Machine Learning (6614ZD010Y)
Group Project

The Group Project will test your understanding of topics covered in the entire course. You will have to demonstrate the ability to independently identify an interesting problem captured in the data, apply machine learning to analyse it and/or make predictions, evaluate the results and draw relevant conclusions.

Plagiarism

The University of Amsterdam rigorously monitors for plagiarism and collusion in assignments and exams. All submissions will be checked using Turnitin Similarity. Any cases of plagiarism will be reported to the Examinations Board for sanctions. Always remember to properly cite your sources. For guidance on citing and referencing, please visit the UvA Website.

Submission

The assignment will count toward your grade and should be submitted through Canvas by **17.03.2025 at 18:29 (CET)**. Your submission consists of:

- A Jupyter Notebook named `code-GROUPNUMBER.ipynb`
- A report in PDF format `report-GROUPNUMBER.pdf`

To test the assignments we will use **Anaconda Python 3.12**. **The report should not be longer than 10 pages**, including the title page, figures, and references. The title page should clearly list the group members and the dataset(s) used for the project. You will also have to provide the data that your code uses to perform the analysis. If this is too large to upload to Canvas, you can provide us with a URL to download the data, packed in a single file `data-GROUPNUMBER.zip` or `data-GROUPNUMBER.tar.gz`.

Grading

You can get at most 30 points for this project, which is 30% of your final grade.

✉ u.sharma@uva.nl, m.r.h.dejonge@uva.nl

1 openML Datasets

In this homework, you will use openML, a library designed to provide simple and fast access to datasets. The usage of this library alleviates the burden of downloading and loading datasets yourself. A single instruction will fetch the dataset for your experiments and allow you to proceed without delay. With the exception of the first dataset ("Public Health and Health Care") and your own datasets, all other choices can be conveniently loaded using this libary.

1.1 Introduction

This tutorial provides a step-by-step guide for students on how to use the `openML` platform in Python. The guide explains how to install the library, load a dataset, summarize its contents, and extract data for further analysis.

1.2 Installing and Importing the openML Library

```
!pip install openml
import openml
```

- **Installation:** The command `!pip install openml` installs the `openML` package so that your Python environment can interact with the openML platform.
- **Importing:** The command `import openml` imports the library and makes its functions and classes available in your code.

1.3 Loading a Dataset

```
# This is done based on the dataset ID.
dataset = openml.datasets.get_dataset(dataset_id="diamonds", version
    =1)
```

- **Fetching the Dataset:** The function `get_dataset` retrieves the dataset based on the specified dataset ID (`"diamonds"`) and version (`1`). For each problem, we provide a unique dataset ID that you can use to retrieve the relevant dataset for that task.
- **Dataset Object:** The returned `dataset` object contains metadata such as the dataset's name, default target attribute, URL for additional details, and a full description.

1.4 Summarizing the Dataset

```
# Print a summary
print(
    f"This is dataset '{dataset.name}', the target feature is "
    f"'{dataset.default_target_attribute}'"
)
print(f"URL: {dataset.url}")
print(dataset.description)
```

- **Dataset Name and Target Feature:** The first print statement displays the dataset's name and its primary target feature using `dataset.name` and `dataset.default_target_attribute`.
- **URL and Description:** The following print statements show the dataset's URL, which leads to additional details on openML, and a detailed description of the dataset.

1.5 Extracting the Data for Analysis

```
X, y, categorical_indicator, attribute_names = dataset.get_data(
    target=dataset.default_target_attribute
)
print(X.head())
print(X.info())
```

- **Data Extraction:** The `get_data` method splits the dataset into:
  - `X`: The features (input variables).
  - `y`: The target variable to predict.
  - `categorical_indicator`: A list indicating which features are categorical.
  - `attribute_names`: The names of the columns.
- **Data Overview:**
  - `X.head()` prints the first few rows of the features to provide a glimpse of the data.
  - `X.info()` gives detailed information about the DataFrame, such as data types and non-null counts.

Datasets

Public Health and Health Care

In the Netherlands, Rijksinstituut voor Volksgezondheid en Milieu (RIVM) publishes public Health datasets throught the platform Volksgezondheidenzorg.info [1]. The list of interesting collections hosted on the website include i.a. mortality from the cardiovascular diseases and cancer per municipality, as well as community health services region (i.e. GGD-regio) as well as related EU statistics. When searching for answers to your research questions, you could combine these datasets with e.g. CBS Neighbourhood Statistics [2].

Tracking Eye State using EEG

In this dataset, you are provided with one continuous EEG measurement with the Emotiv EEG Neuroheadset. The duration of the measurement was 117 seconds. The eye state was detected via a camera during the EEG measurement and added later manually to the file after analyzing the video frames. '1' indicates the eye-closed and '0' the eye-open state. All values are in chronological order with the first measured value at the top of the data. The objective of this task is to detect if the eye was open or closed using EEG measurements alone.

The openML code for this dataset is `"eeg-eye-state"`.

Credit Risk Prediction

The German Credit dataset is designed for the classification of individuals into two groups: those who represent a good credit risk and those who represent a bad credit risk. The primary goal of this dataset is to build models that predict whether a customer is likely to default based on a range of financial and personal attributes. This dataset can be used in credit risk modeling, machine learning classification tasks, and decision-making processes in finance. By understanding both the financial history and personal background of individuals, models can be trained to predict credit risk more accurately.

The openML code for this dataset is `"credit-g"`. Note that the dataset description also includes a cost matrix, which you may choose to use or disregard.

Diamond Price Prediction

The Diamond Price dataset contains prices and other attributes for nearly 54,000 diamonds. It provides not only the prices of the diamonds but also several key attributes that describe their physical characteristics and quality. The task is to predict the price of the diamond using these attributes.

The openML code for this dataset is `"diamonds"`.

Your own dataset

You are welcome to try Machine Learning on a dataset of your own. Keep in mind that this is a group project. Therefore, the difficulty of the dataset and the task must be challenging. Please contact the lecturers if you are not sure about the difficulty of your task/dataset.

References

[1] Volksgezondheid en zorg. https://www.vzinfo.nl/.

[2] CBS. Kerncijfers wijken en buurten 2016. https://www.cbs.nl/nl-nl/maatwerk/2016/30/kerncijfers-wijken-en-buurten-2016.