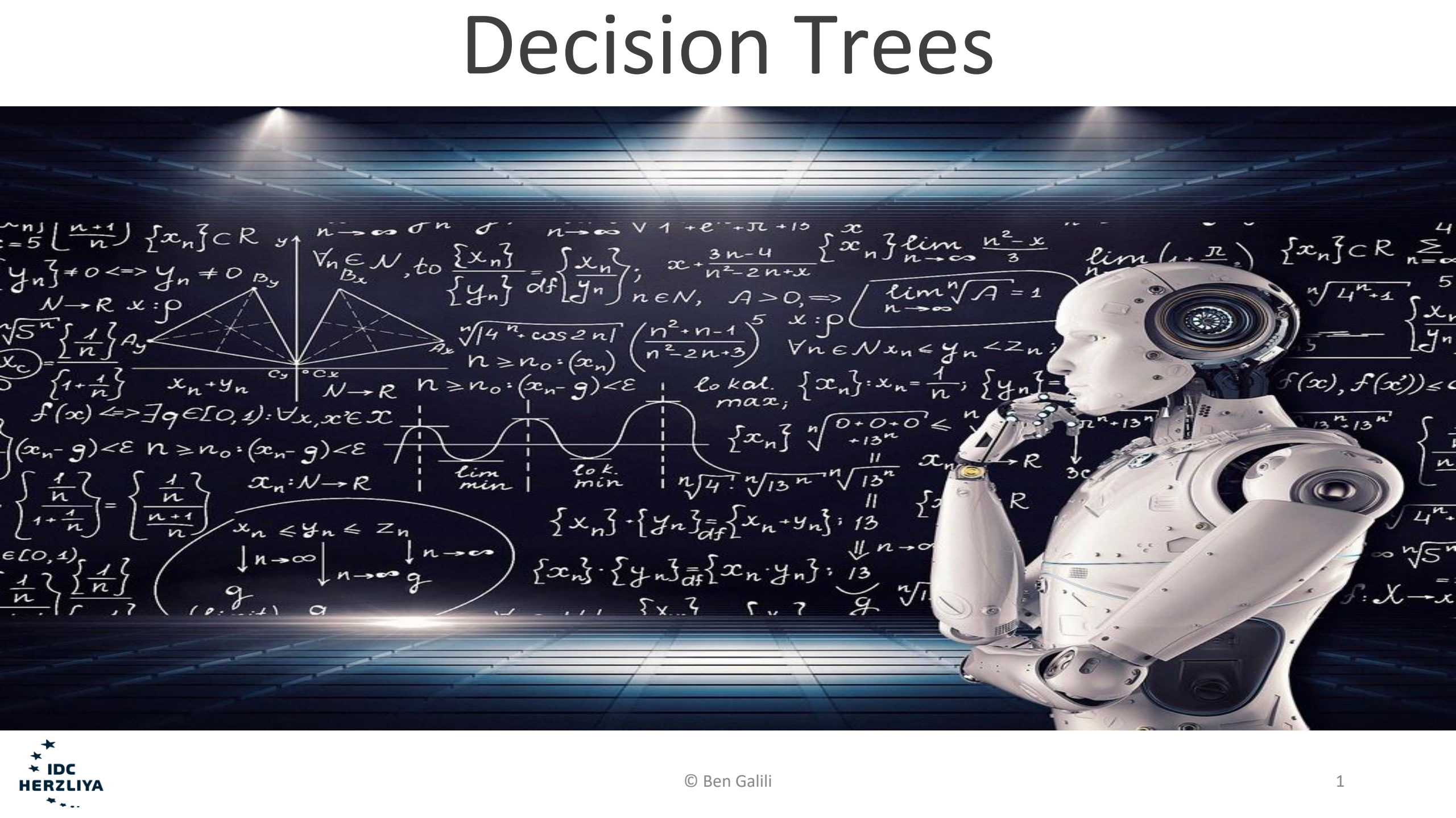


Decision Trees



The background of the slide is a dark blue, textured surface with a grid pattern. It is filled with various mathematical formulas and diagrams. On the left, there is a 3D coordinate system with axes labeled x , y , and z . The origin is labeled O . There are several points labeled A_x , A_y , A_z , B_x , B_y , B_z , C_x , C_y , and C_z . There are also several mathematical expressions, including $\lim_{n \rightarrow \infty} \frac{n^2 - x}{3}$, $\lim_{n \rightarrow \infty} \sqrt[n]{A} = 1$, and $\lim_{n \rightarrow \infty} \frac{1}{n} = 0$. The robot is a white and blue humanoid figure with a large circular eye and a small antenna. It is standing on the right side of the slide, looking towards the left. It is holding a small object in its right hand.



Previous recitation recap

- Gradient – the vector of the partial derivatives
- For some $f(x_1, x_2, \dots, x_n)$, the gradient will be:
$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$
- The gradient points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the graph in that direction

Previous recitation recap



- Gradient descent – going in the opposite direction to the gradient (toward the minimum)
- We need the learning rate, alpha, to determine how fast or slow we will move towards the minimum (optimal weights)



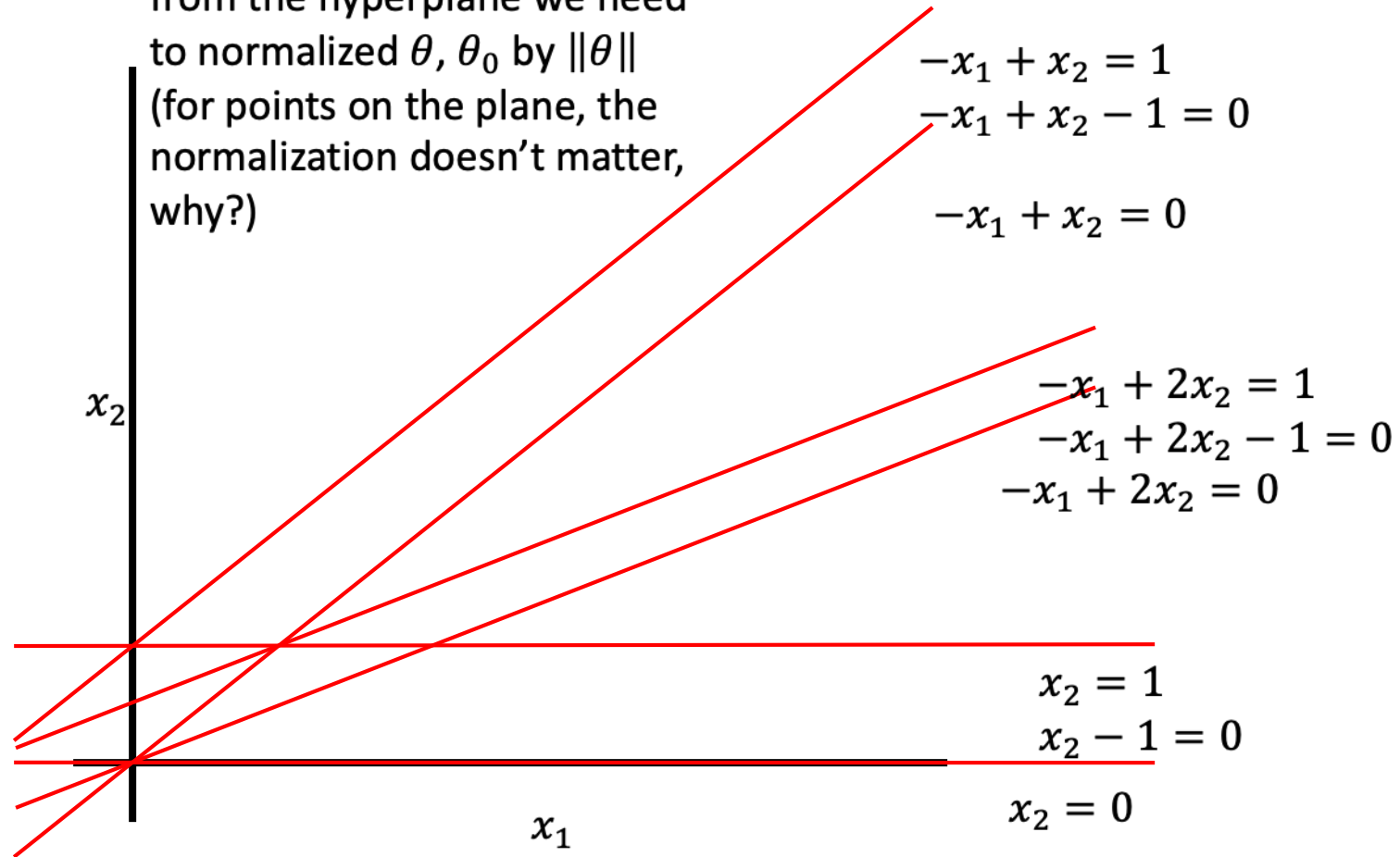
Some Algebra reminder

- How do we define a hyperplane in the space?
 - The hyperplane defined by the vector θ
 - All the point on the hyperplane solve the equation
$$\theta_1 x_1 + \dots + \theta_n x_n = b \quad (= \theta_0)$$
 - Where x are the point coordinates
 - The hyperplane separates the space into two half-spaces
 - All the point that the equation result $> b$
 - All the point that the equation result $< b$

Hyperplane - examples



In order to find the distance from the hyperplane we need to normalized θ , θ_0 by $\|\theta\|$
(for points on the plane, the normalization doesn't matter, why?)





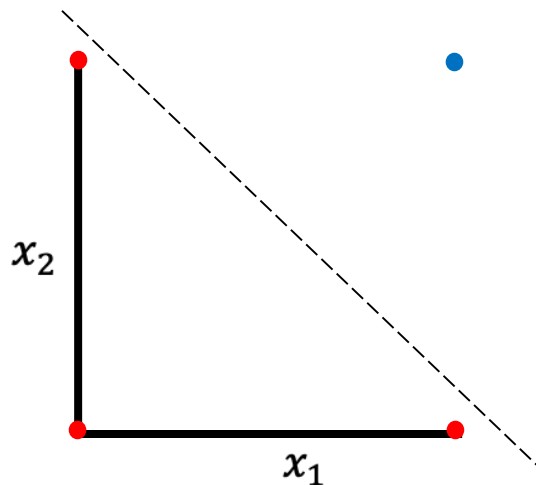
Linear separator

- We want to find linear separator:
 - All point above with result greater than 0, will be belong to the +1 class (or -1)
 - All point under with result lower than 0, will be belong to the -1 class (or +1)
- So, what do we need to find?
 - The hyperplane weights $\theta \in R^{n+1}$ (n hyperplane weights & the bias θ_0)
 - We will predict 1 if $\sum_{i=1}^n \theta_i x_i + \theta_0 > 0$ and -1 otherwise



Boolean functions – AND

- X_1 AND X_2

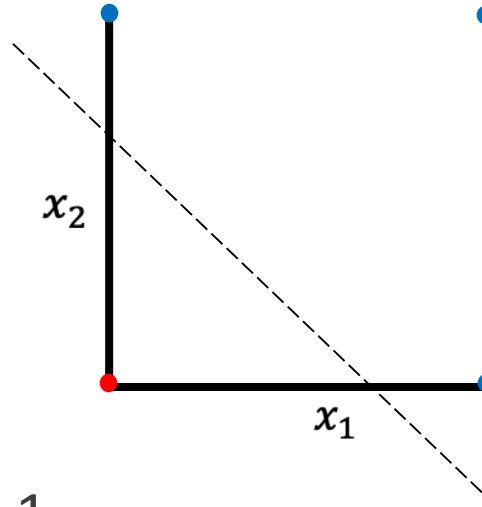


- Solution?
- If $1 \times X_1 + 1 \times X_2 - 1.5 > 0$ predict 1
- Otherwise predict -1.
- i.e. $\theta_0 = -1.5, \theta_1 = 1, \theta_2 = 1$

OR



- X_1 OR X_2

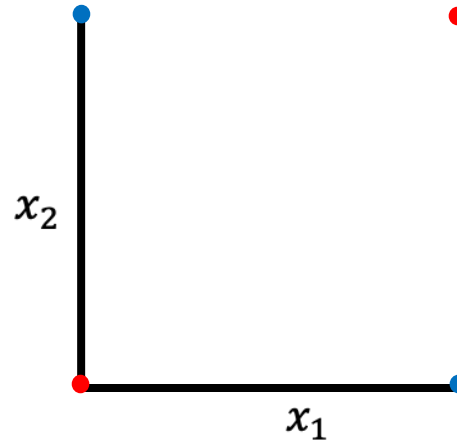


- Solution?
- $X_1 + X_2 - 0.5 > 0$ predict 1
- Otherwise -1

XOR



- $X_1 \text{ XOR } X_2$



- Solution?
- There is no solution
- Many functions cannot be represented using a linear separator, i.e., they are not linearly separable

Decision Trees

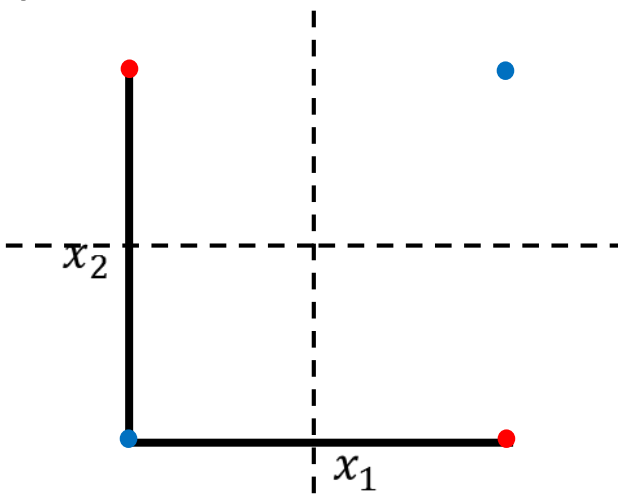


- We will talk about linear classifiers in future recitation
- The problem with linear classifiers is that not all the data is linear separable
- We need more 'tools' to deal with more complex data



Decision Trees

- Lets look on the classic XOR problem

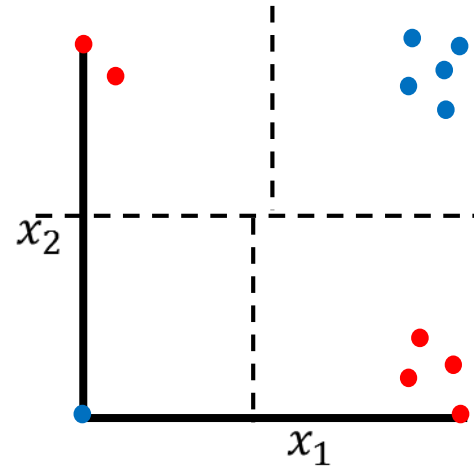


- There is no linear classifier in this dimension that can separate the classes
- How can we separate them?

Decision Trees



- By intuition, where would you put the first line (horizontal or vertical) and why?
- Now, we have 2 different parts, and the same question for each one of them
- This procedure produces a decision tree

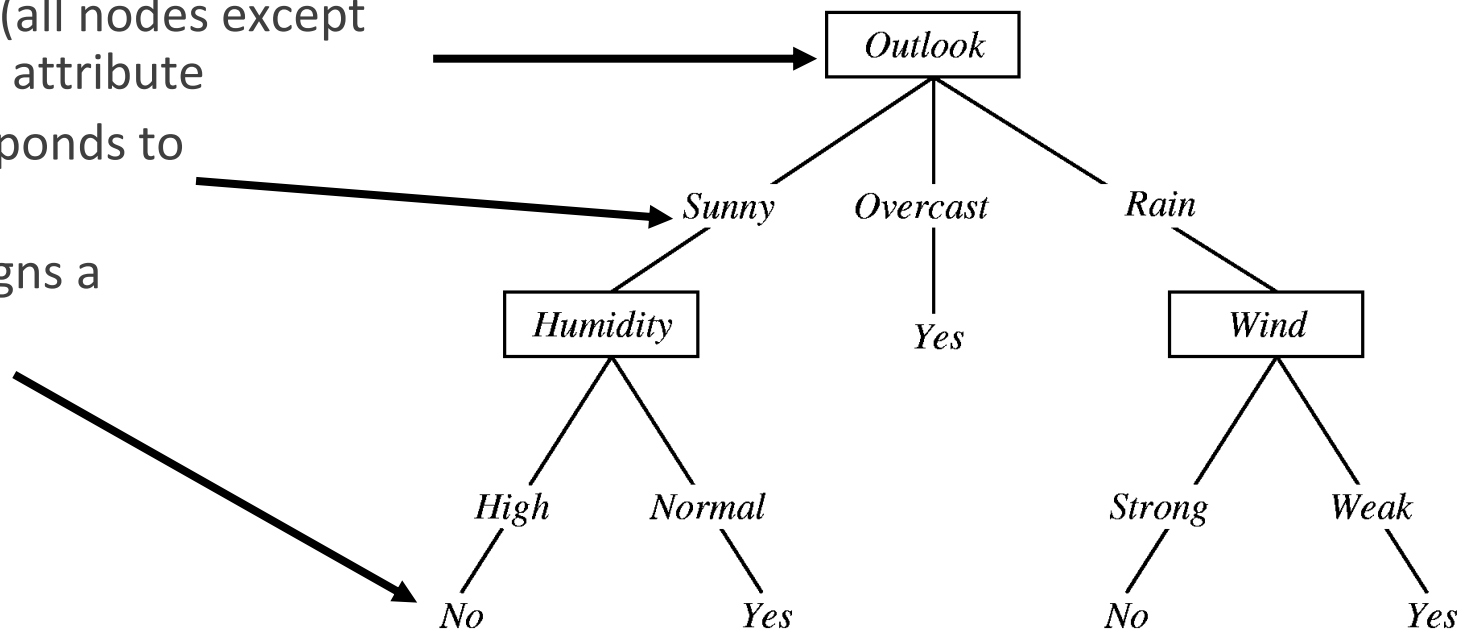




Decision Trees – more formal

- Decision Tree definitions:

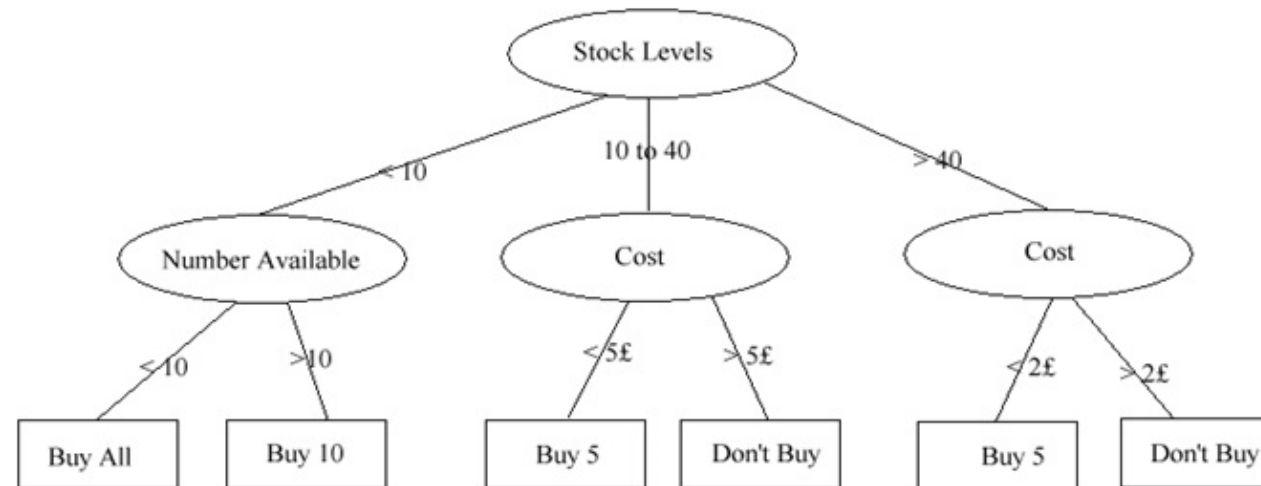
- Each internal node (all nodes except the leaves) tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification



Decision Trees



- For continuous variables we choose a threshold value to split the attribute
- For example:
 - We have a continuous attribute $x \in [0,100]$. If we are testing for this variable then we create a threshold value t and ask $x < t$ or $x \geq t$.





Algorithm for Building A Decision Tree

- **While** there are nodes in the queue **do**
 - Get next node n
 - **If** training examples in n perfectly classified
 - **Then** continue to next node
 - **else**
 - $A \leftarrow$ the “best” decision **attribute** for the set in n
 - Assign A as the decision attribute for n
 - **For Each** value of A
 - Create new descendant of n
 - Distribute training examples to descendant nodes
 - Insert descendent nodes to queue
- **End While**

← But, how can we know which attribute is the best?

Choosing the best attribute



- We want to choose the attribute that brings us closer to perfect classification
- In order to do that we need to measure how far are we from the perfect classification
- This measure called impurity:
 - High impurity means – we are far from perfect classification
 - Low impurity means – we are close to perfect classification
 - * Look in the lecture for formal definition

Choosing the best attribute



- How can we use impurity to choose the best attribute?
 - Calculate the impurity in the current node
 - Calculate a weighted average of the impurity over the children nodes after a split according to the test attribute
 - Reduce the second from the first and you will get the impurity reduce
 - Choose the attribute that cause the largest impurity reduce



Goodness of split

- The formula for the impurity reduce:

$$\Delta\varphi(S, A) = \varphi(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \varphi(S_v)$$

* Where φ is the impurity measure

- **Important fact:**

- The φ measure the impurity according to the classes distribution in a node
- The instances are split according to the test attribute values – A
- This means that you split the instances according to an attribute values and then calculate the impurity according to the classes values



Goodness of split

- There are 2 main implementation of the impurity criterion

	Gini	Entropy
Impurity	$GiniIndex(S) = 1 - \sum_{i=1}^c \left(\frac{ S_i }{ S } \right)^2$	$Entropy(S) = - \sum_{i=1}^c \frac{ S_i }{ S } \log \frac{ S_i }{ S }$
Goodness of split	$Gini_Gain = GiniIndex(S) - \sum_{v \in Values(A)} \frac{ S_v }{ S } GiniIndex(S_v)$	$Information_Gain = Entropy(S) - \sum_{v \in Values(A)} \frac{ S_v }{ S } Entropy(S_v)$



Goodness of split

- Uniform distribution:

- GiniIndex =

$$1 - \sum_{i=1}^c \left(\frac{|S_i|}{|S|} \right)^2 = 1 - c \left(\frac{1}{c} \right)^2 = 1 - \frac{1}{c}$$

- Entropy =

$$- \sum_{i=1}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} = - \sum_{i=1}^c \frac{1}{c} \log_2 \frac{1}{c} = -c \left(\frac{1}{c} \log_2 \frac{1}{c} \right) = -\log_2 \frac{1}{c}$$

- Perfect distribution (all instances have the same class):

- GiniIndex =

$$1 - \sum_{i=1}^c \left(\frac{|S_i|}{|S|} \right)^2 = 1 - \left(\frac{c}{c} \right)^2 = 1 - 1 = 0$$

- Entropy =

$$- \sum_{i=1}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} = -\frac{c}{c} \log_2 \frac{c}{c} = 0$$



Attribute with many values

- We want to predict if you will pass the test
- We have a training set of the last year students (100 students) with 5 attributes – ID, Gender, Bagrut Average, hours spend on study, luck (1-10)
- Which attribute the InformationGain will choose, and why?
- If an attribute has many values, the InformationGain will tend to select it

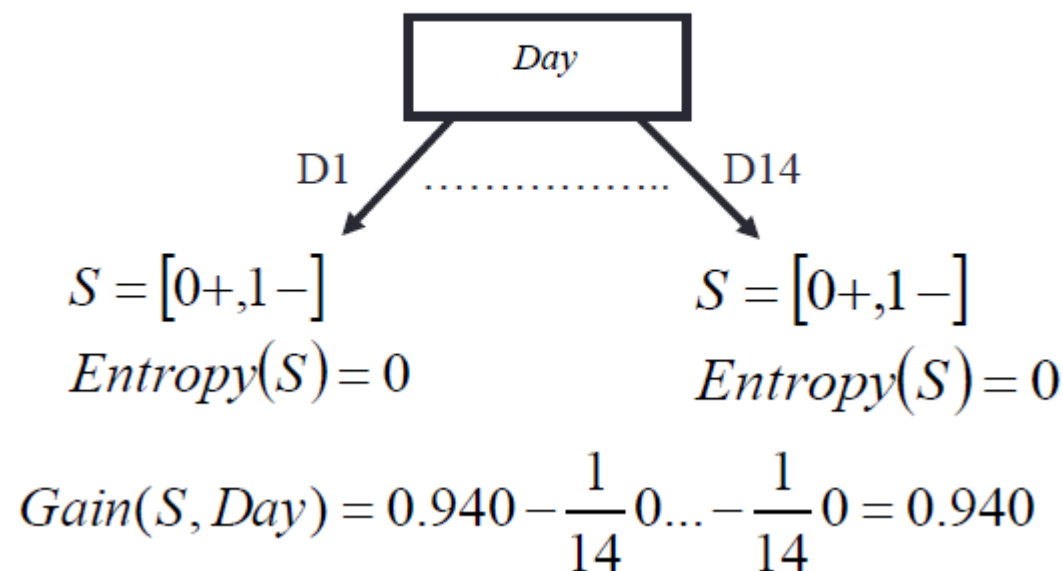


Attribute with many values

- Imagine using the attribute $DAY=[D1,...,D14]$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$S = [9+, 5-]$$
$$Entropy(S) = 0.940$$





Attribute with many values

- To solve that we can use GainRatio:

$$\text{GainRatio}(S, A) = \frac{\text{InformationGain}(S, A)}{\text{SplitInformation}(S, A)}$$

- Where $\text{SplitInformation}(S, A)$ is the Entropy with respect to the attribute A

$$\text{SplitInformation}(S, A) = - \sum_{a \in A} \frac{|S_a|}{|S|} \log \frac{|S_a|}{|S|}$$

* In contrast to what we used as Entropy of S, which was with respect to the target class

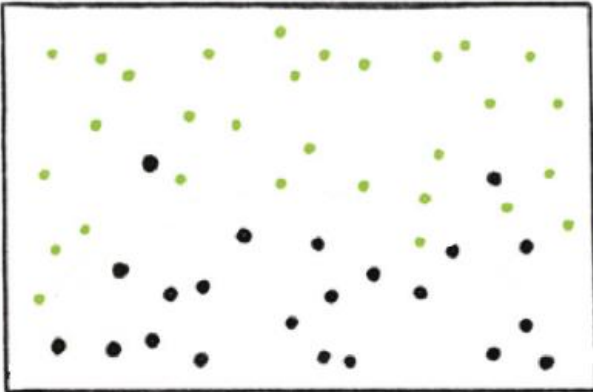
- Example:

$$\begin{aligned} \text{SplitInformation}(S, \text{Day}) &= - \sum_{i=1}^{14} \frac{1}{14} \log \frac{1}{14} = -\log \frac{1}{14} = 3.8074 \\ \text{GainRatio}(S, A) &= \frac{0.94}{3.8074} = 0.2469 \end{aligned}$$

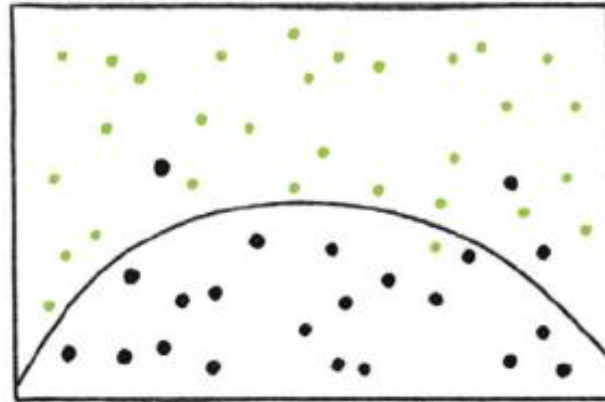
Overfitting



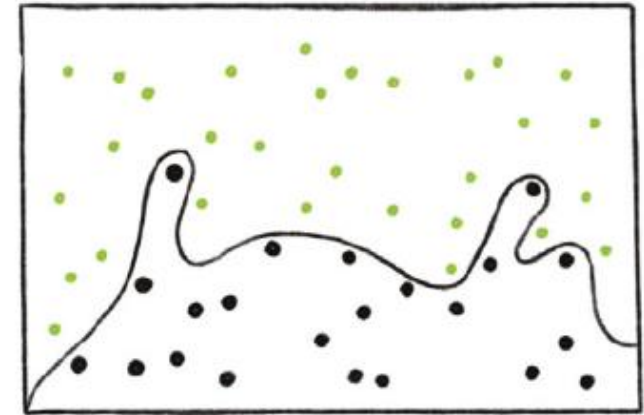
Dataset



Good Fit



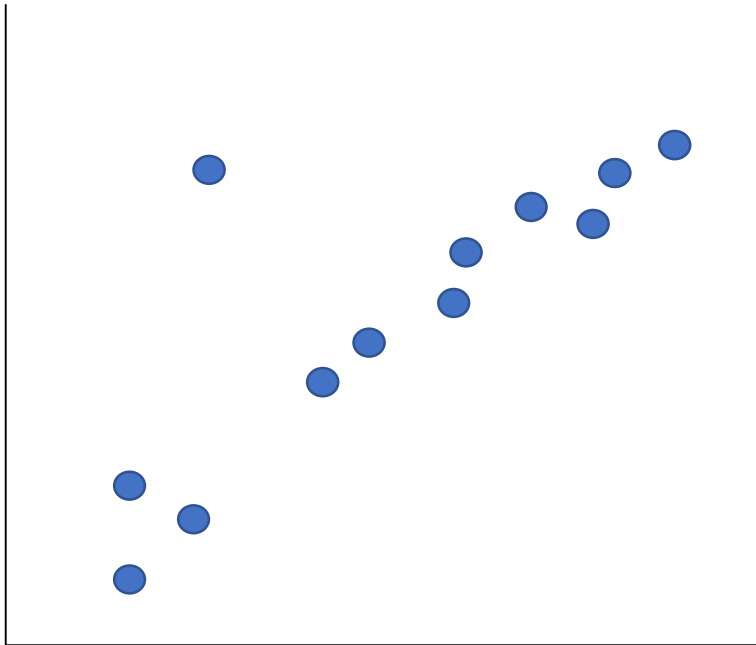
Overfit



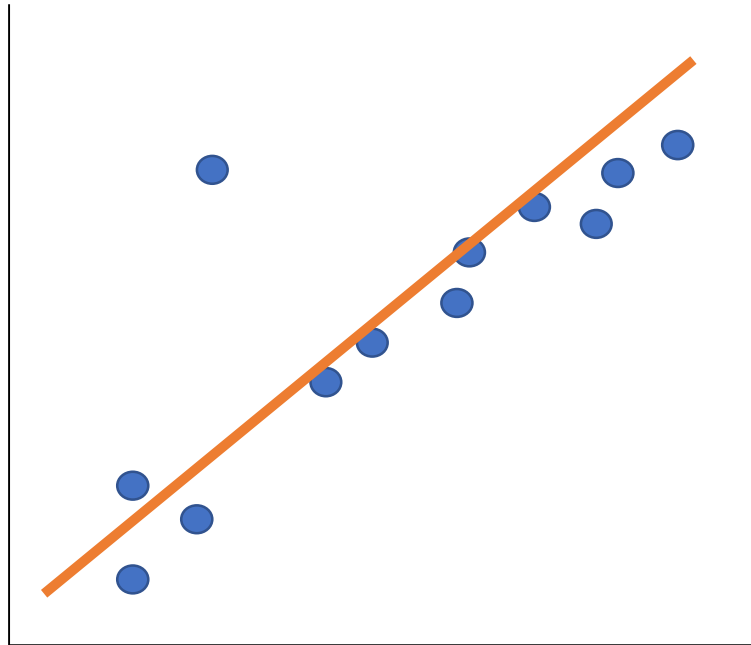
Overfitting - Regression



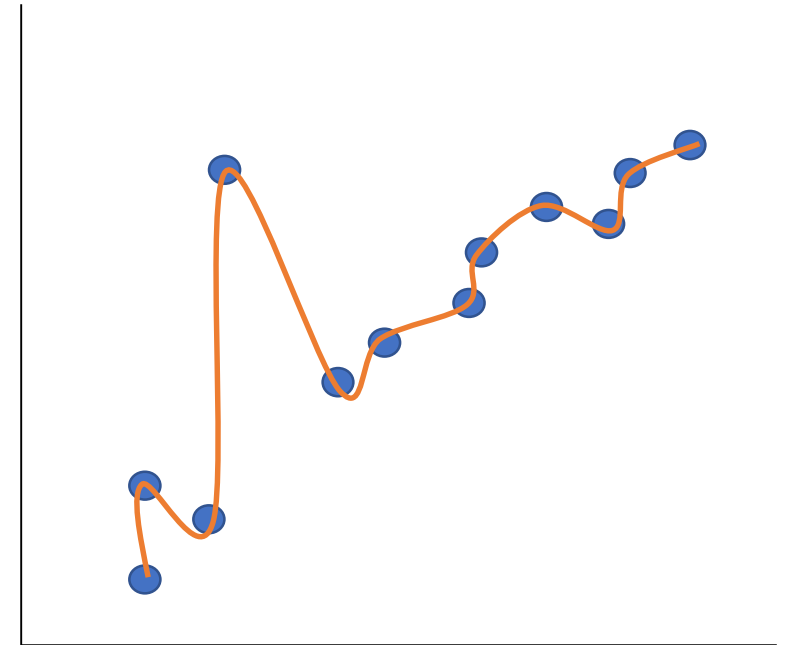
Dataset



Good Fit



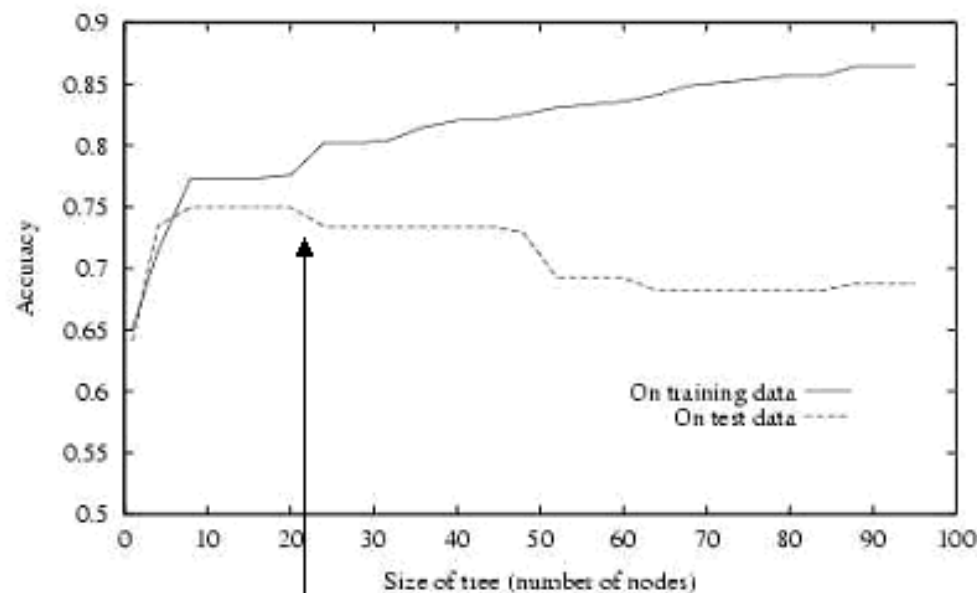
Overfit





Overfitting

- Decision trees tend to overfit
- This means that our tree gets too specific in handling the training data



Overfitting starting here

Pruning



- This suggests that we should prune or cut off some branches of the tree to make it smaller, and therefore get better test error
- How do we do this?

Pruning



- Post pruning
 - when we traverse the tree (starting from the leaf nodes), go all the way up (finishing at the root) and at each node we decide whether or not to completely cut off that branch of the tree
 - We decide whether to cut the branch or not based on whether the splitting attribute helps or not
- We can also prune during the tree building = we won't create the children nodes

Chi Square Test



- The Chi Square test is supposed to tell us: does splitting according to some attribute give us a distribution which is completely random or does it have some predicting power?
- So we check if splitting according to the chosen attribute gives a distribution which is similar to exactly random

Chi Square Test



- What is exactly random?
- If we have in our data $Y=0$ for 10 times and $Y=1$ for 90 times then the probability $Y=0$ is 10% and probability $Y=1$ is 90%
- If we split according to x_j and there are 50 instances where $x_j = 1$, then if splitting according to x_j was completely random we expect to see $50 \cdot 0.1$ instances where $Y=0$ and $50 \cdot 0.9$ instances where $Y=1$. If we don't see this, then splitting according to x_j is not a random distribution and has predictive power.



Chi Square Test

- The test itself (assume Y can only take values of $0 \setminus 1$):

- $P(Y = 0) \approx \frac{\# Y=0 \text{ instances}}{\# \text{Instances}}$

- Call D_f = number of instances where $x_j = f$

- p_f = number of instances where $x_j = f$ & $Y = 0$

- n_f = number of instances where $x_j = f$ & $Y = 1$

- $E_0 = D_f * P(Y = 0)$, $E_1 = D_f * P(Y = 1)$

- So Chi Square statistic is:

$$\chi^2 = \sum_{f \in \text{values}(x_j)} \frac{(p_f - E_0)^2}{E_0} + \frac{(n_f - E_1)^2}{E_1}$$



Checking if significant

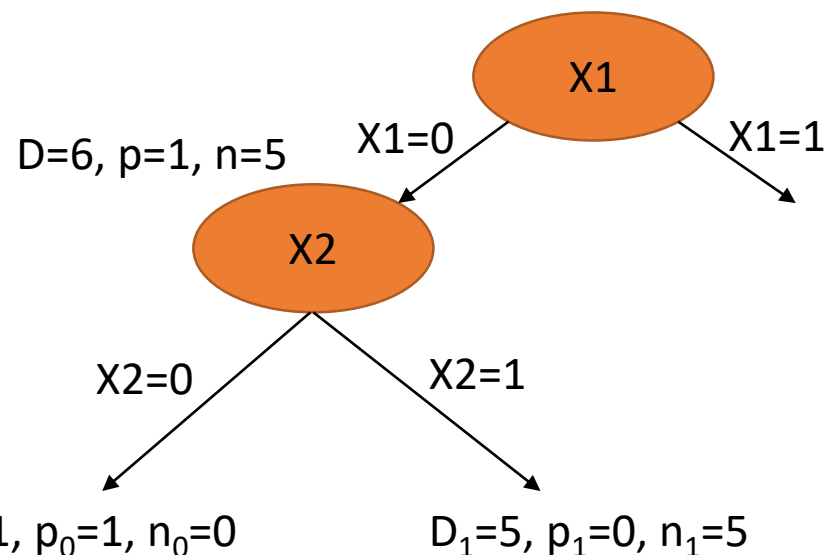
- Once we have the Chi Square statistic we use a chart to check if it is significant or not

Table of Probabilities for the Chi-Squared Distribution														
Alpha Risk														
df	0.995	0.990	0.975	0.95	0.9	0.75	0.5	0.25	0.1	0.05	0.25	0.01	0.005	0.001
1	0.000039	0.000157	0.000982	0.00393	0.0158	0.102	0.455	1.323	2.706	3.841	1.323	6.635	7.879	10.828
2	0.010	0.020	0.051	0.103	0.211	0.575	1.386	2.773	4.605	5.991	2.773	9.210	10.597	13.816
3	0.072	0.115	0.216	0.352	0.584	1.213	2.366	4.108	6.251	7.815	4.108	11.345	12.838	16.266
4	0.207	0.297	0.484	0.711	1.064	1.923	3.357	5.385	7.779	9.488	5.385	13.277	14.860	18.467
5	0.412	0.554	0.831	1.145	1.610	2.675	4.351	6.626	9.236	11.070	6.626	15.086	16.750	20.515
6	0.676	0.872	1.237	1.635	2.204	3.455	5.348	7.841	10.645	12.592	7.841	16.812	18.548	22.458
7	0.989	1.239	1.690	2.167	2.833	4.255	6.346	9.037	12.017	14.067	9.037	18.475	20.278	24.322
8	1.344	1.646	2.180	2.733	3.490	5.071	7.344	10.219	13.362	15.507	10.219	20.090	21.955	26.124
9	1.735	2.088	2.700	3.325	4.168	5.899	8.343	11.389	14.684	16.919	11.389	21.666	23.589	27.877
10	2.156	2.558	3.247	3.940	4.865	6.737	9.342	12.549	15.987	18.307	12.549	23.209	25.188	29.588
11	2.603	3.053	3.816	4.575	5.578	7.584	10.341	13.701	17.275	19.675	13.701	24.725	26.757	31.264
12	3.074	3.571	4.404	5.226	6.304	8.438	11.340	14.845	18.549	21.026	14.845	26.217	28.300	32.909
13	3.565	4.107	5.009	5.892	7.042	9.299	12.340	15.984	19.812	22.362	15.984	27.688	29.819	34.528
14	4.075	4.660	5.629	6.571	7.790	10.165	13.339	17.117	21.064	23.685	17.117	29.141	31.319	36.123
15	4.601	5.229	6.262	7.261	8.547	11.037	14.339	18.245	22.307	24.996	18.245	30.578	32.801	37.697
16	5.142	5.812	6.908	7.962	9.312	11.912	15.338	19.369	23.542	26.296	19.369	32.000	34.267	39.252
17	5.697	6.408	7.564	8.672	10.085	12.792	16.338	20.489	24.769	27.587	20.489	33.409	35.718	40.790
18	6.265	7.015	8.231	9.390	10.865	13.675	17.338	21.605	25.989	28.869	21.605	34.805	37.156	42.312
19	6.844	7.633	8.907	10.117	11.651	14.562	18.338	22.718	27.204	30.144	22.718	36.191	38.582	43.820
20	7.434	8.260	9.591	10.851	12.443	15.452	19.337	23.828	28.412	31.410	23.828	37.566	39.997	45.315
21	8.034	8.897	10.283	11.591	13.240	16.344	20.337	24.935	29.615	32.671	24.935	38.932	41.401	46.797
22	8.643	9.542	10.982	12.338	14.041	17.240	21.337	26.039	30.813	33.924	26.039	40.289	42.796	48.268
23	9.260	10.196	11.689	13.091	14.848	18.137	22.337	27.141	32.007	35.172	27.141	41.638	44.181	49.728
24	9.886	10.856	12.401	13.848	15.659	19.037	23.337	28.241	33.196	36.415	28.241	42.980	45.559	51.179
25	10.520	11.524	13.120	14.611	16.473	19.939	24.337	29.339	34.382	37.652	29.339	44.314	46.928	52.620
26	11.160	12.198	13.844	15.379	17.292	20.843	25.336	30.435	35.563	38.885	30.435	45.642	48.290	54.052
27	11.808	12.879	14.573	16.151	18.114	21.749	26.336	31.528	36.741	40.113	31.528	46.963	49.645	55.476
28	12.461	13.565	15.308	16.928	18.939	22.657	27.336	32.620	37.916	41.337	32.620	48.278	50.993	56.892
29	13.121	14.256	16.047	17.708	19.768	23.567	28.336	33.711	39.087	42.557	33.711	49.588	52.336	58.301
30	13.787	14.953	16.791	18.493	20.599	24.478	29.336	34.800	40.256	43.773	34.800	50.892	53.672	59.703
40	20.707	22.164	24.433	26.509	29.051	33.660	39.335	45.616	51.805	55.758	45.616	63.691	66.766	73.402
50	27.991	29.707	32.357	34.764	37.689	42.942	49.335	56.334	63.167	67.505	56.334	76.154	79.490	86.661
60	35.534	37.485	40.482	43.188	46.459	52.294	59.335	66.981	74.397	79.082	66.981	88.379	91.952	99.607
70	43.275	45.442	48.758	51.739	55.329	61.698	69.334	77.577	85.527	90.531	77.577	100.425	104.215	112.317
80	51.172	53.540	57.153	60.391	64.278	71.145	79.334	88.130	96.578	101.879	88.130	112.329	116.321	124.839
90	59.196	61.754	65.647	69.126	73.291	80.625	89.334	98.650	107.565	113.145	98.650	124.116	128.299	137.208
100	67.328	70.065	74.222	77.929	82.358	90.133	99.334	109.141	118.498	124.342	109.141	135.807	140.169	149.449



Chi Square Example

Count	Y	X2	X1
2	+	1	1
2	+	0	1
5	-	1	0
1	+	0	0



Where X2=0

Where X2=1

$$\chi^2 = \sum_{f \in \text{values}(x_j)} \frac{(p_f - E_0)^2}{E_0} + \frac{(n_f - E_1)^2}{E_1} =$$

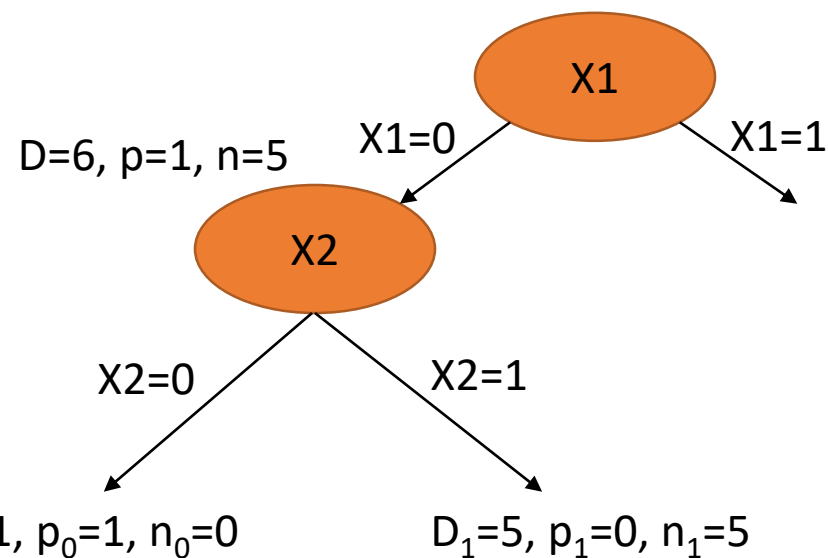
$$= \left(\frac{(p_0 - E_0)^2}{E_0} + \frac{(n_0 - E_1)^2}{E_1} \right) + \left(\frac{(p_1 - E_0)^2}{E_0} + \frac{(n_1 - E_1)^2}{E_1} \right)$$

$$= \left(\frac{\left(1 - \frac{1}{6}\right)^2}{\frac{1}{6}} + \frac{\left(0 - \frac{5}{6}\right)^2}{\frac{5}{6}} \right) + \left(\frac{\left(0 - \frac{5}{6}\right)^2}{\frac{5}{6}} + \frac{\left(5 - \frac{25}{6}\right)^2}{\frac{25}{6}} \right)$$



Chi Square Example

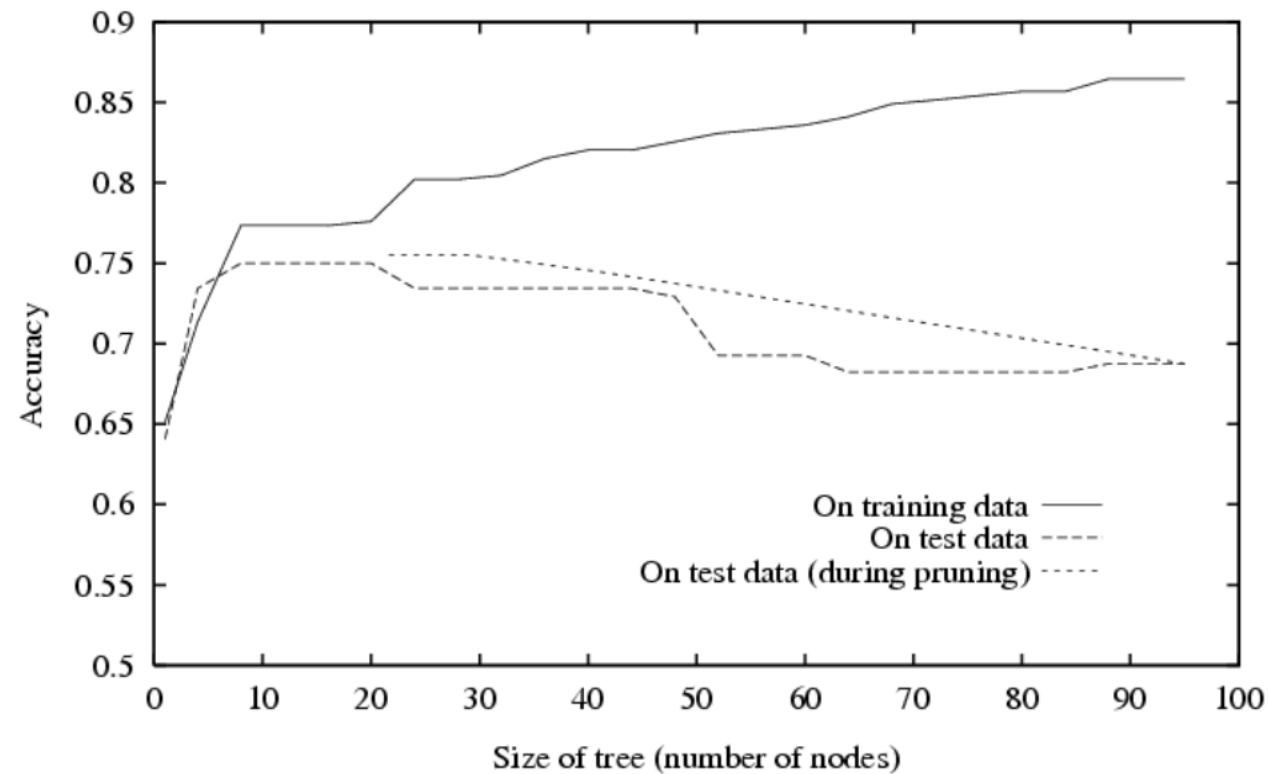
Count	Y	X2	X1
2	+	1	1
2	+	0	1
5	-	1	0
1	+	0	0



$$\chi^2 = \left(\frac{\left(1 - \frac{1}{6}\right)^2}{\frac{1}{6}} + \frac{\left(0 - \frac{5}{6}\right)^2}{\frac{5}{6}} \right) + \left(\frac{\left(0 - \frac{5}{6}\right)^2}{\frac{5}{6}} + \frac{\left(5 - \frac{25}{6}\right)^2}{\frac{25}{6}} \right) = \frac{25}{6} + \frac{5}{6} + \frac{5}{6} + \frac{1}{6} = 6$$

Now, we need to look at the chi square chart for the appropriate degree of freedom (number of attribute values – 1 in the 2 classes case) with 95% of confidence or 0.05 p-value

Effect of Tree Pruning





Example

- What calculations are needed to find the feature to split the root of the decision tree using Information Gain

- Reminder:

$$Information_Gain = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Entropy(S) = - \sum_{i=1}^c \frac{|S_i|}{|S|} \log \frac{|S_i|}{|S|}$$

- c – number of classes
- Values(A) – all the values in the A feature
- We need to calculate:
 - Entropy(root)
 - Weighted average of the Entropy according to “Attraction”
 - Weighted average of the Entropy according to “Weather”

Classification	Weather	Attraction	Instance
-	Hot	Swim	1
+	Hot	Dance	2
+	Hot	Casino	3
-	Hot	Golf	4
-	Mild	Swim	5
-	Mild	Casino	6
+	Mild	Dance	7
-	Mild	Golf	8
+	Mild	Ski	9
+	Cold	Ski	10
-	Cold	Casino	11
-	Cold	Dance	12



Example

- Entropy(root)

$$Entropy(root) = -\left(\frac{7}{12}\log\frac{7}{12} + \frac{5}{12}\log\frac{5}{12}\right)$$

- Weighted average of the Entropy according to “Attraction”

$$\begin{aligned} & \sum_{v \in \text{Values(Attraction)}} \frac{|S_v|}{|S|} Entropy(S_v) \\ &= -\left(\frac{2}{12}\left(\frac{2}{2}\log\frac{2}{2}\right) + \frac{3}{12}\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right) + \frac{3}{12}\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right) + \frac{2}{12}\left(\frac{2}{2}\log\frac{2}{2}\right) + \frac{2}{12}\left(\frac{2}{2}\log\frac{2}{2}\right)\right) \end{aligned}$$

- Weighted average of the Entropy according to “Weather”

$$\begin{aligned} & \sum_{v \in \text{Values(Weather)}} \frac{|S_v|}{|S|} Entropy(S_v) \\ &= -\left(\frac{4}{12}\left(\frac{2}{4}\log\frac{2}{4} + \frac{2}{4}\log\frac{2}{4}\right) + \frac{5}{12}\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) + \frac{3}{12}\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right)\right) \end{aligned}$$

Classification	Weather	Attraction	Instance
-	Hot	Swim	1
+	Hot	Dance	2
+	Hot	Casino	3
-	Hot	Golf	4
-	Mild	Swim	5
-	Mild	Casino	6
+	Mild	Dance	7
-	Mild	Golf	8
+	Mild	Ski	9
+	Cold	Ski	10
-	Cold	Casino	11
-	Cold	Dance	12



Example

Put it all together in the Information Gain formula

$$\begin{aligned} & \text{Information Gain}(\text{root}, \text{Attraction}) \\ &= - \left(\frac{7}{12} \log \frac{7}{12} + \frac{5}{12} \log \frac{5}{12} \right) \\ &+ \left(\frac{2}{12} \left(\frac{2}{2} \log \frac{2}{2} \right) + \frac{3}{12} \left(\frac{2}{3} \log \frac{2}{3} + \frac{1}{3} \log \frac{1}{3} \right) + \frac{3}{12} \left(\frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) \right. \\ &\left. + \frac{2}{12} \left(\frac{2}{2} \log \frac{2}{2} \right) + \frac{2}{12} \left(\frac{2}{2} \log \frac{2}{2} \right) \right) = 0.521 \end{aligned}$$

$$\begin{aligned} & \text{Information Gain}(\text{root}, \text{Weather}) \\ &= - \left(\frac{7}{12} \log \frac{7}{12} + \frac{5}{12} \log \frac{5}{12} \right) \\ &+ \left(\frac{4}{12} \left(\frac{2}{4} \log \frac{2}{4} + \frac{2}{4} \log \frac{2}{4} \right) + \frac{5}{12} \left(\frac{2}{5} \log \frac{2}{5} + \frac{3}{5} \log \frac{3}{5} \right) \right. \\ &\left. + \frac{3}{12} \left(\frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) \right) = 0.012 \end{aligned}$$

Classification	Weather	Attraction	Instance
-	Hot	Swim	1
+	Hot	Dance	2
+	Hot	Casino	3
-	Hot	Golf	4
-	Mild	Swim	5
-	Mild	Casino	6
+	Mild	Dance	7
-	Mild	Golf	8
+	Mild	Ski	9
+	Cold	Ski	10
-	Cold	Casino	11
-	Cold	Dance	12