ACKNOWLEDGMENT

The project report is based a data mining task where we perform classification on two different datasets. We thank the professor **Dr. Chris Ding** as well as the teaching assistant **Qicheng Wang** without who's help preparing this project might have been a hideous task and also their consistent effort to help the students providing us with all the details necessary to prepare the a project made it rather easy and accomplishable.

Sumedh Walujkar (MavID:1001560040)

Venkat Akhil Gangineni (MavID:1001560854)

TABLE OF CONTENTS

PROJECT DESCRIPTION

The project is about using different classification techniques in a diverse ways on two different datasets namely the ATNTFaceImages400 and HandWrittenLetters. Upon these two datasets we perform classification using three different classifiers which are the K-Nearest Neighborhood, Nearest Centroid and Support Vector Machines and in turn implement certain operations like predicting the accuracy for each classifier, separating the training and test instances of the data based on a particular criteria etc.

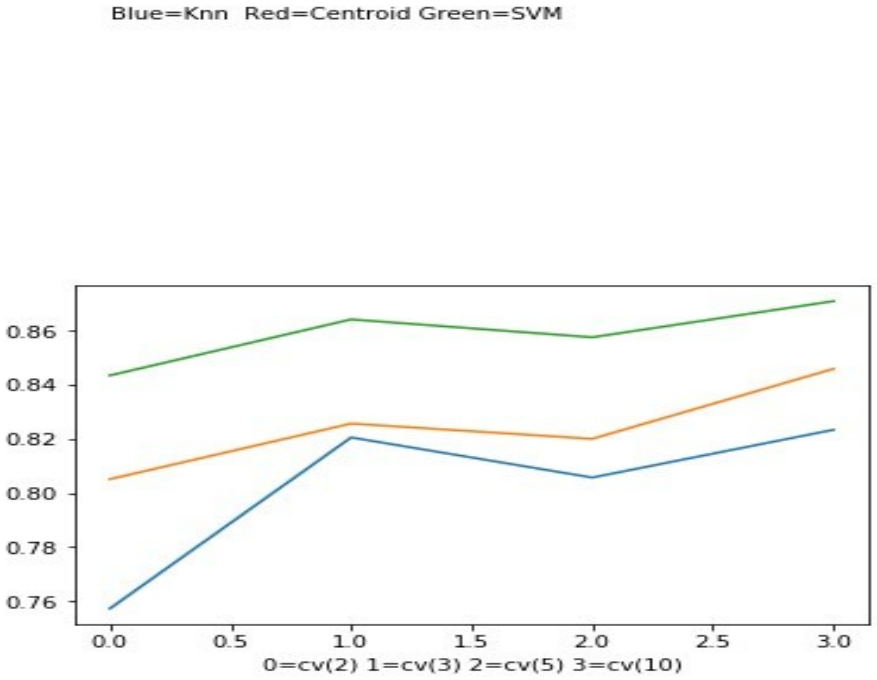
The world these days runs on automation so we decided to go with the world and provided our project with this additional feature. So, the project now along with the ability to perform all the above mentioned operations is also given the ability of automation which makes it very easy for the user, all he needs to do is specify the filename, training and test instances he want to consider, the letters for the subroutines build from where the data is to be picked etc. All of these additional functionalities that have been added to the project makes it easier for a user to implement and understand.

PROJECT SEGMENTS

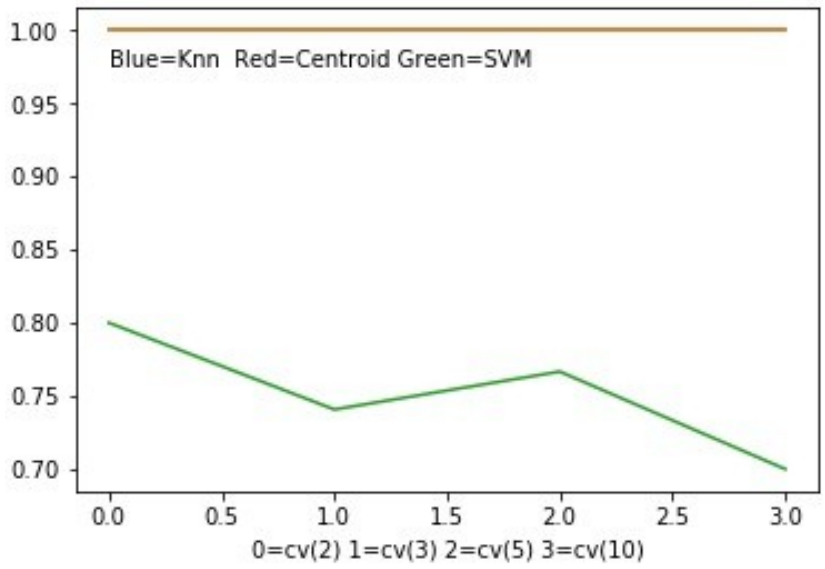
Segment 1A &1B

We were asked to perform run 5-fold, 2-fold, 3-fold and 10-fold CV cross-validation for ATNTFaceImages400 and HandWrittenLetters using each of the three classifiers: KNN, centroid, and SVM and then to report the classification accuracy on each classifier by taking the average of the 5 accuracy numbers and finally plot the average accuracy numbers on a figure.
To perform this segment we used the scikit learn algorithms for the classification part and to automate this we extended the pickData subroutine of the segment 1C and gave it an additional parameter which is choice through which the user can enter his choice to generate cross-validation or prediction results and finally generate graphs if necessary.

Results:



Graph for HandWrittenLetters



Graph for ATNTFaceImages400

from the above graphs we observe a trend that in either of them as the number of folds for cross-validation increases, the resulting accuracy have also increased, but the only exception is for SVM classifier of ATNTFaceImages400 where the reverse happens.

Segment 1C

In the segment 1C we were asked to write three subroutines. Subroutine1 that can generate a training data and test data from the ATNT or hand-written-letter data. The subroutine contains a pickData function with filename, class_numbers, training_instances, test_instances and choice as its parameters.

filename: char_string specifying the data file to read. For example, 'ATNT_face_image.txt'

class_numbers: an array that contains the classes to be pick. For example: (3, 5, 8, 9)

training_instances: the first number of instances in each class to be used as training data.

test_instances: the remaining instances in each class to be used as test data.

choice: to select between cross validation and prediction.
subroutine1: pickData('Handwrittenletters.txt', (3, 15, 26), 30, 9).
Use handwrittenletters.txt data file. Pick classes: 3, 15, 26.
First 30 images for training, last 9 images for testing. This subroutine will output four data matrix/arays: (trainX,trainY,testX,testY).
The data is to be easily feed into a classifier. We write another subroutine2 that will store (trainX,trainY) into a training data file, and store (testX,testY) into a test data file. The format of these files is determined by user's choice: matlab file, a text file, or a file convenient for Python. These files allow the data to be easily read and feed into a classifier. We also need to write subroutine3: "letter_2_digit_convert" that converts a character string to an integer array. For example,letter_2_digit_convert('ACFG') returns array (1, 3, 6, 7).

In the pickData subroutine firstly, the file is read by entering the filename and then the choice where if selected 0 the data is split into TrainX, TestX, TrainY and Test Y. Now when we are done with the splitting part we pass the output of this subroutine in a list which we called final and the first value of this final will be 0 if we are performing a prediction and 1 if we are performing a cross validation. Likewise all the subroutines with the above mentioned functionalities are constructed under the pickData function and linked to the other segments of the project.

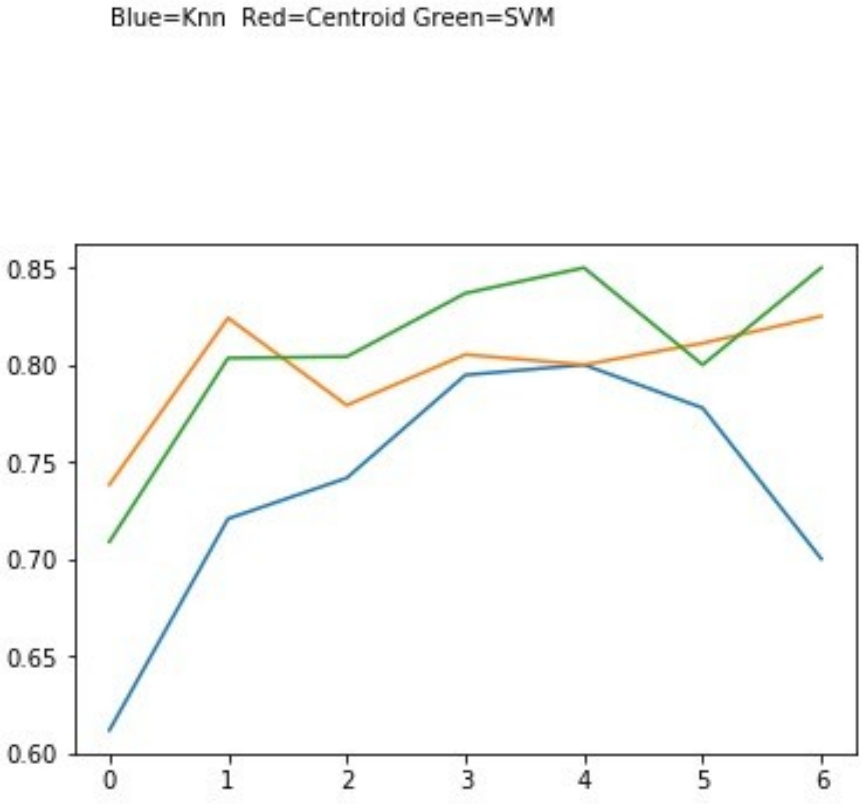
Results here depend upon your splitting criterion so the output changes with the type of split or the training and testing instances you specified.

Segment 1D&1E

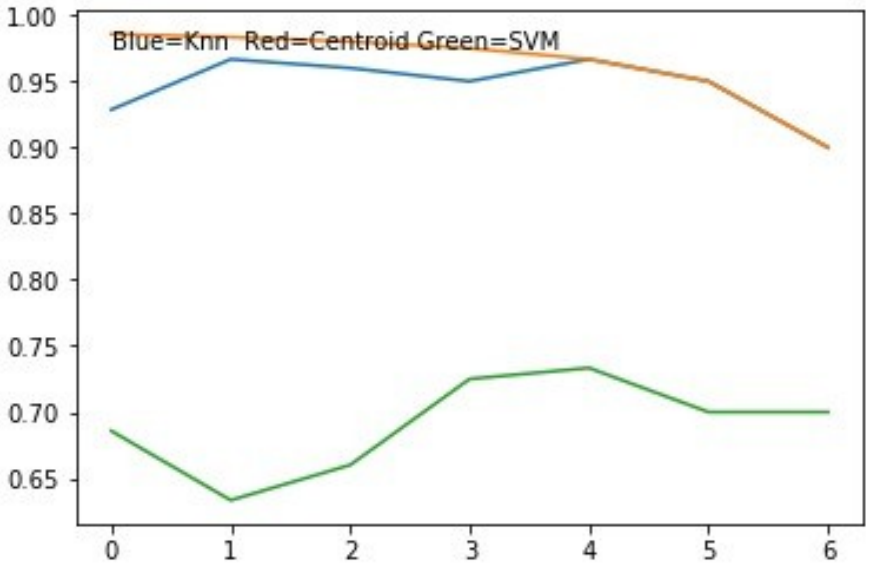
For these two segment. We use the picData routine to generate training and test data files. This is done for seven different splits: (train=5 test=34), (train=10 test=29), (train=15,test=24), (train=20 test=19), (train=25 test=24), (train=30 test=9) ,(train=35 test=4). On these seven different data-split cases, run the centroid classifier to compute average test image classification accuracy. In the 1E segment the similar is performed for seven other splits. Finally, we plot these 7 average accuracy on one curve in a figure. Along with this feature the segments of the project also has the capability to take

any valid number of training instances and automatically take the remaining as test instances and therefore generate the accuracy and plot the graphs.

Results:



Graph for Segment 1D



Graph for segment 1E