AI Engineering / Machine Learning in Production
Midterm, Spring 2022

Christian Kaestner and Eunsuk Kang

Name: _____

Andrew ID: _____

Instructions:

- Not including this cover sheet, your exam should have **9** pages. Make sure you are not missing any pages.
- All questions in this midterm refer to the scenario on the first page. 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 **56** points. The point value of each problem is indicated. We allocated approximately one point per minute.
- 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.
- **Do NOT write anything you want us to grade on the back of pages.** We will scan the exam and will not look at the back sides.
- This is a **closed book exam**; no books or electronics allowed. You may refer to 6 sheets of notes (handwritten or typed, both sides).

Scenario

You are part of a large appliance company with a long pre-internet history, now trying to differentiate their top line of fridges with smart features, such as automatic tracking of contents and automatic ordering.

You are on a team that is developing a new innovative feature called “**EatSoon**”: A mechanism to automatically detect when vegetables, fruit and meats are starting to spoil. You have a promising prototype that uses camera images and an experimental robotic nose sensor to detect early signs of decay visually and through smell. Since science still has no idea how smell actually works and what the robotic nose actually measures, you rely on a deep neural network model to detect decay, which works surprisingly well. For visual detection, you also use a deep neural network model. The final **EatSoon** detector is an ensemble model of both the smell detector and the visual detector.

You’ve gathered labeled visual training data from public Internet photos; additionally, you’ve collected both smell and visual data from placing produce in 50 test fridges over the last two years. You also have access to unlabeled camera pictures from within fridges from older smart fridge models that your company has sold in the past.

You envision that this new feature, **EatSoon**, could be used to send alerts to users on a mobile app or to display what products should be eaten soon or may need to be discarded. **EatSoon** is also planned to be connected to a feature that automatically orders grocery items via your partners at Instacart, which other teams are currently developing in parallel. The infrastructure in past generations of smart fridges by your company has moderate computing power of a typical tablet. You estimate that performing inference of the kind of deep neural networks needed on the fridge might need specialized hardware (a tensor processing ship) which would cost about $80 per unit. The existing fridges assume that they are always online through the user’s home wifi. The company makes money by selling fridges and does not yet have a subscription model (while some higher ups are excited about this, it is not clear whether it would be popular among customers).

Your team is currently focused on the core models of **EatSoon**, but will soon work with other teams to integrate **EatSoon** into the smart fridge software and the mobile app. You want to present a prototype of **EatSoon** to the public soon to create some marketing buzz. Hopefully, you can start shipping actual fridges with **EatSoon** in about 1 year.

Question 1: Goals and Telemetry [15 points]

Your company's organizational objective is to be a successful appliance company and make money for its stakeholders, but that is a very slow metric and the influence of the new fridges or even the new **EatSoon** feature in the fridge will be hard to identify in the stock price. Therefore, define more specific metrics and how you would measure and operationalize from that data (described with enough detail to be independently measured):

(a) **Leading indicator for the success of the new fridge**

Proposed measure:

Data to collect:

Operationalization:

(b) **Model property: How often the EatSoon model incorrectly detects food as spoiled when it isn't actually spoiled**

Proposed measure:

Data to collect:

Operationalization:

(c) **Model property: How often the EatSoon model misses detecting spoiled food or detects it too late**

Proposed measure:

Data to collect:

Operationalization:

(writing below this line is allowed but discouraged)

Question 2: Trade Offs [13 points]

You are trying to decide whether to deploy the **EatSoon** models on the *fridge* or in the *cloud*.

Your team is split about which deployment option to choose and lots of arguments are thrown around. Some arguments are technically true, but do not seem very convincing since the expected difference would not be meaningful in practice.

(a) [4 points] Make one convincing argument for deploying the models on the fridge that is plausible in the scenario. Your argument should refer to at least one model or system quality.

(b) [4 points] Make one convincing argument for deploying the models in the cloud that is plausible in the scenario. Your argument should refer to at least one model or system quality.

(writing below this line is allowed but discouraged)

(c) [5 points] Considering the tradeoffs between both deployment options (including but not limited to your answers above), which do you recommend? Briefly justify your answer by arguing about the relative importance of qualities. If you need more information, indicate what information you would need and how you would make a decision with it once you have it.

☐ Fridge is better ☐ Cloud is better ☐ Need more information

Justification:

(writing below this line is allowed but discouraged)

Question 3: Model and Data Quality [14 points]

(a) [4 points] Accuracy results of your visual decay detector seem promising but you are worried that accuracy in production might be lower even though you do everything right by splitting the data randomly into training, validation, and test data. Briefly given an example of possible *shortcut learning* in this scenario caused by a *mismatch of training distribution and target distribution* to illustrate the potential problem.

(b) [4 points] Your team is skeptical about installing the continuous integration tool Jenkins. They ask “Why don’t we just run our tests locally? Why do we need all of this extra complexity and an extra machine?” Briefly explain some benefits of continuous integration with Jenkins that might convince them.

(writing below this line is allowed but discouraged)

(c) The models are trained based on data collected from multiple sources, including pictures from the Internet and sensor readings from test fridges collected in the last 2 years.

- [2 points] Name one possible data quality problem that could be detected by enforcing a schema:

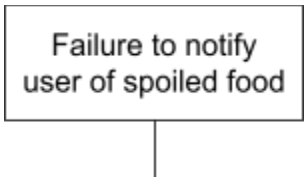
- [4 points] Name one concrete example of an organizational issue that could lead to poor data quality and briefly explain how you would try to mitigate it:

(writing below this line is allowed but discouraged)

Question 4: Mitigating Mistakes [14 points]

Consider the following requirement for the smart fridge system: *The user should receive a notification on the mobile app when the fridge has detected spoiled food.*

(i) [8 points] Draw a fault tree that shows **at least two possible ways** in which the system may fail to satisfy the requirement, with the top-level event given below. When appropriate, group basic events using one or more levels of intermediate events.



(ii) [2 points] State one **environmental assumption** that is necessary for the system to satisfy the requirement.

(iii) [4 points] Describe a mitigation strategy for one of the possible failures that you've identified in the fault tree. The mitigation should be at the system level, outside of the ML component (i.e., not just "train a more accurate model"). Your answer must describe how the strategy reduces the likelihood of the requirement violation. You do not need to update the fault tree.

(writing below this line is allowed but discouraged)