**Module 5 Predictive Analytics**

**Author’s Name:** Shivani Sharma

**Course Number:** ALY6020

**Course Title:** Predictive Analytics

**Academic Term:** Fall 2019 CPS Analytics

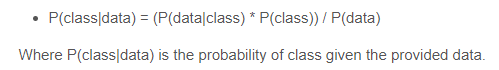
**Instructor’s Name:** Marco Montes de Oca

**Assignment Completion Date:** 10-20-2020



**Introduction**

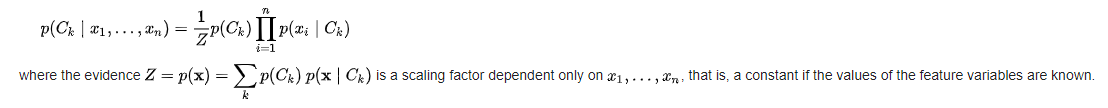
This week’s assignment is based on one of the algorithm i.e “Naïve Bayes” which is used for text classification purpose and it is basically a NLP concept i.e “natural language processing”. Maths behind this algorithm is based on Bayes theorem where we calculate probabilities of finding a event in a set of occurrences. Where we assume one of the fact that each word is independent of the other one.



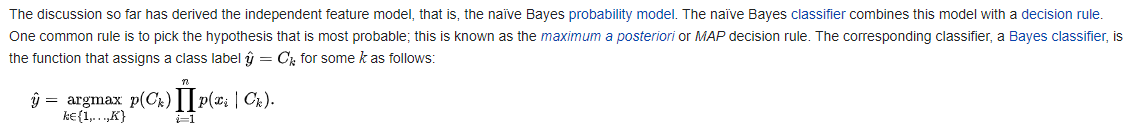
“Naïve Bayes” can be used for both binary and multi class classification problem statements. In this approach we calculate probabilities for each class and make it simplified to make further decisions.

In statistics world “Naïve Bayes Classifiers” is a family of a concept known as “Probabilistic classifier” where we make assumptions among the features.

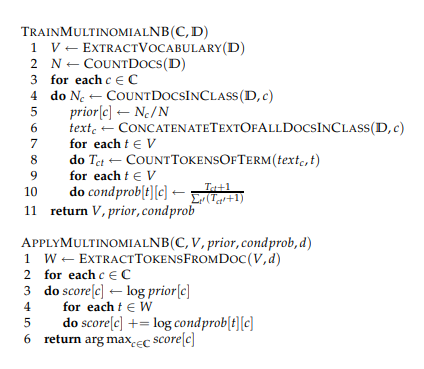
By using the conditional probabilities and “Bayesian probabilities “we develop a probabilistic model:



And then construct classifier from the probability model that we prepare using “naïve bayes” as follows:



We have to apply following algorithm in our assignment as per the research paper provided:

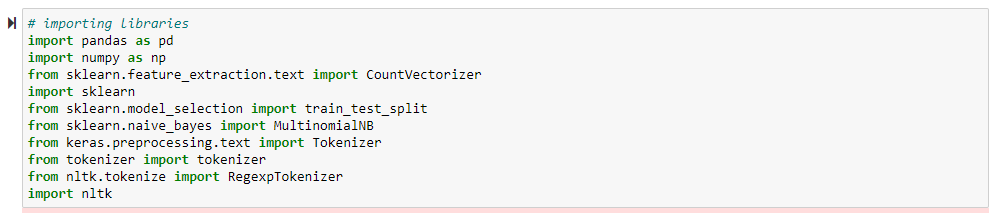


**Analysis**

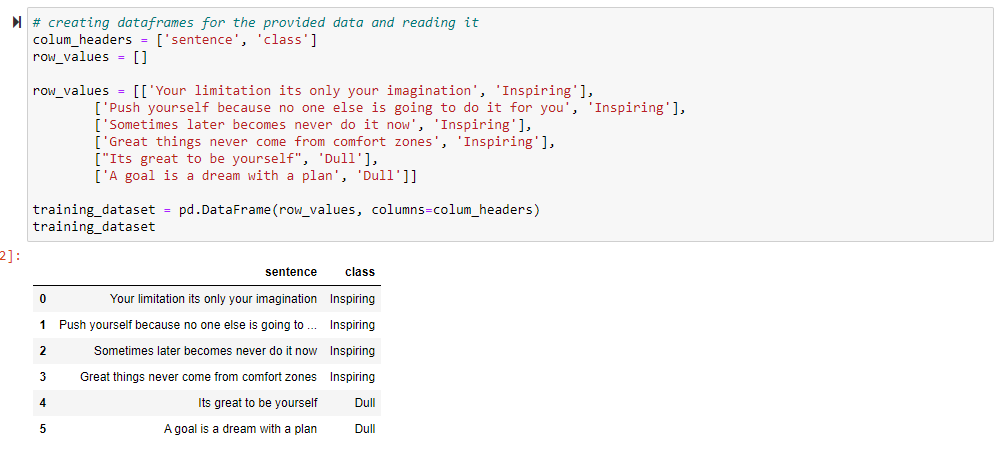
We have been given a set of sentences and the classification for each sentence, based on which we have to classify a sentence if it is “inspiring” or “dull”.

Here is the logic for the same:

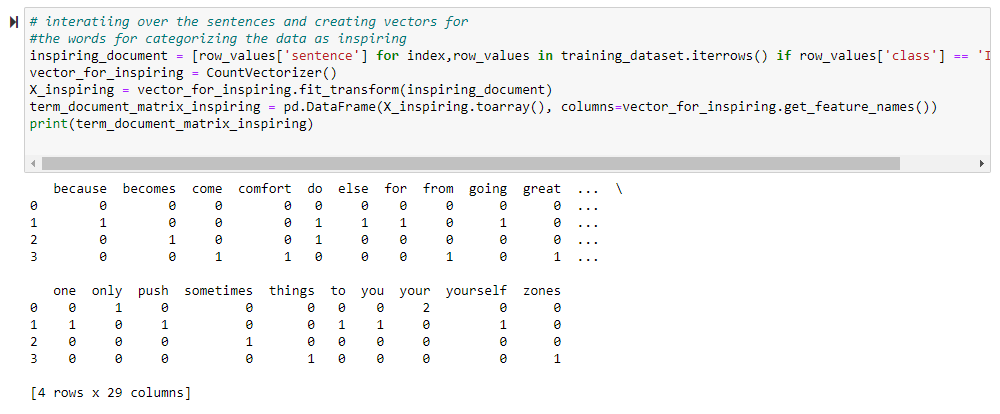
First we have to import the libraries:

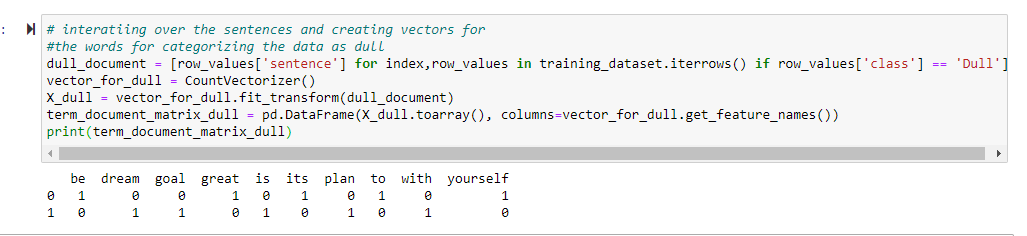


Then we have to create the dataset based on sentences and the classes:

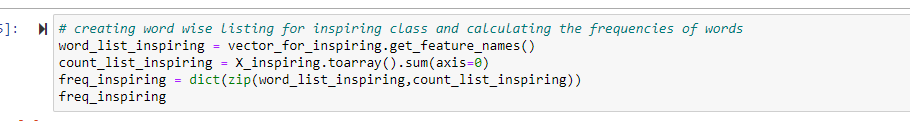


Now since we have data set ready so as per the concept and process explained in the research paper we have to extract vocabulary for each word of both the classes i.e “Inspiring” and “Dull” and logic for which is as follows:

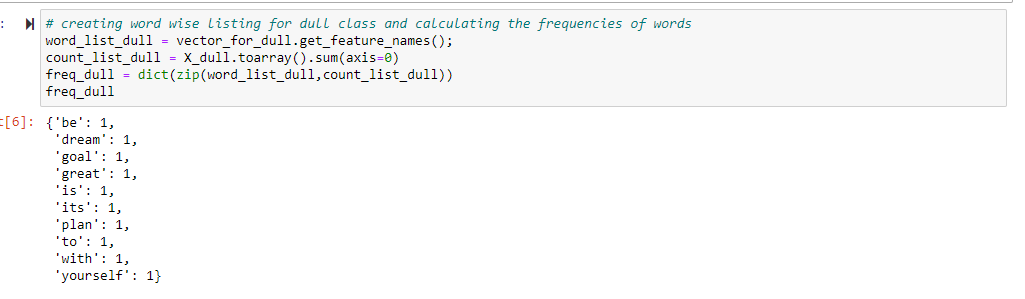




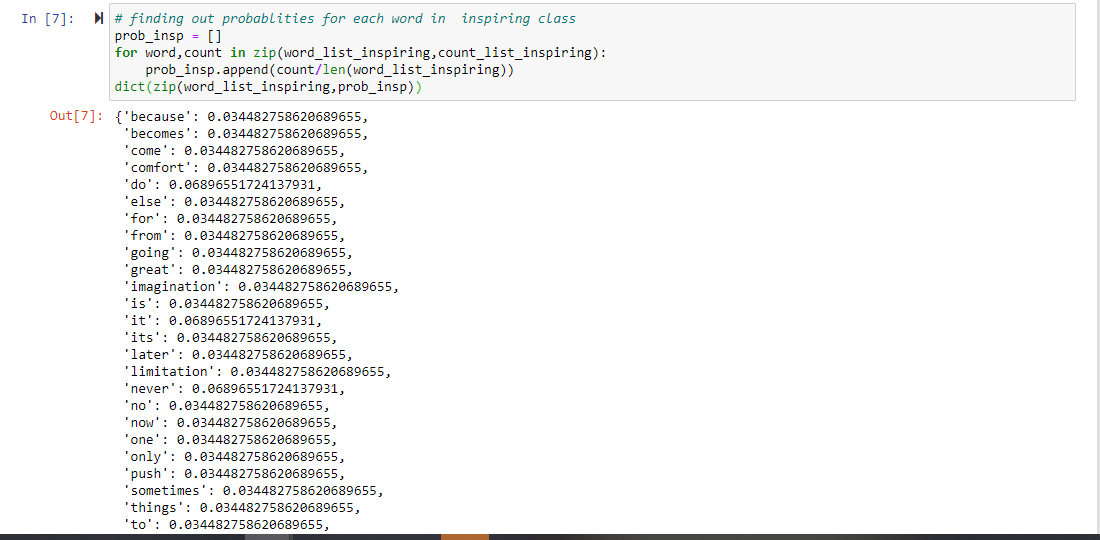
Then we have to create a word wise listing for all the words in category “inspiring” and “Dull” and calculate the frequencies of words. Logic is as follows:

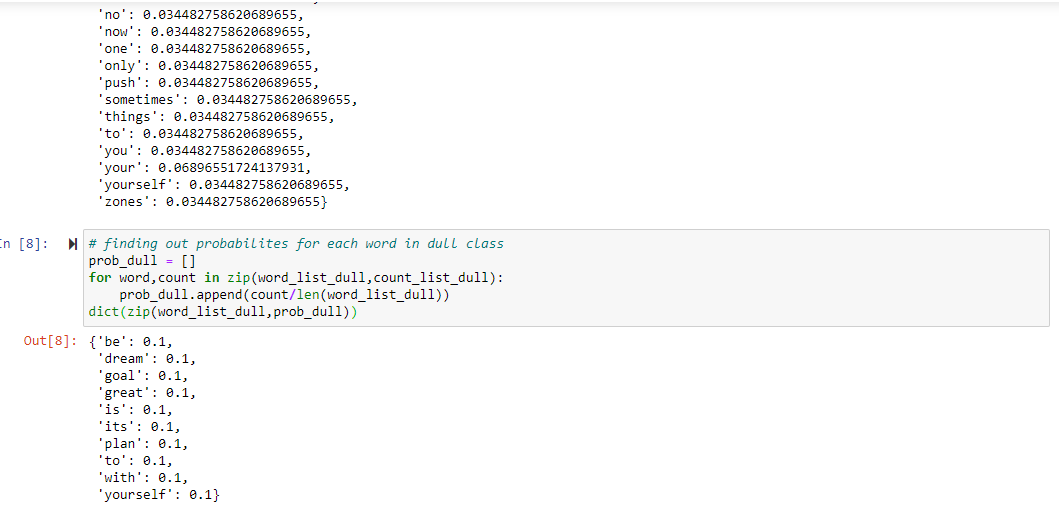




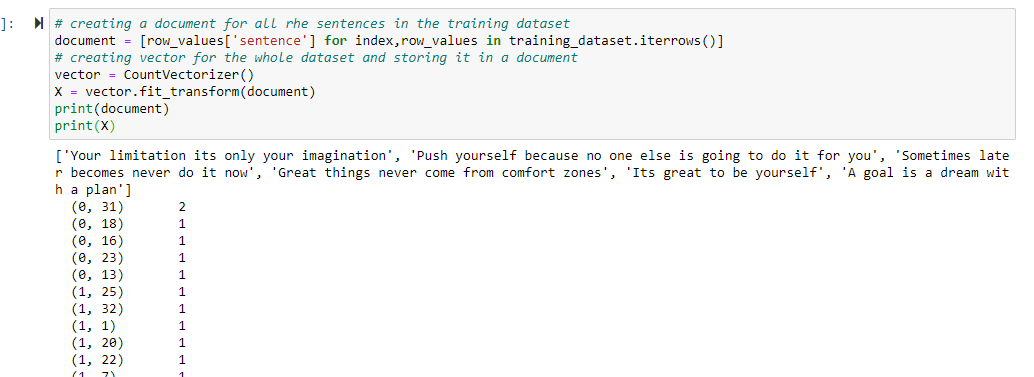


And then calculating the probabilities for each of both the categories as follows:

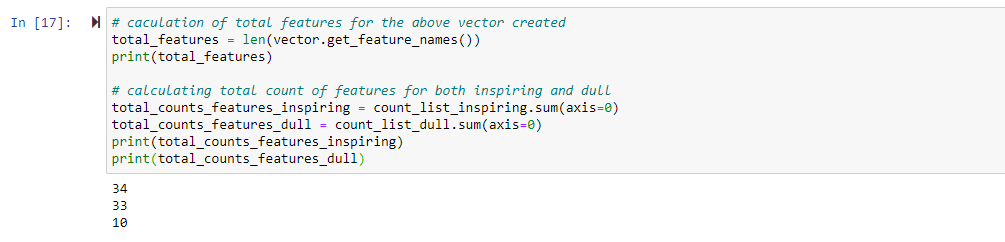




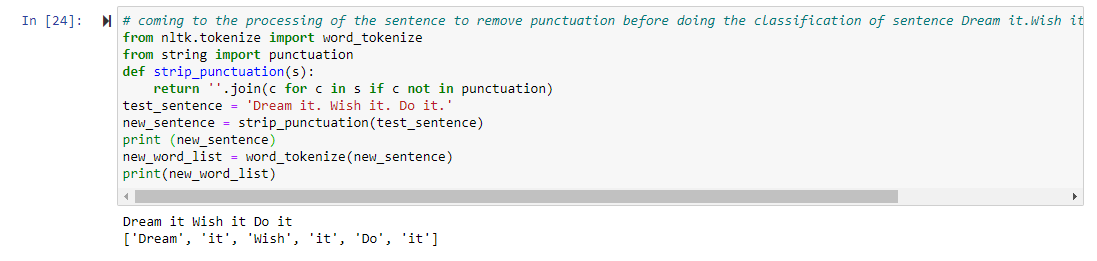
Now creating a single document and count of occurrences of words for whole dataset as follows:



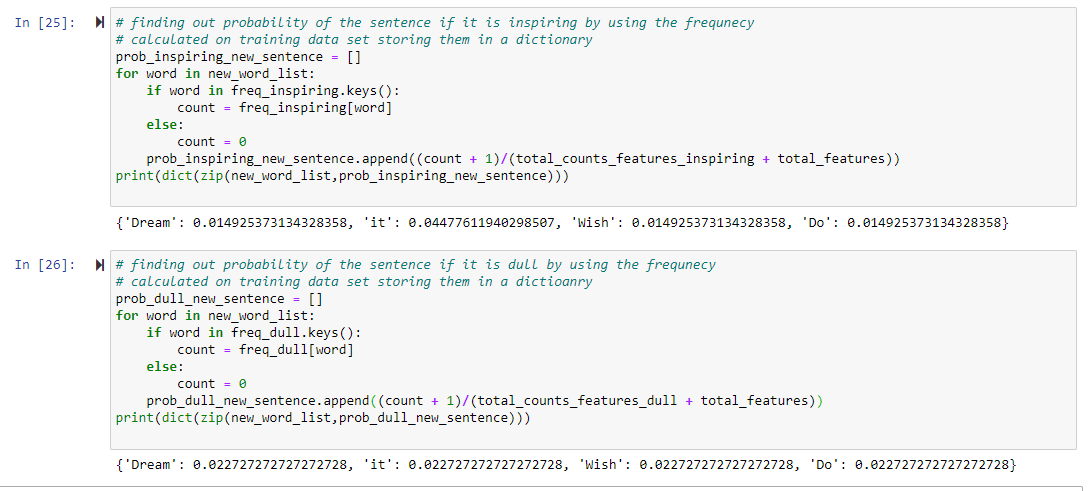
Now calculating the total number of features of the data set and count of features for both the classes:



Since we are done with processing of the data on training set now as per the steps of the algorithm we have to apply the above used algorithm on test data to predict the class of the sentence.

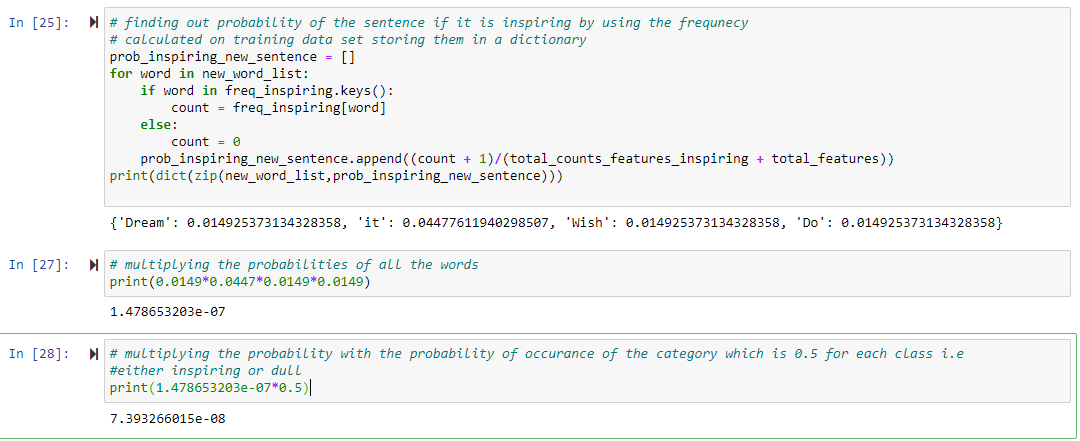


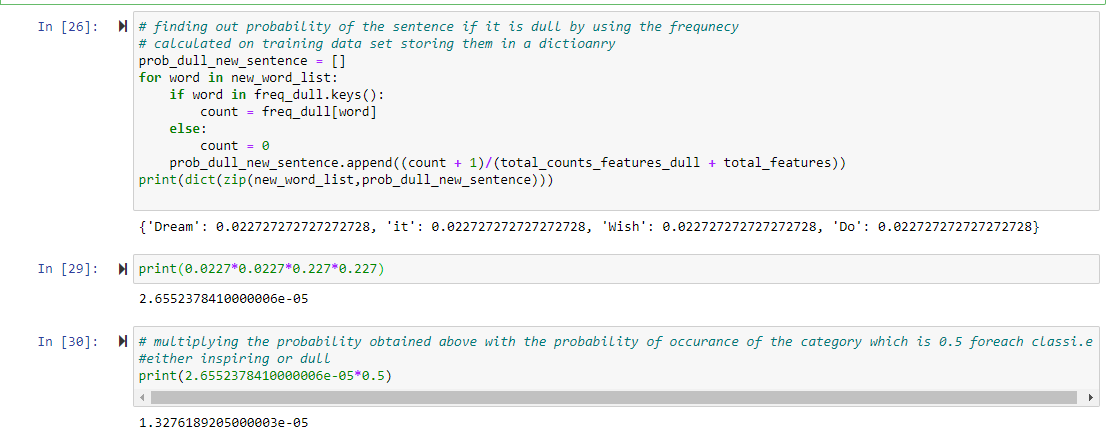
Now we will calculate the probability of each word for the inspiring and dull classes of the test sentence.



**Conclusion**

From the above results and steps 1-6 of the later part of the algorithm we have to apply following concept to categorize the word into one of the class:





**Since the value of probability for “inspiring” class is coming more than “Dull” so the sentence we have been provided to test comes into the category of “Inspiring”.**

Now coming to the second part of the problem statement i.e explaining “Naïve Bayes” using different approach for doing text classification.

We have a library NLTK which is used for applying “naïve bayes classifier” algorithm directly by passing dataset into it. We have to keep the number of features that we are passing into it limited. This is a supervised machine learning algorithm where the model puts the tags into respective labels and categorizes them. NLTK includes a huge set known as “Corpora” which we use for building model. We directly create classifiers using inbuilt function “NaiveBayesClassifer” of “NLTK” library. Similarly, we have “DecisionTreeClassifier” function for categorizing the words. And there are any more powerful functions created and we have to just pass the training datasets into them and use them for building models. Here is the reference for this information regarding this algorithm: <https://www.nltk.org/book/ch06.html>

# References

1. <https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf>
2. Epley, K. (2019, June 30). Naive Bayes Document Classification in Python. Retrieved October 20, 2020, from https://towardsdatascience.com/naive-bayes-document-classification-in-python-e33ff50f937e
3. Aiyappan, J. (2019, September 08). Naive Bayes Classifier for Text Classification. Retrieved October 20, 2020, from https://medium.com/analytics-vidhya/naive-bayes-classifier-for-text-classification-556fabaf252b
4. Deshpande, M. (2019, September 25). Text Classification Tutorial with Naive Bayes. Retrieved October 20, 2020, from https://pythonmachinelearning.pro/text-classification-tutorial-with-naive-bayes/
5. Python Machine Learning Tutorial. (n.d.). Retrieved October 20, 2020, from <https://www.python-course.eu/text_classification_python.php>
6. Brownlee, J. (2019, October 24). Naive Bayes Classifier From Scratch in Python. Retrieved October 20, 2020, from https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/
7. <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>
8. Grg, P. (2020, February 22). Pre-Processing Text in Python. Retrieved October 21, 2020, from https://blog.ekbana.com/pre-processing-text-in-python-ad13ea544dae