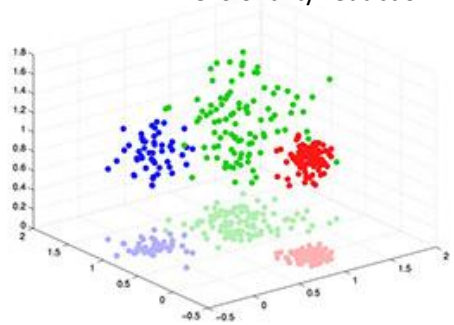

SINTEF
MACHINE LEARNING
FOR CLASSIFICATION

Curriculum

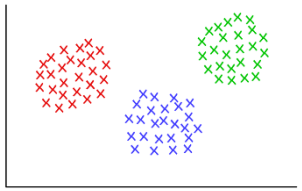
- What is a classification task?
- Linear classifiers and Decision boundaries
  - Logistic regression and its loss function
- Non-linear classifiers
  - Introduction to tree based methods
  - Going from a tree to a random forest
- How do we evaluate the quality of a classifier?

What is a classification task?

Dimensionality reduction



Clustering



Machine Learning

Unsupervised Learning

Dimensionality Reduction

- Meaningful Compression
- Structure Discovery
- Big data Visualistaion
- Feature Elicitation

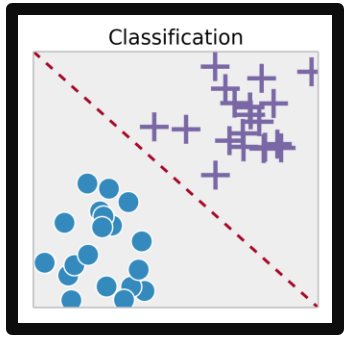
Clustering

- Recommender Systems
- Targetted Marketing
- Customer Segmentation

Supervised Learning

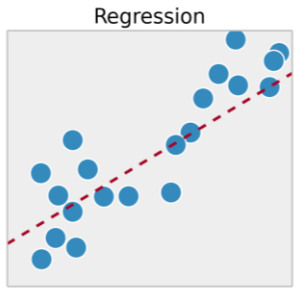
Classification

- Image Classification
- Customer Retention
- Idenity Fraud Detection
- Diagnostics



Regression

- Advertising Popularity Prediction
- Weather Forecasting
- Population Growth Prediction
- Market Forecasting
- Estimating life expectancy



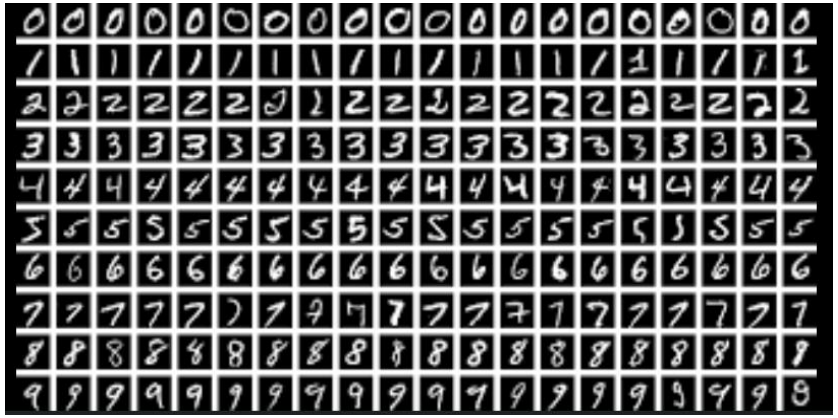
Reinforcement Learning

- Real-time decisions
- Game AI
- Robot Navigation
- Skill Acquisition
- Learning Tasks

towardsdatascience.com

SINTEF

What is a classification task?

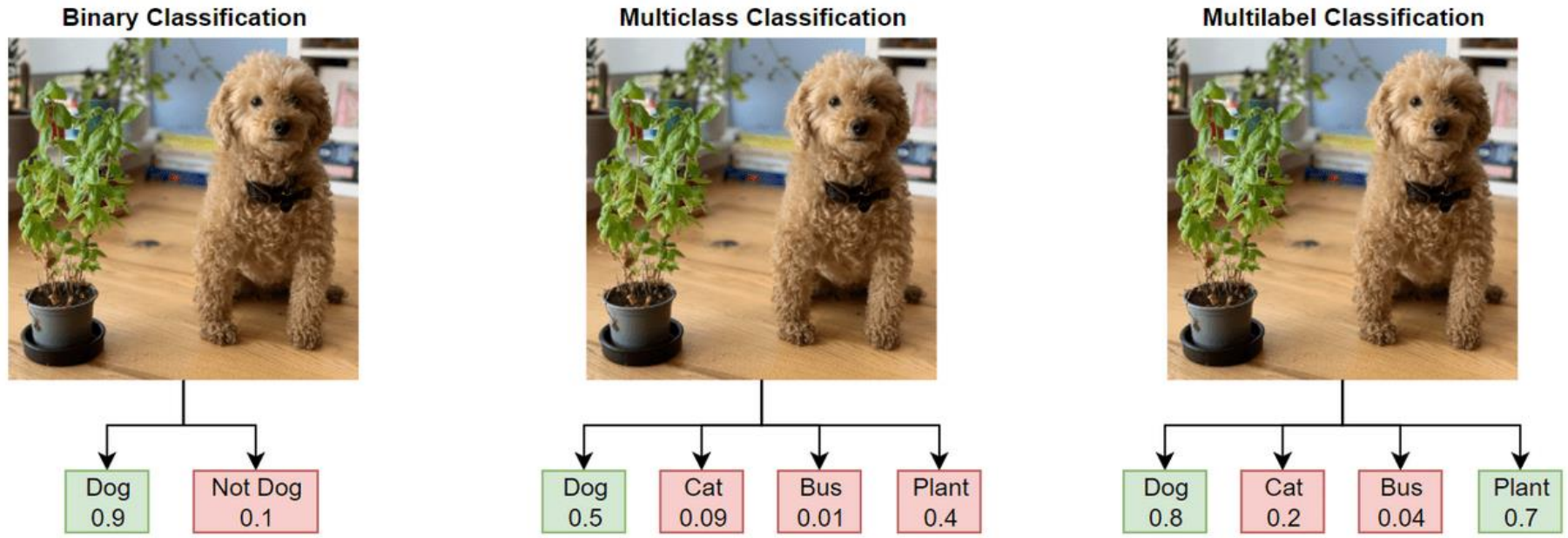


- Tasks: Find a function (aka classifier) that assigns a **class** to a data point
  - Input – A set of independent variable X – Also known as features
  - Output – A class – Also known as target/dependent variable
- Examples:

Data point	Class (label)
Image: hand drawn digits	Digit: 0, 1, 2, ...
Image	Chihuahua, Blueberry Muffin
Patient data	Cancer / No cancer
...	...

SINTEF

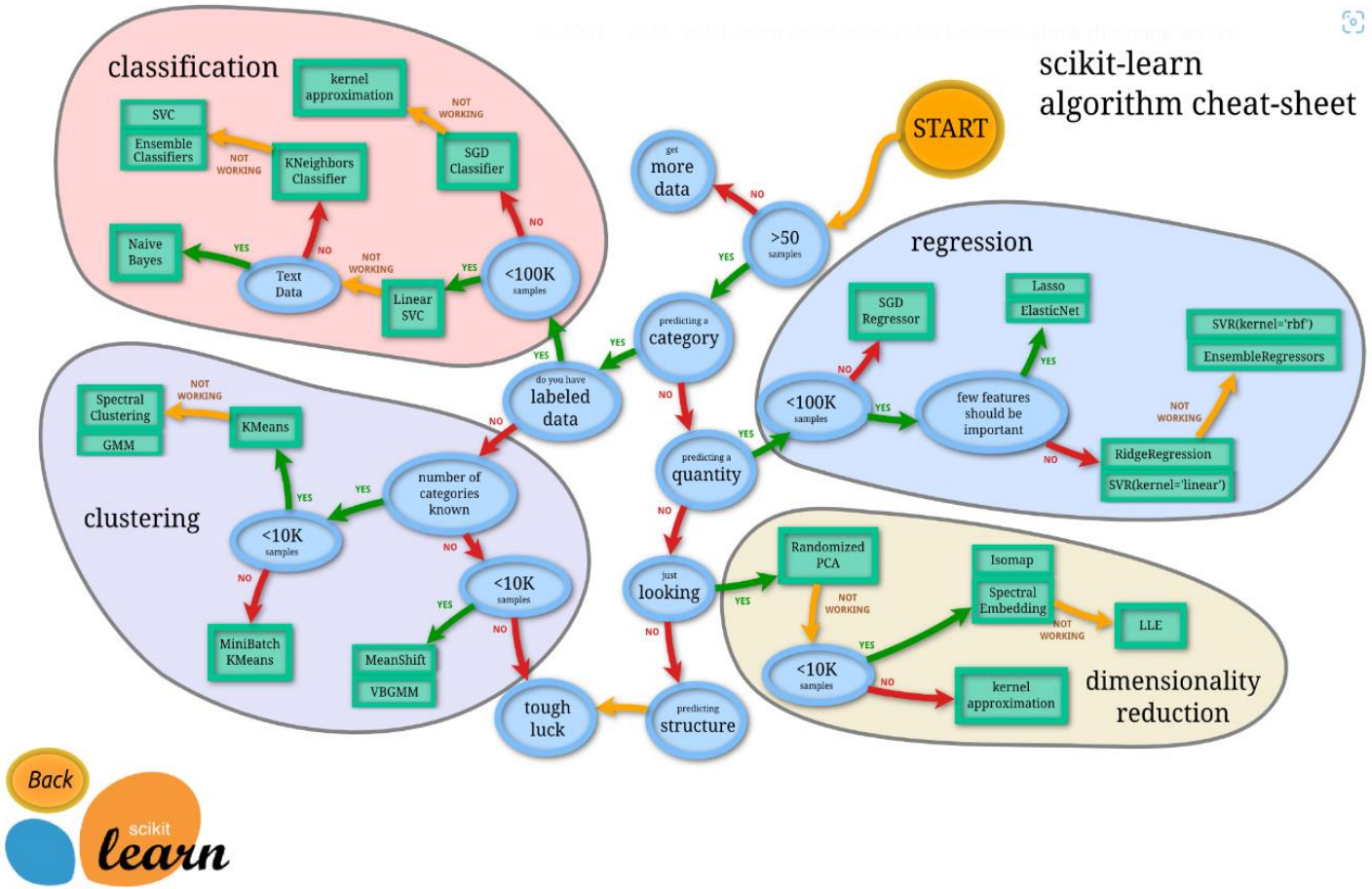
What is a classification task?
- Binary, multiclass and multilabel



https://www.mathworks.com/help/deeplearning/ug/multilabel-image-classification-using-deep-learning.html

(The art of) Choosing the right approaches

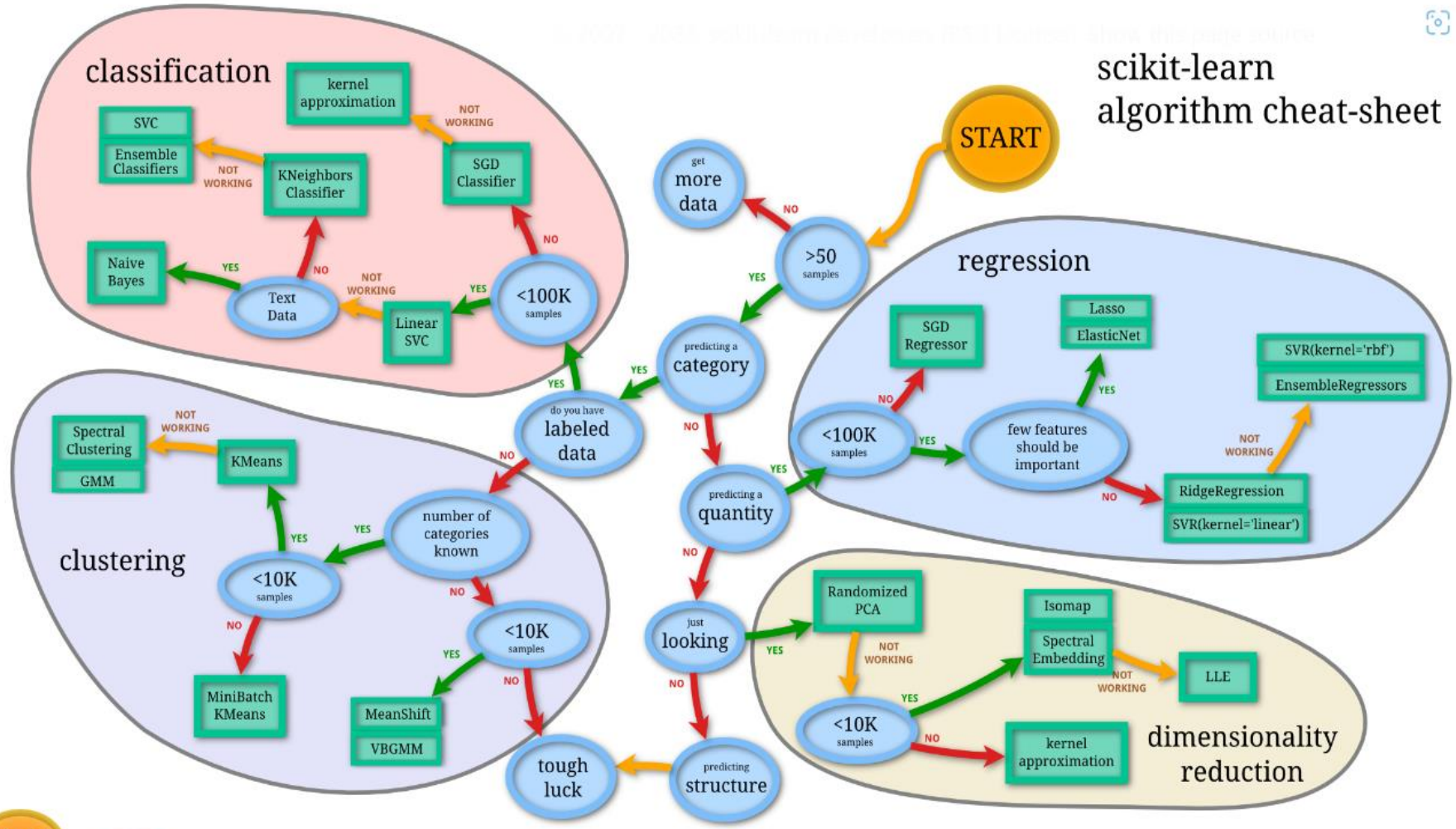
- In general difficult to know what will work best
- Useful guidelines: https://scikit-learn.org/stable/machine_learning_map.html



SINTEF

Quiz -

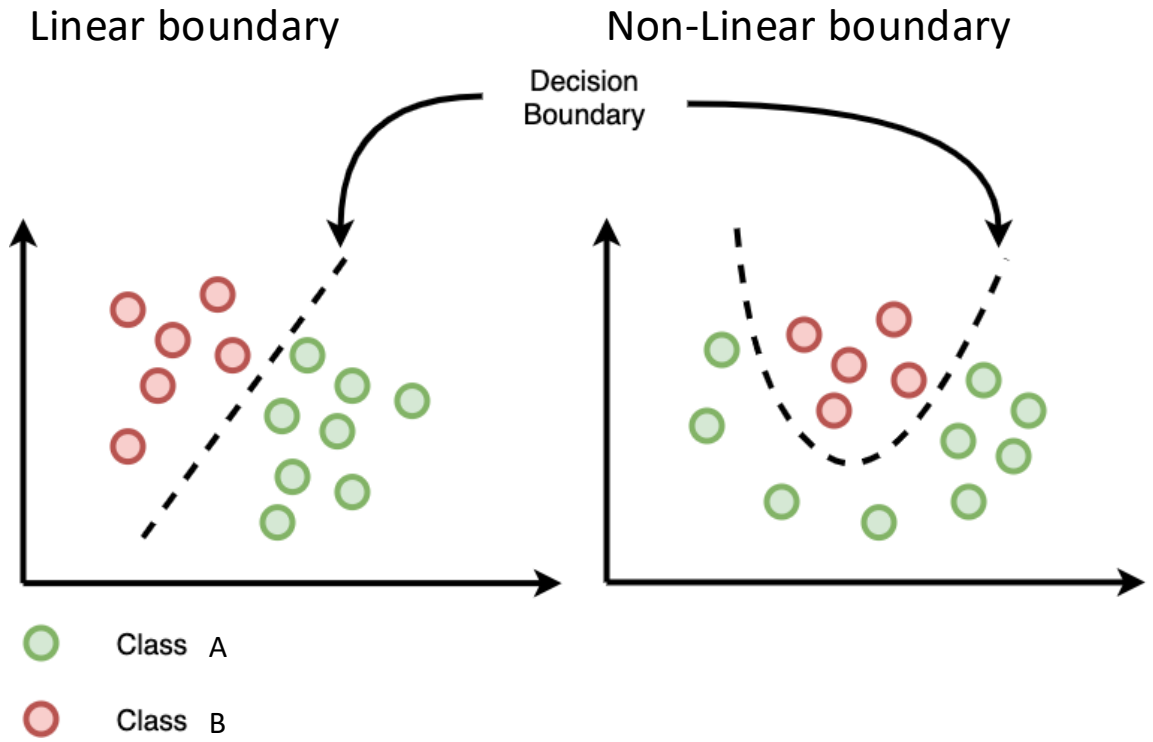
- Use-cases:
  - A - Predicting whether an email is spam from using a labeled dataset of 50k emails.
  - B - Predicting whether a given color is considered *blue* given a labeled dataset of a million opinions linking a given color to whether it can be considered blue :)



Linear classifier and decision boundaries

Linear classifier and decision boundaries

- Decision boundary
  - separates input space into regions,
  - each region corresponds to a class
- A linear decision boundary can be modeled with a **linear function**.
  - 2D: line
  - 3D: plane
  - nD: hyperplane

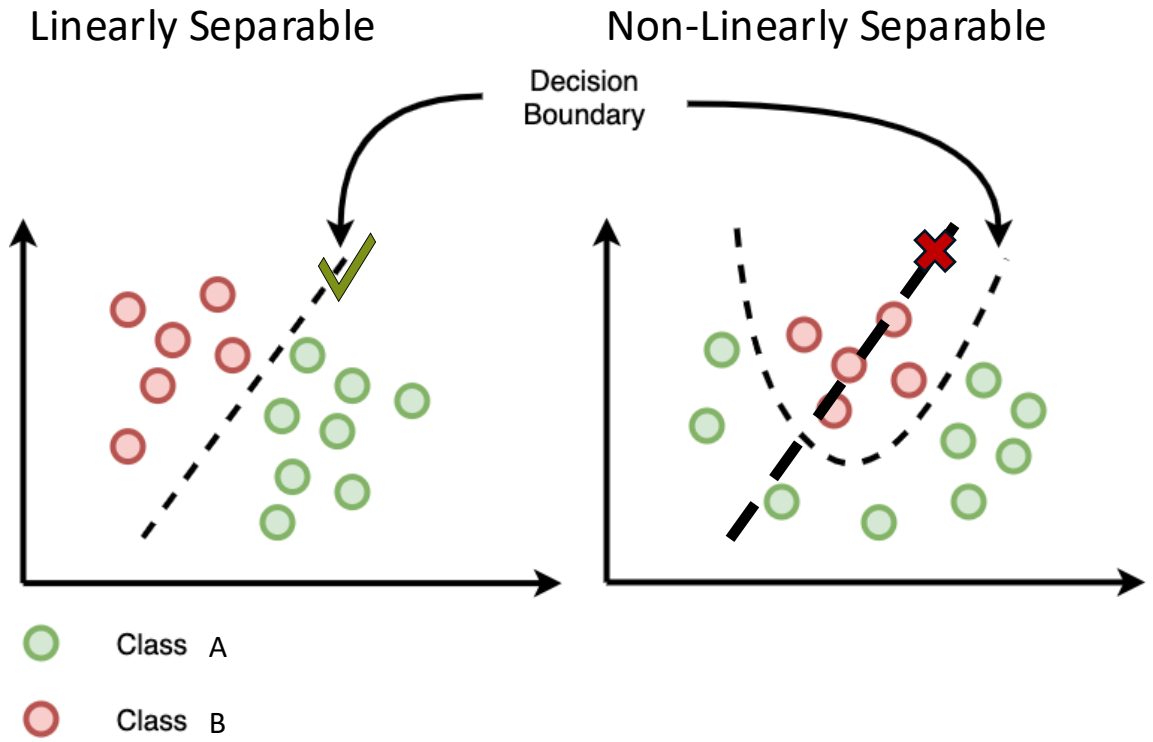


https://medium.com/data-science/logistic-regression-and-decision-boundary-eab6e00c1e8

SINTEF

When Do Linear Classifiers Work Well?

- Linearly Separable Data
  - a straight line (or hyperplane) can divide the classes without error.
  - All examples from one class lie on one side of the line; the other class lies on the other side.
  - No overlap between classes in the feature space.
  - Common in simple problems or clean datasets with well-behaved features



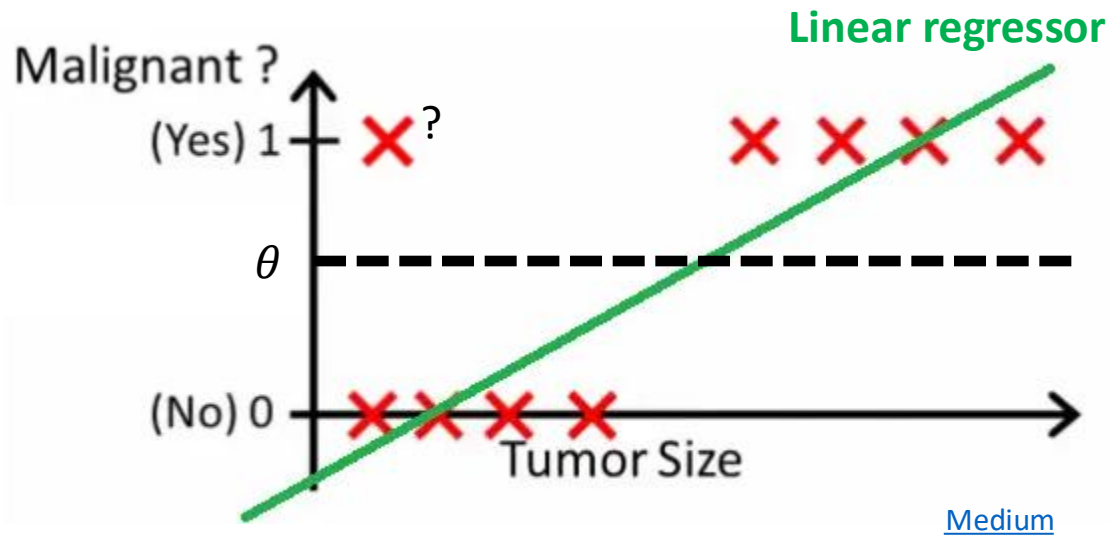
https://medium.com/data-science/logistic-regression-and-decision-boundary-eab6e00c1e8

Logistic regression and its loss function

- A simple binary classifier

Naïve (binary) Classifier: Using Linear Regression

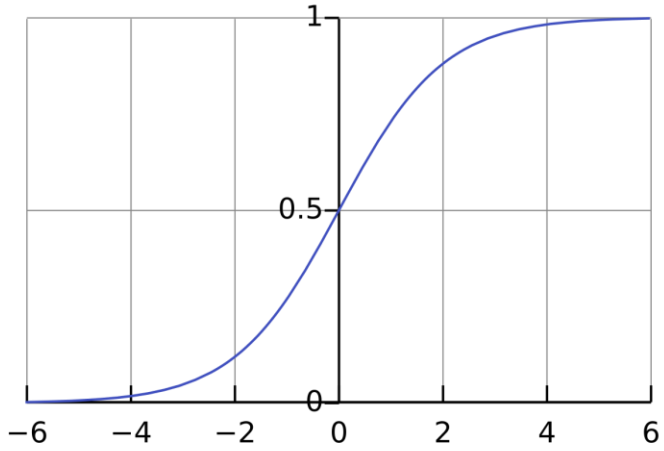
- Transform labels to numerical values y (e.g., yes = 1, no = 0)
- Find a linear function $mX + b$ that fits well for data (x, y)
  → output is a numerical value
- Set a *threshold*: θ
  - If predicted value $\hat{y} > \theta$: predict class A, e.g. yes
  - If predicted value $\hat{y} < \theta$: predict class B, e.g. no
- Problem:
  - $\hat{y}$ has no real interpretation: E.g., $\hat{y} = 5$, $\hat{y} = -1$?
  - Sensitive to outliers



SINTEF

Logistic Regression

- Idea: map output to *probabilities*
  - Probability of being in class A, e.g. “yes”
  - Set a threshold on the *probabilities*
    → $\theta = 0.5$ means: if probability of data point x belonging to class A is higher than 0.5, then predict class A (class B otherwise)
  - Can quantify *uncertainty* of predictions
- Sigmoid function: $\sigma(z) = \frac{1}{1 + e^{-z}}$
  - Maps any real values to [0,1]
  - Interpretable as probabilities

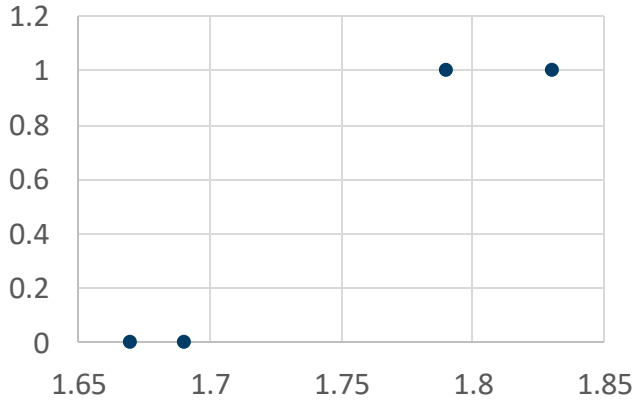


SINTEF

Logistic Regression: Example

- Data:

x (height)	y (tall / not tall)
1.83 m	1 = Tall
1.79 m	1 = Tall
1.67 m	0 = Not Tall
1.69 m	0 = Not Tall



1. Use linear function: $z = mx + b$
2. Apply sigmoid function: $\sigma(z) = \frac{1}{1 + e^{-z}}$
3. Use threshold θ: **Tall** if $\sigma(z) > \theta$ and **Not Tall** if $\sigma(z) \leq \theta$

Towards Data Science

SINTEF

Logistic Regression: Example

- Data:

x (height)	y (tall / not tall)	$z = 1x + 0$	$\sigma(z) = \frac{1}{1 + e^{-z}}$	Result
1.83 m	1 = Tall	1.83	0.86176173	Tall
1.79 m	1 = Tall	1.79	0.85692728	Tall
1.67 m	0 = Not Tall	1.67	0.84157582	Tall
1.69 m	0 = Not Tall	1.69	0.844224116	Tall

1. Use linear function: $z = mx + b$ For $m = 1,\ b = 0$
2. Apply sigmoid function: $\sigma(z) = \frac{1}{1 + e^{-z}}$ $\theta = 0.5$
3. Use threshold θ: **Tall** if $\sigma(z) > \theta$ and **Not Tall** if $\sigma(z) \leq \theta$

Towards Data Science

Logistic Regression: Example

- Data:

x (height)	y (tall / not tall)	$z = 6x - 10.5$	$\sigma(z) = \frac{1}{1 + e^{-z}}$	Result
1.83 m	1 = Tall	0.48	0.61774787	Tall
1.79 m	1 = Tall	0.24	0.55971365	Tall
1.67 m	0 = Not Tall	-0.48	0.38225213	Not Tall
1.69 m	0 = Not Tall	-0.36	0.41095957	Not Tall

1. Use linear function: $z = mx + b$ For $m = 6,\ b = -10.5$
2. Apply sigmoid function: $\sigma(z) = \frac{1}{1 + e^{-z}}$ $\theta = 0.5$
3. Use threshold θ: **Tall** if $\sigma(z) > \theta$ and **Not Tall** if $\sigma(z) \leq \theta$

Towards Data Science

Loss Function: Cross-Entropy

- Finding the best *parameters* (i.e., m and b in the linear function $mx + b$) …

  Minimise Cross-Entropy Loss: $L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$

  - y is the real label for data point x (1 = class A, 0 = class B)
  - $\hat{y}$ is the predicted label for data point x: we have $z = mx + b$ and $\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$

- Why?

✅ **Your Model is Confident and Correct:**
- True label: 1
- Prediction: 0.95 → Almost perfect!
- Loss: Very small (close to 0)

❌ **Your Model is Confident but Wrong:**
- True label: 1
- Prediction: 0.05 → Very wrong!
- Loss: Very large (because you're confidently wrong)

✊ **Your Model is Uncertain:**
- True label: 1
- Prediction: 0.51 → barely right
- Loss: Small, but not as small as being very confident

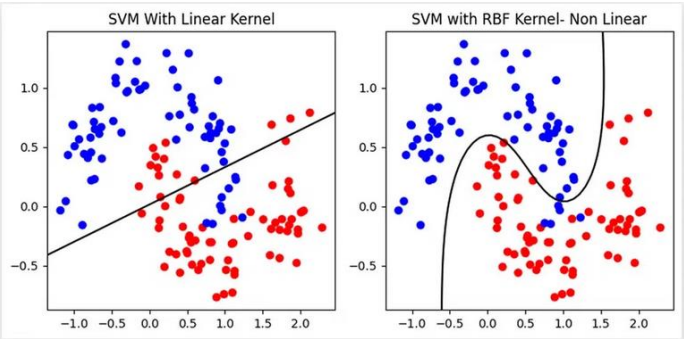
- Decision Boundary:

  Logistic regression's boundary corresponds to where the model is equally confident about each class.

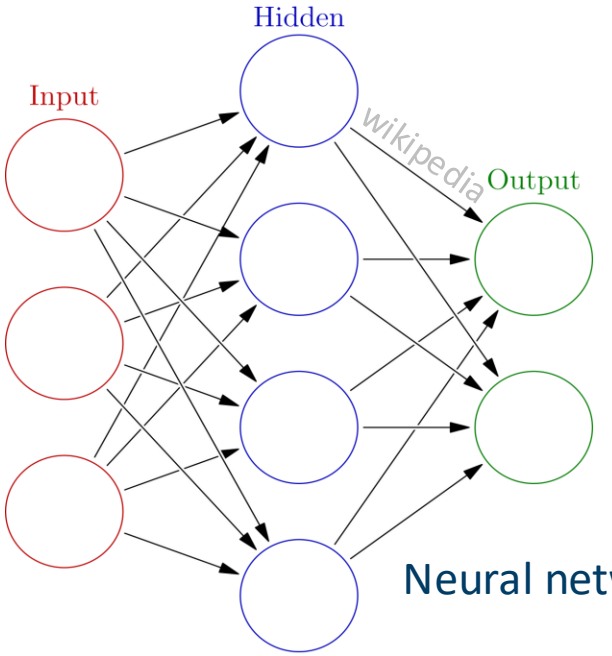
Non-Linear Classifiers

- Overview
- Building a decision tree
- Random Forests as a collection of decision trees

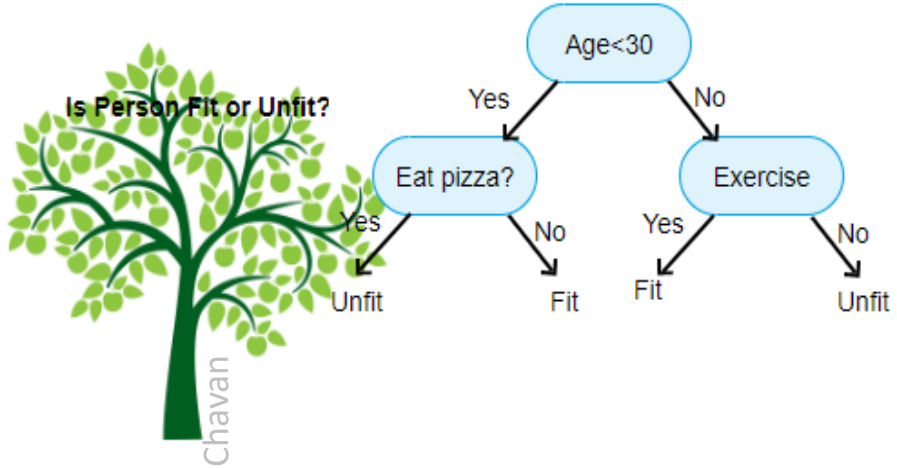
Non-linear classification



Support Vector Machines



Neural networks

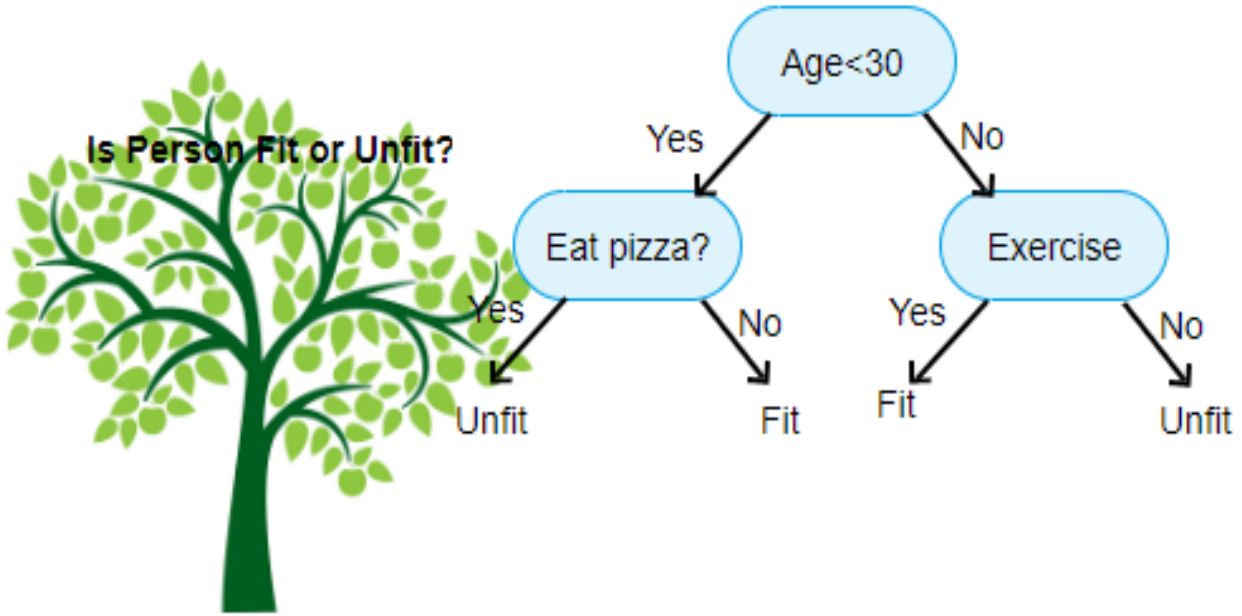


Decision trees

SINTEF

Decision trees

- *Why Use Trees?* Intuitive, non-linear, handle mixed data types
- *How Does it Work?* Split data on features



Chavan

SINTEF

Building a Decision Tree

1. Start with the full dataset.
   For each feature, test multiple split points (thresholds).
   Measure how “good” each split is.
2. Measure how good (purity) each resulting subgroups are

   Purity: mostly or all samples in the group belong to the same class
3. Choose the best split point and split dataset accordingly
4. Repeat the process for each subgroup of data
5. Stop when a) nodes are *pure*, or b) too many splits, or c) not enough data to split further.

Decision Tree: Example

- Data: x = (height, age, voice pitch)
  y = gender
- Try different splits:
  - Pitch = Low → Male ✅
    Pitch = High → Female (1 ❌)
  - Height ≥ 165 → Male ✅
    Height < 165 → Female (1 ❌)
  - Age ≥ 56 → Male ✅
    Age < 56 → Female (2 ❌)
  - Age ≥ 42 → Male (1 ❌)
    Age < 42 → Female (1 ❌)
  - …

Height (cm)	Age	Voice Pitch	Gender
180	56	Low	Male
170	42	Low	Male
160	45	High	Female
150	10	High	Female
145	12	High	Male

Decision Tree: Example

- Data: x = (height, age, voice pitch)
  y = gender
- Try different splits:
  - Pitch = Low → Male ✅
    Pitch = High → Female (1 ❌)
  - Height ≥ 165 → Male ✅
    Height < 165 → Female (1 ❌)
  - Age ≥ 56 → Male ✅
    Age < 56 → Female (2 ❌)
  - Age ≥ 42 → Male (1 ❌)
    Age < 42 → Female (1 ❌)
  - …

Height (cm)	Age	Voice Pitch	Gender
180	56	Low	Male
170	42	Low	Male
160	45	High	Female
150	10	High	Female
145	12	High	Male

Decision Tree: Example

- Data: x = (height, age, voice pitch)
  y = gender

Pitch = Low → Male ✅
Pitch = High → Female (1 ❌)

- Try different splits:
  - Height ≥ 150 → Female ✅
    Height < 150 → Male ✅
  - Age ≥ 45 → Female ✅
    Age < 45 → Male (1 ❌)
  - Age ≥ 12 → Male (1 ❌)
    Age < 12 → Female ✅
  - ...

Height (cm)	Age	Voice Pitch	Gender
180	56	Low	Male
170	42	Low	Male
160	45	High	Female
150	10	High	Female
145	12	High	Male

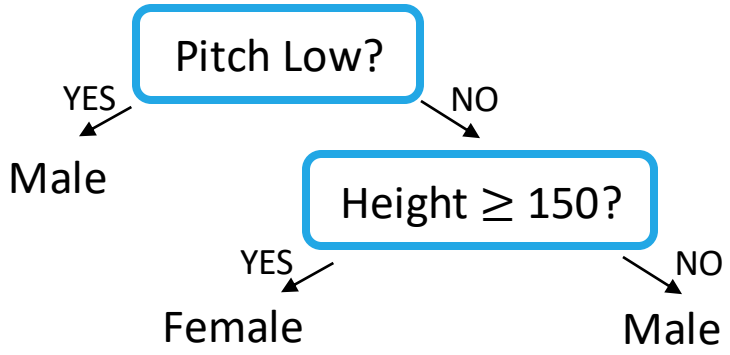
Decision Tree: Example

- Data: x = (height, age, voice pitch)
  y = gender

Pitch = Low → Male ✅
Pitch = High → Female (1 ❌)

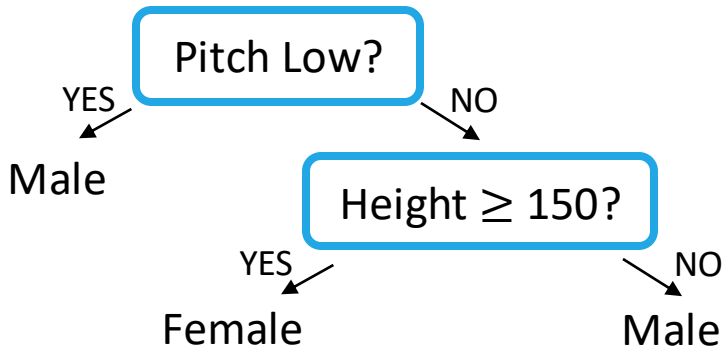
- Try different splits:
  - Height ≥ 150 → Female ✅
    Height < 150 → Male ✅
  - Age ≥ 45 → Female ✅
    Age < 45 → Male (1 ❌)
  - Age ≥ 12 → Male (1 ❌)
    Age < 12 → Female ✅
  - ...

Height (cm)	Age	Voice Pitch	Gender
180	56	Low	Male
170	42	Low	Male
160	45	High	Female
150	10	High	Female
145	12	High	Male



SINTEF

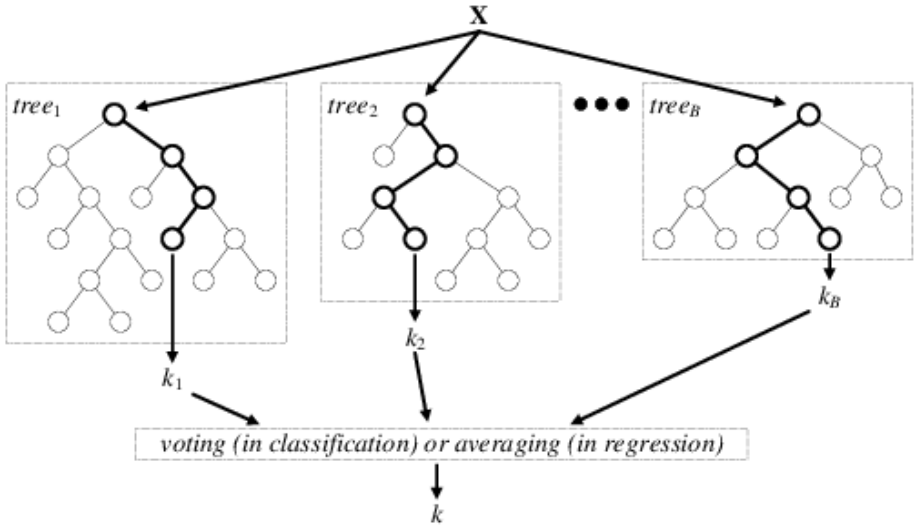
Quiz – Using the tree to predict gender



	Height (cm)	Age	Voice Pitch	Gender
A	168	15	Low	??
B	154	42	High	??

Random forest – An ensemble of trees

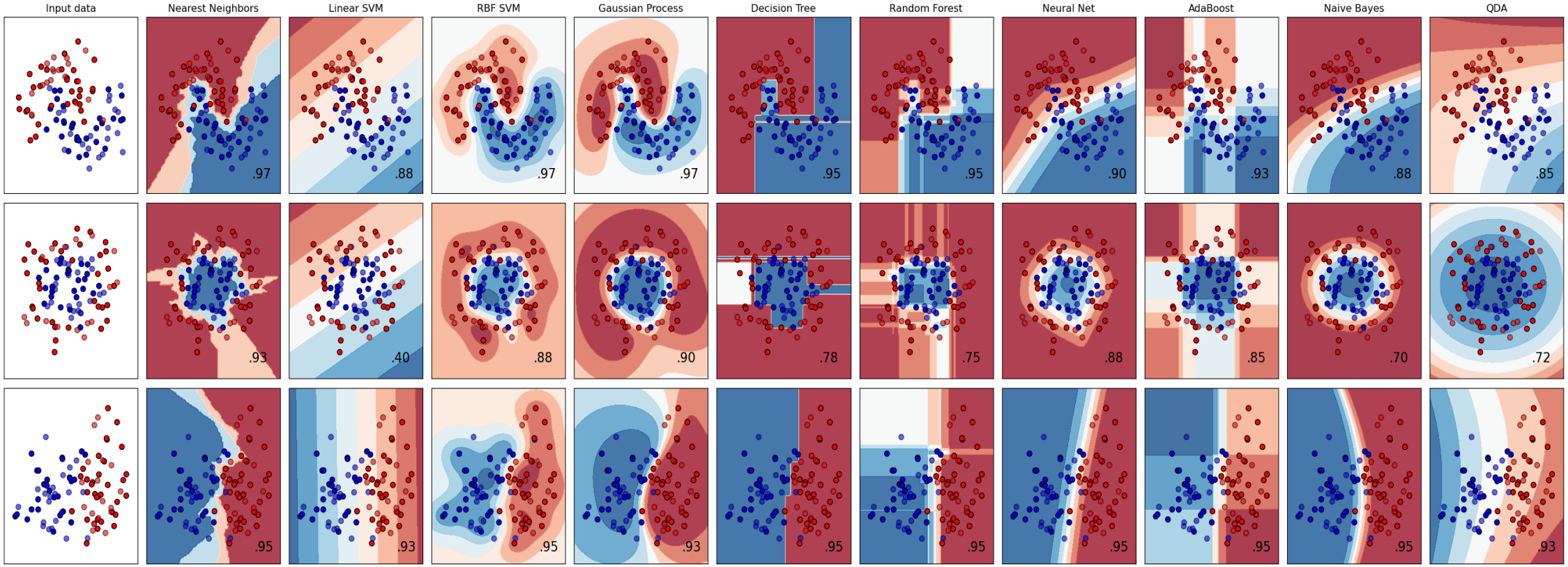
1. **Make Many Decision Trees:**
   - Each trained on a **random subset of the data**.
   - Pick random sample **with replacement** from the training set (Bagging)
2. **Random Feature Selection:**
   - When a tree is splitting nodes, it only considers a **random subset of features** at each step.
   - Adds extra randomness so trees become **more diverse** and don't all make the same splits.
3. **Trees Vote:** Each tree predicts a class, pick the **majority vote**.



Verikas et al. 2016

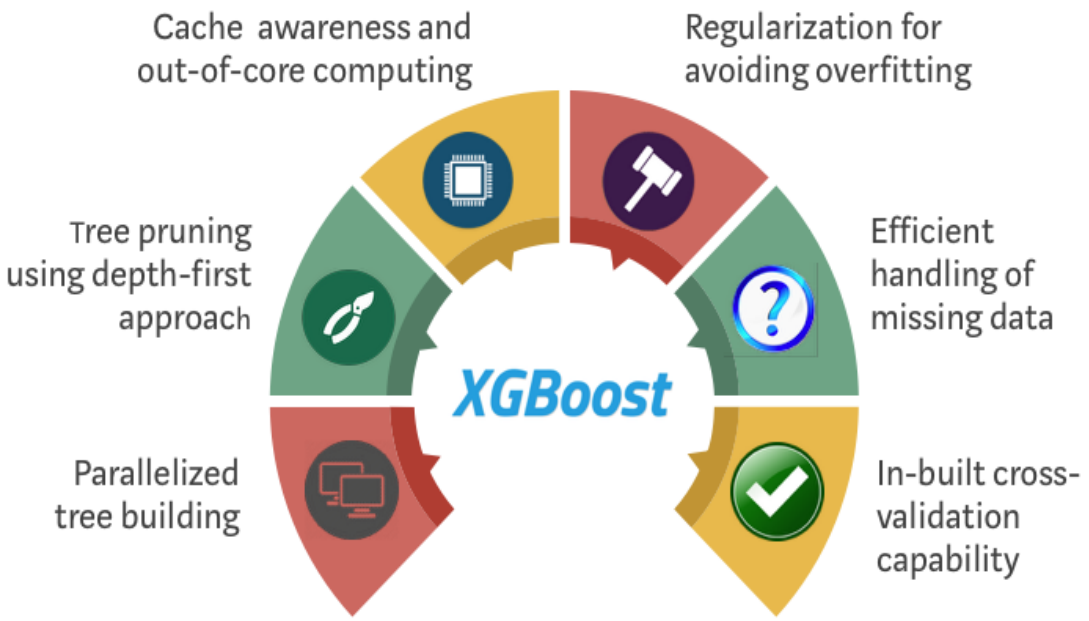
SINTEF

Implementations



Towardsdatascience.com

Implementations



LightGBM

Towardsdatascience.com

How do I evaluate my classifier?

What is a confusion matrix?

- Assumption - *Binary* classification:
  There are only two labels +/- *(or 1/0)*

	Actual Class		
Predicted Class		+	-
	+	**True Positives (TP)**	**False Positives (FP)**
	-	**False Negatives (FN)**	**True Negatives (TN)**

True / False Predictions

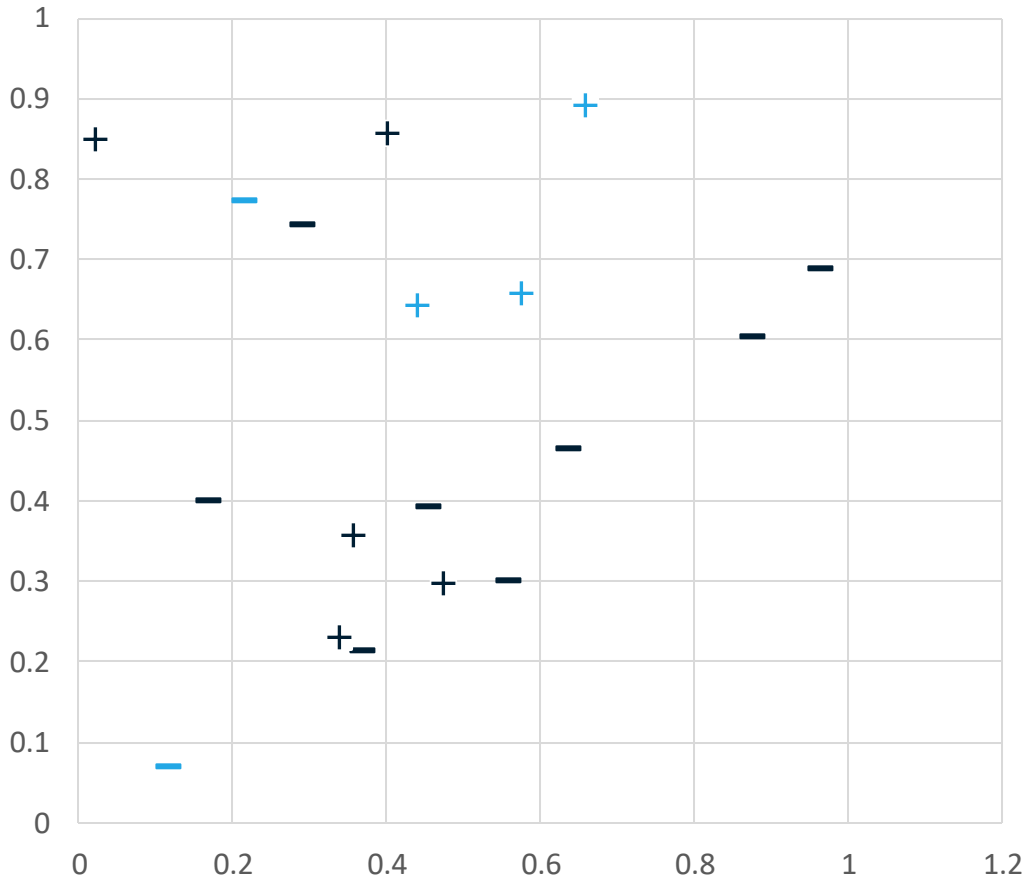
- Assumption:
  There are only two labels +/- *(or 1/0)*
  → *Binary* classification

True Positives (TP) +
datapoints that are *correctly* classified as +

False Positives (FP) +
datapoints that are *falsely* classified as +

True Negatives (TN) -
datapoints that are *correctly* classified as -

False Negatives (FN) -
datapoints that are *falsely* classified as -



SINTEF

True / False Predictions

- Example:
  Predicting Covid
  positive or negative outcome.

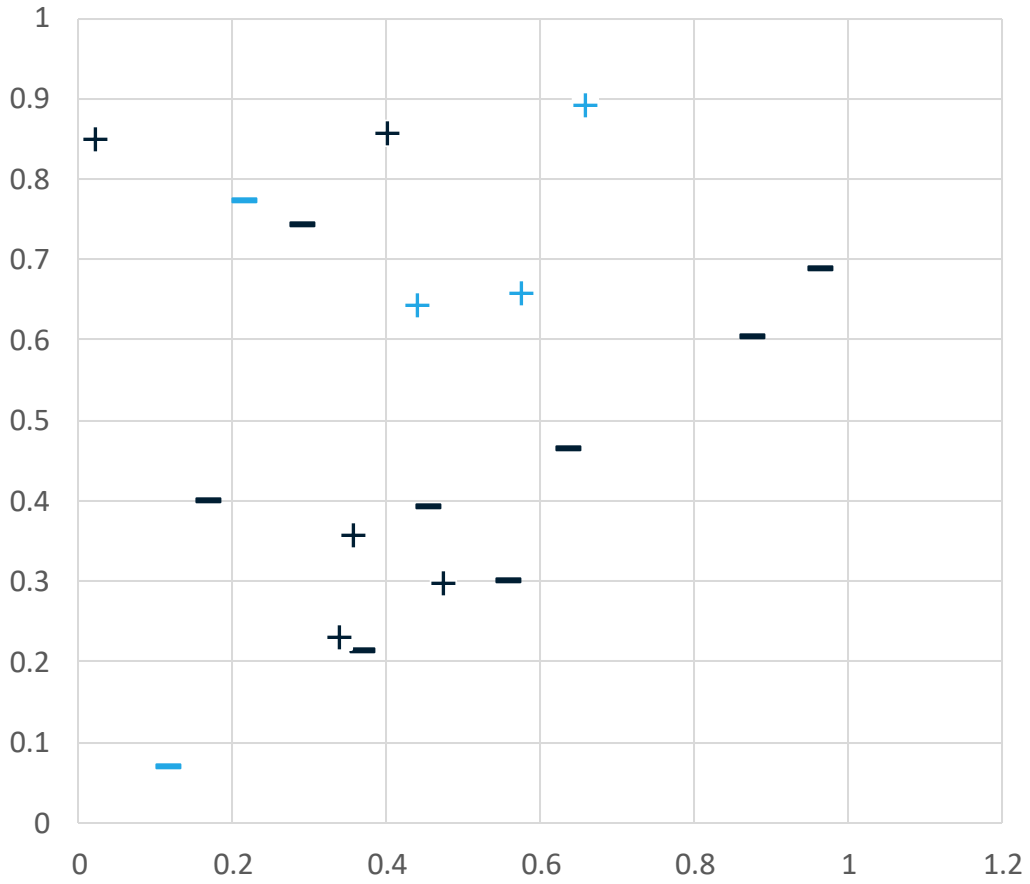
What is the meaning of … ?

True Positives (TP) +
datapoints that are *correctly* classified as +

False Positives (FP) +
datapoints that are *falsely* classified as +

True Negatives (TN) -
datapoints that are *correctly* classified as -

False Negatives (FN) -
datapoints that are *falsely* classified as -



SINTEF

Accuracy - for **binary** classification

- Accuracy: The fraction of correct predictions.

$$Accuracy = \frac{True\ Positives(TP) + True\ Negatives\ (TN)}{Total\ Number\ of\ Predictions}$$

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

Accuracy - for **multi-class** classification

- Accuracy: The fraction of correct predictions.

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

→ This *only* informs about the fraction of correct predictions, and *not* where the errors occur or what type(s) of error we have!

Precision & Recall

- Precision: The fraction of true positive vs. all positive predictions.
  (also: *positive predictive value*)

$$Precision = \frac{True\ Positives(TP)}{Number\ of\ Positive\ Predictions} = \frac{TP}{TP + FP}$$

- Recall: The fraction of true positive predictions vs. actual positives.
  (also: *true positive rate* or *sensitivity*)

$$Recall = \frac{True\ Positives(TP)}{Number\ of\ Actual\ Positives} = \frac{TP}{TP + FN}$$

F1 score – An aggregation of precision and recall

Dealing with one number is sometimes easier

- F1 score is the harmonic mean of precision and recall.

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

Quiz

Given formulas for precision, recall and F1

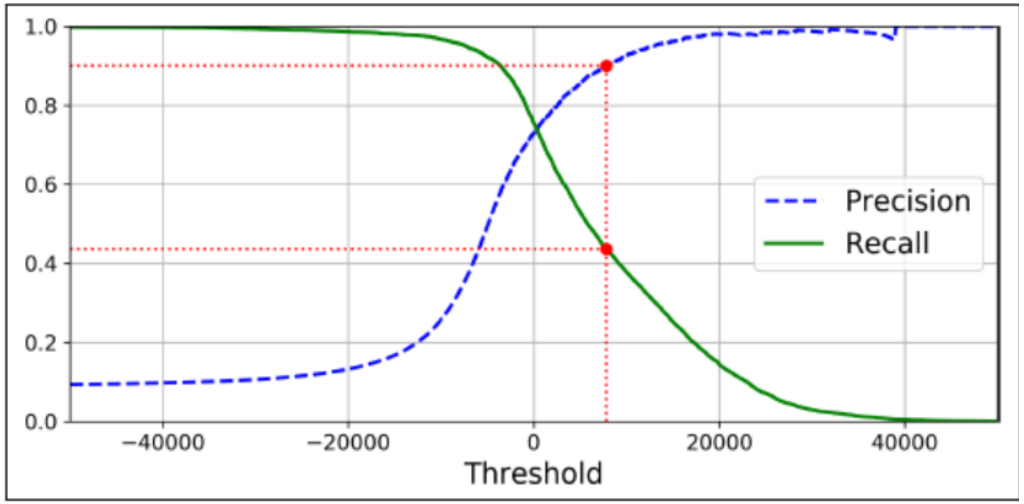
$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

-> What are the highest and lowest F1-scores that can be achieved?

Precision/Recall tradeoff - A discussion



Application:

- A judge wants to use a classifier to predict: "Reoffend (+)", "No more crime (-)"

Reminder:

- **Precision:** Fraction of true positive against all positive predictions $\frac{TP}{TP + FP}$
- **Recall:** Fraction of true positive against all positives $\frac{TP}{TP + FN}$

What do we want to achieve?

- Safe streets: Guarantee catching "**Reoffend**"
  - -> Low **FP** (high **FN**) -> High *Precision -> Low recall*
- Individual justice: Guarantee letting free those that "**No more crime**"
  - -> Low **FN (**high **FP)**-> High Recall -> Low precision

SINTEF

Other metrics for classification

		Predicted condition			
	Total population $= P + N$	**Predicted positive**	**Predicted negative**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate type II error [c] $= \frac{FN}{P} = 1 - TPR$
	Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{TP + FP} = 1 - FDR$	Negative predictive value (NPV) $= \frac{TN}{TN + FN} = 1 - FOR$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{TP + FP} = 1 - PPV$	False omission rate (FOR) $= \frac{FN}{TN + FN} = 1 - NPV$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2\,PPV \times TPR}{PPV + TPR} = \frac{2\,TP}{2\,TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	*phi* or Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV}$ $- \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Sources: [1][2][3][4][5][6][7][8] view · talk · edit

Source: Wikipedia

SINTEF

Quiz/Discussion question

Who is in the best position to decide which metric to optimize for?

- A/ The data scientist/machine learner
- B/ The application owner
- C/ Stakeholders of the application
- D/ All of the above