Experiment-1.3

Student Name: Ashish Kumar **UID: 23MAI10008**
Branch: ME CSE AIML **Section/Group: 23MAI-1**
Semester: 02 **Date of Performance: 31/01/2024**
Subject Name: Machine Learning Lab **Subject Code: 23CSH-651**

Aim of the Experiment :

Develop an application for implementing the Naive Bayes Classifier.

Theory :

Naive Bayes Classifier is a supervised machine learning algorithm which is used to solve the classification problems. It is based on Bayes theorem. The word "Naive" indicates the assumption that all variables are independent to each other and the word "Bayes" relates to Bayes theorem.

Naive Bayes classifiers are computationally efficient and often perform well even with relatively small datasets. However, the assumption of feature independence may not hold true in many real-world scenarios, which can lead to suboptimal performance, especially when features are correlated.

Bayes' Theorem:

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred.

Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are events and $P(B) \neq 0$

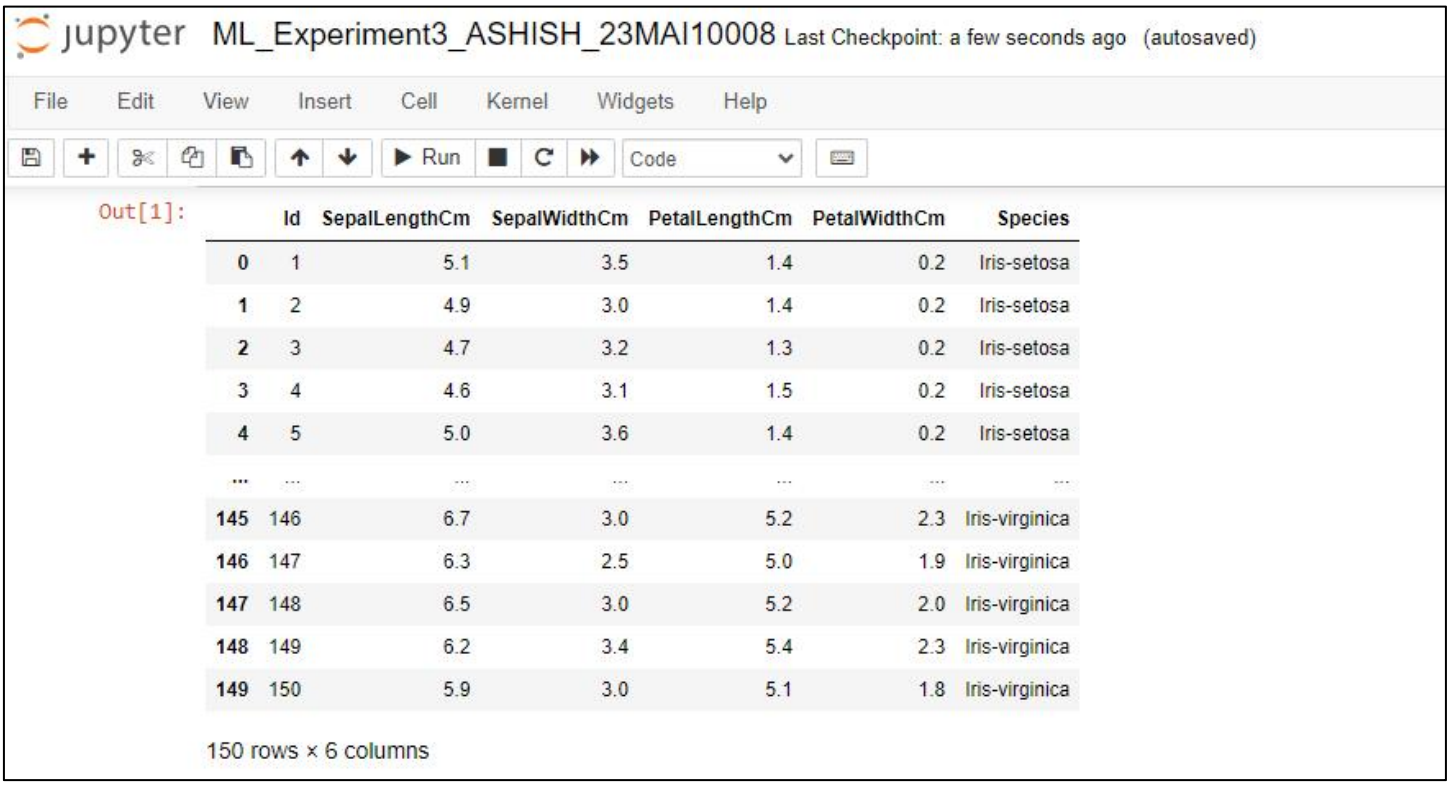
P(A|B) is the probability of occurrence of any event A, when another event B in relation to A has already occurred. This also means the probability of event A depends on another event B.

Code for Experiment :

```
# Importing the libraries
import pandas as pd

# Read the dataset
dataset=pd.read_csv("Iris.csv")
dataset

# Categorical to Numeric
dataset['Species'].replace(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'],[0, 1, 2],inplace=True)
# Assign Independent and Dependent Variables
x = dataset.iloc[:, [1,2,3,4]].values
y = dataset.iloc[:, 5].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 42)

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB

classifier = GaussianNB()
classifier.fit(x_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(x_test)

from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
```

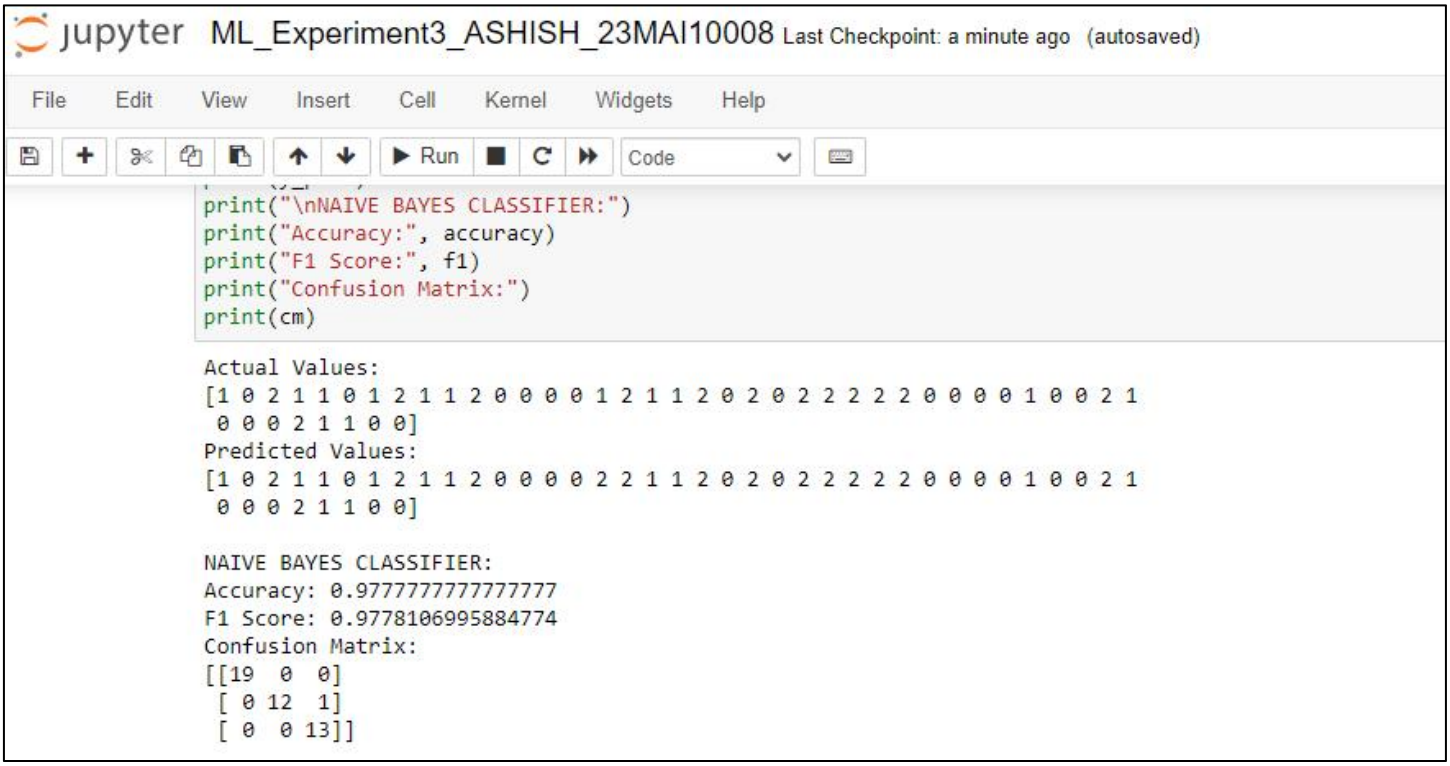
```
# Calculate accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_pred, y_test, average="weighted")
cm = confusion_matrix(y_test, y_pred)

print("Actual Values:")
print(y_test)
print("Predicted Values:")
print(y_pred)
print("\nNAIVE BAYES CLASSIFIER:")
print("Accuracy:", accuracy)
print("F1 Score:", f1)
print("Confusion Matrix:")
print(cm)
```

Result/Output :



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns



Learning outcomes (What I have learnt):

1. I learnt about various python libraries like pandas, sklearn.
2. I learnt about the concept of Naive Bayes Classifier.
3. I learnt about how to read the dataset using pandas.
4. I learnt about how to calculate the Accuracy and F1-score.
5. I learnt about the training and testing dataset for building the model.