CA Tasks Explanations and Sample Output – 04/07/2023

1. *Design Sketches*

- Task: Create clear visual sketches (diagrams) for two architectural designs:
  - Chained Multi-outputs
  - Hierarchical Modelling
- Purpose: These sketches should show how each part of your design connects and works together.
- Marks: 25%
- Sample Output: Example: Chained Multi-outputs

Sketch Description: A diagram showing three stages:

Stage 1: A model (e.g., Random Forest) classifies Type 2.

Stage 2: The same model classifies combined Type 2 and Type 3.

Stage 3: The model classifies combined Type 2, Type 3, and Type 4.

- Sample Output: A flowchart or block diagram with arrows indicating the flow from one stage to the next, with labels for each type.

2. Component Identification

- **Task**: List and describe all parts (components) for each design choice.
- **Purpose**: Explain the role of each part and how they interact within the system.
- **Marks**: 10%
- **Sample Output:** Detailed list and description of components for both designs

Component	Description
Model A	A Random Forest classifier that processes Type 2, Type 2+Type 3, and Type 2+Type 3+Type 4
Data Loader	Component responsible for loading and preprocessing data from CSV files
Data Formatter	Ensures that input data format is consistent across models

3. Connector Identification

- **Task**: List and describe how the parts (components) are connected in each design choice.
- **Purpose**: Explain how these connections help the parts communicate with each other.
- **Marks**: 10%
- **Sample Output:** Detailed list and description of connectors for both designs (Chained outputs and Hierarchical)

Connector	Description
Data Pipeline	Connects the data loader to the model for initial classification
Chained Connector	Passes the output from Type 2 classification to the combined Type 2+Type 3 model
Output Aggregator	Collects and formats the final output from the model chain

4. Data Elements Identification

- **Task**: Identify all the types of data used in each design choice.
- **Purpose**: Describe the data, its sources, and how it is processed, ensuring consistent data format across different models.
- **Marks**: 10%
- **Sample Output:** Detailed list and description of data elements

Data Element	Type	Source	Processing
Email Content	Text	AppGallery.csv	Preprocessed for text analysis
Type 2 Label	Categorical	AppGallery.csv	Used for initial classification
Type 3 Label	Categorical	AppGallery.csv	Filtered based on Type 2 classification
Type 4 Label	Categorical	AppGallery.csv	Filtered based on Type 2 and Type 3 classifications

5. Implementation of Design Choice

- **Task**: Develop a full working example any one design.
- **Purpose**: Write clean, well-documented code and track your progress using a version control system like Git. Add your lecturer as a collaborator to your repository.
- **Marks**: 25%
- **Sample Output:**
    - Detailed explanation of any one design choice implementation (Chained outputs/Hierarchical)
    - Include code snippets and results.

6. Report Presentation

- **Task**: Write a detailed report that includes:
    - Design sketches
    - Descriptions of components, connectors, and data elements
- **Purpose**: Ensure the report is clear, concise, and well-formatted, with proper references.
- **Marks**: 20%
- **Note**: Make sure to include a **'Conclusion'** section at the end to summarize your findings and observations

Using the CSV Files

You have two CSV files: AppGallery.csv and Purchasing.csv. Here's how to use them:

1. Start by looking at the contents of both CSV files to understand the types of data they contain.
2. Determine which columns in the CSV files correspond to the types (Type 1, Type 2, Type 3, Type 4) mentioned in your task.
3. Use the data from these files to create your design sketches and identify components.
4. Describe the data elements in your CSV files, including their types, sources, and how they will be processed in your AI model.
5. When implementing your Chained Multi-outputs design, use the data from the CSV files to train and test your machine learning models.