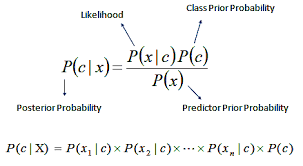
**Assignment -3**

Rashmi S

21BDA02

Naïve Bayes Classifier

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.

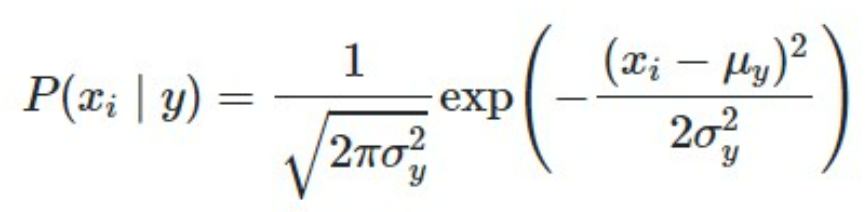


**Question-2**

**In a word document, a. Write the difference between the following:**

1. **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.
* The pdf for the function is as shown below



1. **Multinomial Naive Bayes**

* It is used on discrete distribution with many conditions for different possible features
* Multinomial Naïve Bayes consider a feature vector where a given term represents the number of times it appears or very often i.e. frequency

1. **Complement Naive Bayes**

* Complement Naive Bayes is particularly suited to deal with data that is imbalanced. In Complement Naive Bayes, instead of calculating the probability of an item belonging to a specific class, we calculate the probability of an item being part of all classes.

1. **Bernoulli Naive Bayes**

* Bernoulli Naive Bayes. **BernoulliNB implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions**; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable.

1. **Categorical Naive Bayes**

* The **categorical distribution** is the generalization of the Bernoulli distribution for a categorical random variable, i.e. for a discrete variable with more than two possible outcomes, such as the roll of a die

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

* learning from data that doesn’t fit into main memory. We make use of an online classifier, i.e., one that supports the partial\_fit method, that will be fed with batches of examples. To guarantee that the features space remains the same over time we leverage a HashingVectorizer that will project each example into the same feature space. This is especially useful in the case of text classification where new features (words) may appear in each batch.

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

**Jaccard Similarity** – Is used to check how accurate my predicted values are when compared to the actual values, It’s a method predict the accuracy of the values

* It lies between 0 to 1
* Jaccard similarity = Overlapping positives/ Total positives

**Cosine Similarity** – It is mainly used in recommendation systems. It use the cos angle to predict how close or far away are the predicted v/s actual values

Jaccard similarity is good for cases where duplication does not matter, cosine similarity is good for cases where duplication matters while analyzing text similarity  .

**References:-**

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

[great learning](https://www.mygreatlearning.com/blog/multinomial-naive-bayes-explained/)

[scikit learn](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.ComplementNB.html#sklearn.naive_bayes.ComplementNB)

[Jaccard and Cosine](https://towardsdatascience.com/overview-of-text-similarity-metrics-3397c4601f50#:~:text=Differences%20between%20Jaccard%20Similarity%20and,term%20frequency%20or%20tf%2Didf))