**Semi-Supervised Learning in ML**

Semi-supervised learning is a type of machine learning that falls in between supervised and unsupervised learning. It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model. The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data.

**Examples of Semi-Supervised Learning**

* **Text classification**: In text classification, the goal is to classify a given text into one or more predefined categories. Semi-supervised learning can be used to train a text classification model using a small amount of labeled data and a large amount of unlabeled text data.
* **Image classification**: In image classification, the goal is to classify a given image into one or more predefined categories. Semi-supervised learning can be used to train an image classification model using a small amount of labeled data and a large amount of unlabeled image data.
* **Anomaly** **detection**: In anomaly detection, the goal is to detect patterns or observations that are unusual or different from the norm

**What is Feature Engineering?**

Feature engineering is the process of **transforming raw data into features that are suitable for machine learning models**. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.

1. **Improves Model Performance:**By creating new and more relevant features, the model can learn more meaningful patterns in the data.
2. **Reduces Overfitting:** By reducing the dimensionality of the data, the model is less likely to overfit the training data.
3. **Increases Model Robustness:**Transforming the features can make the model more robust to outliers and other anomalies.
4. **Improves Computational Efficiency:**The transformed features often require fewer computational resources.
5. **Increases Model Flexibility:** By adding new features, the model can be made more flexible to handle different types of data.