Machine Learning in Production
Midterm 2, Spring 2025

Claire Le Goues and Austin Henley

Name: _____

Andrew ID: _____

Instructions:

- Including this cover sheet and the scenario, your exam should have **9** pages. Make sure you are not missing any pages. *You may detach the last page and recycle it after the exam.*
- All questions in this midterm refer to the scenario on **Page 9**. Answers are graded in the context of the scenario; **generic answers that do not relate to the scenario will not receive full credit.**
- The exam has a maximum score of **66** points. The point value of each problem is indicated. We designed the exam anticipating approximately one minute per point.
- **Please write legibly**. We are unlikely to be able to grade your solution if we can't read it.
- We give an amount of space commensurate with what we expect you to need for each question. We use horizontal lines to suggest where to not use the full page. You may exceed those limits if it is clear where to find the rest of your answer. However, we strongly recommend writing concise, careful answers; short and specific is much better than long, vague, or rambling.
- If you write answers outside the allocated space, **clearly indicate** where your answer can be found.
- This is a **closed book exam**; no books or electronics allowed. You may refer to 6 sheets of notes (handwritten or typed, both sides).

Question 1: Scaling and Operations [25 points]

All questions in this exam relate to the scenario on the last page, which you may detach.

(a) [8 points] The typical input to the model (photos and location/orientation data) is usually around 30 megabytes, whereas the model itself is very large (about 500 gigabytes). You are trying to decide how to deploy the model. You consider several choices:

- *batch processing*, such as by running jobs on large amounts of data all at once.
- *stream processing*, like you've seen with Kafka, which decouples producers and consumers of events.
- *as a service* that responds with an action recommendation continuously.

Which is best within the scenario, and why? Your answer must be internally consistent, and must demonstrate an understanding of one or more specific tradeoffs involved between choices.

☐ Batch processing ☐ Stream processing ☐ As a service

Justification:

(b) [5 points] Image data in the training dataset is never changed, only added. However, other training data sometimes needs to be corrected, when the initial data reported to was not correct. This is tabular data of which less than 0.1% per year is corrected. Which of the following versioning strategies is the most *space efficient* for the tabular data in this scenario:

☐ Keeping copies ☐ Tracking deltas ☐ Recording offsets

Justification:

(c) [6 points] You occasionally will need to update the model with new data, or want to experiment with different model versions in production. Therefore, you want to track which prediction was made with which version of the model, and what data the model was trained on. How can MLFlow (or similar tools) can help with this problem? What additional information needs to be tracked outside of MLFlow to fully realize this goal?

How MLFlow can help:

What information needs to be tracked additionally:

(d) [6 points] Deploying and updating a large foundation model can be tricky, and the amount of MLOps tooling out there can be overwhelming. Provide an example of technical debt that your team may take on intentionally, at least in the short term, that would be plausible in the context of the scenario. Describe rationale for taking on the short-term technical debt as well as the "better" solution that your team could implement later to resolve the debt.

Technical debt example:

Rationale for taking on the debt:

Better solution:

Question 2: Fairness [18 points]

You are worried about whether the software supports all families/homes fairly. You analyze the predictions your system produces, grouped by zip code.

(a) [4 points] Making more mistakes when predicting potential intruders in areas with lower median incomes produces:

☐ a harm of allocation ☐ a harm of representation ☐ both ☐ neither

Justification:

(b) [6 points] Consider sources of bias that could lead to lower accuracy for homes in lower-income areas. Select one of the following sources of biases discussed in class – *tainted labels; historical bias; skewed sample; limited features; proxies* – that may be responsible and briefly explain how it might have led to the different model behaviors for low-income vs. high-income areas in this scenario.

Select #1: ☐ *tainted labels* ☐ *historical bias* ☐ *skewed sample* ☐ *limited features* ☐ *proxies*

Explanation:

(c) [2 points] You have access to the last 1000 predictions for houses in low income areas, and the last 5000 predictions for high income areas. You can therefore see how often your model identified potential intruders, and the action the model predicted a homeowner should take. However, you do not have direct information about which predictions were good, and feedback on what customers actually did is delayed and not yet available. Is there any notion of fairness that you could measure with this information you have now? (No explanation required)

☐ Anti-classification ☐ Group fairness ☐ Equalized odds ☐ None of these

(d) [6 points] Propose two plausible interventions that could improve system fairness, with respect to whatever fairness metric you prefer, in this scenario. Do not say to train the model for longer or simply on more data, without a very compelling system-level justification.

Intervention 1:

Intervention 2:

(writing below this line is allowed but discouraged)

Question 3: Explainability/Transparency [9 points]

In the scenario, homeowners would benefit from explanations for better understanding when to trust the model suggestions to make informed decisions about actions to take (e.g., potential intruders or watering their flowers).

(a) [5 points] Describe a technique that could provide *explanations of individual predictions* that could be used by a data scientist on your team to understand why the model made a given (faulty) prediction. Rough intuition behind how such a technique works is sufficient.

(b) [4 points] Would this method you suggested in part (a) provide explanations that are appropriate for homeowners in the scenario? Explain why/why not, and justify your answer.

(writing below this line is allowed but discouraged)

Question 4: Security and Robustness [14 points]

The team is worried about security, robustness, and safety but it has not been a high priority to date. You want to be responsible and mitigate common problems before they occur.

(a) [6 points] Within the context of the scenario, consider a ML-specific attack that could be attempted (perhaps by a competitor, hacker, or thief) and how the product could mitigate that attack. Answer the four prompts below. Your answer must be reasonably realistic in the scenario. You may make assumptions about the product (just state them).

Attacker goal (intended outcomes):

Attack method (what the attacker would do):

Which security property does the attack intend to undermine?

☐ Confidentiality ☐ Integrity ☐ Availability

Mitigation strategy (how the developers can prevent the attack/make it harder):

(b) [4 points] What is one *system level* intervention you could implement to improve robustness in this scenario? (Answers that boil down to improving ML model correctness will receive no credit).

(c) [4 points] You would like to argue in a structured fashion that your system will maintain at least 95% availability. You already have data showing 98% uptime over a 7-day period, but your colleague, an expert in dependable and safety-critical systems, is unconvinced. What are two other arguments or pieces of evidence you or your system could provide as part of a structured assurance case in support of this claim? You may speculate any kind of plausible system architecture, component, or implementation choice as part of your answer.

(writing below this line is allowed but discouraged)

Scenario

You've just joined the Surveillance4All startup that is focused on rapid innovation in AI and home-surveillance technology. The company's mission is to empower home owners, communities, and researchers by capturing the data that is all around us for environmental, security, and policy decisions.

The team has already demonstrated feasibility of the approach, filed patents, and published several papers. They have designed small, very low-cost cameras that can be used to cover a building. Using the data they've collected from their early customers, they trained a powerful vision-based foundation model (like an LLM, but for images). The model is capable of detecting common maintenance actions that a homeowner should take, producing recommendations like "water the flowers", or "call 911, intruder detected", "schedule gutter cleaning". This is possible since dozens of cameras are attached to a given house, and can capture images of wildlife, automobiles, and people.

Your current goal is to transform this promising early-stage proof of concept into a scalable product. The business model is to sell the hardware and software subscription at cost to homeowners, and to sell aggregate or anonymized data to third parties for a profit. For example, federal environmental agencies may use the data to understand the impact of housing developments on ecosystems; commercial entities may use data to decide where to expand development or pursue commercial real estate. The model will be regularly retrained based on recent data collection.

The team has attracted several strong AI researchers who are experienced with state of the art vision and foundation models as well as strong hardware engineers to design the cameras. However, the team lacks extensive software engineering expertise. Innovation and rapid prototyping have been prioritized over building robust, long-term infrastructure.