Machine Learning Laboratory

(410302)

BE Sem I Honors in AI/ML

Academic Year: 2021-22

Lab Assignment No. 4

Name: Aboli Marathe
Roll Number: 41301
Branch: Department of Computer Engineering

Problem Statement

Write a program to solve house price prediction problem for single variable and multivariable linear Regression using python

Lab Exercise 1

Predict Canada's per capita income in year 2020. you will find canada_per_capita_income.csv file. Using this build a regression model and predict the per capita income of Canadian citizens in year 2020

```
import numpy as np
import pandas as pd
```

```
df = pd.read_csv("canada_per_capita_income.csv")
df.head()
```

```
df.shape
```

```
(47, 2)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47 entries, 0 to 46
Data columns (total 2 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   year                    47 non-null     int64  
 1   per capita income (US$)  47 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 880.0 bytes
```

```
df.describe()
```

	year	per capita income (US$)
count	47.000000	47.000000
mean	1993.000000	18920.137063
std	13.711309	12034.679438
min	1970.000000	3399.299037
25%	1981.500000	9526.914515
50%	1993.000000	16426.725480
75%	2004.500000	27458.601420
max	2016.000000	42676.468370

```
import matplotlib.pyplot as plt

plt.scatter(df["year"], df["per capita income (US$)"], color="r")
plt.xlabel("Year")
plt.ylabel("Per capita Income")
```

```
Text(0, 0.5, 'Per capita Income')
```


```
x = df[["year"]].values
y = df["per capita income (US$)"].values
```

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(x, y)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

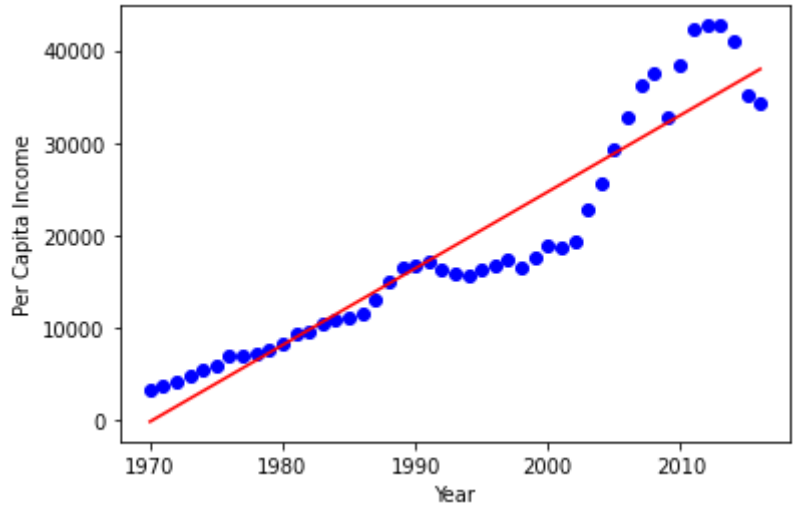
```
m = lr.coef_
c = lr.intercept_

y = m*x + c
y
```

```
array([[ -134.55966672],
       [  693.9054085 ],
       [ 1522.37048373],
       [ 2350.83555895],
       [ 3179.30063417],
       [ 4007.7657094 ],
       [ 4836.23078462],
       [ 5664.69585984],
       [ 6493.16093506],
       [ 7321.62601029],
       [ 8150.09108551],
       [ 8978.55616073],
       [ 9807.02123595],
       [10635.48631118],
       [11463.9513864 ],
       [12292.41646162],
       [13120.88153685],
       [13949.34661207],
       [14777.81168729],
       [15606.27676251],
       [16434.74183774],
       [17263.20691296],
       [18091.67198818],
       [18920.1370634 ],
       [19748.60213863],
       [20577.06721385],
       [21405.53228907],
       [22233.9973643 ],
       [23062.46243952],
       [23890.92751474],
       [24719.39258996],
       [25547.85766519],
       [26376.32274041],
       [27204.78781563],
```

```
       [28033.25289085],
       [28861.71796608],
       [29690.1830413 ],
       [30518.64811652],
       [31347.11319175],
       [32175.57826697],
       [33004.04334219],
       [33832.50841741],
       [34660.97349264],
       [35489.43856786],
       [36317.90364308],
       [37146.3687183 ],
       [37974.83379353]])
```

```
x = df[["year"]]
y = df["per capita income (US$)"]
plt.scatter(x, y, color="b")
plt.plot(x, lr.predict(x), color="r")
plt.xlabel("Year")
plt.ylabel("Per Capita Income")
```

```
Text(0, 0.5, 'Per Capita Income')
```

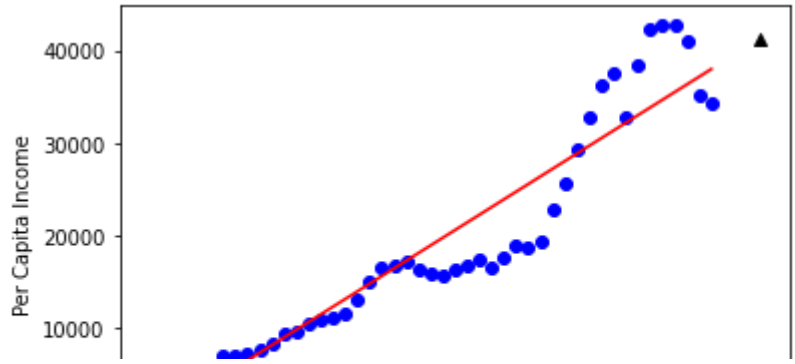



```
lr.predict([[2020]])
```

```
array([41288.69409442])
```

```
y_prediction = lr.predict(x)

plt.scatter(x, y, color="blue")
plt.scatter([[2020]], lr.predict([[2020]]), color="black", marker="^")
plt.plot(x, y_prediction, color="r")
plt.xlabel("Year")
plt.ylabel("Per Capita Income")
```

```
Text(0, 0.5, 'Per Capita Income')
```



Conclusion

The predicted price is 41288.69409442 $.

Lab Exercise 2

You will find hiring.csv. This file contains hiring statics for a firm such as experience of candidate, his written test score and personal interview score. Based on these 3 factors, HR will decide the salary. Given this data, you need to build a machine learning model for HR department that can help them decide salaries for future candidates.

Using this predict salaries for following candidates,

1) 02 yrs. experience, 9 test score, 6 interview score

2) 12 yrs. experience, 10 test score, 10 interview score

```
import pandas as pd
import numpy as np
from sklearn import linear_model
from word2number import w2n
import math


df = pd.read_csv('hiring.csv')
df
```

	experience	test_score(out of 10)	interview_score(out of 10)	salary($)
0	NaN	8.0	9	50000

```
df['experience'] = df['experience'].fillna('zero')
df
```

	experience	test_score(out of 10)	interview_score(out of 10)	salary($)
0	zero	8.0	9	50000
1	zero	8.0	6	45000
2	five	6.0	7	60000
3	two	10.0	10	65000
4	seven	9.0	6	70000
5	three	7.0	10	62000
6	ten	NaN	7	72000
7	eleven	7.0	8	80000

```
df['experience'] = df['experience'].apply(w2n.word_to_num)
df
```

	experience	test_score(out of 10)	interview_score(out of 10)	salary($)
0	0	8.0	9	50000
1	0	8.0	6	45000
2	5	6.0	7	60000
3	2	10.0	10	65000
4	7	9.0	6	70000
5	3	7.0	10	62000
6	10	NaN	7	72000
7	11	7.0	8	80000

```
mean_test_score = math.floor(df['test_score(out of 10)'].mean())
mean_test_score
```

```
7
```

```
df['test_score(out of 10)'] = df['test_score(out of 10)'].fillna(mean_test_score)
df
```

	experience	test_score(out of 10)	interview_score(out of 10)	salary($)
0	0	8.0	9	50000
1	0	8.0	6	45000
2	5	6.0	7	60000
3	2	10.0	10	65000
4	7	9.0	6	70000
5	3	7.0	10	62000

```
# fitting the model
reg = linear_model.LinearRegression()
reg.fit(df[['experience', 'test_score(out of 10)', 'interview_score(out of 10)']], df['sal
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

```
# m value in y = mx + c
reg.coef_
```

```
array([2922.26901502, 2221.30909959, 2147.48256637])
```

```
# c value in y = mx + c
reg.intercept_
```

```
14992.651446693118
```

```
# prediction for question 1
reg.predict([[2, 9, 6]])
```

```
array([53713.86677124])
```

```
# regression-line equation solution for question 1
2922.26901502 * 2 + 2221.30909959 * 9 + 2147.48256637 * 6 + 14992.651446693118
```

```
53713.86677126312
```

```
# prediction for question 2
reg.predict([[12, 10, 10]])
```

```
array([93747.79628651])
```

```
# regression-line equation solution for question 2
2922.26901502 * 12 + 2221.30909959 * 10 + 2147.48256637 * 10 + 14992.651446693118
```

```
93747.79628653312
```

Conclusion

We observe the following predictions:

1. 02 yrs. experience, 9 test score, 6 interview score = 53713
2. 12 yrs. experience, 10 test score, 10 interview score = 93747