Python Project - Churn Emails - Dataset

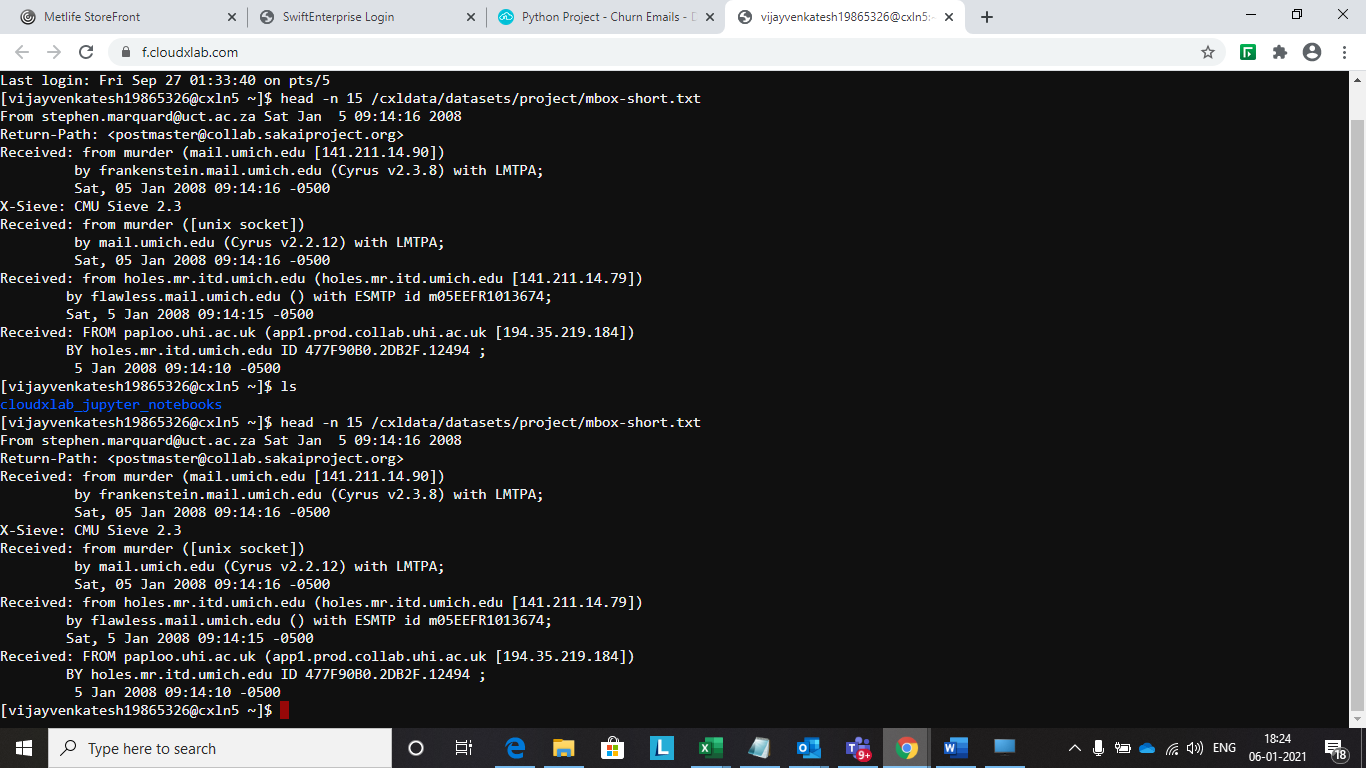
We have a text file which records mail activity from various individuals in an open source project development team. Below is the file location

/cxldata/datasets/project/mbox-short.txt

To see the first 15 lines of mbox-short.txt, please use below command on the console

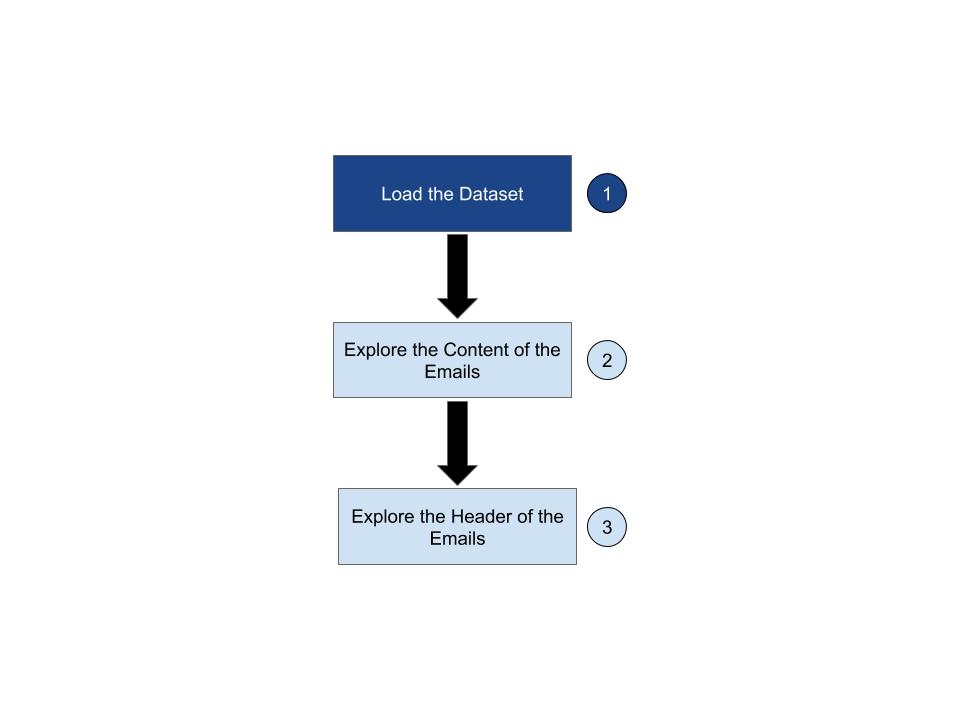
head -n 15 /cxldata/datasets/project/mbox-short.txt

These files are in a standard format for a file containing multiple mail messages. The lines which start with "From " separate the messages and the lines which start with "From:" are part of the messages. For more information about the mbox format, please see [this wikipedia article](https://en.wikipedia.org/wiki/Mbox)



# Python Project - Churn Emails - Step 1 - Load the Dataset

This is the first step where we are going to load the dataset.



# Python Project - Churn Emails - Count the Number of Lines

If we know the file is relatively small compared to the size of our main memory, we can read the whole file into one string using the read method on the file handle.

Example -

fhand = open('/cxldata/datasets/project/mbox-short.txt')

inp = fhand.read()

fhand.close()

**INSTRUCTIONS**

* Define a function number\_of\_lines
* Open the file mbox-short.txt which is located at /cxldata/datasets/project/mbox-short.txt
* Read the file into one string by using read method on file handle
* Write logic to count the number of lines
* Return the count of the number of lines

def number\_of\_lines():

fhand = open('/cxldata/datasets/project/mbox-short.txt')

inp = fhand.read()

fhand.close()

counter = 0

lines = inp.split("\n")

for i in inp:

if i=='\n':

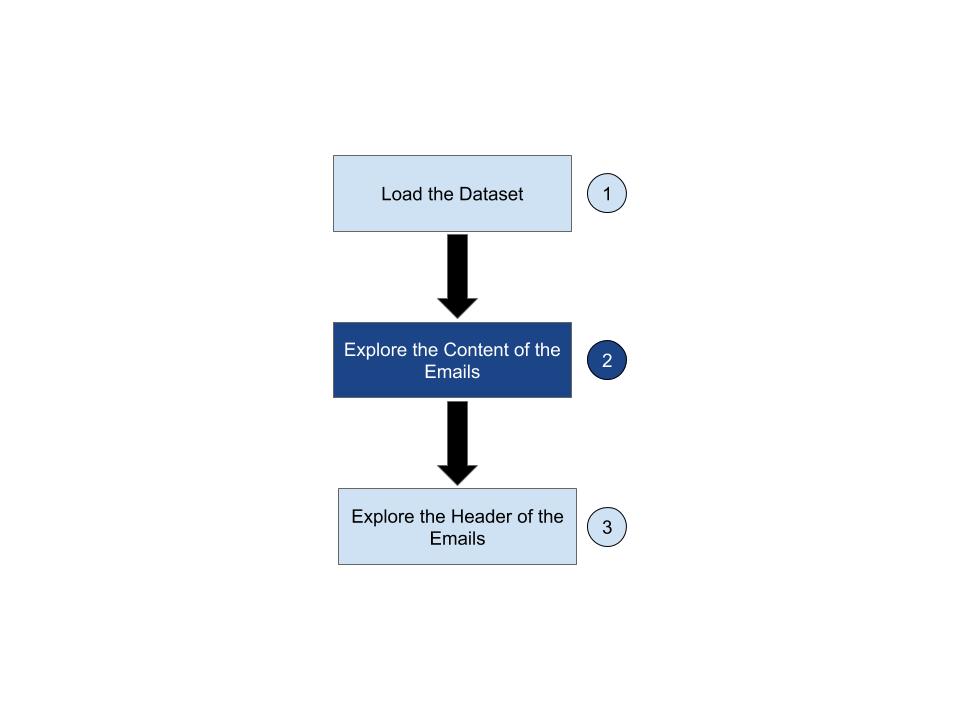
counter+=1

print(counter)

number\_of\_lines()

# Python Project - Churn Emails - Step 2 - Explore Content of Emails

Next, we would explore the contents of the emails.



Python Project - Churn Emails - Count the Number of Subject Lines

We use the string method startswith to select only those lines with the desired prefix.

The below code prints the lines starting with From:

fhand = open('/cxldata/datasets/project/mbox-short.txt')

count = 0

for line in fhand:

line = line.rstrip() # Remove new line characters from right

if line.startswith('From:'):

print(line)

**INSTRUCTIONS**

Write a function count\_number\_of\_lines which returns the count of the number of lines starting with Subject: in the file /cxldata/datasets/project/mbox-short.txt

def count\_number\_of\_lines():

fhand = open('/cxldata/datasets/project/mbox-short.txt')

count = 0

for line in fhand:

line = line.rstrip() # Remove new line characters from right

if line.startswith('Subject:'):

print(line)

count\_number\_of\_lines()

# Python Project - Churn Emails - Find Average Spam Confidence

In the previous exercise, we saw a couple of examples of startswith. Let's do one more hands-on with startswith

**INSTRUCTIONS**

* Define a function average\_spam\_confidence which calculates the average spam confidence and returns it
* Open the file mbox-short.txt which is located at /cxldata/datasets/project/mbox-short.txt
* Loop through the file handle
* Select only those lines starts with X-DSPAM-Confidence:
* Split the lines at : and take the float value which is spam confidence
* Find the average of this spam confidence in the entire file and return it.

PS - If your logic is correct then the correct spam confidence score should be 0.7507185185185187. You can use "Hint" and "See Answer" if you are stuck.

def average\_spam\_confidence():

fhand = open('/cxldata/datasets/project/mbox-short.txt')

sum = 0

count = 0

for line in fhand:

line = line.rstrip() # Remove new line characters from right

if line.startswith('X-DSPAM-Confidence:'):

line = line.split(":")

sum = sum+float(line[1])

count+=1

av = float(sum/count)

return av

average\_spam\_confidence()

# Python Project - Churn Emails - Find Which Day of the Week the Email was sent

Write a function find\_email\_sent\_days which reads the file /cxldata/datasets/project/mbox-short.txt and categorizes each mail message by which day of the week the email was sent.

To do this do the following:

* Open the file and read it line by line
* Look for lines that start with "**From**"
* For those lines which start from "**From**", then look for the **third word** and keep a running **count** of each of the **days of the week**. How do you find the day of the week, is an exercise for you.

**Note**: You have to store the results in a dictionary. Only store those day of the week that exists. For Example, if there is no line for Mon then it should not be in the dictionary elements.

* At the end of the program return the contents of your dictionary (order does not matter)

**Sample Lines from the file:**

From stephen.marquard@uct.ac.za Sat Jan 5 10:14:16 2008

From stephen.marquard@uct.ac.za Sat Jan 5 15:14:16 2008

From stephen.marquard@uct.ac.za Sun Jan 6 09:14:16 2008

**Output:**

{'Sat': 2, 'Sun': 1}

PS - If your logic is correct then your function should return this dictionary {'Sat': 1, 'Fri': 20, 'Thu': 6}. You can use "Hint" and "See Answer" if you are stuck.

def find\_email\_sent\_days():

with open("/cxldata/datasets/project/mbox-short.txt") as f:

days = [i.split(' ')[2] for i in f if i.startswith('From ')]

dic = {}

for day in days:

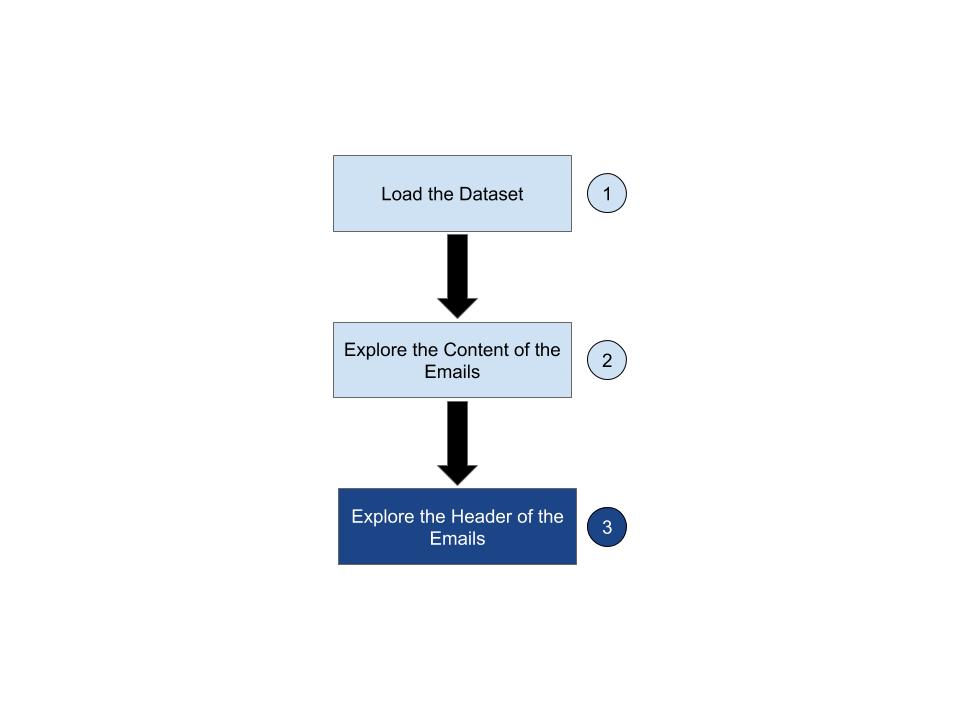
dic[day] = days.count(day)

return dic

find\_email\_sent\_days()

# Python Project - Churn Emails - Step 3 - Explore Header of Emails

And finally, in this step, we will find out about the header of the emails.



# Python Project - Churn Emails - Count Number of Messages From Each Email Address

Write a function count\_message\_from\_email which reads the file /cxldata/datasets/project/mbox-short.txt.

This function builds a histogram using a dictionary to count how many messages have come from each email address and returns the dictionary.

**Output:**

If your logic is correct then your function should return a dictionary like the following:

{'stephen.marquard@uct.ac.za': 2, 'louis@media.berkeley.edu': 3, 'zqian@umich.edu': 4, 'rjlowe@iupui.edu': 2, 'cwen@iupui.edu': 5, 'gsilver@umich.edu': 3, 'wagnermr@iupui.edu': 1, 'antranig@caret.cam.ac.uk': 1, 'gopal.ramasammycook@gmail.com': 1, 'david.horwitz@uct.ac.za': 4, 'ray@media.berkeley.edu': 1}

You can use "Hint" and "See Answer" if you are stuck.

def count\_message\_from\_email():

fhand = open('/cxldata/datasets/project/mbox-short.txt')

ecount={}

for line in fhand:

if line.startswith('From:') and '@' in line:

sp=line.split()

email=sp[1]

ecount[email]=ecount.get(email,0)+1

return ecount

count\_message\_from\_email()

# Python Project - Churn Emails - Count Number of Messages From Each Domain

Write a function count\_message\_from\_domain which reads the file /cxldata/datasets/project/mbox-short.txt.

This function builds a histogram using a dictionary to count how many messages have come from each domain(Instead of from email address), and returns the dictionary.

If your logic is correct then your function should return below dictionary

{'uct.ac.za': 6,

'media.berkeley.edu': 4,

'umich.edu': 7,

'iupui.edu': 8,

'caret.cam.ac.uk': 1,

'gmail.com': 1}

You can use "Hint" and "See Answer" if you are stuck.

def count\_message\_from\_domain():

dictionary={}

with open('/cxldata/datasets/project/mbox-short.txt')as f:

for item in f:

item=item.rstrip()

if item.startswith('From'):

lines=item.split(' ')

if(len(lines)>3):

user,domain=lines[1].split('@')

dictionary[domain]=dictionary.get(domain,0)+1

return dictionary

count\_message\_from\_domain()