CS550 - Machine Learning and Business Intelligence

Jupyter: Training Linear Models

By: Maranata Muluneh

Instructor: Dr. Chang, Henry

2023

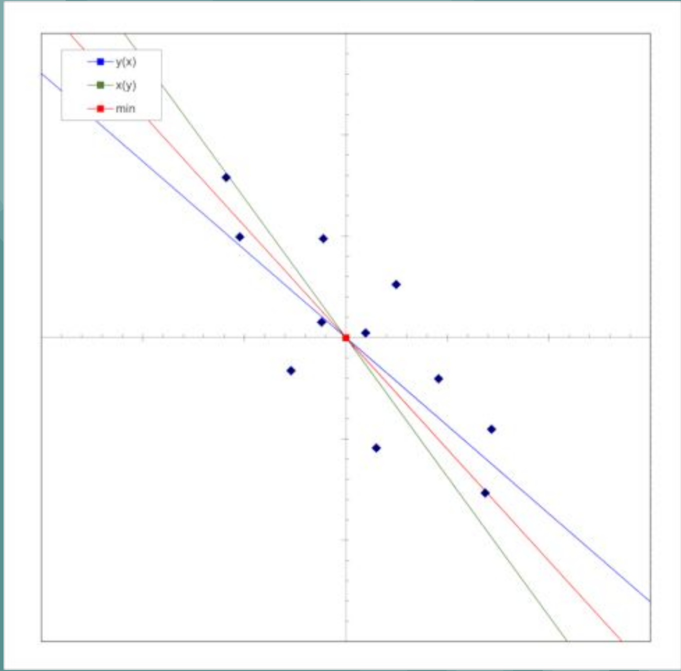


Table of Content

Introduction

The normal equation is a method to find the optimal solution for linear regression problems with a single variable or multiple variables. It is an analytical approach to find the values of the coefficients (weights) that minimize the mean squared error between the predicted output and the true output. This method is efficient for small to medium sized datasets, but becomes computationally expensive for large datasets as the computational complexity is O(n^3).

Theory

Machine Learning Process

During Machine Learning process we have three steps:

- Collecting data
- Creating model
- Prediction

Theory

Collect Data

- After we collected the data we needed we have to classify the collected data in to **training, validation and test** categories .
- Training (50% of the data)
- Evaluation(25% of the data)
- Test (25% of the data)

Theory

Create Model

- To create a model we have different Machine Learning (ML) types to evaluate the models, for this slide we are focusing on supervised learning specifically on Linear Regression.

Theory

Supervised learning

- Regression is one of the types if supervised learning includes Linear and nonlinear regression.
- We use overfitting model to evaluate the model either MSE or RMSE, while MSE is for numbers evaluation RMSE is for strings.

Mean Squared Error (MSE) , Root Mean Squared Error (RMSE)

Theory

Linear Regression

- A linear regression model follows a very particular form.
- Regression Equation(y) = a + bx
- Slope(b) = (NΣXY - (ΣX)(ΣY)) / (NΣX2 - (ΣX)2)
- Intercept(a) = (ΣY - b(ΣX)) / N

Theory

Linear Regression

Where:

- x and y are the variables.
- b = The slope of the regression line
- a = The intercept point of the regression line and the y axis.
- N = Number of values or elements
- X = First Score
- Y = Second Score
- ΣXY = Sum of the product of first and Second Scores
- ΣX = Sum of First Scores
- ΣY = Sum of Second Scores
- ΣX2 = Sum of square First Scores

Theory

Non Linear Regression

- A Non linear regression model is flexible and will not follow a specific rule or form.
- Regression Equation(y) = a + b^2
- Slope(b) = (NΣPY - (ΣP)(ΣY)) / (NΣP2 - (ΣP)2)
- Intercept(a) = (ΣY - b(ΣP)) / N
- Where P = X * X

For this slide we are focusing on Linear Regression

Implementation

Environment : Colab, Tensorflow 2

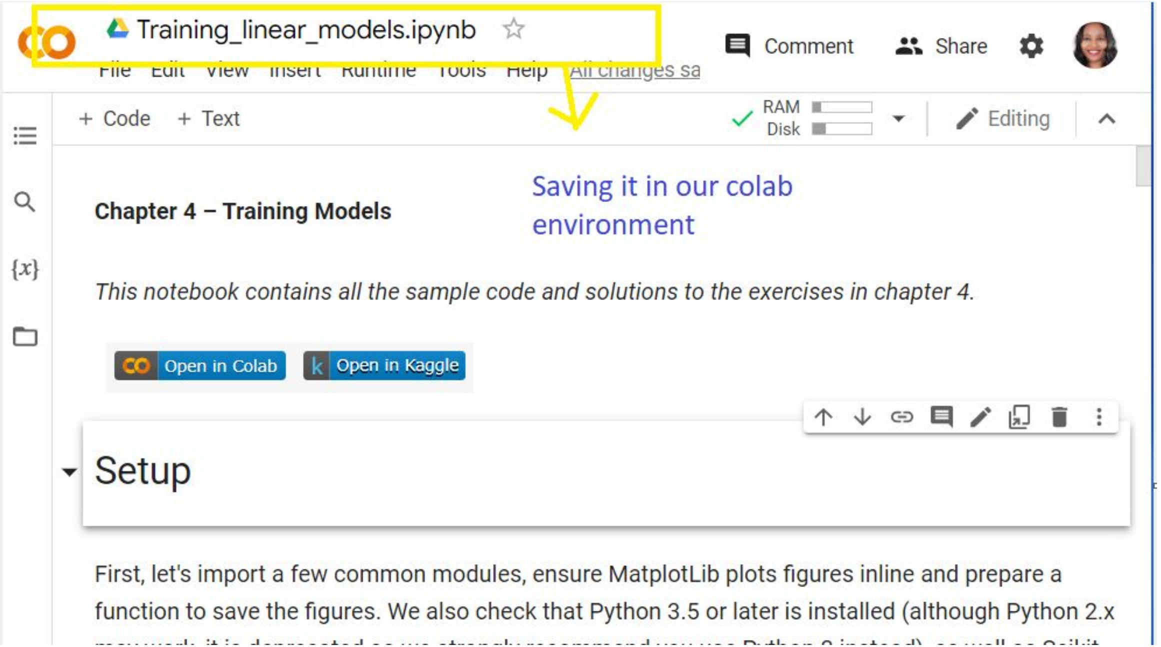
Programming Language: Python

Libraries

```
import matplotlib.pyplot as plt
import numpy as np
import sys
import tensorflow as tf
from sklearn import datasets
from tensorflow.python.framework import ops
ops.reset_default_graph()
sess = tf.Session()
iris = datasets.load_iris()
```

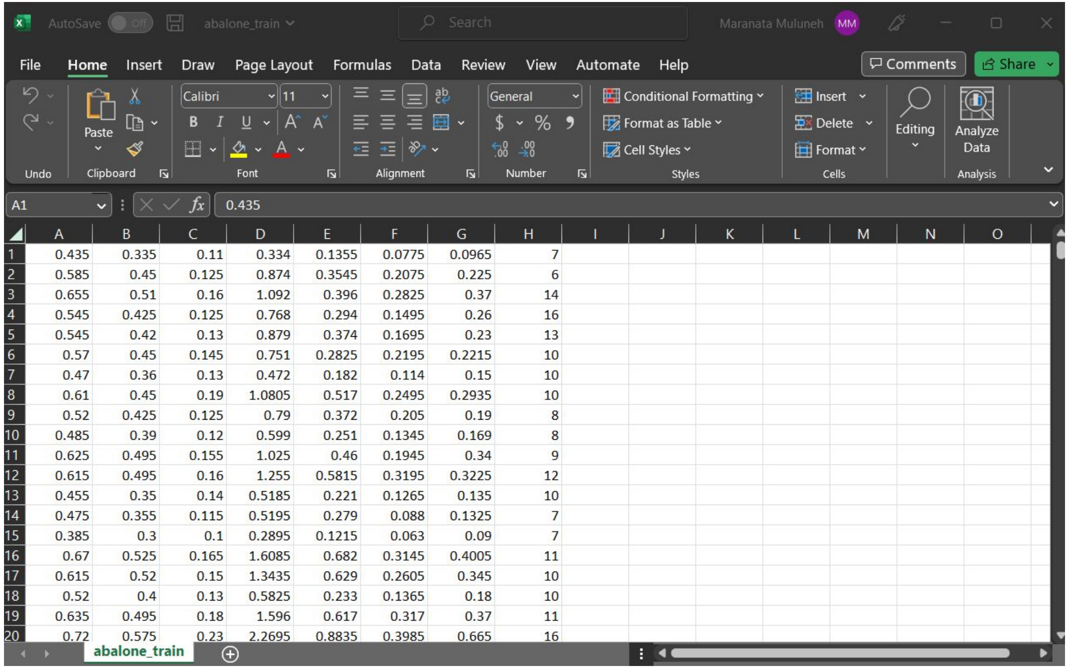
Implementation

Opening the sample code in colab, saving it and running it .



Implementation

Save the csv file (abalone_train.cvs) to a local drive



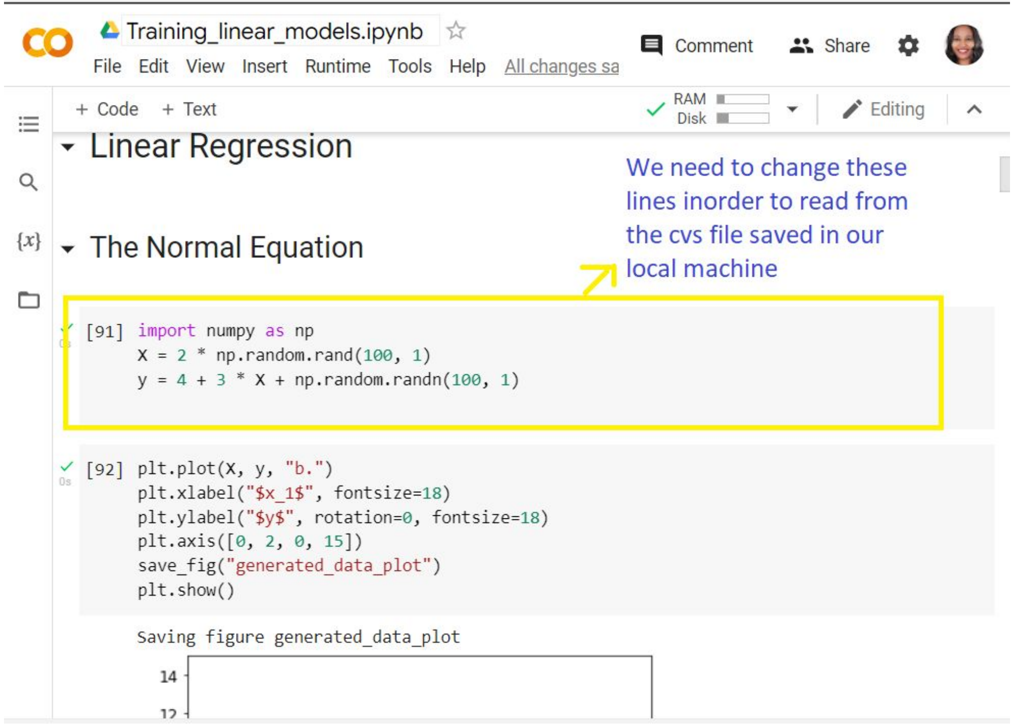
	A	B	C	D	E	F	G	H
1	0.435	0.335	0.11	0.334	0.1355	0.0775	0.0965	7
2	0.585	0.45	0.125	0.874	0.3545	0.2075	0.225	6
3	0.655	0.51	0.16	1.092	0.396	0.2825	0.37	14
4	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16
5	0.545	0.42	0.13	0.879	0.374	0.1695	0.23	13
6	0.57	0.45	0.145	0.751	0.2825	0.2195	0.2215	10
7	0.47	0.36	0.13	0.472	0.182	0.114	0.15	10
8	0.61	0.45	0.19	1.0805	0.517	0.2495	0.2935	10
9	0.52	0.425	0.125	0.79	0.372	0.205	0.19	8
10	0.485	0.39	0.12	0.599	0.251	0.1345	0.169	8
11	0.625	0.495	0.155	1.025	0.46	0.1945	0.34	9
12	0.615	0.495	0.16	1.255	0.5815	0.3195	0.3225	12
13	0.455	0.35	0.14	0.5185	0.221	0.1265	0.135	10
14	0.475	0.355	0.115	0.5195	0.279	0.088	0.1325	7
15	0.385	0.3	0.1	0.2895	0.1215	0.063	0.09	7
16	0.67	0.525	0.165	1.6085	0.682	0.3145	0.4005	11
17	0.615	0.52	0.15	1.3435	0.629	0.2605	0.345	10
18	0.52	0.4	0.13	0.5825	0.233	0.1365	0.18	10
19	0.635	0.495	0.18	1.596	0.617	0.317	0.37	11
20	0.72	0.575	0.23	2.2695	0.8835	0.3985	0.665	16

Implementation

Instead of reading random data

We are going to use the csv file.

We are going to use the csv file.

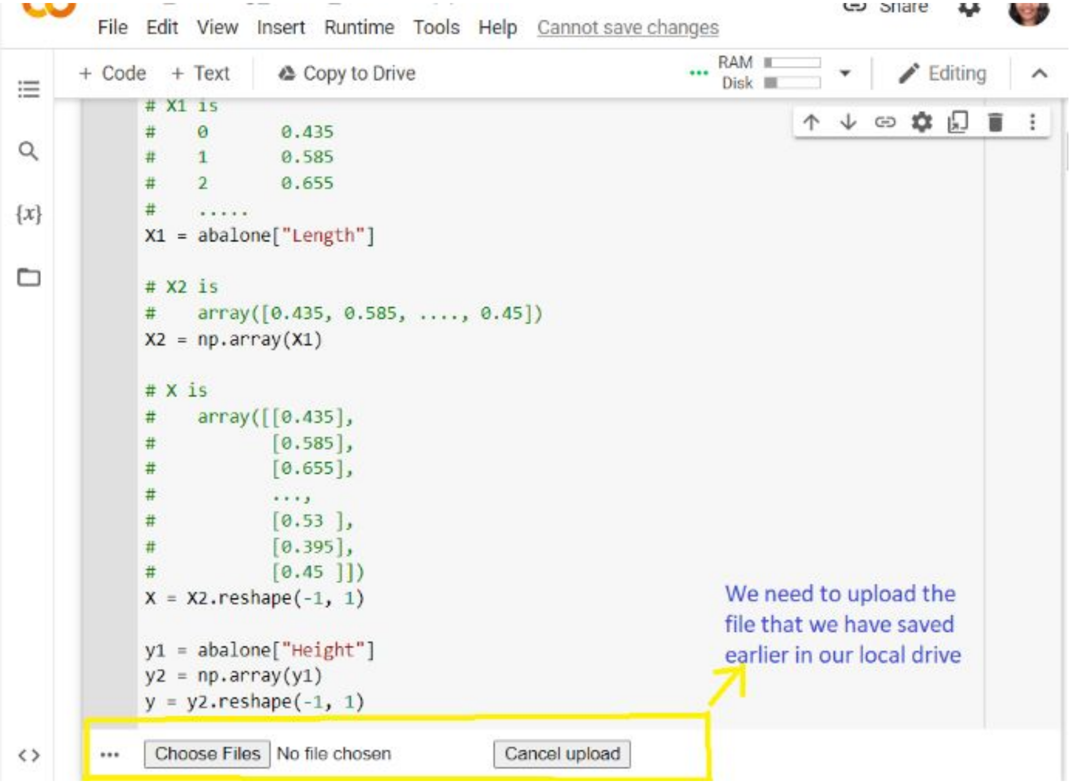


Implementation

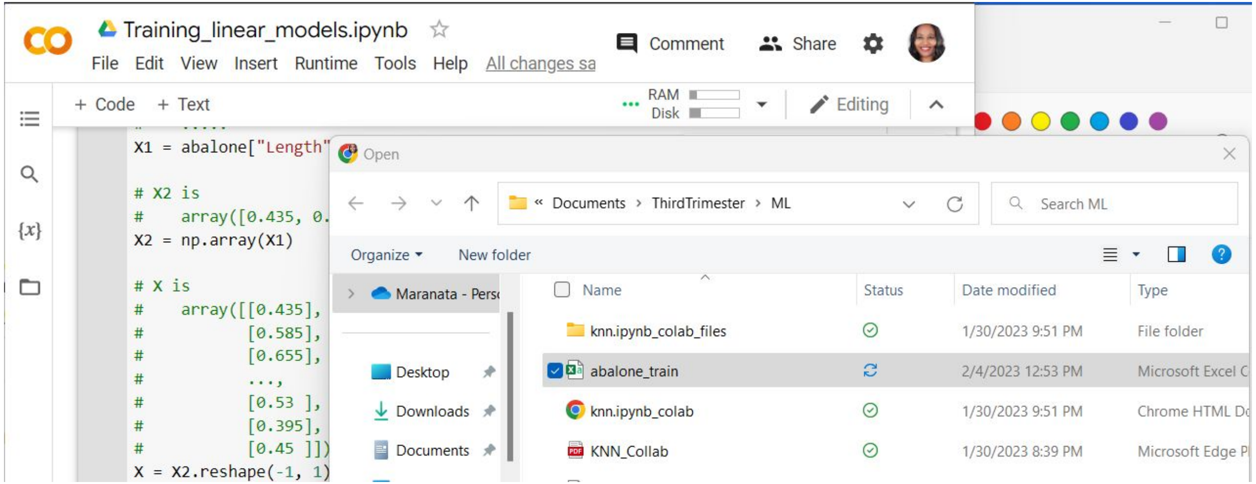
```
import numpy as np
import pandas as pd
# X = 2 * np.random.rand(100, 1)
# y = 4 + 3 * X + np.random.randn(100, 1)
from google.colab import files
uploaded  = files.upload()
import io
abalone = pd.read_csv(
    io.BytesIO(uploaded['abalone_train.csv']),
    names=["Length", "Diameter", "Height",
"Whole weight", "Shucked weight",
           "Viscera weight", "Shell weight",
"Age"])
# X1 is
#    0       0.435
#    1       0.585
#    2       0.655
#
```

```
X1 = abalone["Length"]
# X2 is
#    array([0.435, 0.585, ...., 0.45])
X2 = np.array(X1)
# X is
#    array([[0.435],
#           [0.585],
#           [0.655],
#           ...,
#           [0.53 ],
#           [0.395],
#           [0.45 ]])
X = X2.reshape(-1, 1)
y1 = abalone["Height"]
y2 = np.array(y1)
y = y2.reshape(-1, 1)
```

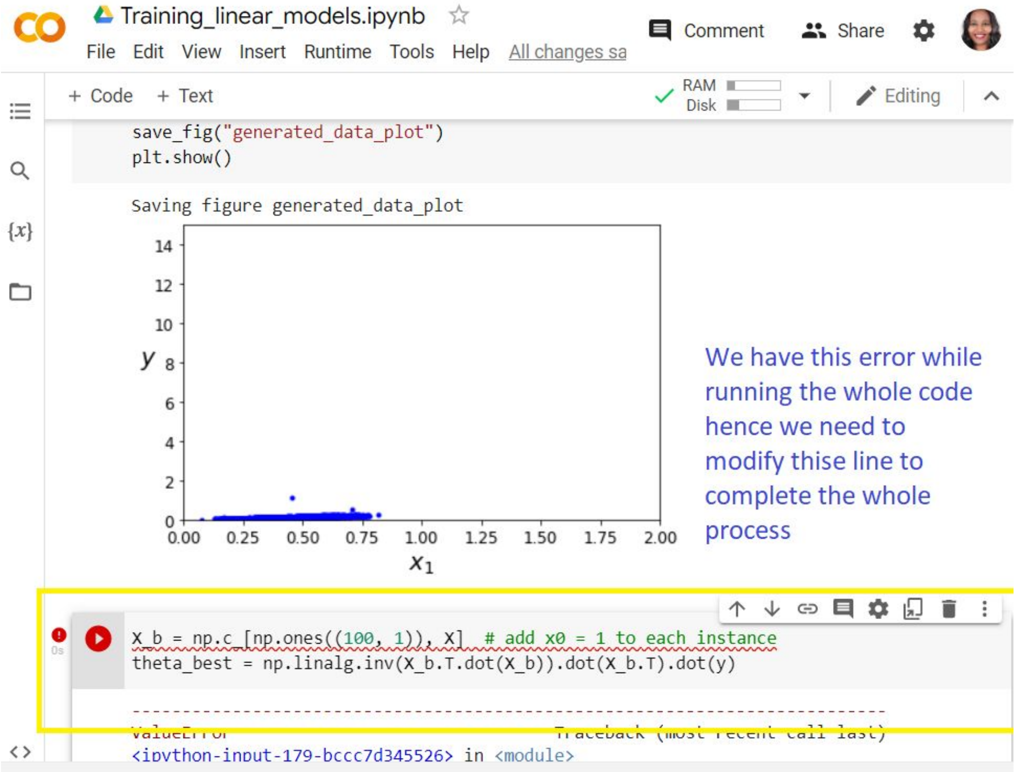
Implementation



Implementation

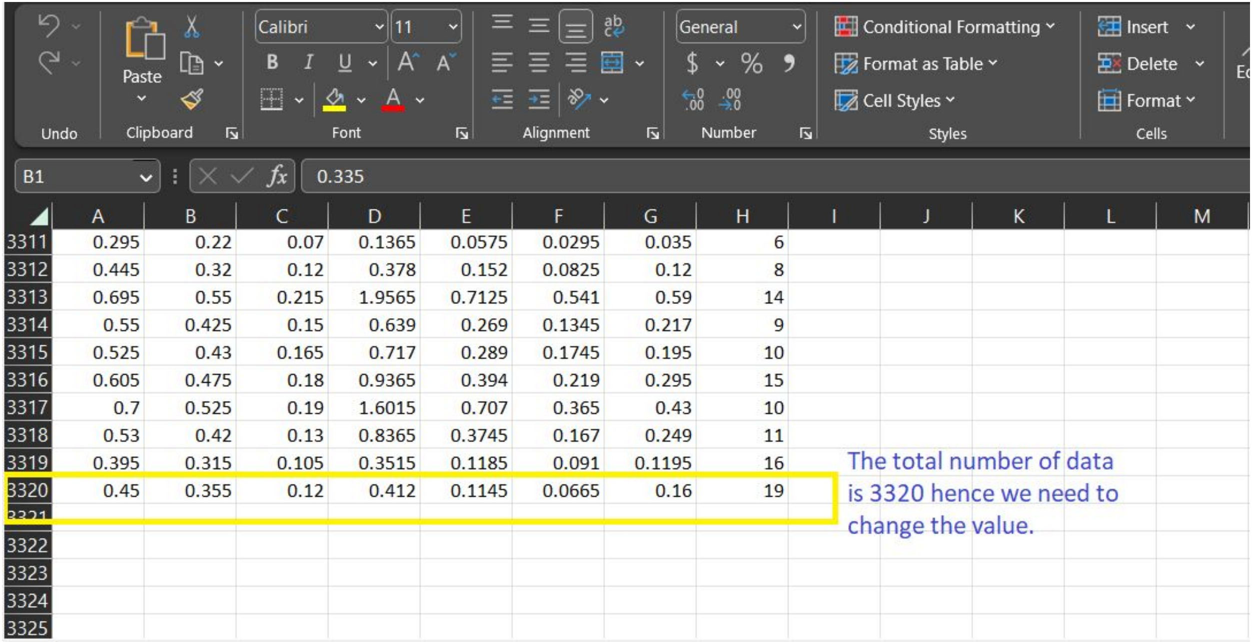


Implementation



We have this error while running the whole code hence we need to modify thise line to complete the whole process

Implementation



	A	B	C	D	E	F	G	H
3311	0.295	0.22	0.07	0.1365	0.0575	0.0295	0.035	6
3312	0.445	0.32	0.12	0.378	0.152	0.0825	0.12	8
3313	0.695	0.55	0.215	1.9565	0.7125	0.541	0.59	14
3314	0.55	0.425	0.15	0.639	0.269	0.1345	0.217	9
3315	0.525	0.43	0.165	0.717	0.289	0.1745	0.195	10
3316	0.605	0.475	0.18	0.9365	0.394	0.219	0.295	15
3317	0.7	0.525	0.19	1.6015	0.707	0.365	0.43	10
3318	0.53	0.42	0.13	0.8365	0.3745	0.167	0.249	11
3319	0.395	0.315	0.105	0.3515	0.1185	0.091	0.1195	16
3320	0.45	0.355	0.12	0.412	0.1145	0.0665	0.16	19
3321								
3322								
3323								
3324								
3325								

Implementation

After we changed the value of the size we have the complete process running.

References

- Google. (n.d.). Google colaboratory. Google Colab. Retrieved February 4, 2023, from https://colab.research.google.com/github/ageron/handson-ml2/blob/master/04_training_linear_models.ipynb
- McClure, N. (n.d.). Tensorflow machine learning cookbook. O'Reilly Online Learning. Retrieved February 4, 2023, from https://learning.oreilly.com/library/view/tensorflow-machine-learning/9781786462169/ch03s04.html