**Problem Statement:**

Apply naïve Bayesian algorithm on buy computer dataset to identify class label of

unknown samples.

**Dataset Description:**

The dataset you provided appears to be related to predicting whether or not people will play golf based on different weather conditions.

Columns:

Outlook: Describes the weather outlook.

Categories: Sunny, Overcast, Rain.

Temp. (Temperature): Describes the temperature during the day.

Categories: Hot, Mild, Cool.

Humidity: Describes the humidity level.

Categories: High, Normal.

Wind: Describes the wind speed.

Categories: Weak, Strong.

Decision: The decision of whether or not to play golf.

Categories: Yes, No.

The dataset contains 10 records, each representing one instance of weather conditions and the corresponding decision regarding playing golf. The columns are a mix of categorical features (Outlook, Temp., Humidity, Wind) and a binary target variable (Decision).

**Procedure:**

Step 1: Prepare the Data: Handle missing values, encode categorical variables using LabelEncoder.

Step 2: Divide the dataset into training and testing sets.

Step 3:

Calculate the prior probability of each class in the target variable:

Step 4: Calculate Conditional Probabilities.

Step 5: Calculate the probabilities using Bayesian formula.

**Source Code:**

import pandas as pd

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

from sklearn.naive\_bayes import GaussianNB

from sklearn.preprocessing import LabelEncoder

# Dataset

data = {

'Age': ['youth', 'youth', 'youth', 'middle', 'senior', 'senior', 'middle', 'youth', 'middle', 'senior'],

'Income': ['low', 'high', 'high', 'medium', 'low', 'low', 'high', 'medium', 'low', 'medium'],

'Student': ['no', 'yes', 'no', 'yes', 'yes', 'yes', 'yes', 'no', 'no', 'yes'],

'Credit\_Rating': ['fair', 'excellent', 'fair', 'fair', 'fair', 'excellent', 'excellent', 'fair', 'excellent', 'fair'],

'Buy\_Computer': ['yes', 'no', 'yes', 'yes', 'yes', 'yes', 'yes', 'no', 'yes', 'yes']

}

# Create DataFrame

df = pd.DataFrame(data)

# Initialize LabelEncoder for each column separately

le\_age = LabelEncoder()

le\_income = LabelEncoder()

le\_student = LabelEncoder()

le\_credit\_rating = LabelEncoder()

le\_buy\_computer = LabelEncoder()

# Fit the LabelEncoder on each column of the training data

df['Age'] = le\_age.fit\_transform(df['Age'])

df['Income'] = le\_income.fit\_transform(df['Income'])

df['Student'] = le\_student.fit\_transform(df['Student'])

df['Credit\_Rating'] = le\_credit\_rating.fit\_transform(df['Credit\_Rating'])

df['Buy\_Computer'] = le\_buy\_computer.fit\_transform(df['Buy\_Computer'])

# Features and target

X = df[['Age', 'Income', 'Student', 'Credit\_Rating']]

y = df['Buy\_Computer']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

# Train a Naïve Bayes classifier

nb = GaussianNB()

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

# Predict class labels for test samples

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

# Predict for unknown samples

unknown\_samples = pd.DataFrame({

'Age': ['senior'],

'Income': ['low'],

'Student': ['yes'],

'Credit\_Rating': ['excellent']

})

print(f"Inserted data\n", unknown\_samples)

# Use the same LabelEncoder transformations to the unknown samples

unknown\_samples['Age'] = le\_age.transform(unknown\_samples['Age'])

unknown\_samples['Income'] = le\_income.transform(unknown\_samples['Income'])

unknown\_samples['Student'] = le\_student.transform(unknown\_samples['Student'])

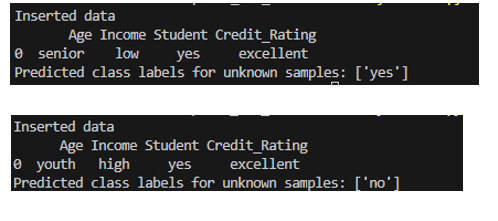
unknown\_samples['Credit\_Rating'] = le\_credit\_rating.transform(unknown\_samples['Credit\_Rating'])

# Make predictions using the Naïve Bayes classifier

predicted\_labels = nb.predict(unknown\_samples[['Age', 'Income', 'Student', 'Credit\_Rating']])

print("Predicted class labels for unknown samples:", le\_buy\_computer.inverse\_transform(predicted\_labels))

**Output:**



**Discussion:**

The implementation of the Naive Bayes algorithm in code is a straightforward and efficient way to apply the model to classification tasks. The code typically involves key steps such as preprocessing the data, calculating prior probabilities, computing conditional probabilities (likelihoods), and applying Bayes’ Theorem to predict the most likely class for new data instances.