Chicago Crime Rate Analysis

Elaheh Aghaarabi
Towson University
eaghaa1@students.towson.edu

Abstract— **High crime rate in cities has been an issue in United States for a long time. Data analysis and visualization can be used as a powerful technique to help policy makers and law enforcement organizations to be able to handle this issue more effectively. Finding meaningful patterns and behaviors in Chicago crime dataset is the goal of this research. The most vulnerable neighborhoods and most effective features on the number of will be investigated to help police or related organizations to adjust their reactions. Moreover, a machine learning classifier with logistic regression will be used to predict the crime rate in this city.**

Keywords—Crime rate prediction, Spark, Python, Machine Learning, Big data, Chicago crime

I. Introduction

The Chicago crime rate analysis has been investigated in many researches around the world and United States to give a better picture of the repetitive patterns and incidents in this city. Law enforcement offices and Police departments will benefit from this information to make more effective policies and decisions. Several hypothesis and features have been investigated by researchers to reveal patterns and behaviors in Chicago dataset. For instance, Bernasco et al. conducted a block-level analysis based on census data to find the effect of crime generators and crime attractors on the frequency of street robbery at adjacent census blocks [1]. In another study, a regression analysis has been done by Block to find the effects of community and environment on crime rate [2]. This study finds that income level and also proximity of high crime rate neighborhoods are the two main factors affecting the crime rate [2]. Another research group employed big data and data mining techniques to show trends in Chicago, Philadelphia and San Francisco [3]. The authors developed a predictive model using LSTM to estimate the crime rate on unseen data.

For this study, I am using the Chicago crime dataset from 2012 to 2017 to find patterns and trends in occurrence of crime in this city. The dataset comes from Kaggle website [4]. Spark will be used to perform the data analysis because the dataset is big in size and Spark will be a powerful tool to handle the big data. Jupyter Notebook and Python will be used to run Pyspark for this project. Spark Python API (Sphinx) is a Spark API in Python interface which mainly uses pyspark package [5].

A. Research Questions and Hypothesis

In this project, I will investigate the following hypothesis to find possible patterns and trends in number of crimes recorded in Chicago crime dataset from 2012 to 2017 to give a better insight to police department to maximize their performance and minimize their expenses. I will investigate the effect of several features on the number of incidents happen in this city using four hypothesis.

1. Top types of crimes in this city will be rubbery, narcotics and burglary. As a result, number of crimes recorded is affected by the type of crime.
2. The number of arrests has increased in Chicago from 2012 to 2017. As a result, time has an effect in number of crimes recorded and police performance can be investigated.
3. Streets, sidewalks and parking lots are the locations that most of the crime will happen. As a result, number of crimes recorded is affected by the location type.
4. Adjacent high community areas are more vulnerable. In other words, lower proximity with a high crime rate community increases the chance of having more crime in a particular community.

B. Feasibility and Availability of Approach

The Chicago crime dataset is a big dataset. Spark is one of the most effective tools to handle big data. I preferred to perform both data analysis and machine leaning model in the same environment. After doing some research, I found pyspark which gives the flexibility of using Spark, machine learning libraries and statistics in the same environment. I installed pyspark on a virtual environment using pipenv. There is no need to use pip and virtualenv separately anymore and pipenv can be used in terminal to install this package. Pyspark will be able to handle big data (using Spark) and also benefits from strong machine learning libraries in Python, which will be the best candidate for this project.

II. Description of the Dataset

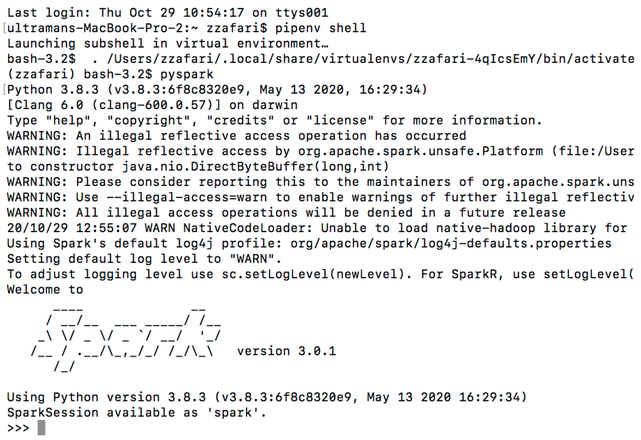
The Chicago crime dataset is provided by Kaggle website [4]. In this analysis, I use the 2012 to 2017 crime data. This dataset needs preprocessing and data cleaning before starting the data analysis. For instance, I renamed the headers to make them more meaningful. Then, a dictionary was used to rename features. This dataset contains several features which some of them will be used for this analysis. To give a better picture of the dataset, I will describe the features I will use;

- ID is a unique identifier for the record.
- Case number is a unique identifier, a string variable representing a number for each case of arrest assigned by the police department.
- Date is a string variable showing the date that crime happened.
- Block represents the location of the occurrence of crime based on census data. The address is shown as block level to protect the privacy of victims.

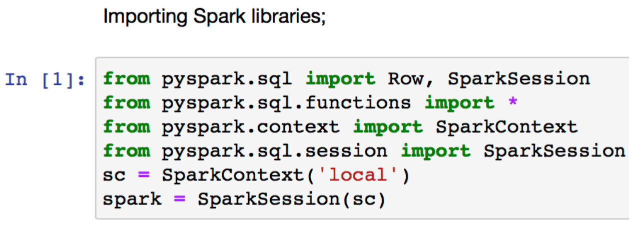
- Primary type is a string variable which is a description of the crime type.
- Location description is a string variable that represents the type of the location that crime/incident happens. For instance, street, residential and parking lot are some of the categories for this feature.
- Arrest is a boolean feature showing if the guilty person was arrested or not.
- District is a double value representing the police district are that crime occurred.
- Community area is a double number representing the area that crime happened based on the community are. Chicago is divided to 77 community areas.
- Year is an integer variable showing the year that incident happened.
- Count is an integer variable to sum all of the records of incidents in dataset.

III. Data Analysis using Spark API

The Spark API is used in Jupyter Notebook to perform data analysis for this project. Pyspark is the main library package for this purpose. Below I would add screenshots of my pyspark installation, importing related libraries to Jupyter Notebook environment, reading the dataset using Spark and the size of dataset I am using for this project.

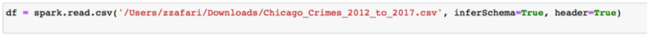


```
Last login: Thu Oct 29 10:54:17 on ttys001
[ultramans-MacBook-Pro-2:~ zzafari$ pipenv shell
Launching subshell in virtual environment…
bash-3.2$  . /Users/zzafari/.local/share/virtualenvs/zzafari-4qIcsEmY/bin/activate
(zzafari) bash-3.2$ pyspark
[Python 3.8.3 (v3.8.3:6f8c8320e9, May 13 2020, 16:29:34)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/User
to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.uns
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflectiv
WARNING: All illegal access operations will be denied in a future release
20/10/29 12:55:07 WARN NativeCodeLoader: Unable to load native-hadoop library for
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.0.1
      /_/

Using Python version 3.8.3 (v3.8.3:6f8c8320e9, May 13 2020 16:29:34)
SparkSession available as 'spark'.
>>> 
```

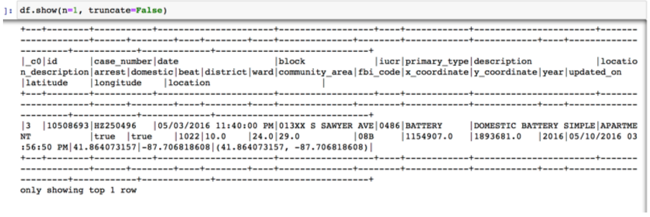


Importing Spark libraries;

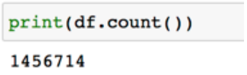
```
In [1]: from pyspark.sql import Row, SparkSession
        from pyspark.sql.functions import *
        from pyspark.context import SparkContext
        from pyspark.sql.session import SparkSession
        sc = SparkContext('local')
        spark = SparkSession(sc)
```



```
df = spark.read.csv('/Users/zzafari/Downloads/Chicago_Crimes_2012_to_2017.csv', inferSchema=True, header=True)
```



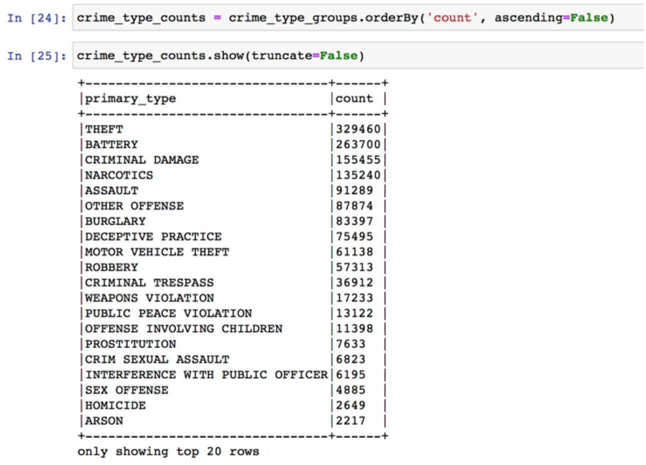
```
]: df.show(n=1, truncate=False)

+---+--------+----------+----------------------+-------------------+----+------------+----------------------+-------
-------------+------+--------+----+--------+----+-------------+--------+------------+------------+----+------------
---------+-----------+------------+-----------------------------+
|_c0|id      |case_number|date                  |block              |iucr|primary_type|description           |locatio
n_description|arrest|domestic|beat|district|ward|community_area|fbi_code|x_coordinate|y_coordinate|year|updated_on
|latitude    |longitude   |location                     |
+---+--------+----------+----------------------+-------------------+----+------------+----------------------+-------
-------------+------+--------+----+--------+----+-------------+--------+------------+------------+----+------------
---------+-----------+------------+-----------------------------+
|3  |10508693|HZ250496  |05/03/2016 11:40:00 PM|013XX S SAWYER AVE|0486|BATTERY     |DOMESTIC BATTERY SIMPLE|APARTME
NT           |true  |true    |1022|10.0    |24.0|29.0          |08B     |1154907.0   |1893681.0   |2016|05/10/2016 03
:56:50 PM|41.864073157|-87.706818608|(41.864073157, -87.706818608)|
+---+--------+----------+----------------------+-------------------+----+------------+----------------------+-------
-------------+------+--------+----+--------+----+-------------+--------+------------+------------+----+------------
---------+-----------+------------+-----------------------------+
only showing top 1 row
```



```
print(df.count())

1456714
```

A. *Hypothesis 1: The effect of type of crimes recorded on the number of crimes recorded*

The dataset was imported using spark. The top type of crimes that happened most are ranked in Figure 1.



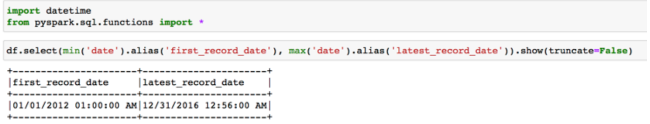
```
In [24]: crime_type_counts = crime_type_groups.orderBy('count', ascending=False)

In [25]: crime_type_counts.show(truncate=False)

         +-----------------------------------+------+
         |primary_type                       |count |
         +-----------------------------------+------+
         |THEFT                              |329460|
         |BATTERY                            |263700|
         |CRIMINAL DAMAGE                    |155455|
         |NARCOTICS                          |135240|
         |ASSAULT                            |91289 |
         |OTHER OFFENSE                      |87874 |
         |BURGLARY                           |83397 |
         |DECEPTIVE PRACTICE                 |75495 |
         |MOTOR VEHICLE THEFT                |61138 |
         |ROBBERY                            |57313 |
         |CRIMINAL TRESPASS                  |36912 |
         |WEAPONS VIOLATION                  |17233 |
         |PUBLIC PEACE VIOLATION             |13122 |
         |OFFENSE INVOLVING CHILDREN         |11398 |
         |PROSTITUTION                       |7633  |
         |CRIM SEXUAL ASSAULT                |6823  |
         |INTERFERENCE WITH PUBLIC OFFICER   |6195  |
         |SEX OFFENSE                        |4885  |
         |HOMICIDE                           |2649  |
         |ARSON                              |2217  |
         +-----------------------------------+------+
         only showing top 20 rows
```

Figure 1. Type of crime

Based on Figure 1, Theft, Battery and criminal damage are the incidents occurring most. It gives the insight to police departments how to invest and train their employees to be able to stop the most occurring crimes. Moreover, hiring more employees for departments responsible for these crimes would be beneficial. As a result, the type of crime has a strong effect on number of crimes.

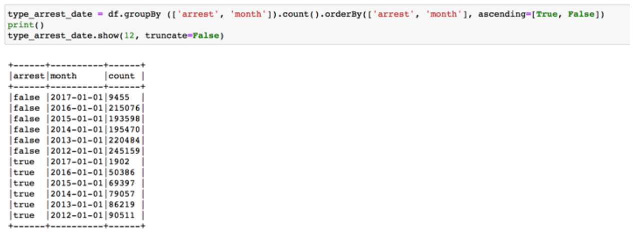
B. *Hypothesis 2: the effect of time on number of arrests*

As I mentioned before, this dataset has recordings of arrests from 2012 to 2017. Figure 2 confirms this claim. Figure 3 shows how many of these recorded incidents ended to an arrest.



```
import datetime
from pyspark.sql.functions import *

df.select(min('date').alias('first_record_date'), max('date').alias('latest_record_date')).show(truncate=False)

+----------------------+----------------------+
|first_record_date     |latest_record_date    |
+----------------------+----------------------+
|01/01/2012 01:00:00 AM|12/31/2016 12:56:00 AM|
+----------------------+----------------------+
```

Figure 2. Dataset time stream



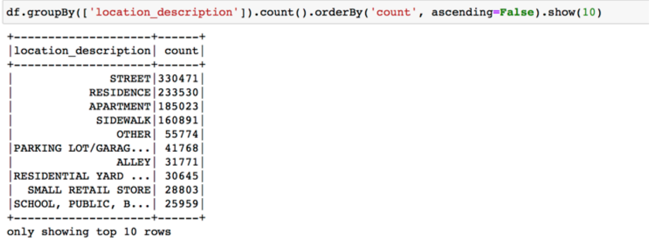
```
type_arrest_date = df.groupBy (['arrest', 'month']).count().orderBy(['arrest', 'month'], ascending=[True, False])
print()
type_arrest_date.show(12, truncate=False)

+------+----------+------+
|arrest|month     |count |
+------+----------+------+
|false |2017-01-01|9455  |
|false |2016-01-01|215076|
|false |2015-01-01|193598|
|false |2014-01-01|195470|
|false |2013-01-01|220484|
|false |2012-01-01|245159|
|true  |2017-01-01|1902  |
|true  |2016-01-01|50386 |
|true  |2015-01-01|69397 |
|true  |2014-01-01|79057 |
|true  |2013-01-01|86219 |
|true  |2012-01-01|90511 |
+------+----------+------+
```

Figure 3. Number of arrests during time

In 2012, almost 26% of incidents resulted into an arrest. In 2013, almost 28% of records had a true arrest. In 2014, 2015, 2016 and 2017, 29%, 26%, 19% and 18% of arrests happened respectively. This means that the hypothesis is not true and percentage of arrests for crimes decreased over time. However, there is not a certain relation between the year and number of incidents recorded.

C. *Hypothesis 3: The effect of location type on number of crimes recoreded*

Figure 4 shows the locations with top numbers of incidents happened.



```
df.groupBy(['location_description']).count().orderBy('count', ascending=False).show(10)
```

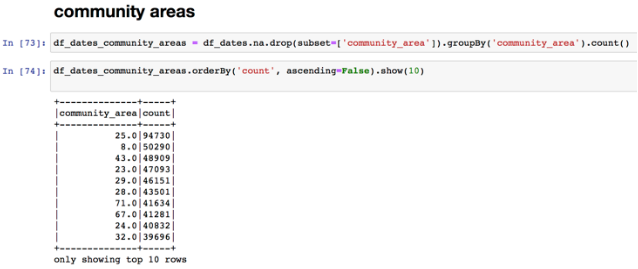
```
+--------------------+------+
|location_description| count|
+--------------------+------+
|              STREET|330471|
|           RESIDENCE|233530|
|           APARTMENT|185023|
|            SIDEWALK|160891|
|               OTHER| 55774|
|PARKING LOT/GARAG...| 41768|
|               ALLEY| 31771|
|RESIDENTIAL YARD ...| 30645|
|   SMALL RETAIL STORE| 28803|
|SCHOOL, PUBLIC, B...| 25959|
+--------------------+------+
only showing top 10 rows
```

Figure 4. Crime location

Streets, residence, apartments and sidewalks are top four locations that crimes happen. It reveals the locations that police should deploy more officers. Moreover, there is a meaningful relation between the location type and number of incidents recorded.

D. *Hypothesis 4: The effect of community areas on number of crimes recorded*

Figure 5 depicts the community areas with the highest number of crimes happening from 2012 to 2017. Chicago is divided to 77 areas and each number represents an area. An interactive map shows this division and corresponding area numbers [6].



community areas

```
In [73]: df_dates_community_areas = df_dates.na.drop(subset=['community_area']).groupBy('community_area').count()
```

```
In [74]: df_dates_community_areas.orderBy('count', ascending=False).show(10)
```

```
+--------------+-----+
|community_area|count|
+--------------+-----+
|          25.0|94730|
|           8.0|50290|
|          43.0|48909|
|          23.0|47093|
|          29.0|46151|
|          28.0|43501|
|          71.0|41634|
|          67.0|41281|
|          24.0|40832|
|          32.0|39696|
+--------------+-----+
only showing top 10 rows
```

Figure 5. Community areas affected by higher crime rates

Top ten areas with highest number of crimes are listed in Figure 5. Using [6], I will check to see if these areas happen to be adjacent. Areas 25, 8, 23, 29, 28, 24 are called Austin, Montclare, Humboldt Park, North Lawndale, Near West Side and West Town respectively, which are neighbors in Figure 6. Areas numbered 71 and 67 are called Auburn Gresham and West Englewood, which share borders according to Figure 6.

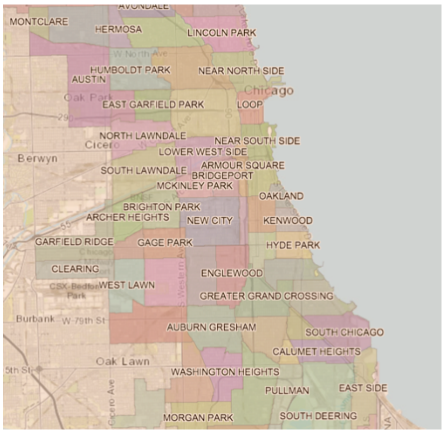
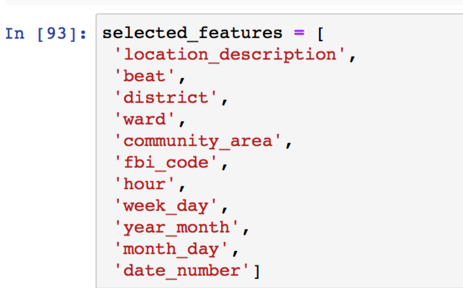


Figure 6. Chicago community areas

According to these findings hypothesis 4 is true. The proximity of community areas has a strong effect on number of incidents recorded. Another interesting point in this analysis is that south side of Chicago is notoriously known as a higher crime rate region. The income level is lower, and it is expected that crime rate will be high. However, there are other parts of city (west side) that had higher crime rates at least from 2012 to 2017.

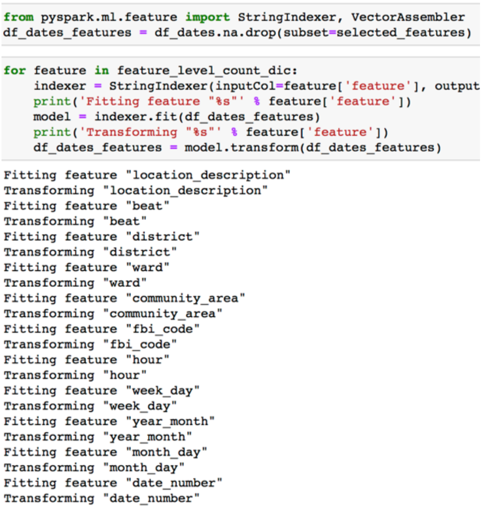
IV. MACHINE LEARNING MODEL (LOGISTIC REGRESSION CLASIFIER)

I will train a classifier by a logistic regression model using selected features. I am trying to predict the primary type of incidents in Chicago. In other words, the machine learning model predicts what type of crime will happen in future given certain features. I used following features for my model.



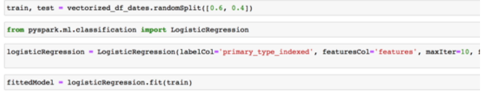
```
In [93]: selected_features = [
          'location_description',
          'beat',
          'district',
          'ward',
          'community_area',
          'fbi_code',
          'hour',
          'week_day',
          'year_month',
          'month_day',
          'date_number']
```

Then, I used Spark string indexer to index these features.



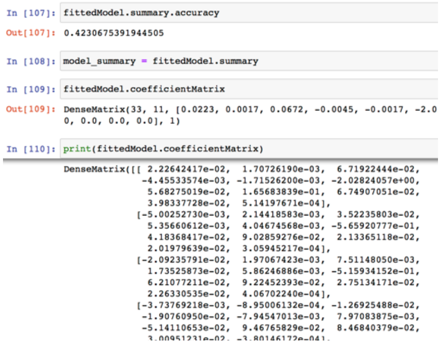
```
from pyspark.ml.feature import StringIndexer, VectorAssembler
df_dates_features = df_dates.na.drop(subset=selected_features)

for feature in feature_level_count_dic:
    indexer = StringIndexer(inputCol=feature['feature'], output
    print('Fitting feature "%s"' % feature['feature'])
    model = indexer.fit(df_dates_features)
    print('Transforming "%s"' % feature['feature'])
    df_dates_features = model.transform(df_dates_features)

Fitting feature "location_description"
Transforming "location_description"
Fitting feature "beat"
Transforming "beat"
Fitting feature "district"
Transforming "district"
Fitting feature "ward"
Transforming "ward"
Fitting feature "community_area"
Transforming "community_area"
Fitting feature "fbi_code"
Transforming "fbi_code"
Fitting feature "hour"
Transforming "hour"
Fitting feature "week_day"
Transforming "week_day"
Fitting feature "year_month"
Transforming "year_month"
Fitting feature "month_day"
Transforming "month_day"
Fitting feature "date_number"
Transforming "date_number"
```

I vectorized the features and labels and started training the model using 60% of data for training and 40% for test.



```
train, test = vectorized_df_dates.randomSplit([0.6, 0.4])

from pyspark.ml.classification import LogisticRegression

logisticRegression = LogisticRegression(labelCol='primary_type_indexed', featuresCol='features', maxIter=10, f

fittedModel = logisticRegression.fit(train)
```

Now let's check the training model performance.



```
In [107]: fittedModel.summary.accuracy
Out[107]: 0.4230675391944505

In [108]: model_summary = fittedModel.summary

In [109]: fittedModel.coefficientMatrix
Out[109]: DenseMatrix(33, 11, [0.0223, 0.0017, 0.0672, -0.0045, -0.0017, -2.0
          0, 0.0, 0.0, 0.0], 1)

In [110]: print(fittedModel.coefficientMatrix)
          DenseMatrix([[ 2.22642417e-02,  1.70726190e-03,  6.71922444e-02,
                        -4.45533574e-03, -1.71526200e-03, -2.02824057e+00,
                         5.68275019e-02,  1.65683839e-01,  6.74907051e-02,
                         3.98337728e-02,  5.14197671e-04],
                       [-5.00252730e-03,  2.14418583e-03,  3.52235803e-02,
                         5.35660612e-03,  4.04674568e-03, -5.65920777e-01,
                         4.18368417e-02,  9.02859276e-02,  2.13365118e-02,
                         2.01979639e-02,  3.05945217e-04],
                       [-2.09235791e-02,  1.97067423e-03,  7.51148050e-03,
                         1.73525873e-02,  5.86246886e-03, -5.15934152e-01,
                         6.21077211e-02,  9.22452393e-02,  2.75134171e-02,
                         2.26330535e-02,  4.06702240e-04],
                       [-3.73769218e-03, -8.95006132e-04, -1.26925488e-02,
                        -1.90760950e-02, -7.94547013e-03,  7.97083875e-03,
                        -5.14110653e-02,  9.46765829e-02,  8.46840379e-02,
                         3.00951231e-02, -3.80146172e-04],
```

The model performance can be improved by assigning more training data to training model. After improving the performance to 65% by using 80% for training and 20% for test data, the model predicts the occurrence of top crime types as following: assaults, battery, criminal damage, theft, burglary, narcotics.

V. Discussion and Conclusion

After investigating the four hypothesis and looking for patterns and trends in Chicago dataset, following results are revealed. Hypothesis 1 was not true and top three crime types are theft, battery and criminal damage. However, there was a relation between crime types and number of incidents recorded. The logistic regression model validates this finding too. Hypothesis 2 is not true and analysis show that arrests percentage have decreased over time and there is no meaningful relation between number of arrests and time (yearly wise). Top three locations that crime happen include streets, residence, apartments, which is partially compatible with hypothesis 3. It confirms there is a relation between the location type and number of incidents recorded.

Hypothesis 4 assumes that crime happens in adjacent neighborhoods which is confirmed by data analysis results. The community areas that share borders have highest crime rates. Moreover, there is a strong relationship between the community area number and number of incidents recorded.

For future analysis, a data analysis on relation between the number of crimes and time of day and month is suggested. It helps the law enforcement organizations to be able to plan accordingly. Moreover, a more comprehensive and updated dataset can be used. For instance, datasets from 2001 to 2019 can be merged and used which produces more reliable results given that Spark is a powerful tool to handle big data.

References

[1] Bernasco, W. and Block, R. (2011) 'Robberies in Chicago: A Block-Level Analysis of the Influence of Crime Generators, Crime Attractors, and Offender Anchor Points', *Journal of Research in Crime and Delinquency*, 48(1), pp. 33–57. doi: 10.1177/0022427810384135.

[2] BLOCK, R. (1979), COMMUNITY, ENVIRONMENT, AND VIOLENT CRIME. Criminology, 17: 46-57. doi:10.1111/j.1745-9125.1979.tb01275.x

[3] M. Feng et al., "Big Data Analytics and Mining for Effective Visualization and Trends Forecasting of Crime Data," in IEEE Access, vol. 7, pp. 106111-106123, 2019, doi: 10.1109/ACCESS.2019.2930410. M. R. Islam and M. F. Zibran, "Leveraging Automated Sentiment Analysis in Software Engineering," 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos Aires, 2017, pp. 203-214, doi: 10.1109/MSR.2017.9.

[4] https://www.kaggle.com/chicago/chicago-crime

[5] https://spark.apache.org/docs/latest/api/python/index.html

[6] https://chicago.maps.arcgis.com/apps/webappviewer/index.html?id=d56603be39824be099557dcdf9d7f7b9