Foundation of Machine Learning

Day 1: An Introduction of Machine Learning Algorithms

Sieu Tran

Introduction

Name **+**

Current Project **+**

What You Hope to Gain

Agenda

What we are going to go over today!

We will focus mostly on Machine Learning Foundation and Algorithms

1 Introduction

2 Key Concepts

3 Family of Machine Learning Algorithms

3.1 Regression

3.2 Decision Tree

3.3 Ensemble Learning

3.4 Bayesian Algorithm

3.5 Clustering Algorithm

3.6 Dimensional Reduction

3.7 Deep Learning Basics

3.8 Rule-based systems

4 Key Takeaway

5 Practice Scenario

6 Coding Exercises

7 Closing

Machine Learning

Grew out of work in AI

New capability for computers

Examples:

- Database mining
- Large datasets from growth of automation/web.
  Example: Web click data, medical records, biology, engineering
- Applications can't program by hand.
  Example: Autonomous helicopter, handwriting recognition, most of Natural Language Processing (NLP), Computer Vision.
- Self-customizing programs
  Example: Amazon, Netflix product recommendations
- Understanding human learning (brain, real AI).

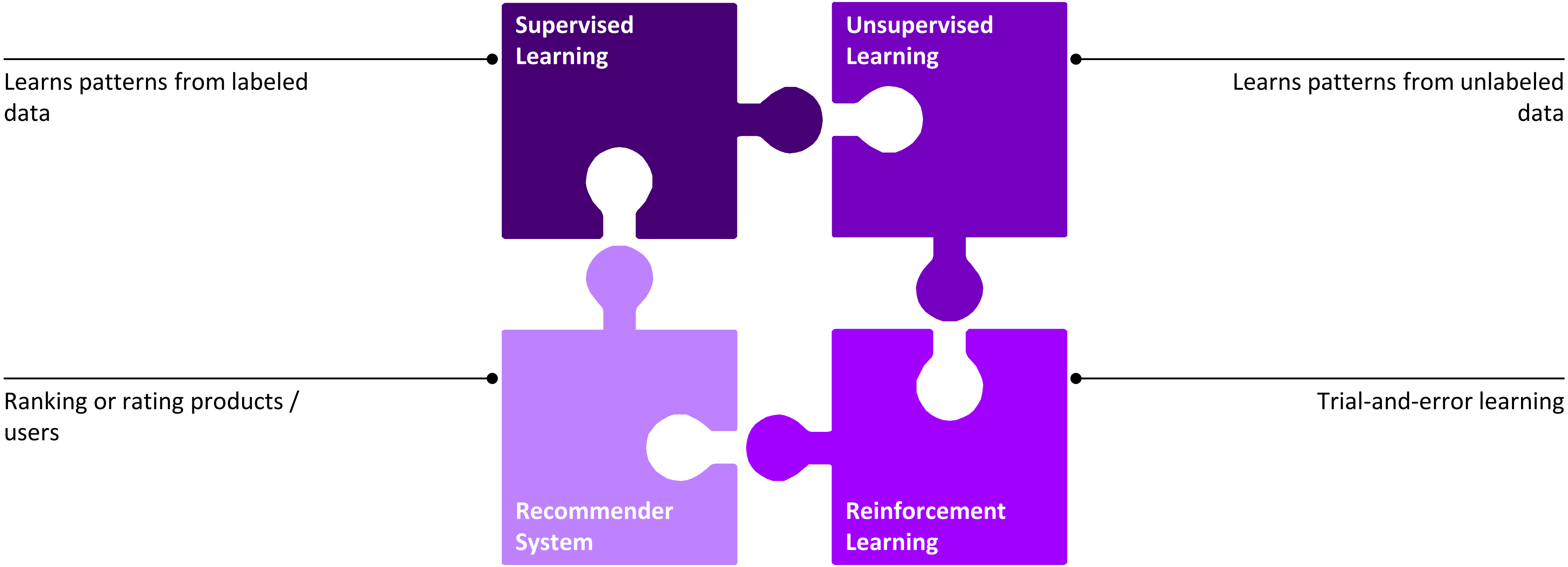
Problem

"A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

- Classifying emails as spam or not spam.
- Watching you label emails as spam or not spam.
- The number (or fraction) of emails correctly classified as spam/not spam.
- None of the above—this is not a machine learning problem.

Machine Learning



Supervised Learning

Learns patterns from labeled data

Unsupervised Learning

Learns patterns from unlabeled data

Recommender System

Ranking or rating products / users

Reinforcement Learning

Trial-and-error learning

Key Concepts

Cost, Loss, and Objective Function

Loss Function

Loss function is usually a function defined on a data point, prediction and label, and measures the penalty. For example:

- Square loss: $loss(f(x_i|\theta), y_i = (f(x_i|\theta) - y_i)^2$
- Hinge loss: $loss(f(x_i|\theta), y_i) = \max(0, 1 - f(x_i|\theta)y_i)$
- 0/1 loss: $loss(f(x_i|\theta), y_i) = 1 \Leftrightarrow f(x_i|\theta) \neq y_i$

Cost Function

Cost function is a more generalized loss function. It might be a sum of loss functions over your training set and model penalty. For example:

- Mean Squared Error: $MSE(\theta) = \frac{1}{N}\sum_{i=0}^{N}(f(x_i|\theta) - y_i)^2$
- SVM cost function: $SVM(\theta) = \| \theta \|^2 + C\sum_{i=1}^{N}\zeta_i$

Objective Function

Objective function is the most general term for any function that you optimize during training. For example:

- A probability of generating training set in maximum likelihood approach.

Interpretability vs. Explainability

Interpretability

The ability to determine the cause and effect of a model

If a model can take the inputs, and routinely get the same outputs, the model is interpretable:

- If you overeat your pasta at dinnertime and you always have troubles sleeping, the situation is interpretable.
- If all 2016 polls showed a Democratic win and the Republican candidate took office, all those models showed low interpretability.

The Blackbox Model of Machine Learning

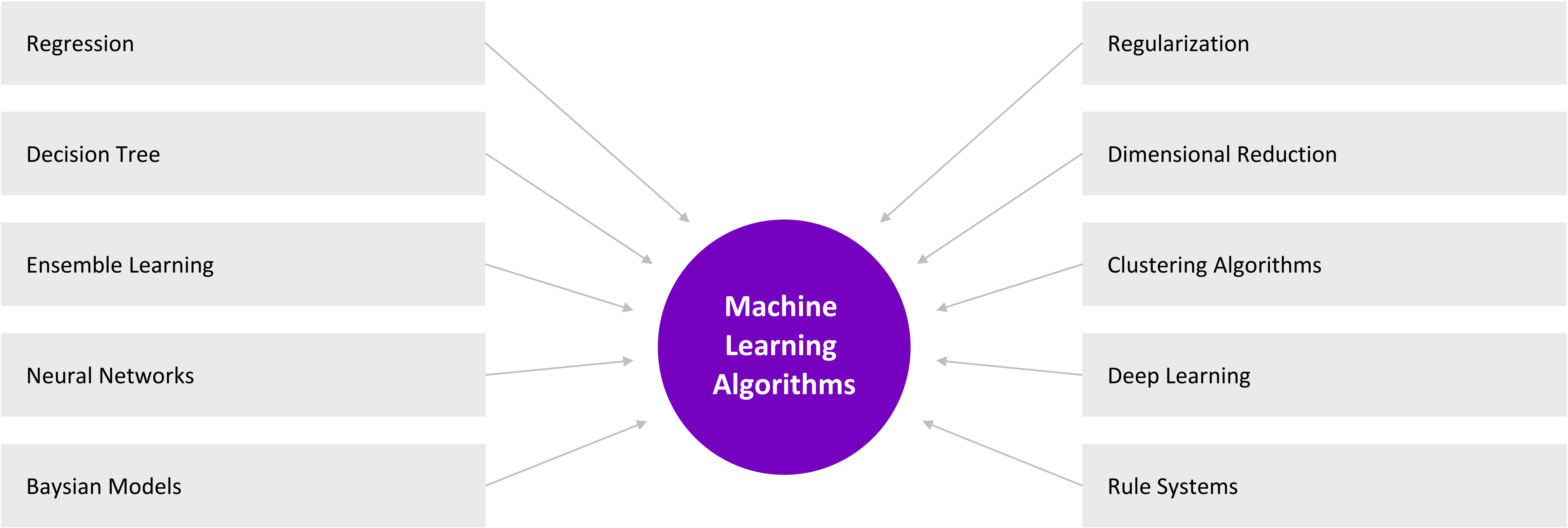
Explainability

The knowledge of both what a node represents and its important in the model’s performance.

Weighing the importance of one node in determining the overall score of the output:

- If that signal is high, that node is significant to the model’s overall performance.
- If that signal is low, the node is insignificant.

Machine Learning Algorithms



Regression

Decision Tree

Ensemble Learning

Neural Networks

Baysian Models

Machine Learning Algorithms

Regularization

Dimensional Reduction

Clustering Algorithms

Deep Learning

Rule Systems

Terminology

A few important terminology

Supervised learning

- Dependent Variable
- Independent Variable

Dynamic Concepts

- Learning Rate
- Fine-tuning
- Training
- Testing
- Cross validation
- Error
- Gradient Descent

Relationship

- Linear
- Non-linear

Modeling Strategy

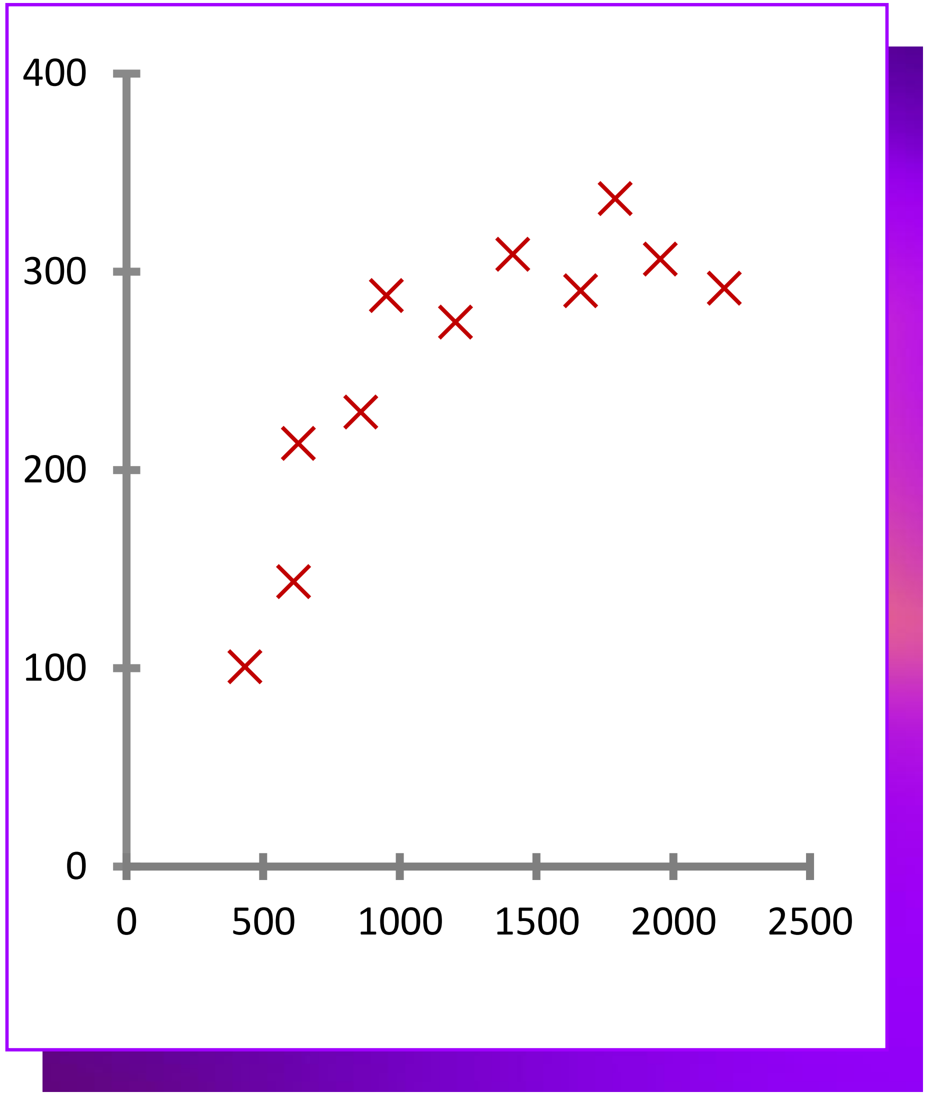
- Parametric
- Non-parametric

Evaluation metrics

- Accuracy
- Precision
- Recall
- AUROC
- AUPRC
- …

MORE INFO ON EVALUATION NEXT WEEK

Regression



Housing Price Prediction

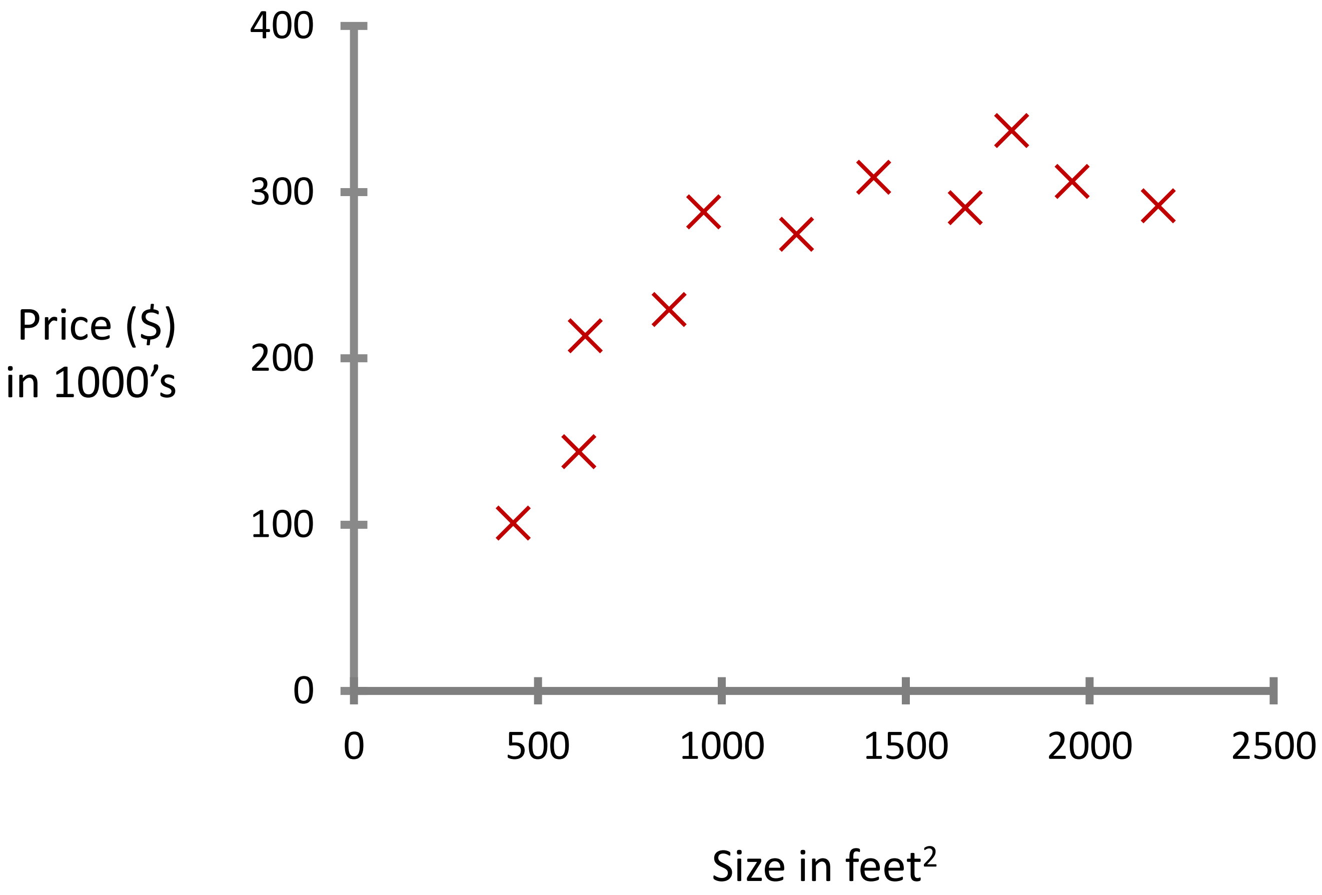
Definitions:

Supervised Learning

- “Right answers” given

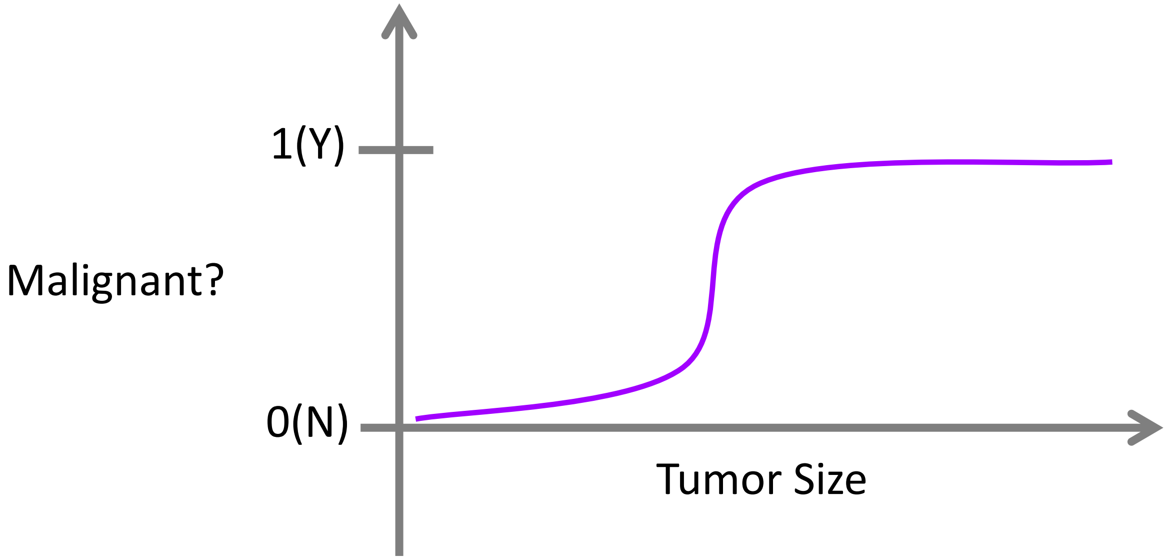
Regression

- Predict continuous valued output (price)



Copyright © 2020 Accenture. All rights reserved.

Breast Cancer Prediction



Classification

- Discrete valued output (0 or 1)

Problem Definition

Key facts (situation): You're running a company, and you want to develop learning algorithms to address each of two problems.

Need for change (complication):

Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.

Problem 2: You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.

Key question: Should you treat these as classification or as regression problems?

1. Treat both as classification problems.
2. Treat problem 1 as a classification problem, problem 2 as a regression problem.
3. Treat problem 1 as a regression problem, problem 2 as a classification problem.
4. Treat both as regression problems.

Why Use Regression?

Advantages

- Very easy to implement and not computationally expensive
- High interpretability and high explainability
- Linear regression performs exceptionally well for linearly separable data
- Easier to implement, interpret and efficient to train
- It handles overfitting well using dimensionally reduction techniques, regularization, and cross-validation
- Extrapolates beyond a specific data set

VS

Disadvantages

- In linear regression, the assumption of linearity between dependent and independent variables
- It is often quite prone to noise and overfitting
- Quite sensitive to outliers
- It is prone to multicollinearity

Notable Regression Algorithms

- **Ordinary Least Squares Regression (OLS)** - Assume variables have a linear relationship
- **Logistic Regression** – Assume variables have an exponential relationship (binary classification problem)
- **Stepwise Regression** – The step-by-step iterative construction of a regression model that involves the selection of independent variables to be used in a final model
- **Multivariate Adaptive Regression Spline (MARS)** – A nonparametric method which models nonlinearities and interactions between variables
- **Locally Estimated Scatterplot Smoothing (LOESS)** – A nonparametric method for smoothing a series of data in which no assumptions are made about the underlying structure of the data
- **Generalized Linear Regression (GLM)** – Combines aspect of both linear and logistic regressions

Decision Tree

What is a Decision Tree?

Problem:

Consider a very basic example that uses titanic data set for predicting whether a passenger will survive or not.

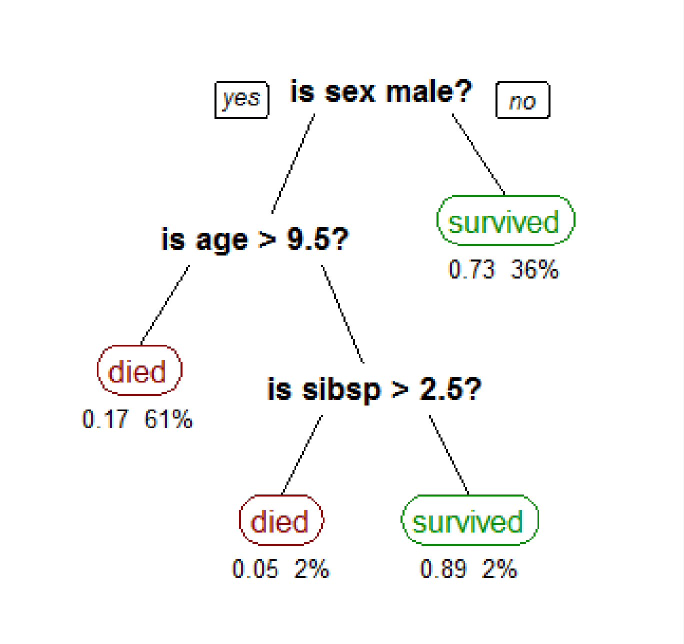
Description:

- **Bold text in black** represents a condition/internal node, based on which the tree splits into branches/ edges.
- The end of the branch that does not split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text, respectively.

Value to Clients/ Benefits:

- Regression trees are represented in the same manner, just they predict continuous values like price of a house.
- In general, Decision Tree algorithms are referred to as Classification and Regression Trees (CART).

Visual Representation:



How Do Decision Trees Work?

	Phase 1	Phase 2	Phase 3	Phase 4
	Initiation	**Splitting**	**Termination**	**Refinement**
Key Definition	**Information Gain and Entropy**	**Recursive Binary Splitting and Cost of Split**	**Termination Condition**	**Pruning**
Details	• Impurity/Entropy (informal) measures the level of impurity in a group of examples $E(S) = \sum_{k=1}^{N} -p_k \log_2(p_k)$ • Information gain tells us how important a given attribute of the feature vectors is. **Information Gain** = E(parent) – [Mean E(children)]	• All the features are considered and different split points are tried and tested using a cost function. • Regression cost: $G = \sum_{k=1}^{N} (y_k - \hat{y}_k)^2$ • Classification cost: $G = \sum_{k=1}^{N} p_k(1 - p_k)$	• One way is to set a minimum number of training inputs to use on each leaf.	• It involves removing the branches that make use of features having low importance. • This reduces the complexity of tree, and thus increasing its predictive power by reducing overfitting.

Why Use Decision Tree?

Advantages

- Simple to understand, interpret, visualize
- Decision trees implicitly perform variable screening or feature selection
- Can handle both numerical and categorical data. Can also handle multi-output problems
- Decision trees require relatively little effort from users for data preparation
- Nonlinear relationships between parameters do not affect tree performance

VS

Disadvantages

- Decision-tree learners can create over-complex trees that do not generalize the data well.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated
- Decision tree learners create biased trees if some classes dominate

Notable Decision Tree Algorithms

CART

- Classical decision three

Iterative Dichotomiser 3 (ID3)

- Algorithm iteratively (repeatedly) dichotomizes (divides) features into two or more groups at each step;
- Used for classification problems with nominal features only.
- Only use Entropy and Information Gain

C4.5

- Handling both continuous and discrete attributes;
- Has pruning step;
- Multiple cost functions;
- Handles missing data

C5.0

- Faster than C4.5
- Allows for boosting

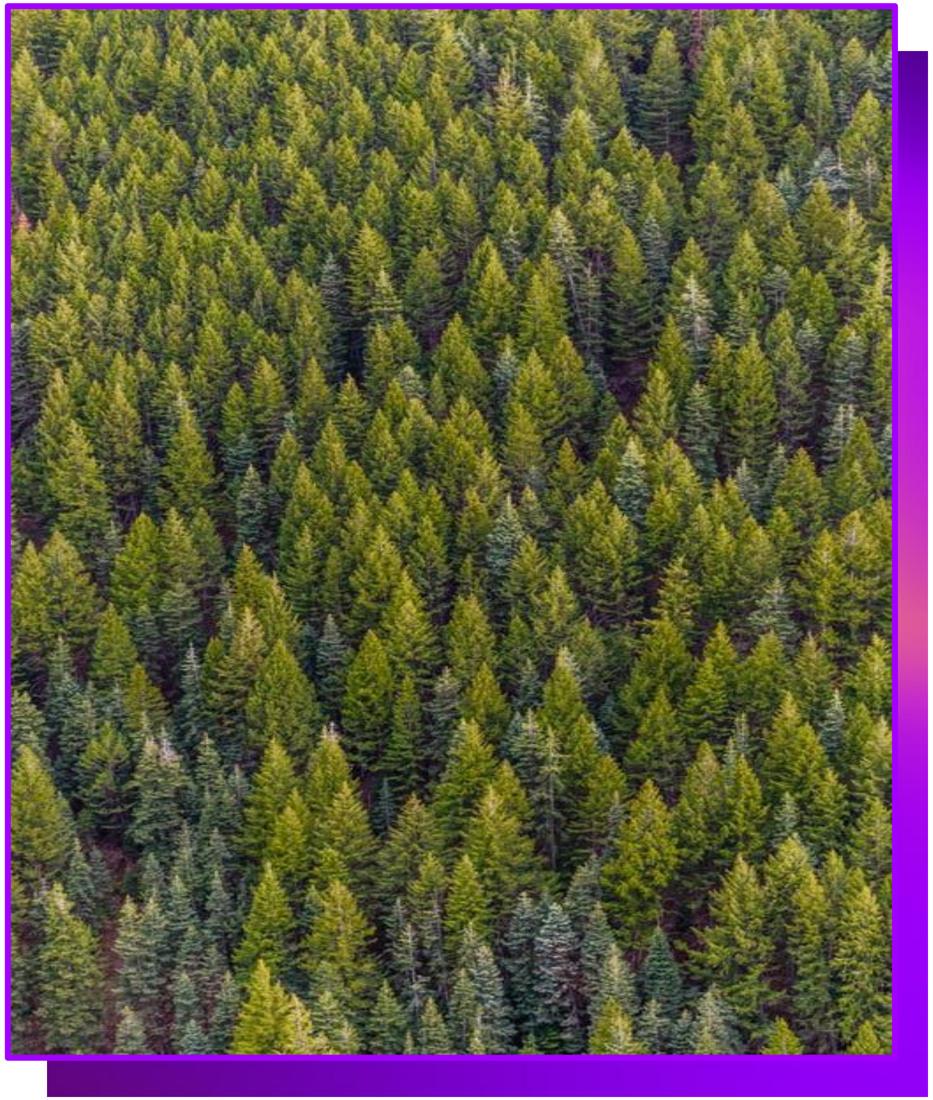
Decision Stump

- Binary classifier with one single split

M5

- Binary regression tree model where the last nodes are the linear regression functions that can produce continuous numerical attributes

Ensemble Learning



What is Ensemble Learning?

Definition:

- Ensemble learning is a combination of several machine learning models in one problem. These models are known as weak learners.

Intuition:

- The intuition is that when you combine several weak learners, they can become strong learners.

Ensemble Learning

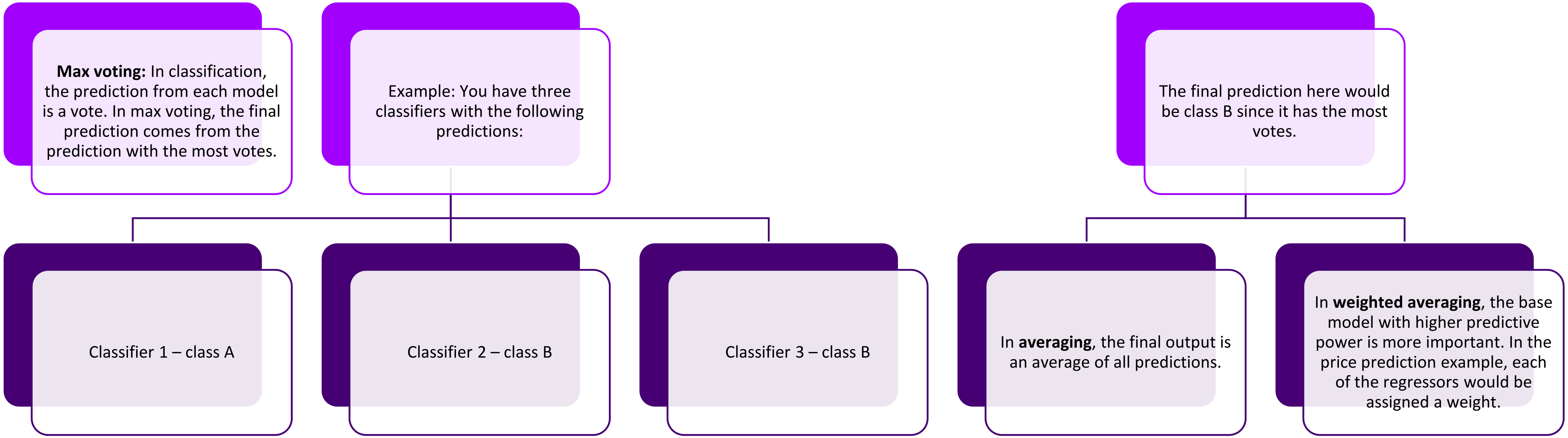
Process:

- Each weak learner is fitted on the training set and provides predictions obtained.

Prediction Method:

- The final prediction result is computed by combining the results from all the weak learners.

How Does Ensemble Learning Work?



Max voting: In classification, the prediction from each model is a vote. In max voting, the final prediction comes from the prediction with the most votes.

Example: You have three classifiers with the following predictions:

Classifier 1 – class A

Classifier 2 – class B

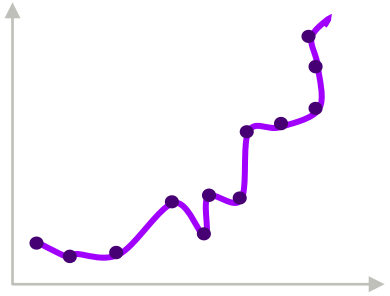
Classifier 3 – class B

The final prediction here would be class B since it has the most votes.

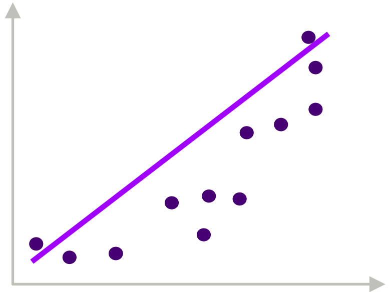
In **averaging**, the final output is an average of all predictions.

In **weighted averaging**, the base model with higher predictive power is more important. In the price prediction example, each of the regressors would be assigned a weight.

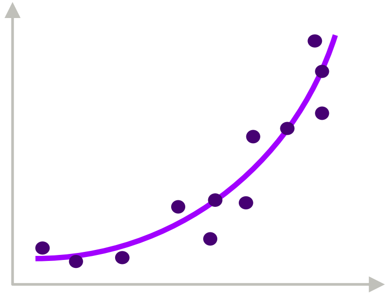
Bias-Variance Tradeoff



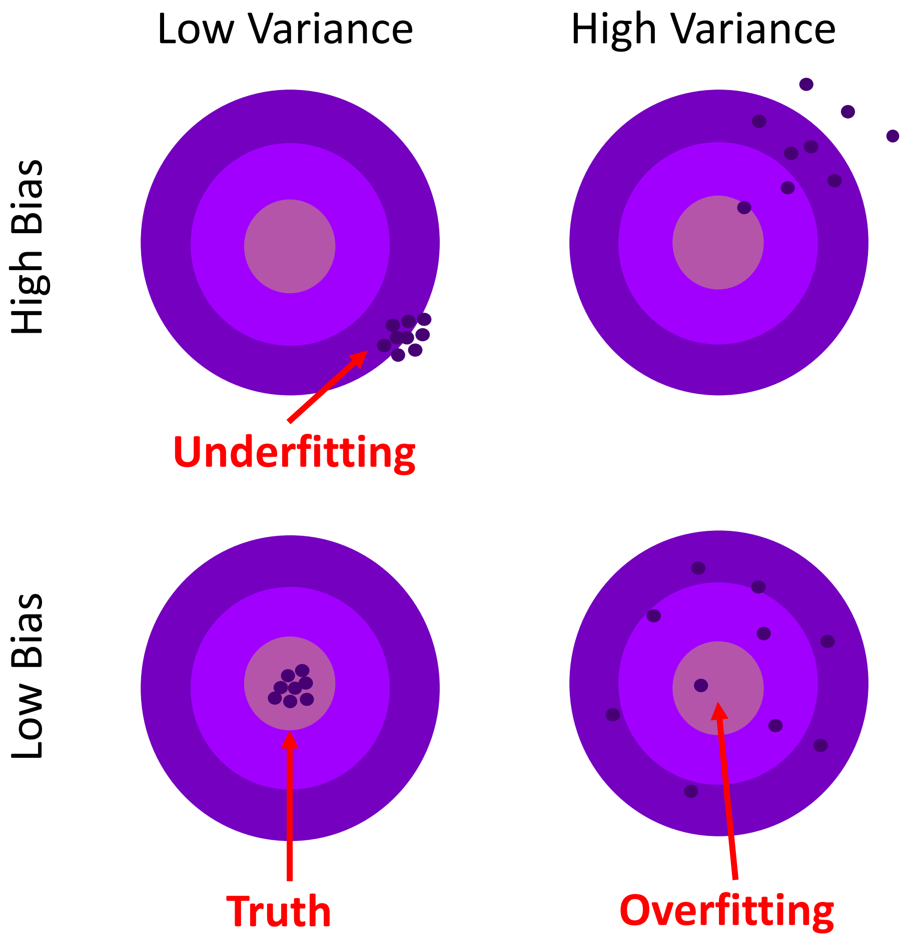
Overfitting



Underfitting



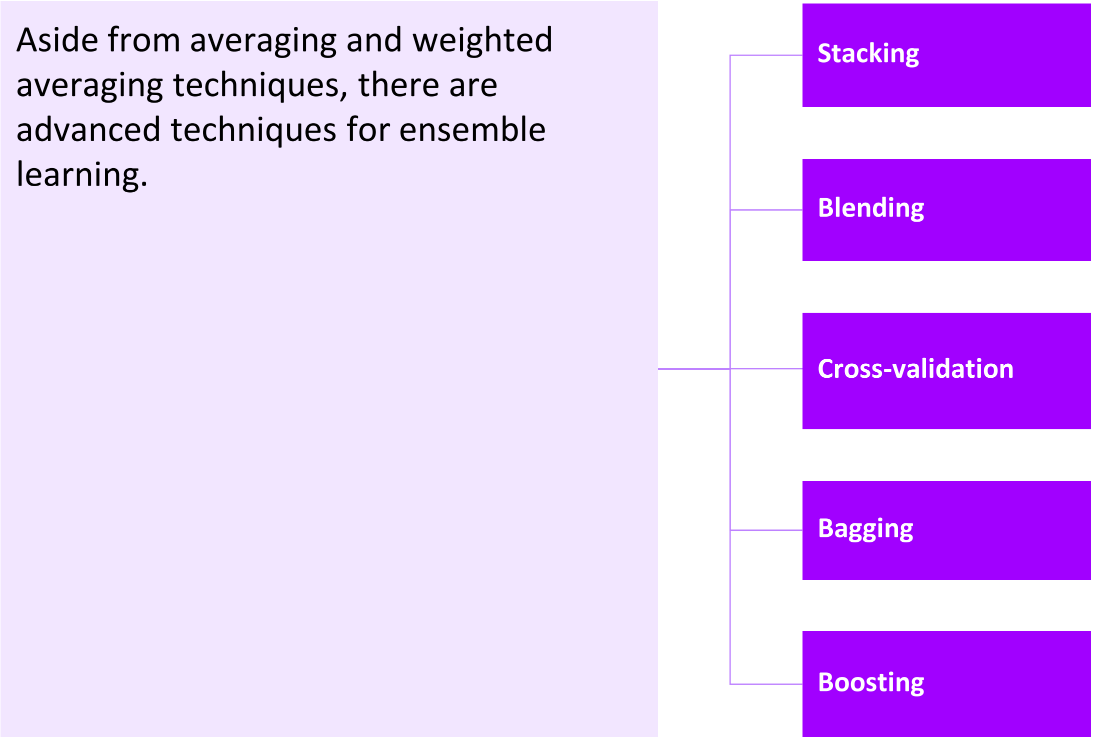
Good Balance



$$\text{Error}(x) = \left(E\left[\hat{f}(x)\right] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right] + \sigma_e^2$$

$$\text{Error(x)} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Advanced Ensemble Learning



Overview

Aside from averaging and weighted averaging techniques, there are advanced techniques for ensemble learning.

Technique	Description
Stacking	The process of combining various estimators in order to reduce their biases. Predictions from each estimator are stacked together and used as input to a final estimator that computes the final prediction.
Blending	Uses a holdout set from the training set to make predictions.
Cross-validation	More solid on stacking than blending. It's calculated over more folds, compared to using a small hold-out dataset in blending.
Bagging	Takes random samples of data, builds learning algorithms, and uses the mean to find bagging probabilities. It's also called bootstrap aggregating. Bagging aggregates the results from several models in order to obtain a generalized result.
Boosting	A machine learning ensemble technique that reduces bias and variance by converting weak learners into strong learners. The weak learners are applied to the dataset in a sequential manner.

Why Use Ensemble Learning?

Preliminary High-Level Business Case

Target Benefits ($)

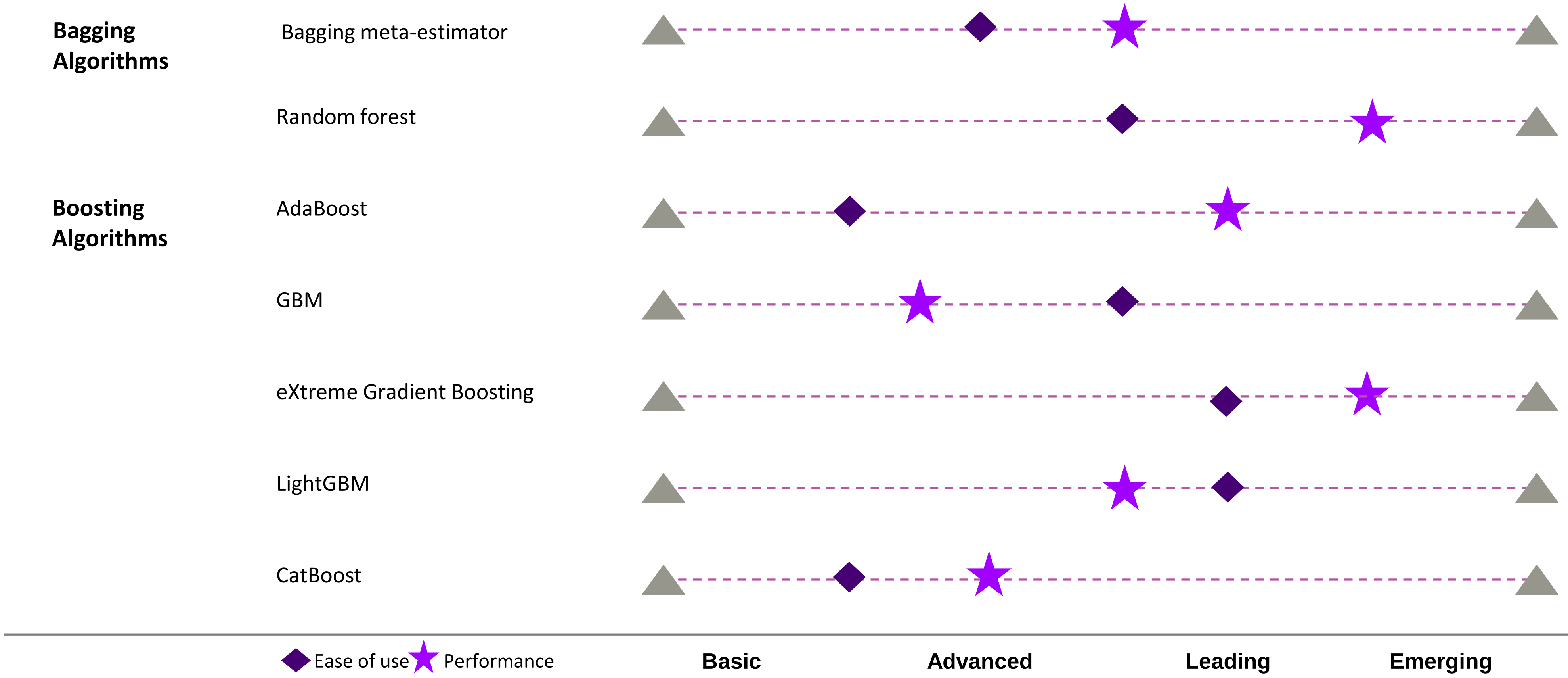
- Ensures reliability of the predictions.

Ensures the stability/robustness of the model.

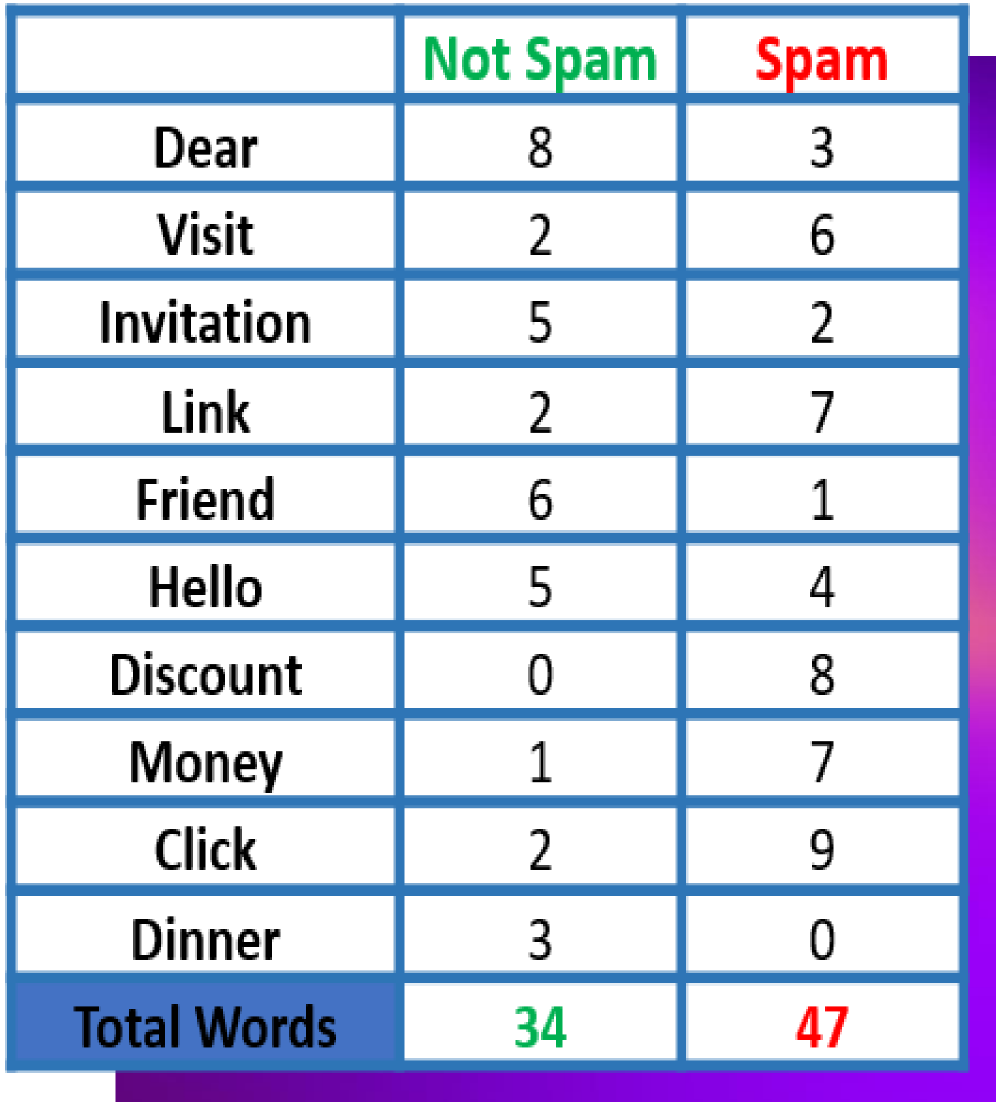
- Combining several weak models can result in what we call a strong learner.
- Use ensemble methods to combine different models in two ways: either using a single base learning algorithm that remains the same across all models (a homogeneous ensemble model) or using multiple base learning algorithms that differ for each model (a heterogeneous ensemble model).
- Generally speaking, ensemble learning is used with decision trees, since they're a reliable way of achieving regularization.

Notable Algorithms

Note: ranking are relative and subject to change

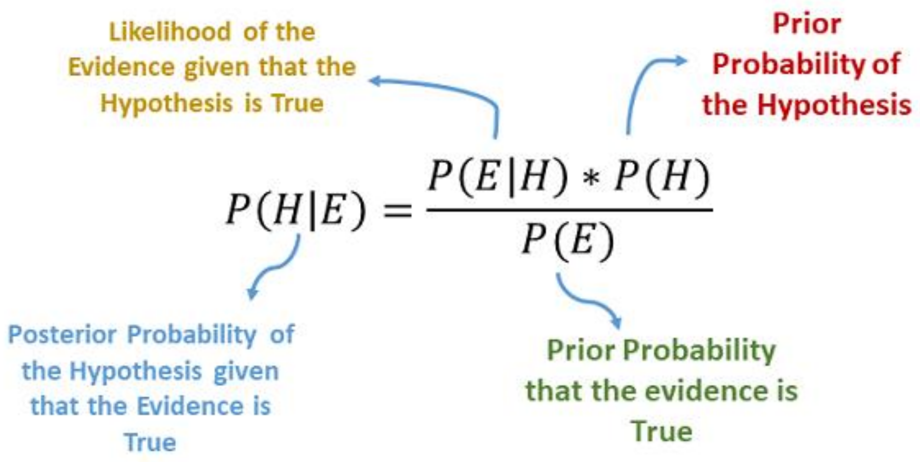


Bayesian Algorithms



	Not Spam	Spam
Dear	8	3
Visit	2	6
Invitation	5	2
Link	2	7
Friend	6	1
Hello	5	4
Discount	0	8
Money	1	7
Click	2	9
Dinner	3	0
Total Words	34	47

What are Bayesian Algorithms?



$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
Posterior Probability of the Hypothesis given that the Evidence is True
Prior Probability that the evidence is True

Definitions and Intuitions:

1. A family of algorithms where all of them share a common principle: every pair of features being classified is independent of each other.
2. Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem.
3. Bayes's formula provides relationship between **P(H|E)** and **P(E|H)**

Examples:

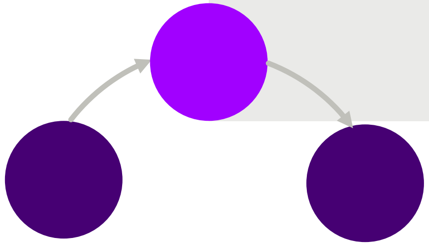
Name		Basic Idea	Details
Naive Bayes	⚖	Assumes that each of the features it uses are conditionally independent of one another given some class	• Useful for very large data sets Since the algorithm has an assumption of independence, you do lose the ability to exploit the interactions between features. • Gaussian Naïve Bayes, we assume that the distribution of probability is Gaussian (normal). Hence, Gaussian Naive Bayes is used in cases when all our features are continuous. • It is usually used when all our features are continuous
Multinomial Naive Bayes	⚙	Each feature has a multinomial distribution	• It's used when we have discrete data In text learning, • This algorithm is mostly used for document classification problem • The features/predictors used by the classifier are the frequency of the words present in the document.
Averaged One-Dependence Estimators (AODE)	🔄	A semi-naive Bayesian Learning method	• Average over all of the models in which all attributes depend upon the class and a single other attribute. • Using it for nominal data is computationally more efficient than regular Naïve Bayes, and achieves very low error rates.
Bayesian Belief Network (BBN)	👪	A probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph	• Enable us to model and reason about uncertainty • The most important use of BBNs is in revising probabilities in the light of actual observations of events • Can be used to understand what caused a certain problem, or the probabilities of different effects given an action in areas like computational biology and medicine for risk analysis and decision support.

Examples:

Bayesian Network (BN)

A type of Probabilistic Graphical Model.

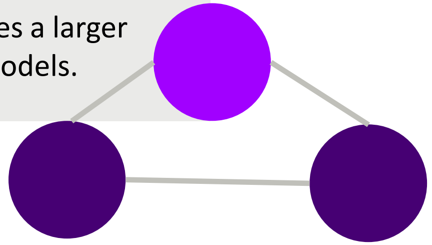
- These networks can be used for predictions, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision making under uncertainty.
- The goal of these networks is to model conditional dependence, and therefore causation.

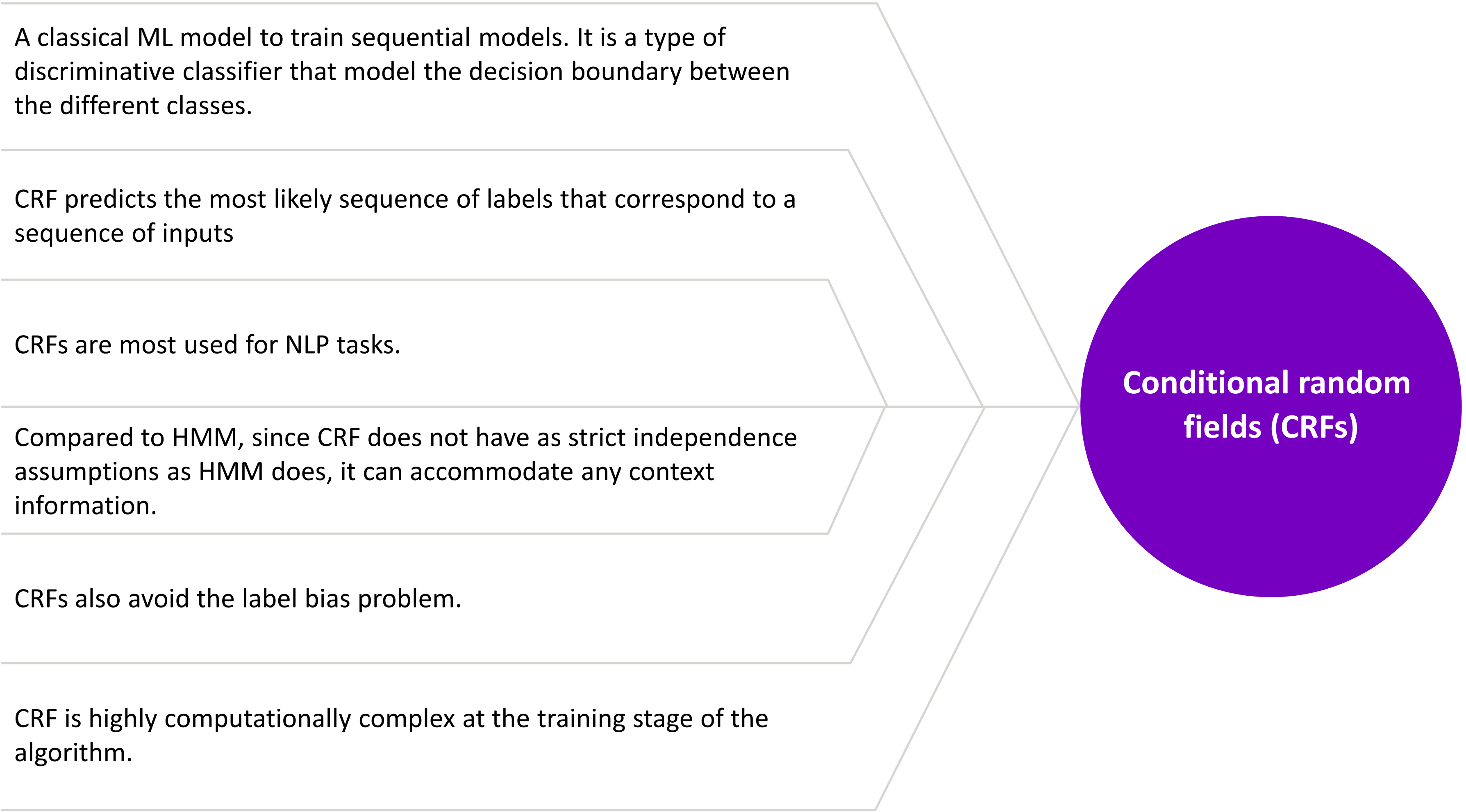


Hidden Markov models (HMM)

A class of probabilistic graphical model that allow us to predict a sequence of unknown (hidden) variables from a set of observed variables.

- HMM are known for their use in in reinforcement learning and temporal pattern recognition such as handwriting, speech, part-of-speech tagging, gesture recognition, and bioinformatics.
- HMM is suitable to be used in application that dealing with recognizing something based on sequence of feature.
- HMMs can be used to model processes which consist of different stages that occur in definite (or typical) orders.
- HMM needs to be trained on a set of seed sequences and generally requires a larger seed than the simple Markov models.





Conditional random fields (CRFs)

A classical ML model to train sequential models. It is a type of discriminative classifier that model the decision boundary between the different classes.

CRF predicts the most likely sequence of labels that correspond to a sequence of inputs

CRFs are most used for NLP tasks.

Compared to HMM, since CRF does not have as strict independence assumptions as HMM does, it can accommodate any context information.

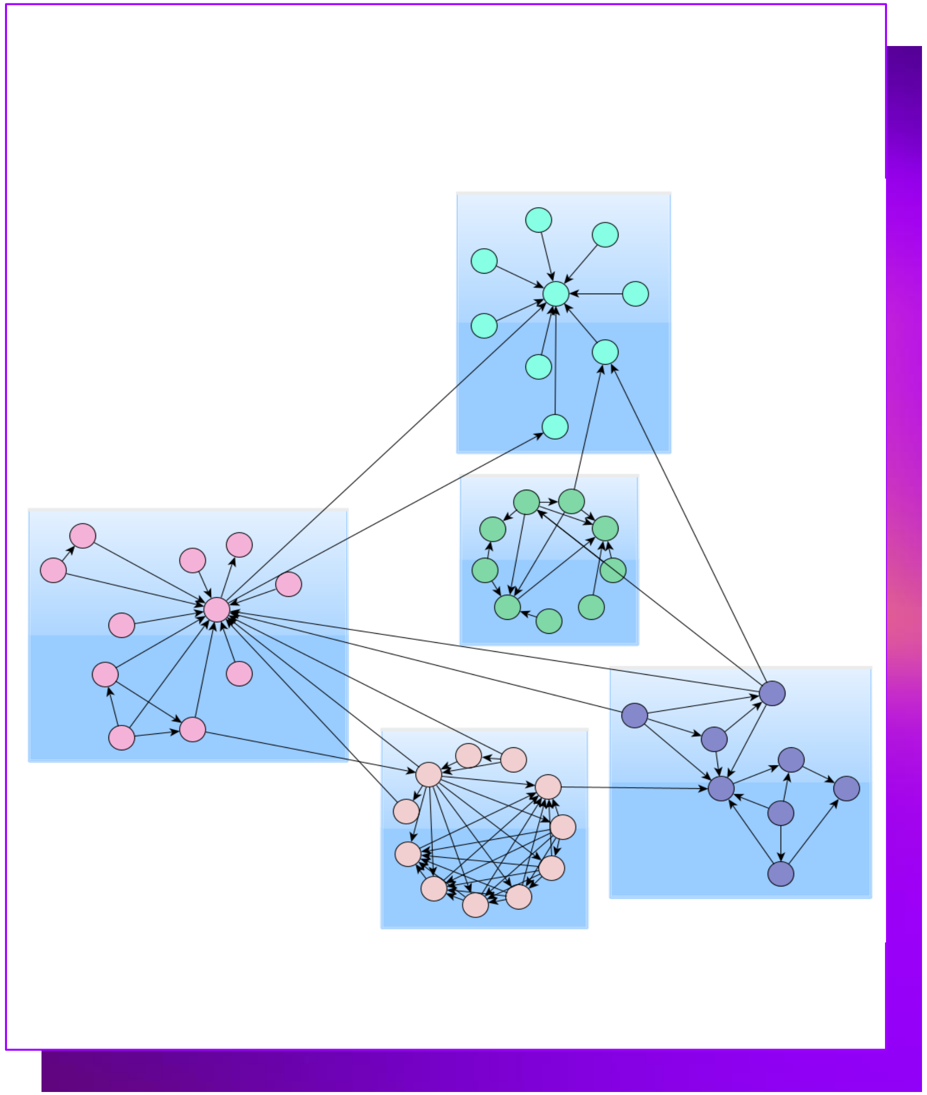
CRFs also avoid the label bias problem.

CRF is highly computationally complex at the training stage of the algorithm.

Flash Quiz

1. What is Linear Regression?
2. How does a Non-Linear regression analysis differ from Linear regression analysis?
3. Compare Linear Regression and Decision Trees
4. How would you deal with Overfitting in Linear Regression models?
5. What does *Random* refer to in Random Forest?
6. Why Random Forest models are considered not interpretable?

Cluster Algorithms



What is a clustering algorithm?

1.

Clustering is an unsupervised machine learning task.

2.

Give the algorithm a lot of input data with no labels and let it find any groupings in the data it can.

3.

A cluster is a group of data points that are similar to each other based on their relation to surrounding data points.

4.

Clustering is used for things like feature engineering or pattern discovery.

5.

When you're starting with data you know nothing about, clustering might be a good place to get some insight.

Type of Clustering Algorithms

DENSITY-BASED CLUSTERING

- Data is grouped by areas of high concentrations of data points surrounded by areas of low concentrations of data points.
- The clusters can be any shape and aren't constrained to expected conditions
- Ignore outliers

DISTRIBUTION-BASED CLUSTERING

- All of the data points are considered parts of a cluster based on the probability that they belong to a given cluster.
- There is a center-point, and as the distance of a data point from the center increases, the probability of it being a part of that cluster decreases.
- If the distribution in the data is unknown, consider a different type of algorithm

CENTROID-BASED CLUSTERING

- Sensitive to the initial parameters, but fast and efficient.
- These types of algorithms separate data points based on multiple centroids in the data.
- Each data point is assigned to a cluster based on its squared distance from the centroid.
- This is the most commonly used type of clustering.

HIERARCHICAL-BASED CLUSTERING

- Typically used on hierarchical data, such as those from a company database or taxonomies.
- It builds a tree of clusters so everything is organized from the top-down.
- This is more restrictive than the other clustering types, but it's perfect for specific kinds of data sets.

When to use clustering?

1. When you have a set of **unlabeled data**,
2. **Anomaly detection** to try and find outliers in your data.
3. You **are not sure of what features to use** for your machine learning model
4. Clustering is especially useful for **exploring data** you know nothing about.
5. Some real-world applications of clustering include **fraud detection** in insurance, categorizing books in a library, and customer segmentation in marketing.

Notable Clustering Algorithms

HIERARCHICAL CLUSTERING

Agglomerative: A “bottom-up” approach where each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

Divisive: A “top-down” approach where all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

NONHIERARCHICAL CLUSTERING

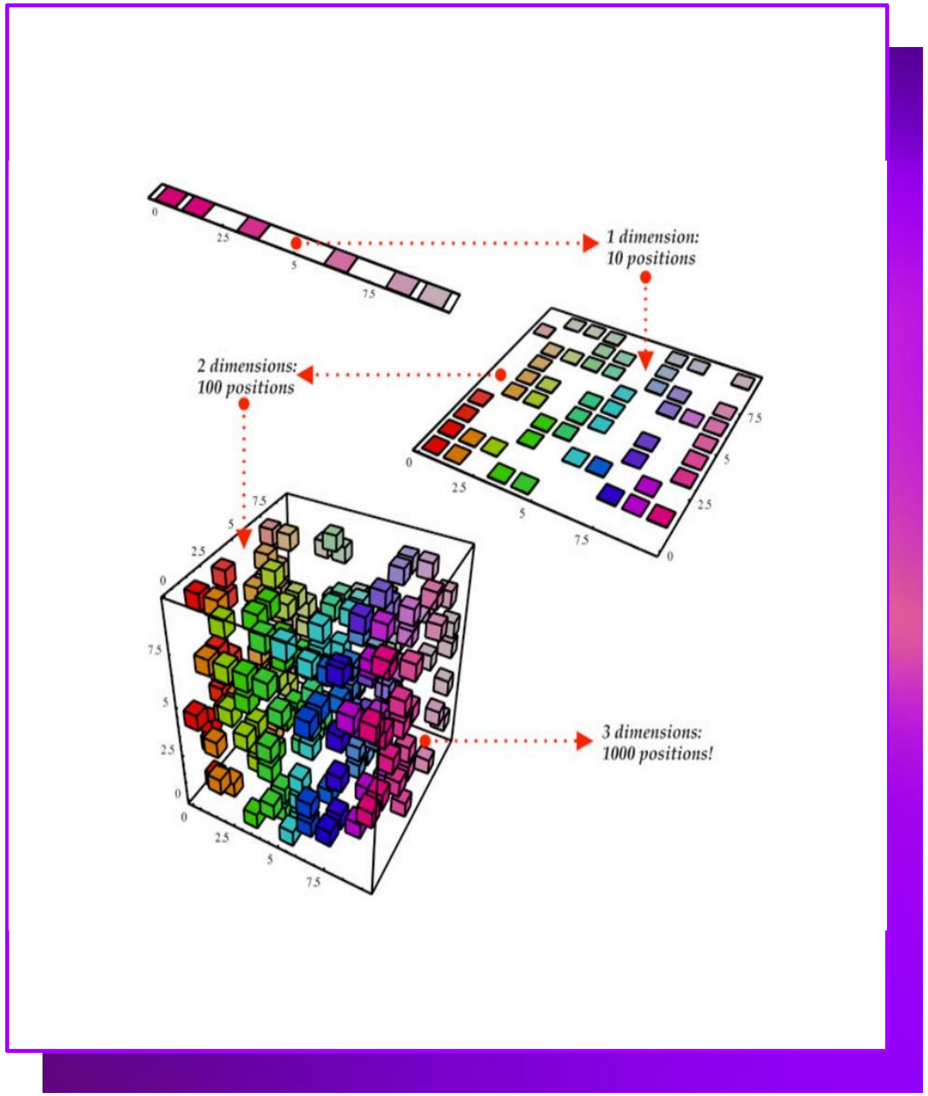
DBSCAN (number of cluster specified not required)

k-means

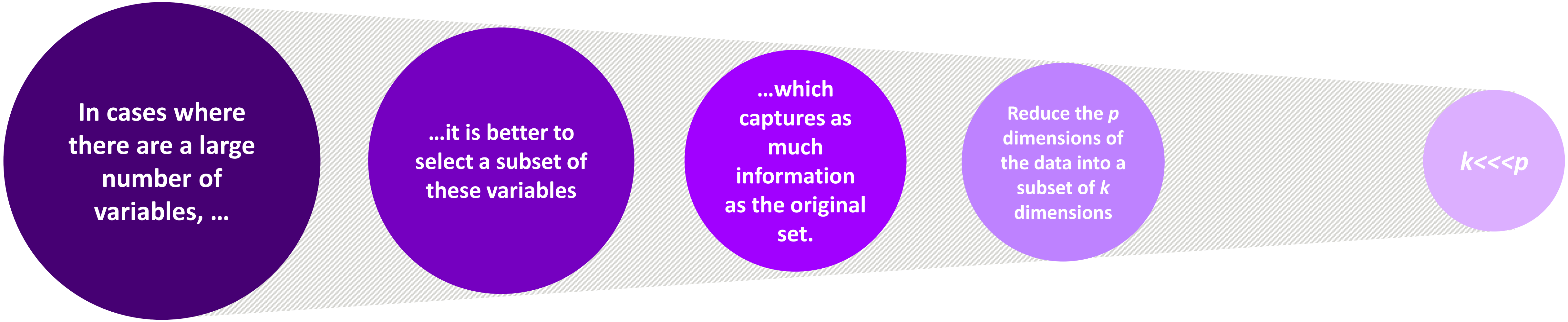
Gaussian Mixture Model

k-modes

Dimensional Reduction



What is dimensional reduction?



In cases where there are a large number of variables, …

…it is better to select a subset of these variables

…which captures as much information as the original set.

Reduce the *p* dimensions of the data into a subset of *k* dimensions

k<<<p

Why is Dimensionality Reduction required?

Space required to store the data is reduced as the number of dimensions comes down

Less dimensions lead to less computation/training time

Some algorithms do not perform well when we have a large dimensions. So reducing these dimensions needs to happen for the algorithm to be useful

It takes care of multicollinearity by removing redundant features.

It helps in visualizing data.

Distinction

By only keeping the most relevant variables from the original dataset (this technique is called **Feature Selection**)

By finding a smaller set of new variables, each being a combination of the input variables, containing basically the same information as the input variables (this technique is called **Dimensionality Reduction**)

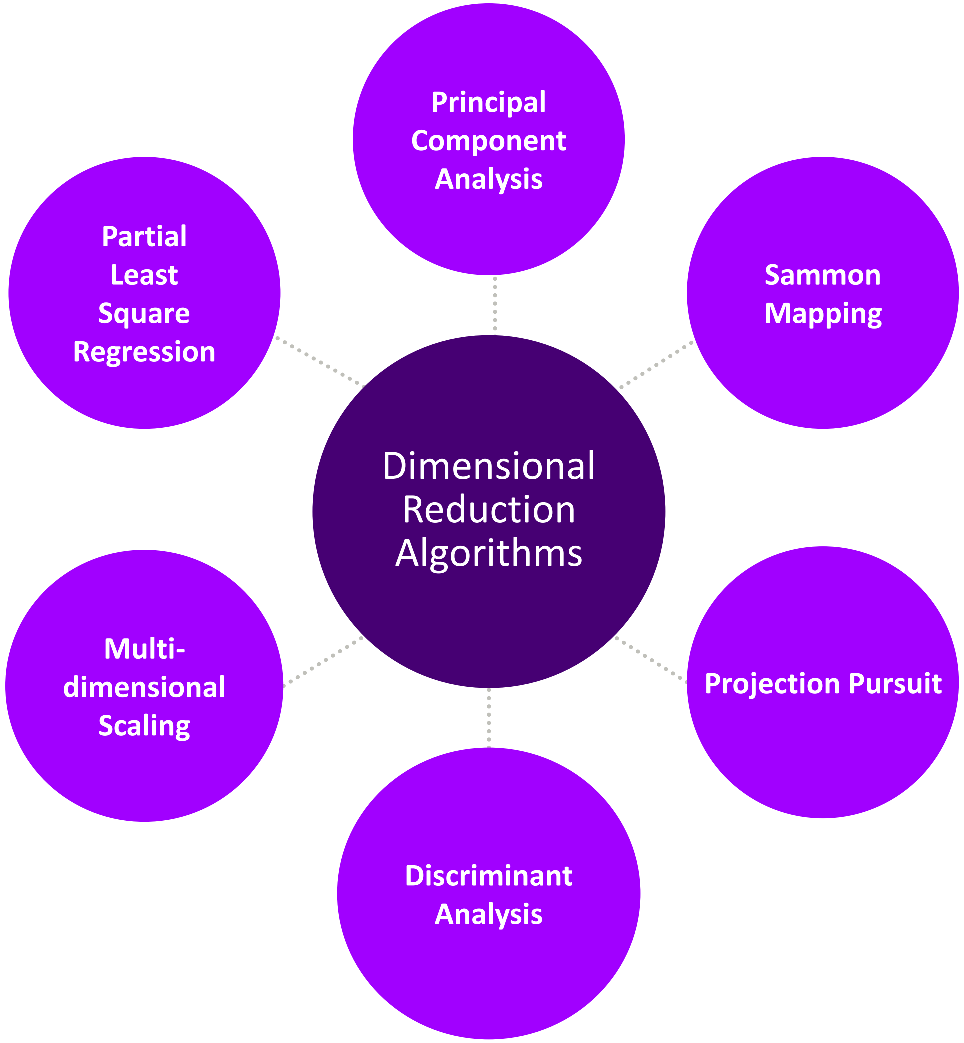
Common Methods of Dimensional Reduction

1	**Missing Values:** Drop the variable if it has more than ~40-50% missing values.	?
2	**Low Variance:** Drop variables having low variance compared to others because these variables will not explain the variation in target variables.	?
3	**Decision Trees:** Use decision trees to select variables	?
4	**Random Forest:** Using the in-built feature importance provided by random forests to select a smaller subset of input features.	✓
5	**High Correlation:** Dimensions exhibiting higher correlation can lower down the performance of model.	✓
6	**Backward Feature Elimination:** Start with all *n* dimensions. Compute the sum of square of error (SSR) after eliminating each variable (*n* times). Then, identifying variables whose removal has produced the smallest increase in the SSR and removing it finally, leaving *n-1* input features. Repeat this process until no other variables can be dropped. Reverse to this, we can use "Forward Feature Selection" method.	✓
7	**Factor Analysis:** EFA (Exploratory Factor Analysis) and CFA (Confirmatory Factor Analysis)	✓
8	**Principal Component Analysis** (PCA)	✓

Usage: ✓ = Common

? = Mostly used in special cases

Notable Algithms


Dimensional Reduction Algorithms
Principal Component Analysis
Sammon Mapping
Projection Pursuit
Discriminant Analysis
Multi-dimensional Scaling
Partial Least Square Regression

Its Coding Time!

Working Session

Case study:

- United States COVID-19 Cases and Deaths by State over Time

Data:

- Source: https://catalog.data.gov/dataset/united-states-covid-19-cases-and-deaths-by-state-over-time
- Descriptions: https://data.cdc.gov/Case-Surveillance/United-States-COVID-19-Cases-and-Deaths-by-State-o/9mfq-cb36/data

Problems:

- Perform some exploratory analysis:
  - Which states has the most deaths?
  - Which month has the most deaths?
  - Are there any correlations?
  - What does the rate of death look like over time?
- Can you perform dimensional reduction on this dataset?
- Can you predict the rate of death in the upcoming month?
- Consider setting threshold for high/medium/low death rate for classification problem
- Example hypotheses:
  - States near the sea are more effected
  - States that are politically conservative have higher death rates

Get started:

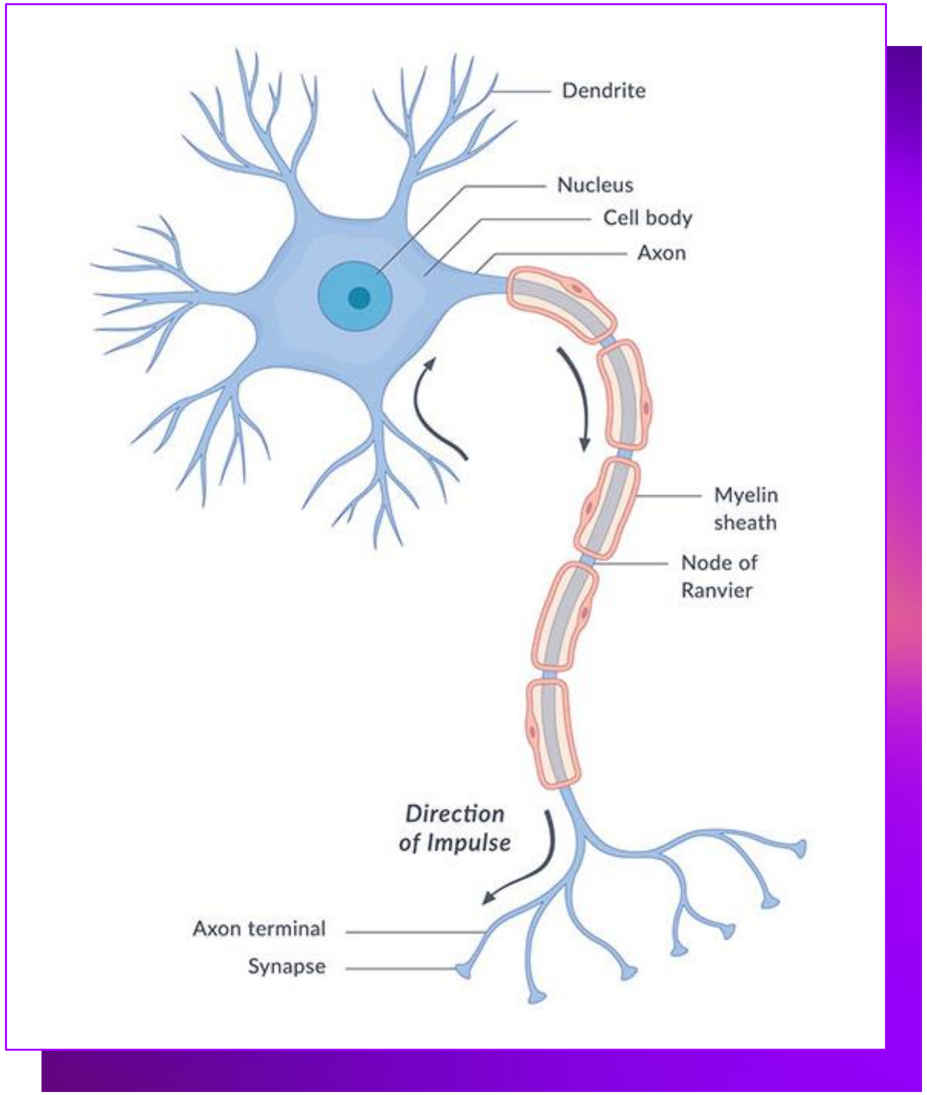
- Python users:
  - Install Anaconda
  - Install Python 3.7
  - Create a virtual environment:
  - *conda create -n yourenvname python=x.x anaconda*
  - Install requirements: *pandas*, *numpy*, and *sklearn* to start

  OR

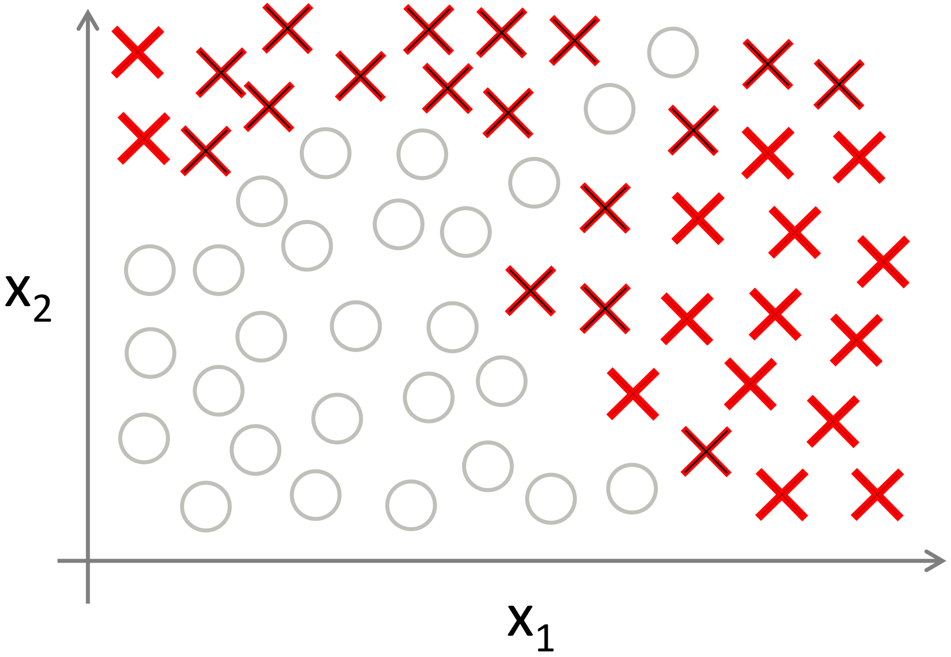
  - Use Google Collab

- R users:
  - Install RStudios
  - Install requirements: *tidyverse* and *gbm* to start

Neural Networks



Non-linear Classification



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 \\ + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \ldots)$$

x_1 = size

x_2 = # bedrooms

x_3 = # floors

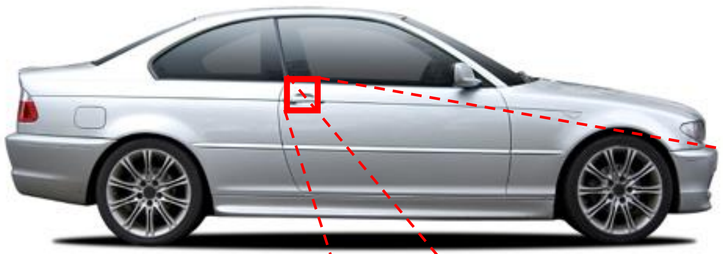
x_4 = age

$\ldots$

x_{100}

What is this?

You see this:

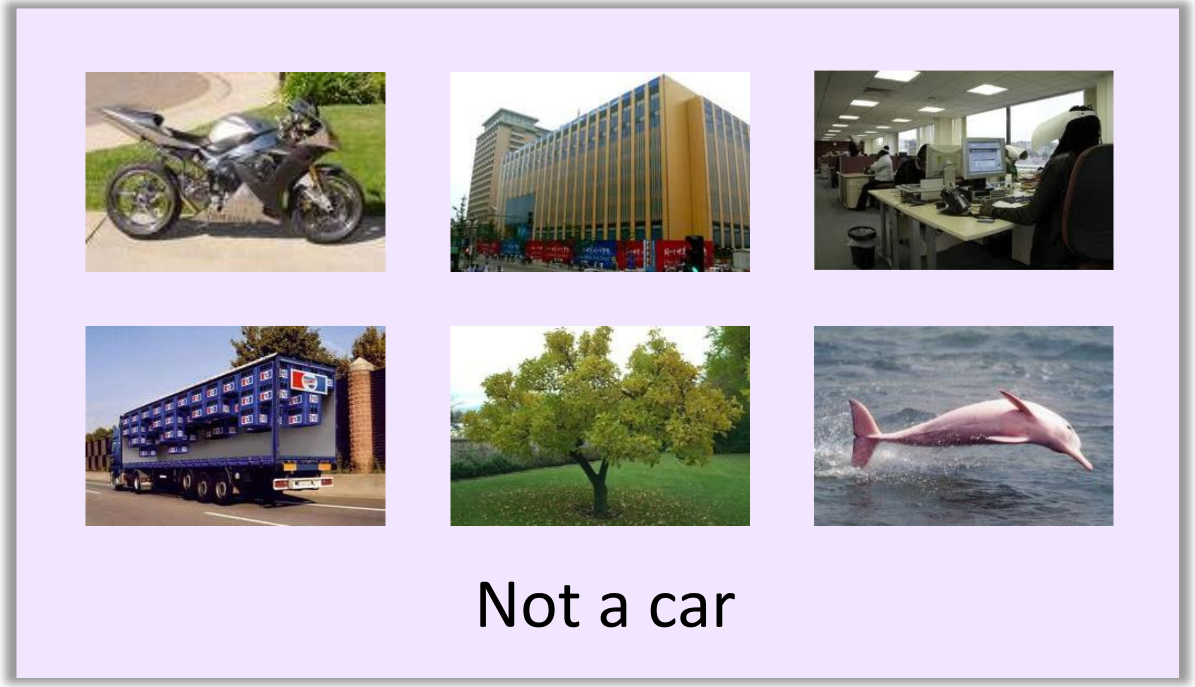


But the camera sees this:

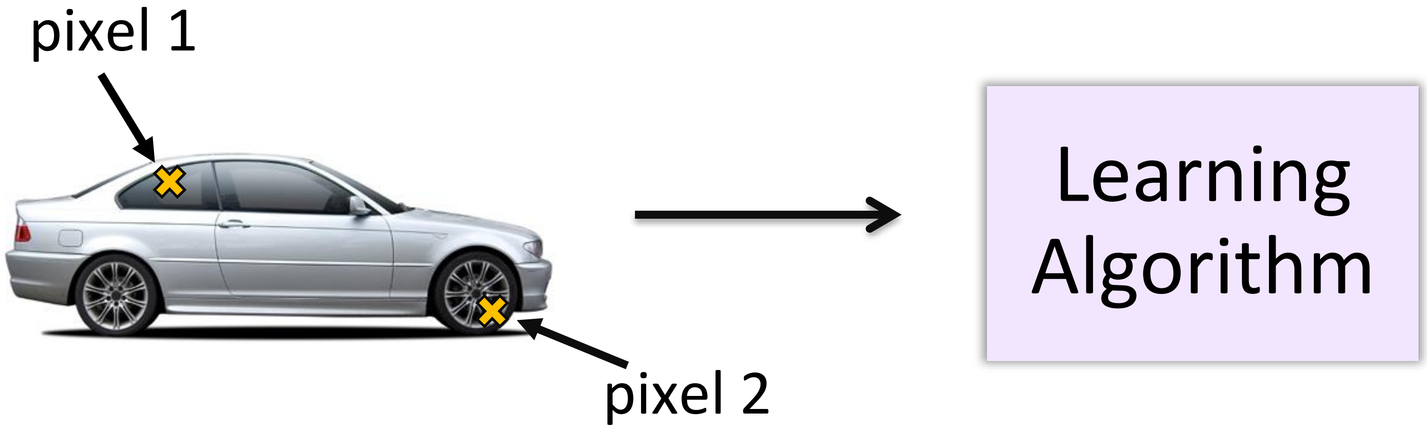
194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Computer Vision: Car detection

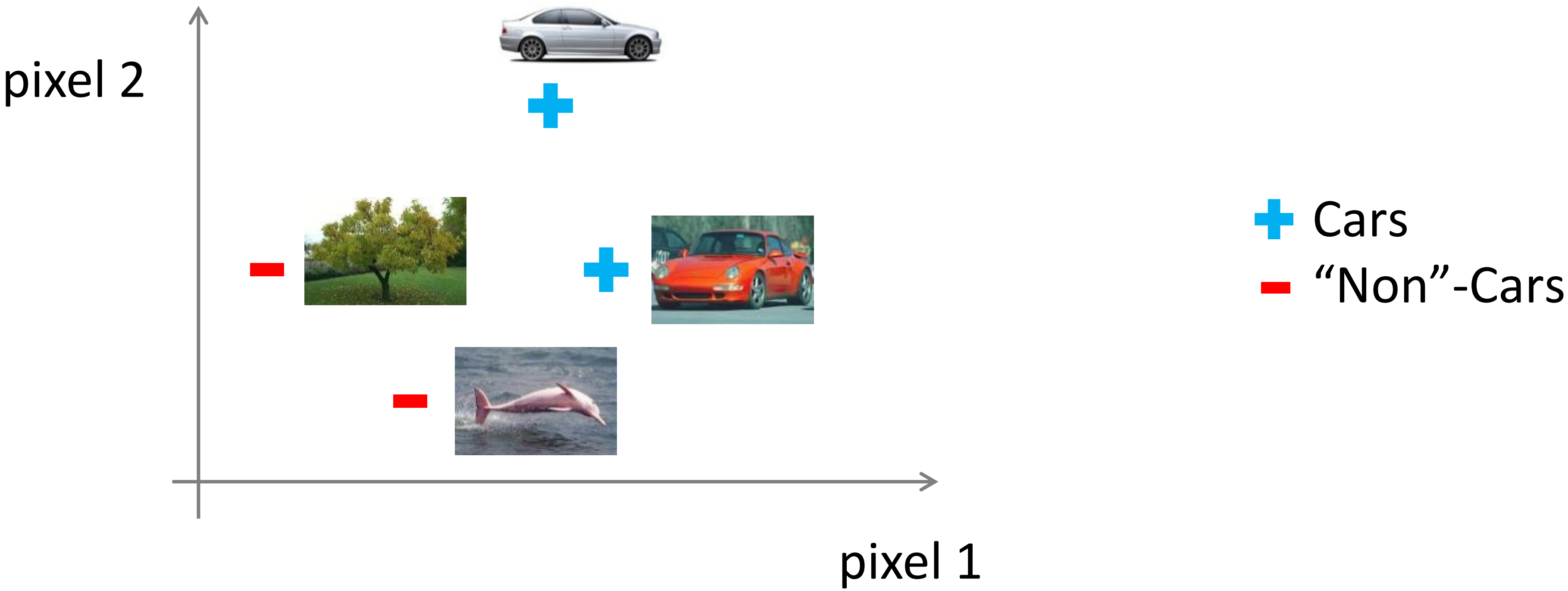
Cars

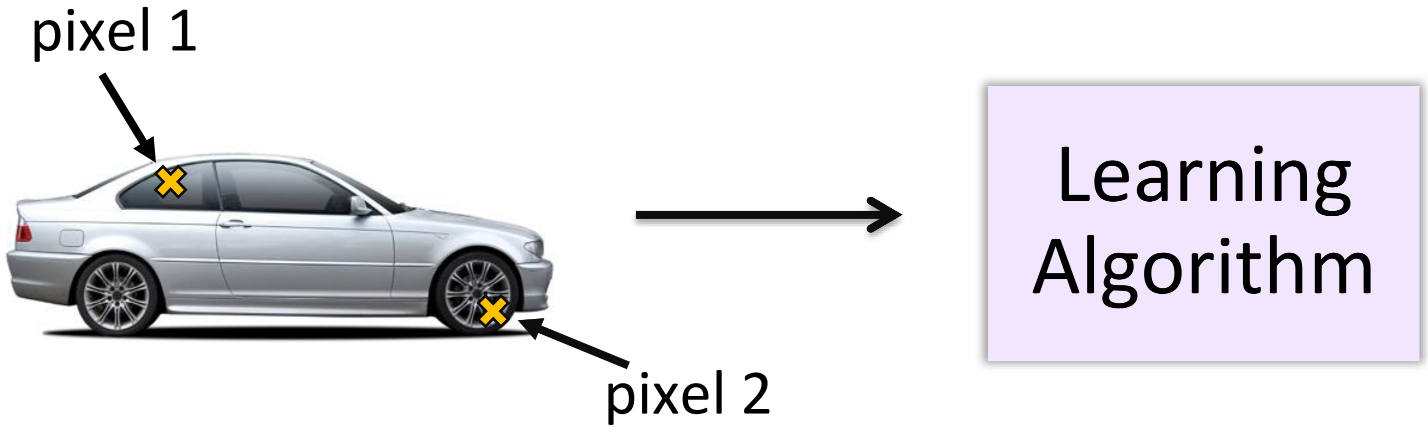

Not a car

Testing: What is this?

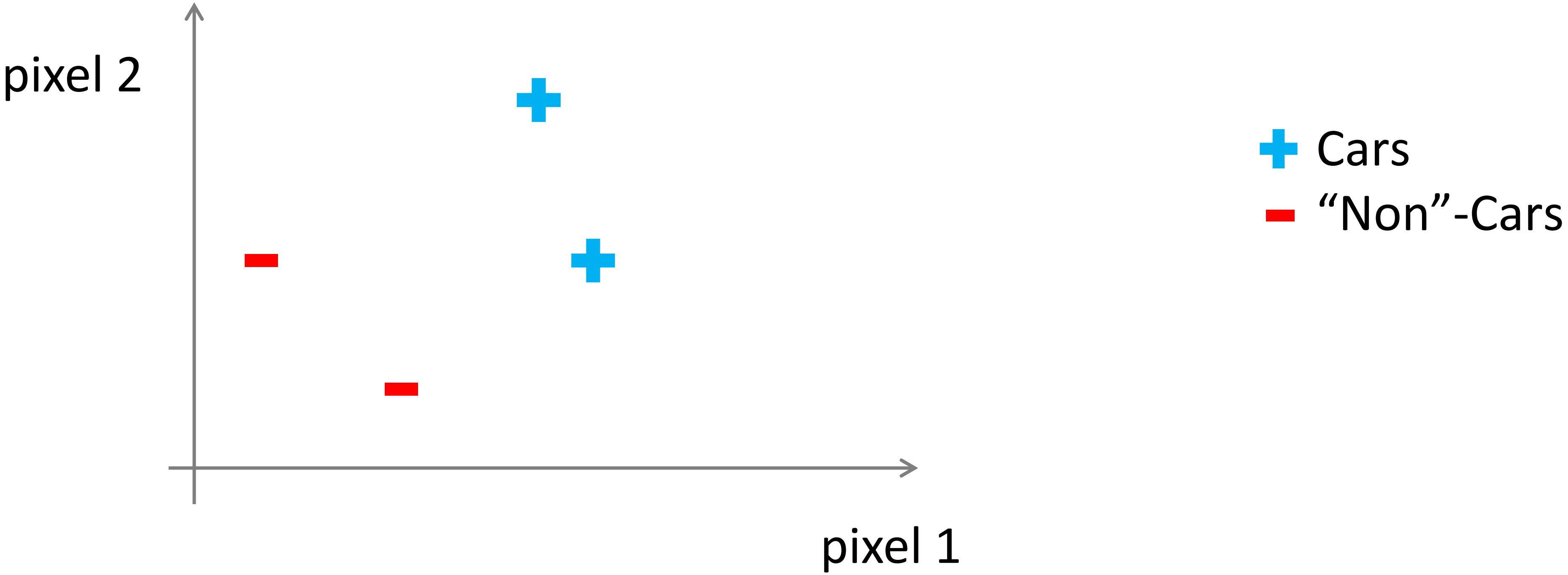


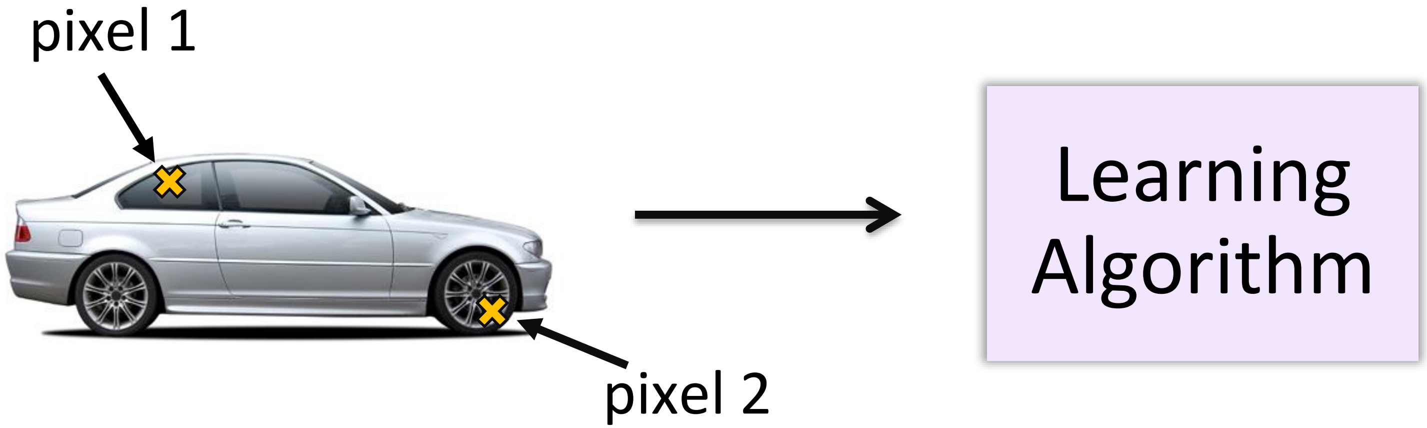
Learning Algorithm

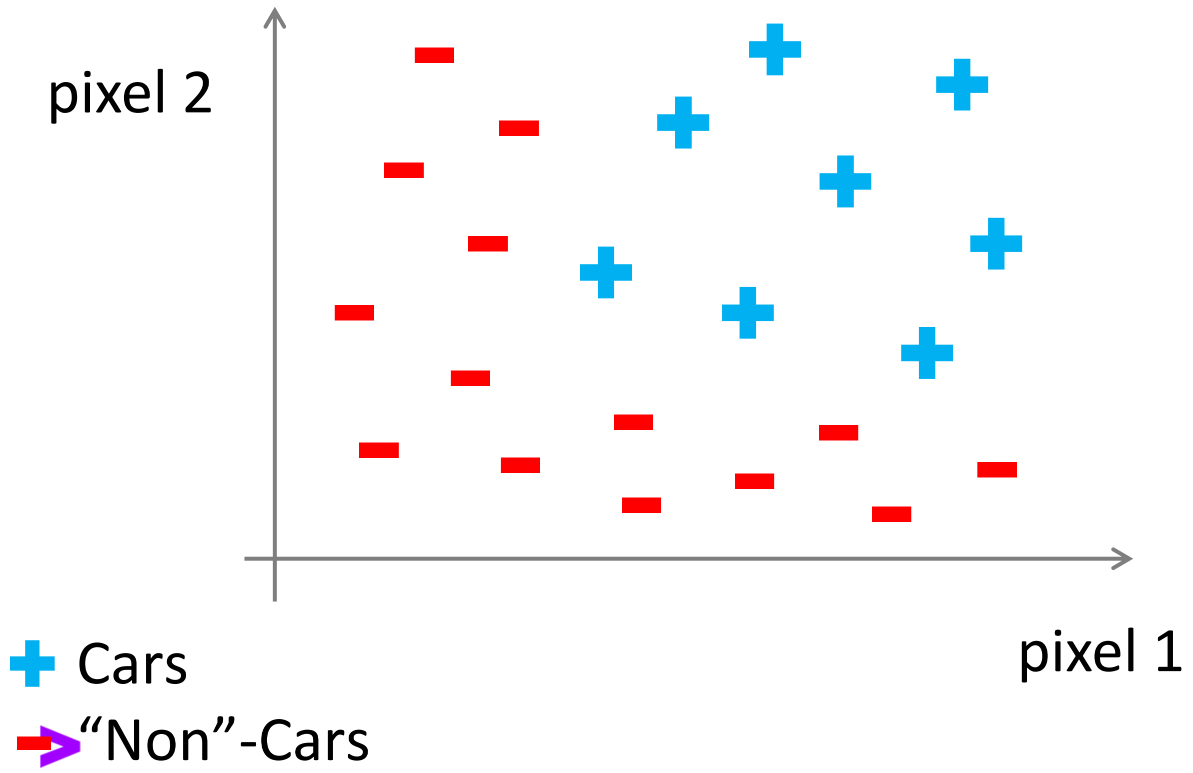




Learning Algorithm







50 x 50 pixel images→ 2500 pixels

$n = 2500$ (7500 if RGB)

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Quadratic features ($x_i \times x_j$): ≈3 million features

Neural Networks

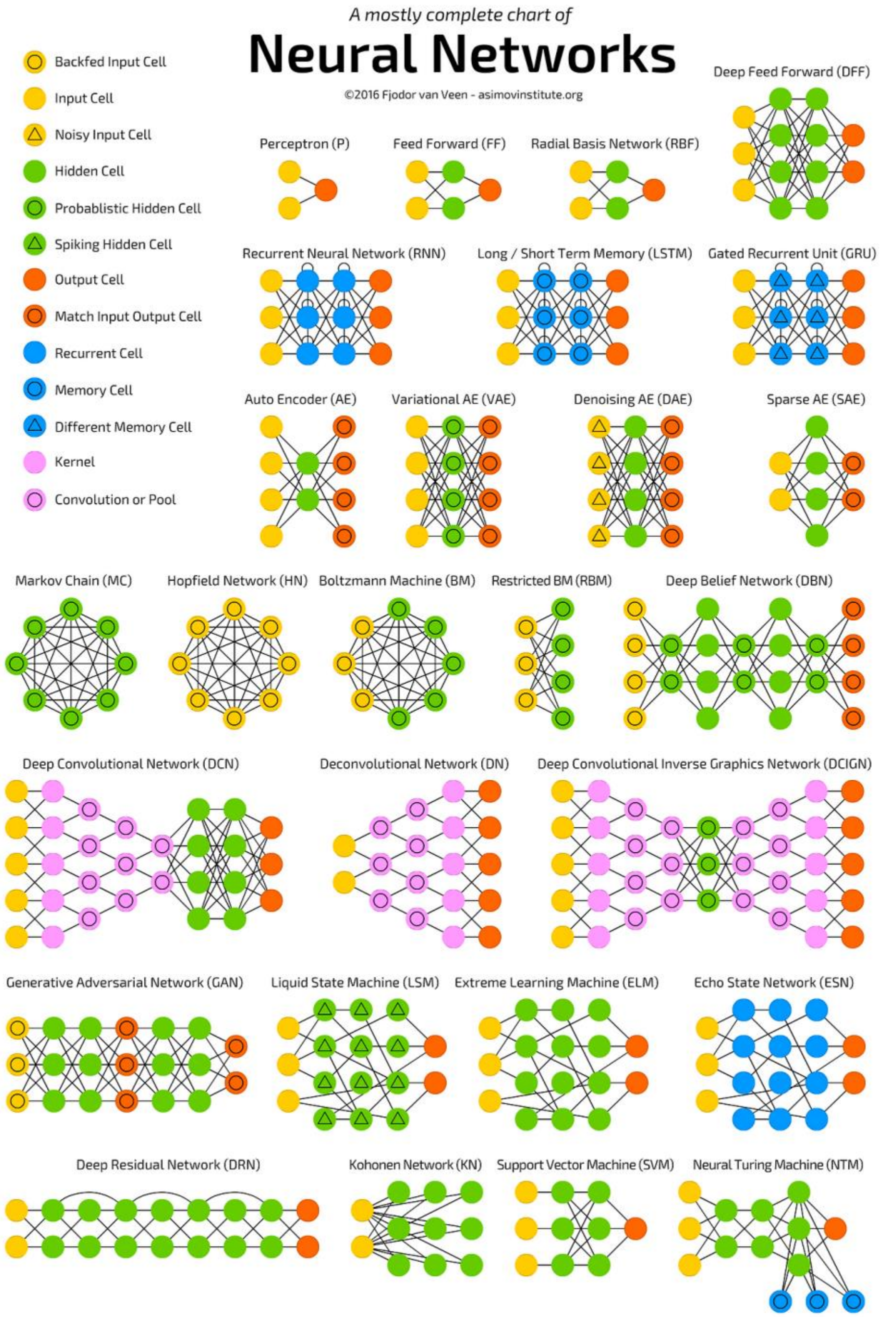
ORIGINS: ALGORITHMS THAT TRY TO MIMIC THE BRAIN.

WAS VERY WIDELY USED IN 80S AND EARLY 90S; POPULARITY DIMINISHED IN LATE 90S.

RECENT RESURGENCE: STATE-OF-THE-ART TECHNIQUE FOR MANY APPLICATIONS

Summary

A collection of neural network visualization


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

Flash Quiz

1. Imagine, you have 1000 input features and 1 target feature in a machine learning problem. You have to select 100 most important features based on the relationship between input features and the target features. Do you think, this is an example of dimensionality reduction?
2. [True of False] It is not necessary to have a target variable for applying dimensionality reduction algorithms.
3. Give some reasons to choose Random Forests over Neural Networks
4. [True of False] Dimensionality reduction algorithms are one of the possible ways to reduce the computation time required to build a model.
5. What happens when you get features in lower dimensions using PCA?

Rule System

What is a rule-based system?

A system designed to achieve artificial intelligence (AI) via a model solely based on predetermined rules is known as a rule-based AI system.

- The makeup of this simple system comprises a set of human-coded rules that result in pre-defined outcomes. These AI system models are defined by ‘if-then’ coding statements.
- Two important elements of rule-based AI models are “a set of rules” and “a set of facts” and by using these, developers can create a basic artificial intelligence model.
- These systems can be viewed as a more advanced form of robotic process automation (RPA).

Rule-based AI models are deterministic by their very nature, meaning they operate on the simple yet effective *‘cause and effect’* methodology. This model is immutably structured and unscalable, therefore it can only perform the tasks and functions it has been programmed for and nothing else. Due to this, rule-based AI models only require very basic data and information in order to operate successfully.

The Key Difference Between Rule-based Artificial Intelligence And Machine Learning Systems

1. **Machine learning systems are probabilistic and rule-based AI models are deterministic.** Machine learning systems constantly evolve, develop and adapt its production in accordance with training information streams. Machine learning models utilize statistical rules rather than a deterministic approach.

2. The other major key difference between machine learning and rule-based systems is **the project scale**. Rule-based artificial intelligence developer models are not scalable. On the other hand, machine learning systems can be easily scaled.

3. **Machine learning systems require more data** as compared to rule-based models. Rule-based AI models can operate with simple basic information and data. However, machine learning systems require full demographic data details.

4. **Rule-based artificial intelligence systems are immutable objects.** On the other hand, machine learning models are mutable objects that enable enterprises to transform the data or value by utilizing mutable coding languages such as java.

When To Use Rule-based System Vs Machine Learning Models

When to utilize machine learning models

- Pure coding processing
- Pace of change
- Simple guidelines don’t apply

When to utilize rule-based models

- No data
- Not planning for machine learning
- Danger of error
- Speedy outputs

Key Takeaway

How do you decide?

CLASSIFICATION

If you have a classification problem “which is predicting the class of a given input.” Keep in mind how many classes you’ll classify your inputs to, as some of the classifiers don’t support multiclass prediction, they only support 2 class prediction.

- Non-linear SVM
- Random Forest
- Neural Network (lots of data needed)
- Gradient Boosting Tree

Slow but accurate

- Explainable models: Decision Tree and Logistic Regression
- Non-explainable Models: Linear SVM and Naive Bayes

Fast

How do you decide?

REGRESSION

If you have a regression problem “which is predicting a continuous value like predicting prices of a house given the features of the house like size, number of rooms, etc.”

- Random Forest
- Neural Network (needs a lot of data points)
- Gradient Boosting Tree (similar to Random Forest, but easier to overfit)

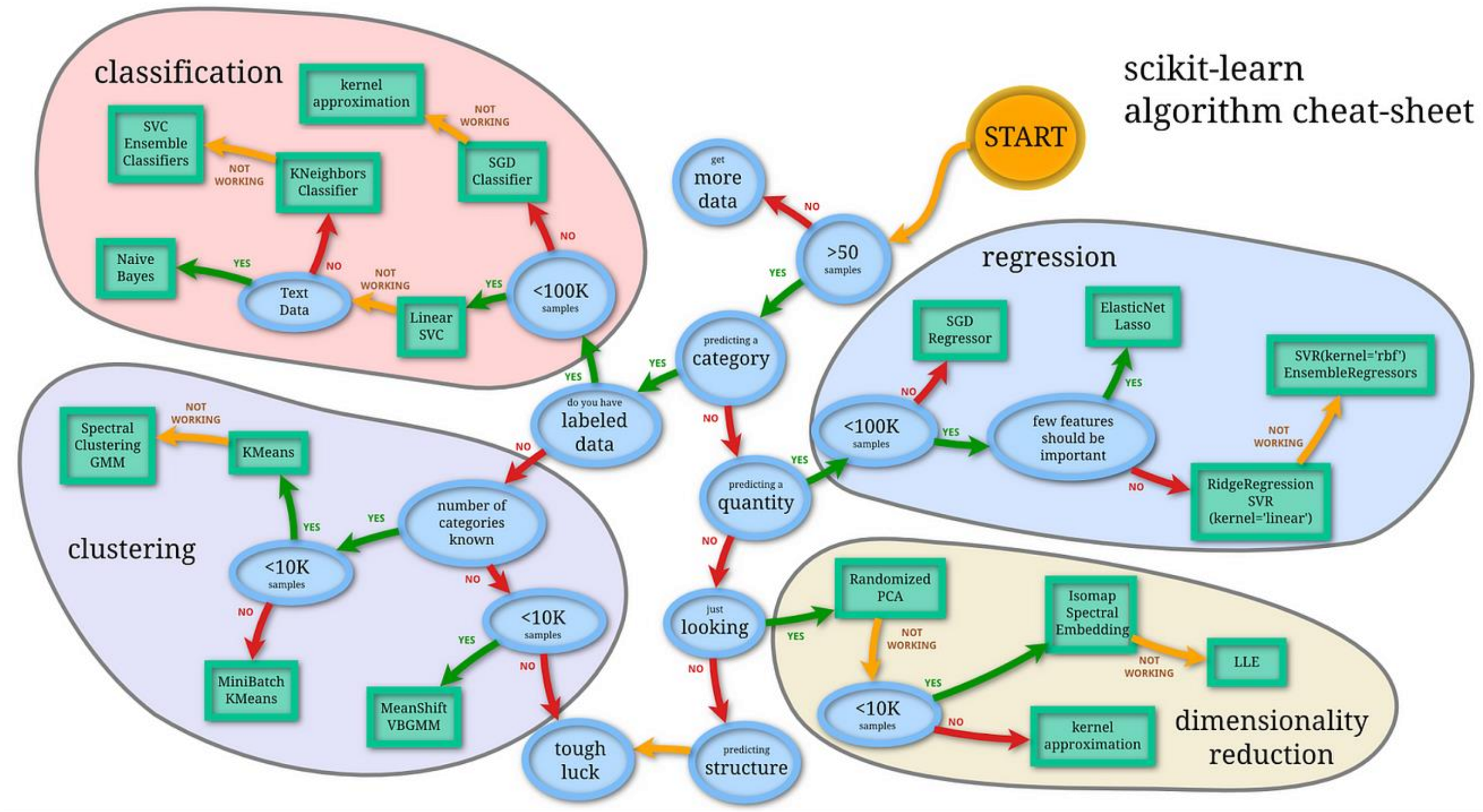
Slow but accurate

- Decision Tree
- Linear Regression

Fast

Example

An example schema to decide on the best machine learning algorithm



Practice Scenarios

Scenario 1:

Key facts (situation): You have the following data from different city in Virginia:

- Zip code of city
- Temperature
- Humidity
- Date and time
- Rain status by date

Key question: Your client would like to predict the chance of rain the upcoming days!

Scenario 2:

Key facts (situation): You have an extensive collection of Facebook ad data which contain the following:

- The ad information
- Audience information
- Reaction (i.e., like, love, wow, sad, haha, angry) count
- Share and comments

Key question: The client would like to know which ad was the best ad for their campaign.

Scenario 3:

Key facts (situation): You have a collection of 350,000 press article tagged with different topics of conversation and sentiment.

Key question: You are asked to build an algorithm that can categorize 50,000 unlabeled articles.

Scenario 4:

Key facts (situation): After your model has been deployed in the previous scenario, you realize you want to be able to quickly fine-tune your model with a small number of data in newer topics that historical data did not cover.

Key question: Your client demand daily delivery but you do not have the time to fine-tune a model.

Referenced Materials

1. Barla, Nilesh. “Dimensionality Reduction for Machine Learning”. MLOps Blog, 19 April 2023, https://neptune.ai/blog/dimensionality-reduction.
2. Gamal , Bassant. “Naïve Bayes Algorithm”. Medium, 17 Dec 2020, https://medium.com/analytics-vidhya/na%C3%AFve-bayes-algorithm-5bf31e9032a2.
3. Gupta, Prashant. “Decision Trees in Machine Learning”. Medium, 17 May 2017, https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052.
4. Johnson, Jonathan. “Interpretability vs Explainability: The Black Box of Machine Learning”. Machine Learning & Big Data Blog, 16 July 2020, https://www.bmc.com/blogs/machine-learning-interpretability-vs-explainability/
5. lejlot (https://stats.stackexchange.com/users/28903/lejlot), Objective function, cost function, loss function: are they the same thing?, URL (version: 2023-04-22): https://stats.stackexchange.com/q/179027.
6. Mwiti, Derrick. “A Comprehensive Guide to Ensemble Learning: What Exactly Do You Need to Know”. MLOps Blog, 21st April 2023, https://neptune.ai/blog/ensemble-learning-guide.
7. Ng, Andrew.” Machine Learning By Prof. Andrew Ng”, (2023), GitHub repository, https://github.com/vkosuri/CourseraMachineLearning.
8. Ray, Sunil. “Commonly used Machine Learning Algorithms (with Python and R Codes)”. Analytics Vidhya, 09 September 2017, https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/.
9. Ray, Sunil. “Beginners Guide To Learn Dimension Reduction Techniques”. Analytics Vidhya, 29 July 2015, https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/.
10. Sharma, Pulkit. “The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes)”. Analytics Vidhya, 27 August 2018, https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/.
11. Shaier, Sagi. “ML Algorithms: One SD (σ)- Bayesian Algorithms”. Medium, 19 Feb 2019, https://towardsdatascience.com/ml-algorithms-one-sd-%CF%83-bayesian-algorithms-b59785da792a#:~:text=A%20family%20of%20algorithms%20where,algorithms%20based%20on%20Bayes'%20Theorem.
12. Singh, Aishwarya. “A Comprehensive Guide to Ensemble Learning (with Python codes)”. Analytics Vidhya, 18 June 2018, https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/.
13. Smith, Robert. “The Key Differences Between Rule-Based AI And Machine Learning”. Medium, 14 Jul 2020, https://becominghuman.ai/the-key-differences-between-rule-based-ai-and-machine-learning-8792e545e6.
14. Singh, Seema. “Understanding the Bias-Variance Tradeoff”. Medium, 21 May 2018, https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229.