

## Implementation of Naïve Bayes Classification Model

**Ex No: 5**

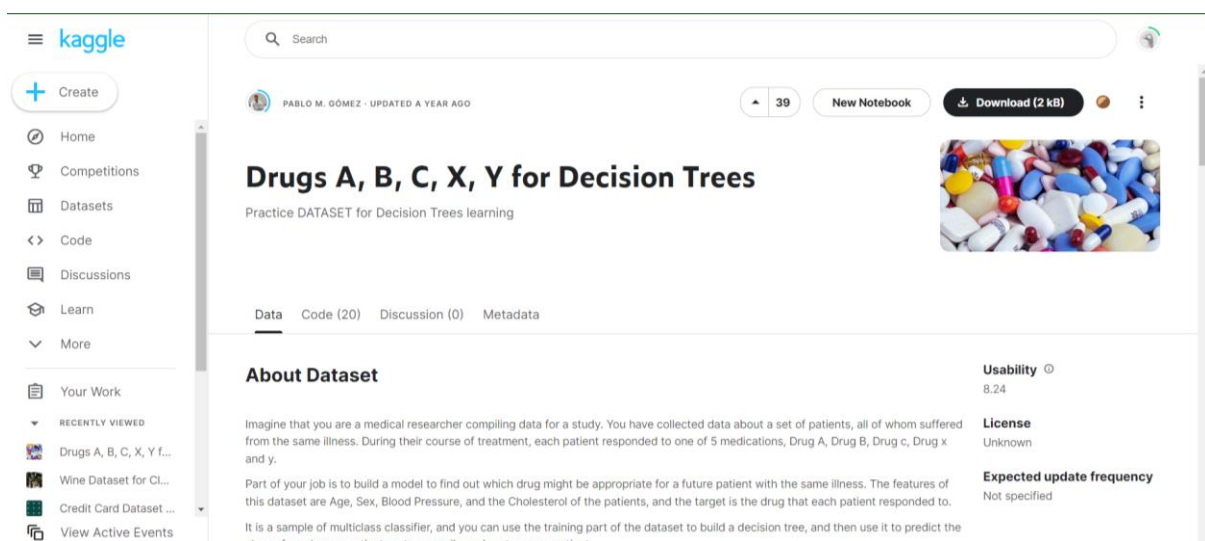
**Date 15/9/22**

### AIM:

The aim of the experiment was to implement Navie Bayes Classification Model in weka, java using weka and in python using machine learning libraries

### DOWNLOADING DATASET:

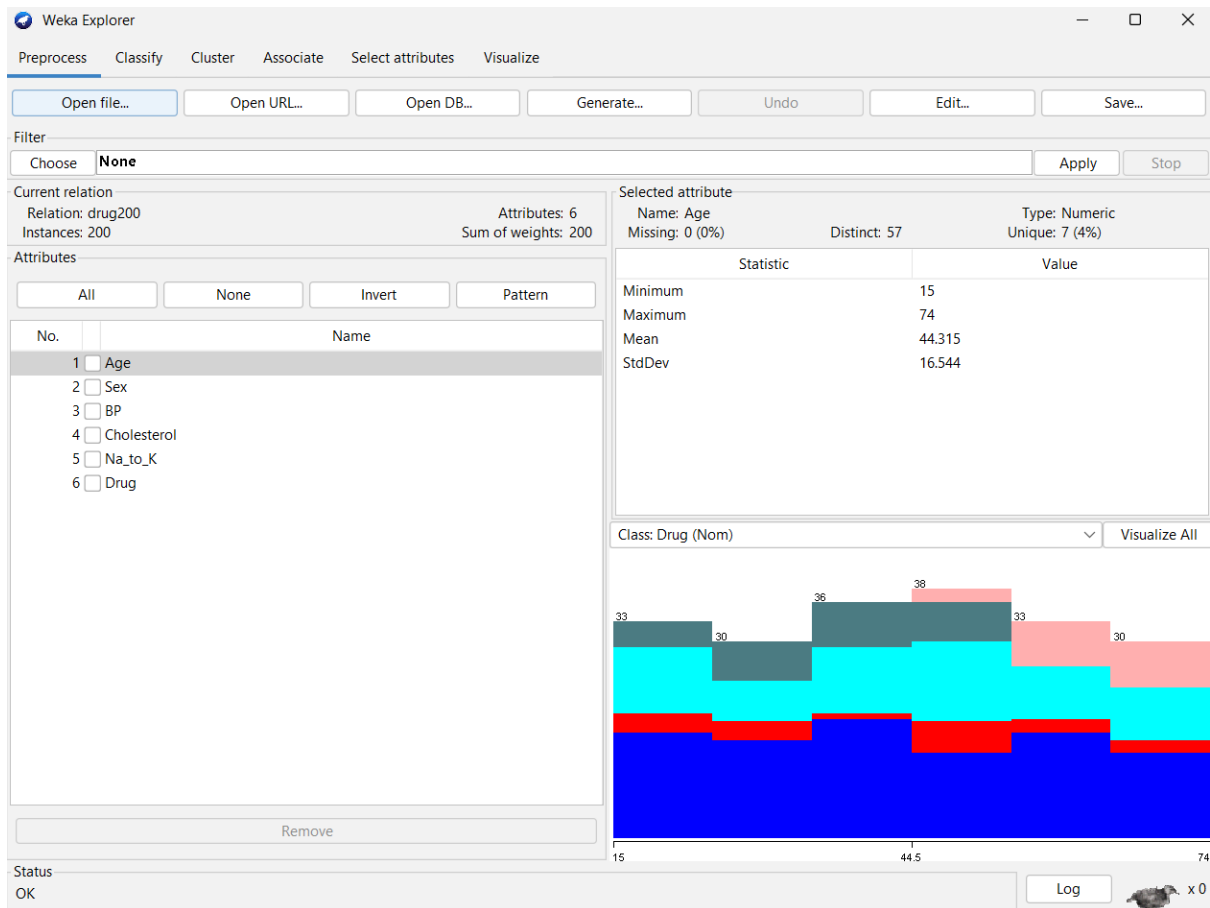
1. Go to Kaggle ([www.kaggle.com](https://www.kaggle.com)) sign in or register new account.
2. After that download the data set you like.



3. Click download button to download the dataset.

### NAÏVE BAYES IN WEKA:

1. Open weka and click explore.
2. In preprocessing tab click open file button.
3. Select the dataset file you going to perform decision tree.
4. Then click classify tab.
5. Select the classifier by clicking the choose button.
6. Select the NaïveBayes which is present under Bayes.
7. Finally click start. Then it automatically built the model



```
=== Run information ===

Scheme:      weka.classifiers.bayes.NaiveBayes
Relation:    drug200
Instances:   200
Attributes:  6
             Age
             Sex
             BP
             Cholesterol
             Na_to_K
             Drug
Test mode:   10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Attribute    Class
            drugY  drugC  drugX  drugA  drugB
            (0.45) (0.08) (0.27) (0.12) (0.08)
=====
Age
mean        43.7637 42.6696 44.0159 35.913 62.4241
std. dev.    16.9209 16.2024 16.2836  9.5704  6.9338
weight sum   91      16      54      23      16
precision    1.0536  1.0536  1.0536  1.0536  1.0536

Sex
F           48.0    8.0    28.0   10.0    7.0
M           45.0   10.0   28.0   15.0   11.0
[total]     93.0   18.0   56.0   25.0   18.0

Correctly Classified Instances      193      96.5
Incorrectly Classified Instances     7       3.5
Kappa statistic                     0.9491
Mean absolute error                  0.0689
Root mean squared error              0.1437
Relative absolute error              24.6937 %
Root relative squared error          38.5477 %
Total Number of Instances           200

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measu
                1.000    0.028    0.968      1.000    0.984
                0.813    0.000    1.000    0.813    0.897
                0.981    0.014    0.964    0.981    0.972
                0.913    0.006    0.955    0.913    0.933
                0.938    0.005    0.938    0.938    0.938
Weighted Avg.   0.965    0.017    0.965    0.965    0.964

=== Confusion Matrix ===

  a  b  c  d  e  <-- classified as
91  0  0  0  0  |  a = drugY
```

## NAIVEBAYES USING JAVA

### Code:

```
package Javatree;
import weka.core.Instances;
import weka.classifiers.bayes.NaiveBayes;
import weka.core.converters.CSVLoader;
import java.io.File;
import weka.classifiers.evaluation.Evaluation;
import java.util.Random;
public class Main {
    public static void main(String[] argv) {
        try {
            CSVLoader loader = new CSVLoader();
            String name = "C:\\Users\\kaush\\Desktop\\drug200.csv";
            loader.setSource(new File(name));
            //
            BufferedReader dataset = new BufferedReader(new
            FileReader(name));

            Instances drug = loader.getDataSet();
            System.out.println(drug);
            NaiveBayes tree = new NaiveBayes();
            drug.setClassIndex(
                drug.numAttributes() - 1);
            tree.buildClassifier(drug);
            System.out.println(tree);
            Evaluation evaluation= new Evaluation(drug);
            evaluation.crossValidateModel(tree, drug, 10,new Random(1));
            System.out.println(evaluation.toSummaryString("\nResults", false));
            System.out.println(evaluation.toMatrixString());
        }
        catch(Exception e) {}
    }
}
```

### Output:

Naive Bayes Classifier					
Attribute	Class	drugC	drugX	drugA	drugB
	(0.45)	(0.08)	(0.27)	(0.12)	(0.08)
=====					
Age					
mean	43.7637	42.6696	44.0159	35.913	62.4241
std. dev.	16.9209	16.2024	16.2836	9.5704	6.9338
weight sum	91	16	54	23	16
precision	1.0536	1.0536	1.0536	1.0536	1.0536
Sex					
F	48.0	8.0	28.0	10.0	7.0
M	45.0	10.0	28.0	15.0	11.0
[total]	93.0	18.0	56.0	25.0	18.0
BP					
HIGH	39.0	1.0	1.0	24.0	17.0
LOW	31.0	17.0	19.0	1.0	1.0
NORMAL	24.0	1.0	37.0	1.0	1.0
[total]	94.0	19.0	57.0	26.0	19.0
Cholesterol					
HIGH	48.0	17.0	21.0	13.0	9.0
NORMAL	45.0	1.0	35.0	12.0	9.0
[total]	93.0	18.0	56.0	25.0	18.0

Na_to_K					
mean	22.3687	10.6221	10.6563	10.9111	11.5454
std. dev.	6.0343	1.9756	2.1855	1.9903	1.812
weight sum	91	16	54	23	16
precision	0.1623	0.1623	0.1623	0.1623	0.1623
Results					
Correctly Classified Instances	193			96.5	%
Incorrectly Classified Instances	7			3.5	%
Kappa statistic	0.9491				
Mean absolute error	0.0689				
Root mean squared error	0.1437				
Relative absolute error	24.6937			%	
Root relative squared error	38.5477			%	
Total Number of Instances	200				
=== Confusion Matrix ===					
a	b	c	d	e	<-- classified as
91	0	0	0	0	a = drugY
1	13	2	0	0	b = drugC
1	0	53	0	0	c = drugX
1	0	0	21	1	d = drugA
0	0	0	1	15	e = drugB

### Result and Confusion Matrix

## NAÏVE BAYES USING PYTHON

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
from sklearn.naive_bayes import GaussianNB
# https://www.kaggle.com/datasets/pablongomez21/drugs-a-b-c-x-y-for-decision-trees
drug = pd.read_csv("drug200.csv")
columnName = drug.columns.tolist()
columnName.pop(0)
columnName.pop(-1)
columnName.pop(-1)
inverseLe = {}
for i in columnName:
    le = preprocessing.LabelEncoder()
    le.fit(drug[i].values)
    a = le.transform(drug[i].values)
    drug[i] = a
    inverseLe[i] = le
x = drug.values[:, 0:5]
y = drug.values[:, 5]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=100)
gnb = GaussianNB()
gnb.fit(x_train, y_train)
y_pred = gnb.predict(x_test)
print("accuracy = ", accuracy_score(y_test, y_pred)*100)
confusion_matrix(y_test, y_pred)
```

### Output:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
..	...	..	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

[200 rows x 6 columns]

Before preprocessing

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	0	0	25.355	drugY
1	47	1	1	0	13.093	drugC
2	47	1	1	0	10.114	drugC
3	28	0	2	0	7.798	drugX
4	61	0	1	0	18.043	drugY
..	...	...	..	...	...	...
195	56	0	1	0	11.567	drugC
196	16	1	1	0	12.006	drugC
197	52	1	2	0	9.894	drugX
198	23	1	2	1	14.020	drugX
199	40	0	1	1	11.349	drugX

[200 rows x 6 columns]

After preprocessing

```

1
2 x = drug.values[:, 0:5]
3 y = drug.values[:, 5]
4 x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=100)
5 gnb = GaussianNB()
6 gnb.fit(x_train, y_train)
7 y_pred = gnb.predict(x_test)
8 print("accuracy = ", accuracy_score(y_test, y_pred)*100)
9 confusion_matrix(y_test, y_pred)
10

```

```

↳ accuracy = 85.0
array([[ 2,  0,  0,  0,  0],
       [ 0,  5,  0,  0,  0],
       [ 0,  0,  2,  0,  0],
       [ 0,  0,  0,  8,  0],
       [ 5,  1,  0,  0, 17]])

```

### Confusion Matrix

## RESULT:

Successfully we implemented Naïve Bayes classification algorithm in weka gui, java with weka library and using python.