A green and black logo

Description automatically generatedA black background with blue text

Description automatically generated

**6CS012/HJ1: Artificial Intelligence and Machine Learning**

**(Q&A)**

**Full Name : Diparshan Baral**

**University ID : 2358244**

**Module Leader : Siman Giri**

**University email :** [**D.Baral@wlv.ac.uk**](mailto:D.Baral@wlv.ac.uk)

**Date of submission: 2025-05-20**

**Table of Contents**

[1. Long Question: 1](#_Toc198638634)

[1.1. Data Drift 1](#_Toc198638635)

[1.2. Imbalanced Data 2](#_Toc198638636)

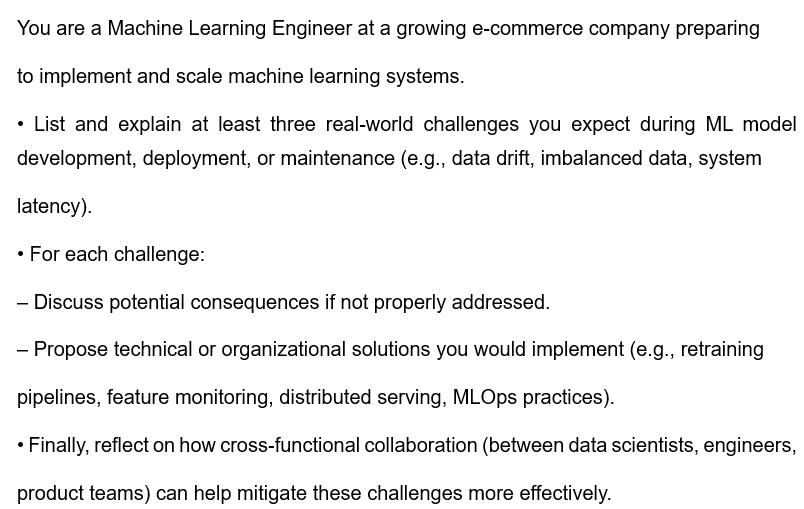
[1.3. Latency and Slow Predictions 2](#_Toc198638637)

[2. Short Question 3](#_Toc198638638)

[2.1. Overfitting 3](#_Toc198638639)

[2.2. Neural Network Architecture 4](#_Toc198638640)

# Long Question:



**Answer:**

As a Machine Learning Engineer at a fast-growing e-commerce company, here are three real-world challenges I’d expect when building and scaling ML systems:

## Data Drift

**What it is:** Over time, the type of data coming in might change. For example, users may start behaving differently or we might add new products that weren't there when the model was trained.

**Why it’s a problem:** If the model was trained in old patterns, it might start making bad predictions.

**How to fix it:** We can set up monitoring tools to keep an eye on incoming data and compare it to training data. If we notice big changes, we can retrain the model regularly to keep it updated.

## Imbalanced Data

**What it is:** Sometimes one class is way more common than the other — like 95% of users don’t return items, but 5% do.

**Why it’s a problem:** The model might just learn to ignore the rare cases and get lazy — it could predict “no return” all the time.

**How to fix it:** Use techniques like oversampling, undersampling, or special algorithms that can handle imbalanced data better (like XGBoost with class weights).

## Latency and Slow Predictions

**What it is:** Models that are too slow can delay responses, especially during peak shopping times.

**Why it’s a problem:** Nobody wants to wait — slow systems can mean lost sales or bad user experience.

**How to fix it:** Use lighter models for real-time tasks or deploy models with optimized serving solutions like TensorFlow Serving or use caching.

# Short Question

## Overfitting

A white background with black text

AI-generated content may be incorrect.

**Answer:**

Overfitting is when your model does great on the training data but messes up on new, unseen data. Two simple ways to handle this are:

**Dropout:**

**How it works:** Dropout randomly turns off some neurons during training, so the model doesn’t rely too much on any one path. This makes it more flexible and better at handling new data.

**Example:** In an image recognition project, like recognizing digits, dropout helps make sure the model doesn’t just memorize the training images but can also work on new ones.

**Early Stopping:**

**How it works:** While training, we keep checking how well the model does on test data. If the performance starts to get worse, we stop training early before it overfits.

**Example:** In a project like classifying positive or negative reviews, early stopping helps make sure the model doesn’t memorize training texts and stays general.

## Neural Network Architecture

A close-up of words

AI-generated content may be incorrect.

**Answer:**

**Main difference:** A regular neural network is built to predict or classify something — like whether a message is spam or not. An autoencoder, on the other hand, tries to copy the input. It first compresses the input (encoder), then try to recreate it (decoder) and in between encoder and decoder it has a bottle neck.

**Example:** Let’s say we give it an image. The autoencoder learns to compress the image into fewer numbers, then rebuilds it. If it does this well, it means it learned the important features.

**Where it helps:** Autoencoders are super useful in finding weird or unexpected data like fraud. If we train one on normal transaction data, it’ll do a good job recreating it. But if a new transaction looks very different (maybe it’s a fraud), it won’t be able to recreate it well and that tells us something’s wrong.

They’re also great for cleaning data or reducing size — especially when we have large and messy info like transaction logs.