Business Analytics

1장, 2장. 데이터 마이닝 프로세스 개요 - West Roxbury 데이터셋을 활용

```
# Loading data
housing.df <- read.csv('https://raw.githubusercontent.com/reisanar/datasets/master/WestRoxbury.csv')
dim(housing.df)
```

```
## [1] 5802   14
```

```
# Random sample of 5 observations
housing.df[sample(row.names(housing.df), 5), ]
```

```
##      TOTAL.VALUE  TAX LOT.SQFT YR.BUILT GROSS.AREA LIVING.AREA FLOORS ROOMS
## 925        290.0 3648     5304     1960       2576        1288    1.0     5
## 3557       371.9 4678     7515     1960       2388        1265    1.5     6
## 2703       468.6 5894    10542     1900       2820        1719    2.0     8
## 5527       589.6 7417     5701     1947       3968        2496    2.0     9
## 729        266.6 3353     2835     1984       2464        1596    2.0     6
##      BEDROOMS FULL.BATH HALF.BATH KITCHEN FIREPLACE REMODEL
## 925         3         1         1       1         1    None
## 3557        3         2         0       1         1    None
## 2703        4         1         1       1         1    None
## 5527        4         2         1       1         2  Recent
## 729         3         1         1       1         1    None
```

```
# Oversample houses with over 10 rooms
s <- sample(row.names(housing.df), 5, prob = ifelse(housing.df$ROOMS > 10, 0.9, 0.01))
housing.df[s, ]
```

```
##      TOTAL.VALUE  TAX LOT.SQFT YR.BUILT GROSS.AREA LIVING.AREA FLOORS ROOMS
## 4310       363.8 4576     3917     1928       2491        1268    1.5     6
## 2789       505.3 6356    10000     1890       4458        2762    2.0    12
## 5688       379.8 4777     5000     1930       2050        1274    2.0     6
## 3458       689.3 8671     7000     1848       6235        3446    2.0    12
## 4860       463.2 5827     5846     1935       2652        1752    2.0     7
##      BEDROOMS FULL.BATH HALF.BATH KITCHEN FIREPLACE REMODEL
## 4310        3         1         1       1         1    None
## 2789        6         2         1       1         0  Recent
## 5688        3         1         1       1         1    None
## 3458        5         3         0       1         2    None
## 4860        3         1         0       1         1    None
```

분류하고자 하는 데이터가 희귀할 경우에는 이를 처리할 수 있도록 해줘야 한다.

이를 하는 방법은 소수 사건에 가중치를 부여하거나, 오분류에 큰 가중치를 주는 방법이 있다.

```r
# Categorical variables -> Dummy variables
xtotal <- model.matrix(~ 0 + BEDROOMS + REMODEL, data = housing.df)
xtotal <- as.data.frame(xtotal)
t(t(names(xtotal)))
```

```
##      [,1]
## [1,] "BEDROOMS"
## [2,] "REMODELNone"
## [3,] "REMODELOld"
## [4,] "REMODELRecent"
```

```r
xtotal <- xtotal[, -4]
head(xtotal, 5)
```

```
##   BEDROOMS REMODELNone REMODELOld
## 1        3           1          0
## 2        4           0          0
## 3        4           1          0
## 4        5           1          0
## 5        3           1          0
```

범주의 순서가 없는 Categorical variables의 값들은 가변수를 만들어주는 Dummy coding이 가능하다.

```r
# Median 값으로 결측치 대체하기
rows.to.missing <- sample(row.names(housing.df), 10)
housing.df[rows.to.missing, ]$BEDROOMS <- NA
summary(housing.df$BEDROOMS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    3.00    3.00    3.23    4.00    9.00      10
```

```r
housing.df[rows.to.missing, ]$BEDROOMS <- median(housing.df$BEDROOMS, na.rm = TRUE)
summary(housing.df$BEDROOMS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   3.229   4.000   9.000
```

```r
set.seed(1)

# Train, valid dataset split
train.rows <- sample(rownames(housing.df), dim(housing.df)[1] * 0.6)
train.data <- housing.df[train.rows, ]
valid.rows <- setdiff(rownames(housing.df), train.rows)
valid.data <- housing.df[valid.rows, ]

# Train, valid, test = 50%, 30%, 20%
train.rows <- sample(rownames(housing.df), dim(housing.df)[1] * 0.5)
valid.rows <- sample(setdiff(rownames(housing.df), train.rows),
                     dim(housing.df)[1] * 0.3)
test.rows <- setdiff(rownames(housing.df), union(train.rows, valid.rows))
```

```
train.data <- housing.df[train.rows, ]
valid.data <- housing.df[valid.rows, ]
test.data <- housing.df[test.rows, ]

dim(train.data)
```

```
## [1] 2901   14
```

```
dim(valid.data)
```

```
## [1] 1740   14
```

```
dim(test.data)
```

```
## [1] 1161   14
```

모델을 구축하기 위해서 데이터셋을 Train, Valid, Test로 나눠주도록 한다.

일반적으로 모델링 과정은 다음과 같은 순서를 갖는다.

데이터 마이닝 모델링 process

1. [목적] 웨스트 록스베리 지역 주택 가격을 예측한다.
2. [데이터 획득] 주택 가격 데이터를 활용한다.
3. [데이터 탐색, 정제, 전처리] 변수에 대한 충분한 이해를 한다.
    - 어떤 변수를 사용할 것인가
    - 이상치 유무를 확인
    - 범주형 변수를 가변수로 변환 (Dummy coding)
4. [차원 축소] 많은 변수를 갖는 경우에는, 차원 축소 방법을 사용한다.
5. [데이터 마이닝 태스크 결정] 주택 가격 예측을 위한 지도학습 방법을 사용한다.
6. [지도학습을 위해 데이터 분할] Train, Valid, Test 데이터로 나눈다.
7. [데이터 마이닝 기법 선택] 이번 분석에서는 다중회귀분석을 사용한다.
8. [태스크를 위한 알고리즘 사용] 모델 평가 지표를 만든다.
9. [결과 해석] 다양한 방법을 통해 나온 결과들 중 가장 좋은 모델을 선택한다.
10. [모델 사용] 최상의 모델에 대해서 목표한 바에 대한 결과를 얻는다.

```
# Modeling
reg <- lm(TOTAL.VALUE ~ ., data = housing.df, subset = train.rows)
tr.res <- data.frame(train.data$TOTAL.VALUE, reg$fitted.values, reg$residuals)
head(tr.res)
```

```
##      train.data.TOTAL.VALUE reg.fitted.values reg.residuals
## 1886                  356.7          356.7175   -0.01745989
## 3515                  333.3          333.2673    0.03267493
## 460                   298.6          298.6111   -0.01111445
## 855                   265.3          265.3050   -0.00503505
## 4094                  575.1          575.0797    0.02026379
## 3581                  348.0          347.9750    0.02502157
```

```r
# 만들어진 모델을 Valid 데이터에 적용하기
pred <- predict(reg, newdata = valid.data)
vl.res <- data.frame(valid.data$TOTAL.VALUE, pred,
                     residuals = valid.data$TOTAL.VALUE - pred)
head(vl.res)
```

```
##      valid.data.TOTAL.VALUE     pred     residuals
## 5642                  318.0 318.0047 -0.0046799114
## 2766                  498.7 498.6882  0.0118467844
## 3676                  331.8 331.8371 -0.0370594670
## 2054                  371.9 371.8998  0.0002217257
## 2217                  436.2 436.2079 -0.0078569690
## 3117                  280.2 280.1660  0.0340387189
```

```r
# 모델의 평가 측도를 계산
accuracy(reg$fitted.values, train.data$TOTAL.VALUE) # train data
```

```
##                      ME       RMSE        MAE          MPE        MAPE
## Test set -3.42853e-17 0.02273201 0.01969155 2.691985e-06 0.005303514
```

```r
pred <- predict(reg, newdata = valid.data)
accuracy(pred, valid.data$TOTAL.VALUE)
```

```
##                       ME       RMSE        MAE           MPE        MAPE
## Test set -0.0001632785 0.02300847 0.01984785 -9.496239e-05 0.005331516
```

5장. 예측 성능 평가

지도학습의 출력 변수

1. 예측된 수치값: 출력 변수가 주택 가격과 같은 수치값
2. 예측된 클래스 소속도: 출력 변수가 범주값
    - 경향의 컷오프(Threshold) 값을 사용하여 클래스 소속도 생성
3. 경향: 출력 변수가 범주값 일 때의 클래스 소속도의 확률