

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

Name: Alaikya S Yemul

Roll No: 62

ASSIGNMENT NO: 12

Title: Prediction in data mining for a given data set

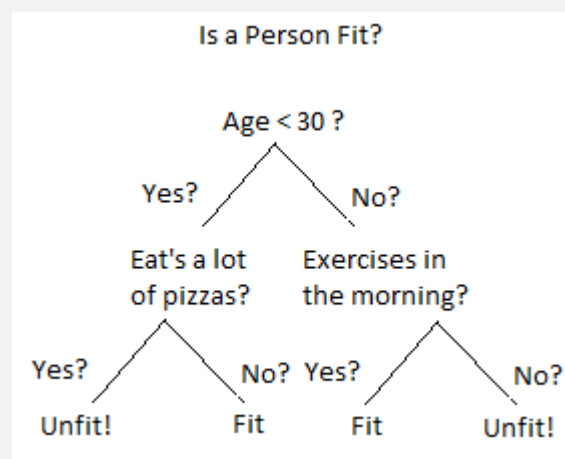
- a. Implement Decision Tree classifier.
- b. Implement Bayesian Classifier for a given dataset.

Theory:

A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.

Decision Tree consists of :

1. **Nodes** : Test for the value of a certain attribute.
2. **Edges/ Branch** : Correspond to the outcome of a test and connect to the next node or leaf.
3. **Leaf nodes** : Terminal nodes that predict the outcome (represent class labels or class distribution).



To understand the concept of Decision Tree consider the above example. Let's say you want to predict whether a person is fit or unfit, given their information like age, eating habits, physical activity, etc. The decision nodes are the questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves represent outcomes like either 'fit', or 'unfit'.

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

Name: Alaikya S Yemul

Roll No: 62

ASSIGNMENT NO: 12

Title: Prediction in data mining for a given data set

- a. Implement Decision Tree classifier.
- b. Implement Bayesian Classifier for a given dataset.

Decision Tree Classifier

- Using the decision algorithm, we start at the tree root and split the data on the feature that results in the **largest information gain (IG)** (reduction in uncertainty towards the final decision).
- In an iterative process, we can then repeat this splitting procedure at each child node **until the leaves are pure**. This means that the samples at each leaf node all belong to the same class.
- In practice, we may set a **limit on the depth of the tree to prevent overfitting**. We compromise on purity here somewhat as the final leaves may still have some impurity.

Bayesian Classifier:

In numerous applications, the connection between the attribute set and the class variable is non-deterministic. In other words, we can say the class label of a test record can't be assumed with certainty even though its attribute set is the same as some of the training examples. These circumstances may emerge due to the noisy data or the presence of certain confusing factors that influence classification, but it is not included in the analysis. For example, consider the task of predicting the occurrence of whether an individual is at risk for liver illness based on individuals eating habits and working efficiency. Although most people who eat healthily and exercise consistently having less probability of occurrence of liver disease, they may still do so due to other factors. For example, due to consumption of the high-calorie street foods and alcohol abuse. Determining whether an individual's eating routine is healthy or the workout efficiency is sufficient is also subject to analysis, which in turn may introduce vulnerabilities into the leaning issue.

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.

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

Name: Alaikya S Yemul

Roll No: 62

ASSIGNMENT NO: 12

Title: Prediction in data mining for a given data set

- a. Implement Decision Tree classifier.
- b. Implement Bayesian Classifier for a given dataset.

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 $P(Y) \neq 0$

$P(X/Y)$ 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**.

Program:

Decision Tree Classifier:

```
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
# Function importing Dataset
def importdata():
    balance_data = pd.read_csv(
        'https://archive.ics.uci.edu/ml/machine-learning- ' +
        'databases/balance-scale/balance-scale.data',
        sep=',', header=None)
    # Printing the dataset shape
    print("Dataset Length: ", len(balance_data))
    print("Dataset Shape: ", balance_data.shape)
    # Printing the dataset observations
    print("Dataset: ", balance_data.head())
    return balance_data
```

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

Name: Alaikya S Yemul

Roll No: 62

ASSIGNMENT NO: 12

Title: Prediction in data mining for a given data set

- a. Implement Decision Tree classifier.
- b. Implement Bayesian Classifier for a given dataset.

```
# Function to split the dataset
def splitdataset(balance_data):
    # Seperating the target variable
    X = balance_data.values[:, 1:5]
    Y = balance_data.values[:, 0]
    # Splitting the dataset into train and test
    X_train, X_test, y_train, y_test = train_test_split(
        X, Y, test_size=0.3, random_state=100)
    return X, Y, X_train, X_test, y_train, y_test
# Function to perform training with giniIndex.
def train_using_gini(X_train, X_test, y_train):
    # Creating the classifier object
    clf_gini = DecisionTreeClassifier(criterion="gini",
        random_state=100, max_depth=3, min_samples_leaf=5)
    # Performing training
    clf_gini.fit(X_train, y_train)
    return clf_gini
# Function to perform training with entropy.
def train_using_entropy(X_train, X_test, y_train):
    # Decision tree with entropy
    clf_entropy = DecisionTreeClassifier(
        criterion="entropy", random_state=100,
        max_depth=3, min_samples_leaf=5)
    # Performing training
    clf_entropy.fit(X_train, y_train)
    return clf_entropy
# Function to make predictions
def prediction(X_test, clf_object):
    # Prediction on test with giniIndex
    y_pred = clf_object.predict(X_test)
    print("Predicted values:")
    print(y_pred)
    return y_pred
# Function to calculate accuracy
def cal_accuracy(y_test, y_pred):
    print("Confusion Matrix: ",
        confusion_matrix(y_test, y_pred))
    print("Accuracy : ",
        accuracy_score(y_test, y_pred) * 100)
```

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

Name: Alaikya S Yemul

Roll No: 62

ASSIGNMENT NO: 12

Title: Prediction in data mining for a given data set

- a. Implement Decision Tree classifier.
- b. Implement Bayesian Classifier for a given dataset.

```
print("Report : ",
      classification_report(y_test, y_pred))

# Driver code
def main():
    # Building Phase
    data = importdata()
    X, Y, X_train, X_test, y_train, y_test = splitdataset(data)
    clf_gini = train_using_gini(X_train, X_test, y_train)
    clf_entropy = train_using_entropy(X_train, X_test, y_train)
    # Operational Phase
    print("Results Using Gini Index:")
    # Prediction using gini
    y_pred_gini = prediction(X_test, clf_gini)
    cal_accuracy(y_test, y_pred_gini)
    print("Results Using Entropy:")
    # Prediction using entropy
    y_pred_entropy = prediction(X_test, clf_entropy)
    cal_accuracy(y_test, y_pred_entropy)
    # Calling main function
    if __name__ == "__main__":
        main()
```

Bayesian Classifier:

```
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import GaussianNB
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import datasets
import pandas as pd
import numpy as np
wine = datasets.load_wine()
print("Features: ", wine.feature_names)
print("Labels: ", wine.target_names)
X = pd.DataFrame(wine['data'])
```

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

Name: Alaikya S Yemul

Roll No: 62

ASSIGNMENT NO: 12

Title: Prediction in data mining for a given data set

- a. Implement Decision Tree classifier.
- b. Implement Bayesian Classifier for a given dataset.

```
print(X.head())
print(wine.data.shape)
y = print(wine.target)
X_train, X_test, y_train, y_test = train_test_split(wine.data,
wine.target, test_size=0.30, random_state=109)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
print(y_pred)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
cm = np.array(confusion_matrix(y_test, y_pred))
print("Confusion matrix:")
print(cm)
dataset = datasets.load_wine()
X = dataset.data
y = dataset.target
print("Features: ", wine.feature_names)
# print the label type of wine
print("Labels: ", wine.target_names)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
model = tree.DecisionTreeClassifier()
model.fit(X_train, y_train)
print(model)
expected_y = y_test
predicted_y = model.predict(X_test)
print(metrics.classification_report(expected_y, predicted_y,
                                     target_names=dataset.target_names))
print("Confusion Matrix:")
print(metrics.confusion_matrix(expected_y, predicted_y))
```

```
Results Using Entropy:
Predicted values:
['R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L' 'L'
 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L'
 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'R' 'R']
Confusion Matrix: [[ 0  6  7]
 [ 0 63 22]
 [ 0 20 70]]
Accuracy : 70.74468085106383
Report :
           precision    recall  f1-score   support

     B       0.00       0.00       0.00        13
     L       0.71       0.74       0.72        85
     R       0.71       0.78       0.74        90

 accuracy          0.71        188
 macro avg         0.47        188
 weighted avg      0.66        188
```

[illegible]

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

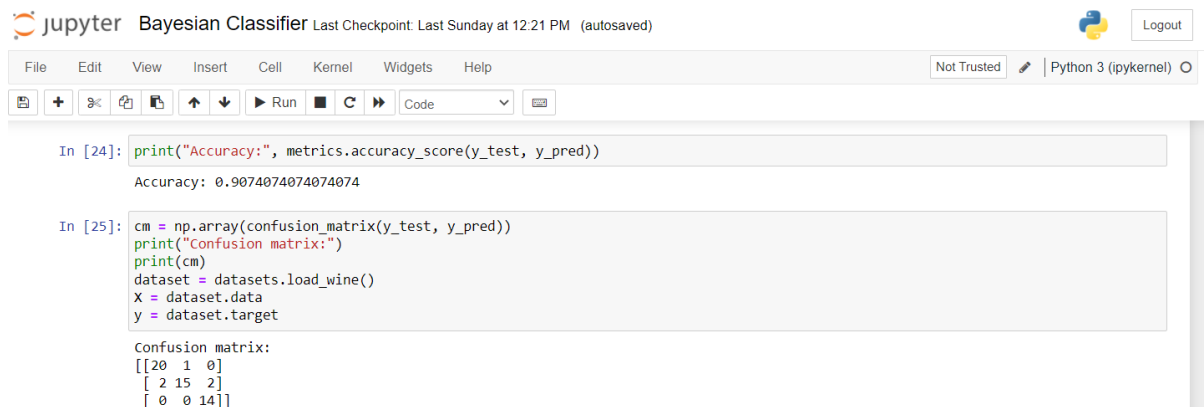
Name: Alaikya S Yemul

Roll No: 62

ASSIGNMENT NO: 12

Title: Prediction in data mining for a given data set

- a. Implement Decision Tree classifier.
- b. Implement Bayesian Classifier for a given dataset.



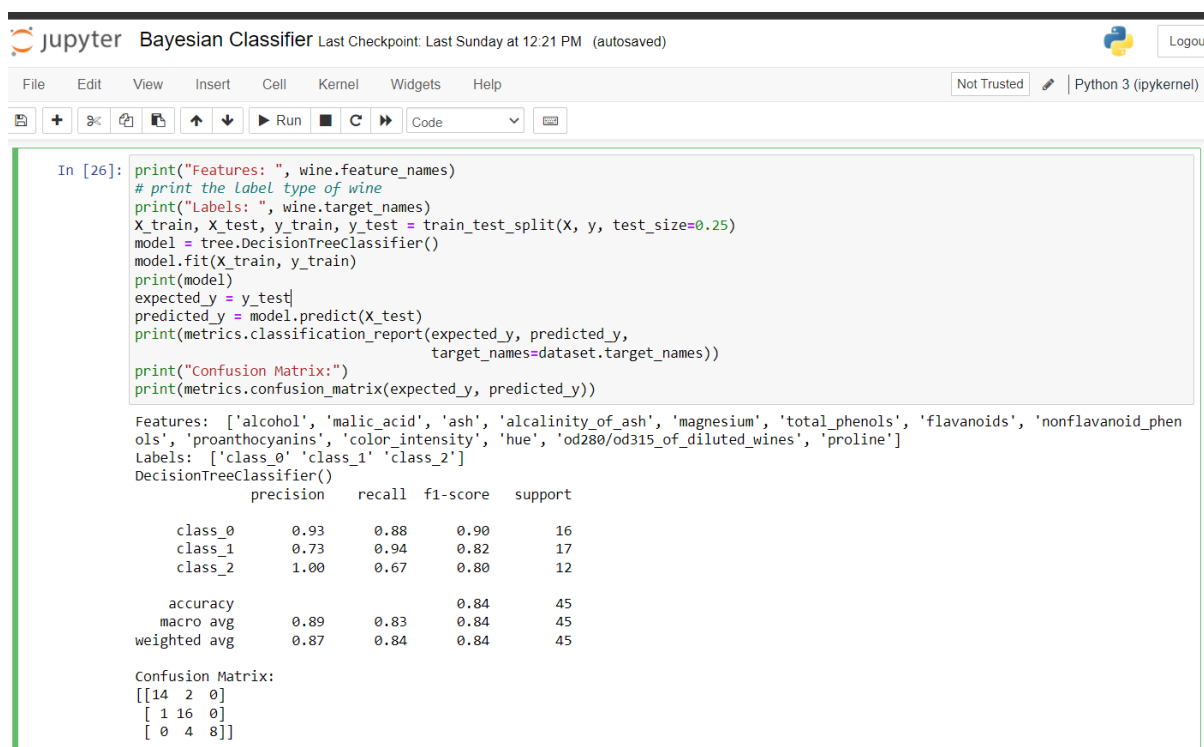
Jupyter Bayesian Classifier Last Checkpoint: Last Sunday at 12:21 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [24]: print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
Accuracy: 0.9074074074074074

In [25]: cm = np.array(confusion_matrix(y_test, y_pred))
print("Confusion matrix:")
print(cm)
dataset = datasets.load_wine()
X = dataset.data
y = dataset.target

Confusion matrix:
[[20  1  0]
 [ 2 15  2]
 [ 0  0 14]]
```



Jupyter Bayesian Classifier Last Checkpoint: Last Sunday at 12:21 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [26]: print("Features: ", wine.feature_names)
# print the label type of wine
print("Labels: ", wine.target_names)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
model = tree.DecisionTreeClassifier()
model.fit(X_train, y_train)
print(model)
expected_y = y_test
predicted_y = model.predict(X_test)
print(metrics.classification_report(expected_y, predicted_y,
                                   target_names=dataset.target_names))

print("Confusion Matrix:")
print(metrics.confusion_matrix(expected_y, predicted_y))

Features: ['alcohol', 'malic_acid', 'ash', 'alkalinity_of_ash', 'magnesium', 'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins', 'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
Labels: ['class_0' 'class_1' 'class_2']
DecisionTreeClassifier()
precision    recall  f1-score   support

   class_0     0.93     0.88     0.90         16
   class_1     0.73     0.94     0.82         17
   class_2     1.00     0.67     0.80         12

 accuracy          0.84         45
  macro avg       0.89     0.83     0.84         45
 weighted avg     0.87     0.84     0.84         45

Confusion Matrix:
[[14  2  0]
 [ 1 16  0]
 [ 0  4  8]]
```