**ML – 1 LAB – 03**

**Date :30-05-22**

**By : 21BDA10**

**Aishwarya c s**

**Question 1 :**

**Write the difference between the following:**

**i. Gaussian Naive Bayes,**

**ii. Multinomial Naive Bayes,**

**iii. Complement Naive Bayes,**

**iv. Bernoulli Naive Bayes,**

**v. Categorical Naive Bayes,**

**vi. Out-of-core naive Bayes model fitting.**

1. **Gaussian Naïve Bayes**

Because of the assumption of the normal distribution, Gaussian Naive Bayes is used in cases when all our features are continuous. For example in the Iris dataset features are sepal width, petal width, sepal length, and petal length. So its features can have different values in the data set as width and length can vary. We can’t represent features in terms of their occurrences. This means data is continuous. Hence we use Gaussian Naive Bayes here.

Gaussian Naïve Bayes supports continuous-valued features and models each conforming to a gaussian (normal) distribution.

An approach to creating a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independence dimensions) between dimensions. This model can be fit by simply finding the mean and standard deviation of the points within each label, which is all that is needed to define such a distribution.

Resource:

[www.quora.com](http://www.quora.com)

<https://iq.opengenus.org/gaussian-naive-bayes/#:~:text=Gaussian%20Naive%20Bayes%20supports%20continuous,(independent%20dimensions)%20between%20dimensions>.

1. **Multinomial Naïve Bayes**

The multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output. The Naive Bayes classifier is made up of a number of algorithms that all have one thing in common: each feature being classed is unrelated to any other feature. A feature's existence or absence has no bearing on the inclusion or exclusion of another feature.

The Naive Bayes method is a strong tool for analyzing text input and solving problems with numerous classes. Because the Naive Bayes theorem is based on the Bayes theorem, it is necessary to first comprehend the Bayes theorem notion.

Resource:<https://www.upgrad.com/blog/multinomial-naive-bayes-explained/#:~:text=The%20Multinomial%20Naive%20Bayes%20algorithm%20is%20a%20Bayesian%20learning%20approach,tag%20with%20the%20greatest%20chance>.

1. **Complement Naïve Bayes**

Complement Naïve Bayes is somewhat an adaptation of the standard multinomial naïve bayes algorithm. Multinomial naïve Bayes does not perform very well on imbalanced datasets. Imbalanced datasets are datasets where the number of examples of some class is higher than the number of examples belonging to other classes. This means that the distribution of examples is not uniform.

In complement naïve Bayes, instead of calculating the probability of an item belonging to a certain class, we calculate the probability of the item belonging to all the classes.

Resource:

<https://www.geeksforgeeks.org/complement-naive-bayes-cnb-algorithm/#:~:text=In%20complement%20Naive%20Bayes%2C%20instead,is%20called%20Complement%20Naive%20Bayes>.

1. **Bernoulli Naïve Bayes**

This is used for discrete data and it works on Bernoulli distribution. The main feature of Bernoulli’s naïve Bayes is that it accepts features only as binary values like true or false, yes or no, success or failure, 0 or1, and so on. So when the feature values are binary we know that we have to use Bernoulli naïve Bayes classifier.

Resource:

<https://iq.opengenus.org/bernoulli-naive-bayes/>

1. **Categorical Naïve Bayes**

The categorical naïve Bayes classifier is suitable for classification with discrete features that are categorically distributed. The categories of each feature are drawn from a categorical distribution.

The features should to encoded using label encoding techniques such that each category would be mapped to a unique number.

Resource;

<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html#:~:text=The%20categorical%20Naive%20Bayes%20classifier,drawn%20from%20a%20categorical%20distribution>.

<https://blog.ineuron.ai/Categorical-Naive-Bayes-Classifier-implementation-in-Python-dAVqLWkf7E>

1. **Out - of - Core Naïve Bayes Model**

Naive Bayes models can be used to tackle large-scale classification problems for which the full training set might not fit in memory. To handle this case, multinomial naïve Bayes, Bernoulli naïve Bayes, and Gaussian naïve Bayes expose a partial fit method that can be used incrementally as done with other classifiers as demonstrated in out of the core classification of text documents. All naïve Bayes classifiers support sample weighting.

Resource:

<https://scikit-learn.org/stable/modules/naive_bayes.html>

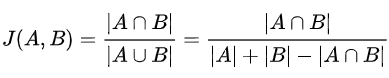
**Question 3:**

**What is Jaccard and cosine similarity?**

**Jaccard Similarity:**

Jaccard similarity index is also called as Jaccard similarity coefficient. It measures the similarity between two sets. The range is 0 to 100%. The more percentage the more similar the two words are.

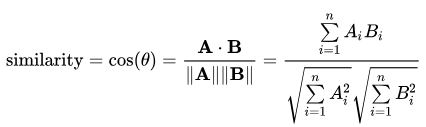
Formula:



**Cosine Similarity:**

The cosine similarity measures the cosine angle between the two vectors. For cosine, have to convert all sentences to vectors. For converting to vector we can use TF-IDF, Word2Vec.

Formula:



Jaccard Similarity takes a set of the unique length of words instead cosine similarity takes a whole sentence vector

If data duplication is not mattered then it’s better to use Jaccard similarity else cosine similarity is good for measuring the similarity between two vectors even if the data duplication is there.

Resource :

<https://medium.com/analytics-vidhya/introduction-to-similarity-metrics-a882361c9be4>