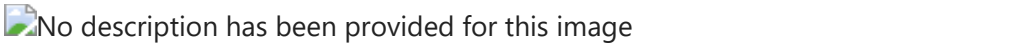
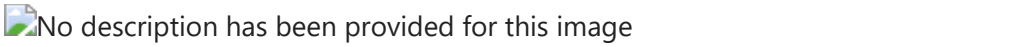
43. K-Nearest Neighbors (Regression)



In [ ]:



In [6]:
```
import pandas as pd
```

In [7]:
```
dataset = pd.read_csv(r'Data/salary_data_2.csv')
dataset.head(3)
```

Out[7]:

	Age	Salary	Experience
0	53	274930.6859	21
1	39	217753.6963	19
2	32	166660.9774	19

In [8]:
```
dataset.isnull().sum()
```

Out[8]:
```
Age           0
Salary        0
Experience    0
dtype: int64
```

Step 1: Split the data into dependent and independent variables

In [9]:
```
x = dataset.drop(columns='Salary')
y = dataset['Salary']
```

In [10]:
```
x
```

Out[10]:

	Age	Experience
0	53	21
1	39	19
2	32	19
3	45	29
4	43	18
...	...	...
995	31	32
996	34	1
997	31	23
998	57	8
999	47	13

1000 rows × 2 columns

In [11]:
```
y
```

Out[11]:
```
0      274930.68590
1      217753.69630
2      166660.97740
3      281857.67490
4      221357.62130
           ...
995    246721.16790
996     98140.45687
997    207088.25770
998    231458.17290
999    213710.38920
Name: Salary, Length: 1000, dtype: float64
```

Step 2: Split the data into train and test variables

In [12]:
```
from sklearn.model_selection import train_test_split
```

In [16]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_st
```

Step 3: Apply KNN Regression Model

In [17]:
```
from sklearn.neighbors import KNeighborsRegressor
```

In [22]:
```
knn = KNeighborsRegressor(n_neighbors=5)
knn.fit(x_train, y_train)
```

Out[22]:
```
▾ KNeighborsRegressor
KNeighborsRegressor()
```

Step 4: Check Accuracy of Model

In [23]:
```
knn.score(x_test, y_test)*100
```

Out[23]:
```
96.56477286387577
```

In [ ]:
```

```