Assignment 4

January 18, 2021

[8]:
```
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
from pyspark.sql import Row
from collections import OrderedDict
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.

[10]:
```
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.

0.1 Assignment 4.1

[11]:
```
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

from email.message import EmailMessage
from pyspark.sql import Row
#import spark.implicits._


def convert_to_row(d: dict) -> Row:
    return Row(**OrderedDict(sorted(d.items())))

def make_spark_df(spark):
    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)
```

```
            with open(current_path,errors='ignore') as m:
                msg = EmailMessage()
                headers = Parser(policy=default).parsestr(m.read())
                msg['To'] = headers['to']
                msg['username'] = root
                msg['From'] = headers['From']
                msg['Subject'] = headers['Subject']
                msg['id'] = headers['Message-ID']
                msg['X-From'] = headers['X-From']
                records.append(msg)



    df = spark.createDataFrame(records).
↪toDF(msg['To'],msg['From'],msg['Subject'],msg['id'],msg['X-From'])

    print(records[0:10])
    #df = pd.DataFrame(records)

    ## TODO: Complete the code to code to create the Spark dataframe
    #return df
```

[105]:
```
def make_spark_df(spark):
    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)

            with open(current_path,errors='ignore') as m:
                #msg = EmailMessage()
                msg = []
                headers = Parser(policy=default).parsestr(m.read())

                #msg␣
↪=[headers['to'],headers['From'],headers['Subject'],headers['Message-ID'],headers['X-From']]
                msg␣
↪=[headers['Message-ID'],headers['Subject'],headers['From'],headers['From'],headers.
↪get_content()]
                records.append(msg)

        ## TODO: Complete the code to code to create the Spark dataframe
```

```
    df = pd.DataFrame(records)
    #df.columns = ['To','From','Subject','Message_id','X-from']
    df.columns = ['id','words','features','username','original_msg']

    return df
```

[106]:
```
df = make_spark_df(spark)
```

[107]:
```
df[['id','words','features','username','original_msg']] =␣
↪df[['id','words','features','username','original_msg']].astype('str')
```

[108]:
```
df = spark.createDataFrame(df)
```

[109]:
```
df.head()
```

[109]:
```
Row(id='<23608348.1075861846229.JavaMail.evans@thyme>', words='TRV Notification:
(NG - Price P/L - 11/21/2001)', features='joey.taylor@enron.com',
username='joey.taylor@enron.com', original_msg='The report named: NG - Price P/L
<http://trv.corp.enron.com/linkFromExcel.asp?report_cd=10&report_name=NG+-+Price
+P/L&category_cd=5&category_name=FINANCIAL&toc_hide=1&sTV1=5&TV1Exp=Y&current_ef
ct_date=11/21/2001>, published as of 11/21/2001 is now available for viewing on
the website.')
```

0.2 Assignment 4.2

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

[23]:
```
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, and look forward to␣
 ↪contributing to the success that the program has enjoyed.

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.postdirect.com/master-images/
 ↪404707/upper_left.gif" alt="" width="103" height="110" hspace="0" vspace="0"␣
 ↪border="0"></a><a href="http://insync-online.p04.com/u.d?
 ↪AkReaQA5eczXE=11"><img src="http://images4.postdirect.com/master-images/
 ↪404707/upper_right.gif" alt="" width="479" height="110" hspace="0"␣
 ↪vspace="0" border="0"></a></nobr></td>
</tr>
<tr valign="top">
```

```
 <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/
↪master-images/404707/clear.gif" width="4" height="1" hspace="0" vspace="0"␣
↪border="0" alt=""></td>
 <td width="20"><img src="http://images4.postdirect.com/master-images/404707/
↪clear.gif" width="20" height="1" hspace="0" vspace="0" border="0" alt=""></
↪td>
 <td width="165"><br><a href="http://insync-online.p04.com/u.d?
↪LkReaQA5eczXL=21"><img src="http://images4.postdirect.com/master-images/
↪404707/screen1.gif" alt="" width="165" height="159" hspace="0" vspace="0"␣
↪border="0"></a><br><img src="http://images4.postdirect.com/master-images/
↪404707/screen1_text.gif" alt="" width="93" height="26" hspace="0" vspace="0"␣
↪border="0"></td>
 <td width="20"><img src="http://images4.postdirect.com/master-images/404707/
↪clear.gif" width="20" height="1" hspace="0" vspace="0" border="0" alt=""></
↪td>
 <td width="165"><br><a href="http://insync-online.p04.com/u.d?
↪BkReaQA5eczXO=31"><img src="http://images4.postdirect.com/master-images/
↪404707/screen2.gif" alt="" width="165" height="159" hspace="0" vspace="0"␣
↪border="0"></a><br><img src="http://images4.postdirect.com/master-images/
↪404707/screen2_text.gif" alt="" width="93" height="26" hspace="0" vspace="0"␣
↪border="0"></td>
 <td width="20"><img src="http://images4.postdirect.com/master-images/404707/
↪clear.gif" width="20" height="1" hspace="0" vspace="0" border="0" alt=""></
↪td>
 <td width="165"><br><a href="http://insync-online.p04.com/u.d?
↪JkReaQA5eczXRs=41"><img src="http://images4.postdirect.com/master-images/
↪404707/screen3.gif" alt="" width="165" height="159" hspace="0" vspace="0"␣
↪border="0"></a><br><img src="http://images4.postdirect.com/master-images/
↪404707/screen3_text.gif" alt="" width="93" height="26" hspace="0" vspace="0"␣
↪border="0"></td>
 <td width="19"><img src="http://images4.postdirect.com/master-images/404707/
↪clear.gif" width="19" height="1" hspace="0" vspace="0" border="0" alt=""></
↪td>
 <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/
↪master-images/404707/clear.gif" width="4" height="1" hspace="0" vspace="0"␣
↪border="0" alt=""></td>
</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" height="1" hspace="0" vspace="0"␣
↪border="0" alt=""></td>
 <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="0" vspace="0" border="0"␣
↪alt=""></td>
     <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-images/404707/title1.gif" alt=""␣
↪width="336" height="20" hspace="0" vspace="0" border="0"></a>
       <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-images/404707/title2.gif" alt=""␣
↪width="460" height="20" hspace="0" vspace="0" border="0"></a>
       <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-images/404707/title3.gif" alt=""␣
↪width="461" height="31" hspace="0" vspace="0" border="0"></a>
       <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/u.d?MkReaQA5eczXRV=91">Click here␣
↪to
       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" height="60" hspace="0"␣
↪vspace="0" border="0">
       <br><img src="http://images4.postdirect.com/master-images/404707/clear.
↪gif" width="474" height="1" hspace="0" vspace="0" border="0" alt="">
       </font></td>
     <td width="50"><img src="http://images4.postdirect.com/master-images/
↪404707/clear.gif" width="50" height="1" hspace="0" vspace="0" border="0"␣
↪alt=""></td>
   </tr>
   </table></td>
```

```
    <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect.com/
↪master-images/404707/clear.gif" width="4" height="1" hspace="0" vspace="0"␣
↪border="0" alt=""></td>
  </tr>
  <tr>
  <td colspan="3"><img src="http://images4.postdirect.com/master-images/404707/
↪bottom.gif" width="582" height="67" hspace="0" vspace="0" border="0"></td>
  </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="0" vspace="0" border="0" alt=""></
↪td>
    <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.com/u.d?
↪RkReaQA5eczXRK=101">Click here</a> for
    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=121">click here</a>.
    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" border="0" alt=""></td>
    <td width="54"><img src="http://images4.postdirect.com/master-images/404707/
↪clear.gif" width="54" height="1" hspace="0" vspace="0" border="0" alt=""></
↪td>
  </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()
```

[117]:
```
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):
    parsed_msg = parse_html_payload(original_msg)
    result = {}

    headers = Parser(policy=default).parsestr(parsed_msg)
    if headers.is_multipart():
        for payload in headers.get_payload():
            body = paylod.get_payload()
    else:
        body = headers.get_payload()

    ## TODO: Use Python's email library to read the payload and the headers
    ## https://docs.python.org/3/library/email.examples.html
    #msg = EmailMessage()
    #headers = Parser(policy=default).parsestr(m.read())
    result['text'] = headers.get_content()
    result['From'] = headers['From']
    result['Subject'] = headers['Subject']
    result['id'] = headers['Message-ID']
    result['X-From'] = headers['X-From']
    result['payload'] = headers.get_payload()
```

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

[118]:
```
parsed_msg = parse_email(plain_msg_example)
```

[119]:
```
#print(parsed_msg.text)
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, and look forward to contributing
to the success that the program has enjoyed.

Thanks and Best Regards,

Jeff Hammad
```

[120]:
```
parsed_html_msg = parse_email(html_msg_example)
```

[121]:
```
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 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.
```

0.3 Assignment 4.3

[122]:
```
## This creates a schema for the email data
email_struct = StructType()

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

[123]:
```
## 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)
email_transformer = ParseEmailsTransformer()
tokenizer = Tokenizer(inputCol='text',outputCol='words')
c_v = CountVectorizer(inputCol='words',outputCol='features')
                      #,vocabSize=3,minDF=2.0)
## Use the custom ParseEmailsTransformer, Tokenizer, and CountVectorizer
## to create a spark pipeline
email_pipeline = Pipeline(stages=[email_transformer,tokenizer,c_v])
    ## TODO: Complete code
model = email_pipeline.fit(df)
result = model.transform(df)
```

```
      ␣
↪---------------------------------------------------------------------------

       PythonException                           Traceback (most recent call␣
↪last)

       <ipython-input-123-859977ceec6c> in <module>
        29 email_pipeline = Pipeline(stages=[email_transformer,tokenizer,c_v])
        30     ## TODO: Complete code
   ---> 31 model = email_pipeline.fit(df)
        32 result = model.transform(df)


       /usr/local/spark/python/pyspark/ml/base.py in fit(self, dataset, params)
       127                 return self.copy(params)._fit(dataset)
       128             else:
   --> 129                 return self._fit(dataset)
       130         else:
       131             raise ValueError("Params must be either a param map or a␣
↪list/tuple of param maps, "


       /usr/local/spark/python/pyspark/ml/pipeline.py in _fit(self, dataset)
       107                     dataset = stage.transform(dataset)
```

```
       108                 else:  # must be an Estimator
   --> 109                     model = stage.fit(dataset)
       110                     transformers.append(model)
       111                     if i < indexOfLastEstimator:


       /usr/local/spark/python/pyspark/ml/base.py in fit(self, dataset, params)
       127                 return self.copy(params)._fit(dataset)
       128             else:
   --> 129                 return self._fit(dataset)
       130         else:
       131             raise ValueError("Params must be either a param map or a␣
↪list/tuple of param maps, "


       /usr/local/spark/python/pyspark/ml/wrapper.py in _fit(self, dataset)
       319
       320     def _fit(self, dataset):
   --> 321         java_model = self._fit_java(dataset)
       322         model = self._create_model(java_model)
       323         return self._copyValues(model)


       /usr/local/spark/python/pyspark/ml/wrapper.py in _fit_java(self, dataset)
       316         """
       317         self._transfer_params_to_java()
   --> 318         return self._java_obj.fit(dataset._jdf)
       319
       320     def _fit(self, dataset):


       /usr/local/spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py in␣
↪__call__(self, *args)
      1302
       1303         answer = self.gateway_client.send_command(command)
   -> 1304         return_value = get_return_value(
       1305             answer, self.gateway_client, self.target_id, self.name)
       1306


       /usr/local/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
       135                 # Hide where the exception came from that shows a␣
↪non-Pythonic
       136                 # JVM exception message.
   --> 137                 raise_from(converted)
       138             else:
       139                 raise
```

```
      /usr/local/spark/python/pyspark/sql/utils.py in raise_from(e)


      PythonException:
    An exception was thrown from Python worker in the executor. The below is␣
↪the Python worker stacktrace.
  Traceback (most recent call last):
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/worker.py", line␣
↪605, in main
      process()
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/worker.py", line␣
↪597, in process
      serializer.dump_stream(out_iter, outfile)
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/serializers.py",␣
↪line 223, in dump_stream
      self.serializer.dump_stream(self._batched(iterator), stream)
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/serializers.py",␣
↪line 141, in dump_stream
      for obj in iterator:
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/serializers.py",␣
↪line 212, in _batched
      for item in iterator:
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/worker.py", line␣
↪450, in mapper
      result = tuple(f(*[a[o] for o in arg_offsets]) for (arg_offsets, f) in␣
↪udfs)
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/worker.py", line␣
↪450, in <genexpr>
      result = tuple(f(*[a[o] for o in arg_offsets]) for (arg_offsets, f) in␣
↪udfs)
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/worker.py", line 88,␣
↪in <lambda>
      return lambda *a: toInternal(f(*a))
    File "/usr/local/spark/python/lib/pyspark.zip/pyspark/util.py", line 107,␣
↪in wrapper
      return f(*args, **kwargs)
    File "<ipython-input-123-859977ceec6c>", line 2, in <lambda>
    File "<ipython-input-117-525a84534bc8>", line 25, in parse_email
    File "/opt/conda/lib/python3.8/email/message.py", line 1096, in get_content
      return content_manager.get_content(self, *args, **kw)
    File "/opt/conda/lib/python3.8/email/contentmanager.py", line 25, in␣
↪get_content
      raise KeyError(content_type)
  KeyError: 'multipart/mixed'
```

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

[ ]:
```

```