Homework 1

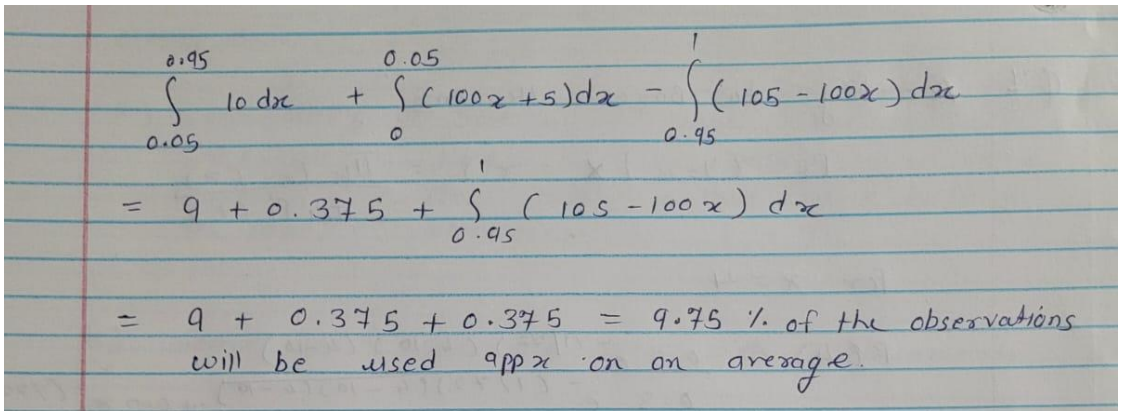
Vaishnavi Mule A20516627

1 Recitation Exercises

1.1 Chapter 4

Exercises: 4

a) The 10 percent range of x = 0.6 is (+ or –) 0.05, hence [0.6 – 0.05, 0.6 + 0.05] = [0.55, 0.65] In cases where x = 0.6, x E [0.05, 0.95] but if x < 0.05, then the range of possible values will be [0, x +0.05] with area of (100x + 5)% and if x > 0.95, it will be (105 – 100x)% so,



b) In the case where X1 and X2 are treated as independent, the % of observations used for making prediction will be (9.75 % * 9.75 %) = 0.95063 %

c) Using one argument among a and b we can tell that the percentage of observations in this case would be (9.75 %) ^ 100 == 0%

d) The percentage of observation used = (9.75 %) ^ p When p -> (infinite) , lim(p->(infinite)) (9.75) ^ p == 0

e) This exercise requires hypercube to have a value of 0.1. The volume of a p-dimensional hypercube with side of length l is:

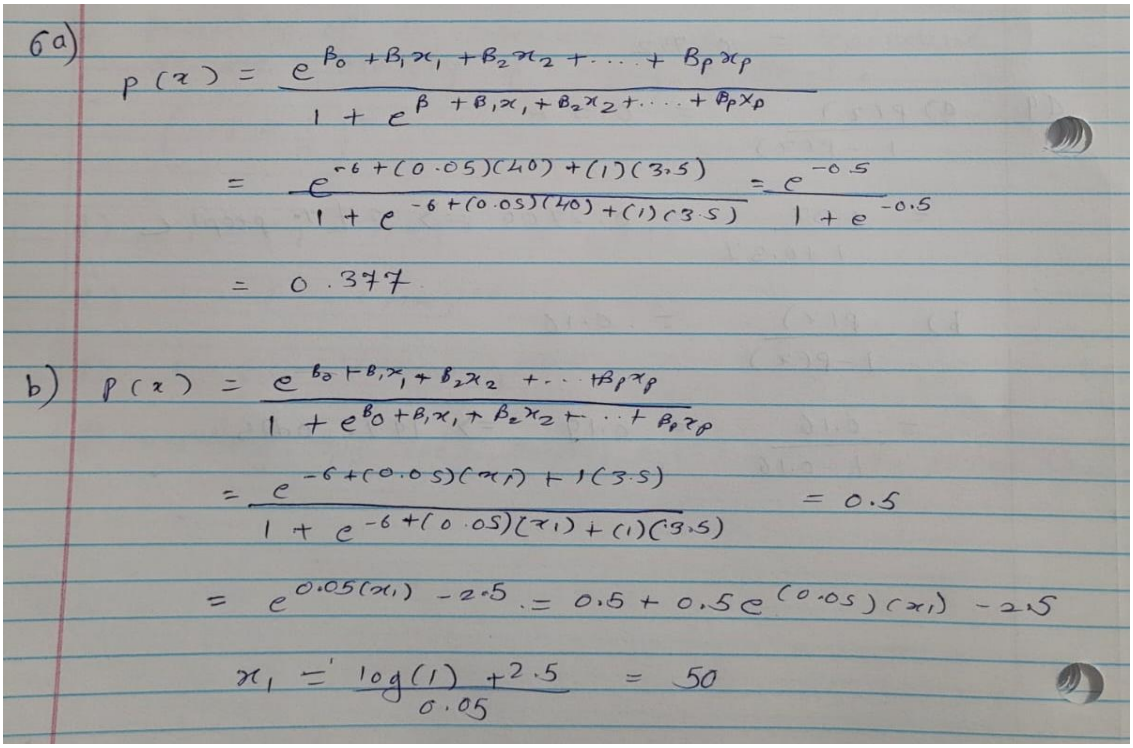
V=l^p ➔ l=v^(1/p)

For p=1, l=0.1

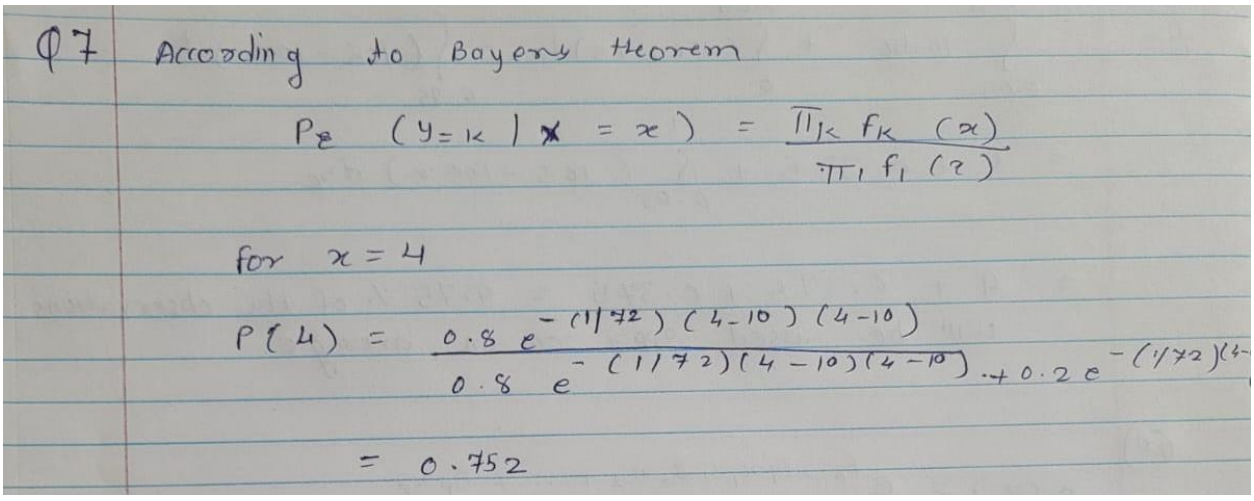
P=2, l=0.1^(1/2) ➔ 0.3162

P=3, l=0.1^(1/infinity) ➔ 0.9772

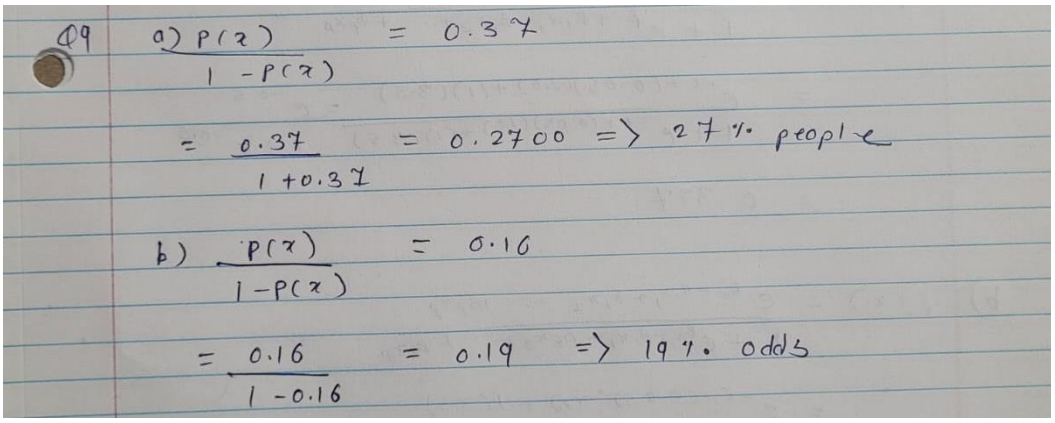
Exercises: 6



Exercises: 7

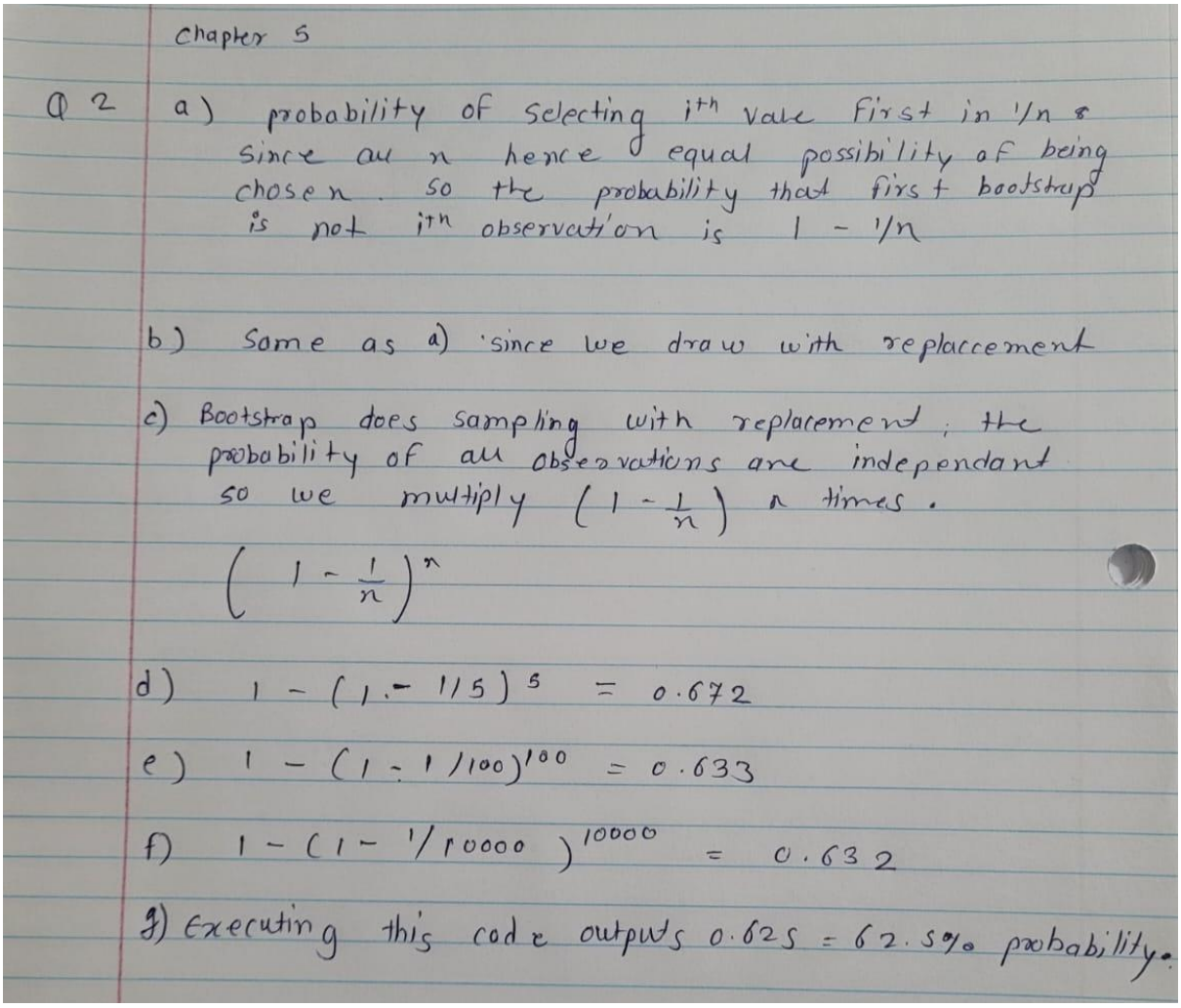


Exercises: 9


Q9
a) $\frac{P(x)}{1-P(x)} = 0.37$
$= \frac{0.37}{1+0.37} = 0.2700 \Rightarrow$ 27% people
b) $\frac{P(x)}{1-P(x)} = 0.16$
$= \frac{0.16}{1-0.16} = 0.19 \Rightarrow$ 19% odds

1.2 Chapter 5

Exercises: 2



Chapter 5

Q 2 a) probability of selecting i^{th} value first in $1/n$ since all n hence equal possibility of being chosen. so the probability that first bootstrap is not i^{th} observation is $1 - 1/n$

b) Same as a) since we draw with replacement

c) Bootstrap does sampling with replacement, the probability of all observations are independant so we multiply $(1 - \frac{1}{n})$ n times.

$$\left(1 - \frac{1}{n}\right)^n$$

d) $1 - (1 - 1/5)^5 = 0.672$

e) $1 - (1 - 1/100)^{100} = 0.633$

f) $1 - (1 - 1/10000)^{10000} = 0.632$

g) Executing this code outputs $0.625 = 62.5\%$ probability.

Exercises: 3

a) 1 : Divide dataset into k equal parts

2 : Put aside one part for test set and k-1 for training.

3 : record the MSE

The same process has to be repeated for all parts one after the other under the test sets and then average out the k MSE values.

b) 1 : Validation set approach

Disadvantage – It depends on which observation is being used in train and test set, accordingly validation estimate of test error can be variable.

Advantage – Validation set approach is easy to implement.

2 : LOOCV

Disadvantage – K fold has better test error than LOOCV as it bies variance trade-off. And k fold is less computationally demanding for common values of k.

Advantage – LOOCV requires less computing power in comparison to k fold in some cases. LOOCV doesn't have randomness.

2 Practicum Problems

2.1 Problem 1

Vaishnavi

2023-09-16

Load dataset

```
abalone_dataset = read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data")
names(abalone_dataset) = c("sex", "length", "diameter", "height", "weight_whole", "weight_shucked", "weight_viscera", "weight_shell", "rings")
```

Drop all rows where sex = Infant

```
abalone_dataset = subset(abalone_dataset, sex != "I")
abalone_dataset$sex = as.factor(ifelse(abalone_dataset$sex == "M", "1", "0"))
```

Split data to training set and testing set

```
library(caret)

## Loading required package: ggplot2
```

```
## Loading required package: lattice

split_80 = createDataPartition(abalone_dataset$sex, p=0.8, list=FALSE)

train<-abalone_dataset[split_80,]
test<-abalone_dataset[-split_80,]
```

Fitting logistic regression using glm function

```
logistic_r = glm(sex ~ ., train, family = binomial)
summary(logistic_r)

##
## Call:
## glm(formula = sex ~ ., family = binomial, data = train)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.0795222  0.5097727   6.041 1.53e-09 ***
## length         -3.0352099  2.2949415  -1.323 0.185980
## diameter       -4.3387127  2.7063497  -1.603 0.108899
## height         -2.2541575  1.7913015  -1.258 0.208250
## weight_whole   -0.0337472  0.8354623  -0.040 0.967779
## weight_shucked  3.3217692  1.0020535   3.315 0.000917 ***
## weight_viscera -2.0155018  1.4144695  -1.425 0.154181
## weight_shell    0.4049781  1.2835337   0.316 0.752368
## rings           0.0002262  0.0181632   0.012 0.990065
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3130.4  on 2267  degrees of freedom
## Residual deviance: 3061.6  on 2259  degrees of freedom
## AIC: 3079.6
##
## Number of Fisher Scoring iterations: 4

confint(logistic_r)

## Waiting for profiling to be done...

##                      2.5 %     97.5 %
## (Intercept)     2.09617515 4.09579510
## length         -7.53933446 1.46289064
## diameter       -9.65822350 0.95901006
```

```
## height          -6.39794450 0.95766292
## weight_whole    -1.68183788 1.60434533
## weight_shucked   1.36658677 5.30283055
## weight_viscera -4.79559259 0.75623327
## weight_shell    -2.11571066 2.92772678
## rings           -0.03539947 0.03585426
```

From above results we observe that the predictors are not doing a good job! as the coefficient values are close to 0 or very low. the value of rings is also close to 0.

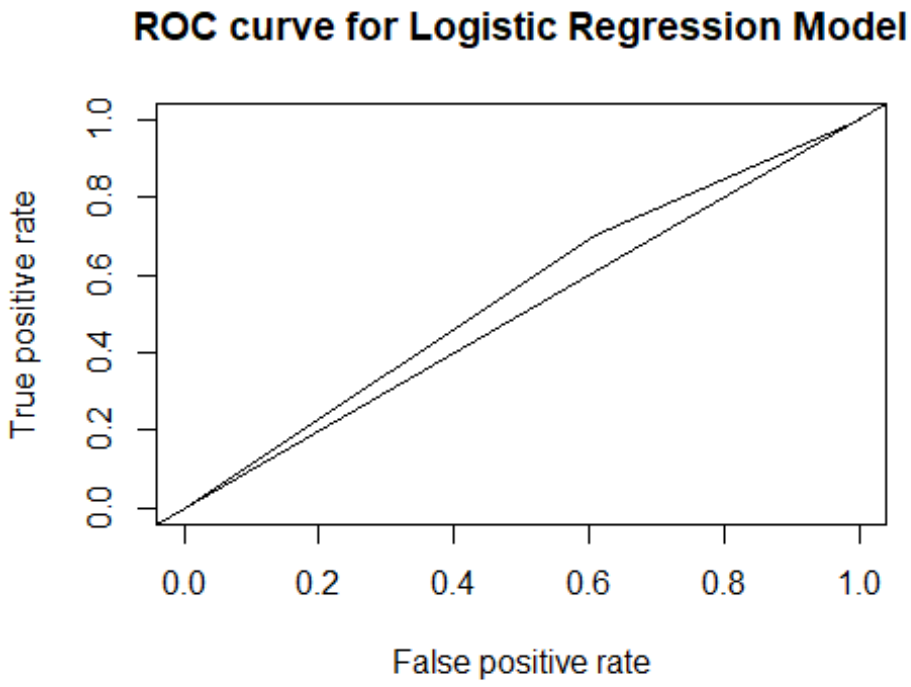
The CI for rings contains 0 so we cannot abandon the NULL hypothesis.

```
y_hat = predict(logistic_r, test, type = "response")
y_hat = ifelse(y_hat > 0.5, 1, 0)

confusionMatrix(table(as.factor(y_hat),as.factor(test$sex)))
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 102  90
##   1 159 215
##
##                Accuracy : 0.5601
##                  95% CI : (0.5181, 0.6014)
##     No Information Rate : 0.5389
##     P-Value [Acc > NIR] : 0.1661
##
##                   Kappa : 0.0976
##
##  Mcnemar's Test P-Value : 1.638e-05
##
##             Sensitivity : 0.3908
##             Specificity : 0.7049
##          Pos Pred Value : 0.5312
##          Neg Pred Value : 0.5749
##              Prevalence : 0.4611
##          Detection Rate : 0.1802
##    Detection Prevalence : 0.3392
##       Balanced Accuracy : 0.5479
##
##        'Positive' Class : 0
##
```

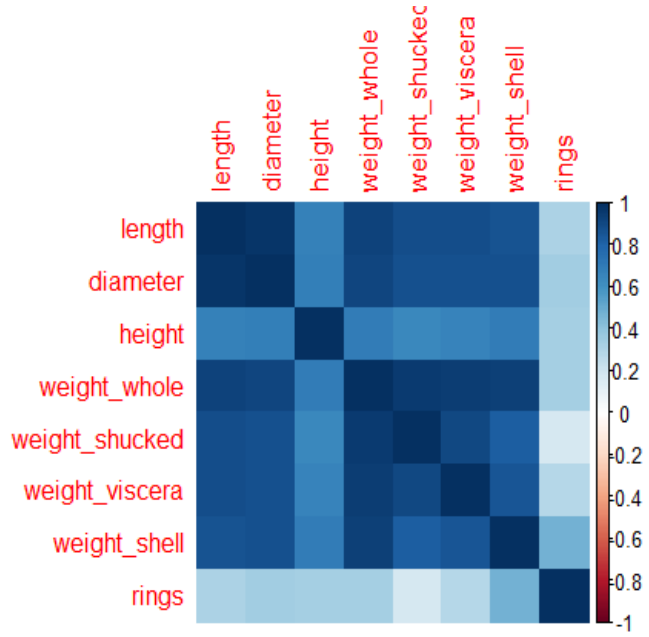
The accuracy of this model is 55.12%

```r
library(ROCR)
pred = prediction(y_hat, test$sex)
perf <- performance(pred, "tpr", "fpr")
plot(perf, main = "ROC curve for Logistic Regression Model")
abline(0, 1)
```



The graph has high rate of area under the curve

```r
library(corrplot)

## corrplot 0.92 loaded

corrplot(cor(train[, -1]), method="color")
```



The correlation between the attributes/predictors is pretty good from the above grid, hence the model's accuracy is most likely influenced by these attributes/predictors.

2.2 Problem 2

Vaishnavi

2023-09-16

```
mushroom_dataset = read.csv(file="https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.data")
names(mushroom_dataset) = c("class","cap_shape","cap_surface","cap_color","bruises","odor","gill_attachment","gill_spacing","gill_size","gill_color","stalk_shape","stalk_root","stalk_surface_above_ring","stalk_surface_below_ring","stalk_color_above_ring","stalk_color_below_ring","veil_type","veil_color","ring_number","ring_type","spore_print_color","population","habitat")

str(mushroom_dataset)

## 'data.frame':    8123 obs. of  23 variables:
##  $ class                   : chr  "e" "e" "p" "e" ...
##  $ cap_shape               : chr  "x" "b" "x" "x" ...
##  $ cap_surface             : chr  "s" "s" "y" "s" ...
##  $ cap_color               : chr  "y" "w" "w" "g" ...
##  $ bruises                 : chr  "t" "t" "t" "f" ...
```

```
##  $ odor                     : chr  "a" "l" "p" "n" ...
##  $ gill_attachment          : chr  "f" "f" "f" "f" ...
##  $ gill_spacing             : chr  "c" "c" "c" "w" ...
##  $ gill_size                : chr  "b" "b" "n" "b" ...
##  $ gill_color               : chr  "k" "n" "n" "k" ...
##  $ stalk_shape              : chr  "e" "e" "e" "t" ...
##  $ stalk_root               : chr  "c" "c" "e" "e" ...
##  $ stalk_surface_above_ring: chr  "s" "s" "s" "s" ...
##  $ stalk_surface_below_ring: chr  "s" "s" "s" "s" ...
##  $ stalk_color_above_ring   : chr  "w" "w" "w" "w" ...
##  $ stalk_color_below_ring   : chr  "w" "w" "w" "w" ...
##  $ veil_type                : chr  "p" "p" "p" "p" ...
##  $ veil_color               : chr  "w" "w" "w" "w" ...
##  $ ring_number              : chr  "o" "o" "o" "o" ...
##  $ ring_type                : chr  "p" "p" "p" "e" ...
##  $ spore_print_color        : chr  "n" "n" "k" "n" ...
##  $ population               : chr  "n" "n" "s" "a" ...
##  $ habitat                  : chr  "g" "m" "u" "g" ...
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.
0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.3     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
mushroom_dataset %>% gather(col_name, value, 1:23) %>% group_by(col_name)%>%t
ally(value == '?')
```

```
## # A tibble: 23 × 2
##    col_name            n
##    <chr>           <int>
##  1 bruises             0
##  2 cap_color           0
##  3 cap_shape           0
##  4 cap_surface         0
##  5 class               0
##  6 gill_attachment     0
##  7 gill_color          0
##  8 gill_size           0
##  9 gill_spacing        0
```

```
## 10 habitat             0
## # ℹ 13 more rows
```

The stalk_root column has 30% of the values as '?' hence we can drop those values and yet it wouldn't effect the entire data much!

```
mushroom_dataset = mushroom_dataset[mushroom_dataset$stalk_root !='?',]
head(mushroom_dataset)
```

```
##   class cap_shape cap_surface cap_color bruises odor gill_attachment
## 1     e         x           s         y       t    a               f
## 2     e         b           s         w       t    l               f
## 3     p         x           y         w       t    p               f
## 4     e         x           s         g       f    n               f
## 5     e         x           y         y       t    a               f
## 6     e         b           s         w       t    a               f
##   gill_spacing gill_size gill_color stalk_shape stalk_root
## 1            c         b          k           e          c
## 2            c         b          n           e          c
## 3            c         n          n           e          e
## 4            w         b          k           t          e
## 5            c         b          n           e          c
## 6            c         b          g           e          c
##   stalk_surface_above_ring stalk_surface_below_ring stalk_color_above_ring
## 1                        s                        s                       w
## 2                        s                        s                       w
## 3                        s                        s                       w
## 4                        s                        s                       w
## 5                        s                        s                       w
## 6                        s                        s                       w
##   stalk_color_below_ring veil_type veil_color ring_number ring_type
## 1                      w         p          w           o         p
## 2                      w         p          w           o         p
## 3                      w         p          w           o         p
## 4                      w         p          w           o         e
## 5                      w         p          w           o         p
## 6                      w         p          w           o         p
##   spore_print_color population habitat
## 1                 n          n       g
## 2                 n          n       m
## 3                 k          s       u
## 4                 n          a       g
## 5                 k          n       g
## 6                 k          n       m
```

```
library(e1071)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

split = createDataPartition(mushroom_dataset$class, p = 0.80, list=FALSE)
train_data = mushroom_dataset[split,]
test_data = mushroom_dataset[-split,]

nb_model= naiveBayes(train_data[,-1], train_data$class)
summary(nb_model)

##           Length Class  Mode
## apriori    2     table  numeric
## tables    22     -none- list
## levels     2     -none- character
## isnumeric 22     -none- logical
## call       3     -none- call

train_pred = predict(nb_model,train_data[,-1])
test_pred = predict(nb_model,test_data[,-1])

cat("Testing Model Accuracy: ",mean(test_pred == test_data$class)*100,"% \n")

## Testing Model Accuracy:  96.18794 %

cat("Training Model Accuracy: ",mean(train_pred == train_data$class)*100,"%")

## Training Model Accuracy:  95.37099 %

table(test_pred, test_data$class)

##
## test_pred   e   p
##         e 693  39
##         p   4 392
```

False positives are 44.

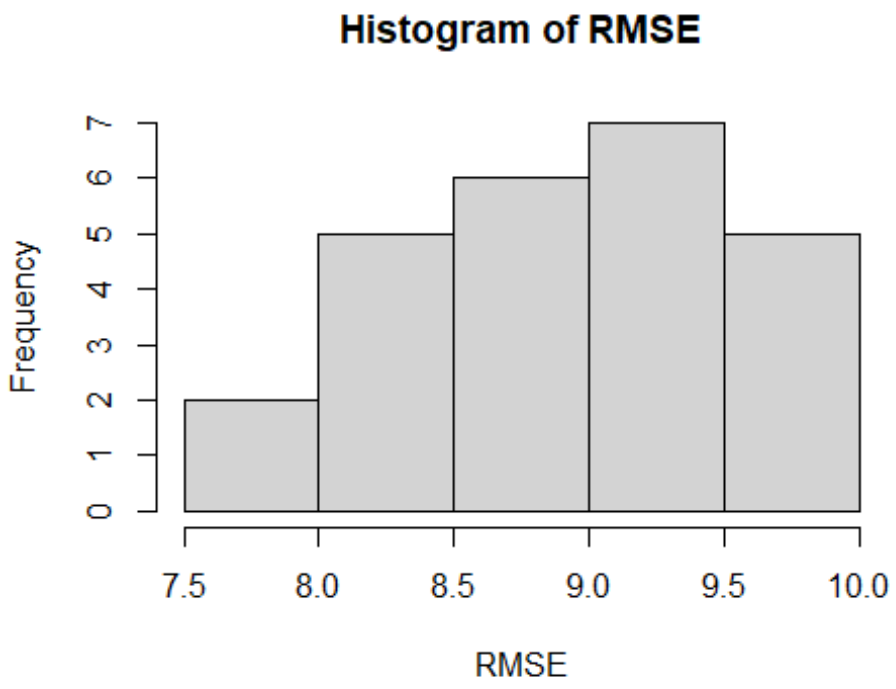
2.3 Problem 3

Vaishnavi

2023-09-16

```
library(data.table)
yacht_dataset = fread("https://archive.ics.uci.edu/ml/machine-learning-databases/00243/yacht_hydrodynamics.data")
names(yacht_dataset) = c("lcg","pr","LDR","BDR","LBR","frNo","Re")

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

split = createDataPartition(y = yacht_dataset$Re , p = 0.8, list = FALSE)
train_set = yacht_dataset[split,]
test_set = yacht_dataset[-split,]

linear_model = lm(Re ~., data = train_set)

summary(linear_model)

##
## Call:
## lm(formula = Re ~ ., data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.786  -7.707  -1.757   5.965  29.010
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -14.5897    31.1233  -0.469    0.640
## lcg           0.1014     0.3871   0.262    0.794
## pr          -23.4757    50.7851  -0.462    0.644
## LDR           3.2170    15.9406   0.202    0.840
## BDR          -0.5618     6.2915  -0.089    0.929
## LBR          -3.1007    15.8957  -0.195    0.846
## frNo        123.0411     5.6253  21.873   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.927 on 241 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.658
## F-statistic: 80.21 on 6 and 241 DF,  p-value: < 2.2e-16

cat("Train data of MSE = ", anova(linear_model)['Residuals', 'Mean Sq'])
```

```
## Train data of MSE =  79.69407

cat("\nTrain data of RMSE = ", sqrt(anova(linear_model)['Residuals', 'Mean Sq
'']))

##
## Train data of RMSE =  8.927153

cat("\nTrain data of R-squared = ",summary(linear_model)$r.sq)

##
## Train data of R-squared =  0.6663387

train_boot =  trainControl(method = "boot", number = 1000)
batch_lm = train(Re~., data = train_set, method = "lm" )
summary(batch_lm$resample$RMSE)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.728   8.495   8.999   8.953   9.423   9.863

summary(batch_lm$resample$Rsquared)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6023  0.6309  0.6527  0.6499  0.6683  0.6885

hist(batch_lm$resample$RMSE, xlab = "RMSE", main = "Histogram of RMSE")
```



2.4 Problem 4

Vaishnavi

2023-09-16

```
library(data.table)
df = fread("https://archive.ics.uci.edu/ml/machine-learning-databases/statlog
/german/german.data-numeric")

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

df$V25 = factor(df$V25)
split = createDataPartition(df$V25 , p = 0.8, list = FALSE)
train_set = df[split,]
test_set = df[-split,]

lg_r = glm(V25~., data = train_set, family=binomial)
summary(lg_r)

##
## Call:
## glm(formula = V25 ~ ., family = binomial, data = train_set)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.079362   1.357216   1.532  0.12550
## V1          -0.581885   0.080964  -7.187 6.63e-13 ***
## V2           0.031307   0.009842   3.181  0.00147 **
## V3          -0.393906   0.098336  -4.006 6.18e-05 ***
## V4           0.005450   0.004293   1.269  0.20428
## V5          -0.200822   0.066316  -3.028  0.00246 **
## V6          -0.091394   0.087092  -1.049  0.29399
## V7          -0.115249   0.127853  -0.901  0.36737
## V8           0.004030   0.093826   0.043  0.96574
## V9           0.210572   0.113565   1.854  0.06371 .
## V10         -0.013144   0.010005  -1.314  0.18892
## V11         -0.261394   0.126774  -2.062  0.03922 *
## V12          0.270469   0.185205   1.460  0.14419
## V13          0.182698   0.259551   0.704  0.48150
```

```
## V14         -0.138304   0.213378  -0.648  0.51688
## V15         -1.475536   0.713472  -2.068  0.03863 *
## V16          0.518374   0.216896   2.390  0.01685 *
## V17         -1.286618   0.397793  -3.234  0.00122 **
## V18          1.325513   0.511221   2.593  0.00952 **
## V19          1.560178   0.655234   2.381  0.01726 *
## V20          0.276446   0.403074   0.686  0.49281
## V21         -0.077600   0.355245  -0.218  0.82709
## V22         -0.294845   0.678292  -0.435  0.66379
## V23          0.042756   0.355148   0.120  0.90418
## V24         -0.029434   0.289484  -0.102  0.91901
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 977.38  on 799  degrees of freedom
## Residual deviance: 748.13  on 775  degrees of freedom
## AIC: 798.13
##
## Number of Fisher Scoring iterations: 5

fitted = ifelse(lg_r$fitted.values > 0.5,2,1)
fitted = factor(fitted)
cm = confusionMatrix(fitted, train_set$V25)

cat("Precision =", cm$byClass[5] * 100, "%\n")

## Precision = 81.15942 %

cat("Recall =", cm$byClass[6] * 100, "%\n")

## Recall = 90 %

cat("F1Score =", cm$byClass[7] * 100, "%\n")

## F1Score = 85.3514 %

train_control =  trainControl(method = "cv", number = 10)
lg_r2 = train(V25~., data = train_set, family = "binomial", tr = train_contro
l)

temp  = lg_r2$finalModel$predicted

cm2=confusionMatrix(temp, train_set$V25)

cat("Cross fold Precision = ", cm2$byClass[5] * 100, "%\n")

## Cross fold Precision =  79.17981 %

cat("cross fold Recall  ", cm2$byClass[6] * 100, "%\n")
```

```
## cross fold Recall   89.64286 %
```

```
cat("cross fold F1Score  ", cm2$byClass[7] * 100, "%\n")
```

```
## cross fold F1Score   84.0871 %
```

original data

```
gc_test_pred = predict(lg_r, test_set, type = "response")
gc_test_fitval = ifelse(gc_test_pred > 0.5,2,1)
gc_test_fitval = factor(gc_test_fitval)
```