6CS012 (AI and ML, ASSIGNMENT III)

Name: Bhojraj Adhikari
ID: 2332281 Group: L6G12

1. Long Question (2)

As a Machine Learning Engineer in an e-commerce company that is expanding, deploying and scaling ML systems comes with some real-world challenges:

1. Data Drift:
Factors such as customer behavior, product trends, or other external factors (e.g., seasonality, economic market trends) can lead to changes in the data distribution over time, referred to as data drift.

- Consequences*:* If left unhandled, model predictions become incorrect, resulting in bad product recommendations, lower click-through rates, and fewer sales. This also erodes user trust.
- Solution*:* Utilize statistical tests and performance monitoring to automate the detection of concept and data drift. Configure a retraining pipeline that reacts to drift events or schedules to allow your model to stay fresh.

2. Imbalanced Data:
In e-commerce, tasks such as fraud detection or application of a rare action (e.g., high-value returns) by a customer have classes underrepresented in the dataset.

- Consequences*:* Models can be biased towards the majority classes and perform poorly in terms of recall for the most crucial minority cases (like missing fraudulent transactions).
- Solution*:* Resample the data (SMOTE or undersampling), consider assigning class weights during the model training, and optimize on F1-score, precision-recall curve or AUC instead of simple accuracy.

3. System Latency During Inference:
Online marketplaces need to perform online (near) real-time model inferences for features such as personalized recommendations, dynamic pricing, or fraud detection.

- Consequences*:* High latency can result in a bad user experience, empty carts, and missed revenue.
- Solution*:* Design models to have a small footprint (for example, lightweight models like XGBoost or DistilBERT) and serving optimizations and strategies, for example TensorFlow Serving, caching or batch processing.

Interdepartmental cooperation is necessary to counterbalance these challenges. Data scientists will make sure the correct modeling decisions are made while engineers ensure that the deployment and monitoring pipelines are optimized and the product teams make sure the business context is understood. Frequent aligning meetings, common dashboards and clear closed feedback loops all help keep your ML systems robust, scalable and aligned with company goals.

2. Short Question(1)

Overfitting:

Overfitting and underfitting are prevalent issues in machine learning that reflect how well a model generalizes the tendencies of the data.

- **Underfitting,** happens when a model is not complex enough to learn the structure of the data. It suffers from high bias and a poor fit to the training and testing data. For instance, a linear regression on non-linear data will underfit due to the inability to express the complicated relationship.
- **Overfitting** occurs when a model is overly complex and learns not just the patterns in the data but the noise in the training data. Because of this, it does well on the training set then delivers poor predictions on new data. For example, if you have a decision tree that is too deep, it may simply memorize all of the training examples rather than be able to generalize to new data.

Why they are problematic:

- **Underfitting** results in inaccurate predictions and is unreasonably costly.
- **Overfitting** yields bad generalization: the model works very poorly in reality.

Solutions:

- In order to decrease underfitting either increase the model complexity or include more relevant features.
- To avoid over fitting use regularization, decrease complexity of model or use cross-validation, early stopping.

3. Neural Network Architecture(1)

Introduction to CNN and RNN CNN and RNN are two different types of deep learning architectures built for processing diverse neural inputs.

- **CNNs** are specifically designed for spatial data (like images). These layers are specialized in detecting patterns such as lines, shapes, textures, etc. CNNs are well-suited for tasks such as image classification, object detection, and facial recognition.

- **RNNs** are created to work with sequential or time based data which does not make sense if processed out of order. They have loops which allow information to be carried across time steps, and so are well suited to tasks such as language modelling, speech recognition and time series forecasting.

Difference is CNN's speciallity is to extract spatial features and RNN's speciallity is to capture temporal dependencies.

Common training challenges:

- **Gradients vanishing**: Gradients can be very small in deep networks, like RNNs (especially with sigmoids or tanh), when you backpropagate, this stops learning or slows it down.
- Solution: Use architectures such as LSTM or GRU and employ gradient clipping.
- **Overfitting**: The model is good at fitting training data, but not good for testing.
- Solution: The model can be prevented from overfitting by using dropout, early stopping, and data augmentation.
- **Training instability**: This is particularly problematic with deep networks.
- Solution: Use batch normalization, or adjust your learning rates.