**ML01-LAB03**

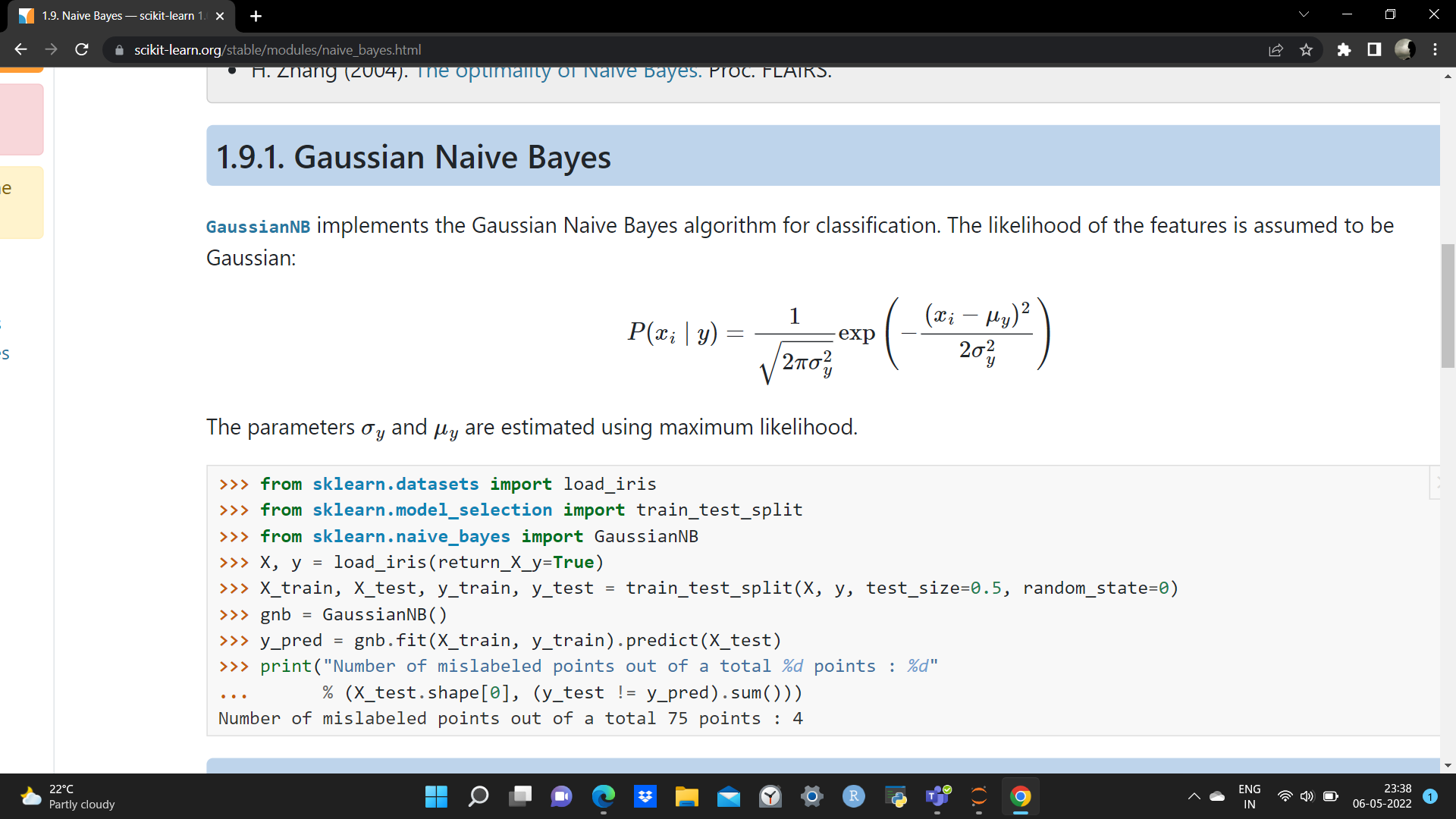
**ASSIGNMENT**

**APARNA k**

**21BDA24**

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.



2. 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.

3. Complement Naive Bayes

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.

4. Bernoulli Naive Bayes

This 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.

5. 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.

6. Out-of-core naive Bayes model fitting

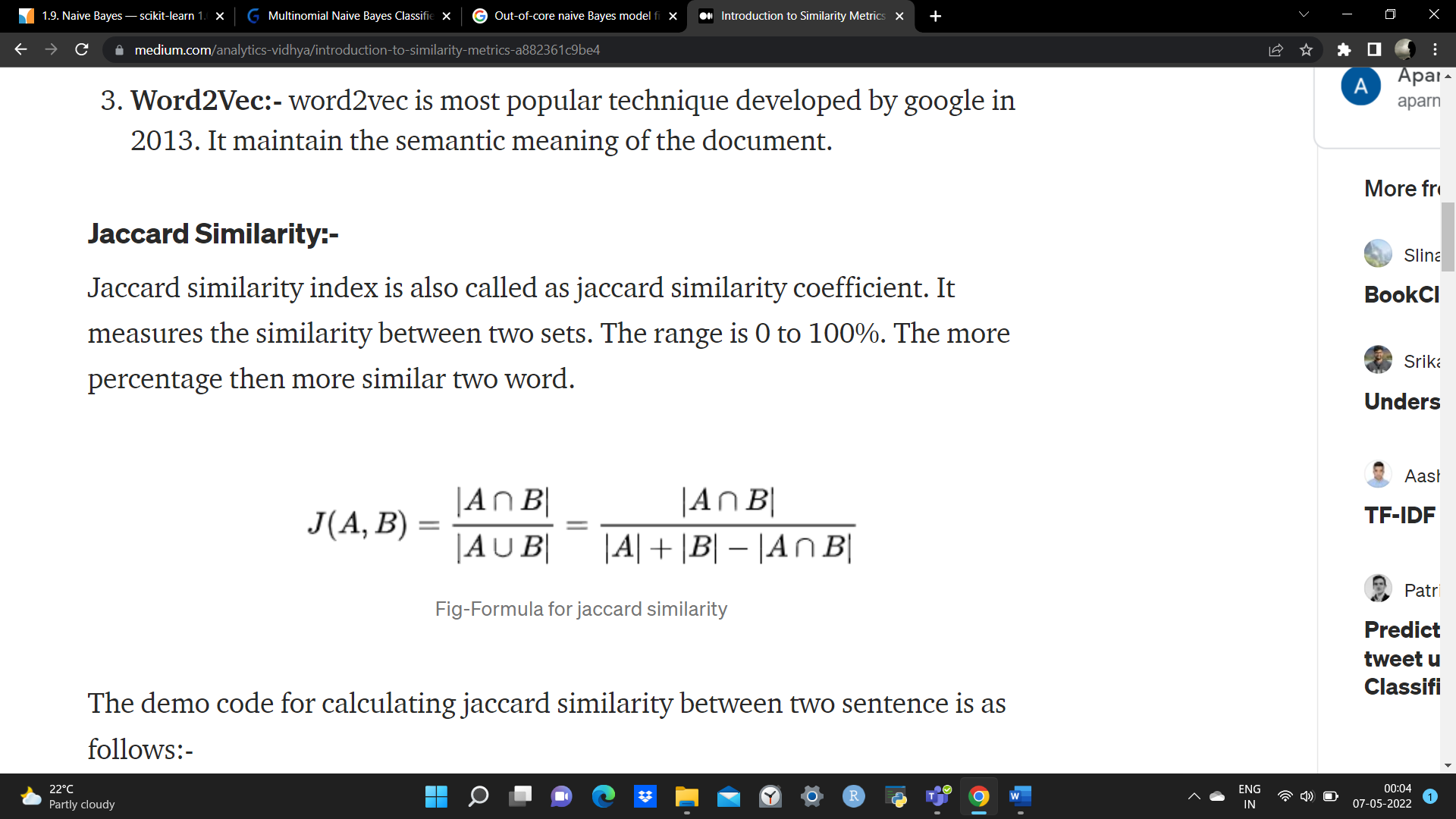
Naive Bayes models can be used to tackle large scale classification problems for which the full training set might not fit in memory. To handle this case, [MultinomialNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB), [BernoulliNB](https://scikit-learn.org/stable/modules/generated/sklearn.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.

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

Jaccard similarity takes only unique set of words for each sentence / document while cosine similarity takes total length of the vectors.

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 then more similar two word.



The cosine similarity is measuring the cosine angle between the two vectors. For cosine we have to convert all sentences to vectors.

