# Deep Learning Model API Documentation Assignment

**Objective**: Develop clear, user-friendly API documentation for interacting with your deep learning model, covering training, prediction, and evaluation functionalities.

4. Include Sample Requests and Usage Examples

1. **Define Core Model Actions**
   List the essential actions your API provides for interacting with your deep learning models:
   - **Training**: Describe the process for initiating or updating model training with new data.
   - **Prediction**: Detail how users can get predictions from the model by sending input data.
   - **Evaluation**: Outline how to evaluate the model's performance using specified metrics or test data.
   - ~~**Versioning**~~: Note if different versions of the model are supported, and document any requirements for version-specific calls.
2. **Describe API Requirements for Each Action**
   For each action, clarify how users should interact with the API, focusing on:
   - **Description**: Explain the function of each action and its purpose for the model (e.g., obtaining predictions, assessing accuracy).
   - **Request Structure**:
     - **Input Data Format**: Define the JSON structure or file format for data (for example, the required fields, data types, and any specific data ranges).
     - **Optional Parameters**: Describe any configuration options, such as hyperparameters for training, filters for predictions, or specific metrics for evaluation.
   - **Response Structure**:
     - Define what users can expect in the response for each action, including any data types, keys, and example values.
     - List relevant **status codes** (e.g., success, invalid input) with descriptions to help users understand API feedback.
   - **Error Handling**: Specify common error messages, particularly those related to model status (e.g., "Model not trained," "Data format error").
3. **Authentication & Authorization**
   - Detail any authentication required to interact with the model (e.g., API key, OAuth token).
   - Specify permissions, if applicable, for different user roles (e.g., prediction access for general users, training access for admin users).
4. **Include Sample Requests and Usage Examples**
   Provide sample API requests for each action (such as curl commands or code snippets) to show users how to structure their calls. Include realistic examples with expected responses to help users get started quickly.

5. **Metadata and Monitoring**
    - **Model Status**: Document how users can check if the model is active or ready to handle requests (e.g., status check for readiness).
    - **Metadata**: Include relevant metadata that may help users understand the model, such as version number, last trained date, and key metrics (e.g., accuracy, loss).

**Example for Training Action Documentation**

For the **Training** action, users would:

- Send training data and optional configurations in (JSON) format.
- Receive confirmation when training has been initiated, including a message with training status or any error that may occur if input data is invalid.

**Example for Prediction Action Documentation**

For the **Prediction** action:

- Users would send input data structured in (JSON).
- They would receive a prediction result, along with a confidence score and status code for successful or unsuccessful requests.

By focusing on model-centric actions and ensuring the instructions are clear, your documentation will enable others to easily interact with the deep learning model. Test each example thoroughly to confirm accuracy.