Problem 1.

Train with 1,000 samples with **mnist**

```
traindataset = datastore('mnist_train.csv')
trainfile =readall(traindataset)
testdata = datastore('mnist_test.csv')
testfile = readall(testdata)
dataset = [trainfile; testfile]

knnmodel = fitcknn(dataset(1:1000, :), dataset.label(1:1000,
    :), NumNeighbors=10)
predictions = predict(knnmodel, dataset(1001:end, :));
accuracy = sum(predictions == dataset.label(1001:end, :))/
   numel(predictions)

neighborMatrix = zeros(20,2)
for i =1:20
    knnmodel = fitcknn(dataset(1:1000, :), dataset.label
       (1:1000, :), NumNeighbors=i);
    predictions = predict(knnmodel, dataset(1001:end, :));
    accuracy = sum(predictions == dataset.label(1001:end, :)
       )/numel(predictions);
    temp = [i accuracy]
    neighborMatrix(i,1) = i;
    neighborMatrix(i,2) = accuracy;
end
```

Problem 2.

Evaluate with test samplesof original datasets

$$(1, 0.8692), (3, 0.8603)$$

- If i = 1, accuracy = 0.8692.
- IF i = 3, accuracy = 0.8603.

Problem 3.

Tune up the parameter and choose your 'best' model based on their performance.

```
%% neural network
nnmodel = fitcnet(dataset(1:1000, :), dataset.label(1:1000,
    :))
predictions = predict(nnmodel, dataset(1001:end, :));
accuracy = sum(predictions == dataset.label(1001:end, :))/
    numel(predictions) %0.2071

%% decision tree
treemodel = fitctree(dataset(1:1000, :), dataset.label
    (1:1000, :))
predictions = predict(treemodel, dataset(1001:end, :));
accuracy = sum(predictions == dataset.label(1001:end, :))/
    numel(predictions) %1
```

- Dicision Tree Model is the best training model for this dataset.