**Machine Learning**

**Impl – 1**

1. Randomly generate 120 values of x in the range [0,1]. Let them be x1, x2, · · · , x120
2. **Regression:**

i. Consider the function yi = 1 + xi + x 2 i . Compute the values y1, · · · , y120. Take the first 80 pairs for training. So, the training set is {(x1, y1),(x2, y2), · · · ,(x80, y80)}. Use the kNN based prediction for x81, · · · , x120 and obtain ˆy81, · · · , yˆ120.

ii. Compute the error as

error = X 120 81 (yi − yˆi) 2.

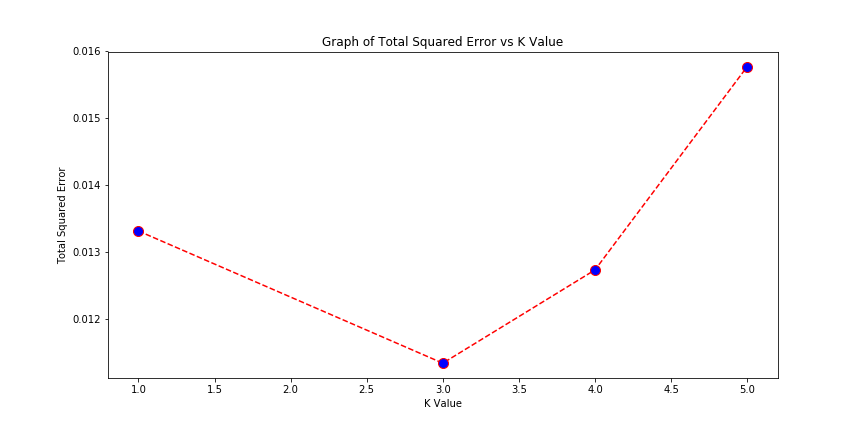
**CODE:**

|  |
| --- |
| Please find the code for Regression as committed file *KNNC\_Regression\_impl\_1.py*   * KNN based regressor is implemented. * Error is calculated between the predicted and expected value of y for different neighbor values of K = 1,3,4,5 |

**RESULT:**

|  |
| --- |
| Below shows the error value for different K=1,3,4,5  [0.013312607265555677, 0.011333462819107638, 0.012729465158012536, 0.01577262840021358] |

**PLOT:**

****

**INFERENCE/ANALYSIS:**

|  |
| --- |
| * The program determines the total squared error value for different values of K nearest neighbors. * The result shows that for K=3, the total squared error is the least. * When K=5, the total squared error is the highest * The program uses a **random number generator** and hence each run produces slightly **different outputs**. |

**RESOURCES USED FOR THE IMPLEMENTATION:**

|  |
| --- |
| * **Environment:**   Anaconda, Jupyter notebook |
| * **Software :**   Python  **Python libraries/modules:** Pandas, Numpy, SkLearn etc |