In [1]:

```
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

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')

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 [2]:

```
#def copy_data_to_local():
    #dst_data_path = data_dir.joinpath('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)

#copy_data_to_local()
```

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

Assignment 4.1

In [3]:

```
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:
            ## Current path is now the file path to the current email.
            ## Use this path to read the following information
            ## original_msg
            ## username (Hint: It is the root folder)
            ## id (The relative path of the email message)
            current_path = Path(root).joinpath(file_path)

            #define username
            record = {}
            #assign to the root folder
            username = os.path.basename(os.path.dirname(root))
            #the relative path of the email message
            email_id = os.path.basename(os.path.dirname(current_path))
            #defining original_msg
            with open (current_path,encoding="utf8", errors='ignore') as f:
                #reading in contents of file and handling bytes conversion
                original_msg = f.read()

            #assign record fields
            record['id'] = email_id
            record['username'] = username
            record['original_msg'] = original_msg
            #append to records
            records.append(record)

    ## TODO: Complete the code to code to create the Spark dataframe
    return spark.createDataFrame(data=records)

df = make_spark_df()
```

In [4]:

```
df.show()
```

```
+--------+--------------------+--------+
|      id|        original_msg|username|
+--------+--------------------+--------+
|2_trash|Message-ID: <1774...| davis-d|
|2_trash|Message-ID: <2467...| davis-d|
|2_trash|Message-ID: <2833...| davis-d|
|2_trash|Message-ID: <1972...| davis-d|
| candis|Message-ID: <1964...| 2_trash|
| candis|Message-ID: <7345...| 2_trash|
| candis|Message-ID: <5686...| 2_trash|
| candis|Message-ID: <7218...| 2_trash|
| candis|Message-ID: <3016...| 2_trash|
| candis|Message-ID: <1233...| 2_trash|
| candis|Message-ID: <2215...| 2_trash|
| candis|Message-ID: <1365...| 2_trash|
| candis|Message-ID: <2251...| 2_trash|
| candis|Message-ID: <8556...| 2_trash|
| candis|Message-ID: <1807...| 2_trash|
| candis|Message-ID: <2705...| 2_trash|
| candis|Message-ID: <2977...| 2_trash|
| candis|Message-ID: <3065...| 2_trash|
| candis|Message-ID: <2798...| 2_trash|
| candis|Message-ID: <3108...| 2_trash|
+--------+--------------------+--------+
only showing top 20 rows
```

In [5]:

```
df.printSchema()
```

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

Assignment 4.2

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

In [6]:

```
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-24F58854-86
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 progran

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?BEReaQA5eczX
</tr>
<tr valign="top">
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/master-images/40
  <td width="20"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" w
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?LkReaQA5eczXL=21"><img src
  <td width="20"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" w
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?BkReaQA5eczXO=31"><img src
  <td width="20"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" w
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?JkReaQA5eczXRs=41"><img sr
  <td width="19"><img src="http://images4.postdirect.com/master-images/404707/clear.gif" w
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/master-images/40
</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/4
  <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.g
      <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 Connect
        extending the special offer of 25% off through November 30, 2001. So there's stil
        increase the functionality of your Palm&#153; III, IIIx, IIIxe, IIIc, V or Vx han
        new Palm OS v4.1 through this extended special offer. You'll receive the brand ne
        <b>for just $29.95 when you use Promo Code <font color="#FF0000">OS41WAVE</font><
        <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
        <br><br>
        <a href="http://insync-online.p04.com/u.d?MkReaQA5eczXRm=61"><img src="http://ima
        <br><br>
        You can do a lot more with your Palm&#153; handheld when you upgrade to the Palm
        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 th
        <LI> Improved Date Book functionality lets you view, snooze or clear multiple ala
        <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://ima
        <br><br>
        <LI> <nobr>Mask/unmask</nobr> private records or hide/unhide directly within the
        <LI> Lock your device automatically at a designated time using the new Autolockin
        <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://ima
        <br><br>
        <LI> Use your GSM compatible mobile phone or modem to get online and access the w
        <LI> Stay connected with email, instant messaging and text messaging to GSM mobil
        <LI> Send applications or records through your cell phone to schedule meetings an
             important information to others</LI>

        <br><br>
        All this comes in a new operating system that can be yours for just $29.95! <a hr
        upgrade to the new Palm&#153; OS v4.1</a> and you'll also get the latest Palm des
        <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
        and is only available through the Palm Store when you use Promo Code <b><font col
        <br><br>
        <img src="http://images4.postdirect.com/master-images/404707/bottom_button.gif" a
        <br><img src="http://images4.postdirect.com/master-images/404707/clear.gif" width=
        </font></td>
      <td width="50"><img src="http://images4.postdirect.com/master-images/404707/clear.g
    </tr>
    </table></td>
    <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/master-images/
  </tr>
  <tr>
```

```
    <td colspan="3"><img src="http://images4.postdirect.com/master-images/404707/bottom.gif'
  </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'
    <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; V
    ** Note: To use the MIK functionality, you need either a Palm OS&#174; compatible mode
    with  <nobr>built-in</nobr> modem or data capability that has either an infrared port
    are using a phone, you must have data services from your mobile service provider.  <a
    a list of tested and supported phones that you can use with the MIK. Cable not provide
    <br><br>
    ------------------<br>
    To modify your profile or unsubscribe from Palm newsletters, <a href="http://insync-or
    Or, unsubscribe by replying to this message, with "unsubscribe" as the subject line of
    <br><br>
    ------------------<br>
    Copyright&#169; 2001 Palm, Inc. Palm OS, Palm Computing, HandFAX, HandSTAMP, HandWEB,
    HotSync, iMessenger, MultiMail, Palm.Net, PalmConnect, PalmGlove, PalmModem, PalmPoint
    and the Palm Platform Compatible Logo are registered trademarks of Palm, Inc. Palm, th
    AnyDay, EventClub, HandMAIL, the HotSync Logo, PalmGear, PalmGlove, PalmPix, Palm Powe
    trade dress, PalmSource, Smartcode, and Simply Palm are trademarks of Palm, Inc. All o
    product names may be trademarks or registered trademarks of their respective owners.</
    <img src="http://images4.postdirect.com/master-images/404707/clear.gif" width="474" he
    <td width="54"><img src="http://images4.postdirect.com/master-images/404707/clear.gif'
  </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 [7]:

```
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):

    #check if html
    if bool (BeautifulSoup(original_msg,"html.parser").find()):
        #remove html formatting if yes
        original_msg = parse_html_payload(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

    #assign variables for result dictionary
    result.update({'payload':msg.get_payload()})
    result.update({'text':msg.get_content()})
    result.update({'Message_D':msg['Message-ID']})
    result.update({'Date':msg['Date']})
    result.update({'From':msg['From']})
    result.update({'To':msg['To']})
    result.update({'Subject':msg['Subject']})
    result.update({'Mime-Version':msg['Mime-Version']})
    result.update({'Content-Type':msg['Content-Type']})
    result.update({'Content-Transfer-Encoding':msg['Content-Transfer-Encoding']})
    result.update({'X-From':msg['X-From']})
    result.update({'X-To':msg['X-To']})
    result.update({'X-cc':msg['X-cc']})
    result.update({'X-bcc':msg['X-bcc']})
    result.update({'X-Folder':msg['X-Folder']})
    result.update({'X-Origin':msg['X-Origin']})
    result.update({'X-FileName':msg['X-FileName']})
    result.update({'Cc':msg['Cc']})
    result.update({'Bcc':msg['Bcc']})

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

In [8]:

```
parsed_msg = parse_email(plain_msg_example)
```

In [9]:

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

```
Andy,

Thanks for giving me the opportunity to meet with you about the Analyst/ Associate progra
m.  I enjoyed talking to you, and look forward to contributing to the success that the pro
gram has enjoyed.

Thanks and Best Regards,

Jeff Hammad
```

In [10]:

```
parsed_html_msg = parse_email(html_msg_example)
```

In [11]:

```
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\u2122 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\u2122 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\u2122 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\u2122 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\u2122 IIIx, Palm\u2122 IIIxe, and Palm\u2122 V
x.
    ** 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 exit
s.  If you
    are using a phone, you must have data services from your mobile service provider.  Cli
ck here for
    a list of tested and supported phones that you can use with the MIK. Cable not provide
d.

    ------------------
    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, Graf
fiti,
    HotSync, iMessenger, MultiMail, Palm.Net, PalmConnect, PalmGlove, PalmModem, PalmPoin
t, PalmPrint,
    and the Palm Platform Compatible Logo are registered trademarks of Palm, Inc. Palm, th
e Palm logo,
    AnyDay, EventClub, HandMAIL, the HotSync Logo, PalmGear, PalmGlove, PalmPix, Palm Powe
red, the Palm
    trade dress, PalmSource, Smartcode, and Simply Palm are trademarks of Palm, Inc. All o
ther brands and
    product names may be trademarks or registered trademarks of their respective owners.
```

Assignment 4.3

In [12]:

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

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

In [73]:

```python
## 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_e
    )
    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.Data
        :returns: transformed dataset
        """
        return dataset.transform(parse_emails)

## Use the custom ParseEmailsTransformer, Tokenizer, and CountVectorizer
## to create a spark pipeline

#instantiate tokenizer
tokenizer = Tokenizer(inputCol="text", outputCol="words")

# Create a Vectorizer Object
vectorizer = CountVectorizer(inputCol='words',outputCol='features',vocabSize=3,minDF=2.0)

#create ParseEmailsTransformer object
email_transformer = ParseEmailsTransformer()

#include stages with objects
email_pipeline = Pipeline(stages=[email_transformer,tokenizer,vectorizer])
model = email_pipeline.fit(df)
result = model.transform(df)
```

In [74]:

```
result.select('id', 'words', 'features').show()
```

```
+-------------+--------------------+--------------------+
|           id|               words|            features|
+-------------+--------------------+--------------------+
|deleted_items|[god,, don't, bre...|(3,[0,1,2],[71.0,...|
|deleted_items|[, i, think, they...|(3,[0,1,2],[8.0,1...|
|deleted_items|[, , , -----origi...|(3,[0,1,2],[13.0,...|
|deleted_items|[, i'm, blowin', ...|(3,[0,1,2],[15.0,...|
|deleted_items|[, , as, we, near...|(3,[0,1,2],[10.0,...|
|deleted_items|[just, an, fyi..,...| (3,[1,2],[2.0,1.0])|
|deleted_items|[allrighty, i, gu...|(3,[0,1,2],[5.0,9...|
|deleted_items|[we, are, trying,...|(3,[0,1,2],[83.0,...|
|deleted_items|[scientech's, iss...|(3,[0,1,2],[131.0...|
|deleted_items|[joe,, , attached...|(3,[0,1,2],[15.0,...|
|deleted_items|[you, can, do, wh...|(3,[0,1,2],[70.0,...|
|deleted_items|[, late, on, octo...|(3,[0,1,2],[6.0,4...|
|deleted_items|[, , , , , , , , ...|(3,[0,1,2],[70.0,...|
|deleted_items|[no, meeting., co...|       (3,[0],[7.0])|
|deleted_items|[looking, for, $1...|       (3,[0],[4.0])|
|deleted_items|[, as, usual........|(3,[0,1,2],[22.0,...|
|deleted_items|[ps., , , your, s...|(3,[0,1,2],[4.0,4...|
|deleted_items|[, , learn, techn...|(3,[0,1,2],[55.0,...|
|deleted_items|[the, question, w...|(3,[0,1,2],[248.0...|
|deleted_items|[as, you, know,, ...|(3,[0,1,2],[21.0,...|
+-------------+--------------------+--------------------+
only showing top 20 rows
```

In [ ]: