Great Place To Work. Certified MAR 2022 - MAR 2023 INDIA

UPES

UNIVERSITY OF TOMORROW

nirf Ranking 2022

Ranked 65th University in India

ACCREDITED WITH GRADE A NAAC

Accredited **Grade** 'A' by NAAC

QS STARS™ RATING SYSTEM 2020

QS 5 Star Rating for Academic Development, Employability, Facilities and Program Strength

E-LEAD

Perfect score of **150/150** as a testament to exceptional E-Learning methods

Unit 3 : Introduction to machine learning

UPES
UNIVERSITY OF TOMORROW

Table of Contents

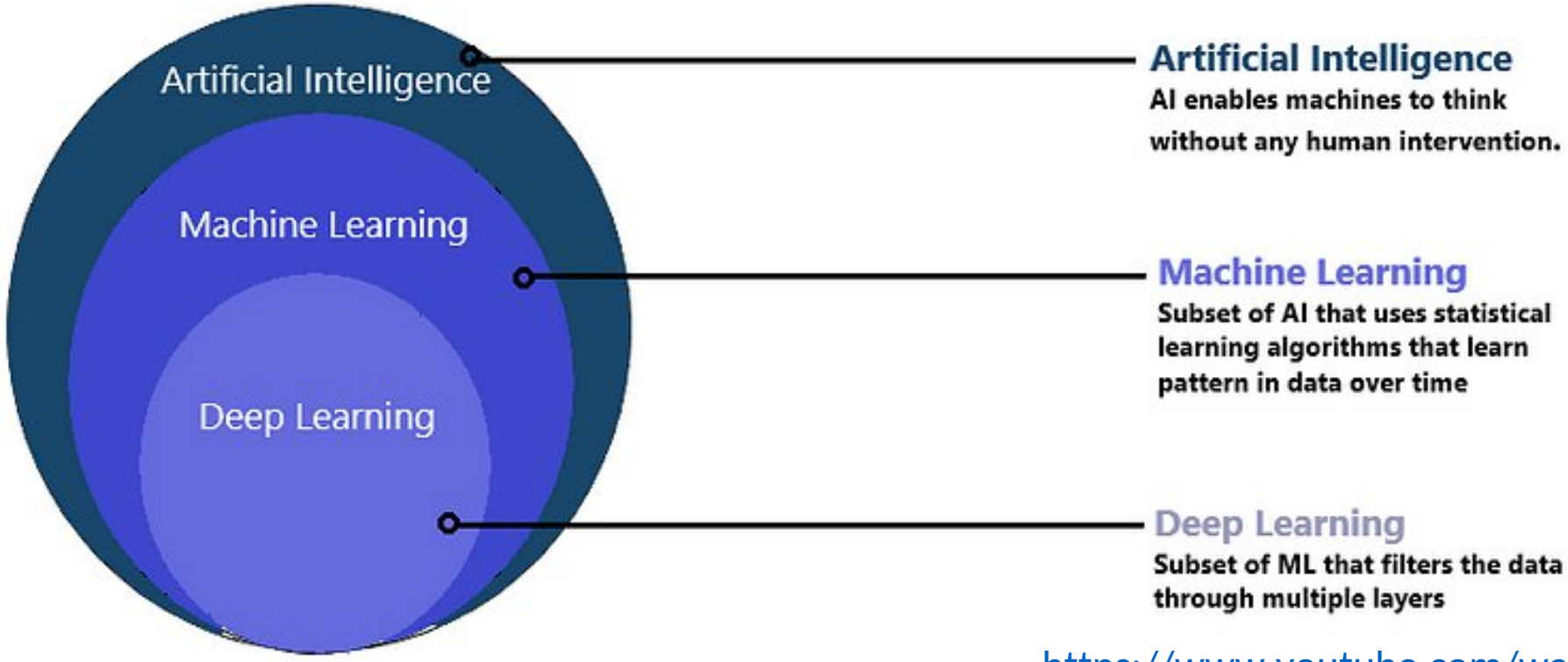
Unit III: Introduction to machine learning

- Introduction to Machine Learning
- Usage of datasets and how to handle them for Machine Learning
- Feature sets
- Dataset division: test, train and validation sets, cross validation,
- Dimensionality Reduction Techniques: PCA, LDA, ICA

UPES
UNIVERSITY OF TOMORROW

Introduction to Machine Learning

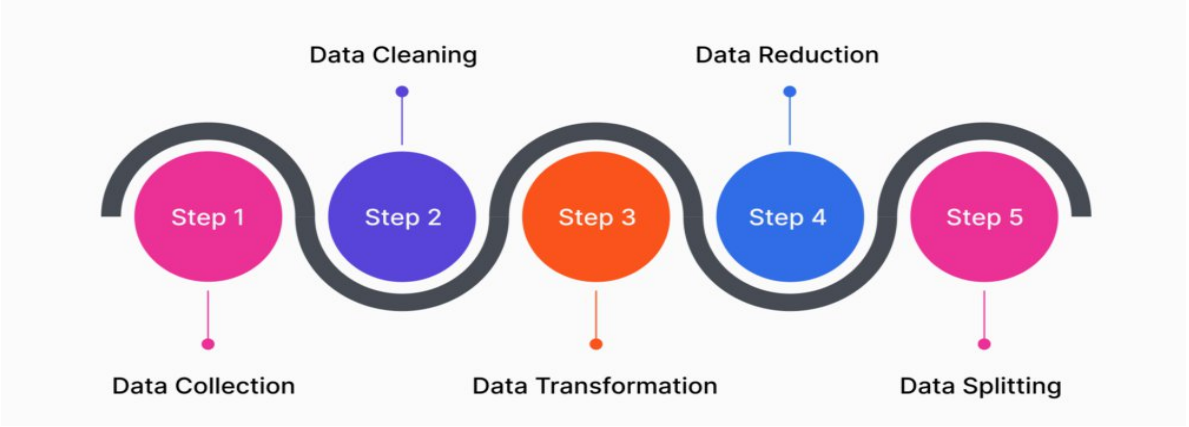
Machine learning is a branch of artificial intelligence (AI) that focuses on developing algorithms and techniques that enable computers to learn from and make predictions or decisions based on data without being explicitly programmed. The primary goal of machine learning is to allow computers to automatically learn and improve from experience.

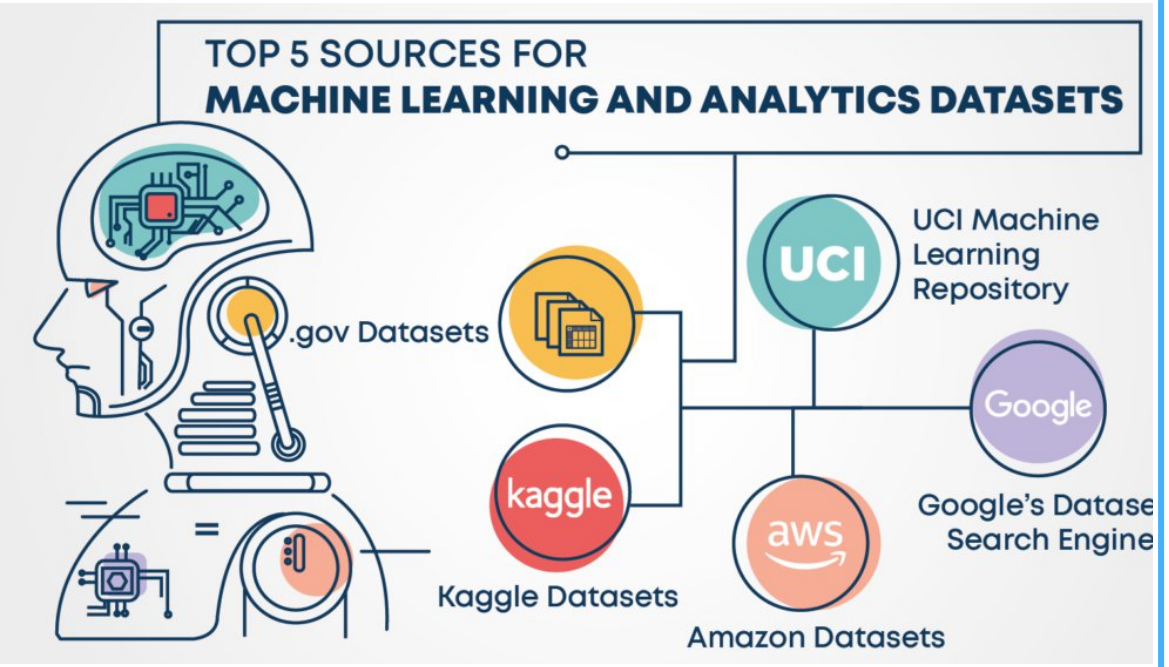


https://www.youtube.com/watch?v=ukzFI9rgwfU

Key Concepts in Machine Learning

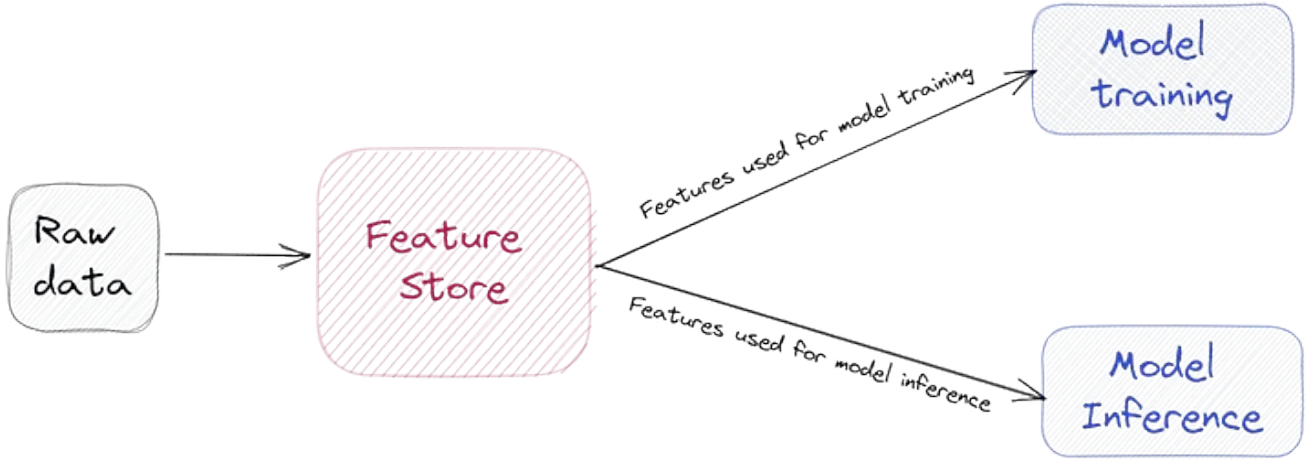
- **Data:** Machine learning algorithms require data to learn from. This data can come in various forms, such as text, images, audio, or numerical values.





Key Concepts in Machine Learning

- **Features:** Features are individual measurable properties or characteristics of the data. For example, in an image classification task, features could be pixel values, color histograms, or texture features.

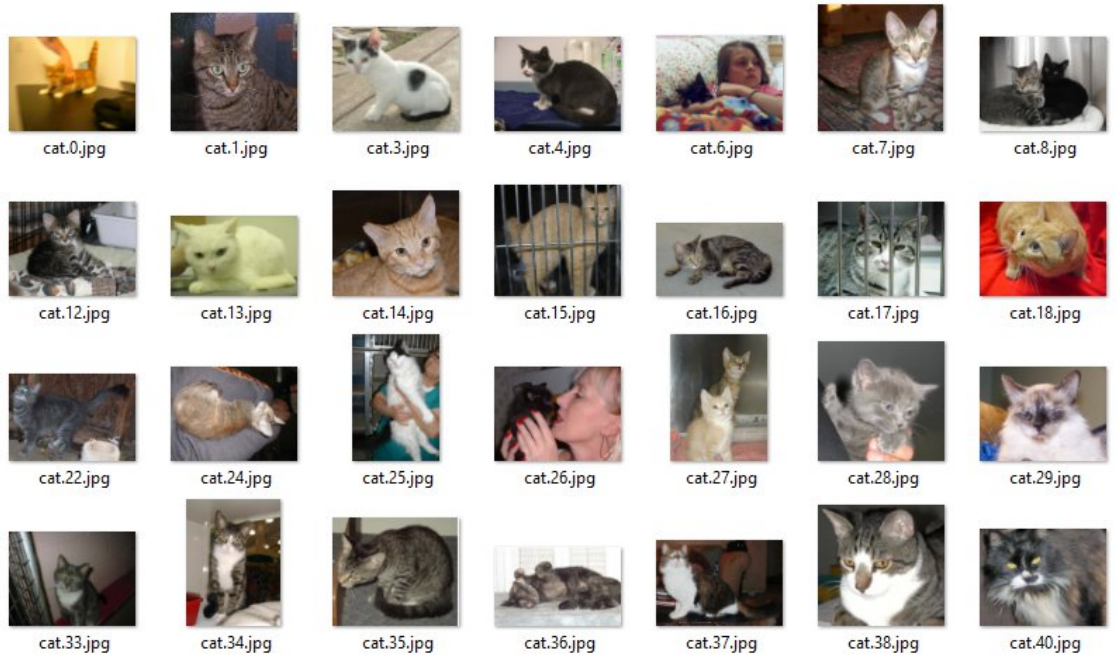


Feature

Dates	Store ID	Product ID	Product name	# of Sales
02-12-2021	001	RP01	Almond milk	21
10-12-2021	005	RS21	Oat milk	15
18-01-2022	004	RK32	Hazelnut milk	9

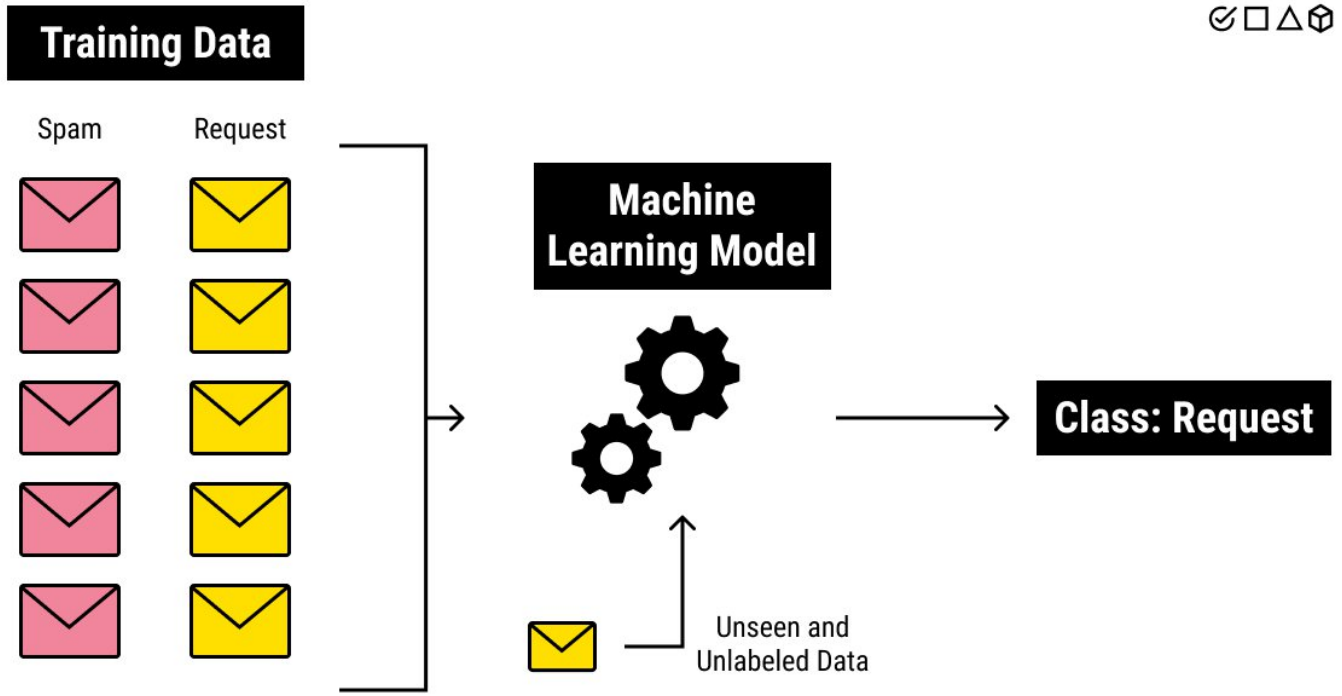
Key Concepts in Machine Learning

- **Labels or Targets:** In supervised learning, the algorithm learns from a dataset that includes both input data and corresponding output labels or targets. The algorithm learns to map input data to the correct output labels.



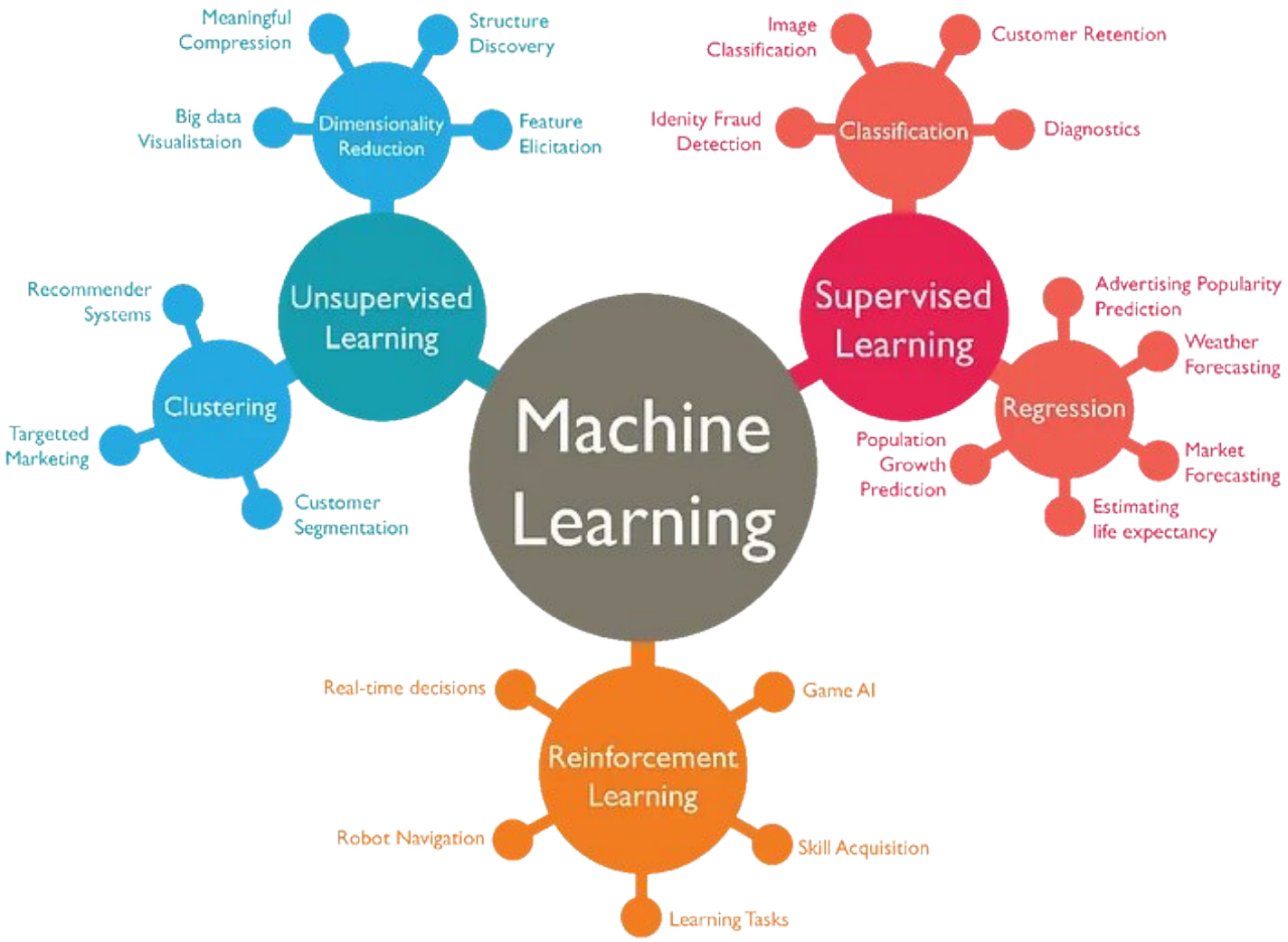
Key Concepts in Machine Learning

- **Training Data:** This is the portion of the dataset used to train the machine learning model. The model learns patterns and relationships from this data.
- **Testing Data:** This is a separate portion of the dataset used to evaluate the performance of the trained model. It helps assess how well the model generalizes to new, unseen data.



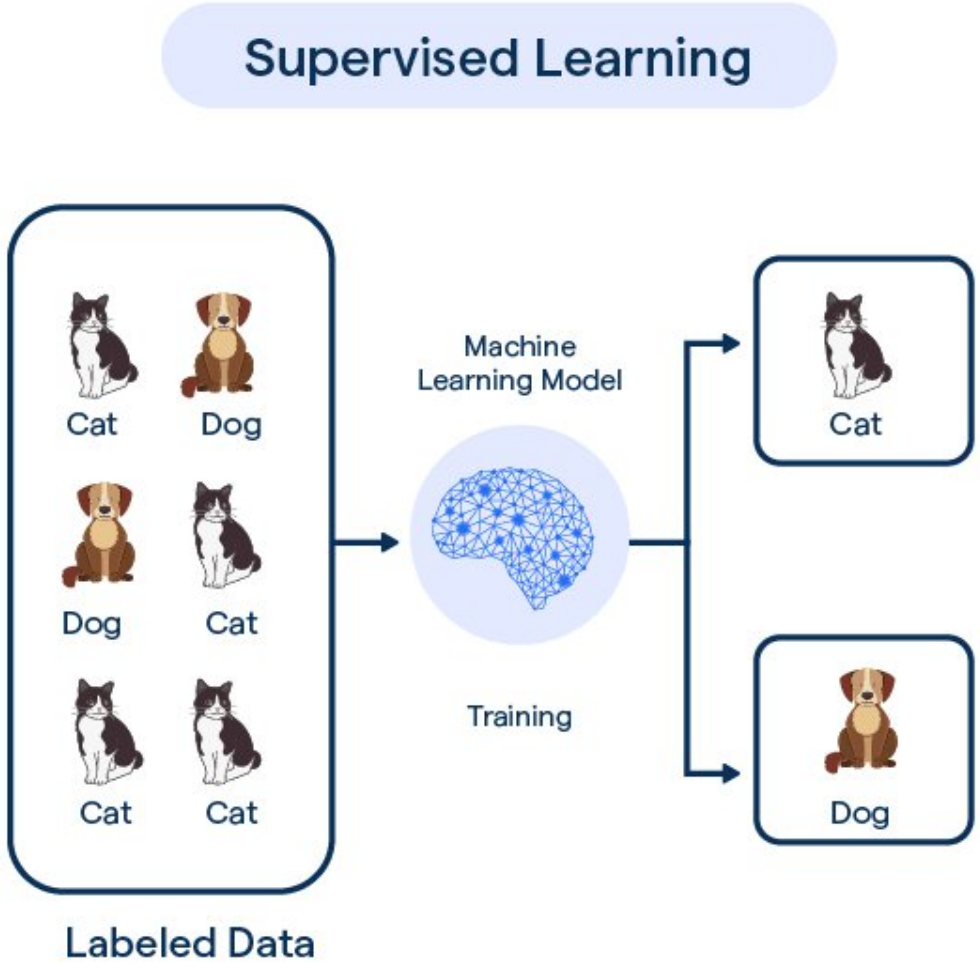
Key Concepts in Machine Learning

- **Algorithms:** Machine learning algorithms are the mathematical models or techniques used to learn patterns from data and make predictions or decisions. These algorithms can be classified into different types based on their learning approach, such as supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.



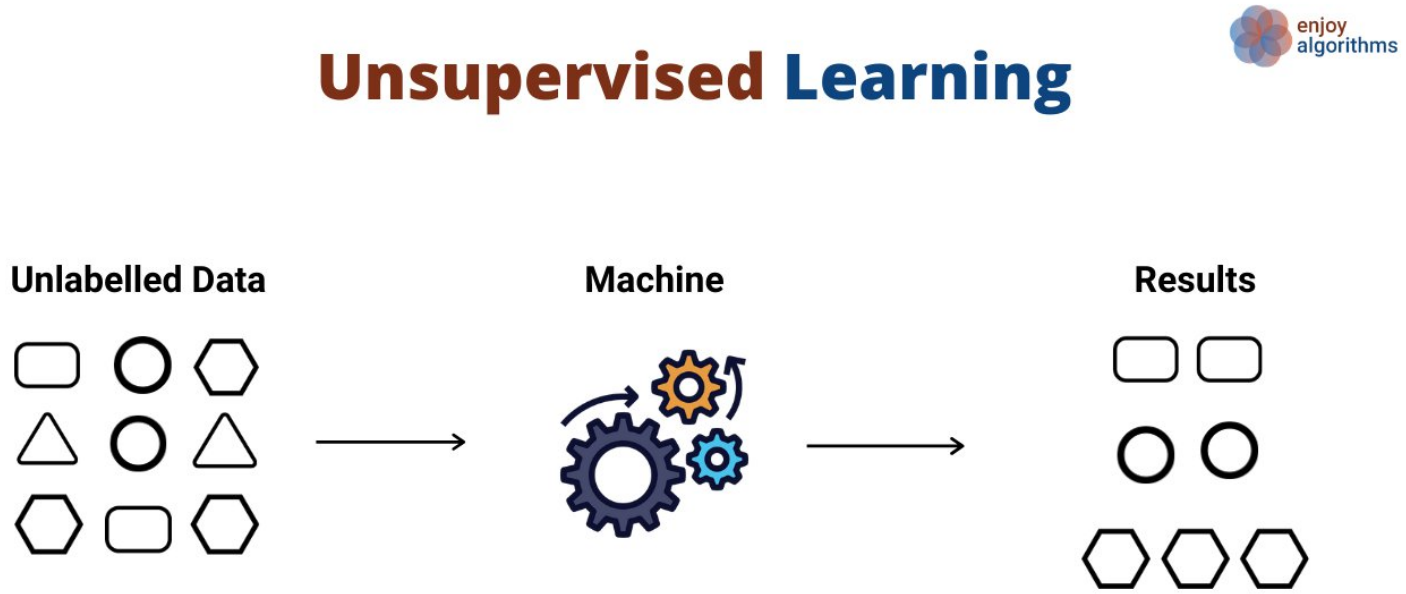
Types of Machine Learning Algorithms

- **Supervised Learning:** In this type of learning, the algorithm learns from labeled data, where each example in the training dataset is paired with a corresponding label. The goal is to learn a mapping from inputs to outputs.
- https://www.youtube.com/watch?v=W01tIRP_Rqs



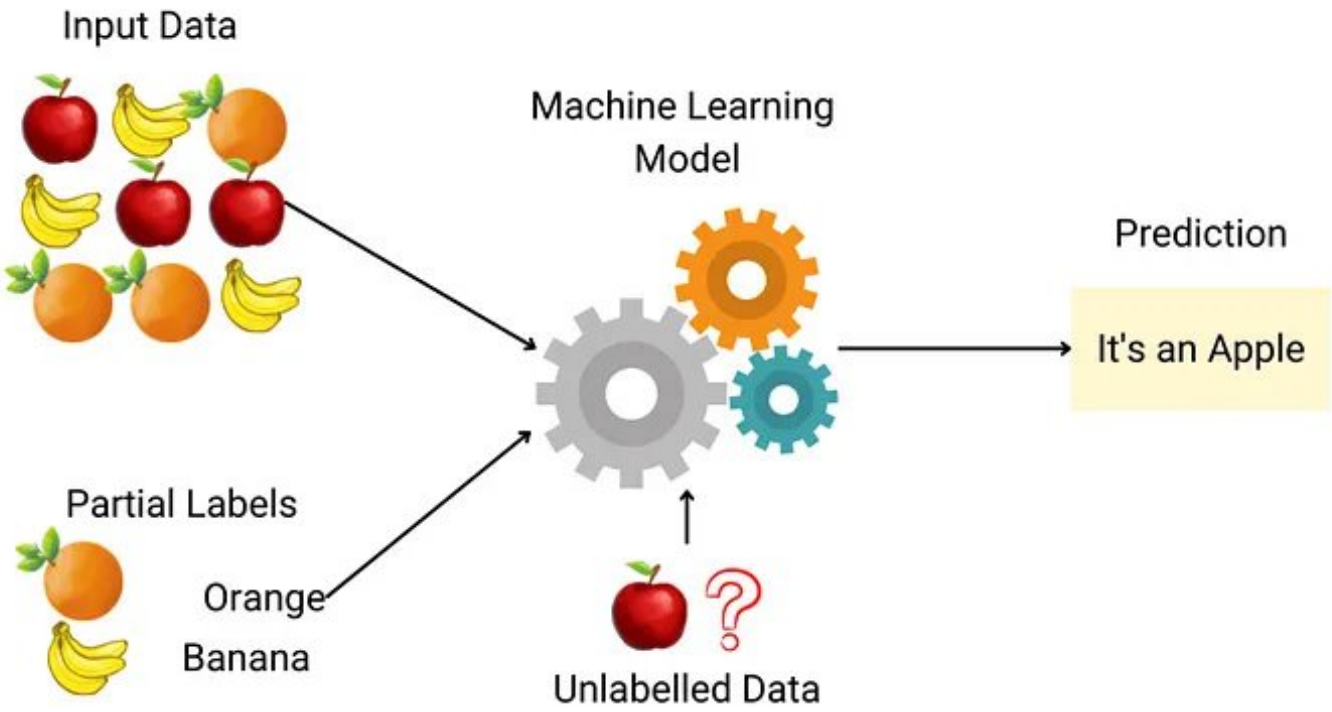
Types of Machine Learning Algorithms

- **Unsupervised Learning:** Here, the algorithm learns from unlabeled data, identifying patterns or structures within the data without explicit guidance. https://www.youtube.com/watch?v=W01tIRP_Rqs



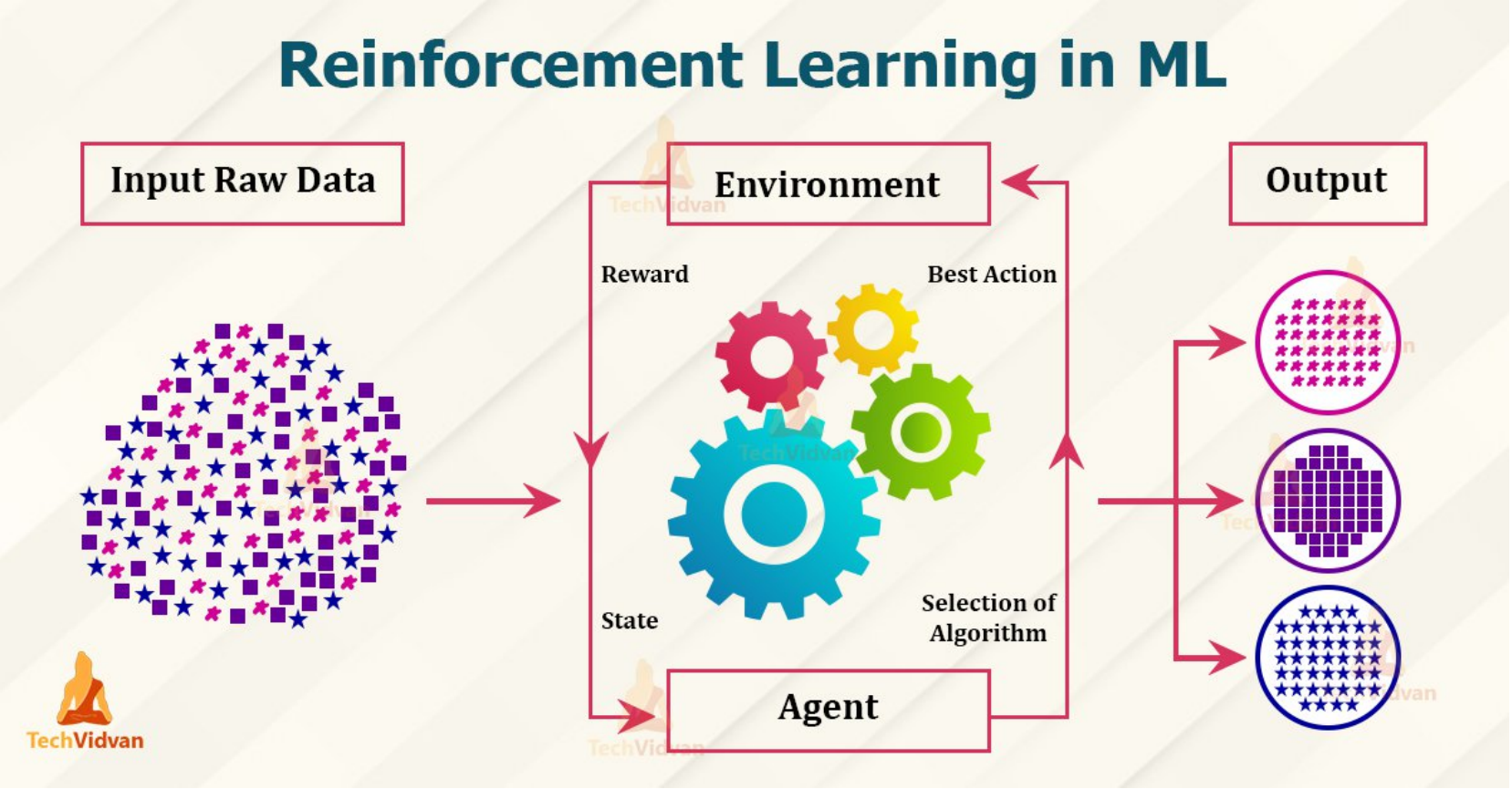
Types of Machine Learning Algorithms

- **Semi-supervised Learning:** This approach combines both labeled and unlabeled data for training. It's useful when labeled data is scarce or expensive to obtain. https://www.youtube.com/watch?v=b-yhKUINb7o



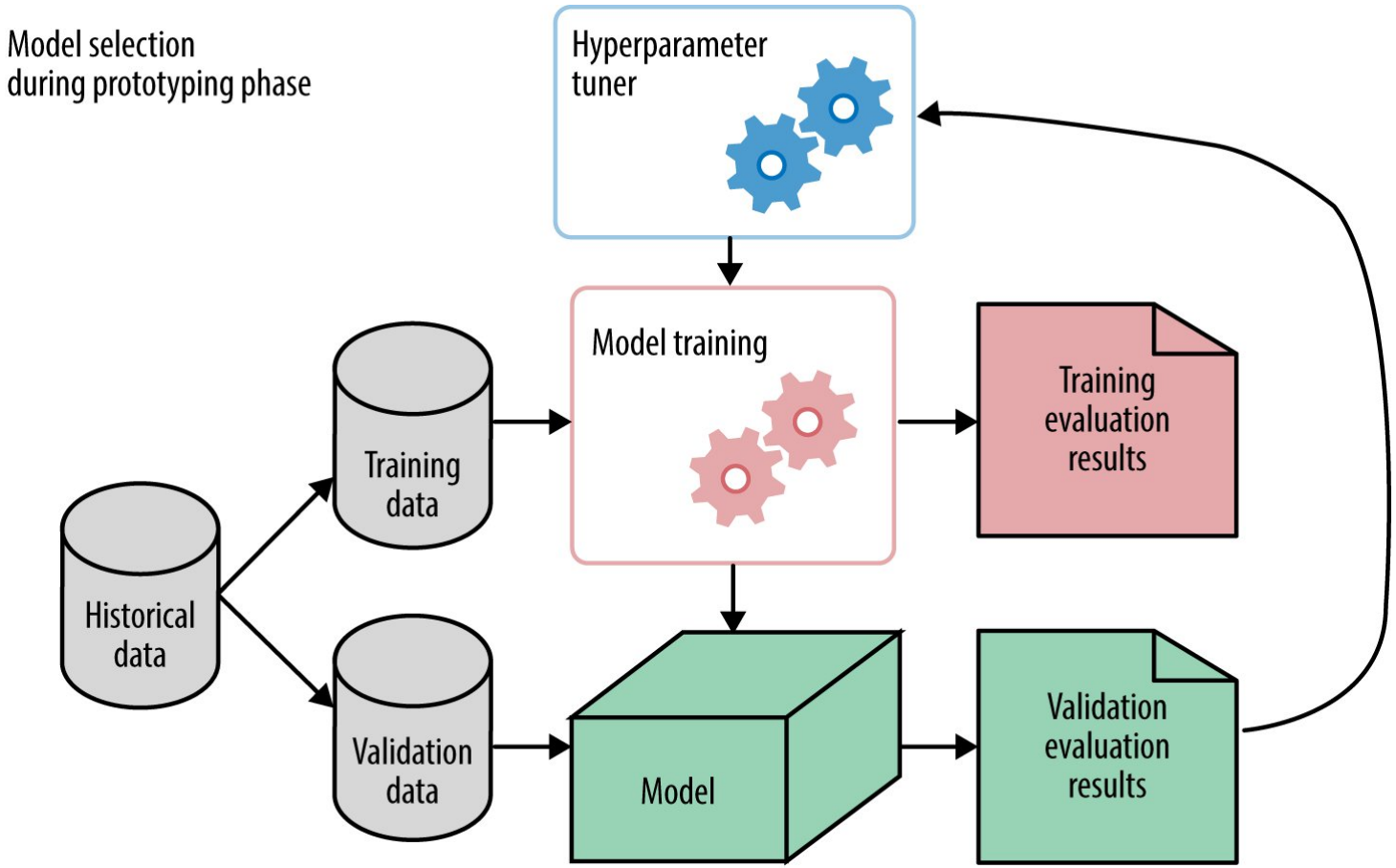
Types of Machine Learning Algorithms

- **Reinforcement Learning:** This type of learning involves an agent learning to interact with an environment to achieve a goal. The agent learns by receiving feedback in the form of rewards or penalties based on its actions. https://www.youtube.com/watch?v=nIgIv4IfJ6s



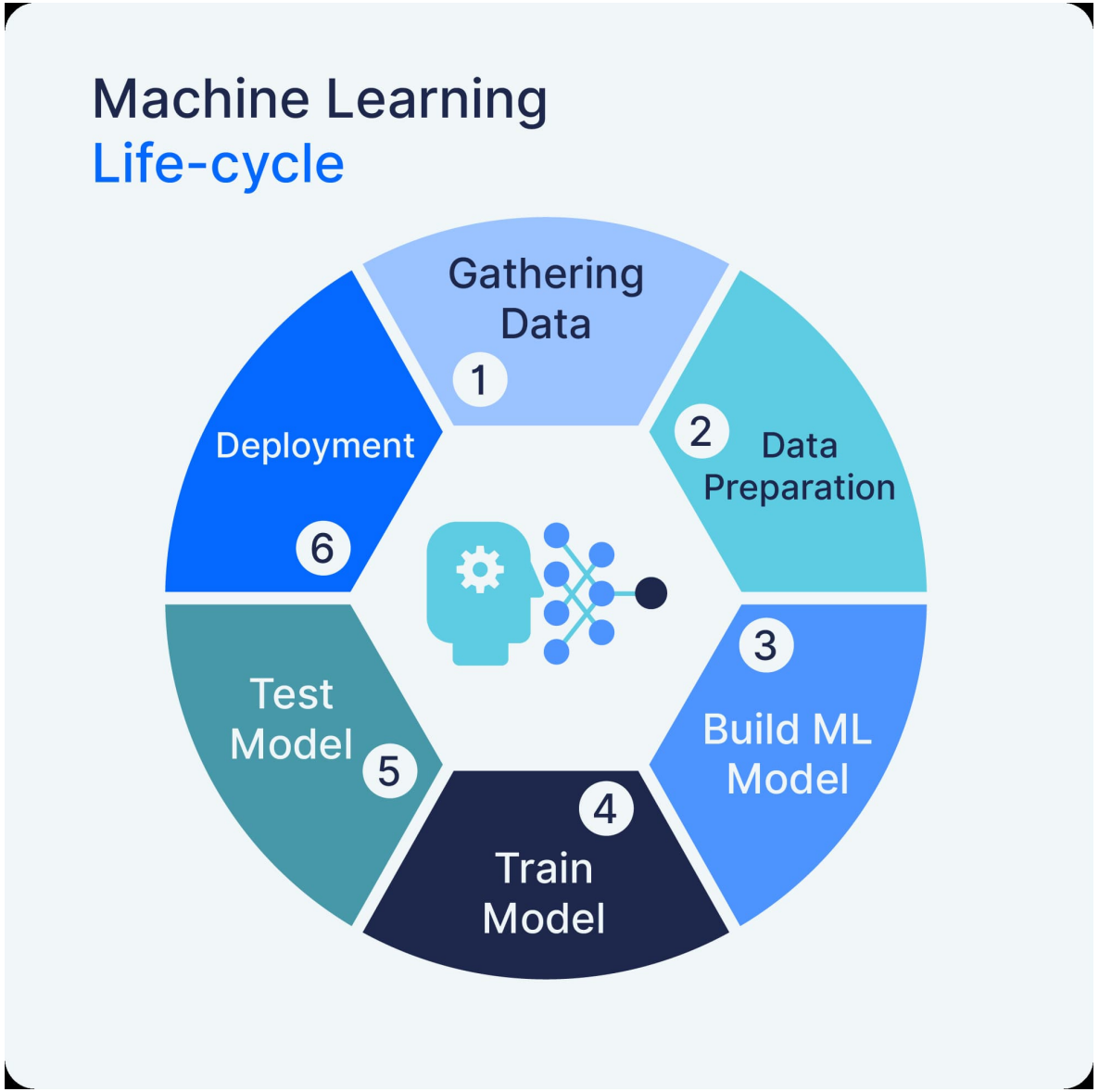
Key Concepts in Machine Learning

- **Model Evaluation:** Evaluating the performance of a machine learning model is crucial to assess its effectiveness. Common evaluation metrics vary depending on the type of problem and the specific goals of the model, such as accuracy, precision, recall, F1-score, and mean squared error.



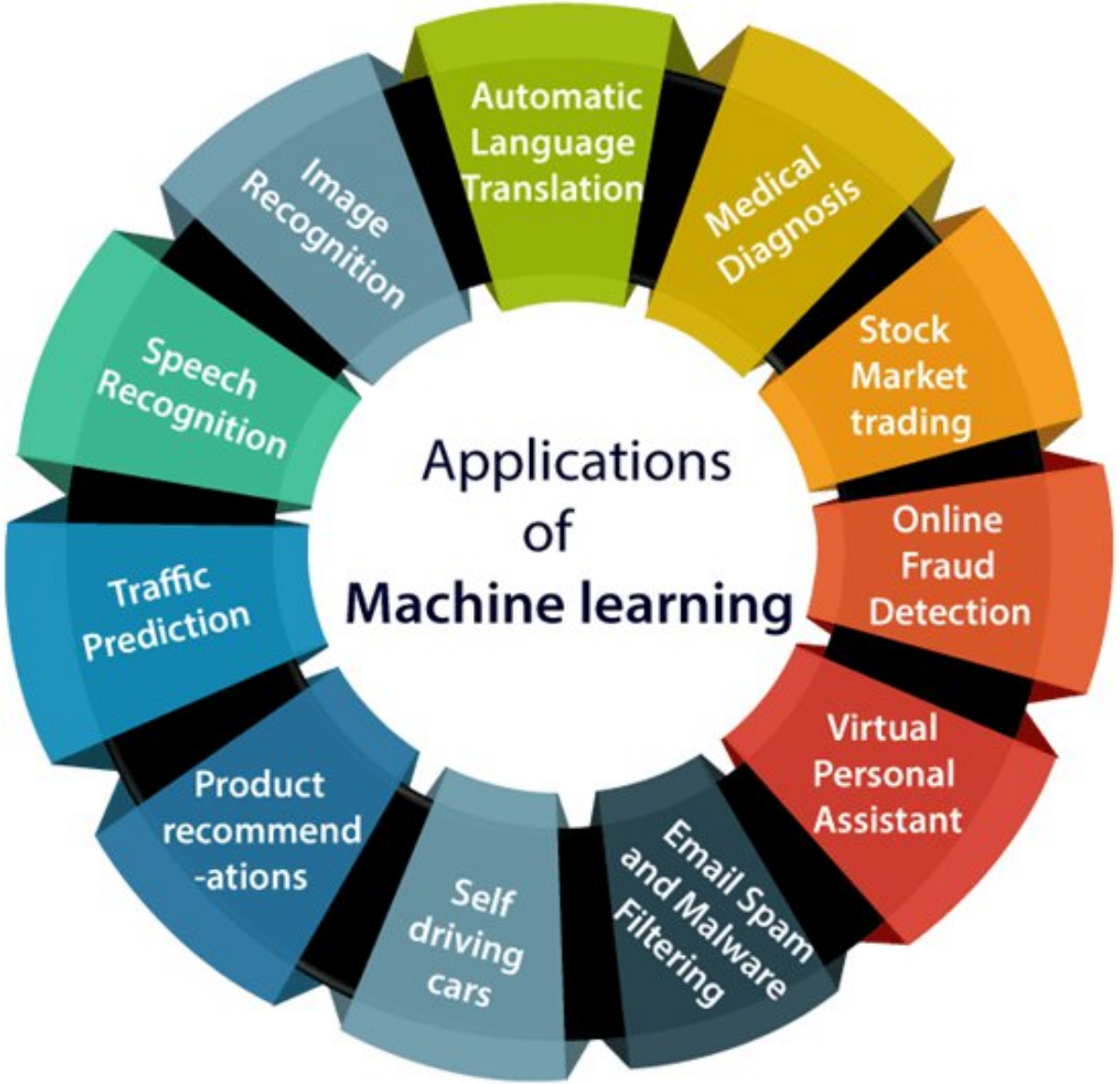
Key Concepts in Machine Learning

- **Model Deployment:** Once a model is trained and evaluated, it can be deployed to make predictions or decisions on new, unseen data. Deployment involves integrating the model into production systems or applications.



Applications of Machine Learning

Machine learning has numerous applications across various domains, including image and speech recognition, natural language processing, recommendation systems, healthcare, finance, and autonomous vehicles. As the field continues to evolve, new algorithms and techniques are constantly being developed to tackle increasingly complex problems and improve the capabilities of machine learning systems.



Applications of Machine Learning

Importance of Datasets in ML

Datasets are fundamental to the field of machine learning (ML). Their importance cannot be overstated, as they influence every aspect of the ML pipeline, from model training to evaluation and deployment. Here are several key reasons why datasets are crucial in ML:

1. **Training Models**
   - **Learning Patterns:** Machine learning models learn patterns, relationships, and features from data. Without a dataset, a model has nothing to learn from.
   - **Generalization:** A diverse and representative dataset helps a model generalize well to new, unseen data. A poor dataset can lead to overfitting or underfitting.
2. **Evaluation and Validation**
   - **Performance Metrics:** Datasets are used to evaluate model performance. By splitting data into training, validation, and test sets, practitioners can measure how well a model performs and tune it accordingly.
   - **Bias Detection:** Evaluation datasets help in identifying biases in model predictions. If a model performs poorly on certain subsets of data, it may indicate biases or deficiencies in the training process.

Importance of Datasets in ML

3. Benchmarking

- **Comparative Analysis:** Standard datasets allow for benchmarking and comparing different models and algorithms under consistent conditions.
- **Research and Development:** Researchers use well-known datasets to validate new methodologies and innovations in ML.

4. Data Quality and Preprocessing

- **Clean and Accurate Data:** High-quality datasets are essential for producing reliable models. Issues like missing values, noise, and inaccuracies must be addressed during data preprocessing.
- **Feature Engineering:** The features derived from datasets are critical for model performance. Effective feature engineering can significantly enhance a model's predictive power.

Importance of Datasets in ML

5. Ethics and Fairness

- **Bias and Fairness:** Datasets must be scrutinized for biases that could lead to unfair or unethical outcomes. Ensuring datasets are fair and representative is vital for creating ethical AI systems.
- **Transparency:** Transparent documentation of datasets, including their sources and characteristics, is important for reproducibility and trust in ML systems.

6. Real-world Applications

- **Relevance to Use Cases:** The dataset must be relevant to the specific application or problem being solved. Different applications may require different types of data (e.g., text, images, time series).
- **Adaptability:** In dynamic environments, datasets need to be continuously updated to reflect changing conditions and maintain model accuracy.

Importance of Datasets in ML

Datasets are the backbone of machine learning. They provide the raw material from which models learn and are essential for training, evaluation, benchmarking, and ensuring ethical standards. The quality, diversity, and relevance of datasets directly impact the success and reliability of ML applications. Therefore, careful consideration and handling of datasets are crucial at every stage of the machine learning process.

Types of Datasets in Machine learning

In machine learning, datasets can be categorized based on their characteristics, structure, and the type of problem they are used to solve. Here are the primary types of datasets:

1. Structured vs. Unstructured Data

Structured Data

- **Definition:** Data that is organized in a predefined manner, often in tabular format with rows and columns.
- **Examples:** Spreadsheets, SQL databases.
- **Use Cases:** Financial records, customer databases, sensor data.

Unstructured Data

- **Definition:** Data that does not have a predefined format or organization.
- **Examples:** Text documents, images, audio files, videos.
- **Use Cases:** Natural language processing (NLP), image recognition, speech-to-text conversion.

Types of Datasets in Machine learning

2. Labeled vs. Unlabeled Data

Labeled Data

- **Definition:** Data that has been tagged with one or more labels, providing explicit information about the target variable.
- **Examples:** Annotated images (with objects labeled), spam vs. non-spam emails.
- **Use Cases:** Supervised learning tasks such as classification and regression.

Unlabeled Data

- **Definition:** Data without any labels or target variables.
- **Examples:** Raw text, unlabeled images, customer behavior data.
- **Use Cases:** Unsupervised learning tasks such as clustering, anomaly detection.

Types of Datasets in Machine learning

3. Training, Validation, and Test Sets

Training Set

- **Definition:** The portion of the dataset used to train the machine learning model.
- **Purpose:** To allow the model to learn patterns and relationships in the data.

Validation Set

- **Definition:** A subset of the dataset used to tune model parameters and make decisions about model architecture.
- **Purpose:** To provide an unbiased evaluation of a model fit on the training dataset while tuning hyperparameters.

Test Set

- **Definition:** The portion of the dataset used to evaluate the final model performance.
- **Purpose:** To provide an unbiased assessment of the model's performance on unseen data.

Types of Datasets in Machine learning

4. Categorical vs. Numerical Data

Categorical Data

- **Definition:** Data that represents categories or groups.
- **Examples:** Gender (male, female), product type (electronics, furniture).
- **Use Cases:** Classification problems, one-hot encoding.

Numerical Data

- **Definition:** Data that represents numbers and can be discrete or continuous.
- **Examples:** Age, income, temperature.
- **Use Cases:** Regression problems, feature scaling.

5. Time Series Data

- **Definition:** Data points collected or recorded at specific time intervals.
- **Examples:** Stock prices, weather data, sensor readings.
- **Use Cases:** Forecasting, anomaly detection, trend analysis.

Types of Datasets in Machine learning

6. Text Data

- **Definition:** Data in the form of natural language text.
- **Examples:** Social media posts, customer reviews, research papers.
- **Use Cases:** Sentiment analysis, language translation, text classification.

7. Image Data

- **Definition:** Data in the form of images.
- **Examples:** Photographs, medical scans, satellite images.
- **Use Cases:** Image classification, object detection, image segmentation.

8. Audio Data

- **Definition:** Data in the form of sound recordings.
- **Examples:** Speech recordings, music, environmental sounds.
- **Use Cases:** Speech recognition, audio classification, music generation.

Types of Datasets in Machine learning

9. Video Data

- **Definition:** Data in the form of moving images.
- **Examples:** Surveillance footage, video clips, movies.
- **Use Cases:** Action recognition, video summarization, video segmentation.

Different types of datasets serve various purposes and are used in different machine learning tasks. Understanding the nature of the data and choosing the right type of dataset for a specific problem is crucial for developing effective and accurate machine learning models.

Data Collection in Machine learning

Data collection is a critical step in the machine learning pipeline, as the quality and quantity of data directly impact the performance of the machine learning models. Here's a comprehensive overview of data collection in machine learning:

1. Define Objectives and Requirements

1. **Clarify Goals:** Understand the problem you are trying to solve and the objectives of the machine learning project.
2. **Identify Data Needs:** Determine the type and amount of data required to achieve your objectives. Consider factors like data attributes, sources, and quality.

2. Identify Data Sources

1. **Internal Sources:** Databases, logs, and records within the organization.
2. **External Sources:** Public datasets, third-party providers, APIs, and web scraping.
3. **Sensors and IoT Devices:** Data collected from physical devices in real-time.

Data Collection in Machine learning

3. Data Acquisition

- **Manual Data Entry:** Collecting data by human effort, such as surveys and interviews.
- **Automated Data Collection:** Using scripts, APIs, or software tools to gather data from various sources automatically.
- **Third-party Data:** Purchasing or licensing data from external vendors.

4. Data Integration

- **Combining Data:** Merge data from different sources to create a comprehensive dataset.
- **Data Cleaning:** Remove duplicates, handle missing values, and correct errors to ensure data quality.
- **Normalization and Standardization:** Ensure consistency in data formats and units.

Data Collection in Machine learning

Data Storage and Management

- **Database Systems:** Use databases to store and manage large volumes of data.
- **Cloud Storage:** Utilize cloud-based solutions for scalable and flexible data storage.
- **Data Warehousing:** Implement data warehouses for efficient querying and analysis

Considerations for Data Collection in Machine learning

1.Data Quality

- **Accuracy:** Ensure the data accurately represents the real-world conditions.
- **Completeness:** Collect all necessary data without significant gaps.
- **Consistency:** Maintain uniformity in data entries and formats.
- **Timeliness:** Ensure the data is up-to-date and relevant.

2.Ethical and Legal Aspects

- **Privacy:** Respect user privacy and comply with data protection regulations (e.g., GDPR, CCPA).
- **Consent:** Obtain necessary permissions from data subjects before collecting data.
- **Bias and Fairness:** Ensure the data is representative and does not introduce biases.

Considerations for Data Collection in Machine learning

3. **Data Volume and Variety**

- ❖ Sufficient Quantity: Collect enough data to train and validate the model effectively.
- ❖ Diversity: Include a variety of data to ensure the model can generalize well to different scenarios.

4. **Cost and Resources**

- ❖ Budget: Consider the costs associated with data collection, including purchasing data and storage costs.
- ❖ Time and Effort: Assess the time and resources required for data collection and preparation.

Challenges in Data Collection in Machine learning

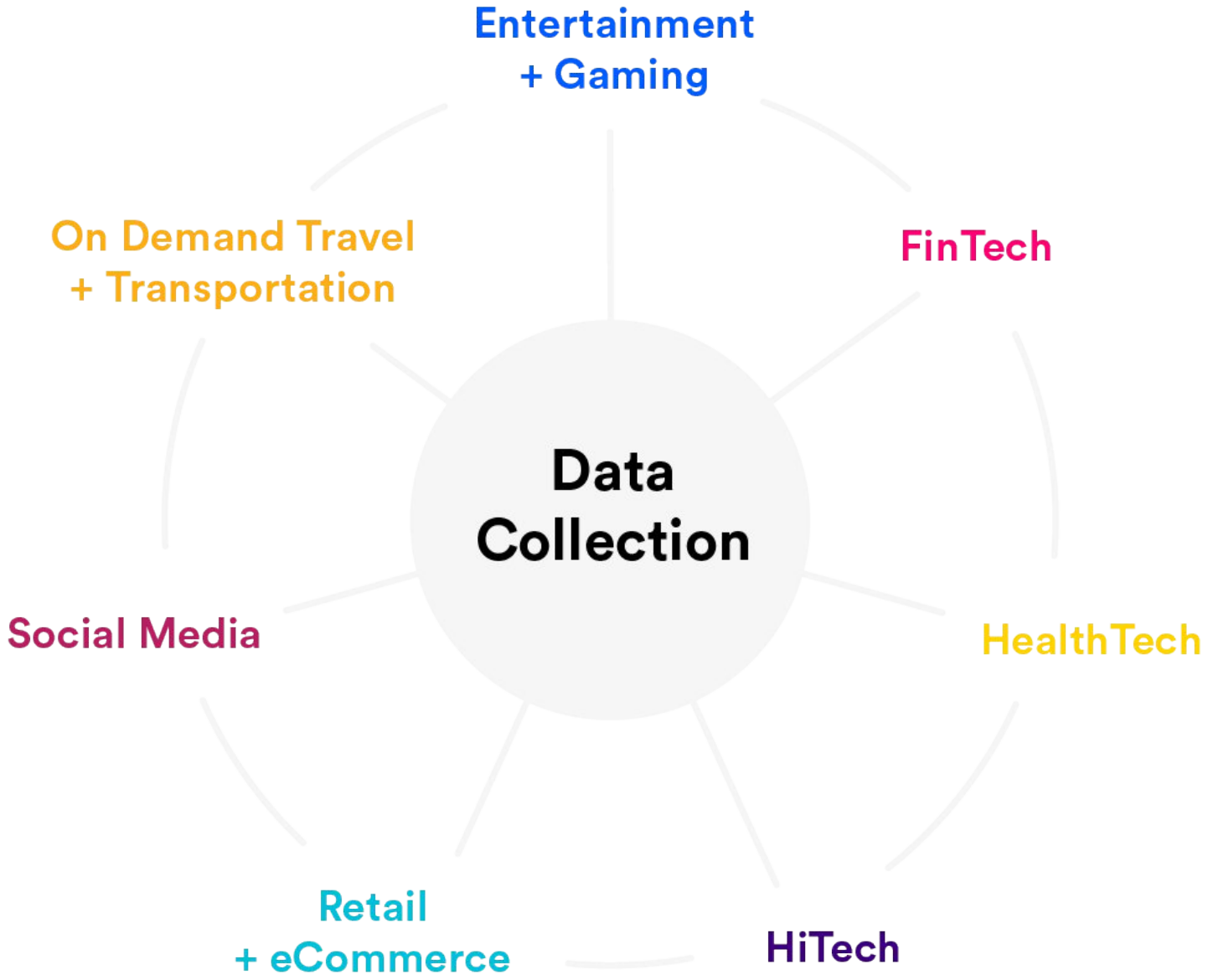
1. **Data Quality Issues:** Ensuring data accuracy and consistency can be challenging, especially with large datasets.
2. **Data Integration:** Combining data from multiple sources can be complex due to different formats and structures.
3. **Scalability:** Handling large volumes of data requires scalable storage and processing solutions.
4. **Privacy and Security:** Protecting sensitive data and ensuring compliance with legal requirements.

1. **Cost and Time:** Data collection can be resource-intensive, both in terms of time and cost.

Handling datasets for Machine Learning Feature sets

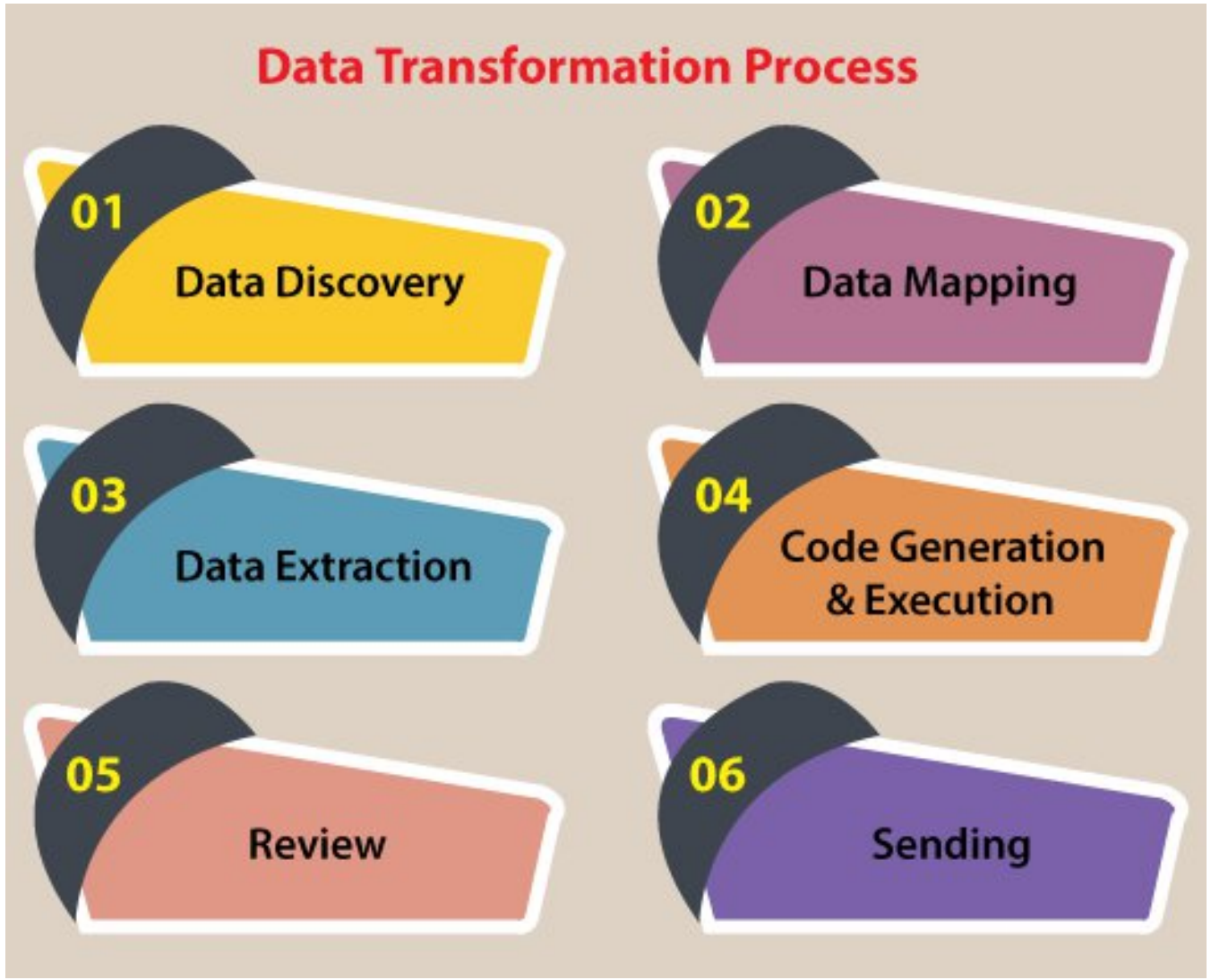
- Handling datasets for machine learning feature sets involves several key steps. Here's a comprehensive guide to manage and prepare your datasets effectively:

- **1. Data Collection**
  - **Identify Data Sources:** Determine the sources from where the data will be collected (databases, APIs, web scraping, sensors, etc.).
  - **Gather Data:** Collect the data ensuring you have enough examples to train a robust model.



Handling datasets for Machine Learning Feature sets

- **2. Data Cleaning**
  - **Remove Duplicates:** Eliminate duplicate records to avoid redundancy.
  - **Handle Missing Values:** Impute missing values using strategies like mean/median imputation, forward/backward fill, or removing the records/columns with excessive missing values.
  - **Correct Errors:** Fix any errors in the data such as incorrect labels, out-of-range values, etc.



Handling datasets for Machine Learning Feature sets

- **3. Data Transformation**
  - **Normalization/Standardization:** Scale the features so they have similar ranges. Common techniques include min-max normalization and z-score standardization.
  - **Encoding Categorical Variables:** Convert categorical variables to numerical using methods like one-hot encoding, label encoding, or target encoding.
  - **Feature Engineering:** Create new features from existing ones to help the model learn better. This includes creating interaction terms, polynomial features, and using domain knowledge to derive new features.

Normalization

- **Min-max normalization**: to [new_min_A, new_max_A]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,600 is mapped to

$$\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$$

- **Z-score normalization** (μ: mean, σ: standard deviation):

$$v' = \frac{v - m_A}{s_A}$$

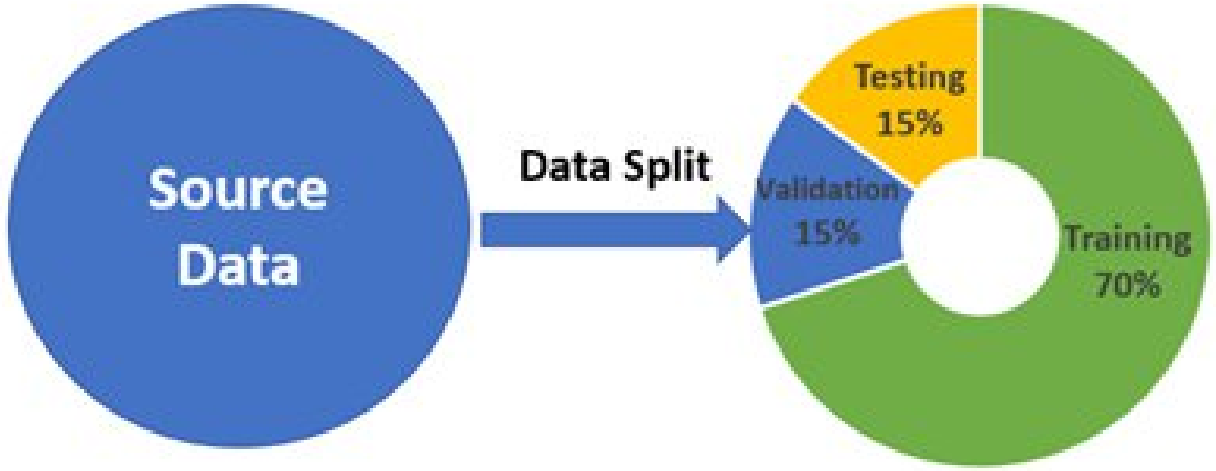
- Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max(|v'|) < 1

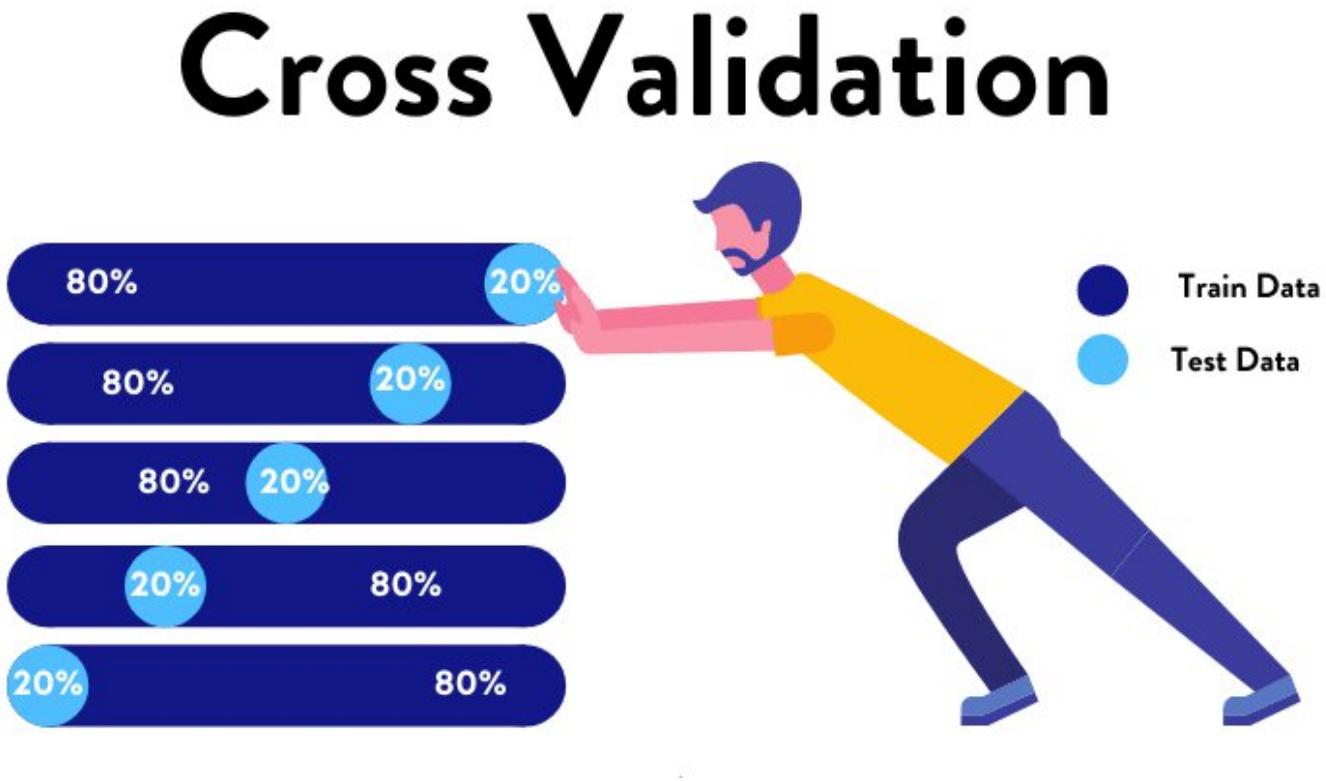
Handling datasets for Machine Learning Feature sets

- **4. Data Splitting**
  - **Train-Test Split:** Split the data into training and testing sets to evaluate the model's performance on unseen data.
  - **Validation Set:** Further split the training data into a training set and a validation set to tune hyperparameters and avoid overfitting.
  - **Cross-Validation:** Use k-fold cross-validation to make the best use of the data, especially when you have limited data.



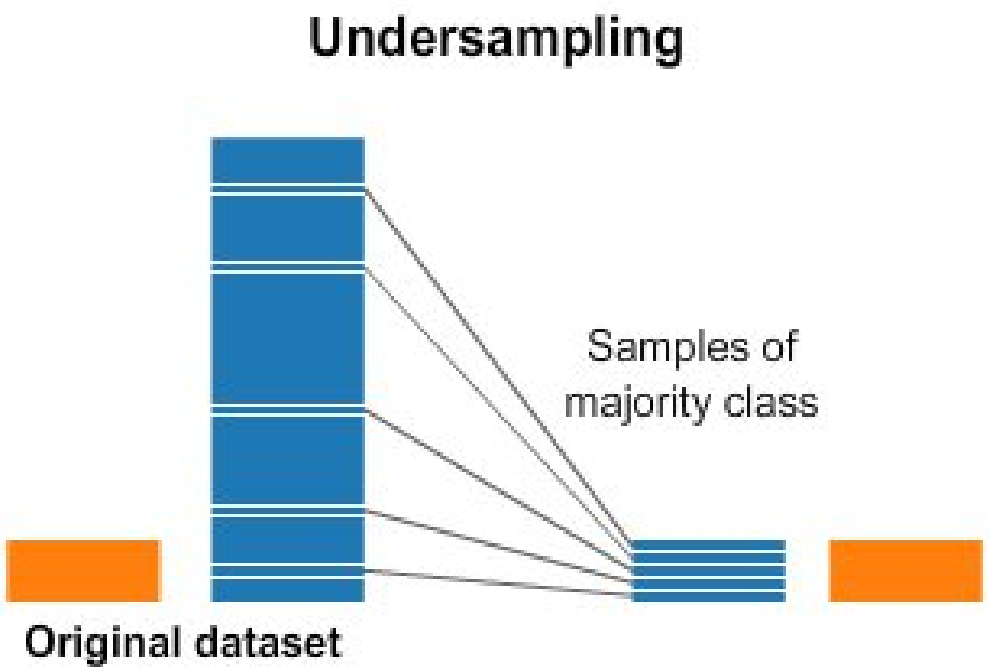
Handling datasets for Machine Learning Feature sets

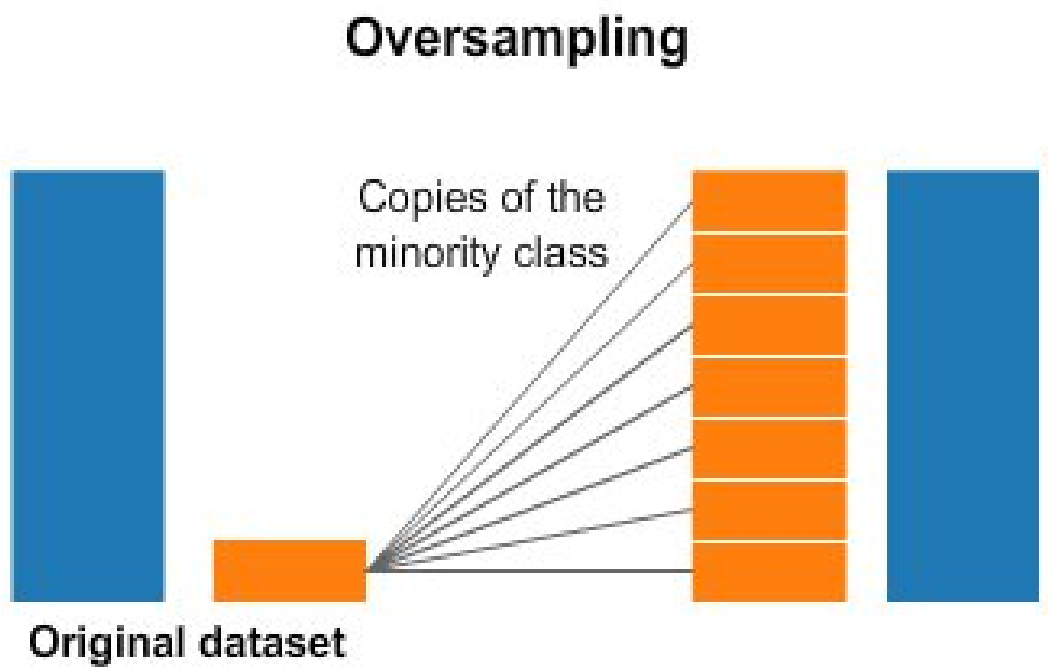
- **Cross-Validation:** Use k-fold cross-validation to make the best use of the data, especially when you have limited data.



Handling datasets for Machine Learning Feature sets

- **5. Handling Imbalanced Data**
  - **Resampling:** Use techniques like oversampling the minority class (e.g., SMOTE) or undersampling the majority class.
  - **Class Weighting:** Assign different weights to classes to balance the influence of each class on the model training.





Handling datasets for Machine Learning Feature sets

5. Handling Imbalanced Data

- **Resampling:** Use techniques like oversampling the minority class (e.g., SMOTE) or undersampling the majority class.
- **Class Weighting:** Assign different weights to classes to balance the influence of each class on the model training.

6. Feature Selection

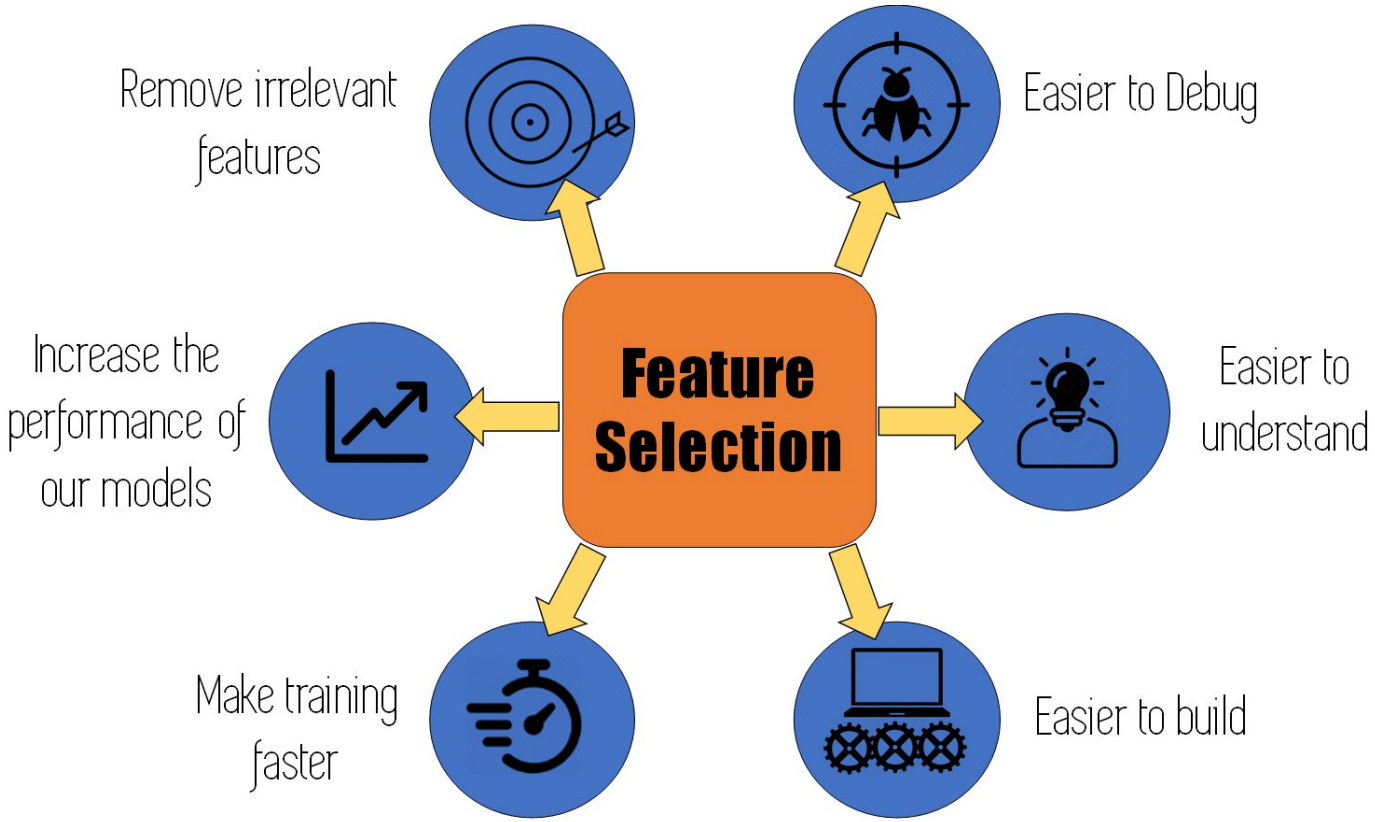
- **Remove Unnecessary Features:** Drop features that do not contribute to the model performance.
- **Use Algorithms:** Employ algorithms (like LASSO, Decision Trees) that help in selecting important features.
- **Correlation Analysis:** Remove highly correlated features to reduce multicollinearity.

7. Feature Scaling

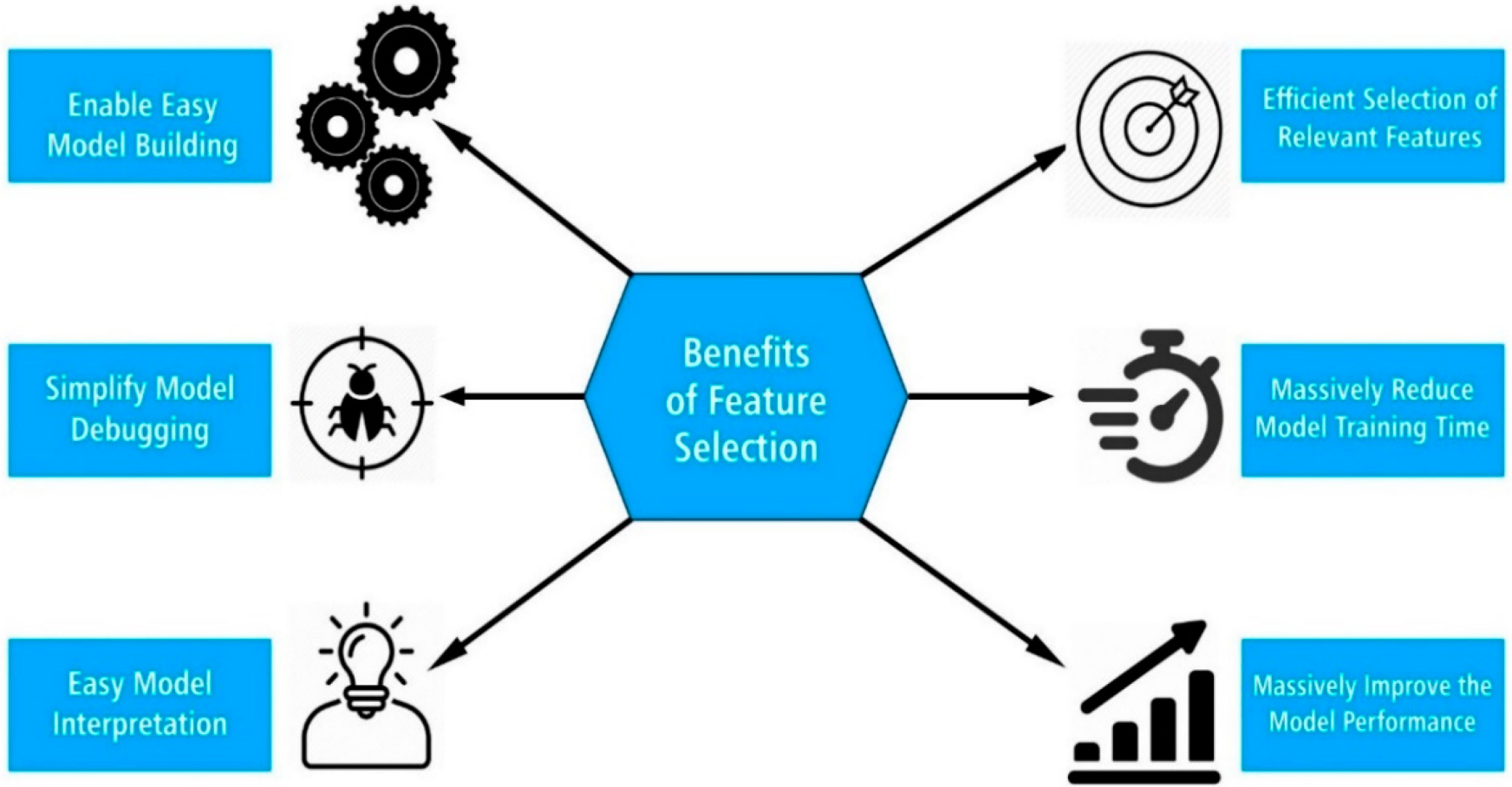
- **Normalization:** Scale features to a range, typically [0, 1].
- **Standardization:** Transform features to have zero mean and unit variance.

Handling datasets for Machine Learning Feature sets

- **6. Feature Selection**
  - **Remove Unnecessary Features:** Drop features that do not contribute to the model performance.
  - **Use Algorithms:** Employ algorithms (like LASSO, Decision Trees) that help in selecting important features.
  - **Correlation Analysis:** Remove highly correlated features to reduce multicollinearity.



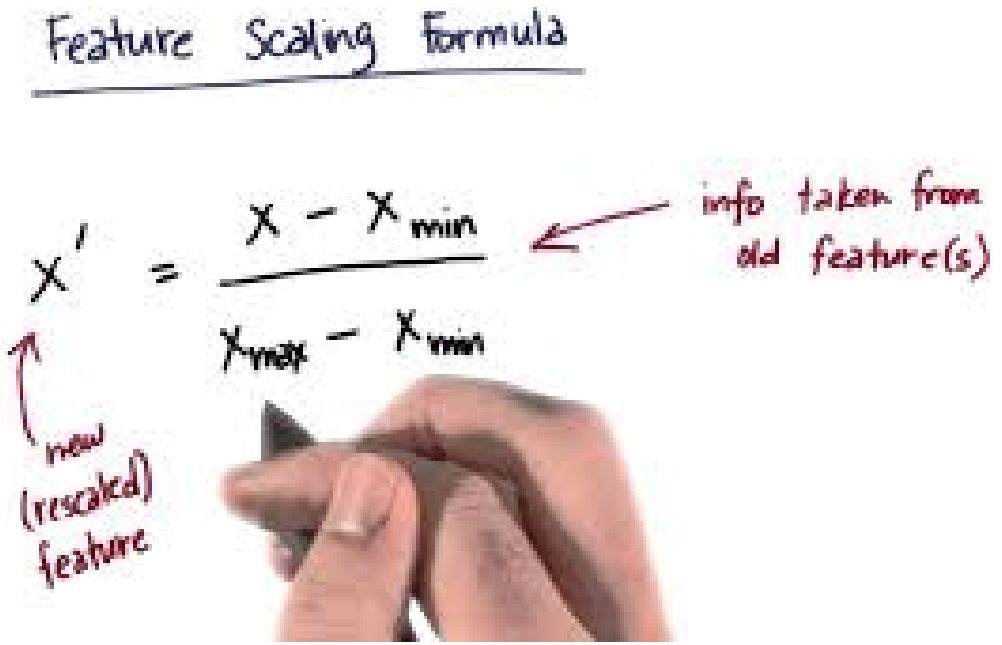
Handling datasets for Machine Learning Feature sets



Handling datasets for Machine Learning Feature sets

7. Feature Scaling: Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

- **Normalization:** Scale features to a range, typically [0, 1].
- **Standardization:** Transform features to have zero mean and unit variance.



Handling datasets for Machine Learning Feature sets

8. Data Augmentation

- **Generate New Data:** For image, text, or audio data, create variations of existing data to increase the dataset size.
- In machine learning, data augmentation is a common method for manipulating existing data to artificially increase the size of a training dataset. In an attempt to enhance the efficiency and flexibility of machine learning models, data augmentation looks for the boost in the variety and volatility of the training data.
- Data augmentation can be especially beneficial when the original set of data is small as it enables the system to learn from a larger and more varied group of samples.

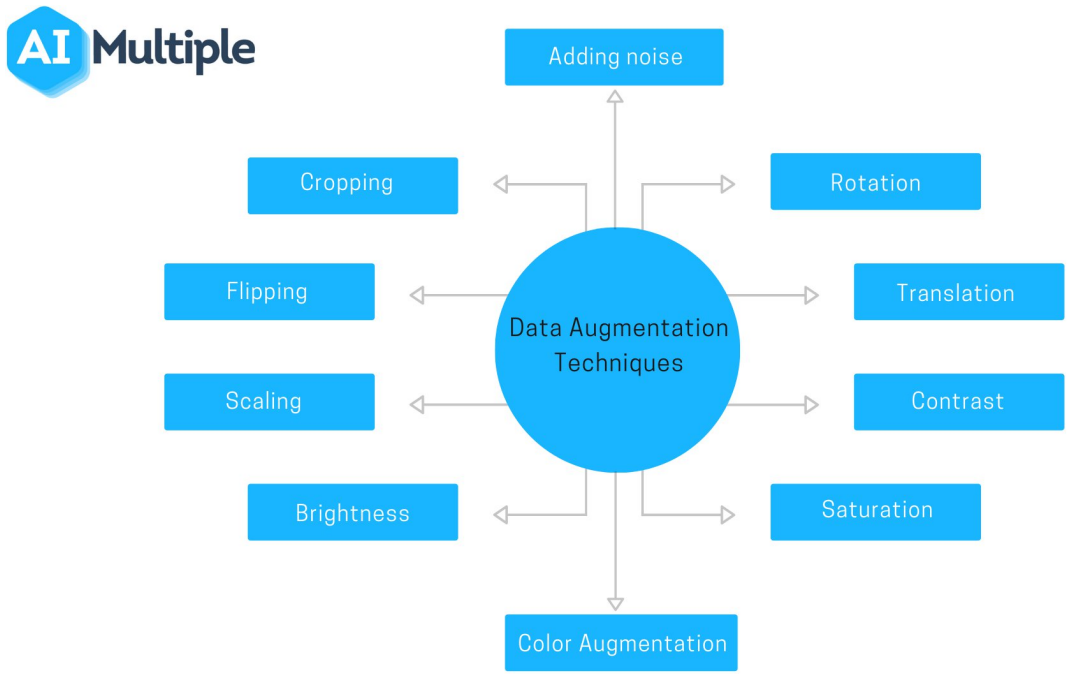
Types of Data Augmentation: Techniques for data augmentation can be used with a variety of data kinds, including time series, text, photos, and audio. Here are a few frequently used methods of data augmentation for image data:

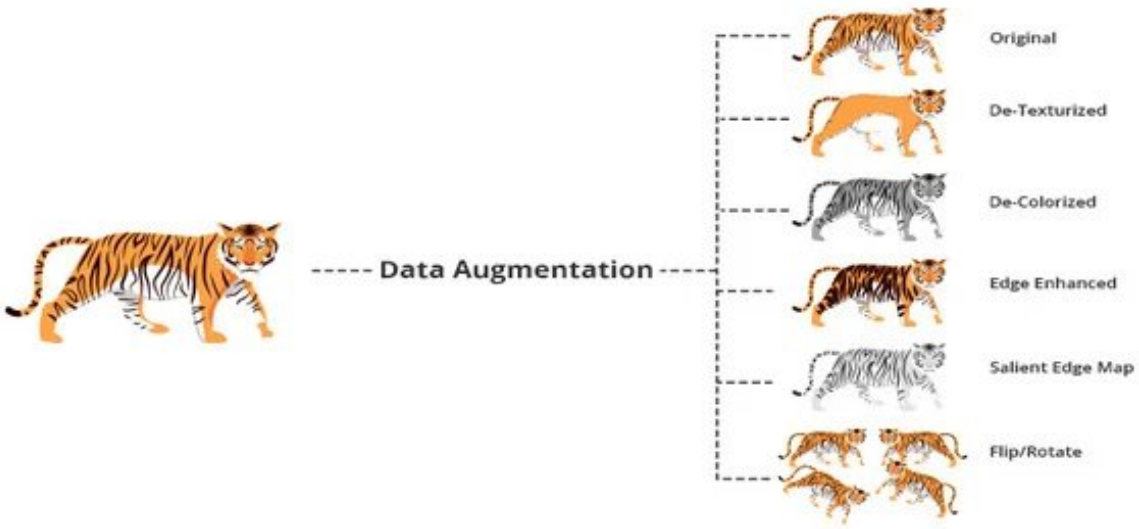
- Images can be rotated at different angles and flipped horizontally or vertically to create alternative points of view.

Handling datasets for Machine Learning Feature sets

- Random cropping and padding: By applying random cropping or padding to the photos, various scales, and translations can be simulated.
- Scaling and zooming: The model can manage various item sizes and resolutions by rescaling the photos to different sizes or zooming in and out.
- Shearing and perspective transform: Changing an image's shape or perspective can imitate various viewing angles while also introducing deformations.
- Color jittering: By adjusting the color characteristics of the images, including their brightness, contrast, saturation, and hue, the model can be made to be more resilient to variations in illumination.
- Gaussian noise: By introducing random Gaussian noise to the images, the model's resistance to noisy inputs can be strengthened.

Handling datasets for Machine Learning Feature sets





Handling datasets for Machine Learning Feature sets

9. Data Storage

- **Save Cleaned Data:** Store the cleaned and preprocessed data in an appropriate format (CSV, HDF5, etc.) for future use.
- **Document the Process:** Keep track of the steps and transformations applied to the data for reproducibility.

What is Data Division?

Data division (or splitting) is when data is divided into two or more subsets.

Data division helps us divide the dataset into different sections of different utility.

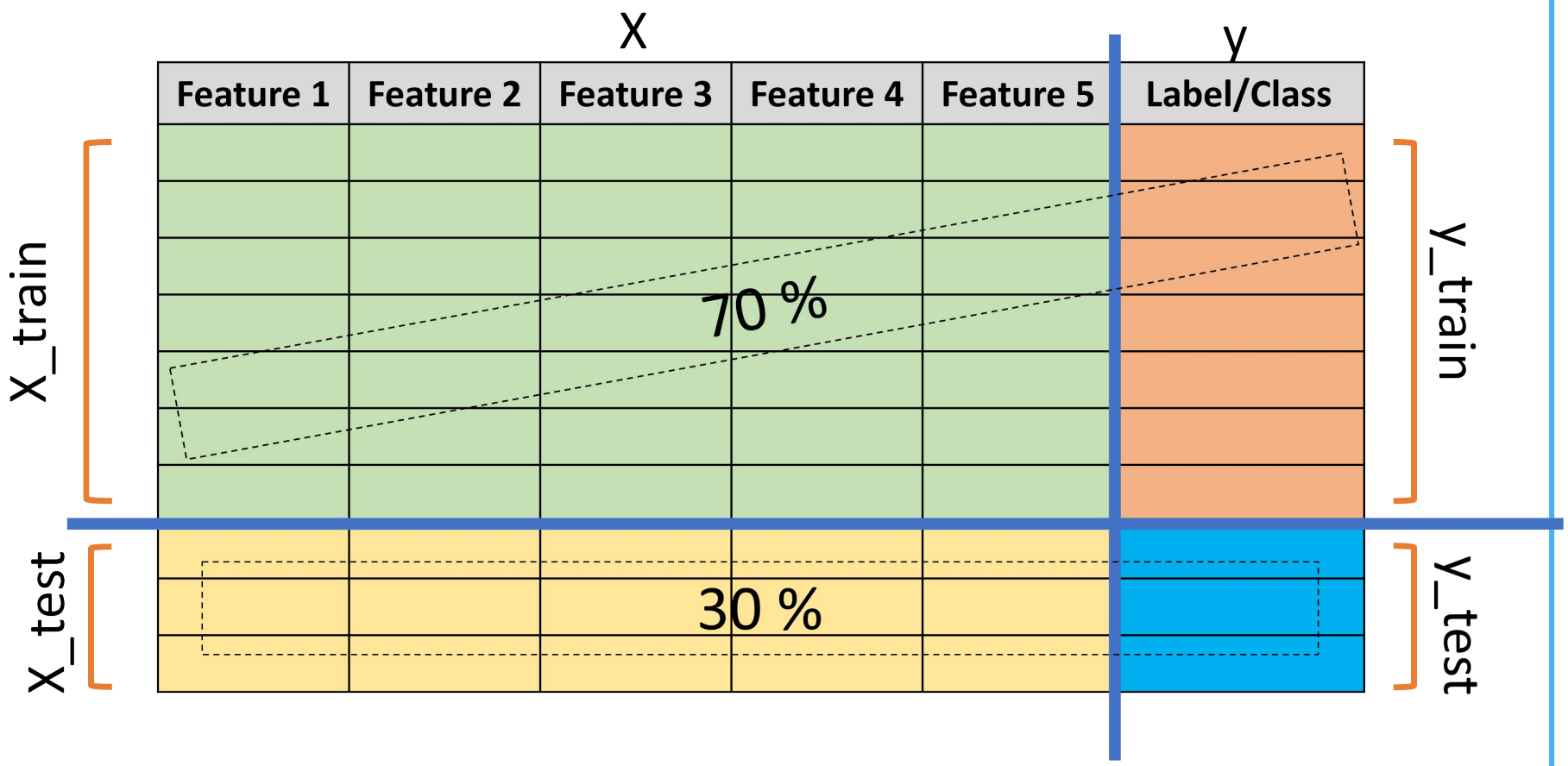
Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model.

Data splitting is an important and fundamental aspect of data science, particularly for creating models based on data.

What happens in Data Division?

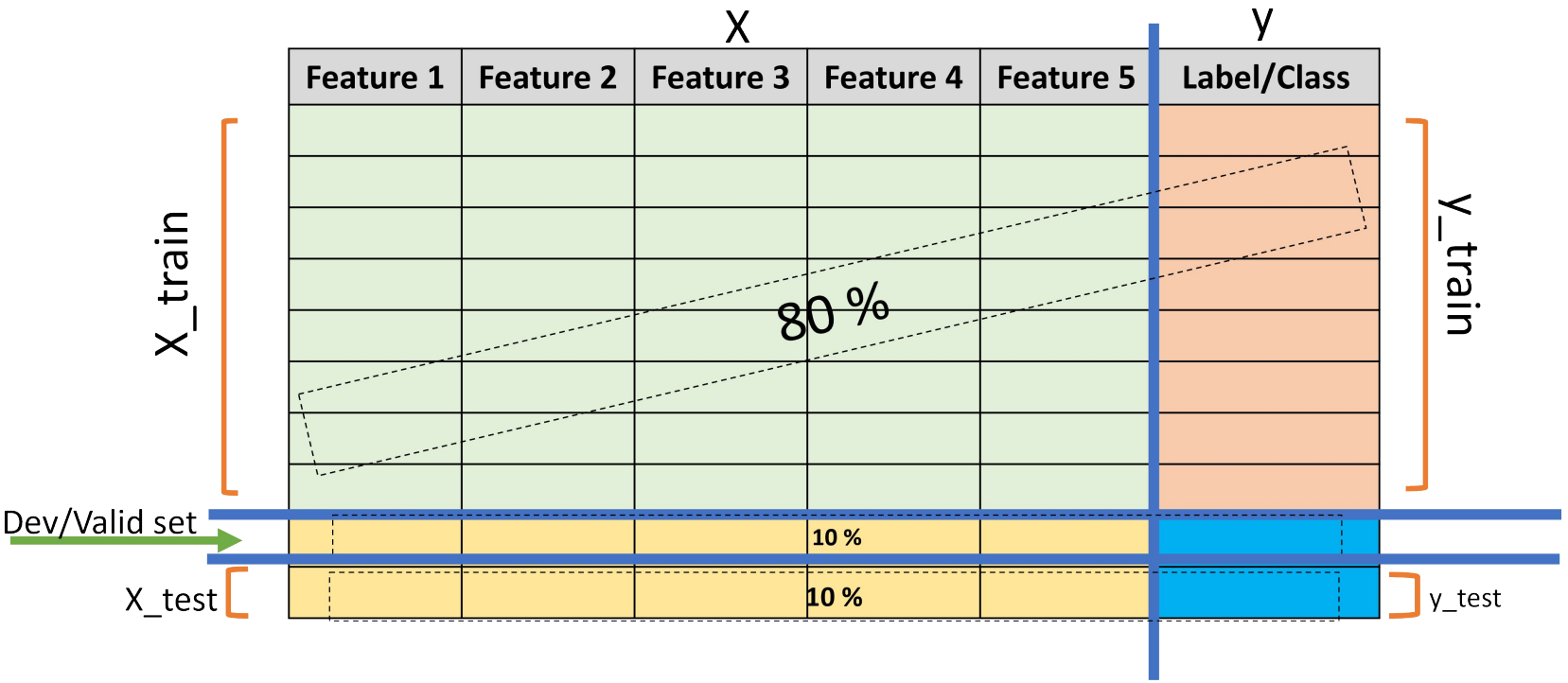
UNDERSTAND YOUR DATA

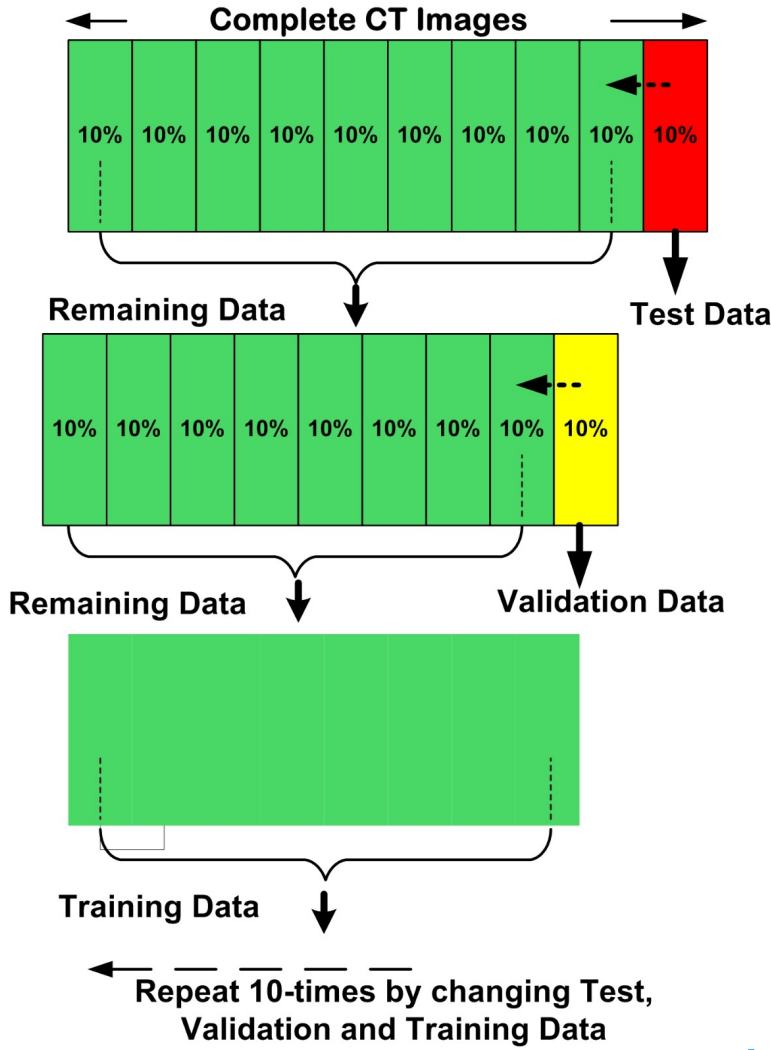
Let's say we have a classification dataset with 5 features (Independent variables- X) and a class (dependent variable- y)



Note: This division can be 70:30, 80:20 or 90:10, depending upon the data and requirements of the experiment.

What happens in





Note: Variables used as per Python's most-popular ML library Scitkit Learn's usage.

What happens in

- Best practices include shuffling of data before splitting, so that the test set represents all members of the dataset.
- The Machine Learning models are trained using X_train and y_train parts of the data.
- After learning is complete, the model is then tested using X_test part of the dataset and the model predicts answers (y_pred) based on its learning.
- At last, the model's prediction (y_pred) is compared with the actual outputs (y_test) and model's performance is evaluated.

Why do we need Data Division?

To avoid *overfitting* in machine learning models.

What is Overfitting?

That is an instance where a machine learning model fits its training data too well and fails to reliably fit additional data.

Components of Data Division

The three sets commonly used are the training set, the dev set and the testing set:

The Training set:

The training set is the portion of the dataset used to train the machine learning model.

It contains examples of input data along with their corresponding output labels or target values.

During the training process, the model learns patterns and relationships in the training data that allow it to make predictions or decisions.

The Development (or dev) set

It is also called validation (or valid) set.

The dev set is used to evaluate the performance of the model during training and tune hyperparameters.

It serves as an independent dataset that helps assess how well the model generalizes to unseen data.

By monitoring the model's performance on the validation set, adjustments can be made to optimize its performance and prevent overfitting.

Test Set

The test set is used to evaluate the final performance of the trained model and to provide unbiased assessment of the model's ability to make predictions or decisions on unseen data.

The test set should not be used during model training or hyperparameter tuning to ensure an objective evaluation of the model's generalization performance.

The typical division ratio among these sets depends on factors such as the size of the dataset and the specific requirements of the problem.

Common practice is to allocate most of the data to the training set (e.g., 70-80%), and smaller portions to the validation set (e.g., 10-15%), and a separate portion to the test set (e.g., 10-15%).

How to do Data Division?

- Datasets are divided such that, the highest portion of it is dedicated for proper learning of the models i.e., the largest portion (70-90%) of the dataset becomes train set.
- Remaining data is kept as validation set and test set.

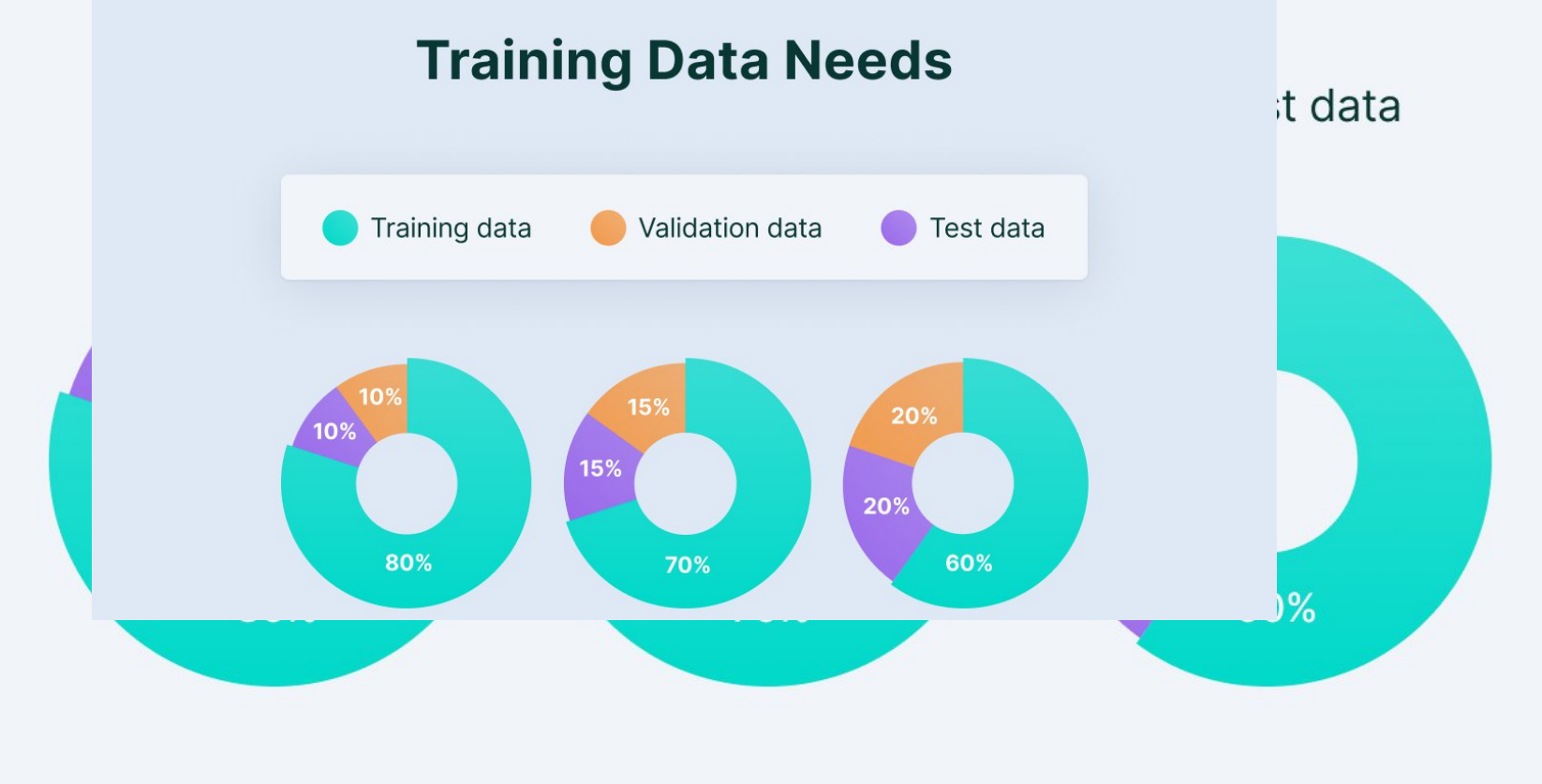


Image Source: https://www.v7labs.com/blog/train-validation-test-set

Facts about Split percentage:

There is no optimal split percentage. It must be a split percentage that suits the requirements and meets the model's needs.

However, there are two major concerns while deciding on the optimum split:

- If there is less training data, the machine learning model will show high variance in training.
- With less testing data/validation data, your model evaluation/model performance statistic will have greater variance.

Essentially, an optimum split that suits the need of the dataset/model is needed.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y)
```

Note: X and y are variables defining independent and dependent variables respectively.
There are other parameters for the train_test_split() function. Check out the source code at this link.

Overfitting : Overfitting occurs when a machine learning model learns not only the underlying patterns in the training data but also the noise and outliers. This results in a model that performs very well on the training dataset but poorly on unseen or test data.

Characteristics:

- High accuracy on the training data.
- Poor generalization to new, unseen data.
- The model is overly complex, with too many parameters relative to the number of observations.

erfitting

Causes:

- A model that is too complex (e.g., a deep neural network with many layers) for the amount of training data available.
- Insufficient training data, leading the model to learn specific details rather than general patterns.
- Too many features in the model, allowing it to fit the noise in the training data.

Solutions:

- **Simplifying the Model:** Use a less complex model with fewer parameters.
- **Cross-Validation:** Use techniques such as k-fold cross-validation to ensure that the model generalizes well. In k-fold cross-validation, splitting the dataset into k subsets or folds, where each fold is used as the validation set in turn while the remaining k-1 folds are used for training
- **Data Augmentation:** Increase the training dataset size with techniques like data augmentation or synthetic data generation.

Underfitting:

Underfitting occurs when a machine learning model is too simple to capture the underlying structure of the data. This results in a model that performs poorly on both the training data and unseen data.

Characteristics:

- Low accuracy on the training data.
- Poor performance on both training and test datasets.
- The model does not capture relevant features or relationships in the data.

Causes:

- A model that is too simple (e.g., a linear model for a nonlinear relationship) relative to the complexity of the data.
- Insufficient training time or iterations, causing the model to not learn adequately.
- Inappropriate feature selection or engineering, leading to the loss of important information.

Solutions:

- **Complexifying the Model:** Choose a more complex model that better captures the data structure (e.g., using polynomial regression instead of linear regression).
- **Feature Engineering:** Include additional relevant features or interactions between features.
- **Increase Training Time:** Allow the model more time to learn from the data by training for more epochs or iterations.
- **Hyperparameter Tuning:** Adjust hyperparameters to improve model performance

What is Cross Validation?

Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data.

It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds.

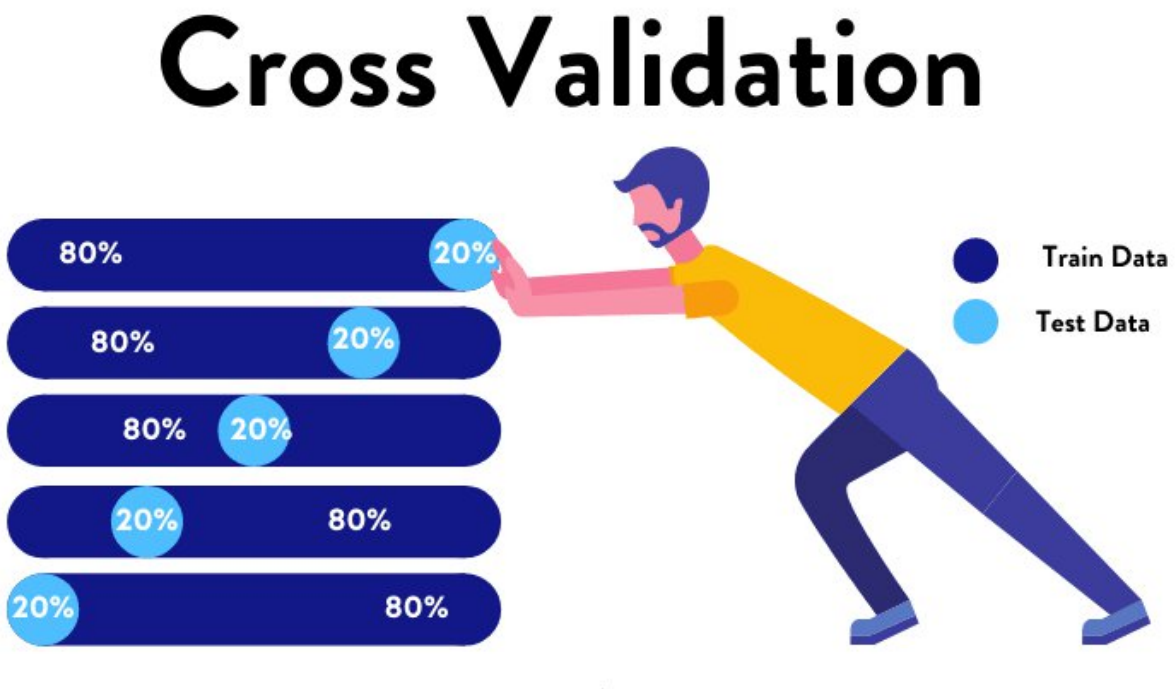
This process is repeated multiple times, each time using a different fold as the validation set.

Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance.

Cross validation is an important step in the machine learning process and helps to ensure that the model selected for deployment is robust and generalizes well to new data.

The **main purpose** of cross validation is to prevent overfitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data.

By evaluating the model on multiple validation sets, **cross validation provides a more realistic estimate of the model's generalization performance**, i.e., its ability to perform well on new, unseen data.

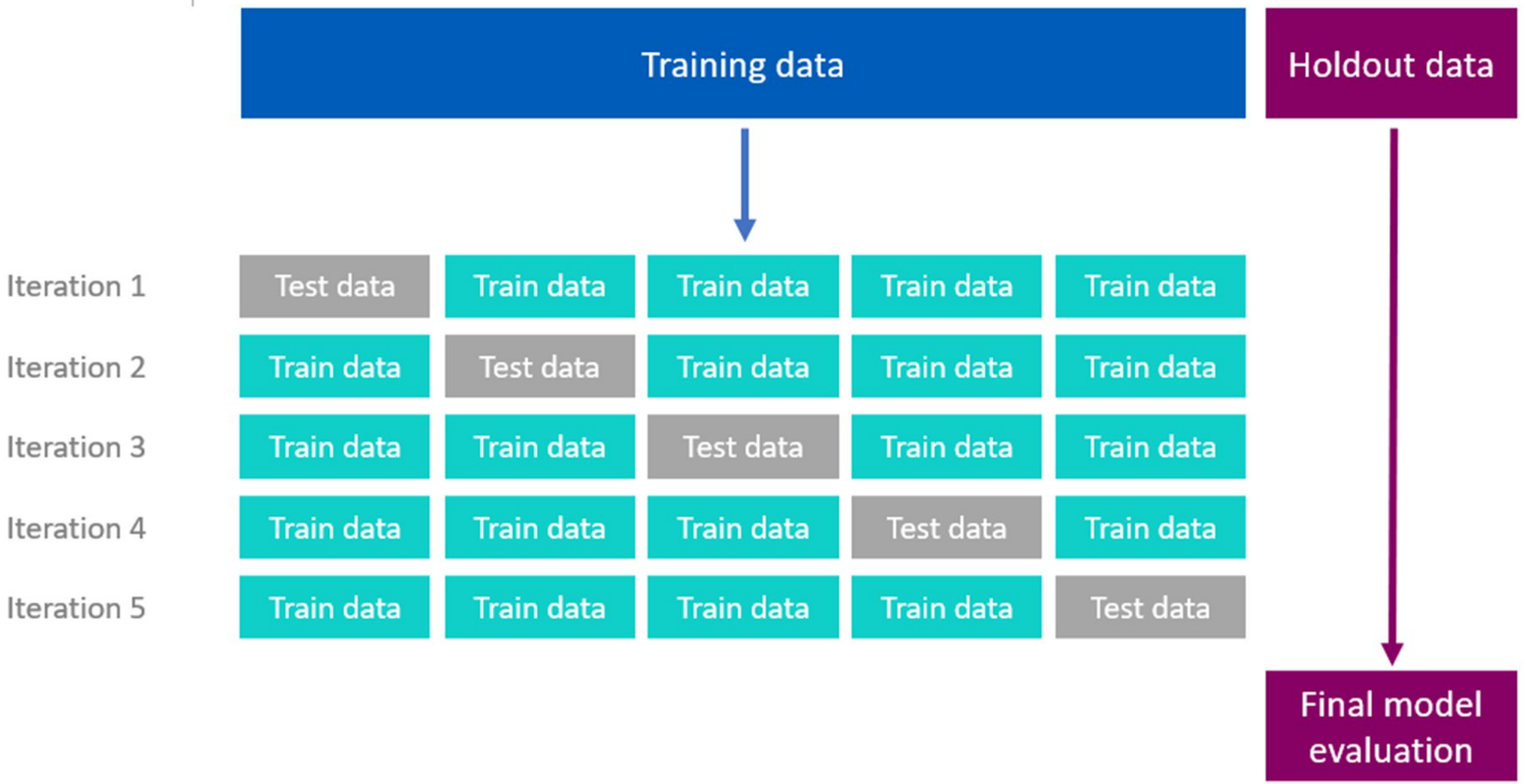


1. Holdout Validation

In Holdout Validation, we perform training on the 50% of the given dataset and rest 50% is used for the testing purpose.

It's a simple and quick way to evaluate a model.

The major drawback of this method is that we perform training on the 50% of the dataset, it may possible that the remaining 50% of the data contains some important information which we are leaving while training our model i.e., higher bias.



services/Subsystems/Hub/Content/Sense_Hub/AutoML/holdout-crossvalidation.htm

2. LOOCV (Leave One Out Cross Validation)

In this method, we perform training on the whole dataset but leaves only one data-point of the available dataset and then iterates for each data-point.

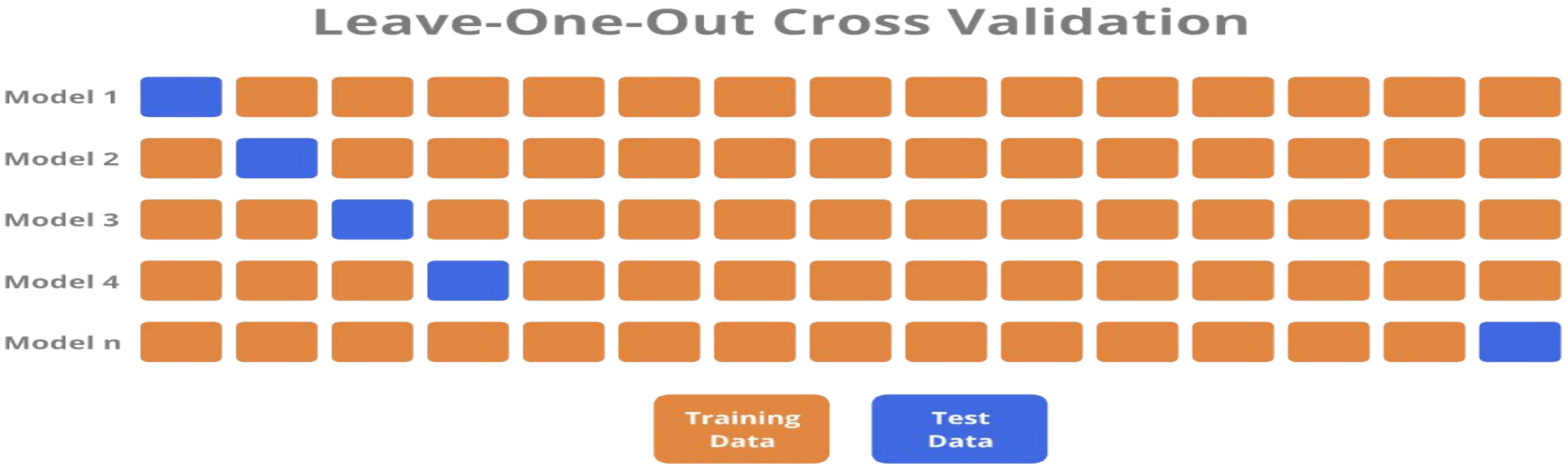
In LOOCV, the model is trained on n-1 samples and tested on the one omitted sample, repeating this process for each data point in the dataset.

An advantage of using this method is that we make use of all data points and hence it is low bias.

The major drawback of this method is that it leads to higher variation in the testing model as we are testing against one data point.

If the data point is an outlier it can lead to higher variation.

Another drawback is it takes a lot of execution time as it iterates over ‘the number of data points’ times.



- Image Source: https://medium.datadriveninvestor.com/leave-one-out-cross-validation-32fa248c1739

UPES
UNIVERSITY OF TOMORROW

3. Stratified Cross-Validation

It is a technique used in machine learning to ensure that each fold of the cross-validation process maintains the same class distribution as the entire dataset.

This is particularly important when dealing with imbalanced datasets, where certain classes may be underrepresented.

The dataset is divided into k folds while maintaining the proportion of classes in each fold.

During each iteration, one-fold is used for testing, and the remaining folds are used for training. The process is repeated k times, with each fold serving as the test set exactly once.

It is essential in dealing with classification problems where maintaining the balance of class distribution is crucial for the model to generalize well to unseen data.

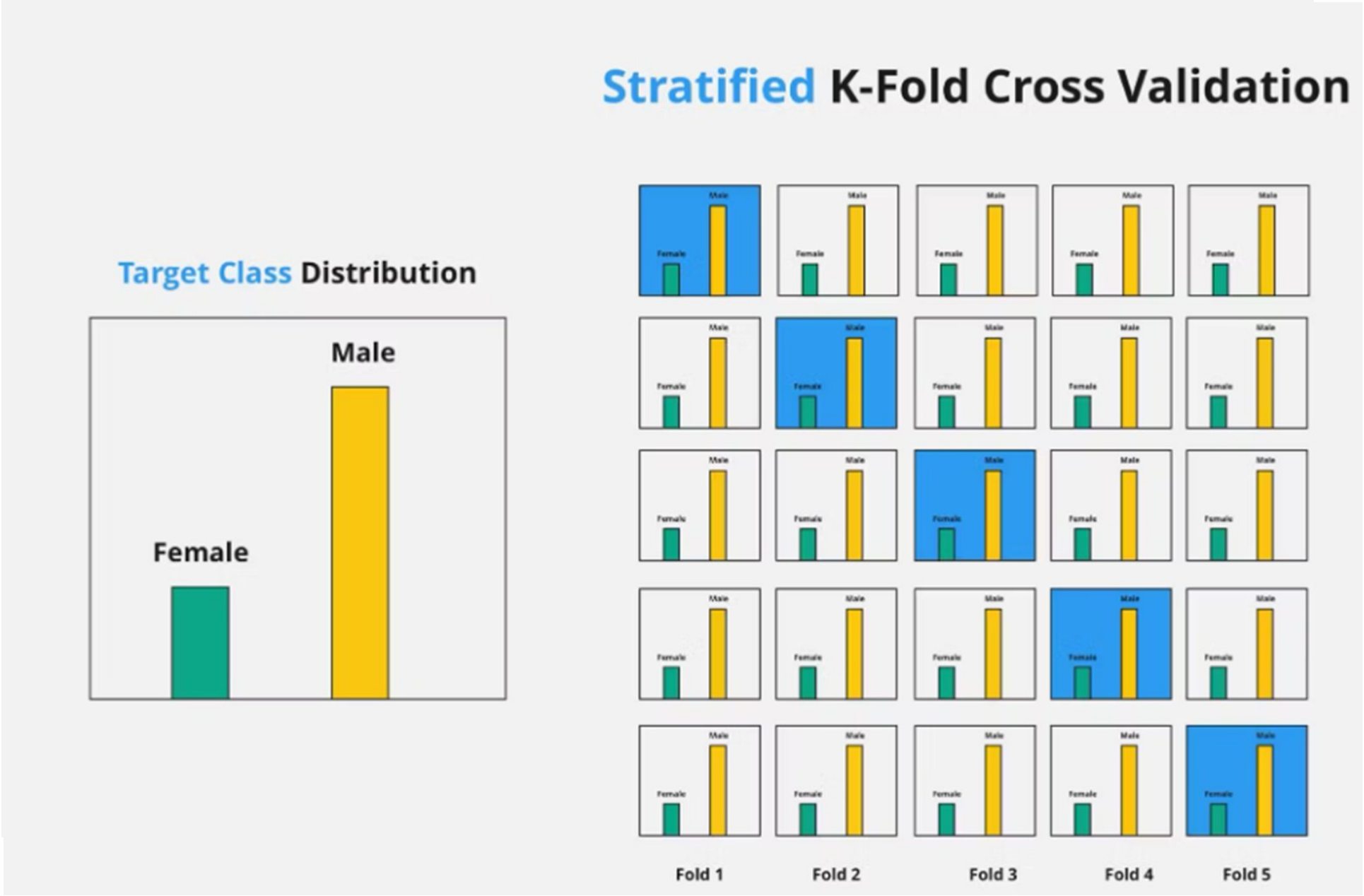
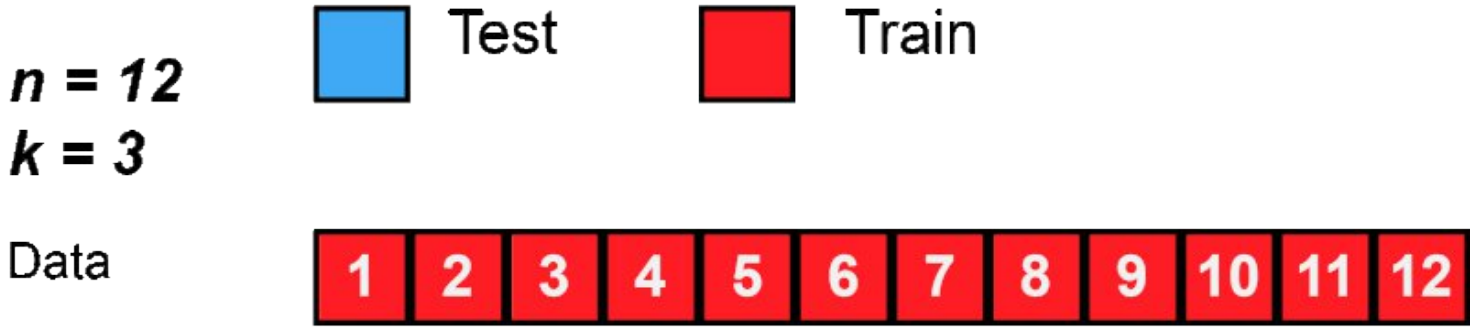


Image Source: https://www.turing.com/kb/different-types-of-cross-validations-in-machine-learning-and-their-explanations

4. K-Fold Cross Validation

In K-Fold Cross Validation, we split the dataset into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model.

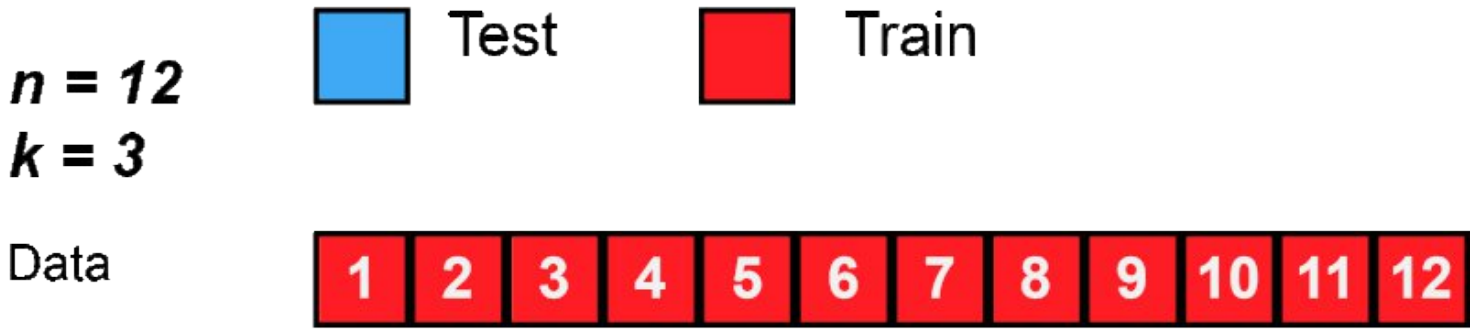
In this method, we iterate k times with a different subset reserved for testing purpose each time.



5. Repeated K-Fold Cross-Validation

This technique involves repeating k-fold cross-validation multiple times with different random splits of the data.

It helps provide a more robust estimate of the model's performance by reducing the variance associated with a single split.



Overcoming Overfitting: Cross validation helps to prevent overfitting by providing a more robust estimate of the model's performance on unseen data.

Model Selection: Cross validation can be used to compare different models and select the one that performs the best on average.

Hyperparameter tuning: Cross validation can be used to optimize the hyperparameters of a model, such as the regularization parameter, by selecting the values that result in the best performance on the validation set.

Data Efficient: Cross validation allows the use of all the available data for both training and validation, making it a more data-efficient method compared to traditional validation techniques.

Computationally Expensive: Cross validation can be computationally expensive, especially when the number of folds is large or when the model is complex and requires a long time to train.

Time-Consuming: Cross validation can be time-consuming, especially when there are many hyperparameters to tune or when multiple models need to be compared.

Bias-Variance Tradeoff: The choice of the number of folds in cross validation can impact the bias-variance tradeoff, i.e., too few folds may result in high variance, while too many folds may result in high bias.

Dimensionality Reduction

Dimensionality reduction techniques are methods used to reduce the number of features (dimensions) in a dataset while retaining as much information as possible.

This is important in machine learning and data analysis for several reasons:

- **Reducing Overfitting:** Fewer dimensions can help to mitigate overfitting, especially in cases where the number of features is large relative to the number of observations.
- **Improving Computational Efficiency:** Reducing dimensionality decreases the amount of computation required, making algorithms faster.
- **Enhancing Visualization:** It allows for visualizing high-dimensional data in 2D or 3D space.
- **Types:**
  - **PCA**
  - **LDA**
  - **ICA**

Principal Component Analysis (PCA)

PCA is a statistical technique that transforms the data into a new coordinate system where the greatest variance by any projection lies on the first coordinate (principal component), the second greatest variance on the second coordinate, and so forth.

Process:

- Standardize the dataset.
- Calculate the covariance matrix.
- Compute the eigenvalues and eigenvectors of the covariance matrix.
- Sort the eigenvalues and select the top k eigenvectors to form a new feature subspace.

Applications: PCA is commonly used in image compression, noise reduction, and exploratory data analysis.

PCA is primarily used for reducing dimensionality while retaining the most variance in the data. It is an unsupervised method that works well with linear relationships and is sensitive to outliers.

Linear Discriminant Analysis (LDA)

LDA is a supervised dimensionality reduction technique used primarily for classification. It aims to find a linear combination of features that best separate two or more classes.

Process:

- Compute the within-class and between-class scatter matrices.
- Calculate the eigenvalues and eigenvectors of the scatter matrices.
- Choose the eigenvectors corresponding to the largest eigenvalues to maximize class separability.

Applications: LDA is often used in face recognition, marketing (customer segmentation), and medical diagnosis.

LDA focuses on maximizing class separability in supervised settings. It is effective for classification tasks, particularly with imbalanced datasets, but it assumes linear separability.

ICA is a computational technique used to separate a multivariate signal into additive, independent components. Unlike PCA, which looks for uncorrelated components, ICA looks for statistically independent components.

Process:

- Center the data (subtract the mean).
- Whiten the data (normalize the variance).
- Apply an iterative algorithm to maximize the statistical independence of the components.

Applications: ICA is widely used in signal processing (e.g., separating audio sources), brain imaging (EEG/MEG), and data compression.

ICA is best suited for applications requiring the separation of independent signals, such as in signal processing. It is an unsupervised technique that does not require class labels but is more complex than PCA.

Thank You