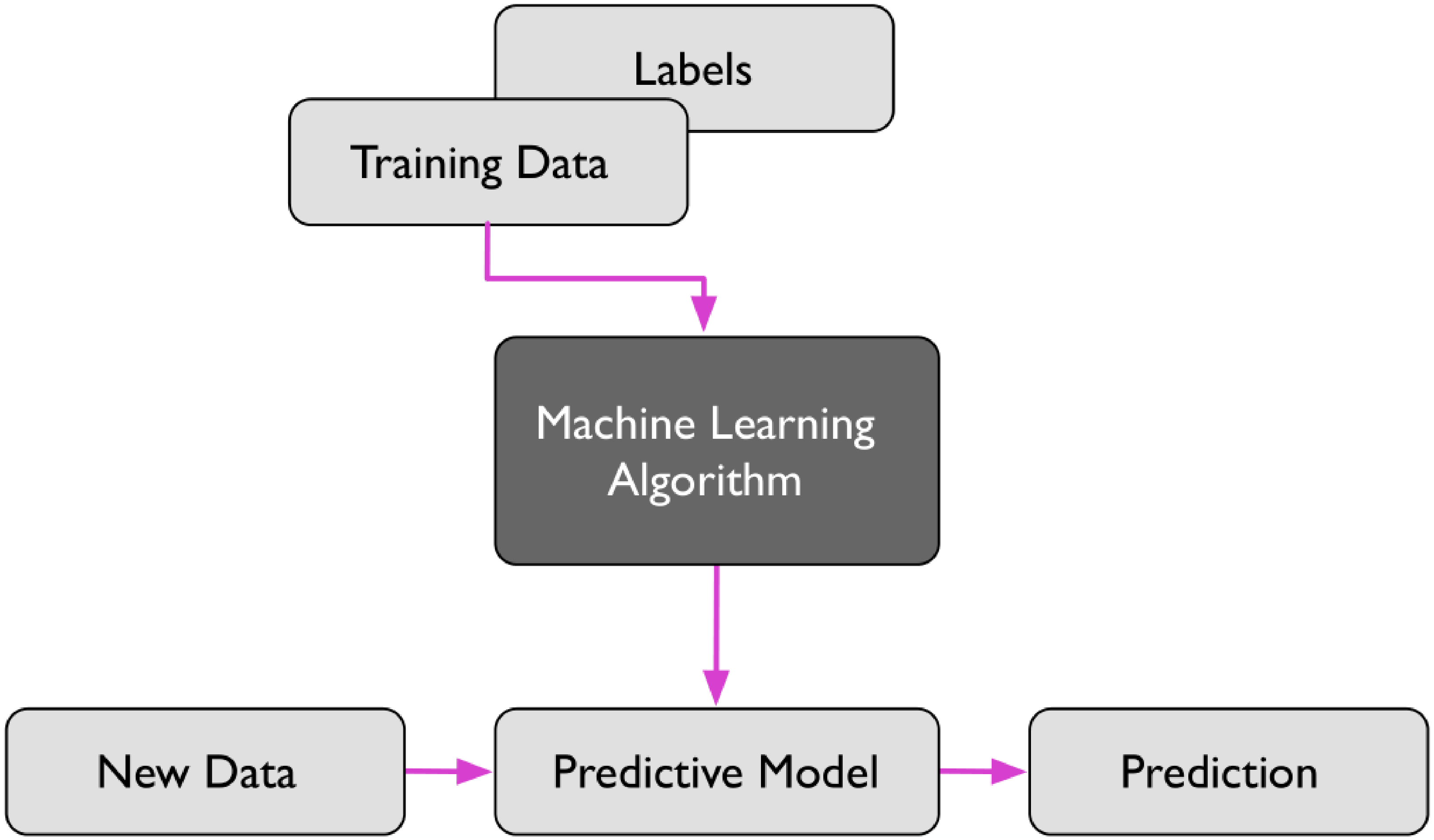
Supervised Learning Workflow



Supervised Learning Notation

Training set: $\mathscr{D} = \{\langle \mathbf{x}^{[i]}, y^{[i]} \rangle, i = 1, \dots, n\}$,

Unknown function: $f(\mathbf{x}) = y$

Hypothesis: $h(\mathbf{x}) = \hat{y}$

$$\begin{tikzcd}
 & h(\mathbf{x}) = \hat{y} \arrow[dl, "\text{Classification}"'] \arrow[dr, "\text{Regression}"] & \\
h : \mathbb{R}^m \rightarrow \_\_ & & h : \mathbb{R}^m \rightarrow \_\_
\end{tikzcd}$$

Hypothesis

Unknown Function (f):

- The unknown function, often denoted as "f," represents the true relationship or mapping between the input and output variables in the underlying data-generating process. It is the ideal or true mapping that we aim to approximate or learn from the available data.
- In supervised learning, the goal is to learn an approximation of this unknown function based on a dataset containing input-output pairs. The machine learning algorithm tries to infer the mapping between inputs and outputs by analyzing the patterns in the provided data.

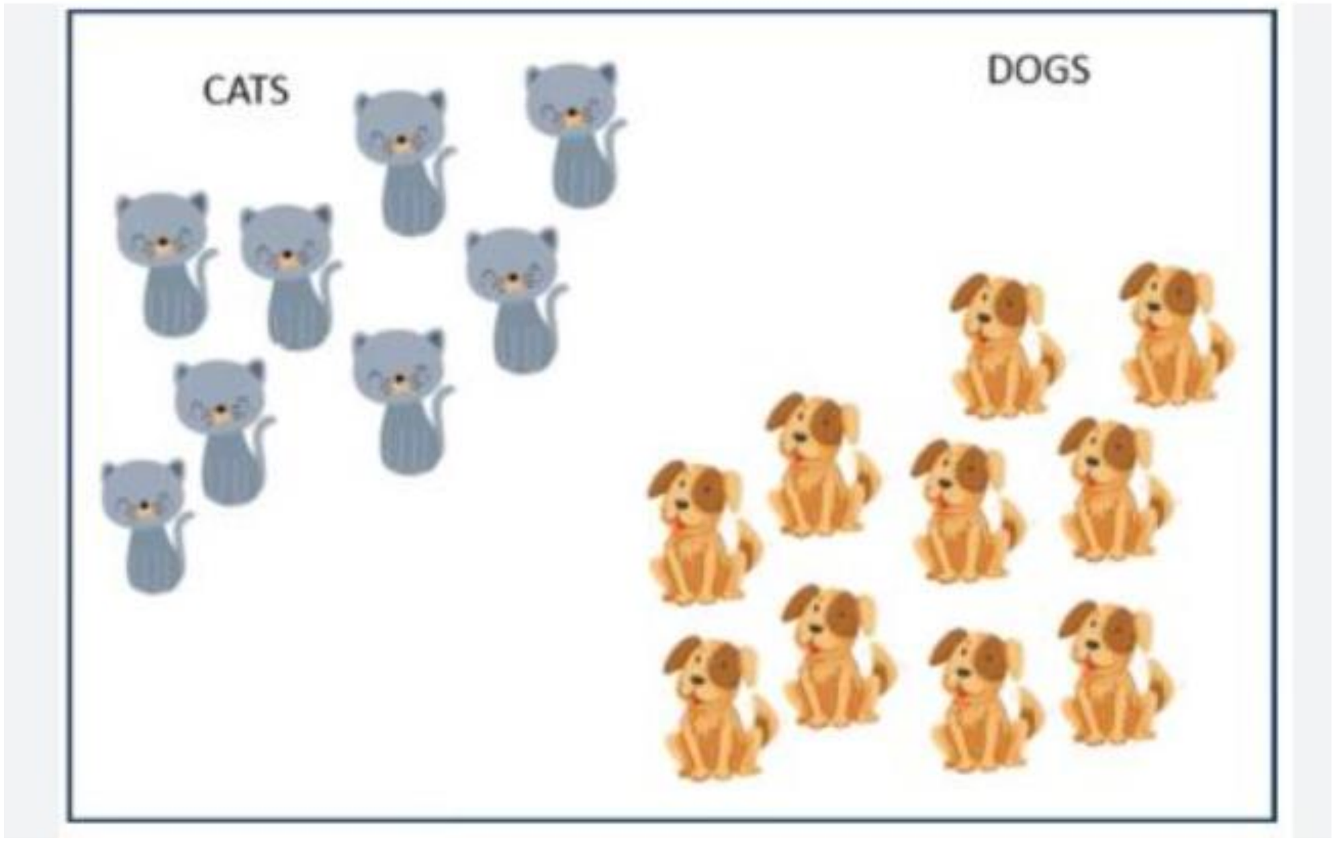
Hypothesis (h):

- The hypothesis, often denoted as "h," represents the learned or proposed mapping between inputs and outputs by the machine learning algorithm. It is the model's approximation of the unknown function.
- During the training phase, the algorithm searches for the hypothesis that best fits the training data. The hypothesis is typically chosen from a predefined class of models, such as linear regression, decision trees, neural networks, etc.
- The quality of the hypothesis is evaluated based on how well it generalizes to new, unseen data. The goal is to find a hypothesis that captures the underlying patterns in the data and can make accurate predictions on new instances.

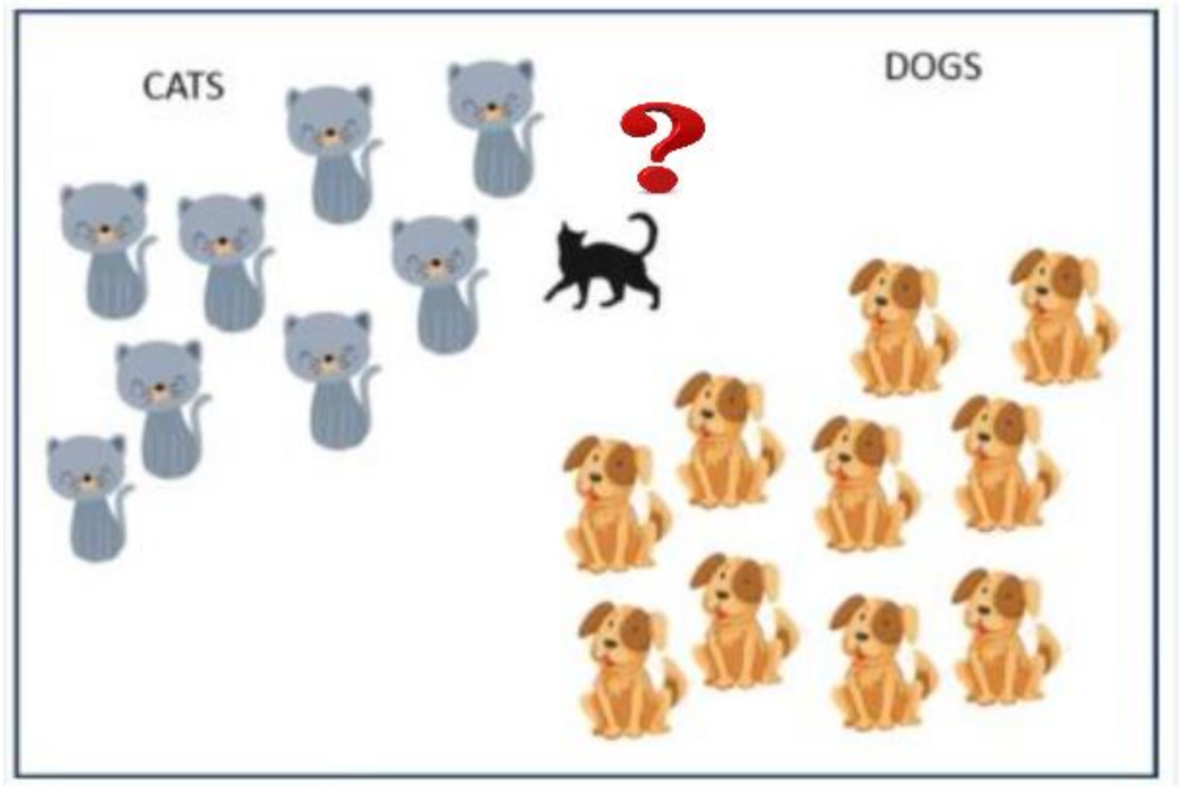
Hypothesis

- The objective is to optimize the model’s parameters to achieve the best predictive performance on new, unseen data, and a cost function is used to assess the hypothesis’ accuracy.

Example

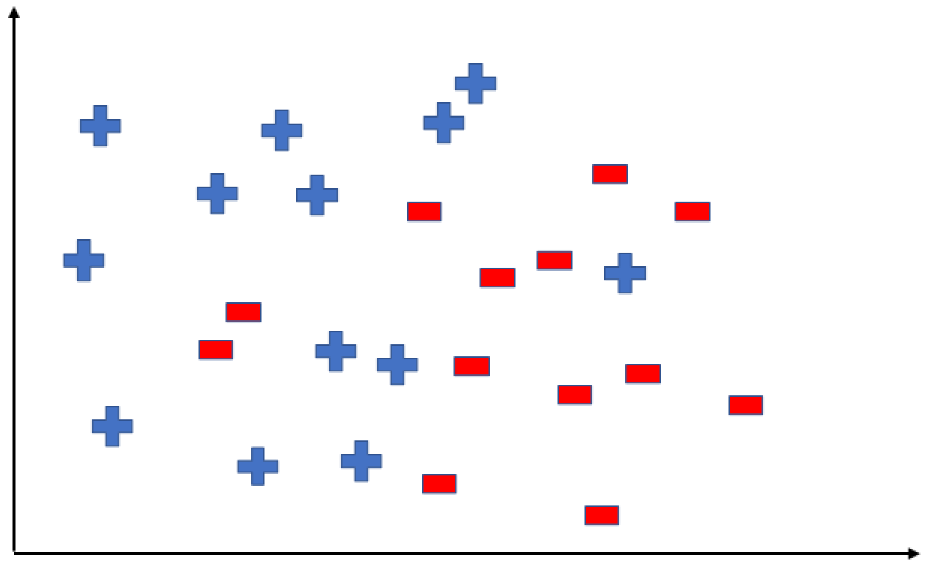


Example

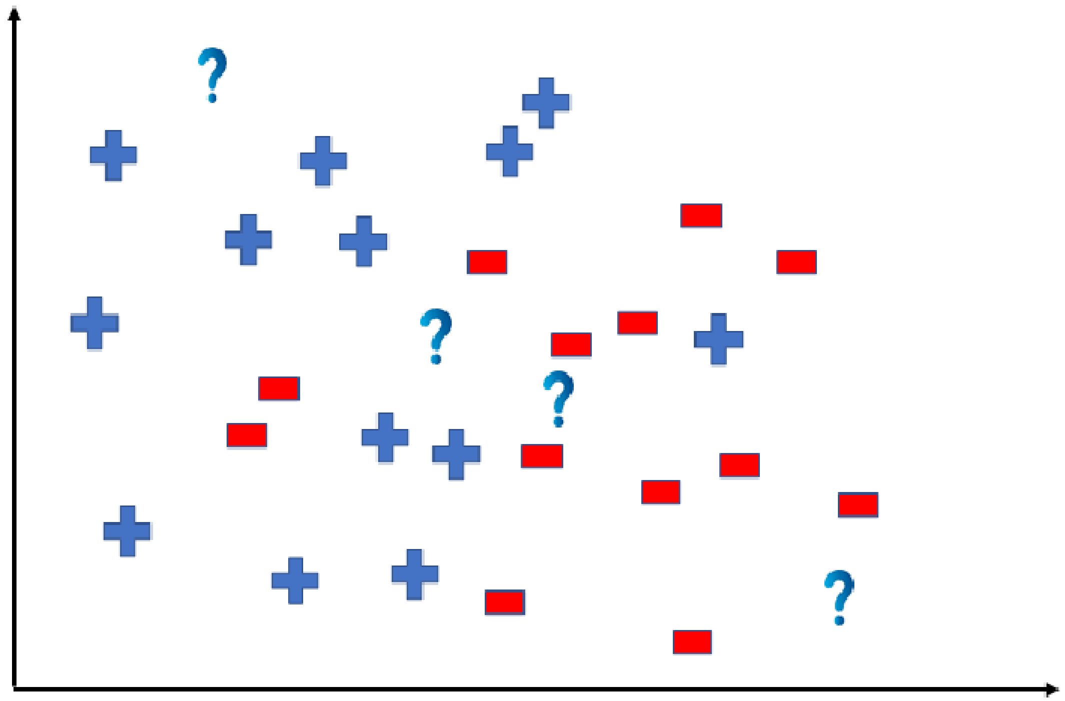


Example

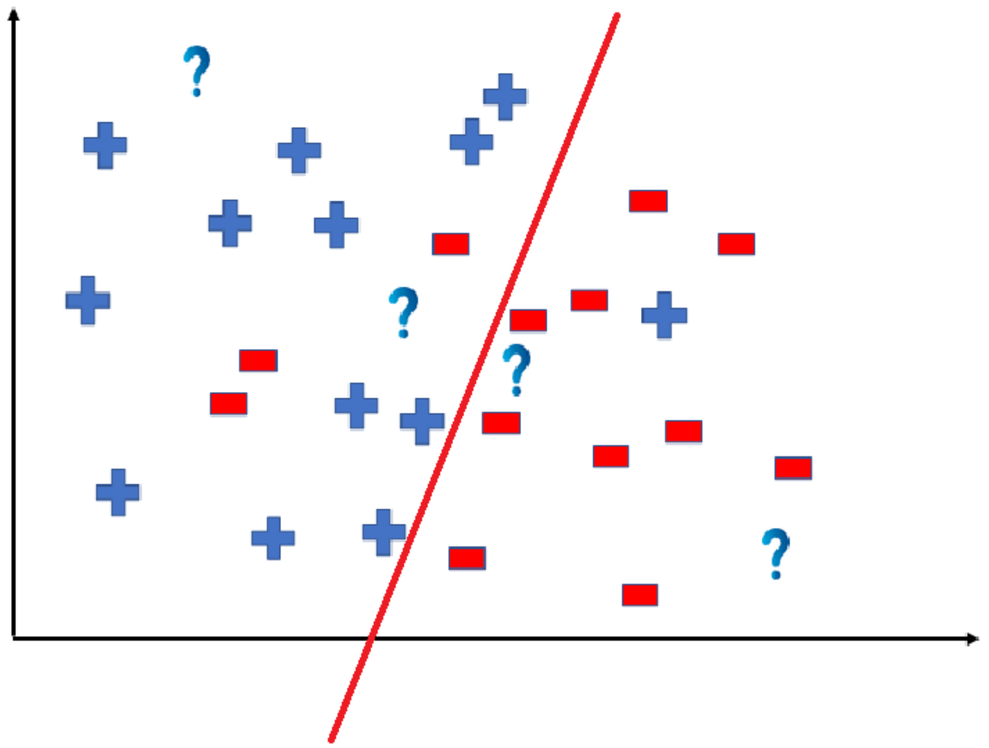
Mapping of training data



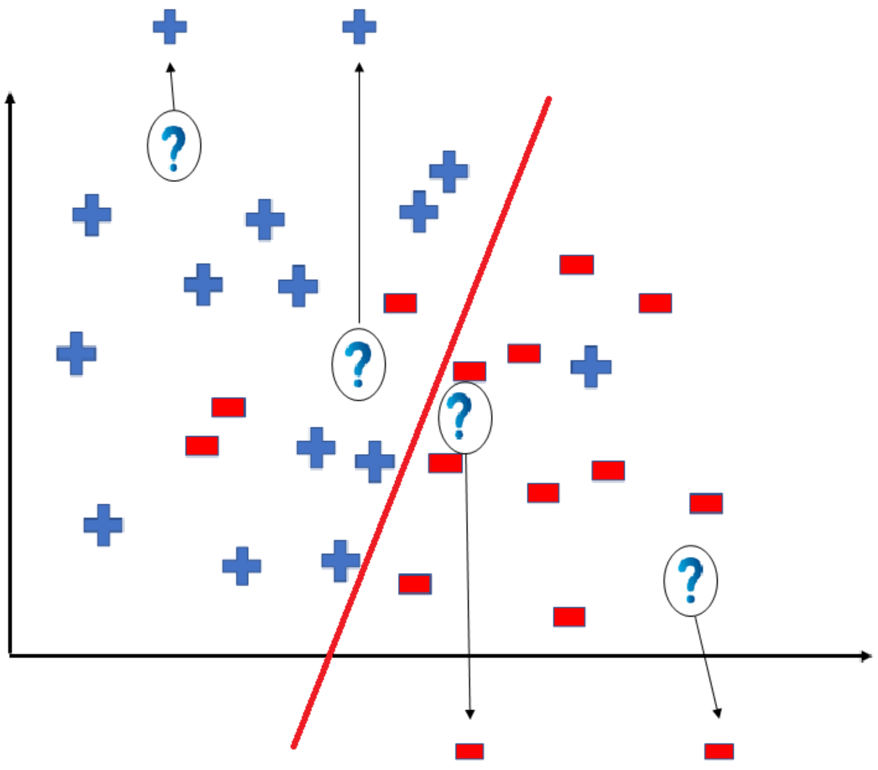
Test data



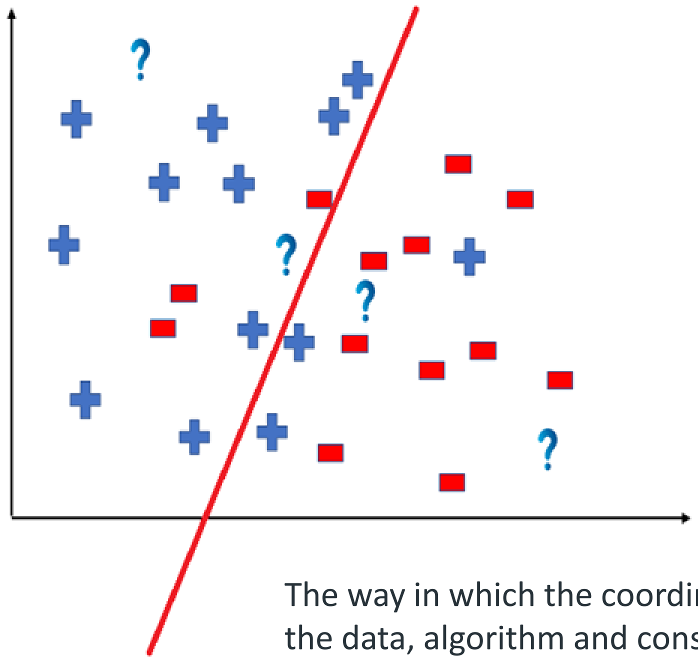
Deciding boundry



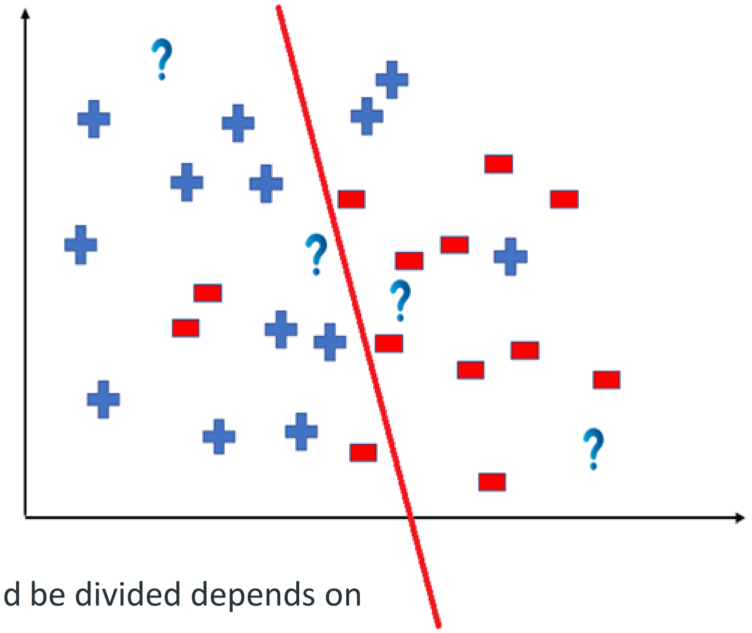
Predicting the unknown data



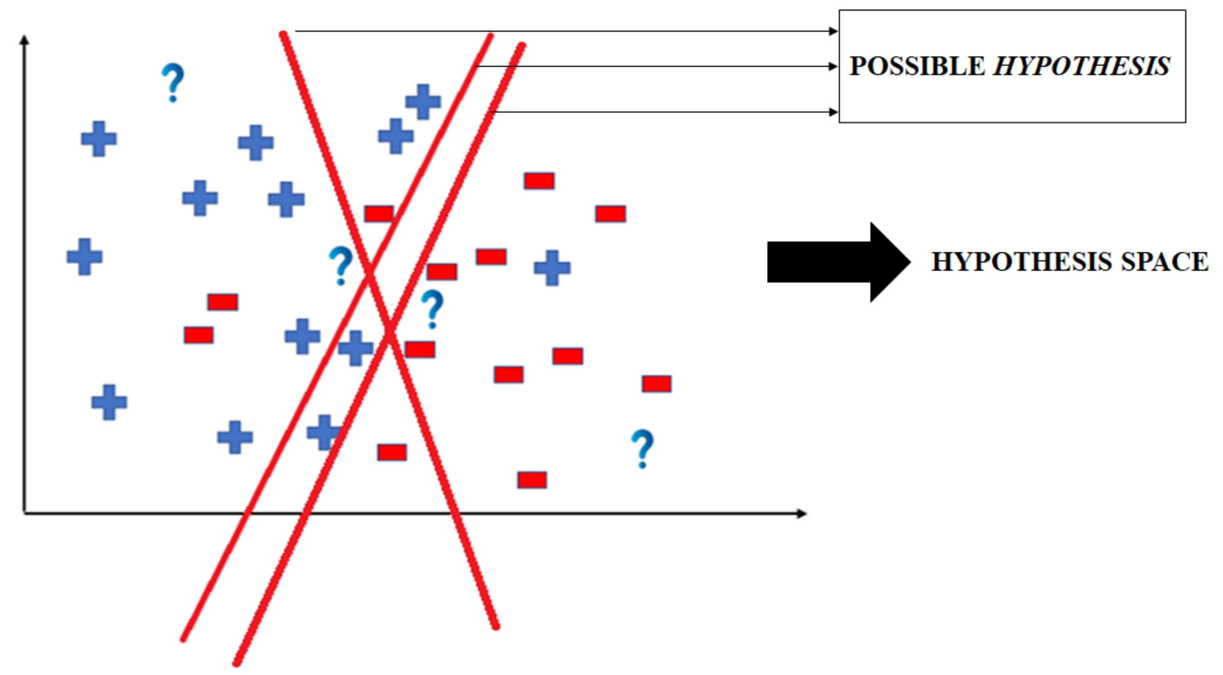
Alternate ways to define the boundry



OR



The way in which the coordinate would be divided depends on the data, algorithm and constraints.


POSSIBLE *HYPOTHESIS*
HYPOTHESIS SPACE

Learning System

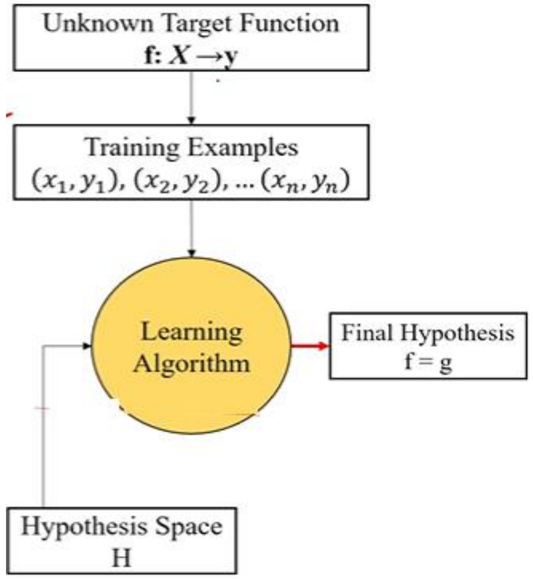
- Learning system is a machine learning model finds the final hypothesis that best approximates the **unknown** target function.

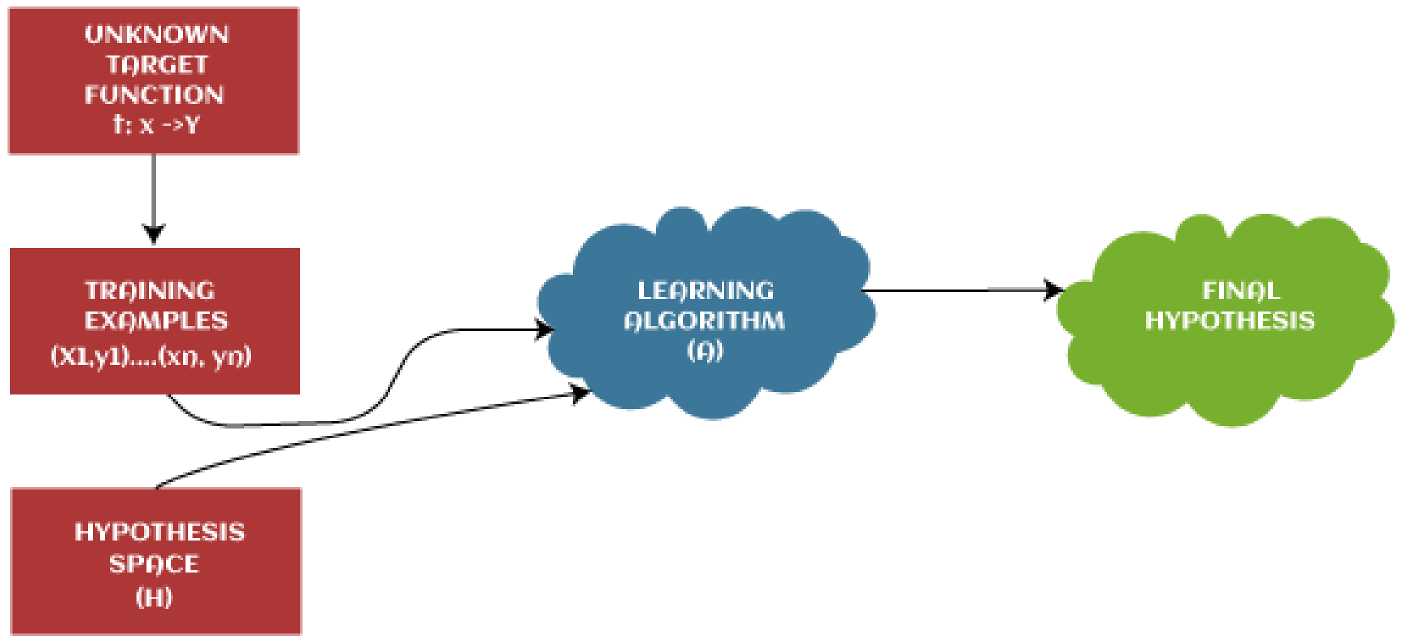
Three elements in learning system

- Task (T), Training Experience (E), and Performance Measure (P).
- The learning process initiates with task T, performance measure P, and training experience E.
- Objective is to find an **unknown** target function.
- The target function is an exact knowledge to be learned from the training experience, and its unknown.

E.g., Credit card approval Learning Problem

- The learning system receives customer application records as experience and task would be to classify whether the given customer application is eligible for a credit card.
- Here, the training examples represented as (x_1, y_1), (x_2, y_2), … (x_n, y_n) where x represents customer application details and y represents the status of credit card approval.




UNKNOWN TARGET FUNCTION f: x ->Y
TRAINING EXAMPLES (X1,y1)....(xn, yn)
HYPOTHESIS SPACE (H)
LEARNING ALGORITHM (A)
FINAL HYPOTHESIS

Dataset Representation for Learning Concept

- Learning system learns the knowledge in the form of pattern from the training dataset.
- Dataset is represented as a table called **Instance Space** X.
- Rows x_i represent the training example **or** observed sample **or** sample **or** instance.
- Columns represent the features **or** attributes.

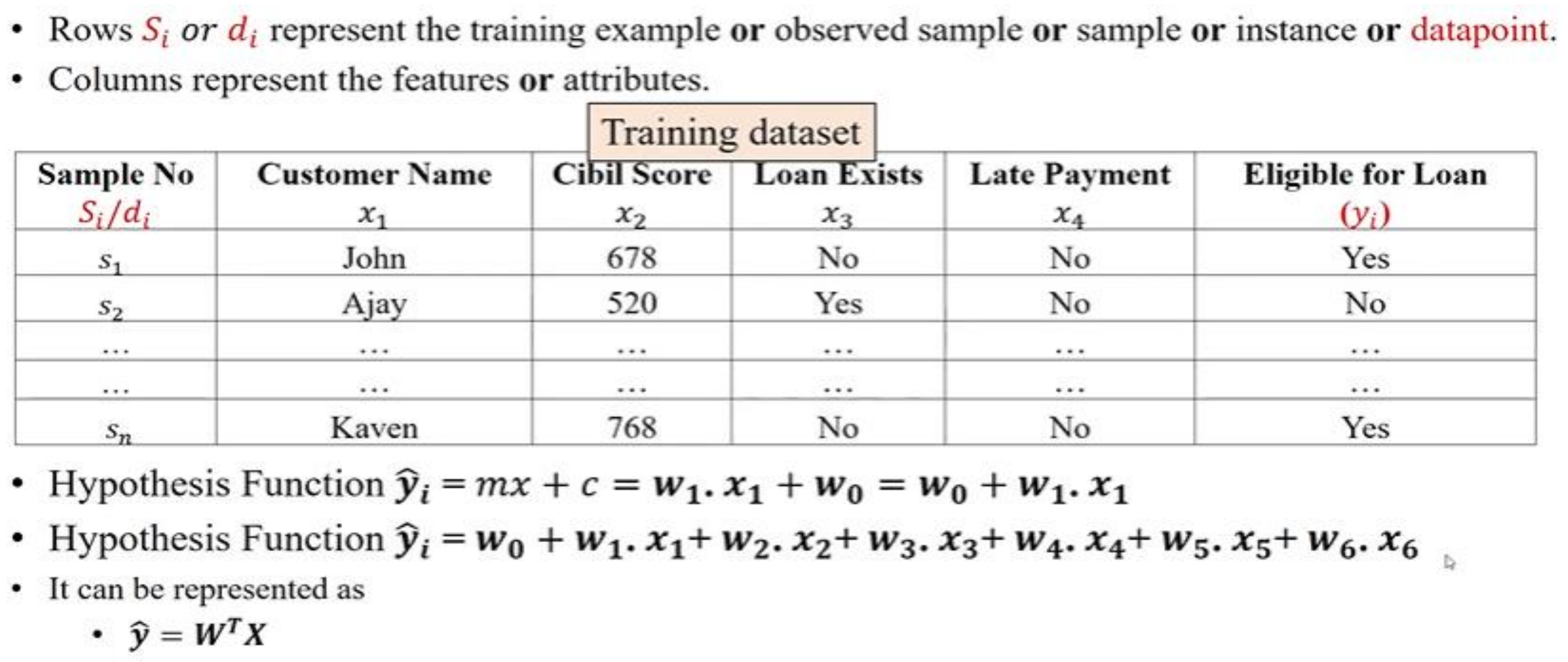
Training dataset

Sample No	**Customer Name**	**Cibil Score**	**Loan Exists**	**Late Payment**	**Eligible for Loan** (y_i)
x_1	John	678	No	No	Yes
x_2	Ajay	520	Yes	No	No
x_3	Nikil	759	Yes	Yes	No
...	...	...	...	...	...
...	...	...	...	...	...
x_n	Kaven	768	No	No	Yes

Testing dataset

Sample No	**Customer Name**	**Cibil Score**	**Loan Exists**	**Late Payment**	**Eligible for Loan** (y_i)
T_1	Andrew	673	No	No	?
T_2	Harish	560	Yes	No	?

Dataset Representation for Hypothesis function

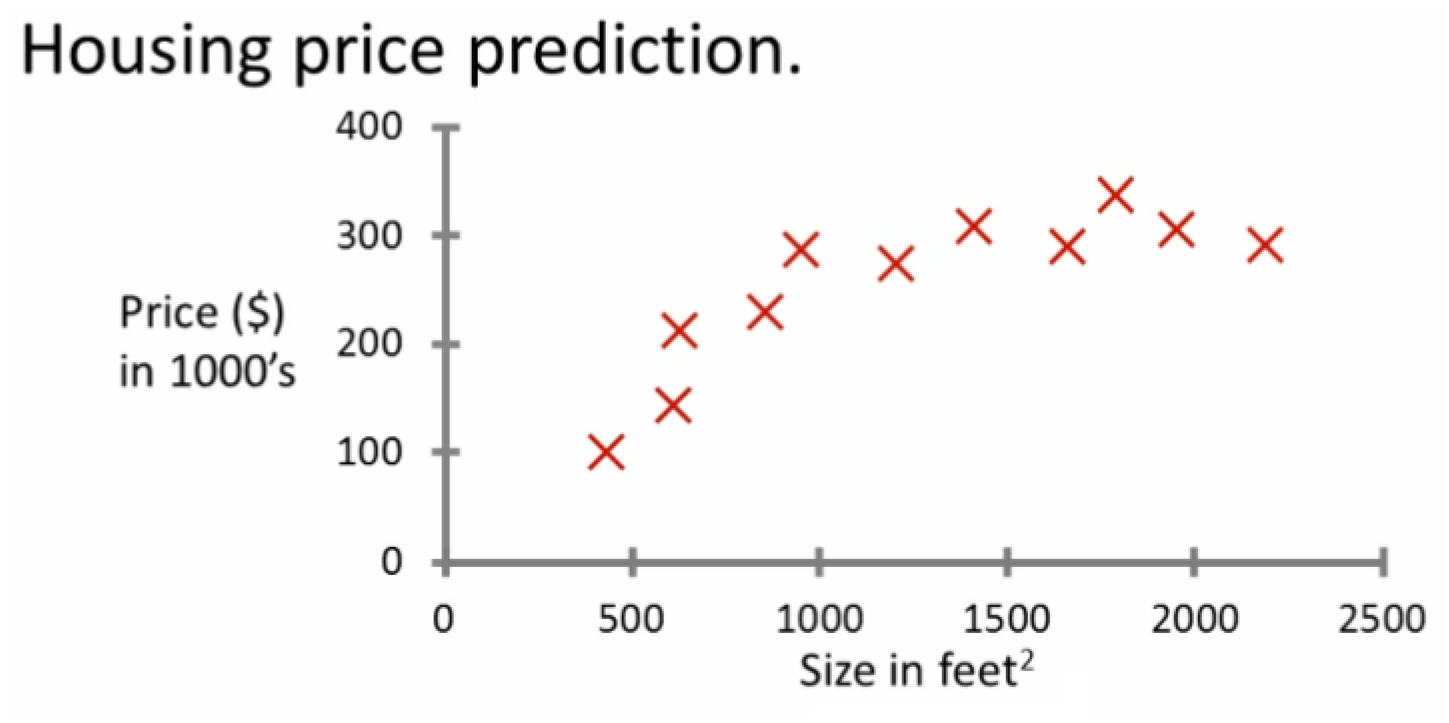


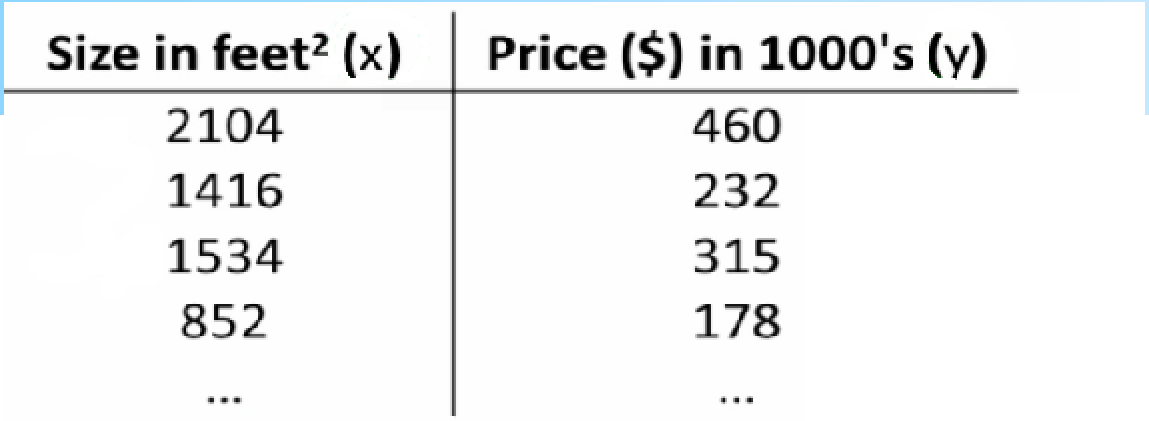
- Rows S_i *or* d_i represent the training example **or** observed sample **or** sample **or** instance **or** datapoint.
- Columns represent the features **or** attributes.

Training dataset

Sample No S_i/d_i	**Customer Name** x_1	**Cibil Score** x_2	**Loan Exists** x_3	**Late Payment** x_4	**Eligible for Loan** (y_i)
s_1	John	678	No	No	Yes
s_2	Ajay	520	Yes	No	No
…	…	…	…	…	…
…	…	…	…	…	…
s_n	Kaven	768	No	No	Yes

- Hypothesis Function $\hat{y}_i = mx + c = \boldsymbol{w_1 . x_1 + w_0 = w_0 + w_1 . x_1}$
- Hypothesis Function $\hat{y}_i = \boldsymbol{w_0 + w_1 . x_1 + w_2 . x_2 + w_3 . x_3 + w_4 . x_4 + w_5 . x_5 + w_6 . x_6}$
- It can be represented as
  - $\hat{y} = \boldsymbol{W^T X}$


Housing price prediction.
Price ($) in 1000's
400
300
200
100
0
0
500
1000
1500
2000
2500
Size in feet^2



Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

- Housing price data example used earlier
  - Supervised learning regression problem
- What do we start with?
  - Training set (this is your data set)
  - Notation (*used throughout the course*)
    - m = number of **training examples**
    - x's = input variables / features
    - y's = output variable "target" variables
      - (x,y) - single training example
      - (x^i, y^j) - specific example (ith training example)

        - i is an index to training set

Data Representation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

Feature vector

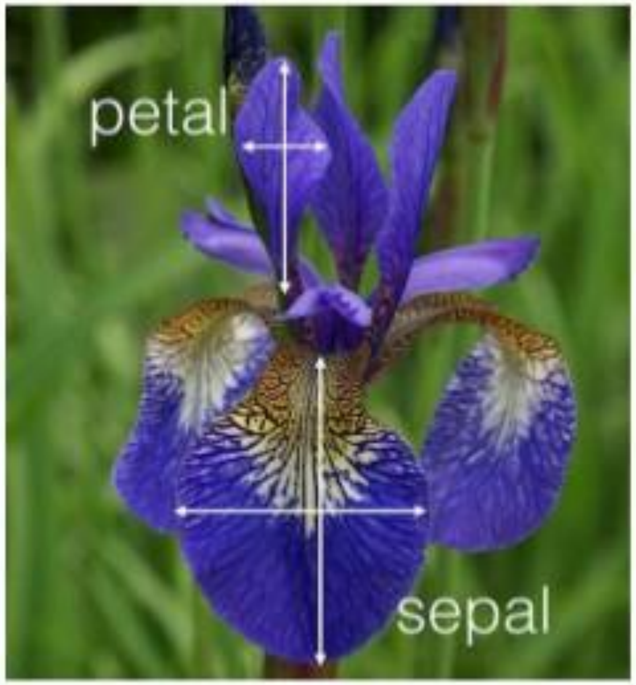
$$\mathbf{X} = \begin{bmatrix} x_1^{[1]} & x_2^{[1]} & \cdots & x_m^{[1]} \\ x_1^{[2]} & x_2^{[2]} & \cdots & x_m^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[n]} & x_2^{[n]} & \cdots & x_m^{[n]} \end{bmatrix}$$

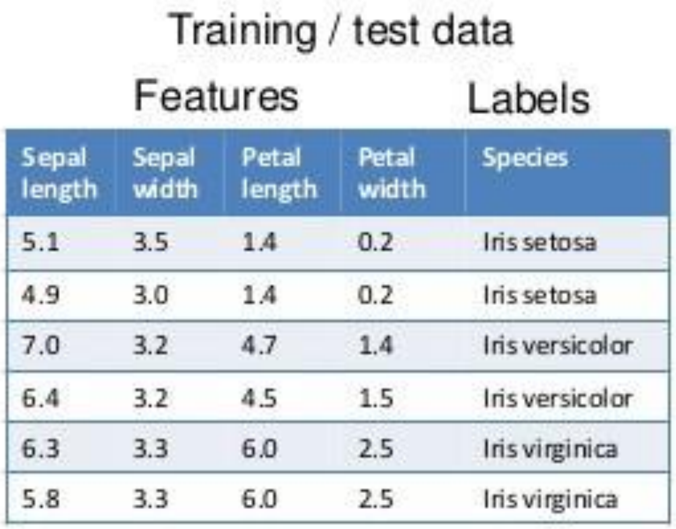
Data Representation

- Iris dataset - (Iris setosa, Iris virginica and Iris versicolor)

Supervised learning *classification* problem

(using the Iris flower data set)





Training / test data

Features				Labels
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	Iris setosa
4.9	3.0	1.4	0.2	Iris setosa
7.0	3.2	4.7	1.4	Iris versicolor
6.4	3.2	4.5	1.5	Iris versicolor
6.3	3.3	6.0	2.5	Iris virginica
5.8	3.3	6.0	2.5	Iris virginica

Data Representation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y^{[1]} \\ y^{[2]} \\ \vdots \\ y^{[n]} \end{bmatrix}$$

Input features

ML Terminology

Training example: A row in the table representing the dataset. Synonymous to an observation, training record, training instance, training sample (in some contexts, sample refers to a collection of training examples)

Feature: a column in the table representing the dataset. Synonymous to predictor, variable, input, attribute, covariate.

Targets: What we want to predict. Synonymous to outcome, output, ground truth, response variable, dependent variable, (class) label (in classification).

Output / prediction: use this to distinguish from targets; here, means output from the model.

Classes of Machine Learning Algorithms

Listed below are some common classes of machine learning algorithms:

- Generalized linear models (e.g., logistic regression)
- Support vector machines (e.g., linear SVM, RBF-kernel SVM)
- Artificial neural networks (e.g., multi-layer perceptrons)
- Tree- or rule-based models (e.g., decision trees)
- Graphical models (e.g., Bayesian networks)
- Ensembles (e.g., Random Forest)
- Instance-based learners (e.g., K-nearest neighbours)

Algorithm Categorization Schemes

To aid our conceptual understanding, each of the algorithms can be categorized into various categories.

- eager vs. lazy;
- batch vs. online;
- parametric vs. nonparametric;
- discriminative vs. generative.

Algorithm Categorization Schemes

Eager vs. lazy learners.

- Eager learners, also known as "memorizing learners", are algorithms that learn the training data by creating a model that fits the data perfectly. They process and analyze the entire training data set during the learning phase, and use this information to make predictions. Eager learners include decision trees, k-nearest neighbors (k-NN), and neural networks.
- Lazy learners, also known as "generalizing learners", are algorithms that do not analyze the training data during the learning phase. Instead, they store the training data and use a simple rule or a small set of examples to make predictions. Lazy learners only process the data when a new example is encountered, and use this information to make a prediction. Lazy learners include the k-NN algorithm and the instance-based learning.
- Lazy learners often have lower computational requirements than eager learners and are well suited for large datasets and problems with many features.

Algorithm Categorization Schemes

Batch vs. online learning

- Batch learners, also known as "offline learners," process the entire dataset at once. They learn from the entire dataset, and then update the model based on the entire dataset. Batch learning algorithms require the entire dataset to be available prior to starting the learning process and are typically executed periodically or as a one-time process. Batch learning algorithms are useful when the data is not changing frequently and when the computational resources are available. Examples of batch learners include gradient descent and batch gradient descent.
- Online learners, also known as "incremental learners," process and update the model based on new data as it becomes available. They learn from one example at a time and update the model after each example is processed. Online learning algorithms are well suited for problems where data is streaming in, and the model must adapt to changing conditions. They are also useful when the data is too large to fit in memory and when the computational resources are limited. Examples of online learners include stochastic gradient descent (SGD) and online gradient descent.

Algorithm Categorization Schemes

Parametric vs. nonparametric models.

- Parametric models are “fixed” models, where we assume a certain functional form for y=f(x).
- For example, linear regression can be considered as a parametric model with

  h(x) = w1 x1 + … + wm xm + b.

  - Nonparametric models are more “flexible” and do not have a pre-specified number of parameters.
  - In fact, the number of parameters grows typically with the size of the training set.
  - For example, a decision tree would be an example of a nonparametric model, where each decision node (e.g., a binary “True/False” assertion) can be regarded as a parameter.

Algorithm Categorization Schemes

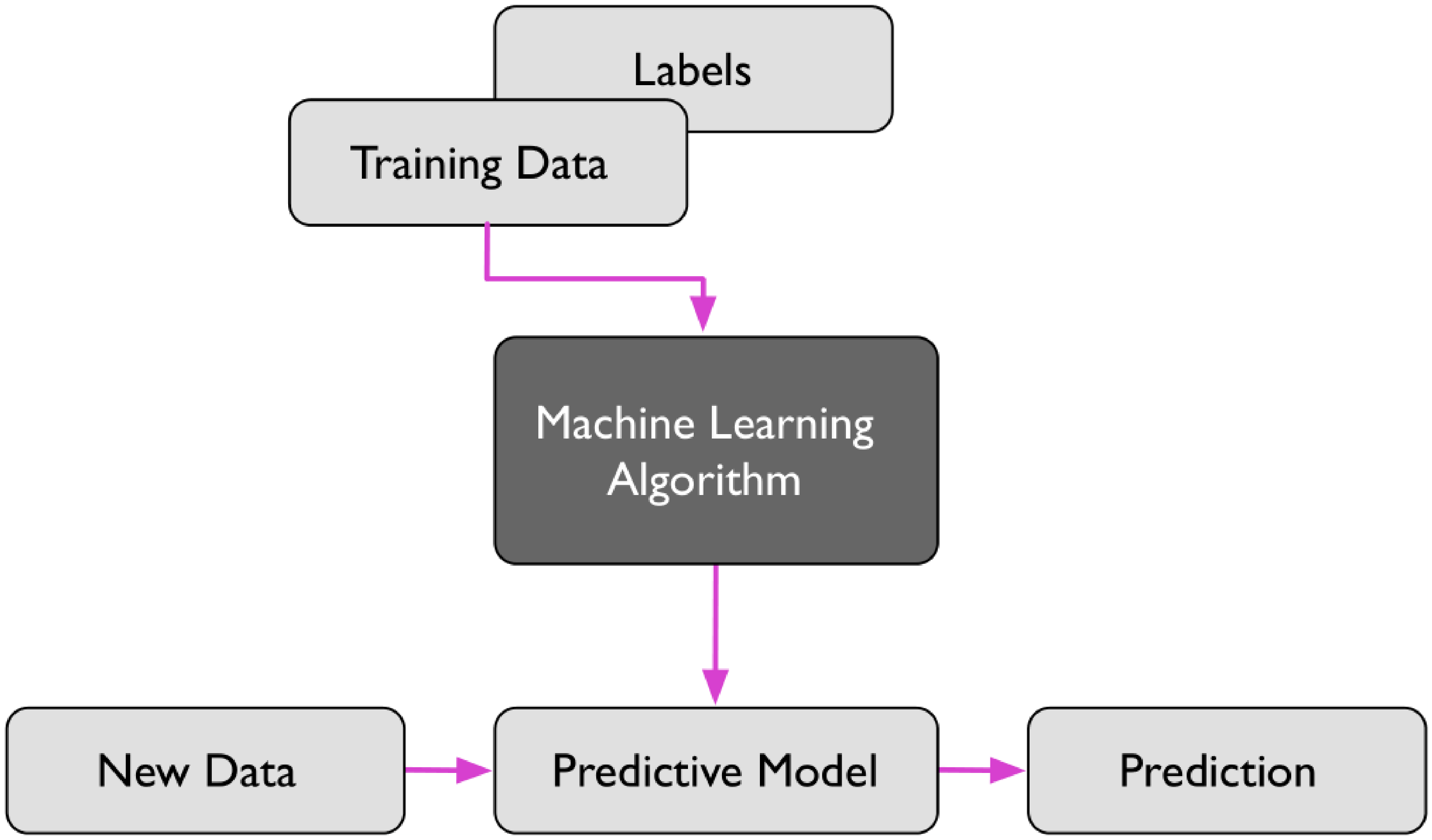
Discriminative vs. generative.

- Discriminative models focus on learning the decision boundary between different classes, while generative models focus on learning the underlying probability distribution of the data in each class.

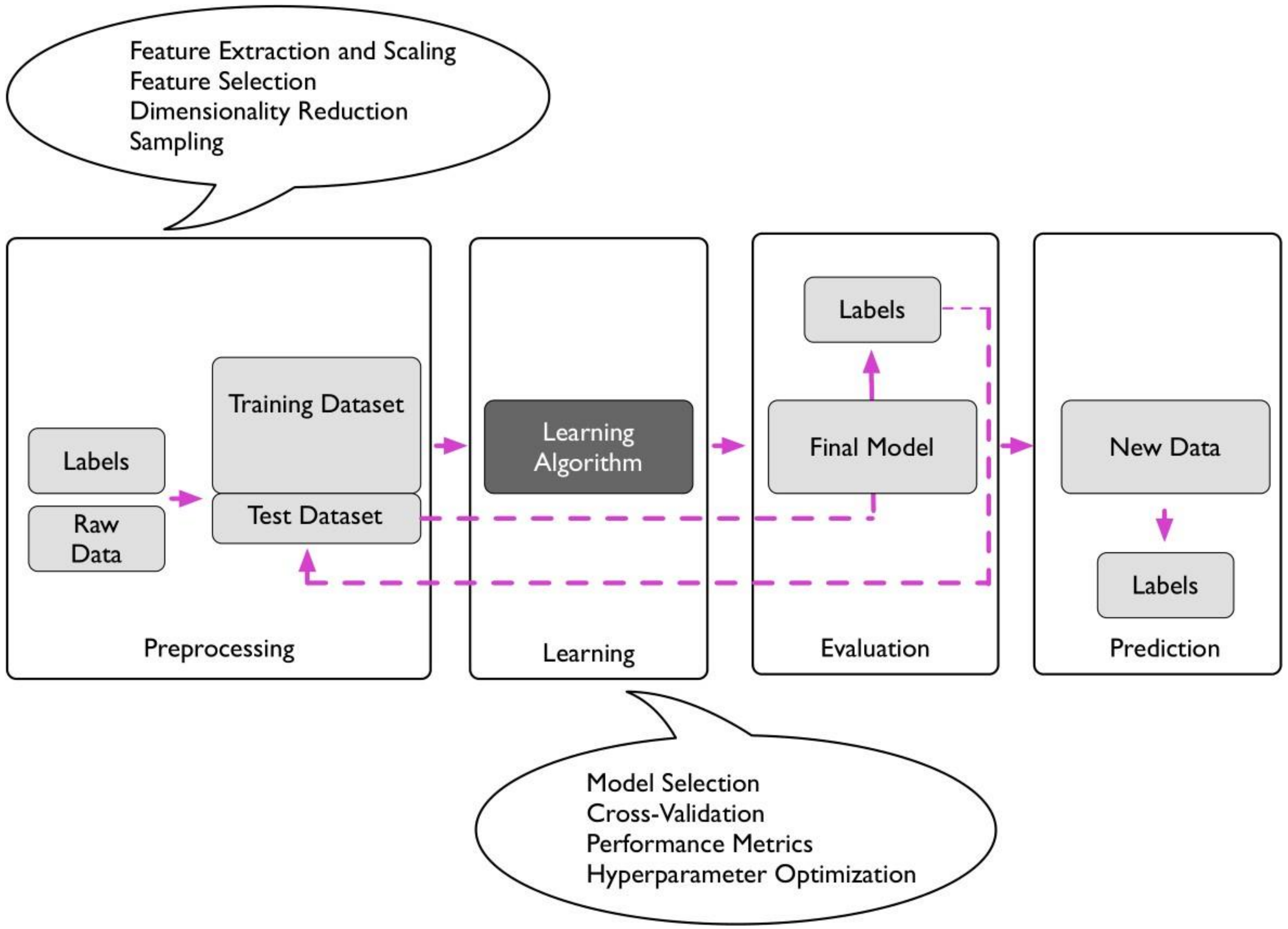
- Discriminative models, such as logistic regression and support vector machines, are typically used for classification tasks and are trained to predict the probability of a given input belonging to a certain class. These models are generally more efficient and require less training data than generative models.

- Generative models, such as Gaussian mixture models and hidden Markov models, are typically used for density estimation and unsupervised learning tasks. These models are trained to learn the underlying probability distribution of the data in each class and can be used for tasks such as data generation and anomaly detection. Generative models are generally more flexible and can fit more complex data distributions, but they may require more data to train.

Learning Workflow



Learning Workflow



Feature Extraction: In the context of machine learning, feature extraction is the process of identifying and extracting the most informative features from a dataset that can be used to train a model. The extracted features are used as input to a machine learning algorithm, rather than the raw data. This is because many machine learning algorithms perform better when working with a smaller number of relevant features than with a large amount of raw data.

Feature scaling: is a technique used in machine learning to standardize the range of independent variables or features of the data. It is also known as data normalization or standardization.

The goal of feature scaling is to ensure that all features of the data are on a similar scale, so that no feature dominates the others in the learning process. This is important because many machine learning algorithms, such as k-nearest neighbors and gradient descent, are sensitive to the scale of the input features.

Feature Selection: is the process of selecting a subset of the most relevant features from a larger set of features to be used as input to a model. The goal of feature selection in ML is to improve the performance, interpretability, and efficiency of the model by reducing the dimensionality of the data and eliminating irrelevant or redundant features.
Dimensionality reduction: is a technique used in machine learning to reduce the number of features (dimensions) in a dataset while retaining as much information as possible. This can be done for a variety of reasons, such as reducing the complexity of a model, improving the interpretability of the data, and decreasing the amount of computation required to process the data. Common techniques for dimensionality reduction include principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed stochastic neighbor embedding (t-SNE).

Difference between feature extraction and feature selection

- Feature extraction and feature selection are two different techniques used in machine learning and data analysis to improve the performance of a model.
- Feature extraction is a process of creating new features from existing data. This is done by applying mathematical operations or transformations to the original data, such as taking the square root or the logarithm. The goal of feature extraction is to create new features that are more informative or more easily processed by the machine learning algorithm.
- Feature selection is the process of selecting a subset of the most relevant features from the original data. This is done by evaluating the importance of each feature and selecting only the ones that contribute the most to the performance of the model. The goal of feature selection is to reduce the dimensionality of the data and improve the interpretability and efficiency of the model.
- In short, Feature extraction is creating new features and Feature selection is choosing the best features from the existing data.

Sampling: refers to the process of selecting a subset of data from a larger dataset for the purpose of training or testing a model. The goal of sampling is to select a representative subset of the data that accurately represents the underlying distribution of the entire dataset.

Model selection in machine learning refers to the process of choosing the best model from a set of candidate models to solve a particular problem. The goal is to select a model that has the highest performance and generalization ability while avoiding overfitting.

There are several different methods that can be used for model selection, including:

- Holdout method: The dataset is split into a training set and a test set, and multiple models are trained on the training set. The model that performs the best on the test set is chosen.
- K-fold cross-validation: The dataset is split into k subsets, and k models are trained, each time using a different subset as the test set and the remaining subsets as the training set. The model that performs the best on average is chosen.
- Bootstrapping: A large number of subsets are created by randomly sampling the data with replacement, and a model is trained on each subset. The model that performs the best on average is chosen.
- Hyperparameter tuning: A grid search or random search method is used to find the best combination of hyperparameters for a given model.

It's important to note that model selection should be done based on a combination of metrics such as accuracy, precision, recall, F1-score, AUC-ROC etc.

Hyperparameters

- Hyperparameters in Machine learning are those parameters that are explicitly defined by the user to control the learning process.
- Improve the learning of the model, and their values are set before starting the learning process of the model
- These are external to the model, and their values cannot be changed during the training process.
- Ex:
  - Learning rate for training a neural network
  - Train-test split ratio
  - Number of clusters in Clustering Algorithm

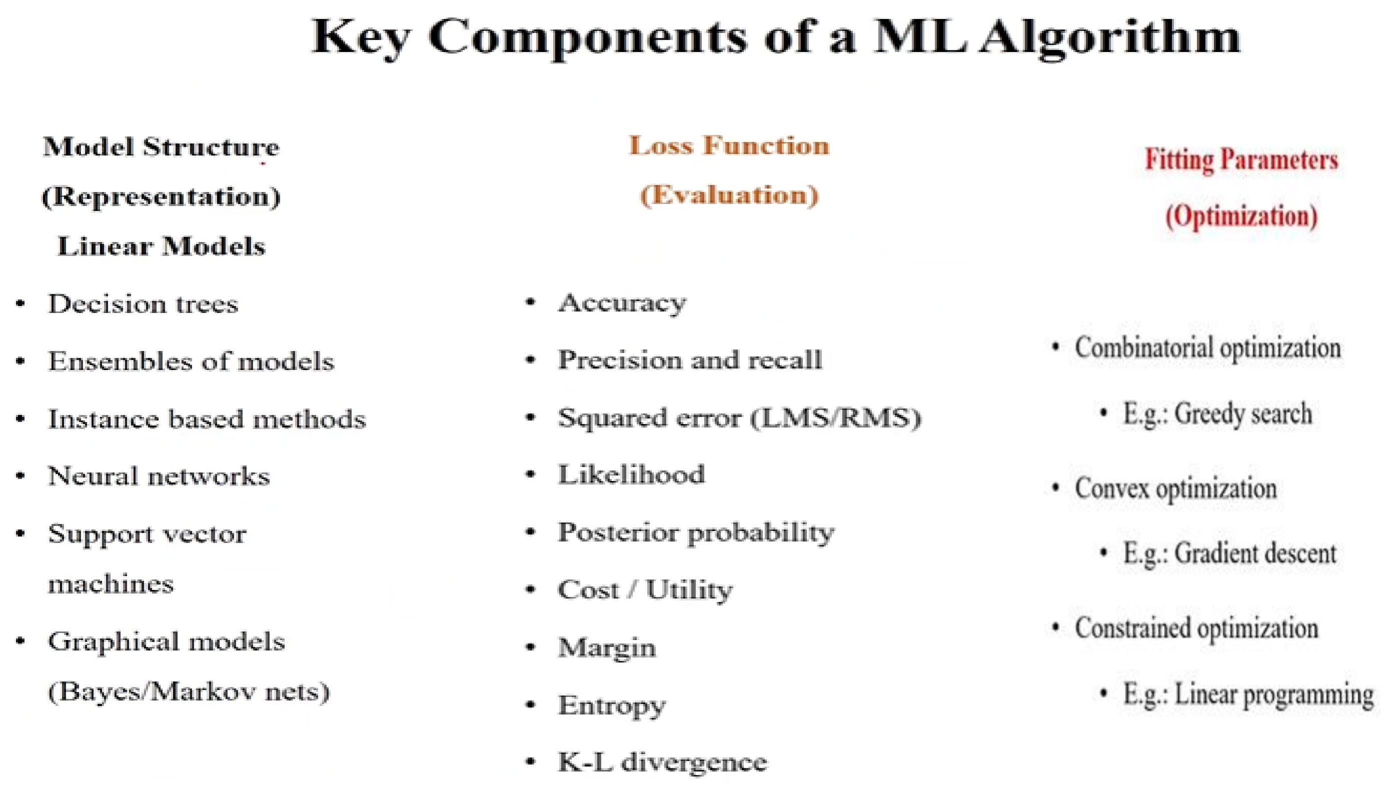
Key Elements of Machine Learning

Every machine learning algorithm has three key elements:

- **Representation**: how to represent knowledge using hypothesis function.
  - Examples include *decision trees*, sets of rules, instances, graphical models, neural networks, support vector machines, model ensembles, etc.,

- **Evaluation**: the way to evaluate candidate programs (hypotheses).
  - Examples include accuracy, precision and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence and etc.,

- **Optimization**: the way candidate programs are generated known as the search process.
  - E.g., Combinatorial optimization, convex optimization, constrained optimization.



Key Components of a ML Algorithm

Model Structure

(Representation)

Linear Models

- Decision trees
- Ensembles of models
- Instance based methods
- Neural networks
- Support vector machines
- Graphical models (Bayes/Markov nets)

Loss Function

(Evaluation)

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

Fitting Parameters

(Optimization)

- Combinatorial optimization
  - E.g.: Greedy search
- Convex optimization
  - E.g.: Gradient descent
- Constrained optimization
  - E.g.: Linear programming