

## 10. Assignment on regression using KNN: Build an application where it can predict a salary based on year of experience using KNN (Use Salary dataset from Kaggle).

```
In [1]: # Importing Dependencies
import pandas as pd      # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import numpy as np       # linear algebra
dataset=pd.read_csv('Salary_dataset.csv')
```

```
In [2]: dataset.describe()
```

```
Out[2]:
```

	Unnamed: 0	YearsExperience	Salary
count	30.000000	30.000000	30.000000
mean	14.500000	5.413333	76004.000000
std	8.803408	2.837888	27414.429785
min	0.000000	1.200000	37732.000000
25%	7.250000	3.300000	56721.750000
50%	14.500000	4.800000	65238.000000
75%	21.750000	7.800000	100545.750000
max	29.000000	10.600000	122392.000000

```
In [3]: # Number of observations and missing values.
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Unnamed: 0      30 non-null    int64
 1   YearsExperience  30 non-null    float64
 2   Salary          30 non-null    float64
dtypes: float64(2), int64(1)
memory usage: 848.0 bytes
```

```
In [4]: # Split data into a training set and a testing set.
from sklearn.model_selection import train_test_split
train,test=train_test_split(dataset,test_size=0.3)
```

```
In [5]: x_train=train['YearsExperience']

y_train=train['Salary']
```

```
In [6]: #Graphical representation of Testing data
viz_test=plt
viz_test.scatter(x_train,y_train,color='red')
viz_test.title('Salary Vs Experience')
viz_test.xlabel('Year of Experience')
viz_test.ylabel('Salary')
viz_test.show()
```



```
In [7]: #Reset the traing and testing datas of the dataset
x_train=train.drop('Salary',axis=1)
x_test=test.drop('Salary',axis=1)
y_test=test['Salary']
```

```
In [8]: x_test
```

```
Out[8]:
```

	Unnamed: 0	YearsExperience
13	13	4.2
27	27	9.7
4	4	2.3
24	24	8.8
0	0	1.2
7	7	3.3
18	18	6.0
21	21	7.2
14	14	4.6

```
In [9]: y_test
```

```
Out[9]:
```

13	57082.0
27	112636.0
4	39892.0
24	109432.0
0	39344.0
7	54446.0
18	81364.0
21	98274.0
14	61112.0

Name: Salary, dtype: float64

```
In [10]: from sklearn import neighbors
from sklearn.metrics import mean_squared_error
from math import sqrt
```

```
In [11]: model=neighbors.KNeighborsRegressor(n_neighbors=3)
model.fit(x_train,y_train)
```

```
Out[11]: KNeighborsRegressor(n_neighbors=3)
```

```
In [12]: #Predict Test Set Results
pred=model.predict(x_test)
print(pred)
```

```
[ 60230.66666667 114981.66666667 45967.        106899.66666667
 42488.          60413.33333333 81020.        102285.
 63642.33333333]
```

Evaluate Model Performance :

```
In [13]: error=sqrt(mean_squared_error(y_test,pred))
print('Error : ',error)

Error :  3754.0709442312077
```