Assignment 4

Name: Kesav Adithya Venkidusamy

Course: DSC650 - Big Data

Instructor: Amirfarrokh Iranitalab

In [31]:
```
import os
import json
from pathlib import Path
import zipfile
import email
from email.policy import default
from email.parser import Parser
from datetime import timezone
from collections import namedtuple

import pandas as pd
import s3fs
from bs4 import BeautifulSoup
from dateutil.parser import parse
from chardet.universaldetector import UniversalDetector

from pyspark.ml import Pipeline
from pyspark.ml.feature import CountVectorizer
from pyspark.ml.feature import HashingTF, Tokenizer
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
from pyspark.ml.pipeline import Transformer
from pyspark.sql.functions import udf
from pyspark.sql.types import StructType, StringType, StructField

import pandas as pd

current_dir = Path(os.getcwd()).absolute()
results_dir = current_dir.joinpath('results')
results_dir.mkdir(parents=True, exist_ok=True)
data_dir = current_dir.joinpath('data')
data_dir.mkdir(parents=True, exist_ok=True)
enron_data_dir = data_dir.joinpath('enron')
```

```
#enron_data_dir =  '/home/jovyan/dsc650/data/external/enron'

output_columns = [
        'payload',
        'text',
        'Message_D',
        'Date',
        'From',
        'To',
        'Subject',
        'Mime-Version',
        'Content-Type',
        'Content-Transfer-Encoding',
        'X-From',
        'X-To',
        'X-cc',
        'X-bcc',
        'X-Folder',
        'X-Origin',
        'X-FileName',
        'Cc',
        'Bcc'
]

columns = [column.replace('-', '_') for column in output_columns]

ParsedEmail = namedtuple('ParsedEmail', columns)

spark = SparkSession\
    .builder\
    .appName("Assignment04")\
    .getOrCreate()
```

The following code loads data to your local JupyterHub instance. You only need to run this once.

In [21]:
```
def copy_data_to_local():
    dst_data_path = data_dir.joinpath('enron.zip')
    dst_data_path = '/home/jovyan/dsc650/dsc650/assignments/assignment04/data/enron.zip'
    endpoint_url='https://storage.budsc.midwest-datascience.com'
    enron_data_path = 'data/external/enron.zip'

    s3 = s3fs.S3FileSystem(
        anon=True,
        client_kwargs={
            'endpoint_url': endpoint_url
        }
    )

    s3.get(enron_data_path, str(dst_data_path))

    with zipfile.ZipFile(dst_data_path) as f_zip:
        f_zip.extractall(path=data_dir)
```

In [22]:

```
## Below code has been commented as the file is present in enron folder
## So, no need to download it again
#copy_data_to_local()
```

In [26]:

```
## Copy the file from data folder to assignment data folder
! cp -a /home/jovyan/dsc650/data/external/enron/. /home/jovyan/dsc650/dsc650/assignments/assignment04/data/enron
```

This code reads emails and creates a Spark dataframe with three columns.

Assignment 4.1

In [37]:

```
def read_raw_email(email_path):
    detector = UniversalDetector()

    try:
        with open(email_path) as f:
            original_msg = f.read()
    except UnicodeDecodeError:
        detector.reset()
        with open(email_path, 'rb') as f:
            for line in f.readlines():
                detector.feed(line)
                if detector.done:
                    break
        detector.close()
        encoding = detector.result['encoding']
        with open(email_path, encoding=encoding) as f:
            original_msg = f.read()

    return original_msg
```

```
def make_spark_df():
    records = []
    for root, dirs, files in os.walk(enron_data_dir):
        for file_path in files:
            ## Declaring dictionary to store each column values
            record = {}

            ## Current path is now the file path to the current email.
            ## Use this path to read the following information
            ## original_msg
            current_path = str(Path(root).joinpath(file_path))

            ## username (Hint: It is the root folder)
            ## id (The relative path of the email message)
            record['id'] = current_path.split('/', 9)[-1]
            ## username (Hint: It is the root folder)
            record['username'] = root.rsplit('/', 2)[-2]
            ## original_msg
            record['original_msg'] = read_raw_email(current_path)

            # Append record to list
            records.append(record)

    ## TODO: Complete the code to code to create the Spark dataframe
    #return spark.createDataFrame()
    schemaString = "id username original_msg"

    #schemaString = "id_path username original_msg"
    fields = [StructField(field_name, StringType(), True) for field_name in schemaString.split()]
    schema = StructType(fields)

    # Apply the schema
    return spark.createDataFrame(records, schema)

## Calling the function and assign the results to datafram
try:
    df = make_spark_df()
except Exception as e:
    print("The dataframe creation has been failed")
else:
    print("PySpark dataframe has been created successfully")
```

```
PySpark dataframe has been created successfully
```

In [38]:

```
## Get count of dataframe
df.count()
```

Out[38]:
```
13725
```

In [39]:
```
## Showing few records from the dataframe
df.show()
```

```
+--------------------+--------+--------------------+
|                  id|username|        original_msg|
+--------------------+--------+--------------------+
|meyers-a/sent_ite...|meyers-a|Message-ID: <3165...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <1679...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <1495...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <3912...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <2536...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <1392...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <1423...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <2231...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <4681...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <4447...|
|meyers-a/sent_ite...|meyers-a|Message-ID: <2649...|
|   meyers-a/inbox/1_|meyers-a|Message-ID: <1179...|
|   meyers-a/inbox/5_|meyers-a|Message-ID: <3368...|
|  meyers-a/inbox/14_|meyers-a|Message-ID: <3227...|
|  meyers-a/inbox/21_|meyers-a|Message-ID: <1538...|
|   meyers-a/inbox/6_|meyers-a|Message-ID: <1242...|
|   meyers-a/inbox/7_|meyers-a|Message-ID: <1649...|
|  meyers-a/inbox/13_|meyers-a|Message-ID: <1811...|
|   meyers-a/inbox/8_|meyers-a|Message-ID: <2862...|
|  meyers-a/inbox/18_|meyers-a|Message-ID: <3081...|
+--------------------+--------+--------------------+
only showing top 20 rows
```

In [40]:
```
## Printing the dataframe schema
df.printSchema()
```

```
root
 |-- id: string (nullable = true)
 |-- username: string (nullable = true)
 |-- original_msg: string (nullable = true)
```

Assignment 4.2

Use `plain_msg_example` and `html_msg_example` to create a function that parses an email message.

In [41]:
```
plain_msg_example = """
Message-ID: <6742786.1075845426893.JavaMail.evans@thyme>
Date: Thu, 7 Jun 2001 11:05:33 -0700 (PDT)
From: jeffrey.hammad@enron.com
To: andy.zipper@enron.com
Subject: Thanks for the interview
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Hammad, Jeffrey </O=ENRON/OU=NA/CN=RECIPIENTS/CN=NOTESADDR/CN=CBBE377A-24F58854-862567DD-591AE7>
X-To: Zipper, Andy </O=ENRON/OU=NA/CN=RECIPIENTS/CN=AZIPPER>
X-cc:
X-bcc:
X-Folder: \Zipper, Andy\Zipper, Andy\Inbox
X-Origin: ZIPPER-A
X-FileName: Zipper, Andy.pst

Andy,

Thanks for giving me the opportunity to meet with you about the Analyst/ Associate program.  I enjoyed talking to you, an

Thanks and Best Regards,

Jeff Hammad
"""

html_msg_example = """
Message-ID: <21013632.1075862392611.JavaMail.evans@thyme>
Date: Mon, 19 Nov 2001 12:15:44 -0800 (PST)
From: insynconline.6jy5ympb.d@insync-palm.com
To: tstaab@enron.com
Subject: Last chance for special offer on Palm OS Upgrade!
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: InSync Online <InSyncOnline.6jy5ympb.d@insync-palm.com>
X-To: THERESA STAAB <tstaab@enron.com>
X-cc:
X-bcc:
```

```
X-Folder: \TSTAAB (Non-Privileged)\Staab, Theresa\Deleted Items
X-Origin: Staab-T
X-FileName: TSTAAB (Non-Privileged).pst

<html>

<html>
<head>
<title>Paprika</title>
<meta http-equiv="Content-Type" content="text/html;">
</head>
<body bgcolor="#FFFFFF" TEXT="#333333" LINK="#336699" VLINK="#6699cc" ALINK="#ff9900">
<table border="0" cellpadding="0" cellspacing="0" width="582">
<tr valign="top">
  <td width="582" colspan="9"><nobr><a href="http://insync-online.p04.com/u.d?BEReaQA5eczXB=1"><img src="http://images4.p
</tr>
<tr valign="top">
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="4" heigh
  <td width="20"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="20" height="1" hspace="0"
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?LkReaQA5eczXL=21"><img src="http://images4.postdirect.com
  <td width="20"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="20" height="1" hspace="0"
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?BkReaQA5eczXO=31"><img src="http://images4.postdirect.com
  <td width="20"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="20" height="1" hspace="0"
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?JkReaQA5eczXRs=41"><img src="http://images4.postdirect.co
  <td width="19"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="19" height="1" hspace="0"
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="4" heigh
</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="582">
<tr valign="top">
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="4" heigh
  <td width="574"><br>
    <table border="0" cellpadding="0" cellspacing="0" width="574" bgcolor="#99ccff">
    <tr>
      <td width="50"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="50" height="1" hspace
      <td width="474"><font face="verdana, arial" size="-2"color="#000000">
        <br>
        Dear THERESA,
        <br><br>
        Due to overwhelming demand for the Palm OS&#174; v4.1 Upgrade with Mobile Connectivity, we are
        extending the special offer of 25% off through November 30, 2001. So there's still time to significantly
        increase the functionality of your Palm&#153; III, IIIx, IIIxe, IIIc, V or Vx handheld. Step up to the
        new Palm OS v4.1 through this extended special offer. You'll receive the brand new Palm OS v4.1
        <b>for just $29.95 when you use Promo Code <font color="#FF0000">OS41WAVE</font></b>. That's a
        <b>$10 savings</b> off the list price.
        <br><br>
        <a href="http://insync-online.p04.com/u.d?NkReaQA5eczXRh=51">Click here to view a full product demo now</a>.
        <br><br>
        <a href="http://insync-online.p04.com/u.d?MkReaQA5eczXRm=61"><img src="http://images4.postdirect.com/master-image
        <br><br>
        You can do a lot more with your Palm&#153; handheld when you upgrade to the Palm OS v4.1. All your
        favorite features just got even better and there are some terrific new additions:
        <br><br>
        <LI> Handwrite notes and even draw pictures right on your Palm&#153 handheld</LI>
        <LI> Tap letters with your stylus and use Graffiti&#174; at the same time with the enhanced onscreen keyboard</LI
        <LI> Improved Date Book functionality lets you view, snooze or clear multiple alarms all with a single tap </LI>
        <LI> You can easily change time-zone settings</LI>

        <br><br>
        <a href="http://insync-online.p04.com/u.d?WkReaQA5eczXRb=71"><img src="http://images4.postdirect.com/master-image
        <br><br>
        <LI> <nobr>Mask/unmask</nobr> private records or hide/unhide directly within the application</LI>
        <LI> Lock your device automatically at a designated time using the new Autolocking feature</LI>
        <LI> Always remember your password with our new Hint feature*</LI>

        <br><br>
        <a href="http://insync-online.p04.com/u.d?VEReaQA5eczXRQ=81"><img src="http://images4.postdirect.com/master-image
        <br><br>
        <LI> Use your GSM compatible mobile phone or modem to get online and access the web</LI>
        <LI> Stay connected with email, instant messaging and text messaging to GSM mobile phones</LI>
        <LI> Send applications or records through your cell phone to schedule meetings and even "beam"
             important information to others</LI>

        <br><br>
        All this comes in a new operating system that can be yours for just $29.95! <a href="http://insync-online.p04.com
        upgrade to the new Palm&#153; OS v4.1</a> and you'll also get the latest Palm desktop software. Or call
        <nobr>1-800-881-7256</nobr> to order via phone.
        <br><br>
        Sincerely,<br>
        The Palm Team
        <br><br>
        P.S. Remember, this extended offer opportunity of 25% savings absolutely ends on November 30, 2001
        and is only available through the Palm Store when you use Promo Code <b><font color="#FF0000">OS41WAVE</font></b>
        <br><br>
        <img src="http://images4.postdirect.com/master-images/404707/bottom_button.gif" align="right" alt="" width="295"
        <br><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="474" height="1" hspace="0" vsp
        </font></td>
      <td width="50"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="50" height="1" hspace
    </tr>
    </table></td>
    <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="4" hei
  </tr>
  <tr>
  <td colspan="3"><img src="http://images4.postdirect.com/master-images/404707/bottom.gif" width="582" height="67" hspace
  </tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="582">
  <tr>
    <td width="54"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="54" height="1" hspace="
    <td width="474"><font face="arial, verdana" size="-2" color="#000000"><br>
    * This feature is available on the Palm&#153; IIIx, Palm&#153; IIIxe, and Palm&#153; Vx. <br><br>
    ** Note: To use the MIK functionality, you need either a Palm OS&#174; compatible modem or a phone
    with  <nobr>built-in</nobr> modem or data capability that has either an infrared port or cable exits.  If you
    are using a phone, you must have data services from your mobile service provider.  <a href="http://insync-online.p04.
    a list of tested and supported phones that you can use with the MIK. Cable not provided.
    <br><br>
    ------------------<br>
    To modify your profile or unsubscribe from Palm newsletters, <a href="http://insync-online.p04.com/u.d?KkReaQA5eczXRE
    Or, unsubscribe by replying to this message, with "unsubscribe" as the subject line of the message.
    <br><br>
    ------------------<br>
    Copyright&#169; 2001 Palm, Inc. Palm OS, Palm Computing, HandFAX, HandSTAMP, HandWEB, Graffiti,
    HotSync, iMessenger, MultiMail, Palm.Net, PalmConnect, PalmGlove, PalmModem, PalmPoint, PalmPrint,
    and the Palm Platform Compatible Logo are registered trademarks of Palm, Inc. Palm, the Palm logo,
    AnyDay, EventClub, HandMAIL, the HotSync Logo, PalmGear, PalmGlove, PalmPix, Palm Powered, the Palm
    trade dress, PalmSource, Smartcode, and Simply Palm are trademarks of Palm, Inc. All other brands and
    product names may be trademarks or registered trademarks of their respective owners.</font>
    <img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="474" height="1" hspace="0" vspace="0"
    <td width="54"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="54" height="1" hspace="
  </tr>
</table><br><br><br><br>
<!-- The following image is included for message detection -->
<img src="http://p04.com/1x1.dyn" border="0" alt="" width="1" height="1">
<img src="http://p04.com/1x1.dyn?0vEGou8Ig30ba2L2bLn" width=1 height=1></body>
</html>

</html>
"""
plain_msg_example = plain_msg_example.strip()
html_msg_example = html_msg_example.strip()
```

In [44]:

```
def parse_html_payload(payload):
    """
    This function uses Beautiful Soup to read HTML data
    and return the text.  If the payload is plain text, then
    Beautiful Soup will return the original content
    """
    soup = BeautifulSoup(payload, 'html.parser')
    return str(soup.get_text()).encode('utf-8').decode('utf-8')

def parse_email(original_msg):
    result = {}
    msg = Parser(policy=default).parsestr(original_msg)
    ## TODO: Use Python's email library to read the payload and the headers
    ## https://docs.python.org/3/library/email.examples.html

    # get email body content from parsed 'original_msg'
    payload = msg.get_payload()

    # extract text from email body 'payload'
    result['text'] = parse_html_payload(payload)

    # extract individual parts of email message
    for item in msg:
        result[item] = msg[item]

    tuple_result = tuple([str(result.get(column, None)) for column in columns])
    return ParsedEmail(*tuple_result)
```

In [49]:

```
try:
    parsed_msg = parse_email(plain_msg_example)
except Exception as e:
    print("The message parsing has been failed with below reason")
    print(e)
else:
    print("Message has been succesfully parsed")
```

```
Message has been succesfully parsed
```

In [50]:

```
print(parsed_msg.text)
```

```
Andy,

Thanks for giving me the opportunity to meet with you about the Analyst/ Associate program.  I enjoyed talking to you, an
d look forward to contributing to the success that the program has enjoyed.

Thanks and Best Regards,
Jeff Hammad
```

In [51]:

```python
try:
    parsed_html_msg = parse_email(html_msg_example)
except Exception as e:
    print("The HTML message pasrsing has been failed with below reason")
    print(e)
else:
    print("The HTML message has been successfully parsed")
```

```
The HTML message has been successfully parsed
```

In [52]:

```python
print(parsed_html_msg.text)
```

```


Paprika
```

Dear THERESA,

Due to overwhelming demand for the Palm OS® v4.1 Upgrade with Mobile Connectivity, we are extending the special offer of 25% off through November 30, 2001. So there's still time to significantly increase the functionality of your Palm™ III, IIIx, IIIxe, IIIc, V or Vx handheld. Step up to the new Palm OS v4.1 through this extended special offer. You'll receive the brand new Palm OS v4.1 for just $29.95 when you use Promo Code OS41WAVE. That's a $10 savings off the list price.

Click here to view a full product demo now.

You can do a lot more with your Palm™ handheld when you upgrade to the Palm OS v4.1. All your favorite features just got even better and there are some terrific new additions:

Handwrite notes and even draw pictures right on your Palm™ handheld
Tap letters with your stylus and use Graffiti® at the same time with the enhanced onscreen keyboard
Improved Date Book functionality lets you view, snooze or clear multiple alarms all with a single tap
You can easily change time-zone settings

Mask/unmask private records or hide/unhide directly within the application
Lock your device automatically at a designated time using the new Autolocking feature
Always remember your password with our new Hint feature*

Use your GSM compatible mobile phone or modem to get online and access the web
Stay connected with email, instant messaging and text messaging to GSM mobile phones
Send applications or records through your cell phone to schedule meetings and even "beam" important information to others

All this comes in a new operating system that can be yours for just $29.95! Click here to upgrade to the new Palm™ OS v4.1 and you'll also get the latest Palm desktop software. Or call 1-800-881-7256 to order via phone.

Sincerely,
The Palm Team

P.S. Remember, this extended offer opportunity of 25% savings absolutely ends on November 30, 2001 and is only available through the Palm Store when you use Promo Code OS41WAVE.

```
* This feature is available on the Palm™ IIIx, Palm™ IIIxe, and Palm™ Vx.
** Note: To use the MIK functionality, you need either a Palm OS® compatible modem or a phone
with  built-in modem or data capability that has either an infrared port or cable exits.  If you
are using a phone, you must have data services from your mobile service provider.  Click here for
a list of tested and supported phones that you can use with the MIK. Cable not provided.

------------------
To modify your profile or unsubscribe from Palm newsletters, click here.
Or, unsubscribe by replying to this message, with "unsubscribe" as the subject line of the message.

------------------
Copyright© 2001 Palm, Inc. Palm OS, Palm Computing, HandFAX, HandSTAMP, HandWEB, Graffiti,
HotSync, iMessenger, MultiMail, Palm.Net, PalmConnect, PalmGlove, PalmModem, PalmPoint, PalmPrint,
and the Palm Platform Compatible Logo are registered trademarks of Palm, Inc. Palm, the Palm logo,
AnyDay, EventClub, HandMAIL, the HotSync Logo, PalmGear, PalmGlove, PalmPix, Palm Powered, the Palm
trade dress, PalmSource, Smartcode, and Simply Palm are trademarks of Palm, Inc. All other brands and
product names may be trademarks or registered trademarks of their respective owners.
```

Assignment 4.3

In [53]:

```
## This creates a schema for the email data
email_struct = StructType()

for column in columns:
    email_struct.add(column, StringType(), True)
```

In [54]:
```
## This creates a user-defined function which can be used in Spark
parse_email_func = udf(lambda z: parse_email(z), email_struct)

def parse_emails(input_df):
    new_df = input_df.select(
        'username', 'id', 'original_msg', parse_email_func('original_msg').alias('parsed_email')
    )
    for column in columns:
        new_df = new_df.withColumn(column, new_df.parsed_email[column])

    new_df = new_df.drop('parsed_email')
    return new_df

class ParseEmailsTransformer(Transformer):
    def _transform(self, dataset):
        """
        Transforms the input dataset.

        :param dataset: input dataset, which is an instance of :py:class:`pyspark.sql.DataFrame`
        :returns: transformed dataset
        """
        return dataset.transform(parse_emails)
```

In [55]:
```
## Use the custom ParseEmailsTransformer, Tokenizer, and CountVectorizer
## to create a spark pipeline

# set up tokenizer for email text
tokenizer = Tokenizer(inputCol="text", outputCol="words")

# set up count vectorizer for tokenized words
cv = CountVectorizer(inputCol="words", outputCol="features",
                     vocabSize=3, minDF=2.0)
email_pipeline = Pipeline(
    ## TODO: Complete code
    stages=[ParseEmailsTransformer(), tokenizer, cv]
)
```

In [56]:
```
# model using pipeline
model = email_pipeline.fit(df)
# transform dataframe using model
result = model.transform(df)
```

```
In [57]: result.select('id', 'words', 'features').show()
```

```
+--------------------+--------------------+--------------------+
|                  id|               words|            features|
+--------------------+--------------------+--------------------+
|meyers-a/sent_ite...|[amy,, , please, ...|(3,[0,1,2],[2.0,3...|
|meyers-a/sent_ite...|[tom:, , if, it, ...|(3,[0,1,2],[5.0,1...|
|meyers-a/sent_ite...|[tom:, , i, would...|(3,[0,1,2],[33.0,...|
|meyers-a/sent_ite...|[bill:, , last, n...|(3,[0,1,2],[3.0,2...|
|meyers-a/sent_ite...|[tom:, , just, wa...|(3,[0,1,2],[19.0,...|
|meyers-a/sent_ite...|[bill:, , for, ye...|(3,[0,1,2],[8.0,6...|
|meyers-a/sent_ite...|[bill:, , please,...|(3,[0,1,2],[7.0,6...|
|meyers-a/sent_ite...|[, , -----origina...|      (3,[0],[28.0])|
|meyers-a/sent_ite...|[tom:, , buffalo,...|       (3,[0],[2.0])|
|meyers-a/sent_ite...|[kathy:, , please...| (3,[0,1],[7.0,1.0])|
|meyers-a/sent_ite...|[tom:, , earlier,...|(3,[0,1,2],[4.0,1...|
|   meyers-a/inbox/1_|[, , start, date:...|       (3,[0],[3.0])|
|   meyers-a/inbox/5_|[, , start, date:...|      (3,[0],[32.0])|
|  meyers-a/inbox/14_|[this, message, i...|(3,[0,1,2],[34.0,...|
|  meyers-a/inbox/21_|[realtime, group-...|(3,[0,1,2],[20.0,...|
|   meyers-a/inbox/6_|[, , start, date:...|      (3,[0],[32.0])|
|   meyers-a/inbox/7_|[, , , , , , ther...|(3,[0,1,2],[22.0,...|
|  meyers-a/inbox/13_|[this, message, i...|(3,[0,1,2],[2.0,2...|
|   meyers-a/inbox/8_|[, , start, date:...|      (3,[0],[32.0])|
|  meyers-a/inbox/18_|[thanks, for, you...|(3,[0,1,2],[9.0,9...|
+--------------------+--------------------+--------------------+
only showing top 20 rows
```

```
In [ ]:
```