Name: Allan Rodrigues
Class: TE IT A
Roll no: 59
Pid:191104

St. Francis Institute of Technology, Mumbai-400 103

**Department Of Information Technology**

**A.Y. 2021-2022**

**Class: TE-ITA/B, Semester: VI**

**Subject: Business Intelligence Lab**

**Experiment – 5:** To implement any one classifier (Decision tree/Naïve Bayes) using any one Language (JAVA/R/Python)

1. **Aim: To implement any one classifier (Decision tree/Naïve Bayes) using any one Language (JAVA/R/Python)**

2. **Objectives: After study of this experiment, the students will be able to Implement Naïve based algorithm**

3. **Outcomes:** After study of this experiment, the students will be able to
   **CO 3:** Design and Implement various classification data mining techniques such as Decision tree, Naïve Bayes Regression etc. In addition, apply metrics to measure its performance

4. **Prerequisite: Introduction to all the three classifiers through algorithms & Problem solving approach.**

5. **Requirements: Personal Computer, Windows XP operating system/Windows 7, Internet Connection, Microsoft Word, WEKA tool, Java/R/Python**

6. **Theory:**
   a. Explain the Classification Algorithm (Decision tree/Naïve Bayes) with example
   b. Applications of Classification Algorithms (Decision tree/Naïve Bayes)
   c. Advantages and Disadvantages of Classification Algorithms (Decision tree/Naïve Bayes)

Allan Rodrigues    T.E 17 A-59                  , Rajdhani

DATE  /  /

**6  a)** Naïve Bayes ~~discuss the~~ theory classifiers are a collection of classification algorithms based on Bayes's theorem. It is a family of algorithm. lets see on example "Buys_computer = Yes". Use the following data Classify x = (Age = "<=30", Income = "Medium", student = "Yes", credit rating = "fair").

| Id | Age | Income | Student | credit_rating | Buys-computer |
|----|-----|--------|---------|---------------|---------------|
| 1 | <=30 | High | NO | fair | NO |
| 2 | <=30 | High | NO | Excellent | NO |
| 3 | 31-40 | High | NO | fair | Yes |
| 4 | >40 | Medium | NO | fair | Yes |
| 5 | >40 | Low | Yes | fair | Yes |
| 6 | >40 | Low | Yes | Excellent | NO |
| 7 | 31-40 | Low | Yes | Excellent | Yes |
| 8 | <=30 | Medium | NO | fair | NO |
| 9 | <=30 | Low | Yes | fair | Yes |
| 10 | >40 | medium | yes | fair | yes |
| 11 | <=30 | Medium | yes | Excellent | Yes |
| 12 | 31-40 | Medium | NO | Excellent | Yes |
| 13 | 31-40 | High | Yes | fair | Yes |
| 14 | >40 | Medium | NO | Excellent | NO |

$Sol^n$

Class label is Buys_computer

$C_1$ : Buys_computer = Yes = 9

$C_2$ : Buys_computer = NO = 5

$P(C_1) = \dfrac{9}{14}$     and     $P(C_2) = \dfrac{5}{14}$

Allan Rodrigues    TE IT A-59

let $x_1$ be Age = "<=30",
event $x_2$ be Income = "Medium"
    $x_3$ be student = "Yes"
and  $x_4$ be credit rating = "fair"

To compute $P(x/C_i)$

From naive Babyesian classification

$$P(x/C_i) = \prod_{k=1}^{m} P(x_k/C_i)$$

$$P(x/C_i) = P(x_1/C_i)\, P(x_2/C_i)\, P(x_3/C_i)\, P(x_4/C_i)$$

$$P(x_1/C_i) = P\left(\frac{Age <= 30}{Buys\_computer=Yes}\right) = \frac{2}{9}$$

$$P(x_2/C_i) = P\left(\frac{Income = "medium"}{Buys\_computer=Yes}\right) = \frac{4}{9}$$

$$P(x_3/C_i) = P\left(\frac{Student = "Yes"}{Buys-computer=Yes}\right) = \frac{6}{9}$$

$$P(x_4/C_i) = P\left(\frac{Credit\ rating = Yes}{Boys\_computer=Yes}\right) = \frac{6}{9}$$

from naive bayesion classification
$$P(x/C_i) = \frac{2}{9} \times \frac{4}{9} \times \frac{6}{9} \times \frac{6}{9} = 0.044$$

$$P(x/C_i)\, P(C_i) = 0.044 \times \frac{9}{14} = 0.028 \quad —①$$

Allan Rodrigues TE IT A-59     Rajdhani
DATE / /

From ①, ② & ③
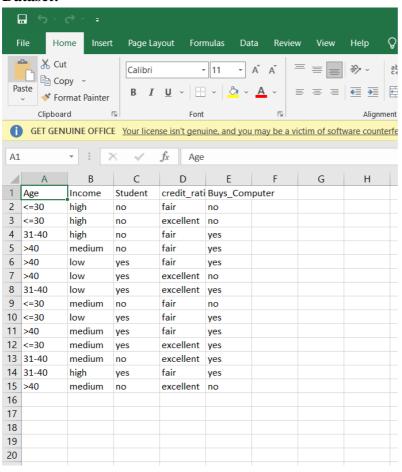
$$P(x|C_1) P(C_1) > P(x|C_2) P(C_2)$$

∴ We conclude that the unknown sample.

$X = ($Age $= "<=30",$ Income $= "Medium",$ Student $= "Yes",$ Credit-rating $= "fair")$

belongs to class.

$C_1 = $ Buys - computer $= Yes.$

b) Applications of naive bayes are

1) As the algorithm is fast and efficient, you can use it to make real-time predictions.

2) This algorithm is popular for multi-class predictions.

3) Email Services use this algorithm to figure out whether an email is a spam or not.

4) Its assumption of feature independence and its effectiveness in solving multi-class problems, makes it perfect for performing sentiment analysis.

5) Collaborative filtering & the naive Bayes algorithm work together to build recommendation systems. These systems use Data mining and machine learning, to predict if the user would like a particular resource or not.

Allan Rodrigues TE IT A-59

c) Advantages of naive bayes.
1) This algorithm, works very fast and can easily predict the class of a test dataset.
2) You can use it to solve multi-class prediction problems as it's quite useful with them.
3) Naive Bayes classifier performs better than other models with less training data if the assumption of independence of features holds
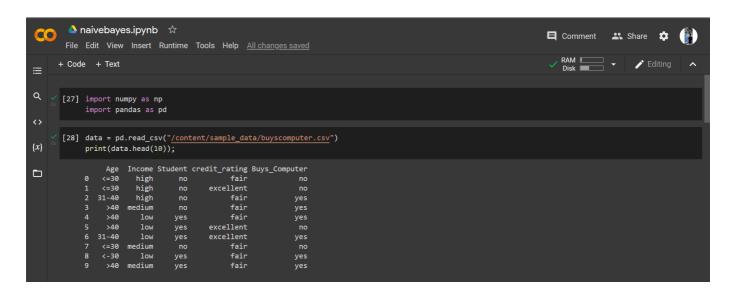
Disadvantages of naive bayes
1) This algorithm is notorious as a lousy estimator. So, you should't take the probability outputs of 'predict_proba' too seriously.
2) It assumes that all the independent. While it might sound great in theory, in real life, you'll hardly find a set of independent features.

---

**7. Laboratory Exercise: Implementation of Classification Algorithm using JAVA/ R/ Python. Printout of implementation along with coding and Output.**

**Dataset:**

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Age | Income | Student | credit_rati | Buys_Computer | | | |
| 2 | <=30 | high | no | fair | no | | | |
| 3 | <=30 | high | no | excellent | no | | | |
| 4 | 31-40 | high | no | fair | yes | | | |
| 5 | >40 | medium | no | fair | yes | | | |
| 6 | >40 | low | yes | fair | yes | | | |
| 7 | >40 | low | yes | excellent | no | | | |
| 8 | 31-40 | low | yes | excellent | yes | | | |
| 9 | <=30 | medium | no | fair | no | | | |
| 10 | <=30 | low | yes | fair | yes | | | |
| 11 | >40 | medium | yes | fair | yes | | | |
| 12 | <=30 | medium | yes | excellent | yes | | | |
| 13 | 31-40 | medium | no | excellent | yes | | | |
| 14 | 31-40 | high | yes | fair | yes | | | |
| 15 | >40 | medium | no | excellent | no | | | |
| 16 | | | | | | | | |
| 17 | | | | | | | | |
| 18 | | | | | | | | |
| 19 | | | | | | | | |
| 20 | | | | | | | | |

Code:

```python
import numpy as np
import pandas as pd
```

```python
data = pd.read_csv("/content/sample_data/buyscomputer.csv")
print(data.head(10));
```

```
     Age  Income Student credit_rating Buys_Computer
0   <=30    high      no          fair            no
1   <=30    high      no     excellent            no
2  31-40    high      no          fair           yes
3    >40  medium      no          fair           yes
4    >40     low     yes          fair           yes
5    >40     low     yes     excellent            no
6  31-40     low     yes     excellent           yes
7   <=30  medium      no          fair            no
8   <-30     low     yes          fair           yes
9    >40  medium     yes          fair           yes
```

CO 🔺 naivebayes.ipynb ☆

File Edit View Insert Runtime Tools Help   All changes saved

💬 Comment   👥 Share ⚙ 👤

+ Code  + Text

RAM ▭ ▾   ✏ Editing  ⌃
Disk ▭

```python
[30] prior = data.groupby('Buys_Computer').size().div(len(data)) #count()['Age']/len(data)
     print(prior);
```

```
Buys_Computer
no     0.357143
yes    0.642857
dtype: float64
```

Calculating likelihood

P(Age|Buyscomputer)

P(Income|Buyscomputer)

P(Student|Buyscomputer)

P(Creditrating|Buyscomputer)

---

CO 🔺 naivebayes.ipynb ☆

File Edit View Insert Runtime Tools Help   All changes saved

💬 Comment   👥 Share ⚙ 👤

+ Code  + Text

RAM ▭ ▾   ✏ Editing  ⌃
Disk ▭

```python
[34] likelihood = {}
     likelihood['credit_rating'] = data.groupby(['Buys_Computer', 'credit_rating']).size().div(len(data)).div(prior)
     likelihood['Age'] = data.groupby(['Buys_Computer', 'Age']).size().div(len(data)).div(prior)
     likelihood['Income'] = data.groupby(['Buys_Computer', 'Income']).size().div(len(data)).div(prior)
     likelihood['Student'] = data.groupby(['Buys_Computer', 'Student']).size().div(len(data)).div(prior)

     print (likelihood)
```

```
{'credit_rating': Buys_Computer  credit_rating
no             excellent      0.600000
               fair           0.400000
yes            excellent      0.333333
               fair           0.666667
dtype: float64, 'Age': Buys_Computer  Age
no             <=30      0.600000
               >40       0.400000
yes            31-40     0.444444
               <=30      0.222222
               >40       0.333333
dtype: float64, 'Income': Buys_Computer  Income
no             high      0.400000
               low       0.200000
               medium    0.400000
yes            high      0.222222
               low       0.333333
               medium    0.444444
```

---

CO 🔺 naivebayes.ipynb ☆

File Edit View Insert Runtime Tools Help   All changes saved

💬 Comment   👥 Share ⚙ 👤

+ Code  + Text

RAM ▭ ▾   ✏ Editing  ⌃
Disk ▭

```
[34]            >40       0.400000
yes            31-40     0.444444
               <=30      0.222222
               >40       0.333333
dtype: float64, 'Income': Buys_Computer  Income
no             high      0.400000
               low       0.200000
               medium    0.400000
yes            high      0.222222
               low       0.333333
               medium    0.444444
dtype: float64, 'Student': Buys_Computer  Student
no             no        0.800000
               yes       0.200000
yes            no        0.333333
               yes       0.666667
dtype: float64}
```

**Calculating posterior**

We need to if a person wil buy computer based on the following new information

{"Age":'<=30', "Income":"medium", "Student":'yes' , "Credit_Rating":'fair'}

Substituing the values in the likehood data and in the bayes formula, we get

```
[35] # Probability that the person will buy
     p_yes = likelihood['Age']['yes']['<=30'] * likelihood['Income']['yes']['medium'] *\
     likelihood['Student']['yes']['yes'] * likelihood['credit_rating']['yes']['fair'] \
             * prior['yes']

     # Probability that the person will NOT buy
     p_no = likelihood['Age']['no']['<=30'] * likelihood['Income']['no']['medium'] *\
     likelihood['Student']['no']['yes'] * likelihood['credit_rating']['no']['fair'] \
             * prior['no']

     print ('Yes : ', p_yes)
     print ('No :  ', p_no)

     Yes :  0.028218694885361544
     No :   0.006857142857142855
```



```
[36] data['Buys_Computer'].hist()

     <matplotlib.axes._subplots.AxesSubplot at 0x7f7ea90aef90>
```
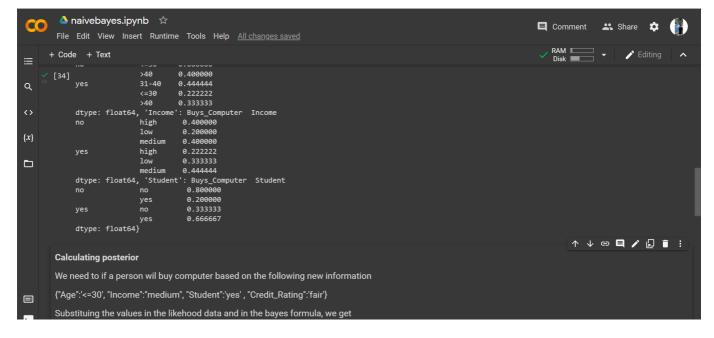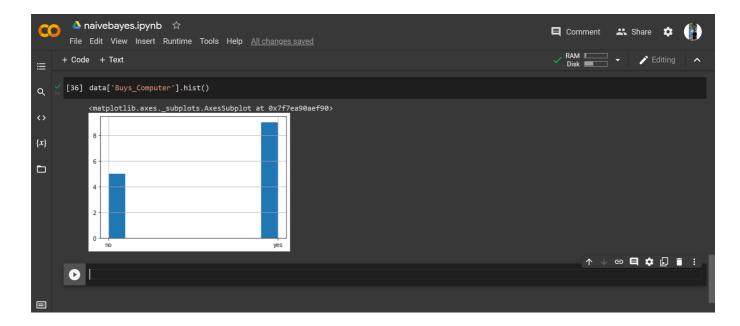
## 8. Post-Experiments Exercise

### a. Questions:

● **Compare and Contrast between Decision Tree & Naïve Bayes**

Allan Rodrigues TE IT A-59    Rajdhani

DATE / /

Q.8 a) Decision Trees are very flexible, easy to understand and easy to debug. They will work with classification problems and regression problems. Probably one of the coolest thing about decision tree is they only need a table of data and they will build classifier directly from that data.

Naive Bayes requires you build a classification by hand. There's not way to just toss a bunch of tabular data at it and have it pick the best features it will use to classify. Naive bayes does quite well when the training data doesn't contain all possibilities so it can be very good with low amounts of data. Decision trees work better with lots of data compared to naive Bayes.

b. **Conclusion:**
   - **Summary of Experiment**
   - **Importance of Experiment**
   - **Application of Experiment**

Allan Rodrigues   TE IT A-59   Rajdhani
DATE / /

b) ⑥ Conclusion

In this experiment we learnt about decision tree and naïve bayes algorithm we solved naïve bayes algorithm and implemented the same problem statement in python and got similar results.

Naïve bayes is suitable for solving multi-class prediction problems. If its assumption of the independence of features holds true, it can perform better than other models and requires much less training data. It is highly scalable with the number of predictors & data points.

○Applications of naïve bayes are

We can use naïve bayes in Face recognition, as a classifier it is used to identify the faces or its other features like nose, mouth, eyes etc. It can be used to predict if the weather will be good or bad. With the help of Naïve bayes classifier, Google news recognizes wheather the news news is political, world news and so on.

9. **Reference: Data Mining: Concept & Techniques, 3rd Edition, Jiawei Han, Micheline Kamber, Jian Pei, Elsevier.**
   **Reference links:**

   - **https://scikit-learn.org/stable/modules/naive_bayes.html**
   - **https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn**
   - 
     **https://www.analyticsvidhya.com/blog/2021/11/implementation-of-gaussian-naive-bayes-in-python-sklearn/**
   - 
     **https://github.com/2796gaurav/Naive-bayes-explained/blob/master/Naive%20bayes/Naive%20Bayes%20in%20scikit%20learn.ipynb**

     - **https://www.upgrad.com/blog/naive-bayes-classifier/**

     **https://stackoverflow.com/questions/10317885/decision-tree-vs-naive-bayes-classifier**
   -