Long Question 1:

Unsupervised Learning at eSewa

As a Data Scientist at eSewa, I believe unsupervised learning can really help in two big areas: catching fraud and understanding our customers better.

1. Fraud Detection

I think detecting fraudulent transactions is key to keeping users safe and trusting us. Since we might not have a lot of labeled fraud data, using unsupervised learning like anomaly detection makes sense.

I recommend using Isolation Forest, which is good for big datasets, One-Class SVM to learn normal patterns, and autoencoders to spot anomalies by how well they reconstruct transactions.

We can use this in eSewa by setting up a system that flags suspicious transactions in real-time for review or blocks them automatically.

2. Customer Segmentation

Understanding our customers can help us offer better, personalized services. I suggest using clustering to group customers based on their behavior.

Algorithms like K-Means can divide customers into groups, DBSCAN can handle different cluster shapes, and hierarchical clustering shows how groups relate.

Once we have these groups, we can tailor marketing to each, suggest services they might like, and even develop new features for specific needs.

Short Question 2.1.1: Overfitting and Underfitting

In machine learning, overfitting and underfitting are common issues that affect how well a model works.

Overfitting: This happens when a model learns the training data too well, including the noise. It does great on the training set but poorly on new data because it memorized instead of learned.

Underfitting: This is when the model is too simple and misses the patterns in the data, so it performs badly on both training and new data.

Both are problems: overfitting means it won’t work on new data, and underfitting means it didn’t learn enough to be useful.

For example, if I try to fit a complex curve to data that’s actually straight with some noise, that’s overfitting. If I use a straight line for data that curves, that’s underfitting.

Short Question 2.2.1: CNN vs. RNN

CNNs and RNNs are two types of neural networks, each good for different tasks.

CNN: Great for images, using layers to spot patterns like edges and shapes. It’s like looking at a picture and noticing details.

RNN: Best for sequences, like time series or text, where order matters. It remembers previous inputs, like reading a story where each word depends on the last.

For example, use CNN for recognizing faces in photos, and RNN for predicting stock prices based on past data.

Training them has challenges like vanishing gradients, where updates get too small, and overfitting. To fix this, use ReLU for gradients, dropout to prevent overfitting, and maybe use pre-trained models to save computing power.

Short Question 2.2.1:

CNN vs. RNNThe difference between CNN and RNN lies in their data handling and applications, with common training challenges addressed below.

| **Aspect** | **CNN (Convolutional Neural Network)** | **RNN (Recurrent Neural Network)** |
| --- | --- | --- |
| **Data Type** | CNNs are used to work with data like images or videos, where the position and space between pixels matter. | RNNs are used for data that comes in a sequence, like sentences or time-series data. |
| **Structure** | CNNs use special layers called convolutional layers that can detect patterns such as edges or textures in images. | RNNs have loops in their structure, which help them remember what came earlier in a sequence. |
| **Applications** | CNNs are mainly used for tasks like recognizing objects in pictures or analyzing videos. | RNNs are useful for tasks like predicting future values in time-series data or understanding language. |
| **Challenges** | CNNs can easily overfit the data and also require a lot of computer power to train. | RNNs can forget earlier information due to a problem called vanishing gradients, and they can also overfit. |
| **Solutions** | To fix CNN problems, we can use techniques like dropout, early stopping, or using already trained models. | For RNNs, we use improved versions like LSTM or GRU, and techniques like gradient clipping and regularization. |

Challenges in Training: Include vanishing gradients (gradients become too small), overfitting (model relies on training data too much), and high computational cost.

Solutions: Use ReLU for gradients, LSTM/GRU for RNNs, dropout and data augmentation for overfitting, and transfer learning for cost efficiency.