**sLength = len(df['Date:'])**

**e = [num] \* sLength**

**e = pd.Series(e)**

**dfnew = df.assign(category=e.values)**

**print(dfnew)**

**return dfnew**

**datF = pd.DataFrame()**

**#for i in range(1):**

**datF = dfGenFun(3)**

**print(datF)**

**df.to\_csv("DataSet.csv", sep=',', encoding='utf-8')**

**Some useful links :**

[**https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/**](https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/)

[**http://bailando.sims.berkeley.edu/enron\_email.html**](http://bailando.sims.berkeley.edu/enron_email.html)

**https://datascience.stackexchange.com/questions/14983/classified-enron-email-dataset**

[**https://www.kaggle.com/wcukierski/enron-email-dataset**](https://www.kaggle.com/wcukierski/enron-email-dataset)

**http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.618.6022&rep=rep1&type=pdf**

https://appliedmachinelearning.wordpress.com/2017/01/23/email-spam-filter-python-scikit-learn/

https://towardsdatascience.com/how-i-used-machine-learning-to-classify-emails-and-turn-them-into-insights-efed37c1e66

<http://jrmeyer.github.io/machinelearning/2016/02/01/TensorFlow-Tutorial.html>

[**https://www.packtpub.com/books/content/introduction-clustering-and-unsupervised-learning**](https://www.packtpub.com/books/content/introduction-clustering-and-unsupervised-learning) **//see this site for clustering**

[**https://mubaris.com/2017/10/01/kmeans-clustering-in-python/**](https://mubaris.com/2017/10/01/kmeans-clustering-in-python/)

**from sklearn.cluster import KMeans  
  
# Number of clusters  
kmeans = KMeans(n\_clusters=3)  
# Fitting the input data  
kmeans = kmeans.fit(X)  
# Getting the cluster labels  
labels = kmeans.predict(X)  
# Centroid values  
centroids = kmeans.cluster\_centers\_**

**# Comparing with scikit-learn centroids  
print(C) # From Scratch  
print(centroids) # From sci-kit learn**

**Preprocessing:**

**1)Removing stop words**

**2)Term frequency - To choose the most frequency used words**

**3)Do LSA on the TFIDF**

**4)Clustering**

**5)Use different Clustering algorithms (Comparing multiple - to get different insights)**

**6) Cluster purification -**

**ML lab Project Review-1:**

1. **Problem Statement**: To filter the emails as spam or ham
2. **Dataset details**:
   1. Dataset name : Enron Email Data Set
   2. Attributes count and Attributes list:

Message-ID:

Date:

From:

To:

Subject:

Cc:

Mime-Version:

Content-Type:

Content-Transfer-Encoding:

Bcc:

X-From:

X-To:

X-cc:

X-bcc:

X-Folder:

X-Origin:

X-FileName:

* 1. Class count : 2(Spam, Ham)
  2. Instance count

1. **Machine Learning Techniques**
   1. Linear SVM
   2. Multinomial Naive Bayes Classifier

**Linear SVM:**

Goal**:** To separate a subset of training data from the rest called support vectors (boundary of separating hyper-plane).

SVMs are supervised binary classifiers which are very effective for a large number of features. The decision function of SVM model that predicts the class of the test data is based on support vectors and makes use of a kernel.

Support vector machine (SVM) algorithms divide the n-dimensional space representation of the data into two regions using a hyperplane. This hyperplane always maximizes the margin between the two regions or classes. The margin is defined by the longest distance between the examples of the two classes and is computed based on the distance between the closest instances of both classes to the margin, which are called supporting vectors.

**Multinomial Naive Bayes Classifier:**

Conventional and a popular method for document classification problem. It is a supervised probabilistic classifier based on Bayes theorem assuming independence between every pair of features.

To classify an instance of unknown class, the “naive” version of Bayes’s rule is used to estimate

* The probability of the instance belonging to the spam class
* The probability of it belonging to the ham class

Then it normalizes the first to the sum of both to produce a spam confidence score between 0.0 and 1.0.

1. **Design document**

**//Lets not differentiate this to architectural design and detailed design and make things complicated and confusing. What say? (by Bhavana)**

**//COP himself doesnt know d exact terminologies of these stuff...lets keep it as design //ya better**

* 1. Architectural Design

//Import Statements come here

* 1. Detailed Design- Pseudo code/Algorithm

//See whether these come here or do they go under architecture design

We have a set of emails in the dataset (both spam and ham) emails from We follow the following steps in order to filter the emails as spam or ham (non spam) :

1. Preparing the text data.

2. Creating word dictionary.

3. Feature extraction process

4. Training the classifier

Further, we will check the results on test set of the subset created.

**1.Pre-Processing ( Preparing the text data )**

Text cleaning is the first step - Words that don’t contribute to information necessary are removed. Emails may contain a lot of undesirable characters like punctuation marks, stop words, digits, etc which may not be helpful in detecting spam.

**2. Creating Word Dictionary**

A dictionary of words and their frequency is created. Once the dictionary is created, all non-words, irrelevant and absurd single characters are removed.

**3. Feature Extraction Process //i dont get this**

**//From which site, is this info obtained??**

//https://appliedmachinelearning.wordpress.com/2017/01/23/email-spam-filter-python-scikit-learn/

**//From this… got all these points from the above site itself**

**//donno...i too dint understand**

**//ok … chinmayi and chaitra see if you can understand?**

**//i too didnt get it**

Once the dictionary is ready, we can extract word count vector (our feature here) of 3000 dimensions for each email of training set. Each word count vector contains the frequency of 3000 words in the training file.

//check ??

**4. Training the classifiers**

We will train two models here namely Naive Bayes classifier and Support Vector Machines (SVM).

Once the classifiers are trained, we can check the performance of the models on test-set. We extract word count vector for each mail in test-set and predict its class(ham or spam) with the trained NB classifier and SVM model.

//check if this link is helpful for baye’s <https://hackernoon.com/how-to-build-a-simple-spam-detecting-machine-learning-classifier-4471fe6b816e>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# **Classifying emails and turn them into insights**

**Problem Statement:**

Emails can be considered as official documents in communication among users. Emails categorization can improve user efficiency in managing the inbox and help users extract data from emails. Machine-generated emails have taken more than 90% of the email traffic. It is easy to extract data from machine-generated emails as they have highly structured documents and some of them share the similar format.

1. Classifying emails into categories like – Human, career, shopping, finance, social, etc
2. To classify emails into spam or no spam emails.

**Dataset details**:

Dataset name : Enron Email Data Set

Attributes count and Attributes list:

Message-ID:

Date:

From:

To:

Subject:

Cc:

Mime-Version:

Content-Type:

Content-Transfer-Encoding:

Bcc:

X-From:

X-To:

X-cc:

X-bcc:

X-Folder:

X-Origin:

X-FileName:

Class count : 2 ( Spam, Ham) , Depending upon the insights gained from the data.

**Preprocessing:**

Each file is read , and each line is processed to categorize the data correctly into the corresponding attributes column . A data frame is created , which has a column per attribute and a row per emain file . The entire contents of the email is placed under the contents header of the column .

**Feature Extraction:**

1. **Content features include features derived from message subject and body.**

A **TFIDF, short for term frequency–inverse document frequency is created from each of the email contents .**

**TFIDF** is a numerical statistic (generally a matrix ) that is intended to reflect how important word is to a document (here each email) in a collection or corpus. To create this , we first need to find out what the top keywords are in these emails. Then only the most occuring words can be used for to build the TFIDF. Otherwise the matrix will be very sparse , causing unnecessary computations . Each row represents an email, and each column represents a word. Each cell in the matrix contains an integer between 0 and infinity which is the count of how many times a given word occured in a given email.

Also we collect a list of words that are highly correlated with Spam emails, . Give that list , we check every email for those words. This list is used to train our model , which is then used to classify the emails.

This simplifies the task to find words from the list in an email, then that email gets classified as Spam. Else the email gets classified as Ham.

**2. Address features include features extracted from the sender email address, including the subdomain ,subject etc.**

**3. Behavior features include features extracted from the sender’s and the recipients’ actions over a given message.**

**K-Means Clustering :**

K-means is a partitional clustering algorithm.

• Let the set of data points (or instances) D be {x1, x2, …, xn}, where xi = (xi1, xi2, …, xir) is a vector in a real valued space X Rr , and r is the number of attributes (dimensions) in the data.

• The k-means algorithm partitions the given data into k clusters. – Each cluster has a cluster center, called centroid. – k is specified by the user.( For spam and Ham k=2)

**Design:**

import numpy

import pandas

import sklearn

**TEAM MEMBERS**

**Bhavana Naga Sai Malepaty 01FB15ECS068**

**Brindavana Sachidanand 01FB15ECS072**

**Chaitra M V 01FB15ECS073**

**Chinmayi P S 01FB15ECS078**