Predicting Felony Rates in a Vicinity and triggering Alerts using Spark

LAB-3

Data Intensive Computing

AMLAN GUPTA (50288686)

ATRAYEE NAG (50288651)

Abstract

The current scenario is witnessing a rise in felony rates in New York city, to combat which, data analysis has played a vital role and we will use Apache Spark for the prediction and analysis. We are developing an interactive application to look up localities in New York city which will use the Spark MLlib library to create a model that will predict which crime is more likely to occur in a vicinity and layer the high-risk zones on top of a clearly-defined map using the statistical dataset provided by NYPD Complaint Data Historic. Users can further register for alerts which will be triggered by the Spark streaming component from live Twitter data.

Problem Statement

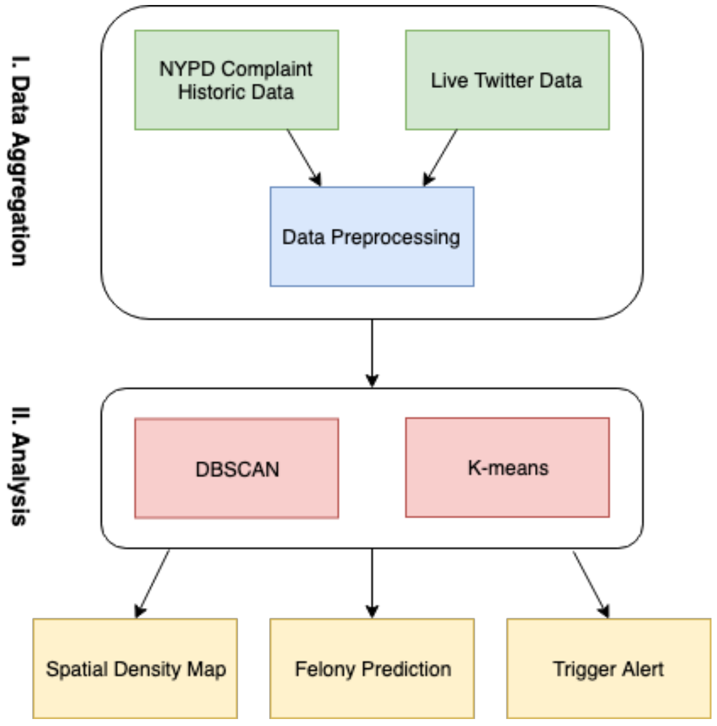
Crime is a socio-economic problem affecting life quality and economic growth. The formalized processes of analyzing crime and establishing specialized units to perform these tasks are relatively new to law enforcement and requires highly skilled individuals for the job. In our attempt to automate the process and give consumers a direct insight to their choice of neighborhood we are designing an application, where they can:

1) View the crime hotspots surrounding their vicinity
2) Get a prediction of felonies most likely to occur
3) Sign up for live alerts for any occurrence of felony offences in their area

We will be solving the above problems using Spark as it has the ability to process both batch data and live streaming data faster compared to other big data solutions. Using popular machine learning algorithms like DBSCAN, K-means we will try to sketch an image for users which will not only increase their quality of life but has the potential to prevent the repetition of these unfortunate events.

Solution Model and Design Document

Architectural block diagram



Data Pipeline

- NYPD Complaint Data Historic

  This dataset includes all valid felony, misdemeanor, and violations reported to the New York City Police Department (NYPD) from 2006 to the end of last year (2017). The dataset contains:

  Columns: 35

  Rows: 6,036,805

- Twitter Live Stream

  We collect live data from twitter using the following keyword list to extract tweets related to crime using the twitter API: larceny, assault, criminal, intoxicated, drugs, weapons, robbery, theft, murder

  Latitude/Longitude of center point of New York city - 40.730610 / -73.935242

  Radius of area from center – 10 miles (16 km)

Solution Architecture

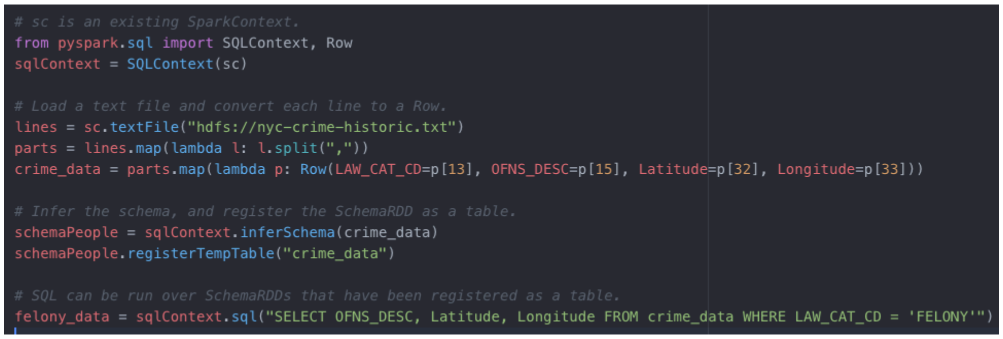
- Spatial Density Map and Prediction

  1. Pre-processing

     Out of all 35 columns we will be using below columns:

LAW_CAT_CD	Level of offense: felony, misdemeanor, violation
OFNS_DESC	Description of offense corresponding with key code
Latitude	Midblock Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)
Longitude	Midblock Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)

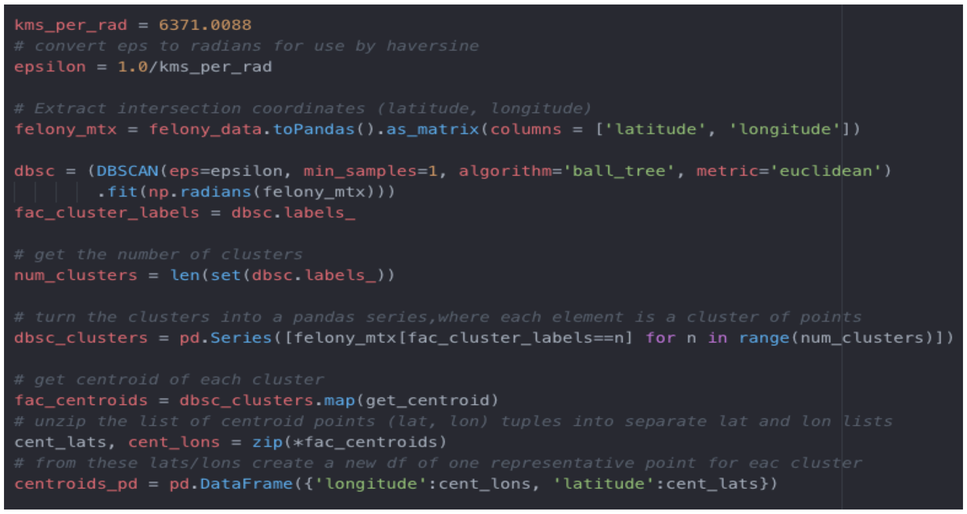
     We will filter the dataset by felony and the resultant row count is: 1,855,915.



```
# sc is an existing SparkContext.
from pyspark.sql import SQLContext, Row
sqlContext = SQLContext(sc)

# Load a text file and convert each line to a Row.
lines = sc.textFile("hdfs://nyc-crime-historic.txt")
parts = lines.map(lambda l: l.split(","))
crime_data = parts.map(lambda p: Row(LAW_CAT_CD=p[13], OFNS_DESC=p[15], Latitude=p[32], Longitude=p[33]))

# Infer the schema, and register the SchemaRDD as a table.
schemaPeople = sqlContext.inferSchema(crime_data)
schemaPeople.registerTempTable("crime_data")

# SQL can be run over SchemaRDDs that have been registered as a table.
felony_data = sqlContext.sql("SELECT OFNS_DESC, Latitude, Longitude FROM crime_data WHERE LAW_CAT_CD = 'FELONY'")
```

  2. Find Hotspots using DBSCAN

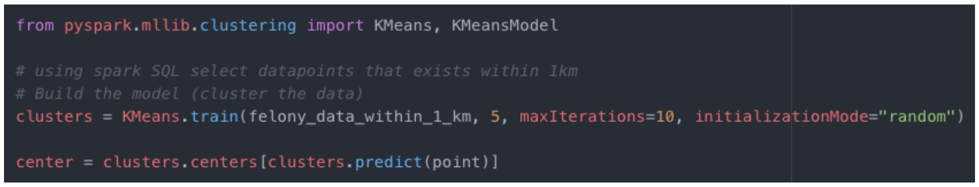
     Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). It has been used to seek areas in the dataset that has a high density of occurrence. 1km is the value of epsilon which means this is the maximum distance between 2 samples for them to be considered in the same neighborhood.



```
kms_per_rad = 6371.0088
# convert eps to radians for use by haversine
epsilon = 1.0/kms_per_rad

# Extract intersection coordinates (latitude, longitude)
felony_mtx = felony_data.toPandas().as_matrix(columns = ['latitude', 'longitude'])

dbsc = (DBSCAN(eps=epsilon, min_samples=1, algorithm='ball_tree', metric='euclidean')
        .fit(np.radians(felony_mtx)))
fac_cluster_labels = dbsc.labels_

# get the number of clusters
num_clusters = len(set(dbsc.labels_))

# turn the clusters into a pandas series,where each element is a cluster of points
dbsc_clusters = pd.Series([felony_mtx[fac_cluster_labels==n] for n in range(num_clusters)])

# get centroid of each cluster
fac_centroids = dbsc_clusters.map(get_centroid)
# unzip the list of centroid points (lat, lon) tuples into separate lat and lon lists
cent_lats, cent_lons = zip(*fac_centroids)
# from these lats/lons create a new df of one representative point for eac cluster
centroids_pd = pd.DataFrame({'longitude':cent_lons, 'latitude':cent_lats})
```

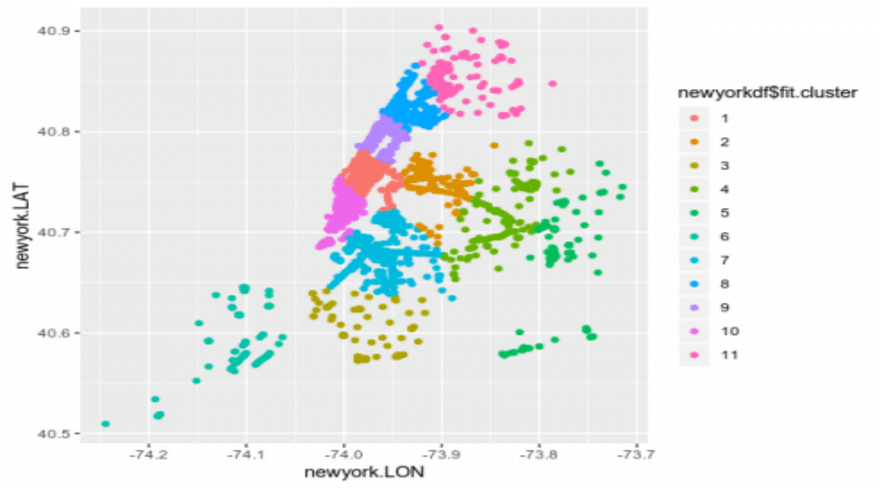
3. Creating clusters using K-means and predict

User can make a query in the application to check which felony is more likely to occur in the neighborhood. Our application takes the latitude and longitude of that neighborhood and uses k-means clustering on the historic crime data within 1km to check in which cluster the neighborhood location belongs to.

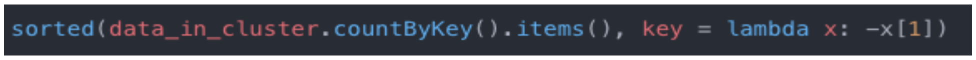


```
from pyspark.mllib.clustering import KMeans, KMeansModel

# using spark SQL select datapoints that exists within 1km
# Build the model (cluster the data)
clusters = KMeans.train(felony_data_within_1_km, 5, maxIterations=10, initializationMode="random")

center = clusters.centers[clusters.predict(point)]
```

If we plot the clusters using Matplotlib it may look like this:



Now if the location belongs to cluster 1, we will use all the data points that belong to that cluster and sort them by their occurrence. Logically, the type of crime which has most number of occurrence in that vicinity is most likely to happen again.



```
sorted(data_in_cluster.countByKey().items(), key = lambda x: -x[1])
```

- ➢ Trigger Alerts

  The mechanism to trigger alert to the subscribers is powered by live twitter data.

  Using Spark streaming we will collect tweets that are originated from New York City and save them in HBase Table. In a certain interval, the data in HBase table will be evaluated and if in a radius of 5km,10 tweets are found, it will trigger a notification to the subscribers located at that area.

Steps to Trigger Alert using Spark:

1. Get live data from Twitter API using Spark streaming.
2. Processes the streaming data and store the location data of the tweet in the HBase Table.
3. A job will run at an interval which will read the data from Hbase and using DBASCAN, will try to form a cluster of locations where a high number of tweets originated, removing the outliers.
4. If a cluster is formed, the system will check if any user has subscribed for alerts from that location and will trigger alerts.

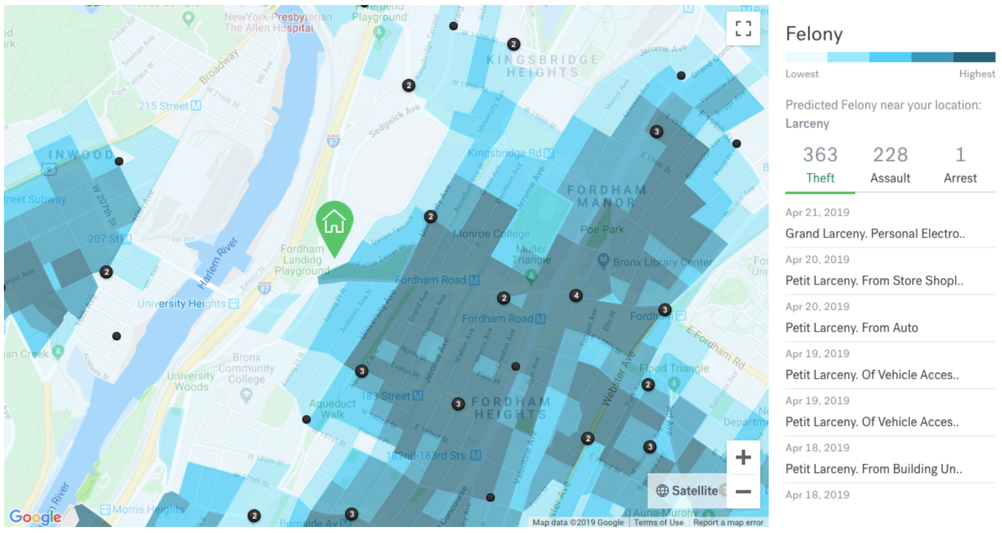
Relevant components

- Spark Streaming
- Spark SQL
- MLlib
- Scikit-learn
- HBase
- D3.js

Language: Python

Outcomes and Visualization

1. User can see a heat map of felonies occurred in their vicinity.
2. Our application has predicted which felony is more likely to occur.
3. Local statistics of frequent felony in recent times is listed.



Felony

Lowest Highest

Predicted Felony near your location: Larceny

363	228	1
Theft	Assault	Arrest

Apr 21, 2019
Grand Larceny. Personal Electro..

Apr 20, 2019
Petit Larceny. From Store Shopl..

Apr 20, 2019
Petit Larceny. From Auto

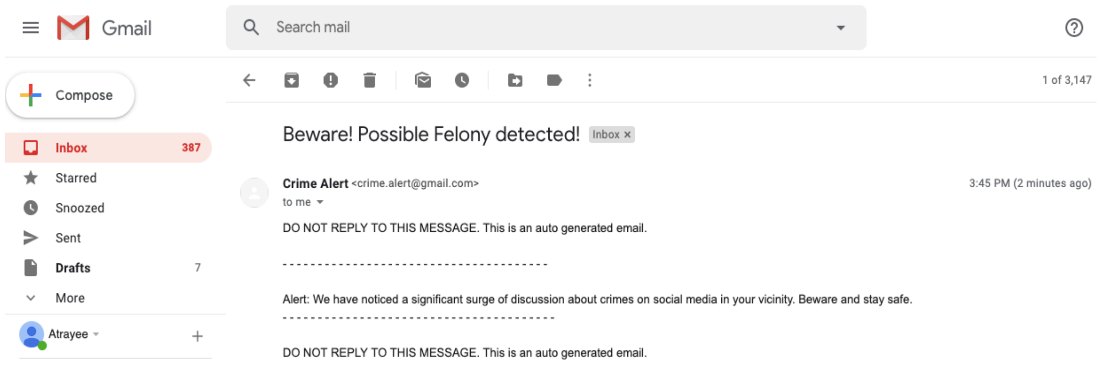
Apr 19, 2019
Petit Larceny. Of Vehicle Acces..

Apr 19, 2019
Petit Larceny. Of Vehicle Acces..

Apr 18, 2019
Petit Larceny. From Building Un..

Apr 18, 2019

4. An alert mail is sent to the subscribers if frequent social media activity has been noticed in the neighborhood.



Beware! Possible Felony detected!

Crime Alert <crime.alert@gmail.com>
to me
3:45 PM (2 minutes ago)

DO NOT REPLY TO THIS MESSAGE. This is an auto generated email.

Alert: We have noticed a significant surge of discussion about crimes on social media in your vicinity. Beware and stay safe.

DO NOT REPLY TO THIS MESSAGE. This is an auto generated email.

Summary

1. The demand for stream processing is increasing every day as processing big volumes of data is not enough. Spark Streaming provides a high-level abstraction called discretized stream or DStream, which represents a continuous stream of data.
2. Spark provides a general machine learning library – Mllib where with the scalability, language compatibility, and speed of Spark, data scientists can solve and iterate through their data problems faster.
3. Spark SQL supports relational processing both within Spark programs (on native RDDs) and on external data sources using a programmer friendly API.
4. We need to generate and predict our felony prediction model on the fly, for which Spark is the perfect fit as it is 100 times faster than Hadoop.

References

1. https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i
2. https://spark.apache.org/docs/1.2.2/sql-programming-guide.html
3. https://spark.apache.org/examples.html
4. https://stackoverflow.com/questions/36145277/predict-clusters-from-data-using-spark-mllib-kmeans