Analise explanatória da Criminalidade Chicago

[1]Alvaro Cristian da Silva Botelho

[1]MBA Data Science e Big Data – Universidade do Vale dos Sinos (UNISINOS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

Cristian.ucpel@gmail.com

INTRODUÇÃO

O presente documento visa relatar os experimentos realizados no conjunto de dados sazonais obtidos através da plataforma Kaggle https://www.kaggle.com/currie32/crimes-in-chicago). A base contem 7.941.282 Linhas e 23 atributos. Durante o processo foi utilizado o algoritmo de forecast FBProphet (https://facebook.github.io/prophet/). Prophet é um algoritmo para forecasting time series com base em modelos aditivos no qual as tendências não lineares se ajustam a sazonalidade, também funciona com dados faltantes e tem uma qualidade alta quanto a outliers. Mapas e os scripts foram anexados juntos deste documento. Mais informações estão no https://github.com/alcristian/MachineLearningTask10-23-2018

ANALISE EXPLANATORIA

Alguns dados estatísticos sobre a massa de dados.

```
In [6]: df_crimedata.describe()
```

Out[6]:

	Unnamed: 0	ID	Beat	District	Ward	Community Area	X Coordinate	Year	Longitude
count	7.941282e+06	7.941282e+06	7.941282e+06	7.941191e+06	7.241058e+06	7.239191e+06	7.835709e+06	7.941282e+06	7.835708e+06
mean	2.673858e+06	5.926071e+06	1.197659e+03	1.131215e+01	2.262089e+01	3.774790e+01	1.164455e+06	2.007672e+03	-8.767203e+01
std	1.816327e+06	2.568290e+06	7.041944e+02	6.944523e+00	1.378632e+01	2.156597e+01	1.751911e+04	4.123451e+00	6.328715e-02
min	0.000000e+00	6.340000e+02	1.110000e+02	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	4.178983e+01	-9.168657e+01
25%	1.160283e+06	3.853209e+06	6.230000e+02	6.000000e+00	1.000000e+01	2.300000e+01	1.152887e+06	2.005000e+03	-8.771401e+01
50%	2.282372e+06	6.165079e+06	1.111000e+03	1.000000e+01	2.200000e+01	3.200000e+01	1.165910e+06	2.008000e+03	-8.766643e+01
75%	4.185491e+06	7.716590e+06	1.732000e+03	1.700000e+01	3.400000e+01	5.800000e+01	1.176336e+06	2.010000e+03	-8.762856e+01
max	6.254267e+06	1.082788e+07	2.535000e+03	3.100000e+01	5.000000e+01	7.700000e+01	1.205119e+06	2.017000e+03	-8.752453e+01

Após analise exploratória nos dados foi verificado uma grande quantidade de dados faltantes em algumas colunas, as quais foram removidas .

```
In [8]: #percentual de dados faltantes
missing_data_per = df_crimedata.isna().sum()*100/len(df_crimedata)
missing_data_per
#existem dados faltantes nas colunas Case Number, Location Description, District, Ward, Community Area, X Coordinate, y Coordinate
```

Out[8]:

```
Unnamed: 0              0.000000
ID                      0.000000
Case Number             0.000065
Date                    0.000000
Block                   0.000000
IUCR                    0.000000
Primary Type            0.000000
Description             0.000000
Location Description    0.031892
Arrest                  0.000000
Domestic                0.000000
Beat                    0.000000
District                0.000794
Ward                    9.964070
Community Area          9.983143
FBI Code                0.000000
X Coordinate            1.364861
Y Coordinate            1.364861
Year                    0.000000
Updated On              0.000000
Latitude                1.364861
Longitude               1.364861
Location                1.364861
dtype: float64
```

```
In [9]: #Remocao das das colunas com a maior porcentagem de dados faltantes.
df_crimedata = df_crimedata.drop(columns=['Unnamed: 0', 'Case Number', 'Block', 'IUCR', 'Location Description',
                                          'Domestic', 'Updated On', 'FBI Code', 'X Coordinate', 'Y Coordinate',
                                          'Latitude', 'Longitude', 'Location'], axis = 1)

df_crimedata.head()
```

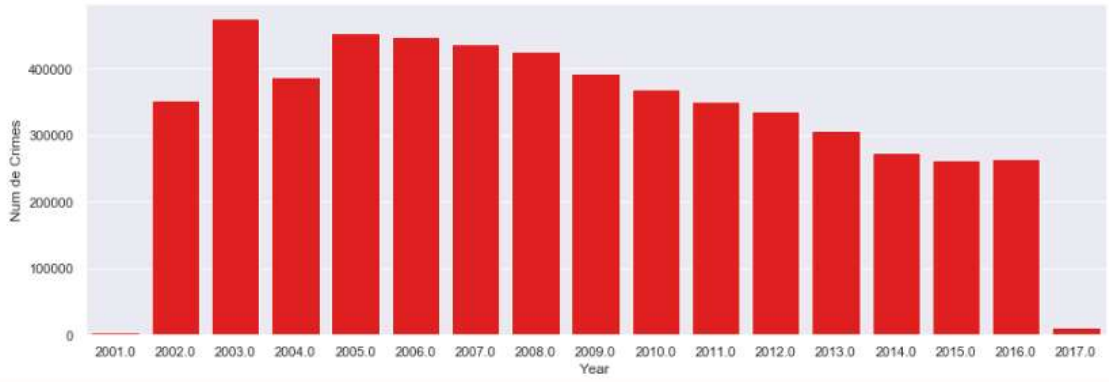
Após a remoção das linhas faltantes foi removida também as linhas com valores nulos, mesmo sabendo que o algoritmo trabalha bem com valores nulos.

```
In [11]: #deletar as linhas que contem null
df_crimedata = df_crimedata.dropna()
df_crimedata.isnull().sum()

Out[11]: ID              0
         Date            0
         Primary Type    0
         Description     0
         Arrest          0
         Beat            0
         District        0
         Ward            0
         Community Area  0
         Year            0
         dtype: int64
```

```
In [12]: df_crimedata.shape

Out[12]: (5554662, 10)
```

Após todas as limpezas sobraram 5.554.662 linhas e 10 atributos.

```
In [13]: sns.countplot(x='Year',data=df_crimedata, color=('RED'))
fig = plt.gcf()
plt.ylabel('Num de Crimes')
fig.set_size_inches(15,5)

plt.show()
#numero total de crimes de Chicago vem diminuindo durante os anos.
```

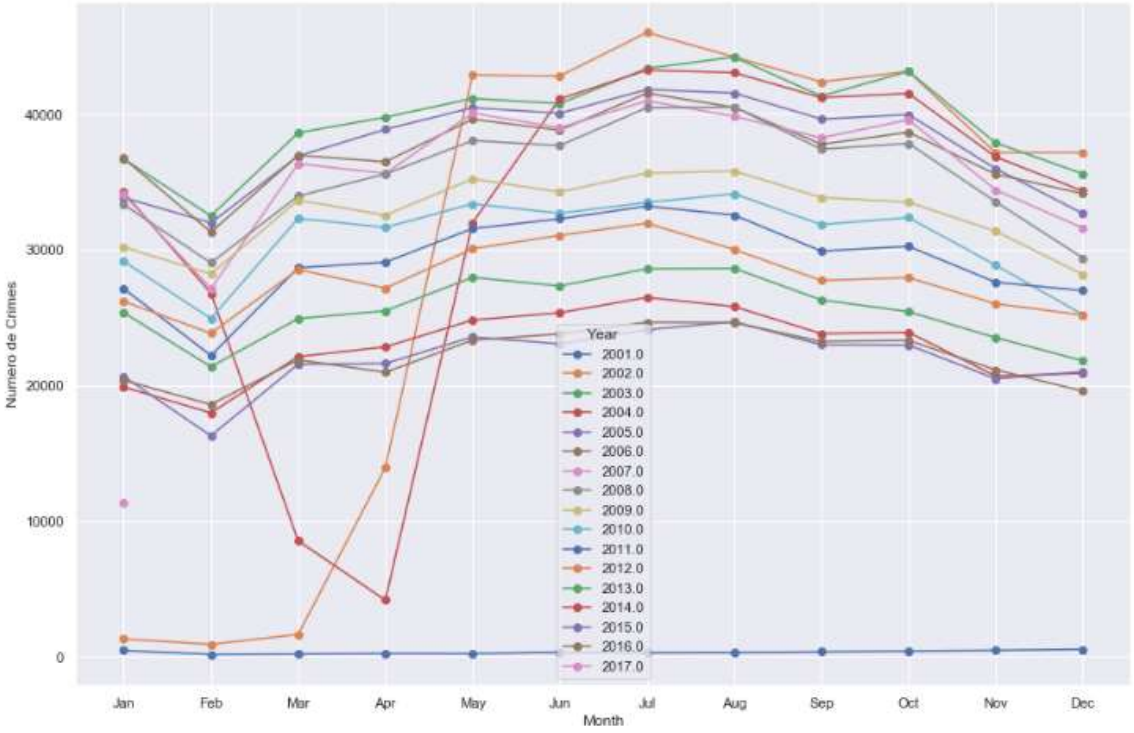


Verificando os dados históricos de Chicago a criminalidade geral vem diminuindo durante os anos. Os dados de 2017 não contemplam todo o ano por esse motivo está com um percentual muito baixo.

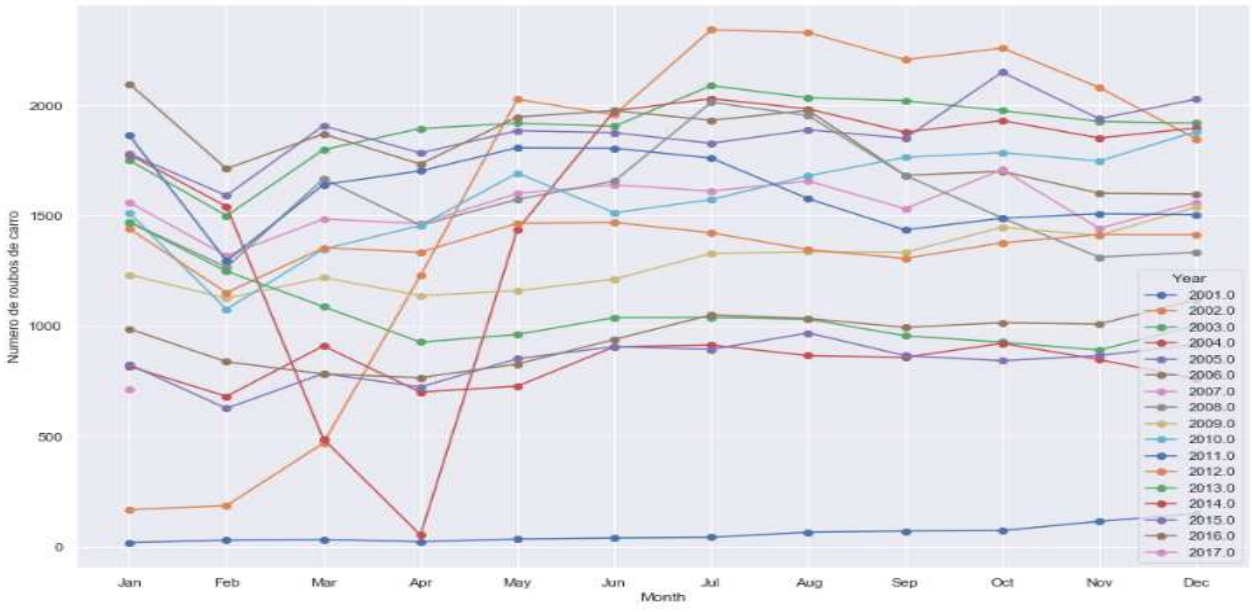
Separados os dados de forma mensal podemos ver que em todos os anos há um aumento da criminalidade nos meses de julho. O pico da criminalidade em Chicago em todos os anos quase sempre é no mesmo mês.

```
In [21]: #Ajustando as datas
df_crimedata['Date'] = pd.to_datetime(df_crimedata['Date'],format='%m/%d/%Y %I:%M:%S %p')
df_crimedata['Month']=(df_crimedata['Date'].dt.month).apply(lambda x: calendar.month_abbr[x])
df_crimedata['Month'] = pd.Categorical(df_crimedata['Month'] , categories=['Jan','Feb','Mar','Apr','May',
                                        'Jun','Jul','Aug','Sep','Oct','Nov','Dec'], ordered=True)
months=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']
```

```
In [22]: df_crimedata.groupby(['Month','Year'])['ID'].count().unstack().plot(marker='o', figsize=(15,10))
plt.xticks(np.arange(12),months)
plt.ylabel('Numero de Crimes')

plt.show()
```



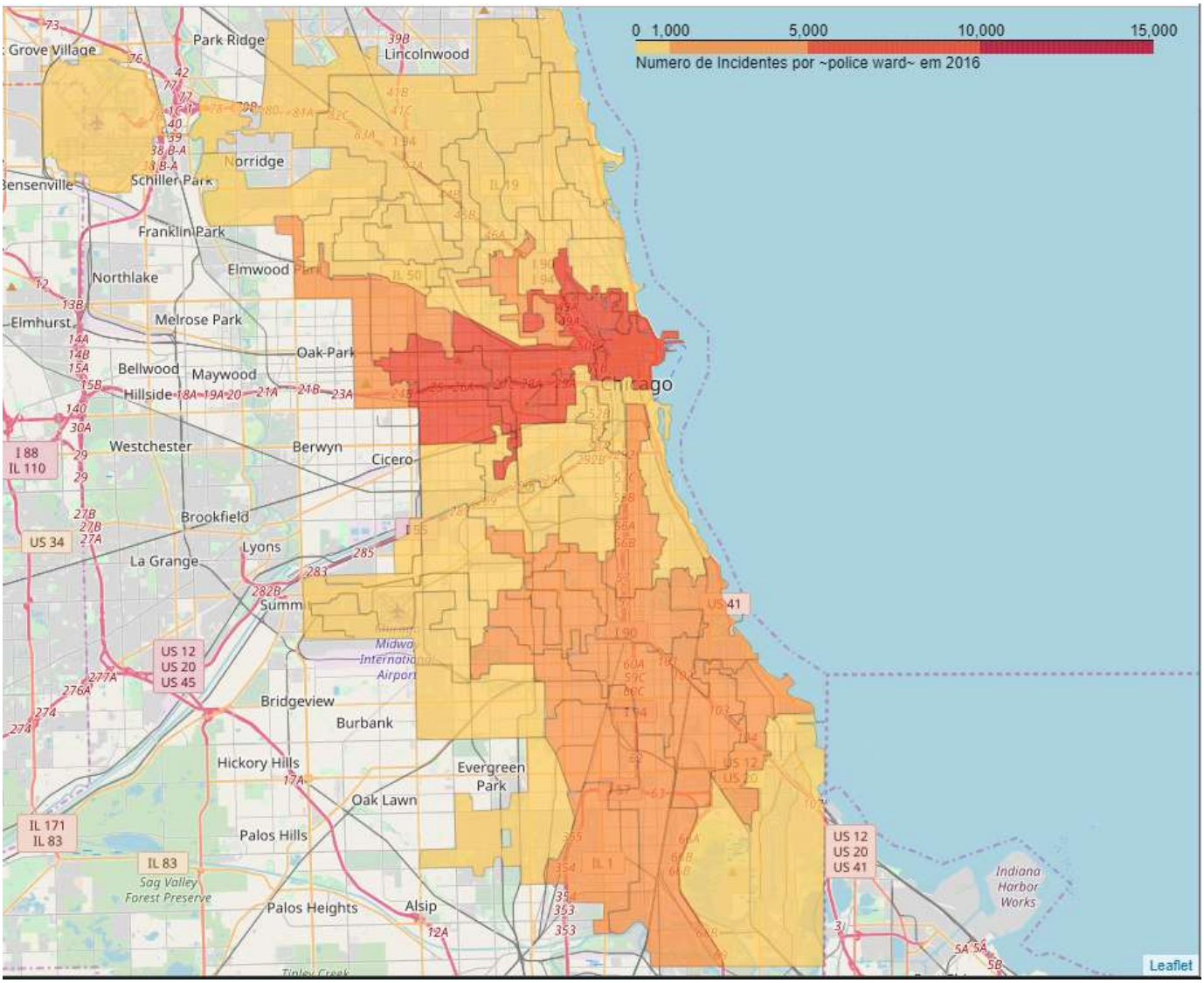
Isolando os dados referentes ao roubo de carros encontramos alguns outliers nos anos de 2012 e 2014, mas como observado no gráfico acima o pico da criminalidade não parece ser nos mesmo meses, e ela aumenta ao decorrer dos meses.



In [16]:

```
#Mapa Crimes Chicago by Ward
#vou utilizar os dados de 2016 porque Chicago implementou o sistema de Boundries em 2015 entao pode haver incoerencias.

#https://www.cityofchicago.org/city/en/depts/doit/dataset/boundaries_-_wards.html

geo_crimemap = 'Boundaries - Wards (2015-).geojson'
#calcular o numero de incidentes por distrito em 2016
df_WardCrimes2016 = pd.DataFrame(df_crimedata2016['Ward'].value_counts().astype(float))


df_WardCrimes2016 = df_WardCrimes2016.reset_index()
df_WardCrimes2016.columns = ['ward', 'Qtd_Crimes']
df_WardCrimes2016.to_json('Ward_Map.json')

Chicago_COORDINATES = (41.895140898, -87.624255632)

chicago_map1 = folium.Map(location=Chicago_COORDINATES, zoom_start=11)
chicago_map1.choropleth(geo_data = geo_crimemap,
                 data = df_WardCrimes2016,
                 columns = ['ward', 'Qtd_Crimes'],
                 key_on = 'feature.properties.ward',
                 fill_color = 'YlOrRd',
                 fill_opacity = 0.7,
                 line_opacity = 0.2,
                 threshold_scale=[0, 1000, 5000, 10000, 15000],
                 legend_name = 'Numero de Incidentes por ~police ward~ em 2016')

chicago_map1.save('map_chicago_by_Ward.html')
```

Criação de mapa para visualizar a quantidade de crimes por região policial em Chicago, foi utilizado nesse gráfico somente os dados do ano de 2016 porque a Policia de Chicago só implementou o sistema de Boundries. Existe uma região onde o crime eh concentrado com uma divergência bem maior que as outras áreas de Chicago.



In [18]:
```
#Mapa de roubo de carros Chicago by Ward /MOTOR VEHICLE THEFT/
#vou utilizar os dados de 2016 porque Chicago implementou o sistema de Boundries em 2015 entao pode haver incoerencias.

#https://www.cityofchicago.org/city/en/depts/doit/dataset/boundaries_-_wards.html

geo_crimemap2 = 'Boundaries - Wards (2015-).geojson'
#calcular o numero de incidentes por distrito em 2016
df_WardCarTheft2016 = pd.DataFrame(df_crimedatacartheft2016['Ward'].value_counts().astype(float))

df_WardCarTheft2016 = df_WardCarTheft2016.reset_index()
df_WardCarTheft2016.columns = ['ward', 'Qtd_Crimes']
df_WardCarTheft2016.to_json('Ward_Map.json')

Chicago_COORDINATES = (41.895140898, -87.624255632)

chicago_map2 = folium.Map(location=Chicago_COORDINATES, zoom_start=11)
chicago_map2.choropleth(geo_data = geo_crimemap2,
                data = df_WardCarTheft2016,
                columns = ['ward', 'Qtd_Crimes'],
                key_on = 'feature.properties.ward',
                fill_color = 'YlOrRd',
                fill_opacity = 0.7,
                line_opacity = 0.2,
                threshold_scale=[0, 100, 200, 400, 600],
                legend_name = 'Numero de Roubos de carro por ~police ward~ em 2016')

chicago_map2.save('map_car_theft_by_Ward.html')
```

O mapa dos roubos de carro mostra uma figura semelhante da distribuição de crimes geral. Porem esta concentrado em 2 areas especificas.