Relatório do projeto Safe Neighborhood

Flávia Nitto
Miguel Donizeti
Vitor Pelicer de Mesquita França

24 de janeiro de 2022

Resumo

Análise da segurança de acordo com as subprefeituras da cidade de São Paulo.

1 Introdução

Com base em dados oficiais da Secretaria de Segurança Pública de São Paulo (SSP-SP), e da Companhia de Engenharia de Trafego (CET), foram coletados dados sobre o acidentes de trânsito com vítimas, homicídios, estupros e roubos. O período analisado é do intervalo dos anos de 2012 e 2020. Com base nesses dados, é calculado uma previsão futura, na qual é estipulado uma distribuição da probabilidade dos possíveis valores de acordo com a distribuição normal.

2 Justificativa

Levando em consideração a área de interesse da pesquisa social, e a notória preocupação por parte civil, empresarial e governamental em analisar e monitorar o grau de segurança dos locais, em específico da cidade de São Paulo. Dessa forma, a aplicação que este trabalho propõe, "Safe Neighborhood", foi pensada para a utilização dos dados do site da Secretaria de Segurança Pública (SSP-SP) e da Companhia de Engenharia de Trânsito (CET) e traçar gráficos que serão de fácil entendimento quando comparados com a base de dados brutas. Assim, criar o comparativo de segurança entre as regiões das subprefeituras de São Paulo é o enfoque deste trabalho.

3 Objetivo

O objetivo do projeto é criar um parâmetro de comparação entre as localidades com foco em explicitar e quantificar o registro de violência local e a sua projeção para o próximo ano. Analisar cada subprefeitura individualmente com base em dados quantitativos oficiais divulgados pela Segurança Pública de São Paulo (SSP-SP), e pela Companhia de Engenharia de Trafego (CET). Sendo os índices levados em conta são acidentes de trânsito com vítimas, homicídios, estupros e roubos. Demonstrar os dados em formato de gráfico, criando maior facilidade para o usuário final tirar sua conclusão.

4 Fundamentação Teórica

4.1 Séries temporais

[1] A série temporal é uma coleção de observações organizadas de maneira ordenada, é mais comum ser medida em tempo, onde existe a interdependência entre os dados passados com os dados futuros, sendo possível modelar essa dependência.

4.2 ARIMA

[2] [3] [0] Este é um modelo de regressão para gerar previsões e aproximações. ARIMA é uma sigla que significa "Auto Regressive Integrated Moving Average", em português significa "modelo de média móvel integrado autorregressivo", este modelo analisa a série temporal com base em três parâmetros (p,q,d), p representa a ordem de auto regressão, q representa a ordem da média móvel e d é o número de diferenciações necessárias para tornar a série temporal estacionária.

4.3 Distribuição normal

Este modelo é uma distribuição, onde a maior probabilidade se concentra na média, é representado por μ. O desvio padrão representa a dispersão da probabilidade, é representado por σ.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

Esta formula gera a distribuição da imagem 1.

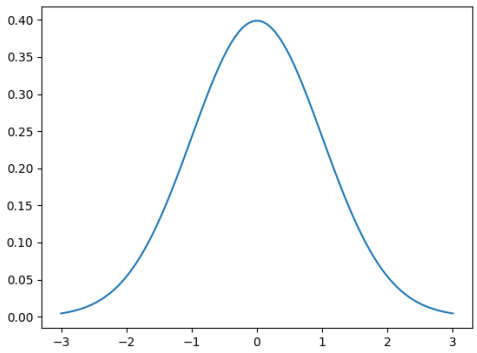

Figura 1: Exemplo de curva normal com $\mu = 0$ e $\sigma = 1$

Z-score é a medida do quantas vezes o desvio padrão as probabilidades se afastam de μ. O Z-score é relacionado com o preenchimento da probabilidade no gráfico.

$$\sigma = \frac{x - \mu}{z} \tag{2}$$

5 Metodologia

A linguagem de programação utilizada para os cálculos e gerar os gráficos é a Python, os dados estão contidos em arquivos do tipo .csv e são recebidos pelo script e tratados como séries temporais, a análise do modelo ARIMA é feito pela função "auto_arima()"da biblioteca "pmdarima". Está função realiza testes consecutivos e ajusta a melhor combinação para os parâmetros (p,q,d). Após o calculo do modelo, é realizado a previsão com o método ".predict()". Para este método são passados como parâmetros além do modelo calculado "return_conf_int=True, alpha=0.05"que significa que o método retorna o intervalo de confiança, sendo definido por alfa porcentagem de erro. Logo, segundo a biblioteca, a acurácia é definida por 1-alpha, o que da uma confiabilidade de 95%. A margem de confiabilidade é utilizada como parâmetro para uma aproximação da curva normal, visto que μ é o valor previsto e os dois valores da faixa de confiança se distanciam de maneira equivalente em relação de μ, logo é intuitivo a semelhança com a distribuição normal pela simetria. O desvio padrão é calculado considerando um Z score de 1.96, pois se aproxima bem para o modelo com 95% da distribuição na curva normal.

Após definir o desvio padrão, é utilizada a função "stats.norm.pdf()", tendo como argumentos "x, mu, sigma", sendo x o intervalo linear utilizado como eixo das abscissas no gráfico.

Após a criação dos gráficos, as imagens são exportadas como arquivos .png e são visualizadas utilizando JavaScript, html e css.

O código que contém os conceitos de probabilidade envolvidos:

```
import pandas as pd
import matplotlib.pyplot as plt
import pmdarima as pm
import numpy as np
import scipy.stats as stats
import warnings
warnings.filterwarnings("ignore")

n_predict=1

#leitura do CSV
col_list = ['Ano', 'Acidentes', 'Homicidios', 'Estupros', 'Roubos']
data = pd.read_csv("subprefeitura.csv", usecols=col_list)

#tratamento dos dados
#separa os dados e anos em listas distintas
ano = data['Ano']
acidente = data['Acidentes']
homicidio = data['Homicidios']
roubo = data['Roubos']
estupro = data['Estupros']
ano = ano.to_numpy()
acidente = acidente.to_numpy()
homicidio = homicidio.to_numpy()
roubo = roubo.to_numpy()
estupro = estupro.to_numpy()
shape = len(ano)

#cria a lista com as datas futuras, para casar com a previs o
```

```
ult_data = ano[-1]
anos_fut = list(range(2021, 2021+n_predict))
anos_fut.insert(0, ano[-1])

ls = [acidente, homicidio, roubo, estupro]
lsword = ["acidente", "homicidio", "roubo", "estupro"]

for num in range(0, 4):
    #cria o ARIMA
    info = ls[num]
    word = lsword[num]
    model = pm.auto_arima(info, seasonal=False)
    #cria a previs o
    forecasts, erro = model.predict(n_predict, return_conf_int=True, alpha=0.05)
    forecasts = forecasts.tolist()

    #plota as margens de confian a nos graficos de distribui  o normal
    for n in range(n_predict):
        mu = forecasts[n]
        sigma = (erro[n][1] - forecasts[n])/1.96
        variance = pow(sigma, 2)
        x = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
        plt.plot(x, stats.norm.pdf(x, mu, sigma), label="Distribui  o da probabilid
        plt.legend()
        plt.show()

    #coloca o ultimo ponto como o primeiro na previsao
    #junta no grafico os pontos, previstos com os dados
    print(erro)
    forecasts.insert(0, info[-1])
    #configura o plot
    x = np.arange(shape+n_predict)
    plt.plot(ano, info, c='blue', label=word)
    print(anos_fut)
    print(forecasts)
    print()
    plt.plot(anos_fut, forecasts, c='green', label='previs o')
    plt.legend()
    plt.show()
```

6 Resultados

Após rodar o script para os dados da subprefeitura de Aricanduva/V. Formosa, obtém-se os seguintes resultados de previsão segundo as tabelas2.

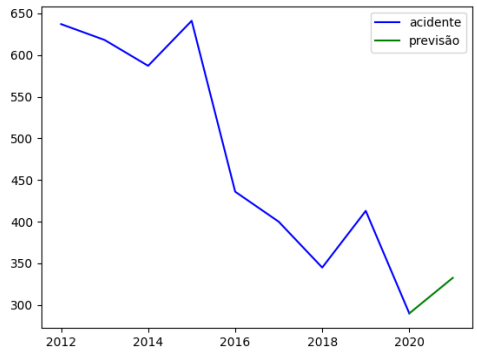

Figura 2: Número de acidentes com vítimas de acordo com cada ano

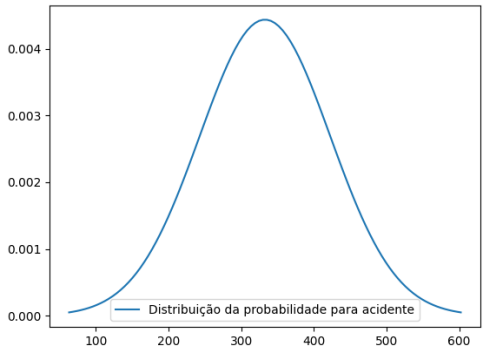

Figura 3: Distribuição da probabilidade de acidentes com vítimas para o ano de 2021

3
4
5
6
7
8
9

Com esse método é possível comparar as subprefeituras pondo em vista estes quatro dados, com perspectiva para o próximo ano.

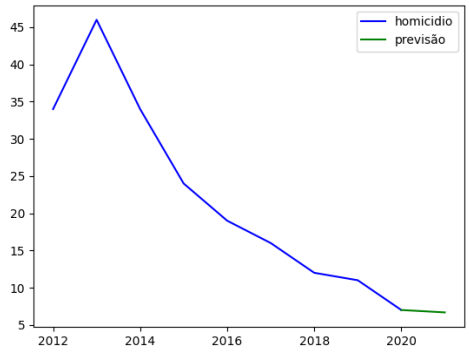

Figura 4: Número de homicídios de acordo com cada ano

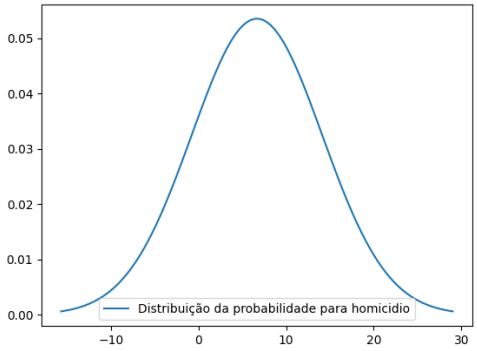

Figura 5: Distribuição da probabilidade de homicídios para o ano de 2021

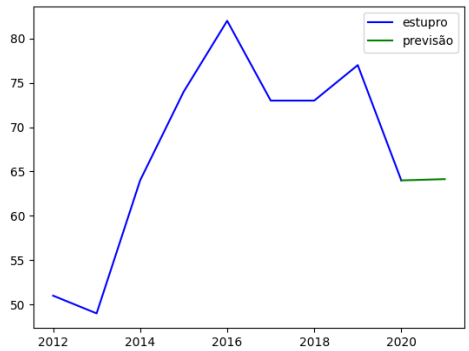

Figura 6: Número de estupros de acordo com cada ano

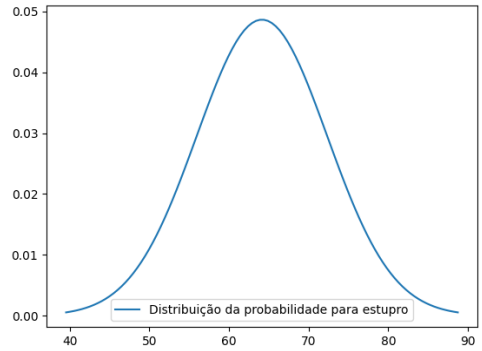

Figura 7: Distribuição da probabilidade de estupros para o ano de 2021

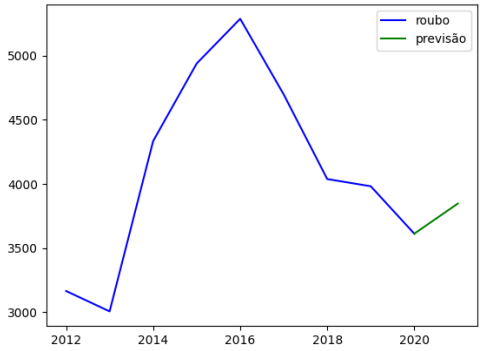

Figura 8: Número de roubos de acordo com cada ano

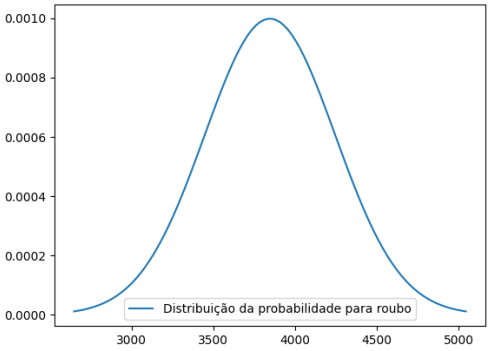

Figura 9: Distribuição da probabilidade de roubos para o ano de 2021

7 Discussão

Como a CET não divulgou os dados relativos ao ano de 2021, não foi possível realizar a análise utilizando este ano. Por outro lado, a Secretaria de Segurança Pública de São Paulo divulgou os dados para 2021, assim, podemos comparar os dados previstos com os dados reais. Neste relatório continuaremos utilizando os dados da subprefeitura de Aricanduva/V. Formosa. Em 2021, na região desta subprefeitura houve a quantidade de delitos descritos na tabela 1:

	Homicídios	Estupros	Roubos
2021	8	69	3468

Tabela 1: Resultados reais de 2021

	Homicídios	Estupros	Roubos
2021	6,67	64,14	3846,79
Margem de confiança	[0 - 21,27]	[48,07 - 80,21]	[3064,14 - 4629,45]
Erro relativo μ e dados reais	0,166	0,070	0.109

Tabela 2: Resultados previstos para 2021

8 Conclusões

Os resultados para o caso da subprefeitura analisada (Aricanduva/V. Formosa), representou uma boa resposta, apresentando valores reais dentro da margem de confiança. O erro relativo em relação à média μ é aceitável, o que mostra que os valores estimados possuem uma diferença pequena entre os valores reais.

Alguns erros cometidos no desenvolver do projeto estiveram na criação dos gráficos de função densidade de probabilidade. Uma melhor opção seria a utilização de um histograma, o que evidenciaria melhor o caráter de uma variável discreta. Este objetivo não foi alcançado pela dificuldade de escrever o programa em python tendo que absorver alguns conceitos novos para este exercício.

Referências

[1] Pedro A. Morettin. *Análise de Séries Temporais-Edgard Blucher (2006)*. Blucher, 2006.

[2] ARIMA Model – Complete Guide to Time Series Forecasting in Python selva prabhakaran. https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. Accessed: 17-01-2022.

[3] Pipelines with auto$_a$*rimapmdarimadocumentation.*. Accessed: 17-01-2022.

Time Series Models ar, ma, arma, arima. https://towardsdatascience.com/time-series-models-d9266f8ac7b0. Accessed: 13-01-2022.