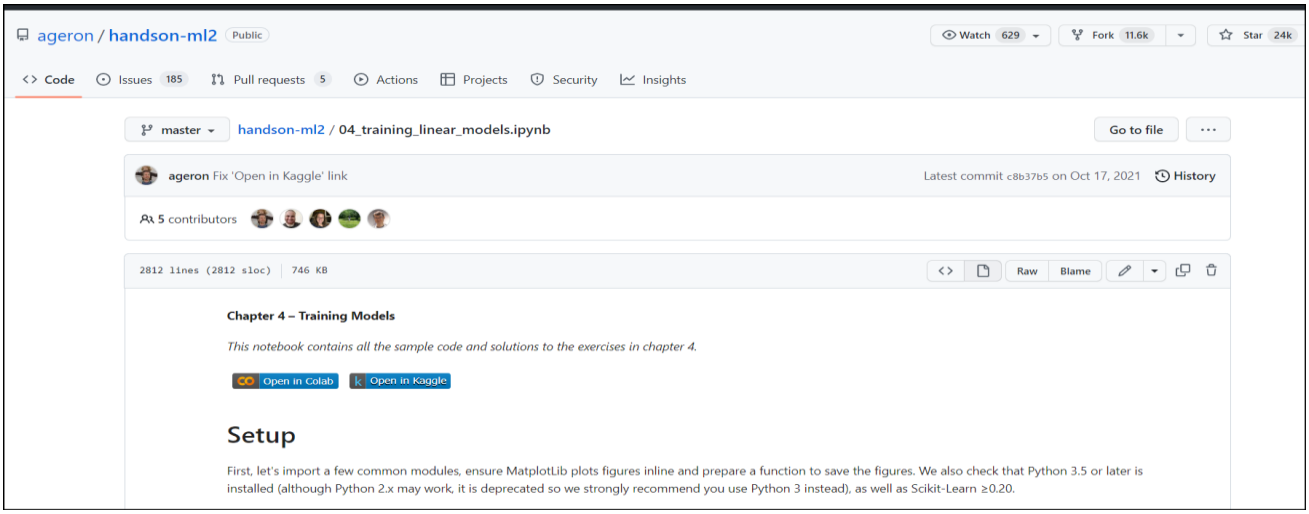
Student Name: Mahmud Omer ID: 19660

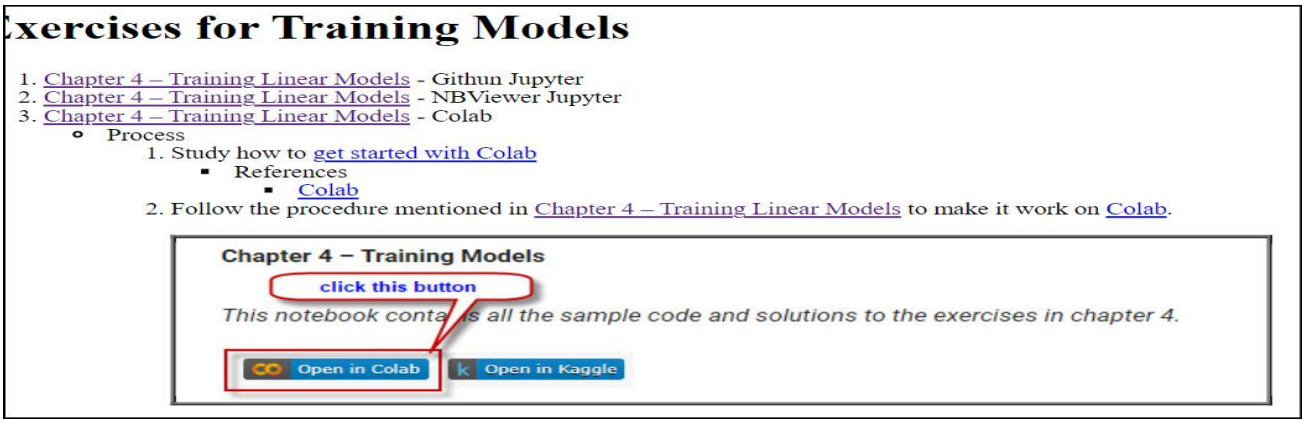
Week2_hw2

Machine Learning - Supervised Learning Linear Regression using Normal Equation

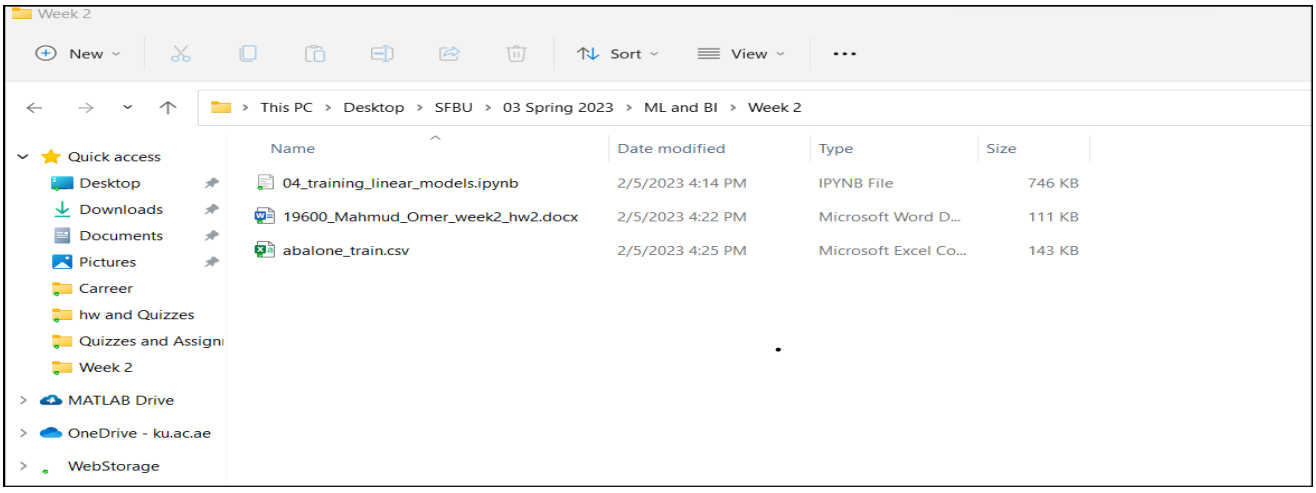
https://hc.labnet.sfbu.edu/~henry/sfbu/course/data_science/algorithm/slide/exercise_algorithm.html

Q28: Jupyter: Training Linear Models

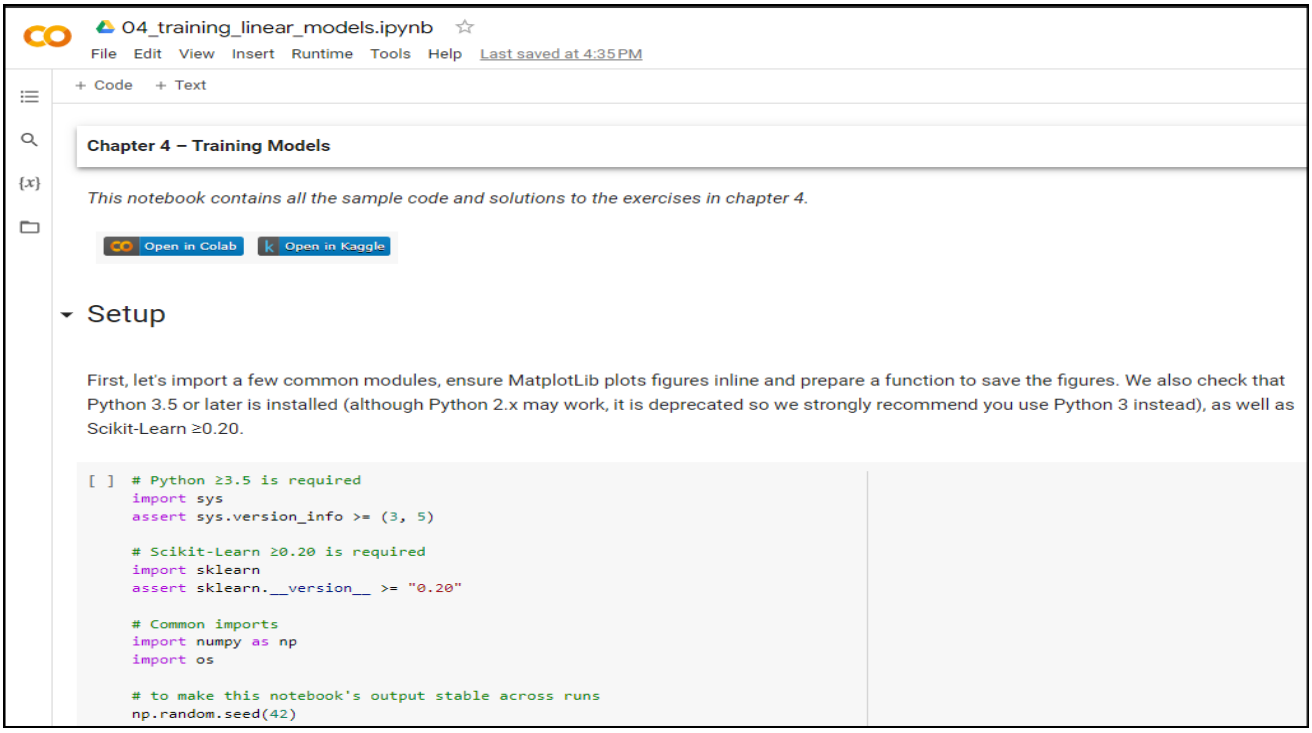




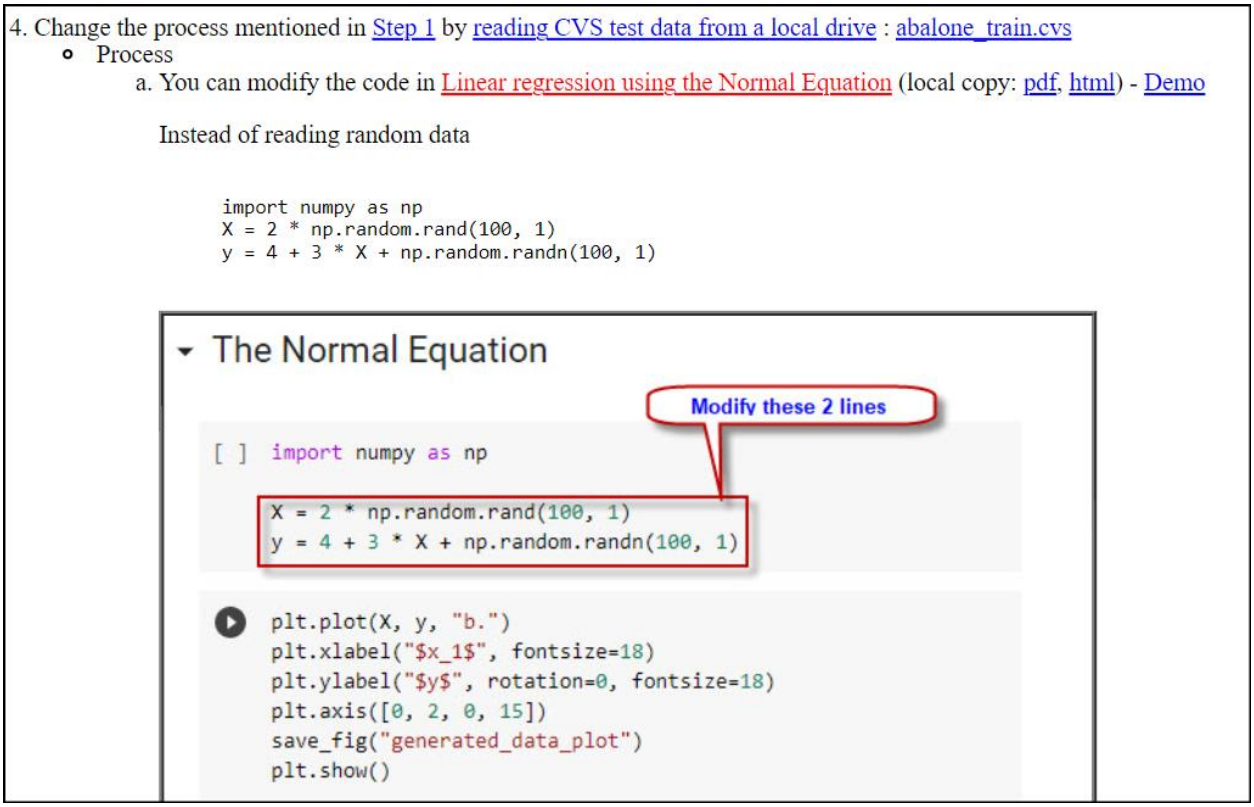
Importing files from local disc to google Colab



Setting up:



Modifying original code in linear regression using Normal equation:



```
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)
```

Choose Files No file chosen Cancel upload

Uploading csv file from local drive to google colab and testing the code

```
Choose Files abalone_train.csv
• abalone_train.csv(text/csv) - 145915 bytes, last modified: 2/5/2023 - 100% done
Saving abalone_train.csv to abalone_train.csv
```

```
[7]  X_b = np.c_[np.ones((3320, 1)), X]  # add x0 = 1 to each instance
     theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
```

Updating code by changing 100 to 3320

```
[7]  X_b = np.c_[np.ones((3320, 1)), X]  # add x0 = 1 to each instance
     theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
```

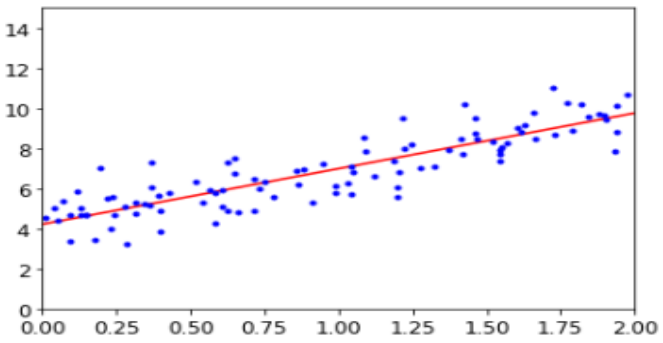
```
[ ]  theta_best
```

```
array([[4.21509616],
       [2.77011339]])
```

```
[ ]  X_new = np.array([[0], [2]])
     X_new_b = np.c_[np.ones((2, 1)), X_new]  # add x0 = 1 to each instance
     y_predict = X_new_b.dot(theta_best)
     y_predict
```

```
array([[4.21509616],
       [9.75532293]])
```

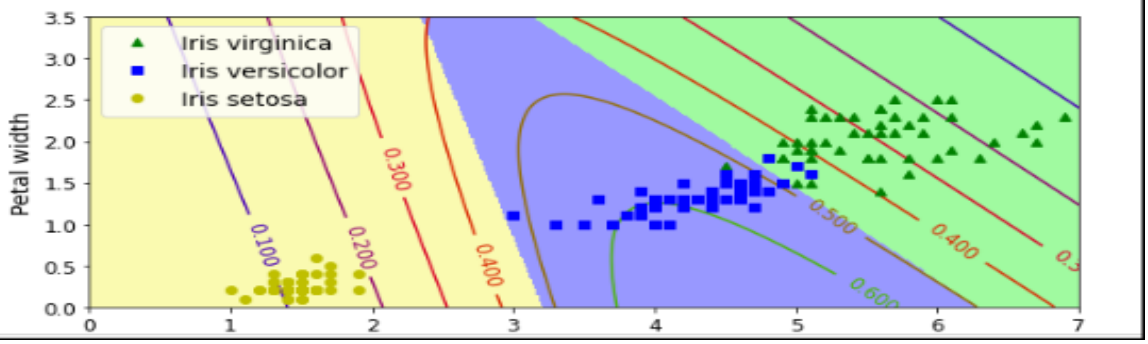
```
plt.plot(X_new, y_predict, "r-")
plt.plot(X, y, "b.")
plt.axis([0, 2, 0, 15])
plt.show()
```



We can also update the code by specifying the number of rows as follows to solve the initially generated error.

```
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"],nrows = 100)
```

```
[93] x0, x1 = np.meshgrid(
             np.linspace(0, 8, 500).reshape(-1, 1),
             np.linspace(0, 3.5, 200).reshape(-1, 1),
         )
     X_new = np.c_[x0.ravel(), x1.ravel()]
     X_new_with_bias = np.c_[np.ones([len(X_new), 1]), X_new]

     logits = X_new_with_bias.dot(Theta)
     Y_proba = softmax(logits)
     y_predict = np.argmax(Y_proba, axis=1)

     zz1 = Y_proba[:, 1].reshape(x0.shape)
     zz = y_predict.reshape(x0.shape)

     plt.figure(figsize=(10, 4))
     plt.plot(X[y==2, 0], X[y==2, 1], "g^", label="Iris virginica")
     plt.plot(X[y==1, 0], X[y==1, 1], "bs", label="Iris versicolor")
     plt.plot(X[y==0, 0], X[y==0, 1], "yo", label="Iris setosa")

     from matplotlib.colors import ListedColormap
     custom_cmap = ListedColormap(['#fafab0','#9898ff','#a0faa0'])

     plt.contourf(x0, x1, zz, cmap=custom_cmap)
     contour = plt.contour(x0, x1, zz1, cmap=plt.cm.brg)
     plt.clabel(contour, inline=1, fontsize=12)
     plt.xlabel("Petal length", fontsize=14)
     plt.ylabel("Petal width", fontsize=14)
     plt.legend(loc="upper left", fontsize=14)
     plt.axis([0, 7, 0, 3.5])
     plt.show()
```



End of code running successfully, showing code is working as desired

```
[94] logits = X_test.dot(Theta)
     Y_proba = softmax(logits)
     y_predict = np.argmax(Y_proba, axis=1)

     accuracy_score = np.mean(y_predict == y_test)
     accuracy_score

     0.9333333333333333
```

Our perfect model turns out to have slight imperfections. This variability is likely due to the very small size of the dataset: depending on how you sample the training set, validation set and the test set, you can get quite different results. Try changing the random seed and running the code again a few times, you will see that the results will vary.

```
[94]
```

Google slides and GitHub links

https://docs.google.com/presentation/d/1CAznmiCZUZ8I4RJ0WW1k1NlIWfW5ahN8WRBblzhybp0/edit?usp=sharing

https://github.com/momer22/Machine-Learning---Supervised-Learning---Linear-Regression-using-Normal-Equation