**Classification Using Naive Bayes Technique**



**CCP Report (Theory Project)**

**Abdul Ahad Raza (RC-294)**

**Aashan Shehzad (RC-295)**

**Muhammad Talha (RC-302)**

Submitted To: **Dr. Sajid Saleem**

Department of Computer Science

**NATIONAL UNIVERSITY OF MODERN LANGUAGES RAWALPINDI**

**May 14th, 2024**

**Classification Using Naive Bayes Technique**

**Introduction**

Naive Bayes classifier for a dataset. It begins by loading the data, conducting exploratory data analysis (EDA), and preprocessing steps such as handling missing values and encoding categorical variables. Then, it splits the data into training and testing sets, builds and trains a Gaussian Naive Bayes classifier, makes predictions on the test data, and evaluates the model's performance using accuracy and a classification report

]**Technologies Used**

* **Python:** The primary programming language used for back-end logic and machine learning algorithms.
* **Pandas:** A Python library used for reading .csv file in this project.
* **Matplotlib:** A Python library utilized for visualization of results in terms of graphs and figures.
* **Scikit-learn:** A Python library used for the purpose of implementing built-in functions on datasets as per requirements.
* **Seaborn:** A Python library used for making statistical graphics.

|  |
| --- |
| **NaiveBayesClassification.py**  import pandas as pd  from sklearn.model\_selection import train\_test\_split  from sklearn.naive\_bayes import GaussianNB  from sklearn.metrics import accuracy\_score, classification\_report  import matplotlib.pyplot as plt  import seaborn as sns  data = pd.read\_csv("/content/OSID.csv")  print(data.head())  numerical\_columns = ['Administrative', 'Informational', 'ProductRelated']  for column in numerical\_columns:      plt.figure(figsize=(8, 6))      sns.histplot(data[column], bins=20, kde=True)      plt.title(f'Histogram of {column}')      plt.xlabel(column)      plt.ylabel('Frequency')      plt.show()  plt.figure(figsize=(8, 6))  sns.countplot(data['Revenue'])  plt.title('Histogram of Revenue')  plt.xlabel('Revenue')  plt.ylabel('Count')  plt.show()  data = data.dropna()  # Encode categorical variables if any  data = pd.get\_dummies(data)  X = data.drop('Revenue', axis=1) # Assuming 'Revenue' is the target variable  y = data['Revenue']  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  nb\_classifier = GaussianNB()  nb\_classifier.fit(X\_train, y\_train)  y\_pred = nb\_classifier.predict(X\_test)  accuracy = accuracy\_score(y\_test, y\_pred)  report\_dict = classification\_report(y\_test, y\_pred, output\_dict=True)  report\_df = pd.DataFrame(report\_dict)  print("Accuracy:", accuracy)  print("Classification Report:")  print(report\_df)  **Explaination:**   * **Loading and Preparing Data**   We start by getting information about Online shoppers purchasing intention from single named OSID.csv. Then, weextracted the features like Administrative, Administrative\_Duration,Informational,Informational\_Duration, ProductRelated, ProductRelated\_Duration, BounceRatesa, ExitRates, PageValues, SpecialDay, Month, OperatingSystems, Browser Regio, TrafficType, VisitorType and Weekend Revenue.   * **Turning Results into Graphs**   Computers understand numbers better than words. So, we convert all the words in the movie descriptions, genres, and cast into numbers. This helps us compare movies later.   * **Finding Classification results**   We calculated classification aspects like accuracy , F1-Score, precission, recall and support results using scikit-learn library functionalities.  **Output:**  **Dateset Visualization**  1  2  **Graphical Representation (Histograms)**  34Capture6  **Classification Report:**  7 |