ICT304 Tutorial 1

Contents

[*1) Differences and Similarities Between Artificial Intelligence, Machine Learning, and Deep Learning* 1](#_Toc177137932)

[*2) Generative AI and Large Language Models (LLMs)* 2](#_Toc177137933)

[*3) Advantages of Fuzzy Logic with Real-World Examples* 3](#_Toc177137934)

[*4) Systems Engineering and Project Life Cycle Phases* 4](#_Toc177137935)

[*5) AI System for Classifying Rice Grain Species* 5](#_Toc177137936)

[*5.1) Selection of AI Techniques* 5](#_Toc177137937)

[*5.2) Justification for SVM* 5](#_Toc177137938)

[*5.3) Data Preprocessing Required* 5](#_Toc177137939)

[*5.4) Model Accuracy, Validation Method, and Performance Metrics* 6](#_Toc177137940)

[*5.5) User Guide* 6](#_Toc177137941)

[*5.6) Potential Model Improvements* 7](#_Toc177137942)

[*6) Selection and Setup of Software Tools for the Project* 8](#_Toc177137943)

[*6.1) Project Management Tool: Trello or ClickUp* 8](#_Toc177137944)

[*6.2) Version Control Tool: Github* 9](#_Toc177137945)

[*6.3) Collaboration Tool: Google Drive* 10](#_Toc177137946)

[*6.4) Team Communication Tool: Discord* 10](#_Toc177137947)

## *1) Differences and Similarities Between Artificial Intelligence, Machine Learning, and Deep Learning*

**Artificial Intelligence (AI)** seeks to develop machines replicating human cognitive functions, such as reasoning and problem-solving. A notable example is a virtual assistant like Siri, which can interpret and respond to spoken commands.

**Machine Learning (ML)** is a branch of AI that empowers computers to learn from data autonomously without explicit programming. For instance, a spam filter learns to identify spam emails by analysing patterns from previous messages.

**Deep Learning (DL)** is a specialised form of ML that employs artificial neural networks to recognise intricate patterns. It is commonly utilised in applications like image recognition, such as enabling a self-driving car to spot pedestrians.

**Similarities:**

* All three fields aim to develop intelligent systems for making decisions.
* Both ML and DL serve as techniques to realise AI.

**Differences:**

* AI is the overarching concept, ML is a method for achieving AI, while DL is specific techniques within ML that utilises neural networks.

## *2) Generative AI and Large Language Models (LLMs)*

**Generative AI** refers to AI models that produce new content, such as text, images, or music, by analysing patterns in existing data. Examples include AI applications that can create artwork or compose stories.

**Large Language Models (LLMs)** are a specific type of generative AI designed to comprehend and generate human-like text. They are trained on extensive text datasets to produce coherent and contextually relevant responses, such as those generated by ChatGPT or GPT-4.

**Relationship:**

* LLMs represent a subset of generative AI that concentrates on text creation.
* Both leverage patterns from existing data to generate new, human-like content.

## *3) Advantages of Fuzzy Logic with Real-World Examples*

**Fuzzy Logic** addresses uncertainty and imprecision, making it particularly effective in complex, real-world scenarios where traditional binary (true/false) logic is inadequate.

**Advantages:**

* Facilitates decision-making even with incomplete or uncertain information
* Offers reasoning that resembles human thought, accommodating “grey areas”.

**Examples:**

* **Climate Control Systems:** Adjusts heating and cooling based on terms like “slightly cold” or “very warm” instead of fixed temperature settings.
* **Anti-locking Braking Systems (ABS):** Utilises fuzzy logic to apply braking force smoothly across varying road conditions.
* **Washing Machines:** Determines wash cycles based on fuzzy descriptors such as “lightly soiled” or “heavily soiled”.

## *4) Systems Engineering and Project Life Cycle Phases*

**Systems Engineering** is an interdisciplinary methodology focused on designing, integrating, and managing complex systems throughout their life cycles.

**Project life Cycle Phases:**

* **Concept Development:** Establish project objectives and requirements. (*Example: Planning a new public transportation system to alleviate traffic congestion.*)
* **System Design:** Develop detailed designs and models. (*Example: Creating routes, stops, and schedules for the new transportation network.*)
* **Implementation:** Construct and integrate system components. (*Example: Building bus stops, acquiring buses, and establishing necessary infrastructure.*)
* **Testing and Validation:** Verify that the system meets specified requirements (*Example: Conducting pilot tests to evaluate efficiency and safety.*)
* **Deployment:** Launch the system for public access. (*Example: Opening the new transportation system to commuters.*)
* **Operation and Maintenance:** Oversee and sustain system performance. (*Example: Performing regular maintenance on buses and continuously monitoring service quality.*)

## *5) AI System for Classifying Rice Grain Species*

### *5.1) Selection of AI Techniques*

Support Vector Machine (SVM) will be used.

### *5.2) Justification for SVM*

SVM is suitable for this task because:

* **High Dimensionality:** SVM works well with high-dimensional data and is effective for small to medium-sized datasets.
* **Robust to Overfitting:** When using regularisation parameters, SVM helps avoid overfitting, which is essential for real-world data with some noise or variability.
* **Kernel Trick:** SVM supports difference kernels (e.g., linear, polynomial, radial basis function (RBF)), making it versatile for various types of data distributions.

### *5.3) Data Preprocessing Required*

Yes, preprocessing is essential:

* **Scaling**: “**StandardScaler”** standardises the features to a standard scale. SVM is sensitive to the scale of input data, and feature scaling ensures that all features contribute equally to the model.
* **Handling Missing Value**s: These are not explicitly mentioned but involve checking for and handling any missing or inconsistent values.

### *5.4) Model Accuracy, Validation Method, and Performance Metrics*

**Accuracy:** The highest cross-validation accuracy achieved was **0.93**.

**Validation Method:** Stratified K-Fold Cross-Validation with five splits was used to ensure that each fold contained approximately the same percentage of samples of each target class, which is crucial for balanced evaluation.

**Performance Metrics:**

* **Accuracy:** The primary performance metric evaluates correctly classified samples' proportions.
* **Classification Report:** Provide detailed metrics, such as precision, recall, and F1-score, essential to evaluate the model’s performance, especially if the classes are imbalanced.

### *5.5) User Guide*

A user guide is provided in the form of a **README** file located within the **codes** folder. This guide contains detailed instructions for setting up the environment, running the model, and understanding the project’s structure and workflow.

### *5.6) Potential Model Improvements*

**1. Feature Engineering:**

* Create new features or transform existing ones to capture more complex patterns. For example, try using polynomial features or interaction terms.

**2. Ensemble Learning:**

* Combine SVM with other models (e.g., Random Forest, Gradient Boosting) to create an ensemble that can reduce variance and bias, potentially increasing overall accuracy

**3. Data Augmentation:**

* Collect more data or use techniques like SMOTE to generate synthetic samples, especially if there are class imbalance. More data can help the model generalise better.

## *6) Selection and Setup of Software Tools for the Project*

For the project, I will use the following software tools:

* Project Management Tool: **Trello** or **ClickUp**
* Version Control Tool: **GitHub**
* Collaboration Tool: **Google Drive**
* Team Communication Tool: **Discord**

### *6.1) Project Management Tool: Trello or ClickUp*

**Option 1: Trello**

1. Create a Trello Account:

* Go to [Trello's website](https://trello.com).
* Click “**Sign Up**”and enter email address.
* Verify email to activate Trello account.

2. Create a Team:

* Click on “**Create a Team**” from the sidebar.
* Enter team’s name and details, then click “**Create**”.

3. Create a Project Board:

* Click on “**Create New Board**”.
* Name the board and select created team.
* Click “**Create Board**” and start adding lists (e.g., To Do, In Progress, Done).

4. Invite Team Members:

* Click on “**Invite**” at the top of the board.
* Enter team members’ emails and send invitations.

**Option 2: ClickUp**

1. Create a ClickUp Account:

* Visit [ClickUp's website](https://clickup.com).
* Click “**Get Started**” and enter email address.
* Follow the instructions to complete account setup.

2. Create a Workspace:

* Once logged in, create a new workspace by entering your team name and details.
* Click “**Next**” to proceed with setup.

3. Set Up Spaces, Folders, and Lists:

* Create a new space for project (e.g., “Project XYZ”).
* Inside the space, create folders and lists to manage tasks.

4. Invite Team Members:

* Go to “**People**” in the workspace settings.
* Enter team members’ email to invite them to join the workspace.

### *6.2) Version Control Tool: Github*

1. Create a GitHub Account:

* Visit [GitHub](github.com) and click “**Sign Up**”.
* Complete the signup process and verify email.

2. Create a New Repository:

* Click on “**New**” to create a repository
* Enter the repository name and set it to **Public** or **Private**.
* Click “**Create Repository**”.

3. Invite Collaborators:

* Go to **Settings** > **Manage Access** > **Invite a Collaborator**.
* Enter the team members’ usernames or emails and send invitations.

### *6.3) Collaboration Tool: Google Drive*

1. Create a Google Account:

* Visit [Google Accounts](https://accounts.google.com/lifecycle/steps/signup/name?continue=https://myaccount.google.com/?pli%3D1&ddm=0&dsh=S-1976980594:1726216128532316&ec=GAlAwAE&flowEntry=SignUp&flowName=GlifWebSignIn&hl=en&service=accountsettings&TL=APps6ebheThOtifxdXCs8eda1yHVYCEgOSrnsatsaPNVzyb_1R1Fy_absexHEkbI) and create an account.

2. Set Up Google Account:

* Log in to your Google account and go to [Google Drive](https://drive.google.com).
* Click “**New**” > “**Folder**” to create a project folder.

3. Share the Folder with Team Members:

* Right-click on the folder and select “**Share**”.
* Enter team members’ emails and set their permissions.
* Click “**Send**”.

### *6.4) Team Communication Tool: Discord*

1. Create a Discord Account:

* Visit [Discord's website](https://discord.com) and click “**Login**” > “**Register**”.
* Enter details to create an account.

2. Create a Server:

* Once logged in, click on “+” on the left sidebar.
* Choose “**Create My Own**” and enter server name.

3. Set Up Channels:

* Click “**Create Channel**” to create text or voice channels (e.g., #general, #project-updates).
* Set channel permissions as needed.

4. Invite Team Members:

* Click on server name > “**Invite People**”.
* Enter team members’ usernames or send them the invite link.