DATA MINING

PERTEMUAN KE-8

Algoritma Klasifikasi

1. Algoritma Klasifikasi

   Konsep dasar klasifikasi

2. Decision Tree & Model Overfitting.

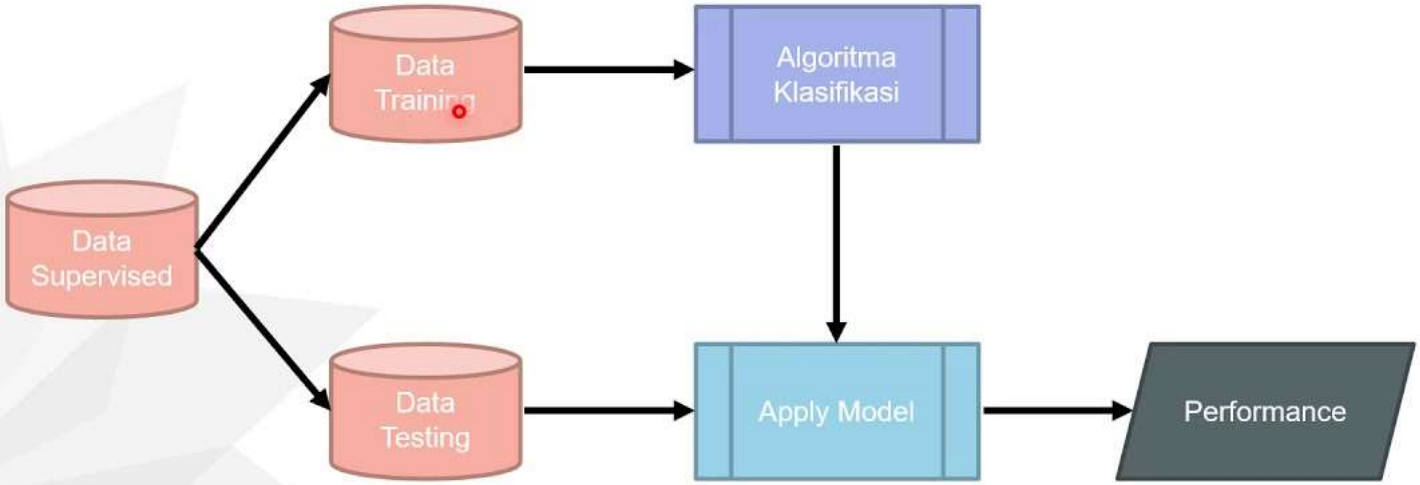
   Evaluasi Kinerja pengklasifikasi

Proses secara umum

Klasifikasi pada Data Mining

Validasi data

- **Split validation**: melakukan validasi sederhana dengan membagi dataset secara acak menjadi dua data terpisah (data latih & data uji).
- **Cross validation**: melakukan validasi berulang di mana dataset dibagi menjadi banyak subset (himpunan) data latih & validasi. Setiap iterasi memvalidasi (menguji) satu subset data dengan subset yang tersisa sebagai data latih.



1. Konsep dasar klasifikasi

- Klasifikasi disebut sebagai supervised learning karena kumpulan data contoh digunakan untuk mempelajari struktur kelompok, seperti halnya seorang guru mengawasi siswanya menuju tujuan tertentu.
- Meskipun kelompok-kelompok yang dipelajari melalui model klasifikasi mungkin sering dikaitkan dengan struktur kesamaan variabel fitur, seperti dalam pengelompokan, hal ini belum tentu demikian.

1. Konsep dasar klasifikasi

- Dalam klasifikasi, contoh data pelatihan (training data) sangat penting dalam memberikan panduan tentang bagaimana kelompok didefinisikan.
- Dengan adanya kumpulan data contoh pengujian, grup yang dibuat oleh model klasifikasi pada contoh pengujian akan mencoba mencerminkan jumlah dan struktur grup yang tersedia dalam contoh kumpulan data pelatihan.

Algoritma klasifikasi biasanya memiliki dua fase:

1. Training phase: Pada fase ini, model pelatihan dibangun dari contoh pelatihan. Secara intuitif, ini dapat dipahami sebagai ringkasan model matematika dari kelompok berlabel dalam kumpulan data pelatihan.
2. Testing phase: Dalam fase ini, model pelatihan digunakan untuk menentukan label kelas (atau pengidentifikasi grup).

Beberapa contoh Masalah klasifikasi (1)

1. **Target pemasaran pelanggan:**

Dalam hal ini, grup (atau label) sesuai dengan minat pengguna terhadap produk tertentu. Misalnya, satu kelompok mungkin berhubungan dengan pelanggan yang tertarik pada suatu produk, dan kelompok lainnya mungkin berisi pelanggan yang tersisa.

2. **Medical disease management:**:

Dalam beberapa tahun terakhir, penggunaan metode penambangan data dalam penelitian medis semakin mendapat perhatian. Fitur-fiturnya dapat diambil dari tes medis dan perawatan pasien, dan label kelas mungkin sesuai dengan hasil pengobatan.

Beberapa contoh Masalah klasifikasi (2)

3. Kategorisasi dan pemfilteran dokumen: Banyak aplikasi, seperti layanan kantor berita, memerlukan klasifikasi dokumen secara real-time. Ini digunakan untuk mengatur dokumen berdasarkan topik tertentu di portal Web. Contoh dokumen sebelumnya dari setiap topik mungkin tersedia. Fitur-fiturnya sesuai dengan kata-kata dalam dokumen. Label kelas sesuai dengan berbagai topik, seperti politik, olahraga, peristiwa terkini, dan sebagainya.

Beberapa contoh Masalah klasifikasi (3)

4. Analisis data multimedia: Seringkali diinginkan untuk melakukan klasifikasi data multimedia dalam jumlah besar seperti foto, video, audio, atau data multimedia lain yang lebih kompleks. Contoh sebelumnya mengenai aktivitas tertentu pengguna yang terkait dengan contoh video mungkin tersedia. Ini dapat digunakan untuk menentukan apakah video tertentu menggambarkan aktivitas tertentu.

Keluaran klasifikasi

Output dari algoritma klasifikasi dapat berupa salah satu dari dua jenis:

1. Prediksi label: Dalam hal ini, label diprediksi untuk setiap contoh pengujian.
2. Skor numerik:

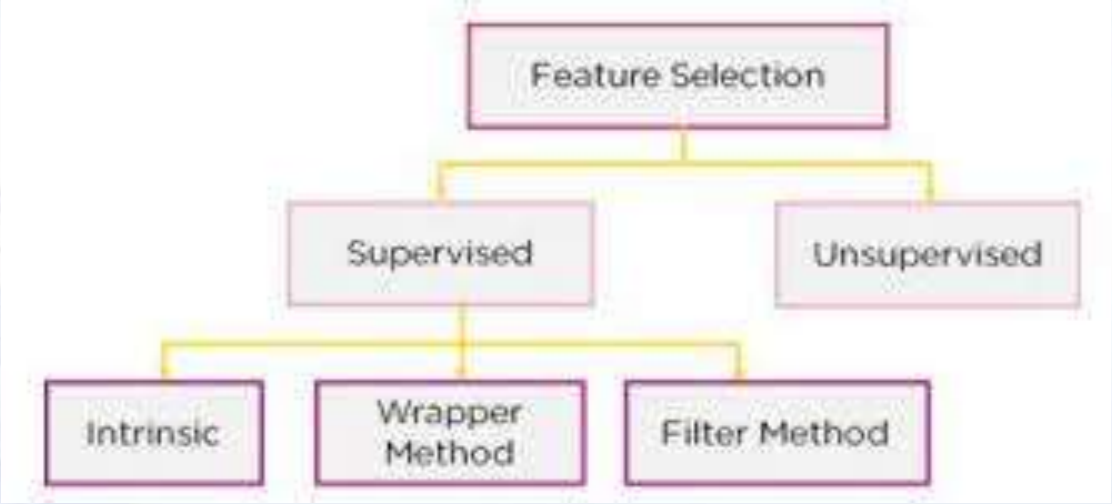
- Dalam kebanyakan kasus, pembelajar memberikan skor pada setiap kombinasi instance-label yang mengukur kecenderungan instance tersebut untuk menjadi bagian dari kelas tertentu. Skor ini dapat dengan mudah dikonversi ke prediksi label dengan menggunakan nilai maksimum, atau nilai maksimum tertimbang biaya dari skor numerik di berbagai kelas.

Seleksi Fitur untuk Klasifikasi

- Seleksi fitur merupakan tahap pertama dalam proses klasifikasi.
- Data nyata mungkin berisi fitur dengan relevansi yang berbeda-beda untuk memprediksi label kelas. Misalnya, jenis kelamin seseorang kurang relevan untuk memprediksi label penyakit seperti “diabetes”, dibandingkan dengan usianya.

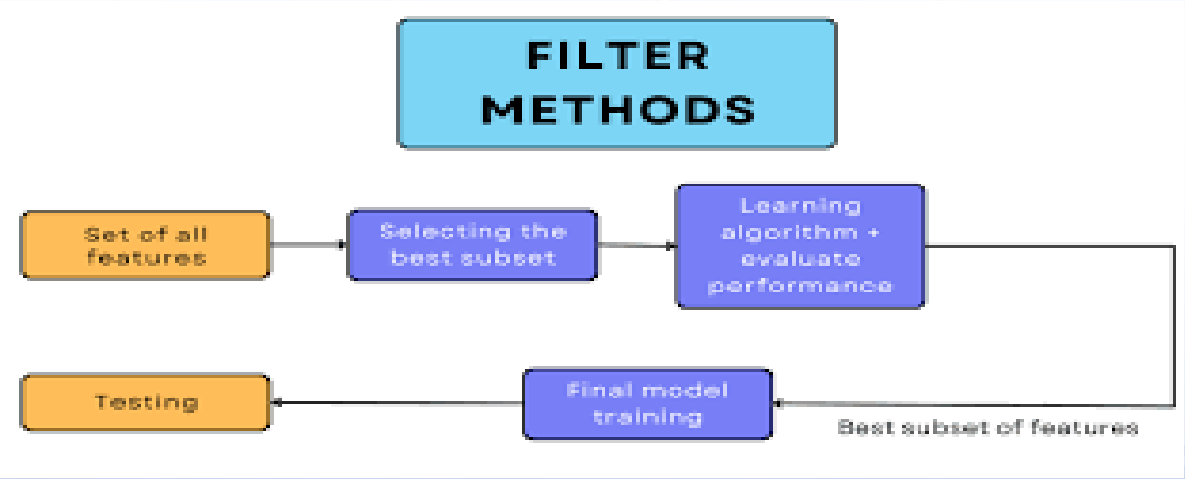
Tiga jenis metode utama digunakan untuk pemilihan fitur dalam klasifikasi

1. *Filter models*
2. *Wrapper models*
3. *Embedded models*



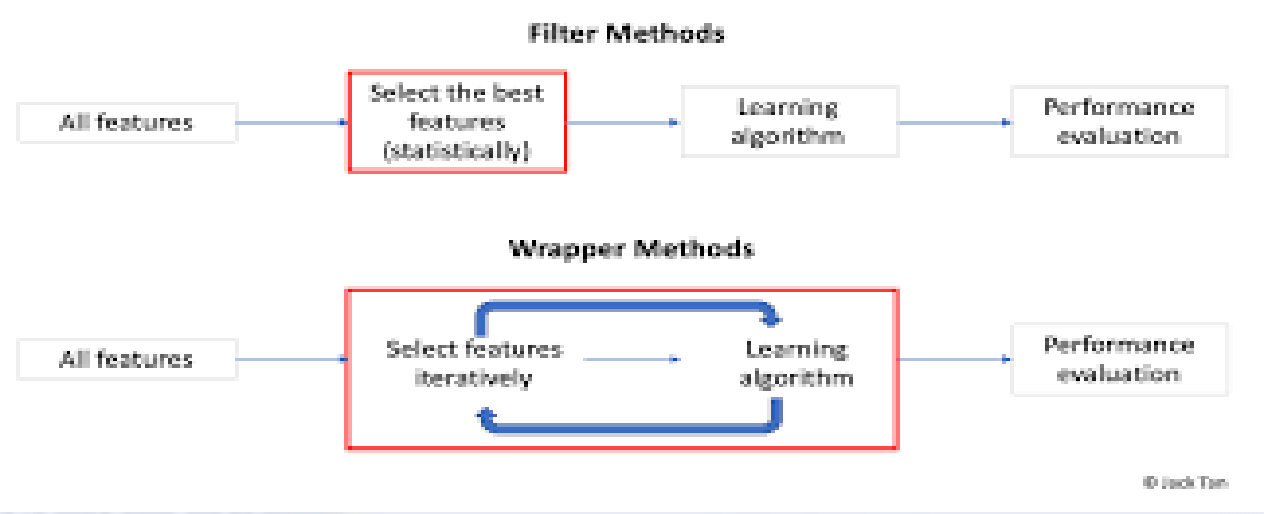
1). Filter models:

- Kriteria matematis yang jelas tersedia untuk mengevaluasi kualitas suatu fitur atau subkumpulan fitur.
- Kriteria ini kemudian digunakan untuk menyaring fitur-fitur yang tidak relevan.



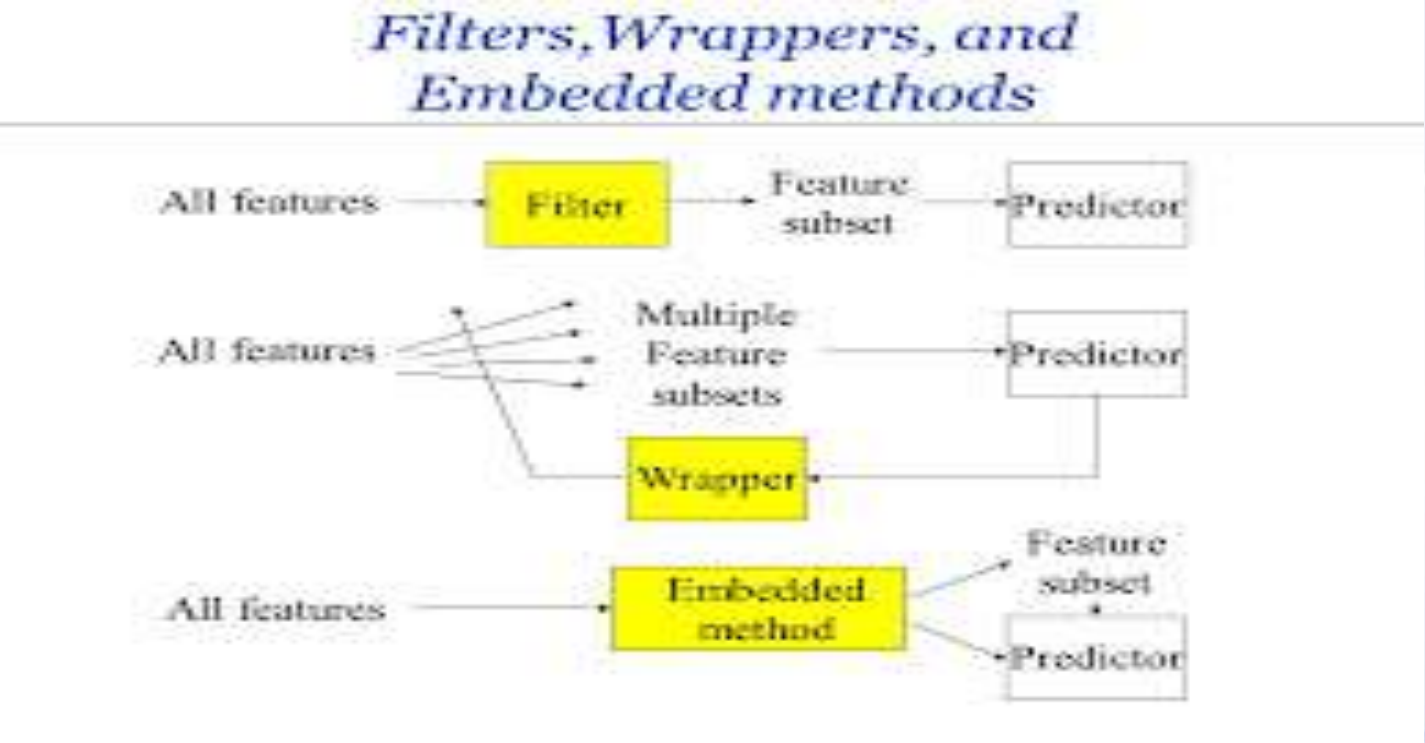
2. Wrapper models:

- Diasumsikan bahwa algoritma klasifikasi tersedia untuk mengevaluasi seberapa baik kinerja algoritma dengan subset fitur tertentu.
- Algoritma pencarian fitur kemudian digabungkan dengan algoritma ini untuk menentukan kumpulan fitur yang relevan.



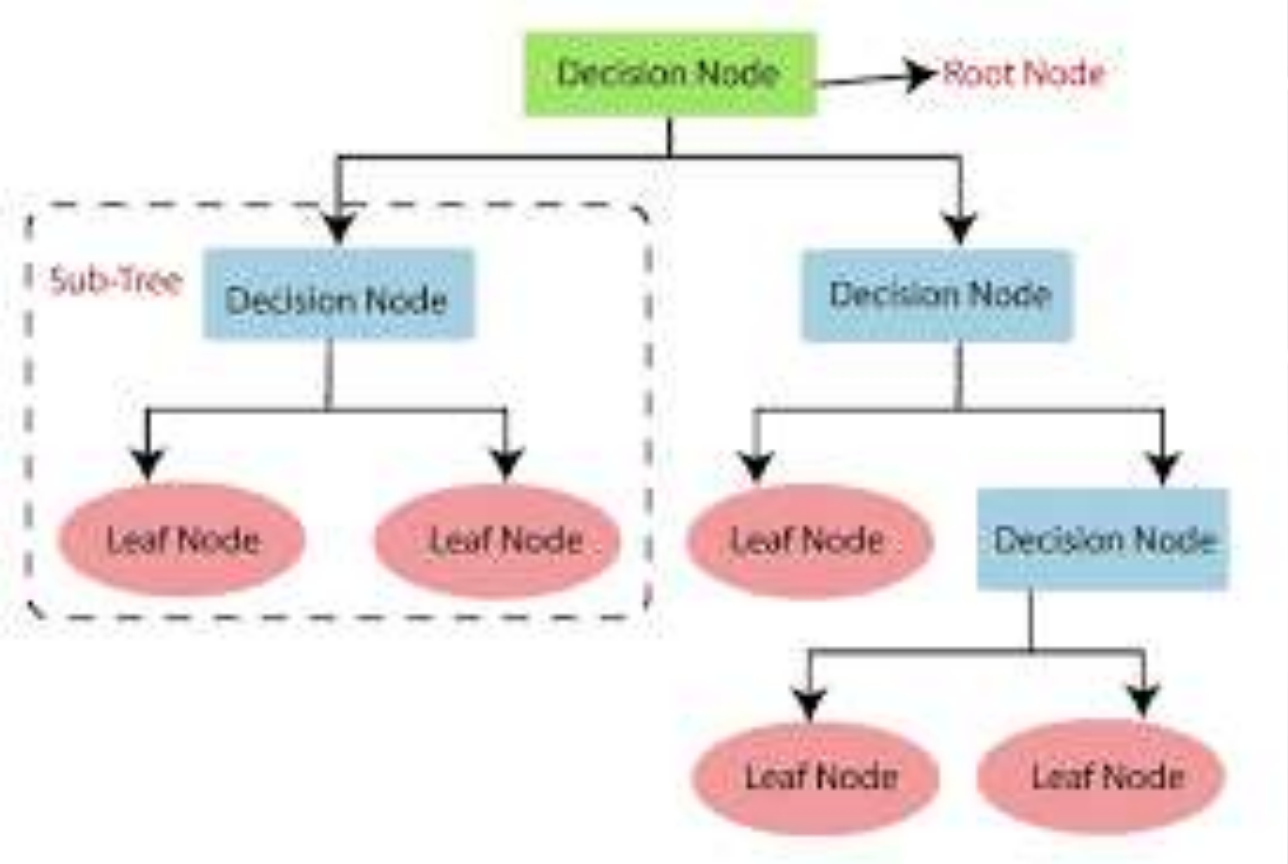
3). Embedded models:

- Solusi terhadap model klasifikasi sering kali berisi petunjuk berguna tentang fitur yang paling relevan.
- Fitur-fitur tersebut diisolasi, dan pengklasifikasinya dilatih ulang berdasarkan fitur-fitur yang telah dipangkas.



Algoritma Klasifikasi

Decision Tree & Model Overfitting.



Decision Tree
Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)

1. Pohon dibangun dengan cara membagi-dan-menaklukkan secara rekursif dari atas ke bawah
2. Pada awalnya, semua contoh pelatihan berada pada akarnya
3. Atribut bersifat kategoris (jika dinilai berkelanjutan, atribut tersebut akan didiskritisasi terlebih dahulu)
4. Contohnya dipartisi secara rekursif berdasarkan atribut yang dipilih
5. Atribut pengujian dipilih berdasarkan ukuran heuristik atau statistik (misalnya, perolehan informasi, rasio perolehan, indeks gini)

Decision Tree
Algorithm for Decision Tree Induction

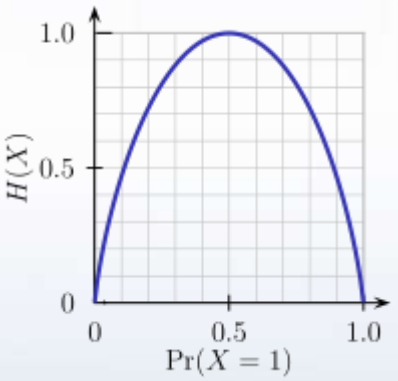
- Conditions for stopping partitioning
  - Semua sampel untuk node tertentu termasuk dalam kelas yang sama
  - Tidak ada atribut yang tersisa untuk partisi lebih lanjut – suara mayoritas digunakan untuk mengklasifikasikan daun
  - Tidak ada sampel yang tersisa

Brief Review of Entropy

- Pohon keputusan menggunakan entropi untuk memilih fitur terbaik untuk membagi data.
- Tujuannya adalah untuk membagi data sedemikian rupa sehingga di setiap cabang pohon keputusan, keberagaman atau ketidakpastian kelas yang diperkirakan seminimal mungkin.
- Pemilihan fitur yang menghasilkan pemisahan data dengan entropi yang lebih rendah dianggap lebih baik karena hal itu menunjukkan pemisahan kelas yang lebih baik.
- Dengan cara ini, entropi digunakan sebagai kriteria untuk mengukur keefektifan suatu pemilihan fitur dalam membuat model yang lebih baik untuk data mining.

Entropy

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random variable
  - Calculation: For a discrete random variable *Y* taking *m* distinct values $\{y_1, \ldots, y_m\}$,
    - $H(Y) = -\sum_{i=1}^{m} p_i \log(p_i)$, where $p_i = P(Y = y_i)$
  - Interpretation:
    - Higher entropy => higher uncertainty
    - Lower entropy => lower uncertainty
- Conditional Entropy
  - $H(Y|X) = \sum_x p(x) H(Y|X = x)$



m = 2

Attribute Selection Measure: Information Gain (ID3)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_i, D|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

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

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

Computing Information-Gain for Continuous-Valued Attributes

- Let attribute *A* be a continuous-valued attribute
- Must determine the best split point for *A*
  - Sort the value *A* in increasing order
  - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
  - The point with the *minimum expected information requirement* for *A* is selected as the split-point for *A*
- Split:
  - D_1 is the set of tuples in *D* satisfying $A \leq$ split-point, and D_2 is the set of tuples in *D* satisfying $A >$ split-point

Tahapan Algoritma Decision Tree (ID3)

1. Siapkan data training
2. Pilih atribut sebagai akar

$$Entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi$$

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

3. Buat cabang untuk tiap-tiap nilai
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yg sama

I.25

1. Siapkan data training

No	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Cloudy	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	Yes
7	Cloudy	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Cloudy	Mild	High	TRUE	Yes
13	Cloudy	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

2. Pilih atribut sebagai akar

- Untuk memilih atribut akar, didasarkan pada nilai Gain tertinggi dari atribut-atribut yang ada. Untuk mendapatkan nilai Gain, harus ditentukan terlebih dahulu nilai Entropy

- Rumus Entropy:

$$Entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi$$

  - S = Himpunan Kasus
  - n = Jumlah Partisi S
  - pi = Proporsi dari Si terhadap S

- Rumus Gain:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

  - S = Himpunan Kasus
  - A = Atribut
  - n = Jumlah Partisi Atribut A
  - | Si | = Jumlah Kasus pada partisi ke-i
  - | S | = Jumlah Kasus dalam S

I.27

Perhitungan Entropy dan Gain Akar

NODE			Jml Kasus (S)	Tidak (S_1)	Ya (S_2)	Entropy	Gain
1	TOTAL						
	OUTLOOK						
		CLOUDY					
		RAINY					
		SUNNY					
	TEMPERATURE						
		COOL					
		HOT					
		MILD					
	HUMIDITY						
		HIGH					
		NORMAL					
	WINDY						
		FALSE					
		TRUE					

Penghitungan Entropy Akar

- Entropy Total

$$Entropy(Total) = (-\frac{4}{14} * \log_2(\frac{4}{14})) + (-\frac{10}{14} * \log_2(\frac{10}{14}))$$

$$Entropy(Total) = 0.863120569$$

- Entropy (Outlook)

$$Entropy(Cloudy) = (-\frac{0}{4} * \log_2(\frac{0}{4})) + (-\frac{4}{4} * \log_2(\frac{4}{4})) = 0.000000000$$

$$Entropy(Rainy) = (-\frac{1}{5} * \log_2(\frac{1}{5})) + (-\frac{4}{5} * \log_2(\frac{4}{5})) = 0.721928095$$

$$Entropy(Sunny) = (-\frac{3}{5} * \log_2(\frac{3}{5})) + (-\frac{2}{5} * \log_2(\frac{2}{5})) = 0.970950594$$

- Entropy (Temperature)

$$Entropy(Cool) = (-\frac{0}{4} * \log_2(\frac{0}{4})) + (-\frac{4}{4} * \log_2(\frac{4}{4})) = 0.000000000$$

$$Entropy(Hot) = (-\frac{2}{4} * \log_2(\frac{2}{4})) + (-\frac{2}{4} * \log_2(\frac{2}{4})) = 1.000000000$$

$$Entropy(Mild) = (-\frac{2}{6} * \log_2(\frac{2}{6})) + (-\frac{4}{6} * \log_2(\frac{4}{6})) = 0.918295834$$

- Entropy (Humidity)

$$Entropy(High) = (-\frac{4}{7} * \log_2(\frac{4}{7})) + (-\frac{3}{7} * \log_2(\frac{3}{7})) = 0.985228136$$

$$Entropy(Normal) = (-\frac{0}{7} * \log_2(\frac{0}{7})) + (-\frac{7}{7} * \log_2(\frac{7}{7})) = 0.000000000$$

- Entropy (Windy)

$$Entropy(False) = (-\frac{2}{8} * \log_2(\frac{2}{8})) + (-\frac{6}{8} * \log_2(\frac{6}{8})) = 0.811278124$$

$$Entropy(True) = (-\frac{4}{6} * \log_2(\frac{4}{6})) + (-\frac{2}{6} * \log_2(\frac{2}{6})) = 0.918295834$$

Penghitungan Entropy Akar

NODE	ATRIBUT		JML KASUS (S)	YA (Si)	TIDAK (Si)	ENTROPY	GAIN
1	**TOTAL**		14	10	4	0,86312	
	OUTLOOK						
		CLOUDY	4	4	0	0	
		RAINY	5	4	1	0,72193	
		SUNNY	5	2	3	0,97095	
	TEMPERATURE						
		COOL	4	0	4	0	
		HOT	4	2	2	1	
		MILD	6	2	4	0,91830	
	HUMADITY						
		HIGH	7	4	3	0,98523	
		NORMAL	7	7	0	0	
	WINDY						
		FALSE	8	2	6	0,81128	
		TRUE	6	4	2	0,91830	

Penghitungan Gain Akar

$$Gain(Total, Outlook) = Entropy(Total) - \sum_{i=1}^{n} \frac{|Outlook_i|}{|Total|} * Entropy\ (Outlook_i)$$

$$Gain(Total, Outlook) = 0.863120569 - \left(\left(\frac{4}{14} * 0.000000000\right) + \left(\frac{5}{14} * 0.721928095\right) + \left(\frac{5}{14} * 0.970950594\right)\right)$$

$$Gain(Total, Outlook) = 0.258521037$$

$$Gain(Total, Temperature) = Entropy(Total) - \sum_{i=1}^{n} \frac{|Temperature_i|}{|Total|} * Entropy\ (Temperature_i)$$

$$Gain(Total, Temperature) = 0.863120569 - \left(\left(\frac{4}{14} * 0.000000000\right) + \left(\frac{4}{14} * 1.000000000\right) + \left(\frac{6}{14} * 0.918295834\right)\right)$$

$$Gain(Total, Temperature) = 0.183850925$$

$$Gain(Total, Humidity) = Entropy(Total) - \sum_{i=1}^{n} \frac{|Humidity_i|}{|Total|} * Entropy\ (Humidity_i)$$

$$Gain(Total, Humidity) = 0.863120569 - \left(\left(\frac{7}{14} * 0.985228136\right) + \left(\frac{7}{14} * 0.000000000\right)\right)$$

$$Gain(Total, Humidity) = 0.370506501$$

$$Gain(Total, Windy) = Entropy(Total) - \sum_{i=1}^{n} \frac{|Windy_i|}{|Total|} * Entropy\ (Windy_i)$$

$$Gain(Total, Windy) = 0.863120569 - \left(\left(\frac{8}{14} * 0.811278124\right) + \left(\frac{6}{14} * 0.918295834\right)\right)$$

$$Gain(Total, Windy) = 0.005977711$$

I.31

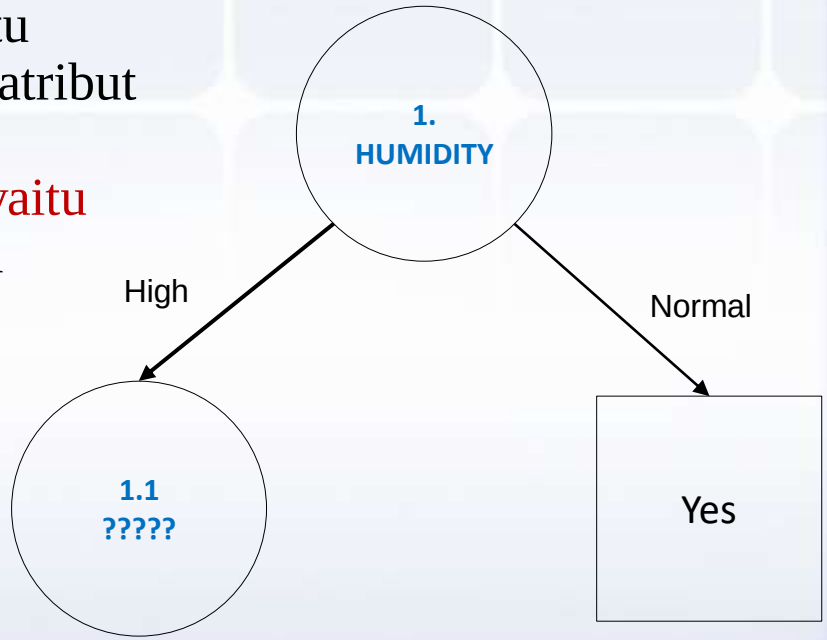
Penghitungan Gain Akar

NODE	ATRIBUT		JML KASUS (S)	YA (Si)	TIDAK (Si)	ENTROPY	GAIN
1	**TOTAL**		14	10	4	0,86312	
	OUTLOOK						0,25852
		CLOUDY	4	4	0	0	
		RAINY	5	4	1	0,72193	
		SUNNY	5	2	3	0,97095	
	TEMPERATURE						0,18385
		COOL	4	0	4	0	
		HOT	4	2	2	1	
		MILD	6	2	4	0,91830	
	HUMADITY						0,37051
		HIGH	7	4	3	0,98523	
		NORMAL	7	7	0	0	
	WINDY						0,00598
		FALSE	8	2	6	0,81128	
		TRUE	6	4	2	0,91830	

Gain Tertinggi Sebagai Akar

- Dari hasil pada Node 1, dapat diketahui bahwa atribut dengan Gain tertinggi adalah HUMIDITY yaitu sebesar 0.37051
  - Dengan demikian HUMIDITY dapat menjadi node akar

- Ada 2 nilai atribut dari HUMIDITY yaitu HIGH dan NORMAL. Dari kedua nilai atribut tersebut, nilai atribut NORMAL sudah mengklasifikasikan kasus menjadi 1 yaitu keputusan-nya Yes, sehingga tidak perlu dilakukan perhitungan lebih lanjut
  - Tetapi untuk nilai atribut HIGH masih perlu dilakukan perhitungan lagi



Covariance (Numeric Data)

- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

Correlation coefficient: $r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B

- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values
- **Negative covariance**: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value
- **Independence**: $Cov_{A,B} = 0$ but the converse is not true:
  - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

I.34 2. Buat cabang untuk tiap-tiap nilai

- Untuk memudahkan, dataset di filter dengan mengambil data yang memiliki kelembaban HUMADITY=HIGH untuk membuat table Node 1.1

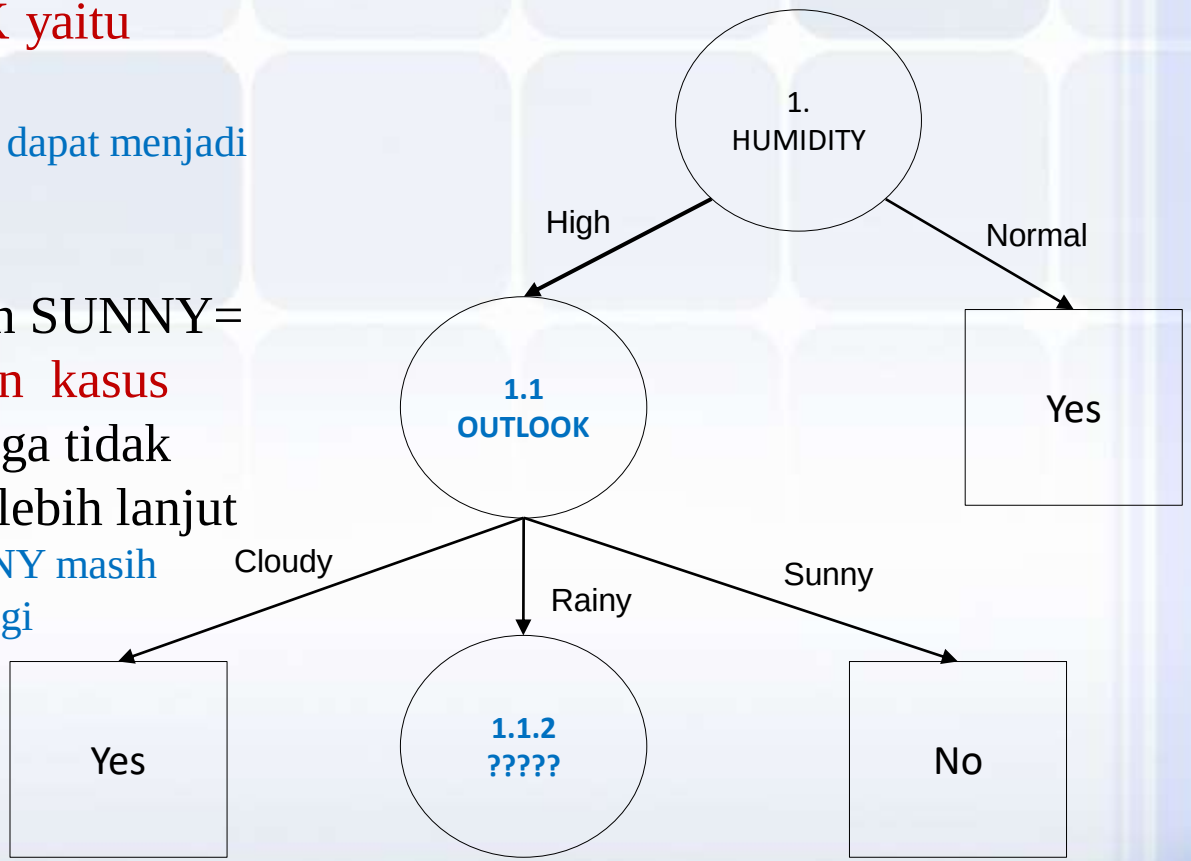
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Cloudy	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Sunny	Mild	High	FALSE	No
Cloudy	Mild	High	TRUE	Yes
Rainy	Mild	High	TRUE	No

I.35 Perhitungan Entropi Dan Gain Cabang

NODE	ATRIBUT		JML KASUS (S)	YA (Si)	TIDAK (Si)	ENTROPY	GAIN
1.1	**HUMADITY**		7	3	4	0,98523	
	OUTLOOK						0,69951
		CLOUDY	2	2	0	0	
		RAINY	2	1	1	1	
		SUNNY	3	0	3	0	
	TEMPERATURE						0,02024
		COOL	0	0	0	0	
		HOT	3	1	2	0,91830	
		MILD	4	2	2	1	
	WINDY						0,02024
		FALSE	4	2	2	1	
		TRUE	3	1	2	0,91830	

Gain Tertinggi Sebagai Node 1.1

- Dari hasil pada Tabel Node 1.1, dapat diketahui bahwa atribut dengan Gain tertinggi adalah OUTLOOK yaitu sebesar 0.69951
  - Dengan demikian OUTLOOK dapat menjadi node kedua
- Artibut CLOUDY = YES dan SUNNY= NO sudah mengklasifikasikan kasus menjadi 1 keputusan, sehingga tidak perlu dilakukan perhitungan lebih lanjut
  - Tetapi untuk nilai atribut RAINY masih perlu dilakukan perhitungan lagi



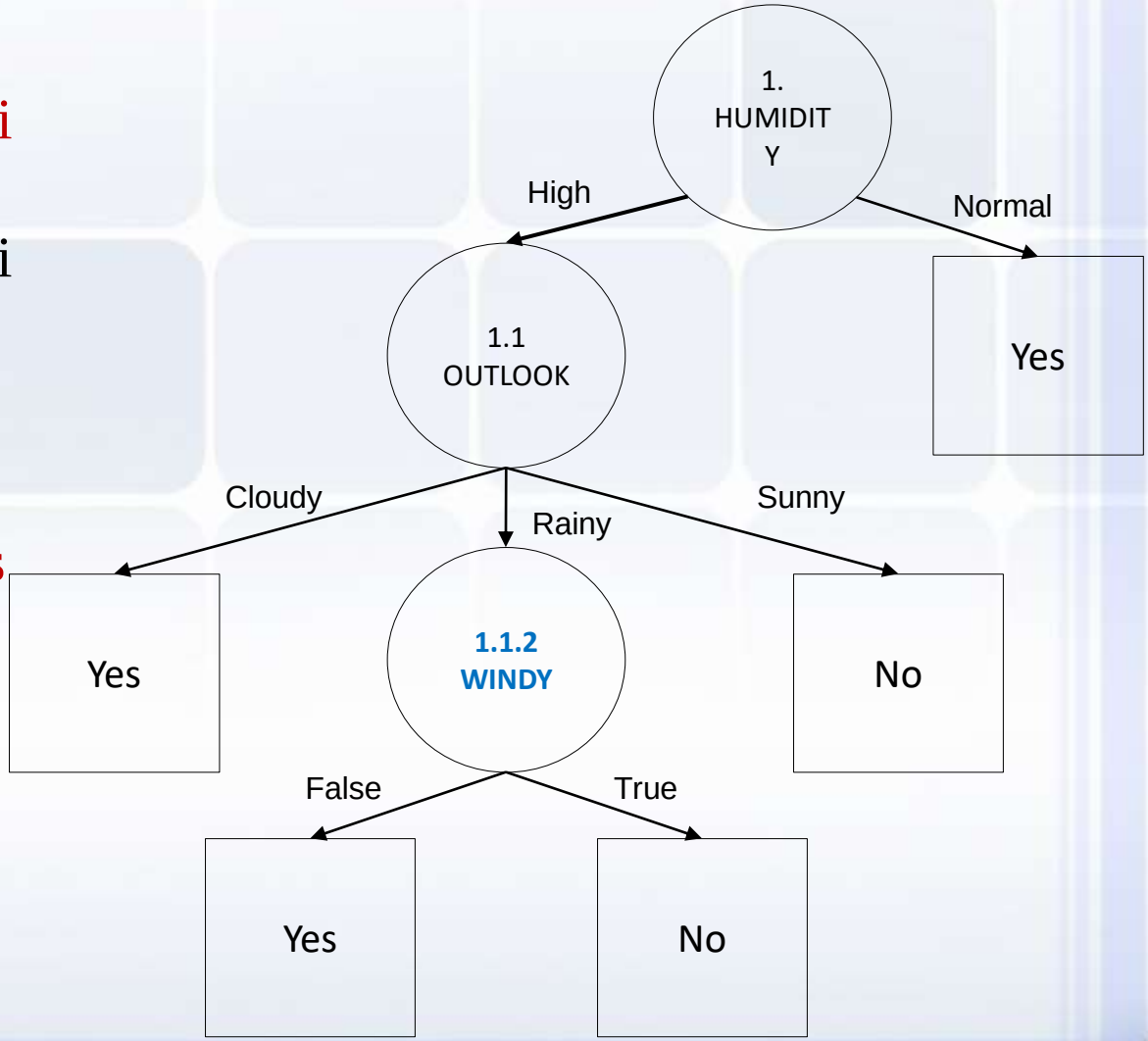
I.37 3. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yg sama

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
Rainy	Mild	High	FALSE	Yes
Rainy	Mild	High	TRUE	No

NODE	ATRIBUT		JML KASUS (S)	YA (Si)	TIDAK (Si)	ENTROPY	GAIN
1.2	HUMADITY HIGH & OUTLOOK RAINY		2	1	1	1	
	TEMPERATURE						0
		COOL	0	0	0	0	
		HOT	0	0	0	0	
		MILD	2	1	1	1	
	WINDY						1
		FALSE	1	1	0	0	
		TRUE	1	0	1	0	

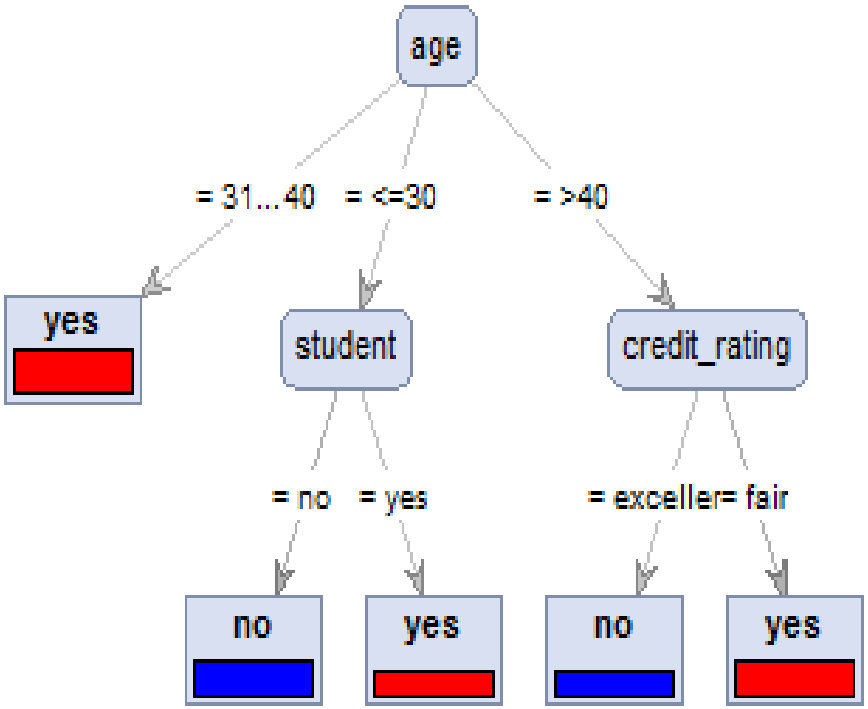
I.38 Gain Tertinggi Sebagai Node 1.1.2

- Dari tabel, Gain Tertinggi adalah WINDY dan menjadi node cabang dari atribut RAINY
- Karena semua kasus sudah masuk dalam kelas
  - Jadi, pohon keputusan pada Gambar merupakan pohon keputusan terakhir yang terbentuk



Decision Tree Induction: An Example

- Training data set: Buys_computer



age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Gain Ratio for Attribute Selection (C4.5)

- Ukuran perolehan informasi bias terhadap atribut dengan jumlah nilai yang besar
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

  - GainRatio(A) = Gain(A)/SplitInfo(A)

- Ex.

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

  - gain_ratio(income) = 0.029/1.557 = 0.019

- The attribute with the maximum gain ratio is selected as the splitting attribute

Gini Index (CART)

- If a data set D contains examples from n classes, gini index, $gini(D)$ is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class j in D

- If a data set D is split on A into two subsets D_1 and D_2, the gini index $gini(D)$ is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Computation of Gini Index

- Ex. D has 9 tuples in buys_computer = "yes" and 5 in "no"

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D_1: *{low, medium}* and 4 in D_2

$$gini_{income \in \{low,medium\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$$

$$= 0.443$$

$$= Gini_{income \in \{high\}}(D).$$

$Gini_{\{low,high\}}$ is 0.458; $Gini_{\{medium,high\}}$ is 0.450. Thus, split on the {low,medium} (and {high}) since it has the lowest Gini index

- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Comparing Attribute Selection Measures

The three measures, in general, return good results but

- Information gain:
  - biased towards multivalued attributes
- Gain ratio:
  - tends to prefer unbalanced splits in which one partition is much smaller than the others
- Gini index:
  - biased to multivalued attributes
  - has difficulty when # of classes is large
  - tends to favor tests that result in equal-sized partitions and purity in both partitions

Other Attribute Selection Measures

- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistic: has a close approximation to χ^2 distribution
- MDL (Minimal Description Length) principle (i.e., the simplest solution is preferred):
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear comb. of attrs.
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

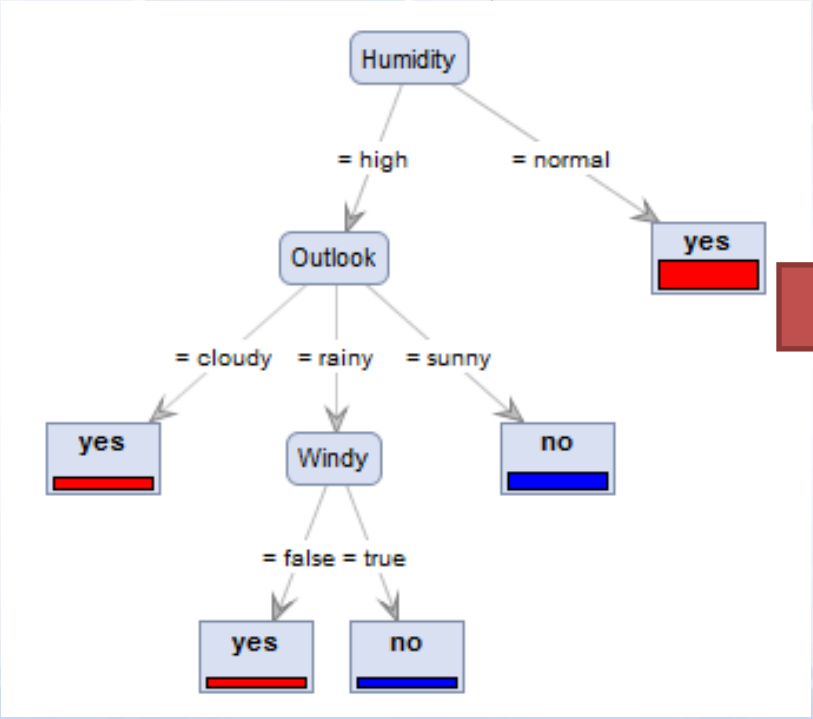
Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples

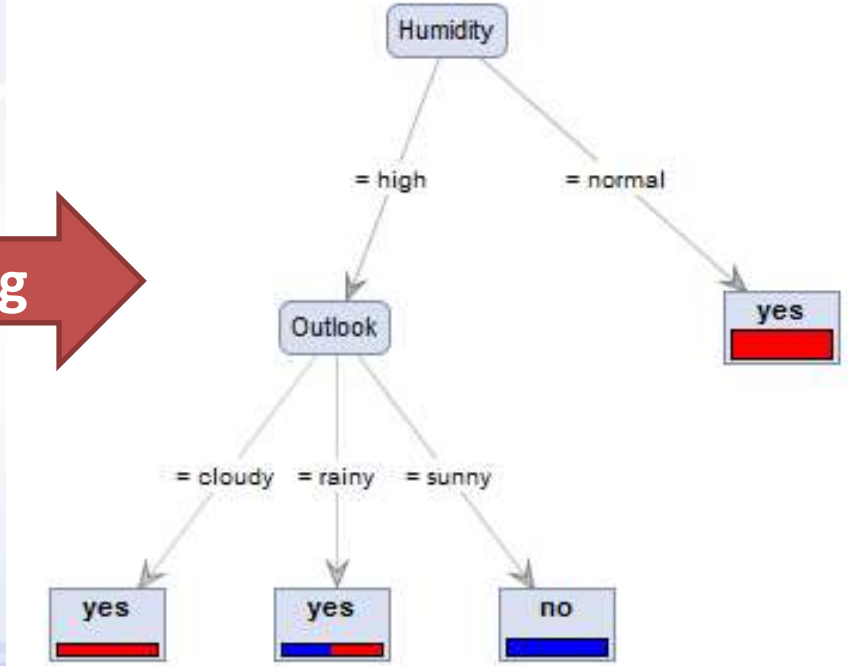
- Two approaches to avoid overfitting
  1. Prepruning: *Halt tree construction early* - do not split a node if this would result in the goodness measure falling below a threshold
     - Difficult to choose an appropriate threshold
  2. Postpruning: *Remove branches from a "fully grown" tree* -get a sequence of progressively pruned trees
     - Use a set of data different from the training data to decide which is the "best pruned tree"

I.46

Row No.	Play	Outlook	Temperature	Humidity	Windy
1	no	sunny	hot	high	false
2	no	sunny	hot	high	true
3	yes	cloudy	hot	high	false
4	yes	rainy	mild	high	false
5	yes	rainy	cool	normal	false
6	yes	rainy	cool	normal	true
7	yes	cloudy	cool	normal	true
8	no	sunny	mild	high	false
9	yes	sunny	cool	normal	false
10	yes	rainy	mild	normal	false
11	yes	sunny	mild	normal	true
12	yes	cloudy	mild	high	true
13	yes	cloudy	hot	normal	false
14	no	rainy	mild	high	true



Pruning



Why is decision tree induction popular?

- Mudah dan cepat dipelajari (dibandingkan metode klasifikasi lainnya)
- Dapat diubah menjadi aturan klasifikasi yang sederhana dan mudah dipahami
- Dapat menggunakan query SQL untuk mengakses database
- Akurasi klasifikasi yang sebanding dengan metode lain

Latihan

- Lakukan eksperimen mengikuti buku Matthew North, Data Mining for the Masses 2nd Edition, 2016, Chapter 10 (Decision Tree), p 195-217
- Datasets:
  - eReaderAdoption-Training.csv
  - eReaderAdoption-Scoring.csv
- Analisis peran metode prunning pada decision tree dan hubungannya dengan nilai confidence
- Analisis jenis decision tree apa saja yang digunakan dan mengapa perlu dilakukan pada dataset tersebut

Latihan

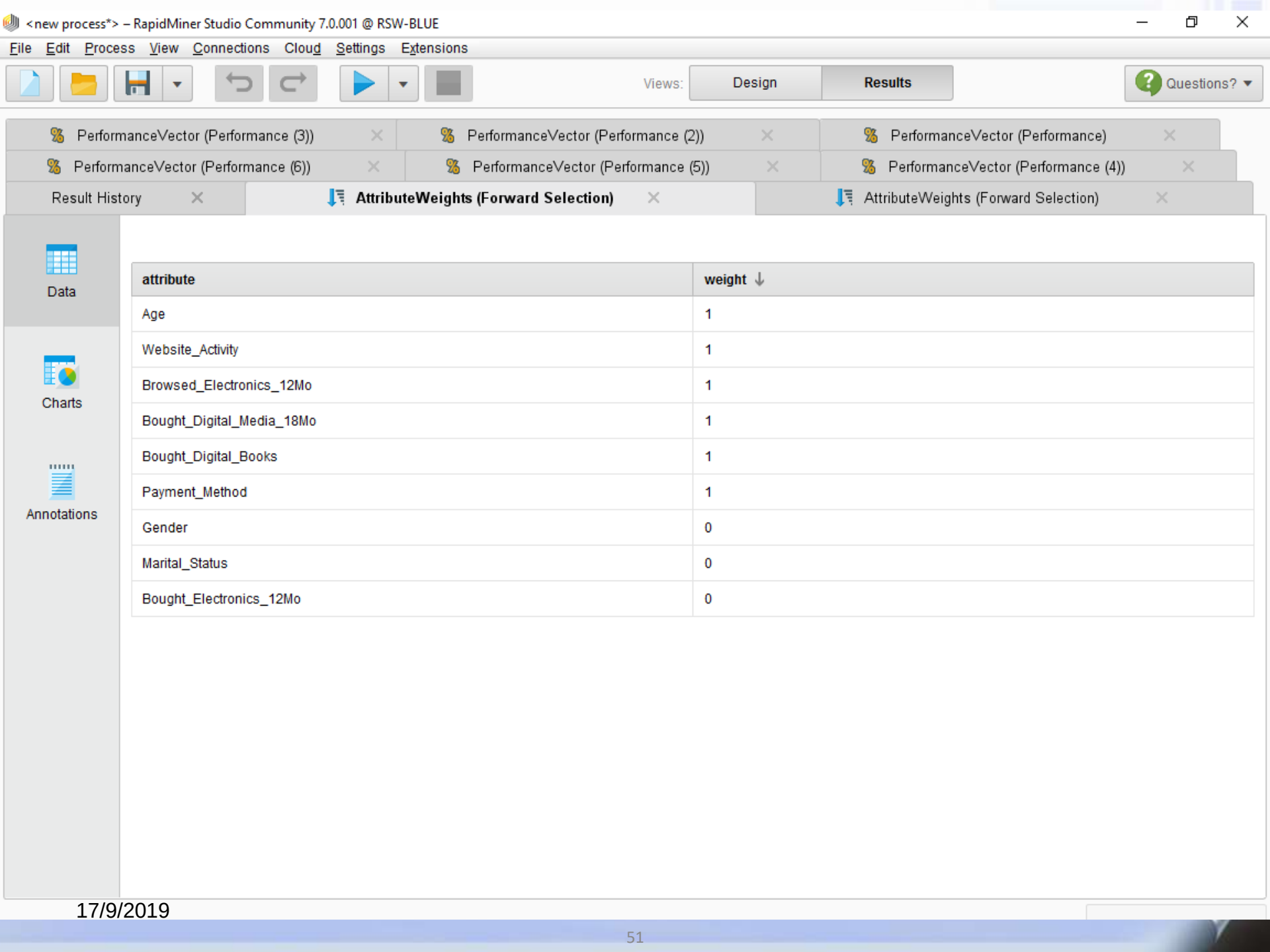
- Lakukan training pada data eReader Adoption (eReader-Training.csv) dengan menggunakan DT dengan 3 alternative criterion (Gain Ratio, Information Gain dan Gini Index)
- Ujicoba masing-masing split criterion baik menggunakan prunning atau tidak
- Lakukan pengujian dengan menggunakan 10-fold X Validation
- Dari model terbaik, tentukan faktor (atribut) apa saja yang berpengaruh pada tingkat adopsi eReader

	DTGR	DTIG	DTGI	DTGR+Pr	DTIG+Pr	DTGI+Pr
Accuracy				58.39	51.01	31.01

Latihan

- Lakukan feature selection dengan Forward Selection untuk ketiga algoritma di atas
- Lakukan pengujian dengan menggunakan 10-fold X Validation
- Dari model terbaik, tentukan faktor (atribut) apa saja yang berpengaruh pada tingkat adopsi eReader

	DTGR	DTIG	DTGI	DTGR+FS	DTIG+FS	DTGI+FS
Accuracy	58.39	51.01	31.01	61.41	56.73	31.01


<new process*> – RapidMiner Studio Community 7.0.001 @ RSW-BLUE
File Edit Process View Connections Cloud Settings Extensions
Views: Design Results
Questions?
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
PerformanceVector (Performance)
PerformanceVector (Performance (6))
PerformanceVector (Performance (5))
PerformanceVector (Performance (4))
Result History
AttributeWeights (Forward Selection)
AttributeWeights (Forward Selection)
Data
Charts
Annotations
attribute
weight
Age 1
Website_Activity 1
Browsed_Electronics_12Mo 1
Bought_Digital_Media_18Mo 1
Bought_Digital_Books 1
Payment_Method 1
Gender 0
Marital_Status 0
Bought_Electronics_12Mo 0

Review dan Latihan

☺ END ☺