Capstone Project - The Battle of Neighborhoods

Siddhant Mahalle

1. Introduction

1.1 Background

Imagine visiting a new city for the first time. You want to visit all the top sites and the best restaurants there. But you have no one to show you around or to suggest these places. You can use numerous travel sites or forums on the internet, but all of those gives different ideas, suggestions and directions. You want to visit these places, but you want to be cautious of crime-riddled neighborhoods and avoid such routes. There is no single site that tells you this. So, what's the solution?

1.2 My Idea

I want to combine the venue and location data from Foursquare along with open source crime data to provide the traveler with a list of attractions and restaurants along with a graphical representation of crime statistics in those areas.

1.3 My Approach

My approach to the problem is as follows:

- The traveler decides on a city location [in this case, Chicago].
- The Foursquare website is scrapped for the top venues in the city
- From this, a list of top venues is augmented with additional geographical data
- Using this data, the top nearby restaurants are selected.
- The historical crime statistics within a predetermined distance of all venues is obtained.
- A map is presented to the traveler showing the selected venues and crime statistics of the area.
- The future probability of a crime happening near or around the selected top sites is also presented to the user.

1.4 Steps for this project

- Data Acquisition
- Data Cleansing
- Data Analysis
- Machine Learning
- Prediction

2. Data

2.1 Data description

This data section is divided in two parts:

- Venues and Location data obtained from FourSquare
- Open source Chicago crime data

2.2 Foursquare Data

This include following steps:

- Query the Foursquare website for the top sites in Chicago
- Use the Foursquare API to get supplemental geographical data about the top sites
- Use the Foursquare API to get top restaurant recommendations closest to each of the top site

Top Sites from Foursquare Website

Although Foursquare provides a comprehensive API, one of the things that API does not easily support is a mechanism to directly extract the top N sites / venues in each city. This data, however, is easily available directly from the Foursquare Website. To do this simply go to www.foursquare.com, enter the city of your choice and select Top Picks from I'm Looking For selection field.

Using BeautifulSoup and Requests the results of the Top Pick for Chicago can retrieved. From this HTML the following data can be extracted:

- Venue Name
- Venue Score
- Venue Category
- Venue HREF
- Venue ID [Extracted from the HREF]

Here is a screenshot of extracted data:

id	score	category	name	href
42b75880f964a52090251fe3	9.7	Park	Millennium Park	/v/millennium-park/42b75880f964a52090251fe3
4b9511c7f964a520f38d34e3	9.6	Trail	Chicago Lakefront Trail	/v/chicago-lakefront-trail/4b9511c7f964a520f38...
49e9ef74f964a52011661fe3	9.6	Art Museum	The Art Institute of Chicago	/v/the-art-institute-of-chicago/49e9ef74f964a5...
4f2a0d0ae4b0837d0c4c2bc3	9.6	Deli / Bodega	Publican Quality Meats	/v/publican-quality-meats/4f2a0d0ae4b0837d0c4c...
4aa05f40f964a520643f20e3	9.6	Theater	The Chicago Theatre	/v/the-chicago-theatre/4aa05f40f964a520643f20e3

Supplemental Geographical Data

Using the id field extracted from the HTML it is then possible to get further supplemental geographical details about each of the top sites from FourSquare using the following sample API call:

```
# Get the properly formatted address and the latitude and longitude
url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(
    venue_id,
    cfg['client_id'],
    cfg['client_secret'],
    cfg['version'])

result = requests.get(url).json()
result['response']['venue']['location']
```

The requests return a JSON object which can then be queried for the details required. The last line in the sample code above returns the following sample JSON:

```
{
   "city":"Chicago",
   "lng":-87.62323915831546,
   "crossStreet":"btwn Columbus Dr & Michigan Ave",
   "neighborhood":"The Loop",
   "postalCode":"60601",
   "cc":"US",
   "formattedAddress":[
      "201 E Randolph St (btwn Columbus Dr & Michigan Ave)",
      "Chicago, IL 60601",
      "United States"
   ],
   "state":"IL",
   "address":"201 E Randolph St",
   "lat":41.8826616030636,
   "country":"United States"
}
```


From this the following attributes are extracted:

- Venue Address
- Venue Postal code
- Venue City
- Venue Latitude
- Venue Longitude

Final FourSquare Top Sites Data

A sample of the final FourSquare Top Sites data is shown below:

id	score	category	name	address	postalcode	city	latitude	longitude
42b75880f964a52090251fe3	9.7	Park	Millennium Park	201 E Randolph St	60601	Chicago	41.882662	-87.623239
4b9511c7f964a520f38d34e3	9.6	Trail	Chicago Lakefront Trail	Lake Michigan Lakefront	60611	Chicago	41.967053	-87.646909
49e9ef74f964a52011661fe3	9.6	Art Museum	The Art Institute of Chicago	111 S Michigan Ave	60603	Chicago	41.879665	-87.623630
4f2a0d0ae4b0837d0c4c2bc3	9.6	Deli / Bodega	Publican Quality Meats	825 W Fulton Market	60607	Chicago	41.886642	-87.648718
4aa05f40f964a520643f20e3	9.6	Theater	The Chicago Theatre	175 N State St	60601	Chicago	41.885578	-87.627286

2.2 Chicago Crime Data

In this part I will be using open source Chicago crime data to provide the user with additional crime data.

This dataset can be download from the Chicago Data Portal and reflects reported incidents of crime (except for murders where data exists for each victim) that occurred in the City of Chicago in the last year, minus the most recent seven days. A full description of the data is available on the site.

Data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. In order to protect the privacy of crime victims' addresses are shown at the block level only and specific locations are not identified.

Column Name	Type	Description
Case Number	Plain Text	The Chicago Police Department RD Number (Records Division Number), which is unique to the incident.
Date	Date & Time	Date when the incident occurred. this is sometimes a best estimate.
Block	Plain Text	The partially redacted address where the incident occurred, placing it on the same block as the actual address.
IUCR	Plain Text	The Illinois Unifrom Crime Reporting code. This is directly linked to the Primary Type and Description. See the list of IUCR codes at https://data.cityofchicago.org/d/c7ck-438e.
Primary Type	Plain Text	The primary description of the IUCR code.
Description	Plain Text	The secondary description of the IUCR code, a subcategory of the primary description.
Location Description	Plain Text	Description of the location where the incident occurred.
Arrest	Plain Text	Indicates whether an arrest was made.
Domestic	Plain Text	Indicates whether the incident was domestic-related as defined by the Illinois Domestic Violence Act.

Beat	Plain Text	Indicates the beat where the incident occurred. A beat is the smallest police geographic area – each beat has a dedicated police beat car. Three to five beats make up a police sector, and three sectors make up a police district. The Chicago Police Department has 22 police districts. See the beats at https://data.cityofchicago.org/d/aerh-rz74.
Ward	Number	The ward (City Council district) where the incident occurred. See the wards at https://data.cityofchicago.org/d/sp34-6z76.
FBI Code	Plain Text	Indicates the crime classification as outlined in the FBI's National Incident-Based Reporting System (NIBRS). See the Chicago Police Department listing of these classifications at http://gis.chicagopolice.org/clearmap_crime_sums/crime_types.html.
X Coordinate	Plain Text	The x coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.
Y Coordinate	Plain Text	The y coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.
Latitude	Number	The latitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
Longitude	Number	The longitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
Location	Location	The location where the incident occurred in a format that allows for creation of maps and other geographic operations on this data portal. This location is shifted from the actual location for partial redaction but falls on the same block.

Not all the attributes are required so only the following data was imported:

- Date of Occurrence
- Block
- Primary Description
- Ward
- Latitude
- Longitude

A sample of the imported data is shown.

	Case Number	Date	Block	Primary Type	Ward	Latitude	Longitude
0	JC540199	12/09/2019 07:30:00 AM	035XX S RHODES AVE	OTHER OFFENSE	4.0	41.830697	-87.614477
1	JC540344	12/09/2019 11:00:00 AM	014XX W FLOURNOY ST	ROBBERY	28.0	41.873334	-87.662844
2	JC541060	12/09/2019 07:55:00 PM	029XX N SOUTHPORT AVE	THEFT	32.0	41.934946	-87.663647
3	JC541313	12/09/2019 11:00:00 PM	002XX N DEARBORN ST	CRIM SEXUAL ASSAULT	42.0	41.886013	-87.629505
4	JC541486	12/09/2019 10:06:00 PM	035XX S RHODES AVE	ASSAULT	4.0	41.830697	-87.614477

2.3 **Clean up the Data and Prepare**

Now that the data has been imported it needs to be cleaned.

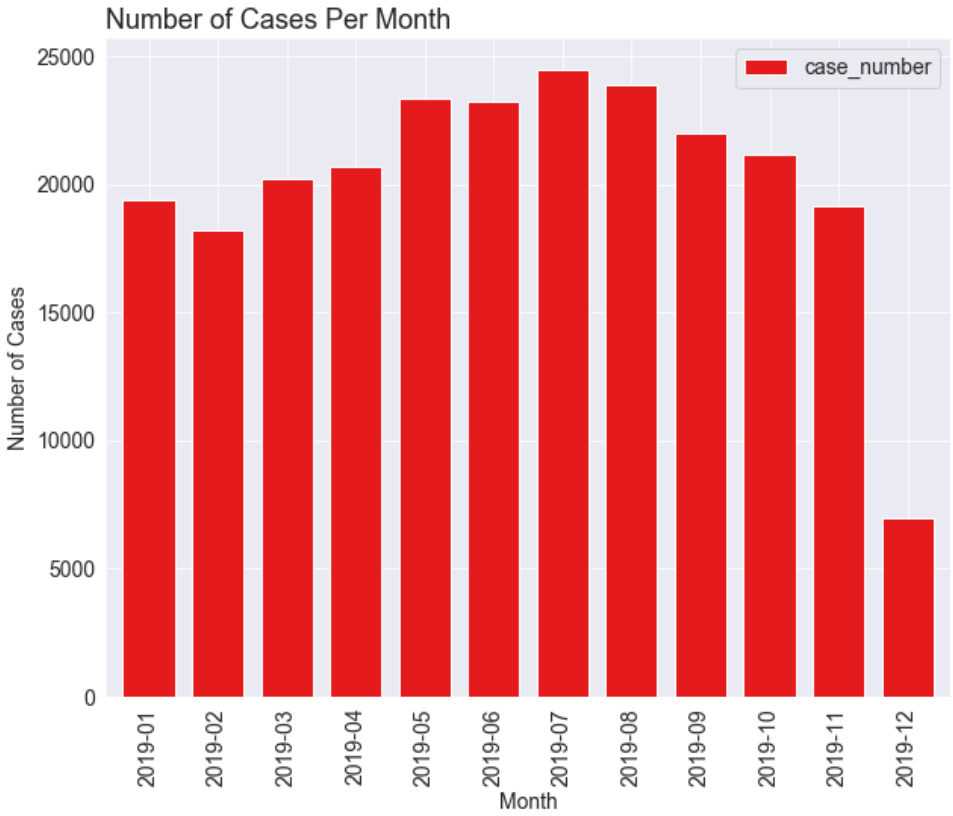
- Clean up the column names:
    1. Strip leading & trailing whitespace
    2. Replace multiple spaces with a single space
    3. Remove # characters
    4. Replace spaces with _
    5. Convert to lowercase
- Change the date of occurrence field to a date / time object
- Add new columns for:
    1. Hour
    2. Day
    3. Month
    4. Year
    5. Etc.
- Split Block into zip code and street
- Verify that all rows have valid data

2.4 Data Analysis and Visualization

Now let's look at some of the attributes and statistics of the crime dataset.

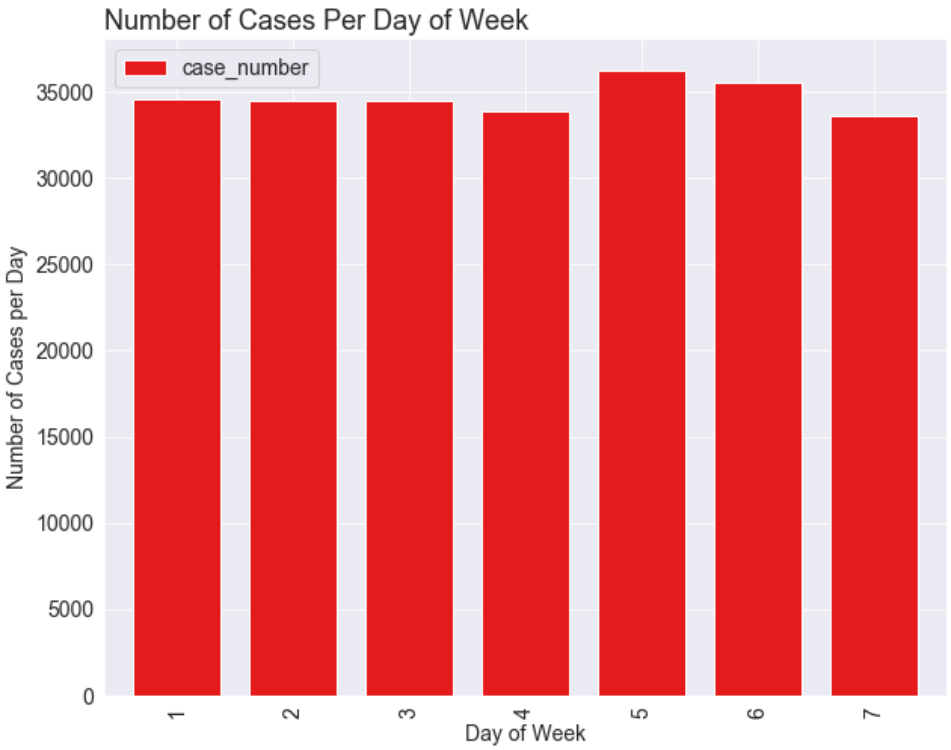
To get a better understanding of the data we will now visualise it. The number of crimes per month, day and hour were calculated.

Let's look at number of crime cases per month:



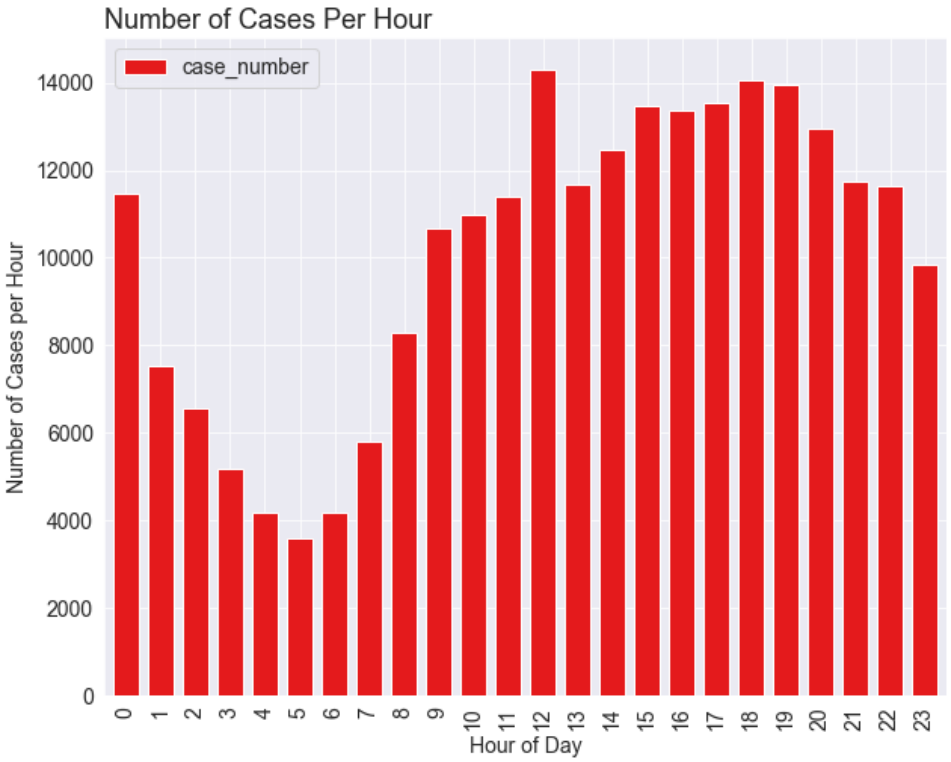
Unsurprisingly there little obvious variation in the number of crimes committed per month other than an apparent increase in july.

Now lets look at number of cases each day of week:



No significant change throughout the week is observed

Now let's look at number of cases each hour:



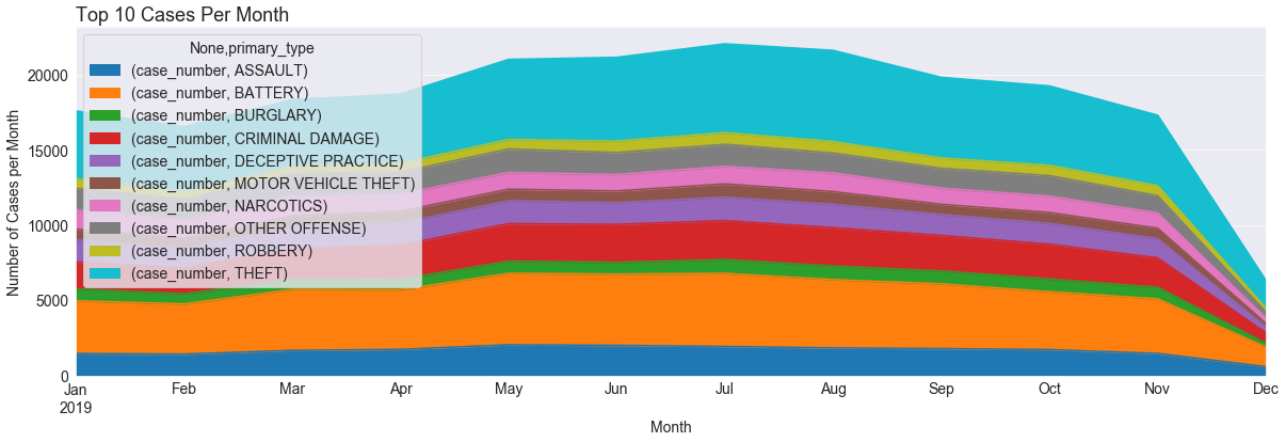
There is an expected fall-off in reported crime rates after midnight before elevating again after eight in the morning. There appears to be a spike around midday.

Now let's look at the 10 most occurring crime.

	primary_type	case_number
29	THEFT	57855
2	BATTERY	46884
6	CRIMINAL DAMAGE	25265
1	ASSAULT	19623
8	DECEPTIVE PRACTICE	16257
22	OTHER OFFENSE	15728
17	NARCOTICS	13245
3	BURGLARY	9010
16	MOTOR VEHICLE THEFT	8434
26	ROBBERY	7498

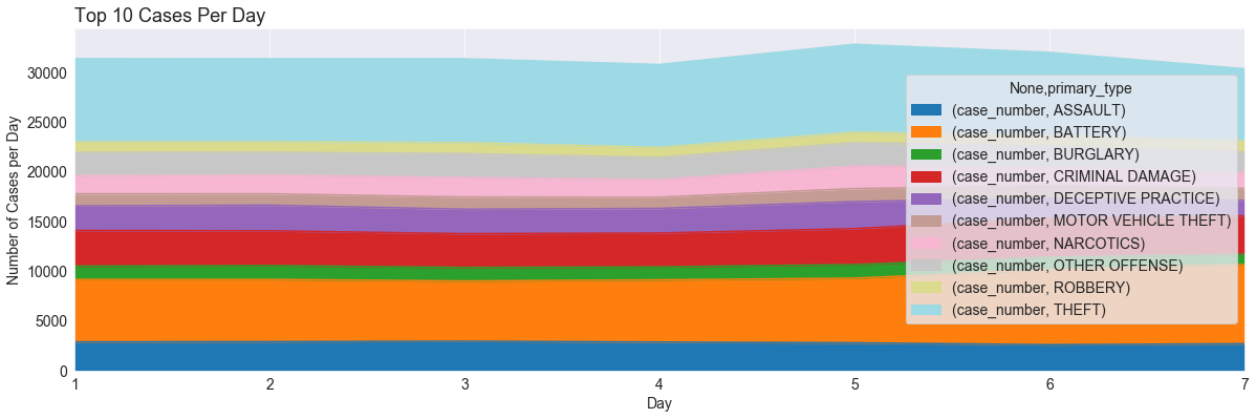
Analysis of these top 10 most occurring crime per month, per day and per hour.

Number of cases per month:



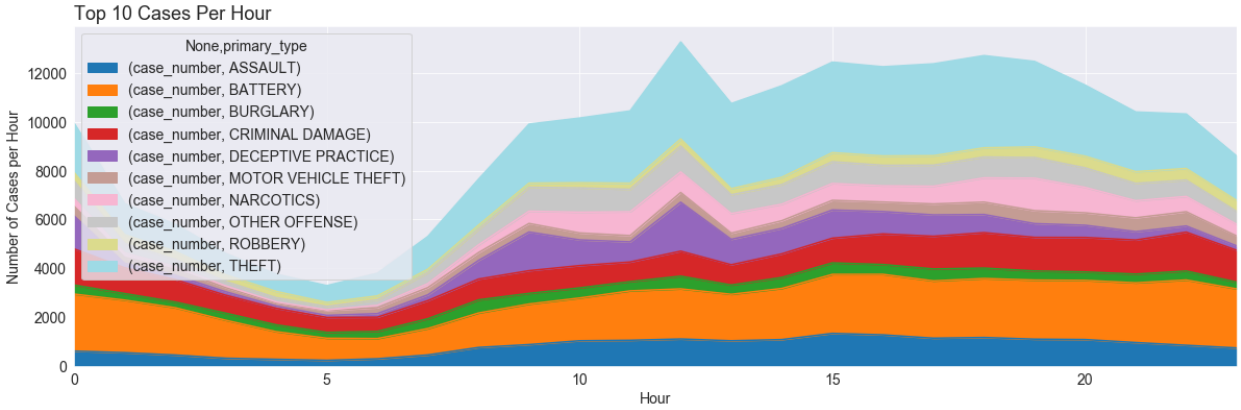
It appears that crimes peak in the Summer months and then fall off in Winter.

Number of cases per day of the week:



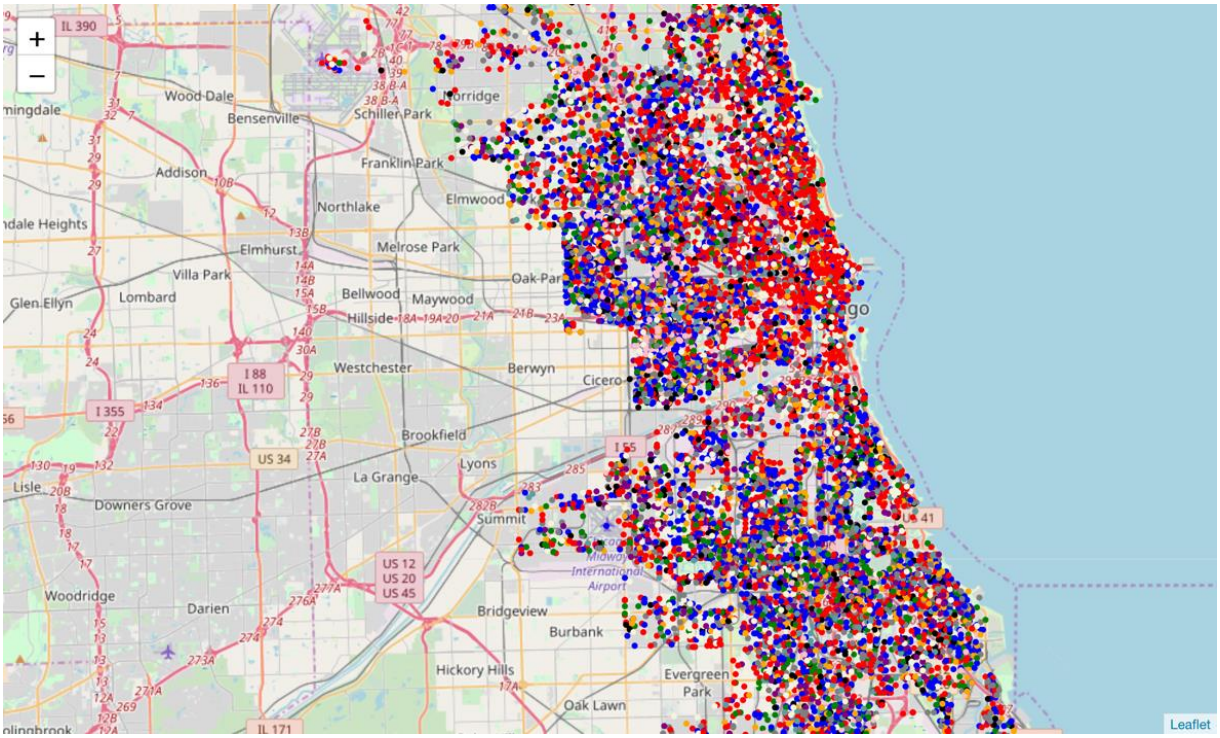
This chart suggests that Saturday, Sunday & Monday (Monday is Day 1) have more crime but that this increase is driven by the crime of Theft.

Number of cases each hour:



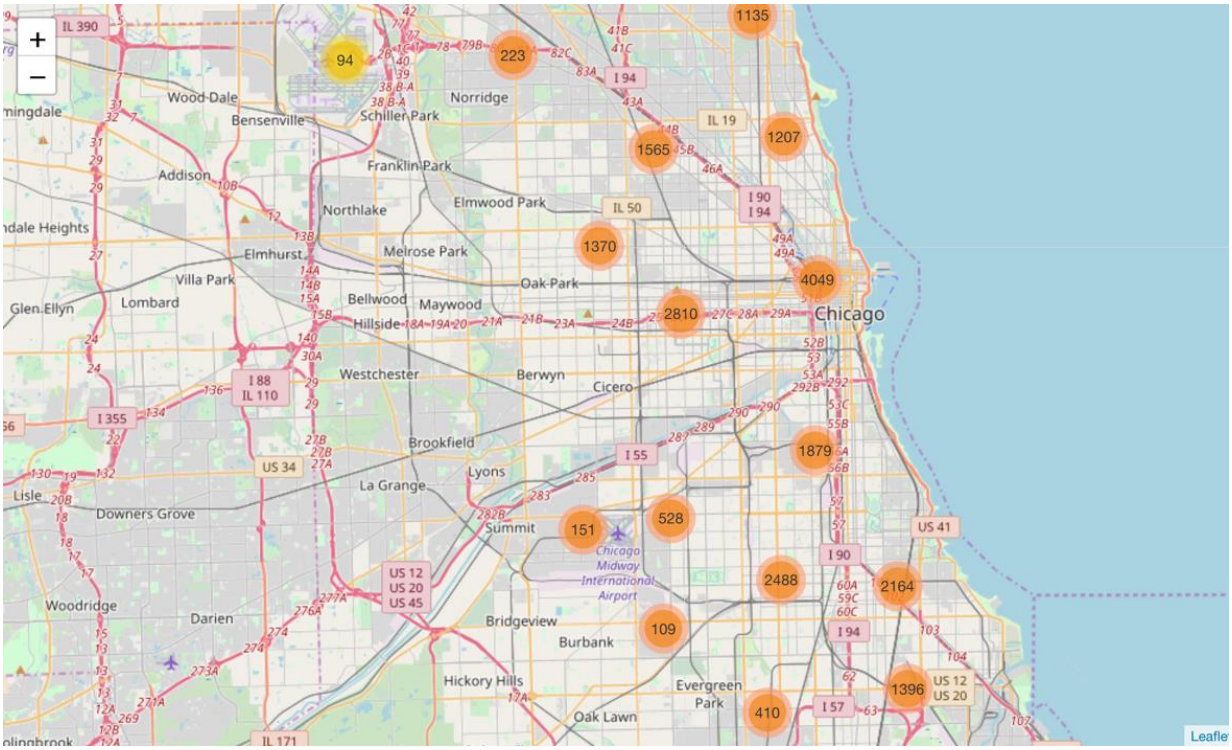
It would appear that 5:00 am in the morning is the safest time in Chicago whilst 12:00 pm in the afternoon is the most dangerous.

Finally, the crimes data for a single month, November 2019 was super-imposed over a map of Chicago to visualise the distribution of that data:



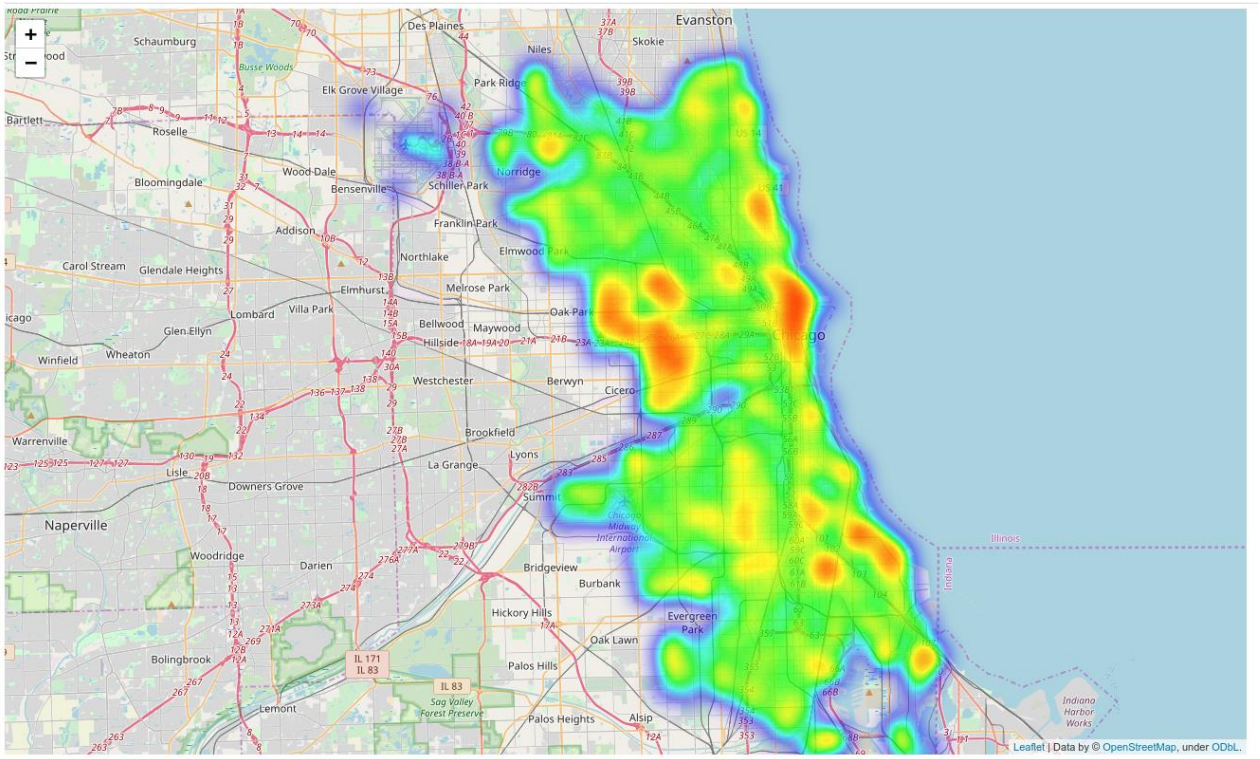
The higher frequency of the top two crimes can be easily seen. Red for Theft and Blue for Battery.

Next the crimes were clustered:



Several obvious clusters of crime locations were visible, particularly in the center of Chicago.

Finally, a heat map of the August crimes was created:



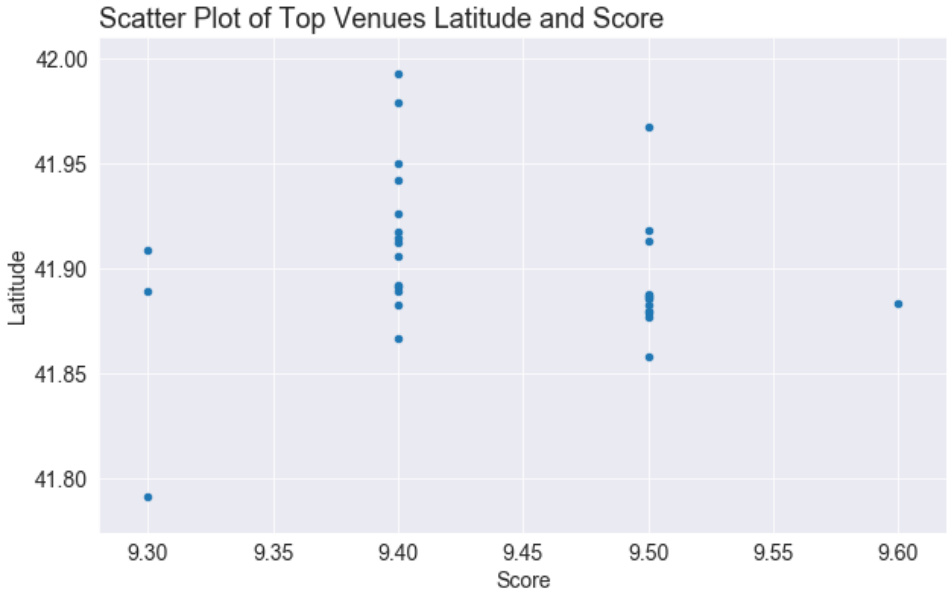
This reinforces the cluster chart where it can clearly be seen that the center of Chicago and the area around Monroe have a high crime rate occurrence. It will be interesting to see later if there is a high probability of crime in these areas if one of the top listed venues are in these areas.

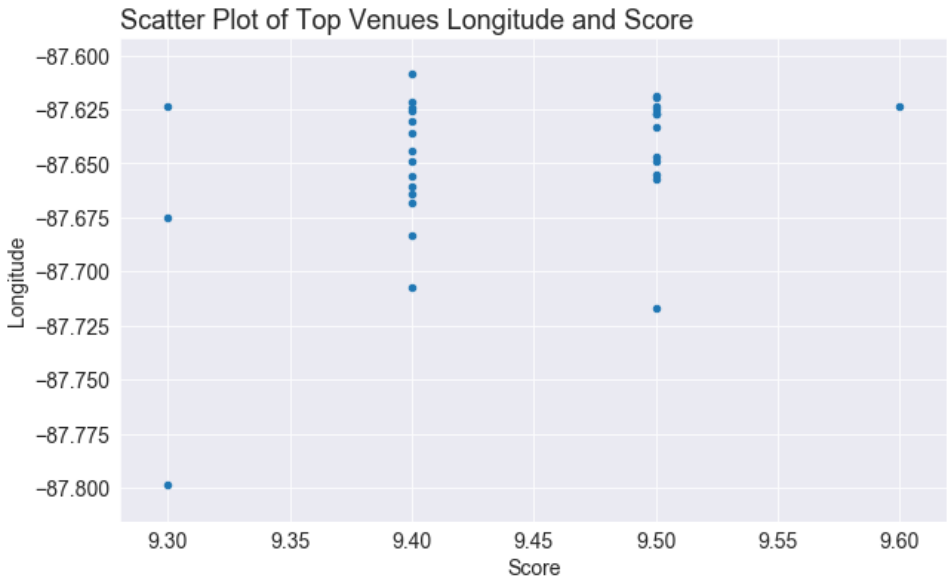
3. Methodology

3.1 Exploratory Data Analysis

The first round of exploratory analysis was to examine the Top Venues and Restaurants Data frames to determine if there was any correlation between variables.

Unfortunately, the only data attributes that could be analyzed were the Latitude and Longitude attributes and their relationship to the venues score. Top Venues was examined first.





Although nothing obvious appears but the top venues are centered around the -87.65 Longitude.

the Restaurant data was examined next.

Unsurprisingly the Restaurant data is also clustered around the -87.65 Longitude given that Restaurants with 500 meters of the top venues were selected.

3.2 Modelling

A new data frame was created after removing unwanted columns and was processed to replace categorical data types with One Hot encoding and the dependent variables were normalized.

Five model type were then chosen to be evaluated:

1. K Nearest Neighbors
2. Decision Trees
3. Logistic Regression
4. Naive Bayes
5. Decision Forest using a Random Forest

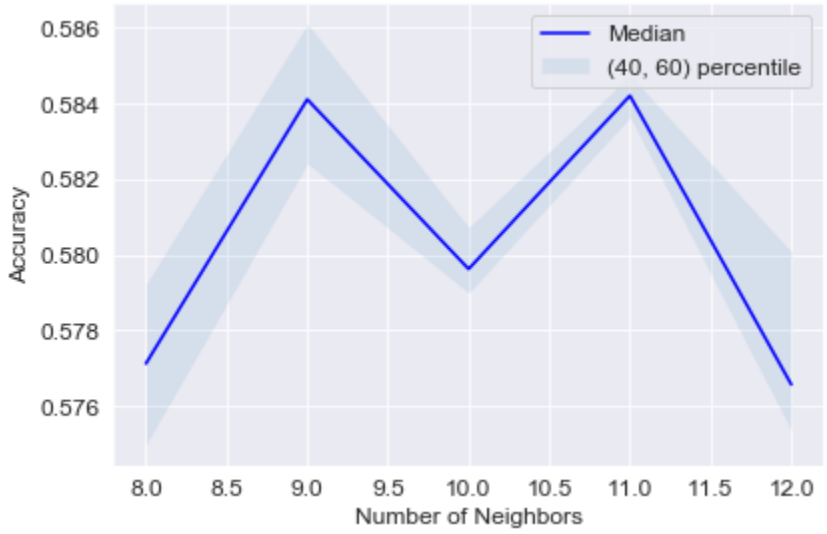
There was one significant issue with the crimes data frame as acquired. Although multiclass classification / prediction is possible, the crimes dataset is unbalanced. Modelling algorithms work best when there is approximately an equal number of samples for each class for example The Curse of Class Imbalance and Class imbalance and the curse of minority hubs.

For this reason, the modelling task was turned into a simple binary classification task by only modelling based on the top two most occurring crimes. For each model development 10-Fold Cross Validation was used to ensure the best results were achieved and a Grid Search approach was used to determine the best setting for each of the models:

3.2.1 - K Nearest Neighbors (KNN)

K Nearest Neighbor (KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN is used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition. KNN algorithm is used for both classification and regression problems.

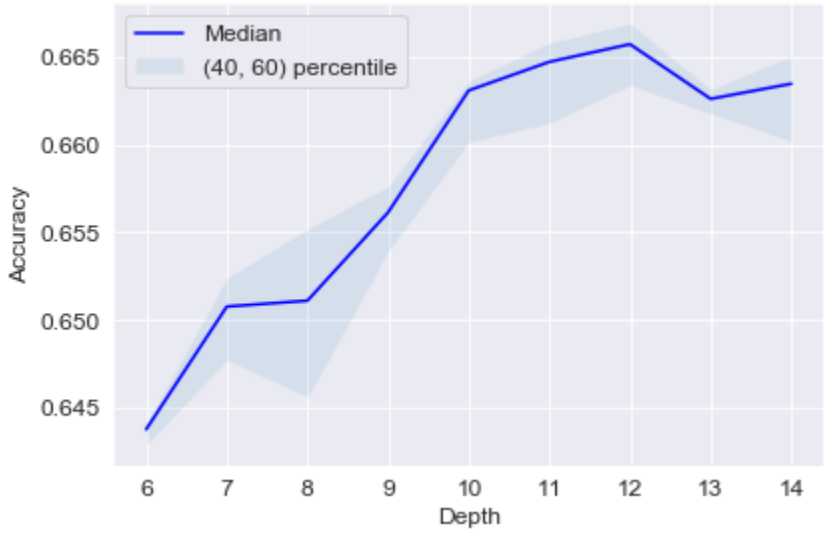
KNN Model was quick to execute and through the process of evaluation it was discovered the K = 11 gave the best results:



3.2.2 - Decision Tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

The Decision Tree model was particularly fast taking only 10 to 15 seconds per model. This meant that it was easy to try multiple different parameters. A tree depth of 12 gave the best model performance:



Depth: 6 2019-12-19 21:19:48.138081

Depth: 7 2019-12-19 21:19:55.593972

Depth: 8 2019-12-19 21:20:04.044722

Depth: 9 2019-12-19 21:20:13.131638

Depth: 10 2019-12-19 21:20:21.405019

Depth: 11 2019-12-19 21:20:28.505880

Depth: 12 2019-12-19 21:20:36.007535

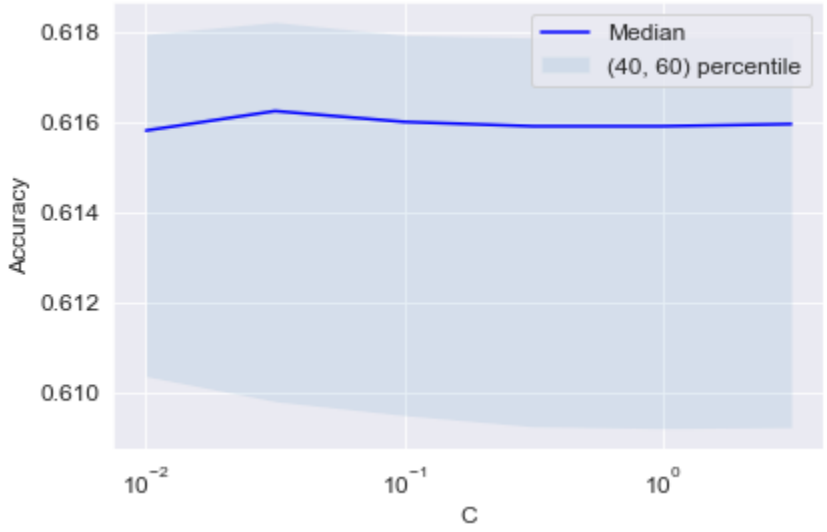
Depth: 13 2019-12-19 21:20:43.852026

Depth: 14 2019-12-19 21:20:52.203263

3.2.2 - Logistic Regression

Logistic Regression model did not return any models with an accuracy greater that 0.61

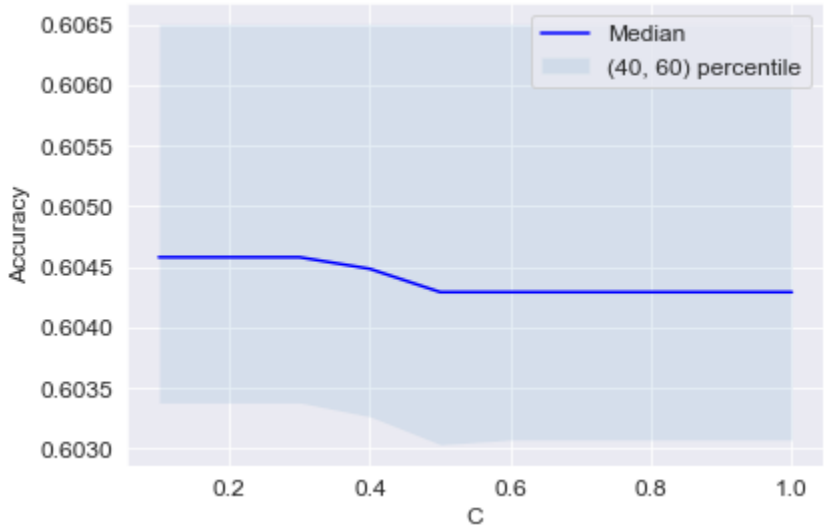
Logistic Regression:



3.2.3 - Naïve Bayes

Naïve Bayes model did not return any models with an accuracy greater that 0.60

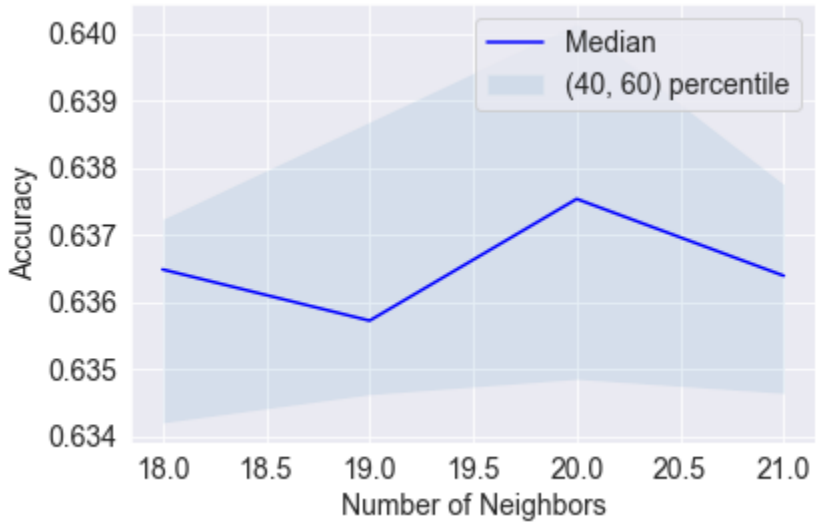
Naïve Bayes:



3.2.4 - Decision Forest using a Random Forest

Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Each model took approximately 4 minutes to create and 20 estimators was found to give the best model accuracy.



3.3 Best Model

Using the crime data for the top two occurring crimes each of the top performing models where further evaluated to determine which model performed the best using F1-Score, Jaccard Score and Log Loss.

Randon forest was determined to be the best model.

Algorithm	Jaccard	F1-Score	LogLoss
KNN	0.687951	0.717168	10.777931
Decision Tree	0.694382	0.703628	10.555772
Bernoulli Naive Bayes	0.611236	0.658090	13.427587
Logistic Regression	0.621219	0.682511	13.082811
Random Forest	0.994625	0.995126	0.185660

3.4 Examination of Random Forest Model

Random Forest is the best model scoring highest in all measurements, F1-Score, Jaccard and Log Loss. Let's now create a new model. The August crime data will become the unseen test data for the final model.

The Top Two Crimes Feature Features Data frame was created again and split into Training Data, everything except August, and Test Data, August.

The F1-Score and Jaccard Score were calculated

Predict the Final Performance of the Model:

```
# Predict yhat using X_Test
yhat = Forest_model_final.predict(X_Test)

# Measure the Jaccard Score of the final Model
jaccard_final = metrics.jaccard_similarity_score(y_Test, yhat)
print('Jaccard Score', jaccard_final)

f1 = metrics.f1_score(y_Test, yhat, average=None)
print('F1-Score of each class', f1)
```

```
Jaccard Score 0.6406824394382128
F1-Score of each class [0.61046393 0.66655003]
```

3.4 Important Features

The most important, or informative, features were then determined. The top ten are shown:

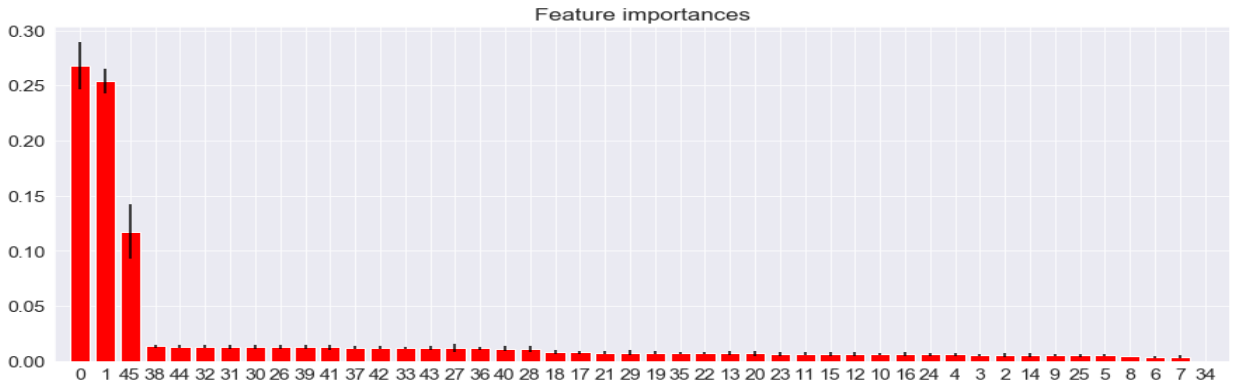
Feature ranking:

1. feature 0 (0.267921)
2. feature 1 (0.254069)
3. feature 45 (0.117401)
4. feature 38 (0.013639)
5. feature 44 (0.013212)
6. feature 32 (0.013077)
7. feature 31 (0.013024)
8. feature 30 (0.012809)
9. feature 26 (0.012749)
10. feature 39 (0.012684)

This shows that the most predictive models are:

1. Latitude
2. Longitude
3. Ward

After these the day and the month of the crime are weak predicters at around 1.1%. The other features, particularly the hour the crime took place, are hardly predictive at all. A plot of this is shown below:



4. Results and Prediction

The idea for the Capstone Project is to show that when driven by venue and location data from FourSquare, backed up with open source crime data, that it is possible to present the cautious and nervous traveler with a list of attractions to visit supplemented with a graphics showing the occurrence of crime in the region of the venue.

A high-level approach is as follows:

- The travelers decide on a city location [in this case Chicago]
- The ForeSquare website is scrapped for the top venues in the city
- From this list of top venues, the list is augmented with additional geographical data
- Using this additional geographical data, the top nearby restaurants are selects
- The historical crime within a predetermined distance of all venues are obtained
- A map is presented to the to the traveler showing the selected venues and crime statistics of the area.
- The future prediction of a crime happening near or around the selected top sites is also presented to the user

So, all goals have been achieved except the final one. In this Results and Predictions Section this goal is addressed.

The purpose of this project was to see if crime can be predicted. However, the nature of the dataset, particularly the number of different crimes and the unbalanced nature of the dataset, makes it difficult to predict what crime will predict and when. We can, however, repurpose the Crimes Data Frame by splitting the dataset into two distinct balanced sets and randomly assigning to 0 to represent no crime and 1 to present a crime happening. The data set looked like this:

	latitude	longitude	hour_0	hour_1	hour_2	hour_3	hour_4	hour_5	hour_6	hour_7	hour_8	hour_9	hour_10	hour_11	hour_12	hour_13	hour_14	hour_1
0	41.871442	-87.725365	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	41.794537	-87.631127	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
2	41.760490	-87.655132	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
3	41.954322	-87.749923	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	41.968235	-87.728572	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Tuesday	Wednesday	April	August	December	February	January	July	June	March	May	November	October	September	ward	crimes	random_crimes
0	1	0	0	1	0	0	0	0	0	0	0	0	0	24.0	CRIMINAL DAMAGE	1
0	0	0	0	1	0	0	0	0	0	0	0	0	0	3.0	ROBBERY	1
0	1	0	0	1	0	0	0	0	0	0	0	0	0	17.0	BATTERY	1
0	1	0	0	1	0	0	0	0	0	0	0	0	0	45.0	OTHER OFFENSE	0
0	1	0	0	1	0	0	0	0	0	0	0	0	0	39.0	THEFT	0

The test data was constructed from the Top Venues Data Frame and the Restaurants Dataframe as follows:

- The two dataframes were joined to form a single dataframe. The venue or restaurant name and the latitude and longitude attributes were added.
- Duplicate entries were dropped as some restaurants appeared multiple times in the dataframe
- Next a random date and time was assigned to each venue.
- The date was then split into Hour, Day of Week, Month and Year as described above

- The data was finally prepared for prediction by applying One Hot encoding and then extracted into a new dataframe that match the format used to create the model.
- y^ (y_hat) or the predictions were then made

The results of the predictions are shown below:

```
# Predict whether crime will happen a each location in the dataframe
yhat = Forest_model_final.predict(df_features_final)
```

```
# Display all the predictions - 0 for possible criminal location and 1 for safe
yhat
```

```
array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0])
```

And the Predictions were re-added to the data (as it was before One Hot encoding was applied).

	name	latitude	longitude	date	hour	day_name	day	month_name	month	year	year_month	prediction
0	Millennium Park	41.883112	-87.623851	2019-03-26 20:09:00	20	Tuesday	1	March	3	2019	2019-03	0
1	The Art Institute of Chicago	41.879610	-87.623552	2019-06-05 00:25:00	0	Wednesday	2	June	6	2019	2019-06	0
2	Grant Park	41.876626	-87.619263	2019-12-05 01:35:00	1	Thursday	3	December	12	2019	2019-12	0
3	Chicago Riverwalk	41.887280	-87.627217	2019-01-12 19:39:00	19	Saturday	5	January	1	2019	2019-01	1
4	Binny's Beverage Depot	41.913048	-87.655320	2019-06-05 18:36:00	18	Wednesday	2	June	6	2019	2019-06	0
5	Garfield Park Conservatory	41.886259	-87.717177	2019-05-07 17:42:00	17	Tuesday	1	May	5	2019	2019-05	0
6	Symphony Center (Chicago Symphony Orchestra)	41.879275	-87.624680	2019-08-13 18:09:00	18	Tuesday	1	August	8	2019	2019-08	0
7	The Chicago Theatre	41.885539	-87.627151	2019-08-05 09:18:00	9	Monday	0	August	8	2019	2019-08	0
8	Publican Quality Meats	41.886642	-87.648718	2019-10-27 09:27:00	9	Sunday	6	October	10	2019	2019-10	0
9	Thalia Hall	41.857832	-87.657292	2019-06-17 23:37:00	23	Monday	0	June	6	2019	2019-06	0
10	Chicago Lakefront Trail	41.967053	-87.646909	2019-10-10 02:35:00	2	Thursday	3	October	10	2019	2019-10	0
11	Nature Boardwalk	41.918102	-87.633283	2019-09-09 15:36:00	15	Monday	0	September	9	2019	2019-09	0
12	Maggie Daley Park	41.882905	-87.618846	2019-12-02 17:12:00	17	Monday	0	December	12	2019	2019-12	0
13	Jay Pritzker Pavilion	41.882614	-87.621782	2019-03-27 03:19:00	3	Wednesday	2	March	3	2019	2019-03	0
14	Restoration Hardware	41.906034	-87.630504	2019-01-20 08:52:00	8	Sunday	6	January	1	2019	2019-01	0
15	Joe's Seafood, Prime Steak & Stone Crab	41.891828	-87.625444	2019-11-21 02:53:00	2	Thursday	3	November	11	2019	2019-11	0
16	Weber's Bakery	41.791695	-87.798656	2019-03-16 19:29:00	19	Saturday	5	March	3	2019	2019-03	1
17	Lost Larson Bakery	41.978617	-87.668411	2019-06-08 14:33:00	14	Saturday	5	June	6	2019	2019-06	1
18	Ghirardelli Ice Cream And Chocolate Shop - Wri...	41.889230	-87.624487	2019-11-17 08:30:00	8	Sunday	6	November	11	2019	2019-11	0
19	Chicago Lakefront	41.866562	-87.608781	2019-12-17 19:28:00	19	Tuesday	1	December	12	2019	2019-12	0

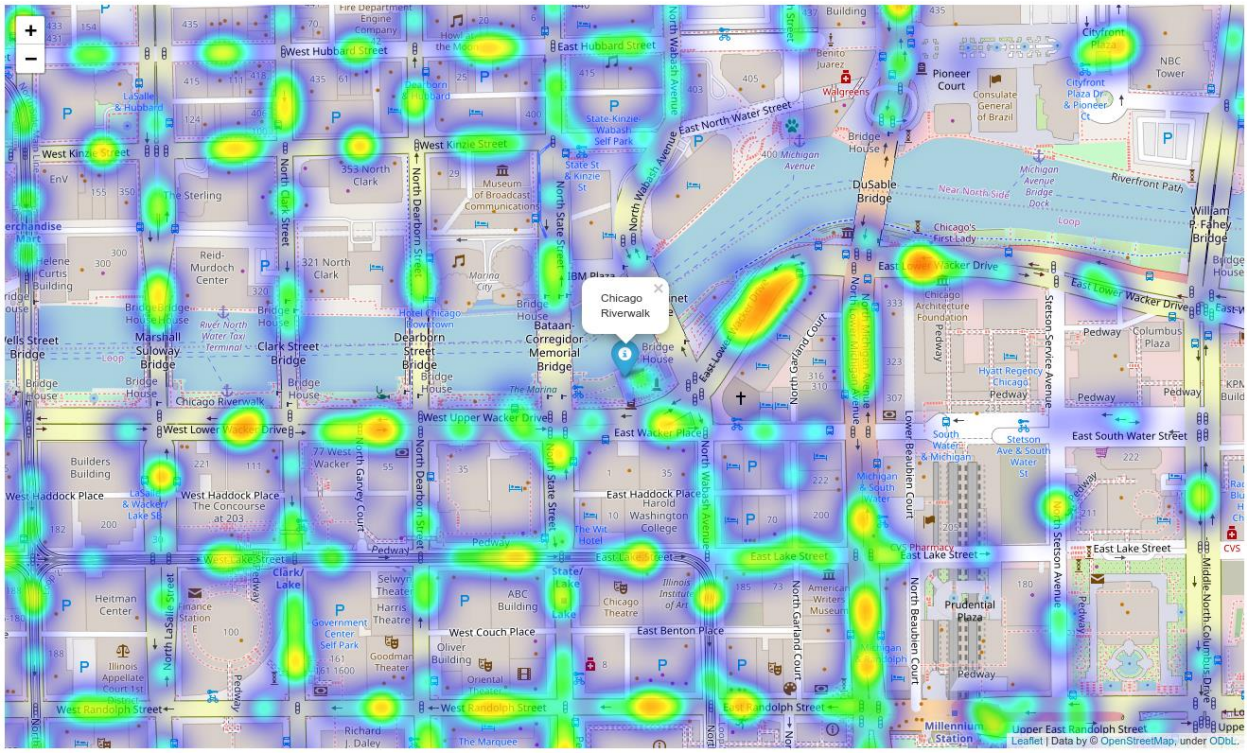
Visualisation of Predictions:

Of the top 20 venues 17 were identified as potentially dangerous to visit and 3 was deems safe. As there is no data to compare the predictions against the best way, we will visualise the data again.

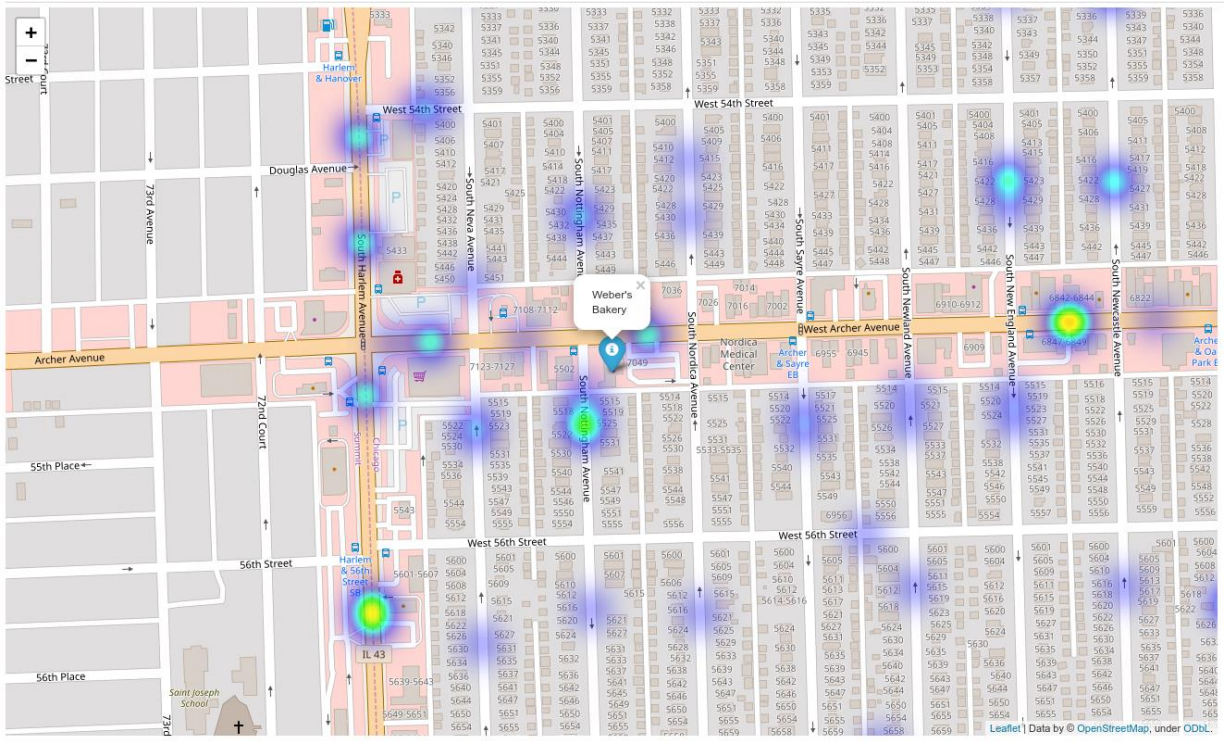
We will look at the following 5 venues:

1. Millennium Park (41.882699, -87.623644)
2. Chicago Riverwalk (41.887280, -87.627217)
3. The Chicago Theatre (41.885539, -87.627151)
4. Weber's Bakery (41.791695, -87.798656)
5. Chicago Lakefront (41.866562, -87.608781)

The Distance Dataframe is recreated again but this time all crimes are included. The Chicago Riverwalk and Weber's Bakery both were identified as likely to be susceptible to crime.




The Chicago Theatre
Leaflet | Data by © OpenStreetMap, under ODbL.


Weber's Bakery
Leaflet | Data by © OpenStreetMap, under ODbL.

5. Conclusions and Discussions

Although all the goals of this project were met there is room for further improvement and development as noted below. However, the goals of the project were met and, with some more work, could easily be developed into a fully-fledged application that could support the cautious traveler in an unknown location.

Of the contributing data the Chicago Crime data is the one where more data would be good to have. Also, not every city in the world makes this data freely available so that is a drawback.

FourSquare proved to be a good source of data but frustrating at times. Despite having a Developer account, I regularly exceeded my hourly limit locking me out for the day. Therefore, Pickle was used to store the captured data.

6. Further Development

The following are suggestions how this project could be further developed:

- Best time to visit each venue
- Suggestions for morning, afternoon, evening and nighttime
- Daily itineraries
- Route planning and transportation
- Time lapse of the crime in the area of the venue
- Favorite dining preferences could be used to choose the restaurants

GitHub Link: https://github.com/siddhantmahalle/Applied-Data-Science-Capstone-

****