**EXPERIMENT 2**

**AIM:**

Implementation of Classification algorithm using:

1. Decision Tree ID3
2. Naïve Bayes algorithm

**THEORY:**

**Naïve Bayes:**

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. Bayes’ theorem states the following relationship, given class variable y and dependent feature vector x1 through xn, :

Text

Description automatically generated

and we can use Maximum A Posteriori (MAP) estimation to estimate P(y) and P(xi∣y); the former is then the relative frequency of class y in the training set.

Text

Description automatically generated

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of P(xi∣y).

**Decision Tree:**

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

**ID3 (Iterative Dichotomiser 3)** was developed in 1986 by Ross Quinlan. The algorithm creates a multiway tree, finding for each node (i.e. in a greedy manner) the categorical feature that will yield the largest information gain for categorical targets. Trees are grown to their maximum size and then a pruning step is usually applied to improve the ability of the tree to generalise to unseen data.

**Advantages Decision Tree:**

* Simple to understand and to interpret. Trees can be visualised.
* Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
* Able to handle both numerical and categorical data.
* Able to handle multi-output problems.
* Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.

**Disadvantages Decision Trees:**

* Decision-tree learners can create over-complex trees that do not generalise the data well. This is called overfitting.
* Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.
* Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

**CODE:**

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

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import r2\_score

from sklearn.metrics import classification\_report , auc ,accuracy\_score , precision\_score , recall\_score, roc\_auc\_score , f1\_score, roc\_curve

# from sklearn.metrics import f1\_sco

from sklearn.naive\_bayes import GaussianNB

from sklearn import tree

def trainModel(X ,y , data\_name):

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

# Naive Bayes

nb = GaussianNB()

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

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

acc\_nb = r2\_score(y\_test , y\_pred\_nb)

cm\_nb = confusion\_matrix(y\_test , y\_pred\_nb)

# precision\_nb = precision\_score(y\_test , y\_pred\_nb ,average='binary')

report\_nb = classification\_report(y\_test , y\_pred\_nb ,labels=[1,0])

print('\n\*\*\*\*\*', data\_name , '\*\*\*\*\*')

print('\nNaives Bayes : ')

print('Accuracy : ' ,acc\_nb)

# print('Precision : ' ,precision\_nb)

print('\nConfusion Matrix : \n' ,cm\_nb)

print('\nReport : \n' , report\_nb)

# Decision Tree

dt = tree.DecisionTreeClassifier()

dt = dt.fit(X\_train, y\_train)

y\_pred\_dt = dt.predict(X\_test)

acc\_dt = r2\_score(y\_test , y\_pred\_dt)

cm\_dt = confusion\_matrix(y\_test , y\_pred\_dt)

# precision\_dt = precision\_score(y\_test , y\_pred\_dt ,average='binary')

report\_dt = classification\_report(y\_test , y\_pred\_dt ,labels=[1,0])

print('\nDecision Tree : ')

print('Accuracy : ' ,acc\_dt)

# print('Precision : ' ,precision\_dt)

print('\nConfusion Matrix : \n' ,cm\_dt)

print('\nReport : \n' , report\_dt)

return (acc\_nb , acc\_dt)

// DATASET

models= []

names = []

from sklearn.datasets import load\_digits

X, y = load\_digits(return\_X\_y=True)

acc\_nb\_4 , acc\_dt\_4 = trainModel(X, y , 'LOAD DIGITS')

models.append((acc\_nb\_4 , acc\_dt\_4))

names.append('LOAD DIGITS')

from sklearn.datasets import load\_ine

X, y = load\_wine(return\_X\_y=True)

acc\_nb\_2 , acc\_dt\_2 = trainModel(X, y , 'DIABETES DATASET')

models.append((acc\_nb\_2 , acc\_dt\_2))

names.append('DIABETES')

from sklearn.datasets import load\_breast\_cancer

X, y = load\_breast\_cancer(return\_X\_y=True)

acc\_nb\_3 , acc\_dt\_3 = trainModel(X, y , 'BREAST CANCER')

models.append((acc\_nb\_3 , acc\_dt\_3))

names.append('BREAST CANCER')

from sklearn.datasets import load\_iris

X, y = load\_iris(return\_X\_y=True)

acc\_nb\_1 , acc\_dt\_1 = trainModel(X, y , 'IRIS DATASET')

models.append(( acc\_nb\_1 , acc\_dt\_1))

names.append('IRIS')

from sklearn.datasets import fetch\_olivetti\_faces

X, y = fetch\_olivetti\_faces(return\_X\_y=True)

acc\_nb\_5 , acc\_dt\_5 = trainModel(X, y , 'OLIVETTI FACES')

models.append((acc\_nb\_5 , acc\_dt\_5))

names.append('OLIVETTI FACES')

**OUTPUT:**

**PART A**

1. Diabetes Dataset

Table

Description automatically generated

Table

Description automatically generated

1. Breast Cancer

Table

Description automatically generated

Table

Description automatically generated

1. Load DigitCalendar

   Description automatically generatedA picture containing calendar

   Description automatically generated
2. **Iris Dataset**

Table

Description automatically generatedTable

Description automatically generated

1. Olivetti Faces

Table

Description automatically generated

Table

Description automatically generated with low confidence

**Comparison**

Table

Description automatically generated

**Conclusion:**In this experiment I implemented Naïve Bayes and Decision Tree on 5 datasets viz, Iris, Diabetes, Load Digits, Breast Cancer and Olivetti Faces and displayed the accuracy, precision, recall, confusion matrix. Both Decision Tree and Naïve Bayes did well on Daibetes and Iris dataset. Decision Tree had very low accuracy in Olivetti Faces.