UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Artificial Intelligence and Machine Learning (6CS012)

Assignment-III

University ID: 2358554

Name: Sabin Chaulagain

Group: L6CG7

Module Leader: Mr. Siman Giri

Module Tutor: Ms. Durga

Submission Date: 20th May 2025

Contents

1. Long Question

1.1. 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:**

  Define the problem clearly (e.g., customer behavior clustering, fraud pattern detection).

  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).

  Briefly explain how the outputs from these models can be integrated into eSewa's products or services to drive business decisions.

Answer: Currently, online payment is an exponentially increasing business. People throughout the world now rely on e-wallets and e-banking as an alternative to cash. Mobile banking, Khalti, eSewa etc., other facilities, are available in Nepal which have made money transfer easier, bill payment and online shopping easier. Expanding use of these sites and apps means their behavior must be researched and transactions are required to be safe. Unsupervised learning can be useful for these aspects as it could detect secrets in data without labeled examples.

1. Grouping Customers by Behavior

Problem:

eSewa is used differently by people. Some use them every day, while others use them in intervals." However, the same offers and messages came from eSewa for all. If we can know which habits different people who use eSewa have, then we will be in a position to bring better and more personalized service to them.

Solution:

We can utilize a method called clustering to segment users based on their usage. For example, we can segment frequent bill payers or users who mostly send money. An easy method to segment like this is by K-Means. If we want to determine people who behave very differently compared to other users, we can use DBSCAN.

How it Helps:

After we learn what kind of user each of them are, we will be able to send them messages depending on their needs. For example:

- Frequent users of eSewa can be offered exclusive rewards.
- Those who haven't used it for a long time can be sent some reminders, or discounts.
- This keeps the users happy and makes eSewa popular.

1.2. Finding Suspicious Transactions:

Problem:

eSewa processes thousands of transactions on a daily basis. Sometimes fraudulent or suspicious transactions get tripped. For example:

- Someone who's transferring the same small amounts repeatedly.
- A person who is doing a giant sale at unbelievably late hours.

Fixed rules alone are difficult to use to identify these.

Solution:

We can employ an anomaly detection method. It allows detecting transactions which appear to be unusual relative to the usual ones. Two good tools to this end are:

- Isolation Forest
- Autoencoders

They get a picture of what "normal" is and then tag anything which looks irregular.

How it Helps:

In case of suspicious transactions, we can:

- Hold if for verification.
- Or flag the risk team.

This protects the users and protects eSewa from fraud.

1.1. You are a Machine Learning Engineer at a growing e-commerce company preparing to implement and scale machine learning system.

List and explain at list three real-world challenges you expect during ML model development, deployment, or maintenance (e.g., data drift, imbalanced data, system latency).

For each challenge:

Discuss potential consequences if not properly addressed.

Propose technical or organizational solutions you would implement (e.g. retraining pipelines, feature monitoring, distributed serving, MLOps practices).

Finally, reflect on how cross-functional collaboration (between data scientists, engineers, product teams) can help mitigate these challenges more effectively.

Answer: As a Machine Learning Engineer in a scaling e-commerce company these are the real life challenges in building and deploying ML models:

1. Data Drift Explanation: Down the road, the situation may change and there may be product trends, or the customer behavior may change, and the model can have data different from that, which it has learned from. Consequences: This may diminish predictability, and this way poor recommendations or inappropriate pricing can result. Solution: Set up regular retrain of automated models with new data pipelines along with data drifts detection.
2. Imbalanced Data: In e-commerce, there are classes, and they are not very dense (e. g. high-value transactions, refunds). This may lead to low model performance on those instances that are important.

   Impacts: Models are not able to detect fraud or are not capable of detecting important segments of users.

   Solution: Employ such methods as SMOTE, class weighting, or anomaly detection and periodically inspect the labeling approaches for data to handle the imbalance.

3. System Latency:

- Explanation: Sluggish or cumbersome models can cause responses to be slow, particularly when used in real-time behaviors such as search or recommendation.
- Consequences: Degraded user experience, cart abandonment; or decreased sales.
- Solution: Quantize the model (model quantization, distillation) or cache the model at the edge, or use distributed model serving to reduce latency.

Role of Cross-Functional Teamwork

Installation work with the product teams, software engineers, and data scientists defines the actual business needs, defines the right metrics, develops scalable user-friendly systems.

Product teams prioritize, engineers make the system run and data scientists make the model stay into shape – that's what makes ML more efficient and business objective-minded.

Shared goals and clear communication between teams as great necessities to generate stable and scalable ML systems.

2. Short Question:

2.1. Define and differentiate between overfitting and underfitting.

Answer: Overfitting occurs when the model learns training data in too many details, even noise, and very tiny errors. This is causing high variance (performance has huge variance between train data and test data). But underfitting occurs when a model undercomplete itself and is unable to detect the pattern of the data. Underfitting enables us to have high bias and low precision.

Why Overfitting and Underfitting are a problem:

Overfitting provides very high accuracy over the training data, but low accuracy on new (test) data. The model cannot generalize well.

Underfitting means low accuracy on training and testing sets since underfitting does not have the ability to even learn the data that it sees.

Example:

Overfitting: A model fits a wiggly curve that goes through all the points in the set of training (high variance, low generalization).

Underfitting: A model maps a linear trend for a curved pattern (high bias, does not reflect the trend).

Questions: Overfitting is a common challenge in deep learning models.

Two common techniques to reduce overfitting are Dropout and Data Augmentation:

1. Dropout:

How it works: Dropout randomly kills (or "turns off") some of the neurons during learning. This pushes the model away from being excessive in some of its routes and makes it learn more generic routes.

Example: If there exists a dropout in an image classification model, it will come after dense layers. When the model begins to overfit for training images, dropout positions like (0. 5 rate) mitigates this by making the model less susceptible to certain characteristics.

2. Data Augmentation”3

How it works: It generates new training instances of minimally processed original data — such as rotating, flipping, or zooming images. That assists model learning more generalization.

Example: For a deep learning flower model classification, we can rotate, crop or brighten the images to obtain more diversified samples. This discourages the model from remembering image details and leads to its better generalization of new images.

2.2. Neural Network Architecture

Definitions:

CNN (Convolutional Neural Network): Applied for the processing of grid-like data, i.e., images. It employs convolutional layers to use pattern detection (automatically: edges, shapes).

RNN (Recurrent Neural Network): Suitable for sequential data, such as, text or time series. It has loops and can remember previous inputs, thus helpful for context or memory-based task.

Use Cases:

CNN is better suitable for such use cases as image classification (handwritten digit recognition or object detection in image).

RNN is better for time series prediction, speech recognition, or sentiment analysis in text in which data sequence is meaningful.

Challenges in Deep Learning:

Vanishing Gradients: Gradients are very small to adjust weights with any sense in the deep RNNs.

Solution: Use architectures such as LSTM or GRU which are better at long term memory.

Overfitting: Model learns by heart the training data. However, it cannot generalize well to novel data.

Solution: Get improvement to generalization by resorting to early stopping dropout or data augmentation.

Hence, both CNNs and RNNs are excellent, yet applicable to different kinds of data and jobs. That is the secret to creating good models – using the right one and solving training problems.

Questions: Difference Between Neural Network and Autoencoders:

Answers:

Normal Neural Networks are usually applied for the purpose of classification or regression task. They are given input data, and they attempt to forecast a label or value through output layers.

Autoencoder is one of the neural networks that is used to learn compressed forms of data. They have two main parts:

Encoder: Compresses the input to a latent space.

Decoder: Attempts to reconstruct the original input from such a compressed form.

Example: Imaging compression for instance, it is possible to make an image of a high level of resolution reduce to a few features and restore it again with the help of autoencoder.

Applications:

Anomaly Detection: Train autoencoders on normal data. When an input cannot be reconstructed well, it is an anomaly.

Example: In fraud detection, an unusual transaction will be poorly reconstructed by the autoencoder, suggesting potential fraud.

Autoencoders help to reduce noise, detect outliers, and compress data, so they can be used in other machine learning applications besides prediction.