Classification

Decision Tree

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$Gain(A) = I(p,n) - E(A)$$ E(A) →expected I

- Class P: buys_computer = "yes"
- Class N: buys_computer ="no"
- *I*(*p*, *n*) = *I*(9, 5) =0.940
- Compute the entropy for *age*:

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

$$E(age) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

Hence,

$$Gain(age) = I(p,n) - E(age) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Thus, we should select "age" as the root node of the decision tree.

2. You are working for the FRIDAY telecom company and are given some customer records. Your manager asks you to find the classification rule(s) for high and low usage customers. The data are given below.

Customer ID	Monthly Income	Age	Education	Marital Status	Usage
9100123	Low	Old	University	Married	Low
9303034	High	Young	College	Single	High
9210126	Medium	Young	College	Married	High
9142020	Medium	Old	High School	Single	Low
9910111	High	Old	University	Single	High
9576732	Low	Old	High School	Married	Low

a) Suppose you take use of the decision tree to solve the problem. What are the (theoretical) maximum and minimum depths of the tree being formed?

Ans.
For a database consisting of four feature attributes, the theoretical maximum and minimum depths of the tree being formed are 4 and 0 respectively.

b) Construct a decision tree, based on information gain, to classify customers as "high usage" and "low usage". Show your steps.

Ans.
I(p,n)=I(3,3)=1

M.Income	I(pi,ni)
Low	0
Medium	1
High	0

Age	I(pi,ni)
Old	0.81
Young	0

Education	I(pi,ni)
University	1
College	0
High School	0

Mari. Status	I(pi,ni)
Single	0.92
Married	0.92

Information_Gain(Monthly Income)=1-0.33=0.67
Information_Gain(Age)=1-0.67*0.81=0.46
Information_Gain(Education)=1-0.33=0.67
Information_Gain(Marital Status)=1-0.92=0.08

Outlook	Temperature	Humidity	Windy	Play Tennis
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

Decision Tree Process:

1. overall Entropy

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$I(3,2) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} \approx 0.971$$

2. Information Gain for different Attributes

1. Outlook

- Split into: `Sunny`, `Overcast`, `Rainy`.

Outlook	Yes	No	Entropy
Sunny	0	2	$I(0,2) = 0.0$
Overcast	1	0	$I(1,0) = 0.0$
Rainy	2	0	$I(2,0) = 0.0$

Weighted entropy:

$$E(Outlook) = \frac{2}{5}(0.0) + \frac{1}{5}(0.0) + \frac{2}{5}(0.0) = 0.0$$

- **Gain(Outlook):**

$$Gain(Outlook) = 0.971 - 0.0 = 0.971$$

3. Choose Best Attributes (Highest Gain)

4. build tree

Standard Deviation

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = P(Overcast)*S(Overcast) + P(Rainy)*S(Rainy) + P(Sunny)*S(Sunny)
= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87
= 7.66

Standard Deviation Reduction (SDR)

SDR(Hours , Outlook) = S(Hours) – S(Hours, Outlook)
= 9.32 – 7.66 = 1.66

s.d. all hour number – (expected prob.) * (sd.)

Naïve Bayes

Naïve Bayes: $P(A|B) = \frac{P(B|A)\cdot P(A)}{P(B)}$

1. Given the following table.

Parcel ID	*Origin*	*Destination*	*Type*	*Weight*
1	HK	HK	Parcel	Light
2	Kln	Kln	Letter	Light
3	NT	Kln	Letter	Light
4	HK	HK	Parcel	Heavy

P (Parcel) = 10/20
P(Letter)=10/20

Origin	
P (OHK | Parcel) = 6/10	P (OHK | Letter) = 4/10
P (OKln | Parcel) = 4/10	P (OKln | Letter) = 3/10
P (ONT | Parcel) = 0/10	P (ONT | Letter) = 3/10
Destination	
P (DHK | Parcel) = 5/10	P (DHK | Letter) = 3/10
P (DKln | Parcel) = 3/10	P (DKln | Letter) = 5/10
P (DNT | Parcel) = 2/10	P (DNT | Letter) = 2/10
Weight	
P (Light | Parcel) = 5/10	P (Light | Letter) = 10/10
P (Heavy | Parcel) = 5/10	P (Heavy| Letter) = 0/10

PID	Formula	Prediction	Prediction Correct or Not
1	P(X|Parcel)P(Parcel) = P(OHK|Parcel)* P(DHK|Parcel)* P(Light|Parcel)*P(Parcel) = 6/10*5/10*5/10*10/20 = 0.075 P(X|Letter)P(Letter)=0.06 use condition prob. of all feature	Parcel	Yes
2	P(X|Parcel)P(Parcel) = 0.03 P(X|Letter)P(Letter)=0.075	Letter	Yes
3	P(X|Parcel)P(Parcel) = 0 P(X|Letter)P(Letter)=0.075	Letter	Yes
4	P(X|Parcel)P(Parcel) = 0.075 P(X|Letter)P(Letter)=0	Parcel	Yes

KNN

$$d(i,j) = \frac{p-m}{p}$$

ID	Food	Chat	Fast	Price	Bar	BigTip
1	great	yes	yes	normal	no	yes
2	great	no	yes	normal	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	normal	yes	yes

← Stored Records For BigTip Classification

Similarity metric: Number of matching attributes
Number of nearest neighbors: k=2

Classifying new data:

- New data (x, great, no, no, normal, no)
  → most similar: ID=2 (1 mismatch, 4 match) → yes
  → Second most similar example: ID=1 (2 mismatch, 3 match) → yes
  So, classify it as "Yes".
- New data (y, mediocre, yes, no, normal, no)
  → Most similar: ID=3 (1 mismatch, 4 match) → no
  → Second most similar example: ID=1 (2 mismatch, 3 match) → yes
  So, classify it as "Yes/No"?!

Measuring Error

	Predicted class	
True Class	Yes	No
Yes	TP: True Positive	FN: False Negative
No	FP: False Positive	TN: True Negative

- Error rate = # of errors / # of instances = (FN+FP) / N
- Recall = # of found positives / # of positives
  = TP / (TP+FN) = sensitivity = hit rate
- Precision = # of found positives / # of found
  = TP / (TP+FP)
- Specificity = TN / (TN+FP)
- False alarm rate = FP / (FP+TN) = 1 - Specificity

Neural Networks – MLP/SVM/CNN/RNN…

Random Forest – bagging (select N subsets/N features)

Adaboost – boosting (retrain with different importance weighted)