Unit 2 -Modelling and Evaluation

Ms. Geetanjali R
Assistant Professor,
Department of MCA
Ramaiah Institute of Technology

Introduction

- Representation of raw input data to the meaningful pattern is called a model.

- The process of assigning a model, and fitting a specific model to a data set is called model training.

SELECTING A MODEL

- Input variables – they are also called predictors, attributes, features, independent variables, or simply variables

- Output variables - also called response or dependent variable.

- Input variables can be denoted by X, while individual input variables are represented as x1,x2,x3 .. Output variable by symbol Y.

SELECTING A MODEL

- **The relationship between X and Y is represented in the general form: $Y = f(X) + e$, where 'f ' is the target function and 'e' is a random error term.**

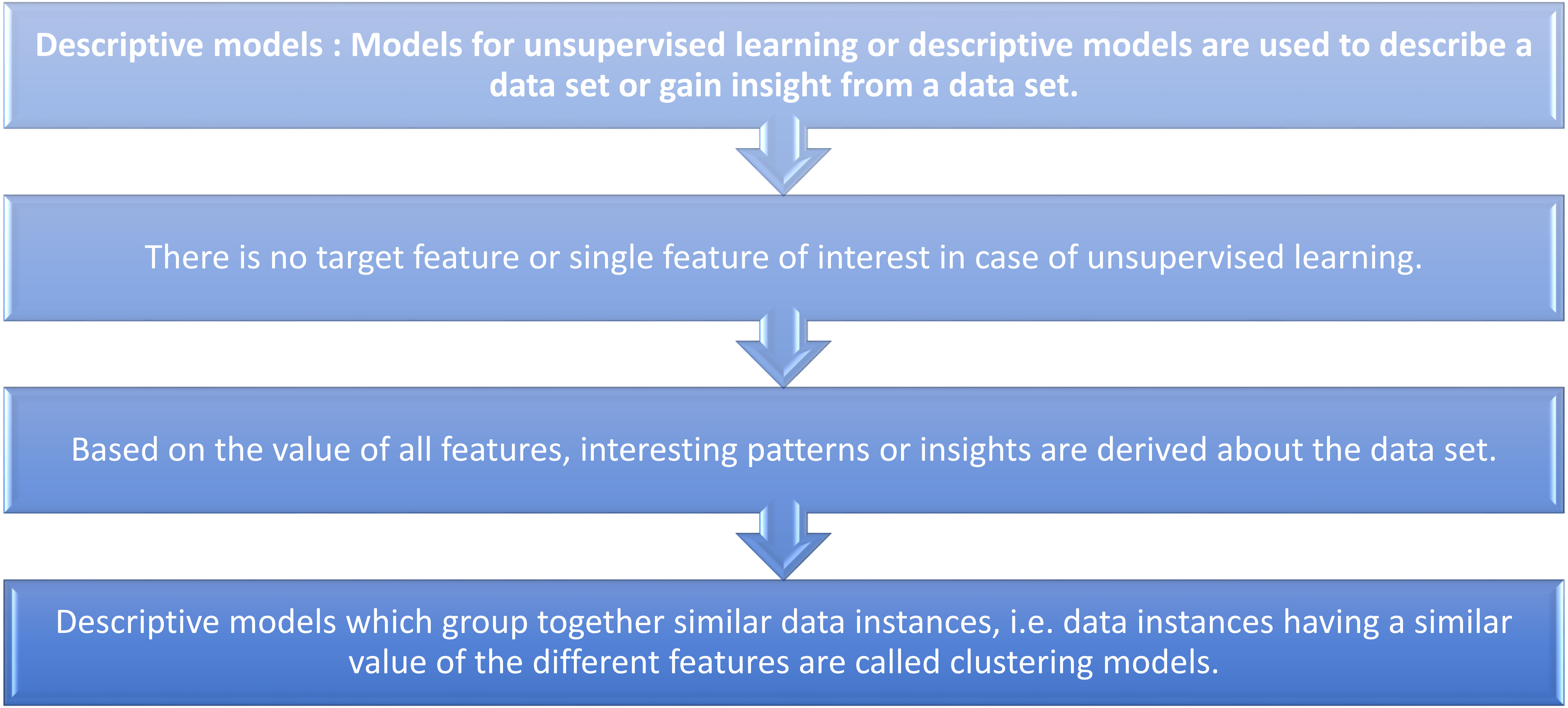
SELECTING A MODEL

Predictive models : Try to predict certain value using the values in an input data set.

The learning model attempts to establish a relation between the target feature, i.e. the feature being predicted, and the predictor features.

The predictive models have a clear focus on what they want to learn and how they want to learn

SELECTING A MODEL



Descriptive models : Models for unsupervised learning or descriptive models are used to describe a data set or gain insight from a data set.

There is no target feature or single feature of interest in case of unsupervised learning.

Based on the value of all features, interesting patterns or insights are derived about the data set.

Descriptive models which group together similar data instances, i.e. data instances having a similar value of the different features are called clustering models.

TRAINING A MODEL (FOR SUPERVISED LEARNING)

However, how can we understand the performance of the model? The test data may not be available immediately. Also, the label value of the test data is not known.

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- **That is the reason why a part of the input data is held back (that is how the name holdout originates) for evaluation of the model.**

- **However, a different proportion of dividing the input data into training and test data is also acceptable.**

- **To make sure that the data in both the buckets are similar in nature, the division is done randomly. Random numbers are used to assign data items to the partitions.**

TRAINING A MODEL (FOR SUPERVISED LEARNING)

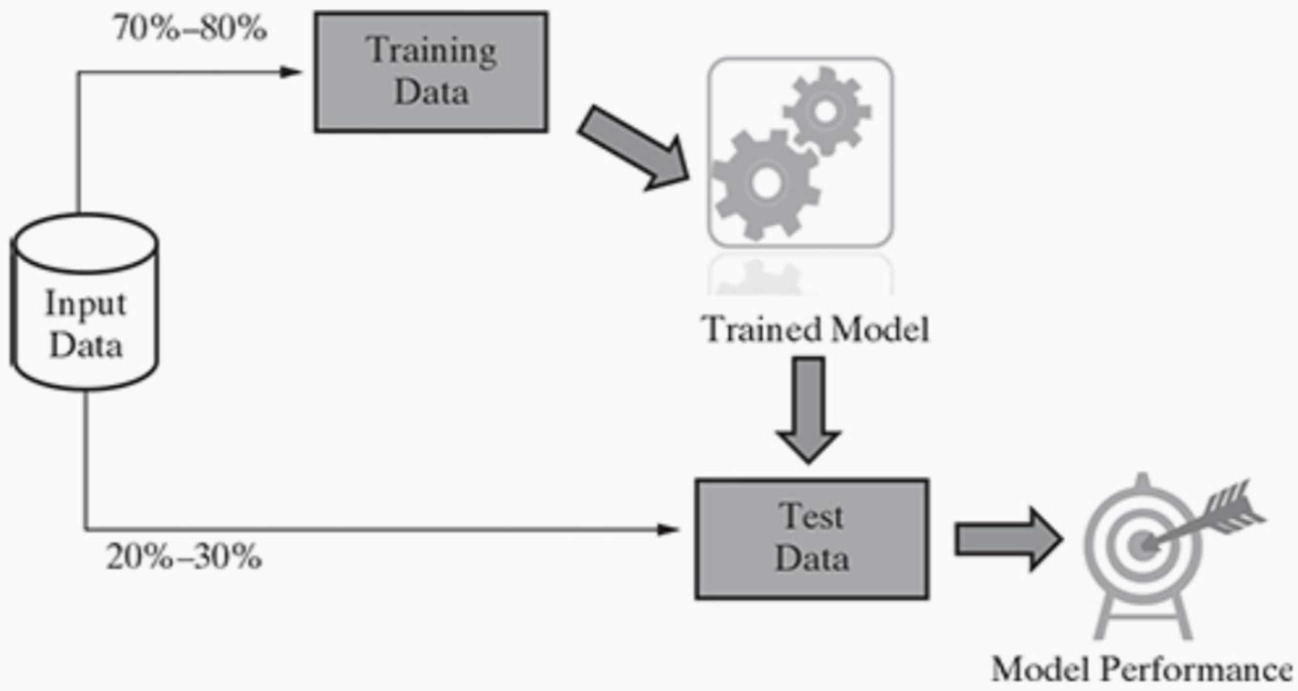


FIG. 3.1 Holdout method

TRAINING A MODEL (FOR SUPERVISED LEARNING)

Once the model is trained using the training data, the labels of the test data are predicted using the model’s target function.

Then the predicted value is compared with the actual value of the label.

This is possible because the test data is a part of the input data with known labels.

The performance of the model is in general measured by the accuracy of prediction of the label value

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- **In certain cases, the input data is partitioned into three portions – a training and a test data, and a third validation data.**

- **The validation data is used in place of test data, for measuring the model performance. It is used in iterations and to refine the model in each iteration.**

- **The test data is used only for once, after the model is refined and finalized, to measure and report the final performance of the model as a reference for future learning efforts.**

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- **An obvious problem in this method is that the division of data of different classes into the training and test data may not be proportionate.**

- **This situation is worse if the overall percentage of data related to certain classes is much less compared to other classes.**

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- **stratified random sampling : the whole data is broken into several homogenous groups or strata and a random sample is selected from each such stratum. This ensures that the generated random partitions have equal proportions of each class.**

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- **K-fold Cross-validation method : the smaller data sets may have the challenge to divide the data of some of the classes proportionally amongst training and test data sets.**

- **A special variant of holdout method, called repeated holdout, is sometimes employed to ensure the randomness of the composed data sets.**

- **In repeated holdout, several random holdouts are used to measure the model performance. In the end, the average of all performances is taken.**

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- **This process of repeated holdout is the basis of k-fold cross validation technique.**

- **In k-fold cross-validation, the data set is divided into k-completely distinct or non-overlapping random partitions called folds.**

TRAINING A MODEL (FOR SUPERVISED LEARNING)

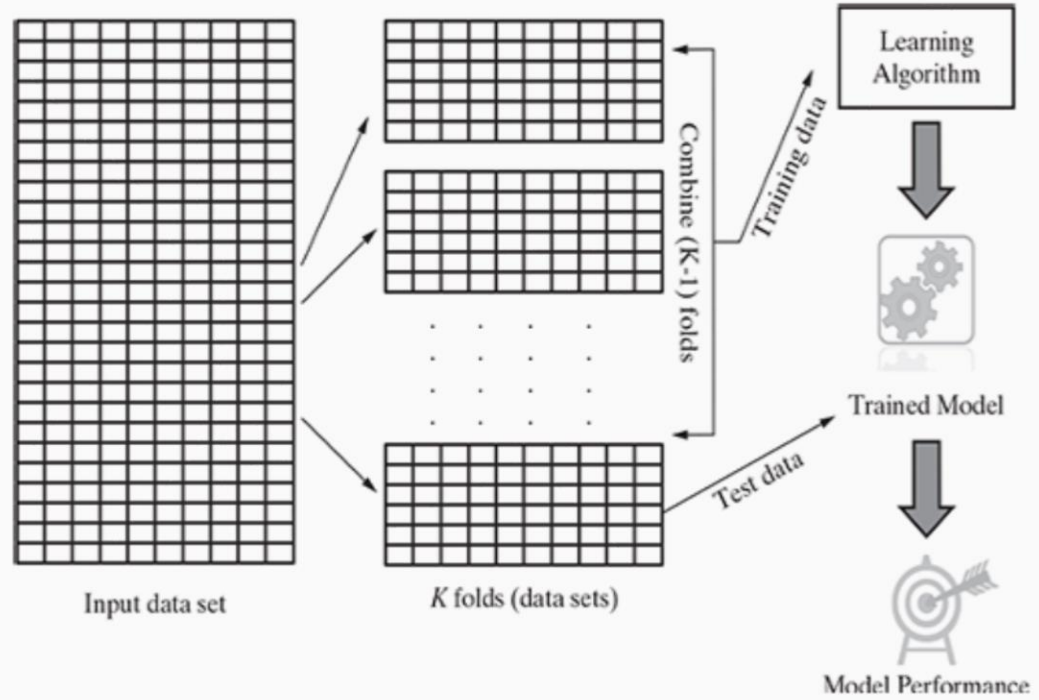


FIG. 3.2 Overall approach for *K*-fold cross-validation

TRAINING A MODEL (FOR SUPERVISED LEARNING)

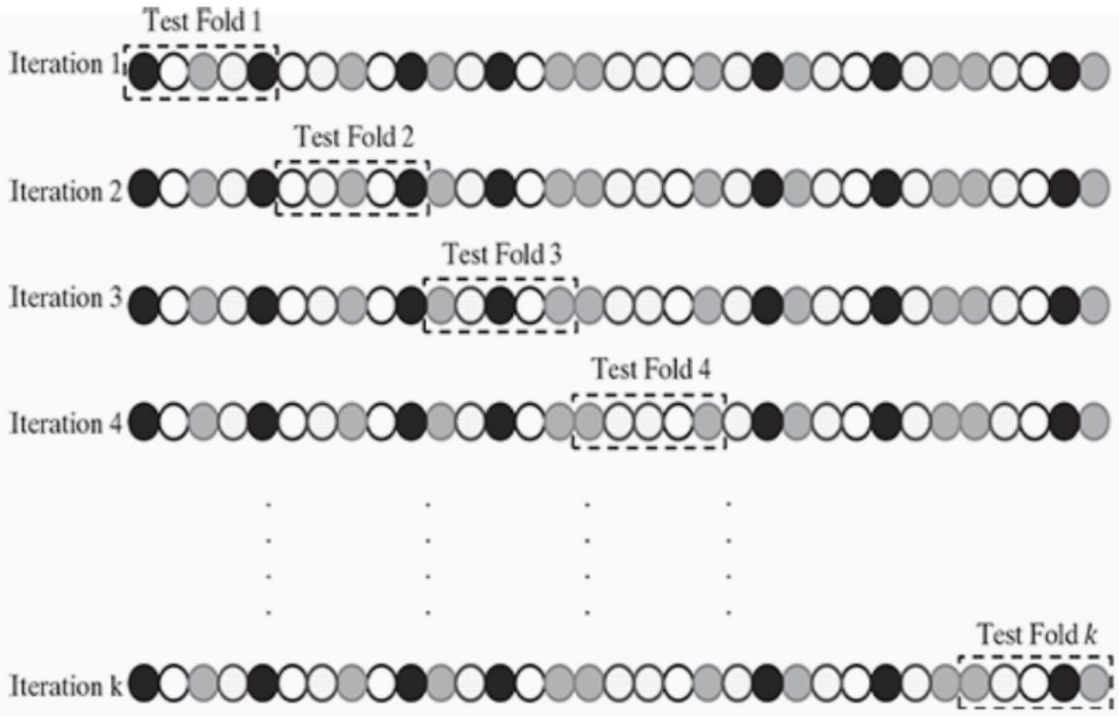


FIG. 3.3 Detailed approach for fold selection

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- **Leave-one-out cross-validation (LOOCV) is an extreme case of k-fold cross-validation using one record or data instance at a time as a test data.**

- **This is done to maximize the count of data used to train the model. It is obvious that the number of iterations for which it has to be run is equal to the total number of data in the input data set.**

- **Hence, obviously, it is computationally very expensive and not used much in practice.**

TRAINING A MODEL (FOR SUPERVISED LEARNING)

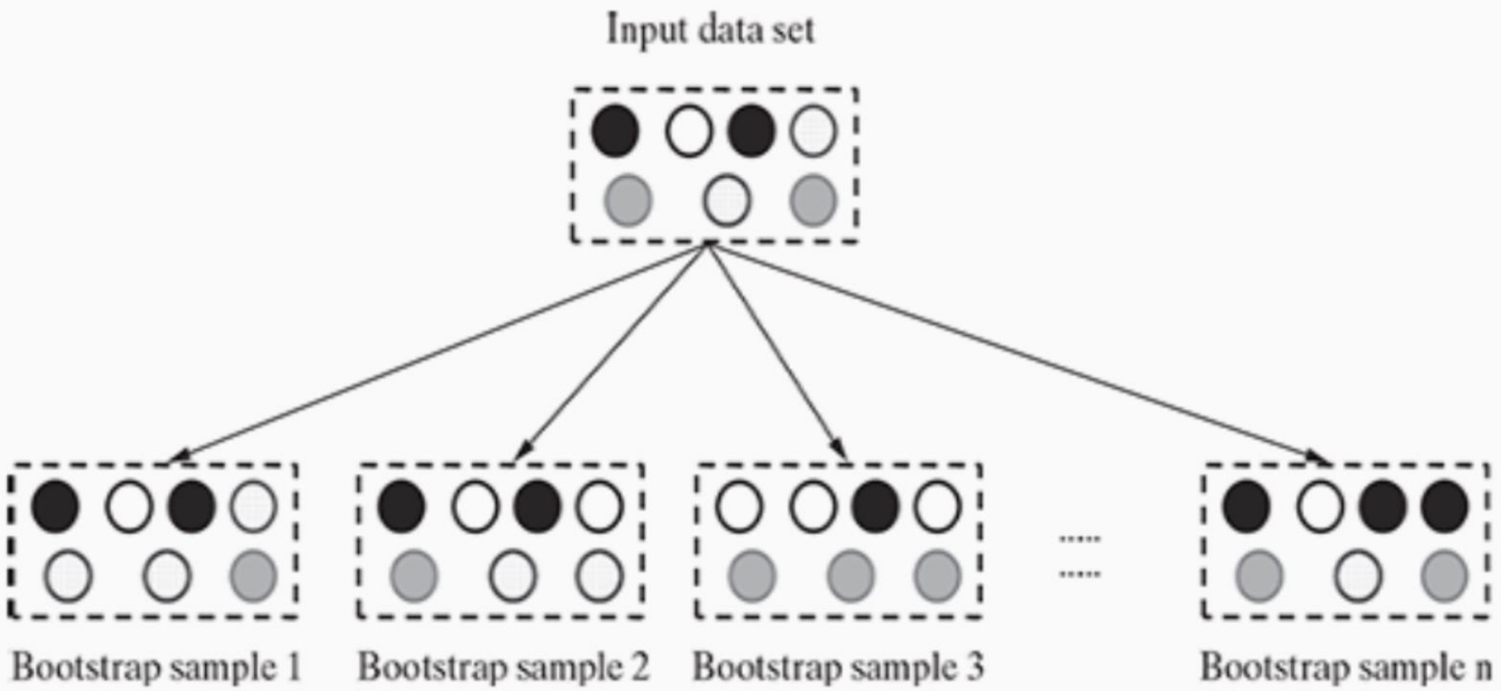
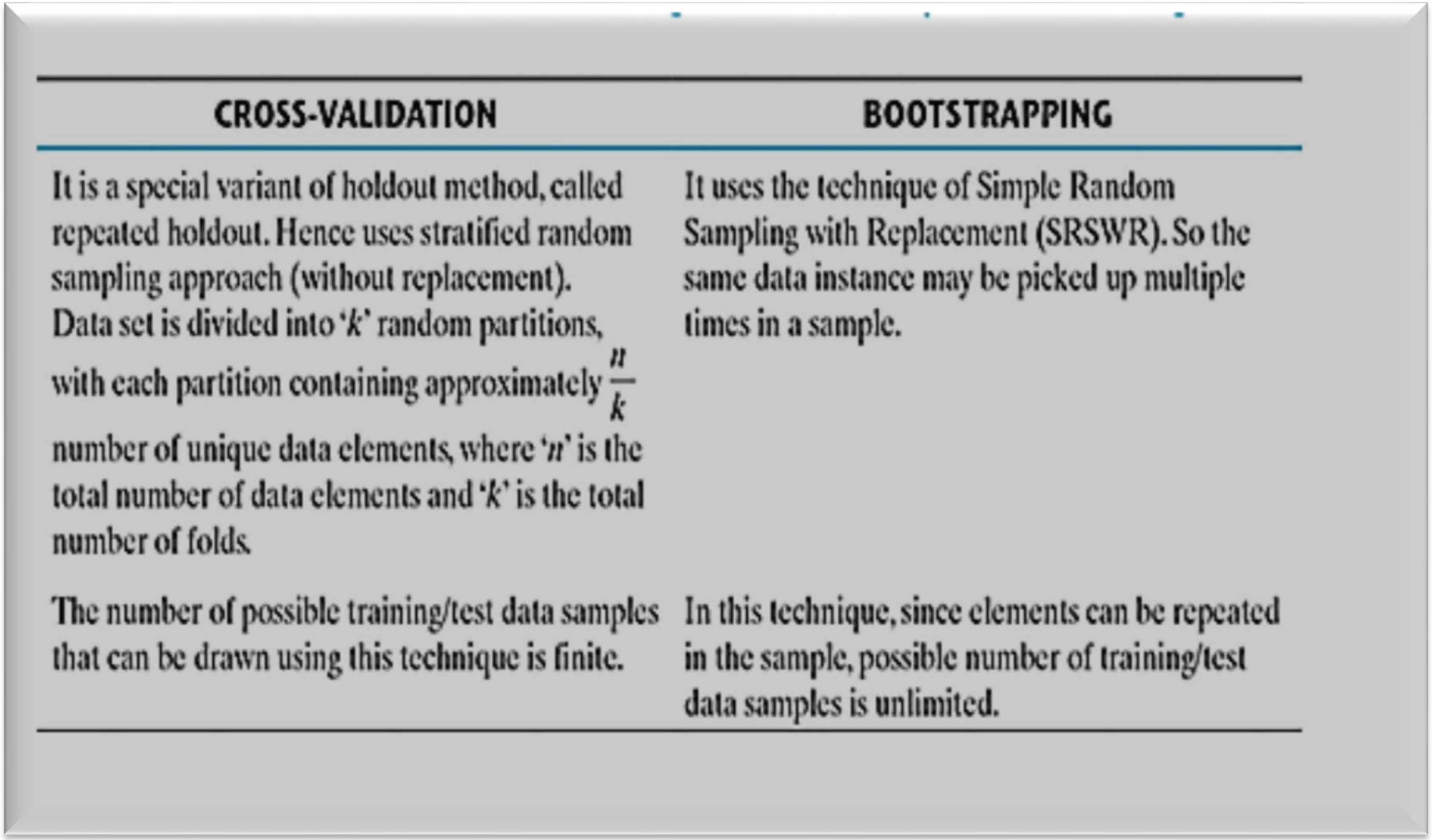


FIG. 3.4 Bootstrap sampling

TRAINING A MODEL (FOR SUPERVISED LEARNING)



CROSS-VALIDATION	BOOTSTRAPPING
It is a special variant of holdout method, called repeated holdout. Hence uses stratified random sampling approach (without replacement). Data set is divided into 'k' random partitions, with each partition containing approximately $\frac{n}{k}$ number of unique data elements, where 'n' is the total number of data elements and 'k' is the total number of folds.	It uses the technique of Simple Random Sampling with Replacement (SRSWR). So the same data instance may be picked up multiple times in a sample.
The number of possible training/test data samples that can be drawn using this technique is finite.	In this technique, since elements can be repeated in the sample, possible number of training/test data samples is unlimited.

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- Eager learning follows the general principles of machine learning – it tries to construct a generalized, input-independent target function during the model training phase.

- It follows the typical steps of machine learning, i.e. abstraction and generalization and comes up with a trained model at the end of the learning phase.

- Hence, when the test data comes in for classification, the eager learner is ready with the model and doesn’t need to refer back to the training data.

TRAINING A MODEL (FOR SUPERVISED LEARNING)

- Lazy learning, on the other hand, completely skips the abstraction and generalization processes, as explained in context of a typical machine learning process.
- In that respect, strictly speaking, lazy learner doesn't 'learn' anything. It uses the training data in exact, and uses the knowledge to classify the unlabelled test data.
- Since lazy learning uses training data as-is, it is also known as rote learning (i.e. memorization technique based on repetition). Due to its heavy dependency on the given training data instance, it is also known as instance learning.
- They are also called non-parametric learning.

Feature Selection

- **What is a feature?**

- **A feature is an attribute of a data set that is used in a machine learning process.**

- **only those attributes which are meaningful to a machine learning problem are to be called as features.**

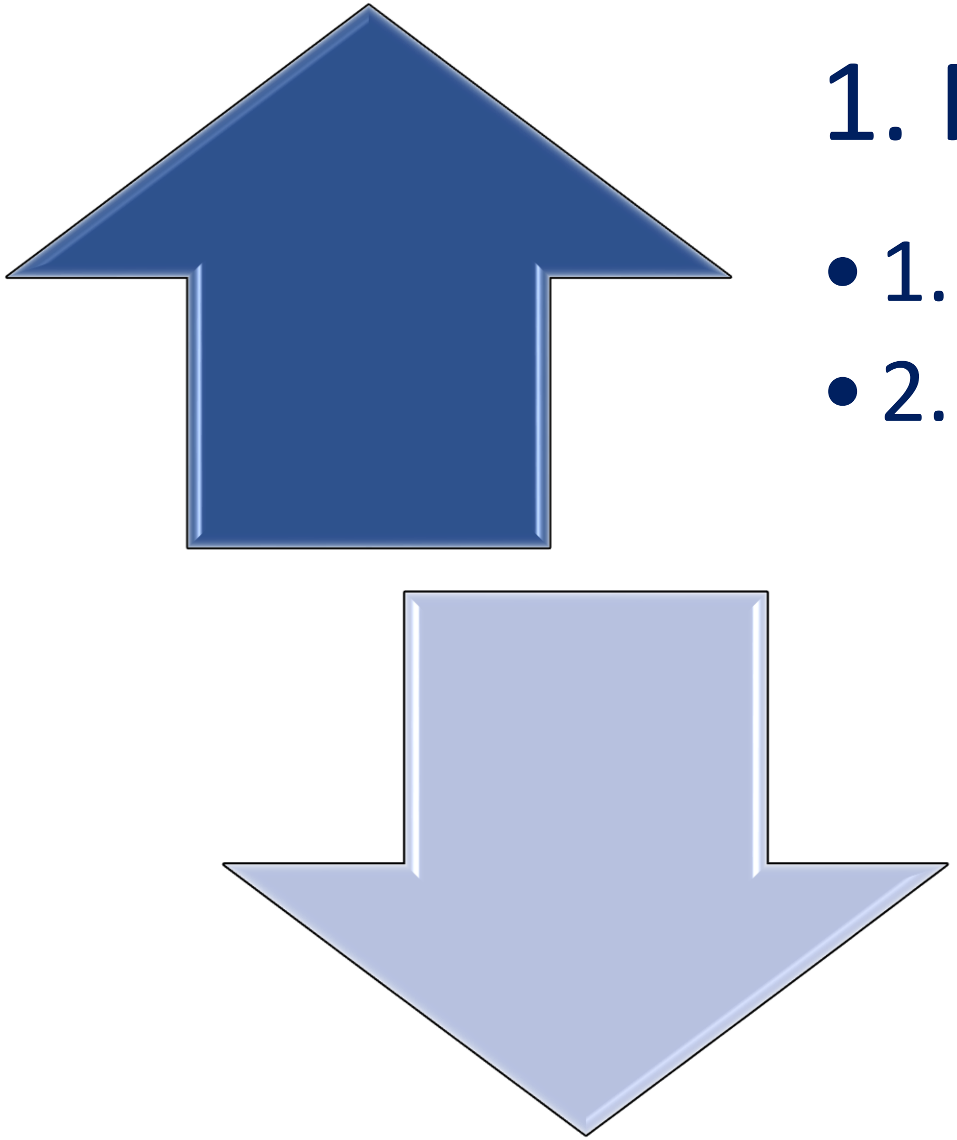
- **The features in a data set are also called its dimensions. So a data set having ‘n’ features is called an n-dimensional data set.**

Feature Engineering

- **What is feature engineering?**

- **Feature engineering refers to the process of translating a data set into features such that these features are able to represent the data set more effectively and result in a better learning performance.**

Feature Engineering

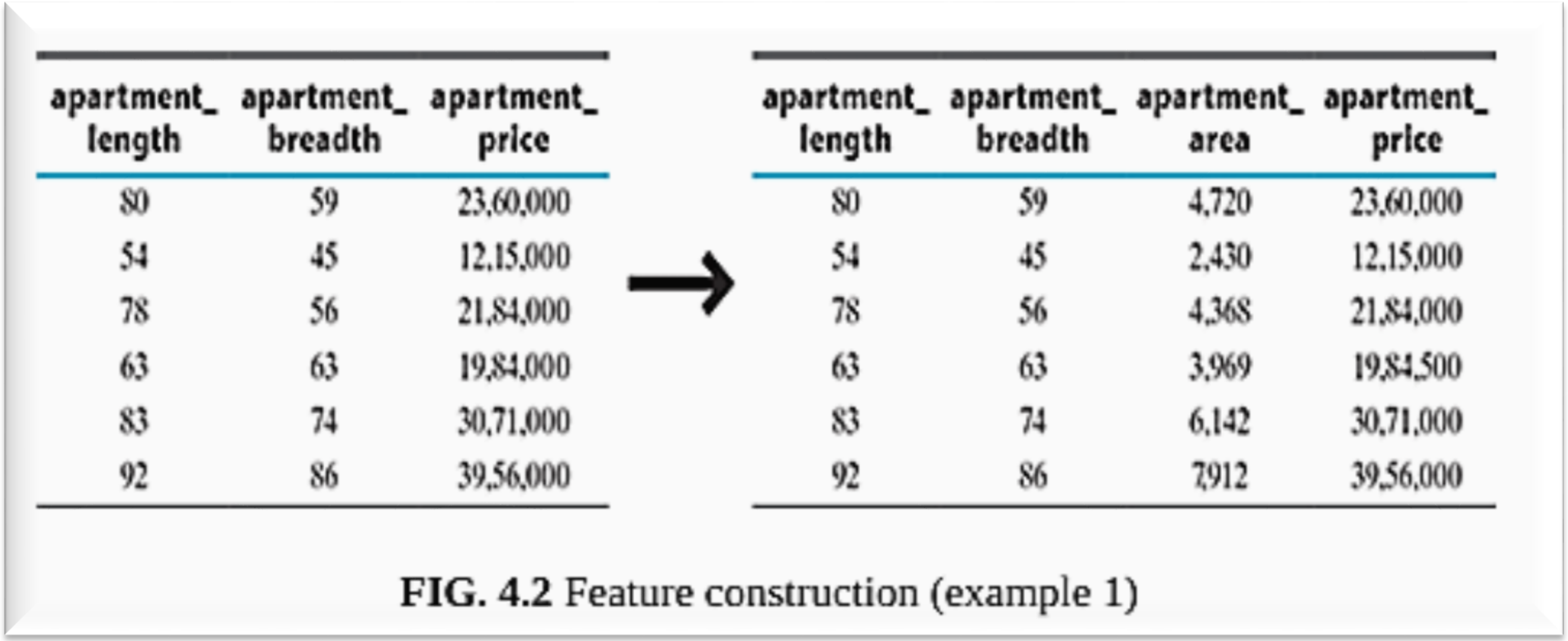


1. Feature Transformation

- 1. Feature Construction
- 2. Feature Extraction

2. Feature Subset Selection

Feature construction



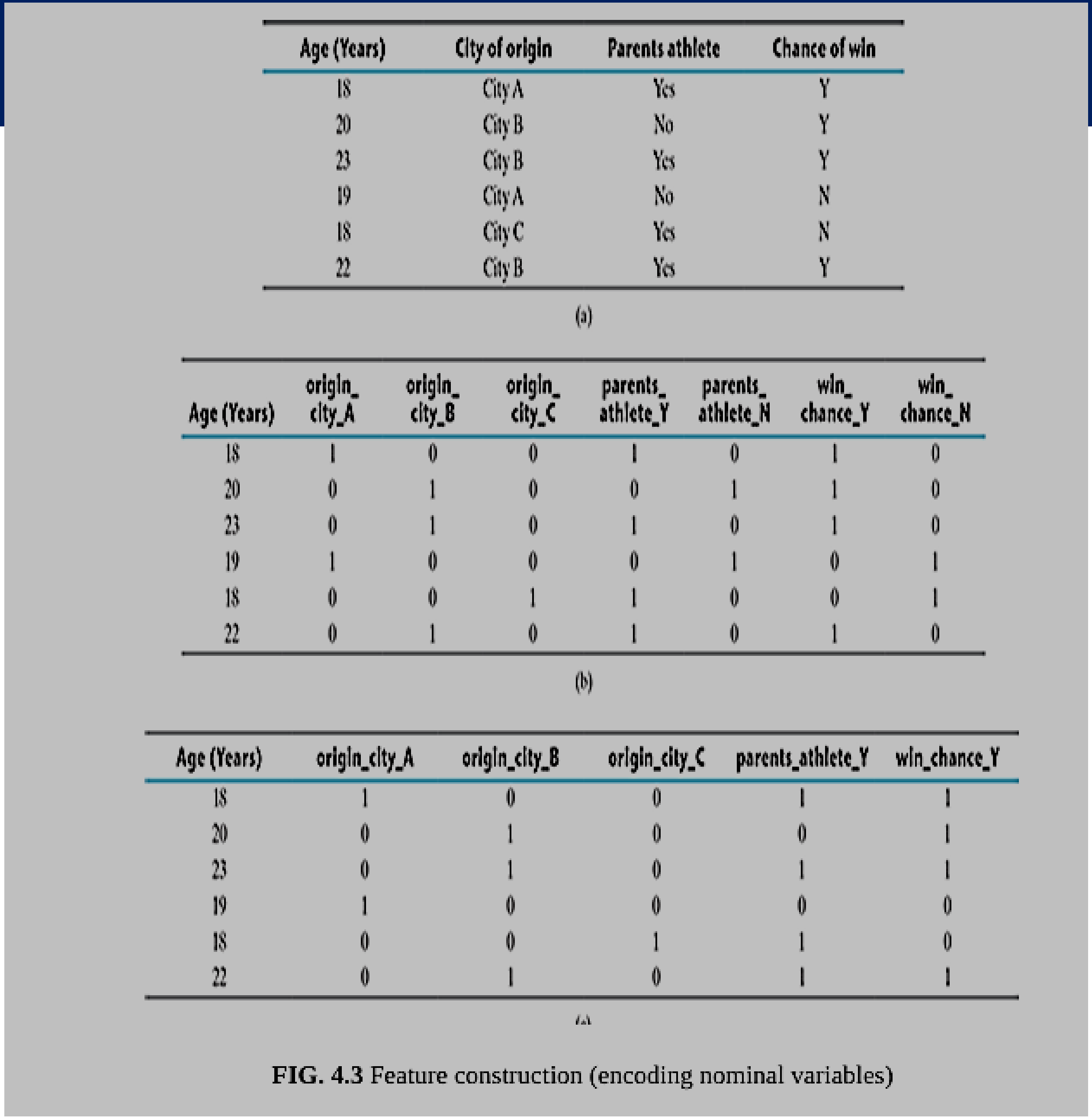
apartment_ length	apartment_ breadth	apartment_ price
80	59	23,60,000
54	45	12,15,000
78	56	21,84,000
63	63	19,84,000
83	74	30,71,000
92	86	39,56,000

→

apartment_ length	apartment_ breadth	apartment_ area	apartment_ price
80	59	4,720	23,60,000
54	45	2,430	12,15,000
78	56	4,368	21,84,000
63	63	3,969	19,84,500
83	74	6,142	30,71,000
92	86	7,912	39,56,000

FIG. 4.2 Feature construction (example 1)

Feature construction



Age (Years)	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N
22	City B	Yes	Y

(a)

Age (Years)	origin_ city_A	origin_ city_B	origin_ city_C	parents_ athlete_Y	parents_ athlete_N	win_ chance_Y	win_ chance_N
18	1	0	0	1	0	1	0
20	0	1	0	0	1	1	0
23	0	1	0	1	0	1	0
19	1	0	0	0	1	0	1
18	0	0	1	1	0	0	1
22	0	1	0	1	0	1	0

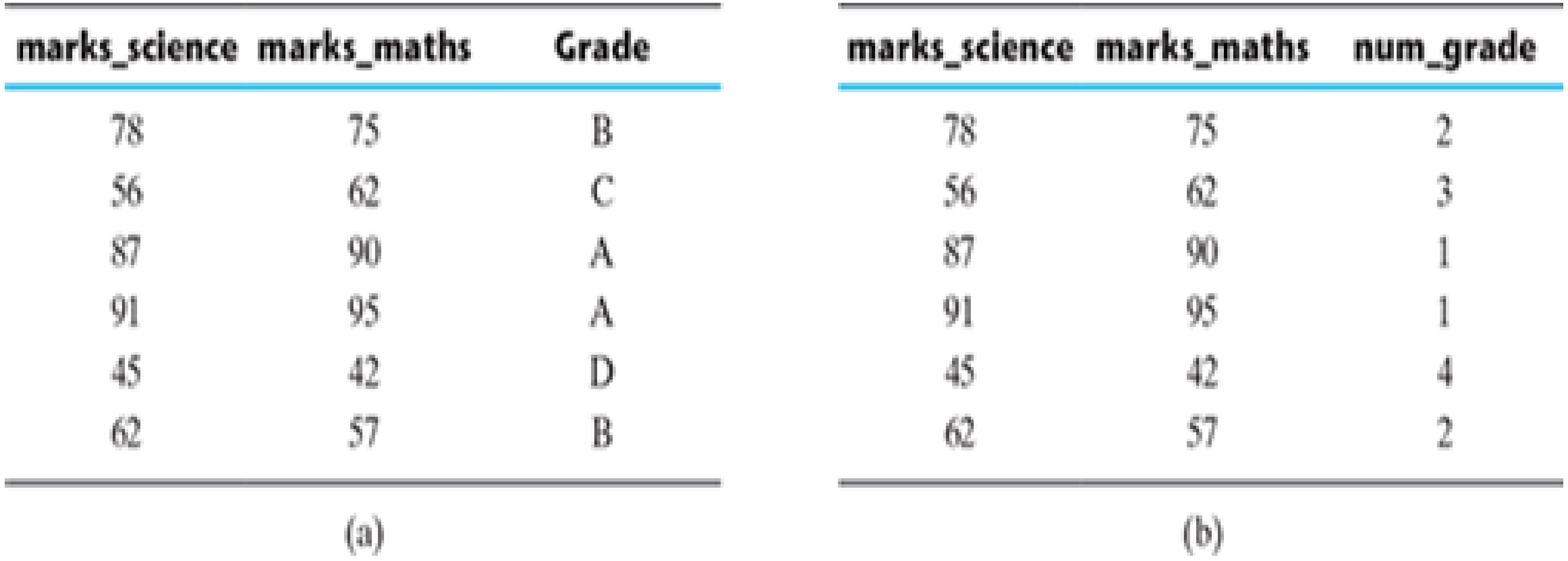
(b)

Age (Years)	origin_city_A	origin_city_B	origin_city_C	parents_athlete_Y	win_chance_Y
18	1	0	0	1	1
20	0	1	0	0	1
23	0	1	0	1	1
19	1	0	0	0	0
18	0	0	1	1	0
22	0	1	0	1	1

(c)

FIG. 4.3 Feature construction (encoding nominal variables)

Feature construction Encoding categorical (ordinal) variables



marks_science	marks_maths	Grade
78	75	B
56	62	C
87	90	A
91	95	A
45	42	D
62	57	B

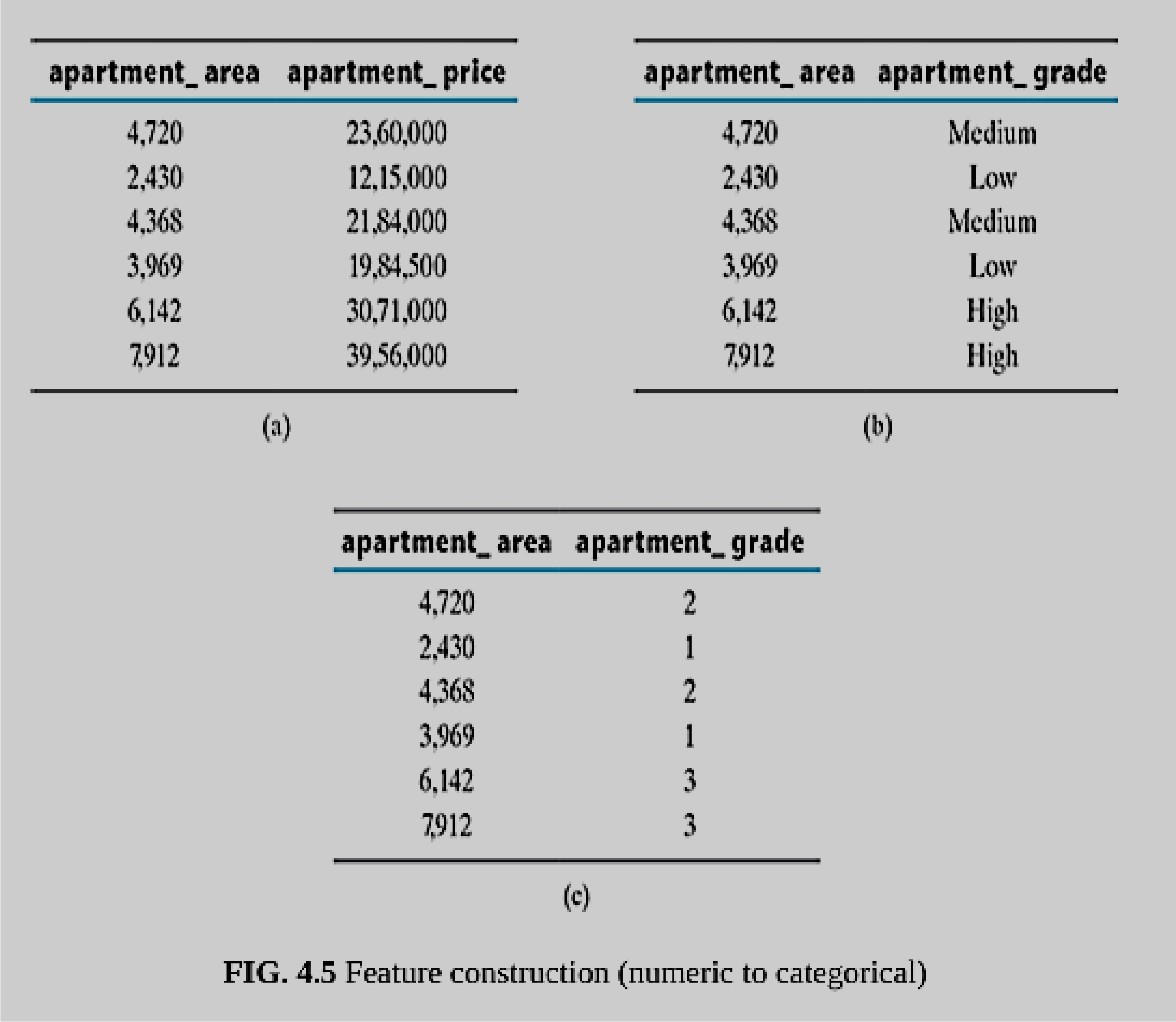
(a)

marks_science	marks_maths	num_grade
78	75	2
56	62	3
87	90	1
91	95	1
45	42	4
62	57	2

(b)

FIG. 4.4 Feature construction (encoding ordinal variables)

Feature construction Transforming numeric (continuous) features to categorical features



apartment_ area	apartment_ price
4,720	23,60,000
2,430	12,15,000
4,368	21,84,000
3,969	19,84,500
6,142	30,71,000
7,912	39,56,000

(a)

apartment_ area	apartment_ grade
4,720	Medium
2,430	Low
4,368	Medium
3,969	Low
6,142	High
7,912	High

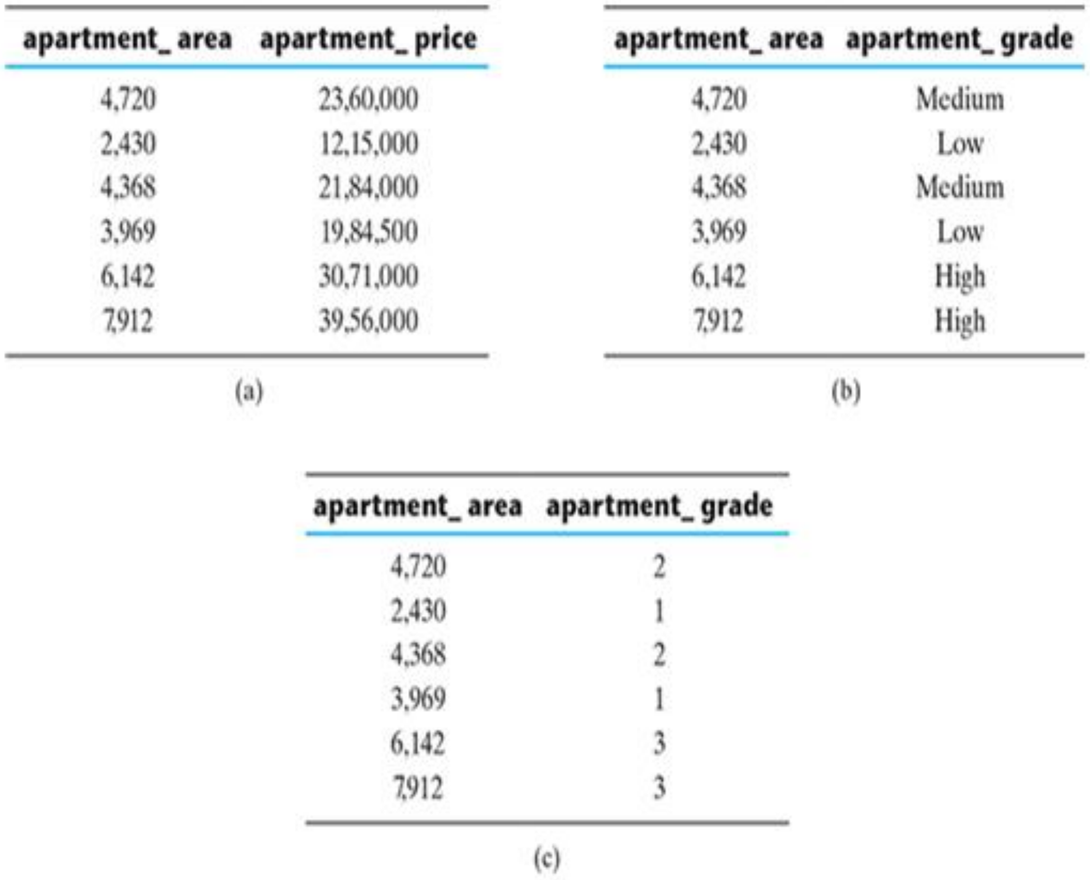
(b)

apartment_ area	apartment_ grade
4,720	2
2,430	1
4,368	2
3,969	1
6,142	3
7,912	3

(c)

FIG. 4.5 Feature construction (numeric to categorical)

Feature construction Text-specific feature construction



apartment_ area	apartment_ price
4,720	23,60,000
2,430	12,15,000
4,368	21,84,000
3,969	19,84,500
6,142	30,71,000
7,912	39,56,000

(a)

apartment_ area	apartment_ grade
4,720	Medium
2,430	Low
4,368	Medium
3,969	Low
6,142	High
7,912	High

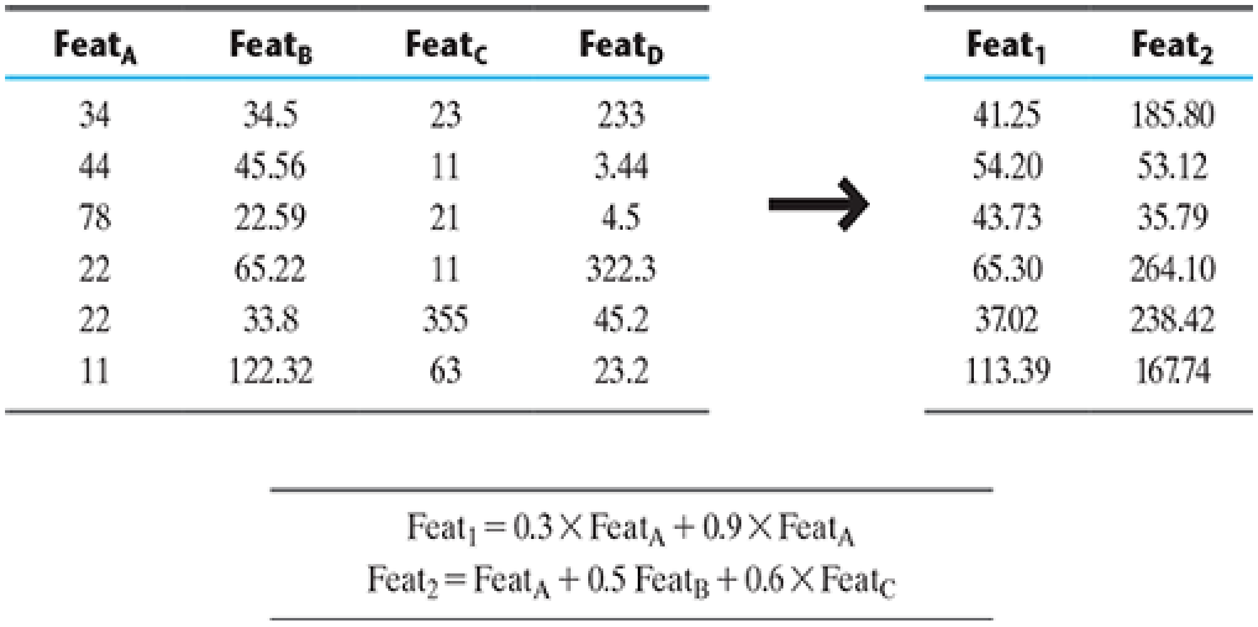
(b)

apartment_ area	apartment_ grade
4,720	2
2,430	1
4,368	2
3,969	1
6,142	3
7,912	3

(c)

FIG. 4.5 Feature construction (numeric to categorical)

Feature extraction



$Feat_A$	$Feat_B$	$Feat_C$	$Feat_D$
34	34.5	23	233
44	45.56	11	3.44
78	22.59	21	4.5
22	65.22	11	322.3
22	33.8	355	45.2
11	122.32	63	23.2

→

$Feat_1$	$Feat_2$
41.25	185.80
54.20	53.12
43.73	35.79
65.30	264.10
37.02	238.42
113.39	167.74

$$Feat_1 = 0.3 \times Feat_A + 0.9 \times Feat_A$$
$$Feat_2 = Feat_A + 0.5\ Feat_B + 0.6 \times Feat_C$$

FIG. 4.7 Feature extraction

Principal Component Analysis

A key to the success of machine learning lies in the fact that the features are less in number as well as the similarity between each other is very less.

This is the main guiding philosophy of principal component analysis (PCA) technique of feature extraction.

Principal Component Analysis

A key to the success of machine learning lies in the fact that the features are less in number as well as the similarity between each other is very less.

This is the main guiding philosophy of principal component analysis (PCA) technique of feature extraction.

Principal Component Analysis

In PCA, a new set of features are extracted from the original features which are quite dissimilar in nature.

So an n dimensional feature space gets transformed to an m dimensional feature space, where the dimensions are orthogonal to each other, i.e. completely independent of each other.

To understand the concept of orthogonality, we have to step back and do a bit of dip dive into vector space concept in linear algebra.

Principal Component Analysis

$$v = \sum_{i=1}^{n} a_i u_i$$

where, a represents ‘n’ scalars and u represents the basis vectors. Basis vectors are orthogonal to each other.

i Orthogonality of vectors in n-dimensional vector space can be thought of an extension of the vectors being perpendicular in a two-dimensional vector space.

Principal Component Analysis

The feature vector can be transformed to a vector space of the basis vectors which are termed as principal components.

These principal components, just like the basis vectors, are orthogonal to each other.

Principal Component Analysis

So a set of feature vectors which may have similarity with each other is transformed to a set of principal components which are completely unrelated.

However, the principal components capture the variability of the original feature space. Also, the number of principal component derived, much like the basis vectors, is much smaller than the original set of features.

Principal Component Analysis

The objective of PCA is to make the transformation in such a way that
1. The new features are distinct, i.e. the covariance between the new features, i.e. the principal components is 0.

2. The principal components are generated in order of the variability in the data that it captures. Hence, the first principal component should capture the maximum variability, the second principal component should capture the next highest variability etc.

3. The sum of variance of the new features or the principal components should be equal to the sum of variance of the original features.

Principal Component Analysis

1. **First, calculate the covariance matrix of a data set.**
2. **Then, calculate the eigenvalues of the covariance matrix.**
3. **The eigenvector having highest eigenvalue represents the direction in which there is the highest variance. So this will help in identifying the first principal component.**
4. **The eigenvector having the next highest eigenvalue represents the direction in which data has the highest remaining variance and also orthogonal to the first direction. So this helps in identifying the second principal component.**
5. **Like this, identify the top ‘k’ eigenvectors having top ‘k’ eigenvalues so as to get the ‘k’ principal components.**

Singular value decomposition

SVD of a matrix A (m × n) is a factorization of the form :

$$A = U \sum V$$

where, U and V are orthonormal matrices, U is an m × m unitary matrix, V is an n × n unitary matrix and ∑ is an m × n rectangular diagonal matrix.

The diagonal entries of ∑ are known as singular values of matrix A.

The columns of U and V are called the left-singular and right-singular vectors of matrix A, respectively.

Singular value decomposition

SVD of a matrix A (m × n) is a factorization of the form :

$$A = U\sum V$$

where, U and V are orthonormal matrices, U is an m × m unitary matrix, V is an n × n unitary matrix and ∑ is an m × n rectangular diagonal matrix.

The diagonal entries of ∑ are known as singular values of matrix A.

The columns of U and V are called the left-singular and right-singular vectors of matrix A, respectively.

Singular value decomposition

SVD of a data matrix is expected to have the properties highlighted below:

1. **Patterns in the attributes are captured by the right-singular vectors, i.e. the columns of V.**
2. **Patterns among the instances are captured by the left-singular, i.e. the columns of U.**
3. **Larger a singular value, larger is the part of the matrix A that it accounts for and its associated vectors.**
4. **New data matrix with ‘k’ attributes is obtained using the equation ’ $D = D \times [v , v , \dots , v]$ Thus, the dimensionality gets reduced to k**

Linear Discriminant Analysis

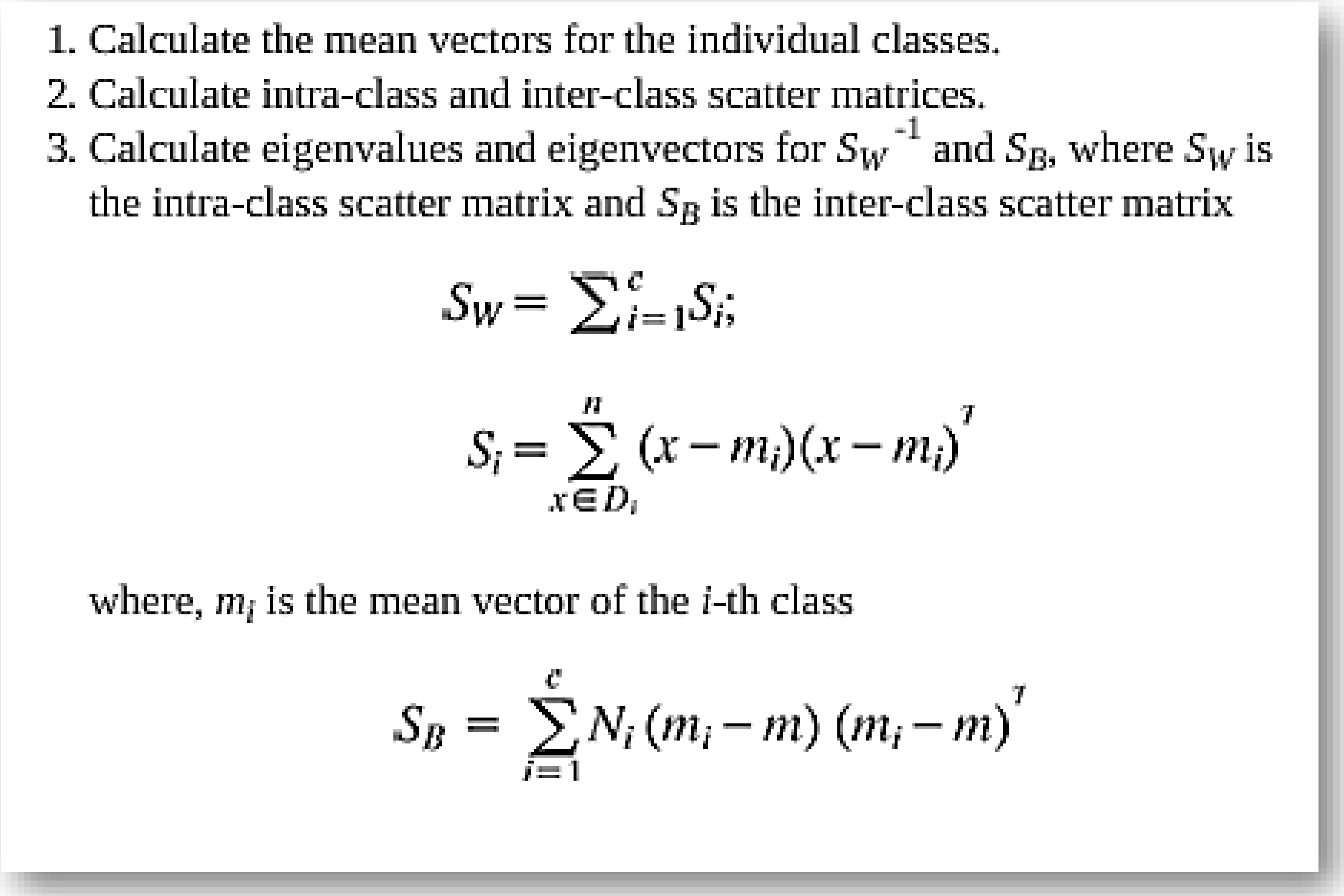
Linear discriminant analysis (LDA) is another commonly used feature extraction technique like PCA or SVD.

The objective of LDA is similar to the sense that it intends to transform a data set into a lower dimensional feature space.

However, unlike PCA, the focus of LDA is not to capture the data set variability.

Instead, LDA focuses on class separability

Linear Discriminant Analysis



1. Calculate the mean vectors for the individual classes.
2. Calculate intra-class and inter-class scatter matrices.
3. Calculate eigenvalues and eigenvectors for S_W^{-1} and S_B, where S_W is the intra-class scatter matrix and S_B is the inter-class scatter matrix

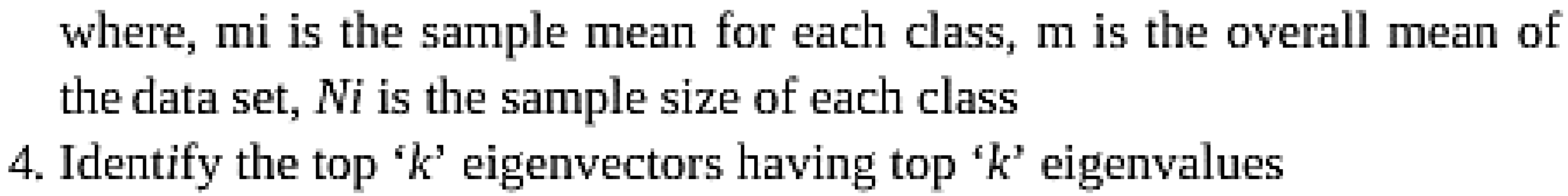
$$S_W = \sum_{i=1}^{c} S_i;$$

$$S_i = \sum_{x \in D_i}^{n} (x - m_i)(x - m_i)^T$$

where, m_i is the mean vector of the i-th class

$$S_B = \sum_{i=1}^{c} N_i \, (m_i - m) \, (m_i - m)^T$$

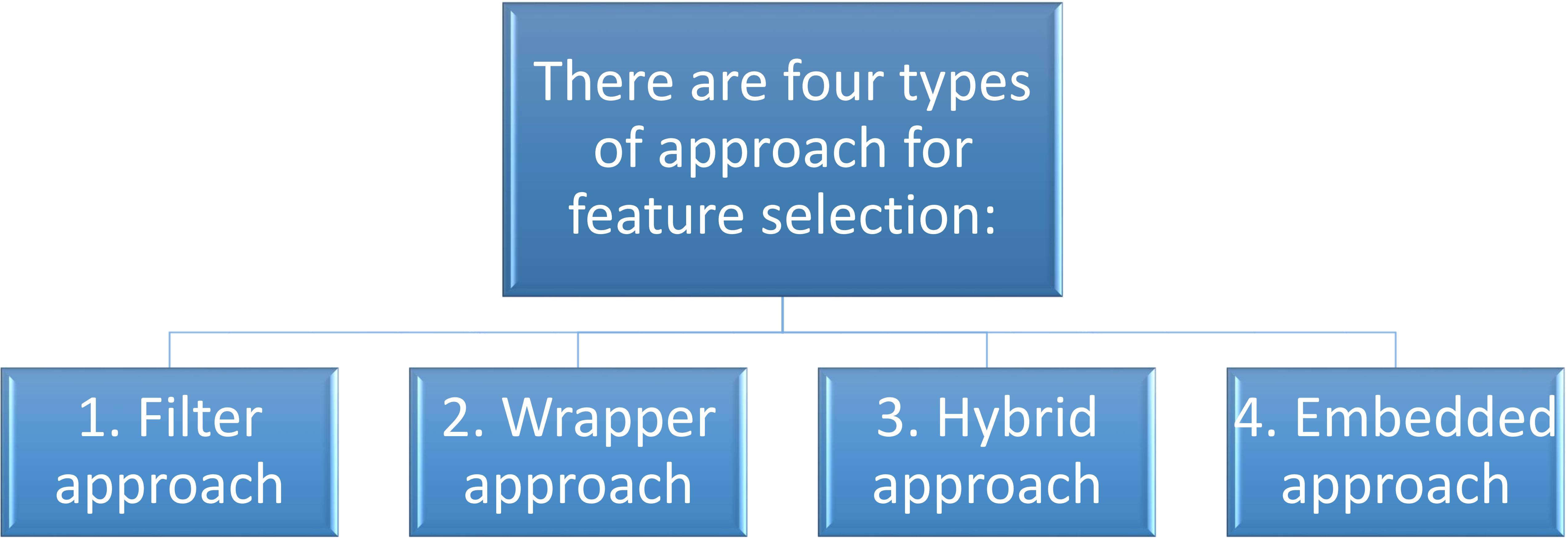
Linear Discriminant Analysis



where, mi is the sample mean for each class, m is the overall mean of the data set, *Ni* is the sample size of each class

4. Identify the top ‘*k*’ eigenvectors having top ‘*k*’ eigenvalues

Feature selection approaches



There are four types of approach for feature selection:

1. Filter approach

2. Wrapper approach

3. Hybrid approach

4. Embedded approach

Feature selection approaches

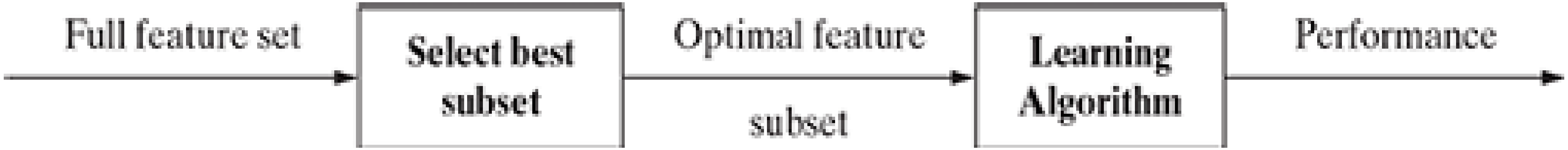
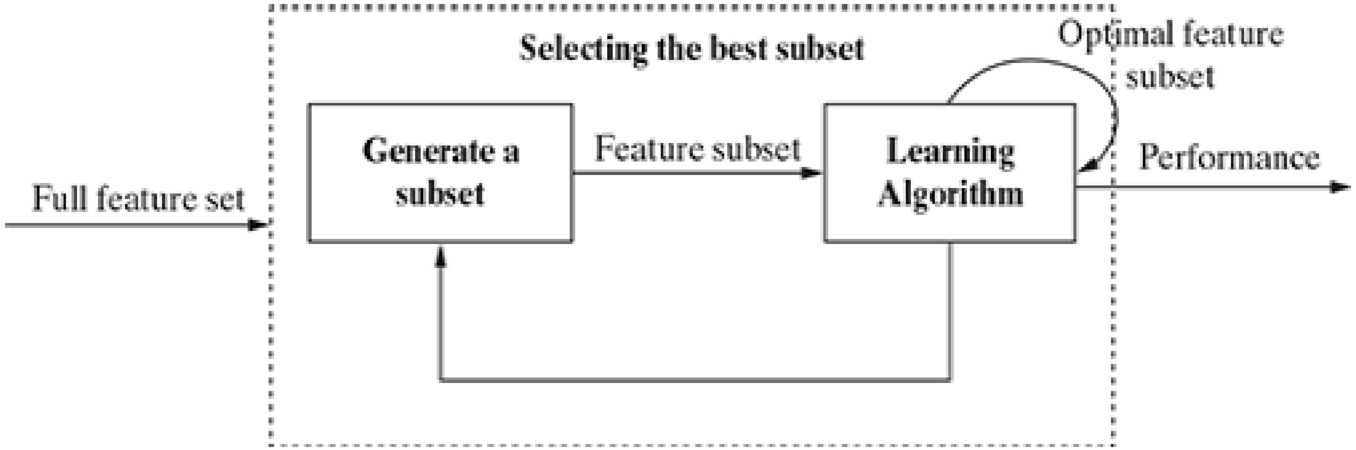


FIG. 4.13 Filter approach

Feature selection approaches



Feature selection approaches

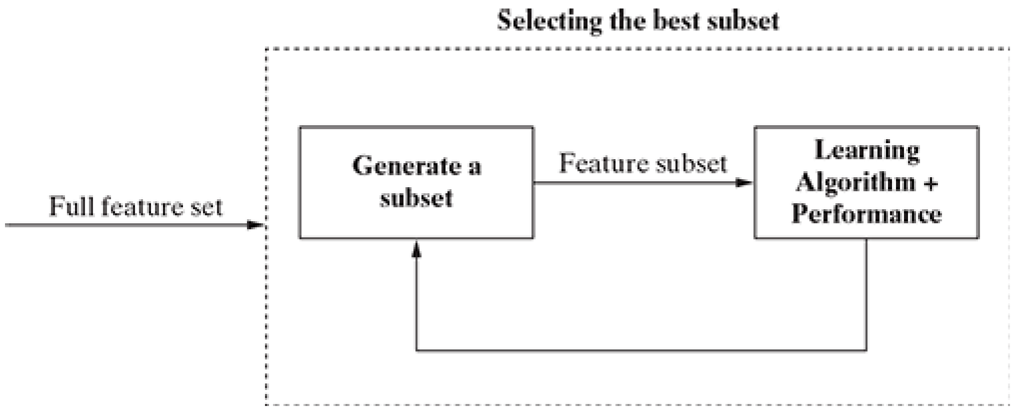


FIG. 4.15 Embedded approach

Feature selection approaches

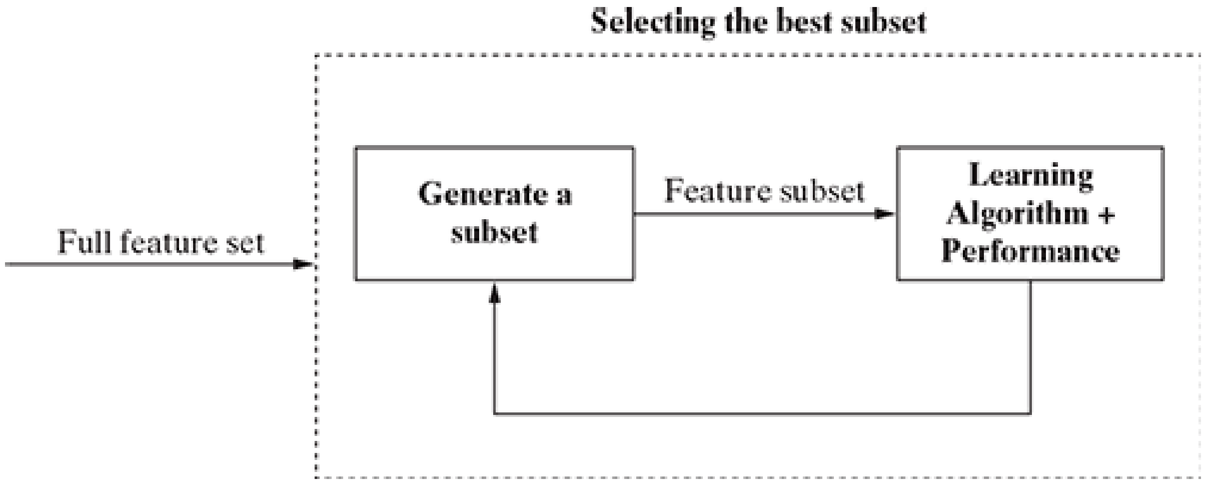


FIG. 4.15 Embedded approach