"""Loads the trained model, preprocessor, and label encoder."""

This selected text is a **docstring** (documentation string). In Python, docstrings are used to explain the purpose and functionality of modules, classes, functions, and methods.

* **Purpose:** It serves as a brief, high-level summary of what the load\_artifacts() function does.
* **Content:** It clearly states that the function's responsibility is to load three key components:

1. The **trained machine learning model** (e.g., your Logistic Regression, Random Forest, XGBoost, or Neural Network model).
2. The **preprocessor** (your ColumnTransformer pipeline that handles scaling and one-hot encoding of input features).
3. The **label encoder** (your LabelEncoder that maps numerical predictions back to 'H', 'D', 'A' labels).

* **Accessibility:** Docstrings are accessible at runtime using help() or by inspecting the \_\_doc\_\_ attribute of the function. For example, if you were to run help(load\_artifacts) in a Python interpreter, this docstring would be displayed.
* **Readability:** It makes the code more readable and understandable for anyone (including yourself in the future) who needs to know what a particular function is supposed to do without having to read through all its implementation details.

You're asking for an explanation of the @app.on\_event("startup") decorator in the Canvas.

This line is a **FastAPI decorator** that registers a function to be executed **once, when the FastAPI application starts up.**

Here's a breakdown:

* **@app.on\_event("startup")**: This is a decorator provided by FastAPI. It tells the FastAPI application that the function immediately following it (async def startup\_event(): in this case) should be run when the application initializes and becomes ready to serve requests.
* **async def startup\_event():**: This defines an asynchronous function named startup\_event. FastAPI expects event handlers to be async functions.
* **load\_artifacts()**: Inside startup\_event(), the load\_artifacts() function is called.

**Why is this important for your ML service?**

Loading machine learning models, preprocessors, and label encoders from disk can be a time-consuming operation. If you were to load them inside the /predict endpoint function, they would be reloaded **every single time** a prediction request comes in. This would make your API very slow and inefficient.

By using @app.on\_event("startup"), you ensure that:

1. **Efficiency:** The model, preprocessor, and label encoder are loaded into memory **only once** when the FastAPI application starts.
2. **Performance:** Subsequent prediction requests can immediately access the loaded artifacts from memory, leading to much faster response times.
3. **Resource Management:** It's a clean way to manage resources that need to be initialized before the application begins its main work.

In essence, it's a critical part of setting up your ML service to be performant and ready to handle prediction requests efficiently.