Machine Learning with Python(M1)

Created	@December 8, 2025 12:55 PM
Module Code	IBMM01: Introduction to Machine Learning

Module 1: Introduction to Machine Learning

Overview of Machine learning

Machine learning (ML) is a subset of AI that uses algorithms to learn from data. It typically requires practitioners to manually engineer features. Deep learning, by contrast, uses multi-layered neural networks that automatically extract features from complex, unstructured data.

Machine learning (ML) is a part of Artificial Intelligence that allows computers to learn patterns from data and make predictions or decisions **without being explicitly programmed.**

To choose a machine learning technique, you look at:

- **What problem you want to solve**
- **Your data type**
- **Your resources**
- **Your desired output**

1. Supervised Learning Techniques

These require labeled data (input + correct output).

a) Classification

Used when the output is a **category/class**.

Examples:

- Benign vs malignant cell
- Will the customer churn or not
- Spam vs not spam

The model learns from labeled examples and predicts the class of new cases.

b) Regression

Used when the output is a **continuous numerical value**.

Examples:

- Predicting house prices
- Predicting car CO_2 emissions

The model predicts a number, not a category.

---

2. Unsupervised Learning Techniques

These work on **unlabeled data**.

a) Clustering

Groups similar items together.

Examples:

- Customer segmentation in banking
- Grouping similar patients

A clustering algorithm discovers natural groups in your data.

b) Association

Finds items that frequently occur together.

Examples:

- Grocery products bought together
- Market-basket analysis

c) Anomaly Detection

Finds unusual or abnormal data points.

Example:

- Credit card fraud detection

d) Sequence Mining

Predicts the next event in a sequence.

Example:

- Website clickstream prediction

e) Dimensionality Reduction

Reduces the number of features to make data smaller and models faster.

Example:

- PCA used in large datasets

f) Recommendation Systems

Suggest items based on user behavior and similarity with other users.

Examples:

- Netflix recommending movies
- Amazon product suggestions

---

3. Example Application: Cancer Cell Classification

You collect thousands of cell samples with measurements like:

- Clump thickness
- Cell size
- Cell adhesion

These features differ between **benign and malignant** cells.

Using this data:

1. You clean the dataset
2. Choose a classification algorithm
3. Train the model
4. The model learns patterns from previous samples
5. It predicts whether a new cell is benign or malignant

This improves early detection and saves lives.

---

4. Machine Learning Use Cases in Real Life

- **Amazon/Netflix** → recommend items/movies
- **Banks** → predict loan default probability
- **Telecom** → predict customer churn
- **Computer Vision** → identify cats vs dogs
- **Face unlock**, **chatbots**, **games**, etc.

ML models automatically learn patterns like eyes, ears, tail, shapes, etc., to differentiate cats and dogs.

Machine Learning Model Lifecycle

A Machine Learning project follows a **lifecycle**—a set of stages from start to finish. In reality, this process is **iterative**. You move back and forth between stages as needed.

1. Problem Definition

This is the most important step.

Clearly define:

- What problem you want to solve

- What outcome you expect
- Why ML is needed

Example: Recommend the best beauty products for customers.

2. Data Collection

Gather data from sources such as:

- Databases
- APIs
- User logs
- Sensors
- Third-party datasets

This begins the **ETL (Extract, Transform, Load)** process.

3. Data Preparation

Data preparation includes:

- Cleaning the data (removing errors and duplicates)
- Transforming data (normalization and encoding)
- Creating features
- Storing the cleaned data in a central location, such as a data warehouse

This ensures the data is ready for training.

4. Model Development & Evaluation

The ML engineer:

- Selects the algorithm
- Trains the model
- Tunes hyperparameters
- Evaluates performance using metrics
- Compares models to choose the best one

Evaluation ensures the model is accurate and reliable.

5. Model Deployment

Once the model performs well, it is:

- Deployed to production
- Integrated into the product or application
- Used by real users

Examples:

- A recommendation model suggesting beauty products
- A fraud detection model running in a banking app

Iterative Nature

If the model fails in deployment or becomes outdated:

- Revisit earlier steps
- Collect more data
- Refine the problem definition
- Retrain or rebuild the model

This cycle continues, improving the model over time.

Data Scientist / AI Engineer

Data Scientists

- Focus on analyzing structured data
- Do descriptive + predictive analytics
- Use small ML models: regression, classification, clustering
- Build insights and predictions

AI Engineers

- Focus on building generative AI applications
- Work with huge amounts of unstructured data

- Use foundation models (LLMs, multimodal models)
- Build systems like chatbots, coding assistants, recommendation engines

The rise of generative AI has created a new field—**AI Engineering**—which builds large-scale intelligent applications using powerful pre-trained foundation models.

Machine learning tools are software libraries, frameworks, and platforms that help manage the complete machine learning pipeline—from data preprocessing to model building, evaluation, optimization, and deployment. These tools simplify complex tasks like handling big data, performing statistical analyses, and building predictive models. For example, **Pandas** helps with data manipulation, while **Scikit-learn** provides traditional machine learning algorithms.

Programming languages serve as the backbone for implementing these tools. **Python** is the most widely used language because of its rich ecosystem of libraries for data processing, statistical analysis, visualization, and model development. **R** is preferred for statistical computing. Other languages like **Julia**, **Scala**, **Java**, and **JavaScript** are used for high-performance computing, big data processing, production systems, and web-based machine learning applications.

Machine learning tools fall into several major categories:

---

1. Data Processing and Analytics Tools

These tools help store, manage, and interact with data—especially large datasets.

- **PostgreSQL** – open-source relational database using SQL.
- **Hadoop** – scalable framework for storing and batch-processing massive datasets.
- **Spark** – fast, in-memory distributed processing engine supporting real-time analytics.
- **Apache Kafka** – streaming platform for data pipelines and real-time analytics.
- **Pandas** – Python library for cleaning, transforming, and analyzing structured data.
- **NumPy** – supports numerical operations, linear algebra, and GPU-based computations.

---

2. Data Visualization Tools

These tools help visually understand data through plots and charts.

- **Matplotlib** – fundamental Python library for customizable visualizations.
- **Seaborn** – higher-level statistical visualization library built on Matplotlib.
- **ggplot2** – popular visualization package in R using layered graphics.
- **Tableau** – interactive business intelligence tool for dashboards.

---

3. Machine Learning Tools

These focus on classical ML algorithms and model building.

- **NumPy** – enables efficient numerical computations.
- **Pandas** – used for data wrangling and preparation.
- **SciPy** – supports scientific computing, optimization, and regression.
- **Scikit-learn** – offers algorithms for classification, regression, clustering, and dimensionality reduction.

---

4. Deep Learning Tools

Tools for building neural networks and large-scale AI models.

- **TensorFlow** – large-scale numerical computation and deep learning framework.
- **Keras** – high-level neural network API easy for beginners.
- **Theano** – supports defining and optimizing mathematical expressions.
- **PyTorch** – flexible and widely used library for research in deep learning and applications like vision and NLP.

---

5. Computer Vision Tools

Used for image and video analysis tasks.

- **OpenCV** – library for real-time image processing and computer vision.

- **Scikit-Image** – algorithms for image filtering, segmentation, and feature extraction.
- **TorchVision** – PyTorch package with datasets, pre-trained CNN models, and transformations.

---

6. Natural Language Processing (NLP) Tools

These tools help computers understand and generate human language.

- **NLTK** – complete toolkit for text processing and tokenization.
- **TextBlob** – simplifies sentiment analysis and POS tagging.
- **Stanza** – powerful NLP library from Stanford offering pre-trained models.

---

7. Generative AI Tools

Used to generate text, images, and other media.

- **Hugging Face Transformers** – large library of transformer-based NLP models.
- **ChatGPT** – advanced language model for text generation and conversational tasks.
- **DALL-E** – tool for generating images from text prompts.
- **PyTorch** – used for developing generative models like GANs and Transformers.