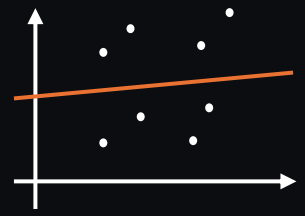
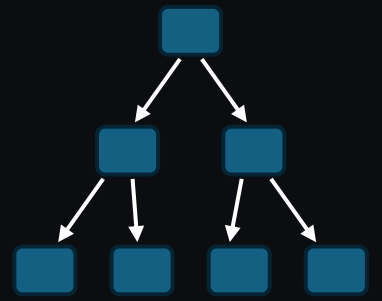
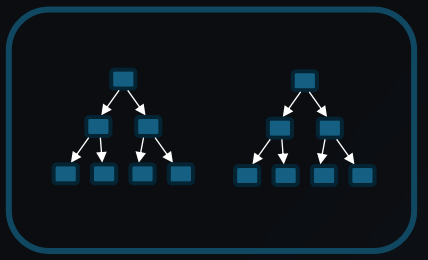
Modellfamiljer för klassificering

Vad betyder Modellfamilj?

- Olika modeller = olika antaganden om mönster
- Tradeoffs:
  - Prestanda
  - Tolkbarhet
  - Robusthet
  - Hastighet
- Samma data → olika resultat







Jämföra modeller

1. Vilka mönster kan modellen fånga?
2. Vad kräver den i preprocessing?
3. Hur hanterar den överanpassning?
4. Hur lätt är den att förklara?

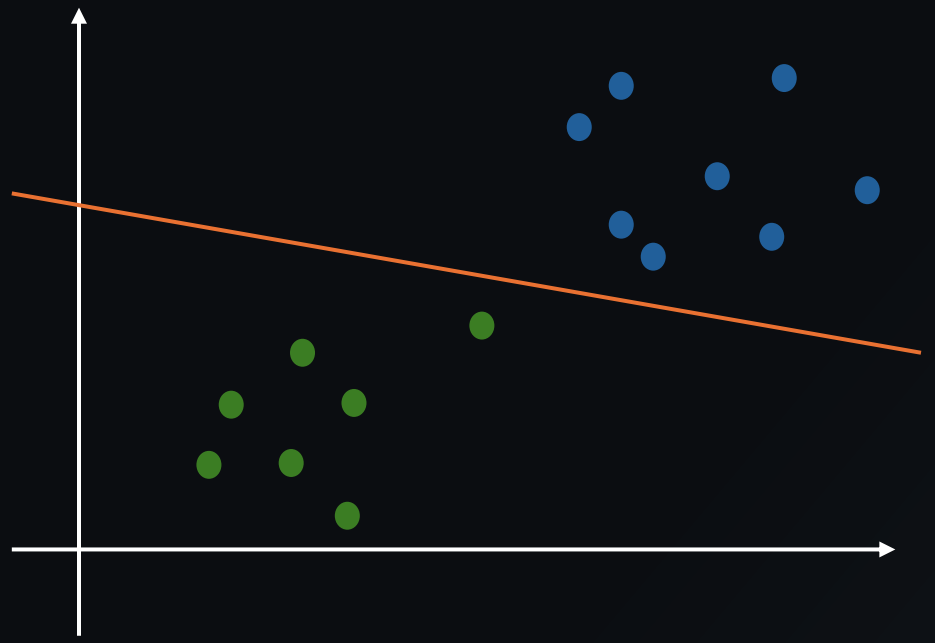
Score → Beslut

- Modellen output är ofta ett score / en sannolikhet
- Beslut kräver en regel: threshold eller top-X

Logistic Regression

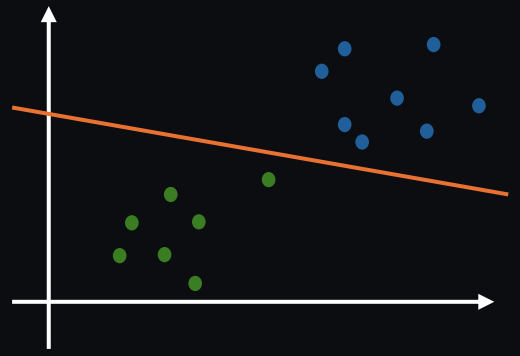
Logistic Regression

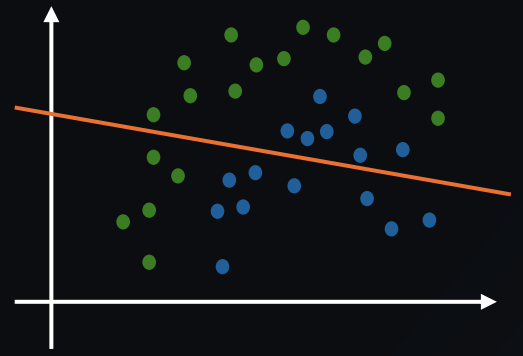
- Linjär modell för klassificering
- Stabil
- Ger ofta bra sannolikheter



Vad betyder ”linjär” i praktiken?

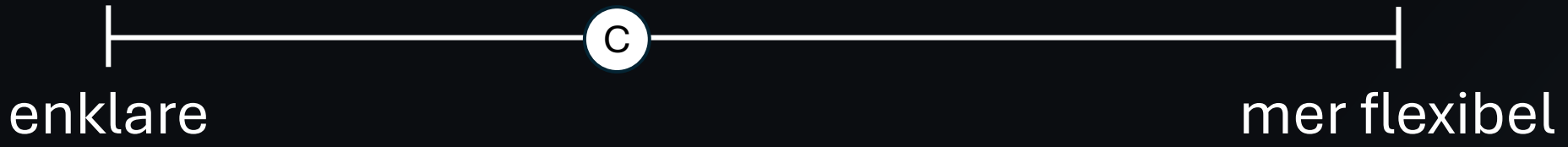
- Rak gräns: linje/plan/hyperplan
- Begränsning: kan missa ”krokiga” mönster utan bättre features





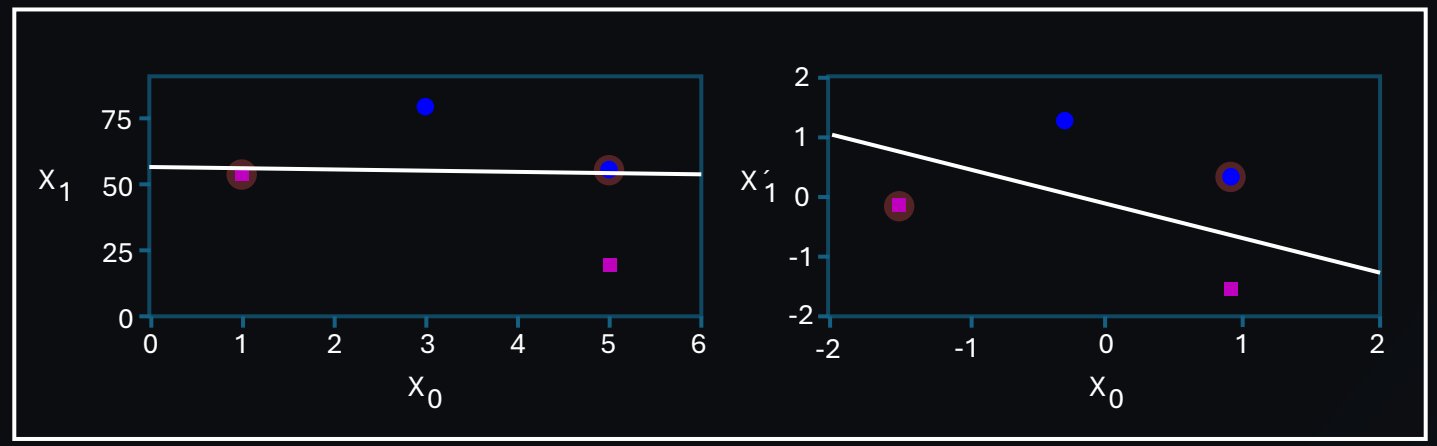
Regularisering

- Regularisering = ”håll modellen enkel”
- Moverkar överanpassning
- Scikit-learn: mindre C → mer regularisering



Preprocessing

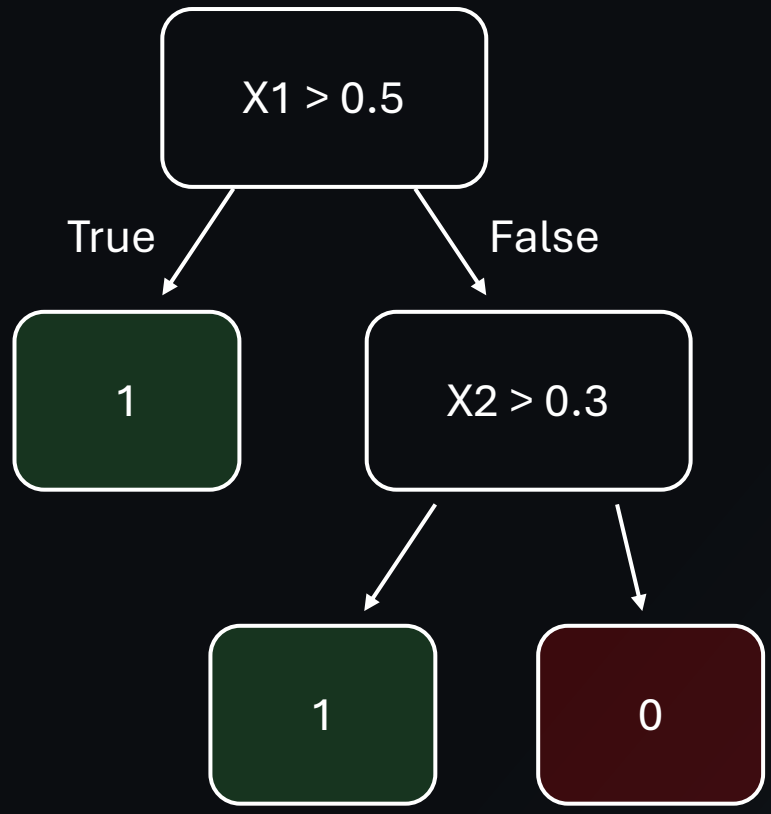
- Skalning är ofta viktigt
- Kategorier behöver kodas (t.ex. one-hot)
- Missing values kräver en strategi



Decision Trees

Decision Tree: intuition

- Bygger regler: ”om X > t”
- Fångar icke-linjära mönster
- Kan vara intuitiv att förstå



Styrkor och svagheter hos träd

- Styrkor:
  - Fångar komplexa samband
  - Ofta ingen scaling
  - Intuitiv logik

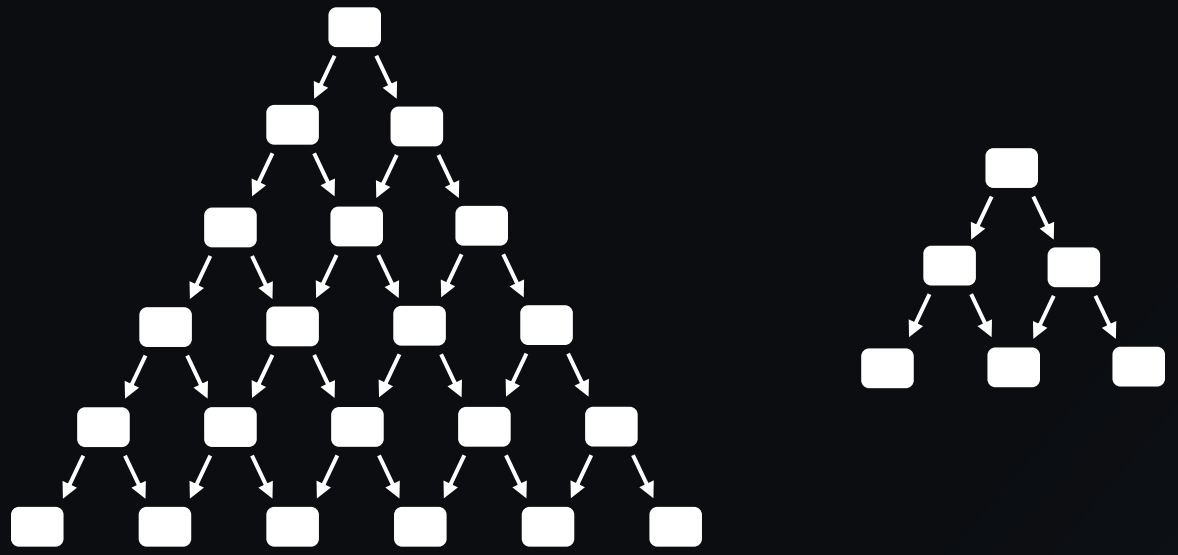
- Svagheter:
  - Överanpassar lätt
  - Instabilt: små dataändringar → nytt träd

Overfitting i träd

- Djupare träd → fler och mer detaljerade regler
- För små blad → “specialregler”
- Stort gap mellan train och test = varningssignal

Trädets reglage

- max_depth
- min_samples_leaf
- min_samples_split



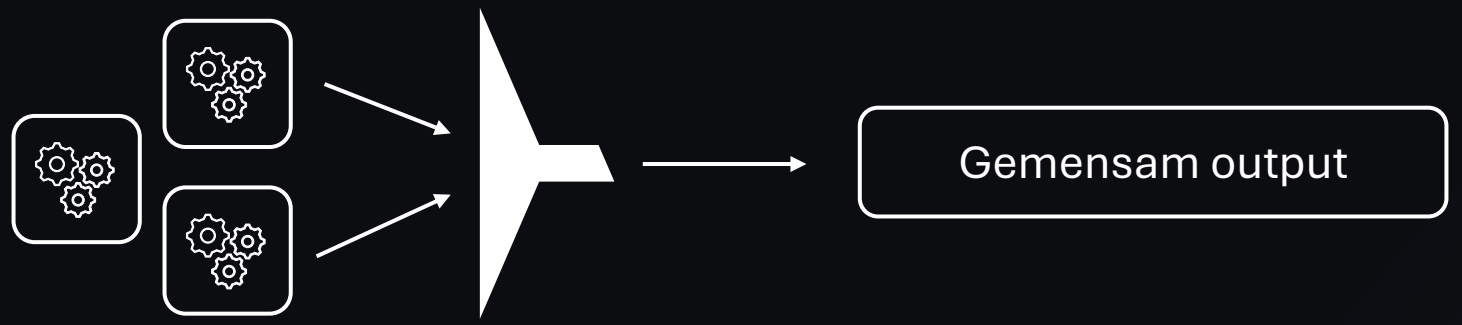
Preprocessing för träd

- Scaling behövs oftast inte
- Kategorier/missing kräver fortfarande strategi
- Viktigt med konsekvens (pipeline)

Ensemble learning & Random Forest

Ensemble learning: grundidén

- Kombinera flera modeller
- Målet: stabilare och ofta bättre generalisering
- ”Wisdom of the crowd”



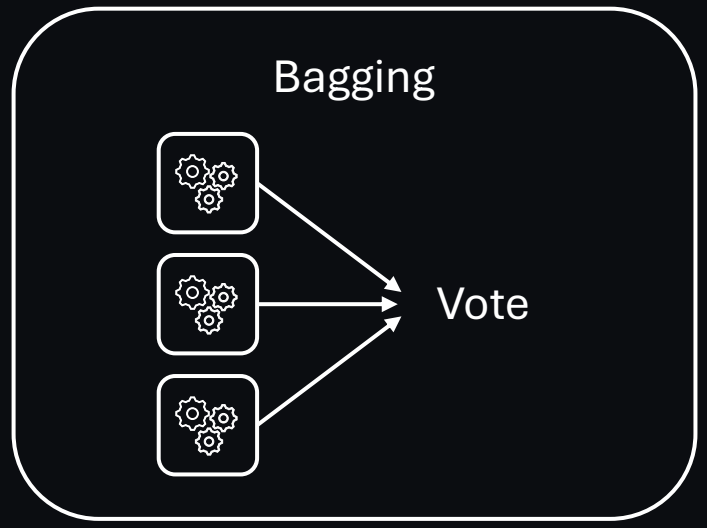
Hard vs Soft voting

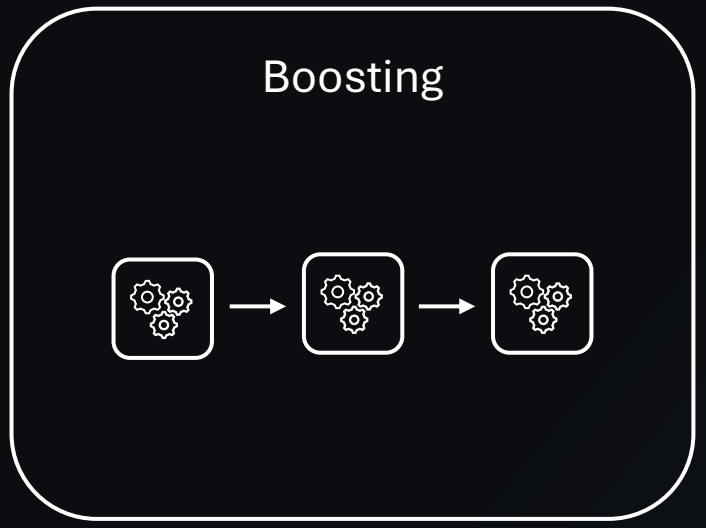
- Hard voting: majoritet på klass (0/1)
- Soft voting: medel av sannolikheter → beslut
- Soft kräver att modeller kan ge probabilities

- Hard: [1, 0, 1] → 1
- Soft: [0.6, 0.4, 0.9] → 0.63 → 1

Bagging vs Boosting

- Bagging: många modeller parallellt på olika sample → vote/medel
  - Pasting: sampling utan återläggning
- Boosting: modeller i serie, nästa fokuserar på tidigare fel





Random Forest

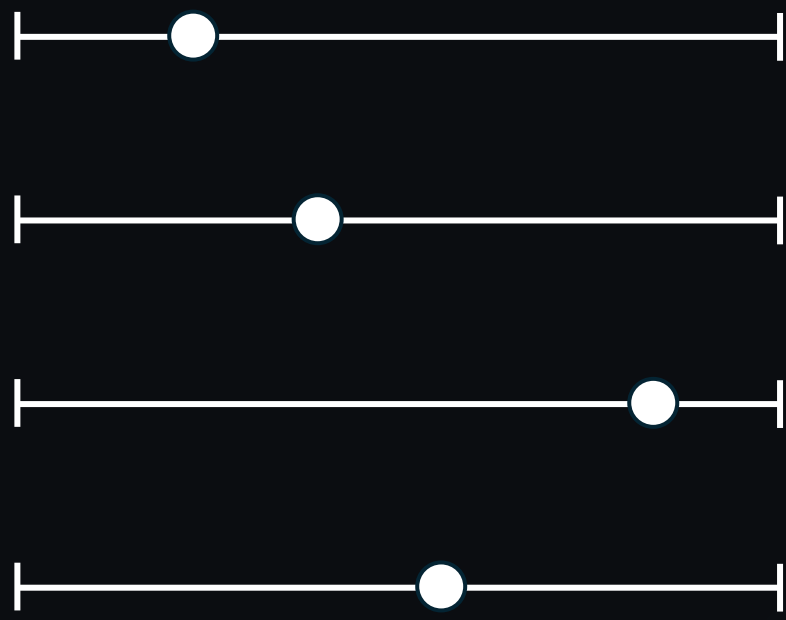
- Ensemble av beslutsträd
- Bagging + slump i features
- Röstar → prediktion

Varför Random Forest ofta är stark

- Enskilt träd: instabilt, överanpassa lätt
- Många träd: stabilare, generaliserar bättre
- Oftast behövs ingen scaling

RF: viktigaste hyperparametrar

- n_estimators
- max_depth
- min_samples_leaf
- max_features



Från score till beslut

Threshold & top-X

Modellen är inte hela systemet

- Modell → score
- Beslut → regel/policy
- Fel har konsekvenser

Data → model → score → decision

Confusion Matrix

- T**P**: Modellen sa positivt och det var positivt
- T**N**: Modellen sa negativ och det var negativt
- F**P**: Modellen sa positiv men det var egentligen negativt – alltså ett falskt alarm
- F**N**: Modellen sa negativ men det var egentligen positivt – alltså en miss

- True/False = hade modellen rätt?
- Positive/Negative = vad sa modellen?

		Predicted	
		Negative (**N**) -	Positive (**P**) +
Actual	Negative -	True Negatives (T**N**)	False Positives (F**P**)
	Positive +	False Negatives (F**N**)	True Positives (T**P**)

Threshold

flytta gränsen, ändra beteendet

- Regel: positiv om proba ≥ t
- Lägre t → fler flaggas (ofta fler TP men också fler FP)
- Högre t → färre flaggas (ofta färre FP men också fler FN)

Top-X

prioritera de högst rankade

- Flagga de X% högst score
- Styr volym
- Fokus: kvalitet i toppen

Probabillity
0.87
0.82
0.73
0.55
0.54
0.51
0.48
0.42

Modellfamiljer för klassificering

Joakim Lindh