Code Highlights

Parse all sub-folders and files from a given folder:

- Universal Detector: Detects the encoding type of a file and returns the file text properly decoded. This is so helpful!
- os.walk

  for root, dirs, files in os.walk(enron_data_dir):

  ```
      for file_path in files:
  ```

- Parse email using the python email package: https://docs.python.org/3/library/email.examples.html (https://docs.python.org/3/library/email.examples.html)
- Create PySpark Dataframe and Schema:
  - https://www.geeksforgeeks.org/how-to-create-pyspark-dataframe-with-schema/ (https://www.geeksforgeeks.org/how-to-create-pyspark-dataframe-with-schema/)
  - https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/ (https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/)
- Proper Installation Notes to get Pyspark running in Juypter.
  - https://changhsinlee.com/install-pyspark-windows-jupyter/ (https://changhsinlee.com/install-pyspark-windows-jupyter/)

In [31]:
```
#//*** Run this code may or may not ensure pyspark is happy
"""
import findspark
findspark.init()

import pyspark # only run after findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()

df = spark.sql('''select 'spark' as hello ''')
df.show()
"""
print()

```

In [32]:
```
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
import datetime
```

```
10
11 from collections import namedtuple
12
13 import pandas as pd
14 #import s3fs
15 from bs4 import BeautifulSoup
16 from dateutil.parser import parse
17 from chardet.universaldetector import UniversalDetector
18
19 #//*** Must Run before pyspark
20 import findspark
21 findspark.init()
22
23 from pyspark.ml import Pipeline
24 from pyspark.ml.feature import CountVectorizer
25 from pyspark.ml.feature import HashingTF, Tokenizer
26 from pyspark.sql import SparkSession
27 from pyspark.sql.functions import col
28 from pyspark.ml.pipeline import Transformer
29 from pyspark.sql.functions import udf
30 from pyspark.sql.types import *
31
32 import pandas as pd
33
34 current_dir = Path(os.getcwd()).absolute()
35 results_dir = current_dir.joinpath('results')
36 results_dir.mkdir(parents=True, exist_ok=True)
37 data_dir = current_dir.joinpath('data')
38 data_dir.mkdir(parents=True, exist_ok=True)
39 enron_data_dir = data_dir.joinpath('enron')
40
41 output_columns = [
42         'username',
43         'original_msg',
44         'payload',
45         'Message-ID',
46         'Date',
47         'From',
48         'To',
49         'Subject',
50         'Mime-Version',
51         'Content-Type',
52         'Content-Transfer-Encoding',
53         'X-From',
54         'X-To',
55         'X-cc',
56         'X-bcc',
57         'X-Folder',
58         'X-Origin',
59         'X-FileName',
60         'Cc',
61         'Bcc'
62 ]
63
64 columns = [column.replace('-', '_') for column in output_columns]
65
```

```
ParsedEmail = namedtuple('ParsedEmail', columns)

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

```

In [ ]:

In [ ]:

In [33]:
```
print("Current Dir: ", current_dir)
print("Results Dir: ", results_dir)
print("Data Dir: ", data_dir)
print("Enron Data Dir: ", enron_data_dir)
print(ParsedEmail.username)

```

```
Current Dir:  C:\Users\family\DSCProjects\DSC\DSC650\assignment04
Results Dir:  C:\Users\family\DSCProjects\DSC\DSC650\assignment04\resu
lts
Data Dir:  C:\Users\family\DSCProjects\DSC\DSC650\assignment04\data
Enron Data Dir:  C:\Users\family\DSCProjects\DSC\DSC650\assignment04\d
ata\enron
<property object at 0x00000168A2252D18>
```

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

In [34]:
```
#//*** Copied Files manually, avoided Amazon S3 due to ongoing S3 iss
"""
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()
"""
print()
```

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

Assignment 4.1

In [35]:

```
#//*** Use Universal Detector to ascertain the message encoding type.
#//*** Returns a text based on the detected encoding type
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():

    #//*** All Fields are Stringtype except Date which is Timestamp t
    #//*** PySpark will accept a Datetime Object for timestamp type
    schema = StructType([
        StructField("id",StringType(),True),
        StructField("username",StringType(),True),
        StructField("original_msg",StringType(),True)
  ])
    records = []
    sc = spark.sparkContext

    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 emai
            ## 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)

            #//*** Get raw Email Text Message from File
            raw_email = read_raw_email(current_path)

            row = []


```

```
            #//*** Append ID
            id_path = str(current_path).replace(os.getcwd(),"").repla
            row.append(id_path)

            #//*****************
            #//*** Find username
            #//*****************


            limit = 10
            tgt = 'enron'
            val = ""
            i = -1
            n = 0
            #//*** Username will be the folder name after enron
            #//*** Search through the Path Parent names to find the i
            #//*** Username will be parent[i-1].name
            while val != tgt:
                i += 1
                val = str(Path(root).parents[i].name)

                #//*** Prevent Infinite Loops with a maximum Loop Lim
                n += 1
                if n > limit:
                    #//*** Limit Reached Set i to 2
                    i=1
                    break
            #//*** i can't be negative, reset to 0
            if i == 0:
                i = 1

            try:
                username = str(Path(root).parents[i-1].name)
            except:
                print(i)
            #//*** Append Username
            row.append(username)



            #//*** Add Original Message
            row.append(raw_email)

            #//*** Add Row as Record
            records.append(row)

            #//*** Print a Sample, every 200 records
            if len(records) % 200 == 0:
                print(f"username: {username} Path: {id_path} Msg Len:

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

df = make_spark_df()

```

```
username: davis-d Path: \davis-d\all_documents\244_ Msg Len: 716
```

```
username: davis-d Path: \davis-d\all_documents\425_ Msg Len: 938
username: davis-d Path: \davis-d\deleted_items\101_ Msg Len: 1436
username: davis-d Path: \davis-d\deleted_items\296_ Msg Len: 2400
username: davis-d Path: \davis-d\discussion_threads\129_ Msg Len: 1000
username: davis-d Path: \davis-d\discussion_threads\30_ Msg Len: 698
username: davis-d Path: \davis-d\finanial_operations\14_ Msg Len: 1816
username: davis-d Path: \davis-d\inbox\family\13_ Msg Len: 1187
username: davis-d Path: \davis-d\sap\18_ Msg Len: 2219
username: davis-d Path: \davis-d\sent\64_ Msg Len: 1122
username: davis-d Path: \davis-d\_sent_mail\55_ Msg Len: 2223
username: gay-r Path: \gay-r\all_documents\238_ Msg Len: 3570
username: gay-r Path: \gay-r\all_documents\41_ Msg Len: 678
username: gay-r Path: \gay-r\discussion_threads\138_ Msg Len: 3172
username: gay-r Path: \gay-r\inbox\18_ Msg Len: 1792
username: gay-r Path: \gay-r\sent\210_ Msg Len: 2076
username: gay-r Path: \gay-r\sent\393_ Msg Len: 3247
username: gay-r Path: \gay-r\_sent_mail\40_ Msg Len: 3233
username: may-l Path: \may-l\all_documents\75_ Msg Len: 1647
username: may-l Path: \may-l\inbox\1009_ Msg Len: 1354
username: may-l Path: \may-l\inbox\196_ Msg Len: 2635
username: may-l Path: \may-l\inbox\380_ Msg Len: 36915
username: may-l Path: \may-l\inbox\565_ Msg Len: 5189
username: may-l Path: \may-l\inbox\746_ Msg Len: 6925
username: may-l Path: \may-l\inbox\929_ Msg Len: 1937
username: may-l Path: \may-l\sent_items\1_ Msg Len: 713
username: meyers-a Path: \meyers-a\deleted_items\1123_ Msg Len: 2359
username: meyers-a Path: \meyers-a\deleted_items\287_ Msg Len: 1716
username: meyers-a Path: \meyers-a\deleted_items\467_ Msg Len: 2675
username: meyers-a Path: \meyers-a\deleted_items\695_ Msg Len: 1605
username: meyers-a Path: \meyers-a\deleted_items\883_ Msg Len: 1955
username: mims-thurston-p Path: \mims-thurston-p\all_documents\133_ Ms
g Len: 717
username: mims-thurston-p Path: \mims-thurston-p\deleted_items\111_ Ms
g Len: 4454
username: mims-thurston-p Path: \mims-thurston-p\deleted_items\297_ Ms
g Len: 3746
username: mims-thurston-p Path: \mims-thurston-p\deleted_items\490_ Ms
g Len: 1207
username: mims-thurston-p Path: \mims-thurston-p\deleted_items\673_ Ms
g Len: 1436
username: mims-thurston-p Path: \mims-thurston-p\inbox\11_ Msg Len: 54
65
username: mims-thurston-p Path: \mims-thurston-p\inbox\320_ Msg Len: 2
230
username: mims-thurston-p Path: \mims-thurston-p\sent\167_ Msg Len: 50
4
username: mims-thurston-p Path: \mims-thurston-p\sent_items\159_ Msg L
en: 1793
username: mims-thurston-p Path: \mims-thurston-p\_sent_mail\138_ Msg L
en: 1283
username: mims-thurston-p Path: \mims-thurston-p\_sent_mail\99_ Msg Le
n: 677
username: quenet-j Path: \quenet-j\deleted_items\8_ Msg Len: 2588
username: reitmeyer-j Path: \reitmeyer-j\deleted_items\103_ Msg Len: 2
343
username: reitmeyer-j Path: \reitmeyer-j\inbox\207_ Msg Len: 889
```

```
username: reitmeyer-j Path: \reitmeyer-j\inbox\67_ Msg Len: 1577
username: sanchez-m Path: \sanchez-m\inbox\41_ Msg Len: 2489
username: shively-h Path: \shively-h\all_documents\144_ Msg Len: 458
username: shively-h Path: \shively-h\all_documents\55_ Msg Len: 542
username: shively-h Path: \shively-h\deleted_items\283_ Msg Len: 533
username: shively-h Path: \shively-h\deleted_items\507_ Msg Len: 991
username: shively-h Path: \shively-h\deleted_items\700_ Msg Len: 11895
username: shively-h Path: \shively-h\discussion_threads\2_ Msg Len: 10
23
username: shively-h Path: \shively-h\inbox\81_ Msg Len: 4439
username: shively-h Path: \shively-h\sent\238_ Msg Len: 934
username: shively-h Path: \shively-h\sent_items\53_ Msg Len: 534
username: shively-h Path: \shively-h\_sent_mail\231_ Msg Len: 794
username: staab-t Path: \staab-t\crestone\29_ Msg Len: 657
username: staab-t Path: \staab-t\deleted_items\275_ Msg Len: 1527
username: staab-t Path: \staab-t\mark_whitt\6_ Msg Len: 1194
username: zipper-a Path: \zipper-a\all_documents\45_ Msg Len: 694
username: zipper-a Path: \zipper-a\deleted_items\161_ Msg Len: 924
username: zipper-a Path: \zipper-a\deleted_items\352_ Msg Len: 1603
username: zipper-a Path: \zipper-a\deleted_items\540_ Msg Len: 1295
username: zipper-a Path: \zipper-a\inbox\114_ Msg Len: 803
username: zipper-a Path: \zipper-a\inbox\92_ Msg Len: 575
username: zipper-a Path: \zipper-a\sent_items\222_ Msg Len: 609
username: zipper-a Path: \zipper-a\sent_items\89  Msg Len: 998
```

In [36]:

```
print(df)
print(df.printSchema())
df.show(n=10,truncate = True)
ids = df.select("id").collect()
print(ids[0].id)
print(len(ids))

```

```
DataFrame[id: string, username: string, original_msg: string]
root
```

Assignment 4.2

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

In [37]:

```
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
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 Analy

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'
<table border="0" cellpadding="0" cellspacing="0" width="582">
<tr valign="top">
  <td width="582" colspan="9"><nobr><a href="http://insync-online.p04
</tr>
<tr valign="top">
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect
  <td width="20"><img src="http://images4.postdirect.com/master-image
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?LkRea
  <td width="20"><img src="http://images4.postdirect.com/master-image
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?BkRea
  <td width="20"><img src="http://images4.postdirect.com/master-image
  <td width="165"><br><a href="http://insync-online.p04.com/u.d?JkRea
  <td width="19"><img src="http://images4.postdirect.com/master-image
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect
</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="582">
<tr valign="top">
  <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdirect
  <td width="574"><br>
    <table border="0" cellpadding="0" cellspacing="0" width="574" bgc
    <tr>
      <td width="50"><img src="http://images4.postdirect.com/master-i
      <td width="474"><font face="verdana, arial" size="-2"color="#00
        <br>
        Dear THERESA,
        <br><br>
        Due to overwhelming demand for the Palm OS&#174; v4.1 Upgrade
        extending the special offer of 25% off through November 30, 2
        increase the functionality of your Palm&#153; III, IIIx, IIIx
        new Palm OS v4.1 through this extended special offer. You'll
        <b>for just $29.95 when you use Promo Code <font color="#FF00
        <b>$10 savings</b> off the list price.
        <br><br>
        <a href="http://insync-online.p04.com/u.d?NkReaQA5eczXRh=51">
        <br><br>
        <a href="http://insync-online.p04.com/u.d?MkReaQA5eczXRm=61">
        <br><br>
        You can do a lot more with your Palm&#153; handheld when you
        favorite features just got even better and there are some ter
        <br><br>
        <LI> Handwrite notes and even draw pictures right on your Pal
        <LI> Tap letters with your stylus and use Graffiti&#174; at t
        <LI> Improved Date Book functionality lets you view, snooze c
        <LI> You can easily change time-zone settings</LI>

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

```
104
105         <br><br>
106         <a href="http://insync-online.p04.com/u.d?VEReaQA5eczXRQ=81">
107         <br><br>
108         <LI> Use your GSM compatible mobile phone or modem to get onl
109         <LI> Stay connected with email, instant messaging and text me
110         <LI> Send applications or records through your cell phone to
111              important information to others</LI>
112
113         <br><br>
114         All this comes in a new operating system that can be yours fo
115         upgrade to the new Palm&#153; OS v4.1</a> and you'll also get
116         <nobr>1-800-881-7256</nobr> to order via phone.
117         <br><br>
118         Sincerely,<br>
119         The Palm Team
120         <br><br>
121         P.S. Remember, this extended offer opportunity of 25% savings
122         and is only available through the Palm Store when you use Pro
123         <br><br>
124         <img src="http://images4.postdirect.com/master-images/404707/
125         <br><img src="http://images4.postdirect.com/master-images/404
126         </font></td>
127       <td width="50"><img src="http://images4.postdirect.com/master-i
128     </tr>
129     </table></td>
130     <td width="4" bgcolor="#CCCCCC"><img src="http://images4.postdire
131   </tr>
132   <tr>
133   <td colspan="3"><img src="http://images4.postdirect.com/master-imag
134   </tr>
135 </table>
136 <table border="0" cellpadding="0" cellspacing="0" width="582">
137   <tr>
138     <td width="54"><img src="http://images4.postdirect.com/master-ima
139     <td width="474"><font face="arial, verdana" size="-2" color="#000
140     * This feature is available on the Palm&#153; IIIx, Palm&#153; II
141     ** Note: To use the MIK functionality, you need either a Palm OS&
142     with  <nobr>built-in</nobr> modem or data capability that has eit
143     are using a phone, you must have data services from your mobile s
144     a list of tested and supported phones that you can use with the M
145     <br><br>
146     ------------------<br>
147     To modify your profile or unsubscribe from Palm newsletters, <a h
148     Or, unsubscribe by replying to this message, with "unsubscribe" a
149     <br><br>
150     ------------------<br>
151     Copyright&#169; 2001 Palm, Inc. Palm OS, Palm Computing, HandFAX,
152     HotSync, iMessenger, MultiMail, Palm.Net, PalmConnect, PalmGlove,
153     and the Palm Platform Compatible Logo are registered trademarks o
154     AnyDay, EventClub, HandMAIL, the HotSync Logo, PalmGear, PalmGlov
155     trade dress, PalmSource, Smartcode, and Simply Palm are trademark
156     product names may be trademarks or registered trademarks of their
157     <img src="http://images4.postdirect.com/master-images/404707/clea
158     <td width="54"><img src="http://images4.postdirect.com/master-ima
159   </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=
<img src="http://p04.com/1x1.dyn?0vEGou8Ig30ba2L2bLn" width=1 height=
</html>

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


```

In [38]:

```
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 h
    ## https://docs.python.org/3/library/email.examples.html

    for column in output_columns:

        #//*** Username is the root parent folder name
        if column == 'username':
            continue

        #//*** Append the whole unprocessed text file
        if column == 'original_msg':
            result[column] = original_msg
            continue

        #//*** Append the Payload, which is the message body
        #//*** Send raw Text to parse_html_payload. This will use bea
        #//*** html and return plain text
        if column == 'payload':
            #print('payload: ')
            result[column] = parse_html_payload(msg.get_content())
            continue

        #//*** Convert Date to Datetime
        #if column == 'Date':
            #//*** Convert Text to Datetime Object
        #    dt = datetime.datetime.strptime(msg[column],"%a, %d %b %

        #    result[column] = dt
        #    continue
        #//*** All other headers are pass-thru strings
        if column in msg.keys():
            #print(column, )
```

```
            result[column] = msg[column]
        else:

            #//*** Empty Field: return Zero Length String
            result[column] = ""


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

```

In [39]:
```
msg = Parser(policy=default).parsestr(plain_msg_example)
msg.keys()

```

Out[39]:
```
['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']
```

In [40]:
```
parsed_msg = parse_email(plain_msg_example)
print(parsed_msg.payload)

```

```
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 [41]:
```
parsed_html_msg = parse_email(html_msg_example)
```

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 multi
ple alarms all with a single tap
 You can easily change time-zone settings



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



 Use your GSM compatible mobile phone or modem to get online and acces
s the web
 Stay connected with email, instant messaging and text messaging to GS
M mobile phones
 Send applications or records through your cell phone to schedule meet
ings 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 lates
t 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 Prom
o Code OS41WAVE.









    * This feature is available on the Palm™ IIIx, Palm™ IIIxe, and Pa
lm™ 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 infrare
d port or cable exits.  If you
    are using a phone, you must have data services from your mobile se
rvice provider.  Click here for
    a list of tested and supported phones that you can use with the MI
K. 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, HandS
TAMP, 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, PalmGlov
e, 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

Notes:

https://www.analyticsvidhya.com/blog/2019/11/build-machine-learning-pipelines-pyspark/ (https://www.analyticsvidhya.com/blog/2019/11/build-machine-learning-pipelines-pyspark/)

https://towardsdatascience.com/data-transformation-in-pyspark-6a88a6193d92 (https://towardsdatascience.com/data-transformation-in-pyspark-6a88a6193d92)

Pyspark UDF - User Defined Functions:

https://sparkbyexamples.com/pyspark/pyspark-udf-user-defined-function/ (https://sparkbyexamples.com/pyspark/pyspark-udf-user-defined-function/)

In [42]:
```
## This creates a schema for the email data
email_struct = StructType()

#//*** All Columns are StringType except the Date Column which is Tim
for column in columns:
    #if column == "Date":
    #    email_struct.add(column, TimestampType(), True)
    #else:
```

In [43]:
```
## This creates a user-defined function which can be used in Spark
```

```
## It transforms the custom function parse_emails into a function use
parse_email_func = udf(lambda z: parse_email(z), email_struct)

#//*** Takes the existing df, passes each column into parsed_email()
#//*** This step parses the email header into separate columns
def parse_emails(input_df):
    #//*** Selects the first three columns of the input_df
    #//*** The fourth column applies the transformed parse_email_func
    #//*** and outputs the results in parsed_email column
    new_df = input_df.select(
        'username', 'id', 'original_msg', parse_email_func('original_
    )

    #//*** Extracts each (sub)column in parsed_email into a df column
    for column in columns:
        new_df = new_df.withColumn(column, new_df.parsed_email[column

    #//*** Removed the parsed_email column filled with sub columns
    new_df = new_df.drop('parsed_email')

    return new_df

#//*** Transformer function used to apply parse_emails, which in turn
class ParseEmailsTransformer(Transformer):
    def _transform(self, dataset):
        """
        Transforms the input dataset.

        :param dataset: input dataset, which is an instance of :py:cl
        :returns: transformed dataset
        """
        return dataset.transform(parse_emails)


## Use the custom ParseEmailsTransformer, Tokenizer, and CountVectori
## to create a spark pipeline
email_pipeline = Pipeline(
    ## TODO: Complete code
    stages=[
        ParseEmailsTransformer(),
        Tokenizer(inputCol='payload', outputCol="words"),
        CountVectorizer(inputCol='words', outputCol='features')
        ]

)
model = email_pipeline.fit(df)
result = model.transform(df)


```

In [44]:

```
result.select('id', 'words', 'features').show(n=20, truncate=True)

```

```
+--------------------+--------------------+--------------------+
|                  id|               words|            features|
+--------------------+--------------------+--------------------+
| \davis-d\2_trash\1_|[, >, , , , , >, ...|(99771,[0,1,2,3,4...|
| \davis-d\2_trash\2_|[fyi..., thanks.,...|(99771,[0,1,2,3,5...|
| \davis-d\2_trash\3_|[----------------...|(99771,[0,1,2,6,7...|
| \davis-d\2_trash\4_|[-----original, m...|(99771,[0,2,6,7,9...|
|\davis-d\2_trash\...|[hi, mommy!, , ye...|(99771,[0,1,2,6,7...|
|\davis-d\2_trash\...|[hey, sweetie,, ,...|(99771,[0,1,7,10,...|
|\davis-d\2_trash\...|[----------------...|(99771,[0,10,25,2...|
|\davis-d\2_trash\...|[----------------...|(99771,[0,1,2,3,5...|
|\davis-d\2_trash\...|[----------------...|(99771,[0,2,3,6,7...|
|\davis-d\2_trash\...|[----------------...|(99771,[0,1,2,3,7...|
|\davis-d\2_trash\...|[----------------...|(99771,[0,1,2,3,4...|
|\davis-d\2_trash\...|[----------------...|(99771,[0,10,25,2...|
|\davis-d\2_trash\...|[, , , , --------...|(99771,[0,7,10,12...|
|\davis-d\2_trash\...|[----------------...|(99771,[0,10,14,2...|
|\davis-d\2_trash\...|[are, you, on, th...|(99771,[0,1,7,10,...|
|\davis-d\2_trash\...|[listen, girly!, ...|(99771,[0,2,3,7,1...|
|\davis-d\2_trash\...|[candis, all, you...|(99771,[0,1,2,3,7...|
|\davis-d\2_trash\...|[what, is, your, ...|(99771,[0,2,7,11,...|
|\davis-d\2_trash\...|[candis, -, , why...|(99771,[0,7,12,14...|
|\davis-d\2_trash\   |[----------------   |(99771,[0,4,10,13   |
```