**Objective:** The project's goal was to extend an existing machine learning application by integrating a K-Nearest Neighbor (KNN) classifier. This integration aimed to complement the pre-existing Naive Bayes classifier by adding a model based on a different foundational principle, thus providing a broader analysis toolset for classification tasks. I also used the Iris.csv dataset for this project.

**Overview of the Existing System:** The existing framework, developed under the guidance of our professor, included a Data Handler Class for managing data operations and a Naive Bayes Classifier designed to predict outcomes based on probability calculations. This setup served as the baseline upon which I used to implement my KNN classifier.

**Development Process:**

1. **Design Decisions:**
   * **Choice of KNN Algorithm:** I chose the KNN algorithm for its ability to classify data based on the closest training examples in the feature space, providing a straightforward, instance-based learning method that contrasts well with the probabilistic approach of Naive Bayes.
   * **Parameter Selection (k=3):** The number of neighbors, k, was set to 3. This decision was based on the standard practice of choosing a small k to minimize the noise's impact on the classification, yet large enough to allow for the generalization of the data.
2. **Implementation Details:**
   * **Integration with Existing Codebase:** The main function was enhanced to accommodate the execution of both the Naive Bayes and KNN classifiers. This involved adjusting data handling methods to ensure compatibility with both classifiers' input requirements.
   * **Key Functions in KNN Implementation:** The implementation of KNN involved calculating Euclidean distances between feature vectors, followed by a voting mechanism among the k-nearest neighbors to decide the label of new instances. Efficient handling of these operations was crucial due to their computational intensity.
3. **Challenges Faced:**
   * **Data Handling Issues:** Initially, there were issues in data formatting and preprocessing arose, requiring adjustments to ensure that the inputs to both classifiers were correctly formatted and normalized.
   * **Algorithmic Complexity:** The KNN algorithm's computational load, primarily due to distance calculations over large datasets, posed challenges. Optimizations were necessary to improve efficiency without compromising the accuracy. This may have been more of a problem with my machine and visual studio code errors.
   * **Integration Complexity:** Integrating the KNN classifier into a framework already built was a challenge as sometimes I didn’t understand pre-existing code, which was already made, and it required me to read through the code and understand it before implementing new code within it.
4. **Role of ChatGPT:**
   * **Debugging Assistance:** ChatGPT played a critical role in debugging the code by identifying logical and syntax errors quickly. For instance, it helped correct a data type mismatch that was causing runtime errors during the distance calculation phase.
   * **Optimization Suggestions:** Suggestions from ChatGPT included using list comprehensions and generator expressions to streamline the code, thereby enhancing readability and performance.
   * **Theoretical Clarifications:** Whenever there were ambiguities in the understanding of the KNN algorithm's mechanics, ChatGPT provided clear, concise explanations that aided in the correct implementation and optimization of the classifier.

**Results:**

**Functionality Testing:** Both classifiers were tested to ensure their functionality. The integration was validated by running both classifiers on a standard dataset and comparing their prediction outputs.

**Performance Metrics:** Results indicated that while the Naive Bayes classifier performed well in terms of speed as expected, the KNN classifier provided good accuracy in certain scenarios, especially where the decision boundary was not linearly definable. The accuracy of the KNN classifier was recorded at 97%, highlighting its effectiveness.

**Conclusion:**

**Future Work:** Future improvements might include experimenting with different values of k and integrating additional machine learning algorithms to further expand the application's scope and usability.