**Case study – Enron email classification**

The Enron corpus is a large database of over 600,000 emails generated by 158 employees of the Enron Corporation and acquired by the Federal Energy Regulatory Commission during its investigation after the company's collapse.

We have cleaned the email headers and provided you with only sent messages body. Please see the sample below

|  |  |  |
| --- | --- | --- |
|  | file | body |
| 0 | allen-p/\_sent\_mail/1. | Here is our forecast\n\n |
| 1 | allen-p/\_sent\_mail/10. | Traveling to have a business meeting takes the... |
| 2 | allen-p/\_sent\_mail/100. | test successful.  way to go!!! |
| 3 | allen-p/\_sent\_mail/1000. | Randy,\n\n Can you send me a schedule of the s... |
| 4 | allen-p/\_sent\_mail/1001. | Let's shoot for Tuesday at 11:45. |

**Fields**

1) file : file name under mailbox directory and the hierarchy is constructed as <sender\_name>/<mailbox\_type>/<file\_number>

2) body : email body of sent messages

**Download link to dataset** - [https://drive.google.com/file/d/1yNMKT2-DoLCZMLlrAdqy6iNWQVCYq1OS/](https://nam04.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdrive.google.com%2Ffile%2Fd%2F1yNMKT2-DoLCZMLlrAdqy6iNWQVCYq1OS%2F&data=02%7C01%7Csdeepika%40vmware.com%7Cf235f06f616b4ec5715608d6a1f3d451%7Cb39138ca3cee4b4aa4d6cd83d9dd62f0%7C0%7C0%7C636874465968874760&sdata=VQDKb0u%2F8wCUZa1dX8uEUc%2FL6woHfsrmkElJTBM5Lv4%3D&reserved=0)

**Instructions**

Download the data from the link.

1) Analyse the corpus

2) Build a text processing pipeline to clean the data

3) Identify and extract top 10 senders

4) Build a classification model to predict sender given email content

5) Document your solution as

                - Solution developed

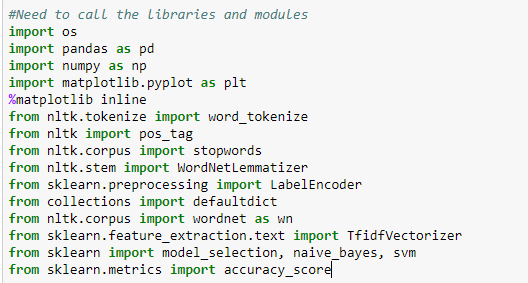
                - Process diagram (explaining all the steps you took to arrive at the solution)

                - Explain model performance

                - Visualisation to support your solution

**Solution Documentation**

Dataset given has two columns [file] & [body],Body holds the email message or text which we need to analyse and file has data in the format <sender\_name>/<mailbox\_type>/<file\_number> we will be creating separate columns for each of these values and make a new dataset , But before we need to set the working dir and import our ‘enron\_cleaned\_sen\_emails.csv’ data set to python environment like this



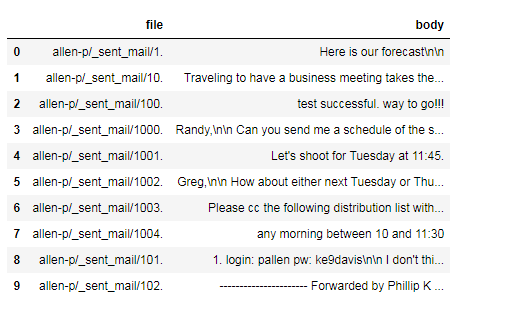
Directory set to current working dir

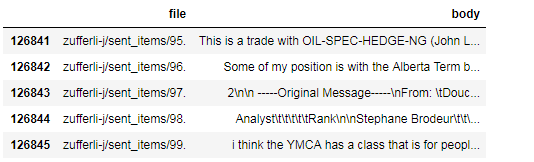




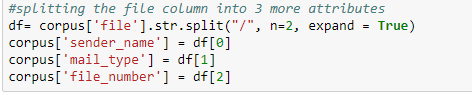
After importing the data and reading it using pandas let’s check if everything is imported properly

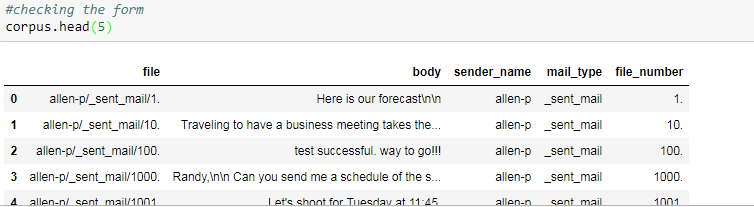




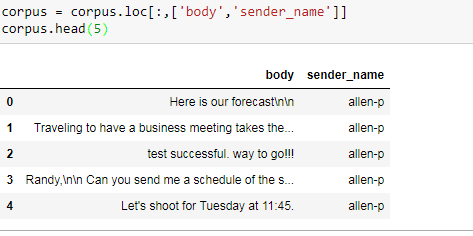


Now as discussed above let’s break the file columns into 3 different columns which will help us in using supervised machine learning algorithms.



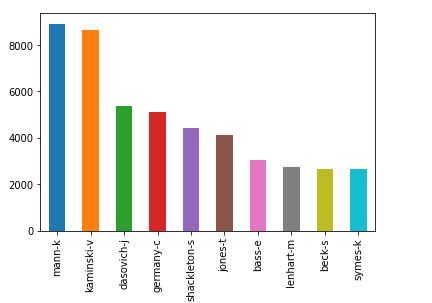
Lets check if we were able to successfully do it.

So since we want sender classification model so we will use only 2 columns going ahead [sender\_name] and [body]



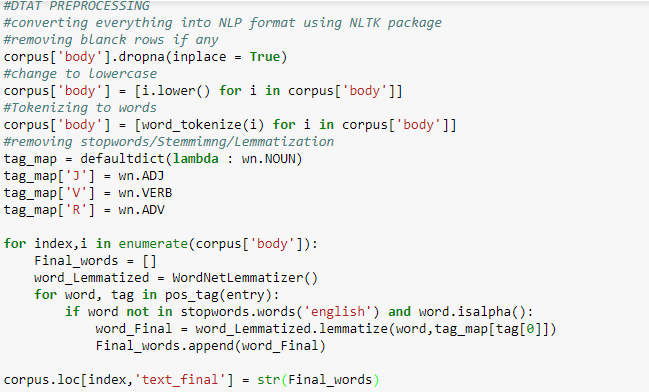
To calculate the 10 most senders name we can use the count method on [name] column and find about the **below list are the top 10 senders** also lets plot bar chart





**DATA PREPROCESSING**

Now let’s dirty our hand doing some sanity work of text data present in the corpus [body] column. For that we will take the help of wonderful package for text analytics “NLTK”. Since machine doesn’t understands the text, need to send numeric values only also in the NLP acceptance format like tokenizing the text, stemming, removing stop-words, also need to check if any missing value in [body].

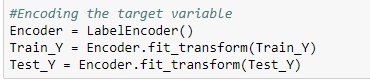


**PREPARING TRAIN AND TEST DATASET**

The Corpus will be split into two data sets, Training and Test. The training data set will be used to fit the model and the predictions will be performed on the test dataset. This can be done through the train\_test\_split from the sklearn library. The Training Data will have 70% of the corpus and Test data will have the remaining 30% as we have set the parameter test\_size=0.3 .

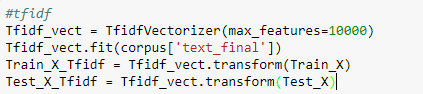


Label encoding the target variable as it will be easy for machine to learn



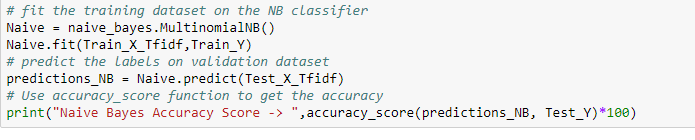
WORD VECTORIZATION

Converting collection of text documents into numeric feature vectors. We will be using TF-IDF so to get the scores assigned to each words. Finally we will transform Train\_X and Test\_X to vectorized Train\_X\_Tfidf and Test\_X\_Tfidf. These will now contain for each row a list of unique integer number and its associated importance as calculated by TF-IDF.

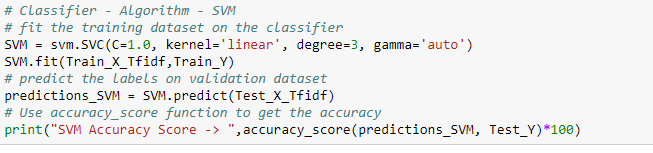


**Use the ML Algorithms to Predict the outcome**

**Naïve Bayes**

****

**SVM**

****

With enough computational power we can even use CNN/RNN(LSTM) accuracy could have been increased more but since my system is very basic config cannot do such heavy calculation.