



## DELIVERABLE 4.6

### AgriBIT data services

Project Acronym	AgriBIT
Project Title	Artificial intelligence applied to pPrecision farming By the use of GNSS and Integrated Technologies
Grant Agreement number	101004259
Call	SU-SPACE-EGNSS-3
Funding Scheme	Innovation Action (IA)
Project duration	36 Months

Document Information			
Work Package:	WP4	Task:	T4.4
Due Date:	28/02/2023		
Version:	1.0	Status:	Final
Dissemination level:	PUBLIC		
Type	Other		
Lead Partner:	ENG		
Contributors:	RFSAT, AGROAPPS, AGENSO.		
Keywords:	Infrastructure design, Machine learning, Service composition, Data ingestions		
Abstract:	This document describes the main services available through the AgriBIT Big Data Analytics (BDA) platform. BDA platform provide fast prototyping, deployment, execution and monitoring of both stream and batch AI-based big data analytics (BDA) applications to support the analysis of the data acquire in AgriBIT project.		

Document History			
Version	Date	Contributor(s)	Description
V0.1	02/12/2022	ENG	Initial Version
V0.2	19/12/2022	ENG	General overview added
V0.3	20/01/2023	ENG	BDA Services added
V0.4	20/02/2023	ENG	Complete version for team review
V0.5	02/03/2023	ENG	Ready for internal peer review
0.8	08/03/2023	RFSAT / ACP	Peer-review corrections
V1.0	10/03/2023	ENG	Ready for submission to EC

Document Authors	
ENG	Piero Scrima <a href="mailto:piero.scrima@eng.it">piero.scrima@eng.it</a>
ENG	Giuseppe Vella <a href="mailto:giuseppe.vella@eng.it">giuseppe.vella@eng.it</a>

Document Internal Reviewers	
ACP	Traianos Terzis <a href="mailto:easgiannitsa.sitira@gmail.com">easgiannitsa.sitira@gmail.com</a>
RFSAT	Artur Krukowski <a href="mailto:artur.krukowski@rfsat.com">artur.krukowski@rfsat.com</a>

## DISCLAIMER

This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of its content. This document may contain material, which is the copyright of certain AgriBIT consortium parties, and may not be reproduced or copied without permission. All AgriBIT consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the AgriBIT consortium as a whole, nor a certain party of the AgriBIT consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and does not accept any liability for loss or damage suffered by any person using this information.

## ACKNOWLEDGEMENT



This project has received funding from the European Union Agency for the Space Programme under the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004259.

## Executive Summary

This document describes the main services available through the AgriBIT Big Data Analytics (BDA) platform. BDA platform provide fast prototyping, deployment, execution and monitoring of both stream and batch AI-based big data analytics (BDA) applications to support the analysis of the data acquire in AgriBIT project.

In Section 2 a presentation of the BDA platform is given, describing the general objective in the context of the project and introducing BDA services (i.e., Machine Learning algorithms). Section 3 firstly provides an overview of the main elements of the system, describing the workflow and its purpose. Then the section 3.1, focus on the main components of the system.

In the section 3.2 the most important functionalities are presented. It's explained how to navigate among available BDA applications (3.3) and how to create a new one (3.3.2).

The rest of section 2 explains how to create machine learning model and how to visualize data obtained by running the BDA application.

Finally, the section 4 presents and lists all the most important BDA services included in the ALIDA platform providing to the user a catalogue to consult in order to explore the available services and their requirements. Each service is presented by a table with a brief description. Apart from the general information the table contains the list of the attribute to configure in order to run each service and the related Machine Learning algorithm.

Finally, conclusions are presented describing the status of the platform and its future development.

# 1. Contents

Executive Summary .....	4
List of Figures .....	6
List of Tables .....	7
2. Introduction .....	8
3. General overview .....	10
3.1. Architecture.....	11
3.2. ALIDA model lifecycle.....	13
3.3. ALIDA Web graphical Interface .....	14
3.3.1. BDA application.....	15
3.3.2. Application Designer .....	16
3.3.3. Application Detail.....	17
3.3.4. Machine learning model .....	22
3.3.5. Visualization .....	23
3.3.6. BDA Services .....	24
4. BDA services.....	26
4.1. Machine learning algorithms categorization .....	26
4.2. BDA Services analysis .....	27
5. BDA Services Catalogue .....	29
5.1. Decision Tree .....	29
5.1.1. Decision Tree Classifier Model.....	29
5.1.2. Decision Tree Classifier Predict.....	32
5.1.3. Decision Tree Regressor Model .....	36
5.1.4. Decision Tree Regressor Predict .....	41
5.2. Gradient Boosted .....	44
5.2.1. Gradient Boosted Tree Classifier Model.....	44
5.2.2. Gradient Boosted Tree Classifier Predict.....	47
5.2.3. Gradient Boosted Tree Regressor Model .....	50
5.2.4. Gradient Boosted Tree Regressor Predict .....	53
5.3. Isolation Forest.....	56
5.3.1. Isolation Forest Model .....	56
5.3.2. Isolation Forest Predict .....	59
5.4. Random Forest .....	61
5.4.1. Random Forest Classifier Model .....	61
5.4.2. Random Forest Classifier Predict .....	63

5.4.3.	Random Forest Regressor Model .....	66
5.4.4.	Random Forest Regressor Predict .....	68
5.5.	Gaussian Mixture.....	71
5.5.1.	Gaussian Mixture Model.....	71
5.5.2.	Gaussian Mixture Predict.....	73
5.6.	Kmeans .....	76
5.6.1.	Kmeans Model .....	76
5.6.2.	Kmeans Predict .....	78
5.7.	Linear Regression .....	81
5.7.1.	Linear Regression Model.....	81
5.7.2.	Linear Regression Predict.....	83
5.8.	Linear Support Vector Machine .....	86
5.9.	Logistic Regression .....	89
5.10.	Min Max Fit .....	92
5.11.	Min Max Scaler Process.....	95
5.12.	Multilayer Perceptron .....	97
5.12.1.	Multilayer Perceptron Classifier Model .....	97
5.12.2.	Multilayer Perceptron Classifier Predict.....	100
5.13.	Naïve Bayes .....	103
5.13.1.	Naïve Bayes Model .....	103
5.13.2.	Naïve Bayes Predict .....	105
5.14.	Principal Component Analysis.....	108
5.14.1.	Principal Component Analysis Model.....	108
5.14.2.	Principal Component Analysis Predict.....	110
6.	Conclusions .....	112
	List of Abbreviations .....	113

## List of Figures

Figure 1 - ALIDA ecosystem .....	9
Figure 2 - ALIDA dashboard .....	10
Figure 3 - ALIDA Architecture .....	12
Figure 4 - ALIDA model lifecycle .....	14
Figure 5 - BDA Applications list.....	15
Figure 6 - Application Designer - Choose Application Mode.....	16
Figure 7 - Application Designer - Compose Workflow.....	17
Figure 8 - Batch Application.....	18
Figure 9 - Application detail modal.....	19

Figure 10 - Dataset details .....	19
Figure 11 - Model detail.....	20
Figure 12- Application schedule .....	21
Figure 13 - Streaming Application Detail .....	22
Figure 14 - Machine learning model.....	22
Figure 15 - Visualization page .....	23
Figure 16 - Chart selection .....	24
Figure 17 - Attribute Mapping .....	24
Figure 18 - BDA Services list.....	25

## List of Tables

Table 1 - Decision Tree Classifier Model.....	31
Table 2 - Decision Tree Classifier Predict.....	35
Table 3 - Decision Tree Regressor Model .....	40
Table 4 - Decision Tree Regressor Predict .....	43
Table 5 - Gradient Boosted Tree Classifier Model.....	46
Table 6 - Gradient Boosted Tree Classifier Predict .....	49
Table 7 - Gradient Boosted Tree Regressor Model .....	52
Table 8 - Gradient Boosted Tree Regressor Predict .....	55
Table 9 - Isolation Forest Model .....	58
Table 10 - Isolation Forest Predict .....	60
Table 11 - Random Forest Classifier Model .....	63
Table 12 - Random Forest Classifier Predict .....	65
Table 13 - Random Forest Regressor Model .....	67
Table 14 - Random Forest Regressor Predict .....	70
Table 15 - Gaussian Mixture Model.....	72
Table 16 - Gaussian Mixture Predict.....	75
Table 17 - Kmeans Model .....	77
Table 18 - Kmeans Predict .....	80
Table 19 - Linear Regression Model .....	82
Table 20 - Linear Regression Predict .....	85
Table 21 - Linear Support Vector Machine.....	88
Table 22 - Logistic Regression.....	91
Table 23 - Min Max Fit .....	94
Table 24 - Min Max Scaler Process .....	96
Table 25 - Multilayer Perceptron Classifier Model .....	99
Table 26 - Multilayer Perceptron Classifier Predict .....	102
Table 27 - Naïve Bayes Model .....	104
Table 28 - Naïve Bayes Predict.....	107
Table 29 - Principal Component Analysis Model .....	109
Table 30 - Principal Component Analysis Predict .....	111

## 2. Introduction

Nowadays most of the activities that are performed during the day generate a large amount of data. Data are collected through IT tools that support most of the production activities, but also by personal devices and sensory detection systems, which are increasingly cheaper and more accessible. The result of this big availability of data sources is the increasing volume of data produced. In addition to the volume, the availability of different devices and applications also produces a diversity in the data structure of the data itself, i.e. the data variety. Furthermore, the data are continuously collected, producing an ever-increasing velocity in their acquisition. These characteristics, Volume, Variety, Velocity, are the so called 3 Vs, and are the essential characteristics of what are defined as Big Data.

Big data are considered as a fundamental resource to support many activities. From the information extracted by big data analysis we can obtain great advantages, for example to predict trends, models, behaviours of both natural and social processes. The complexity of big data analysis, however, requires adequate means and methods. The techniques, and methods for analysing big data and extracting information are called big data analytics.

The AgriBIT project aims to increase the efficiency of the agriculture activities through ICT support, apart from the development of Precision Agriculture services. It aims also to provide third party users with tools that can extend the technological support of agricultural activities. For this reason, AgriBIT project will deliver APIs (Application Program Interfaces) to make the data produced available to third-party users, which will be able to use them to enrich the offer of services. In addition to the raw data, however, the project also provides a big data analytics platform to support the analysis of the data acquired.

As mentioned, raw data can be elaborated to obtain important information that can offer a competitive advantage to the end user. However, this type of activity is not simple, and to execute it third parties need not only the skills of a data analyst but, most important, also a valid infrastructure to manage and process complex data such as those acquired in an agricultural production context. For this reason, the project furnishes a platform in order to process data acquired in the fields, providing to third parties data analyst a tool to create BDA (Big Data Analytics) application deployed on the cloud, extract information and advice farmers. The big data analytics system supplied within AgriBIT is called ALIDA.

ALIDA<sup>1</sup> is a Big data Analytics system powered by Engineering SPA which provides fast prototyping, deployment, execution and monitoring of both stream and batch AI-based BDA applications.

ALIDA allows data processing with modern machine learning and data mining algorithms.

As can be seen in Figure 1, ALIDA can interface with various big data storage technologies and streaming technologies. The data acquisition and integration process will be discussed and deepened in task T5.2, this deliverable will focus on the heart of ALIDA which are its big data analytics services.

A general overview of ALIDA and its functions is given in deliverables D4.4 and D4.5, for completeness also in this document the main functions of the system will be presented in section 3, later in section 4 all the services and algorithms currently present in the system will be illustrated. These algorithms can be extended with further services by anyone wishing to enrich the system and add functionalities. To add a service, it is necessary to use the APIs for BDA services which will be released in deliverable D5.7. Moreover, Task T5.2 (AgriBIT Integrated Services Development), will provide data integration

---

<sup>1</sup> <https://www.eng.it/en/case-studies/alida-per-migliorare-la-flessibilita-reattivita-del-business>



methods so that third party developers can leverage AgriBIT data in BDA applications, data integration will be released in D5.6. The rest of this document describes the ALIDA platform and its BDA services which are the fundamentals blocks of BDA.

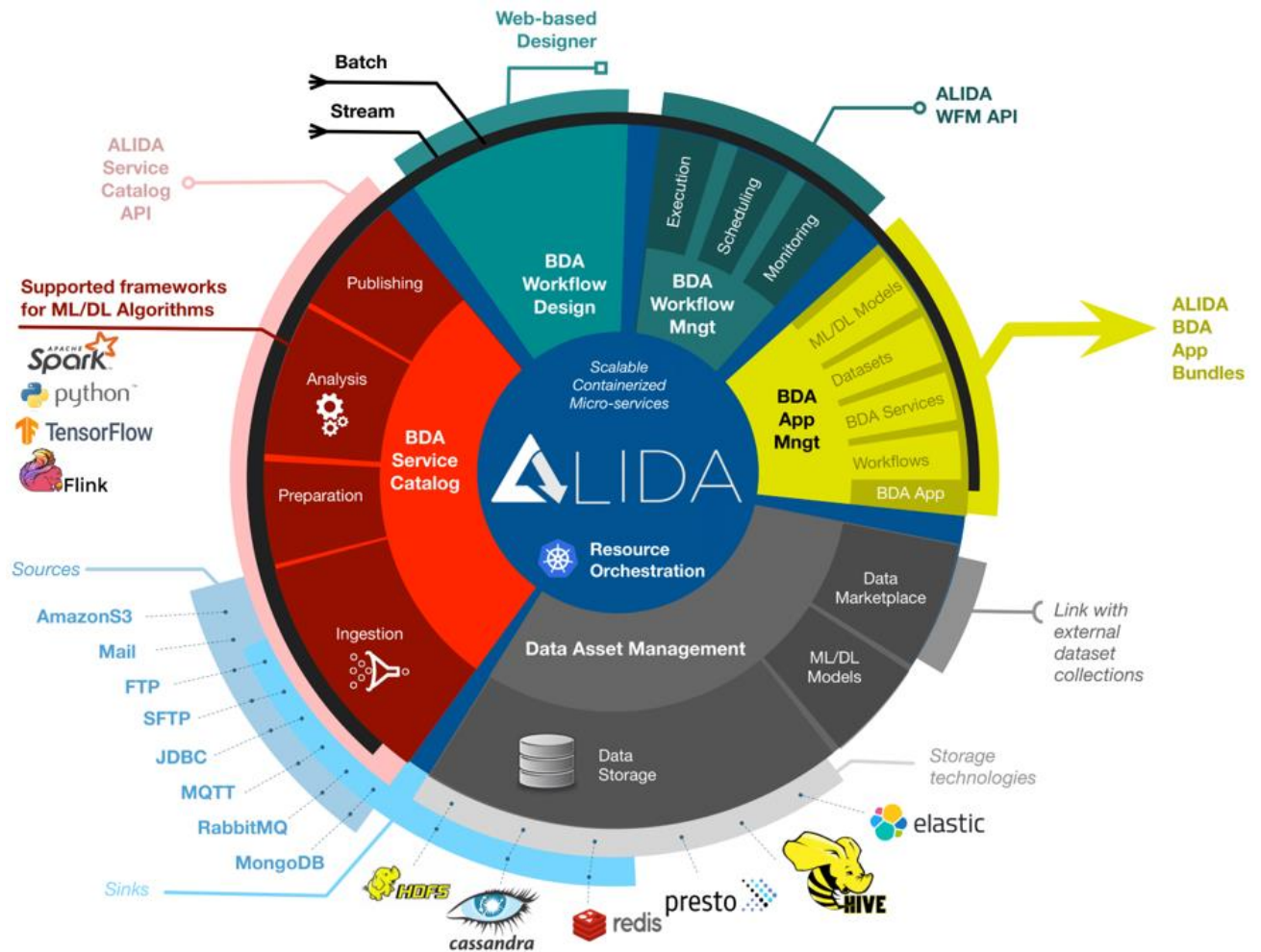


Figure 1 - ALIDA ecosystem

### 3. General overview

The analytics performed within the ALIDA Platform are based on machine learning, a branch of Artificial Intelligence (AI) that apply algorithms able to improve and "learn" automatically during their execution, iteration by iteration, machine learning algorithms are discussed in section 4.1.

In ALIDA, algorithms are embedded in standalone processors that can be joined together to compose a workflow, a pipeline of blocks of code executed in sequence.

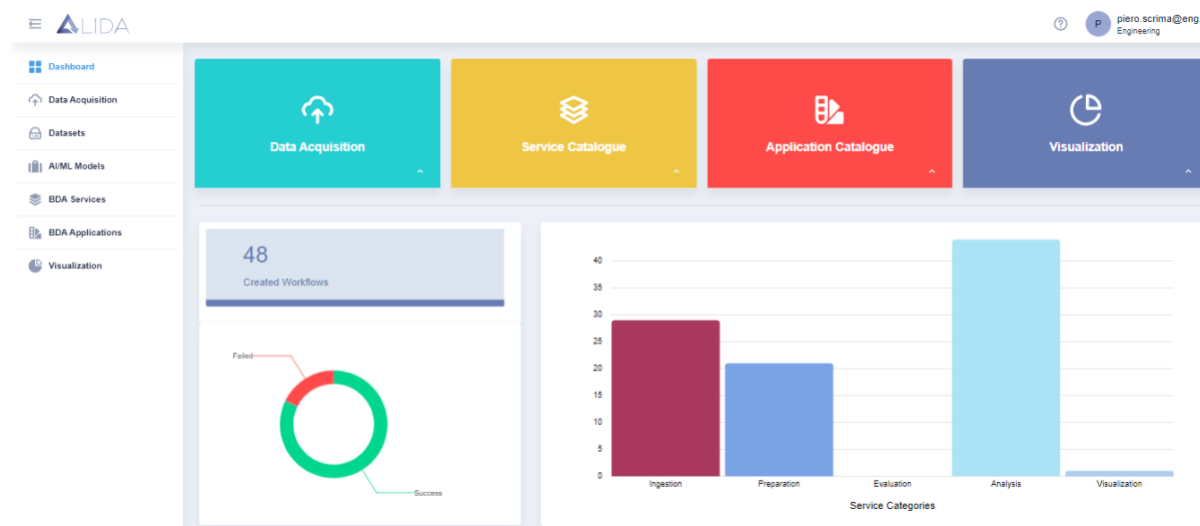


Figure 2 - ALIDA dashboard

A workflow is a system of connected processors that elaborate a data stream and that made up a BDA Application. The result of the workflow depends on the processors and on their configuration. In ALIDA, these processors are called BDA-services. ALIDA contains a catalogue of BDA-services, which can be divided into batch-type and streaming-type. Each of these types can be used in their respective types of workflows: batch-type workflows and streaming-type workflows.

Streaming-type BDA-services are divided into three types of streaming applications:

- Source: a streaming application that takes data from a data source and loads it into a channel.
- Processor: a streaming application that takes data from a channel, processes it (mostly applying machine learning algorithms) and finally loads it in another channel.
- Sink: a streaming application that takes data from a channel and stores it on a target storage.

A streaming workflow, therefore, is a process that is always running (and can be stopped by the user at any time), in which the data follows the flow of the graph drawn by the user.

ALIDA provides a web graphic interface that enable user to manage BDA Applications. The interface consists of a menu on the left, with the main sections, and a central area where the functions of each section are available. The sections are: Dashboard, AI-ML models, BDA-services, BDA Application and Visualization.

The current chapter explore first the architecture of the system, then provide the information in order to list, create and manage BDA applications by means of the ALIDA web graphic interface.

### 3.1. Architecture

ALIDA is based on a microservices architecture, in which each functionality is implemented on different components each of which runs on a container. A container is, technically speaking, a special software process that runs on an operating system and that is isolated from all the others processes, thanks to this isolation the container is therefore particularly safe. In practice the container is used as a software block that allows the virtualization of an operating system and which contains within it all the dependencies necessary for its execution which moreover takes place in isolated mode. BDA services are also distributed using containers, each BDA service it's a containerized OCI-compliant<sup>2</sup> (a well spread container standard) micro-service application. Since containers are self-consistent monolithic blocks, they can be started duplicated and stopped very easily. The container system is therefore well suited to redundant, resilient and scalable cloud architecture. To allow containers to be easily managed, special software called orchestrators is used. ALIDA uses an orchestrator called Kubernetes<sup>3</sup>.

The use of containers and Kubernetes allow ALIDA to be a scalable system, i.e. to be able to use an arbitrary number of resources by duplicating its components and making them work in parallel. For example, if an algorithm contained in a BDA service requires the use of more resources, the system can be requested to instantiate more instances of that BDA service. In fact, the *"spark.executor.instances"* parallelization parameter is present in the BDA services, which allows us to instantiate an adequate number of instances of the component (an example cab be see in Table 2 - Decision Tree Classifier Predict).

Another advantage of containerization is that each element is a standardized module, so each new block that extends the system can also be seen as a simple package, distributed using a so-called image, a static version of a container. This allows for the potential extensibility of BDA services, which are containers. The implementation of the APIs interface to allow the creation of containers that embed new BDA services will be addressed in the deliverable D5.6.

Apart from Kubernetes ALIDA uses other open source technologies:

- Argo Workflows<sup>4</sup>: an open source container-native workflow engine for orchestrating parallel jobs on Kubernetes;
- Apache Kafka<sup>5</sup>: an open-source distributed event streaming platform;
- React Flow<sup>6</sup>: a component on which the graphic designer is based in order to design pipelines of BDA services.
- MinIO<sup>7</sup>: an High Performance Object Storage released under GNU Affero General Public License v3.0. It is API compatible with the Amazon S3 cloud storage service.

These components are the basis of the ALIDA software architecture shown in Figure 3.

<sup>2</sup> <https://opencontainers.org/>

<sup>3</sup> <https://kubernetes.io/>

<sup>4</sup> <https://argoproj.github.io/argo-workflows/>

<sup>5</sup> <https://kafka.apache.org/>

<sup>6</sup> <https://reactflow.dev/>

<sup>7</sup> <https://min.io/>

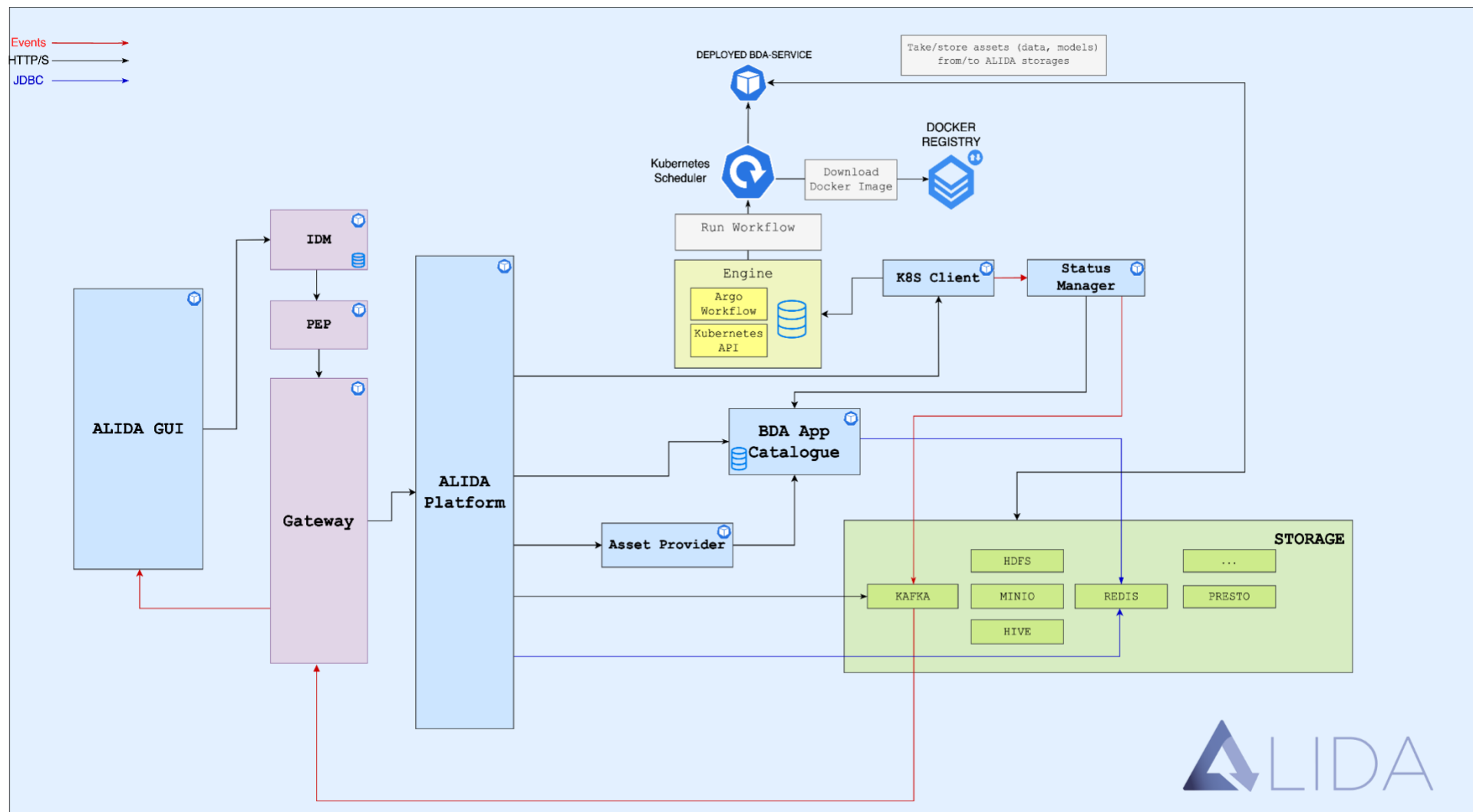


Figure 3 - ALIDA Architecture

The main components of ALIDA shown in Figure 3 are:

- *ALIDA GUI*: the graphical user interface of ALIDA, based on React<sup>8</sup>.
- *ALIDA Platform*: the main ALIDA back-end service, responsible for managing all requests coming from the GUI and others components;
- *BDA Application Catalogue*: contains all operations (creation, reading, updating, deletion) related to the objects managed in ALIDA: datasets and models metadata, BDA services, BDA applications, and more;
- *Asset Provider*: component of ALIDA responsible for parsing the BDA application pipeline generated through the ALIDA graphical designer and providing assets such as exporting the BDA application and the generated model;
- *K8S Client*: component for executing/stopping BDA applications workflows and sending BDA application events that are executed in ALIDA, interacting with Kubernetes and Argo Workflow APIs;
- *Status Manager*: based on incoming status events, manages and updates the states of the BDA applications by interfacing with the *BDA Catalogue*.

### 3.2. ALIDA model lifecycle

The process for using ALIDA follows a well-defined and intuitive flow, described in Figure 5.

The first step to run a BDA Application is to upload the data. The data can be uploaded to the platform using the ALIDA client application.

The ALIDA client application can be download from the Datasets section on the ALIDA web graphical interface. The downloaded app contains many sections to be compiled:

- ALIDA URL: the URL of the ALIDA platform
- username of the ALIDA account
- password of the ALIDA account
- Access level of the dataset (the visibility level).
- The local absolute path of the dataset's folder
- A name that identifies the dataset uniquely. Name can include only letters, numbers, dashes and underscore, avoid spaces. Note that if another dataset exists with the same name, an error message will be prompt. Example of valid name: my-dataset-1

After all the fields are filled, user can click on Upload. The progress bar will show the upload progress. Note that the speed might depend on the number of files, on the size of the dataset and on the internet connection speed.

After the files have been uploaded, the dataset needs to be registered in ALIDA in order to be used inside a workflow.

- Go to the Datasets section, then click on Register Dataset on the upper-right corner.
- Choose a dataset name (this time spaces are allowed)
- An access level
  - Private (Only you will be able to use the dataset)
  - Team (All members all your team will be able to use it)

---

<sup>8</sup> <https://it.reactjs.org/>

- Public (Everyone inside ALIDA will be able to use the dataset)
- Inside Data Source section choose ALIDA-MINIO
- Choose the dataset based on the name you gave it when uploading it with the client application.
- Add a description
- Edit the columns types if necessary
- Click on save

At the end of this process the dataset will be available to be used in a BDA application.

The next step is to create a workflow in a BDA Application, which is explained in 3.3.2, and where the data uploaded are available to be elaborated. User can create a workflow using the different BDA services available in the platform, once the workflow is created and connected to the dataset the BDA application can be run to evaluate results.

Based on the type of workflow created by the application, the application can produce two types of output, a final result ready to be delivered and exploited or a data model. A machine learning algorithm is a learning process. So generally, to run a machine learning algorithm three stages are performed where one is learning. In the first phase a set of data, called training set data, are provided to the algorithm that "learns", this knowledge is stored in a data model. The second phase of the algorithm is the "Evaluate your results" testing phase (Figure 4), in which the algorithm is evaluated to understand its performance. Finally, the algorithm is run to produce the final result or prediction and to "Deliver and exploit the output".

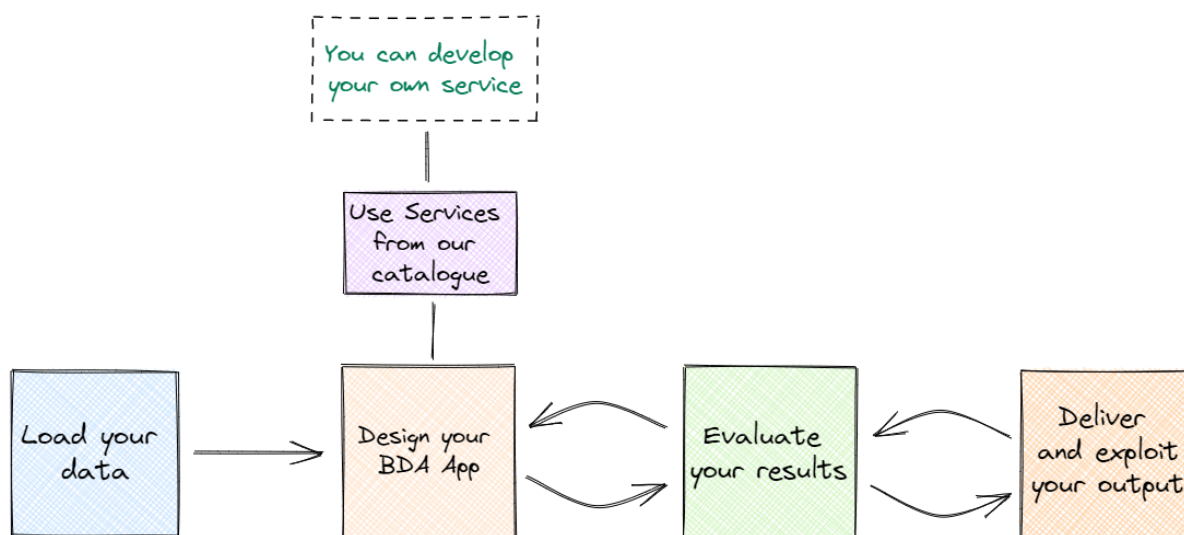


Figure 4 - ALIDA model lifecycle

### 3.3. ALIDA Web graphical Interface

In this section the main pages of the ALIDA web interface are described. It is important to consider that the graphics and in general the look and feel of the interfaces could change respect to the image present in this document, as the platform is currently maintained and subject to continuous evolution for its improvement. However, the main sections, functions and their grouping will remain essentially the same, so the images and descriptions in this chapter will always be effective.

The chapter takes into consideration the main pages of the system:

- BDA application
- BDA Services
- Visualization
- AI/ML Models
- Application designer
- Application details

### 3.3.1. BDA application

The BDA Applications are the heart of the ALIDA. Basically, BDA Applications are bundles of three main elements:

- Workflow – the analytics core, a sequence of machine learning algorithms.
- Assets – datasets and machine learning models given as inputs or generated by the algorithms.
- Visualization Configurations – all chart-related options and preferences saved by the user during the visualization of application results, virtual or public datasets.

In BDA Application section it is possible to manage workflows: create, modify and delete them. As shown in Figure 5, the page contains two main parts, the navigation bar which consist of the two rows at the top of the page, and the applications list. The navigation bar makes the browsing thought the applications easier by enable filtering and search: there is a search input (1) that allows user to find a precise application and two selects, one to filter BATCH and STREAMING applications (2) and the other to filter applications based on their status (3).

Below the search and filter bar user can find the action bar that contains the title of the page and the actions user can perform: (4) a big green button lets the user add a new application and, on the right, two buttons (5) let user import an external application or change the sorting logic.

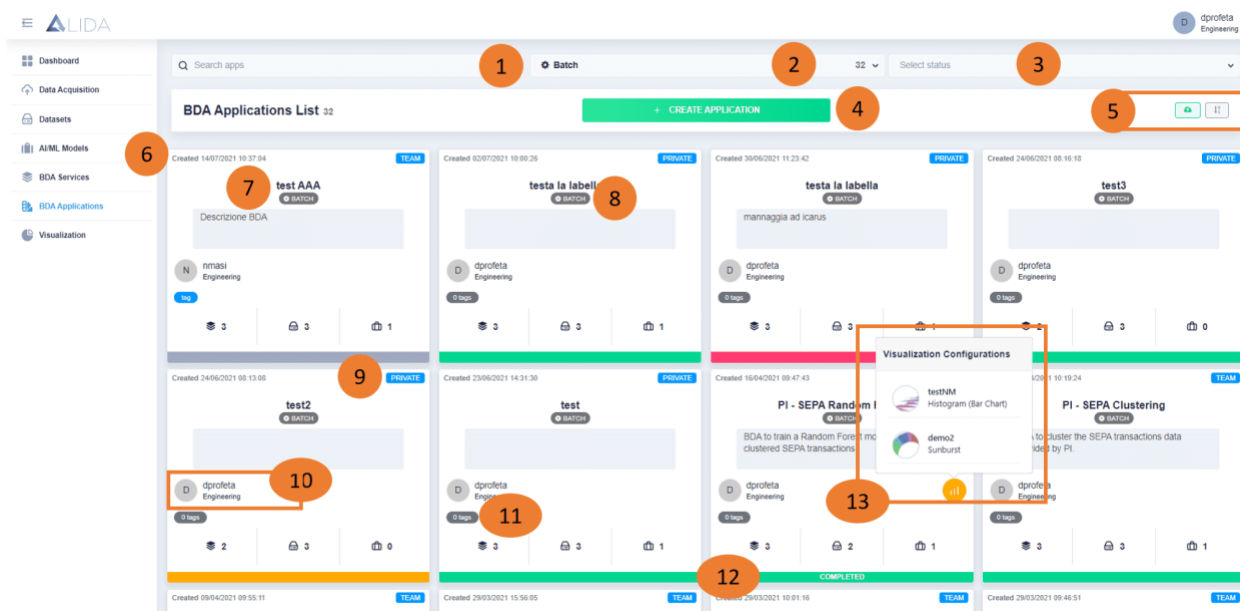


Figure 5 - BDA Applications list



Under these bars, the BDA Applications available are presented in form of cards, Each of them provided with a selection of the most relevant information to have an overall view at first glance; on top, user can read the creation date and time (6) together with the application privacy level (9).

The main info displayed in the middle are the application title and description (7) as well as the application mode (8). In addition to this, in the application cards is also specified the corresponding owner user and organization (10), together with the application tags (11), if any.

Below the tags, a set of numbers indicating how many algorithms the application includes and how many datasets and models are used or produced by its algorithms' workflow.

In some applications, user could also notice an orange circle that, when clicked, triggers a context menu (13): that's the list of visualization configurations stored in the application bundle.

On the lower side of the cards you will notice a thick coloured bar, that is the application current status (7). Application Status can take the following values:

- READY (slate grey): the application has been created and has never been executed before.
- RUNNING (yellow): application execution is currently in progress.
- ERROR (red): some error occurred during the last application execution.
- COMPLETE (green): the last application execution completed successfully.

### 3.3.2. Application Designer

At the top right of the navigation bar of the BDA application page (Figure 5) there is the button create application (4), clicking this button the workflow designed will be opened. After the click of the “create application” button the user will be asked to choose between a STREAMING or a BATCH application (Figure 6).

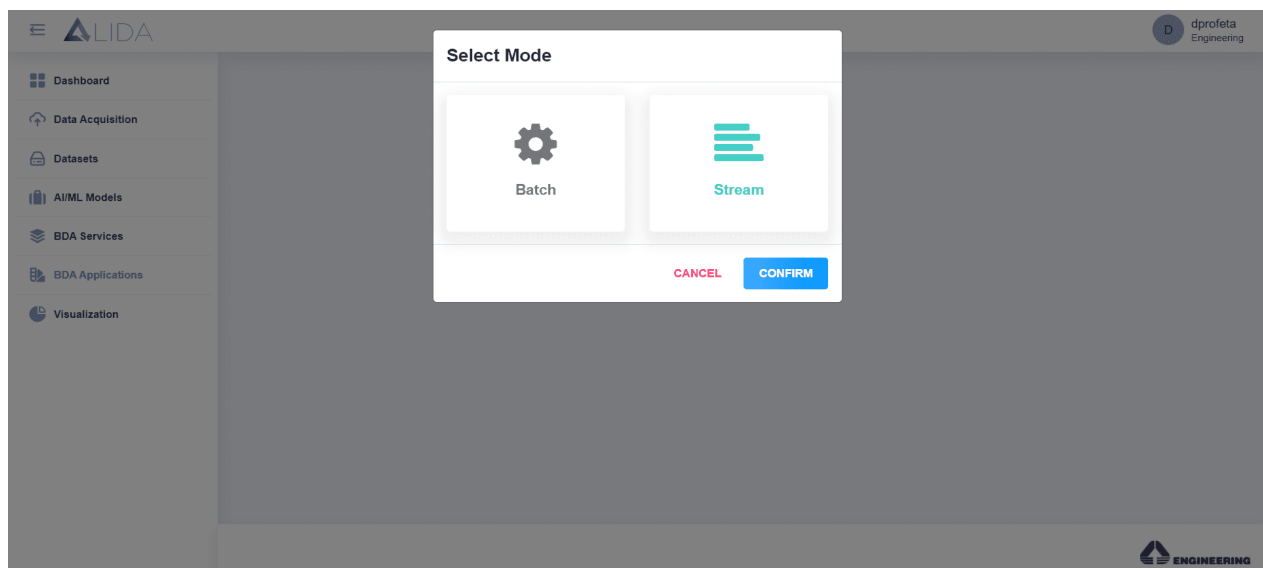


Figure 6 - Application Designer - Choose Application Mode

Selecting the application type the application designer will be shown (Figure 7).



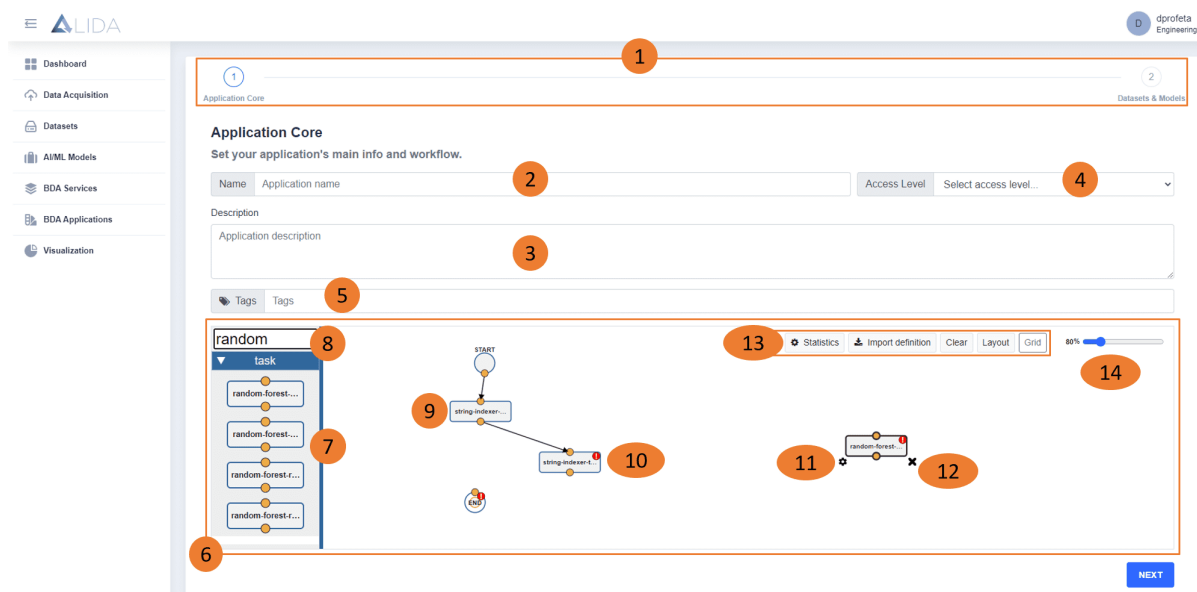


Figure 7 - Application Designer - Compose Workflow

The first step of Application Designer is depicted in Figure 7. First of all, user should choose the new application name (2) and privacy level (3) and optionally provide a description of its purpose, use case and/or composition (4). User can also add some tags to make the application browsing and management easier (5).

To compose the application workflow the Application Designer is provided with an interactive canvas (6) where user can drag blocks around and link them together in sequence, following the timeline of their execution and joining the arrows in their input/output ports (9). On the left (7) there are several BDA services which runs Machine Learning algorithms, BDA services can be chosen based on the analysis user want to perform. To check the available services go to section 5.

To compose a valid workflow user must create a pipeline of algorithms (BDA services) from the START point to the END, otherwise user may see some red alerts indicating that something is wrong (10). User can also click on one of the chosen algorithms to select it, thus enabling two additional buttons on its lower sides, one gear to configure the algorithms properties (11) and one to remove it (12).

The designer is provided with some utility buttons to empty the canvas and start over, fix the workflow layout or toggle a background grid (13), as well as a slider to adjust the zoom level (14).

### 3.3.3. Application Detail

Clicking on an application card from the Applications List, user can access the Application Detail. Since application features and components vary based on the application mode, either BATCH or STREAMING, the Application Detail looks slightly different in the two cases.

BATCH Application page is divided in panels, based on the application main components (Figure 8).

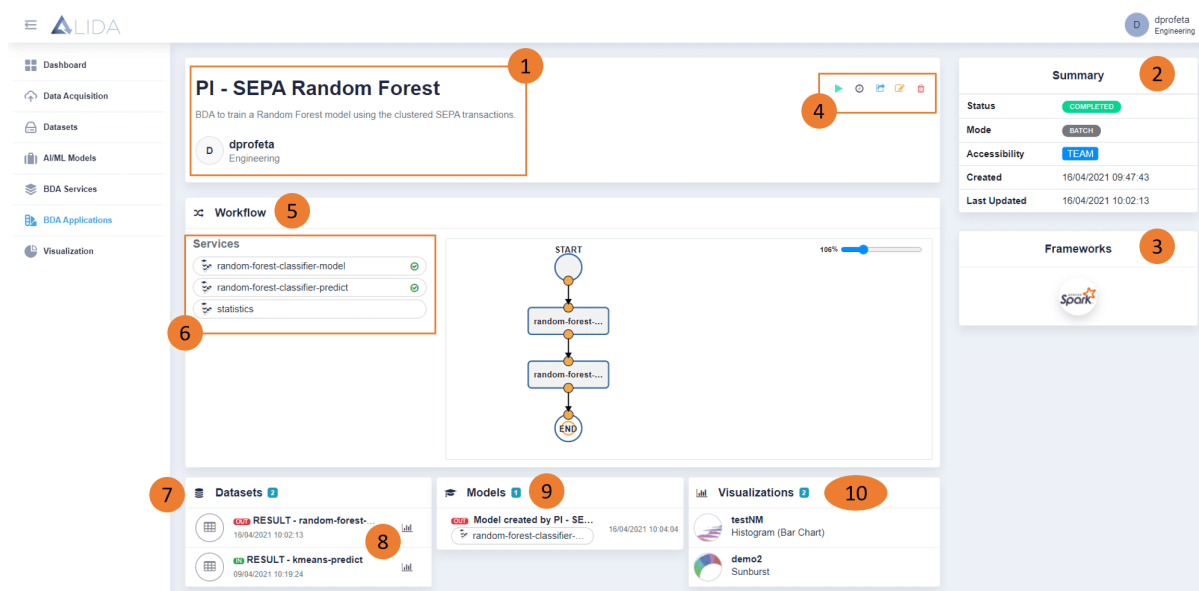


Figure 8 - Batch Application

The panel on top includes the basic information of the application, i.e. name, description and owner (1). On the right, user can find the application summary: listing status, create and update dates and the application access level (2); the frameworks section right below (3) containing the images of the frameworks the workflow algorithms are developed with.

By the application action buttons user can: run the application, edit it, access its schedules, export the application, delete it. All these actions are grouped on the top right of the first box (4):

- Application Run - user can run applications whose status is READY or COMPLETE. When user clicks on the "play" button an alert is shown informing that the application has been started successfully and the status will change accordingly, or a warning will show up in case of error.
- Application Schedules - BATCH application executions can be scheduled and become cron<sup>9</sup> jobs; user can set the run for a specific date and time or can make it periodic.
- Application Export – application can be exported and acquired in difference instances of ALIDA.
- Application Edit - user can edit all applications no matter their status. The edit button redirects the user to the Application Designer, containing the application ready to be edit.
- Application Delete – application can be deleted, after the delete of the application user will be redirected to the application list.

Moving down from the information panel user can find the rest of the panels (where the number in bracket represent the reference position in Figure 8.):

- Workflow (5)
- Datasets (7)
- Model (9)
- Visualizations (10)

<sup>9</sup> The **Cron** is a Linux system utility that allows activities (jobs) to be executed automatically by the system at specified intervals. The cron utility defines a format to specify the frequency of executions, this format is very popular and is used in Alida to configure application scheduling. <https://pubs.opengroup.org/onlinepubs/007904975/utilities/crontab.html>

The Workflow panel contains a read-only graph representing the sequence of algorithms used to compose the workflow. On the left, the algorithms are listed in sequence (6) and can be clicked to open the algorithm detail modal shown in Figure 9.

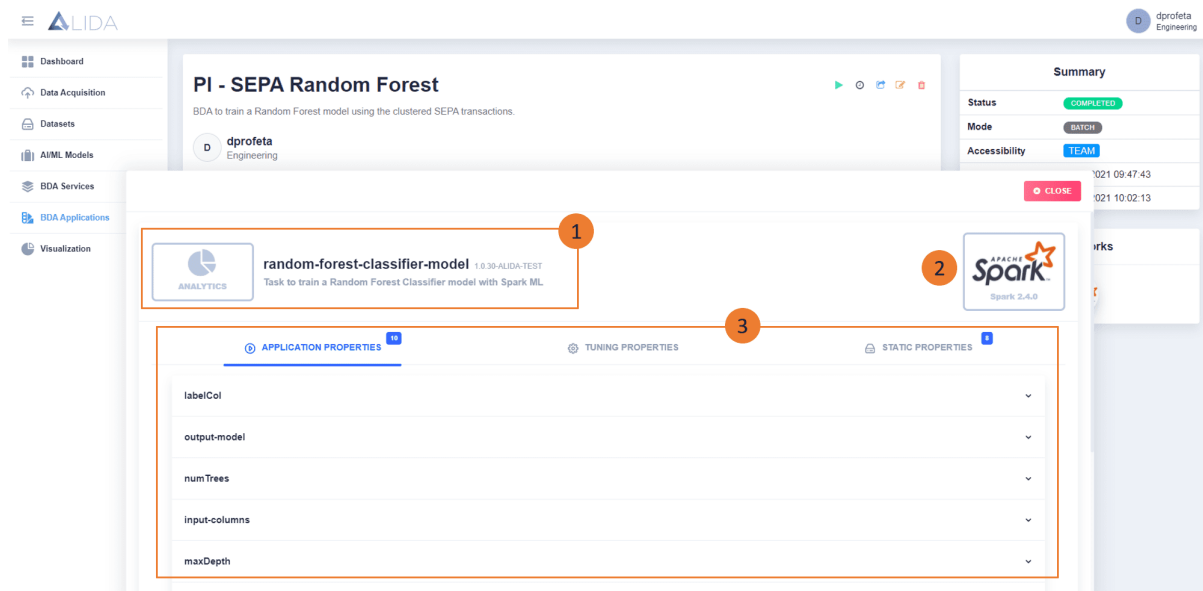


Figure 9 - Application detail modal

In this dialog user can check the algorithm's basic information (1), such as name, version, description, the algorithm framework (2), as well as the list of properties (3) that characterize its execution.

The Datasets panel (7- Figure 8) contains the list of all the datasets that belong to the application, including both the input datasets, indicated by a green IN badge, and the output datasets, indicated by a red OUT badge, based on whether the dataset has been given as input to the algorithms or has been generated at the end of their execution.

Some datasets are provided with one additional button to access the data visualization section to plot a chart of the data (8 in Figure 8)

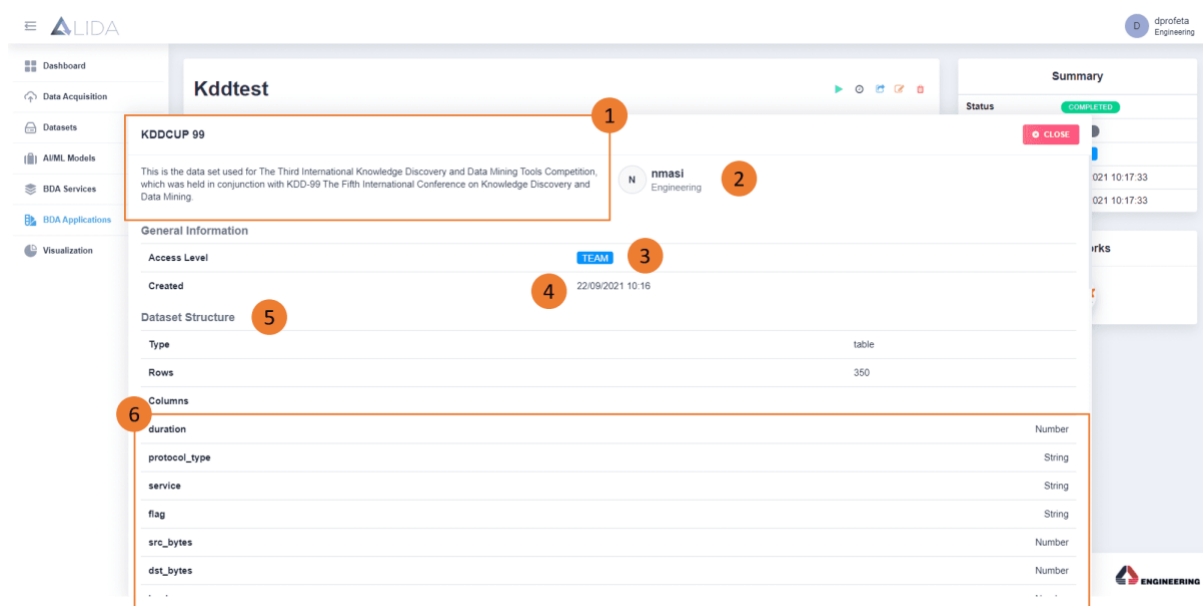


Figure 10 - Dataset details

If output datasets appear in grey and look disabled, it's because the application has not been executed yet; on the other hand, user can interact with input datasets and completed applications output datasets to open the dataset detail dialog (Figure 10).

As can be seen from Figure 10, the modal contains: dataset name description (1) and the owner on the top then the panel is divided into general information, containing the access level (3) and the creation date (4), and below the general information there is the dataset structure information(5), with the number of rows and the list of columns that compose the table.

Models panel (9 in Figure 8) follows the same IN/OUT convention as the datasets list, since models can be either generated by or can be given as inputs to -model algorithms. In the same way as algorithms and datasets, models are clickable too: the click triggers the model detail dialog showed in Figure 11.

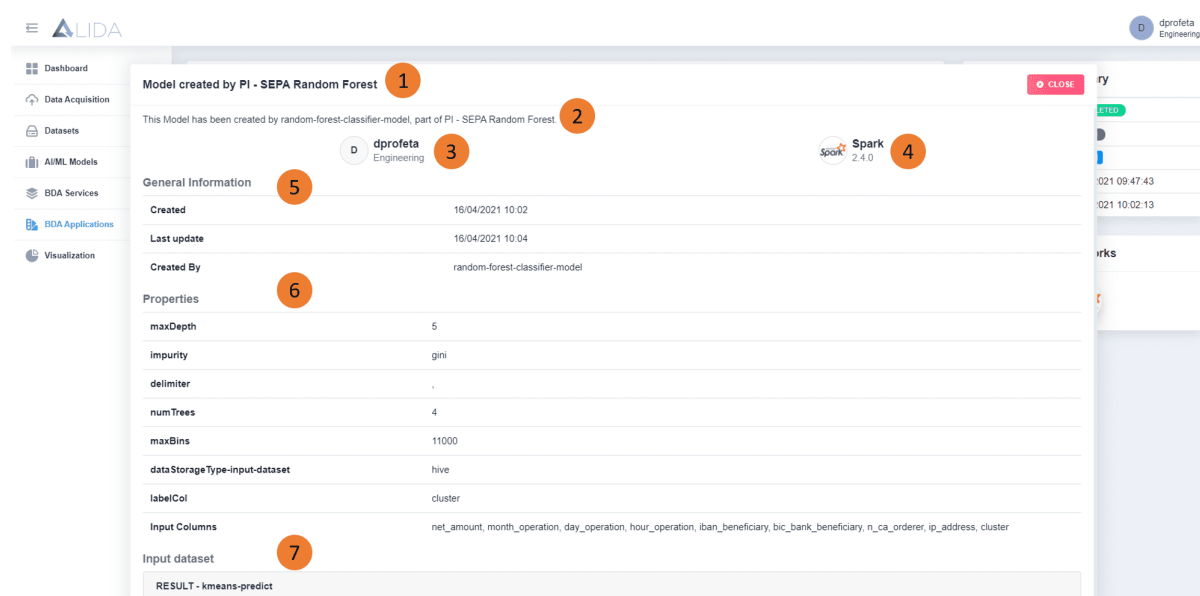


Figure 11 -Model detail

Model Detail dialog title indicates the BDA Application the model belongs to (1). Inside the dialog user can find the model description (2) , in the row below the model owner (3) and the framework it was calculated with (4). Model General Information (5) includes the creation and last update dates, as well as the BDA service it was generated by. Below these information, it is also possible to check the model-specific properties (6) and the detail of the dataset used to create the model (7).

Visualizations container (10 in Figure 8) includes all the visualization configurations user decide to use in the application: in fact when an application is creating user can decide to visualize the data elaborated and save the chart that can be recall from this panel.

The click button opens the application schedules dialog. The dialog lists all the scheduled jobs for the specific application (Figure 12).

In the dialog each entry of the table is provided with a button to remove the registered schedules. In order to create a new schedule, user have to select “add schedule” button, then user will be asked to fill the schedule name (1) and the cron expression of the job (2).

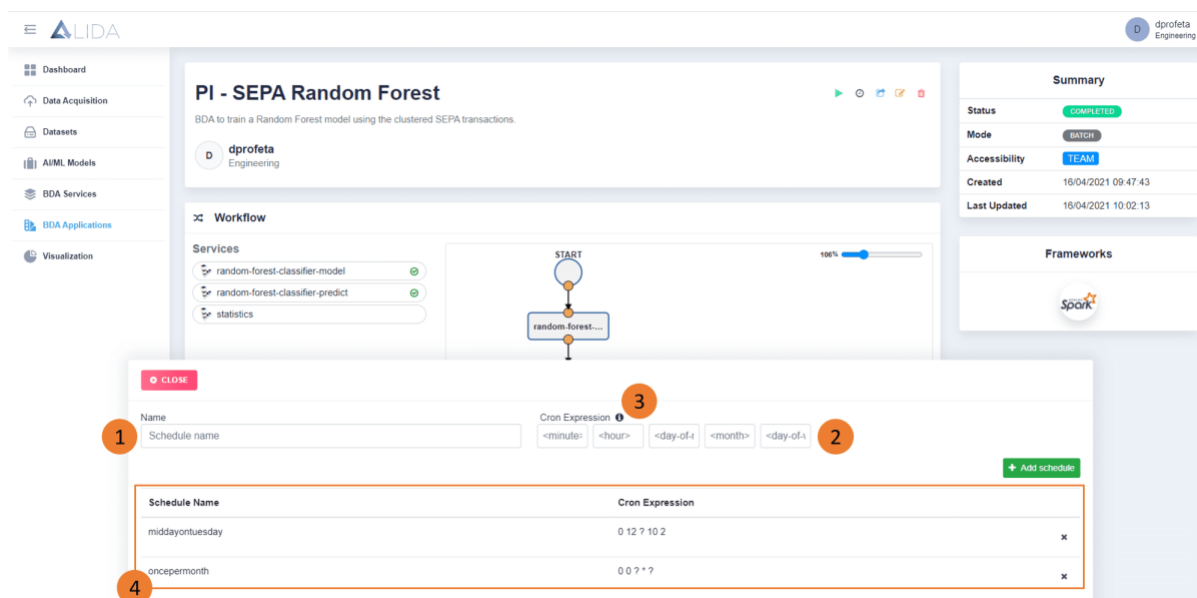


Figure 12- Application schedule

A cron expression is a string consisting of five fields. Each field is related to a specific time period, according to the following schema:

<minute> <hour> <day-of-month> <month> <day-of-week>

Each field can be a numeric value to state that the execution should happen once at that specified moment in time, or a special character to make it recurring. Accepted special characters are listed below:

- (all) specifies that event should happen for every time unit. For example, “\*” in the <minute> field means “for every minute.”
- ? (any) is utilized in the <day-of-month> and <day-of-week> fields to denote the arbitrary value and thus neglect the field value. For example, if we want to fire a script at “5th of every month” irrespective of what day of the week falls on that date, we specify a “?” in the <day-of-week> field.
- - (range) determines the value range. For example, “10-11” in the <hour> field means “10th and 11th hours.”
- , (values) specifies multiple values. For example, “MON, WED, FRI” in <day-of-week> field means on the days “Monday, Wednesday and Friday.”
- / (increments) specifies the incremental values. For example, a “5/15” in the <minute> field means at “5, 20, 35 and 50 minutes of an hour.”
- # specifies the N<sup>th</sup> occurrence of a weekday of the month, for example, “third Friday of the month” can be indicated as “6#3”.
- If you don't remember how a cron expression works, you can revise the fields meaning and the accepted values anytime by checking the help icon on top of the cron editor (3).

Below some examples of cron expression below:

- At 12:00 p.m. (noon) every day
- 0 12 \* \* ?
- At 9:30 a.m. every Monday, Tuesday, Wednesday, Thursday and Friday
- 30 9 ? \* MON-FRI

- At 12 midnight on every day for five days starting on the 10th day of the month
- 0 0 10/5 \* ?

Under the cron editor user can also find the list of all existing cron jobs for that application, if any (4).

If the application run in streaming mode there are no explicit assets generated by its execution or provided as inputs, its detail page is way simpler than a batch one( Figure 13).

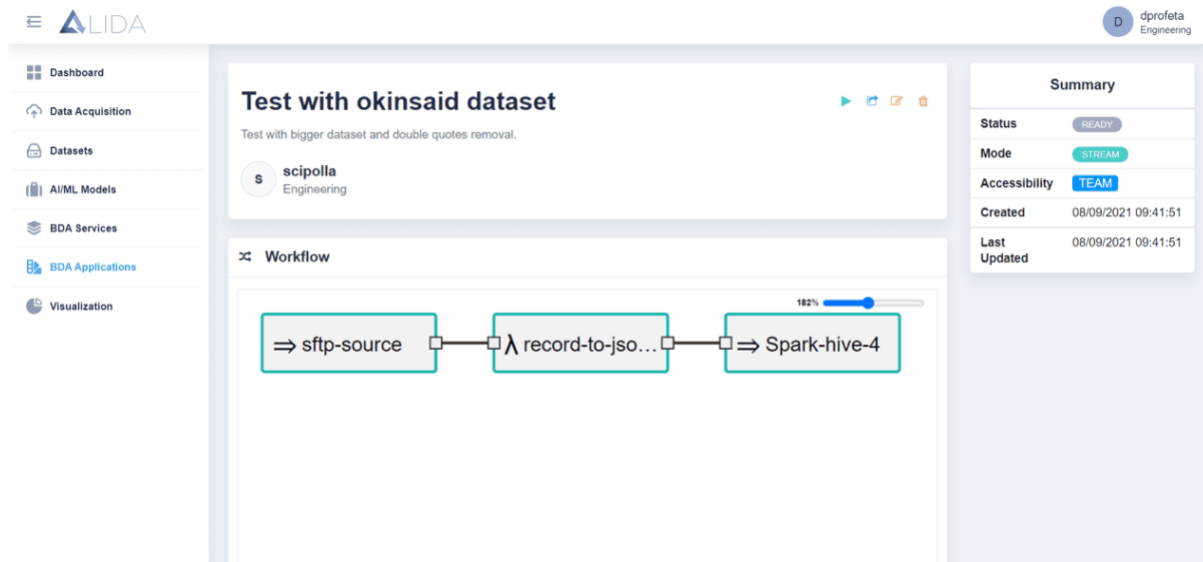


Figure 13 - Streaming Application Detail

### 3.3.4. Machine learning model

ALIDA allows to use and configure machine learning algorithms. These algorithms have a training phase that creates a model, which can then be used for the actual processing. The ML section contains all the models that have been generated by the machine learning algorithms run on ALIDA (Figure 14).

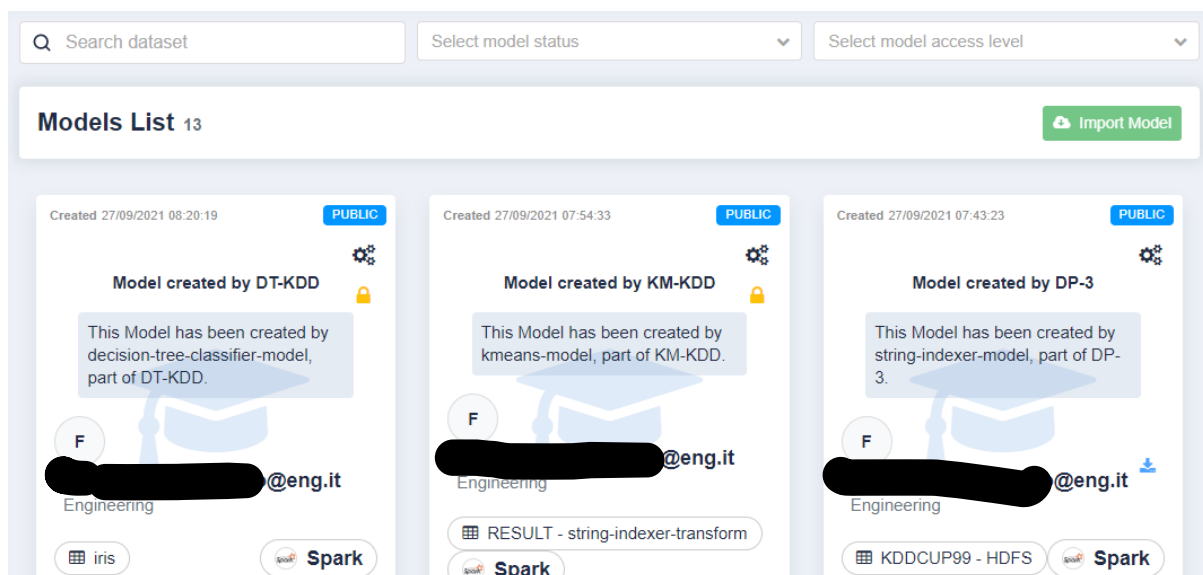
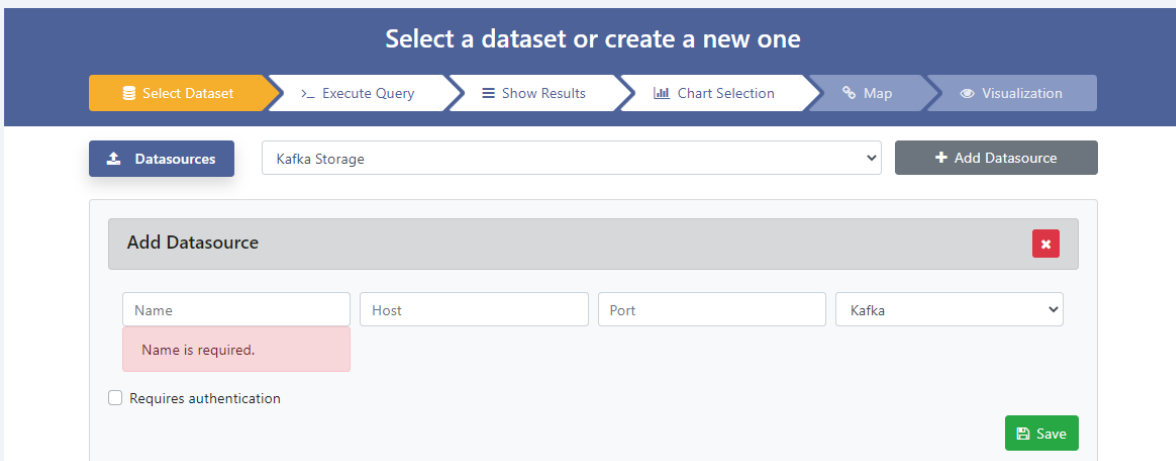


Figure 14 - Machine learning model

The models are listed through cards that contain information about the model, such as: description, scope (with a locker if the model is private), creation framework, and owner.

### 3.3.5. Visualization

After executing a workflow, it is possible to save the data in a dataset, through the visualization section, using a wizard, and it is possible to view the data in charts.



The screenshot shows a web interface titled "Select a dataset or create a new one". At the top, there is a navigation bar with several steps: "Select Dataset" (highlighted in orange), ">\_ Execute Query", "≡ Show Results", "Chart Selection", "Map", and "Visualization". Below the navigation bar, there is a "Datasources" section with a dropdown menu showing "Kafka Storage" and a "+ Add Datasource" button. A modal window titled "Add Datasource" is open, containing fields for "Name", "Host", "Port", and a dropdown for "Kafka". A red error message "Name is required." is displayed below the "Name" field. There is also a checkbox for "Requires authentication" and a green "Save" button.

Figure 15 - Visualization page

The display wizard consists of a series of steps:

- Dataset selection
- Query execution
- Show results
- Chart selection
- Map
- Visualization

First user has to select the dataset, different datasets are available based on the configuration of the application. The datasets are the results of the BDA Applications performed on the platform. Also, dataset can be connected using the add data source button.

In the next step, "Execute Query" user can select one of the dataset contained in the data source. By selecting a dataset, a date preview is shown. The next step consists in selecting the chart based on the type of data user wants to visualize Figure 16.

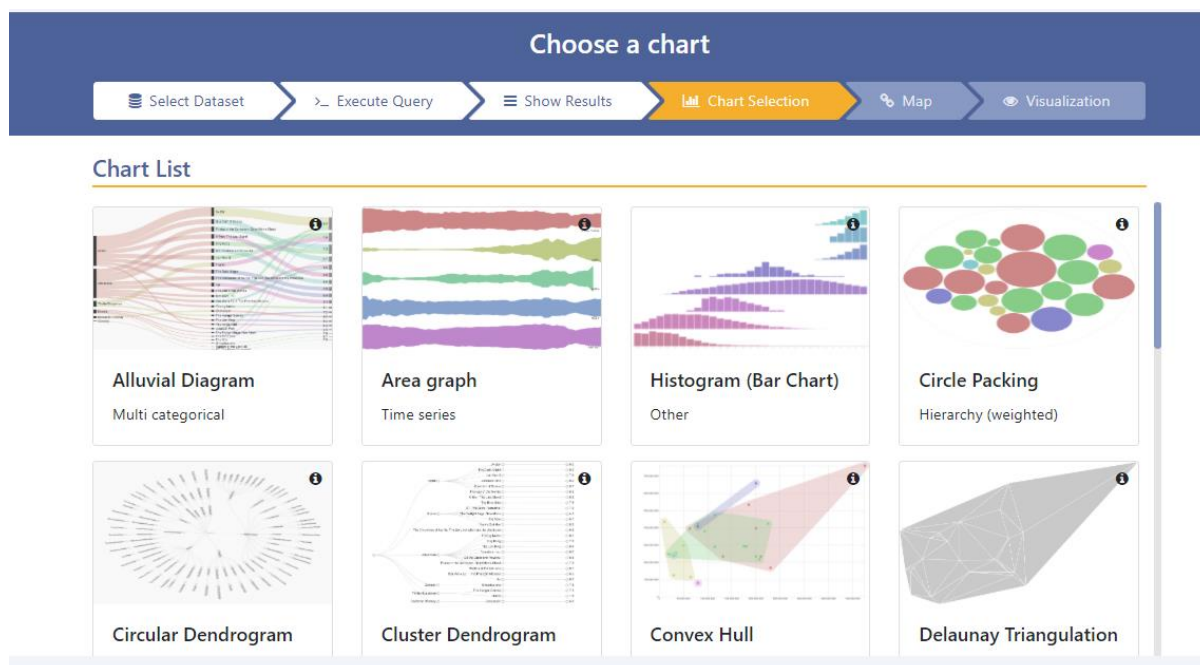


Figure 16 - Chart selection

In order to render the data in the chart, the chart engine need to be specified with the association between attributes of the datasets and dimensions of the chart, this mapping operation is performed in the penultimate step "Map" Figure 17.

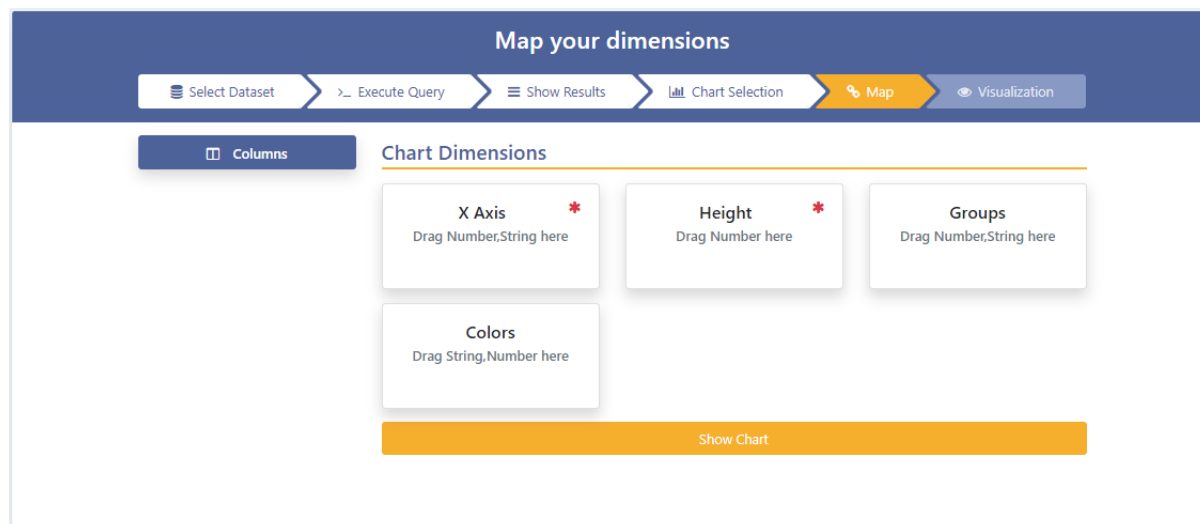


Figure 17 - Attribute Mapping

At the end of this operation the data will be visualized on the chart in the visualization step.

### 3.3.6. BDA Services

The BDA Services section contains the service catalogue where the services are listed and described (Figure 18). Clicking on the service, the system shows the details with all the configuration parameters. Each of these services can be used in a workflow. At the top of the area there are drop-down filters that allow a quick selection.



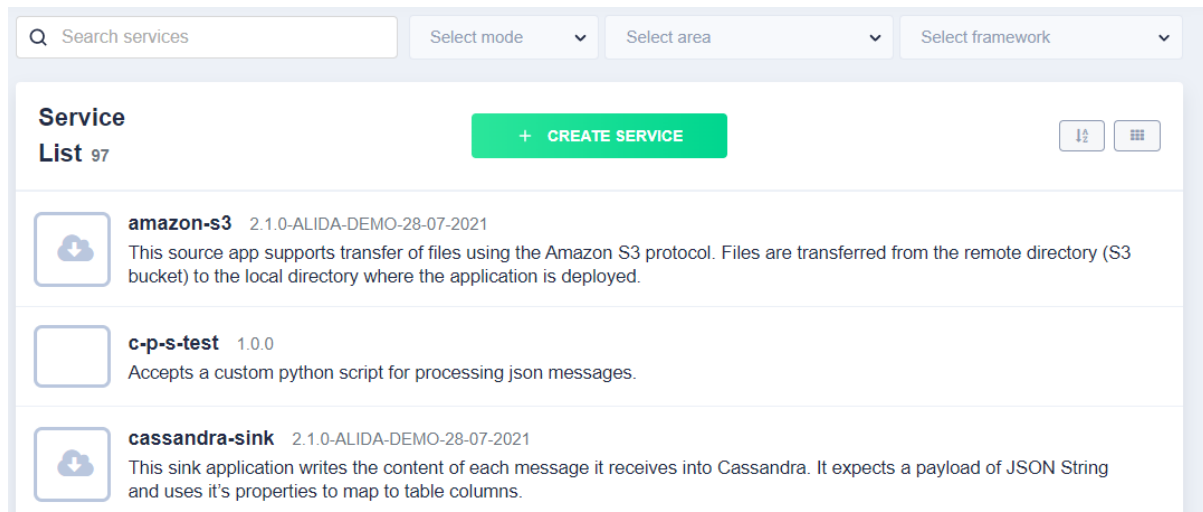


Figure 18 - BDA Services list

Services are organized by mode (batch and streaming) and by area, i.e., the general function they perform. Below some of the most common areas:

- **Preparation:** It includes all the services intended to do pre-processing steps, for example resampling a time series, data aggregation or even simple arithmetic operations.
- **Analysis:** This is a macro area containing ML and statistical algorithms wrapped inside ALIDA services.
- **Evaluation:** These services have been designed in order to help evaluating generated models' performance. They include confusion matrix calculations, precision and accuracy measuring and so on.
- **Ingestion:** Ingestion services are useful whenever user need to do data ingestion from remote servers or devices.

## 4. BDA services

As explained before, ALIDA services are docker containerized microservices. Each microservice implements a specific Machine Learning algorithm, there are data acquisition functions, especially as seen in the streaming configuration, and there are also BDA services that allow user to prepare data for processing. In any case, the beating heart of BDA's services are its elaboration services, capable of processing the data acquired on the platform. The fundamental processing algorithms for the platform are Machine Learning algorithms, processing BDA services therefore can be analysed and evaluated based on the classic classifications of machine learning algorithms. In the next section the machine learning algorithms currently contained in the platform will be presented, while in section 5 all the machine learning BDA services will be listed including the main attributes and their description. Also with regard to this list it must be considered that the system is currently evolving and that therefore the list of algorithms and services will be updated and extended, including more and more algorithms and functions, those present in this document constitute the main nucleus.

### 4.1. Machine learning algorithms categorization

The purpose of Machine Learning is developing methods that automatically look for patterns in the data and use them such patterns to make predictions or to describe the intrinsic structure of the data. So the discipline can be divide into

- supervised learning
- unsupervised learning,

depending on the study to be carried out.

The purpose of supervised learning is predicting the value of a particular attribute Y starting from the values of other attributes X; the techniques of unsupervised learning instead aim to explore particular structures contained in the data that summarize the relationships underlying them.

Another way according to which it is possible to distinguish different categories of tasks is the type of output expected from a certain machine learning system. Among the main categories we find:

- **Classification**, in which inputs are divided into two or more classes and the learning system must produce a model capable of assigning one or more classes among those available to an input. These types of tasks are typically addressed using supervised learning techniques. An example of classification is the assignment of one or more labels to an image based on the objects or subjects contained in it;
- **Regression**, conceptually similar to classification with the difference that the output has a continuous and non-discrete domain. It too is typically addressed with supervised learning. The domain of the output in question is virtually infinite, and not limited to some discrete set of possibilities;
- **Clustering**, in which, as in classification, a set of data is divided into groups which, however, unlike this, are not known a priori. The very nature of problems belonging to this category typically makes them unsupervised learning tasks.

Just like the human mind, computer systems also need information to implement this form of learning. To provide the necessary data, we go through the so-called Data Observation and Data Preparation phases. During these phases, systems are exposed to large amounts of data that support the learning process. By experiencing this data, systems can improve an understanding of data patterns and their meaning. Hence, supporting the generation of analysis and forecasts that user wants to produce.

## 4.2. BDA Services analysis

The primary purpose of machine learning is therefore to generalize the information contained in the training data into useful models (ex: Table 1 - Decision Tree Classifier Model) and then make predictions on new data (example: Table 2 - Decision Tree Classifier Predict). In this type of approach, it is therefore essential to evaluate how much the models created will be able to make correct predictions on the new data.

As explored in preview section, there is a significant difference between classification and regression: in classification, algorithms explore the attribute space to separate two or more classes of nominal data, while in regression, algorithms explore the attribute space to find a generalization to a numerical data. More than 50 years of research in Machine Learning have developed a well-defined experimental method that evaluates in a very precise way the quality of the models that are created. Theoretically, there are two factors that affect the quality of a model:

- The deviation (**bias**), due to a poor ability to generalize the essential information or the relationships between attributes from the data. A model with a high bias produces a difference between the value predicted by the model and the real value, ending up with low accuracy performance (underfitting). Having too many attributes (the curse of dimensionality) can increase the bias considerably.
- The variance (**variance**), due to the complexity of models that predict a lot of variability. A high variance leads a model to capture too detailed information in the training phase and then generate more likely fallacious predictions on new data (overfitting).

The best models are those that manage to reduce both the bias and the variance, but generally the two factors tend to counterbalance each other: if we reduce the variance too much, the bias will tend to increase and vice versa. Machine Learning provides many supervised and unsupervised algorithms, and each family has its own characteristics, with strengths and weaknesses, in this section we classify briefly the algorithms available in ALIDA. The Bayesian algorithms are algorithms that typically allow classification. These are algorithms based on Bayes' conditional probability theorem, one of this algorithm is the **Naïve Bayes** (ref: section 5.13) which assume the statistic independence of the attributes (variables). This algorithm is very useful in case of multi-class classification, especially when the classes have different probabilities of happening to each other. Bayesians tend to generate linear models, with high bias and low variance.

There are also many types of classification / regression algorithms. The most used are based on functions. These are mathematical classification and regression algorithms that associate numerical weights to attributes in relation to their predictive power. Being very versatile they are used in many fields. These algorithms include **Linear Regression** (ref: section 5.7), which linearly separates nominal classes or generalizes numerical values to a multidimensional cartesian plane, and **Linear Support Vector Machine** (ref: section 5.8), which finds the best line by which to separate or generalize the multidimensional attribute space, and is able to transform it to separate non-linearly separable spaces with polynomial functions. Models that produce linear functions tend to have high bias and low variance, while polynomial functions produce more complex models that reduce bias and increase variance. When function-based algorithms are put into a network structure, we have neural networks, classification and regression algorithms inspired by the functioning of the brain. These algorithms include the **Multilayer Perceptron** (ref: section 5.12). Generalizing, these algorithms implement a network of perceptrons. A perceptron can be considered as a mathematical model of a binary neuron, the perceptron is in fact implemented by a nonlinear function, which has an input vector and two outputs. The function, on the basis of a vector of weights which is trained during the learning process,

decides whether the input vector will be "classified" (ie directed) to one output rather than another. The Multilayer Perceptron Network consists of at least three layers of perceptron, capable of calculating the weights of the combinations of the most predictive attributes by exploring the attribute space in a non-linear way. Being non-linear models, they greatly reduce the bias and considerably increase the variance, exposing to the risk of overfitting.

Rule-based classifiers are generally underrated classification and regression algorithms, but very useful for data analysis because they produce interpretable models made of if-then rules. In this family there are algorithms that minimize the number of rules extracted, producing models with more bias and less variance, or algorithms that find as many rules as possible.

**Decision Tree** (ref: section 5.1) are algorithms that enhance rule classifiers by placing the most informative attributes at the root of the tree. The more general rules are those that include in the root of the tree, while the more detailed rules are found in the leaves. Decision trees with many leaves tend to reduce the bias and increase the variance, so pruning is used, which allows to reduce the variance by removing the leaves.

To improve the effectiveness of decision trees, so-called ensemble techniques are used, which are based on the creation of meta-models. In ensemble learning we can find the so-called bagging techniques, which aim to create a set of classifiers which contribute to the final decision by having the same importance or weight. At the time of classification, each model will vote on the outcome of the prediction and the overall output will be the class that has received the highest number of votes. Among bagging technics there is the **Random Forest** (ref: section 5.4) algorithm which works building several trees with different placements of the most informative attributes, and then doing a meta-model that optimizes the relationship between bias and variance. Algorithms of the **Isolation Forest** (ref: section 5.3) type are particular types of Random Forest that isolate – as the name suggests – each point of the dataset based on its greater or lesser distance from the other points. Outliers typically correspond to the leaves that are cut from the tree first. All points are then assigned a score based on their own height in the tree so the points closest to the root will have the highest score. Among the ensemble techniques we also find boosting. **Gradient Boosted** (ref: section 5.2) is an algorithm in which a series of decision trees are concatenated in series in a single model of boosting type. In Gradient Boosted each classifier influences the final grade with a certain weight. This weight is calculated on the basis of the accuracy error that each model will make in the learning phase, in this case using gradient function.

Clustering algorithms also fall into several families. Centroid clustering, which includes algorithms **Kmeans** (ref: section 5.6), finds prototypical instances and aggregates the closest ones into circular or elliptical clusters, maximizing similarity at the expense of purity. The Kmeans is one of those clustering algorithms that do not rely on formal models for the structure of clusters in the data. One of the clustering approaches that postulates a formal statistical model for the population from which data is sampled is known as a finite mixture density model. This model postulates that the overall population is actually made up of a number of subpopulations (the "clusters"), each with a different multivariate probability density function. The **Gaussian Mixture** (ref: section 5.5) algorithm adopt the Gaussian as a statistical model. In this approach, the clustering problem estimates the parameters of the assumed mixture and uses the estimated parameters to calculate the (posterior) probabilities of cluster membership for each item. Furthermore, determining the number of clusters reduces to a model selection problem for which objective procedures exist. Finite model-based cluster analysis is also known as model-based clustering methods.

## 5. BDA Services Catalogue

The following sections described the services that have been analysed in previous section BDA services 4. Each service has been described using a table that contains the general data plus attributes that must be configured in order to perform the algorithm. For each service, each of the attributes is described in such a way as to be able to give the user all the general information. The algorithms implemented here exploit the Spark Mllib library<sup>10</sup>, so, apart from section 4.2, it is possible to refer to this documentation for further clarifications on individual services<sup>11</sup>. Moreover, each single spark attribute is further explained in the Spark configuration documentation page<sup>12</sup>.

### 5.1. Decision Tree

Decision trees and their ensembles are popular methods for the machine learning tasks of classification and regression. Decision trees are widely used since they are easy to interpret, handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions. Tree ensemble algorithms such as random forests and boosting are among the top performers for classification and regression tasks.<sup>13</sup>

#### 5.1.1. Decision Tree Classifier Model

Service name	Decision Tree Classifier Model			
Technology	Spark			
Description	Task to train a Decision Tree Classifier model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid	STRING	Always	FALSE

<sup>10</sup> <https://spark.apache.org/mllib/>

<sup>11</sup> <https://spark.apache.org/docs/latest/ml-guide.html>

<sup>12</sup> <https://spark.apache.org/docs/3.1.2/configuration.html>

<sup>13</sup> <https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-trees>

	values are Always, Never, and IfNotPresent.			
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	decision-tree-classifier-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/decision-tree-classifier:1.1.2	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced kmeans model	STRING		TRUE
maxDepth	Maximum depth of the tree. ( $\geq 0$ )	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous	INT	32	FALSE

	features. Must be $\geq 2$ and $\geq$ number of categories for any categorical feature.			
impurity	Criterion used for information gain calculation. Supported options: entropy, gini	STRING	gini	FALSE
labelCol	Label column name	STRING		FALSE
delimiter	Dataset delimiter	STRING		FALSE

Table 1 - Decision Tree Classifier Model

### 5.1.2. Decision Tree Classifier Predict

Service name	decision-tree-classifier-predict			
Technology	Spark			
Description	ML-Predict task			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and	INT	1	FALSE



	parallelize when not set by user			
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE
spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE

spark.name	Spark app name	STRING	ml-predict	FALS E
pythonModule	Python module	STRING	main	FALS E
spark.executor.instances	Number of executors	INT	3	FALS E
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALS E
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	FALS E
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALS E
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALS E
input-dataset	Input dataset containing data for processing.	STRING		FALS E
input-columns	Selected columns from table	STRING	ANY	FALS E
output-dataset	Input dataset containing data for processing.	STRING	ANY	FALS E
input-model	Input kmeans model	STRING		FALS E

outputLabel	Label column name.	STRING	prediction	FALS E
delimiter	Dataset delimiter	STRING	,	FALS E

*Table 2 - Decision Tree Classifier Predict*

### 5.1.3. Decision Tree Regressor Model

Service name	Decision Tree Regressor Model			
Technology	Spark			
Description	Task to train a Decision Tree Regressor model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/ alida/analytics/spark-analytics/ decision-tree-regressor-model:1.1.2	FALSE

spark.name	Spark app name	STRING	decision-tree-regressor-model	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
output-model	Produced kmeans model	STRING		TRUE
labelCol	Label column name	STRING		FALSE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
maxDepth	Maximum depth of the tree. (>=0)	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous features. Must be >=2 and >= number of categories for any categorical feature.	INT	32	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/decision-tree-regressor-model:1.1.2	FALSE
spark.name	Spark app name	STRING	decision-tree-regressor-model	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
output-model	Produced kmeans model	STRING		TRUE
labelCol	Label column name	STRING		FALSE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
maxDepth	Maximum depth of the tree. ( $\geq 0$ )	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous features. Must be $\geq 2$ and $\geq$ number of categories for any categorical feature.	INT	32	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

input-dataset	Input dataset containing data for processing.	STRING		TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/decision-tree-regressor-model:1.1.2	FALSE
spark.name	Spark app name	STRING	decision-tree-regressor-model	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
output-model	Produced kmeans model	STRING		TRUE
labelCol	Label column name	STRING		FALSE

input-columns	Selected columns from table	STRING	NUMBER	TRUE
maxDepth	Maximum depth of the tree. ( $\geq 0$ )	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous features. Must be $\geq 2$ and $\geq$ number of categories for any categorical feature.	INT	32	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

Table 3.- Decision Tree Regressor Model



#### 5.1.4. Decision Tree Regressor Predict

Service name	Decision Tree Regressor Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	TRUE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 4 - Decision Tree Regressor Predict

## 5.2. Gradient Boosted

Gradient Boosted is part of ensemble techniques of decision trees. Gradient Boosted is an algorithm in which a series of decision trees are concatenated in series in a single model of the Boosting type, in Gradient Boosted algorithm each classifier influences the final grade with a certain weight. This weight will be calculated on the basis of the accuracy error that each model will make in the learning phase, in this case using gradient function.

### 5.2.1. Gradient Boosted Tree Classifier Model

Service name	Gradient Boosted Tree Classifier Model			
Technology	Spark			
Description	Task to train a Gradient Boosted Tree Classifier model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	gbt-classifier-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE

spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/gradient-boosted-tree-classifier-model:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced gbt-classifier-model	STRING		TRUE
labelCol	Label column name	STRING		FALSE
maxDepth	Maximum depth of the tree. ( $\geq 0$ )	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous features. Must be $\geq 2$ and $\geq$ number of categories for any categorical feature.	INT	32	FALSE
minInstancesPerNode	Minimum number of instances each child must have after split. If a split causes the left or right child to have fewer than minInstancesPerNode,	INT	1	FALSE

	the split will be discarded as invalid			
minInfoGain	Minimum information gain for a split to be considered at a tree node	INT	0	FALSE
maxIter	Max number of iterations	INT	20	FALSE
stepSize	Step size (a.k.a. learning rate) in interval (0, 1] for shrinking the contribution of each estimator	DOUBLE	0.1	FALSE
subsamplingRate	Fraction of the training data used for learning each decision tree, in range (0, 1]	DOUBLE	1	FALSE

Table 5 - Gradient Boosted Tree Classifier Model

### 5.2.2. Gradient Boosted Tree Classifier Predict

Service name	Gradient Boosted Tree Classifier Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	TRUE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE



spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 6 - Gradient Boosted Tree Classifier Predict

### 5.2.3. Gradient Boosted Tree Regressor Model

Service name	Gradient Boosted Tree Regressor Model			
Technology	Spark			
Description	Task to train a Gradient Boosted Tree Regressor model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	gbt-regressor-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/gradient-boosted-tree-regressor-model:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced gbt-classifier-model	STRING		TRUE
labelCol	Label column name	STRING		FALSE
maxDepth	Maximum depth of the tree. ( $\geq 0$ )	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous features. Must be $\geq 2$ and $\geq$ number of categories for any categorical feature.	INT	32	FALSE
minInstancesPerNode	Minimum number of instances each child must have after split. If a split causes the left or right child to have fewer than minInstancesPerNode, the split will be discarded as invalid	INT	1	FALSE
minInfoGain	Minimum information gain for a split to be considered at a tree node	INT	0	FALSE
maxIter	Max number of iterations	INT	20	FALSE

stepSize	Step size (a.k.a. learning rate) in interval (0, 1] for shrinking the contribution of each estimator	DOUBLE	0.1	FALSE
subsamplingRate	Fraction of the training data used for learning each decision tree, in range (0, 1]	DOUBLE	1	FALSE

Table 7 - Gradient Boosted Tree Regressor Model

#### 5.2.4. Gradient Boosted Tree Regressor Predict

Service name	Gradient Boosted Tree Regressor Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	TRUE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 8 - Gradient Boosted Tree Regressor Predict

### 5.3. Isolation Forest

Isolation Forest are types of Random Forest that isolates – as the name suggests – each point of the input dataset based on its greater or lesser distance from the other points. Outliers typically correspond to the leaves that are cut from the tree first. All points are then assigned a score based on their own depth in the tree so the points closest to the root will have the highest score.

#### 5.3.1. Isolation Forest Model

Service name	Isolation Forest Model			
Technology	Spark			
Description	Generates an isolation forest model.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	isolation-forest-model-cm	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE



spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-synapseml/isolation-forest-model-cm:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced an isolation forest model	STRING		TRUE
bootstrap	If true, draw sample for each tree with replacement. If false, do not sample with replacement.	BOOLEAN	TRUE	FALSE
number-of-estimators	The number of trees in the ensemble.	INT	100	FALSE
max-samples	The number of samples used to train each tree. If this value is between 0.0 and 1.0, then it is treated as a fraction. If it is >1.0, then it is treated as a count.	DOUBLE	256	FALSE
max-features	The number of features used to train each tree. If this value is between 0.0 and 1.0, then it is treated as a fraction. If it is	DOUBLE	1	FALSE

	>1.0, then it is treated as a count.			
contamination	Number of principal components to obtain	DOUBLE	0.002	FALSE
contamination-error	The error allowed when calculating the threshold required to achieve the specified contamination fraction. The default is 0.0, which forces an exact calculation of the threshold. The exact calculation is slow and can fail for large datasets. If there are issues with the exact calculation, a good choice for this parameter is often 1% of the specified contamination value	DOUBLE	0.0	FALSE
random-seed	The seed used for the random number generator.	INT	1	FALSE

Table 9 - Isolation Forest Model

### 5.3.2. Isolation Forest Predict

Service name	Isolation Forest Predict			
Technology	Spark			
Description	Generates predictions based on a previously generated Isolation Forest model.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	isolation-forest-predict	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-synapseml/isolation-forest-predict:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING		TRUE
input-model	Input Isolation Forest model	STRING		FALSE
append	If true, . If false, .	BOOLEAN	TRUE	FALSE

Table 10 - Isolation Forest Predict

## 5.4. Random Forest

To improve the effectiveness of decision trees, so-called ensemble techniques are used, which are based on the creation of meta-models. In ensemble learning we can find the so-called bagging techniques, which aim to create a set of classifiers having the same importance. At the time of classification, each model will vote on the outcome of the prediction and the overall output will be the class that has received the highest number of votes. Among bagging technics there is the Random Forest algorithm which works building several trees with different placements of the most informative attributes, and then doing a meta-model that optimizes the relationship between bias and variance.

### 5.4.1. Random Forest Classifier Model

Service name	Random Forest Classifier Model			
Technology	Spark			
Description	Task to train a Random Forest Classifier model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	random-forest-classifier-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE

spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/random-forest-classifier-model:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced Random Forest Classifier model	STRING		TRUE
maxDepth	Maximum depth of the tree. ( $\geq 0$ )	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous features. Must be $\geq 2$ and $\geq$ number of categories for any categorical feature.	INT	32	FALSE
impurity	Criterion used for information gain calculation. Supported options: entropy, gini	STRING	gini	FALSE

numTrees	Number of trees to train	INT	4	FALSE
labelCol	Label column name	STRING		TRUE

Table 11 - Random Forest Classifier Model

#### 5.4.2. Random Forest Classifier Predict

Service name	Random Forest Classifier Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join,	INT	1	FALSE

	reduceByKey, and parallelize when not set by user			
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE
spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE



spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 12 - Random Forest Classifier Predict

### 5.4.3. Random Forest Regressor Model

Service name	Random Forest Regressor Model			
Technology	Spark			
Description	Task to train a Random Forest Classifier model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	random-forest-regressor-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/random-forest-regressor-model:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced Random Forest Classifier model	STRING		TRUE
maxDepth	Maximum depth of the tree. (>=0)	INT	5	FALSE
maxBins	Max number of bins for discretizing continuous features. Must be >=2 and >= number of categories for any categorical feature.	INT	32	FALSE
numTrees	Number of trees to train	INT	4	FALSE
labelCol	Label column name	STRING	NUMBER	FALSE

Table 13 - Random Forest Regressor Model

#### 5.4.4. Random Forest Regressor Predict

Service name	Random Forest Regressor Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 14 - Random Forest Regressor Predict

## 5.5. Gaussian Mixture

The Gaussian Mixture is a clustering algorithm which rely on formal models for the structure of clusters in the data. This model postulates that the overall population is made up of several clusters, each with a gaussian multivariate probability density function. The clustering problem estimates the parameters of the Gaussian mixture, and then using the estimated parameters to calculate the probabilities of cluster membership for each item.

### 5.5.1. Gaussian Mixture Model

Service name	Gaussian Mixture Model			
Technology	Spark			
Description	Task to train a Gaussian Mixture model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	gaussian-mixture-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE

spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/gaussian-mixture-model:1.1.1	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced kmeans model	STRING		TRUE
k	Number of independent Gaussians in the mixture model.	INT	2	FALSE
maxIterations	The maximum number of iterations.	INT	100	FALSE
tol	The convergence tolerance for iterative algorithms.	DOUBLE	0.01	FALSE

Table 15 - Gaussian Mixture Model



### 5.5.2. Gaussian Mixture Predict

Service name	Gaussian Mixture Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	TRUE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 16 - Gaussian Mixture Predict

## 5.6. Kmeans

Kmeans 3.8 is a Centroid clustering class algorithm, which finds prototypical instances and aggregates the closest ones into circular or elliptical clusters, maximizing similarity at the expense of purity.

### 5.6.1. Kmeans Model

Service name	Kmeans Model			
Technology	Spark			
Description	Task to train a KMeans model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	kmeans-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/k-means-model:1.2.8	FALSE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced kmeans model	STRING		TRUE
k	The number of desired clusters. Note that it is possible for fewer than k clusters to be returned, for example, if there are fewer than k distinct points to cluster.	INT	3	FALSE
maxIterations	The maximum number of iterations to run.	INT	100	FALSE
distanceMeasure	The distance measure. Supported options: "Euclidean" and "cosine"	STRING	euclidean	FALSE

Table 17 - Kmeans Model

### 5.6.2. Kmeans Predict

Service name	Kmeans Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 18 - Kmeans Predict



## 5.7. Linear Regression

The linear regression is a regression algorithm that linearly separates the nominal classes or generalizes the numerical values in a multidimensional Cartesian plan.

### 5.7.1. Linear Regression Model

Service name	Linear Regression Model			
Technology	Spark			
Description	Task to train a Linear Regression model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	linear-regression-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/linear-regression-model:1.1.0	FALSE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced kmeans model	STRING		TRUE
maxIterations	The maximum number of iterations to run.	INT	100	FALSE
labelCol	Label column name	STRING		FALSE
regParam	Regularization parameter ( $\geq 0$ )	DOUBLE	0.0	FALSE
elasticNetParam	ElasticNet mixing parameter, in range [0, 1]	DOUBLE	0.0	FALSE

Table 19 - Linear Regression Model

### 5.7.2. Linear Regression Predict

Service name	Linear Regression Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 20 - Linear Regression Predict

## 5.8. Linear Support Vector Machine

Linear Support Vector Machine finding the best line by which separates nominal classes or generalizes numerical of a multidimensional attribute space, and is able to transform it to separate non-linearly separable spaces with polynomial functions.

Service name	Linear Support Vector Machine Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE

spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE
spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE

spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 21 - Linear Support Vector Machine



## 5.9. Logistic Regression

Logistic regression is a classical regression -based classification system to which an activation function is applied. The overall result represents the probability that it has the entrance to belong to the positive class, where by positive class means the class identified by the values placed above the decision Boundary.

Service name	Logistic Regression Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE

spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE
spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE

spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 22 - Logistic Regression

### 5.10. Min Max Fit

Service name	Min Max Fit			
Technology	Spark			
Description	Scale all values between a minimum value and a maximum value.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	min-max-fit	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/min-max-fit:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced model	STRING		TRUE
min	Minimum value for scaling	DOUBLE	0	FALSE
max	Maximum value for scaling	DOUBLE	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	min-max-fit	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/min-max-fit:1.1.0	FALSE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced model	STRING		TRUE
min	Minimum value for scaling	DOUBLE	0	FALSE
max	Maximum value for scaling	DOUBLE	1	FALSE

Table 23 - Min Max Fit

### 5.11. Min Max Scaler Process

Service name	Min Max Scaler Process			
Technology	Spark			
Description	Apply min max scaling to a dataset.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	min-max-scaler-process	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/min-max-predict:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-dataset	Input dataset containing data for processing.	STRING		TRUE
input-model	Input model	STRING		TRUE

Table 24 - Min Max Scaler Process



## 5.12. Multilayer Perceptron

Multilayer Perceptron algorithm implements a network of perceptrons. A perceptron can be considered as a mathematical model of a binary neuron, the perceptron is in fact implemented by a nonlinear function, which has an input vector and two outputs. The function, on the basis of a vector of weights which is trained during the learning process, decides whether the input vector will be "classified" (ie directed) to one output rather than another. The Multilayer Perceptron Network consists of at least three layers of perceptron: an input layer, a hidden layer and an output layer, capable of calculating the weights of the combinations of the most predictive attributes by exploring the attribute space in a non-linear way. Being non-linear models, they greatly reduce the bias and considerably increase the variance, exposing to the risk of overfitting.

### 5.12.1. Multilayer Perceptron Classifier Model

Service name	Multilayer Perceptron Classifier Model			
Technology	Spark			
Description	Task to train a Multilayer Perceptron Classifier model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	multilayer-perceptron-cl-mod	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE

spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/multilayer-perceptron-classifier-model:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced kmeans model	STRING		TRUE
labelCol	Label column name	STRING		FALSE
maxIterations	Max number of iterations	INT	100	FALSE
tol	The convergence tolerance for iterative algorithms ( $\geq 0$ )	INT	1.0E-6	FALSE
layers	Sizes of layers from input layer to output layer E.g., 780,100,10 means 780 inputs, one hidden layer with 100 neurons and output layer of 10 neurons	STRING	4,12,4	FALSE

stepSize	Step size to be used for each iteration of optimization ( $\geq 0$ )	DOUBLE	0.03	FALSE
blockSize	Block size for stacking input data in matrices. Data is stacked within partitions. If block size is more than remaining data in a partition then it is adjusted to the size of this data. Recommended size is between 10 and 1000	INT	128	FALSE
solver	The solver algorithm for optimization. Supported options: l-bfgs, gd	STRING	l-bfgs	FALSE

Table 25 - Multilayer Perceptron Classifier Model

### 5.12.2. Multilayer Perceptron Classifier Predict

Service name	Multilayer Perceptron Classifier Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 26 - Multilayer Perceptron Classifier Predict

### 5.13. Naïve Bayes

Naïve Bayes algorithm is based on Bayes' conditional probability theorem, is called Naïve because it assumes the independence of the attributes of the dataset in input. This algorithm is very useful in case of multi-class classification, especially when the classes have different probabilities of happening to each other. Bayesians tend to generate linear models, with high bias and low variance.

#### 5.13.1. Naïve Bayes Model

Service name	Naïve Bayes Model			
Technology	Spark			
Description	Task to train a Naive Bayes model with Spark ML			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	naive-bayes-model	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE

spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/naive-bayes-model:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced kmeans model	STRING		TRUE
labelCol	Label column name	STRING		FALSE
smoothing	The smoothing parameter, should be $\geq 0$	DOUBLE	1.0	FALSE
modelType	The model type: "multinomial" (default) or "Bernoulli"	STRING	multinomial	FALSE

Table 27 - Naïve Bayes Model



### 5.13.2. Naïve Bayes Predict

Service name	Naïve Bayes Predict			
Technology	Spark			
Description	ML-Predict Task.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form	INT	0	FALSE
spark.io.compression.codec	The codec used to compress internal data such as RDD partitions, event log, broadcast variables and shuffle outputs	INT	0	FALSE
spark.reducer.maxSizeInFlight	Maximum size of map outputs to fetch simultaneously from each reduce task	INT	50331648	FALSE
spark.default.parallelism	Default number of partitions in RDDs returned by transformations like join, reduceByKey, and parallelize when not set by user	INT	1	FALSE
spark.broadcast.blockSize	Size of each piece of a block for TorrentBroadcastFactory	INT	4194304	FALSE
spark.task.cpus	Number of cores to allocate for each task	INT	1	FALSE

spark.speculation	If set to true, performs speculative execution of tasks	BOOLEAN	0	FALSE
spark.shuffle.compress	Whether to compress map output files	BOOLEAN	1	FALSE
spark.broadcast.compress	Whether to compress broadcast variables before sending them	BOOLEAN	1	FALSE
spark.shuffle.spill.compress	Whether to compress data spilled during shuffles	BOOLEAN	1	FALSE
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	TRUE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	TRUE
spark.name	Spark app name	STRING	ml-predict	TRUE
pythonModule	Python module	STRING	main	TRUE
spark.executor.instances	Number of executors	INT	3	TRUE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	TRUE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/ml-predict:1.1.7	TRUE

spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	TRUE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	TRUE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	ANY	TRUE
output-dataset	Input dataset containing data for processing.	STRING	ANY	TRUE
input-model	Input kmeans model	STRING		TRUE
outputLabel	Label column name.	STRING	prediction	FALSE
delimiter	Dataset delimiter	STRING	,	FALSE

Table 28 - Naïve Bayes Predict

## 5.14. Principal Component Analysis

The Principal Component Analysis, an example of reducing dimensionality, aims to reduce the more or less large number of variables that describe a set of data to a smaller number of latent variables, limiting the loss of information as much as possible. The attribute space is modified by transposing it into a space that uses correlation and co-variance of the data set.

### 5.14.1. Principal Component Analysis Model

Service name	Principal Component Analysis Model			
Technology	Spark			
Description	Generate principal component analysis model.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	pca-fit	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE

spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/pca-model:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE
hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-model	Produced kmeans model	STRING		TRUE
k	Number of principal components to obtain	INT	2	FALSE
scale	Set to true to scale the center the input data (advised); else set to false.	BOOLEAN	TRUE	FALSE

Table 29 - Principal Component Analysis Model

### 5.14.2. Principal Component Analysis Predict

Service name	Principal Component Analysis Predict			
Technology	Spark			
Description	Apply principal component analysis.			
Input attribute				
Key	Description	Type	Default Value	Mandatory
spark.kubernetes.container.image.pullPolicy	Container image pull policy used when pulling images within Kubernetes. Valid values are Always, Never, and IfNotPresent.	STRING	Always	FALSE
spark.master	the spark master	STRING	k8s://159.69.137.144:6443	FALSE
spark.name	Spark app name	STRING	pca-predict	FALSE
pythonModule	Python module	STRING	main	FALSE
spark.executor.instances	Number of executors	INT	3	FALSE
spark.kubernetes.container.image.pullSecrets	Kubernetes secrets used to pull images from private image registries	STRING	alida-regcred	FALSE
spark.kubernetes.container.image	Container image to use for the Spark application	STRING	gitlab.alidalab.it:5000/alida/analytics/spark-analytics/pca-predict:1.1.0	FALSE
spark.kubernetes.namespace	The namespace that will be used for running the driver and executor pods	STRING	alida-develop	FALSE

hdfsUrl	The URL for HDFS service	STRING	hdfs://alida-develop-hdfs-namenode:8020	FALSE
input-dataset	Input dataset containing data for processing.	STRING		TRUE
input-columns	Selected columns from table	STRING	NUMBER	TRUE
output-dataset	Input dataset containing data for processing.	STRING		TRUE
input-model	Input kmeans model	STRING		TRUE
append	Set to true to keep the original columns; else set to false.	BOOLEAN	FALSE	FALSE

Table 30 - Principal Component Analysis Predict

## 6. Conclusions

In this deliverable the BDA Services have been released and described. These services will allow AgriBIT third party developers to perform analysis on the data acquired as part of the AgriBIT project and more generally during agricultural activities on the fields. The services are part of the ALIDA platform, a Big Data Analytics platform also released as part of the AgriBIT project. The platform contains BDA Services for data acquisition, the data preparation and for the storage, but above all it contains machines of machines learning which have been described in section 4.2. The services allow third -party users to exploit the data collected, elaborating complex analysis and providing to the AgriBIT farmers more information. The services can also be integrated into an external data flow by using streaming applications. Task T5.2, will provide data integration methods so that third party developers can leverage AgriBIT data in BDA applications, data integration will be released in D5.6. Finally, the services can be extended and enriched using the APIs that will be released in the deliverable D5.7.



## List of Abbreviations

Abbreviation	Explanation/Definition
BDA	Big Data Analytics
ICT	Information and Communications Technology
APIs	Application Program Interfaces
GUI	Graphical User Interface
AI	Artificial Intelligence
ML	Machine Learning
AI – ML	Artificial Intelligence and Machine Learning
OCI	Open Container Initiative
K8S	Kubernetes
URL	Uniform Resource Locator



## Internal Deliverable Review Form

Project Acronym	AgriBIT
Project Title	Artificial intelligence applied to pPrecision farming By the use of GNSS and Integrated Technologies
Grant Agreement number	101004259
Call	SU-SPACE-EGNSS-3
Funding Scheme	Innovation Action (IA)
Project duration	36 Months

Document Information			
Deliverable:	D4.6 – AgriBIT data services		
Work Package:	WP4	Task:	T4.4
Date of Review:	9/03/2023		
Internal Reviewer(s):	Artur Krukowski (RFSAT) Traianos Tertzis (ACP)		
Classification:	Public		

Topic	Answer	IF “No”, classify as “Major” or “Minor” issues	Comments
1. Is the content and structure of the deliverable compliant with the DoA?	Yes	-	n/a
2. Is the content of the deliverable scientifically relevant?	Yes	-	n/a
3. Is the content of the deliverable useful for the subsequent work on the project?	Yes	-	To be followed by guide to integration in D5.6.
4. Is the deliverable suitable to be submitted to the EC?	Yes	-	n/a
If not:			
4.1. Does it need formatting adjustments?	Yes	-	Some adjustments should be performed regarding the appearance. Not big spacing and figures alignment.
4.2. Does it need content adjustments?	n/a	-	n/a
4.3. Does it need to be significantly refined (e.g. content improvement, structure changes, etc.)?	n/a	-	n/a
Additional comments			
It has been corrected following the internal peer-review process. It is ready for submission to EC.			