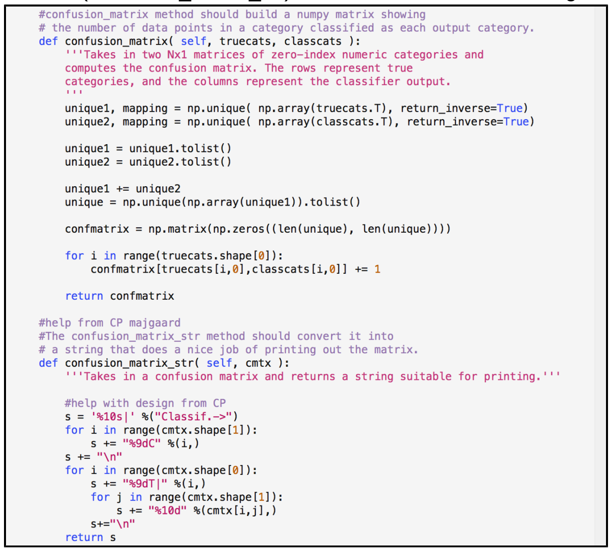
CS251 Proj8

<u>CS251 Project 8 -Machine Learning</u>

The main goal of this project was to build two simple classifiers that can be train from data. We were asked to design and implement a Naive Bayes classifier and a K-nearest neighbor (KNN) classifier. To begin, in the lab we modified our Classifier parent class with two child classes for NB and KNN. Then, during the lab we implemented the code to execute and classify within each classifier file. Then, for the project we were asked to design code to result in a confusion matrix that organizes the number of data points in a category. In addition, two new files that executed our KNN and NB classifications. Finally, we then had to make this work with our GUI.

Code Overview & GUI Implementation:

So, after completing the lab and the code for the two classifier children classes, I began the project by writing two confusion matrix methods: one of the methods (confusion_matrix) builds a numpy matrix showing the number of data points in a category classified as each output -- the second method (confusion_matrix_str) converts the matrix to a string and prints it nicely to the terminal. Here is the code:



```
    #confusion_matrix method should build a numpy matrix showing
    # the number of data points in a category classified as each output category.
    def confusion_matrix( self, truecats, classcats ):
        '''Takes in two Nx1 matrices of zero-index numeric categories and
        computes the confusion matrix. The rows represent true
        categories, and the columns represent the classifier output.
        '''
        unique1, mapping = np.unique( np.array(truecats.T), return_inverse=True)
        unique2, mapping = np.unique( np.array(classcats.T), return_inverse=True)

        unique1 = unique1.tolist()
        unique2 = unique2.tolist()

        unique1 += unique2
        unique = np.unique(np.array(unique1)).tolist()

        confmatrix = np.matrix(np.zeros((len(unique), len(unique))))

        for i in range(truecats.shape[0]):
            confmatrix[truecats[i,0],classcats[i,0]] += 1

        return confmatrix

    #help from CP majgaard
    #The confusion_matrix_str method should convert it into
    # a string that does a nice job of printing out the matrix.
    def confusion_matrix_str( self, cmtx ):
        '''Takes in a confusion matrix and returns a string suitable for printing.'''

        #help with design from CP
        s = '%10s|' %("Classif.->")
        for i in range(cmtx.shape[1]):
            s += "%9dC" %(i,)
        s += "\n"
        for i in range(cmtx.shape[0]):
            s += "%9dT|" %(i,)
            for j in range(cmtx.shape[1]):
                s += "%10d" %(cmtx[i,j],)
            s+="\n"
        return s
```

Next, I created two new files that implemented the classifiers with data sets and into my GUI! The format of both these files followed the same basic structure detailed in the Project8 instructions. The steps for creating the classification file are as follows:

Write a python function, probably in a new file, that does following.

1. **Reads in a training set and its category labels, possibly as a separate file.**
2. **Reads a test set and its category, possibly as a separate file.**
3. **Builds a classifier using the training set.**
4. **Classifies the training set and prints out a confusion matrix.**
5. **Classifies the test set and prints out a confusion matrix.**
6. **Writes out a new CSV data file with the test set data and the categories as an extra column. Your application should be able to read this file and plot it with the categories as colors.**

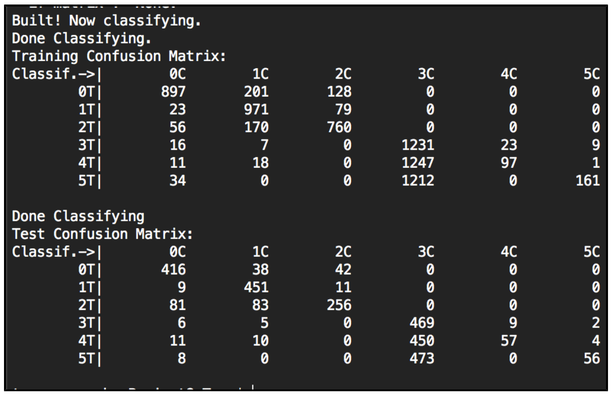
Then, after creating successful classification methods, we were asked to test them on a variety of data sets provided in the Lab and then plot the data in our GUI application. In addition, we were asked to complete this same task on a data set of our choosing.

<u>*Activity Recognition Data Set Screenshots:*</u>

<u>*Naive Bayes Classifier:*</u>

First, start by typing this into terminal: *tonys-new-mbp:Project8 Tony$ python2.7 naivebaye_classifier.py UCI-X-train.csv UCI-X-test.csv UCI-Y-train.csv UCI-Y-test.csv*

Next, here is the resulting confusion-matrix output in terminal:

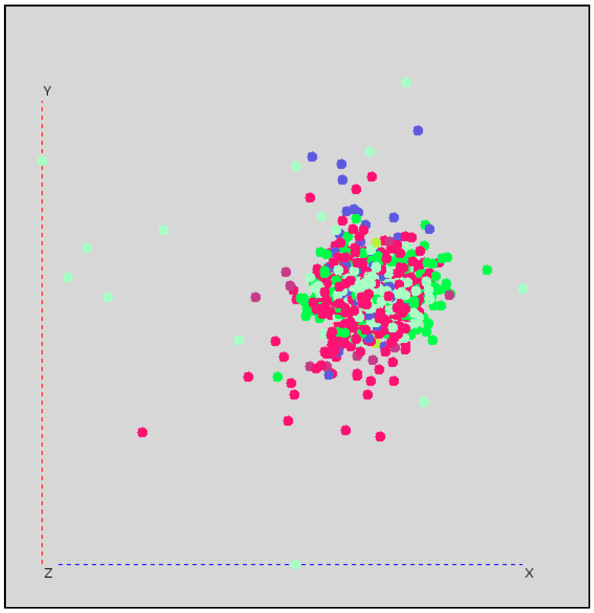


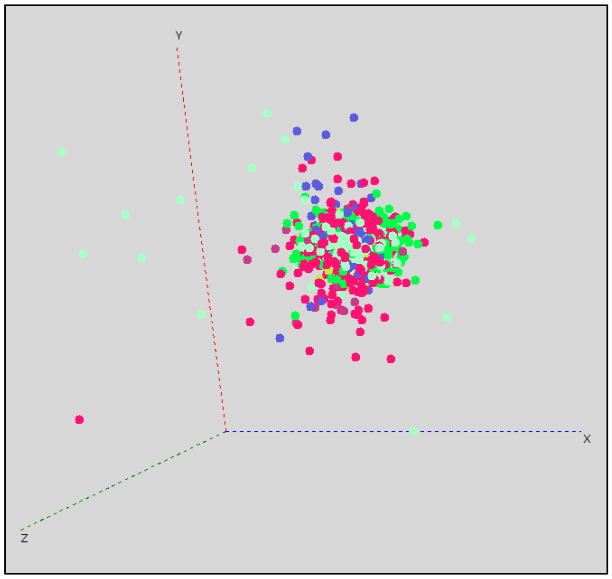
```
Built! Now classifying.
Done Classifying.
Training Confusion Matrix:
Classif.->|        0C        1C        2C        3C        4C        5C
        0T|       897       201       128         0         0         0
        1T|        23       971        79         0         0         0
        2T|        56       170       760         0         0         0
        3T|        16         7         0      1231        23         9
        4T|        11        18         0      1247        97         1
        5T|        34         0         0      1212         0       161

Done Classifying
Test Confusion Matrix:
Classif.->|        0C        1C        2C        3C        4C        5C
        0T|       416        38        42         0         0         0
        1T|         9       451        11         0         0         0
        2T|        81        83       256         0         0         0
        3T|         6         5         0       469         9         2
        4T|        11        10         0       450        57         4
        5T|         8         0         0       473         0        56
```

Finally, this confusion matrix data is then written into a .csv file titled "nbClass.csv"

Now, I ran my display and opened the "nbClass" csv file and plotted the points on my display. Then, I ran Color Fix method to organize the categories by color. Here are the GUI Display Plotted screenshots for the resulting displays:

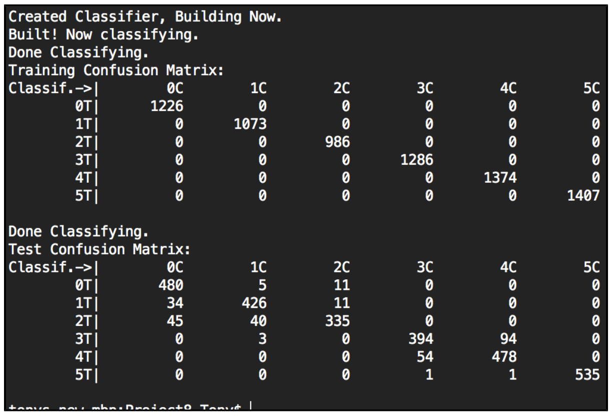




<u>KNN Classifier:</u>

First, start by typing this into terminal: *tonys-new-mbp:Project8 Tony$ python2.7 knn_classifier.py UCI-X-train.csv UCI-X-test.csv UCI-Y-train.csv UCI-Y-test.csv*

Next, here is the resulting confusion-matrix output in terminal:

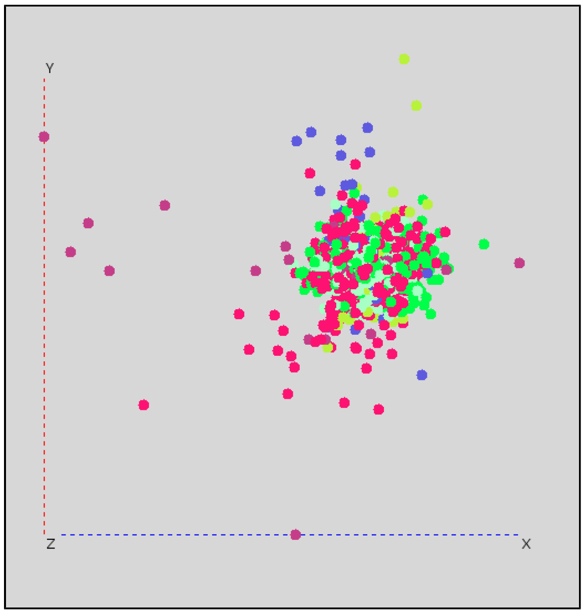


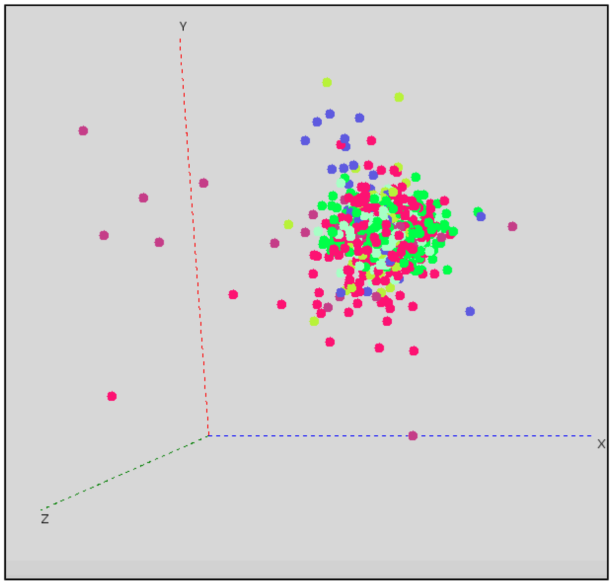
```
Created Classifier, Building Now.
Built! Now classifying.
Done Classifying.
Training Confusion Matrix:
Classif.->|        0C        1C        2C        3C        4C        5C
        0T|      1226         0         0         0         0         0
        1T|         0      1073         0         0         0         0
        2T|         0         0       986         0         0         0
        3T|         0         0         0      1286         0         0
        4T|         0         0         0         0      1374         0
        5T|         0         0         0         0         0      1407

Done Classifying.
Test Confusion Matrix:
Classif.->|        0C        1C        2C        3C        4C        5C
        0T|       480         5        11         0         0         0
        1T|        34       426        11         0         0         0
        2T|        45        40       335         0         0         0
        3T|         0         3         0       394        94         0
        4T|         0         0         0        54       478         0
        5T|         0         0         0         1         1       535
```

Finally, this confusion matrix data is then written into a .csv file titled "knnClass.csv"

Now, I ran my display and opened the "knnClass" csv file and plotted the points on my display. Then, I ran Color Fix method to organize the categories by color. Here are the GUI Display Plotted screenshots for the resulting displays:



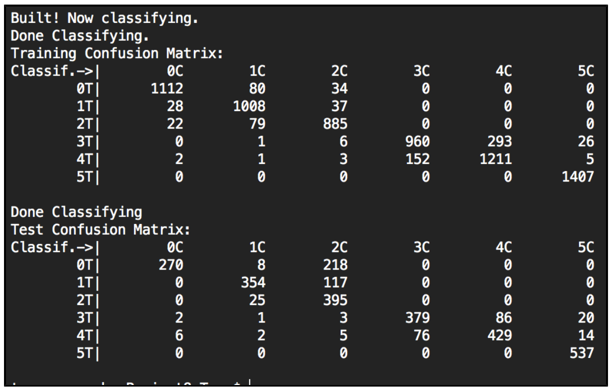


PCA-Transformed Data Set Screenshots:

Naive Bayes Classifier:

First, start by typing this into terminal: *tonys-new-mbp:Project8 Tony$ python2.7 naivebaye_classifier.py UCI-X-train_pca_0_34.csv UCI-X-test_pca_0_34.csv UCI-Y-train.csv UCI-Y-test.csv*

Next, here is the resulting confusion-matrix output in terminal:

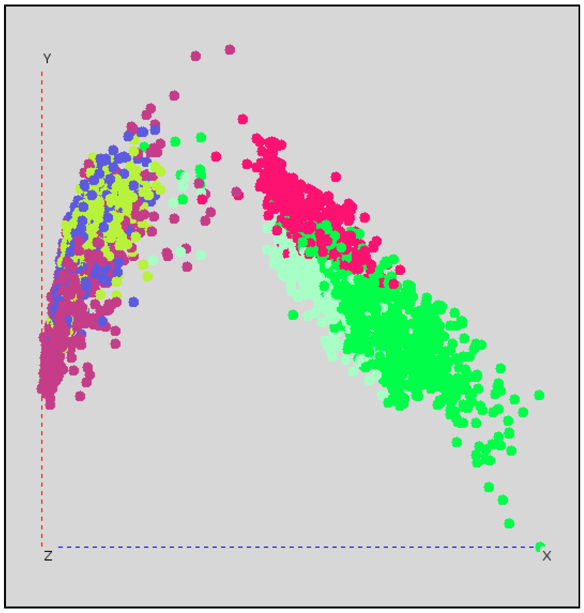


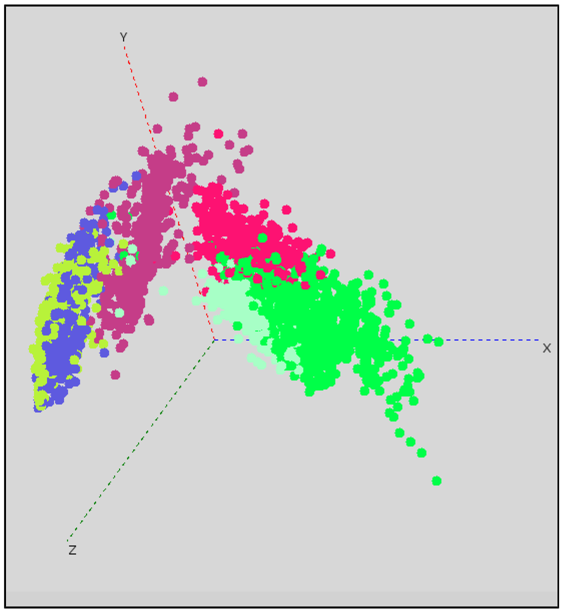
```
Built! Now classifying.
Done Classifying.
Training Confusion Matrix:
Classif.->|       0C        1C        2C        3C        4C        5C
        0T|     1112        80        34         0         0         0
        1T|       28      1008        37         0         0         0
        2T|       22        79       885         0         0         0
        3T|        0         1         6       960       293        26
        4T|        2         1         3       152      1211         5
        5T|        0         0         0         0         0      1407

Done Classifying
Test Confusion Matrix:
Classif.->|       0C        1C        2C        3C        4C        5C
        0T|      270         8       218         0         0         0
        1T|        0       354       117         0         0         0
        2T|        0        25       395         0         0         0
        3T|        2         1         3       379        86        20
        4T|        6         2         5        76       429        14
        5T|        0         0         0         0         0       537
```

Finally, this confusion matrix data is then written into a .csv file titled "nbClass.csv"

Now, I ran my display and opened the "nbClass" csv file and plotted the points on my display. Then, I ran Color Fix method to organize the categories by color. Here are the GUI Display Plotted screenshots for the resulting displays:

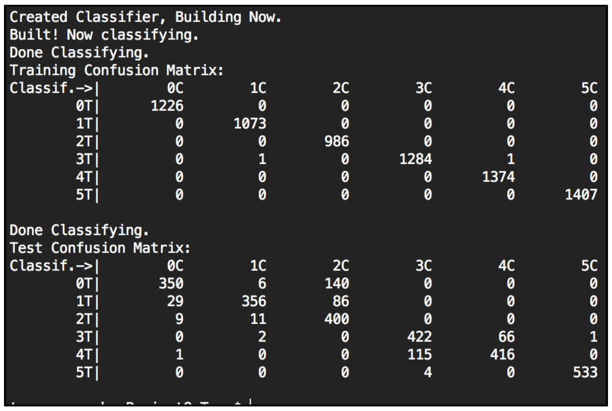




KNN Classifier:

First, start by typing this into terminal: *tonys-new-mbp:Project8 Tony$ python2.7 knn_classifier.py UCI-X-train_pca_0_34.csv UCI-X-test_pca_0_34.csv UCI-Y-train.csv UCI-Y-test.csv*

Next, here is the resulting confusion-matrix output in terminal:

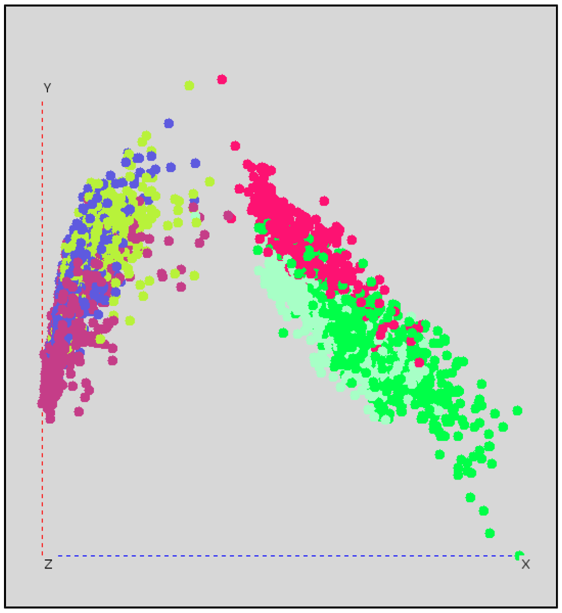


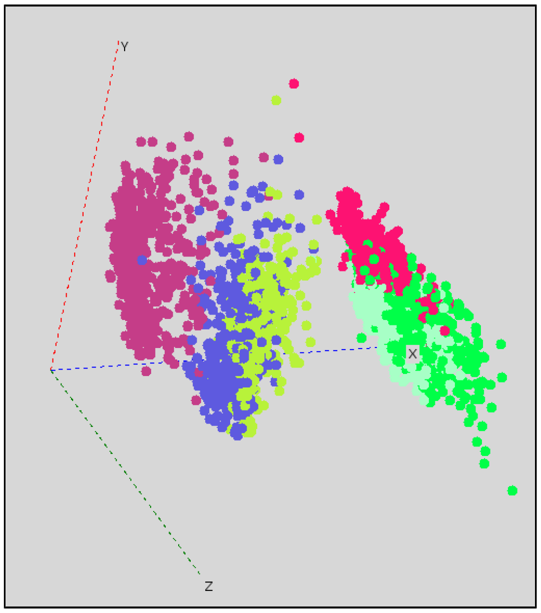
```
Created Classifier, Building Now.
Built! Now classifying.
Done Classifying.
Training Confusion Matrix:
Classif.->|       0C        1C        2C        3C        4C        5C
        0T|     1226         0         0         0         0         0
        1T|        0      1073         0         0         0         0
        2T|        0         0       986         0         0         0
        3T|        0         1         0      1284         1         0
        4T|        0         0         0         0      1374         0
        5T|        0         0         0         0         0      1407

Done Classifying.
Test Confusion Matrix:
Classif.->|       0C        1C        2C        3C        4C        5C
        0T|      350         6       140         0         0         0
        1T|       29       356        86         0         0         0
        2T|        9        11       400         0         0         0
        3T|        0         2         0       422        66         1
        4T|        1         0         0       115       416         0
        5T|        0         0         0         4         0       533
```

Finally, this confusion matrix data is then written into a .csv file titled "knnClass.csv"

Now, I ran my display and opened the "knnClass" csv file and plotted the points on my display. Then, I ran Color Fix method to organize the categories by color. Here are the GUI Display Plotted screenshots for the resulting displays:





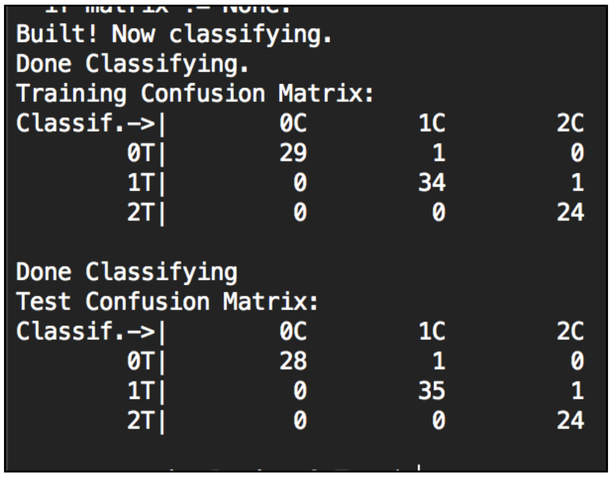
Wine Data Set Screenshots(data set of our choice):

For this part of the project, we were asked to find our own data set of our choosing and make it compatible with our program. So, I went to http://a rchive.ics.uci.edu/ml/datasets.html and found a data set on Wine classification that interested me. The data set described 13 different constituents that make up a wine and the amount found in each of the three types(classes) of wine. Here is the CSV file "wine-all.csv" that contained all the original data: wine-all.csv I then split the classes up by halves and creates wine-train and wine-test CSV files: wine-train.csv and wine-test.csv T hen, I ran the same steps as above on the Wine data:

Naive Bayes Classifier:

First, start by typing this into terminal: *tonys-new-mbp:Project8 Tony$ python2.7 naivebaye_classifier.py wine-train.csv wine-test.csv*

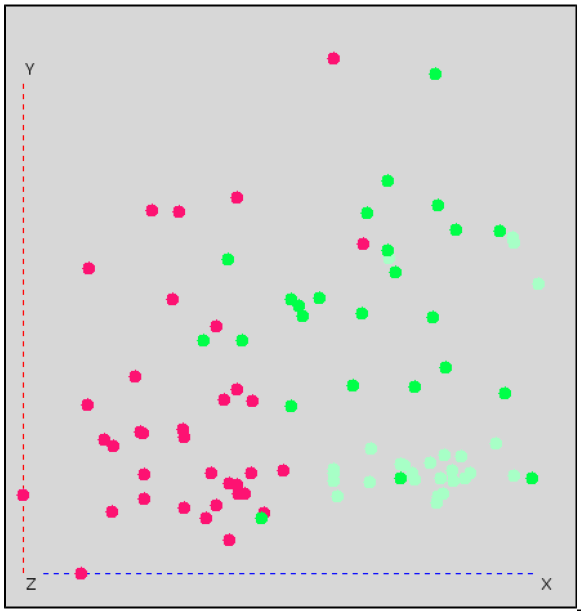
Next, here is the resulting confusion-matrix output in terminal:

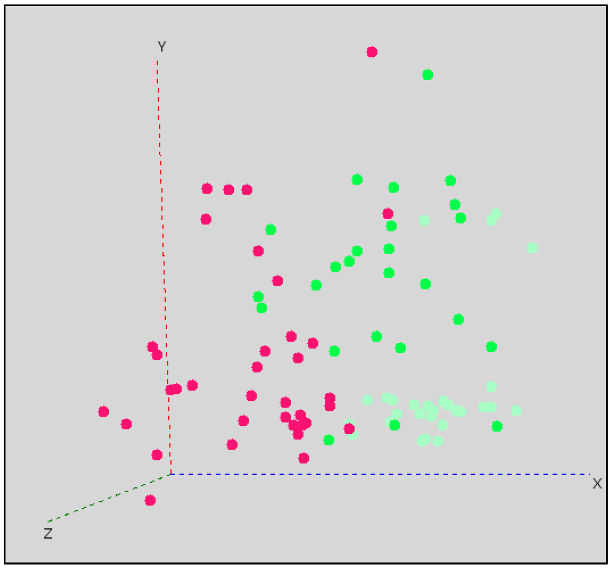


```
Built! Now classifying.
Done Classifying.
Training Confusion Matrix:
Classif.->|        0C         1C         2C
        0T|        29          1          0
        1T|         0         34          1
        2T|         0          0         24

Done Classifying
Test Confusion Matrix:
Classif.->|        0C         1C         2C
        0T|        28          1          0
        1T|         0         35          1
        2T|         0          0         24
```

Graph on Display:(I graphed alcohol, malic acid, and ash)[using nbClass.csv file still...]

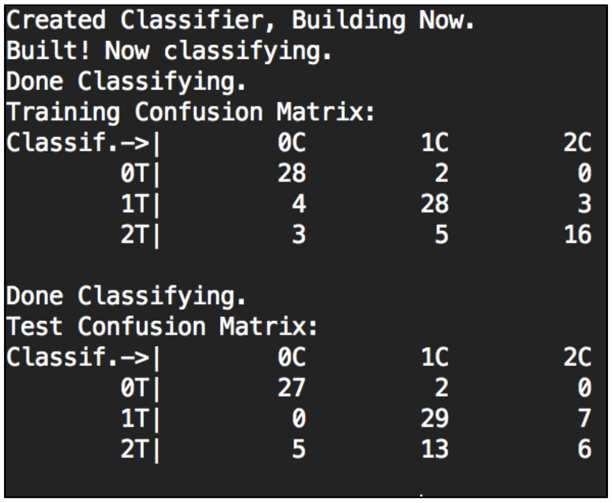




KNN Classifier:

First, start by typing this into terminal: *tonys-new-mbp:Project8 Tony$ python2.7 knn_classifier.py wine-train.csv wine-test.csv*

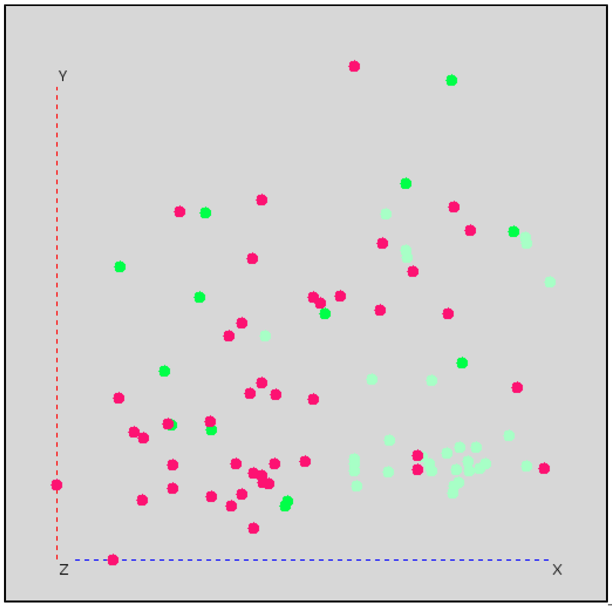
Next, here is the resulting confusion-matrix output in terminal:

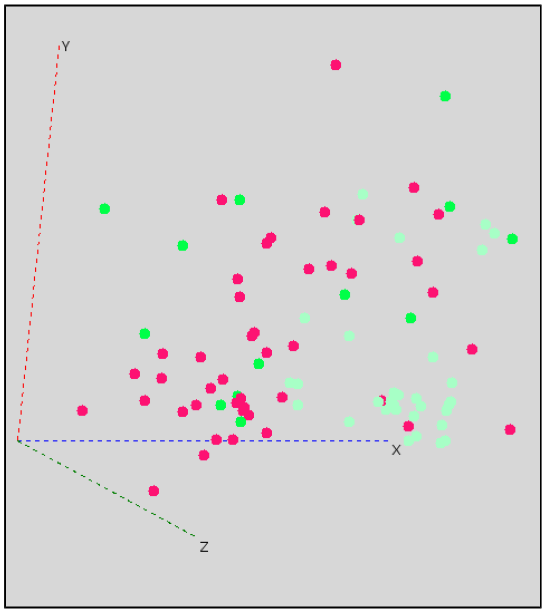


```
Created Classifier, Building Now.
Built! Now classifying.
Done Classifying.
Training Confusion Matrix:
Classif.->|        0C         1C         2C
        0T|        28          2          0
        1T|         4         28          3
        2T|         3          5         16

Done Classifying.
Test Confusion Matrix:
Classif.->|        0C         1C         2C
        0T|        27          2          0
        1T|         0         29          7
        2T|         5         13          6
```

Graph on Display:(I graphed alcohol, malic acid, and ash)[using knnClass.csv file still...]-





Note to Grader: !!!!!!!

So, I hard-coded colorFix in a weird way so that you have to actually input the name of the category you wanted to color by. So, when doing Naive Bayes, it must read "NB Classification" and when doing KNN it must read "KNN Classification" Sorry about this, but this is the best way I could get it to work. So when you're testing you might run into a few problems and have to change the wording in the colorFix method in my display.py file. Also, **important** when running my wine tests, the files still save as nbClass.csv and knnClass.csv, so when running the display, still open those dataSets.

Who helped me:

For this project, I got help from CP Majgaard and met with Bruce and Stephanie a few times. In addition, I worked closely with Steve Parrott and Brendan Doyle.