# Warming up for Spark Streaming Project:

**from** **pyspark** **import** SparkContext

**from** **pyspark.streaming** **import** StreamingContext

*# Create a local StreamingContext with two working thread and batch interval of 1 second*

sc = SparkContext("local[2]", "NetworkWordCount")

ssc = StreamingContext(sc, 1)

*# Create a DStream that will connect to hostname:port, like localhost:9999*

*# Firewalls might block this!*

lines = ssc.socketTextStream("localhost", 9999)

*# Split each line into words* words = lines.flatMap(**lambda** line: line.split(" "))

*# Count each word in each batch*

pairs = words.map(**lambda** word: (word, 1))

wordCounts = pairs.reduceByKey(**lambda** x, y: x + y)

*# Print the first ten elements of each RDD generated in this DStream to the console*

wordCounts.pprint()

Now we open up a Unix terminal and type:

$ nc -lk 9999

$ hello world any text you want

With this running run the line below, then type Ctrl+C to terminate it.

ssc.start() *# Start the computation*

ssc.awaitTermination() *# Wait for the computation to terminate*

# Yammer Feed

In order to use all of this though, we need to setup a Developer API acocunt with yammer and create an application to get credentials. Connect with administrator of social media account to get API authentication key, and head on to <https://developer.yammer.com/docs> for detailed instruction.

Once you have that you also need to install yampy, a python library to connect your Python to the Yammer dev account. You probably won't be able to run this example and then previous in the same notebook, you need to restart you kernel.

Let's get started! Begin by running the YammerRead.py file. Make sure to add your own IP Adress and your credential keys.

For Details on yampy visit below links:

* <https://developer.yammer.com/docs/python-sdk>
* <https://pythonhosted.org/yampy/quickstart.html>

*# May cause deprecation warnings, safe to ignore, they aren't errors*

**from** **pyspark** **import** SparkContext

**from** **pyspark.streaming** **import** StreamingContext

**from** **pyspark.sql** **import** SQLContext

**from** **pyspark.sql.functions** **import** desc

*# Can only run this once. restart your kernel for any errors.*

sc = SparkContext()

ssc = StreamingContext(sc, 10 )

sqlContext = SQLContext(sc)

socket\_stream = ssc.socketTextStream("127.0.0.1", 5555)

lines = socket\_stream.window( 20 )

**from** **collections** **import** namedtuple

fields = ("tag", "count" )

Message = namedtuple( 'Message', fields )

*# Use Parenthesis for multiple lines or use \.*

( lines.flatMap( **lambda** text: text.split( " " ) ) *#Splits to a list*

.filter( **lambda** word: word.lower().startswith("#") ) *# Checks for hashtag calls*

.map( **lambda** word: ( word.lower(), 1 ) ) *# Lower cases the word*

.reduceByKey( **lambda** a, b: a + b ) *# Reduces*

.map( **lambda** rec: Message( rec[0], rec[1] ) ) *# Stores in a Message Object*

.foreachRDD( **lambda** rdd: rdd.toDF().sort( desc("count") ) *# Sorts Them in a DF*

.limit(10).registerTempTable("tweets") ) ) *# Registers to a table.*

**Now run TweetRead.py**

ssc.start()

**import** **time**

**from** **IPython** **import** display

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**

*# Only works for Jupyter Notebooks!*

%matplotlib inline

count = 0

**while** count < 10:

time.sleep( 3 )

top\_10\_messages = sqlContext.sql( 'Select tag, count from messages' )

top\_10\_df = top\_10\_messages.toPandas()

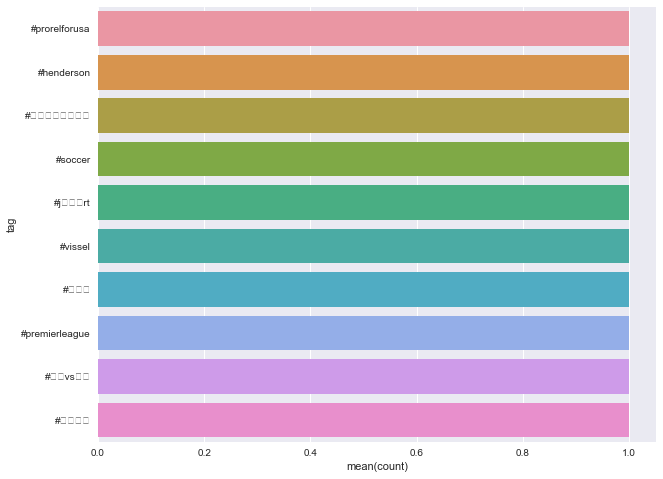
display.clear\_output(wait=**True**)

sns.plt.figure( figsize = ( 10, 8 ) )

sns.barplot( x="count", y="tag", data=top\_10\_df)

sns.plt.show()

count = count + 1



ssc.stop()