Untitled5

March 25, 2023

[12]:

```
#loading the packages
library(tidyverse)
library(reshape2)
library(caret)
library(dplyr)
#reading the csv file
housing <- read.csv('C:/esra/housing.csv')

# displaying the first few rows of the housing dataset
head(housing)


# checking the dimensions of the dataset
dim(housing)

# checking variable types
str(housing)

# checking summary statistics of the dataset
summary(housing)

summary(housing$median_house_value)



# cleaning the data by imputing missing values
housing$total_bedrooms[is.na(housing$total_bedrooms)] <-␣
↪median(housing$total_bedrooms , na.rm = TRUE)

# creating new features
housing$mean_bedrooms <- housing$total_bedrooms/housing$households
housing$mean_rooms <- housing$total_rooms/housing$households
housing$price_per_sqft <- housing$median_house_value / housing$total_rooms

# removing unnecessary features
housing <- housing %>%
  select(-total_bedrooms, -total_rooms, -median_house_value)
```

```r
# splitting the data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(housing$price_per_sqft, p = 0.7, list = FALSE)
trainData <- housing[trainIndex, ]
testData <- housing[-trainIndex, ]

# training a linear regression model
lm.fit <- lm(price_per_sqft ~ ., data = trainData)
summary(lm.fit)

# evaluating the performance of the trained model using RMSE
# predict target feature using the trained model for training set
pred_train <- predict(lm.fit, trainData)

# compute the RMSE for training set
train_rmse <- sqrt(mean((trainData$price_per_sqft - pred_train)^2))

# predict target feature using the trained model for testing set
pred_test <- predict(lm.fit, testData)

# compute the RMSE for testing set
test_rmse <- sqrt(mean((testData$price_per_sqft - pred_test)^2))

# print the RMSE values for both training and testing sets
cat("Training RMSE: ", train_rmse, "\n")
cat("Testing RMSE: ", test_rmse, "\n")

# checking for overfitting and underfitting
cat("Training RMSE: ", train_rmse, "\n")
cat("Testing RMSE: ", test_rmse, "\n")


# creating a new observation and predicting its target feature value
new_observation <- data.frame(
  total_rooms = 2500,
  housing_median_age = 30,
  population = 1500,
  households = 600,
  median_income = 4.5,
  ocean_proximity = "NEAR OCEAN"
)

# predict the median house value using the trained model
new_observation$median_house_value <- predict(lm.fit, newdata = new_observation)

# display the predicted median house value
```

```r
cat("Predicted median house value: $", new_observation$median_house_value)
```

A data.frame: 6 × 10

	longitude <dbl>	latitude <dbl>	housing_median_age <dbl>	total_rooms <dbl>	total_bedrooms <dbl>	populat <dbl>
1	-122.23	37.88	41	880	129	322
2	-122.22	37.86	21	7099	1106	2401
3	-122.24	37.85	52	1467	190	496
4	-122.25	37.85	52	1274	235	558
5	-122.25	37.85	52	1627	280	565
6	-122.25	37.85	52	919	213	413

1. 20640 2. 10

```
'data.frame':   20640 obs. of  10 variables:
 $ longitude         : num  -122 -122 -122 -122 -122 …
 $ latitude          : num  37.9 37.9 37.9 37.9 37.9 …
 $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 …
 $ total_rooms       : num  880 7099 1467 1274 1627 …
 $ total_bedrooms    : num  129 1106 190 235 280 …
 $ population        : num  322 2401 496 558 565 …
 $ households        : num  126 1138 177 219 259 …
 $ median_income     : num  8.33 8.3 7.26 5.64 3.85 …
 $ median_house_value: num  452600 358500 352100 341300 342200 …
 $ ocean_proximity   : chr  "NEAR BAY" "NEAR BAY" "NEAR BAY" "NEAR BAY" …
```

```
   longitude         latitude     housing_median_age  total_rooms
 Min.   :-124.3   Min.   :32.54   Min.   : 1.00      Min.   :    2
 1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1448
 Median :-118.5   Median :34.26   Median :29.00      Median : 2127
 Mean   :-119.6   Mean   :35.63   Mean   :28.64      Mean   : 2636
 3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00      3rd Qu.: 3148
 Max.   :-114.3   Max.   :41.95   Max.   :52.00      Max.   :39320

 total_bedrooms     population      households     median_income
 Min.   :   1.0   Min.   :    3   Min.   :   1.0   Min.   : 0.4999
 1st Qu.: 296.0   1st Qu.:  787   1st Qu.: 280.0   1st Qu.: 2.5634
 Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
 Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
 3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
 Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
 NA's   :207
 median_house_value ocean_proximity
 Min.   : 14999     Length:20640
 1st Qu.:119600     Class :character
 Median :179700     Mode  :character
 Mean   :206856
 3rd Qu.:264725
 Max.   :500001
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  14999  119600  179700  206856  264725  500001 


Call:
lm(formula = price_per_sqft ~ ., data = trainData)

Residuals:
   Min     1Q Median     3Q    Max 
 -3090    -95    -57    -11  68339 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)               -4.172e+02  1.376e+03  -0.303   0.7618    
longitude                 -8.614e+00  1.594e+01  -0.540   0.5889    
latitude                  -1.197e+01  1.578e+01  -0.758   0.4483    
housing_median_age        -9.787e-01  6.800e-01  -1.439   0.1501    
population                 9.988e-03  1.536e-02   0.650   0.5156    
households                -2.946e-01  4.626e-02  -6.369 1.96e-10 ***
median_income              4.714e+01  5.123e+00   9.202  < 2e-16 ***
ocean_proximityINLAND      1.806e+01  2.690e+01   0.672   0.5018    
ocean_proximityISLAND      1.208e+02  4.489e+02   0.269   0.7878    
ocean_proximityNEAR BAY    8.490e+01  2.973e+01   2.856   0.0043 ** 
ocean_proximityNEAR OCEAN  2.638e+01  2.440e+01   1.081   0.2796    
mean_bedrooms              1.544e+02  2.551e+01   6.053 1.46e-09 ***
mean_rooms                -4.462e+01  5.816e+00  -7.671 1.82e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 896.9 on 14435 degrees of freedom
Multiple R-squared:  0.02005,	Adjusted R-squared:  0.01923 
F-statistic: 24.61 on 12 and 14435 DF,  p-value: < 2.2e-16


Training RMSE:  896.5064 
Testing RMSE:  634.9935 
Training RMSE:  896.5064 
Testing RMSE:  634.9935 
```

```
Error in eval(predvars, data, env): 'longitude' nesnesi bulunamadı
Traceback:

1. predict(lm.fit, newdata = new_observation)
2. predict.lm(lm.fit, newdata = new_observation)
3. model.frame(Terms, newdata, na.action = na.action, xlev = object$xlevels)
4. model.frame.default(Terms, newdata, na.action = na.action, xlev = 
 ↪object$xlevels)
5. eval(predvars, data, env)
```

```
6. eval(predvars, data, env)
```

[11]:

```
#loading the packages
library(tidyverse)
library(reshape2)
library(caret)
library(dplyr)
#reading the csv file
housing <- read.csv('C:/esra/housing.csv')

# displaying the first few rows of the housing dataset
head(housing)
names(housing) # display column names

# checking the dimensions of the dataset
dim(housing)

# checking variable types
str(housing)

# checking summary statistics of the dataset
summary(housing)


# cleaning the data by imputing missing values
housing$total_bedrooms[is.na(housing$total_bedrooms)] <-␣
 ↪median(housing$total_bedrooms , na.rm = TRUE)

# creating new features
housing$mean_bedrooms <- housing$total_bedrooms/housing$households
housing$mean_rooms <- housing$total_rooms/housing$households
housing$price_per_sqft <- housing$median_house_value / housing$total_rooms

# removing unnecessary features
housing <- housing %>%
  select(-total_bedrooms, -total_rooms, -median_house_value)

# splitting the data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(housing$price_per_sqft, p = 0.7, list = FALSE)
trainData <- housing[trainIndex, ]
testData <- housing[-trainIndex, ]

# training a linear regression model
lm.fit <- lm(price_per_sqft ~ ., data = trainData)
summary(lm.fit)
```

```r
# evaluating the performance of the trained model using RMSE
# predict target feature using the trained model for training set
pred_train <- predict(lm.fit, trainData)

# compute the RMSE for training set
train_rmse <- sqrt(mean((trainData$price_per_sqft - pred_train)^2))

# predict target feature using the trained model for testing set
pred_test <- predict(lm.fit, testData)

# compute the RMSE for testing set
test_rmse <- sqrt(mean((testData$price_per_sqft - pred_test)^2))

# print the RMSE values for both training and testing sets
cat("Training RMSE: ", train_rmse, "\n")
cat("Testing RMSE: ", test_rmse, "\n")

# checking for overfitting and underfitting
cat("Training RMSE: ", train_rmse, "\n")
cat("Testing RMSE: ", test_rmse, "\n")


# creating a new observation and predicting its target feature value
new_observation <- data.frame(
  total_rooms = 2500,
  housing_median_age = 30,
  population = 1500,
  households = 600,
  median_income = 4.5,
  ocean_proximity = "NEAR OCEAN"
)

# predict the median house value using the trained model
new_observation$median_house_value <- predict(lm.fit, newdata = new_observation)

# display the predicted median house value
cat("Predicted median house value: $", new_observation$median_house_value)
```

A data.frame: 6 × 10

	longitude <dbl>	latitude <dbl>	housing_median_age <dbl>	total_rooms <dbl>	total_bedrooms <dbl>	populat <dbl>
1	-122.23	37.88	41	880	129	322
2	-122.22	37.86	21	7099	1106	2401
3	-122.24	37.85	52	1467	190	496
4	-122.25	37.85	52	1274	235	558
5	-122.25	37.85	52	1627	280	565
6	-122.25	37.85	52	919	213	413

1. 'longitude' 2. 'latitude' 3. 'housing_median_age' 4. 'total_rooms' 5. 'total_bedrooms' 6. 'pop-

ulation' 7. 'households' 8. 'median_income' 9. 'median_house_value' 10. 'ocean_proximity'

1. 20640 2. 10

```
'data.frame':   20640 obs. of  10 variables:
 $ longitude         : num  -122 -122 -122 -122 -122 …
 $ latitude          : num  37.9 37.9 37.9 37.9 37.9 …
 $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 …
 $ total_rooms       : num  880 7099 1467 1274 1627 …
 $ total_bedrooms    : num  129 1106 190 235 280 …
 $ population        : num  322 2401 496 558 565 …
 $ households        : num  126 1138 177 219 259 …
 $ median_income     : num  8.33 8.3 7.26 5.64 3.85 …
 $ median_house_value: num  452600 358500 352100 341300 342200 …
 $ ocean_proximity   : chr  "NEAR BAY" "NEAR BAY" "NEAR BAY" "NEAR BAY" …
```

```
   longitude         latitude     housing_median_age  total_rooms
 Min.   :-124.3   Min.   :32.54   Min.   : 1.00      Min.   :    2
 1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1448
 Median :-118.5   Median :34.26   Median :29.00      Median : 2127
 Mean   :-119.6   Mean   :35.63   Mean   :28.64      Mean   : 2636
 3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00      3rd Qu.: 3148
 Max.   :-114.3   Max.   :41.95   Max.   :52.00      Max.   :39320

 total_bedrooms     population      households     median_income
 Min.   :   1.0   Min.   :    3   Min.   :   1.0   Min.   : 0.4999
 1st Qu.: 296.0   1st Qu.:  787   1st Qu.: 280.0   1st Qu.: 2.5634
 Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
 Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
 3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
 Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
 NA's   :207
 median_house_value ocean_proximity
 Min.   : 14999     Length:20640
 1st Qu.:119600     Class :character
 Median :179700     Mode  :character
 Mean   :206856
 3rd Qu.:264725
 Max.   :500001
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  14999  119600  179700  206856  264725  500001
```

```
Call:
lm(formula = price_per_sqft ~ ., data = trainData)

Residuals:
   Min     1Q Median     3Q    Max
```

```
 -3090    -95    -57    -11  68339

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)               -4.172e+02  1.376e+03  -0.303   0.7618
longitude                 -8.614e+00  1.594e+01  -0.540   0.5889
latitude                  -1.197e+01  1.578e+01  -0.758   0.4483
housing_median_age        -9.787e-01  6.800e-01  -1.439   0.1501
population                 9.988e-03  1.536e-02   0.650   0.5156
households                -2.946e-01  4.626e-02  -6.369 1.96e-10 ***
median_income              4.714e+01  5.123e+00   9.202  < 2e-16 ***
ocean_proximityINLAND      1.806e+01  2.690e+01   0.672   0.5018
ocean_proximityISLAND      1.208e+02  4.489e+02   0.269   0.7878
ocean_proximityNEAR BAY    8.490e+01  2.973e+01   2.856   0.0043 **
ocean_proximityNEAR OCEAN  2.638e+01  2.440e+01   1.081   0.2796
mean_bedrooms              1.544e+02  2.551e+01   6.053 1.46e-09 ***
mean_rooms                -4.462e+01  5.816e+00  -7.671 1.82e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 896.9 on 14435 degrees of freedom
Multiple R-squared:  0.02005,	Adjusted R-squared:  0.01923
F-statistic: 24.61 on 12 and 14435 DF,  p-value: < 2.2e-16


Training RMSE:  896.5064
Testing RMSE:  634.9935
Training RMSE:  896.5064
Testing RMSE:  634.9935
```

```
Error in eval(predvars, data, env): 'longitude' nesnesi bulunamadı
Traceback:

1. predict(lm.fit, newdata = new_observation)
2. predict.lm(lm.fit, newdata = new_observation)
3. model.frame(Terms, newdata, na.action = na.action, xlev = object$xlevels)
4. model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels)
5. eval(predvars, data, env)
6. eval(predvars, data, env)
```

[ ]:
```
California Housing Prices
In this homework we are Predicting of median house prices for California districts and make a regression analysis for it.

   Packcages
```

```r
library(tidyverse)#that helps to transform and better present data. It assists
  ↪with data import, tidying, manipulation, and data visualization
library(reshape2)#package is used for restructuring data frames into a format
  ↪that is suitable for analysis.
library(caret)#package provides a set of functions for training and testing
  ↪predictive models. It includes tools for data preprocessing, feature
  ↪selection, model tuning, and performance evaluation.
library(dplyr)#to make data manipulation
    Dataset
We import dataset from kaggle.

#reading the csv file
housing <- read.csv('C:/esra/housing.csv')

# checking the dimensions of the dataset
dim(housing)

A data.frame: 6 × 10
longitude        latitude        housing_median_age        total_rooms        total_bedrooms
<dbl>        <dbl>        <dbl>        <dbl>        <dbl>        <dbl>        <dbl>        <db
1        -122.23        37.
  ↪88        41        880        129        322        126        8.
  ↪3252        452600        NEAR BAY
2        -122.22        37.
  ↪86        21        7099        1106        2401        1138        8.
  ↪3014        358500        NEAR BAY
3        -122.24        37.
  ↪85        52        1467        190        496        177        7.
  ↪2574        352100        NEAR BAY
4        -122.25        37.
  ↪85        52        1274        235        558        219        5.
  ↪6431        341300        NEAR BAY
5        -122.25        37.
  ↪85        52        1627        280        565        259        3.
  ↪8462        342200        NEAR BAY
6        -122.25        37.
  ↪85        52        919        213        413        193        4.
  ↪0368        269700        NEAR BAY


# checking variable types
str(housing)

# checking summary statistics of the dataset
summary(housing)
```

```
'data.frame':        20640 obs. of  10 variables:
 $ longitude         : num  -122 -122 -122 -122 -122 ...
 $ latitude          : num  37.9 37.9 37.9 37.9 37.9 ...
 $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 ...
 $ total_rooms       : num  880 7099 1467 1274 1627 ...
 $ total_bedrooms    : num  129 1106 190 235 280 ...
 $ population        : num  322 2401 496 558 565 ...
 $ households        : num  126 1138 177 219 259 ...
 $ median_income     : num  8.33 8.3 7.26 5.64 3.85 ...
 $ median_house_value: num  452600 358500 352100 341300 342200 ...
 $ ocean_proximity   : chr  "NEAR BAY" "NEAR BAY" "NEAR BAY" "NEAR BAY" ...

the summary of the housing data has 20640 observations and 10 variables. Here
 ↪is the description of each variable:

longitude: The longitude of the location of the house.
latitude: The latitude of the location of the house.
housing_median_age: The median age of the houses in the location.
total_rooms: Total number of rooms in the houses.
total_bedrooms: Total number of bedrooms in the houses.
population: Total population of the location.
households: Total number of households in the location.
median_income: Median income of the households in the location.
median_house_value: Median value of the houses in the location.
ocean_proximity: Proximity of the location to the ocean.

 longitude         latitude     housing_median_age  total_rooms
 Min.   :-124.3   Min.   :32.54   Min.   : 1.00      Min.   :    2
 1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1448
 Median :-118.5   Median :34.26   Median :29.00      Median : 2127
 Mean   :-119.6   Mean   :35.63   Mean   :28.64      Mean   : 2636
 3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00      3rd Qu.: 3148
 Max.   :-114.3   Max.   :41.95   Max.   :52.00      Max.   :39320

 total_bedrooms     population      households     median_income
 Min.   :   1.0   Min.   :     3   Min.   :   1.0   Min.   : 0.4999
 1st Qu.: 296.0   1st Qu.:   787   1st Qu.: 280.0   1st Qu.: 2.5634
 Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
 Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
 3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
 Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
 NA's   :207

median_house_value ocean_proximity
 Min.   : 14999     Length:20640
 1st Qu.:119600     Class :character
 Median :179700     Mode  :character
```

```
 Mean   :206856
 3rd Qu.:264725
 Max.   :500001

Training

Regression Model

Call:
lm(formula = price_per_sqft ~ ., data = trainData)

Residuals:
   Min     1Q Median     3Q    Max
 -3090    -95    -57    -11  68339

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)               -4.172e+02  1.376e+03  -0.303   0.7618
longitude                 -8.614e+00  1.594e+01  -0.540   0.5889
latitude                  -1.197e+01  1.578e+01  -0.758   0.4483
housing_median_age        -9.787e-01  6.800e-01  -1.439   0.1501
population                 9.988e-03  1.536e-02   0.650   0.5156
households                -2.946e-01  4.626e-02  -6.369 1.96e-10 ***
median_income              4.714e+01  5.123e+00   9.202  < 2e-16 ***
ocean_proximityINLAND      1.806e+01  2.690e+01   0.672   0.5018
ocean_proximityISLAND      1.208e+02  4.489e+02   0.269   0.7878
ocean_proximityNEAR BAY    8.490e+01  2.973e+01   2.856   0.0043 **
ocean_proximityNEAR OCEAN  2.638e+01  2.440e+01   1.081   0.2796
mean_bedrooms              1.544e+02  2.551e+01   6.053 1.46e-09 ***
mean_rooms                -4.462e+01  5.816e+00  -7.671 1.82e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 896.9 on 14435 degrees of freedom
Multiple R-squared:  0.02005,	Adjusted R-squared:  0.01923
F-statistic: 24.61 on 12 and 14435 DF,  p-value: < 2.2e-16

Trained RMSE
# evaluating the performance of the trained model using RMSE
# predict target feature using the trained model for training set
# compute the RMSE for training set
# predict target feature using the trained model for testing set
# compute the RMSE for testing set
# print the RMSE values for both training and testing sets
# checking for overfitting and underfitting

Training RMSE:  896.5064
```

```
Testing RMSE:  634.9935
Training RMSE:  896.5064
Testing RMSE:  634.9935

Predicting
# creating a new observation and predicting its target feature value
According to One-Hotline Enconding ocean proximity
ocean_proximity1 ocean_proximity2 ocean_proximity3 ocean_proximity4␣
 ↪ocean_proximity
1                0                0                0                4         ␣
 ↪ISLAND
2                0                0                3               -1      NEAR␣
 ↪OCEAN
3                0                2               -1               -1         ␣
 ↪INLAND
4                1               -1               -1               -1       <1H␣
 ↪OCEAN
5               -1               -1               -1               -1       ␣
 ↪NEAR BAY


There is no predicting the model because i couldn't find the right endcoding␣
 ↪Hotline
```