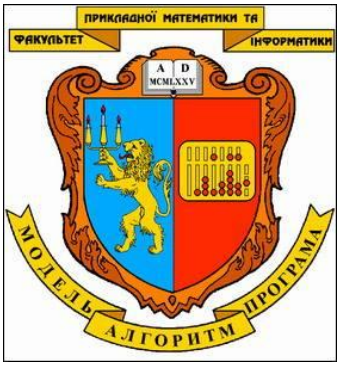
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Факультет прикладної математики та інформатики
Кафедра програмування



Індивідуальне завдання № 3

Класифікація

Виконала:
студентка групи ПМОм-11
Кравець Ольга

Львів 2025

Хід роботи

Варіант - 3

Визначаю значення змінної variant. Встановлюю set.seed(3) та генерую redundant як випадкове ціле число з рівномірного розподілу. Виконую ті самі дії для змінної year, але на інтервал (2006, 2008).

```
> variant=3
> variant
[1] 3
> set.seed(variant)
> redundant=floor(runif(1,5,25))
> redundant
[1] 8
> year=floor(runif(1,2006,2008))
> year
[1] 2007
```

Завдання 1.

Моделі, матриці помилок та загальні частки правильних прогнозів на тестових даних:

Логістична регресія:

```
> library(ISLR)
> set.seed(variant)
> train_data=subset(Weekly, Year >= 1990 & Year <= year)
> test_data=subset(Weekly, Year == 2010)
> log_m=glm(Direction ~ Lag2, data = train_data, family = 'binomial')
> prob=predict(log_m, test_data, type = 'response')
> pred = ifelse(prob > 0.5, 'Up', 'Down')
> log_conf_m = table(Predicted = pred, Actual = test_data$Direction)
> log_ac = sum(diag(log_conf_m)) / sum(log_conf_m)
> print(log_conf_m)
         Actual
Predicted Down Up
     Down    2  0
     Up     18 32
> print(log_ac)
[1] 0.6538462
```

Лінійний дискримінантний аналіз:

```
> library(MASS)
> lda_m = lda(Direction ~ Lag2, data = train_data)
> lda_prob = predict(lda_m, test_data)$class
> lda_conf_m = table(Predicted = lda_prob, Actual = test_data$Direction)
> lda_ac = sum(diag(lda_conf_m)) / sum(lda_conf_m)
>
> print(lda_conf_m)
         Actual
Predicted Down Up
     Down    2  0
     Up     18 32
> print(lda_ac)
[1] 0.6538462
```

Квадратичний дискримінантний аналіз:

```
> qda_m = qda(Direction ~ Lag2, data = train_data)
> qda_prob = predict(qda_m, test_data)$class
> qda_conf_m = table(Predicted = qda_prob, Actual = test_data$Direction)
> qda_ac = sum(diag(qda_conf_m)) / sum(qda_conf_m)
> print(qda_conf_m)
         Actual
Predicted Down Up
     Down    0  0
     Up     20 32
> print(qda_ac)
[1] 0.6153846
```

Метод K-найближчих сусідів з K = 1:

```
> library(class)
> lag_train=cbind(train_data$Lag2)
> lag_test=cbind(test_data$Lag2)
> set.seed(variant)
> knn_w=knn(lag_train, lag_test, train_data$Direction,1)
> table(knn_w,test_data$Direction)

knn_w  Down Up
  Down    8 16
  Up     12 16
> conf_m = table(knn_w, test_data$Direction)
> accuracy = sum(diag(conf_m)) / sum(conf_m)
> print(accuracy)
[1] 0.4615385
```

Порівняння отриманих результатів:

- Логістична регресія: 0.65.
- Лінійний дискримінантний аналіз: 0.65.
- Квадратичний дискримінантний аналіз: 0.61
- K-найближчих сусідів (K = 1): 0.46.

За якого K точність методу K-найближчих сусідів буде найбільшою?

```
> knn_accuracy = 0
> best_k = 1
> for (k in 1:10) {
+   knn_pred = knn(train = as.matrix(train_data$Lag2), test = as.matrix(test_data$Lag2), cl = train_data$Direction, k = k)
+   knn_conf_m = table(Predicted = knn_pred, Actual = test_data$Direction)
+   knn_acc = sum(diag(knn_conf_m)) / sum(knn_conf_m)
+   if (knn_acc > knn_accuracy) {
+     knn_accuracy = knn_acc
+     best_k = k
+ }}
> print(knn_accuracy)
[1] 0.6538462
> print(best_k)
[1] 4
```

Найкраща точність досягається при K=4 і вона складає 65.38%.

Завдання 2.

```
> library(ISLR)
> data("Auto")
> n_total=nrow(Auto)
> n_r=round((redundant/100)*n_total)
> r_ind=sample(1:n_total,n_r)
> Auto_mod=Auto[-r_ind,]
>
> mpg_mean=mean(Auto_mod$mpg)
> Auto_mod$mpg01=ifelse(Auto_mod$mpg>mpg_mean,1,0)
>
> set.seed(variant)
> test_size=round((2*redundant/100)*n_total)
> test_ind=sample(1:nrow(Auto_mod), test_size)
> test_data=Auto_mod[test_ind,]
> train_data=Auto_mod[-test_ind,]
>
> predictors=c('weight','displacement','horsepower')
>
> x_train=train_data[, predictors]
> y_train=train_data$mpg01
> x_test=test_data[, predictors]
> y_test=test_data$mpg01
```

Лінійний дискримінантний аналіз:

```
> library(MASS)
> library(class)
> library(stats)
> lda_m = lda(mpg01 ~ weight + displacement + horsepower, data = train_data)
> lda_pred = predict(lda_m, x_test)$class
> lda_conf_m = table(Predicted = lda_pred, Actual = y_test)
> lda_acc = sum(diag(lda_conf_m)) / sum(lda_conf_m)
> print(lda_conf_m)
         Actual
Predicted  0  1
        0 28  0
        1  8 27
> print(lda_acc)
[1] 0.8730159
```

Квадратичний дискримінантний аналіз:

```
> qda_m = qda(mpg01 ~ weight + displacement + horsepower, data = train_data)
> qda_pred = predict(qda_m, x_test)$class
> qda_conf_m = table(Predicted = qda_pred, Actual = y_test)
> qda_acc = sum(diag(qda_conf_m)) / sum(qda_conf_m)
> print(qda_conf_m)
         Actual
Predicted  0  1
        0 30  1
        1  6 26
> print(qda_acc)
[1] 0.8888889
```

Логістична регресія:

```
> log_m = glm(mpg01 ~ weight + displacement + horsepower, data = train_data, family = binomial)
> log_prob = predict(log_m, x_test, type = 'response')
> log_pred = ifelse(log_prob > 0.5, 1, 0)
> log_conf_m = table(Predicted = log_pred, Actual = y_test)
>
> log_acc = sum(diag(log_conf_m)) / sum(log_conf_m)
> print(log_conf_m)
         Actual
Predicted  0  1
        0 29  2
        1  7 25
> print(log_acc)
[1] 0.8571429
```

Метод K-найближчих сусідів з різними значеннями K:

```
> knn_accuracy = 0
> best_k = 1
> for (k in 1:10) {
+   knn_pred = knn(train = as.matrix(x_train), test = as.matrix(x_test), cl = y_train, k = k)
+   knn_conf_m = table(Predicted = knn_pred, Actual = y_test)
+   knn_acc = sum(diag(knn_conf_m)) / sum(knn_conf_m)
+   if (knn_acc > knn_accuracy) {
+     knn_accuracy = knn_acc
+     best_k = k
+ }}
> print(best_k)
[1] 6
> print(knn_accuracy)
[1] 0.8888889
```

Порівняння отриманих результатів:

- Лінійний дискримінантний аналіз: 0.87.
- Квадратичний дискримінантний аналіз: 0.88
- Логістична регресія: 0.85.
- K-найближчих сусідів: 0.88.

За якого K точність методу K-найближчих сусідів буде найбільшою?

Найкраща точність досягається при K=6 і вона складає 88.8%.