

Budapest University of Technology and Economics

Faculty of Electrical Engineering and Informatics

Department of Automation and Applied Informatics

Achref Mekni

**Graphical user interface to   
manage applying machine learning algorithms in prediction tasks**

SUPERVISOR

Mrad Mohammed Azzouz

" PhD Candidate"

BUDAPEST, 2023

Contents

[ABSTRACT 5](#_Toc151572848)

[ABSTRAIT 6](#_Toc151572849)

[1 Introduction 7](#_Toc151572850)

[2 Technologies 11](#_Toc151572851)

[2.1 Python Streamlit 11](#_Toc151572852)

[2.2 AWS EC2 12](#_Toc151572853)

[2.3 Keras & tensorflow 12](#_Toc151572854)

[2.4 Amazon S3 Bucket 13](#_Toc151572855)

[2.5 AWS Lambda 13](#_Toc151572856)

[2.6 Git & Gitlab 14](#_Toc151572857)

[2.7 Visual Studio Code 15](#_Toc151572858)

[2.8 Postman 16](#_Toc151572859)

[3 Literature 17](#_Toc151572860)

[3.1 Actors Identification 18](#_Toc151572861)

[3.2 Requirements Specification 18](#_Toc151572862)

[3.2.1 Functional Requirements 18](#_Toc151572863)

[3.2.2 Non-Functional Requirements 19](#_Toc151572864)

[3.3 Use Cases 19](#_Toc151572865)

[3.3.1 Global Use Case 19](#_Toc151572866)

[3.3.2 Virtual Machine Control 20](#_Toc151572867)

[3.3.3 Test Cases 21](#_Toc151572868)

[3.3.4 Checking Statistics 22](#_Toc151572869)

[3.3.5 Report Bug 23](#_Toc151572870)

[3.4 Rest API 23](#_Toc151572871)

[3.4.1 Test Case Management Service 24](#_Toc151572872)

[3.4.2 Branching & Module Management Service 24](#_Toc151572873)

[3.4.3 Virtual Machine Deployment Service 25](#_Toc151572874)

[3.4.4 Virtual Machine Parameters Management 25](#_Toc151572875)

[3.4.5 Reporting Management 26](#_Toc151572876)

[3.5 Database 26](#_Toc151572877)

[3.5.1 Reports Collection 27](#_Toc151572878)

[3.6 Application Architecture 28](#_Toc151572879)

[3.6.1 Back-End Layering 28](#_Toc151572880)

[3.6.2 Front-End Layering 29](#_Toc151572881)

[4 Implementation 31](#_Toc151572882)

[4.1 Front-End Overview 31](#_Toc151572883)

[4.1.1 Structure 31](#_Toc151572884)

[4.1.2 UI Layer 32](#_Toc151572885)

[4.1.3 Business Layer 38](#_Toc151572886)

[4.1.4 Service Layer 38](#_Toc151572887)

[4.2 Back-End Overview 40](#_Toc151572888)

[4.2.1 Structure 40](#_Toc151572889)

[4.2.2 Data Access Layer 40](#_Toc151572890)

[4.2.3 Business Layer 44](#_Toc151572891)

[4.2.4 Integration Layer 46](#_Toc151572892)

[5 Testing 48](#_Toc151572893)

[5.1 Reporting 48](#_Toc151572894)

[5.2 Branching & Module Management 50](#_Toc151572895)

[5.3 Test Case Management 51](#_Toc151572896)

[5.3.1 Gathering all the FT cases 52](#_Toc151572897)

[5.3.2 Gathering Executed FT cases 52](#_Toc151572898)

[5.3.3 Comparison 53](#_Toc151572899)

[5.4 Deployment Management 53](#_Toc151572900)

[6 DevOps 55](#_Toc151572901)

[7 Summary 56](#_Toc151572902)

[References 57](#_Toc151572903)

STUDENT DECLARATION

I, **Achref Mekni**, the undersigned, hereby declare that the present MSc thesis work has been prepared by myself and without any unauthorized help or assistance. Only the specified sources (references, tools, etc.) were used. All parts taken from other sources word by word, or after rephrasing but with identical meaning, were unambiguously identified with explicit reference to the sources utilized.

I authorize the Faculty of Electrical Engineering and Informatics of the Budapest University of Technology and Economics to publish the principal data of the thesis work (author's name, title, abstracts in English and in a second language, year of preparation, supervisor's name, etc.) in a searchable, public, electronic and online database and to publish the full text of the thesis work on the internal network of the university (this may include access by authenticated outside users). I declare that the submitted hardcopy of the thesis work and its electronic version are identical.

Full text of thesis works classified upon the decision of the Dean will be published after a period of three years.

Budapest, 10 December 2023

...…………………………………………….

Achref Mekni

ABSTRACT

ABSTRAIT

# Introduction

Artificial Intelligence (AI) has ushered in a new era of technological innovation, transforming industries across the spectrum. From enabling self-driving cars to enhancing natural language understanding, AI has rapidly evolved and diversified its applications. In this context, this thesis project is dedicated to the development and application of AI in the pharmaceutical industry, specifically focusing on image classification and prediction based on the viscosity and screw speed of pharmaceutical droplets.

The pharmaceutical industry plays a pivotal role in healthcare, and it continuously seeks advanced technologies to enhance drug development, manufacturing, and quality control processes. The efficient characterization and prediction of pharmaceutical droplets can have a profound impact on these processes. This thesis project is set to leverage the power of AI to address this need, offering a solution that encompasses artificial neural networks and convolutional neural networks (CNN & ANN) for image classification and predictive analysis. The central objective is to develop a cloud-based Graphical User Interface (GUI) that will empower users to interact seamlessly with AI models. This GUI will enable users to conduct testing, training, and application of AI models efficiently, providing a user-friendly and accessible platform.

The measurement of liquid viscosity of droplets in the pharmaceutical industry is a critical aspect of drug development and manufacturing. Viscosity is a key parameter that directly influences the behavior of pharmaceutical formulations, impacting factors such as drug release, stability, and manufacturing processes. Accurate viscosity measurements allow pharmaceutical researchers and manufacturers to ensure product consistency and quality. By understanding the viscosity of droplets, pharmaceutical professionals can make informed decisions regarding formulation design, process optimization, and quality control. Moreover, viscosity measurements play a vital role in predicting how droplets will interact and disperse within a given formulation, which is crucial for achieving desired drug delivery outcomes. In summary, the precise measurement of liquid viscosity of droplets is fundamental in ensuring the safety, efficacy, and quality of pharmaceutical products.

The objective is to develop a CNN-based model for accurately predicting the viscosity values of droplets of water-PVP solutions. By training the CNN model on a dataset of droplet images, the model will learn the patterns and features that correspond to different viscosity values. Once trained, the model will be able to classify new droplets based on their images and predict their viscosity values with high accuracy.

We will begin by analyzing the data, which consists of grouping several photos based on different ratios of water PVP solutions (000, 005, 010, 015). Then, we will move to the model architecture and training which involves selecting the number and types of layers, the number of neurons in each layer, and the activation function to be used. Afterward, we will test our model using a separate testing data set and also check if the model predicts the viscosity of the given test picture correctly. Then, we will show the results such as the Accuracy, MAE, MSE, and RMSE.

Moreover, Wet granulation is a fundamental process in the pharmaceutical industry for the formulation of solid oral dosage forms, such as tablets and capsules. This technique involves the mixing of active pharmaceutical ingredients (APIs) and excipients with a liquid binder to create granules, which are then dried and subsequently compressed into tablets or filled into capsules. Wet granulation offers several advantages, such as improving the flow properties, compressibility, and content uniformity of the blend. It also aids in controlling the release of APIs in the final dosage form.

For this, machine vision can be used as a tool for monitoring continuous twin-screw wet granulation, where the colored tracer would be dissolved in the granulation liquid alongside the API so their concentrations would simultaneously change.

The granule size can be measured using a high-speed imaging camera and image analysis, from which we can deduce the momentary liquid/solid (L/S) ratio

We will start with the data analysis where we have many images splitted into groups based on the feeding speed. Then we will apply an image processing on those pictures to extract several features of the granules which we will store in excel files. Those excel sheets are used later on to train our AI models which were trained to do the classification of the pictures based on the feeding speed (revolutions per minute). Three Artificial neural networks (ANN) algorithms were used which are MLPClassifier, Decision Tree Classifier and Kneighbors Classifier.

Subsequently, the project will explore the machine learning algorithms employed and the environment in which they will be applied. It is essential to elucidate how these algorithms will effectively process and analyze data relevant to pharmaceutical droplets, leading to accurate predictions that can aid decision-making within the industry. The collected data will undergo necessary preprocessing and visualization steps to ensure its suitability for AI model training and testing.

A core aspect of this project involves testing multiple methods for predictions, allowing for a comparative analysis of the results. This comparative approach will facilitate the selection of the most accurate and effective predictive models, ensuring their practical utility within the pharmaceutical industry.

Furthermore, to provide a user-friendly and accessible interface for industry professionals, a cloud-based user interface (web app) will be developed, with a particular emphasis on cloud solutions such as Amazon Web Services (AWS). This interface will serve as a central hub for seamless interaction with AI models, enabling users to test, train, and apply these models efficiently. The GUI will not only facilitate the process but also serve as a platform to represent and visualize the results.

We will employ Streamlit in Python to develop a user-friendly web application. Streamlit allows you to create an intuitive interface for users to seamlessly interact with your AI models, making it easier for researchers and pharmaceutical professionals to test, train, and apply AI models efficiently.

To efficiently handle the vast amount of image data related to pharmaceutical droplets, we will use AWS S3 buckets to store and retrieve these images. AWS S3's secure and scalable object storage ensures that your image data is readily available for model training and testing. This integration ensures that your project maintains a robust and reliable data storage and retrieval system, crucial for AI model performance.

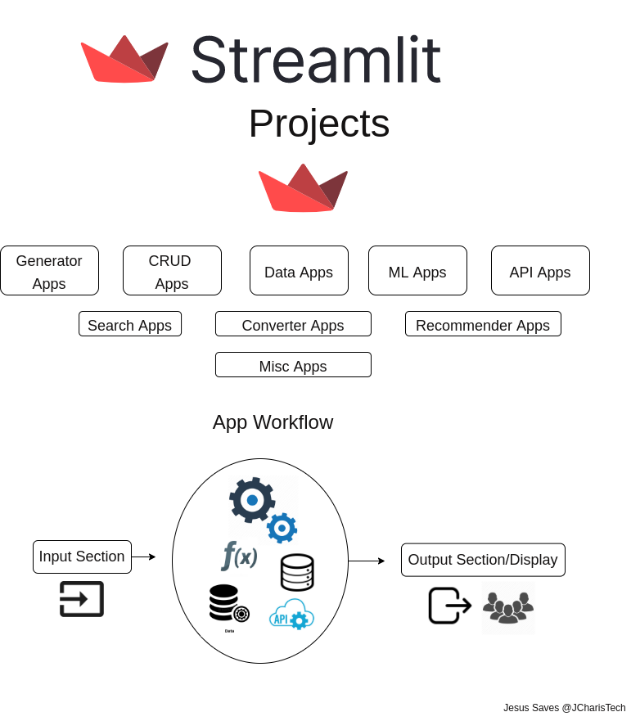
In addition, i will deploy the Streamlit-based web application on an AWS EC2 instance which can provide the reliability and scalability. By hosting the web app on an EC2 instance, we ensure that the AI models and the user interface are accessible to users 24/7, regardless of the user load.

In summary, the combination of Streamlit for the user interface, AWS S3 for data storage, and AWS EC2 for deployment enhances the effectiveness and accessibility of your AI-based pharmaceutical droplet prediction project, aligning with the project's goals of making AI technology accessible and practical for the pharmaceutical industry.

In conclusion, this project embarks on a journey to harness the power of AI for the pharmaceutical industry, specifically focusing on image classification and prediction related to pharmaceutical droplets. The development of AI models, integration of machine learning algorithms, and the creation of a user-friendly cloud-based GUI form the core components of this research, collectively aiming to provide a valuable tool for enhancing pharmaceutical processes. With the potential to revolutionize drug development and manufacturing, this project stands as a testament to the potential of AI in solving real-world challenges and driving innovation within the pharmaceutical sector.

# Technologies

## Python Streamlit



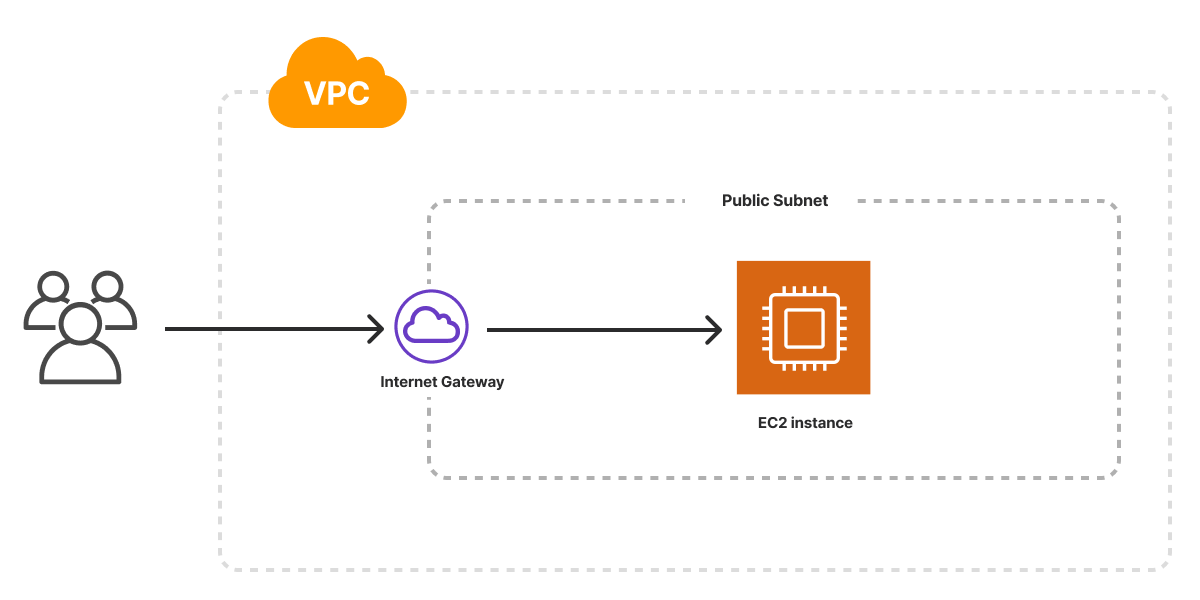
.. Figure: Streamlit

Streamlit is an open-source Python library used for creating web applications for data science and machine learning projects. It's designed to make it simple for data scientists, engineers, and researchers to build interactive and customizable web apps directly from Python scripts.

It has a lot of usefull features like interactive dashboards, visualizations, and data applications with relatively little code. It also has built-in widgets and components that make it easy to add features like sliders, buttons, text inputs, and charts to the web app.

The reason why I have chosen streamlit is because it allows developers to create interactive web applications with minimal code. Its simple and intuitive API allows users to focus on the ML model, visualizations, and data analysis, rather than spending time on complex web development. It also simplifies the deployment of ML models as web applications.

## AWS EC2



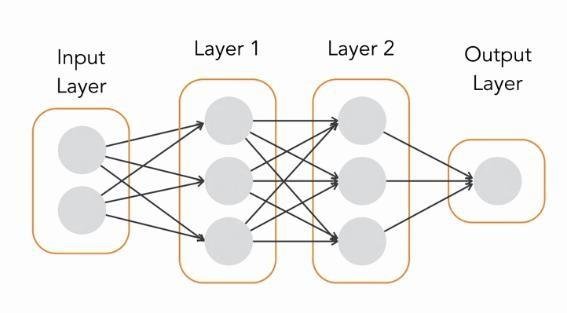
.. Figure: EC2 Architecture

Elastic Compute Cloud is a core and fundamental web service provided by Amazon Web Services (AWS). It enables users to rent virtual computing resources in the cloud, allowing them to run applications and workloads on a scalable infrastructure without the need to invest in physical hardware.

It allows me to create and manage virtual machines known as instances. These instances can be configured with different types of computing power, memory, and storage based on the user's requirements.

Moreover, it is secure because i have control over the security configurations of their EC2 instances. AWS provides various security features like security groups and key pairs to control network access and secure data.

## Keras & tensorflow



.. Figure: Sequential Keras model

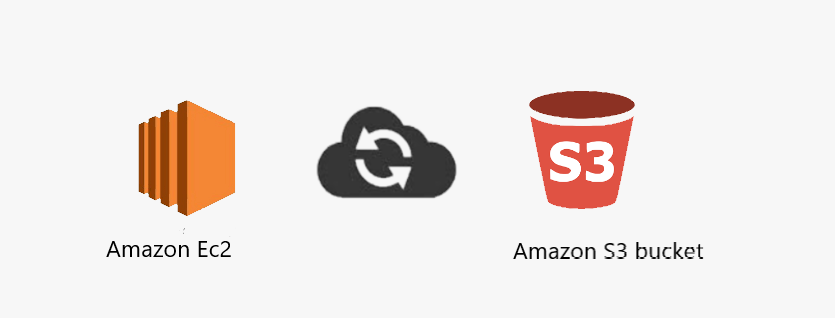
For building my sequential Machine Learning models i decided to use Keras and TensorFlow which are two powerful and popular libraries used in the field of deep learning and artificial intelligence, particularly in the development and implementation of neural networks.

Since TensorFlow 2.0, Keras has been integrated as the high-level API within TensorFlow, meaning that Keras can now be accessed directly from TensorFlow, providing a simpler and more user-friendly approach to building neural networks while maintaining the power and versatility of TensorFlow's core capabilities.

In Keras, the Sequential model is an easy way to build neural networks by stacking layers one after another.

Layers are added sequentially, with one following the other. Each layer is connected to the previous and the next layer in the stack.

## Amazon S3 Bucket



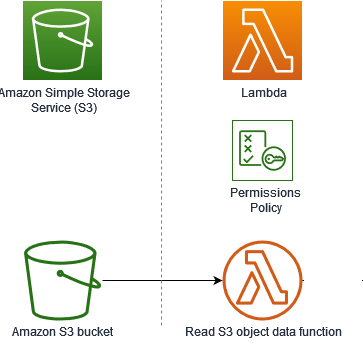
.. Figure: Amazon S3 bucket & EC2 relation

For storing the images that I will use to train the models I decided to use Amazon Simple Storage Service (Amazon S3) which is a scalable and highly durable object storage service offered by Amazon Web Services (AWS). It is designed to store and retrieve any amount of data from anywhere on the web. Amazon S3 is commonly used for data storage, backup, and archiving, as well as for hosting static websites, distributing large amounts of data, and serving as a reliable and cost-effective storage solution for various applications.

The reason why I am using it is because S3 seamlessly integrates with other AWS services, making it a central component in many cloud-based architectures. It can be used in conjunction with services like AWS Lambda.

It also offers a range of security features, including access control mechanisms using bucket policies and Access Control Lists (ACLs), server-side encryption, and integration with AWS Identity and Access Management (IAM) for fine-grained access control.

## AWS Lambda



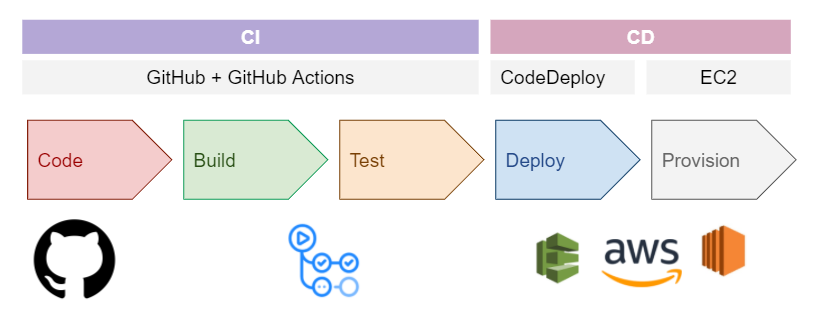
.. Figure: Usage of AWS Lambda with Amazon S3 Bucket

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). It allows you to run code without provisioning or managing servers, handling the infrastructure details, and automatically scaling based on the demand or the events triggering the function. With AWS Lambda, you only pay for the compute time consumed by your code.

Lambda functions automatically scale to handle varying workloads, providing high availability and performance.

It can be triggered by various AWS services and custom events, enabling you to build applications that respond to changes in data, execute code in response to HTTP requests, or perform other automated actions.

## Git & Github



.. Figure: Github for continuous integration

For the version control system, I decided to use Git & Gitlab since our company uses Gitlab.

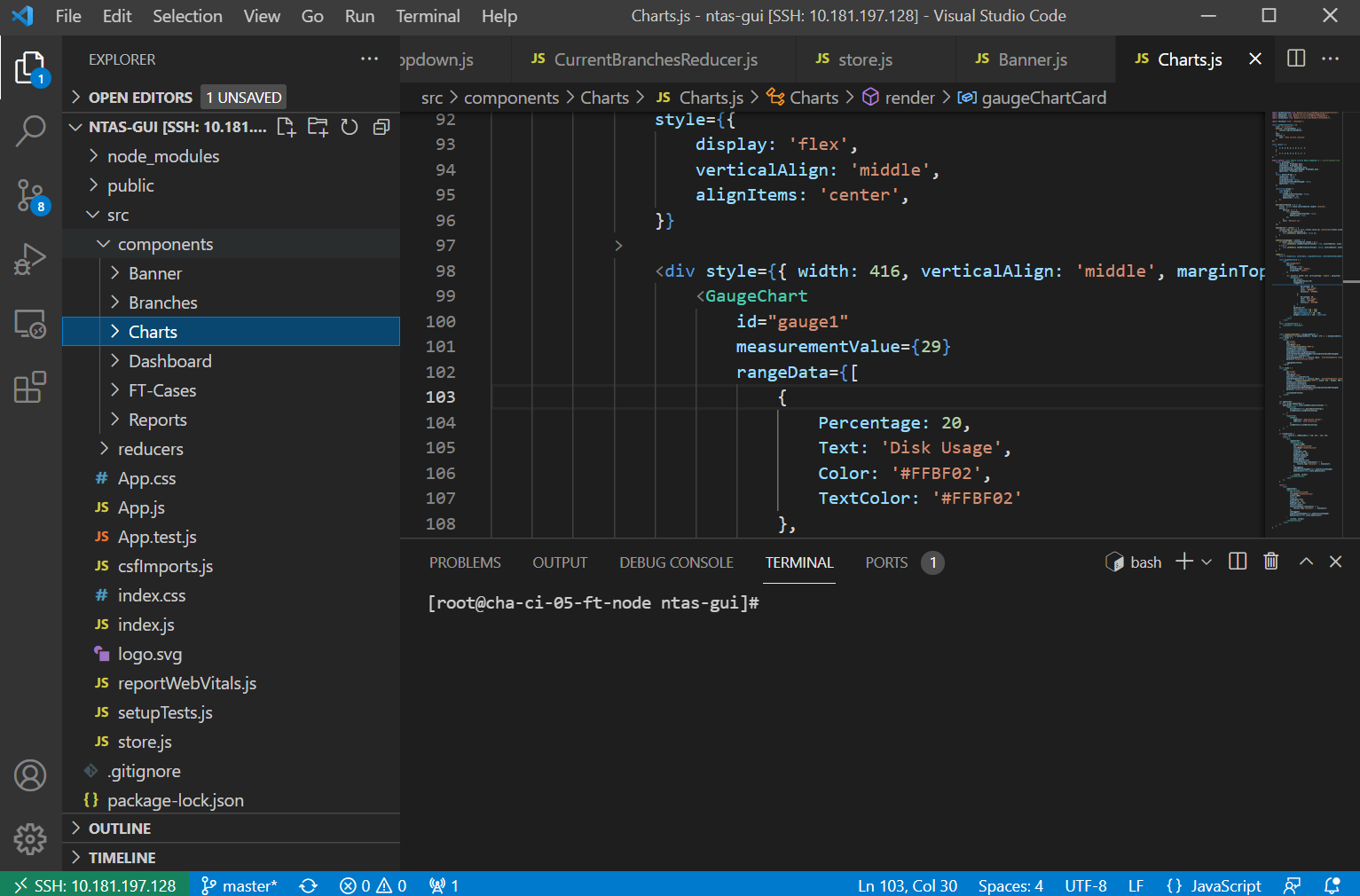
It is very secure since it uses SSH protocol to communicate with Git in a very secure way. we don’t need to use username and password each time since we are using SSH private & public keys to access the remote server of Gitlab.

Moreover, it helps in tracking the file changes and stores each update that you pushed so you will have a record of all actions done and that is why you can go back to specific versions in case you made a mistake in the most recent one.

It also supports CI/CD which helps us to continuously test, fix errors and improve the product.

It also makes the teamwork and collaboration simpler by allowing many changes coming from different sources to be merged into one source.

## Visual Studio Code



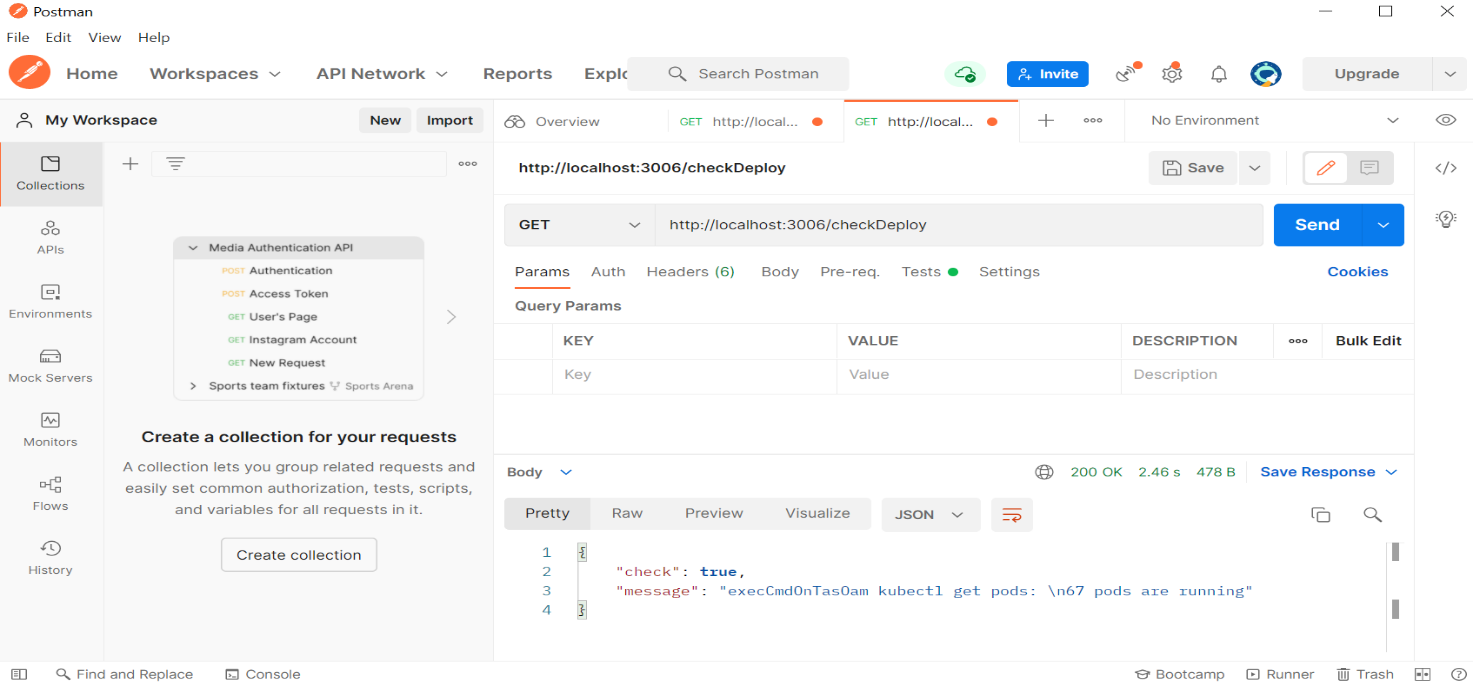
.. Figure: Visual Studio Interface

I decided to choose Visual Studio Code as an editor which is a lightweight but powerful source code editor. It is available on all the platforms. It supports multiple programming languages like TypeScript, JavaScript, Python, C# and it also contains a lot of extensions that provides further helpful features in terms of coding and extending the supported languages and runtimes like .NET and Unity.

Since the Project is huge so it contains multiple files and components. Visual Studio Code helped me to manage components by separating them in different folders. It helped to keep a clear structure of the project in terms of organizing all the included items.

Moreover, the project is on a virtual machine and visual studio supports SSH connection to a virtual machine and allows us to access its content.

## Postman



‎2.10. Figure: Postman Interface

For Testing, I have chosen Postman as a tool which allows me to test my API and its functionalities.

It is a client used to test HTTP requests and analyze responses to see if our API is working efficiently and correctly or not.

Postman reduces the amount of code written to test a single functionality by just making simple requests, so it eases the testing job.

Moreover, it allowed me to test the speed performance of my HTTP requests by writing simple tests to check weather a HTTP request exceeded a certain amount of time.

# Literature

The pharmaceutical sector places paramount importance on liquid viscosity measurement, utilizing tools such as capillary viscometers, orifice viscometers, rotational viscometers, vibrational viscometers, and ultrasonic viscometers. Viscosity, a key fluid property, profoundly influences processes critical to pharmaceutical manufacturing. Accurate viscosity determination is essential for achieving precise control and optimization in the production of pharmaceutical droplets.

The assessment of liquid viscosity holds significant importance in various industries. Various instruments, including capillary viscometers, orifice viscometers, rotational viscometers, vibrational viscometers, and ultrasonic viscometers, can be employed for this purpose. Among these, the capillary viscometer stands out as the most widely utilized due to its cost-effectiveness and simplicity. This method involves measuring the time required for a defined volume of liquid to traverse a narrow-bore tube under specific pressure conditions [7]. Despite their prevalence, these invasive methods are associated with high costs, rendering them unsuitable for continuous viscosity measurement conducive to in-process monitoring and timely interventions in case of errors. Consequently, there is a pressing need for a cost-effective and time-efficient alternative.

One promising approach for viscosity measurement involves the estimation of viscosity using machine learning and image processing algorithms based on liquid droplet characteristics. Numerous researchers have explored correlations between liquid viscosity and droplet properties. For instance, H. Zhu et al. established a correlation between extensional viscosity and spray droplet sizes in polymer spray solutions [8]. Gotaas et al. investigated the impact of viscosity on droplet-droplet collision outcomes [8]. Wang et al. demonstrated a logarithmic increase in droplet diameter in vertical gas-liquid annular flows with rising liquid viscosity [9].

In addition to these correlations, researchers have explored the application of image processing techniques for liquid viscosity measurement. Kheloufi et al., for instance, measured the fall height of a ball in falling ball viscometers by capturing video scenes during its descent and utilizing them to compute viscosity [10]. Santhosh et al. successfully estimated viscosity by capturing refracted images of a laser through a liquid-containing tube using a camera. These images were processed using thresholding, filtering, and histogram techniques, and an artificial neural network model was employed to establish the relationship between the resultant data and viscosity [11].

In addition to viscosity, the thesis explores machine vision-based methods for real-time quantification of ultralow drug content during continuous twin-screw wet/dry granulation and tableting. This extends the scope of AI application in pharmaceutical processes, aiming for enhanced accuracy and efficiency.

Solid oral dosage forms, such as tablets and capsules, can undergo various continuous processing methods like direct compression or dry granulation (Jarvinen et al., 2013; Mangal et al., 2016). Among these methods, wet granulation remains a widely adopted approach (Bandari et al., 2020). Its popularity lies in its capability to produce agglomerated particles or granules with superior downstream properties, particularly beneficial for subsequent processes like tableting. Enhanced critical material attributes, including flowability, bulk density, and compactibility, contribute to the consistent production of high-quality tablets (van den Ban and Goodwin, 2017). In the context of low-dose drug products (where the Active Pharmaceutical Ingredient <API> content is below 2% w/w), additional challenges, such as meeting content uniformity requirements, necessitate careful consideration in selecting the appropriate manufacturing technology (Almaya, 2009).

The advantages of continuous processing have led to a surge in studies and publications on Twin-Screw Wet Granulation (TSWG) (Keleb et al., 2004; Ito and Kleinebudde, 2019; Dhenge et al., 2010; Meier et al., 2015; Arndt et al., 2018; Osorio et al., 2017; AlAlaween et al., 2020). These studies predominantly focus on evaluating the impact of specific process parameters, such as Liquid-to-Solid ratio (L/S) (Nicolaï et al., 2018), screw configuration (Djuric and Kleinebudde, 2008; Li et al., 2014), barrel fill level (Lute et al., 2018), and overall process settings (Portier et al., 2020; Portier et al., 2020). Beyond these technological considerations, the distribution of granulating liquid's significance (El Hagrasy et al., 2013; Dhenge et al., 2012) and the exploration of implementing Process Analytical Technology (PAT) tools coupled with real-time process control have also been subjects of investigation (Madarasz et al., 2018; Harting and Kleinebudde, 2018).

The study aims to develop a system which could indirectly measure the API content in concentrations below 0.1 w/w% with the addition of a colored excipient. For this, machine vision can be used as a tool for monitoring continuous twin-screw wet granulation, where the colored tracer would be dissolved in the granulation liquid alongside the API so their concentrations would simultaneously change.

In recent years, the integration of Artificial Intelligence (AI) and machine learning (ML) into pharmaceutical processes has emerged as a transformative force, offering unprecedented insights and efficiencies. We will focus on the pivotal role of cloud-based user interfaces in hosting machine learning models within the pharmaceutical industry. The overarching goal is to explore how these interfaces facilitate seamless interaction, testing, training, and application of AI models, specifically tailored for tasks related to pharmaceutical droplets and granules.

# Requirements

## Actors Identification

Our application will only be used internally by me (user) and externally by the people interested in using a cloud-based machine learning tool in the pharmaceutical industry in order to facilitate the models trainings, testing and data analysis

## Requirements Specification

In order to have a good functional cloud-based machine learning web platform that could serve the users, there are number of elements qualified as requirements for the project.

### Functional Requirements

* **AI Model Preparation:**

Develop Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) models tailored for pharmaceutical droplets.

* **Viscosity Classification and Analysis:**

Train our AI models to do the classification of pictures based on the liquid viscosity.

* **Screw Speed classification and Analysis:**

Train our AI models to do the classification of the pictures based on the screw speed (revolutions per minute).

* **Data Study and Analysis:**

Collect relevant data for training and testing the AI models.

Perform thorough data analysis, including preprocessing and visualization, to enhance the quality of input data for the models.

* **Cloud-Based User Interface (Web App) Development:**

Design and develop a user-friendly cloud-based Graphical User Interface (GUI) hosted on a platform like AWS.

* **Results Representation:**

Implement a mechanism on the user interface to represent and display all relevant results generated by the AI models.

* **Testing of Models and Interface Functionalities:**

Conduct rigorous testing of the developed AI models to ensure their reliability.

Test the functionalities of the cloud-based user interface, including its responsiveness, ease of use, and overall performance.

### Non-Functional Requirements

In addition to functional requirements, the application also needs to satisfy these requirements to improve the quality of its functionality, such as:

* **Security:**

Data Protection: Ensure the security and confidentiality of pharmaceutical data, complying with relevant regulations and standards.

User Authentication and Authorization: Implement secure login mechanisms and access controls to protect against unauthorized use.

* **Performance:**

Response Time: The system should provide timely responses to user inputs, ensuring a responsive and efficient experience.

* **Scalability:** The solution should handle varying workloads and adapt to changes in user numbers or dataset sizes without compromising performance.
* **Simplicity**: Our Application should be simple to use
* **User Experience:** User experience (UX) focuses on having a deep understanding of users, what they need, what they value, their abilities, and also their limitations since the goal of our application is to ensure easing the job for the users.

# Design

## Use Cases

### Global Use Case

The detailed study of the specifications allows us to identify several use cases. The use case diagram visually represents the interactions between different actors and the system under consideration. In figure 3.1 below, we represent the main use cases to have a global view of the functioning of our application, as well as the possible interactions that can take place. The tasks of the actor (user) include controlling the training the model, testing the model, creating data frames, managing pictures storage and checking results and visualizations.

A diagram of a person's mind

Description automatically generated

.. Figure: Global Use Case Diagram

### Training machine learning models

**Generate Data frames**

**In the context of machine learning, a data frame is a two-dimensional tabular structure that organizes data into rows and columns. Each row typically represents an observation or data point, and each column represents a feature or variable. The data frame serves as the input to the machine learning model, providing a structured format for training and testing.**

**The data frames will be generated through a systematic process that involves collecting and aggregating relevant data from various sources which includes information about droplet characteristics (area, perimeter, color intensity…). The data frame generation process often includes data preprocessing steps such as cleaning, normalization, and encoding to ensure the data is suitable for machine learning algorithms.**

**The user will be able to generate the data frames and store them into excel files that can be downloaded or stored in the cloud.**

**Data Retrieving**

The user will be able to retrieve his data from the AWS S3 bucket using certain lambda functions by interacting with the Amazon Simple Storage Service (S3) API.

**Model Feeding**

The user will be able to feed the model with the necessary pictures and for training the convolutional neural network (CNN) model and the necessary data frame for training the artificial neural network (ANN) model.

**Algorithm Selection**

The user Select a variety of machine learning algorithms. Common algorithms include convolutional neural network algorithm, MLP classifiers, k-nearest neighbors and decision tree classifiers.

### Testing machine learning models

The user can apply each trained model to the testing set and obtain predictions. This step assesses how well the models generalize to unseen data.

**Selecting Metrics:**

The user will choose the appropriate evaluation metrics. Common metrics for classification tasks include accuracy, precision, recall.

**Calculating Metrics:**

The user will evaluate the performance of each model using the selected metrics. This step provides a quantitative measure of how well the models are performing.

### Visualize Results

The application provides some statistics and data collection to the user which will allow him to have an overview on the data correlation, the predicted values versus the actual ones, graphs to show how the model looks like from inside in terms of layers.

The Testing Error measurements and Prediction accuracy values will be also measured and displayed.

### Store Data to S3 buckets

The user will be able to store pictures, excel files and models to S3 buckets using the web interface or manually within the AWS console.

handling different file formats appropriately should be ensured. For example, when working with Excel files, specific libraries like pandas for data manipulation might be used.

Appropriate permissions will be set for the uploaded objects. Configuring access control lists (ACLs) or bucket policies may be needed based on the security requirements.

### Download & Load Model

The user will be able to download the models locally for personal use or load them into the application so it can be tested.

### Authentication

Authentication is a process that verifies the identity of a user, system, or entity attempting to access a particular resource or service. It is a fundamental aspect of information security and is crucial for ensuring that only authorized individuals or systems can gain access to protected resources. Authentication is typically achieved through the use of credentials, such as usernames and passwords, tokens, or digital certificates.

The user should be able to login using the provided credentials which gives him total administrative control over the application.

### Report Bug

The application allows the user to report errors or bugs that he faced when using the platform in order for us to figure out what was wrong with a certain feature so it can be fixed for the sake of providing a better user experience and a better performance.

The user is able to write a description about the problem in a textual format by providing related details to allow us to have a better understanding on what occurred.

He can also include pictures to provide us with a visual aspect of the bug.

Moreover, the user should include his professional email and his full name and the severity of the bug by selecting one of the options in the dropdown list.

The email, full name, severity and description are required fields in order to be able to submit the reporting form. The pictures are optional.

## Machine Learning Part Design

A pie chart with numbers and symbols

Description automatically generated

### Viscosity Model:

1. Define the Objective:

The objective is to develop a CNN-based model for accurately predicting the viscosity values of droplets of water-PVP solutions. By training the CNN model on a dataset of droplet images, the model will learn the patterns and features that correspond to different viscosity values. Once trained, the model will be able to classify new droplets based on their images and predict their viscosity values with high accuracy.

2. Collect and Prepare Data:

Our data is a collection of pictures that are split into 11 classes based on different ratios of water PVP solutions. Each class has 3 subclasses called AfterDetachment, AtDtachment & BeforeDetachment. For training & testing I have chosen to use the BeforeDetachment pictures in each category.

The viscosity of the various samples was determined using an Anton Paar DMA 4500 M viscometer. Through the samples in the capillary tube, we calculated the rolling time of a steel ball with a 1.5 mm diameter. The capillary's angle was 45 degrees, and the measurement's temperature was 25 C.

3. Define Metrics for Success:

The metrics that will be measured are the prediction accuracy, MAE, MSE, RMSE and R2 score

4. Split the Data:

* Training data size: 3784 images.
* Validation data size: 946 images.
* Testing data size: 1271 images.
* Viscosities: the data is collected in an Excel spreadsheet that looks like this in the picture down below

A table with numbers and a number of objects

Description automatically generated with medium confidence

5. Select Algorithm:

The chosen algorithm Convolutional Neural Network (ConvNet/CNN) which is a Deep Learning algorithm that can take images as input and identify different objects in the image by giving them learnable weights and biases. ConvNet needs a lot less preprocessing than other algorithms for classification.

6. Train the Model:

Train the selected model using the training dataset. Adjust hyperparameters as needed. Monitor the training process and use validation data to prevent overfitting.

7. Evaluate Model Performance:

Assess the model's performance on the test set using the predefined metrics. Analyze the results and iterate on the model or data preprocessing if needed.

8. Deployment:

Prepare the model for deployment. This involves converting the trained model into a format suitable for deployment, and integrating it into the web app.

### Twin-screw Model

1. Define the Problem:

The objective is to develop machine learning models for image classification based on the screw speed. Artificial Neural networks algorithms will be applied and also Convolutional Neural network algorithm will be applied and then we will compare the results.

2. Collect and Prepare Data:

Our data is a collection of pictures which are splitted to 6 classes based on the screw speed (revolutions per minute):

5.7 rpm: 2796 pictures

5.7 rpm vissza: 3193 pictures

6.1 rpm: 2908 pictures

6.1 rpm vissza: 2876 pictures

6.5 rpm: 3011 pictures

The pictures contain a dark background were on top there are some captured dried granules that has different sizes. Those granules were obtained after applying wet granulation with basic screw speeds: 5.7, 6.1, 6.5 rpm.

3. Define Metrics for Success:

The metric that will be measured is the prediction accuracy.

4. Split the Data:

we split our data to training and validation data (25% of the pictures will be used as validation data). For testing, 6007 pictures will be used.

5. Select a Model:

Three Artificial neural networks (ANN) algorithms were used which are MLPClassifier, Decision Tree Classifier and Kneighbors Classifier. Moreover, we will also apply a convolutional neural network (CNN) algorithm which is designed specifically to process grid-like structure data like an image.

6. Train the Model:

Train the chosen model on the training data. During training, the model learns patterns and relationships within the data. This involves adjusting the model's parameters to minimize a loss function, which quantifies the difference between predicted and actual values.

7. Evaluate Model Performance:

Evaluate the model's performance on the test set to assess its generalization to new, unseen data. Common evaluation metrics include accuracy, precision, recall, F1 score, and others, depending on the problem.

8. Deployment:

Once satisfied with the model's performance, we will deploy it to our web application where it can be trained, tested and make predictions on new, real-world data.

### Class Diagram

A computer screen shot of a computer program

Description automatically generated with medium confidence

## Web Application Part Design

My web application is a python streamlit application which is deployed on AWS cloud using services like EC2, lambda, identity and Access Management and S3 bucket.

### Application Architecture

A diagram of a computer chip

Description automatically generated

.. Figure: Database Structure

**User Interface (UI) Layer:**

Streamlit Script: The core of a Streamlit application is a Python script. This script contains the code for defining the UI components and specifying the app's behavior. Streamlit scripts are typically written in a top-down, procedural style.

Widgets and Components: Streamlit provides a range of widgets (e.g., sliders, buttons, text inputs) that you can use to interact with your application. These widgets are used to gather user input and trigger actions in response.

**Application Logic Layer:**

Data Processing: it includes data loading, preprocessing, splitting images, creating dataframes and excel files. This layer grants access to certain data from a data source like AWS S3 bucket with which we can apply CRUD operations like Creating, Reading, Updating and Deleting data.

Business Logic: Streamlit scripts often include business logic for handling user input, making decisions, and triggering actions. This logic is usually embedded within the script and executed in response to user interactions. Moreover, The API will handle the data retrieving and uploading from and to AWS S3 bucket.

**Visualization Layer:**

Charts and Visualizations: Streamlit allows you to integrate data visualizations seamlessly. Libraries like Matplotlib and Plotly will be used to create charts and plots. The visualizations are embedded within the Streamlit script and are updated dynamically based on user input.

**Integration with External Services:**

Data Sources: Streamlit applications can interact with external services, databases, or APIs such as AWS S3 bucket, AWS lambda service. Data can be loaded from these sources within the application logic layer.

Machine Learning Models: trained model can be integrated for predictions or classifications. Streamlit can be used to showcase model results and predictions.

**Deployment Layer:**

Deployment Platform: Our Streamlit application will be deployed on AWS using the EC2 service alongside other services like lambda, s3 bucket and identity and Access Management. The deployment process is simplified, and Streamlit apps can often be deployed with minimal effort.

### Data Management

In our application we will be using a lot of pictures and excels containing the data needed for training. Which means that we need to find a way to store these data somewhere instead of uploading them manually each time we want to train a model. For this purpose, I have decided to use AWS S3 Bucket alongside AWS Lambda to manage the data including both data upload and retrieval based on user input.

A white paper with black text

Description automatically generated

**1. Data Upload (User Input):**

* Create an S3 Bucket:

Begin by creating an S3 bucket to store the data. This can be done through the AWS Management Console or programmatically using the AWS CLI or SDKs.

* Configure Bucket Policy:

Set up a bucket policy or Access Control Lists (ACLs) to control who can upload data to the bucket. This ensures that only authorized users can write to the bucket.

* User Uploads Data:

When a user uploads data, a Lambda function is invoked to handle the upload process. The user input is passed as an event parameter to the Lambda function.

Users can also upload data to the S3 bucket either through the AWS Management Console.

**2. Data Retrieval (User Input):**

* User Requests Data:

Users request specific data based on their input. This input will be passed through the user interface and API call.

* Invoke Lambda Function for Retrieval:

When a user requests data, a Lambda function is triggered to manage the uploading process. The input provided by the user is transferred as an event parameter to the Lambda function.

* Lambda Execution Role:

Ensure that the Lambda function for retrieval has the necessary permissions (IAM role) to read from the S3 bucket.

* User Receives Data:

The Lambda function retrieves the requested data from the S3 bucket based on the user input and returns it to the user, through the API.

### API

# Implementation

## Viscosity Machine Learning Model

### Structure

A screenshot of a computer

Description automatically generated

Each part of the web application is in a file that is packed in a separate folder to make dealing with changes in the future much easier.

The code is devided into classes that contains the necessary functions for data processing, model training & testing, metrics calculation and results visualization.

As I mentioned previously, for building my sequential Machine Learning models i decided to use Keras and TensorFlow.

For the web application, i used python streamlit library.

The Homepage.py file is the main entry point of the application.

### User Interface/Main Components

We can see that this layer is composed of several components which will outline the view

#### SideBar



A screenshot of a computer

Description automatically generated

‎4.2. Figure: Navigation SideBar Design

The side drawer as seen on Figure 4.2 contains all the necessary tabs for the user to be able to navigate to different pages of the web platform. Whenever the user clicks on a certain tab, the script related to that page will be loaded.

#### Twin Screw Page

The Twin Screw is a page that handles the upload of the dataset (excel, csv file), the pandas dataframe generation, the profiling of the data inside the dataframe, the classifiers creation then the option of downloading them.

It contains radio buttons that will lead us to the sub-pages:

* Upload
* Profiling
* Modelling
* Download

A file\_uploader component will handle the uploading of the dataset then a Dataframe will be created and shown underneath it.

A screenshot of a data sheet

Description automatically generated

After Clicking on profiling a whole report will be generated from that data using this function:

from streamlit\_pandas\_profiling import st\_profile\_report

streamlit\_pandas\_profiling library is used to generate and display a Pandas Profiling report within a Streamlit app. This is particularly useful for quickly exploring and visualizing the characteristics of a DataFrame.

The st\_profile\_report function takes a Pandas DataFrame (data) as an argument and generates a Pandas Profiling report for that DataFrame. This report includes various statistics, visualizations, and insights about the data, such as summary statistics, distribution of values, missing values, correlations, and more.

Some examples are shown below. One of them shows the correlation matrix which is a visual representation of the correlation between numeric variables. Each cell in the table represents the correlation between two variables. The values range from -1 to 1.

Correlation matrices help in identifying patterns and relationships between variables.

In machine learning, high correlations between features (independent variables) may indicate multicollinearity. Multicollinearity can impact the performance of certain algorithms, and feature selection may be required to remove redundant features.

Identifying strong correlations can highlight potential errors in the data. For instance, if two supposedly independent variables show a high correlation, it might indicate an error in data collection or preprocessing which will help me to clean the data and drop certain unnecessary columns.

A screenshot of a computer

Description automatically generated A screenshot of a data analysis

Description automatically generated

# Summary

References

1. [Indirect monitoring of ultralow dose API content in continuous wet granulation and tableting by machine vision - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0378517321008140)
2. [Integrated twin-screw wet granulation, continuous vibrational fluid drying and milling: A fully continuous powder to granule line - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S037851732031111X)
3. What is Git & why should you use it? (2021.november) <https://www.nobledesktop.com/blog/what-is-git-and-why-should-you-use-it>
4. Why did we build Visual Studio Code? (2021.november) <https://code.visualstudio.com/docs/editor/whyvscode>
5. [Processes | Free Full-Text | Classification of Droplets of Water-PVP Solutions with Different Viscosity Values Using Artificial Neural Networks (mdpi.com)](https://www.mdpi.com/2227-9717/10/9/1780)
6. [Welcome to AWS Documentation (amazon.com)](https://docs.aws.amazon.com/)
7. Brooks, R.; Dinsdale, A.; Quested, P. The measurement of viscosity of alloys—A review of methods, data and models. Meas. Sci. Technol. 2005, 16, 354. [The measurement of viscosity of alloys—a review of methods, data and models - IOPscience](https://iopscience.iop.org/article/10.1088/0957-0233/16/2/005)
8. Zhu, H.; Dexter, R.; Fox, R.; Reichard, D.; Brazee, R.; Ozkan, H. Effects of polymer composition and viscosity on droplet size of recirculated spray solutions. J. Agric. Eng. Res. 1997, 67, 35–45. [Effects of Polymer Composition and Viscosity on Droplet Size of Recirculated Spray Solutions - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0021863497901517?via%3Dihub)
9. Gotaas, C.; Havelka, P.; Jakobsen, H.A.; Svendsen, H.F.; Hase, M.; Roth, N.; Weigand, B. Effect of viscosity on droplet-droplet collision outcome: Experimental study and numerical comparison. Phys. Fluids 2007, 19, 102106. [Effect of viscosity on droplet-droplet collision outcome: Experimental study and numerical comparison | Physics of Fluids | AIP Publishing](https://pubs.aip.org/aip/pof/article-abstract/19/10/102106/938495/Effect-of-viscosity-on-droplet-droplet-collision?redirectedFrom=fulltext)
10. Wang, Z.; Liu, H.; Zhang, Z.; Sun, B.; Zhang, J.; Lou, W. Research on the effects of liquid viscosity on droplet size in vertical gas–liquid annular flows. Chem. Eng. Sci. 2020, 220, 115621. [Research on the effects of liquid viscosity on droplet size in vertical gas–liquid annular flows - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0009250920301536?via%3Dihub)
11. Kheloufi, N.; Lounis, M. An Optical Technique for Newtonian Fluid Viscosity Measurement Using Multiparameter Analysis. Appl. Rheol. 2014, 24, 15–22. 11.
12. Santhosh, K.; Shenoy, V. Analysis of liquid viscosity by image processing techniques. Indian J. Sci. Technol. 2016, 9, 98693. [Analysis of Liquid Viscosity by Image Processing Techniques (indjst.org)](https://indjst.org/articles/analysis-of-liquid-viscosity-by-image-processing-techniques)