**1 : Introduction**

In this section I will clearly define the idea of my choosing, where I leverage the Foursquare location data to solve the imagined business opportunity.

**Background**

There are 100's, maybe even 1000's, of travel sites on the Internet, including four square, that will tell you all about places to go, things to see, restaurants to eat at, bars to drink in, nightclubs to part the night away in and then where to go in the morning to get breakfast and a strong coffee. The problems with these sites is that they are one dimensional. If you want to find out all this information about a city you plan to visit next month,  have to do the hard work. Also, just because a venue is the hottest place to go for a night out does not always mean that the unwitting tourist should just ramble in unprepared. The areas surrounding this new venue might be riddled with crime including muggings, car theft and assault, for example. Approach the venue from any direction other than from the north and you could be putting your life in danger. This is when my idea comes in.

Imagine the following scenario:

1. You like to plan ahead and always review your options and make your choices about where you will visit and eat up front before you travel.
2. You are flying to Chicogo for a Data Science Conference.
3. You arrive in Chicago the day the conference starts but you've managed to convince your boss to delay your return by a few days giving you time to explore.
4. But you know no one in Chicago to show you around to all the top sites and to bring you to the best restaurants.
5. Also the last time you went to a conference you were mugged and had you passport. money and credit cards stolen so you're now nervous of going somewhere without first researching the venue and the surrounding area.
6. The conference is next week and you don't have time to do all the research you'd like.
7. The data is extracted from the crime data of chicago

**What do you do ... ?**

**Project Idea**

My 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 traveller with a list of attractions to visit supplementd with a graphics showing the occurance of crime in the region of the venue.

A high level approach is as follows:

1. The travellers decides on a city location [in this case Chicago]
2. The ForeSquare website is scrapped for the top venues in the city
3. From this list of top venues the list is augmented with additional grographical data
4. Using this additional geographical data the top nearby restaurents are selects
5. The historical crime within a predetermined distance of all venues are obtained
6. A map is presented to the to the traveller showing the selected venues and crime statistics of the area.
7. The future probability of a crime happening near or around the selected top sites is also presented to the user

**Who is this solution targeted at**

This solution is targeted at the cautious traveller. The want to see all the main sites of a city that they have never visited before but at the same time, for whatever reaons unknown, they want to be able to do all that they can to make sure that they stay clear of trouble i.e. is it safe to visit this venue and this restaurant at 4:00 pm in the afternoon.

Some examples of envisioned users include:

* A single white female traveller
* An elderly traveller that has had previous back experiences when travelling

There are many data science aspect of this project including:

1. Data Acquisition
2. Data Cleansing
3. Data Analysis
4. Machine Learning
5. Prediction

Now that the conference is over the Data Sceintist can explore Chigago and feel much safer.

**2: Data**

**Data Description**

In this section, I will describe the data used to solve the problem as described previously.

As noted below in the Further Development Section, it is possible to attempt quite complex and sophisticated scenarios when approaching this problem. However, given the size of the project and for simplicity only the following scenario will be addressed:

1. Query the FourSqaure website for the top sites in Chicago
2. Use the FourSquare API to get supplemental geographical data about the top sites
3. Use the FourSquare API to get top restaurent recommendations closest to each of the top site
4. Use open source Chicago Crime data to provide the user with additional crime data

**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 a given city. This data, however, is easily available directly from the FourSquare Website. To do this simply go to four square API , enter the city of your choise and select Top Picks from *I'm Looking For*selection field.

Using BeautifulSoup and Requests the results of the Top Pick for Chicago was retrieved. A sample venue is shown below:

<div class="venueDetails">

<div class="venueName">

<h2>

<a href="/v/millennium-park/42b75880f964a52090251fe3" target="\_blank">Millennium Park

</a>

</h2>

</div>

<div class="venueMeta">

<div class="venueScore positive" style="background-color: #00B551;" title="9.7/10 - People like this place">9.7</div>

<div class="venueAddressData">

<div class="venueAddress">201 E Randolph St (btwn Columbus Dr &amp; Michigan Ave), Chicago</div>

<div class="venueData"><span class="venueDataItem"><span class="categoryName">Park</span><span class="delim"> • </span></span>

</div>

</div>

</div>

</div>

From this HTML the following data can be extracted:

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

We will have a closer look at this data gather later on when the supplemental geographical data has been added.

**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 returns 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 Postalcode
* Venue City
* Venue Latitude
* Venue Longitude

**Final FourSquare Top Sites Data**

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

**Data Analysis and Visualisation**

An initial look at the data shows that there are 30 rows of data [as expected] each with 10 attributes. The variable types are all correct except the Venue Rating or Score which will be converted to a float. After converting the score column to a float it can clearly be seen that we have the top venues with a mean of 9.532.

df\_top\_venues.shape

(30, 10)

df\_top\_venues.dtypes

id object

score object

category object

name object

address object

postalcode object

city object

href object

latitude float64

longitude float64

dtype: object

df\_top\_venues.score.describe()

count 30.000000

mean 9.523333

std 0.072793

min 9.400000

25% 9.500000

50% 9.500000

75% 9.600000

max 9.700000

Name: score, dtype: float64

We are now ready to get the top restaurents within 500 meters of each of the top sites.

**FourSquare Restaurent Recommendation Data**

Using the the list of all id values in the Top Sites DataFrame and the FourSquare categoryID that represents all food venues we now search for restaurants within a 500 meter radius.

# Configure additional Search parameters

categoryId = '4d4b7105d754a06374d81259'

radius = 500

limit = 15

url = 'https://api.foursquare.com/v2/venues/search?client\_id={}&client\_secret={}&ll={},{}&v={}&categoryId={}&radius={}&limit={}'.format(

cfg['client\_id'],

cfg['client\_secret'],

ven\_lat,

ven\_long,

cfg['version'],

categoryId,

radius,

limit)

results = requests.get(url).json()

The requests returns a JSON object which can then be queried for the restaurant details required. A sample restaurnt from the results returned is shown below:

{

"referralId":"v-1538424503",

"hasPerk":"False",

"venuePage":{

"id":"135548807"

},

"id":"55669b9b498ee34e5249ea61",

"location":{

"labeledLatLngs":[

{

"label":"display",

"lng":-87.62460021795313,

"lat":41.88169538551873

}

],

"crossStreet":"btwn E Madison & E Monroe St",

"postalCode":"60603",

"formattedAddress":[

"12 S Michigan Ave (btwn E Madison & E Monroe St)",

"Chicago, IL 60603",

"United States"

],

"distance":155,

"city":"Chicago",

"lng":-87.62460021795313,

"neighborhood":"The Loop",

"cc":"US",

"state":"IL",

"address":"12 S Michigan Ave",

"lat":41.88169538551873,

"country":"United States"

},

"name":"Cindy's",

"categories":[

{

"pluralName":"Gastropubs",

"id":"4bf58dd8d48988d155941735",

"name":"Gastropub",

"primary":"True",

"icon":{

"prefix":"https://ss3.4sqi.net/img/categories\_v2/food/gastropub\_",

"suffix":".png"

},

"shortName":"Gastropub"

}

]

},

From this JSON the following attributes are extraced and added to the Dataframe:

* Restaurant ID
* Restaurant Category Name
* Restaurant Category ID
* Restaurant Nest\_name
* Restaurant Address
* Restaurant Postalcode
* Restaurant City
* Restaurant Latitude
* Restaurant Longitude
* Venue Name
* Venue Latitude
* Venue Longitude

The only piece of data that is missing is the Score or Rating of the Restaurant. To get this we need to make another FourSquare API query using the id of the Restaurant:

# Get the restaurant score and href

rest\_url = 'https://api.foursquare.com/v2/venues/{}?client\_id={}&client\_secret={}&v={}'.format(

rest\_id,

cfg['client\_id'],

cfg['client\_secret'],

cfg['version'])

result = requests.get(rest\_url).json()

rest\_score = result['response']['venue']['rating']

Using just the data in this DataFrame we will be able to generate maps displaying the chosen Top List Venue and the best scored surrounding restaurants. A sample of this data is shown below:

Looking at the data we get an interesting insight into the range of restuarants that are included. From a list of 30 top venues only 28 actually had more than 10 to provide the user with a real choice. In total there were 387 restaurants found of which 240 were unique occuring only once in the data. There were 72 categories of restaurants. The mean score of all the restaurants wa 8.23 with a manimum value of 9.5 and a minimum value of 5.3.

Coffee Shops (52) and Pizza Places (29) were the top two most frequently occurring categories but Pie Shops (9.4000) and French Restaurants (9.4000) were the restaurant categories with the highest average score.

# What is the shape of the Restaurants DataFrame

df\_restaurant.shape

(387, 13)

# Get a count of the top venues that had more than 10 restaurant within 500 meters

# The number of unique restaurants

# The number of unique restaurant categories

df\_restaurant.venue\_name.nunique()

28

df\_restaurant.name.nunique()

240

df\_restaurant.category.nunique()

72

# Look at the data types

df\_restaurant.dtypes

id object

score float64

category object

categoryID object

name object

address object

postalcode object

city object

latitude float64

longitude float64

venue\_name object

venue\_latitude float64

venue\_longitude float64

dtype: object

# Describe the Score attribute

df\_restaurant.score.describe()

count 387.000000

mean 8.286563

std 0.930138

min 5.300000

25% 7.800000

50% 8.500000

75% 9.000000

max 9.500000

Name: score, dtype: float64

df\_restaurant.groupby('category')['name'].count().sort\_values(ascending=False)[:10]

category

Coffee Shops 52

Pizza Places 29

Cafés 24

Bakeries 15

Burger Joints 15

Gastropubs 15

New American Restaurants 15

Mexican Restaurants 14

Breakfast Spots 13

Fast Food Restaurants 13

df\_restaurant.groupby('category')['score'].mean().sort\_values(ascending=False)[:10]

category

Pie Shops 9.4000

French Restaurants 9.4000

Molecular Gastronomy Restaurants 9.3000

Filipino Restaurants 9.2000

Cuban Restaurants 9.1000

Ice Cream Shops 9.0625

Mediterranean Restaurants 9.0600

Korean Restaurants 9.0000

Latin American Restaurants 9.0000

Fish & Chips Shops 9.0000

**Chicago Crime Data**

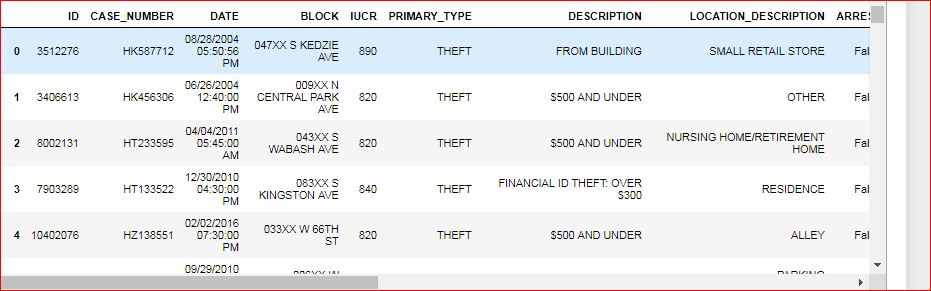
This dataset can be download from the coursera crime data and reflects reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago in the last year, minus the most recent seven days. A full desription of the data is available on the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

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

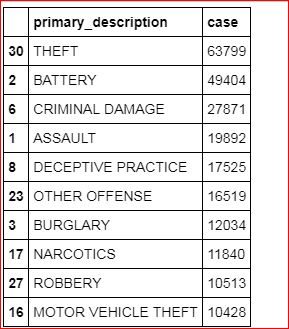
* Date of Occurance
* Block
* Primary Description
* Ward
* Latitude
* Longitude

A sample of the imported data is shown.



This data was then processed as follows:

1. Move September 2017 dates to September 2018 The extract of data used was taken mid September which meant that there was half a months data for September 2017 and half a months data for september 2018. These were combined to create a single month.
2. 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
3. Change the date of occurance field to a date / time object
4. Add new columns for:
   1. Hour
   2. Day
   3. Month
   4. Year
   5. etc.
5. Split Block into zip\_code and street
6. Verify that all rows have valid data



**Data Analysis and Visualisation**

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

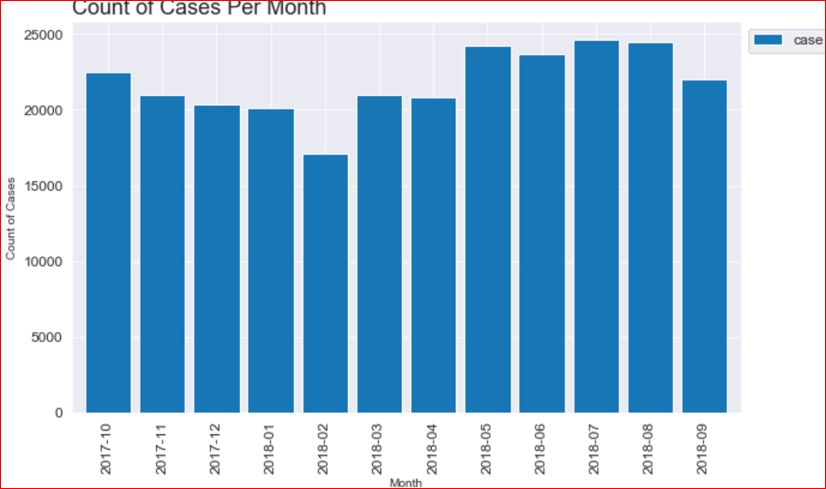
We will start by looking at the top three crimes and a total count for each crime type:

# What Crimes are the 3 most commonly occuring ones

df[['primary\_description', 'case']].groupby(

['primary\_description'], as\_index=False).count().sort\_values(

'case', ascending=False).head(3)



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

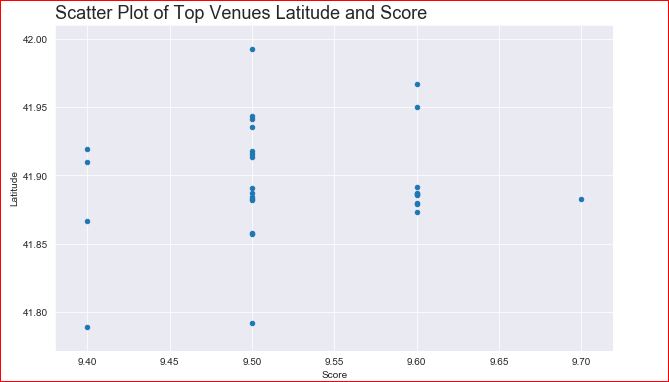
**Section 3: Methodology**

Methodology section which represents the main component of the report where you discuss and describe any exploratory data analysis that you did, any inferential statistical testing that you performed, and what machine learnings were used and why.

**Exploratory Data Analysis**

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

Unforfunately the only data attributes that could be analysed were the Latitude and Longitude attributes and their relationship to the venuse score. Top Venues was examined First.



Although nothing obvious to would appear that the top venues are centered arounf the -87.65 Longitude.

the Restaurant data was examined next.

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

**Further Visualisation**

Because it was not possible, because of the categorical nature of the data, to do more details inferential statistical analysis of the data further exploratory visualisation was undertaken. It shouldbe noted, however, that this visualisation would actually become part of the final presentation to the traveller. It would be important for the traveller to see the crime, venue and restaurant data presented in this manner.

Display each of the Top 10 Venues

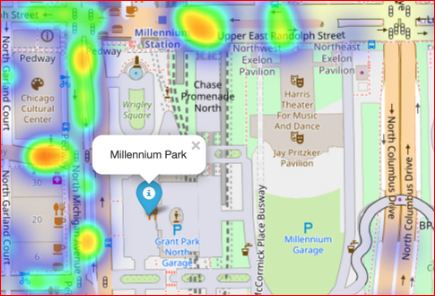
In this section a preview of the type of data that will be displayed to a user of the proposed solution is shown.

For each of the Top 10 Venues:

1. All crimes within 750 meters of the venue are added to a dataframe
2. All restaurants associated with the venue are added to a dataframe
3. A folium Map is created centered on the venue
4. A heatmap of the crimes in the area are overlayed
5. the venue is marked on the map
6. The top 10 scored restaurants are marked on the map

It is possible to fully automate this through full iteration but in order to clearly show each of the 10 maps each is generated manually (to a degree).

The generated map is shown below.



The first map below is the top rated venue *Millennium Park*. The location of the attraction and the 10 top rated venues are clearly shown. The Top Venue is shown using a blue marker, the restaurants are shown using a red marker. Also shown is the heatmap of cimes within 750 meters over the course of the entire previous year. The hotter, redder, the heatmap the more crimes there are recorded. Some Restaurants, for example the two located at the top left of the map, appear to be in areas where crime is quite frequent. On the other hand others are in areas which are obviously not as crime ridden.

Visiting this venue appears to be a much safer option with very little crime recored in the immediate vicinity. Also shown in the map above is the extra details provided about each Restaurant. The restaurant name, *Tango Sur*, it's food type *Argentinian*, and its average score are given.

**Modelling**

Before we start modelling we need to prepare the data frame to include only mumerical data and by removing unneeded columns.

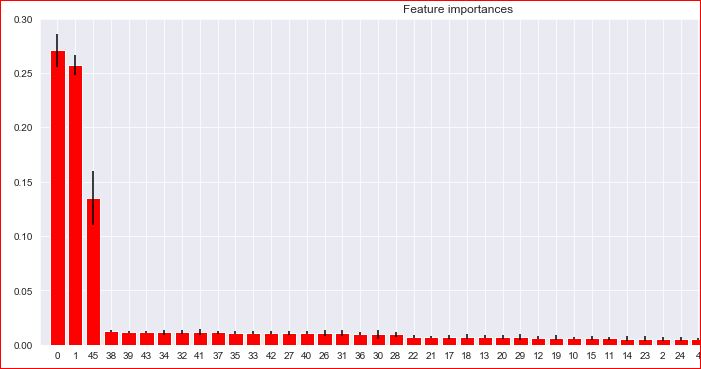
Rather than removing colums from df a new df\_features DataFrame was created with just the required columns. This df\_features DataFrame was then processed to remove Categorical Data Types and replace them with One Hot encoding. Finally the Dependant Variables were normalised

**What are the important Features**

The most important, or informative, features are:

1. Ward
2. Latitude
3. Longitude

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



**Results & Prediction**

Let's review the goals of this project.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 traveller with a list of attractions to visit supplementd with a graphics showing the occurance of crime in the region of the venue.

A high level approach is as follows:

*  The travellers decides 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 grographical data
*  Using this additional geographical data the top nearby restaurents are selects
*  The historical crime within a predetermined distance of all venues are obtained
*  A map is presented to the to the traveller 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 DataFrame by spliting the dataset into two distinct balanced sets and randonly assigning to 0 to represent no crime and 1 to present a crime happening.

**Test Data**

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

1. The two dataframes were joined together to form a single dataframe. The venue or restaurant name and the latitude and longitude attributes were added.
2. Duplicate entries were dropped as some restaurants appeared multiple times in the dataframe
3. Next a random date and time was assigned to each venue.
4. The date was then split into Hour, Day of Week, Month and Year as described above
5. 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.
6. y^ (y\_df) or the predictions were then made

**Visualisation of Predictions**

Of the top ten venues 8 were identified as potentially dangerous to visit and 2 were 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 4 venues:

1. Millennium Park 41.882699 -87.623644
2. The Chicago Theatre 41.885578 -87.627286
3. Grant Park 41.873407 -87.620747
4. Nature Boardwalk 41.918102 -87.633283

The Distance Dataframe is recreated again but this time all crimes are included.

The image of Millennium Park and of The Chicago Theatre. Both of these venues were identified as likely to be susceptible to crime.

**Conclusions and Discussions**

Although all of the goals of this project were met there is definitely 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 devleoped into a fully phledged application that could support the cautious traveller 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.

**Further Development**

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

1. Best time to visit each venue
2. Suggestions for morning, afternoon, evening and night time
3. Daily itineraries
4. Route planning and transportation
5. Time lapse of the crime in the area of the venue
6. Favourite dining preferences could be used to choose the restaurants