**CSE-6363-003 Machine Learning Assignment**

**KNN Classification for three different datasets**

**Name:** Harshita Chegondi **Student ID:** 1002115738

**Introduction:**

For classification and regression applications, K-Nearest Neighbors (KNN) is a straightforward and understandable machine learning technique. Being non-parametric and instance-based, it doesn't make any assumptions about the distribution of the underlying data and instead bases predictions on how similar the data points are to one another.

The number of nearest neighbors taken into account while producing a forecast is indicated by the 'K' in KNN. Each neighbor can choose their favorite class in classification tasks, and the class with the most votes wins. In regression tasks, the algorithm calculates an estimation by averaging the values of the 'K' nearest neighbors.

KNN uses a distance metric to determine how close or far apart data points are, often the Euclidean distance but also the Manhattan, Minkowski, or cosine similarity. The effectiveness of the algorithm can be affected by the distance metric selection.

**Data Preprocessing:**

Preprocessing of the data enables the detection and correction of mistakes and irregularities in the dataset, such as missing values, outliers, and noisy data. Data that is more accurate and cleaner as a result. Missing values are a common feature of real-world data. The data can be made complete and suitable for analysis or modelling with the aid of data pretreatment techniques like imputation (replacing missing values with reasonable guesses).

Here, in the three datasets given, import the necessary libraries and pass the path of a dataset file. For each dataset LabelEncoding have been used to convert categorical data to numerical data so that helps to pass the data into the model for prediction. Each unique category within each column is given a separate integer.

**KNN without SKLearn:**

To determine distances and locate the closest neighbors in a Python implementation, we can use data structures like lists, NumPy arrays, and loops. Understanding the inner workings of the algorithm and obtaining insight into its benefits and drawbacks can be necessary for implementing KNN from scratch.

Although it gives users more control over the behavior of the algorithm, employing libraries like scikit-learn is frequently more practical and effective for real-world applications. It's an instance-based learning system that uses data point similarity to anticipate outcomes.

**KNN using SKLearn:**

Scikit-learn, or sklearn, is a robust Python machine learning toolkit that includes the popular machine learning algorithm K-Nearest Neighbors (KNN). You should perform preprocessing on your data, such as feature scaling, addressing missing values, and dividing it into training and test sets, before utilizing KNN. StandardScaler and other scikit-learn preprocessing techniques are at your disposal. Using the proper class KNeighborsClassifier for classification tasks or KNeighborsRegressor for regression tasks—you must build a KNN model. In KNN, selecting the number of neighbors (K) is an important hyperparameter. For your particular situation, you might wish to undertake hyperparameter tweaking, which usually involves methods like grid search or cross-validation to determine the ideal value of K.

**K Fold Cross Validation**

K-fold cross-validation is a method that is frequently used in statistics and machine learning to evaluate the effectiveness of a prediction model. It is especially helpful for figuring out how well a model would generalize to new data. K-fold cross-validation's key advantages are that it gives a more accurate assessment of model performance and guarantees that the complete dataset is used for both training and testing. Although 5 and 10 are typical choices for K, you may also pick various values based on your dataset and requirements.

**Why t-test is used?**

When comparing the means of two groups to see if there is a statistically significant difference between them, the t-test is performed. When you have two separate sets of data and wish to determine whether there is a statistically significant difference between their means, you typically utilize it for hypothesis testing. Here, for KNN, we use t-test to compare and decide which model is efficient.

**T-statistic:** The t-statistic is a measurement of the variance within two groups in relation to the difference between their means.

**P-Value:** The p-value, which assumes that there is no actual difference between the two groups, is the likelihood of observing a t-statistic as extreme as the one calculated.

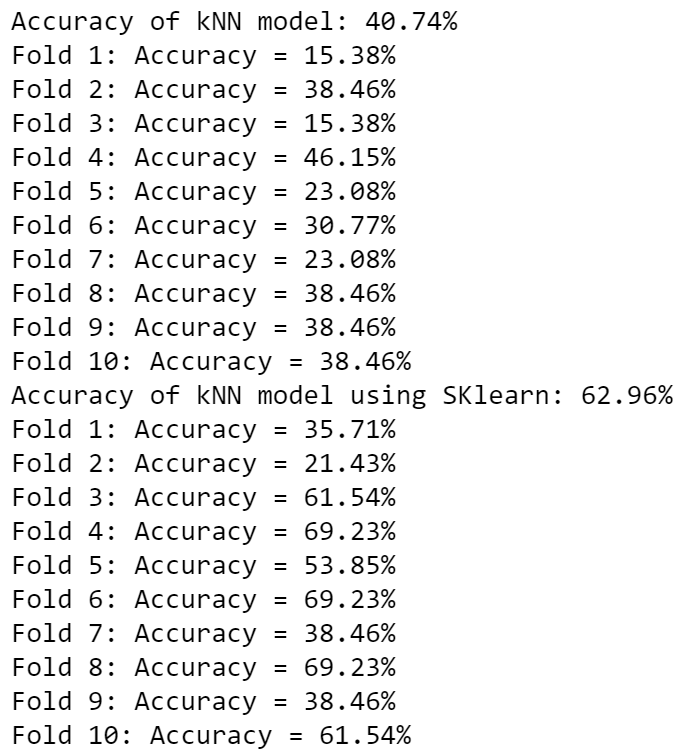
**Accuracies of all three datasets:**

**Accuracy of Hayes Roth dataset:**

Accuracy of KNN without SKLearn : 40.74%

Accuracy of KNN with SKLearn : 62.96%

Below are the10 fold cross validation accuracies of Hayes Roth dataset



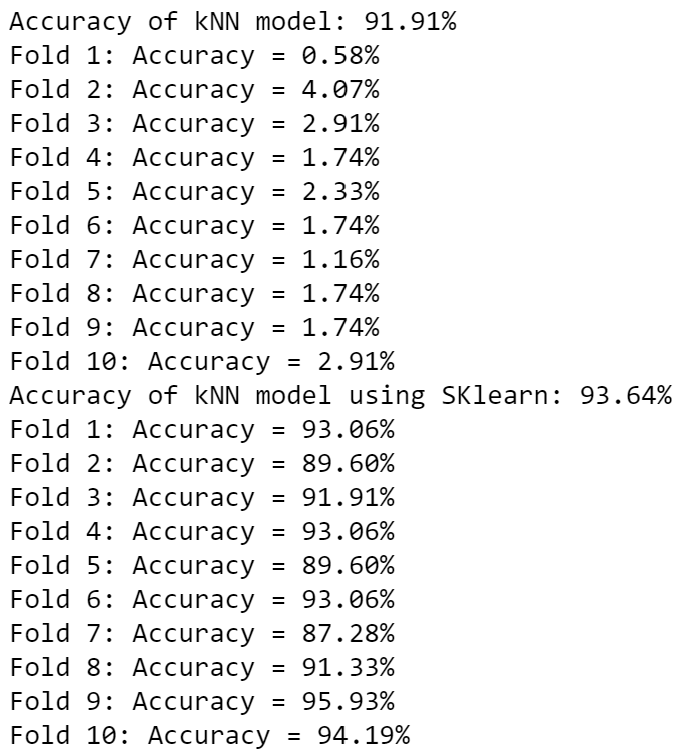
|  |  |  |
| --- | --- | --- |
| **Fold Number** | **Accuracy of KNN without SKLearn** | **Accuracy of KNN with SKLearn** |
| 1 | 15.38% | 35.71% |
| 2 | 38.46% | 21.43% |
| 3 | 15.38% | 61.54% |
| 4 | 46.15% | 69.23% |
| 5 | 23.08% | 53.85% |
| 6 | 30.77% | 69.23% |
| 7 | 23.08% | 38.46% |
| 8 | 38.46% | 69.23% |
| 9 | 38.46% | 38.46% |
| 10 | 38.46% | 61.54% |

**Accuracy of Car Evaluation dataset:**

Accuracy of KNN without SKLearn : 91.91%

Accuracy of KNN with SKLearn : 93.64%

Below are the 10 fold cross validation accuracies of Car Evaluation dataset



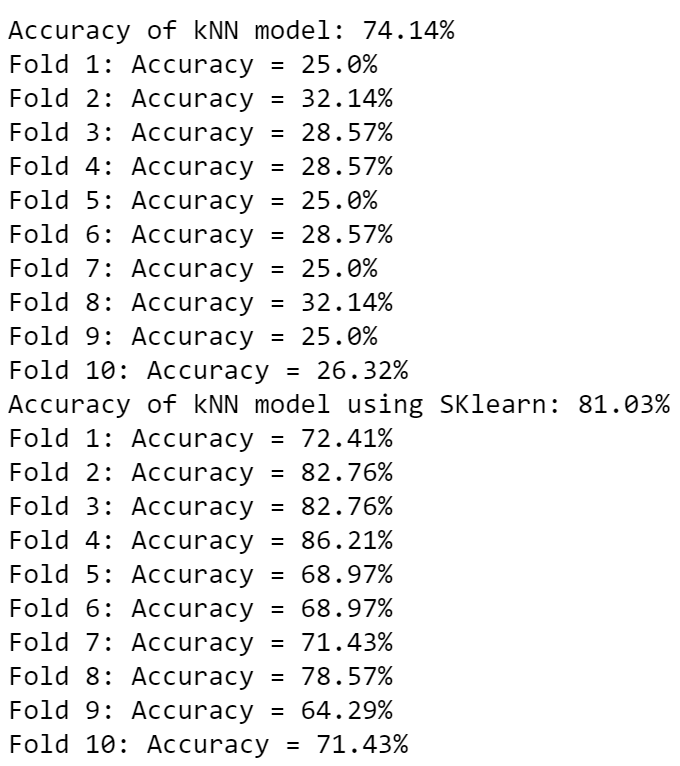
|  |  |  |
| --- | --- | --- |
| **Fold Number** | **Accuracy of KNN without SKLearn** | **Accuracy of KNN with SKLearn** |
| 1 | 0.58% | 93.06% |
| 2 | 4.07% | 89.60% |
| 3 | 2.91% | 91.91% |
| 4 | 1.74% | 93.06% |
| 5 | 2.33% | 89.60% |
| 6 | 1.74% | 93.06% |
| 7 | 1.16% | 87.28% |
| 8 | 1.74% | 91.33% |
| 9 | 1.74% | 95.93% |
| 10 | 2.91% | 94.19% |

**Accuracy of Breast Cancer dataset:**

Accuracy of KNN without SKLearn : 74.14%

Accuracy of KNN with SKLearn : 81.03%

Below are the 10 fold cross validation accuracies of Breast Cancer dataset

****

|  |  |  |
| --- | --- | --- |
| **Fold Number** | **Accuracy of KNN without SKLearn** | **Accuracy of KNN with SKLearn** |
| 1 | 25.0% | 72.41% |
| 2 | 32.14% | 82.76% |
| 3 | 28.57% | 82.76% |
| 4 | 28.57% | 86.21% |
| 5 | 25.0% | 68.97% |
| 6 | 28.57% | 68.97% |
| 7 | 25.0% | 71.43% |
| 8 | 32.14% | 78.57% |
| 9 | 25.0% | 64.29% |
| 10 | 26.32% | 71.43% |

**t- Test values for three different datasets:**

**t-test for Hayes Roth dataset:**

T-statistic: -3.287

P- value: 0.00

A notable significant difference exists between KNN and SKLearn-KNN Accuracies.

The observed difference in accuracies between KNN and SKLearn-KNN is statistically significant, according to a low p-value (0.004). This offers compelling justification for disproving the null hypothesis and demonstrating that there is, in fact, a notable and significant difference between the two approaches.

**t-test for Car Evaluation dataset:**

T-statistic: -90.30

P- value: 2.699

A notable significant difference exists between KNN and SKLearn-KNN Accuracies.

These findings demonstrate a big and highly statistically significant difference between the two approaches, providing strong support for rejecting the null hypothesis.

**t-test for Breast Cancer dataset:**

T-statistic: -17.40

P- value: 1.0

A notable significant difference exists between KNN and SKLearn-KNN Accuracies.

These results show a significant difference between the two methods that is large and statistically significant, which is strong evidence against the null hypothesis.

**T-test and P-values comparison table:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Hayes Roth dataset** | **Car Evaluation dataset** | **Breast Cancer dataset** |
| **T-statistic** | -3.287 | -90.30 | -17.40 |
| **P-value** | 0.00 | 2.699 | 1.0 |

**Links for datasets:**

Hayes Roth Dataset: <https://archive.ics.uci.edu/dataset/44/hayes+roth>

Car Evaluation Dataset: <https://archive.ics.uci.edu/dataset/19/car+evaluation>

Breast Cancer Dataset: <https://archive.ics.uci.edu/dataset/14/breast+cancer>

**References:**

1. <https://machinelearningmastery.com/k-fold-cross-validation/>
2. <https://pub.aimind.so/unveiling-k-nearest-neighbors-knn-a-beginners-guide-4a3df2c31fd>
3. <https://towardsdatascience.com/create-your-own-k-nearest-neighbors-algorithm-in-python-eb7093fc6339>