Big data Analytics - End Project

Big Data Analysis using Hadoop

Team:
S20180020225 - Nanubala Gnana Sai
S20180010040 - Cherukuri Nikhilesh
S20180010188 - Vasimalla Amrutha

I. Introduction

Hadoop

Hadoop is a big data framework which excels in paralleling processing of terabytes of data. The framework allows processing structured or unstructured data. The framework utilizes the concept of nodes and clusters in a master-slave type architecture. It's salient features are:

- **Hadoop Distributed File System (HDFS)**: A fault tolerant and scalable file system stored in data clusters.
- **Yet Another Resource Negotiator (YARN)**: A resource management system which handles scheduling, resource allocation-deallocation etc. The Map-reduce algorithm is built atop of it.

PIG

For grouping the data we have used PIG Framework.

PIG Latin is a high level scripting language that is used with Apache Hadoop. Apache Pig is an abstraction over MapReduce. It is a tool which is used to analyze large sets of data used with Hadoop. All kinds of data manipulation operations can be performed in Hadoop using Pig.

II. Dataset Analysis:

Description - The formal description of the problem is to work with a dataset consisting of size in the range 100 - 500 MB using Hadoop.

DataSet - The dataset *'Crimes in chicago*' (around 320 MB) reflects incidents of crimes occured in the city Chicago from the year 2012 to 2017, consists of 23 columns and 1,00,000+ rows/

a) Exploratory Data Analysis (EDA):

- **Initial**: Int
- **ID**: ID of the crime - int
- **Case Number:** Number assigned to the case - chararray
- **Date:** Date of the incident occured.- DateTime
- **Block:** Name of the block - chararray
- **IUCR:** Code used to classify criminal incidents - chararray
- **Primary Type:** Type of the crime - chararray
- **Description:** Description of the crime - chararray
- **Location Description:** Description of the location where the incident occurred.
- **Arrest:** If the culprit is arrested or not - Boolean - True or False
- **Domestic:** Boolean
- **Beat:** int
- **District:**District number - float
- **Ward:** Ward number - float
- **Community Area:** Area number - float
- **FBI Code:** FBI code for a particular case. -chararray
- **X coordinate:** X coordinate of the location where the incident occurred - float
- **Y coordinate:** Y coordinate of the location where the incident occurred - float
- **Year:** Year of the incident occurred - int
- **Updated on:** Time and date,when details of the case last updated on - DateTime
- **Latitude:** Latitude of the location - float
- **Longitude:** Longitude of the location - float
- **Location:** Range of coordinates - chararray

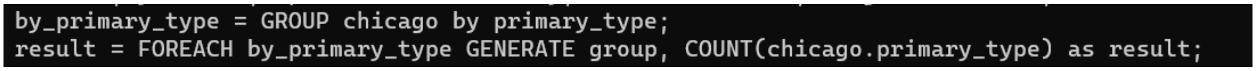
b) Pre-processing and grouping:

For pre-processing, cleaning and grouping we have utilized the PIG framework. For the current task the workflow was:

- Load the CSV File in the PIG framework in local mode.
- To handle “DateTime” objects, we have used the `ToDate()` function.
- Remove headers using `org.apache.pig.piggybank.storage.CSVExcelStorage().`

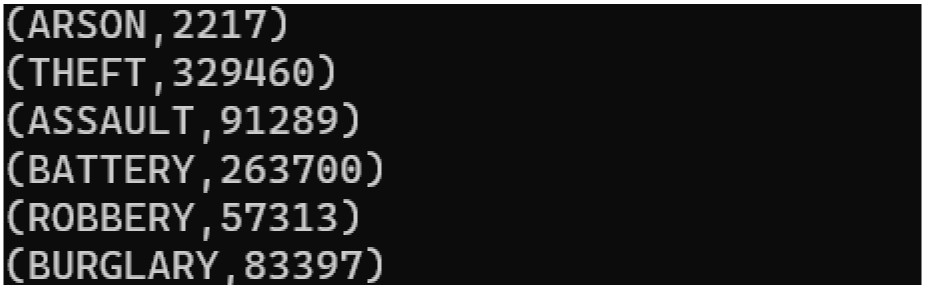
Grouping:

- File name: `group-data.pig.`
- Group the dataset based on the type of crime committed (`primary_type`).
- Aggregate the table by the count of each crime.



```
by_primary_type = GROUP chicago by primary_type;
result = FOREACH by_primary_type GENERATE group, COUNT(chicago.primary_type) as result;
```

- This gives the output.



```
(ARSON,2217)
(THEFT,329460)
(ASSAULT,91289)
(BATTERY,263700)
(ROBBERY,57313)
(BURGLARY,83397)
```

- Store into csv.

Pre-processing:

- File name: `preprocess.pig.`
- For training purposes, we have only taken the following columns: `(arrest, domestic, beat, community_area, latitude, longitude, year);`
- Drop the remaining columns and store into csv.

c) Plots:

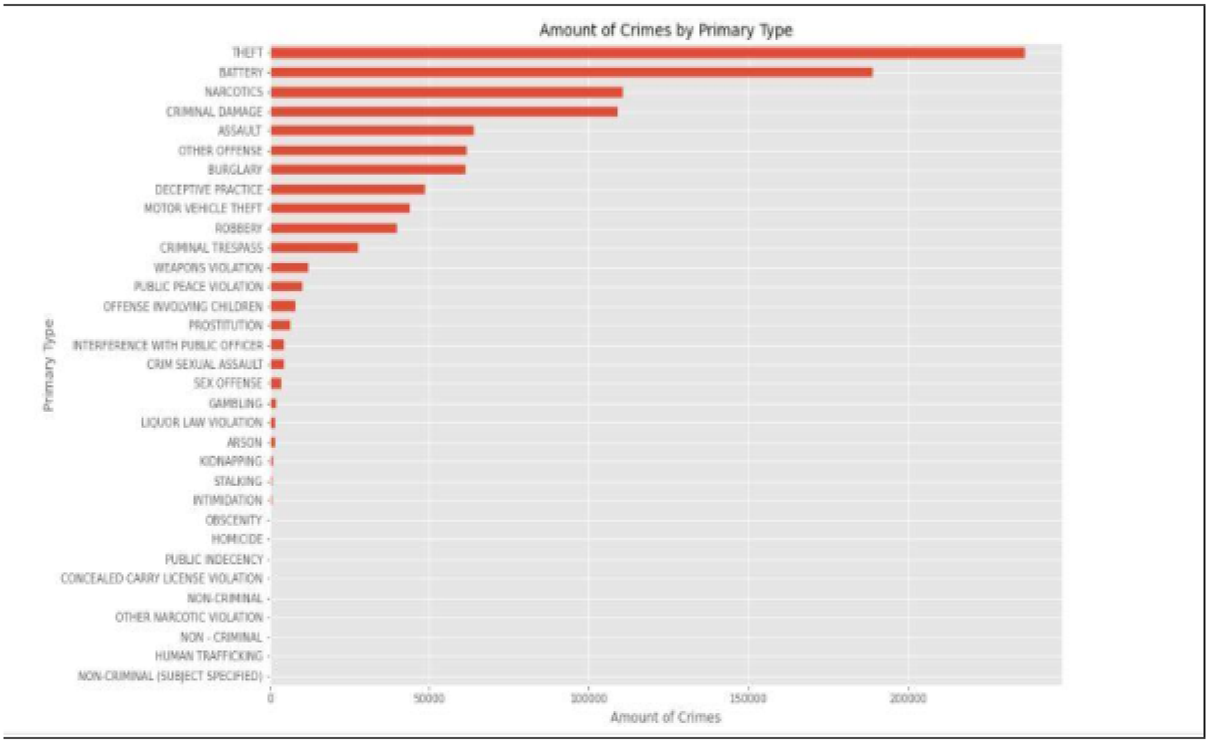


Fig 1: Count of crimes from the year 2012 to 2017.

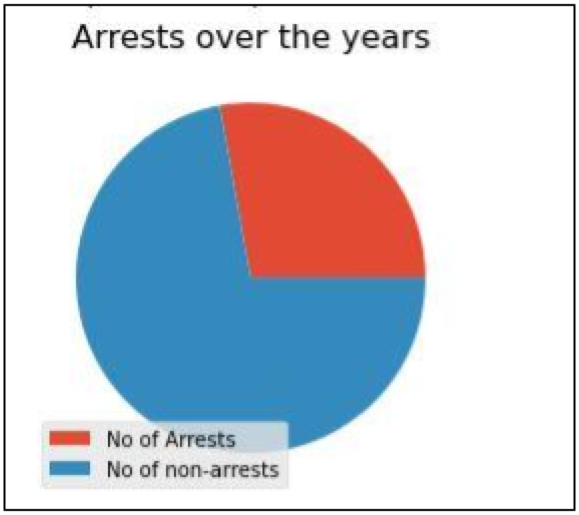


Fig 2: A pie chart of the arrest count.
Notice only 27% were arrested.

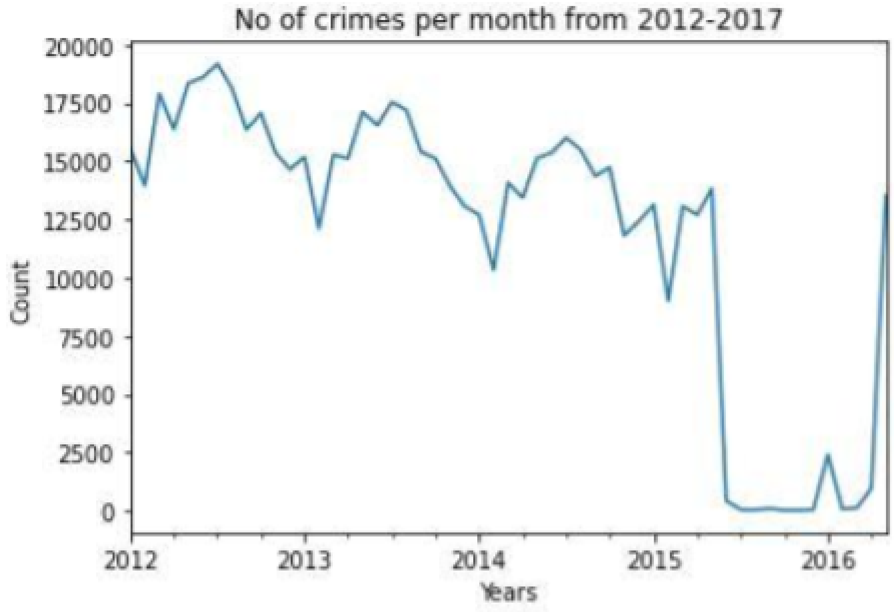
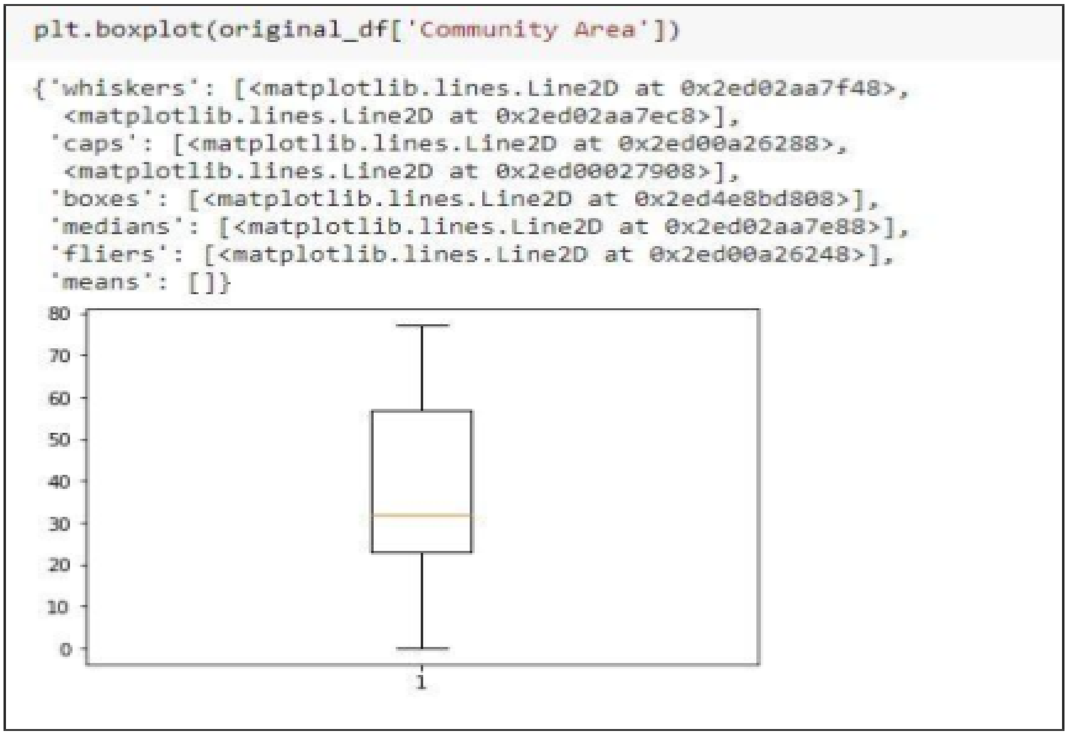


Fig 3: Crimes per month distribution.



```
plt.boxplot(original_df['Community Area'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x2ed02aa7f48>,
  <matplotlib.lines.Line2D at 0x2ed02aa7ec8>],
 'caps': [<matplotlib.lines.Line2D at 0x2ed00a26288>,
  <matplotlib.lines.Line2D at 0x2ed00027908>],
 'boxes': [<matplotlib.lines.Line2D at 0x2ed4e8bd808>],
 'medians': [<matplotlib.lines.Line2D at 0x2ed02aa7e88>],
 'fliers': [<matplotlib.lines.Line2D at 0x2ed00a26248>],
 'means': []}
```

Fig 4: Quantiles displaying range of Community Area.

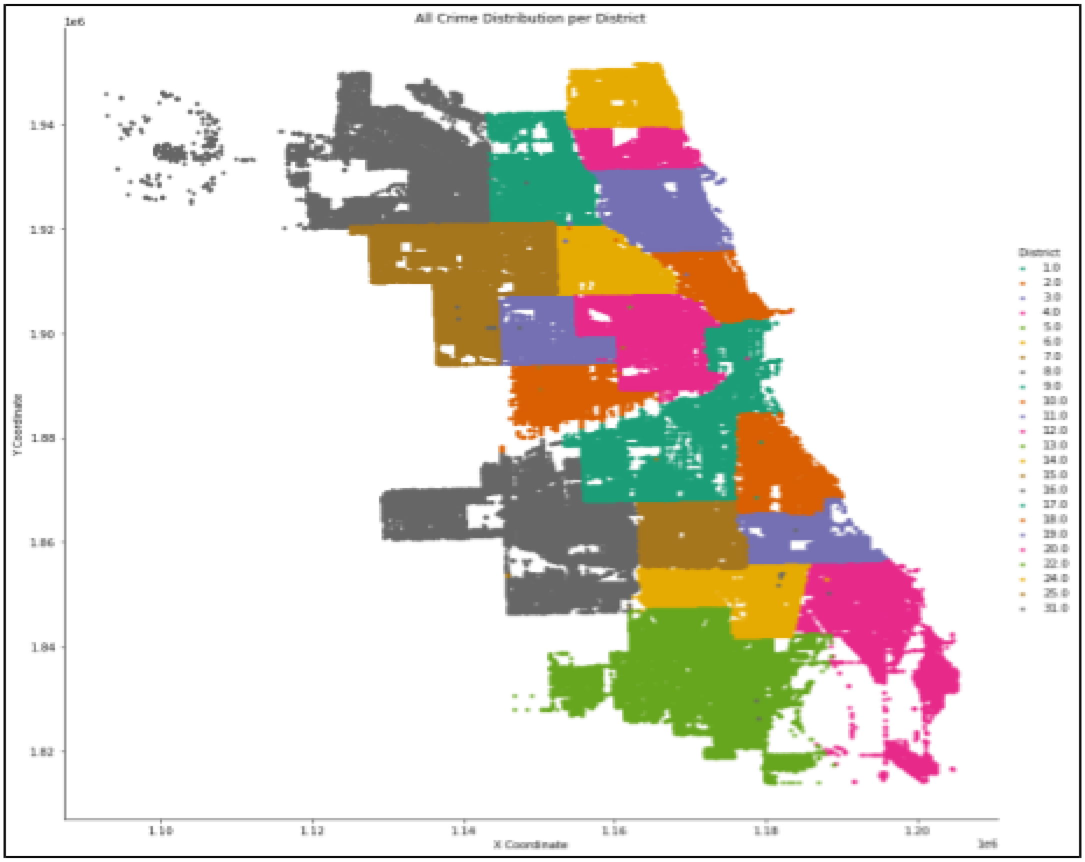
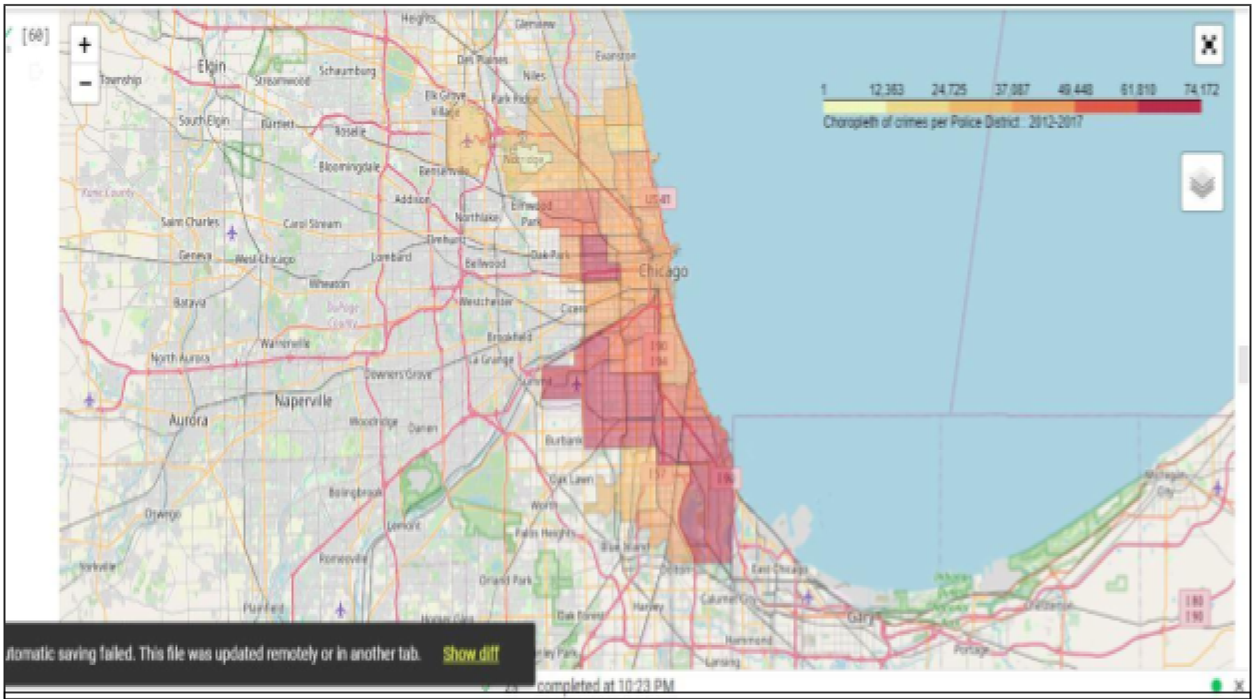


Fig 5: Crime distribution of each district.



utomatic saving failed. This file was updated remotely or in another tab.

completed at 10:23 PM

Fig 6: Geoplot of crime density. Darker shade represents higher density.

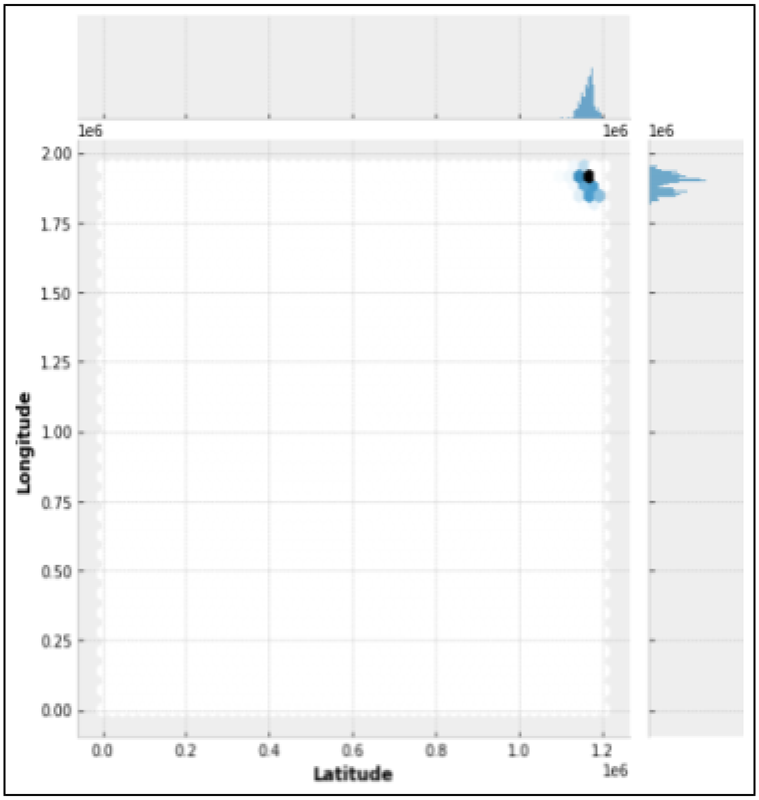
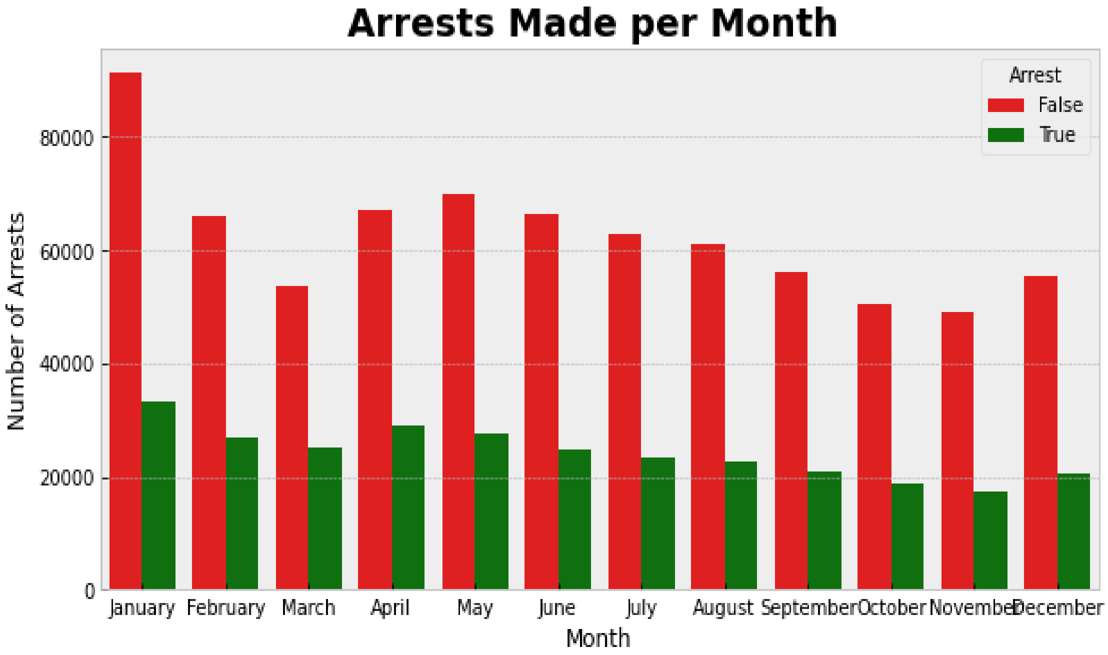


Fig 7: Latitude-Longitude based crime density.

The above plot shows the crime concentration for a specific crime type ‘THEFT’. The most concentrated location coordinates are dark blue.The type of crime can be tweaked in the code.



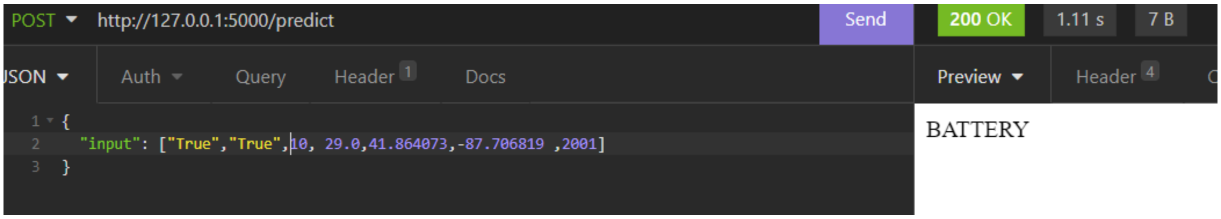
The above bar graph shows the number of arrests made vs the number of not made for each month in the year 2020.

III. Modelling:

- We trained a classifier to predict the type of crime committed from a set of 33 different types of crimes using a unique set of attributes.
- We used OneVSRestClassifier for this specific problem, this strategy consists of fitting one classifier per class. For each classifier, the class is fitted against all the other classes.
- Ada boost classifier is used as estimator for OneVSRestClassifier which is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.
- We chose test_size as 33% of the whole dataset(~=346029)
- Accuracy achieved for classification is **40%**

IV. Deployment:

Trained model is saved as a pickle file and loaded into a flask application to make predictions over data input from an endpoint.



```
{
  "input": ["True","True",10, 29.0,41.864073,-87.706819 ,2001]
}
```

BATTERY

Given `'Arrest'`, `'Domestic'`, `'Beat'`, `'Community Area'`,`'Latitude'`, `'Longitude'`, `'Year'` as input to the model.

Model predicts and outputs `primary_type` (type of crime) attribute.

The above picture shows a sample query which is classified as BATTERY

V. Code:

Please visit the github project for the code.

THE END