# ANÁLISE DAS ESTAÇÕES DO ANO

relatório do projeto final

Vitória Nazareth
121076766
2024.1

## Apresentação

### Objetivo

O tema do meu projeto aborda as estações do ano. A ideia principal é estudar o comportamento dos atributos que compõem o ‘tempo’ (temperatura) durante a passagem de um ano, dividido por períodos de atuação de cada estação. Pretendo utilizar as ferramentas vistas em sala de forma analítica, descrevendo as curvas de temperatura características de cada estação.

### Métricas

A principal métrica utilizada são métodos vistos em sala que serão especificados durante as análises. Também vou usar fontes para comparar os resultados obtidos aqui com dados ‘reais’.

### Nota

O projeto que descrevo ao longo deste relatório é uma versão mais ‘robusta’ daquela apresentada, visto que era muito mais simples. Dito isto, o relatório também apresentará comparações e as devidas mudanças entre ambas as versões.

# Dados

## Versão 1.

Inicialmente, minha fonte era um link para uma página de dados de tempo onde eu tinha acesso a apenas um gráfico com as médias de temperatura e precipitação no Rio de Janeiro no período de um ano. Não havia forma de extrair os dados de maneira automática, então eu os anotei, manualmente, criando duas listas para cada estação. A lista de índice 'x' guardava as temperaturas em graus Celsius e o 'y' armazenava a precipitação em mm (litro por metro quadrado).

Cada estação tinha 3 meses bem definidos, o que me fazia perder precisão logo de início uma vez que as estações não são exatamente correspondentes com os meses (março, por exemplo, se divide entre outono e verão). Como solução, eu havia optado por atribuir o mês da estação dominante: verão ocupava a maior parte de março, logo o mês de março faria parte apenas dessa estação e não de outono.

```
#outono
o_x = [20, 17, 11]
o_y = [100, 77, 51]
#inverno
i_x = [12, 8, 17]
i_y = [52, 42, 77]
#primavera
p_x = [19, 27, 32]
p_y = [99, 47, 152]
#verão
v_x = [34, 23, 31]
v_y = [174, 123, 151]
```

## Versão 2.

Meus dados agora vem de uma tabela (.csv) de tempo contendo números de mais ou menos 4 anos de análise (2013 - 2017). Cada linha da tabela representa um dia e as colunas são atributos do tempo que foram levados em consideração. São eles: data, temperatura média, umidade, velocidade do vento e média da pressão atmosférica, respectivamente. Eu escolhi representar o intervalo de um ano inteiro (2013) e também escolhi utilizar apenas os dados da temperatura média e da pressão atmosférica.

| | date | meantemp | humidity | wind_speed | meanpressure |
|---|---|---|---|---|---|
| 0 | 2013-01-01 | 10.000000 | 84.500000 | 0.000000 | 1015.666667 |
| 1 | 2013-01-02 | 7.400000 | 92.000000 | 2.980000 | 1017.800000 |
| 2 | 2013-01-03 | 7.166667 | 87.000000 | 4.633333 | 1018.666667 |
| 3 | 2013-01-04 | 8.666667 | 71.333333 | 1.233333 | 1017.166667 |
| 4 | 2013-01-05 | 6.000000 | 86.833333 | 3.700000 | 1016.500000 |
| ... | ... | ... | ... | ... | ... |
| 1457 | 2016-12-28 | 17.217391 | 68.043478 | 3.547826 | 1015.565217 |
| 1458 | 2016-12-29 | 15.238095 | 87.857143 | 6.000000 | 1016.904762 |
| 1459 | 2016-12-30 | 14.095238 | 89.666667 | 6.266667 | 1017.904762 |
| 1460 | 2016-12-31 | 15.052632 | 87.000000 | 7.325000 | 1016.100000 |
| 1461 | 2017-01-01 | 10.000000 | 100.000000 | 0.000000 | 1016.000000 |

1462 rows × 5 columns

*[Daily Climate time series data (kaggle.com)]*

Separei essa tabela em tabelas menores, uma para cada estação. Como cada linha corresponde a um dia, eu poderia agora separar perfeitamente os dias de cada estação sem precisar considerar meses inteiros. A divisão de estações se encontrava assim:

- Primavera: 21 de março - 21 de junho (dias 89 a 172)
- Verão: 21 de junho - 21 de setembro (dias 172 a 263)
- Outono: 21 de setembro - 21 de dezembro (dias 263 a 353)
- Inverno: 21 de dezembro - 21 de março (dias 353 a 89)

| | dias | meantemp | meanpressure |
|---|---|---|---|
| 89 | 90 | 23.200000 | 1008.600000 |
| 90 | 91 | 25.375000 | 1008.500000 |
| 91 | 92 | 25.166667 | 1009.500000 |
| 92 | 93 | 26.200000 | 1009.000000 |
| 93 | 94 | 24.600000 | 1007.800000 |
| ... | ... | ... | ... |
| 167 | 168 | 26.875000 | 994.750000 |
| 168 | 169 | 28.400000 | 996.400000 |
| 169 | 170 | 29.857143 | 999.000000 |
| 170 | 171 | 33.000000 | 997.166667 |
| 171 | 172 | 34.833333 | 995.333333 |

83 rows × 3 columns

| | dias | meantemp | meanpressure |
|---|---|---|---|
| 172 | 173 | 35.600000 | 996.200000 |
| 173 | 174 | 35.166667 | 995.666667 |
| 174 | 175 | 33.142857 | 996.285714 |
| 175 | 176 | 30.571429 | 999.166667 |
| 176 | 177 | 30.666667 | 999.000000 |
| ... | ... | ... | ... |
| 258 | 259 | 29.666667 | 1002.333333 |
| 259 | 260 | 29.250000 | 1002.875000 |
| 260 | 261 | 29.142857 | 1003.142857 |
| 261 | 262 | 29.800000 | 1001.800000 |
| 262 | 263 | 28.666667 | 1003.000000 |

91 rows × 3 columns

*[primavera e verão da esquerda para a direita]*

| | dias | meantemp | meanpressure |
|---|---|---|---|
| 262 | 263 | 28.666667 | 1003.000000 |
| 263 | 264 | 25.200000 | 1003.000000 |
| 264 | 265 | 28.333333 | 1001.666667 |
| 265 | 266 | 30.285714 | 1001.000000 |
| 266 | 267 | 30.750000 | 1002.750000 |
| ... | ... | ... | ... |
| 348 | 349 | 15.500000 | 1015.125000 |
| 349 | 350 | 15.250000 | 1015.625000 |
| 350 | 351 | 14.750000 | 1014.250000 |
| 351 | 352 | 14.875000 | 1012.625000 |
| 352 | 353 | 16.125000 | 1012.875000 |

91 rows × 3 columns

| | dias | meantemp | meanpressure |
|---|---|---|---|
| 0 | 354 | 15.375000 | 1016.000000 |
| 1 | 355 | 14.750000 | 1017.000000 |
| 2 | 356 | 15.250000 | 1018.375000 |
| 3 | 357 | 14.250000 | 1020.500000 |
| 4 | 358 | 13.500000 | 1021.375000 |
| ... | ... | ... | ... |
| 95 | 85 | 24.142857 | 1008.857143 |
| 96 | 86 | 21.000000 | 1009.000000 |
| 97 | 87 | 22.428571 | 1009.571429 |
| 98 | 88 | 21.250000 | 1009.750000 |
| 99 | 89 | 23.500000 | 1008.625000 |

100 rows × 3 columns

*[outono e inverno da esquerda para a direita]*

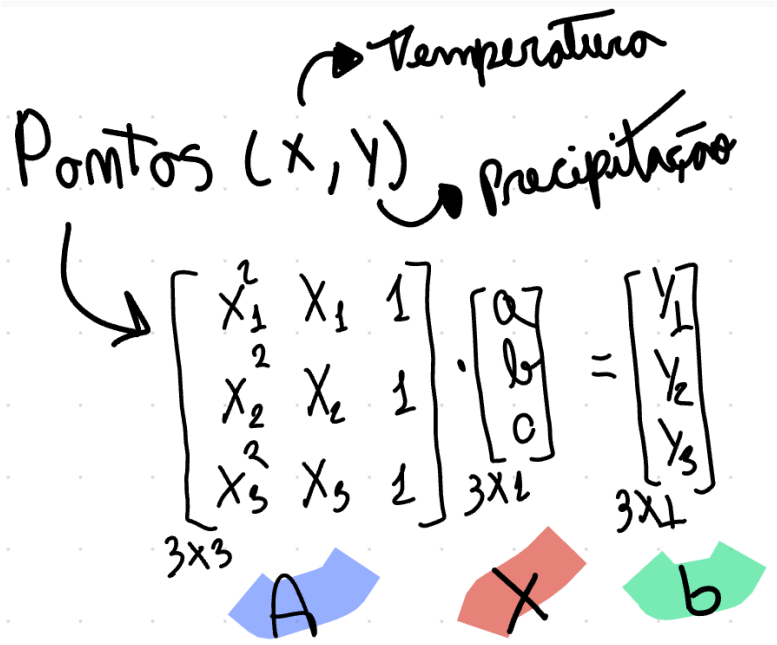
Outra diferença que eu preciso apontar entre a nova versão e a antiga é a origem dos dados. A minha primeira fonte me fornecia dados de temperatura com origem no Rio de Janeiro e o meu novo banco me fornece dados de New Delhi, capital da Índia. Por isso segui a ordem das estações na Índia. Esses dados foram os mais completos que eu consegui achar, por isso a ordem das estações está diferente.

A mudança do Brasil para a Índia não é assim tão simples. Eu adotei o nosso modelo na minha representação, mas os indianos possuem um modelo bastante diferente (que eu ignorei). Para maiores detalhes, eu separei a *[ref1]* para uma leitura sobre as estações do ano na Índia. O que importa aqui é que a “tradução” ficou equivalente. Confia.

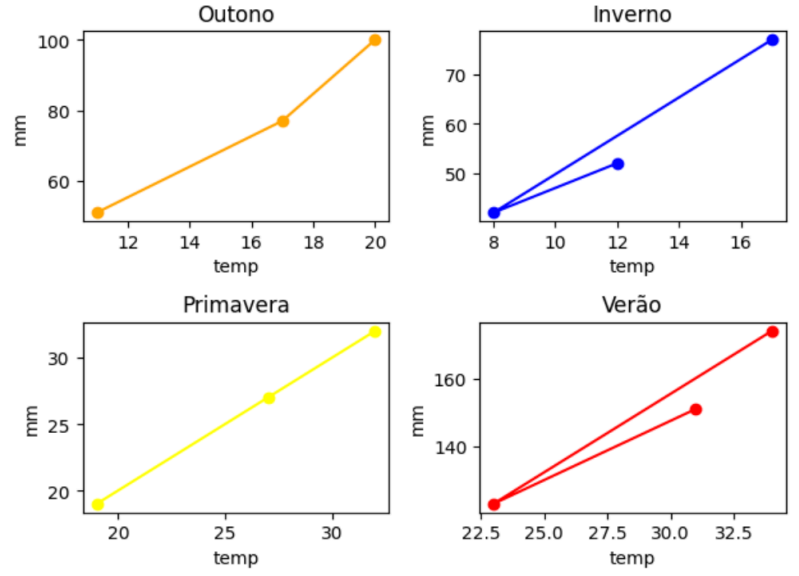
# Projeto

## Versão 1.

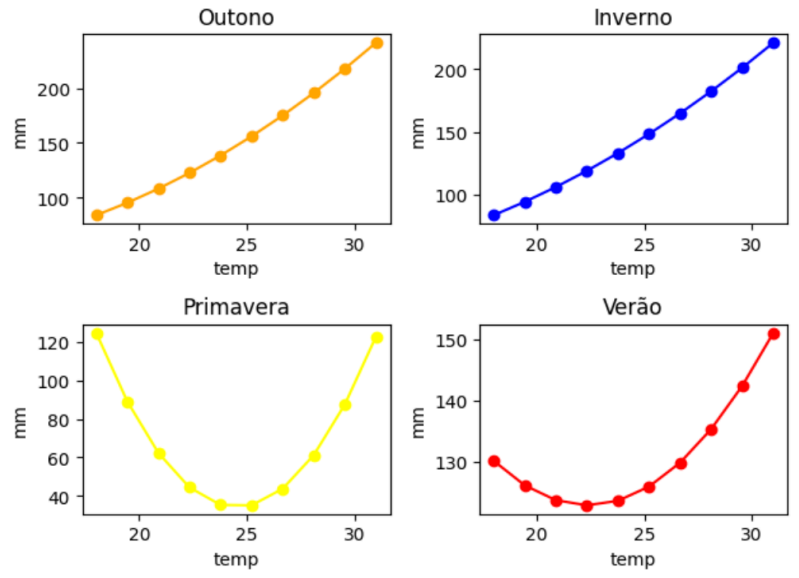
A interpolação no meu projeto serviria para descrever a curva característica de cada estação. Com isso, meu objetivo era poder identificar padrões nas características de mudança no tempo para cada estação. Para interpolar, eu preciso de um modelo preciso, isto é, meu modelo precisa ‘cravar’ meus pontos. Como eu tinha 3 pontos, escolhi interpolar por uma quadrática, gerando uma curva que passava exatamente por cima de cada ponto. O desenho abaixo mostra como fiz a modelagem do sistema $Ax = b$ para a interpolação de dados de cada estação.




Nas duas imagens abaixo, eu mostro o antes e o depois do plot dos pontos com as curvas já definidas.




*plotagem dos pontos para cada estação (sem interpolação)*




*plotagem de 10 pontos para cada estação (com interpolação)*

Nesta última imagem, eu gerei mais pontos dentro dessas curvas apenas para melhor visualização. Esses pontos estão no intervalo de 18 e 31 graus Celsius, pois, segundo o google, são as temperaturas médias (mínima e máxima) anuais no Rio de Janeiro. E o que podemos tirar dessa plotagem? Não há muita coisa. Considerando que o verão é a última estação do ano (começa no final) e outono a primeira, eu esperava ver, de alguma forma, as curvas se complementando, demonstrando essa transição entre as estações vizinhas, mas isso não acontece. Isso decorre pela forma como eu escolhi plotar (temp x mm) e também pelo fato de que, apesar de cada estacão ter sua média de temperatura natural, eu forcei as médias do ano. O que podemos observar é que a precipitação parece estar conectada com a temperatura de forma que, quando a temperatura está alta, a precipitação é mais alta também. O que faz sentido se pensarmos nas tempestades que costumamos ter durante o verão, por exemplo, com as temperaturas mais altas.

```
#plotando as curvas
def outono(x):
 return c_o[0]*(x**2) + c_o[1]*(x) + c_o[2]

x_o = np.linspace(18,31,10)
y_o = outono(x_o)
```

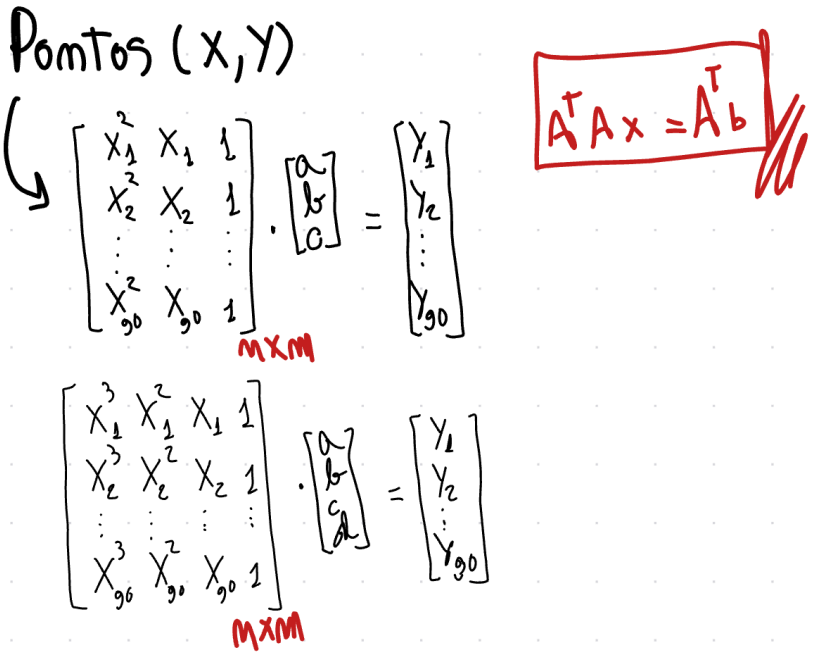
O trecho acima mostra um pedaço do código como exemplo de como interpolei e plotei as curvas exibidas anteriormente.

## Versão 2.

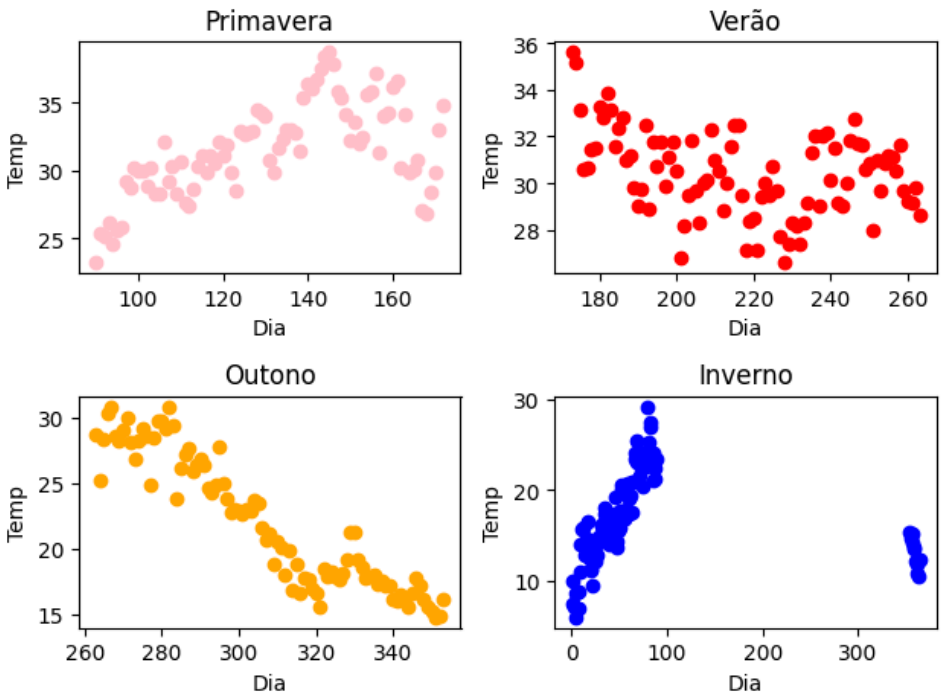
Eu fiz o mesmo processo para a minha nova versão. Para interpolar eu precisaria de novamente escolher um bom modelo. Contudo, diferentemente da

primeira versão, eu definitivamente não escolhi um polinômio de grau 90 para cravar meus +/- 90 pontos por estação. Meu objetivo é conseguir analisar as curvas características de cada estação, ou seja, um polinômio que permita enxergar um padrão do “movimento” da temperatura ao longo dos dias do ano era o suficiente, além de evitar os problemas de um overfit.

Minha ideia inicial foi comparar. Interpolei todos os meus pontos por uma quadrática e por uma cúbica. Apesar de saber que a cúbica me daria mais precisão, meu objetivo era saber se a quadrática já seria o suficiente para o meu objetivo. Por ser um sistema não quadrado, vou utilizar mínimos quadrados para sua solução. Minha coordenada ‘x’ representa meu dia do ano e ‘y’ minha temperatura.

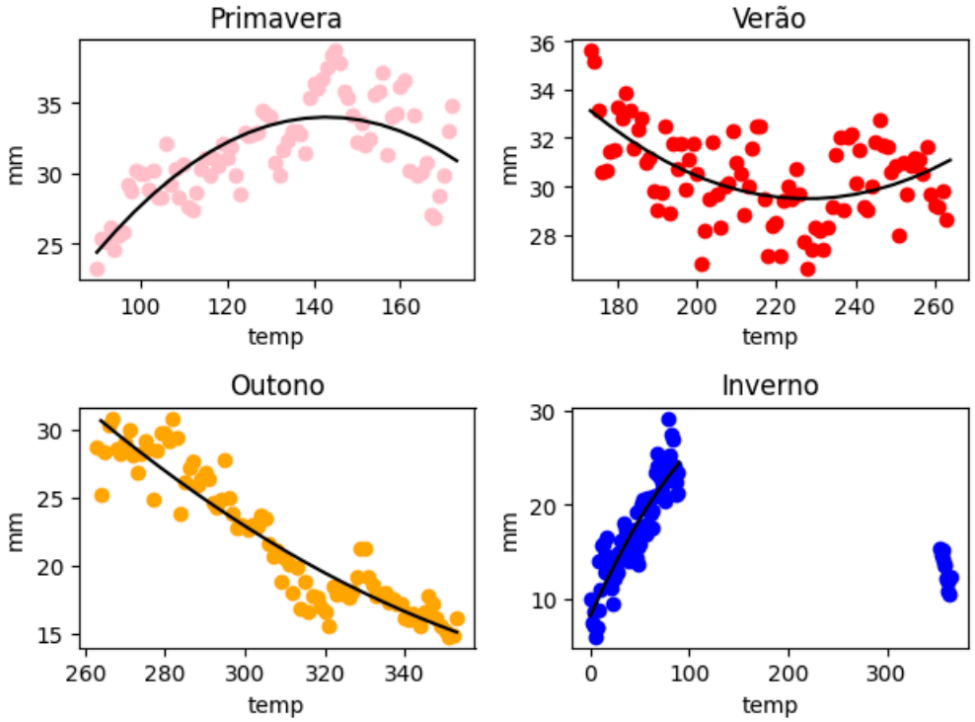



Abaixo seguem imagens das plotagens.



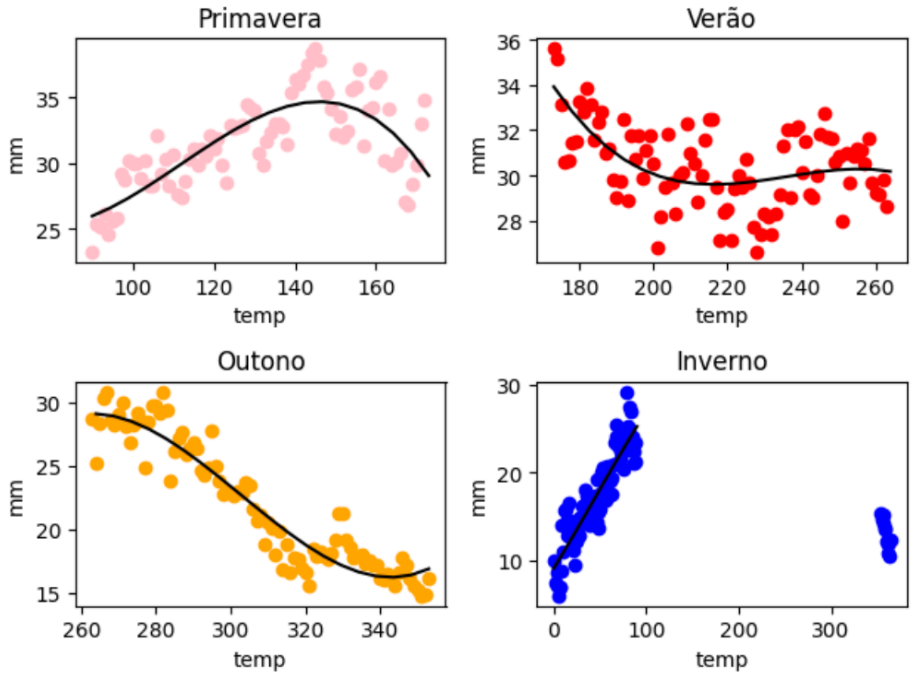

*[plot antes da interpolação: (dia, temp)]*

A estação do inverno está ‘partida’ assim porque começa no final do ano e termina no início. Podemos observar que as temperaturas começam baixas (canto direito) e vão subindo no início do ano, se aproximando da primavera.




Começando na primavera, as temperaturas se iniciam amenas e vão subindo no decorrer dos dias, com a chegada do verão. No verão, observamos que a maioria das temperaturas se encontram acima dos 30 graus, o que é uma característica comum dessa época do ano. No outono temos uma queda quase

linear de temperatura. Por último, no inverno, a temperatura parece subir de forma acelerada. Contudo, devido ao aglomerado de pontos na curva, podemos concluir que temos muitos dias com temperaturas parecidas. Entre 10 e 20 graus, por exemplo, temos muitos dias da estação sendo representados nesse intervalo.




Como esperado, a interpolação por uma cúbica é mais precisa. As curvas da cúbica são mais desenhadas expondo melhor as mudanças no tempo na passagem dos dias. Por exemplo, na primavera, temos uma leve acentuação da curva cúbica próxima ao dia 140 onde as temperaturas sobem mais, enquanto que na curva da quadrática não é possível observar isso.

Outra forma de confirmar a melhora na aproximação da quadrática para a cúbica é a diferença entre os valores do meu b para o Ax' calculado. Ou seja, posso fazer $||b - Ax'||^2$ e confirmar que $E^2 > E^3$, isto é, que o erro da quadrática é maior que o erro da cúbica.

```
#checando os erros
cp_e3 < cp_e2 and cv_e3 < cv_e2 and co_e3 < co_e2 and ci_e3 < ci_e2
```

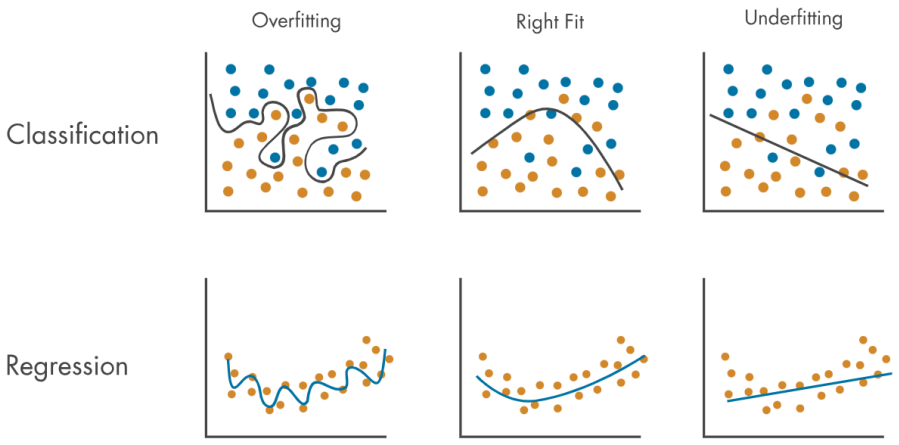
```
array([ True])
```

## Conclusão

Ao meu ver, a aproximação da quadrática é bastante boa. É possível ver os padrões de transição com bastante facilidade, o que eu queria. Talvez não no inverno. Isso provavelmente se deve a forma como eu escolhi representar seus dados na tabela. Como eu queria pegar do início ao fim, precisei pegar os dias em dois intervalos distantes para moldar sua plotagem, o que dificultou sua visualização. Na primavera para o verão, com os gráficos lado a lado podemos ver que suas curvas praticamente se complementam, o que é bastante legal considerando que são estações vizinhas. Vamos que a primavera começa com temperaturas baixas e subindo, pois está saindo do inverno, e então começa a subir até mudar para o verão com as temperaturas entre as estações praticamente se mantendo estáveis.

Em relação a cúbica, apesar de mais precisa, o fitting da curva não me pareceu tão bom a ponto de eu achar a curva da cúbica muito melhor se comparada com a da quadrática, por exemplo. E dado os desenhos, eu inclusive acredito que um polinômio de grau maior já poderia me distanciar do que o que eu queria ver. Entretanto, escolher uma curva mais precisa sem overfit não é uma decisão ruim.

Uma última coisa que podemos comentar é em relação a diferença demonstrada. A diferença calculada ali é o erro entre os pontos originais da tabela e aqueles novos que foram calculados. A ordem do erro é de 300 (eu mostro no código) o que é absurdamente alto quando eu penso nas atividades em sala de aula onde observava erros muito próximos de zeros, com expoente negativo. Isso faz parte do fato que estamos interpolando 90 pontos com um polinômio de grau 3. Eu sei, por teoria, que um polinômio de grau maior daria um erro menor e um polinômio de grau 90 me daria um erro igual a 0 (visto que eu teria uma solução única). Mas então entramos na questão do overfit.




Se eu tivesse plotado (nem acho que dá para fazer isso) com um polinômio de grau 90, meu gráfico seria muito poluído para ver e interpretar, como o primeiro exemplo da imagem acima. Todos aqueles noventas pontos estariam na minha curva que, por sinal, seria muito sinistra de feia.

ref1: Entenda as 6 estações do calendário hindu (casadaindia.com.br)