Machine Learning & Generative AI Program

Week 1: Foundations - Python (Data Science Foundations)

Saturday (2 hours)

- **Introduction to Python**
  - Python installation and setup (Jupyter Notebook, Anaconda, IDEs)
  - Variables, data types (int, float, string, boolean), and type conversion
  - Arithmetic operations, string manipulation, and basic I/O functions

- **Control Flow**
  - Conditionals: if, else, elif
  - Loops: for and while
  - Nested loops and conditionals

Sunday (2 hours)

- **Functions and Modules**
  - Defining functions, function arguments, return values
  - Using built-in functions and importing standard modules (math, os, etc.)
  - Lambda functions and map/filter/reduce

- **Data Structures**
  - Lists: indexing, slicing, and operations on lists
  - Tuples and sets: creation and usage scenarios
  - Dictionaries: key-value pairs, methods for accessing and modifying dictionaries
- Introduction to **NumPy** and **Pandas** for data manipulation

Outcome:
By the end of the week, students will have a solid grasp of Python syntax, control flow, and data structures essential for data manipulation. They will be familiar with basic programming techniques and data manipulation using NumPy and Pandas required for data science workflows.

Week 2: Foundations - Linear Algebra

Saturday (2 hours)

- **Vectors and Matrices**
  - Introduction to tensors, vectors and matrices
  - Operations on vectors: addition, scalar multiplication, dot product
  - Common tensor Operations - transposition, arithmetic, reduction, dot product
  - Matrix representation, matrix addition, and multiplication

- **Matrix Inverses and Determinants**
  - Computing determinants and inverses of matrices
  - Importance of invertibility in machine learning algorithms
  - Applications in solving systems of equations

Sunday (2 hours)

- **Eigenvalues and Eigenvectors**
  - Definition and importance in data science
  - Calculation of eigenvalues and eigenvectors
  - Principal Component Analysis (PCA) concept based on eigenvalues

- **Singular Value Decomposition (SVD)**
  - Understanding SVD and its applications in dimensionality reduction
  - Use cases in recommendation systems and image compression

Outcome:
By the end of the week, students will have a solid grasp of Python syntax, control flow, and data structures essential for data manipulation. They will be familiar with basic programming techniques and data manipulation using NumPy and Pandas required for data science workflows.

Week 3: Foundations - Calculus

Saturday (2 hours)

- **Introduction to Derivatives**
  - Definition and basic rules (power rule, product rule, chain rule)
  - Partial derivatives and multivariate calculus
  - Applications of derivatives in optimization

- **Gradient Descent**
  - Concept of gradients and its importance in machine learning
  - Derivation of gradient descent and learning rate
  - Hands-on gradient descent algorithm implementation

Sunday (2 hours)

- **Integration**
  - Definite and indefinite integrals, integration by parts
  - Applications in calculating area under the curve (important in loss functions)
  - Numerical integration methods

- **Optimization Techniques**
  - Optimization in machine learning: global vs local minima
  - Stochastic Gradient Descent (SGD) and its variants
  - Hands-on implementation with machine learning models

Outcome:
Students will understand the fundamental calculus concepts such as derivatives and integrals, and their applications in optimization and machine learning model training.

Week 4: Foundations - Probability

Saturday (2 hours)

- **Probability Basics**
  - Definition of probability, outcomes, and events
  - Conditional probability and independence of events
  - Bayes' Theorem and its application in machine learning

- **Probability Distributions**
  - Uniform, Bernoulli, Binomial, and Normal distributions
  - Probability density functions and cumulative distribution functions (CDF)
  - Central Limit Theorem

Sunday (2 hours)

- **Random Variables and Expectation**
  - Definition of discrete and continuous random variables
  - Expected value, variance, and standard deviation
  - Law of large numbers

- **Application in Machine Learning**
  - Probabilistic reasoning in machine learning models
  - Generative vs Discriminative models (Naive Bayes, Bayesian Networks)
  - Application of probability in data generation and classification problems

Outcome:
Students will gain a solid foundation in probability theory, crucial for understanding machine learning models, data distributions, and uncertainty in predictions.

Week 5: Foundations - Statistics

Saturday (2 hours)

- **Descriptive Statistics**
  - Measures of central tendency: mean, median, mode
  - Measures of spread: variance, standard deviation, range, IQR
  - Skewness and kurtosis

- **Data Distribution and Visualization**
  - Histograms, boxplots, scatter plots
  - Visualizing distribution with normal curves
  - Z-scores and standard deviation distances

Sunday (2 hours)

- **Inferential Statistics**
  - Hypothesis testing, null and alternative hypotheses
  - p-values, significance levels, and statistical power
  - t-tests, z-tests, and chi-square tests

- **Correlation and Causation**
  - Pearson and Spearman correlation coefficients
  - Difference between correlation and causation
  - Regression vs correlation

Outcome:
Students will learn to summarize and interpret data, conduct hypothesis tests, and analyze correlations in datasets, giving them the skills to draw meaningful conclusions from data.

Week 6: Exploratory Data Analysis (EDA) & Visualizations

Saturday (2 hours)

- **Data Cleaning and Preprocessing**
  - Handling missing data: imputation techniques
  - Removing duplicates and outlier detection methods
  - Data normalization and standardization

- **Feature Engineering**
  - Creating new features from existing ones (feature extraction)
  - Binning, encoding categorical data, handling skewed features
  - Hands-on example with Pandas for preprocessing

Sunday (2 hours)

- **Data Visualization**
  - Introduction to visualization tools: Matplotlib, Seaborn
  - Visualizing distributions, relationships, and trends in data
  - Creating heatmaps, pairplots, and time-series visualizations

- **Exploratory Data Analysis (EDA)**
  - Identifying patterns, correlations, and anomalies
  - Importance of EDA in preparing data for machine learning
  - Hands-on EDA with a dataset (e.g., Iris or Titanic dataset)

Outcome:
Students will develop skills in data cleaning, preprocessing, and creating visualizations to extract insights from data. This week will help them prepare datasets for machine learning tasks effectively.

Week 7: Supervised Learning - Classification

Saturday (2 hours)

- **Introduction to Supervised Learning**
  - Difference between supervised, unsupervised, and reinforcement learning
  - Types of classification problems (binary, multiclass, multilabel)
  - k-Nearest Neighbors (KNN) classifier: algorithm and implementation

- **Logistic Regression**
  - Sigmoid function and decision boundary
  - Maximum likelihood estimation
  - Logistic Regression for binary classification

Sunday (2 hours)

- **Support Vector Machines (SVM)**
  - Linear SVMs and the concept of hyperplanes
  - Non-linear SVM using kernel trick
  - Practical implementation of SVM with Sklearn

- **Model Evaluation for Classification**
  - Confusion matrix, accuracy, precision, recall, and F1 score
  - ROC-AUC curve, sensitivity and specificity
  - Hands-on model evaluation with a sample dataset

Outcome:
Students will gain practical experience in building classification models, including KNN, Logistic Regression, and SVMs. They will also learn how to evaluate model performance using classification metrics.

Week 8: Supervised Learning - Regression

Saturday (2 hours)

- **Linear Regression**
  - Introduction to regression and linear models
  - Simple linear regression: fitting a line to data
  - Assumptions of linear regression (linearity, independence, homoscedasticity)
  - Ordinary Least Squares (OLS) method

- **Multiple Linear Regression**
  - Extending linear regression to multiple variables
  - Multicollinearity: detection using VIF (Variance Inflation Factor)
  - Model evaluation metrics: R-squared, Adjusted R-squared

Sunday (2 hours)

- **Regularization Techniques**
  - Lasso (L1) and Ridge (L2) regularization
  - Trade-off between bias and variance
  - Implementing regularization using Scikit-Learn

- **Polynomial Regression**
  - Understanding the need for non-linear regression
  - Polynomial regression for capturing non-linear relationships
  - Hands-on implementation with a dataset

Outcome:
Students will gain proficiency in applying linear and multiple regression techniques, along with regularization methods (Lasso, Ridge) to prevent overfitting, and extend their understanding to polynomial regression for non-linear data.

Week 9: Unsupervised Learning

Saturday (2 hours)

- **Introduction to Unsupervised Learning**
  - Key differences between supervised and unsupervised learning
  - Types of unsupervised learning tasks: clustering, dimensionality reduction
  - Introduction to clustering methods

- **k-Means Clustering**
  - k-Means algorithm: steps and intuition
  - Choosing the right number of clusters using the Elbow method
  - Practical implementation of k-Means with Scikit-Learn

Sunday (2 hours)

- **Hierarchical Clustering**
  - Agglomerative vs divisive clustering
  - Dendrograms: how to interpret and cut dendrograms
  - Hands-on implementation of hierarchical clustering

- **Dimensionality Reduction with PCA**
  - Introduction to Principal Component Analysis (PCA)
  - Visualizing data in lower dimensions (2D, 3D)
  - Implementing PCA and interpreting principal components

Outcome:
Students will be able to apply clustering techniques like k-Means and hierarchical clustering to uncover patterns in data, and reduce the dimensionality of data using PCA.

Week 10: Decision Trees and Random Forest

Saturday (2 hours)

- **Decision Trees**
  - Structure of decision trees: nodes, branches, leaves
  - Splitting criteria: Gini Index, Information Gain, and Entropy
  - Overfitting in decision trees and pruning methods

- **Building Decision Trees**
  - Hands-on implementation using Scikit-Learn
  - Visualizing decision trees
  - Importance of feature selection in decision trees

Sunday (2 hours)

- **Random Forest**
  - Concept of ensemble learning: bagging and boosting
  - Random Forest algorithm: creating multiple decision trees
  - Out-of-bag error, feature importance in random forests

- **Hyperparameter Tuning in Random Forests**
  - Tuning parameters: number of trees, depth, and split criteria
  - Using GridSearchCV for hyperparameter tuning
  - Practical implementation with a dataset

Outcome:
Students will understand the workings of decision trees and random forests, along with their practical applications in classification and regression tasks. They will learn how to tune hyperparameters for better performance.

Week 11: Time Series

Saturday (2 hours)

- **Introduction to Time Series Analysis**
  - Definition of time series data: trends, seasonality, noise
  - Components of time series: trend, seasonal, residual
  - Time series data preprocessing: indexing, handling missing values

- **Stationarity and Differencing**
  - Stationary vs non-stationary time series
  - Augmented Dickey-Fuller (ADF) test for stationarity
  - Differencing to achieve stationarity

Sunday (2 hours)

- **ARIMA Models**
  - AutoRegressive Integrated Moving Average (ARIMA) model
  - Model selection using ACF and PACF plots
  - Fitting an ARIMA model and forecasting future values

- **Seasonal Decomposition of Time Series (STL)**
  - Decomposing time series data into trend, seasonal, and residual components
  - Seasonal ARIMA (SARIMA) for handling seasonality
  - Hands-on implementation with a time series dataset

Outcome:
Students will learn how to analyze and forecast time series data using ARIMA and seasonal decomposition techniques. They will gain hands-on experience in making predictions from time series data.

this week>

Learning

BREAK

Week 12: Neural Networks

Saturday (2 hours)

- **Introduction to Neural Networks**
  - Biological vs artificial neurons
  - Structure of a perceptron: inputs, weights, activation function
  - Feedforward neural networks

- **Activation Functions**
  - Linear vs non-linear activation functions (ReLU, Sigmoid, Tanh)
  - Role of activation functions in introducing non-linearity
  - Implementing feedforward networks with a single hidden layer

Sunday (2 hours)

- **Backpropagation**
  - Concept of backpropagation in neural networks
  - Deriving gradients using the chain rule
  - Implementing backpropagation with gradient descent

- **Overfitting and Regularization in Neural Networks**
  - Preventing overfitting: dropout, weight decay, early stopping
  - Using regularization techniques (L2 regularization)
  - Hands-on implementation of neural networks with regularization techniques

Outcome:
Students will be proficient in building and training feedforward neural networks, understanding how backpropagation works, and implementing regularization techniques to prevent overfitting.

Week 13: Convolutional Neural Networks (CNNs)

Saturday (2 hours)

- **Introduction to CNNs**
  - Basics of CNNs: convolution, pooling, and fully connected layers
  - Convolution operation: kernels, feature maps
  - Pooling layers: max-pooling, average-pooling

- **CNN Architectures**
  - CNN architecture overview: LeNet, AlexNet, VGG
  - Hands-on implementation of a basic CNN for image classification
  - Understanding hyperparameters in CNNs (filter size, stride, padding)

Sunday (2 hours)

- **Advanced CNN Techniques**
  - Transfer learning: using pre-trained CNN models
  - Data augmentation for CNNs
  - Implementing a transfer learning model with a custom dataset

Outcome:
Students will gain practical experience in building CNN models from scratch for image classification tasks. They will also understand how to use transfer learning to leverage pre-trained models.

Week 14: Graph Neural Networks (GNNs)

Saturday (2 hours)

- **Introduction to Graphs**
  - Definition of graph structures: nodes and edges
  - Types of graphs: undirected, directed, weighted, etc.
  - Applications of graphs in social networks, knowledge graphs, etc.

- **Graph Convolutional Networks (GCNs)**
  - Introduction to GCNs and their importance in graph data
  - GCN layers and graph embedding techniques
  - Implementing GCNs for node classification

Sunday (2 hours)

- **Advanced GCN Applications**
  - Graph attention networks (GAT)
  - Graph neural networks for link prediction
  - Hands-on implementation of GCNs with TensorFlow or PyTorch

Outcome:
Students will understand the structure and workings of graphs, and be able to apply graph neural networks for classification and prediction tasks on graph-structured data.

Week 15: Deep Learning

Saturday (2 hours)

- **Recurrent Neural Networks (RNNs)**
  - Introduction to RNNs and sequence modeling
  - Exploding and vanishing gradient problem in RNNs
  - Applications of RNNs in time-series and text data

- **Long Short-Term Memory (LSTM) Networks**
  - Structure of LSTM: gates and cell states
  - Implementing LSTMs for sequential data (text, time-series)
  - LSTM for text generation and predictive modeling

Sunday (2 hours)

- **Attention Mechanism and Transformers**
  - Attention mechanism: self-attention and encoder-decoder architectures
  - Introduction to transformer models (e.g., BERT, GPT)
  - Hands-on implementation of transformers for NLP tasks

Outcome:
Students will have a strong understanding of deep learning models for sequential data, including RNNs and LSTMs, and an introduction to transformer models for natural language processing.

Week 16: Recommendation Systems

Saturday (2 hours)

- **Collaborative Filtering**
  - User-based vs item-based collaborative filtering
  - Cosine similarity, Pearson correlation for collaborative filtering
  - Matrix factorization and the role of SVD (Singular Value Decomposition)

- **Content-Based Filtering**
  - Building recommendation systems using user/item attributes
  - Feature extraction from text/images for content-based filtering
  - Implementing basic recommendation systems with Scikit-Learn

Sunday (2 hours)

- **Hybrid Recommendation Systems**
  - Combining collaborative and content-based filtering
  - Introduction to deep learning-based recommendation systems
  - Hands-on building of a hybrid recommendation system

Outcome:
Students will be proficient in designing and implementing recommendation systems, using both collaborative and content-based filtering techniques, and combining them into hybrid systems.

Week 17: ML Ops

Saturday (2 hours)

- **Introduction to MLOps**
  - Overview of MLOps: importance and lifecycle
  - Differences between traditional DevOps and MLOps
  - Data pipeline: data collection, preprocessing, versioning, and storage

- **Version Control for Machine Learning Models**
  - Tools like DVC (Data Version Control) for tracking data/model versions
  - Model reproducibility and deployment challenges
  - Introduction to CI/CD pipelines in MLOps

Sunday (2 hours)

- **Model Deployment**
  - Introduction to model serving with Flask/FastAPI
  - Deploying models to cloud platforms (AWS)
  - Managing APIs and endpoints for deployed models

- **Monitoring and Maintenance**
  - Monitoring deployed models: concept drift, data drift
  - Retraining models and continuous improvement
  - Logging and scaling deployed machine learning applications

Outcome:
Students will understand the end-to-end process of managing machine learning models from version control to deployment, monitoring, and continuous integration. They will be able to apply MLOps principles to real-world AI/ML projects.

this week>

Learning

BREAK & OPTIONAL CASE STUDIES

Week 18: Deep Generative Models (Generative AI)

Saturday (2 hours)

- **Introduction to Generative Models**
  - Overview of generative models: what are they and how they work
  - Key concepts: latent space, prior and posterior distributions
  - Introduction to Variational Autoencoders (VAEs)

- **Variational Autoencoders (VAEs)**
  - Structure of VAEs: encoder, decoder, and latent space
  - KL divergence and reconstruction loss
  - Hands-on implementation of a VAE for image generation

Sunday (2 hours)

- **Generative Adversarial Networks (GANs)**
  - GAN architecture: generator and discriminator networks
  - Training GANs and dealing with instability
  - Hands-on implementation of a simple GAN for image generation

- **Advanced GAN Techniques**
  - Conditional GANs (CGANs) and their applications
  - CycleGAN for image-to-image translation
  - Implementing a CGAN for conditional image generation

Outcome:
Students will gain hands-on experience with deep generative models like VAEs and GANs, understanding how these models work and implementing them to generate realistic data, images, or content.

Week 19: Large Language Models (LLMs)

Saturday (2 hours)

- **Introduction to Large Language Models (LLMs)**
  - Understanding LLMs: BERT, GPT, and their architectures
  - Pre-training and fine-tuning in LLMs
  - Differences between BERT (bidirectional) and GPT (unidirectional)

- **Applications of LLMs**
  - LLMs in NLP tasks: text classification, question answering, summarization
  - Using pre-trained LLMs for transfer learning
  - Hands-on text classification using BERT or GPT models

Sunday (2 hours)

- **Training LLMs on Custom Data**
  - Understanding tokenization (BPE, WordPiece)
  - Fine-tuning LLMs on domain-specific text data
  - Hands-on fine-tuning of GPT/BERT models for specific NLP tasks

- **LLMs in Production**
  - Deploying LLMs for real-world applications (chatbots, virtual assistants)
  - Managing large-scale LLM inference on cloud infrastructure
  - Hands-on deployment of a fine-tuned LLM in a web-based application

Outcome:
Students will learn how LLMs are structured and gain practical skills in fine-tuning these models for custom tasks. They will be able to deploy LLMs for NLP tasks and integrate them into production-level applications.

Week 20: LLMs - Transformer Architecture

Saturday (2 hours)

- **The Transformer Model**
  - Key components: encoder-decoder architecture, attention mechanism
  - Self-attention: calculating attention weights and understanding how it works
  - Positional encoding in transformers

- **Multi-Head Attention**
  - Concept of multi-head attention in transformers
  - Advantages of multi-head attention for parallel learning
  - Hands-on understanding of attention weights through visualization

Sunday (2 hours)

- **Feed-Forward Networks in Transformers**
  - Layer normalization, feed-forward layers, and residual connections
  - Implementing a basic transformer model from scratch
  - Hands-on example of building a custom transformer for sequence tasks

- **Scaling Transformers**
  - Concept of scaling transformer models for larger datasets
  - Transformer variants: BERT, GPT, T5, and their applications
  - Hands-on implementation of a scaled transformer model

Outcome:
Students will develop a deep understanding of the transformer architecture and how it powers LLMs. They will gain practical skills in building and scaling transformers for complex NLP tasks.

Week 21: Prompt Engineering

Saturday (2 hours)

- **Introduction to Prompt Engineering**
  - Understanding the importance of prompts for LLMs
  - Techniques for crafting effective prompts
  - Overview of prompt-based learning for few-shot and zero-shot learning

- **Prompt Construction**
  - Crafting prompts for different NLP tasks (question answering, summarization)
  - Strategies for guiding model responses through prompt variations
  - Hands-on example of using GPT-3 or GPT-4 for few-shot learning tasks

Sunday (2 hours)

- **Advanced Prompt Engineering Techniques**
  - Chain-of-thought prompting for complex reasoning tasks
  - Adversarial prompting and understanding model biases
  - Fine-tuning models through prompt templates for domain-specific applications

- **Ethics in Prompt Engineering**
  - Challenges with biased or unethical prompts in generative models
  - Designing ethical prompts to mitigate biases in LLMs
  - Practical session on ethical prompt engineering techniques

Outcome:
Students will be able to craft and optimize prompts for various NLP tasks, leveraging LLMs in effective ways. They will also understand the ethical considerations surrounding the use of LLMs in real-world applications.

Week 22: Training and Fine-Tuning Large Language Models

Saturday (2 hours)

- **Pre-training LLMs**
  - Overview of unsupervised pre-training: masked language modeling (MLM), causal language modeling (CLM)
  - Pre-training LLMs from scratch: data collection and tokenization
  - Training a BERT/GPT model from scratch on a custom dataset

- **Fine-tuning Strategies**
  - Transfer learning: adapting pre-trained models to specific tasks
  - Hands-on fine-tuning of BERT or GPT models for classification tasks
  - Evaluating fine-tuned models for performance and accuracy

Sunday (2 hours)

- **Hyperparameter Tuning for LLMs**
  - Understanding hyperparameters: learning rate, batch size, sequence length
  - Fine-tuning hyperparameters for model optimization
  - Using tools like Hugging Face's Transformers library for fine-tuning

- **Model Compression and Optimization**
  - Techniques for compressing large models (quantization, pruning, knowledge distillation)
  - Optimizing LLMs for faster inference on edge devices
  - Practical implementation of LLM compression methods

Outcome:
Students will have hands-on experience in training and fine-tuning LLMs, tuning hyperparameters, and optimizing models for deployment in production environments.

Week 23: Semantic Search and Retrieval Augmented Generation (RAG)

Saturday (2 hours)

- **Introduction to Semantic Search**
  - Differences between keyword-based and semantic search
  - Embedding-based search: using vector representations for search tasks
  - Implementing a simple semantic search engine using embeddings

- **Embedding Models**
  - Overview of embedding models (Word2Vec, GloVe, Sentence-BERT)
  - Calculating semantic similarity between queries and documents
  - Hands-on example of semantic search with Sentence-BERT

Sunday (2 hours)

- **Introduction to Retrieval Augmented Generation (RAG)**
  - Combining retrieval mechanisms with generative models for RAG
  - Implementing a basic RAG system using pre-trained transformers
  - Practical application of RAG for question-answering systems

- **Building a Full-Scale RAG System**
  - Integrating semantic search with generative models
  - Deploying RAG models for real-time query answering
  - Hands-on project: building and deploying a RAG-based application

Outcome:
Students will develop practical skills in building semantic search systems and integrating them with generative models using RAG. They will be able to apply these techniques in information retrieval and conversational AI.

Week 24: Building Applications Using LLMs

Saturday (2 hours)

- **Designing LLM Applications**
  - Key considerations in designing applications with LLMs
  - Choosing the right LLMs for the task: GPT, T5, BERT-based models
  - Architecting an LLM-based chatbot or question-answering system

- **Natural Language Generation (NLG) Applications**
  - Using LLMs for text generation: chatbots, content generation
  - Hands-on implementation of a chatbot using an LLM (GPT-4 or similar)
  - Using reinforcement learning to improve chatbot conversations

Sunday (2 hours)

- **Introduction to LangChain**
  - Overview and purpose of LangChain
  - Key features and components
  - Setting Up LangChain - Installation, environment setup and basic configuration

- **Building Applications with Multiple LLMs**
  - Understanding Multi-Model Architectures , benefits and examples
  - Integrating and orchestrating Multiple LLMs with LangChain
  - Hands-on example: Building a chatbot that uses different models for intent recognition and response generation
  - Best Practices for Multi-Model Applications - performance optimization, managing API calls and costs.

Outcome:
Students will develop a comprehensive understanding of building applications using large language models, with a strong emphasis on practical implementation using LangChain.

Week 25: Capstone Project - Orientation

Saturday (2 hours)

- **Introduction to the Capstone Project**
  - Overview of the capstone project objectives
  - Expectations and deliverables for the final project
  - Forming project groups (if applicable) and assigning mentors
  - Discussion on how to approach the problem-solving process

Sunday (2 hours)

- **Project Planning and Research**
  - Guidelines on selecting the project topic
  - Understanding market trends and how to align the project with industry needs
  - Tools and technologies available for the project
  - Initial project brainstorming session with mentor feedback

Note:
The specific capstone projects will be decided when we reach Week 20, based on the latest market requirements and trends in AI and ML, ensuring relevance to real-world applications.

Week 26: Capstone Project - Presentations

Saturday (2 hours)

- **Finalizing Project Reports**
  - Final submission of project reports and code
  - Mentor review and feedback on projects
  - Practice session for presentations

Sunday (2 hours)

- **Capstone Project Presentations**
  - Final project presentations to a panel of industry experts and mentors
  - Evaluation based on problem-solving, technical implementation, and creativity
  - Discussion on improvements and real-world applicability

Outcome:

Students will complete a market-relevant, real-world capstone project, showcasing their AI and ML skills, and will receive valuable feedback from industry professionals.

For more information about the Program or any inquiries, please feel free to reach out to us:

TALK TO US

+91 96866-66663

info@prochaintech.in