Report

데이터마이닝 7.1장 7.3장 연습문제

32201805 박정민

DKU 단국대학교 DANKOOK UNIVERSITY

7.1

범주형 변수의 거리 계산 작은 데이터세트를 이용하는 이 연습문제에서는 유클리드 거리 계산과 이진 가변수를 만드는 과정을 나타낸다. 온라인 교육 회사인 Statistics.com은 고객과 잠재 고객을 IT 전문가(IT), 통계 전문가(Stat), 및 기타(Other)의 세 범주로 나눈다. 또한 각 고객에 대해 첫 번째 연락 이후 경과 일수(연수)를 추적한다. 다음은 고객의 코스 수강 여부(예상되는 결과)에 관한 정보이다:

고객 1: 통계, 1년, 코스 수강하지 않음
고객 2: 기타, 1.1년, 코스 수강함

a. 다음 새 잠재고객을 살펴보면: 잠재고객 1: IT, 1년

상단의 두 고객과 한 명의 잠재고객에 대한 정보를 바탕으로 범주형 예측변수를 2개의 이진변수로 변환하여 세 명에 대한 하나의 데이터세트를 만들고, 또한 범주형 예측변수를 3개의 이진변수로 변환하여 비슷한 데이터세트를 만드시오.

고객1, 고객2, 잠재고객에 대한 데이터세트를 만들어준다. 고객을 나타내는 변수를 expert, 수강일수를 나타내는 변수를 day, 수강여부를 나타내는 변수를 course라고 하여 데이터프레임 df를 만들어준다. 잠재고객의 수강여부는 모르는 값이므로 NA로 출력된다.

R코드

```
#데이터프레임 만들어줌
expert<-c("Stat","Other","IT") #고객
day<-c(1,1.1,1) #수강일수
course<-c(0,1,NA) #수강여부
df<-data.frame(expert, day, course)
df
```

출력결과

```
  expert day course
1   Stat 1.0      0
2  Other 1.1      1
3     IT 1.0     NA
```

2개의 이진변수와 3개의 이진변수로 변환해주기 위해 model.matrix() 함수를 이용하여 고객변수를 가변수로 변환해준다. 2개의 이진변수로 변환해줄 때 제거할 가변수가 무엇인지에 따라 유클리드 거리가 다르게 나오기 때문에 3개의 데이터세트를 모두 만들어주었다. df11은 IT 전문가 가변수를 제거해준 데이터프레임이고 df12는 Other(기타) 가변수를 제거해준 데이터프레임이다. df13는 Stat 전문가 가변수를 제거해준 데이터프레임이다. df2는 3개의 이진변수로 변환해준 데이터프레임이다.

R코드

```
#2개의 이진변수변환
df1 <-model.matrix(~ 0 + expert,data=df)
```

```
df1<-as.data.frame(df1)
t(t(names(df1)))

remove_IT<-df1[,-1]
remove_Other<-df1[,-2]
remove_Stat<-df1[,-3]

df11<-cbind(df[,-1],remove_IT)
df12<-cbind(df[,-1],remove_Other)
df13<-cbind(df[,-1],remove_Stat)

df11
df12
df13

#3개의 이진변수 변환
df2 <-model.matrix(~ 0 + expert,data=df)
df2<-as.data.frame(df2)
t(t(names(df2)))
df2<-cbind(df[-1],df2)
df2
```

출력되는 결과

```
> df11
  day course expertOther expertStat
1 1.0      0           0          1
2 1.1      1           1          0
3 1.0     NA           0          0
> df12
  day course expertIT expertStat
1 1.0      0        0          1
2 1.1      1        0          0
3 1.0     NA        1          0
> df13
  day course expertIT expertOther
1 1.0      0        0           0
2 1.1      1        0           1
3 1.0     NA        1           0
```

```
> df2
  day course expertIT expertOther expertStat
1 1.0      0        0           0          1
2 1.1      1        0           1          0
3 1.0     NA        1           0          0
```

b. 각각의 도출된 세트에서, 잠재고객과 각 고객의 유클리드 거리를 계산하시오.(주의: k-최근접이웃에서는 데이터를 표준화하는 것이 일반적이지만, 철칙은 아니므로 여기서는 정규화하지 않고 진행하도록 한다.)

각 데이터세트에서 3번째 행이 잠재고객을 나타내는 것은 공통적이므로 3번째 행의 수강일수와 1,2번째 행의 수강일수사이의 유클리드 거리를 비교해본다. R함수인 dist()함수를 통해 유클리도 거리를 계산하였다. k-최근접이웃을 사용할 때에는 단위가 맞지 않을 수 있어 표준화를 해주는 것이 일반적이지만 문제에서 하지 않고 진행하라하였다.

R코드

```
dist(df11[,-2], method = "euclidean")
dist(df12[,-2], method = "euclidean")
dist(df13[,-2], method = "euclidean")
dist(df2[,-2], method = "euclidean")
```

출력결과

```
> dist(df11[,-2], method = "euclidean")
         1        2
2 1.417745
3 1.000000 1.004988
> dist(df12[,-2], method = "euclidean")
         1        2
2 1.004988
3 1.414214 1.004988
> dist(df13[,-2], method = "euclidean")
         1        2
2 1.004988
3 1.000000 1.417745
> dist(df2[,-2], method = "euclidean")
         1        2
2 1.417745
3 1.414214 1.417745
```

위 R코드의 결과를 보면,

[IT 가변수 제거한 데이터세트]

-Stat 고객과 잠재고객 사이의 거리: 1

-Other 고객과 잠재고객 사이의 거리: 1.004988

∴ Stat 고객과 잠재고객 사이의 거리가 더 가깝다.

[Other 가변수 제거한 데이터세트]

-Stat 고객과 잠재고객 사이의 거리: 1.414214

-Other 고객과 잠재고객 사이의 거리: 1.004988

∴ Other 고객과 잠재고객 사이의 거리가 더 가깝다.

[Stat 가변수 제거한 데이터세트]

-Stat 고객과 잠재고객 사이의 거리: 1

-Other 고객과 잠재고객 사이의 거리: 1.417745

∴ Stat 고객과 잠재고객 사이의 거리가 더 가깝다.

[3개의 이진변수 변환 데이터세트]

-Stat 고객과 잠재고객 사이의 거리: 1.414214

-Other 고객과 잠재고객 사이의 거리: 1.417745

∴ Stat 고객과 잠재고객 사이의 거리가 더 가깝다.

따라서, Other 고객 가변수를 제거한 데이터세트(2개의 이진변수)와 다른 데이터세트들과의 유클리드 거리 비교의 차이가 있다.

C. k=1일 때 k-최근접이웃을 통해, 잠재고객을 두 개의 도출된 각각의 데이터세트를 이용하여 코스를 수강함 또는 코스를 수강하지 않음으로 분류하시오. 2개 또는 3개의 가변수를 사용하는 경우 차이가 있는가?

R코드

```
#2개의 이진변수 knn 알고리즘
library(FNN)
for (i in 3:5){
  knn_remove <- knn(train = df2[1:2,-c(i,2)],test=df2[3,-c(i,2)],cl=df2[1:2,2],k=1)
  print(colnames(df2[i]))
  print(knn_remove)
}
#3개의 이진변수 knn 알고리즘
knn(train=df2[1:2,-2],test=df2[3,-2],cl=df2[1:2,2],k=1)
```

출력결과

```
[1] "expertIT"
[1] 0
attr(,"nn.index")
     [,1]
[1,]    1
attr(,"nn.dist")
     [,1]
[1,]    1
Levels: 0
[1] "expertOther"
[1] 1
attr(,"nn.index")
     [,1]
[1,]    2
attr(,"nn.dist")
         [,1]
[1,] 1.004988
Levels: 1
[1] "expertStat"
[1] 0
attr(,"nn.index")
     [,1]
[1,]    1
attr(,"nn.dist")
     [,1]
[1,]    1
Levels: 0
```

```
[1] 0
attr(,"nn.index")
     [,1]
[1,]    1
attr(,"nn.dist")
         [,1]
[1,] 1.414214
Levels: 0
```

위 R코드에서 왼쪽에 보이는 출력결과가 2개의 이진변수로 변환한 3개의 데이터세트를 각각 FNN패키지에 있는 knn()함수를 통해 코스를 수강함 또는 코스를 수강하지 않음으로 분류하였다. 이 결과에서는 Other 가변수를 제거한 데이터세트에서 y=1(수강함)으로 예측하고 나머지 데이터세트에서는 y=0(수강하지 않음)으로 예측한다. 오른쪽 결과값은 3개의 이진변수로 변환한 데이터세트를 knn()함수를 통해 분류한 값이다. 이 데이터세트는 y=0(수강하지 않음)으로 예측하였다. 2개의 가변수를 사용한 경우와 3개의 가변수를 사용한 경우가 차이가 있다. 따라서, k-nn 알고리즘은 통계적 모델이 아닌 데이터 기반의 알고리즘으로 데이터에 관한 어떠한 가정도 하지 않는 비모수적 방법이다. 따라서 m개의 데이터가 있는 범주형 변수를 가변수할 때에 선형회귀모형에서와 같이 다중공선성 문제를 일으키지 않기 때문에 가변수 m개를 생성

하여야 한다.

7.3

보스턴 주택가격 예측 BostonHousing.csv 파일은 보스턴 지역의 500개의 인구 통계조사 주택단지 정보를 포함하고 있으며, 각 주택단지별로 여러 변수들에 대한 값들이 기록되어 있다. 마지막 열에 있는 CAT.MEDV는 주택가격의 중앙값(MEDV)의 파생된 변수로서, 주택가격의 중앙값이 30 이상이면(MEDV>30) 1을 갖고 그렇지 않으면 0을 갖는다. 처음 12개 열에 주어진 정보로 주택가격의 중앙값을 예측하고자 한다.

데이터를 학습 세트 60%와 검증 세트 40%로 분할하시오.

데이터세트 df로 파일을 불러온다음 전체데이터중 60%를 학습데이터, 나머지를 검증데이터로 분할하였다.

R코드

```
df<-read.csv("BostonHousing.csv")
str(df)
#데이터 분할
set.seed(123)
train.index <- sample(row.names(df), 0.6*dim(df)[1])
valid.index <- setdiff(row.names(df), train.index)

train.df <- df[train.index,]
valid.df <- df[valid.index,]
```

a. k값을 1에서부터 5까지 바꿔가면서, 12개의 예측변수(CAT.MEDV열은 제외)들을 사용하여 k-최근접이웃 예측을 수행하시오. 데이터를 정규화하였는지 확인하고 패키지 FNN보다는 패키지 class에 포함된 함수 knn()을 선택하시오. R이 class 패키지에 있는 함수 knn()을 사용하는지 확인하기 위해서는 class::knn()를 실행하시오. 최적의 k값은 얼마인가? 이것은 무엇을 의미하는가?

우선, 예측변수를 표준화시켜주기 위해 caret 패키지에 있는 preProcess()함수를 통해 학습데이터의 평균과 표준편차를 사용하여 표준화를 해준다. 그런 다음, 문제에서 k값을 1에서 5까지로 바꿔가면서 12개의 예측변수들을 사용하여 k-nn 예측을 수행하라 하였다. 분류 문제에서는 총 에러율을 사용하였지만 예측문제에서는 RMSE를 사용한다. 이 문제는 수치변수를 예측하는 문제이기 때문에 RMSE를 사용하여 최적의 k를 찾는다.

R코드

```
# 예측변수의 표준화
library(caret)
norm <- preProcess(train.df[,1:12], method = c('center', 'scale'))
```

```
train.n.df <- train.df
valid.n.df <- valid.df

train.n.df[,1:12] <- predict(norm, train.df[,1:12])
valid.n.df[,1:12] <- predict(norm, valid.df[,1:12])
#k-nn 알고리즘 수행
RMSE.df <- data.frame(k=seq(1:5), RMSE=rep(0,5))

for(i in 1:5){
  knn.pred <- class::knn(train.n.df[,1:12], valid.n.df[,1:12],
                         cl=train.n.df[,13], k=i)
  RMSE.df[i,2] <- caret::RMSE(as.numeric(as.character(knn.pred)), valid.df[,13])
}
RMSE.df
```

출력결과

	k	RMSE
1	1	4.941440
2	2	5.143047
3	3	6.191194
4	4	6.772547
5	5	6.961959

RMSE는 평균제곱오차의 제곱근으로 값이 작을수록 최적의 k가 되므로 k=1일 때 최적이고 k의 값이 커질수록 RMSE값이 커진다는 것을 알 수 있다.

b. 최적의 k값을 사용하여 다음 정보를 가진 주택단지의 중앙값(MEDV)을 예측하시오.

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	LSTAT
0.2	0	7	0	0.538	6	62	4.7	4	307	21	10

새로운 데이터를 데이터프레임 df2에 저장하여 학습데이터의 평균과 표준편차를 이용해 표준화를 시켜준다. 그리고 최적의 k인 1을 이용하여 다음 새로운 데이터의 주택단지의 중앙값을 k-nn 알고리즘을 통해 예측하였다.

R코드

```
#새 데이터프레임
df2<-data.frame(CRIM=0.2, ZN=0, INDUS=7, CHAS=0, NOX=0.538, RM=6, AGE=62,
                DIS=4.7, RAD=4, TAX=307, PTRATIO=21, LSTAT=10)
#표준화
df2.n<-df2
```

```
df2.n <- predict(norm, df2)
#예측
knn <- class::knn(train.n.df[,1:12], df2.n,
                  cl=train.n.df[,13], k=1)
knn
```

출력결과

```
> knn
[1] 20.4
180 Levels: 5 5.6 6.3 7 7.2 7.5 8.1 8.3 8.4 8.5 8.8 9.5 9.7 10.2 10.5 10.8 11.3 11.7 ... 50
```

위 R코드의 값은 20.4가 나온다. 따라서, 새로운 데이터의 주택단지의 중앙값(MEDV)는 20.4라고 예측할 수 있다.

C. 만약 k-최근접이웃 알고리즘을 사용하여 학습 데이터들의 점수를 낸다면 학습 세트에 대한 오류는 얼마가 되겠는가?

R코드

```
RMSE1.df <- data.frame(k=seq(1,20), RMSE=rep(0,20))
for(i in 1:20){
  knn.pred <- class::knn(train.n.df[,1:12], train.n.df[,1:12],
                         cl=train.n.df[,13], k=i)
  RMSE1.df[i,2] <- caret::RMSE(as.numeric(as.character(knn.pred)), train.df[,13])
}
RMSE1.df
```

출력 결과

	k	RMSE
1	1	0.000000
2	2	2.730382
3	3	4.584581
4	4	5.099388
5	5	5.337992
6	6	5.297431
7	7	5.609248
8	8	5.692720
9	9	5.356594
10	10	5.998025
11	11	6.290918
12	12	6.419694
13	13	6.015810
14	14	6.258054
15	15	6.557753
16	16	6.414245
17	17	6.329779
18	18	6.279181
19	19	6.278945
20	20	6.873844

학습 데이터들의 점수를 내기 위해서 knn함수의 test 데이터에도 train 데이터를 넣어 예측하였다. 출력 결과 k=1일 때 RMSE=0이 나온다. 이 결과를 통해 학습데이터에 대한 오류는

k=1일 때 자기자신이기 때문에 RMSE=0이 나올 수밖에 없음을 알 수 있다.

d. 검증 세트 오류는, 새로운 데이터에 k-최근접이웃을 적용할 때의 오류율에 비해 지나치게 낙관적이다. 왜 그러한가?

검증세트는 여러 모델에 반복적으로 적용하여 성능이 뛰어난 모델을 선정하므로 최종적으로 한번만 적용하는 새로운 데이터에 k-nn을 적용할 때의 오류율에 비해 낮게 나온다. 모델의 성능을 평가할 때에 사용한 데이터가 검증데이터이기 때문에 최종적인 모델을 위해 사용하는 새로운 데이터에 비해서 지나치게 낙관적이다.

e. 만약에 몇천 개의 새로운 주택단지에 대한 주택가격 중앙값(MEDV)을 예측하는 것이 목적이라면, k-최근접이웃 예측을 사용하는 것의 단점은 무엇인가? 각 예측을 위하여 알고리즘이 거치는 수행 절차에 대한 목록을 작성하시오.

k-최근접이웃 예측은 학습 세트로부터 모수를 추정하는 데 걸리는 시간은 거의 없다고 할지라도, 학습 세트가 클 경우에는 근접이웃들을 찾는 데 걸리는 시간이 엄청나게 소요된다. 따라서, 몇천 개의 새로운 주택단지에 대한 주택가격 중앙값을 예측하기 위해 학습 세트가 지나치게 크기도 하고 새로운 데이터가 들어올 경우 처음부터 다시 예측해야하기 때문에 소요되는 시간이 크다는 것이 단점이다. 각 예측을 위해서는 첫 번째 단계로 거리를 계산하여 이웃을 결정하고 클래스를 결정하기 위해 k-최근접이웃들의 평균 결과값을 사용한다. 최적의 k를 구할 때에는 RMSE를 사용하여 수행한다. 이 단계를 새로운 데이터가 들어올때마다 수행한다.