Portfolio-Exam

This is the description of the different tasks for the portfolio-exam of the Data Science course MADS-ML (Machine Learning). The portfolio exam contains several tasks. Over the course of the term, you will learn the necessary means to complete these tasks. Your solutions must be submitted at the end of the term and tell the story of one data science experiment. This exam is supposed to be done during the self study time of the module. Students are allowed to exchange ideas. However, this is **NOT a teamwork exercise**. Every student must derive and write up their own solutions in their own words and programming style.

Your task: Conduct a supervised machine learning project on a suitable real-world dataset of your choice (see criteria below). The result will be a Jupyter Notebook containing comprehensive experiments, conducted according to the following requirements.

Read this document (5 pages) carefully – it describes the conditions and criteria for grading!

1 Submission

All parts of the portfolio should be contained in one Jupyter Notebook called `experiments.ipynb`. You may submit either just the notebook or a zip file containing

- the notebook,
- an optional resources folder if you have additional resources (e.g. images), and
- a data folder if your data is not simply available online (cf. Section 3).

Upload your results to Moodle **before 23:59 o'clock (German time) Jan. 06, 2024**.
Submissions after the deadline or via means other than Moodle will not be considered!

2 The Portfolio Tasks

The full portfolio will resemble a term paper including story, experiments, and conclusions. The main idea is to conduct a comprehensive experiment on a real-world dataset using classification or regression methodology. All tasks will be submitted together in one Jupyter Notebook. Think of the notebook as a report of a (small) project which you – as a data scientist in a data science company – conduct for a customer. The following tasks are supposed to be completed in order.
Note: Depending on the project and datasets you choose, some of the following steps of the portfolio may be more or less extensive. E.g. if your dataset is already cleaned up, preprocessing might be very quick. However, in other situations you might want to make use of various preprocessing steps and iteratively refine the dataset before you run algorithms on it.

Hint: For the portfolio, you will choose between a classification and a regression experiment. In the lectures, we focus mostly on classification algorithms. Regression is only discussed in the last part of the module. However, each of the discussed classification algorithms has a regression counterpart, e.g., `KNeighborsRegressor` is the regression algorithm similar to `KNeighborsClassifier`.

2.1 Task 1 – Story

In the first part, explain the story and circumstances of the experiments.

- Imagine and describe a fictitious situation or describe a real scenario in which you are a data scientist working for some organization.
- Explain the context – project or task – for your experiment. The task must be suitable for a regression or classification experiment.
- Explain the plan of the experiment and the purpose – the value you expect to create for the organization.

2.2 Task 2 – The Data

In the second part, load and present a dataset.

- Explain the dataset itself (e.g., what do the features represent, what are the classes or what is the target quantity?).
- Explain how the dataset is suitable for the project from Task 1.
- Condition 1: Use real-life data. Do NOT use artificial datasets NOR datasets from the lecture!
- Condition 2: There are two possible availability scenarios for the data: A) It is available online with proper license. In that case indicate in your notebook where the data can be downloaded. B) You (legally!) obtain a dataset and share it with me via Moodle. Such data should not come with any form of NDA or other obligations. If you are not sure about these criteria regarding the dataset you consider, please contact me before you invest too much time in the experiments. If you write your own code to acquire data (e.g. querying an API), create a **separate notebook** for that purpose only and submit everything in a zip file.

2.3 Task 3 – IDA

Conduct an initial data analysis.

- Present some distributions and statistical properties that inform the reader about the dataset or that are relevant for your project.

2.4 Task 4 – EDA, Preprocessing

Bring the dataset into the form that you need for the experiments.

- Explore the data and conduct necessary transformations.
- If necessary, use different means of preprocessing until the dataset is suitable.
- Define target and features and set them appart in appropriate data structures.

2.5 Task 5 – A First Impression

Create some preliminary results to get a first impression on the difficulty of the problem.

- Create a single train-test split to run some preliminary experiments.
- Determine an appropriate baseline for your task and run it on the created split.
- Use a suitable classification or regression algorithm with default parameters to get a first impression on its performance.

2.6 Task 6 – Algorithms and their Parameters

Select 3 ML algorithms from the lectures (or their regression counterparts) and create appropriate validation curves.

- Select 3 algorithms. For

  classification use decision trees and two further algorithms from the lectures.

  regression use a decision tree regressor and two further algorithms, using the regression-counterparts of the algorithms from the lectures.

- Discuss and use (separately) two different ways of handling overfitting in decision trees.
- For each algorithm, create and interpret one or more validation diagram(s) to get an impression on the hyperparameter influence.
- For each algorithm, choose a reasonable grid of hyperparameters that will be investigated in the next portfolio task.

2.7 Task 7 – Nested Cross Validation

Setup a proper nested cross validation experiment to assess and compare the performance of different algorithms.

- Use the three algorithms from Task 6 and the hyperparameter grids you determined.
- Compare your final performance estimates of the algorithms in a table using more than just one performance metrics.

- Do not forget the baseline in that comparison.
- Compare average performance and standard deviation!
- Discuss your results and formulate a recommendation with regard to the choice of models.

2.8 Task 8 – Final Production Model

Determine the model you would run in production.

- Based on your above recommendation, pick one algorithm as model for production.
- Use cross validation to determine the best hyperparameters.
- Store the model in a suitable format, such that it could be deployed in a production environment (running Python).

2.9 Task 9 – Conclusions and Future Work

- Summarize and interpret the achieved results.
- Critically reflect and assess the usability of the applied methodology in the context of your task (Task 1).
- Explain the generated value.
- Explain limitations.
- Propose ideas for future work (a short sketch or enumeration of ideas is sufficient, no further experiments).

3 Conditions

Dataset Adhere to the conditions mentioned in Task 2!

Programming Language Use Python for the submission.

Language Choose between English and German for all textual content.

Code from other Sources You may reuse all the code from this module's lectures and exercises. Copying (and adapting) from other sources is allowed in small quantities – e.g. a function from stackoverflow. Quote the respective source. WARNING: Copying code in large quantities will be treated as intent to deceive and result in a score of zero points.

4 Expectations

Your final score will be composed of 25 points for story, 65 points for the actual experiments, and 10 points for presentation. (Note that poor presentation can lead to loss of points in the other two categories as well, e.g. if it's too confusing!)

4.1 Story

The report should follow a straight story (Task 1). The steps and experiments in Tasks 2–8 should fit to that story. The conclusion (Task 9) should be written in the context of that story and comment on how the expectations were fulfilled or why they have not been fulfilled.
Hint: Your notebook should contain MUCH MORE TEXT, introduction, comments, etc. than the notebooks we use to demonstrate methods in the lectures or the exercises!

4.2 Experiments

When grading your experiments, I consider technical soundness, completeness, and fit to the proposed task. I expect (among others):

1. The task is a non-trivial data science task.
2. Your experiments contain all the above requirements (see Section 2).
3. Your code is executable and yields reasonable and reliably replicable results.
4. All cells of the notebook have been executed.
5. The experiments address the influence of random choices and and of class imbalance and they use appropriate steps to mitigate them.

4.3 Presentation

When grading the presentation, I will put myself into the position of your project's customer. I expect (among others),

1. that the imports are organized,
2. that the code is documented using appropriate means of Jupyter,
3. the portfolio steps are highlighted,
4. that there is no unnecessary code, no lengthy debug output, no error messages,
5. that functions are used to avoid repeating similar code (DRY principle),
6. that results are presented in tables and customized diagrams which are referenced and **interpreted** in the text,
7. that diagrams are easy to understand (appropriate colors, tics, scaling, labelling, legends, ...),
8. that the document is structured through appropriate means (highlighting, sections and subsections, bullet point lists, ...),
9. that I am guided through the different parts of the experiments and told what the purpose of upcoming code blocks will be.

Final Hint: Keep in mind that this is NOT your Master Thesis. The experiments should be comprehensive and created and conducted solely by you. However, limit your work towards solving ONE task – even though along the way you might recognize other interesting angles to follow up on.