Machine learning documentation

Prepared by:

Afnan Zayed Bait Zayed

Machine learning

is a branch of artificial intelligence that relies on data-driven algorithms to build models able to carry out tasks typically requiring human intelligence—such as image classification, data interpretation, and price forecasting. It has become one of the most widely used AI technologies today, forming the backbone of many digital products and services we interact with daily.

Type of Machine learning

- **Supervised learning**: is a ML type that algorithms trained labeled data, which fed the data that includes a label to describe how it should be interpreted. For example, an algorithm may be fed images of flowers that include label for each flower type so that it will be able to identify the flower better again when fed a new flower photo. Also, it used to create ML models that are used for prediction and classification purposes.
- **Unsupervised learning**: is a type of machine learning where algorithms work with unlabeled data to find hidden patterns or groupings without predefined outputs. For example, a company might use unsupervised learning to analyze customer purchase data and automatically segment customers into different groups based on their buying behavior.

Common ML Tasks

- **Classification:** is a supervised learning task where the goal is to assign input data to predefined categories or classes. The model learns from labeled examples to predict the correct class for new, unseen data. For example, Detecting whether an email is spam or not spam.
- **Regression:** is another supervised learning task, but instead of predicting categories, it predicts continuous numeric values. Regression models learn from labeled data with known numerical outcomes. For example, predicting house prices based on features such as size, location, and number of rooms.
- **Clustering:** is an unsupervised learning task that groups similar data points together without predefined labels. The model identifies inherent structures or patterns in the data. For example, Segmenting customers based on their purchasing behavior to target marketing strategies.

Gradient-based learning

Gradient-based learning is a method used in machine learning to improve the performance of AI models, particularly their accuracy. The model starts with initial parameters and makes predictions on the data. It then calculates the error, or loss, between its predictions and the true values. By computing the gradient of this loss, the model adjusts its parameters in the direction that reduces the error. Repeating this process over many iterations allows the model to learn patterns from the data and make more accurate predictions on new, unseen examples. This approach is widely used, especially in training neural networks, to optimize model performance effectively.

The algorithms that are used in classification, regression and clustering

Classification algorithms

- **Logistic Regression:**
  - A simple algorithm that predicts two choices, like yes or no. It learns patterns in the data and gives a probability for each class.
    Example: Predicting if a person is wearing a mask or not.

- **Decision Trees**
  - This algorithm works like a flowchart. It asks a series of questions ("Is age > 20?", "Is the person coughing?") and follows the branches until it reaches a final decision.
    Example: Deciding if a patient is at risk based on symptoms.

- **Random Forest**
  - A random forest is a group of many decision trees working together. Each tree makes a decision, and the forest chooses the answer most trees agree on.
    This makes it more accurate and stable than a single tree.
    Example: Classifying types of plants based on leaf features.

- **Support Vector Machine (SVM)**
  - SVM tries to draw the best line (or boundary) that separates different classes.
    It finds the widest possible gap between categories to make clear decisions.
    Example: Separating images of cats vs. dogs.

- **K-Nearest Neighbors (KNN)**
  - This algorithm looks at the “closest” data points around a new example and assigns the class that most neighbors belong to.
    It’s like asking your nearest friends what they think.
    Example: Classifying a new student’s performance based on similar past students.

- **Neural Networks**
  - Neural networks try to learn like the human brain.
    They use layers of interconnected “neurons” to learn complex patterns and make

decisions.
They are very powerful for images, audio, and text.
Example: Detecting whether an image contains someone wearing a mask or not.

Regression Algorithms

- **Linear Regression**
  - This is the simplest regression algorithm. It draws a straight line through the data to predict a continuous value.
    Example: Predicting house prices based on size.

- **Polynomial Regression**
  - Instead of a straight line, this method fits a curved line to the data. It's useful when the relationship is not linear.
    Example: Predicting temperature changes over time when the trend is curved.

- **Decision Tree Regression**
  - Like classification trees, this regression version makes decisions by splitting data into smaller groups. But instead of predicting a class, it predicts a number.
    Example: Predicting the price of a product based on multiple features.

- **Random Forest Regression**
  - This uses many decision trees together. Each tree predicts a value, and the final answer is the average of all trees. This improves accuracy and reduces errors.
    Example: Predicting a car's resale value.

- **Support Vector Regression (SVR)**
  - This algorithm tries to fit the best line (or curve) within a margin that captures the main trend of the data. It focuses on avoiding errors beyond a certain threshold.
    Example: Predicting stock movement within a certain range.

- **Neural Network Regression**
  - Neural networks can also predict numbers by learning complex patterns. They use multiple layers of "neurons" to understand relationships in the data.
    Example: Predicting energy consumption from many factors like temperature, time, and usage history.

Clustering Algorithms

- **K-Means Clustering**
  - This is the most popular clustering algorithm. It groups data into *K* clusters by finding points that are close to each other. It keeps adjusting the groups until each cluster is as compact as possible.
    Example: Grouping customers into different types based on their shopping habits.

- **Hierarchical Clustering**
    - This algorithm builds clusters step-by-step like a family tree.It can either start with each point alone and merge them (bottom-up) or start with all points together and split them (top-down).

      Example: Grouping similar documents or research papers based on content.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**
    - DBSCAN groups points that are close together and have high density.
      It is great at finding clusters of any shape and identifying noise/outliers.
      Example: Detecting unusual patterns in location data, like identifying areas where people crowd during events.

- **Mean Shift Clustering**
    - This algorithm shifts points toward the area with the highest data density.
      It doesn't require specifying the number of clusters beforehand.
      Example: Image segmentation — dividing an image into meaningful parts based on color or texture.

- **Gaussian Mixture Models (GMM)**
    - GMM assumes the data is made up of several overlapping "Gaussian" distributions (bell-shaped curves). It assigns probabilities of belonging to each cluster instead of hard assignments.
      Example: Classifying customers where some belong partially to multiple groups.

Machine Learning Workflow

- **Problem Framing**
    - The first step is to clearly define the problem you want to solve. This means identifying whether it's a classification task (like predicting categories), regression (predicting numbers), clustering (grouping data), or something else. A well-defined problem ensures that your workflow stays focused and measurable.
- **Data Collection**
    - Next, you gather the data needed to train your model. This could come from APIs, databases, sensors, or files. The quality and quantity of data directly affect your model's performance, so it's important to collect enough relevant examples that represent the problem space.
- **Data Cleaning & Preprocessing**
    - Raw data is rarely ready for machine learning. Preprocessing involves cleaning missing values, removing duplicates, normalizing numerical features, and encoding categorical variables. This step also includes feature engineering, where you create new variables that help the model learn patterns more effectively.
- **Feature Selection**

    - Here, you choose which algorithms might be suitable for your problem. For example, decision trees, random forests, or neural networks. The choice depends on factors like interpretability, accuracy, training speed, and the type of data you have.
- **Splitting**
    - You split your dataset into training and validation sets, then feed the training data into the chosen model. During training, the model learns patterns from the data by adjusting internal parameters. Hyperparameter tuning is often done here to improve performance.
- **Evaluation**
    - Once trained, the model is tested on unseen data (the validation or test set). You measure performance using metrics such as accuracy, precision, recall, F1-score, or RMSE depending on the task. This step helps you understand how well the model generalizes to new data.
- **Deployment**
    - After evaluation, the best-performing model is deployed into a real-world environment. This could mean wrapping it in an API, integrating it into an application, or running it as part of a scheduled pipeline. Deployment ensures the model can provide predictions to end users or systems.
- **Maintenance**
    - Finally, you monitor the model's performance over time. Data distributions can change, causing models to degrade. Regular monitoring, logging, and retraining help maintain accuracy. Maintenance ensures the workflow remains reliable and adapts to new conditions.