# Naïve Bayes Classifier Algorithm

**CSE 303: Machine Learning**

Submitted by

Name: Roda Chinthapalli

Roll No: AP22110011496

Section: M

Lab Date: 28/10/24

Submission Date: 4/11/24

**A picture containing text

Description automatically generated**

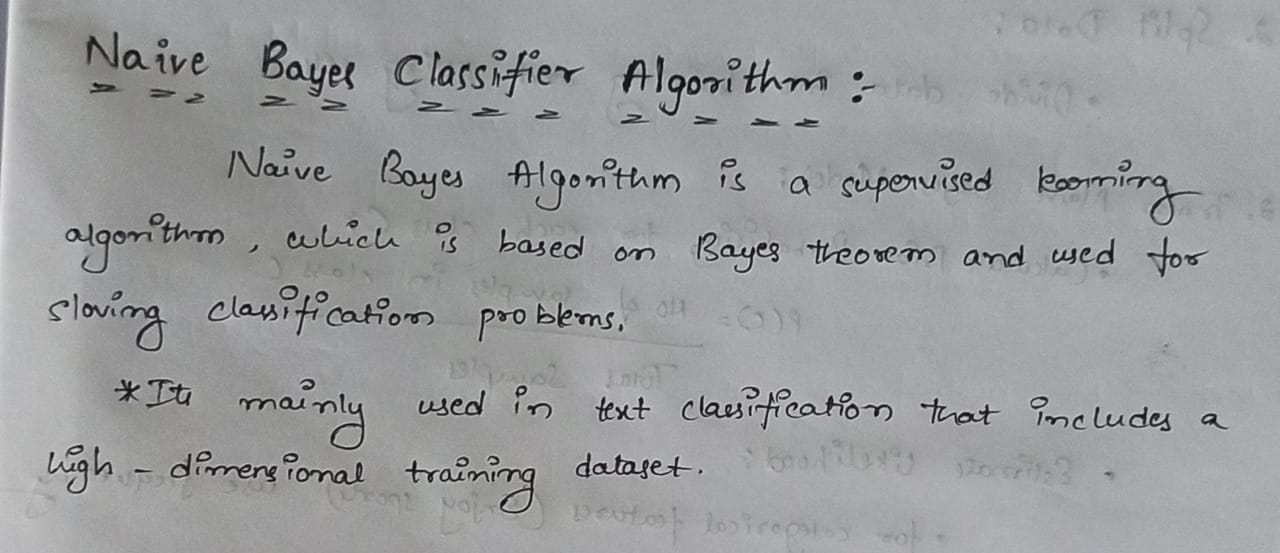
**Department Computer Science and Engineering**

**School of Engineering and Sciences**

**SRM University–AP**

**Amaravati, Andhra Pradesh – 522 240, India**

**NAIVE BAYES CLASSIFIER**

****

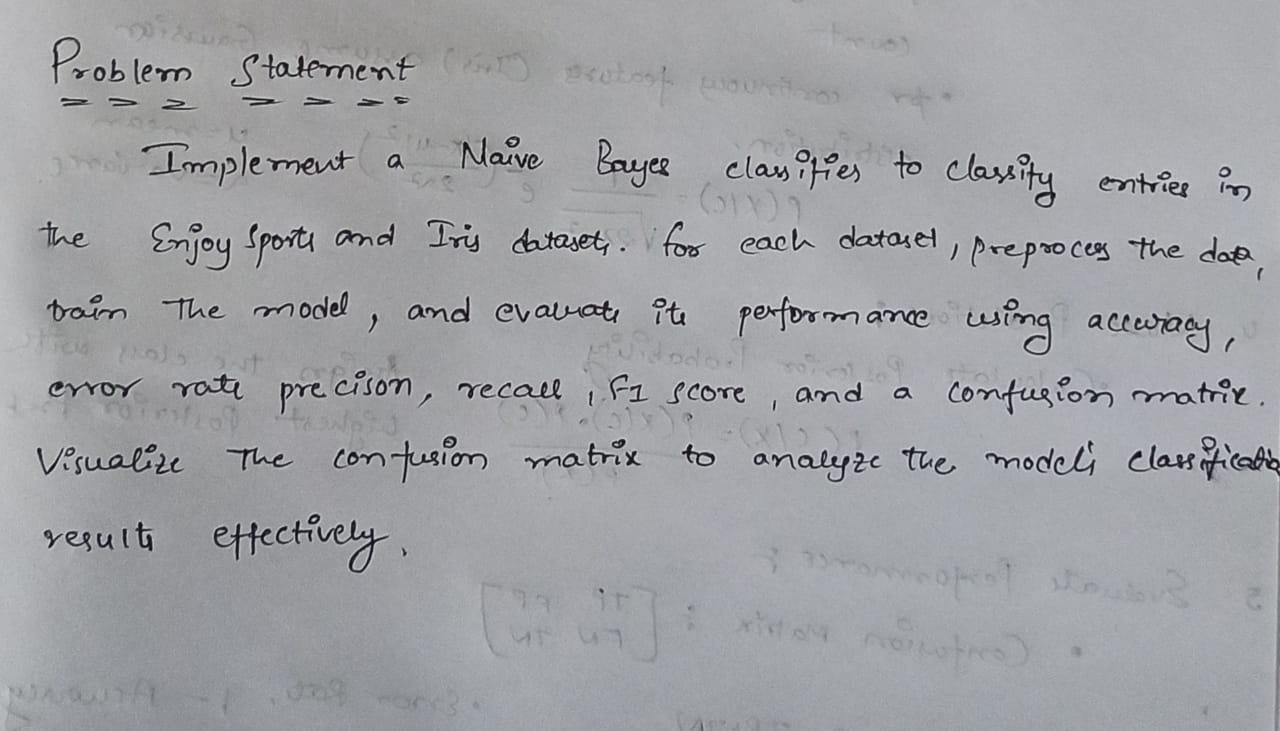
**1.Question:**

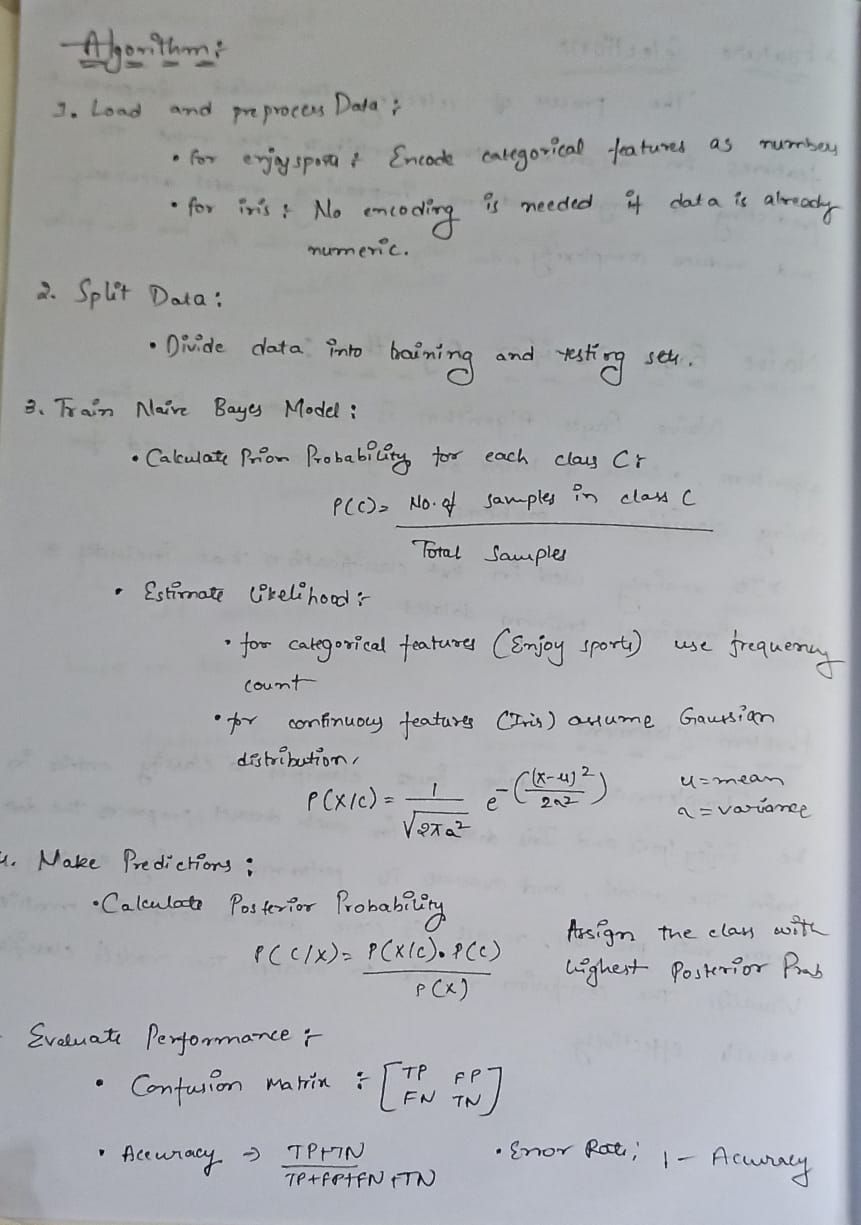
**Implement Naïve Bayes classifier for following datasets and evaluate the classification performance. Draw the confusion matrix, compute accuracy, error and other measures as applicable.**

**Datasets:1) EnjoySports**

**2) Iris**

**2.Problem Statement and Algorithm:**

****

****

**3.Solution:**

**Enjoysports dataset:**

**# Import necessary libraries**

**import pandas as pd**

**import numpy as np**

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

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

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

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**# Load the dataset**

**data = pd.read\_csv('/content/Enjoy sports.csv')**

**# Data Pre-processing: Convert categorical features to numerical using Label Encoding**

**from sklearn.preprocessing import LabelEncoder**

**labelencoder = LabelEncoder()**

**for column in data.columns:**

**data[column] = labelencoder.fit\_transform(data[column])**

**# Separate features (X) and target (y)**

**X = data.iloc[:, :-1].values # All columns except the last one as features**

**y = data.iloc[:, -1].values # Last column as the target variable**

**# Split the dataset into training and testing sets (70% train, 30% test)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)**

**# Fitting Naive Bayes to the Training set**

**nb\_classifier = GaussianNB()**

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

**# Predicting the test results**

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

**# Test accuracy of the result (Creation of Confusion matrix)**

**conf\_matrix = confusion\_matrix(y\_test, y\_pred)**

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

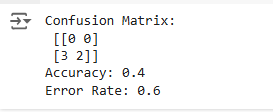
**error\_rate = 1 - accuracy**

**# Print the Confusion Matrix, Accuracy, and Error Rate**

**print("Confusion Matrix:\n", conf\_matrix)**

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

**print("Error Rate:", error\_rate)**

****

**# Visualize the Confusion Matrix using a heatmap**

**plt.figure(figsize=(6, 4))**

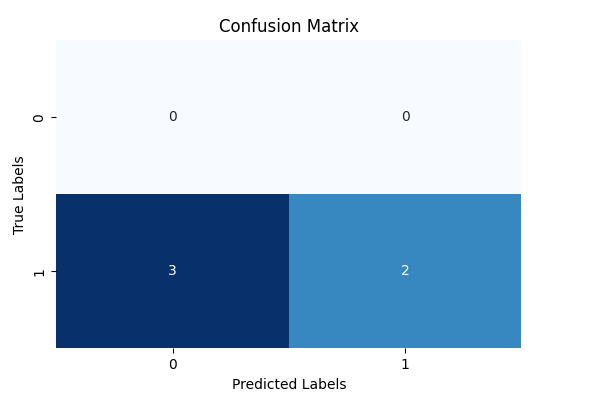
**sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)**

**plt.title("Confusion Matrix")**

**plt.xlabel("Predicted Labels")**

**plt.ylabel("True Labels")**

**plt.show()**

****

**Iris Dataset:**

**# Import necessary libraries**

**import pandas as pd**

**import numpy as np**

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

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

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

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**# Load the dataset**

**data = pd.read\_csv('/content/iris.csv')**

**# Separate features (X) and target (y)**

**X = data.iloc[:, :-1].values # All columns except the last one as features**

**y = data.iloc[:, -1].values # Last column as the target variable**

**# Split the dataset into training and testing sets (70% train, 30% test)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)**

**# Fitting Naive Bayes to the Training set**

**nb\_classifier = GaussianNB()**

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

**# Predicting the test results**

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

**# Test accuracy of the result (Creation of Confusion matrix)**

**conf\_matrix = confusion\_matrix(y\_test, y\_pred)**

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

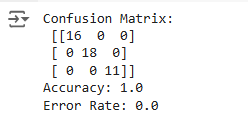
**error\_rate = 1 - accuracy**

**# Print the Confusion Matrix, Accuracy, and Error Rate**

**print("Confusion Matrix:\n", conf\_matrix)**

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

**print("Error Rate:", error\_rate)**

****

**# Visualize the Confusion Matrix using a heatmap**

**plt.figure(figsize=(6, 4))**

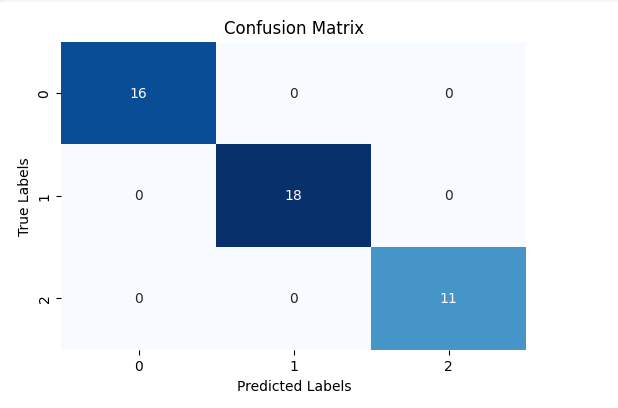
**sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)**

**plt.title("Confusion Matrix")**

**plt.xlabel("Predicted Labels")**

**plt.ylabel("True Labels")**

**plt.show()**

****

**4.Code Repository:**

**Github: https://github.com/Roda1458/Naive-Bayes-Classifier**