Coincent 3-Year Program in Data Science with Python
Partnered by Worisgo

Year 1: Live Industrial Training – Build Your Foundation

Gain hands-on industry exposure from day one with 2.5 months of live training in a professional environment. Learn the latest tools and technologies through skill-focused sessions, guided by expert mentors from the industry

Data Science Curriculum

Chapter 1 – Introduction to Data Science

1.1 Introduction to Data Science

- What is Data Science?
- Applications across industries (healthcare, finance, marketing, etc.)
- Difference between Data Science, AI, ML, and Big Data

1.2 Data Science Pipeline

- Problem definition
- Data collection
- Data cleaning & preprocessing
- Exploratory Data Analysis (EDA)
- Feature Engineering
- Model Building
- Evaluation & Deployment

1.3 Data Science Career

- Roles: Data Analyst, Data Engineer, Data Scientist, ML Engineer
- Tools: Python, R, SQL, Tableau, Spark
- Career paths and skill roadmap

Chapter 2 – Introduction to Python for Data Science

2.1 Introduction to Python

- Why Python for Data Science?
- Python installation (Anaconda/Jupyter/VSCode)

2.2 First Program

- print(), understanding code structure

2.3 Comments

- Single-line, multi-line, docstrings

2.4 Data Types

- 2.4.1: Numbers, Strings
- 2.4.2: Lists, Tuples, Sets, Dictionaries

2.5 Operators

- 2.5.1: Arithmetic, Comparison
- 2.5.2: Logical, Bitwise, Membership

2.6 Variables

- 2.6.1: Variable declaration, rules
- 2.6.2: Type casting

2.7 Syntax

- Python indentation, blocks, line continuation

2.8 Working with Numbers

- math module, rounding, random numbers

2.9 Working with Strings

- 2.9.1: Basic operations, slicing
- 2.9.2: String methods, formatting

2.10 Date & Time

- Working with datetime module

2.11 Conditional Statements

- 2.11.1: if, else
- 2.11.2: elif, nested conditions

2.12 For Loop

- 2.12.1: Basic loop, range()
- 2.12.2: Nested loops, break, continue

2.13 While Loop

- Loop conditions and control

2.14 Lists, Tuples, Sets

- 2.14.1: List operations
- 2.14.2: Tuple properties
- 2.14.3: Set operations

2.15 Dictionaries

- 2.15.1: Keys, values
- 2.15.2: Methods, nesting

2.16 Functions

- 2.16.1: Defining functions, parameters, return, scope

2.17 NumPy

- 2.17.1: Arrays, creation, indexing
- 2.17.2: Array operations, broadcasting
- 2.17.3: Statistical & mathematical functions

2.18 Matplotlib

- 2.18.1: Line & bar charts
- 2.18.2: Histograms, pie charts, customization

2.19 Pandas

- 2.19.1: Series, DataFrame creation, indexing
- 2.19.2: Data cleaning, merging, groupby, filtering

Chapter 3 – Mathematics for Data Science

3.1 Overview of Math Requirements

- Algebra, statistics, probability, linear algebra

3.2 Introduction to Statistics

- Mean, median, mode

3.3 (Duplicate removed)

3.4 Measure of Spread

- Variance, standard deviation, IQR, range

3.5 Probability

- Classical, conditional, joint and marginal probability

3.6 Conditional Probability

- Bayes Theorem, independence

3.7 Data Scientist vs Statistician

- Skills, tools, responsibilities

3.8 Data Preprocessing

- Normalization, standardization, handling missing values

Chapter 4 – Introduction to Machine Learning

4.1 Introduction to ML

- What is ML, differences with AI & DL

4.1.1 Types of ML Algorithms

- Supervised, Unsupervised, Reinforcement

4.2 Steps in Building ML Model

- Define problem → Collect Data → Clean Data → Train → Test → Evaluate

4.3 Linear Regression

- Concept, simple & multiple regression, evaluation metrics

4.4 Logistic Regression

- Binary classification, sigmoid function

4.5 K-Nearest Neighbors (KNN)

- Distance metric, choosing K, overfitting/underfitting

4.6 Naive Bayes

- Bayes theorem, spam classification

4.7 Clustering Overview

- Types and applications

4.8 K-Means Clustering

- Centroid selection, elbow method

4.9 Hierarchical Clustering

- Dendrograms, Agglomerative vs Divisive

4.10 Dimensionality Reduction

- Purpose and techniques

4.10.1 Principal Component Analysis (PCA)

- Variance, eigenvalues, dimensional reduction

4.10.2 Linear Discriminant Analysis (LDA)

4.11 Supervised vs Unsupervised Learning

- Characteristics and examples

4.12 Semi-Supervised Learning

- Real-world examples, pseudo-labeling

4.13 Reinforcement Learning

- 4.13.1: Agents, environment, reward system
- 4.13.2: Q-learning, exploration vs exploitation

4.14 Data Science vs Machine Learning

- Comparison table and use cases

Chapter 5 – Deep Learning and Data Mining

5.1 Deep Learning

- Neural networks, perceptrons, backpropagation

5.2 TensorFlow and Keras

- Building a neural network, training, evaluating

5.3 Data Mining

- Patterns, knowledge discovery, CRISP-DM

5.3.1 Preprocessing in Data Mining

- Feature selection, discretization, transformation

5.4 Natural Language Processing

- Tokenization, stemming, lemmatization

5.4.1 Preprocessing in NLP

- Stopword removal, word embeddings (TF-IDF, Word2Vec)

Year 2: Real-Time Projects – Apply What You've Learned

Transform your knowledge into real-world experience by working on 8 industry-level projects that build your technical and professional skills. Each project enhances your portfolio, strengthening your resume and showcasing your practical abilities. You'll also collaborate in teams, gaining valuable experience in communication, teamwork, and project management—just like in a real work environment.

PROJECTS

Hierarchical Clustering

Hierarchical Clustering is an unsupervised learning technique used to group similar data points into clusters based on distance metrics. It builds a tree-like structure (dendrogram) to show how clusters are merged or split. The algorithm does not require the number of clusters to be specified in advance. It is useful in market segmentation, social network analysis, and gene classification. Tools like **Scikit-learn** and **SciPy** are commonly used.

ChatBot

A ChatBot is an AI-powered tool designed to simulate human-like conversations using natural language processing (NLP). It can be rule-based or machine learning-based, handling tasks like customer support or appointment scheduling. ChatBots learn from data and improve over time, making them increasingly accurate. Libraries like **NLTK**, **spaCy**, and frameworks like **Rasa** or **Dialogflow** are widely used. It enhances user engagement and automation.

Linear Discriminant Analysis (LDA)

LDA is a supervised dimensionality reduction technique that maximizes class separability by projecting data into lower dimensions. It is commonly used in classification tasks to improve performance and interpretability. LDA assumes normally distributed classes and equal class covariances. It's effective in fields like face recognition and medical diagnosis. Tools like **Scikit-learn** offer built-in LDA support.

Hate Speech Detection

Hate Speech Detection is a text classification task that identifies and filters offensive or harmful content online. It involves preprocessing text, feature extraction (TF-IDF or embeddings), and training classification models like Logistic Regression or LSTM. It's vital for maintaining safe online platforms and social media moderation. Datasets like **Twitter Hate Speech** and tools like **TensorFlow**, **scikit-learn**, and **BERT** are commonly used.

SMS Spam Classification

SMS Spam Classification is a binary text classification project that identifies whether a message is spam or ham (not spam). It involves preprocessing the text (removing stop words, tokenization), feature extraction using TF-IDF or CountVectorizer, and training models like Naive Bayes or Logistic Regression. This project helps automate spam detection in messaging systems. Evaluation metrics like accuracy, precision, and recall are used to assess performance. Tools used include **Scikit-learn**, **NLTK**, and **Pandas**.

Surprise Housing Case Study

The Surprise Housing Case Study aims to build a regression model to predict house prices based on features like location, size, and amenities. It involves exploratory data analysis (EDA), feature engineering, multicollinearity checks, and model building using linear regression. The goal is to provide accurate price predictions to aid decision-making in real estate. The project also focuses on model interpretability and performance evaluation. Tools used include **Pandas**, **Matplotlib**, **Seaborn**, and **Scikit-learn**.

Credit EDA (Exploratory Data Analysis)

Credit EDA involves analyzing credit-related datasets to uncover patterns, detect anomalies, and understand borrower behavior. It includes univariate and bivariate analysis, missing value treatment, and correlation study. Visualizations like histograms, heatmaps, and box plots help reveal relationships between features and credit risk. The insights guide the development of credit scoring or risk models. Tools used include **Pandas**, **Seaborn**, **Matplotlib**, and **NumPy**.

Year 3 – Placement & Internship Phase:

In the 3rd year of Coincent's program, students are guaranteed an internship with partner companies, complete with a formal Internship Offer Letter and a Completion Certificate upon successful completion. This internship is a complimentary part of the 3-Year "Industrial Training + Internship" model, which also includes live classes, expert mentorship, and hands-on project work. This phase bridges academic learning with real-world application, providing students with valuable professional exposure before graduation.

Coincent also offers structured placement preparation to ensure students are job-ready. This includes portfolio building through 8 real-time projects, certifications aligned with Microsoft standards, and dedicated training for interviews. From mock interviews to resume reviews and HR/technical round prep, every element is designed to transition students from classroom learning to career success. By the 4th year, students are equipped not just with knowledge, but with experience, credentials, and confidence to enter the workforce.