MIDS W207
Applied Machine Learning

Week 07
Live Session Slides

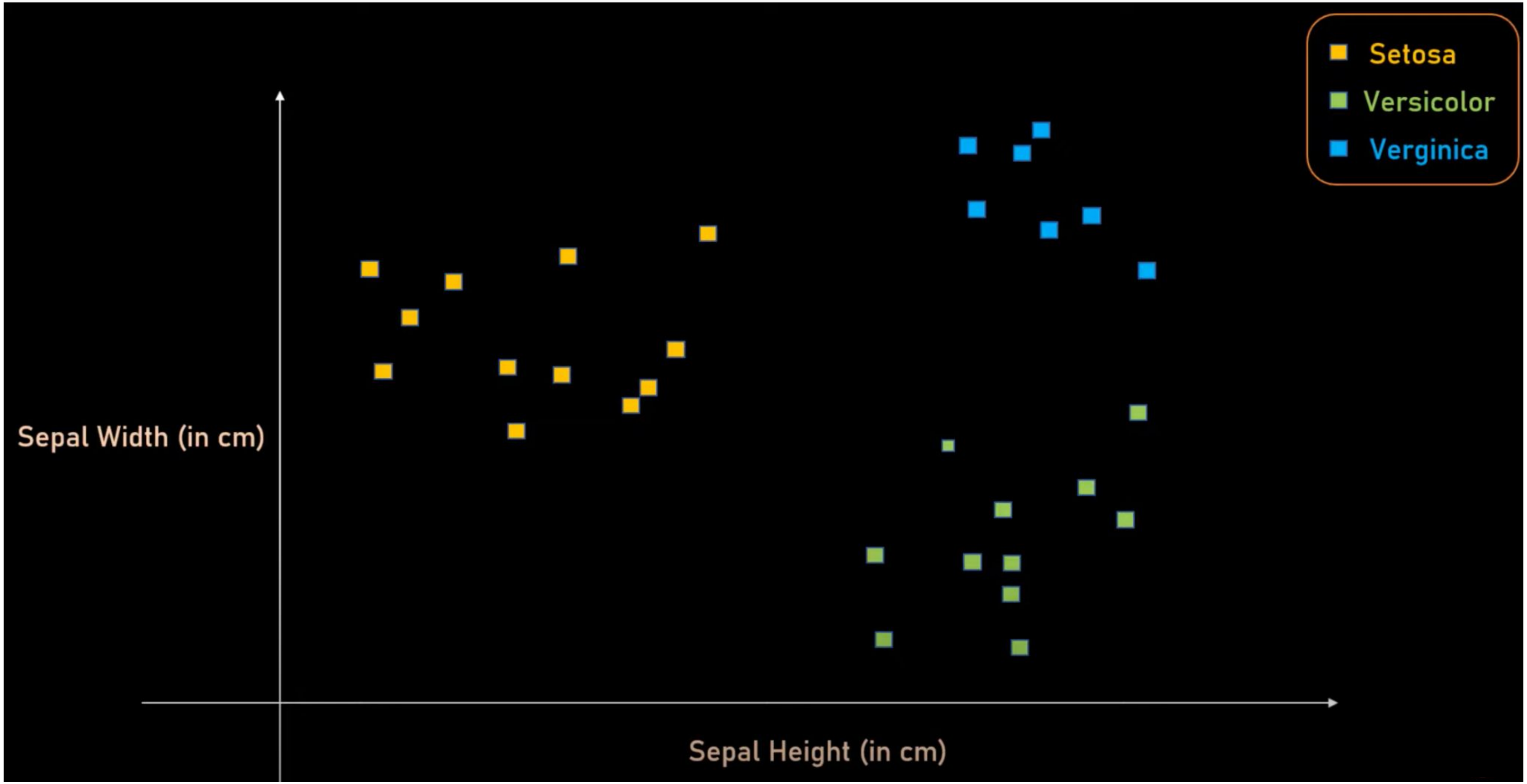
K- Nearest Neighbors Algorithm

Setosa

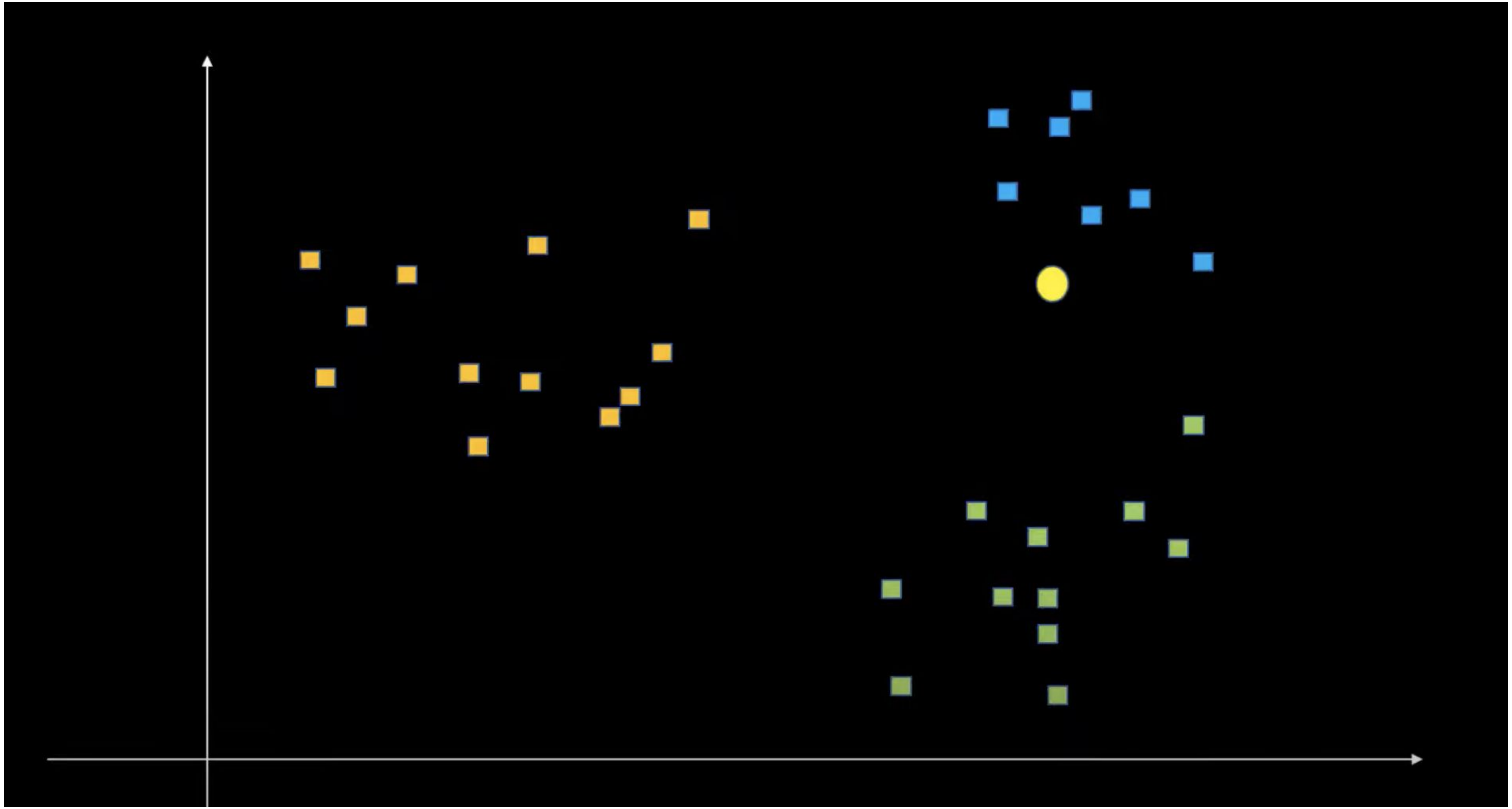
Versicolor

Virginica

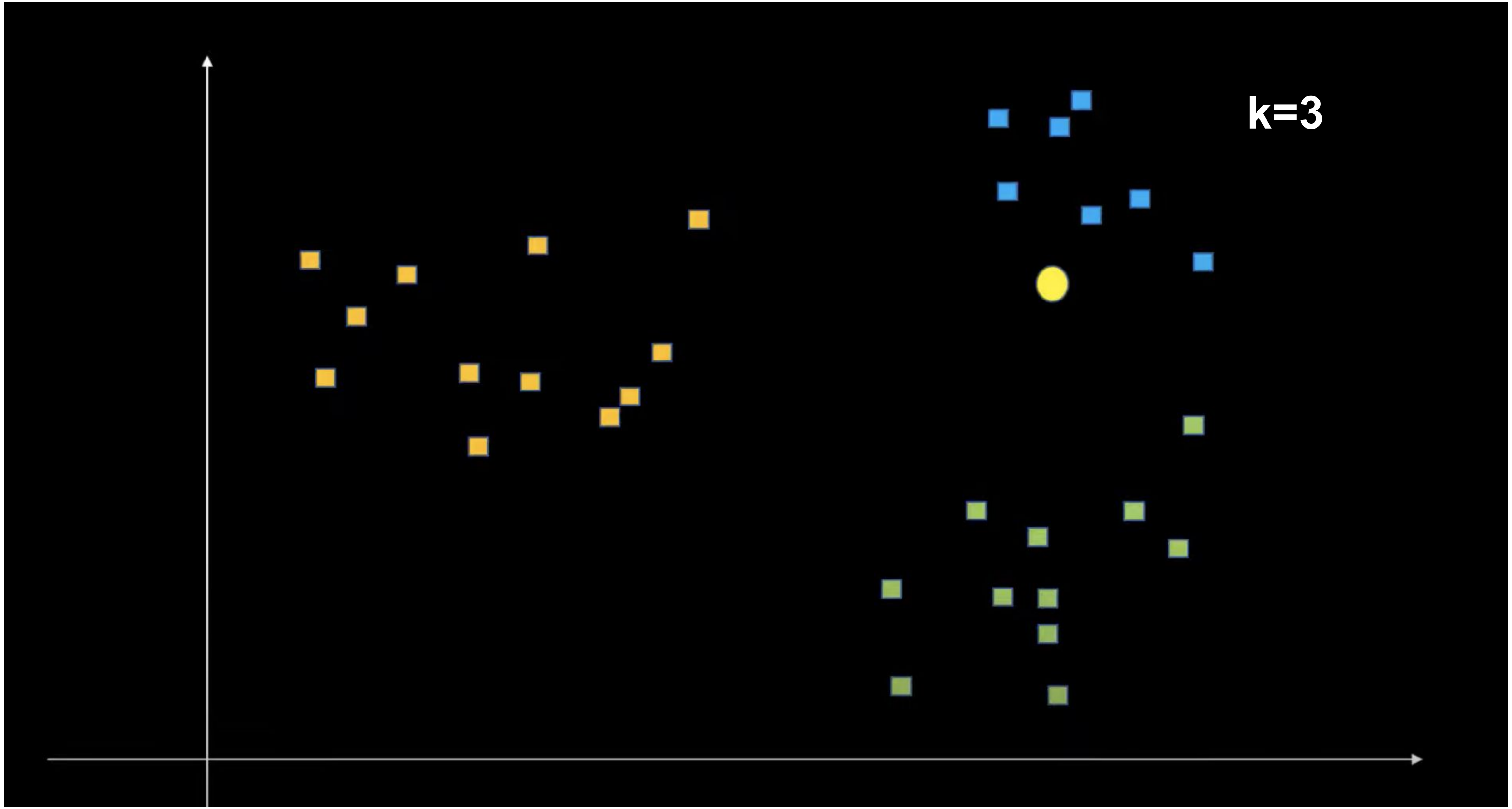
K- Nearest Neighbors Algorithm



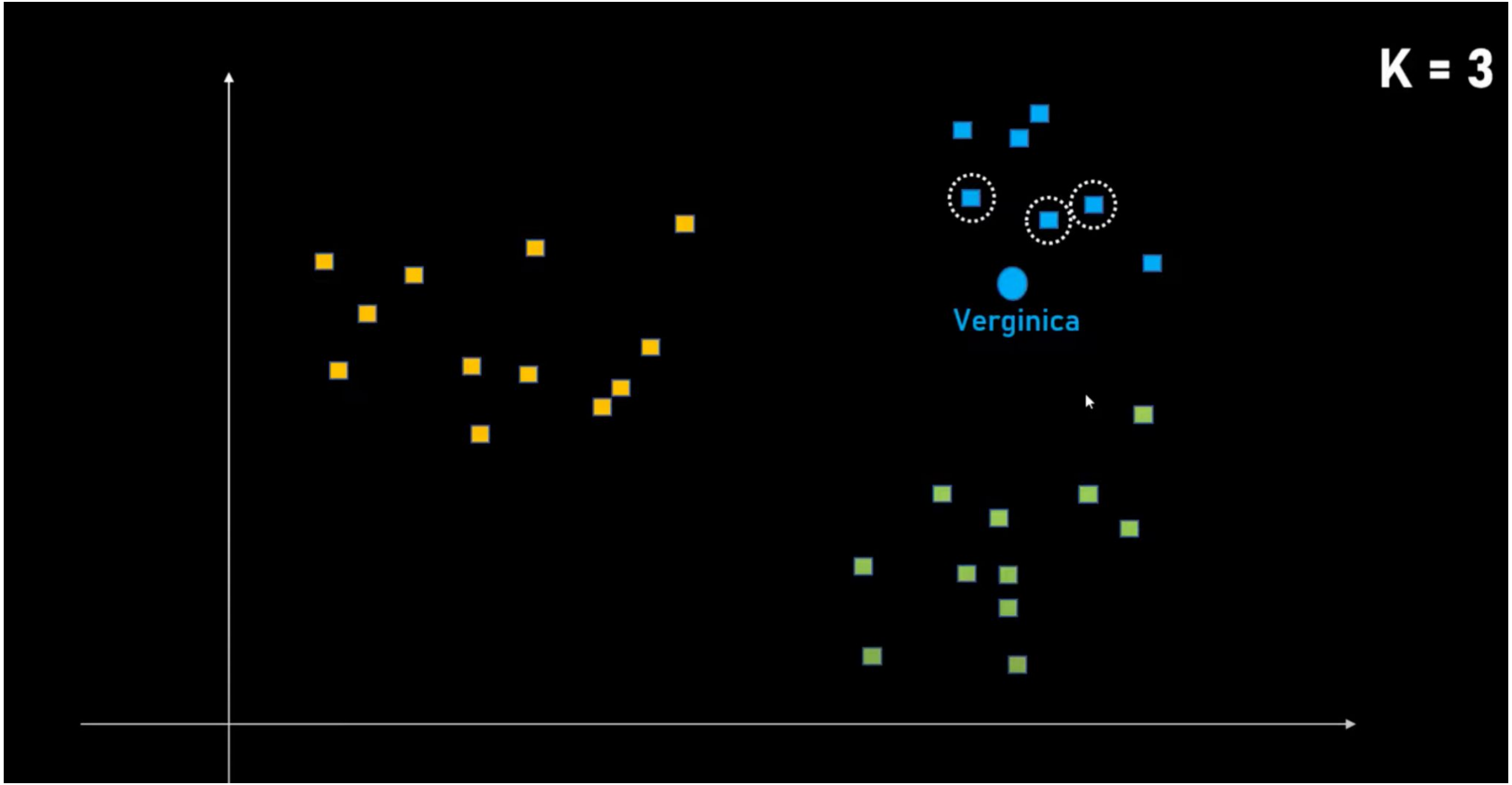
K- Nearest Neighbors Algorithm



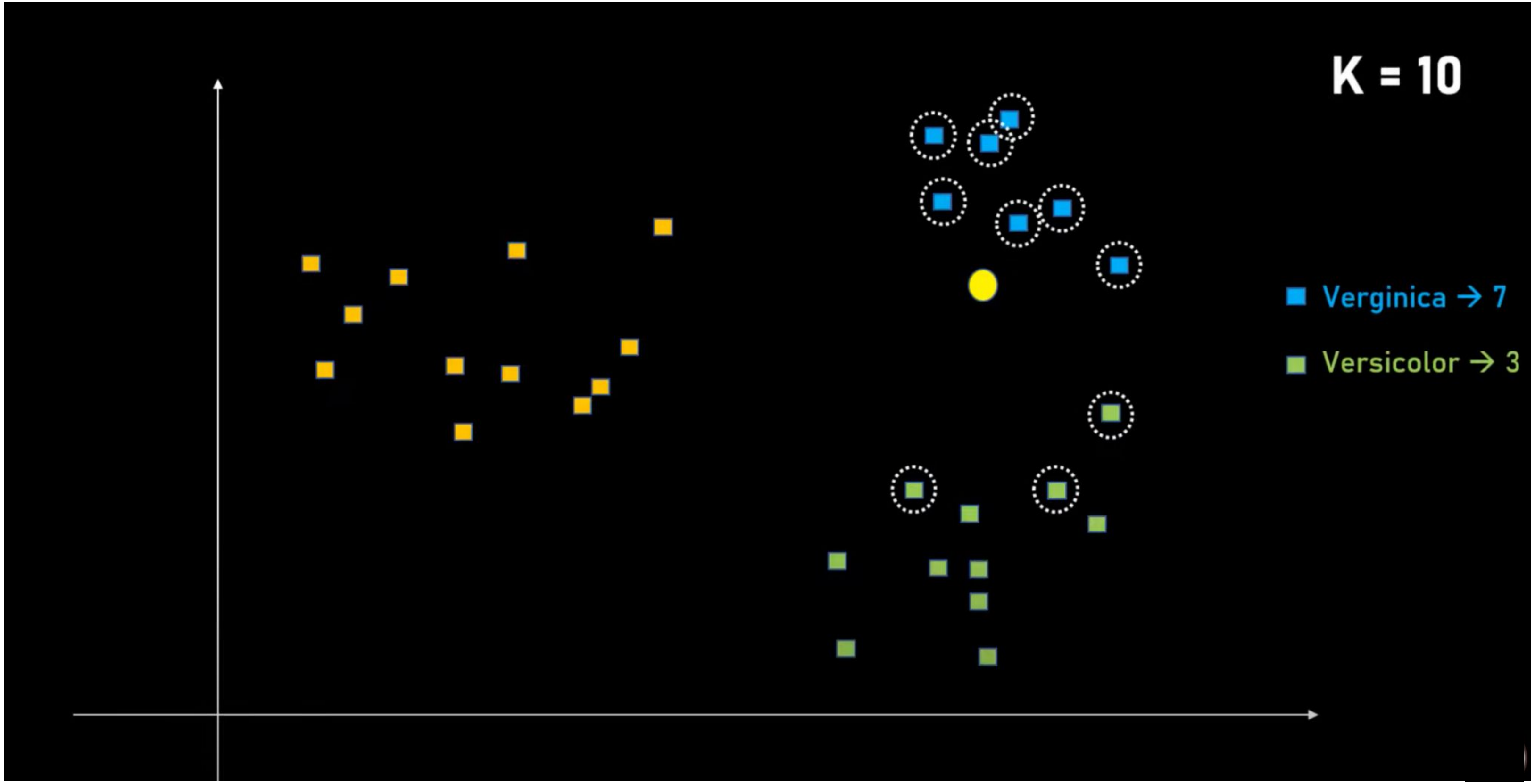
K- Nearest Neighbors Algorithm



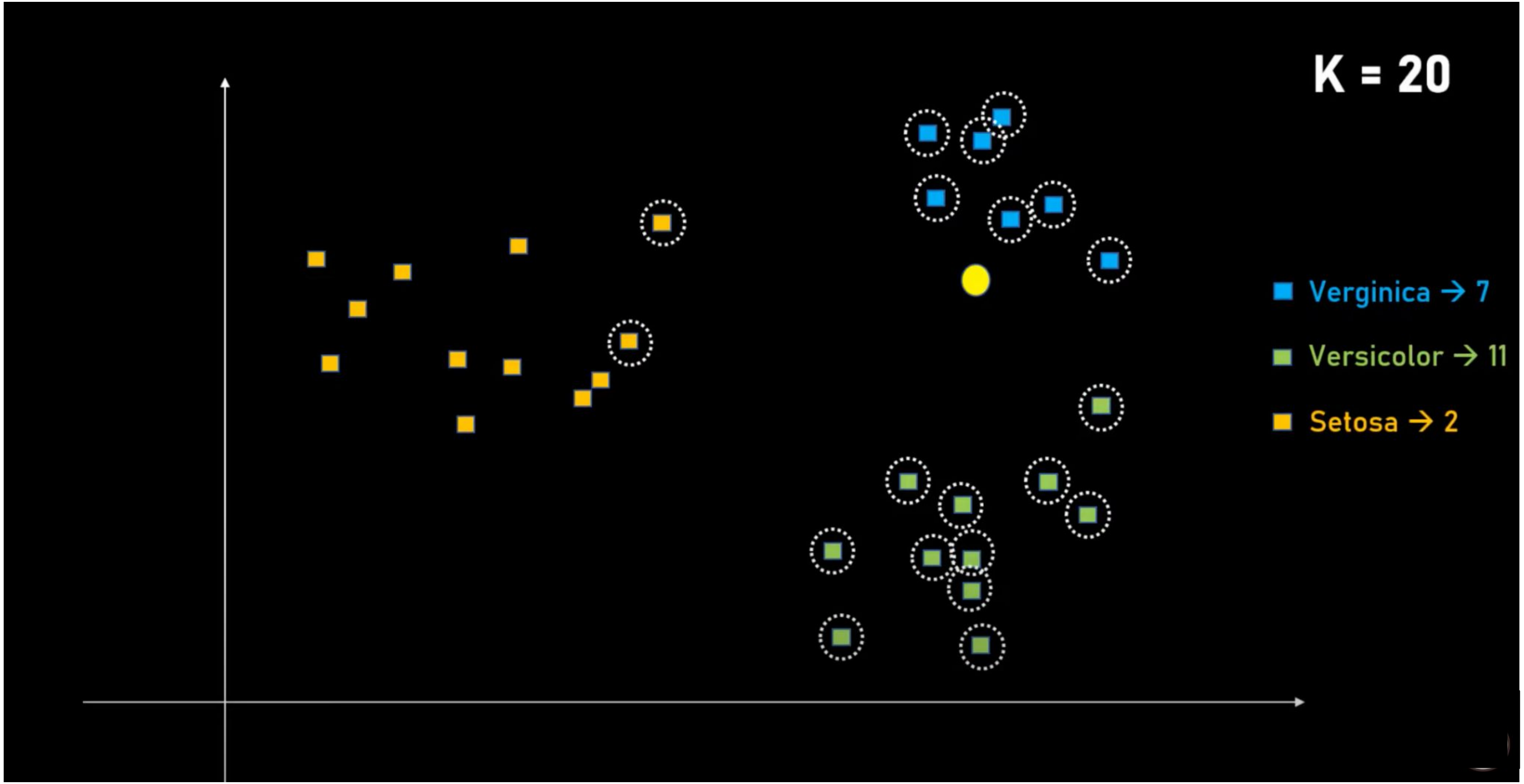
K- Nearest Neighbors Algorithm



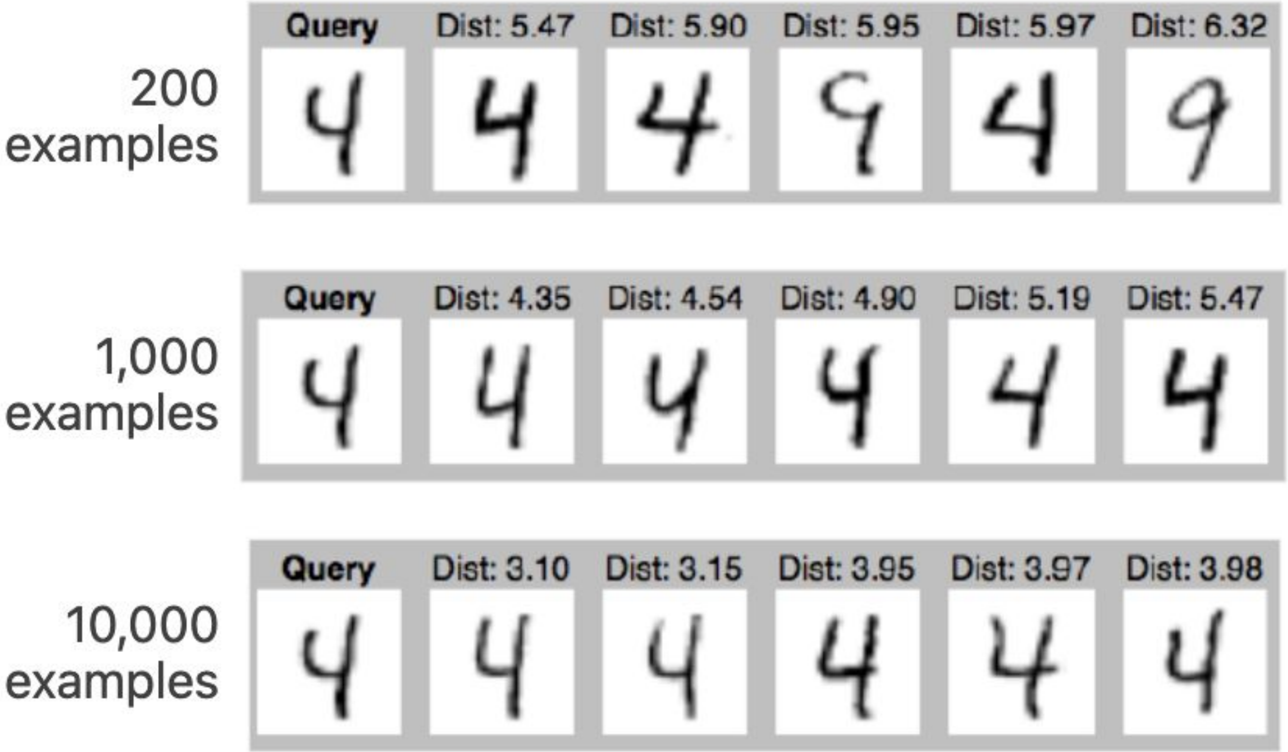
K- Nearest Neighbors Algorithm



K- Nearest Neighbors Algorithm



Digit Classification



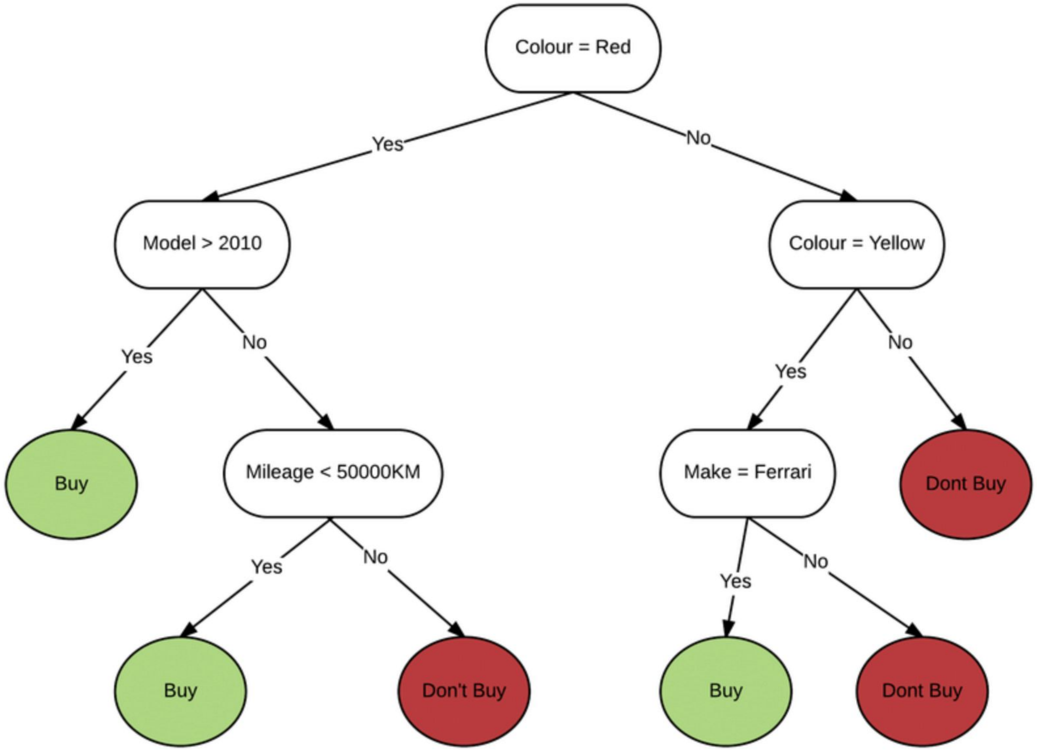
Digit Classification

- Test set size = 10,000 digits

k = 1; Euclidean (L2) distance

Training	Error %	Time (secs)
100	30.0	0.38
1,000	12.1	2.34
10,000	5.3	28.7
60,000	2.7	2202
Deskewing	2.3	
Blurring	1.8	
Pixel shifting	1.2	

Decision Trees

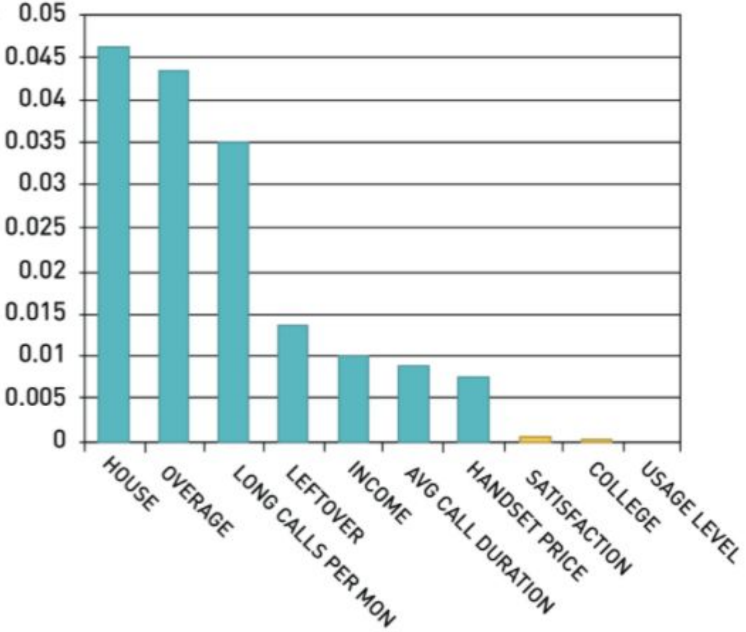


Decision Tree: Customer Churn Example

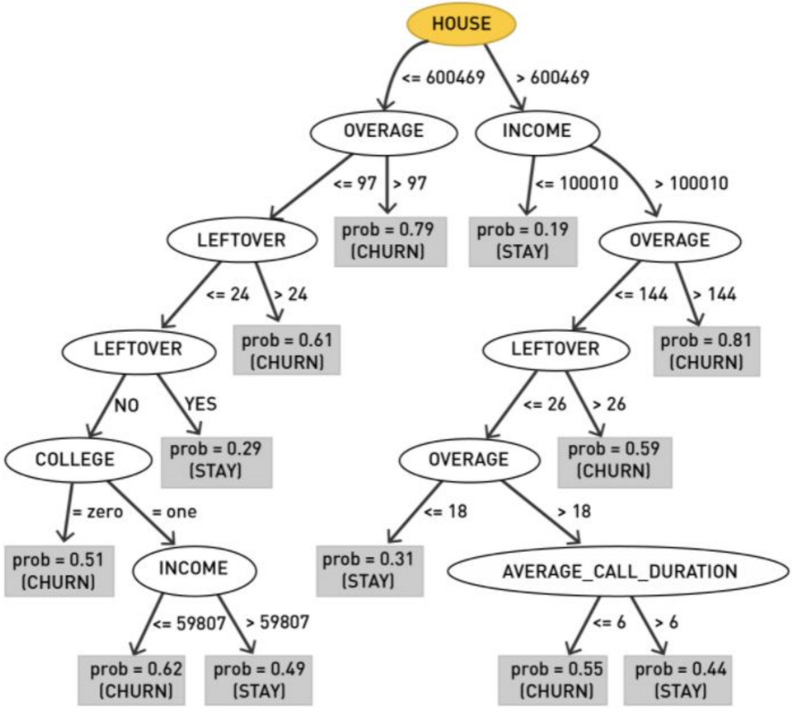
Variable	Explanation
COLLEGE	Is the customer college-educated?
INCOME	Annual income
OVERAGE	Average overcharges per month
LEFTOVER	Average number of leftover minutes per month
HOUSE	Estimated value of dwelling (from census tract)
HANDSET_PRICE	Cost of phone
LONG_CALLS_PER_MONTH	Average number of long calls (15 min or over) per month
AVERAGE_CALL_DURATION	Average duration of a call
REPORTED_SATISFACTION	Reported level of satisfaction
REPORTED_USAGE_LEVEL	Self-reported usage level
LEAVE	*Target variable: Did the customer stay or leave (churn)?*

Decision Tree: Customer Churn Example

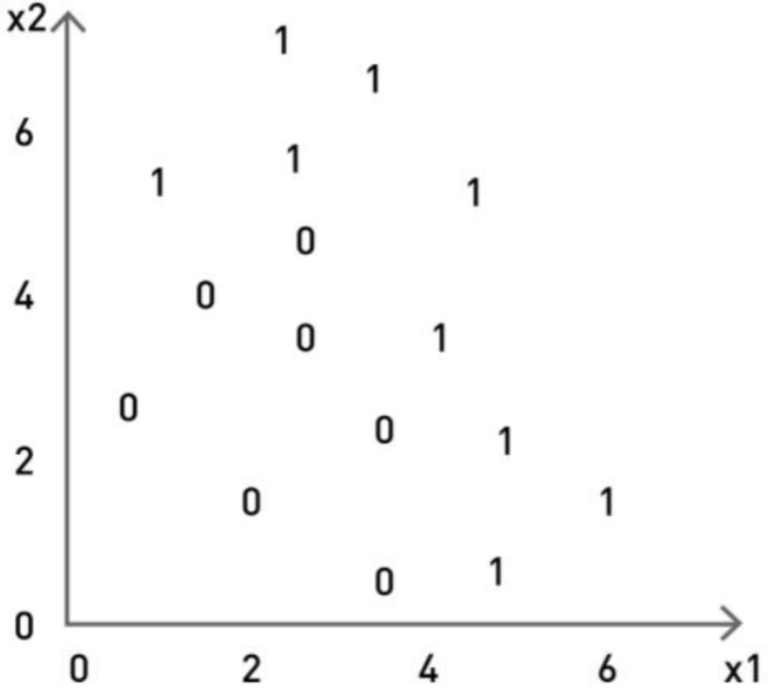
Rank	Info. Gain	Attribute Name
1	0.0461296	HOUSE
2	0.0435518	OVERAGE
3	0.0350337	LONG CALLS PER MON
4	0.013648	LEFTOVER
5	0.0100534	INCOME
6	0.0088899	AVG CALL DURATION
7	0.007624	HANDSET PRICE
8	0.0003062	SATISFACTION
9	0.0001553	COLLEGE
10	0.0000388	USAGE LEVEL



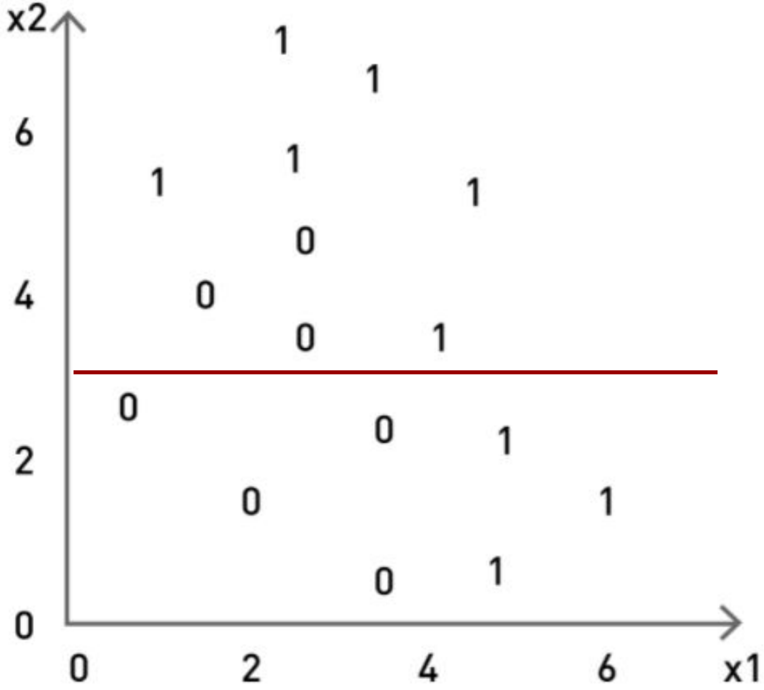
Decision Tree: Customer Churn Example



Decision Tree: Decision Boundaries

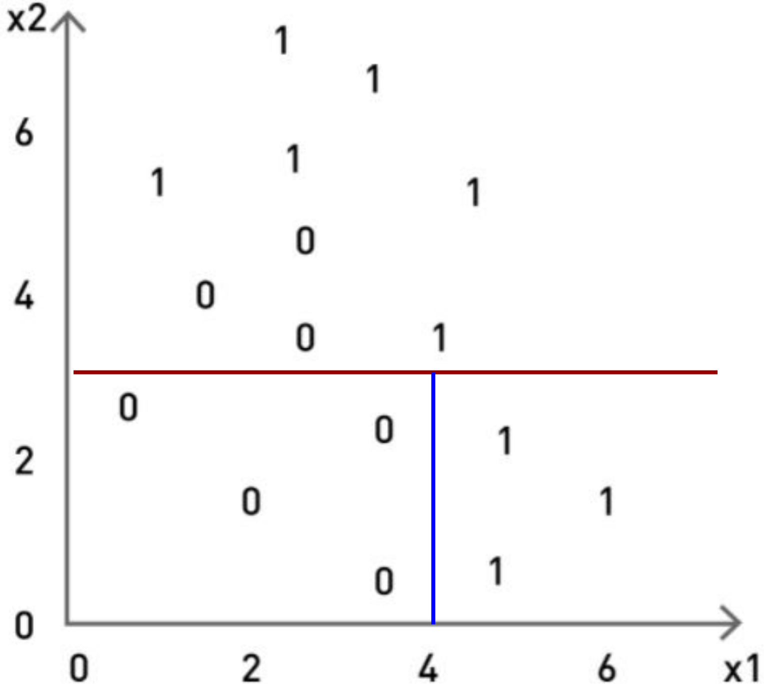


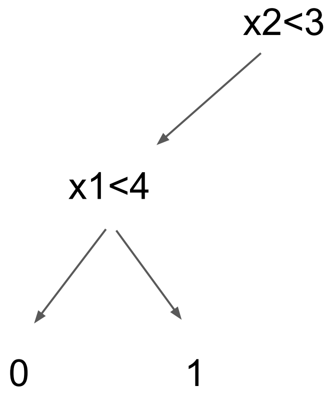
Decision Tree: Decision Boundaries



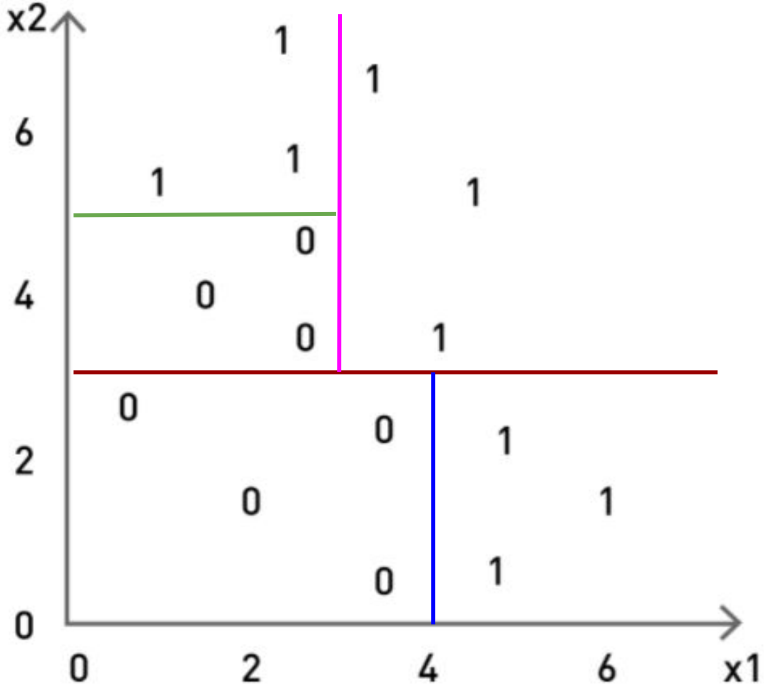
x2<3

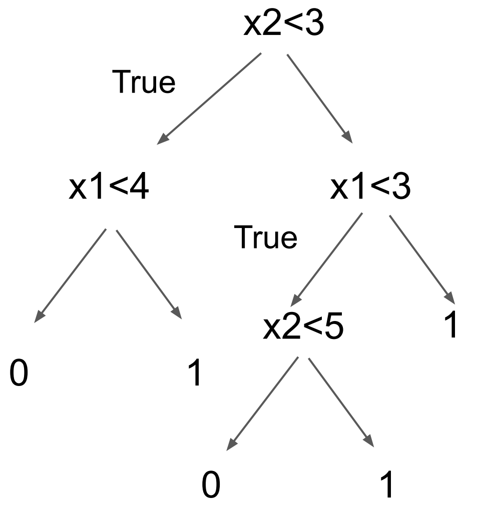
Decision Tree: Decision Boundaries





Decision Tree: Decision Boundaries

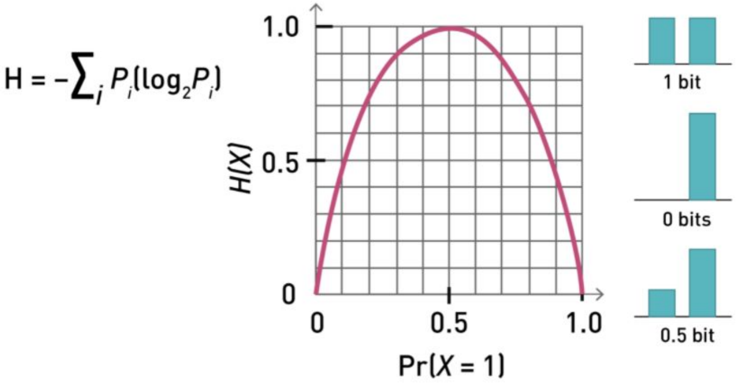




Decision Tree: Entropy and Information Gain

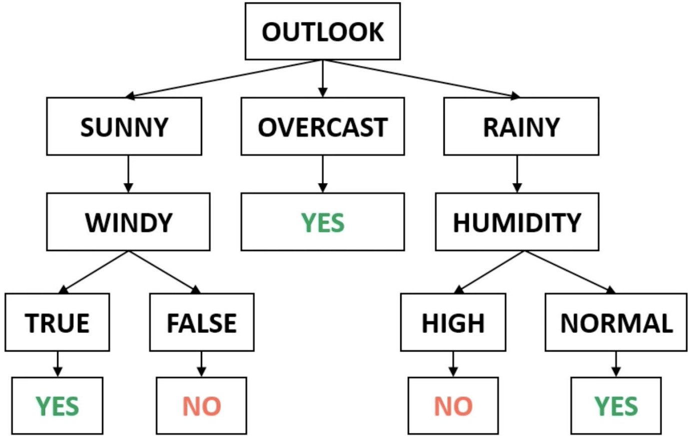
Entropy is the measure of uncertainty
The more uncertainty there is about a random variable, the more information is conveyed about the value
Entropy is maximized when there is complete uncertainty through uniform distribution
0 when there is complete certainty

1



Decision Tree: Example

outlook	temperature	humidity	wind	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cold	normal	false	yes
rainy	cold	normal	true	no
overcast	cold	normal	true	yes
sunny	mild	high	false	no
sunny	cold	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



Decision Tree: Example

The **H(X)** Shannon-entropy of a dicrete random variable **X** with possible values $x_1 x_2 \dots x_n$ and probability mass function **P(X)** is defined as:

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_2 P(x_i)$$

For a homogeneous dataset (all TRUE or all FALSE values, entropy is 0
For a dataset that is equally distributed (same amount of TRUEs and FALSEs, entropy is 1

A branch with entropy more than 1 needs splitting

Decision Tree: Example

outlook	temperature	humidity	wind	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cold	normal	false	yes
rainy	cold	normal	true	no
overcast	cold	normal	true	yes
sunny	mild	high	false	no
sunny	cold	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

PLAYING GOLF

→ **9** times **YES**

→ **5** times **NO**

We just have to use the Shannon-entropy formula to calculate the **H(x)** values

H(PlayingGolf) = H(9,5) =

$$= -(0.64 \log_2 0.64) - (0.36 \log_2 0.36) = 0.94$$

Decision Tree: Example

outlook	temperature	humidity	wind	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cold	normal	false	yes
rainy	cold	normal	true	no
overcast	cold	normal	true	yes
sunny	mild	high	false	no
sunny	cold	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

$$E(T,X) = \sum_{x} P(x)\, E(x)$$

We have to calculate the entropy with respect to a given predictor/feature in order to be able to calculate information gain

		PLAY GOLF	
		YES	**NO**
OUTLOOK	sunny	**2**	**3**
	overcast	**4**	**0**
	rainy	**3**	**2**

(PlayGolf,Outlook) = P(sunny)E(2,3) + P(overcast)E(4,0) + P(rainy)E(3,2)

$$\frac{5}{14}\, 0.971 + \frac{4}{14}\, 0 + \frac{5}{14}\, 0.971 = 0.6936$$

Decision Tree: Example

Information gain: the decrease in entropy after a dataset is split on an attribute/feature

→ feature/attribute with the highest information gain will be the root node in the tree

Information Gain = H(PlayGolf) - E(PlayGolf,Outlook) =
= 0.94 – 0.693 = 0.247

		PLAY GOLF	
		YES	**NO**
	sunny	**2**	**3**
OUTLOOK	overcast	**4**	**0**
	rainy	**3**	**2**

Decision Tree: Example

Information gain: the decrease in entropy after a dataset is split on an attribute/feature

→ feature/attribute with the highest information gain will be the root node in the tree

Information Gain (outlook) = 0.247

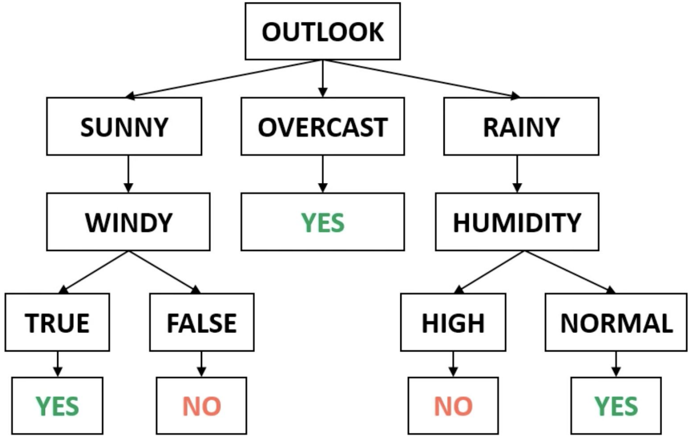
Information Gain (temperature) = 0.029

Information Gain (humidity) = 0.152

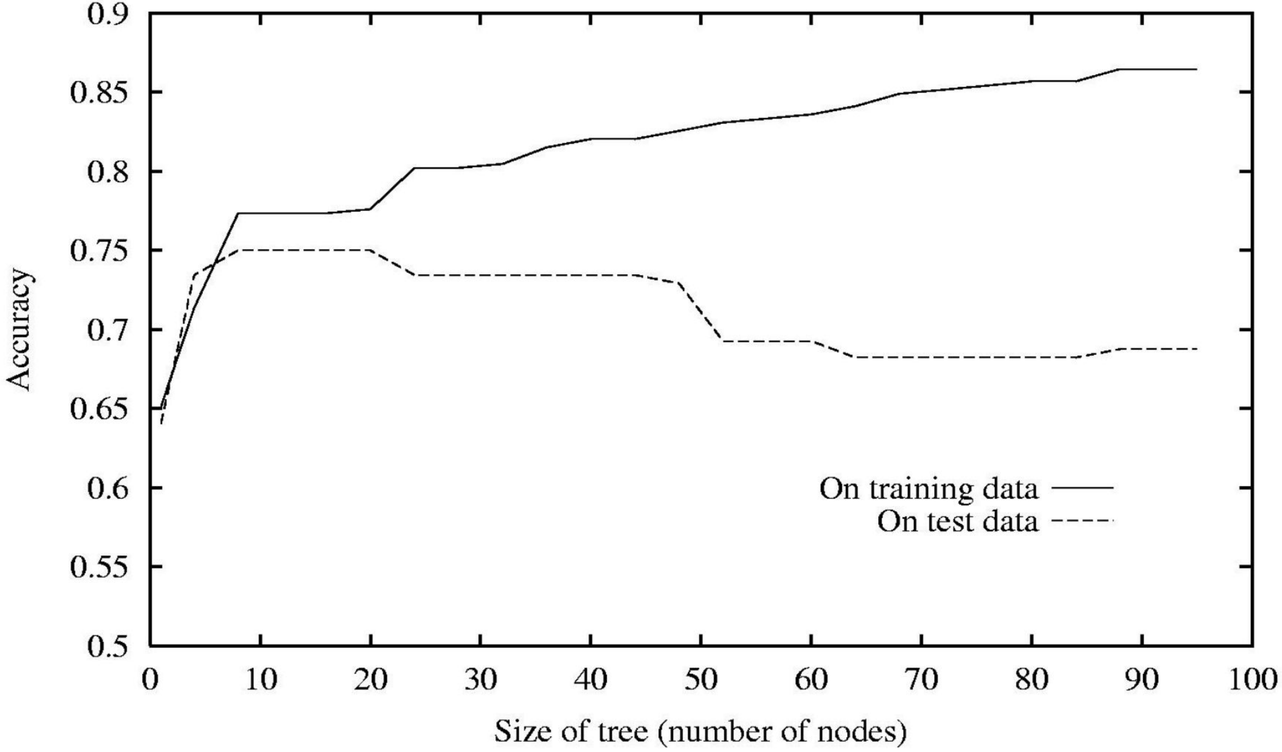
Information Gain (wind) = 0.048

Decision Tree: Example

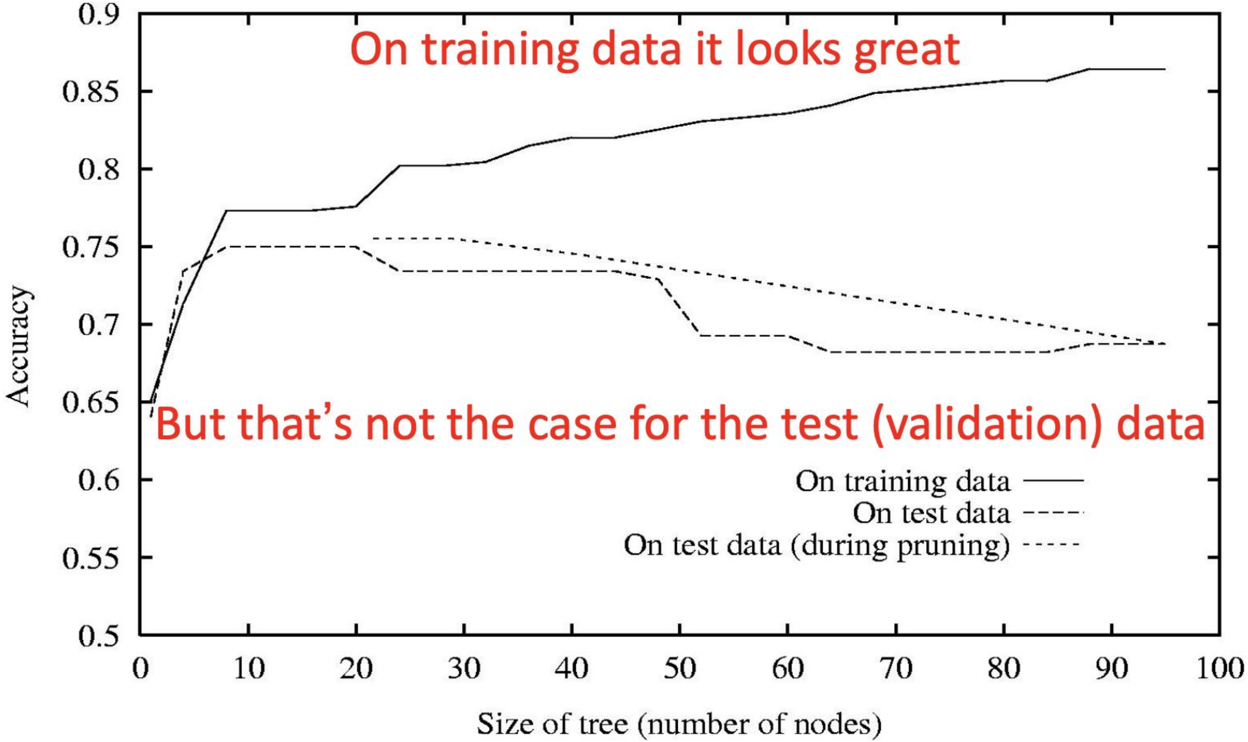
outlook	temperature	humidity	wind	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cold	normal	false	yes
rainy	cold	normal	true	no
overcast	cold	normal	true	yes
sunny	mild	high	false	no
sunny	cold	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



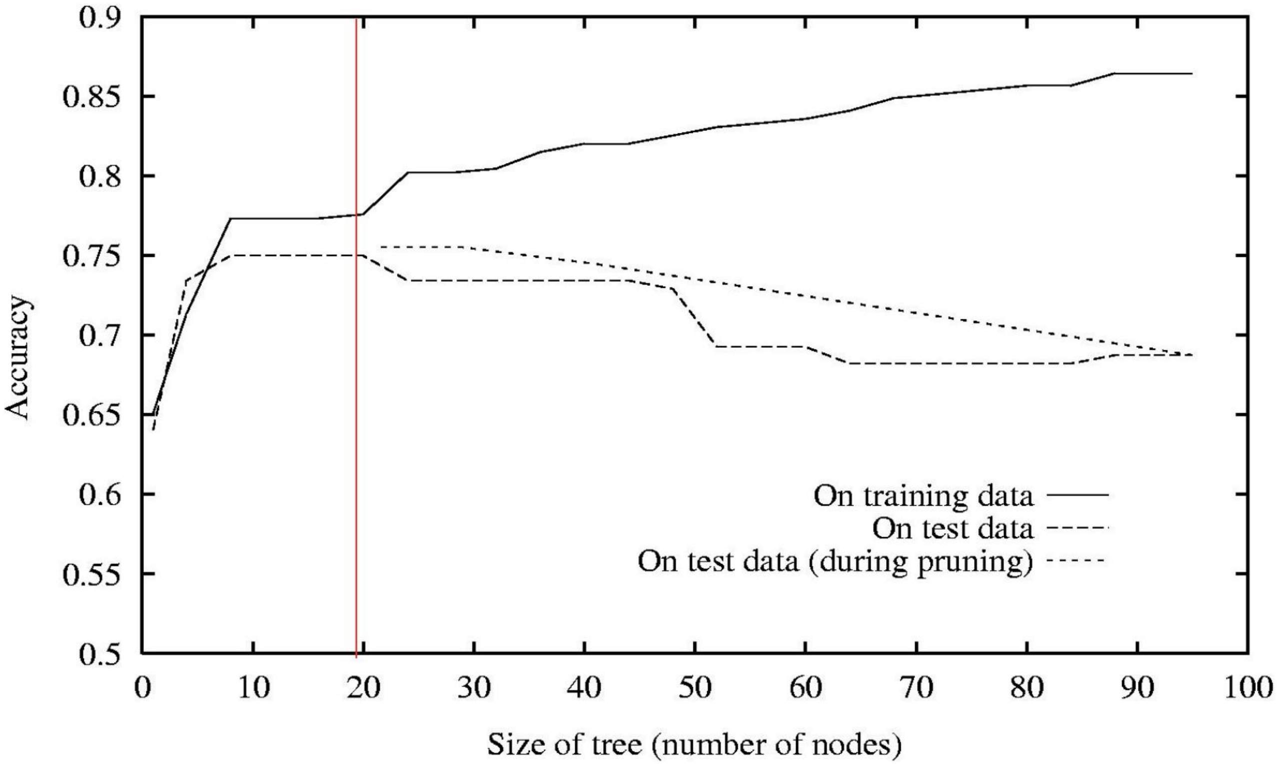
Decision Tree: Overfitting



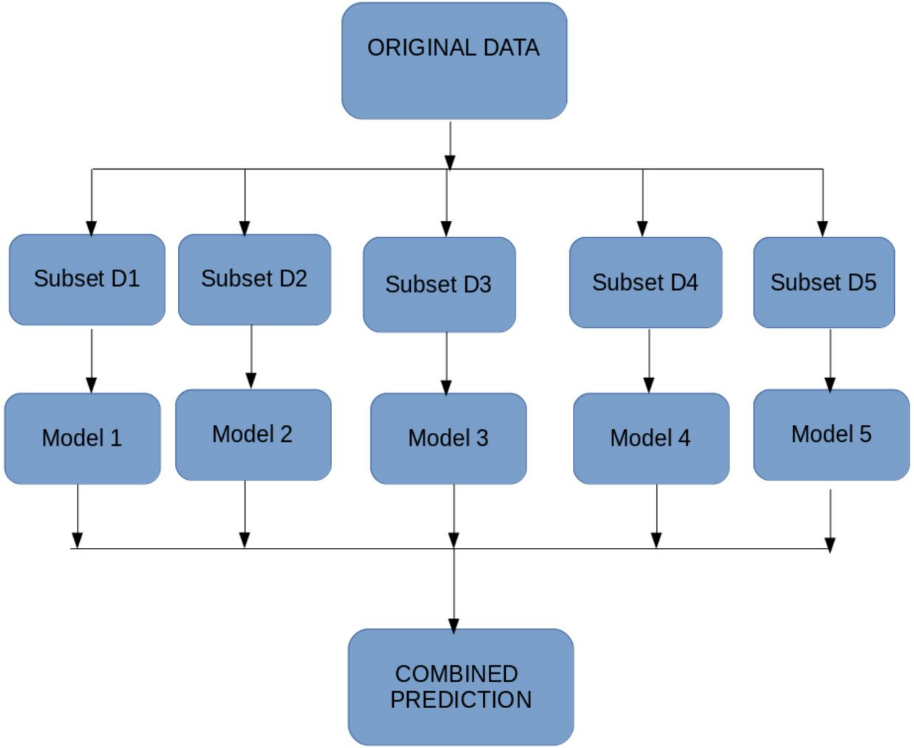
Decision Tree: Overfitting



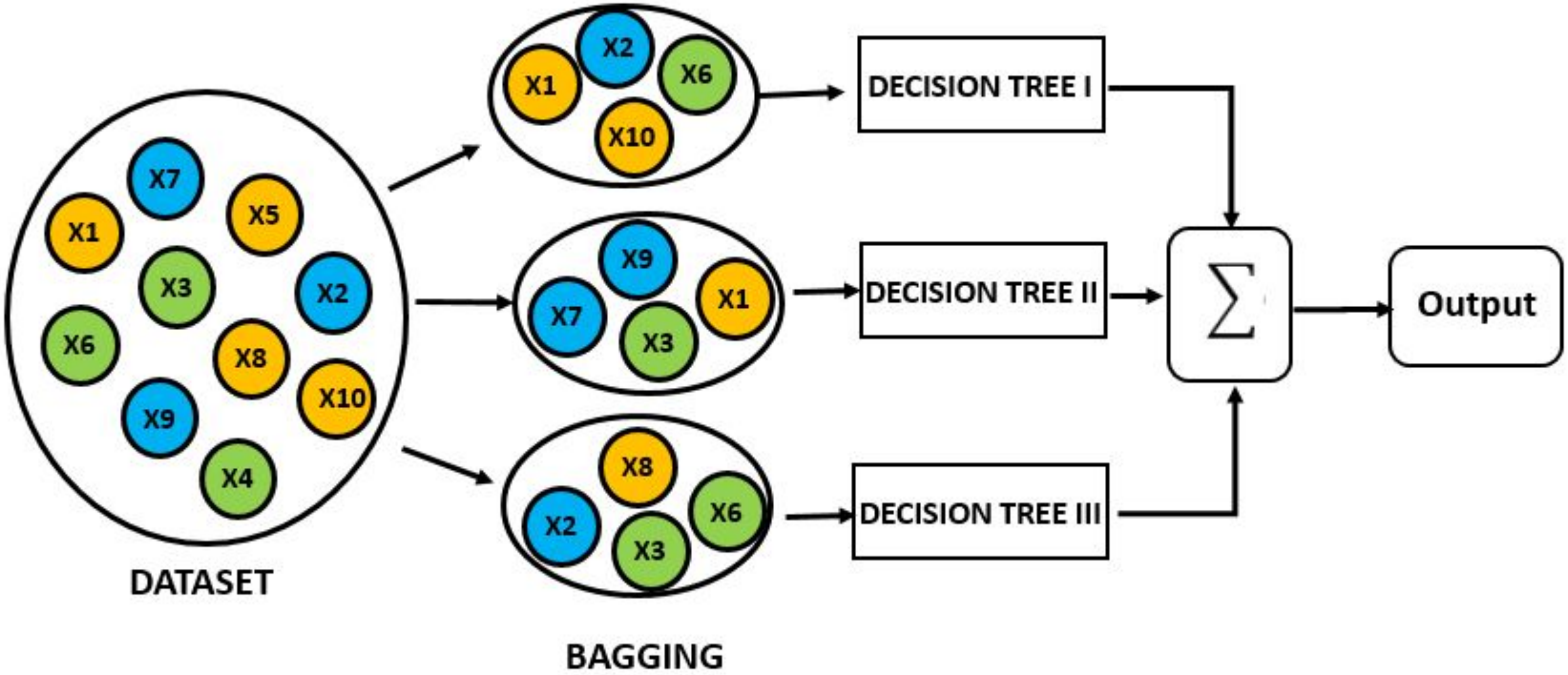
Decision Tree: Overfitting



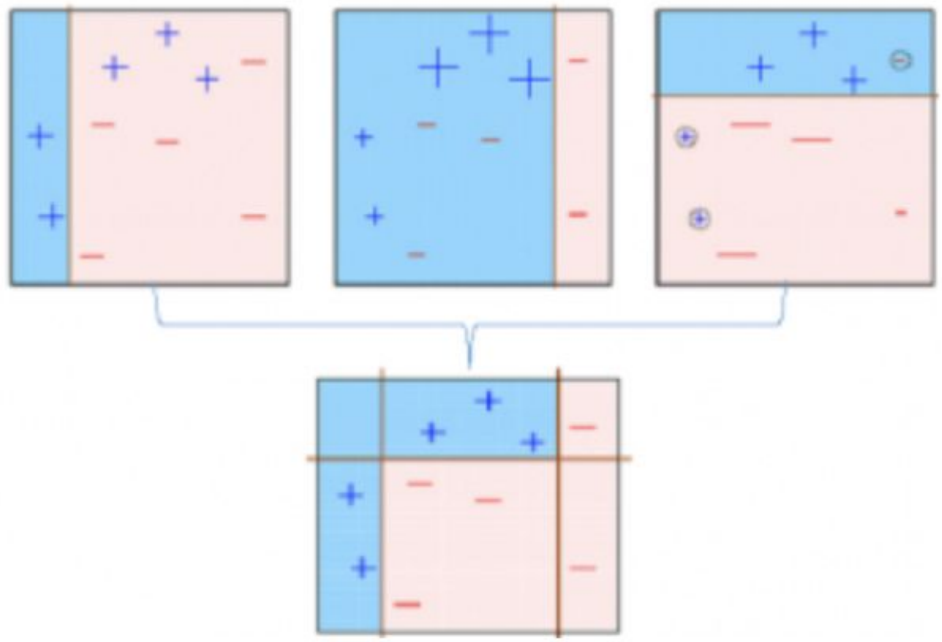
Decision Tree: Ensemble: Bagging



Decision Tree: Ensemble: Random Forests



Decision Tree: Ensemble: Boosting



Code Review