Task 1: [2.5 marks]

1. Given the two confusion tables below, compute
   1. Microaveraged precision, recall, and F1
   2. Macroaveraged precision, recall, and F1
   3. Explain the reason for the difference between the obtained Microaveraged and Macroaveraged F1 measures.

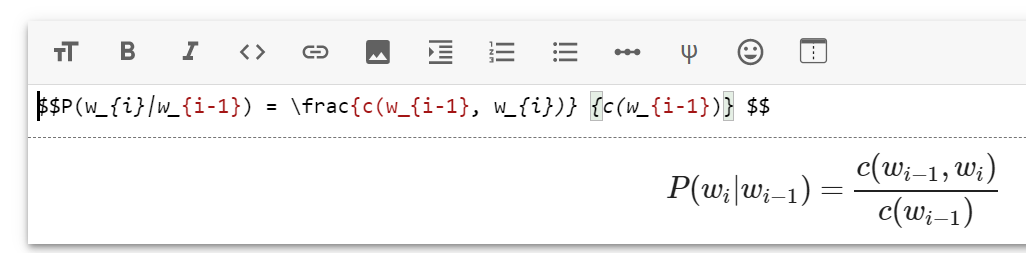
**Class Food**

|  | Truth: YES | Truth: NO |
| --- | --- | --- |
| Classifier: YES | 800 | 200 |
| Classifier: NO | 200 | 500 |

**Class Drink**

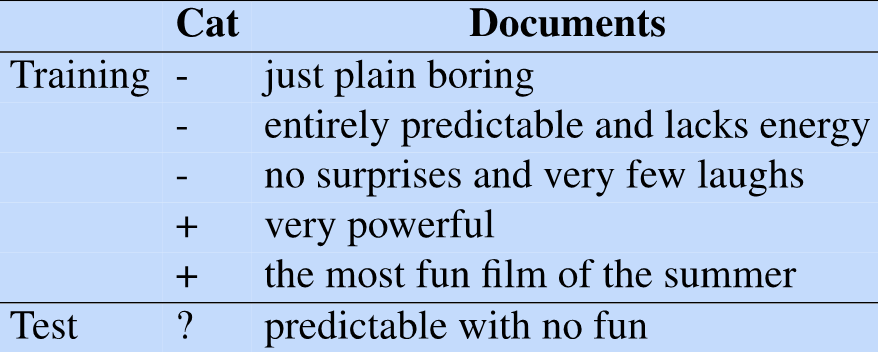
|  | Truth: YES | Truth: NO |
| --- | --- | --- |
| Classifier: YES | 70 | 30 |
| Classifier: NO | 30 | 100 |

* Show your work with the help of Latex.
* Round all numbers to **3 decimal points**.
* [Intro to LaTeX: Learn to write beautiful math equations](https://youtu.be/Jp0lPj2-DQA)
* [**Adding Latex to your notebook**](https://colab.research.google.com/github/bebi103a/bebi103a.github.io/blob/master/lessons/00/intro_to_latex.ipynb)



# Task 2: [1.0 marks]

1. Modify the Multinomial Naïve Bayes classifier function that you developed in **Etivity5, Task3** to train and test a sentiment classifier using the data shown in the table below.



# import math

# from collections import Counter

# def naiveBayesClassifier(trainingSet,testSet):

# YOUR CODE HERE

# trainingSet = [('just plain boring','-'),('entirely predictable and lacks energy','-'),('no surprises and very few laughs','-'),('very powerful','+'),('the most fun film of the summer','+')]

# testSet = [('predictable with no fun','?')]

# naiveBayesClassifier(trainingSet,testSet)

# Task 3: [1.5 marks]

Write a Sentiment Analysis function that takes a string as input and identifies its sentiment using the [TextBlob library](https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis).

Note1: all 5 sample sentences should be tested

**Sample Output:**



## 

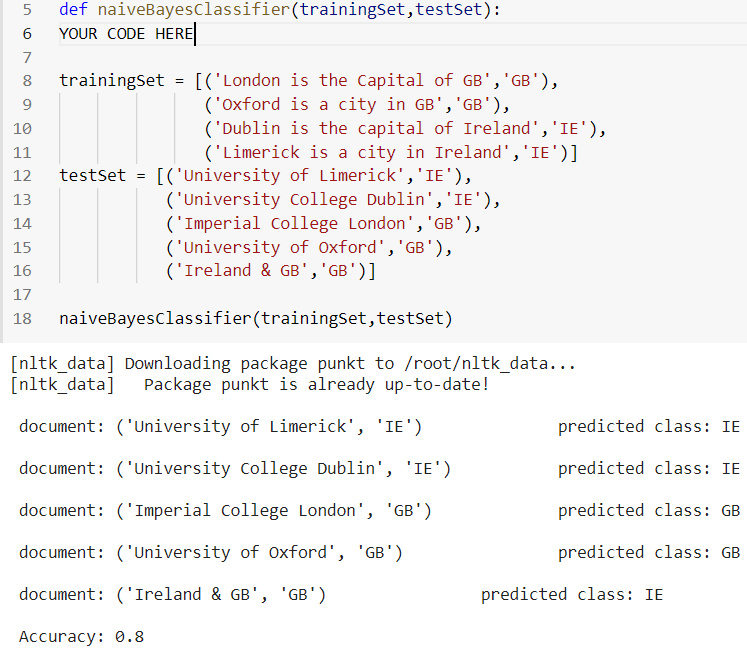
# Task 4: [1 mark]

Write a function that takes sets of training and testing documents as input and uses the [textblob.classifiers](https://textblob.readthedocs.io/en/dev/classifiers.html) module to:

a) train a Naive Bayes classifier using the training set

b) test it using the test set and print out the accuracy of the model.

## Sample Output:



trainingSet = [('London is the Capital of GB','GB'),

('Oxford is a city in GB','GB'),

('Dublin is the capital of Ireland','IE'),

('Limerick is a city in Ireland','IE')]

testSet = [('University of Limerick','IE'),

('University College Dublin','IE'),

('Imperial College London','GB'),

('University of Oxford','GB'),

('Ireland & GB','GB')]

# Task 5: [4 marks]

Write a Python code snippet to:

1. Download the BBC text classification dataset from <https://storage.googleapis.com/dataset-uploader/bbc/bbc-text.csv>
2. Print out:
   1. A few sample documents from the dataset
   2. The total number of documents in the dataset
   3. Total number of documents per class
3. Split the data into training and test sets (**80/20** training/test split); and print out some samples from each set. (don’t use an external library for splitting your dataset). Consider the 10-fold cross-validation for the revised version of your notebook over the weekend.
4. Use the [textblob.classifiers](https://textblob.readthedocs.io/en/dev/classifiers.html) moduleto train and test a Naive Bayes classifier and print out the model’s accuracy.\*

\*Given the large size of the BBC dataset, the training and testing process could take a few minutes.

## Expected Output:

## 

## </END OF ETIVITY>

for lecturer:

* Level of difficulty (easy/fair/hard) easy
* Task1.b Binary Multinomial Naïve Bayes
* Task4, 10-fold cross-validation (quiz why cross valid, do 5 fold)