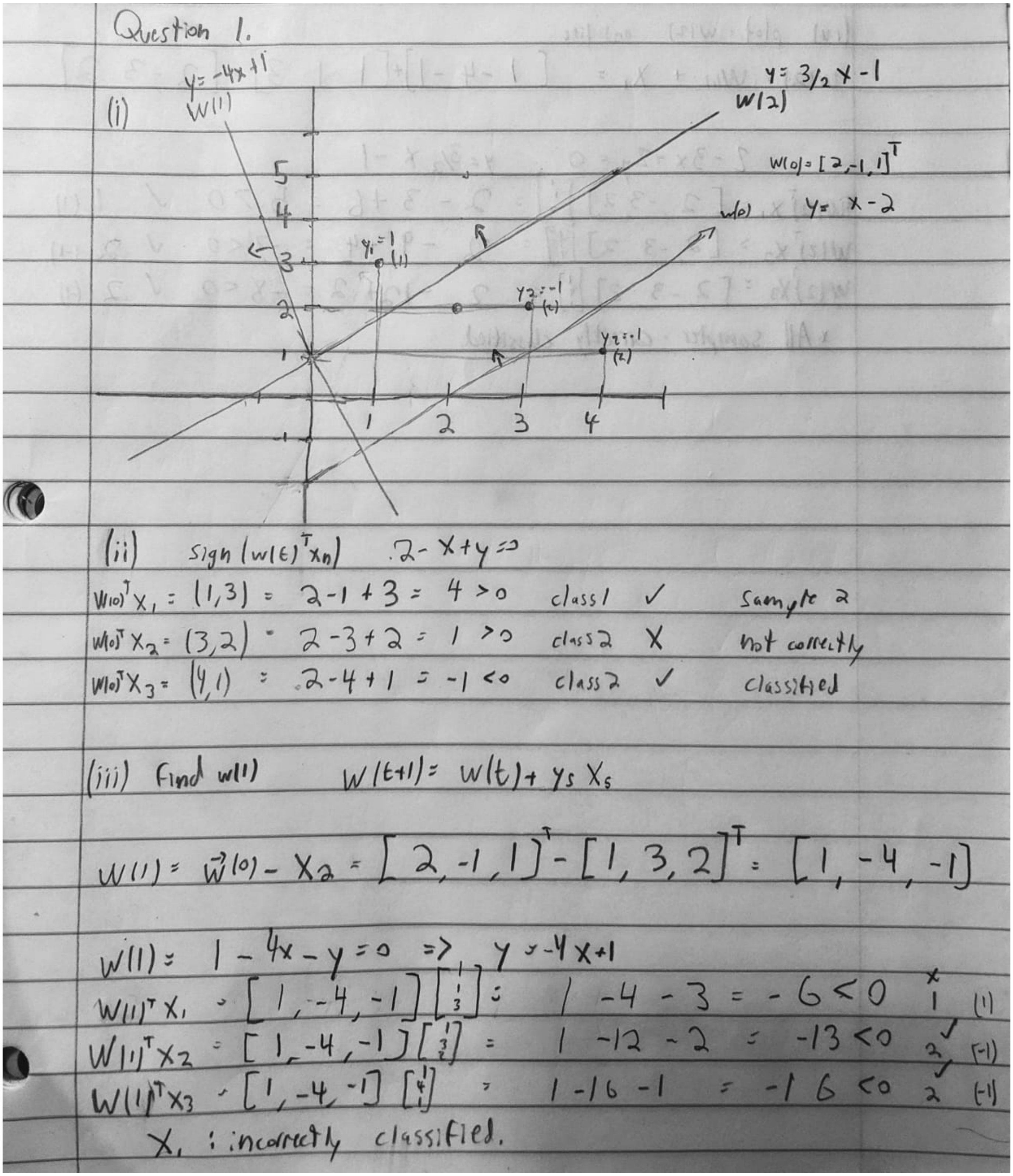
Question 1:



Question 1.

(i)

y = -4x+1
W(1)
y = 3/2 x - 1
W(2)
W(0) = [2,-1,1]^T
w(0) y = x - 2
5
4
3
2
1
-1
1
2
3
4
y_1 = 1
(1)
y_2 = -1
(2)
y_2 = -1
(2)

(ii) $\text{sign}(w(t)^T x_n)$ $2 - x + y =>$

$W(0)^T x_1 = (1,3) = 2 - 1 + 3 = 4 > 0$ class 1 ✓ sample 2

$W(0)^T x_2 = (3,2) \cdot 2 - 3 + 2 = 1 > 0$ class 2 ✗ not correctly

$W(0)^T x_3 = (4,1) = 2 - 4 + 1 = -1 < 0$ class 2 ✓ classified

(iii) Find w(1) $W(t+1) = w(t) + y_s X_s$

$$W(1) = \vec{w}(0) - X_2 = [2,-1,1]^T - [1,3,2]^T = [1,-4,-1]$$

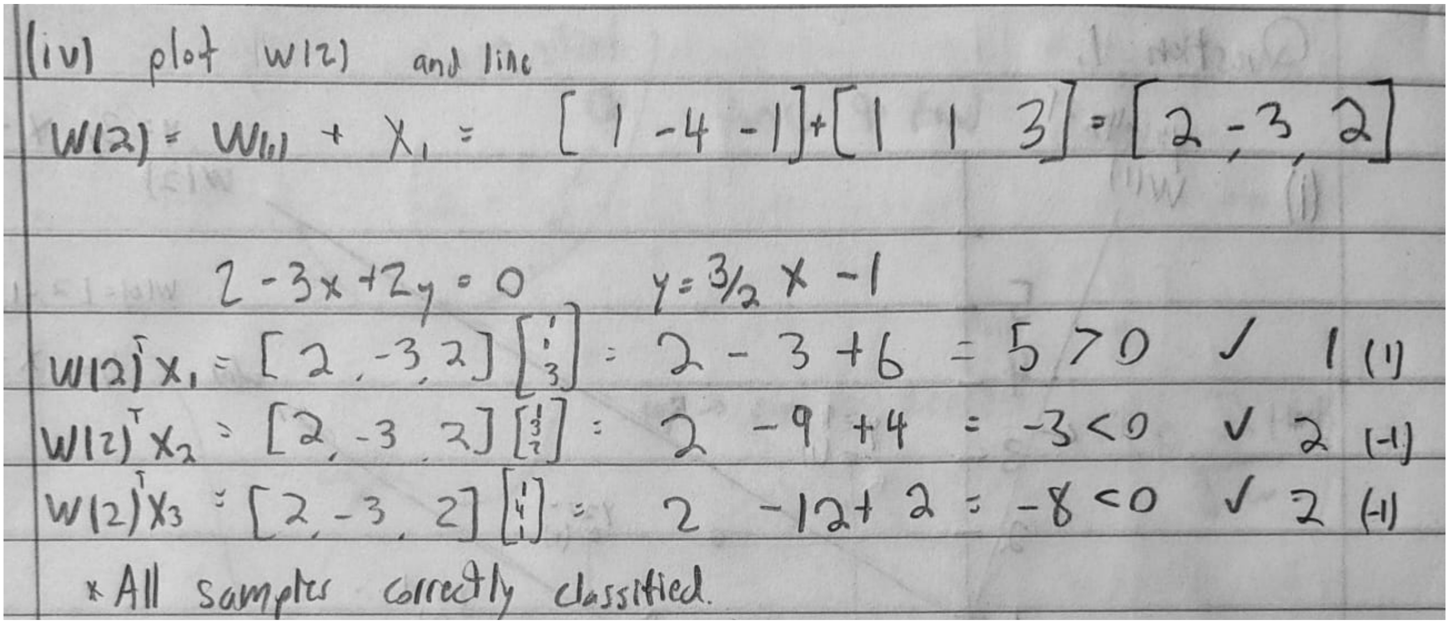
$W(1) = 1 - 4x - y = 0 \Rightarrow y = -4x + 1$

$W(1)^T x_1 = [1,-4,-1]\begin{bmatrix}1\\1\\3\end{bmatrix} = 1 - 4 - 3 = -6 < 0$ ✗ 1 (1)

$W(1)^T x_2 = [1,-4,-1]\begin{bmatrix}1\\3\\2\end{bmatrix} = 1 - 12 - 2 = -13 < 0$ ✓ 2 (-1)

$W(1)^T x_3 = [1,-4,-1]\begin{bmatrix}1\\4\\1\end{bmatrix} = 1 - 16 - 1 = -16 < 0$ ✓ 2 (-1)

$X_1$ : incorrectly classified.



(iv) plot W(2) and line

$W(2) = W(1) + X_1 = [1\ -4\ -1] + [1\ 1\ 3] = [2, -3, 2]$

$2 - 3x + 2y = 0 \qquad y = 3/2\, x - 1$

$W(2)^T X_1 = [2, -3, 2] \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} = 2 - 3 + 6 = 5 > 0$ ✓ 1 (1)

$W(2)^T X_2 = [2, -3, 2] \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} = 2 - 9 + 4 = -3 < 0$ ✓ 2 (-1)

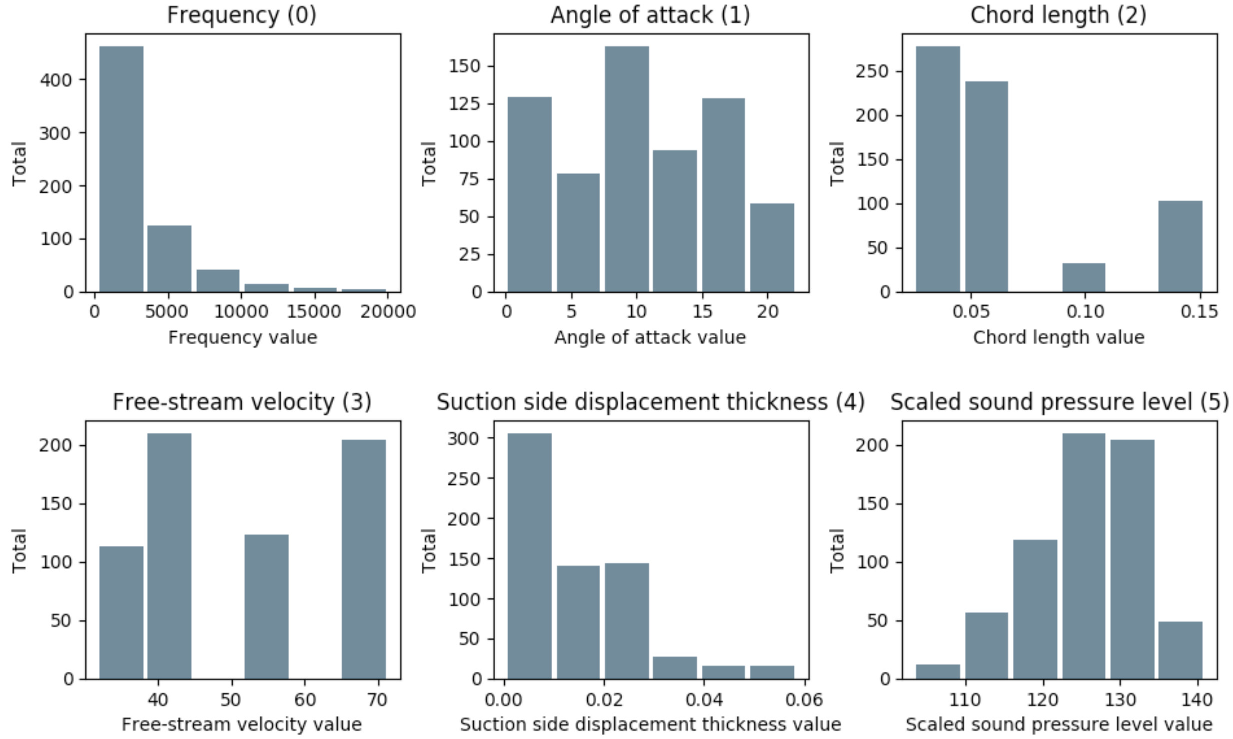
$W(2)^T X_3 = [2, -3, 2] \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix} = 2 - 12 + 2 = -8 < 0$ ✓ 2 (-1)

* All samples correctly classified.

Question 2:

**i)**

Histogram:



Observing the histograms above, we can see how the training data behaves in respect to itself. Features 0, 4, 5, do have a clear pattern, whereas the other features, might not have such a clear indication. The feature's distribution varies between each feature, where some are skewed left, others skewed right, others seem to follow a random distribution.

**ii)**

**(on code)**

The dimensionality of the weights: 6 x 1

```
--------- w ------------
[[ 1.32303685e+02]
 [-7.11322380e-04]
 [-4.54774159e-01]
 [-5.23017057e+01]
 [ 8.39826774e-02]
 [-9.14287813e+01]]
```

**iii)**

```
----------- RSS Train ----------
[[18033.32549924]]

----------- RSS Test ----------
[[5136.39637151]]
```

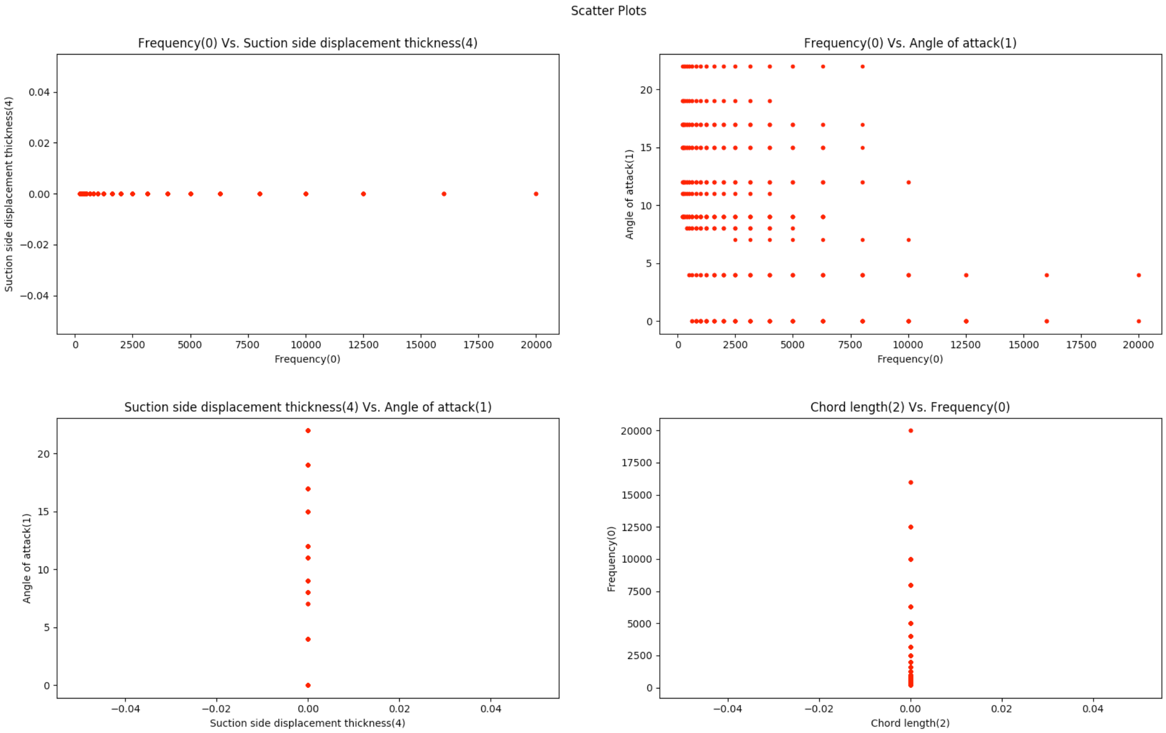
**iv)**

**--------- Extra Credit ------------**

*got 3 random features*

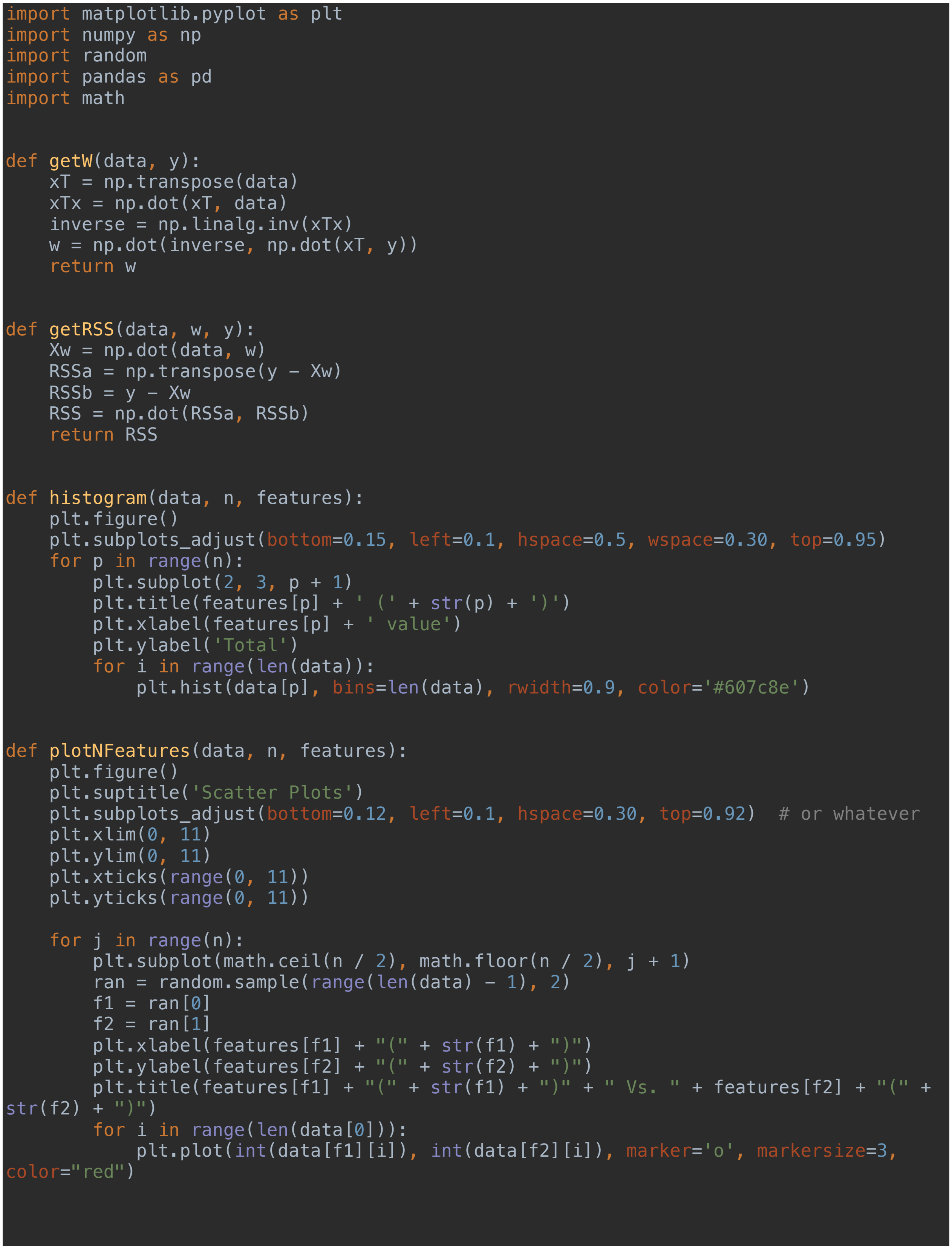
```
f1 = 3
f2  = 4
f3 = 0

--------- w ------------
[[ 1.30031399e+02]
 [ 6.92981952e-02]
 [-5.94872936e+01]
 [-4.50649292e-01]]
----------- RSS ----------
[[21330.4436531]]
----------- RSS Test ----------
[[12166.2178024]]
```

I observed that the Residual sum of squares is a lot lower when using a combination of three features as compared to the five. This is clear since we see the discrepancy between the data and the estimated model calculated between less data points, therefore less room for vagueness within the model.

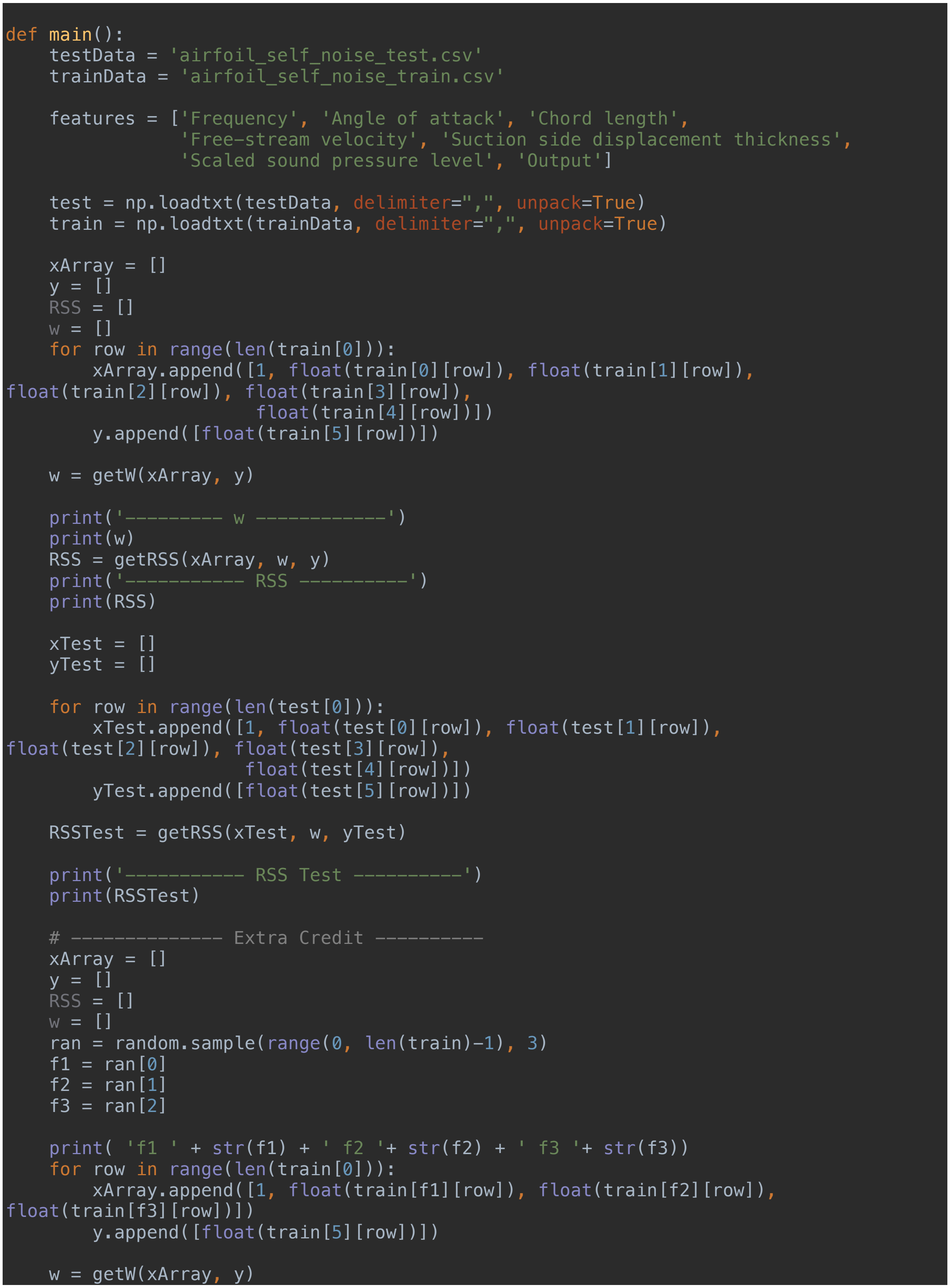


Plotting and comparing two features, it is challenging to predict where things would be, so the RSS is still relatively higher than we would like it to be but it does lower with the less features we compare.

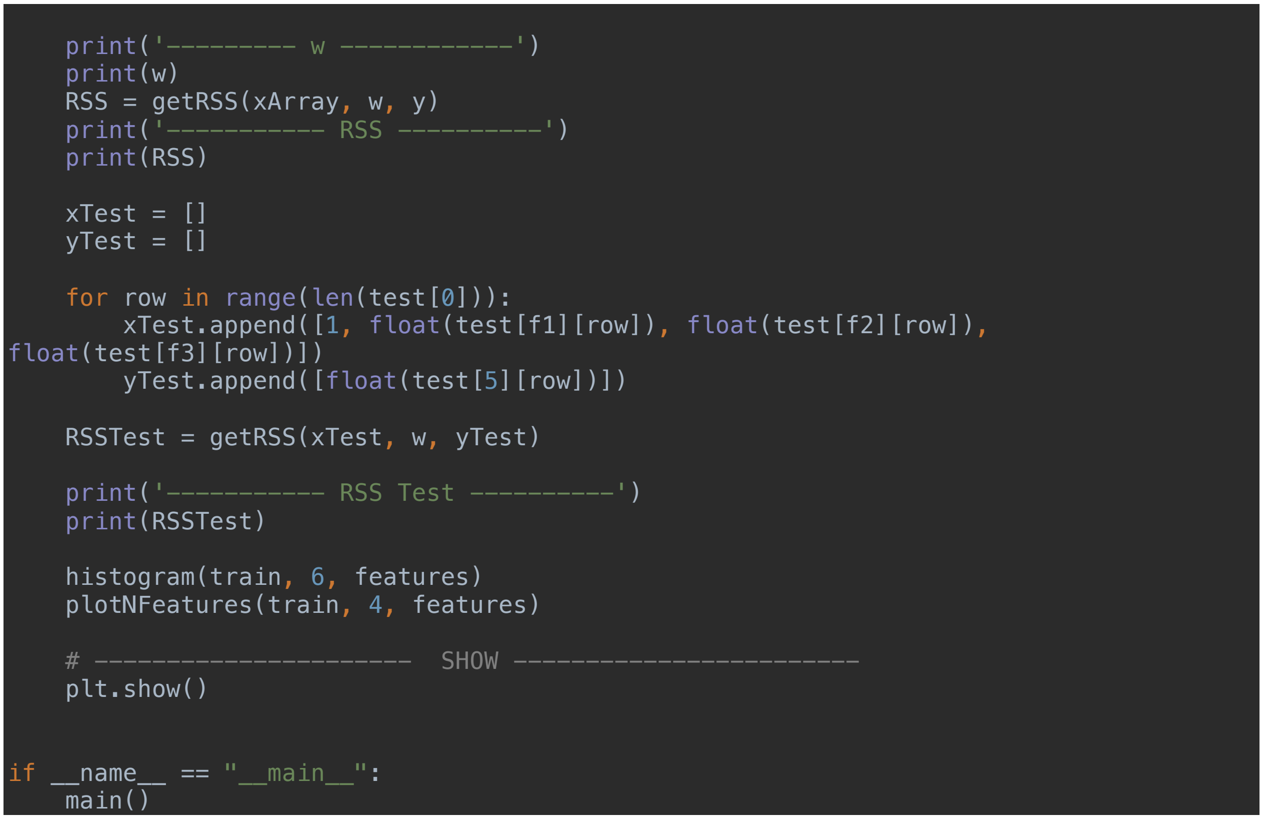
# CODE



```
import matplotlib.pyplot as plt
import numpy as np
import random
import pandas as pd
import math


def getW(data, y):
    xT = np.transpose(data)
    xTx = np.dot(xT, data)
    inverse = np.linalg.inv(xTx)
    w = np.dot(inverse, np.dot(xT, y))
    return w


def getRSS(data, w, y):
    Xw = np.dot(data, w)
    RSSa = np.transpose(y - Xw)
    RSSb = y - Xw
    RSS = np.dot(RSSa, RSSb)
    return RSS


def histogram(data, n, features):
    plt.figure()
    plt.subplots_adjust(bottom=0.15, left=0.1, hspace=0.5, wspace=0.30, top=0.95)
    for p in range(n):
        plt.subplot(2, 3, p + 1)
        plt.title(features[p] + ' (' + str(p) + ')')
        plt.xlabel(features[p] + ' value')
        plt.ylabel('Total')
        for i in range(len(data)):
            plt.hist(data[p], bins=len(data), rwidth=0.9, color='#607c8e')


def plotNFeatures(data, n, features):
    plt.figure()
    plt.suptitle('Scatter Plots')
    plt.subplots_adjust(bottom=0.12, left=0.1, hspace=0.30, top=0.92)  # or whatever
    plt.xlim(0, 11)
    plt.ylim(0, 11)
    plt.xticks(range(0, 11))
    plt.yticks(range(0, 11))

    for j in range(n):
        plt.subplot(math.ceil(n / 2), math.floor(n / 2), j + 1)
        ran = random.sample(range(len(data) - 1), 2)
        f1 = ran[0]
        f2 = ran[1]
        plt.xlabel(features[f1] + "(" + str(f1) + ")")
        plt.ylabel(features[f2] + "(" + str(f2) + ")")
        plt.title(features[f1] + "(" + str(f1) + ")" + " Vs. " + features[f2] + "(" +
str(f2) + ")")
        for i in range(len(data[0])):
            plt.plot(int(data[f1][i]), int(data[f2][i]), marker='o', markersize=3,
color="red")
```



```
def main():
    testData = 'airfoil_self_noise_test.csv'
    trainData = 'airfoil_self_noise_train.csv'

    features = ['Frequency', 'Angle of attack', 'Chord length',
                'Free-stream velocity', 'Suction side displacement thickness',
                'Scaled sound pressure level', 'Output']

    test = np.loadtxt(testData, delimiter=",", unpack=True)
    train = np.loadtxt(trainData, delimiter=",", unpack=True)

    xArray = []
    y = []
    RSS = []
    w = []
    for row in range(len(train[0])):
        xArray.append([1, float(train[0][row]), float(train[1][row]),
float(train[2][row]), float(train[3][row]),
                       float(train[4][row])])
        y.append([float(train[5][row])])

    w = getW(xArray, y)

    print('---------- w -------------')
    print(w)
    RSS = getRSS(xArray, w, y)
    print('------------ RSS -----------')
    print(RSS)

    xTest = []
    yTest = []

    for row in range(len(test[0])):
        xTest.append([1, float(test[0][row]), float(test[1][row]),
float(test[2][row]), float(test[3][row]),
                      float(test[4][row])])
        yTest.append([float(test[5][row])])

    RSSTest = getRSS(xTest, w, yTest)

    print('------------ RSS Test -----------')
    print(RSSTest)

    # --------------- Extra Credit -----------
    xArray = []
    y = []
    RSS = []
    w = []
    ran = random.sample(range(0, len(train)-1), 3)
    f1 = ran[0]
    f2 = ran[1]
    f3 = ran[2]

    print( 'f1 ' + str(f1) + ' f2 '+ str(f2) + ' f3 '+ str(f3))
    for row in range(len(train[0])):
        xArray.append([1, float(train[f1][row]), float(train[f2][row]),
float(train[f3][row])])
        y.append([float(train[5][row])])

    w = getW(xArray, y)
```



```
    print('--------- w ------------')
    print(w)
    RSS = getRSS(xArray, w, y)
    print('----------- RSS ----------')
    print(RSS)

    xTest = []
    yTest = []

    for row in range(len(test[0])):
        xTest.append([1, float(test[f1][row]), float(test[f2][row]),
float(test[f3][row])])
        yTest.append([float(test[5][row])])

    RSSTest = getRSS(xTest, w, yTest)

    print('----------- RSS Test ----------')
    print(RSSTest)

    histogram(train, 6, features)
    plotNFeatures(train, 4, features)

    # ----------------------  SHOW ------------------------
    plt.show()


if __name__ == "__main__":
    main()
```