module_3_assesment

Gafur Mammadov

2025-02-28

Question 1

```
titanic <- read.csv("~/Documents/math133/datasets/titanic.csv")
titanic <- na.omit(titanic)

titanic$Sex <- as.numeric(as.factor(titanic$Sex))

set.seed(44)
train_indices <- sample(1:nrow(titanic), size = 0.7 * nrow(titanic))

train_set <- titanic[train_indices,]
test_set <- titanic[-train_indices,]
```

k=5 results

```
titanic_knn_5 = knn(train_set[,c("Pclass","Sex", "Age", "Siblings.Spouses.Aboard", "Parents.Children.Aboard", "Fare")], test_set[,c("Pclass","Sex", "Age", "Siblings.Spouses.Aboard", "Parents.Children.Aboard", "Fare")], cl=train_set$Survived, k=5)
sum(titanic_knn_5!=test_set$Survived)/length(test_set$Survived)
```

```
## [1] 0.2921348
```

k=10 results

```
titanic_knn_10 = knn(train_set[,c("Pclass","Sex", "Age", "Siblings.Spouses.Aboard", "Parents.Children.Aboard", "Fare")], test_set[,c("Pclass","Sex", "Age", "Siblings.Spouses.Aboard", "Parents.Children.Aboard", "Fare")], cl=train_set$Survived, k=10)
sum(titanic_knn_10!=test_set$Survived)/length(test_set$Survived)
```

```
## [1] 0.2771536
```

k=15 results

```
titanic_knn_15 = knn(train_set[,c("Pclass","Sex", "Age", "Siblings.Spouses.Aboard", "Parents.Children.Aboard", "Fare")], test_set[,c("Pclass","Sex", "Age", "Siblings.Spouses.Aboard", "Parents.Children.Aboard", "Fare")], cl=train_set$Survived, k=15)
sum(titanic_knn_15!=test_set$Survived)/length(test_set$Survived)
```

```
## [1] 0.2808989
```

The lowest error rate was found with k=10.I did not understand what it means to compare with the null model

Question 2

```
titanic <- read.csv("~/Documents/math133/datasets/titanic.csv")
titanic <- na.omit(titanic)
if ("Name" %in% names(titanic)) {
  titanic <- titanic[, !names(titanic) %in% c("Name")]
}
set.seed(44)
train_indices <- sample(1:nrow(titanic), size = 0.7 * nrow(titanic))
train_set <- titanic[train_indices,]
test_set <- titanic[-train_indices,]

titanic_glm=glm(Survived~.,family="binomial",data=train_set)
summary(titanic_glm)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = "binomial", data = train_set)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              5.605064   0.686818   8.161 3.32e-16 ***
## Pclass                  -1.173409   0.177157  -6.624 3.51e-11 ***
## Sexmale                 -2.790851   0.239883 -11.634  < 2e-16 ***
## Age                     -0.052518   0.009527  -5.513 3.53e-08 ***
## Siblings.Spouses.Aboard -0.450126   0.137066  -3.284  0.00102 **
## Parents.Children.Aboard -0.072468   0.142804  -0.507  0.61183
## Fare                     0.003101   0.003057   1.014  0.31040
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 833.72  on 619  degrees of freedom
## Residual deviance: 544.95  on 613  degrees of freedom
## AIC: 558.95
##
## Number of Fisher Scoring iterations: 5
```

Results of Logistic Regression

```
test_set$prob = predict(titanic_glm,
                    newdata = test_set,
                    type = "response")
test_set$prediction = ifelse(test_set$prob > 0.5, 1, 0)
er_logistic=sum(test_set$prediction!=test_set$Survived)/length(test_set$Survived)
er_logistic
```

```
## [1] 0.1835206
```

The error rate of logistic regression lower than knn's ER. Which means that Logistic regression is performing better accuracy

Question 3

```
summary(titanic_glm)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = "binomial", data = train_set)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              5.605064   0.686818   8.161 3.32e-16 ***
## Pclass                  -1.173409   0.177157  -6.624 3.51e-11 ***
## Sexmale                 -2.790851   0.239883 -11.634  < 2e-16 ***
## Age                     -0.052518   0.009527  -5.513 3.53e-08 ***
## Siblings.Spouses.Aboard -0.450126   0.137066  -3.284  0.00102 **
## Parents.Children.Aboard -0.072468   0.142804  -0.507  0.61183
## Fare                     0.003101   0.003057   1.014  0.31040
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 833.72  on 619  degrees of freedom
## Residual deviance: 544.95  on 613  degrees of freedom
## AIC: 558.95
##
## Number of Fisher Scoring iterations: 5
```

1.What is the predicted odds ratio for males compared to females in terms of survival, assuming all other variables are held constant?

```
exp(coef(titanic_glm)[3])
```

```
##    Sexmale
## 0.06136895
```

The odds ratio for males compared to females in terms of survival is 0.061. Males have 0.061 times the odds of surviving compared to females, holding all other variables constant. Since the odds ratio is much less than 1, this means that males are significantly less likely to survive compared to females.

2 How does the odds ratio change if the Pclass is increased by one, while holding all other variables constant?

```
exp(coef(titanic_glm)[2])
```

```
##    Pclass
## 0.3093107
```

The odds ratio for an increase of one unit in Pclass is 0.309. In simple terms, moving from 1st class to 3rd class significantly reduces the likelihood of survival.

Question 3

```
conf_matrix_knn <- table(Predicted = titanic_knn_10, Actual = test_set$Survived)
conf_matrix_knn
```

```
##          Actual
## Predicted   0   1
##         0 143  45
##         1  29  50
```

```
conf_matrix_glm <- table(Predicted = test_set$prediction, Actual = test_set$Survived)
conf_matrix_glm
```

```
##          Actual
## Predicted   0   1
##         0 149  26
##         1  23  69
```

```
TP_knn <- conf_matrix_knn[2, 2]
TN_knn <- conf_matrix_knn[1, 1]
FP_knn <- conf_matrix_knn[2, 1]
FN_knn <- conf_matrix_knn[1, 2]

accuracy_knn <- (TP_knn + TN_knn) / sum(conf_matrix_knn)
precision_knn <- TP_knn / (TP_knn + FP_knn)
recall_knn <- TP_knn / (TP_knn + FN_knn)
specificity_knn <- TN_knn / (TN_knn + FP_knn)
balanced_accuracy_knn <- (recall_knn + specificity_knn) / 2

TP_glm <- conf_matrix_glm[2, 2]
TN_glm <- conf_matrix_glm[1, 1]
FP_glm <- conf_matrix_glm[2, 1]
FN_glm <- conf_matrix_glm[1, 2]

accuracy_glm <- (TP_glm + TN_glm) / sum(conf_matrix_glm)
precision_glm <- TP_glm / (TP_glm + FP_glm)
recall_glm <- TP_glm / (TP_glm + FN_glm)
specificity_glm <- TN_glm / (TN_glm + FP_glm)
balanced_accuracy_glm <- (recall_glm + specificity_glm) / 2

list(
  KNN = list(
    Confusion_Matrix = conf_matrix_knn,
    Accuracy = accuracy_knn,
    Precision = precision_knn,
    Recall = recall_knn,
    Specificity = specificity_knn,
    Balanced_Accuracy = balanced_accuracy_knn
  ),
  Logistic_Regression = list(
    Confusion_Matrix = conf_matrix_glm,
    Accuracy = accuracy_glm,
    Precision = precision_glm,
    Recall = recall_glm,
    Specificity = specificity_glm,
    Balanced_Accuracy = balanced_accuracy_glm
  )
)
```

```
## $KNN
## $KNN$Confusion_Matrix
##          Actual
## Predicted   0   1
##         0 143  45
##         1  29  50
##
## $KNN$Accuracy
## [1] 0.7228464
##
## $KNN$Precision
## [1] 0.6329114
##
## $KNN$Recall
## [1] 0.5263158
##
## $KNN$Specificity
## [1] 0.8313953
##
## $KNN$Balanced_Accuracy
## [1] 0.6788556
##
##
## $Logistic_Regression
## $Logistic_Regression$Confusion_Matrix
##          Actual
## Predicted   0   1
##         0 149  26
##         1  23  69
##
## $Logistic_Regression$Accuracy
## [1] 0.8164794
##
## $Logistic_Regression$Precision
## [1] 0.75
##
## $Logistic_Regression$Recall
## [1] 0.7263158
##
## $Logistic_Regression$Specificity
## [1] 0.8662791
##
## $Logistic_Regression$Balanced_Accuracy
## [1] 0.7962974
```