Introduction to Machine Learning Workflow and Model Types

Evan Misshula

June 6, 2025

End to end process

An ML workflow is a sequence of steps to build and deploy a model that solves a problem using data.

The pipeline

Ingestion & Preprocessing	Analysis	Modeling	Deployment
Definition	EDA	Selection	Tuning
Data Collection	Feature Engineering	Training	Deployment
Cleaning		Evaluation	Monitoring

ML Workflow Graph

Define the problem

- What are you trying to do?
- Who is the end user of the prediction?
- What decisions will be based on this output?
- Predict tomorrow's temperature?
- Predict a stock price?
- Is this email spam?

Classify or predict?

Reminder : the dependent variable is what you are trying to predict
The other names are: response, "response variable", "regressand", "criterion", "predicted variable", "measured variable", "explained variable", "experimental variable", "responding variable", "outcome variable", "output variable", "target" or "label".

Classification vs Regression

classification aims to categorize data into distinct groups or classes, while regression involves estimating a continuous value, like a number or a date.

Classification vs Prediction Quiz

- Housing prices?
- If an individual is unhoused?
- Is the email spam?
- Will I get a job?
- Income distribution of programmers?
- Which party will win the Presential Election?

Data Collection

- What do I have and how is it organized?
  - What kind of file is it?
    - CSV, JSON, Pickle, Excel Spreadsheet
  - Does my data exist in a database?
    - Do I need SQL to retrieve it?
  - Is the data available through an API?
    - In general better (more accurate, more frequent) data can be expensive?
    - Do we think we will learn enough to justify the costs? Is the project a demo or is there real money riding on the answer?

Data Collection tools

Elementary libraries

- pandas csv, json and some Relational Database
- requests good for well defined APIs
- openpyxl great way to import data from existing Excel Spreadsheets
- SQL if your data is in a relational database
- BeautifulSoup or Selenium for web scraping (if needed)
- duckdb or sqlite for lightweight DB queries

Big Data Data Collection Tools

Big Data & Distributed Libraries

- PySpark for distributed reading of CSV, JSON, Parquet, Avro, ORC files
- Dask scales pandas-like operations to multi-core or cluster setups
- Apache Kafka for real-time data ingestion from event streams
- HDFS / S3 APIs for direct access to distributed file systems
- Delta Lake / Iceberg transactional layers on big data storage lakes
- SQL Engines: Hive, Presto, Trino, Spark SQL for querying large-scale data

Clean your data

Data is always messier than you are told!

- Be aware of missing values, outliers and duplicates
- Verify your data types

Data Anomaly Definitions

- Missing Values: Observations where data is not recorded or unavailable. Common causes include data entry errors, system glitches, or sensor failures.
- Outliers: Data points that differ significantly from other observations in the dataset. They may indicate variability in measurement, experimental errors, or novel events.
- Duplicates: Records that appear more than once in a dataset but represent the same real-world entity. These can bias results and arise from repeated logging or failed deduplication.

Data Cleaning subtasks

- Convert types (e.g., dates, categorical)
- Don't normalize or scale numeric features (wait until modeling)
- Detect inconsistent labels or typos in categorical data

Never clean data by hand

- Never clean your data by hand. Always use scripts so that your results can be reproduced.
- Documentation is a way to be kind to your future self. The truth is you will never remember why you did what did. Write it down!

Explanation of Isolation Forest

- The isolation forest was introduced by Liu, Ting and Zhou in 2008.

Explanation of Isolation Forest

- The isolation forest was introduced by Liu, Ting and Zhou in 2008.
- Now it's time for some math

Isolation Forest: Mathematical Intuition

Problem Setting

Given a dataset $D = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$, the goal is to assign an anomaly score $s(x) \in [0, 1]$ to each point $x \in D$ based on how easily it can be isolated.

Core Ideas

- Anomalies are rare and different — they are easier to isolate.
- Instead of profiling normal points, we attempt to isolate each point using random partitions.
- The fewer splits needed to isolate a point, the more likely it is to be an anomaly.

Isolation Forest: Tree Construction

Isolation Tree Definition

An isolation tree is a binary tree where each node splits data based on a randomly chosen feature and a randomly chosen split point within that feature's range.

Sampling and Splitting

1. Select a random subsample $D_t \subset D$, of fixed size ψ (typically $\psi = 256$).
2. Recursively partition:
   - Randomly select a feature index $j \in \{1, \ldots, d\}$.
   - Choose a split point $p \sim \text{Uniform}(\min x_j, \max x_j)$ for that feature.
   - Split the data:

$$D_L = \{x \in D_t : x_j < p\}, \quad D_R = \{x \in D_t : x_j \geq p\}$$

   - Recurse on D_L and D_R until:
     - Node contains a single instance, or
     - Tree reaches max depth $\lceil \log_2 \psi \rceil$

Isolation Forest: Scoring Mechanism

Path Length

- For a point x, the path length $h_t(x)$ is the number of edges from the root of the tree to the leaf where x ends up.

Expected Path Length

- The average path length over all trees:

$$E[h(x)] = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$$

Anomaly Score

- The anomaly score is defined as:

Interpretation

Interpreting the Score

- $s(x) \approx 1$: x is likely an outlier (isolated in fewer steps).
- $s(x) \approx 0$: x is likely normal (harder to isolate).
- Use a threshold (e.g., $s(x) > 0.7$) to flag anomalies.

Summary and Use

- Unsupervised: Needs no labeled data
- Fast, interpretable
- Well-suited for high-dimensional data
- Implementation: ‘sklearn.ensemble.IsolationForest‘