



Apresenta:  
Machine Learning com TensorFlow - Webinar Series

# S1E2 - Regressão Linear na Prática

Sandro Moreira

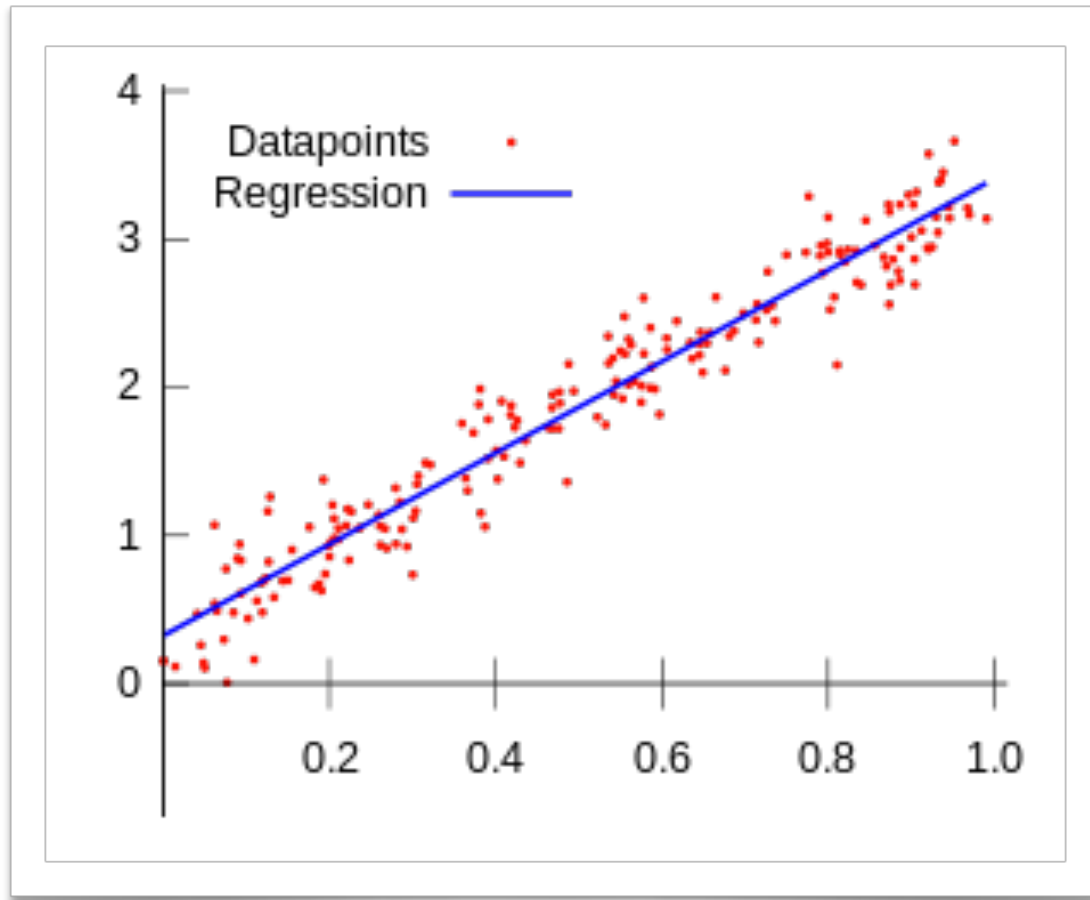


@sandro\_moreira

[moreira.sandro@gmail.com](mailto:moreira.sandro@gmail.com)



# Regressão Linear



A regressão linear é um algoritmo que encontra uma relação linear entre uma variável dependente e uma ou mais variáveis independentes. A variável dependente também é chamada de “labels” (rótulo) e variáveis independentes são chamadas de “features” (recursos)

# Para que serve uma Regressão?

Em um problema de regressão, o objetivo é prever as saídas (outputs) de um valor contínuo, como um preço ou probabilidade. Em contraste de problemas de classificação, onde temos o propósito de escolher uma classe em uma lista de classificações (por exemplo, se uma imagem contém uma maçã ou laranja, assim reconhecendo qual fruta é representada na imagem).

TensorFlow Docs

# Antes, um pouco sobre as versões do TensorFlow

- Versão Atual: 2.0
- Códigos escritos para versões 1.x podem não funcionar

Ex: `tf.placeholder` que funciona com o TensorFlow 1.x, não funciona com o 2.0  
(`AttributeError: module 'tensorflow' has no attribute 'placeholder'` )

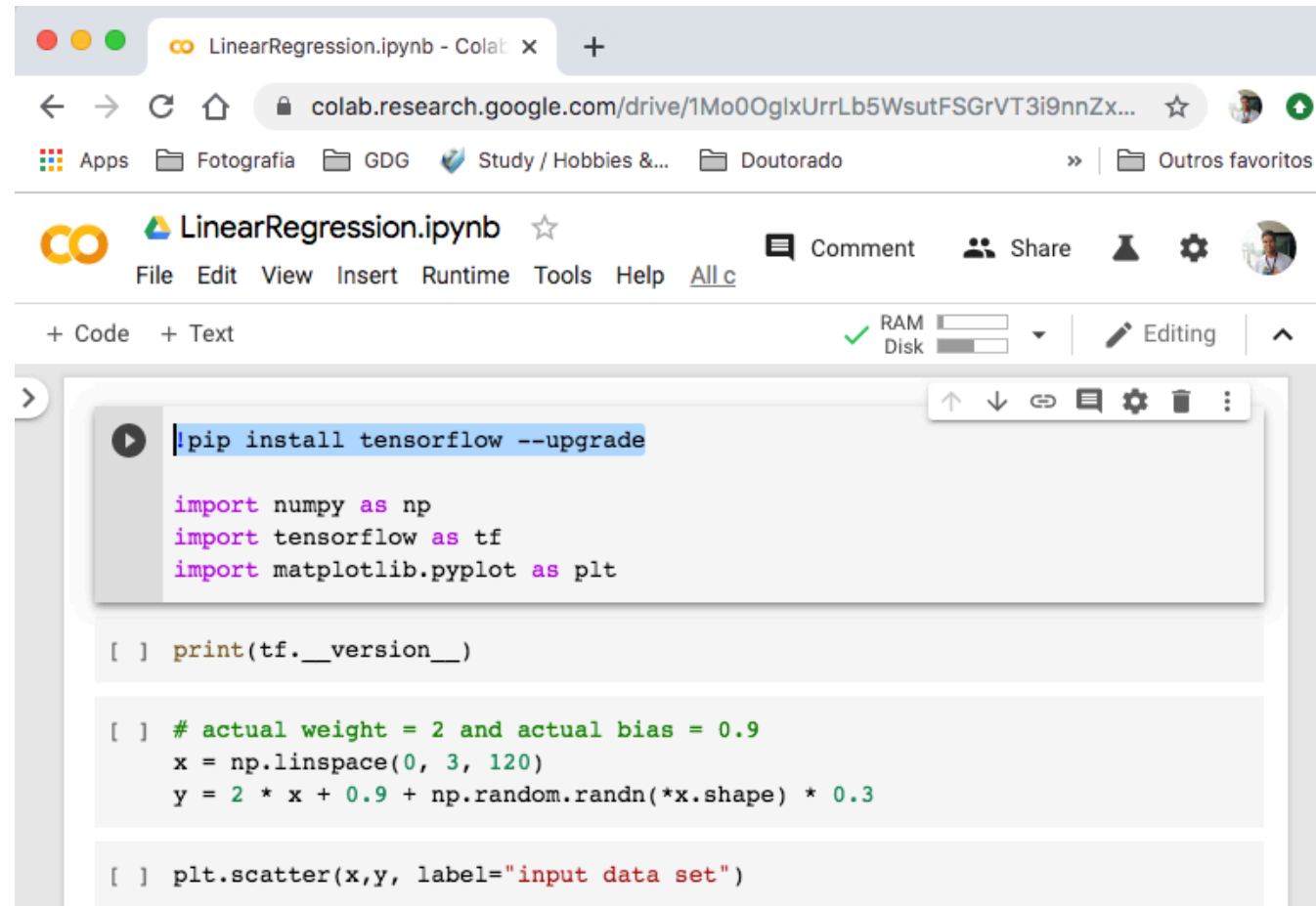
- Se você deseja executar o código existente (escrito na versão 1.x) com a versão 2.0, você tem duas opções:

## 1. Modo de compatibilidade:

```
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
```

## 2. Modifique seu código para trabalhar com a versão 2.0

# Google Colaboratory



The screenshot displays the Google Colaboratory web interface. The browser tab is titled 'LinearRegression.ipynb - Colab'. The address bar shows the URL 'colab.research.google.com/drive/1Mo0OglxUrrLb5WsutFSGrVT3i9nnZx...'. The notebook interface includes a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu bar, there are buttons for '+ Code' and '+ Text', and a status bar indicating 'RAM' and 'Disk' usage. The main code area contains the following Python code:

```
!pip install tensorflow --upgrade

import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

[ ] print(tf.__version__)

[ ] # actual weight = 2 and actual bias = 0.9
    x = np.linspace(0, 3, 120)
    y = 2 * x + 0.9 + np.random.randn(*x.shape) * 0.3

[ ] plt.scatter(x,y, label="input data set")
```

colab

- <https://colab.research.google.com>

# Como construir isso?

- Linguagens de programação como Python, R, Matlab são mais indicadas
- Nesses webinars utilizaremos **Python**
- Bibliotecas Python (Pandas, Matplot, Scikit Learning)
- Framework **TensorFlow** (nosso foco principal)



# Links Recomendados

- <https://ai.google>
- <https://aiyprojects.withgoogle.com>
- <https://cloud.google.com/products/ai/>
- <https://developers.googleblog.com/2018/05/introducing-ml-kit.html>
- <https://developers.google.com/machine-learning/crash-course>
- <http://playground.tensorflow.org>
- <https://www.tensorflow.org/guide>
- <https://keras.io>
- <https://colab.research.google.com>

Repositório: <https://github.com/smoreira>

Canal Youtube: <https://www.youtube.com/channel/UCnYATTd1RoLQryTqhySl6DQ>



Apresenta:

## Machine Learning com TensorFlow - Webinar Series

# Dúvidas?

Sandro Moreira



@sandro\_moreira

[moreira.sandro@gmail.com](mailto:moreira.sandro@gmail.com)

