# How to Create and Understand Your Flask App for ML Model Deployment

## 1. Create a Project Folder

Open your terminal and create a new folder for your project:

mkdir income\_classifier\_api

cd income\_classifier\_api

## 2. Create a Virtual Environment

Use Python to create a virtual environment to isolate dependencies:

python -m venv venv\_income\_api

Activate it:

- Windows: venv\_income\_api\Scripts\activate

- macOS/Linux: source venv\_income\_api/bin/activate

## 3. Install Required Packages

Install the libraries needed for your model and Flask API:

pip install flask pandas scikit-learn joblib

## 4. Freeze Your Environment

To create a record of all dependencies:

pip freeze > requirements.txt

Later, others can recreate the same environment using:

pip install -r requirements.txt

## 5. Understand the Flask App Structure

app.py: Contains the logic for your API.

Key components:

- Import necessary libraries (Flask, joblib, pandas)

- Load trained model (joblib.load)

- Define routes:

- '/' : Homepage

- '/predict' : Accepts JSON input, preprocesses it, and returns a prediction

## 6. Running the App

Start the API locally by running:

python app.py

You should see it hosted at http://127.0.0.1:5000

## 7. Test Your API

Using curl:

curl -X POST http://127.0.0.1:5000/predict -H "Content-Type: application/json" -d @test\_payload.json

Using PowerShell:

Invoke-RestMethod -Uri http://127.0.0.1:5000/predict -Method POST -Body (Get-Content .\test\_payload.json -Raw) -ContentType "application/json"

## 8. Project Folder Example

income\_classifier\_api/

├── app.py # Flask app

├── income\_model.pkl # Trained model

├── model\_features.pkl # Feature names used in training

├── requirements.txt # Dependencies

├── test\_payload.json # Sample test input