RITWIK SRIVASTAVA - i630300

MACHINE LEARNING - LAB 2

K-NEAREST NEIGHBOURS

Introduction

This report evaluates enhancements and testing strategies for the kNN (k-Nearest Neighbors) classifier within the context of the provided for Lab-02. The kNN approach is specifically designed for datasets featuring numeric input attributes.

TASK B

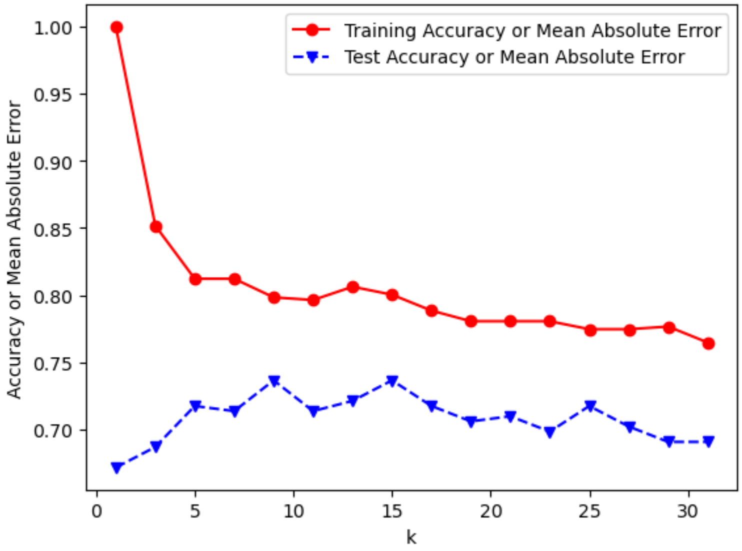
We are asked to add to the kNN class a method named `**normalize**`. This method facilitates the normalization of input attributes within both training **(`X_train`)** and test **(`X_test`)** data. Attribute normalization holds significance since it ensures equal weighting across all attributes during the computation of instance distances. This yields data that is more relative to analyse.

The method `**normalize**` utilizes the `**max**` and `**min**` method from `**pandas.DataFrame**` to achieve attribute normalization.

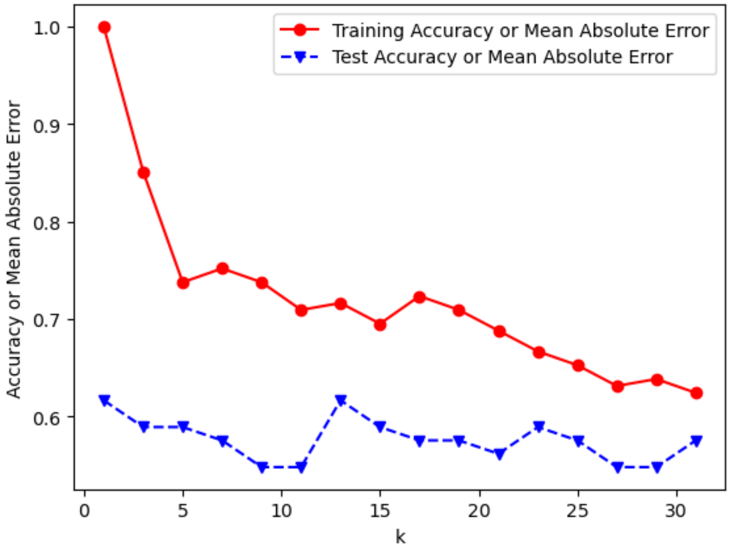
TASK B - TESTING

The kNN classifier is evaluated using two distinct datasets: **diabetes** and **glass** classification datasets. The evaluation includes scenarios where data remains unnormalized and after normalization. From the looks of it, I believe that normalisation does not impact the accuracy rates for the two datasets we have used. The plots have come up identical for before and after. (I had commented out the normalisation function for before normalisation plots).

Here are the plots that were generated by the first testing script before normalization:

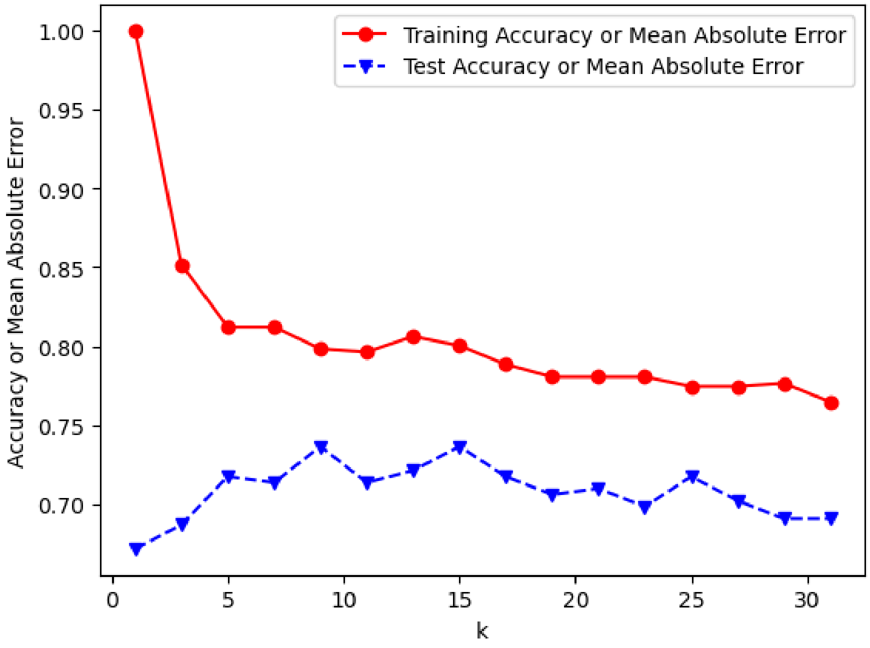


Diabetes Classification DataSet

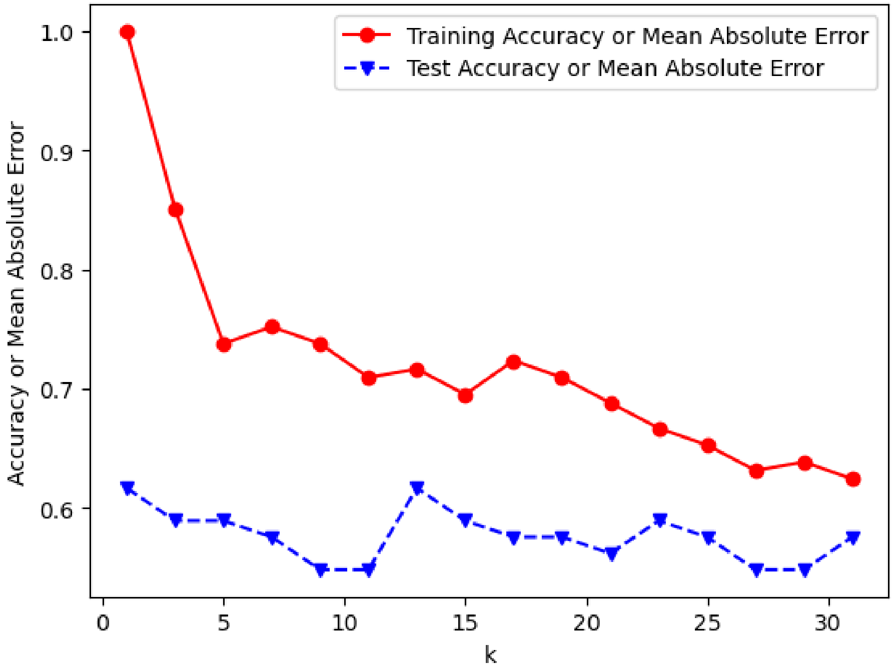


Glass Classification DataSet

Here are the plots that were generated by the first testing script after normalization:



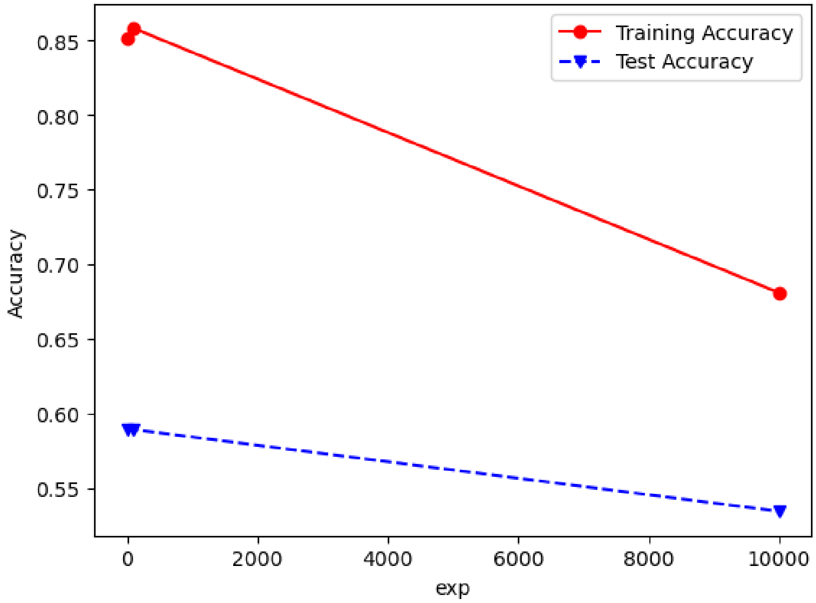
Diabetes Classification DataSet



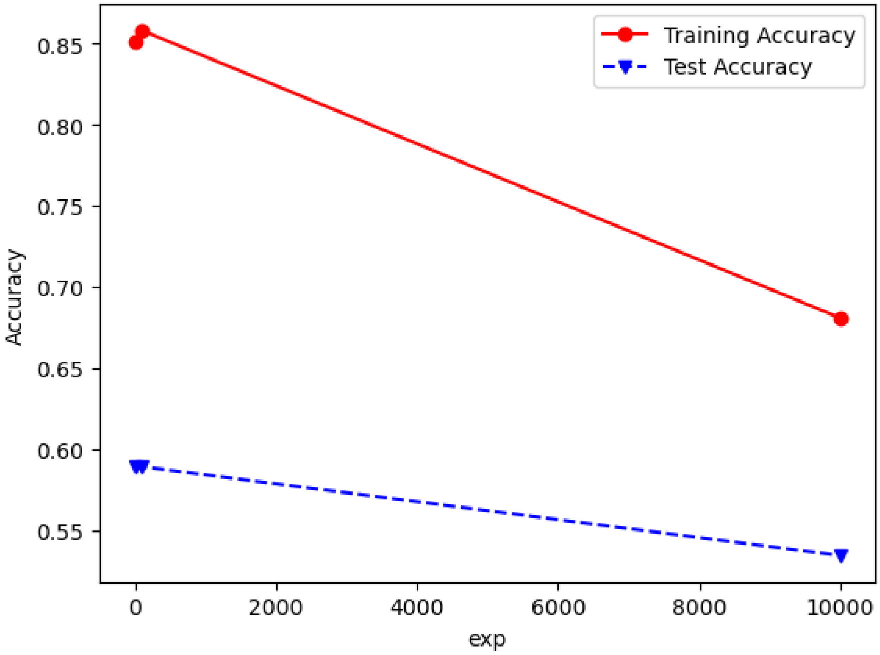
Glass Classification DataSet

Additionally, the kNN classifier is tested on the glass classification dataset with normalized data across various values of the **`'exp'`** parameter of the **`Minkowski distance`**. The analysis aims to identify alterations in training and hold-out accuracy rates due to varying **`'exp'`** values. It is very evidently visible that the accuracy rates do change due to the varying exp values. The accuracy is highest for low values of exp and there is a strong negative correlation.

Here are the plots that were generated by the second testing script:



Diabetes Classification DataSet



Glass Classification DataSet

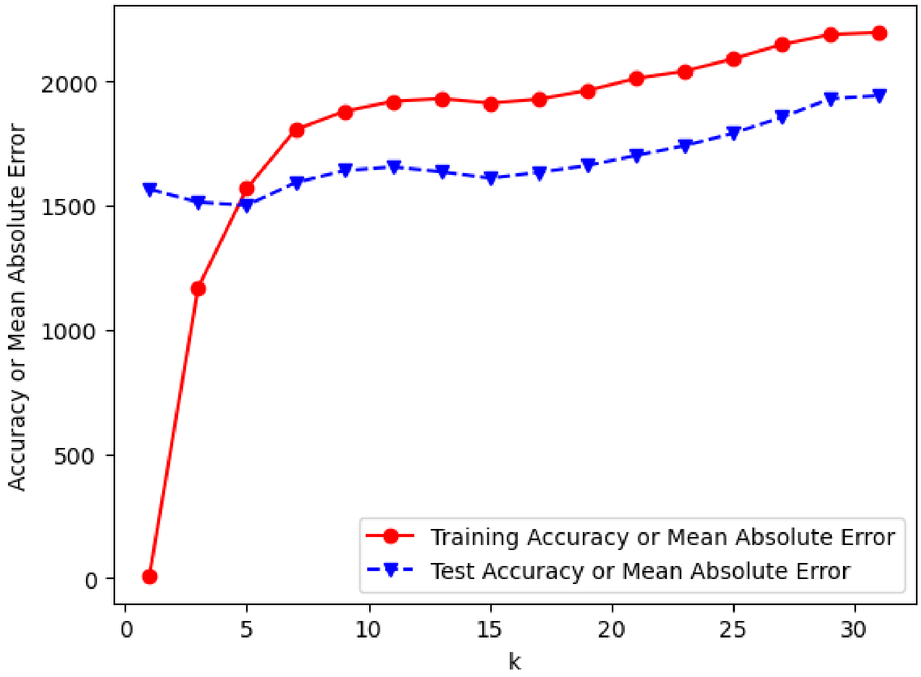
TASK C AND D

Method **getClassProbs**: The method computes posterior class probabilities for all instances in `X_test`. The output of this method contains rows denoting the probability of each class for respective instances in `X_test`. This method is specifically applied to the normalized **glass** classification dataset with the parameter **'exp**.'

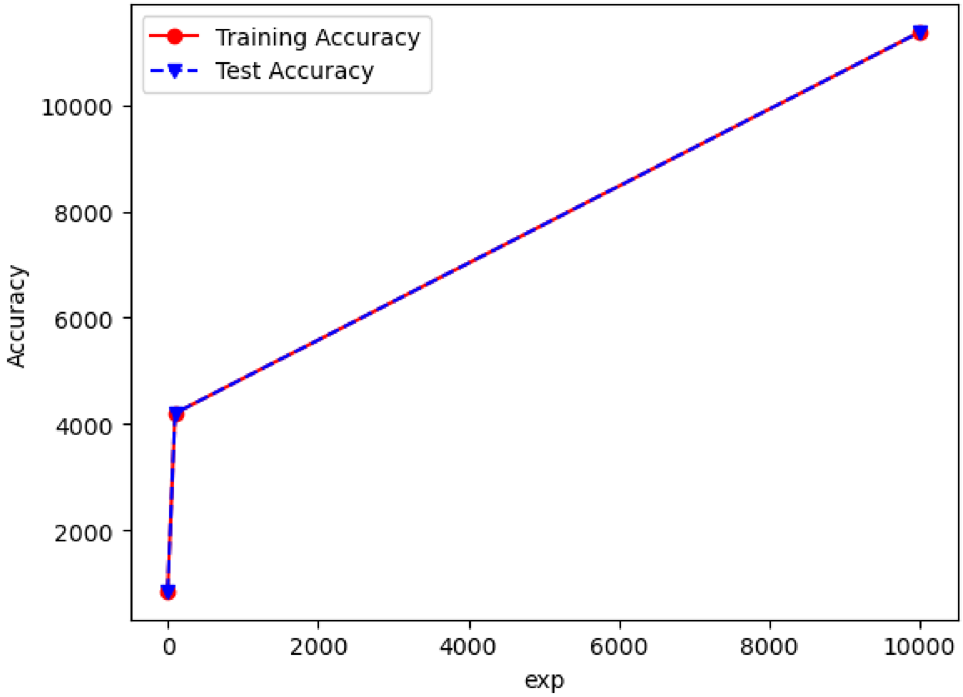
Method **getPrediction**: For regression tasks, this method calculates regression values for all instances in `X_test`. The computed regression values represent the average of 'y' values in `Y_train` for the k-nearest neighbors of each instance in `X_train`. The output of this method contains the computed regression values.

TASK D - TESTING

The `getPrediction` method is tested on the **autoprice** dataset, which is a regression dataset. The testing script, adapted from Task B, utilizes mean absolute error as the primary metric for regression performance evaluation. The `mean_absolute_error` method is used to compute the mean absolute error. I was not sure if both the testing scripts had to be utilised or not, so I did. I also only tested on normalised data because it wasn't specifically mentioned. We were also not given something to analyse or compare to I have just attached the plots below.



Test Script 1 - autopilot



Test Script 2 - autopilot