Introduction to Data Science
WS24/25 Assignment Part 1

Prof. Dr. Wil van der Aalst, Nina Graves,
Lukas Liss, Christian Rennert, Christopher Schwanen,
Leah Tacke genannt Unterberg

Chair of Process and Data Science
RWTH Aachen University

November 4, 2024

Introduction

You want to practice your data science skills using a dataset describing patients with heart diseases. You start by exploring the overall dataset before creating and evaluating different classification models. Finally, you want to cluster the patients to identify different patient groups.

In your attempt to build a good classifier, you start with explainable models such as decision trees and logistic regression. To get a better understanding of the "black box" Classifiers, you do a deep-dive into Support Vector Machines and Neural Networks. Your Support Vector Machine (SVM) classifier is a binary classifier (*healthy heart* vs. *heart condition*), while all other classifiers are multi-class models for the severity of the condition of a patient's heart.

The Data

The dataset contains a variety of patients' features, including information about whether they have heart disease and its severity. Each patient recorded in the data set has the following features:

Column Name	Description
id	Unique patient identifier
age	Age of the patient in years
origin	Location of the hospital collecting the data
sex	Sex of the patient
cp	Chest pain type
trestbps	Resting blood pressure (in mmHg on admission)
chol	Serum cholesterol (in mg/dl)
fbs	True if fasting blood sugar $> 120\,\text{mg/dl}$
restecg	Resting electrocardiographic results
thalch	Maximum heart rate achieved in stress test
exang	Exercise-induced angina (True/False)
oldpeak	ST depression (a numerical medical test value) induced by exercise relative to rest
slope	The slope of the peak exercise ST segment
ca	Number of major vessels (0 to 3) colored by fluoroscopy
thal	Thalassemia (normal/fixed defect/reversible defect)
num	Target feature: Severity of the heart disease (0 to 4; with 0 = no heart disease, 4 = critical heart disease)

The raw data is stored in the `heart_disease_raw.csv`. The preprocessed and one-hot encoded data, split into training (`train_dataset.csv`) and test (`test_dataset.csv`) data is also provided. All the data files can be found in the provided zip in the data folder.

Assignment Details

- Total number of points obtainable: 100
- Group size: 2 to 3 students
- Input:
  - this assignment PDF,
  - Jupyter notebook templates (one per question),
  - Data: Raw data (`heart_disease_raw.csv`), preprocessed training and test data (`train_dataset.csv`, `test_dataset.csv`)
- Deadline: 02.12.2024, 23:59:59 (CET)
- Deliverables (to be uploaded as zip file in Moodle):
  - PDF report (max. 30 pages)
  - Jupyter notebooks (one per question)
  - If you used LLMs: Provide a document detailing your use of LLMs in accordance with the Study Guide.

Report Your written report is the main basis for grading. In your report, you should present your methods, motivation, results, and explanations. Doing so, **clearly indicate which answer belongs to which question.** Please order your questions according to the numbering of this assignment to avoid confusion. Moreover, it should be **self-contained** (i.e., it should not require references to the notebook). The length should be at **most 30 pages**, including the title page. Make sure to include **all group members' names on the title page**. Please respect the following reporting criteria (Severe problems may lead to a point deduction of up to 10 points):

- Proper spelling, punctuation, readability, and comprehensive structure
- Use of **adequate** visualizations for showing (aggregated) results or illustrating methods
- Figures have captions, axes have labels, diagrams have headers
- Figure quality (e.g., resolution and relevance)
- All figures, tables, and similar are numbered and referred to in the text
- All your comments, descriptions, discussions and interpretations should be explicit and concise. Usually, no more than 2-3 sentences are needed to answer individual parts of the question.

Notebooks Your Jupyter notebooks will be used for reference and potentially for testing your code. Therefore, it should satisfy the following requirements:

- Commented and structured code (if not, this will be penalized in the style points)
- Questions separated by markdown headers
- Top-to-bottom runnable cells to reproduce your results
- DO NOT CLEAR THE OUTPUT of the notebooks you are submitting!
- It should be runnable in the bundled conda environment.
- Ensure that the code in the notebook runs if placed in the same folder as all of the provided files, delivering the same outputs as the ones you submit in the notebook and report on in your report.

Do not re-upload the data. Besides, notebooks that intentionally access files outside of the notebook's directory or are in any way harmful will be graded with zero points.

Hints

When answering the questions, document what you did and carefully describe and explain your results. In particular, explain how you derived your results, on which facts you base your claims, and motivate the methods you used. Results from previous questions can (and should) be referred to, to improve your discussion and explanation.
The template notebooks already contain many useful imports and a few examples of helpful code snippets that have not been explicitly part of the official documentation when publishing the assignment.

Optional Resources

- Jupyter: `https://jupyter.org/index.html`
- Jupyter Lab/Notebook installation guide on Moodle

Question:	1	2	3	4	5	Total
Points:	24	11	8	42	15	100

Question 1: Data Exploration

(a) (2 points) Load the `heart_disease_raw.csv` as a *pandas* dataframe and state the number of rows and columns.

First, you want to get an overview by investigating some basic statistics.

(b) (2 points) Use the `describe()` method of the *pandas* dataframe to get an overview of the basic statistics. Provide a screenshot of the resulting table showing the basic statistics resulting from the `describe()` method. Note that the `describe()` method only gives statistics for a subset of columns. Explain why there are no basic statistics computed for the other features.

(c) (1 point) Your analysis reveals that the mean of the *num* column (which is the target feature) is about 0.996. State and explain in maximally two sentences whether this metric is well suited to describe the *num* column or not.

Next, you investigate individual features further by using the visualizations you have learned.

(d) (2 points) Give the number of female and male patients in the dataset. Assume that you want to balance out their representation. Name two sampling approaches that were introduced in the lecture that you could use to reach your goal.

Note: You do not need to implement or apply any of the sampling methods.

(e) (1 point) Provide a visualization of your choice that is able to show the distribution of values of the *num* feature. Also, state the mode of the *num* feature.

The *thalch* feature describes the maximum archived heart rate during a stress test. To understand its distribution better, you want to create a histogram using equal width binning.

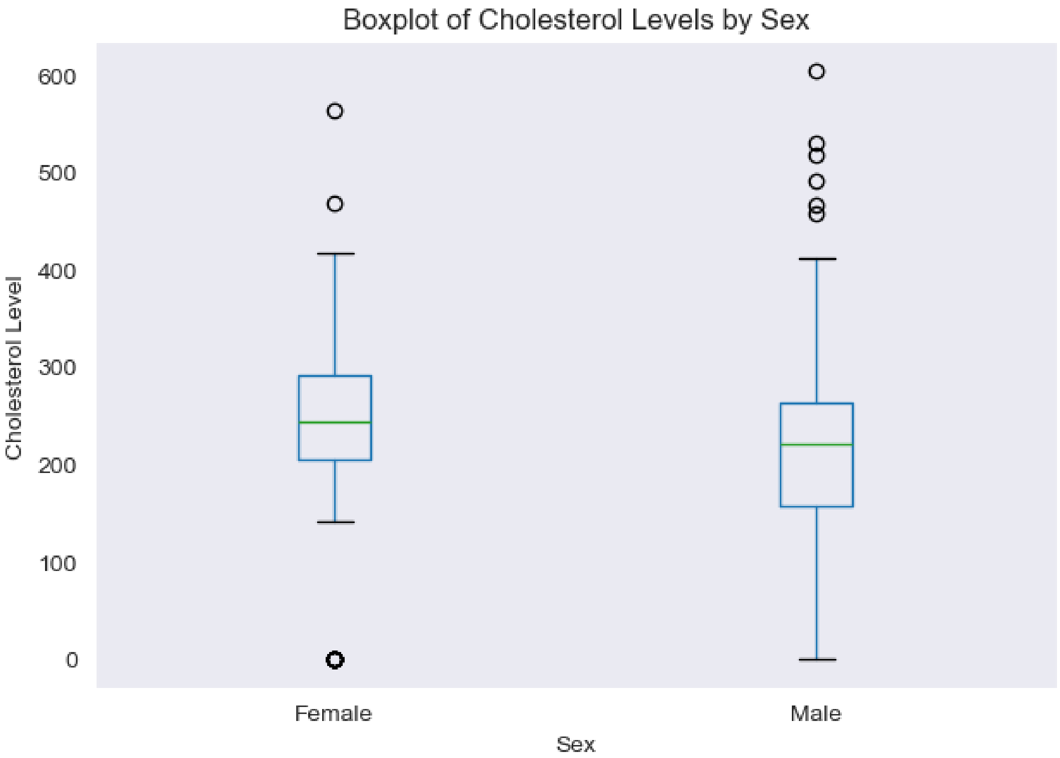
(f) (3 points) Create and provide a screenshot of a histogram of the *thalch* feature using binning. Choose a good number of bins and state the number of bins you used. Explain why your number of bins is good for the given data. Also, name the type of the resulting histogram using the types of histograms that were introduced in the lecture.

Next, you investigate the relations between the features to understand which features have relations that you can later use, for example, for predictions.

(g) (2 points) Create a scatter plot matrix for the columns *age*, *trestbps*, *chol*, and *thalch* and provide a screenshot of the resulting scatterplot matrix.

(h) (2 points) Compute the correlations for all the feature pairs from *age*, *trestbps*, *chol*, and *thalch* and provide a screenshot or table showing these correlations. State the strongest absolute correlation for the given feature pairs and explain what this type of correlation means.

(i) (4 points) In the following, you see boxplots (as introduced in the lecture) of the cholesterol levels grouped by the sex of the patient and questions about the boxplots. Answer each of the following questions or state why the question cannot be answered.



- Do males or females have a higher average cholesterol level?
- Are there outliers for the male group that have values below the lower fence value?
- Are there more outliers for the male or female group?
- Is the median of females lower than the upper end of the 3rd quartile of males?

Now that you have gained a first understanding of the features of the dataset, you decide to preprocess the dataset for further analysis.

(j) (4 points) For each column, provide the number of missing values (in *pandas* they are represented as `NaN` values) and state the feature that has the most missing values. Based on the number of rows of the dataset and the information about the `NaN` values, provide a lower and upper bound for how many rows contain at least one `NaN` value. Explain your upper and lower bound.

(k) (1 point) Create a dataframe that contains only rows where all features are present, so only rows without any `NaN` values. Provide the row count of this dataframe.

Question 2: Decision Trees

(a) (2 points) Create a baseline prediction model by predicting always the mode of the *num* feature from the `train_dataset.csv`. Then evaluate the baseline using the `test_dataset.csv`. Compute and report the accuracy of the baseline.

(b) (6 points) You want to predict the *num* feature using a decision tree. Use the `tree.DecisionTreeClassifier` algorithm from the *scikit-learn* library and set the criterion as “entropy”, `min_samples_leaf=6`, and `random_state=42`.

You are interested in finding out what the best parameter for the `max_depth` of the tree is to minimize the error. Therefore, you decide to test out different parameters.

Create a decision tree for each integer from 1 up to and including 10 as the `max_depth`. Compute the accuracy for each decision tree, using the `test_dataset.csv`.

Create a summarizing plot, in which the x-axis represents the depth of the tree and the y-axis the accuracy. Provide this plot and briefly state which values you would choose for the `max_depth` and explain your choice (up to two sentences). Also, provide the highest accuracy found.

(c) (3 points) Create and visualize a decision tree with the following parameters (criterion “entropy”, `min_samples_leaf=6`, `random_state=42`, and `max_depth=3`) from the `test_dataset.csv` and provide the visualized tree. State the first decision criterion and briefly (up to two sentences) explain what it means.

Question 3: Regression

(a) (2 points) Create a logistic regression model that predicts the *num* feature based on the *thalch* and *age* features. Since *num* is not binary use the `multiclass='ovr'` parameter for the logistic regression from *scikit-learn.* Train the logistic regression model using the `train_dataset.csv`. Provide the weights and intercepts.

(b) (3 points) Use `test_dataset.csv` to evaluate your created logistic regression model. Compute and provide the accuracy. Reason whether the accuracy is better or worse than random guessing and state the expected accuracy of random guessing.

(c) (3 points) Provide a graph that shows the age over the *thalch* value of the `train_dataset.csv` and color the dots based on their *num* value. Use that plot to explain why the performance of the logistic regression model is not that good. Provide two meaningful ways you would try out to improve the result of the logistic regression model.

Question 4: Support Vector Machines and Neural Networks

You decide you want to train a binary SVM classifier to predict whether a person has a heart condition or not, referred to as *binary classification.* In contrast, you want to use a Neural Network to directly predict a patient's heart condition, referred to as *severity classification* (the original target value *num*).

Prepare and Explore the Data

Before you can start training the models, you need to load the data, prepare the two different target values for the two different classification problems and separate the descriptive features from the target variable. Hence, you must do the following (find further guidance in your Python notebook):

- Load the training data `train_dataset.csv` and test data `test_dataset.csv`.
- Create a new version of each dataset for the binary classification: *healthy heart* for all instances with no heart disease (i.e., the value of *num* is 0) and *heart condition* otherwise (i.e., $value \geq 1$). We consider a healthy heart to be positive and a heart condition to be negative.
- Create the feature matrices X for the training and test data—make sure neither the original nor any transformed target values are included. You can use the same matrices for training both types of classifiers.
- Create the target vectors y for the training and test data for each of the classification problems.

Now you want to explore the data to get a better understanding of the distribution of the instances and the features.

(a) (2 points) You first consider the severity classification data. Provide a bar chart showing the frequency of the different classes in both the test and the training data. What do you observe? How does the distribution of the training data compare to the test data?

(b) (3 points) You want to explore the feature space for the binary classification to get a better understanding of the distribution of the instances. As there are many features, most of which are one-hot encoded categorical features, you focus on the five features with continuous numeric values: *age*, *trestbps*, *chol*, *thalch*, and *oldpeak*.

As you cannot visualize more than three dimensions, you just look at different combinations of three and two features in the training data—you create the possible 3D plots and a pairplot for the five features. Consider the data closely and select three features you think show a space in which the two classes could be separated reasonably well using a linear SVM classifier. Provide the following:

- A scatter plot matrix (aka pairplot) showing the distribution of the five numeric features in the training dataset (each chart showing either two features or a histogram). Make sure to visually distinguish between the two classes by using different colors and different markers.
- A 3D scatter plot showing the distribution of the three features that you think depict can be separated well using a linear decision boundary. Use different markers and colors for the two classes.
- Explain why you think the data of these features are well suited for a linear SVM classifier.

Hint: Multiple selections are possible.

Model Configurations

(c) (4 points) You build an initial SVM binary classifier using only the three features you selected in the previous task. Create a smaller descriptive feature matrix X for the training and test data, containing only the three selected features. Then train a linear SVM classifier with a regularization parameter of $C = 10$ on the training data and predict the target feature of the test instances. State and briefly interpret your model's accuracy and precision.

(d) (3.5 points) You want to investigate by how much you can improve this SVM classifier by using the same training data as before (only your 3 selected features) but considering different model configurations. Use a grid search to find the best performing configuration (w.r.t. accuracy). Consider the following options in your 5-fold cross-validation:

- Kernel function: linear, polynomial (degree of 3; default in *scikit-learn*), radial basis function (RBF)
- Regularization parameter: 0.1, 10, 50, 100, 500

Provide the following:

- The best performing configuration of the $3 \times 5 = 15$ options.
- A plot describing the results of the grid search. Show the regularization parameters on the x-axis and the accuracy on the y-axis with the different kernel functions represented with different colors and markers.
- Briefly comment on the linear kernel's performance compared to the other kernel functions at different regularization parameters.

Hint: The grid search might take a few minutes. You can speed this up by taking a look at the `n_jobs` attribute of the `GridSearch` object.

(e) (3.5 points) For each of the three kernel functions: Train a model on the training data using the three selected features and the best performing regularization parameter. Then predict the target feature of the test instances using each of the trained models. Provide the accuracy and precision of each of the models. Comment on the robustness of the best configuration's performance.

Baseline Models

(f) (4.5 points) You previously used three features to get an impression of the impact the different kernel functions and regularization parameters have. Now you want to train a baseline SVM classifier using all features, using a linear kernel and a regularization parameter of $C = 10$. Evaluate the model's performance on the test data:

- Provide and interpret the classifier's accuracy, precision and confusion matrix for predictions on the test data. Consider the domain of your data.
- Is the model balanced or does it favor one class over the other?
- What percentage of patients classified as healthy actually have a heart disease?

(g) (4.5 points) Train a Multi-Layer Perceptron (MLP) using all descriptive features and the targets for severity classification. Use the trained model to predict the target feature of the test instances. Use the following configuration for this baseline Neural Network:

- Hidden layers: 5 hidden layers with 5 neurons each
- Activation function: Logistic
- Maximum of iterations: 5000

Report on:

- The accuracy and confusion matrix of the Neural Network on the test data.
- What strikes you about the confusion matrix? How can you explain this phenomenon based on your previous analysis of the data?
- How high is the percentage of diseased hearts that were classified as healthy?

Feature Engineering

In your lectures you learned that feature engineering is a crucial part of the machine learning process and of particular importance for SVMs and Neural Networks. You learned about different techniques to transform the data to improve the performance of the model, so now you want to apply and compare different techniques.

(h) (6 points) To be more flexible in the application of the transformations, you decide to create three functions in your notebook, allowing for the following transformations of training and test data to specified features. For every transformation you consider, you use the training data to determine the relevant parameters for the transformation (i.e., calculate a measure or fit a transformation object). Then transform both the training data and the test data using the parameters determined from the training data. You want to consider the following transformations to the dataset:

- Scaling: Scale the numeric features to have a mean of 0 and a standard deviation of 1, e.g., using *scikit-learn*'s `StandardScaler`.
- Normalizing: Normalize the numeric features to have a minimum of 0 and a maximum of 1 in accordance with the minimum and maximum values of the training dataset, e.g., using *scikit-learn*'s `MinMaxScaler`.
- Deviation Mean: For each feature, determine the mean of the training data. Then subtract the mean from each instance in the training and test data.

You now use these functions to create three new train and test datasets, in each of which the descriptive features are transformed accordingly.

- Based on what you learned in the course, briefly explain the general idea behind feature engineering.
- For each of the transformed datasets: create a scatter plot matrix (aka pairplot) of the five numeric features in the training dataset (each chart showing either two features or a histogram). Depict the two classes in the pairplot.
- Consider the pairplots showing the original (previous task), scaled, and normalized data to explain the commonalities and differences between these transformations. How do these differences affect SVM classifiers? Why do Neural Networks and SVMs benefit from scaling and normalization more than decision trees?

Hint: You can complete the placeholder functions in the provided jupyter notebook.

Network Structure

(i) (2 points) You know that a more complex structure of a Neural Network allows for more differentiated results. Hence, you decide to increase the number of hidden layers and neurons per layer in your Network. You also use the scaled training and test datasets for the severity classification. Create an MLP Classifier for severity prediction with a higher number of layers and neurons that predicts every class in the scaled *training data* at least once.

- Provide the confusion matrix for this classifier on the training data.
- How many hidden layers and neurons per layer did you use?
- How high is this model's accuracy on the training data?

(j) (2 points) Now use this model to predict the severity of the heart conditions on the scaled test instances.

- Provide the confusion matrix for this classifier on the test data.
- How high is the model's accuracy on the test data?
- How do you explain the difference in performance between the training and test data?

A Good Classifier

You decide that the dataset is not suitable for training a Neural Network, so you focus your efforts on the binary SVM classifier. Apart from accurately predicting the presence of a heart disease, you want to make sure the number of people whose heart disease cannot be detected is low. With this in mind, you consider the different aspects that influence the performance of models you have looked into so far:

- different model configurations,
- included features, and
- feature engineering.

(k) (6 points) Your goal is to create a model with a higher accuracy and precision than the baseline classifier and a smaller number of misclassified heart conditions on the test data. You are free to exclude features or make any transformations (or transformation combinations) to individual features you think are reasonable or want to try out. You may also improve your model by testing different configurations—but keep in mind that you should not overfit the model to the training data. Briefly report on each of the following:

- Provide the model configuration (kernel function and regularization parameter) of your classifier.
- Describe the input data you used for the prediction: What features were used, how were they transformed?
- Provide the confusion matrix, accuracy and precision for both the training data and the test data.
- Briefly describe your approach to finding a good SVM classifier (one sentence each is sufficient):
  - How many classifiers did you train?
  - How did you decide on the parameters you used (e.g., by using a grid search and cross validation)?
  - How did you decide which features and transformations to use? In case you had interesting findings, please provide them (e.g., which features were most important, which transformations worked best).
  - How did you evaluate the intermediary classifiers you trained metrics, visualizations, etc., no need to explain how you decided on which model to settle)?
- Discuss the performance of your classifier (compared to the baseline SVM) and its suitability for the use case.

Hint: This question is more about you getting some hands on experience with the different aspects of model training and evaluation and working in an informed and structured manner. We want you to demonstrate that you understand what you are doing and that you can interpret your than about finding the perfect model and that you can work in a structured way. Hence, this question is not about a correct or incorrect *answer but for giving you some guidance in terms of structure and you reporting on your work.*

(l) (1 point) What would you have to change about the input data, so you could provide a better model for this use case?

Total for Question 4: 42

Question 5: Clustering

In the following, you want to use clustering to determine cohorts according to the numerical features in the dataset.

(a) (2 points) Explain which two of the original eight numerical features (based on `heart_disease_raw.csv`) you definitely do not want to use in your clustering approach.

(b) (6 points) Load the training dataset `train_dataset.csv` and apply k-means clustering to identify 3 cohorts based on the features *age*, *trestbps*, *chol*, *thalch*, and *oldpeak* with parameters `init='k-means++'` and `random_state=42`. In order to account for the different natures of the selected features, apply standard scaling (using *scikit-learn*'s `StandardScaler`) before clustering.

Describe the cohorts you identified by providing their size and specifying an "average" representative (i.e., centroid) with respect to the selected features.

(c) (3 points) Plot a graph that shows the resulting clusters for each pair of features (pairplot) and provide it in your report.

In your plot, you should be able to see that the feature *chol* has quite a particular influence on the clustering (because of a frequent particular value). Shortly describe this influence and suggest a way how you could reduce this particular influence.

(d) (4 points) Create a prediction model based on the identified clusters. For any instance, this model should always predict the mode of the *num* feature of the closest centroid's cluster. Provide the confusion matrix and compute the accuracy based on the test dataset `test_dataset.csv`.

Shortly discuss why this is not a suitable approach for prediction.