ASA_project_version_7

October 8, 2019

[6]:
```
!pip install pydotplus
```

```
Requirement already satisfied: pydotplus in
/home/nbuser/anaconda3_501/lib/python3.6/site-packages (2.0.2)
Requirement already satisfied: pyparsing>=2.0.1 in
/home/nbuser/anaconda3_501/lib/python3.6/site-packages (from pydotplus) (2.3.0)
WARNING: You are using pip version 19.2.2, however version 19.2.3 is
available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

[7]:
```
!pip install graphviz
```

```
Requirement already satisfied: graphviz in
/home/nbuser/anaconda3_501/lib/python3.6/site-packages (0.11.1)
WARNING: You are using pip version 19.2.2, however version 19.2.3 is
available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

[3]:
```
#Initialise spark
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

[11]:
```
# get raw from github and create pyspark dataframe
from pyspark import SparkFiles
url="https://raw.githubusercontent.com/IPGreene/FW-Neural-net/master/ASA_log.csv"
spark.sparkContext.addFile(url)
data = spark.read.csv(SparkFiles.get("ASA_log.csv"), header=True)
```

[5]:
```
# necessary imports except for keras that will be done within the appropriate␣
 ↪cell
from pyspark.ml import Pipeline
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorAssembler,␣
 ↪StandardScaler, OneHotEncoderEstimator
from pyspark.sql.functions import udf
from pyspark.sql.types import *
from pyspark.sql.functions import rand, countDistinct
from sklearn.model_selection import train_test_split
```

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus
import matplotlib.pyplot as plt
```

[12]:
```
# Data exploration and discovery
# for now we will cast columns to an integer for visualization and inspect data
data.createOrReplaceTempView("firewall")
data = spark.sql('SELECT * FROM firewall')
data = data.withColumn('sourcePort', data['sourcePort'].cast(IntegerType()))
data = data.withColumn('destinationPort', data['destinationPort'].
 ↪cast(IntegerType()))
data = data.withColumn('deviceId', data['deviceId'].cast(IntegerType()))
data = data.withColumn('event_category', data['event_category'].
 ↪cast(IntegerType()))
data = data.withColumn('relevance', data['relevance'].cast(IntegerType()))
data = data.withColumn('credibility', data['credibility'].cast(IntegerType()))
data = data.withColumn('severity', data['severity'].cast(IntegerType()))
data = data.withColumn('magnitude', data['magnitude'].cast(IntegerType()))
data = data.fillna('Unknown')
data.show()
```

```
+----------+--------------+------------+-------------+--------+--------------+-
-----------------+--------------------+---------+-----------+--------+---------
+--------------------+----------+
|sourcePort|destinationPort|protocolName|
IPgeo|deviceId|event_category|categoryDescription|
eventDescription|relevance|credibility|severity|magnitude|
Event_DateTime|eventCount|
+----------+--------------+------------+-------------+--------+--------------+-
-----------------+--------------------+---------+-----------+--------+---------
+--------------------+----------+
|     52217|          2000|      tcp_ip|      Unknown|   31410|          5010|
Misc Exploit|Unrecognized Palo...|        8|         10|       9|        9|Mar
14, 2019, 8:2...|         1|
|     51405|            80|      tcp_ip|      Unknown|   31410|          7024|
Information Leak|A directory trave...|        8|         10|       1|
6|Mar 14, 2019, 8:2...|         1|
|     36002|           445|      tcp_ip|      Unknown|   31410|          7024|
Information Leak|This alert indica...|        8|         10|       1|
6|Mar 14, 2019, 8:2...|         1|
|     35074|           445|      tcp_ip|      Unknown|   31410|          7024|
```

```
Information Leak|This alert indica...|        8|         10|       1|
6|Mar 14, 2019, 8:2...|         1|
|     55631|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|        66|
|     55991|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|        39|
|     49792|             80|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         9|
|     64121|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         4|
|     51662|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         3|
|     50545|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         2|
|     65255|             80|      tcp_ip|United States|     507|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     65025|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64920|            443|      tcp_ip|United States|     507|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64680|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64674|             80|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64672|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64670|             80|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64668|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64576|            443|      tcp_ip|United States|   31410|          4002|
Firewall Permit|Session was allow...|       10|          10|       0|
6|Mar 14, 2019, 8:2...|         1|
|     64533|            443|      tcp_ip|United States|   31410|          4002|
```

```
Firewall Permit|Session was allow...|       10|         10|       0|
6|Mar 14, 2019, 8:2...|         1|
+----------+--------------+-----------+------------+--------+--------------+-
-----------------+--------------------+---------+-----------+--------+---------
+--------------------+----------+
only showing top 20 rows
```

[14]:
```
# manually inspect how many entries of each category exist
# need to remove columns where many categories have counts <3
column_names = data.columns
for c in column_names:
    data.groupby(c).count().show()
```

```
+----------+-----+
|sourcePort|count|
+----------+-----+
|     64121|    1|
|      9900|    1|
|     57380|    1|
|     64822|    1|
|     43256|    1|
|     51388|    1|
|     59086|    1|
|     49686|    1|
|     21058|    1|
|     57412|    1|
|     33118|    1|
|     59115|    1|
|     59832|    1|
|     50610|    1|
|     48308|    1|
|       137|   31|
|     59142|    1|
|     55607|    1|
|     59547|    1|
|     57783|    1|
+----------+-----+
only showing top 20 rows

+---------------+-----+
|destinationPort|count|
+---------------+-----+
|           3000|    1|
|            137|   39|
|          50610|    1|
|             53|  225|
```

```
|           5061|    1|
|           8612|   23|
|            593|    2|
|           8011|    1|
|          44777|    1|
|             81|    3|
|           6667|   10|
|          49154|    3|
|           4899|    1|
|           2323|    1|
|           8318|    6|
|             22|    5|
|          51219|    1|
|          33333|    3|
|           7938|    3|
|           5442|    2|
+---------------+-----+
only showing top 20 rows

+------------+-----+
|protocolName|count|
+------------+-----+
|      tcp_ip|  773|
|     icmp_ip|   31|
|      udp_ip|  696|
+------------+-----+

+--------------+-----+
|         IPgeo|count|
+--------------+-----+
|     Singapore|    3|
|        Europe|    6|
| United States|  431|
|       Unknown| 1046|
|       Denmark|    2|
|       Ireland|    2|
|        Canada|    5|
|        Poland|    1|
|     Australia|    1|
|United Kingdom|    3|
+--------------+-----+

+--------+-----+
|deviceId|count|
+--------+-----+
|   31410| 1312|
|     507|  128|
|   32415|   12|
```

```
|   30096|   40|
|   31175|    8|
+--------+-----+

+--------------+-----+
|event_category|count|
+--------------+-----+
|          5010|    1|
|          4015|   29|
|          4002| 1331|
|          7024|    3|
|          4003|  136|
+--------------+-----+

+-------------------+-----+
|categoryDescription|count|
+-------------------+-----+
|   Information Leak|    3|
|      Access Denied|   29|
|       Misc Exploit|    1|
|    Firewall Permit| 1331|
|      Firewall Deny|  136|
+-------------------+-----+

+--------------------+-----+
|    eventDescription|count|
+--------------------+-----+
|Unrecognized Palo...|    1|
|Session was denie...|  136|
|A directory trave...|    1|
|Session was allow...| 1331|
|Session denied du...|   29|
|This alert indica...|    2|
+--------------------+-----+

+---------+-----+
|relevance|count|
+---------+-----+
|        1|   33|
|        6|  132|
|        3|  128|
|        5|   51|
|        8|  757|
|       10|  399|
+---------+-----+

+-----------+-----+
|credibility|count|
```

```
+-----------+-----+
|          5|  272|
|         10| 1228|
+-----------+-----+

+--------+-----+
|severity|count|
+--------+-----+
|       1|    3|
|       6|    5|
|       9|    1|
|       4|  160|
|       0| 1331|
+--------+-----+

+---------+-----+
|magnitude|count|
+---------+-----+
|        6|  389|
|        3|  136|
|        5|  717|
|        9|    1|
|        4|  126|
|        8|   14|
|        7|   49|
|        2|   68|
+---------+-----+

+--------------------+-----+
|      Event_DateTime|count|
+--------------------+-----+
|Mar 14, 2019, 8:2...|  108|
|Mar 14, 2019, 8:2...|  113|
|Mar 14, 2019, 8:2...|    1|
|Mar 14, 2019, 8:2...|  181|
|Mar 14, 2019, 8:2...|    2|
|Mar 14, 2019, 8:2...|  162|
|Mar 14, 2019, 8:2...|   13|
|Mar 14, 2019, 5:2...|    1|
|Mar 14, 2019, 8:2...|  152|
|Mar 14, 2019, 5:2...|    1|
|Mar 14, 2019, 8:2...|  168|
|Mar 14, 2019, 7:2...|    1|
|Mar 14, 2019, 8:2...|  107|
|Mar 14, 2019, 7:2...|    4|
|Mar 14, 2019, 7:2...|    1|
|Mar 14, 2019, 8:2...|  120|
|Mar 14, 2019, 7:2...|    1|
```

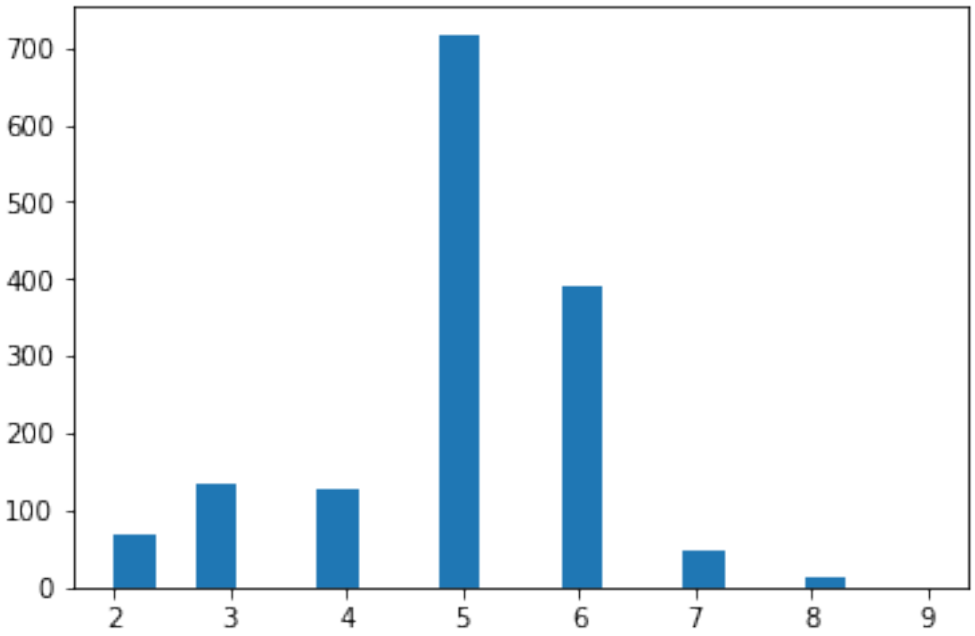
```
|Mar 14, 2019, 5:2...|    4|
|Mar 14, 2019, 5:2...|    2|
|Mar 14, 2019, 7:2...|    4|
+--------------------+-----+
only showing top 20 rows

+----------+-----+
|eventCount|count|
+----------+-----+
|         7|    4|
|        11|    2|
|        42|    1|
|         3|    6|
|         8|    5|
|        28|    2|
|         5|    6|
|        17|    1|
|         6|    4|
|        33|    1|
|         9|    3|
|         1| 1409|
|        10|    1|
|        65|    1|
|         4|    9|
|        39|    1|
|        12|    1|
|        14|    1|
|        66|    1|
|         2|   41|
+----------+-----+
```

[15]:
```
# Remove the appropraite columns
data = data.drop('sourcePort', 'destinationPort', 'Event_DateTime',␣
 ↪'categoryDescription', 'eventDescription', 'eventCount')
```

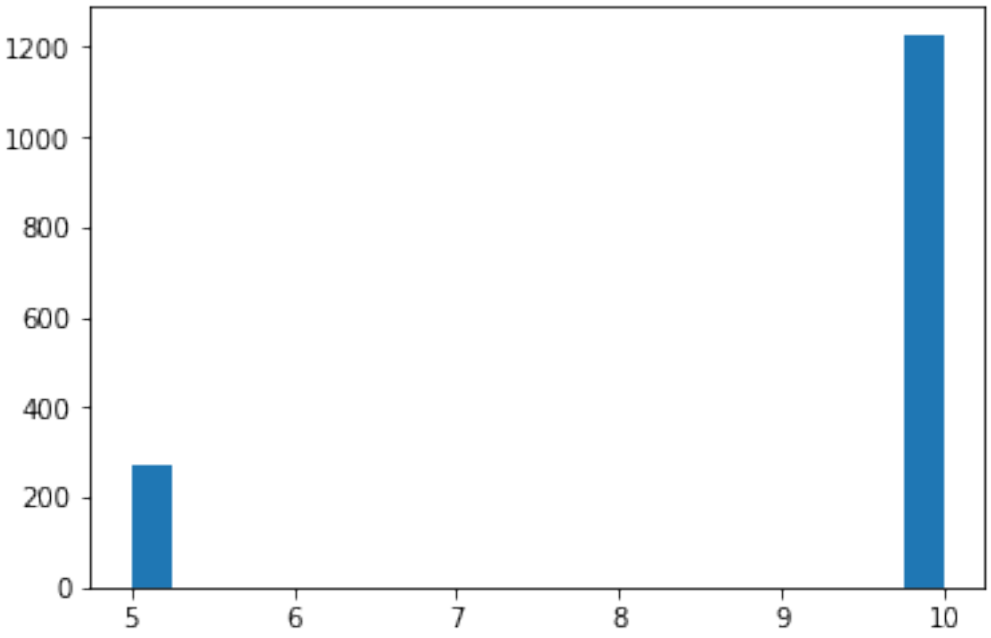
[16]:
```
# make historgrams of columns on a scale
bins, counts = data.select('relevance').rdd.flatMap(lambda x: x).histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[16]:
```
(array([ 68.,   0., 136.,   0.,   0., 126.,   0.,   0., 717.,   0.,   0.,
        389.,   0.,   0.,  49.,   0.,   0.,  14.,   0.,   1.]),
 array([2.  , 2.35, 2.7 , 3.05, 3.4 , 3.75, 4.1 , 4.45, 4.8 , 5.15, 5.5 ,
        5.85, 6.2 , 6.55, 6.9 , 7.25, 7.6 , 7.95, 8.3 , 8.65, 9.  ]),
 <a list of 20 Patch objects>)
```



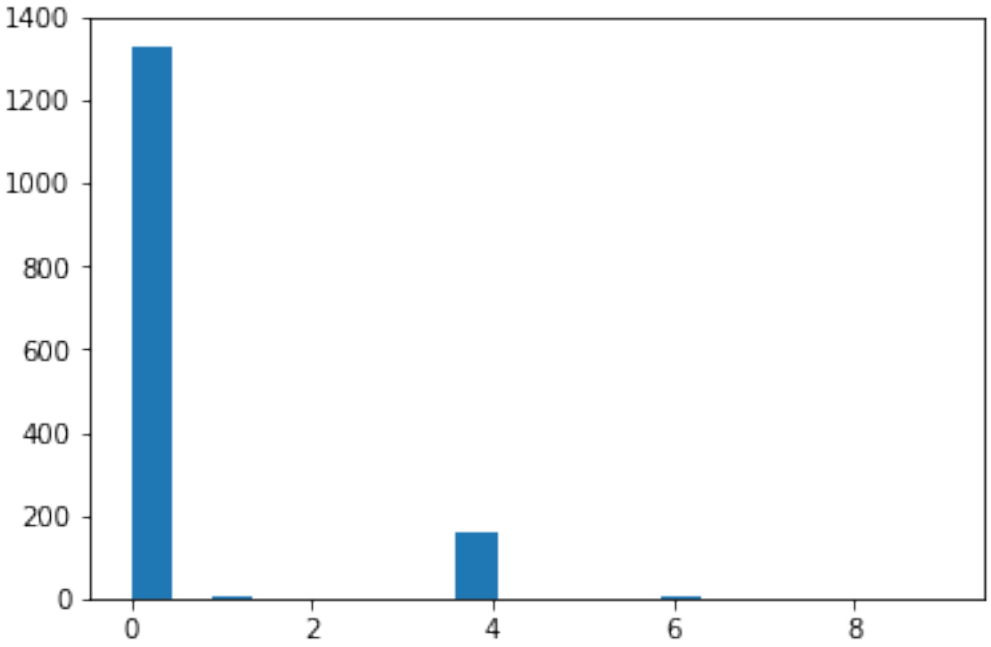
[17]:
```
bins, counts = data.select('credibility').rdd.flatMap(lambda x: x).histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[17]:
```
(array([ 272.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
           0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
           0., 1228.]),
 array([ 5.  ,  5.25,  5.5 ,  5.75,  6.  ,  6.25,  6.5 ,  6.75,  7.  ,
        7.25,  7.5 ,  7.75,  8.  ,  8.25,  8.5 ,  8.75,  9.  ,  9.25,
        9.5 ,  9.75, 10.  ]),
 <a list of 20 Patch objects>)
```



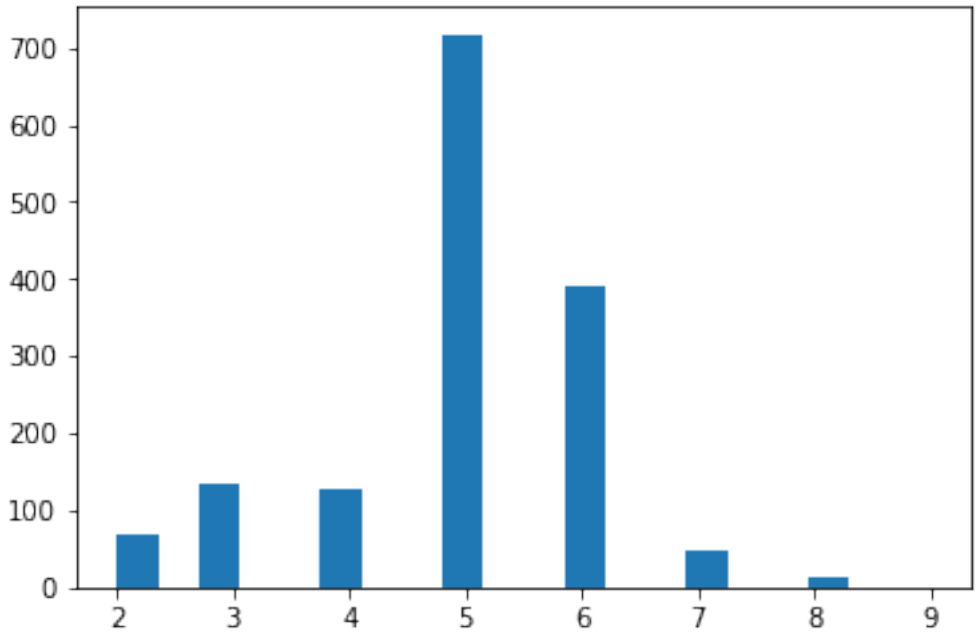
[18]:
```
bins, counts = data.select('severity').rdd.flatMap(lambda x: x).histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[18]:
```
(array([1.331e+03, 0.000e+00, 3.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 1.600e+02, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 5.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 1.000e+00]),
 array([0.  , 0.45, 0.9 , 1.35, 1.8 , 2.25, 2.7 , 3.15, 3.6 , 4.05, 4.5 ,
        4.95, 5.4 , 5.85, 6.3 , 6.75, 7.2 , 7.65, 8.1 , 8.55, 9.  ]),
 <a list of 20 Patch objects>)
```



[19]:
```
bins, counts = data.select('magnitude').rdd.flatMap(lambda x: x).histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[19]:
```
(array([ 68.,   0., 136.,   0.,   0., 126.,   0.,   0., 717.,   0.,   0.,
        389.,   0.,   0.,  49.,   0.,   0.,  14.,   0.,   1.]),
 array([2.  , 2.35, 2.7 , 3.05, 3.4 , 3.75, 4.1 , 4.45, 4.8 , 5.15, 5.5 ,
        5.85, 6.2 , 6.55, 6.9 , 7.25, 7.6 , 7.95, 8.3 , 8.65, 9.  ]),
 <a list of 20 Patch objects>)
```



[20]:
```
# drop appropraite row entries from previous exploration and see how many␣
 ↪entries were lost, we see there were 19
data = data.filter(data.IPgeo != "Canada")
data = data.filter(data.IPgeo != "United Kingdom")
data = data.filter(data.IPgeo != "Australia")
data = data.filter(data.IPgeo != "Poland")
data = data.filter(data.IPgeo != "Ireland")
data = data.filter(data.IPgeo != "Singapore")
#drop 5010 and 7024 event categories
data = data.filter(data.event_category != 5010)
data = data.filter(data.event_category != 7024)

data.count()
```

[20]: 1481

[22]:
```
# since we're assuming time agnostic lets randomize the dataframe
r_data = data.orderBy(rand())
```

[23]:
```
# now we string index our categorical columns
str_col = ['protocolName', 'event_category', 'IPgeo', 'deviceId']
label = 'event_category'
stages = []
for c in str_col:
    indexer = StringIndexer(inputCol=c, outputCol=c+'_index')
    stages += [indexer]
pipeline = Pipeline(stages=stages)
```

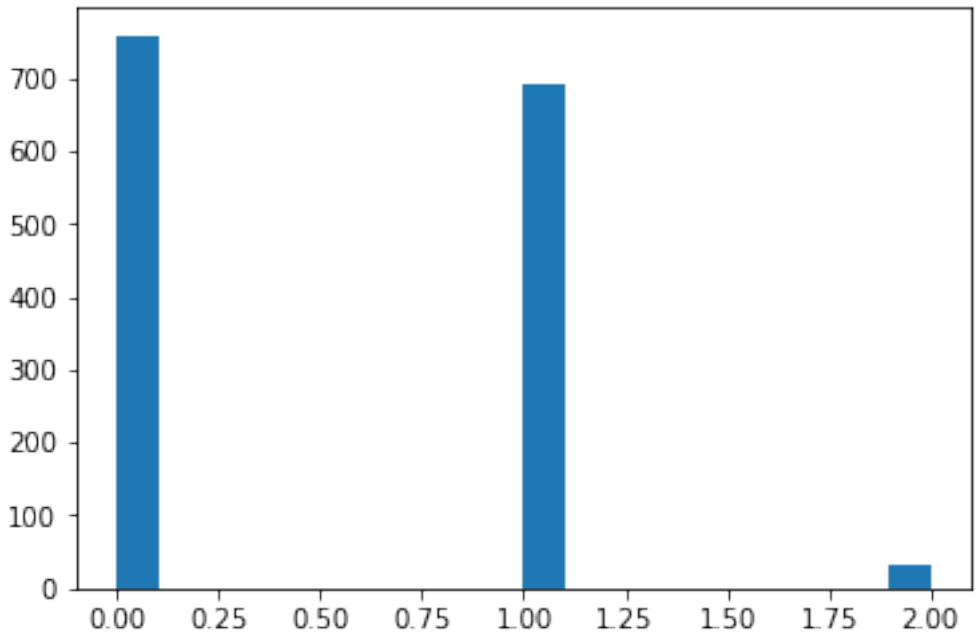
```
model = pipeline.fit(r_data)
transformed = model.transform(r_data)
transformed = transformed.drop('protocolName', 'IPgeo', 'event_category',␣
 ↪'deviceId')
transformed.show()
df = transformed.toPandas()
```

```
+---------+-----------+--------+---------+-----------------+------------------
-+-----------+--------------+
|relevance|credibility|severity|magnitude|protocolName_index|event_category_inde
x|IPgeo_index|deviceId_index|
+---------+-----------+--------+---------+-----------------+------------------
-+-----------+--------------+
|       10|         10|       0|        6|              0.0|
0.0|        1.0|          0.0|
|        8|         10|       0|        5|              1.0|
0.0|        0.0|          0.0|
|       10|         10|       0|        6|              0.0|
0.0|        1.0|          0.0|
|        8|         10|       0|        5|              1.0|
0.0|        0.0|          0.0|
|        8|         10|       0|        5|              1.0|
0.0|        0.0|          0.0|
|        8|         10|       0|        5|              1.0|
0.0|        0.0|          0.0|
|        3|          5|       4|        4|              0.0|
1.0|        0.0|          0.0|
|        3|          5|       4|        4|              0.0|
1.0|        1.0|          0.0|
|        8|         10|       0|        5|              1.0|
0.0|        0.0|          0.0|
|       10|         10|       0|        6|              0.0|
0.0|        1.0|          1.0|
|        8|         10|       0|        5|              0.0|
0.0|        0.0|          0.0|
|        8|         10|       0|        5|              0.0|
0.0|        0.0|          0.0|
|       10|         10|       0|        6|              1.0|
0.0|        0.0|          0.0|
|        8|         10|       0|        5|              0.0|
0.0|        0.0|          0.0|
|        8|         10|       0|        5|              1.0|
0.0|        0.0|          0.0|
|        8|         10|       0|        5|              1.0|
0.0|        0.0|          0.0|
|        8|         10|       0|        5|              0.0|
0.0|        0.0|          0.0|
```

```
|       10|          10|       0|        6|                1.0|
0.0|        0.0|           0.0|
|        8|          10|       0|        5|                0.0|
0.0|        0.0|           0.0|
|        8|          10|       4|        7|                1.0|
1.0|        0.0|           0.0|
+---------+-----------+--------+---------+------------------+-------------------
-+-----------+--------------+
only showing top 20 rows
```

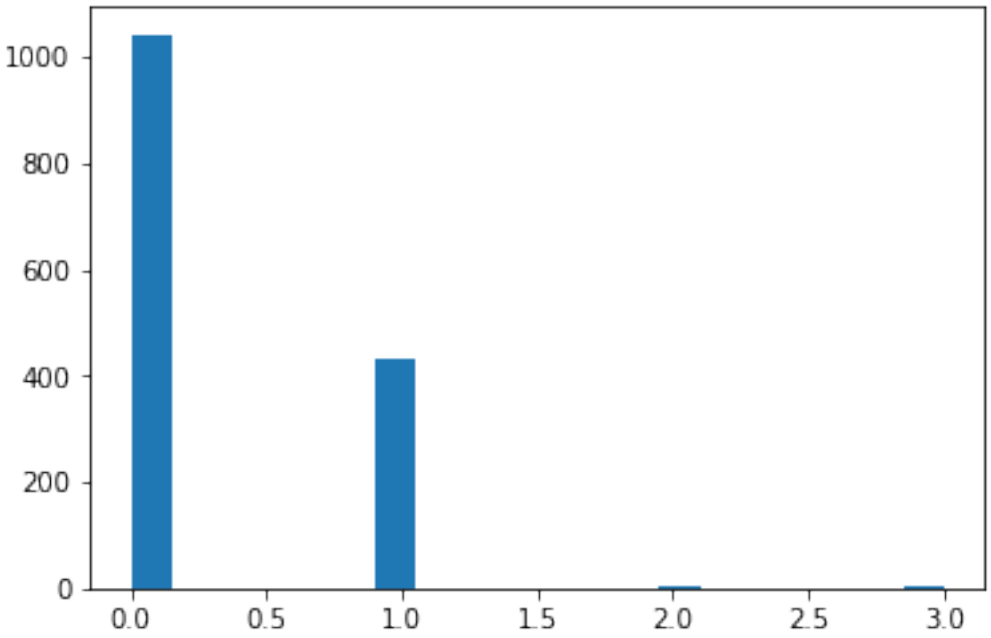
[25]:
```
# histograms of the columns we skipped earlier
bins, counts = transformed.select('protocolName_index').rdd.flatMap(lambda x: x).
 ↪histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[25]:
```
(array([758.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 692.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  31.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
        1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. ]),
 <a list of 20 Patch objects>)
```



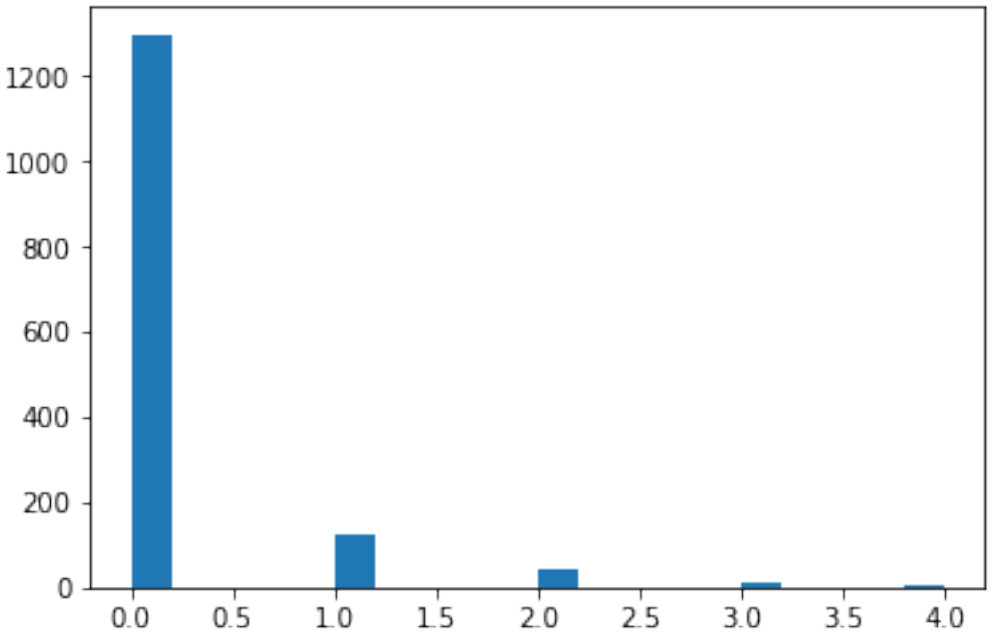
[26]:
```
bins, counts = transformed.select('IPgeo_index').rdd.flatMap(lambda x: x).
 ↪histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[26]:
```
(array([1042.,    0.,    0.,    0.,    0.,    0.,  431.,    0.,    0.,
           0.,    0.,    0.,    0.,    6.,    0.,    0.,    0.,    0.,
           0.,    2.]),
 array([0.  , 0.15, 0.3 , 0.45, 0.6 , 0.75, 0.9 , 1.05, 1.2 , 1.35, 1.5 ,
        1.65, 1.8 , 1.95, 2.1 , 2.25, 2.4 , 2.55, 2.7 , 2.85, 3.  ]),
 <a list of 20 Patch objects>)
```



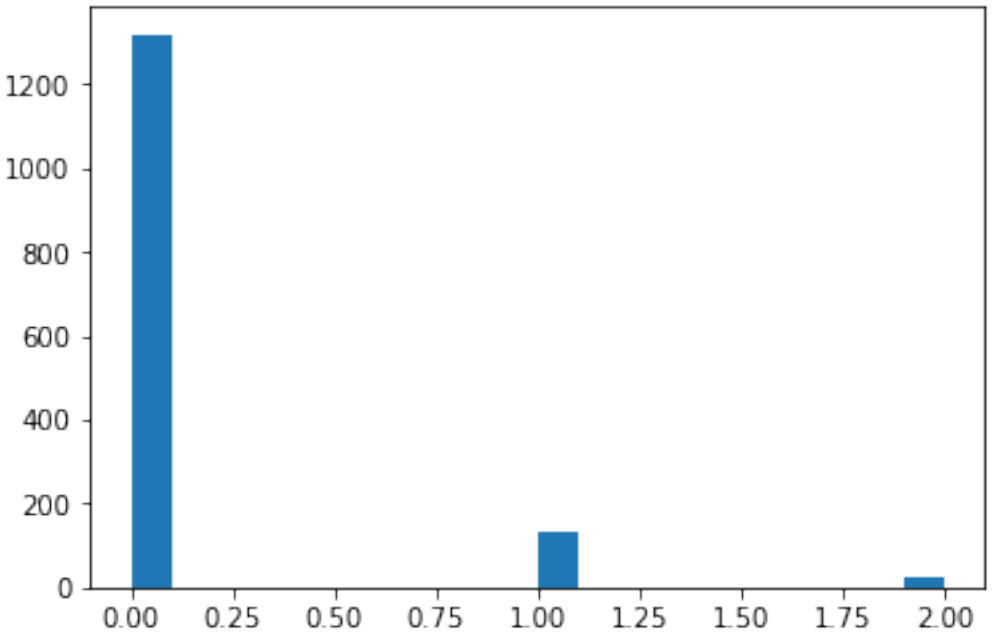
[28]:
```
bins, counts = transformed.select('deviceId_index').rdd.flatMap(lambda x: x).
 ↪histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[28]:
```
(array([1299.,    0.,    0.,    0.,    0.,  123.,    0.,    0.,    0.,
           0.,   40.,    0.,    0.,    0.,    0.,   11.,    0.,    0.,
           0.,    8.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4,
        2.6, 2.8, 3. , 3.2, 3.4, 3.6, 3.8, 4. ]),
 <a list of 20 Patch objects>)
```



[29]:
```
bins, counts = transformed.select('event_category_index').rdd.flatMap(lambda x:␣
 ↪x).histogram(20)
plt.hist(bins[:-1], bins=bins, weights=counts)
```

[29]:
```
(array([1319.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
          0.,  135.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
          0.,   27.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
        1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. ]),
 <a list of 20 Patch objects>)
```



[34]:
```
#not lets do our test train split
train, test = transformed.randomSplit([0.70, 0.30], seed=1234)
x_train = train.drop('event_category_index')
y_train = train.select('event_category_index')
x_test = test.drop('event_category_index')
y_test = test.select('event_category_index')
```

[35]:
```
# manually inspect how many entries of each category exist
#we see we are mssing one category in x_test however there aren't any categories␣
 ↪in test that arent in train
traintest_names = x_train.columns
for c in traintest_names:
    x_train.groupby(c).count().show()
    x_test.groupby(c).count().show()
```

```
+---------+-----+
|relevance|count|
+---------+-----+
|        1|   22|
|        6|   89|
|        3|   85|
|        5|   36|
|        8|  541|
|       10|  280|
+---------+-----+
```

```
+---------+-----+
|relevance|count|
+---------+-----+
|        1|   11|
|        6|   38|
|        3|   43|
|        5|   15|
|        8|  212|
|       10|  109|
+---------+-----+

+-----------+-----+
|credibility|count|
+-----------+-----+
|          5|  185|
|         10|  868|
+-----------+-----+

+-----------+-----+
|credibility|count|
+-----------+-----+
|          5|   82|
|         10|  346|
+-----------+-----+

+--------+-----+
|severity|count|
+--------+-----+
|       6|    3|
|       4|   98|
|       0|  952|
+--------+-----+

+--------+-----+
|severity|count|
+--------+-----+
|       6|    2|
|       4|   59|
|       0|  367|
+--------+-----+

+---------+-----+
|magnitude|count|
+---------+-----+
|        6|  276|
|        3|   94|
|        5|  519|
|        4|   79|
```

```
|        8|    5|
|        7|   31|
|        2|   49|
+---------+-----+

+---------+-----+
|magnitude|count|
+---------+-----+
|        6|  103|
|        3|   37|
|        5|  198|
|        4|   47|
|        8|    6|
|        7|   18|
|        2|   19|
+---------+-----+

+------------------+-----+
|protocolName_index|count|
+------------------+-----+
|               0.0|  535|
|               1.0|  496|
|               2.0|   22|
+------------------+-----+

+------------------+-----+
|protocolName_index|count|
+------------------+-----+
|               0.0|  223|
|               1.0|  196|
|               2.0|    9|
+------------------+-----+

+-----------+-----+
|IPgeo_index|count|
+-----------+-----+
|        0.0|  743|
|        1.0|  304|
|        3.0|    2|
|        2.0|    4|
+-----------+-----+

+-----------+-----+
|IPgeo_index|count|
+-----------+-----+
|        0.0|  299|
|        1.0|  127|
|        2.0|    2|
```

```
+-----------+-----+

+--------------+-----+
|deviceId_index|count|
+--------------+-----+
|           0.0|  918|
|           1.0|   92|
|           4.0|    6|
|           3.0|    9|
|           2.0|   28|
+--------------+-----+

+--------------+-----+
|deviceId_index|count|
+--------------+-----+
|           0.0|  381|
|           1.0|   31|
|           4.0|    2|
|           3.0|    2|
|           2.0|   12|
+--------------+-----+
```

[37]:
```
#same for target
y_train.groupby('event_category_index').count().show()
y_test.groupby('event_category_index').count().show()
```

```
+--------------------+-----+
|event_category_index|count|
+--------------------+-----+
|                 0.0|  952|
|                 1.0|   83|
|                 2.0|   18|
+--------------------+-----+

+--------------------+-----+
|event_category_index|count|
+--------------------+-----+
|                 0.0|  367|
|                 1.0|   52|
|                 2.0|    9|
+--------------------+-----+
```

[38]:
```
# move it all to numpy
x_train_pd = x_train.toPandas()
y_train_pd = y_train.toPandas()
```

```
x_test_pd = x_test.toPandas()
y_test_pd = y_test.toPandas()
y_train_np = y_train_pd.as_matrix()
y_test_np = y_test_pd.as_matrix()
x_train_np = x_train_pd.as_matrix()
x_test_np = x_test_pd.as_matrix()
```

```
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/ipykernel/__main__.py:6:
FutureWarning: Method .as_matrix will be removed in a future version. Use
.values instead.
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/ipykernel/__main__.py:7:
FutureWarning: Method .as_matrix will be removed in a future version. Use
.values instead.
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/ipykernel/__main__.py:8:
FutureWarning: Method .as_matrix will be removed in a future version. Use
.values instead.
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/ipykernel/__main__.py:9:
FutureWarning: Method .as_matrix will be removed in a future version. Use
.values instead.
```

[39]:
```
# one hot encode target
from keras.utils import to_categorical
y_train_2 = to_categorical(y_train_np)
y_test_2 = to_categorical(y_test_np)
y_train_2[0:5]
```

```
Using TensorFlow backend.
```

[39]:
```
array([[1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.]], dtype=float32)
```

[40]:
```
# build keras model
from keras.models import Sequential
from keras.layers import Dense
from keras.callbacks import EarlyStopping
#create model
model = Sequential()

#get number of columns in training data
n_cols = x_train_pd.shape[1]
print(n_cols)
#add model layers
model.add(Dense(250, activation='relu', input_shape=(n_cols,)))
model.add(Dense(250, activation='relu'))
```

```
model.add(Dense(250, activation='relu'))
model.add(Dense(3, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
#set early stopping monitor so the model stops training when it won't improve␣
 ↪anymore
early_stopping_monitor = EarlyStopping(patience=3)
model.summary()
#train model
model.fit(x_train_np, y_train_2, batch_size=64, validation_split=0.2, epochs=30)
```

```
7
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 250)               2000
_________________________________________________________________
dense_2 (Dense)              (None, 250)               62750
_________________________________________________________________
dense_3 (Dense)              (None, 250)               62750
_________________________________________________________________
dense_4 (Dense)              (None, 3)                 753
=================================================================
Total params: 128,253
Trainable params: 128,253
Non-trainable params: 0
_________________________________________________________________
Train on 842 samples, validate on 211 samples
Epoch 1/30
842/842 [==============================] - 6s 7ms/step - loss: 0.2494 - acc:
0.9133 - val_loss: 0.1223 - val_acc: 0.9810
Epoch 2/30
842/842 [==============================] - 0s 457us/step - loss: 0.0686 - acc:
0.9834 - val_loss: 0.0554 - val_acc: 0.9810
Epoch 3/30
842/842 [==============================] - 0s 418us/step - loss: 0.0444 - acc:
0.9834 - val_loss: 0.0477 - val_acc: 0.9810
Epoch 4/30
842/842 [==============================] - 0s 457us/step - loss: 0.0347 - acc:
0.9834 - val_loss: 0.0456 - val_acc: 0.9810
Epoch 5/30
842/842 [==============================] - 0s 427us/step - loss: 0.0321 - acc:
0.9846 - val_loss: 0.0415 - val_acc: 0.9810
Epoch 6/30
842/842 [==============================] - 0s 418us/step - loss: 0.0305 - acc:
0.9869 - val_loss: 0.0437 - val_acc: 0.9810
Epoch 7/30
```

```
842/842 [==============================] - 0s 519us/step - loss: 0.0268 - acc:
0.9857 - val_loss: 0.0431 - val_acc: 0.9810
Epoch 8/30
842/842 [==============================] - 0s 442us/step - loss: 0.0299 - acc:
0.9857 - val_loss: 0.0464 - val_acc: 0.9763
Epoch 9/30
842/842 [==============================] - 0s 418us/step - loss: 0.0306 - acc:
0.9822 - val_loss: 0.0379 - val_acc: 0.9810
Epoch 10/30
842/842 [==============================] - 0s 445us/step - loss: 0.0264 - acc:
0.9786 - val_loss: 0.0373 - val_acc: 0.9810
Epoch 11/30
842/842 [==============================] - 0s 422us/step - loss: 0.0237 - acc:
0.9857 - val_loss: 0.0356 - val_acc: 0.9716
Epoch 12/30
842/842 [==============================] - 0s 475us/step - loss: 0.0226 - acc:
0.9846 - val_loss: 0.0422 - val_acc: 0.9810
Epoch 13/30
842/842 [==============================] - 0s 509us/step - loss: 0.0224 - acc:
0.9869 - val_loss: 0.0348 - val_acc: 0.9716
Epoch 14/30
842/842 [==============================] - 0s 529us/step - loss: 0.0223 - acc:
0.9881 - val_loss: 0.0369 - val_acc: 0.9810
Epoch 15/30
842/842 [==============================] - 0s 443us/step - loss: 0.0222 - acc:
0.9869 - val_loss: 0.0405 - val_acc: 0.9810
Epoch 16/30
842/842 [==============================] - 0s 444us/step - loss: 0.0226 - acc:
0.9881 - val_loss: 0.0381 - val_acc: 0.9810
Epoch 17/30
842/842 [==============================] - 0s 373us/step - loss: 0.0204 - acc:
0.9917 - val_loss: 0.0506 - val_acc: 0.9810
Epoch 18/30
842/842 [==============================] - 0s 425us/step - loss: 0.0231 - acc:
0.9881 - val_loss: 0.0404 - val_acc: 0.9763
Epoch 19/30
842/842 [==============================] - 0s 510us/step - loss: 0.0238 - acc:
0.9857 - val_loss: 0.0443 - val_acc: 0.9810
Epoch 20/30
842/842 [==============================] - 0s 391us/step - loss: 0.0246 - acc:
0.9869 - val_loss: 0.0342 - val_acc: 0.9810
Epoch 21/30
842/842 [==============================] - 0s 552us/step - loss: 0.0208 - acc:
0.9905 - val_loss: 0.0420 - val_acc: 0.9810
Epoch 22/30
842/842 [==============================] - 0s 375us/step - loss: 0.0207 - acc:
0.9857 - val_loss: 0.0328 - val_acc: 0.9810
Epoch 23/30
```

```
842/842 [==============================] - 0s 407us/step - loss: 0.0183 - acc:
0.9905 - val_loss: 0.0368 - val_acc: 0.9905
Epoch 24/30
842/842 [==============================] - 0s 382us/step - loss: 0.0187 - acc:
0.9893 - val_loss: 0.0339 - val_acc: 0.9810
Epoch 25/30
842/842 [==============================] - 0s 407us/step - loss: 0.0206 - acc:
0.9893 - val_loss: 0.0337 - val_acc: 0.9810
Epoch 26/30
842/842 [==============================] - 0s 523us/step - loss: 0.0541 - acc:
0.9786 - val_loss: 0.0591 - val_acc: 0.9810
Epoch 27/30
842/842 [==============================] - 0s 391us/step - loss: 0.0277 - acc:
0.9869 - val_loss: 0.0332 - val_acc: 0.9810
Epoch 28/30
842/842 [==============================] - 0s 383us/step - loss: 0.0236 - acc:
0.9893 - val_loss: 0.0378 - val_acc: 0.9810
Epoch 29/30
842/842 [==============================] - 0s 395us/step - loss: 0.0263 - acc:
0.9857 - val_loss: 0.0318 - val_acc: 0.9810
Epoch 30/30
842/842 [==============================] - 0s 431us/step - loss: 0.0205 - acc:
0.9917 - val_loss: 0.0378 - val_acc: 0.9905
```

[40]:
```
<keras.callbacks.History at 0x7fc9b017e128>
```

[41]:
```
# look at predictions
test_y_predictions = model.predict(x_test_pd)
print(np.around(test_y_predictions, decimals=1))
#test_y_predictions.around([0:100], decimals=1)
```

```
[[0.  0.9 0.1]
 [0.  0.6 0.4]
 [1.  0.  0. ]
 ...
 [1.  0.  0. ]
 [1.  0.  0. ]
 [0.  1.  0. ]]
```

[42]:
```
# decision tree classifier via scikit
dt_df = df
x_cols = ['relevance', 'credibility', 'severity', 'magnitude',␣
 ↪'protocolName_index', 'IPgeo_index', 'deviceId_index']
y_cols = ['event_category_index']
x = dt_df[x_cols]
y = dt_df[y_cols]
x_train_sk, x_test_sk, y_train_sk, y_test_sk = train_test_split(x, y,␣
 ↪test_size=0.3, random_state=1)
```

[43]:
```
# decision tree classifier
dtclf = DecisionTreeClassifier()
dtclf = dtclf.fit(x_train_sk, y_train_sk)
y_pred_sk = dtclf.predict(x_test_sk)
```

[44]:
```
#decision tree classifier accuracy
print("Accuracy:",metrics.accuracy_score(y_test_sk, y_pred_sk))
```

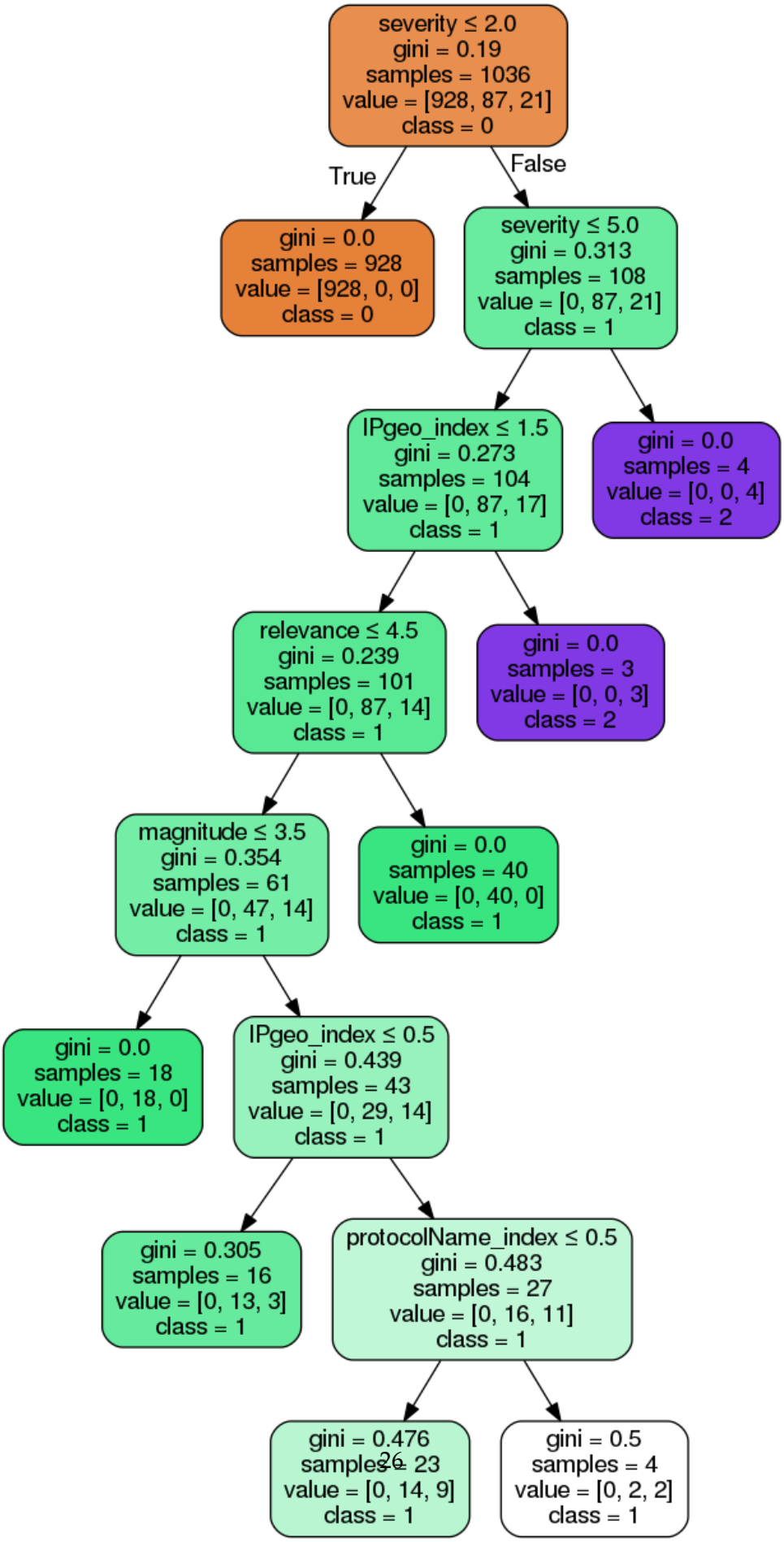
```
Accuracy: 0.9887640449438202
```

[45]:
```
# Create Decision Tree classifer using entropy measure
dtclf1 = DecisionTreeClassifier(criterion="entropy", max_depth=3)
dtclf1 = dtclf1.fit(x_train_sk,y_train_sk)
y_pred_sk1 = dtclf1.predict(x_test_sk)
# Model Accuracy
print("Accuracy:",metrics.accuracy_score(y_test_sk, y_pred_sk1))
```

```
Accuracy: 0.9887640449438202
```

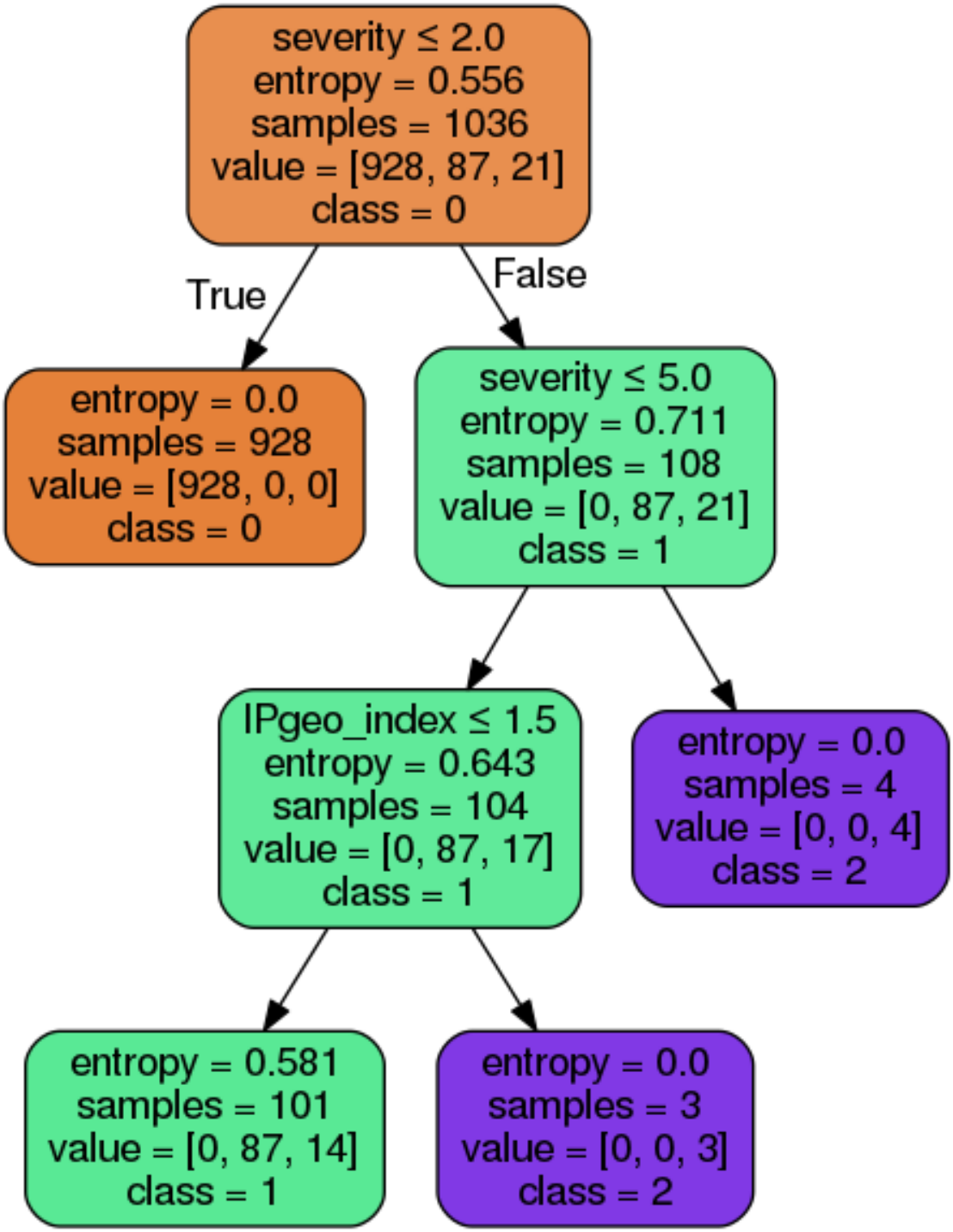
[46]:
```
#decision tree classifier graph gini
dot_data = StringIO()
export_graphviz(dtclf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names =␣
 ↪x_cols,class_names=['0','1','2'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('firewall.png')
Image(graph.create_png())
```

[46]:


severity ≤ 2.0
gini = 0.19
samples = 1036
value = [928, 87, 21]
class = 0
True
False
gini = 0.0
samples = 928
value = [928, 0, 0]
class = 0
severity ≤ 5.0
gini = 0.313
samples = 108
value = [0, 87, 21]
class = 1
IPgeo_index ≤ 1.5
gini = 0.273
samples = 104
value = [0, 87, 17]
class = 1
gini = 0.0
samples = 4
value = [0, 0, 4]
class = 2
relevance ≤ 4.5
gini = 0.239
samples = 101
value = [0, 87, 14]
class = 1
gini = 0.0
samples = 3
value = [0, 0, 3]
class = 2
magnitude ≤ 3.5
gini = 0.354
samples = 61
value = [0, 47, 14]
class = 1
gini = 0.0
samples = 40
value = [0, 40, 0]
class = 1
gini = 0.0
samples = 18
value = [0, 18, 0]
class = 1
IPgeo_index ≤ 0.5
gini = 0.439
samples = 43
value = [0, 29, 14]
class = 1
gini = 0.305
samples = 16
value = [0, 13, 3]
class = 1
protocolName_index ≤ 0.5
gini = 0.483
samples = 27
value = [0, 16, 11]
class = 1
gini = 0.476
samples = 23
value = [0, 14, 9]
class = 1
gini = 0.5
samples = 4
value = [0, 2, 2]
class = 1

[47]:
```
#Decision tree classifier graph entropy
dot_data1 = StringIO()
export_graphviz(dtclf1, out_file=dot_data1,
                filled=True, rounded=True,
                special_characters=True, feature_names =␣
 ↪x_cols,class_names=['0','1','2'])
graph1 = pydotplus.graph_from_dot_data(dot_data1.getvalue())
graph1.write_png('firewall1.png')
Image(graph1.create_png())
```

[47]:


severity ≤ 2.0
entropy = 0.556
samples = 1036
value = [928, 87, 21]
class = 0
True
False
entropy = 0.0
samples = 928
value = [928, 0, 0]
class = 0
severity ≤ 5.0
entropy = 0.711
samples = 108
value = [0, 87, 21]
class = 1
IPgeo_index ≤ 1.5
entropy = 0.643
samples = 104
value = [0, 87, 17]
class = 1
entropy = 0.0
samples = 4
value = [0, 0, 4]
class = 2
entropy = 0.581
samples = 101
value = [0, 87, 14]
class = 1
entropy = 0.0
samples = 3
value = [0, 0, 3]
class = 2

[ ]: