CIS 490 MACHINE LEARNING HOMEWORK 2

(Full Score: **45** + 4 bonus points)

October 13, 2017 Fall

Group number: 08

Group leader: Kevin D’Cruz (G) (01656193)

Group members and student IDs: Chetan Kumar (G) (01672511)

Abhishek Manoj Kumar (G) (01675536)

Madhuri Gawali (G) (01654141)

Apoorva ShivaShankar (G) ()

(Please mark “G” after names for Graduate Students in your group)

INSTRUCTOR: Dr. Julia Hua Fang

**Please write your answers for each question:**

1. **4 points.**
2. use the multiple linear regression model to fit the “Demand estimation” data below (you can copy and paste the data into a file and save it as, e.g., \*.txt or \*.xlsx); compute SSE, MSE; and predict the number of weekly riders of a new city, given its known values on X1 = $35, X2=$1,600,000, X3 = $12,000 and X4= $ 110.

Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as “Group#\_Reg.R” (e.g., Group1\_Reg.R) or “Group#\_Reg.m” (e.g., “Group1\_Reg.m”).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Y** | **X1** | **X2** | **X3** | **X4** |
| **City** | **Number of weekly riders** | **Price per week** | **Population of city** | **Monthly income of riders** | **Average parking rates per month** |
| 1 | 192,000 | $15 | 1,800,000 | $5,800 | $50 |
| 2 | 190,400 | $15 | 1,790,000 | $6,200 | $50 |
| 3 | 191,200 | $15 | 1,780,000 | $6,400 | $60 |
| 4 | 177,600 | $25 | 1,778,000 | $6,500 | $60 |
| 5 | 176,800 | $25 | 1,750,000 | $6,550 | $60 |
| 6 | 178,400 | $25 | 1,740,000 | $6,580 | $70 |
| 7 | 180,800 | $25 | 1,725,000 | $8,200 | $75 |
| 8 | 175,200 | $30 | 1,725,000 | $8,600 | $75 |
| 9 | 174,400 | $30 | 1,720,000 | $8,800 | $75 |
| 10 | 173,920 | $30 | 1,705,000 | $9,200 | $80 |
| 11 | 172,800 | $30 | 1,710,000 | $9,630 | $80 |
| 12 | 163,200 | $40 | 1,700,000 | $10,570 | $80 |
| 13 | 161,600 | $40 | 1,695,000 | $11,330 | $85 |
| 14 | 161,600 | $40 | 1,695,000 | $11,600 | $100 |
| 15 | 160,800 | $40 | 1,690,000 | $11,800 | $105 |
| 16 | 159,200 | $40 | 1,630,000 | $11,830 | $105 |
| 17 | 148,800 | $65 | 1,640,000 | $12,650 | $105 |
| 18 | 115,696 | $102 | 1,635,000 | $13,000 | $110 |
| 19 | 147,200 | $75 | 1,630,000 | $13,224 | $125 |
| 20 | 150,400 | $75 | 1,620,000 | $13,766 | $130 |
| 21 | 152,000 | $75 | 1,615,000 | $14,010 | $150 |
| 22 | 136,000 | $80 | 1,605,000 | $14,468 | $155 |
| 23 | 126,240 | $86 | 1,590,000 | $15,000 | $165 |
| 24 | 123,888 | $98 | 1,595,000 | $15,200 | $175 |
| 25 | 126,080 | $87 | 1,590,000 | $15,600 | $175 |
| 26 | 151,680 | $77 | 1,600,000 | $16,000 | $190 |
| 27 | 152,800 | $63 | 1,610,000 | $16,200 | $200 |

**Solution:** The number of weekly riders of a new city, given its known values on X1 = $35, X2=$1,600,000, X3 = $12,000 and X4= $ 110 is 190056.7

SSE: 630848248

MSE: 23364750

|  |
| --- |
| #Importing the Dataset  library(readxl)  Demand\_Estimation <- read\_excel("C:/Users/Kevin D'Cruz/Downloads/UMass/Sem 3/Machine Learning/Assignment 2/Demand Estimation.xlsx")  View(Demand\_Estimation)  head(Demand\_Estimation)  summary(Demand\_Estimation)  Demand\_Estimation <- data.frame(Demand\_Estimation)  model1 <- lm(Demand\_Estimation$Number.of.weekly.riders ~ ., data = Demand\_Estimation)  summary(model1)  str(Demand\_Estimation)  #Creating A Dataframe  newdf = data.frame(City = numeric(1), Price.per.week = numeric(1), Population.of.city = numeric(1), Monthly.income.of.riders = numeric(1), Average.parking.rates.per.month = numeric(1) )  newdf$City = 1  newdf$Price.per.week = 35  newdf$Population.of.city = 1600000  newdf$Monthly.income.of.riders = 12000  newdf$Average.parking.rates.per.month = 110  predict(model1, newdf)  #Computing SSE  SSE <-sum(model1$residuals^2)  SSE  #Computing MSE  MSE <-mean(model1$residuals^2)  MSE |

1 Bonus point: Plot a graph where you have a best fit line for this dataset.

Graph:

#Plotting Best Fit

plot(model1, pch=16,which = 1)

A close up of a map

Description generated with very high confidence

**Output Code:**

|  |
| --- |
| > library(readxl)  > Demand\_Estimation <- read\_excel("C:/Users/Kevin D'Cruz/Downloads/UMass/Sem 3/Machine Learning/Assignment 2/Demand Estimation.xlsx")  > View(Demand\_Estimation)  > head(Demand\_Estimation)  # A tibble: 6 x 6  City `Number of weekly riders` `Price per week` `Population of city` `Monthly income of riders`  <dbl> <dbl> <dbl> <dbl> <dbl>  1 1 192000 15 1800000 5800  2 2 190400 15 1790000 6200  3 3 191200 15 1780000 6400  4 4 177600 25 1778000 6500  5 5 176800 25 1750000 6550  6 6 178400 25 1740000 6580  # ... with 1 more variables: `Average parking rates per month` <dbl>  > summary(Demand\_Estimation)  City Number of weekly riders Price per week Population of city Monthly income of riders  Min. : 1.0 Min. :115696 Min. : 15.00 Min. :1590000 Min. : 5800  1st Qu.: 7.5 1st Qu.:149600 1st Qu.: 27.50 1st Qu.:1617500 1st Qu.: 8400  Median :14.0 Median :161600 Median : 40.00 Median :1695000 Median :11600  Mean :14.0 Mean :160026 Mean : 49.93 Mean :1680111 Mean :11063  3rd Qu.:20.5 3rd Qu.:176000 3rd Qu.: 75.00 3rd Qu.:1725000 3rd Qu.:13888  Max. :27.0 Max. :192000 Max. :102.00 Max. :1800000 Max. :16200  Average parking rates per month  Min. : 50  1st Qu.: 75  Median :100  Mean :107  3rd Qu.:140  Max. :200  > Demand\_Estimation <- data.frame(Demand\_Estimation)  > model1 <- lm(Demand\_Estimation$Number.of.weekly.riders ~ ., data = Demand\_Estimation)  > summary(model1)  Call:  lm(formula = Demand\_Estimation$Number.of.weekly.riders ~ ., data = Demand\_Estimation)  Residuals:  Min 1Q Median 3Q Max  -8831.7 -3194.1 -918.8 1952.5 11016.2  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 1.598e+05 1.665e+05 0.960 0.3482  City -1.503e+03 2.360e+03 -0.637 0.5311  Price.per.week -6.979e+02 9.762e+01 -7.150 4.75e-07 \*\*\*  Population.of.city 1.520e-02 9.629e-02 0.158 0.8761  Monthly.income.of.riders 7.002e-01 3.549e+00 0.197 0.8455  Average.parking.rates.per.month 2.133e+02 1.214e+02 1.757 0.0935 .  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 5481 on 21 degrees of freedom  Multiple R-squared: 0.9465, Adjusted R-squared: 0.9338  F-statistic: 74.34 on 5 and 21 DF, p-value: 1.242e-12  > str(Demand\_Estimation)  'data.frame': 27 obs. of 6 variables:  $ City : num 1 2 3 4 5 6 7 8 9 10 ...  $ Number.of.weekly.riders : num 192000 190400 191200 177600 176800 ...  $ Price.per.week : num 15 15 15 25 25 25 25 30 30 30 ...  $ Population.of.city : num 1800000 1790000 1780000 1778000 1750000 ...  $ Monthly.income.of.riders : num 5800 6200 6400 6500 6550 6580 8200 8600 8800 9200 ...  $ Average.parking.rates.per.month: num 50 50 60 60 60 70 75 75 75 80 ...  > #Creating A Dataframe  > newdf = data.frame(City = numeric(1), Price.per.week = numeric(1), Population.of.city = numeric(1), Monthly.income.of.riders = numeric(1), Average.parking.rates.per.month = numeric(1) )  > newdf$City = 1  > newdf$Price.per.week = 35  > newdf$Population.of.city = 1600000  > newdf$Monthly.income.of.riders = 12000  > newdf$Average.parking.rates.per.month = 110  > predict(model1, newdf)  1  190056.7  > #Computing SSE  > SSE <-sum(model1$residuals^2)  > SSE  [1] 630848248  > #Computing MSE  > MSE <-mean(model1$residuals^2)  > MSE  [1] 23364750  > #Plotting Best Fit  > plot(model1, pch=16,which = 1) |

1. **8 points**

Use logistic regression model to fit the “Graduate Admission” data, in “logistic.txt” (see the data file under Homework2 in Week 6 at myCourses)

Scenario: A researcher is interested in how three variables, (X1=GRE, Graduate Record Exam scores; X2=GPA, grade point average; and X3: Rank, prestige of the undergraduate institution) effect admission into graduate school (the response variable, Y, admit/don’t admit, is a binary variable).

|  |
| --- |
| #read file named logistic  data <- read.table(file.choose(),header =T)  #data  #factor variables  data$admit <- as.factor(data$admit)  data$rank <- as.factor(data$rank)  set.seed(1234)  ind <- sample(2, nrow(data), replace = T, prob = c(0.8, 0.2))  #Logistic Regression  lrmodel <- glm(admit ~ . , data = data, family = "binomial")  summary(lrmodel)  ##confusion matrix  p3 <- predict(lrmodel, data, type = "response")  p3 <- ifelse(p3>0.5,1,0)  (tab3 <- table(predicted = p3, Actual = data$admit))  #prediction if student will get admit or not Student will be admitted or not  newdf = data.frame( gre = numeric(1), gpa = double(1), rank = numeric(1) )  newdf$gre = 720  newdf$gpa = 3.8  newdf$rank= 2  newdf$rank = as.factor(newdf$rank)  str(newdf)  predict(lrmodel, newdata = newdf, interval = 'confidence') |

1. If a new student has X1 = 720; X2=3.8; X3= 2, do you admit or not admit him? (1 point)

Solution: From the output of the code above, the output is as follows:

1

0.02030725

This shows that, the new student gets an admit.

1. Fill in the number of students in the confusion matrix (2 points)

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Observed/True (Y) | 0= don’t admit | 1= Admit |
| 0= don’t admit | 254 | 97 |
| 1= Admit | 19 | 30 |

1. Compute misclassification errors, sensitivity and specificity. (3 points)

Please attach your R or Matlab code and output in your word file as Appendices. Please also save and name your scripts as “Group#\_Logit.R” or “Group#\_Logit.m” and any related R or Matlab output files with the same file names, e.g “Group#\_Logit.xxx”.

Solution:

#Misclassification rate

1 - sum(diag(tab3))/sum(tab3)

0.29

Sensitivity from the confusion matrix obtained is: 97

Specificity from the confusion matrix: 254

Output:

|  |
| --- |
| > data <- read.table(file.choose(),header =T)  > data$admit <- as.factor(data$admit)  > data$rank <- as.factor(data$rank)  > set.seed(1234)  > ind <- sample(2, nrow(data), replace = T, prob = c(0.8, 0.2))  > lrmodel <- glm(admit ~ . , data = data, family = "binomial")  > summary(lrmodel)  Call:  glm(formula = admit ~ ., family = "binomial", data = data)  Deviance Residuals:  Min 1Q Median 3Q Max  -1.6268 -0.8662 -0.6388 1.1490 2.0790  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -3.989979 1.139951 -3.500 0.000465 \*  gre 0.002264 0.001094 2.070 0.038465 \*  gpa 0.804038 0.331819 2.423 0.015388 \*  rank2 -0.675443 0.316490 -2.134 0.032829 \*  rank3 -1.340204 0.345306 -3.881 0.000104 \*  rank4 -1.551464 0.417832 -3.713 0.000205 \*  ---  Signif. codes: 0 ‘\*’ 0.001 ‘\*’ 0.01 ‘’ 0.05 ‘.’ 0.1 ‘ ’ 1  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 499.98 on 399 degrees of freedom  Residual deviance: 458.52 on 394 degrees of freedom  AIC: 470.52  Number of Fisher Scoring iterations: 4  > p3 <- predict(lrmodel, data, type = "response")  > p3 <- ifelse(p3>0.5,1,0)  > (tab3 <- table(predicted = p3, Actual = data$admit))  Actual  predicted 0 1  0 254 97  1 19 30  > 1 - sum(diag(tab3))/sum(tab3)  [1] 0.29  > newdf = data.frame( gre = numeric(1), gpa = double(1), rank = numeric(1) )  > newdf$gre = 720  > newdf$gpa = 3.8  > newdf$rank= 2  > newdf$rank = as.factor(newdf$rank)  > str(newdf)  'data.frame': 1 obs. of 3 variables:  $ gre : num 720  $ gpa : num 3.8  $ rank: Factor w/ 1 level "2": 1  > predict(lrmodel, newdata = newdf, interval = 'confidence')  1  0.02030725 |

1. What’s relationship among probability, odds and log odds? Use an example to demonstrate their relationship. What’s the range for probability, odds and log odds? (2 points)

Solution:

**Probability:** It gives us the fraction of times an event might occur in any number of trails, and it ranges from 0 to 1.

**Odds:** Odds are found by dividing the probability that the event will occur by probability that the event won't occur.

Example, consider that the probability of a event occurring is X= 0.8(i.e. 80%), then the probability of event not occurring is: Y= 1-0.8 = 0.2(i.e. 20 %)

Odds is given by, odds = X/Y

=0.8/1-0.8

=4/1

The problem of odds ratio is that, for small to moderate sample size the sampling distribution is highly skewed. To overcome the problem, we use Log Odds ratio.

**Log odds:** They provide another way of expressing probability, intended to simplify the process of updating them with new data or evidence.

It is difficult to convert probability to log odds. Log odds is the log of odds ratio. That is , Log odds = log(Odds) So, from the above example Log odd ratio is, Log odds = log(Odds) Log odds =log(1.5)

1. 8 **points**
2. What is Bias-variance trade off? (0.5 point)

Solution:

**Bias-variance tradeoff** –It is the problem in supervised learning. The problem takes place when two sources of error prevent supervised learning algorithm from simplifying beyond their training set.

1. **Bias**

Bias is an error from inaccurate hypothesis. High bias can cause an algorithm to omission the relevant relations between structures and goal output

1. **Variance**

Variance is an error from compassion to small variability in the training set. High variance can cause an algorithm to establish the random noise in the training data, rather than the proposed outputs

1. What is overfitting? (0.5 point)

Solution:

**Overfitting:** It takes place when the model is extremely complex, such as having too many parameters relative to the numbers of observations. Overfitted model have poor predictive performance, because it exaggerates to slight variations in the training data. The existence of overfitting depends on the measures used for training the model, but it is not the same as the measures used to magistrate the usefulness of a model. The model is classically trained by maximizing its piece on some set of training data and its usefulness is determine not by its piece on the training data but by its capacity to perform well on unseen data.

1. What are the two common types of regularization in linear regression? (0.5 point)

Solution:

1. **Ridge Regression**

Ridge regression is a practice for scrutinizing multiple regression data that agonize from multi regression. The multi regression take place when the smallest squares assessments are impartial, as their discrepancy are large, so they may be far from the true value, ridge regression castigates the size of the regression coefficients, the ridge regression coefficient assessment is defined as the value of ***β*** that minimizes

1. **Lasso Regression**

Lasso regression is a method that accomplishes both variable collection and control because they increase the calculation, precision and accountability of the statistical model, Lasso is slightest complete contraction and collection operator. Its coefficients are the solution to the optimization problem.

Minimize

Such that

1. Write out the objective functions for these two regularization types. Please identify which term is for SSR and which term is for shrinkage penalty term in each function. What is λ called in these regularized regressions? (2 points)

Solution:

1. **Ridge Regression**
2. **Lasso Regression**

Minimize

Such that

1. Write out the corollaries of Ridge Regression. (0.5 point)

Solution: The solution for regular regression problem is assumed by

Therefore,

Therefore, Corollary:

2. what is the major difference between the two regularization types? (hint: why one of them is preferred in real world practice) (1 point)

Solution:

|  |  |
| --- | --- |
| Ridge Regression | Lasso regression |
| In rigid regression, the drawback is the sum of square of coefficient | In lasso it’s the sum of the complete value of the coefficient |
| Unsteady Solution | Constant Solution |
| Multiple solutions may be possible | Only one solution |
| This is the regularization method and uses l2 regularization | This is the regularization method that uses l1 regularization |
| It contracts the value of coefficient but does not reaches zero, which suggests no feature selection | It contracts coefficient to zero which certainly help in feature selection |

1. Download credit data from under Homework2 in Week 6 at myCourses and use Lasso regression with cross-validation to select a parsimonious model (Occam’s Razor), ie. what attributes will enter into your final Lasso regression model and what optimal value will you pick for λ? Please replicate the following two graphs. (3 points)

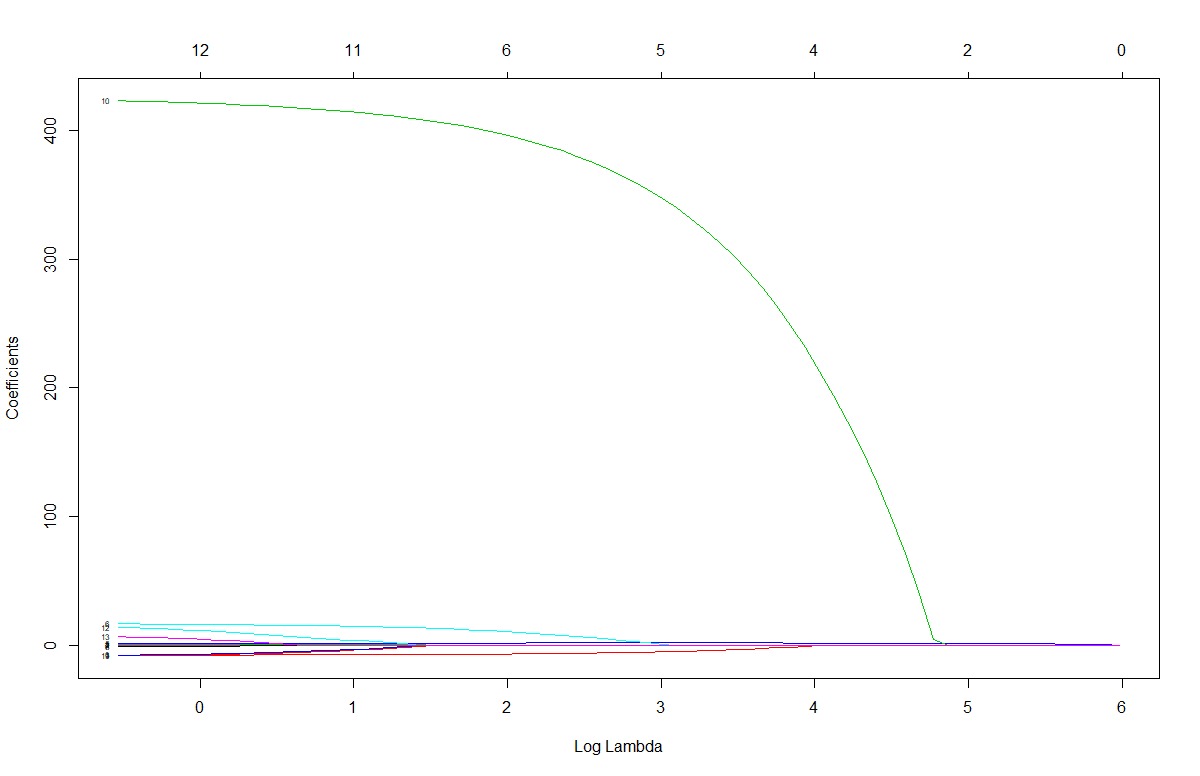
Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as “Group#\_Lasso.R” (e.g., Group1\_Lasso.R) or “Group#\_Lasso.m” (e.g., “Group1\_Lasso.m”).

Solution(Code):

|  |
| --- |
| library(glmnet)  data <- read.csv(file.choose(), header= T)  x=model.matrix(Balance~.,data=data)  y <- as.double(as.matrix(data[, 12])) # Only class  #head(y)  set.seed(999)  cv.lasso <- cv.glmnet(x, y, alpha=1, parallel=TRUE, standardize=TRUE, type.measure='auc')  plot(cv.lasso)  plot(cv.lasso$glmnet.fit, xvar="lambda", label=TRUE)  cv.lasso$lambda.min  cv.lasso$lambda.1se  coef(cv.lasso, s=cv.lasso$lambda.1se)  coef <- coef(cv.lasso, s='lambda.1se')  coef |

Graph:



Output:

|  |
| --- |
| library(glmnet)  > data <- read.csv(file.choose(), header= T)  > x=model.matrix(Balance~.,data=data)  > y <- as.double(as.matrix(data[, 12])) # Only class  > set.seed(999)  > cv.lasso <- cv.glmnet(x, y, alpha=1, parallel=TRUE, standardize=TRUE, type.measure='auc')  Warning message:  In cv.elnet(list(list(a0 = c(535.825, 454.345391736194, 380.104207303334, :  Only 'mse', 'deviance' or 'mae' available for Gaussian models; 'mse' used  > plot(cv.lasso)  > plot(cv.lasso$glmnet.fit, xvar="lambda", label=TRUE)  > cv.lasso$lambda.min  [1] 0.6462984  > cv.lasso$lambda.1se  [1] 8.744726  > coef(cv.lasso, s=cv.lasso$lambda.1se)  14 x 1 sparse Matrix of class "dgCMatrix"  1  (Intercept) -473.4815857  (Intercept) .  X .  Income -6.6522100  Limit 0.1342854  Rating 1.7122347  Cards 9.2763395  Age -0.2859738  Education .  GenderFemale .  StudentYes 391.2431284  MarriedYes .  EthnicityAsian .  EthnicityCaucasian . |

Here we get 2 values of λ , i.e. min = 0.6462984 and 1se = 8.744726. values closer to zero signify the unimportant variables.

1. **7 points**
2. What are the differences between parametric and non-parametric methods? (1.5 point)

Solution:

|  |  |
| --- | --- |
| Parametric method | Non-parametric method |
| Parametric method makes several hypotheses about parameters | Non-parametric method does not make several hypotheses about parameters. |
| The inhabitants have the same changes. | Variant under study has inherent stability. |
| The parametric method emphases on the nasty difference and corresponding non-parametric test emphases on the difference between means | In non- parametric method records are changed from scores to levels or symbols |
| In parametric method numerical method trust on hypothesis about the shape of the distribution | In non-parametric method numerical method trust on few hypothesis about the outline of the population distribution from which the model was drawn. |

1. What is called curse of dimensionality? (0.5 point)

Solution:

Curse of dimensionality is a process that take place at the time of examining and unifying data in high-dimensional location, they do not follow in low-dimensional location, it is somewhat ambiguous in definition as it describes different but related things in different disciplines.

1. Why should we consider rescaling the attributes when using KNN? (0.5 point)

Solution:

We consider rescaling the attribute data to value so that we can scale the range in [0,1] or [-1,1] which is useful for the optimizing algorithms and it is useful for algorithms that use distance measurement such as KNN.

1. What options can we use to determine the class from a nearest neighbor list? (0.5 point)

Solution:

The options we can use to determine the class from a nearest neighbor list is:

i. To take Mainstream division of class labels among the k-nearest neighbors.

ii. To balance the division according to distance

Weight factor, w=1/

1. What method can we use to choose the number of k for nearest neighbor classification (0.5 point)

Solution:

i. **k** should be large so that error rate is minimized, if **k** is too small it will lead to noisy decision boundaries.

ii. **k** should be small enough so that only nearby models are involved.

if **k** is too large it will lead to over-smoothed limitations

(d) Given data below, compute the distance between nominal attribute values: d(Married,Divorced). Please show you calculation steps below. (1 point)



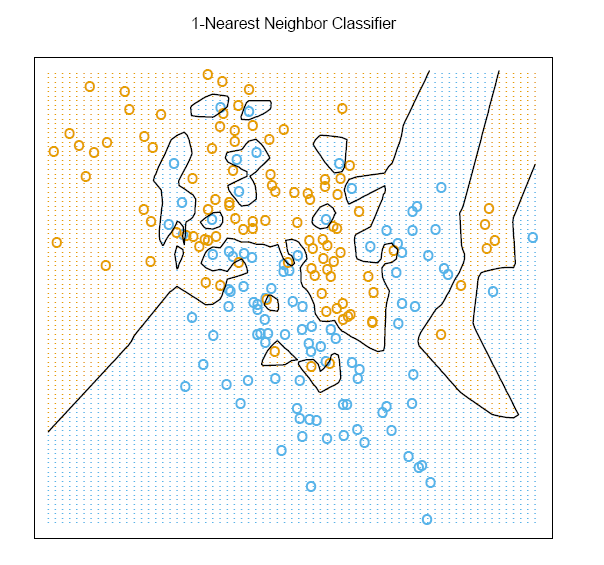
Solution:

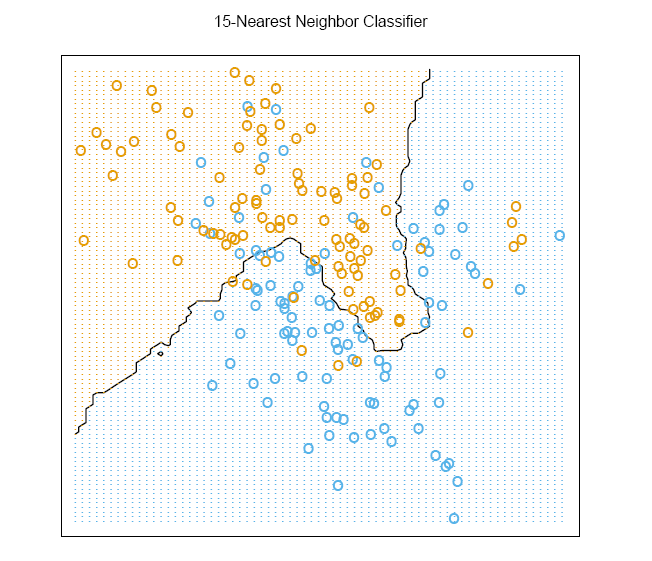


d(Married,Divorced):

=| - + | -

= 1

1. KNN is a local or global method? (0.5 point) Solution:

Which decision boundary in the graph seems to derive from more nearest neighbors? (0.5 point)

(i) (ii)

Solution:

KNN is a local method because it is the simple and active method that guesses the class label of a query model based on information contained in its neighborhood

(i) is the decision boundary in the graph which seems to derive from more nearest neighbors.

(g) Please download IRIS data from UCI machine learning repository (<https://archive.ics.uci.edu/ml/datasets.html> ), and run KNN on IRIS data, compute the misclassification error rates. (1.5 points)

Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as “Group#\_KNN.R” (e.g., Group1\_KNN.R) or “Group#\_KNN.m” (e.g., “Group1\_IRIS.m”). ( points)

Solution:

|  |
| --- |
| data("iris")  head(iris,10)  set.seed(1234)  ind <- sample(2, nrow(iris), replace = T, prob = c(0.8, 0.2))  ind  train\_data <- iris[ind == 1, 1:4]  test\_data <- iris[ind==2, 1:4]  train\_label <- iris[ind == 1, 5]  test\_label <- iris[ind==2, 5]  library(class)  iris\_test\_pred<-knn(train=train\_data, test=test\_data,cl=train\_label,k=13)  iris\_test\_pred  tab1 <- table(predicted = iris\_test\_pred, Actual = test\_label)  tab1  accuracy <- sum(diag(tab1)) / sum(tab1)  accuracy  misclassification\_error\_rate = 1 - accuracy  misclassification\_error\_rate |

Since there are very few cases in the test data, we get 100% accuracy and a misclassification rate of 0.

Output:

|  |
| --- |
| data("iris")  > head(iris,10)  Sepal.Length Sepal.Width Petal.Length Petal.Width Species  1 5.1 3.5 1.4 0.2 setosa  2 4.9 3.0 1.4 0.2 setosa  3 4.7 3.2 1.3 0.2 setosa  4 4.6 3.1 1.5 0.2 setosa  5 5.0 3.6 1.4 0.2 setosa  6 5.4 3.9 1.7 0.4 setosa  7 4.6 3.4 1.4 0.3 setosa  8 5.0 3.4 1.5 0.2 setosa  9 4.4 2.9 1.4 0.2 setosa  10 4.9 3.1 1.5 0.1 setosa  > set.seed(1234)  > ind <- sample(2, nrow(iris), replace = T, prob = c(0.8, 0.2))  > ind  [1] 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1  [46] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2  [91] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 1 1 1 2 2 2 1 1 1 1 1 1 2 1 1 1 2  [136] 1 2 1 1 2 1 2 1 1 1 1 1 1 2 1  > train\_data <- iris[ind == 1, 1:4]  > test\_data <- iris[ind==2, 1:4]  > train\_label <- iris[ind == 1, 5]  > test\_label <- iris[ind==2, 5]  > library(class)  > iris\_test\_pred<-knn(train=train\_data, test=test\_data,cl=train\_label,k=13)  > iris\_test\_pred  [1] setosa setosa setosa setosa setosa setosa setosa setosa  [9] versicolor versicolor versicolor versicolor versicolor versicolor versicolor virginica  [17] virginica virginica virginica virginica virginica virginica virginica virginica  [25] virginica virginica virginica  Levels: setosa versicolor virginica  > tab1 <- table(predicted = iris\_test\_pred, Actual = test\_label)  > tab1  Actual  predicted setosa versicolor virginica  setosa 8 0 0  versicolor 0 7 0  virginica 0 0 12  > accuracy <- sum(diag(tab1)) / sum(tab1)  > accuracy  [1] 1  > misclassification\_error\_rate = 1 - accuracy  > misclassification\_error\_rate  [1] 0 |

2 Bonus Points: Using the same IRIS data, use cross validation to choose the optimal number of nearest neighbors for KNN. Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as “Group#\_KNNCV.R” (e.g., Group1\_KNNCV.R) or “Group#\_KNNCV.m” (e.g., “Group1\_KNNCV.m”).

Solution(Code):

|  |
| --- |
| iris\_test\_pred2<-knn.cv(train=train\_data,cl=train\_label,k=13)  tab2 <- table(predicted = iris\_test\_pred2, Actual = train\_label)  tab2 |

Output:

|  |
| --- |
| Actual  predicted setosa versicolor virginica  setosa 42 0 0  versicolor 0 42 2  virginica 0 1 36 |

After varying the k value, we get the least classification error for k=13

1. **6 points**
2. What are the differences between Frequentist and Bayesian approaches? (2 point)

Solution:

|  |  |
| --- | --- |
| Frequentist | Bayesian |
| In frequentist data are repeatable unplanned sample | In Bayesian data are observed from the comprehended sample |
| Parameters are fixed | Data are fixed |
| It does not use prospects for hypothesis | It uses prospects for both hypothesis and data |
| It depends on the possibility for both practical and unseen data | It depends on the prior and possibility of practical data |
| Does not require a previous | Require one to know a particular preceding |
| It is less computationally intensive | It is computationally intensive due to integration over many parameters |

(b) What are the four terms called in the following equation from Bayesian classification approach. (1 point)



Solution:

p(C|x) is called posterior probability.

p(x|C) is called likelihood.

p(C) is called prior probability.

p(x) is called evidence.

(c) Given the evidence/observed data below



what is the class label for a new animal with the following attribute values? Please show your calculation steps using Naïve Bayes Formulas. (3 points)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Give Birth | Can fly | Live in water | Have Legs | Class |
| Yes | No | yes | no | ? |

Solution:

X: attributes

M: Class: Mammal

N: Class: Non-Mammal

p(X|M) = (6/7) \* (6/7) \* (2/7) \* (2/7) = 0.06

p(X|N) = (1/13) \* (10/13) \* (3/13) \* (4/13) = 0.0042

p(X|M) \* p(M) = 0.06 \* (7/20) = 0.021

p(X|N) \* p(N) = 0.004\*(13/20) = 0.0027

p(X|M) \* p(M) > p(X|N) \* p(N)

Hence, the class label for a new animal with the following attribute values is **Mammal**.

1. 8 **points**
2. What is the relationship between LDA and Naïve Bayes? ( points)

Linear Discriminant Analysis (LDA) and naïve bayes both are used for linear classification. They both fall in the type of generative models and that is used to estimate the posterior probability as:

LDA and naïve bayes both accepts that Gaussian is with in class distribution. In naïve bayes, this is assumed that variables are independent while in LDA it is assumed that covariance for all the classes is same. If data is normally distributed then we can use naïve bayes and as well as LDA because it also uses normal distribution.

1. What is the relationship between LDA and Logistic regression? ( points)

Logistic regression aim is to find the conditional distribution i.e, while linear discriminant analysis aim is to find the joint distribution as . Logistic regression maximizes the conditional likelihood while LDA maximizes the joint likelihood.

Gross outliers are not handled by LDA while logistic regression can handle non Gaussian type of data as it takes into consider few assumption.

Often similar results are obtained by both of the models, logistic regression and LDA.

1. When do we use Logistic regression, LDA or Naïve Bayes? ( points)

**Logistic Regression:**

It is a regression analysis and it is used when dependent variable, the variable which we need to analyze based on given data is in the form of binary that is when one of the two outcomes are needed to be predicted. Logistic regression defines the relationship between a binary dependent variable with one or many independent variables.

**LDA:**

Linear discriminant analysis is preferred to use when number of classes are more than 2. It works on the assumption that the provided data is in Gaussian that is a bell curve is formed for each variable if plotted and other assumption is that each variable has the same variance.

It is also used for dimensionality reduction as a preprocessing task for machine learning. Aim of this to reduce the dataset into smaller form for better class prediction in order to avoid overfitting i.e., cure of dimensionality and this also minimizes the required computational power.

**Naïve Bayes:**

Naïve Bayes is practically useful when we have very large dataset. It can coverage fast when conditional assumption holds so it requires less training data and less time is consumed for training the model.

1. Please download IRIS data from UCI machine learning repository (<https://archive.ics.uci.edu/ml/datasets.html> ), compare the misclassification error rates, false positive rates and false negative rates between Naïve Bayes and LDA. Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as “Group#\_IRIS.R” (e.g., Group1\_IRIS.R) or “Group#\_IRIS.m” (e.g., “Group1\_IRIS.m”). ( points)

**R Code: Naïve Bayes**

|  |
| --- |
| data <- iris  indep = iris[,-5]  dep <- iris$Species  library("caret")  library("klaR")  library("e1071")  library("MASS")  nb = train(indep,dep,'nb',trControl=trainControl(method='cv',number=8))  nb  predict(nb$finalModel,indep)  (tab1 <- table(predict(nb$finalModel,indep)$class,dep))  (missclassification\_error <- 1 - sum(diag(tab1))/sum(tab1)) |

**Output:**

|  |
| --- |
| dep  setosa versicolor virginica  setosa 50 0 0  versicolor 0 47 3  virginica 0 3 47 |

**Misclassification Error for Naïve Bayes:**

|  |
| --- |
| * 0.04 🡺 4% |

**R Code: LDA**

|  |
| --- |
| library(MASS)  data <- iris  lda\_model <- lda(Species~., data)  summary(lda\_model)  p1 <- predict(lda\_model, data[,1:4])$class  (tab2 <- table(p1, data$Species))  (missclassification\_error <- 1 - sum(diag(tab2))/sum(tab2)) |

**Output:**

|  |
| --- |
| p1 setosa versicolor virginica  setosa 50 0 0  versicolor 0 48 1  virginica 0 2 49 |

**Misclassification Error for Naïve Bayes:**

|  |
| --- |
| * 0.02 🡺 2% |

**Description:**

When Naïve Bayes algorithm is applied on Iris dataset, we obtained misclassification error of 4% while LDA gave less misclassification error of 2%.

1. **4 points**

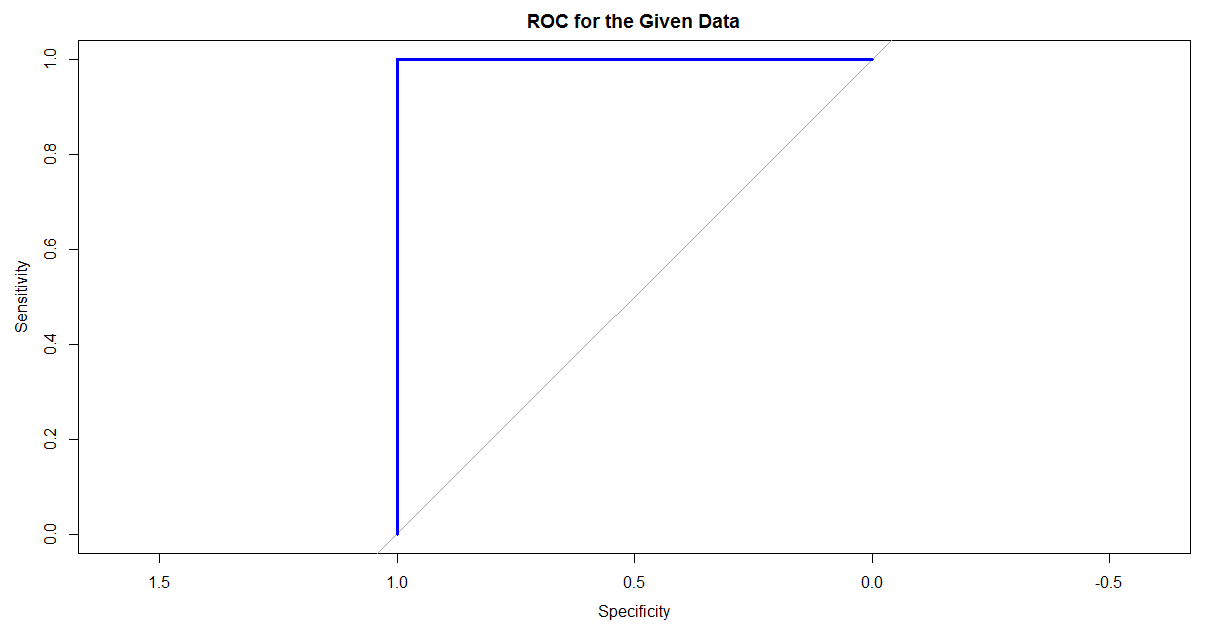
Generate ROC curve using the following tables. Please attach your R or Matlab code and output in your word file as Appendices. Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as “Group#\_ROC.R” (e.g., Group1\_ROC.R) or “Group#\_ROC.m” (e.g., “Group1\_ROC.m”).

|  |  |  |
| --- | --- | --- |
| **New measure** | **Disease** | **Non-Disease** |
| 5 or less | 18 | 1 |
| 5.1 - 7 | 7 | 17 |
| v7.1 - 9 | 4 | 36 |
| 9 or more | 3 | 39 |
| **Totals:** | 32 | 93 |
| **Cut point** | Sensitivity | Specificity |
| **5** | 0.56 | 0.99 |
| **7** | 0.78 | 0.81 |
| **9** | 0.91 | 0.42 |

R Code: ROC

|  |
| --- |
| cutPoints <- c(5,7,9)  truePositive <- c(0.56,0.78,0.91)  falsePositive <- c(0.1, 0.19, 0.58)  library(pROC)  plot(roc(cutPoints, truePositive, falsePositive), col="blue", lwd=3, main="ROC for the Given Data") |

**ROC Curve:**



# References:

<https://datascience.stackexchange.com/questions/19468/difference-between-lda-and-naive-bayes>

<https://www.quora.com/Classification-machine-learning-What-are-the-main-differences-between-the-LDA-Linear-Discriminant-Analysis-and-Naive-Bayes-classifiers>

<https://en.wikipedia.org/wiki/Linear_discriminant_analysis>

<https://en.wikipedia.org/wiki/Naive_Bayes_classifier>

<https://onlinecourses.science.psu.edu/stat857/book/export/html/17>

<https://www.quora.com/What-is-the-difference-between-logistic-regression-and-discriminant-analysis>

<http://www.statisticssolutions.com/what-is-logistic-regression/>

<https://en.wikipedia.org/wiki/Linear_discriminant_analysis>

<http://sebastianraschka.com/Articles/2014_python_lda.html>

<https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/>

<https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>

<https://discuss.analyticsvidhya.com/t/how-to-decide-when-to-use-naive-bayes-for-classification/5720/2>

<https://discuss.analyticsvidhya.com/t/how-to-decide-when-to-use-naive-bayes-for-classification/5720/2>

<http://www.aaai.org/Papers/FLAIRS/2004/Flairs04-097.pdf>

<https://www.quora.com/In-what-real-world-applications-is-Naive-Bayes-classifier-used>

<https://www.quora.com/When-and-why-is-a-naive-Bayes-classifier-a-better-worse-choice-than-a-random-forest-classifier>

<http://rischanlab.github.io/NaiveBayes.html>

<http://rstudio-pubs-static.s3.amazonaws.com/19670_4e38a32de389450a8d34f273efc50e08.html>