**NAME : MANGESH A. GHADWAJE**

**ROLL NO:24**

**BATCH : B2**

**COURSE: ML PRACTICAL**

**Assginment No. 6**

**Problem Statement :**

**Implement Naïve Bayes Classifier on Tennisdata Data set.**

**Evaluate the classifier&#39;s performance**

**Code :**

***# Import libraries***

***import pandas as pd***

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

***from sklearn.metrics import accuracy\_score, precision\_score,classification\_report, confusion\_matrix***

***from sklearn.preprocessing import StandardScaler, LabelEncoder***

***from sklearn.naive\_bayes import GaussianNB***

***import matplotlib.pyplot as plt***

***# Load the dataset***

***data = pd.read\_csv('tennis.csv')***

***print("Dataset:")***

***print(data)***

***print("\n")***

***# Convert categorical variables to numerical using label encoding***

***le = LabelEncoder()***

***data['outlook'] = le.fit\_transform(data['outlook'])***

***data['temp'] = le.fit\_transform(data['temp'])***

***data['humidity'] = le.fit\_transform(data['humidity'])***

***data['windy'] = le.fit\_transform(data['windy'])***

***data['play'] = le.fit\_transform(data['play'])***

***print("--------Independent Variables--------")***

***# select all rows & columns from 2nd up to but not including, the last column***

***X=data.iloc[:, : -1] #X=data.iloc[:,:4] #another option***

***print(X)***

***print("\n")***

***print("---------Dependent Variables---------")***

***Y = data['play'] # Target variable***

***print(Y)***

***print("\n")***

***# Split the dataset into training and testing sets***

***X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.20, random\_state=42)***

***# Standardize features***

***scaler = StandardScaler()***

***X\_train = scaler.fit\_transform(X\_train)***

***X\_test= scaler.transform(X\_test)***

***#print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*",X\_test)***

***# Gaussian Naive Bayes classifier***

***guassian\_classifier = GaussianNB()***

***guassian\_classifier.fit(X\_train, y\_train)***

***# Predictions on test dataset***

***y\_pred = guassian\_classifier.predict(X\_test)***

***print("Classfication report")***

***print(classification\_report(y\_test, y\_pred))***

***print("\n")***

***print("Confusion Matrix")***

***print(confusion\_matrix(y\_test, y\_pred))***

***print("\n")***

***# Calculate accuracy***

***accuracy = accuracy\_score(y\_test, y\_pred)\*100***

***print("Accuracy:",accuracy,"%")***

***# Calculate precision***

***precision = precision\_score(y\_test, y\_pred)\*100***

***print("Precision:", precision, "%")***

**Output :**



