Email Tracking BOT Detection

March 15, 2020

1 Email Tracking BOT Detection - Data Exploration

1.1 Problem

More and more, email tracking data is originating from BOT activity as opposed to organically by the actual person the email is intended for. BOTs that look at emails are automated programs that open the email and/or click on links within the email. A BOT's intentions can be malicious, data gathering or beneficial. An example of a BOT could be a service that subscribes to various email lists and clicks on links to extract coupons.

Through manual investigation, most of the BOTs found in this dataset are not malicious but come from protective services like Anti-Phishing devices that protects the email contact from dangerous emails.

1.2 Goal

To find ways to identify which requests are from BOT activity and which are organic actively from real people. If we can reliably identify which requests are from BOT activity, we would then know IP Addresses or blocks of IPs that the BOTS are coming from. Knowing the IPs used by BOTs would allow us to take different actions between BOT activity and organic activity.

But since manually review shows most of the BOTs are protective, we don't want to change the response of the request. In other words, we want the click to take you to the same URL if it is a BOT or organic. But we can do things like adding the activity to reporting.

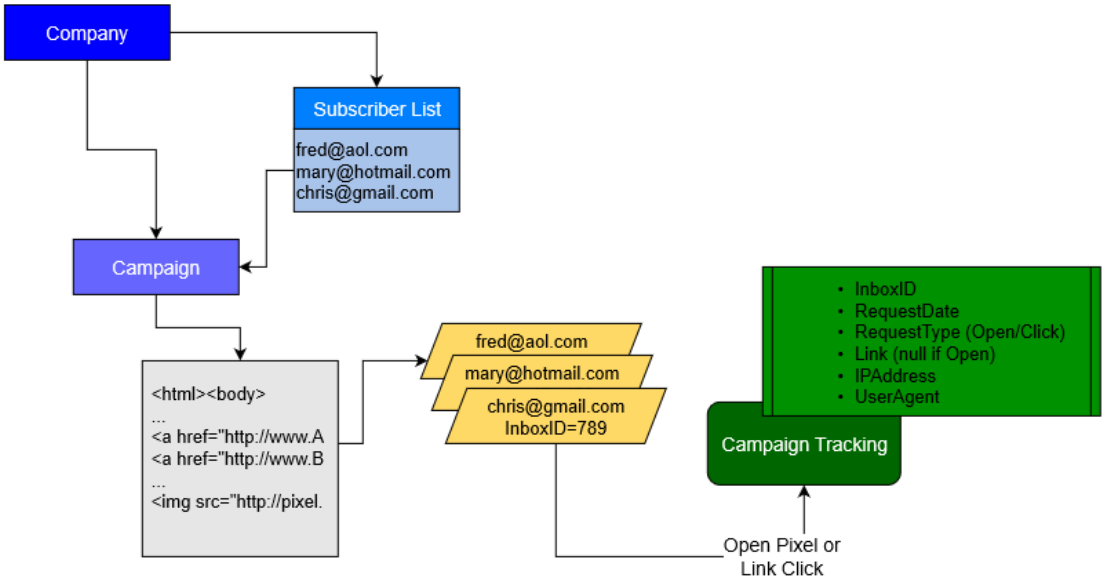
1.3 Why do we Care?

It is critical for customers who send email to be able to judge the effectiveness of the email campaign. Without accurate open and click tracking, customers can not make modifications to improve their campaigns and cannot do things like A/B testing.

1.4 What is Not looked at?

Though it would be a big benefit to know if the BOT is malicious, data mining or protective, this is not looked at. To determine this, a lot of manually research along with subjective conclusions are required and not in the scope of this project.

1.5 Email Tracking Data Overview:



1.6 Terminology

- **Subscriber List** - A list of email addresses that have opted into receiving emails from the company. A list have have many emails. A company can have many subscriber lists. An email can belong to many lists and many companies. But a subscriber list can only belong to one company.
- **Campaign** - An email message send out by a company. This email is sent to some or all the subscribers on the list. With the exception of some limited personalization, all subscribed receive the same content. A list can have many campaigns, but a campaign can only have one subscriber list.
- **Inbox** - What the contact sees as a unique message in their inbox (Outlook, GMail, etc.). If a company sends a single campaign to 1 million contacts, there will be 1 million InboxIDs for that campaign.
- **Request** - A tracking HTTP request from an action the contact takes on a message in their inbox. There are several types of requests, but they are primarily Opens which are pixel image reqeusts and Clicks.
- **Link** - A clickable and trackable URL embedded in the email message. The tracking system identifies the click, saved the tracking information and returns a redirect URL so the contact gets to their intended URL destination.
- **AS Number/Name** - Who owns the block of IP address. https://en.wikipedia.org/wiki/Autonomous_system_(Internet)
- **CIDR** - A notation of shows us what range of IP address are in a block. https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing
- **Network Address Trasnlation - NAT** - Most computers and phones do not have a public IP Address. NAT is a way for many different computer to reuse the same public IP Address. https://en.wikipedia.org/wiki/Network_address_translation

1.7 Columns

Raw columns recieved with HTTP Request:

- **InboxID** - Uniquely identifies the Campaign and the email it was sent to.
- **Company** - Who owns the campaign
- **RequestType** -
  - *Open* - The contact downloaded images for the message. A tracking pixel added to the HTML body of the message makes a request to the tracking system to identify who opened the message.
  - *Read* - The message was opened for a pre-defined period of time.
  - *Click* - The contact clicked on a URL in the message and the tracking system sees this request.
  - *Browser Link* - The contact requested seeing the message in a web browser. Mostly used if the message did not render well in the email client.
- **RequestDate** - The date of the request (i.e. Open or Click).
- **Link** - A hash of the URL requested.
- **IPAddress** - The IP Address the requested was generated from.
- **UserAgent** - A string submitted by email clients/browsers and Bots that attempts to identify the browser type. This can be made any string you want to it can be easily modified by Bots.

Columns that can be looked up from the InboxID:

- **SendDate** - The date and time (1 minute precision) of when the message was sent to that email. If a company sends a campaign to 5 million email, the send might take 10-20 mins. The SendDate is the date when that message was sent out and not the date the campaign started.
- **List** - A unique ID of the subscription list.

- **Campaign** - A unique ID of the campaign.
- **UserID** - A unique ID of the email address the message was sent to.
- **DomainID** - A unique ID of the domain of the email address the message was sent to.
- **EmailDomain** (redundant with DomainID)- A unique ID of the domain of the email address the message was sent to.
- **EmailRootLevelDomain** - The top level domain of the email domain. i.e. fred@gamil.com - ".com", mary@harvard.edu -

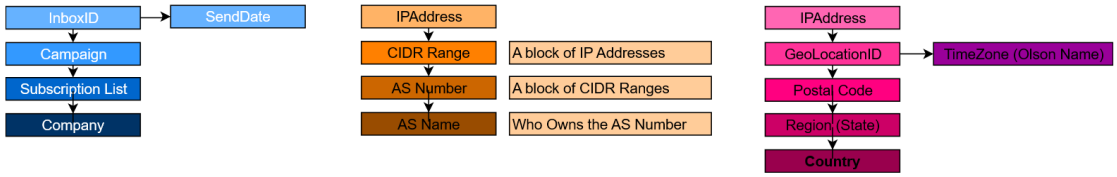
Columns that are derived from IP lookup data:

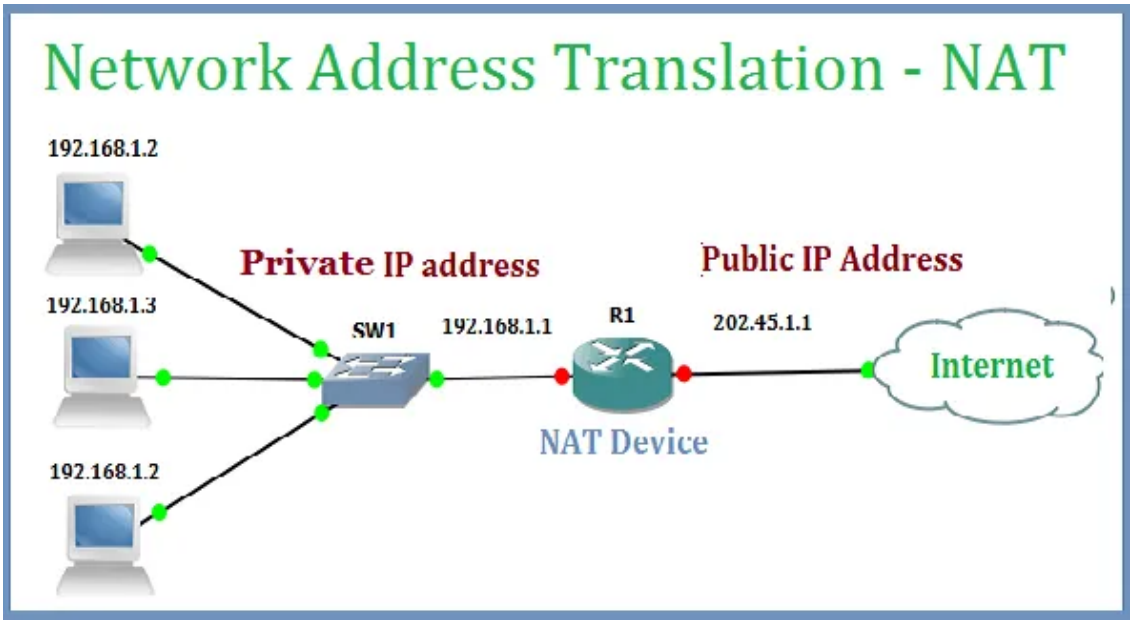
- **AS Name** - Owner of the IP Range (see above). Can be Null if owner not found in ASN database.
- **AS Number** - Owner of the IP Range (see above). Can be Null if owner not found in ASN database.
- **CIDR Range** - The block the IPAddress belongs to (see above). Can be Null if owner not found in ASN database.".edu"
- **Lat/Long** - Found from an IP Lookup service. Data available is not center of mass, but just the first Lat/Log in the polygon.
- **OlsonName** - Time Zone of the location the request came from. Also found via the IP Lookup.

Calculated columns after InboxID lookup is performed

- **SendRequestSeconds** - Number of seconds between the time the message was sent to the request date.

Hierarchies in Data





Most computers and phones do not use a stand-along IP Address. NAT is used to give a common public IP address for out web requests to the internet.

1.7.1 Because of this, an IP Address can be used my many people and many computers/phones.

So a single IP might be 100 People Common places where NAT is used: - **Work** - Most companies will use NAT with a small number of public IP Address. If the company has 100 employees, all 100 employees will reuse the same few public IP Address. - **Internet Service Provider** - The subscribers to an ISP will share public IP Address. Bigger ISPs can have many subnets and therefore a lot of public IP Address. But there can still be hundreds of people reusing the same IP Address. - **Cell Phone Carriers** - NAT is used by you phone carriers. Carriers like AT&T have thousands of public IP Addresses, but millions of cell phones. - **Cloud Computer Platforms (AWS)** - Users of these platforms may or may not use NAT. So, if the IP Address is traced back to Amazon Web Services, this public IP may or may not belong to a single AWS account.

1.7.2 The result, IP Address does not map to a single person or computer

[1]:
```
import numpy as np
from collections import defaultdict
from collections import Counter
import pandas as pd
from matplotlib import pyplot as plt
from datetime import datetime
from datetime import timedelta
from IPython.display import display
import pytz
from dateutil.tz import tzutc
import seaborn as sns
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
import os.path
from os import path
import pickle
```

1.7.3 BotDataSet Class

We will use a class to make sure the **loading**, **cleanup** and **sessionization** is done consistantly for both the IP Reputation and the Campaing dataset. This class will also give us code reuse for the trainging and testing datasets.

[2]:
```
exec(open("BotDataSet.py").read())
```

[3]:
```
botDataSet = BotDataSet()
botDataSet.loadCSV('C:/data/C1D499D6-AA43-9ABA-4EFD-0D97D0A33BE8.csv')
df = botDataSet.df
```

[4]:
```
pd.options.display.max_columns = None
display(df.head())
print(df.shape)
print('RequestDate Range: {}:{}'.format(df['RequestDate'].min(),␣
 ↪df['RequestDate'].max()))
df.nunique()
```

```
                                  InboxID  \
0  {85A0B8A0-4712-1803-93FC-B8461995CFC9}
1  {85A0B8A0-4712-1803-93FC-B8461995CFC9}
2  {85A0B8A0-4712-1803-93FC-B8461995CFC9}
3  {85A0B8A0-4712-1803-93FC-B8461995CFC9}
4  {F4724933-4570-E4BE-9B17-3D0928E9EE76}

                                  Company  \
0  {6C357BE7-9D8A-9D1E-FD3C-9D4C45077541}
```

```
1  {6C357BE7-9D8A-9D1E-FD3C-9D4C45077541}
2  {6C357BE7-9D8A-9D1E-FD3C-9D4C45077541}
3  {6C357BE7-9D8A-9D1E-FD3C-9D4C45077541}
4  {6C357BE7-9D8A-9D1E-FD3C-9D4C45077541}

                                    List  \
0  {15D09884-3FDD-8390-8493-A428F11F308D}
1  {15D09884-3FDD-8390-8493-A428F11F308D}
2  {15D09884-3FDD-8390-8493-A428F11F308D}
3  {15D09884-3FDD-8390-8493-A428F11F308D}
4  {15D09884-3FDD-8390-8493-A428F11F308D}

                                 Campaign RequestType            SendDate  \
0  {C1D499D6-AA43-9ABA-4EFD-0D97D0A33BE8}       Click 2020-01-17 09:03:00
1  {C1D499D6-AA43-9ABA-4EFD-0D97D0A33BE8}       Click 2020-01-17 09:03:00
2  {C1D499D6-AA43-9ABA-4EFD-0D97D0A33BE8}       Click 2020-01-17 09:03:00
3  {C1D499D6-AA43-9ABA-4EFD-0D97D0A33BE8}       Click 2020-01-17 09:03:00
4  {C1D499D6-AA43-9ABA-4EFD-0D97D0A33BE8}       Click 2020-01-17 09:03:00

          RequestDate                                    Link  \
0 2020-01-17 09:03:00  {2091AC3A-12C7-8FBF-3AA8-8C8C5F874EE2}
1 2020-01-17 09:03:00  {C7757DCC-7F71-8FAB-6836-0B061F8668F0}
2 2020-01-17 09:03:00  {B68FB2B6-F5FF-B7DA-22DD-589DF7AE29F7}
3 2020-01-17 09:03:00  {81E9F200-AB2B-54DE-B91D-506F173DA998}
4 2020-01-17 09:03:00  {048D23DD-04F1-D783-DB55-4BA2891DAA60}

                                 IPAddress  \
0  {9AD7D4CE-E247-DA9B-6E99-F5255E45F42F}
1  {1508EC2E-DA30-4661-442A-74EAD9243FE2}
2  {08E594F2-2009-3FB3-17F6-A37ABD24F6C9}
3  {4A03B84C-1010-D5C7-3541-127FBB258929}
4  {49F7F9A3-19F9-6E8A-5896-BC10581E5868}

                                 UserAgent            AS Name AS Number  \
0  {613F6166-E32A-4AE1-9FC4-7927349BDFE7}            Unknown   Unknown
1  {613F6166-E32A-4AE1-9FC4-7927349BDFE7}            Unknown   Unknown
2  {613F6166-E32A-4AE1-9FC4-7927349BDFE7}            Unknown   Unknown
3  {613F6166-E32A-4AE1-9FC4-7927349BDFE7}            Unknown   Unknown
4  {30634283-D150-237A-87FD-DA097EBBE5B9}  Amazon.com, Inc.     16509

                                CIDR Range  \
0                                  Unknown
1                                  Unknown
2                                  Unknown
3                                  Unknown
4  {4ED21F0C-8573-052A-670B-E4D315133AF5}

                                   UserID  \
```

```
0  {689E664D-1110-0BB1-7C6F-0DB872A731FA}
1  {689E664D-1110-0BB1-7C6F-0DB872A731FA}
2  {689E664D-1110-0BB1-7C6F-0DB872A731FA}
3  {689E664D-1110-0BB1-7C6F-0DB872A731FA}
4  {F89AA673-49FE-433F-7447-040277C7D897}

                                 DomainID  \
0  {80050545-EC9D-D79E-A18E-1666F4C81AAB}
1  {80050545-EC9D-D79E-A18E-1666F4C81AAB}
2  {80050545-EC9D-D79E-A18E-1666F4C81AAB}
3  {80050545-EC9D-D79E-A18E-1666F4C81AAB}
4  {C595CAFD-E63F-58ED-E8C8-8DB70C548720}

                             EmailDomain EmailRootLevelDomain CountryCode  \
0  {685CD069-4D61-F6AD-A3BE-0D814C0BC40C}                .org          US
1  {685CD069-4D61-F6AD-A3BE-0D814C0BC40C}                .org          US
2  {685CD069-4D61-F6AD-A3BE-0D814C0BC40C}                .org          US
3  {685CD069-4D61-F6AD-A3BE-0D814C0BC40C}                .org          US
4  {C6A21CB1-169B-B0F6-A812-35899D02D459}                .org          US

  RegionCode PostalCode        Lat        Long            OlsonName  \
0         VA      22102   38.98191  -77.256903     America/New_York
1         VA      22102   38.98191  -77.256903     America/New_York
2         VA      22102   38.98191  -77.256903     America/New_York
3         VA      22102   38.98191  -77.256903     America/New_York
4         OR      97207  45.850021 -119.631154  America/Los_Angeles

   SendRequestSeconds  SendRequestSeconds_ln
0                   0                    1.0
1                   0                    1.0
2                   0                    1.0
3                   0                    1.0
4                   0                    1.0


(24058, 25)
RequestDate Range: 2020-01-17 09:03:00:2020-01-21 11:49:00
```

[4]:
```
InboxID                11566
Company                    1
List                       1
Campaign                   1
RequestType                1
SendDate                  13
RequestDate             2187
Link                      37
IPAddress              10339
UserAgent               1181
```

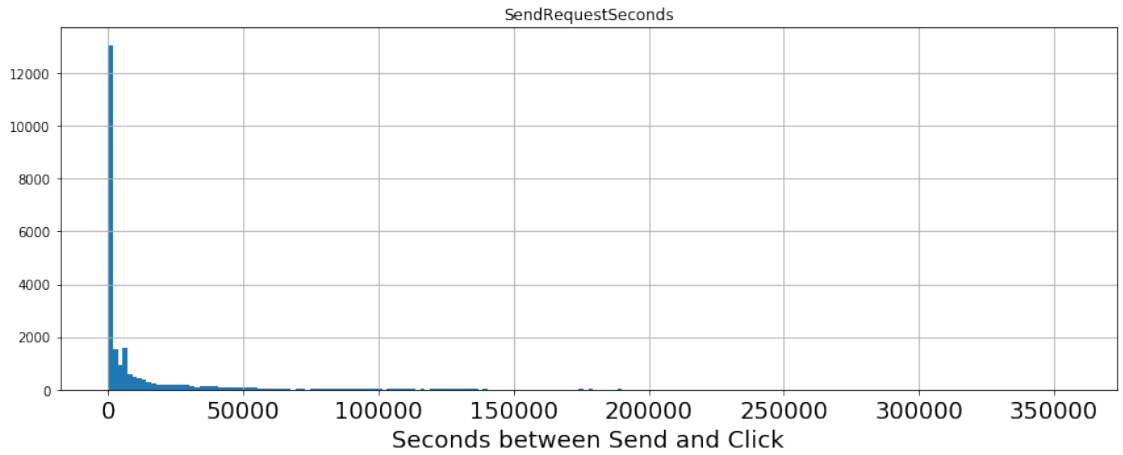
```
AS Name                    21
AS Number                  22
CIDR Range                354
UserID                  11566
DomainID                  952
EmailDomain               952
EmailRootLevelDomain       32
CountryCode                77
RegionCode                148
PostalCode               2572
Lat                      1612
Long                     1659
OlsonName                  79
SendRequestSeconds       2180
SendRequestSeconds_ln    2180
dtype: int64
```

2 Only One Metrizable Column in Raw Request Data

2.1 But this metric is one of the major factors that we anticipate will help us identify BOTs.

Mose of the click BOTs are safty devices that examin the risk of the email before it is released to the contact's inbox. So to prevent causing a long delay, the BOTs will want to process the links as soon as they are recieved. Through manual, most of the suspected BOTs click the message within 1 or 2 minutes after send went out.

[5]:
```
df.hist(column='SendRequestSeconds', bins=200, figsize=(12,5), xlabelsize=18)
plt.xlabel("Seconds between Send and Click", fontsize=18)
plt.tight_layout()
plt.show()
```

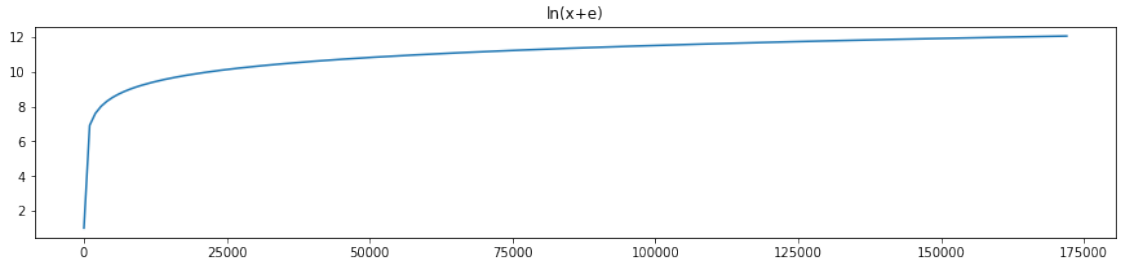


2.2 As you can see from the histogram, there is an overwelming number of clicks in the first few minusts after the send.

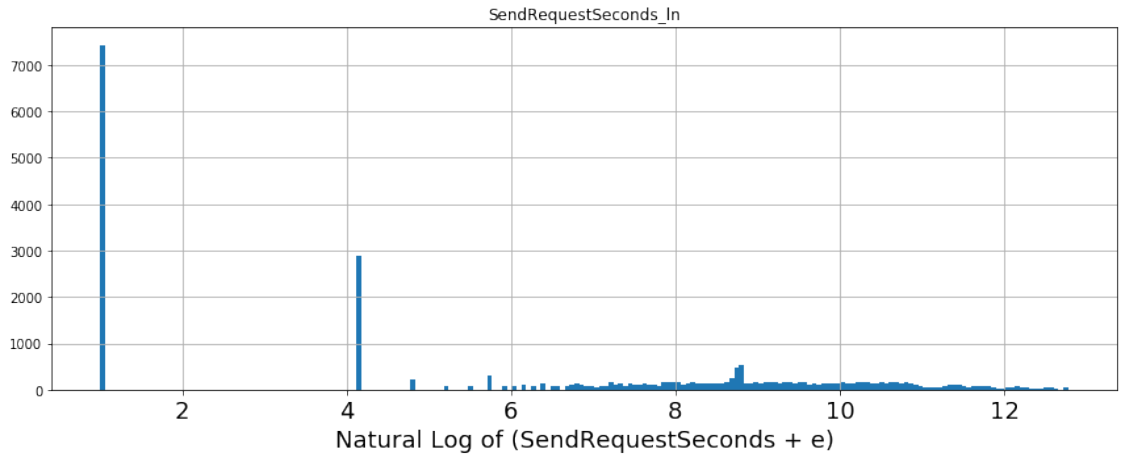
To even out this distribution, the BotDataSet class created a calculated column called SendRequestSeconds_ln This was done by taking the natural log of the SendRequestSeconds. To avoid zero and negative, the number e was added before the the log was taking. This make the bottom of the scale = 1. To give a feel of the upper limit of the this value, 2 days (172800 seconds) would give a value of 14.77.

2.3 $ SendRequestSeconds_ln = log_{e}(SendRequestSeconds + e) $

[6]:
```
def ln_seconds_plus_e(t):
    return np.log(t + np.exp(1))
seconds = range(0, 172800, 1000)
seconds_ln =  ln_seconds_plus_e(seconds)
f = plt.figure(figsize=(15, 3))
plt.plot(seconds, seconds_ln)
plt.title('ln(x+e)')
plt.show()
```



[7]:
```
df.hist(column='SendRequestSeconds_ln', bins=200, figsize=(12,5), xlabelsize=18)
plt.xlabel("Natural Log of (SendRequestSeconds + e)", fontsize=18)
plt.tight_layout()
plt.show()
```



3 Sessionization

Because of NAT, an IP Address might be reused by many different people and many different computers/phones. So we cannot say click activity on the same IP Address for a 1,000 different emails must be a BOT. It is just as likely that the IP Address is used by a cell phone carrier.

Sessionization help to mitigate this problem. Requests coming from the same IP Address in a short sequence of time has a much greater chance of being from the same person/computer. Combining IP and User Agent would tend to strengthen the correlation to a single person.

If we sessionize based on InboxID, we have a very good chance of getting requests from a single person. But if the requests are from a BOT, they could be processing many different InboxIDs on several IP/UserAgents at the same time.

InboxID	RequestDate	IPAddress	InboxSession	IPSession
123	10:03:01	123.456.1.1	**AAA**	**ZZZ**
123	10:03:01	123.456.1.1	**AAA**	**ZZZ**
456	10:03:02	12.45.8.7	**BBB**	**YYY**
456	10:03:03	45.65.1.4	**BBB**	**XXX**
123	10:03:58	45.65.1.4	**AAA**	**XXX**
789	10:22:22	74.124.5.6	**CCC**	**WWW**
456	10:22:23	74.124.5.6	**DDD**	**WWW**
123	10:22:24	123.456.1.1	**EEE**	**VVV**

InboxSession - Same InboxID without any gaps between requests over 2 mins (IP Address does not matter)

- AAA - Inbox **123** starting at 10:03:01 (3 requests)
- BBB - Inbox **456** starting at 10:03:02 (2 requests)
- CCC - Inbox **789** starting at 10:22:22 (1 request)
- DDD - Inbox **456** starting at 10:22:23 (1 request)
- EEE - Inbox **123** stariting at 10:22:24 (1 request)

IPSession - Same IP Address without any gaps between requests over 2 mins (InboxID does not matter)

- ZZZ - IP **123.456.1.1** starting at 10:03:01 (2 requests)
- YYY - IP **12.45.8.7** starting at 10:03:02 (1 request)
- XXX - IP **45.64.1.4** starting at 10:03:03 (2 request)
- WWW - IP **74.124.5.6** starting at 10:22:23 (2 requests)
- VVV - IP **123.456.1.1** starting at 10:22:24 (1 request)

3.1 Extend Dataframe with Session Columns

We want to sessionize in 3 ways, the InboxID, P Address and the combination of IP Address and User Agent

3.1.1 Inbox Based

- **InboxID** - This is straight forward. Any requests from the same inbox message will have the same session if there are not any gaps over 2 mins. This type of session will be used to judge behavior of the "Contact".

3.1.2 IP Based

- **IP Address** - This type of sessionization will be used to judge the behavior of IP Address from where the requests come from. If I click links from 2 different messages, the 2 InboxIDs will come from the same IP Address. But it might be unusual to have requests from many different InboxIDs from the same IP. Especially if the inboxes are to different emails and different email domains.
- **IP & UserAgent** - Historically, BOT detection has been done with a status list of IP/UserAgent combinations. Unsophisticated BOTs can use only 1 or 2 IP Address and always use the same User Agent string. These were easy to detect. Even though this approach is not adequate with the current times, the combination of IP/UserAgent is still very much a valid ay to sessionize requests. This holds up with manually scanning of data where we see the same UserAgent string used over and over again. Even if the UA they use is a common one, i.e. iPhone 10, the fact that the same IP Address uses the exact same User Agent over and over again for many different emails could be highly suspicious.

Below, we use the BotDataSet class to append session columns to our dataframe.

[8]:
```
max_session_time_gap = timedelta(seconds=120)

session_columns = {'InboxSessionID':('InboxID', None),
                   'IPSessionID':('IPAddress', None),
                   'IPUASessionID':('IPAddress', 'UserAgent')}

for t in session_columns.items():
    new_column_name, group_tuple = t
    group_column_1, group_column_2 = group_tuple
    botDataSet.loadSessionColumn(new_column_name, max_session_time_gap,␣
 ↪group_column_1, group_column_2)

df = botDataSet.df
```

[9]:
```
display_columns = ['RequestDate', 'InboxID', 'IPAddress', 'CIDR Range', 'AS␣
 ↪Number']
display_columns.extend(list(session_columns.keys()))

sdf = df[display_columns]
display(sdf.head())
sdf.nunique()
```

```
           RequestDate                                    InboxID  \
0  2020-01-17 09:03:00  {85A0B8A0-4712-1803-93FC-B8461995CFC9}
71 2020-01-17 09:03:00  {834A4BE1-D5F7-2143-E451-B7D2189175AE}
```

```
91 2020-01-17 09:03:00  {183046DF-1656-1D94-BBF1-63289C49F43C}
92 2020-01-17 09:03:00  {A36C9EE8-B8CC-AC57-DC47-CEA3D1F99726}
93 2020-01-17 09:03:00  {30E5C814-9797-EA1B-ED40-4165172C81F8}

                                  IPAddress  \
0   {9AD7D4CE-E247-DA9B-6E99-F5255E45F42F}
71  {6AC33884-ADE3-2F3E-6DD7-1731EC5D2945}
91  {0793D471-B890-C0C1-2C0B-DF4882B1D58C}
92  {0793D471-B890-C0C1-2C0B-DF4882B1D58C}
93  {0793D471-B890-C0C1-2C0B-DF4882B1D58C}

                                CIDR Range AS Number       InboxSessionID  \
0                                  Unknown   Unknown -5705089130875684729
71  {BF61FA52-8010-2731-2B4E-0C77A6C05E6A}     16509 -8410455302372937811
91  {7EF79BFD-BC3A-903D-C2BD-F5E295ACF4DE}     16509 -2155537913788110381
92  {7EF79BFD-BC3A-903D-C2BD-F5E295ACF4DE}     16509  4197718670201218189
93  {7EF79BFD-BC3A-903D-C2BD-F5E295ACF4DE}     16509 -3467654119446110596

            IPSessionID        IPUASessionID
0  -5152777450074098447 -4629743431653316648
71   263756345889371100  2402734463790523777
91   482115446712771047  6232456867269654117
92   482115446712771047  6232456867269654117
93   482115446712771047  6232456867269654117
```

[9]:
```
RequestDate        2187
InboxID           11566
IPAddress         10339
CIDR Range          354
AS Number            22
InboxSessionID    12398
IPSessionID       11532
IPUASessionID     11638
dtype: int64
```

4 Data exploration of the 3 Session Types

4.0.1 Approach (for each Session Type):

- **Build Aggregations** - i.e. UniqueInboxIDs, UniqueEmails, SessionDuration.
- **Reduce Aggregations** - Build another DF with just the aggregations we want to use in our unsupervised learning. We will use the other aggregations for visualization after we add labels.
- **Rescale Features** - Use MinMaxScaler.
- **Look at Covariance** - Ideally, we would like to see a lot of independence between the

features.

- **Explore Clustering** - Since the features are derived from aggregations, most of the data will be count data. Count data tends to follow an exponential decay PDF. Since our features are not Gaussian distributed, they will not be clustered around a mean and will be non-isotropic. A lot of the scatter plots will tend to show grouping around lines and curves. Because of this, DBSCAN looks to be a good fit for our initial clustering.

4.1 Build Aggregations

Most of the aggregations will be the same for all the session types. But there will be some differences. For example, the IP/UserAgent session does not make sense to use **UniqueUserAgents** since this type of session only has one User Agent string.

4.1.1 Build Column Lists

[10]:
```
# todo: Explicit lists vs something more generic?  Here explicit was chosen,␣
 ↪but might be able to find a cleaner way?

# Inbox Session
Inbox_agg = {  'InboxID':'min',
               'InboxSessionID':'size',
               'UserAgent':'nunique',
               'CIDR Range':'nunique',
               'AS Number':'nunique',
               'Link':'nunique',
               'SendRequestSeconds':'mean',
               'SendRequestSeconds_ln' : 'mean',
               'RequestDate':lambda x: (x.max() - x.min()).seconds}
Inbox_agg_renames = {'InboxSessionID': 'RequestCount',
                     'UserAgent': 'UniqueUserAgents',
                     'CIDR Range': 'UniqueCIDRs',
                     'AS Number': 'UniqueASNs',
                     'Link': 'UniqueLinks',
                     'SendRequestSeconds': 'MeanSendRequestSeconds',
                     'SendRequestSeconds_ln': 'MeanSendRequestSeconds_ln',
                     'RequestDate': 'SessionDuration'}

Inbox_unsup_columns = ['RequestCount',
                       'UniqueUserAgents',
                       'UniqueCIDRs',
                       'UniqueASNs',
                       'UniqueLinks',
                       'MeanSendRequestSeconds_ln',
                       'SessionDuration']

# IP Sessions
IP_agg = {     'IPAddress':'min',
```

```
                'UserAgent':'nunique',
                'IPSessionID':'size',
                'InboxID':'nunique',
                'UserID':'nunique',
                'EmailDomain':'nunique',
                'EmailRootLevelDomain':'nunique',
                'Link':'nunique',
                'SendRequestSeconds':'mean',
                'SendRequestSeconds_ln' : 'mean',
                'RequestDate':lambda x: (x.max() - x.min()).seconds}


IP_agg_renames = {   'IPSessionID': 'RequestCount',
                     'UserAgent': 'UniqueUserAgents',
                     'InboxID': 'UniqueInboxIDs',
                     'UserID': 'UniqueEmails',
                     'EmailDomain': 'UniqueEmailDomains',
                     'EmailRootLevelDomain': 'UniqueEmailRootLevelDomain',
                     'Link': 'UniqueLinks',
                     'SendRequestSeconds': 'MeanSendRequestSeconds',
                     'SendRequestSeconds_ln': 'MeanSendRequestSeconds_ln',
                     'RequestDate': 'SessionDuration'}

IP_unsup_columns = [   'RequestCount',
                       'UniqueInboxIDs',
                       'UniqueEmails',
                       'UniqueEmailDomains',
                       'UniqueEmailRootLevelDomain',
                       'UniqueLinks',
                       'MeanSendRequestSeconds_ln',
                       'SessionDuration',
                       'UniqueUserAgents']

# IPUA Sessions
IPUA_agg = {   'IPAddress':'min',
                'UserAgent':'min',
                'IPUASessionID':'size',
                'InboxID':'nunique',
                'UserID':'nunique',
                'EmailDomain':'nunique',
                'EmailRootLevelDomain':'nunique',
                'Link':'nunique',
                'SendRequestSeconds':'mean',
                'SendRequestSeconds_ln' : 'mean',
                'RequestDate':lambda x: (x.max() - x.min()).seconds}
```

```
IPUA_agg_renames = { 'IPUASessionID': 'RequestCount',
                     'InboxID': 'UniqueInboxIDs',
                     'UserID': 'UniqueEmails',
                     'EmailDomain': 'UniqueEmailDomains',
                     'EmailRootLevelDomain': 'UniqueEmailRootLevelDomain',
                     'Link': 'UniqueLinks',
                     'SendRequestSeconds': 'MeanSendRequestSeconds',
                     'SendRequestSeconds_ln': 'MeanSendRequestSeconds_ln',
                     'RequestDate': 'SessionDuration'}

IPUA_unsup_columns = [ 'RequestCount',
                       'UniqueInboxIDs',
                       'UniqueEmails',
                       'UniqueEmailDomains',
                       'UniqueEmailRootLevelDomain',
                       'UniqueLinks',
                       'MeanSendRequestSeconds_ln',
                       'SessionDuration']
```

4.2 Run groupby for each Session Type

4.2.1 Inbox

[11]:
```
# Inbox
df_Inbox_agg = df.groupby(['InboxID']).agg(Inbox_agg)
df_Inbox_agg.rename(columns=Inbox_agg_renames, inplace=True)

# IP Only
df_IP_agg = df.groupby(['IPSessionID']).agg(IP_agg)
df_IP_agg.rename(columns=IP_agg_renames, inplace=True)

#IP and User Agent
df_IPUA_agg =  df.groupby(['IPUASessionID']).agg(IPUA_agg)
df_IPUA_agg.rename(columns=IPUA_agg_renames, inplace=True)
```

4.3 Reduce Aggregations

use copy() to avoid views

[12]:
```
df_Inbox_agg_unsup = df_Inbox_agg[Inbox_unsup_columns].copy()
df_IP_agg_unsup = df_IP_agg[IP_unsup_columns].copy()
df_IPUA_agg_unsup = df_IPUA_agg[IPUA_unsup_columns].copy()
```

4.4 Rescale Features

We will use these new DataFrames for out unsupervised data exploration

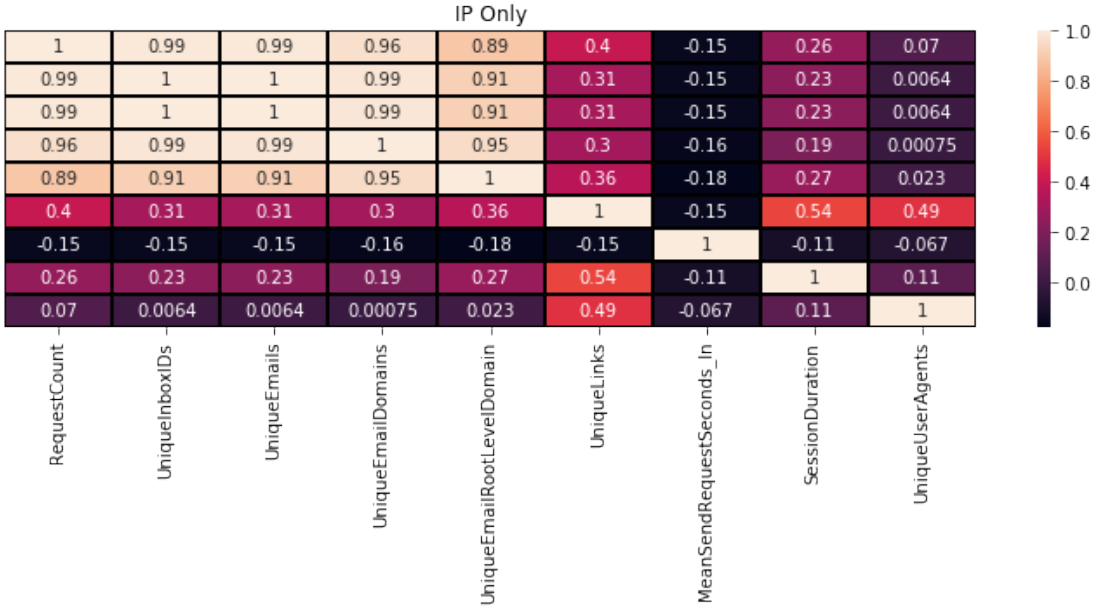
[13]:
```
def getScaledDataFrame(df):
    mms = MinMaxScaler()
    mms.fit(df)

    result = pd.DataFrame(mms.transform(df))
    result.columns = df.columns
    result.index = df.index
    return result

df_Inbox_agg_unsup_s = getScaledDataFrame(df_Inbox_agg_unsup)
df_IP_agg_unsup_s = getScaledDataFrame(df_IP_agg_unsup)
df_IPUA_agg_unsup_s = getScaledDataFrame(df_IPUA_agg_unsup)
unsup_data = {'Inbox':df_Inbox_agg_unsup_s, 'IP Only':df_IP_agg_unsup_s, 'IP +␣
 ↪UA':df_IPUA_agg_unsup_s}
```

4.5 Look at Covariance

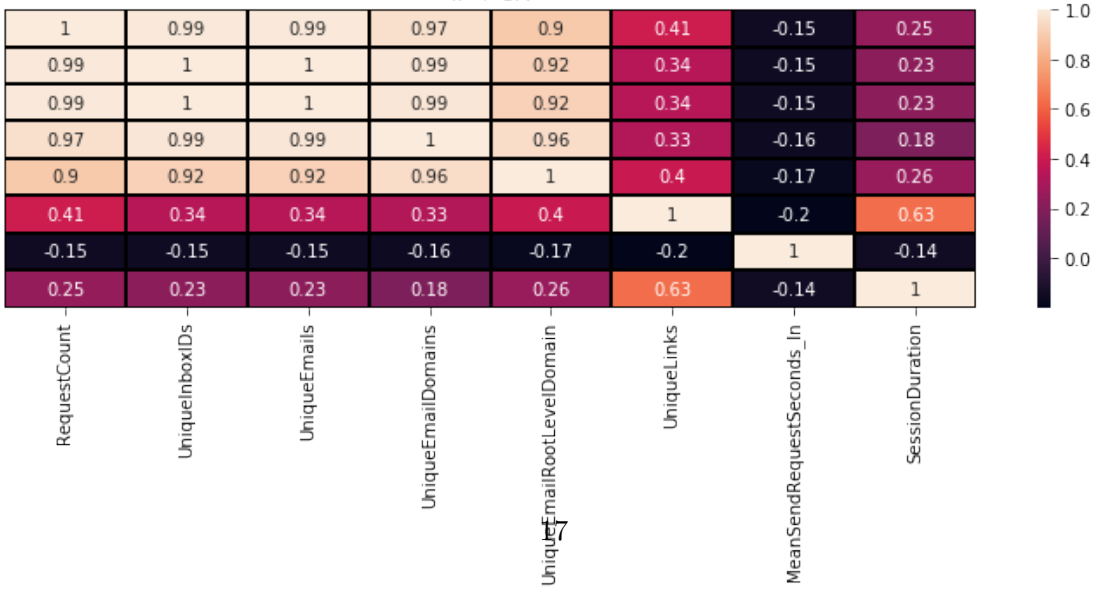
[14]:
```
f = plt.figure(figsize=(10, 15))
gs = f.add_gridspec(3,1)
i = 0

for df_name, df in unsup_data.items():
    ax = f.add_subplot(gs[i])
    corr = df.corr()
    sns.heatmap(corr,
        xticklabels=True,
        yticklabels=False,
        linewidths=1,
        linecolor='black',
        annot=True)
    plt.title(df_name)
    i += 1

plt.tight_layout()
plt.show()
```

Inbox

IP Only



IP + UA

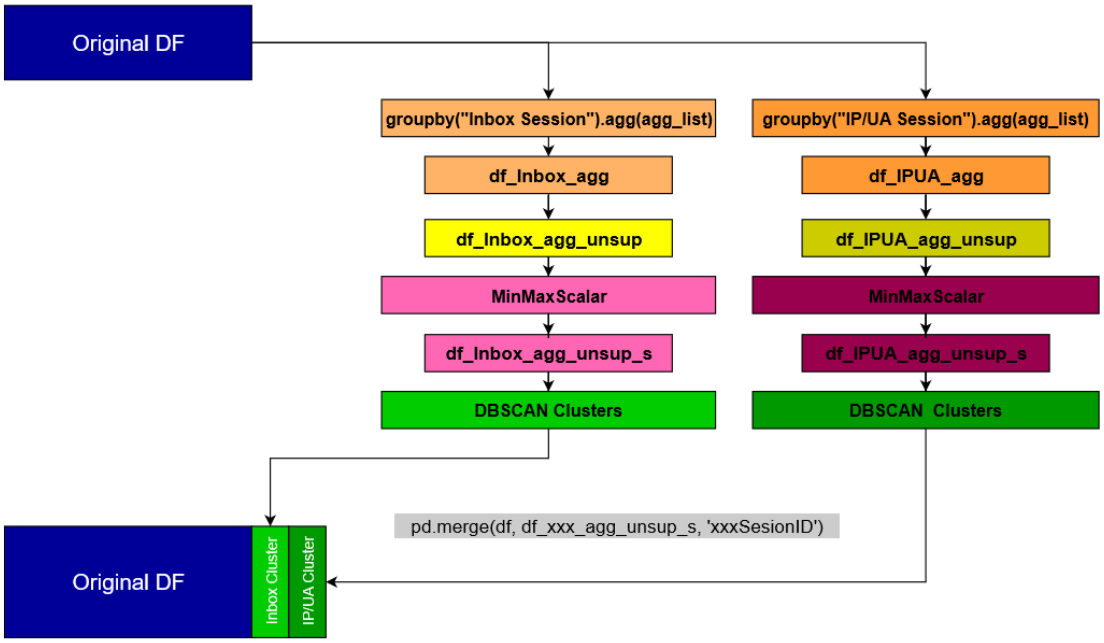


4.6 Conclusions:

The Inbox session data does look there is a fair amount of covariance, but without excessively dependent. PCA might make sense, but since we only have 7 features, we will skip this step.

Between the 2 IP bases sessions, the correlation matrix does not look very different. Since historically we know the combination of IP and User Agent have be successful in filtering out BOT activity, we will choose to use it for out IP session.

5 Explore Clustering



6 Clustering IP/UA Session Data

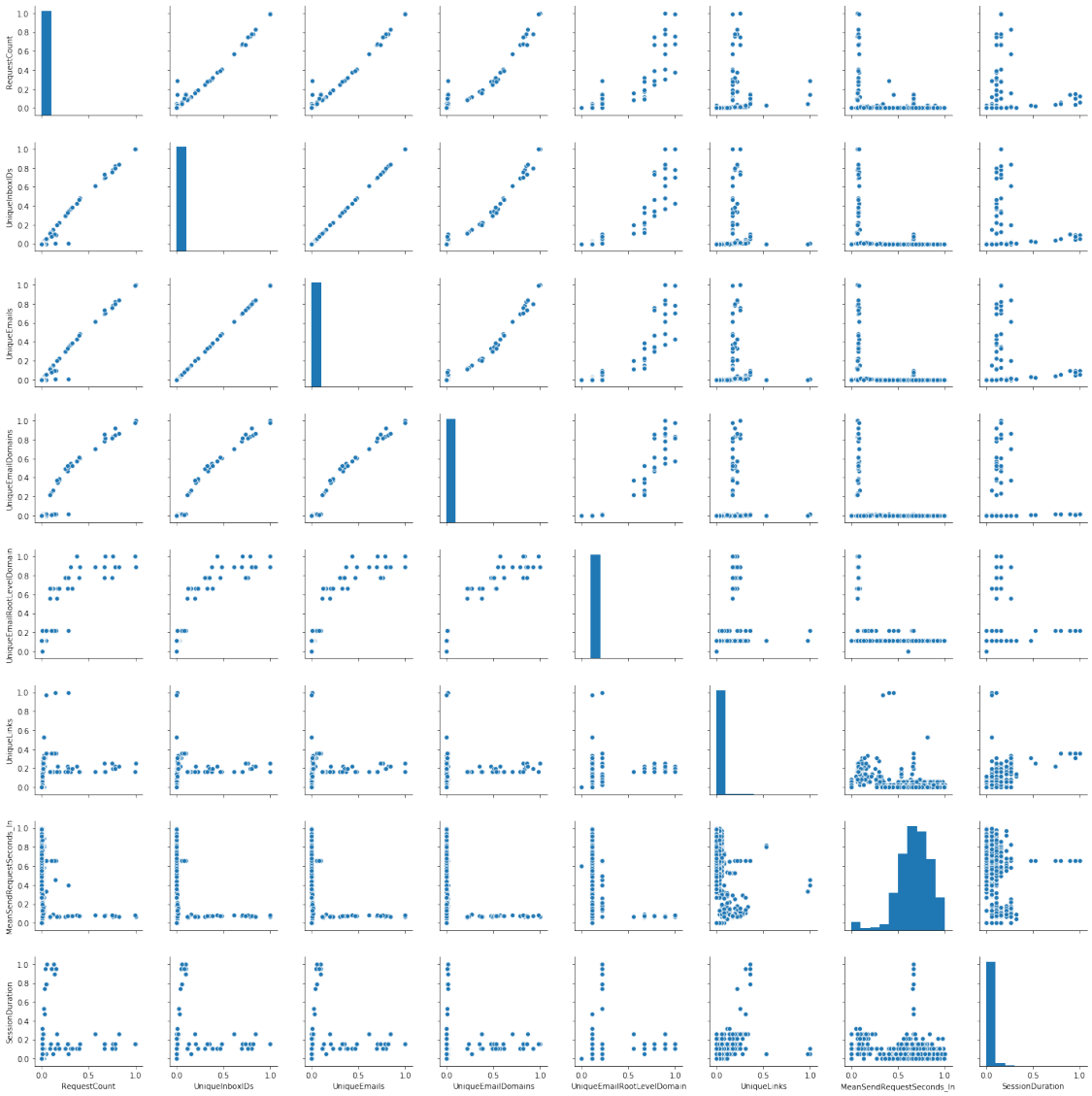
6.1 Let's look at all the scatter plots to help us choose the best unsupervised clustering model

As you can see, the data does not form any obvious globular clusters. Since this data is mostly aggregate counts, the distributions are more likely exponential decay then Gaussian.

Because of this, K-Mean would probably **not** be a good fit. An approach using nearest neighbors like **DBSCAN** should be a good probably be a good starting point.

[15]:
```
sns.pairplot(df_IPUA_agg_unsup_s)
```

[15]:
```
<seaborn.axisgrid.PairGrid at 0x18cdaf1f948>
```



7 DBSCAN IP/UA Session

[16]:
```
filename = 'C:/data/IPUA_DBSCAN.model'
build_new_model = True
if not path.exists(filename):
    build_new_model = True

if build_new_model:
    model = DBSCAN(eps=.4, min_samples=2, n_jobs=-1)
    model.fit(df_IPUA_agg_unsup_s)
    pickle.dump(model, open(filename, 'wb'))
else:
```

```
    model = pickle.load(open(filename, 'rb'))

labels = model.labels_
```

[17]:
```
print('Number of clusteres found = {}'.format(np.nanmax(labels)))
```

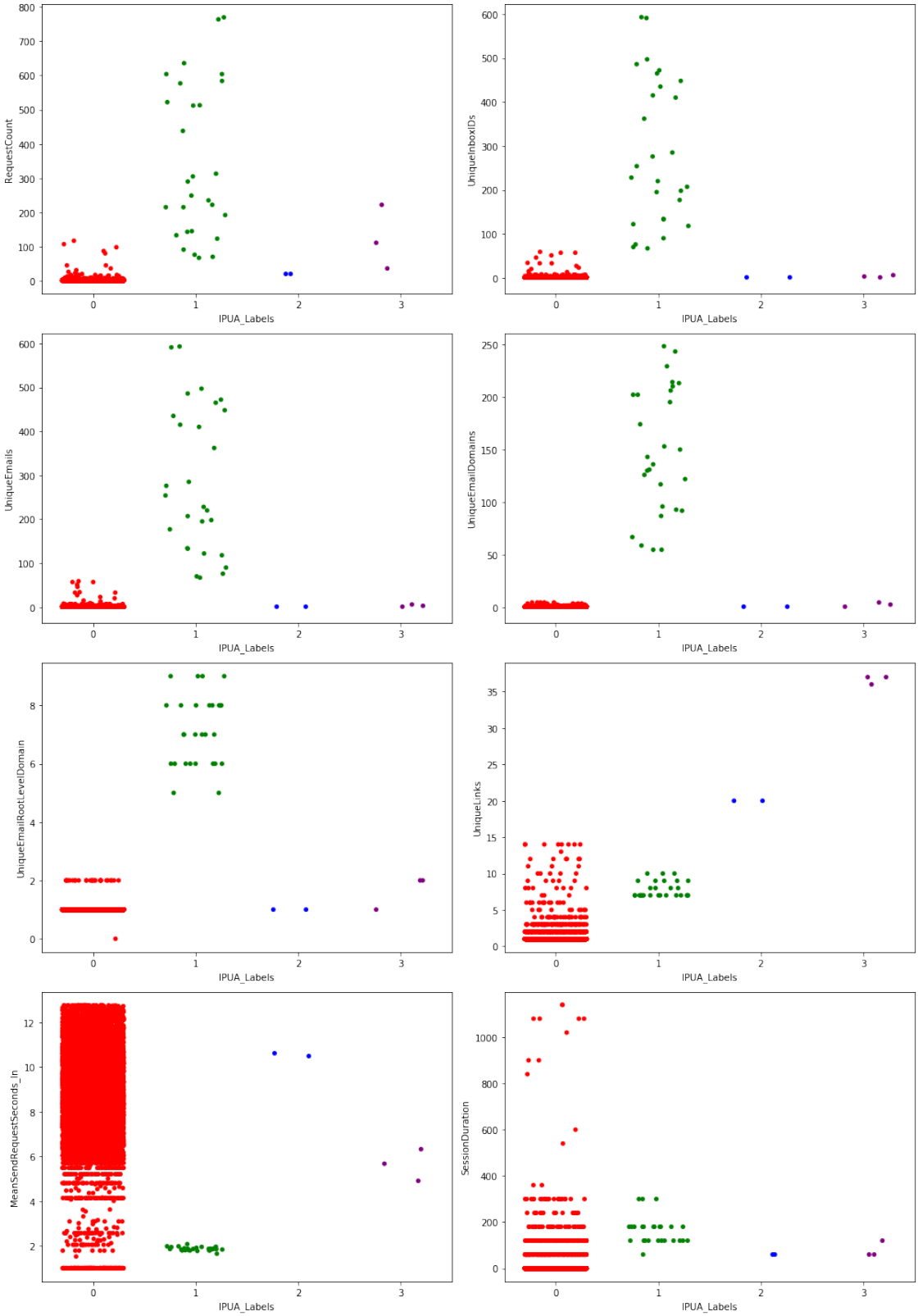
```
Number of clusteres found = 3
```

7.1 Append the DBSCAN Lables to the 2 IP group by DataFrames

[18]:
```
df_IPUA_agg['IPUA_Labels'] = labels #Origional Group By with additional agg␣
 ↪columns
df_IPUA_agg_unsup['IPUA_Labels'] = labels #Normilized Aggregations
df_IPUA_agg_unsup_s['IPUA_Labels'] = labels #Not=Normilized Aggregations
```

7.2 Build visualizations to manually inspect effectiveness of clusters

[19]:
```
palette = {0:'r', 1:'g',2:'b', 3:'purple', 4:'cyan', 5:'black'}
i = 0
j = 0
row_count = len(IPUA_unsup_columns) // 2
f = plt.figure(figsize=(14, 20))
gs = f.add_gridspec(row_count, 2)

for feature in IPUA_unsup_columns:
    ax = f.add_subplot(gs[i, j])
    sns.stripplot(x='IPUA_Labels', y=feature, palette=palette, jitter=.3,␣
 ↪data=df_IPUA_agg)
    if j == 1:
        j = 0
        i += 1
    else:
        j += 1

f.tight_layout()
plt.show()
```


RequestCount
UniqueInboxIDs
UniqueEmails
UniqueEmailDomains
UniqueEmailRootLevelDomain
UniqueLinks
MeanSendRequestSeconds_ln
SessionDuration
IPUA_Labels

[20]:
```
np.seterr(divide='ignore', invalid='ignore')
sns.pairplot(df_IPUA_agg_unsup_s, hue='IPUA_Labels', palette=palette)
```

[20]: <seaborn.axisgrid.PairGrid at 0x18c9bc3cbc8>

8 Merge the DBSCAN labels for IP/US GroupBy with original df

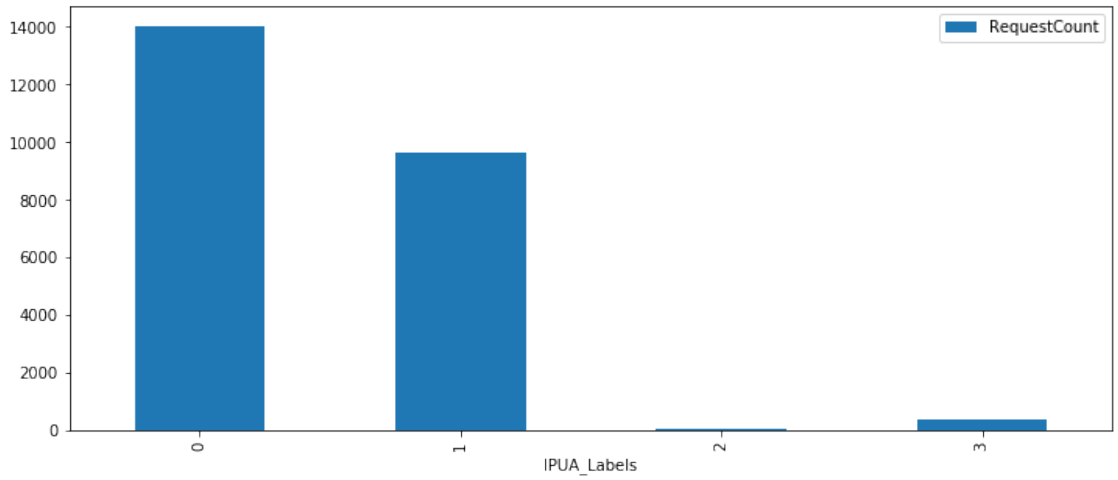
8.1 Remember, so far we have been looking at the group by DataFrame

Now that we merged the DBSCAN Lables into the raw date, we will see different cardinality then our groupby dataframes. This is because many rows from the origional DataFrame can have the same IP and User Agent

[21]:
```
df = botDataSet.df

#Note:  This will be an inner join
df_IPUA_clusters = pd.merge(df, df_IPUA_agg['IPUA_Labels'], on="IPUASessionID")
```

[22]:
```
p= pd.DataFrame(df_IPUA_clusters.groupby(['IPUA_Labels'])['InboxID'].count())
p = p.unstack(0, )
p = p.reset_index('IPUA_Labels')
p.columns = ['IPUA_Labels', 'RequestCount']
p.plot.bar('IPUA_Labels', 'RequestCount', figsize=(12,5))
```

[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x18c9f540a88>
```



[23]:
```
nunique_columns = [ 'InboxID',
                    'Company',
                    'List',
                    'Campaign',
                    'Link',
                    'IPAddress',
                    'UserAgent',
                    'AS Name',
                    'AS Number',
                    'CIDR Range',
                    'EmailDomain',
                    'EmailRootLevelDomain']
gb = df_IPUA_clusters.groupby(['IPUA_Labels'])[nunique_columns].nunique()
gb=gb.transpose()
gb
```

[23]:
IPUA_Labels	0	1	2	3
InboxID	10154	1409	2	10
Company	1	1	1	1
List	1	1	1	1
Campaign	1	1	1	1
Link	37	12	27	37
IPAddress	10327	28	2	2
UserAgent	1177	1	2	2
AS Name	21	1	1	2
AS Number	22	1	1	2
CIDR Range	353	11	2	2
EmailDomain	590	361	2	8
EmailRootLevelDomain	29	9	1	2

8.2 Show distribution of these labels across the AS Names

[24]:
```
def groupByPivot(group_column_top, group_column_left, count_column):
    gb = df_IPUA_clusters.groupby([group_column_top,␣
 ↪group_column_left])[count_column].count()
    gb.fillna(0, inplace=True)
    gb = gb.unstack(0)
    gb.fillna(0, inplace=True)
    gb['All'] = gb[0] + gb[1] + gb[2] + gb[3] #+ gb[4]
    gb.sort_values('All', ascending=False, inplace=True)
    return gb
```

[25]:
```
groupByPivot('IPUA_Labels', 'AS Name','InboxID')
```

[25]:
IPUA_Labels	0	1	2	\
AS Name				
Amazon.com, Inc.	1158.0	9608.0	40.0	
Unknown	10562.0	0.0	0.0	
AT&T Mobility LLC	1118.0	0.0	0.0	
MCI Communications Services, Inc. Verizon Business	294.0	0.0	0.0	
T-Mobile USA, Inc.	253.0	0.0	0.0	
Scalair SAS	217.0	0.0	0.0	
CenturyLink Communications, LLC	91.0	0.0	0.0	
Google LLC	86.0	0.0	0.0	
Comcast Cable Communications, LLC	81.0	0.0	0.0	
DigitalOcean, LLC	20.0	0.0	0.0	
Level 3 Parent, LLC	54.0	0.0	0.0	
Windstream Communications LLC	31.0	0.0	0.0	
TELUS Communications Inc.	27.0	0.0	0.0	
Shaw Communications Inc.	16.0	0.0	0.0	
Microsoft Corporation	12.0	0.0	0.0	
GENESCO INC	6.0	0.0	0.0	
Johns Hopkins University	4.0	0.0	0.0	

```
DoD Network Information Center                          3.0     0.0   0.0
Sprint                                                  3.0     0.0   0.0
Cellco Partnership DBA Verizon Wireless                 3.0     0.0   0.0
Cogent Communications                                   2.0     0.0   0.0
```

```
IPUA_Labels                                             3      All
AS Name
Amazon.com, Inc.                                    333.0  11139.0
Unknown                                               0.0  10562.0
AT&T Mobility LLC                                     0.0   1118.0
MCI Communications Services, Inc. Verizon Business    0.0    294.0
T-Mobile USA, Inc.                                    0.0    253.0
Scalair SAS                                           0.0    217.0
CenturyLink Communications, LLC                       0.0     91.0
Google LLC                                            0.0     86.0
Comcast Cable Communications, LLC                     0.0     81.0
DigitalOcean, LLC                                    36.0     56.0
Level 3 Parent, LLC                                   0.0     54.0
Windstream Communications LLC                         0.0     31.0
TELUS Communications Inc.                             0.0     27.0
Shaw Communications Inc.                              0.0     16.0
Microsoft Corporation                                 0.0     12.0
GENESCO INC                                           0.0      6.0
Johns Hopkins University                              0.0      4.0
DoD Network Information Center                        0.0      3.0
Sprint                                                0.0      3.0
Cellco Partnership DBA Verizon Wireless               0.0      3.0
Cogent Communications                                 0.0      2.0
```

[26]:
```
groupByPivot('IPUA_Labels', 'AS Number','InboxID')
```

[26]:
```
IPUA_Labels        0       1     2      3      All
AS Number
Unknown      10562.0     0.0   0.0    0.0  10562.0
16509          261.0  9608.0  40.0    0.0   9909.0
14618          897.0     0.0   0.0  333.0   1230.0
20057         1118.0     0.0   0.0    0.0   1118.0
701            294.0     0.0   0.0    0.0    294.0
21928          253.0     0.0   0.0    0.0    253.0
206002         217.0     0.0   0.0    0.0    217.0
209             91.0     0.0   0.0    0.0     91.0
15169           86.0     0.0   0.0    0.0     86.0
7922            81.0     0.0   0.0    0.0     81.0
14061           20.0     0.0   0.0   36.0     56.0
3356            54.0     0.0   0.0    0.0     54.0
7029            31.0     0.0   0.0    0.0     31.0
852             27.0     0.0   0.0    0.0     27.0
```

6327	16.0	0.0	0.0	0.0	16.0
8075	12.0	0.0	0.0	0.0	12.0
20242	6.0	0.0	0.0	0.0	6.0
5723	4.0	0.0	0.0	0.0	4.0
6167	3.0	0.0	0.0	0.0	3.0
721	3.0	0.0	0.0	0.0	3.0
1239	3.0	0.0	0.0	0.0	3.0
174	2.0	0.0	0.0	0.0	2.0

[27]:
```
groupByPivot('IPUA_Labels', 'CIDR Range','InboxID').head(20)
```

[27]:

IPUA_Labels	0	1	2	3	All
CIDR Range					
Unknown	10562.0	0.0	0.0	0.0	10562.0
{4ED21F0C-8573-052A-670B-E4D315133AF5}	75.0	3823.0	0.0	0.0	3898.0
{7EF79BFD-BC3A-903D-C2BD-F5E295ACF4DE}	18.0	1731.0	0.0	0.0	1749.0
{E3615A90-F5C2-C4A8-40A1-2CCC4102B390}	7.0	812.0	0.0	0.0	819.0
{D2185CAC-6E67-82FB-603B-F5B33514B577}	9.0	781.0	0.0	0.0	790.0
{F9164193-183A-7A69-1AE7-25BD6243ED72}	9.0	596.0	0.0	0.0	605.0
{11E5AB29-4623-8156-AB63-E7A2FD82A79A}	6.0	584.0	0.0	0.0	590.0
{60D94A4C-7F09-D0F9-2810-83B93ACBC03E}	12.0	438.0	0.0	0.0	450.0
{7E737FAE-7868-5BE1-84B6-EA9011F3B32E}	33.0	0.0	0.0	333.0	366.0
{6194D26B-CF4C-CA86-F018-CDE54FED8393}	291.0	0.0	0.0	0.0	291.0
{855313BB-3482-1ADA-36A8-5D073B29C265}	2.0	235.0	20.0	0.0	257.0
{6893DEE9-A105-29E9-F9D4-98B3411CF42B}	4.0	249.0	0.0	0.0	253.0
{BF61FA52-8010-2731-2B4E-0C77A6C05E6A}	0.0	222.0	0.0	0.0	222.0
{EEA776DA-8B86-986D-AAD8-0A4A169106A4}	217.0	0.0	0.0	0.0	217.0
{9DA7FFE3-A644-988D-A6E7-0D6305395B19}	216.0	0.0	0.0	0.0	216.0
{71EC7D71-D378-E92E-CE68-0212EA13E3FC}	192.0	0.0	0.0	0.0	192.0
{98A8C5B5-B34B-963A-3240-EE64FE8394FD}	184.0	0.0	0.0	0.0	184.0
{4D1C1E09-9D9F-8960-FEA5-FA478BC0CE02}	1.0	137.0	20.0	0.0	158.0
{801AB1BE-FFF2-B6EF-D3FA-8768D3109CD4}	120.0	0.0	0.0	0.0	120.0
{69CC6342-B73F-0C05-D298-B7EE23E2217E}	114.0	0.0	0.0	0.0	114.0

9

10 Looking at the raw data in SQL, we deteremined:

- **Label 0** - Organic click activity
- **Label 1** - BOT activity from AWS Oregon
- **Label 2** - BOT activity from AWS Oregon
- **Label 3** - BOT activity from AWS Virginia

11 ______________________________

12 Extract Results

12.0.1 Now that we have named the unsupervised labels, we can make our conclusions

13 $ReputationScore = \frac{SessionCount(Label>0)}{SessionCount(AllLabels)}$

14 IPAddress/UserAgent Reputation

[28]:
```
df_IPUA_rep = pd.DataFrame(df_IPUA_clusters.groupby(['IPAddress', 'UserAgent',␣
↪'IPUA_Labels'])['InboxID'].count())
df_IPUA_rep = df_IPUA_rep.unstack('IPUA_Labels').fillna(0)
df_IPUA_rep.columns = df_IPUA_rep.columns.droplevel()
df_IPUA_rep['TotalCount'] = df_IPUA_rep[0] + df_IPUA_rep[1] + df_IPUA_rep[2] +␣
↪df_IPUA_rep[3]
df_IPUA_rep['BotCount'] = df_IPUA_rep[1] + df_IPUA_rep[2] + df_IPUA_rep[3]
df_IPUA_rep['ReputationScore'] = df_IPUA_rep['BotCount'] /␣
↪df_IPUA_rep['TotalCount']
df_IPUA_rep.sort_values('ReputationScore', ascending=False)
```

[28]:
```
IPUA_Labels
0  \
IPAddress                              UserAgent
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
0.0
…
…
{55EC17BD-287C-2F25-173A-298E9A4FEE24} {F87DC183-2EAD-5AED-B536-4545C440750D}
1.0
{55EC7CAB-1C40-F94C-7A67-87C6859B96E6} {CBBC6612-3713-CBB1-C09F-0E774A647016}
1.0
{55F5ECBC-468A-ACE5-B1A6-B3A78D6C7B6C} {F87DC183-2EAD-5AED-B536-4545C440750D}
1.0
{55FAAD1E-483A-5850-8CE4-C33C6096E326} {F87DC183-2EAD-5AED-B536-4545C440750D}
1.0
{FFF90752-2F9B-BAF1-E41F-2C9A924C70D2} {F87DC183-2EAD-5AED-B536-4545C440750D}
```

```
1.0

IPUA_Labels
1  \
IPAddress                                UserAgent
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
143.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
145.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
222.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
76.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
0.0
…
…
{55EC17BD-287C-2F25-173A-298E9A4FEE24} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55EC7CAB-1C40-F94C-7A67-87C6859B96E6} {CBBC6612-3713-CBB1-C09F-0E774A647016}
0.0
{55F5ECBC-468A-ACE5-B1A6-B3A78D6C7B6C} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55FAAD1E-483A-5850-8CE4-C33C6096E326} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{FFF90752-2F9B-BAF1-E41F-2C9A924C70D2} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0

IPUA_Labels
2  \
IPAddress                                UserAgent
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
20.0
…
…
{55EC17BD-287C-2F25-173A-298E9A4FEE24} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55EC7CAB-1C40-F94C-7A67-87C6859B96E6} {CBBC6612-3713-CBB1-C09F-0E774A647016}
0.0
```

```
{55F5ECBC-468A-ACE5-B1A6-B3A78D6C7B6C} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55FAAD1E-483A-5850-8CE4-C33C6096E326} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{FFF90752-2F9B-BAF1-E41F-2C9A924C70D2} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0

IPUA_Labels
3  \
IPAddress                                UserAgent
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
0.0
…
…
{55EC17BD-287C-2F25-173A-298E9A4FEE24} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55EC7CAB-1C40-F94C-7A67-87C6859B96E6} {CBBC6612-3713-CBB1-C09F-0E774A647016}
0.0
{55F5ECBC-468A-ACE5-B1A6-B3A78D6C7B6C} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55FAAD1E-483A-5850-8CE4-C33C6096E326} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{FFF90752-2F9B-BAF1-E41F-2C9A924C70D2} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0

IPUA_Labels
TotalCount  \
IPAddress                                UserAgent
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
143.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
145.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
222.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
76.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
20.0
…
```

```
…
{55EC17BD-287C-2F25-173A-298E9A4FEE24} {F87DC183-2EAD-5AED-B536-4545C440750D}
1.0
{55EC7CAB-1C40-F94C-7A67-87C6859B96E6} {CBBC6612-3713-CBB1-C09F-0E774A647016}
1.0
{55F5ECBC-468A-ACE5-B1A6-B3A78D6C7B6C} {F87DC183-2EAD-5AED-B536-4545C440750D}
1.0
{55FAAD1E-483A-5850-8CE4-C33C6096E326} {F87DC183-2EAD-5AED-B536-4545C440750D}
1.0
{FFF90752-2F9B-BAF1-E41F-2C9A924C70D2} {F87DC183-2EAD-5AED-B536-4545C440750D}
1.0

IPUA_Labels
BotCount  \
IPAddress                                UserAgent
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
143.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
145.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
222.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
76.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
20.0
…
…
{55EC17BD-287C-2F25-173A-298E9A4FEE24} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55EC7CAB-1C40-F94C-7A67-87C6859B96E6} {CBBC6612-3713-CBB1-C09F-0E774A647016}
0.0
{55F5ECBC-468A-ACE5-B1A6-B3A78D6C7B6C} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55FAAD1E-483A-5850-8CE4-C33C6096E326} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{FFF90752-2F9B-BAF1-E41F-2C9A924C70D2} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0

IPUA_Labels
ReputationScore
IPAddress                                UserAgent
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
```

```
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
1.0
…
…
{55EC17BD-287C-2F25-173A-298E9A4FEE24} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55EC7CAB-1C40-F94C-7A67-87C6859B96E6} {CBBC6612-3713-CBB1-C09F-0E774A647016}
0.0
{55F5ECBC-468A-ACE5-B1A6-B3A78D6C7B6C} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{55FAAD1E-483A-5850-8CE4-C33C6096E326} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0
{FFF90752-2F9B-BAF1-E41F-2C9A924C70D2} {F87DC183-2EAD-5AED-B536-4545C440750D}
0.0

[10688 rows x 7 columns]
```

15 CIDR Reputation

Since most BOT requests will use different IP Addresses for their HTTP Requests, we want to score the reputation of the CIDR Range.

This might be too aggressive if the CIDR Range is used by multiple accounts, i.e. NAT belongs to AWS and is used by mutliple accounts.

[29]:
```
df_CIDR_rep = pd.DataFrame(df_IPUA_clusters.groupby(['CIDR Range',␣
↪'IPUA_Labels'])['InboxID'].count())
df_CIDR_rep = df_CIDR_rep.unstack('IPUA_Labels').fillna(0)
df_CIDR_rep.columns = df_CIDR_rep.columns.droplevel()
df_CIDR_rep['TotalCount'] = df_CIDR_rep[0] + df_CIDR_rep[1] + df_CIDR_rep[2] +␣
↪df_CIDR_rep[3]
df_CIDR_rep['BotCount'] = df_CIDR_rep[1] + df_CIDR_rep[2] + df_CIDR_rep[3]
df_CIDR_rep['ReputationScore'] = df_CIDR_rep['BotCount'] /␣
↪df_CIDR_rep['TotalCount']
df_CIDR_rep.sort_values('ReputationScore', ascending=False)
```

[29]:
```
IPUA_Labels                              0      1     2    3  TotalCount  \
CIDR Range
{BF61FA52-8010-2731-2B4E-0C77A6C05E6A}  0.0  222.0   0.0  0.0       222.0
{4D1C1E09-9D9F-8960-FEA5-FA478BC0CE02}  1.0  137.0  20.0  0.0       158.0
{855313BB-3482-1ADA-36A8-5D073B29C265}  2.0  235.0  20.0  0.0       257.0
{E3615A90-F5C2-C4A8-40A1-2CCC4102B390}  7.0  812.0   0.0  0.0       819.0
{11E5AB29-4623-8156-AB63-E7A2FD82A79A}  6.0  584.0   0.0  0.0       590.0
…                                       …    …   …  …         …
{53C74F82-9196-8077-6415-809E96574400}  1.0    0.0   0.0  0.0         1.0
```

```
{536F1AD9-7CB0-BD49-C927-CD9A7895DC2F}  1.0    0.0   0.0  0.0         1.0
{5352877C-27D3-B09C-D4EA-55295EEC1358}  1.0    0.0   0.0  0.0         1.0
{524E0F39-04BE-5781-E6FD-F38096077136}  1.0    0.0   0.0  0.0         1.0
{FEF1369D-B9BF-3EB1-2C8C-30C548F1EC07}  2.0    0.0   0.0  0.0         2.0

IPUA_Labels                             BotCount  ReputationScore
CIDR Range
{BF61FA52-8010-2731-2B4E-0C77A6C05E6A}     222.0         1.000000
{4D1C1E09-9D9F-8960-FEA5-FA478BC0CE02}     157.0         0.993671
{855313BB-3482-1ADA-36A8-5D073B29C265}     255.0         0.992218
{E3615A90-F5C2-C4A8-40A1-2CCC4102B390}     812.0         0.991453
{11E5AB29-4623-8156-AB63-E7A2FD82A79A}     584.0         0.989831
…                                          …              …
{53C74F82-9196-8077-6415-809E96574400}       0.0         0.000000
{536F1AD9-7CB0-BD49-C927-CD9A7895DC2F}       0.0         0.000000
{5352877C-27D3-B09C-D4EA-55295EEC1358}       0.0         0.000000
{524E0F39-04BE-5781-E6FD-F38096077136}       0.0         0.000000
{FEF1369D-B9BF-3EB1-2C8C-30C548F1EC07}       0.0         0.000000

[354 rows x 7 columns]
```

15.0.1 Save lables to csv file for manual analysis with raw non-anonymized data in SQL.

[30]:
```
output = df_IPUA_rep[df_IPUA_rep['ReputationScore'] > 0]
output.to_csv('C:/data/IPUA_Reputation_Scores.csv')
output
```

[30]:
```
IPUA_Labels
0  \
IPAddress                               UserAgent
{0793D471-B890-C0C1-2C0B-DF4882B1D58C} {30634283-D150-237A-87FD-DA097EBBE5B9}
3.0
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{19547553-CF45-F7D9-4A02-394BEED09DC4} {30634283-D150-237A-87FD-DA097EBBE5B9}
5.0
{231BEBFF-35AE-DBE5-0B5B-FC948A6B647C} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
0.0
{41D3D20E-D0C2-80D5-90F7-1DFE369F08E8} {30634283-D150-237A-87FD-DA097EBBE5B9}
2.0
{45EA4474-3101-DAE1-F2C7-AB5DCBDC5C9E} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{47C10F28-63E1-6830-A065-8378B6E54BB9} {30634283-D150-237A-87FD-DA097EBBE5B9}
3.0
```

{49F7F9A3-19F9-6E8A-5896-BC10581E5868} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
{4ADEB3FD-B967-E72B-B0C7-CDA39EA744F0} {30634283-D150-237A-87FD-DA097EBBE5B9}
2.0
{500B9043-A0ED-989D-B645-5B635BAE6841} {30634283-D150-237A-87FD-DA097EBBE5B9}
6.0
{543DBB24-4FA1-DA0B-E6FC-6E8BB392C8BF} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
{663BEEA8-FAB0-8223-D501-75488AA44C8B} {30634283-D150-237A-87FD-DA097EBBE5B9}
4.0
{692FDBEC-0D93-2758-D331-CE66EB83FE72} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{8B00AE4B-9328-B1DC-57BA-17B7D11AFE9F} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{907BAFFD-A66B-2D34-89B1-768E3FCBD951} {30634283-D150-237A-87FD-DA097EBBE5B9}
6.0
{9CBE8C71-0131-336C-5211-6EE9060D8B10} {199313F3-4901-ACA6-FFBA-D8C868D9FA00}
0.0
{9EA619AE-3EC0-3405-1CC9-282C8FBFBDC2} {30634283-D150-237A-87FD-DA097EBBE5B9}
2.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BBB7787E-A8EA-C6AF-4BBD-DBF37A23230A} {30634283-D150-237A-87FD-DA097EBBE5B9}
3.0
{C39D9FAE-0933-7C52-B6C0-837E5BD72598} {30634283-D150-237A-87FD-DA097EBBE5B9}
2.0
{D418BA33-88B2-4F10-0B5B-878B938CABF5} {30634283-D150-237A-87FD-DA097EBBE5B9}
4.0
{DA8A5BA2-BCB6-0C4D-B449-DCD5EF31D732} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{DBE066F2-9F22-A9FC-9B3F-11CE933DB934} {75570E27-930A-E412-95FA-B6F033F3988E}
0.0
{E11DF7EC-7BAF-593B-C8F7-B84EC4255F66} {30634283-D150-237A-87FD-DA097EBBE5B9}
4.0
{E16B4100-C6C6-E5C7-4411-02AD0C4B2D88} {59E0B8D9-09F2-ADC8-9686-9015A09FB3B8}
0.0
{E8D9A46E-8D4D-FAE6-F845-2D16E4CD8720} {30634283-D150-237A-87FD-DA097EBBE5B9}
4.0
{F27685B7-6847-BAEE-2549-CF30545EF54E} {30634283-D150-237A-87FD-DA097EBBE5B9}
3.0
{F65478DB-6A17-9059-1B5A-E144A44D06F4} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.0
{FC7FE18F-0F08-55F9-E6BC-67CB932298EE} {30634283-D150-237A-87FD-DA097EBBE5B9}

```
1.0

IPUA_Labels
1  \
IPAddress                                UserAgent
{0793D471-B890-C0C1-2C0B-DF4882B1D58C} {30634283-D150-237A-87FD-DA097EBBE5B9}
512.0
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
143.0
{19547553-CF45-F7D9-4A02-394BEED09DC4} {30634283-D150-237A-87FD-DA097EBBE5B9}
604.0
{231BEBFF-35AE-DBE5-0B5B-FC948A6B647C} {30634283-D150-237A-87FD-DA097EBBE5B9}
192.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
0.0
{41D3D20E-D0C2-80D5-90F7-1DFE369F08E8} {30634283-D150-237A-87FD-DA097EBBE5B9}
636.0
{45EA4474-3101-DAE1-F2C7-AB5DCBDC5C9E} {30634283-D150-237A-87FD-DA097EBBE5B9}
67.0
{47C10F28-63E1-6830-A065-8378B6E54BB9} {30634283-D150-237A-87FD-DA097EBBE5B9}
290.0
{49F7F9A3-19F9-6E8A-5896-BC10581E5868} {30634283-D150-237A-87FD-DA097EBBE5B9}
313.0
{4ADEB3FD-B967-E72B-B0C7-CDA39EA744F0} {30634283-D150-237A-87FD-DA097EBBE5B9}
577.0
{500B9043-A0ED-989D-B645-5B635BAE6841} {30634283-D150-237A-87FD-DA097EBBE5B9}
764.0
{543DBB24-4FA1-DA0B-E6FC-6E8BB392C8BF} {30634283-D150-237A-87FD-DA097EBBE5B9}
235.0
{663BEEA8-FAB0-8223-D501-75488AA44C8B} {30634283-D150-237A-87FD-DA097EBBE5B9}
305.0
{692FDBEC-0D93-2758-D331-CE66EB83FE72} {30634283-D150-237A-87FD-DA097EBBE5B9}
91.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
222.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
145.0
{8B00AE4B-9328-B1DC-57BA-17B7D11AFE9F} {30634283-D150-237A-87FD-DA097EBBE5B9}
123.0
{907BAFFD-A66B-2D34-89B1-768E3FCBD951} {30634283-D150-237A-87FD-DA097EBBE5B9}
770.0
{9CBE8C71-0131-336C-5211-6EE9060D8B10} {199313F3-4901-ACA6-FFBA-D8C868D9FA00}
0.0
{9EA619AE-3EC0-3405-1CC9-282C8FBFBDC2} {30634283-D150-237A-87FD-DA097EBBE5B9}
513.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
76.0
```

```
{BBB7787E-A8EA-C6AF-4BBD-DBF37A23230A} {30634283-D150-237A-87FD-DA097EBBE5B9}
604.0
{C39D9FAE-0933-7C52-B6C0-837E5BD72598} {30634283-D150-237A-87FD-DA097EBBE5B9}
215.0
{D418BA33-88B2-4F10-0B5B-878B938CABF5} {30634283-D150-237A-87FD-DA097EBBE5B9}
249.0
{DA8A5BA2-BCB6-0C4D-B449-DCD5EF31D732} {30634283-D150-237A-87FD-DA097EBBE5B9}
133.0
{DBE066F2-9F22-A9FC-9B3F-11CE933DB934} {75570E27-930A-E412-95FA-B6F033F3988E}
0.0
{E11DF7EC-7BAF-593B-C8F7-B84EC4255F66} {30634283-D150-237A-87FD-DA097EBBE5B9}
438.0
{E16B4100-C6C6-E5C7-4411-02AD0C4B2D88} {59E0B8D9-09F2-ADC8-9686-9015A09FB3B8}
0.0
{E8D9A46E-8D4D-FAE6-F845-2D16E4CD8720} {30634283-D150-237A-87FD-DA097EBBE5B9}
522.0
{F27685B7-6847-BAEE-2549-CF30545EF54E} {30634283-D150-237A-87FD-DA097EBBE5B9}
584.0
{F65478DB-6A17-9059-1B5A-E144A44D06F4} {30634283-D150-237A-87FD-DA097EBBE5B9}
215.0
{FC7FE18F-0F08-55F9-E6BC-67CB932298EE} {30634283-D150-237A-87FD-DA097EBBE5B9}
70.0

IPUA_Labels
2  \
IPAddress                                UserAgent
{0793D471-B890-C0C1-2C0B-DF4882B1D58C} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{19547553-CF45-F7D9-4A02-394BEED09DC4} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{231BEBFF-35AE-DBE5-0B5B-FC948A6B647C} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
20.0
{41D3D20E-D0C2-80D5-90F7-1DFE369F08E8} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{45EA4474-3101-DAE1-F2C7-AB5DCBDC5C9E} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{47C10F28-63E1-6830-A065-8378B6E54BB9} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{49F7F9A3-19F9-6E8A-5896-BC10581E5868} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{4ADEB3FD-B967-E72B-B0C7-CDA39EA744F0} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{500B9043-A0ED-989D-B645-5B635BAE6841} {30634283-D150-237A-87FD-DA097EBBE5B9}
```

```
0.0
{543DBB24-4FA1-DA0B-E6FC-6E8BB392C8BF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{663BEEA8-FAB0-8223-D501-75488AA44C8B} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{692FDBEC-0D93-2758-D331-CE66EB83FE72} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{8B00AE4B-9328-B1DC-57BA-17B7D11AFE9F} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{907BAFFD-A66B-2D34-89B1-768E3FCBD951} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{9CBE8C71-0131-336C-5211-6EE9060D8B10} {199313F3-4901-ACA6-FFBA-D8C868D9FA00}
0.0
{9EA619AE-3EC0-3405-1CC9-282C8FBFBDC2} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BBB7787E-A8EA-C6AF-4BBD-DBF37A23230A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{C39D9FAE-0933-7C52-B6C0-837E5BD72598} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{D418BA33-88B2-4F10-0B5B-878B938CABF5} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{DA8A5BA2-BCB6-0C4D-B449-DCD5EF31D732} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{DBE066F2-9F22-A9FC-9B3F-11CE933DB934} {75570E27-930A-E412-95FA-B6F033F3988E}
20.0
{E11DF7EC-7BAF-593B-C8F7-B84EC4255F66} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{E16B4100-C6C6-E5C7-4411-02AD0C4B2D88} {59E0B8D9-09F2-ADC8-9686-9015A09FB3B8}
0.0
{E8D9A46E-8D4D-FAE6-F845-2D16E4CD8720} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{F27685B7-6847-BAEE-2549-CF30545EF54E} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{F65478DB-6A17-9059-1B5A-E144A44D06F4} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{FC7FE18F-0F08-55F9-E6BC-67CB932298EE} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0

IPUA_Labels
3  \
IPAddress                               UserAgent
```

{0793D471-B890-C0C1-2C0B-DF4882B1D58C} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{19547553-CF45-F7D9-4A02-394BEED09DC4} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{231BEBFF-35AE-DBE5-0B5B-FC948A6B647C} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
0.0
{41D3D20E-D0C2-80D5-90F7-1DFE369F08E8} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{45EA4474-3101-DAE1-F2C7-AB5DCBDC5C9E} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{47C10F28-63E1-6830-A065-8378B6E54BB9} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{49F7F9A3-19F9-6E8A-5896-BC10581E5868} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{4ADEB3FD-B967-E72B-B0C7-CDA39EA744F0} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{500B9043-A0ED-989D-B645-5B635BAE6841} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{543DBB24-4FA1-DA0B-E6FC-6E8BB392C8BF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{663BEEA8-FAB0-8223-D501-75488AA44C8B} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{692FDBEC-0D93-2758-D331-CE66EB83FE72} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{8B00AE4B-9328-B1DC-57BA-17B7D11AFE9F} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{907BAFFD-A66B-2D34-89B1-768E3FCBD951} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{9CBE8C71-0131-336C-5211-6EE9060D8B10} {199313F3-4901-ACA6-FFBA-D8C868D9FA00}
36.0
{9EA619AE-3EC0-3405-1CC9-282C8FBFBDC2} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{BBB7787E-A8EA-C6AF-4BBD-DBF37A23230A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{C39D9FAE-0933-7C52-B6C0-837E5BD72598} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{D418BA33-88B2-4F10-0B5B-878B938CABF5} {30634283-D150-237A-87FD-DA097EBBE5B9}

```
0.0
{DA8A5BA2-BCB6-0C4D-B449-DCD5EF31D732} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{DBE066F2-9F22-A9FC-9B3F-11CE933DB934} {75570E27-930A-E412-95FA-B6F033F3988E}
0.0
{E11DF7EC-7BAF-593B-C8F7-B84EC4255F66} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{E16B4100-C6C6-E5C7-4411-02AD0C4B2D88} {59E0B8D9-09F2-ADC8-9686-9015A09FB3B8}
333.0
{E8D9A46E-8D4D-FAE6-F845-2D16E4CD8720} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{F27685B7-6847-BAEE-2549-CF30545EF54E} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{F65478DB-6A17-9059-1B5A-E144A44D06F4} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0
{FC7FE18F-0F08-55F9-E6BC-67CB932298EE} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.0

IPUA_Labels
TotalCount  \
IPAddress                                UserAgent
{0793D471-B890-C0C1-2C0B-DF4882B1D58C} {30634283-D150-237A-87FD-DA097EBBE5B9}
515.0
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
143.0
{19547553-CF45-F7D9-4A02-394BEED09DC4} {30634283-D150-237A-87FD-DA097EBBE5B9}
609.0
{231BEBFF-35AE-DBE5-0B5B-FC948A6B647C} {30634283-D150-237A-87FD-DA097EBBE5B9}
193.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
20.0
{41D3D20E-D0C2-80D5-90F7-1DFE369F08E8} {30634283-D150-237A-87FD-DA097EBBE5B9}
638.0
{45EA4474-3101-DAE1-F2C7-AB5DCBDC5C9E} {30634283-D150-237A-87FD-DA097EBBE5B9}
67.0
{47C10F28-63E1-6830-A065-8378B6E54BB9} {30634283-D150-237A-87FD-DA097EBBE5B9}
293.0
{49F7F9A3-19F9-6E8A-5896-BC10581E5868} {30634283-D150-237A-87FD-DA097EBBE5B9}
314.0
{4ADEB3FD-B967-E72B-B0C7-CDA39EA744F0} {30634283-D150-237A-87FD-DA097EBBE5B9}
579.0
{500B9043-A0ED-989D-B645-5B635BAE6841} {30634283-D150-237A-87FD-DA097EBBE5B9}
770.0
{543DBB24-4FA1-DA0B-E6FC-6E8BB392C8BF} {30634283-D150-237A-87FD-DA097EBBE5B9}
236.0
{663BEEA8-FAB0-8223-D501-75488AA44C8B} {30634283-D150-237A-87FD-DA097EBBE5B9}
309.0
```

```
{692FDBEC-0D93-2758-D331-CE66EB83FE72} {30634283-D150-237A-87FD-DA097EBBE5B9}
91.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
222.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
145.0
{8B00AE4B-9328-B1DC-57BA-17B7D11AFE9F} {30634283-D150-237A-87FD-DA097EBBE5B9}
123.0
{907BAFFD-A66B-2D34-89B1-768E3FCBD951} {30634283-D150-237A-87FD-DA097EBBE5B9}
776.0
{9CBE8C71-0131-336C-5211-6EE9060D8B10} {199313F3-4901-ACA6-FFBA-D8C868D9FA00}
36.0
{9EA619AE-3EC0-3405-1CC9-282C8FBFBDC2} {30634283-D150-237A-87FD-DA097EBBE5B9}
515.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
76.0
{BBB7787E-A8EA-C6AF-4BBD-DBF37A23230A} {30634283-D150-237A-87FD-DA097EBBE5B9}
607.0
{C39D9FAE-0933-7C52-B6C0-837E5BD72598} {30634283-D150-237A-87FD-DA097EBBE5B9}
217.0
{D418BA33-88B2-4F10-0B5B-878B938CABF5} {30634283-D150-237A-87FD-DA097EBBE5B9}
253.0
{DA8A5BA2-BCB6-0C4D-B449-DCD5EF31D732} {30634283-D150-237A-87FD-DA097EBBE5B9}
133.0
{DBE066F2-9F22-A9FC-9B3F-11CE933DB934} {75570E27-930A-E412-95FA-B6F033F3988E}
20.0
{E11DF7EC-7BAF-593B-C8F7-B84EC4255F66} {30634283-D150-237A-87FD-DA097EBBE5B9}
442.0
{E16B4100-C6C6-E5C7-4411-02AD0C4B2D88} {59E0B8D9-09F2-ADC8-9686-9015A09FB3B8}
333.0
{E8D9A46E-8D4D-FAE6-F845-2D16E4CD8720} {30634283-D150-237A-87FD-DA097EBBE5B9}
526.0
{F27685B7-6847-BAEE-2549-CF30545EF54E} {30634283-D150-237A-87FD-DA097EBBE5B9}
587.0
{F65478DB-6A17-9059-1B5A-E144A44D06F4} {30634283-D150-237A-87FD-DA097EBBE5B9}
216.0
{FC7FE18F-0F08-55F9-E6BC-67CB932298EE} {30634283-D150-237A-87FD-DA097EBBE5B9}
71.0

IPUA_Labels
BotCount  \
IPAddress                                UserAgent
{0793D471-B890-C0C1-2C0B-DF4882B1D58C} {30634283-D150-237A-87FD-DA097EBBE5B9}
512.0
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
143.0
{19547553-CF45-F7D9-4A02-394BEED09DC4} {30634283-D150-237A-87FD-DA097EBBE5B9}
```

604.0
{231BEBFF-35AE-DBE5-0B5B-FC948A6B647C} {30634283-D150-237A-87FD-DA097EBBE5B9}
192.0
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
20.0
{41D3D20E-D0C2-80D5-90F7-1DFE369F08E8} {30634283-D150-237A-87FD-DA097EBBE5B9}
636.0
{45EA4474-3101-DAE1-F2C7-AB5DCBDC5C9E} {30634283-D150-237A-87FD-DA097EBBE5B9}
67.0
{47C10F28-63E1-6830-A065-8378B6E54BB9} {30634283-D150-237A-87FD-DA097EBBE5B9}
290.0
{49F7F9A3-19F9-6E8A-5896-BC10581E5868} {30634283-D150-237A-87FD-DA097EBBE5B9}
313.0
{4ADEB3FD-B967-E72B-B0C7-CDA39EA744F0} {30634283-D150-237A-87FD-DA097EBBE5B9}
577.0
{500B9043-A0ED-989D-B645-5B635BAE6841} {30634283-D150-237A-87FD-DA097EBBE5B9}
764.0
{543DBB24-4FA1-DA0B-E6FC-6E8BB392C8BF} {30634283-D150-237A-87FD-DA097EBBE5B9}
235.0
{663BEEA8-FAB0-8223-D501-75488AA44C8B} {30634283-D150-237A-87FD-DA097EBBE5B9}
305.0
{692FDBEC-0D93-2758-D331-CE66EB83FE72} {30634283-D150-237A-87FD-DA097EBBE5B9}
91.0
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
222.0
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}
145.0
{8B00AE4B-9328-B1DC-57BA-17B7D11AFE9F} {30634283-D150-237A-87FD-DA097EBBE5B9}
123.0
{907BAFFD-A66B-2D34-89B1-768E3FCBD951} {30634283-D150-237A-87FD-DA097EBBE5B9}
770.0
{9CBE8C71-0131-336C-5211-6EE9060D8B10} {199313F3-4901-ACA6-FFBA-D8C868D9FA00}
36.0
{9EA619AE-3EC0-3405-1CC9-282C8FBFBDC2} {30634283-D150-237A-87FD-DA097EBBE5B9}
513.0
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
76.0
{BBB7787E-A8EA-C6AF-4BBD-DBF37A23230A} {30634283-D150-237A-87FD-DA097EBBE5B9}
604.0
{C39D9FAE-0933-7C52-B6C0-837E5BD72598} {30634283-D150-237A-87FD-DA097EBBE5B9}
215.0
{D418BA33-88B2-4F10-0B5B-878B938CABF5} {30634283-D150-237A-87FD-DA097EBBE5B9}
249.0
{DA8A5BA2-BCB6-0C4D-B449-DCD5EF31D732} {30634283-D150-237A-87FD-DA097EBBE5B9}
133.0
{DBE066F2-9F22-A9FC-9B3F-11CE933DB934} {75570E27-930A-E412-95FA-B6F033F3988E}
20.0

{E11DF7EC-7BAF-593B-C8F7-B84EC4255F66} {30634283-D150-237A-87FD-DA097EBBE5B9}
438.0
{E16B4100-C6C6-E5C7-4411-02AD0C4B2D88} {59E0B8D9-09F2-ADC8-9686-9015A09FB3B8}
333.0
{E8D9A46E-8D4D-FAE6-F845-2D16E4CD8720} {30634283-D150-237A-87FD-DA097EBBE5B9}
522.0
{F27685B7-6847-BAEE-2549-CF30545EF54E} {30634283-D150-237A-87FD-DA097EBBE5B9}
584.0
{F65478DB-6A17-9059-1B5A-E144A44D06F4} {30634283-D150-237A-87FD-DA097EBBE5B9}
215.0
{FC7FE18F-0F08-55F9-E6BC-67CB932298EE} {30634283-D150-237A-87FD-DA097EBBE5B9}
70.0

IPUA_Labels
ReputationScore
IPAddress UserAgent
{0793D471-B890-C0C1-2C0B-DF4882B1D58C} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.994175
{1555FBA3-2FF0-D329-AC22-154C821082CF} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.000000
{19547553-CF45-F7D9-4A02-394BEED09DC4} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.991790
{231BEBFF-35AE-DBE5-0B5B-FC948A6B647C} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.994819
{2EDE7EB8-4F33-998A-C2B0-464296C61AC0} {5437D43E-47AD-69BD-A1D7-D310B07D7D5A}
1.000000
{41D3D20E-D0C2-80D5-90F7-1DFE369F08E8} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.996865
{45EA4474-3101-DAE1-F2C7-AB5DCBDC5C9E} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.000000
{47C10F28-63E1-6830-A065-8378B6E54BB9} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.989761
{49F7F9A3-19F9-6E8A-5896-BC10581E5868} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.996815
{4ADEB3FD-B967-E72B-B0C7-CDA39EA744F0} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.996546
{500B9043-A0ED-989D-B645-5B635BAE6841} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.992208
{543DBB24-4FA1-DA0B-E6FC-6E8BB392C8BF} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.995763
{663BEEA8-FAB0-8223-D501-75488AA44C8B} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.987055
{692FDBEC-0D93-2758-D331-CE66EB83FE72} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.000000
{6AC33884-ADE3-2F3E-6DD7-1731EC5D2945} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.000000
{810FA2C8-982B-6FEE-C299-5CA6CDE57F3D} {30634283-D150-237A-87FD-DA097EBBE5B9}

```
1.000000
{8B00AE4B-9328-B1DC-57BA-17B7D11AFE9F} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.000000
{907BAFFD-A66B-2D34-89B1-768E3FCBD951} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.992268
{9CBE8C71-0131-336C-5211-6EE9060D8B10} {199313F3-4901-ACA6-FFBA-D8C868D9FA00}
1.000000
{9EA619AE-3EC0-3405-1CC9-282C8FBFBDC2} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.996117
{BA3EBEE5-218C-5865-B9D7-E6C93F3C852A} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.000000
{BBB7787E-A8EA-C6AF-4BBD-DBF37A23230A} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.995058
{C39D9FAE-0933-7C52-B6C0-837E5BD72598} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.990783
{D418BA33-88B2-4F10-0B5B-878B938CABF5} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.984190
{DA8A5BA2-BCB6-0C4D-B449-DCD5EF31D732} {30634283-D150-237A-87FD-DA097EBBE5B9}
1.000000
{DBE066F2-9F22-A9FC-9B3F-11CE933DB934} {75570E27-930A-E412-95FA-B6F033F3988E}
1.000000
{E11DF7EC-7BAF-593B-C8F7-B84EC4255F66} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.990950
{E16B4100-C6C6-E5C7-4411-02AD0C4B2D88} {59E0B8D9-09F2-ADC8-9686-9015A09FB3B8}
1.000000
{E8D9A46E-8D4D-FAE6-F845-2D16E4CD8720} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.992395
{F27685B7-6847-BAEE-2549-CF30545EF54E} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.994889
{F65478DB-6A17-9059-1B5A-E144A44D06F4} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.995370
{FC7FE18F-0F08-55F9-E6BC-67CB932298EE} {30634283-D150-237A-87FD-DA097EBBE5B9}
0.985915
```

[31]:
```
output = df_CIDR_rep[df_CIDR_rep['ReputationScore'] > 0]
output.to_csv('C:/data/CIDR_Reputation_Scores.csv')
output
```

[31]:

IPUA_Labels	0	1	2	3	TotalCount	\
CIDR Range						
{11E5AB29-4623-8156-AB63-E7A2FD82A79A}	6.0	584.0	0.0	0.0	590.0	
{4D1C1E09-9D9F-8960-FEA5-FA478BC0CE02}	1.0	137.0	20.0	0.0	158.0	
{4ED21F0C-8573-052A-670B-E4D315133AF5}	75.0	3823.0	0.0	0.0	3898.0	
{60D94A4C-7F09-D0F9-2810-83B93ACBC03E}	12.0	438.0	0.0	0.0	450.0	
{6893DEE9-A105-29E9-F9D4-98B3411CF42B}	4.0	249.0	0.0	0.0	253.0	
{7E737FAE-7868-5BE1-84B6-EA9011F3B32E}	33.0	0.0	0.0	333.0	366.0	
{7EF79BFD-BC3A-903D-C2BD-F5E295ACF4DE}	18.0	1731.0	0.0	0.0	1749.0	

{855313BB-3482-1ADA-36A8-5D073B29C265}	2.0	235.0	20.0	0.0	257.0
{BF61FA52-8010-2731-2B4E-0C77A6C05E6A}	0.0	222.0	0.0	0.0	222.0
{D2185CAC-6E67-82FB-603B-F5B33514B577}	9.0	781.0	0.0	0.0	790.0
{E3615A90-F5C2-C4A8-40A1-2CCC4102B390}	7.0	812.0	0.0	0.0	819.0
{F9164193-183A-7A69-1AE7-25BD6243ED72}	9.0	596.0	0.0	0.0	605.0
{FBDFF399-BA15-4F23-EC40-C8E961EF58A7}	1.0	0.0	0.0	36.0	37.0

IPUA_Labels CIDR Range	BotCount	ReputationScore
{11E5AB29-4623-8156-AB63-E7A2FD82A79A}	584.0	0.989831
{4D1C1E09-9D9F-8960-FEA5-FA478BC0CE02}	157.0	0.993671
{4ED21F0C-8573-052A-670B-E4D315133AF5}	3823.0	0.980759
{60D94A4C-7F09-D0F9-2810-83B93ACBC03E}	438.0	0.973333
{6893DEE9-A105-29E9-F9D4-98B3411CF42B}	249.0	0.984190
{7E737FAE-7868-5BE1-84B6-EA9011F3B32E}	333.0	0.909836
{7EF79BFD-BC3A-903D-C2BD-F5E295ACF4DE}	1731.0	0.989708
{855313BB-3482-1ADA-36A8-5D073B29C265}	255.0	0.992218
{BF61FA52-8010-2731-2B4E-0C77A6C05E6A}	222.0	1.000000
{D2185CAC-6E67-82FB-603B-F5B33514B577}	781.0	0.988608
{E3615A90-F5C2-C4A8-40A1-2CCC4102B390}	812.0	0.991453
{F9164193-183A-7A69-1AE7-25BD6243ED72}	596.0	0.985124
{FBDFF399-BA15-4F23-EC40-C8E961EF58A7}	36.0	0.972973

16 Findings

16.1 When the CIDR Reputation Scores were matched against this dataset, 100% of the suspected BOTs were identified!

But since the BOT determination is subjective, this finding must be taken with a grain of salt. The primary way of identifying BOTs in this dataset is by finding any activity the IP Address traces back to AWS. At this current point, it is very unlikely any organic activity would come from AWS, but this is not a good long-term assumption.

But since unsupervised learning in this notebook only looked at IP Address and User Agent, this bias would not be included in this approach. Random sampling of about 50 rows with high BOT Reputation Scores was done and all pointed to the same known BOT.

There were 36 requests from DigitalOcean, LLC. Digital Ocean is another cloud-based platform. Manually research into these 36 requests revealed that there were from a BOT that is used to harvest coupons from eCommerce's email campaigns. So, the DBSCAN clustering was able to detect this same type of pattern outside of AWS.

17 Future Work

There were 1158 requests with false negatives in that some requests from AWS did not have a Reputation score. When tracking these down, it looked like the request counts were low for these IP Addresses.

Since AWS has thousands and thousands of IP Address and this dataset was only from about 2 days and from a single campaign, chances are these IPs were only used once or twice in this dataset.

Another know BOT was also not detected here, mainly due to the fact the email list did not include this subset of email domains known to be associated with this dataset.

17.1 A larger dataset should be run through this same process

A large dataset (i.e. a full week) might be needed to identify all the CIDRs bound to BOT activity. But this would be 10s of millions of requests. This large a dataset is needed for two primary reasons: - **Heterogenous Data** - Most companies send only once a day, some less. So, a subset of data will miss data from email addresses, campaigns and accounts. - **Number of IP Address** - There are thousands of IP Address for some of these cloud platforms. A subset of small timeframe of data might not show activity on all the IP Addresses.

It would be a simple task to just extend the dataset to a much broader set of requests, but trials on datasets about 4 times larger make DBSCAN not complete due to the time complexity math on the algorithm. So, another clustering model or a new scaling approach would be needed. Some of the possible ways of scaling might be dimension reduction (i.e. PCA) or running the analysis in blocks of time (i.e. 4 hours of requests at a time).