Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Continuous Assessment for Laboratory / Assignment sessions

Academic Year 2022-23

Name: Ayush Jain SAP ID: 60004200132

Course: Machine Learning Laboratory Course Code: **DJ19CEEL6021**

Year: **T.Y. B.Tech.** Sem: **VII** Batch: B 3

Department: Computer Engineering

Performance Indicators (Any no. of Indicators) (Maximum 5 marks per indicator)	1	2	3	4	5	6	7	8	9	10	11	Σ	Avg	A1	A2	Σ	Avg
Course Outcome	2, 4	2, 4	2, 4	2, 4	2, 4	3	2, 4	2, 4	5								
1. Knowledge (Factual/Conceptual/Procedural/ Metacognitive)	4	5	5	5	5	5	5	5									
2. Describe (Factual/Conceptual/Procedural/ Metacognitive)	4	4	4	5	5	4	4	4									
3. Demonstration (Factual/Conceptual/Procedural/ Metacognitive)	5	5	5	5	5	5	5	4									
4. Strategy (Analyse & / or Evaluate) (Factual/Conceptual/ Procedural/Metacognitive)	4	4	4	4	4	5	5	5									
5. Interpret/ Develop (Factual/Conceptual/ Procedural/Metacognitive)	–	–	–	–	–	–	–	–									
6. Attitude towards learning (receiving, attending, responding, valuing, organizing, characterization by value)	4	4	4	4	5	4	4	5									
7. Non-verbal communication skills/ Behaviour or Behavioural skills (motor skills, hand-eye coordination, gross body movements, finely coordinated body movements speech behaviours)	–	–	–	–	–	–	–	–									
Total	21	22	22	23	24	23	23	23									
Signature of the faculty member																	

Outstanding (5), Excellent (4), Good (3), Fair (2), Needs Improvement (1)

Laboratory marks Σ Avg. =	Assignment marks Σ Avg. =	Total Term-work (25) =
Laboratory Scaled to (15) =	Assignment Scaled to (10) =	Sign of the Student:

Signature of the Faculty member:
Name of the Faculty member:

Signature of Head of the Department
Date:

Machine Learning

Experiment 8

Ayush Jain 60004200132 B3

Aim : Implement Bayesian Classification

Theory:

Bayesian classification uses Bayes theorem to predict the occurrence of any event. Bayesian classifiers are the statistical classifiers with the Bayesian probability understandings. The theory expresses how a level of belief, expressed as a probability.

Bayes theorem came into existence after Thomas Bayes, who first utilized conditional probability to provide an algorithm that uses evidence to calculate limits on an unknown parameter.

Bayes's theorem is expressed mathematically by the following equation that is given below.

$$P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}$$

Where X and Y are the events and P (Y) ≠ 0

P(X/Y) is a **conditional probability** that describes the occurrence of event **X** is given that **Y** is true.

P(Y/X) is a **conditional probability** that describes the occurrence of event **Y** is given that **X** is true.

P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the **marginal probability**.

Bayesian interpretation:

In the Bayesian interpretation, probability determines a "**degree of belief**." Bayes theorem connects the degree of belief in a hypothesis before and after accounting

for evidence. For example, Lets us consider an example of the coin. If we toss a coin, then we get either heads or tails, and the percent of occurrence of either heads and tails is 50%. If the coin is flipped numbers of times, and the outcomes are observed, the degree of belief may rise, fall, or remain the same depending on the outcomes.

For proposition X and evidence Y,

- P(X), the prior, is the primary degree of belief in X
- P(X/Y), the posterior is the degree of belief having accounted for Y.
- The quotient $\frac{P(Y/X)}{P(Y)}$ represents the supports Y provides for X.

Bayes theorem can be derived from the conditional probability:

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

Where P (X⋂Y) is the **joint probability** of both X and Y being true, because

$$P(Y \cap X) = P(X \cap Y)$$

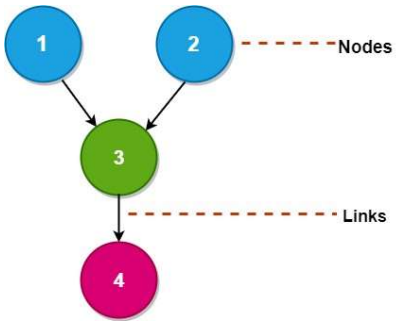
$$\text{or, } P(X \cap Y) = P(X/Y)P(Y) = P(Y/X)P(X)$$

$$\text{or, } P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}, \text{ if } P(Y) \neq 0$$

Bayesian network:

A Bayesian Network falls under the classification of Probabilistic Graphical Modelling (PGM) procedure that is utilized to compute uncertainties by utilizing the probability concept. Generally known as **Belief Networks, Bayesian Networks** are used to show uncertainties using **Directed Acyclic Graphs** (DAG)

A **Directed Acyclic Graph** is used to show a Bayesian Network, and like some other statistical graph, a DAG consists of a set of nodes and links, where the links signify the connection between the nodes.



The nodes here represent random variables, and the edges define the relationship between these variables.

A DAG models the uncertainty of an event taking place based on the Conditional Probability Distribution (CDP) of each random variable. A **Conditional Probability Table** (CPT) is used to represent the CPD of each variable in a network

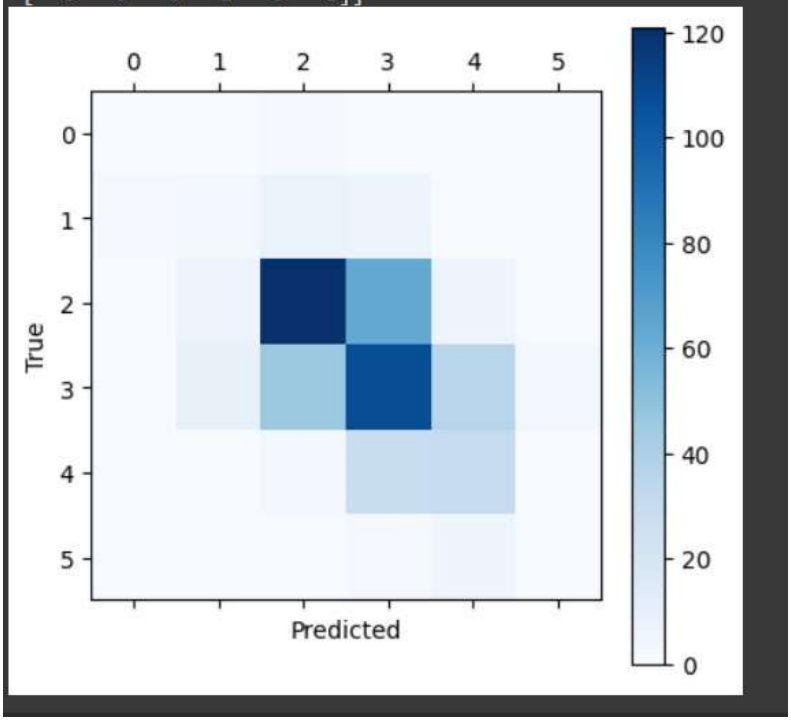
Code:

```
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 import
matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-
databases/winequality/winequality-red.csv', delimiter=';')

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data.drop('quality', axis=1), data['qua
lity'], test_size=0.3, random_state=42)

# Create a Gaussian Naive Bayes classifier
clf = GaussianNB()

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Test the classifier on the testing data
y_pred = clf.predict(X_test)

# Evaluate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
# Print the confusion matrix cm =
confusion_matrix(y_test, y_pred)
print('Confusion Matrix:') print(cm)

# Plot the confusion matrix plt.matshow(cm,
cmap=plt.cm.Blues)
plt.colorbar() plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

Output:

```
Accuracy: 0.5416666666666666
Confusion Matrix:
[[  0   0   1   0   0   0]
 [  1   2   8   6   0   0]
 [  0   6 121  63   5   0]
 [  0   9  46 107  35   3]
 [  0   0   3  28  30   0]
 [  0   0   0   1   5   0]]
```



Conclusion: We successfully implemented Bayesian Classification.