In [1]:

```
import os
from google.colab import drive
drive.mount('/content/drive', force_remount = True)
os.chdir('/content/drive/My Drive/assignment04')
!pwd
```

```
Mounted at /content/drive
/content/drive/My Drive/assignment04
```

In [2]:

```
!pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simpl
e,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-
python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: pyspark in /usr/local/lib/python3.9/dis
t-packages (3.3.2)
Requirement already satisfied: py4j==0.10.9.5 in /usr/local/lib/python
3.9/dist-packages (from pyspark) (0.10.9.5)
```

In [3]:

```
! pip install chardet
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simpl
e,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-
python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: chardet in /usr/local/lib/python3.9/dis
t-packages (4.0.0)
```

In [4]:
```
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
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, StructField, 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()
```

Assignment 4.1

In [5]:
```
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)

            ## Use this path to read the following information
            record = {}

            # get id (The relative path of the email message)
            record['id'] = os.path.relpath(current_path, os.path.dirnam

            # get username (Hint: It is the root folder)
            record['username'] = root.rsplit('/', 2)[-2]

            # get original message
            record['original_msg'] = read_raw_email(str(current_path))

            records.append(record)

    ## TODO: Complete the code to create the Spark dataframe
    email_schema = StructType([
        StructField("id", StringType(), True),
        StructField("username", StringType(), True),
        StructField("original_msg", StringType(), True)
    ])

    return spark.createDataFrame(records, email_schema)

df = make_spark_df()
```

In [6]:
```
df.show()
```

```
+--------------------+--------+--------------------+
|                  id|username|        original_msg|
+--------------------+--------+--------------------+
|davis-d/all_docum...| davis-d|Message-ID: <1098...|
|davis-d/all_docum...| davis-d|Message-ID: <5201...|
|davis-d/all_docum...| davis-d|Message-ID: <8786...|
|davis-d/all_docum...| davis-d|Message-ID: <1957...|
|davis-d/all_docum...| davis-d|Message-ID: <7671...|
|davis-d/all_docum...| davis-d|Message-ID: <1404...|
|davis-d/all_docum...| davis-d|Message-ID: <1086...|
|davis-d/all_docum...| davis-d|Message-ID: <1756...|
|davis-d/all_docum...| davis-d|Message-ID: <4468...|
|davis-d/all_docum...| davis-d|Message-ID: <7842...|
|davis-d/all_docum...| davis-d|Message-ID: <5659...|
|davis-d/all_docum...| davis-d|Message-ID: <2923...|
|davis-d/all_docum...| davis-d|Message-ID: <2644...|
|davis-d/all_docum...| davis-d|Message-ID: <8150...|
|davis-d/all_docum...| davis-d|Message-ID: <1163...|
|davis-d/all_docum...| davis-d|Message-ID: <1214...|
|davis-d/all_docum...| davis-d|Message-ID: <1173...|
|davis-d/all_docum...| davis-d|Message-ID: <1230...|
|davis-d/all_docum...| davis-d|Message-ID: <2728...|
|davis-d/all_docum...| davis-d|Message-ID: <2689...|
+--------------------+--------+--------------------+
only showing top 20 rows
```

In [7]:
```
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 [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=CI
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" A
<table border="0" cellpadding="0" cellspacing="0" width="582">
<tr valign="top">
  <td width="582" colspan="9"><nobr><a href="http://insync-online.p04.c
</tr>
<tr valign="top">
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.c
  <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 terri
        <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*</

        <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 tl
        <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 al
        and is only available through the Palm Store when you use Promo
        <br><br>
        <img src="http://images4.postdirect.com/master-images/404707/bo
        <br><img src="http://images4.postdirect.com/master-images/40470
        </font></td>
      <td width="50"><img src="http://images4.postdirect.com/master-ima
    </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-image
    <td width="474"><font face="arial, verdana" size="-2" color="#00000
    * This feature is available on the Palm&#153; IIIx, Palm&#153; IIIx
    ** Note: To use the MIK functionality, you need either a Palm OS&#1
    with  <nobr>built-in</nobr> modem or data capability that has eithe
    are using a phone, you must have data services from your mobile ser
    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 hre
    Or, unsubscribe by replying to this message, with "unsubscribe" as
    <br><br>
    ------------------<br>
    Copyright&#169; 2001 Palm, Inc. Palm OS, Palm Computing, HandFAX, H
    HotSync, iMessenger, MultiMail, Palm.Net, PalmConnect, PalmGlove, P
    and the Palm Platform Compatible Logo are registered trademarks of
    AnyDay, EventClub, HandMAIL, the HotSync Logo, PalmGear, PalmGlove,
    trade dress, PalmSource, Smartcode, and Simply Palm are trademarks
    product names may be trademarks or registered trademarks of their r
    <img src="http://images4.postdirect.com/master-images/404707/clear.
    <td width="54"><img src="http://images4.postdirect.com/master-image
  </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]:
```
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 head
    ## https://docs.python.org/3/library/email.examples.html

    # get email body content from parsed 'original msg'
    result['payload'] = msg.get_payload()

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

    try:
        for key, value in msg.items():
            result[key.replace('-', '_')] = value
    except Exception as e:
        print(f'Problem parsing email: email_path e')

    try:
        result['Date'] = parse(result['Date'], ignoretz = False).isofor
    except Exception as e:
        print(f"Problem converting date: {result.get('date')} {e}")

    tuple_result = tuple([str(result.get(column, None)) for column in c

    return ParsedEmail(*tuple_result)
```

In [10]:
```
parsed_msg = parse_email(plain_msg_example)
```

```
<ipython-input-9-9046de64f67e>:7: MarkupResemblesLocatorWarning: The i
nput looks more like a filename than markup. You may want to open this
file and pass the filehandle into Beautiful Soup.
  soup = BeautifulSoup(payload, 'html.parser')
```

In [11]:
```
print(parsed_msg.text)
```

```
Andy,

Thanks for giving me the opportunity to meet with you about the Analys
t/ Associate program.  I enjoyed talking to you, and look forward to c
ontributing to the success that the program has enjoyed.

Thanks and Best Regards,

Jeff Hammad
```

In [12]:
```
parsed_html_msg = parse_email(html_msg_example)
```

In [13]:
```
print(parsed_html_msg.text)
```

```



Paprika
```

In [ ]: