Project Instructions — Stage 13: Productization

Today's Project Contribution:

Today you'll prepare and complete your analysis project for reuse, clarity, stakeholder handoff, and optional deployment. This task aligns with the Productization stage, where you will:

- Package your analysis for reuse, technical stakeholder handoff and automation of risk aware presentation outputs and optional deployment
- Document and modularize your project code
- Implement API and/or dashboard deployment of your analysis
- Prepare deliverables suitable for stakeholder review and technical reproduction

By the end of this assignment, your project should include the elements listed below.

Deliverable Options

- Add options to
    - save your model during your analysis (overriding any that may exist) (`model/model.pkl`)
    - use a saved model, if it exists, rather than running it anew
- Develop reusable functions for running your full analysis including prediction, plotting, or anything else you might need, in `/src/` (e.g., `src/utils.py`).
- Create **at least one of the following**:
    - **Flask API**
        - `/predict` endpoint accepting JSON POST requests
        - Optional path parameters: `/predict/<input1>` and `/predict/<input1>/<input2>`
        - Additional routes (optional but recommended):
            - `/run_full_analysis` — runs the full analysis from start to finish and generates stakeholder-ready outputs (PDF, CSV, charts etc)
            - `/run_full_analysis/<param1>/<param2>` — allows user-provided inputs to modify part of the analysis and returns those results
    - **Streamlit or Dash dashboard**
        - Accepts user inputs for prediction
        - Displays charts, tables, or other visual outputs
        - optional: have it present your stakeholder-ready presentation (or a version of it)
- Include error handling for invalid inputs.
- Create `requirements.txt` for reproducibility.

- Document repo usage in `README.md` including:
    - Setup instructions
        - How to use your repo from start to finish from a fresh git pull
    - Example API requests or screenshots
    - Notes on assumptions, risks, and next steps
- Add a **Stakeholder Handoff Summary** with:
    - Overview of project and purpose
    - Key findings and recommendations
    - Assumptions and limitations
    - Risks and potential issues
    - Instructions for using deliverables
    - Suggested next steps
- Validate your README and repo structure by **cloning into a fresh environment** and running your analysis end-to-end.

How This Fits Into Your Final Project

Your work today builds toward and completes your end-to-end financial engineering project.

- You are taking a trained model and packaging it for reuse, demonstration, and handoff.
- By adding modular functions, an API or dashboard, and proper documentation, you are ensuring your project can be used, understood, and extended by others.
- Stakeholder-ready outputs show how your results can be presented in a professional setting.
- Practicing Flask, Streamlit, or Dash prepares you for optional future deployment.

Before next class:

- Save your files in the appropriate folders (`/data/`, `/src/`, `/notebooks/`, `/model/`, `/reports/`)
- Commit and push changes to your GitHub repo
- Add any new functions to your modular, stage-driven libraries for easy reuse and flexibility.
- Confirm that your README and deliverables work from a fresh clone
- Ensure your stakeholder summary clearly communicates results and instructions

By the end of the course, your full project will follow the complete lifecycle from raw data to reusable model, analysis, stakeholder-ready deliverables, and optional deployment.

Suggested Folder Structure

```
project/
    /data/       # raw and processed data
    /src/        # reusable functions and scripts
    /notebooks/  # exploratory and final notebooks
    /reports/    # stakeholder summaries, charts, PDFs
    /model/      # pickled model objects
```

```
app.py       # Flask API (if implemented)
app_streamlit.py  # Streamlit dashboard (if implemented)
requirements.txt
README.md
```

Optional Enhancements

- Add a Dash dashboard as an alternative to Streamlit.
- Include batch scripts or automated app launching and/or automated pipelines for repeated analyses.
- Use logging to record analysis steps and errors.
- Version control your models (`model_v1.pkl`, `model_v2.pkl`) to track improvements.
