**Lecture 2: Perception and Knn**

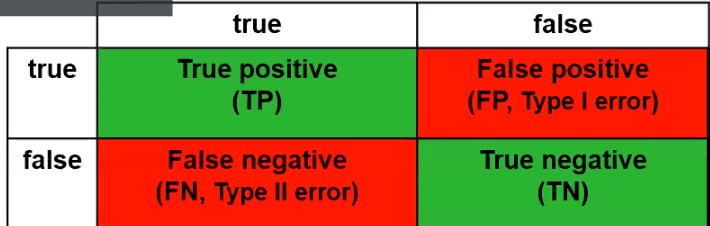
* Perception is a simple linear model – y= w1.x1 + w2+x2

We want to separate the two classes with line. The line is always in the middle of the x and y axis. In order to change its position is to add bias. This bias we also change with the other hyperparams.

* K-nn – we say that one point is the same class as how many k neighbors it has. If k == len(x) then the majority of the class will win always. This is not a model. This is lazy learner.

For regression, it takes the mean value from the k neighbors

* Confusion matrix:



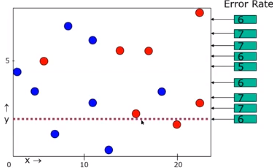
Percision = TP / TP + FP – We have to be careful to label as positive.

Recall = TP/TP + FN – How to optimize? Highest possible recall. To minimize False Negatives.

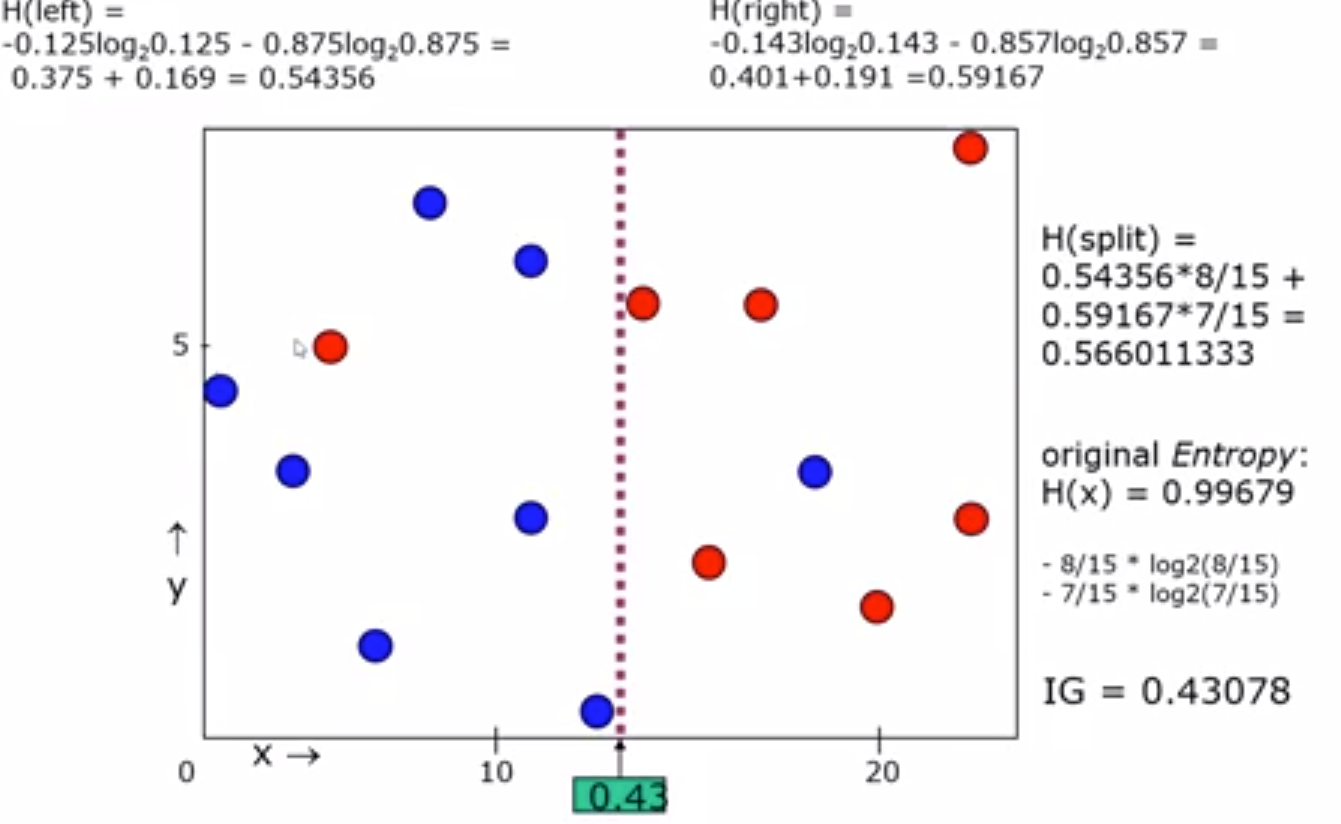
**Lecture 3: Decision Trees data types and preparation**

-DT: For each attribute identify possible splits. Compute best split, over all attributes

Error rate is computed per split. We do the splits and error rate is computed



We do on y and x axis. Every potentials split.



How many possible splits? – For each attributes number of unique values -1

Measure of uncertainty – High probability (low entropy)

The highest **information gain** for a split. 7 Blue samples.

We split only on X or Y axis. ALSO GINI INDEX

**Lecture 4:**

Impute the missing values after train test split. Data leakage – data from test set must not use in test. Our test data shouldn’t be modified.

Effectivness and efficiency – trade-off

Bias-variance, over and under fitting

Decision Trees are overfitting. To regularize it how deep it should be? For each node how many samples we will have.

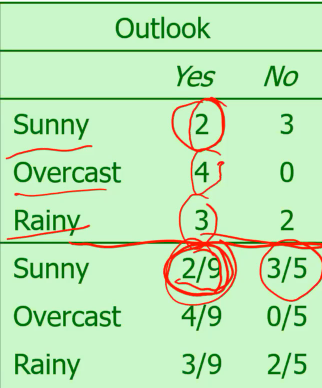
Small chance of the data will change the structure of the tree. High variance.

Random forest – taking random sample, random attributes from dataset. Build small DT algos.

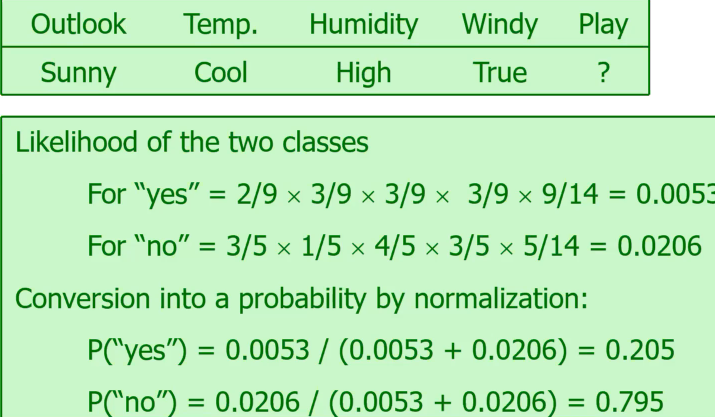
**Lecture 5: Naive Bayes, Covering Algorithms**

Naïve Bayes – Opposite of 1 Root tree. Assumptions are:

* Equally important attributes
* Statistically independent (independence Is never the case)

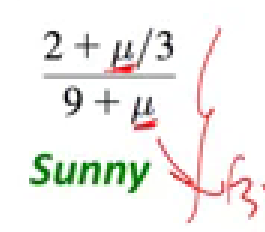


Probability – of yes for sunny is 2 from 9 all



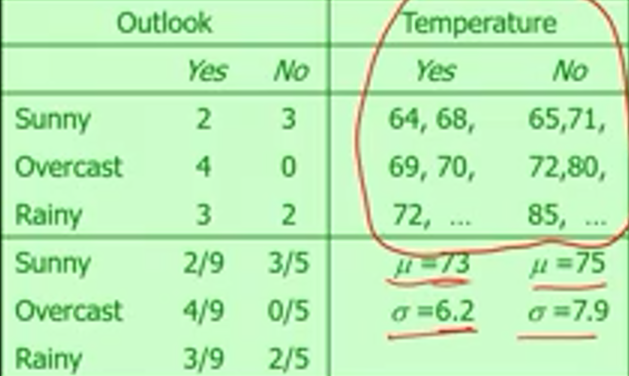
9/14 is the class yes, 5/14 class no

Zero-freq problem. If one attribute don’t occur once, everything will be zero. Add const to be not zero.



For numeric values, our assumption is to have normal or Gaussian distribution.

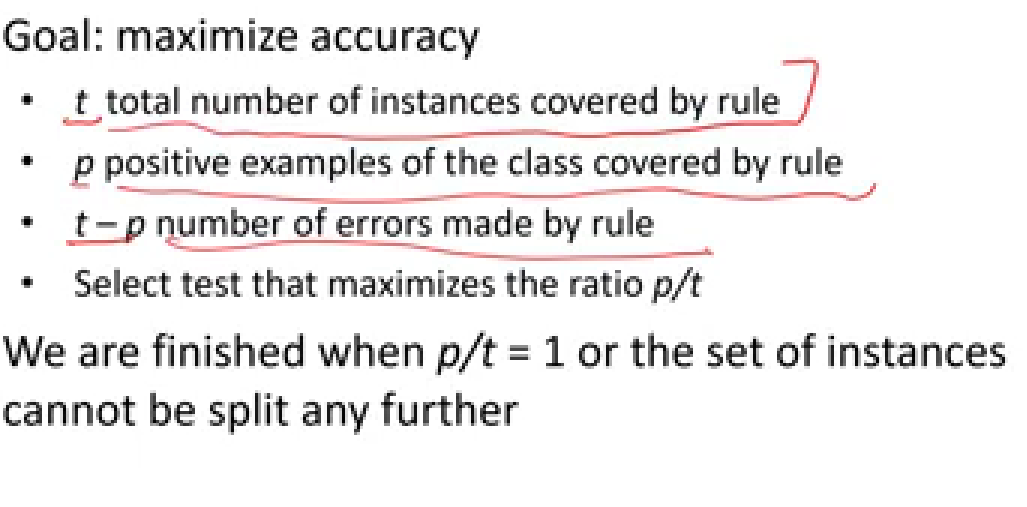
Sample mean, std,



Covering algorithms – rule based learning algorithms.

Basic idea: generate a rule by adding tests that maximize the rule’s accuracy.

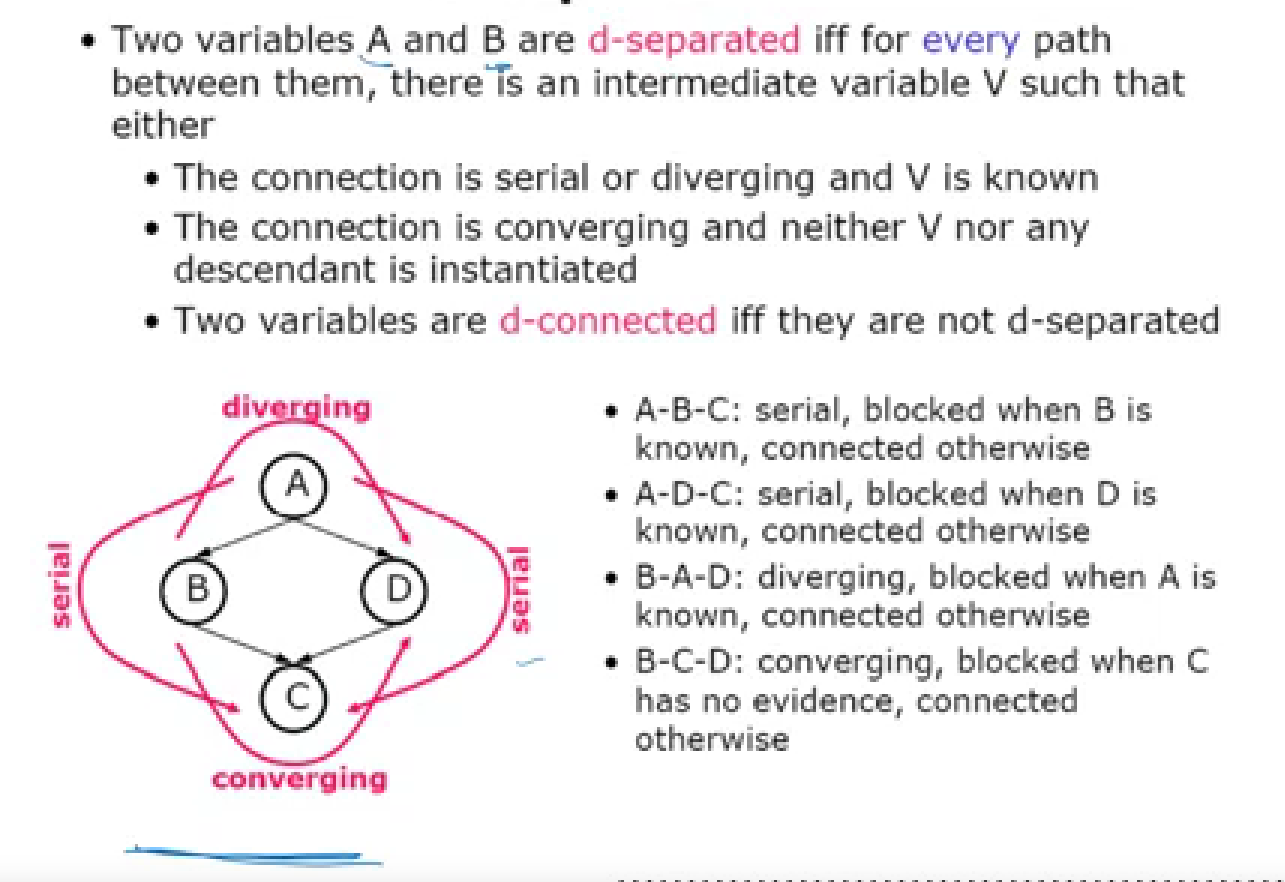
0We add conditions. If else,



**Lecture 6: Bayesian Networks**

Acyclic graph should be there. We have the probabilities for every variable (node) given its parents.

The model is connections with the probabilities.



For n variables and for each variable we have 2 probabilities we have 2power n. We will have a lot of probabilities.

D-separation – exploit independence between variables.

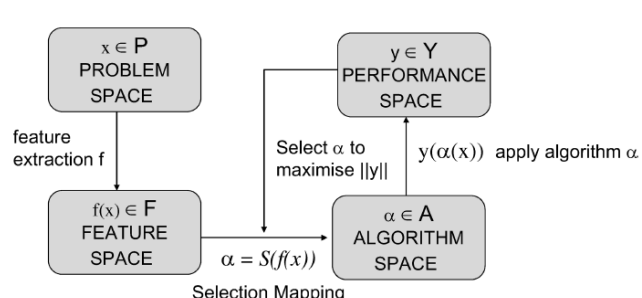
**Lecture 7: AutoML**

Compare differnet ml algos.

Metalearninig is learning about learning. Accumulating experience on specific learning task.

Accumulating experience on the performance of the machine leaning algos.

Algorithm selection with Rice framework



In P we have instances of a problem class

F – feature space that contains measurable characteristics of instances, Set A of all algos, Y is the set of performance measures (accuracy)



– Ratio of examples to attributes – Average class entropy – Degree of correlation between features and target concept