Data Science with Generative AI

Comprehensive Course Curriculum

Level: Intermediate to Advanced | **Prerequisites:** Basic programming knowledge

---

Executive Summary

This comprehensive course curriculum integrates foundational data science concepts with cutting-edge generative AI technologies. Students will progress from Python programming fundamentals through advanced machine learning, deep learning, and specialized applications of large language models (LLMs). The course is designed for engineering students and professionals seeking expertise in data-driven AI solutions.

Course Learning Objectives

By completing this course, students will:

- Master Python programming and data manipulation using NumPy, Pandas, and related libraries
- Develop advanced statistical knowledge for hypothesis testing and inferential analysis
- Design and implement supervised and unsupervised machine learning models
- Build deep learning architectures including CNNs and RNNs for real-world applications
- Understand transformer-based models and large language models (LLMs)
- Apply prompt engineering and LangChain for generative AI applications
- Deploy AI/ML solutions on cloud platforms (AWS, Azure)
- Implement retrieval-augmented generation (RAG) systems for intelligent applications

---

Module Structure Overview

- Module 1: Python Programming and Flask Framework
- Module 2: Data Analysis and Visualization
- Module 3: Advanced Statistics and Probability
- Module 4: SQL and Business Intelligence (Power BI)
- Module 5: Machine Learning - Supervised Learning
- Module 6: Deep Learning Fundamentals
- Module 7: CNN and Computer Vision
- Module 8: Natural Language Processing
- Gen AI Modules 1-16: Generative AI Specialization

Detailed Module Breakdown

MODULE 1: Python Programming and Flask Framework

Prerequisites: None

Learning Outcomes

Upon completion, students will:

- Write efficient Python code using Jupyter Notebook environment
- Understand and implement control flow, loops, and conditional statements
- Create and manipulate data structures (lists, dictionaries, tuples, sets)
- Implement object-oriented programming concepts
- Handle file I/O and exceptions effectively
- Import and utilize Python standard library modules

Topics

Python Fundamentals

- Introduction to Python and its applications in data science
- Why Python is essential for data science workflows
- Anaconda installation and environment setup
- Jupyter Notebook IDE: capabilities and best practices
- Python syntax and basic commands
- Identifiers, operators, and variable types
- Strings, data types, and type conversion

Data Structures

- Lists: creation, indexing, slicing, and operations
- Tuples: immutability and use cases
- Dictionaries: key-value pairs and manipulation
- Sets: unique elements and set operations
- List and dictionary comprehensions

Control Flow and Functions

- Conditional operators and logical operations
- if, elif, else statements
- While loops and for loops
- Nested loops and loop control (break, continue, pass)
- Function definition and types
- Lambda functions and map, filter, reduce operations
- Scope and variable lifetime
- Code optimization and argument passing

Object-Oriented Programming

- Classes and objects
- The **init**() constructor
- Instance methods and self reference
- Modifying object properties

- Object deletion
- Encapsulation principles

File Handling and Modules

- Creating, reading, and writing files
- File operations and context managers
- Errors and exception handling (try, except, finally)
- Importing modules and using help()
- Important standard library modules: math, random, datetime, os
- Module aliasing and best practices

Practical Projects

- Python basics: Calculator application with user input validation
- File manipulation: Text file processor with error handling
- OOP project: Simple library management system

MODULE 2: Data Analysis and Visualization in Python

Prerequisites: Module 1

Learning Outcomes

Upon completion, students will:

- Manipulate and analyze data using NumPy and Pandas
- Create professional visualizations with Matplotlib and Seaborn
- Conduct exploratory data analysis (EDA) effectively
- Handle missing data and prepare data for analysis
- Perform data merging, joining, and aggregation operations

Topics

NumPy - Numerical Python

- NumPy array fundamentals
- Array creation and initialization methods
- Array indexing and slicing techniques
- Broadcasting and element-wise operations
- Mathematical functions and linear algebra operations
- Basic statistics with NumPy

Pandas - Data Manipulation

- Series and DataFrame structures
- Data importing from Excel, CSV, and other formats
- Data exporting techniques
- Indexing, slicing, and conditional filtering
- GroupBy operations and aggregations
- Pivot tables and crosstabs
- Concatenating, merging, and joining datasets
- Removing duplicates and handling missing data
- String manipulation on DataFrames

- Descriptive statistics

Data Visualization - Matplotlib and Seaborn

- Introduction to Matplotlib and plot structure
- Figure and axes management
- Line plots, scatter plots, and bar charts
- Histograms, box plots, and violin plots
- Pie charts and advanced plotting
- Subplot creation and layout management
- Formatting plots for publication quality
- Seaborn for statistical visualization
- Pair plots, count plots, heatmaps, and categorical plots

Exploratory Data Analysis (EDA)

- EDA methodology and best practices
- Univariate analysis techniques
- Bivariate analysis and relationships
- Feature distributions and outlier detection
- Data quality assessment
- Visual storytelling with data

Unstructured Data Processing

- Regular expressions: patterns and metacharacters
- Pattern matching in Pandas
- Text preprocessing and cleaning
- Working with text data

Practical Projects

- **Web Scraping and EDA Project**
  - Collect raw data from web sources
  - Convert unstructured data to structured format
  - Perform complete EDA pipeline
  - Create visualizations for insights
  - Generate summary reports

MODULE 3: Advanced Statistics and Probability

Prerequisites: Module 1, 2

Learning Outcomes

Upon completion, students will:

- Apply statistical techniques for data analysis
- Understand sampling methodologies and distributions
- Conduct hypothesis testing and confidence interval estimation
- Identify and interpret statistical relationships in data

Topics

Descriptive Statistics

- Population vs. sample concepts
- Parameters vs. statistics
- Variable types and classification
- Measures of central tendency (mean, median, mode)
- Measures of dispersion (variance, std dev, range, IQR)
- Skewness and kurtosis analysis
- Box plots and outlier detection methods
- Covariance and correlation analysis

Data Gathering and Sampling

- Data collection techniques
- Sampling methodologies:
  - Simple random sampling
  - Stratified sampling
  - Cluster sampling
  - Systematic sampling
  - Convenience sampling
- Bias in sampling and mitigation strategies

Probability Distributions

- Probability fundamentals and axioms
- Discrete probability distributions:
  - Bernoulli distribution
  - Binomial distribution
  - Poisson distribution
- Continuous probability distributions:
  - Normal distribution
  - Standard normal distribution
  - T-distribution
  - Chi-square distribution
  - F-distribution
- Distribution fitting and evaluation

Inferential Statistics

- Central limit theorem and its applications
- Confidence intervals: construction and interpretation
- Margin of error and sample size determination
- Hypothesis testing framework
- Type I and Type II errors
- P-values and significance levels
- Statistical tests:
  - Z-test and t-test
  - Chi-square goodness-of-fit test
  - F-test and ANOVA
  - Post-hoc tests (Tukey, LSD)
- Assumptions and their verification

Practical Exercises

- Distribution analysis and fitting
- Hypothesis testing on real datasets
- Confidence interval construction and interpretation
- Statistical power analysis

MODULE 4: SQL and Business Intelligence (Power BI)

Prerequisites: Module 1, 2

Learning Outcomes

Upon completion, students will:

- Write complex SQL queries for data extraction and transformation
- Design relational databases effectively
- Create interactive dashboards and reports in Power BI
- Perform data modeling and DAX calculations
- Implement row-level security (RLS) in Power BI

Topics

SQL for Data Science

- Database fundamentals and relational model
- SQL basics: DML, DDL, DCL
- SELECT, FROM, WHERE clauses
- Logical operators (AND, OR, NOT)
- Filtering and sorting:
  - IN, BETWEEN, LIKE operators
  - GROUP BY and ORDER BY
  - HAVING clauses
- SQL JOINs:
  - INNER JOIN
  - LEFT/RIGHT/FULL OUTER JOINs
  - Self joins and cross joins
- Aggregation functions (COUNT, SUM, MIN, MAX, AVG)
- String and date functions
- NULL value handling
- Subqueries and derived tables
- Temporary tables and CTEs (Common Table Expressions)
- Window functions:
  - ROW_NUMBER(), RANK(), DENSE_RANK()
  - LAG() and LEAD() for sequence analysis
  - Running totals and averages

Power BI Fundamentals

- Business Intelligence concepts
- Power BI ecosystem and components
- Power BI Desktop overview and workflow
- Installation and configuration

- Comparison with other BI tools
- Power BI service and cloud deployment

Data Import and Visualization

- Import data from multiple sources
- Web data import and transformation
- Visualization types and selection:
  - Categorical visualizations
  - Trend and time-series visualizations
  - Relationship visualizations
  - Distribution visualizations
- KPI visuals and scorecards
- Tables and matrix visuals
- Slicers and filtering mechanisms
- Visual formatting and styling
- Dashboard design principles

Power Query and Data Transformation

- Power Query editor interface
- Data transformation benefits and best practices
- Column transformations (split, merge, extract)
- Row transformations (filter, sort, remove duplicates)
- Combine and merge queries
- Data type handling and conversion
- Creating custom columns
- M Language basics

Power Pivot and Data Modeling

- Introduction to data modeling
- Creating relationships between tables
- Relationship cardinality (1:1, 1:N, M:N)
- Relationship view and diagram
- Star schema design
- Calculated columns vs. measures

DAX (Data Analysis Expressions)

- DAX syntax and fundamentals
- Logical functions (IF, AND, OR)
- Text functions (CONCATENATE, UPPER, LOWER)
- Mathematical and statistical functions
- Aggregation functions (SUM, AVERAGE, COUNT)
- Filter functions (FILTER, ALL, RELATED)
- Time-intelligent functions (YEAR-TO-DATE, MONTH-TO-DATE)
- Creating date dimension tables
- Row and context relationships
- Implicit and explicit measures

Power BI Services and Deployment

- Power BI service overview
- Publishing reports and dashboards
- Sharing and collaboration
- Row-level security (RLS) implementation
- Refresh strategies
- Mobile app deployment
- Embedded analytics

Advanced Features

- Visual interactions and drill-through
- Drill-down functionality
- Conditional formatting
- Creating dynamic buttons
- Custom visuals
- Python script visuals

Practical Projects

- **Data Warehouse Project**
    - Design normalized database schema
    - Write complex SQL queries
    - Build comprehensive Power BI dashboard
    - Implement DAX measures and calculations
    - Deploy RLS for different user groups

MODULE 5: Machine Learning - Supervised Learning

Prerequisites: Module 1, 2, 3

Learning Outcomes

Upon completion, students will:

- Implement and evaluate regression and classification algorithms
- Understand model selection and hyperparameter tuning
- Apply ensemble methods effectively
- Build production-ready ML models

Topics

Supervised Learning Fundamentals

- Machine learning taxonomy
- Supervised vs. unsupervised learning
- Regression vs. classification problems
- Training, validation, and test sets
- Cross-validation techniques
- Model evaluation metrics and selection
- Bias-variance tradeoff

Linear Algebra for ML

- Matrices and vectors

- Matrix operations and properties
- Vector spaces and subspaces
- Eigenvalues and eigenvectors
- Decompositions (SVD, QR)

Regression Techniques

- Simple linear regression:
  - Coefficient estimation (OLS)
  - R-squared and adjusted R-squared
  - MSE and RMSE metrics
- Multiple linear regression
- OLS assumptions and diagnostics
- Multicollinearity detection and handling
- Feature selection methods
- Gradient descent optimization
- Polynomial regression
- Regularization techniques:
  - Ridge regression (L2)
  - Lasso regression (L1)
  - ElasticNet
- Regression model evaluation:
  - Homoscedasticity and heteroscedasticity
  - Residual analysis and Q-Q plots
  - Cook's distance for influential points
  - Shapiro-Wilk test for normality
  - Identifying line of best fit
- Qualitative predictors and interaction terms
- Non-linear transformations

Classification Techniques

- Classification problem overview
- Logistic regression:
  - Logistic model and interpretation
  - Sigmoid and logit functions
  - Decision boundaries
  - Multi-class classification
- Classification evaluation metrics:
  - Confusion matrix
  - Accuracy and error rate
  - Sensitivity (TPR) and specificity (TNR)
  - Precision and recall
  - F1-score and F-beta scores
  - ROC curves and AUC
  - Kappa score for inter-rater agreement
- Naive Bayes classifier:
  - Bayes theorem and conditional probability
  - Prior and posterior probabilities
  - Likelihood estimation
  - Multinomial Naive Bayes
  - Bernoulli and Gaussian variants

Tree-Based Models

- Decision trees:
  - Tree terminology (root, nodes, leaves)
  - Regression trees and classification trees
  - Gini index and information gain
  - Overfitting and pruning strategies
  - Stopping criteria
  - Accuracy estimation
- Ensemble methods:
  - Bootstrap aggregation (bagging)
  - Random forests:
    - Algorithm and theory
    - Variable importance
    - Hyperparameter tuning
    - Pros and cons
  - Gradient boosting machines
- Resampling methods:
  - k-fold cross-validation
  - Leave-one-out cross-validation
  - Bias-variance analysis

Distance-Based Models

- K-Nearest Neighbors (KNN):
  - Algorithm mechanics
  - Eager vs. lazy learners
  - Choosing k value
  - Curse of dimensionality
  - Distance metrics
  - Performance improvements
- Support Vector Machines (SVM):
  - Maximal margin classifier
  - Hyperplane concept
  - Support vectors
  - Hard and soft margin classification
  - Kernel trick and kernel methods
  - Polynomial and radial basis kernels
  - Multi-class SVM
  - Hyperparameter tuning (gamma, C, epsilon)

Practical Projects

- **Regression Case Study**
  - Data understanding and EDA
  - Feature engineering
  - Model selection and validation
  - Hyperparameter optimization
  - Final model evaluation and interpretation
- **Classification Project**
  - Multi-class classification problem
  - Model comparison and ensemble methods

- ROC and AUC analysis
- Threshold optimization
- Business metric evaluation

MODULE 6: Deep Learning Fundamentals

Prerequisites: Module 1, 2, 3, 5

Learning Outcomes

Upon completion, students will:

- Understand neural network architecture and training
- Implement and optimize deep learning models
- Apply deep learning to regression and classification
- Utilize TensorFlow and Keras effectively

Topics

Neural Network Basics

- Perceptron and historical context
- Activation functions:
  - Sigmoid function
  - ReLU (Rectified Linear Unit)
  - Leaky ReLU
  - Tanh
  - Softmax
- Forward propagation
- Backpropagation and chain rule
- Gradient descent and learning rates
- Weight initialization strategies
- Fully connected (dense) layers
- Cross-entropy loss
- Regularization techniques:
  - L1 and L2 regularization
  - Dropout
  - Batch normalization

Optimization and Training

- Optimization algorithms:
  - Stochastic gradient descent (SGD)
  - Momentum
  - Adam optimizer
  - RMSprop
- Hyperparameter tuning
- Batch processing and mini-batches
- Epochs and convergence
- Early stopping

TensorFlow 2.0 and Keras

- TensorFlow installation and setup
- Keras API and high-level interface
- Tensors and computational graphs
- TensorBoard visualization
- Google Colab for cloud-based training
- Building models with Sequential API
- Building models with Functional API
- Custom layers and functions

Artificial Neural Networks (ANN)

- ANN for regression tasks
- ANN for classification tasks
- Model evaluation and interpretation
- Improving and tuning ANN performance
- Saving and restoring models
- Model checkpointing

Practical Projects

- **Neural Network for Regression**
  - Build multi-layer perceptron
  - Hyperparameter optimization
  - Performance comparison with classical ML
- **Classification with Deep Learning**
  - Binary and multi-class classification
  - Model regularization and dropout
  - Training history and learning curves

MODULE 7: CNN and Computer Vision

Prerequisites: Module 6

Learning Outcomes

Upon completion, students will:

- Understand convolutional neural networks and image processing
- Implement CNN architectures from scratch
- Apply transfer learning effectively
- Build computer vision applications (object detection, segmentation)

Topics

Working with Images and CNN Fundamentals

- Image representation and properties
- Pixel digitization, sampling, and quantization
- Image filtering and preprocessing
- 2D convolutions for images
- Backward convolutions
- Transposed convolutions
- Pooling operations:

    - Max pooling
    - Average pooling
    - Other pooling strategies
- Fully connected layers as convolutions

CNN Architectures

- LeNet: foundational CNN design
- AlexNet: deep learning breakthrough
- ZFNet and VGGNet: increasing depth
- GoogleNet and Inception modules
- ResNet: residual connections and skip connections
- MobileNet for lightweight deployment
- GPU vs. CPU computing considerations

Transfer Learning

- Transfer learning principles
- Fine-tuning pre-trained models
- Feature extraction approaches
- Domain adaptation techniques
- Visualization of learned features
- T-SNE for embedding visualization
- Occlusion experiments and interpretability

Object Detection

- Region-based CNN approaches
- YOLO (You Only Look Once) architecture
- SSD (Single Shot Detector)
- Bounding box regression
- Non-maximum suppression
- Evaluation metrics (mAP, IoU)

Semantic Segmentation

- Semantic segmentation fundamentals
- Segmentation process and pipeline
- U-Net architecture
- FCN (Fully Convolutional Networks)
- Dilated convolutions
- Other segmentation variants

Advanced Topics

- Siamese networks for metric learning
- One-shot and few-shot learning
- Attention mechanisms in vision
- Vision transformers

Practical Projects

- **Transfer Learning with SVHN Dataset**
  - Use pre-trained models from ImageNet
  - Fine-tune for target task
  - Achieve high accuracy with limited data
- **Object Detection Project**
  - Implement YOLO or SSD
  - Evaluate on custom dataset
  - Real-time inference optimization
- **Semantic Segmentation**
  - Build U-Net for image segmentation
  - Pixel-wise classification
  - Performance metrics and visualization

MODULE 8: Natural Language Processing (NLP)

Prerequisites: Module 1, 2, 3, 6

Learning Outcomes

Upon completion, students will:

- Preprocess and analyze text data
- Implement NLP techniques for various tasks
- Build sequence models for language understanding
- Apply transformers and advanced NLP architectures

Topics

NLP Fundamentals and Statistical Techniques

- Introduction to NLP and applications
- Text preprocessing:
  - Tokenization
  - Stop word removal
  - Normalization (case folding)
  - Stemming and lemmatization
- Feature extraction:
  - Bag of words (BoW)
  - TF-IDF (Term Frequency-Inverse Document Frequency)
  - Sparse and dense representations
- Language models:
  - Probabilistic models
  - N-gram models
  - Channel models
- NLTK toolkit and practical applications

Word Embeddings and Semantic Representations

- Word2Vec:
  - Skip-gram model
  - CBOW (Continuous Bag of Words)

- Training and applications
- GloVe (Global Vectors)
- FastText embeddings
- Embedding visualization and interpretation
- Part-of-Speech (POS) tagging with NLTK
- Named Entity Recognition (NER)
- Semantic similarity and word analogies

Sequential Models

- Introduction to sequential data
- Recurrent Neural Networks (RNN):
  - RNN architecture and theory
  - Vanishing gradient problem
- Long Short-Term Memory (LSTM):
  - LSTM architecture and gates
  - Forward pass computations
  - Backpropagation through time (BPTT)
  - Practical LSTM implementations with Keras
- Gated Recurrent Units (GRU)
- Bidirectional RNNs and LSTMs
- Attention mechanisms in RNNs

NLP Applications

- Sentiment analysis:
  - Polarity classification
  - Opinion mining
  - Emotion detection
- Text generation and language modeling
- Machine translation:
  - Sequence-to-sequence models
  - Encoder-decoder architecture
  - Attention mechanisms
- Chatbot development:
  - Conversational AI
  - Intent recognition
  - Entity extraction
  - Response generation
- Advanced LSTM structures

Practical Projects

- **Sentiment Analysis Project**
  - Text preprocessing pipeline
  - Feature extraction (TF-IDF, embeddings)
  - Model comparison (Naive Bayes, LSTM)
  - Evaluation and interpretation
- **Text Generation with LSTM**
  - Character-level or word-level models
  - Text generation from seed sequences
  - Temperature and diversity control

- **Machine Translation**
  - Encoder-decoder architecture
  - Attention mechanism implementation
  - Evaluation with BLEU scores
- **Chatbot Development**
  - Intent classification
  - Entity extraction
  - Response generation
  - Conversation flow management

GENERATIVE AI SPECIALIZATION (Modules 1-16)

GEN AI MODULE 1: Generative AI Principles and Applications

Topics

- Generative AI fundamentals and taxonomy
- Types of generative models:
  - Autoencoders
    - GANs (Generative Adversarial Networks)
    - Diffusion models
    - Flow-based models
  - Applications across industries:
    - Image generation and editing
    - Text generation and summarization
    - Code generation and assistance
    - Audio and music generation
    - Drug discovery and molecular design
    - Content creation and marketing
  - Machine learning algorithms in generative AI context
  - Advantages of generative AI:
    - Automation and efficiency
    - Creativity and innovation
    - Personalization at scale
    - Cost reduction
  - Disadvantages and challenges:
    - Training data requirements
    - Computational costs
    - Mode collapse and failure modes
    - Interpretability challenges
  - Ethical considerations:
    - Bias in training data
    - Copyright and intellectual property
    - Deepfakes and misinformation
    - Environmental impact
    - Responsible AI principles

  ### GEN AI MODULE 2: NLP and Deep Learning for Gen AI

  #### Topics

- NLP essentials for generative models
- Text classification for content understanding
- Text preprocessing at scale
- Basic NLP tasks:
  - Named entity recognition
  - Dependency parsing
  - Coreference resolution
  - Semantic role labeling
- Deep learning architectures for NLP
- Neural network training strategies
- Backpropagation and gradient flow
- RNN, LSTM, GRU foundations
- Deep learning applications in NLP:
  - Machine translation
  - Question answering
  - Summarization
  - Dialogue systems

GEN AI MODULE 3: Autoencoders and GANs

Topics

Autoencoders
- **Autoencoder architecture and training**
- **Encoder and decoder components**
- **Bottleneck layers for dimensionality reduction**
- **Variational Autoencoders (VAEs):**
  - **Probabilistic framework**
  - **Latent space interpretation**
  - **KL divergence regularization**
  - **Reparameterization trick**
- **Applications:**
  - **Data compression**
  - **Anomaly detection**
  - **Image generation**
  - **Denoising autoencoders**

Generative Adversarial Networks (GANs)
- **GAN architecture:**
  - **Generator network**
  - **Discriminator network**
  - **Adversarial training process**
- **GAN training dynamics:**
  - **Min-max game theory**
  - **Loss functions and optimization**
  - **Mode collapse and solutions**
  - **Training instability and fixes**
- **GAN variants:**
  - **DCGAN (Deep Convolutional GAN)**
  - **StyleGAN**
  - **CycleGAN for unpaired image translation**

* Conditional GANs (cGANs)
* Wasserstein GANs (WGAN)

– Applications:
* Image generation
* Style transfer
* Image-to-image translation
* Data augmentation
* Inpainting and super-resolution

Practical Projects

- Variational Autoencoder for Image Generation
- GAN for Artistic Style Transfer
- CycleGAN for Unpaired Image Translation

GEN AI MODULE 4: Transformer Architecture and Language Models

Topics

Transformer Architecture

- Transformer fundamentals
- Self-attention mechanism:
  * Query, key, value computations
  * Scaled dot-product attention
  * Attention weights and interpretability
- Multi-head attention:
  * Parallel attention heads
  * Information fusion
  * Complexity analysis
- Positional encoding:
  * Sinusoidal encodings
  * Relative position representations
  * Rotary position embeddings
- Feed-forward networks
- Layer normalization and residual connections
- Encoder-decoder architecture
- Encoder-only and decoder-only variants
- Efficiency improvements:
  * Sparse attention
  * Linear attention approximations
  * Efficient transformers (Linformer, Performer)

Language Models

- Language model taxonomy
- Autoregressive models (GPT family)
- Autoencoding models (BERT family)
- Sequence-to-sequence models
- Pre-training objectives:
  * Causal language modeling
  * Masked language modeling
  * Next sentence prediction

    * Text infilling
- Transfer learning in LLMs
- Fine-tuning strategies
- Prompt engineering for pre-trained models

Advanced Transformer Models
- GPT Series:
    * GPT-2: Large-scale pre-training
    * GPT-3: Few-shot learning capabilities
    * GPT-4: Multimodal understanding
    * Instruction-tuned variants (InstructGPT, GPT-3.5)
- BERT and Variants:
    * BERT: Bidirectional encoder
    * RoBERTa: Optimized pre-training
    * ALBERT: Parameter efficiency
    * DistilBERT: Knowledge distillation
- Multimodal Transformers:
    * Vision Transformers (ViT)
    * CLIP: Image-text alignment
    * Multimodal fusion architectures
- Large Language Models (LLMs):
    * Llama
    * Claude
    * PaLM
    * Falcon
    * Open source alternatives

Applications of Transformer Models
- Text generation and completion
- Text summarization:
    * Abstractive summarization
    * Extractive summarization
    * Multi-document summarization
- Question answering systems:
    * Closed-book QA
    * Open-domain QA
    * Reading comprehension
- Machine translation
- Semantic similarity and retrieval
- Named entity recognition
- Text classification
- Sentiment analysis
- Dialogue and conversational AI

GEN AI MODULE 5: Prompt Engineering

Topics
- Prompt engineering fundamentals
- Definition and importance of effective prompts
- Applications and real-world impact
- Prompt design principles:
    * Clarity and specificity

    * Context provision
    * Role assignment
    * Constraint specification
- Types of prompting techniques:
    * Zero-shot prompting
    * Few-shot prompting
    * Chain-of-thought prompting
    * Self-consistency
    * Tree-of-thoughts
    * In-context learning
- Crafting effective prompts:
    * Task definition
    * Example provision
    * Output format specification
    * Instruction chaining
    * Iterative refinement
- Parameter tuning for LLMs:
    * Temperature (randomness)
    * Top-k sampling
    * Top-p (nucleus) sampling
    * Maximum length settings
    * Frequency and presence penalties
- Adversarial prompting and jailbreaking
- Prompt optimization techniques
- Prompt injection vulnerabilities
- Best practices and guidelines

GEN AI MODULE 6: Generative AI with Large Language Models

Topics
- LLM project lifecycle
- Data collection and preparation:
    * High-quality data sourcing
    * Data cleaning and preprocessing
    * Deduplication strategies
- Pre-training at scale:
    * Model architecture selection
    * Distributed training
    * Compute optimization
    * Checkpoint management
- LLM scaling laws
- Fine-tuning LLMs:
    * Instruction fine-tuning
    * Supervised fine-tuning (SFT)
    * Domain-specific adaptation
- Parameter-efficient fine-tuning:
    * LoRA (Low-Rank Adaptation)
    * QLoRA for efficient tuning
    * Adapters and prefix tuning
    * Prompt tuning

- Reinforcement learning from human feedback (RLHF):
    * Reward model training
    * Human annotation process
    * Policy optimization
    * DPO (Direct Preference Optimization)
- Model evaluation:
    * Automatic metrics (BLEU, ROUGE, METEOR)
    * Human evaluation
    * Benchmark datasets
- Safety and alignment considerations
- Deployment strategies

GEN AI MODULE 7: LLMs for Search, Prediction, and Generation

Topics
- Search query completion:
    * Query understanding
    * Intent prediction
    * Completion ranking
    * Diversity in suggestions
- Next word and token prediction:
    * Language modeling for prediction
    * Beam search and decoding strategies
    * Vocabulary and subword tokenization
- Word embeddings and semantic spaces:
    * Word2Vec and GloVe recap
    * Contextual embeddings (ELMo, BERT)
    * Sentence and document embeddings
    * Embedding applications
- Transformer encoder representations
- Text generation techniques:
    * Greedy decoding
    * Beam search
    * Sampling strategies
    * Nucleus and top-k sampling
    * Temperature-based generation
- Attention layer stacking for generation
- Constrained decoding
- Controllable generation:
    * Style transfer
    * Tone control
    * Length constraints
    * Attribute control

GEN AI MODULE 8: LangChain for LLM Application Development

Topics
- LangChain ecosystem overview
- LangChain foundations:

    - Framework philosophy
    - Core abstractions
    - Module organization
- Benefits of using LangChain:
    - Simplified LLM interaction
    - Agent and tool integration
    - Memory management
    - Chain composition
    - Standardized interfaces
- Components of LangChain:
    - Models and LLM providers
    - Prompts and prompt templates
    - Output parsers
    - Document loaders
    - Text splitters
    - Embeddings
    - Vector stores
    - Retrievers
    - Tools and agents
    - Chains and composition
- Building LLM applications:
    - Basic LLM chains
    - Prompt engineering with templates
    - Memory management (conversation history)
    - Session management
- Off-the-shelf chains:
    - LLMChain
    - ConversationChain
    - RetrievalQA
    - SqlDatabaseChain
- Building custom chains and logic
- Agents and tools:
    - Agent architecture
    - ReAct (Reasoning and Acting)
    - Tool definition and usage
    - Agent loop
- Building and deploying LLM-powered applications:
    - Application architecture
    - API design
    - Deployment strategies
    - Monitoring and logging
    - Error handling
- LangChain for document processing
- Integration with external APIs
- Performance optimization

GEN AI MODULE 9: Retrieval-Augmented Generation (RAG)

Topics
- RAG fundamentals and motivation:

    * **Limitations of LLMs (knowledge cutoff, hallucination)**
    * **Knowledge augmentation approaches**
    * **Retrieval-augmented generation concept**
  - **Document loading and preparation:**
    * **Document loaders for multiple formats (PDF, HTML, etc.)**
    * **Metadata extraction**
    * **Document preprocessing**
    * **Text cleaning and normalization**
  - **Document chunking and splitting:**
    * **Fixed-size chunking**
    * **Semantic chunking**
    * **Overlap strategies**
    * **Hierarchical chunking**
  - **Vector embeddings:**
    * **Embedding models and providers**
    * **Dense embeddings (BERT-style)**
    * **Sparse embeddings (BM25)**
    * **Hybrid embeddings**
    * **Embedding optimization**
  - **Vector stores and databases:**
    * **Vector database options (Pinecone, Weaviate, Milvus)**
    * **In-memory vector stores**
    * **Similarity search algorithms**
    * **Indexing strategies**
    * **Metadata filtering**
  - **Retrieval strategies:**
    * **Semantic similarity search**
    * **BM25 and keyword matching**
    * **Hybrid retrieval**
    * **Multi-query retrieval**
    * **Reranking approaches**
    * **Query expansion**
  - **Ranking and relevance:**
    * **Relevance scoring**
    * **Diversity in results**
    * **Diversity-aware ranking**
  - **Question answering with RAG:**
    * **QA pipeline architecture**
    * **Context retrieval and ranking**
    * **Answer generation**
    * **Chain of thought in QA**
  - **Chatbots with RAG:**
    * **Conversational context**
    * **Memory in chatbots**
    * **Multi-turn interactions**
    * **Context window management**
  - **Building RAG systems with LangChain:**
    * **Loading documents**
    * **Creating embeddings**
    * **Setting up vector stores**
    * **Building retrieval chains**

    * QA chain creation
  - Evaluation of RAG systems:
    * Retrieval evaluation metrics
    * Generation evaluation
    * End-to-end evaluation
    * Human evaluation
  - Advanced RAG techniques:
    * Parent-child relationships in document chunks
    * HyDE (Hypothetical Document Embeddings)
    * Query transformation
    * Multi-index retrieval
  - Production considerations for RAG

GEN AI MODULE 10: Generative AI on Cloud Platforms

Topics
  - Cloud computing fundamentals:
    * IaaS, PaaS, SaaS models
    * Cloud vs. on-premise tradeoffs
    * Cost optimization
  - AWS for Generative AI:
    * AWS S3 for data storage
    * Amazon SageMaker for model training
    * Trn1 instances for efficient training
    * Inf2 instances for inference optimization
    * Amazon Bedrock for LLM access
    * Amazon CodeWhisperer for code generation
  - Azure for Generative AI:
    * Azure OpenAI Service
    * Azure Cognitive Services
    * Azure Machine Learning
    * Azure AI Foundry
    * Deployment options
  - Google Cloud for Generative AI:
    * Vertex AI platform
    * PaLM API access
    * Generative AI Studio
  - Model deployment:
    * Containerization (Docker)
    * Kubernetes orchestration
    * Serverless deployment
    * API gateway setup
  - Monitoring and management:
    * Model monitoring
    * Performance metrics
    * Cost tracking
    * Scaling strategies
  - Security and compliance
  - Multi-region deployment

GEN AI MODULE 11: Working with ChatGPT

Topics

- **ChatGPT introduction and capabilities**
- **ChatGPT API:**
  - **API access and authentication**
  - **Request and response formats**
  - **Models and versioning**
  - **Rate limiting and usage**
- **Leveraging ChatGPT for productivity:**
  - **Writing assistance**
  - **Email composition**
  - **Content brainstorming**
  - **Summarization**
- **ChatGPT for Excel and data analysis:**
  - **Formula generation**
  - **Data cleaning suggestions**
  - **Analysis and insights**
  - **Visualization recommendations**
- **ChatGPT for data science:**
  - **Algorithm selection**
  - **Code generation**
  - **Debugging and optimization**
  - **Model evaluation guidance**
- **ChatGPT with Power BI:**
  - **DAX formula generation**
  - **Visualization selection**
  - **Dashboard design advice**
  - **Performance optimization**
- **Content marketing applications:**
  - **Blog post generation**
  - **Email campaigns**
  - **Social media content**
  - **Keyword research**
- **Social media marketing:**
  - **Caption writing**
  - **Hashtag generation**
  - **Engagement strategies**
- **SEO and keyword optimization:**
  - **Keyword research**
  - **Content optimization**
  - **Meta description writing**
  - **Schema markup generation**
- **Customer service applications:**
  - **Chatbot development**
  - **Response generation**
  - **FAQ creation**
  - **Ticket handling**
- **ChatGPT for developers:**
  - **Code generation**

        - Debugging and fixing bugs
        - Feature implementation suggestions
        - API integration assistance
        - Testing strategies
        - Code documentation

GEN AI MODULE 12: Python with Generative AI

Topics
- Code generation with ChatGPT and alternatives
- AI-powered coding tools:
    - GitHub Copilot
    - Amazon CodeWhisperer
    - TabNine
    - Codex
- Advanced code optimization:
    - Performance profiling
    - Algorithmic improvements
    - Memory optimization
    - Parallelization strategies
- Coding with ChatGPT:
    - Function generation
    - Algorithm implementation
    - Library usage assistance
    - Code refactoring
- Building Python applications with ChatGPT assistance:
    - Architecture design
    - Module development
    - Integration patterns
    - Configuration management
- Debugging with AI assistance:
    - Error diagnosis
    - Stack trace analysis
    - Solution recommendations
- Testing strategies and generation
- Documentation generation
- Code quality analysis and improvement

GEN AI MODULE 13: Evaluating LLM Performance

Topics
- LLM performance evaluation fundamentals
- Comparative metrics:
    - Perplexity:
        - Definition and interpretation
        - Calculation and examples
        - Cross-entropy relationship
    - BLEU (Bilingual Evaluation Understudy):
        - Precision-based metric

            - N-gram matching
            - Brevity penalty
            - Applications and limitations
        * ROUGE (Recall-Oriented Understudy for Gisting Evaluation):
            - Recall-based metrics
            - ROUGE-N, ROUGE-L variants
            - F-measure computation
            - Summarization evaluation
        * METEOR (Metric for Evaluation of Translation with Explicit Ordering):
            - Harmonic mean of precision and recall
            - Alignment-based evaluation
            - Paraphrase and stemming support
    - Human evaluation:
        * Annotation guidelines
        * Inter-annotator agreement
        * Likert scale ratings
        * Comparative evaluation
        * Error analysis
    - Task-specific metrics:
        * Machine translation: TER, ChrF
        * Question answering: F1, EM
        * Summarization: ROUGE variants
        * Dialogue: METEOR, human ratings
    - Benchmark datasets:
        * SuperGLUE for NLU
        * GLUE for language understanding
        * SQuAD for QA
        * BLEU corpus for translation
        * CNN/DailyMail for summarization
    - Choosing appropriate metrics
    - Result interpretation and analysis:
        * Statistical significance
        * Error patterns
        * Model comparison
        * Ablation studies
    - Bias detection in LLM outputs
    - Hallucination and factuality evaluation

GEN AI MODULE 14: Industry Case Studies and Capstone Project

Case Studies

Case Study 1: Generative AI for Personalized Learning
- Adaptive learning systems
- Student performance prediction
- Personalized content generation
- Assessment automation

Case Study 2: Generative AI for Creative Content Generation

- Image generation with Midjourney/DALL-E
- Video content creation
- Music composition
- Story and narrative generation

Case Study 3: Generative AI for Business Applications
- Customer service automation
- Sales and marketing automation
- Business document generation
- Market analysis and insights

Capstone Project: AI-Powered Text and Image Generator

Project Description:

Build a comprehensive application that:

- Accepts text prompts from users
- Generates AI-created text content (blogs, social media, product descriptions)
- Generates corresponding images using DALL-E or Midjourney API
- Combines both modalities for marketing materials
- Implements LangChain for intelligent prompt handling
- Deploys on cloud platform (AWS or Azure)
- Includes API for integration with external systems

Project Deliverables:

- Application architecture and design
- Backend implementation (Python with FastAPI)
- Frontend interface
- API documentation
- Deployment guide
- User and developer guides
- Performance evaluation report

GEN AI MODULE 15: Bonus - Machine Learning with Generative AI

Topics
- AI and machine learning essentials:
  * AI vs. machine learning vs. deep learning
  * Broad AI applications
  * Current limitations
- AI disciplines overview:
  * Machine learning
  * Deep learning
  * Natural language processing
  * Computer vision
  * Robotics
  * Knowledge representation

- Types of AI:
  * Narrow AI (specialized)
  * General AI (multi-domain)
  * Super AI (hypothetical)
- Machine learning fundamentals:
  * Supervised learning review
  * Unsupervised learning review
  * Reinforcement learning introduction
- Predictive ML models:
  * Classification and regression
  * Time series forecasting
  * Anomaly detection
- ML algorithms deep dive:
  * Algorithm families
  * Complexity analysis
  * Practical considerations
- Supervised learning models:
  * Linear models
  * Tree-based models
  * Distance-based models
  * Kernel methods
- Unsupervised learning models:
  * Clustering techniques
  * Dimensionality reduction
  * Anomaly detection
- Semi-supervised learning
- Reinforcement learning introduction:
  * Markov decision processes
  * Q-learning
  * Policy gradient methods

GEN AI MODULE 16: Bonus - Generative AI Tools and Platforms

Topics
- Hugging Face Transformers library:
  * Model hub and model cards
  * PyTorch vs. TensorFlow implementations
  * Fine-tuning with Hugging Face
  * Tokenizers and preprocessing
  * AutoClass for easy model loading
- OpenAI API and GPT-3:
  * API authentication and setup
  * Available models and capabilities
  * Completion and embedding endpoints
  * Cost optimization
  * Rate limiting and quotas
- Google Cloud AI Platform:
  * Vertex AI services
  * Generative AI Studio
  * Model deployment

        * AutoML capabilities
    - Image Generation Tools:
        * Midjourney:
            · Discord interface usage
            · Prompt engineering for images
            · Style and quality parameters
            · Subscription and pricing
        * DALL-E 2:
            · API access and usage
            · Image generation and editing
            · Variation generation
            · Inpainting and outpainting
        * Stable Diffusion:
            · Open-source model
            · Local deployment
            · Integration options
            · Fine-tuning approaches
    - Other generative tools:
        * Stability AI ecosystem
        * Anthropic Claude API
        * Open-source alternatives
    - Tool selection criteria
    - Comparative analysis

Assessment and Evaluation

Continuous Assessment

- Weekly coding assignments and quizzes (30%)
- Module-based projects (40%)
- Capstone project (30%)

Evaluation Criteria

undefined

Learning Resources

Required Software and Tools
    - Python 3.8+ with Anaconda distribution
    - Jupyter Notebook
    - Git and GitHub
    - VSCode or PyCharm IDE
    - TensorFlow and Keras
    - PyTorch
    - Scikit-learn
    - Pandas and NumPy

- SQL database (PostgreSQL or MySQL)
- Power BI Desktop
- Hugging Face Transformers
- LangChain library
- Cloud CLI tools (AWS CLI, Azure CLI)

Recommended Reading
- "Hands-On Machine Learning" by Aurélien Géron
- "Deep Learning" by Goodfellow, Bengio, and Courville
- "Natural Language Processing with Transformers" by Lewis Tunstall, Leandro von Werra, and Thomas Wolf
- "Generative Deep Learning" by David Foster
- "The Hundred-Page Machine Learning Book" by Andriy Burkov

Online Resources
- Coursera and edX courses
- arXiv papers (https://arxiv.org)
- GitHub repositories
- PyPI documentation
- Kaggle datasets and competitions
- Hugging Face course (https://huggingface.co/course)
- Google Cloud AI/ML documentation
- AWS Machine Learning services
- Azure AI services

Course Delivery Format

Class Structure

- Lectures: Conceptual foundations with real-world examples (40%)
- Hands-on Labs: Practical implementation and coding (40%)
- Project Work: Real-world problem-solving (20%)

Prerequisites

- Basic Python programming knowledge
- Elementary statistics and probability
- Linear algebra fundamentals
- Familiarity with command-line interface

Hardware Requirements

- Minimum: 4GB RAM, 50GB storage
- Recommended: 8GB+ RAM, GPU (NVIDIA with CUDA), 100GB storage

---

Capstone Project Details

Objective

Students will design, develop, deploy, and present a comprehensive AI/ML application integrating concepts from the entire course.

Project Phases

1. **Ideation and Proposal (Week 1-2)**
   - **Problem definition**
   - **Literature review**
   - **Proposed solution architecture**
   - **Resource requirements**
2. **Data Collection and EDA (Week 3-4)**
   - **Data gathering**
   - **Exploratory analysis**
   - **Preprocessing pipeline**
   - **Data visualization**
3. **Model Development (Week 5-8)**
   - **Feature engineering**
   - **Model selection and training**
   - **Hyperparameter tuning**
   - **Evaluation and validation**
4. **Implementation and Deployment (Week 9-10)**
   - **Application development**
   - **API creation**
   - **Cloud deployment**
   - **Performance monitoring**
5. **Documentation and Presentation (Week 11-12)**
   - **Technical documentation**
   - **User guide**
   - **Final presentation**
   - **Peer review**

Project Evaluation Rubric

Criterion	Weight	Points
Problem Definition	10%	/10
Data Quality	15%	/15
Model Performance	25%	/25
Code Quality	20%	/20
Documentation	15%	/15
Presentation	15%	/15
Total	**100%**	**/100**

Capstone Project Evaluation Rubric

Course Outcomes

Upon successful completion, students will be able to:

1. **Analyze complex datasets using statistical and machine learning techniques**
2. **Design end-to-end AI/ML solutions for real-world problems**

3. Implement deep learning architectures for computer vision and NLP tasks
4. Apply transformer-based models and large language models in practical applications
5. Develop retrieval-augmented generation systems for intelligent knowledge processing
6. Deploy machine learning models on cloud platforms with proper monitoring
7. Evaluate AI/ML models using appropriate metrics and methodologies
8. Communicate findings and insights effectively through visualization and documentation

---

References and Citations

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
[2] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
[3] Tunstall, L., von Werra, L., & Wolf, T. (2022). *Natural Language Processing with Transformers*. O'Reilly Media.
[4] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.
[5] Foster, D. (2023). *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*. O'Reilly Media.
[6] Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
[8] Radford, A., Wu, J., Child, R., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.
[9] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.
[10] OpenAI. (2023). GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Appendix: Sample Course Schedule

Module 1: Python Programming

- Python fundamentals, syntax, data types
- Data structures and control flow
- Functions and scope
- Object-oriented programming
- File handling and exceptions
- Standard library modules + Project

Module 2: Data Analysis

- **NumPy and arrays**
- **Pandas fundamentals**
- **Data manipulation and aggregation**
- **Matplotlib and Seaborn visualization**
- **EDA case study + Web scraping project**

Continuation...

[Modules continue with similar structure for subsequent topics]

Conclusion

This comprehensive Data Science with Generative AI course curriculum provides a structured pathway from foundational programming concepts to cutting-edge applications of large language models and generative AI systems. The combination of theoretical knowledge, practical implementation, and real-world projects ensures students graduate with both conceptual understanding and hands-on expertise ready for industry application.

The curriculum emphasizes iterative learning, where each module builds upon previous knowledge, and incorporates modern tools and frameworks widely used in the industry. Through the capstone project and continuous evaluation, students demonstrate mastery of the course material and readiness for advanced roles in data science and AI/ML engineering.

Document Version: 1.0
Last Updated: December 2024
Target Audience: Engineering students and AI/ML professionals