Chicago

Using crime data to mark safe places for people to visit

A.INTRODUCTION

1.Background:

Chicago officially the City of Chicago, is the most populous city in Illinois, as well as the third most populous city in the United States. With an estimated population of 2,716,450 (2017), it is the most populous city in the Midwest. Chicago is the principal city of the Chicago metropolitan area, often referred to as Chicagoland, the county seat of Cook County, the second most populous county in the United States. The metropolitan area, at nearly 10 million people, is the third-largest in the United States.

There are miles of beautiful Chicago beaches, Chicago museums that rank among the world's best and the friendliest city-dwellers out there. With so much to see and do,it can be tough for visitors to decide which Chicago attractions are really worth experiencing. Whether we are an out-of-towner or a tried-and-true Chicagoan planning a staycation, Oh, and if we are feeling decadent, we must cap off our day of sightseeing with a meal at one of the best restaurants in Chicago.

2.Business problem:

people usually try to search for restaurants which are *affordable*, near to them and has basic facilities like parking, air conditioning, etc. But usually, people don't consider the factor called safety. Hence a system which recommends safe places to visit would really help people and also help in the reduction of crime in Chicago. Police can use this exploratory data to keep a check of public areas where crime is committed more and help in reducing it.

3.Data:

In order to solve the business problem, we will be using the Chicago crime data of 2012-2017.we will take the location of crime happening and cluster them. Then we will use clusters centroids to identify tourist places and other places around these clusters mark them safe and unsafe accordingly.The final data will serve a few additional features like price tier(affordability), rating, safety(analysis result), etc.
venue categories will be made and safest venue of type near to user will be recommended.

3.1source:

https://www.kaggle.com/currie32/crimes-in-chicago#Chicago_Crimes_2012_to_2017.csv

This dataset reflects reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago from 2001 to present, minus the most recent seven days. 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

3.2 Columns:

ID - Unique identifier for the record.

Case Number - The Chicago Police Department RD Number (Records Division Number), which is unique to the incident.

Date - Date when the incident occurred. this is sometimes a best estimate.

Block - The partially redacted address where the incident occurred, placing it on the same block as the actual address.

IUCR - 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 - The primary description of the IUCR code.

Description - The secondary description of the IUCR code, a subcategory of the primary description.

Location Description - Description of the location where the incident occurred.

Arrest - Indicates whether an arrest was made.

Domestic - Indicates whether the incident was domestic-related as defined by the Illinois Domestic Violence Act.

Beat - 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.

District - Indicates the police district where the incident occurred. See the districts at https://data.cityofchicago.org/d/fthy-xz3r.

Ward - The ward (City Council district) where the incident occurred. See the wards at https://data.cityofchicago.org/d/sp34-6z76.

Community Area - Indicates the community area where the incident occurred. Chicago has 77 community areas. See the community areas at https://data.cityofchicago.org/d/cauq-8yn6.

FBI Code - 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 - 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 - 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.

Year - Year the incident occurred.

Updated On - Date and time the record was last updated.

Latitude - 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 - 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 - The location where the incident occurred in a format that allows for the 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.

3.3 Sample dataset:

Chicago_Crimes_2012_to_2017.csv (89.54 MB) — 20 of 23 columns — Views

	#	ID	Case Number	Date	Block	IUCR
1	3	10508693	HZ250496	05/03/2016 11:40:00 PM	013XX S SAWYER AVE	
2	89	10508695	HZ250409	05/03/2016 09:40:00 PM	061XX S DREXEL AVE	
3	197	10508697	HZ250503	05/03/2016 11:31:00 PM	053XX W CHICAGO AVE	
4	673	10508698	HZ250424	05/03/2016 10:10:00 PM	049XX W FULTON ST	
5	911	10508699	HZ250455	05/03/2016 10:00:00 PM	003XX N LOTUS AVE	
6	1108	10508702	HZ250447	05/03/2016 10:35:00 PM	082XX S MARYLAND AVE	
7	1130	10508703	HZ250489	05/03/2016 10:30:00 PM	027XX S STATE ST	
8	1801	10508704	HZ250514	05/03/2016 09:30:00 PM	002XX E 46TH ST	
9	1868	10508709	HZ250523	05/03/2016 04:00:00 PM	014XX W DEVON AVE	
10	1891	10508982	HZ250667	05/03/2016 10:30:00 PM	069XX S ASHLAND AVE	
11	1935	10508710	HZ250469	05/03/2016 09:44:00 PM	074XX S SOUTH SHORE DR	

Chicago_Crimes_2012_to_2017.csv (89.54 MB) 20 of 23 columns Views

	# Ward	# Community Area	# FBI Code	# X Coordinate	# Y Coordinate	# Year
0	24.0	29.0	08B	1154907.0	1893681.0	
0	20.0	42.0	08B	1183066.0	1864330.0	
0	37.0	25.0	24	1140789.0	1904819.0	
0	28.0	25.0	08B	1143223.0	1901475.0	
0	28.0	25.0	06	1139890.0	1901675.0	
0	8.0	44.0	04B	1183336.0	1850642.0	
0	3.0	35.0	08B	1176730.0	1886544.0	
0	3.0	38.0	08B	1178514.0	1874573.0	
0	40.0	1.0	08B	1165696.0	1942616.0	
0	17.0	67.0	08B	1166876.0	1858796.0	
0	7.0	43.0	15	1195696.0	1856719.0	
0	42.0	8.0	08B	1176630.0	1904401.0	

3.4 Usage of foursquare API:

# Latitude	# Longitude
41.864073157	-87.706818608
41.782921527	-87.60436317
41.894908283	-87.758371958
41.885686845	-87.749515983
41.886297242	-87.761750709
41.745354023	-87.603798903
41.844023772	-87.626923253
41.811133958	-87.62074077
41.99813061	-87.665814038
41.768096835	-87.663878589
41.761733286	-87.558309979
41.893026751	-87.626750829

These will help to access neighbourhood data.

METHODOLOGY:

A.Data collection

- Collected data from chicago crime database it has each of the crime location Type ,time of occurrence .It also has Latitude and longitude.

Primary Type	Description	Location Description	Arrest	...	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	Location
BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	True	...	24.0	29.0	08B	1154907.0	1893681.0	2016	05/10/2016 03:56:50 PM	41.864073	-87.706819	(41.864073157, -87.706818608)
BATTERY	DOMESTIC BATTERY SIMPLE	RESIDENCE	False	...	20.0	42.0	08B	1183066.0	1864330.0	2016	05/10/2016 03:56:50 PM	41.782922	-87.604363	(41.782921527, -87.60436317)
PUBLIC PEACE VIOLATION	RECKLESS CONDUCT	STREET	False	...	37.0	25.0	24	1140789.0	1904819.0	2016	05/10/2016 03:56:50 PM	41.894908	-87.758372	(41.894908283, -87.758371958)
BATTERY	SIMPLE	SIDEWALK	False	...	28.0	25.0	08B	1143223.0	1901475.0	2016	05/10/2016 03:56:50 PM	41.885687	-87.749516	(41.885686845, -87.749515983)
THEFT	$500 AND UNDER	RESIDENCE	False	...	28.0	25.0	06	1139890.0	1901675.0	2016	05/10/2016 03:56:50 PM	41.886297	-87.761751	(41.886297242, -87.761750709)

- **Gathering coordinates:**
  As the above database has we can collect latitude and longitude of each location of crime location

B.Data processing

- For further processing of data we need to do feature modeling and might also need to create new features that might be more correlated to crime happening.
- So first using label encoders we encoded features ‘primary type’ and then ‘arrest ‘ .This helped us by converting categorical values to numerical values
- Then we according to count of occurence of crime in each block created a new Feature ‘count’ which indicates (frequency) occurrence of crime in a Block

```
In [5]: k=df['Block'].value_counts()
        dic=k.to_dict()
        dic
```

```
Out[5]: {'001XX N STATE ST': 3634,
         '0000X W TERMINAL ST': 2746,
         '008XX N MICHIGAN AVE': 2465,
         '076XX S CICERO AVE': 2116,
         '0000X N STATE ST': 1844,
         '064XX S DR MARTIN LUTHER KING JR DR': 1349,
         '083XX S STEWART AVE': 1216,
         '063XX S DR MARTIN LUTHER KING JR DR': 1138,
         '051XX W MADISON ST': 1115,
         '046XX W NORTH AVE': 1113,
         '009XX W BELMONT AVE': 1102,
         '011XX S CANAL ST': 1084,
         '008XX N STATE ST': 1046,
         '040XX W LAKE ST': 1007,
         '100XX W OHARE ST': 965,
         '006XX N MICHIGAN AVE': 953,
         '038XX W ROOSEVELT RD': 925,
         '0000X W HUBBARD ST': 922,
         '033XX W FILLMORE ST': 919,
```

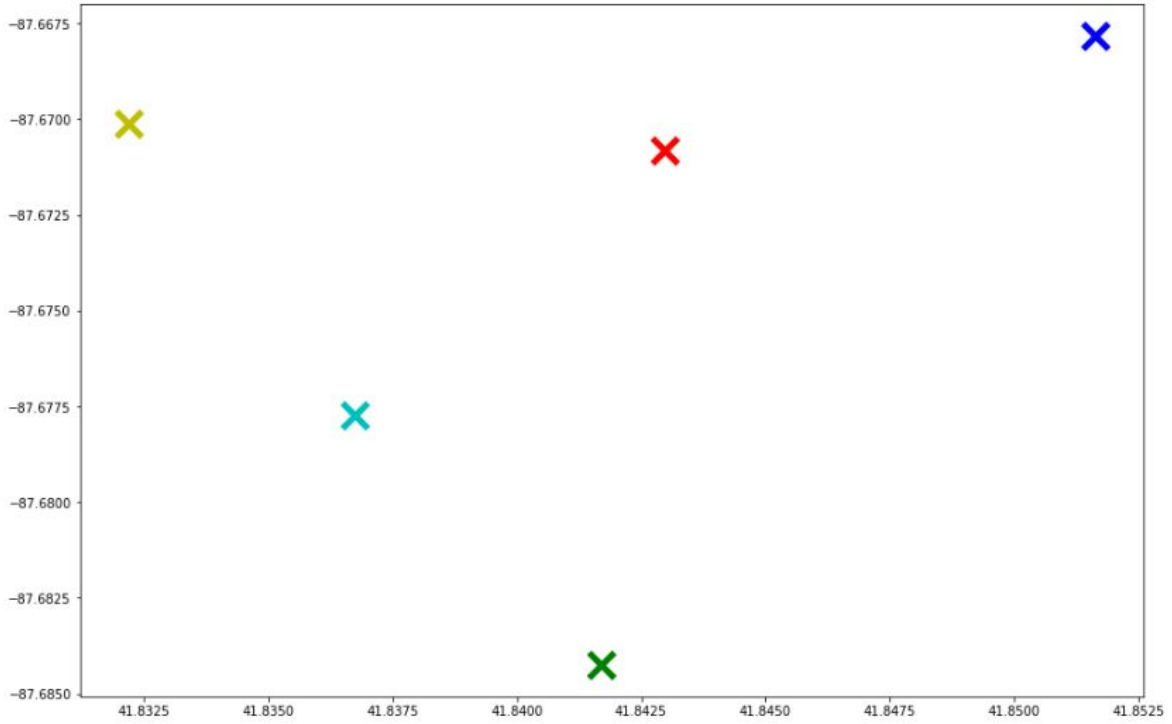
```
In [6]: df['Counts'] = df['Block'].map(dic)
        df.loc[df['Counts']>0,'Frequency']=0
        df.loc[df['Counts']>500,'Frequency']=1
        df.loc[df['Counts']>1000,'Frequency']=2
        df.head()
```

Out[6]:

UCR	Primary Type	Description	Location Description	Arrest	...	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	Location	Counts	Frequency
0486	2	DOMESTIC BATTERY SIMPLE	APARTMENT	1	...	08B	1154907.0	1893681.0	2016	05/10/2016 03:56:50 PM	41.864073	-87.706819	(41.864073157, -87.706818608)	132	0.0
0486	2	DOMESTIC BATTERY SIMPLE	RESIDENCE	0	...	08B	1183066.0	1864330.0	2016	05/10/2016 03:56:50 PM	41.782922	-87.604363	(41.782921527, -87.60436317)	97	0.0
0470	27	RECKLESS CONDUCT	STREET	0	...	24	1140789.0	1904819.0	2016	05/10/2016 03:56:50 PM	41.894908	-87.758372	(41.894908283, -87.758371958)	370	0.0
0460	2	SIMPLE	SIDEWALK	0	...	08B	1143223.0	1901475.0	2016	05/10/2016 03:56:50 PM	41.885687	-87.749516	(41.885686845, -87.749515983)	115	0.0
0820	31	$500 AND UNDER	RESIDENCE	0	...	06	1139890.0	1901675.0	2016	05/10/2016 03:56:50 PM	41.886297	-87.761751	(41.886297242, -87.761750709)	125	0.0

C.Clustering of data(Using K Means)

- Then using k means we will cluster data into 5 clusters and obtain their centroids



- Importance of centroids is that around these we will find all venues and then the venues occuring in overlapping regions of these clusters will be marked more unsafe the ranging from there.

```
In [9]: from sklearn.cluster import KMeans
        kclus=KMeans(init="k-means++",n_clusters=5).fit(x)
        location=kclus.cluster_centers_
        location
```

D.Foursquare API use

- Now taking cluster centroid we find all the venues around those centroids and add all of them to a single dataframe.

```
In [19]: for i in range(0,5):
             latitude=coordinates[i][0]
             longitude=coordinates[i][1]
             radius=1000
             url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format(CL
             results = requests.get(url).json()
             venues=results['response']['venues']
             dataframe1 = json_normalize(venues)
             dataframe=dataframe.append(dataframe1)
             print(dataframe1.shape)

C:\Users\vivek\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\frame.py:6692: FutureWarning: Sorting because no
n-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  sort=sort)

(141, 18)
(150, 24)
(132, 25)
(150, 25)
```

The image shows no. of responses added to dataframe in each iteration.

- Then we will count occurence of each record in our data frame and make their occurrence count as a column and drop all duplicates then of the dataframe

```
In [47]: z1=dataframe.pivot_table(index=['id'], aggfunc='size').to_dict()

In [41]: dataframe.pivot_table(index=['id'], aggfunc='size').value_counts()

Out[41]: 1    435
         2     99
         3     11
         dtype: int64

In [49]: dataframe['count']=dataframe['id'].map(z1)

In [50]: dataframe.head()

Out[50]:
```

...	location.labeledLatLngs	location.lat	location.lng	location.neighborhood	location.postalCode	location.state	name	referralId	venuePage.id	count
...	[{'label': 'display', 'lat': 41.84610154558017...	41.846102	-87.670743	NaN	60608	IL	Sims Metal Management-Midwest	v-1557904821	NaN	2
...	NaN	41.856953	-87.662681	NaN	NaN	IL	Pilsen	v-1557904821	NaN	1
...	[{'label': 'display', 'lat': 41.851776, 'lng':...	41.851776	-87.668513	NaN	60608	IL	Spa Nordstom	v-1557904821	NaN	1
...	[{'label': 'display', 'lat': 41.8517, 'lng': -...	41.851700	-87.667830	NaN	USA	IL	GameStop	v-1557904821	NaN	1
...	[{'label': 'display', 'lat': 41.85144498508381...	41.851445	-87.666309	NaN	60608	IL	Manjares Restaurant	v-1557904821	NaN	1

- Our aim to check whether venue is safe or not and we obtained one parameter that is count to obtain other parameters we use foursquare api to get details of each venue and extract features like restaurant is verified or not,likes ,tips etc.

```
In [57]: for i in k:
             url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(i, CLIENT_ID, CLIENT_SECRET, VERSI
             result=requests.get(url).json()

             try:
                 li_verified.append(result['response']['venue']['verified'])
             except:
                 li_verified.append("unknown")
                 pass


             try:
                 li_likes.append(result['response']['venue']['likes']['count'])
             except:
                 li_likes.append(np.nan)
                 pass

             try:
                 li_tips.append(result['response']['venue']['tips']['count'])
             except:
                 li_tips.append(np.nan)
                 pass

             try:
                 li=[]
                 types=result['response']['venue']['listed']['groups']
                 for i in types:
                     li.append(i['type'])
                 li_types.append(li)
             except:
                 li_types.append(np.nan)
                 pass
```

- Now we have obtained almost all data we need to model and predict whether a venues in chicago is safe,super safe not that safe

F.Model building

- Before model building we preprocess our data and encode ‘has perk’,’verified’ and ‘category’
- Then we use logistic regression to model our data.

```
In [143]: from sklearn.linear_model import LogisticRegression
          from sklearn.model_selection import train_test_split
          x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=20)
          lm=LogisticRegression()
          lm.fit(x_train,y_train)
```

This model will help us to predict whether or not a venues is safe and also give probability of safety in that venue.

CONCLUSION

[1]

In summation using K means to cluster data and Logistic regression model we added a extra tag to each venue that is range of safety which is a major concern nowadays

In addition we can also identify whether the venue we are going to is safe or not and also now probability that our prediction is true.

This is help us to build a safer place for people.venues with less safety can take up measures to prevent crimes and also help people to be safe from such places.

[1] VIVEK GOPALSHETTY