**MACHINE LEARNING**

Venkata Harshitha Bathala – 1002204412

**AGENDA:**

* INTRODUCTION
* ALGORITHMS USED
* HOW DOES IT WORKS
* ACCUARCY
* HYPOTHESIS TESTING
* CONCLUSION
* REFERENCE

**INTRODUCTION:**

In this project, we aim to explore and evaluate various implementations of the k-Nearest Neighbors (KNN) algorithm, comparing their performance using datasets from different domains. The KNN algorithm is a simple yet effective classification method, where the class label of a new instance is determined by the majority class among its closest neighbors.

We implemented three variations of the KNN classifier:

* **Custom KNN**: A manually crafted version of the KNN algorithm that allows flexibility in choosing the distance metric (Euclidean, Manhattan, Cosine, or Mahalanobis).
* **Scikit-learn KNN**: The standard KNN classifier available in the Scikit-learn library.
* **D-KNN**: A Discernibility-based KNN model that introduces a weighting mechanism to differentiate instances based on class variety within the nearest neighbors.

The datasets used for this evaluation come from a variety of sources, including the Breast Cancer, Car Evaluation, and Hayes-Roth datasets. Each dataset presents unique challenges, including categorical data and missing values, which were carefully preprocessed before applying the classification algorithms.

The primary objective of this study is to compare the accuracy of custom KNN, D-KNN, and standard KNN classifiers using 10-fold cross-validation. We aim to assess how the performance of custom KNN and D-KNN compares to that of standard KNN, particularly focusing on whether the discernibility-based approach (D-KNN) and custom KNN offers a significant improvement over both standard KNN. Additionally, hypothesis testing is conducted to determine if the observed differences in performance among the three algorithms are statistically significant.

**ALGORITHMS:**

**1. Standard K-Nearest Neighbors (KNN)**

Standard KNN is a non-parametric, instance-based learning algorithm used for classification and regression. The algorithm works by identifying the k nearest neighbors to a given data point based on a distance metric (commonly Euclidean distance). The class label of the new point is determined by the majority class among its k nearest neighbors. KNN is simple and effective but can suffer from high computational costs and sensitivity to the choice of k and the distance metric.

**2. Custom K-Nearest Neighbors (Custom KNN)**

Custom KNN is an enhanced version of the standard KNN algorithm, incorporating specific modifications to improve classification accuracy. These modifications may include optimized distance metrics, feature weighting, or different strategies for selecting the neighbors. The aim of custom KNN is to better adapt to the dataset's unique characteristics, thereby achieving improved classification performance compared to the standard KNN.

**3. Discernibility-based K-Nearest Neighbors (D-KNN)**

D-KNN is a variant of the KNN algorithm that focuses on discernibility, which refers to the ability to distinguish between different classes based on feature values. D-KNN utilizes a discernibility measure to prioritize the selection of neighbors that are more informative for classification. This approach aims to enhance the algorithm’s robustness against noise and irrelevant features, potentially leading to improved accuracy in classifying instances. By emphasizing the most discernible neighbors, D-KNN seeks to provide a more effective classification compared to both standard KNN and custom KNN.

**HOW IT WORKS:**

**Custom K-Nearest Neighbors (KNN)**

* Users can select the distance metric (e.g., Euclidean, Manhattan, Cosine, Mahalanobis) and specify the value of k.
* **Distance Calculation**: Computes distances between data points based on the selected metric.
* **Prediction**: Determines the most common class label among the k nearest neighbors.

**D-KNN (Discernibility K-Nearest Neighbors)**

* **Discernibility Vector**: Evaluates class diversity among neighbors.
* **Score Calculation**: Computes scores for each class using discernibility and distances to neighbors, enhancing prediction accuracy.
  + Users can select the distance metric (e.g., Euclidean, Manhattan, Cosine, Mahalanobis) and specify the value of k.
* **Prediction**: More robust against class imbalances by emphasizing diverse neighbor contributions.

**Sklearn KNN (Scikit-Learn KNN)**

* **Standard Implementation**: Utilizes Scikit-Learn’s efficient methods for KNN, following traditional principles.

**Cross-Validation**

* **k-Fold Cross-Validation**: Applied to all algorithms to ensure robust evaluation of model performance and to minimize overfitting.

**Hypothesis Testing**

* **Paired t-Tests**: Conducted to compare the accuracy of Custom KNN, D-KNN, and Sklearn KNN, assessing the significance of performance differences across models.

**ACCURACY:**

The choice of distance metric can significantly affect accuracy in K-Nearest Neighbors algorithms. Different metrics may perform better depending on the dataset's characteristics, leading to varying classification results across the algorithms. We evaluated the accuracy of each implementation using 10-fold cross-validation, allowing us to obtain reliable performance metrics. The mean accuracy was then calculated for each method, revealing the differences in performance among Custom KNN, D-KNN, and Scikit-Learn KNN (Scikit-Learn KNN). This analysis highlights the strengths of each method based on the chosen distance metric and the value of k.

**HYPOTHESIS TEST:**

To assess the performance of these algorithms, paired t-tests are employed to compare their accuracy rates. This statistical analysis enables us to determine whether the differences in accuracy are significant, offering important insights into the effectiveness of each algorithm. This evaluation provides a clearer understanding of which algorithm is more effective under different conditions and configurations.

The **Accuracy** and **Hypothesis test** results for all algorithms are presented in the snapshots below.

**Distance Metrics: Manhattan**

**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**Distance Metrics: EUCLIDEAN**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

In a similar manner, we will evaluate the accuracy and perform hypothesis testing for all algorithms using various distance metrics.

**Conclusion:**

In conclusion, the performance of the algorithms varies based on the distance metric used. For Euclidean and Manhattan distances, Custom KNN and Sklearn KNN show similar accuracy, while D-KNN consistently performs worse. There are no significant differences on the Breast Cancer and Hayes-Roth datasets, but the Car Evaluation dataset reveals some variation. In contrast, with cosine similarity, both Custom KNN and D-KNN outperform Sklearn KNN. Lastly, for Mahalanobis distance, each algorithm excels with different datasets. These findings highlight the importance of choosing the right distance metric and algorithm to improve classification accuracy.

**REFERENCE:**

FOR D-KNN

* [(PDF) Extensions of the k nearest neighbour methods for classification problems (researchgate.net)](https://www.researchgate.net/publication/228961513_Extensions_of_the_k_nearest_neighbour_methods_for_classification_problems)
* [Microsoft Word - Thesis \_post-viva\_ - FINAL \_No Comments\_.doc (bbk.ac.uk)](https://www.dcs.bbk.ac.uk/site/assets/files/1025/voulgaris.pdf)