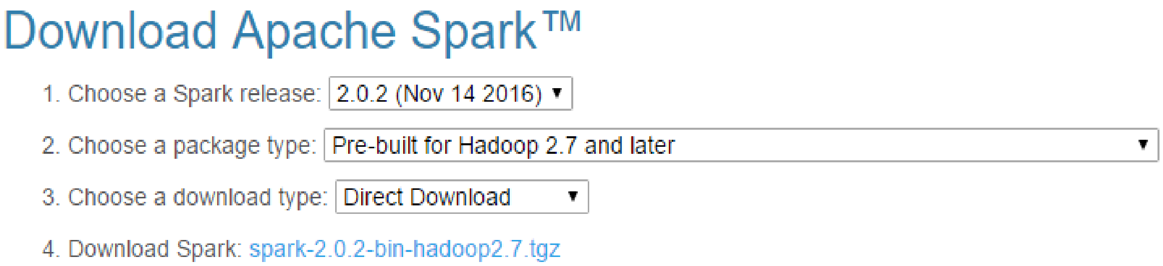
Steps to install spark on window:

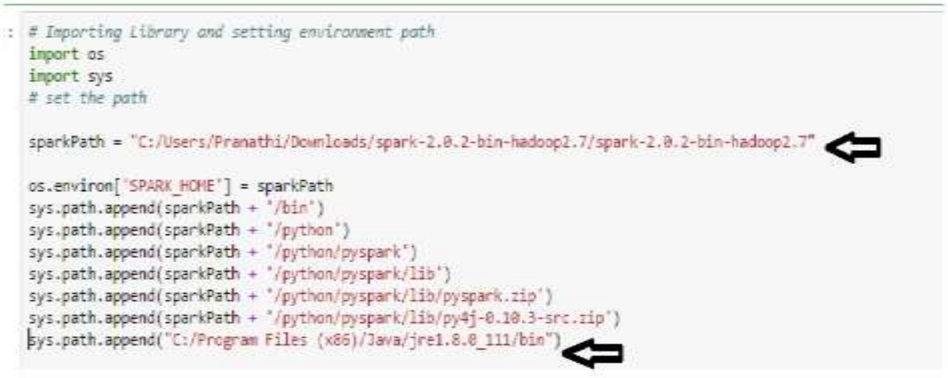
1. Download latest Spark from http://spark.apache.org/downloads.html



Download Apache Spark™

1. Choose a Spark release: 2.0.2 (Nov 14 2016) ▾
2. Choose a package type: Pre-built for Hadoop 2.7 and later ▾
3. Choose a download type: Direct Download ▾
4. Download Spark: spark-2.0.2-bin-hadoop2.7.tgz

2. Extract the tar file to a desired location
3. Open command prompt and navigate to the folder where it is extracted using command cd. For example navigate it to the below link C:\Users\KK\Desktop\MSBA\FALL_SEMESTER\big_data\spark_installation\spark-2.0.2-bin-hadoop2.7
4. Once you navigate to the above folder press "bin/pyspark"
   a) You might run into errors like jar file not found etc. if you don't have Java on your machine. Download and Install Java on your machine from https://java.com/en/download/. Restart the system and follow step no. 3 and 4
5. Install py4j using "*pip install py4j*"
6. Open Jupyter notebooks and copy the below the code



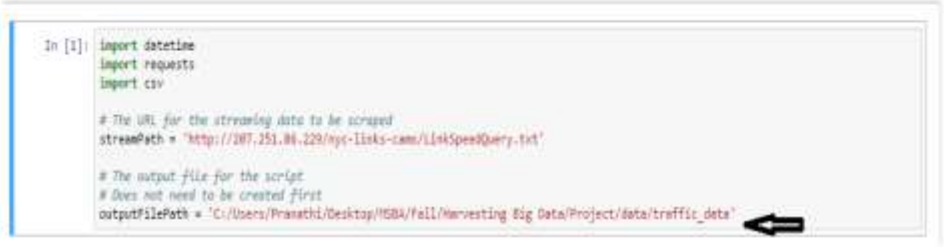
```
# Importing Library and setting environment path
import os
import sys
# set the path

sparkPath = "C:/Users/Pranathi/Downloads/spark-2.0.2-bin-hadoop2.7/spark-2.0.2-bin-hadoop2.7"

os.environ['SPARK_HOME'] = sparkPath
sys.path.append(sparkPath + '/bin')
sys.path.append(sparkPath + '/python')
sys.path.append(sparkPath + '/python/pyspark')
sys.path.append(sparkPath + '/python/pyspark/lib')
sys.path.append(sparkPath + '/python/pyspark/lib/pyspark.zip')
sys.path.append(sparkPath + '/python/pyspark/lib/py4j-0.10.3-src.zip')
sys.path.append("C:/Program Files (x86)/Java/jre1.8.0_111/bin")
```

Change sparkpath and java file path to the location they are located in your system in location marked with arrow

7. Run “import pyspark” to check if it is installed properly

Steps to recreate analysis:

1. Open 1_webscraping.ipynb. Change the outputFilePath in the first block to a location on your system where you want to store the files



```
In [1]: import datetime
        import requests
        import csv

        # The URL for the streaming data to be scraped
        streamPath = 'http://207.251.86.229/nyc-links-cams/LinkSpeedQuery.txt'

        # The output file for the script
        # Does not need to be created first
        outputFilePath = 'C:/Users/Pranathi/Desktop/MSBA/Fall/Harvesting Big Data/Project/data/traffic_data'
```

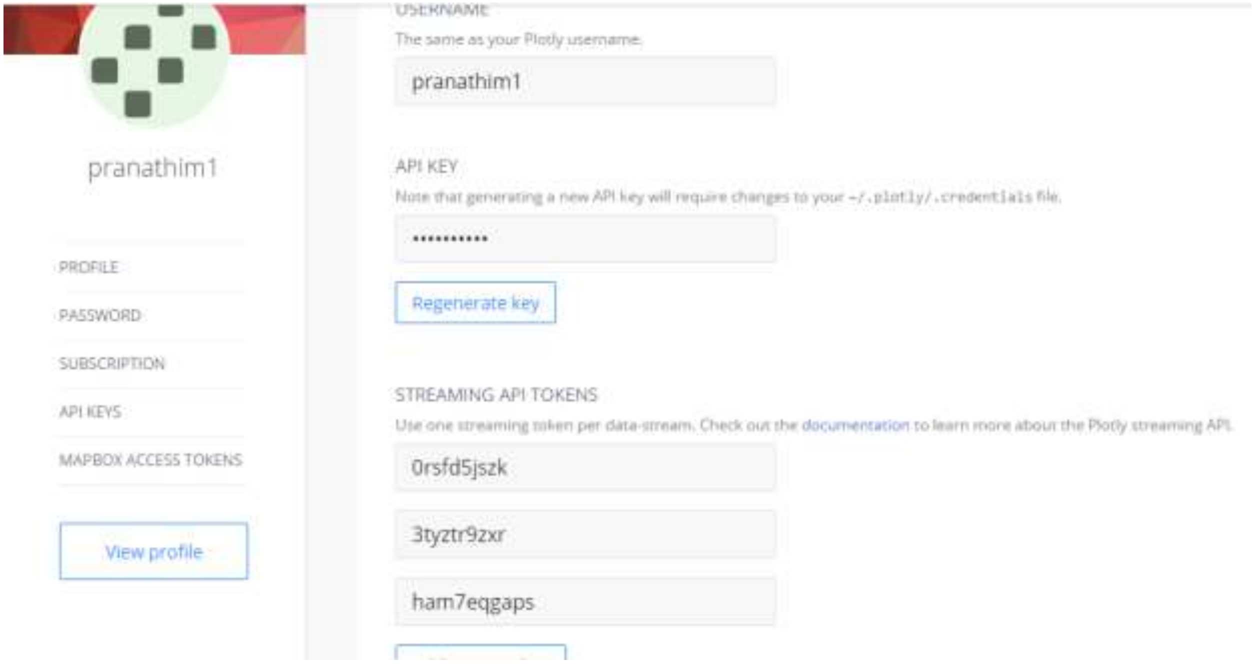
2. Open any of the 2a_Link speed Graph/ 2b_average speed graph – borough/ 2c_average time graph – borough. Change sparkPath to spark file location and streamFilePath to webscraping data in whichever code you want to run.
3. 2a_Link speed Graph generates Average speeds in the links. 2b_average speed graph generates average speed in boroughs of New York. 2c_average time graph generates average time taken to travel across any links in a borough.
4. Run the webscraper code and run the 1st 3 blocks of code in any of the 2a/2b/2c. Then run the last block of the code. Make sure the webscraper code is still running while the 4th block of code is on run.
5. You can see the streaming graph above the 4th block.

Note:

plotly.tools.set_credentials_file(username='pranathim1', api_key='VWHz644nRnSifZD1MIe6')

token_1 = '0rsfd5jszk'

The above credentials can be replaced with your credentials. Each graph has to be given a separate token. You can generate as many tokens as you want on plotly website. Go to https://plot.ly/settings/api and new tokens can be generated



USERNAME

The same as your Plotly username.

pranathim1

API KEY

Note that generating a new API key will require changes to your ~/.plotly/.credentials file.

•••••••••••

STREAMING API TOKENS

Use one streaming token per data-stream. Check out the documentation to learn more about the Plotly streaming API.

0rsfd5jszk

3tyztr9zxr

ham7eqgaps