Generative AI Training Syllabus

Course Overview Comprehensive Gen AI training program covering Python, MySQL, Machine learning, NLP, Azure/AWS/GCP designed to transform participants into job ready.

---

1. Programming Fundamentals (2 weeks)

Objective: Establish a strong foundation in Python, which is essential for AI and data science applications.

A. Python Basics

- **Understanding Python Syntax & Data Types**
    - Variables, Data Types (int, float, string, boolean)
    - Type conversion and casting
- **Control Structures**
    - Conditional Statements: if-else, elif
    - Looping Constructs: for, while, break, continue
- **Functions & Modules**
    - Defining Functions (def, return statements)
    - Creating and Using Python Modules
- **File Handling**
    - Reading and Writing Files (open(), read(), write(), close())
    - Working with CSV and JSON files
- **Error Handling**
    - Try-Except Blocks for Exception Handling

B. Library Utilization (Essential Libraries for AI and Data Science)

- **NumPy (Numerical Computation)**
    - Arrays and array operations
    - Mathematical and statistical functions
    - Broadcasting and vectorization
- **Pandas (Data Manipulation & Analysis)**
    - DataFrames and Series
    - Importing and Cleaning Data (Handling missing values, duplicates)
    - Data Filtering, Sorting, and Grouping
- **Matplotlib & Seaborn (Data Visualization)**
    - Basic Plots: Line, Bar, Scatter
    - Histogram and Density Plots
    - Heatmaps and Correlation Analysis
- **Working with APIs/Django Framework**
    - Making API calls using requests
    - Parsing JSON responses
    - Django Framework

Practical Component

- Writing Python scripts to automate data handling
- Using NumPy and Pandas to clean and transform datasets
- Creating simple visualizations for insights

---

2. SQL Fundamentals (1 week)

Objective: Provide essential database skills required for AI applications.

- **Topics Covered:**
  - Basics of relational databases and SQL
  - CRUD operations (Create, Read, Update, Delete)
  - Joins, aggregations, and subqueries
  - Indexing and performance optimization
- **Practical Component:**
  - Work with datasets stored in SQL databases
  - Query data for AI model training

---

3. Understanding Generative AI (1 week)

Objective: Provide strong foundation in Generative AI, its principles, real-world applications, and the underlying technologies that drive it.

- **What is Generative AI?**
  - Definition and key characteristics
  - Difference between Generative AI and Traditional AI
  - Types of AI models (Discriminative vs. Generative)
- **Evolution of Generative AI**
  - Early AI models and rule-based systems
  - Rise of Deep Learning & Transformers
  - The impact of models like GPT, DALL·E, and Stable Diffusion
- **Key Components of Generative AI**
  - Neural Networks: How they work (basic overview)
  - Latent Space Representation: How AI learns patterns in data
  - Training vs. Inference: How AI learns vs. how AI generates
- **Applications of Generative AI**
  - Text Generation (GPT-based models)
  - Automated content creation (Summarization)
  - Code Generation
- **Overview of Popular Generative AI Models**
  - Transformers and LLMs
  - VAE and GANs

---

4. **Machine Learning Fundamentals (1 week)**

Objective: Fundamental concepts of Machine Learning (ML), including different types of ML, essential algorithms, and practical applications in AI.

A. Introduction to Machine Learning

1. **What is Machine Learning?**
   - Definition and importance of ML
   - Difference between AI, ML, and Deep Learning
2. **Types of Machine Learning**
   - **Supervised Learning** (Labeled Data)
   - **Unsupervised Learning** (Unlabeled Data)
   - **Reinforcement Learning** (Learning from rewards)

B. Supervised Learning Algorithms

1. **Regression Models (Continuous Output)**
   - Linear Regression (Predicting trends)
   - Multiple Linear Regression
2. **Classification Models (Categorical Output)**
   - Logistic Regression
   - Decision Trees
   - Random Forest
   - Support Vector Machines (SVM)
   - k-Nearest Neighbors (k-NN)

C. Unsupervised Learning Algorithms

1. **Clustering**
   - k-Means Clustering
   - Hierarchical Clustering
2. **Dimensionality Reduction**
   - Principal Component Analysis (PCA)
   - t-SNE (t-Distributed Stochastic Neighbor Embedding)

D. Introduction to Neural Networks & Deep Learning

1. **What are Neural Networks?**
   - Perceptron Model
   - Activation Functions (ReLU, Sigmoid, Softmax)
   - Forward and Backward Propagation
2. **Deep Learning Overview**
   - Convolutional Neural Networks (CNN) (Image Recognition)
   - Recurrent Neural Networks (RNN) (Sequential Data & Text)
   - Long Short-Term Memory (LSTM) Networks

E. Practical Component (Hands-on Exercises & Mini-Projects)

Exercise 1: Data Preprocessing with Pandas

- Cleaning and handling missing data

Exercise 2: Build a Simple Linear Regression Model

- Predict house prices using historical data

---

5. **Natural Language Processing (NLP) (1 week)**

Objective: Foundational and advanced skills to process, analyze, and generate textual data using Machine Learning (ML) and Deep Learning models, including Large Language Models (LLMs) like BERT and GPT.

A. Introduction to NLP

1. **What is NLP?**
   - Definition and significance in AI
   - Difference between NLP and traditional text processing
   - Common challenges in NLP (ambiguity, context, sarcasm, etc.)
2. **Real-World Applications of NLP**
   - Chatbots & Virtual Assistants (e.g., Siri, Alexa, ChatGPT)
   - Sentiment Analysis (Product reviews, social media monitoring)
   - Machine Translation (Google Translate)
   - Text Summarization (News aggregation, auto-summary)
   - Information Retrieval (Search Engines)
   - Named Entity Recognition (NER) (Extracting names, locations, etc.)

B. Techniques for Text Preprocessing and Vectorization

1. **Basic Text Preprocessing**
   - Tokenization (word-level, sentence-level)
   - Stopword removal (removing common words like “the,” “is”)
2. **Vectorization Techniques (Converting text to numbers)**
   - Bag of Words (BoW)
   - Word Embeddings

C. Large Language Models (LLMs) in NLP

1. **Understanding BERT (Bidirectional Encoder Representations from Transformers)**
   - Concept of bidirectionality in understanding text
   - Use cases: Sentiment analysis, NER, text classification
   - Fine-tuning BERT for custom NLP tasks
2. **Understanding GPT (Generative Pre-trained Transformer)**
   - Difference between BERT and GPT
   - How GPT models generate human-like text
   - Applications: Chatbots, text generation, and AI-powered content creation

D. Practical Component: Hands-on NLP Projects

Mini-Project: Chatbot Development

- Develop a basic chatbot using Python and NLP technique

---

6. **Building and Deploying Generative AI Applications (1 week)**

Objective: Guide students through the development and deployment of AI-driven applications.

- **Topics Covered:**
  - Best practices for model integration into real-world applications.
  - Utilization of cloud platforms and APIs for deployment.
- **Practical Component:** Create and deploy a Generative AI application, such as a content generator or recommendation system.

---

7. Azure AI/ML Services for Generative AI (3 weeks)

Azure Service	**Use Case in Generative AI**
Azure Virtual Machines, Azure Functions	Hosting ML models & running inference
Azure Machine Learning	Model training, fine-tuning LLMs
Azure OpenAI Service	Accessing GPT models (GPT-4, Codex, DALL·E)
Azure Cognitive Services - Language	NLP, Text Analytics, Sentiment Analysis
Azure Cognitive Services - Speech	Speech-to-text, text-to-speech
Azure Cognitive Services - Vision	Image recognition & object detection
Azure Bot Service	Building AI-powered chatbots
Azure Cognitive Search	Intelligent search & document processing
Azure Blob Storage	Storing AI/ML datasets & models
Azure SQL Database, Cosmos DB	Storing structured data for AI applications

1. **Azure Setup & Fundamentals (Basics of AI & ML)**
   a. Hands-on: Set up an Azure account & assign IAM roles for AI/ML workloads.
   b. Objective: Learn Azure security best practices for AI development.

2. **Azure for NLP & Chatbots (NLP & Conversational AI)**
   a. Use Azure Cognitive Services - Language for Text Processing & Sentiment Analysis.
   b. Use Azure Bot Service to build a chatbot with Azure OpenAI (GPT models).

3. **Azure for Text Summarization (Text Summarization)**
   a. Use Azure OpenAI GPT-4 APIs for abstractive text summarization.

4. **Azure for Generative AI & LLMs (Generative AI Models)**
   a. Use Azure OpenAI Service to fine-tune GPT models.
   b. Use Azure ML for Custom LLM Training & Deployment.

5. **Azure for Model Deployment (Model Deployment & Optimization)**
   a. Deploy models using Azure Machine Learning & Azure Functions.
   b. Use Azure Kubernetes Service (AKS) for scalable AI model deployment.

6. **Azure for Data Engineering & Storage (SQL Module)**
   a. Store AI datasets in Azure Blob Storage.
   b. Store structured data in Azure SQL Database & Cosmos DB.

8. **AWS Services for Machine Learning & Generative AI (3 weeks)**

AWS Service	**Use Case in Generative AI**
Amazon EC2, AWS Lambda	Hosting ML models & running inference
Amazon SageMaker	Model training, fine-tuning LLMs
AWS Bedrock	foundation models (FMs) like Meta Llama
Amazon Lex	Building conversational AI & chatbots
Amazon Transcribe	Converting speech to text
Amazon Polly	Converting text to speech
Amazon Rekognition	Image & video analysis
Amazon S3	Storing training datasets & ML models
Amazon RDS (PostgreSQL, MySQL), DynamoDB	Storing structured data for AI applications
AWS Key Management Service (KMS), IAM	Managing API keys & securing access

1. **AWS Setup & Fundamentals**
   - Hands-on: Create an AWS account, set up IAM roles for AI/ML workloads.
   - Objective: Ensure students understand AWS identity & security best practices.

2. **AWS for NLP & Chatbots (NLP)**
   - Use Amazon Lex to build a chatbot.
   - Integration: Use Amazon Polly for text-to-speech capabilities.

3. **AWS for Text Summarization (Text Summarization)**
   - Use Amazon Bedrock to fine-tune Claude or Llama models for abstractive summarization.

4. **AWS for Generative AI & LLMs (Generative AI Models)**
   - Use AWS Bedrock to access LLMs like Claude, Llama, and Falcon.
   - Fine-tuning: Train custom models using Amazon SageMaker JumpStart.

5. **AWS for Model Deployment (Model Deployment & Optimization)**
   - Use Amazon SageMaker for deploying ML models
   - Use AWS Lambda for serverless inference

6. **AWS for Data Engineering & Storage (SQL Module)**
   - Store NLP datasets in Amazon RDS (PostgreSQL, MySQL)

9. GCP AI/ML Services for Generative AI (3 weeks)

GCP Service	Use Case in Generative AI
Compute Engine, Cloud Functions	Hosting ML models & inference
Vertex AI	End-to-end ML model training, tuning, & deployment
Google AI Studio (Gemini API)	Accessing Google's Gemini LLMs for text & image generation
Cloud Natural Language API	Text classification, entity recognition, sentiment analysis
Speech-to-Text & Text-to-Speech	Converting speech to text & vice versa
Vision AI	Image recognition & processing
Document AI	Processing PDFs, invoices, receipts using AI
Dialogflow CX	Building AI-powered chatbots
BigQuery ML	Running ML models on large datasets
Cloud Storage	Storing datasets, models, & AI-generated content
BigQuery, Firestore	Storing structured data for AI applications
Cloud Build, AI Pipelines	Automating AI/ML deployment

1. **GCP Setup & Fundamentals**
   - Hands-on: Set up a GCP account & enable Vertex AI.
   - Objective: Learn GCP security & IAM roles for AI projects.

2. **GCP for NLP & Chatbots**
   - Use Cloud Natural Language API for text processing & sentiment analysis.
   - Use Dialogflow CX to build an AI chatbot.

3. **GCP for Text Summarization**
   - Use Gemini API for abstractive text summarization.
   - Use Cloud Natural Language API for extractive summarization.

4. **GCP for Generative AI & LLMs**
   - Use Vertex AI to fine-tune LLMs (Gemini, T5, BERT).
   - Use AI Studio (Gemini API) for AI-powered content generation.

5. **GCP for Model Deployment**
   - Deploy ML models using Vertex AI Model Registry & AI Pipelines.
   - Use Cloud Functions for lightweight AI inference.
   - Use Cloud Run to deploy scalable AI web apps.

6. **GCP for Data Engineering & Storage**
   - Store AI datasets in Cloud Storage.