

|  |
| --- |
| D1.4 Open-source library for data mining techniques |
|  |
| D 1.4 | 30.08.2023 |

***Logo, company name

Description automatically generated***

Deliverable No. 1.4 Data Mining Methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Acronym** | **Grant Agreement #** | **Project Title** | **Deliverable Reference #** | **Deliverable Title** |
| I4Driving | 101076165 | Integrated 4D Driver Modelling under Uncertainty | 1.4 | Open-source library for data mining techniques |

**AUTHORS**

|  |  |
| --- | --- |
| Andrea Saltelli | Institute for Cognitive Sciences and Technologies (Cnr-ISTC) |
| Giulia Vannucci | Institute for Cognitive Sciences and Technologies (Cnr-ISTC) |
|  |  |
|  |  |
|  |  |

**DISSEMINATION LEVEL**

|  |  |  |
| --- | --- | --- |
| X | P | PUBLIC |
|  | C | CONFIDENTIAL |

This project has received funding from the European Union’s Horizon Europe programme, under grant agreement No 101076165.

**Disclaimer**

Funded by the European Union. Views and opinions expressed in this publication are, however, those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

|  |  |
| --- | --- |
|  | Funded by the Horizon 2020 programme of the European Union  Grant Agreement No 101076165 |

Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Authors | Organisaton | Description |
| Original | August 30 | Giulia Vannucci, Andrea Saltelli | CNR | Level 0 draft |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Executive Summary

The major scientific challenge of the **i4Driving Project** is a human driver model that captures the relevant behavioral mechanisms for safety assessment. For the Driving Behavioral Analysis, Data Mining and Machine Learning must be considered. In this deliverable, we suggest and describe a list of available open-sources Language Programs and Libraries that are useful to implement data mining techniques. The present deliverable is integrated by a GitHub repository that contains proper linkages to the resources described here. The GitHub will be a living entity, and register worked cases from i4Driving as these are developed in the course of the project.

Contents

[Executive Summary 3](#_Toc144298425)

[1. Introduction 5](#_Toc144298426)

[2. Open-source Software 5](#_Toc144298427)

[2.1 Python Software 5](#_Toc144298428)

[2.1.1 Text editors and integrated development environments (IDEs) available 5](#_Toc144298429)

[2.2 R Software 6](#_Toc144298430)

[2.2.1 Text editors and integrated development environments (IDEs) available 6](#_Toc144298431)

[3. Libraries for data mining techniques 6](#_Toc144298432)

[3.1 Data manipulation and visualization 6](#_Toc144298433)

[3.2 Unsupervised learning 7](#_Toc144298434)

[3.2.1 Python libraries for unsupervised learning 7](#_Toc144298435)

[3.2.2 R libraries for unsupervised learning 7](#_Toc144298436)

[3.3 Supervised learning 8](#_Toc144298437)

[3.3.1 Python libraries for supervised learning 8](#_Toc144298438)

[3.3.2 R libraries for supervised learning 9](#_Toc144298439)

[3.4 How to install packages In Python and R 10](#_Toc144298440)

[References 11](#_Toc144298441)

# Introduction

This report describes a selection of Open-source library for data mining techniques which are linked on GitHub repository NAME OF GITHUB, keeping in mind that research in the area of data mining techniques and thus machine learning is constantly evolving. we believe, however, that the links in the GitHub are a good selection of the hugeness of material on the web.

In Section 2, we describe two of the most widely used software Python and; in Section 3 we describe some of the most useful libraries for data mining techniques.

# Open-source Software

## Python Software

Python (<https://www.python.org>) is a high-level, general-purpose programming language. It is a a multi-paradigm programming language, where object-oriented programming and structured programming are fully supported.

The latest Python source distribution is always available from python.org, at <https://www.python.org/downloads/>. The latest development sources can be obtained at <https://github.com/python/cpython/>. The standard documentation for the current stable version of Python is available at <https://docs.python.org/3/>.

### Text editors and integrated development environments (IDEs) available

Here, some of the most widely used IDEs and Text editor for Python.

* **PyCharm (**[**https://www.jetbrains.com/pycharm/**](https://www.jetbrains.com/pycharm/)**)**: PyCharm is a powerful IDE used for programming in Python. It offers code analysis, debugging, integrated testing, and support for web development frameworks, scientific libraries, and more.
* **Visual Studio Code (VS Code,** [**https://code.visualstudio.com**](https://code.visualstudio.com)**):** VS Code is a lightweight and highly customizable code editor by Microsoft. It has a strong Python extension ecosystem, providing features like debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality.
* **Jupyter Notebook / JupyterLab (**[**https://jupyter.org**](https://jupyter.org)**):** Jupyter Notebook and its updated version JupyterLab are interactive environments widely used for data science and scientific computing not only in Python, since Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R. They allow you to combine code, visualizations, and narrative text in a single document.
* **Spyder (**[**https://www.spyder-ide.org**](https://www.spyder-ide.org)**):** Spyder is an IDE designed specifically for scientific computing with Python. It offers features like an integrated IPython console, variable explorer, and support for data visualization libraries. It integrates with a number of prominent packages in the scientific Python stack.
* **Atom (**[**https://github.blog/2022-06-08-sunsetting-atom/**](https://github.blog/2022-06-08-sunsetting-atom/)**):** Atom is a customizable and free text editor developed by GitHub with support for plug-ins written in JavaScript, and embedded Git Control. It can be extended using packages to support Python development.

## R Software

R (<https://www.r-project.org>) is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. It provides a wide variety of statistical techniques (i.e. linear and nonlinear modelling, statistical tests, time-series analysis, classification, clustering, and machine learning algorithms) and graphical techniques, and is highly extensible. R has gained immense popularity in the fields of statistics, data analysis, and data visualization due to its rich set of built-in functions and packages tailored for these tasks. R provides an Open Source route to participation in that activity.

### Text editors and integrated development environments (IDEs) available

* **RStudio (**[**https://posit.co/download/rstudio-desktop/**](https://posit.co/download/rstudio-desktop/)**):** RStudio is a widely-used and feature-rich IDE designed specifically for R programming. It provides an intuitive interface for writing R code, managing packages, and creating visualizations.
* **Visual Studio Code (VS Code):** as mentioned for Python, VS Code has extensions that allow you to work with R. It provides features like syntax highlighting, debugging, and integrated terminal support for R.
* **Jupyter Notebook / JupyterLab:** as mentioned for Python, Jupyter Notebook and JupyterLab support R kernels, enabling you to create interactive documents with R code and visualizations.
* **Tinn-R (**[**https://tinn-r.org/en/**](https://tinn-r.org/en/)**):** Tinn-R is an editor/word processor ASCII/UNICODE generic for the Windows operating system, very well integrated into the R, with R specific features including syntax highlighting, code completion, and integration with R's help system.

# Libraries for data mining techniques

There are several important open-source libraries for data mining techniques for Python and R. First, we briefly describe the methods most used for unsupervised and supervised learning, and then we listed some of the most widely used libraries. For a comprehensive description of the statistical models and machine learning methods mentioned in this section, the reader is referred to the reading of Anderson et al. (1958), Agresti (2015), McCullagh, P. (2019), Hastie, Tibshirani et al. (2021).

## Data manipulation and visualization

As a crucial step of every type of analysis is an efficient manipulation of data and a good visualization of the results, we recall some useful libraries for these steps. For data manipulation, in Python, Pandas, NumPy and scipy are some very important libraries for data manipulation and data analysis. They provide data structures and functions needed to effectively manipulate and analyse structured data. In R, dplyr and data.table are excellent for data manipulation, the latter is strongly efficient in handling large datasets. For data visualization, ggplot2 it is the most widely used package for plotting and visualizing data. It provides helpful commands to create complex plots from data in a data frame and a more programmatic interface for specifying what variables to plot, how they are displayed, and general visual properties. Its counterpart in R is ggplot (ggpy), which provides a high-level interface for creating complex visualizations.

## Unsupervised learning

Algorithms of unsupervised learning have the objective to identify patterns and structures in data and the relationship among the features without a target variable to predict or classify. Common tasks in unsupervised learning include Association rules mining, clustering, where data points are grouped based on similarities, and dimensionality reduction, where the algorithm reduces the complexity of the data while preserving important features. Among dimensionality reduction methods, the most widely used methods are the Principal Component Analysis, Correspondence Analysis and Multiple Correspondence Analysis, Non-linear factorial methods, Multidimensional Scaling Methods.

### Python libraries for unsupervised learning

* scikit-learn: This is one of the most popular libraries for machine learning in Python. Among clustering algorithms, if offers from sklearn.cluster: KMeans, AgglomerativeClustering for hierarchical clustering, and DBSCAN for density-based clustering. Among dimensionality reduction techniques, it offers PCA from sklearn.decomposition for Principal Component Analysis (PCA), t-SNE from sklearn.manifold, t-Distributed Stochastic Neighbor Embedding that is useful for visualizing high-dimensional data by reducing it to a lower-dimensional space. If the work involves multi-label classification tasks, Scikit-multilearn provides options like MLkNN, which combines multi-label k-nearest neighbors and dimensionality reduction.
* Apache Spark: it is a fast and general-purpose cluster computing system that also includes libraries for machine learning (MLlib) and graph processing (GraphX). It's well-suited for big data processing and analysis.
* PyCaret: While it's a high-level machine learning library, PyCaret also provides support for various clustering algorithms, making it easier to experiment with different techniques.
* Yellowbrick: It focuses on visualization tools to understand the effect of dimensionality reduction and tools to help visualize clustering results, along with other aspects of machine learning,
* Prince: This library specializes in factor analysis and dimensionality reduction, offering implementations of various techniques, including PCA and correspondence analysis.

### R libraries for unsupervised learning

* arules: it is focused on association rule mining, a technique that identifies interesting relationships or associations between items in a dataset. This package provides tools for discovering frequent items and generating association rules from transactional data.
* cluster: it provides various clustering algorithms, including hierarchical clustering, K-Means clustering, and more advanced techniques like PAM (Partitioning Around Medoids) and CLARA (Clustering Large Applications).
* dbscan: it provides an implementation of the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, which is effective for identifying clusters of varying shapes and sizes.
* mclust: it is focused on model-based clustering, and it offers tools for fitting Gaussian Mixture Models to the data, allowing the identification of clusters with different shapes and orientations.
* clustermesh: it offers tools for hierarchical clustering using dendrograms, including the possibility of creating clustered heatmaps and visualizing the results.
* dendextend: it is a versatile package for extending the functionality of dendrograms, providing additional options for manipulating and visualizing hierarchical clustering results.
* NbClust: it is designed to determine the number of clusters in a dataset, and it offers a comprehensive suite of indices for cluster validation
* factoextra: it is a helpful tool for extracting and visualizing information from various multivariate analysis methods. This package provides functions to visualize the results of clustering algorithms, including hierarchical clustering and K-Means, and includes plotting cluster assignments, dendrograms, and more. It offers tools to visualize the outcomes of dimensionality reduction techniques like PCA (Principal Component Analysis) and t-SNE (t-Distributed Stochastic Neighbor Embedding), to visualize the factor analysis results, including factor loadings and factor correlations, and it provides heatmap functions that can display clustering results and PCA outcomes.
* FactoMineR: this comprehensive package offers a range of multivariate analysis techniques, including Principal Component Analysis (PCA), Multiple Correspondence Analysis (MCA), and Factor Analysis of Mixed Data (FAMD).

## Supervised learning

In supervised learning, the algorithm is provided with input data (the set of explanatory variables) and corresponding target outputs (the set of response variables), allowing it to learn how to map inputs to desired outputs. When the target is numerical, we deal with a regression problem, instead when the target is categorical with a set of response classes, we deal with a classification problem. The goals of the models of supervised learning are to identify the best model capable of understanding how the response variable is related to the explanatory variables and to make accurate predictions. Among the supervised learning models, we recall linear regression, logistic regression and discriminant analysis. With high dimensional data, we recall LASSO, RIDGE, ELASTIC-NET Regression, or dimension reduction methods such as Principal Component Regression or Partial Least Squares Regression., we Among nonparametric models we recall Regression Splines, Smoothing Splines, and Local Regression. Among algorithms based on decision trees the most popular are Classification and Regression Trees, Boosting, Random Forests and Bayesian Additive Regression Trees (BART). Finally, we recall Deep Learning with Neural Networks.

### Python libraries for supervised learning

* statsmodels: it is a library for statistical modeling and hypothesis testing, which can be useful for supervised learning when interpretability and inference are important.
* scikit-learn: this is one of the most popular libraries for machine learning in Python. It covers a wide range of supervised learning algorithms, including linear and logistic regression, penalized regression, PCA, decision trees, random forests, support vector machines, and more. It also provides various tools for model fitting, data pre-processing, model selection, model evaluation, and many other utilities.
* XGBoost: Short for "Extreme Gradient Boosting" it is a powerful machine learning algorithm known for its speed and performance.
* LightGBM: it is another gradient-boosting framework that is designed to be efficient and scalable. It's particularly useful for large datasets and is known for its speed and accuracy.
* CatBoost: A gradient boosting library that's capable of handling categorical features effectively without extensive preprocessing.
* TensorFlow: it is an open-source deep learning library that provides a flexible framework for building and training various neural network models. It's particularly well-suited for tasks involving neural networks and deep learning.
* PyTorch: it is another popular deep learning library. It's known for its dynamic computation graph and is widely used for research in machine learning and artificial intelligence.

### R libraries for supervised learning

* stats package is one of the core packages in R and comes pre-installed with R itself. It provides a wide range of statistical functions, including basic statistical tests, linear and nonlinear modelling, regression, and much more.
* kknn: it implements k-nearest neighbours (k-NN) algorithms, which is particularly useful for classification tasks based on similarity metrics.
* glmnet: This package is particularly popular for fitting Lasso and Ridge regression models. It offers efficient implementations of these regularization techniques.
* rpart and tree: these two packages are the most widely used for fitting classification and regression trees.
* party: it implements recursive partitioning methods, including the C4.5 algorithm and conditional inference trees, which can be used for decision tree-based models.
* caret: it is a versatile and powerful tool for training, tuning, and evaluating machine learning models. It provides a unified interface for working with a wide range of algorithms, preprocessing techniques, and model evaluation methods. It supports parallel processing, which can significantly speed up the training and tuning process.
* randomForest: it implements the random forest algorithm, an ensemble learning method that's effective for classification and regression tasks. It's known for its robustness and ability to handle high-dimensional data.
* ranger: it is a fast and efficient implementation of random forest algorithm.
* xgboost: it offers an implementation of the extreme gradient boosting algorithm, which is highly efficient and performs well on a variety of machine learning tasks.
* gbm: The gbm (Generalized Boosted Regression Models) package provides an implementation of gradient boosting, a machine learning technique for regression and classification tasks.
* nnet: it is used for training and evaluating neural networks. While more specialized deep learning frameworks are available (i.e. TensorFlow, Keras of Python), nnet offers a simpler approach for basic neural network tasks.
* e1071: it provides a collection of functions for support vector machines (SVMs), as well as other machine learning algorithms for classification and regression.

## How to install packages In Python and R

Essentially, the installation of the packages depends on the operating system of the PC. In Python, the installation can be done by opening the Command Prompt or terminal and use the *pip install* command followed by the package name. The detailed instruction can be founded at <https://packaging.python.org/installing/>. In R, the installation of packages is straightforward and can be done using the *install.packages()* function. More detailed instruction can be found at <https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages>.

# References

Anderson, T. W., Anderson, T. W., Anderson, T. W., & Anderson, T. W. (1958). An introduction to multivariate statistical analysis (Vol. 2, pp. 1468-1482). New York: Wiley.

Agresti, A. (2015). Foundations of linear and generalized linear models. John Wiley & Sons.

McCullagh, P. (2019). Generalized linear models. Routledge.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning. Springer, New York, NY.