

Assignment on Regression using KNN.

Build an application where it can predict Salary based on year of experience using KNN (use salary dataset from Kaggle). Display MSE.

```
In [2]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

Loading Dataset

```
In [3]: data_frame=pd.read_csv("C:/Users/3yearb1/Desktop/ML_datasets/Salary_dataset.csv")
```

In [4]: data_frame

Out[4]:

	Unnamed: 0	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0
5	5	3.0	56643.0
6	6	3.1	60151.0
7	7	3.3	54446.0
8	8	3.3	64446.0
9	9	3.8	57190.0
10	10	4.0	63219.0
11	11	4.1	55795.0
12	12	4.1	56958.0
13	13	4.2	57082.0
14	14	4.6	61112.0
15	15	5.0	67939.0
16	16	5.2	66030.0
17	17	5.4	83089.0
18	18	6.0	81364.0
19	19	6.1	93941.0
20	20	6.9	91739.0
21	21	7.2	98274.0
22	22	8.0	101303.0
23	23	8.3	113813.0
24	24	8.8	109432.0
25	25	9.1	105583.0
26	26	9.6	116970.0
27	27	9.7	112636.0
28	28	10.4	122392.0
29	29	10.6	121873.0

```
In [5]: data_frame.head()
```

```
Out[5]:
```

	Unnamed: 0	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0

```
In [6]: data_frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
Unnamed: 0      30 non-null int64
YearsExperience  30 non-null float64
Salary          30 non-null float64
dtypes: float64(2), int64(1)
memory usage: 800.0 bytes
```

```
In [7]: df = pd.DataFrame(data_frame)
```

```
In [8]: df = df.drop('Unnamed: 0', axis=1)
```

```
In [9]: display(df)
```

	YearsExperience	Salary
0	1.2	39344.0
1	1.4	46206.0
2	1.6	37732.0
3	2.1	43526.0
4	2.3	39892.0
5	3.0	56643.0
6	3.1	60151.0
7	3.3	54446.0
8	3.3	64446.0
9	3.8	57190.0
10	4.0	63219.0
11	4.1	55795.0
12	4.1	56958.0
13	4.2	57082.0
14	4.6	61112.0
15	5.0	67939.0
16	5.2	66030.0
17	5.4	83089.0
18	6.0	81364.0
19	6.1	93941.0
20	6.9	91739.0
21	7.2	98274.0
22	8.0	101303.0
23	8.3	113813.0
24	8.8	109432.0
25	9.1	105583.0
26	9.6	116970.0
27	9.7	112636.0
28	10.4	122392.0
29	10.6	121873.0

```
In [10]: df.describe()
```

```
Out[10]:
```

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.413333	76004.000000
std	2.837888	27414.429785
min	1.200000	37732.000000
25%	3.300000	56721.750000
50%	4.800000	65238.000000
75%	7.800000	100545.750000
max	10.600000	122392.000000

```
In [11]: import sklearn
```

TRAIN TEST SPLIT

```
In [12]: from sklearn.model_selection import train_test_split  
train , test = train_test_split(df, test_size = 0.3)
```

```
In [13]: x_train = train.drop('Salary', axis=1)  
y_train = train['Salary']
```

Visualizing the Test set results

```
In [17]: viz_test = plt
viz_test.scatter(x_train, y_train, color='red')
#viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Salary VS Experience (data set)')
viz_test.xlabel('Year of Experience')
viz_test.ylabel('Salary')
viz_test.show()
```



```
In [18]: x_test = test.drop('Salary', axis=1)
y_test = test['Salary']
```

```
In [19]: x_test
```

Out[19]:

YearsExperience	
17	5.4
27	9.7
28	10.4
9	3.8
21	7.2
5	3.0
22	8.0
25	9.1
7	3.3

```
In [20]: y_test
```

```
Out[20]: 17      83089.0
          27     112636.0
          28     122392.0
           9      57190.0
          21     98274.0
           5      56643.0
          22     101303.0
          25     105583.0
           7      54446.0
          Name: Salary, dtype: float64
```

```
In [21]: from sklearn import neighbors
          from sklearn.metrics import mean_squared_error
          from math import sqrt
          import matplotlib.pyplot as plt
          %matplotlib inline
```

```
In [22]: model = neighbors.KNeighborsRegressor(n_neighbors = 3)
          model.fit(x_train, y_train)
          pred=model.predict(x_test)
          error = sqrt(mean_squared_error(y_test,pred))
```

```
In [23]: pred
```

```
Out[23]: array([ 71777.66666667, 116091.66666667, 116091.66666667,  58657.33333333,
                  99831.        ,  54829.66666667, 104994.66666667, 113405.        ,
                  62605.33333333])
```

```
In [24]: error
```

```
Out[24]: 6044.597175357981
```

```
In [ ]:
```