DSAI SET 7

EURON

1. Scenario: Fraud Detection in Banking

Question: You are working for a bank that wants to build a fraud detection system. You have imbalanced data (99% non-fraud, 1% fraud). How would you handle this and build a robust model?

Dealing with imbalanced data in fraud detection is a common challenge. Since the majority class dominates, a model trained directly on such data would tend to predict non-fraud most of the time, leading to poor recall for fraud cases. Here's a structured approach:

1. **Data Resampling Techniques**
    - **Oversampling Minority Class:** Use **SMOTE (Synthetic Minority Over-sampling Technique)** to synthetically generate fraud cases.
    - **Undersampling Majority Class:** Reduce the number of non-fraud cases to balance the dataset.
    - **Hybrid Sampling:** A combination of oversampling and undersampling.
2. **Use Anomaly Detection Methods**
    - Fraud cases are rare, making it suitable for anomaly detection algorithms like **Isolation Forests**, **One-Class SVM**, or **Autoencoders**.
3. **Adjust Model's Loss Function**
    - Use **weighted loss function** in models like Random Forest, XGBoost, or Neural Networks to penalize misclassification of fraud cases more.
4. **Evaluation Metrics**
    - Use **Precision-Recall Curve, F1-score, and AUC-ROC** rather than accuracy, as accuracy would be misleading in imbalanced settings.
5. **Feature Engineering**
    - Aggregate transaction history, frequency of high-value transactions, and customer behavior features for better fraud identification.

2. Scenario: Predicting House Prices with Limited Data

Question: A real estate company wants to predict house prices but only has 500 samples. How would you build an effective model?

With limited data, the risk of overfitting is high. Here's a strategic approach:

1. **Data Augmentation**
    - **Generate Synthetic Data:** Use techniques like **bootstrapping** or synthetic data generation via GANs.
2. **Feature Engineering**
    - Use domain knowledge to create better features, such as **price per square foot**, **distance to major roads**, **crime rates**, etc.
3. **Use Simpler Models**
    - Instead of complex models, opt for **Regularized Regression (Ridge/Lasso)** or **Decision Trees with Pruning**.
4. **Leverage Pre-trained Models**

  - If open-source datasets exist, fine-tune a pre-trained model instead of training from scratch.
5. **Cross-validation**
   - Use **K-Fold cross-validation** to maximize model performance with limited data.

---

3. Scenario: Preventing Overfitting in a CNN Model

Question: You are training a CNN for image classification, but your training accuracy is 99% while test accuracy is only 75%. What would you do?

This is a classic case of **overfitting**. Solutions include:

1. **Data Augmentation**
   - Use **rotation, flipping, scaling, cropping, color jittering** to generate more training samples.
2. **Regularization Techniques**
   - **Dropout Layers:** Randomly drop neurons to prevent reliance on specific features.
   - **L2 Regularization (Weight Decay):** Adds a penalty to large weights to reduce overfitting.
3. **Batch Normalization**
   - Helps in stabilizing learning and reducing overfitting.
4. **Early Stopping**
   - Monitor validation loss and stop training when it starts increasing.
5. **Use Transfer Learning**
   - If the dataset is small, fine-tune a **pre-trained model** (like ResNet, EfficientNet) instead of training from scratch.

---

4. Scenario: Face Recognition in Low Light Conditions

Question: You are building a facial recognition system, but images taken in low light have poor performance. How do you improve it?

1. **Enhancing Image Quality**
   - Use **histogram equalization** or **CLAHE (Contrast Limited Adaptive Histogram Equalization)** to improve contrast.
   - Apply **GANs (Pix2Pix, CycleGAN)** to generate clearer versions of low-light images.
2. **Data Augmentation**
   - Train on images with artificial **brightness variations** to improve generalization.

3. **Use Robust Features**
    - Extract **Edge-based Features (SIFT, ORB, HOG)** instead of pixel intensities.
4. **Use IR/Depth Data**
    - Combine **Infrared (IR) and Depth cameras** to detect facial structures even in darkness.

---

5. Scenario: Sentiment Analysis with Sarcasm Detection

Question: Your sentiment analysis model misclassifies sarcastic tweets. How would you fix it?

1. **Use Context-Aware Embeddings**
    - Word2Vec and TF-IDF fail for sarcasm. Use **BERT**, **RoBERTa**, or **GPT-based models**.
2. **Add Contextual Features**
    - Extract **emoji usage, sentence polarity shifts**, and **user intent** for better detection.
3. **Multi-Task Learning**
    - Train sentiment analysis along with sarcasm detection using **multi-head models**.

---

6. Scenario: Hallucination in a LLM Chatbot

Question: Your chatbot generates false but confident-sounding responses. How do you prevent this?

1. **Reinforcement Learning with Human Feedback (RLHF)**
    - Fine-tune responses based on **human feedback**.
2. **Fact-Checking Mechanisms**
    - Integrate a **retrieval-augmented generation (RAG)** pipeline with external databases.
3. **Reduce Temperature in Sampling**
    - Lowering **temperature** during inference reduces randomness.

---

7. Scenario: Bias in a Generative AI Model

Question: Your text generation model exhibits racial or gender bias. What would you do?

1. **Bias Detection & Debiasing Techniques**
    - Use **SHAP, LIME** for interpretability.
    - Train on **fair and diverse datasets**.
2. **Fairness-Aware Training**
    - Use adversarial debiasing models like **FairGAN**.

---

8. Scenario: Real-time Object Detection

Question: Your self-driving car needs real-time pedestrian detection. Which model and techniques would you use?

1. **Use Lightweight Models**
    - YOLOv8, MobileNet SSD for fast inference.
2. **Edge Computing**
    - Deploy on **Nvidia Jetson** or **Coral Edge TPU**.

---

9. Scenario: Zero-shot Learning for Image Classification

Question: How would you classify images of unseen categories?

1. **Use CLIP (Contrastive Language-Image Pretraining)**
    - Maps images and text into a shared latent space.

---

10. Scenario: Personalizing Recommendations using Gen AI

Question: How would you build a recommendation system using Gen AI?

1. **Fine-tune Transformer Models**
    - Use **LLMs like GPT-4** to generate personalized responses.
2. **Use Reinforcement Learning**
    - **Multi-armed bandit algorithms** optimize recommendations.

11. ML Scenario: Loan Approval System (Overfitting & Regularization)

Scenario:
You are building a **loan approval system** using machine learning. After training, your model performs **extremely well on training data (98% accuracy)** but drops to **75% on test data.** What is happening, and how can you fix it?

Your model is likely **overfitting**, meaning it has memorized the training data instead of learning **generalizable patterns**. To fix this:

1. **Regularization (L1/L2)**: Apply **L2 regularization (Ridge regression)** to prevent large weight values.
2. **Reduce Complexity**: Use feature selection, PCA, or a simpler model (e.g., logistic regression instead of deep networks).
3. **Increase Training Data**: Collect more diverse data to help the model generalize better.
4. **Cross-validation**: Implement **k-fold cross-validation** to ensure consistent performance.

---

12. DL Scenario: Self-Driving Cars (Vanishing Gradient Problem)

Scenario:
You are developing a **deep learning model for a self-driving car** that uses a CNN-based architecture to recognize road signs. During training, deeper layers of the network **fail to update their weights** properly. What could be the issue?

This sounds like the **vanishing gradient problem**, where gradients become **too small** as they propagate backward, preventing effective weight updates.

Solutions:

1. **Use ReLU instead of Sigmoid/Tanh**: ReLU prevents gradient shrinkage by keeping non-zero gradients.
2. **Batch Normalization**: Normalizes activations to ensure stable gradients.
3. **Residual Connections (ResNets)**: Helps gradients flow smoothly across deep layers.

---

13. CV Scenario: Detecting Fake Images (GANs & Deepfakes)

Scenario:
You work in cybersecurity, and your task is to build a system that detects **AI-generated fake images (deepfakes).** How would you approach this?

1. **Train a Discriminator Network**: Use a pre-trained CNN (ResNet or EfficientNet) to classify real vs. fake images.
2. **Analyze Artifacts**: AI-generated images often have inconsistencies in lighting, unnatural textures, or blurry edges.
3. **Use Frequency Analysis**: GANs struggle to generate fine-grained high-frequency details. A Fourier Transform can detect these discrepancies.
4. **Train on GAN-Generated Data**: Expose the model to various GAN-generated images to improve robustness.

---

14. NLP Scenario: Sentiment Analysis (Imbalanced Data Issue)

Scenario:
You are working on a **customer review sentiment analysis** model. Your dataset is **highly imbalanced** (95% positive, 5% negative reviews). How do you handle this?

1. **Data Augmentation**: Generate synthetic negative reviews using back-translation (translating to another language and back).
2. **Resampling**: Use **oversampling (SMOTE)** for minority class or **undersampling** the majority class.
3. **Weighted Loss Function**: Penalize misclassifications of negative reviews more heavily.
4. **F1-Score Over Accuracy**: Since accuracy is misleading in imbalanced datasets, use **F1-score, precision-recall curves**.

---

15. GenAI Scenario: AI Art Generation (Bias in AI Models)

Scenario:
You built a **DALL·E-like image generation model**, but users report that the model generates **stereotypical or biased outputs** when given prompts about people. How do you address this?

1. **Dataset Curation**: Remove biased data and ensure diverse representation.
2. **Fairness Constraints**: Implement fairness-aware training by adjusting loss functions to balance different demographics.

3. **Human-in-the-Loop (HITL) Moderation**: Add user feedback mechanisms to detect and correct biases.
4. **Post-Processing Filtering**: Apply **content moderation** to filter biased outputs.

---

16. ML Scenario: Fraud Detection System (Class Imbalance & Cost-Sensitive Learning)

Scenario:
You are designing a **fraud detection model for a bank**, but fraud cases are **extremely rare (0.1%)**. How would you ensure the model detects fraud effectively?

1. **Anomaly Detection**: Use **Isolation Forests or Autoencoders** to detect outliers.
2. **Cost-Sensitive Learning**: Assign higher penalties to misclassified fraud cases.
3. **Hybrid Models**: Combine supervised learning with **unsupervised anomaly detection.**
4. **Resampling Techniques**: Use **SMOTE** to generate synthetic fraud cases.

---

17. DL Scenario: Training Chatbots (Seq2Seq & Attention Mechanisms)

Scenario:
You are building a **customer support chatbot**, but it often forgets the conversation context. What improvements can you make?

1. **Use Transformers (BERT/GPT)**: Unlike RNNs, transformers maintain long-range dependencies.
2. **Implement Attention Mechanism**: Ensures the model focuses on relevant parts of the conversation.
3. **Fine-tune on Context-Rich Data**: Train on real conversations where user context persists.

---

18. CV Scenario: Medical Image Diagnosis (Explainability & AI Bias)

Scenario:
Your AI model predicts lung cancer from CT scans, but doctors **distrust the model's decision-making process.** How do you improve explainability?

1. **Use SHAP/LIME**: Visualize which pixels influenced predictions.
2. **Grad-CAM**: Highlights important image regions the model focused on.
3. **Model Transparency**: Use ensemble models with interpretable architectures.

---

19. NLP Scenario: Document Summarization (Extractive vs. Abstractive Summarization)

Scenario:
You are tasked with **automatically summarizing news articles.** Should you use an **extractive or abstractive** approach?

- **Extractive Summarization** selects key sentences verbatim (e.g., TextRank).
- **Abstractive Summarization** generates new text (e.g., BART, T5).
- **Best Choice**: Abstractive is ideal for human-like summaries but needs large training data.

---

20. GenAI Scenario: Music Generation AI (Evaluating AI Creativity)

Scenario:
Your team developed an AI that generates music. How do you evaluate the **quality of generated music?**

1. **Human Evaluations**: Conduct listening tests.
2. **Quantitative Metrics**: Use **tonal stability, rhythm coherence,** and **chord progression validity.**
3. **Diversity & Novelty Metrics**: Ensure it doesn't just copy training data.

21-15: Quick Scenarios & Answers

11. **ML – Recommender Systems**: How do you handle the **cold start problem** in a new recommendation system?

- Use **content-based filtering** for new users & items.

12. **DL – Autonomous Drones**: Why does your **drone navigation model fail in foggy conditions?**

- It was trained on clear-weather data only → **use data augmentation**.

13. **CV – Object Detection Failures**: Your self-driving car model misclassifies **pedestrians at night. Why?**

- Model lacks night-time training data → **collect more diverse datasets**.

14. **NLP – Spam Detection**: Your spam filter incorrectly flags business emails as spam. How do you fix it?

- Train on **more diverse email samples** & use **context-aware embeddings**.

15. **GenAI – AI Story Writing**: How do you prevent a **text generator from producing repetitive outputs?**

- Use **top-k sampling & temperature scaling** to ensure diverse responses.