Day 3

Regression Example

Our data: Boston housing

Housing data for 506 census tracts of Boston from the 1970 census. The dataframe `Boston` contains the original data by Harrison and Rubinfeld (1979).

Foto de Scott Webb: https://www.pexels.com/pt-br/foto/casa-de-madeira-branca-e-vermelha-com-cerca-1029599/

Response variable: median value of owner-occupide homes in USD 1000's

The datasets packages MASS and mlbench in R already contains it (data frame named **boston**). 🏘️

Understand the Boston housing data set

- Number of observation: 506
- Number of variables: 13 numeric/categorical predictive.
- The Median Value (14) is the response variable.

1. CRIM per capita crime rate by town.
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS proportion of non-retail business acres per town
4. CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX nitric oxides concentration (parts per 10 million)
6. RM average number of rooms per dwelling
7. AGE proportion of owner-occupied units built prior to 1940
8. DIS weighted distances to five Boston employment centres
9. RAD index of accessibility to radial highways
10. TAX full-value property-tax rate per $10,000
11. PTRATIO pupil-teacher ratio by town 1
12. B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT % lower status of the population

13. LSTAT % lower status of the population
14. PRICE Median value of owner-occupied homes in $1000's

▾ Code

```
library(MASS)
data("Boston")
head(Boston)
```

```
     crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
  medv
1 24.0
2 21.6
3 34.7
4 33.4
5 36.2
6 28.7
```

Package

Caret package

- Classification and Regression Training - Version 6.0-94
- Functions for training and plotting classification and regression models.

📌 Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 28(5), 1–26. https://doi.org/10.18637/jss.v028.i05

http://topepo.github.io/caret/index.html

The packages

▾ Code

```
library(tidyverse)
library(tidymodels)
library(skimr)
library(caret)
library(broom)
library(ggplot2)
library(mlbench)
```

Summarize data set

▾ Code

```
glimpse(Boston)
```

```
Rows: 506
Columns: 14
$ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985, 0.08829,…
$ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12.5, 12.5, 1…
$ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87, 7.87, 7.…
$ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
$ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.524, 0.524,…
$ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.172, 5.631,…
$ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100.0, 85.9, 9…
$ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.5605, 5.9505…
$ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4,…
$ tax     <dbl> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 311, 311, 31…
$ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2, 15.2, 15…
$ black   <dbl> 396.90, 396.90, 392.83, 394.63, 396.90, 394.12, 395.60, 396.90…
$ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29.93, 17.10…
$ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15…
```

Descriptive statistics

▾ Code

```
Boston %>% skim()
```

Data summary

Name	Piped data
Number of rows	506
Number of columns	14
_______________________	
Column type frequency:	
numeric	14
________________________	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
crim	0	1	3.61	8.60	0.01	0.08	0.26	3.68	88.98	▇▁▁▁▁
zn	0	1	11.36	23.32	0.00	0.00	0.00	12.50	100.00	▇▁▁▁▁
indus	0	1	11.14	6.86	0.46	5.19	9.69	18.10	27.74	▇▆▁▇▁

chas	0	1	0.07	0.25	0.00	0.00	0.00	0.00	1.00	
nox	0	1	0.55	0.12	0.38	0.45	0.54	0.62	0.87	
rm	0	1	6.28	0.70	3.56	5.89	6.21	6.62	8.78	
age	0	1	68.57	28.15	2.90	45.02	77.50	94.07	100.00	
dis	0	1	3.80	2.11	1.13	2.10	3.21	5.19	12.13	
rad	0	1	9.55	8.71	1.00	4.00	5.00	24.00	24.00	
tax	0	1	408.24	168.54	187.00	279.00	330.00	666.00	711.00	
ptratio	0	1	18.46	2.16	12.60	17.40	19.05	20.20	22.00	
black	0	1	356.67	91.29	0.32	375.38	391.44	396.22	396.90	
lstat	0	1	12.65	7.14	1.73	6.95	11.36	16.96	37.97	
medv	0	1	22.53	9.20	5.00	17.02	21.20	25.00	50.00	

Optional: skim (Boston)

Time to create some models

Regression

Predicted variable: class of iris plant.

Model	Col2
lm	Multiple linear regression
KNN	K-nearest neighbor
DT	Decision Tree
RF	Random Forest

Metrics

- R^2
- RMSE (Root mean square error)
- MAE (Mean absolute error)

Split data: train and test

Select 80% of the data for training and use the remaining 20% to test. The iris data frame was renamed to dataset.

▾ Code

```
dataset<-Boston
```

▾ Code

```
validation_index <- createDataPartition(dataset$medv, p=0.80, list=FALSE)
test <- dataset[-validation_index,]
train <- dataset[validation_index,]
```

Train control

Control the computational *nuances* of the caret `train` function. The function `train` *train* the model.

Resampling methods, pre-processing,....

https://rdrr.io/rforge/caret/man/trainControl.html

▾ Code

```
control <- trainControl(method="cv", number=10)
```

We are using cross-validation and number of $folds = 10$.

Now, the models 😊!

Multiple linear regression

Training the model (named as *fit.lm*)

fit.lm <- train(**response variable ~. predictors**, **data**=train, **method**="machine learning method", **trControl**="nuances", **metric** = metric to select optimal model,...)

More information about the train function: tipe **?train** or **help(train)**

▾ Code

```
set.seed(7)
fit.lm <- train(medv~., data=train, method="lm", trControl=control)
fit.lm
```

```
Linear Regression

407 samples
 13 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 365, 366, 366, 367, 366, 366, ...
Resampling results:
```

```
  RMSE      Rsquared   MAE     
  4.870279  0.7236559  3.482161

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Multiple Linear Regression - results

▾ Code

```
summary(fit.lm)
```

```

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.7314  -2.8525  -0.5013   1.9473  24.5332 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  39.635788   5.719247   6.930 1.73e-11 ***
crim         -0.067993   0.048260  -1.409 0.159659    
zn            0.053186   0.015245   3.489 0.000540 ***
indus         0.030776   0.070268   0.438 0.661643    
chas          1.822603   1.027923   1.773 0.076988 .  
nox         -17.073492   4.464321  -3.824 0.000152 ***
rm            3.471845   0.457995   7.581 2.50e-13 ***
age           0.015247   0.015321   0.995 0.320277    
dis          -1.430413   0.231794  -6.171 1.69e-09 ***
rad           0.309473   0.075538   4.097 5.09e-05 ***
tax          -0.012686   0.004183  -3.033 0.002584 ** 
ptratio      -1.033587   0.148060  -6.981 1.26e-11 ***
black         0.009003   0.003196   2.817 0.005096 ** 
lstat        -0.605533   0.059146 -10.238  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.833 on 393 degrees of freedom
Multiple R-squared:  0.7372,	Adjusted R-squared:  0.7285 
F-statistic: 84.79 on 13 and 393 DF,  p-value: < 2.2e-16
```

KNN K-nearest neighbors

▾ Code

```
set.seed(7)
fit.knn <- train(medv~., data=train, method="knn",  trControl=control)
fit.knn
```

```
k-Nearest Neighbors

407 samples
 13 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 365, 366, 366, 367, 366, 366, ...
Resampling results across tuning parameters:

  k  RMSE      Rsquared   MAE
  5  6.288218  0.5377518  4.374566
  7  6.617072  0.4926125  4.552685
  9  6.782052  0.4657837  4.656384

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 5.
```

SVM

▾ Code

```r
set.seed(7)
fit.svm <- train(medv~., data=train, method="svmRadial",  trControl=control)
fit.svm
```

```
Support Vector Machines with Radial Basis Function Kernel

407 samples
 13 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 365, 366, 366, 367, 366, 366, ...
Resampling results across tuning parameters:

  C     RMSE      Rsquared   MAE
  0.25  5.003312  0.7469303  2.962434
  0.50  4.550092  0.7805059  2.673854
  1.00  4.182837  0.8100139  2.455039

Tuning parameter 'sigma' was held constant at a value of 0.1162036
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were sigma = 0.1162036 and C = 1.
```

Decision tree

▾ Code

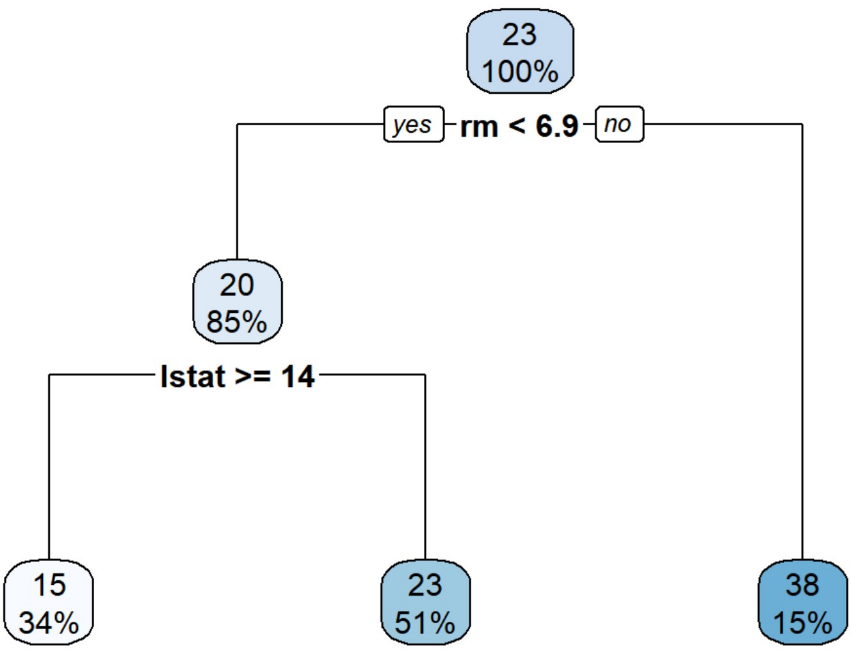
```
set.seed(7)
fit.dt <- train(medv~., data=train, method="rpart", trControl=control)
fit.dt
```

```
CART

407 samples
 13 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 365, 366, 366, 367, 366, 366, ...
Resampling results across tuning parameters:

  cp          RMSE      Rsquared   MAE
  0.06436844  5.476396  0.6615684  3.905818
  0.15933830  6.404642  0.5262825  4.782638
  0.48410952  8.199192  0.4222233  5.920834

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was cp = 0.06436844.
```

Plotting the regression tree

▾ Code

```
library(rpart.plot)
rpart.plot(fit.dt$finalModel)
```



Random Forest

▼ Code

```
set.seed(7)
fit.rf <- train(medv~., data=train, method="rf", trControl=control)
fit.rf
```

```
Random Forest

407 samples
 13 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 365, 366, 366, 367, 366, 366, ...
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   2    3.735710  0.8568368  2.477678
   7    3.319245  0.8769558  2.226898
  13    3.371298  0.8650413  2.288587

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 7.
```

Select the best model

- select best model - resampling and looking the results
- summarize accuracy of models : summary (results)

▼ Code

```
results <- resamples(list(lm=fit.lm, knn=fit.knn, dt= fit.dt,svm=fit.svm,rf=fit.rf))
summary(results)
```

```
Call:
summary.resamples(object = results)

Models: lm, knn, dt, svm, rf
Number of resamples: 10

MAE
        Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
lm  2.765596 2.964188 3.544285 3.482161 3.898753 4.195774    0
```

```
knn 3.256098 3.942607 4.316043 4.374566 4.618768 5.934000    0
dt  3.238224 3.593764 3.782765 3.905818 4.029158 4.935664    0
svm 1.844987 2.099272 2.487437 2.455039 2.675540 3.076773    0
rf  1.524718 2.106078 2.297552 2.226898 2.469792 2.628594    0

RMSE
        Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
lm  3.730852 4.166125 4.934179 4.870279 5.353522 6.178058    0
knn 4.872264 5.637115 6.246201 6.288218 6.583387 8.283836    0
dt  4.253609 4.777866 5.325916 5.476396 6.066377 7.053901    0
svm 2.503149 3.775430 4.272882 4.182837 4.954876 5.199541    0
rf  2.158573 2.975016 3.373441 3.319245 3.742984 4.379391    0

Rsquared
         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
lm  0.5409714 0.6766174 0.7353430 0.7236559 0.7987698 0.8144139    0
knn 0.3051559 0.4707871 0.5459107 0.5377518 0.6350185 0.7114189    0
dt  0.4846939 0.6034444 0.6928479 0.6615684 0.7388793 0.7609270    0
svm 0.7437077 0.7595417 0.7896694 0.8100139 0.8395469 0.9326640    0
rf  0.7699901 0.8574486 0.8760226 0.8769558 0.9172695 0.9413548    0
```

Best Model

🏠 Random forest 🌟

▼ Code

```
print(fit.rf)
```

```
Random Forest

407 samples
 13 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 365, 366, 366, 367, 366, 366, ...
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   2    3.735710  0.8568368  2.477678
   7    3.319245  0.8769558  2.226898
  13    3.371298  0.8650413  2.288587

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 7.
```

Make some predictions

Predictions with test data

▾ Code

```
predictions <- predict(fit.rf, test)
```

Accuracy

Our results

▾ Code

```
postResample(pred=predictions,obs=test$medv)
```

```
     RMSE  Rsquared       MAE 
2.6578095 0.9245503 2.1056066 
```

▾ Code

```
ggplot(test)+geom_point(aes(x=test$medv,y=predictions))
```