```
import math
import pandas as pd
```

Candace Edwards

ICS 635: Homework 3

Part 1: Notebook Link

Part 2: Q5: Notebook Link

Question 1: Regularization

$$\boldsymbol{\theta}_A = \begin{bmatrix} 0.42 \\ 2.85 \\ 11.21 \end{bmatrix}$$

$$\boldsymbol{\theta}_B = \begin{bmatrix} 98.22 \\ -42.11 \\ 12.92 \end{bmatrix}$$

The first run, $\boldsymbol{\theta}_A$, is likely the result of λ=1, because regularization dampened the learned parameter values. When λ=0, the model is only minimizing the loss function, with no regularizaion, so it learned parameters ($\boldsymbol{\theta}_B$) that is likely biased to the training data. When λ=1, the parameters are more generalized.

Question 2: K- Nearest Neighbor

```
def calc_dist(x1_1,x2_1,x1_2,x2_2):
  '''
  Calculate the Euclidian distance between two points
  '''
  distance = math.sqrt((x1_2 - x1_1)**2 + (x2_2 - x2_1)**2)
  return distance


#test (3,4)(6,8) #expected 5
#print(calc_dist(3,4,6,8))


x1 = [0,0,0,5,5,5,10,10,10]
x2 = [0,5,10,0,5,10,0,5,10]
y =[1,1,0,1,1,0,0,0,0]
data = zip(x1,x2)


x3= [calc_dist(d[0],d[1],6,6) for d in data]


df = pd.DataFrame({
    'x1':x1,
    'x2':x2,
    'y':y,
    'distance from (6,6)':x3
})


df
```

	x1	x2	y	distance from (6,6)
0	0	0	1	8.485281
1	0	5	1	6.082763

Using Euclidean distance to calculate the distance between that data and new point (6,6), and assuming the distances are unweighted/uniform:

- When `k=1`, (6,6) would be classified as `y=1`, because it's closest neighbor is point (5,5), which is also classifed as `y=1`.
- When `k=3`, (6,6) would be classifed as `y=0`, because the second/third nearest neighbors (5,10) and (10,5) are classified as `y=0`. The `y=0` label wins by 2/3 majority.

8	10	10	0	5.656854

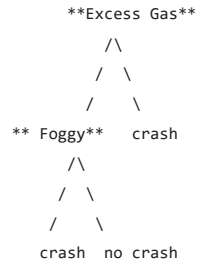
Question 3: Decision Trees

The full (hand-written) calculations for the tree are attached as a pdf. The information gain calculation to determin the root/base of the tree is as follows:

- Foggy: 0.4591
- **Excess Gasoline: 0.7670**
- Runway Too Close: 0
- Too High: 0.0261

From this calculation, Excess Gasoline was selected as the base of the tree. Splitting on Excess Gasoline yielded on pure crash branch, so I did not continue calculations for subset Excess Gasoline = no.

For subset Excess Gasoline = yes, Foggy had an information gain of 1, which naturally indicates the next splitting parameter. Foggy produced two pure nodes foggy = yes = crash, foggy = no = no crash. This is the end of the tree. The other features in the data did not provide additional information gain, so the result is a pre-pruned tree.



Question 4: Support Vector Machines

The equation of the line describing the decision boundary for a hard margin SVM trained on the given dataset is:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$$