



# **Artificial Intelligence (Machine Learning & Deep Learning) [Course]**

**Week 4 – Machine Learning Core - Foundation  
[See examples / code in GitHub code repository]**

**It is not about Theory, it is 20% Theory and 80% Practical –  
Technical/Development/Programming [Mostly Python based]**

# Machine Learning - Meaning

Machine learning (ML) is a branch of artificial intelligence (AI) focused on enabling computers and machines to imitate the way that humans learn, to perform tasks autonomously, and to improve their performance and accuracy through experience and exposure to more data.

**Machine learning (ML) is a discipline of artificial intelligence (AI) that provides machines with the ability to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention.**

Machine learning methods enable computers to operate autonomously without explicit programming. ML applications are fed with new data, and they can independently learn, grow, develop, and adapt.

## Examples

25

### Reference:

<https://www.datacamp.com/blog/what-is-machine-learning>

<https://www.ibm.com/think/topics/machine-learning>

<https://www.coursera.org/articles/what-is-machine-learning>



# Machine Learning – Workflow - 1

1



## Project setup

### 1. Understand the business goals

Speak with your stakeholders and deeply understand the business goal behind the model being proposed. A deep understanding of your business goals will help you scope the necessary technical solution, data sources to be collected, how to evaluate model performance, and more.

### 2. Choose the solution to your problem

Once you have a deep understanding of your problem—focus on which category of models drives the highest impact. See this [Machine Learning Cheat Sheet](#) for more information.

2



## Data preparation

### 1. Data collection

Collect all the data you need for your models, whether from your own organization, public or paid sources.

### 2. Data cleaning

Turn the messy raw data into clean, tidy data ready for analysis. Check out this [data cleaning checklist](#) for a primer on data cleaning.

### 3. Split the data

Randomly divide the records in the dataset into a training set and a testing set. For a more reliable assessment of model performance, generate multiple training and testing sets using cross-validation.

### 4. Feature engineering

Manipulate the datasets to create variables (features) that improve your model's prediction accuracy. Create the same features in both the training set and the testing set.

Reference:

<https://www.datacamp.com/blog/a-beginner-s-guide-to-the-machine-learning-workflow>



# Machine Learning – Workflow - 2

3



## Modeling

### 1. Hyperparameter tuning

For each model, use hyperparameter tuning techniques to improve model performance.

### 2. Train your models

Fit each model to the training set.

### 4. Assess model performance

For each model, calculate performance metrics on the testing set such as accuracy, recall and precision.

### 3. Make predictions

Make predictions on the testing set.

4



## Deployment

### 1. Deploy the model

Embed the model you chose in dashboards, applications, or wherever you need it.

### 2. Monitor model performance

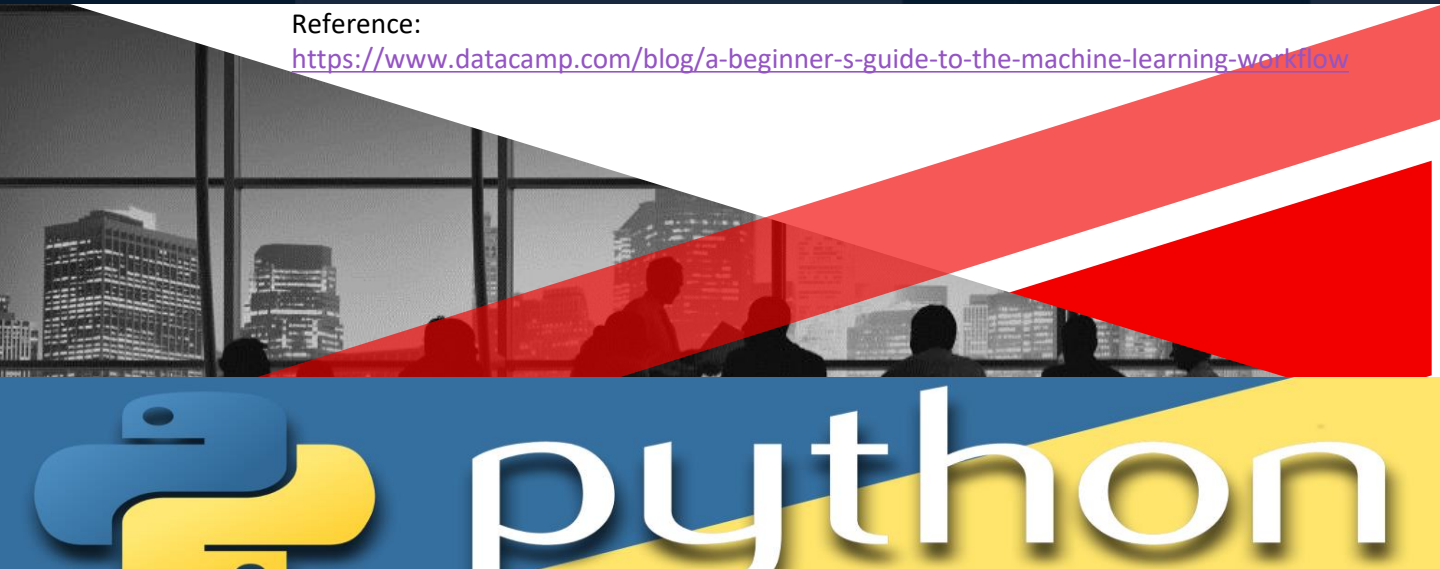
Regularly test the performance of your model as your data changes to avoid model drift.

### 3. Improve your model

Continuously iterate and improve your model post-deployment. Replace your model with an updated version to improve performance.

Reference:

<https://www.datacamp.com/blog/a-beginner-s-guide-to-the-machine-learning-workflow>



# Feature Engineering

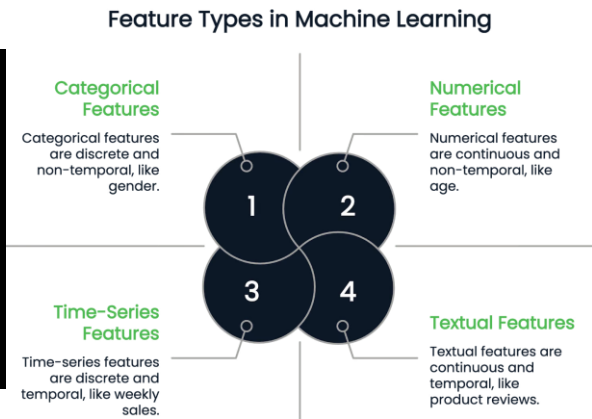
Feature engineering is the process of turning raw data into useful features that help improve the performance of machine learning models. It includes choosing, creating and adjusting data attributes to make the model's predictions more accurate. The goal is to make the model better by providing relevant and easy-to-understand information.

## Feature Engineering Techniques

### 1. Feature scaling

Feature scaling ensures that numerical features lie within a standardized range, preventing some features from dominating the learning process due to their larger values. Machine learning models that rely on distance-based calculations (e.g., linear regression, k-nearest neighbors, and neural networks) can be affected when features have vastly different scales.

<https://www.datacamp.com/tutorial/normalization-vs-standardization>



### 2. Encoding categorical variables

Machine learning models cannot directly process categorical variables, so they must be converted into numerical representations. Below, we discuss some popular encoding techniques.

- ❑ One-hot encoding:
- ❑ Label encoding:
- ❑ Ordinal encoding:
- ❑ Target encoding:

### Tools and Libraries for Feature Engineering

- ❑ Pandas
- ❑ Scikit-Learn
- ❑ Feature-Engine

### Automated feature engineering tools

- ❑ Featuretools:
- ❑ TSMF:
- ❑ AutoFeat:

### Reference:

<https://www.ibm.com/think/topics/feature-engineering>

<https://www.datacamp.com/tutorial/feature-engineering>

<https://www.geeksforgeeks.org/machine-learning/what-is-feature-engineering>

<https://medium.com/kgxperience/feature-engineering-for-machine-learning-a-step-by-step-guide-part-1-33b52b>



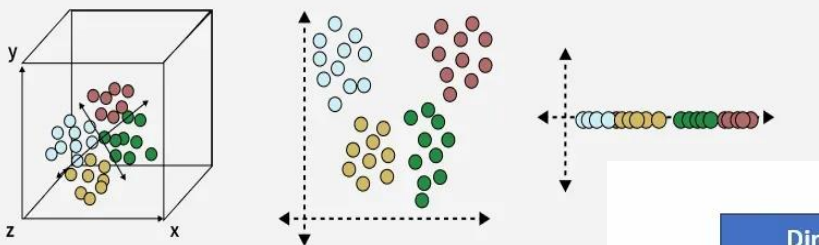


# Dimensionality Reduction

When working with machine learning models, datasets with too many features can cause issues like slow computation and overfitting. Dimensionality reduction helps to reduce the number of features while retaining key information. Techniques like principal component analysis (PCA), singular value decomposition (SVD) and linear discriminant analysis (LDA) convert data into a lower-dimensional space while preserving important details.

## What is Dimensionality Reduction

**Dimensionality Reduction** is the process of reducing the number of input variables (features) in a dataset while preserving as much important information as possible.



### Dimensionality Reduction Techniques

#### Feature selection

- Filter method
- Wrapper method
- Embedded method

#### Feature extraction

- Principal Component Analysis
- Missing Value Ratio
- Backward Feature Selection
- Forward Feature Selection
- Factor Analysis
- Independent Component Analysis

## Advantages and Disadvantages of Dimensionality Reduction

### Advantages

- ✓ Simplify the pattern representation and the classifiers
- ✓ Faster classifier with less memory consumption
- ✓ Alleviate curse of dimensionality with limited data sample



### Disadvantages

- ✗ Loss information
- ✗ Increased error in the resulting recognition system



25

### Reference:

<https://www.geeksforgeeks.org/machine-learning/dimensionality-reduction/>

<https://learninglabbb.com/dimensionality-reduction-in-machine-learning/>

<https://www.ibm.com/think/topics/dimensionality-reduction>

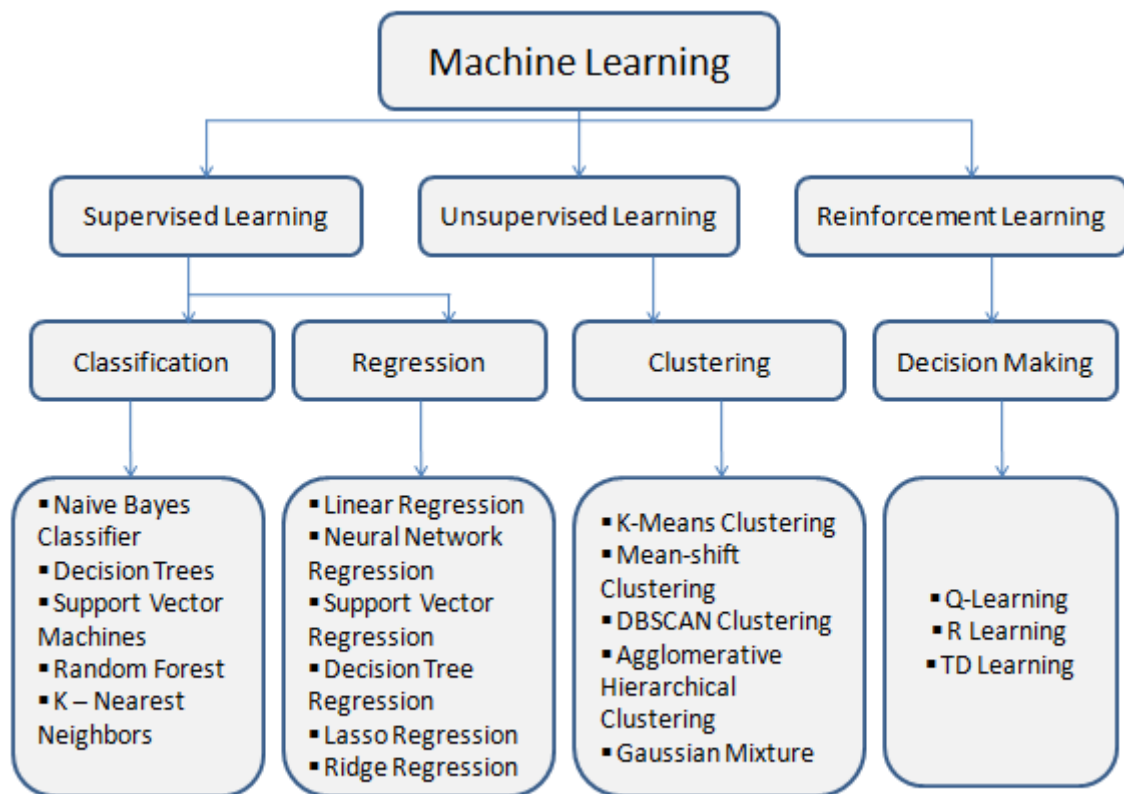
<https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/>

<https://learninglabbb.com/dimensionality-reduction-in-machine-learning/>



# python

# Machine Learning - Types



<https://www.geeksforgeeks.org/types-of-machine-learning/>

<https://www.datacamp.com/blog/what-is-machine-learning>

<https://www.ibm.com/think/topics/machine-learning>

<https://www.coursera.org/articles/what-is-machine-learning>



# Important Term



## References:

<https://www.geeksforgeeks.org/50-machine-learning-terms-explained/>  
[https://www.w3schools.com/ai/ai\\_ml\\_terminology.asp](https://www.w3schools.com/ai/ai_ml_terminology.asp)  
<https://developers.google.com/machine-learning/glossary>  
<https://www.springboard.com/blog/data-science/machine-learning-terminology/>





# Machine Learning Cheat Sheet



Reference:

25

<https://www.datacamp.com/cheat-sheet/machine-learning-cheat-sheet>

<https://www.datacamp.com/cheat-sheet/category/machine-learning>

<https://www.geeksforgeeks.org/machine-learning-algorithms-cheat-sheet/>

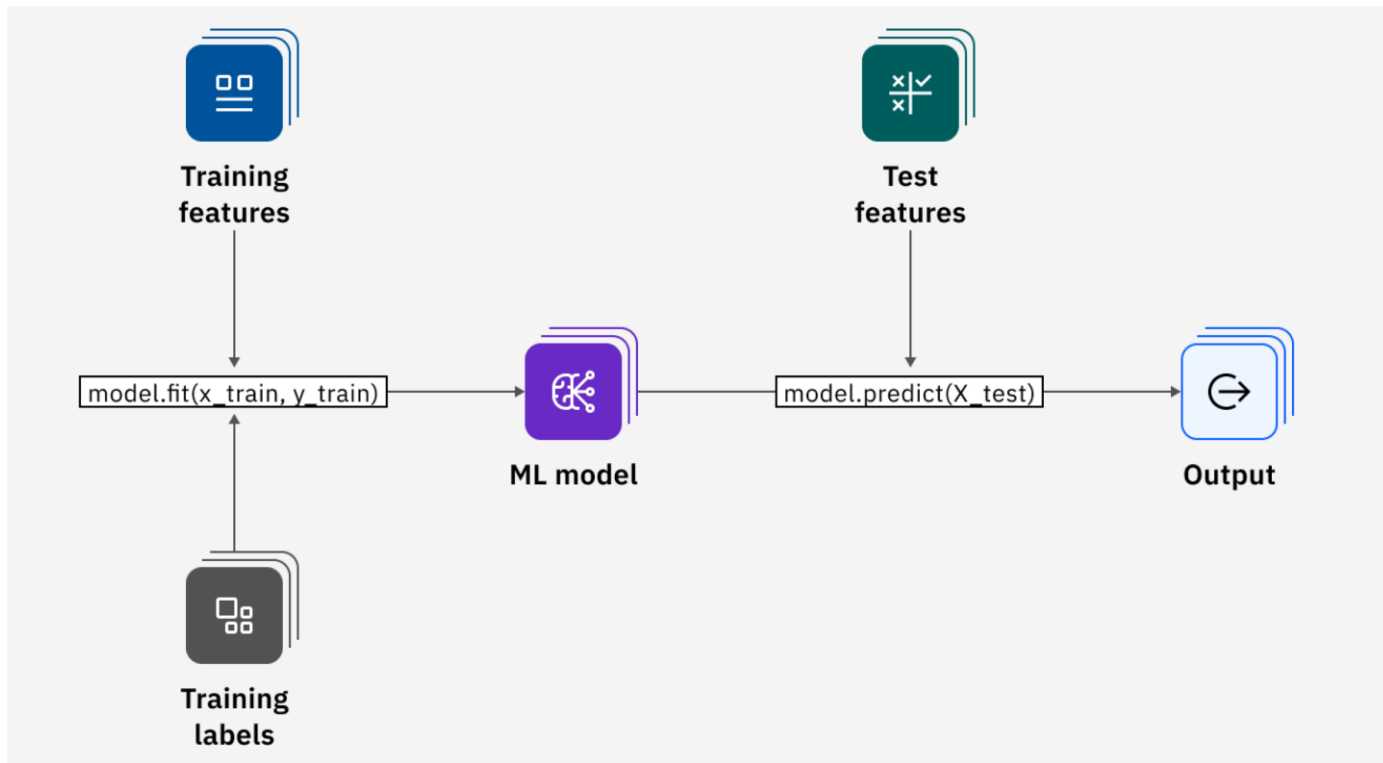
[https://www.tutorialspoint.com/machine\\_learning/machine\\_learning\\_cheatsheet.htm](https://www.tutorialspoint.com/machine_learning/machine_learning_cheatsheet.htm)

<https://datasciencedojo.com/blog/machine-learning-algorithms/>



# scikit-learn Overview

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license



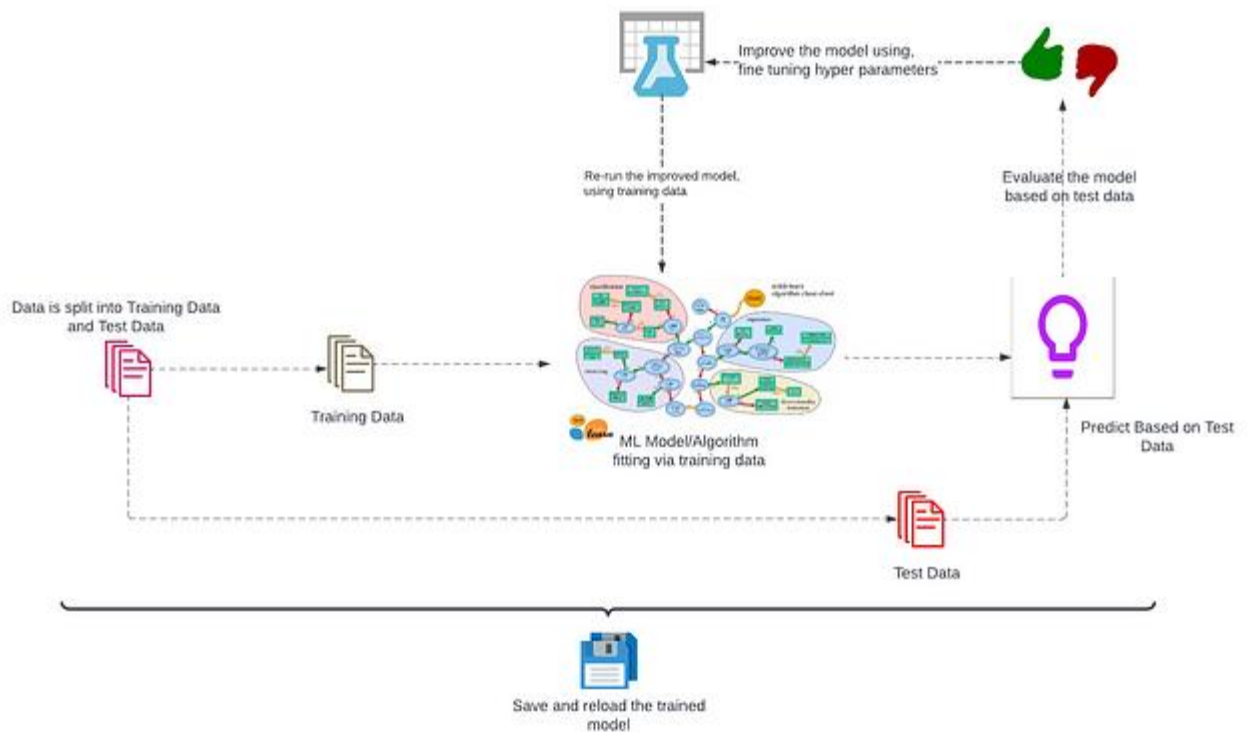
Reference:

<https://www.ibm.com/think/topics/scikit-learn>

<https://scikit-learn.org/>



# scikit-learn - Workflow



25

Reference:

<https://medium.com/codenx/machine-learning-using-scikit-learn-sklearn-high-level-workflow-and-preparing-the-data-ecb3b6af4425>

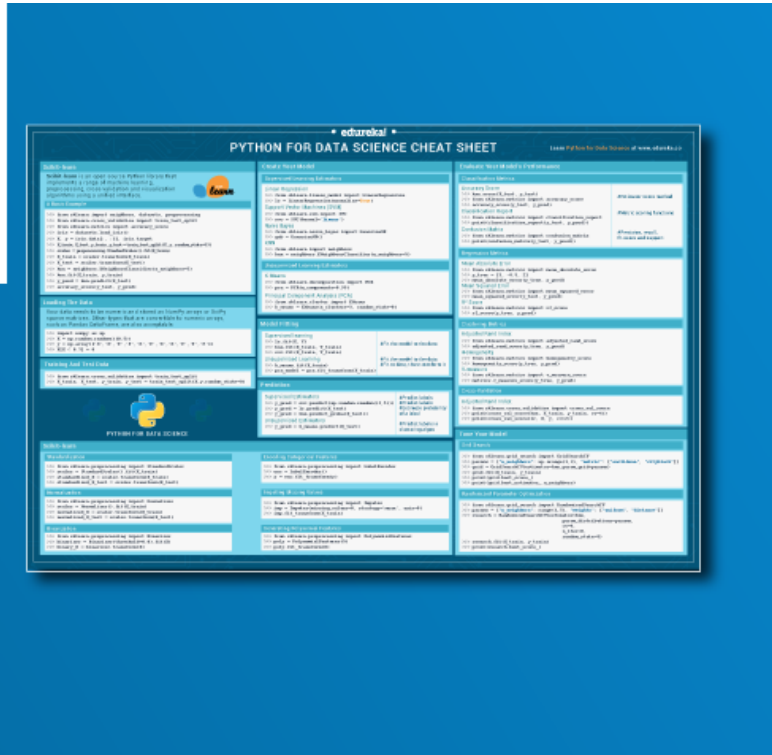




# scikit-learn – cheat sheet



## Python Scikit Cheat Sheet



### Reference:

<https://intellipaat.com/blog/tutorial/python-tutorial/scikit-learn-cheat-sheet/>  
<https://www.datacamp.com/cheat-sheet/scikit-learn-cheat-sheet-python-machine-learning>  
<https://www.geeksforgeeks.org/scikit-learn-cheatsheet/>  
[https://scikit-learn.org/stable/machine\\_learning\\_map.html](https://scikit-learn.org/stable/machine_learning_map.html)  
[https://s3.amazonaws.com/assets.datacamp.com/blog\\_assets/Scikit\\_Learn\\_Cheat\\_Sheet\\_Python.pdf](https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Scikit_Learn_Cheat_Sheet_Python.pdf)

25



# Parameters/Hyperparameter

A model parameter is a configuration variable that is internal to the model and whose value can be estimated from the given data.

They are required by the model when making predictions.

Their values define the skill of the model on your problem.

They are estimated or learned from data.

They are often not set manually by the practitioner.

They are often saved as part of the learned model.

Some examples of model parameters include:

- The weights in an artificial neural network.
- The coefficients in a linear regression or logistic regression.

## What is a Hyperparameter in a Machine Learning Model?

A model hyperparameter is a configuration that is external to the model and whose value cannot be estimated from data.

- They are often used in processes to help estimate model parameters.
- They are often specified by the practitioner.
- They can often be set using heuristics.
- They are often tuned for a given predictive modeling problem

Some examples of model hyperparameters include:

- The learning rate for training a neural network.
- The C and sigma hyperparameters for support vector machines.
- The k in k-nearest neighbors.

### Reference:

<https://www.datacamp.com/tutorial/parameter-optimization-machine-learning-models>





# ML Foundation

## Difference between training data and testing data

<https://www.geeksforgeeks.org/training-data-vs-testing-data/>

<https://testsigma.com/blog/difference-between-training-data-and-testing-data/>

<https://www.zams.com/blog/the-difference-between-training-data-vs-test-data-in-machine-learning>

## Machine learning feature engineering

<https://www.ibm.com/think/topics/feature-engineering>

<https://www.geeksforgeeks.org/what-is-feature-engineering/>

<https://www.analyticsvidhya.com/blog/2021/10/a-beginners-guide-to-feature-engineering-everything-you-need-to-know/>

## Parameters and Hyperparameters in Machine Learning and Deep Learning

<https://www.geeksforgeeks.org/difference-between-model-parameters-vs-hyperparameters/>

<https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac/>

<https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/>

<https://www.datacamp.com/tutorial/parameter-optimization-machine-learning-models>





Thank you - for listening and participating

- ☐ Questions / Queries
- ☐ Suggestions/Recommendation
- ☐ Ideas.....?

Shahzad Sarwar  
Cognitive Convergence

<https://cognitiveconvergence.com>  
[shahzad@cognitiveconvergence.com](mailto:shahzad@cognitiveconvergence.com)

voice: +1 4242530744 (USA) +92-3004762901 (Pak)