Task 7: Recommendation System and Advanced Intelligent Systems

Part-1

Ques-1: What is Azure OpenAI?

Ans-1:

Azure OpenAI is a service that integrates OpenAI's AI language models and services into Microsoft Azure applications and platforms. It provides access to large-scale, generative AI models that can be used for a variety of use cases, such as: writing assistance, code generation, reasoning over data, creating chatbots and summarizing text.

Few uses cases that are being used by different domain industries:

- Finance: AI models can be used to automate stock market trends or for fraud detection.
- Healthcare: Medical professionals can use NLP and ML to diagnose diseases more accurately. It can help in analyzing medical records, images to assist healthcare professionals in diagnosing diseases and cure.
- Education: Different educational institutions are using these services to create educational contest that adapts as per student's learning style and making more improvement over content as per the student's feedback.
- Customer Support: In today's world many companies, organization have created there own chatbots or virtual assistants that can handle customer queries and can help in troubleshooting issues.
- Manufacturing: Azure's IoT services combined with machine learning algorithms are used to predict equipment failures and schedule maintenance activities proactively, reducing downtime and optimizing production efficiency in manufacturing facilities.

Ques-2: What is Tokenizer?

Ans-2:

Tokenization is a preprocessing technique in natural language processing that breaks down sequence of text into smaller units called as tokens. Tokens can be as small as characters or as long as words. These tokens are the basic building blocks for NLP tasks such as text classification, named entity recognition, sentiment analysis and more.

Ex: Let input be: "Yatharth you are permitted to go ahead."

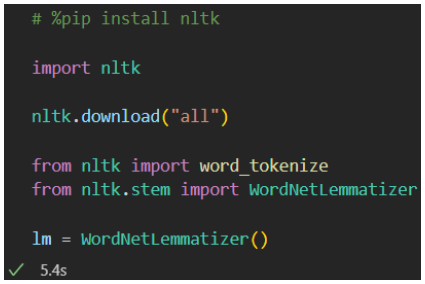
Tokenized as words like: ['Yatharth', 'you', 'are', 'permitted', 'to', 'go', 'ahead']

Few tools that can be used for tokenization:

- Natural Language Toolkit (NLTK): An open-source python library for NLP.
- Keras: Open-source library that provides a python interface for artificial neural networks and used as a layer for TensorFlow library.
- spaCy
- Lemmatization

Example of using tokenizer using NLTK:

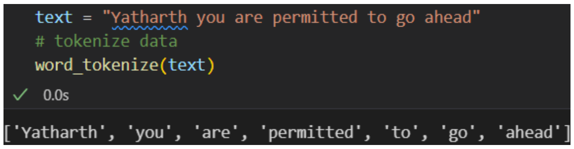
- Importing NLTK libraries



```
# %pip install nltk

import nltk

nltk.download("all")

from nltk import word_tokenize
from nltk.stem import WordNetLemmatizer

lm = WordNetLemmatizer()
```

```
✓ 5.4s
```

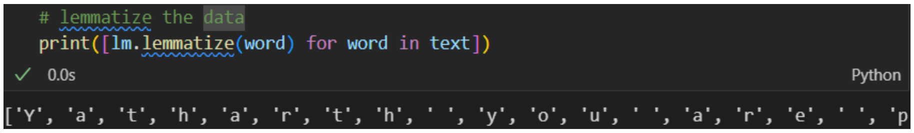
- Tokenizing text



```
text = "Yatharth you are permitted to go ahead"
# tokenize data
word_tokenize(text)
```

```
✓ 0.0s
['Yatharth', 'you', 'are', 'permitted', 'to', 'go', 'ahead']
```

- Lemmatizing the text



```
# lemmatize the data
print([lm.lemmatize(word) for word in text])
```

```
✓ 0.0s                                                                    Python
['Y', 'a', 't', 'h', 'a', 'r', 't', 'h', ' ', 'y', 'o', 'u', ' ', 'a', 'r', 'e', ' ', 'p
```

Ques-3: What is few-shot and Zero-shot Learning and their advantages on LLMs and prompting?

Ans-3:

Few-shot Learning

Few shot learning refers to a technique where we provide the Large Language Models (LLM) with a small set of examples to guide it towards a specific task or style. In simple language, it involves fine-tuning a pre-trained model with limited number of examples. When this learning is combined with prompting techniques, it enhances the capabilities of LLMs by enabling them to generalize, adapt and perform a wide range of tasks with minimal training data.

Advantages:

- Rapidly adapt to new tasks or domains by providing a few examples as prompts.
- Enables users to fine tune the model for specific application/domain without need of extensive labeled datasets.
- It is more efficient and cost-effective.
- Excel at quickly adapting to new tasks or classes with minimal labeled examples.

Zero-shot Learning

Zero shot learning enables LLMs to tackle tasks they weren't explicitly trained for, relying solely on their pre-existing knowledge and general language understanding. In simple language, model can classify/predict things that were not present in the training data. It empowers LLMs to generalize and adapt to new tasks without explicit training data, providing versatility, efficiency and scalability.

Advantages:

- Understand and generate text for a wide range of topics or concepts, even if they were not explicitly seen during training.
- Allows LLMs to continuously learn and adapt to new tasks or concepts over time by incorporating new prompts or descriptions during inference.
- Reduces the need for extensive labeled training data for each specific task or concept.

Both few-shot learning and zero-shot learning improve the abilities of language models like LLMs. When used together with prompting, which means giving the model specific instructions or examples to help it, they empower LLMs to handle many different tasks accurately and efficiently. This is true even when there isn't much training data available or when facing new tasks. These techniques help LLMs become more adaptable and versatile, which makes them incredibly valuable for all kinds of tasks involving natural language processing.

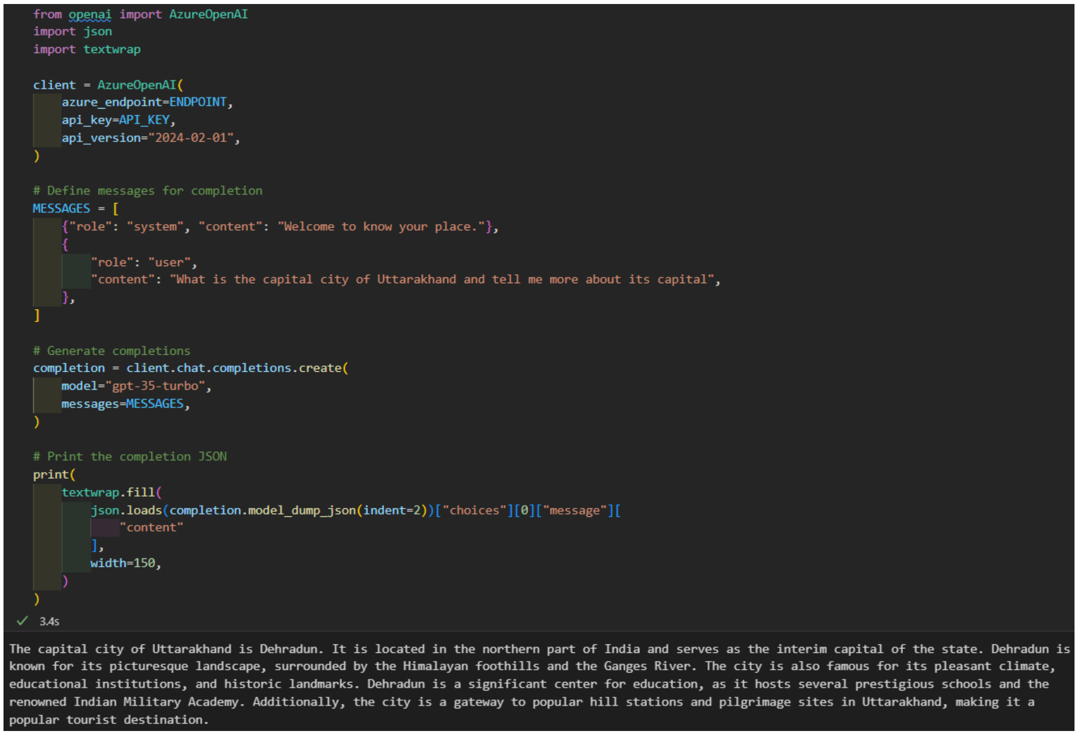
Ques-4: What is the difference between System Prompt and meta prompt? Provide an example.

Ans-4:

In large language models (LLMs), both system prompts and meta prompts are used to guide its response for text generation.

- System Prompt:
  System prompt is a set of instructions that defines the AI's role, capabilities, limitations and expected response format. It is also known as task prompt or input prompt. These are used in various NLP applications to specify the desired behavior or context for the model's output. Examples of system prompt:
  - Text completion.
  - Text Generation
  - Act as virtual assistant
- Meta Prompt:
  Meta prompt is a higher-level instruction that guides the AI on how to approach user queries. It is also known as a meta-task prompt or global prompt. These prompts are mre abstract and general, focusing on shaping the overall behavior of the model. Examples of meta prompt:
  - Story telling
  - Creative text generation
  - Prioritize user safety in all interactions.
  - Continuously learn from user feedback to improve responses.

Question Answering example of prompt:



```python
from openai import AzureOpenAI
import json
import textwrap

client = AzureOpenAI(
    azure_endpoint=ENDPOINT,
    api_key=API_KEY,
    api_version="2024-02-01",
)

# Define messages for completion
MESSAGES = [
    {"role": "system", "content": "Welcome to know your place."},
    {
        "role": "user",
        "content": "What is the capital city of Uttarakhand and tell me more about its capital",
    },
]

# Generate completions
completion = client.chat.completions.create(
    model="gpt-35-turbo",
    messages=MESSAGES,
)

# Print the completion JSON
print(
    textwrap.fill(
        json.loads(completion.model_dump_json(indent=2))["choices"][0]["message"][
            "content"
        ],
        width=150,
    )
)
```

✓ 3.4s

```
The capital city of Uttarakhand is Dehradun. It is located in the northern part of India and serves as the interim capital of the state. Dehradun is
known for its picturesque landscape, surrounded by the Himalayan foothills and the Ganges River. The city is also famous for its pleasant climate,
educational institutions, and historic landmarks. Dehradun is a significant center for education, as it hosts several prestigious schools and the
renowned Indian Military Academy. Additionally, the city is a gateway to popular hill stations and pilgrimage sites in Uttarakhand, making it a
popular tourist destination.
```

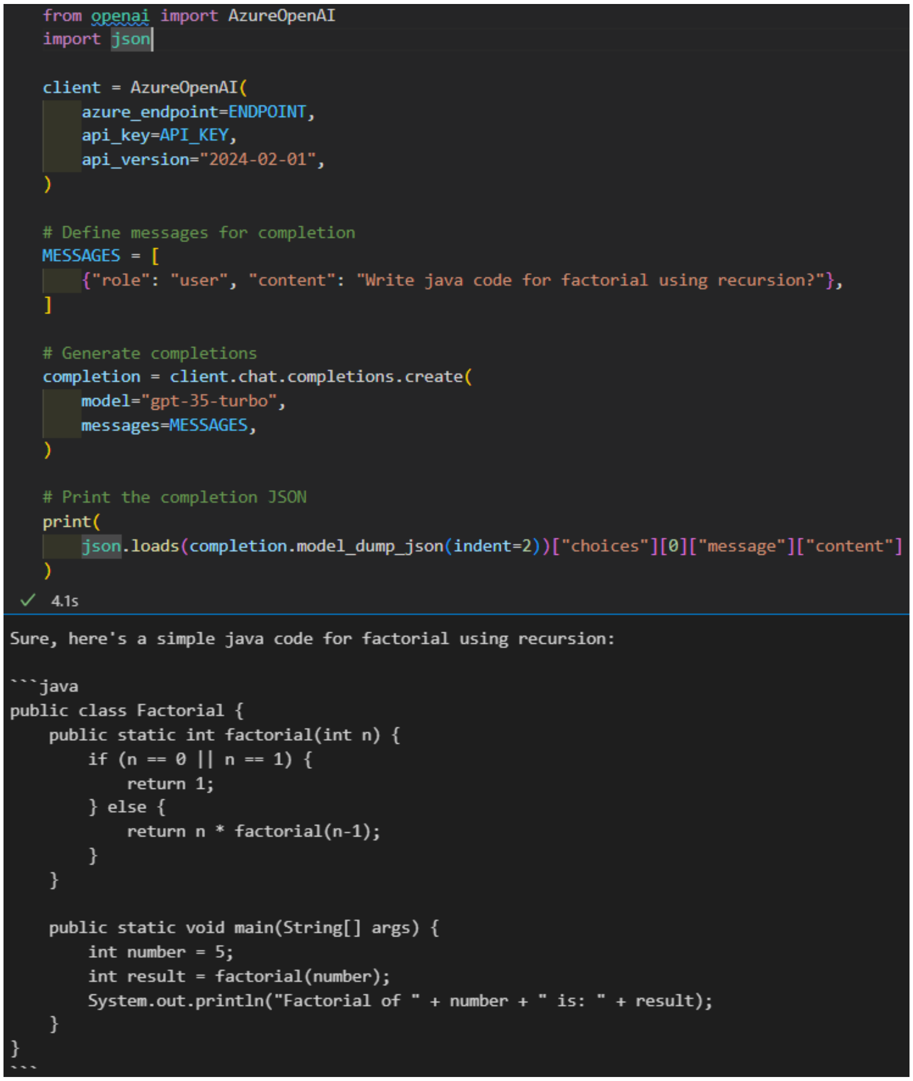
Ques-5: Explain generate the code with Azure OpenAI service? What's the advantage of using this service?

Ans-5:

Azure OpenAI services enables developer to generate and improve existing programming code in various languages to increase efficiency and understanding. It utilizes OpenAI's language models, such as Generative Pre-trained Transformer (GPT). User need to provide the statement with its requirement and task for code generation and our service will work over that. As we can see the example where this service has generated code for factorial code with recursion in java.

To use this service, we need to adhere to these settings:

- Import AzureOpenAI and create a client using endpoint and api_key of OpenAI.
- Provide the prompt to the client on which client will work and generate the required output.



```python
from openai import AzureOpenAI
import json

client = AzureOpenAI(
    azure_endpoint=ENDPOINT,
    api_key=API_KEY,
    api_version="2024-02-01",
)

# Define messages for completion
MESSAGES = [
    {"role": "user", "content": "Write java code for factorial using recursion?"},
]

# Generate completions
completion = client.chat.completions.create(
    model="gpt-35-turbo",
    messages=MESSAGES,
)

# Print the completion JSON
print(
    json.loads(completion.model_dump_json(indent=2))["choices"][0]["message"]["content"]
)
```

✓ 4.1s

~~~
Sure, here's a simple java code for factorial using recursion:

```java
public class Factorial {
    public static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * factorial(n-1);
        }
    }

    public static void main(String[] args) {
        int number = 5;
        int result = factorial(number);
        System.out.println("Factorial of " + number + " is: " + result);
    }
}
```
~~~

Advantages of this service:

- Code generation with Azure service decreases the time and effort required for writing code. Developers can quickly generate code without doing any manual intervention.
- As code generation can be automated, makes developers to focus over other parts also like high-level design (HLD), low-level designs (LLD) and more tasks.
- Developers can improve the code quality using this service.
- Ease of integration with different services, development tools.
- Streamline the development process and productivity.

Ques-6: What is DALL-E? explain it in the context of Azure OpenAI services?

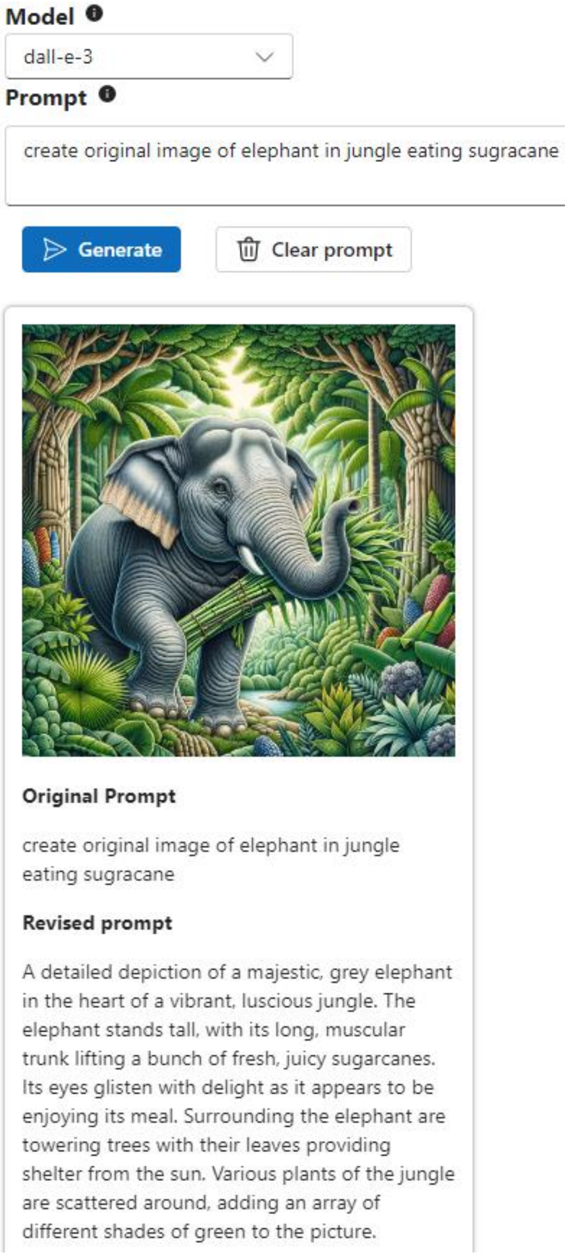
Ans-6:

The name "DALL-E" is a combination of "DALI" (a reference to the surrealist artist Salvador Dalí) and "Wall-E" (the Pixar character). DALL-E is a text-to-image model that uses deep learning to generate images from natural language descriptions, or "prompts". DALL-E accepts both image and text inputs, which are broken down into smaller units called tokens. It then compares these tokens with its training data and uses the results to produce unique images. DALL-E 3 is generally available for use with the REST APIs. DALL-E 2 and DALL-E 3 with client SDKs.

DALL-E uses in Azure OpenAI services:

- Developers can send textual descriptions of images to DALL-E and receive generated images as responses.
- Generate personalized visual content for users based on their preferences or input.
- Developers could enhance the user experience by providing visually engaging and relevant content.

Example of Dall-E-3 model creating image.



Model

dall-e-3

Prompt

create original image of elephant in jungle eating sugracane

Original Prompt

create original image of elephant in jungle eating sugracane

Revised prompt

A detailed depiction of a majestic, grey elephant in the heart of a vibrant, luscious jungle. The elephant stands tall, with its long, muscular trunk lifting a bunch of fresh, juicy sugarcanes. Its eyes glisten with delight as it appears to be enjoying its meal. Surrounding the elephant are towering trees with their leaves providing shelter from the sun. Various plants of the jungle are scattered around, adding an array of different shades of green to the picture.

Ques-7: What is RAG? Summarize your understanding of your understanding from the Lecture.

Ans-7:

RAG combines retrieval- and generative-based AI models to create unique answers, instructions, or explanations in human-like language. It uses knowledge that's more contextual than the data in a generalized LLM, and can be used in chatbots, email, text messaging, and other conversational applications. RAG models build knowledge repositories based on the organization's own data. RAG has shown success in support chatbots and Q&A systems that need to maintain up-to-date information or access domain-specific knowledge. It is a framework introduced by Facebook AI.

The retrieval component retrieves relevant passages or documents from a large knowledge source, such as a corpus of text documents or a search engine index, based on the input query or context.

The generation component takes the retrieved knowledge and the input query or context and generates a response that is fluent and coherent. This generation process may involve techniques such as fine-tuning pre-trained language models (e.g., BERT or GPT) on conversational data or using specialized architectures designed for dialogue generation.

Benefits:

- By combining retrieval and generation, RAG addresses some of the limitations of purely retrieval-based or generation-based approaches.
- Retrieval helps provide grounded and factual responses based on existing knowledge, while generation enables the model to be creative and flexible in its responses.
- RAG models can generate responses that are both informative and contextually relevant, leveraging the best of both retrieval and generation techniques.

RAG Applications:

- Question and answer chatbots: Incorporating LLMs with chatbots allows them to automatically derive more accurate answers from company documents and knowledge bases.
- Refining search processes: RAG can help refine search processes.
- Enhancing content generation: RAG can help enhance content generation.

Ques-8: What is the Azure AI Search Hybrid Retrieval? Explain Vector Embedding.

Ans-8:

Azure AI Search Hybrid Retrieval is a feature that combines vector and keyword queries in a single search request, and then uses a Reciprocal Rank Fusion (RRF) algorithm to select the most relevant matches from each query. The results are merged into a single response, and the response includes the top results ordered by search score. Hybrid search can outperform vector search in the following ways:

- Keyword and vector search tackle search from different perspectives.
- The L2 ranking step can significantly improve the quality of results in the top positions.

Vector embeddings, also known as embeddings, are a machine learning process that converts words, sentences, and other data into numerical representations that capture their meaning and relationships. This allows machines to process and understand data more effectively. The length or dimensionality of the vector depends on the specific embedding technique and how the data is represented. For example, word embeddings often have dimensions ranging from a few hundred to a few thousand. Popular techniques for generating vector embeddings include Word2Vec, GloVe, and FastText, which learn vector representations from large text corpora using neural network models.

Azure AI Search uses clever methods like vector embedding and hybrid retrieval to make searching smarter. This means it helps you find exactly what you're looking for in a huge pile of text. So, if you're searching for something, it quickly finds the most important and accurate information for you.

Ques-9: Explain the fundamentals of Responsible GenAI?

Ans-9:

Responsible GenAI refers to the ethical and thoughtful use of artificial intelligence (AI) technologies, particularly in the context of generative AI, which creates new content such as images, text, or music.

Fundamentals of Responsible GenAI:

- Responsible GenAI means using AI in a way that is fair, just, and respectful to everyone. It's important to consider how AI might affect people and make sure it doesn't harm anyone or cause problems.
- It's essential for AI systems to be transparent, meaning people should understand how they work and why they make certain decisions. This helps build trust and allows people to know what to expect from AI.
- AI systems should be safe to use and not pose any risks to people's health or well-being. This includes making sure AI-generated content is appropriate and doesn't promote harmful behavior.
- AI systems should be designed to be fair and unbiased.
- Respecting people's privacy is crucial. AI systems should handle personal information carefully and only use it for the intended purpose. It's important to protect people's data and keep it safe from unauthorized access.

Part-2

Advanced Intelligent System: Fraud Detection in Banking

Introduction

In today's world, every place is going with digitalization, where different services, organizations and other things are going up with new technologies to speed-up their process. So here we will be looking into a specific domain of banking sector, that how advanced intelligent system has helps in improving fraud detection.

Before this revolution, fraud detection was done with some manual checks and the process was much complicated because of which total security and fraud prevention goal wasn't able to get meet up. Traditional systems sometimes miss in identifying fraudulent patterns. These systems are limited with the resources and because of which slowly adapt to different types of fraud techniques. Hence, advanced intelligent systems are being rapidly used by banking sectors to improve this issue efficiently, that can analyze large volumes of data and help with the happy path scenario.

Banking fraud detection is the ability to monitor transactions and payments to identify suspicious activity. Banks need strong systems to detect fraud, prevent it, and comply with regulations.

Fraud detection can include:

- Behavioral analytics
  Monitoring user behaviors to identify patterns that indicate deviations from normal patterns. For example, unusual login locations, sudden changes in spending, or atypical transaction amounts.

- Biometric authentication
  Using technologies like fingerprint recognition, facial recognition, and voice recognition to verify customer identities during logins and transactions.
- AI
  Using machine learning models to identify anomalies in customer behaviors and connections, as well as patterns of accounts and behaviors that fit fraudulent characteristics.
- Data collection
  Collecting and ingesting behavior and device data.
- Data preprocessing
  Automatically cleaning and formatting collected data to ensure consistency and accuracy.
- Automated behavioral analysis
  Using algorithms that continuously scrutinize incoming data for patterns indicative of fraud. These algorithms evolve over time, learning from fresh data and adapting to the ever-evolving tactics employed by fraudsters.

Some tips for detecting online fraud transactions include:

- Using an address verification service
- Checking CVV (Card Verification Values)
- Using 3D Secure payer authentication
- Looking up email addresses
- Using device identification
- Flagging large transactions
- Looking for patterns
- Comparing user location and shipping destination

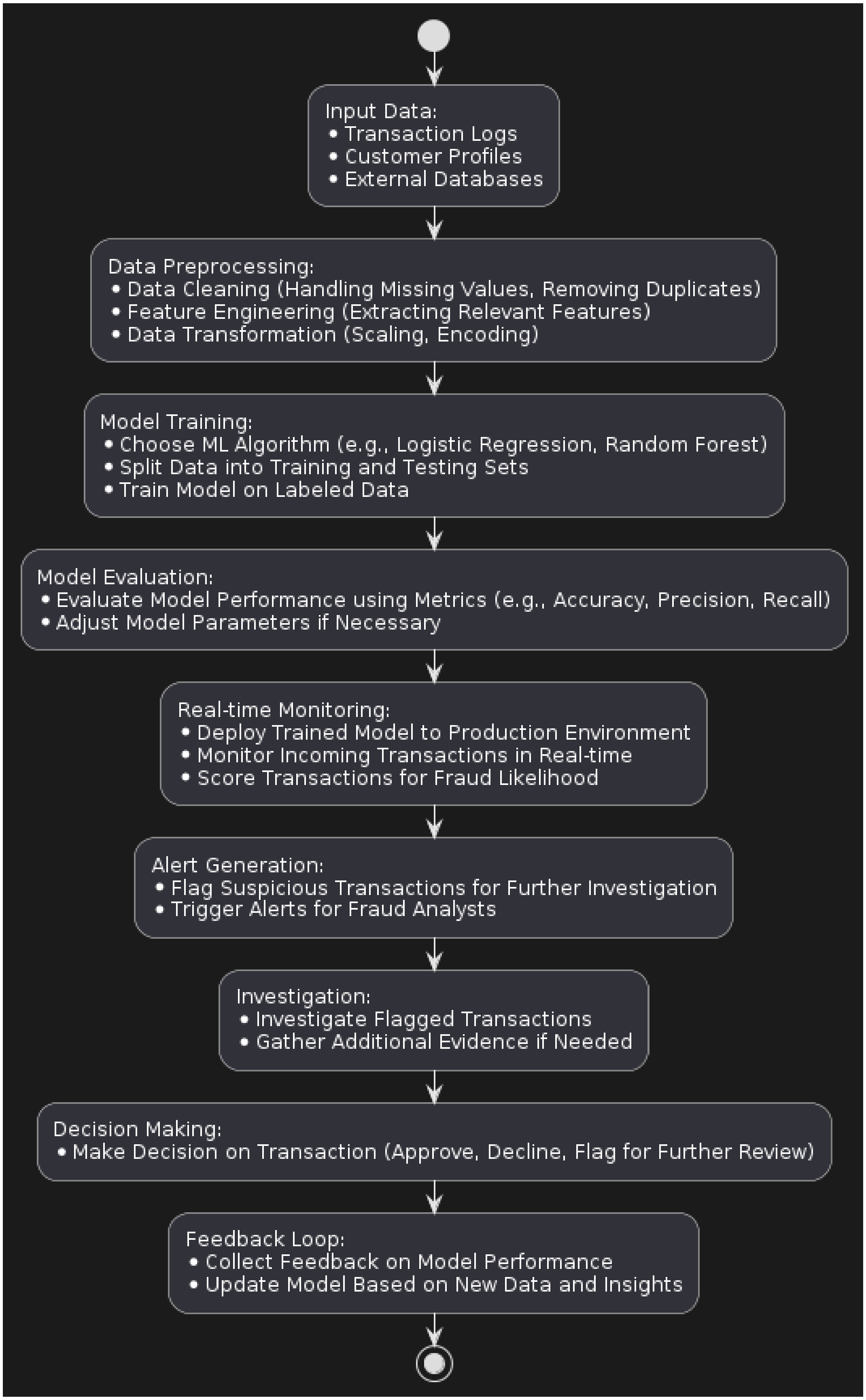
Need of advanced intelligent system

- AI systems can handle high volume transaction efficiently. Advanced intelligent systems are equipped to handle large-scale data processing and analysis, enabling banks to sift through vast amounts of transactional data in real-time to identify suspicious patterns and anomalies indicative of fraudulent activity.
- With the rise of online banking and digital transactions, the need for real-time fraud detection and response has become critical. Advanced intelligent systems can analyze transactions in real-time, enabling banks to detect and prevent fraudulent activities as they occur, thereby minimizing financial losses and protecting customers' assets.
- By automating repetitive tasks and streamlining fraud detection processes, advanced intelligent systems help banks reduce operational costs associated with manual labor, resource allocation, and investigation efforts.
- Prioritizing alerts and focusing on high-risk transactions, fraud analysts can make more informed decisions and allocate their time and resources more effectively, leading to better outcomes in fraud detection and prevention.
- Manual fraud detection processes often result in a high number of false positives, where legitimate transactions are incorrectly flagged as fraudulent. This can lead to unnecessary investigations and customer inconvenience. Advanced intelligent systems leverage sophisticated algorithms to reduce false positives, ensuring that fraud alerts are more accurate and actionable, thereby minimizing the need for manual intervention.
- Advanced intelligent systems automate many routine tasks involved in fraud detection, such as data collection, preprocessing, and initial analysis. By automating these processes, banks can significantly reduce the manual effort required by fraud analysts, allowing them to focus on more complex and high-value tasks.
- Traditional manual methods of fraud detection struggle to scale with the increasing volume of transactions and data in the banking sector. Advanced intelligent systems, powered by artificial intelligence and machine learning, are highly scalable and efficient, capable of processing large volumes of data in real-time without the need for extensive human intervention.

Working

- Problem identification and Model Requirement
  Analysts and designers identify the specific types of fraudulent activities they want to detect, determining which features and data sources are relevant for the task. They also decide on the most suitable machine learning models to address the problem effectively, considering factors like data volume, complexity of fraud patterns, and computational resources.
- Data Collection
  Teams gather relevant data from various sources, including transaction logs, customer profiles, historical fraud cases, and external databases. This data may include information such as transaction amounts, timestamps, geographical locations, customer demographics, and past fraud indicators.
- Data Cleansing
  The collected data undergoes cleaning to remove inaccuracies, inconsistencies, and noise. This involves identifying and correcting errors, handling missing values, handling outliers and standardizing formats to ensure the integrity and quality of the dataset.
- Data Labelling
  Each data entry is labelled as either fraudulent or legitimate based on historical records or expert knowledge. In cases where labelled data is limited, techniques such as semi-supervised learning or active learning may be used to iteratively label additional data points.
- Feature Engineering
  Analysts extract and engineer informative features from the cleaned data to represent patterns indicative of fraudulent behavior. This may involve transforming raw data into meaningful features, such as transaction frequency, velocity, deviation from usual behavior, and relationships between different variables.
- Model Training
  Machine learning models are trained on the labelled and feature-engineered dataset to learn patterns and relationships between features and fraud outcomes. Supervised learning algorithms are commonly used for fraud detection, with techniques like logistic regression, decision trees, random forests, and neural networks being employed.
- Model Evaluation
  The trained models are evaluated using validation datasets to assess their performance in detecting fraudulent transactions. Metrics such as precision, recall, F1-score, and ROC-AUC are calculated to measure the model's accuracy, sensitivity, and specificity.
- Model Deployment
  The trained and validated models are deployed into production systems, where they analyze incoming transactions in real-time and flag suspicious activities for further investigation. This deployment involves integrating the models with existing banking infrastructure and ensuring scalability, reliability, and security.
- Model Monitoring
- The deployed models are continuously monitored for any deviations or anomalies in their performance. This includes tracking model accuracy, false positive rates, and detection latency, as well as monitoring for changes in fraud patterns and emerging threats. Any issues or errors detected during monitoring are addressed promptly to maintain the effectiveness of the fraud detection system.
- Feedback and model improvement
- Banks establish a feedback loop mechanism to gather insights from the performance of deployed fraud detection models. This feedback loop involves capturing information on flagged transactions, including those confirmed as fraudulent and those identified as false positives.

Flowchart


Input Data:
• Transaction Logs
• Customer Profiles
• External Databases
Data Preprocessing:
• Data Cleaning (Handling Missing Values, Removing Duplicates)
• Feature Engineering (Extracting Relevant Features)
• Data Transformation (Scaling, Encoding)
Model Training:
• Choose ML Algorithm (e.g., Logistic Regression, Random Forest)
• Split Data into Training and Testing Sets
• Train Model on Labeled Data
Model Evaluation:
• Evaluate Model Performance using Metrics (e.g., Accuracy, Precision, Recall)
• Adjust Model Parameters if Necessary
Real-time Monitoring:
• Deploy Trained Model to Production Environment
• Monitor Incoming Transactions in Real-time
• Score Transactions for Fraud Likelihood
Alert Generation:
• Flag Suspicious Transactions for Further Investigation
• Trigger Alerts for Fraud Analysts
Investigation:
• Investigate Flagged Transactions
• Gather Additional Evidence if Needed
Decision Making:
• Make Decision on Transaction (Approve, Decline, Flag for Further Review)
Feedback Loop:
• Collect Feedback on Model Performance
• Update Model Based on New Data and Insights

Conclusion

Advanced intelligent system for fraud detection in banking services provides different AI techniques to analyze large volumes of transaction data, detect anomalies and identify fraud activities. As these ML models and AI services are automated it offers scalability, adaptability and accuracy far better then traditional approach. As different fraud techniques are emerging AI self-adaptability greatly helps in detection and prevent from fraudulent activities.

References

blogs.nvidia.com, n.d. *How Is AI Used in Fraud Detection.* [Online]
Available at: https://blogs.nvidia.com/blog/ai-fraud-detection-rapids-triton-tensorrt-nemo/

Great Learning, n.d. *SIG788 - Engineering AI solutions Content.* [Online]
Available at: https://olympus.mygreatlearning.com/courses/109578?module_id=779943

www.formica.ai, n.d. *Real-Time Fraud Detection in Banking: Protecting with AI.* [Online]
Available at: https://www.formica.ai/blog/real-time-fraud-detection-in-banking-protecting-with-ai

www.fraud.com, n.d. *Artificial Intelligence – How it's used to detect financial fraud.* [Online]
Available at: https://www.fraud.com/post/artificial-intelligence