CSE 572: Data Mining
Spring 2017
Assignment 3 / Mini Project 1

Team members:-

1. Saurabh Jagdhane (1209572595)
2. Himanshu Tyagi (1209283279)
3. Rachita Gupta (1209247724)

Problem 1

a) To solve this problem, we are using *fitcknn,* an in built MATLAB function. In the file 'KNN_1a.m' the first three lines, we load the relevant dataset files into matlab matrices. Next we use the matlab's 'fitcknn' function to create the model for K-nearest neighbor classification with k=5 and distance measure as 'euclidean' on the given dataset. Once the model is created we predict the classes and store them in the matrix 'label'. The accuracy of the model is predicted by using the matlab's 'classperf' method with the predicted and true labels as function parameters.

Accuracy= 90.0238%

b) The program first loads the dataset in the matlab matrices 'x_train', 'y_train' and 'z_train'. The number of unique classes in the dataset are '6' (found by matlab function 'unique'). Since we have to perform one-against-the other classification, we have to create a n*6 matrix, which will be storing the intermittent results.Next we perform preprocessing on dataset, by creating to temporary matrix 'temp' which stores the training class labels and is input matlab's 'fitcsvm' function to which is used to train our model. Once the model is trained, we use matlab's 'predict' method to predict the class labels for the dataset 'X_test' and store the results in matrix 'res'. Now, we again convert the resultant dataset into the column vector 'label', this vector along with the true labels stored in matrix 'y_test' are given as input to the method 'classperf' for calculating the accuracy of the model.

Accuracy=97.8201%

Problem 2

a) The first three lines, we load the relevant dataset files into matlab matrices. Next we use the matlab's 'fitcknn' function to create the model for K-nearest neighbor classification with k=5 and distance measure as 'euclidean' on the given dataset. Once the model is created we predict the classes and store them in the matrix 'label'. The accuracy of the model is predicted by using the matlab's 'classperf' method with the

predicted and true labels as function parameters.
Accuracy : 98.60%

b) The training set X_train, y_train is used to train a feedforward neural network using patternnet function having 1 hidden layer with 25 neurons and train function one time. The y_train is converted to a target matrix using ind2vec() before utilizing it in the train() function. The class labels for the test set is fetched from the generated neural network. The result is compared with the actual y_test to compute the accuracy.
Accuracy : 97.80% (Accuracy will change each time because of initial random weights)

c) The program first loads the dataset in the matlab matrices 'x_train', 'y_train' and 'z_train'. The number of unique classes in the dataset are '6' (found by matlab function 'unique'). Since we have to perform one-against-the other classification, we have to create a n*6 matrix, which will be storing the intermittent results.Next we perform preprocessing on dataset, by creating to temporary matrix 'temp' which stores the training class labels and is input matlab's 'fitcsvm' function to which is used to train our model. Once the model is trained, we use matlab's 'predict' method to predict the class labels for the dataset 'X_test' and store the results in matrix 'res'. Now, we again convert the resultant dataset into the column vector 'label', this vector along with the true labels stored in matrix 'y_test' are given as input to the method 'classperf' for calculating the accuracy of the model.
Accuracy : 99.7980%

Functions used:

1. fitcknn(): Fit k-nearest neighbor classifier. Using this function we specify value of K and run it on our training data

   ```
   fitcknn(X_train,Y_train,'NumNeighbors',5,Distance,'Euclidean')
   ```

   Using this function we specify that value of K is 5 and we use euclidean distance to find the distance between pair of samples.

2. fitcsvm(): This function trains binary support vector machine classifier for two-class binary classification for a moderate-dimensional predictor data set

   ```
   fitcsvm(X,Y,'kernelfunction','polynomial','polynomialOrder',2)
   ```

   In this situation we have 6 classes so to use svm we use technique 1-r and train 1 against the rest using kernel function.

3. patternnet (Pattern recognition networks): These are feedforward networks that can be trained to classify inputs according to target classes. It is used when there is a hidden layer in neural network

```
net = patternnet(25);
```

4. predict(trained_model, test_data): It returns predicted class labels for the predictor data in test_data, based on trained discriminant analysis.

5. classperf(): It is used to evaluate the performance of classifier.

```
classperf(truelabels, classout)
```

   In above case, truelables are true class labels for each observation and classout contains classifier output labels.

6. ind2vec(): Convert indices to vectors.

7. vec2ind(): Convert vectors to indices.

8. train(net,transpose(X_train),target): Trains neural network as per topology specified. net - Network, transpose(X_train) - network inputs, target - network targets.

9. importdata(filename): Imports data from a file.