Coincent
Learning Drives Excellence

Coincent 3 Year Program Curriculum
MACHINE LEARNING

Partnered by

Microsoft

WORLD RISK & GOVERNANCE

Empowering Learners,
Accelerating Careers.

coincent.ai

ABOUT COINCENT

Coincent offers a 3-Year Program that is a well-structured, career-focused initiative designed to equip students with practical skills, real-world experience, and strong placement support. The program is tailored to ensure progressive learning and career readiness across three key phases.

Why It's Unique

- Only one batch per year with limited seats (150 students) per Domain to maintain quality.
- Prepares students step-by-step to become job-ready by graduation.

support@coincent.ai
993, 20th Cross Rd, 7th Sector, HSR Layout,
Bengaluru, Karnataka 560102

DETAILED ABOUT COINCENT 3 YEAR AI PROGRAM

"ML Program at Coincent – Learn by Doing"

Coincent's ML program is designed to help students to build real-world skills through a structured, hands-on learning experience. With a blend of live sessions, recorded lectures, and guided projects, students gain deep insights into ML fundamentals, machine learning, and neural networks — even without prior experience.

Here, mentors from top MNC's will guide you and assist you through the sessions and live projects, training etc.....

support@coincent.ai
993, 20th Cross Rd, 7th Sector, HSR Layout,
Bengaluru, Karnataka 560102

3-Year Program Structure Breakdown

TABLE OF CONTENTS

Year 1 :- Industrial Training

– "Before going to the curriculum preparation, Let's see the Basic Intro about ML and how coincent is going to deep dive into its chapters" –

What makes this program unique is

- it runs **parallel to your college academics**, allowing you to learn, apply, and grow without compromising your education.
- Through hands-on projects, Microsoft certifications, real-world mentorship, and a clear yearly progression — you'll be industry-ready by the time you graduate.

Program Overview

What Machine Learning is and why it's important?

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that focuses on enabling machines to learn from data and improve their performance over time without being explicitly programmed.

In simple words:
Machine Learning teaches computers to learn from experience (data) just like humans do. Instead of writing step-by-step instructions, we

feed data to algorithms, and the machine learns patterns and makes decisions.

Types of Machine Learning:

1. Supervised Learning
   Learns from labeled data (e.g., images tagged as “cat” or “dog”)

2. Unsupervised Learning
   Finds patterns in unlabeled data (e.g., grouping similar customers)

3. Reinforcement Learning
   Learns by trial and error, using rewards and penalties (e.g., training robots or game-playing AIs)

Importance of ML

Machine Learning is important because it helps automate complex tasks, analyze massive amounts of data, and make intelligent decisions faster than humans can. Here's why it matters:

- Solves Real-World Problems
- Handles Huge Data Efficiently
- Improves with Time
- Drives Innovation in Every Industry
- Enables Personalization

1.Introduction to Python (For ML Foundations)

1.1 Python Crash Course Introduction

- What is Python?
- Use cases in AI/ML/Data Science

1.2 Python Demo & Installation

- Installing Python (Anaconda/Miniconda)
- Jupyter Notebook setup

1.3 Python Introduction and Installation

- Python IDEs, interpreters
- Running .py scripts

1.4 Basic Python and Datatypes

- int, float, bool, str
- Type conversion

1.5 Numbers & Strings

- String slicing, formatting
- Arithmetic & logical operations

1.6 Data Types

- Lists, Tuples, Sets, Dictionaries

Benefits and Outputs:

- Understand Python's importance in AI, ML, and data science.
- Set up Python using Anaconda and run code in Jupyter Notebooks or .py scripts.
- Learn core data types, type conversion, arithmetic, and string operations.
- Work with key data structures: lists, tuples, sets, and dictionaries.
- Outcome: Build a strong Python foundation to support advanced programming and data tasks

2. Control Flow

2.1 If-Else Conditions

- Conditional expressions
- Nested if, elif

2.2 While & For Loops

- Loop control: break, continue
- Iterating through collections

Benefits and Outputs:

- Understand how to use conditional expressions for decision-making.
- Learn loop structures (for, while) and control flow using break and continue.
- Practice iterating over data collections like lists and strings.

- Outcome: Write structured, logic-driven programs for automation and control.

3. Exception Handling

3.1 Exception Handling

- Try-Except-Else-Finally
- Common exceptions in ML workflows (e.g., ZeroDivisionError, ValueError)

Benefits and Outputs:
Learn to handle runtime errors gracefully using `try-except-else-finally` blocks, ensuring smooth program execution.
Outcome: Build robust ML and data processing scripts that can manage common exceptions like `ZeroDivisionError` and

4. Functions

- Defining functions
- Parameters, default values
- Lambda functions
- Scope and return values

Benefits and Outcomes:
Understand how to define reusable functions with parameters, default values, and return statements for clean, modular code.
Outcome: Write concise logic using lambda functions and manage variable scope effectively in larger programs.

5. Object-Oriented Programming (OOP)

5.1 Classes

- Objects, attributes, methods

5.2 OOP

- Inheritance, Polymorphism
- Encapsulation and abstraction
- Relevance to ML (e.g., custom model classes)

Benefits and Outputs:

- Learn object-oriented programming concepts like classes, objects, inheritance, and polymorphism to structure code efficiently.
- Outcome: Build scalable applications and custom components (e.g., model classes) relevant to real-world ML workflows

6. Deep Learning Basics

6.1 Logistic Regression vs Deep Learning

- When to use Logistic Regression vs Neural Networks
- Linear boundaries vs non-linear decision making

6.2 TensorFlow and Keras

- Building a neural network in Keras
- Layers, activation functions, compiling, training, evaluating

Benefits and Outputs:

- Understand the differences between logistic regression and neural networks, including when to use linear vs. non-linear models.

- Outcome: Gain practical skills in building, training, and evaluating neural networks using TensorFlow and Keras for real-world AI task

7. Python Libraries for ML

7.1 Introduction to Libraries

- Why libraries matter in ML
- Popular libraries overview

7.2 Library Introduction

- Setting up with pip or conda

7.3 Matplotlib

- Plotting graphs
- Visualizing data & model performance (loss curves, confusion matrices)

7.4 NumPy

- Arrays, matrix operations
- Random number generation

7.5 Pandas

- DataFrames, filtering, grouping
- Handling missing values

Benefits and Outputs:

- Learn the importance of libraries in streamlining machine learning workflows and data analysis.
- Outcome: Use key libraries like Matplotlib for visualization, NumPy for numerical operations, and Pandas for efficient data handling and preprocessing.

8. Math for Machine Learning

8.1 Data

- Types: categorical, numerical, ordinal
- Data collection and integrity

8.2 Linear Algebra

- Vectors, matrices, dot product
- Applications in ML: weights, embeddings

8.3 Statistics

- Mean, median, mode
- Variance, standard deviation

8.4 Probability & Stats

- Distributions: binomial, normal, Poisson
- Bayes theorem

Benefits and Outcomes:

- Understand essential data types and the role of clean, well-structured data in ML success.
- Outcome: Build a strong mathematical foundation using linear algebra, statistics, and probability to support core machine learning concepts like model weights, data distributions, and inference.

9. Probability & Data Visualization

9.1 Introduction to Probability, Statistics & SQL

- Probability in ML models
- Basic SQL for data querying

9.2 Data Visualization with Tableau

- Dashboards, plots, charts
- Connecting Tableau to data sources

9.3 LSTM (Intro)

- Long Short-Term Memory networks
- Use in time series and NLP
- Basic model using Keras

Benefits and Outcomes:

- Learn how probability supports ML predictions and use SQL for effective data querying and manipulation.
- Outcome: Visualize insights using Tableau dashboards and gain an introductory understanding of LSTM networks for handling sequential data in time series and NLP tasks.

10. Machine Learning Models

10.1 Clustering

- K-Means Clustering
- Hierarchical clustering
- Applications: customer segmentation

10.2 Evaluation Metrics

- Accuracy, Precision, Recall, F1-score
- Confusion matrix, ROC-AUC
- Cross-validation

10.3 Logistic Regression – Feature Regression

- Feature selection: correlation, chi-squared test
- Feature importance, regularization

10.4 Logistic Regression

- Binary classification
- Sigmoid function
- Decision boundary

10.5 Simple Linear Regression

- Linear relationship
- Least squares method
- Evaluation: R^2, MAE, MSE

10.6 Multiple Linear Regression

- Handling multiple predictors
- Multicollinearity

Benefits and Outputs:

- Learn core machine learning techniques like clustering and regression, along with how to evaluate model performance using key metrics.
- Outcome: Apply K-Means for segmentation, build classification/regression models, and interpret results using tools like ROC-AUC, R^2, and feature importance for better decision-making.

Year 2 – Application & Project Phase

– Year 2 is full of hands-on-experience on 8 live projects –

PROJECTS

1. **Restaurant Review using NLP**

   This project uses Natural Language Processing (NLP) to analyze and classify restaurant reviews as positive or negative. It involves text preprocessing, tokenization, and feature extraction using TF-IDF or Word2Vec. Machine learning models like Logistic Regression or Random Forest are trained on labeled data. The goal is to automate sentiment analysis and enhance customer feedback interpretation. It helps restaurants track customer satisfaction and improve service. The project includes evaluation using accuracy and confusion matrix. Tools used: **Python**, **NLTK**, **Scikit-learn**, **Pandas**.

2. **Self-Driving Car**

   This project simulates a self-driving car using machine learning and computer vision techniques. It detects road lanes, signs, and objects using OpenCV and deep learning models like CNNs. Real-time decisions are made based on sensor input or video feed. Reinforcement learning can be applied for path optimization. It is often trained in simulators before real-world deployment. The project demonstrates automation in the transportation sector. Tools: **TensorFlow/Keras**, **OpenCV**, **Python**,

Carla/DonkeyCar.

3. **Vehicle Price Prediction**

   This regression-based project predicts the selling price of vehicles based on features like model, mileage, fuel type, and age. It uses data preprocessing, encoding, and model building with algorithms like Linear Regression, Decision Trees, or XGBoost. The project aims to help users and dealerships estimate fair vehicle pricing. Feature importance analysis enhances model interpretability. Performance is measured using RMSE or R^2 score. Tools: **Pandas**, **Scikit-learn**,**Matplotlib**, **Seaborn**.

4. **Dogs vs Cats using CNN**

   This classification project uses Convolutional Neural Networks (CNNs) to distinguish between images of dogs and cats. The model learns visual features like ears, fur, and shapes from labeled images. The dataset is augmented and normalized for improved training. Accuracy improves with deeper CNN architectures and dropout layers. It's a foundational project in computer vision and image classification. Tools: **Keras**, **TensorFlow**, **OpenCV**, **Matplotlib**.

5. **Lead Scoring Case Study**

   This project builds a classification model to score and prioritize sales leads based on conversion likelihood. It uses historical lead data with features like source, activity level, and demographics. Logistic Regression or Decision Trees are commonly applied. It helps sales

teams focus on high-potential leads. The model is evaluated using precision, recall, and ROC curve. Tools: **Python, Scikit-learn, Pandas, Seaborn**

6. **Diabetes Prediction using ML**

   This binary classification task predicts whether a patient is likely to develop diabetes based on medical parameters like BMI, glucose level, and age. The Pima Indians Diabetes dataset is commonly used. Models like Logistic Regression, Random Forest, or SVM are trained and evaluated. Feature scaling and handling class imbalance are key steps. The goal is to assist early diagnosis and preventive care. Tools: Scikit-learn, Pandas, Matplotlib, NumPy.

7. **Number of Orders Prediction**

   This time series or regression project predicts the future number of orders in a business or delivery service. It uses historical order data, seasonality patterns, and external factors (e.g., holidays). Models like Linear Regression, ARIMA, or LSTM are used depending on the data type. Accurate predictions help in inventory and logistics planning. The model is validated using RMSE or MAE.

   Tools: Python, Scikit-learn, Statsmodels, Keras.

8. Bike Sharing Demand Prediction

This project predicts the demand for bike rentals based on features like weather, time, and day. It uses regression models such as Random Forest, Gradient Boosting, or Neural Networks. EDA is performed to analyze trends and correlations in user behavior. Feature engineering improves accuracy by incorporating time-based variables. The model supports operational planning in urban mobility. Tools: Python, Pandas, Scikit-learn, Matplotlib.

Year 3 – Placement & Internship Phase:

1. Guaranteed Internship Phase

   In Year 3, Coincent guarantees an internship with partner companies for every student at no extra cost. The internship includes a formal Internship Offer Letter and a Completion Certificate upon successful completion.

   This is part of their "Industrial Training + Internship" model — training fees cover live classes, mentorship, and project work, but the internship phase itself is completely complimentary.

2. Structured Placement Preparation

   Coincent supports students in portfolio-building with multiple completed projects (typically around 9+) and Microsoft-aligned certifications .

   They provide mock interviews, resume reviews, and training for HR and technical rounds — all aimed at preparing you for real-world hiring.

3. Final Take

Coincent's 3rd year transforms theory into practical experience through a guaranteed internship, builds a robust credentials portfolio, and equips you with placement-ready skills via mock interviews and resume prep. If you're in your 4th year, this phase sets you on a clear trajectory from "training" to "hired."

Step Into Top Tech Roles

The leading and high-demand roles in the Machine Learning field, along with a brief description of each:

Machine Learning Engineer

Role: Designs, builds, and deploys ML models into production systems.

Skills: Python, TensorFlow/PyTorch, model training, MLOps, data pipelines.

Data Scientist

Role: Analyzes and interprets complex data to inform decisions; uses ML for predictions.

Skills: Python, R, SQL, Pandas, Scikit-learn, statistical modeling, data visualization.

AI Engineer

Role: Develops AI applications like chatbots, recommendation engines, and vision systems.

Skills: ML + Deep Learning (DL), Natural Language Processing (NLP), OpenCV, TensorFlow.

Deep Learning Engineer

Role: Specializes in building deep neural networks for image, video, or speech-based applications.

Skills: CNNs, RNNs, LSTM, PyTorch, TensorFlow, GPUs.

Data Engineer

Role: Builds and manages data pipelines and architectures to support ML workflows.

Skills: SQL, ETL tools, Spark, Hadoop, cloud platforms (AWS, GCP, Azure).

Computer Vision Engineer

Role: Works on image/video processing applications like object detection or facial recognition.

Skills: OpenCV, YOLO, CNNs, PyTorch, TensorFlow.

NLP Engineer

Role: Develops models for language understanding — chatbots, translators, sentiment analysis.

Skills: Transformers, BERT, spaCy, NLTK, HuggingFace.

ML Ops Engineer

Role: Focuses on automating and scaling ML model deployment and monitoring.

Skills: Docker, Kubernetes, CI/CD, cloud ML tools, model versioning.