

Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

a. Define a hypothetical AI problem

Predicting daily supermarket/ store sales demand for fresh bread.

b. List 3 objectives

Minimize Food Waste- Accurately predict demand to prevent over-baking and reduce the amount of unsold bread.

Optimize Production- Improve the bakery's operational efficiency by creating more accurate schedules for staffing and raw ingredient orders.

Prevent stockouts- Ensure sufficient stock is available to meet customer demand, particularly during peak times, thereby protecting revenue and customer satisfaction.

Stakeholders:

Store Manager- deals with overall store profitability, which is directly impacted by minimizing waste (a cost) and preventing stockouts (a loss of revenue).

Bakery Department Head- needs the daily prediction number to plan labour, manage ingredient inventory and execute the baking schedule.

c. Propose 1 Key Performance Indicator (KPI) to measure success.

Percentage reduction in daily bread waste.

2. Data Collection & Preprocessing (8 points)

a. Identify 2 data sources for your problem.

Historical sales data from cash registers (POS)- records from the store's cash registers Point-of-Sale system.

External data- local weather forecasts and holiday/event calendars.

b. Explain 1 potential bias in the data.

Stockout Bias- sales data might show low demand just because the store ran out of bread, leading the model to under-predict future demand.

c. Outline 3 preprocessing steps

Handling Missing Data- fill gaps from register downtime with averages.

Feature Engineering- create new, meaningful columns from raw data, such as converting a date into a DayOfWeek (1-7).

Normalization- rescale numerical features like temperature to a 0-1 range, to ensure the model treats them equally.

3. Model Development (8 points)

a. Choose a model (e.g., Random Forest, Neural Network) and justify your choice.

Random Forest- because it effectively predicts numerical values (regression) using a mix of different feature types (categorical like holidays, numerical like no. of loaves).

b. Describe how you would split data into training/validation/test sets.

Training (70%) on older data.

Validation (15%) on recent data for tuning

Test (15%) on the most recent data for final evaluation.

c. Name 2 hyperparameters you would tune and why.

n_estimators (number of trees)- more trees generally make the model more accurate and stable but adding too many just makes it slow to train.

max_depth- to control model complexity and prevent overfitting. If the trees are too deep, the model "memorizes" the old sales data instead of learning the real underlying pattern. Making it very bad at predicting sales on new, future days.

4. Evaluation & Deployment (8 points)

a. Select 2 evaluation metrics and explain their relevance.

Mean Absolute Error (MAE) to measure the average error in loaves per day.

This is an important metric especially for the store manager. An MAE of 6 provides a simple, direct answer: on average, our forecast will be wrong by 6 loaves. This number can be directly translated into the cost of waste or the risk of stockouts.

R-squared to see how much sales variance the model explains.

Shows how well your features like weather or holiday predict the sales. A high R squared score means your features are very good at predicting the sales patterns. A low R squared means the features are bad at explaining sales, and the model is mostly just guessing.

b. What is concept drift? How would you monitor it post-deployment?

Concept drift is what happens when the real-world patterns your model was trained on change over time, making its predictions less accurate for example an outdated model/concept.

For example, if a new rival supermarket with its own bakery opens nearby. This steals a portion of your customers. The AI model, trained on old data (before the rival existed), still expects high sales. It will consistently over-predict demand, leading to significant and continuous food waste.

To monitor it, continuously track the model's daily error.

Automatically log the model's prediction (e.g. Predict 150) versus the actual sales (e.g. Sold 120). If the Mean Average Error gets much worse and stays worse for several days, it triggers an alert. This alert tells you it's time to retrain the model on this new data so it can learn the new, lower sales pattern.

c. Describe 1 technical challenge during deployment

Ensuring the automated system successfully pulls the sales logs of each day, weather forecasts of the next day, and the stores promotion calendar schedules every single night without failure, could be a challenge for example if the weather API is down, the whole prediction fails.