Prediction of Median House Prices For California Districts

Rümeysa KURT

3/20/23

Calling the dataset

The dataset in this report is from Kaggle.

```r
library(readr)
housing <- read_csv("housing.csv")
```

```r
housing_new <- na.exclude(housing)
```

Since the data contains NA, na.exclude() was used to exclude these observations.

1.Detail your task with the problem, features, and target.

Problem

In this report predicted median house prices for California districts.

Features

1. Longitude: A measure of how far west a house is; a higher value is farther west.
2. Latitude: A measure of how far north a house is; a higher value is farther north.
3. Housing Median Age: Median age of a house within a block; a lower number is a newer building.
4. Total Rooms: Total number of rooms within a block.

5. Total Bedrooms: Total number of bedrooms within a block.
6. Population: Total number of people residing within a block.
7. Households: Total number of households, a group of people residing within a home unit, for a block.
8. Median Income: Median income for households within a block of houses (measured in tens of thousands of US Dollars)
9. Ocean Proximity: Location of the house w.r.t ocean/sea.

Target

Median House Value: Median house value for households within a block (measured in US Dollars)

2.Describe the dataset in your task in terms of the dimension, variable type, and some other that you want to add.

```
install.packages("DALEX")
install.packages("ggplot2")
library(DALEX)
library(ggplot2)

str(housing_new)
```

```
tibble [20,433 x 10] (S3: tbl_df/tbl/data.frame)
 $ longitude         : num [1:20433] -122 -122 -122 -122 -122 ...
 $ latitude          : num [1:20433] 37.9 37.9 37.9 37.9 37.9 ...
 $ housing_median_age: num [1:20433] 41 21 52 52 52 52 52 52 42 52 ...
 $ total_rooms       : num [1:20433] 880 7099 1467 1274 1627 ...
 $ total_bedrooms    : num [1:20433] 129 1106 190 235 280 ...
 $ population        : num [1:20433] 322 2401 496 558 565 ...
 $ households        : num [1:20433] 126 1138 177 219 259 ...
 $ median_income     : num [1:20433] 8.33 8.3 7.26 5.64 3.85 ...
 $ median_house_value: num [1:20433] 452600 358500 352100 341300 342200 ...
 $ ocean_proximity   : chr [1:20433] "NEAR BAY" "NEAR BAY" "NEAR BAY" "NEAR BAY" ...
 - attr(*, "na.action")= 'exclude' Named int [1:207] 291 342 539 564 697 739 1098 1351 1457 
  ..- attr(*, "names")= chr [1:207] "291" "342" "539" "564" ...
```

The `str()` function is used to browse the dataset. In this way, the number of observations in the dataset, the number of features, the type of features, and the dimension of the dataset were reached.

Longitude, latitude, housing median age, total rooms, total bedrooms, population, households, median income, median house value features are numerical, ocean proximity feature is categorical.

The size of the dataset is [20.433 x 10].

3.Train a linear regression model.

```
set.seed(123) # for reproducibility
index <- sample(1 : nrow(housing_new), round(nrow(housing_new) * 0.80))
train <- housing_new[index, ]
test  <- housing_new[-index, ]
```

The `sample()` function is used to split the dataset into `test` and `train` sets.

Using the `index` object, the observations are set to `train` and `test`.

```
dim(test)
```

```
[1] 4087   10
```

```
dim(train)
```

```
[1] 16346    10
```

The `dim()` function is used to see how many rows and how many columns are in data sets separated into `test` and `train`.

While there are 4087 observations and 10 features in the test part, there are 16346 observations and 10 features in the train part.

```
model <- lm(median_house_value ~ longitude + latitude + housing_median_age + total_rooms +
              total_bedrooms + population + households + median_income +
              factor(ocean_proximity) , data = train)
```

There is a categorical property in the inputs, we specified it as factor(ocean_proximity) in the R function.

```r
model
```

```

Call:
lm(formula = median_house_value ~ longitude + latitude + housing_median_age +
    total_rooms + total_bedrooms + population + households +
    median_income + factor(ocean_proximity), data = train)

Coefficients:
                      (Intercept)                          longitude
                       -2.292e+06                         -2.708e+04
                         latitude                 housing_median_age
                       -2.576e+04                          1.057e+03
                      total_rooms                     total_bedrooms
                       -6.195e+00                          9.151e+01
                       population                         households
                       -3.734e+01                          5.891e+01
                    median_income      factor(ocean_proximity)INLAND
                        3.923e+04                         -3.781e+04
    factor(ocean_proximity)ISLAND    factor(ocean_proximity)NEAR BAY
                        1.709e+05                         -3.503e+03
factor(ocean_proximity)NEAR OCEAN
                        4.828e+03
```

```r
summary(model)
```

```

Call:
lm(formula = median_house_value ~ longitude + latitude + housing_median_age +
    total_rooms + total_bedrooms + population + households +
    median_income + factor(ocean_proximity), data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-555127  -42863  -10715   28683  756919

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                       -2.292e+06  9.858e+04 -23.254  < 2e-16 ***
longitude                         -2.708e+04  1.143e+03 -23.696  < 2e-16 ***
```

```
latitude                          -2.576e+04  1.127e+03 -22.849  < 2e-16 ***
housing_median_age                 1.057e+03  4.940e+01  21.407  < 2e-16 ***
total_rooms                       -6.195e+00  8.724e-01  -7.102 1.28e-12 ***
total_bedrooms                     9.151e+01  7.551e+00  12.119  < 2e-16 ***
population                        -3.734e+01  1.181e+00 -31.606  < 2e-16 ***
households                         5.891e+01  8.214e+00   7.172 7.71e-13 ***
median_income                      3.923e+04  3.761e+02 104.308  < 2e-16 ***
factor(ocean_proximity)INLAND     -3.781e+04  1.955e+03 -19.339  < 2e-16 ***
factor(ocean_proximity)ISLAND      1.709e+05  3.450e+04   4.953 7.38e-07 ***
factor(ocean_proximity)NEAR BAY   -3.503e+03  2.141e+03  -1.636    0.102
factor(ocean_proximity)NEAR OCEAN  4.828e+03  1.757e+03   2.748    0.006 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 68910 on 16333 degrees of freedom
Multiple R-squared:  0.6459,    Adjusted R-squared:  0.6457
F-statistic:  2483 on 12 and 16333 DF,  p-value: < 2.2e-16
```

Since our p-value is less than 0.05, 0 is rejected and the model is said to be significant.

It returns some statistics about the residuals, the model parameters, and also some performance metric values such as the multiple R^2 and the adjusted R^2. These values show the model performance on train data.

If we have to interpret the multiple square value of R(0.6459), it means that the features explain the target by 64%.

4.Report the performance of the trained model with only one metric that you learned in the lecture and share the reason why you chose the metric.

Measuring model performance

The performance of the model needs to be checked on the test set. To do this, first of all, the predicted values of the target variable in the test set were calculated. The actual values of the target variable were removed from the test set.

```r
predicted_y <- predict(model, test[,-9])
head(predicted_y)
```

```
       1        2        3        4        5        6 
378962.7 255538.2 257098.7 188356.3 160631.5 222061.5 
```

Then, some performance criteria of the trained model were calculated. These; Mean squared error (MSE), root mean square error (RMSE), median absolute error (MAE).

```
error <- test$median_house_value - predicted_y
head(error)
```

```
        1          2          3          4          5          6
-26862.66  -14138.23  -15298.66  -88656.28  -55731.52 -112361.46
```

```
mse_model  <- mean(error ^ 2)
rmse_model <- sqrt(mean(error ^ 2))
mae_model  <- mean(abs(error))
```

```
mse_model
```

```
[1] 4580152023
```

```
rmse_model
```

```
[1] 67676.82
```

```
mae_model
```

```
[1] 49349.14
```

5.Check any problem related to over and underfitting.

Model performance on the train and test set is compared to check for any problems with over- or under-fitting in the model. Let's use RMSE for this:

```
rmse_train <- sqrt(mean((model$residuals) ^ 2))
rmse_test  <- rmse_model
```

```
rmse_train
```

```
[1] 68883.98
```

```r
rmse_test
```

```
[1] 67676.82
```

Then, the difference between the RMSE values was calculated.

```r
rmse_train - rmse_test
```

```
[1] 1207.155
```

Since the difference is positive, we can say that the model's performance on the train set is better than on the test set. Therefore, we can say that the model learned more from the test set, resulting in lower performance on the train set.

We can say that there may be an underfitting problem here, as the model learns more from the test test and insufficiently from the train set.

6.Create a new observation (it is up to you, just create an observation with the feature values you want), and predict the value of its target feature.

In this section, a new model was created by extracting longitude, latitude and population features and the values of the target feature were estimated.

```r
model.new <- lm(median_house_value ~  housing_median_age + total_rooms +
                total_bedrooms +  households + median_income +
              factor(ocean_proximity) , data = train)
```

```r
model.new
```

```

Call:
lm(formula = median_house_value ~ housing_median_age + total_rooms +
    total_bedrooms + households + median_income + factor(ocean_proximity),
    data = train)

Coefficients:
                      (Intercept)                 housing_median_age
                         11012.70                            1208.93
```

```
                    total_rooms                    total_bedrooms 
                         -14.78                            133.41 
                     households                     median_income 
                         -40.01                          42553.47 
  factor(ocean_proximity)INLAND     factor(ocean_proximity)ISLAND 
                      -63301.56                         194658.60 
factor(ocean_proximity)NEAR BAY factor(ocean_proximity)NEAR OCEAN 
                       12497.37                          18816.99 
```

```r
summary(model.new)
```

```
Call:
lm(formula = median_house_value ~ housing_median_age + total_rooms + 
    total_bedrooms + households + median_income + factor(ocean_proximity), 
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-580860  -44866  -11884   29447  494188 

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)                        1.101e+04  2.804e+03   3.928 8.61e-05 ***
housing_median_age                 1.209e+03  5.133e+01  23.552  < 2e-16 ***
total_rooms                       -1.478e+01  8.743e-01 -16.906  < 2e-16 ***
total_bedrooms                     1.334e+02  7.568e+00  17.627  < 2e-16 ***
households                        -4.001e+01  7.391e+00  -5.413 6.29e-08 ***
median_income                      4.255e+04  3.827e+02 111.207  < 2e-16 ***
factor(ocean_proximity)INLAND     -6.330e+04  1.449e+03 -43.690  < 2e-16 ***
factor(ocean_proximity)ISLAND      1.947e+05  3.606e+04   5.398 6.85e-08 ***
factor(ocean_proximity)NEAR BAY    1.250e+04  1.931e+03   6.471 1.00e-10 ***
factor(ocean_proximity)NEAR OCEAN  1.882e+04  1.784e+03  10.546  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 72060 on 16336 degrees of freedom
Multiple R-squared:  0.6128,    Adjusted R-squared:  0.6126 
F-statistic:  2873 on 9 and 16336 DF,  p-value: < 2.2e-16
```

```r
predicted_y.new <- predict(model.new, test[,-9])
head(predicted_y.new)
```

```
       1        2        3        4        5        6 
391784.5 239027.0 244724.6 186172.1 152460.7 217089.7 
```