In [24]:
```
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
import shutil
```

In [26]:
```
from pyspark.sql import functions
```

In [2]:

```
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_ID',
        '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 [3]:

```
def move_data_to_local():
    dst_data_path = data_dir.joinpath('enron.zip')
    src_data_path = "C:/Users/bibek/Documents/GitHub/dsc650/data/externa
    shutil.copy(src_data_path, dst_data_path)

    with zipfile.ZipFile(src_data_path) as f_zip:
        f_zip.extractall(path=enron_data_dir)

move_data_to_local()
```

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

Assignment 4.1

In [4]:

```
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.
            current_path = Path(root).joinpath(file_path)

            # get original message
            original_msg = read_raw_email(current_path)

            # get username
            path = os.path.normpath(root)
            x = path.split(os.sep)
            username = x[11]

            # get relative path
            relative_path = os.path.relpath(current_path, enron_data_dir
            id = relative_path

            records.append((id,username,original_msg))

    # PySpark dataframe
    df = spark.createDataFrame(records).toDF("id","username","original_m
    return df
```

In [5]:

```
df = make_spark_df()
```

In [6]:

```
df.printSchema()
```

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

In [7]:

```
df.show()
```

```
+--------------------+--------+--------------------+
|                  id|username|        original_msg|
+--------------------+--------+--------------------+
|  davis-d\2_trash\1_| davis-d|Message-ID: <1774...|
|  davis-d\2_trash\2_| davis-d|Message-ID: <2467...|
|  davis-d\2_trash\3_| davis-d|Message-ID: <2833...|
|  davis-d\2_trash\4_| davis-d|Message-ID: <1972...|
|davis-d\2_trash\c...| davis-d|Message-ID: <1964...|
|davis-d\2_trash\c...| davis-d|Message-ID: <7345...|
|davis-d\2_trash\c...| davis-d|Message-ID: <5686...|
|davis-d\2_trash\c...| davis-d|Message-ID: <7218...|
|davis-d\2_trash\c...| davis-d|Message-ID: <3016...|
|davis-d\2_trash\c...| davis-d|Message-ID: <1233...|
|davis-d\2_trash\c...| davis-d|Message-ID: <2215...|
|davis-d\2_trash\c...| davis-d|Message-ID: <1365...|
|davis-d\2_trash\c...| davis-d|Message-ID: <2251...|
|davis-d\2_trash\c...| davis-d|Message-ID: <8556...|
|davis-d\2_trash\c...| davis-d|Message-ID: <1807...|
|davis-d\2_trash\c...| davis-d|Message-ID: <2705...|
|davis-d\2_trash\c...| davis-d|Message-ID: <2977...|
|davis-d\2_trash\c...| davis-d|Message-ID: <3065...|
|davis-d\2_trash\c...| davis-d|Message-ID: <2798...|
|davis-d\2_trash\c...| davis-d|Message-ID: <3108...|
+--------------------+--------+--------------------+
only showing top 20 rows
```

In [7]:

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.master("local").getOrCreate()
print(spark.sparkContext.version)
```

```
2.4.5
```

Assignment 4.2

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

In [8]:

```
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=CB
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/

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" AL
<table border="0" cellpadding="0" cellspacing="0" width="582">
<tr valign="top">
  <td width="582" colspan="9"><nobr><a href="http://insync-online.p04.co
</tr>
<tr valign="top">
```

```
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.co
  <td width="20"><img src="http://images4.postdirect.com/master-images/4
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?LkReaQA5
  <td width="20"><img src="http://images4.postdirect.com/master-images/4
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?BkReaQA5
  <td width="20"><img src="http://images4.postdirect.com/master-images/4
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?JkReaQA5
  <td width="19"><img src="http://images4.postdirect.com/master-images/4
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.co
</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="582">
<tr valign="top">
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.co
  <td width="574"><br>
    <table border="0" cellpadding="0" cellspacing="0" width="574" bgcolo
    <tr>
      <td width="50"><img src="http://images4.postdirect.com/master-imag
      <td width="474"><font face="verdana, arial" size="-2"color="#00000
        <br>
        Dear THERESA,
        <br><br>
        Due to overwhelming demand for the Palm OS&#174; v4.1 Upgrade wi
        extending the special offer of 25% off through November 30, 2001
        increase the functionality of your Palm&#153; III, IIIx, IIIxe,
        new Palm OS v4.1 through this extended special offer. You'll rec
        <b>for just $29.95 when you use Promo Code <font color="#FF0000"
        <b>$10 savings</b> off the list price.
        <br><br>
        <a href="http://insync-online.p04.com/u.d?NkReaQA5eczXRh=51">Cli
        <br><br>
        <a href="http://insync-online.p04.com/u.d?MkReaQA5eczXRm=61"><im
        <br><br>
        You can do a lot more with your Palm&#153; handheld when you upg
        favorite features just got even better and there are some terrif
        <br><br>
        <LI> Handwrite notes and even draw pictures right on your Palm&#
        <LI> Tap letters with your stylus and use Graffiti&#174; at the
        <LI> Improved Date Book functionality lets you view, snooze or c
        <LI> You can easily change time-zone settings</LI>

        <br><br>
        <a href="http://insync-online.p04.com/u.d?WkReaQA5eczXRb=71"><im
        <br><br>
        <LI> <nobr>Mask/unmask</nobr> private records or hide/unhide dir
        <LI> Lock your device automatically at a designated time using t
        <LI> Always remember your password with our new Hint feature*</L

        <br><br>
        <a href="http://insync-online.p04.com/u.d?VEReaQA5eczXRQ=81"><im
        <br><br>
        <LI> Use your GSM compatible mobile phone or modem to get online
        <LI> Stay connected with email, instant messaging and text messa
        <LI> Send applications or records through your cell phone to sch
            important information to others</LI>

        <br><br>
        All this comes in a new operating system that can be yours for j
        upgrade to the new Palm&#153; OS v4.1</a> and you'll also get th
        <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 ab
        and is only available through the Palm Store when you use Promo 
        <br><br>
        <img src="http://images4.postdirect.com/master-images/404707/bot
        <br><img src="http://images4.postdirect.com/master-images/404707
        </font></td>
      <td width="50"><img src="http://images4.postdirect.com/master-imag
    </tr>
    </table></td>
    <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.
  </tr>
  <tr>
  <td colspan="3"><img src="http://images4.postdirect.com/master-images/
  </tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="582">
  <tr>
    <td width="54"><img src="http://images4.postdirect.com/master-images
    <td width="474"><font face="arial, verdana" size="-2" color="#000000
    * This feature is available on the Palm&#153; IIIx, Palm&#153; IIIxe
    ** Note: To use the MIK functionality, you need either a Palm OS&#17
    with  <nobr>built-in</nobr> modem or data capability that has either
    are using a phone, you must have data services from your mobile serv
    a list of tested and supported phones that you can use with the MIK.
    <br><br>
    ------------------<br>
    To modify your profile or unsubscribe from Palm newsletters, <a href
    Or, unsubscribe by replying to this message, with "unsubscribe" as t
    <br><br>
    ------------------<br>
    Copyright&#169; 2001 Palm, Inc. Palm OS, Palm Computing, HandFAX, Ha
    HotSync, iMessenger, MultiMail, Palm.Net, PalmConnect, PalmGlove, Pa
    and the Palm Platform Compatible Logo are registered trademarks of P
    AnyDay, EventClub, HandMAIL, the HotSync Logo, PalmGear, PalmGlove, 
    trade dress, PalmSource, Smartcode, and Simply Palm are trademarks o
    product names may be trademarks or registered trademarks of their re
    <img src="http://images4.postdirect.com/master-images/404707/clear.g
    <td width="54"><img src="http://images4.postdirect.com/master-images
  </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><
</html>

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

In [9]:
```python
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):

    if bool(BeautifulSoup(original_msg, "html.parser").find()):
        # convert HTML to plain email message
        original_msg = parse_html_payload(original_msg)

    result = {}
    email = Parser(policy=default).parsestr(original_msg) # email parser
    # get values for each field
    result['text']= email.get_content()
    result['payload'] = email.get_payload()
    for key, value in email.items():
        result[key] =  email[key]

    # Convert the dictionary to a tuple and return the tuple
    tuple_result = tuple([str(result.get(column, None)) for column in co
    return ParsedEmail(*tuple_result)
```

In [10]:
```python
parsed_msg = parse_email(plain_msg_example)
print(parsed_msg.text)
```

```
Andy,

Thanks for giving me the opportunity to meet with you about the Analyst/ As
sociate program.  I enjoyed talking to you, and look forward to contributin
g to the success that the program has enjoyed.

Thanks and Best Regards,

Jeff Hammad
```

In [11]:
```
parsed_html_msg = parse_email(html_msg_example)
print(parsed_html_msg.text)
```

```


        You can do a lot more with your Palm\u2122 handheld when you upgr
ade to the Palm OS v4.1. All your
        favorite features just got even better and there are some terrifi
c 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 th
e 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 applicati
on
 Lock your device automatically at a designated time using the new Autolo
cking feature
```

Functions to read all example emails from path: '.../dsc650/assignments/assignment04/examples'

In [12]:
```
def get_email_paths(directory):
    pathlist = []
    for root, dirs, files in os.walk(directory):
        for file_path in files:
            ## Current path is now the file path to the current email.
            current_path = Path(root).joinpath(file_path)
            pathlist.append(str(current_path))
    return pathlist
```

In [13]:
```
def get_email_text(i):
    original_msg = read_raw_email(i)
    original_msg = original_msg.strip()
    parsed_msg = parse_email(original_msg)
    text = parsed_msg.text
    return text
```

In [14]:
```
examples_dir = current_dir.joinpath('examples')
examples_dir.mkdir(parents=True, exist_ok=True)
```

In [15]:
```
pathlist = get_email_paths(examples_dir)
for item in pathlist:
    text = get_email_text(item)
    print(text)
```

```
4 ROUND-TRIP AIR TICKETS - ABSOLUTELY FREE
 to exciting destinations across the U.S., Caribbean,
 Hawaii or Mexico. Take the family or take two trips!
 And except for your hotel stay, there's NO OBLIGATION
 and NO PURCHASE REQUIRED for your trip. Imagine
 flying your family FREE to Florida, California,
 Las Vegas, Jamaica, Aruba, Bahamas, or other exciting
 vacation paradise! Plus, you get a 30-day FREE
 trial in Preferred Traveller, where you'll save big
 whenever and wherever you travel,
 dine or entertain!
Act now! This FREE offer is good only for a limited
 time!




 The 4 FREE AIRLINE TICKETS are yours to keep,
 just for registering! Plus, your FREE TICKETS
```

Assignment 4.3

In [16]:
```
## This creates a schema for the email data
email_struct = StructType()

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

In [35]:
```
## 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
    )
    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
        :returns: transformed dataset
        """
        return dataset.transform(parse_emails)

## Use the custom ParseEmailsTransformer, Tokenizer, and CountVectorizer
## to create a spark pipeline
email_pipeline = Pipeline(stages=[ParseEmailsTransformer(),
                                  Tokenizer(inputCol='text',
                                            outputCol='words'),
                                  CountVectorizer(inputCol='words',
                                                  outputCol='features',
                                                  vocabSize=3)])
```

In [37]:
```
model = email_pipeline.fit(df)
```

In [38]:
```
result = model.transform(df)
```

In [78]:
```
result.select('id', 'words', 'features').show()
```

```
+--------------------+--------------------+--------------------+
|                  id|               words|            features|
+--------------------+--------------------+--------------------+
|  davis-d\2_trash\1_|[, >, , , , , >, ...|(3,[0,1,2],[697.0...|
|  davis-d\2_trash\2_|[fyi..., thanks.,...|(3,[0,1,2],[57.0,...|
|  davis-d\2_trash\3_|[----------------...|(3,[0,1,2],[118.0...|
|  davis-d\2_trash\4_|[-----original, m...|(3,[0,2],[17.0,1.0])|
|davis-d\2_trash\c...|[hi, mommy!, , ye...|(3,[0,1,2],[4.0,1...|
|davis-d\2_trash\c...|[hey, sweetie,, ,...| (3,[0,1],[8.0,1.0])|
|davis-d\2_trash\c...|[----------------...|      (3,[0],[41.0])|
|davis-d\2_trash\c...|[----------------...|(3,[0,1,2],[65.0,...|
|davis-d\2_trash\c...|[----------------...|(3,[0,2],[58.0,1.0])|
|davis-d\2_trash\c...|[----------------...|(3,[0,1,2],[29.0,...|
|davis-d\2_trash\c...|[----------------...|(3,[0,1,2],[18.0,...|
|davis-d\2_trash\c...|[----------------...|      (3,[0],[43.0])|
|davis-d\2_trash\c...|[, , , , --------...|      (3,[0],[11.0])|
|davis-d\2_trash\c...|[----------------...|      (3,[0],[14.0])|
|davis-d\2_trash\c...|[are, you, on, th...| (3,[0,1],[1.0,1.0])|
|davis-d\2_trash\c...|[listen, girly!, ...| (3,[0,2],[9.0,1.0])|
|davis-d\2_trash\c...|[candis, all, you...|(3,[0,1,2],[16.0,...|
|davis-d\2_trash\c...|[what, is, your, ...| (3,[0,2],[1.0,1.0])|
|davis-d\2_trash\c...|[candis, -, , why...|       (3,[0],[3.0])|
|davis-d\2_trash\c...|[----------------...|      (3,[0],[16.0])|
+--------------------+--------------------+--------------------+
only showing top 20 rows
```

In [ ]:
```

```