CS 412 Intro. to Data Mining

Chapter 8. Classification: Basic Concepts

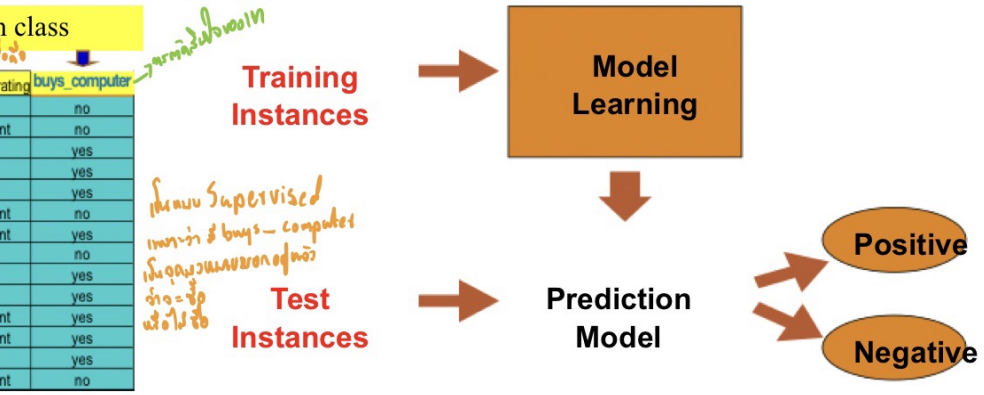
Jiawei Han, Computer Science, Univ. Illinois at Urbana-Champaign, 2017

Supervised vs. Unsupervised Learning (1)

- Supervised learning (classification)
  - Supervision: The training data such as observations or measurements are accompanied by **labels** indicating the classes which they belong to
  - New data is classified based on the models built from the training set

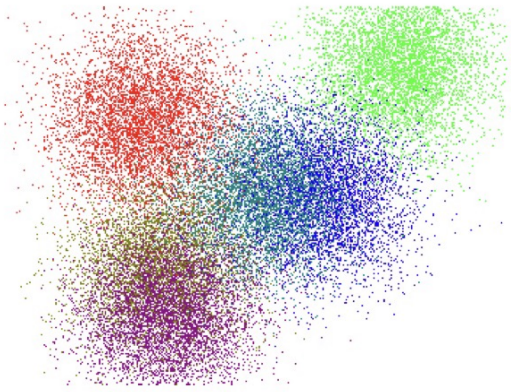
Training Data with class label:

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



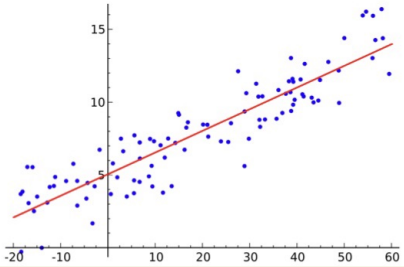
Supervised vs. Unsupervised Learning (2)

- Unsupervised learning (clustering)
  - The class labels of training data are unknown
  - Given a set of observations or measurements, establish the possible existence of classes or clusters in the data



Prediction Problems: Classification vs. Numeric Prediction

- Classification
  - Predict categorical class labels (discrete or nominal)
  - Construct a model based on the training set and the **class labels** (the values in a classifying attribute) and use it in classifying new data
- Numeric prediction
  - Model continuous-valued functions (i.e., predict unknown or missing values)
- Typical applications of classification
  - Credit/loan approval
  - Medical diagnosis: if a tumor is cancerous or benign
  - Fraud detection: if a transaction is fraudulent
  - Web page categorization: which category it is



Classification—Model Construction, Validation and Testing

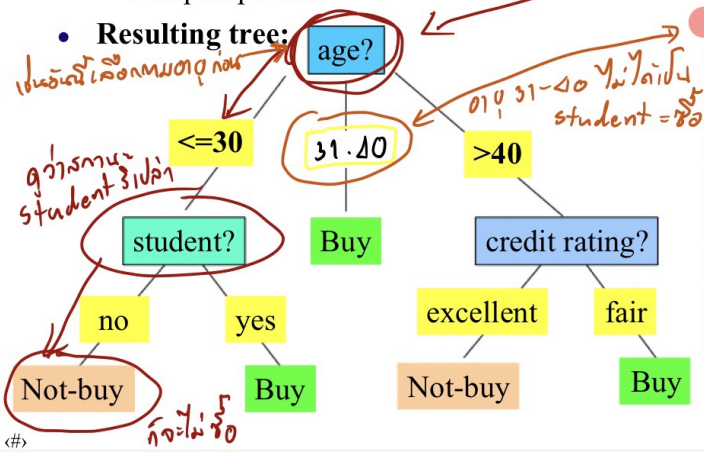
- **Model construction**
  - Each sample is assumed to belong to a predefined class (shown by the **class label**)
  - The set of samples used for model construction is **training set**
  - Model: Represented as decision trees, rules, mathematical formulas, or other forms
- **Model Validation and Testing**:
  - **Test:** Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - *Accuracy:* % of test set samples that are correctly classified by the model
    - Test set is independent of training set
  - **Validation**: If *the test set* is used to select or refine models, it is called **validation** (or development) **(test) set**
- **Model Deployment:** If the accuracy is acceptable, use the model to classify new data

Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- Decision Tree Induction
- Bayes Classification Methods
- Linear Classifier
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Additional Concepts on Classification
- Summary

Decision Tree Induction: An Example

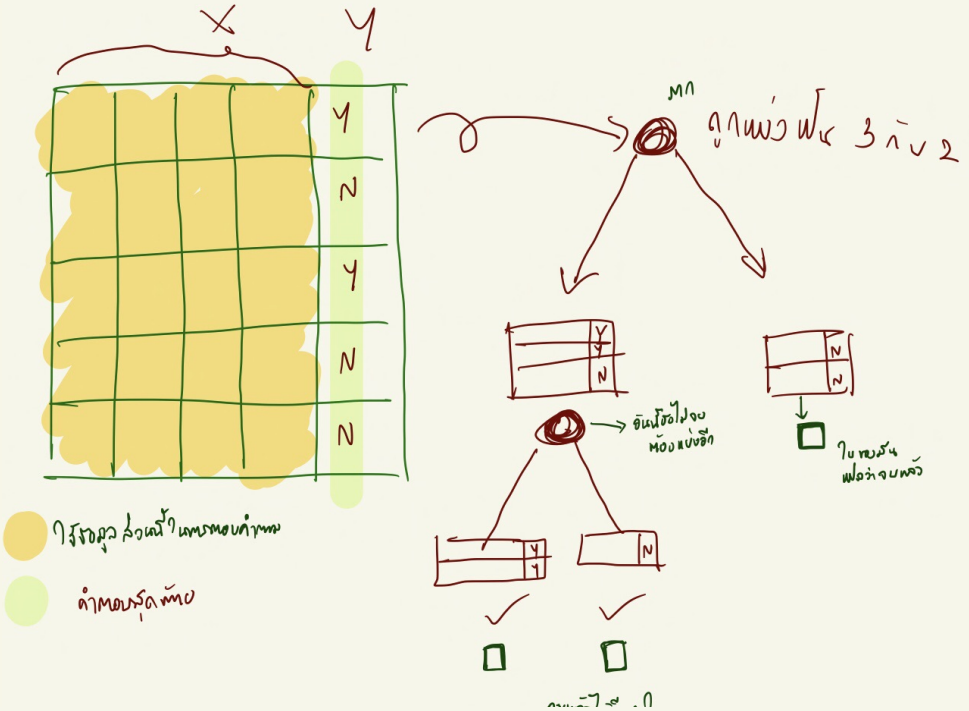
- **Decision tree construction**:
  - A top-down, recursive, divide-and-conquer process
- **Resulting tree:**



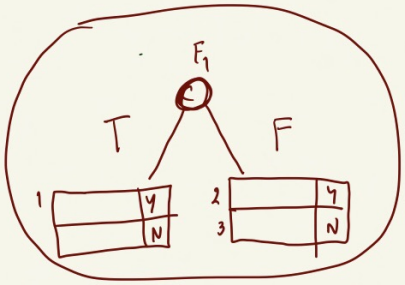
Training data set: Who buys computer?

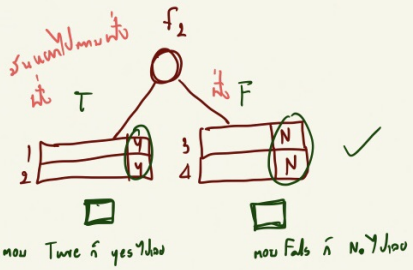
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	
31...40	high	no	fair	
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent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

Note: The data set is adapted from "Playing Tennis" example of R. Quinlan



F_1	F_2	F_3	Y
T	T	F	Y
F	T	F	Y
F	F	F	N
T	F	T	N





Information Gain: An Attribute Selection Measure

- Select the attribute with the highest information gain (used in typical decision tree induction algorithm: ID3/C4.5)
- Let *pi* be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci, 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

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

Example: Attribute Selection with Information Gain

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

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

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

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

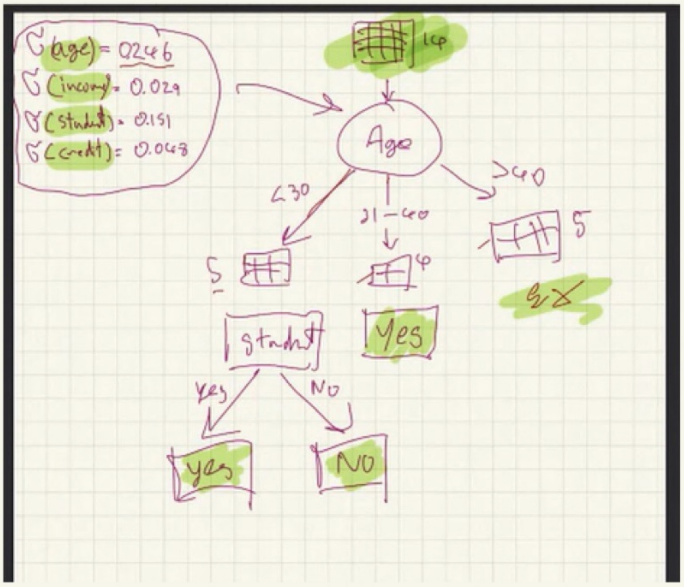
$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes's and 3 no's.

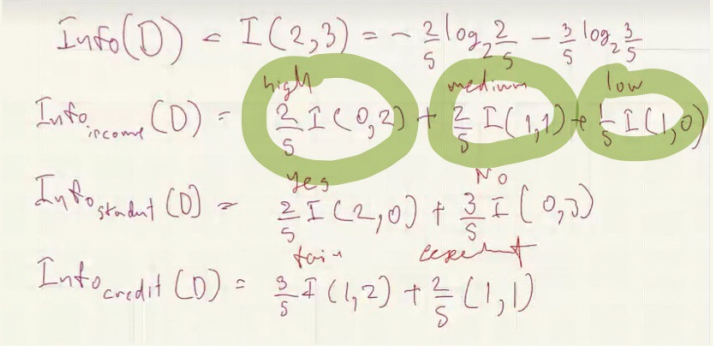
Hence

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

Similarly, we can get

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





Bayes' Theorem: Basics

- Total probability Theorem:

$$p(B) = \sum_i p(B|A_i)p(A_i)$$

- Bayes' Theorem:

$$p(H|X) = \frac{p(X|H)P(H)}{p(X)} \propto p(X|H)\,P(H)$$

test data — training data

posteriori probability — What we should choose

likelihood — What we just see

prior probability — What we knew previously

- X: a data sample ("*evidence*")
- H: X belongs to class C

Prediction can be done based on Bayes' Theorem:

Classification is to derive the maximum posteriori

Thanapong I.

Naïve Bayes Classifier: Training Dataset

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data to be classified:
X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

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

Training Data

‹#›

Naïve Bayes Classifier: Training Dataset

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data to be classified:
X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

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

Training Data

29

$$P^{y} = P(b^{y})\,P(a^{30}|b^{y})\,P(i^{m}|b^{y})\,P(s^{y}|b^{y})\,P(c^{f}|b^{y})$$

$$\frac{P(H^{-y}|X')}{P(H^{-N}|X')} = ?$$

$$= P(X|H^{-y})\,P(H^{-y})$$

training Data

Naïve Bayes Classifier: An Example

- P(Ci): P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute P(X|Ci) for each class

P(age = "<=30"|buys_computer = "yes") = 2/9 = 0.222
P(age = "<= 30"|buys_computer = "no") = 3/5 = 0.6
P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
P(student = "yes" | buys_computer = "yes") = 6/9 = 0.667
P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

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

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

P(X|Ci) : P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

P(X|Ci)*P(Ci) : P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

$\frac{3}{9} \times \frac{6}{9} \times \frac{9}{14} = 0.33$

$\hat{X}$ = age = <42 , student = yes ?

$P(H|\hat{X}) = ?$

$P(H_{buy=Y} | age=<42, stdn=yes) = P(age=<42 | buy=Y) P(stdn=yes | buy=Y) P(buy=Y)$

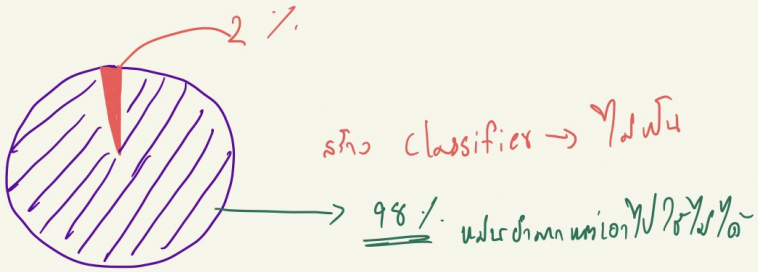
$\frac{3}{9} \times \frac{6}{9} \times 9/14$

$P(H_{buy=N} | (age=<42, stdn=yes) =$

Model Evaluation and Selection

- Evaluation metrics
  - How can we measure accuracy?
  - Other metrics to consider?
- Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier's accuracy
  - Holdout method
  - Cross-validation
  - Bootstrap
- Comparing classifiers:
  - ROC Curves

‹#›



47

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- **Precision:** Exactness: what % of tuples that the classifier labeled as positive are actually positive?

$$\text{P} = \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall:** Completeness: what % of positive tuples did the classifier label as positive?

$$\text{R} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  - Range: [0, 1]
- The "inverse" relationship between precision & recall
- ***F* measure (or *F*-score):** harmonic mean of precision and recall
  - In general, it is the weighted measure of precision & recall

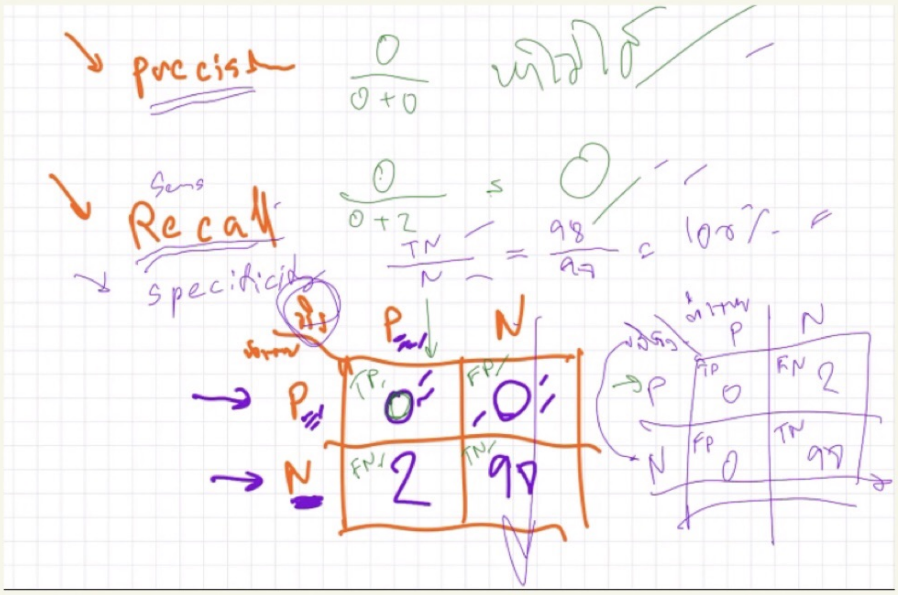
$$F_\beta = \frac{1}{\alpha \cdot \frac{1}{\text{P}} + (1 - \alpha) \cdot \frac{1}{\text{R}}} = \frac{(\beta^2 + 1)\text{PR}}{\beta^2\text{P} + \text{R}}$$

Assigning β times as much weight to recall as to precision)

- ***F1-measure (balanced F-measure)***
  - That is, when β = 1,

$$F_1 = \frac{2\text{PR}}{\text{P} + \text{R}}$$

48



ทำนายว่าเป็น 1 แล้วถูก

สิ่ง \ ทำนาย	0	1
0	[30, TP	6] FN
1	[2, FP	48] TN

ทำนายว่าเป็น 0 แล้วถูก

0 = Moligm

1 = Bcnign

ใช้กลางเพื่อคิดกับ

ใช้ 1 เป็น pos

ใช้ 0 เป็น pos

Sens = ตอบ pos ถูก 0.83

Spec = ตอบ Neg ถูกเป็นเท่าไหร่ $\left(\frac{48}{48+2}\right)$

	precision	recall	f1-score	support
0	0.94	0.83	0.88	36
1	0.89	0.96	0.92	50
accuracy			0.91	86
macro avg	0.91	0.90	0.90	86
weighted avg	0.91	0.91	0.91	86

จำนวนตัวเฉลี่ย 0

จำนวนตัวเฉลี่ย 1

เอาค่า P และ R มาหาค่าเฉลี่ย

$\frac{30}{30+2}$ $\frac{30}{30+6}$ $\frac{49}{49+6}$ $\frac{49}{49+2}$ $\frac{0.83+0.96}{2}$ $\frac{49+30}{86}$

$\frac{(0.83*36)+(0.96*50)}{36+50}$

- **Confusion Matrix:**

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

- In a confusion matrix w. *m* classes, $CM_{i,j}$ indicates # of tuples in class *i* that were labeled by the classifier as class *j*
  - May have extra rows/columns to provide totals
- **Example of Confusion Matrix:**

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- **Confusion Matrix:**

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

- In a confusion matrix w. *m* classes, $CM_{i,j}$ indicates # of tuples in class *i* that were labeled by the classifier as class *j*
  - May have extra rows/columns to provide totals
- **Example of Confusion Matrix:**

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

- **Classifier accuracy,** or recognition rate
  - Percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All
- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **Class imbalance problem**
  - One class may be *rare*
    - E.g., fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - Measures handle the class imbalance problem
    - **Sensitivity** (recall): True positive recognition rate
      - **Sensitivity = TP/P**
    - **Specificity**: True negative recognition rate
      - **Specificity = TN/N**

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- **Precision**: Exactness: what % of tuples that the classifier labeled as positive are actually positive?

$$P = \text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** Completeness: what % of positive tuples did the classifier label as positive?

$$R = \text{Recall} = \frac{TP}{TP + FN}$$

  - Range: [0, 1]
- The "inverse" relationship between precision & recall
- ***F* measure (or *F*-score):** harmonic mean of precision and recall
  - In general, it is the weighted measure of precision & recall

- ***F1-measure (balanced F-measure)***
  - That is, when β = 1,

$$F_1 = \frac{2PR}{P + R}$$