**NAVIE BAYES LEARNING**

**CSE 303: Machine Learning**

Submitted by

Name: Ridhi Guntur

Roll No: AP22110011467

Section: CSE M

Lab Date: 29/10/2024

Submission Date: 04/11/2024

**A picture containing text

Description automatically generated**

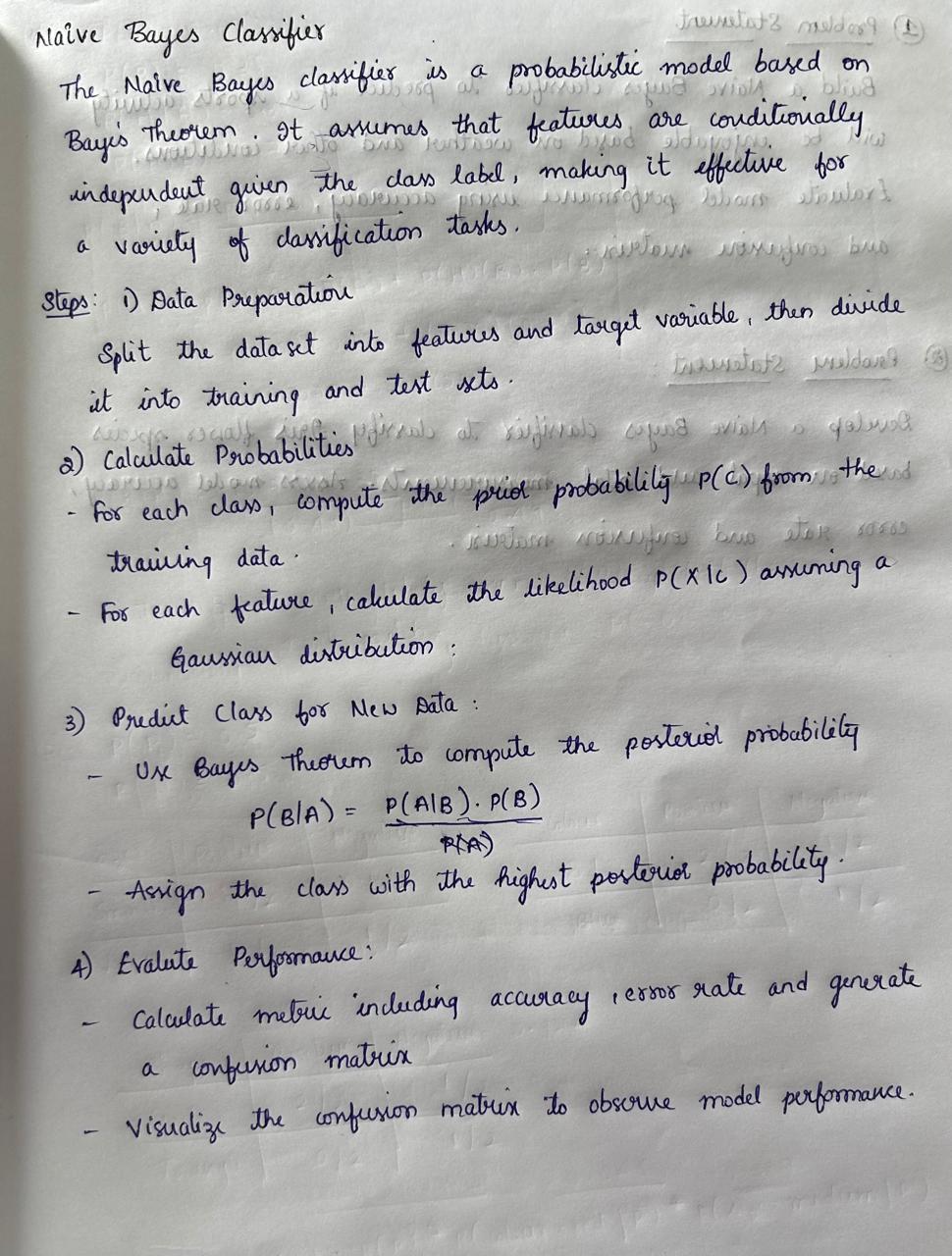
**Department Computer Science and Engineering**

**School of Engineering and Sciences**

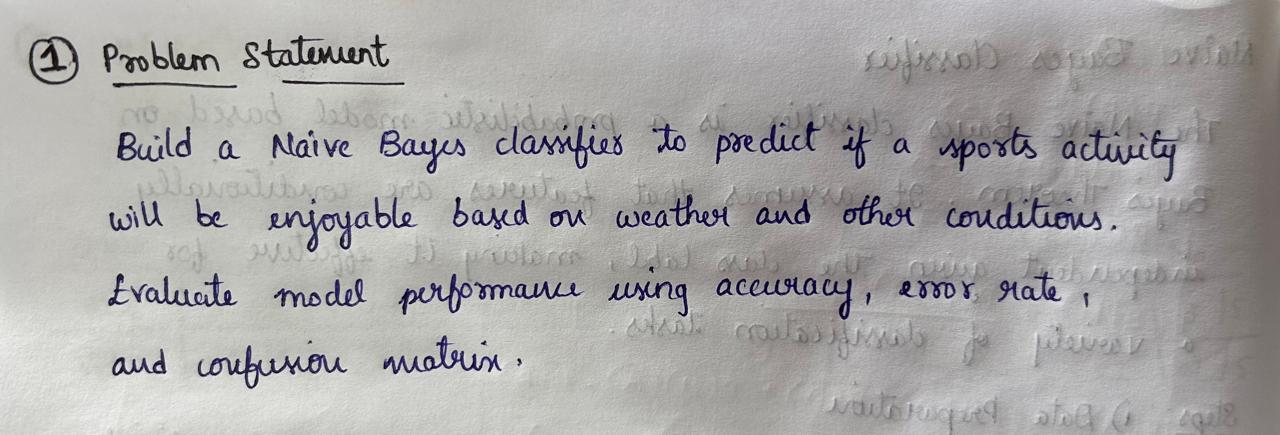
**SRM University–AP**

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

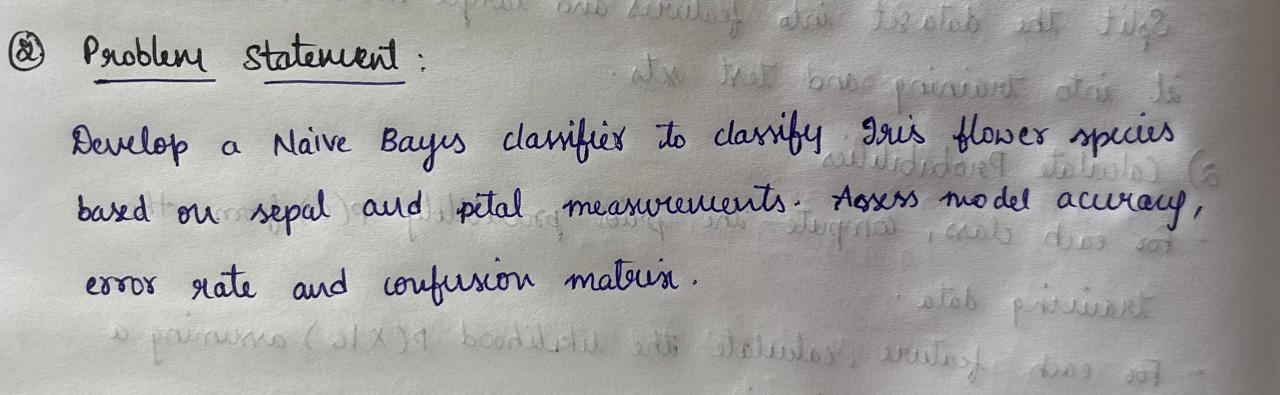
1. **Algorithm Description**



1. **Solution**



|  |
| --- |
| # Importing necessary libraries  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import LabelEncoder  from sklearn.naive\_bayes import GaussianNB  from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report  import pandas as pd  import seaborn as sns  import matplotlib.pyplot as plt  # Load dataset  data = pd.read\_csv(r"C:\Users\guntu\Downloads\Enjoy sports.csv")  # Drop 'Day' column, as it is not a feature  data = data.drop(columns=['Day'])  # Encode categorical variables  le = LabelEncoder()  for column in data.columns:  data[column] = le.fit\_transform(data[column])  # Split dataset into features (X) and target (y)  X = data.drop(columns=['Decision'])  y = data['Decision']  # Split data into training and test sets (70% train, 30% test)  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)  # Initialize and train Naive Bayes model  nb\_model = GaussianNB()  nb\_model.fit(X\_train, y\_train)  # Predict test set results  y\_pred = nb\_model.predict(X\_test)  conf\_matrix = confusion\_matrix(y\_test, y\_pred)  print("Confusion Matrix:\n", conf\_matrix)  plt.figure(figsize=(6, 4))  sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Predicted No', 'Predicted Yes'], yticklabels=['Actual No', 'Actual Yes'])  plt.xlabel('Predicted')  plt.ylabel('Actual')  plt.title('Confusion Matrix Heatmap')  plt.show()  accuracy = accuracy\_score(y\_test, y\_pred)  print("Accuracy:", accuracy)  error\_rate = 1 - accuracy  print("Error Rate:", error\_rate) |



|  |
| --- |
| # Import necessary libraries  import pandas as pd  from sklearn.model\_selection import train\_test\_split  from sklearn.naive\_bayes import GaussianNB  from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report  import seaborn as sns  import matplotlib.pyplot as plt  # Load the iris dataset  data = pd.read\_csv(r"C:\Users\guntu\Downloads\iris.csv" )  # Display the first few rows (optional)  print(data.head())  # Assuming the Iris dataset has features in columns 1-4 and target in the last column  X = data.iloc[:, :-1] # Features  y = data.iloc[:, -1] # Target  # Split data into training and test sets (70% train, 30% test)  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)  # Initialize and train the Naive Bayes model  nb\_model = GaussianNB()  nb\_model.fit(X\_train, y\_train)  # Predict test set results  y\_pred = nb\_model.predict(X\_test)  # Display confusion matrix  conf\_matrix = confusion\_matrix(y\_test, y\_pred)  print("Confusion Matrix:\n", conf\_matrix)  plt.figure(figsize=(6, 4))  sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=nb\_model.classes\_, yticklabels=nb\_model.classes\_)  plt.xlabel('Predicted')  plt.ylabel('Actual')  plt.title('Confusion Matrix Heatmap')  plt.show()  accuracy = accuracy\_score(y\_test, y\_pred)  print("Accuracy:", accuracy)  error\_rate = 1 - accuracy  print("Error Rate:", error\_rate) |

1. **Code Repository:**

GitHub link : <https://github.com/Ridhi-215/Navie_Bayes>