Birla Institute of Technology and Science Pilani, Hyderabad Campus

2nd Semester 2023-24, BITS F464: Machine Learning

Assignment No: 4 (**Gaussian Naïve Bayes Generative Model**)

Date Given: 26.03.2024 Max. Marks: 5 Submission date: **05.04.2024**

The Naive Bayes Generative Classifier is a widely-used algorithm in machine learning, it operates on the principles of Bayes' theorem and assumes independence among features, allowing it to make predictions quickly and with minimal computational resources. It predicts the class for a given instance based on the class probabilities computed using Bayes' theorem. After calculating the posterior probability of each class given the instance's features, the algorithm selects the class with the highest probability as the predicted class for that instance. In other words, it assigns the class that maximizes the posterior probability. As we discussed in the class, the maximum a posteriori (MAP) that selects the best hypothesis for Naïve Bayes classifier is as given below:

$$\text{Maximum A Posteriori} = \underset{h \in H}{\operatorname{argmax}}\, P(h) \prod_{i=1}^{n} P(x_i|h) \qquad \text{Equation (1)}$$

In Gaussian Naive Bayes, the likelihood $P(x_i|y)$ is often modelled using a Gaussian (normal) distribution. The probability density function (PDF) of a Gaussian distribution is:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \qquad \text{Equation (2)}$$

In practice, the parameters μ and σ^2 are estimated from the training data for each feature x_i and each class y. Then, during classification, these parameters are used to compute the likelihood $P(x_i|y)$ for each feature given each class.

Your task in this assignment is to experiment with Gaussian NaiveBayes algorithm for the grading file attached here (Data-NB.xlsx). Grading is based on the test scores. Below are the code snippets in Scikit learn to import the classifier and other required libraries:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
```

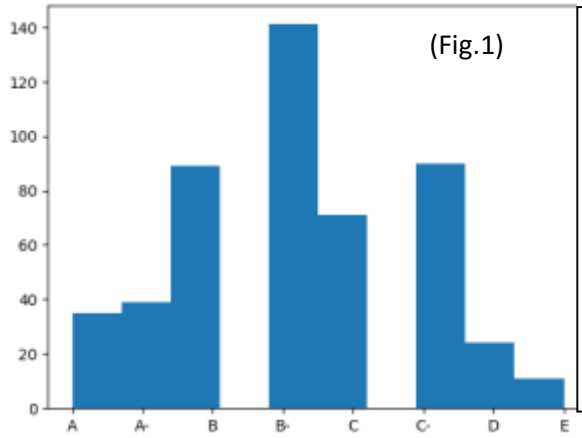
To get first few records of the Pandas DataFrame, use the following:

```
df.head()
```

The output in your notebook should be as below:

	Lab-Test1(30)	Lab-Test2(24)	Midsem Test (90)	Gender	Attendance	Grade
0	13.00	24	66.0	Male	High	A
1	15.00	24	67.0	Female	High	A
2	5.25	24	45.0	Male	High	B-
3	2.75	19	34.0	Male	High	C-
4	7.25	24	30.0	Male	High	C-

The data distribution of the given xlsx file (Grading) is Gaussian as discussed in the class. Plot the below pattern (Fig.1) to visualize it in your Python code. The Gaussian Naïve Bayes Classifier's performance metric for an 80-20 rule is as shown below (Fig.2).



(Fig.1)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2
1	1.00	1.00	1.00	4
2	0.89	0.94	0.92	18
3	0.92	0.64	0.75	36
4	0.37	0.78	0.50	9
5	0.91	0.95	0.93	22
6	0.86	0.75	0.80	8
7	1.00	1.00	1.00	1
accuracy			0.81	100
macro avg	0.87	0.88	0.86	100
weighted avg	0.86	0.81	0.82	100

(Fig.2)

The second part of this assignment is to classify flowers using iris.csv data file that is also attached with this assignment using GaussianNB. There are 150 records in this file, and plot the flowers using matplotlib.imshow method to view the flowers as shown below:

Each record has features as Sepal length, Sepal width, Petal length, Petal width, and Species (Categorical feature: Setosa, Versicolor, and Virginica). You may also import the in-built iris dataset from sklearn learn as below:

```
from sklearn import datasets
# loading IRIS dataset from sklearn
iris = datasets.load_iris()
```

The classification report is as given below with a prediction accuracy of 97% and other related metrics.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.90	1.00	0.95	9
2	1.00	0.91	0.95	11
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

Complete the following tasks:

- In the Data-NB.xlsx file, few attributes like Gender, Attendance and Grade columns are nominal variables and require encoding before the model training. Use appropriate encoding.
- Split the dataset into training and testing subsets (80-20 or 70-30). Train a Gaussian Naive Bayes classifier on the training data and predict the grades in the test data. Calculate the accuracy and the confusion matrix to assess the classifier's performance.
- Split the Iris dataset (iris.csv) into training and testing subsets, followed by training a Gaussian Naive Bayes classifier on the training data. Evaluate the classifier's performance by plotting the classification report as shown above.
- Compare and contrast the performance of the Naïve Bayes classifier built in this assignment with that of Random Forest and Gradient Boosted Trees (developed in Assignment 2) on the identical datasets. Analyse the reasons behind any observed differences in their performances.
- For both the datasets visualize the correlation matrix to check and verify the assumptions of Naïve Bayes algorithm.

Submission Instructions: Same as that of earlier assignments. Any clarification on this coding assignment may be emailed to I/C or Paryetri Banerjee (f20202001@hyderabad.bits-pilani.ac.in) or Anish Shandilya (f20210982@hyderabad.bits-pilani.ac.in).

References: 1. https://scikit-learn.org/stable/modules/naive_bayes.html

2. https://towardsdatascience.com/the-naive-bayes-classifier-how-it-works-e229e7970b8