Summary:

My contributions

Contribution # 1: Data Preprocessing and Normalizing

Contribution # 2:Doing everything from scratch without using Scikit Learn for model building and to find the accuracy of the model.

Contribution # 3:Create a simpler and easy-to-understand version of the sklearn.naive bayes estimator that delivers identical results on a testing dataset.

Method Used

In this Project,I will be working on an email classification problem using one of the simplest techniques called naive Bayes classification., which we learned about in class.Here,I will classify the email whether it is spam or ham as discussed in our class.Naive Bayes classifiers are the classifiers that are based on Bayes' theorem, a theorem that gives the probability of an event based on prior knowledge of conditions related to the event. It can be used to build a naive but good enough spam classifier.

The most difficult of all is unwanted spam and commercial communication. Because of the continual battle between spam filtering software and anonymous spam and promotional mail senders, spam communication algorithms must be iterated on a regular basis. We receive a large number of emails in our everyday lives. Some emails are useful, while others are not. Spam email is an unsolicited email sent in bulk. Because we don't want spam emails, spam classifiers direct them to spam folders before they reach our inbox. With the advancements in machine learning and deep learning, spam filters have begun to employ them to protect clients, and they have been largely effective. Spam filters have done a good job in minimizing losses and enhancing efficiency, from saving email reading time to protecting customers from frauds, deceptions, and phishing.Modern spam filtering software is always struggling to appropriately categorize emails.

For binary (two-class) and multiclass classification problems, Naive Bayes is a popular classification algorithm. Because the computations of the probability for each class are simplified to make their calculations tractable, it is known as Naive Bayes or stupid Bayes.
Instead of seeking to determine the probability of each attribute value, the class value is considered to be conditionally independent.This is a significant assumption that is unlikely to be true in real data, namely that the attributes do not interact. Nonetheless, the approach performs surprisingly well on data where this assumption does not hold.

I'm trying to provide a complete step-by-step pythonic implementation of naive bayes without the use of any frameworks or Scikit learn, and by keeping in mind the mathematical and probabilistic difficulties I usually encounter when trying to delve deep into the algorithmic insights of machine learning algorithms. I chose to develop the method from scratch while studying about Naive Bayes classifiers to help solidify my understanding of the math.