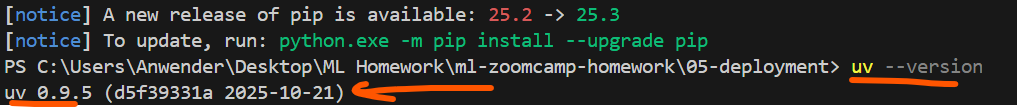
**Question 1**

pip install uv  
uv –version



**Question 2**

mkdir homework  
cd homework  
uv init  
uv add scikit-learn==1.6.1  
  
A computer screen with text on it

AI-generated content may be incorrect.

cat uv.lock | grep -A 5 "name = \"scikit-learn\""

**Question 3**

Download file “wget <https://github.com/DataTalksClub/machine-learning-zoomcamp/raw/refs/heads/master/cohorts/2025/05-deployment/pipeline_v1.bin>”

**Create** “q3\_test.py”

import pickle

# *Load the model*

with open('pipeline\_v1.bin', 'rb') as f\_in:

pipeline = pickle.load(f\_in)

# *The record to score*

client = {

"lead\_source": "paid\_ads",  
 "number\_of\_courses\_viewed": 2,  
 "annual\_income": 79276.0  
}

# *Make prediction*

probability = pipeline.predict\_proba([client])[0, 1]

print(f"Probability: {probability:.3f}")

md5sum pipeline\_v1. Bin  
uv run python q3\_test.py (To see the results)



**Question 4**

uv add fastapi uvicorn  
  
**Create** “predict.py” then add this code

import pickle  
from fastapi import FastAPI  
from pydantic import BaseModel  
import uvicorn

# *Load model*

with open('pipeline\_v1.bin', 'rb') as f\_in:

    pipeline = pickle.load(f\_in)

# *Define the INPUT schema*

class Lead(BaseModel):  
    lead\_source: str  
    number\_of\_courses\_viewed: int  
    annual\_income: float

# *Define the OUTPUT schema*

class PredictResponse(BaseModel):  
    conversion\_probability: float  
    will\_convert: bool

# *FastAPI app*

app = FastAPI(title="Lead Scoring Service")

@app.post("/predict")  
def predict(lead: Lead) -> PredictResponse:

    # *Convert to the format*

    lead\_dict = lead.model\_dump()

    # *Make prediction*

    probability = float(pipeline.predict\_proba([lead\_dict])[0, 1])

    return PredictResponse(  
        conversion\_probability=probability,  
        will\_convert=probability >= 0.5  
    )

if \_\_name\_\_ == "\_\_main\_\_":  
    uvicorn.run(app, host="0.0.0.0", port=9696)

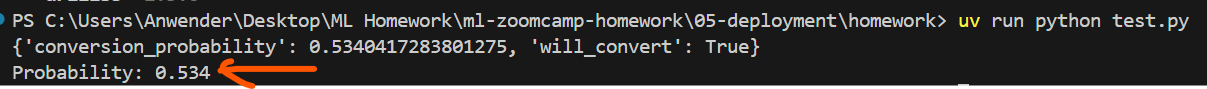
**Create** “test.py” then add this code

import requests  
url = "http://localhost:9696/predict"

client = {  
    "lead\_source": "organic\_search",  
    "number\_of\_courses\_viewed": 4,  
    "annual\_income": 80304.0  
}

response = requests.post(url, json=client).json()  
print(response)  
print(f"Probability: {response['conversion\_probability']:.3f}")

uv run python test.py



**Question 5 – Docker (using Linux / WSL)**

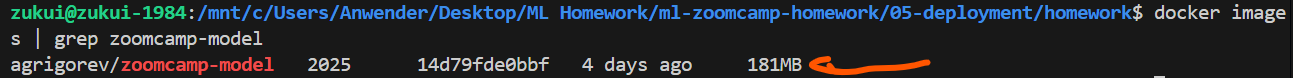
<https://hub.docker.com/r/agrigorev/zoomcamp-model>

docker pull agrigorev/zoomcamp-model:2025

A computer screen with white text

AI-generated content may be incorrect.

docker images | grep zoomcamp-model (to check image size) – 181 MB



**Question 6**

Update the code on “**predict.py**”

**import** pickle  
**from** fastapi **import** FastAPI  
**from** pydantic **import** BaseModel  
**import** uvicorn

*# Load the model from the base image*

**with** open('pipeline\_v2.bin', 'rb') **as** f\_in:  
 pipeline = pickle.load(f\_in)

docker build -t lead-scoring .

A screen shot of a computer

AI-generated content may be incorrect.

docker run -it --rm -p 9696:9696 lead-scoring

A screen shot of a computer

AI-generated content may be incorrect.

Run in new terminal “uv run python test.py”

