Zena's CS251 Project 8

For this project, we learned how to make two classifiers, which are forms of machine learning. We had to make both a Naive Bayes Classifier and a KNN (K Nearest Neighbors) classifier. We trained them using data with given classes. We created a file to test both types; I allowed the user to specify which kind to do at command line. With my implementation, KNN takes a lot longer than naive bayes, but KNN also is a lot more accurate. I also tested it on testing and training datasets I created.

Task 1 was to make a method that would create a confusion matrix method. For this, I read in both the correct categories and the ones the classifier got, and I have a for loop that loops through the amount of points and compares the classes at the given index to each other and adds 1 to the area in the matrix corresponding to the spaces those classes have in the confusion matrix. For example, if a point is in the first class, and its corresponding classifier point is in the third class, it will go in confusionMatrix[0,2].

Task 2 was to create the file that tests the classifiers, which I called classifyTest.py. I take input for which training and test classes to use, what to call the output file (which is the test file plus the column for classifier result classes), which classifier to use, and an optional pair of separate header files.

Task 3 was to run the code on the activity set. The following pictures are my results:

Using Naive Bayes on the activity set:

```
TRAINING CONFUSION MATRIX:
    897.0    201.0    128.0    0.0    0.0    0.0
    23.0    971.0    79.0    0.0    0.0    0.0
    56.0    170.0    760.0    0.0    0.0    0.0
    16.0    7.0    0.0    1231.0    23.0    9.0
    11.0    18.0    0.0    1247.0    97.0    1.0
    34.0    0.0    0.0    1212.0    0.0    161.0


TESTING CONFUSION MATRIX:
    416.0    38.0    42.0    0.0    0.0    0.0
    9.0    451.0    11.0    0.0    0.0    0.0
    81.0    83.0    256.0    0.0    0.0    0.0
    6.0    5.0    0.0    469.0    9.0    2.0
    11.0    10.0    0.0    450.0    57.0    4.0
    8.0    0.0    0.0    473.0    0.0    56.0

Zenas-MacBook-Pro:Project8 School$
```

Using KNN on the activity dataset:

```
TRAINING CONFUSION MATRIX:
    1226.0    0.0    0.0    0.0    0.0    0.0
    0.0    1073.0    0.0    0.0    0.0    0.0
    0.0    0.0    986.0    0.0    0.0    0.0
    0.0    0.0    0.0    1286.0    0.0    0.0
    0.0    0.0    0.0    0.0    1374.0    0.0
    0.0    0.0    0.0    0.0    0.0    1407.0


TESTING CONFUSION MATRIX:
    480.0    5.0    11.0    0.0    0.0    0.0
    34.0    426.0    11.0    0.0    0.0    0.0
    45.0    40.0    335.0    0.0    0.0    0.0
    0.0    3.0    0.0    395.0    93.0    0.0
    0.0    0.0    0.0    55.0    477.0    0.0
    0.0    0.0    0.0    1.0    1.0    535.0
```

Using Naive Bayes on the activity set with PCA:

```
TRAINING CONFUSION MATRIX:
    1112.0    80.0    34.0    0.0    0.0    0.0
    28.0    1008.0    37.0    0.0    0.0    0.0
    22.0    79.0    885.0    0.0    0.0    0.0
    0.0    1.0    6.0    960.0    293.0    26.0
    2.0    1.0    3.0    152.0    1211.0    5.0
    0.0    0.0    0.0    0.0    0.0    1407.0


TESTING CONFUSION MATRIX:
    270.0    8.0    218.0    0.0    0.0    0.0
    0.0    354.0    117.0    0.0    0.0    0.0
    0.0    25.0    395.0    0.0    0.0    0.0
    2.0    1.0    3.0    379.0    86.0    20.0
    6.0    2.0    5.0    76.0    429.0    14.0
    0.0    0.0    0.0    0.0    0.0    537.0
```

Using KNN on the activity set with PCA:

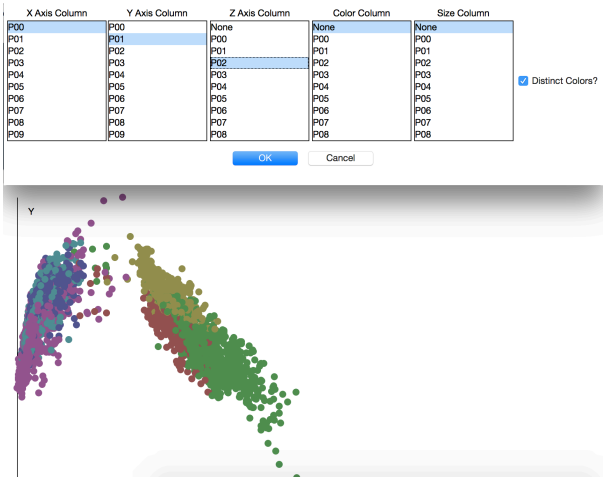
```
TRAINING CONFUSION MATRIX:
    1226.0    0.0    0.0    0.0    0.0    0.0
    0.0    1073.0    0.0    0.0    0.0    0.0
    0.0    0.0    986.0    0.0    0.0    0.0
    0.0    0.0    0.0    1286.0    0.0    0.0
    0.0    0.0    0.0    0.0    1374.0    0.0
    0.0    0.0    0.0    0.0    0.0    1407.0


TESTING CONFUSION MATRIX:
    350.0    6.0    140.0    0.0    0.0    0.0
    29.0    356.0    86.0    0.0    0.0    0.0
    9.0    11.0    400.0    0.0    0.0    0.0
    0.0    2.0    0.0    423.0    65.0    1.0
    1.0    0.0    0.0    115.0    416.0    0.0
    0.0    0.0    0.0    4.0    0.0    533.0
```

Using the PCA data made the naive bayes classification more accurate while it made the KNN slightly worse. KNN seems to be more accurate for the given data while Naive Bayes is better for generalizing patterns, since it uses a general idea of a distribution while KNN may overfit while calculating the best class each iteration. It is also a lot more expensive because of its looping of every point and every class mean. KNN is more accurate at the expense of speed, while NB is better for making estimates that hold for different size datasets.

For task 4, we had to plot the clusters a classifier received as the colors of a plot of the first three PCA headers

PCA plot picture (using the output of doing KNN on the activity set with PCA data):



Task 5 was to complete analyses for a dataset using both analyses again, on the normal and the pca versions.

I used my usual dataset of wage data.

This is the result for using naive bayes on the set:

```
TRAINING CONFUSION MATRIX:
    4.0    0.0    0.0    0.0
    1.0    9.0    0.0    0.0
    0.0    0.0    9.0    1.0
    0.0    0.0    0.0    6.0


TESTING CONFUSION MATRIX:
    3.0    0.0    0.0    0.0
    1.0    7.0    0.0    0.0
    0.0    0.0    8.0    1.0
    0.0    0.0    0.0    5.0
```

This is the result for using KNN on it:

```
TRAINING CONFUSION MATRIX:
    3.0    1.0    0.0    0.0
    1.0    8.0    1.0    0.0
    0.0    2.0    7.0    1.0
    0.0    0.0    1.0    5.0


TESTING CONFUSION MATRIX:
    2.0    1.0    0.0    0.0
    1.0    6.0    1.0    0.0
    0.0    2.0    6.0    1.0
    0.0    0.0    1.0    4.0
```

This is the result for using Naive Bayes on the first three PCA columns of the PCA version of the set made from all of the headers:

```
TRAINING CONFUSION MATRIX:
    4.0    0.0    0.0    0.0
    1.0    8.0    1.0    0.0
    0.0    0.0    8.0    2.0
    0.0    0.0    0.0    6.0


TESTING CONFUSION MATRIX:
    4.0    1.0    0.0    0.0
    0.0    6.0    1.0    0.0
    0.0    0.0    6.0    0.0
    0.0    0.0    1.0    6.0
```

This is the result for using KNN on the first three PCA columns of the PCA version of the set made from all of the headers:

```
TRAINING CONFUSION MATRIX:
    4.0    0.0    0.0    0.0
    0.0    10.0    0.0    0.0
    0.0    0.0    10.0    0.0
    0.0    0.0    0.0    6.0


TESTING CONFUSION MATRIX:
    4.0    1.0    0.0    0.0
    0.0    7.0    0.0    0.0
    0.0    0.0    6.0    0.0
    0.0    0.0    2.0    5.0
```

To make my training set, I selected some of the original ones and assigned them classes based on what decade they were in (1940s = 0, 1950s = 1, etc) then did the same for the test set's classes, which are used in the confusion matrix to show effectiveness. I did about the same thing for the PCA training and test data but I made them through the UI's kmeans file-writing method, then adjusted them to not have the kmeans information. I've included the four files with what are hopefully helpful names.

In this project, I learned about how the Naive Bayes and KNN algorithms work. There were a lot of algorithms to fill in although they were outlined, and it took a lot of testing and debugging to get them to work. As usual, dealing with errors from indexing into matrices was the biggest challenge. I also took the chance to fix my get_numeric_headers method in this project, because the test files assume that the headers are in the order they were in the file. Up until now, I had used a dictionary to retrieve them, but I realized I could loop through the raw headers list and return only those with numeric data. This also helped to fix the problem I was having with doing kmeans on PCA data.

Special thanks: Melody, Stephanie