Name – Alamin Seikh

Roll No – 19

Date - 30/10/25

Topic - Naive Bayes Algorithm Report

Naive Bayes is a simple yet powerful machine learning algorithm used for classification tasks. It's based on Bayes' Theorem and assumes that all features in a dataset are independent of each other—a "naive" assumption that often works surprisingly well in practice. The algorithm calculates the probability of each class given the input features and selects the class with the highest likelihood. It's especially effective in text classification problems like spam detection and sentiment analysis due to its speed, efficiency, and ability to handle high-dimensional data. Despite its simplicity, Naive Bayes can deliver strong performance, though it may struggle when features are highly correlated.

1. **Importing Libraries:** The notebook starts by importing necessary libraries for data manipulation, visualization, and machine learning. These include libraries for working with data (like pandas), numerical operations (like numpy), plotting (like matplotlib and seaborn), and machine learning tools (like scikit-learn).
2. **Loading and Preparing Data:** The Iris dataset is loaded, which contains measurements of different parts of iris flowers and their corresponding species. This data is then organized into a structured format (a pandas DataFrame) for easier handling. The species information, initially represented as numbers, is converted into human-readable names.
3. **Exploratory Data Analysis (EDA):** The notebook performs several steps to understand the characteristics of the data. This includes:
    - Checking the dataset's structure, data types, and missing values.
    - Viewing the first few rows to get a sense of the data.
    - Checking the accuracy of the model.
    - Analyzing the distribution of the different iris species to see if the dataset is balanced.
    - Visualizing the distribution of individual features (like sepal length) for each species.
    - Creating pair plots to visualize the relationships between all pairs of features and how they relate to the species.
    - Generating a heatmap to show the correlations between the numerical features, helping to identify which features are strongly related.
4. **Preparing Data for Modeling:** The data is split into features (the measurements) and the target variable (the species). This data is then divided into training and

testing sets. The training set is used to train the machine learning model, and the testing set is used to evaluate its performance on data it hasn't seen before.

5. **Building and Training the Model:** A Gaussian Naive Bayes model is chosen and trained using the training data. This model learns to classify iris flowers into their respective species based on the provided features.
6. **Evaluating the Model:** After training, the model's performance is assessed using the test set. Several metrics are calculated:
    - **Accuracy Score:** This measures the overall percentage of correct predictions made by the model.
    - **Confusion Matrix:** This table shows how many instances of each species were correctly and incorrectly classified. It helps identify which species are being confused with each other.
    - **Classification Report:** This provides more detailed metrics like precision, recall, and F1-score for each species, giving a more nuanced view of the model's performance for each class.
7. **Visualizing the Confusion Matrix:** A heatmap is created to visually represent the confusion matrix, making it easier to interpret the model's performance across different species.
8. **Visualizing Decision Boundaries (for Understanding):** The notebook also includes a section to visualize the decision boundaries of the trained model, but using only two features (sepal length and sepal width) for simplicity. This helps to understand how the model separates the different classes in the feature space. This part seems to be primarily for illustrative purposes.