1 Long Question

You are a Data Scientist at eSewa, Nepal’s leading digital payment platform.

• Identify two high - impact areas where unsupervised learning could add significant business value.

• For each area:

1. Define the problem clearly (e.g., customer behavior clustering, fraud pattern detection).
2. Propose a specific unsupervised learning approach (e.g., clustering, anomaly detection, dimensionality reduction) and recommend one or more algorithms (e.g., K-Means, DBSCAN, Autoencoders).
3. Briefly explain how the outputs from these models can be integrated into eSewa’s products or services to drive business decisions.

Answer:

As a Data Scientist at e-Sewa, I mostly explore how unsupervised learning can bring strategic value to our digital services. Two key areas where I find significant impact are user behavior segmentation and fraud detection.

1. User Behavior Segmentation:

Understanding how customers interact with our platform is essential for personalization and engagement. I analyze transaction histories covering factors like usage frequency, average transaction size, and preferred services (e.g., mobile top ups, utility payments). I apply clustering methods such as K Means and DBSCAN to group users with similar patterns. For instance, I can identify a segment of low activity users who only use e-Sewa to pay electricity bills. Based on this insight, we can target them with promotional credits or suggest additional features like ticket booking. On the other hand, high frequency users can be prioritized for loyalty rewards. These segments feed directly into our marketing pipelines, enabling more relevant communication and offers.

2. Anomaly Detection for Fraud:

Traditional rule-based systems often fall short in identifying new or subtle fraudulent behaviors. Since most fraud cases are not labeled in advance, I use unsupervised anomaly detection techniques like Autoencoders and Isolation Forests. These models learn what typical user activity looks like and flag unusual behaviors such as rapid, high-value transactions or access from different devices within a short span. Once detected, these anomalies trigger alerts or automated safeguards, such as temporary account holds or transaction verification prompts.

I integrate both model outputs into e-Sewa's product ecosystem. The user clusters enhance our CRM and marketing dashboards, while fraud alerts support our real time security systems. By integrating unsupervised learning techniques into decision pipelines, I help improve customer retention, optimize promotional spending, and strengthen platform security resulting toward building a smarter and safer digital payment experience for our users.

2 Short Question

2.1. Overfitting is a common challenge in deep learning models.

• Describe at least two techniques commonly used to prevent or reduce overfitting (e.g, dropout, early stopping, data augmentation).

• For each method:

1. Briefly explain how it works.
2. Provide a practical example of how it would be applied in a real-world deep learning project (e.g., image classification, sentiment analysis).

Answer:

Overfitting appears when a model learns the training data with high accuracy, including its noise or outliers, which results in poor performance on new data. To prevent this, we mostly used two effective techniques. They are dropout and data augmentation.

1. Dropout

Dropout is a technique that helps prevent overfitting by randomly turning off some neurons during training. This makes the model learn in different ways and stops it from relying too much on specific features. For example, in a sentiment analysis project using an LSTM model, applying dropout to the dense layers helps the model learn overall emotional tone instead of memorizing exact words or phrases.

2. Data Augmentation

Data augmentation increases the amount of training data by slightly changing the original data by the means of rotating, flipping, or changing the brightness of pictures to create new examples. For example, if I'm building a model to recognize Nepali currency notes, I can use images with different lighting, angles, or even partly covered notes. This makes the model more reliable when used in real situations.

In conclusion, both methods reduce overfitting by improving the model's ability to generalize data. This ensures the final model performs well not just during training, but also in practical, real-world applications.

2.2. Difference between CNN and RNN:

Explain the fundamental differences between a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN).

1. Provide examples of scenarios where one would be more suitable than the other (e.g., image recognition vs. time-series prediction).
2. Briefly discuss common challenges faced during the training of deep learning models (e.g., vanishing gradients, overfitting).
3. Provide possible solutions or techniques to address these challenges (e.g., batch normalization, early stopping).

Answer:

Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are different deep learning architectures developed to handle different types of data. We require proper understanding of their strengths to choose the right architecture for relevant tasks.

CNNs are ideal for handling spatial data, especially images. They extract features like edges, textures, and shapes through convolutional filters. For example, in an image related task like verifying user documents during KYC at banking sectors, a CNN model can accurately recognize document types and detect quality issues like blurriness or missing fields.

On the other hand, RNNs best fits for sequential data. They maintain a memory of past inputs, which is useful for time series predictions or text processing. For example, we can use RNNs in tasks like analyzing user transaction sequences to predict future behavior or building smart chatbots that remember conversation context.

Training both networks comes with challenges. RNNs, especially deeper ones, often suffer from vanishing gradients, where long-term dependencies are hard to capture. I mitigate this by using LSTM or GRU units, which are developed to save memory across long sequences. CNNs mostly faces overfitting problems on small datasets, so I apply early stopping and batch normalization to maintain generalization and stabilize training.