Inside the Black Box — Machine Learning

A theoretical overview of machine learning building blocks

As companies of all sizes race to adopt artificial intelligence (AI), understanding the basics of machine learning (ML) is quickly becoming a must-have skill for engineers and anyone working alongside them. Whether you're navigating a data-heavy presentation at work or exploring my *Inside the Black Box* series, understanding ML basics empowers you to engage confidently in tech discussions. In this post, I'll break down common ML terminology and walk you through the different families of machine learning models so you can better understand how these systems work and where each one fits.

Machine learning is a branch of artificial intelligence focused on enabling computers to imitate how humans learn, perform tasks autonomously, and improve their performance and accuracy through experience and exposure to more data[1]. The 'what to watch next' recommendations made to you on YouTube, Netflix, and other streaming platforms are an example of ML in everyday use. Machine learning algorithms are often referred to as 'black boxes.' But what does this mean? In machine learning, when we call something a 'black box,' we mean that the internal process is hidden from us. In this post, I hope to demystify some of the jargon used in the AI and ML world.

AI Ecosystem

Before we dive into the technical details, let's zoom out to visualize how machine learning and other buzzwords fit into the broader AI landscape. The diagram below breaks down AI's subfields and highlights where ML sits among related disciplines, starting from the broad umbrella of AI. Think of this as a roadmap to the ecosystem of AI - a cheat sheet for decoding the jargon you'll encounter in B2B SaaS pitches or engineering meetings.

[1] "What Is Machine Learning (ML)?" *IBM*, IBM, 11 Feb. 2025, www.ibm.com/think/topics/machine-learning.

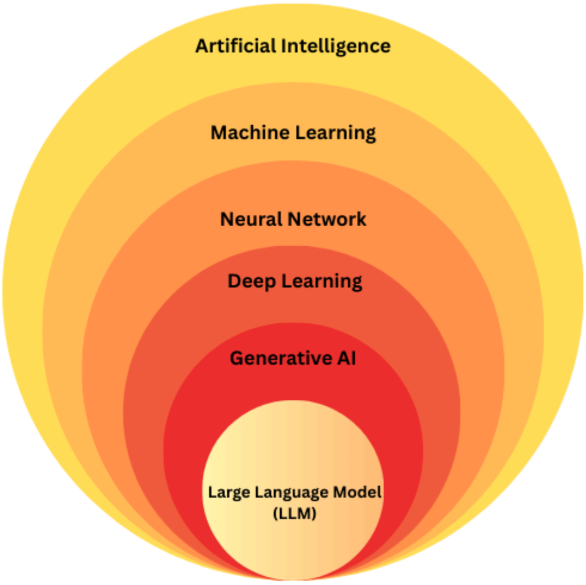

Figure 1

- **Artificial intelligence** is the widest category, and it includes any machine-based system that mimics human intelligence.
- **Machine Learning** is a subset of AI focused on algorithms that learn from data.
- **Neural Networks** are a specific type of ML model inspired by the human brain.
- **Deep Learning** are neural network with many layers, capable of recognizing complex patterns.
- **Generative AI** includes models that can create new content—text, images, code, and more—rather than just make predictions.
- **Large Language Models (LLMs)**, like GPT, specialize in understanding and producing human-like text.

To understand why models sometimes fail or shine, we need to explore the core concepts engineers wrestle with daily: bias, variance, etc. These aren't just academic buzzwords—they're the hidden gears powering your Netflix recommendations, email spam filters, and autocorrect.

ML Terminology

1. Algorithm

An algorithm is a set of instructions designed to accomplish a task [2]. In machine learning, algorithms process data, identify patterns, and transform these insights into outputs, forming the foundation of any model.

[2] "Algorithm." *NNLM*, NNLM, www.nnlm.gov/guides/data-glossary/algorithm. Accessed 15 Apr. 2025.

2. Model

A model is the outcome of an algorithm processing data. It represents the learned patterns or relationships from the data, enabling predictions or decisions when presented with new inputs.

Here's an analogy to understand these terms better: an algorithm is like the blueprint for constructing a building, whereas a model is the completed building created using the blueprint and materials (data). The training process acts as the architect, refining the design to ensure the final structure is stable and functional, just as it turns an algorithm into a well-performing model.

3. Features and Feature Engineering

When discussing regressions, we almost always use the term 'independent variables' to describe predictors. However, when talking about machine learning models, we say *features*. Though these terms mean the same thing (inputs), their usage reflects distinct disciplinary traditions: independent variables originate from statistical frameworks, whereas features emphasize the engineering-driven workflows of machine learning, where raw data is transformed into usable inputs for algorithms[3].

This emphasis on transformation is where *feature engineering* comes into play. Unlike traditional statistical modeling, which often relies on raw variables, feature engineering involves refining or combining data to extract signals that algorithms can interpret more effectively. For example, a timestamp might be split into 'month of the year' features to capture temporal patterns. Other techniques include scaling numerical values, encoding categorical variables, or imputing missing data. In essence, feature engineering bridges the gap between raw data and model-ready inputs - deriving powerful predictors from ordinary, messy data.

4. Labels

[3] Flatley, Dave. "Features vs. Variables." *Statistics.com: Data Science, Analytics & Statistics Courses*, Statistics.com, 7 July 2023, www.statistics.com/02-03-2015-week-5-features-vs-variables/.

Just as independent variables correspond to features, dependent variables correspond to labels in machine learning. Labels, also called targets, represent the outcome or value a model aims to predict.

5. Parameters and Hyperparameters

In machine learning, *parameters* are the internal values a model learns during training, such as how much weight to assign to specific features. The parameters are adjusted automatically as the system studies data to improve its predictions. On the other hand, *hyperparameters* are "settings" chosen by engineers before training begins, such as how quickly the model should learn or how complex its rules can be. Hyperparameters are like the number you dial on a phone, while parameters are the internal components of the phone that handle the actual connection to the carrier and execute the call. Both are critical to building AI systems that work reliably.

6. Training Data and Testing Data

Machine learning models are trained and evaluated on the same dataset, which is typically split into training and testing data. Conventionally, 80% of the original dataset is used for training, and 20 % is used for testing. However, depending on the dataset and the model builder's preference, the split could be 70/30 or 60/40.

7. Bias

Bias refers to errors that lead models to produce unfair, inaccurate, or prejudiced outcomes. In machine learning, bias arises when a model relies too heavily on simplified assumptions derived from flawed training data. This often leads to *underfitting*, where the model is too simplistic to capture meaningful patterns in the data, resulting in inaccurate predictions for both the training set and unseen data. For example, using a two-dimensional model to fit a 15-dimensional dataset will most likely cause underfitting, as the model is incapable of understanding the complexity inherent in the data. Like this:

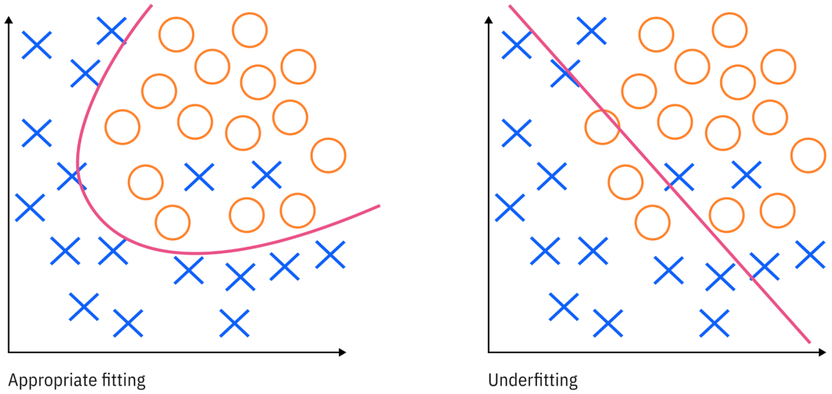

Figure 2[4]

Bias can arise from:

a. Bad data: Using incomplete or skewed training data. Example: training a facial recognition system mostly on light-skinned faces, it will struggle with darker skin tones.
b. Overly Simplistic Models: Using models that lack the complexity required to match the dataset's intricacy can result in underfitting. Example: A linear regression model applied to a highly non-linear dataset may miss significant relationships between variables.
c. Feedback Loops in Training: Models trained on biased outputs can create self-reinforcing cycles if errors go unpenalized. Example: a house price prediction model mistakenly correlates ZIP codes with race, perpetuating discriminatory patterns unless an engineer intervenes to correct the bias.

Machine learning systems don't just reflect existing biases—they often amplify them. Without intervention, biased training data can lead to models that reinforce or exacerbate societal inequalities. Addressing bias is essential for building models that are both fair and effective. Techniques to mitigate bias include using diverse and well-balanced datasets, fine-tuning models to address imbalances, and choosing algorithms that better capture the complexity of the problem.

[4] IBM. "What Is Overfitting vs. Underfitting?" *IBM*, 12 Dec. 2024, www.ibm.com/think/topics/overfitting-vs-underfitting.

8. Variance

Variance is most commonly understood as a measure of how far individual observations deviate from the average in a data set[5]. In machine learning, this concept evolves to describe a model's sensitivity to variations in its training data. A high-variance model reacts too strongly to minor fluctuations in data, causing its predictions to shift dramatically with small changes in input features.

When a model suffers from high variance, it struggles to identify general trends and becomes overly attuned to specific details in the training data. This leads to *overfitting*, where the model performs exceptionally well on the training data but poorly on new, unseen data. Overfitting typically occurs when a highly complex model is applied to a relatively simpler dataset, resulting in less reliable predictions. Here is a visual example of overfitting:

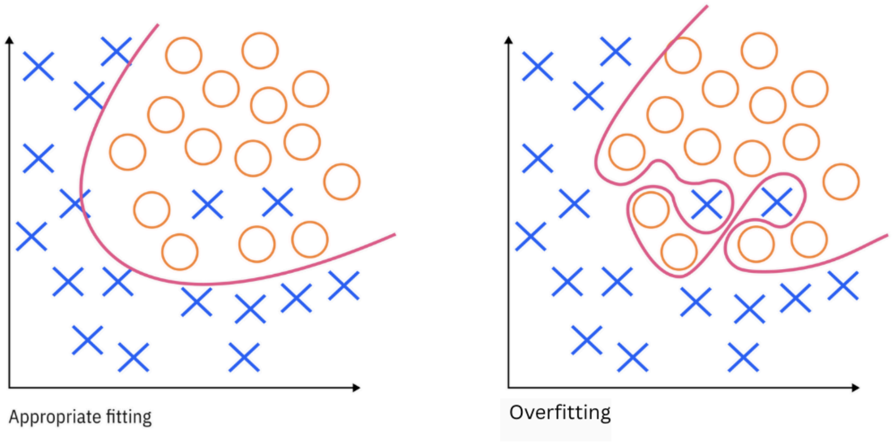

Figure 3[6]

Variance can arise from:

a. Complex Models: Overly intricate models, such as deep neural networks with excessive layers, can memorize the training data rather than identify broader trends.
b. Limited Training Data: With insufficient or highly variable data, a model may overfit by clinging to specific training examples rather than learning the underlying patterns.

[5] Hayes, Adam. "What Is Variance in Statistics? Definition, Formula, and Example." *Investopedia*, Investopedia, www.investopedia.com/terms/v/variance.asp. Accessed 15 Apr. 2025.
[6] IBM. "What Is Overfitting vs. Underfitting?" *IBM*, 12 Dec. 2024, www.ibm.com/think/topics/overfitting-vs-underfitting.

c. Feature Overload: Including too many irrelevant or highly correlated features can increase the chances of the model overfitting to noise in the data.

High variance doesn’t just affect a model’s accuracy; it undermines its ability to generalize trends, making it less effective in real-world scenarios. Managing variance is essential to avoid overfitting and to build models that balance flexibility and reliability.

9. Bias-Variance Tradeoff

While bias stems from oversimplifying patterns in data, variance arises from overcomplicating them. Striking the right balance between bias and variance is crucial for creating models that perform well on training and unseen data - a concept known as the *bias-variance tradeoff.*

How can we identify whether a model is biased or has high variance? Here's a guideline:

	Bias errors	**Variance errors**
Training	High	Low
Testing	High	High

- If a model shows high errors in training and testing data, it indicates underfitting due to excessive bias.
- If a model performs well on training data but poorly on testing data, it points to overfitting caused by high variance.

The bias-variance tradeoff goal is to minimize the combined effect of bias and variance, ensuring the model captures meaningful patterns while avoiding noise. Metrics like Mean Squared Error (MSE) help quantify this balance by penalizing large errors and variability. The lower the MSE score, the more harmonious the balance between bias and variance. One method of minimizing bias and variance is by scaling datasets.

10. Normalization vs. Standardization

Scaling datasets after they've been split into training and testing almost always improves the model's understanding of relationships between features and the label. Scaling ensures that all features contribute equally to the model's learning process, preventing bias toward inputs with a bigger range of values. There are two popular scaling techniques: *normalization,* which is used when the algorithm employed assumes that the distribution of values is normal, and *standardization,* which is used when the algorithm requires a fixed range of values. Let's break these differences down further:

	Normalization	**Standardization**
Approach	This technique scales data to fit a range of values, usually between 0 and 1	This technique assumes a normal distribution, so the data is scaled to have a mean of 0 and a standard deviation of 1
Formula	Uses minimum and maximum values	Uses mean and standard deviation
Range of Values	Specified fixed range	No fixed range
Sensitivity to Outliers	This approach is more sensitive to outliers because outliers can directly influence max/ min values, thereby squishing valuable data points in between.	This approach is not very sensitive to outliers because it scales data using mean and standard deviation. The mean accounts are calculated using the outliers, and so the standard deviation provides a measure of spread that is less volatile.
Use Case	Algorithms such as k-NN requiring a fixed range	Algorithms such as SVMs assume a normal distribution

So far, we've laid the groundwork by discussing algorithms, models, and the importance of features and feature engineering in representing data effectively. We've also explored labels, the role of training and testing data, and techniques like standardization and normalization to prepare data for analysis. Additionally, we've examined key concepts like bias, variance, and the bias-variance tradeoff to evaluate model performance. With this foundation in place, it's time to shift our focus to the models themselves.

ML Families

To truly understand machine learning models, we need to categorize them into distinct families. By grouping models by how they learn from data, we uncover the types of problems each family is best suited to solve and gain insights into their underlying structures. Let's dive into the mindmap below to explore these families in detail.

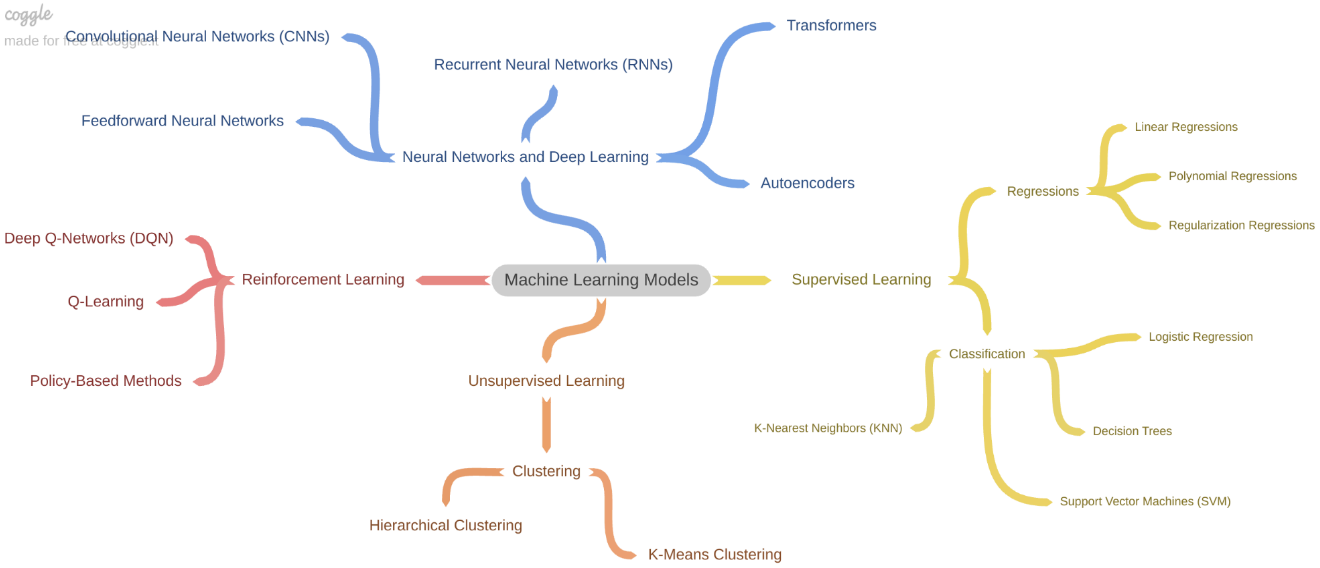

Figure 4

11. Supervised Learning

If you've been following the *In the Black Box* series, you've already encountered models like linear regression and logistic regression, both of which are prime examples of supervised learning. In supervised learning, models uncover patterns and relationships between features and target data (output labels) by learning from labeled datasets. These labels guide the model during training, enabling it to predict outcomes accurately.

Supervised learning models can be categorized based on the structure of their outputs:

- **Regression Models:** Linear regression, polynomial regression, and regularization techniques are tailored for datasets where the target variable is continuous. For instance, predicting house prices—a numerical value—would require a regression model. However, it’s worth noting that not all models with "regression" in their name fall into this category; for example, logistic regression is actually a classification model.

- **Classification Models:** These models excel with datasets where the target variable is discrete. Their objective is to assign observations to specific categories or predict the probability of an event occurring. For example, when labels are binary, classifiers like logistic regression, decision trees, and k-nearest neighbors (kNN) are employed to predict ‘1’ or ‘0’ outcomes. Whether it’s determining if an email is spam or predicting customer churn, supervised learning classifiers shine in scenarios requiring categorical predictions.

12. Unsupervised Learning

Sometimes, we need to uncover patterns or make predictions from datasets without labels—this is where unsupervised learning models become essential. Unsupervised learning is an approach where models uncover patterns in datasets without predefined outputs. Instead of predicting outcomes, these models aim to group, structure, and interpret the inherent relationships within the data. This process is known as *clustering*.

Clustering algorithms segment data into distinct groups based on similarities among features. They are particularly useful when the goal is to discover hidden patterns or natural groupings within data. For instance, k-means and hierarchical clustering can segment customers into demographic-based groups for targeted marketing or to identify genetic similarities in biological research. Clustering is a foundational tool for understanding and organizing unstructured data.

Let’s recap what we’ve learned about supervised and unsupervised learning by comparing the two:

	Supervised Learning	**Unsupervised Learning**
Dataset type	Labeled	Unlabeled
Goal	Prediction/ Classification	Clustering

Algorithms	Decision trees, Linear regressions, Neural Networks	k-Nearest Neighbors, hierarchical clustering
Use Cases	Sales forecasting, attrition forecast, image recognition	Customer segmentation, anomaly detection

Model Evaluation

When working with machine learning models or products, it is critical to assess a model. Conventionally, here's how models are evaluated and how to judge their scores for yourself:

a. Regression Models

i. Mean Absolute Error (MAE) tells you how far the model's predictions are from the actual values on average. The lower the score, the better.

ii. Mean Squared Error (MSE) calculates the average of the difference between predicted and actual values squared, which helps MSE highlight significant mistakes in predictions. The lower the score, the better.

iii. To convert the MSE into the same units as the target variable, we take the root of the score and call it the Root Mean Squared Error. The RMSE gives us the average magnitude of the error. The lower the score, the better.

iv. The R-squared score tells us how well the model can explain variance in the target variable. Also known as the coefficient of determination, the closer an R^2 value is to 1, the better the model.

b. Classification Models

i. Accuracy measures the proportion of correct predictions out of all predictions made. The higher the score, the better.

ii. Precision indicates how many of the predictions labeled as ‘yes’ or ‘1’ are correct. It is useful when tweaking models to minimize false positives, like for a model predicting pregnancy based on symptoms. The higher the score, the better.

iii. Recall measures how many actual positive cases were correctly identified by the model. It is essential when minimizing false negatives, like for a model predicting fatal diseases at hospitals. The higher the score, the better.

iv. F1-Score is the mean of precision and recall, balancing their trade-offs. It is useful when a single metric is required to evaluate both false positives and false negatives. The higher the score, the better.

c. Clustering Models

i. Silhouette Score measures how well-separated and cohesive the clusters are. A higher score (range: -1 to 1) indicates well-defined and distinct clusters.

ii. The Dunn Index evaluates clustering quality by comparing the minimum distance between clusters to the maximum size of a cluster. A higher value suggests better compactness within clusters and greater separation between them.

Conclusion

As artificial intelligence continues to shape the way we work, live, and interact with the world, understanding the fundamentals of machine learning is more critical than ever. From optimizing workflows to crafting predictive models, ML equips us with tools to solve problems and uncover opportunities that were once out of reach.

By understanding concepts like bias, variance, and feature engineering, you're building the skills to evaluate and design intelligent systems, equipping yourself to contribute to innovation in AI-driven industries. Whether you're an aspiring engineer or a business professional deciphering model outputs, this knowledge empowers you to engage confidently in conversations about your models.

Stay tuned as we unlock our next black box: Decision Trees. Discover how this intuitive yet powerful algorithm mimics human decision-making to solve complex problems.