EXPERIMENT NO 4

NAME: Vishal Shashikant Salvi **UID: 2019230069**

Class: TE Comps **BATCH:C**

Aim: To implement Naïve Bayes Algorithm.

Theory:

Naïve Bayes Classifier Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem

Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

Problem: If the weather is sunny, then the Player should play or not?

Solution: To solve this, first consider the below dataset:

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes

6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Applying Bayes'theorem:

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)

P(Sunny|Yes)= 3/10= 0.3

P(Sunny)= 0.35

P(Yes)=0.71

So P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**

P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)

P(Sunny|NO)= 2/4=0.5

P(No)= 0.29

P(Sunny)= 0.35

So P(No|Sunny)= 0.5*0.29/0.35 = **0.41**

So as we can see from the above calculation that **P(Yes|Sunny)>P(No|Sunny)**

Hence on a Sunny day, Player can play the game.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

Code:

```
import collections
import numpy as np
import pandas as pd
import matplotlib as plt
from collections import Counter

dataset = pd.read_csv('comp.csv')
print("Dataset is given below")
print(dataset)
print("\n")
columns = dataset.columns
print("All columns are", columns)
print("\n")
dependent = list(collections.Counter(dataset[columns[-1]]).items())
probability = [(x[0], x[1] / dataset.shape[0]) for x in dependent]

print("Dependent variables are ", dependent)
print("\n")
print("Probabilities are ", probability)
print("\n")
inputval = {'Age': '<=30', 'Income': 'High', 'Student': 'No',
'Credit_Rating': 'Fair','Computer_Buy': 'NO'}
print("Input values are ", inputval)
print("\n")
eachclass = []
print("Classes are")
for column in columns[:-1]:
    print(column)
    for item in dependent:
        val = sum(((dataset[column] == inputval[column]) & (dataset.values
== item[0])).astype(int))
        eachclass.append((item[0], column, val / item[1]))
ans = {'Yes': 1, 'No': 1}
for item in eachclass:
    if item[0] == 'No':
        ans['No'] *= item[2]
    else:
        ans['Yes'] *= item[2]
for item in probability:
    ans[item[0]] *= item[1]
print("\n")
print("Final answer ", ans)
```

Output:

```
C:\Users\Vishal\AppData\Local\Programs\Python\Python38\python.exe C:/Python/naive.py
Dataset is given below
       Age Income  Student Credit_rating Computer_Buy
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
10    <=30  Medium     Yes     Excellent          Yes
11   31..40  Medium      No     Excellent          Yes
12   31..40    High     Yes          Fair          Yes
13     >40  Medium      No     Excellent           No


All columns are Index(['Age', 'Income ', 'Student', 'Credit_rating', 'Computer_Buy'], dtype='object')


Dependent variables are  [('No', 5), ('Yes', 9)]


Probabilities are  [('No', 0.35714285714285715), ('Yes', 0.6428571428571429)]


Input values are  {'Age': '<=30', 'Income': 'High', 'Student': 'No', 'Credit_Rating': 'Fair', 'Computer_Buy': 'NO'}
```

```
Classes are
Age
Income
Student
Credit_Rating
Computer Buy

Final answer is{‘Yes’: 0.028218694885361547, ‘NO’: 0.006857142857142857142858}
```

Conclusion:

Thus from this experiment I learn about naïve base classifier algorithm. Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.