National College of Ireland

BSc (Honours) in Computing

Software Development

2022/2023

Maximiliano Herrera

X20103212
x20103212@student.ncirl.ie

Artificial intelligence as a service (AIaaS)

Technical Report

Contents

Executive Summary

The purpose of this document is to present my final project of BSc (Honours) in Computing, that is Artificial Intelligence as a Service (AIaaS). My proposed solution intends to simplify the development of machine learning models and the consumption of AI services by clients. Additionally, this document outlines the project's requirements, plan and the software development process followed to accomplish the goals and objectives stablished on this report.

I would like to highlight the most important sections on this document. On the beginning of the document there is an introduction section that provides a general overview of the project's goals and explain what technologies would be used and how. Following, it is provided a requirements specification section, which includes functional and non-functional requirements. The architecture of proposed solution will be detailed on "Design & Architecture" section. The Graphical User Interface (GUI) section includes descriptive screenshots of the main pages within the implemented solution, offering readers a clear understanding of the solution's functionality.

At the end of the system section, it is provided a test plan and evaluation. These two sections are very important as they define the test plan, evidence of testing and how the solution would be evaluated.

An appendices section is included at the end of the document, which presents all the documentation provided to and validated by the supervisor during the project's life span. This section also includes the project proposal that contains a very detailed description of the proposed solution including a project plan, technologies and software development process chosen. Lastly, the appendices section attaches the reflective journals, which acts as a retrospective analysis of work done during each development cycle.

Relevant links

GitHub

Frontend repository

https://github.com/herreramaxi/MH-AIaaS-UI

Backend repository

https://github.com/herreramaxi/MH-AIaaS

Python scikit-learn code for evaluation

https://github.com/herreramaxi/MH-AIaaS-Python

Cloud

Website (Frontend)

This is the website of the proposed solution, it is the user facing application that allows users to sign in, sign out and perform all the use cases detailed in section "Functional requirements".

URL: https://mh-aiaas.herokuapp.com/

NB: User credentials would be provided via other medium.

Web API (Backend)

The web API provides backend services to the frontend website, such as database storage, JWT token authentication and authorization, prediction services or ML module consumption so on so forth.

URL: https://ec2-3-249-105-85.eu-west-1.compute.amazonaws.com

1.0 Introduction

1.1. Background

Back in 2010 when I started my career as a Software developer, I had a chance to work on a small OCR (Optical character recognition) that was implemented using a neural network. That was my first contact with Artificial Intelligence and since then I started to feel attracted to the field.

Recently, I had the opportunity to work on a ETL in the financial industry. The ETL was quite old, its configuration tool was outdated and quite complex to manage. It was so complex than a team was required for the configuration. One of my tasks was to modernize the ETL following a graphical approach to represent data transformations, very similar to SQL Server Integration Services (SSIS), where you have a designer that allows you to represent data flows in the form of interconnected nodes. As main technologies I used C# .NET, React and SQL Server. That experience is one of the reasons for me to be interested on building a graphical automation tool.

Artificial Intelligence as a Service (AIaaS) is a mix of the technologies that I am attracted by and without a doubt, working on this project would be an enriching experience and will deepen my knowledge not only on backend technologies, but also on front-end, AI and Cloud technologies.

To meet my objectives, I have created a detailed plan of work following Kanban as my software development framework. I have detailed principal Milestones (in the form of Releases), epics (large piece of work) broken down into sub tasks with start date, end date and an estimated duration. The plan includes a roadmap that offers a Gantt view of the project, which is used for planning long piece of work and dependency management. I will use Jira Board for tasks management, which will be shared with my project supervisor to give a real time progress status.

1.2. Aims

The main goal for this project is to build a cloud-based and distributed solution that would provide and facilitate the use of Artificial Intelligence services to enterprise applications.

1.2.1. Cloud-based solution

This solution is intended to function on the Cloud as a distributed system, and it would follow a Software as a Service approach.

1.2.2. Provide and facilitate the use of Artificial Intelligence services

My implementation of Artificial Intelligence as a Service (or AIaaS) will allow to create machine learning (ML) models and make them available to be consumed by clients from the Cloud.

The creation of a ML model would follow a graphical approach and will support the implementation of a typical machine learning workflow [1] containing tasks such as: data input, data pre-processing, selection of features and target, selection of most fitted ML algorithm, training, and model evaluation.

1.2.3. Enterprise applications as customers

This product is not intended to be used by any user from Internet, it is intended for enterprise applications. And by enterprise applications I mean applications or systems that operates in a corporate environment.

1.3. Technology

1.3.1. Backend

The backend technologies used on the Web API provides all the required services by the UI and clients via endpoints.

- C# .NET 6

- ML.NET, machine learning for .NET
- Metaprogramming in .NET [2]
- SignalR, Web Sockets
- Entity Framework as ORM
- SQL Server
- Nginx
- Docker containers
- CQRS MediatR
- FluentValidation
- Automapper

1.3.2. Frontend

The frontend technologies would be used on the UI of solution. I intend to use the following technologies:

- Angular 15 [3]
- HTML
- CSS
- JavaScript
- Typescript
- Angular Material
- Angular Kendo Grids
- NGRX: state management for Angular

1.3.3. Principal libraries

- ML.NET: Machine learning algorithms for C#
- ng-flowchart*:* A lightweight Angular Library for building drag and drop flow charts

1.3.4. Cloud

- Auth0 Identity and Access Management (IAM)
- AWS EC2 Elastic Compute Cloud
- RDS Managed Relational Database Service
- CloudWatch Logs
- Heroku

1.3.5. CI/CD

- CircleCI
- Heroku auto deploy

1.4. Structure

This document presents my final project that is Artificial Intelligence as a Service (AIaaS), explains what problems is trying to solve and how is going to achieve it. The requirements specification provides a clear definition and description of functional and non-functional requirements, followed by an architecture section that explains technological details about how the solution could be implemented. The document also includes a testing and evaluation section. And at the end of the document there is a conclusions section, further development, references, and appendices.

1.4.1. Introduction

This section provides an overview of the proposed solution to implement Artificial Intelligence as a Service (AIaaS), background, objectives, and main technologies to be used.

1.4.2. System

It defines the functional requirements in ranked order highlighting dependencies. Each functional requirement includes a brief description, priority, use case flow and diagram. It also defines non-functional requirements, which are verifiable. Next, the design and architecture are explained, and relevant code snippets are shown on implementation section. Lastly, it is explained the testing strategy and technologies and how the solution could be evaluated.

1.4.3. Conclusions

This section describes the scope of the project, what the solution can do, limitations and general conclusions.

1.4.4. Further development or research

The section outlines what would be the next steps and the direction that should take in case of more resources are allocated.

1.4.5. References

This section presents any relevant reference to be included.

1.4.6. Appendices

Supplementary information is added on this section together with project proposal, reflective journals, and ethic declaration form.

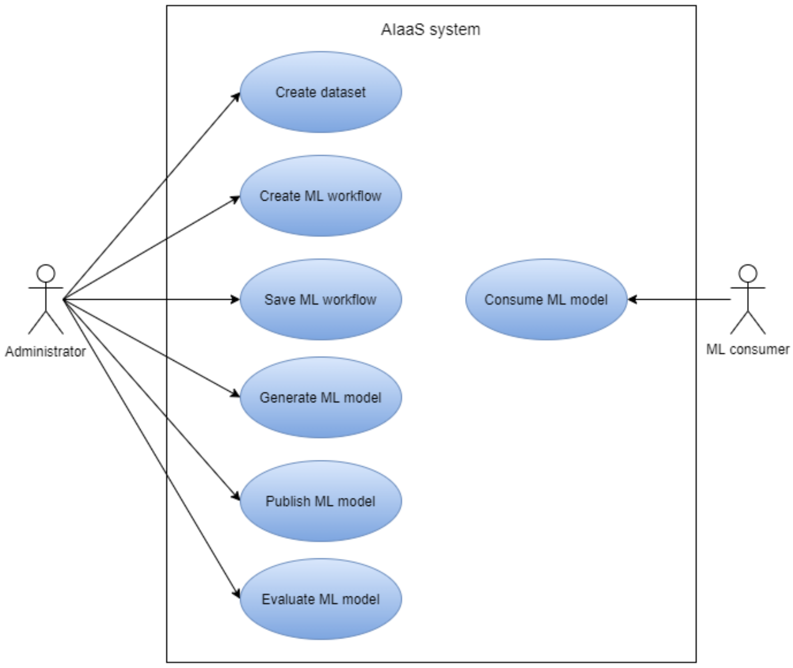
2.0 System

2.1. Requirements

2.1.1. Functional Requirements

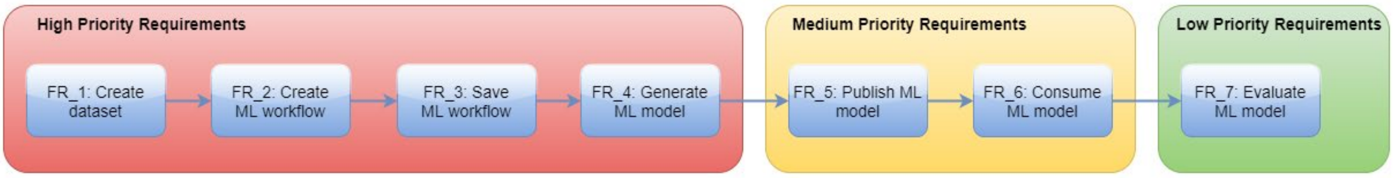
ID	Functional Requirement	Functional Requirement Description
FR_1	Create dataset for machine learning workflow	The user should be able to create a dataset to be used on any machine learning workflow
FR_2	Create machine learning workflow	The user should be able to create a machine learning workflow
FR_3	Save machine learning workflow	The system must allow the user to save the ML workflow
FR_4	Generate machine learning model	The system must allow the user to generate a ML model by running a workflow
FR_5	Publish machine learning model	The user should be able to publish a ML model
FR_6	Consume ML model from endpoint	The user should be able to consume the ML model via endpoint
FR_7	Evaluate ML model	The user should be able to visualize metrics for a generated ML model

2.1.1.1. Use Case Diagram



2.1.1.2. Requirements priority and dependency chart

The following chart shows the priorities and dependencies between the requirements. I have stablished the following three levels of priorities: high priority, medium priority, and low priority.



2.1.1.3. FR_1: Create dataset

2.1.1.3.1. Description & Priority

The user will be able to create a dataset from datasets page. A dataset on this context is basically a file (csv or tsb file) that will be used as data input for ML workflows. Once the dataset is created it would be possible to select features and label, which are essential for machine learning tasks. This requirement is top priority and need to be implemented first because the rest of them requirements depend on it.

Priority: High

2.1.1.3.2. Use Case

FR_1: Create dataset.

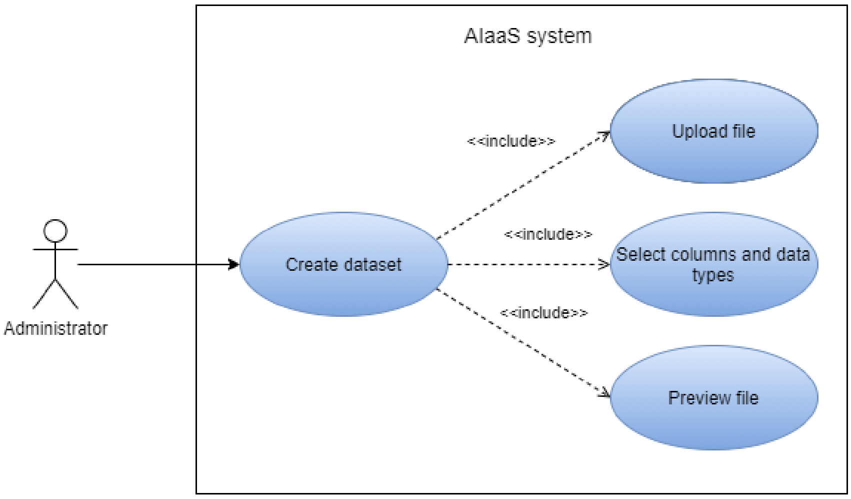
Scope

The scope of this use case is for a system administrator to create a dataset.

Description

This use case describes how the system administrator creates a dataset on the datasets page.

Use Case Diagram



Flow Description

Precondition

A user is logged in into the system as administrator and has accessed to the datasets page.

Activation

This use case starts when an administrator clicks on “create new dataset”.

Main flow

1. The administrator clicks on “create new dataset”.
2. The system redirects the administrator to the dataset page.
3. The administrator set a dataset name.
4. The administrator selects a file to be uploaded.
5. The administrator clicks on upload button.
6. The system shows a preview of file content.
7. The administrator selects columns to be included and data types.
8. The administrator clicks in save button. [a1 Errors on validations]
9. The system notifies the administrator that the dataset has been saved successfully.

Alternate flow

A1: Errors on validations

1. The system displays validation errors.
2. The administrator applies changes to fix the errors.
3. The use case continues at position 8 of the main flow.

Exceptional flow

N/A

Termination

The system saves the dataset and inform to the user that the operation has been successful.

Post condition

The system goes into a wait state.

2.1.1.4. FR_2: Create machine learning workflow

2.1.1.4.1. Description & Priority

The user will be able to create a machine learning workflow on the designer provided by the user interface. This requirement is essential and has top priority because most of the other requirements depend on it. On this context, a workflow is a graphic representation of tasks to be executed in a certain order to generate a machine learning model. Examples of workflow tasks are clean data, normalize data, features and target selection, split data (test and train datasets), training, evaluation so on so forth.

Priority: High

2.1.1.4.2. Use Case

FR_2: Create machine learning workflow.

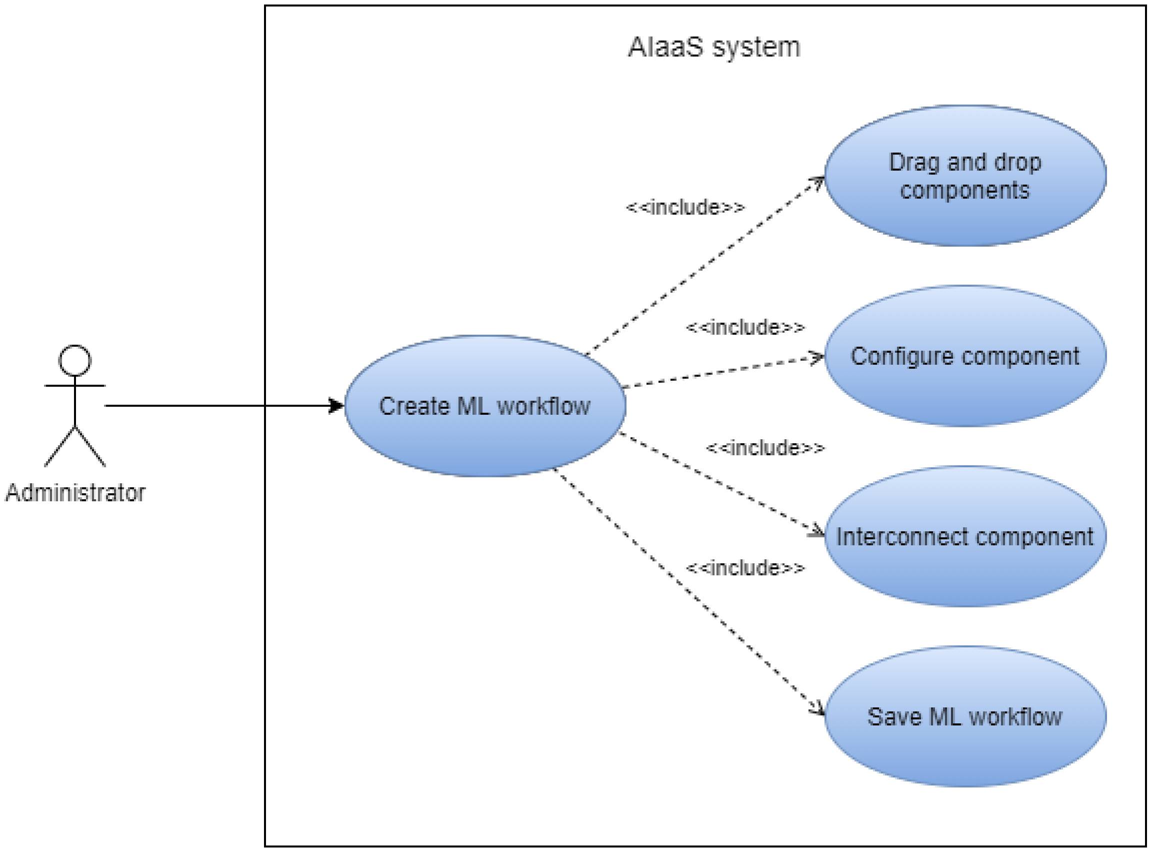
Scope

The scope of this use case is for a system administrator to create a machine learning workflow.

Description

This use case describes how the system administrator creates a machine learning workflow on the designer provided by the user interface.

Use Case Diagram



Flow Description

Precondition

A user is logged-in into the system as administrator and has accessed to workflow designer page.

Activation

This use case starts when an administrator clicks on “create new workflow”.

Main flow

1. The administrator clicks on “create new workflow”.
2. The system redirects the administrator to the designer or ML studio.
3. The administrator drags and drop components into the designer.
4. The administrator interconnects the component with other workflow’s components.
5. The administrator configures the new component.
6. The administrator repeats steps 3, 4, and 5 if more components are required to be added.
7. The administrator clicks in save. [a1 Errors on validations]
8. The system notifies the administrator that the workflow has been saved successfully.

Alternate flow

A1: Errors on validations

4. The system displays validation errors.
5. The administrator applies changes to fix the errors.
6. The use case continues at position 7 of the main flow.

Exceptional flow

N/A

Termination

The system saves the ML workflow and inform the user that the operation has been successful.

Post condition

The system goes into a wait state.

2.1.1.5. FR_3: Save machine learning workflow

2.1.1.5.1. Description & Priority

Once the ML workflow is created, the user will be able to save the workflow into the system and any modification on it. This will allow the user to restart its work after logging out of the system.

Priority: High

2.1.1.5.2. Use Case

FR_3: Save machine learning workflow.

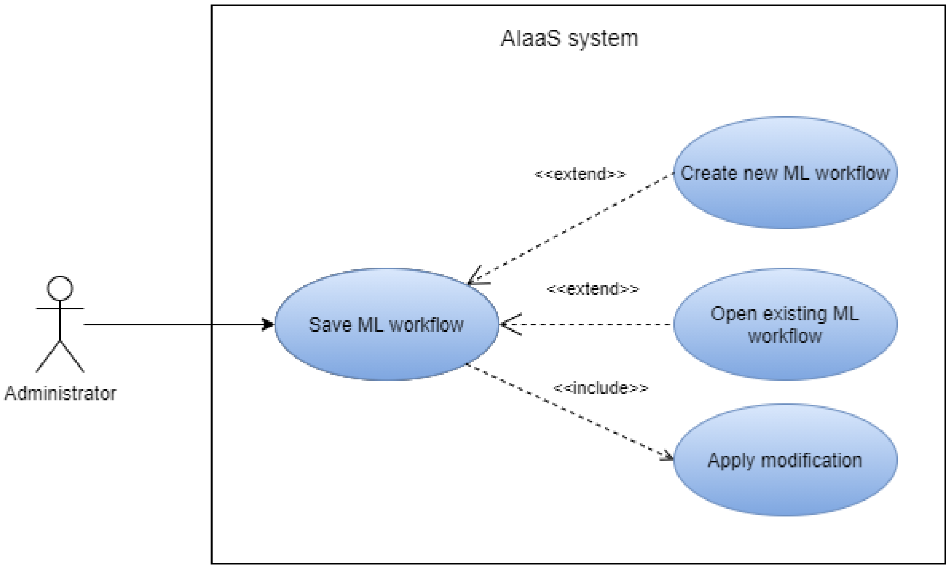
Scope

The scope of this use case is for an administrator to save a machine learning workflow into the system.

Description

This use case describes how the administrator saves a machine learning workflow on the designer provided by the user interface.

Use Case Diagram



Flow Description

Precondition

A user is logged-in into the system as administrator and has created a ML workflow or opened an existing one.

Activation

This use case starts when an administrator applies changes to the ML workflow and clicks on save button.

Main flow

1. The administrator applies changes to the ML workflow.
2. The administrator clicks on “save” button. [a1 errors on validations]
3. The system stores the changes made on the ML workflow.
4. The system shows a successful message.

Alternate flow

A1: Errors on validations

1. The system displays validation errors.
2. The administrator applies changes to fix the errors.
3. The use case continues at position 2 of the main flow.

Exceptional flow

N/A

Termination

The system saves the ML workflow and inform to the user that the operation has been successful.

Post condition

The system goes into a wait state.

2.1.1.6. FR_4: Generate machine learning model

2.1.1.6.1. Description & Priority

The administrator will be able to generate a ML model by running a ML workflow. Client will be able to consume the ML model from an endpoint once the model is published.

Priority: High

2.1.1.6.2. Use Case

FR_4: Generate machine learning model

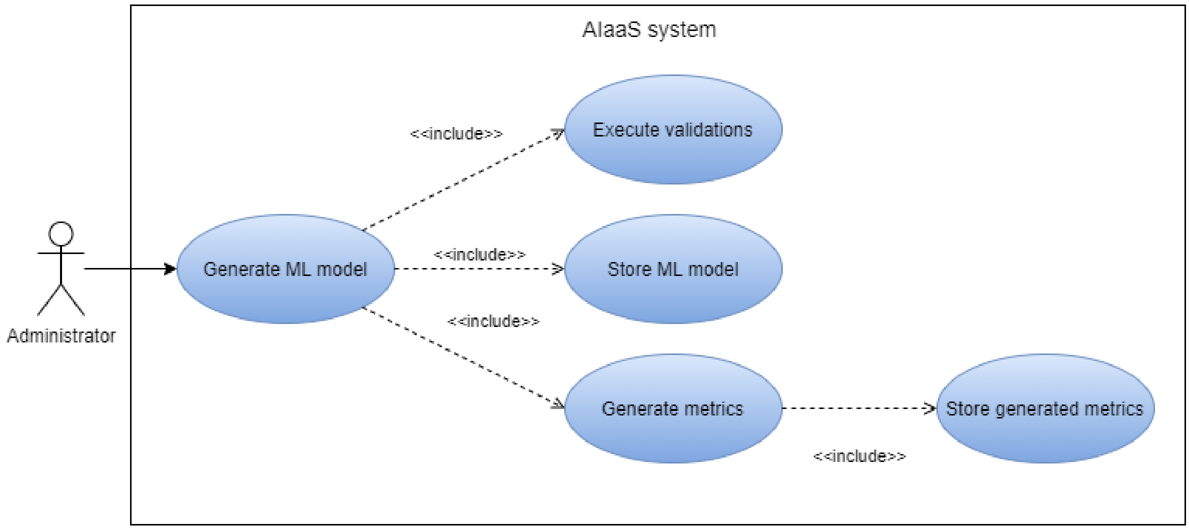
Scope

The scope of this use case is for an administrator to run a machine learning workflow.

Description

This use case describes how the administrator generates a machine learning model by running a ML workflow.

Use Case Diagram



Flow Description

Precondition

A user is logged-in into the system as administrator and has created a ML workflow or opened an existing one.

Activation

This use case starts when an administrator clicks on run button.

Main flow

1. The administrator clicks on run ML workflow.
2. The system runs workflow validations. [a1 errors on validations]
3. The system generates a ML model.
4. The system generates metrics for the model.
5. The system stores the generated model and metrics.
6. The system shows a successful message.

Alternate flow

A1: Errors on validations

1. The system displays validation errors.
2. The administrator applies changes to fix the errors.
3. The use case continues at position 1 of the main flow.

Exceptional flow

N/A

Termination

The system generates the ML model and informs the user that the operation has been successful.

Post condition

The system goes into a wait state.

2.1.1.7. FR_5: Publish machine learning model

2.1.1.7.1. Description & Priority

The administrator will be able to publish a ML model, which was created previously by generating a machine learning model. Once a model is published, it will be available to be used on a ML endpoint, which will be automatically created when publishing the model.

Priority: Medium

2.1.1.7.2. Use Case

FR_5: Publish machine learning model.

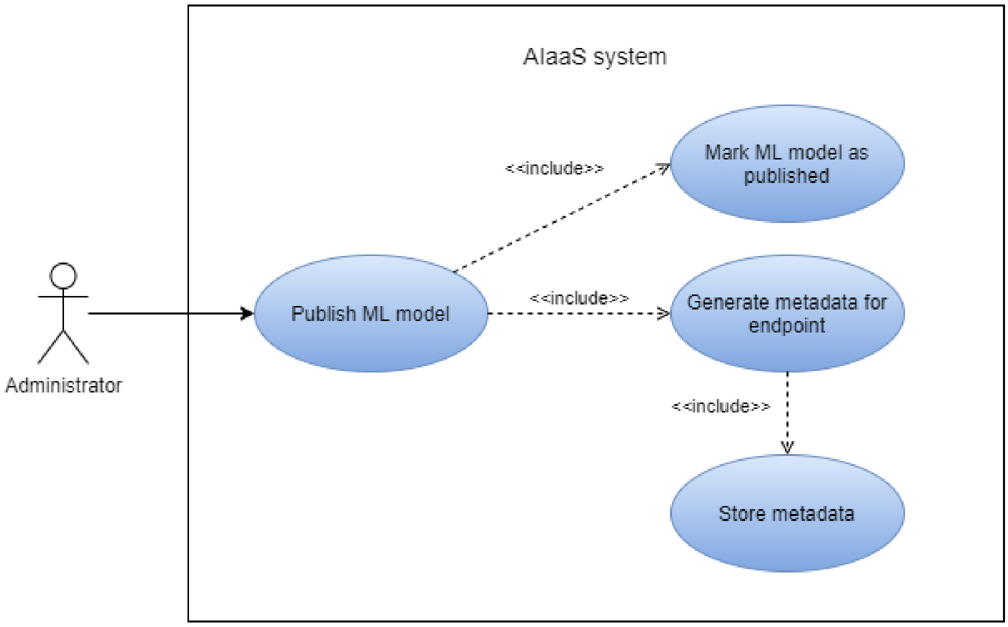
Scope

The scope of this use case is for an administrator to publish a machine learning model into the system.

Description

This use case describes how the administrator publishes a machine learning model from a previously generated one.

Use Case Diagram



Flow Description

Precondition

A user is logged-in into the system as administrator and has generated a ML model.

Activation

This use case starts when an administrator clicks on publish button within model page or the designer.

Main flow

1. The administrator clicks on publish button. [e1 error when publishing model]
2. The system marks the ML model as published.
3. The system generates metadata for the model.
4. The system store metadata generated.

Alternate flow

N/A

Exceptional flow

E1: Error when publishing model

1. The system shows an error message with details of exceptional error.
2. The use case continues at position 1 of the main flow.

Termination

The system publishes the ML model and inform the user that the operation has been successful.

Post condition

The system goes into a wait state.

2.1.1.8. FR_6: Consume ML model from endpoint

2.1.1.8.1. Description & Priority

The client will be able to consume a ML model via endpoint. This means that AI services (such as price prediction) would be provided by this endpoint utilising the previously generated and published ML model.

Priority: Medium

2.1.1.8.2. Use Case

FR_6: Consume ML model from endpoint

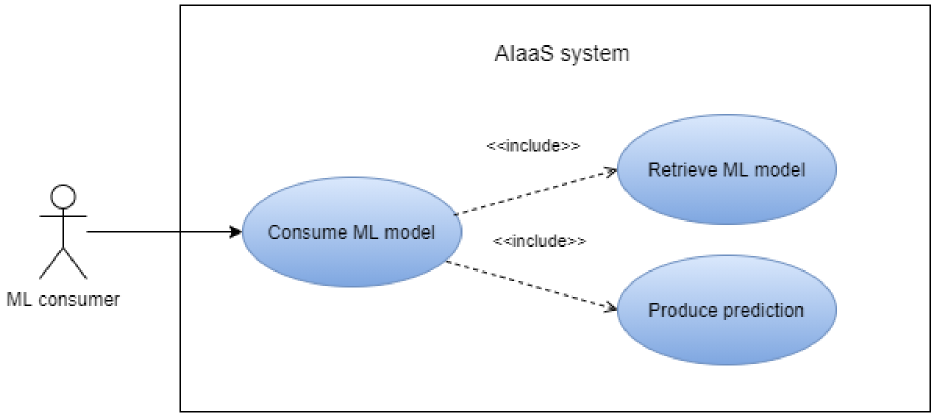
Scope

The scope of this use case is for a ML consumer to consume a machine learning model that was previously published to get predictions for a given input.

Description

This use case describes how the ML consumer consumes a machine learning model to get predictions for a given input

Use Case Diagram



Flow Description

Precondition

The user is registered into the system and is assigned to the ML consumer role.

Activation

This use case starts when a user sends a request to the "consume ML model" endpoint.

Main flow

1. The consumer user sends a request to "consume ML model" endpoint.
2. The system retrieves the latest published ML model. [a1 model not found]
3. The system generates a prediction from the ML model and user input.
4. The system returns a successful response including the generated prediction.

Alternate flow

A1: Model not found

1. The system returns a not found response to client finalizing the main flow.

Exceptional flow

N/A

Termination

The system returns prediction values from the user's input and model associated.

Post condition

The system goes into a wait state.

2.1.1.9. FR_7: Evaluate machine learning model

2.1.1.9.1. Description & Priority

The administrator will be able to evaluate a ML model, which was created previously by running a workflow. From the user perspective, evaluating a model means to visualize model metrics that describes how well the model behaves, the accuracy and the quality of it. Some examples of model metrics are:

- Root Mean Squared Error (RMSE)
- R-Squared
- Prediction errors
- Predicted vs Actual

Priority: Low

2.1.1.9.2. Use Case

FR_7: Evaluate machine learning model.

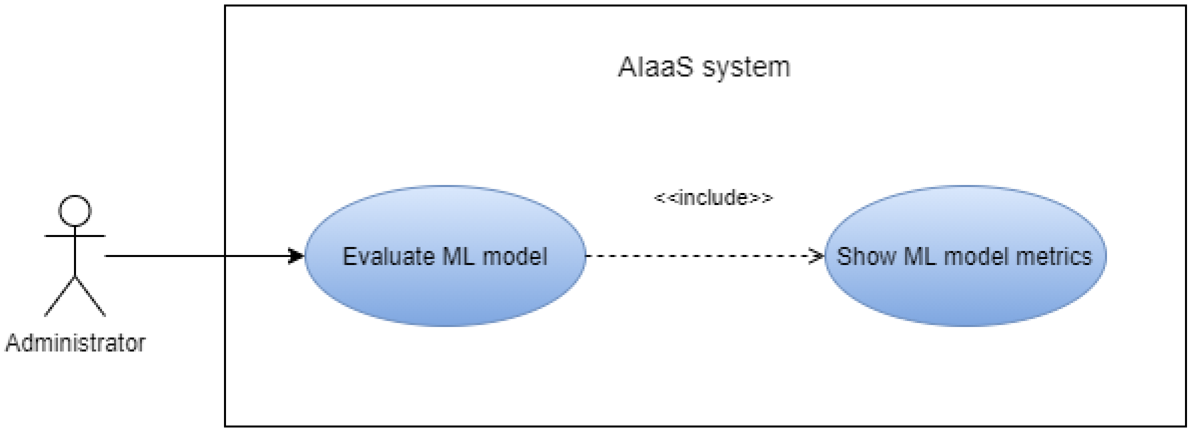
Scope

The scope of this use case is for an administrator to evaluate a machine learning model that was previously generated.

Description

This use case describes how the administrator evaluates a machine learning model from a previously generated one.

Use Case Diagram



Flow Description

Precondition

A user is logged-in into the system as administrator and has generated a ML model.

Activation

This use case starts when an administrator clicks on “Visualize” option on Evaluate workflow operator or from Model page under metrics section.

Main flow

1. The administrator clicks on “Visualize” option on Evaluate workflow operator.
2. The system shows the metrics that were previously calculated for the generated model.

Alternate flow

N/A

Exceptional flow

N/A

Termination

The system shows available metrics for the generated model.

Post condition

The system goes into a wait state.

2.1.2. Data Requirements

2.1.2.1. Database

This solution will use SQL Server as main database, specifically I would use AWS RDS for SQL Server.

2.1.2.2. Authentication and authorization

Identity and access management (IAM) will be delegated into Auth0, therefore there would be not necessary to store authentication/authorization data into the system.

2.1.2.3. Machine learning Workflow storage

For the storage of workflows, I would use a “workflows” table that would store workflows as JSON objects. In that way, I will be able to use a relational database such as Microsoft SQL Server, to easily store graph structures into a NVARCHAR(max) column. The benefit of this strategy is that it would not be necessary to translate a graph structure (nodes and edges) into relational structure (tables and foreign keys) allowing me to have just one column (named “Data”) for storing the JSON object.

2.1.2.4. Machine learning Model storage

To persist machine learning models I will use database, it would be stored in a VARBINARY(MAX) column on MLModels table. The reason is because the model files are very small in size, for example for a dataset of around 150MB the model size is less than 100KB. This allow to have a fast access to the model, which is needed while requesting predictions service.

2.1.2.5. Cloud Storage Services for large files

The solution proposed is designed to handle large datasets that in a later stage are processed for machine learning workflows. To store those large files, the system uses AWS S3 as cloud storage services with versioning enabled. In addition, each operation on a given workflow, produces intermediate data. This intermediate data, which is also known as binary data (".idv" file extension), provides an output file for each transformation that is stored in the system that can be used to visualize a preview of a transformation. As this binary data can also be large (for a 150MB dataset the intermediate data files have a size between 30 and 50MB), they are stored in AWS S3.

The strategy used to store on S3 was to generate a key with a random prefix using GUID and save the key on database using a column named S3Key. That allows the retrieval of the file using the previously generated key. Below are listed the tables that uses this strategy:

- FileStorages: Used for the storage of original dataset files.
- DataViewFiles: Used for the storage of binary data derived from dataset. This helps to load a dataset and apply transformations on the dataset data.
- WorkflowDataViews: Used for the storage of intermediate data, that is generated by workflow operators.

2.1.2.6. Tables required

- ColumnSettings
- Datasets
- DataViewFiles
- FileStorages
- MLModels
- ModelMetrics
- WorkflowDataViews
- WorkflowNodeRunHistory
- WorkflowRunHistory
- Workflows

2.1.2.7. User Requirements

The main requirement for a user would be to have access to internet. The user would be required to enter a registered email and password that has already a role assigned to it and therefore it has granted a set of permissions under role-based access control (RBAC). The user would have the option to sign up as a new user or to use an existing Gmail account. In that case, the new user account must be assigned to a role on Auth0 by the administrator.

2.1.2.8. Environmental Requirements

2.1.2.8.1. Security

Security will be provided in the form of role-based access control, where a user will be assigned to a role and a role will have associated a set of granted permissions. This role-based access control will be delegated into Auth0.

Another important aspect of security are the endpoints. All the endpoints will be secured by JSON Web Token (JWT), which will be generated by Auth0. The token would be validated by the Web API and verify the required claims for each endpoint. That means that the requestor user will be required to have specific claims when calling the Web API endpoints. In addition to the backend validations, the Frontend also will validate the claims from the JWT token to grant or denied access to specific screens or components.

2.1.2.8.2. Sensitive information

All the sensitive information related to environment configuration will be stored as environment variables. Examples of sensitive information are connection strings, passwords, certificates, clientId and so on so forth.

2.1.2.8.3. Cloud

The solution proposed would require a minimum of two computing nodes on the Cloud, one for the Angular application (Frontend) and the other for the .NET Web API (Backend).

2.1.2.8.4. Frontend requirements

The frontend node will have low computational requirements because it will run an Angular application and will not participate on the processing of datasets or workflows. Heroku was chosen to host this Frontend node, and that brings the following advantages:

- Provides a flexible and cheap PaaS.
- Facilitates the configuration of CI/CD pipeline.
- Enables developer to rapidly deploy into the Cloud.

2.1.2.8.5. Backend requirements

The backend node runs a Web API that provides all the required services to the Frontend or Angular application. The Web API node oversees the processing of datasets and workflows, by running compute-intensive processes. To support the high computational requirements, an AWS EC2 was chosen. Below is listed the EC2 specifications:

- EC2 type: t2.micro
- Platform: Amazon Linux 2
- Processor: 1 vCPU
- Ram: 1GB
- Storage: 20GB

2.1.2.8.6. Usability Requirements

2.1.2.8.6.1. User friendliness

The system will be designed to be user friendly, intuitive, simple, and minimalist as possible. Tooltips will be available on HTML components to provide context information to the user.

2.1.2.8.6.2. Learnability

The UI should guide the user when interacting with it, and it should allow the user to learn from the interaction with the system.

2.1.2.8.7. Performance

2.1.2.8.7.1. Response time for dataset creation

The response time when creating a dataset, it should be below 30 seconds for files under 5MB.

2.1.2.8.7.2. Response time for generating a machine learning model

The response time when generating a ML model, it should be below 30 seconds for files under 5MB.

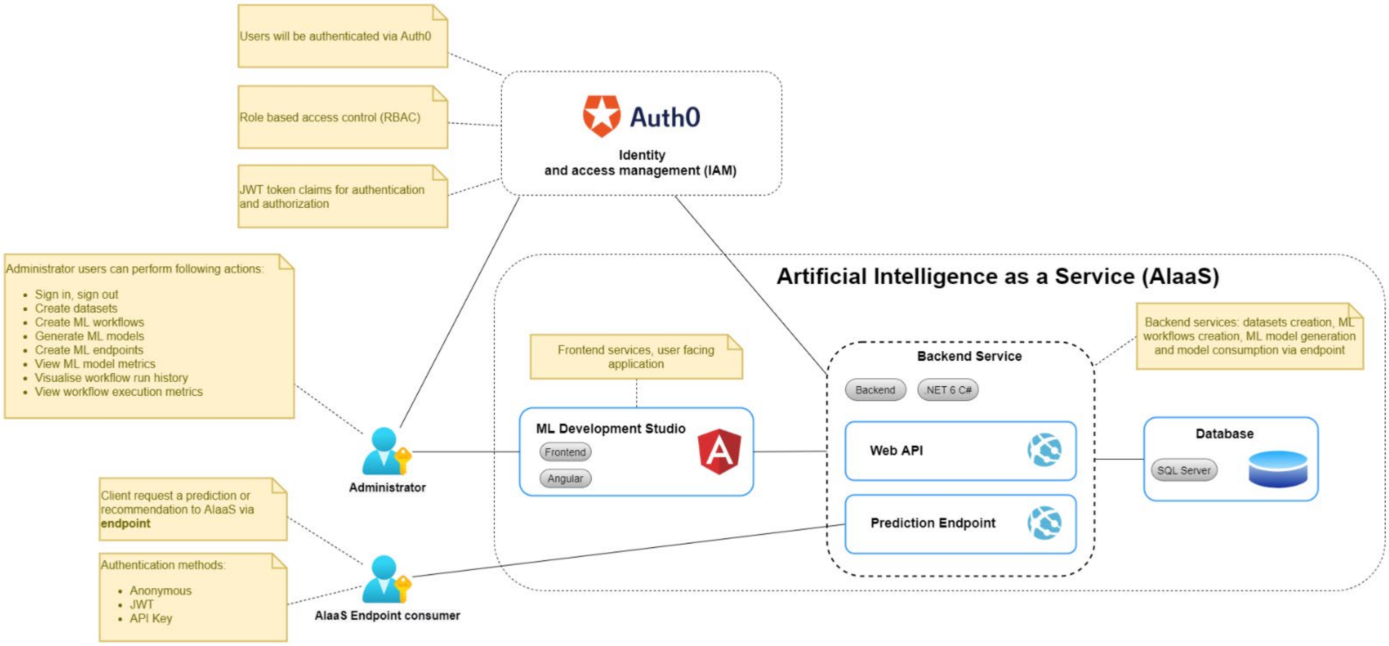
2.1.2.8.7.3. Response time for AIaaS endpoint (or prediction endpoint)

The response time for AIaaS endpoint, it should be below 5 seconds for files under 5MB.

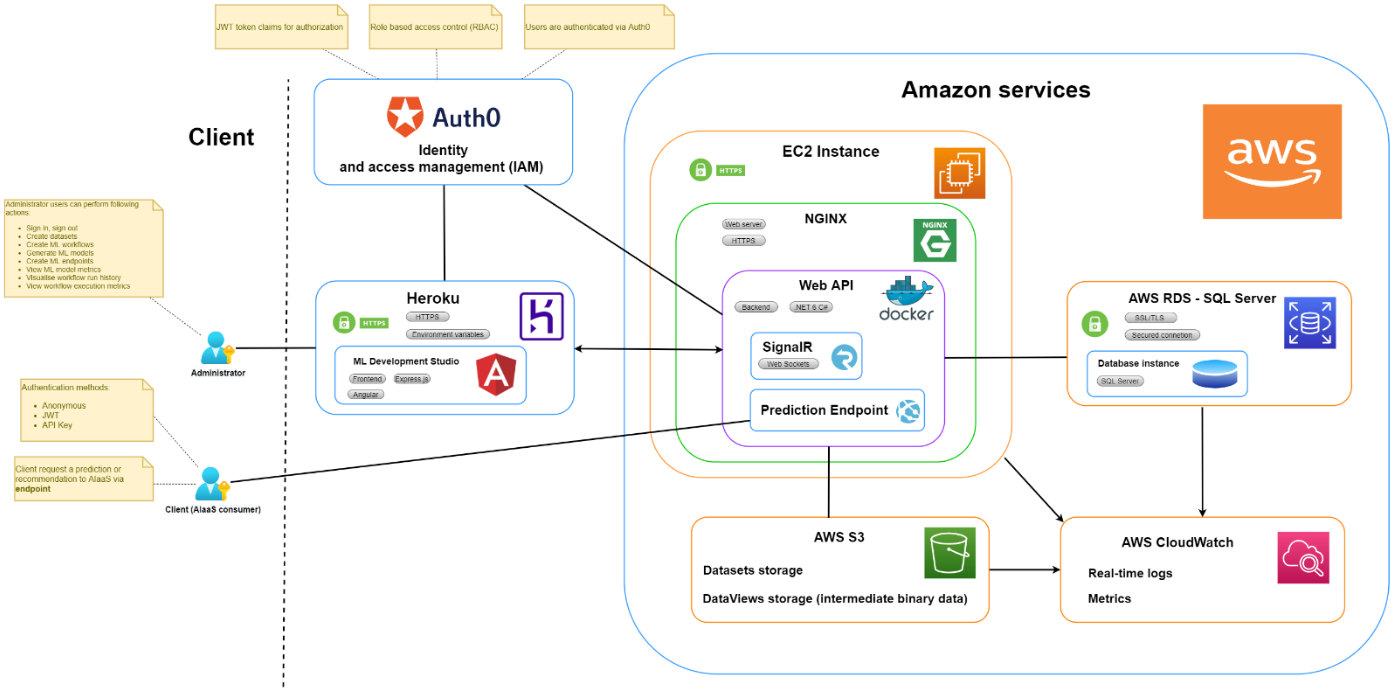
2.2. Design & Architecture

The proposed solution to Artificial Intelligence as a service (AIaaS) is implemented as a distributed system. On this architecture there are two nodes, one for backend services or Web API and the other for frontend or UI which is the user facing web application. Below it is shown two views of the system architecture, the first is a simplified representation, and the second presents a more detailed view including Cloud technologies.

2.2.1. Distributed system architecture diagram

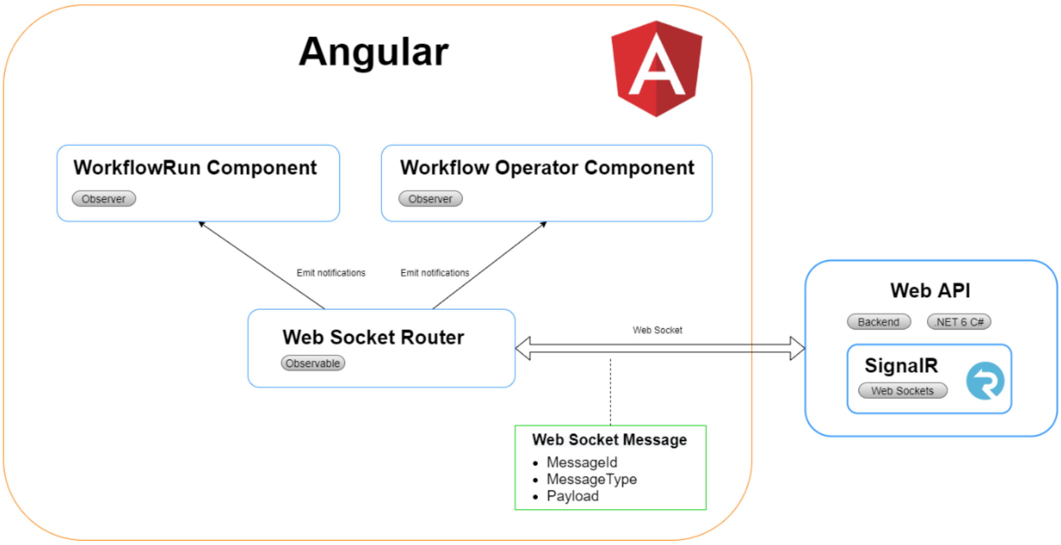

Users will be authenticated via Auth0
Role based access control (RBAC)
JWT token claims for authentication and authorization
Auth0
Identity
and access management (IAM)
Administrator users can perform following actions:
Sign in, sign out
Create datasets
Create ML workflows
Generate ML models
Create ML endpoints
View ML model metrics
Visualise workflow run history
View workflow execution metrics
Artificial Intelligence as a Service (AIaaS)
Backend services: datasets creation, ML workflows creation, ML model generation and model consumption via endpoint
Frontend services, user facing application
Backend Service
Backend
NET 6 C#
ML Development Studio
Frontend
Angular
Web API
Database
SQL Server
Prediction Endpoint
Administrator
Client request a prediction or recommendation to AIaaS via endpoint
Authentication methods:
Anonymous
JWT
API Key
AIaaS Endpoint consumer

2.2.2. System architecture and Cloud technologies



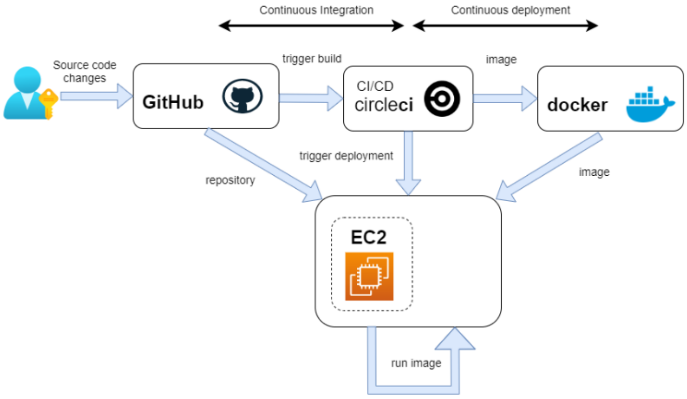
2.2.3. Web Sockets

SignalR is used to enhance the communication between the Frontend and Backend, enabling a bidirectional communication between both nodes. This communication channel is used by the backend to notify relevant events to frontend, such as workflow start processing, workflow finish processing, and events at operator level. A web socket router was implemented on the frontend to efficiently manage multiple message types through a single web socket connection and send notifications to observers that are registered to a particular message type. That allows to minimize the quantity of connections which enables the application to scale.



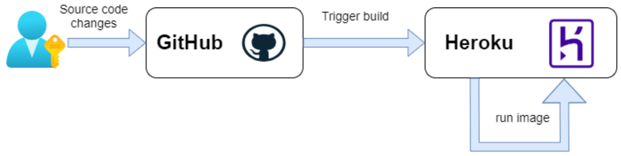
2.2.4. Web API CI/CD pipeline

The Web API service is deployed using a continuous integration and continuous deployment pipeline that is configured on CircleCI. One of the benefits of using CircleCI is that it can be used for free (with certain limits), allows you to define the CI/CD pipeline relatively easy, and the platform is very well documente. The configuration file for the CI/CD can be found on: https://github.com/herreramaxi/MH-AIaaS/blob/main/.circleci/config.yml

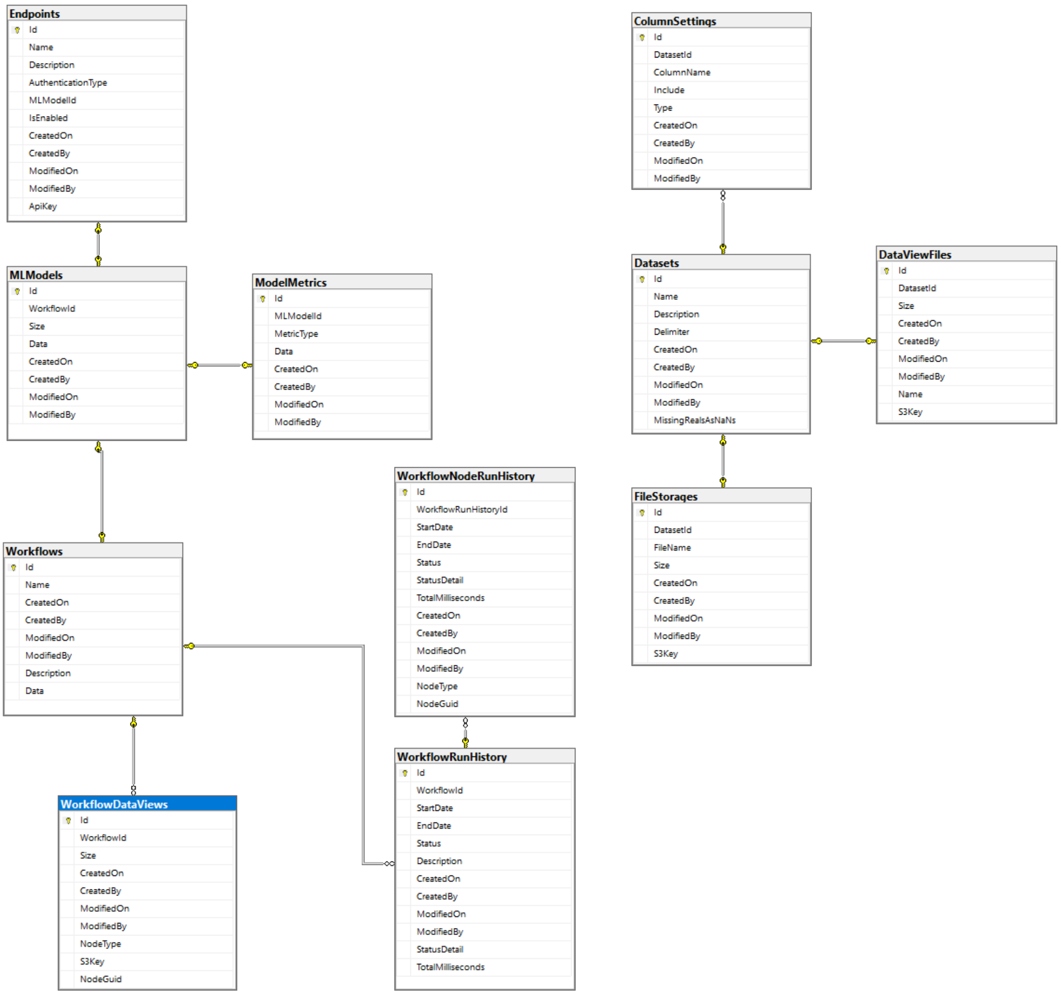


2.2.5. Frontend CI/CD pipeline

For the frontend component initially was configured using CircleCI, but it was moved directly to Heroku automatic deployments to accelerate the software development cycle and have fast deployments. Below can be found a diagram of the CI/CD for the frontend component.



2.2.6. Database entity relationship diagram



2.2.7. System components

2.2.7.1. ML development studio (Frontend)

It is the user facing application and enables an administrator to interact with the system and perform all the use cases detailed on this document (section “Functional Requirements”). It is implemented on Angular 15 as a single page application (SPA), using standards components from Angular material and Kendo Grid. The frontend is a lightweight webapp and does not require intensive computing power. Therefore, it is deployed on Heroku, which provides some key features such as fast deployments, HTTPS certificates management, protection of sensitive configuration using environment variables and a nice domain name. The webapp is integrated with Auth0 and enables users to sign-in with authentication providers such as Gmail, Microsoft or just by user and password.

2.2.7.2. Key features

- Angular 15 Single Page Application (SPA)
- Integrated with Auth0 Identity and Access Management (IAM)

- Utilization of Web Sockets for handling backend notifications
- Angular Material and Angular Kendo grids
- Deployed into Heroku
- HTTPS
- Implements the following use cases
    - Sign in, sign out
    - Create datasets
    - Create ML workflows
    - Generate ML models by running a workflow
    - Publish ML models
    - Create ML endpoints
    - *View ML model metrics*
    - *Visualize intermediate data for each workflow transformation*
    - *Visualize workflow metrics*

2.2.7.3. Web API (Backend)

It provides all the computational services required for the ML workflow designer (Frontend) and the predict endpoint that allows users to consume ML models such as price predictions. The web API delegates on Auth0 the authentication and authorization, which is based on JWT token, where each user possesses determined user Claims. As it is shown on the "System architecture and Cloud technologies" diagram, this node is deployed into an EC2 and runs on a docker container, which takes advantage of some important features from Docker such as isolation, security of host environment and protection of sensitive information, reduced dependencies on host environment, and easy service management.

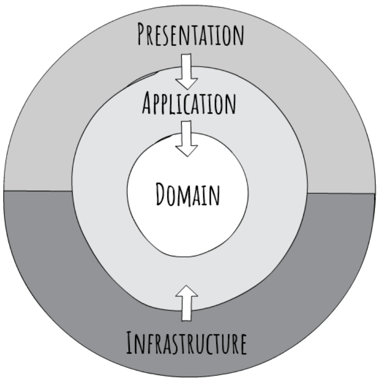
Another important feature from the backend node is that the EC2 makes use of NGINX as a high performing web server and reverse proxy, to redirect external request to the web API that runs on a Docker container. Additionally, the NGINX webserver and the web API use a SLL offloading strategy to handle HTTPS traffic simplifying the web API configuration and offering a secure and reliable connection to clients.

In relation to database access, the web API offers a secure access to data storage by using AWS RDS for SQL Server. Only the EC2 registered can have access to the database, and equally important AWS RDS uses SSL/TLS to encrypt a connection to a DB instance.

The Web API has been developed following best practices, such as SOLID, design patterns [5], architectural patterns [6], and clean architecture. Latest technologies are being used on this component such as .NET 6, ML.NET, ML.AutoML, SignalR web sockets, Entity Framework, Ardalis.Specification, Automapper, AWSSDK.S3, and so on so forth.

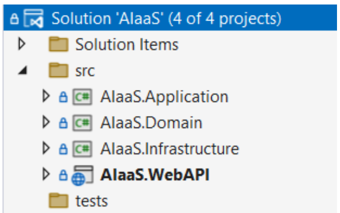
2.2.7.3.1. Clean Architecture design

The backend component was implemented on .NET C# following clean architecture design, which provides a guideline on how to organize the whole application splitting the software into different layers and enforcing a defined and clear separation of concerns. Specifically, I have followed the Jayson Tayler [7] and Ardalis [8] approach to clean architecture, that can be described by following diagram:



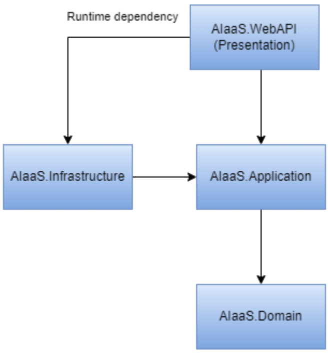
Clean architecture diagram [7]

If we look at the .NET solution, we will see that the project structure follows clean architecture and that can also be seen on Github repository https://github.com/herreramaxi/MH-AIaaS/tree/main/src and on the solution explorer with Visual Studio:



Solution 'AIaaS' (4 of 4 projects)
- Solution Items
- src
  - AIaaS.Application
  - AIaaS.Domain
  - AIaaS.Infrastructure
  - **AIaaS.WebAPI**
- tests

At the assembly level we can corroborate that the clean architecture was followed, which is shown on below diagram:

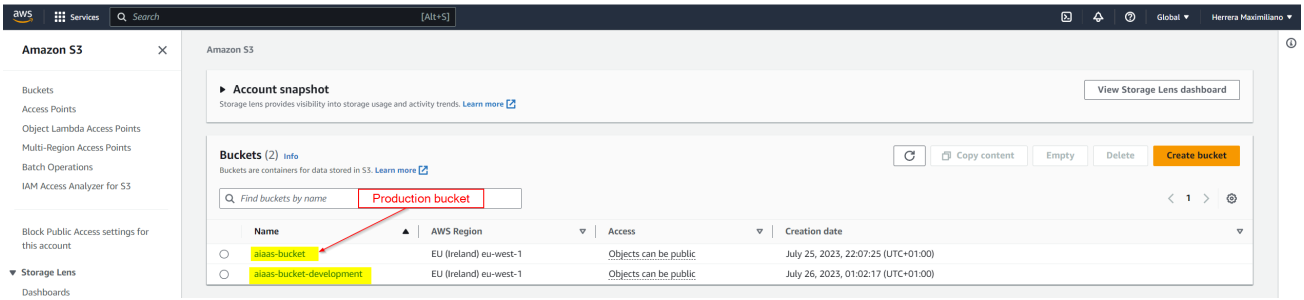


2.2.7.3.2. Key features

- Clean Architecture
- Implements CQSR (Command and Query Responsibility Segregation) pattern
- Integrated with Auth0
- JWT authentication and authorization
- SignalR Web Sockets
- Docker container
- NGINX (web server, reverse proxy, and HTTPS configuration)
- SSL offloading for HTTPS traffic
- Secure access to AWS RDS SQL Server instance
- Monitored by AWS CloudWatch
- File storage on AWS S3

2.2.7.4. AWS S3

AWS S3 [9] is an object repository provided as a service by Amazon and allows the storage and retrieval of files which are stored into buckets. I have configured two environments for the buckets, one for development and one for production:



Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets are containers for data stored in S3. Learn more

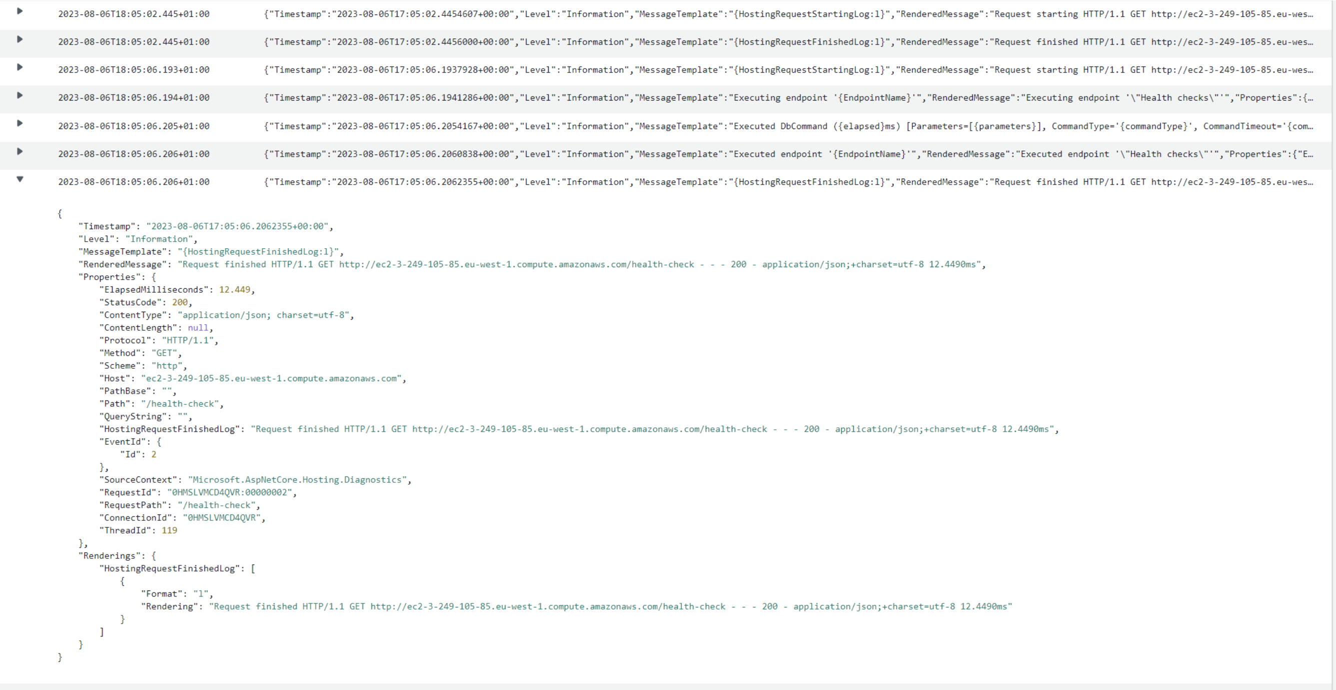
	Name	AWS Region	Access	Creation date
○	aiaas-bucket	EU (Ireland) eu-west-1	Objects can be public	July 25, 2023, 22:07:25 (UTC+01:00)
○	aiaas-bucket-development	EU (Ireland) eu-west-1	Objects can be public	July 26, 2023, 01:02:17 (UTC+01:00)

The S3 storage service is used to store dataset files (original file and binary representation IDataView) and intermediate data (binary files .idv) which is produced by executing a workflow operator. The use of S3 provides an enhanced performance on the system while processing medium or large files, compared to the initial design where files were stored into database. It is worth to mention that S3 versioning is enabled, that allows to have for a given file different versions of it.

2.2.7.5. AWS CloudWatch

Amazon CloudWatch [10] is used by the system to send logs from the Web API application to a LogGroup on AWS. The collected logs from AWS CloudWatch can be visualized in real-time, providing a powerful tool to troubleshoot or just to monitor the current state of the system. Below it is shown event logs coming from the Web API:

2.2.7.5.1. Logs

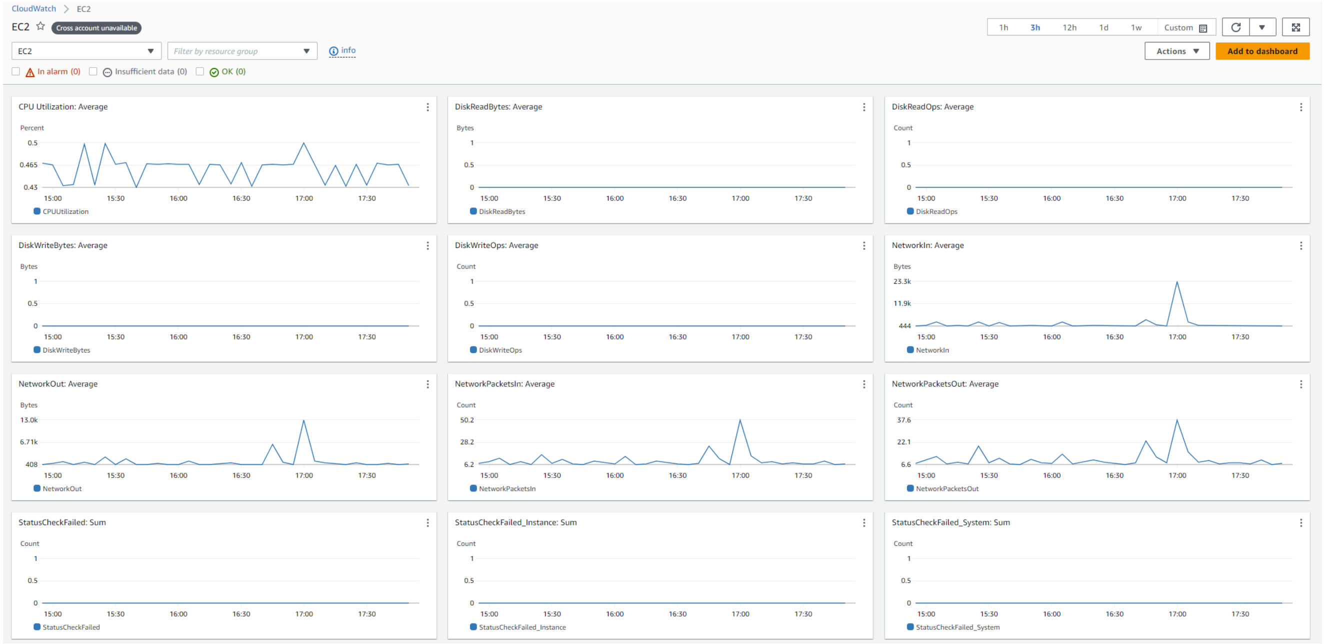


```
▶   2023-08-06T18:05:02.445+01:00    {"Timestamp":"2023-08-06T17:05:02.4454607+00:00","Level":"Information","MessageTemplate":"{HostingRequestStartingLog:l}","RenderedMessage":"Request starting HTTP/1.1 GET http://ec2-3-249-105-85.eu-wes…
▶   2023-08-06T18:05:02.445+01:00    {"Timestamp":"2023-08-06T17:05:02.4456000+00:00","Level":"Information","MessageTemplate":"{HostingRequestFinishedLog:l}","RenderedMessage":"Request finished HTTP/1.1 GET http://ec2-3-249-105-85.eu-wes…
▶   2023-08-06T18:05:06.193+01:00    {"Timestamp":"2023-08-06T17:05:06.1937928+00:00","Level":"Information","MessageTemplate":"{HostingRequestStartingLog:l}","RenderedMessage":"Request starting HTTP/1.1 GET http://ec2-3-249-105-85.eu-wes…
▶   2023-08-06T18:05:06.194+01:00    {"Timestamp":"2023-08-06T17:05:06.1941286+00:00","Level":"Information","MessageTemplate":"Executing endpoint '{EndpointName}'","RenderedMessage":"Executing endpoint '\"Health checks\"'","Properties":{…
▶   2023-08-06T18:05:06.205+01:00    {"Timestamp":"2023-08-06T17:05:06.2054167+00:00","Level":"Information","MessageTemplate":"Executed DbCommand ({elapsed}ms) [Parameters=[{parameters}], CommandType='{commandType}', CommandTimeout='{com…
▶   2023-08-06T18:05:06.206+01:00    {"Timestamp":"2023-08-06T17:05:06.2060838+00:00","Level":"Information","MessageTemplate":"Executed endpoint '{EndpointName}'","RenderedMessage":"Executed endpoint '\"Health checks\"'","Properties":{"E…
▼   2023-08-06T18:05:06.206+01:00    {"Timestamp":"2023-08-06T17:05:06.2062355+00:00","Level":"Information","MessageTemplate":"{HostingRequestFinishedLog:l}","RenderedMessage":"Request finished HTTP/1.1 GET http://ec2-3-249-105-85.eu-wes…

    {
        "Timestamp": "2023-08-06T17:05:06.2062355+00:00",
        "Level": "Information",
        "MessageTemplate": "{HostingRequestFinishedLog:l}",
        "RenderedMessage": "Request finished HTTP/1.1 GET http://ec2-3-249-105-85.eu-west-1.compute.amazonaws.com/health-check - - - 200 - application/json;+charset=utf-8 12.4490ms",
        "Properties": {
            "ElapsedMilliseconds": 12.449,
            "StatusCode": 200,
            "ContentType": "application/json; charset=utf-8",
            "ContentLength": null,
            "Protocol": "HTTP/1.1",
            "Method": "GET",
            "Scheme": "http",
            "Host": "ec2-3-249-105-85.eu-west-1.compute.amazonaws.com",
            "PathBase": "",
            "Path": "/health-check",
            "QueryString": "",
            "HostingRequestFinishedLog": "Request finished HTTP/1.1 GET http://ec2-3-249-105-85.eu-west-1.compute.amazonaws.com/health-check - - - 200 - application/json;+charset=utf-8 12.4490ms",
            "EventId": {
                "Id": 2
            },
            "SourceContext": "Microsoft.AspNetCore.Hosting.Diagnostics",
            "RequestId": "0HMSLVMCD4QVR:00000002",
            "RequestPath": "/health-check",
            "ConnectionId": "0HMSLVMCD4QVR",
            "ThreadId": 119
        },
        "Renderings": {
            "HostingRequestFinishedLog": [
                {
                    "Format": "l",
                    "Rendering": "Request finished HTTP/1.1 GET http://ec2-3-249-105-85.eu-west-1.compute.amazonaws.com/health-check - - - 200 - application/json;+charset=utf-8 12.4490ms"
                }
            ]
        }
    }
```

2.2.7.5.2. Metrics

AWS CloudWatch collects and provides a dashboard to visualize EC2 metrics, this is shown on following screenshot:



2.3. Implementation

On this section, it will be described a use case implantation to showcase and explain all the component involved and relevant code snippets will be shown. The use case selected was one of the most important in the system, which is “Run a machine learning workflow”. To help to understand the implementation details, we can split the system into two main components, backend, and frontend.

Let’s start from the user’s perspective, where the user interacts with the UI, events are raised, and requests are being sent to the backend for a later processing.

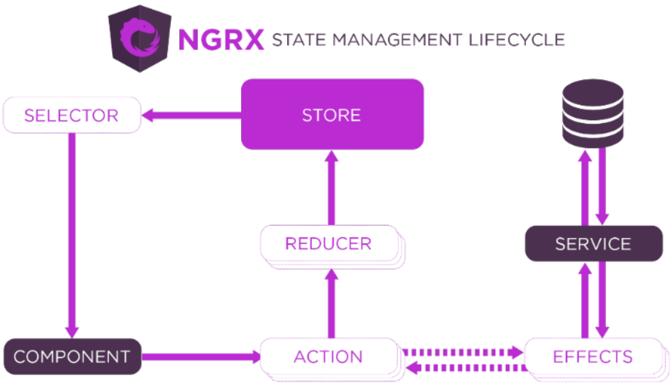
2.3.1. Frontend (Angular application)

The frontend component is a single page application developed on Angular 15, which run on Heroku Platform. The GitHub repository can be found on https://github.com/herreramaxi/MH-AIaaS-UI.

2.3.1.1. NGRX as state management for Angular

To explain how the events are being managed by the workflow designer component I must briefly introduce NGRX [11] state management for Angular, because this library is used on the component that I am about to explain. NGRX was introduced on the workflow-designer component to manage the complexity on the user interaction and the application internal state. This state management allows to clearly defines a state machine to handle the complexity of the interaction between user and application by defining the following elements:

- Actions: events that can be dispatched from components.
- Reducers: functions that can handle state changes.
- Effects: listeners that can react to specific actions and interact with services or other components.



Source: https://ngrx.io/guide/store

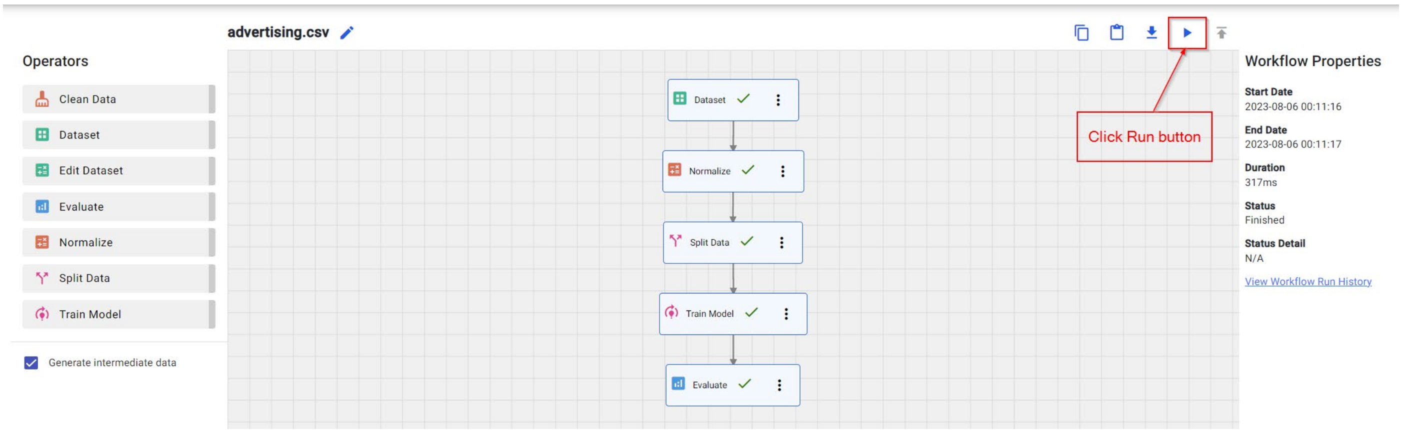
The use of NGRX brought several benefits to the application:

- Centralized state management to handle complex user interactions and centralize and protect the application’s state.
- Well defined state machine, with clear states, transition functions and side effects.
- Components decoupling by delegating into the state machine logic associated to events.

The state machine can be found on following folder: src/app/state-management

2.3.1.2. Run a workflow use case (Frontend)

On the workflow designer page, the user will start by clicking on run button that is shown below and the Angular application will react to the event and send the request to the backend.



Start Date
2023-08-06 00:11:16

End Date
2023-08-06 00:11:17

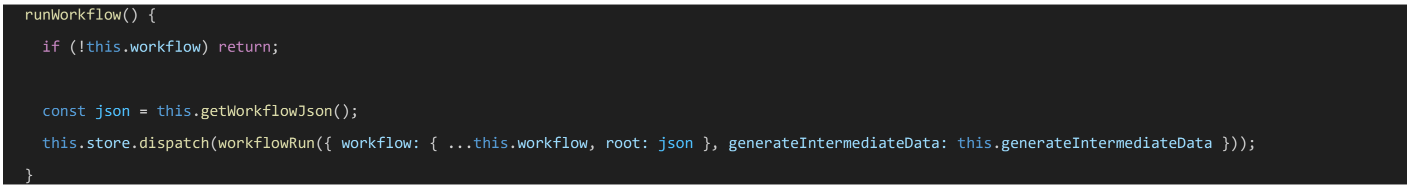
Duration
317ms

Status
Finished

Status Detail
N/A

2.3.1.2.1. Run workflow handler

The runWorkflow() function handles the click event on the ml-workflow-designer.component.ts#L133 and dispatch a workflowRun action that will be handled by the state management (NGRX).



```
runWorkflow() {
  if (!this.workflow) return;

  const json = this.getWorkflowJson();
  this.store.dispatch(workflowRun({ workflow: { ...this.workflow, root: json }, generateIntermediateData: this.generateIntermediateData }));
}
```

2.3.1.2.2. Change on state, workflow.reducer

A reducer will react to the workflowRun action and will change the internal state, in this case will change the status property (workflow.reducers.ts#L16) from the internal state:

```
    on(workflowRun, (state) => {
        return ({ ...state, status: "Running..." })
    }),
```

src/app/state-management/reducers/workflow.reducers.ts#L42

2.3.1.2.3. Workflow run effect

The effect or side effect for the workflowRun action will call service.run() and depending on the returned status code the state machine will transition to workflowRunSuccess or WorkflowEunFailed.

```
workflowRun$ = createEffect(() => this.actions$.pipe(
        ofType(workflowRun),
        concatMap((action) => this.service.run(action.workflow, action.generateIntermediateData)
            .pipe(
                map(response => {
                    return workflowRunSuccess(response)
                }),
                catchError((response: any) => {
                    console.log(`Error on action ${workflowRunType}`)
                    return of(workflowRunFailed({ error: response.error }))
                })
            ))
    ));
```

src/app/state-management/effects/workflow.effects.ts#L62

2.3.1.2.4. Workflow service, run() function

The workflow service once the run() function is called, it will send a post request to the backend as following:

```
  run(workflow: Workflow, generateIntermediateData: boolean = false) {
    return this.http.post<Workflow>('api/workflows/run', { ...workflow, generateIntermediateData:
generateIntermediateData });
  }
```

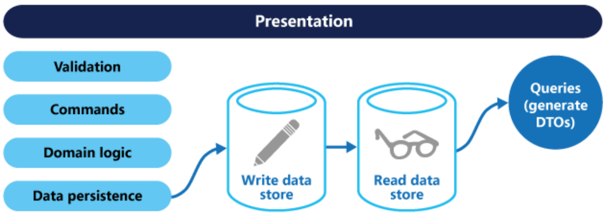
src/app/core/services/workflow.service.ts#L44

2.3.1.3. Backend (.NET Web API)

The backend service is a Web API developed with C# .NET 6 and runs on a Docker container that is deployed into a EC2 with a web server NGINX. The GitHub repository can be found on https://github.com/herreramaxi/MH-AIaaS.

Before starting the description of the use case "Run Workflow" I would briefly describe the architecture on the backend so that it can be easier to understand. The Web API, as mentioned on "Design & Architecture" section, is implemented following Clean Architecture.

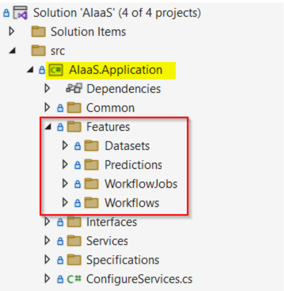
In addition, the design pattern CQRS (Command and query responsibility segregation) is being used on the application utilizing the MediatR implementation to segregate read and write operations to database. This brings to the application the ability to scale, maximize performance and allows to better evolve over the time. Below it is showing a high-level overview of the CQRS pattern.



CQRS diagram [12]

2.3.1.3.1. CQRS on the .NET solution

On the .NET solution, CQRS can be found on AlaaS.Application project under the folder "Features" as can be found below:



```
Solution 'AlaaS' (4 of 4 projects)
  Solution Items
  src
    AlaaS.Application
      Dependencies
      Common
      Features
        Datasets
        Predictions
        WorkflowJobs
        Workflows
      Interfaces
      Services
      Specifications
      ConfigureServices.cs
```

2.3.1.3.2. Run a workflow use case (Backend)

2.3.1.3.2.1. Workflow Controller

When the Angular application send a post request to "api/workflow/run" endpoint, the Web API will dispatch the request the a controller-action, that in this case is defined on AlaaS.WebAPI/Controllers/WorkflowsController.cs#L88 as it is shown on below code snippet:

```
[HttpPost("run")]
public async Task<ActionResult<WorkflowDto>> Run(WorkflowDto workflowDto)
{
    var command = new RunWorkflowCommand(workflowDto);
    var workflow = await _mediator.Send(command);

    return workflow.ToActionResult(this);
}
```

The important part from above code is when the command is sent through the mediator (_mediator), which means that the Mediator will dispatch the command to unique handler:

var workflow = await _mediator.Send(command);

2.3.1.3.2.2. Workflow Run handler

Once the RunWorkflowCommand is sent by the mediator, It must be received and processed by a unique command handler. In this case is AlaaS.Application/Features/Workflows/Commands/RunWorkflow/RunWorkflow.cs#L53, and the code associated is shown on below code snippet:

```
public async Task<Result<WorkflowDto>> Handle(RunWorkflowCommand request, CancellationToken
cancellationToken)
{
    try
    {
        var workflow = await _workflowRepository.FirstOrDefaultAsync(new
WorkflowByIdIncludeAllSpec(request.WorkflowDto.Id), cancellationToken);
        if (workflow is null)
        {
            return Result.NotFound("Workflow not found");
        }

        var context = new WorkflowContext()
        {
            MLContext = new MLContext(seed: 0),
            Workflow = workflow,
            RunWorkflow = true,
            GenerateIntermediateData = request.GenerateIntermediateData
        };

        _nodeProcessor.NodeStartProcessingEvent += (node, workflowRunHistoryId) =>
_nodeProcessor_NodeStartProcessingEvent(node, workflowRunHistoryId, cancellationToken);
        _nodeProcessor.NodeFinishProcessingEvent += (node, result) =>
_nodeProcessor_NodeFinishProcessingEvent(node, result, cancellationToken);

        var result = await _nodeProcessor.Run(request.WorkflowDto, context, cancellationToken);

        if (!result.IsSuccess)
        {
            return result;
        }

        workflow.UpdateData(result.Value.Root);
        await _workflowRepository.UpdateAsync(workflow, cancellationToken);

        var mapped = _mapper.Map<Workflow, WorkflowDto>(workflow);

        return Result.Success(mapped);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "Error when running workflow", ex.Message);

        return Result.Error($"Error when running workflow: {ex.Message}");
    }
    finally
    {
        _nodeProcessor.NodeStartProcessingEvent -= (node, workflowRunHistoryId) =>
_nodeProcessor_NodeStartProcessingEvent(node, workflowRunHistoryId, cancellationToken);
        _nodeProcessor.NodeFinishProcessingEvent -= (node, result) =>
_nodeProcessor_NodeFinishProcessingEvent(node, result, cancellationToken);
    }
}
```

The above code start validating if the workflow exists, if it does not then a not found is returned. Once the validation passed, the handler builds a context for the workflow operators' execution:

```
var context = new WorkflowContext()
{
    MLContext = new MLContext(seed: 0),
    Workflow = workflow,
    RunWorkflow = true,
    GenerateIntermediateData = request.GenerateIntermediateData
};
```

The handler also attaches some events (NodeStartProcessingEvent and NodeFinishProcessingEvent) that are events to be sent to UI via Web Socket to inform that a specific node has started or finished. That is done on below code:

```
_nodeProcessor.NodeStartProcessingEvent += (node, workflowRunHistoryId) => _nodeProcessor_NodeStartProcessingEvent(node, workflowRunHistoryId, cancellationToken);

_nodeProcessor.NodeFinishProcessingEvent += (node, result) => _nodeProcessor_NodeFinishProcessingEvent(node, result, cancellationToken);
```

After that, it just calls to Run() method on the _nodeProcessor, because that service has only the responsibility of running or executing operators (Single responsibility principle):

```
var result = await _nodeProcessor.Run(request.WorkflowDto, context, cancellationToken);
```

We can see on the former code snippet that after running the workflow, if everything is ok then we update database and we return a successful response.

2.3.1.3.2.3. Node processor service

This component will oversee the execution of the workflow nodes and emit events to inform the UI that the processing of a specific operator has been started or completed. That code can be found on AIaaS.Application/Services/NodeProcessorService.cs#L28 and the whole code snippet is shown below:

```
public async Task<Result<WorkflowDto>> Run(WorkflowDto workflowDto, WorkflowContext context, CancellationToken cancellationToken, bool
ignoreNodeErrors = false)
{
    var workflowGraphDto = workflowDto.GetDeserializedWorkflowGraph();
    if (workflowGraphDto is null)
    {
        return Result.Error("Not able to process workflow");
    }

    var nodes = workflowGraphDto.Root.ToList(doubleLinked: true, generateGuidIfNotExist: true);
    if (!nodes.Any())
    {
        return Result.Error("There are no operators in the workflow, please configure your workflow with operators");
    }

    foreach (WorkflowNodeDto node in nodes)
    {
        await OnNodeStartProcessing(node, _workflowRunHistoryContext.WorkflowRunHistory?.Id);

        var result = await ProcessNode(node, context, cancellationToken);

        await OnNodeFinishProcessing(node, _workflowRunHistoryContext.WorkflowRunHistory?.Id, result);

        if (!ignoreNodeErrors && !result.IsSuccess)
        {
            return result;
        }
    }

    workflowDto.UpdateSerializedRootFromGraph(workflowGraphDto);

    if (string.IsNullOrEmpty(workflowDto.Root))
    {
        return Result.Error("Serialized workflow is null or empty");
    }

    return Result.Success(workflowDto);
}
```

The important part of above code is on the foreach loop, when the ProcessNode() is called, also is possible to see when the OnNodeStartProcessing and OnNodeFinishProcessing events are being raised.

2.3.1.3.2.4. Processing a node

On the same class it can be found the method ProcessNode() that has all the logic to execute any operator defined into the system, below there is the code snippet:

```
private async Task<Result> ProcessNode(WorkflowNodeDto node, WorkflowContext context, CancellationToken
cancellationToken)
{
    try
    {
        var workflowOperator = _workflowOperators.FirstOrDefault(x => x.Type.Equals(node.Type,
StringComparison.InvariantCultureIgnoreCase));
        if (workflowOperator is null)
        {
            return Result.Error($"Workflow operator not found for type {node.Type}");
        }

        workflowOperator.Preprocessing(context, node);
        await workflowOperator.Hydrate(context, node);
        workflowOperator.PropagateDatasetColumns(context, node);
        var validationResult = workflowOperator.Validate(context, node);

        if (!validationResult.IsSuccess || !context.RunWorkflow)
        {
            return validationResult;
        }

        var runResult = await workflowOperator.Run(context, node, cancellationToken);
        if (!runResult.IsSuccess)
        {
            return runResult;
        }

        if (context.GenerateIntermediateData)
        {
            var generateOutputresult = await workflowOperator.GenerateOuput(context, node,
cancellationToken);
            return generateOutputresult;
        }

        return Result.Success();
    }
    catch (Exception ex)
    {
        return Result.Error($"Error when executing operator {node.Type}: {ex.Message}");
    }
}
```

Logically what I am doing is first try to retrieve the operator object, which is a class that implements IWorkflowOperator. All the workflow operators can be found on the following path: AIaaS.Application/Features/Workflows/Commands/Common/Operators.

Once the class instance is found, it proceeds to execute a series of steps to execute correctly the operator as it is shown below:

```
        workflowOperator.Preprocessing(context, node);
        await workflowOperator.Hydrate(context, node);
       workflowOperator.PropagateDatasetColumns(context, node);
       var validationResult = workflowOperator.Validate(context, node);
```

If everything goes well, then the operator is executed:

```
 var runResult = await workflowOperator.Run(context, node, cancellationToken);
```

And if the result is success, the operator is instructed to generate intermediate data:

```
var generateOutputresult = await workflowOperator.GenerateOuput(context, node, cancellationToken);
```

Intermediate data is stored on AWS S3, and it contains a binary file that is the output of the execution of a workflow operator, that can be used for the user to visualize a preview of the transformation applied for a given operator.

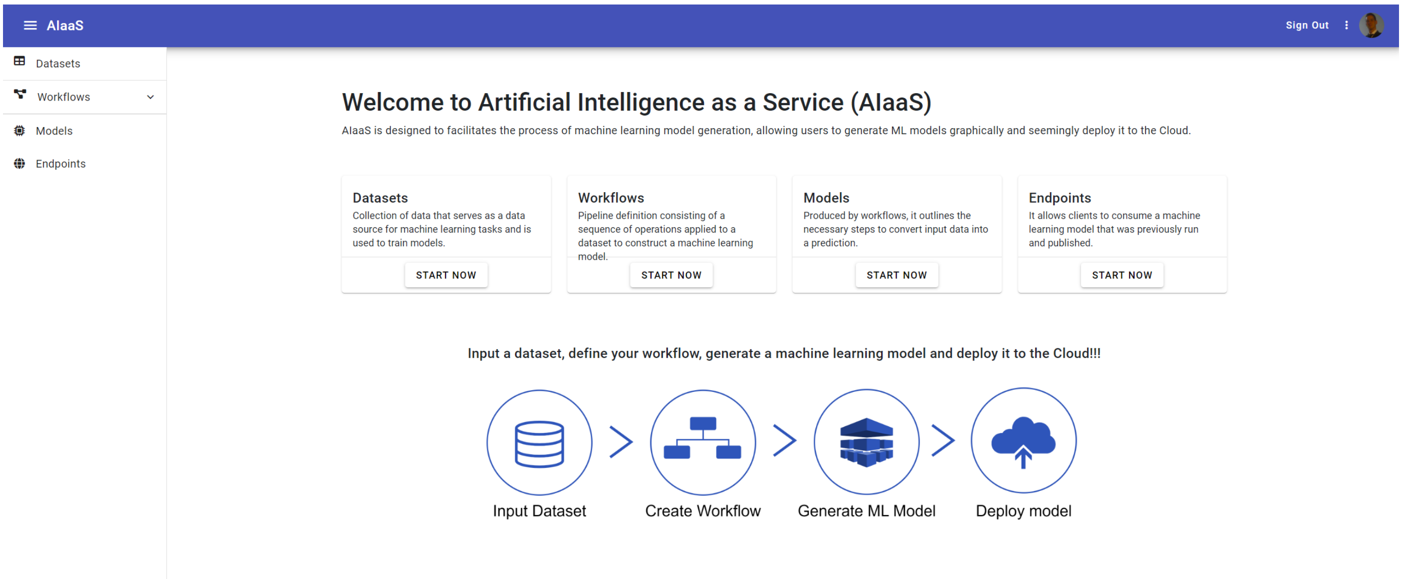
2.4. Graphical User Interface (GUI)

2.4.1. Home page

It is the landing page of the system and one of the most important screens because provides a brief explanation of the main concepts such as Datasets, Workflows, Models and Endpoints and a quick access to each of them. Additionally, it provides a graphical representation of the steps required to deploy a machine learning model, which is the ultimate objective of this solution.

The home page provides one navbar on the top, where there is a button for signing-in and signing-out, and access to the user's profile by clicking the three vertical dots on the right corner. On the left-hand side there is a collapse button to expand or collapse the left-hand side menu.

The home page also provides on the left-hand side a collapsible menu to help the user to navigate through the website and provide more space if needed.



AIaaS

Sign Out

Datasets

Workflows

Models

Endpoints

Welcome to Artificial Intelligence as a Service (AIaaS)

AIaaS is designed to facilitates the process of machine learning model generation, allowing users to generate ML models graphically and seemingly deploy it to the Cloud.

Datasets
Collection of data that serves as a data source for machine learning tasks and is used to train models.
START NOW

Workflows
Pipeline definition consisting of a sequence of operations applied to a dataset to construct a machine learning model.
START NOW

Models
Produced by workflows, it outlines the necessary steps to convert input data into a prediction.
START NOW

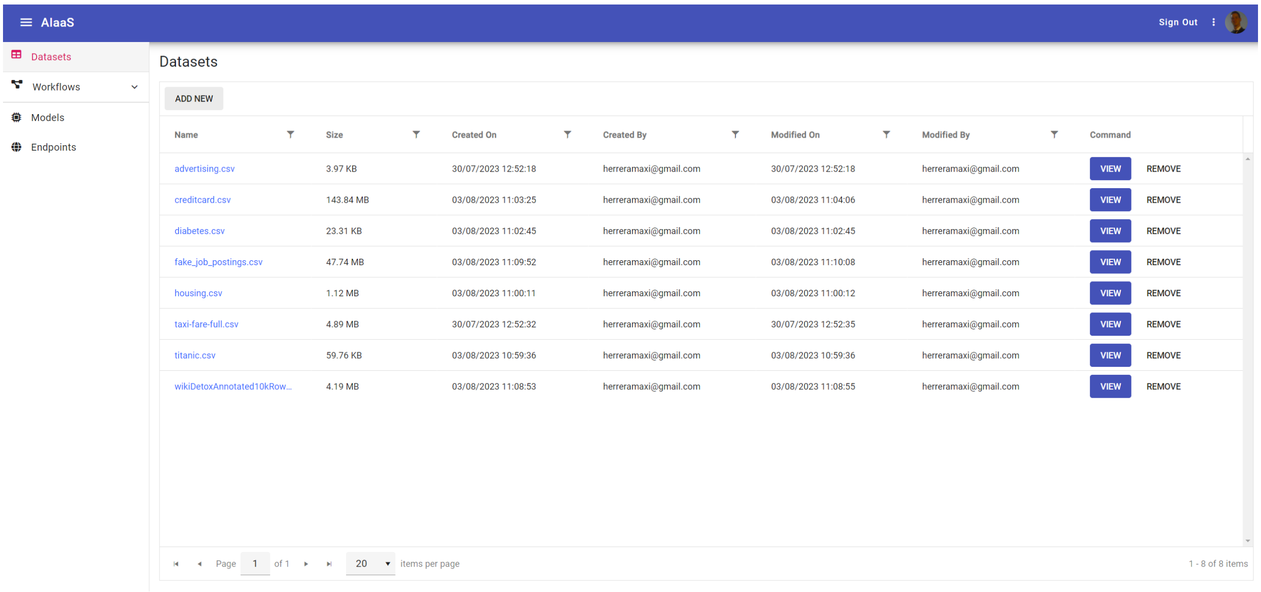
Endpoints
It allows clients to consume a machine learning model that was previously run and published.
START NOW

Input a dataset, define your workflow, generate a machine learning model and deploy it to the Cloud!!!

2.4.2. Datasets

A dataset is an input file that is required to generate machine learning models by running a workflow. The system accepts two formats as dataset files, one is CSV (comma separated values) and the other is TSV (Tab separated values). Those are the most standards file formats that systems usually accept and very well known in the world of Machine Learning.

The datasets screen provides a list of existing datasets on the system, showing the most relevant information such as name, size, and audit fields. Moreover, it provides the remove and view commands which allows the user to explore in more detail the dataset or remove a dataset from the system.



Name	Size	Created On	Created By	Modified On	Modified By	Command
advertising.csv	3.97 KB	30/07/2023 12:52:18	herreramaxi@gmail.com	30/07/2023 12:52:18	herreramaxi@gmail.com	VIEW REMOVE
creditcard.csv	143.84 MB	03/08/2023 11:03:25	herreramaxi@gmail.com	03/08/2023 11:04:06	herreramaxi@gmail.com	VIEW REMOVE
diabetes.csv	23.31 KB	03/08/2023 11:02:45	herreramaxi@gmail.com	03/08/2023 11:02:45	herreramaxi@gmail.com	VIEW REMOVE
fake_job_postings.csv	47.74 MB	03/08/2023 11:09:52	herreramaxi@gmail.com	03/08/2023 11:10:08	herreramaxi@gmail.com	VIEW REMOVE
housing.csv	1.12 MB	03/08/2023 11:00:11	herreramaxi@gmail.com	03/08/2023 11:00:12	herreramaxi@gmail.com	VIEW REMOVE
taxi-fare-full.csv	4.89 MB	30/07/2023 12:52:32	herreramaxi@gmail.com	30/07/2023 12:52:35	herreramaxi@gmail.com	VIEW REMOVE
titanic.csv	59.76 KB	03/08/2023 10:59:36	herreramaxi@gmail.com	03/08/2023 10:59:36	herreramaxi@gmail.com	VIEW REMOVE
wikiDetoxAnnotated10kRow...	4.19 MB	03/08/2023 11:08:53	herreramaxi@gmail.com	03/08/2023 11:08:55	herreramaxi@gmail.com	VIEW REMOVE

Page 1 of 1 20 items per page 1 - 8 of 8 items

2.4.2.1. Datasets – Add New

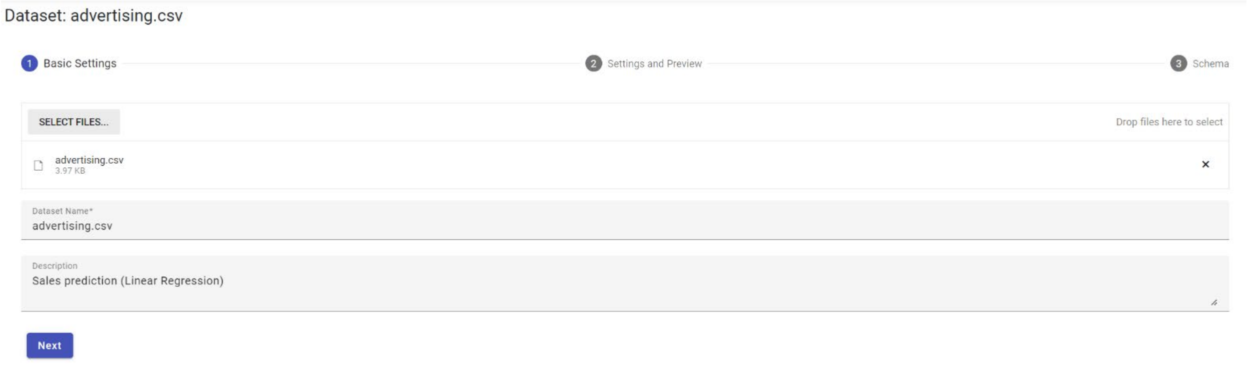
This screen enables an administrator to create a dataset in the system, which can be any file in the format of CSV or TSV. This solution does not require to have a C# class with the schema definition (property names and types) of a given Dataset, because they are created on runtime. This means that the system can create dynamically new datasets and provides the user an inferred data type which can be changed if needed.

The Add New page is implemented as a stepper with the three following steps:

2.4.2.1.1. Datasets – Add New - Basic Settings

It allows user to select a file from its computer and enter a dataset name with an optional description.

2.4.2.1.2. Datasets – Add New – Settings and Preview



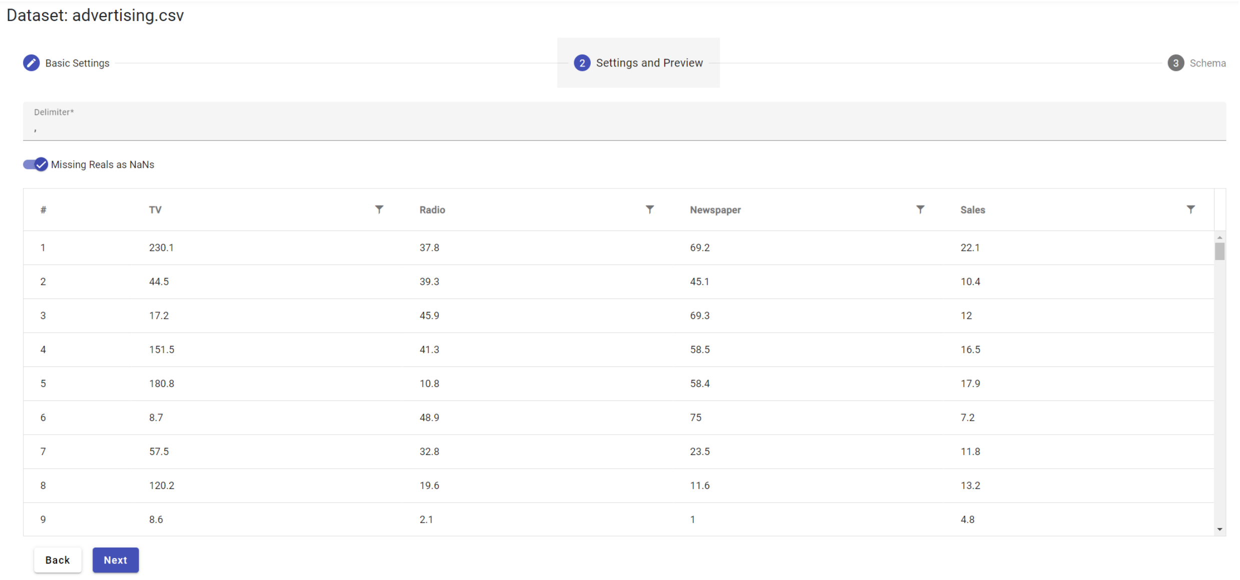
Dataset: advertising.csv

advertising.csv
3.97 KB

Dataset Name*
advertising.csv

Description
Sales prediction (Linear Regression)

This screen enables the user to configure file parse settings and it provides an initial preview. The file parse settings include the file delimiter which is necessary to parse the file correctly. The second setting is “Missing Reals as NaN”, that is used to set Not a Number (NaN) to a cell value when there is no value available for the given cell. This is to avoid having “zeros” to represent a missing number.



Dataset: advertising.csv

Delimiter*
,

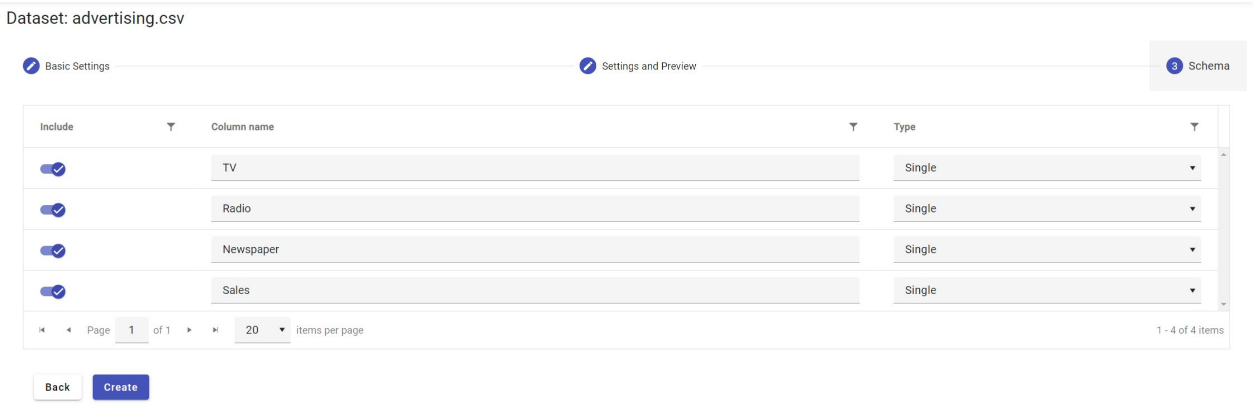
Missing Reals as NaNs ✓

#	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	12
4	151.5	41.3	58.5	16.5
5	180.8	10.8	58.4	17.9
6	8.7	48.9	75	7.2
7	57.5	32.8	23.5	11.8
8	120.2	19.6	11.6	13.2
9	8.6	2.1	1	4.8

2.4.2.1.3. Datasets – Add New – Schema

The last step to create a dataset is "Schema" and provides an inferred schema definition of the dataset that includes column name and data types. To infer the dataset schema (column data types), it is used Microsoft.ML.AutoML also known as Automated Machine Learning for .NET.

In addition, this screen allows the user to configure which columns to include on the dataset, change columns data type and enables the user to rename any column if required.



Dataset: advertising.csv

Include	Column name	Type
✓	TV	Single
✓	Radio	Single
✓	Newspaper	Single
✓	Sales	Single

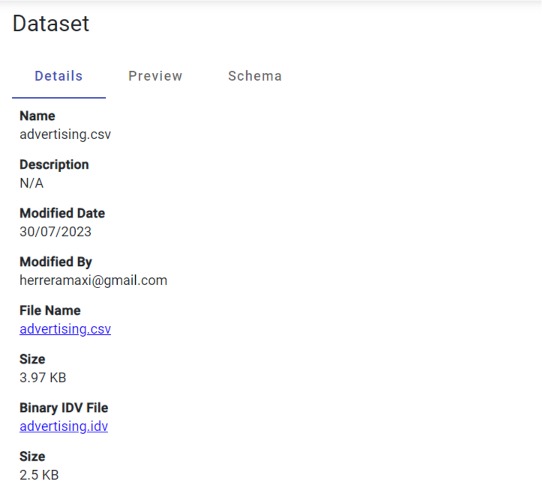
Page 1 of 1 20 items per page 1 - 4 of 4 items

2.4.2.2. Datasets – View

The View screen displays all the relevant fields of a dataset, including a file preview, and the schema definition. It is split into three tabs to provide a better user experience and group those fields that are related to each other.

2.4.2.2.1. Datasets – View – Details tab

It shows the information related to the dataset file itself, such as name, description, size, and audit information. It also provides information about the binary data file (.idv) that is generated from the original file. This binary file contains not only the original data but also the schema definition enabling the system to quickly load the dataset when needed without incurring in file processing or parsing. An important feature on this screen is the download of the two files, the original file, and the binary data file. The link provided allows the user to download the files from AWS S3, which is much faster than downloading a file from the website.



Dataset

Name
advertising.csv

Description
N/A

Modified Date
30/07/2023

Modified By
herreramaxi@gmail.com

File Name
advertising.csv

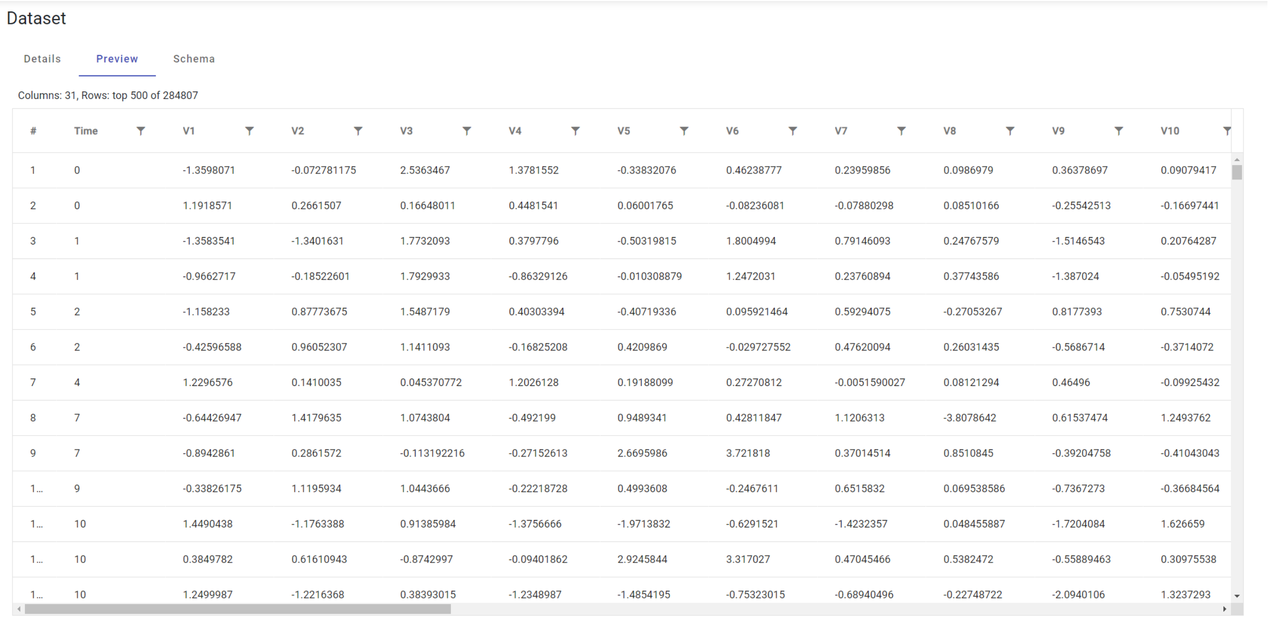
Size
3.97 KB

Binary IDV File
advertising.idv

Size
2.5 KB

2.4.2.2.2. Datasets – View – Preview tab

This screen provides a preview of the top 500 records of the dataset, and the quantity of the dataset's columns. The limit on 500 rows is to avoid loading big quantity of data on memory and protect in that way from consuming all the available and scarce memory on a EC2 t2.micro.



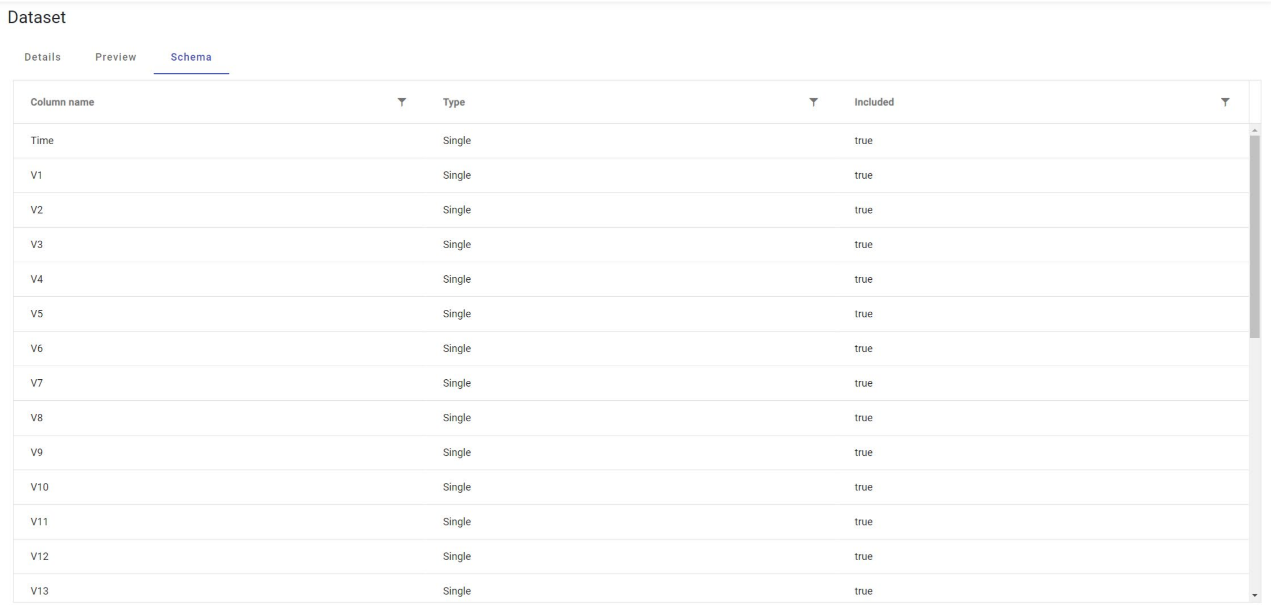
Dataset

Columns: 31, Rows: top 500 of 284807

#	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	0	-1.3598071	-0.072781175	2.5363467	1.3781552	-0.33832076	0.46238777	0.23959856	0.0986979	0.36378697	0.09079417
2	0	1.1918571	0.2661507	0.16648011	0.4481541	0.06001765	-0.08236081	-0.07880298	0.08510166	-0.25542513	-0.16697441
3	1	-1.3583541	-1.3401631	1.7732093	0.3797796	-0.50319815	1.8004994	0.79146093	0.24767579	-1.5146543	0.20764287
4	1	-0.9662717	-0.18522601	1.7929933	-0.86329126	-0.010308879	1.2472031	0.23760894	0.37743586	-1.387024	-0.05495192
5	2	-1.158233	0.87773675	1.5487179	0.40303394	-0.40719336	0.095921464	0.59294075	-0.27053267	0.8177393	0.7530744
6	2	-0.42596588	0.96052307	1.1411093	-0.16825208	0.4209869	-0.029727552	0.47620094	0.26031435	-0.5686714	-0.3714072
7	4	1.2296576	0.1410035	0.045370772	1.2026128	0.19188099	0.27270812	-0.0051590027	0.08121294	0.46496	-0.09925432
8	7	-0.64426947	1.4179635	1.0743804	-0.492199	0.9489341	0.42811847	1.1206313	-3.8078642	0.61537474	1.2493762
9	7	-0.8942861	0.2861572	-0.113192216	-0.27152613	2.6695986	3.721818	0.37014514	0.8510845	-0.39204758	-0.41043043
1...	9	-0.33826175	1.1195934	1.0443666	-0.22218728	0.4993608	-0.2467611	0.6515832	0.069538586	-0.7367273	-0.36684564
1...	10	1.4490438	-1.1763388	0.91385984	-1.3756666	-1.9713832	-0.6291521	-1.4232357	0.048455887	-1.7204084	1.626659
1...	10	0.3849782	0.61610943	-0.8742997	-0.09401862	2.9245844	3.317027	0.47045466	0.5382472	-0.55889463	0.30975538
1...	10	1.2499987	-1.2216368	0.38393015	-1.2348987	-1.4854195	-0.75323015	-0.68940496	-0.22748722	-2.0940106	1.3237293

2.4.2.2.3. Datasets – View – Schema tab

This screen provides information about the schema definition for a given dataset, such as column name and data type.

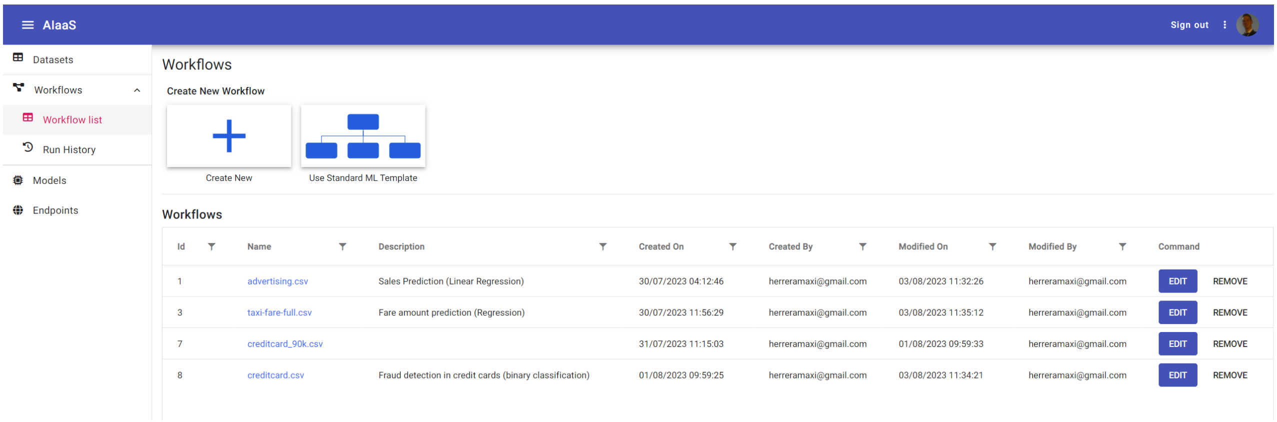


Column name	Type	Included
Time	Single	true
V1	Single	true
V2	Single	true
V3	Single	true
V4	Single	true
V5	Single	true
V6	Single	true
V7	Single	true
V8	Single	true
V9	Single	true
V10	Single	true
V11	Single	true
V12	Single	true
V13	Single	true

2.4.3. Workflows

2.4.3.1. Workflow list

The workflows list page shows the existing workflows in the system and allows the user to quickly visualize relevant fields of a workflow such as id, name, description, and audit fields. This page also allows an administrator to create, edit or remove a workflow. On the top of the screen can be found two buttons for workflow creation that intend to improve the user experience. The first button on the left is called "Create New" and allows a user to create an empty workflow. The second on the right is called "Use Standard ML Template" and allows the user to create a workflow with a basic standard configuration.



Id	Name	Description	Created On	Created By	Modified On	Modified By	Command
1	advertising.csv	Sales Prediction (Linear Regression)	30/07/2023 04:12:46	herreramaxi@gmail.com	03/08/2023 11:32:26	herreramaxi@gmail.com	EDIT REMOVE
3	taxi-fare-full.csv	Fare amount prediction (Regression)	30/07/2023 11:56:29	herreramaxi@gmail.com	03/08/2023 11:35:12	herreramaxi@gmail.com	EDIT REMOVE
7	creditcard_90k.csv		31/07/2023 11:15:03	herreramaxi@gmail.com	01/08/2023 09:59:33	herreramaxi@gmail.com	EDIT REMOVE
8	creditcard.csv	Fraud detection in credit cards (binary classification)	01/08/2023 09:59:25	herreramaxi@gmail.com	03/08/2023 11:34:21	herreramaxi@gmail.com	EDIT REMOVE

2.4.3.2. Workflow Designer

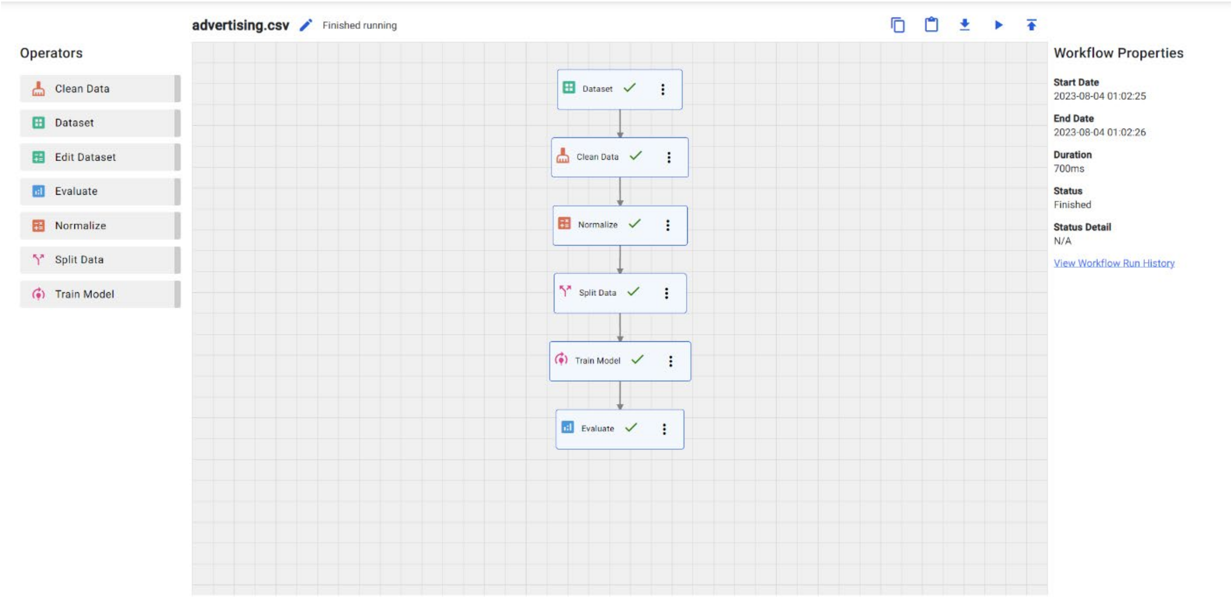
Workflow designer is one of the most important pages on the system and allows the user to define and create machine learning workflows. A ML workflow represents a series of steps, each of them can apply a transformation to the dataset until a model is generated. It is worth to mention that the execution of workflows generates not only a ML model, but also many other artefacts such as:

- Intermediate data: "Dataview" files in the form of binary data (.idv), which is stored on AWS S3. This intermediate data is the output of each operator and allows the user to visualize a preview for the transformation applied.
- ML Model: This is the trained model generated, and it is used for the system to provide prediction services via endpoint.

- Model metrics: Metrics related to the model generated such as R Squared (R^2), mean squared error (MSE), root mean squared error (RMSE) and so on so forth. It is important to mention that the model metrics are available from the "Evaluate" operator (clicking on the three dots) and from Models page (View model). In addition, there are different metrics that can be generated for a given workflow. The metrics generated will depend on the ML task selected, such as linear regression or binary classification.
- Workflow metrics and execution details: This information contains a description of the execution of a workflow as a job, which includes the start date, end date, status, and other information. These metrics are stored into database and include metrics for the execution of each operator.

Below it is shown how the workflow designer looks like and the second screenshot highlight the important sections within the screen.

Workflow designer



Start Date
2023-08-04 01:02:25

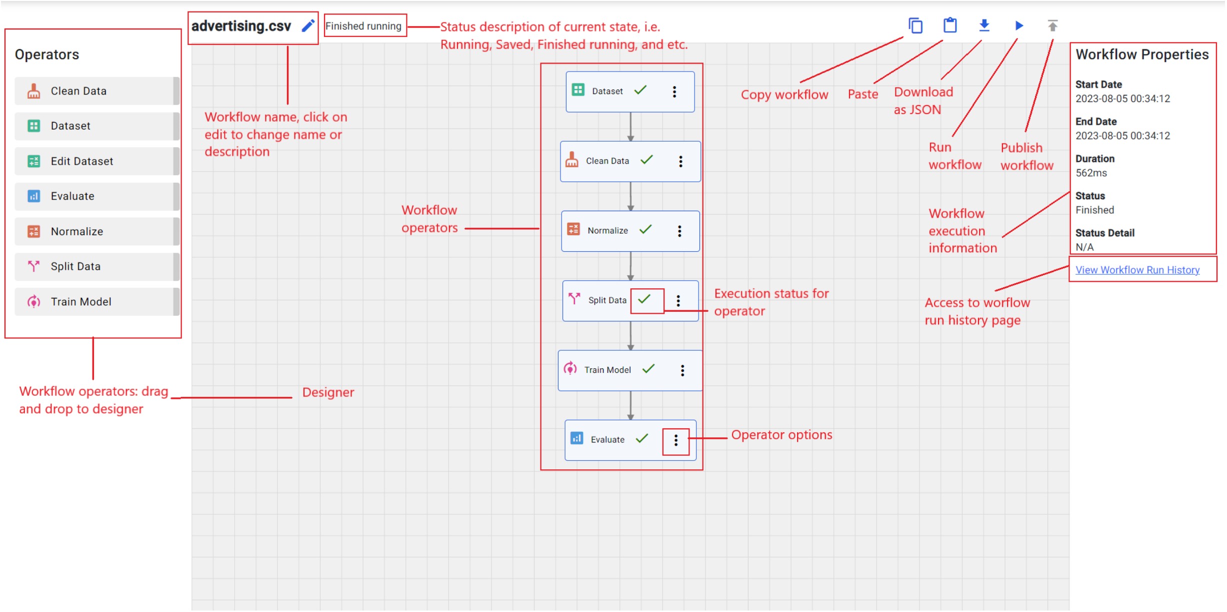
End Date
2023-08-04 01:02:26

Duration
700ms

Status
Finished

Status Detail
N/A

Workflow Designer highlighted



Start Date
2023-08-05 00:34:12

End Date
2023-08-05 00:34:12

Duration
562ms

Status
Finished

Status Detail
N/A

Status description of current state, i.e. Running, Saved, Finished running, and etc.

Workflow name, click on edit to change name or description

Copy workflow

Paste

Download as JSON

Run workflow

Publish workflow

Workflow operators

Workflow execution information

Execution status for operator

Access to worflow run history page

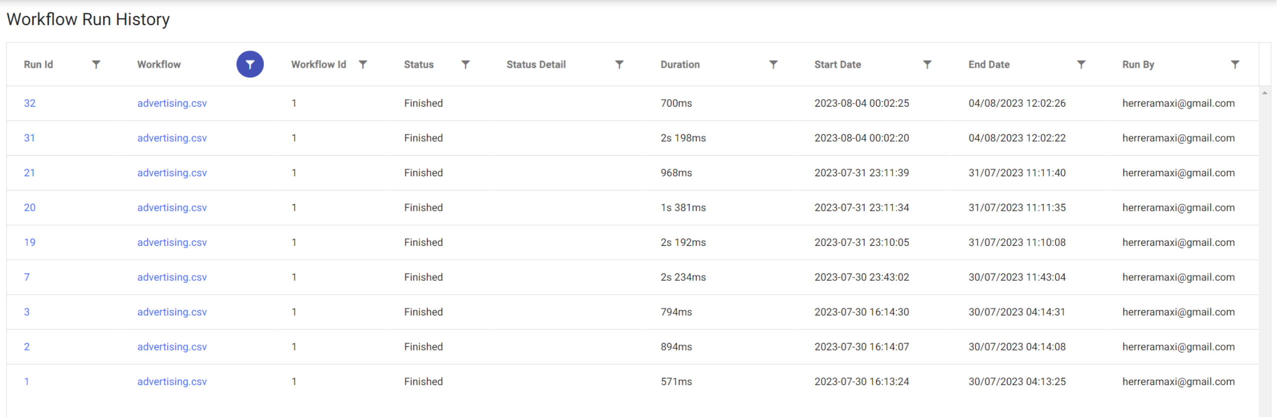
Workflow operators: drag and drop to designer

Designer

Operator options

2.4.3.3. Workflow Run History

The “Workflow Run History” screen provides information about a workflow execution, including the status, status detail, duration, and audit information. It also provides a link to the associated workflow and to “Workflow Run History Details” that provides a more detailed information of the workflow execution but at operator level.

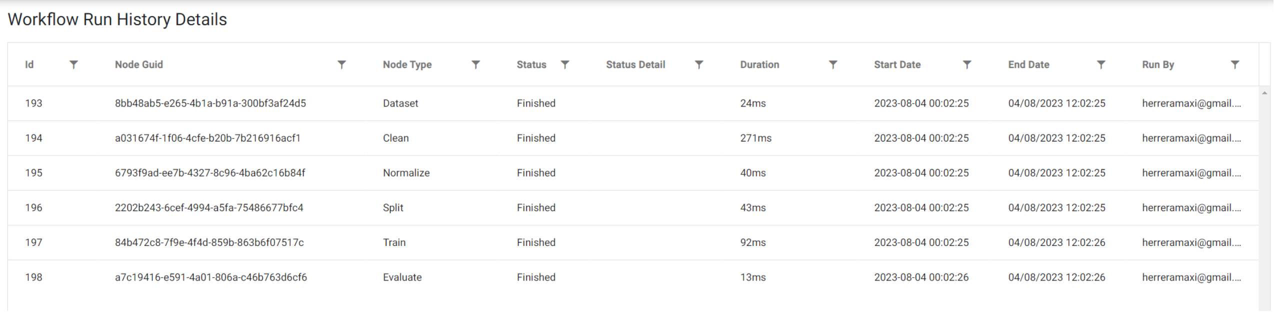


Workflow Run History

Run Id	Workflow	Workflow Id	Status	Status Detail	Duration	Start Date	End Date	Run By
32	advertising.csv	1	Finished		700ms	2023-08-04 00:02:25	04/08/2023 12:02:26	herreramaxi@gmail.com
31	advertising.csv	1	Finished		2s 198ms	2023-08-04 00:02:20	04/08/2023 12:02:22	herreramaxi@gmail.com
21	advertising.csv	1	Finished		968ms	2023-07-31 23:11:39	31/07/2023 11:11:40	herreramaxi@gmail.com
20	advertising.csv	1	Finished		1s 381ms	2023-07-31 23:11:34	31/07/2023 11:11:35	herreramaxi@gmail.com
19	advertising.csv	1	Finished		2s 192ms	2023-07-31 23:10:05	31/07/2023 11:10:08	herreramaxi@gmail.com
7	advertising.csv	1	Finished		2s 234ms	2023-07-30 23:43:02	30/07/2023 11:43:04	herreramaxi@gmail.com
3	advertising.csv	1	Finished		794ms	2023-07-30 16:14:30	30/07/2023 04:14:31	herreramaxi@gmail.com
2	advertising.csv	1	Finished		894ms	2023-07-30 16:14:07	30/07/2023 04:14:08	herreramaxi@gmail.com
1	advertising.csv	1	Finished		571ms	2023-07-30 16:13:24	30/07/2023 04:13:25	herreramaxi@gmail.com

2.4.3.4. Workflow Run History Details

Detailed information of the workflow execution at operator level, so this page can show for example how long took the execution of a particular workflow operator as well as the status. Another important information available on this screen is the GUID for each node type, which allows to track on AWS S3 the artefacts for a given operator.

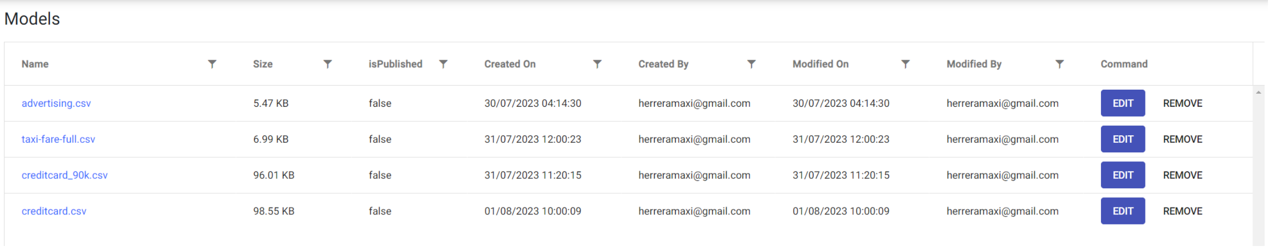


Workflow Run History Details

Id	Node Guid	Node Type	Status	Status Detail	Duration	Start Date	End Date	Run By
193	8bb48ab5-e265-4b1a-b91a-300bf3af24d5	Dataset	Finished		24ms	2023-08-04 00:02:25	04/08/2023 12:02:25	herreramaxi@gmail....
194	a031674f-1f06-4cfe-b20b-7b216916acf1	Clean	Finished		271ms	2023-08-04 00:02:25	04/08/2023 12:02:25	herreramaxi@gmail....
195	6793f9ad-ee7b-4327-8c96-4ba62c16b84f	Normalize	Finished		40ms	2023-08-04 00:02:25	04/08/2023 12:02:25	herreramaxi@gmail....
196	2202b243-6cef-4994-a5fa-75486677bfc4	Split	Finished		43ms	2023-08-04 00:02:25	04/08/2023 12:02:25	herreramaxi@gmail....
197	84b472c8-7f9e-4f4d-859b-863b6f07517c	Train	Finished		92ms	2023-08-04 00:02:25	04/08/2023 12:02:26	herreramaxi@gmail....
198	a7c19416-e591-4a01-806a-c46b763d6cf6	Evaluate	Finished		13ms	2023-08-04 00:02:26	04/08/2023 12:02:26	herreramaxi@gmail....

2.4.4. Models

The models screen shows a list of generated machine learning models that are stored in the system. It presents relevant information about the model such as name, size, if is published, and audit information. It also provides the remove and edict actions.

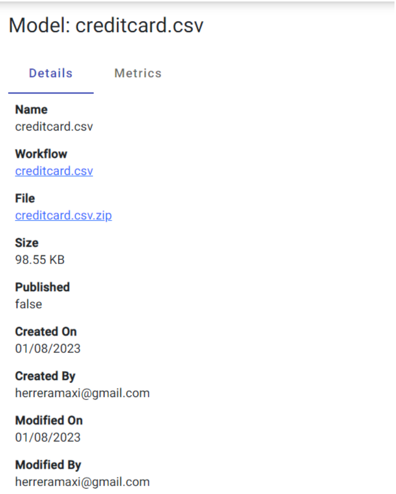


Models

Name	Size	isPublished	Created On	Created By	Modified On	Modified By	Command
advertising.csv	5.47 KB	false	30/07/2023 04:14:30	herreramaxi@gmail.com	30/07/2023 04:14:30	herreramaxi@gmail.com	EDIT REMOVE
taxi-fare-full.csv	6.99 KB	false	31/07/2023 12:00:23	herreramaxi@gmail.com	31/07/2023 12:00:23	herreramaxi@gmail.com	EDIT REMOVE
creditcard_90k.csv	96.01 KB	false	31/07/2023 11:20:15	herreramaxi@gmail.com	31/07/2023 11:20:15	herreramaxi@gmail.com	EDIT REMOVE
creditcard.csv	98.55 KB	false	01/08/2023 10:00:09	herreramaxi@gmail.com	01/08/2023 10:00:09	herreramaxi@gmail.com	EDIT REMOVE

2.4.4.1. Model Details

This screen is composed by two tabs, details, and metrics. The Details tab display all the available fields for a given model, such as name, workflow, file, size, and other valuable information. It also provides the option to download the model in case the user needed it. An important note is that the model is stored into database to provides an easy and fast access for the system.



Model: creditcard.csv

Name
creditcard.csv

Workflow
creditcard.csv

File
creditcard.csv.zip

Size
98.55 KB

Published
false

Created On
01/08/2023

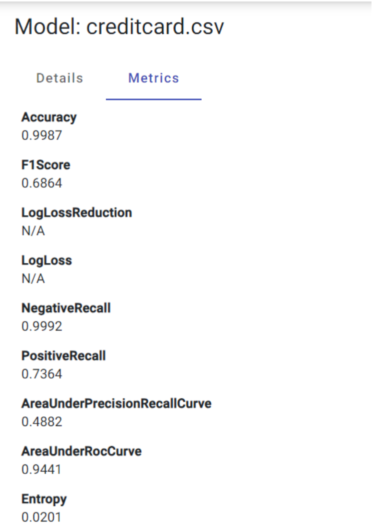
Created By
herreramaxi@gmail.com

Modified On
01/08/2023

Modified By
herreramaxi@gmail.com

2.4.4.2. Model Metrics

Model metrics screen allows the user to have a minimum evaluation tool to evaluate a model. The metrics shown on the screen depend on the machine learning task selected, that can be linear regression or binary classification.



Model: creditcard.csv

Accuracy
0.9987

F1Score
0.6864

LogLossReduction
N/A

LogLoss
N/A

NegativeRecall
0.9992

PositiveRecall
0.7364

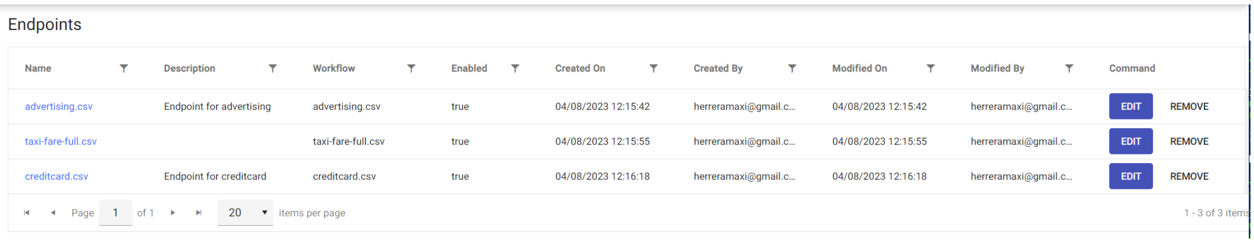
AreaUnderPrecisionRecallCurve
0.4882

AreaUnderRocCurve
0.9441

Entropy
0.0201

2.4.5. Endpoints

The endpoints page shows a list of available endpoints for those models that are generated and published. For each endpoint it is shown relevant fields such as name, description, workflow, if the endpoint is enabled and audit fields. The list offers the edit and remove command so the user can perform further actions on the endpoint entity.



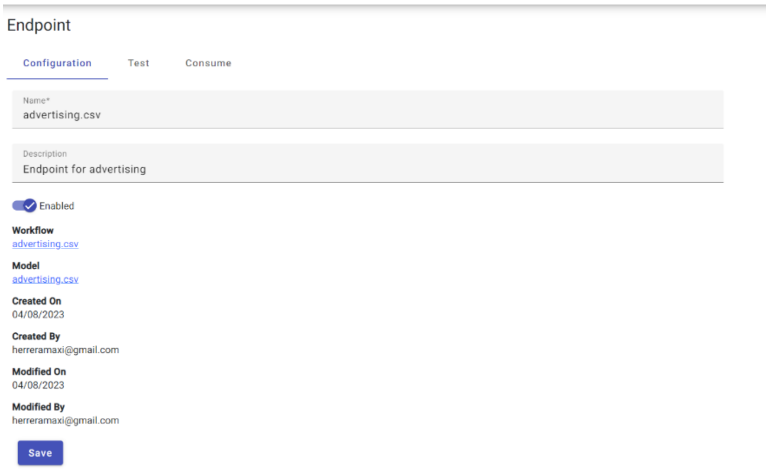
Endpoints

Name	Description	Workflow	Enabled	Created On	Created By	Modified On	Modified By	Command
advertising.csv	Endpoint for advertising	advertising.csv	true	04/08/2023 12:15:42	herreramaxi@gmail.c...	04/08/2023 12:15:42	herreramaxi@gmail.c...	EDIT REMOVE
taxi-fare-full.csv		taxi-fare-full.csv	true	04/08/2023 12:15:55	herreramaxi@gmail.c...	04/08/2023 12:15:55	herreramaxi@gmail.c...	EDIT REMOVE
creditcard.csv	Endpoint for creditcard	creditcard.csv	true	04/08/2023 12:16:18	herreramaxi@gmail.c...	04/08/2023 12:16:18	herreramaxi@gmail.c...	EDIT REMOVE

2.4.5.1. Endpoint Edit

This page is composed by three tabs, configuration, test and consume. The first tab configuration is shown when the user clicks on edit dataset and allows the user to visualize all the relevant fields associated to an endpoint.

The screen allows the user to edit the name and description of an endpoint and enable and disable the endpoint if required. In addition, audit fields are also present on this screen to let the administrator know who created or modified the endpoint and when.



Name*
advertising.csv

Description
Endpoint for advertising

Enabled

Workflow
advertising.csv

Model
advertising.csv

Created On
04/08/2023

Created By
herreramaxi@gmail.com

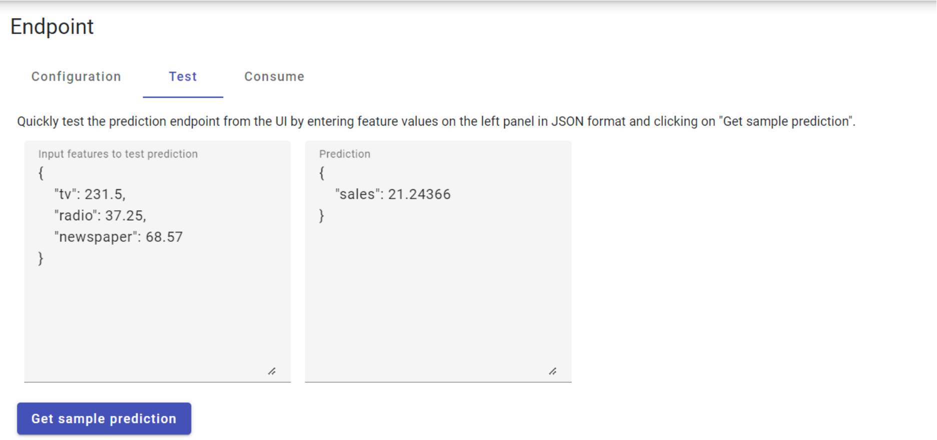
Modified On
04/08/2023

Modified By
herreramaxi@gmail.com

2.4.5.2. Endpoint Test

The endpoint test screen offers a quick way to test a prediction endpoint before making the endpoint available for consumption. This is a critical feature and tries to help the user to validate the model.

We can see on below screenshot two panels, the left-hand side panel enables the user to input features with values. On the right-hand side panel, a prediction would be shown when the user clicks on “Get sample prediction” button.



Quickly test the prediction endpoint from the UI by entering feature values on the left panel in JSON format and clicking on "Get sample prediction".

Input features to test prediction

```
{
   "tv": 231.5,
   "radio": 37.25,
   "newspaper": 68.57
}
```

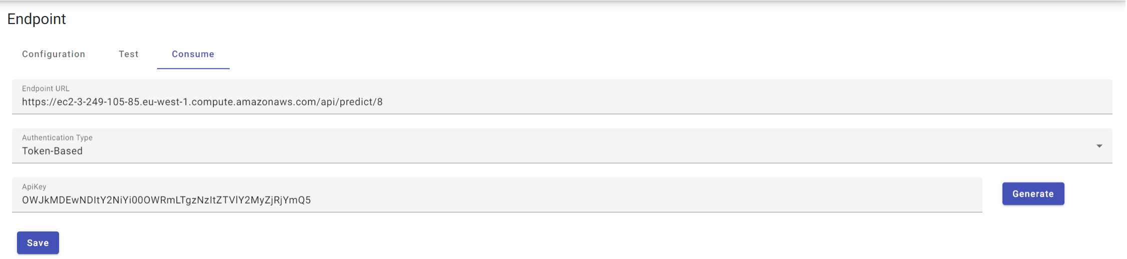
Prediction

```
{
   "sales": 21.24366
}
```

2.4.5.3. Endpoint Consume

This screen provides the URL of the predict endpoint so users can copy that URL and use it on their systems or test it using Postman or similar. Using this screen, the user is enabled to configure the authentication for the predict endpoint, using three available authentication types or methods:

- Anonymous: No authentication is required, so any consumer can make use of the predict endpoint.
- Token-Based: An API key would be generated by the system, when clicking on “Generate” button, and would be passes on the authorization header as a Bearer token.
- JWT from Auth0: It uses a JWT token generated by Auth0 that must be passed as Bearer token. The JWT token must include the claim “AIaaS-consumer” to successfully utilize the endpoint.



Endpoint

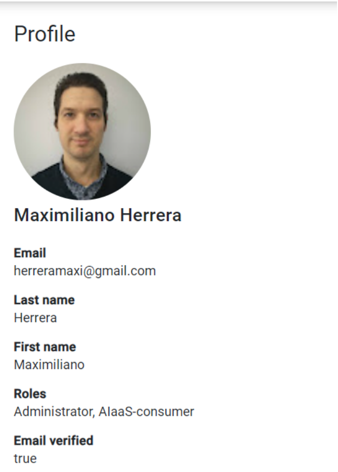
Endpoint URL
https://ec2-3-249-105-85.eu-west-1.compute.amazonaws.com/api/predict/8

Authentication Type
Token-Based

ApiKey
OWJkMDEwNDItY2NiYi00OWRmLTgzNzItZTVlY2MyZjRjYmQ5

2.4.6. Profile

Simple screen to provide basic information about the logged-in user such as full name, email, assigned roles from the JWT token, and if the email was verified.



Profile

Maximiliano Herrera

Email
herreramaxi@gmail.com

Last name
Herrera

First name
Maximiliano

Roles
Administrator, AIaaS-consumer

Email verified
true

2.5. Testing

2.5.1. Introduction

Given the technological complexity of this solutions which includes ML.NET (machine learning for .NET), Angular, SQL Server, and AWS for file storage, I have decided to go for a manual testing strategy rather than automated. To ensure a proper testing I have created a comprehensive test plan that covers different aspects of the solution, covering end-to-end functionality and security concerns.

2.5.2. Testing Scope

This test plan is limited to the validation and verification of the Artificial Intelligence as a Service (AIaaS) system and will be carried out following a manual strategy.

2.5.3. Test Objectives

The objectives of this test plan are to ensure that the system can perform all his functionality defined on the requirements section of this document. It will cover not only the functional requirements but also the non-functional including requirements related to security and performance.

2.5.4. Test Strategy

The test strategy chosen for this test plan defines the test levels of testing to be conducted following a manual approach.

2.5.4.1. Test Levels

- Functional Testing
- Regression Testing
- Non-Functional Testing: Security and Performance

2.5.5. Test Environment

All the test cases defined on this test plan will be run on the production environment, where the system is deployed.

2.5.6. Test Items

1. Create dataset
2. View dataset
3. Remove dataset
4. Create Workflow
5. Rename workflow
6. Run workflow
7. Publish workflow
8. Consume machine learning model from endpoint
9. Verify that sign-out users have limited access
10. Predict endpoint is accessible when passing a valid API token
11. Verify that response time when creating a dataset complies with performance non-functional requirement
12. Verify that response time when generating a machine learning model complies with performance non-functional requirement
13. Verify that response time when consuming AIaaS endpoint complies with performance non-functional requirement.

2.5.7. Test Cases

2.5.7.1. Test case: Create dataset

2.5.7.1.1. Objective

Verify that an administrator can successfully create a dataset.

2.5.7.1.2. Preconditions

The user is logged into the system as an administrator.

2.5.7.1.3. Steps

1. Navigate to Datasets page.
2. Click on “Add New” button.
3. Select a file from your computer, preferable the csv file: advertising.csv.
4. Enter a name and description.
5. Click on next button.
6. Change the delimiter only if the default value (comma) is not suitable for the selected file.
7. Click on next button.
8. If applicable change the datatype of a column.
9. Click on “Create“ button.

2.5.7.1.4. Expected results

- The dataset is successfully created.
- The system inform that the dataset was successfully created.
- User is redirected to Datasets page, where the new dataset is added into the dataset list.

2.5.7.2. Test case: View dataset

2.5.7.2.1. Objective

Verify that the dataset created is stored into the system and the settings selected were saved.

2.5.7.2.2. Preconditions

The user is logged into the system as an administrator.

2.5.7.2.3. Steps

1. Navigate to Datasets page.
2. Select the dataset previously created and click on “View” button.
3. Verify the information shown on the three available tabs: Details, Preview and Schema.

2.5.7.2.4. Expected results

- The dataset is stored into the system and is visible on the datasets list page.
- When clicking on “View” button, the user is redirected to the dataset view page.
- All the information provided on the three tabs is correct.

2.5.7.3. Test case: Remove dataset

2.5.7.3.1. Objective

Verify that the system allows the user to successfully remove a dataset from the system.

2.5.7.3.2. Preconditions

The user is logged into the system as an administrator.

2.5.7.3.3. Steps

1. Navigate to Datasets page.
2. Select the dataset previously created and click on “Remove” button.
3. Click on “Yes” button on the confirmation dialog.

2.5.7.3.4. Expected results

- A confirmation dialog is popped out.
- The system deletes the dataset from the system when user confirm by clicking on “Yes” button.
- The dataset is removed from the dataset list.

2.5.7.4. Test case: Create Workflow

2.5.7.4.1. Objective

The objective is to verify that user can create a workflow from the UI.

2.5.7.4.2. Preconditions

The user is logged into the system as an administrator.

2.5.7.4.3. Steps

1. Navigate to Workflows page.
2. Click on “Create New” button.
3. Drag and drop following operators: Dataset, Clean Data, Normalize, Split Data, Train Model and Evaluate operator.
4. Configure at least one operator by clicking on the three dots on the right-hand side of the operator.
5. Verify a status description on the top bar followed by the workflow name.
6. Reload the page.

2.5.7.4.4. Expected results

- The system shows a message on the top, beside the workflow name with a status description, with the following pattern: Saved at HH:MM:ss. The time is expressed in 24 four hours format.

- The system has automatically saved the workflow.
- After the reload of the page, the workflow is shown again with all the configuration entered by the user.
- The workflow is added into the workflow list.

2.5.7.5. Test case: Rename workflow

2.5.7.5.1. Objective

The objective is to verify that the workflow can be renamed.

2.5.7.5.2. Preconditions

The user is logged into the system as an administrator.

2.5.7.5.3. Steps

1. Navigate to Workflows page.
2. Select the previously created workflow.
3. Clicks on the edition icon, beside the workflow name.
4. Rename the workflow and enter a description.

2.5.7.5.4. Expected results

- The workflow is successfully renamed.
- The name and description are visible on the workflow list.

2.5.7.6. Test case: Run workflow

2.5.7.6.1. Objective

The objective is to verify that the workflow can be successfully run, and related artefacts are created.

2.5.7.6.2. Preconditions

The user is logged into the system as an administrator and has selected the previously created workflow.

2.5.7.6.3. Steps

1. Edit the first operator, which is dataset.
2. Select the previously created dataset advertising.csv and select all the columns.
3. Configure the normalize operator, with the following settings:
   a. Normalize mode: Log Mean Variance.
   b. Select only the feature columns not the target: TV, Radio, Newspaper.
4. Configure the Split operator with a test fraction of 0.2, that mean that a 20% of the data goes to the test set.
5. Configure the Train Model operator with the following settings:
   a. Label: Sales.
   b. Task: Regression.
   c. Trainer: SDCA Regression.
6. Click on run icon on the top right of the designer.
7. Once the workflow has finished, click “Visualize “option from the Evaluate operator.

2.5.7.6.4. Expected results

- The workflow successfully runs.
- The main spinner appears on the screen, indicating that the workflow is running.
- The operators also show a spinner in the form of an icon and a green check mark.
- The Evaluate operator has the option “Visualize”.
- The model metrics shown from the Visualize option should show similar values to the following:
  - RSquared: 0.8618

- MeanSquaredError: 4.1968
- RootMeanSquaredError: 2.0486
- LossFunction: 4.1968

2.5.7.7. Test case: Publish workflow

2.5.7.7.1. Objective

The objective is to verify that the workflow can be successfully published.

2.5.7.7.2. Preconditions

The user is logged into the system as an administrator and has selected the previously created workflow.

2.5.7.7.3. Steps

1. Click on the publish icon (upload button), beside the run button.

2.5.7.7.4. Expected results

- The system shows the following success message: Endpoint successfully created, and model is marked as published.
- The new endpoint is accessible and listed on the Endpoints page.
- The model is marked as published on the Models list page.

2.5.7.8. Test case: Consume machine learning model from endpoint

2.5.7.8.1. Objective

The objective is to verify that the preciously generate endpoint is accessible for clients.

2.5.7.8.2. Preconditions

The user is logged into the system as an administrator.

2.5.7.8.3. Steps

1. Navigate to endpoints page.
2. Select the previously created endpoint and click in View button.
3. Ensure that the endpoint is enabled.
4. Select the Consume tab.
5. Ensure that the authentication type is None, which is anonymous.
6. Copy the Endpoint URL shown on the page and use it on Postman.
7. In Postman, create a Post request with above URL.
8. Ensure that “No Auth” is selected, this should be the default.
9. Past following Raw JSON body:

```
{
  "tv": 230,
  "radio": 37.5,
  "newspaper": 70.5
}
```

2.5.7.8.4. Expected results

- The endpoint returns HTTP 200 OK.
- The response body should be something like the following JSON:

```
{
  "TV": 230,
  "Radio": 37.5,
  "Newspaper": 70.5,
    "Sales": 20.456013
}
```

2.5.7.9. Test case: Verify that sign-out users have limited access

2.5.7.9.1. Objective

The objective is to verify that signed-out users have limited access to the system, and they are not able to access to any administrative page.

2.5.7.9.2. Preconditions

The user is logged-out from the system.

2.5.7.9.3. Steps

1. Navigate to home page: https://mh-aiaas.herokuapp.com/
2. Check the content on the toolbar on the top.
3. Check if the navigation menu on the left-hand side is visible.
4. Click on “Start work” buttons.
5. Verify the access to the following URLS:
    - https://mh-aiaas.herokuapp.com/datasets
    - https://mh-aiaas.herokuapp.com/workflows
    - https://mh-aiaas.herokuapp.com/models
    - https://mh-aiaas.herokuapp.com/endpoints
    - https://mh-aiaas.herokuapp.com/profile

2.5.7.9.4. Expected results

- On the tool bar, only the sign in button is visible, and it redirects to the user to log in page.
- No navigation menu is shown on the left.
- When clicking on any of the start button, the user is redirected to the log in page.
- When trying to access to any of the URLs provides, the user is redirected to log in page.

2.5.7.10. Test case: Predict endpoint is accessible when passing a valid API token

2.5.7.10.1. Objective

The objective is to verify that the predict endpoint is correctly protected when selecting token based as authentication method.

2.5.7.10.2. Preconditions

A machine learning model was generated and published, and a predict endpoint has been created with authentication type as token based.

2.5.7.10.3. Steps

1. Open Postman application or similar.
2. Use following URL to create a post request and replace with a valid workflow id: https://ec2-3-249-105-85.eu-west-1.compute.amazonaws.com/api/predict/{workflow_id}
3. Select authentication Bearer and enter the token shown on endpoint page.
4. Enter the following raw JSON body:

```
{
    "tv": 230,
    "radio": 37.5,
    "newspaper": 70.5
}
```

5. Send the request.
6. Try an invalid token or “none” authentication on postman.

2.5.7.10.4. Expected results

- The system returns a HTTP 200 OK with a valid prediction when passing a valid token.
- If the token is invalid or the authentication method is none, the system returns HTTP 401 Unauthorized.

2.5.7.11. Test case: Verify that response time when creating a dataset complies with performance non-functional requirement

2.5.7.11.1. Objective

The objective is to verify that the system complies with the non-functional requirements related to the response time when creating a dataset.

2.5.7.11.2. Preconditions

- The user is signed-in as a system’s administrator.
- The user has downloaded the dataset: taxi-fare-full.csv

2.5.7.11.3. Steps

1. Navigate to datasets page.
2. Follows the steps to create a dataset using a file of 5MB (Taxi-fare-full.csv)
3. Measure the response time to create the dataset (use browser developer tools, Network tab)

2.5.7.11.4. Expected results

- The dataset is successfully created.
- The response time is under 30 seconds.

2.5.7.12. Test case: Verify that response time when generating a machine learning model complies with performance non-functional requirement

2.5.7.12.1. Objective

The objective is to verify that the system complies with the non-functional requirements related to the response time when generating a machine learning model.

2.5.7.12.2. Preconditions

- The user is signed-in as a system’s administrator.
- The user has created a ML workflow for dataset: taxi-fare-full.csv

2.5.7.12.3. Steps

1. Navigate to the workflow designer for dataset taxi-fare-full.csv.
2. Click in run button on the top right of the designer.
3. Verify the “Duration” field on the top right panel (Workflow Properties).

2.5.7.12.4. Expected results

- The workflow was successfully run, and the model was generated.
- The Duration time is under 30 seconds.

2.5.7.13. Test case: Verify that response time when consuming AIaaS endpoint complies with performance non-functional requirement.

2.5.7.13.1. Objective

The objective is to verify that the system complies with the non-functional requirements related to the response time when consuming AIaaS endpoint.

2.5.7.13.2. Preconditions

- The user is signed-in as administrator.
- The user has created a ML workflow for dataset: taxi-fare-full.csv
- A model was generated and published for the dataset.
- An endpoint was created, is enabled and public.

2.5.7.13.3. Steps

1. Navigate to the Endpoints page.
2. Select endpoint for taxi-fare-full.csv dataset.
3. Get the URL of the predict/AIaaS endpoint from "Consume" tab.
4. Open postman and create a post request with above URL.
5. Use following as raw JSON on the request body:

```
{
  "vendor_id": null,
  "rate_code": 0,
  "passenger_count": 0,
  "trip_time_in_secs": 0,
  "trip_distance": 0,
  "payment_type": null
}
```

6. Send request and measure response time.

2.5.7.13.4. Expected results

- The system responds with a HTTP 200 OK.
- The response time us under 5 seconds.

2.5.8. Test cases execution results

#	Test case	Execution result
1	Create dataset	Passed
2	View dataset	Passed
3	Remove dataset	Passed
4	Create Workflow	Passed
5	Rename workflow	Passed
6	Run workflow	Passed
7	Publish workflow	Passed
8	Consume machine learning model from endpoint	Passed
9	Verify that sign-out users have limited access	Passed
10	Predict endpoint is accessible when passing a valid API token	Passed
11	Verify that response time when creating a dataset complies with performance non-functional requirement	Passed
12	Verify that response time when generating a machine learning model complies with performance non-functional requirement	Passed

13	Verify that response time when consuming AIaaS endpoint complies with performance non-functional requirement.	Passed

2.6. Evaluation

The evaluation of the implemented solution primarily focuses on three key factors: user-friendliness, correctness, and performance. To establish a benchmark for assessing the outcomes, I have created identical machine learning workflows or pipelines for each dataset. One of the machine learning workflows was implemented with the proposed AIaaS solution, while the other was implemented with Jupyter Notebook using scikit-learn, which is a popular machine learning library for Python.

User friendliness is a subjective evaluation factor, and on this context indicates how easy is for the user to interact with a given system and how easy is to implement a machine learning workflow or pipeline.

Regarding to the correctness evaluation, I have considered two aspects. First, it was analysed the metrics obtained from the generated machine learning model from both machine learning pipelines. Secondly, it was assessed the system's ability to predict a random sample.

On the performance evaluation, both workflows underwent multiple executions, and it was calculated the average execution time.

For this evaluation analysis, it was utilized well-known and widely used datasets that are commonly utilized in the field. These datasets are used not only for learning but also as established benchmarks for evaluating and comparing various machine learning algorithms and models. Below it is shown the datasets used on this experiment with relevant information such as size, quantity of columns and rows.

2.6.1. Datasets

Dataset	Size	Columns	Rows
advertising.csv	3.97 KB	4	200
housing.csv	1.12 MB	10	20,640
taxi-fare-full.csv	4.89 MB	7	200,000
creditcard.csv	143.84 MB	31	284,807
diabetes.csv	23.31 KB	9	768
titanic.csv	59.76 KB	12	891

2.6.2. Results

2.6.2.1. User-friendliness

From a user perspective, comparing the implemented solution AIaaS versus Jupyter Notebook using python scikit-learn, the product AIaaS provides a much faster and intuitive user interface compared to Jupyter Notebook or any IDE for writing python code. This is simply because when using scikit-learn, the user requires some degree of knowledge on the python language itself, on the library and on machine learning. Instead, with the implemented solution of AIaaS, the user does not need to know how to code in python or about the library scikit-learn.

The product AIaaS enables the user to quickly create, run and evaluate a machine learning workflow in no time. Additionally, this product provides some interesting features such as copy and paste workflows, create from template, preview of intermediate data that simply cannot be compared with the experience a user can have with Jupyter Notebook. Additionally, the solution offers a test screen on Endpoint page that allows the user to quickly test the prediction service without the need of external tools such as postman.

2.6.2.2. Accuracy

To evaluate the system in terms of accuracy it was evaluated the model generated by comparing two main metrics from both machine learning pipelines, one from the product AIaaS, the other from scikit-learn.

The following metrics were utilized to evaluate the accuracy of the model generated:

- R Squared [1]: it provides information about how well data fit the model (regression).
- Mean Squared Error (MSE) [13]: describe how close the regression line is to a set of data points.
- Accuracy [4]: fraction of predictions that were correct.
- F1 score [14]: it is a measure of the models' accuracy that considers the false positive and false negatives.

Overall, the AIaaS solution performed very well, and in some cases, it outperformed scikit-learn. For example, in binary classification tasks (creditcard.csv, diabetes.csv, and titanic.csv datasets), the AIaaS solution achieved comparable accuracy and F1 scores. On the linear regression tasks, the AIaaS solution outperformed scikit-learn.

2.6.2.2.1. Metrics comparison

scikit-learn code	ML task	Metric	AIaaS	scikit-learn
advertising.ipynb	Linear regression	R Squared	0.9124	0.875
advertising.ipynb	Linear regression	Mean squared error	2.6741	2.409
housing.ipynb	Linear regression	R Squared	0.6338	0.628
housing.ipynb	Linear regression	Mean squared error	0.0839	0.367
taxi-fare.ipynb	Linear regression	R Squared	0.7597	0.741
taxi-fare.ipynb	Linear regression	Mean squared error	24.5355	28.310
creditcard.ipynb	Binary classification	Accuracy	0.9987	0.9992
creditcard.ipynb	Binary classification	F1 score	0.6864	0.84
diabetes.ipynb	Binary classification	Accuracy	0.75	0.7792
diabetes.ipynb	Binary classification	F1 score	0.6019	0.74
titanic.ipynb	Binary classification	Accuracy	0.8032	0.7877
titanic.ipynb	Binary classification	F1 score	0.7376	0.77

2.6.2.2.2. Ability to predict value

On this evaluation analysis, it was measured the AIaaS solution's ability to predict values compared to scikit-learn for specific test cases. To simplify the experiment the first value from first row was taken as target. In most cases, the AIaaS solution's predictions were very close to those from scikit-learn, indicating good accuracy and consistency.

scikit-learn code	AIaaS	scikit-learn	Value to be predicted (from first row)
advertising.ipynb	21.0456	21.1454	22.1
housing.ipynb	1.7942	1.7858	2.12
taxi-fare.ipynb	16.702	16.7478	17.5
creditcard.ipynb	false	false	false
diabetes.ipynb	true	true	true
titanic.ipynb	false	false	false

2.6.2.3. Performance

To evaluate the performance on the implemented solution against python scikit-learn, multiple execution times were recorded for the same machine learning task comparing both solutions. The results showed that the AIaaS solution generally performed well, with execution times often comparable or even faster than scikit-learn.

Something that brought my attention was the ability to outperform scikit-learn while processing large datasets, for example while processing the dataset creditcard.csv which is around 150MB. That result seems to be in accordance with the information provided by Microsoft [15] in relation to the performance of ML.NET while processing large datasets.

scikit-learn code	AIaaS	scikit-learn
advertising.ipynb	348.6ms	257.7875ms
housing.ipynb	9.21s	333.2871ms
taxi-fare.ipynb	8.1746s	651.424ms
creditcard.ipynb	10.849s	11.4338s
diabetes.ipynb	595.7143ms	87.26429ms
titanic.ipynb	719.5ms	251.7714ms

3.0. Conclusions

As a conclusion, my experience while working on this project was extremely positive and it was very challenging from the beginning until the end. The project has given me a great opportunity to explore some of the most modern and diverse technologies, that normally I would not have a chance to test or use on my current role as software engineer. As an unexpected benefit, the development of this solution from scratch has provided me a lot of experience in terms of technologies, that I was able to use on my current job allowing me to be seen as a more experienced professional. Now, I would like to mention some of the aspects that represents advantages, disadvantages strength or limitation of the project.

3.1. Advantages and strengths

The development of the Artificial Intelligence as a Service (AIaaS) solution has resulted in a highly promising and user-friendly solution with great potential for implementing machine learning workflows in a graphical manner.

Through the evaluation of user-friendliness, correctness, and performance, the AIaaS project demonstrates significant advantages compared to the traditional approach of using Jupyter Notebook with Python and scikit-learn. It was able to deliver comparable accuracy and prediction capabilities. Even on performance terms, the solution has shown a great potential and has completely exceeded my own expectations, which I had defined on the initial non-functional requirements related to performance.

The solution offers a seamless user interface, allowing users to quickly create, execute, and evaluate machine learning tasks without the need of having knowledge on the Python language or machine learning algorithms. Additionally, the Test screen on Endpoint page provides a convenient way to test prediction services directly within the website, without the need of using external tools.

In relation to the architecture, the solution is already prepared to handle large dataset files as it is integrated with AWS S3 storage services. The solution could use auto scale in the future if needed, as each node is able to work independently. Because it is integrated with CloudWatch, the solution provides access to logs in real-time, which is fundamental to troubleshoot production issues. In addition, CloudWatch metrics offers a good enough view of the EC2 resource and could allow in a later stage the addition of more metrics from .NET.

In addition to the advantages previously mentioned, the solution offers a good level of security by integrating the product to Auth0 and protecting the prediction endpoint using token-based authentication method or JWT from Auth0.

3.2. Disadvantages and limitations

One possible disadvantage of this product could be that the user interface for creating workflows could be not suitable for complex graphs. Now the library that is being used allows only to create diagrams in the form of pipelines with some degree of freedom. But if more complex graphs are required, the product should migrate the user interface to other library for graph representation. In relation to troubleshooting, my experience was very good as the solution provides good enough information when there is an error. But in a very few situations I had to look at the logs to understand the problem.

One of the limitations of the architecture implemented is that the machine learning workflow run within a request scope. That is fine for relatively short processing time like under five or ten minutes.

The solution offers a limited quantity of machine learning tasks, only regression and classification are provided. Moreover, the metrics provided could be not enough and there are no charts to visualize the results.

4. Further Development or Research

To decide which direction to take, this product should be showcased to potential customers to understand the business needs, which can be very different from the developer perspective. From my point of view, the addition of more machine learning tasks would be beneficial, especially those tasks related to natural language processing, image classification, autoencoders, and deep learning [4].

It would enhance the product if more metrics were added, furthermore the addition of charts would help the user to understand the results by visualizing the metrics. To finalize, there is more work to be done on the visualization of intermediate data, which is the output of each operator. As well as adding more workflow operators.

5. References

[1] S. Jansen, Machine learning for algorithmic trading : predictive models to extract signals from market and alternative data for systematic trading strategies with Python. Birmingham ; Mumbai: Packt Publishing, 2020.
[2] K. Hazzard and J. Bock, Metaprogramming in .NET. Manning, 2013.
[3] V. D. Sanctis, ASP.NET Core 2 and Angular 5. Birmingham: Packt Publishing, 2017.
[4] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, Massachusetts: The Mit Press, 2016.
[5] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design patterns : elements of reusable object-oriented software. Boston: Addison-Wesley, 1994.
[6] M. Fowler, Patterns of enterprise application architecture. Boston, Mass. ; Munich: Addison-Wesley, 2015.
[7] J. Taylor, Clean Architecture with .NET Core: Getting Started – Jason Taylor. https://jasontaylor.dev/clean-architecture-getting-started/
[8] S. Smith Ardalis, Clean Architecture, https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures#clean-architecture

[9] AWS, Cloud Object Storage | Store & Retrieve Data Anywhere | Amazon Simple Storage Service, Amazon Web Services, Inc., 2018. https://aws.amazon.com/s3/

[10] AWS, “Amazon CloudWatch - Application and Infrastructure Monitoring,” Amazon Web Services, Inc., 2018. https://aws.amazon.com/cloudwatch/

[11]“@ngrx,” GitHub, Aug. 06, 2023. https://github.com/ngrx/platform

[12] EdPrice-MSFT, “CQRS pattern - Azure Architecture Center,” learn.microsoft.com. https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs

[13] Stephanie Glen. "Mean Squared Error: Definition and Example" From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/

[14] N. Sharma, Understanding and Applying F1 Score: A Deep Dive with Hands-On Coding, https://arize.com/blog-course/f1-score

[15] ML.NET | Machine Learning made for .NET, Microsoft. https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet

6. Appendices

6.1. Project Proposal

National College of Ireland

Project Proposal

Artificial intelligence as a service (AIaaS)

2022

BSc (Honours) in Computing

Software Development

2022/2023

Maximiliano Herrera

X20103212

x20103212@student.ncirl.ie

Contents

6.1.1 Objectives

The main goal for this project is to build a cloud-based and distributed solution that would provide and facilitate the use of Artificial Intelligence services to enterprise applications.

6.1.1.1 Cloud-based solution

This solution is intended to function on the Cloud as a distributed system, and it would follow a Software as a Service approach.

6.1.1.2 Provide and facilitate the use of Artificial Intelligence services

My implementation of Artificial Intelligence as a Service (or AIaaS) will allow to create machine learning (ML) models and make them available to be consumed from the Cloud.

The creation of a ML model would follow a graphical approach and will support the implementation of a typical machine learning workflow containing tasks such as: data input, data pre-processing, selection of features and target, selection of most fitted ML algorithm, training, and model evaluation.

6.1.1.3 Enterprise applications as customers

This product is not intended to be used by a regular user; it is intended for enterprise applications. And by enterprise applications I mean applications or systems that operates in a corporate environment.

6.1.2 Background

Back in 2010 when I started to work as a Software developer, I had a chance to work on a small OCR (Optical character recognition) that was implemented using a neural network. That was my first contact with Artificial Intelligence and since then I started to feel attracted to the field.

Recently, I had the opportunity to work on a ETL on the financial industry. The ETL was quite old, and the configuration tool was outdated and quite complex to manage. It was so complex than a team was required for the configuration. One of my tasks was to modernize the ETL following a graphical approach to represent data transformations with React, like SQL Server Integration Services (SSIS), where you have a designer that allows you to represent data flows in the form of interconnected nodes. That experience is one of the reasons for me to be interested on building a graphical automation tool.

I consider that Artificial Intelligence as a Service (AIaaS) is a mix of the two things that I would like to work with. Without a doubt, working on a project like this would be an enriching experience and will deepen my knowledge not only on backend technologies, but also on front-end, AI and Cloud technologies.

To meet my objectives, I have created a detailed plan of work following Kanban as my software development framework. I have detailed principal Milestones (in the form of Releases), epics (large piece of work) broken down into sub tasks with start date, end date and an estimated duration. The plan includes a roadmap that offers a Gantt view of the project, which is used for planning long piece of work and dependency management. I will use Jira Board for tasks management, which will be shared with my project supervisor to give a real time progress status.

6.1.3 State of the Art

6.1.3.1 Similar products to AIaaS on the market

AIaaS is a product that many of the top companies are working on it, and they already have a product on the market. Some of the most relevant examples are:

- Azure Machine Learning from Microsoft.
- Watson Machine Learning from IBM.
- Amazon SageMaker.
- Vertex AI from Google.

6.1.3.2 Solution proposed

My solution will stand out over current products on the market and will differ from them because:

- It will not be coupled to Cloud provider resources: some of the products named before are coupled to Cloud provider resources where the product runs. For example, in Azure ML to create a ML model first the user must create a lot of cloud resources such as workspace, storage account key vault, app insight, container register, etc.
- It will not follow Jupyter notebook approach like Amazon SageMaker.
- It will be a Software as a Service product.
- This solution will take a graphical approach to design and create ML models.
- It will be simple to use from user perspective.

6.1.4 Technical Approach

6.1.4.1 Development approach

I will use Kanban from Jira Software as development approach. Why Kanban? As I need to work as much efficient as possible due to time constraints and technical challenges, I found that Kanban could be a suitable agile process for a single developer. It looks much lighter and more flexible than Scrum because it does not have Sprints and all the overhead related to it.

About Jira Software, I have chosen this tool because it provides a powerful and easy to understand “Board” for tasks management, that contains just three columns: To Do, In Progress and Done. This visual management of tasks that can describe the current state of progress. Moreover, can be easily shared with the project supervisor.

For planning and dependency management I will use Jira “Roadmap” that provides a similar experience to Gantt, but it follows an agile approach.

As benefit of using the same platform for the Board and Roadmap is that there is a bidirectional connection between them. That makes easier to apply changes on the Roadmap and reflect the changes on the Board tasks. At the same time the Roadmap allows to visualize the current progress from the Board, for example to visualize on the Gantt the current task status (in progress, done or to do). To conclude on the benefits, using the same platform avoid duplication of efforts because there is no need to create same tasks on Gantt and tasks management.

6.1.4.2 Jira links

NB: permissions must be granted to access below resources.

- Roadmap: https://mh-aiaas.atlassian.net/jira/software/projects/MA/boards/3/roadmap
- Board: https://mh-aiaas.atlassian.net/jira/software/projects/MA/boards/3

6.1.4.3 Requirements

Requirements will be identified from main functionalities of this solution, and they will be detailed on requirements specification document. The requirements will be broken down into epics (large piece of work that can be broken down into task) and tasks following Kanvan framework.

6.1.4.4 Milestones

The major milestones will be identified as Releases on Jira Software. That is because Jira software does not support the concept of Milestones. For that reason, I have created Releases that can be shown on the roadmap (Gantt) and in that way I could emulate the Milestone concept. Releases also help to understand what tasks are needed for a deliverable because tasks can be linked to a release. A release also provides useful information such as release date, description, and progress.

6.1.5 Technical Details

6.1.5.1 Implementation languages

6.1.5.1.1 Backend

- C# .NET Core 3.1 or .NET 6 depending on support from Cloud providers.
- SQL Server, or PostgreSQL or MySQL, also depending on support from Cloud providers.

6.1.5.1.2 Frontend

- Angular 15
- HTML
- CSS
- Typescript
- Angular Material or Kendo or similar
- NGRX, state management for Angular

6.1.5.2 Principal libraries

- ML.NET: Machine learning algorithms for C#.
- QuickGraph: Graph manipulation and algorithms.
- ngx-graph: graph visualization library for Angular.

6.1.5.3 Important considerations

- The ML Development Studio (Backend) and UI could run on a cheap PaaS environment such as Heroku. As this system aims to be distributed, the two nodes should run separately. This could enable the architecture to escalate horizontally.
- It is important to build a proof of concept (PoC) to generate ML models and evaluate how CPU-intensive it is. Depending on the results it could be necessary to extract that processing into a new node, which is detailed in the next section (Special Resources Required section).

6.1.6 Special Resources Required

- It is important to consider that generating ML models can be CPU-Intensive and it might be necessary to extract and delegate that processing on an exclusive node which I call “Workflow Runner”. That node could execute on AWS Lambda or Azure Function, that are billed for the time the process is running. This will be implemented if the plan goes as planned and PoC shows slowness while generating ML models.
- The same applies for the AIaaS endpoint, which enables the client to consume a ML model and get predictions or other AI services.
- To store a ML workflow, which is shown on the UI as a graph, it could be required to use a NoSQL database. That is because a graph is difficult to store on a relational database but is easy on a NoSQL as JSON.

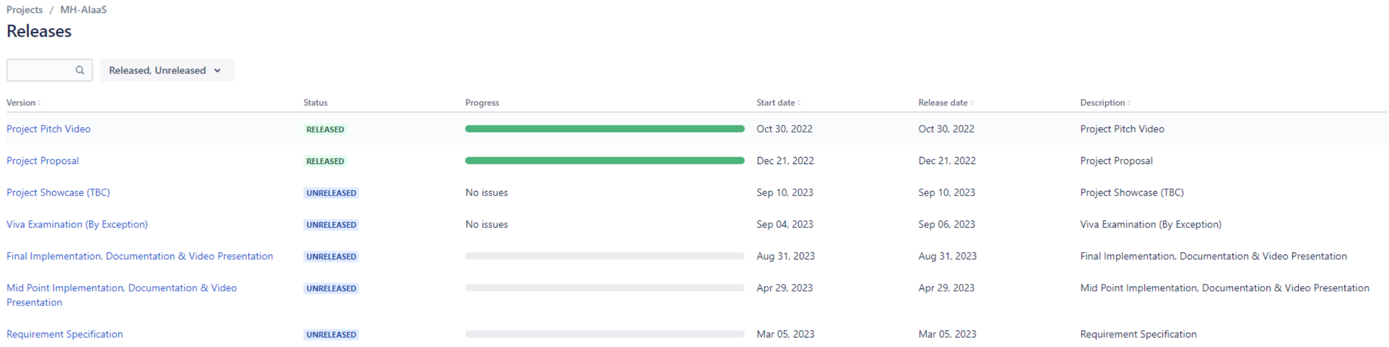
6.1.7 Project Plan

The overall project plan will be detailed below through the following sections: Releases, Reflective Journals, and Roadmap.

6.1.7.1 Releases

As commented before, I am using Releases from Jira Software to emulate milestones, which in theory are tasks with zero duration that represents deliverables.

Below there is a list of releases with a release date, status, name, and description. Reflective journals are excluded from releases just to make cleaner the roadmap and make it easy to understand. Reflective journals will be detailed in next section.



Projects / MH-AlaaS

Releases

Released, Unreleased

Version	Status	Progress	Start date	Release date	Description
Project Pitch Video	RELEASED		Oct 30, 2022	Oct 30, 2022	Project Pitch Video
Project Proposal	RELEASED		Dec 21, 2022	Dec 21, 2022	Project Proposal
Project Showcase (TBC)	UNRELEASED	No issues	Sep 10, 2023	Sep 10, 2023	Project Showcase (TBC)
Viva Examination (By Exception)	UNRELEASED	No issues	Sep 04, 2023	Sep 06, 2023	Viva Examination (By Exception)
Final Implementation, Documentation & Video Presentation	UNRELEASED		Aug 31, 2023	Aug 31, 2023	Final Implementation, Documentation & Video Presentation
Mid Point Implementation, Documentation & Video Presentation	UNRELEASED		Apr 29, 2023	Apr 29, 2023	Mid Point Implementation, Documentation & Video Presentation
Requirement Specification	UNRELEASED		Mar 05, 2023	Mar 05, 2023	Requirement Specification

6.1.7.2 Reflective Journals

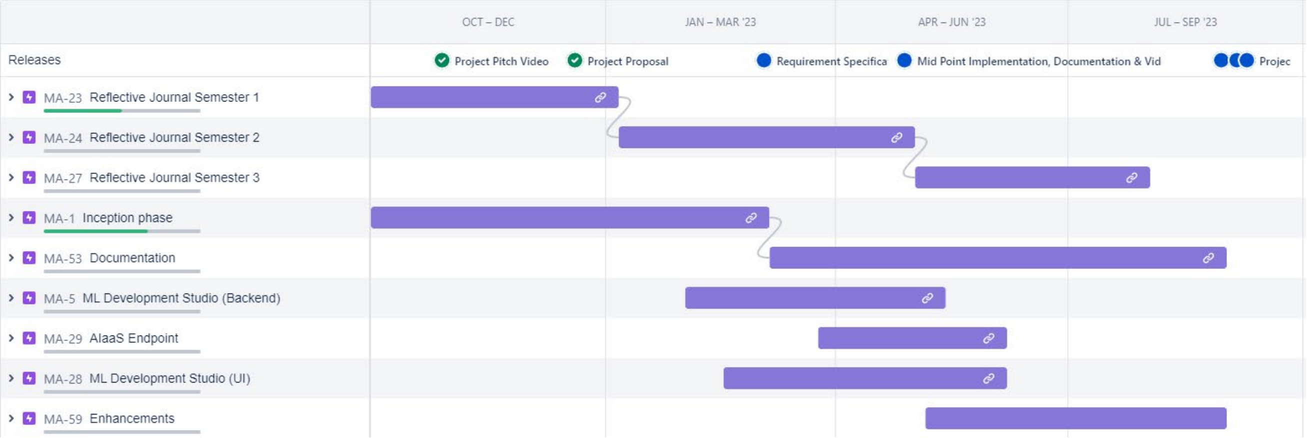
Reflective journals are a summary of progress done on a lapsus of 1 moth time, like a retrospective on Scrum. Below there is a detailed list of the journals to be submitted. Notice that Journals are shown on Roadmap and can be found as tasks on the Backlog or Board.

- November Journal due December 1st, 2022
- December Journal due January 5th, 2023
- January Journal due February 1st, 2023
- February Journal due March 1st, 2023
- March Journal due April 1st, 2023
- April Journal due May 1st, 2023
- May Journal due June 1st, 2023
- June Journal due July 1st, 2023
- July Journal due August 1st, 2023

6.1.7.3 Roadmap

A Roadmap in Jira allows to visualize and plan long term activities and is shown in the form of a Gantt chart. On the header of Roadmap, we can find the dates by column and on the first row we have the Releases, which are deliverables. Finally, there is a list of epics and tasks in a Gantt fashion.

6.1.7.3.1 Roadmap at high level



OCT – DEC
JAN – MAR '23
APR – JUN '23
JUL – SEP '23
Releases
Project Pitch Video
Project Proposal
Requirement Specifica
Mid Point Implementation, Documentation & Vid
Projec
MA-23 Reflective Journal Semester 1
MA-24 Reflective Journal Semester 2
MA-27 Reflective Journal Semester 3
MA-1 Inception phase
MA-53 Documentation
MA-5 ML Development Studio (Backend)
MA-29 AlaaS Endpoint
MA-28 ML Development Studio (UI)
MA-59 Enhancements

6.1.7.3.2 Description of Epics

6.1.7.3.2.1 Reflective Journal Semester 1/2/3

- Start: 1st October 2022
- End: 31st August 2023 (Final Implementation, Documentation & Video Presentation)
- Duration: 10 months and 30 days

It contains tasks related to the reflective Journal documents and they can be worked in parallel with other activities such as inception, documentation, and development.

6.1.7.3.2.2 Inception phase

- Start: 1st October 2022
- End: 5th March 2023
- Duration: 5 months, 4 days

It contains activities that occurs at the beginning of the project and allows to define the scope and requirements for it. These activities are:

- Project Pitch video
- Project Proposal
- Requirement specification

6.1.7.3.2.3 Documentation

- Start: 6th March 2023
- End: 31st August 2023 (Final Implementation, Documentation & Video Presentation)
- Duration: 5 months, 25 days.

It contains activities for the documentation of the project, which can be worked in parallel to other activities. There are only two tasks for this epic:

- Midpoint documentation & slide
- Complete documentation & slide

6.1.7.3.2.4 ML Development Studio (Backend)

- Start: 1st February 2023
- End: 13th May 2023
- Duration: 3 months, 12 days

It contains all the tasks for building the backend logic required for this solution, this module is the core of AIaaS, and the main responsibilities can be outlined below:

- Store machine learning (ML) configuration, coming from UI as a graph.
- Generate ML models.
- Store ML models to be consumed by AIaaS endpoint.

6.1.7.3.2.5 AIaaS Endpoint

- Start: 25th March 2023
- End: 6th June 2023
- Duration: 2 months, 12 days

Activities required to build a generic endpoint, that would allow to consume ML models.

6.1.7.3.2.6 ML Development Studio (UI)

- Start: 16th February 2023
- End: 6th June 2023
- Duration: 3 months, 21 days

Group of tasks associated to the development of User Interface. Main responsibilities:

- Allow user to configure and generate a ML model.

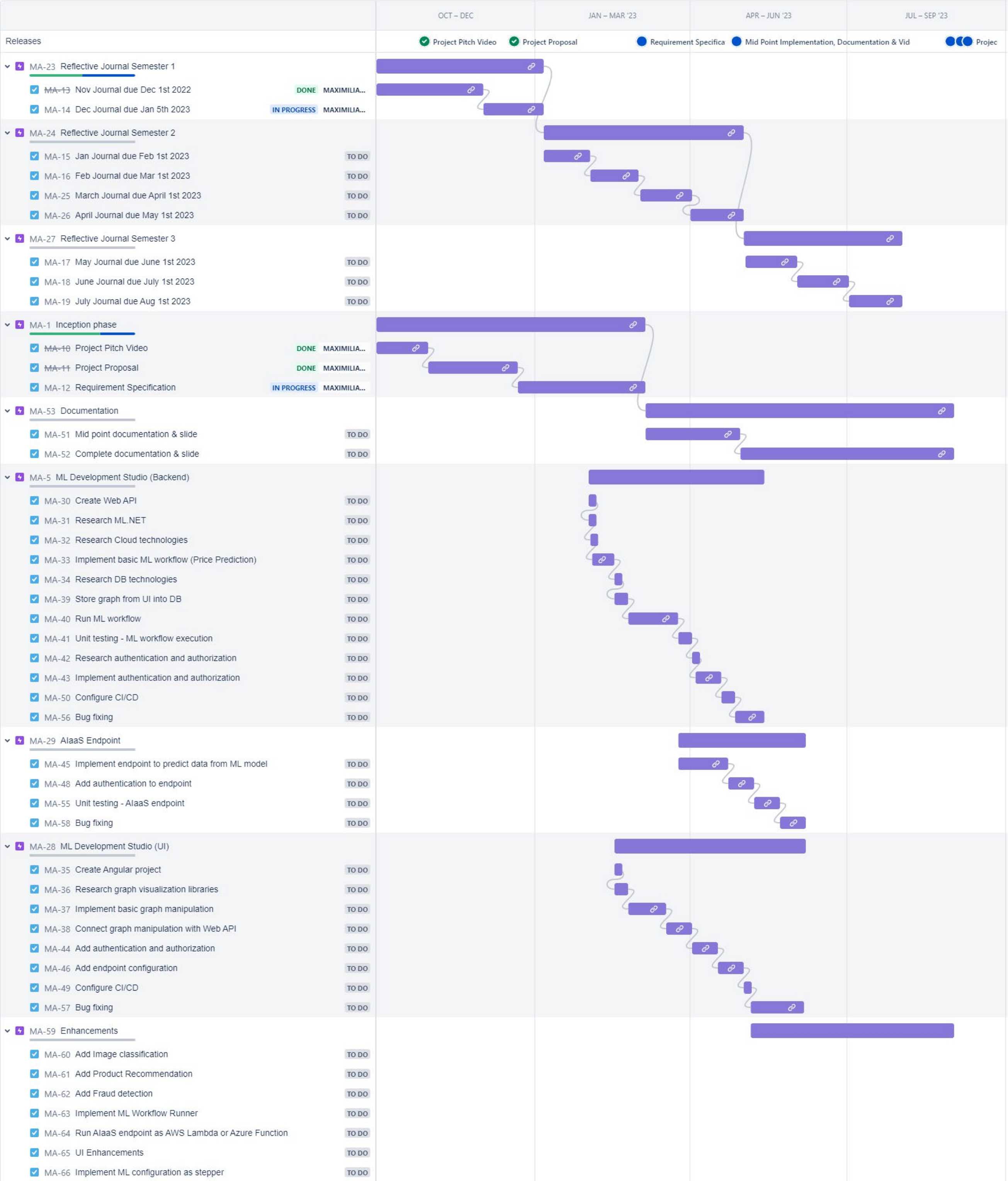
- Configure the AIaaS endpoint.
- Provide user authentication and authorization.

6.1.7.3.2.7 Enhancements

- Start: 6th May 2023
- End: 31st August 2023 (Final Implementation, Documentation & Video Presentation)
- Duration: 3 months, 25 days

Optional tasks related to improve the solution’s architecture. They will be carried out if there is enough time or if they are really needed. For example, there is the workflow runner which is thought to run CPU-intensive processes on AWS Lambda or Azure Function. That component only will make sense to implement it if running ML models in .NET is too slow for a cheap environment such as Heroku.

6.1.7.3.2.8 Roadmap detailed



OCT – DEC | JAN – MAR '23 | APR – JUN '23 | JUL – SEP '23

Releases: ✓ Project Pitch Video ✓ Project Proposal ● Requirement Specifica ● Mid Point Implementation, Documentation & Vid ●● Projec

Key	Summary	Status	Assignee
MA-23	Reflective Journal Semester 1		
~~MA-13~~	Nov Journal due Dec 1st 2022	DONE	MAXIMILIA...
MA-14	Dec Journal due Jan 5th 2023	IN PROGRESS	MAXIMILIA...
MA-24	Reflective Journal Semester 2		
MA-15	Jan Journal due Feb 1st 2023	TO DO	
MA-16	Feb Journal due Mar 1st 2023	TO DO	
MA-25	March Journal due April 1st 2023	TO DO	
MA-26	April Journal due May 1st 2023	TO DO	
MA-27	Reflective Journal Semester 3		
MA-17	May Journal due June 1st 2023	TO DO	
MA-18	June Journal due July 1st 2023	TO DO	
MA-19	July Journal due Aug 1st 2023	TO DO	
MA-1	Inception phase		
~~MA-10~~	Project Pitch Video	DONE	MAXIMILIA...
~~MA-11~~	Project Proposal	DONE	MAXIMILIA...
MA-12	Requirement Specification	IN PROGRESS	MAXIMILIA...
MA-53	Documentation		
MA-51	Mid point documentation & slide	TO DO	
MA-52	Complete documentation & slide	TO DO	
MA-5	ML Development Studio (Backend)		
MA-30	Create Web API	TO DO	
MA-31	Research ML.NET	TO DO	
MA-32	Research Cloud technologies	TO DO	
MA-33	Implement basic ML workflow (Price Prediction)	TO DO	
MA-34	Research DB technologies	TO DO	
MA-39	Store graph from UI into DB	TO DO	
MA-40	Run ML workflow	TO DO	
MA-41	Unit testing - ML workflow execution	TO DO	
MA-42	Research authentication and authorization	TO DO	
MA-43	Implement authentication and authorization	TO DO	
MA-50	Configure CI/CD	TO DO	
MA-56	Bug fixing	TO DO	
MA-29	AIaaS Endpoint		
MA-45	Implement endpoint to predict data from ML model	TO DO	
MA-48	Add authentication to endpoint	TO DO	
MA-55	Unit testing - AIaaS endpoint	TO DO	
MA-58	Bug fixing	TO DO	
MA-28	ML Development Studio (UI)		
MA-35	Create Angular project	TO DO	
MA-36	Research graph visualization libraries	TO DO	
MA-37	Implement basic graph manipulation	TO DO	
MA-38	Connect graph manipulation with Web API	TO DO	
MA-44	Add authentication and authorization	TO DO	
MA-46	Add endpoint configuration	TO DO	
MA-49	Configure CI/CD	TO DO	
MA-57	Bug fixing	TO DO	
MA-59	Enhancements		
MA-60	Add Image classification	TO DO	
MA-61	Add Product Recommendation	TO DO	
MA-62	Add Fraud detection	TO DO	
MA-63	Implement ML Workflow Runner	TO DO	
MA-64	Run AIaaS endpoint as AWS Lambda or Azure Function	TO DO	
MA-65	UI Enhancements	TO DO	
MA-66	Implement ML configuration as stepper	TO DO	

6.1.8 Testing

Testing will be fundamental on this solution not only for validation/verification of requirements or the correct functioning of modules but also to compare ML model metrics between ML.NET and ML code from Jupyter notebook. Testing will be implemented as a combination between automated and manual testing.

6.1.8.1 Automated testing

To validate and verify the correct functioning of the solution, automated testing will be implemented and added into the pipeline of Continuous Integration and Continuous Delivery (CI/CD).

The automated testing could include unit tests, integration tests and functional tests implemented at backend and frontend level. End to end tests could be automated with Selenium, but this would be done if there is enough time for it. The reason is that the UI for this solution will be quite complex and could be very difficult to incorporate Selenium.

6.1.8.1.1 Backend

Backend includes ML Development Studio (Backend) and AIaaS Endpoint modules, and both will be implemented with C#. Having said that, I will utilise NUnit as C# unit testing framework combined with a mocking library called Moq. Those two tools will allow me to automate my tests and add a verification step on the CI/CD pipeline.

6.1.8.1.2 Front-end

The frontend for this solution will be implemented on Angular, moreover I will use Angular testing to implement and automate UI test cases. Automated testing in UI will be much simpler than backend, it will be part of the CI/CD and will test important components such as authentication.

6.1.8.2 Manual testing

6.1.8.2.1 End-to-end tests

End to end tests will be executed manually, and they will validate complex scenarios of user interaction with the system. Below there are some examples of end-to-end tests:

- Authentication at UI level.
- Authentication at endpoint level.
- Users' management.

6.1.8.2.2 Acceptance testing

It will also be executed as manual testing and they will verify business requirements. Some examples of acceptance testing are:

- Create a ML model from UI.
- Evaluate a ML model.
- Consume data from a model through the AIaaS endpoint.

6.1.8.2.3 Machine learning model evaluation

As one of the main functionalities of this solution is to generate ML models, I will use manual testing to compare results against ML code from Jupyter notebook. In this way, I would be able to compare ML model evaluation metrics such as: R-Squared, mean squared error (MSE), mean absolute error (MAE), score, etc.

The data sets for ML model evaluation would be taken from:

- https://github.com/dotnet/machinelearning-samples
- https://www.kaggle.com/

6.2. Reflective Journals

6.2.1. Reflective Journal, semester 1, November 2022

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: November 2022

What?

After completing the project pitch, I did focus on analysing similar products on the market to find out how my proposed solution could differ from it, improve it, and make it simple to use and understand. Followed the analysis, I worked on a research and selection of main technologies to be used on the project, including Cloud platforms. I explored the utilisation of some libraries that could be used on the development of proposed solution. And lastly, I was able to start working on the Project Proposal, but that is still a work in progress.

Summary

- Analysis on similar products on the market.
- Research and selection of main technologies and libraries.
- Started with Project Proposal.

So What?

Research and selection of main technologies and libraries

Backend and frontend technologies

I believe I had a good progress on the definition of main technologies to be used, now I know that I will go for C# .NET on backend and Angular 15 on Frontend. The .NET version could be .NET 6 or the long term 3.1, depending on the Cloud platform to be defined in a later stage. I also ensured that I have the software tools required to start the development phase, such as Visual Studio 2017/2022 and Visual Studio Code.

Cloud platforms

I did an initial analysis of features provided vs pricing. I requested information to IT Help Center from NCI, to understand about Cloud resources available for students. It is still challenging at this stage to choose one over the others, more research needs to be done.

Libraries

I explored some libraries that could be useful, but this is just a starting point:

- QuickGraph: Graph manipulation and algorithms.
- ML.NET: Machine learning algorithms for C#.
- ngx-graph: graph visualization library for Angular.

Challenges remain

- Cloud platforms
- Database technology
- Libraries

Project Proposal

I started working on the document and did progress on the first six points. I found some challenges on the project plan and testing; I will keep working on those two sections.

Challenges remain

- Gantt software tool

- Testing plan

Now What?

Challenges remain

- Cloud platforms: I would like to do a deeper analysis of Cloud resources, maybe I could use a combination between cheap Cloud platform such as Heroku for non-compute intensive resources, and AWS or Azure for compute-intensive resources. More investigation needs to be done.
- Database technology: The technology to be used for database will be defined depending on cloud platform selection. For example, I could go for Postgres over Heroku or Microsoft SQL Server on Azure.
- Libraries: Once development phase starts, I will be testing different libraries and see if they are suitable for proposed solution.
- Gantt software tool: Probably I would end up using a known application “ganttpro”, but it is priced. I will research for free applications or if I can get a license of Microsoft Project.
- Testing plan: I need to read more about testing to understand better how I could design a test plan.

Student Signature	Maximiliano Herrera

6.2.2. Reflective Journal, semester 1, December 2022

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: December 2022

What?

This month I have completed and submitted the project proposal and started to work on the requirements specification document.

So What?

Project progress

From the project progress perspective, completing the project proposal document was a very important and fundamental step because now I have a very detailed plan about what need to be done to implement my solution and have a test plan in place, both will be reviewed and amended if needed during the Software development life cycle (SDLC).

What went well

I believe that the project proposal document was a success on terms of project progress and planning. I found very useful the following items from proposal document:

- Objectives: Definition of high-level objectives.
- Technical approach: Introduction of development approach to be follow, tools and relevant notes.
- Technical details: Definition of main programming languages and libraries to be used and important considerations that I should keep in mind during the project.
- Special Resources Required: Hypothetic resources required during SDLC of project.
- Project Plan: detailed project plan with epics and subtasks, with a clear description of dependency between them. Epic and subtasks include a relevant name, start and end date and status. Releases and Journals are listed with a due date and clear name. I also provide a roadmap (or Gantt) with a description of every epic.
- Testing: I introduced my testing plan that describes how I am thinking to validate and verify that my solution achieves all the specified requirements.

What challenges remain?

- Requirement specification document.
- Requirements identification and use cases description.
- Improve communication with project supervisor.

Now What?

I will work more closely with my project supervisor to give updates of my progress while working on requirement specification document and get feedback to apply any change or amend. If possible, I would like to have recurrent meetings with my supervisor, to improve the communication and push myself with the project.

Student Signature	Maximiliano Herrera

6.2.3. Reflective Journal, semester 2, January 2023

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: January 2023

What?

This month I have been working on the requirements specification document and did some research on Identity and Access Management (IAM) solutions.

So What?

At the moment, I have completed the functional requirements with the specification of use cases and use case diagrams. Now I have a better understanding of each requirement, trigger, main flow and let me think about the screens to be implemented. I also added a “requirements priority and dependency chart” that was recommended by my supervisor, so now the priority and dependency between requirements is crystal clear.

As challenges remain, I am still working on the rest of requirements (Data Requirements, User Requirements, Environmental Requirements and Usability Requirements), and I have to validate with my supervisor the project proposal document and current progress on requirement specification.

Now What?

I will meet with my supervisor and review current progress on requirements specification document.

Student Signature	Maximiliano Herrera

6.2.4. Reflective Journal, semester 2, February 2023

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: February 2023

What?

On February, I have worked on completing requirements specification document, including non-functional requirements. Additionally, I did research on ML .NET which is the main backend library to generate machine learning models on .NET C#. About the Frontend, I did research about Angular libraries for building drag and drop flow chart. I also have time to investigate what Cloud technologies I could use for my project.

So What?

After completing the requirements specification, and I did research about backend, Frontend and Cloud technologies. After that, I felt ready to start with the development phase and created two GitHub repository, one for the backend (.NET Web API) and the other for the front end (Angular). Both repository links were added at the beginning of the technical report on “Relevant Links” section.

As a part of my research on backend and frontend technologies, I successfully created two proofs of concepts (PoC). The PoC for backend was based on ML.NET, and it allowed me to generate ML models, evaluate the model and predict single values. I used the same data set on Jupyter notebook, and the model evaluation metrics between ML .NET and Jupyter notebook were very

similar. The second PoC was built for the frontend and allowed me to create a very basic drag and drop flow chart using the Angular library ng-flowchart. As challenges remain, I still have to choose the Cloud technologies where I would deploy my solution.
Now What? I will research more about available options on Cloud technologies, and will try to create a PoC so I can see how easy or difficult is to create a CI/CD pipeline for my solution.

Student Signature	Maximiliano Herrera

6.2.5. Reflective Journal, semester 2, March 2023

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: March 2023

What? In March, I continue working on the Cloud research, including AWS tutorials from explore.skillbuilder.aws. The research also covered an investigation on Continuous Integration, Continuous Delivery (CI/CD) technologies and platforms including CircleCI, Github Actions, AWS Code commit and so on so forth. Additionally, I discussed some of my concerns about GDPR with my supervisor after doing an investigation for another module.
So What? The research done on Cloud and CI/CD technologies and platform allows me to better understand the automation of deployments. Furthermore, the practice on deployments involving EC2s has improved my experience and I have a better understanding of AWS EC2 technology, which would be very beneficial for my project. As a remaining challenge to be solved, I still have to define where to host the database engine or if I should go for AWS RDS with SQL Server.
Now What? I will start creating the CI/CD pipeline and start deploying into AWS EC2. Following, I will add database access on my local and add a simple Entity framework database migrator. Once I have those things configured and working, I will be in a better position to define where to host the database engine.

Student Signature	Maximiliano Herrera

6.2.6. Reflective Journal, semester 2, April 2023

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: April 2023

What? In April, I have worked on the pipelines for Continuous Integration, Continuous Deployment (CI/CD) for the backend and frontend nodes. I was able to move my database from Azure to AWS RDS SQL Server as it was easier for the integration with the EC2 and provides a secure connection between both. Once I defined my database environment, I started working with Entity Framework as ORM and create some migration scripts. In addition, I was able to work on priority use cases such as create dataset and create workflow. For the rest of requirements, I created a dummy version of them to be showcase on the midpoint presentation.
So What? With database access in place, now is easier to make progress on the project. Also, I was able to store and retrieve files into database as well as storing and retrieving workflow graphs, which represents a very important milestone achieved. In relation to the UI, I was able to incorporate main technologies such as Angular Material and Kendo grids. Additionally, I implemented a basic version of workflow designer that allows user to create and save ML workflows. This represents another important milestone, and next developments will be based on current architecture. I still have pending to create the UI for workflow nodes or operators and implement the backend logic to execute machine learning workflows. Once I complete that part, I will be working on the endpoint for ML consumption. I consider those pending items as the most critical and challenging part of this project because it would involve working with complex code to make dynamic a static typed language as it is C#.
Now What? I will start working on the creation of the main workflow operators, such as dataset, clean, split and training. Once I have those implemented, I will implement the backend logic to execute a machine learning workflow which will generate a ML model. The good thing is that I already implemented a proof of concept for workflow execution, using reflection from C#, so now I have to "connect" it to the UI.

Student Signature	Maximiliano Herrera

6.2.7. Reflective Journal, semester 3, May 2023

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: May 2023

What?

In May, my focus was on working on two of the most critical use cases which are "FR_3: Save ML Workflow" and "FR_4: Generate ML model". In relation to ML model generation, I implemented some of the required workflow operators such as, Dataset, Split data, Clean Data, Normalize and Train model. Even though they are partially implemented, some of the operators allows a basic configuration such as dataset selection, columns selection and test fraction (Split data operator). I also added a basic architecture for validations when generating a ML model.

So What?

From the project progress point of view, now that I have a reasonable partial implementation of the high priority requirements (FR_1, FR_2, FR_3 and FR_4), I could start working on medium priority requirements such as "FR_5: Publish ML model" and "FR_6: Consume ML model". Also, I feel necessary to start with a low priority requirement "FR_7: Evaluate ML model" so that I can visualize how changes on the ML workflow affect the metrics of a given ML model.

On this sprint, I was able to achieve on of the most difficult challenges of my project that is to generate a ML model from a dataset or csv file. This is a very important milestone and will allow me to keep my project moving forward and within the schedule.

My pending challenges are to refine and improve the machine learning model generation, implement the publishing of ML models, consumption of ML models via endpoint and model evaluation.

Now What?

To address my pending challenges, I will improve the machine learning model generation by completing the workflow operators and implement those that are missing. In addition, I will start working on the requirement "FR_7: Evaluate ML model" to better understand how each operator on the workflow affects the metrics of a ML model. As a part of the requirement "FR_4: Generate ML model" I will implement a "preview" feature on each operator that will allow me to visualise the output for each transformation. That would be crucial for debugging purpose and will give a better understanding of the machine learning pipeline.

Student Signature	Maximiliano Herrera

6.2.8. Reflective Journal, semester 3, June 2023

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: June 2023

What?

The focus for this sprint was to have a basic but full functionality of the system, and that was achieved with the completion of following requirements FR_5: Publish ML model, FR_6: Consume ML model, and FR_7: Evaluate ML model.

From the UI point of view, the workflow designer was enhanced by improving the configuration of existing operators and adding a new operator called “Edit dataset”. Additionally, I added “NgRx Store”, that provides a state management for Angular in a similar way to Redux. This state management helps to manage the complexity from workflow designer and improves user experience. Lastly, new screens for endpoint configuration, testing and consumption were added.

On the backend, binary classification was added as a training model task, previously only linear regression was supported. The Endpoint to generate predictions was added, also a basic authentication module was incorporated for this endpoint using a simple token based, JWT from Auth0 or anonymous.

So What?

As commented before, a full system functionality was achieved on this sprint, and that give me time to work on enhancements, focus on testing and work on the completion of documentation and project deliverables.

The improvements on the UI, especially the addition of NgRx as state management was a great enhancement for the project. NgRx allows to handle the complexity of the workflow designer as a state machine, and that translates to a better user experience and make the maintenance easier.

The addition of binary classification as a training model task was a key milestone for my project because it exceeded my original expectations, which was implementing only price prediction using linear regression.

With latest improvements on the UI and backend, I was able to upload a dataset of 50Mb in around 20 seconds, run the ML workflow for the dataset under 10 seconds and generate predictions from endpoint with a response time under 300ms without any caching. That exceed completely my expectations and has superior performance compared with the original non-functional requirements from my documentation (section Performance).

What is pending is to improve the UI, make it more intuitive and easier to use. I would like to add some way to generate previews of intermediate data when processing a machine learning workflow. Improve the quality of software, add automated testing if possible and complete the documentation.

Now What?

To address outstanding challenges, I will start by improving the UI, adding missing screens if necessary and fixing issues that may arise. I will investigate how to generate intermediate data, such as the preview of dataset after being processed by an operator (for example, clean or normalization). As a part of intermediate data preview, I would like to add some statistics information for dataset columns, but this is out of the scope and may not be added. To improve the quality of software, I would like to implement on the backend the CQRS pattern (Command and Query Responsibility Segregation). This would bring some benefits such as improve scalability, performance, maintainability, testability and so on so forth. I will dedicate time to complete some of my automated tests, so that they can be added into the CI/CD pipeline.

Student Signature	Maximiliano Herrera

6.2.9. Reflective Journal, semester 3, July 2023

Student Name	Maximiliano Herrera
Student Number	20103212
Course	BSc (Honours) in Computing
Supervisor	Enda Stafford

Month: July 2023

What?

On the last sprint of my project, I have been working on the Project Closure, that means on its final phase. Focusing on activities such as project review, execution of test plan, bug fixing, enhancements, final deliveries, and the completion of the technical report.

So What?

The project review and execution of test plan was a fundamental step to ensure that the system can perform all its functionality, which is defined on the functional requirements section of technical report and is able to do it in compliance with the non-functional requirements. While executing the test plan, issues were raised, fixed and deployed into production environment.

On this sprint I was able to work on enhancements, that improved not only the user experience (UX) but also the usability and performance of the product. The three most important enhancements are:

- The ability of the system to handle and process large dataset files: The storage of files (datasets and intermediate data) was moved from database storage to AWS S3. This not only improved the performance of the system but also enabled to handle and process large files. With this in place, I was able to process a dataset of 150MB in around 40 seconds.
- Adding tracking for workflow execution: To have metrics for the execution of a workflow, I have added two new tables into database to persist execution details not only for the workflow but also for their operators. This information includes status details, start date, end date, total milliseconds, and operator details.
- Enhance the user experience when running workflows by adding web sockets (SignalR): By adding web sockets on the workflow designer screen, I was able to improve the communication to the user while running a workflow. To do that, the system sends notifications to the UI when relevant events happen on the system such as operator start, operator finish, or workflow finish. The UI can update accordingly and inform the user what is going on the system in real time. As secondary benefit, the addition of web sockets has reduced the complexity of the logic on the UI, now the UI follows an observer pattern rather than trying to process a request response and update multiple

components.

About challenges remain, I can see a major challenge and is about what to do with this product, I mean to look for potential customers and understand how this product could be useful for a company and adapt. Now the system can execute only two basic machine learning tasks: linear regression and binary classification. But from an enterprise perspective, this product would need to perform more complex machine learning tasks such as object detection, product recommendation, spike detection, sales prediction using time series, and so on so forth.

Now What?

To address outstanding challenges, I would need to get in contact with potential customers and get feedback from them, so that the product could adapt to the user needs. One of the ways to achieve this could be to present the product on my company and talk with the people that has expertise on machine learning so I can get a better picture of the potentiality of this product and its feasibility to solve real world problems.

Student Signature	Maximiliano Herrera