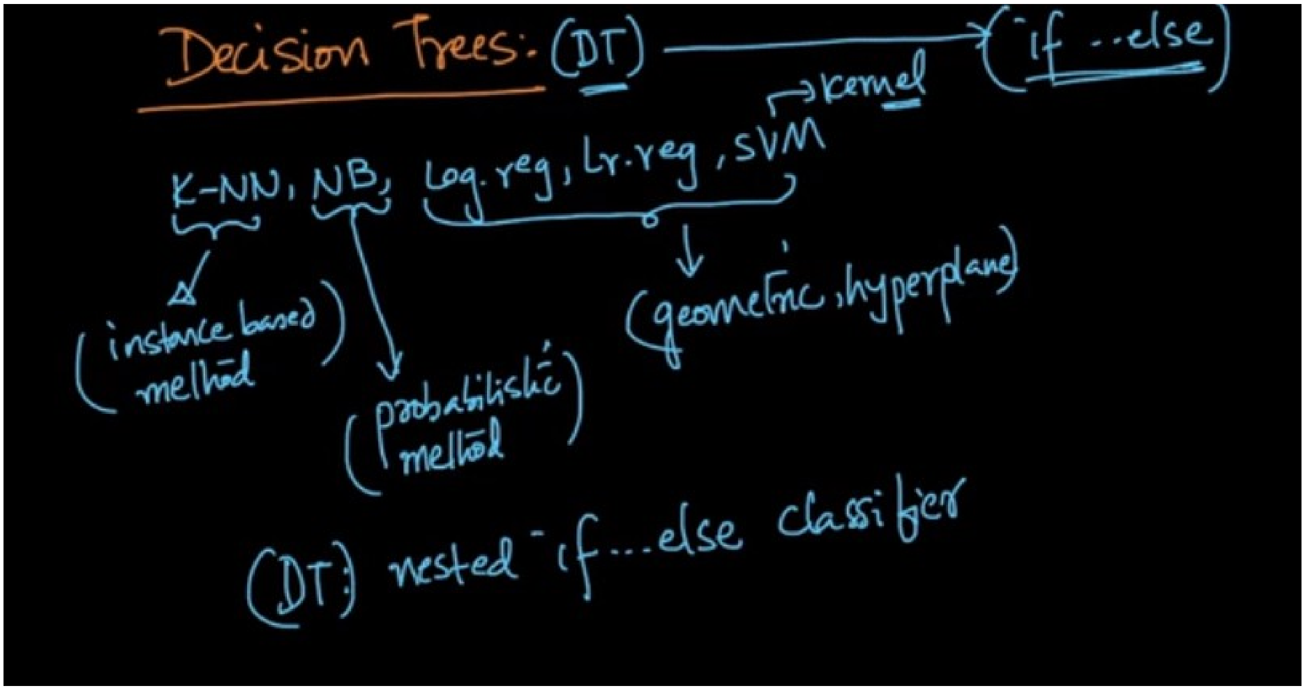
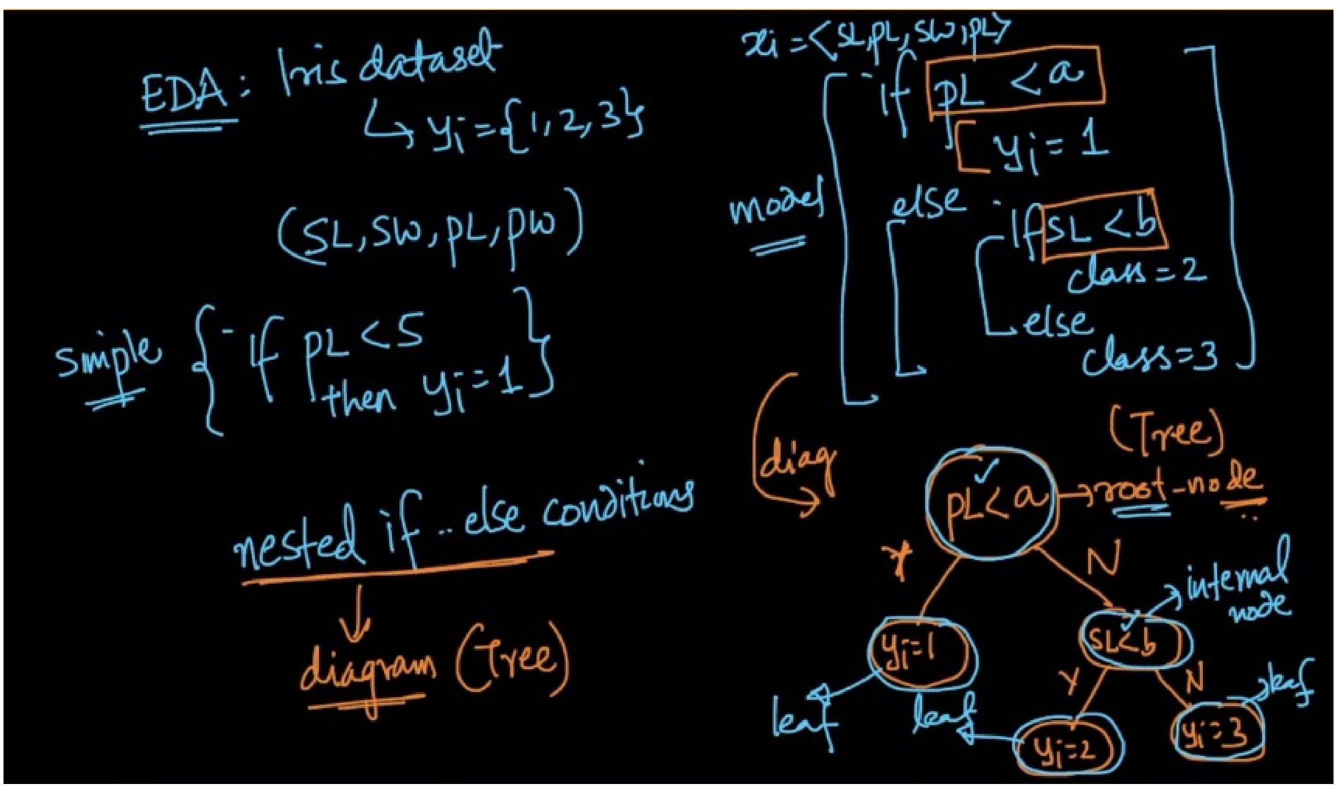
NOTES

Different classifiers and there properties.

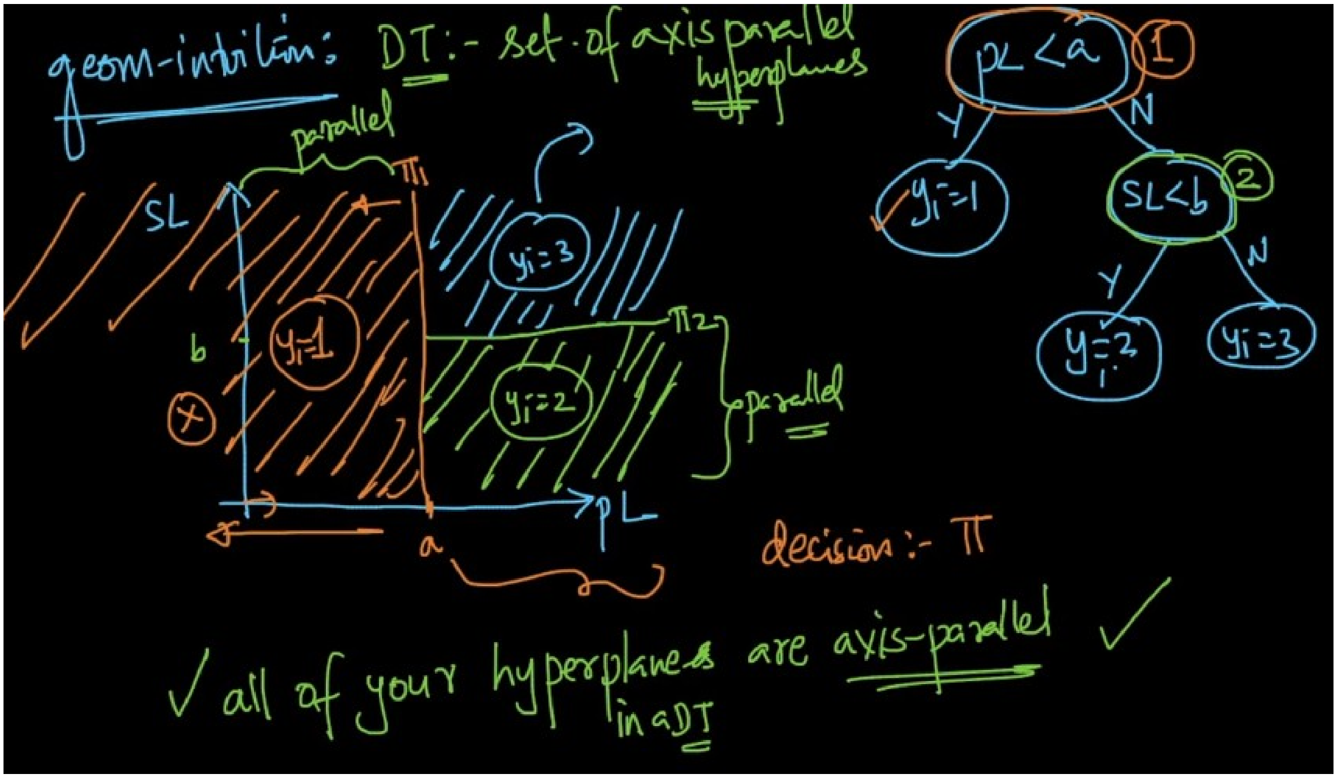


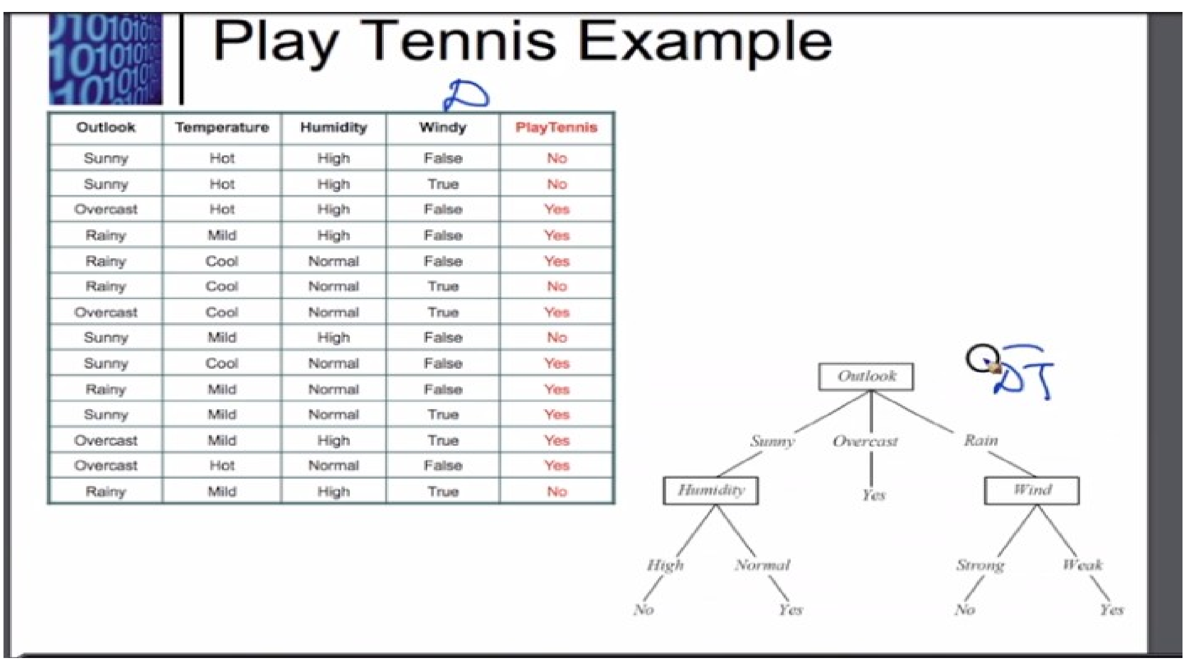
Nested if – else:
Decision tree starts with a node with conditions and divided into several nodes, this diagram is called a tree.



Root – node: This is the starting node.
Leaf – node: These are branches of the root node.
Internal – nodes: These are neither leaf nodes or root nodes.
Non – leaf nodes: These are nodes that make decisions, these are called the nodes of decision tree.

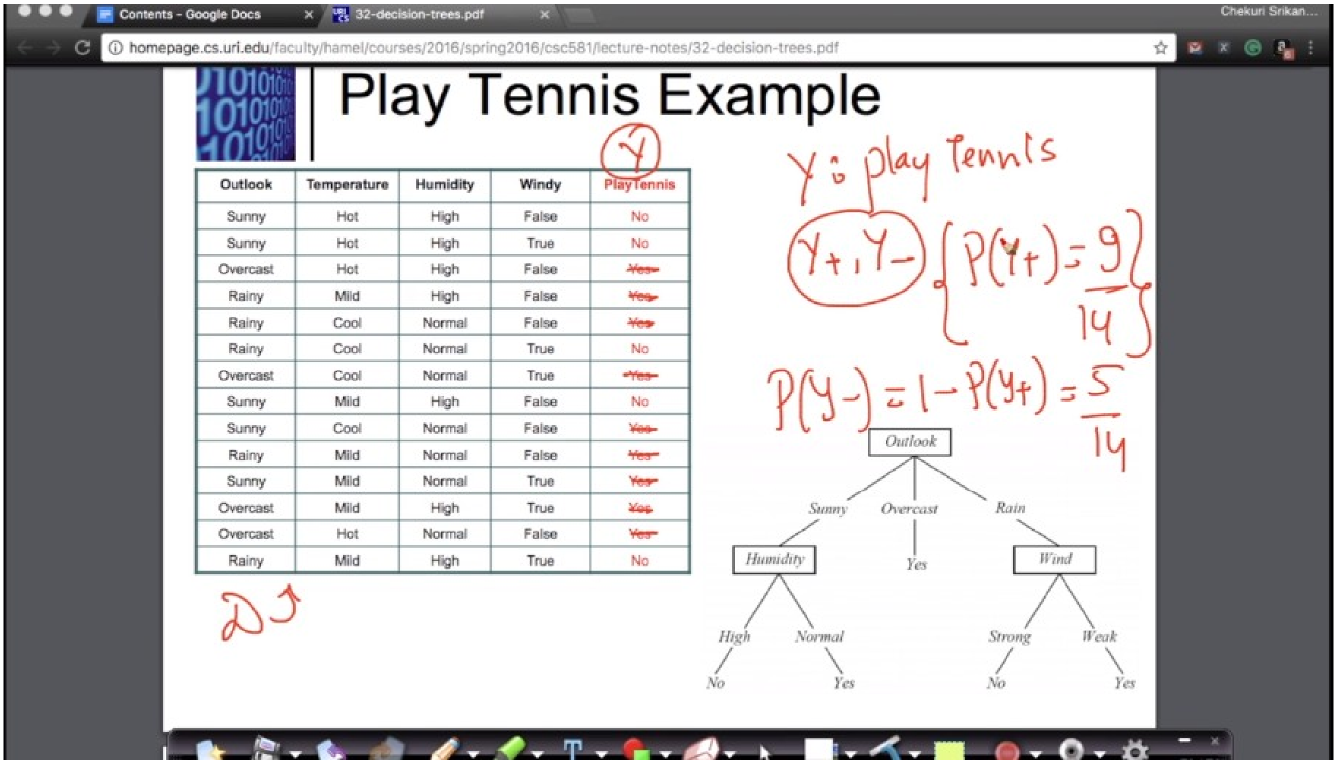
Geometry:
In decision tree all of the hyper planes are axis parallel.



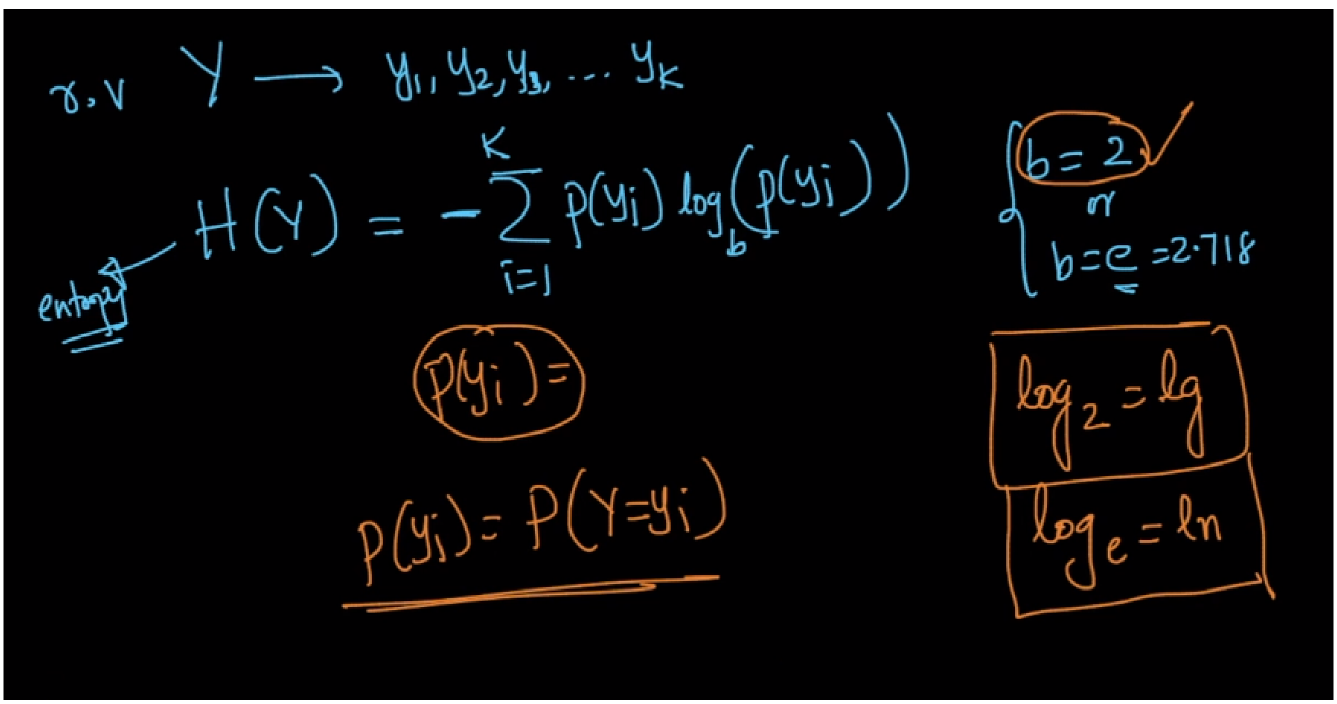


Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

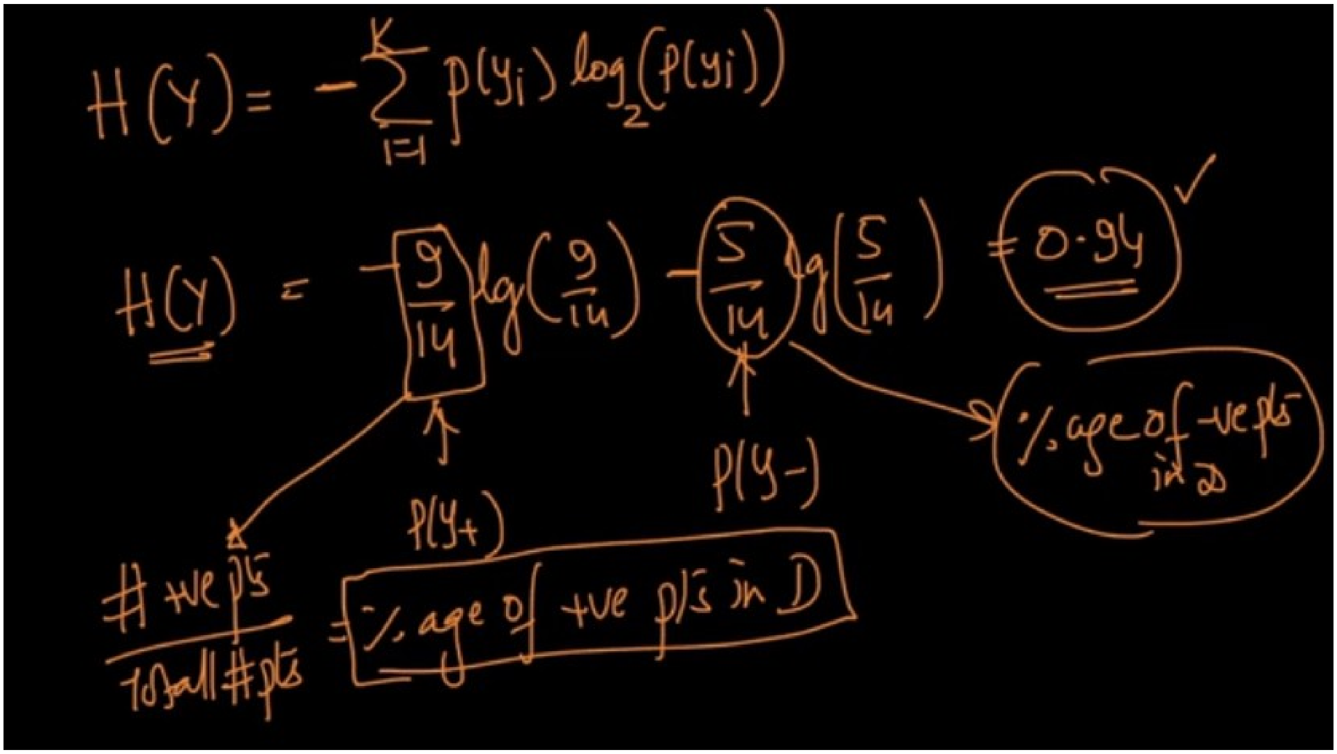
Building a Decision Tree:
Entropy:



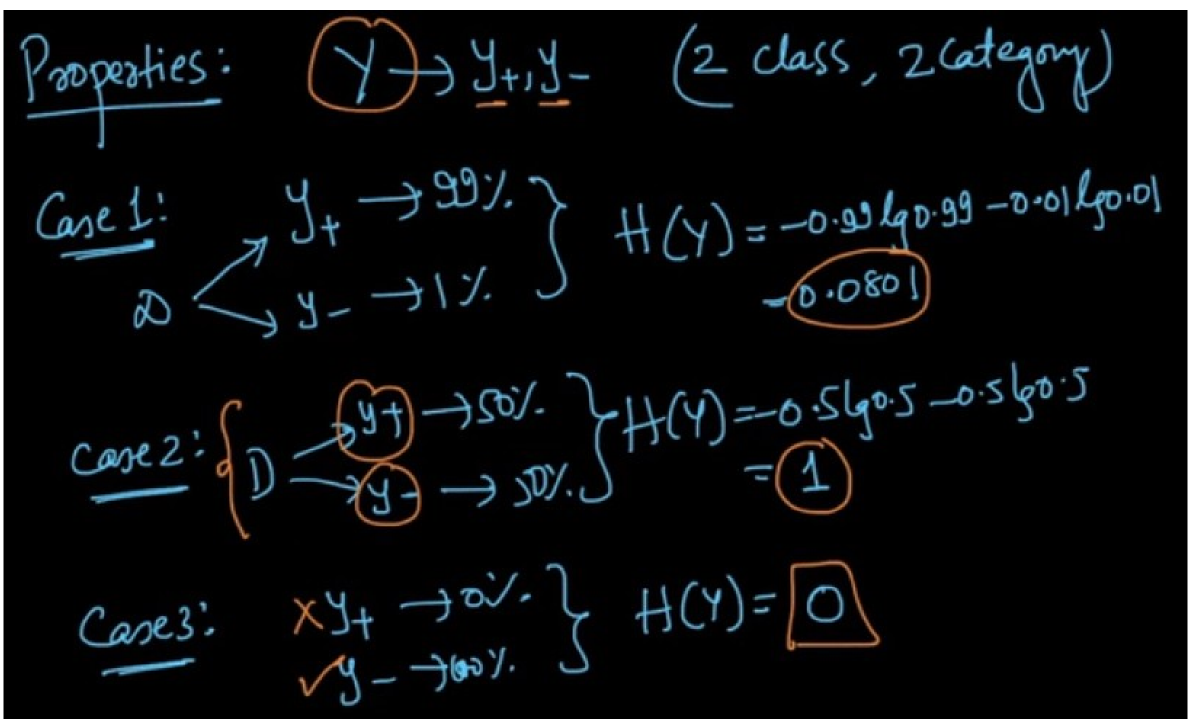
Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	~~Yes~~
Rainy	Mild	High	False	~~Yes~~
Rainy	Cool	Normal	False	~~Yes~~
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	~~Yes~~
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	~~Yes~~
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



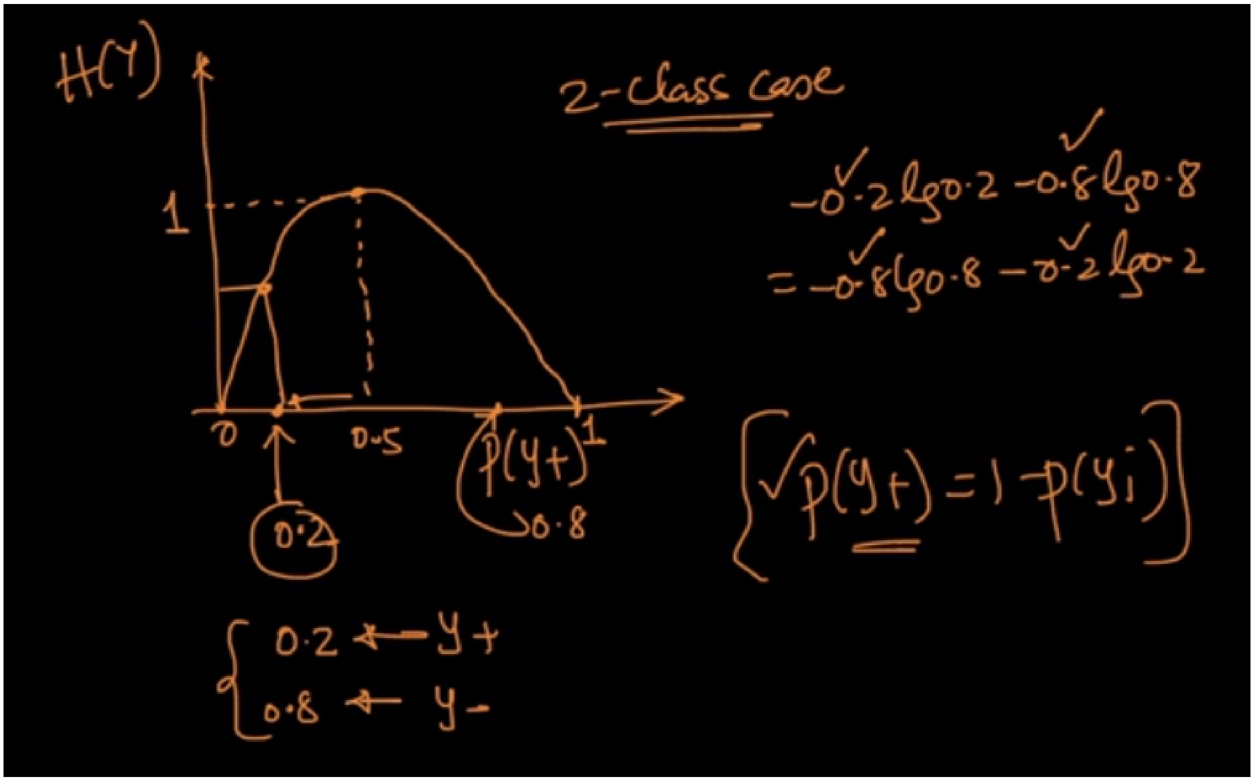
Calculation of the Entropy



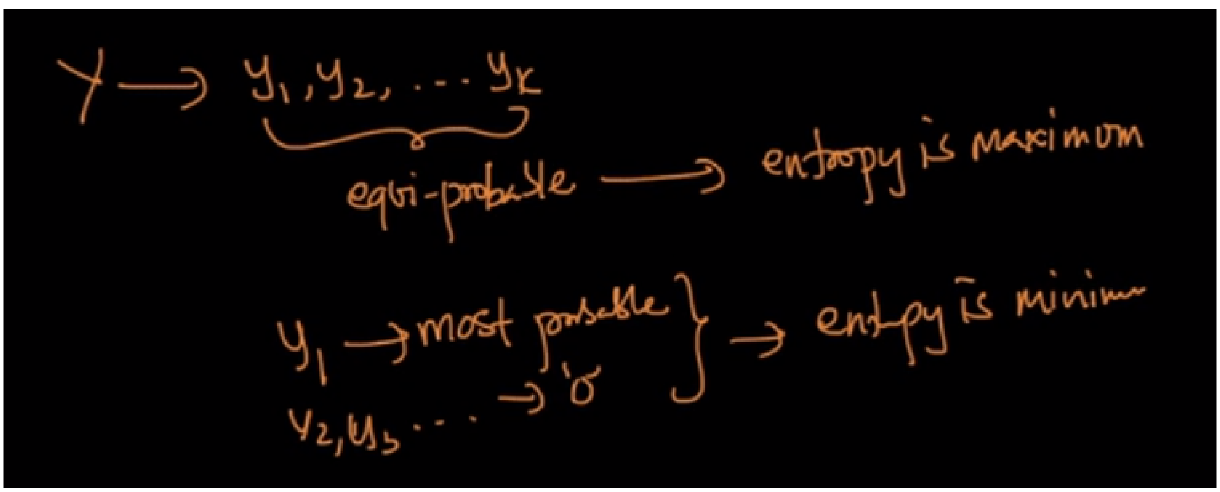
Properties of Entropy:
Various cases:



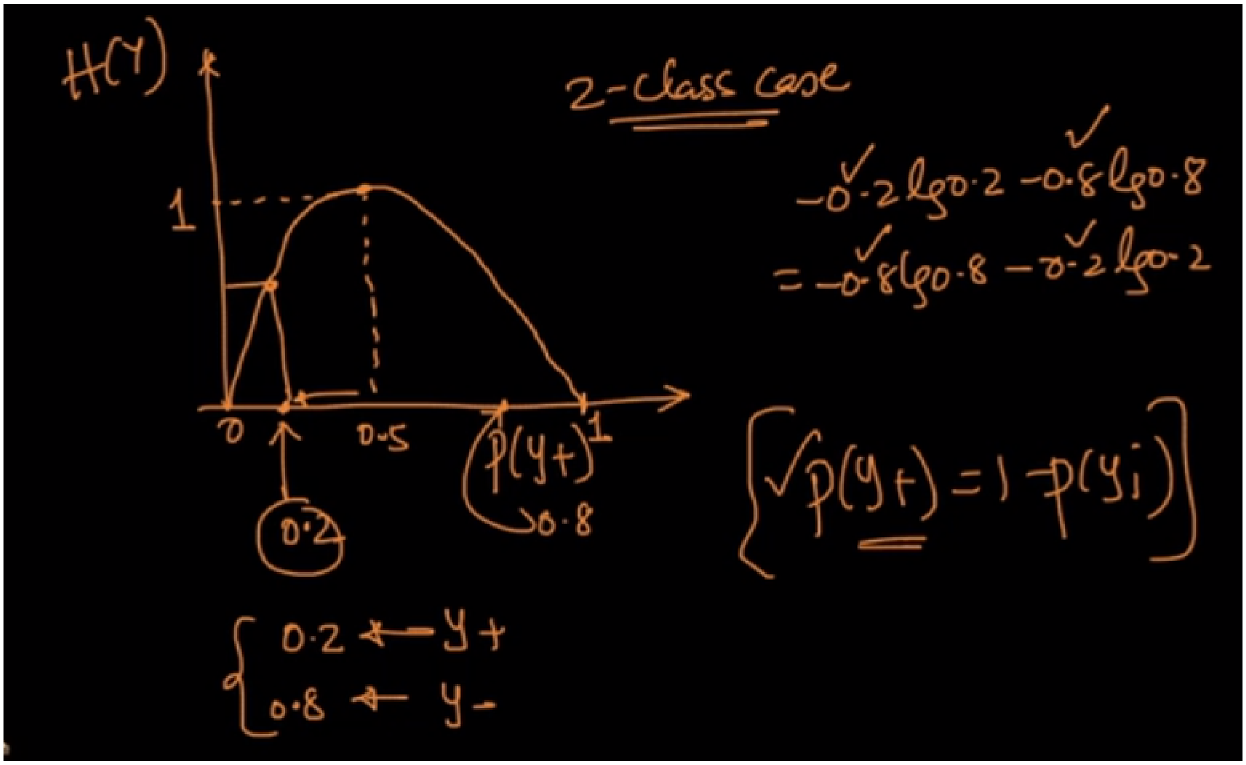
Entropy curve:
This is a symmetric curve.



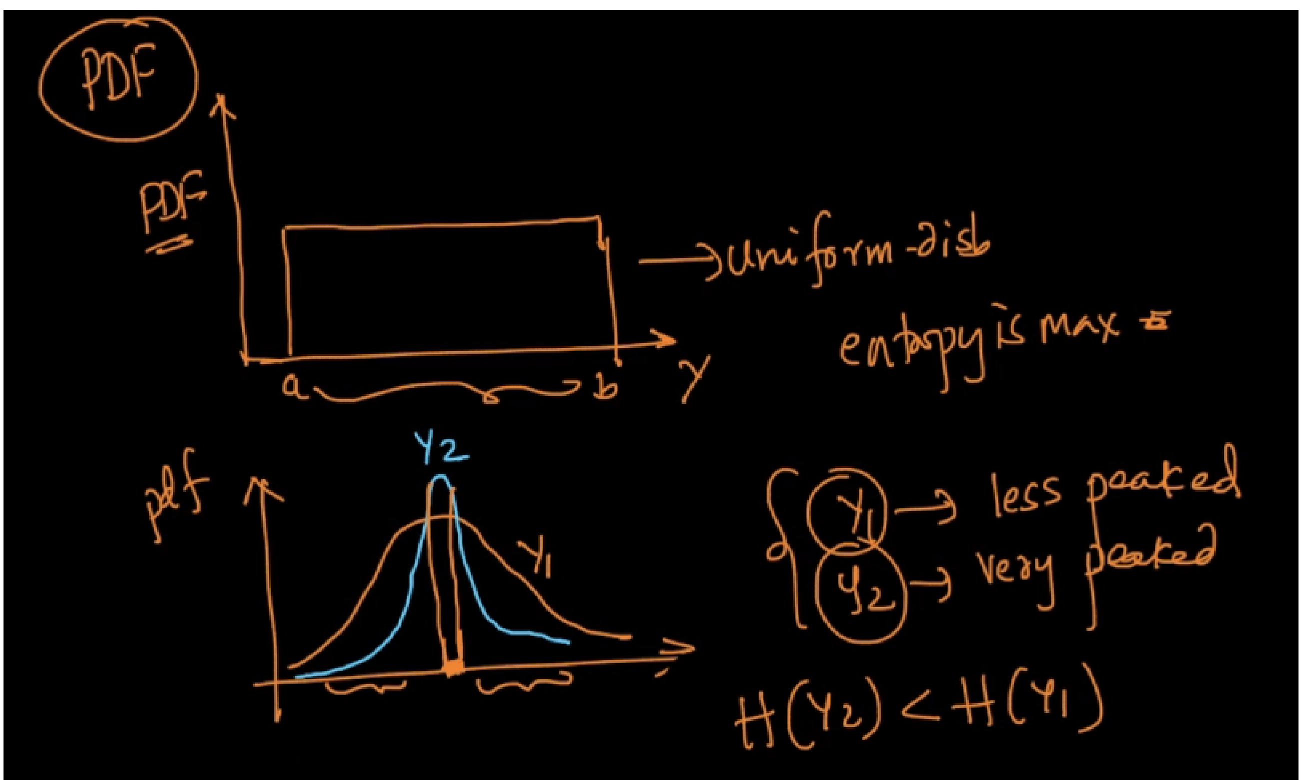
Conclusion:



Entropy for real – valued feature:

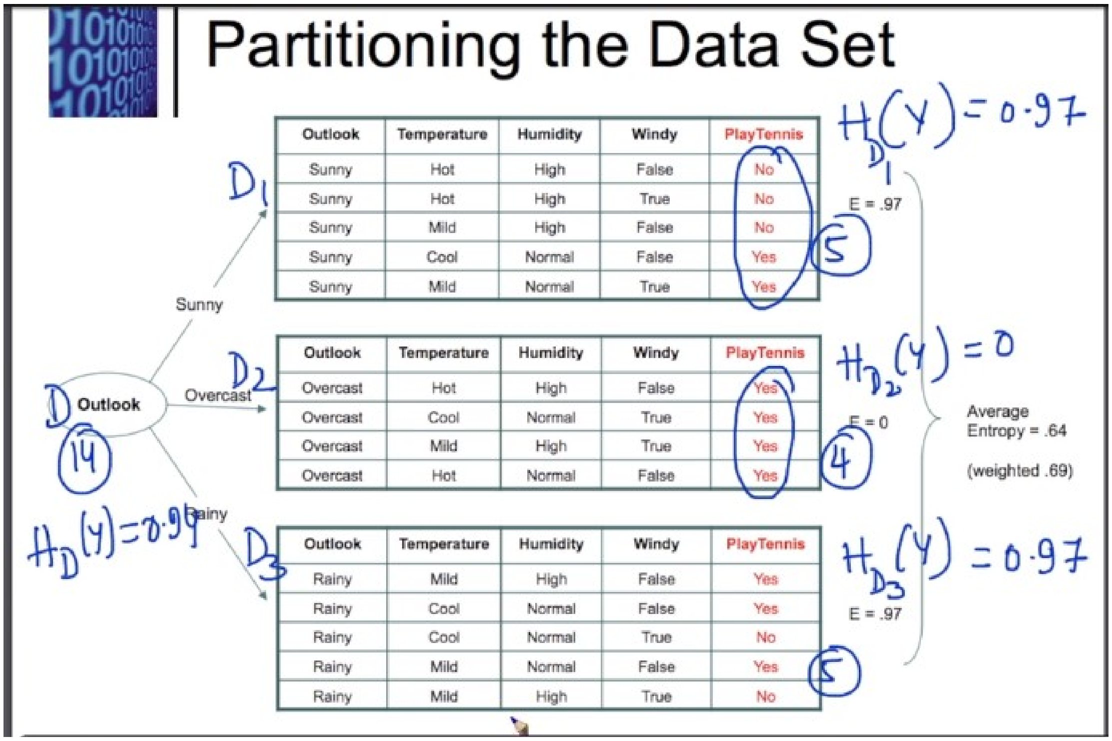


Given a random variable, if all of them are equi – probable, entropy is maximum.
If the data distribution is more uniform then the entropy is more.

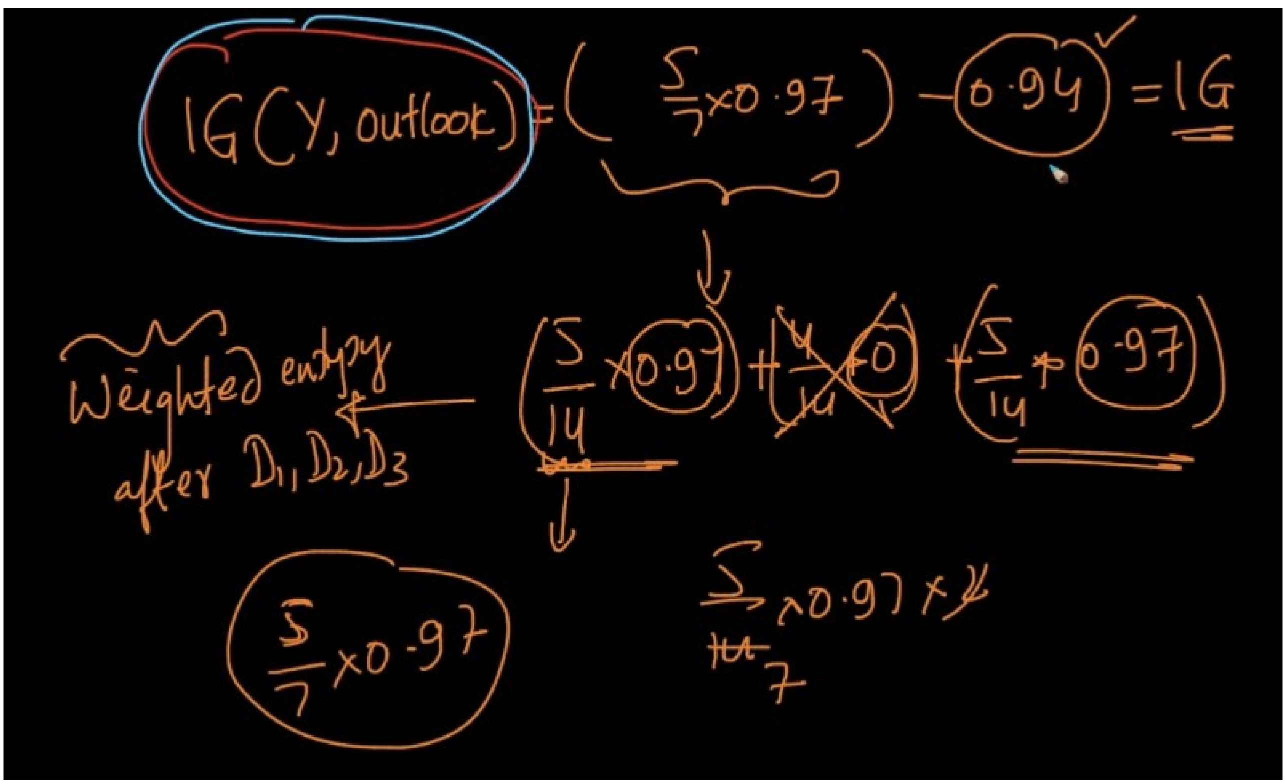


If the data is more like gaussian then the data has less entropy.

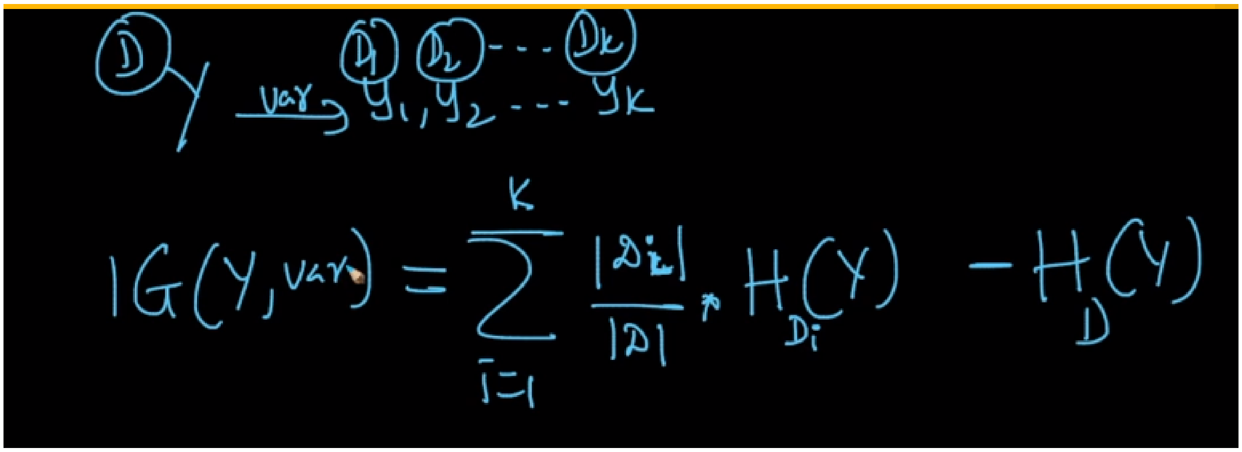
Information gain:



Calculation of information gain:



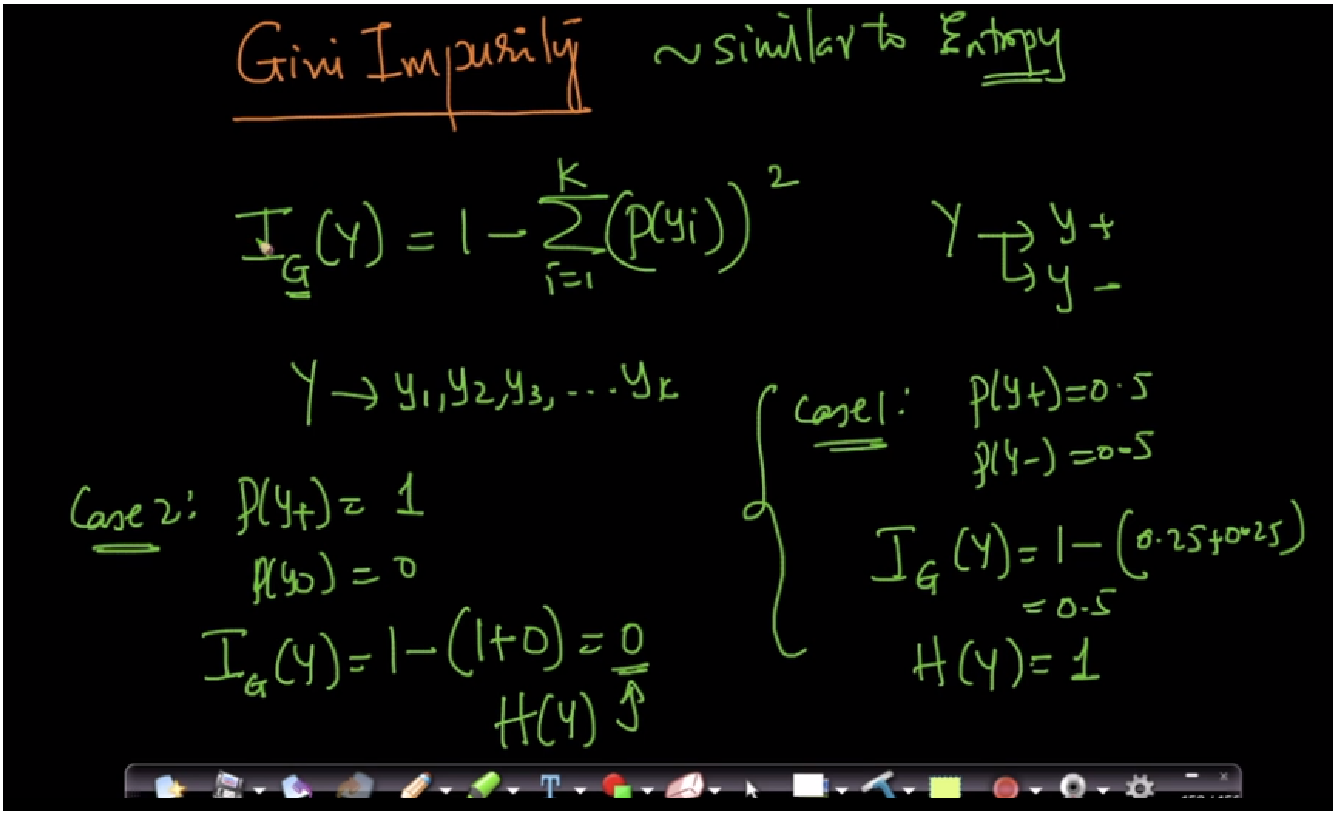
Formula for Information gain:



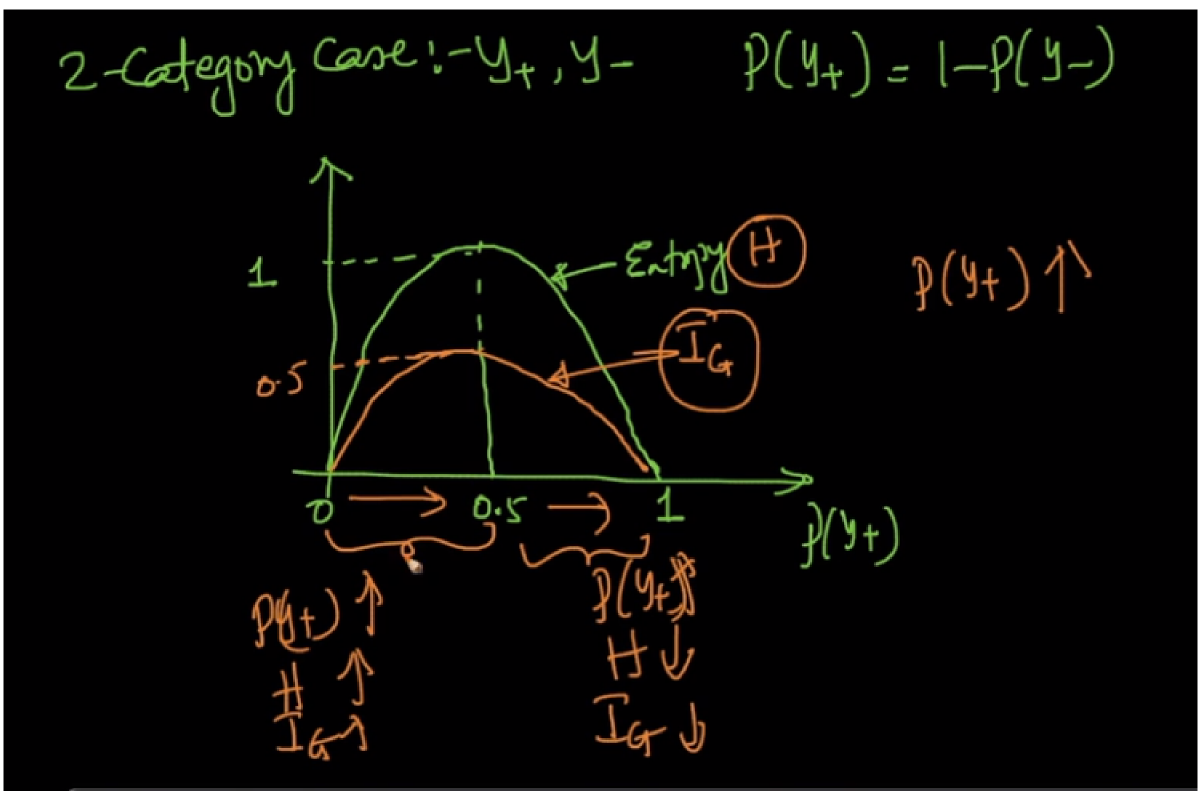
$$D \; Y \xrightarrow{var} \underset{y_1}{D_1}, \underset{y_2}{D_2} \cdots \underset{y_k}{D_k}$$

$$IG(Y, var) = \sum_{i=1}^{K} \frac{|D_i|}{|D|} * H_{D_i}(Y) - H_D(Y)$$

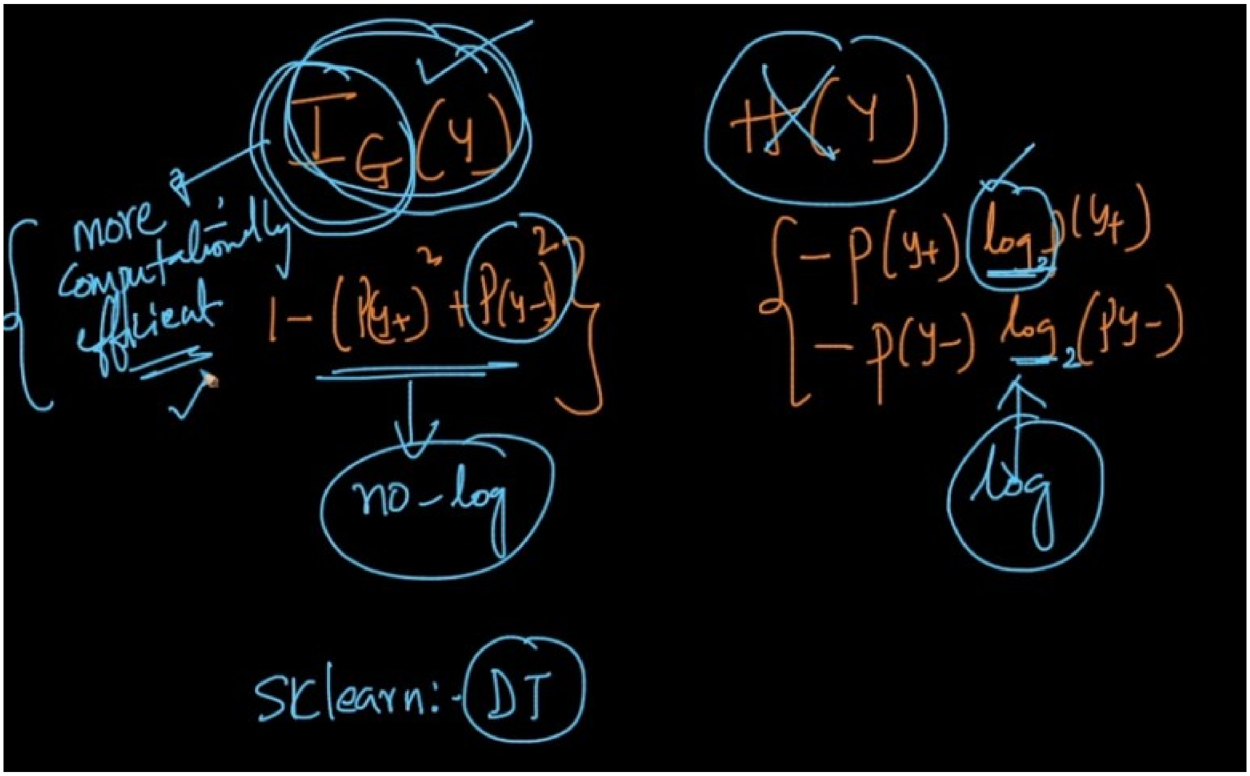
Gini – Impurity:
Case 1:



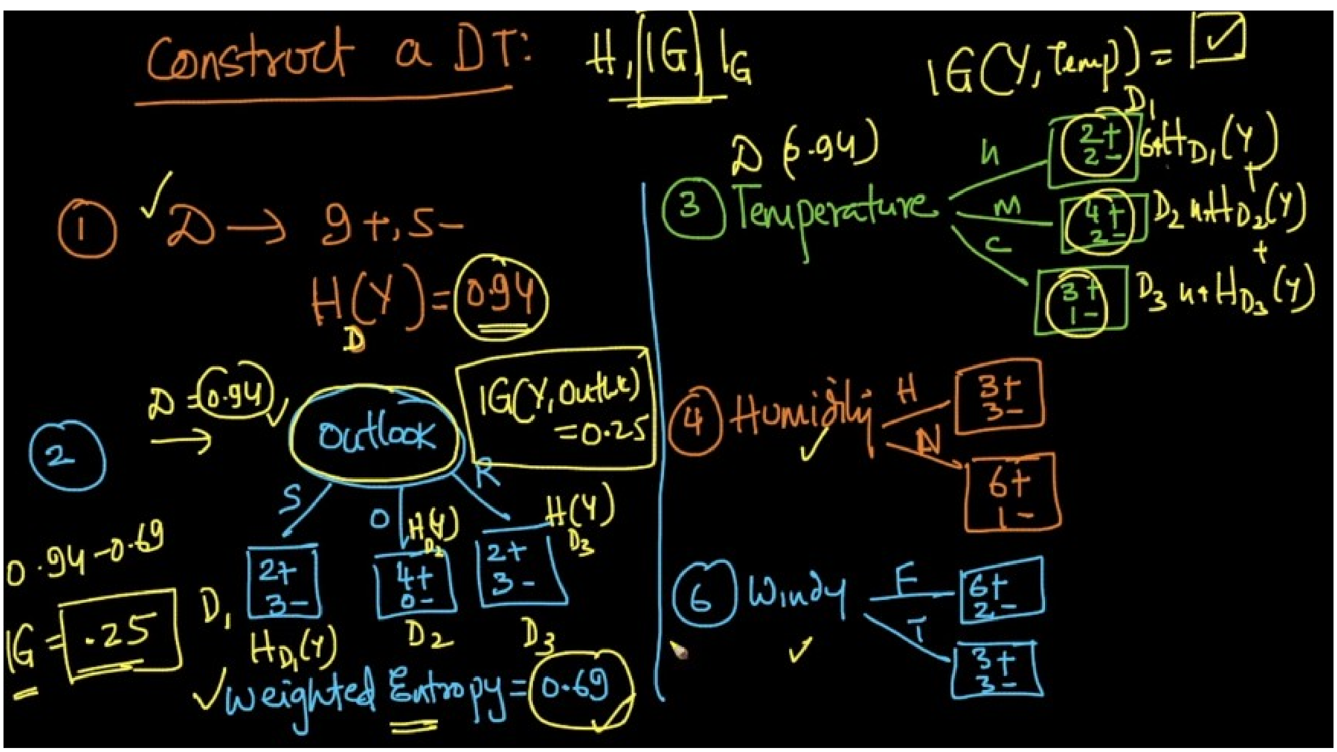
We assume there are only two classes of a feature:

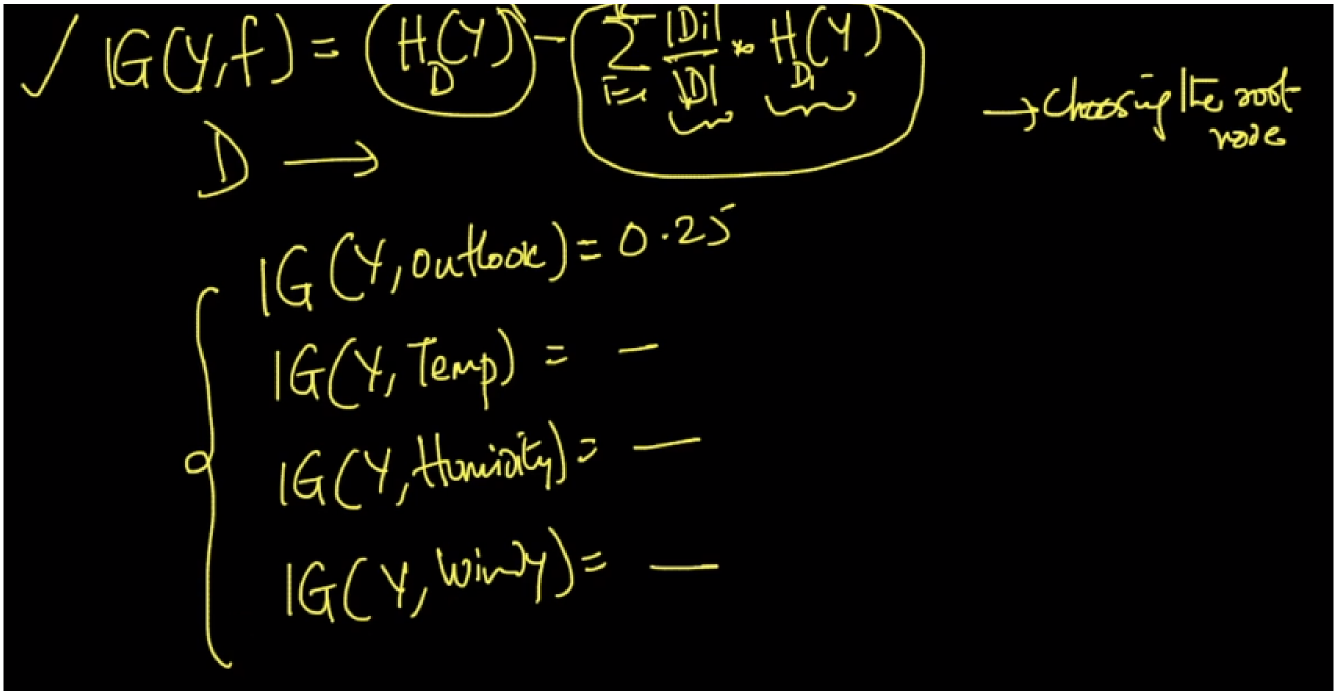


Calculation of information gain is computationally more expensive than entropy, In real world people will take gini - impurity as the measure.

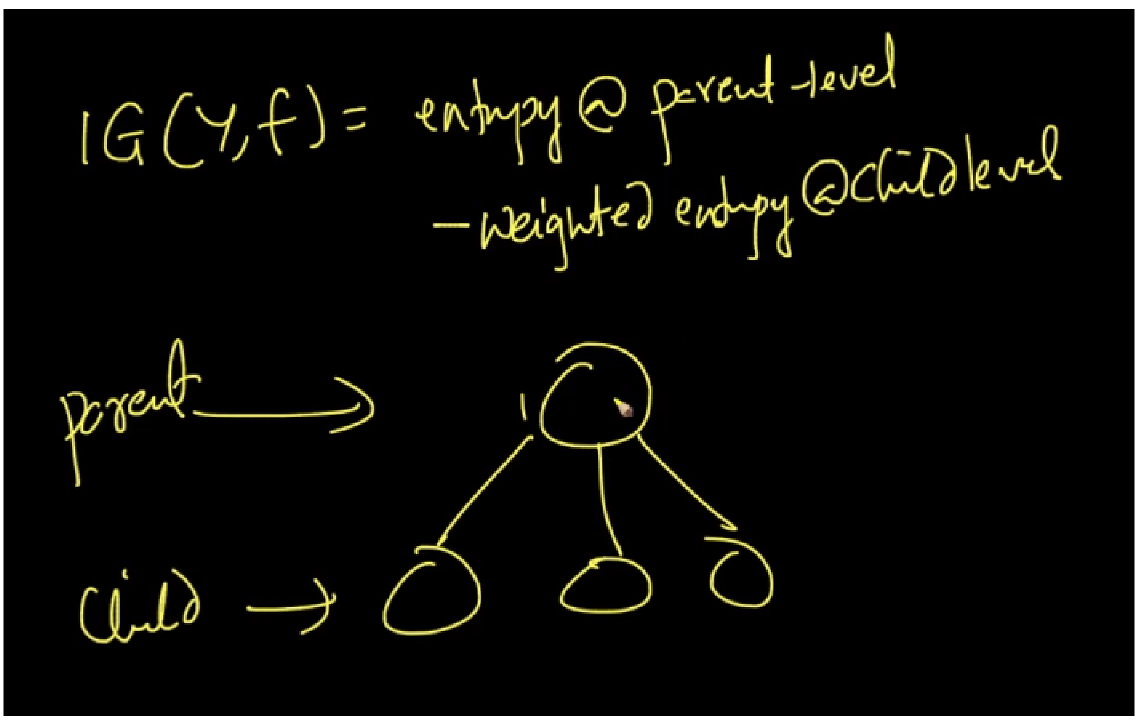


Building a decision Tree: Constructing a DT on a data set.
Breaking the data set with the feature outlook

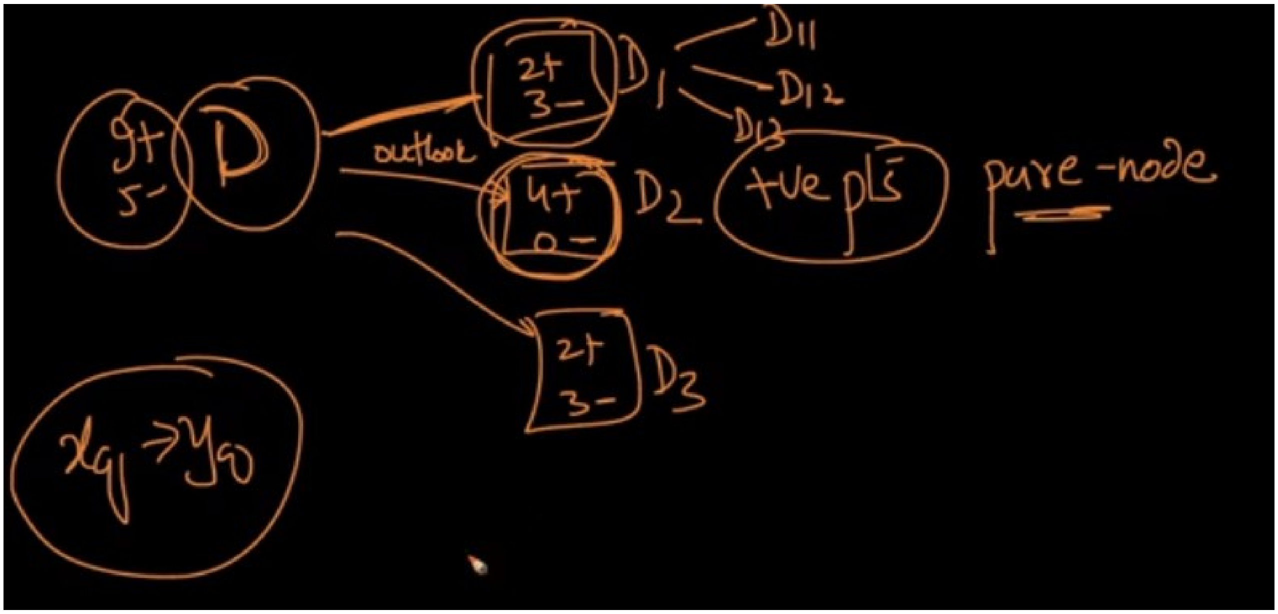




Information gain on breaking the data set is the (entropy at the parent level) – (weighted entropy at the child level). **The node is chosen which has most information gain.**

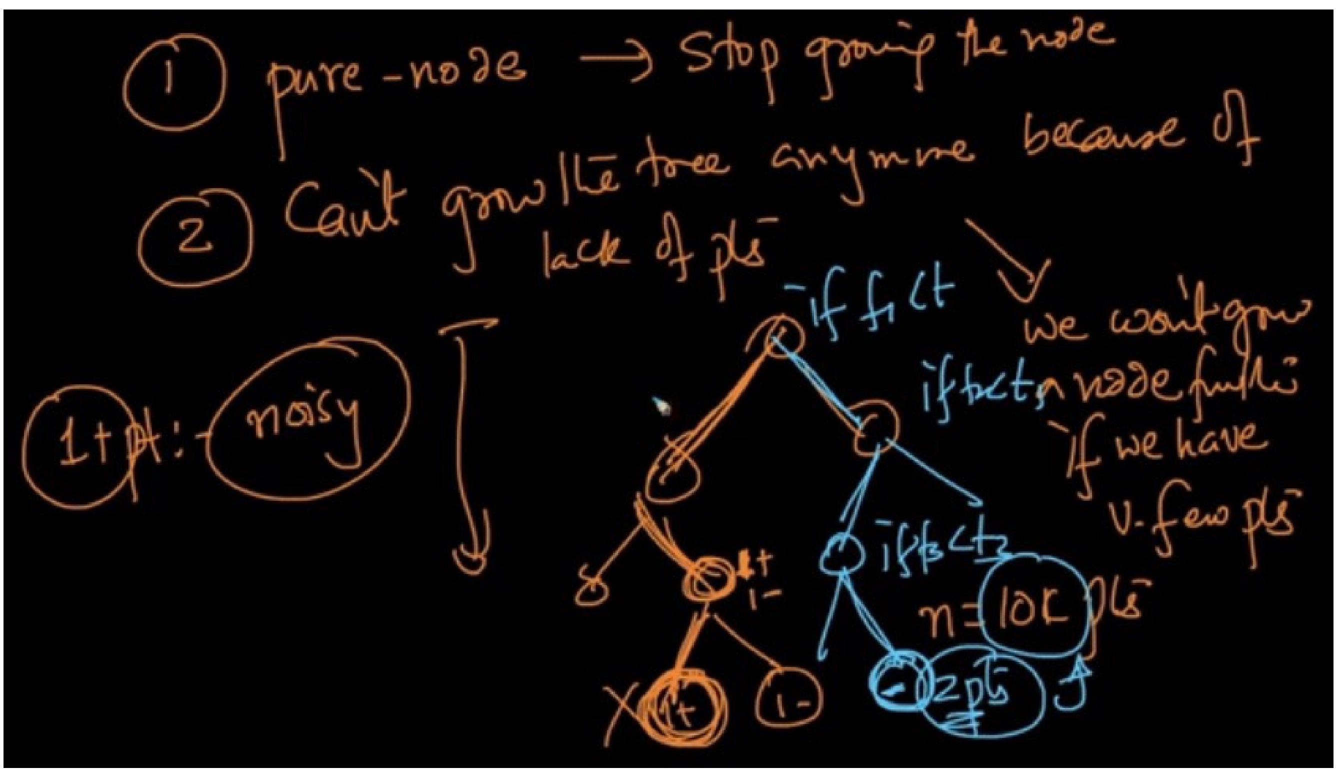


Node that has only one class label then the node is called pure – node.



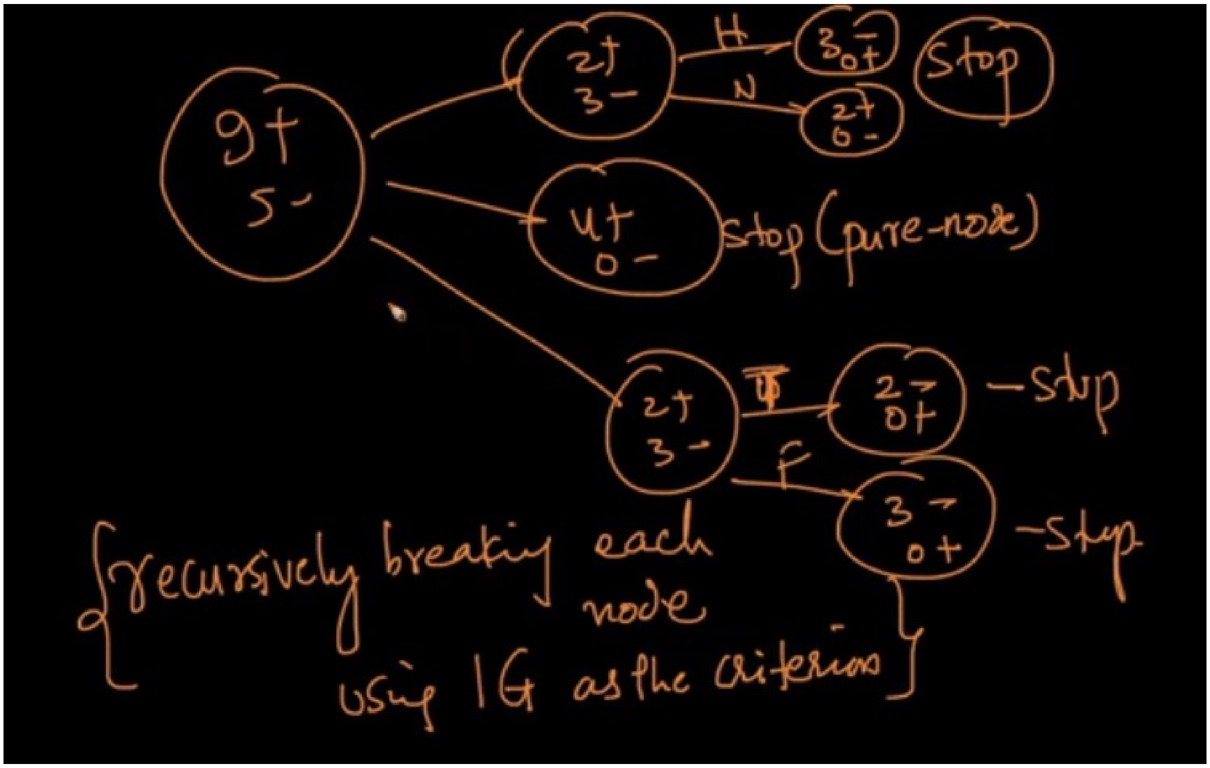
Recursively breaking each node basing the maximum **information gain** as the criteria till we get the **pure node**.

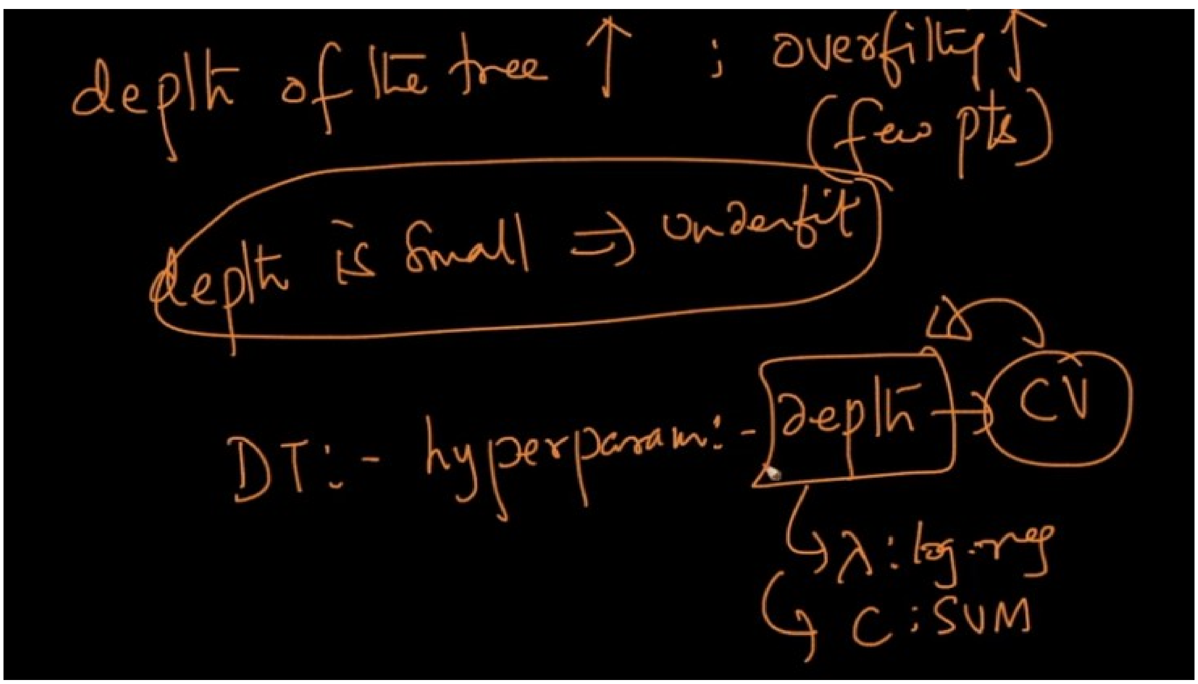
Cases of stop growing the tree:



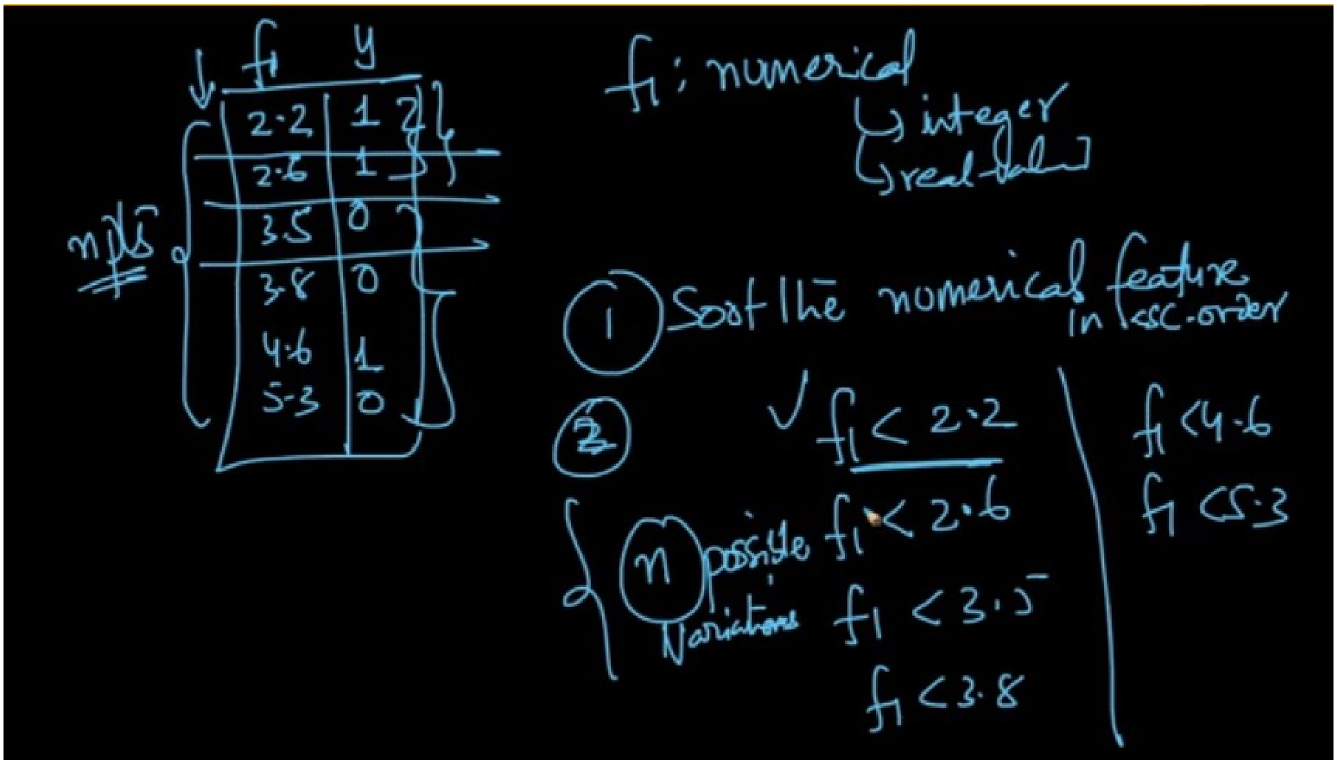
If the depth of the tree increases chances of over fitting the data increases.

If depth is small then the decision tree tend to under fit.
The Hyper parameter in decision tree is the height of the tree.
We use cross validation to choose the depth of the decision tree.

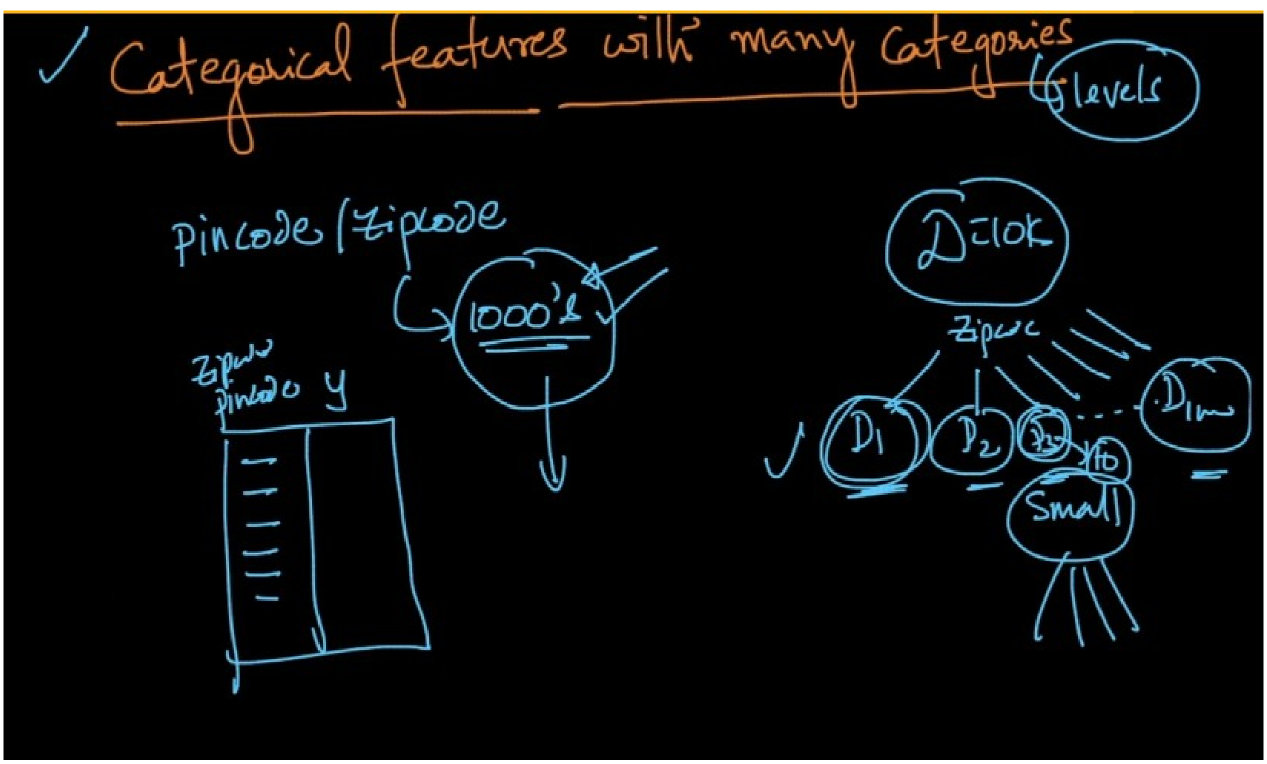




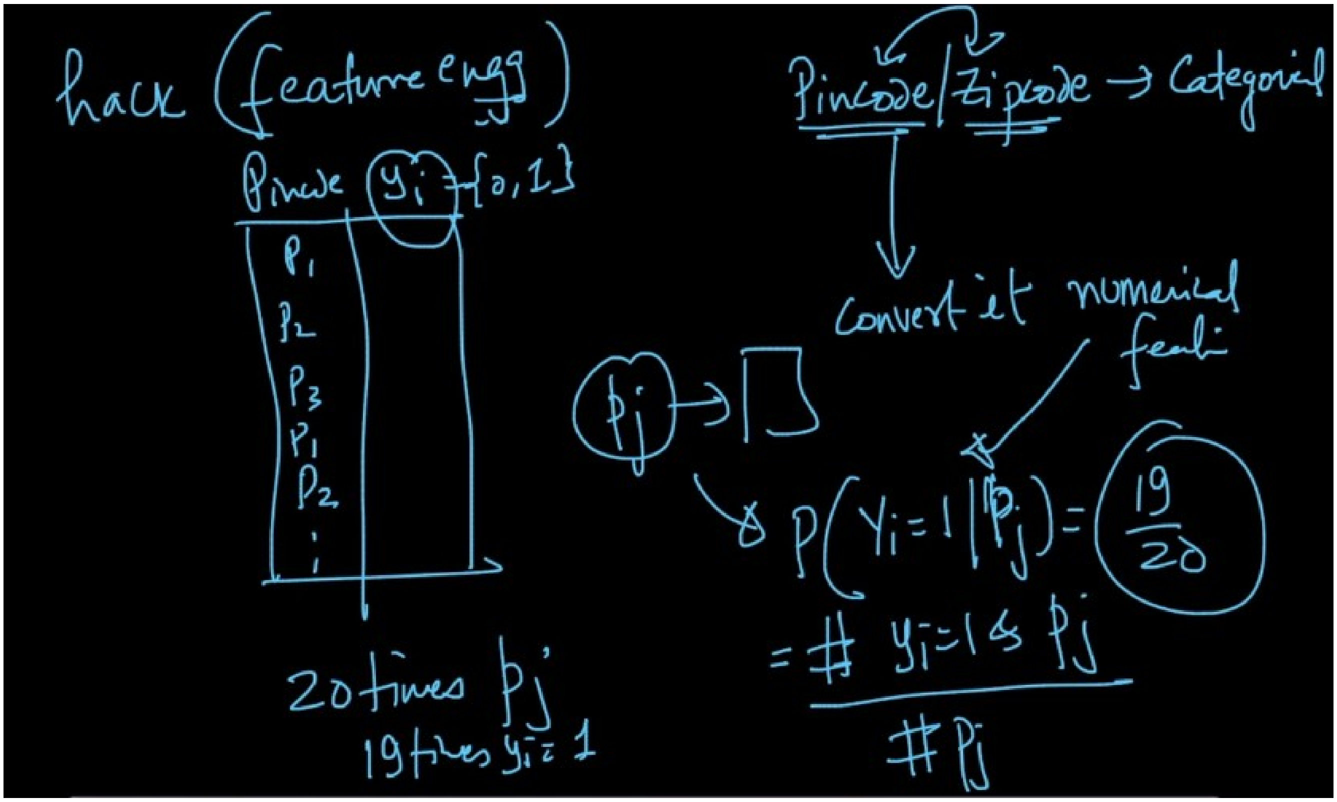
Building a decision Tree: Splitting numerical features:



Building a decision Tree: Categorical features with many possible values:
Example: ZIP code, PIN code.



Converting a categorical feature to a numeric feature:

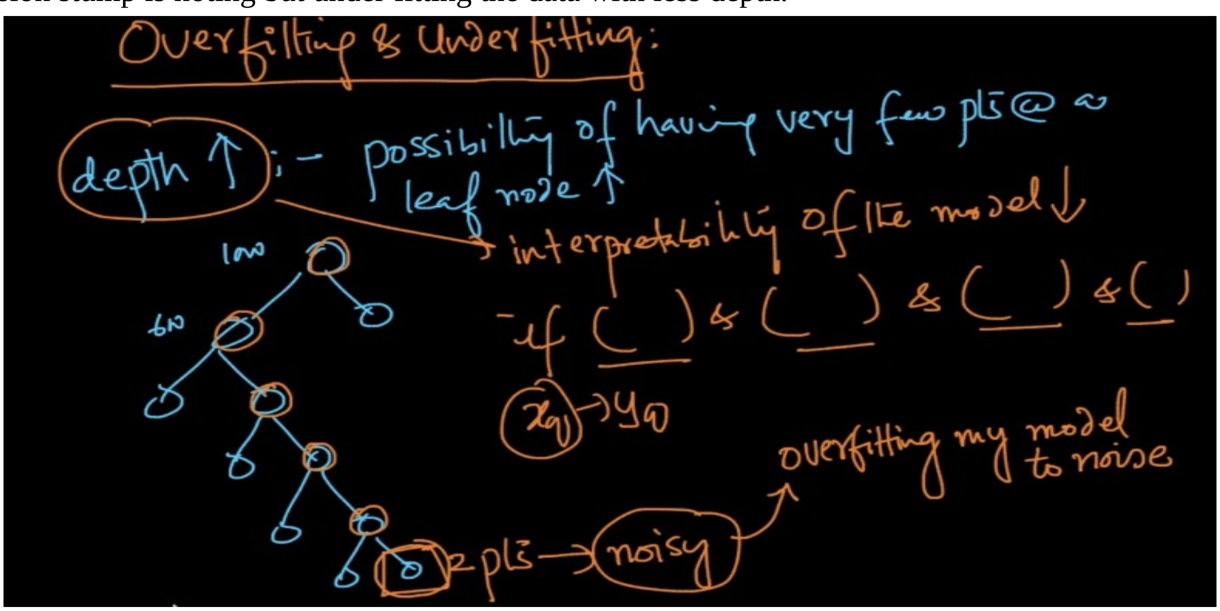


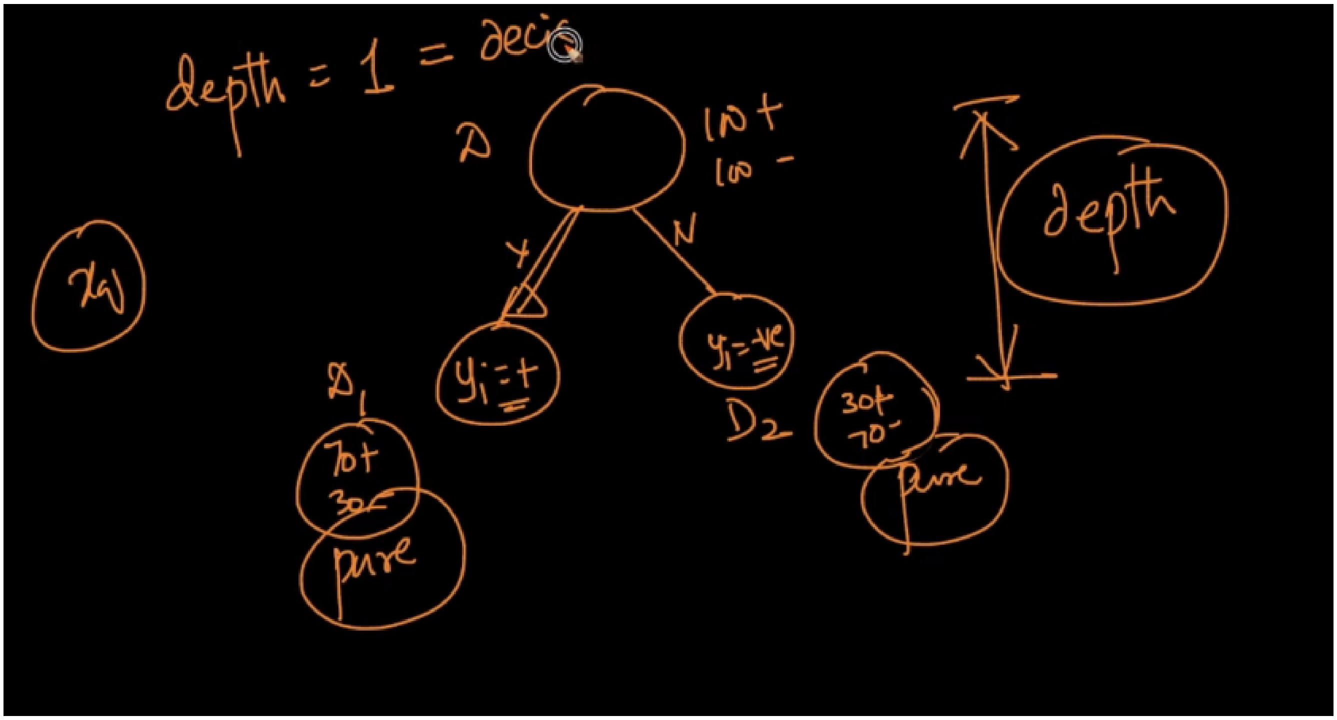
Each feature is converted according to the class of the feature by calculating the probability of occurrence of the feature variables.

Overfitting and Underfitting:

If there are outliers or noise in the data then the decision tree tend to fit these points and make the model to over fit the data.

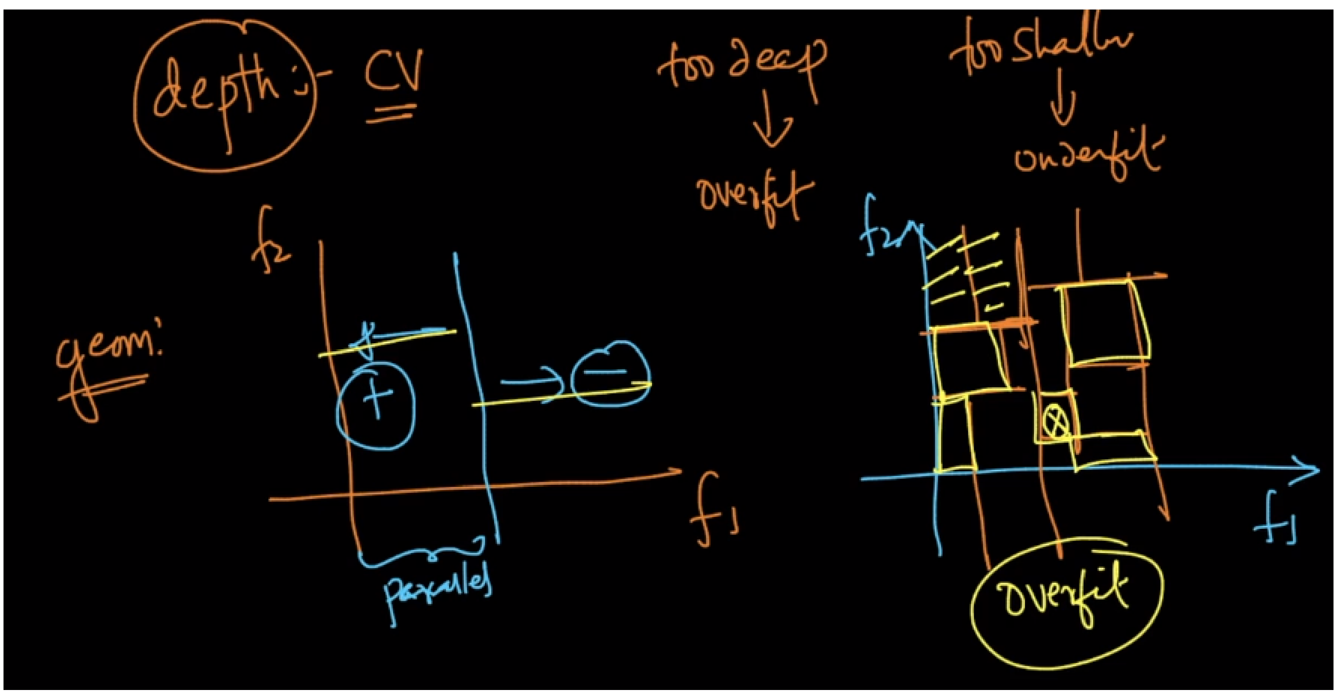
A decision stump is noting but under fitting the data with less depth.





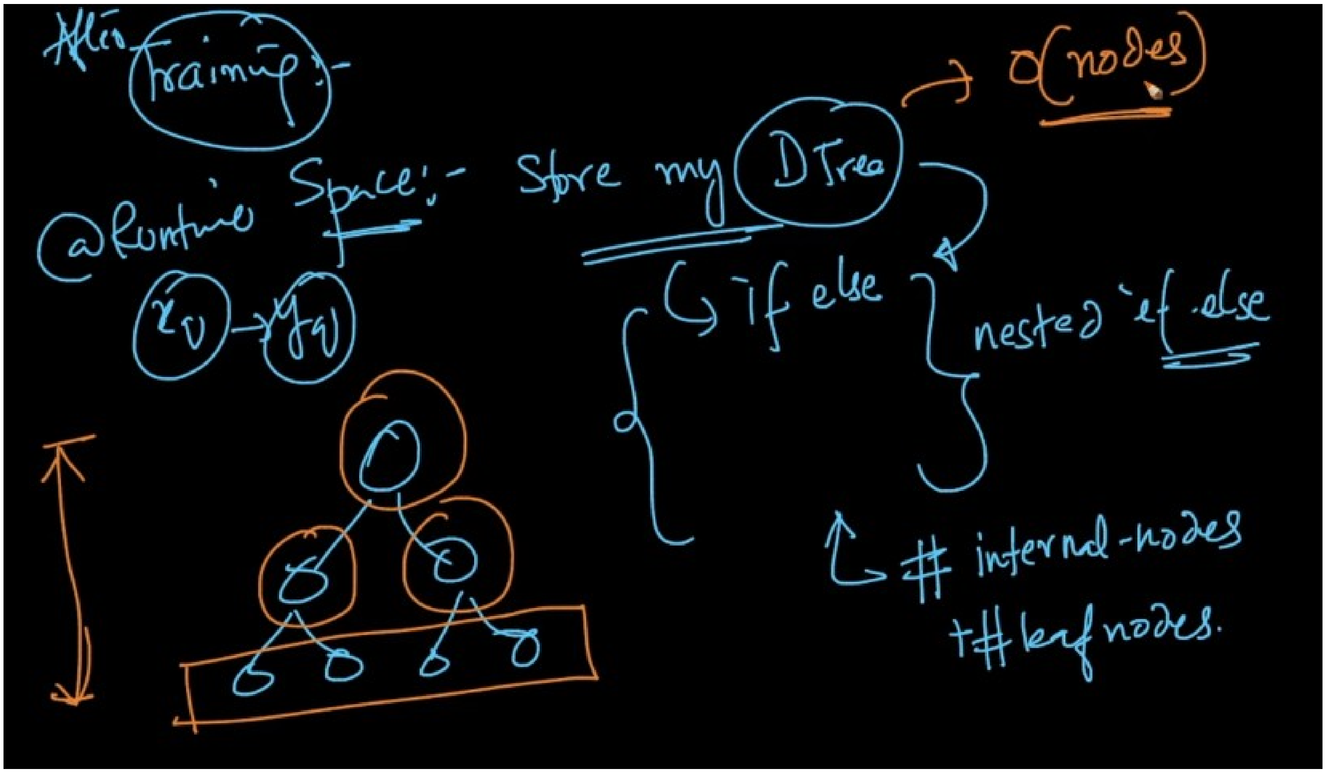
Depth is calculated using cross – validation

Visualizing Over-fitting and under-fitting:

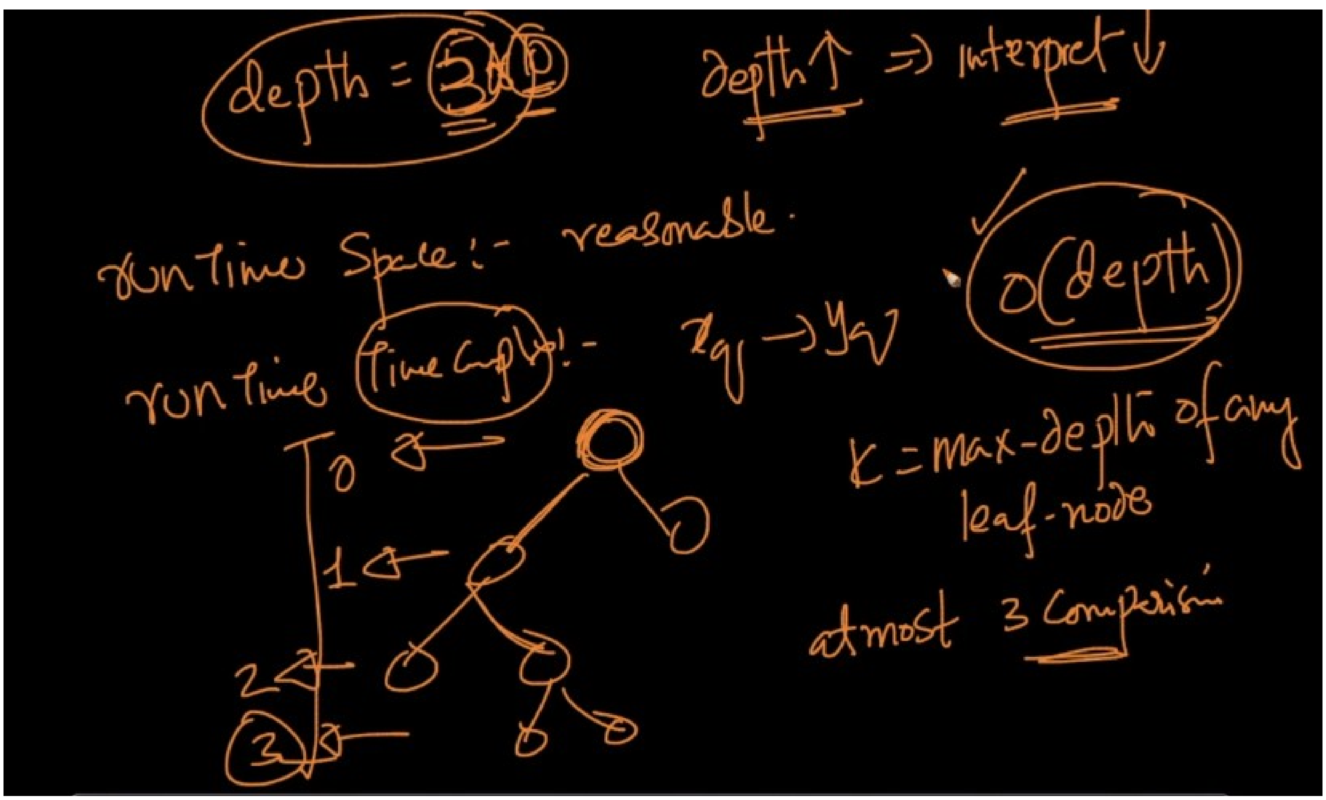


Train and run time complexity:

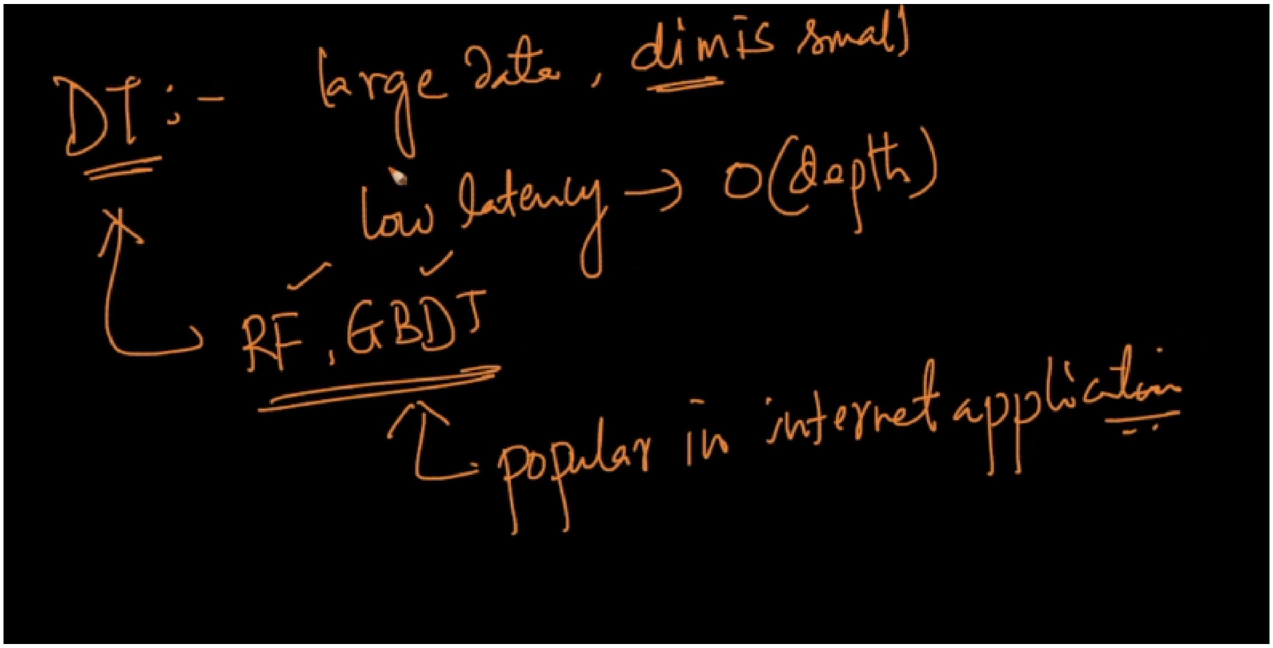
Converting a decision tree to nested if – else conditions can save space.



At max. a decision tree is trained to be 5 – 10 levels of depth.

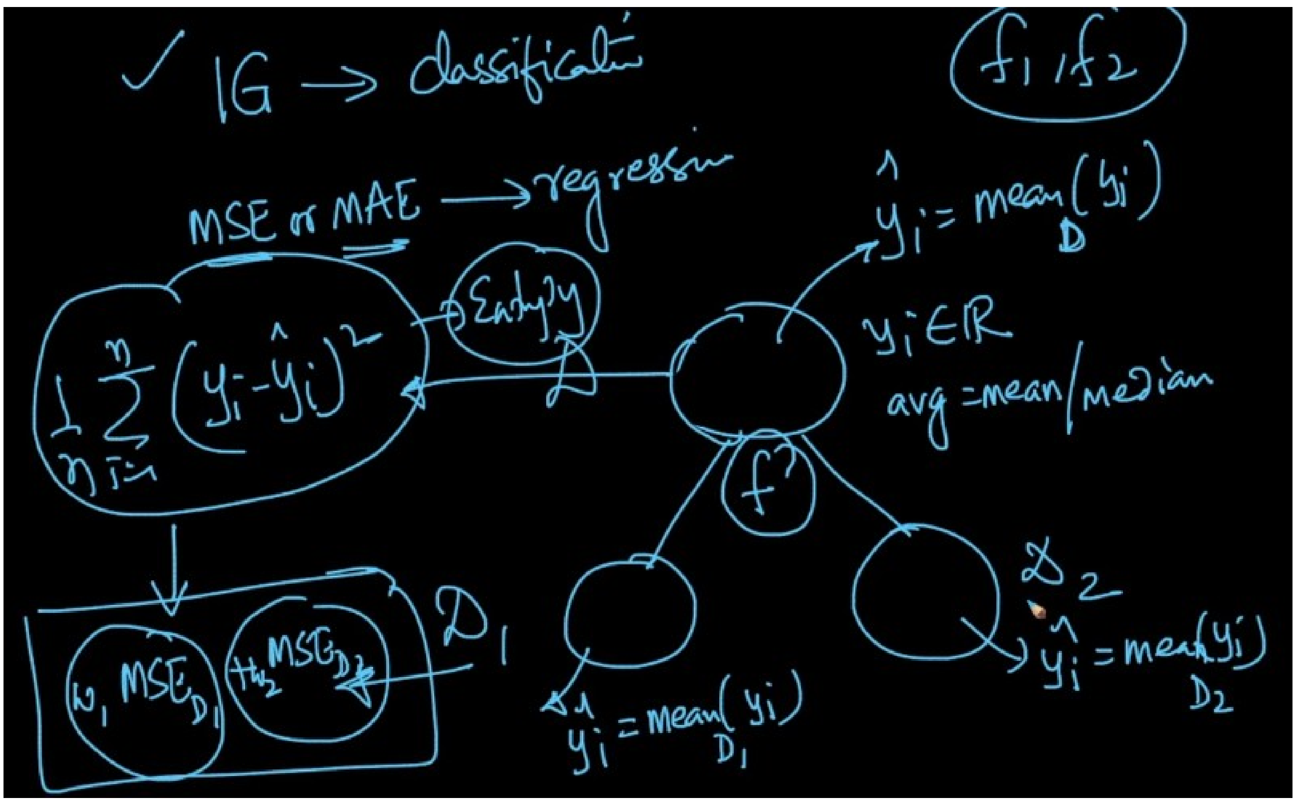


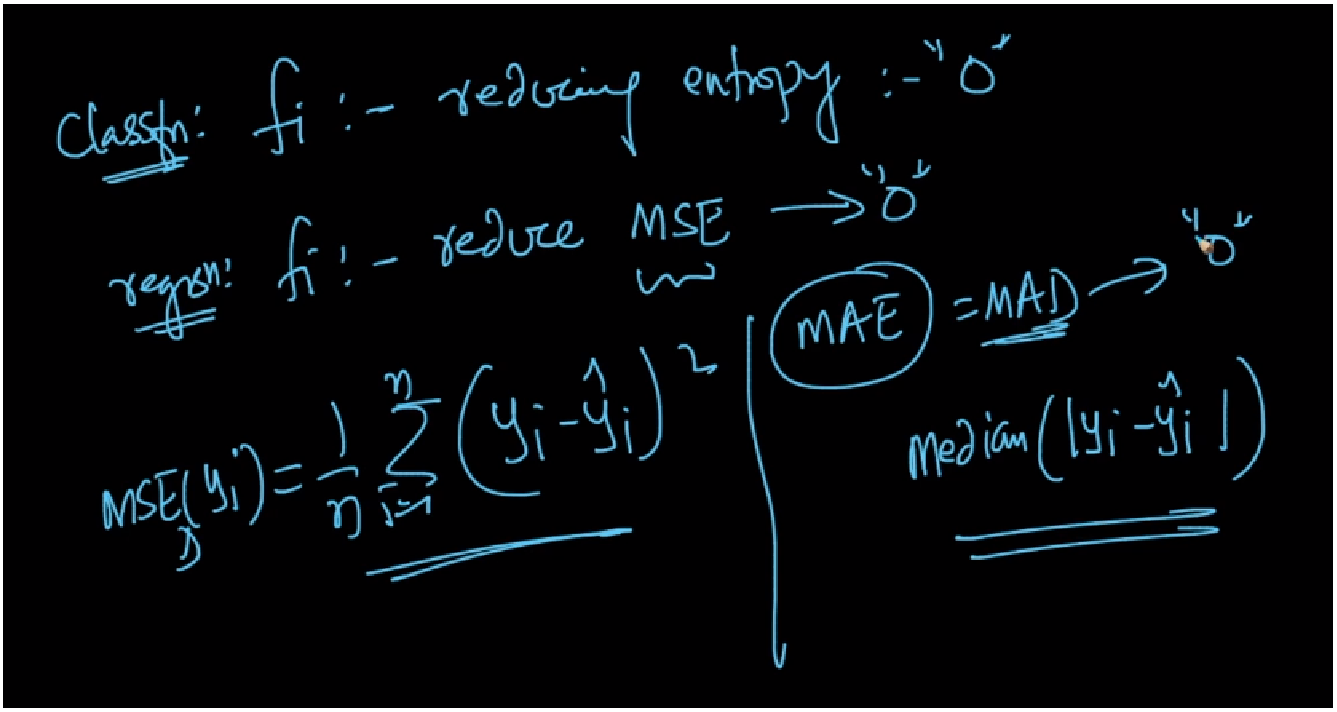
Decision tree can handle Large data, dimensionality is small or reasonable, low latency.

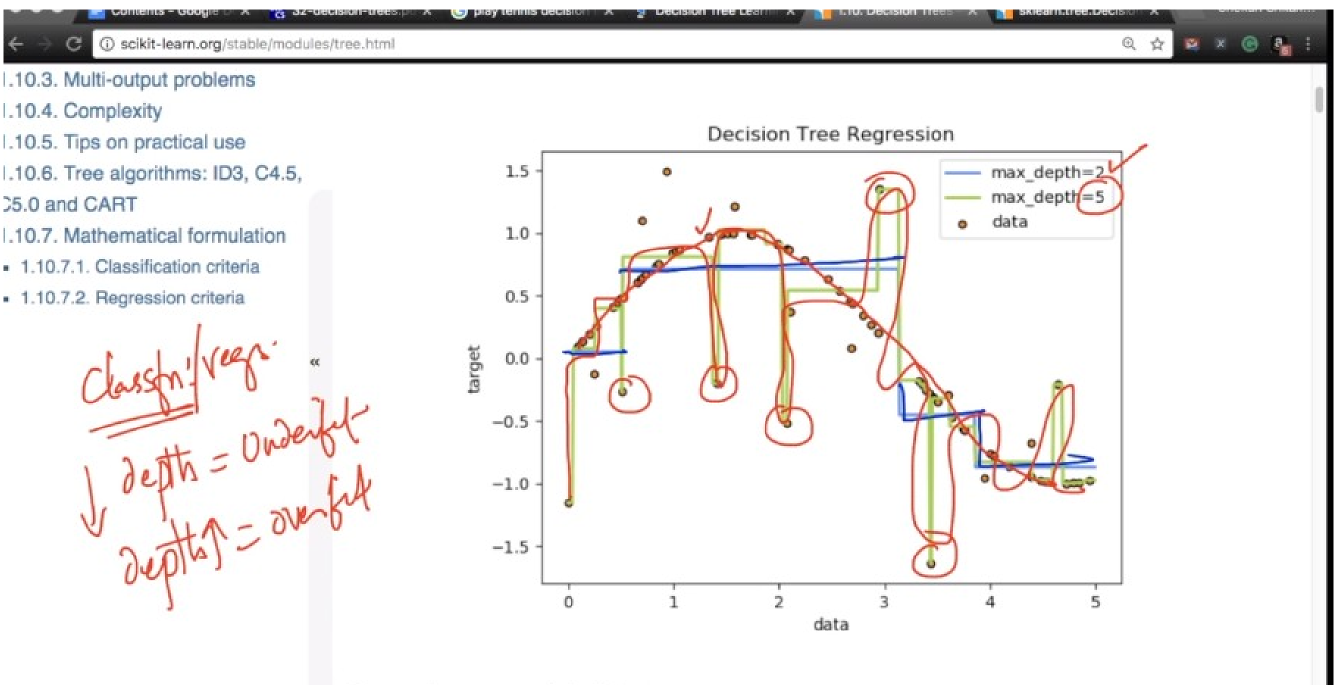


Regression using Decision Trees:

Instead of using Information gain we use Mean square error (or) mean absolute deviation is used to make regression trees.

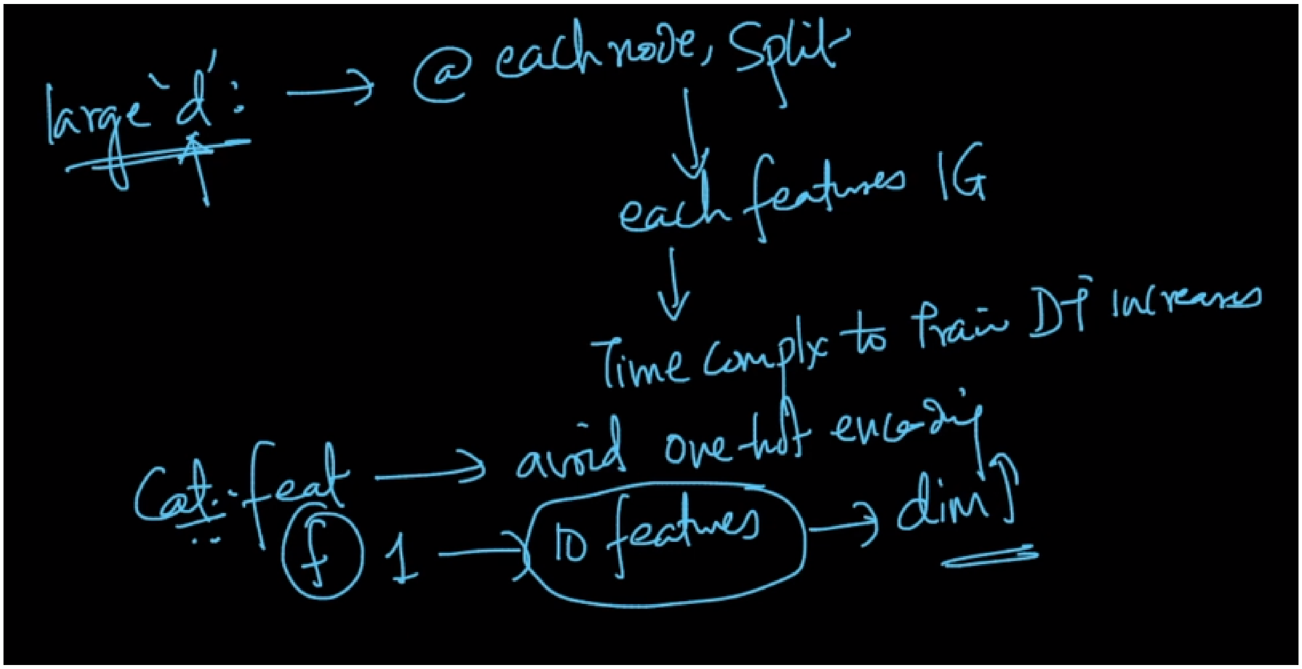






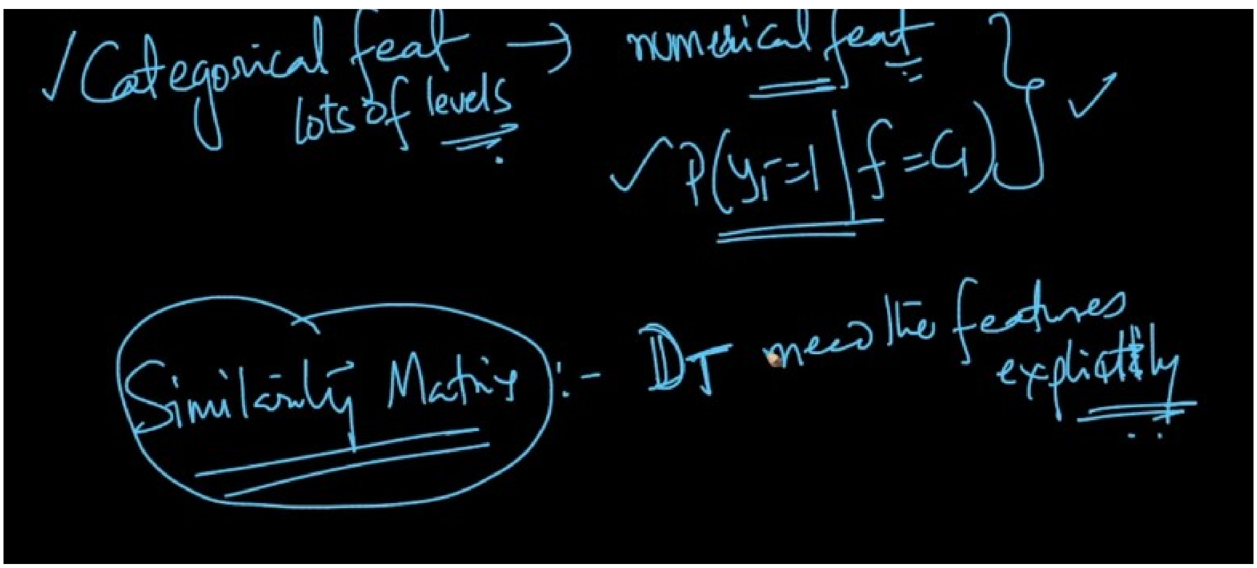
All the lines are axis parallel in decision tree model for regression / classification.

If dimensions are large, then the time training the data.
One should avoid One – hot encoding.

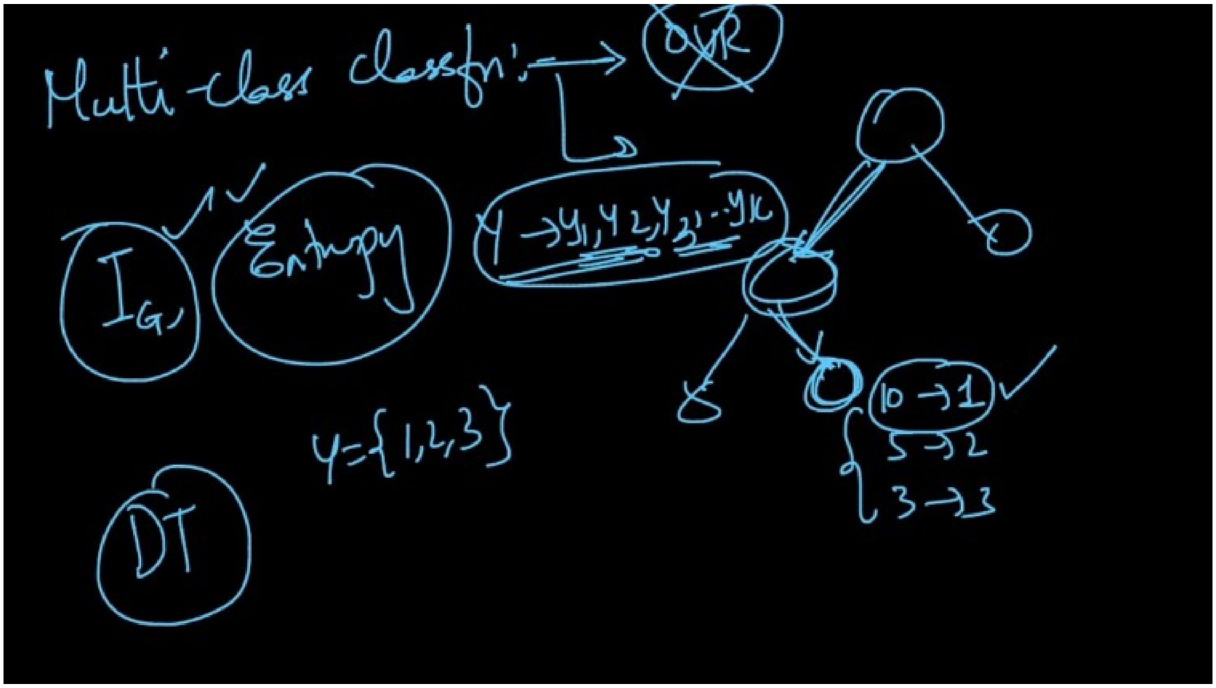


Converting into numerical features will save a lot of time.

Decision trees can read the data explicitly not in case is similarity matrix.

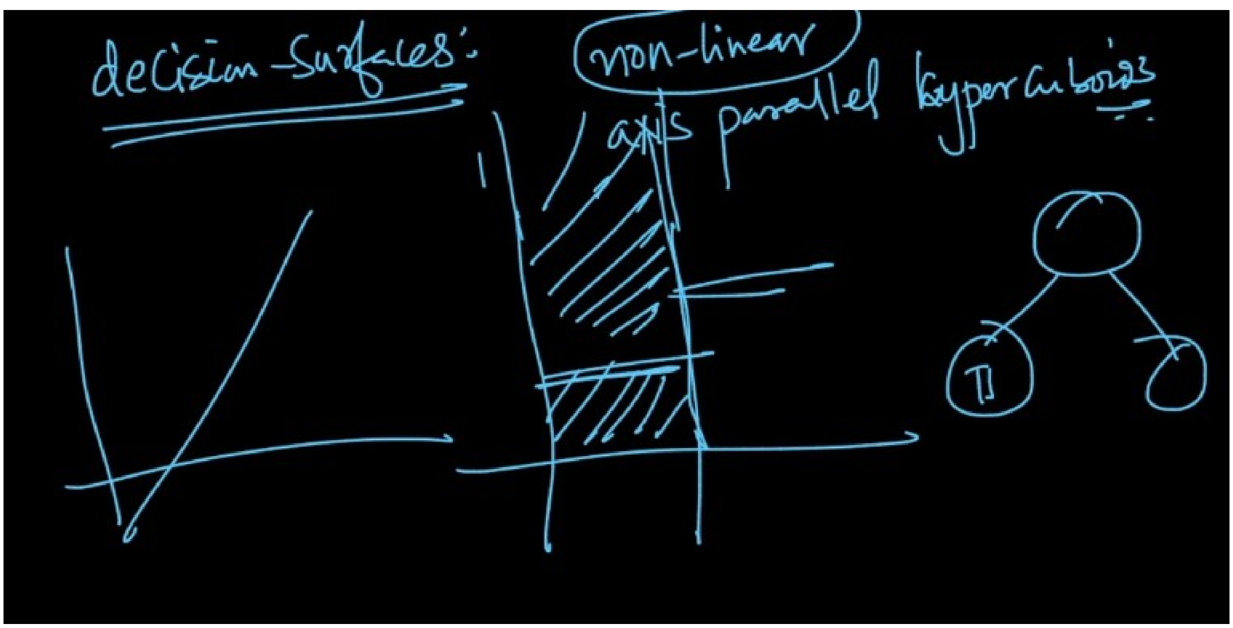


Decision trees can naturally can be extended to multi – class classification.



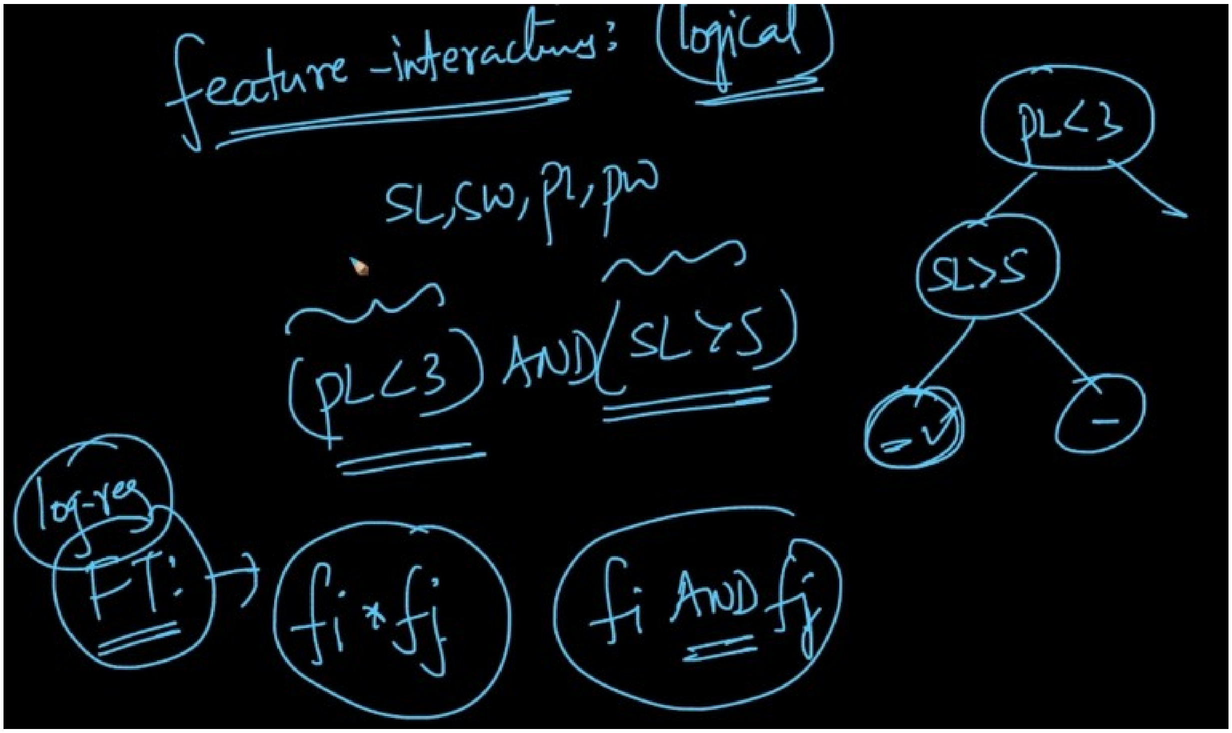
Decision surface:

The decision that we get are non – linear. It basically divides the data into axis – parallel planes / hyper planes.



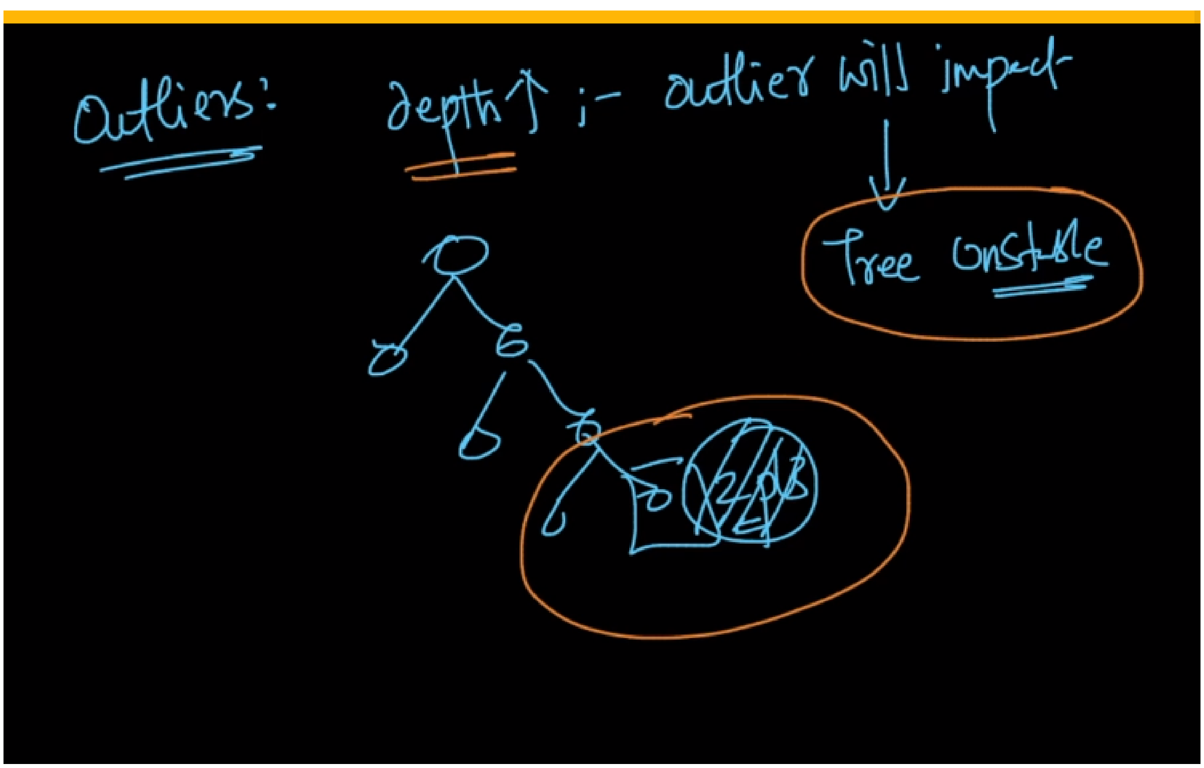
Feature interactions and decision trees:

There are feature interactions in decision trees to decide the class of the query point.
There are logical feature interactions in decision trees.



Outliers:

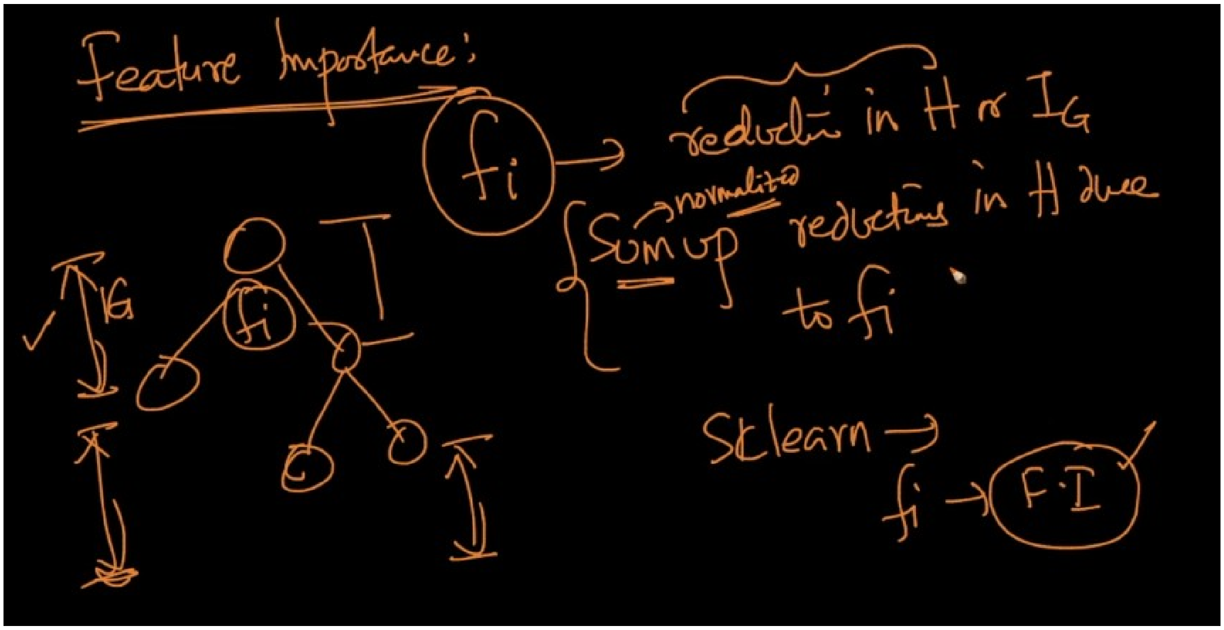
When depth is large then the model is prone to outliers.



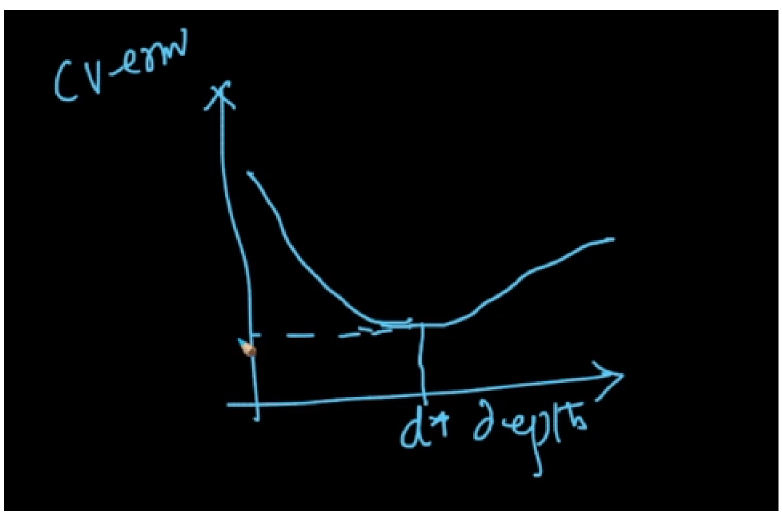
Interpret – ability:

Interpret – ability is very easy in decision trees.

Feature importance:
We can sum up the reductions in entropy of each feature based in the importance of the feature.
If one feature occurs more than one time then we can conclude that feature is more important.



Exercise:



Ex: DT on Amazon Food reviews:

Try:

$\left\{\begin{array}{l} \text{BoW} \\ \text{tfidf} \end{array}\right\} \rightarrow$ d is large (10's of thousand)

w2v $\rightarrow \left\{\begin{array}{l} \text{avg} \\ \text{tfidf} \end{array}\right\} \rightarrow$ 200 dim

DT on w2v $\rightarrow$ depth $\rightarrow$ CV