Exercises ISLR – Ch.4

Marcelo Previato Simoes Nº 2367070

08/09/2025

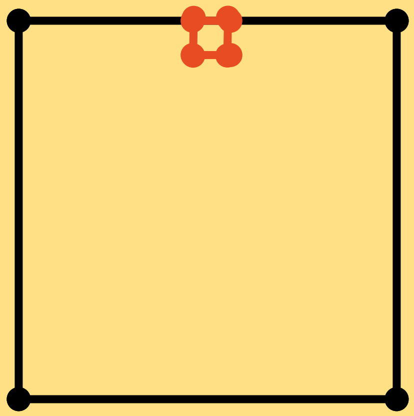
Conceptual

Exercises: 4

Exercise 4) Curse of Dimensionality (a-b)

a) **1 feature:** **0.1** of the observations

b) **2 features:** 0.1 x 0.1 = **0.01** of the observations



Exercise 4) Curse of Dimensionality (c-d)

c) 100 features: $(0.1)^{100} =$ 10^(-100)

d) Points near observation decrease exponentially, so it is necessary to increase the distance to get more neighbors, which increase the noise

Exercise 4) Curse of Dimensionality (e)

Length of the size of the hypercube to have 10% of data

$p = 1 \rightarrow k = 0.100$

$p = 2 \rightarrow k^2 = 0.1 \rightarrow k = 0.1^{(1/2)} \sim 0.316$

$p = 100 \rightarrow k^{100} = 0.1 \rightarrow k = 0.1^{(1/100)} \sim 0.977$

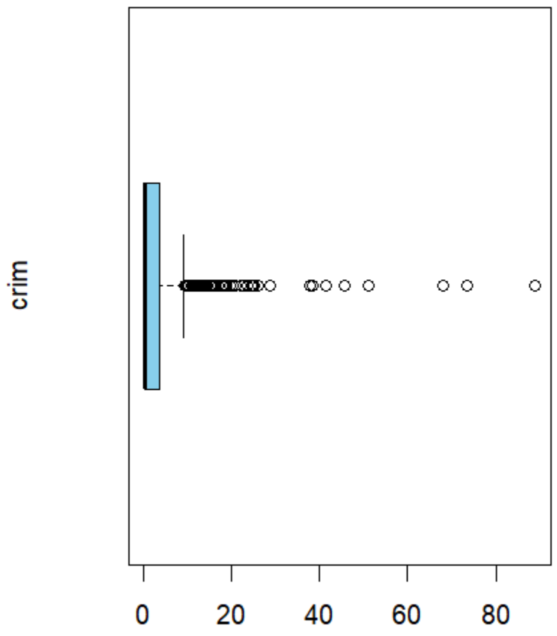
The size of the hypercube is getting bigger, so points are getting further from the observation.

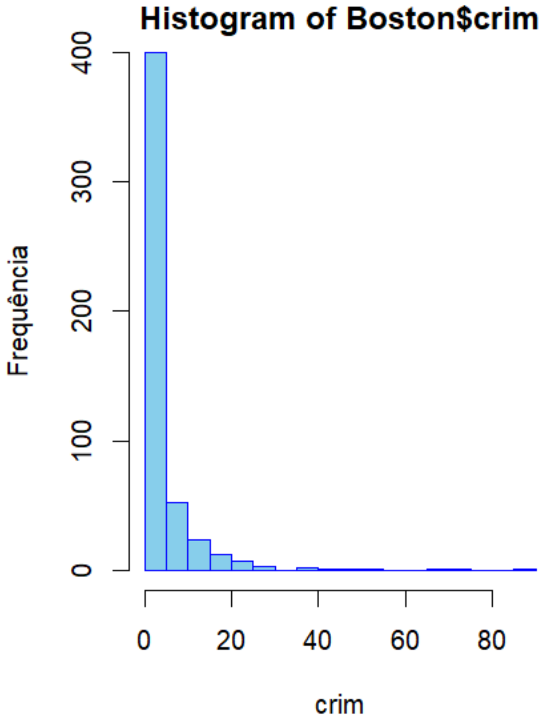
Applied

Exercise 16 – Boston dataset

Ex.16) Data Prep

- Create **binary variable of high crime** (1 – above median, 0 – below median)
- Create **training and test subsets:** 70% training | 30% test (random)






```
      crim
Min.   : 0.00632
1st Qu.: 0.08205
Median : 0.25651
Mean   : 3.61352
3rd Qu.: 3.67708
Max.   :88.97620
```

Ex.16) Logistic Regression (all predictors)

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -16.137952   9.184105  -1.757 0.078890 .
zn           -0.076043   0.036929  -2.059 0.039478 *
indus        -0.064802   0.062073  -1.044 0.296500
chas          1.237067   0.984795   1.256 0.209056
nox          46.772792   8.707875   5.371 7.82e-08 ***
rm           -0.542088   0.834311  -0.650 0.515858
age           0.019331   0.013568   1.425 0.154236
dis           0.633277   0.262238   2.415 0.015740 *
rad           0.601721   0.172811   3.482 0.000498 ***
tax          -0.006860   0.003594  -1.909 0.056278 .
ptratio       0.334189   0.158840   2.104 0.035384 *
black        -0.047817   0.019060  -2.509 0.012114 *
lstat        -0.012217   0.060798  -0.201 0.840743
medv          0.139175   0.077494   1.796 0.072502 .
```

```
Confusion Matrix - Test Data:
    True
Pred  0  1
   0 73  4
   1 12 63

Accuracy:    0.895
Sensitivity: 0.940
Specificity: 0.859
```

- Very good performance in predicting classification (89.5%, 136 out of 152)
- Higher sensitivity (better in identifying positives) than specificity (negatives)

Ex.16) Logistic Regression (few predictors)

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -8.34010    7.05097  -1.183   0.2369    
zn          -0.06917    0.03068  -2.255   0.0241 *  
nox         32.13938    5.95502   5.397 6.78e-08 ***
dis          0.41376    0.20660   2.003   0.0452 *  
rad          0.52514    0.13423   3.912 9.15e-05 ***
ptratio     -0.02735    0.10340  -0.265   0.7914    
black       -0.03180    0.01463  -2.173   0.0298 *  
```

```
Confusion Matrix - Test Data:
    True
Pred  0  1
   0 71  7
   1 14 60

Accuracy:    0.862
Sensitivity: 0.896
Specificity: 0.835
```

- Deterioration of classification: 89.5% → 86.2% (16 → 21 errors)
- Keep full or try different subsets or information criteria

Ex.16) Linear Discriminant Analysis (LDA)

Predictors: All variables

Priors: based on training data

```
        0         1
0.4745763 0.5254237
```

```
Confusion Matrix - Test Data:
     True
Pred  0  1
   0 79 13
   1  6 54

Accuracy:    0.875
Sensitivity: 0.806
Specificity: 0.929
```

- Slightly worse than Logistic regression with all predictors: 89.5% → 87.5%
- Distinct balance between type of errors: worse sensitivity (94.0% → 80.6%), better specificity (85.9%→92.9%)

Ex.16) KNN: N = 1 to 10

K = 1

```
Confusion Matrix - Test Data:
     True
Pred  0  1
   0 72  5
   1 13 62

Accuracy:    0.882
Sensitivity: 0.925
Specificity: 0.847
```

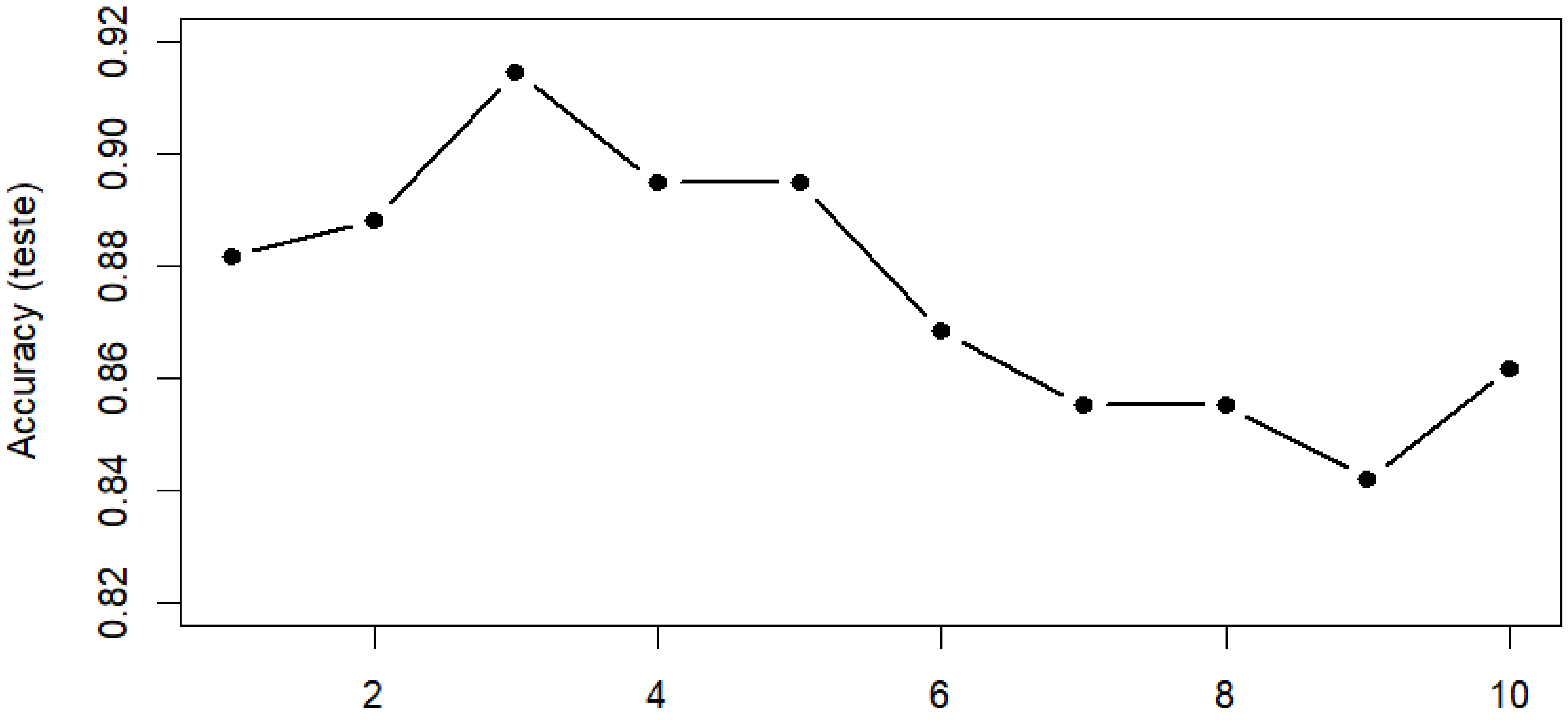
K = 3

```
Confusion Matrix - Test Data:
     True
Pred  0  1
   0 77  5
   1  8 62

Accuracy:    0.914
Sensitivity: 0.925
Specificity: 0.906
```

- KNN could perform better than the other methods depending on the choice K
- Improved performance for K=3: 89.5% → 91.4% (16 → 13 errors)

Ex.16) KNN: N = 1 to 10



Ex.16) Full Comparison

	Metodo <chr>	Acuracia <dbl>	Sensibilidade <dbl>	Especificidade <dbl>
7	KNN (k=3)	0.914	0.925	0.906
1	Reg Log (full)	0.895	0.940	0.859
8	KNN (k=4)	0.895	0.925	0.871
9	KNN (k=5)	0.895	0.896	0.894
6	KNN (k=2)	0.888	0.940	0.847
5	KNN (k=1)	0.882	0.925	0.847
3	LDA (full)	0.875	0.806	0.929
2	Reg Log (short)	0.862	0.896	0.835
4	Naive Bayes (full)	0.836	0.821	0.847