**Open ended problem (OEP)**

**Study of different python libraries for data science and natural language processing such as nltk, scikit-learn, numpy, pandas and matploatlib.**

Python has been a charmer for data scientists for a while now. People in Data Science definitely know about the Python libraries that can be used in Data Science but when asked in an interview to name them or state its function, we often fumble up or probably not remember more than 5 libraries.

* ***Pandas***

*Pandas* is an open-source Python package that provides high-performance, easy-to-use data structures and data analysis tools for the labeled data in Python programming language. Pandas stand for *Python Data Analysis Library.*Who ever knew that?

***When to use?*** Pandas is a perfect tool for data wrangling or munging. It is designed for quick and easy data manipulation, reading, aggregation, and visualization.

Pandas take data in a CSV or TSV file or a SQL database and create a Python object with rows and columns called a data frame. The data frame is very similar to a table in statistical software, say Excel or SPSS.

***What can you do with Pandas?***

1. Indexing, manipulating, renaming, sorting, merging data frame
2. Update, Add, Delete columns from a data frame
3. Impute missing files, handle missing data or NANs
4. Plot data with histogram or box plot

This makes *Pandas* a foundation library in learning Python for Data Science.

* ***NumPy***

One of the most fundamental packages in Python, NumPy is a general-purpose array-processing package. It provides high-performance multidimensional array objects and tools to work with the arrays. NumPy is an efficient container of generic multi-dimensional data.

NumPy’s main object is the homogeneous multidimensional array. It is a table of elements or numbers of the same datatype, indexed by a tuple of positive integers. In NumPy, dimensions are called axes and the number of axes is called rank. NumPy’s array class is called ndarray aka array.

***When to use?*** NumPy is used to process arrays that store values of the same datatype. NumPy facilitates math operations on arrays and their vectorization. This significantly enhances performance and speeds up the execution time correspondingly.

***What can you do with NumPy?***

1. Basic array operations: add, multiply, slice, flatten, reshape, index arrays
2. Advanced array operations: stack arrays, split into sections, broadcast arrays
3. Work with DateTime or Linear Algebra
4. Basic Slicing and Advanced Indexing in NumPy Python

* ***Matplotlib***

You can create stories with the data visualized with Matplotlib. Another library from the SciPy Stack, Matplotlib plots 2D figures.

***When to use?***

Matplotlib is the plotting library for Python that provides an object-oriented API for embedding plots into applications. It is a close resemblance to MATLAB embedded in Python programming language.

***What can you do with Matplotlib?***

Histogram, bar plots, scatter plots, area plot to pie plot, Matplotlib can depict a wide range of visualizations. With a bit of effort and tint of visualization capabilities, with Matplotlib, you can create just any visualizations:

1. Line plots
2. Scatter plots
3. Area plots
4. Bar charts and Histograms
5. Pie charts
6. Stem plots
7. Contour plots
8. Quiver plots
9. Spectrograms

Matplotlib also facilitates labels, grids, legends, and some more formatting entities with Matplotlib. Basically, everything that can be drawn!

* ***Scikit Learn***

Introduced to the world as a Google Summer of Code project, Scikit Learn is a robust machine learning library for Python. It features ML algorithms like SVMs, random forests, k-means clustering, spectral clustering, mean shift, cross-validation and more... Even NumPy, SciPy and related scientific operations are supported by Scikit Learn with Scikit Learn being a part of the SciPy Stack.

***When to use?***

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. Supervised learning models like Naive Bayes to grouping unlabeled data such as KMeans, Scikit learn would be your go-to.

***What can you do with Scikit Learn?***

1. Classification: Spam detection, image recognition
2. Clustering: Drug response, Stock price
3. Regression: Customer segmentation, Grouping experiment outcomes
4. Dimensionality reduction: Visualization, Increased efficiency
5. Model selection: Improved accuracy via parameter tuning
6. Pre-processing: Preparing input data as a text for processing with machine learning algorithms.

Scikit Learn focuses on modeling data; not manipulating data. We have NumPy and Pandas for summarizing and manipulation.

* ***TensorFlow***

TensorFlow is an AI library that helps developers to create large-scale neural networks with many layers using data flow graphs. TensorFlow also facilitates the building of Deep Learning models, push the state-of-the-art in ML/AI and allow easy deploy of ML-powered applications.

One of the most developed websites amongst all libraries is of TensorFlow. Giants like Google, Coca-Cola, Airbnb, Twitter, Intel, DeepMind, everyone uses TensorFlow!

***When to Use?***

TensorFlow is quite efficient when it comes to classification, perception, understanding, discovering, predicting, and creating data.

***What to do with TensorFlow?***

1. Voice/Sound Recognition — IoT, Automotive, Security, UX/UI, Telecom
2. Sentiment Analysis — Mostly for CRM or CX
3. Text-Based Apps — Threat Detection, Google Translate, Gmail smart reply
4. Face Recognition — Facebook’s Deep Face, Photo tagging, Smart Unlock
5. Time Series — Recommendation from Amazon, Google, and Netflix
6. Video Detection — Motion Detection, Real-Time Threat Detection in Gaming, Security, Airports

**Implementation of Spam-or-Ham-Email-using Classification**

This is a project that collects spam messages and identifies spammers by publishing email address that humans would know not to contact but that bots might target with spam.

***Loading the Dataset***

Begin by loading the dataset emails.csv into a data frame called emails. Remember to pass the stringsAsFactors=FALSE option when loading the data.

emails = read.csv("emails.csv", stringsAsFactors=FALSE)

1. Importing libraries

# Supress Warnings

import warnings

warnings.filterwarnings('ignore')

#loading all necessary libraries

import numpy as np

import pandas as pd

import string

import collections

from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

import matplotlib.cm as cm

import matplotlib.pyplot as plt

% matplotlib inline

1. Reading the data and understanding the file

The first line of code below reads in the data as pandas data frame, while the second line prints the shape. Majority of the emails are 'ham' emails, labelled as '0', constituting 76 percent of the total data.

# loading the data file

df = pd.read\_csv('emails.csv')

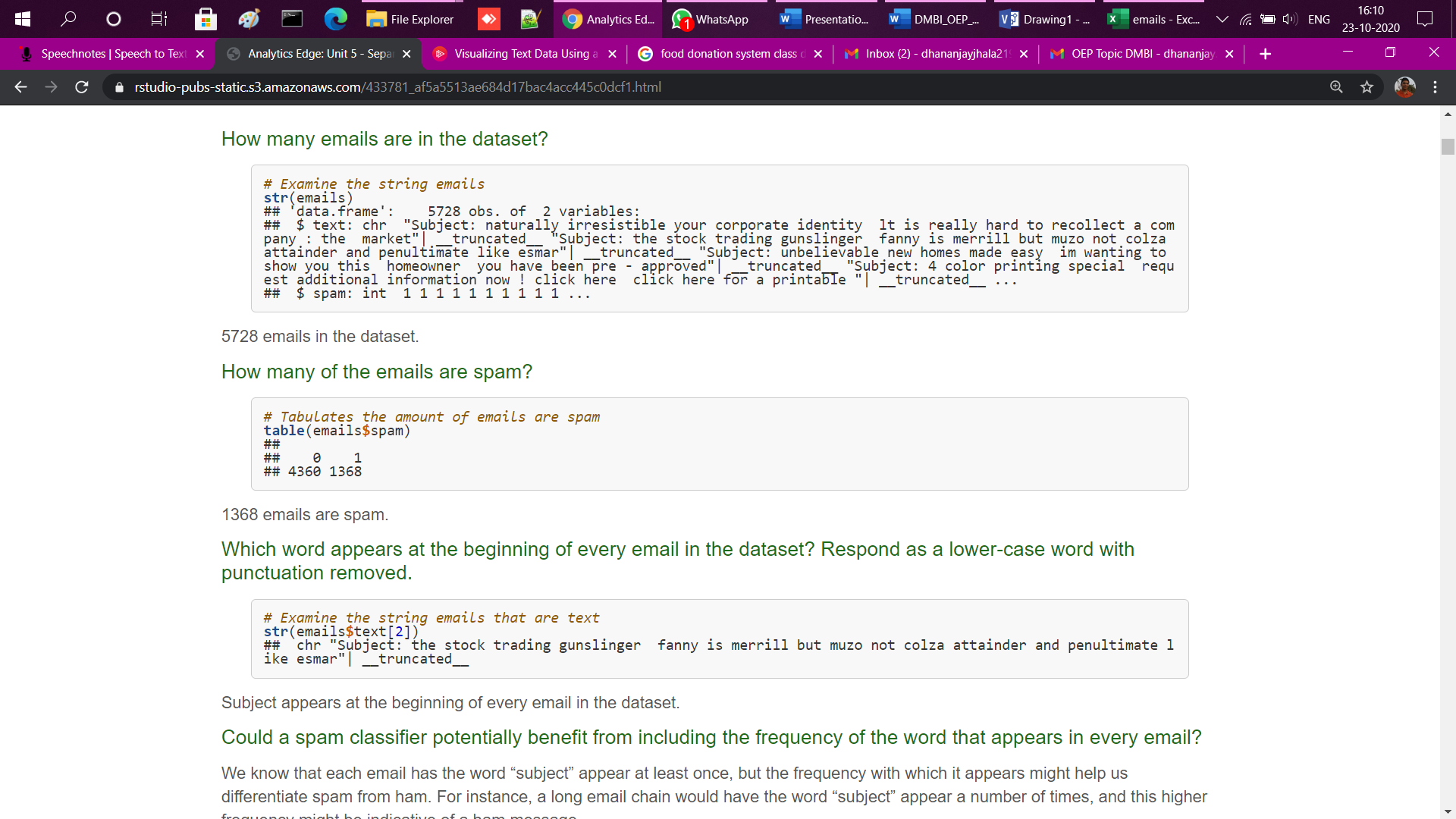
#shape of the dataframe

print('The shape of the dataframe is :',df.shape)

#first few records

df.head()

1. How many emails are there in the dataset (emails.csv)?



*5730 emails*

1. How many of the emails are spam?

*# Tabulates the amount of emails are spam*

**table**(emails**$**spam)

##

## 0 1

## 4195 1535