Gokul Krishna

2/28/2017

```
1. import org.apache.spark.sql.functions._
2. import org.joda.time.format.DateTimeFormat
3. val inputPath = "/Users/gkrishnan/Downloads/pollution/*" val data1 =
   sqlContext.read
4. .format("com.databricks.spark.csv") .option("header", "true")
5. header .option("delimiter", ",") .option("inferSchema", "true")
6. .load(inputPath)
7. data1.toDF().registerTempTable("data1")
8. select ozone, timestamp from data1
9. data1.show()
10. import matplotlib
11. import matplotlib.pyplot as plt
12. import seaborn as sns
13. import StringIO
14. def show(p):
15. img = StringIO.StringIO()
16. p.savefig(img, format='svg')
17. img.seek(0)
18. print "%html " + img.buf
19. df = sqlContext.sql("SELECT ozone, latitude,longitude FROM data1 group
    by latitude, longitude"
20.         data = df.toPandas()
21. value = "ozone"
22.         x = "latitude, longitude"
23.         grouping = ["Month"]
24.         heatmap_data = data.pivot_table(values=value, index=x,
    columns=grouping)
25.         heatmap_data = heatmap_data[0:100]
26.         a4_dims = (len(heatmap_data.columns),50)
27.         fig, ax = plt.subplots(figsize=a4_dims)
```

```
28.       ax.set_title("Avg Arrival Delay")
29.       sns.heatmap(heatmap_data, ax=ax, annot=True, fmt=".
30.       show(plt)
31. data1.registerTempTable("data1")
32.       from pyspark.sql import SQLContext
33.       df = sqlContext.table("data1")
34.       df.head(10)
35.       df.count()
36.       import os
37.       import pandas as pd
38.       import glob
39.       listf =[]
40.       for row in glob.glob('/Users/gkrishnan/Downloads/pollution/*'):
    data = pd.read_csv(row)
41. listf.append(data)
42.       data.count()
43.       import plotly.plotly as py
44.        import plotly.graph_objs as go
45.       py.sign_in('gkrishnan', 'raCcZvtt0wUkyyr2YlplL')
46.       trace = go.Scatter( x = data['timestamp'], y = data['ozone'] )
47.       layout = dict(title = 'Time Series Plot Of Ozone', xaxis = dict(title =

    'Timestamp'),

48.       yaxis = dict(title = 'Ozone'), )

49.       plot = [trace]

50.       fig = dict(data=plot, layout=layout)

51. py.plot(fig, filename='Line Chart')

52.       for col in df.columns: df[col] = df[col].astype(str)

53.       scl = [[0.0, 'rgb(242,240,247)'],[0.2, 'rgb(218,218,235)'],[0.4,

    'rgb(188,189,220)'],

54.       \ [0.6, 'rgb(158,154,200)'],[0.8, 'rgb(117,107,177)'],[1.0,

    'rgb(84,39,143)']]

55.       df['text'] = df['ozone']

56.       mapplot = [ dict( type='choropleth', colorscale = scl,

    autocolorscale = False,

57.       locations = df['latitude'], z = df['ozone'].astype(float),
58.        locationmode = 'USA-states', text = df['text'],
59.        marker = dict( line = dict ( color = 'rgb(255,255,255)', width =
    2 ) ),
60.       colorbar = dict( title = "Millions USD") ) ]
61. layout = dict( title = '2011 US Agriculture Exports by State
62.       (Hover for breakdown)', geo = dict( scope='usa', projection=dict(
    type='albers usa' ),
```

```
63.       showlakes = True, lakecolor = 'rgb(255, 255, 255)'), )
64.       fig = dict( data=mapplot, layout=layout )
65.       py.iplot( fig, filename='d3-cloropleth-map' )
66.       his = [ go.Histogram( x=x ) ]
67.       layout = dict(title = 'Distribution Of Ozone', xaxis = dict(title =
   'Count'),
68.       yaxis = dict(title = 'Ozone'), )
69.       figh1 = dict(data=his, layout=layout)
70.       py.plot(figh1)
71. import plotly.plotly as py
72.        import plotly.graph_objs as go
73.       x = data['carbon_monoxide']
74.       his1 = [ go.Histogram( x=x ) ]
75.       layout = dict(title = 'Distribution Of Carbon Monoxide',
76.       xaxis = dict(title = 'Count'), yaxis = dict(title = 'Carbon Monoxide'),
   )
77.       figh2 = dict(data=his1, layout=layout)
78.       py.plot(figh2)
79.       import plotly.plotly as py
80.       import plotly.graph_objs as go
81. import numpy as np
82.       x = data['sulfure_dioxide']
83.       his2 = [ go.Histogram( x=x ) ]
84.       layout = dict(title = 'Distribution Of Sulfure Dioxide',
85.        xaxis = dict(title = 'Count'),
86.        yaxis = dict(title = 'Sulfure Dioxide'), )
87.       figh3 = dict(data=his2, layout=layout)
88.       x = data['nitrogen_dioxide']
89.       his4 = [ go.Histogram( x=x ) ]
90.       layout = dict(title = 'Distribution Of Nitrogen Dioxide', xaxis =
   dict(title = 'Count'),
91. yaxis = dict(title = 'Nitrogen Dioxide'), )
92.       figh5 = dict(data=his4, layout=layout)
93.       trace3 = go.Scatter( x = data['timestamp'], y =
   data['nitrogen_dioxide'] )
94.       plot3 = [trace3] layout = dict(title = 'Time Series Plot Of Nitrogen
   Dioxide',
```

```
95.       xaxis = dict(title = 'Timestamp'), yaxis =
      dict(title = 'Nitrogen Dioxide'), )
96.       fig4 = dict(data=plot3, layout=layout)
97.       from tweepy.streaming import StreamListener
98.       from tweepy import OAuthHandler
99.       from tweepy import Stream
100.      access_token = "824754411826188288-
   YIKCP2DQNf6IgGrrStyzr8RX5LGh5jG"
101.      access_token_secret = "
          bB5W9q7IfTFVZA64fQ0jEFfjMWhj7DchS8EPqocIE9V55"
102.      consumer_key = "SArrLUs3FtHIQWMIdzLgrJZQZR"
103.      consumer_secret =
   "kWKcIhkj04eWxBGveDIztIIMbay7ny6aY6i2890mA5FVGTMVdy"
```