**ML1-LAB03**

NIKHITHA ELEZEBETH BABY

21BDA14

**1.Write the difference between the following:**

**i.Gaussian Naive Bayes**

Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution. An approach to create a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independent 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 what is needed to define such a distribution.

Resource:

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

**ii.Multinomial Naive Bayes**

The Multinomial Naive Bayes algorithm is a Bayesian learning approach popular in Natural Language Processing (NLP). The program guesses the tag of a text, such as an email or a newspaper story, using the Bayes theorem. It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance. 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. The Bayes theorem, which was developed by Thomas Bayes, estimates the likelihood of occurrence based on prior knowledge of the event's conditions. When predictor B itself is available, we calculate the likelihood of class A. It's based on the formula below: P(A|B) = P(A) \* P(B|A)/P(B).

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>

**iii.Complement Naive Bayes**

Complement Naive Bayes is somewhat an adaptation of the standard Multinomial Naive Bayes algorithm. Multinomial Naive 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. This type of dataset can be difficult to work with as a model may easily overfit this data in favor of the class with more number of examples.Complement Naive Bayes is particularly suited to work with imbalanced datasets.In complement Naive 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. This is the literal meaning of the word, complement and hence is called Complement Naive Bayes.

Resource:

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

**iv.Bernoulli Naive Bayes**

**Bernoulli Naive Bayes** is a variant of Naive Bayes which is used for discrete data and it works on Bernoulli distribution. The main feature of Bernoulli Naive Bayes is that it accepts features only as binary values like true or false, yes or no, success or failure, 0 or 1 and so on. So when the feature values are binary we know that we have to use Bernoulli Naive Bayes classifier. Sometimes machine learning algorithms do not work well if the dataset is small, but this is not the case with this algorithm because it gives more accurate results compared to other classification algorithms in the case of a small dataset.It’s fast and can also handle irrelevant features easily.

**Resource:**

<https://thecleverprogrammer.com/2021/07/27/bernoulli-naive-bayes-in-machine-learning/>

**v.Categorical Naive Bayes**

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

Resource:

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

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

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, [MultinomialNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB), [BernoulliNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html#sklearn.naive_bayes.BernoulliNB), and [GaussianNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB) expose a partial\_fit method that can be used incrementally as done with other classifiers as demonstrated in [Out-of-core classification of text documents](https://scikit-learn.org/stable/auto_examples/applications/plot_out_of_core_classification.html#sphx-glr-auto-examples-applications-plot-out-of-core-classification-py). All naive Bayes classifiers support sample weighting.

Contrary to the fit method, the first call to partial\_fit needs to be passed the list of all the expected class labels.

Resource:

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

**3. What is Jaccard and Cosine Similarity?**

Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents. Jaccard similarity can be used to find the similarity between two asymmetric binary vectors or to find the similarity between two sets. In literature, Jaccard similarity, symbolized by J, can also be referred to as Jaccard Index, Jaccard Coefficient, Jaccard Dissimilarity, and Jaccard Distance.

**Jaccard Index = (the number in both sets) / (the number in either set) \* 100**

*Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.*

The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

Resource:

<https://www.statisticshowto.com/jaccard-index/>

<https://www.machinelearningplus.com/nlp/cosine-similarity/>