

DATA SCIENCE WITH GEN AI / ML



Learn. Build. Innovate. Lead the Future.

Master the power of Data Science, Machine Learning and Generative AI to solve real-world problems and build intelligent solutions.

WHAT YOU'LL LEARN

- ✓ Python for Data Science
- ✓ Statistics & Data Analysis
- ✓ SQL for Data Science
- ✓ Data Cleaning & Data Wrangling
- ✓ Exploratory Data Analysis (EDA)
- ✓ Data Visualization with Power BI & Excel
- ✓ Machine Learning Algorithms
- ✓ Deep Learning Fundamentals
- ✓ Generative AI (LLMs, Prompt Engineering, RAG, LangChain)
- ✓ Real-world Projects & Capstone
- ✓ Deployment & MLOps Basics

WHY CHOOSE US?

- ✓ Industry-Aligned Curriculum
- ✓ Hands-on Real-world Projects
- ✓ Expert Mentors from Industry
- ✓ Placement Assistance
- ✓ Interview Preparation
- ✓ Flexible Batches & Support

TOOLS YOU WILL MASTER



Python



SQL



Excel



Power BI



Gen AI
(ChatGPT, LLMs)



Scikit-learn
(Machine Learning)



Jupyter
Notebook

WHO CAN JOIN?



Students &
Graduates



Working
Professionals



Developers &
Programmers



Anyone Passionate
About Data & AI

TURN DATA INTO DECISIONS.
BUILD THE FUTURE.
— JOIN TODAY! —



PRACTICAL
LEARNING



INNOVATIVE
APPROACH



CERTIFICATION
OF COMPLETION



CAREER
SUPPORT

Data Science Using Generative Artificial Intelligence (Gen AI) Course Contents

Contents

Microsoft Excel	1
Statistics	4
SQL (Structured Query Language)	7
Python	11
Feature Engineering	14
Machine Learning	15
Generative AI	17

Microsoft Excel

Excel serves as a widely-used tool for data analysis, offering functionalities such as sorting, filtering, and basic statistical calculations. Its user-friendly interface and spreadsheet-based structure make it accessible for organizing and analyzing datasets, making it a popular choice for small-scale data tasks and quick analyses.

Excel Fundamentals

Introduction to Excel

- ✓ Importance of Excel in Data Analysis
- ✓ Real-world use cases (reporting, dashboards, ad-hoc analysis)
- ✓ Understanding workbook, worksheets, rows, and columns
- ✓ Navigation and shortcuts for productivity

Data Handling & Preparation

Sorting & Filtering

- ✓ Sorting data (single & multi-level sorting)
- ✓ Applying filters for data exploration

- ✓ Advanced filtering techniques

Basic Formatting

- ✓ Cell formatting (number, date, currency)
- ✓ Font styling, alignment, borders
- ✓ Formatting best practices for reports

Flash Fill (Quick Transformations)

- ✓ Automatically extracting and transforming data
- ✓ Pattern recognition for fast data cleaning

Data Cleaning in Excel

- ✓ Removing duplicates
- ✓ Handling missing values
- ✓ Text-to-columns
- ✓ TRIM, CLEAN for data standardization

Data Highlighting & Insights

Conditional Formatting

- ✓ Highlighting key insights (top/bottom values, duplicates)
- ✓ Color scales, data bars, icon sets
- ✓ Rule-based formatting for dashboards

Excel Functions for Data Analysis

Text Functions

- ✓ LEFT(), RIGHT(), MID()
- ✓ LEN(), TRIM(), UPPER(), LOWER()
- ✓ CONCAT(), TEXT()
- ✓ Real-world use cases (cleaning messy data)

Statistical Functions

- ✓ SUM(), AVERAGE(), COUNT(), MAX(), MIN()
- ✓ MEDIAN(), MODE()
- ✓ Basic descriptive statistics

Logical Functions

- ✓ IF(), Nested IF

- ✓ AND(), OR(), NOT()
- ✓ Building decision-based conditions

Data Lookup & Retrieval

LOOKUP Functions

- ✓ VLOOKUP(), HLOOKUP(), XLOOKUP()
- ✓ Approximate vs exact match
- ✓ Limitations of VLOOKUP

INDEX & MATCH (Advanced Lookup)

- ✓ Dynamic lookups using INDEX + MATCH
- ✓ Replacing VLOOKUP

Data Validation & Control

- ✓ Creating drop-down lists
- ✓ Restricting data input (numbers, dates, text length)
- ✓ Error alerts and input messages
- ✓ Ensuring data accuracy in reports

Data Analysis & Reporting

Pivot Tables

- ✓ Creating Pivot Tables from datasets
- ✓ Summarizing large datasets
- ✓ Grouping (dates, categories)
- ✓ Filtering and sorting within Pivot Tables
- ✓ Calculated fields and values

Power Query

- ✓ Introduction to Power Query
- ✓ Importing data from multiple sources (CSV, Excel, Web)
- ✓ Removing columns
- ✓ Filtering rows
- ✓ Changing data types
- ✓ Merging and appending datasets

Automation in Excel

Recording Macros

- ✓ Automating repetitive tasks
- ✓ Recording and running macros
- ✓ Assigning macros to buttons
- ✓ Basic understanding of VBA

Data Visualization in Excel

- ✓ Creating charts:
 - Bar chart, line chart, pie chart, column chart
- ✓ Customizing visuals:
 - Titles, labels, legends, colors
- ✓ Best practices for clear visualization

Excel using AI

Tools Covered: Microsoft Excel

Statistics

Statistics is the backbone of data analysis, enabling data analysts to extract meaningful insights from raw data. It provides tools to summarize data (descriptive statistics), make predictions (inferential statistics), and identify trends or patterns. By understanding statistical measures such as mean, variance, and percentiles, data analysts can interpret datasets, validate hypotheses, and make data-driven decisions crucial for business strategies.

Descriptive Statistics (Understanding Data)

Measures of Central Tendency

- ✓ Mean (Average)
- ✓ Median
- ✓ Mode
- ✓ When to use each measure

- ✓ Impact of outliers on central tendency

Measures of Dispersion (Data Spread)

- ✓ Range
- ✓ Variance
- ✓ Standard Deviation
- ✓ Interquartile Range (IQR)
- ✓ Understanding variability in datasets
- ✓ Identifying consistency and volatility in data

Data Distribution (Understanding Patterns)

Normal Distribution

- ✓ Bell curve concept
- ✓ Real-world examples (heights, marks)

Skewness

- ✓ Left-skewed vs right-skewed data
- ✓ Impact on analysis

Binomial Distribution

- ✓ Success/failure scenarios
- ✓ Use cases in business decisions

Poisson Distribution

- ✓ Event occurrence over time
- ✓ Real-time examples (customer arrivals, defects)

Percentiles & Quartiles

- ✓ Understanding percentiles
- ✓ Quartiles (Q1, Q2, Q3)
- ✓ Interpreting data positions
- ✓ Use cases in ranking and segmentation

Probability Fundamentals

- ✓ Basics of probability
- ✓ Probability rules

Correlation & Covariance

Correlation

- ✓ Measures relationship between two variables
- ✓ Positive, Negative, and No correlation
- ✓ Correlation coefficient range (-1 to +1)

Covariance

- ✓ Direction of relationship between variables
- ✓ Positive vs Negative covariance
- ✓ Difference between covariance and correlation

Z-Score (Standardization)

- ✓ Understanding how far a data point is from the mean
- ✓ Standardizing different datasets for comparison
- ✓ Identifying outliers using Z-score

Sampling & Sampling Distribution

- ✓ What is sampling?
- ✓ Types of sampling (random, stratified, etc.)
- ✓ Importance of sampling in real-world data

Hypothesis Testing

- ✓ Null Hypothesis (H_0)
- ✓ Alternative Hypothesis (H_1)
- ✓ P-value concept
- ✓ Significance level (alpha)
- ✓ Type I Error & Type II Error
- ✓ Steps in hypothesis testing

Tools Covered: No Software tool required

SQL (Structured Query Language)

SQL plays a crucial role in data analysis by enabling users to extract, manipulate, and analyze data stored in relational databases efficiently. Its declarative nature allows for complex data queries, aggregations, and transformations, facilitating the extraction of meaningful insights from structured datasets.

Database Fundamentals

Introduction to SQL

- ✓ What is SQL and why it is important for Data Analysts
- ✓ Role of SQL in real-world analytics workflows

Database Design

- ✓ Tables, rows, columns
- ✓ Primary keys and foreign keys
- ✓ Normalization and Denormalization

Data Warehousing Concepts

- ✓ OLTP vs OLAP systems
- ✓ Introduction to Data Warehousing
- ✓ Fact and Dimension tables
- ✓ Star Schema vs Snowflake Schema
- ✓ Importance of data modeling in analytics

SQL Core Concepts

SQL Data Types

- ✓ Numeric, String, Date/Time types
- ✓ Choosing appropriate data types

SQL Operators

- ✓ Arithmetic, Comparison, Logical operators
- ✓ Real-time filtering use cases

SQL Commands

DDL (Data Definition Language)

- ✓ CREATE, ALTER, DROP

DML (Data Manipulation Language)

- ✓ INSERT, UPDATE, DELETE

DQL (Data Query Language)

- ✓ SELECT statements

DCL (Data Control Language)

- ✓ GRANT, REVOKE

TCL (Transaction Control Language)

- ✓ COMMIT, ROLLBACK

Filtering & Sorting Data

- ✓ WHERE clause
- ✓ HAVING clause
- ✓ ORDER BY
- ✓ LIMIT & OFFSET
- ✓ Writing efficient filtering conditions

Aggregations & Grouping

- ✓ GROUP BY clause
- ✓ Aggregate functions: COUNT, SUM, AVG, MIN, MAX
- ✓ Real-world reporting scenarios

Joins

- ✓ Inner Join
- ✓ Left Join
- ✓ Right Join
- ✓ Full Join

Advanced SQL for Data Analysis

Window Functions

- ✓ OVER clause
- ✓ Partitioning and ordering

LEAD() & LAG() Functions

- ✓ Time-based analysis
- ✓ Comparing previous and next records

Frames in SQL

- ✓ ROWS BETWEEN
- ✓ Running totals and moving averages

Conditional Logic

- ✓ CASE statements
- ✓ Creating derived columns
- ✓ Business rule implementation

Subqueries & CTEs

Subqueries

- ✓ Nested queries
- ✓ Correlated vs non-correlated subqueries

Common Table Expressions (CTEs)

- ✓ Writing modular and readable queries

Database Objects

Views

- ✓ Creating reusable query layers
- ✓ Simplifying complex queries

Triggers

- ✓ Automating actions on data changes
- ✓ Use cases (audit logs, validations)

Query Optimization & Performance

- ✓ Understanding query execution basics
- ✓ Indexing concepts (high-level)
- ✓ Writing optimized queries
- ✓ Avoiding common performance issues

Structured Problem Solving in SQL

- ✓ Breaking down complex problems
- ✓ Step-by-step query building approach
- ✓ Writing clean and efficient SQL logic
- ✓ Handling interview-style questions

Advanced SQL Queries

- ✓ Complex joins + aggregations
- ✓ Multi-level subqueries
- ✓ Real-world business problem scenarios
- ✓ Case study-based query solving

SQL Using AI

- ✓ Writing SQL Queries using AI

Real-Time Project: IMDB Movies Analysis

- ✓ Understanding dataset structure
- ✓ Data cleaning and preparation
- ✓ Writing analytical queries:
 - Top-rated movies
 - Genre analysis
 - Revenue trends
 - Actor/Director performance
- ✓ Applying joins, aggregations, and window functions
- ✓ Generating business insights

Tools Covered: MySQL Workbench

Python

Python serves as a powerhouse for data analysis, offering robust libraries like Pandas for manipulation, NumPy for numerical computing, and Matplotlib for visualization. With its rich ecosystem and easy-to-use syntax, Python empowers data analysts to efficiently explore, process, and visualize datasets, driving insights and decision-making

Core Python Fundamentals

Introduction to Python

- ✓ Importance of Python in Data Analysis
- ✓ Real-world use cases in analytics and business decision-making

Installation & Setup

- ✓ Installing Anaconda Distribution
- ✓ Overview of Jupyter Notebook for analysis

Variables & Data Types

- ✓ Integer, Float, String, Boolean
- ✓ Type casting and dynamic typing
- ✓ Understanding memory and variable assignment

Operators in Python

- ✓ Arithmetic, Comparison, Logical, Assignment operators
- ✓ Real-time use cases in data calculations

Arithmetic & String Operations

- ✓ Mathematical computations
- ✓ String manipulation for data cleaning (split, replace, format)

Control Flow & Functions

Conditional Statements

- ✓ if, elif, else for decision-making logic
- ✓ Real-time filtering scenarios

Loops

- ✓ for loops for iteration over datasets
- ✓ while loops for condition-based execution
- ✓ Practical use cases in data processing

User Input Handling

- ✓ Taking dynamic input for flexible programs

Functions

- ✓ Creating reusable functions
- ✓ Parameters, return values
- ✓ Writing modular and efficient data processing logic

Data Structures for Data Handling

Strings

- ✓ Indexing, Slicing
- ✓ String Functions

Lists

- ✓ Indexing, slicing, looping
- ✓ Use in storing and processing data collections

Tuples

- ✓ Immutable data structures and their use cases

Sets

- ✓ Removing duplicates, set operations

Dictionaries

- ✓ Key-value data storage
- ✓ JSON-like data handling (very important for APIs & real-world data)

Numerical Computing with NumPy

- ✓ Introduction to NumPy arrays
- ✓ Array creation and operations
- ✓ Mathematical and statistical functions
- ✓ Vectorized operations for performance optimization

- ✓ Use cases in large-scale numerical computations

Data Analysis with Pandas

- ✓ Introduction to Series and DataFrames
- ✓ Reading data from CSV, Excel, JSON
- ✓ Data inspection: head(), info(), describe()
- ✓ Data selection and filtering
- ✓ Handling missing values
- ✓ Data transformation and feature engineering
- ✓ GroupBy operations and aggregations
- ✓ Merging and joining datasets
- ✓ Sorting and ranking data

Data Visualization

Matplotlib

- ✓ Line charts, bar charts, histograms
- ✓ Customization (labels, titles, colors)

Seaborn

- ✓ Advanced visualizations (heatmaps, pairplots, boxplots)
- ✓ Statistical plotting for deeper insights

Data Cleaning & Preprocessing

- ✓ Identifying and handling missing values
- ✓ Removing duplicates
- ✓ Data type corrections
- ✓ Outlier detection and treatment
- ✓ String cleaning and formatting
- ✓ Preparing raw data for analysis

Exploratory Data Analysis (EDA)

Univariate Analysis

- ✓ Distribution analysis (mean, median, mode)

Bivariate Analysis

- ✓ Relationships between two variables (correlation, scatter plots)

Multivariate Analysis

- ✓ Analyzing multiple variables together
- ✓ Pattern identification and insights generation

Python using AI

- ✓ Performing Data Analysis using Python with the help of AI

Real-Time Project: Risk Analysis for Banking Data

- ✓ Understanding business problem (loan risk / credit analysis)
- ✓ Data cleaning and preprocessing of banking dataset
- ✓ Exploratory Data Analysis (EDA)
- ✓ Identifying high-risk customers
- ✓ Feature analysis and insights generation
- ✓ Creating visual dashboards using Python
- ✓ Presenting actionable business insights

Tools Covered: Jupyter Notebook, Google Colab

Feature Engineering

Feature Engineering empowers data scientists to extract, transform, and optimize features from raw datasets to enhance the accuracy and efficiency of Machine Learning models. By applying techniques such as data preprocessing, feature scaling, encoding, outlier treatment, and feature selection, organizations can build robust predictive models and derive actionable insights from data.

- ✓ Introduction to Feature Engineering
- ✓ Missing Value Treatment
- ✓ Handling Categorical Variables
- ✓ Label Encoding
- ✓ One Hot Encoding

- ✓ Ordinal Encoding
- ✓ Standardization
- ✓ Normalization
- ✓ Outlier Detection and Treatment
- ✓ Binning Techniques
- ✓ Train Test Split
- ✓ Feature Engineering using Scikit-Learn

Machine Learning

Machine Learning (ML) empowers data professionals to build intelligent systems that learn patterns from historical data and make accurate predictions or decisions without being explicitly programmed. It enables organizations to solve complex business problems through predictive analytics, classification, clustering, and forecasting, driving data-driven decision-making across various industries.

Linear Regression

- ✓ Introduction to Simple Linear Regression
- ✓ Simple Linear Regression in Python
- ✓ Gradient Descent-Linear Regression
- ✓ Multiple Linear Regression
- ✓ Multiple Linear Regression in Python
- ✓ Industry Relevance of Linear Regression

Logistic Regression

- ✓ Univariate Logistic Regression
- ✓ Multivariate Logistic Regression – Model Building
- ✓ Multivariate Logistic Regression – Model Evaluation
- ✓ Multiclassification Logistic Regression
- ✓ Industry Relevance of Logistic Regression

K-Mean Clustering

- ✓ Introduction to Clustering
- ✓ K Means Clustering
- ✓ Executing K Means in Python
- ✓ Hierarchical Clustering

Decision Trees

- ✓ Introduction to Decision Trees
- ✓ Algorithms for Decision Tree Construction
- ✓ Hyperparameter Tuning in Decision Trees
- ✓ Ensembles and Random Forests

Boosting

- ✓ Introduction to Boosting
- ✓ Ensemble Models
- ✓ Introduction to AdaBoost
- ✓ AdaBoost Algorithm
- ✓ AdaBoost Distribution and Parameter Calculation
- ✓ Introduction to Gradient Boosting
- ✓ Gradient Boosting Algorithm
- ✓ XGBoost Algorithm

Time Series Forecasting

- ✓ Types of Forecasting
- ✓ Smoothing Techniques
- ✓ Regression vs Time Series
- ✓ Holt's Exponential Smoothing
- ✓ Introduction Auto Regressive (AR) Models
- ✓ ACF and PACF
- ✓ Moving Average Model (MA)
- ✓ Auto Regressive Moving Average (ARMA)
- ✓ Auto Regressive Integrated Moving Average (ARIMA)
- ✓ Seasonal Auto Regressive Integrated Moving Average (SARIMA)

Principal Component Analysis

- ✓ Vectorial Representation of Data
- ✓ Change of Basis
- ✓ Directions of Maximum Variance
- ✓ Covariance Matrix
- ✓ Eigen decomposition
- ✓ Algorithm of Principle Component Analysis
- ✓ Applying PCA using Python
- ✓ Scree Plots
- ✓ Dimensionality Reduction

Naïve Bayes

- ✓ Introduction to Naive Bayes Classification
- ✓ Bayes Theorem and Conditional Probability
- ✓ Types of Naive Bayes
- ✓ Naive Bayes in Python

Bike Sharing Project

Lead Scoring Project

Telecom Churn Project

Tools Covered: Jupyter Notebook, Google Colab

Generative AI

Generative AI empowers data analysts to accelerate data analysis by automating data cleaning, generating insights, and enabling natural language interactions with data. It helps transform raw structured and unstructured data into meaningful summaries, visualizations, and predictive insights. With its ability to assist in code generation, pattern discovery, and decision support, GenAI enhances productivity and enables analysts to build intelligent, scalable, and insight-driven data workflows efficiently.

Landscape of Generative AI

- ✓ Introduction to Generative AI Landscape
- ✓ What is Generative AI?
- ✓ The Generative AI Tech Stack
- ✓ Paradigms of Artificial Intelligence: ML and DL
- ✓ Paradigms of Artificial Intelligence: Gen AI

Working with OpenAI APIs

- ✓ Introduction to OpenAI Models
- ✓ OpenAI API Instructions and Best Practices

Chat Completions API

- ✓ Single-turn Conversations with Chat Completions API
- ✓ Multi- turn Conversations with Chat Completions API
- ✓ Tracking API usage and costs

Prompt Engineering

- ✓ What and Why of Prompts
- ✓ Engineering a Good Prompt
- ✓ Applying Prompt Engineering
- ✓ Types of Prompting

Retrieval Augmented Generation (RAG)

- ✓ Basics of RAG
- ✓ Why RAG?
- ✓ What is RAG?
- ✓ Semantic Search System

Vector Stores

- ✓ Types of Vectors Stores
- ✓ Vector Database – Chroma
- ✓ Creating Chroma Collections
- ✓ Querying Chroma Collections
- ✓ Updating Chroma Collections

AI + SQL

- ✓ Converting natural language to SQL
- ✓ Query Optimization using AI

AI + Python

- ✓ Generate Pandas Code using AI
- ✓ Data Cleaning Automation

LLM Frameworks/LangChain

- ✓ Why frameworks are needed
- ✓ Introduction to LangChain
- ✓ Chains
- ✓ Agents

Tools Covered: Google Colab, Open AI API, Chroma