Canada_per_capita-2030

November 24, 2018

Machine Learning With Python: Linear Regression With One Variable
Problem Statement: Predict canada's per capita income in year 2023 using canada_gdp.csv

In [82]:
```
import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt
```

In [83]:
```
df = pd.read_csv('canada_gdp.csv')
df
```

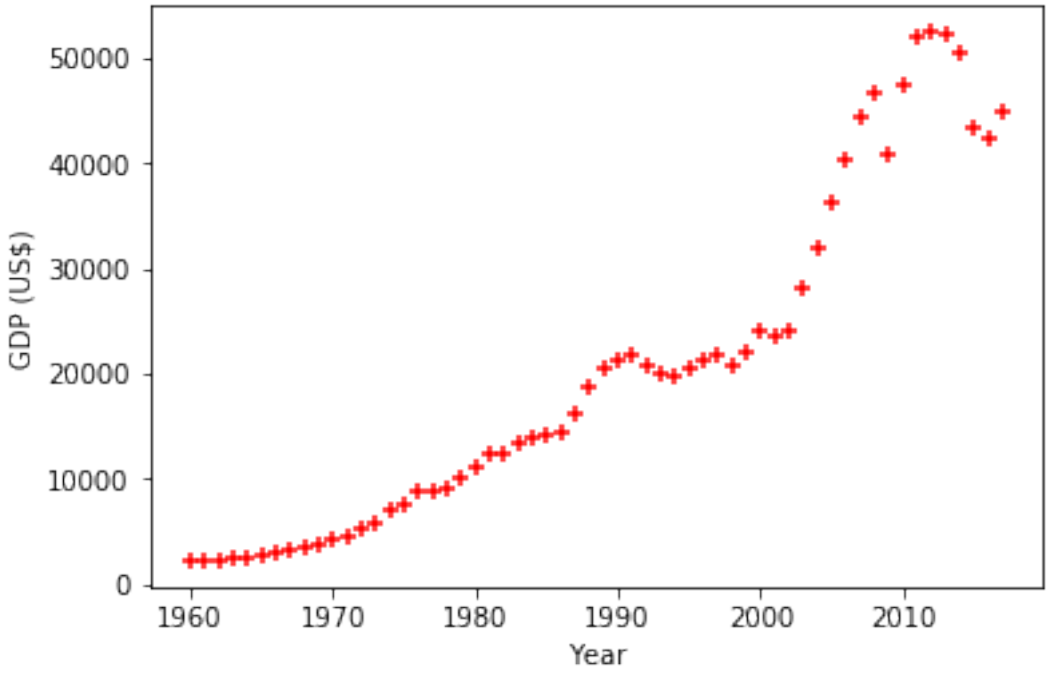
Out[83]:

	year	gdp
0	1960	2294.568814
1	1961	2231.293824
2	1962	2255.230044
3	1963	2354.839122
4	1964	2529.518179
5	1965	2739.585849
6	1966	3010.705908
7	1967	3173.076194
8	1968	3411.060154
9	1969	3703.990405
10	1970	4121.932809
11	1971	4586.255848
12	1972	5141.616725
13	1973	5870.600564
14	1974	7043.474351
15	1975	7489.940531
16	1976	8783.721592
17	1977	8892.761680
18	1978	9096.058722
19	1979	10012.443967
20	1980	11135.437985
21	1981	12297.785688
22	1982	12439.747841
23	1983	13377.895655
24	1984	13826.649992
25	1985	14060.461778

```
26  1986  14403.828702
27  1987  16245.451679
28  1988  18864.262918
29  1989  20638.290049
30  1990  21371.291098
31  1991  21664.598644
32  1992  20771.250353
33  1993  20017.429848
34  1994  19859.203978
35  1995  20577.489386
36  1996  21183.220083
37  1997  21770.134081
38  1998  20887.839467
39  1999  22167.225850
40  2000  24124.169175
41  2001  23691.594719
42  2002  24167.804306
43  2003  28172.148831
44  2004  31979.871951
45  2005  36189.588384
46  2006  40386.699484
47  2007  44544.526800
48  2008  46596.335991
49  2009  40773.454364
50  2010  47447.476024
51  2011  52082.210760
52  2012  52496.694870
53  2013  52418.315062
54  2014  50633.208822
55  2015  43525.370187
56  2016  42348.945461
57  2017  45032.119908
```

In [84]:
```
%matplotlib inline
plt.xlabel('Year')
plt.ylabel('GDP (US$)')
plt.scatter(df.year,df.gdp,color='red',marker='+')
```

Out[84]:
```
<matplotlib.collections.PathCollection at 0x7ff949d72d68>
```




```
In [85]: year = df[['year']]
         year
```

Out[85]:

	year
0	1960
1	1961
2	1962
3	1963
4	1964
5	1965
6	1966
7	1967
8	1968
9	1969
10	1970
11	1971
12	1972
13	1973
14	1974
15	1975
16	1976
17	1977
18	1978
19	1979
20	1980

```
21  1981
22  1982
23  1983
24  1984
25  1985
26  1986
27  1987
28  1988
29  1989
30  1990
31  1991
32  1992
33  1993
34  1994
35  1995
36  1996
37  1997
38  1998
39  1999
40  2000
41  2001
42  2002
43  2003
44  2004
45  2005
46  2006
47  2007
48  2008
49  2009
50  2010
51  2011
52  2012
53  2013
54  2014
55  2015
56  2016
57  2017
```

In [86]:
```
gdp = df.gdp
gdp
```

Out[86]:
```
0       2294.568814
1       2231.293824
2       2255.230044
3       2354.839122
4       2529.518179
5       2739.585849
6       3010.705908
```

7	3173.076194
8	3411.060154
9	3703.990405
10	4121.932809
11	4586.255848
12	5141.616725
13	5870.600564
14	7043.474351
15	7489.940531
16	8783.721592
17	8892.761680
18	9096.058722
19	10012.443967
20	11135.437985
21	12297.785688
22	12439.747841
23	13377.895655
24	13826.649992
25	14060.461778
26	14403.828702
27	16245.451679
28	18864.262918
29	20638.290049
30	21371.291098
31	21664.598644
32	20771.250353
33	20017.429848
34	19859.203978
35	20577.489386
36	21183.220083
37	21770.134081
38	20887.839467
39	22167.225850
40	24124.169175
41	23691.594719
42	24167.804306
43	28172.148831
44	31979.871951
45	36189.588384
46	40386.699484
47	44544.526800
48	46596.335991
49	40773.454364
50	47447.476024
51	52082.210760
52	52496.694870
53	52418.315062
54	50633.208822

```
55    43525.370187
56    42348.945461
57    45032.119908
Name: gdp, dtype: float64
```

```
In [87]: # Create linear regression object
         reg = linear_model.LinearRegression()
         reg.fit(year,gdp)
```

```
Out[87]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                  normalize=False)
```

```
In [88]: reg.predict([[2023]])
```

```
Out[88]: array([51053.28148316])
```

```
In [89]: reg.coef_
```

```
Out[89]: array([888.64448804])
```

```
In [90]: reg.intercept_
```

```
Out[90]: -1746674.5178212621
```

```
In [91]: year_df = pd.read_csv("year.csv")
         year_df.head(3)
```

```
Out[91]:    year
         0  2018
         1  2019
         2  2020
```

```
In [92]: p = reg.predict(year_df)
         p
```

```
Out[92]: array([46610.05904296, 47498.703531  , 48387.34801904, 49275.99250708,
                50164.63699512, 51053.28148316, 51941.9259712 , 52830.57045924,
                54607.85943532, 55496.50392336, 56385.1484114 , 57273.79289944])
```

```
In [93]: year_df['income']=p
         year_df
```

```
Out[93]:     year        income
         0   2018  46610.059043
         1   2019  47498.703531
         2   2020  48387.348019
         3   2021  49275.992507
         4   2022  50164.636995
         5   2023  51053.281483
         6   2024  51941.925971
```

```
7   2025  52830.570459
8   2027  54607.859435
9   2028  55496.503923
10  2029  56385.148411
11  2030  57273.792899
```

```
In [94]: from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(year,gdp,test_size=0.2,random_sta
```

```
In [95]: from sklearn.linear_model import LinearRegression
         regressor = LinearRegression()
         regressor.fit(x_train, y_train)
         lr = LinearRegression().fit(x_train, y_train)
```

```
In [96]: y_pred = regressor.predict([[2023]])
         #y_pred = regressor.predict(x_test)
```

```
In [97]: y_pred
```

```
Out[97]: array([51813.24223421])
```

```
In [98]: year_df = pd.read_csv("year.csv")
         year_df.head(3)
```

```
Out[98]:    year
         0  2018
         1  2019
         2  2020
```

```
In [99]: q = regressor.predict(year_df)
         q
```

```
Out[99]: array([47287.945618  , 48193.00494124, 49098.06426448, 50003.12358773,
                50908.18291097, 51813.24223421, 52718.30155745, 53623.3608807 ,
                55433.47952718, 56338.53885042, 57243.59817366, 58148.65749691])
```

```
In [100]: year_df['income']=q
          year_df
```

```
Out[100]:     year        income
          0   2018  47287.945618
          1   2019  48193.004941
          2   2020  49098.064264
          3   2021  50003.123588
          4   2022  50908.182911
          5   2023  51813.242234
          6   2024  52718.301557
          7   2025  53623.360881
          8   2027  55433.479527
          9   2028  56338.538850
          10  2029  57243.598174
          11  2030  58148.657497
```

```
In [101]: print("Training set score: {:.2f}".format(lr.score(x_train, y_train)))
          print("Test set score: {:.7f}".format(lr.score(x_test, y_test)))
```

```
Training set score: 0.91
Test set score: 0.8580458
```

```
In [102]: from sklearn.preprocessing import PolynomialFeatures
          from sklearn.pipeline import Pipeline


          steps = [
              ('poly', PolynomialFeatures(degree=2)),
              ('model', LinearRegression())
          ]

          pipeline = Pipeline(steps)

          pipeline.fit(x_train, y_train)

          print('Training score: {}'.format(pipeline.score(x_train, y_train)))
          print('Test score: {}'.format(pipeline.score(x_test, y_test)))
```

```
Training score: 0.9461193606805535
Test score: 0.9397330898242072
```

```
In [103]: pipeline.predict([[2023]])
```

```
Out[103]: array([62213.95344429])
```

```
In [104]: # Now Read Years
          year_f = pd.read_csv("year.csv")
          year_f.head(3)
```

```
Out[104]:    year
          0  2018
          1  2019
          2  2020
```

```
In [105]: qr = pipeline.predict(year_f)
          qr
```

```
Out[105]: array([53849.33102671, 55472.99309315, 57121.28636813, 58794.21085164,
                 60491.76654371, 62213.95344429, 63960.77155343, 65732.22087109,
                 69349.01313204, 71194.35607533, 73064.33022716, 74958.9355875 ])
```

```
In [106]: year_f['gdp']=qr
          print('Forecast per capita GDP (US$) : ')
          year_f
```

Forecast per capita GDP (US$) :

Out[106]:

	year	gdp
0	2018	53849.331027
1	2019	55472.993093
2	2020	57121.286368
3	2021	58794.210852
4	2022	60491.766544
5	2023	62213.953444
6	2024	63960.771553
7	2025	65732.220871
8	2027	69349.013132
9	2028	71194.356075
10	2029	73064.330227
11	2030	74958.935588

In [107]:
```
%matplotlib inline
plt.xlabel('Year')
plt.ylabel('GDP (US$)')
plt.scatter(year_f.year,qr,color='red',marker='+')
plt.scatter(df.year,df.gdp,color='g',marker='+')
```

Out[107]: <matplotlib.collections.PathCollection at 0x7ff949ce3400>