

Reviewing ML

general remarks:

I didn't find any flaws in the content so far. However, there is little told about the functions in R, was that intended?

In the very beginning I had minor difficulties understanding the structure of the course, for some reason.

Maybe it is just me - if not we could add a little more detailed description.

After 2 chapters the structure was clear and also made sense.

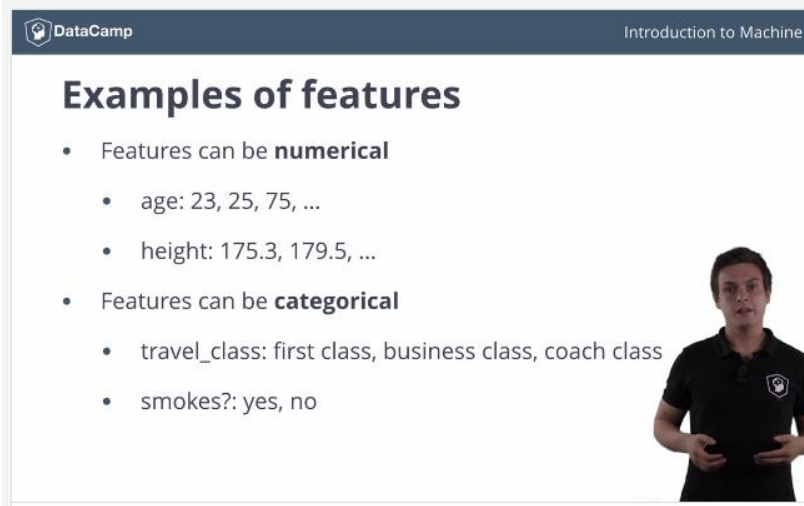
I think the exercises are on a more advanced level than most other courses but they are interesting and creative.

Chapter 3

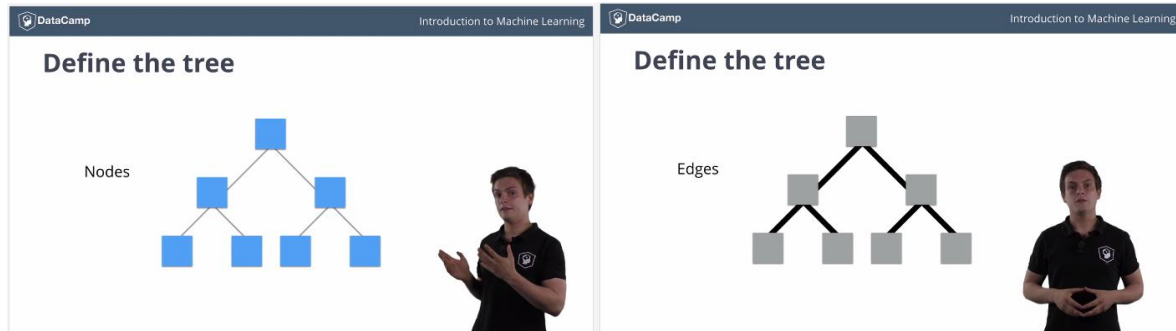
-Video1

Vincent looks a bit purple

Examples of features: Smokes? I think the question mark could be left out

The image is a screenshot of a video player showing a slide from a DataCamp course. The slide has a dark blue header with the DataCamp logo on the left and the text 'Introduction to Machine Learning' on the right. The main title of the slide is 'Examples of features' in a large, bold, white font. Below the title, there are two main bullet points, each with its own sub-bullets. The first main bullet point is 'Features can be **numerical**' and has two sub-bullets: 'age: 23, 25, 75, ...' and 'height: 175.3, 179.5, ...'. The second main bullet point is 'Features can be **categorical**' and has two sub-bullets: 'travel_class: first class, business class, coach class' and 'smokes?: yes, no'. In the bottom right corner of the slide, there is a small video inset showing a man in a dark polo shirt with a logo on the chest, gesturing with his hands. The overall background of the slide is white with a light blue border on the left and bottom.

Define a tree: I would change the slide - just use two different colors for nodes and edges and write the term in the same color



I think here the grey for edges doesn't really reflect on the edges. Maybe a slide with blue nodes and red edges would clarify it better (just an idea).

-Exercise

in ex10 the incorrect submission is a bit sloppy - I would also change the option orange into Black or green, since red and orange could be confused in this case

Interpreting a Voronoi diagram

35 XP

A cool way to visualise how 1-Nearest Neighbor works with two-dimensional features is the Voronoi Diagram. It's basically a plot of all the training instances, together with a set of tiles around the points. This tile represents the region of influence of each point. When you want to classify a new observation, it will receive the class of the tile in which the coordinates fall. Pretty cool, right?

In the plot on the right you can see training instances that belong to either the blue or the red class. Each instance has two features: x and y . The top left instance, for example, has an x value of around 0.05 and a y value of 0.9.

Suppose you are given an unseen observation with features $(x, y) = (0.5, 0.5)$. Looking at the Voronoi diagram, which class would you give this observation?

Possible Answers

- ☐ Blue
- ☐ Red
- ☐ It's impossible to tell!
- ☒ Orange

[Get Hint \(-15 XP\)](#) [Submit Answer](#)

✖ Incorrect submission

Orange is not a possible class! If you're colorblind or have trouble identifying the colors, the answer is 'blue' :)

[Hint](#)

Plots

1/1

R Console

[Click or Press Ctrl+2 to focus](#)

> |

Chapter 5

- Video1

Slides on cluster, why? could be better structured I think

 DataCamp

Introduction to Machine Learning

Clustering, why?

- Pattern Analysis
- Visualise Data
- pre-Processing Step
- Outlier Detection
- ...
- Targeted Marketing Programs
- Student Segmentations
- Data Mining
- ...



-Exercises

in ex2 hints: ?means doesnt exist

k-means: how well did you do earlier? 70 XP

Remember the `seeds` dataset from Chapter 2 which you clustered using the k-means method? Well, we've found the labels of the seeds. They are stored in the vector `seeds_type`; there were indeed three types of seeds!

While clusters are made without the use of true labels, if you happen to have them, it is simply interesting to see how well the clusters you made correspond to these true labels.

Up to you now: cluster the instances in `seeds` and compare the resulting clusters with `seeds_type`. Both objects are available in your workspace.

Instructions

- Group the `seeds` in three clusters using `kmeans()`. Set `nstart` to 20 to let R randomly select the centroids 20 times. Assign the result to `seeds_km`.
- Print out `seeds_km` to see what it contains.
- Compare the resulting clusters, in the `cluster` component of `seeds_km`, with `seeds_type` using the `table()` function.
- Plot the `width` as function of `length`, using `plot()`. Set `col` to `seeds_km$cluster`.

Hint

- Take a look at `library()`.
- The resulting clusters can be found in `seeds_km$cluster`.
- `table()` puts the first argument as rows and the second argument as columns.
- Setting `col = seeds_km$cluster` colors your points based on the corresponding object's cluster.

```
1 # seeds and seeds
2
3 # Set random seed
4 set.seed(100)
5
6 # Do k-means clus
7
8
9 # Print out seeds
10
11
12 # Compare cluster
13
14
15 # Plot the width
16
```

R Console

> |

- Video2: Performance and scaling issues

Gille appears to be a bit pink, maybe we can filter that

Exercises

in ex7 error in success_msg: substantially (typo)

It's always important to assess the scale of your variables prior to clustering. If your variables are on a different scale, it's best to either rescale some of them or ultimately standardize them all.

Which of the following datasets should best be fully standardized? Before you answer this question, you can take a look at the summary of each dataset. They are all available in the workspace.

Possible Answers

Exercise Completed

50 XP

- `small_result`, containing scores
- `week_data`, containing the distance to the start of the week and the first's period (in earth days)
- `run_records`, that contains the olympic run records for 50 countries for the 100m, 200m ... to the Marathon. The
- Indeed! The scaling between the run records differ **substantially**. Moreover, they all share the same unit, so standardizing will not make the interpretation of the clusters any harder!

Next Exercise

in ex12 hint: first hint includes one too many parenthesis, second hint does not makes sense.

the `border` argument to `"red"`.

Hint

- Click on `dist()`. It requires a dataset and a method argument.
- The `hclust()` function requires a distance matrix as first argument and a linkage method.
- `rect.hclust()` takes as first argument the cluster tree `run_single`. You can specify at which point to cut the tree, by specifying `k`. The remaining clusters will be boxed, the colour of these can be chosen by setting `border`.

R Console

```
> |
```

Leave class

in ex15 - Do you remember what Dunn's index measured => I would use present tense, also in the instructions

hint is missing

Interpreting Dunn's Index

35 XP

Do you remember what Dunn's index **measured**? Let's put that to the test!

Which of the following statements about your clusterings of the countries in `run_record_sc` is correct? Use the results of Dunn's index, found in `dunn_km`, `dunn_single` and `dunn_complete` in your workspace.

Possible Answers

- ☐ The complete-linkage method returned the lowest ratio of maximal intercluster-distance to minimal cluster diameter.
- ☐ Based on Dunn's index, the complete-linkage method returned the most compact and separated clusters.
- ☒ The single-linkage method returned the highest ratio of minimal intercluster-distance to cluster diameter.
- ☐ Based on Dunn's index, the single-linkage method returned the least compact and separated clusters.

Get Hint (-16 XP)

Submit Answer

Hint

Hint

success_msg: If you think[it] about, it can make sense

you remember what Dunn's index measured? Let's put that to the test!

Which of the following statements about your clusterings of the countries in `run_record_sc` is correct? Use the results of Dunn's index, found in `dunn_km`, `dunn_single` and `dunn_complete` in your workspace.

Possible Answers

Exercise Completed

50 XP

☐ The complete-linkage method returned the lowest ratio of maximal intercluster-distance to minimal cluster diameter.

☐ Correct! Are you satisfied with this result? The single-linkage method that caused chaining effects, actually returned the most compact and separated clusters ... If you think about, it

☒ Based on Dunn's index, the single-linkage method returned the least compact and separated clusters. The single-linkage method returned the highest ratio of minimal intercluster distance to maximal cluster diameter. The simple linkage method puts every outlier in its own cluster, increasing the intercluster distances and reducing the diameters, hence giving a higher Dunn's index.

☐ Based on Dunn's index, the single-linkage method returned the least compact and separated clusters.

Therefore, you could conclude that the single linkage method did a fine job identifying the

outliers. However, if you'd like to report your clusters to the local newspapers, then [Submit Answer](#)

complete linkage or k-means are probably the better choice!

What

Next Exercise

What

Press Enter

However, if you'd like to report your clusters to the local newspapers, then complete linkage or k-means are probably the better choice!

==> why? Maybe give a short reasoning. Or am I missing something?

Chapter 4

Video1 – really good in my opinion!

Ex4: I think it either R-squared value or Rsquare

$$SS_{res} = \sum_{i=1}^n res_i^2$$

whereas $SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$

Exercise Completed

0/100 XP

Great! Apart from rounding, the **Multiple R-squared** is exactly the same as the `r_sq` you calculated. An R-squared of 0.77 is pretty neat!

Next Exercise

Press Enter

Instructions

- Calculate SS_{res} , assign it to `ss_res`.
- Determine SS_{tot} , assign it to `ss_tot`.
- Use SS_{res} and SS_{tot} to calculate R^2 . Assign your outcome to `r_sq` and print it.
- Call `summary()` to generate a summary of `lm_kang`. Take a look at the outcome *multiple R-squared*.

Ex7, typo

Interpreting R-squared

50 XP

In your last exercise, you transformed your predictor wonderfully. Your prediction differed **substantially** between your two models, so which one should we take? To answer this, you can have a look at the R^2 values of both attempts. The first linear model you built, that did not use a transformation, is available as `lm_yb`. The second model, where `log()` was used, is pre-loaded as `lm_yb_log`.

Which of the following statements is true?

Possible Answers

- ☐ 1 `lm_yb` returns the best prediction, its explained variance is the lowest.
- ☒ 2 `lm_yb_log` returns the best prediction, its explained variance is the highest.
- ☐ 3 `lm_yb` returns the best prediction, its explained variance is the highest.
- ☐ 4 `lm_yb_log` returns the best prediction, its explained variance is the lowest.

Get Hint (-15 XP) **Submit Answer**

mpts. The first linear model you built, that did not use a transformation, is available as `lm_shop`. The second model, where `log()` was used, is pre-loaded as `lm_shop_log`.

Which of the following statements is true?

Exercise Completed

50 XP

Great! The second model clearly is better to predict the percentage of urban population based on the GDP per capita. The type of the second model is called *linear-log*, as you take the logarithm of your predictor variable, but leave your response variable unchanged. Overall your model could still have quite a lot of unexplained variance. If you want more precise predictions, you'll have to add other relevant variables in a *multivariable linear model*. Check out the next video to learn how!

Incorrect submission

Next Exercise

Press Enter

I think the word order can be improved here. Is it linear-log or log-linear or does it matter at all?

Ex10

There is a shop owner that didn't participate in the questionnaire, who has caught wind of your amazing model. He asked us to provide a confidence interval for his expected sales. Can you help him?

Exercise Completed

100 XP

Well done! There is no clear pattern in your residuals. Moreover, the residual quantiles are approximately on one line. From the small p-values you can conclude that every predictor is important. The shop owner is now 95% certain his expected sales will be higher than 249.385\$ and lower than 275.616\$, he expresses his gratitude!

The model you produced is still not finetuned, if you're more familiar with correlation, you might have noticed potential issues with the multicollinearity of the predictors. Don't worry, you'll address this in a next course on multivariable regression!

Next Exercise

Press Enter

```

3 # Plot the residuals
4 plot(lm_shop)
5 xlab = "Residuals"
6
7 # Make a qqnorm
8 qqnorm(lm_shop$residuals)
9
10 # Summarize the model
11 summary(lm_shop)
12
13 # Predict the confidence interval
14 predict(lm_shop, newdata = data.frame(gdp_per_capita = 262.5, urban_population = 0.249385))

```

R Console

```

---
Signif. codes:  0 ' '

Residual standard error: 1.2625086
Multiple R-squared:  0.6116
F-statistic: 611.6
> # Predict the net
> predict(lm_shop, newdata = data.frame(gdp_per_capita = 262.5, urban_population = 0.249385))
[1] 262.5086 249.385

```

Maybe it sounds better to say: potential issues with multicollinearity among the predictors...

And just to be sure: is it multivariable or multivariate – or can both words be used?! I thought they address two different things, no? I am not sure.

Ex15

In my opinion it would sound better to write “The output consisten of...”

Your own k-NN algorithm!

100 XP

In the video, Gilles shortly showed you how to set up your own k-NN algorithm. Now it's your turn to program one in R and test it out on the world bank example.

We went ahead and defined a function `my_knn`, that will contain your k-NN algorithm. As input it takes a vector with new observations for the predictor (`x_pred`), the training set values of your predictor (`x`), the corresponding response values (`y`) and the number of neighbours (`k`). The output are the predicted values for your new observations (`predict_knn`).

Your job is to finish the function by adding three steps to the for-loop, then apply your algorithm to the GDP / capita of the countries in `world_bank_test` to predict their percentage of urban population.

Instructions

my_script.R

```
1 # wc
2
3 # Tr
4 my_k
5
6 m
7
8 pr
9 fc
10
11
12
13
14
15
16
17
18
19
20 }
21 re
22 }
23
```

R Console