EMAIL SPAM DETECTION USING PYTHON

AND

MACHINE LEARNING

ABSTRACT

E-mail spam is the very recent problem for every individual. The e-mail spam is nothing it’s an advertisement of any company/product or any kind of virus which is receiving by the email client mailbox without any notification. To solve this problem the different spam filtering technique is used. The spam filtering techniques are used to protect our mailbox for spam mails. In this project, we are using the Naïve Bayesian Classifier for spam classification. The Naïve Bayesian Classifier is very simple and efficient method for spam classification. Here we are using the Emails.csv dataset for classification of spam and non-spam mails. The feature extraction technique is used to extract the feature. The result is to increase the accuracy of the system.

CHAPTER – 1

INTRODUCTION

#### 

* 1. INTRODUCTION

Email has become one of the most important forms of communication. In 2014, there are estimated to be 4.1 billion email accounts worldwide, and about 196 billion emails are sent each day worldwide.Spam is one of the major threats posed to email users. In 2013, 69.6% of all email flows were spam. Links in spam emails may lead to users to websites with malware or phishing schemes, which can access and disrupt the receiver’s computer system. These sites can also gather sensitive information from. Additionally, spam costs businesses around $2000 per employee per year due to decreased productivity. Therefore, an effective spam filtering technology is a significant contribution to the sustainability of the cyberspace and to our society. Current spam techniques could be paired with **content-based spam filtering** methods to increase effectiveness. Content-based methods analyze the content of the email to determine if the email is spam. The goal of our project was to analyze machine learning algorithms and determine their effectiveness as content-based spam filters.

1.2 MOTIVATION

* Unwanted e-mails irritating internet connection
* Critical e-mail message are missed and / or delayed
* Millions of compromised computers
* Billions of dollars lost worldwide
* Identity theft
* Spam can crash mail servers and fill up hard drives

1,3 POBLEM STATEMENT

Spam e-mails can be not only annoying but also dangerous to consumers. Spam e-mails can be defined as :

* Anonymity
* Mass Mailings
* Unsolicited

Spam e-mails are messages randomly sent to multiple addresses by all sorts of groups, but mostly lazy advertisers and criminals who wish to lead us to phishing sites. Therefore to avoid these situation ,in our project we are detecting the spam emails from huge content of incoming mails by using Multinomial Naive Bayesian Algorithm of Machine Learning using python. The naive Bayesian Classifier is one of the most popular and simplest methods for classification. Naive Bayesian Classifiers are highly scalable, learning problem the number of features are required for the number of linear parameter. Training of the large data simple can be easily done with Naive Bayesian Classifier, which takes a very less time as compared to other classifier. The accuracy of system is increase using Naïve Bayesian Classifier.

* 1. OBJECTIVES

The objective of identification of Spam e-mails are:

* To give knowledge to the user about the fake e-mails and relevant e-mails.
* To classify that mail spam or not.

CHAPTER – 2

METHODOLOGY

2.1 IMPLEMENTATION

E-mail spam classification has major issue in today’s electronic world. To solve this problem the different spam classification methods are used. Using this spam detection technique we can identifies the spam and non-spam mails in our mailbox. In this work we are using the Naïve Bayesian Classifier for email spam classification.

In this work also use feature extraction techniques for providing efficient dataset. The feature extraction techniques are used when the input data is too large and it is redundant in nature so feature is extracted to obtain an accurate result. In this work we are using the word-count algorithm for extracting feature from the dataset. Here we use the Emails.csv data set which contains total 5730 mails.

The Feature Extraction: The word-count algorithm is very simple to implement and provide a flexible result. In this algorithm we pre-process the dataset and remove the stop-words and non-words in dataset. And then it counts the total number of unique word out of the total word and finds the frequency of that word in a particular document. The main thing about this algorithm is to makes a dictionary. In that dictionary the path of the file is stored which is pre-processed. So the redundancy problem is removed. For counting the word and store the frequency of that word is very helpful to find the unique word.

Naive Bayes Classifier:

The Bayesian classification exemplifies a supervised learning technique and at the same time a statistical technique for classification. It acts as a fundamental probabilistic model and let us seize ambiguity about the model in an ethical way by influencing the probabilities of the results. It is used to provide solution to analytical and predictive problems . Bayesian classification is named after Thomas Bayes (1702–1761), who proposed the algorithm. The classification offers practical learning algorithms and previous knowledge and experimental data can be merged. Bayesian Classification offers a beneficial viewpoint for comprehending and appraising several learning algorithms. It computes exact likelihoods for postulation and it is robust to noise in input data. A Naive Bayes classifier is a straightforward probabilistic classifier that is founded on Bayes theorem with sound assumptions that are independent in nature. A better expression for the probability model should be autonomous characteristic model Bayes Theorem: Prob (B given A) = Prob (A and B)/Prob (A). The notion of class restrictive autonomy was created to make computation easier, and is the basis of tagging the algorithm ‘naïve’. Nevertheless, the algorithm is effective and very robust. It performs just like other supervised learning algorithms. There have been an upsurge in the acceptance of NB as a simple and computationally efficient algorithm with satisfactory performances in solving real-world problems. As a result of its exceptional qualities, NB classifiers has found application as classification algorithm in text, spam email, sentiment analysis, recommender systems, spam reviews, and other online applications. Naive Bayes classifiers are particularly utilised in text classification (because it produces superior result in multi class problems and independence rule) and have greater success rate when compared to some other machine learning algorithms. Due to this obvious advantage, it is extensively applied in the field of spam filtering (detect spam e-mail) and sentiment analysis (in social media analysis, to recognise positive and negative customer opinions). Spam filtering is the most famous use of the NB classifier. It is a general method for differentiating unauthorised emails i.e. spam from the lawful ones, often referred to as ham. Most mail clients implement Bayesian spam filtering these days. Whereas users can install email-filtering software, server-side email filters utilising Bayesian spam filtering methods are entrenched inside software that makes e-mail facilities to perform effectively [61]. Virtually all the statistic-based spam filtering techniques are using Naïve Bayes' classifier to group the statistics of each token to a total score [62,126], and the score is used in making resolution on the filtering. According to [63], the token T which denote the spamminess (spam rating) is computed as illustrated in Eq. (12): S[T ] = Cspam[T] /Cspam[T] + Cham[T]. Where: CSpam(T) = The number of spam messages containing token T, CHam(T) = The number of ham messages containing token T, There will be need to merge the different token's spamminess to calculate the overall message spamminess in order to compute the probability for a message M with tokens {T1,......,TN}. Computing the product of specific token's spamminess and comparing it with the product of specific token's hamminess is a straightforward way to make classifications.

Email Spam filtering process:

An email message is made up of two major components which are the header and the body. The header is the area that have broad information about the content of the email. It includes the subject, sender and receiver. The body is the heart of the email. It can include information that does not have a pre-defined data. Examples include web page, audio, video, analog data, images, files, and HTML markup. The email header is comprised of fields such as sender's address, the recipient's address, or timestamp which indicate when the message was sent by intermediary servers to the Message Transport Agents (MTAs) that function as an office for organising mails. The header line usually starts with a “From” and it goes through some modification whenever it moves from one server to another through an in-between server. Headers allow the user to view the route the email passes through, and the time taken by each server to treat the mail. The available information have to pass through some processing before the classifier can make use of it for filtering .The necessary stages that must be observed in the mining of data from an email message can be categorised into the following:

Pre-processing: This is the first stage that is executed whenever an incoming mail is received. This step consists of tokenization.

Tokenization: This is a process that removes the words in the body of an email. It also transforms a message to its meaningful parts. It takes the email and divides it into a sequence of representative symbols called tokens.

Lemmatization: In linguistics, it is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. In computational linguistics, lemmatization is the process of determining the lemma for a given word.

Bag of Words Model: We cannot directly feed our text into that algorithm. Hence, Bag of Words model is used to preprocess the text by converting it into a bag of words, which keeps a count of the total occurrences of most frequently used words.

This model can be visualized using a table, which contains the count of words corresponding to the word itself.

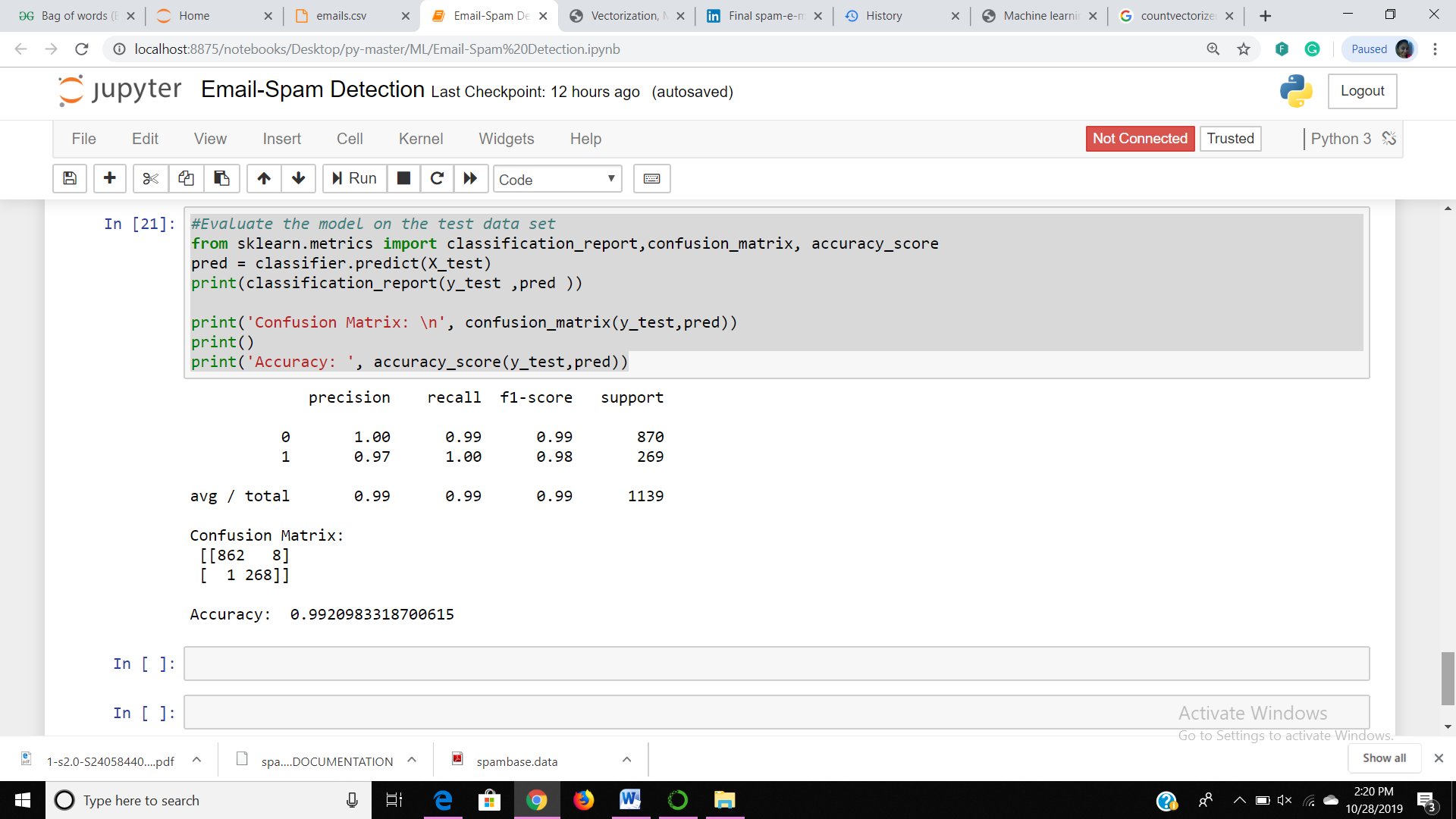
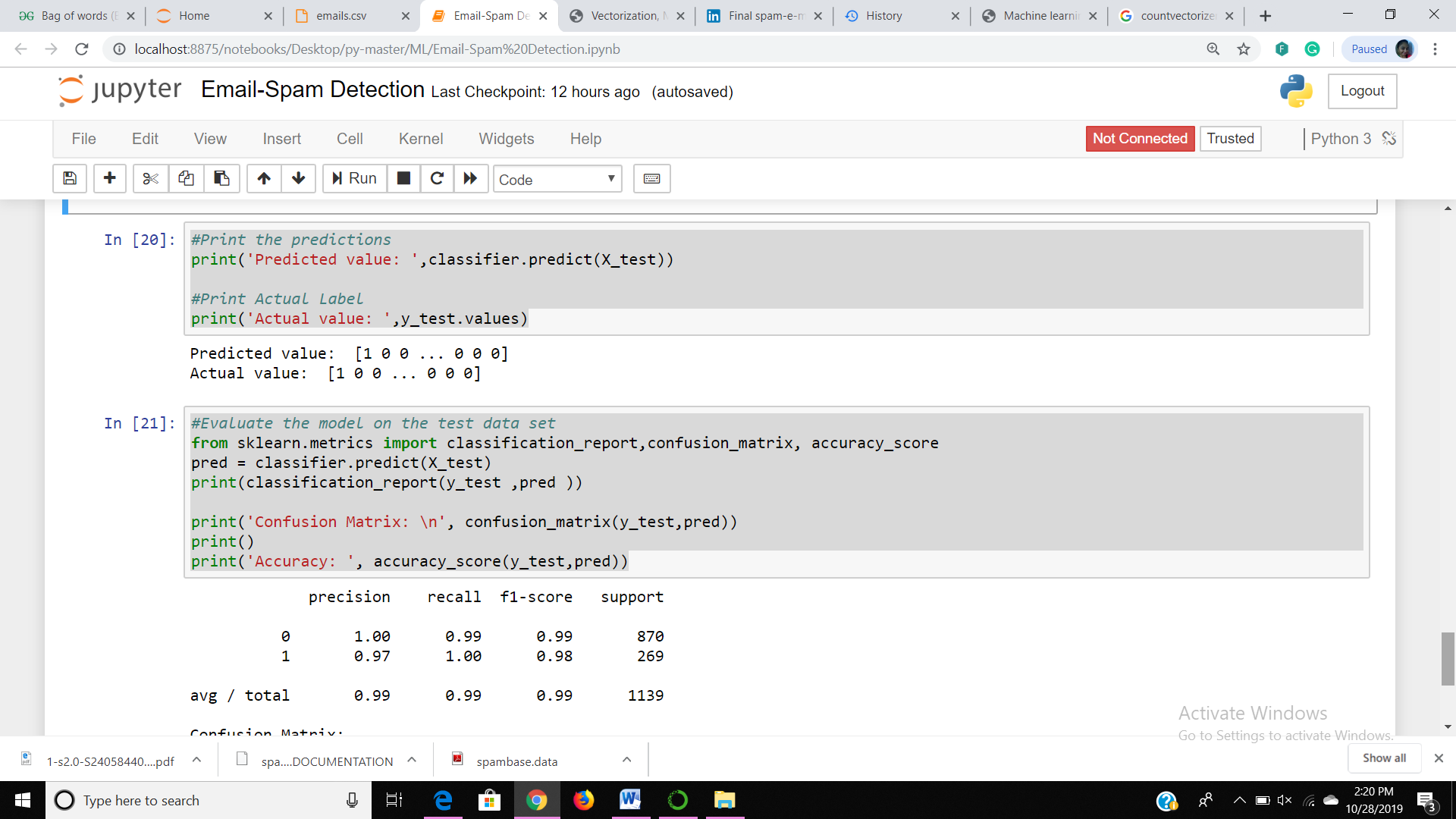
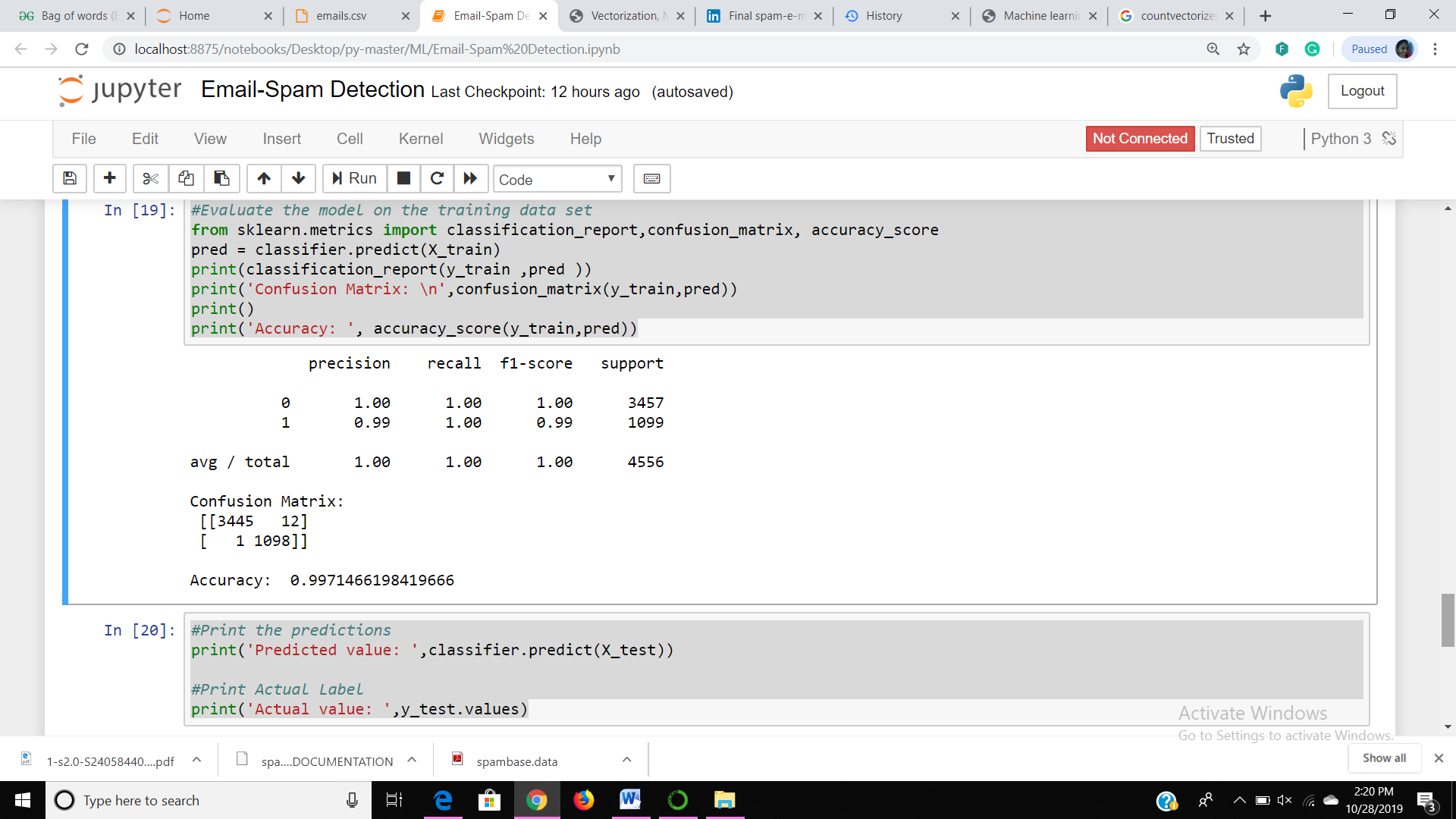
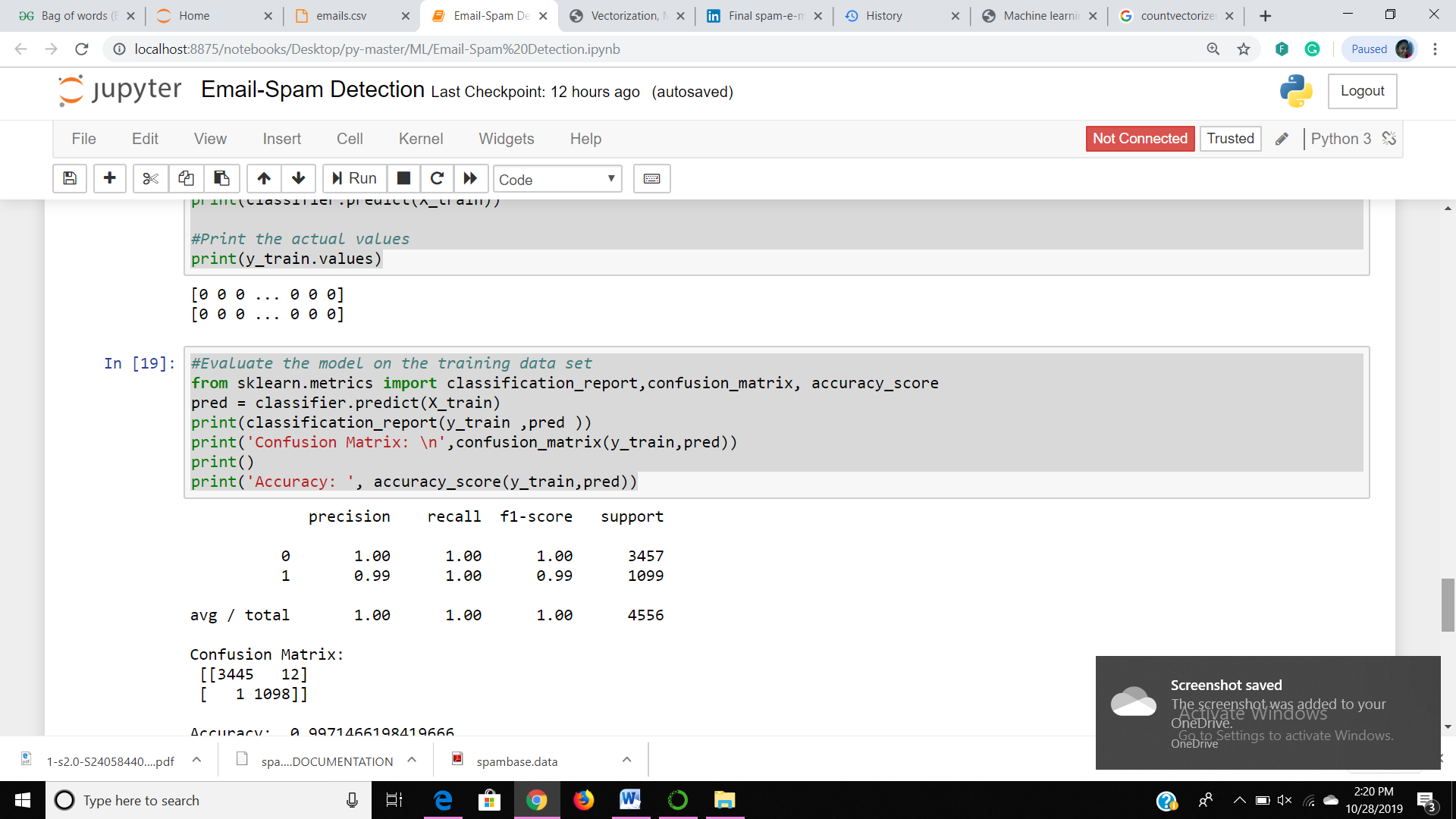
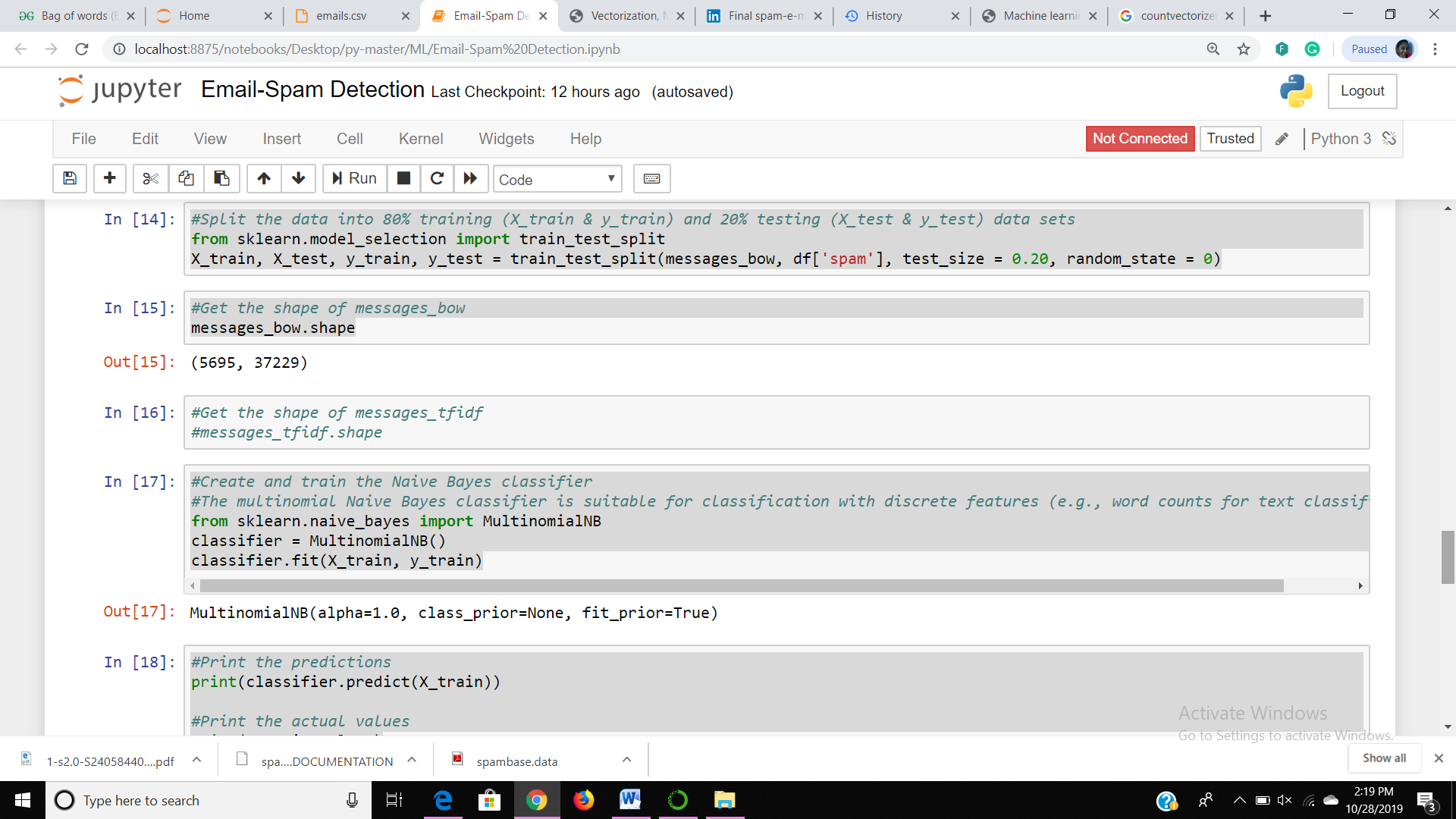
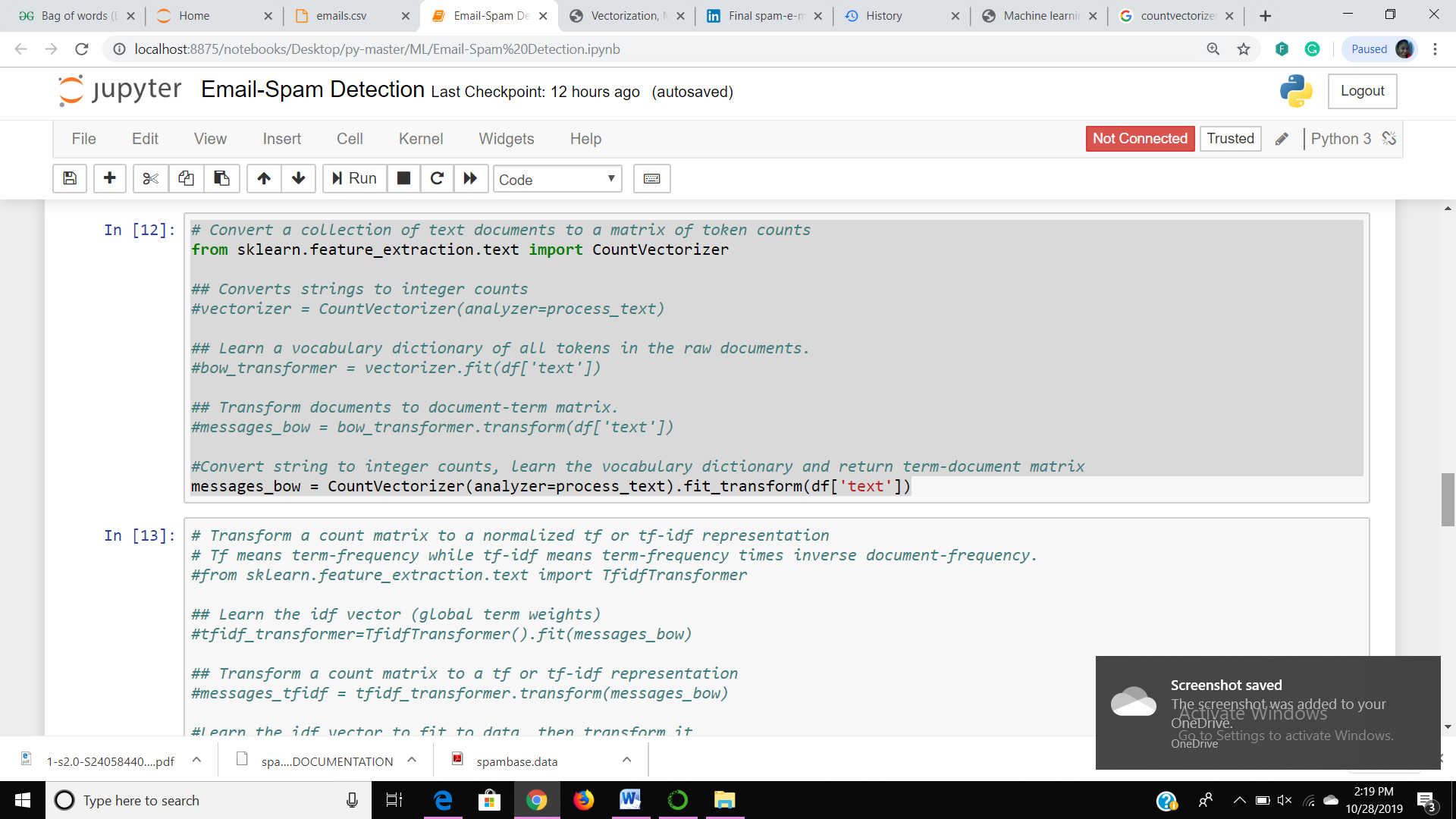
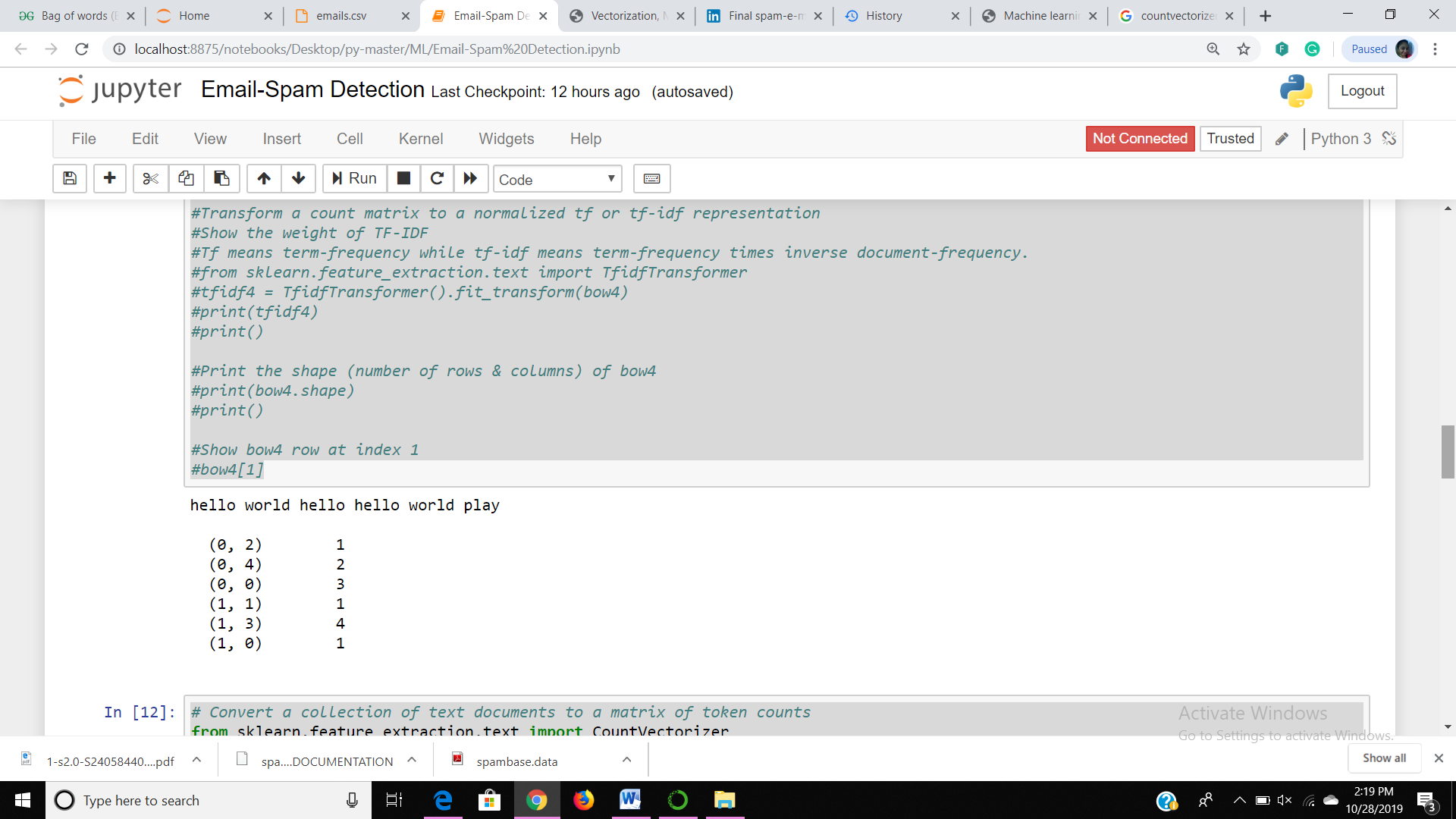
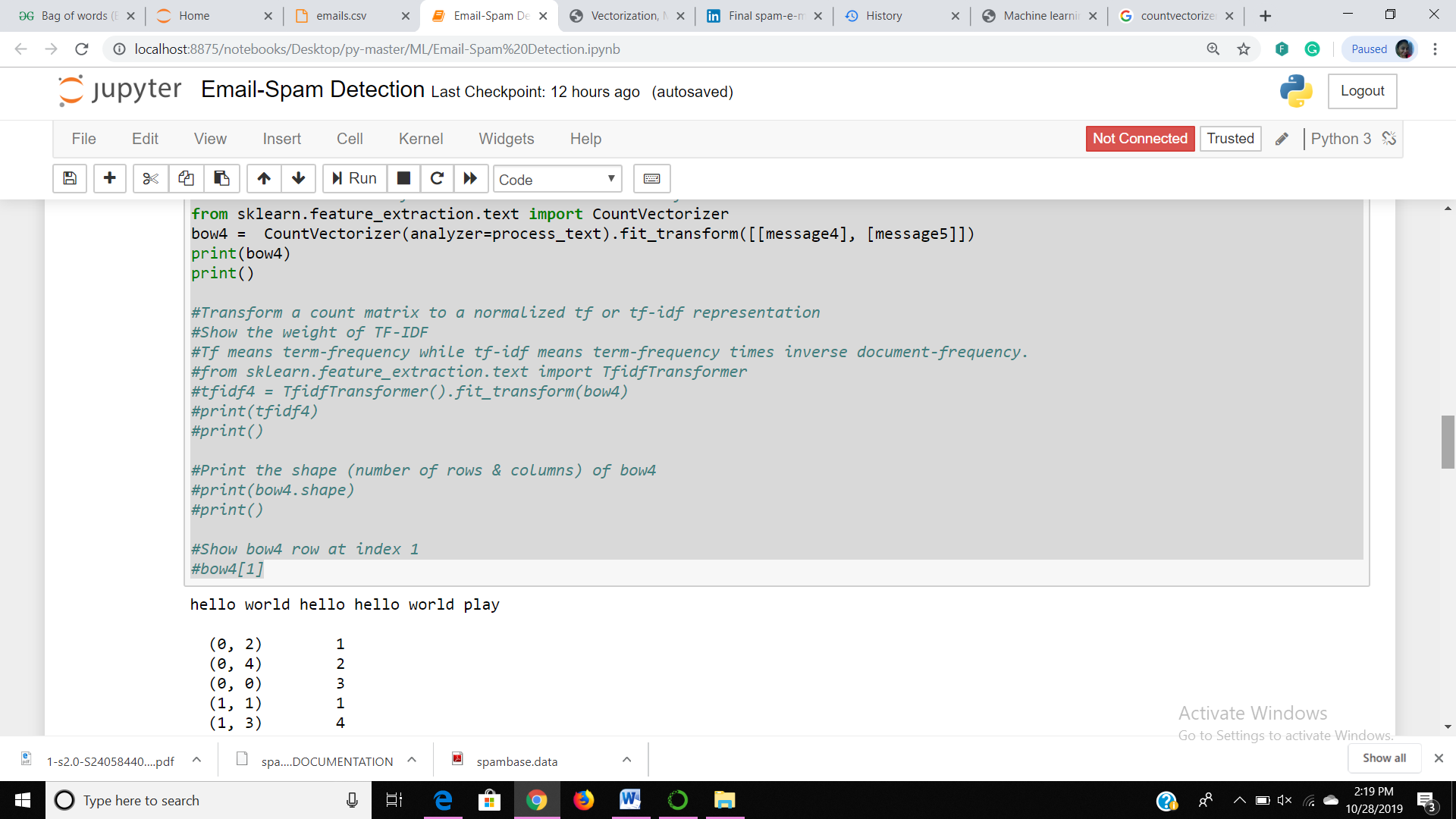
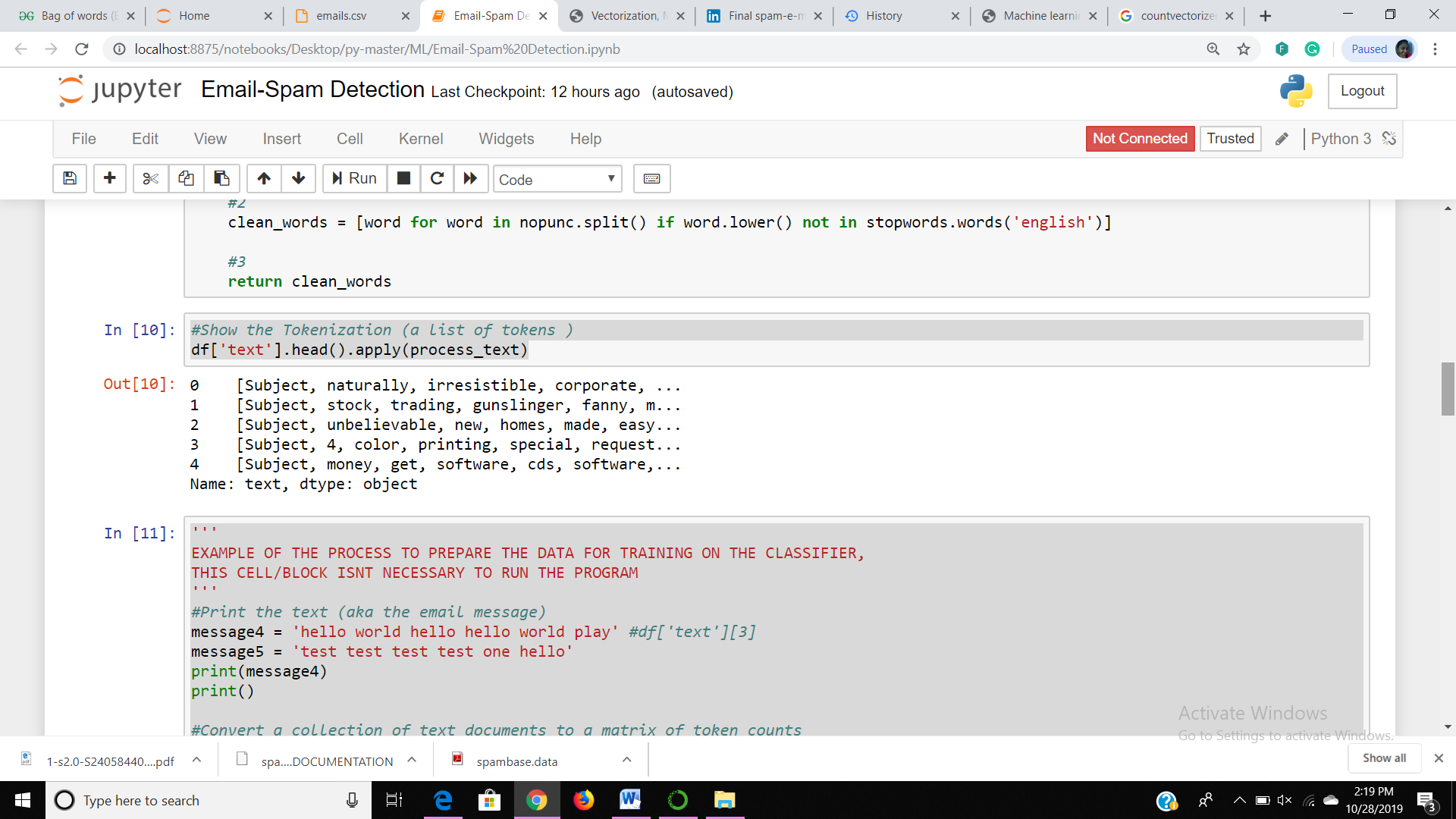
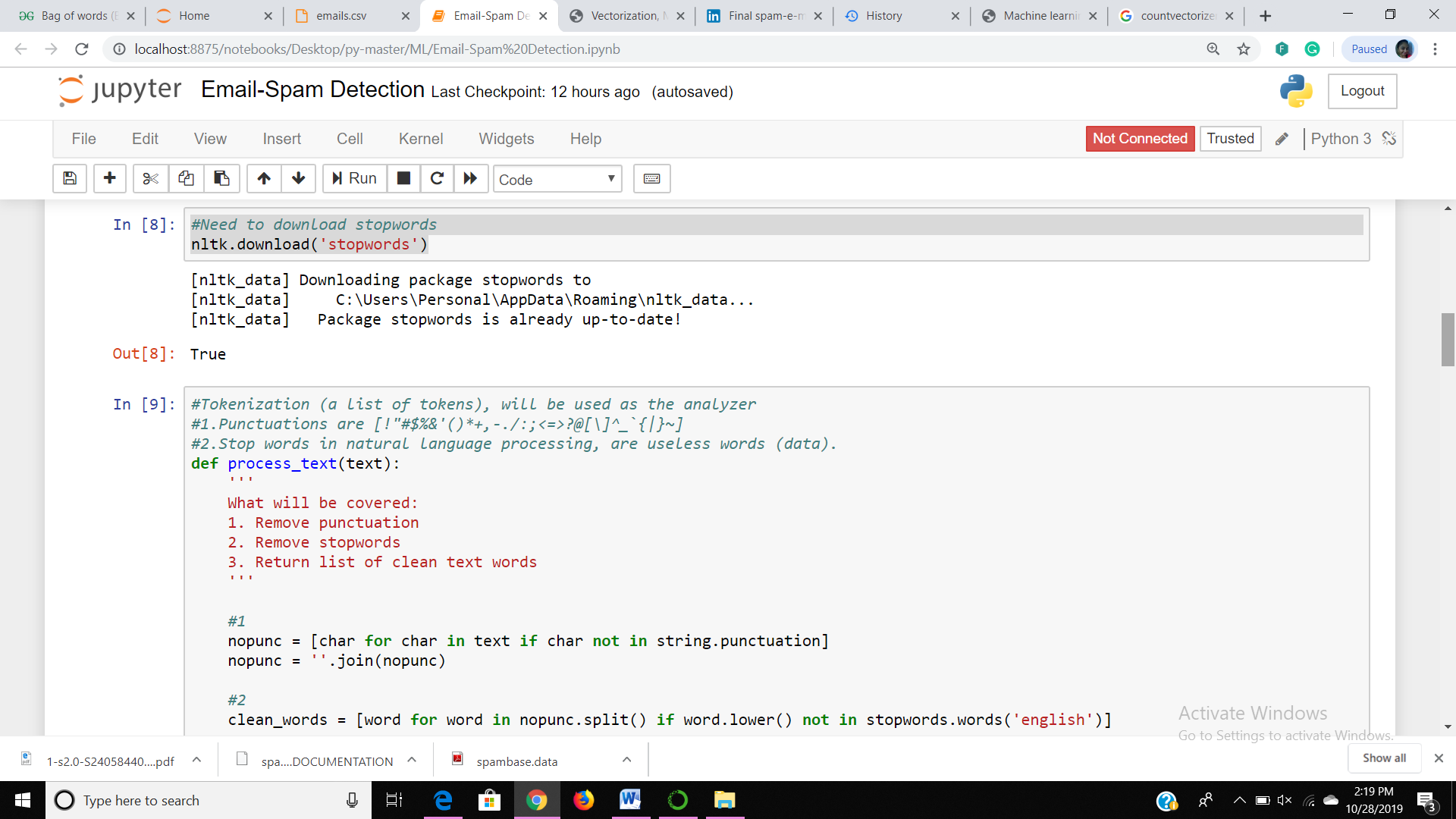
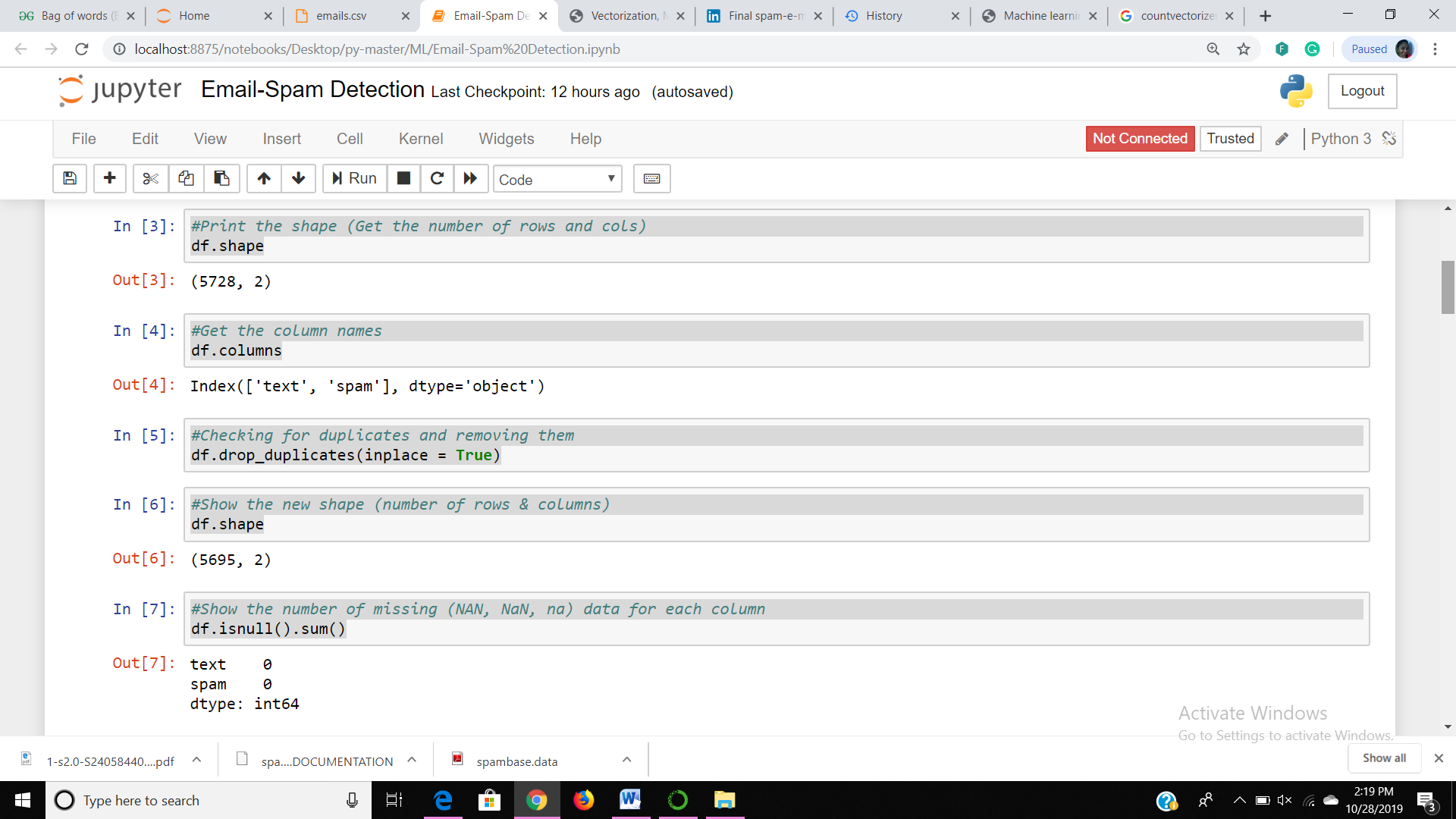
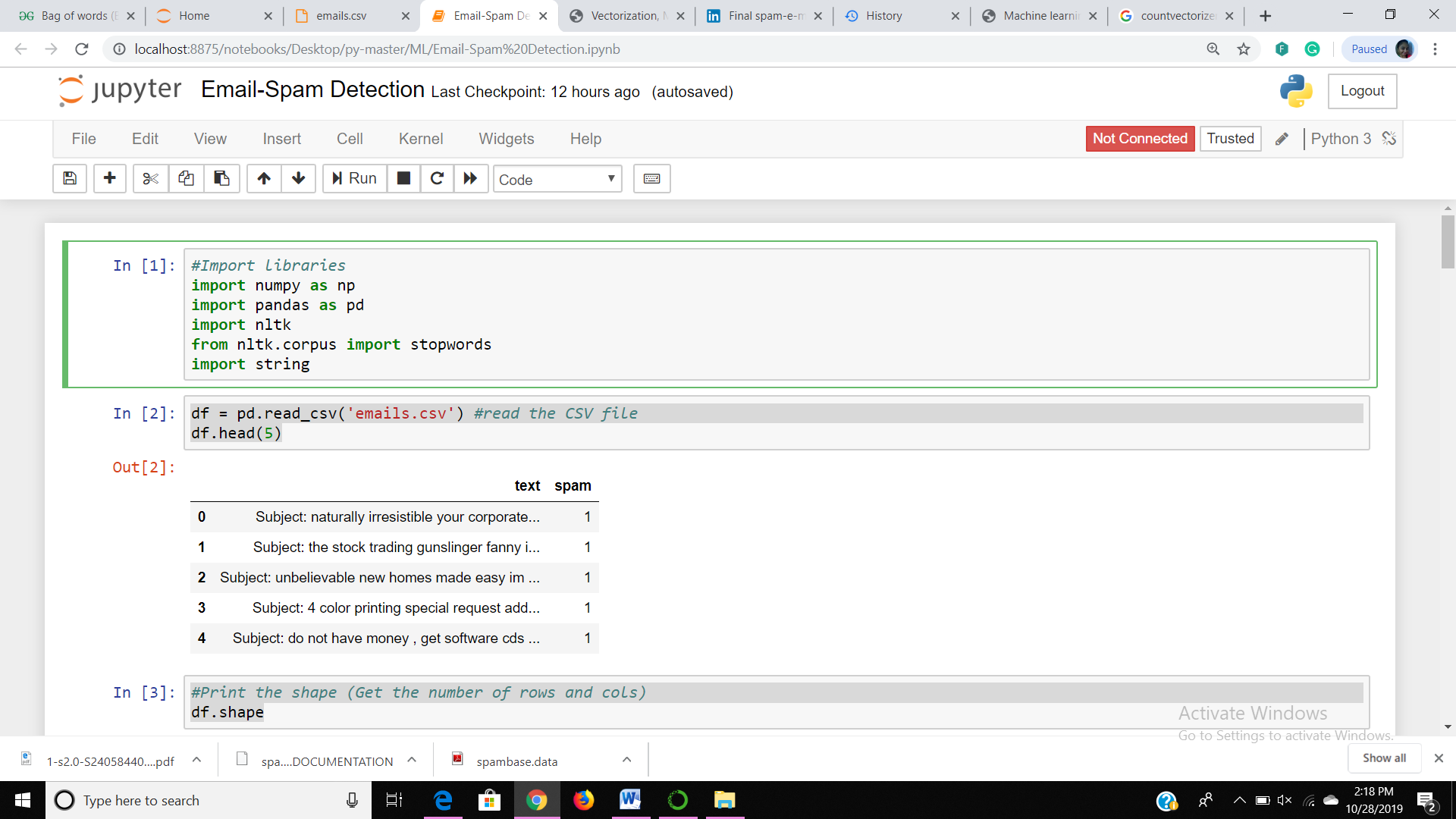
TfidfTransformer: Tf means term-frequency while tf-idf means term-frequency times inverse document-frequency. This is a common term weighting scheme in information retrieval, that has also found good use in document classification.The goal of using tf-idf instead of the raw frequencies of occurrence of a token in a given document is to scale down the impact of tokens that occur very frequently in a given corpus and that are hence empirically less informative than features that occur in a small fraction of the training corpus.

Count Vectorizer: The **CountVectorizer** provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary. Calls the fit() function in order to learn a vocabulary from one or more documents.

\

CHAPTER – 3

RESULTS



CONCLUSION

Spam is a big problem of today’s world; to solve this problem the spam classification system is created to identify the spam and non-spam mails. The spam messages are the unwanted messages which the end user clients are receiving in our daily life. Spam mails are nothing it is the advertisement of any company, any kind of virus etc.

To solve this problem create an email spam classification system and identifies the spam and non-spam mails. Here we are using the Naïve Bayesian Classifier and extracting the word using word-count algorithm. After calculation we found that naïve Bayesian classifier is more accurate. The error rate is very low .So we can say that Naïve Bayesian Classifier produce better result.

REFERENCES

[1] Sharma K. and Jatana N. (2014)“Bayesian Spam Classification: Time Efficient Radix Encoded Fragmented Database Approach” IEEE 2014 pp. 939-942.

[2] Sharma A. and Anchal (2014), "SMS Spam Detection Using Neural Network Classifier",ISSN: 2277 128X Volume 4, Issue 6, June 2014, pp. 240-244.

[3] Ali M. et al (2014), , "Multiple Classifications for Detecting Spam email by Novel Consultation Algorithm", CCECE 2014, IEEE 2014, pp. 1-5.

[4] Panigrahi P. (2012) , "A Comparative Study of Supervised Machine Learning Techniques for Spam E-Mail Filtering", Fourth International Conference on Computational Intelligence and Communication Networks, IEEE 2012, pp. 506-512.