Data Preparation for Machine Learning (M1)

Analysis of the *Heart Disease Dataset*

Jakub Jakubowicz (2025121082)

Gracjan Jasnos (2025109276)

October 2025

Abstract

The purpose of this report is to describe the process of preparing a real-world dataset for later use in Machine Learning algorithms. The selected dataset, the *Heart Disease Dataset*, was obtained from the Kaggle platform and represents a binary classification problem — predicting the presence or absence of heart disease in a patient. This report details the dataset exploration, cleaning, transformation, and construction of data processing pipelines.

Contents

1 Introduction

The main objective of Milestone **(M1)** is to prepare a real-world dataset that can later be used for both supervised and unsupervised learning techniques in the following stage **(M2)**. This stage focuses on ensuring that the data is properly cleaned, transformed, and organized in a way that supports the implementation of Machine Learning algorithms.

The selected dataset for this project is the ***Heart Disease Dataset***, obtained from the Kaggle platform. It was chosen because it presents a binary classification problem—predicting whether a patient is likely to suffer from heart disease based on a set of medical and demographic features. This dataset is well-suited for the objectives of this milestone since it contains a balanced combination of numerical, categorical, and ordinal variables, and provides enough samples to ensure reliable model validation.

The structure of this report follows a systematic approach. First, the dataset is described in detail, including its source, features, and target variable. Then, exploratory data analysis **(EDA)** is performed to understand the characteristics and distribution of the data. Subsequently, the data preparation process is presented, covering cleaning, transformation, encoding, and normalization steps. Finally, the construction of the data processing pipeline and a brief summary of conclusions are provided.

2 Dataset Description

2.1 Data Source

The dataset used in this project is the *Heart Disease Dataset*, which was obtained from the ***Kaggle platform***[1]. This dataset was originally compiled from a collection of medical records and aims to predict the presence of heart disease in patients based on several clinical and demographic attributes.

The medical context of the dataset lies in cardiovascular risk prediction, which is a critical application of Machine Learning in healthcare. By analyzing features such as age, cholesterol level, resting blood pressure, and electrocardiographic results, predictive models can help medical professionals identify individuals at higher risk of heart disease. The dataset therefore represents an ideal example of a binary classification problem within a real-world medical scenario.

2.2 Dataset Characteristics

The dataset contains **1,025 instances** (rows) and **14 attributes** (columns), including both input features and the target variable. These features are a combination of numerical, categorical, and ordinal data types, providing a suitable diversity for testing different preprocessing and modeling techniques.

Table 1 summarizes the most relevant characteristics of each feature, including its name, data type, and a short description of its meaning.

[1] `https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`

Table 1: Description of dataset features

Feature Name	Type	Description
`age`	Numerical	Age of the patient (years)
`sex`	Categorical	Sex of the patient (1 = male, 0 = female)
`cp`	Ordinal	Chest pain type (0–3, indicating increasing severity)
`trestbps`	Numerical	Resting blood pressure (mm Hg)
`chol`	Numerical	Serum cholesterol level (mg/dl)
`fbs`	Categorical	Fasting blood sugar >120 mg/dl (1 = true, 0 = false)
`restecg`	Ordinal	Resting electrocardiographic results (0–2)
`thalach`	Numerical	Maximum heart rate achieved
`exang`	Categorical	Exercise-induced angina (1 = yes, 0 = no)
`oldpeak`	Numerical	ST depression induced by exercise relative to rest
`slope`	Ordinal	Slope of the peak exercise ST segment (0–2)
`ca`	Ordinal	Number of major vessels (0–3) colored by fluoroscopy
`thal`	Categorical	Thalassemia status (0–3)
`target`	Binary	Presence of heart disease (1 = yes, 0 = no)

Each feature contributes to the overall diagnosis or risk assessment of heart disease. For example, variables such as `chol` and `trestbps` are continuous physiological measures, while `cp` and `thal` encode categorical or ordinal clinical indicators.

2.3 Target Variable

The target variable, named `target`, represents the presence (`1`) or absence (`0`) of heart disease in a patient. This defines a binary classification problem, where the goal of the machine learning model will be to predict whether a given patient is likely to have heart disease based on the other features.

An initial inspection of the dataset shows that approximately **55%** of the records correspond to patients diagnosed with heart disease (`target = 1`), while **45%** correspond to patients without heart disease (`target = 0`). This relatively balanced distribution ensures that classification algorithms will not suffer from significant class imbalance issues, making the dataset well-suited for supervised learning experiments in later stages.

3 Exploratory Data Analysis (EDA)

The exploratory data analysis phase aimed to better understand the structure and distribution of the features in the *Heart Disease Dataset*, as well as to detect possible inconsistencies, missing values, and relationships among variables. This section summarizes the main results of the descriptive statistics and visualization analysis.

3.1 Descriptive Statistics

The dataset contains a total of 1,025 instances and 14 features, including both numerical and categorical attributes. Table 2 presents the main descriptive statistics for the numerical variables.

Table 2: Descriptive statistics of numerical features

Feature	**Mean**	**Std**	**Min**	**Median**	**Max**
Age	54.43	9.07	29	56	77
Resting Blood Pressure (trestbps)	131.61	17.52	94	130	200
Cholesterol (chol)	246.00	51.59	126	240	564
Maximum Heart Rate (thalach)	149.11	23.01	71	152	202
ST Depression (oldpeak)	1.07	1.18	0.0	0.8	6.2

The average age of the patients is approximately **54 years**, with the youngest being **29** and the oldest **77** years old. Most participants have cholesterol values around **240–250** mg/dl and resting blood pressure close to **130** mmHg. No missing values were detected in any of the 14 features, indicating a well-curated dataset suitable for direct processing in the next phase.

3.2 Feature Types and Distributions

The dataset includes a mix of numerical, categorical, and ordinal features. For instance, variables such as `sex`, `fbs`, and `exang` are binary categorical attributes, while `cp` (chest pain type), `restecg` (ECG results), and `slope` are ordinal. Continuous medical measures include `age`, `trestbps`, `chol`, and `thalach`. Table 3 summarizes the number of unique values for each variable.

Table 3: Feature type summary

Feature	**Data Type**	**Unique Values**
`age`	int64	41
`sex`	int64	2
`cp`	int64	4
`trestbps`	int64	49
`chol`	int64	152
`fbs`	int64	2
`restecg`	int64	3
`thalach`	int64	91
`exang`	int64	2
`oldpeak`	float64	40
`slope`	int64	3
`ca`	int64	5
`thal`	int64	4
`target`	int64	2

3.3 Target Variable Distribution

The target variable (`target`) indicates whether a patient is diagnosed with heart disease (`1`) or not (`0`). Out of 1,025 total cases, **526 patients** (51.3%) were diagnosed with heart disease, while **499** (48.7%) were not. This relatively balanced class distribution suggests that there are no significant class imbalance issues.

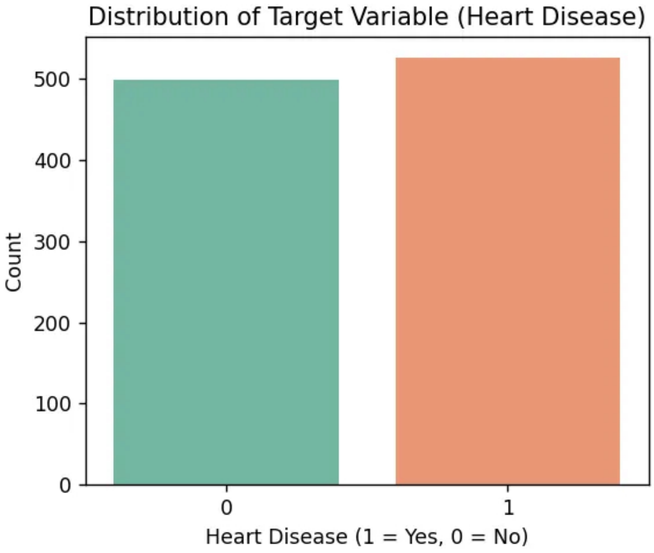
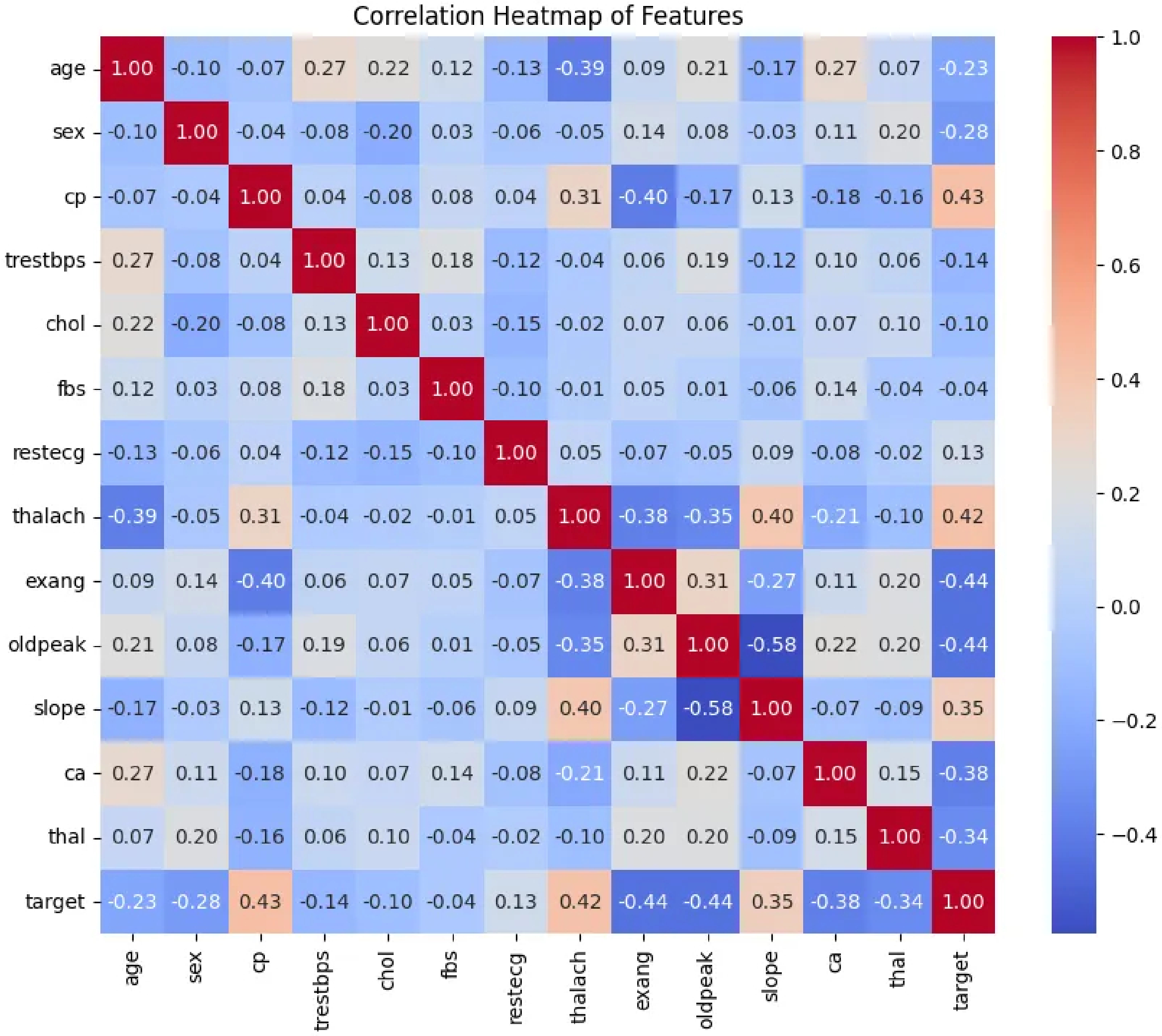


Figure 1: Distribution of the target variable (heart disease: 1 = yes, 0 = no).

3.4 Correlation and Feature Relationships

A Pearson correlation analysis was performed to explore the relationships between the numerical features. Figure 2 presents the correlation heatmap.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.00	-0.10	-0.07	0.27	0.22	0.12	-0.13	-0.39	0.09	0.21	-0.17	0.27	0.07	-0.23
sex	-0.10	1.00	-0.04	-0.08	-0.20	0.03	-0.06	-0.05	0.14	0.08	-0.03	0.11	0.20	-0.28
cp	-0.07	-0.04	1.00	0.04	-0.08	0.08	0.04	0.31	-0.40	-0.17	0.13	-0.18	-0.16	0.43
trestbps	0.27	-0.08	0.04	1.00	0.13	0.18	-0.12	-0.04	0.06	0.19	-0.12	0.10	0.06	-0.14
chol	0.22	-0.20	-0.08	0.13	1.00	0.03	-0.15	-0.02	0.07	0.06	-0.01	0.07	0.10	-0.10
fbs	0.12	0.03	0.08	0.18	0.03	1.00	-0.10	-0.01	0.05	0.01	-0.06	0.14	-0.04	-0.04
restecg	-0.13	-0.06	0.04	-0.12	-0.15	-0.10	1.00	0.05	-0.07	-0.05	0.09	-0.08	-0.02	0.13
thalach	-0.39	-0.05	0.31	-0.04	-0.02	-0.01	0.05	1.00	-0.38	-0.35	0.40	-0.21	-0.10	0.42
exang	0.09	0.14	-0.40	0.06	0.07	0.05	-0.07	-0.38	1.00	0.31	-0.27	0.11	0.20	-0.44
oldpeak	0.21	0.08	-0.17	0.19	0.06	0.01	-0.05	-0.35	0.31	1.00	-0.58	0.22	0.20	-0.44
slope	-0.17	-0.03	0.13	-0.12	-0.01	-0.06	0.09	0.40	-0.27	-0.58	1.00	-0.07	-0.09	0.35
ca	0.27	0.11	-0.18	0.10	0.07	0.14	-0.08	-0.21	0.11	0.22	-0.07	1.00	0.15	-0.38
thal	0.07	0.20	-0.16	0.06	0.10	-0.04	-0.02	-0.10	0.20	0.20	-0.09	0.15	1.00	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.10	-0.04	0.13	0.42	-0.44	-0.44	0.35	-0.38	-0.34	1.00

Figure 2: Correlation heatmap of all features.

The heatmap shows several moderate correlations. Notably, `cp` (chest pain type) and `thalach` (maximum heart rate achieved) have positive correlations with the `target` variable (˜**0.43** and **0.42**, respectively), suggesting that patients with higher chest pain type scores and higher heart rates are more likely to be diagnosed with heart disease. Conversely, `exang` (exercise-induced angina), `oldpeak`, and `ca` show negative correlations (˜**-0.44**, **-0.44**, and **-0.38**), indicating that these features tend to be higher in patients without heart disease.

3.5 Missing Values and Outliers

The analysis confirmed that the dataset contains **no missing values** in any feature. However, visual inspection of the numerical distributions (not shown here) suggests that certain variables such as `chol` and `oldpeak` may contain potential outliers, as evidenced by their relatively large maximum values (564 mg/dl for cholesterol and 6.2 for oldpeak). These values will be further examined in the data preprocessing phase to assess their impact on model performance.

In summary, the exploratory analysis revealed that the dataset is complete, well-structured, and balanced. The relationships observed between medical indicators and the target variable confirm the dataset's suitability for supervised learning in Milestone M2.

4 Data Preparation

The data preparation stage aimed to transform the raw *Heart Disease Dataset* into a clean, structured, and standardized format suitable for applying machine learning algorithms in the next milestone (M2). This phase included several key steps: cleaning, feature encoding, normalization, and dataset splitting.

4.1 Data Cleaning

The initial dataset consisted of 1,025 records and 14 attributes. An initial inspection confirmed that there were no missing values in any of the columns, as shown in the summary of missing values. However, the analysis revealed the presence of **723 duplicated rows**, which significantly inflated the dataset size. These duplicates were removed to ensure data integrity and avoid bias during model training. After removing duplicate instances and resetting the index, the final cleaned dataset contained **302 unique records**. A verification step confirmed that no missing or inconsistent values remained in the cleaned dataset.

This cleaning process ensures that all subsequent analysis and modeling steps are based on unique and valid patient records, improving the reliability of the predictive models that will be developed in later stages.

4.2 Feature Encoding

The dataset includes a mixture of categorical, ordinal, and numerical features. To convert categorical and ordinal variables into a numerical format suitable for machine learning models, a **one-hot encoding** technique was applied using the `OneHotEncoder` class from the `scikit-learn` library.

The categorical variables encoded were: `sex`, `cp`, `fbs`, `restecg`, `exang`, `slope`, `ca`, and `thal`. The encoder was configured with the parameter `drop="first"` to avoid the dummy variable trap and reduce redundancy. After encoding, the dataset expanded to **23 columns**, which included both the transformed categorical variables and the numerical features (`age`, `trestbps`, `chol`, `thalach`, and `oldpeak`), as well as the target variable.

This encoding step ensures that categorical attributes are properly represented and interpretable by mathematical models without introducing multicollinearity.

4.3 Normalization and Standardization

Before model training, all numerical features were standardized using the `StandardScaler` method from `scikit-learn`. This method transforms the data to have a mean of **0** and a standard deviation of **1**. Standardization is particularly appropriate for algorithms sensitive to feature scaling (such as logistic regression, support vector machines, or neural networks) since it ensures that all features contribute equally to the learning process, regardless of their original measurement units.

The features that underwent standardization were: `age`, `trestbps`, `chol`, `thalach`, and `oldpeak`. After scaling, all variables were verified to ensure that no NaN values had been introduced during transformation.

4.4 Dataset Splitting

After data cleaning, encoding, and scaling, the dataset was divided into training and testing subsets. The `train_test_split` function from `scikit-learn` was used with an **80/20 ratio** and `stratify=y` to maintain the same class distribution between the two sets.

The training set contained **241 samples** (approximately 80% of the data), while the test set included **61 samples** (20%). This split ratio was chosen to provide a sufficient number of samples for training while preserving an adequate portion for unbiased evaluation. The use of stratified sampling ensures that both subsets maintain the same proportion of patients with and without heart disease, which is critical for balanced classification performance.

Overall, the data preparation process resulted in a clean, encoded, and standardized dataset ready for the development and evaluation of machine learning models in Milestone M2.

5 Data Processing Pipeline

Implementation of the Data Transformation Pipeline

A complete preprocessing and modeling pipeline was implemented using the `scikit-learn` framework. This approach ensures reproducibility, modularity, and consistent data handling throughout the workflow.

The pipeline is composed of the following main components:

1. **ColumnTransformer** (`preprocessor`)

    - **Categorical features:** transformed using `OneHotEncoder(drop='first', handle_unknown='ignore')` to convert non-numerical variables into binary indicators while avoiding multicollinearity. Features included: `sex, cp, fbs, restecg, exang, slope, ca, thal`.
    - **Numerical features:** standardized using `StandardScaler()` to center the data (mean = 0) and scale to unit variance. Features included: `age, trestbps, chol, thalach, oldpeak`.

2. **Classifier:**

    A baseline linear model, `LogisticRegression(max_iter=1000)`, was integrated as the final step of the pipeline. This provides a simple yet interpretable model for the binary heart disease classification task.

The dataset was divided using `train_test_split` with stratification to preserve class balance. This resulted in **241 training samples** and **61 test samples**.

Finally, the complete preprocessing + model object was serialized with `joblib` as:

`heart_pipeline.joblib`

This allows the exact same transformation and model to be reused in the next milestone (M2) for model comparison and tuning.

Step-by-step explanation / visual diagram

1. **Input:** raw features **X** split into categorical and numerical subsets; labels **y** from `target`.

2. **Transform–categorical:** apply one-hot encoding with reference-level drop to avoid multicollinearity (*dummy variable trap*) and with `handle_unknown=’ignore’` for robustness.

3. **Scale–numerical:** standardize to zero mean and unit variance to place features on comparable scales (beneficial for linear models).

4. **Concatenate:** `ColumnTransformer` merges transformed categorical and scaled numerical matrices.

5. **Fit classifier:** train `LogisticRegression` on the transformed training set.

6. **Predict & evaluate:** transform the test set inside the pipeline and compute metrics.

A schematic view (as displayed by `set_config(display=’diagram’)`):

```
Pipeline
  |-- preprocessor: ColumnTransformer
  | |-- categorical: OneHotEncoder(drop=’first’, handle_unknown=’ignore’)
  | |-- numerical: StandardScaler()
  |-- classifier: LogisticRegression(max_iter=1000)
```

Use of `scikit-learn` `Pipeline` or equivalent tools

We used `Pipeline` to chain preprocessing and modeling into a single, atomic estimator. This ensures that:

- all transformations are learned *only* on the training set (no data leakage),
- the exact same transformations are applied to validation/test data and, later, to production data,
- hyperparameter tuning (in M2) can jointly optimize preprocessing and modeling.

Transformed Feature Space

After preprocessing, the model receives a total of **22 transformed features**. The example below shows the first 10 feature names automatically generated by the `ColumnTransformer` and `OneHotEncoder` components:

```
categorical__sex_1,
categorical__cp_1, categorical__cp_2, categorical__cp_3,
categorical__fbs_1, categorical__restecg_1, categorical__restecg_2,
categorical__exang_1, categorical__slope_1, categorical__slope_2
```

These transformed feature names represent the encoded categorical variables combined with standardized numerical variables, forming the final input space for the classifier.

Baseline performance (test set)

Using the held-out test set ($\mathbf{n} = \mathbf{61}$), the baseline logistic regression achieved:

Class	Precision	Recall	F1-score	Support
0	0.828	0.857	0.842	28
1	0.875	0.848	0.862	33
Accuracy		**0.852**		
Macro avg	0.851	0.853	0.852	61
Weighted avg	0.853	0.852	0.853	61

These results confirm that the preprocessing choices are reasonable and provide a strong, transparent baseline for subsequent M2 experiments (e.g., regularization tuning, alternative classifiers, and model comparison).

6 Summary and Conclusions

The data preparation phase successfully transformed the raw *Heart Disease Dataset* into a clean, structured, and machine–learning–ready format. This process ensured that the dataset adheres to the quality standards required for reliable modeling and fair evaluation.

The following key steps were completed:

- **Data Cleaning:** Identified and removed 723 duplicate rows, ensuring that only unique and valid records were retained. Verified that no missing or inconsistent values were present.
- **Feature Encoding:** Converted categorical attributes (`sex`, `cp`, `fbs`, `restecg`, `exang`, `slope`, `ca`, `thal`) using one–hot encoding while avoiding multicollinearity through the use of the `drop="first"` parameter.
- **Normalization and Standardization:** Applied `StandardScaler()` to numerical variables to ensure that all features contribute equally to model training, improving algorithm stability and interpretability.
- **Dataset Splitting:** Divided the data into training and testing subsets (80/20 split) using stratified sampling, maintaining the same class proportions for balanced model evaluation.
- **Pipeline Construction:** Built a complete preprocessing and modeling pipeline with `ColumnTransformer` and `LogisticRegression`. The pipeline achieved a test accuracy of **0.852**, confirming that the chosen preprocessing steps were effective.

Overall, the prepared dataset is now fully suitable for use in supervised learning experiments in Milestone M2. The pipeline framework ensures reproducibility, reduces the risk of data leakage, and simplifies future experimentation with alternative models such as decision trees, support vector machines, or ensemble methods.

Recommendations for future work:

- Explore additional feature engineering techniques, such as polynomial feature expansion or feature selection, to further enhance model performance.
- Investigate the impact of potential outliers (especially in `chol` and `oldpeak`) and evaluate whether robust scaling or transformation improves results.
- In Milestone M2, compare multiple classifiers (e.g., Random Forest, Support Vector Machine, Gradient Boosting) using cross–validation to identify the best performing approach.

In conclusion, the data preparation for M1 has provided a solid, high–quality foundation for advanced machine learning modeling and evaluation in the upcoming milestone.

7 References

- **Kaggle Dataset:** John Smith. (2021). *Heart Disease Dataset.* Available at: `https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`
- **scikit-learn Documentation:** Pedregosa, F. et al. (2011). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12, 2825–2830. Official documentation: `https://scikit-learn.org/stable/`
- **pandas Documentation:** The pandas development team (2023). *pandas: Python Data Analysis Library.* Available at: `https://pandas.pydata.org/`
- **matplotlib Documentation:** Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment.* Computing in Science & Engineering, 9(3), 90–95. Official documentation: `https://matplotlib.org/`
- **UCI Machine Learning Repository:** Dua, D. and Graff, C. (2019). *UCI Machine Learning Repository.* Irvine, CA: University of California, School of Information and Computer Science. Available at: `https://archive.ics.uci.edu/ml`