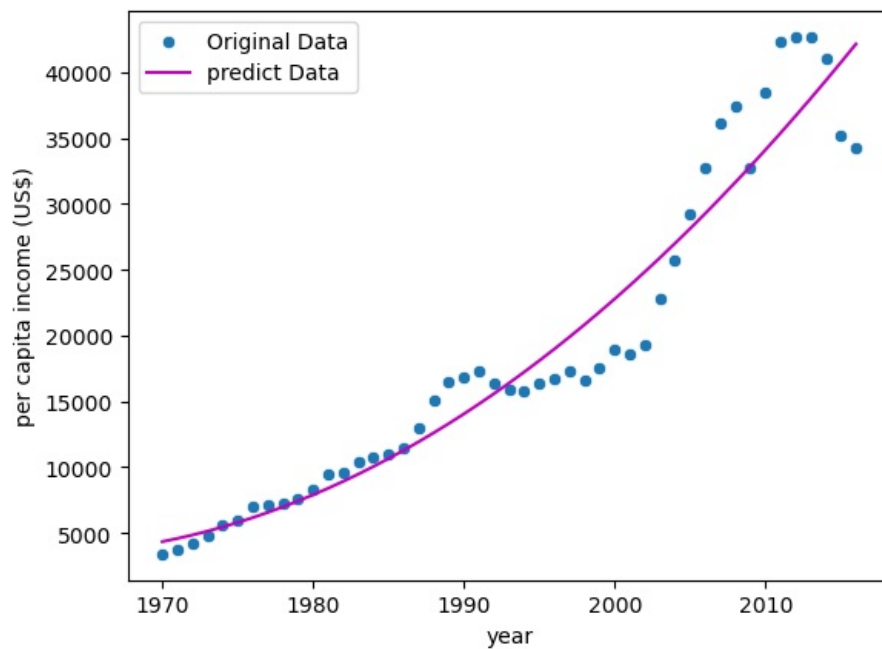
```
In [7]:  # create object of LinearRegression and fit model
         lr=LinearRegression()
         lr.fit(x_train,y_train)
```

Out[7]: ▾ LinearRegression

         LinearRegression()

```
In [8]:  y_pred=lr.predict(x)
```

```
In [9]:  sns.scatterplot(data=df,x="year",y="per capita income (US$)",label="Original Data")
         plt.plot(df["year"],y_pred,c="m",label="predict Data")
         plt.legend()
         plt.show()
```

```
In [10]: t=plf.transform([[1970]])
```

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature name
s, but PolynomialFeatures was fitted with feature names
  warnings.warn(

```
In [11]: lr.predict(t)
```

```
Out[11]: array([[4349.12297122]])
```

```
In [12]: lr.score(x_test,y_test)
```

```
Out[12]: 0.8774412967514764
```

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js