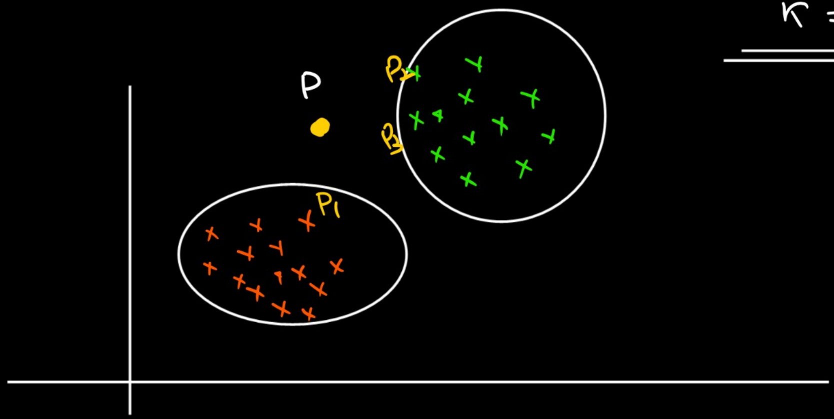
KNN → K - Nearest Neighours

↓

Supervised ML → classification / Regression

K → # neighours

K = 3



Distance metric → Euclidean distance, Manhattan distance, Hamming distance

Continuous value

- P1 —— P = 4 ✓
- P2 —— P = 3 ✓
- P3 —— P = 1 ✓
- P4 —— P = 7
- P5 —— P = 10
- P6 —— P = 15
- P7 —— P = 12

classification → Majority voting → P → green points

Regression ↳ average/mean

Task → KNN → K → user → 5 / 3

①

Heap

① distance

② sort

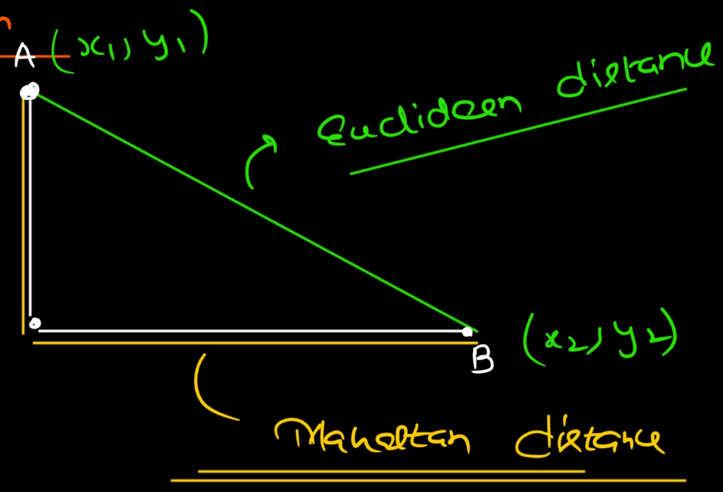
③ first K values → Points → Predict

Euclidean Distance $\rightarrow \sqrt{(x_1 - x)^2 + (y_1 - y)^2}$ → More sensitive to outliers

L2 Norm

Manhattan Distance $\rightarrow |x_1 - x| + |y_1 - y|$

L1 Norm



Hamming distance → string

(how many positions differ)

string $\begin{cases} \rightarrow & A\ T\ C✓\ G✓ \\ \rightarrow & A\ T\ G✓\ G✓ \end{cases}$

$1 \longrightarrow \underline{\underline{1}}$

Quantifies genetic info

binary vector

$1\ 0\ 1\ 1\ 1\ 0\ 0$

$1\ 0\ 0\ 1\ 0\ 0\ 0$

$1 + 1 \longrightarrow \underline{\underline{2}}$

KNN → Lazy learner

↳ Limitation → huge amount of data

$y = mx + c$

↳ optimized value of m & c

Train → it does nothing

Prediction → distance, sort & get K nearest neighbours

Real time usage →

↳ Prediction

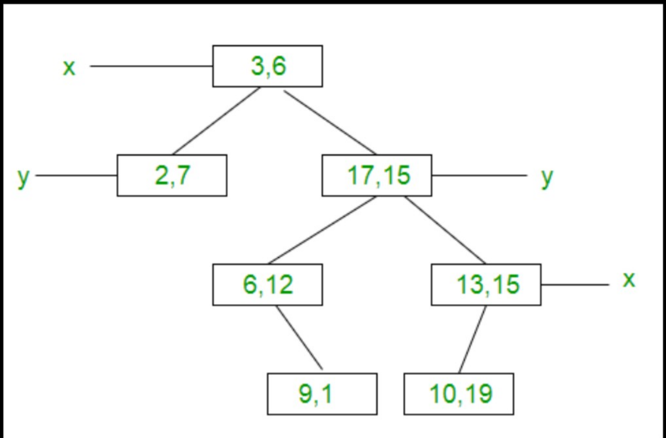
Static → imputation

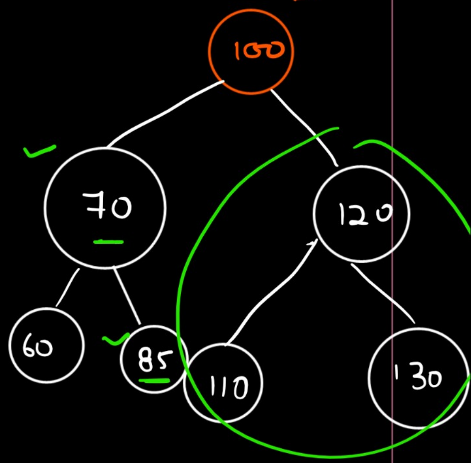
SMOTE —— Task 2

Optimisation in Tree

- BST (Binary Search Tree)
- AVL Tree
- Red Black Tree

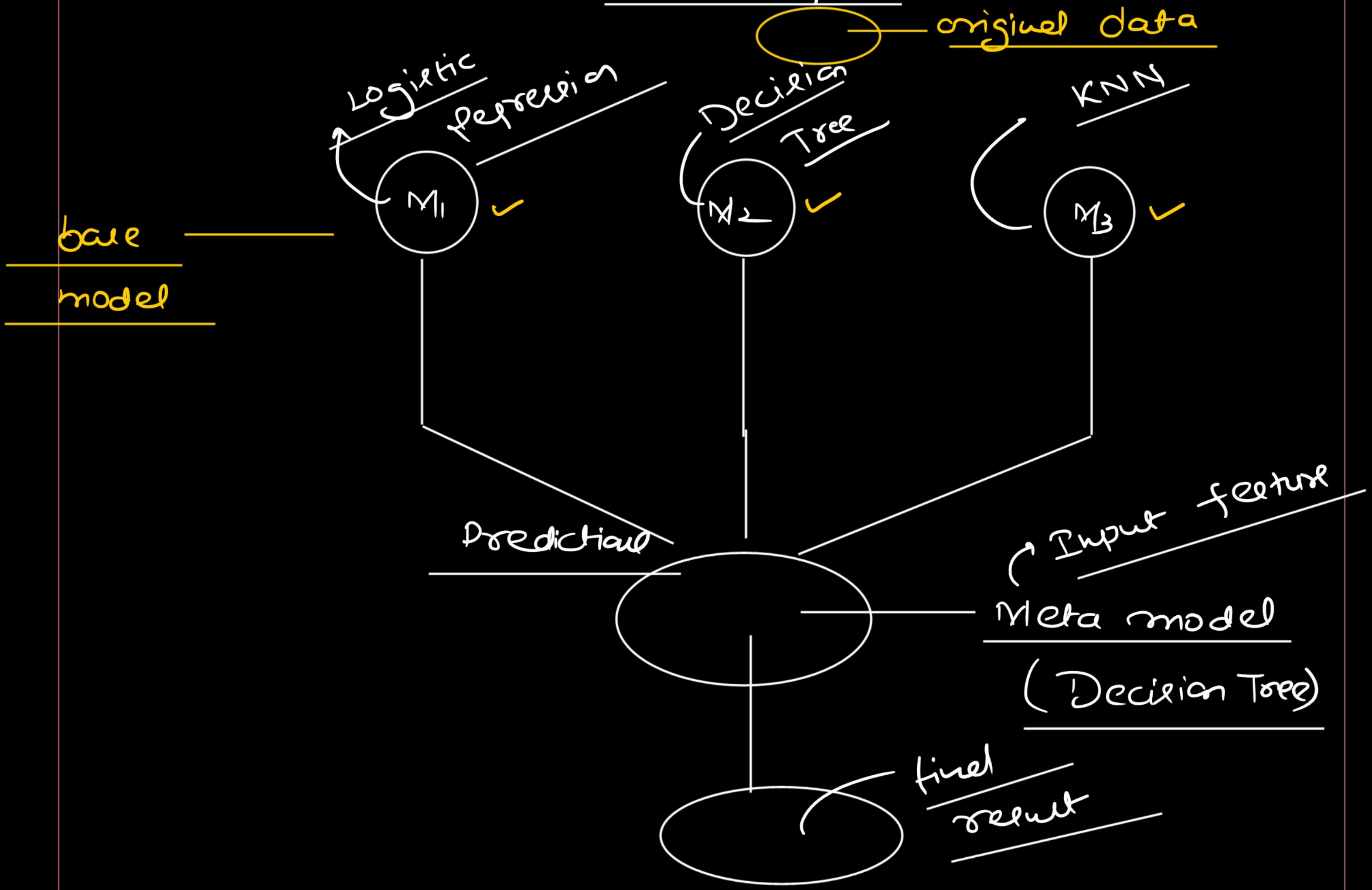
← Kd Tree / ball-tree





xx

Stacking



Stacking

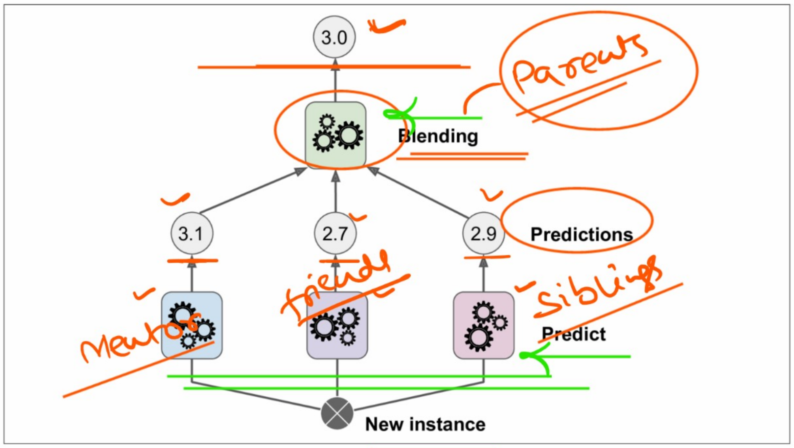


Figure 7-12. Aggregating predictions using a blending predictor