Faculty of Computer
and Information Sciences
Ain Shams University

كلية الحاسبات و المعلومات
جامعة عين شمس

Data Science

Code:

Instructor

Prof.Dr. Abeer M. Mahmoud

Professorof Computer Science -faculty of Computer and Information Sciences
- Ain Shams University

Data Science and Big Data Analytics v2

DELL Technologies

Topics : Data Science and Big Data Analytics Course

Introduction to Big Data Analytics + Data Analytics Lifecycle	Review of Basic Data Analytic Methods Using R	Advanced Analytics – Theory and Methods	Advanced Analytics - Technology and Tools	The Endgame, or Putting it All Together + Final Lab on Big Data Analytics
Big Data Overview State of the Practice in Analytics The Data Scientist Big Data Analytics in Industry Verticals Data Analytics Lifecycle	Using R to Look at Data - Introduction to R Analyzing and Exploring the Data Statistics for Model Building and Evaluation	K-means Clustering Association Rules Linear Regression Logistic Regression Naive Bayesian Classifier Decision Trees Time Series Analysis Text Analysis	Analytics for Unstructured Data (MapReduce and Hadoop) The Hadoop Ecosystem In-database Analytics – SQL Essentials Advanced SQL and MADlib for In-database Analytics	Operationalizing an Analytics Project Creating the Final Deliverables Data Visualization Techniques + Final Lab – Application of the Data Analytics Lifecycle to a Big Data Analytics Challenge

DELL Technologies

Advanced analytics—theory and methods

DELL Technologies

Lesson: Naïve Bayes

Naïve Bayes

During this lesson, the following topics are covered:

- Theoretical foundations of the Naïve Bayes classifier
- Use cases
- Evaluating the effectiveness of the classifier
- Reasons to choose (+) and cautions (-)

Classifiers

Where in the catalog should I place this product listing?

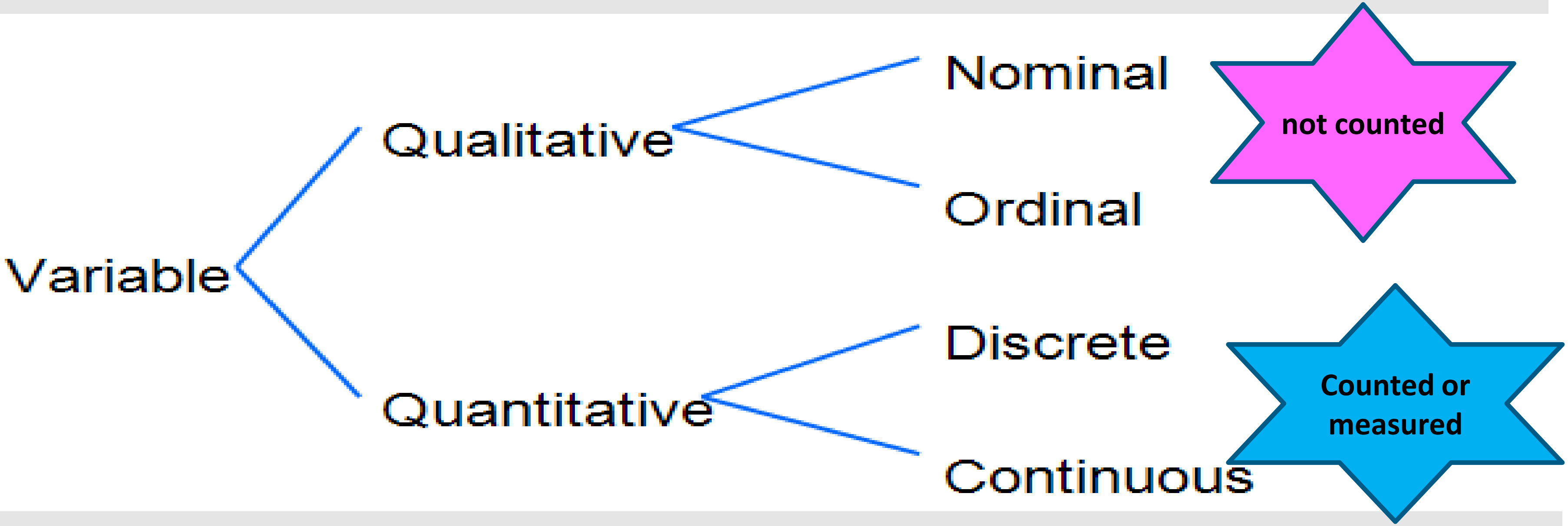
Is this email spam?

Will the customer buy the product?

- Classification
  - Assign labels to objects.
  - Usually supervised - training dataset of preclassified observations
- Commonly used classifiers
  - Naïve Bayes
  - Decision Trees
  - Logistic Regression

Types of Variables

Qualitative : a broad category for any variable that can’t be counted (i.e. has no numerical value). Nominal and ordinal variables fall under this umbrella term.



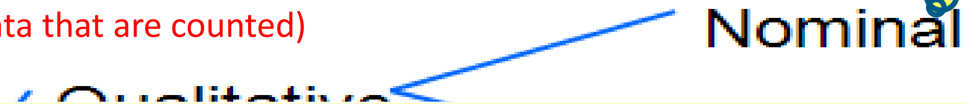
Quantitative : A broad category that includes any variable that **can be counted**, or **has a numerical value** associated with it. Examples of variables that fall into this category include discrete variables and Continuous variables.

DELL Technologies

Types of Variables

"named", i.e. classified into one or more qualitative categories or description

(data that are counted)

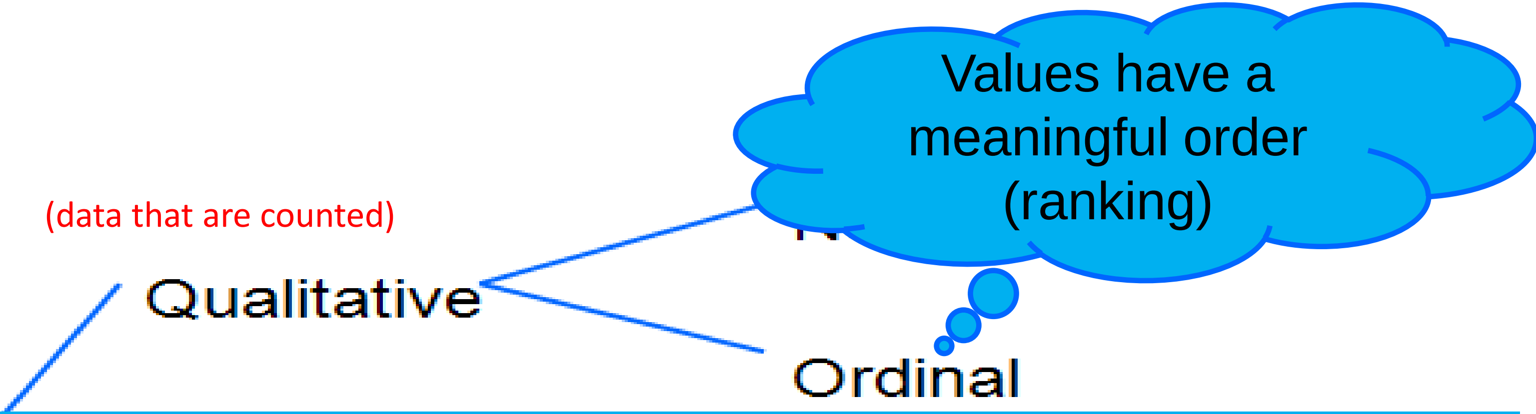


In medicine, nominal variables are often used to describe the patient. Examples of nominal variables might include:

- Gender (male, female)
- Eye color (blue, brown, green, hazel)
- Surgical outcome (dead, alive)
- Blood type (A, B, AB, O)

DELL Technologies

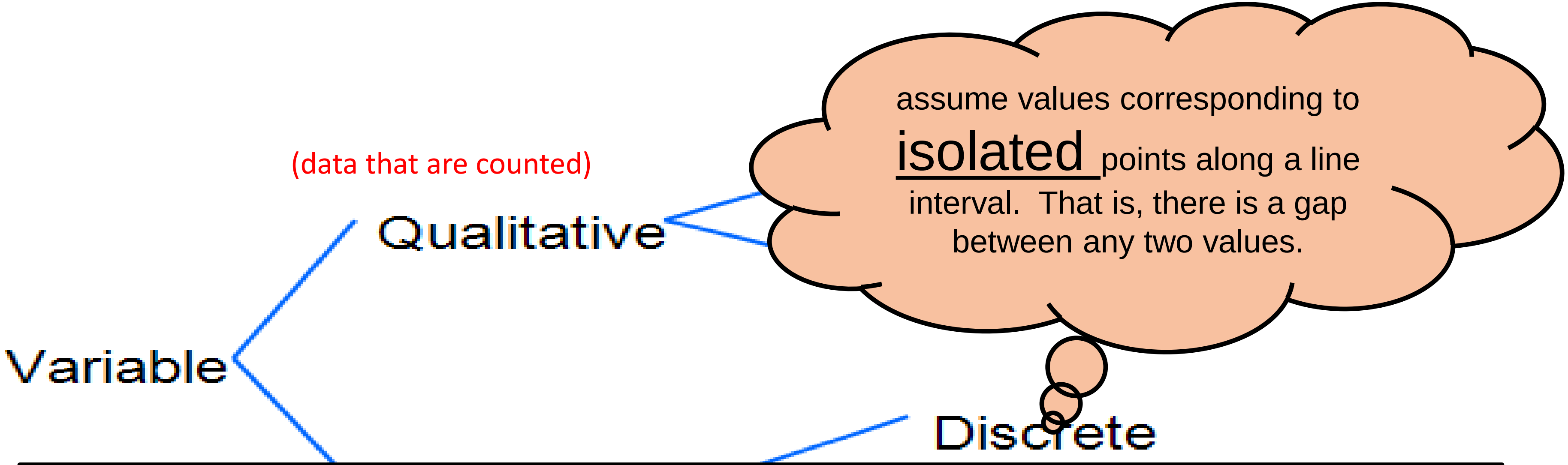
Types of Variables



In medicine, ordinal variables often describe the patient's characteristics, attitude, behavior, or status. Examples of ordinal variables might include:

- Stage of cancer (stage I, II, III, IV)
- Education level (elementary, secondary, college)
- Pain level (mild, moderate, severe)
- Satisfaction level (very dissatisfied, dissatisfied, neutral, satisfied, very satisfied)
- Agreement level (strongly disagree, disagree, neutral, agree, strongly agree)

Types of Variables



Discrete Variables that have constant, equal distances between values, but the zero point is arbitrary.

Examples of interval variables:

- Intelligence (IQ test score of 100, 110, 120, etc.)
- Pain level (1-10 scale)
- Body length in infant

Types of Variables

uncountable number of values.
can assume any value along a
line interval, including every
possible value between
any two values.

(data that are counted)

Qualitative

Variable

Discrete

Quantitative

(data that are measured)

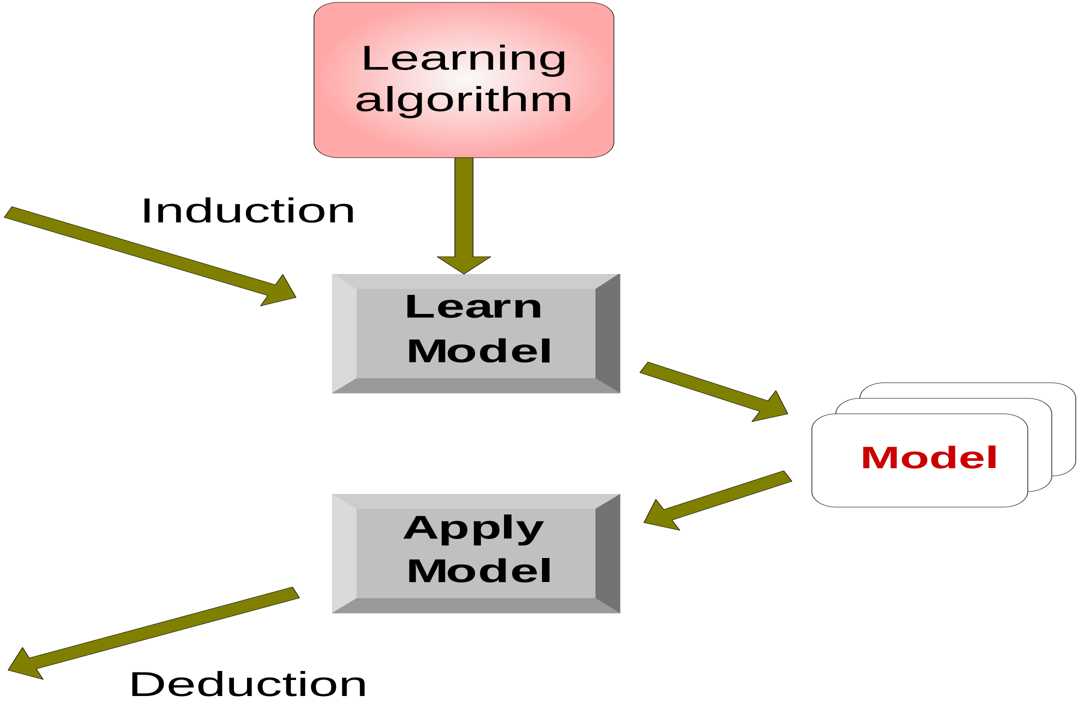
Continuous

Practically, **real values** can only be measured and represented using a finite number of digits
Continuous attributes are typically represented as floating-point variables

Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Dr. Abeer Mahmoud
(course coordinator)

Naïve Bayes classifier approach

- Based on the observed object attributes
  - Naïvely assumed to be conditionally independent of each other
  - Class label probabilities are determined using Bayes' Law
  - Determine the most probable class label for each object
- Example:
  - Classify an object based on its attributes {shape, color, weight}
  - Given an object that is {spherical, yellow, < 60 grams}
    - P(tennis ball, given spherical, yellow, < 60 grams) = 0.32
    - P(apple, given spherical, yellow, < 60 grams) = 0.09
    - P(bowling ball, given spherical, yellow, < 60 grams) = 0.00000001
- **Input** variables are discrete, categorical
- **Output**:
  - Probability score for each possible class label
    - Proportional to the true probability
  - Assigned class label, based on the highest probability score

> If two people live in the same city, the probability that person **A** gets home in time for dinner, and the probability that person **B** gets home in time for dinner are independent; that is, we wouldn't expect one to have an affect on the other. But if a snow storm hits the city and introduces a probability **C** that traffic will be at a stand still, you would expect that the probability of both **A** getting home in time for dinner and **B** getting home in time for dinner, would change.

DELL Technologies

Naïve Bayes—use cases

- Insurance fraud detection
- Text classification
  - Spam filtering
  - Document classification
- Medical diagnosis
- Applicable for cases with:
  - Many input variables and values
  - Multiple class labels

DELL Technologies

Build training dataset to predict customer purchase

- Predict if the customer will purchase the product based on their profile:
  - Age bins
  - Occupation
  - Income tier
- Note: Continuous variables are transformed into categorical variables.

Purchase_flag	Age_tiers	Occupation	Income_tiers_1000s
Yes	40 to 50	Professor	<80
Yes	30 to 40	Data Scientist	>200
Yes	50 to 60	Professor	>200
Yes	30 to 40	Professor	>200
Yes	>60	Doctor	>200
Yes	50 to 60	Professor	80 to 120
Yes	>60	Doctor	120 to 200
Yes	30 to 40	Professor	120 to 200
Yes	50 to 60	Professor	80 to 120
Yes	40 to 50	Electrician	120 to 200
Yes	40 to 50	Doctor	80 to 120
Yes	20 to 30	Data Scientist	120 to 200
Yes	50 to 60	Data Scientist	80 to 120
Yes	20 to 30	Professor	>200
No	30 to 40	Electrician	>200
No	30 to 40	Electrician	120 to 200
No	20 to 30	Electrician	>200
No	30 to 40	Professor	>200
No	40 to 50	Electrician	>200
No	>60	Professor	>200
No	30 to 40	Electrician	<80
No	50 to 60	Electrician	120 to 200
No	30 to 40	Electrician	120 to 200
No	>60	Doctor	80 to 120
No	20 to 30	Professor	<80
No	30 to 40	Electrician	>200
No	>60	Electrician	120 to 200
No	>60	Data Scientist	80 to 120
No	30 to 40	Professor	>200
No	40 to 50	Doctor	<80
No	30 to 40	Electrician	80 to 120
No	>60	Doctor	>200
No	30 to 40	Professor	120 to 200
No	30 to 40	Doctor	>200
No	20 to 30	Data Scientist	80 to 120

Conditional probability

The probability of event C occurring given event A has occurred

Denoted as P(C | A)

Example:

A fair 6-sided die is thrown

Let A = {an even number is rolled}

If C = {a 3 is rolled}, then P(C | A) = 0

If C = {a 4 is rolled}, then P(C | A) = 1/3

Knowing that A occurred, provides information about the probability of C

Formal definition:

$$P(C \mid A) = \frac{P(A \cap C)}{P(A)} \quad for\ P(A) > 0$$

where $P(A \cap C)$ denotes probability of events A and C occurring

Derivation of Bayes' Law

By definition of conditional probability,

$$P(C \mid A) = \frac{P(A \cap C)}{P(A)} \qquad (1)$$

Alternatively,

$$P(A \mid C) = \frac{P(A \cap C)}{P(C)} \quad \rightarrow \quad P(A \cap C) = P(A \mid C)P(C) \qquad (2)$$

Substituting back into the definition yields:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Known as Bayes' Law

A conditional probability can be expressed as a function of another conditional probability

Application of Bayes' Law

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Scenario

John flies frequently and likes to upgrade his seat to first class.

If John arrives at least two hours early, then he will get the upgrade 75 percent of the time.

Otherwise, he will get the upgrade 35 percent of the time.

John arrives at least two hours early only 40 percent of the time.

Suppose that John did not receive an upgrade on his most recent attempt.

What is the probability that he arrived late?

$$P(Late \mid No\ Upgrade) = \frac{P(No\ Upgrade \mid Late)P(Late)}{P(No\ Upgrade)}$$

$$= \frac{(1 - 0.35)(1 - 0.40)}{1 - (0.40 * 0.75 + 0.60 * 0.35)} = 0.80$$

Apply Naïve assumption and remove constant

For observed attributes $A = (a_1, a_2, \ldots a_m)$, compute

$$P(C_i \mid A) = \frac{P(a_1, a_2, \ldots, a_m \mid C_i) P(C_i)}{P(a_1, a_2, \ldots, a_m)} \qquad i = 1, 2, \ldots, n$$

and assign the classifier C_i with the largest $P(C_i \mid A)$

Two simplifications to the calculations

Apply naïve assumption - each a_j is conditionally independent of each other, then

$$P(a_1, a_2, \ldots, a_m \mid C_i) = P(a_1 \mid C_i) P(a_2 \mid C_i) \cdots P(a_m \mid C_i) = \prod_{j=1}^{m} P(a_j \mid C_i)$$

Denominator $P(a_1, a_2, \ldots, a_m)$ is a constant and can be ignored

Building Naïve Bayesian classifier

Applying the two simplifications

$$P(C_i \mid a_1, a_2, \ldots, a_m) \propto \left(\prod_{j=1}^{m} P(a_j \mid C_i) \right) P(C_i) \qquad i = 1, 2, \ldots, n$$

To build a Naïve Bayesian Classifier, collect the following statistics from the training data:

$P(C_i)$ for all the class labels

$P(a_j \mid C_i)$ for all possible a_j and C_i

Assign the classifier label C_i that maximizes the value of

$$\left(\prod_{j=1}^{m} P(a_j \mid C_i) \right) P(C_i) \qquad i = 1, 2, \ldots, n$$

Naïve Bayesian classifiers for product purchase example

- Class labels: {Yes, No}
  - P(Yes) = 0.39
  - P(No) = 0.61
- Conditional Probabilities
  - P(Electrician|Yes) = 0.42
  - P(Electrician|No) = 0.27
  - P(Data Scientist|Yes) = 0.21
  - P(Data Scientist|No) = 0.27
  - … and so on

Purchase_flag	Age_tiers	Occupation	Income_tiers_1000s
Yes	40 to 50	Professor	<80
Yes	30 to 40	Data Scientist	>200
Yes	50 to 60	Professor	>200
Yes	30 to 40	Professor	>200
Yes	>60	Doctor	>200
Yes	50 to 60	Professor	80 to 120
Yes	>60	Doctor	120 to 200
Yes	30 to 40	Professor	120 to 200
Yes	50 to 60	Professor	80 to 120
Yes	40 to 50	Electrician	120 to 200
Yes	40 to 50	Doctor	80 to 120
Yes	20 to 30	Data Scientist	120 to 200
Yes	50 to 60	Data Scientist	80 to 120
Yes	20 to 30	Professor	>200
No	30 to 40	Electrician	>200
No	30 to 40	Electrician	120 to 200
No	20 to 30	Electrician	>200
No	30 to 40	Professor	>200
No	40 to 50	Electrician	>200
No	>60	Professor	>200
No	30 to 40	Electrician	<80
No	50 to 60	Electrician	120 to 200
No	30 to 40	Electrician	120 to 200
No	>60	Doctor	80 to 120
No	20 to 30	Professor	<80
No	30 to 40	Electrician	>200
No	>60	Electrician	120 to 200
No	>60	Data Scientist	80 to 120
No	30 to 40	Professor	>200
No	40 to 50	Doctor	<80
No	30 to 40	Electrician	80 to 120
No	>60	Doctor	>200
No	30 to 40	Professor	120 to 200
No	30 to 40	Doctor	>200
No	20 to 30	Data Scientist	80 to 120

Naïve Bayesian classifier example, cont.

- Given applicant attributes of

  A= {Age 30–40,

  Occupation Electrician,

  Income 80–120}

- Since P(No|A) > (Yes|A), assign the label No, the customer will not purchase.

  P(Yes|A) ~ (0.21*0.42*0.28)*0.39 = 0.009

  P(No|A) ~ (0.36*0.22*0.40)*0.61 = 0.019

a_j	C_i	$P(a_j \mid C_i)$
30-40	Yes	0.21
30-40	No	0.36
Electrician	Yes	0.42
Electrician	No	0.22
80-120	Yes	0.28
80-120	No	0.40

Naïve Bayesian implementation considerations

- Numerical underflow
  - Resulting from multiplying several probabilities near zero
  - **Preventable** by computing the logarithm of the products
- Zero probabilities due to unobserved attribute/classifier pairs
  - Resulting from rare events
  - Handled by smoothing—adjusting each probability by a small amount
- Assign the classifier label, C_i, that maximizes the value of

$$\left(\sum_{j=1}^{m} \log P'(a_j \mid C_i)\right) + \log P(C_i)$$

where i = 1,2,…,n and

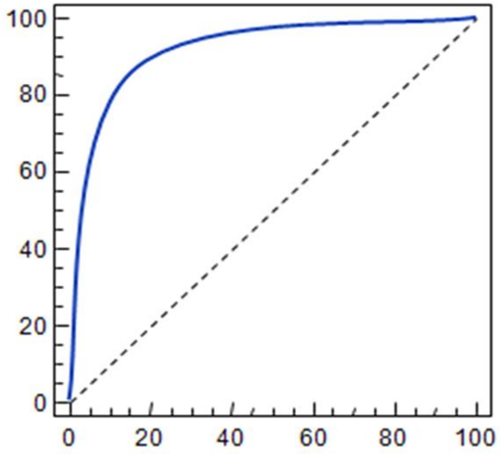
P' denotes the adjusted probabilities

DELL Technologies

Diagnostics

- Hold-out data
  - How well does the model classify new instances?
- Cross-validation
- ROC curve/AUC
- Confusion Matrix

True Positive rate
(Sensitivity)



False Positive rate (100-Specify)

DELL Technologies

Naïve Bayesian classifier—reasons to choose (+) and cautions (-)

Reasons to choose (+)	Cautions (-)
Handles missing values quite well	Numeric variables must be discrete, categorized, Intervals
Robust to irrelevant variables	Sensitive to correlated variables Double-counting
Easy to implement	Not good for estimating probabilities Stick to class label or yes/no
Easy to score data	
Resistant to overfitting	
Computationally efficient Handles very high-dimensional problems Handles categorical variables with many levels	

Check your knowledge

1. Consider the following training dataset:
   - Apply the Naïve Bayesian Classifier to this dataset and compute the probability score for $P(y = 1|X)$ for $X = (1,0,0)$
   - Show your work
2. List some prominent use cases of the Naïve Bayesian Classifier.
3. What gives the Naïve Bayesian Classifier the advantage of being computationally inexpensive?
4. Why should you use logs in the probability scoring calculations?

X1	X2	X3	Y
1	1	1	0
1	1	0	0
0	0	0	0
0	1	0	1
1	0	1	1
0	1	1	1

Training Dataset

Check your knowledge, cont.

1. Consider the following dataset with two input features, temperature and season:
   A. What is the Naïve Bayesian assumption?
   B. Is the Naïve Bayesian assumption satisfied for this problem?

Temperature	Season	Electricity Usage
-10 to 50 F	Winter	High
50 to 70 F	Winter	Low
70 to 85 F	Summer	Low
85 to 110 F	Summer	High

Naïve Bayesian classifiers—summary

During this lesson, the following topics were covered:

- Naïve Bayesian Classifier
- Theoretical foundations of the classifier
- Use cases
- Evaluating the effectiveness of the classifier
- The reasons to choose (+) and cautions (-) with the use of the classifier

DELL Technologies

Example on Naïve Bayesian classifiers

An example

- Compute all probabilities required for classification

A	B	C
m	b	t
m	s	t
g	q	t
h	s	t
g	q	t
g	q	f
g	s	f
h	b	f
h	q	f
m	b	f

$\Pr(C = t) = 1/2,$ $\Pr(C = f) = 1/2$

$\Pr(A{=}m \mid C{=}t) = 2/5$
$\Pr(A{=}m \mid C{=}f) = 1/5$
$\Pr(B{=}b \mid C{=}t) = 1/5$
$\Pr(B{=}b \mid C{=}f) = 2/5$

$\Pr(A{=}g \mid C{=}t) = 2/5$
$\Pr(A{=}g \mid C{=}f) = 2/5$
$\Pr(B{=}s \mid C{=}t) = 2/5$
$\Pr(B{=}s \mid C{=}f) = 1/5$

$\Pr(A{=}h \mid C{=}t) = 1/5$
$\Pr(A{=}h \mid C{=}n) = 2/5$
$\Pr(B{=}q \mid C{=}t) = 2/5$
$\Pr(B{=}q \mid C{=}f) = 2/5$

Now we have a test example:

$A = m \quad B = q \quad C = ?$

DELL Technologies

An Example (cont …)

C = t is more probable. t is the final class.

A=m and B=q C??

- For C = t, we have

$$\Pr(C=t)\prod_{j=1}^{2}\Pr(A_j=a_j \mid C=t)=\frac{1}{2}\times\frac{2}{5}\times\frac{2}{5}=\frac{2}{25}$$

- For class C = f, we have

$$\Pr(C=f)\prod_{j=1}^{2}\Pr(A_j=a_j \mid C=f)=\frac{1}{2}\times\frac{1}{5}\times\frac{2}{5}=\frac{1}{25}$$

Pr(C = t) = 1/2,	Pr(C= f) = 1/2	
Pr(A=m | C=t) = 2/5	Pr(A=g | C=t) = 2/5	Pr(A=h | C=t) = 1/5
Pr(A=m | C=f) = 1/5	Pr(A=g | C=f) = 2/5	Pr(A=h | C=n) =2/5
Pr(B=b | C=t) = 1/5	Pr(B=s | C=t) = 2/5	Pr(B=q | C=t) = 2/5
Pr(B=b | C=f) = 2/5	Pr(B=s | C=f) = 1/5	Pr(B=q | C=f) = 2/5

Now we have a test example:

A = m B = q C = ?

Your Turn

A=g and
B=b C??

Dr. Abeer Mahmoud
(course coordinator)

Lesson: Decision trees

Decision Trees

During this lesson, the following topics are covered:

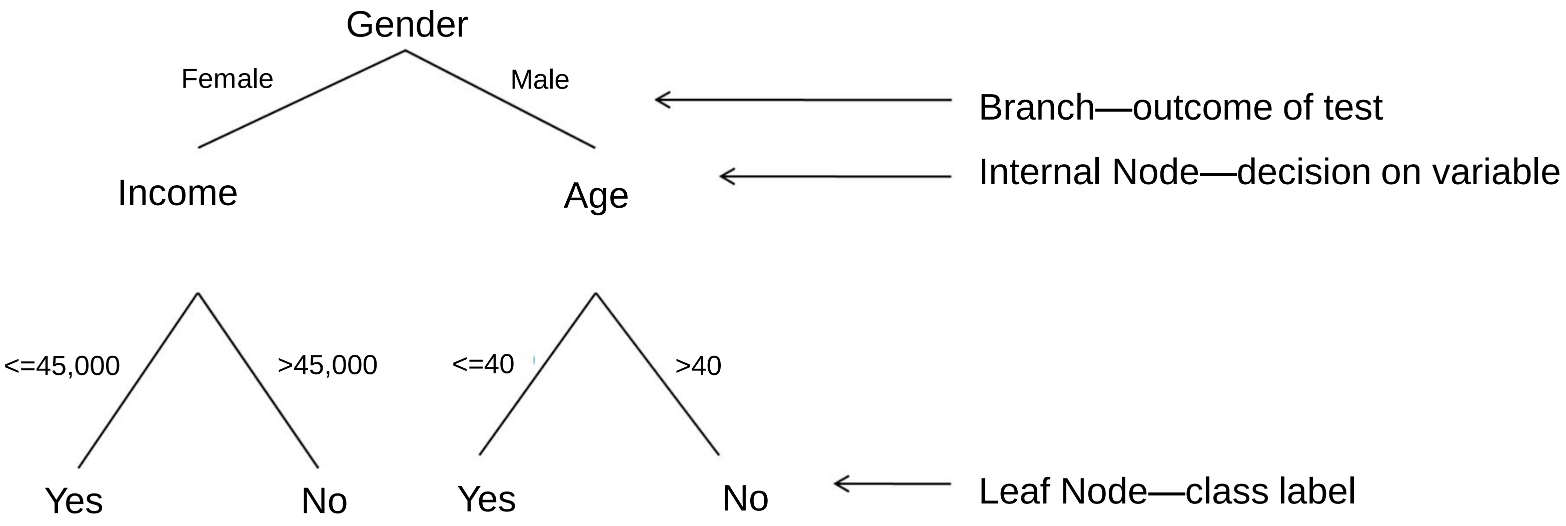
- Overview of Decision Tree classifier
- General algorithm for Decision Trees
- Decision Tree use cases
- Entropy, Information gain
- Reasons to choose (+) and cautions (-) of Decision Tree classifier
- Classifier methods and conditions in which they are best suited

Decision Tree classifier—what is it?

- Used for classification:
  - Returns probability scores of class membership
    - Well-calibrated, as is logistic regression
    - Assigns label based on highest scoring class
    - Some Decision Tree algorithms return simply the most likely class
  - Regression Trees: a variation for regression
    - Returns average value at every node
    - Predictions can be discontinuous at the decision boundaries
- **Input** variables can be continuous or discrete
- **Output**:
  - This output is a tree that describes the decision flow.
  - Leaf nodes return either a probability score or simply a classification.
  - Trees can be converted to a set of "decision rules."
    - "IF income < $50,000 AND mortgage_amt > $100K THEN default=T with 75% probability"

DELL Technologies

Decision Tree—example of visual structure

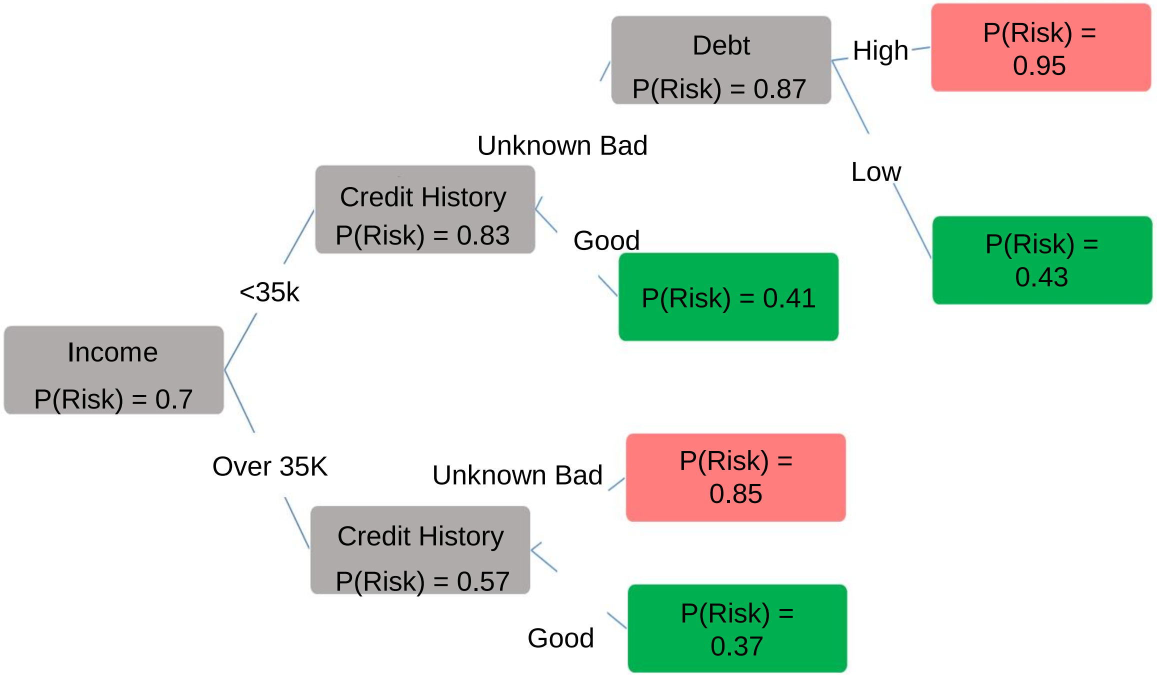


DELL Technologies

Decision Tree classifier—use cases

- When a series of questions (yes/no) are answered to arrive at a classification
  - Biological species classification
  - Checklist of symptoms during a doctor's evaluation of a patient
- When "if-then" conditions are preferred to linear models.
  - Customer segmentation to predict response rates
  - Financial decisions such as loan approval
  - Fraud detection
- Short Decision Trees are the most popular "weak learner" in ensemble learning techniques

Example—credit risk problem



General algorithm

- To construct tree T from training set S
  - If all examples in S belong to some class in C, or S is sufficiently "pure", then make a leaf labeled C.
  - Otherwise:
    - Select the "most informative" attribute A.
    - Partition S according to A's values.
    - Recursively construct subtrees T1, T2, and so on, for the subsets of S.
- The details vary according to the specific algorithm—CART, ID3, C4.5—but the general idea is the same.

Step 1—Pick most informative attribute

Entropy-based methods are one common way:

$$H = -\sum_{c} p(c) \log_2 p(c)$$

H=0 if p(c) = 0 or 1 for any class.

So, for binary classification, H=0 is a pure node.

H is maximum when all classes are equally probable.

For binary classification, H=1 when classes are 50/50.

Step 1—Pick most informative attribute—conditional entropy

$$H_{attr} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

The weighted sum of the class entropies for each value of the attribute.

In English: attribute values—homeowner vs. renter—give more information about class membership.

> "Homeowners are more likely to have good credit than renters."

Conditional entropy should be lower than unconditioned entropy.

Step 1—which attribute is best classifier?

A statistical property called information gain, measures how well a given attribute separates the training examples.

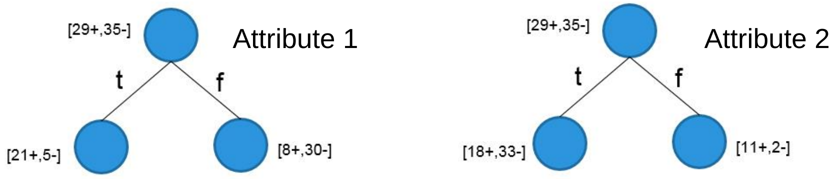
Information gain uses the notion of entropy, commonly used in information theory.

$$\text{InfoGain}_{attr} = H - Hattr$$

Information gain = expected reduction of entropy.

H is entropy at the first node, and H(attr) is entropy of the leaf nodes.

Here is an example of two separate attributes from a dataset with 29 positive and 35 negative responses.



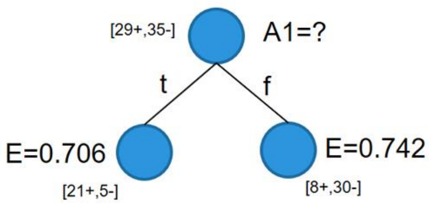
DELL Technologies

Step 1—which attribute is best classifier? (cont.)

Entropy Calculation at root node

$$\text{Entropy}\ ([29+, 35-] = -\left(\frac{29}{64}\right) log_2\left(\frac{29}{64}\right) - \left(\frac{35}{64}\right) log_2\left(\frac{35}{64}\right)$$
$$= 0.994$$

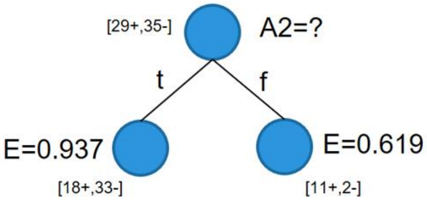
Information gain of the two variables



= 0.994 - (26/64)*0.706 - (38/64)*0.742

= 0.266

Information Gain A1 = 0.266



= 0.994 - (51/64)*0.937 - (13/64)*0.619

= 0.121

Information Gain A2 = 0.121

DELL Technologies

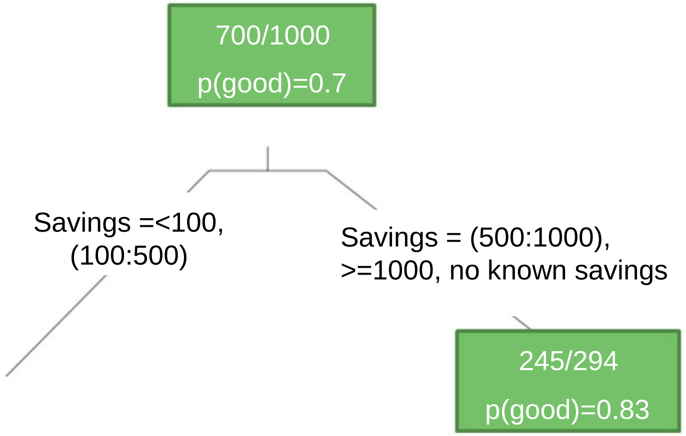
Conditional entropy example

	For free	Own	Rent
P(housing)	0.108	0.713	0.179
P(bad | housing)	0.407	0.261	0.391
P(good | housing)	0.592	0.739	0.601

H (housing/credit) = -[0.108 * (0.407 log2(0.407) + 0.592 log2 (0.592))
+ 0.713 * (0.261 log2 (0.261) + 0.739 log2 (0.739))
+ 0.179 * (0.391 log2 (0.391) + 0.601 log2 (0.601))
=0.868

Steps 2 and 3—partition on selected variable

- Step 2: Find the partition with the highest InfoGain.
  - In this example, the selected partition has InfoGain = 0.028.
- Step 3: At each resulting node, repeat Steps 1 and 2.
  - Until node is "pure enough"
- Pure nodes → no information gain by splitting on other attributes



DELL Technologies

Your Turn

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:

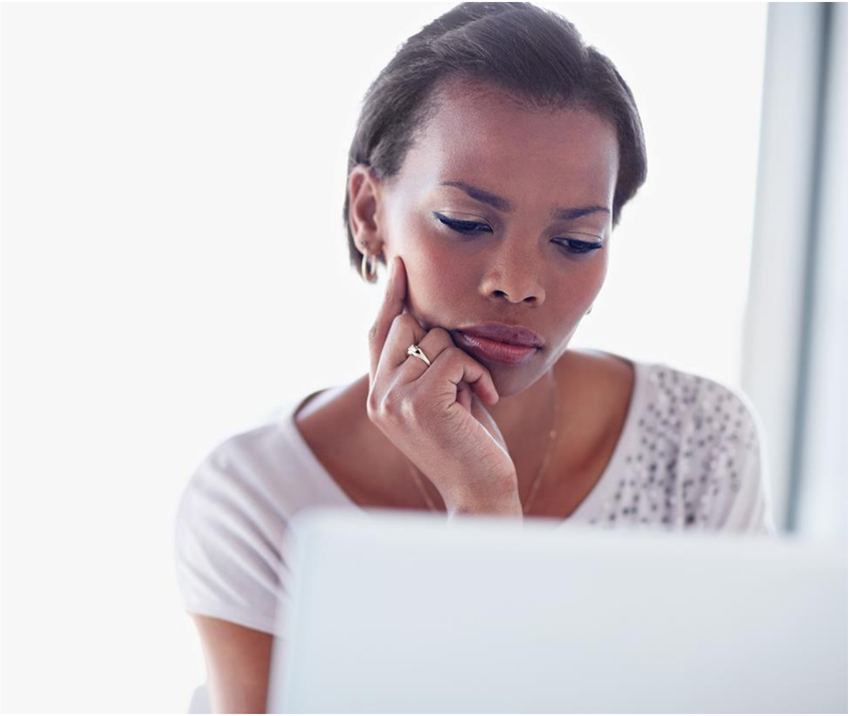
For the following record set ,

1. classify if the data is training or testing sets ……………………..
2. Choose one <u>categorical attribute</u> and show how you construct it as a node showing its children ………………………………………..
3. Choose one <u>numerical attribute</u> and show how you construct it as a node showing its children ………………………………………..

47

Diagnostics

- Hold-out data
- ROC/AUC
- Confusion Matrix
- FPR/FNR, Precision/Recall
- Do the splits—or the rules—make sense?
  - What does the domain expert say?
- How deep is the tree?
  - Too many layers are prone to overfit.
- Do you get nodes with very few members?
  - Overfit



Decision Tree classifier– reasons to choose (+) and cautions (-)

Reasons to choose (+)	Cautions (-)
Takes any input type—numeric, categorical.	Tree structure is sensitive to small changes in the training data.
Robust with redundant variables, correlated variables.	
Naturally handles variable interaction.	Avoid an overly deep tree that may be overfit.
Handles variables that have nonlinear effect on outcome.	
Computationally efficient to build.	Does not naturally handle missing values; However. most implementations include a method for dealing with this issue.
Easy to score data.	In practice, decision rules can be fairly complex.
Many algorithms can return a measure of variable importance.	
Decision rules are easy to understand.	

Which classifier should I try?

Typical Questions	Recommended Method
Do I want class probabilities, rather than just class labels?	Logistic regression Decision Tree
Do I want insight into how the variables affect the model?	Logistic regression Decision Tree
Is the problem high-dimensional?	Naïve Bayes
Do I suspect some of the inputs are correlated?	Decision Tree Logistic regression
Do I suspect some of the inputs are irrelevant?	Decision Tree Naïve Bayes
Are there categorical variables with many levels?	Naïve Bayes Decision Tree
Are there mixed variable types?	Decision Tree Logistic regression
Is there nonlinear data or discontinuities in the inputs that will affect the outputs?	Decision Tree

Check your knowledge

1. How do you define information gain?
2. For what conditions is the value of entropy at a maximum and when is it at a minimum?
3. List three use cases of decision trees.
4. What are weak learners and how are they used in ensemble methods?
5. Why do you end up with an overfitted model with deep trees and in datasets when you have outcomes that depend on many variables?
6. What classification method would you recommend for the following cases:
   - High-dimensional data
   - Data in which outputs are affected by nonlinearity and discontinuity in the inputs

Decision trees—summary

During this lesson, the following topics were covered:

- Overview of decision tree classifier
- General algorithm for decision trees
- Decision tree use cases
- Entropy, information gain
- Reasons to choose (+) and cautions (-) of decision tree classifier
- Classifier methods and conditions in which they are best suited

Example

No	Risk	Credit history	Debt	Collateral	Income
1	high	bad	high	none	0-15 $
2	high	unknown	high	none	15-35$
3	moderate	unknown	low	none	15-35$
4	high	unknown	low	none	0-15 $
5	low	unknown	low	none	Over 35$
6	low	unknown	low	adequate	Over 35$
7	high	bad	low	none	0-15 $
8	moderate	bad	low	adequate	Over 35$
9	low	good	low	none	Over 35$
10	low	good	high	adequate	Over 35$
11	high	good	high	none	0-15 $
12	moderate	good	high	none	15-35$
13	low	good	high	none	Over 35$
14	High	bad	high	none	15-35$

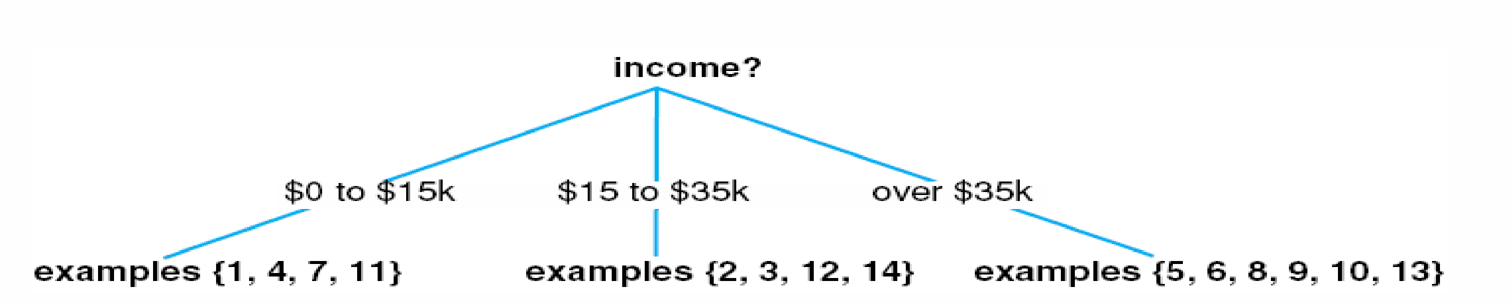
53

Dr. Abeer Mahmoud
(course coordinator)

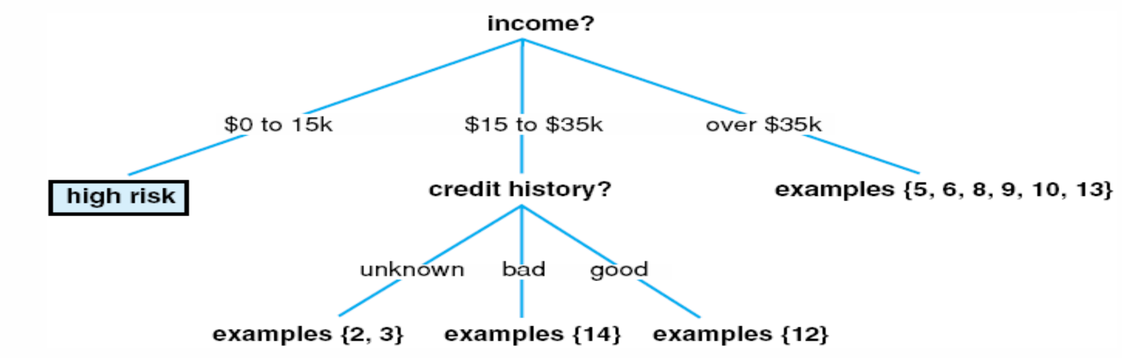
Example

starting with the population of loans

- suppose we first select the income property
- this separates the examples into three partitions



- all examples in leftmost partition have same conclusion – HIGH RISK
- other partitions can be further subdivided by selecting another property



(course coordinator)

ID3 & information theory

- the selection of which property to split on next is based on **information theory** the *information content* of a tree is defined by

$$I[tree] = \Sigma \text{-prob}(classification_i) * \log_2(\text{ prob}(classification_i))$$

  - e.g., In credit risk data, there are 14 samples

prob(high risk) = 6/14 prob(moderate risk) = 3/14

prob(low risk) = 5/14

the information content of a tree that correctly classifies these examples:

  - $I[tree] = -6/14 * \log_2(6/14) + -3/14 * \log_2(3/14) + -5/14 * \log_2(5/14)$

    $= -6/14 * -1.222 + -3/14 * -2.222 + -5/14 * -1.485$

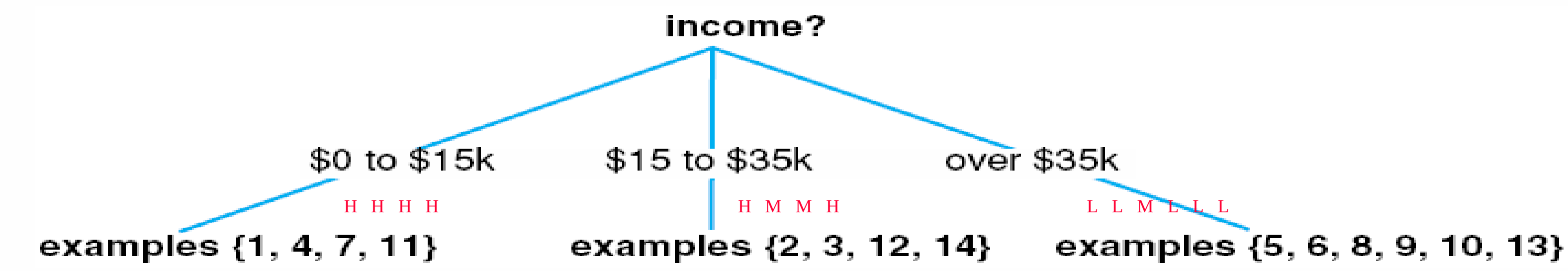
    $= 1.531$

55

ID3 & more information theory- example

– after splitting on a property, consider the expected (or remaining) content of the subtrees

E[property] = Σ (# in subtree$_i$ / # of samples) * I [subtree$_i$]



$$
\begin{aligned}
\mathbf{E[income]} &= 4/14 * I[subtree_1] + 4/14 * I[subtree_2] + 6/14 * I[subtree_3] \\
&= 4/14 * (-4/4 \log_2(4/4) + -0/4 \log_2(0/4) + -0/4 \log_2(0/4)) + \\
&\quad 4/14 * (-2/4 \log_2(2/4) + -2/4 \log_2(2/4) + -0/4 \log_2(0/4)) + \\
&\quad 6/14 * (-0/6 \log_2(0/6) + -1/6 \log_2(1/6) + -5/6 \log_2(5/6)) \\
&= 4/14 * (0.0+0.0+0.0) + 4/14 * (0.5+0.5+0.0) + 6/14 * (0.0+0.43+0.22) \\
&= 0.0 + 0.29 + 0.28 \\
&= 0.57
\end{aligned}
$$

Credit risk example (cont.)

- what about the other property options?
  - E[debt]? E[history]? E[collateral]?
    - after further analysis
      E[income] = 0.57
      E[debt] = 1.47
      E[history] = 1.26
      E[collateral] = 1.33

the ID3 selection rules splits on the property that produces the **minimal E[property]**

- in this example, income will be the first property split
- then repeat the process on each subtree

DELL Technologies

Missing Values

- What is the problem?
- During computation of the splitting predicate, we can selectively **<u>ignore</u>** records with missing values (note that this has some problems)
- But if a record r misses the value of the variable in the splitting attribute, r can not participate further in tree construction Algorithms for missing values address this problem.
- Simplest algorithm to solve this problem :

-If X is numerical (categorical), impute the overall <u>mean</u>

- if X is discrete attribute set the most common value

58

Dr. Abeer Mahmoud
(course coordinator)

Your Turn

Calculate for the E[debt] = ………………………