데이터마이닝 과제 3

2020136087 윤아현

[HW3] 분류 실습

유리 데이터셋에 포함된 다른 속성들을 기반으로 분류하기

- 과제는 RI 및 Percentages of Na, Mg, Al, Si, K, Ca, Ba, Fe를 기반하여 분류
- 214개의 데이터에서 임의로 훈련 집합과 테스트 집합을 설정
- 분류한 결과(데이터프레임)와 성능 평가까지 진행

해결방법

※ KNN 모델, Decision Tree 모델을 생성하여 성능을 측정하였습니다.

1번. KNN 모델

수행 방법

- 강의 자료와 동일하게, 200개와 14개로 나누어 train 및 test 개수를 설정해주었다.
- KNN에서 주변 몇 개의 points를 확인할지 설정할 때, 1개 / 3개 / 5개로 설정하여 성능을 측정하였다.
- test_data를 통해 예측된 값을 dataframe으로 나타내고, Confustion Matrix를 산출하였다.

결과

k 값에 관계없이 정확도가 일정하게 92.86%로 나타났다.

Code

```
// 1번. train, test data 분할하기
> n <- length(fgl$type)
> nt <- 200
> train <- sample(1:n, nt)
> train
> x <- fgl[, c(1:9)]
// 2번. knn 생성하기 (k = 1, 3, 5)
> nearest1 <- knn(train = x[train, ], test = x[-train, ], cl = fgl$type[train], k = 1)
> nearest3 <- knn(train = x[train, ], test = x[-train, ], cl = fgl$type[train], k = 3)
> nearest5 <- knn(train = x[train, ], test = x[-train, ], cl = fgl$type[train], k = 5)
// 3번. 정확도 산출하기
> pcornn1 = 100 * sum(fgl$type[-train]==nearest1)/(n-nt)
> pcornn1
```

[1] 92.85714

```
> pcornn3 = 100 * sum(fgl$type[-train]==nearest3)/(n-nt)
> pcornn3
[1] 92.85714
> pcornn5 = 100 * sum(fgl$type[-train]==nearest5)/(n-nt)
> pcornn5
[1] 92.85714
```

// 4번. dataframe으로 (실제 값, 예측 값) 나타내기 – k=3일 때를 중심으로 나타냄

```
> results_df <- data.frame(Actual = fgl$type[-train], Predicted = nearest3)
> results_df
```

```
> results_df
   Actual Predicted
1    WinF       WinF
2    WinF       WinF
3    WinF       WinF
4    WinF       WinF
5    WinF       WinF
6   WinNF      WinNF
7   WinNF       WinF
8   WinNF      WinNF
9   WinNF      WinNF
10    Veh        Veh
11   Tabl       Tabl
12   Head       Head
13   Head       Head
14   Head       Head
```

//5번. 혼동행렬 출력하기

```
> confusion_matrix2 <- confusionMatrix(nearest3, fgl$type[-train])
> confusion_matrix2
```

```
> confusion_matrix2
Confusion Matrix and Statistics

          Reference
Prediction WinF WinNF Veh Con Tabl Head
     WinF     5     1   0   0    0    0
     WinNF    0     3   0   0    0    0
     Veh      0     0   1   0    0    0
     Con      0     0   0   0    0    0
     Tabl     0     0   0   0    1    0
     Head     0     0   0   0    0    3

Overall Statistics

               Accuracy : 0.9286
                 95% CI : (0.6613, 0.9982)
    No Information Rate : 0.3571
    P-Value [Acc > NIR] : 1.439e-05

                  Kappa : 0.9021
```

2번. Decision Model 생성하기

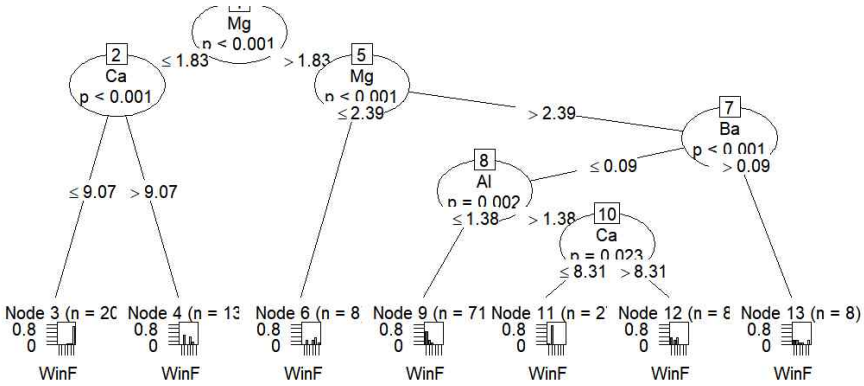
수행 방법

- train과 test 비율을 각각 70%, 30%로 설정해주었다.
- Decision tree 모델을 사용하여 train_data에 대한 학습을 수행하였다.
  이 경우, 종속 변수는 type, 독립 변수로는 그 외의 데이터로 설정하였다.
- test data를 통해 예측을 수행한다.
- 예측된 값을 dataframe으로 나타내고, 정확도 및 Confustion Matrix를 산출하였다.

결과

accuracy를 확인해본 결과, 모델의 정확도는 약 55.93%이다.
Confusion Matrix를 확인했을 때, WinF와 WinNF 클래스에서는 비교적 높은 정확도를 보이지만, 다른 클래스는 상대적으로 낮은 성능을 보인다.

아래는, 생성된 의사결정 트리를 시각화한 모습이다.



Code

```
// 1번. train, test 분할하기
ind <- sample(2, nrow(fgl), replace=TRUE, prob=c(0.7, 0.3))
> train_data <- fgl[ind==1,]
> test_data <- fgl[ind==2,]
// 2번. 의사결정나무 생성하기
> fgl_ctree <- ctree(type ~ RI + Na + Mg + Al + Si + K + Ca + Ba + Fe, data = train_data)
// 3번. test data 예측하기
> test_pred <- predict(fgl_ctree, newdata = test_data)
// 4번. dataframe으로 (실제 값, 예측 값) 나타내기
> result_df <- data.frame(
+       Actual = test_data$type,
+       Predicted = test_pred)
> result_df
```

```
> result_df
   Actual Predicted
1    WinF      WinF
2    WinF     WinNF
3    WinF      WinF
4    WinF     WinNF
5    WinF      WinF
6    WinF      WinF
7    WinF      WinF
8    WinF      WinF
9    WinF      WinF
10   WinF      WinF
11   WinF      WinF
12   WinF      WinF
13   WinF      WinF
14   WinF      WinF
15   WinF      WinF
16   WinF      WinF
17   WinF      WinF
18   WinF      WinF
19  WinNF      WinF
20  WinNF     WinNF
21  WinNF     WinNF
22  WinNF     WinNF
23  WinNF     WinNF
24  WinNF     WinNF
25  WinNF      WinF
26  WinNF      WinF
27  WinNF      WinF
28  WinNF     WinNF
29  WinNF     WinNF
30  WinNF     WinNF
31  WinNF      WinF
32  WinNF     WinNF
33  WinNF      WinF
34  WinNF      WinF
35  WinNF     WinNF
```

// 5번. 정확도 산출하기

```
> sum(test_pred==test_data$type)/length(test_pred) * 100
[1] 55.9322
```

// 6번. 혼동행렬 출력하기

```
> confusion_matrix <- confusionMatrix(test_pred, test_data$type)
> print(confusion_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction WinF WinNF Veh Con Tabl Head
     WinF    16     8   5   0    1    0
     WinNF    2    14   0   3    1    4
     Veh      0     0   0   0    0    0
     Con      0     0   0   0    0    0
     Tabl     0     0   0   0    0    0
     Head     0     0   0   1    1    3

Overall Statistics

               Accuracy : 0.5593
                 95% CI : (0.424, 0.6884)
    No Information Rate : 0.3729
    P-Value [Acc > NIR] : 0.00274

                  Kappa : 0.3549

 Mcnemar's Test P-Value : NA
```