Bonus Project 1 (200 Points) (Optional)

In this project you will implement multiclass classification with logistic regression

Requirements:

1. Change the base code given in logistic regression from scratch Python code so that it will work for multiclass classification problems. Recall one-vs-all algorithm
2. Test your code on iris flower dataset (3 classes) from Kaggle for multi class classification problem given in this project folder

Note: You can use following code as a base to create columns in y (outcome, Species class) for each class. So, the number of columns in y1 (given below) will be equal to the total number classes. That is 3 for iris flower dataset. For instance, for the first column (class is 1 which is Iris-setosa), whichever row in dataset is 'Iris-setosa' will be marked as 1 in y1's corresponding row. Any other class (classes 2 and 3: Iris-versicolor and Iris-virginica) will be marked as 0 in y1's first column.

Recall that as we changed y's representation (from single column) to y1 (3 columns), we need to change theta vector. Since iris flower dataset has 3 classes theta vector will become a matrix with 3 columns where each column corresponds to a different class. So, y1 and new theta matrix size must match. For each one-vs-all experiment, we will have a separate set of theta values for our decision boundary line.

```
for i in range(0, len(y.unique())):
    for j in range(0, len(y1)):
        if y[j] == y.unique()[i]:
            y1.iloc[j, i] = 1 #one vs. all
        else:
            y1.iloc[j, i] = 0 #all others will be 0
```

You can change class categorical values to numerical values as follows:

```
data['Species'].replace(1, ' Iris-setosa ',inplace=True)
data['Species'].replace(2, ' Iris-versicolor',inplace=True)
data['Species'].replace(3, ' Iris-virginica',inplace=True)
```

Deliverables:

1. Your source code including code for all plots (e.g. **YourFullNAme_Project1_CPSC4370.py**). Make sure your code works for iris flower dataset.
2. A report that shows your gradient descent cost function plot, accuracy of your predictions for different learning rates, and different number of iterations.