Introduction to Data Science

WS25/26 Assignment Part 1

November 5, 2025

Prof. Dr. Wil van der Aalst
A. Küsters, N. Graves, L. Liss, C. Pitsch, C. Rennert, C. Schwanen

Chair of Process and Data Science
RWTH Aachen University

Introduction

In this assignment, you will deal with a medical insurance cost data set containing medical information of individuals from the United States. Throughout the assignment, you will familiarize yourself with the application of basic Python libraries for data loading, preprocessing, visualization, the computation of decision trees and clusters, and the training of support vector machines and neural networks. You can carry out the Python part of your analysis in a Python notebook. A template has already been set up for you. In addition, you will have to deliver a written report explaining your methods and results.

The Data

The dataset consists of the following attributes:

Column Name	Description
ID	The ID of the patient must not be changed, as this feature is artificially added by the IDS team and enumerates the individuals. Using this ID, it allows for reproducibility of your results for us
Age	Age of the insured individual
Sex	The sex of the individual
bmi	The body mass index. A measure that approximates whether individuals are underweight (below 18.5), normal weight (18.5 to 24.9), overweight (25 to 29.9), or obese (30 and above)
children	The number of children that the individual has
smoker	Whether the individual smokes
region	The region of the U.S. that the individual comes from, i.e., northeast, northwest, southeast, or southwest of the U.S.
charges	The medical insurance cost of the individual
chargesGroup	Added by the IDS team for the one hot encoded train and test data. Values: low (charges <8000), medium (8000 ≤ charges <32000), or high (charges ≥ 32000)

The data is stored in the `insurance.csv` and `insurance_orig.csv` (where `insurance.csv` is a data set that is already removed from invalid entries). For convenience, we also provide a one-hot encoded data set (`insurance_one_hot.csv`) and the train/test split (`train_insurace.csv`, `test_insurance.csv`, `train_insurance_one_hot.csv`, `test_insurance_one_hot.csv`). Use these files whenever the task says so. All the data files can be found in the provided zip in the data folder.

Assignment Details

- Total number of points obtainable: 100 (20 % of final course grade)
- Group size: 2–3
- Input:
  - JupyterLab template, and
  - the data as described above
- Deadline: 03.12.2025, 23:59:59 (CET)
- Deliverables:
  - PDF report (max. 20 pages)
  - Jupyter notebooks (one per question)
  - If you used LLMs, provide a document detailing your use of LLMs.
  - If group problems arise that require our involvement, please let us know by 26.11.2025 at the latest.

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 20 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
- The precision of all decimal numbers should be up to two decimal digits
- Figures have captions, axes have labels, and 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 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 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 what motivated 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	6	Total
Points:	21	24	14	12	15	14	100

Question 1: Data Exploration (21 points)

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

First, you 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.

The count of features from the last subtask shows deviating numbers. This indicates missing instances.

(c) (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.

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

(d) (3 points) In the medical domain, the BMI is distinguished into classes: underweight - $[0, 18.5)$, normal weight - $[18.5, 25)$, overweight - $[25, 30)$, and obese - $[30, \infty)$. Introduce a new feature *bmi_class* based on the ranges of the BMI. Give the number of patients who are in the different body mass index (BMI) classes in the dataset. Discuss whether you would like or not like to balance out the dataset. Name two sampling approaches that were introduced in the lecture that could be used in this case.

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

(e) (1 point) Provide a visualization of your choice that can show the distribution of values of the *children* feature. Furthermore, state the mode of the *children* feature.

The *charges* feature describes the medical insurance costs of the patient. To understand its distribution better, you want to create a histogram using equal-width binning.

(f) (3 points) Do the following:

- Create and provide a screenshot of a histogram of the *charges* feature using binning.
- Improve the expressiveness of your histogram by using the *bmi_class* as hue parameter in the histogram.
- 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.

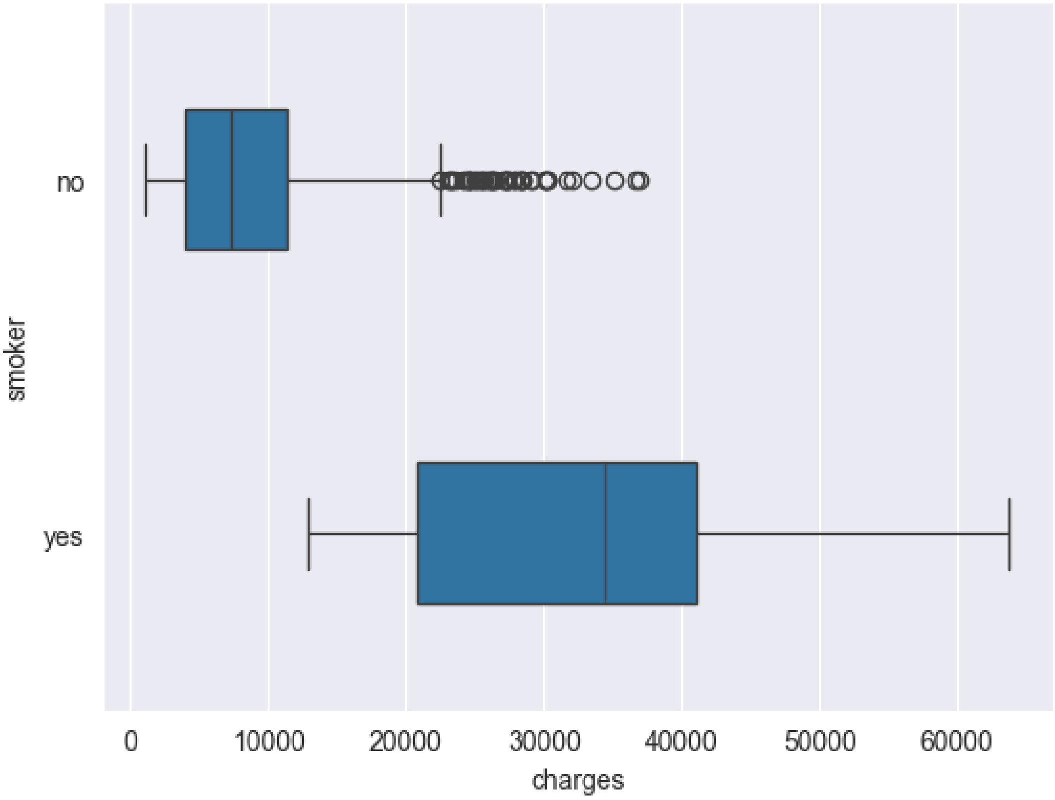
(g) (1 point) Explain whether there are trends that you can recognize in the histogram for the charges of patients based on their BMI.

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

(h) (2 points) Create a scatter plot matrix for the numerical features of the dataset. Use the *smoker* feature as a hue parameter in Seaborn's pairplot. Report one interesting finding for the relation between *charges* and the other features considering the *smoker* feature.

(i) (2 points) Compute the correlations for all the feature pairs from *age*, *bmi*, *children*, and *charges*. Provide a screenshot or table showing these correlations. State the strongest absolute correlation for the given distinguishing feature pairs and explain what this type of correlation means.

(j) (4 points) In the following, you see boxplots (as introduced in the lecture) of the charges grouped by smokers and non-smokers among the individuals. See the questions below and answer each of the following questions (or state why the question cannot be answered).



- Are there more smokers or non-smokers with charges above 20,000?
- Are there outliers for the smoker group that have values below the lower fence value?
- Is the median of charges for the group of smokers higher than the mean of charges for the group of smokers?
- Is the lower 1st quartile fence of smokers lower than the upper whisker of the non-smokers?

Question 2: Decision Trees (24 points)

(a) (3 points) Create a prediction model, which we call the *wishful-thinking* model, for predicting the *chargeGroup* feature by always predicting *low*. Then evaluate the model using the `test_insurance_one_hot.csv`. Compute and report the accuracy of the *wishful-thinking* model.

(b) (7 points) Create a prediction model, which we call the *mode-based* model, for predicting the *chargeGroup* feature. This model should always predict the mode of the *chargeGroup* feature of the `train_insurance_one_hot.csv`.

- Compute and report the mode of the *chargeGroup* feature in the `train_insurance_one_hot.csv`.
- Then evaluate the *mode-based* model using the `test_insurance_one_hot.csv`. Compute and report the accuracy of the *mode-based* model.
- State whether the accuracy of the *mode-based* model is higher or lower than the accuracy of the *wishful-thinking* model and **explain why** in about 2-3 sentences.

(c) (8 points) You want to predict the *chargeGroup* feature using a decision tree. Load the `train_insurance_one_hot.csv` as training data and the `test_insurance_one_hot.csv` as test data. The target variable is the *chargeGroup* feature. All other features **except for the charges, chargeGroup, and id features** are to be used as descriptive features. Make sure to remove the unwanted features from the dataset before training the decision trees. 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 9 as the `max_depth`. Compute the accuracy for each decision tree, using the `test_insurance_one_hot.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.
- 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.

(d) (6 points) Create and visualize a decision tree with the following parameters (criterion *gini*, `min_samples_leaf=6`, `random_state=42`, and `max_depth=2`) from the `train_insurance_one_hot.csv` (with the same features as target and input features as in the task above).

- Provide the visualized tree.
- Explain which features of a person lead to a classification of the *chargeGroup* as medium in the leaf nodes of the decision tree.
- Give the predicted *chargeGroup* for a 42-year-old man with a BMI of 36 and two children who does not smoke and lives in the northwest.

Question 3: Clustering (14 points)

(a) (14 points) Perform K-means clustering (from sklearn.cluster) on the `insurance.csv` dataset (only use these features for clustering: ’age’, ’bmi’, ’children’, and ’charges’) with 3 clusters. Use the StandardScaler from sklearn (sklearn.preprocessing) to standardize the data before applying K-Means. Use as the init value ”k-means++” and for the random state 42.

- Provide the centroids for the three clusters.
- Also give the cluster size for each cluster.
- Provide the number of smokers and non-smokers in each cluster.
- Finally, visualize the clusters using a pairplot from the seaborn library (sns.pairplot). Use the three clusters as hue.
- Comment in 2-3 sentences on any patterns you observe in the clusters.

Question 4: Regression (12 points)

(a) (6 points) Plot the charges over the age with the Seaborn library's *lmplot* function and show the linear regression line on the `train_insurance_one_hot.csv` dataset. Also, comment (in 2-3 sentences) whether a linear regression on the age feature alone seems to be a good fit for the data.

(b) (6 points) Create a linear regression model using the `train_insurance_one_hot.csv` dataset (excluding the id and chargeGroup feature) to predict the charges feature. Use the `LinearRegression` class from the *scikit-learn* library.

- Print the coefficients of the model and provide a screenshot of the coefficients.
- Print the mean absolute error (MAE) on the train dataset.
- Evaluate the model using the `test_insurance_one_hot.csv` dataset and report the Mean Absolute Error (MAE) of your model.

Question 5: Support Vector Machines (15 points)

In this question, you train a binary SVM classifier to predict whether a person has very high charges or not, referred to as *binary classification.*

Therefore, load the data, prepare the two different target values for the classification problem, and separate the descriptive features from the target variable. Hence, you must do the following:

- Load the training data `train_insurance_one_hot.csv` and test data `test_insurance_one_hot.csv`.
- Create a new version of each dataset for the binary classification: *expensive* is true for all instances with charges above 25,000 USD and false otherwise.
- From this new version, remove the features *chargeGroup* and *charges.*
- Create the target vectors y for the training and test data for the classification problem.

(a) (3 points) You build initially two SVM binary classifiers using (1) only the three features *age*, *bmi*, and *smoker* and (2) all descriptive features (except *id*). Train two linear SVM classifiers with a regularization parameter $C = 10$ on the training data and predict the target feature of the test instances. Provide for both classifiers the corresponding confusion matrices.

(b) (4 points) For the previous two confusion matrices, state and interpret your model's accuracy and precision of the SVM based on only the three features. Further, discuss which model is more accurate and precise and if it matches with your expectations.

You want to further investigate why the two models deviate and you want to fix the issue that makes them deviate.

(c) (4 points) For both training data sets, normalize them using `sklearn`'s `StandardScaler` and retrain on the data two new SVMs. Provide the confusion matrices and describe what has improved when comparing to the SVMs trained from the unscaled data. Name which SVM provides better results. Describe if you notice any changes in the processing duration of your SVM training pipeline (including time).

(d) (2 points) Investigate and communicate how decreasing the value of the regularization parameter (C) influences the accuracy of the resulting SVM that is trained on the unnormalized data. Explain why changing the regularization influences the accuracy.

(e) (2 points) Given your results from the normalization, explain in at most three sentences why normalization influences the accuracy.

Question 6: Neural Networks & Naïve Bayes (14 points)

(a) (5 points) Load `train_insurance_one_hot.csv` and `test_insurance_one_hot.csv` as the training data and test data, respectively. Train a multi-layer perceptron (MLP) using all descriptive features (except *charges* and *id*) for predicting the charge group. 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 number of iterations: 2000

Report on:

- The accuracy and confusion matrix of the neural network on the test data.
- What strikes you about the confusion matrix?
- Investigate further the predictions of the neural network: Use `predict_proba` to obtain the probability estimates for all individual instances.
  - What do you observe about the probability estimates and their variance?
  - Give the probability estimates for the first five instances of the test set.

(b) (5 points) Suggest an approach to improve the accuracy of the neural network and demonstrate that it indeed improves the result. To that end, you are allowed to preprocess the data, to modify the neural network, and to retrain the neural network. For the chosen approach, describe your adjustments in detail and show the resulting confusion matrix, and give the resulting accuracy.

(c) (4 points) Train a naïve Bayesian classifier using `sklearn`'s `GaussianNB` with and without standardization. For both classifiers, give the confusion matrix and the accuracy of the classifier.