MACHINE LEARNING 1 LAB

ASSIGNMENT 3

21BDA50

ANAGHA ASHOK

PART B:

In a word document,

a.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

GAUSSIAN NAÏVE BAYES

[**GaussianNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB) implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

P(xi∣y)=12πσy2exp⁡(−(xi−μy)22σy2)

The parameters σy and μy are estimated using maximum likelihood.

MULTINOMIAL NAÏVE BAYES

[**MultinomialNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB) implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). The distribution is parametrized by vectors θy=(θy1,…,θyn) for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θyi is the probability P(xi∣y) of feature i appearing in a sample belonging to class y.

The parameters θy is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

θ^yi=Nyi+αNy+αn

where Nyi=∑x∈Txi is the number of times feature i appears in a sample of class y in the training set T, and Ny=∑i=1nNyi is the total count of all features for class y.

The smoothing priors α≥0 accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting α=1 is called Laplace smoothing, while α<1 is called Lidstone smoothing.

COMPLEMENT NAÏVE BAYES

[**ComplementNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.ComplementNB.html#sklearn.naive_bayes.ComplementNB) implements the complement naive Bayes (CNB) algorithm. CNB is an adaptation of the standard multinomial naive Bayes (MNB) algorithm that is particularly suited for imbalanced data sets. Specifically, CNB uses statistics from the *complement* of each class to compute the model’s weights. The inventors of CNB show empirically that the parameter estimates for CNB are more stable than those for MNB. Further, CNB regularly outperforms MNB (often by a considerable margin) on text classification tasks. The procedure for calculating the weights is as follows:

θ^ci=αi+∑j:yj≠cdijα+∑j:yj≠c∑kdkjwci=log⁡θ^ciwci=wci∑j|wcj|

where the summations are over all documents j not in class c, dij is either the count or tf-idf value of term i in document j, αi is a smoothing hyperparameter like that found in MNB, and α=∑iαi. The second normalization addresses the tendency for longer documents to dominate parameter estimates in MNB. The classification rule is:

c^=arg⁡minc∑itiwci

i.e., a document is assigned to the class that is the *poorest* complement match.

BERNOULLI NAÏVE BAYES

[**BernoulliNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html#sklearn.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. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a BernoulliNB instance may binarize its input (depending on the binarize parameter).

The decision rule for Bernoulli naive Bayes is based on

P(xi∣y)=P(i∣y)xi+(1−P(i∣y))(1−xi)

which differs from multinomial NB’s rule in that it explicitly penalizes the non-occurrence of a feature i that is an indicator for class y, where the multinomial variant would simply ignore a non-occurring feature.

In the case of text classification, word occurrence vectors (rather than word count vectors) may be used to train and use this classifier. BernoulliNB might perform better on some datasets, especially those with shorter documents. It is advisable to evaluate both models, if time permits.

CATEGORICAL NAÏVE BAYES

[**CategoricalNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html#sklearn.naive_bayes.CategoricalNB) implements the categorical naive Bayes algorithm for categorically distributed data. It assumes that each feature, which is described by the index i, has its own categorical distribution.

For each feature i in the training set X, **[CategoricalNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html" \l "sklearn.naive_bayes.CategoricalNB" \o "sklearn.naive_bayes.CategoricalNB)** estimates a categorical distribution for each feature i of X conditioned on the class y. The index set of the samples is defined as J={1,…,m}, with m as the number of samples.

The probability of category t in feature i given class c is estimated as:

P(xi=t∣y=c;α)=Ntic+αNc+αni,

where Ntic=|{j∈J∣xij=t,yj=c}| is the number of times category t appears in the samples xi, which belong to class c, Nc=|{j∈J∣yj=c}| is the number of samples with class c, α is a smoothing parameter and ni is the number of available categories of feature i.

[**CategoricalNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html#sklearn.naive_bayes.CategoricalNB) assumes that the sample matrix X is encoded (for instance with the help of OrdinalEncoder) such that all categories for each feature i are represented with numbers 0,...,ni−1 where ni is the number of available categories of feature i.

OUT OF CORE NAÏVE 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" \l "sklearn.naive_bayes.MultinomialNB" \o "sklearn.naive_bayes.MultinomialNB)**, **[BernoulliNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html" \l "sklearn.naive_bayes.BernoulliNB" \o "sklearn.naive_bayes.BernoulliNB)**, and **[GaussianNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html" \l "sklearn.naive_bayes.GaussianNB" \o "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.

For an overview of available strategies in scikit-learn, see also the [out-of-core learning](https://scikit-learn.org/stable/computing/scaling_strategies.html#scaling-strategies) documentation.

b.What is Jaccard and Cosine Similarity?

1. Jaccard similarity takes only **unique set of words** for each sentence / document while cosine similarity takes **total length of the vectors**. (these vectors could be made from bag of words term frequency or tf-idf)
2. This means that if you repeat the word “friend” in Sentence 1 several times, cosine similarity **changes** but Jaccard similarity does not. For ex, if the word “friend” is repeated in the first sentence 50 times, cosine similarity drops to 0.4 but Jaccard similarity remains at 0.5.
3. Jaccard similarity is good for cases where duplication does not matter, cosine similarity is good for cases where duplication matters while analyzing text similarity. For two product descriptions, it will be better to use Jaccard similarity as repetition of a word does not reduce their similarity.