



AI ALGORITHMS WITH MODEL ACCURACY MEASUREMENTS – (Module 3)

Dr. Eric Hitimana
Theoneste Murangira

Agenda

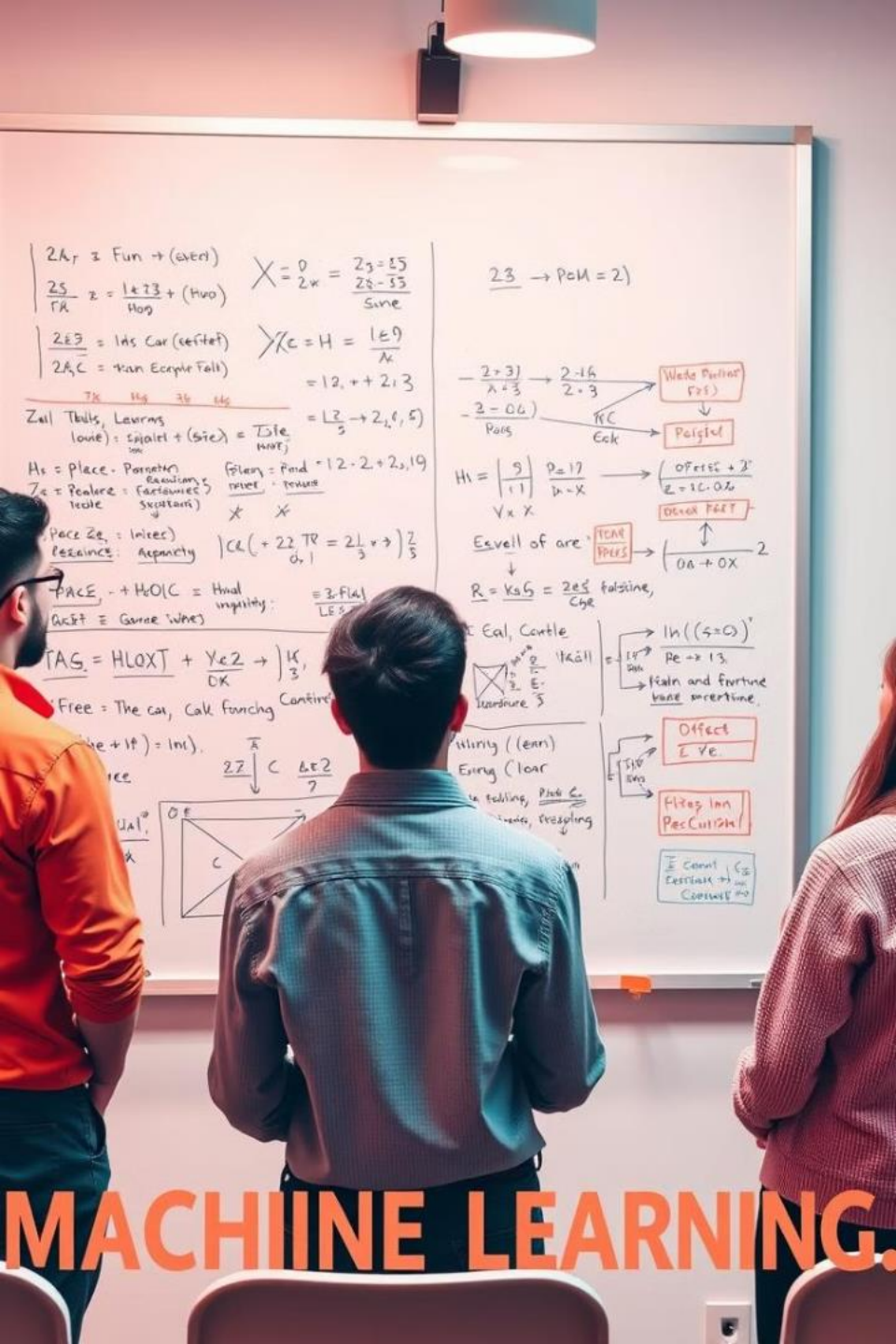
1. Why Machine Learning
2. Types of Machine Learning Systems
3. Types of Model Learning
4. Main challenges of Machine Learning
5. Dataset for Machine Learning
6. Most Common Supervised Algorithms
7. Most Common Unsupervised Algorithms
8. Most common Deep Learning Algorithms
9. Most Common Vision Transformers Algorithms
- 10. Performance Evaluations and Metrics**
- 11. Steps when Modeling**



Machine Learning Landscape

Exploration of the Machine Learning landscape. We'll delve into algorithm types and address common challenges.





Why Use Machine Learning?

1

Rule-Based Approach

- Maintaining a long list of rules for complex problems can be difficult.

2

ML Advantage

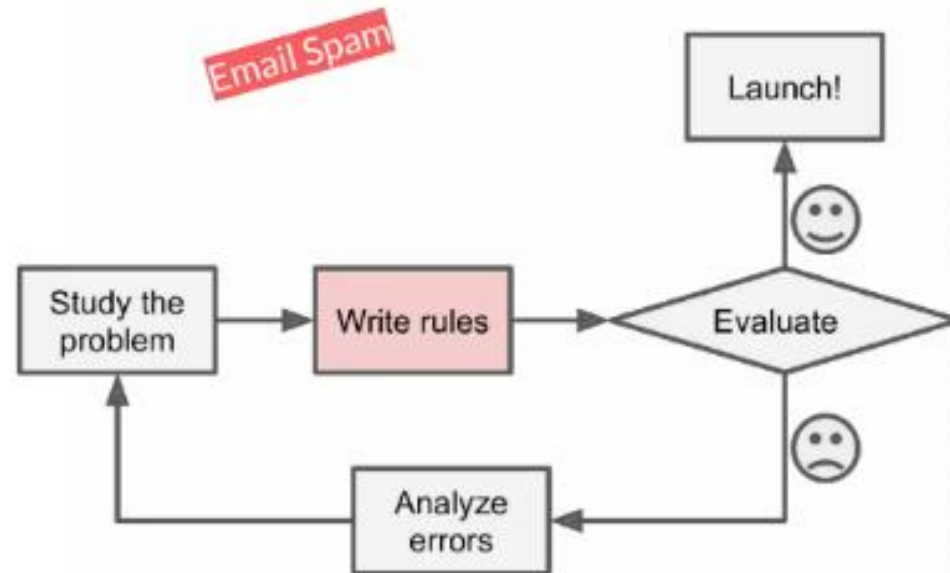
- ML systems can be shorter, easier to maintain, and more accurate.

3

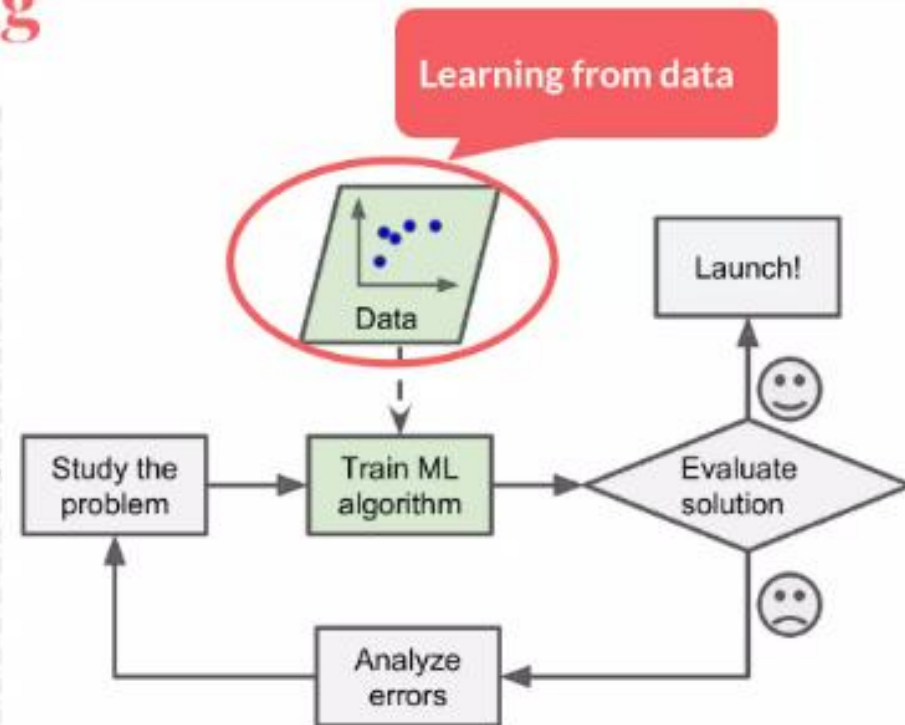
Data Mining

- Training algorithms on large datasets can help understand data relationships.

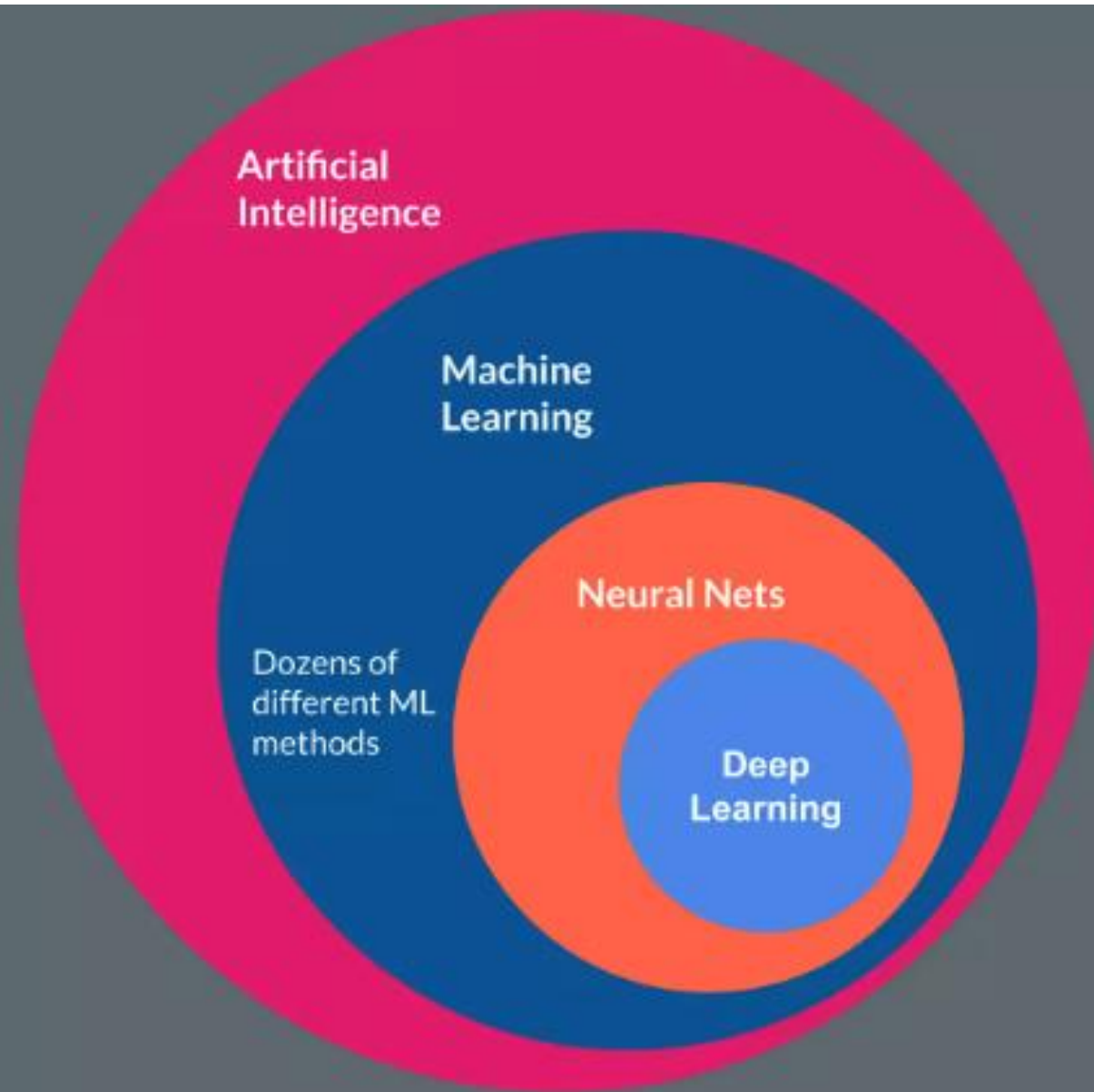
Why use machine learning



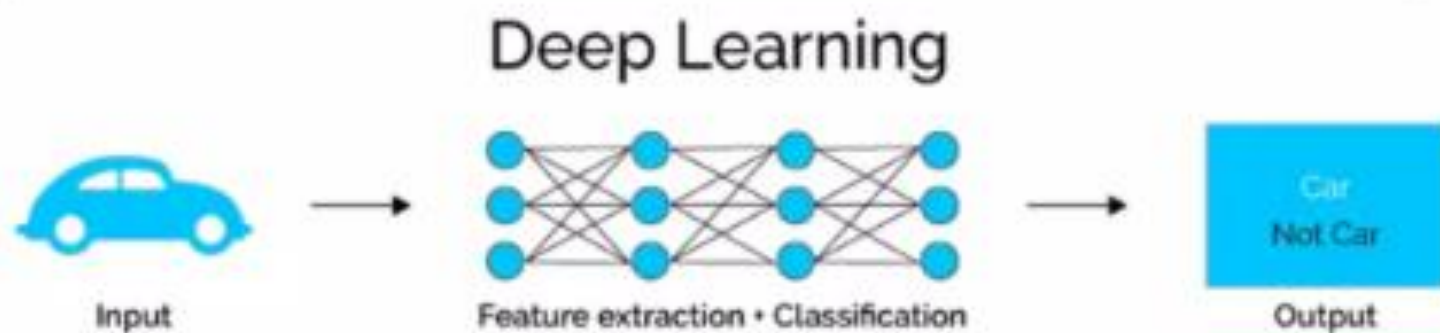
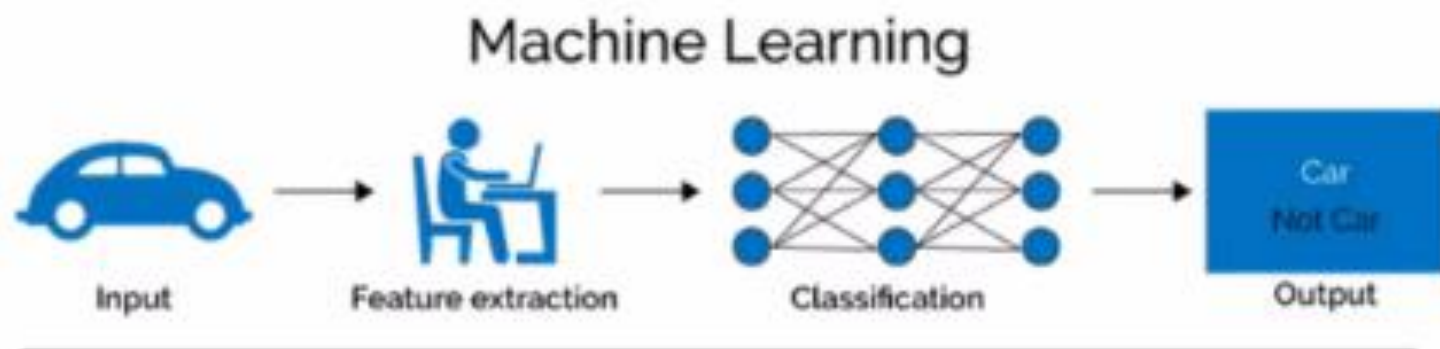
Traditional Approach



Machine Learning Approach



Difference between deep learning and usual ML



Types of Machine Learning Systems

✓ Supervised Learning

- Training data includes labels, used for classification and regression.

🔧 Semi-supervised Learning

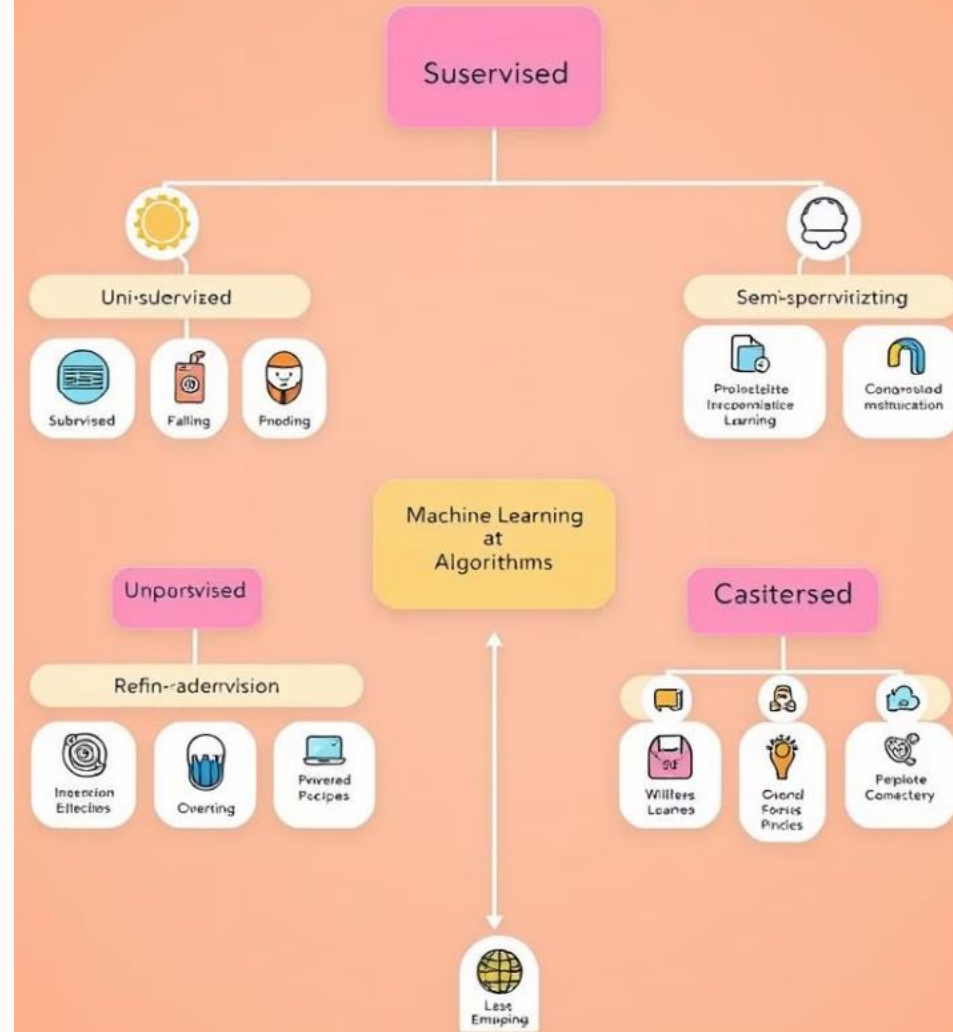
- Utilizing partially labeled data for learning.

? Unsupervised Learning

- Data is unlabeled, finding internal structure within the dataset.

🤖 Reinforcement Learning

- An agent interacts with the environment to learn through rewards.

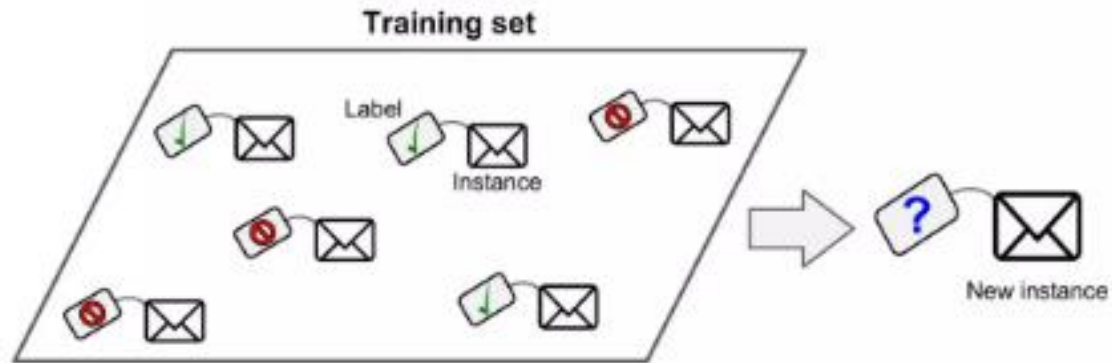


Three Types of Machine Learning

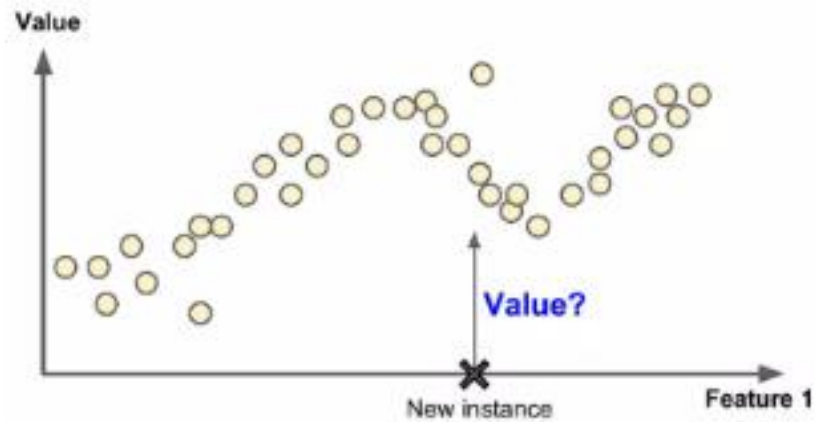
- Supervised learning: classification, regression
- Unsupervised learning: clustering
- Reinforcement learning: chess engine

Supervised Learning	<ul style="list-style-type: none">> Labeled data> Direct feedback> Predict outcome/future
Unsupervised Learning	<ul style="list-style-type: none">> No labels> No feedback> Find hidden structure in data
Reinforcement Learning	<ul style="list-style-type: none">> Decision process> Reward system> Learn series of actions

Supervised Learning



Classification



Regression

Regression / classification models

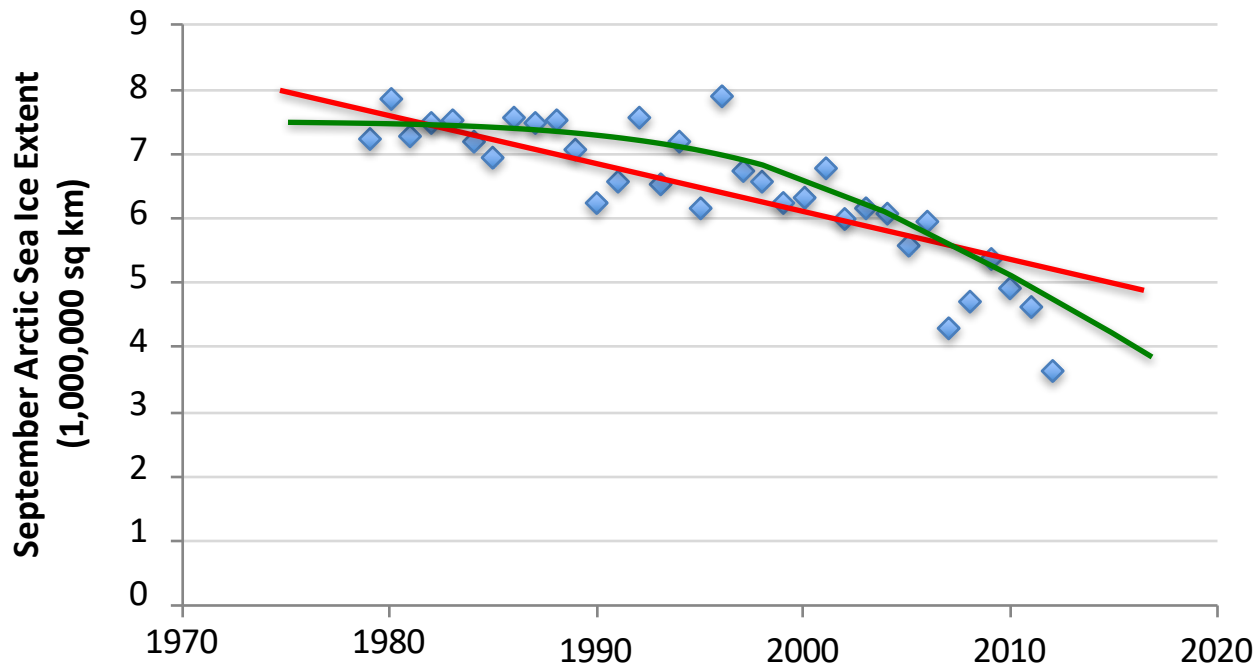
- Predictive modeling / Supervised learning
- A **model** is a specification of mathematical/probabilistic relationships that exist between different variables.
- The goal is usually to use existing data to develop models that we can use to predict outcomes for new data, such as
 - Predicting whether an email message is spam or not
 - Predicting whether a credit card transaction is fraudulent
 - Predicting which advertisement, a shopper is most likely to click on
 - Predicting which football team is going to win the Super Bowl
- Predicting stock price of a given company
- Predicting number of buyers of a certain product
- Predicting user ratings of a new movie
- Predicting the grade of a disease

Nominal
(categorical with
No particular
order)

Continuous / ordinal

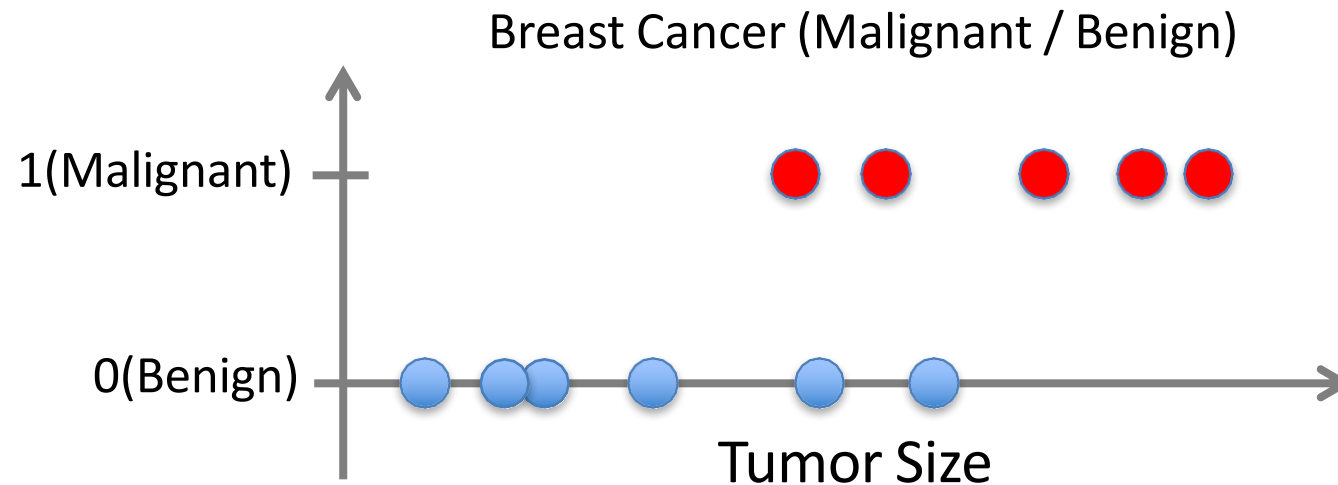
Supervised Learning: Regression

- ❖ Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- ❖ Learn a function $f(x)$ to predict y given x
 - y is **real-valued** == regression

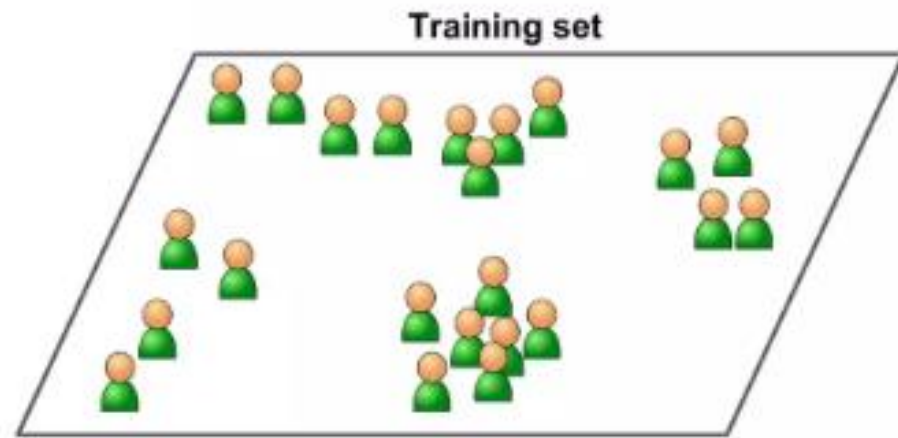


Supervised Learning: Classification

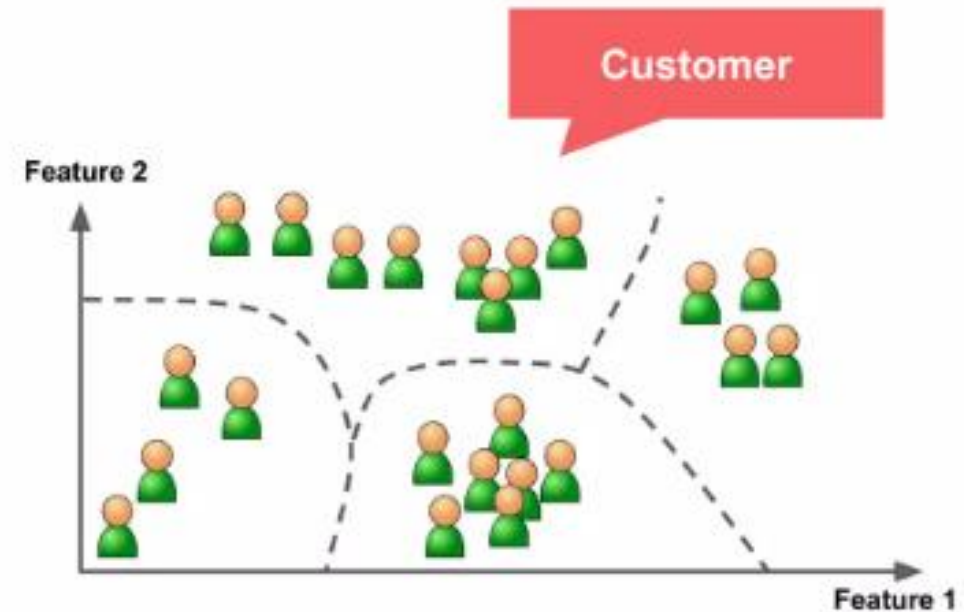
- ❖ Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- ❖ Learn a function $f(x)$ to predict y given x
 - y is **categorical** == classification



Unsupervised Learning

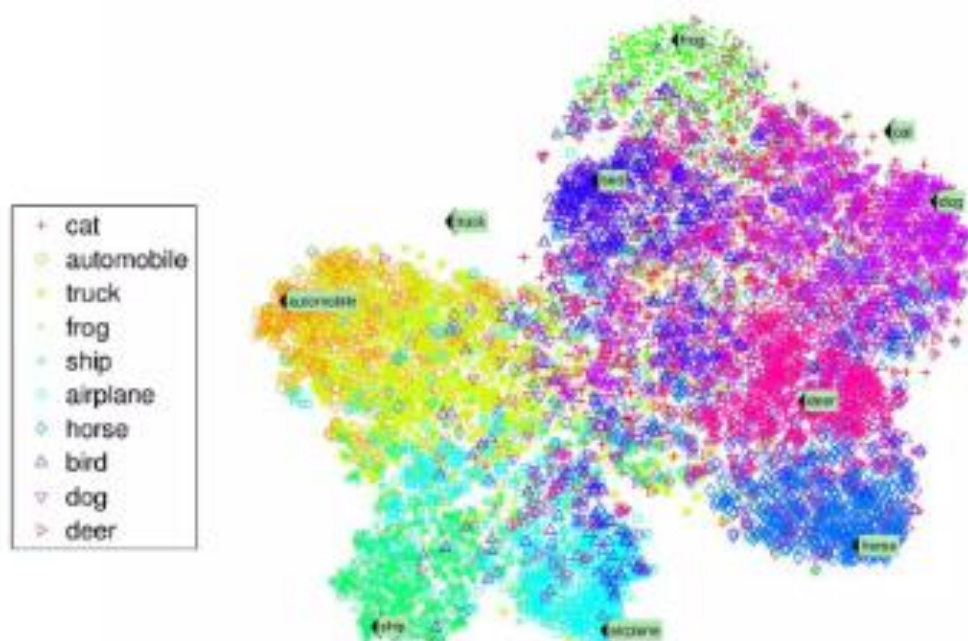


Unlabeled training set

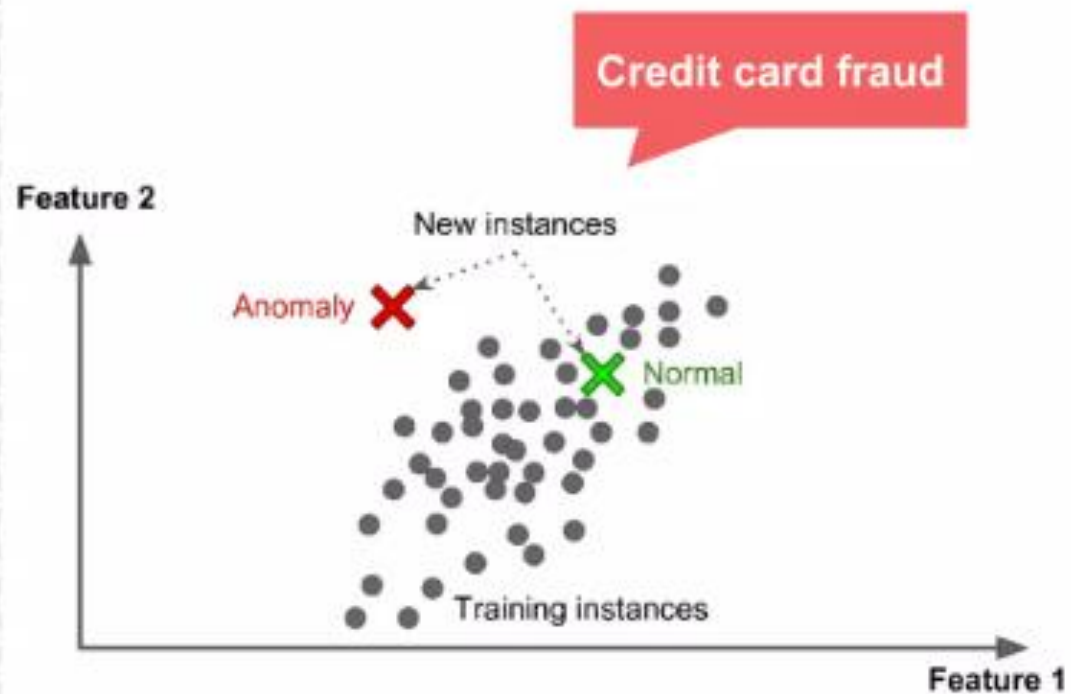


Clustering

Unsupervised Learning

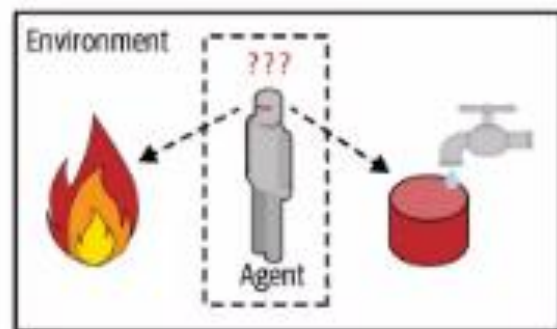


(Visualization) Semantic Clusters



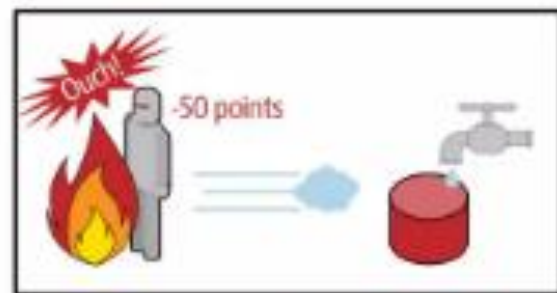
Anomaly Detection

Reinforcement Learning



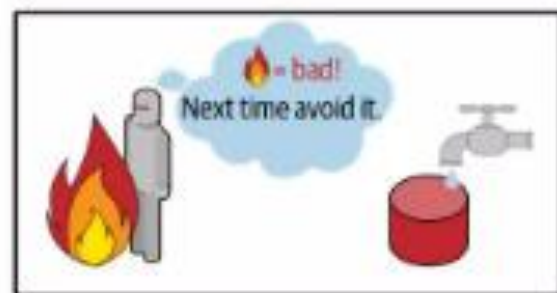
1 Observe

2 Select action using policy



3 Action!

4 Get reward or penalty



5 Update policy (learning step)

6 Iterate until an optimal policy is found

agent



actions

rewards

observations

environment



Learning System

Instance-based vs. Model-based Learning

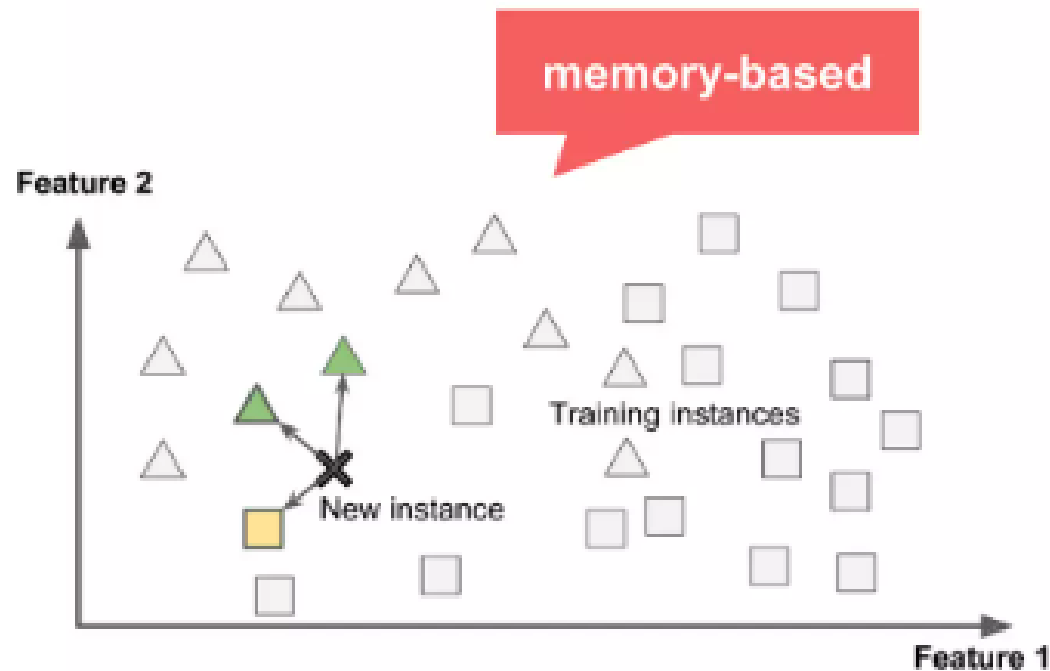
Instance-based Learning

- Classifying based on similarity to the training set.

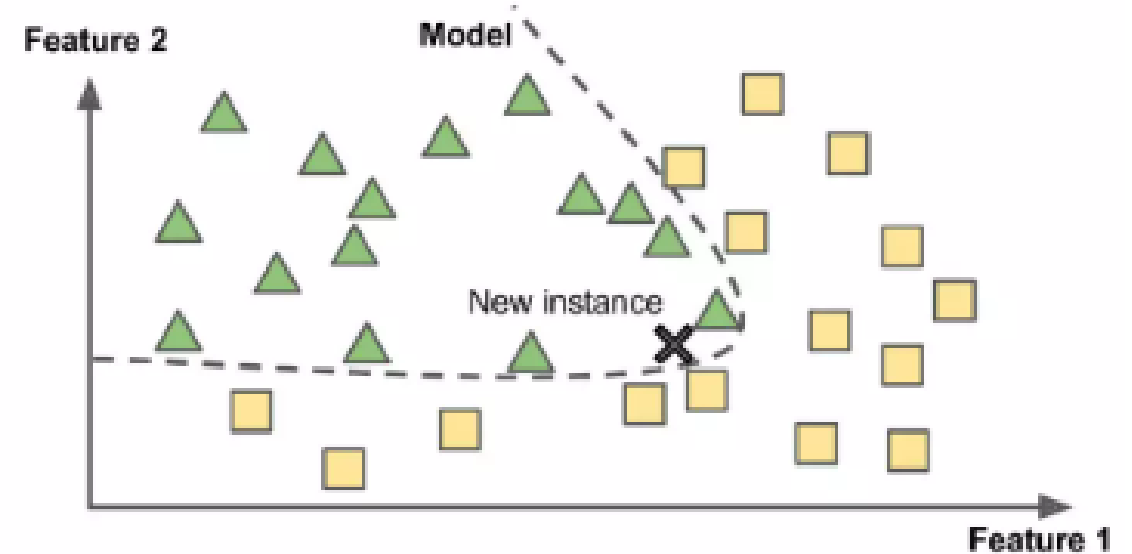
Model-based Learning

- Building models for each class and using them to classify new data.

Instance-based vs Model-based Learning



Instance-based learning



model-based learning



Main Challenges of Machine Learning

1

Data Quality

- Outliers, errors, and noise in data can hinder model performance.

2

Overfitting

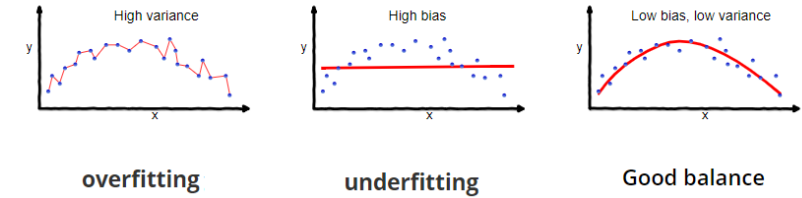
- The model performs well on training data but fails to generalize.

3

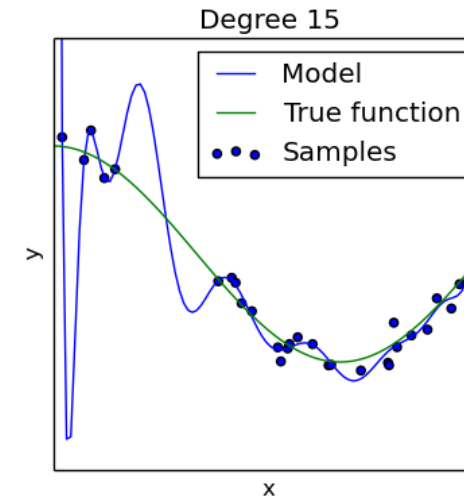
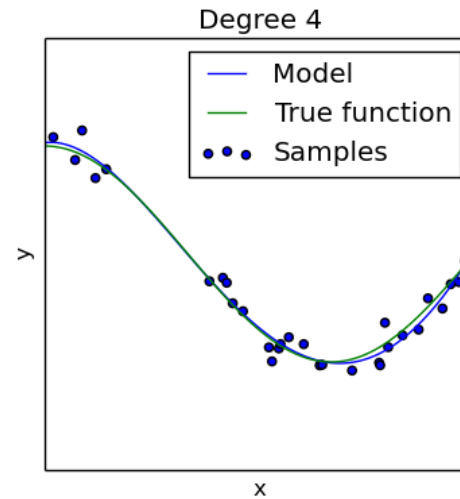
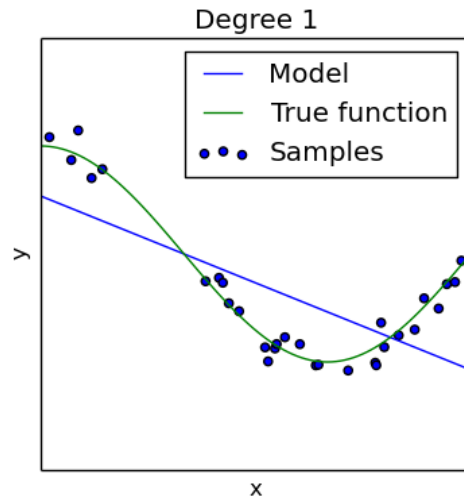
Underfitting

- The model is too simple to capture the underlying data structure.

Overfitting vs underfitting



- **Overfitting** – fitting the training data too precisely - usually leads to poor results on new data.
- **Underfitting** – the model does not fit training data well.



Underfitting

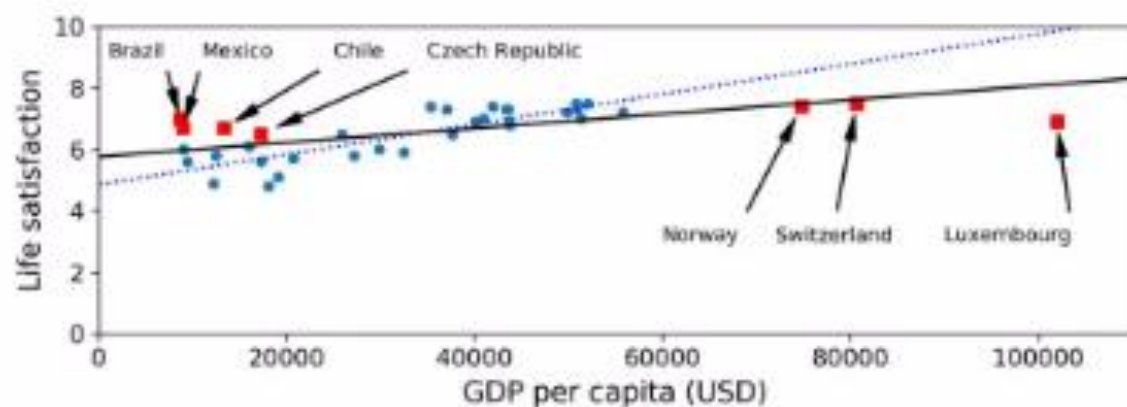
- High bias, low variance
- Increase # of features or complexity of the model

Overfitting

- Low bias, high variance
- Get more training data, or reduce # of features or complexity of the model

Bad Data

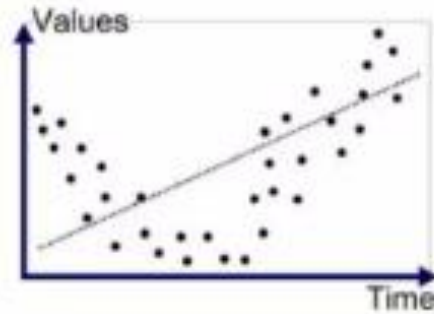
1. Insufficient **Quality** of Training Data
2. **Nonrepresentative** Training Data
 - a. too small → sampling noise
 - b. very large → sampling bias
3. **Poor-Quality** Data
 - a. Missing value
 - b. Outliers
4. **Irrelevant** Features
 - a. Feature Extraction
 - b. Feature Selection



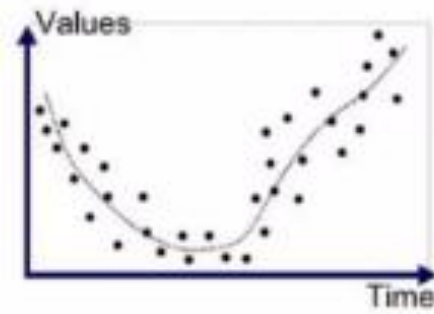
Model Bias

Bad Algorithms

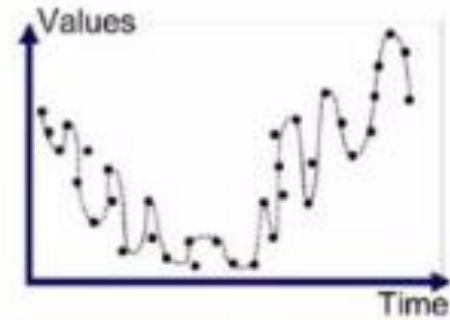
Regression



Underfitted

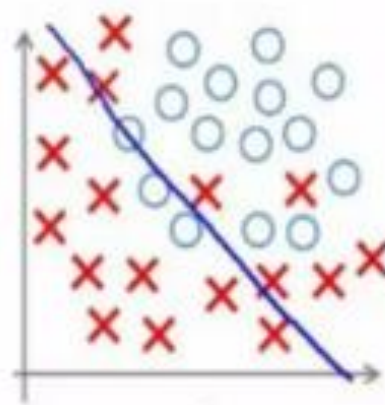


Good Fit/Robust

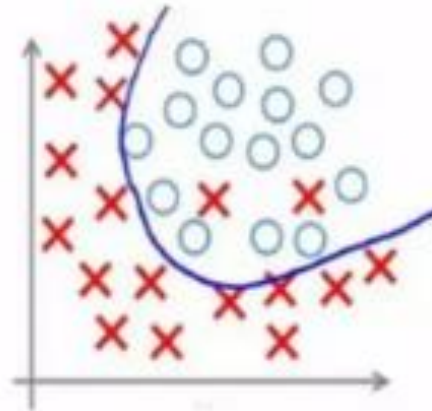


Overfitted

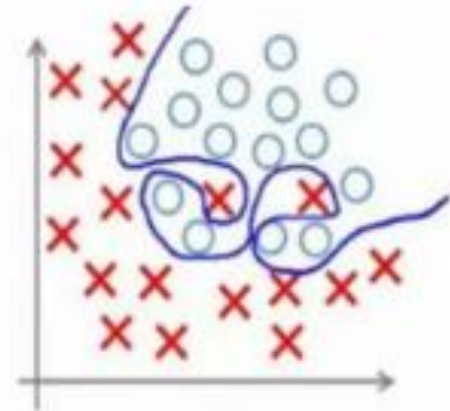
Classification



Under-fitting

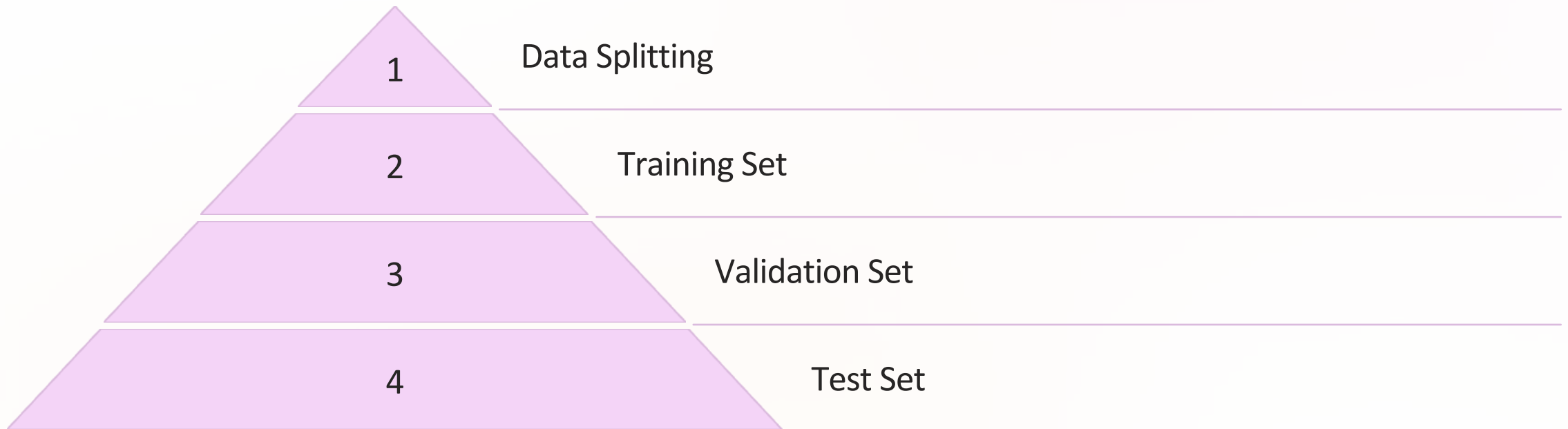


Appropriate-fitting

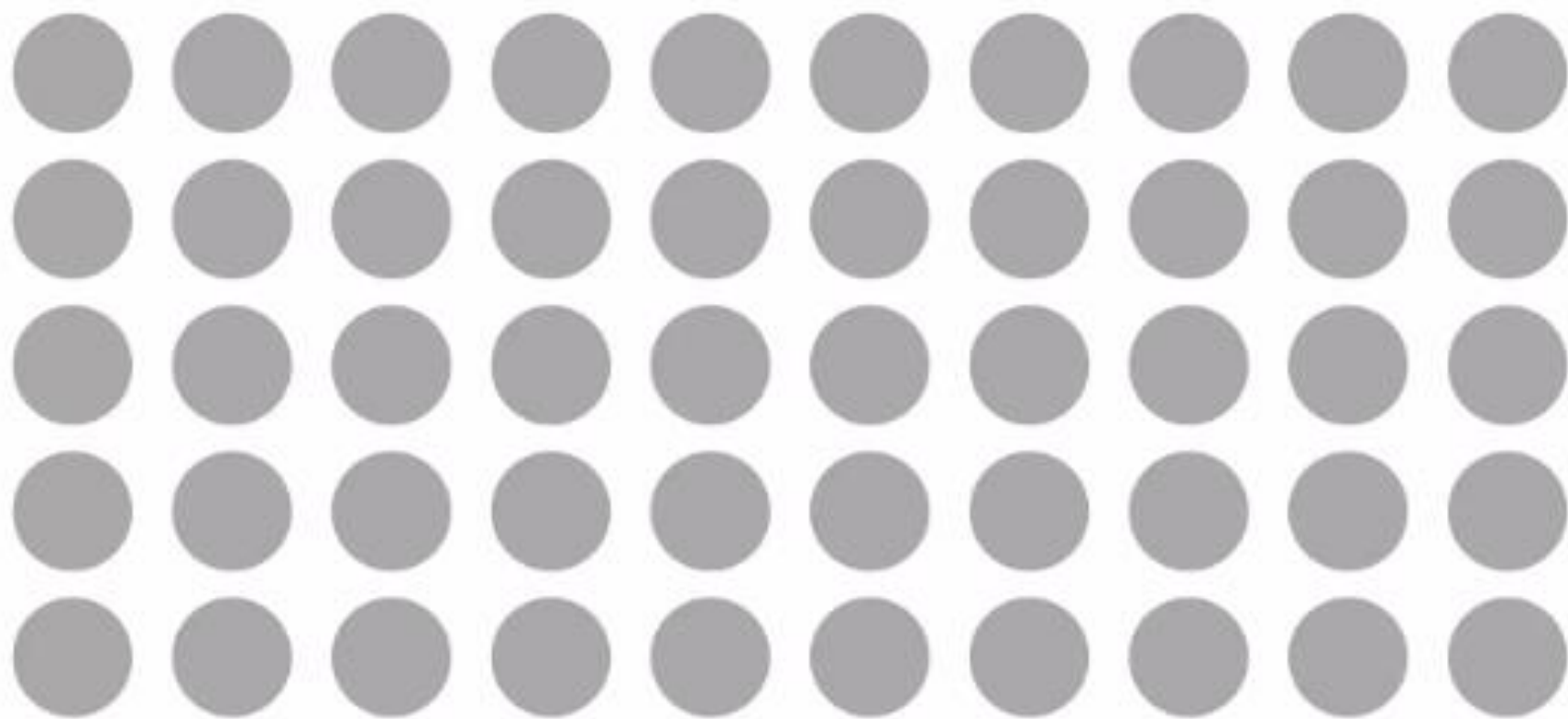


Over-fitting

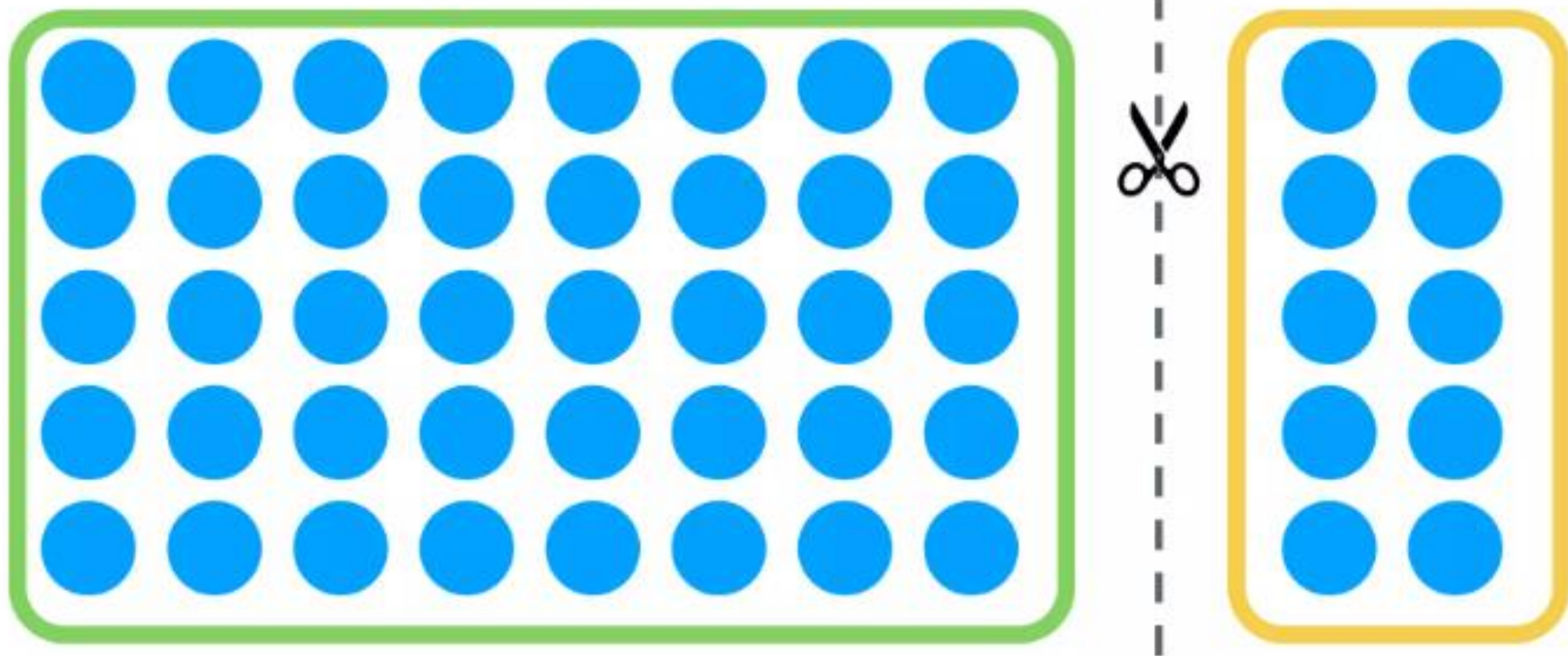
Data in Machine Learning Modeling



Dataset

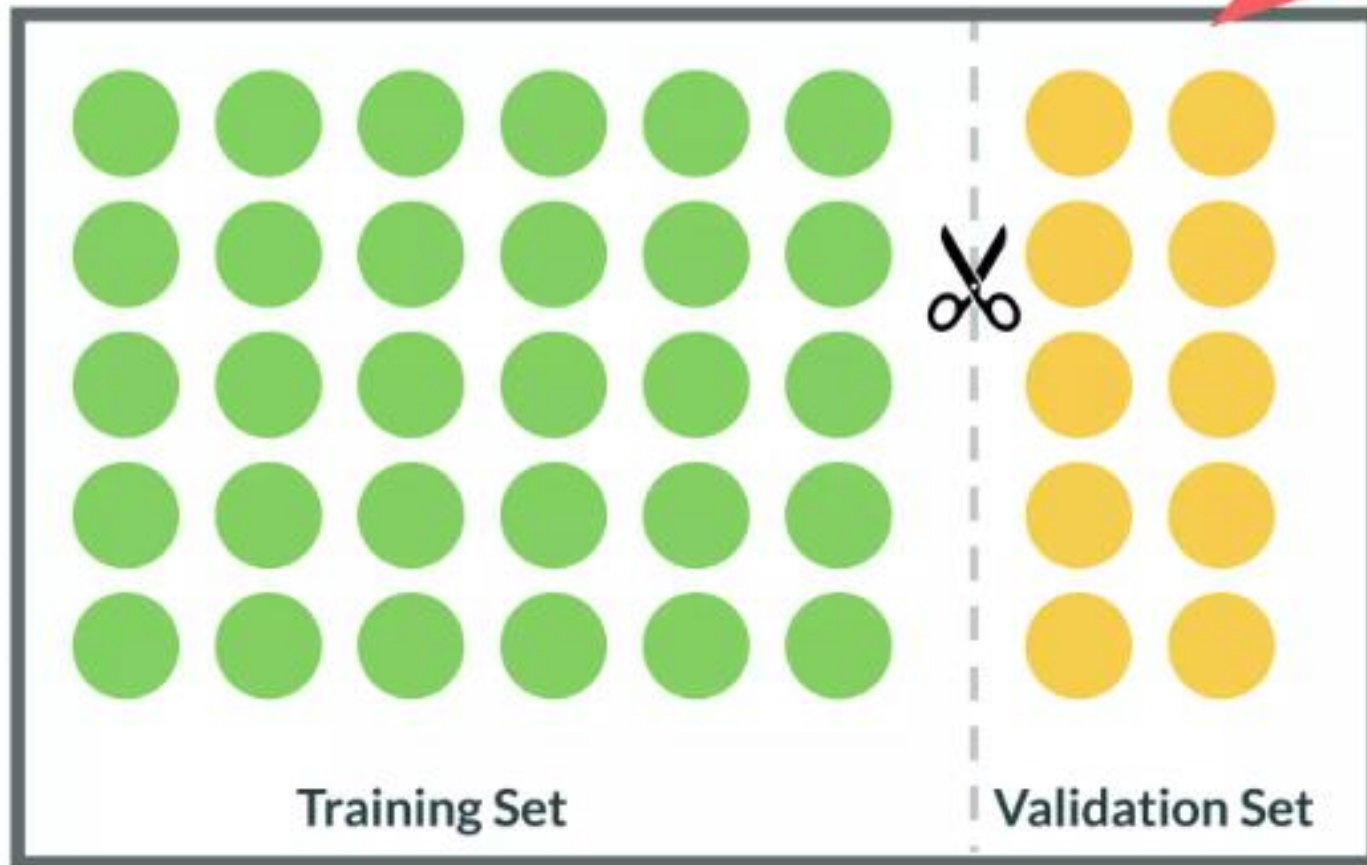


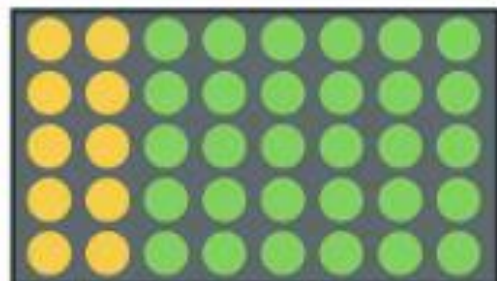
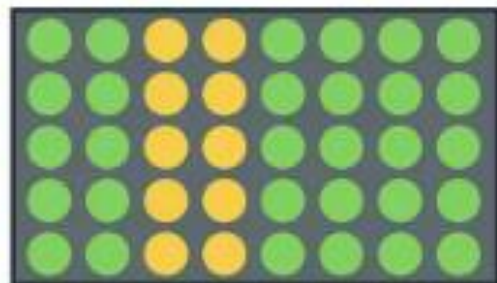
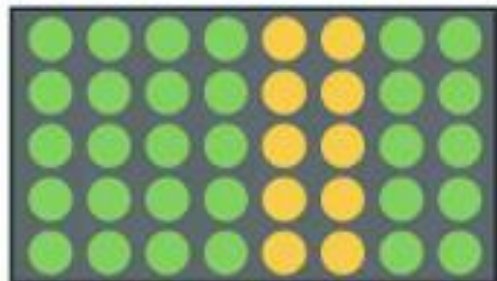
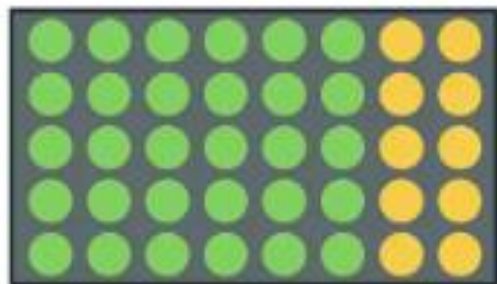
Training Set and Test Set



Holdout Validation

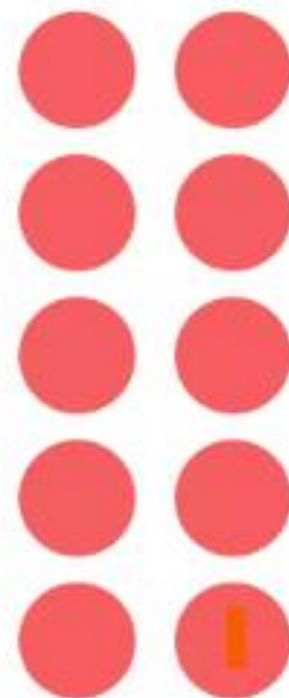
aka development set or dev set





n-fold
cross-validation

- Training Set
- Validation Set



Test Set

Evaluation on “LARGE” data

- If many (thousands) of examples are available, then how can we evaluate our model?
- A simple evaluation is sufficient
 - Randomly split data into training and test sets (e.g. 2/3 for train, 1/3 for test)
 - For classification, make sure training and testing have a similar distribution of class labels
- Build a model using the *train* set and evaluate it using the *test* set.

Evaluation on “SMALL” data

- What if we have a small data set?
- The chosen $2/3$ for training may not be representative.
- The chosen $1/3$ for testing may not be representative.

Cross-validation

- *Cross-validation more useful in small datasets*
 - **First step:** data is split into k subsets of equal size.
 - **Second step:** each subset in turn is used for testing and the remainder for training.
- This is called ***k -fold cross-validation***
- For classification, often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

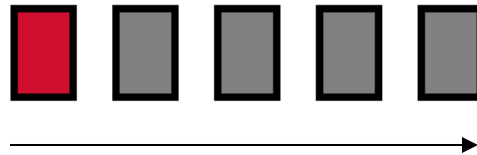
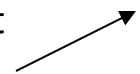
Cross-validation example:

- Break up data into groups of the same size

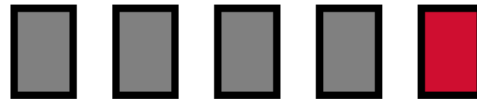


- Hold aside one group for testing and use the rest to build model

Test



- Repeat



More on cross-validation

- Standard method for evaluation: **stratified ten-fold cross-validation**
- **Why ten?** Extensive experiments have shown that this is the best choice to get an accurate estimate
- Stratification reduces the estimate's variance
- **Even better:** repeated stratified cross-validation
 - E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance)

Data Sets

Category	Type	Description	Examples
1. Based on Structure	Structured Data	Organized in rows/columns.	Purchase history, census data, transactions.
	Unstructured Data	Not pre-defined.	Text, images, audio, videos.
	Semi-Structured Data	Some structure but not rigid.	JSON, XML, logs, emails.

2. Based on Modality	Text Data	Text used in NLP tasks.	News, tweets, reviews.
	Image Data	Visual data for recognition and detection.	X-rays, satellite images, faces.
	Audio Data	Sound for speech/emotion recognition.	Podcasts, call recordings, sound effects.
	Video Data	Visual/audio frames for analysis.	Surveillance, movies, sports videos.
	Time-Series Data	Time-indexed for trends and forecasts.	Stock prices, weather, sensor readings.
	Tabular Data	Data in tables (CSV/Excel).	Sales reports, patient records, statistics.

Most common Supervised Algorithms

1. Linear Algorithms:

- Linear Regression
- Logistic Regression

2. Tree-Based Algorithms:

- Decision Trees
- Random Forest
- Gradient Boosting (e.g., XGBoost, LightGBM, CatBoost)

Most common Supervised Algorithms

3. Support Vector Machines (SVM):

- Effective for classification and regression.

4. Nearest Neighbors:

- K-Nearest Neighbors (KNN)

5. Neural Networks:

- Feedforward Neural Networks
- Multi-layer Perceptron (MLP)

6. Bayesian Algorithms:

- Naive Bayes

7. Ensemble Methods:

- AdaBoost
- Bagging
- Stacking

8. Others:

- Ridge and Lasso Regression (for regularized regression tasks)

Most common Unsupervised Algorithms

1. Clustering:

- K-Means Clustering
- Hierarchical Clustering
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

2. Dimensionality Reduction:

- Principal Component Analysis (PCA)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection (UMAP)

Most common Deep Learning Algorithms

1. Convolutional Neural Networks (CNNs):

- Primarily used for image and video analysis tasks.

2. Recurrent Neural Networks (RNNs):

- Variants include LSTMs (Long Short-Term Memory) and GRUs (Gated Recurrent Units), used for sequential data like time series and text.

3. Generative Models:

- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)

4. Transformers:

- Basic Transformer architecture (used in language tasks like GPT, BERT).
- Adaptations for vision tasks (Vision Transformers).

Vision Transformers Algorithms

1. Vision Transformer (ViT):

- The original Vision Transformer model adapts the transformer architecture for image classification tasks.

2. DeiT (Data-efficient Image Transformers):

- A more efficient and robust version of ViT that requires less data for training.

3. Swin Transformer:

- A hierarchical transformer that processes images at different scales, improving efficiency and accuracy for vision tasks.

4. ConvNext:

- Combines transformer and convolutional approaches for image processing.

Model Performance Evaluation

Evaluation

- **Evaluation** = Process of judging the merit or worth of something.
- Evaluation is key to building *effective* and *efficient* Data Science systems
 - Usually carried out in controlled experiments.
 - *Online* testing can also be done

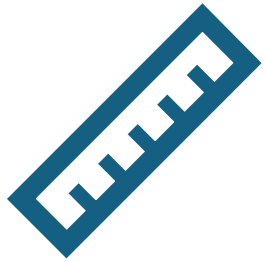
Why System Evaluation?

There are many models/
algorithms/ systems,
which one is the best?

Performance evaluation

- How predictive is the model we learned?
 - For regression, usually R^2 or MSE
 - For classification, many options (discuss later today)
 - Accuracy can be used, with caution
- Performance on the training data (data used to build models) is *not* a good indicator of performance on future data
 - **Q: Why?**
 - A: New data will probably not be **exactly** the same as the training data!

Performance Metrics



- Performance evaluation or measurement for Machine Learning (ML) models involves assessing how well a model performs on a given task.
- This evaluation ensures the model's reliability, accuracy, and generalization to unseen data.

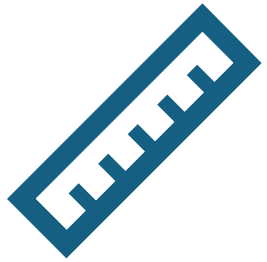


Confusion Matrix

- A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data.
- **True positives (TP)**: These are cases in which we predicted positive (they have the disease), and they do have the disease.
- **True negatives (TN)**: We predicted negative, and they don't have the disease.
- **False positives (FP)**: We predicted positive, but they don't have the disease. (Also known as a "Type I error.")
- **False negatives (FN)**: We predicted negative, but they do have the disease. (Also known as a "Type II error.")

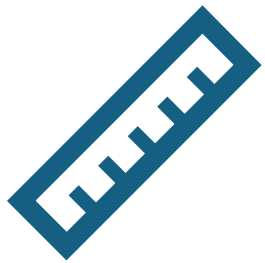
	Actual: Positive	Actual: Negative
Predicted: Positive	tp	fp
Predicted: Negative	fn	tn

a. Classification Models



- **Accuracy:**
- **Precision, Recall, and F1-Score**
- ROC-AUC Score
- Logarithmic Loss (Log Loss)

1. Accuracy



- **Accuracy:** performance metric that measures the proportion of correctly predicted instances (both positive and negative) out of the total number of predictions made by the model.
- Useful when Data are balanced.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Example

Suppose you have a model that classifies 100 instances into "cat" or "not cat." The results are:

- True Positives (TP): 50
- True Negatives (TN): 30
- False Positives (FP): 10
- False Negatives (FN): 10

Accuracy:

$$\text{Accuracy} = \frac{50 + 30}{50 + 30 + 10 + 10} = \frac{80}{100} = 0.8 (80\%)$$

2. Precision, Recall, and F1 Score

- **Precision:** How many predicted positives are true?

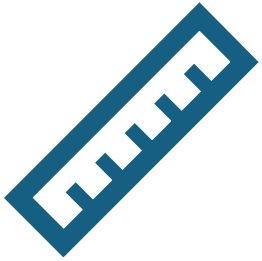
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall (Sensitivity):** How many actual positives are correctly identified?

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1 Score:** Harmonic mean of precision and recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$



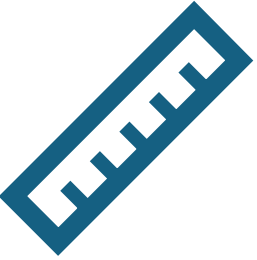
F Measure (F1/Harmonic Mean)

- One measure of performance that takes into account both recall and precision.
- Harmonic mean of recall and precision:

$$F = \frac{1}{\frac{1}{2} \left(\frac{1}{R} + \frac{1}{P} \right)} = \frac{2RP}{(R+P)}$$

- What does harmonic mean?
 - harmonic mean emphasizes the importance of **small values**, whereas the arithmetic mean is affected more by unusually large outliers
 - Data are extremely skewed; over 99% of documents are non-relevant. This is why accuracy is not an appropriate measure
 - Compared to arithmetic mean, both need to be high for harmonic mean to be high.

Example - 1



	Predicted: Positive ("cat")	Predicted: Negative ("not cat")
Actual Positive ("cat")	True Positives (TP): 50	False Negatives (FN): 10
Actual Negative ("not cat")	False Positives (FP): 10	True Negatives (TN): 30

1. Precision

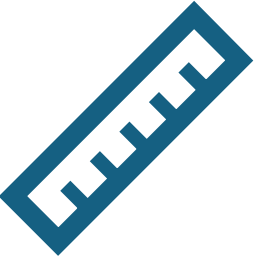
Formula:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Substituting the values:

$$\text{Precision} = \frac{50}{50 + 10} = \frac{50}{60} = 0.8333 \text{ (83.33\%)}$$

Example - 2



	Predicted: Positive ("cat")	Predicted: Negative ("not cat")
Actual Positive ("cat")	True Positives (TP): 50	False Negatives (FN): 10
Actual Negative ("not cat")	False Positives (FP): 10	True Negatives (TN): 30

2. Recall (Sensitivity)

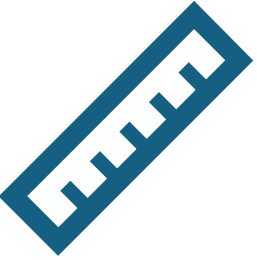
Formula:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Substituting the values:

$$\text{Recall} = \frac{50}{50 + 10} = \frac{50}{60} = 0.8333 \text{ (83.33\%)}$$

Example - 3



	Predicted: Positive ("cat")	Predicted: Negative ("not cat")
Actual Positive ("cat")	True Positives (TP): 50	False Negatives (FN): 10
Actual Negative ("not cat")	False Positives (FP): 10	True Negatives (TN): 30

3. F1 Score

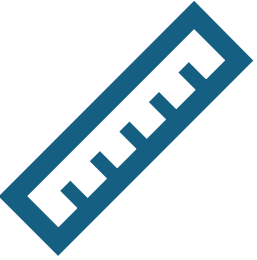
Formula:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Substituting the values:

$$\text{F1 Score} = 2 \cdot \frac{0.8333 \cdot 0.8333}{0.8333 + 0.8333} = 2 \cdot \frac{0.6944}{1.6666} = 0.8333 \text{ (83.33\%)}$$

Example - 4

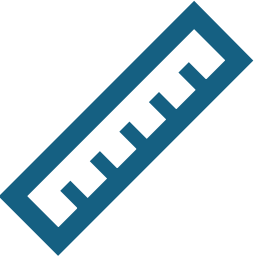


	Predicted: Positive ("cat")	Predicted: Negative ("not cat")
Actual Positive ("cat")	True Positives (TP): 50	False Negatives (FN): 10
Actual Negative ("not cat")	False Positives (FP): 10	True Negatives (TN): 30

Interpretation

- **Precision:** Of all instances predicted as "cat," 83.33% are actually cats.
- **Recall:** Of all actual "cat" instances, 83.33% were correctly identified by the model.
- **F1 Score:** Combines Precision and Recall into a single score, useful when there's a trade-off between the two metrics.

Example - 4

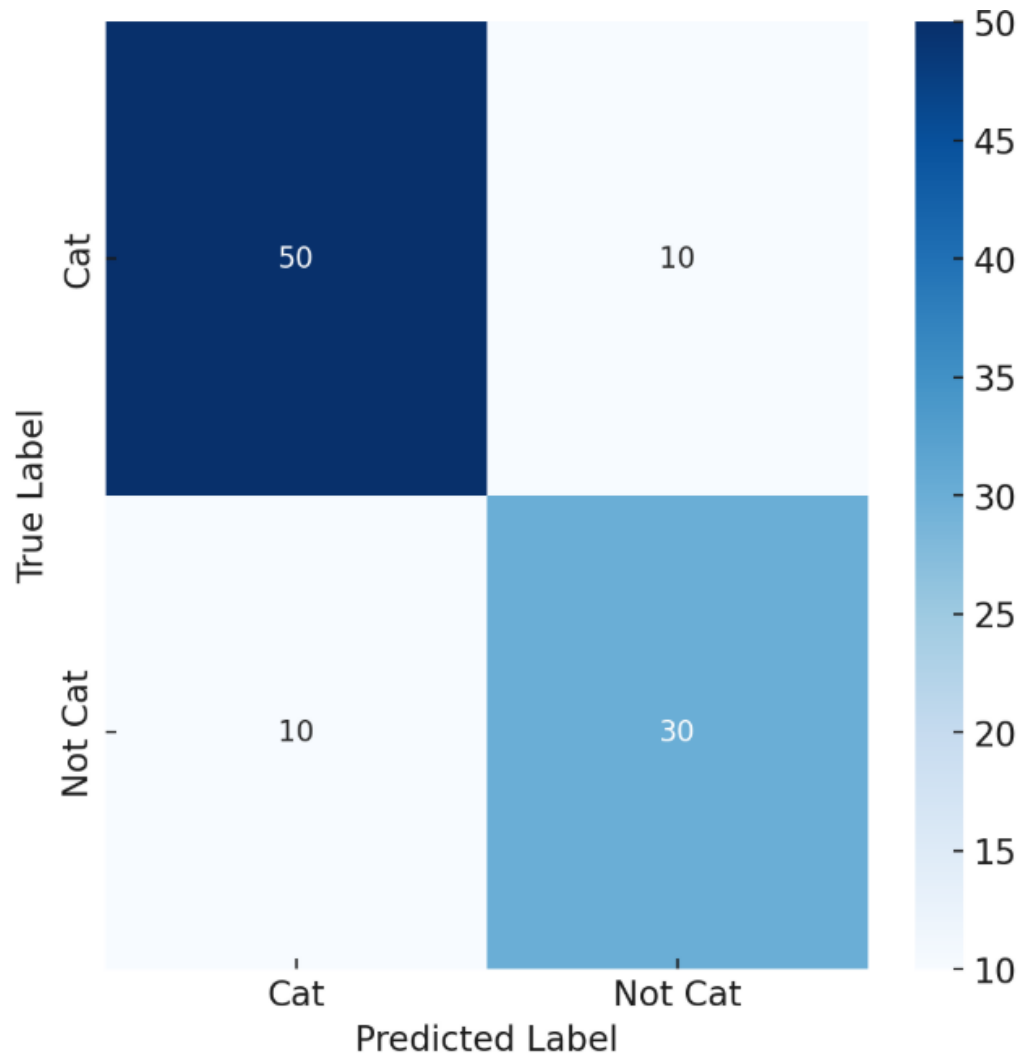


	Predicted: Positive ("cat")	Predicted: Negative ("not cat")
Actual Positive ("cat")	True Positives (TP): 50	False Negatives (FN): 10
Actual Negative ("not cat")	False Positives (FP): 10	True Negatives (TN): 30

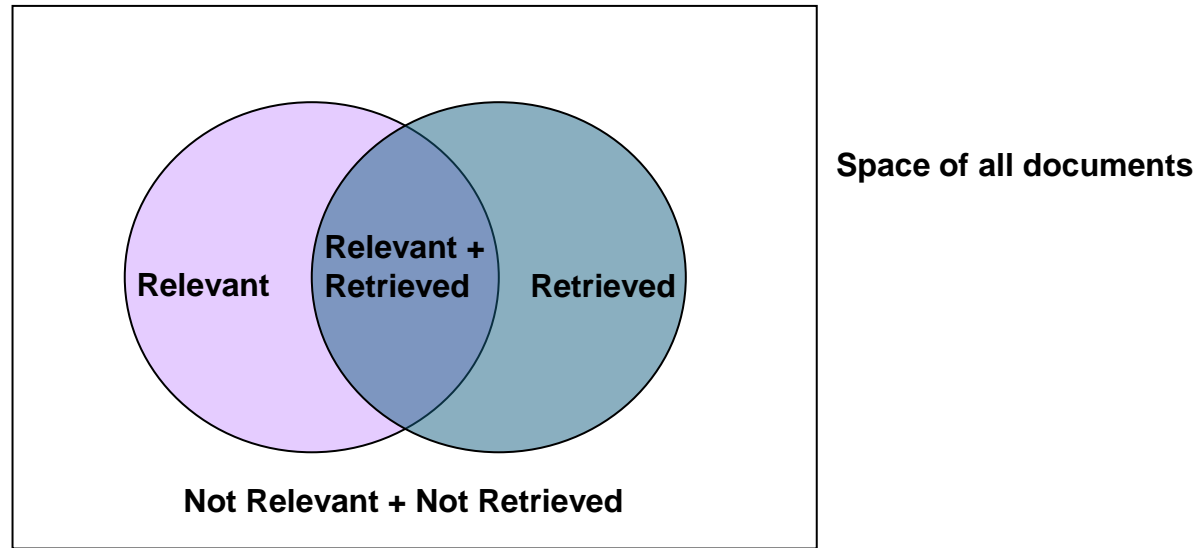
Interpretation

- **Precision:** Of all instances predicted as "cat," 83.33% are actually cats.
- **Recall:** Of all actual "cat" instances, 83.33% were correctly identified by the model.
- **F1 Score:** Combines Precision and Recall into a single score, useful when there's a trade-off between the two metrics.

Example – Confusion Matrix



More example on searching scenario



$$recall = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}$$

$$precision = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$

Precision and Recall in Text Retrieval

- **Precision**

- The ability to retrieve top-ranked documents that are mostly relevant.
- Precision $P = tp / (tp + fp)$

- **Recall**











- The ability of the search to find *all* of the relevant items in the corpus.
- Recall $R = tp / (tp + fn)$

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

Precision/Recall : Example



Ranking #1

										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

$$\text{Recall} = 2/6 = 0.33$$

$$\text{Precision} = 2/3 = 0.67$$


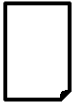





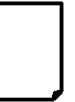
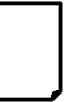

$$\text{Precision} = \frac{\text{Relevant Retrieved}}{\text{Retrieved}}$$

$$\text{Recall} = \frac{\text{Relevant Retrieved}}{\text{Relevant}}$$

Precision/Recall : Example



Ranking #1

										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

$$\text{Recall} = 5/6 = 0.83$$

$$\text{Precision} = 5/6 = 0.83$$

$$\text{Precision} = \frac{\text{Relevant Retrieved}}{\text{Retrieved}}$$

$$\text{Recall} = \frac{\text{Relevant Retrieved}}{\text{Relevant}}$$

Accuracy

- Overall, how often is the classifier correct?
 - Number of correct predictions / Total number of predictions
 - **Accuracy** = $\text{tp} + \text{tn} / (\text{tp} + \text{fp} + \text{fn} + \text{tn})$

	Positive	Negative
Predicted Positive	1	1
Predicted Negative	8	90

- **Accuracy** = $1 + 90 / (1 + 1 + 8 + 90) = 0.91$
- 91 correct predictions out of 100 total examples
- Precision = $1/2$ and Recall = $1/9$
- Accuracy alone doesn't tell the full story when you're working with a **class-imbalanced data set**

Activity 15

	Relevant	Nonrelevant
Retrieved	tp = ?	fp = ?
Not Retrieved	fn = ?	tn = ?












Accuracy of a retrieval model is defined by,

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}$$












Calculate the tp, fp, fn, tn and accuracy for Ranking algorithm #1 and #2 for the highlighted location in the ranking.

 = the relevant documents

Ranking #1

											
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.5	0.6











Ranking #2

											
Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.56	0.6

F Measure (F1/Harmonic Mean) : example

 = the relevant documents

Ranking #1

										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

$$\text{Recall} = 2/6 = 0.33$$

$$\text{Precision} = 2/3 = 0.67$$


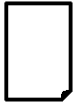





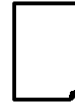
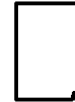

$$F = 2 * \text{Recall} * \text{Precision} / (\text{Recall} + \text{Precision})$$

$$= 2 * 0.33 * 0.67 / (0.33 + 0.67) = 0.44$$

F Measure (F1/Harmonic Mean) : example

 = the relevant documents

Ranking #1

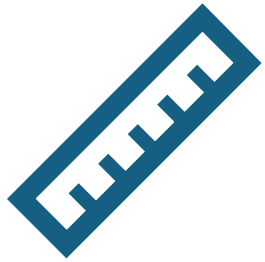
										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

$$\text{Recall} = 5/6 = 0.83$$

$$\text{Precision} = 5/6 = 0.83$$

$$\begin{aligned} F &= 2 * \text{Recall} * \text{Precision} / (\text{Recall} + \text{Precision}) \\ &= 2 * 0.83 * 0.83 / (0.83 + 0.83) = 0.83 \end{aligned}$$

b. Regression Models

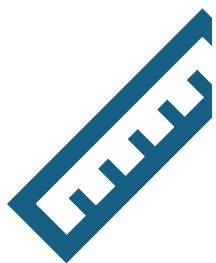


- **Mean Absolute Error (MAE)**
- **Mean Square Error (MSE)**
- Root Mean Squared Error (RMSE)
- R-Squared (R^2)
- Mean Absolute Percentage Error (MAPE)

1. Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- **Definition:** Measures the average absolute difference between the actual values (y_i) and predicted values (\hat{y}_i).
- **Key Insight:** It gives equal weight to all errors, regardless of their magnitude.

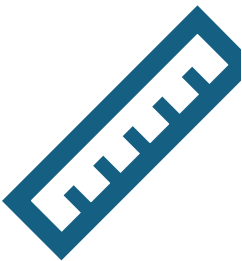


1. Mean Absolute Error (MAE)

Example:

Predicted values (\hat{y}_i): [3, 5, 2.5, 7]

Actual values (y_i): [3, 4.5, 2, 8]


$$\text{MAE} = \frac{|3 - 3| + |5 - 4.5| + |2.5 - 2| + |7 - 8|}{4} = \frac{0 + 0.5 + 0.5 + 1}{4} = 0.5$$

Interpretation of MAE = 0.5

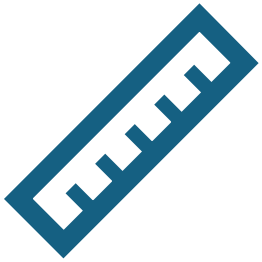
The **Mean Absolute Error (MAE)** of 0.5 indicates that, on average, the predicted values differ from the actual values by **0.5 units**. This is the average magnitude of errors between the predicted and actual values without considering the direction (whether over- or under-predicted).

- The lower the **Mean Absolute Error (MAE)**, the **better the model's performance** in terms of accuracy. Specifically, a lower MAE indicates that the model's predictions are **closer to the actual values** on average.

2. Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **Definition:** Measures the average of squared differences between actual and predicted values.
- **Key Insight:** Penalizes larger errors more than smaller ones due to squaring.

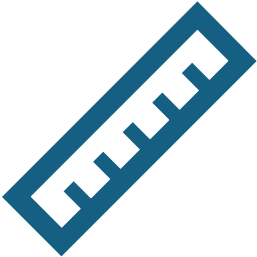


2. Mean Squared Error (MSE)

Example:

Predicted values (\hat{y}_i): [3, 5, 2.5, 7]

Actual values (y_i): [3, 4.5, 2, 8]

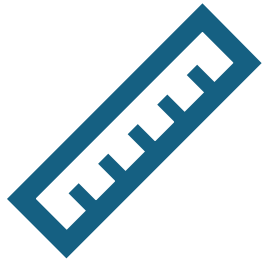

$$\text{MSE} = \frac{(3 - 3)^2 + (5 - 4.5)^2 + (2.5 - 2)^2 + (7 - 8)^2}{4} = \frac{0 + 0.25 + 0.25 + 1}{4} = 0.375$$

Interpretation of **MSE = 0.375**

The **Mean Squared Error (MSE)** of **0.375** represents the average of the squared differences between the predicted and actual values.

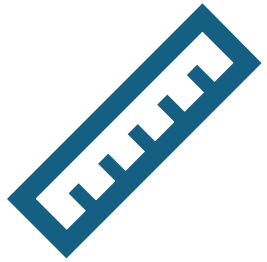
- When comparing models, a **lower MSE** typically indicates a better-performing model.

b. Clustering Models

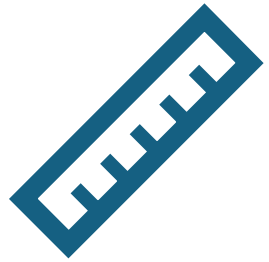


- Silhouette Score
- Davies-Bouldin Index
- Adjusted Rand Index (ARI)

c. Time Series Models

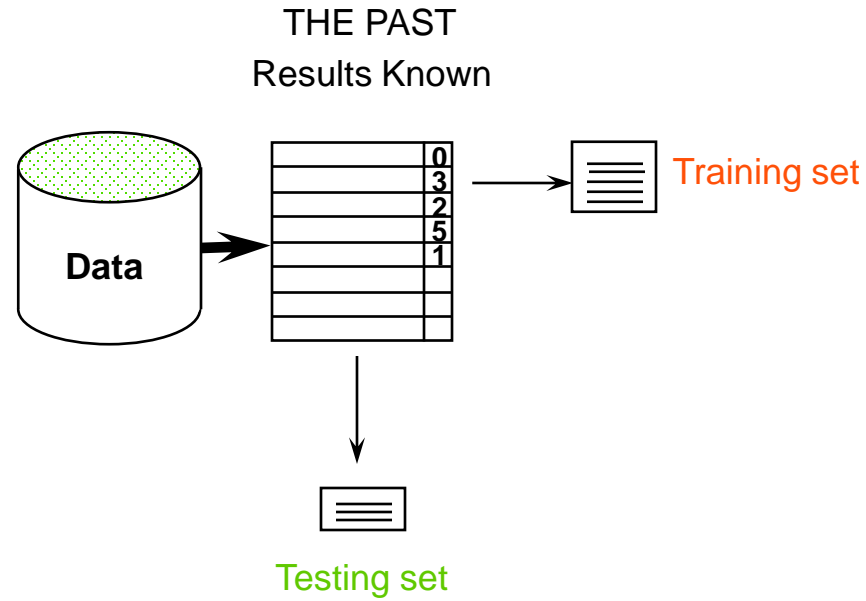


- Mean Absolute Scaled Error (MASE)
- Symmetric Mean Absolute Percentage Error (SMAPE)
- Dynamic Time Warping (DTW)

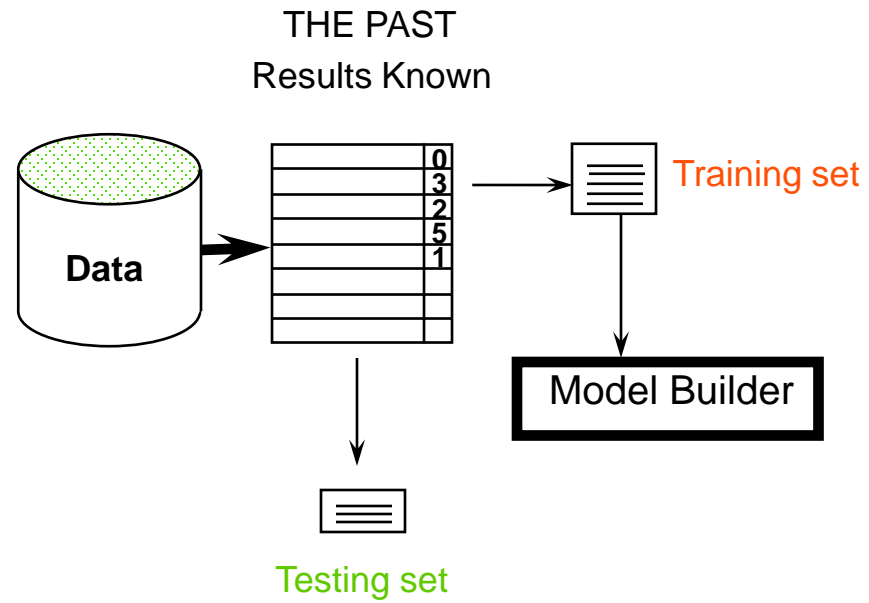


Steps When Modeling

Model Evaluation Step 1: Split data into train and test sets



Model Evaluation Step 2: Build a model on a training set



Model Evaluation Step 3: Evaluate the test set

