Generative AI Training

Course Duration: 5.5 months, Weekly 6 days, Each Class: 1 hr.
Trainer: D. Chaitanya (IIT Roorkee Alumni), 21+ yrs Exp.

a) **Python:** ~ 2 month

I) Complete Python

- Introduction:
  i. Why Python?
  ii. Who Uses Python today
  iii. What can we do with Python
  iv. How Python Developed and Supported
  v. Python – Technical Strengths.

- Python Interpreter
- Program Execution – programmer's view, Python's view
- Installation
  i. Python
  ii. All Related Software: PyCharm, VS Code, Jupyter Notebook
  iii. Setup, configure Python in Laptop
- All Numeric types in Python, coding/Hands-on
- Python Variables, objects, References, Shared References, coding/Hands-on
- Garbage Collection of objects
- All built in types in python: Strings, Lists, Dictionaries, Tuples, sets, Files
- Python Statements - coding/Hands-on
- Assignments, Expressions and Prints
  i. if-else, if-elif-else, if-else ternary expression
  ii. while and for loops
  iii. Comprehensions vs regular
  iv. Parallel Traversals: map and zip functions
  v. Other important functions: range, len, enumerate

- Iterations and Comprehensions - coding/Hands-on
- Python online Documentation
- Python Functions – def, nested functions
- Variable Scopes – basics, LEGB rules, global, nonlocal
- Function Arguments - coding/Hands-on
  i. Arguments and shared references

ii. Arguments passing basics
iii. Arguments matching basics
iv. Arguments matching syntax

- Advanced Function Concepts –Attributes, lambda functions
- Generators and comprehensions
  i. Generator functions, yield statement
  ii. Generator expression
- Python Modules
  i. Definition, why modules?
  ii. Typical Python program architecture
  iii. Import statement - coding/Hands-on
  iv. How Import works: Find it, Compile it, Run it
  v. Standard library modules
  vi. __pycache__ folder for byte code files
  vii. Module search path

- Module coding Basics
  i. Module creation
  ii. import statement, from statement, from * statement
  iii. Module Namespaces, Namespace dictionaries: __dict__

- Module Packages
  i. Package import basics
  ii. Why Package imports?
  iii. Relative import basics
  iv. Why relative imports?

- Advanced Module Topics
  i. Mixed usage modes: __name__ and __main__ - coding/Hands-on
  ii. The as extension for import and from - coding/Hands-on

- Introduction to Python Classes - coding/Hands-on
  i. Why Classes?
  ii. Classes, constructors and Instances
  iii. Method calls
  iv. Attribute inheritance search
  v. OOP is about code reuse
  vi. Subclassing by Inheritance

vii. Polymorphism in Action
viii. Class vs instance attributes

- Coding with Classes - coding/Hands-on
  i. Abstract super classes
  ii. Nested classes

- Operator Overloading - coding/Hands-on
  i. Constructors: __init__
  ii. Indexing, Slicing: __getitem__ and __setitem__
  iii. Attribute Access: __getattr__ and __setattr__
  iv. String Representation: __repr__ and __str__
  v. Right side and In-Place Uses: __radd__ and __iadd__
  vi. Call Expressions: __call__
  vii. Comparisons: __lt__, __gt__ and others
  viii. Boolean Tests: __bool__ and __len__
  ix. Destructors: __del__

- Special features of Classes - coding/Hands-on
  i. Inheritance: “IS-a” relationship
  ii. Composition: “HAS-a” relationship

- Advanced Class Topics - coding/Hands-on
  i. “New style” class model
  ii. Diamond inheritance change
  iii. MRO: Method Resolution Order
  iv. Static and Class methods
  v. The “super” built-in function

- Exception Basics - coding/Hands-on
  i. Why Exceptions?
  ii. Default Exception handler
  iii. Catching Exceptions
  iv. Raising Exceptions

- Coding Exceptions - coding/Hands-on
  i. The try/except/else statement
  ii. try/finally statement
  iii. raise statement

iv. assert statement
v. with/as context managers
vi. Nesting Exception Handlers

- Exception Objects
  i. Class based exceptions
  ii. Why Exception hierarchies?
  iii. Built-in Exception Classes
  iv. Custom Exceptions

II) Regular Expressions with Python:

- What are regular expressions?
- regex module in python
- The match Function
- The search Function
- Matching vs searching
- Search and Replace
- Meta characters, advanced patterns

III) Data Libraries:

- Introduction to numpy
- Creating arrays
- Indexing Arrays
- Array Transposition
- Universal Array Function
- Array Processing
- Array Input and Output
- Introduction to Pandas, Series, Dataframes
- Data reading with Pandas
- Data cleaning with Pandas
- Data wrangling with Pandas
- Data selection with Pandas
- Data extraction with Pandas

b) **Mathematics:** ~ 0.1 month

- Linear Algebra
- Statistics:

- Probability:
- Differential Calculus:

c) **Machine Learning:** ~ 2 month

- Introduction to Machine Learning:
  i. What is Machine Learning
  ii. Why use Machine Learning
  iii. Examples of ML applications
  iv. Types of ML Systems.
  v. Supervised Learning
  vi. Unsupervised Learning
  vii. Batch vs Online Learning
  viii. Instance-based vs Model-based Learning
  ix. Challenges of Machine Learning
  x. Overfitting vs underfitting training data
  xi. All phases of End to End ML Project.

- Classification Models
  i. Binary Classifier
  ii. Performance Measures
    1. Accuracy
    2. Confusion Matrix
    3. Precision and Recall
  iii. Multi Class Classification
  iv. Multi Label Classification
  v. Multi Output Classification

- Regression Models
  i. Linear Regression
  ii. Gradient Descent
    1. Batch Gradient Descent
    2. Stochastic Gradient Descent
    3. Mini-batch Gradient Descent
  iii. Polynomial Regression
  iv. Regularized Linear Models
    1. Lasso Regression
    2. Early Stopping
  v. Logistic Regression

- Decision Trees
  i. Introduction to Decision Tree
  ii. Training Decision Tree
  iii. Visualizing Decision Tree
  iv. Estimating Class Probabilities
  v. The CART Training Algorithm
  vi. Computational Complexity
  vii. Gini Impurity vs Entropy
  viii. Regularization of Hyperparameters
  ix. Regression, Instability

- Random Forests
  i. Ensemble Learning
  ii. Voting Classifiers
  iii. Bagging and Pasting
  iv. Bagging & Pasting in sci-kit Learn
  v. Out of bag evaluation
  vi. Random Patches and Random Subspaces
  vii. Random Forests
  viii. Extra Tree and Feature importance

- Unsupervised Learning Techniques
  i. Clustering
  ii. K-Means Algorithm and Limits of K-Means
  iii. KMeans++
  iv. Semi-supervised Learning using Clustering

d) **Deep Learning:** ~ 1.5 month

- Introduction to Artificial Neural Networks with Keras
  i. Biological Neuron
  ii. The Perceptron
  iii. Multilayer Perceptron and Back propagation
  iv. Regression MLPs
  v. Classification MLPs
  vi. Implementing MLPs with Keras
  vii. Building an Image classifier using Sequential API
  viii. Building Regression MLP using sequential API
  ix. Saving and Restoring Model
  x. Fine tuning neural network hyper parameters

xi. Number of Hidden layers
xii. Number of neurons per hidden layer
xiii. Learning rate, Batch size and other hyper parameters

- Training Deep Neural Networks
  i. Vanishing/Exploding Gradients problems.
     1. Glorot and He initialization
     2. Batch Normalization
     3. Gradient Clipping
  ii. Transfer Learning
  iii. Avoiding overfitting through Regularization
     1. Dropout
     2. MC (Monte Carlo) Dropout

e) **Generative AI:** ~1.5 months

- Introduction to Generative AI, LLMs
  i. Introduction
  ii. AI, ML , DL, Gen AI, LLMs
  iii. Capabilities of Gen AI Models
  iv. Categories of Gen AI Models
  v. Improvements in LLM Models
  vi. Applications of LLMs
  vii. GPT Models, Evolution,
  viii. Popular Players
  ix. Foundational, Open-Source, Closed-Source LLMs
  x. Gen AI Models with Modalities
- LLMs
  i. LLM Limitations
  ii. Mitigating LLM Limitations
  iii. LLM App vs Traditional App
  iv. Few examples of LLM App
  v. OpenAI LLM Model Setup, usage
  vi. Google Gemini LLM Model Setup, usage
  vii. Hugging Face LLM Models, usage
  viii.
- LangChain Orchestration Framework Intro
  i. What is Langchain
  ii. Key Components

iii. Modules in LacgChain

- Build LLM Apps
  i. Customer Service Apps
  ii. Sentiment Analysis
  iii. Text classification, Text Summarization, Document summarization
  iv. Extracting structured information from a document
  v. Output parsers, parsers in LangChain
  vi. Tokens, monitoring Token usage & Cost.
  vii. App using Streamlit
  viii. Key Components
  ix. Modules in LacgChain
- RAG
  i. Why RAG
  ii. RAG pipeline architecture – in context, Context window
  iii. RAG pipeline architecture – external data source
  iv. Document Loaders in Langchain
  v. Document Splitters in Langchain
  vi. Embeddings, Embedding Models in Langchain
  vii. Vector Storage in Langchain
  viii. Vector Databases and providers
  ix. Retrievers in Langchain
  x. LLM in Action
  xi. Memory in Langchain
  xii. Chains in Langchain
  xiii. Prompt Templates in Langchain
  xiv. Usages of Chains
  xv. Hallucinations, Moderating Responses
  xvi. Building Chatbot App
- Prompt Engineering
  i. Introduction
  ii. Instruction based prompting
  iii. Complex prompts, building complex prompts
  iv. Zero shot, One shot, Few Shot promptings
  v. Chain Prompting
  vi. Chain of Thought
  vii. Self-Consistency
  viii. Tree of Thought
  ix. Verifying Output, Controlling Output