Guide for Module 1.2 exercises

Data Preparation, Summarization and Visualization

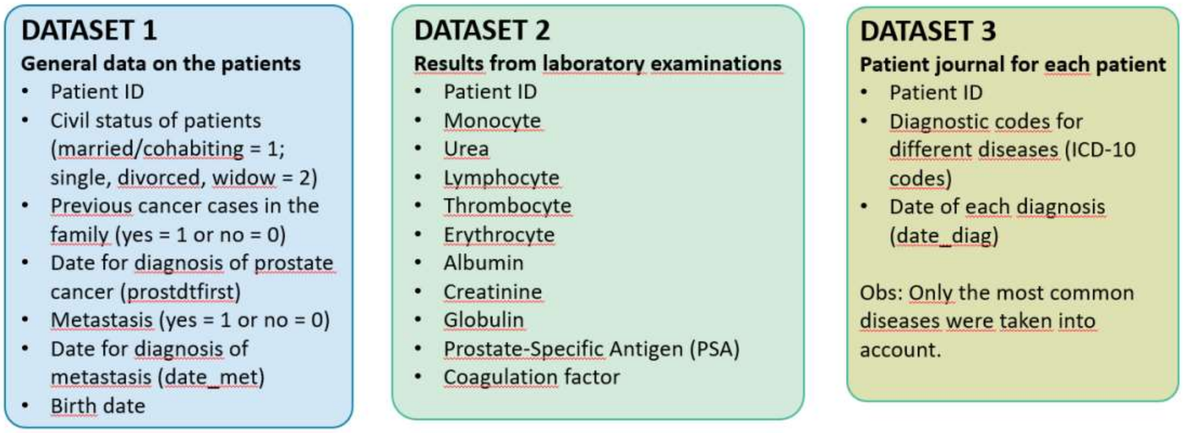
Study description:

Urologists from the Urology department at Odense University Hospital have selected 1159 patients that were diagnosed with prostate cancer (PC) at their first visit with them.

Some of those patients were also diagnosed with metastasis (metastasis = 1) and others do not present metastasis (metastasis = 0). Metastatic cases were detected by the appropriate method (e.g. CT scans, MRI).

The urologists have requested a bunch of laboratory examinations from these patients (measured in blood and urine), as well as retrieved their medical history (i.e. patient journals).

They got in contact with us, the AI team, to help them to identify ways of predicting which patients will present metastasis and which will not. This could be further used as a prediction tool in an earlier stage, and would also avoid overtreatment and invasive examinations.



DATASET 1

General data on the patients

- Patient ID
- Civil status of patients (married/cohabiting = 1; single, divorced, widow = 2)
- Previous cancer cases in the family (yes = 1 or no = 0)
- Date for diagnosis of prostate cancer (prostdtfirst)
- Metastasis (yes = 1 or no = 0)
- Date for diagnosis of metastasis (date_met)
- Birth date

DATASET 2

Results from laboratory examinations

- Patient ID
- Monocyte
- Urea
- Lymphocyte
- Thrombocyte
- Erythrocyte
- Albumin
- Creatinine
- Globulin
- Prostate-Specific Antigen (PSA)
- Coagulation factor

DATASET 3

Patient journal for each patient

- Patient ID
- Diagnostic codes for different diseases (ICD-10 codes)
- Date of each diagnosis (date_diag)

Obs: Only the most common diseases were taken into account.

Can PSA (Prostate-Specific Antigen) be used as a biomarker to detect metastatic prostate cancer?

Previous studies say the following:

“Based on results from some small, single centre studies published in the beginning of the 1990s, PSA levels above 100 ng/mL have been used as a proxy for metastatic prostate cancer” Thomsen et al., 2020

“PSA < 20 ng/ml have high predictive value in ruling out skeletal metastasis” Kamaleshwaran et al., 2012

TASK 1 – Measures of test performance based on PSA (30 minutes)

Based on the data provided and the context information, let’s make a confusion matrix for each PSA cutoff (PSA ≥ 20 ng/ml and PSA ≥ 100 ng/ml) and calculate the following:

	Prostate cancer metastasis PSA > 20 ng/ml	Prostate cancer metastasis PSA > 100 ng/ml
Sensitivity		
Specificity		
Accuracy		

Tips:

1) For that task, you will only need few variables: metastasis (binary variable, 1 or 0) and PSA levels.

2) You will need to create a variable indicating whether the test is positive or negative for metastasis (e.g. test_result):

```
data$test_result <- ifelse(data$psa > 100, 1,0)
```

3) You can calculate the amount of trust positives, trust negatives, false positives and false negatives, and from there calculate the measures of test performance: e.g in R:

```
data$TP <- ifelse((data$Metastasis == 1 & data$test_result == 1), 1, 0)

TP <- sum(data$TP)
```

TASK 2 – Discuss the results (15 minutes)

Based on the results you found, discuss in small groups the following points:

1) When looking at the overall results for both tests (using PSA cutoff of 20 or 100 ng/ml), what can you conclude when comparing the results obtained for sensitivity and specificity?

2) At the moment, what would you say is the best cutoff to predict PC-related metastasis (20 or 100 ng/ml)? What would be the advantages for the healthcare system when using each of the tests?

TASK 3 – Using the ROC curve to find the optimal cutoff (30 minutes)

Using the data from the PC patients:

- Find the optimal cutoff point when using PSA to classify patients into those with and without metastasis, based on the Youden index method.

- Discuss in groups: would you say that the Youden index method is most appropriate approach? Try at least another method to optimize the cutoff point.

TIP for this exercise:

If you are using R, I suggest the package OptimalCutpoints*

- https://www.rdocumentation.org/packages/OptimalCutpoints/versions/1.1- 4/topics/optimal.cutpoints

- https://cran.r-project.org/web/packages/OptimalCutpoints/OptimalCutpoints.pdf

- Paper: OptimalCutpoints: An R Package for Selecting Optimal Cutpoints in Diagnostic Tests

- As an example for the code using the OptimalCutpoints package:

```
cutpoint.Youden <- optimal.cutpoints(X = "psa", status = "metastasis", tag.healthy = 0, methods = "Youden", data = data, pop.prev = NULL, control = control.cutpoints(), ci.fit = FALSE, conf.level = 0.95, trace = FALSE)
```

```
summary(cutpoint.Youden)
```

```
plot(cutpoint.Youden)
```

- To try different methods to find the optimal cutpoint, you can just play around with the optimal.cutpoints function. The suggested literature can help you here.

TASK 4 – Principal Component Analysis (45 minutes)

Now, let’s use Principal Component Analysis to extract a new set of variables (i.e. Principal Components - PCs) considering 10 biomarkers (starting with monocyte and ending with coag_factor). In R, one option is to use the prcomp function, e.g.:

```
pca.model <- prcomp(df[x:y], center = TRUE, scale. = TRUE)
```

```
summary(pca.model)
```

- Observe the proportion of variance explained by each PC (in R, this can be seen with the summary() function). Make a plot with the results, so that it is easier to visualize. How many PCs would you consider sufficient?

- Now, select the final number of PCs that you would like to keep.

In R, I suggest you to base this decision on the Eigen values. To calculate Eigen values, you can use pca.model$sdev^2

- Inspect the components (i.e. loadings) for each PC. For that, you can use print(pca.model$rotation). Discuss in groups (and interpret) the loadings for each of the PCs you decided to keep. Remember that large loadings (either positive or negative) indicate that a variable has a strong effect on the principal component.

- Generate a new dataset of the number of principal components you decided to keep.

To extract the principal components in R, you can use:

```
components <- pca.model$x[, 1:x]
```

<where x in the number of principal components you decided to keep.>

To combine the principal components to your original dataset in R, you can use the following dplyr function:

```
df <- bind_cols(df, components)
```

To export the combined dataset you can, for instance, use the command:

```
write.csv(df, “your_file_name.csv”, row.names = FALSE)
```

<observation: here the csv file will be saved on your working directory unless you specified otherwise. If you have no working directory, you can write the file path in the command:

```
write.csv(df, “file path/your_file_name.csv”, row.names = FALSE)>
```

For the report, make sure to include (at least) the following:

- Include the results you obtained in TASK 1 (optional: to include the confusion matrix for each cutoff) and discuss what you found and what you have discussed in TASK 2.
- Include the results you found for the optimal cutoff and the ROC curve you obtained in TASK 3 and discuss what you have found and what you have discussed in the classroom.
- Report the number of PCs you decided to keep in TASK 4 and justify your decision.
- Briefly describe the main loadings for each principal component in TASK 4. Can you see any pattern here?

Remember to always save the script!