Homework 4
Applied Machine Learning
Fall 2017
CSCI-P 556/INFO-I 526

Manoj Joshi
manjoshi@iu.edu

November 27, 2017

All the work done is solely mine. - Manoj Joshi

Directions

Please follow the syllabus guidelines in turning in your homework. I am providing the LaTeX of this document too. This homework is due Monday November 20, 2017 11:59p.m. **OBSERVE THE TIME**. Absolutely no homework will be accepted after that time. Bring a hardcopy to Tuesdays class on the 21th. If you do not bring a hardcopy with the statement of your own work, the homework will not be accepted. All the work should be your own. Within a week, AIs can contact students to examine code; students must meet within three days. The session will last no longer than 5 minutes. If the code does not work, the grade for the program may be reduced. Lastly, source code cannot be modified post due date.

K-Fold Cross Validation for Model Selection

```
ALGORITHM k-fold cross validatiaon
INPUT
    • training data $\Delta = (\mathbf{x_1}, y_1), \ldots, (\mathbf{x_m}, y_m)$
    • set of parameter values $\Theta$
    • learning algorithm $\mathcal{H}$
    • integer $k$
OUTPUT
    • $\theta^* = \text{argmin}_\theta[error(\theta)]$
    • $h_{\theta^*} = \mathcal{H}(\Delta; \theta^*)$
Randomly partition $\Delta$ into $\Delta_1, \ldots, \Delta_k$
*** $\Delta_1 \cup \Delta_2 \ldots \cup \Delta_k = \Delta$ and $\Delta_i \cap \Delta_j = \varnothing$ for $i \neq j \in [1, 2, \ldots, k]$
for $\theta \in \Theta$ do
    for $i = 1 \ldots k$ do
        *** Train a model for each training set
        $h_{i,\theta} = \mathcal{H}(\Delta \setminus \Delta_i; \theta)$
    end for
    *** Use the trained models over $\Delta_i$ (test data sets) to evaluate the models for each parameter
    $\text{error}(\theta) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}_{\Delta_i}(h_{i,\theta})$
end for
```

K-Nearest Neighbors (KNN) Algorithm in Theory

```
ALGORITHM K-nearest neighbors
INPUT
    • training data $\Delta$
    • test data $\Delta'$
    • distance metric $d$, i.e., $d : \Delta^2 \to \mathbb{R}_{\geq 0}$
    • integer $k$: nearest neighbors number
OUTPUT
    • class label of each $z \in \Delta'$
for $z = (\mathbf{x}', y') \in \Delta'$ do
    Compute $d(\mathbf{x}, \mathbf{x}')$, the distance between z and every example $(\mathbf{x}, y) \in \Delta$
    Select $\Delta_z \subseteq \Delta$, the set of closest $k$ training examples to z
    Voting:
    • majority voting: $y' = \text{argmax}_v \sum_{(\mathbf{x_i}, y_i) \in \Delta_z} I(v = y_i)$
    • distance-weighted voting: $y' = \text{argmax}_v \sum_{(\mathbf{x_i}, y_i) \in \Delta_z} w_i \times I(v = y_i)$ where $w_i = \frac{1}{d(\mathbf{x}', \mathbf{x_i})^2}$
end for
```

Naive Bayes Classifier

```
ALGORITHM  Training of naive bayes classifier (continuous attributes)
*** training set: $\Delta = \{(\mathbf{x_j}, y_j)\}_{j=1}^m$
for $i = 1, \ldots, k$ do
    *** class-specific subsets
    $\Delta_i \leftarrow \{\mathbf{x_j} | y_j = c_i, j = 1, \ldots, m\}$
    *** size of the subsets
    $m_i \leftarrow |\Delta_i|$
    *** prior probability
    $\hat{P}(c_i) \leftarrow m_i / m$
    *** mean
    $\hat{\mu}_i \leftarrow \frac{1}{m_i} \sum_{\mathbf{x_j} \in \Delta_i} \mathbf{x_j}$
    *** centered data
    $\mathcal{Z}_i \leftarrow \Delta_i - \mathbb{I}_{m_i} \hat{\mu}_i^T$
    *** variance
    $\hat{\sigma}_i \leftarrow \frac{1}{m_i} \mathcal{Z}_i^T \mathcal{Z}_i$
end for
return $\hat{P}(c_i), \hat{\mu}_i, \hat{\sigma}_i$ for all $i = 1, \ldots k$

TESTING( $\mathbf{x}$ and $\hat{P}(c_i), \hat{\mu}_i, \hat{\sigma}_i$, for all $i \in [1, k]$):
$\hat{y} \leftarrow \text{argmax}_{c_i} \{f(\mathbf{x} | \hat{\mu}_i, \hat{\sigma}_i) \ \hat{P}(c_i)\}$
return $\hat{y}$
```

M-estimate of Conditional Probability

If the class-conditional probability for one of the attributes is zero, then overall posterior probability for the class vanishes. This problem can be addressed by using the m-estimate approach for estimating the conditional probability:

$$P(x_i|y_j) = \frac{n_c + m \times p}{n + m}$$

- x_i: training example x_i, y_j**:** class y_j
- n_c : number of training examples from class y_j that take on the value x_i
- n : total number of instances from class y_j
- m : equivalent sample size. m determines the trade-off between the prior probability p and the observed probability n_c/n
- p : user-specified parameter. p can be regarded as the prior probability of observing the attribute value x_i among records with class y_j

In this homework, you are asked to implement k-nearest neighbors (KNN), naive bayes classifier and k-fold cross validation for model selection. You will test/compare them over Ionosphere, car evaluation and credit approval data sets. Click on the links below to obtain the data sets.

- Ionosphere Data Set
- Car Evaluation Data Set
- Credit Approval Data Set

Problem 1: K-Fold Cross Validation [25 points]

Implement k- fold cross validation and select $k = 5$ to create 5 training and 5 test data sets from each data set and save these 30 files. You will use these data sets for model comparison and parameter selection.

Each dataset was divided into 5 block. If the size of the data was not divisible by 5 then the last block would get the extra part.

Each data block becomes a test set set exactly once. So we have 5 train and test sets for each data set. For three different datasets we would get 30 files.

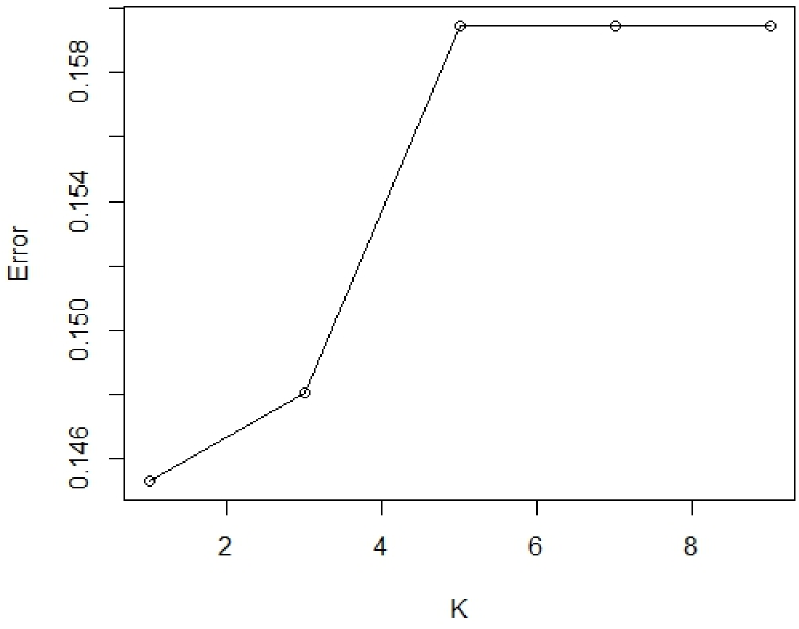
Ionosphere data sets begin with the prefix "ionosphere.data.txt".
Car Evaluation data sets begin with the prefix "car.data.txt".
Credit Approval data sets begin with the prefix "crx.data.txt".
All the file names have the term "train" or "test" at the end to indicate the same.

Problem 2: K-Nearest Neighbors (KNN)[55 points]

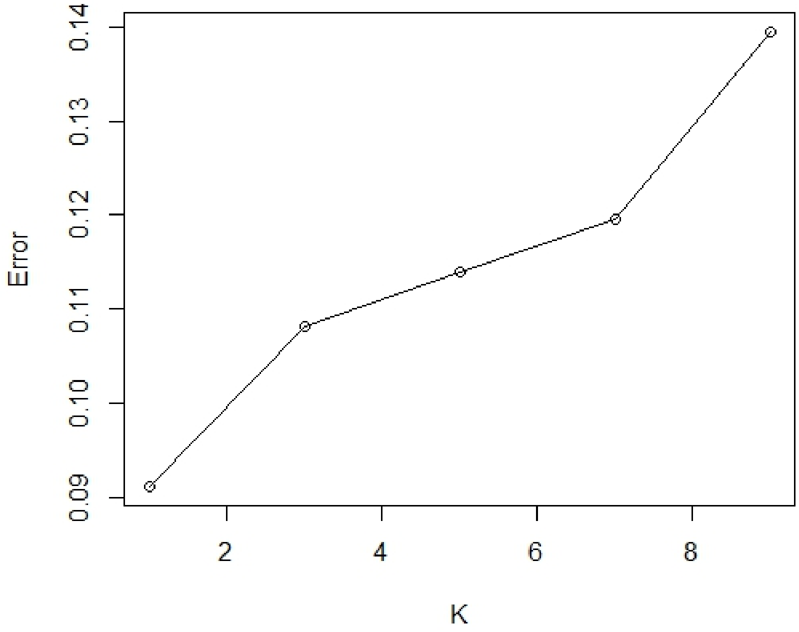
2.1 Implement KNN algorithm with two different distance functions. You can either use existing distance functions, i.e., Euclidean or design your own.
The distance functions used are Euclidean and Manhattan.
2.2 Use the data sets obtained in problem 1 to determine the optimal k over each data set for KNN algorithm. For 5 different k values, plot the test error for each data set. Total number of figures = 3 (data set number) $\times$ 2 (distance function number) = 6. Report the best k and distance function for each data set.

The two graphs for Ionosphere dataset are

For Euclidean distance, find the graph below:



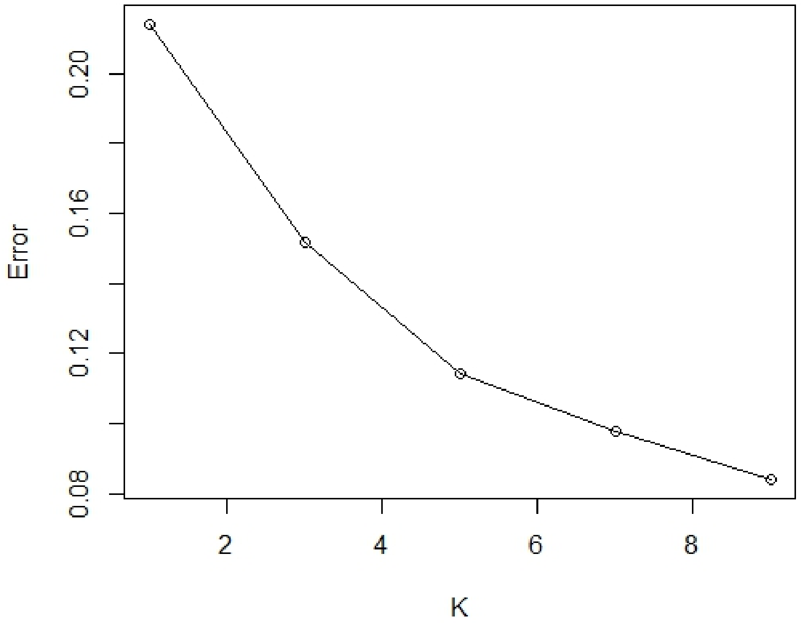
For Manhattan distance, find the graph below:



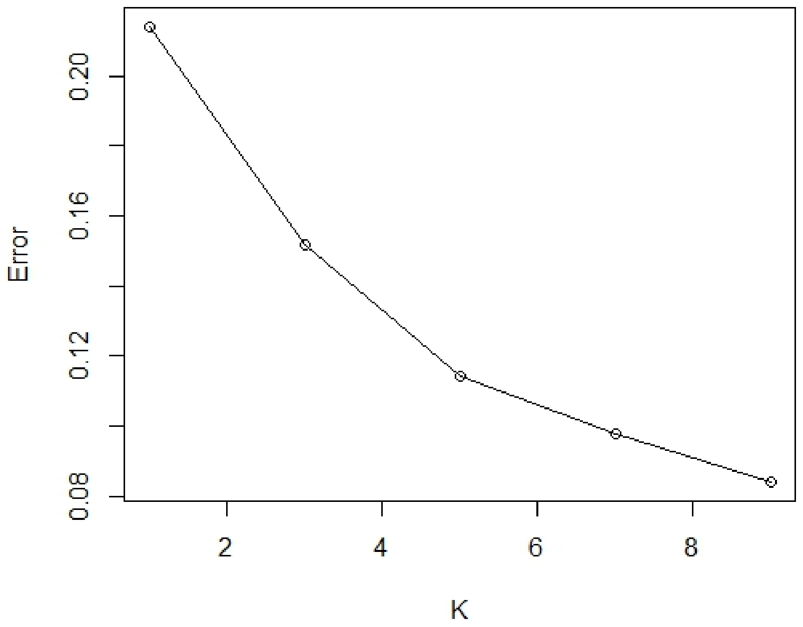
From the graphs above we see that k=1 seems to be a good choice with Manhattan distance.

The two graphs for Car Evaluation dataset are

For Euclidean distance, find the graph below:

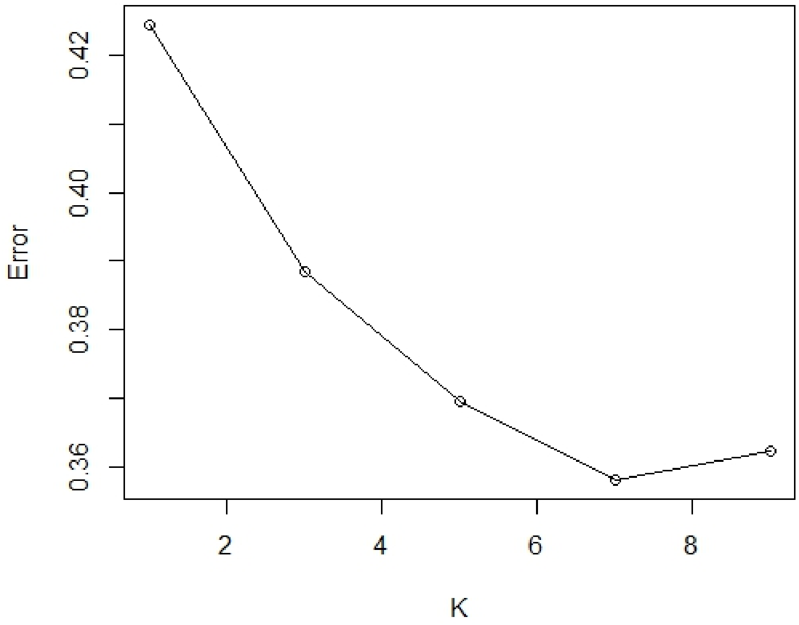


For Manhattan distance, find the graph below:

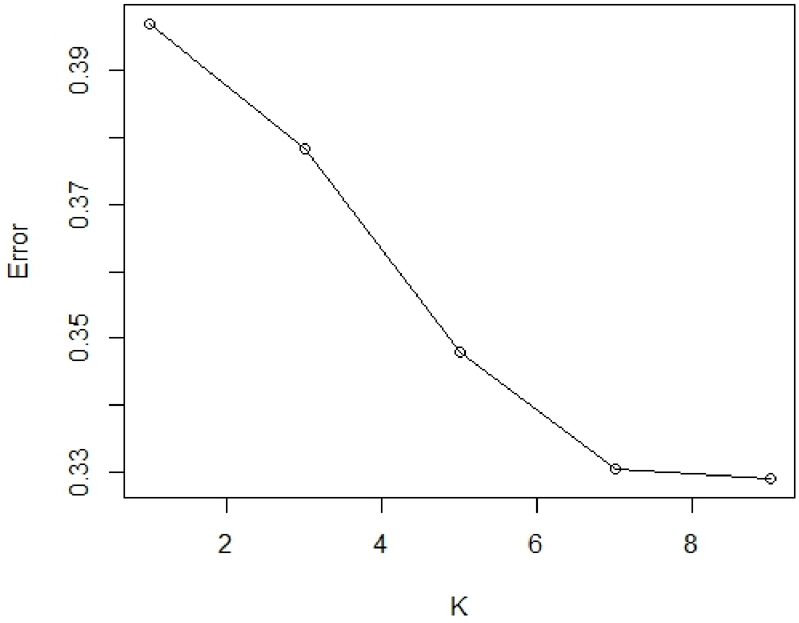


From the graphs above we see that k=9 seems to be a good choice with either the Euclidean or the Manhattan distance.

The two graphs for Credit Approval dataset are
For Euclidean distance, find the graph below:



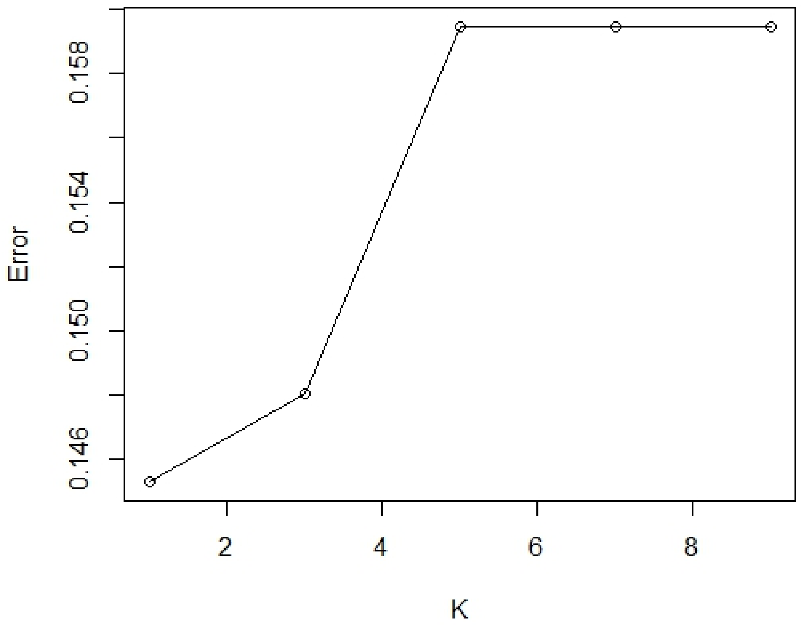
For Manhattan distance, find the graph below:



From the graphs above we see that k=7 seems to be a good choice with Manhattan distance performing a little better.

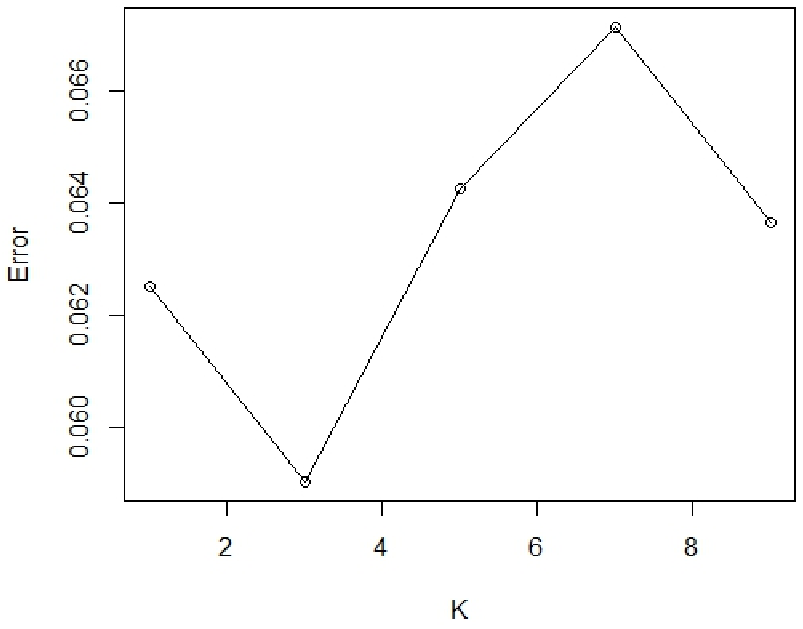
2.3 Use the KNN package in R for validation.

The graph for Ionosphere dataset using the knn package is shown below



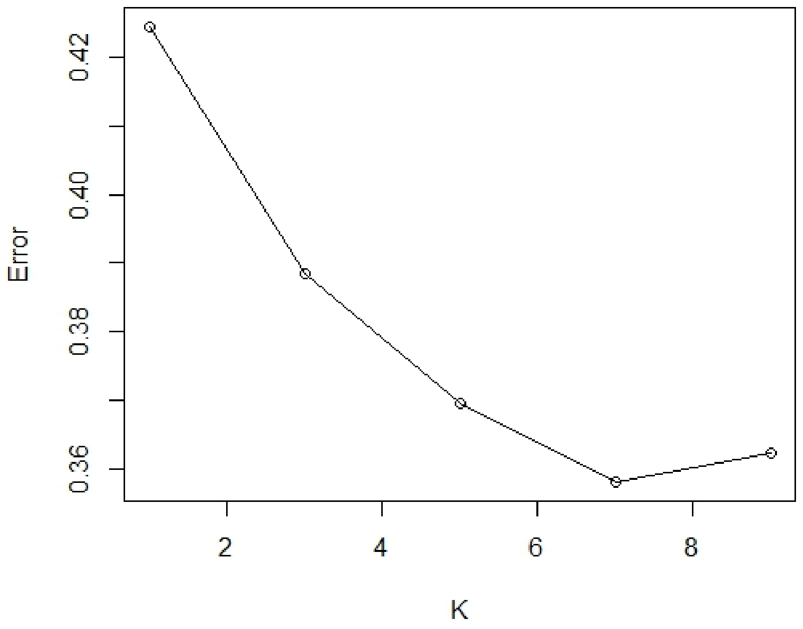
From the graph above we see that k=1 seems to be a good choice.

The graph for Car Evaluation dataset using the knn package is shown below



From the graph above we see that k=3 seems to be a good choice.

The graph for Credit Approval dataset using the knn package is shown below



From the graph above we see that k=7 seems to be a good choice.

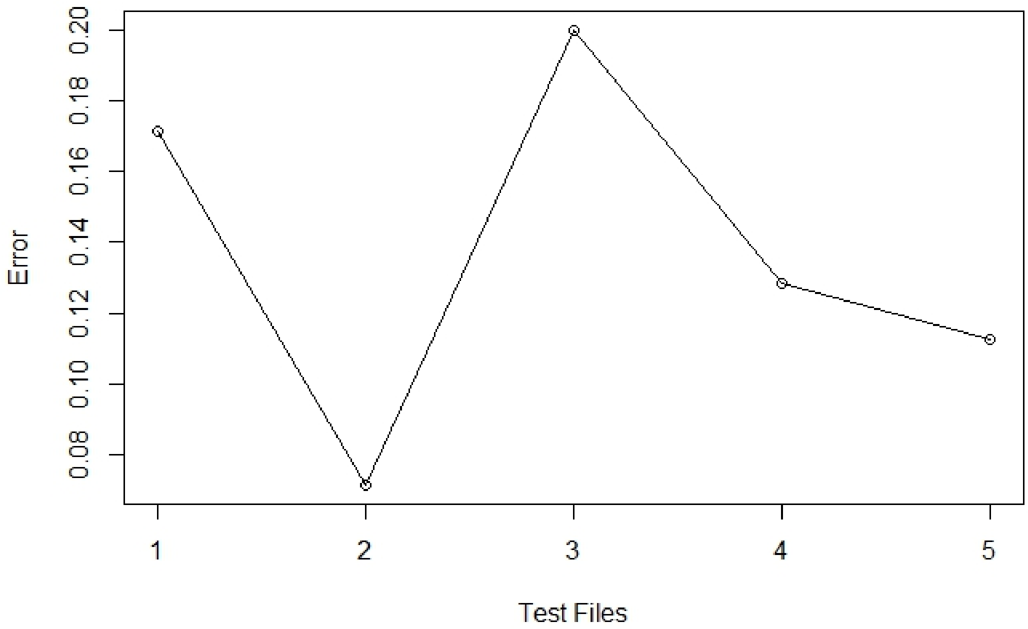
Problem 3: Naive Bayes Classifier [55 points]

3.1 Implement Naive Bayes classifier. The Pseudo-code for naive bayes algorithm is provided above. You may need to modify it for categorical variables. To handle unseen feature values, you may need to make use of m-estimate of conditional probability method. There are also other techniques, i.e., Laplace smoothing.

The m-estimate was used for the Car Evaluation data set since all the features were categorical in it. The features in the Credit Approval data set were converted to numeric features using as.numeric function in R. The results for each dataset is shown in 3.2

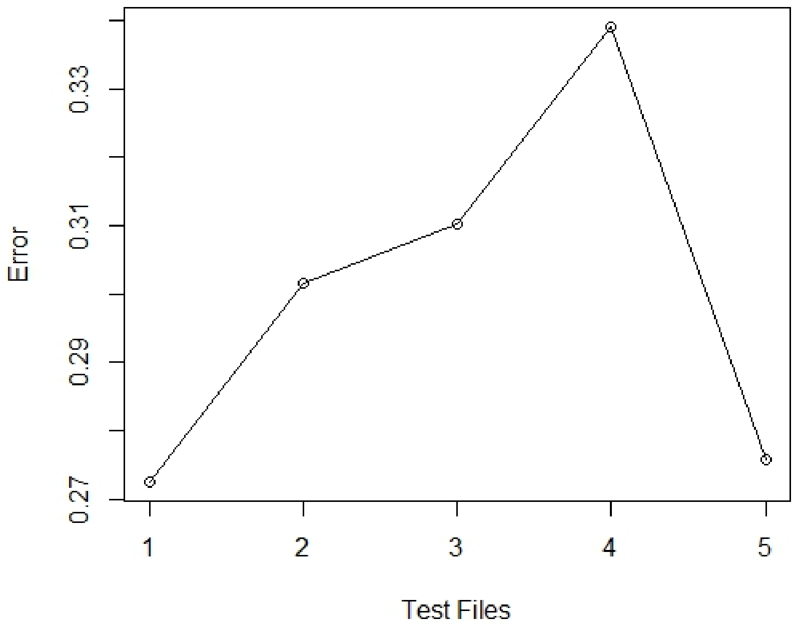
3.2 Train Naive Bayes classifiers over training data sets and test each classifier against corresponding test data. Make a plot that shows the error over each test data. Report the average error rate for 5-fold cross validation for each data sets.

Each of the datasets have 5 test errors. Each error is on its test data sets.
Find the graph below for the Ionosphere data set.



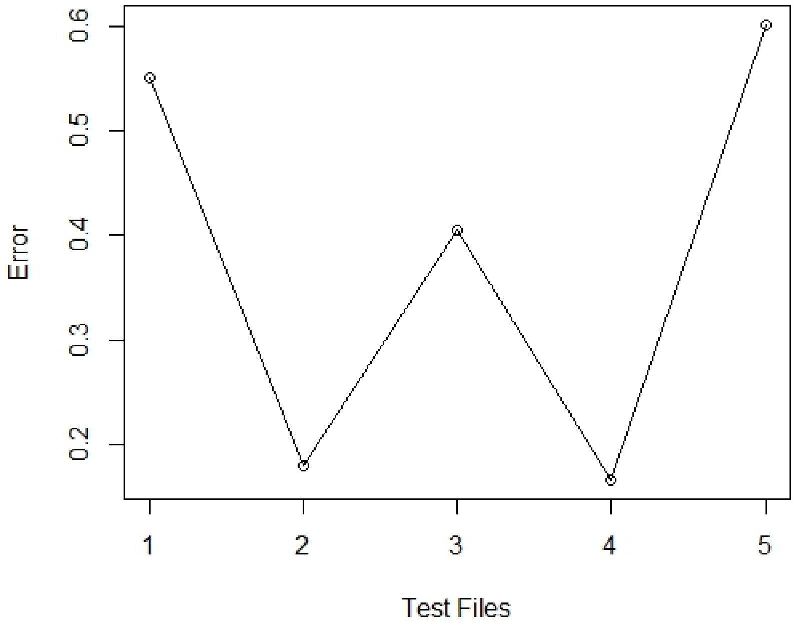
From the graph we see that the least error was for the second test set with an error of around 0.06.

Find the graph below for the Car Evaluation data set.



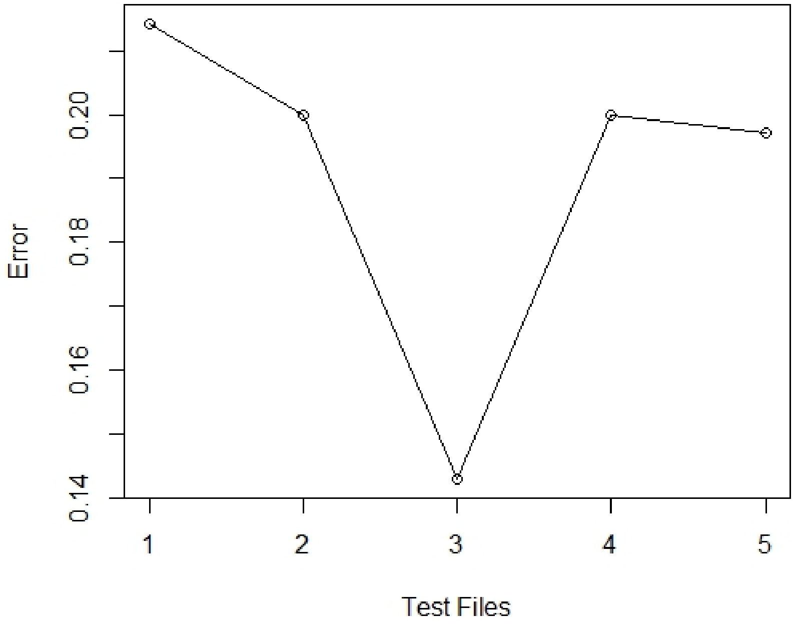
From the graph we see that the least error was for the first test set with an error of around 0.27.

Find the graph below for the Credit Approval data set.



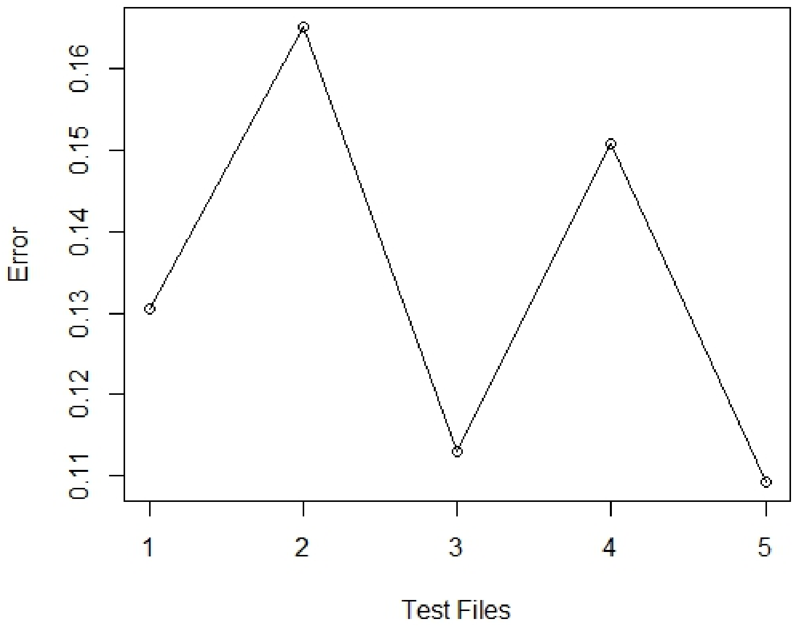
From the graph, the least error was for the fourth test set with an error of around 0.1.

3.3 Use Naive Bayes package in R for validation.
Find the graph below for the Ionosphere data set.



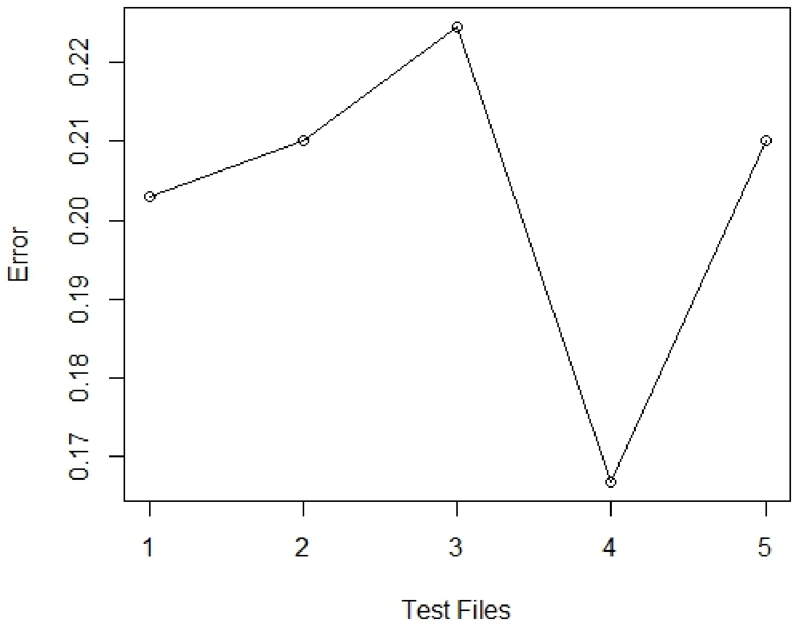
From the graph we see that the least error was for the third test set with an error of around 0.14.

Find the graph below for the Car Evaluation data set.



From the graph we see that the least error was for the fifth test set with an error of around 0.11.

Find the graph below for the Credit Approval data set.

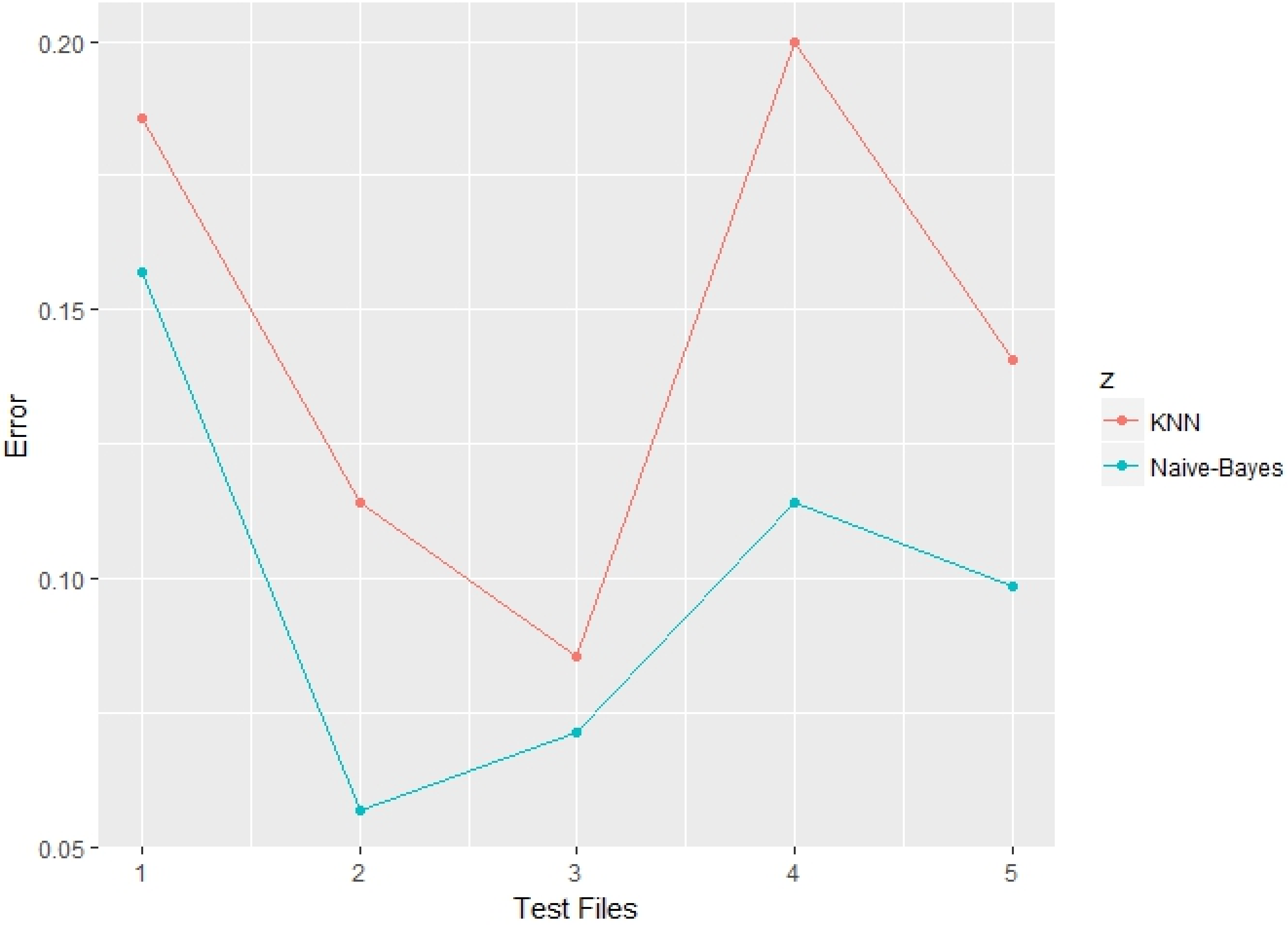


From the graph, the least error was for the fourth test set with an error of around 0.17.

Problem 4: Naive Bayes Classifier vs. K-Nearest Neighbors [30 points]

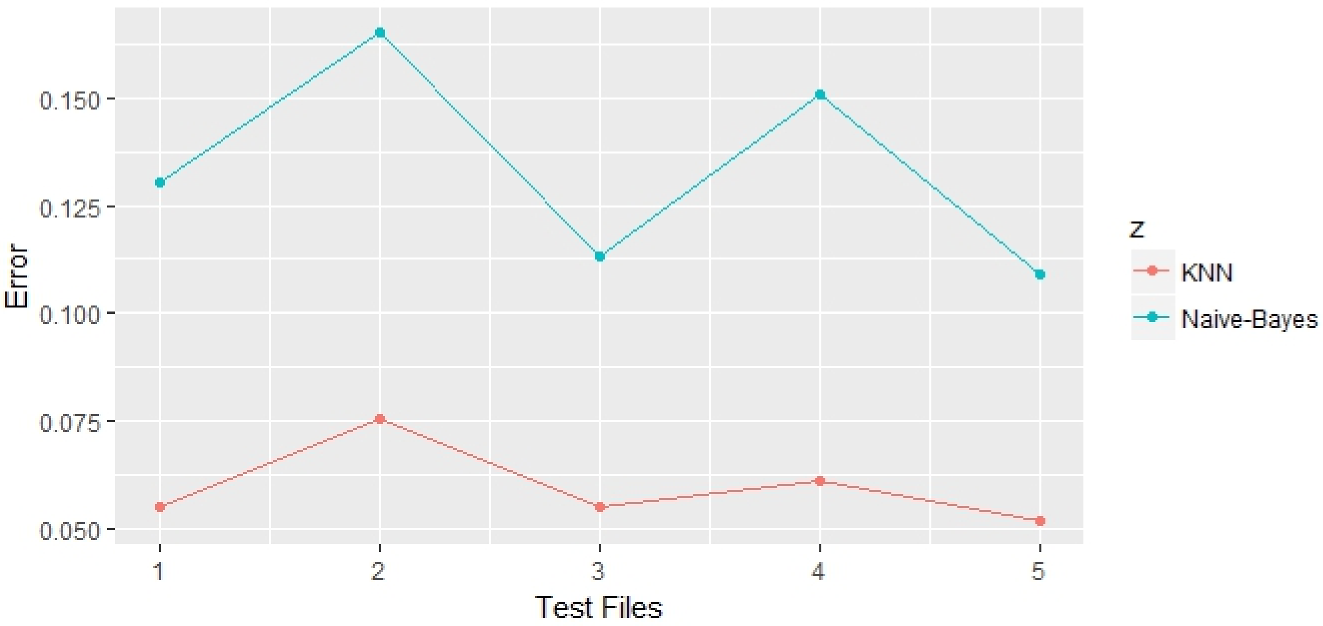
In this question, you are asked to compare Naive Bayes classifier with k-nn algorithm. First, determine the best KNN model for each data set. Then, Make a plot that reveals comparison of two algorithms using test error for each data set. (Total number of figures = 3)

For Ionosphere data set.
KNN gave best results for k=1. KNN was run for k=1 on 5 train and test set pairs. Naive Bayes was run on the same train and test set pairs.
Since there are 5 files, we get 5 test errors which are shown as 5 points in the graph for KNN and Naive Bayes both.



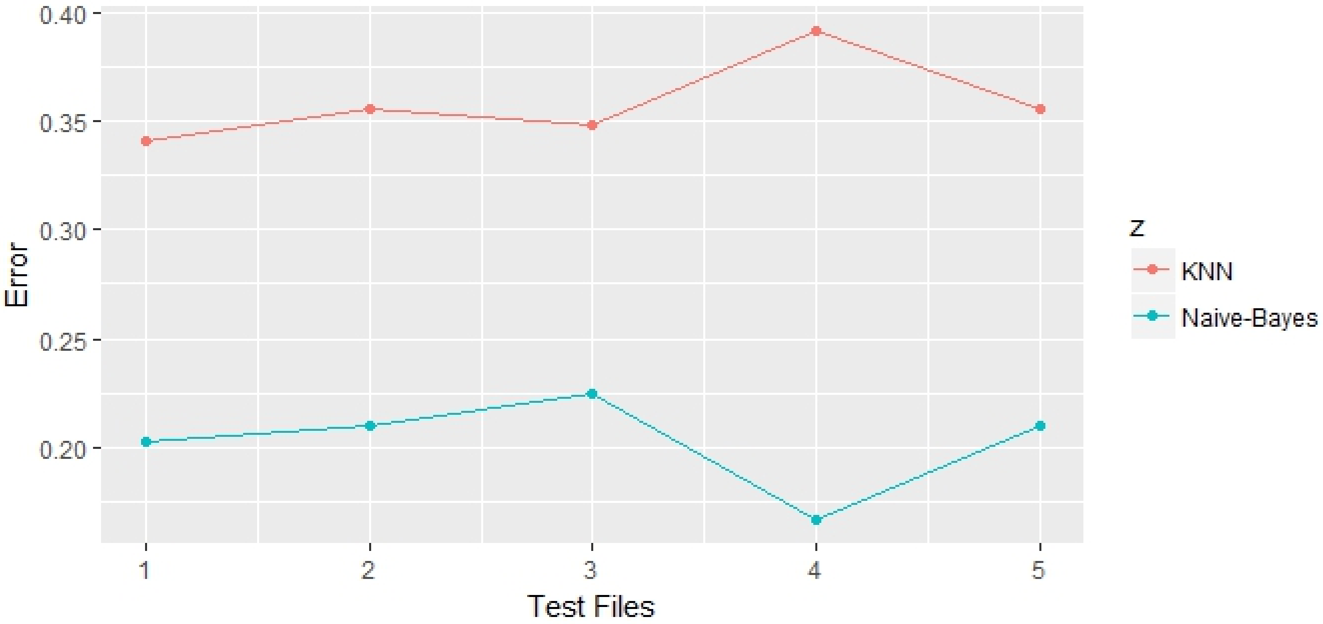
Looking at the graph, Naive Bayes clearly outperforms KNN for k=1.

For Car Evaluation data set.
KNN gave best results for k=3. KNN was run for k=3 on 5 train and test set pairs. Naive Bayes was run on the same train and test set pairs.
Since there are 5 files, we get 5 test errors which are shown as 5 points in the graph for KNN and Naive Bayes both.



Looking at the graph,KNN for k=3 clearly outperforms Naive Bayes.

For Credit Approval data set.
KNN gave best results for k=7. KNN was run for k=7 on 5 train and test set pairs. Naive Bayes was run on the same train and test set pairs.
Since there are 5 files, we get 5 test errors which are shown as 5 points in the graph for KNN and Naive Bayes both.

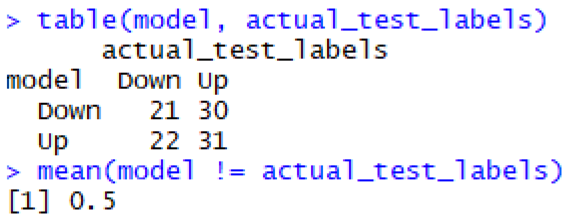


Looking at the graph, Naive Bayes clearly outperforms KNN for k=7.

Problem 5 [15 points]

From textbook, Chapter 4 exercise 10.g and 13 (only for k-nn and logistic regression)

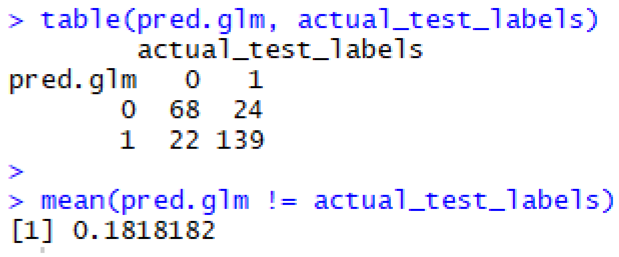
[**Exercise 10.g**]
Training data is between the years 1990 to 2008. Test data is between the years 2009 and 2010. For k=1 and "Lag2" as the only predictor, I got an error of 0.5 along with the below table. The primary diagonal elements indicate the right answers.



```
> table(model, actual_test_labels)
      actual_test_labels
model  Down Up
  Down   21 30
  Up     22 31
> mean(model != actual_test_labels)
[1] 0.5
```

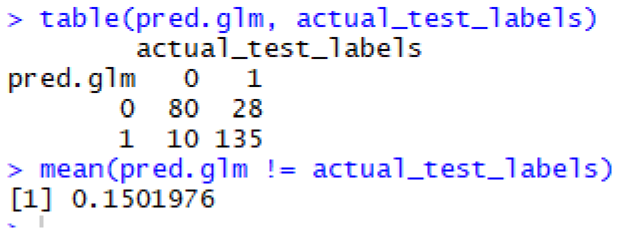
[**Exercise 13**]
Logistic Regression:
A new column "crime" was introduced and it is one if "crim" is greater than mean(crim) else it is zero. The training and test split was 50-50.

For the first logistic regression, I used all the features except "crim" and "crime" to predict "crime". I got a test error of 0.1818(18*percent*). Also, the table is shown below:



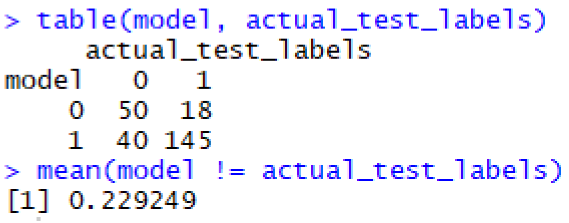
```
> table(pred.glm, actual_test_labels)
        actual_test_labels
pred.glm   0   1
       0  68  24
       1  22 139
>
> mean(pred.glm != actual_test_labels)
[1] 0.1818182
```

For the first logistic regression, I used all the features except "crim","zn",indus","chas","nox" and "crime" to predict "crime". I got a test error of 0.1501(15*percent*). Also, the table is shown below:



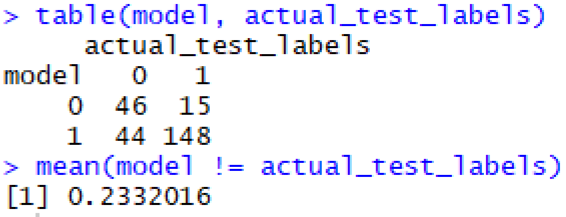
```
> table(pred.glm, actual_test_labels)
        actual_test_labels
pred.glm   0   1
       0  80  28
       1  10 135
> mean(pred.glm != actual_test_labels)
[1] 0.1501976
```

KNN:
Two different set of predictors were used and knn was run for k=3 and 5.
For the first approach, I used all the features except "crim" and "crime". Below are the results:
For k=3, I got an error of 0.2292 along with the following table



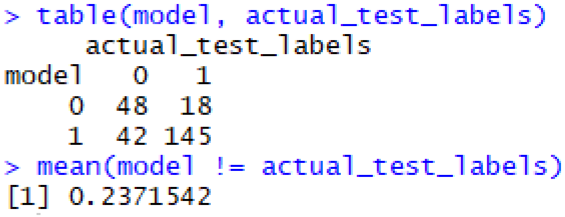
```
> table(model, actual_test_labels)
     actual_test_labels
model   0   1
    0  50  18
    1  40 145
> mean(model != actual_test_labels)
[1] 0.229249
```

For k=5, I got an error of 0.2332 along with the following table



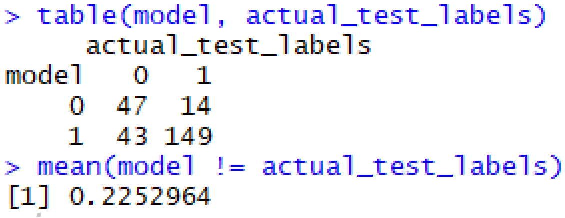
```
> table(model, actual_test_labels)
     actual_test_labels
model   0   1
    0  46  15
    1  44 148
> mean(model != actual_test_labels)
[1] 0.2332016
```

For the second approach, I used all the features except "crim","lstat", "medv" and "crime". Below are the results:
For k=3, I got an error of 0.2371 along with the following table



```
> table(model, actual_test_labels)
     actual_test_labels
model   0   1
    0  48  18
    1  42 145
> mean(model != actual_test_labels)
[1] 0.2371542
```

For k=5, I got an error of 0.2252 along with the following table



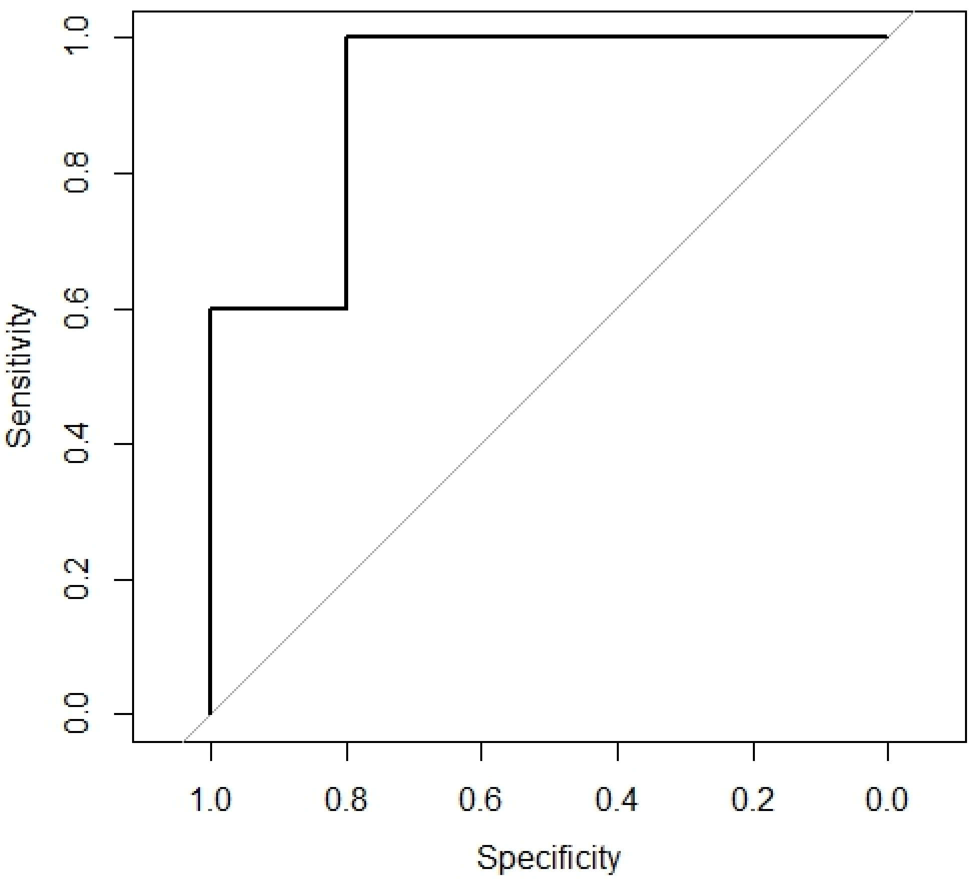
```
> table(model, actual_test_labels)
     actual_test_labels
model   0   1
    0  47  14
    1  43 149
> mean(model != actual_test_labels)
[1] 0.2252964
```

From the results, 0.2252 is the best score. This score was obtained in the second approach with k=5.

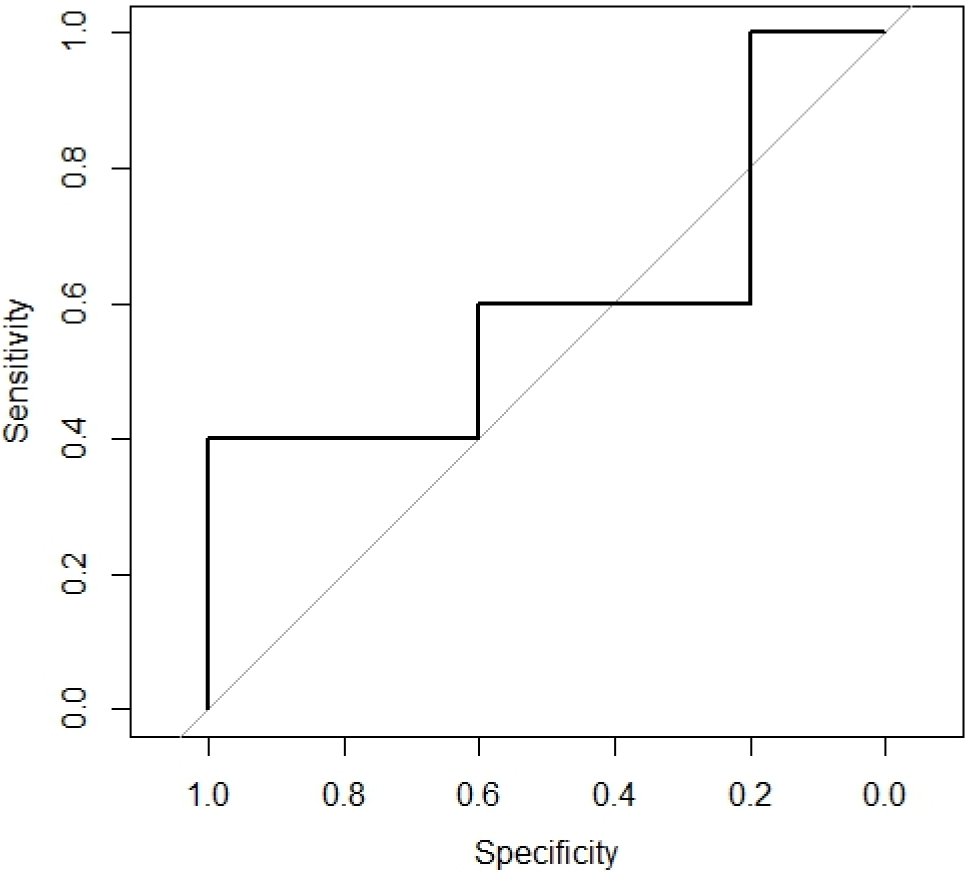
Extra credit (optional) [40 points]

1. You are asked to evaluate the performance of two classifier models, M_1 and M_2. The test set you have chosen contains 26 binary attributes, labeled as A through Z. The table below shows the posterior probabilities obtained by applying the models to the test set. (Only the posterior probabilities for the positive class are shown). As this is a two-class problem, P(-) = 1 - P(+) and P(-|A,...,Z) = 1 - P(+|A,...,Z). Assume that we are mostly interested in detecting instances from the positive class.

   (a) Plot the ROC curve for both M_1 and M_2. Which model do you think is better. Explain your reasons?
   Find the ROC plot for M_1 below



Find the ROC plot for M_1 below



From the plots, we see that AUC for M_1 is more and hence M_1 is a better model.

(b) For model M_1, suppose you choose the cutoff threshold to be $t = 0.5$. In other words, any test instances whose posterior probability is greater than t will be classified as a positive example. Compute the precision, recall, and F-measure for the model at this threshold value.

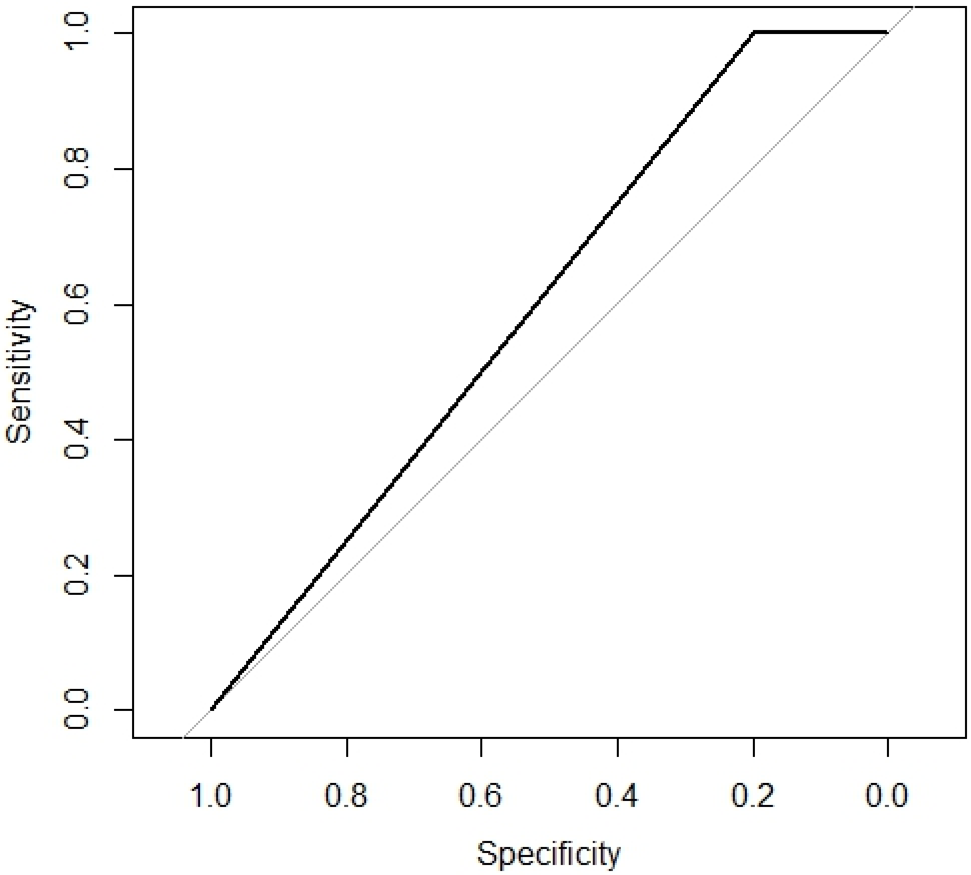
Precision for M_1 is : 0.75
Recall for M_1 is : 0.6
F-measure for M_1 is : 0.67

(c) Repeat the analysis for part (c) using the same cutoff threshold on model M_2. Compare the F-measure results for both models. Which model is better? Are the results consistent with what you expect from the ROC curve?

Precision for M_2 is : 0.5
Recall for M_2 is : 0.2
F-measure for M_2 is : 0.28
The results are consistent with the ROC curve results. The F-score of M_2 is very less compared to F-score of M_1.

(d) Repeat part (c) for model M_1 using the threshold $t = 0.1$. Which threshold do you prefer, $t = 0.5$ or $t = 0.1$? Are the results consistent with what you expect from ROC curve?

Precision for M_1 is : 0.55
Recall for M_1 is : 1
F-measure for M_1 is : 0.71
The threshold of $t = 0.1$ gives better F-score than for $t = 0.5$. This result is not clearly evident from the ROC shown below with $t = 0.1$.



2. Student/s who design/s the best either Naives Bayes classifier or KNN algorithm for the given data sets will receive 20 points.

What to Turn-in (Submission Instructions)

Put the below files in a zipped folder for your submission. The zipped folder should be named as "usename-section number", i.e., hakurban-P556

1. The *tex and *pdf of the written answers to this document.
2. Code and Data
   (a) Question 1: crossValidation.R, output of cross validation: training and test data sets
   (b) Question 2.1: knn.R, Question 2.3: knnValidation.R
   (c) Question 3.1: naiveBayes.R, Question 3.3: naiveBayes-Validation.R
3. A README file that explains how to run your code and other files in the folder