Tamilarasi R - 240711017

**Ex 6: Naive Bayesian Classifier**

**The Naive Bayes Classifier algorithm using Python and apply it to the Play Tennis dataset to predict whether a game will be played based on weather conditions.**

**Aim:**

To implement the Naive Bayesian Classifier using the Tennis dataset to predict whether a person will play tennis or not, based on weather conditions, and evaluate the model's performance..

**Algorithm:**

1. Import necessary libraries: pandas, sklearn.tree, sklearn.preprocessing, sklearn.naive\_bayes, and sklearn.metrics.
2. Load the dataset (tennisdata.csv) using pandas.read\_csv().
3. Separate the features (X) and the output class label (y).
4. Encode all categorical columns (Outlook, Temperature, Humidity, Windy, and PlayTennis) using LabelEncoder to convert strings to numerical values.
5. Split the dataset into training and testing sets using train\_test\_split() with a test size of 20%.
6. Initialize the Naïve Bayes classifier using GaussianNB().
7. Train the model on the training set using classifier.fit(X\_train, y\_train).
8. Predict the class label for test data using classifier.predict(X\_test).
9. Evaluate the model's accuracy using accuracy\_score().
10. End

**Program:**

**Step 1: Import Libraries**

import pandas as pd

from sklearn import tree

from sklearn.preprocessing import LabelEncoder

from sklearn.naive\_bayes import GaussianNB

**Step 2: Load Dataset**

data = pd.read\_csv('tennisdata.csv')

print("The first 5 values of data is:\n", data.head())

**Step 3: Separate Features and Output**

X = data.iloc[:, :-1]

print("\nThe First 5 values of train data are:\n", X.head())

y = data.iloc[:, -1]

print("\nThe first 5 values of Train output are:\n", y.head())

**Step 4: Encode Categorical Features**

le\_outlook = LabelEncoder()

X.Outlook = le\_outlook.fit\_transform(X.Outlook)

le\_Temperature = LabelEncoder()

X.Temperature = le\_Temperature.fit\_transform(X.Temperature)

le\_Humidity = LabelEncoder()

X.Humidity = le\_Humidity.fit\_transform(X.Humidity)

le\_Windy = LabelEncoder()

X.Windy = le\_Windy.fit\_transform(X.Windy)

print("\nNow the Train data is:\n", X.head())

**Step 5: Encode Output Label**

le\_PlayTennis = LabelEncoder()

y = le\_PlayTennis.fit\_transform(y)

print("\nNow the Train output is:\n", y)

**Step 6: Train the Naive Bayes Classifier**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)

classifier = GaussianNB()

classifier.fit(X\_train, y\_train)

from sklearn.metrics import accuracy\_score

print("Accuracy is:", accuracy\_score(classifier.predict(X\_test), y\_test))

**OUTPUT:**

Now the Train output is

[0 0 1 1 1 0 1 0 1 1 1 1 1 0]

Accuracy is: 0.6666666666666666