**Assignment Documentations**

This document outlines the process of developing, debugging, and deploying a FastAPI server to serve pre-trained regression models and provide predictions. The server is designed to load models, accept input data, process it, and return predictions. It also includes endpoints for listing available models and a home endpoint to check if the server is running.

**Process:**

The provided FastAPI server is designed to serve pre-trained regression models and provide predictions based on input data. Here is a brief overview of its functionality:

1. Server Setup:

* The server is created using FastAPI.
* CORS is configured to allow all origins.

2. Model Loading:

* Pre-trained models (RandomForest, SVR, KNN) are loaded from specified file paths using joblib.

3. Input Data Handling:

* The InputData class is defined using Pydantic to validate that the input data is a list of floats.

4. Endpoints:

* Prediction Endpoint (/predict/{model\_name}): Accepts input data and returns predictions from the specified model.
* List Models Endpoint (/models): Lists all available models.
* Home Endpoint (/): Provides a welcome message to check if the server is running.

5. Error Handling and Logging:

* Detailed logging is implemented to capture input data, prediction requests, and results.
* Error handling is added to manage issues during model loading and prediction.

**Findings:**

1. Model Loading:

- Successfully loaded models from specified file paths.

- Added logging to confirm successful model loading.

2. Input Data Validation:

- Ensured that input data is a list of numeric values (int or float).

- Added error handling for invalid input data.

3. Prediction Endpoint:

- Implemented the prediction endpoint to accept input data and return predictions.

- Added logging to capture input data and prediction results.

4. Common Issues:

- Encountered issues with incorrect working directory when running the server.

- Fixed by ensuring the correct directory is set before running the Uvicorn command.

**Recommendations:**

1. Model Compatibility:

- Ensure that the pre-trained models are compatible with the input data format.

- Validate model files to avoid loading errors.

2. Input Data Format:

- Clearly document the expected input data format for users.

- Provide examples of valid input data.

3. Error Handling:

- Continue to improve error handling to manage unexpected issues during prediction.

- Log detailed error messages to facilitate debugging.

4. Testing:

- Regularly test the server with different input data to ensure robustness.

- Implement unit tests for critical functions.

**Conclusion:**

The FastAPI server was successfully developed and debugged to serve regression models and provide predictions. Detailed logging and error handling were implemented to ensure reliability and facilitate debugging. The server is now ready for deployment and further testing.