Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Estatística

Processos Estocásticos

Relatório do Trabalho Final

Professor: Omar Cleo Neves Pereira

Vítor Hugo Santos de Camargo RA: 116426

1 Introdução

Processos estocásticos são uma coleção de variáveis aleatórias e são uma ferramenta poderosa em muitos campos, incluindo a matemática financeira e a engenharia. Em meteorologia, eles são essenciais para prever estados futuros com base em dados históricos. Este relatório busca analisar dados climáticos de várias cidades ao longo de um período de tempo e aplicar processos estocásticos para fazer previsões sobre o clima futuro. A importância deste estudo reside na necessidade crescente de prever mudanças climáticas, especialmente à luz dos crescentes desafios das mudanças climáticas. Através deste estudo, esperamos fornecer insights sobre o comportamento climático e fornecer ferramentas para prever mudanças futuras utilizando de matrizes de transições como principal base.

2 Desenvolvimento

O conjunto de dados usado para esta análise foi obtido do Kaggle. Ele contém informações sobre descrições de climas em 36 cidades diferentes, registradas hora a hora, ao longo de um período de 5 anos, desde 2013 até 2017.

Os processos estocásticos são métodos matemáticos que permitem prever futuros estados com base na probabilidade e nos estados anteriores. A matriz de transição, em particular, fornece uma visão detalhada de como um estado pode mudar para outro em uma sequência temporal.

O código a seguir foi desenvolvido para transformar e analisar esses dados.

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import os
import numpy as np
from statsmodels.tsa.seasonal import seasonal_decompose
```

Neste trecho, importamos as bibliotecas necessárias para trabalhar com manipulação de dados, plotagem de gráficos, cálculos numéricos, entre outros.

```

# Carregando os dados
data = pd.read_csv('weather_description.csv')

# Convertendo 'datetime' para datetime object e criando colunas para ano, mes, dia, hora e ano-mes
data['datetime'] = pd.to_datetime(data['datetime'])
data['year'] = data['datetime'].dt.year
data['month'] = data['datetime'].dt.month
data['day'] = data['datetime'].dt.day
data['hour'] = data['datetime'].dt.hour
data['year_month'] = data['datetime'].dt.to_period('M')
```

No trecho acima, carregamos o arquivo weather_description.csv, que é a nossa base de dados. Em seguida, processamos as datas para permitir uma análise mais fácil e eficaz dos dados em Python.

```
1
2  cities = data.columns[1:-5]
3
4  results_list = []
5
6  for city in cities:
7      city_data_by_hour = data.groupby('hour')[city].value_counts().unstack
       ().fillna(0)
8      plot_heatmap(city_data_by_hour, city, matrix_type='frequency')
9      transition_matrix = create_transition_matrix(data[city])
10     plot_heatmap(transition_matrix, city)
11     plot_monthly_trends(data, city)
12     plot_seasonal_decomposition(data, city)
13
14     data_until_2016 = data[data['year'] < 2017]
15     data_2017 = data[data['year'] == 2017]
16     transition_matrix_until_2016 = create_transition_matrix(
       data_until_2016[city])
17
18     states = transition_matrix_until_2016.index.tolist()
19
20     # Abordagem 1: Prever 2017 com base apenas no ltimo estado de 2016
21     current_state = data_until_2016.iloc[-1][city]
22     predictions_based_on_last = [predict_next_state(current_state,
       transition_matrix_until_2016, states) for _ in range(len(data_2017))
       ]
23
24     # Abordagem 2: Prever de forma sequencial, atualizando o estado atual
        a cada previso
25     current_state = data_until_2016.iloc[-1][city]
26     predictions_sequential = []
27     for _ in range(len(data_2017)):
28         next_state = predict_next_state(current_state,
       transition_matrix_until_2016, states)
29         predictions_sequential.append(next_state)
30         current_state = next_state
31
32     actual_data_2017 = data_2017[city].tolist()
33
34     results_based_on_last = [1 if predictions_based_on_last[i] ==
       actual_data_2017[i] else 0 for i in range(len(
       predictions_based_on_last))]
35     results_sequential = [1 if predictions_sequential[i] ==
       actual_data_2017[i] else 0 for i in range(len(predictions_sequential
       ))]
36
```

```
    accuracy_based_on_2017 = sum(results_based_on_last) / len(
    results_based_on_last)
    accuracy_based_on_2013_2016 = sum(results_sequential) / len(
    results_sequential)

    new_row = {
        "City": city,
        "Accuracy_2017": accuracy_based_on_2017,
        "Accuracy_based_on_2013_2016": accuracy_based_on_2013_2016
    }
    results_list.append(new_row)
```

O código acima realiza uma análise aprofundada e modelagem de previsão de condições climáticas para diversas cidades. Inicialmente, ele segmenta os dados por hora para cada cidade, analisando as condições climáticas registradas a cada hora do dia. Isso é visualizado por meio de mapas de calor, que ajudam a representar a frequência de diferentes condições climáticas durante as horas do dia. Para entender as mudanças nas condições climáticas, o código constrói uma matriz de transição, que essencialmente captura a probabilidade de mudar de uma condição climática para outra. Novamente, esta matriz é visualizada usando um mapa de calor para uma representação gráfica mais intuitiva.

Além das análises por hora, o código examina as tendências mensais das condições climáticas em cada cidade. Isso é feito agregando dados em uma base mensal e observando quais condições climáticas são mais prevalentes ao longo dos meses. Este insight é ainda mais aprofundado com a decomposição sazonal. A decomposição sazonal é uma técnica estatística que divide uma série temporal em várias componentes, como tendência, sazonalidade e resíduo. Isso ajuda a identificar padrões subjacentes e potenciais ciclos no clima que podem não ser imediatamente visíveis.

Uma vez realizada a análise, o foco se volta para a previsão. O objetivo é prever as condições climáticas para o ano de 2017 usando dois métodos distintos. O primeiro método faz uma previsão para todo o ano de 2017 com base apenas na última observação registrada em 2016. O segundo método é mais abrangente. Ele utiliza uma matriz de transição derivada de dados que abrangem de 2013 a 2016. Este método começa com a última observação de 2016 e faz previsões sequenciais para 2017, usando cada nova previsão como base para a próxima, mas sempre guiado pelas tendências e padrões identificados nos dados de 2013 a 2016.

Depois de realizar essas previsões, o código compara os resultados previstos com os dados reais de 2017 para avaliar a precisão de ambos os métodos. Esta avaliação quantifica a eficácia dos métodos de previsão, indicando qual método se saiu melhor em termos de precisão. Os resultados são então compilados em uma lista que documenta a precisão de previsão para cada cidade.

Finalmente, comparamos as previsões com os dados reais de 2017 para entender a precisão das nossas abordagens.

```

```

```python
def create_transition_matrix(city_column):
    unique_conditions = city_column.dropna().unique()
    transition_matrix = pd.DataFrame(0, index=unique_conditions, columns=
    unique_conditions)
    for i in range(len(city_column) - 1):
        current_condition = city_column.iloc[i]
        next_condition = city_column.iloc[i+1]
        if pd.notna(current_condition) and pd.notna(next_condition):
            transition_matrix.loc[current_condition, next_condition] += 1
    transition_matrix = transition_matrix.divide(transition_matrix.sum(
    axis=1), axis=0)
    return transition_matrix
```

A função create_transition_matrix é central para entender as mudanças nas condições climáticas em uma cidade específica. No coração dessa função está a ideia de criar uma matriz de transição, que é, essencialmente, uma representação das probabilidades de mudar de uma condição climática para outra.

O primeiro passo na função é identificar todas as condições climáticas únicas que foram registradas para uma cidade. Ao fazer isso, a função garante que qualquer condição climática que tenha ocorrido ao menos uma vez esteja representada na matriz. Este processo envolve remover quaisquer dados nulos e catalogar todas as condições climáticas que aparecem nos registros.

Uma vez que essas condições climáticas únicas são identificadas, a matriz de transição é inicializada. Esta matriz é uma representação tabular onde cada linha e coluna correspondem a uma condição climática única. Inicialmente, todas as entradas na matriz são definidas como zero, denotando que ainda não registramos nenhuma transição.

Com a estrutura da matriz pronta, o próximo passo é preenchê-la com dados reais. A função faz isso iterando por todos os registros de condições climáticas para a cidade. Para cada ponto no tempo, a função observa a condição climática atual e a próxima condição registrada. Com essas informações, ela incrementa a contagem na matriz para essa transição específica. Este processo é repetido para todos os registros, criando um mapa detalhado de quantas vezes cada transição específica ocorreu.

No entanto, contagens puras não são particularmente úteis por si só. O que realmente queremos saber é a probabilidade de uma transição específica ocorrer. Para transformar as contagens em probabilidades, a matriz é normalizada. Cada valor é dividido pelo total de transições registradas para a respectiva condição climática. Isso garante que cada linha da matriz soma um total de 1, com cada entrada representando a probabilidade da transição correspondente.

Finalmente, com a matriz preenchida e normalizada, a função a retorna, oferecendo uma representação abrangente e probabilística das transições entre condições climáticas para a cidade em questão. Esta matriz de transição é fundamental para fazer previsões informadas sobre como as condições climáticas podem mudar no futuro com base nos dados históricos.

Foram utilizadas duas abordagens: uma que faz previsões com base apenas no último estado registrado e outra que atualiza o estado após cada previsão,

permitindo uma sequência de previsões.

```
1
2  def plot_matrix(matrix_data, city_name, matrix_type='transition'):
3      plt.figure(figsize=(20, 15))
4      if matrix_type == 'frequency':
5          fmt_str = ".0f"
6      else:
7          fmt_str = ".2f"
8      sns.heatmap(data=matrix_data, cmap='YlGnBu', annot=True, fmt=fmt_str)
9      if matrix_type == 'transition':
10         save_path = f'matrix/{city_name}_{matrix_type}_matrix.png'
11     else:
12         save_path = f'heatmap/{city_name}_{matrix_type}_matrix.png'
13     plt.title(f'{matrix_type.capitalize()} Matrix for {city_name}')
14     plt.savefig(save_path, bbox_inches='tight')
15     plt.close()
```

A função plot_matrix é uma expressão elegante da necessidade de visualizar dados em um formato que destaque relações e tendências. No contexto do estudo climático deste código, entender a dinâmica das condições climáticas através de uma representação visual pode ser muito mais perspicaz do que simplesmente olhar para os números brutos.

Ao iniciar a função, é configurada a base para a visualização, estabelecendo um tamanho específico para o gráfico, assegurando que a matriz final seja de fácil leitura e interpretação. Uma característica distintiva desta função é sua capacidade de lidar com dois tipos de matrizes: de frequência e de transição. Enquanto uma matriz de frequência reflete as contagens diretas e, assim, é expressa em números inteiros, uma matriz de transição, que indica as probabilidades de transição de uma condição climática para outra, é mais sutil e, portanto, é representada em formato decimal para capturar essa nuance.

3 Resultados

Após a execução meticulosa do algoritmo, somos apresentados a uma ampla gama de resultados visuais que encapsulam as complexidades e nuances do clima em várias cidades ao longo de cinco anos.

Primeiramente, temos 36 heatmaps distintos originados da função plot_matrix previamente citada. Cada um deles representa a contagem de diferentes condições climáticas ao longo dos cinco anos sob análise. Esses mapas de calor oferecem uma representação gráfica das variações de clima, permitindo-nos discernir rapidamente os padrões e frequências de determinados eventos climáticos para cada cidade.

Abaixo serão mostrado apenas 3 heatmaps de cidades diferentes, Albuquerque, Atlanta e Vancouver, como exemplo do resultado final, uma vez que mostrar todos os 36 heatmaps ocuparia muito espaço, o resto pode ser obtido ao executar o código final.

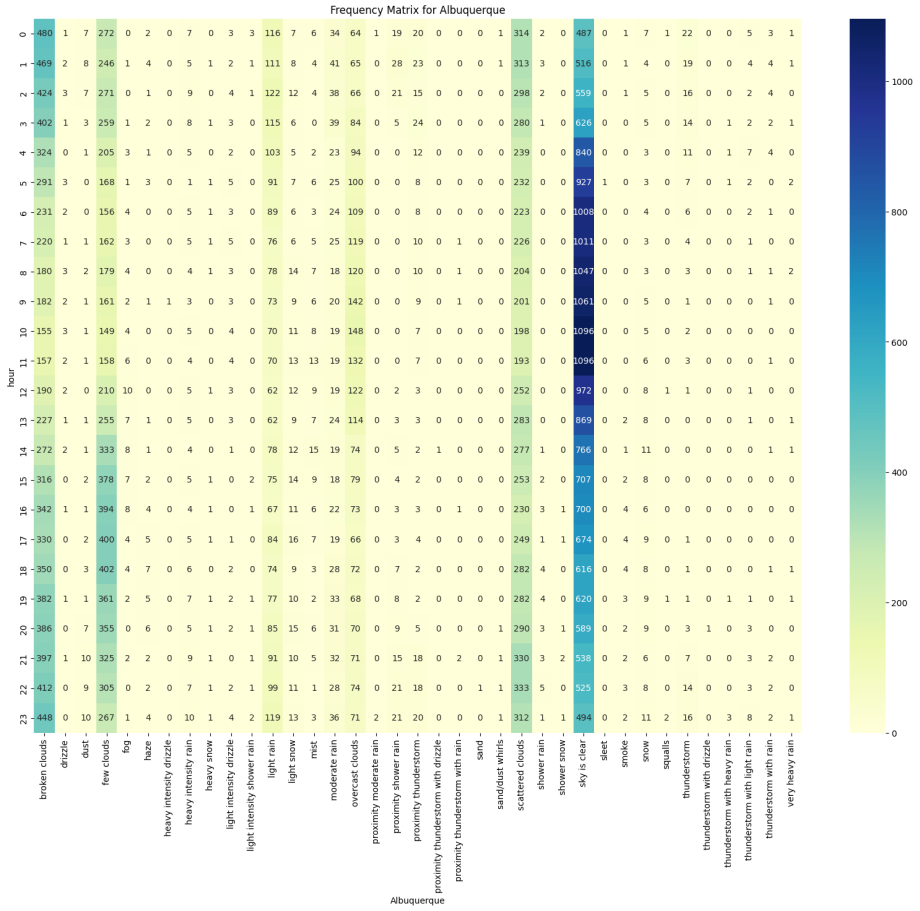

Figura 1: Matriz de frequência para a cidade de Albuquerque

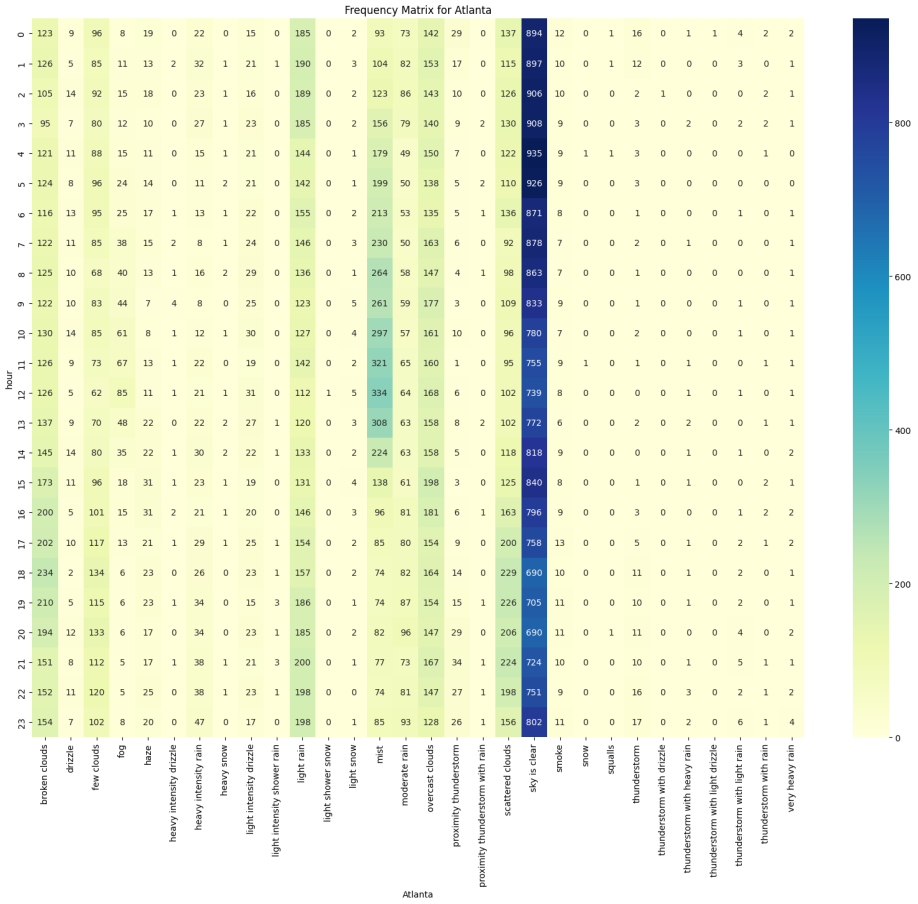

Figura 2: Matriz de frequência para a cidade de Atlanta

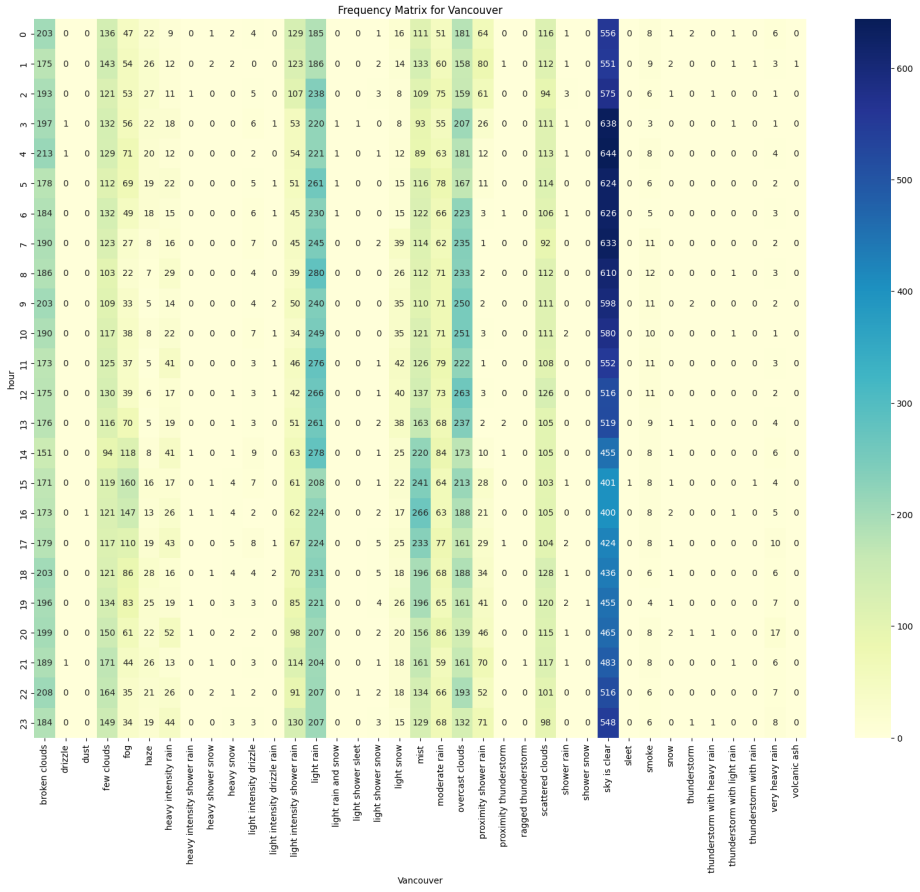

Figura 3: Matriz de frequência para a cidade de Vancouver

Em seguida, temos 36 matrizes de transição também originada da mesma função. Essas matrizes são cruciais para compreender a natureza dinâmica das condições climáticas. Elas mostram a probabilidade de transição de uma condição climática específica para outra, permitindo-nos prever, com base na condição climática atual, qual condição é mais provável de ocorrer a seguir. Novamente, será mostrado 3 matrizes de transição, originadas das mesmas 3 cidades acima, com suas totalidades podendo ser vistas ao executar o algoritmo.

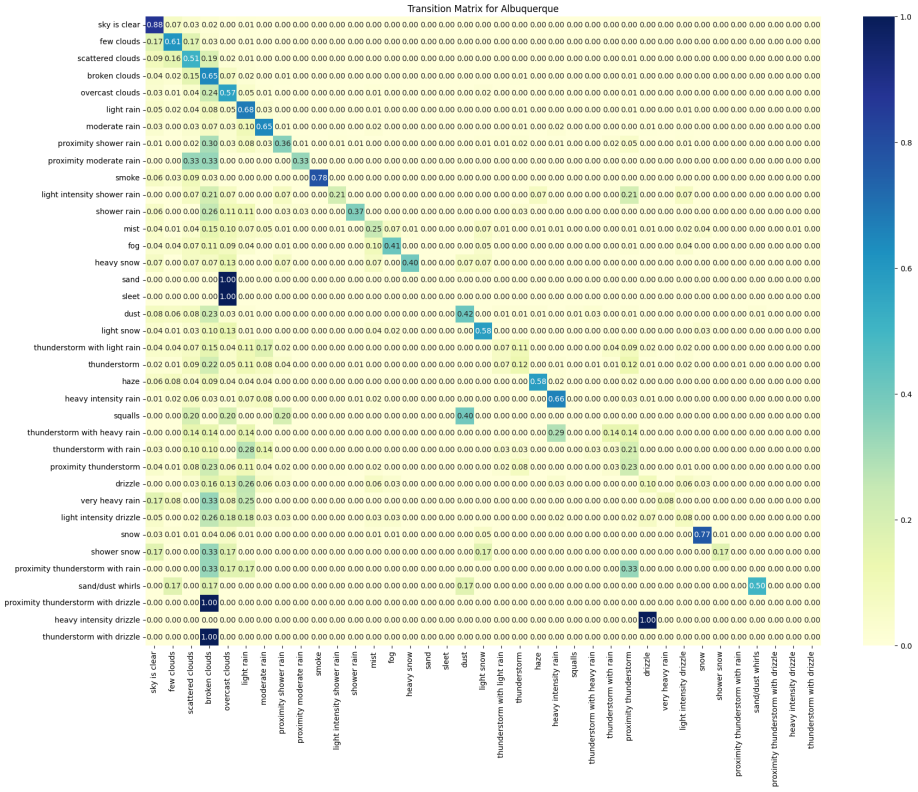

Figura 4: Matriz de transição para a cidade de Albuquerque

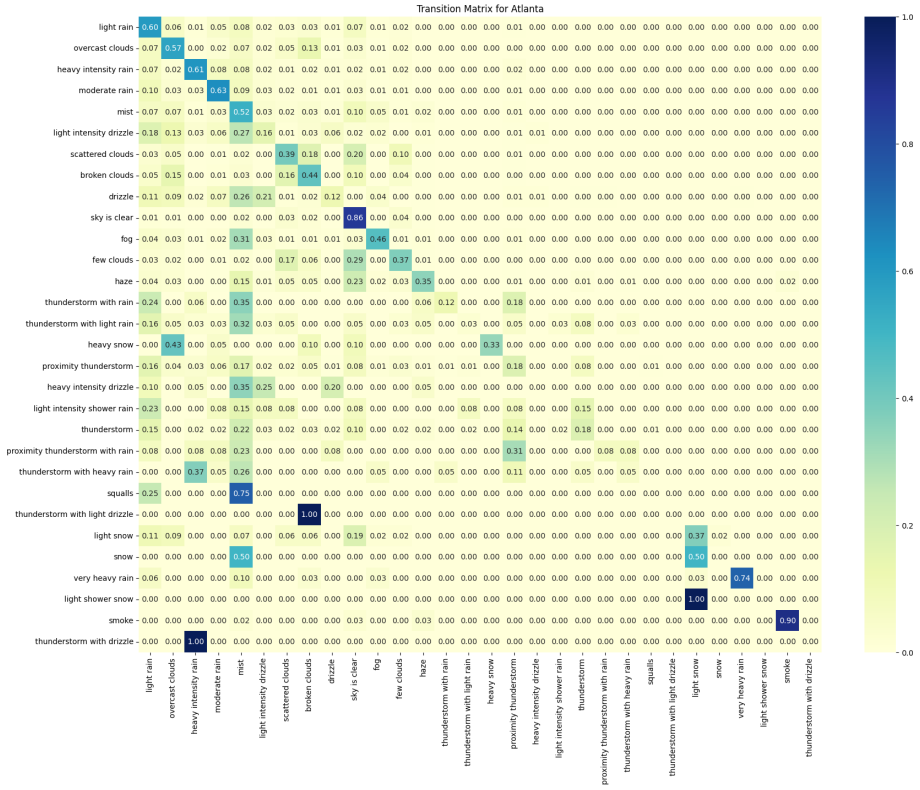

Figura 5: Matriz de transição para a cidade de Atlanta

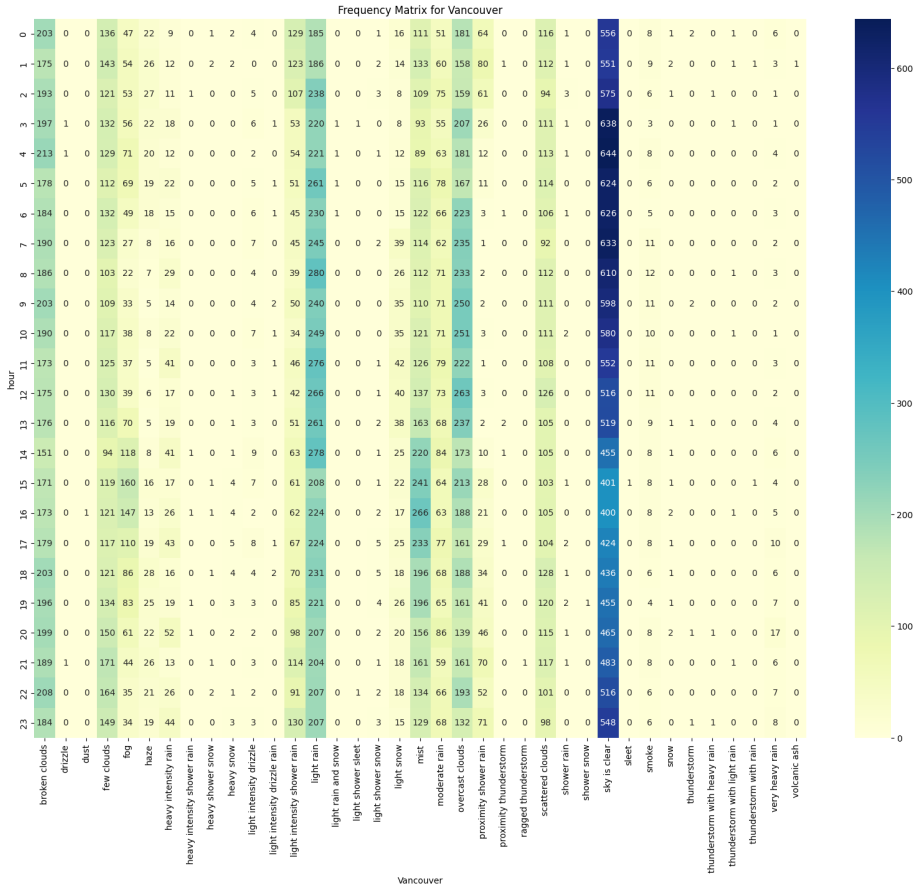

Figura 6: Matriz de transição para a cidade de Vancouver

```
1
2  def predict_next_state(current_state, transition_matrix, states):
3
4      #Funcao para prever o proximo estado usando a matriz de transicao.
5
6      probabilities = transition_matrix.loc[current_state].values
7      return np.random.choice(states, p=probabilities)
```

Por fim, o código acima faz previsões baseado nos dados de 2017 e os de 2013 à 2016 e, logo após retorna tais previsões para originar um gráfico de barras que nos apresenta as previsões. Essas previsões são o resultado de um esforço para antecipar as condições climáticas do ano de 2017. A primeira abordagem inicia com a última observação de 2016 para prever a primeira observação de 2017. Então, essa previsão é usada para prever a próxima e assim por diante, de forma sequencial, até o final do ano. A segunda abordagem, por outro

lado, é informada por um conjunto mais extenso de dados, que vai de 2013 a 2016. Ela faz previsões sequenciais ao longo de 2017, adaptando-se com base em suas próprias previsões anteriores e na matriz de transição derivada dos anos de 2013 a 2016. Este gráfico de barras compara a precisão de ambas as abordagens, avaliando o quão bem elas se alinharam com as condições climáticas reais observadas em 2017. A imagem abaixo mostra os resultados.

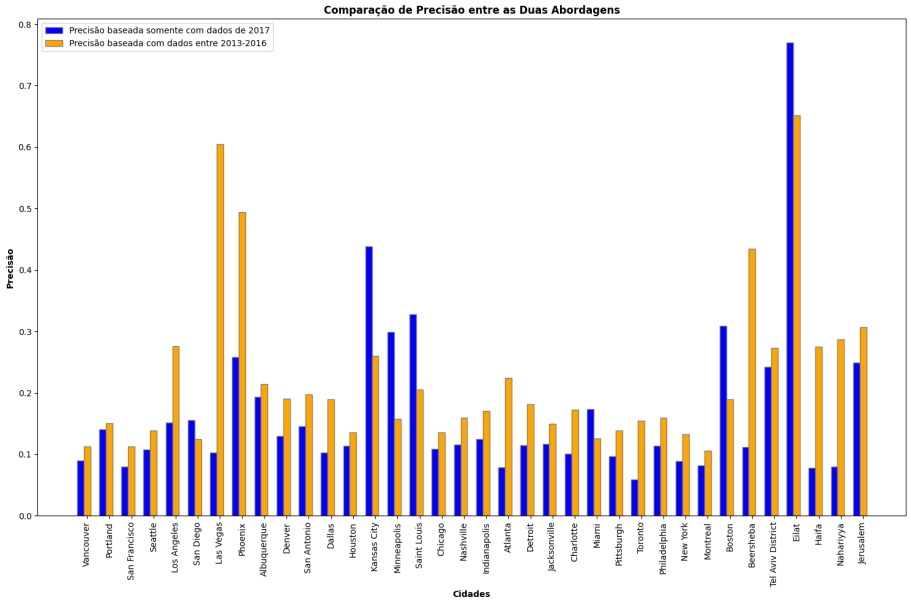

Figura 7: Precisão de acertos

É notável que, as previsões de 2017 baseadas com dados de 2013 à 2016 se sobre saiu na grande maioria dos casos, o que significa que utilizar uma base maior de dados para prever novos valores futuros parece ser o mais aconselhável. Porém, também é notável que a grande maioria das previsões ficou abaixo dos 50% de acertos, provavelmente pelo fato de que, existem tantas transições possíveis que, cada vez as previsões se tornam menos efetivas.

4 Conclusão

Nesta análise, exploramos a aplicação de processos estocásticos para prever condições climáticas em várias cidades, usando dados coletados de 2013 a 2017. A matriz de transição emergiu como uma ferramenta valiosa, permitindo-nos capturar a probabilidade de transição entre diferentes estados do clima.

Os resultados obtidos demonstram a complexidade e a variabilidade inerente às condições climáticas. Embora algumas de nossas previsões tenham alcançado um nível razoável de precisão, em muitos casos a taxa de acerto ficou abaixo de

50%. Isso destaca a natureza imprevisível do clima e a necessidade de abordagens de modelagem ainda mais refinadas.

Um insight valioso extraído de nossa análise é a eficácia relativa de usar um conjunto de dados mais amplo para previsões. As previsões baseadas em dados de 2013 a 2016 se mostraram mais precisas em comparação com aquelas baseadas apenas no último estado registrado em 2016. Isso sugere a importância de utilizar um volume significativo de dados históricos ao tentar modelar e prever fenômenos complexos como o clima.

A aplicação de processos estocásticos ao estudo climático é apenas um dos muitos métodos disponíveis, e nossa análise sugere que pode haver valor em explorar abordagens complementares ou alternativas no futuro. À medida que a necessidade de compreender e prever as condições climáticas se torna cada vez mais crítica, esforços como este servem como um passo vital em direção a um futuro mais informado e preparado.

Referências

[1] Selfish Gene. Historical hourly weather data. `https://www.kaggle.com/datasets/selfishgene/historical-hourly-weather-data`, Ano de acesso. Acessado em: 07/10/2023.