CS 559: Overview of Machine Learning

Fall 21 - Lecture 2

In Jang

ijang@stevens.edu

Outline

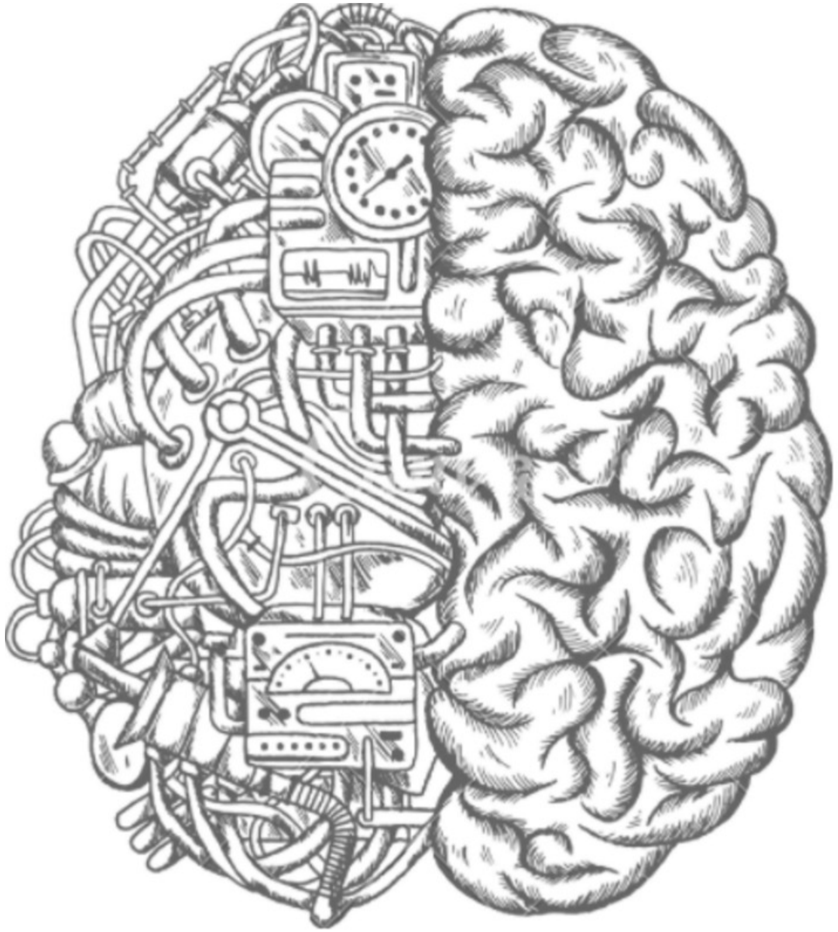
- Machine Learning (ML) Overview
- ML Project Workflow

Machine Learning (ML) Overview

- **Introduction**
- ML from different perspectives
- Different Learnings in ML

Machine Learning

- ML is everywhere!
  - Computer Science
  - Healthcare
  - Retail
  - Manufacturing
  - Energy
  - Financial Service
  - …



What is Machine Learning?

A computer program is said to learn from ***experience,*** *E,* with respect to some class of ***tasks,*** *T,* and performance ***measure,*** *P,* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.

What is Machine Learning?

Machine Learning:

- The term first coined in 1959, by Arthur Samuel from IBM
- A branch of Artificial Intelligence (AI),
- Focused on design and development of algorithm
- Input: empirical data, such as that from sensors or databases,
- Output: ***patterns*** or ***predictions*** thought to be features of the underlying mechanism that generated the data.

Learner (the algorithm):

- Takes advantage of ***data*** to capture ***characteristics of interest*** of their unknown underlying probability distribution.

One fundamental difficulty:

- **Generalization**: The set of all possible behaviors given all possible inputs ***is too large*** to be included in the set of observed examples (training data). Hence the learner must ***generalize*** from the given examples in order to produce a useful output in new cases.

Machine Learning (ML) Overview

- Introduction
- **ML from different perspectives**
- Different Learnings in ML

ML from Other Aspects

The Artificial Intelligence (AI) View:

- Learning is central to **human** knowledge and intelligence, and likewise, it is also essential for building **intelligent machines**.
- Years of effort in AI has shown that trying to build intelligent computers by programming all the rules cannot be done; automatic learning is crucial.
- For example, we humans are not born with the ability to understand language. ***We learn it*** and it makes sense to try to have computers learn language instead of trying to program it all it.

ML from Other Aspects

The Software Engineering View:

- Machine learning allows us to program computers by example, which can be easier than writing code in the traditional way.

The Statistics View:

- Machine learning is the marriage of computer science and statistics: computational techniques are applied to statistical problems.
- Machine learning has been applied to a vast number of problems in many contexts, beyond the typical statistics problems.
- Machine learning is often designed with different considerations than statistics (e.g., speed is often more important than accuracy).

Examples of ML

- Spam Filtering
- Goal: given an email, decide whether it is spam
- The learner learns from
  - Emails marked as spam
  - Emails not marked as spam (inbox)

Examples of ML

- Face Detection

Examples of ML

- Games

Machine Learning (ML) Overview

- Introduction
- ML from different perspectives
- **Different Learnings in ML**

Data

Target

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY

Labels:
- headers
- column names
- feature names

Column: Features, predictors, attributes
Categorical – discrete data
- Integer (0 or 1)
- Text

Numerical – continuous data

Rows: observations, examples

Learning Types of ML

Supervised Learning	Unsupervised Learning	Reinforcement Learning
• Labeled Data • Direct Feedback • Predict outcome • Forecast future	• No labels/targets • No Feedback • Find hidden structure in data	• Decision Process • Reward system • Learn series of actions

Learning Types of ML

Supervised Learning

x	→	Classification	→	y	Discrete
x	→	Regression	→	y	Continuous

Unsupervised Learning

x	→	Clustering	→	y	Discrete ID
x	→	Dimensionality Reduction	→	y	Continuous

ML Project Workflow

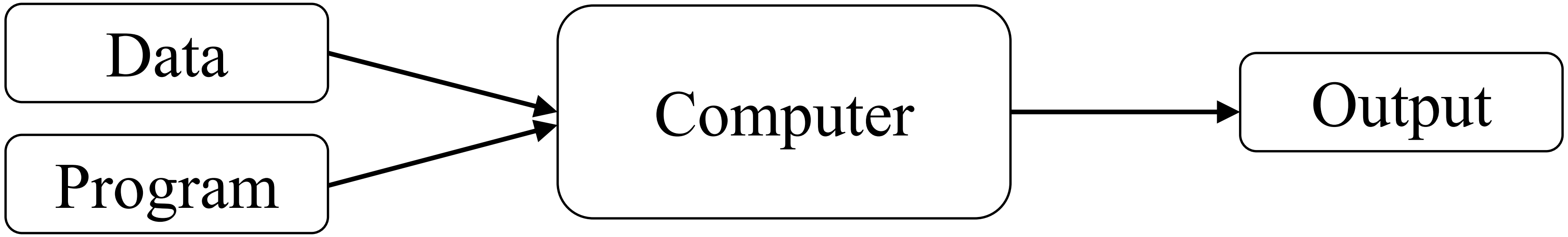
- What makes ML so special? Old School vs. New School
- What is the workflow in ML project?
  - What is preprocessing and exploratory data analysis (EDA)?
  - How do we make models and what is after?
  - How can we make the models better?

ML Project Workflow

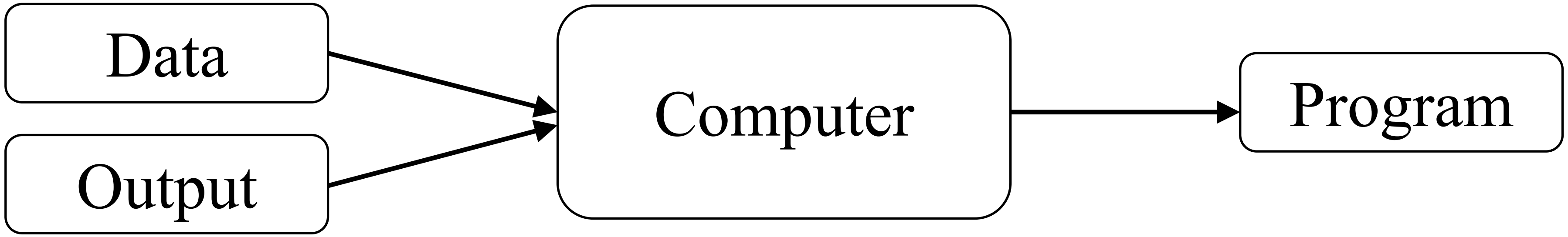
- What makes ML so special? Old School vs. New School
- What is the workflow in ML project?
  - What is preprocessing and exploratory data analysis (EDA)?
  - How do we make models and what is after?
  - How can we make the models better?

ML vs Traditional Approach

- Traditional Programming



- Machine Learning



ML Project Workflow

- What makes ML so special? Old School vs. New School
- What is the workflow in ML project?
  - What is preprocessing and exploratory data analysis (EDA)?
  - How do we make models and what is after?
  - How can we make the models better?

ML in Practice

ML is about:

- Given a collections of examples, called “training data”
- We want to predict something about novel examples, called “test data”

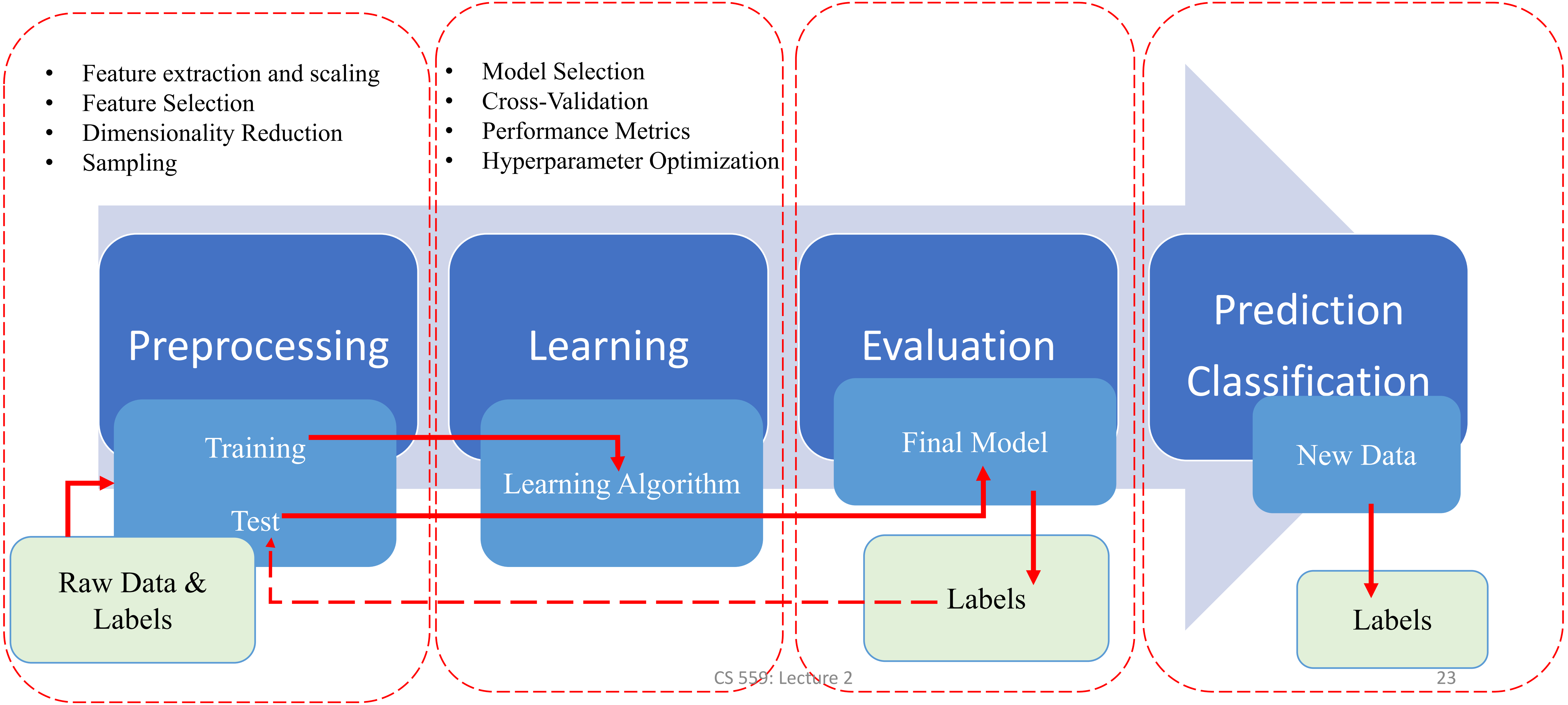
What we usually do:

- Build ***idealized models*** of the application area we are working in
- Develop algorithms and implement in code
- Use historical data to learn numeric parameters, and sometimes model structure
- Use test data to validate the learned model, quantitatively measure its predictions
- Assess errors and repeat…

ML in a Nutshell

- Every machine learning algorithm has three components:
  - Representation / Model Class
  - Evaluation / Objective Function
  - Optimization

Roadmap for ML



Representation / Model Class

- Decision trees
- Sets of rules / Logic programs
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles

Evaluation / Objective Function

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence

Optimization

- Discrete optimization
  - Minimal Spanning Tree
  - Shortest Path

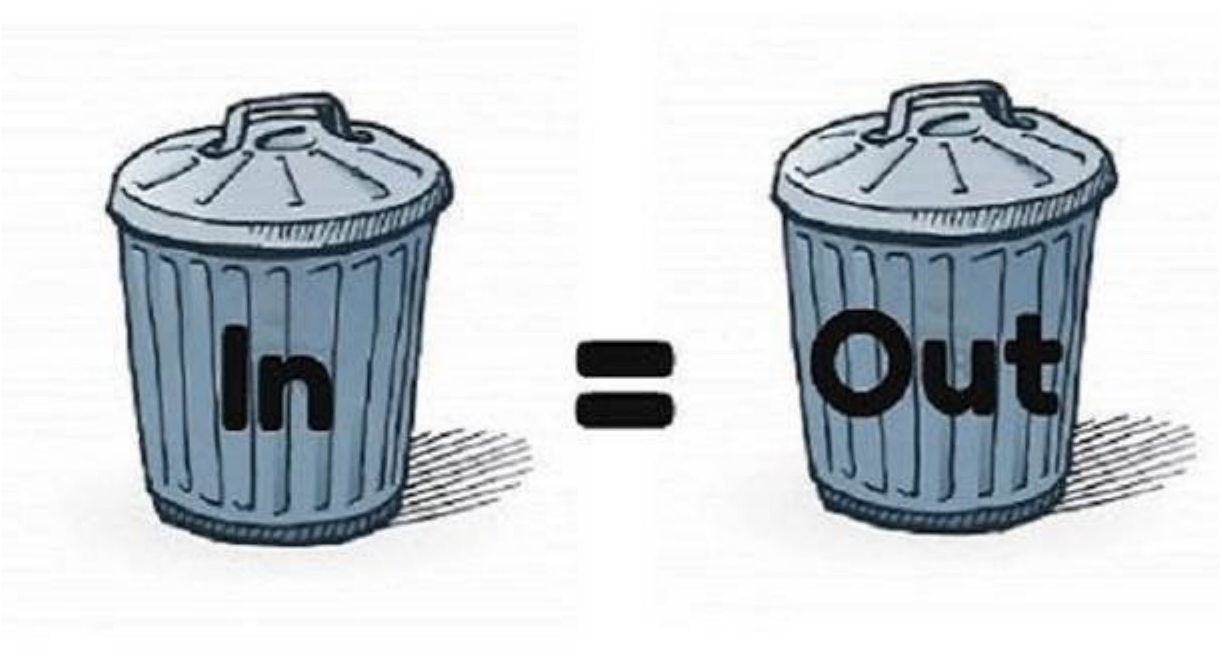
- Continuous Optimization
  - Gradient Descent
  - Linear Programming

ML Project Workflow

- What makes ML so special? Old School vs. New School
- What is the workflow in ML project?
  - What is preprocessing and exploratory data analysis (EDA)?
  - How do we make models and what is after?
  - How can we make the models better?

Importance of data preprocessing

- Data preprocessing is to make sure we have sensible data for ML



[https://www.linkedin.com/pulse/supply-chain-planning-garbage-out-cheryl-wiebe]

Some data issues need to be addressed before applying ML algorithms

Missing values:

- Observation we intended to collect but did not get them
  - Data entry issues, equipment errors, incorrect measurement etc
    - An individual may only have responded to certain questions in a survey, but not all
- Problems of missing data
  - Reduce representativeness of the sample
  - Complicating data handling and analysis
  - Bias resulting from differences between missing and complete data

Missing data handling

Reducing the data set

- Elimination of samples with missing values
- Elimination of features (columns) with missing values

Imputing missing values

- Replace the missing value with the mean/median (numerical) or most common (categorical) value of that feature

Treating missing attribute values as a special value

- Treat missing value itself as a new value and be part of the data analysis
    - Make a simple model to estimate the missing value

Some data issues need to be addressed before applying ML algorithms

Data in different scales

- Weight of a person (Pounds) vs weight of an elephant (US ton)
  - 1 US ton = 2000 Pounds
- For predicting weights for them, the error of elephant weights will significantly bias the prediction accuracy relative to the error for the persons weights

Data in different scale

- Approaches to bring different values onto the same scale
  - Normalization: rescale the feature to a range of [0,1]
  - Standardization: re-center the feature to the mean and scaled by variance

$$x_{norm}^{(j)} = \frac{x^{(j)} - x_{min}}{x_{\max} - x_{min}}$$

x_{min} and x_{max} are the min/max values of feature column $x^{(j)}$

$$x_{std}^{(j)} = \frac{x^{(j)} - \mu_x}{\sigma_x}$$

μ_x and σ_x are the mean and standard deviation of feature column $x^{(j)}$

- Data scaling should be one of the first steps of data preprocessing for many machine learning algorithms
  - Some machine learning algorithms can handle data in different scales (e.g., decision trees and random forests)

Data in different scale

- Standardization:
  - When measurements are in different units, we standardize the feature around the center 0 with 1σ.
  - Values at different scales can cause bias.
  - Assumes that data has a Gaussian distribution and if ML algorithm holds the assumption (e.g., Linear Regression, Logistic Regression, Linear Discriminant Analysis).

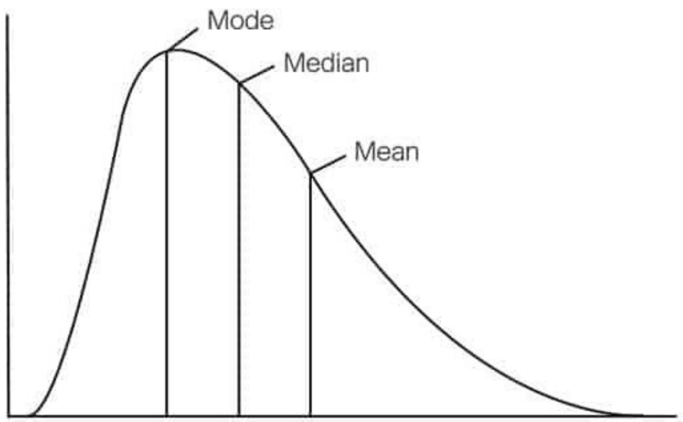
- Normalization:
  - To changes values to a common scale (between 0 and 1) without distorting differences in the ranges of values.
  - Typically used when features are in different ranges.
  - Use when distribution is not known or skewed.
  - K-Nearest Neighbors and Neural Networks

Handling Skewed Data

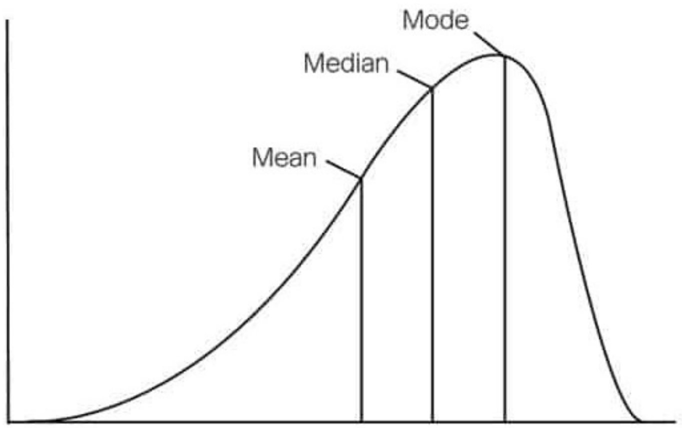
- Real-world data can be messy and contains attributes that need modifications before they can be used in modeling.
- In case of normal distribution, the mean, median, and mode are approximately close to each other at the center of distribution.
- The skewness of data can be determined by how these quantities are related to one another.

Handling Skewed Data

- Right skewed or Positive Skewed:
  - Mean > Median > Mode
- Left Skewed or Negative Skewed
  - Mode > Median > Mean
- The tail region may act outliers that can affect the model's performance in regression models.



Right-Skewed



Left-Skewed (Negative Skewness)

Handling Skewed Data

- Log transformation transforms skewed distribution to a normal distribution. (usually applies to right skewed data)
  - Values ≤ 0 cannot be transformed.
  - Add some constant so the minimum value be greater than 1 $\Rightarrow \log(1) = 0$
- Remove outliers (both)
- Normalize (applies to right skewed data)
- Cube root, square root (applies to right skewed data)
- Reciprocal (applies to right skewed data)
- Square (applies to left skewed data)
- Box Cox transformation (applies to both)
  - Transform using equations below:
  - $y(\lambda) = \begin{cases} (y^{\lambda} - 1)/\lambda & if\ \lambda \neq 0\ and\ y > 0 \\ \log y & if\ \lambda = 0\ and\ y > 0 \end{cases}$
  - $y(\lambda) = \begin{cases} ((y + \lambda_2)^{\lambda_1} - 1)/\lambda_1 & if\ \lambda_1 \neq 0\ and\ y < 0 \\ \log(y + \lambda_2) & if\ \lambda = 0\ and\ y < 0 \end{cases}$
  - Usually, $\lambda = [-5,5]$ but we use a λ value that gives the best approximation to a normal distribution.

Categorical data handling

- for ordinal data, convert the strings into comparable integer values
  - E.g., XL > L > M > S → 5 (XL) > 4 (L) > 3 (M) > 2 (S)
  - Note that the value of integer itself has no special meaning besides for ordering
  - Mapping needs to be unique: 1 to 1 mapping for going back and forth
- For nominal data, convert the strings into integers
  - E.g., Red (0), Blue (1), Green (2)
  - A common practice to avoid software glitches in handling strings
  - Note that the value of integer itself has no special meaning (non-comparable)
  - Mapping needs to be unique: 1 to 1 mapping for going back and forth
- To avoid mistakenly comparing encoded integers for nominal data, one- hot encoding can be used
  - Each unique value becomes a separate dummy feature

Correlation between features & Feature Engineering

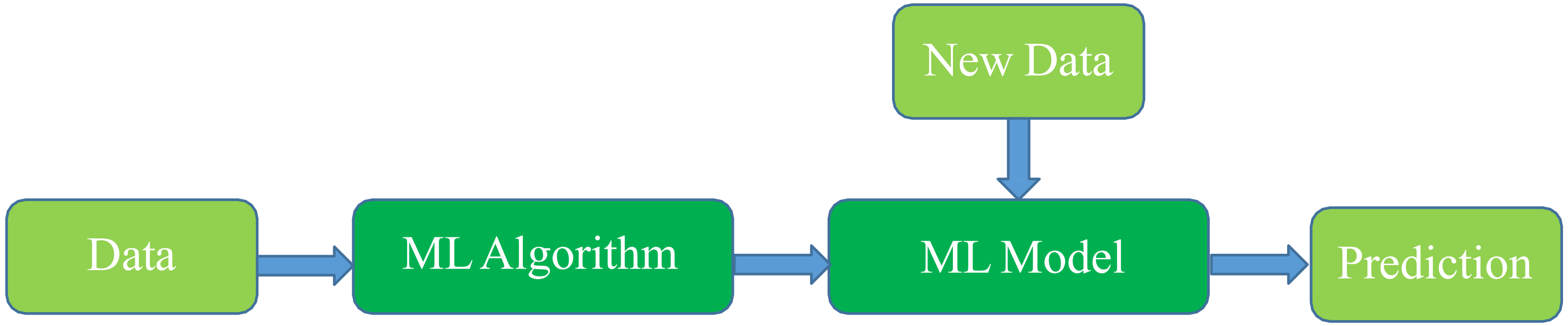
- One good way to reduce the data size
- Correlations between two features explains how they are related to each other.
  - Pearson correlation coefficient is widely used.
  - Ranges from -1 to 1.
- Feature engineering extract features using domain knowledge
  - Improves the performance of ML
  - Sometimes can be considered as applied ML
- For example, if X and Y are tightly correlated
  - We can use only X as an independent variable
  - Or make a new feature call Z = XY as an independent variable

ML Project Workflow

- What makes ML so special? Old School vs. New School
- What is the workflow in ML project?
  - What is preprocessing and exploratory data analysis (EDA)?
  - **How do we make models and what is after?**
  - How can we make the models better?

Machine learning, models and data

- Machine learning is an algorithm that learns a model from data (training), so that the model can be used to predict certain properties about new data (generalization)

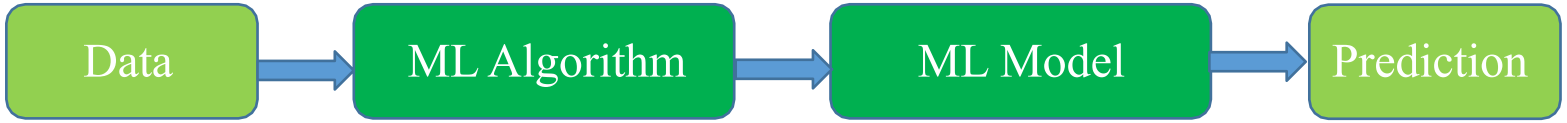


ML Project Workflow

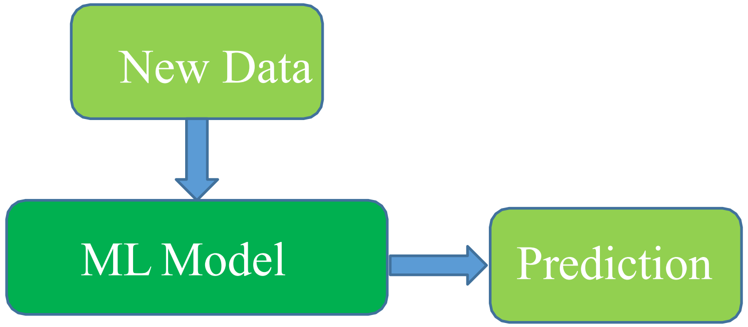
- What makes ML so special? Old School vs. New School
- What is the workflow in ML project?
  - What is preprocessing and exploratory data analysis (EDA)?
  - How do we make models and what is after?
  - **How can we make the models better?**

Training vs Inference

- Training is to build the ML model from data



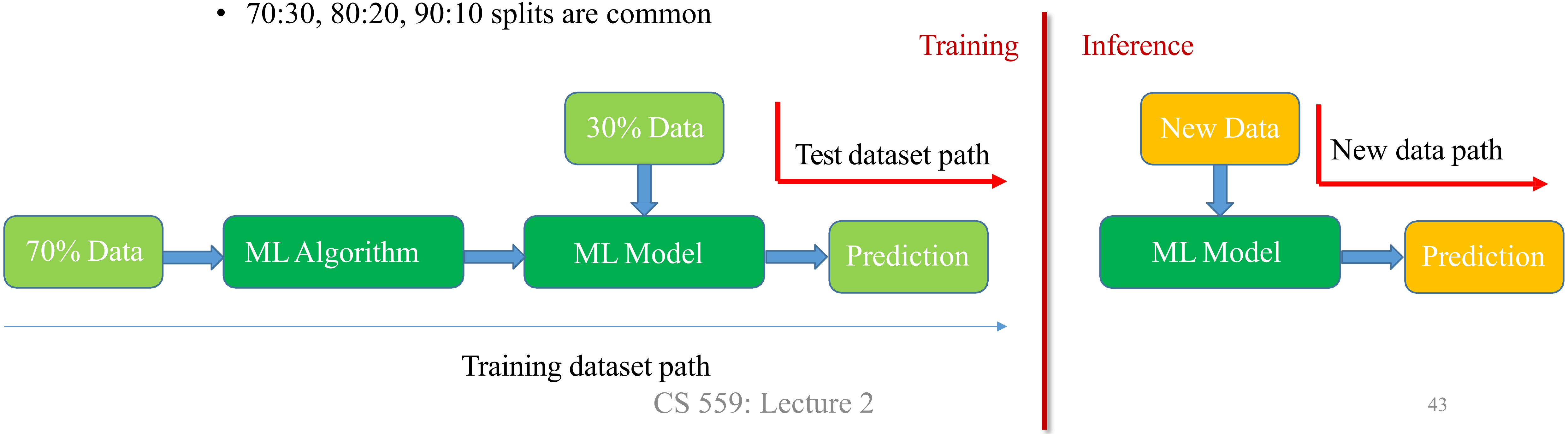
  - Typically, training is a one-time effort, but computationally intensive
  - Speed is a main concern
- Inference is to use the ML model to predict results for new data (generalization – most interesting for applications)



- Typically, inference is fast but happens more frequently with a lot of more new data (unlabeled)
- Scalability is a main concern

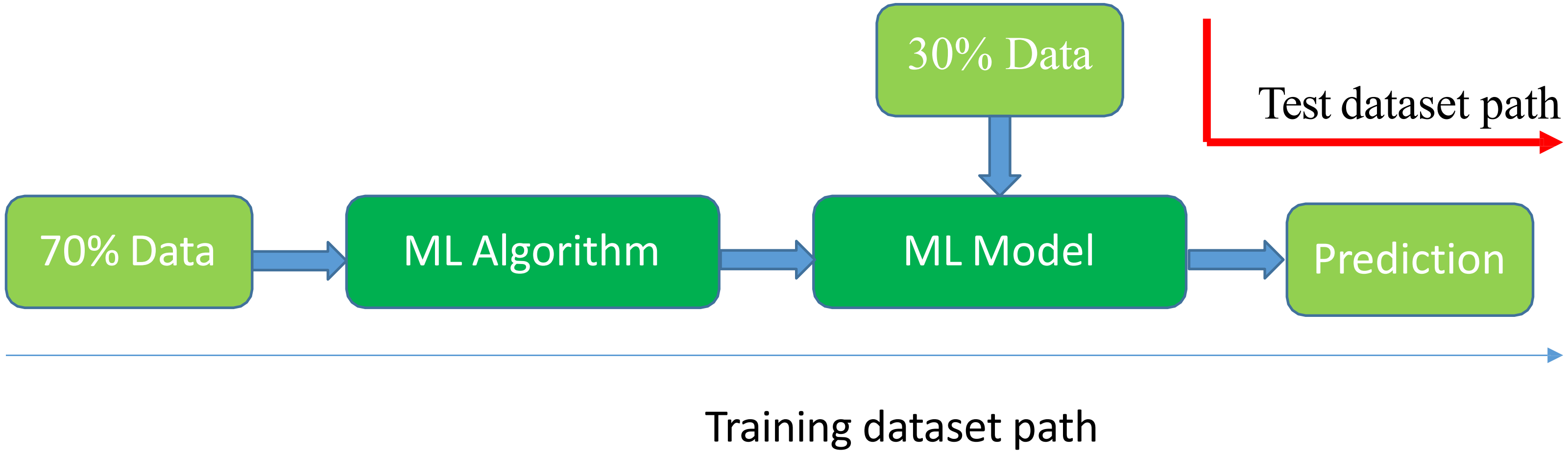
Split known data into training and test datasets

- Data known to ML model developers are split into two sets
  - Training dataset: data used to train the model
  - Test dataset: data used to give an indication on how well the trained model will generalize to new data (unknown at this point)
    - Test dataset is kept till the very end to evaluate the final model
    - Since test dataset withholds valuable information that the learning algorithm could benefit from, we don't want to put too much data into the test dataset either
      - 70:30, 80:20, 90:10 splits are common



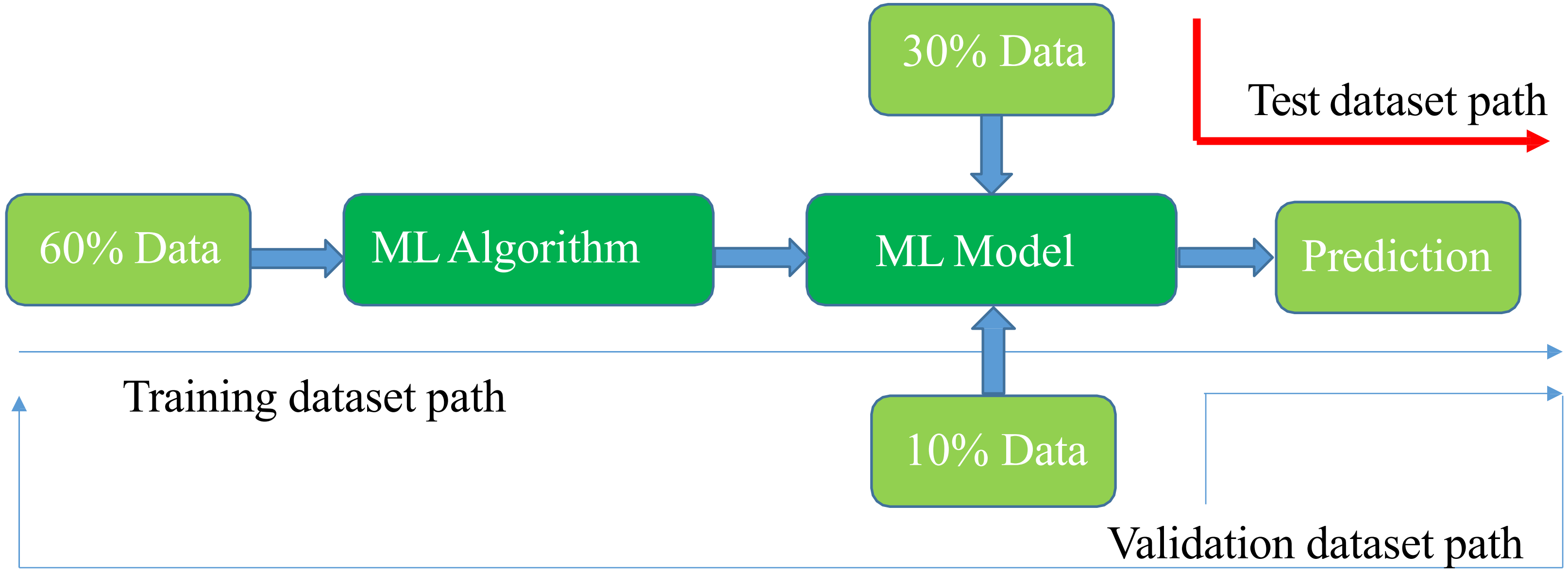
Cross-validation: a model tuning process

- How can we make the model training process to be aware of the targeted generalization quality so that training can do something about it?
- We need to put the predicted generalization results as part of the training optimization goal
  - We can NOT use the predicated generalization results from the test data, otherwise, the test data would become part of the training process
  - We want to keep the test data still independent of training so that its predication can still be a good indication of generalization quality for future unknown new data



Holdout cross-validation

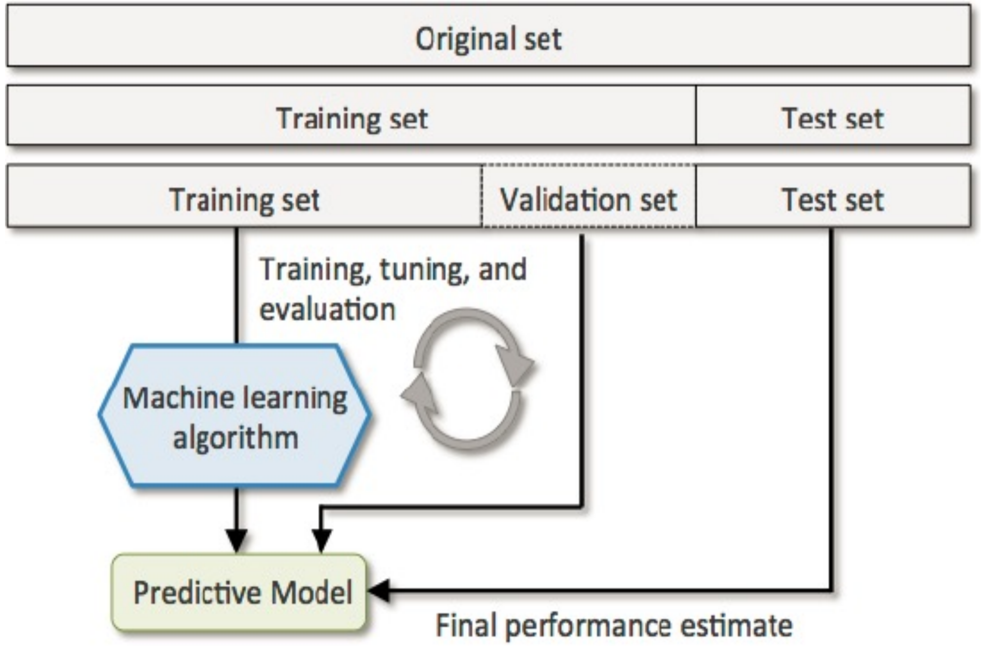
- Holdout cross-validation method
  - Training dataset is further split into two sets: training set + validation set



- Validation results are used to drive the continuation of training process
  - Until we obtain a reasonable validation result
- We still use test data to report the predicated generalization quality

Pros and Cons of holdout cross-validation

- Another view of the holdout cross-validation



- Pros: validation set is used to tune the model parameters for better generalization
- Cons: final results may be sensitive to how the dataset was split for validation

K-fold cross-validation

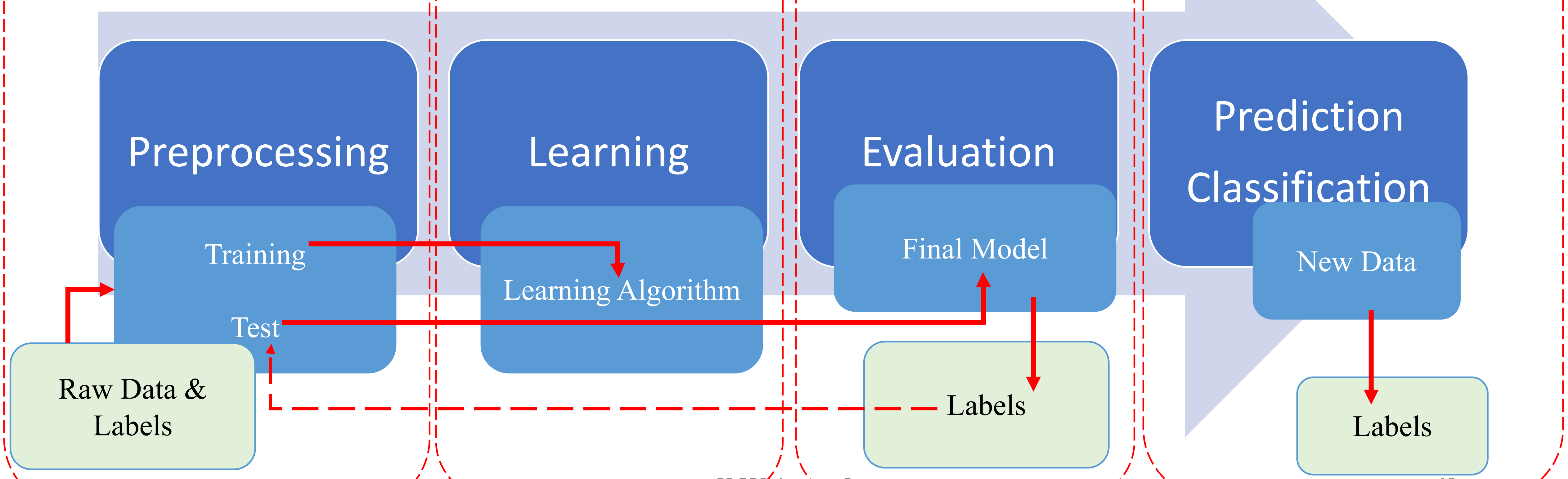
- Repeat holdout cross-validation k times on k subsets of the training data
  - Randomly split the training dataset into k folds without replacement
    - K-1 folds are used for training, and one fold used for validation
  - Repeat this k times so that we obtain k models
  - Typically k=10, but larger k for smaller dataset, and smaller k for larger dataset

- Pro: average performance from k models is less sensitive to the split
- Con: more computation time

Roadmap for ML

- Feature extraction and scaling
- Feature Selection
- Dimensionality Reduction
- Sampling

- Model Selection
- Cross-Validation
- Performance Metrics
- Hyperparameter Optimization



Conclusion

ML Overview

- Machine Learning is everywhere!
- Garbage in Garbage out – ML does not over perform from the input.
- Pre-processing is important and the most time consuming part in ML.
- ML projects are broadly split into supervised learning and unsupervised learning.
- Splitting dataset to improve the performance is a standard way in ML.