**ML Lab 3**

**Alexei Beuno. H**

**21BDA01**

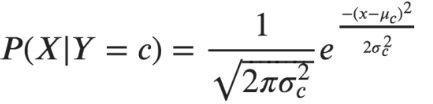
1. Write the difference between the following:

a.

* Gaussian Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm used for many classification functions and is based on the Bayes theorem. Gaussian Naive Bayes is the extension of Naive Bayes. Gaussian Naive Bayes is simplest to implement as you will need to calculate the mean and standard deviation for the training data.

X’s follow a Gaussian or normal distribution, therefore, we substitute the probability density of the normal distribution. That is Gaussian distribution. To compute this, we need the mean and variance of X.



Sigma and mu is the variance and mean of the continuous random variable X computed for a given class c of Y.

Reference: [(2) New Messages! (upgrad.com)](https://www.upgrad.com/blog/gaussian-naive-bayes/#:~:text=Gaussian%20Naive%20Bayes%20Na%C3%AFve%20Bayes%20is%20a%20probabilistic,Na%C3%AFve%20Bayes%20is%20the%20extension%20of%20na%C3%AFve%20Bayes.)

* Multinomial Naive Bayes

Multinomial Naive Bayes algorithm is a probabilistic learning method 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.

Reference: [(2) New Messages! (upgrad.com)](https://www.upgrad.com/blog/multinomial-naive-bayes-explained/)

* Complement Naive Bayes

Complement Naive Bayes is 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.

1. For each class we calculate the probability of the given instance not belonging to it.
2. After calculation for all the classes, we check all the calculated values and select the smallest value.
3. The smallest value(lowest probability) is selected because it is the lowest probability that it is not that particular class. This implies that it has the highest probability to actually belong to that class.

Reference: [Complement Naive Bayes (CNB) Algorithm - GeeksforGeeks](https://www.geeksforgeeks.org/complement-naive-bayes-cnb-algorithm/#:~:text=In%20complement%20Naive%20Bayes%2C%20instead%20of%20calculating%20the,complement%20and%20hence%20is%20called%20Complement%20Naive%20Bayes.)

* Bernoulli Naive Bayes

Bernoulli Naive Bayes only takes binary values. The most general example is where we check each value will be a word or not that appears in the document.

Bernoulli distribution has two mutually exclusive outcomes: P(X=1)=p or P(X=0)=1-p. In Bernoulli theorem we can have multiple features but each one is assumed to be binary valued variable i.e., boolean. Therefore, this class requires samples to be represented as binary-valued feature vectors. In case, any other kind of data is provided, then Bernoulli instance may binarize its input.

Reference: [Bernoulli Naive Bayes and it’s implementation | by NANDINI SHARMA | Medium](https://medium.com/@nansha3120/bernoulli-naive-bayes-and-its-implementation-cca33ccb8d2e)

* Categorical Naive Bayes

It is suitable for classification with discrete features which assumes categorical distribution for each feature. The features should be encoded using label encoding techniques such that each category would be mapped to a unique number.

Reference:[Categorical Naive Bayes Classifier implementation in Python :: InBlog (ineuron.ai)](https://blog.ineuron.ai/Categorical-Naive-Bayes-Classifier-implementation-in-Python-dAVqLWkf7E)

* 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, MultinomialNB, BernoulliNB and GaussianNB expose a partial\_fit method that can be used incrementally as done with other classifiers. All naive bayes classifiers support sample weighting.

The partial\_fit method call of naive Bayes models introduces some computational overhead. It is recommended to use data chunk sizes that are as large as possible, that is as the available RAM allows.

Reference: [1.9. Naive Bayes — scikit-learn 1.0.2 documentation](https://scikit-learn.org/stable/modules/naive_bayes.html)

b. Define which text preprocessing and text transformation steps did you use for the above.

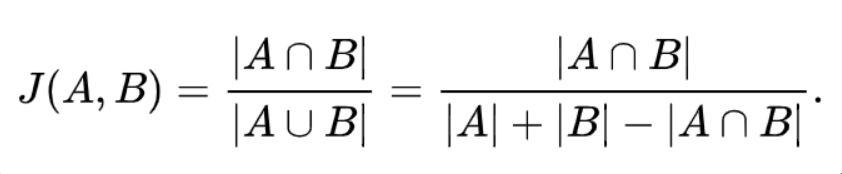
I did not use any text or transformation steps for above.

c. What is Jaccard and Cosine similarity?

**Jaccard Similarity:**

It is the method of measuring similarity between two sets, which means that when you apply two sets as documents, the size of the intersection between two documents is divided by the size of the union. In other words, the similarity measure for two datasets in the range 0 or 1, the closer you are to 1, the more similar the two populations are.

Formula:



**Cosine Similarity:**

Cosine similarity refers to the similarity that can be obtained by vectorizing a document and using the cosine angle between two vectors. If the two vectors have exactly the same orientation, they will have a value of 1, and if they have an angle of 90 degree, they will have a value of -1 if they have opposite directions of 0 or 180 degrees. If the value is closer to 1, the higher the similarity between the two documents.

Formula:

