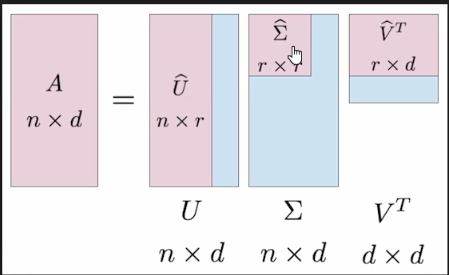
PLS Regression: Avoid multi-collinear for linear regression. Instead of removing attributes, project to other dimension. Features after projecting: Latent feature: du lieu an, cannot have multi-collinear

PCA: explanation ratio

PCR:

SVD: Matrix Factorization



A got multi-collinear.

d = 30000, r = 40, 30 r << d -> r: number of latent features (> 0.001 -> remove this feature). Apply model on U

Naïve Bayes: hypothesis = model

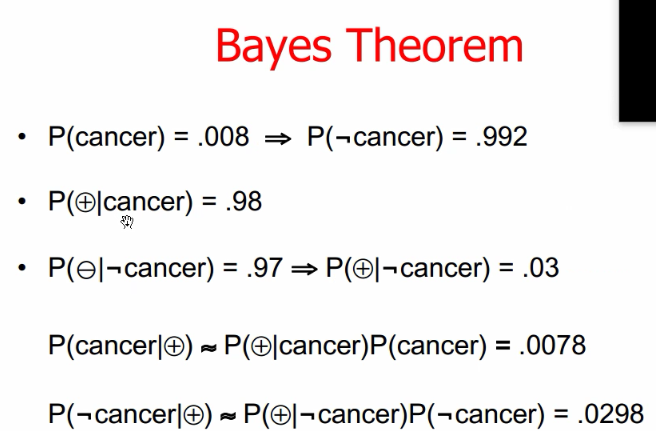
P(positive|cancer)=0.98

P(negative|non\_cancer)=0.97

P(cancer)=0.008

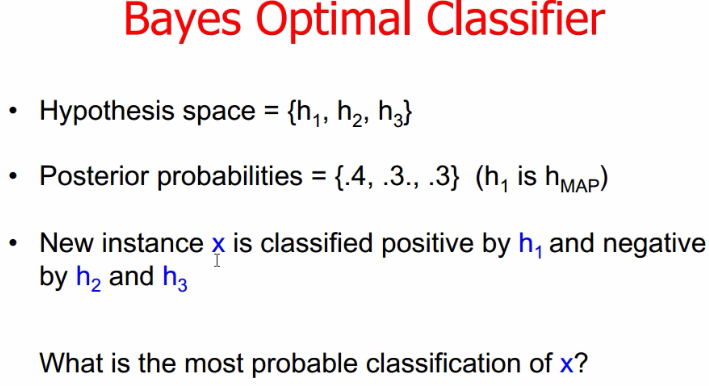
P(cancer| positive) = P(positive| cancer)\*P(cancer)/P(positive)

P(positive) is proba comes in nature you cannot have and don’t need to have



Model fit tot nhat -> hypo co proba lon nhat

Dua test set vao kiem tra



Choose theo so dong -> negative

P(c|h): proba h that tell yes/no

0\*0.4+1\*0.3+1\*0.3

1\*0.4+0\*0.3+0\*0.3

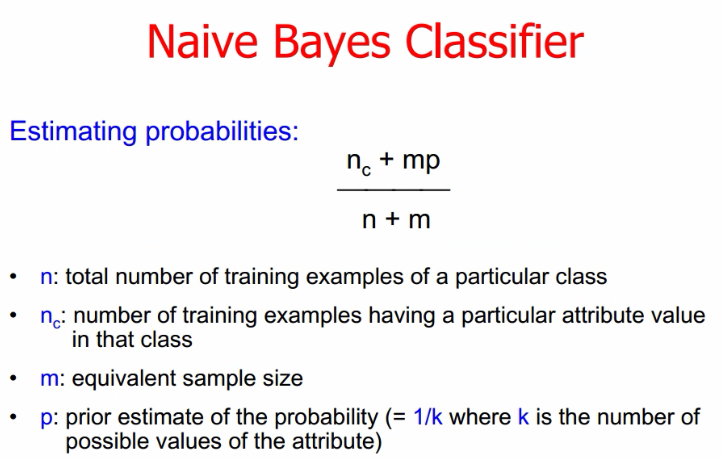
Deep learning: yoo don’t need to detect feature

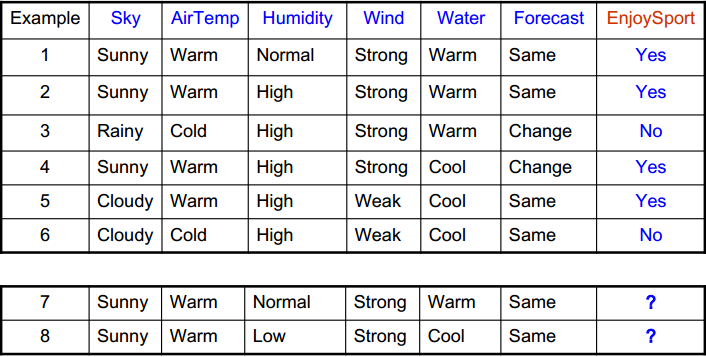
If you detect features well naïve bayes do well and you can also explain to boss why yes/no, deep learning like black box, you cannot explain

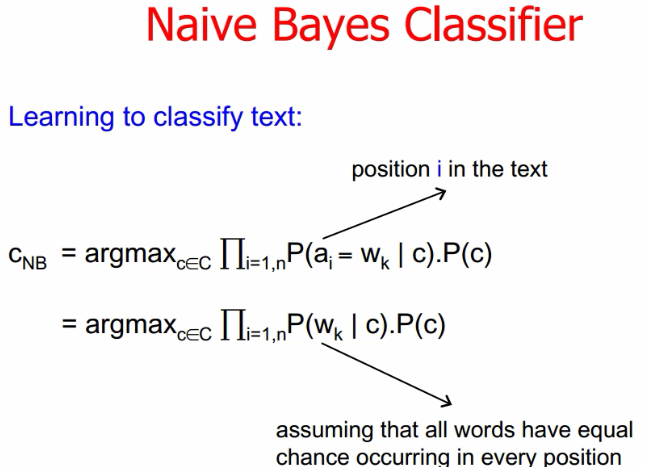
p:low 1/3

nc = 0 # of low

n: # of class ‘yes’/ ‘no’



* Given the dataset, calculate Naïve Bayes to find the class of **last row**
* P(Ci): 
* P(Sky= ‘Sunny’ | class = ‘Yes’) = 
* P(Sky= ‘Sunny’ | class = ‘No’) = 0
* P(Sky= ‘Rainy’| class= ‘Yes’)=0
* P(Sky= ‘Rainy’| class= ‘No’)= 1
* Notice: value ‘Low’ is not in ‘Humidity’, so P(Humidity= ‘Low’| class= ‘Yes’) = P(Humidity= ‘Low’| class= ‘No’)=0 → Cannot decide → Apply Estimating Probabilities
* P(Humidity= ‘Low’| class= ‘Yes’)= 
* P(Humidity= ‘Low’| class= ‘No’)=
* Put X = (Sunny, Warm, Low, Strong, Cool, Same)
* P(X | Class = ‘Yes’) = P(Sky= ‘Sunny’ | class = ‘Yes’) \* P(AirTemp= ‘Warm’ | class = ‘Yes’) \* P(Humidity= ‘Low’ | class = ‘Yes’) \* P(Wind= ‘Strong’ | class = ‘Yes’) \* P(Water= ‘Cool’ | class = ‘Yes’) \* P(Forecast= ‘Same’ | class = ‘Yes’)
* P(X | Class = ‘No) = P(Sky= ‘Sunny’ | class = ‘No) \* P(AirTemp= ‘Warm’ | class = ‘No) \* P(Humidity= ‘Low’ | class = ‘No) \* P(Wind= ‘Strong’ | class = ‘No) \* P(Water= ‘Cool’ | class = ‘No) \* P(Forecast= ‘Same’ | class = ‘No)
* Compare: P(X | Class = ‘Yes’) \* P(Class = ‘Yes) and P(X | Class = ‘No’) \* P(Class = ‘No’)



ai: text at pos ai

ko context vi assumption; A loves B = B loves A. wk tan so xuat hien ai

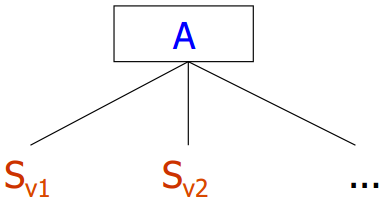
tf idf naïve bayes

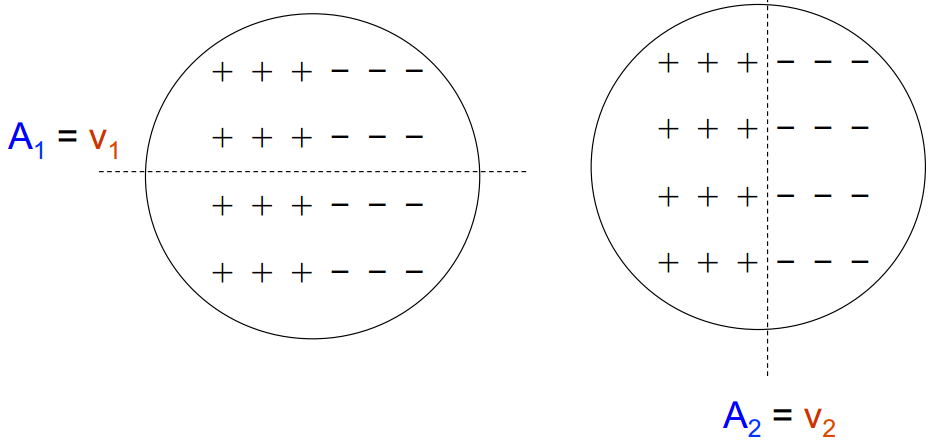
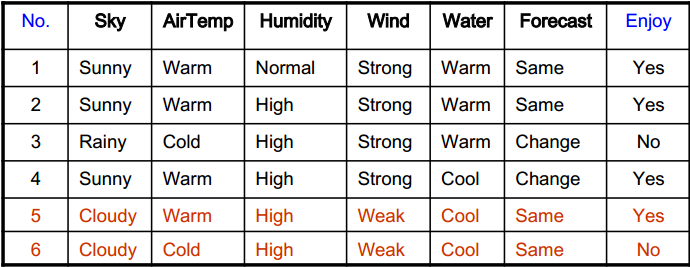
voting model – ensemble learning – random forest

# Decision Tree

ID3

* Entropy S: đo sự hỗn loạn của S (Sample).
  + E.g. If your dataset has 2 classes, Entropy(S) = 
* Gain(S, A) trong đó A là feature: lượng thông tin A chứa để giải thích cho S, so you choose feature which has maximum Gain

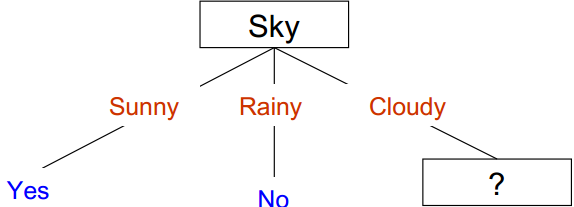
Gain(S, A) = Entropy(S) - 

* E.g.
  + Should choose A2=v2 to minimize the entropy S
* Given the dataset, calculate ID3:
  + Entropy(S) = 
  + Gain(S, Sky) 



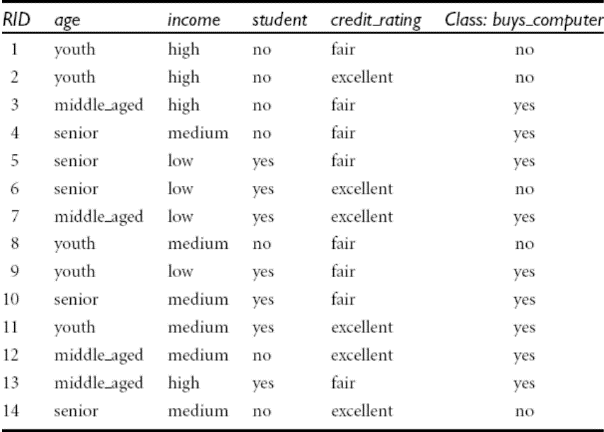
(Intuitively, you see that Entropy(SSunny)=0 because Sunny just include one class: ‘Yes’ and Entropy(SRainy)=0 because Rainy just include one class: ‘No’)



* + Do similarly for all other features and choose maximum Gain, and the largest Gain is feature Sky
  + Delete all rows which does not include Cloudy in Sky
  + 
  + Gain(SCloudy, AirTemp)



C4.5

* One weakness of ID3: if one feature in your dataset is continuous, the ID3 is highly likely to get overfitting when every edge in Decision Tree of this continuous feature will be one number
* SplitInfoAD = 
* Gain(A) = Gain(S, A)
* GainRatio(A) = 
* Choose feature max GainRatio
* Given the dataset, Calculate C4.5
  + SplitInfoIncomeS

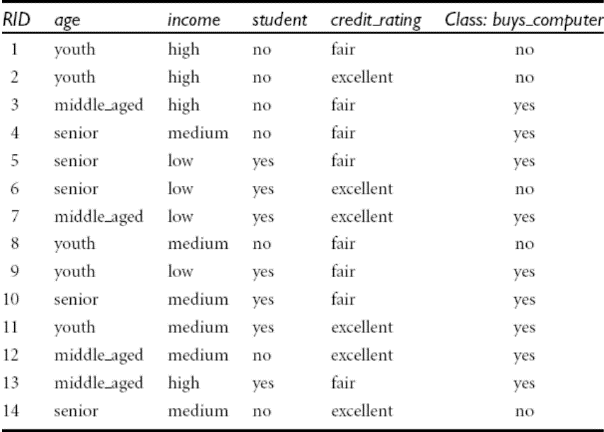


* + Gain(Income)

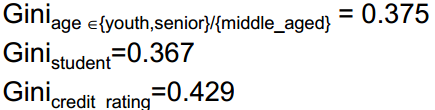


* + GainRatio(Income)=
  + Do similarly for all other features and choose feature max GainRatio

Cart

* Binary Split for feature A. S­A is subset of A which has 1 or v-1 unique values of A
* Gini(D) = 
* GiniAD = 
* ∆Gini(A) = Gini(D) - GiniAD
* Choose feature has min GiniAD or max ∆Gini(A)
* Given the dataset, Calculate C4.5
* Gini(S) = 



* Giniincome∈(Low, High) = Giniincome∈(Medium)=0.315
* Giniincome∈(Medium, High) = Giniincome∈(High)=0.3
* You choose Giniincome∈(Medium, High)=0.3
* 
* Choose Income to split, GiniAD is min

# Pruning Decision Tree

