

# Full Stack **DATA SCIENCE**

with  
**Gen AI & Agentic AI**



Agentic AI



Generative AI



Prompt  
Engineering



LLM Models



GANs



RAG



MLOps



CICD Pipeline



Neural Network

In Collaboration With



**NASSCOM<sup>®</sup>**

# PYTHON

## Introduction to Data Science

- ✓ Introduction of Data Science
- ✓ Discussion on Course Curriculum
- ✓ Introduction to Programming

## Python – Basics

- ✓ Introduction to Python: Installation and Running  
(Jupyter Notebook, .py file from terminal, Google Colab)
- ✓ Data types and type conversion
- ✓ Variables
- ✓ Operators
- ✓ Flow Control : If, Elif, Else
- ✓ Loops
- ✓ Python Identifier
- ✓ Building Functions (print, type, id, sys, len )

## Python - Data Types & Utilities

- ✓ List, List of Lists and List Comprehension
- ✓ List creation
- ✓ Create a list with variable
- ✓ List mutable concept
- ✓ len() || append() || pop()
- ✓ insert() || remove() || sort() || reverse()



- ✓ Backward Indexing
- ✓ Forward slicing
- ✓ Backward slicing
- Step slicing

## Set

- ✓ SET creation with variable
- ✓ len() || add() || remove() || pop()
- ✓ union() | intersection() || difference()

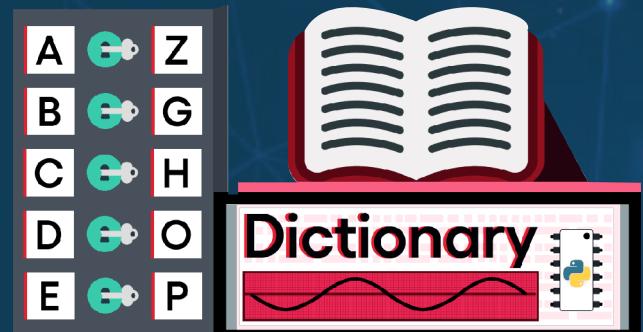


## Tuple

- ✓ TUPLE Creation
- ✓ Create Tuple with variable
- ✓ Tuple Immutable concept
- ✓ len() || count() || index()
- ✓ Forward indexing
- ✓ Backward Indexing

## Dictionary & Dictionary comprehension

- ✓ Create a dictionary using variable
- ✓ keys:values concept
- ✓ len() || keys() || values() || items()
- ✓ get() || pop() || update()
- ✓ comparision of datastructure
- ✓ Introduce to range()
- ✓ pass range() in the list
- ✓ range() arguments
- ✓ For loop introduction using range()



## Functions

- ✓ Inbuilt vs User Defined
- ✓ User Defined Function
- ✓ Function Argument
- ✓ Types of Function Arguments
- ✓ Actual Argument
- ✓ Global variable vs Local variable
- ✓ Anonymous Function | LAMBDA

## Packages | Map Reduce | OOP's

## Class & Object:

- ✓ What is mean by inbuild class
- ✓ How to creat user class
- ✓ crate a class & object
- ✓ `__init__` method
- ✓ Python constructor
- ✓ Constructor, self & comparing objects
- ✓ Instane variable & class variable

## Methods

- ✓ what is instance method
- ✓ what is class method
- ✓ what is static method
- ✓ Accessor & Mutator

## Python DECORATOR:

- ✓ how to use decorator
- ✓ inner class, outerclass
- ✓ Inheritance

## Polymorphism

- ✓ duck typing
- ✓ operator overloading
- ✓ method overloading
- ✓ method overriding
- ✓ Magic method
- ✓ Abstract class & Abstract method
- ✓ Iterator
- ✓ Generators in python

## Python - Production Level

- ✓ Error / Exception Handling
- ✓ File Handling
- ✓ Docstrings
- ✓ Modularization

## Pickling & Unpickling

## Pandas

- ✓ Introduction, Fundamentals, Importing, Pandas, Aliasing, DataFrame



- ✓ Series – Intro, Creating Series Object, Empty Series Object, Create series from List/Array/Column from DataFrame, Index in Series, Accessing values in Series
- ✓ NaN Value
- ✓ Series – Attributes (Values, index, dtypes, size)
- ✓ Series – Methods – head(), tail(), sum(), count(), nunique() etc.,
- ✓ Date Frame
- ✓ Loading Different Files
- ✓ Data Frame Attributes
- ✓ Data Frame Methods
- ✓ Rename Column & Index
- ✓ Inplace Parameter
- ✓ Handling missing or NaN values
- ✓ iLoc and Loc
- ✓ Data Frame – Filtering
- ✓ Data Frame – Sorting
- ✓ Data Frame – GroupBy
- ✓ Merging or Joining
- ✓ Data Frame – Concat
- ✓ DataFrame - Adding, dropping columns & rows
- ✓ DataFrame - Date and time
- ✓ DataFrame - Concatenate Multiple csv files



## Numpy

- ✓ Introduction, Installation, pip command, import numpy package, ModuleNotFoundError, Famous Alias name to Numpy
- ✓ Fundamentals – Create Numpy Array, Array, Manipulation, Mathematical Operations, Indexing & Slicing

- ✓ Numpy Attributes
- ✓ Important Methods- min(),max(), sum(), reshape(), count\_nonzero(), sort(), flatten() etc.,
- ✓ adding value to array of values
- ✓ Diagonal of a Matrix
- ✓ Trace of a Matrix
- ✓ Parsing, Adding and Subtracting Matrices
- ✓ "Statistical Functions: numpy.mean()
- ✓ numpy.median()
- ✓ numpy.std()
- ✓ numpy.sum()
- ✓ numpy.min()"
- ✓ Filter in Numpy



## Matplotlib

- ✓ Introduction
- ✓ Pyplot
- Figure Class
- ✓ Axes Class
- ✓ Setting Limits and Tick Labels
- ✓ Multiple Plots
- ✓ Legend
- ✓ Different Types of Plots:

- ✓ Line Graph
- ✓ Bar Chart
- ✓ Histograms,
- ✓ Scatter Plot
- ✓ Pie Chart
- ✓ 3D Plots
- ✓ Working with Images
- ✓ Customizing Plots+



## Seaborn

- ✓ catplot() function
- ✓ stripplot() function
- ✓ boxplot() function
- ✓ violinplot() function
- ✓ pointplot() function

- ✓ barplot() function
- ✓ Visualizing statistical relationship with Seaborn relplot() function
- ✓ scatterplot() function

- ✓ regplot() function
- ✓ lmplot() function
- Seaborn Facetgrid() function
- ✓ Multi-plot grids
- ✓ Statistical Plots:
- ✓ Color Palettes:
- ✓ Faceting:
- ✓ Regression Plots:
- ✓ Distribution Plots
- ✓ Categorical Plots:
- ✓ Pair Plots

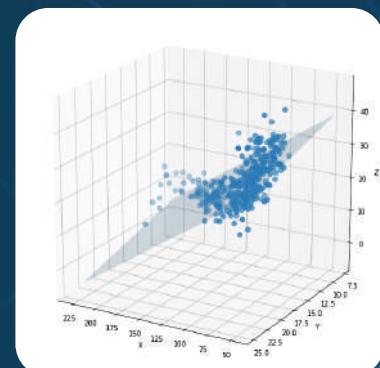


## Scipy

- ✓ Signal and Image Processing (scipy.signal,scipy.ndimage):
- ✓ Linear Algebra (scipy.linalg):
- ✓ Integration (scipy.integrate)
- ✓ Statistics (scipy.stats):
- ✓ Spatial Distance and Clustering (scipy.spatial):

## Statsmodels

- ✓ Linear Regression (statsmodels.regression):
- ✓ Time Series Analysis (statsmodels.tsa):
- ✓ Statistical Tests (statsmodels.stats)
- ✓ Anova (statsmodels.stats.anova):
- ✓ Datasets (statsmodels.datasets):



# Mathematics

## Distributions

- ✓ Data Representation & Database Operations

## Combinatorics

- ✓ Feature Selection
- ✓ Permutations and Combinations for Sampling
- ✓ Hyperparameter Tuning
- ✓ Experiment Design
- ✓ Data Partitioning and Cross-Validation

## Probability

- ✓ Basics
- ✓ Theoretical Probability
- ✓ Empirical Probability
- ✓ Addition Rule
- ✓ Multiplication Rule
- ✓ Conditional Probability
- ✓ Total Probability
- ✓ Probability Decision Tree
- ✓ Bayes Theorem
- ✓ Sensitivity & Specificity in Probability
- ✓ Bernouli Naïve Bayes, Gausian Naïve Bayes
- ✓ Multinomial Naïve Bayes



## Distributions

- ✓ Binomial, Poisson, Normal Distribution, Standard Normal Distribution
- ✓ Gaussian Distribution, Uniform Distribution
- ✓ Z Score
- ✓ Skewness
- ✓ Kurtosis
- ✓ Geometric Distribution
- ✓ Hyper Geometric Distribution
- ✓ Markov Chain



## Linear Algebra

- ✓ Linear Equations
- ✓ Matrices(Matrix Algebra: Vector Matrix Vector matrix multiplication  
Matrix matrix multiplication)
- ✓ Determinant
- ✓ Eigen Value and Eigen Vector

## Euclidean Distance & Manhattan Distance Calculus

- ✓ Differentiation
- ✓ Partial Differentiation
- ✓ Max & Min

## Indices & Logarithms

# STATISTICS

## Introduction

- ✓ Population & Sample
- ✓ Reference & Sampling technique



## Types of Data

- ✓ Qualitative or Categorical – Nominal & Ordinal
- ✓ Quantitative or Numerical – Discrete & Continuous
- ✓ Cross Sectional Data & Time Series Data

## Measures of Central Tendency

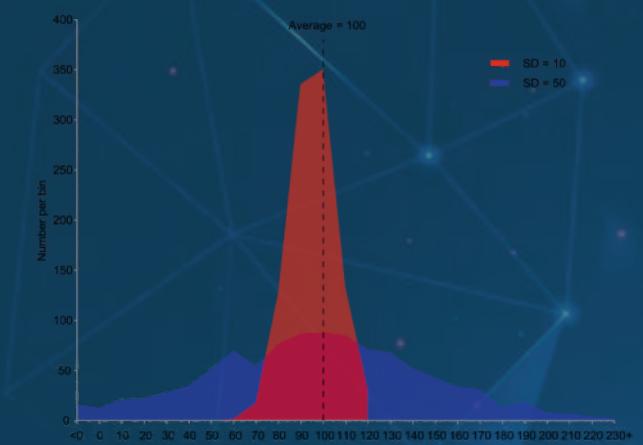
- ✓ Mean, Mode & Median – Their frequency distribution

## Descriptive statistic Measures of symmetry

- ✓ skewness (positive skew, negative skew, zero, skew)
- ✓ kurtosis (Leptokurtic, Mesokurtic, Platykurtic)

## Measurement of Spread

- ✓ Range, Variance, Standard Deviation



## Measures of variability

- ✓ Interquartile Range (IQR):
- ✓ Mean Absolute Deviation (MAD)
- ✓ Coefficient of variation
- ✓ Covariance

## Levels of Data Measurement

- ✓ Nominal, Ordinal, Interval, Ratio

## Variable

- ✓ Types of Variables.
- ✓ Categorical Variables - Nominal variable & ordinal variables
- ✓ Numerical Variables: discrete & continuous
- ✓ Dependent Variable
- ✓ Independent Variable
- ✓ Control Moderating & Mediating

## Frequency Distribution Table

- ✓ Nominal, Ordinal, Interval, Ratio

## Types of Variables.

- ✓ Categorical Variables - Nominal variable & ordinal variables
- ✓ Numerical Variables: discrete & continuous
- ✓ Dependent Variable
- ✓ Independent Variable
- ✓ Control Moderating & Mediating

## Frequency Distribution Table

- ✓ Relative Frequency, Cumulative Frequency
- ✓ Histogram ✓ Scatter Plots
- ✓ Range ✓ Calculate Class Width:
- ✓ Create Intervals ✓ Count Frequencies
- ✓ Construct the Table



## Correlation, Regression & Collinearity

- ✓ Pearson & Spearman Correlation Methods
- ✓ Regression Error Metrics

## Others

- ✓ Percentiles, Quartiles, Inner Quartile Range
- ✓ Different types of Plots for Continuous, Categorical variable
- ✓ Box Plot, Outliers
- ✓ Confidence Intervals
- ✓ Central Limit Theorem
- ✓ Degree of freedom

**Bias & Variance in ML | Entropy in ML | Information Gain | Surprise in ML**

## Loss Function & Cost Function

- ✓ Mean Squared Error, Mean Absolute Error – Loss Function
- ✓ Huber Loss Function
- ✓ Cross Entropy Loss Function

## Inferential Statistics

- ✓ Hypothesis Testing: One tail, two tail and pvalue
- ✓ Formulation of Null & Alternate Hypothesis
- ✓ Type-I error & Type-II error
- ✓ Statistical Tests:
- ✓ Sample Test
- ✓ ANOVA Test
- ✓ Chi-square Test
- ✓ Z-Test & T-Test



# SQL

## Introduction

- ✓ DBMS vs RDBMS
- ✓ Intro to SQL
- ✓ SQL vs NoSQL
- ✓ MySQL Installation

## Keys

- ✓ Primary Key
- ✓ Foreign Key

## Constraints

- |            |                  |
|------------|------------------|
| ✓ Unique   | ✓ Default        |
| ✓ Not NULL | ✓ Auto Increment |
| ✓ Check    |                  |

## CRUD Operations

- |            |          |
|------------|----------|
| ✓ Create   | ✓ Update |
| ✓ Retrieve | ✓ Delete |

## SQL Languages

- ✓ Data Definition Language (DDL)
- ✓ Data Query Language
- ✓ Data Manipulation Language (DML)
- ✓ Data Control Language
- ✓ Transaction Control Language



## SQL Commands

- ✓ Create
- ✓ Insert
- ✓ Alter, Modify, Rename, Update
- ✓ Delete, Truncate, Drop
- ✓ Grant, Revoke
- ✓ Commit, Rollback
- ✓ Select

## SQL Clause

- ✓ Where      ✓ GroupBy
- ✓ Distinct    ✓ Having
- ✓ OrderBy    ✓ Limit



## Operators

- ✓ Comparison Operators
- ✓ Logical Operators
- ✓ Membership Operators
- ✓ Identity Operators

## Wild Cards | Aggregate Functions

## SQL Joins

- ✓ Inner Join & Outer Join
- ✓ Left Join & Right Join
- ✓ Self & Cross Join
- ✓ Natural Join



# EDA & ML

## EDA

- ✓ Univariate Analysis
- ✓ Bivariate Analysis
- ✓ Multivariate Analysis

## Data Visualisation

- ✓ Various Plots on different datatypes
- ✓ Plots for Continuous Variables
- ✓ Plots for Discrete Variables
- ✓ Plots for Time Series Variables

## ML Introduction

- ✓ What is Machine Learning?
- ✓ Types of Machine Learning Methods
  - ✓ Supervised Learning
  - ✓ Unsupervised Learning
  - ✓ Reinforcement Learning)
- ✓ Classification problem in general
- ✓ Validation Techniques: CV, OOB
- ✓ Different types of metrics for Classification
- ✓ Curse of dimensionality
- ✓ Feature Transformations
- ✓ Feature Selection
- ✓ Imbalanced Dataset and its effect on Classification
- ✓ Bias Variance Tradeoff



## Important Element of Machine Learning

### Multiclass Classification

- ✓ One-vs-All
- ✓ Overfitting and Underfitting
- ✓ Error Measures
- ✓ PCA learning
- ✓ Statistical learning approaches
- ✓ Introduce to SKLEARN FRAMEWORK



### Data Processing

- ✓ Creating training and test sets, Data scaling & Normalisation
- ✓ Feature Engineering – Adding new features as per requirement, Modifying the data
- ✓ Data Cleaning – Treating the missing values, Outliers
- ✓ Data Wrangling – Encoding, Feature Transformations, Feature Scaling
- ✓ Feature Selection – Filter Methods, Wrapper Methods, Embedded Methods
- ✓ Dimension Reduction – Principal Component Analysis (Sparse PCA & Kernel PCA), Singular Value Decomposition
- ✓ Non Negative Matrix Factorization

### Regression

- ✓ Introduction to Regression
- ✓ Mathematics involved in Regression
- ✓ Regression Algorithms:
- ✓ Simple Linear Regression
- ✓ Multiple Linear Regression
- ✓ Polynomial Regression
- ✓ Lasso Regression
- ✓ Ridge Regression
- ✓ Elastic Net Regression

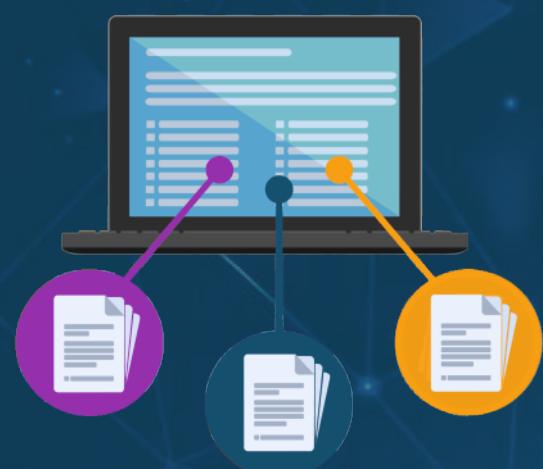
## Evaluation Metrics for Regression:

- ✓ Mean Absolute Error (MAE)
- ✓ Mean Squared Error (MSE)
- ✓ Root Mean Squared Error (RMSE)
- ✓  $R^2$  ✓ Adjusted  $R^2$



## Classification

- ✓ Introduction
- ✓ K-Nearest Neighbors
- ✓ Logistic Regression:
  - ✓ Implementation & Optimizations
  - ✓ Stochastic gradient descent algorithms
  - ✓ Finding the optimal HyperParameters
  - ✓ through Grid Search
- ✓ Support Vector Machines (Linear SVM):
  - ✓ Linear support vector machines
  - ✓ Scikit-learn implementation
- ✓ Linear Classification
- ✓ Kernel-based classification
  - ✓ Radial Basis Function
  - ✓ Polynomial Kernel
  - ✓ Sigmoid Kernel
  - ✓ Custom Kernels
- ✓ Non-linear examples
- ✓ 2 features forms straight line & 3 features forms plane



- ✓ Hyperplane and Support vectors
- ✓ Controlled support vector machines
- ✓ Support vector Regression
- ✓ Kernel SVM (Non-Linear SVM)
- ✓ Naives Bayes:
  - ✓ Bayes theorem
  - ✓ Naive Bayes Classifiers
  - ✓ Naive Bayes in scikit learn ( Bernoulli, Naive Bayes, Multinomial Naive Bayes, Gaussian Naive Bayes)"
- ✓ Decision Trees:
  - ✓ Binary Decision Trees
  - ✓ Binary decisions
  - ✓ CART Algorithm
  - ✓ Impurity measures (Gini impurity, index, Cross-entropy impurity index, Misclassification impurity index)
  - ✓ Feature importance
  - ✓ Decision tree classification with scikit learn
- ✓ Random Forest / Bagging:
  - ✓ Random Forests and Features importance in Random Forest
  - ✓ AdaBoost
  - ✓ Gradient tree boosting
  - ✓ Voting classifier
  - ✓ Ensemble:Bagging
  - ✓ Ensemble:Boosting"

- ✓ Ada Boost
- ✓ Gradient Boost
- ✓ XG Boost
- ✓ Evaluation Metrics for Classification:

- ✓ Confusion Matrix
- ✓ Accuracy & F1 Score
- ✓ Precision & Recall
- ✓ Sensitivity & Specificity
- ✓ True Positive Rate, False Positive Rate
- ✓ ROC & ROC\_AUC



## Clustering | Introduction | K-Means Clustering:

- ✓ Finding the optimal number of clusters
- ✓ Optimizing the inertia
- ✓ Cluster instability
- ✓ Elbow method

## Hierarchical Clustering | Agglomerative clustering

## DBSCAN Clustering | Association Rules

- ✓ Market Basket Analysis
- ✓ Apriori Algorithm

## Recommendation Engines

- ✓ Collaborative Filtering:
- ✓ User based collaborative filtering
- ✓ Item based collaborative filtering
- ✓ Recommendation Engines

## Time Series & Forecasting

- ✓ What is Time series data
- ✓ Different components of time series data
- ✓ Stationary of time series data
- ✓ ACF, PACF ✓ Time Series Models:
  - ✓ AR ✓ ARMA ✓ ARIMA ✓ SARIMAX

## Model Selection & Evaluation | Over Fitting & Under Fitting

- ✓ Bias-Variance Tradeoff
  - ✓ Cross Validation:
    - ✓ Stratified Cross validation
    - ✓ K-Fold Cross validation
  - ✓ Hyper Parameter Tuning
  - ✓ Joblib And Pickling

## Others

- ✓ Dummy Variable, Onehotencoding
- ✓ gridsearchcv vs randomizedsearchcv

## ML Pipeline | ML Model Deployment in Flask

# PowerBI

## Introduction

- ✓ Power BI for Data scientist
- ✓ Types of reports
- ✓ Data source types
- ✓ Installation

## Basic Report Design

- ✓ Data sources and Visual types
- ✓ Canvas and fields
- ✓ Table and Tree map
- ✓ Format button and Data Labels
- ✓ Legend,Category and Grid
- ✓ CSV and PDF Exports



## Visual Sync, Grouping

- ✓ Slicer visual
- ✓ Orientation,selection process
- ✓ Slicer:Number,Text,slicer list
- ✓ Bin count,Binning

## Hierarchies, Filters

- ✓ Creating Hierarchies
- ✓ Drill Down options
- ✓ Expand and show
- ✓ Visual filter,Page filter,Report filter
- ✓ Drill Thru Reports



## Power Query

- ✓ Power Query transformation
- ✓ Table and Column Transformations
- ✓ Text and time transformations
- ✓ Power query functions
- ✓ Merge and append transformations

## DAX Functions

- ✓ DAX Architecture, Entity Sets
- ✓ DAX Data types, Syntax Rules
- ✓ DAX measures and calculations
- ✓ Creating measures
- ✓ Creating Columns

# Deep Learning

## Deep learning at Glance

- ✓ Introduction to Neural Network
- ✓ Biological and Artificial Neuron
- ✓ Introduction to perceptron
- ✓ Perceptron and its learning rule & drawbacks
- ✓ Multilayer Perceptron, loss function
- ✓ Neural Network Activation function



## Training MLP: Backpropagation | Cost Function

## Gradient Descent Backpropagation - Vanishing & Exploding Gradient Problem

## Introduce to Py-torch | Regularization | Optimizers

## Hyperparameters and tuning of the same TENSORFLOW FRAMEWORK

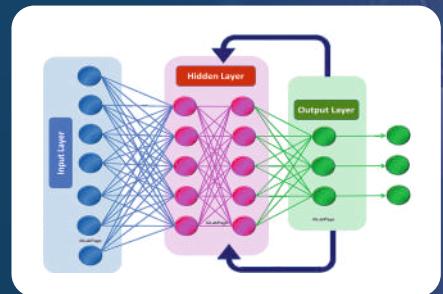
- ✓ Introduction to TensorFlow
- ✓ TensorFlow Basic Syntax
- ✓ TensorFlow Graphs
- ✓ Variables and Placeholders
- ✓ TensorFlow Playground

## ANN (Artificial Neural Network)

- ✓ ANN Architecture
- ✓ Forward & Backward Propagation, Epoch
- ✓ Introduction to TensorFlow, Keras
- ✓ Vanishing Gradient Descent
- ✓ Fine-tuning neural network hyperparameter
- ✓ Number of hidden layers, Number of neurons per hidden layer

## RNN (Recurrent Neural Network)

- ✓ Introduction to RNN
- ✓ Back Propagation through time
- ✓ Input and output sequences
- ✓ RNN vs ANN
- ✓ LSTM (Long Short-Term Memory)
- ✓ Different types of RNN: LSTM, GRU
- ✓ Biirectional RNN
- ✓ Sequential-to-sequential architecture (Encoder Decoder)
- ✓ BERT Transformers
- ✓ Text generation and classification using Deep Learning
- ✓ Generative-AI (Chat-GPT)



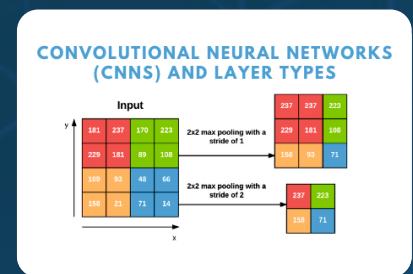
## Basics of Image Processing

- ✓ Histogram of images
- ✓ Basic filters applied on the images



## Convolutional Neural Networks (CNN)

- ✓ ImageNet Dataset
- ✓ Project: Image Classification
- ✓ Different types of CNN architectures
- ✓ Recurrent Neural Network (RNN)
- ✓ Using pre-trained model: Transfer Learning



# Natural Language Processing (NLP)

## Natural Language Processing (NLP)

- ✓ Text Cleaning
- ✓ Texts, Tokens
- ✓ Basic text classification based on Bag of Words

## Document Vectorization

- ✓ Bag of Words
- ✓ TF-IDF Vectorizer
- ✓ n-gram: Unigram, Bigram
- ✓ Word vectorizer basics, One Hot Encoding
- ✓ Count Vectorizer
- ✓ Word cloud and gensim
- ✓ Word2Vec and Glove
- ✓ Text classification using Word2Vec and Glove
- ✓ Parts of Speech Tagging (PoS Tagging or POST)
- ✓ Topic Modelling using LDA
- ✓ Sentiment Analysis

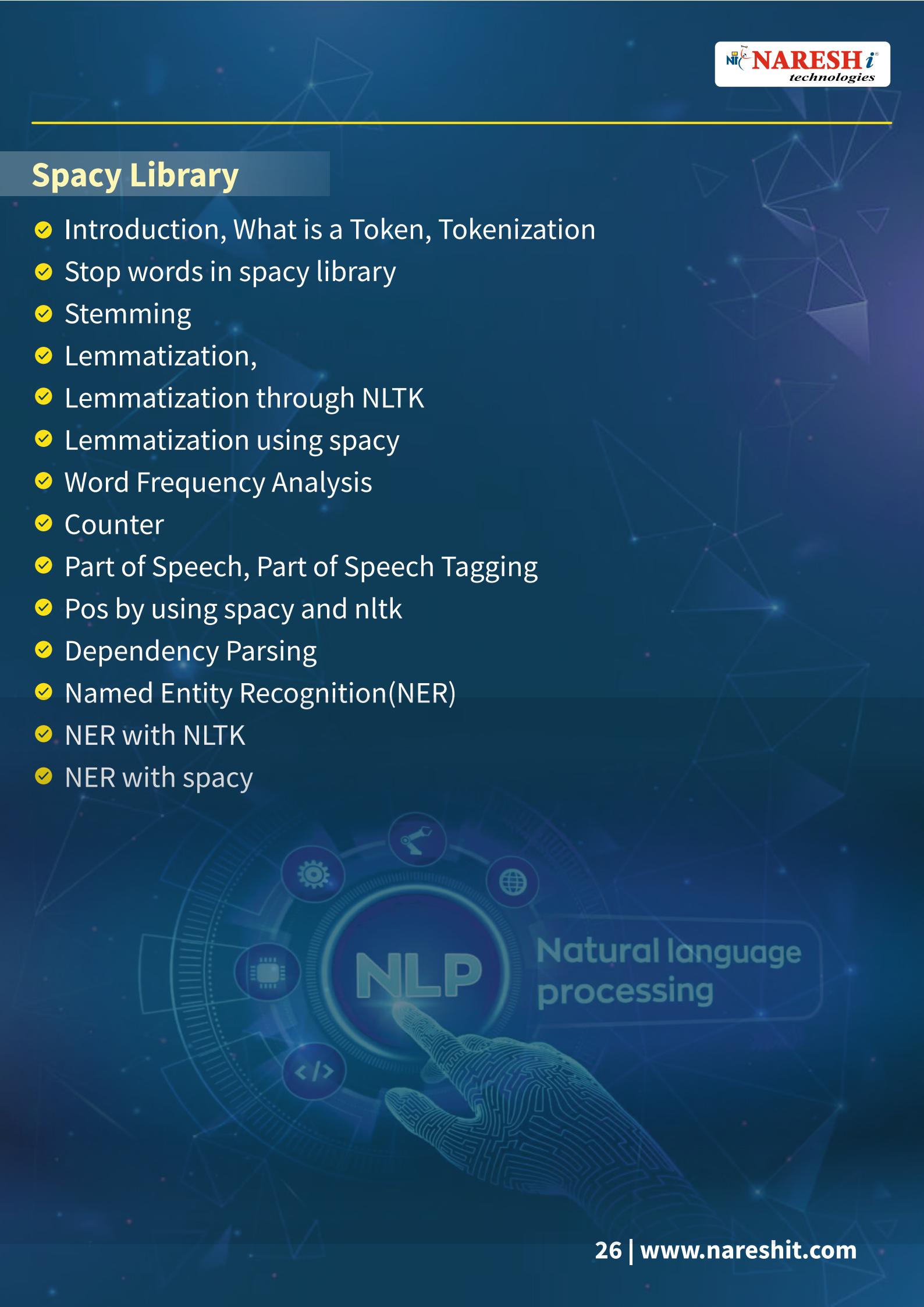


## Twitter Sentiment Analysis Using Textblob

- ✓ TextBlob
- ✓ Installing textblob library
- ✓ Simple TextBlob Sentiment Analysis Example
- ✓ Using NLTK's Twitter Corpus

## Spacy Library

- ✓ Introduction, What is a Token, Tokenization
- ✓ Stop words in spacy library
- ✓ Stemming
- ✓ Lemmatization,
- ✓ Lemmatization through NLTK
- ✓ Lemmatization using spacy
- ✓ Word Frequency Analysis
- ✓ Counter
- ✓ Part of Speech, Part of Speech Tagging
- ✓ Pos by using spacy and nltk
- ✓ Dependency Parsing
- ✓ Named Entity Recognition(NER)
- ✓ NER with NLTK
- ✓ NER with spacy



Natural language  
processing

# Computer Vision

## Human vision vs Computer vision

- ✓ CNN Architecture
- ✓ CONVOLUTION – MAX POOLING – FLATTEN LAYER – FULLY CONNECTED LAYER
- ✓ CNN Architecture
- ✓ Striding and padding
- ✓ Max pooling
- ✓ Data Augmentation
- ✓ Introduction to OpenCV & YoloV3 Algorithm



## Image Processing with OpenCV

- ✓ Image basics with OpenCV
- ✓ Opening Image Files with OpenCV
- ✓ Drawing on Images, Image files with OpenCV
- ✓ Face Detection with OpenCV

## Video Processing with OpenCV

- ✓ Introduction to Video Basics, Object Detection
- ✓ Object Detection with OpenCV

## Reinforcement Learning

- ✓ Introduction to Reinforcement Learning
- ✓ Architecture of Reinforcement Learning
- ✓ Reinforcement Learning with Open AI
- ✓ Policy Gradient Theory

## OPEN AI

- ✓ Introduction to Open AI
- ✓ Generative AI
- ✓ Chat Gpt (3.5)
- ✓ LLM (Large Language Model)
- ✓ Classification Tasks with Generative AI
- ✓ Content Generation and Summarization with Generative AI
- ✓ Information Retrieval and Synthesis workflow with Gen AI

## Time Series & Forecasting

- ✓ Time Series Forecasting using Deep Learning
- ✓ Seasonal-Trend decomposition using LOESS (STL) models.
- ✓ Bayesian time series analysis

## MakerSuite Google

- ✓ PaLM API
- ✓ MUM models Bayesian time series analysis

## Azure ML

# Advanced Course on Agentic AI: Single-Agent & Multi-Agent Systems

## Module 1: Introduction to Agentic AI

- ✓ What is Agentic AI? From Traditional AI to Autonomous AI
- ✓ Single-Agent vs Multi-Agent Systems (MAS)
- ✓ Agentic AI in Large Language Models (LLMs)
- ✓ Applications of Agentic AI in Automation, Research, and Business

### Hands-on:

- ✓ Running a basic autonomous agent using OpenAI API
- ✓ Setting up a local agent with Python

## Module 2: Building Single-Agent Systems



- ✓ Single-Agent AI: Concepts & Architectures
- ✓ Decision-Making in Single-Agent AI
- ✓ Memory & Long-Term Planning
- ✓ Integrating LLMs with Agents (LangChain, OpenAI, Hugging Face)
- ✓ Handling User Inputs & Actions with Tool-Use Capabilities

### Hands-on:

- ✓ Building a goal-driven single AI agent with LangChain
- ✓ Implementing memory-based reasoning with FAISS

## Module 3: Multi-Agent Systems (MAS)

- ✓ What are Multi-Agent Systems (MAS)?
- ✓ Agent Communication & Coordination
- ✓ Role Assignment in Multi-Agent Environments
- ✓ Task-Oriented AI vs Autonomous Decision-Making
- ✓ Swarm Intelligence & Distributed Agents

### Hands-on:

- ✓ Creating a multi-agent research assistant
- ✓ Implementing a task-based workflow with CrewAI

## Module 4: AI Agents with CrewAI

- ✓ Overview of CrewAI: AI Agents Working in Teams
- ✓ Role-Based Agent Assignments
- ✓ Orchestrating Task Execution Between Agents
- ✓ Implementing Workflow Pipelines with CrewAI



### Hands-on:

- ✓ Creating an AI-powered content generation team
- ✓ Using CrewAI for multi-step research tasks

## Module 5: SmolAgent – Lightweight AI Agents

- ✓ Introduction to SmolAgent: Minimalistic AI Agents
- ✓ When to Use SmolAgent vs Heavyweight AI Agents
- ✓ Optimizing AI Agents for Cost and Performance
- ✓ Combining SmolAgent with LLMs for Fast Execution

### Hands-on:

- ✓ Deploying a SmolAgent-based chatbot
- ✓ Running SmolAgent on an edge device

## Module 6: Phi Data – Memory & Context Optimization

- ✓ Introduction to Phi Data: AI Agent Memory & Learning
- ✓ Vector Database Integration (FAISS, Pinecone, ChromaDB)
- ✓ Retrieval-Augmented Generation (RAG) for AI Agents
- ✓ Personalized AI Assistants with Memory

### Hands-on:

- ✓ Implementing Phi Data for long-term memory
- ✓ Storing and retrieving context for AI-powered Q&A

## Module 7: Building & Deploying Agentic AI Applications

- ✓ Deploying AI Agents on Cloud (Hugging Face, AWS, Azure)
- ✓ Integrating AI Agents into Web Applications (FastAPI, Streamlit)
- ✓ Security Considerations in Autonomous AI

# Mastering Prompt Engineering for LLMs

## Module 1: Introduction to Prompt Engineering

### ➲ What is Prompt Engineering?

- ✓ The role of prompts in LLMs
- ✓ How LLMs process and interpret prompts

### ➲ Why Learn Prompt Engineering?

- ✓ Optimizing LLM performance
- ✓ Reducing hallucinations & improving accuracy
- ✓ Enhancing AI-driven applications

#### Hands-on:

- ✓ Experimenting with OpenAI's API for basic text generation

## Module 2: LLM Model Settings & Configurations

### ➲ Understanding LLM Settings:

- ✓ Temperature, Top-k, Top-p (Nucleus Sampling)
- ✓ Stop Tokens, Context Length & Tokenization

### ➲ Fine-tuning vs Prompt Optimization

Customizing Model Behavior with System Prompts

#### Hands-on:

- ✓ Experimenting with different model settings in OpenAI Playground

## Module 3: Prompt Elements & Structuring

⇒ Key Components of a Well-Designed Prompt:

✓ Instructions ✓ Context ✓ Input Data

⇒ Optimizing Prompts for Accuracy & Consistency

### Hands-on:

✓ Structuring prompts for summarization, classification, & question-answering

## Module 4: Prompt Engineering Techniques (Shot-Based Prompting)

- ✓ Zero-Shot Prompting
- ✓ One-Shot Prompting
- ✓ Few-Shot Prompting
- ✓ Comparing Shot-Based Techniques for Different Use Cases

### Hands-on:

✓ Designing and testing different shot-based prompts on GPT models

## Module 5: Chain of Thought (CoT) Prompting

- ✓ What is CoT Prompting?
- ✓ Step-by-Step Reasoning in LLMs
- ✓ Implementing CoT in Math, Logic & Coding Tasks

### Hands-on:

✓ Using CoT to improve reasoning-based problem-solving

## Module 6: Self-Consistency in Prompt Engineering

- ✓ What is Self-Consistency?
- ✓ Generating Multiple Answers & Selecting the Best
- ✓ Improving Output Reliability with Self-Consistency

### Hands-on:

- ✓ Implementing Self-Consistency for multi-answer tasks

## Module 7: Out-of-Date Learning in Prompt Engineering

- ✓ How LLMs Handle Outdated Information
- ✓ Strategies to Overcome Out-of-Date Learning:
  - ✓ Prompting with External Data
  - ✓ Fine-Tuning vs Retrieval-Augmented Generation (RAG)
- ✓ When to Use Updated APIs & Tools

### Hands-on:

- ✓ Experimenting with model responses on time-sensitive queries

## Module 8: Role-Playing in Prompt Engineering

- ✓ What is Role-Playing in Prompting?
- ✓ Creating AI Personas for Specialized Tasks
- ✓ Enhancing Response Accuracy with Role-Based Prompts

### Hands-on:

- ✓ Designing AI assistants with different personas (e.g., Doctor, Lawyer, Coder)

## Module 9: RAG (Retrieval-Augmented Generation) in Prompt Engineering

- ✓ What is RAG & Why is it Important?
- ✓ Integrating Vector Databases (FAISS, Pinecone, ChromaDB)
- ✓ Enhancing AI Responses with External Knowledge

### Hands-on:

- ✓ Implementing a RAG-based chatbot using FAISS & OpenAI

## Module 10: ReAct (Reasoning + Acting) in Prompt Engineering

- ✓ What is ReAct Framework?
- ✓ Combining CoT + Tool Use for Autonomous Agents
- ✓ Building AI Agents that Reason & Execute Actions

### Hands-on:

- ✓ Implementing a ReAct-based agent using LangChain

## Module 11: DSP (Dynamic Structured Prompting)

- ✓ What is Dynamic Structured Prompting (DSP)?
- ✓ Generating Structured & Dynamic Prompts Based on Context
- ✓ Using DSP for Adaptive AI Interactions

### Hands-on:

- ✓ Creating dynamically structured prompts for personalized AI responses

# Advanced Course on Generative AI:

## Module 1: Introduction to Generative AI

- ✓ What is Generative AI?
- ✓ Types of Generative AI Models:
  - ✓ Text-based (GPT, LLaMA, Claude)
  - ✓ Multimodal (CLIP, DALL·E, Stable Diffusion)
    - Text-based (GPT, LLaMA, Claude)
    - Multimodal (CLIP, DALL·E, Stable Diffusion)
  - ✓ Use Cases in NLP, Image Generation, & Code Generation

### Hands-on:

- ✓ Running a simple text-based generative model using OpenAI API

## Module 2: Text-Based Generative Models

- ✓ How Text-Based Models Work
- ✓ Training LLMs (Large Language Models) with Transformers
- ✓ Pretrained Models vs Fine-Tuned Models
- ✓ Popular LLMs: GPT-4, LLaMA, Mistral, Falcon

### Hands-on:

- ✓ Generating text using Hugging Face Transformers

## Module 3: Multimodal Models (Text + Image + Audio)

- ✓ What are Multimodal Models?
- ✓ Combining Text & Vision Models for AI Applications
- ✓ Examples of Multimodal Models: CLIP, DALL·E, Gemini, GPT-4 Turbo

**Hands-on:**

- ✓ Running OpenAI's CLIP model for text-to-image retrieval

**Module 4: CLIP (Contrastive Language-Image Pretraining) Architecture**

- ✓ How CLIP Works: Text-Image Pairing
- ✓ Applications of CLIP in Image Search & Generation
- ✓ Fine-tuning CLIP for Custom Tasks

**Hands-on:**

- ✓ Using CLIP to find relevant images based on text prompts

**Module 5: VQGAN & Taming Transformers**

- ✓ Introduction to VQGAN (Vector Quantized GAN)
- ✓ How Taming Transformers Improve Image Quality
- ✓ Combining VQGAN + CLIP for AI Art

**Hands-on:**

- ✓ Generating AI Art using VQGAN + CLIP

**Module 6: Autoencoders & VAEs (Variational Autoencoders)**

- ✓ What is an Autoencoder?
- ✓ Difference Between Autoencoders & VAEs
- ✓ Generating High-Resolution Images with VAEs

**Hands-on:**

- ✓ Implementing a simple Variational Autoencoder (VAE) in PyTorch

## Module 7: Retrieval-Augmented Generation (RAG)

- ✓ What is RAG & Why It Matters for AI?
- ✓ Enhancing LLMs with External Knowledge
- ✓ Vector Databases for RAG (FAISS, Pinecone, ChromaDB)

### Hands-on:

- ✓ Implementing a RAG-based chatbot using LlamaIndex & FAISS

## Module 8: Hugging Face Ecosystem

- ✓ Overview of Hugging Face Transformers
- ✓ Fine-tuning LLMs with Hugging Face
- ✓ Deploying Models Using Hugging Face Spaces

### Hands-on:

- ✓ Fine-tuning a text generation model on Hugging Face

## Module 9: CrewAI for Multi-Agent AI Systems

- ✓ What is CrewAI?
- ✓ Building Teams of AI Agents
- ✓ Role-Based Task Assignment in CrewAI

### Hands-on:

- ✓ Setting up an AI research team using CrewAI

## Module 10: Groq – High-Speed AI Inference

- ✓ What is Groq?
- ✓ Running AI Models at Lightning Speed
- ✓ Optimizing Large Models for Low Latency

### Hands-on:

- ✓ Deploying a transformer model with Groq hardware

## Module 11: Stable Diffusion for Image Generation

- ✓ Understanding Stable Diffusion Architecture
- ✓ Text-to-Image Generation with Diffusion Models
- ✓ Fine-Tuning & Customizing Stable Diffusion

### Hands-on:

- ✓ Running Stable Diffusion on a local machine

## Module 12: GitHub Copilot for AI-Powered Coding

- ✓ How GitHub Copilot Uses AI for Code Generation
- ✓ Best Practices for Using AI in Software Development
- ✓ Comparing Copilot with Other AI Coding Tools

### Hands-on:

- ✓ Writing AI-assisted Python scripts using GitHub Copilot

## Module 13: LlamaIndex – AI-Powered Document Processing

- ✓ What is LlamaIndex?
- ✓ Connecting LLMs to Private Data
- ✓ Using LlamaIndex for Enterprise AI Applications

### Hands-on:

- ✓ Implementing LlamaIndex for a document-based AI assistant

## Module 14: FastAPI for AI Model Deployment

- ✓ Introduction to FastAPI for AI
- ✓ Building a REST API for LLMs
- ✓ Deploying AI Models as Web Services

### Hands-on:

- ✓ Deploying a text-based LLM using FastAPI

# Advanced Course on : LLMs

## Module 1: Introduction to Generative AI & LLMs

- ✓ What is Generative AI?
- ✓ Types of Generative AI Models (Text, Image, Multimodal, Speech)
- ✓ Comparison of Leading AI Models (GPT, Gemini, LLaMA, Claude, Mixtral, DeepSeek, Grok)
- ✓ Foundation Models vs Fine-Tuned Models

### Hands-on:

- ✓ Running a basic LLM-powered chatbot using OpenAI API

## Module 2: OpenAI's AI Ecosystem | LangChain Framework

- ✓ Introduction to LangChain
- ✓ Building AI Agents with LangChain
- ✓ Memory & Context Handling in LangChain
- ✓ Connecting LLMs with External Data Sources

### Hands-on:

- ✓ Implementing a chatbot using OpenAI + LangChain
- OpenAI Whisper (Speech-to-Text AI)
  - ✓ What is OpenAI Whisper?
  - ✓ Multilingual Speech Recognition
  - ✓ Building Real-World Applications with Whisper

### Hands-on:

- ✓ Transcribing audio into text using OpenAI Whisper

## Module 3: Google's Generative AI Ecosystem

- ✓ Gemini AI: Google's Multimodal LLM
- ✓ Introduction to Gemini AI
- ✓ Comparison with OpenAI & Meta Models
- ✓ Using Gemini API for Text & Image Generation

### Hands-on:

- ✓ Generating text & images using Gemini API
- Google Vision: AI for Image Analysis & Recognition
  - ✓ ·What is Google Vision?
  - ✓ ·AI-Powered Image Processing & OCR
  - ✓ ·Building AI-powered Image Search Systems

### Hands-on:

- ✓ Using Google Vision API for image classification

## Module 4: DeepSeek, Mistral, Mixtral, Grok & Claude

- ✓ DeepSeek AI
  - ✓ Overview of DeepSeek Language Model
  - ✓ Optimizing Search and AI Retrieval with DeepSeek
  - ✓ Applications in AI-Assisted Knowledge Systems

### Hands-on:

- ✓ Running DeepSeek for document-based AI search
- Mistral &Mixtral: High-Performance Open-Source AI
  - ✓ What is Mistral &Mixtral?
  - ✓ Dense vs Sparse Transformer Models
  - ✓ Optimizing Mixtral for Multi-Task AI

### Hands-on:

- ✓ Running Mixtral models on Hugging Face
- Grok (X AI by Elon Musk)
  - ✓ How Grok is Designed for Real-Time AI Processing
  - ✓ Comparison with GPT, Gemini & Claude
  - ✓ Use Cases of Grok in AI Chatbots & Assistants

### Hands-on:

- ✓ Running Grok on X (formerly Twitter) API
- Claude (Anthropic AI)
  - ✓ What is Claude & How It Differs from Other LLMs?
  - ✓ Claude's Approach to AI Safety & Constitutional AI
  - ✓ Fine-Tuning Claude for Enterprise Applications

### Hands-on:

- ✓ Building an AI-powered assistant with Claude API

## Module 5: Meta's Generative AI Ecosystem

- ✓ LLaMA 3: Open-Source LLM by Meta
  - ✓ Introduction to LLaMA 3
  - ✓ Comparison with GPT-4, Gemini, Mixtral, & Claude
  - ✓ Fine-Tuning & Customizing LLaMA for Specific Tasks

### Hands-on:

- ✓ Running LLaMA 3 on a local machine using Hugging Face
- Building Generative AI on Cloud
  - ✓ Cloud Platforms for AI (AWS, GCP, Azure, Meta Cloud)
  - ✓ Deploying LLMs on Cloud for Scalability
  - ✓ Building AI-Driven Web Apps with Cloud-Based LLMs

### Hands-on:

- ✓ Deploying an LLM-powered chatbot on Cloud
- Meta's Foundation Models
  - ✓ Understanding Meta's AI Foundation Models
  - ✓ Pre-Trained Models vs Custom Models
  - ✓ Adapting Foundation Models for Industry Use Cases

### Hands-on:

- ✓ Using Meta's AI Models for custom NLP tasks

## Module 6: Fine-Tuning LLMs with Quantization, LoRA&QLoRA

- ✓ Fine-Tuning Large Language Models (LLMs)
  - Why Fine-Tune an LLM?
  - Datasets & Preprocessing for LLM Fine-Tuning
  - Fine-Tuning vs Prompt Engineering
- ✓ LoRA (Low-Rank Adaptation) Fine-Tuning
  - What is LoRA?
  - Reducing Computation for LLM Training
  - Implementing LoRA with Hugging Face &PyTorch
- ✓ QLoRA (Quantized LoRA) for Efficient Model Fine-Tuning
  - What is QLoRA?
  - Memory Optimization for Large LLMs
  - Running Fine-Tuned Models on Low-End Hardware

### Hands-on:

- ✓ Fine-tuning a LLaMA 3 model using LoRA&QLoRA

## Module 7: End-to-End AI Model Deployment & Optimization

- ✓ Deploying AI Models with APIs (FastAPI, Flask)
- ✓ Optimizing AI Models for Performance & Cost
- ✓ Best Practices for AI Model Security & Governance

### Hands-on:

- ✓ Deploying a fine-tuned LLM as a FastAPI web service

# Vector database

## Module 1: Introduction to Vector Databases

- ✓ What is a Vector Database?
- ✓ Difference Between Traditional & Vector Databases
- ✓ Why Use Vector Databases in Generative AI & LLMs?
- ✓ How Vector Embeddings Work in AI Search & Retrieval

### Hands-on:

- ✓ Generating vector embeddings from text using OpenAI's text-embedding-ada-002

## Module 2: Understanding Vector Embeddings

- ✓ What Are Embeddings in AI?
- ✓ How LLMs Convert Text, Images & Audio to Vectors
- ✓ Similarity Metrics: Cosine Similarity, Euclidean Distance, Dot Product
- ✓ Choosing the Right Embedding Model (OpenAI, Hugging Face, SentenceTransformers, BERT, etc.)

### Hands-on:

- ✓ Generating embeddings with OpenAI, Hugging Face, and BERT models

## Module 3: Implementing FAISS (Facebook AI Similarity Search)

- ✓ What is FAISS & How It Works?
- ✓ Indexing & Searching Large-Scale Vectors with FAISS
- ✓ Optimizing FAISS for Fast Retrieval

### Hands-on:

- ✓ Implementing a FAISS-based search engine for document retrieval

## Module 4: Using Pinecone for Scalable AI Search

- ✓ Introduction to Pinecone: A Managed Vector Database
- ✓ Building Real-Time AI Search Applications with Pinecone
- ✓ Comparing FAISS vs Pinecone vs Milvus

### Hands-on:

- ✓ Creating a question-answering chatbot using OpenAI + Pinecone

## Module 5: ChromaDB for LLMs & AI Applications

- ✓ What is ChromaDB?
- ✓ How ChromaDB Works with LangChain
- ✓ Building RAG (Retrieval-Augmented Generation) Pipelines with ChromaDB

### Hands-on:

- ✓ Integrating ChromaDB with OpenAI's GPT for AI-powered search

## Module 6: Exploring Weaviate & Its AI Capabilities

- ✓ Overview of Weaviate as a Hybrid Search Engine
- ✓ Using Weaviate for Semantic Search & Knowledge Graphs
- ✓ Deploying Weaviate on Cloud & Local Environments

### Hands-on:

- ✓ Implementing a semantic search engine with Weaviate

## Module 7: Milvus & Qdrant for Large-Scale AI Applications

- ✓ Milvus (Distributed & Cloud-Based Vector Search)
  - ✓ Introduction to Milvus for AI Applications
  - ✓ Optimizing Milvus for Large-Scale Data Processing
- ✓ Qdrant (High-Performance Open-Source Vector DB) What is Qdrant & How It Works?
  - ✓ Fine-Tuning Qdrant for AI Search & Recommendation Systems

### Hands-on:

- ✓ Deploying Milvus & Qdrant for AI-driven search and recommendations

## Module 8: Building AI-Powered Search & RAG Applications

- ✓ What is Retrieval-Augmented Generation (RAG)?
- ✓ Integrating Vector Databases with LLMs for Intelligent Search
- ✓ Building Enterprise AI Assistants Using Vector Databases

### Hands-on:

- ✓ Building an RAG pipeline using LangChain, Pinecone, and OpenAI

# MLOps & CI/CD Pipeline for AI & Machine Learning

## Module 8: Building AI-Powered Search & RAG Applications

- ✓ What is MLOps & Why is it Important?
- ✓ DevOps vs MLOps: Key Differences
- ✓ MLOps Lifecycle & Stages
  - Understanding CI/CD Pipelines in AI/ML Projects
  - Challenges in Deploying ML Models at Scale

### Hands-on:

- ✓ Setting up a basic CI/CD pipeline for ML models using GitHub Actions

## Module 2: Containerization & Model Packaging with Docker

- ✓ Why Containerize ML Models?
- ✓ Building & Running Docker Containers for ML Applications
- ✓ Deploying ML Models inside Docker Containers
- ✓ Optimizing Containers for AI Workloads

### Hands-on:

- ✓ Containerizing an ML Model with Docker & Running it Locally

## Module 3: CI/CD Pipeline for ML with GitHub Actions & Jenkins

- ✓ What is CI/CD & How Does It Work in ML?
- ✓ Setting up a CI/CD Pipeline for ML Models
- ✓ Automating Model Testing, Validation & Deployment
- ✓ CI/CD with GitHub Actions,

### Hands-on:

- ✓ Implementing a CI/CD Pipeline for an AI Model Deployment Using GitHub Actions

## Module 4: MLOps with MLflow

- ✓ Introduction to Kubeflow for AI & ML
- ✓ Integrating with MLflow for Experiment Tracking

### Hands-on:

- ✓ Building an ML Workflow with Kubeflow Pipelines

# Students Placed in -2024



## INTERNSHIP PROGRAM



**PYTHON | EDA | DATA SCIENCE | DATA ANALYTICS**

Academic Pursuing Students (1st Year/2nd Year/3rd Year)

📍 Ameerpet | 📍 K.P.H.B | 📍 JNTU



# Google Reviews

4.4 ★★★★★ (6,755)

NareshIT

★★★★★

I have learned Full stack data science course in Naresh IT .It the best institute to have the great trainer like @prakash senapathi sir . he can take the institutes pride to the greatest Read more...

V Vinay Raj 147

2 months ago

★★★★★

I have learned the full stack data science under the guidance of Prakash Senapathi . Sir's class was truly excellent. His clear explanations and interactive teaching methods Read more...

B Bhumi Smiley

3 month ago

★★★★★

I am incredibly fortunate to have had Mr. Prakash Senapati as my instructor for the 'Data Analytics & Business Intelligence'. His dedication to teaching and passion for the subject Read more...

S Sridhar padhi

5 Years ago

★★★★★

Hi Guys My name is Dinesh Sharma Let me share My experiences, about NareshIT's, I enrolled in Data Science course under Prakash Senapati Sir's guidance. It was amazing experience Read more...

Dinesh Sharma

5 months ago

★★★★★

I had an incredible experience learning Full Stack Data Science. I feel fortunate to have had the chance to learn from a wonderful teacher and mentor. I'm always grateful for the Read more...

A Akshay Narot

3 month ago

★★★★★

One of the best faculty in nareshIT for Data Science is KODI PRAKASH SENAPATI sir. He has charisma to teach the course so interesting and I never felt how 5 months completed. I Read more...

A Alan Mcman

7 years ago

★★★★★

I recently completed a full-stack Data Science course under the guidance of Mr. Prakash Senapati Sir. The course was conducted at 10:00 AM. Now, I am placed as a data scientist in Read more...

Harsh Mishra

a year ago

★★★★★

I have completed the Full Stack Data Science and AI course, the instructor was Prakash Senapati sir. Learning this course was fun and interesting as sir used to assign daily tasks, provide Read more...

★★★★★

I am a working IT professional from Dallas USA. Currently I am taking Full Stack Data Science course from NareshIT Hyderabad. I am really impressed by Prakash Read more...

★★★★★

I belong to IT background with 5 years career gap. After 5 years, I joined a full stack data science and AI course under the guidance of Mr. Kodi Prakash Senapati sir. From the first Read more...

Anjali Sharma

a year ago

★★★★★

faculty: Mr. Prakash Senapathi I joined Full stack data science course. He is very friendly and professional. This course helps to learn from basics as I am a non-technical student. Live Read more...

M M N S RAMYA

2 weeks ago

★★★★★

I have completed Full stack Data Science & AI course under Prakash Senapati sir. He covered Python basics, Data Cleansing, ML, NLP, Deep Learning (ANN, CNN, RNN) in detail. Read more...

Amit Mishra

a year ago

# COURSES OFFERED



Classroom Training



Online Training



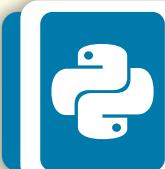
Hybrid Training



Full Stack  
**JAVA | .NET**



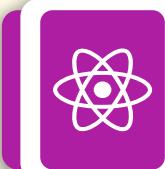
Full Stack  
**DATA SCIENCE**



Full Stack  
**PYTHON**



**GEN AI**



UI Full Stack  
with **React**



**DATA ANALYTICS**



**DEVOPS**



**BUSINESS ANALYTICS**



**AWS**



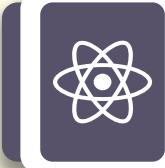
**Power BI**



Multi  
**CLOUD**



**MS Azure DevOps**



**React JS**



**Azure Data Engineer**



**C | DSA**



**CRT**

**Head Office :**

2nd Floor, Durga Bhavani Plaza, Ameerpet,  
Hyderabad, 500016.

**Ameerpet Branch :**

Ground Floor, Manjeera square, opp Prime Hospital,  
Ameerpet, Hyd.

**KPHB Branch :**

2<sup>nd</sup> Floor, Sreeramoju Complex, K P H B Phase 1,  
Hyderabad, 500072.

**JNTU Branch :**

4th Floor, Plot No. 47-48, Above HDFC Bank,  
JNTU , Hyd.

**USA :**

5007 Arbor View Pkwy NW Acworth, GA, 30101

