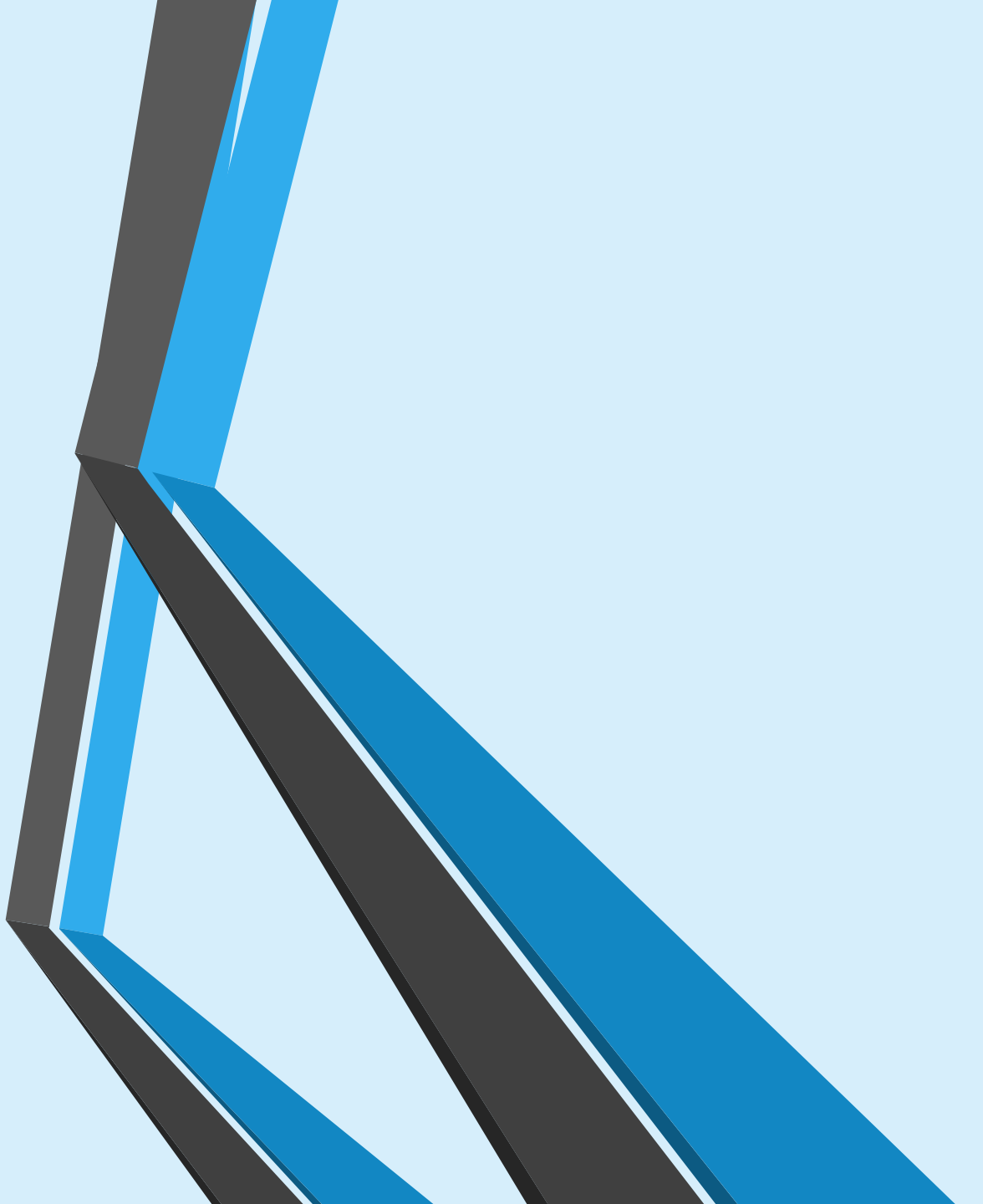


Data Mining Techniques

Classification

Unit IV

Topics to be Covered

Statistical
- Regression
- Bayesian

Distance
- Simple
- K Nearest Neighbors

Decision
- ID3
- CART
- C4.5

Neural Networks
- Perceptron
- Radial Basis Function

Rule based Algorithms
- Generating Rules from DT and NN

Combining Techniques

Introduction

Database **D = $\{t_1, t_2, t_3 \ldots. t_n\}$** of tuples and a set of classes **C=$\{C_1, \ldots, C_m\}$** the classification problem is to define a mapping **f: D→C** where each t_i is assigned to one class. A class, C_j contains precisely those tuples mapped to it; **$C_j = \{t_i / f(t_i) = C_j, 1 \leq i \leq n\}$.**

- Classification is a viewed as a **mapping** from the **database to the set of classes**.
- The classes in the classification problem are **equivalence classes**.

Classification vs Regression

Classification - process of finding a function - helps in dividing the dataset into classes – using different parameters.

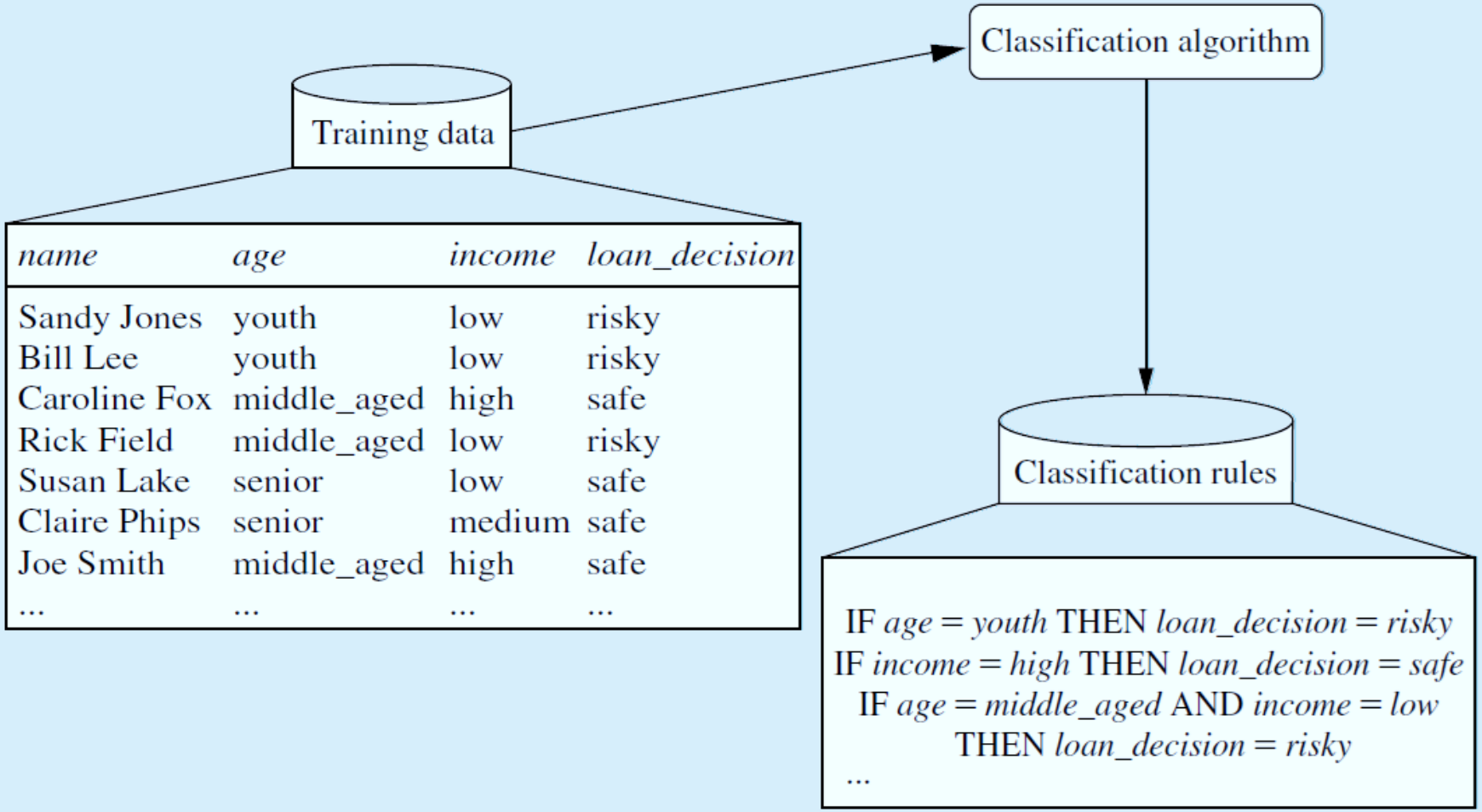
A computer program is trained on the training dataset - categorizes the test data samples (unseen samples) into different classes.

Regression - process of finding the correlations between dependent and independent variables.

Predict - continuous variables -Market Trends, House prices, etc.

Find the mapping function - map the input variable(x) to the continuous output variable(y).

Classification vs Regression



name	*age*	*income*	*loan_decision*
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

Issues in Classification

The preprocessing steps may be applied to the data for classification and prediction are : Data cleaning ,feature selection, and data transformation.

Data cleaning: This preprocesses the data in order to reduce noise and handle missing values.

Data transformation: it is used to generalize or normalize data.

Relevance analysis: Removes irrelevant or redundant attribute.

Evaluation of Classification Algorithms

Speed: Time to construct the model and also time to use the model.

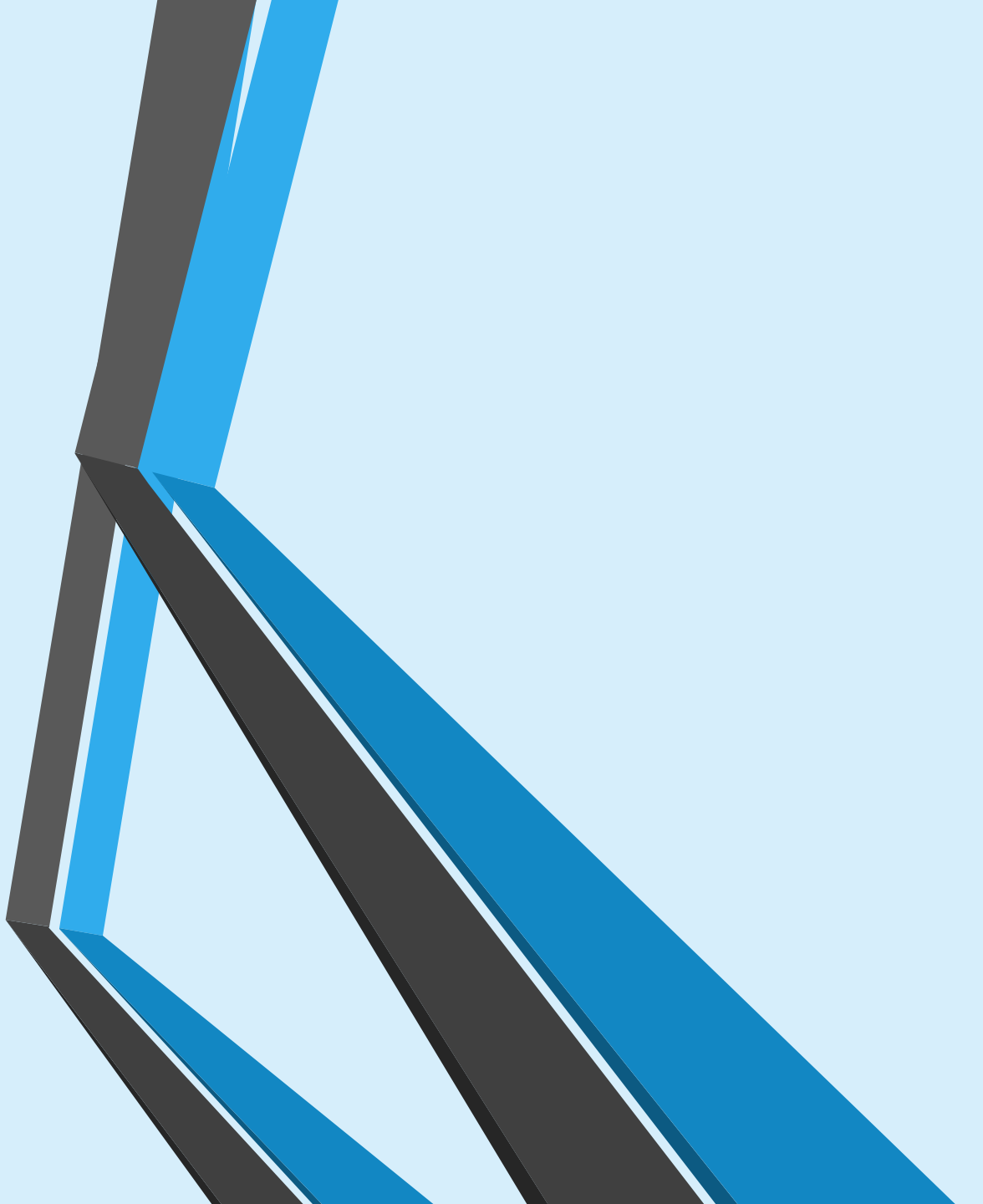
Robustness: Classifier to make correct predictions given noisy data or data with missing values

Scalability: Construct the classifier efficiently given large amounts of data.

Interpretability: Level of understanding and insight provided by the classifier.

Accuracy: Percentage of samples that are correctly classified.

Thank You



Data Mining Techniques

Classification

Statistical Methods

Unit IV

Bayesian Classification

- Bayesian classifiers are statistical classifiers.
- Naïve → Assumes the independence between the various attribute values.
- Predict class membership probabilities such as the probability that a given tuple belongs to a particular class.
- Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.
- X – Evidence, H – Hypothesis; $P(X|H)$ - Likelihood
- P(H|X) – Posterior Probability- Hypothesis holds the given evidence or observed data tuple. Posterior probability of H conditioned on X.
- P(H) – Prior Probability – This is an independent of X.

$$P(H|X) = \frac{P(X|H) * P(H)}{P(X)}$$

Bayesian Classification

Let D be a training set of tuples and their associated class labels.
Each tuple is represented by an n-dimensional attribute vector, X= $\{X_1, X_2, X_3 \ldots X_n\}$ n measurements made on n attributes.
classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X.

$$P(C_i \mid X) > P(C_j \mid X)$$

We maximize $P(C_i \mid X)$ (Maximum Posteriori Hypothesis)

$$P(C_i \mid X) = \frac{P(X \mid C_i) * P(C_i)}{P(X)}$$

P(X) is constant across all classes. So, $P(X \mid C_i) * P(C_i)$ is to be maximized.
If the class prior probabilities $P(C_i)$ are not known well in advance; it is assumed to be equally likely.

$$P(X \mid C_i) = \prod_{k=1}^{n} P(X_k \mid C_i) = P(X_1 \mid C_i) * P(X_2 \mid C_i) * P(X_3 \mid C_i) * \ldots\ldots P(X_n \mid C_i)$$

Bayesian Classification

age	income	student	Credit rating	buys_ computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle	medium	no	excellent	yes
middle	high	yes	fair	yes
senior	medium	no	excellent	no

Find out the class label for the following test sample.
X = (age = youth, income = medium, student = yes, credit rating = fair)

C1: class buys computer = Yes
C2: class buys computer = No

P(buys computer = yes) = 9/14 = 0.643
P(buys computer = no) = 5/14 = 0.357

Bayesian Classification

P(age = youth | buys computer = yes) = 2/9 = 0.222
P(age = youth | 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

P(X | buys computer = yes) =0.222*0.444*0.667*0.667=0.044
P(X | buys computer = no) =0.6*0.4*0.2*0.4=0.019

Bayesian Classification

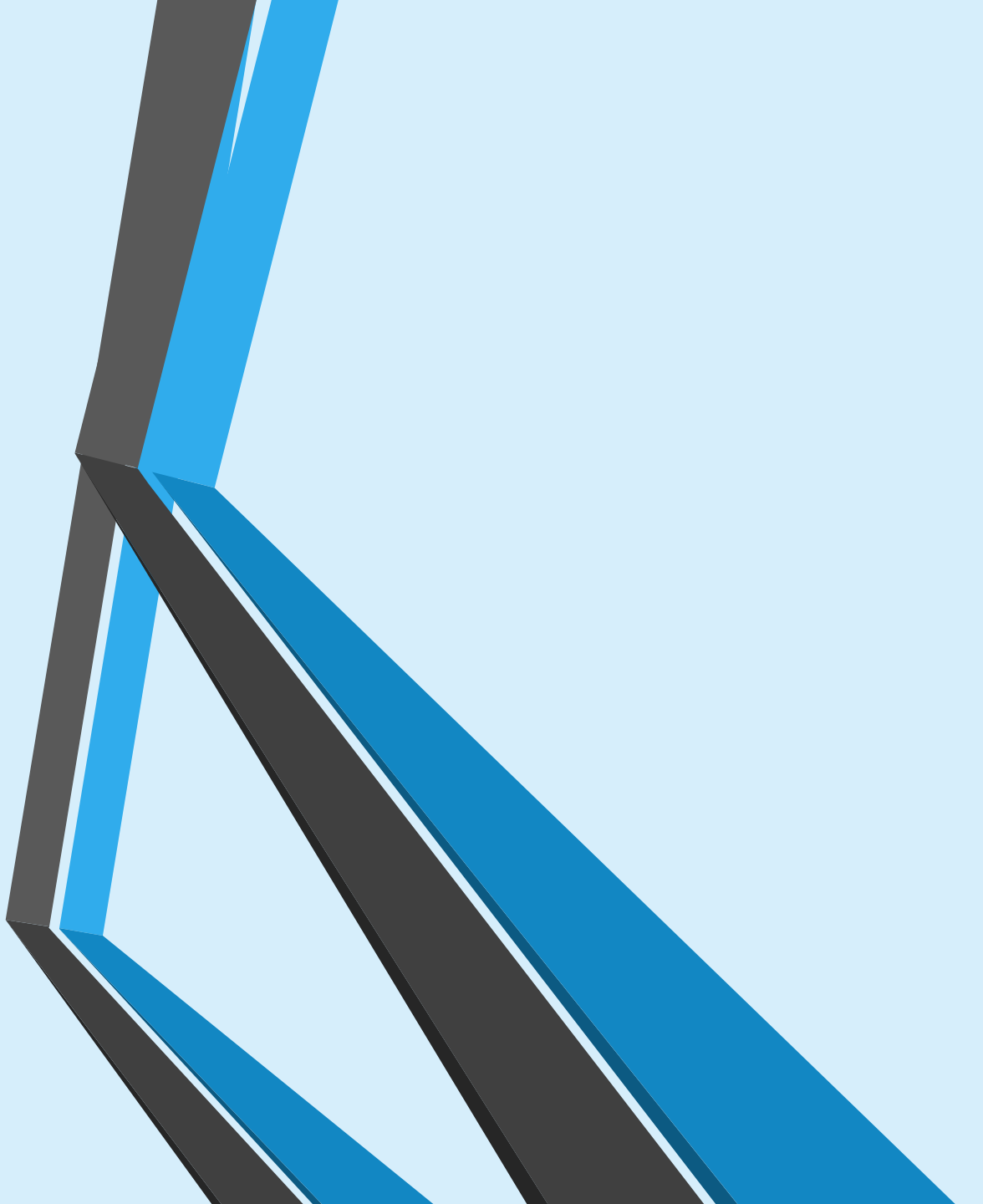
To find the class C_i that maximizes $P(X|C_i)*P(C_i)$

P(X | buys computer = yes) P(buys computer = yes)= 0.044*0.643=0.028

P(X | buys computer = no) P(buys computer = no)= 0.019*0.357=0.007

Therefore, the naive Bayesian classifier predicts as

buys computer = yes



Data Mining Techniques

Classification

Statistical Methods

Regression

Regression

In Linear regression, data is modelled using a straight line.
Bivariate linear regression models a random variable, Y (response variable) as a function of another random variable X (Predictor variable).

$$Y=\alpha+\beta X$$

where the variance of Y is considered as constant and α, β are regression coefficients specifying the Y intercept and slope of the line respectively. These coefficients are solved using method of least squares.

$$\beta = \frac{\sum_{i=1}^{s}(xi-\bar{x})(yi-\bar{y})}{\sum_{i=1}^{s}(xi-\bar{x})^2}$$

$$\alpha = \bar{y} - \beta\bar{x}$$

where $\bar{x}, \bar{y}$ is the averages of $x_1, x_2 \ldots x_s$; $y_1, y_2 \ldots y_s$ respectively.

Regression

X (Years of Exp)	3	8	9	13	3	6	11	21	1	16
Y (Salary)	30	57	64	72	36	43	59	90	20	83

$$\beta=\frac{(3-9.1)(30.55.4)+(8-9.1)(57-55.4)+(9-9.1)(64-55.4)+\cdots\ldots+(16-9.1)(83-55.4)}{(3-9.1)^2+(8-9.1)^2+(9-9.1)^2+\cdots\ldots+(16-9.1)^2}=3.5$$

$$\alpha=55.4-(3.5)9.1=23.6$$

by replacing the values of α, β

$$Y=23.6+3.5*X$$

Salary of agraduate with 10 years of experience as 58.6 thousands.

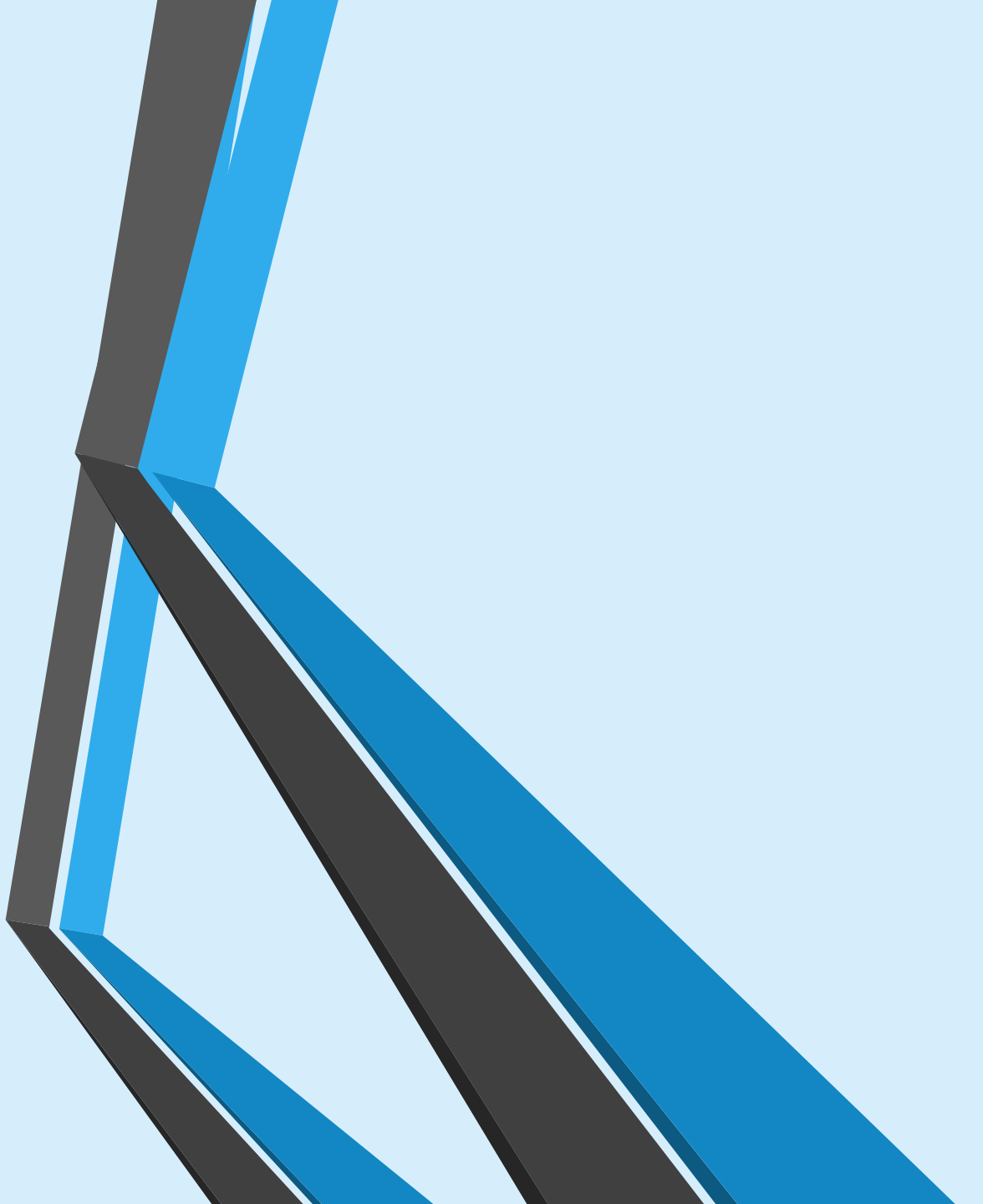
Multiple Regression

Multiple regression is an extension of linear regression involving more than one predictor variable.
Response variable Y is modelled as a linear function of multidimensional feature vector.

$$Y=\alpha+\beta_1 X_1+\beta_2 X_2$$

Polynomial regression can be modelled by adding polynomial terms to the basic linear model.
By applying transformations to the variables, nonlinear model can be converted into linear model.

Thank You



Data Mining Techniques

Classification

Distance Methods

Simple Approach and KNN

Simple Approach

Input:
$C_1, C_2 \ldots C_m$ // Centres for each class
t // Input tuple to classify
Output:
C // Class to which T is assigned
// Simple Distance-based Algorithm//
dist=∞
for i=1 to m do
If dis(C_i,t) < dist then
C=i;
dist=dist(C_i,t);

K Nearest Neighbor Classifier

K-Nearest Neighbour classifier searches the pattern space for the k training samples and identifies the closest to the unknown sample. The closeness among the samples are defined in terms of Euclidean distance.

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(xi-yi)^2}$$

k=1, unknown sample is assigned to the training sample that is closest in sample space.

$$d(x,y)=\sqrt[P]{(xi-yi)^P}$$

P=1 Manhattan Distance
P=2 Euclidean Distance
P=>3 Minkowski Distance

K Nearest Neighbor Classifier

Pros of KNN

- Simple and very easy to understand.
- Handles nonlinear data also.
- Applicable for both classification and regression.

Cons of KNN

- Need of high storage.
- Prediction rate is also slow.
- Stores all the training data.

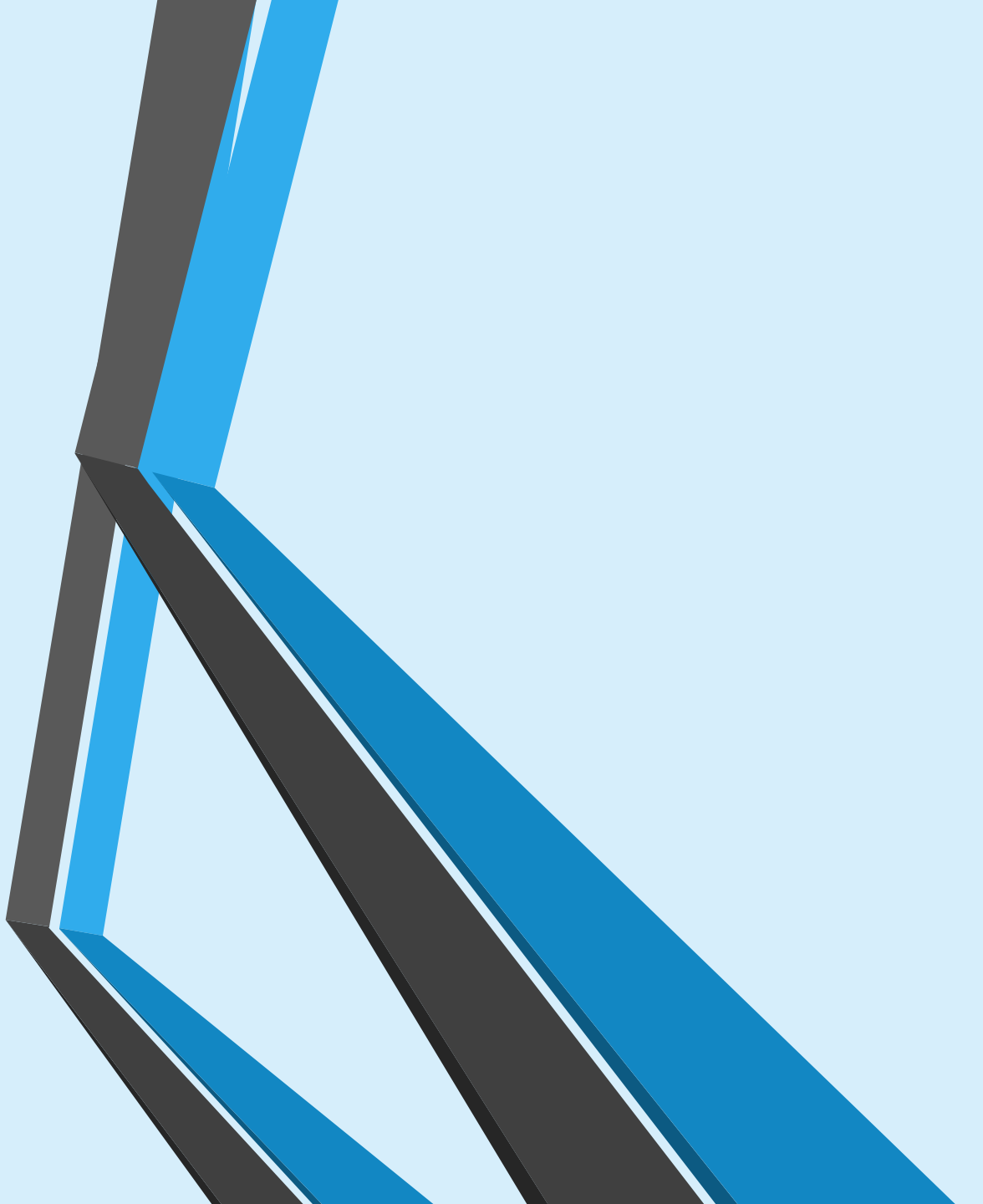
K Nearest Neighbor Classifier

Problem: In paper testing laboratory, quality of the paper is assessed with two attributes such as acid durability and its strength. Using 3-KNN compute the class label of the given paper sample with attributes values as (3,7)

Acid Durability	Strength	Class Label	Distance	Rank	3NN
7	7	Bad	4	3	Yes
7	4	Bad	5	4	No
3	4	Good	3	1	Yes
1	4	Good	3.6	2	Yes

Three nearest class labels are **Good, Good** and **Bad**.
Based on the higher frequency of Good, **Good** is assigned as class label for the given test sample.

Thank You



Data Mining Techniques

Classification

Decision Tree Induction

Decision Tree Induction

- Decision tree is a flow chart like tree structure
  - ✓ Each internal node denotes a test on an attribute and each branch represents an outcome of the test, and leaf nodes represent classes or class distributions.
- Top most node in a tree is the root node.
- Decision trees can easily be converted into classification rules.
- Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data.
- In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree.
  - ✓ A path is traced from the root to a leaf node that holds the class prediction for that sample.

Decision Tree Induction

Method:
Create a node N;
If samples are all of the same class, C then
Return N as a leaf node labeled with the class C;
If attribute – list is empty then
Return N as a leaf node labeled with the most common class in samples ; // majority voting
Select test attribute, the attribute among attribute – list with the highest information gain;
Label node N with test – attribute;
For each known value ai of test – attribute // partition the sample
Grow a branch from node N for the condition test – attribute = ai;
Let si be the set of samples in samples for which test – attribute = ai; // a partition
if si is empty then
attach a leaf labeled with the most common class in samples;
else attach the node returned by Generate_decision_tree

Decision Tree Induction (ID3)

Select the attribute with the highest information gain

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

Let p_i be the probability of 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:

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

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

Information gained by branching on attribute A

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

Decision Tree Induction (ID3)

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

$$\text{Info}(D) = -9/14 \log(9/14)_2 \\ -5/14 \log(5/14)_2 \\ = 0.94$$

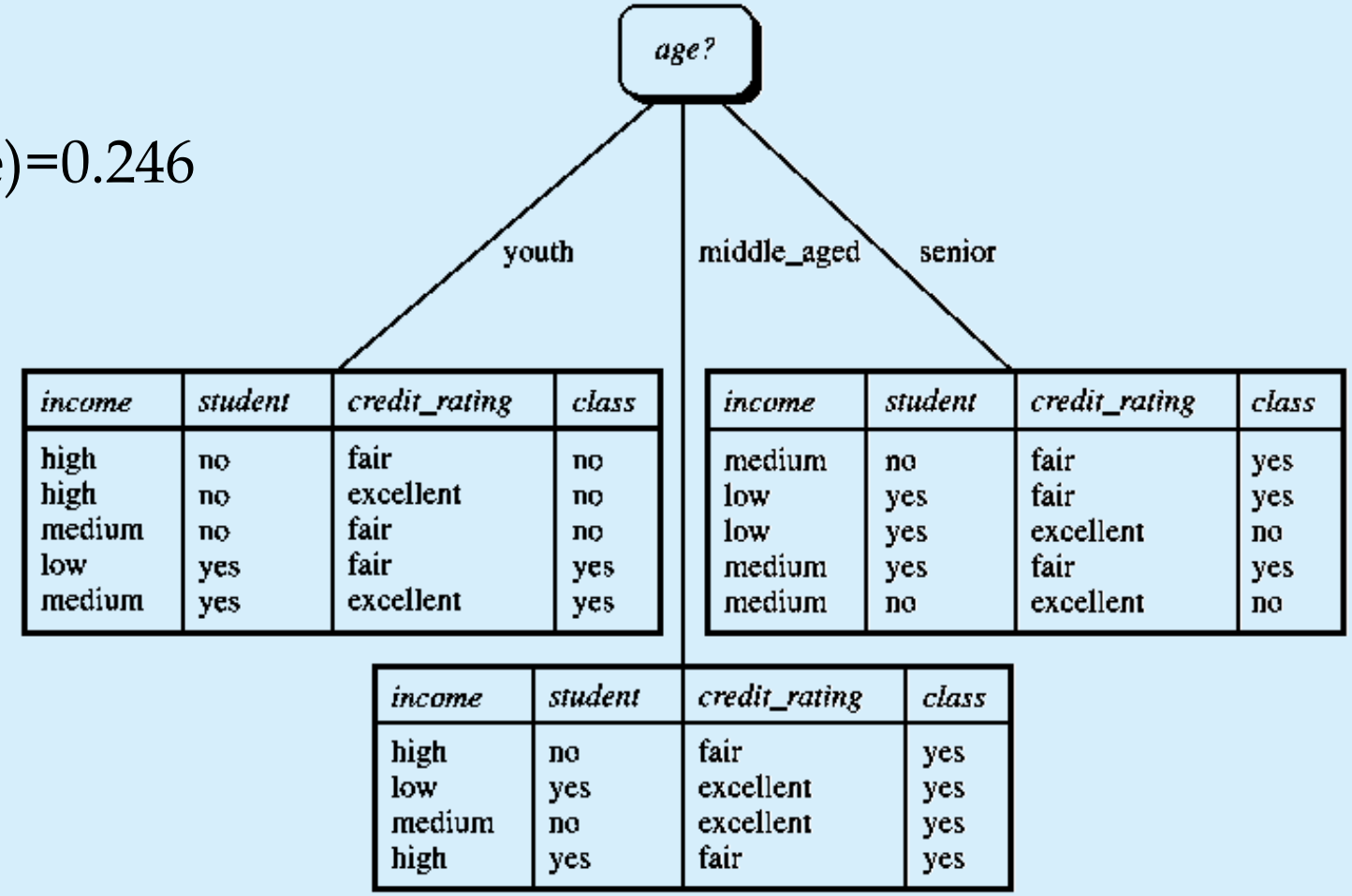
Decision Tree Induction (ID3)

Entropy(age)={5/14(-2/5 $\log(2/5)_2$-3/5 $\log(3/5)_2$) }
+ {4/14(-4/14 $\log(4/14)_2$-0/14 $\log(0/14)_2$)}
+ {5/14(-3/5 $\log(3/5)_2$-2/5 $\log(2/5)_2$)}
= 0.694

Gain(age)=Info(D)-E(age)=0.246

Gain (income)=0.029

Gain(student)=0.151



income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

Decision Tree Induction C4.5

The information gain measure is biased toward tests with many outcomes. Information Gain prefers to choose attributes with a large number of values.

C4.5, a successor of ID3, uses an extension to information gain known as gain ratio, which attempts to overcome this bias.

It applies normalization to information gain using a "split information".

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

It differs from information gain, which measures the information with respect to classification that is acquired based on the same partitioning.

$$\text{GainRatio(A)} = \frac{Gain(A)}{SplitInfo_A(D)}$$

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

Decision Tree Induction C4.5

$SplitInfo_{income}$(D)= -4/14 $\log(4/14)_2$-6/14 log $(6/14)_2$-4/14 $\log(4/14)_2$

= 1.557

Gain(income)= 0.029

GainRatio(income) = 0.029/1.557=0.019

Decision Tree Induction CART

The Gini index is used in CART.
Gini index measures the impurity of D, a data partition or set of training tuples, as

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} P_i^2$$

Pi is the probability of a tuple in D belongs to class C_i

The Gini index considers a binary split for each attribute. D_1 is satisfying A ≤split_point and D_2 is satisfying A>split_point.

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

$$\text{Gini(A)} = \text{Gini(D)} - Gini_A(D)$$

Decision Tree Induction CART

$$\text{Gini}(D) = 1-(\frac{9}{14})^2-(\frac{5}{14})^2= 0.459$$

$$Gini_{income \in \{low,medium\}}.D/$$

$$= \frac{10}{14} Gini.D_1/ + \frac{4}{14} Gini.D_2/$$

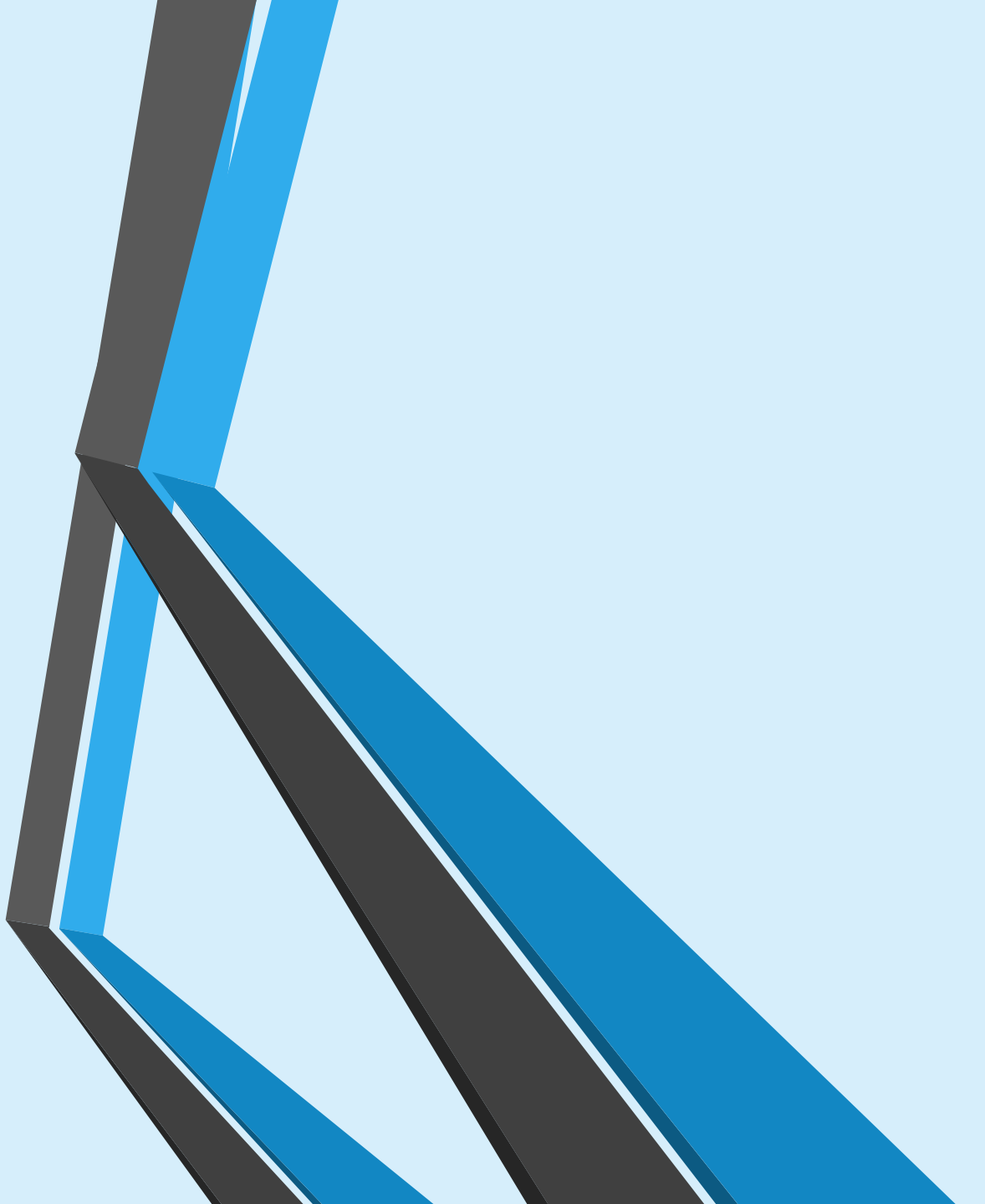
$$= \frac{10}{14}\left(1-\left(\frac{7}{10}\right)^2-\left(\frac{3}{10}\right)^2\right)+\frac{4}{14}\left(1-\left(\frac{2}{4}\right)^2-\left(\frac{2}{4}\right)^2\right)$$

$$= 0.443$$

$$= Gini_{income \in \{high\}}.D/.$$

Similarly, the Gini index values for splits on the remaining subsets are 0.458 (for the subsets (low, high) and medium; 0.450 (for the subsets (medium, high) and low. Therefore, the best binary split for attribute income is on (low, medium) and high because it minimizes the Gini index.

Thank You



Data Mining Techniques

Classification

Neural Networks

Simple Perceptron, RBFNN

Perceptron

Perceptron is a simplified model of the real neuron

Perceptron takes the input signals, x1, x2, …, xn, computes a weighted sum z of those inputs, then passes it through a threshold function ϕ and outputs the result.

$$z = \sum_{i=1}^{n} x_i w_i$$

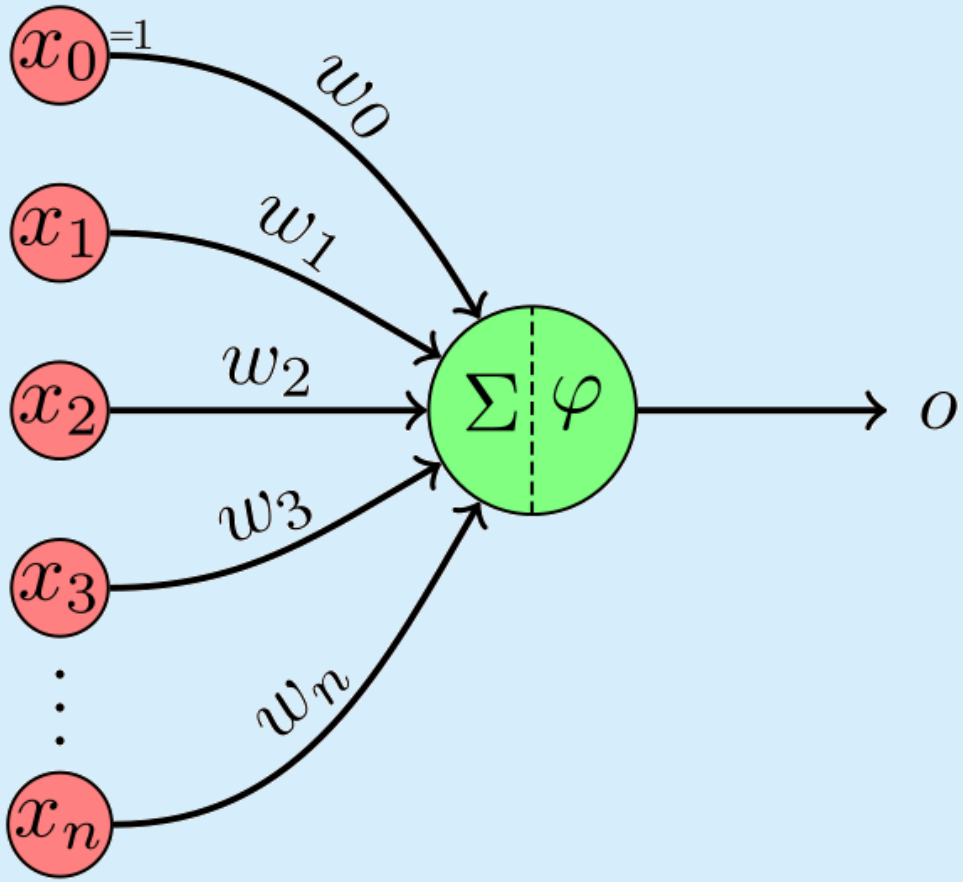
$$\phi(z) = \begin{cases} -1 & z \leq w_0 \\ 1 & z > w_0 \end{cases}$$

w0 as a threshold, adding w0 to the sum as bias and having as threshold 0. That is, we consider an additional input signal x0 that is always set to 1.

Perceptron

$$z = \sum_{i=0}^{n} x_i w_i$$

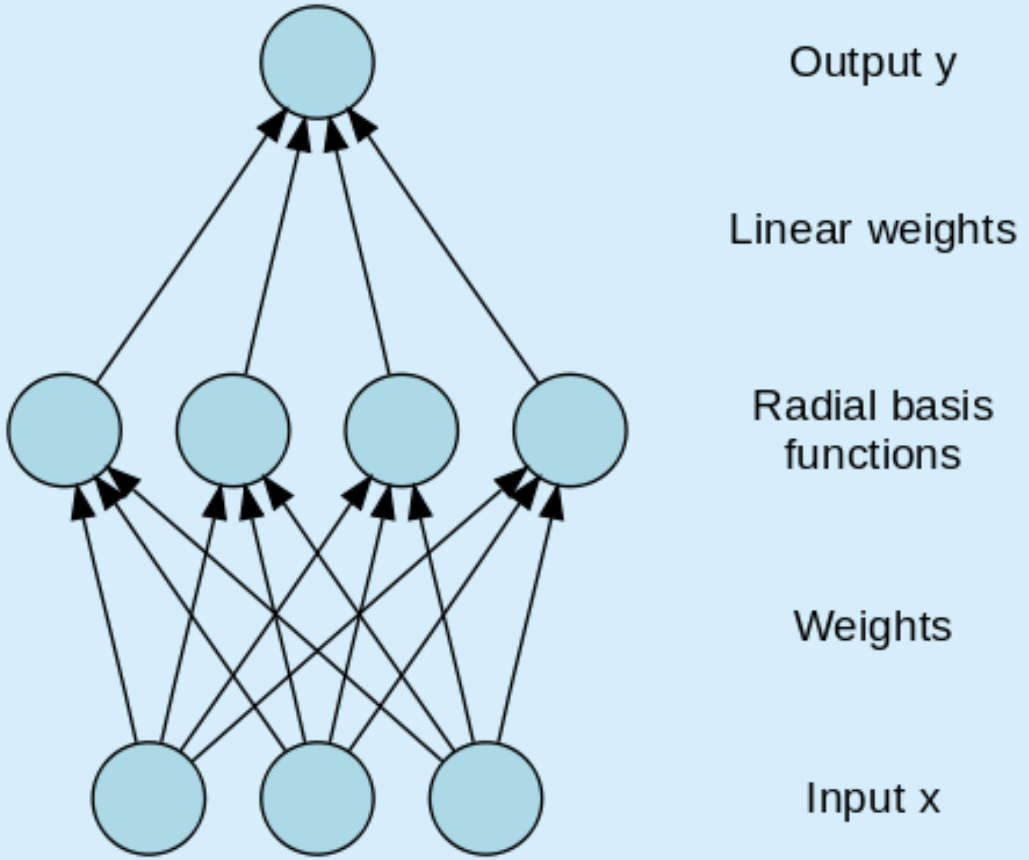
$$\phi(z) = \begin{cases} -1 & z \leq 0 \\ 1 & z > 0 \end{cases}$$



Redial Basis Function Neural Network

In **Single Perceptron / Multi-layer Perceptron (MLP)**, we only have linear separability because they are composed of input and output layers(some hidden layers in MLP).

For example, AND, OR functions are **linearly**-separable & XOR function is **not** linearly separable.

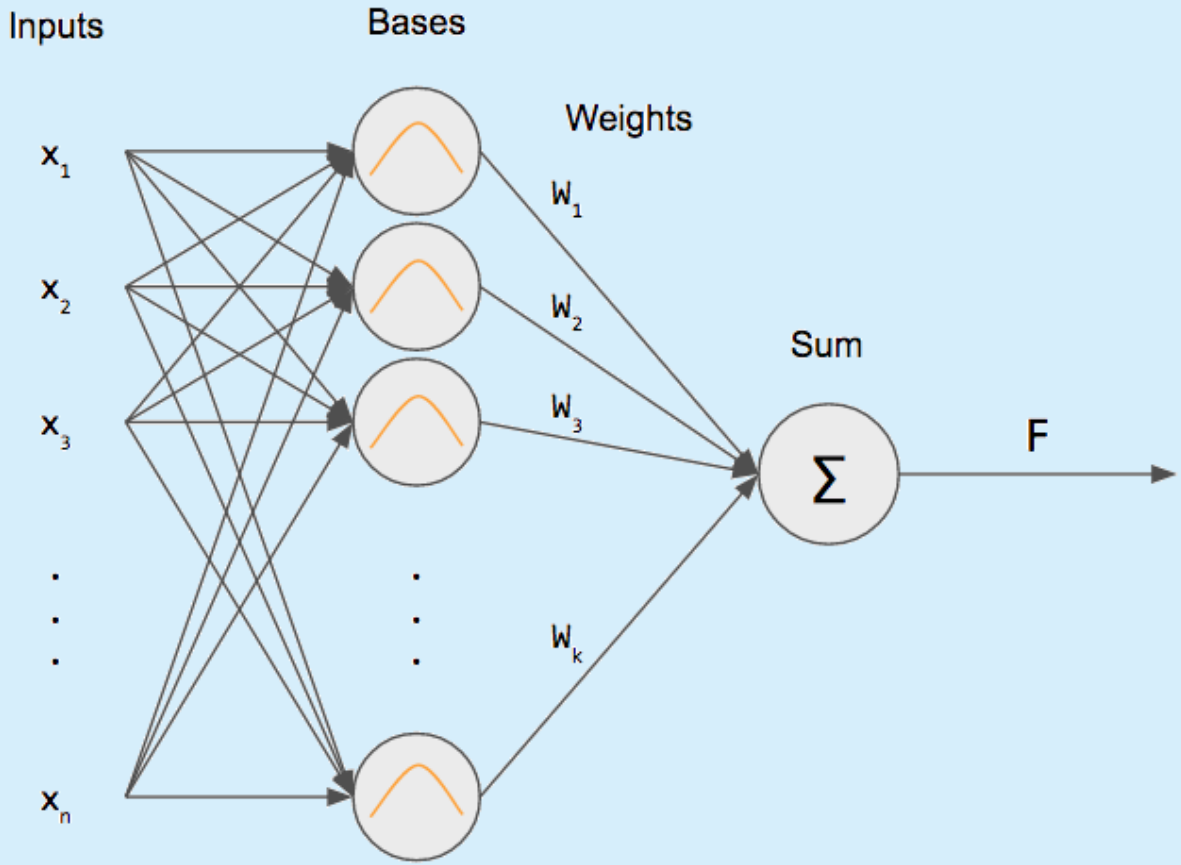


Radial Basis Function Neural Network

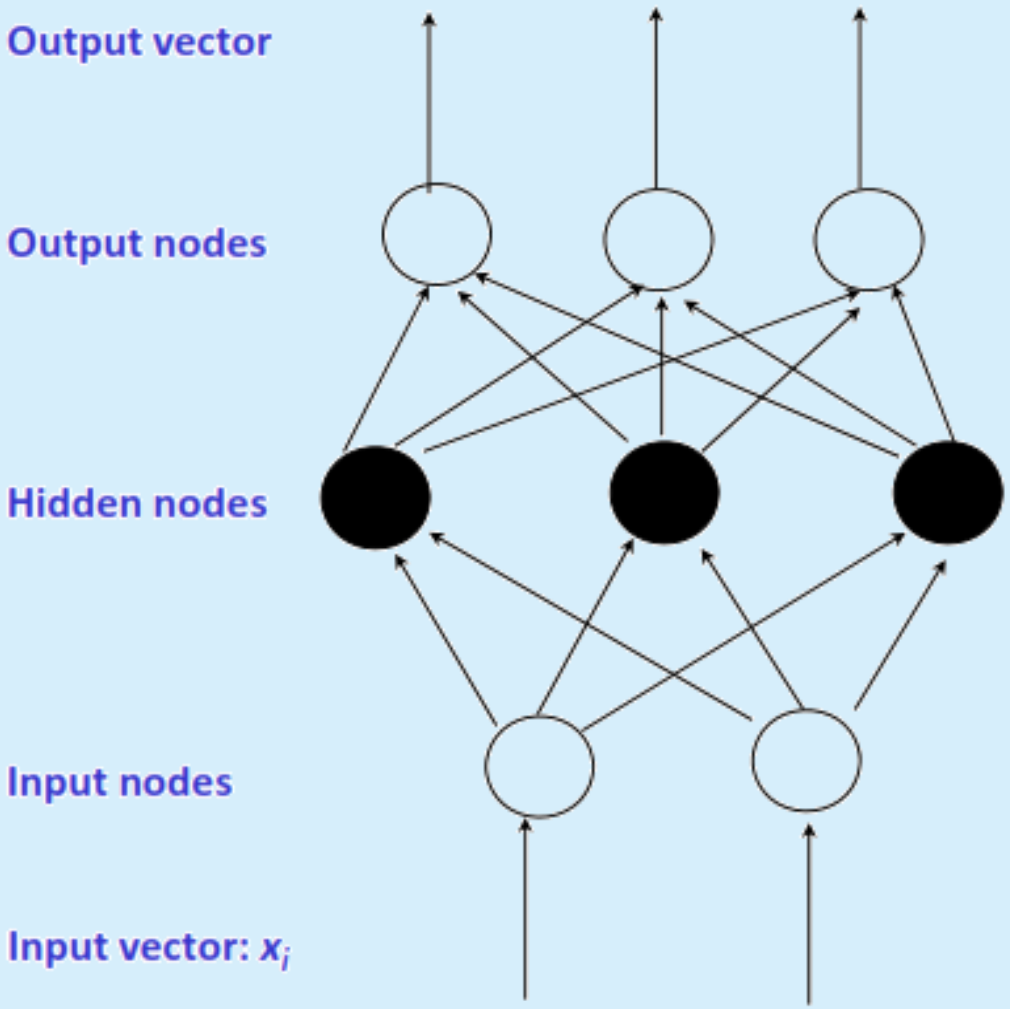
RBNN is composed of **input, hidden,** and **output** layer.

RBNN is **strictly limited** to have exactly **one hidden layer**.

Hidden layer is also termed as **feature vector**



Back Propagation Neural Network



$$\theta_j = \theta_j + (r)Err_j$$

to update the bias

$$w_{ij} = w_{ij} + (r)Err_j O_i$$

to update the weights

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

error for a node in the hidden layer

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

error for a node in the output layer

$$O_j = \frac{1}{1+e^{-I_j}}$$

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

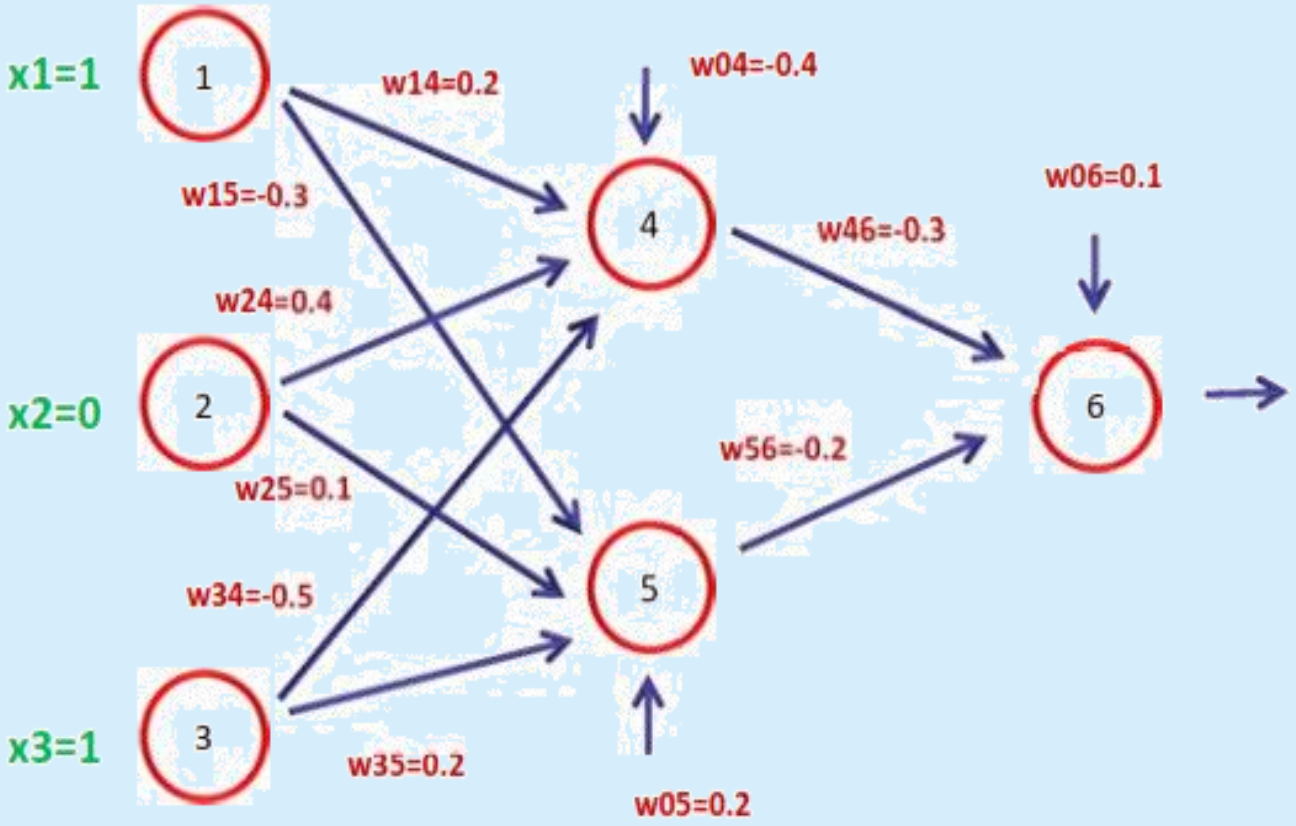
Terminating Conditions

Training stops when

- ✓ all Δw_{ij} in the previous epoch were so small as to be below some specified threshold.
- ✓ The percentage of samples misclassified in the previous epoch is below some threshold
- ✓ A pre specified number of epochs have expired.

Problem

X1	X2	X3	W14	W15	W24	W25	W34	W35	W46	W56	$\theta 4$	$\theta 5$	$\theta 6$	L
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1	0.9



Problem

The net input and output calculations

neuron	input	output
4	0.2x1+0.4x0-0.5x1-0.4=-0.7	1/(1+e0.7)=0.332
5	-0.3x1+0.1x0+0.2x1+0.2=0.1	1/(1+e-0.1)=0.525
6	-0.3x0.332-0.2x0.525+0.1=-0.105	1/(1+e0.105)=0.474

Calculation of error at each node Updation of weights and biases

neuron	error
6	0.474 x (1 - 0.474) x (1 - 0.474) = 0.1311
5	0.525 x (1 - 0.525) x (-0.2) x 0.1311 = -0.0065
4	0.332 x (1 – 0.332) x (-0.3) x 0.1311 = -0.0087

Problem

The net input and output calculations

neuron	input	output
4	0.2x1+0.4x0-0.5x1-0.4=-0.7	1/(1+e0.7)=0.332
5	-0.3x1+0.1x0+0.2x1+0.2=0.1	1/(1+e-0.1)=0.525
6	-0.3x0.332-0.2x0.525+0.1=-0.105	1/(1+e0.105)=0.474

Calculation of error at each node Updation of weights and biases

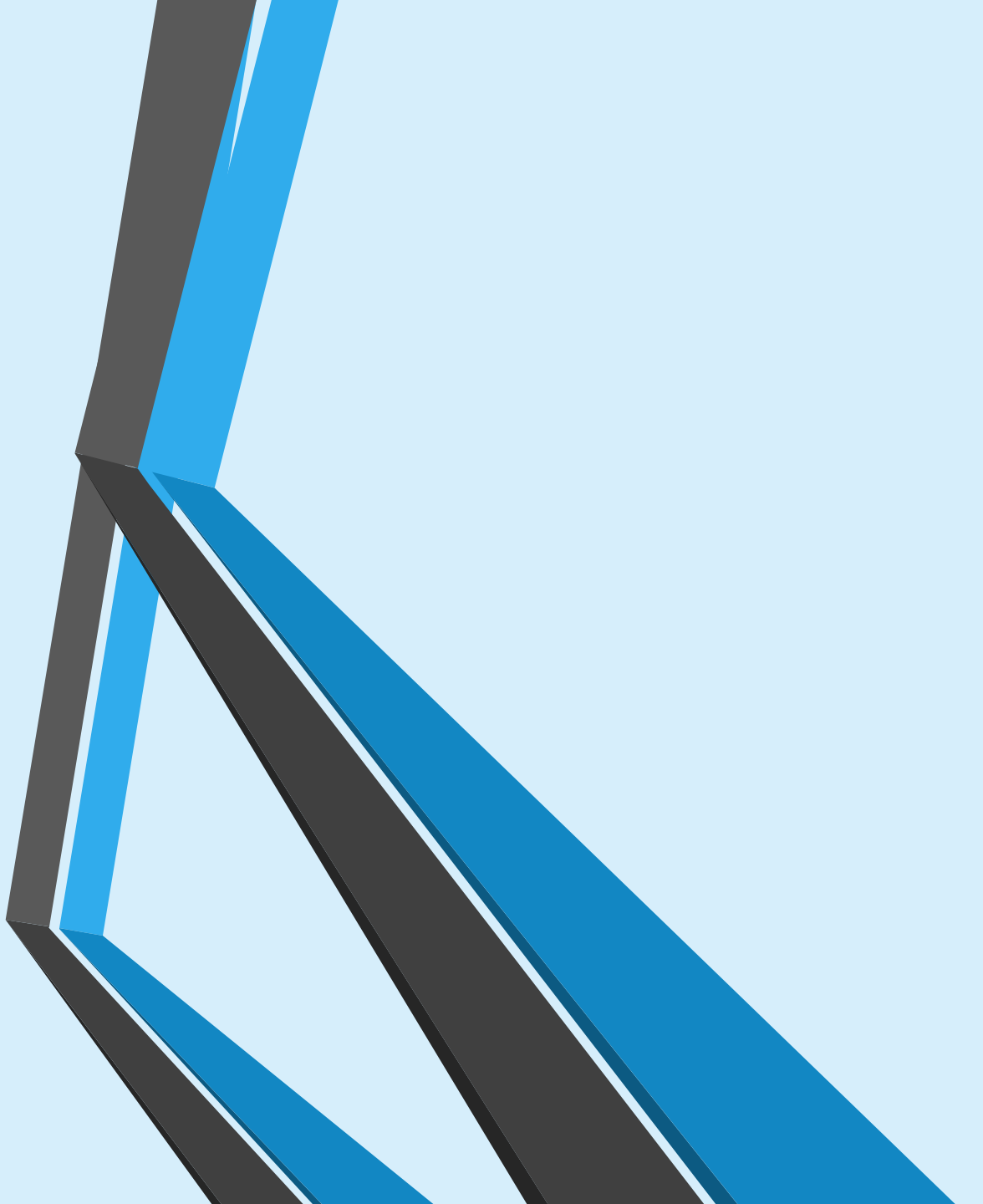
neuron	error
6	0.474 x (1 - 0.474) x (1 - 0.474) = 0.1311
5	0.525 x (1 - 0.525) x (-0.2) x 0.1311 = -0.0065
4	0.332 x (1 – 0.332) x (-0.3) x 0.1311 = -0.0087

Problem

Neuron	Output	Error
4	0.332	-0.0087
5	0.525	-0.0065
6	0.474	0.1311

weight	New value
w46	-0.3 + 0.9 x 0.1311 x 0.332 = -0.261
w56	-0.2 + 0.9 x 0.1311 x 0.525 = -0.138
w14	0.2 + 0.9 x -0.0087 x 1 = 0.192
w15	-0.3 + 0.9 x -0.0065 x 1 = -0.306
w24	0.4 + 0.9 x -0.0087 x 0 = 0.4
w25	0.1 + 0.9 x -0.0065 x 0 = 0.1
w34	-0.5 + 0.9 x -0.0087 x 1 = -0.508
w35	0.2 + 0.9 x -0.0065 x 1 = 0.194
$\theta 6$	0.1 + 0.9 x 0.1311 = 0.218
$\theta 5$	0.2 + 0.9 x -0.0065 = 0.194
$\theta 4$	-0.4 + 0.9 x -0.0087 = -0.408

Thank You



Data Mining Techniques

Classification

Generating Decision Rules

Generation of Decision Rules - OneR

- ✓ OneR algorithm generates one rule for each predictor in the data.
- ✓ It selects the rule with the smallest total error.
- ✓ To create a rule for a predictor, construct a frequency table for each predictor against the target.
- ✓ Low Accuracy

One R

No.	1: outlook	2: temperature	3: humidity	4: windy	5: play
	Nominal	Nominal	Nominal	Nominal	Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	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	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Attribute	Values	Play Golf	
		Yes	No
Outlook	Sunny	2	3
	Overcast	4	0
	Rainy	3	2
Temperature	Hot	2	2
	Mild	4	2
	Cool	3	1
Humidity	High	3	4
	Normal	6	1
Windy	False	3	2
	True	3	3

If Outlook=sunny then Play Golf = No

If Outlook= Overcast then Play Golf = Yes

OneR Algorithm

For each predictor,

For each value of that predictor, make a rule as follows;

Count how often each value of target (class) appears

Find the most frequent class

Make the rule assign that class to this value of the predictor

Calculate the total error of the rules of each predictor

Choose the predictor with the smallest total error.

Generating rules from Neural Net

Input:

D // Training data

N // Initial neural network

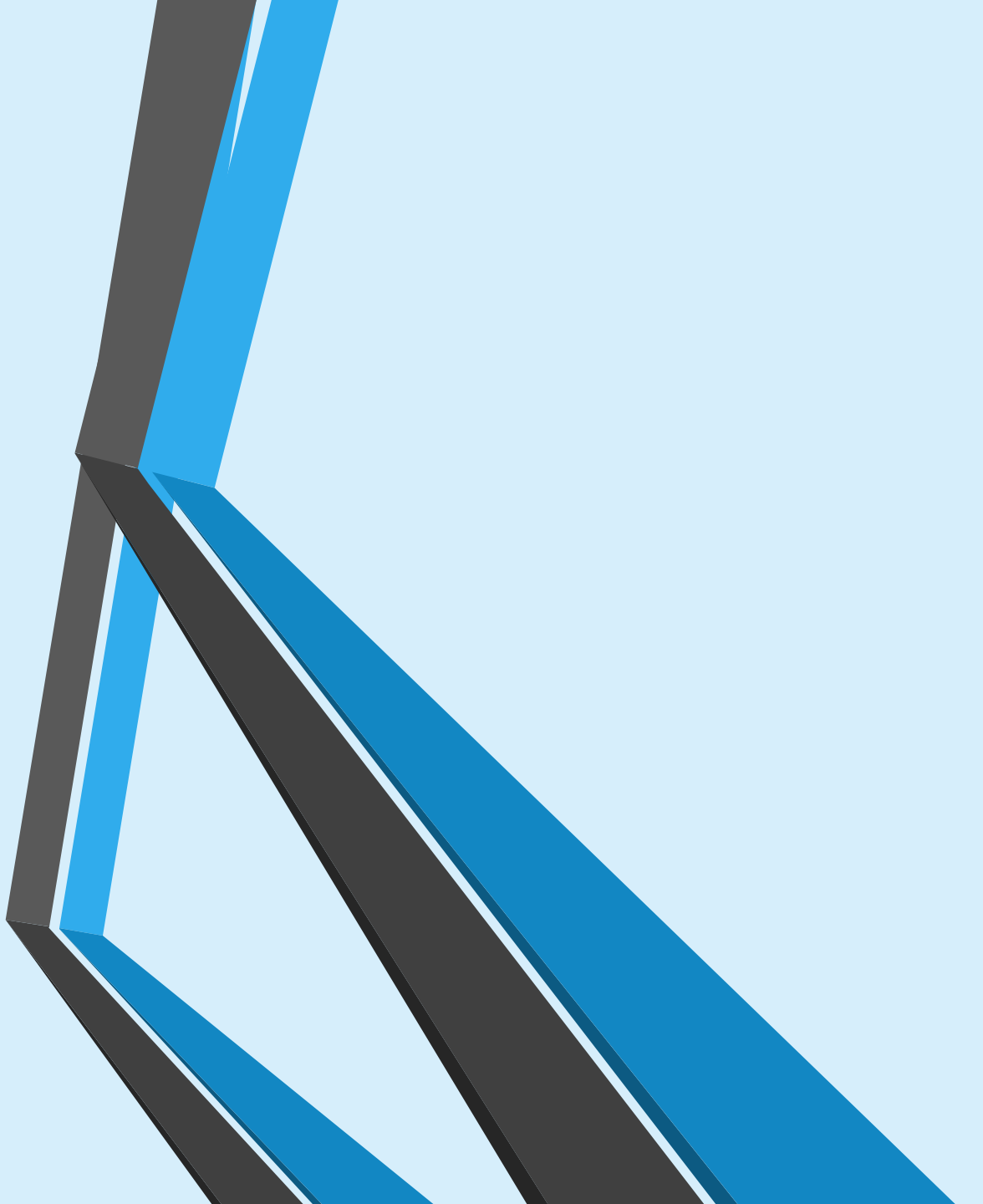
Output:

R // Derived rules

Algorithm:

- ✓ Cluster output node activation values;
- ✓ Cluster hidden nodes activation values;
- ✓ Generate rules that describe the output values in terms of the hidden activation values;
- ✓ Generate rules that describe hidden output values in terms of inputs
- ❖ Combine the two sets of rules

Thank You



Data Mining Techniques

Classification

Counting Techniques

Holdout Method and Random Subsampling

- ✓ In Holdout, the given data are randomly partitioned into two independent sets, a training set and a test set.
  - ❖ Typically, two-thirds of the data are allocated to the training set, and the remaining one-third is allocated to the test set.
- ✓ The training set is used to derive the model. The model's accuracy is then estimated with the test set.

- ✓ Random subsampling is a variation of the holdout method in which the holdout method is repeated k times.
- ✓ The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration.

Cross Validation

- ✓ In k-fold cross-validation, the initial data are randomly partitioned into k mutually exclusive subsets or "folds," D1, D2, … , D_k , each of approximately equal size.
- ✓ Training and testing is performed k times.
- ✓ In iteration i, partition Di is reserved as the test set, and the remaining partitions are collectively used to train the model.

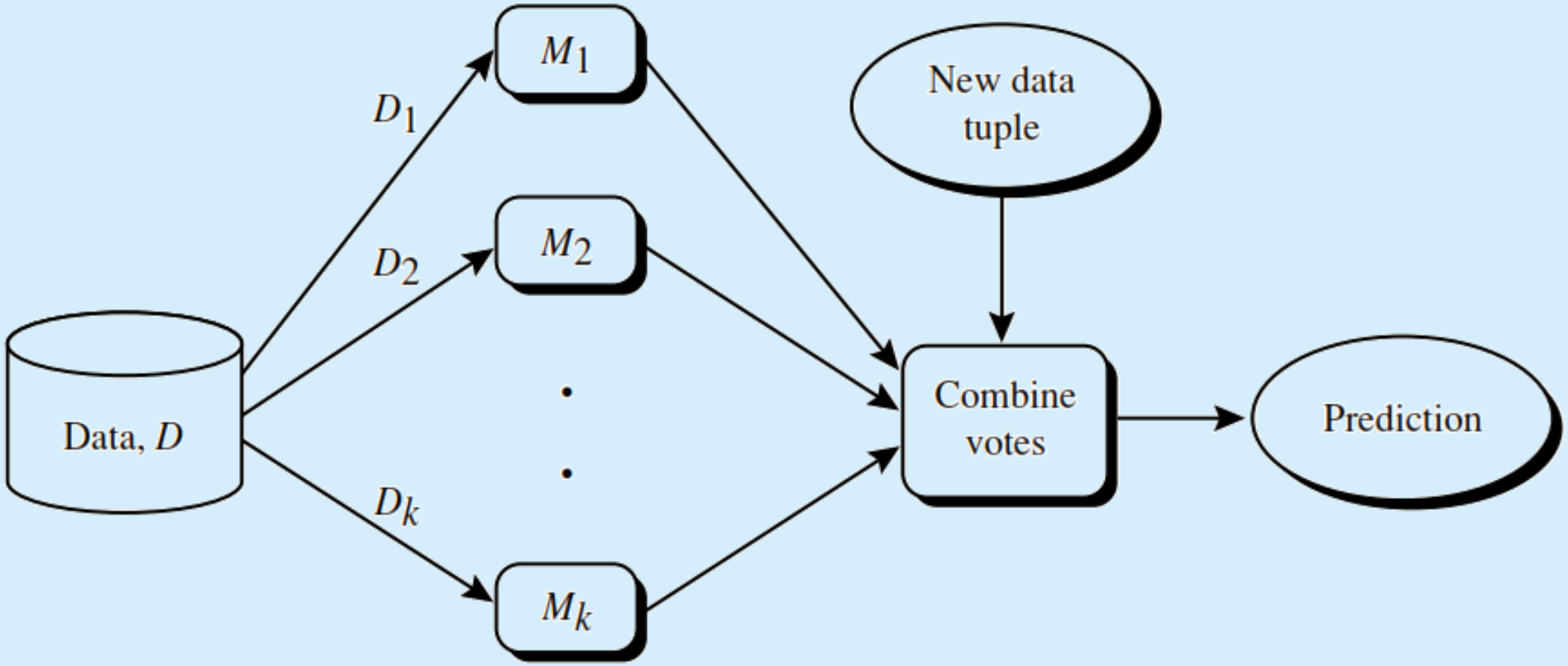
- ✓ Classification accuracy is estimated as the overall number of correct classifications from the k iterations.

- ✓ Stratified 10-fold cross-validation is recommended for estimating accuracy due to its relatively low bias and variance.

Ensemble Methods

- An ensemble for classification is a composite model, made up of a combination of classifiers.
- Given a new data tuple to classify, the base classifiers each vote by returning a class prediction. The ensemble returns a class prediction based on the votes of the base classifiers.
- Ensembles tend to be more accurate than their component classifiers.

Ensemble Methods



Ensemble Methods

The bagging algorithm—create an ensemble of classification models for a learning scheme where each model gives an equally weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a classification learning scheme (decision tree, naive Bayesian, etc.).

Output: The ensemble—a composite model, M.

Method:

(1) **for** i D 1 to k do // create k models:
(2) create bootstrap sample, D, by sampling D with replacement;
(3) use Dvand the learning scheme to derive a model, M;
(4) **endfor**

To use the ensemble to classify a tuple, X:
let each of the k models classify X and return the majority vote;

Thank You