

MultipleRegClasswork

November 9, 2022

WTF23 DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

GROUP C SUBGROUP 1

CLASSWORK ON MULTIPLE REGRESSION

QUESTION

In exercise folder (same level as this notebook on github) there is hiring.csv. This file contains hiring statistics 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,

- * 2 yr experience, 9 test score, 6 interview score
- * 12 yr experience, 10 test score, 10 interview score

Import libraries

```
[1]: import pandas as pd
import numpy as np
from sklearn import linear_model
from word2number import w2n
import math
import seaborn as sns
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline
```

Read data and perform EDA and Data cleaning

```
[2]: hire = pd.read_csv('hiring.csv')
hire
```

```
[2]:  experience  test_score(out of 10)  interview_score(out of 10)  salary($)
0         NaN                    8.0                        9      50000
1         NaN                    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
---	--------	-----	---	-------

```
[3]: hire.info()
hire.describe()
hire.isna().sum()
hire.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 4 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   experience                            6 non-null      object
1   test_score(out of 10)                 7 non-null      float64
2   interview_score(out of 10)            8 non-null      int64
3   salary($)                             8 non-null      int64
dtypes: float64(1), int64(2), object(1)
memory usage: 384.0+ bytes
```

```
[3]: Index(['experience', 'test_score(out of 10)', 'interview_score(out of 10)',
          'salary($)'],
          dtype='object')
```

Fill up the NaNs

```
[4]: hire.experience = hire.experience.fillna('zero')
hire
```

```
[4]:   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
```

Use the word-to number modules to convert the worded digit to numbers

```
[5]: hire.experience = hire.experience.apply(w2n.word_to_num)
hire
```

```
[5]:   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

Compute the average test score

```
[6]: test_scoreavg = math.floor(hire['test_score(out of 10)'].mean())
test_scoreavg
```

```
[6]: 7
```

fill up the null values with the average

```
[7]: hire['test_score(out of 10)'] = hire['test_score(out of 10)'].
    ↪ fillna(test_scoreavg)
hire
```

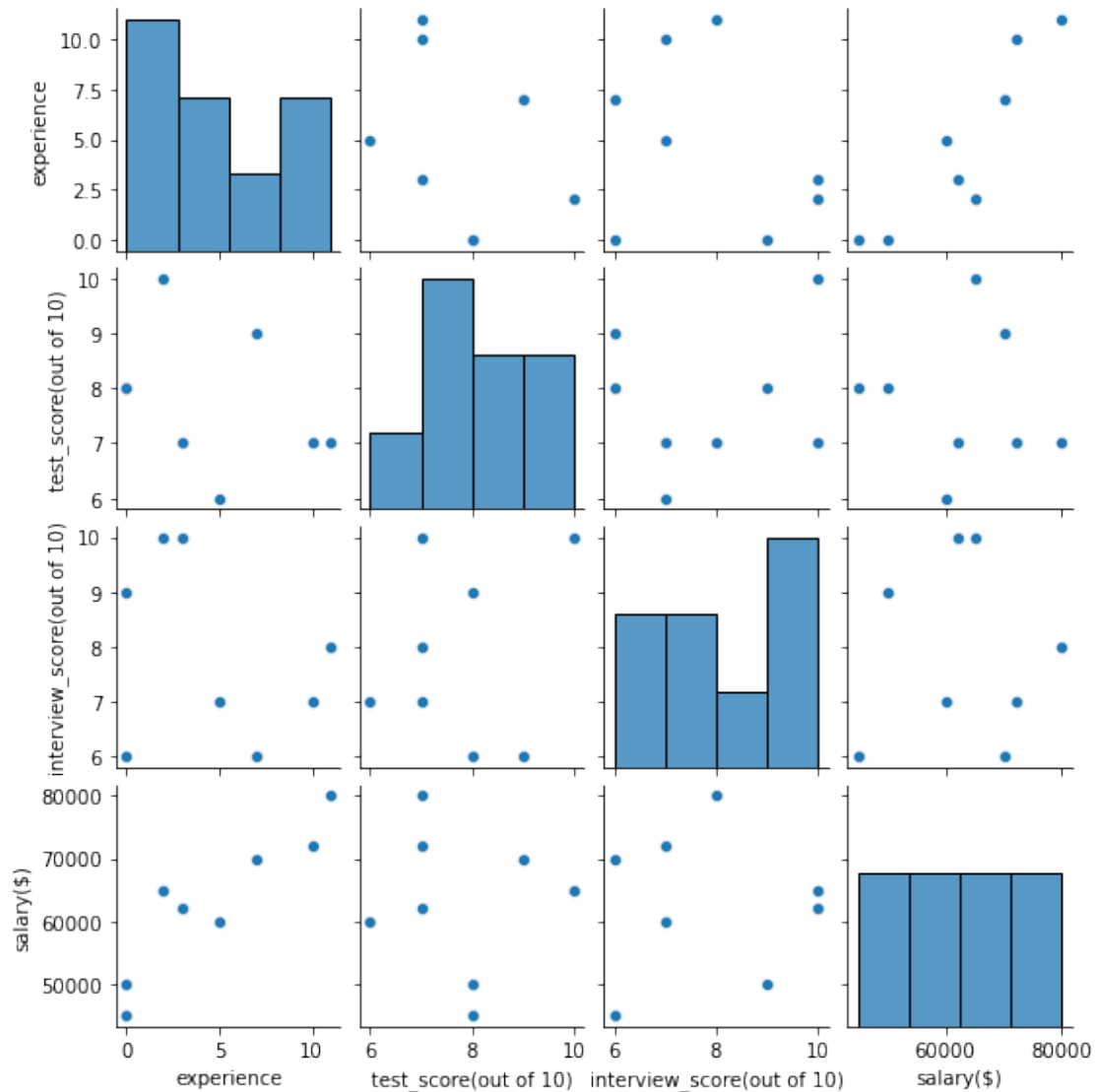
```
[7]:
```

	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	7.0	7	72000
7	11	7.0	8	80000

Visualize the data

```
[8]: sns.pairplot(data = hire, height = 2)
```

```
[8]: <seaborn.axisgrid.PairGrid at 0x24cc17b6e50>
```



COMPUTE THE REGRESSION

```
[9]: reg = linear_model.LinearRegression()
reg.fit(hire[['experience', 'test_score(out of 10)', 'interview_score(out of 10)']], hire['salary($)'])
```

```
[9]: LinearRegression()
```

finding the regression coefficient and intercept

```
[16]: print ("Regression coefficient are:")
print(reg.coef_)
print ("Regression intercept is:")
print(reg.intercept_)
```

Regression coefficient are:
[2922.26901502 2221.30909959 2147.48256637]
Regression intercept is:
14992.65144669314

Salary prediction for candidates with 2 yr experience, 9 test score, 6 interview score
using our prediction model

```
[12]: predicted1 = reg.predict([[2, 9, 6]])  
      predicted1
```

C:\Users\HP\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

```
[12]: array([53713.86677124])
```

To validate our predictions using our regression coefficient

Results shows our predictions to be correct

```
[13]: 2922.26901502*2 + 2221.30909959*9 + 2147.48256637*6 + 14992.651446693118
```

```
[13]: 53713.86677126312
```

Salary prediction for candidates with 12 yr experience, 10 test score, 10 interview score
using our prediction model

```
[14]: predicted2= reg.predict([[12, 10, 10]])  
      predicted2
```

C:\Users\HP\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

```
[14]: array([93747.79628651])
```

To validate our predictions using our regression coefficient

```
[15]: 2922.26901502*12 + 2221.30909959*10 + 2147.48256637*10 + 14992.651446693118
```

```
[15]: 93747.79628653312
```

CONCLUSION

Our models both the predictive multiple regression model and the normal multiple regression model shows same result.

Candidates with 2 yr experience, 9 test score, 6 interview score will be paid 53713.8668

Candidates with 12 yr experience, 10 test score, 10 interview score 93747.7963

CONTRIBUTORS

Margaret Oluwadare

Loveth Osuagwu

Oluchi Okoro (Oluchi Oluchi)

Monsurat Onabajo

Mariam Anishere

Olubusayo Solola

Olayemi Ibiloye

Maryann Amaefula

Olayemi Oloyede

Oluwadunsin Olajide

[]: