

226359

September 29, 2019

## 0.1 Jimi Togni - RA: 226359

---

### 0.1.1 Parte 1 - atividade teorica

---

### 0.1.2 Parte 2 – Atividade computacional

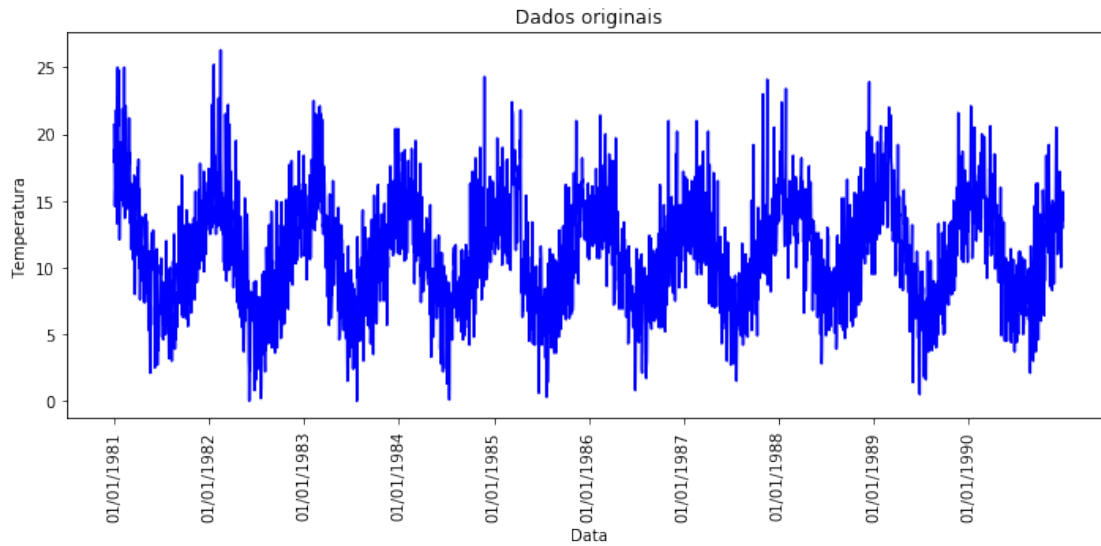
---

Ultimos 10 registros da base de dados

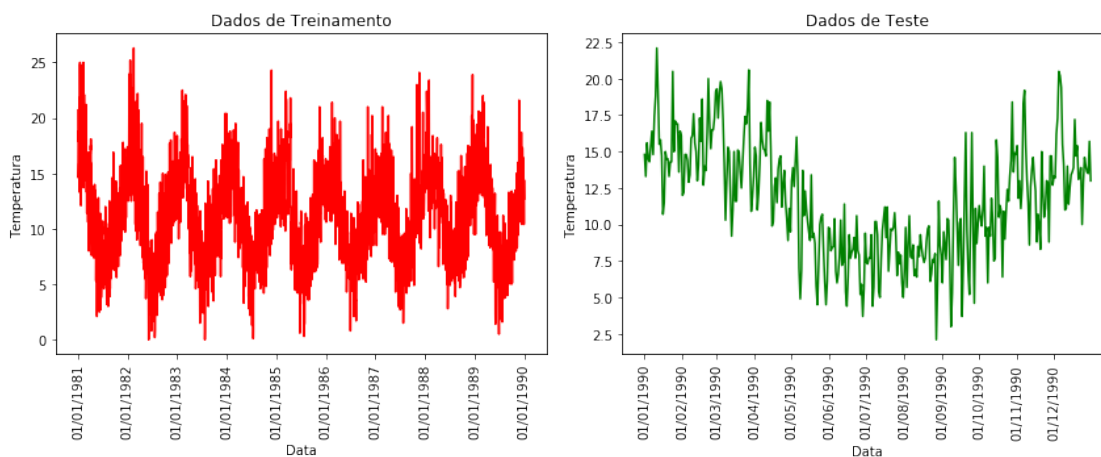
	Data	Temperature
3640	22/12/1990	13.2
3641	23/12/1990	13.9
3642	24/12/1990	10.0
3643	25/12/1990	12.9
3644	26/12/1990	14.6
3645	27/12/1990	14.0
3646	28/12/1990	13.6
3647	29/12/1990	13.5
3648	30/12/1990	15.7
3649	31/12/1990	13.0

```
/usr/lib/python3/dist-packages/ipykernel_launcher.py:7: UserWarning: This pattern has match gr
import sys
```

Grafico de toda a serie temporal



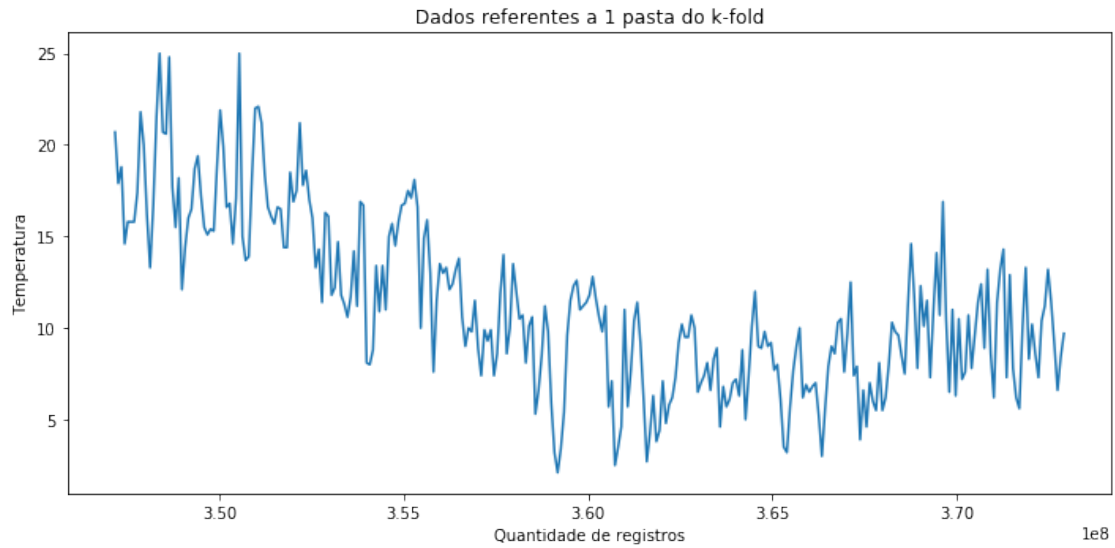
Divisão dos dados para treinamento e o ultimo ano, para teste



Utilização de K-Folds para dividir os dados de treinamento em pequenas “pastas” para verificar melhor configuração de treinamento dado os dados.

Conforme solicitado, os dados serão divididos em até 30 pastas, além disso, será testado a possibilidade de cada pasta conter dados randomicamente misturados de diferentes épocas para avaliar se o modelo se comporta de modo melhor ou pior em questão a temporalidade das informações.

`Out[55]: ' folds = kfold(train, k=8, shuffle=True)\n\nplt.figure(figsize=(8, 5))\nplt.title("D`



### Exercício 1 Gerando dados para os k atrasos

Valores:

Min: 0.0, Max: 24.72

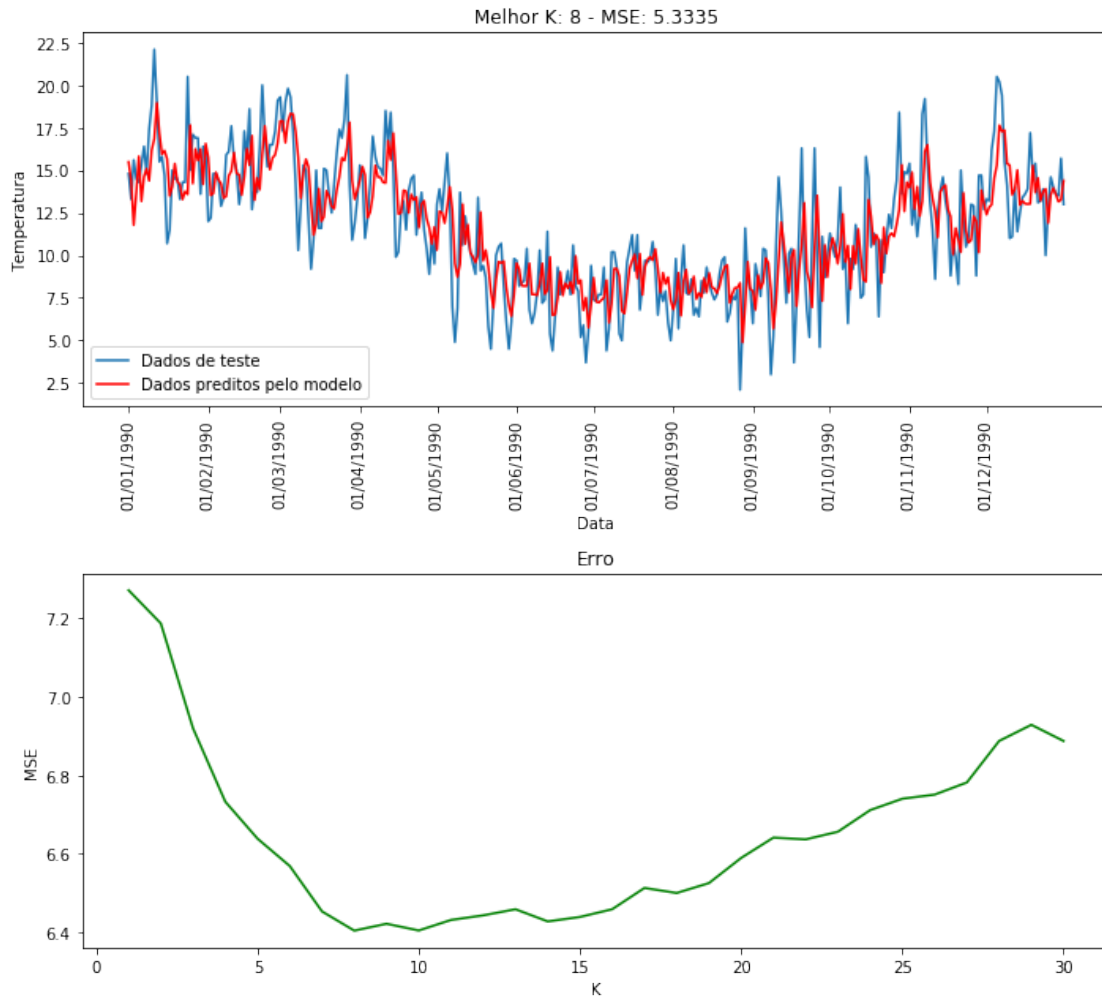
Executando a Regressão Linear, com K variando de 1 a 30 e k-fold variando de 1 a 20 folds

Resultados:

Melhores valores encontrados:

K: 8

k-fold: 6



Os gráficos acima, apresentam os valores após filtro pós-processamento para escolher o melhor valor de K.

Abaixo, são apresentados os primeiros 10 itens da iteração total executada. O primeiro item não representa a melhor opção, pois para escolha da melhor opção foi calculada a média dos valores.

	K	k-fold	Fold de Validacao	Média MSE
257	15	6	6	6.131738
275	16	6	6	6.139017
311	18	6	6	6.147777
131	8	6	6	6.149170
239	14	6	6	6.154521
130	8	5	5	6.156247
326	19	3	3	6.157078
164	10	3	3	6.157835
200	12	3	3	6.160508
167	10	6	6	6.160558

É possível também, usar de outra alternativa no método de K-Fold... no caso estamos embaralhando os dados antes de passar para o método e consequentemente o modelo. Por fim, chegamos aproximadamente no mesmo resultado, entretanto tomando um caminho de certa maneira diferente... Neste sentido, podemos encontrar os melhores valores  $W$  para o modelo em folds totalmente diferentes.

```
Out[85]: 'fold_N, fold_val = 0, 0\nJ_k = []\nfor k in range(1, 32):\n    for kf in range(1, 20):
```

```
Out[89]: 'best_result = None\nbest_J = 1e+10\nfor m in range(1, 32):\n    c = J_k[(J_k[:, 0] ==
```

```
Out[88]: '# Encontramos o melhor K e Validation Set.\nk, kf, i, mse, folds = best_result\nk = :
```

Abaixo, são apresentados os primeiros 10 itens da iteração total executada. O primeiro item não representa a melhor opção, pois para escolha da melhor opção foi calculada a média dos valores.

```
Out[86]: 'df = df.sort_values(by=["Média MSE", "K"]).head(10)\nprint(df)'
```

**Exercício 2** No exercício 2 usando o mesmo dataset usando anteriormente com a mesma questão de atraso, passaremos cada um dos itens por uma Rede Neural, usando como função de ativação a função hiperbólica.

Para validar a quantidade de unidades ( ou neurônios ) faremos a geração dessas unidades variando de 1 até 100 com seus pesos dentro de uma distribuição uniforme variando de -1 até 1.

Como valores para  $\lambda$  (regularização) será utilizado o seguinte range:  $1e+1$  até  $1e-6$ , dando espaçamentos de 0.1. Para visualmente ficar mais legível (devido a grande variação), os dados (os valores de regularização) são apresentados em escala logarítmica.

Para a normalização dos dados, evitando a saturação da tangente hiperbólica, os dados serão normalizados entre os valores de mínimo e máximo dos dados (os quais já foram apresentados acima).

Valores de  $K$ , estão dentro da faixa de 5 até 20 e o K-Fold utilizado foi de 1 até 10 folds.

6 valores para o lambda -> [1.e+01 1.e+00 1.e-01 1.e-02 1.e-03 1.e-04]

```
1 -> K: 7
2 -> K: 8
3 -> K: 9
4 -> K: 10
5 -> K: 11
6 -> K: 12
7 -> K: 13
8 -> K: 14
9 -> K: 15
10 -> K: 16
11 -> K: 17
12 -> K: 18
13 -> K: 19
```

14 -> K: 20  
15 -> K: 21  
16 -> K: 22

Melhores resultados obtidos:

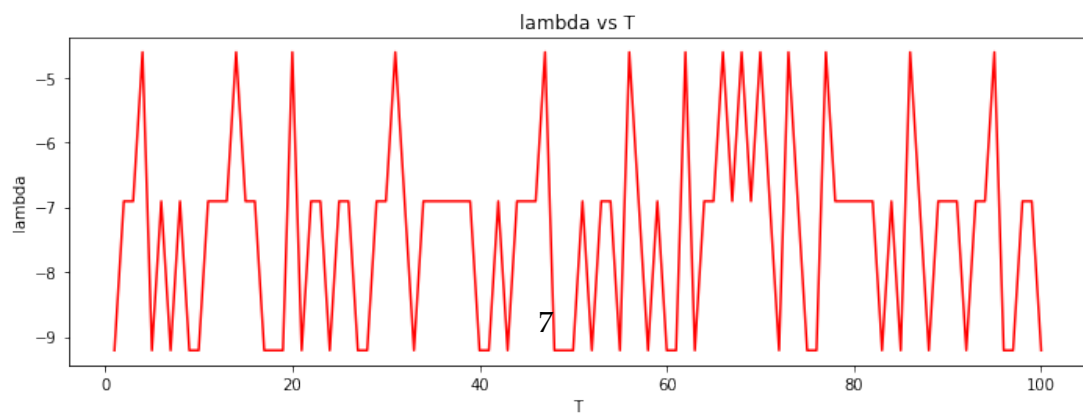
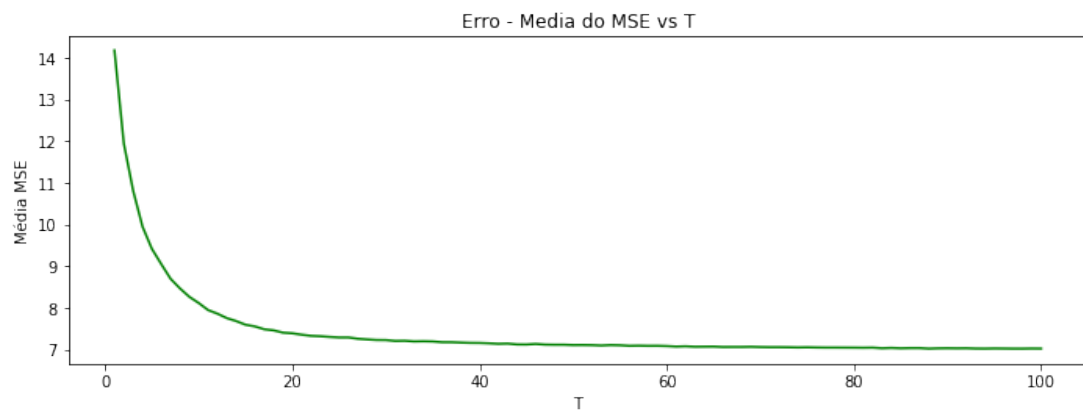
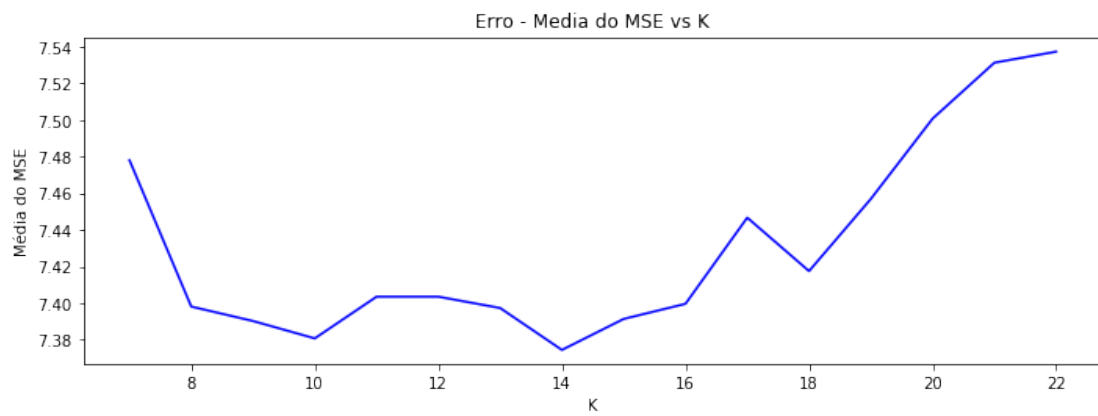
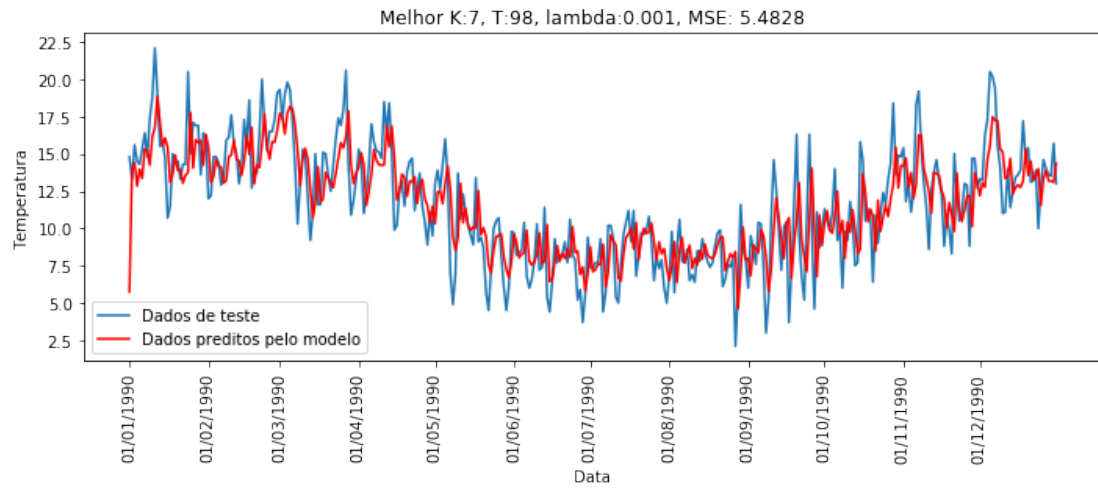
k-fold: 1

K'atrasos': 7

T: 98

lambda: 0.001

MSE: 6.8248264372249245



Abaixo, são apresentados os primeiros 10 itens da iteração total executada. O primeiro item não representa a melhor opção, pois para escolha da melhor opção foi calculada a média dos valores.

	K	K-Fold	Validation Fold	T	Regularizacao	Média MSE
13622	18	3	2	89	0.0010	6.549255
9969	15	3	2	72	0.0001	6.553487
11181	16	3	2	72	0.0010	6.558932
11209	16	3	2	100	0.0001	6.561989
9994	15	3	2	97	0.0001	6.565056
13619	18	3	2	86	0.0100	6.566511
8774	14	3	2	89	0.0010	6.568872
8786	14	3	2	101	0.0010	6.568935
11190	16	3	2	81	0.0010	6.569512
8778	14	3	2	93	0.0010	6.571622

O resultado do modelo acima ficou bem próximo do executado usando apenas a Regressão Linear simples (sem uma camada intermediária entre as entradas e o Regressor). Neste sentido, pela natureza dos dados, mesmo usando modelos mais complexos podemos acabar por chegar no mesmo resultado.