Experiment 3.2

Student Name: Suraj Sagar **UID: 21BCS3388**
Branch: CSE **Section/Group: 602-B**
Semester: 5th **Date of Performance:24/10/23**
Subject Name: AI&ML **Subject Code:21CSH-316**

1. **Aim:** Implement Naïve Bayes theorem to classify the English text

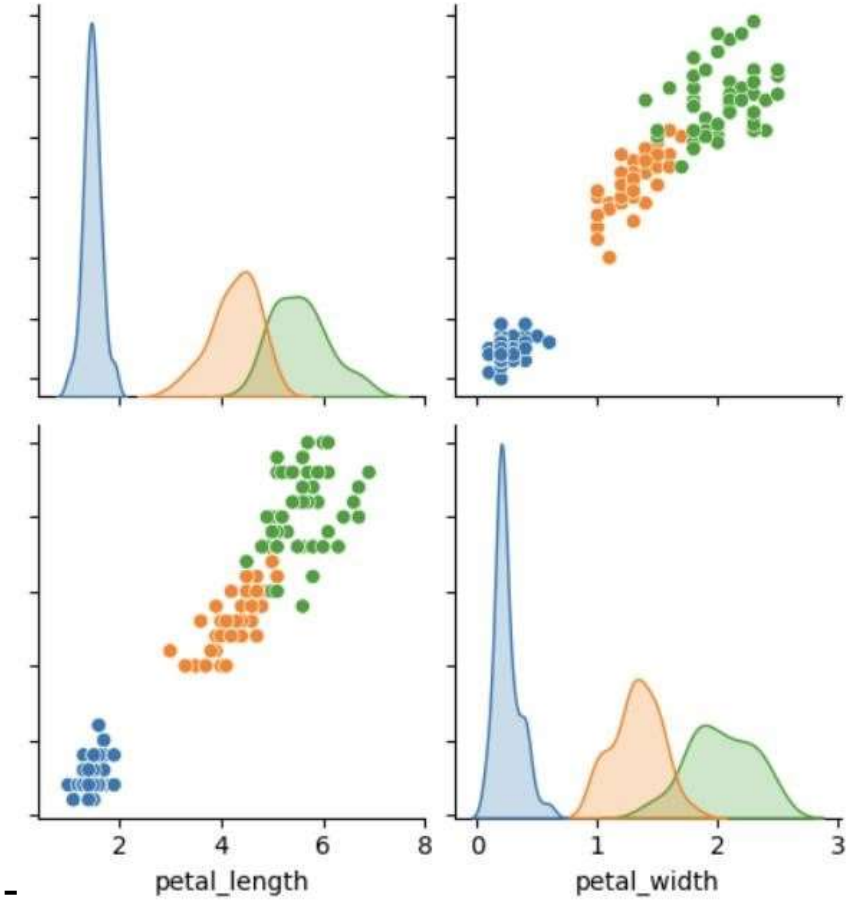
2. **Objective:** The objective of this experiment is to understand how to calculate the probabilities required by the Naive Bayes algorithm and How to implement the Naive Bayes algorithm from scratch How to apply Naive Bayes to a real-world predictive modeling problem.

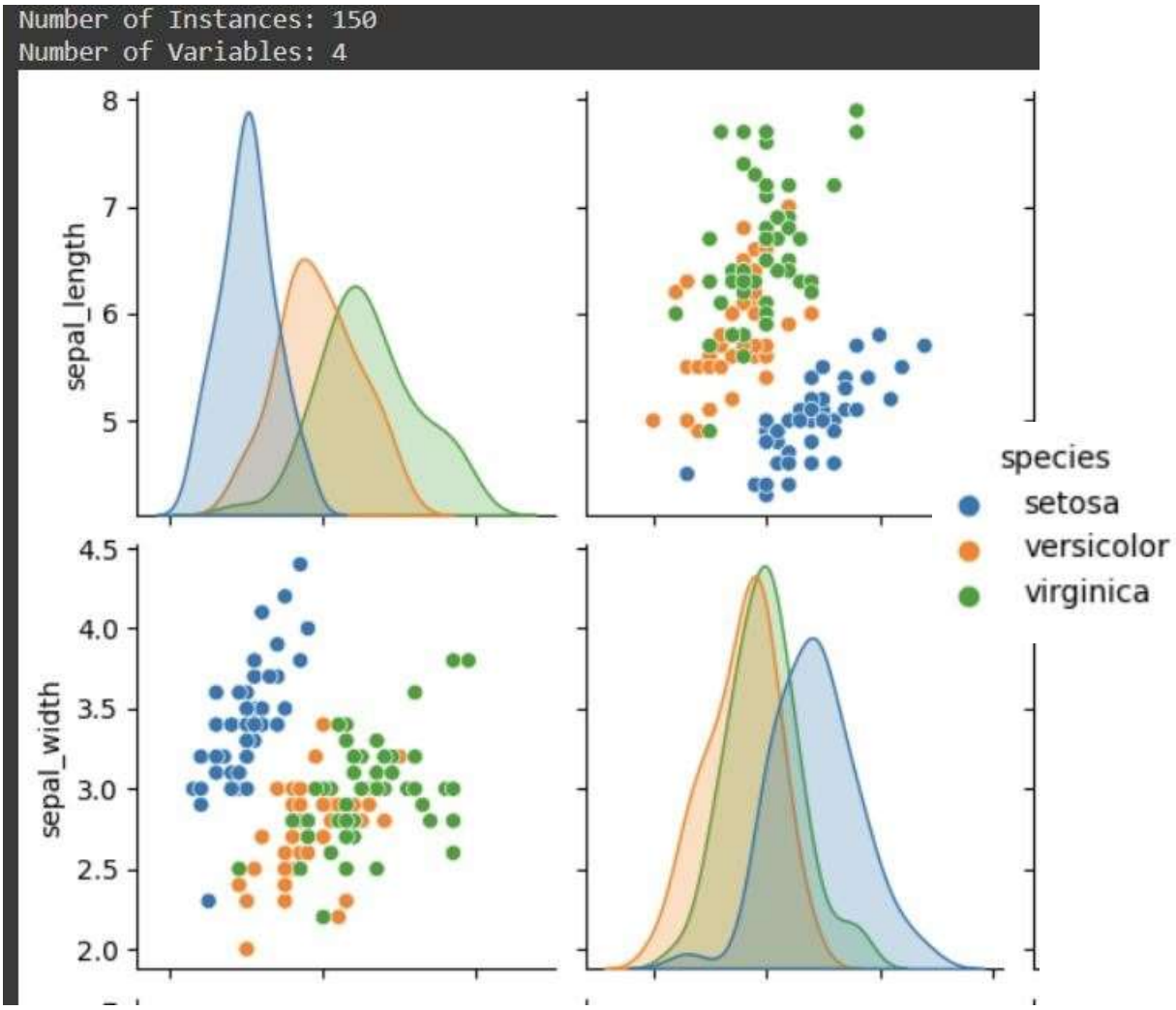
3. **Sample Code-**

```
from sklearn import datasets
iris = datasets.load_iris() X
= iris.data
y = iris.target num_instances = len(X)
num_variables = X.shape[1] print("Number
of Instances:", num_instances) print("Number
of Variables:", num_variables) import
seaborn as sns import matplotlib.pyplot as plt
iris_df = sns.load_dataset("iris")
sns.pairplot(iris_df, hue="species") plt.show()
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

clf = GaussianNB() clf.fit(X_train,
y_train)

y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred) print("Accuracy:",
accuracy)
```

4. Outcome





5. Code Explanation-

1. This line imports the 'datasets' module from the Scikit-Learn library, which provides access to built-in datasets, including the Iris dataset.
2. The code loads the Iris dataset using the **load_iris** function from the datasets module. It assigns the dataset to the variable **iris**.
3. Here, **X** represents the feature matrix, and **y** represents the target labels. **X** contains the measurements of sepal length, sepal width, petal length, and petal width for

each sample, and **y** contains the corresponding class labels (0 for Setosa, 1 for Versicolor, and 2 for Virginica).

4. These lines compute the number of instances (rows) and the number of variables (columns) in the feature matrix **X** and store them in the variables **num_instances** and **num_variables**.
5. In this section, Seaborn and Matplotlib libraries are used for data visualization. It loads the Iris dataset into a Pandas DataFrame **iris_df**, and then it creates a pair plot (scatterplot matrix) using **sns.pairplot**. The **hue="species"** parameter colorcodes the data points by species. The **plt.show()** function displays the plot.
6. This code uses Scikit-Learn's **train_test_split** function to split the dataset into training and testing sets. It randomly shuffles the data and assigns 80% of it to the training set (**X_train** and **y_train**) and 20% to the testing set (**X_test** and **y_test**).
7. Here, a Gaussian Naïve Bayes classifier is created and initialized as **clf**. This classifier assumes that the features follow a Gaussian (normal) distribution.
8. The **fit** method is used to train the Naïve Bayes classifier on the training data. It learns the probability distribution of each feature for each class.
9. After training, the **predict** method is used to make predictions on the test set (**X_test**), and the predicted labels are stored in **y_pred**.
10. Finally, the code calculates the accuracy of the model's predictions by comparing the predicted labels (**y_pred**) to the true labels (**y_test**). The accuracy score is printed to the console.