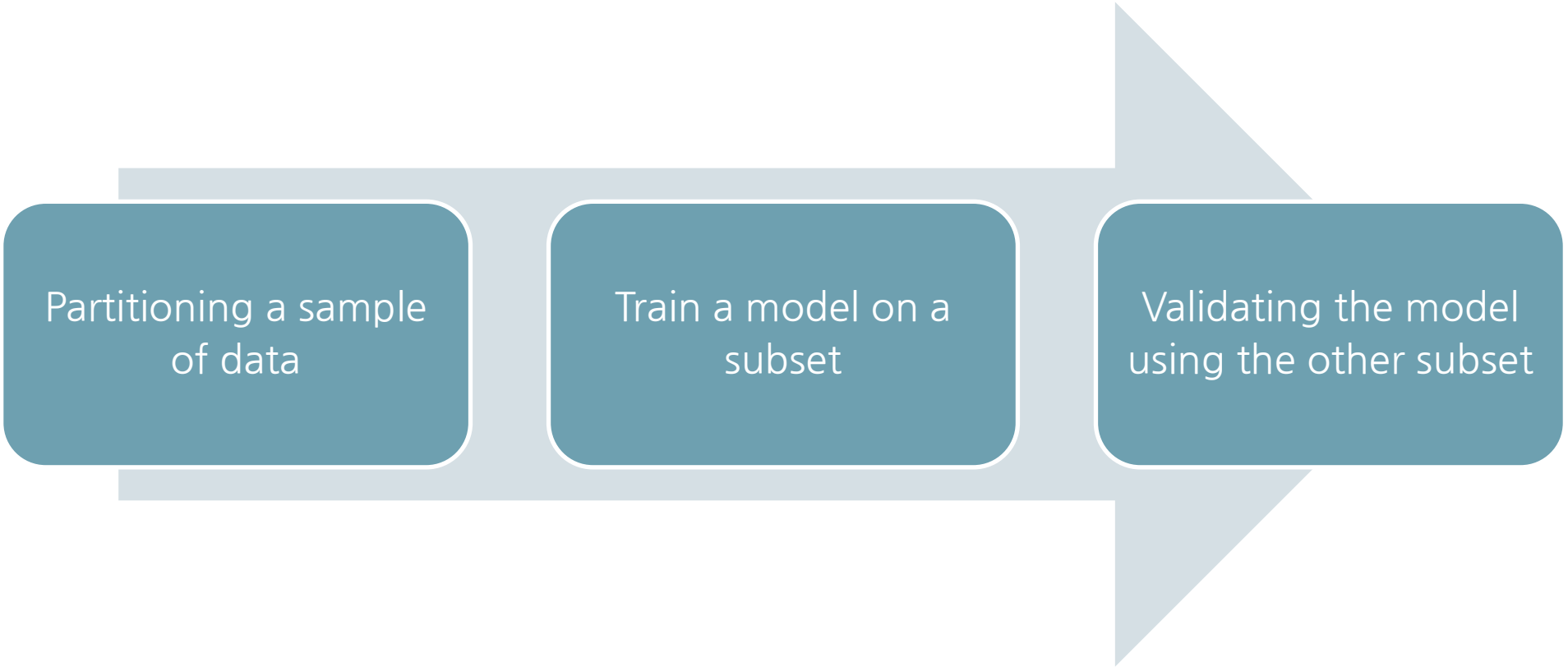
Week04

Model Validation

Model Validation

- A model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set
  - Estimate how accurately a predictive model will perform in practice
  - It is also used to determine the best set of parameters



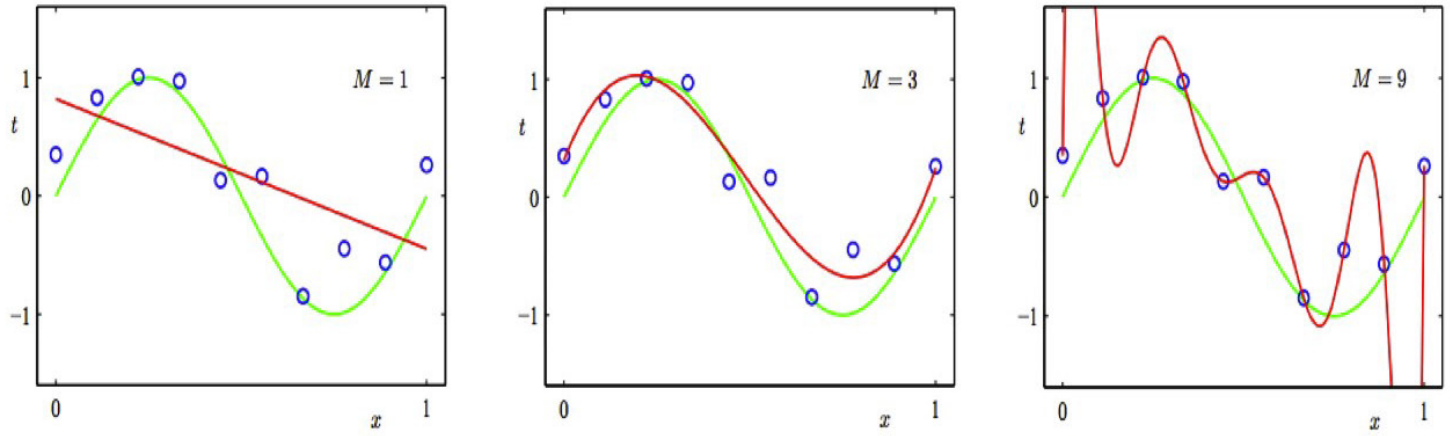
Underfitting and Overfitting

- Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data
  - Not fit the data well enough
  - Underfitting is often a result of an excessively simple model

- Overfitting occurs when a model is excessively complex
  - Have too many parameters relative to the number of observations
  - Show poor predictive performance

Underfitting and Overfitting

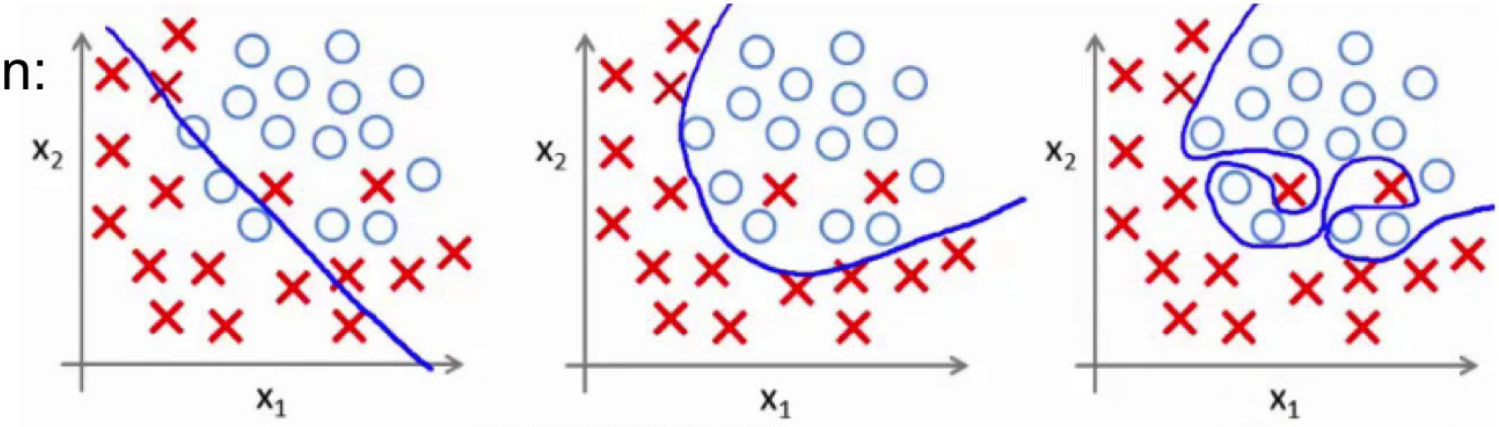
Regression:



predictor too inflexible:
cannot capture pattern

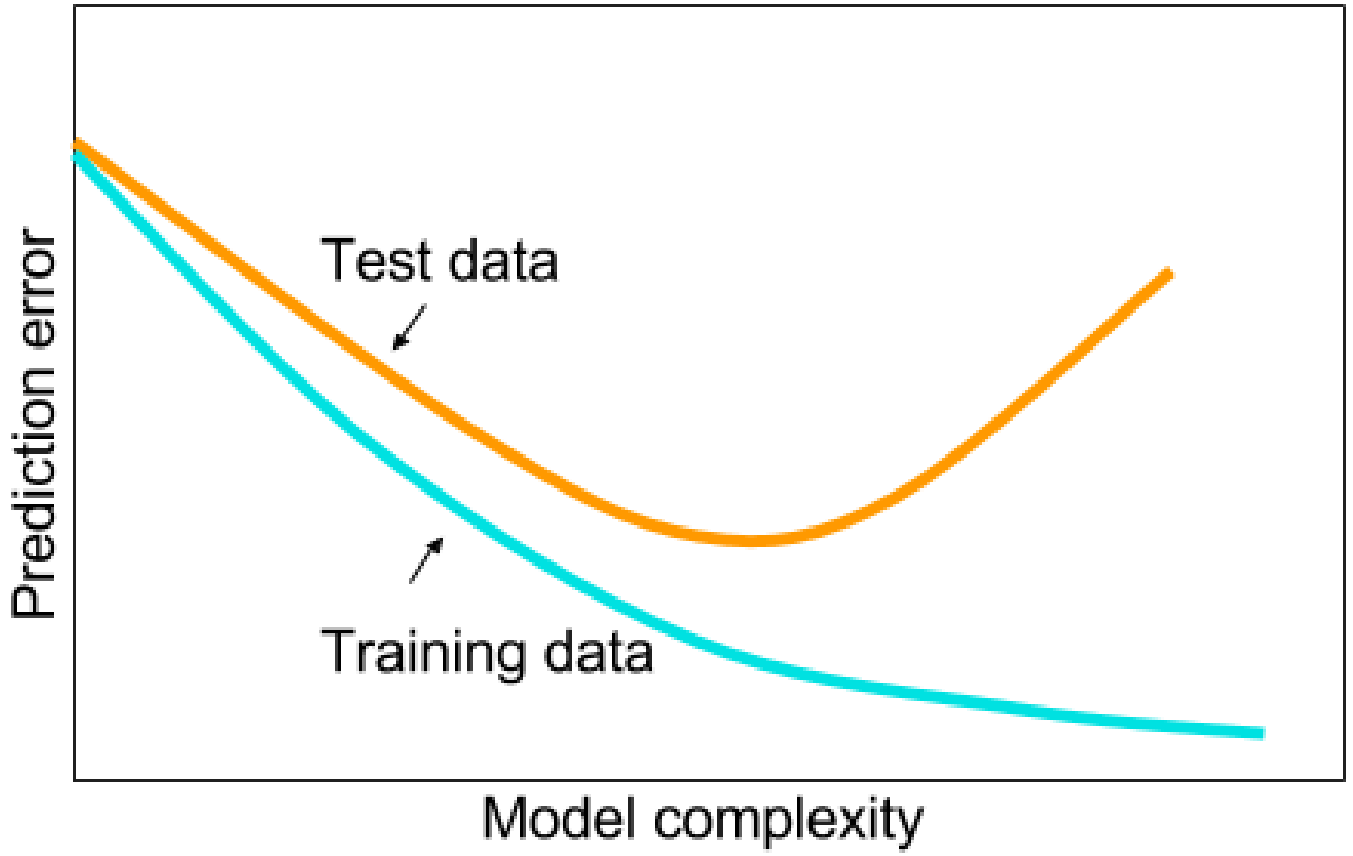
predictor too flexible:
fits noise in the data

Classification:



Underfitting and Overfitting

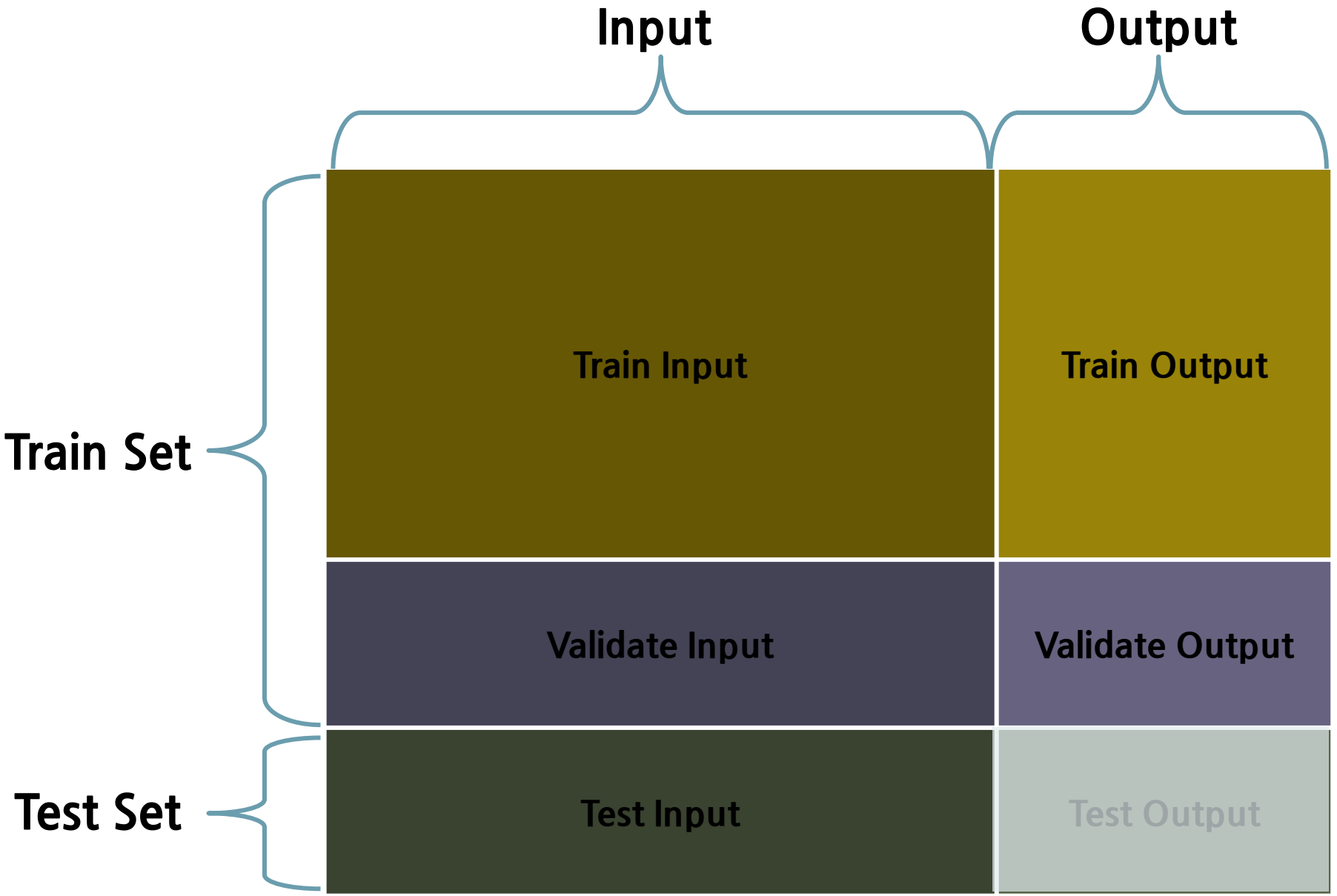
- Model complexity vs. Prediction error



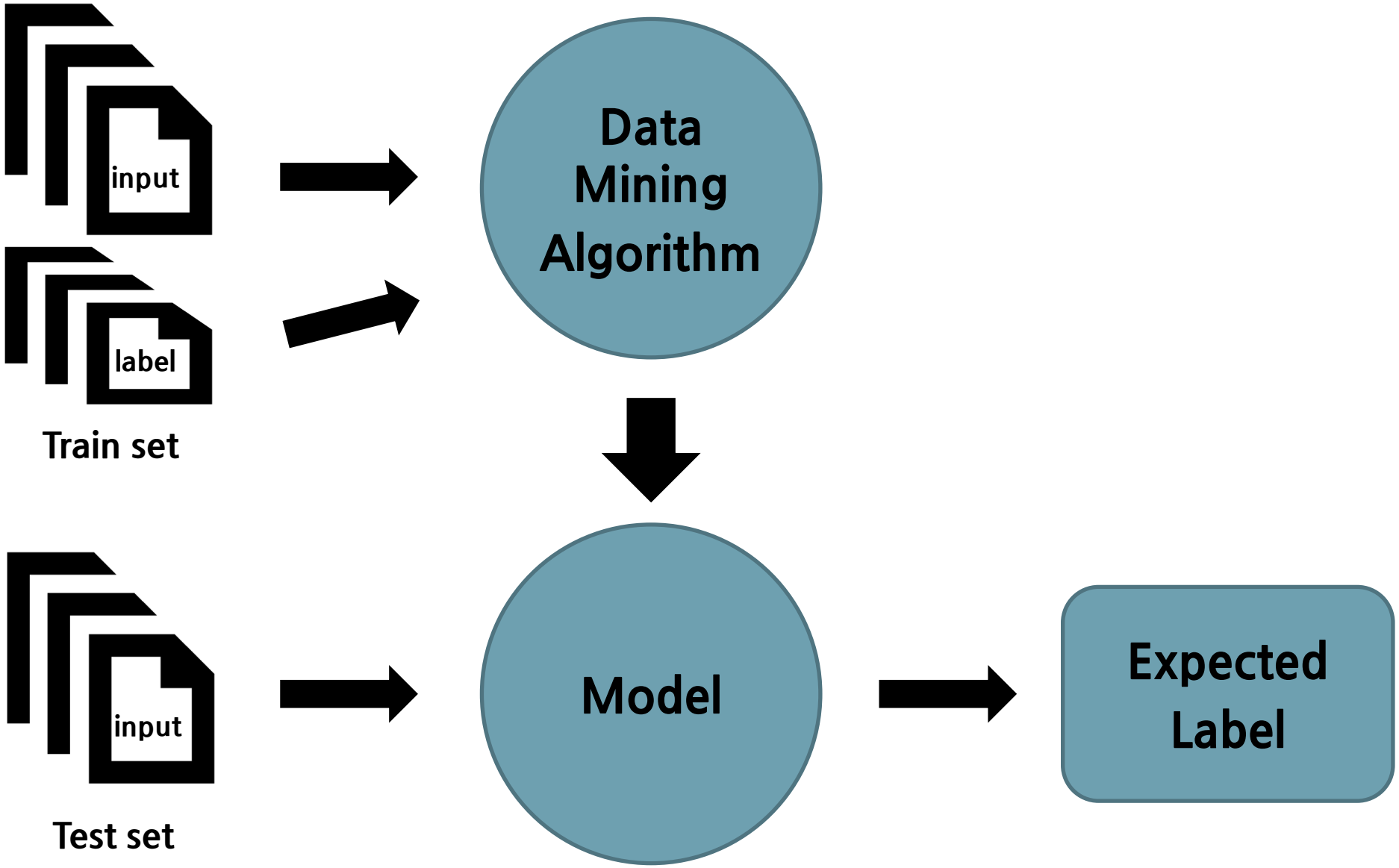
Model Validation

- Model validation
  - Goal
    - Quickly and consistently test algorithms against a fair representation of the problem
- Data partition
  - The data for evaluating a model should be different from training data
- Performance measure
  - The way to evaluate a model to the problem
  - It is important to set an appropriate performance measure to each problem

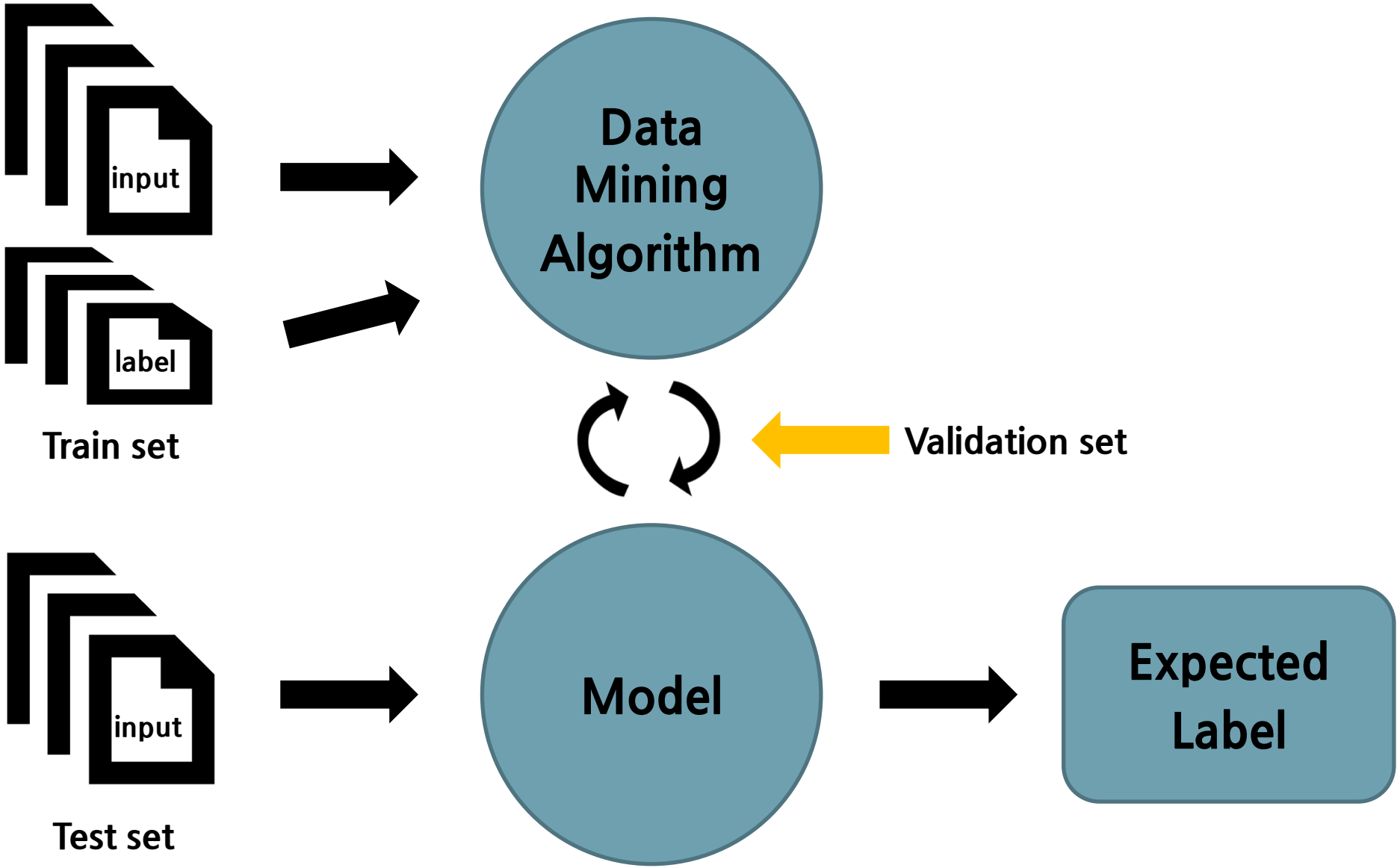
Data Partition



Process of Supervised Learning

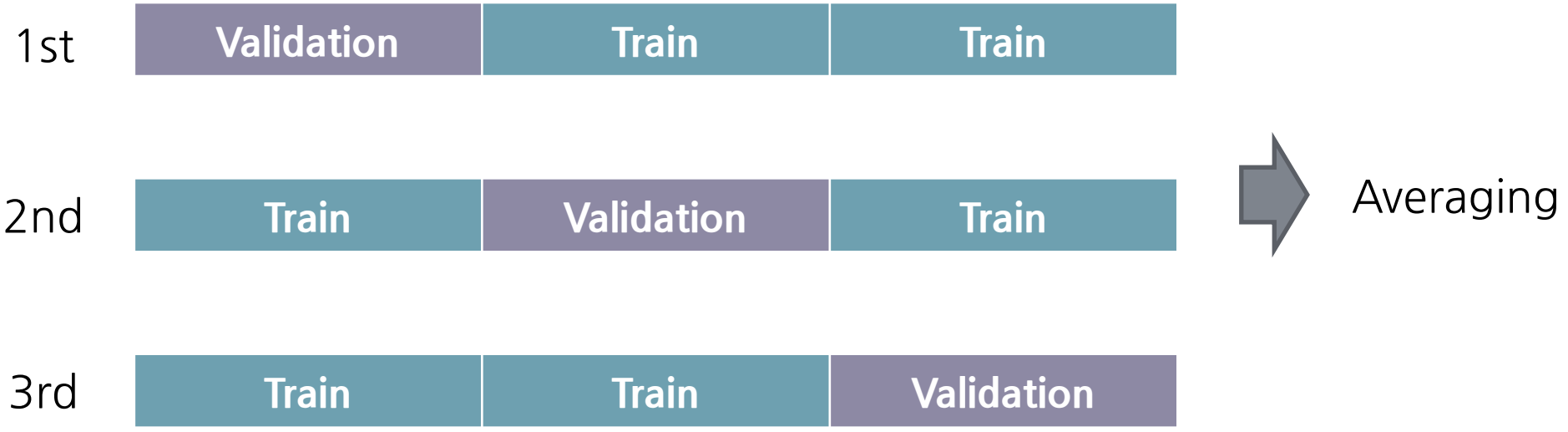


Process of Supervised Learning



k-fold Cross-validation

- In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples
  - Of the k subsamples, a single subsample is retained as the validation data
  - The remaining $k - 1$ subsamples are used as training data
  - The cross-validation process is then repeated k times with each of the k samples used exactly once as the validation data

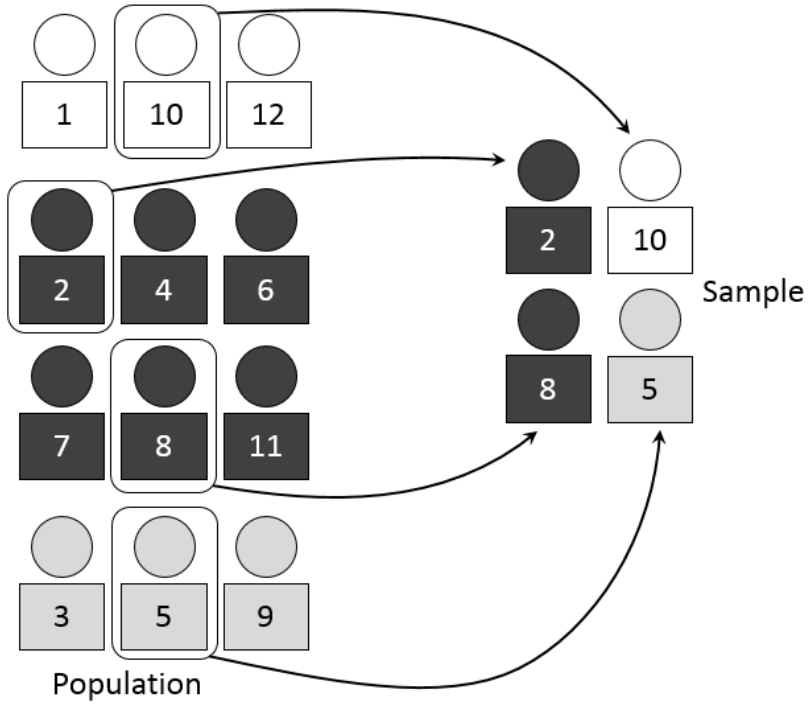


Leave One Out

- Leave one out
  - A special case of k-fold cross-validation
    - k is equal to the number of samples
  - Each learning set is created by taking all the samples except one, the test set being the sample left out
  - It is more computationally expensive than k-fold cross-validation and often results in high variance as an estimator for the test error
  - With such a small sample set leave one out cross validation would be the best

Stratified Sampling

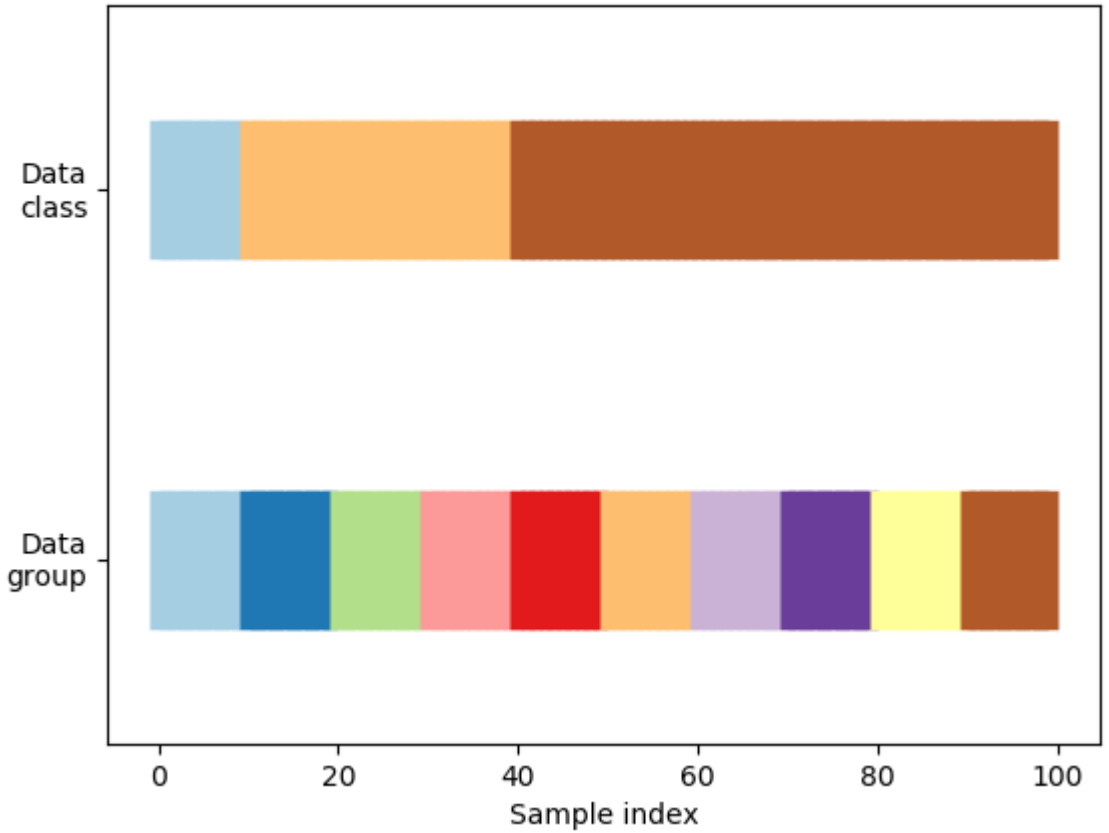
- Stratified sampling is a method of sampling from a population which can be partitioned into subpopulations
  - For classification analysis, stratified sampling aims at splitting one data set so that each split are similar with respect to class distribution
    - To ensure that the train and test sets have approximately the same percentage of samples of each target class as the complete set



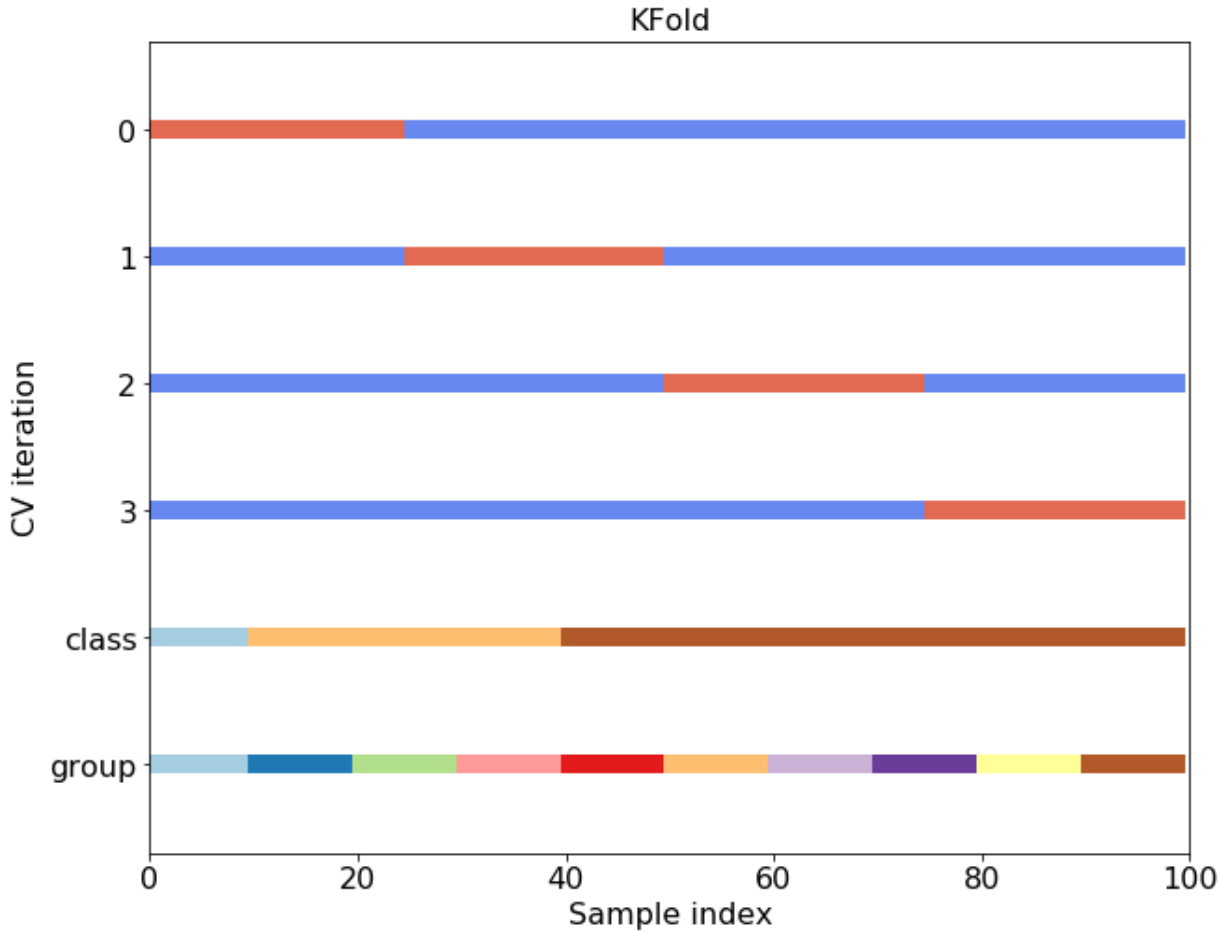
Group k-fold Cross-validation

- Group k-fold
  - A variation of k-fold which ensures that the same group is not represented in both testing and training sets
  - It can be used if the data is obtained from different subjects with several samples per-subject

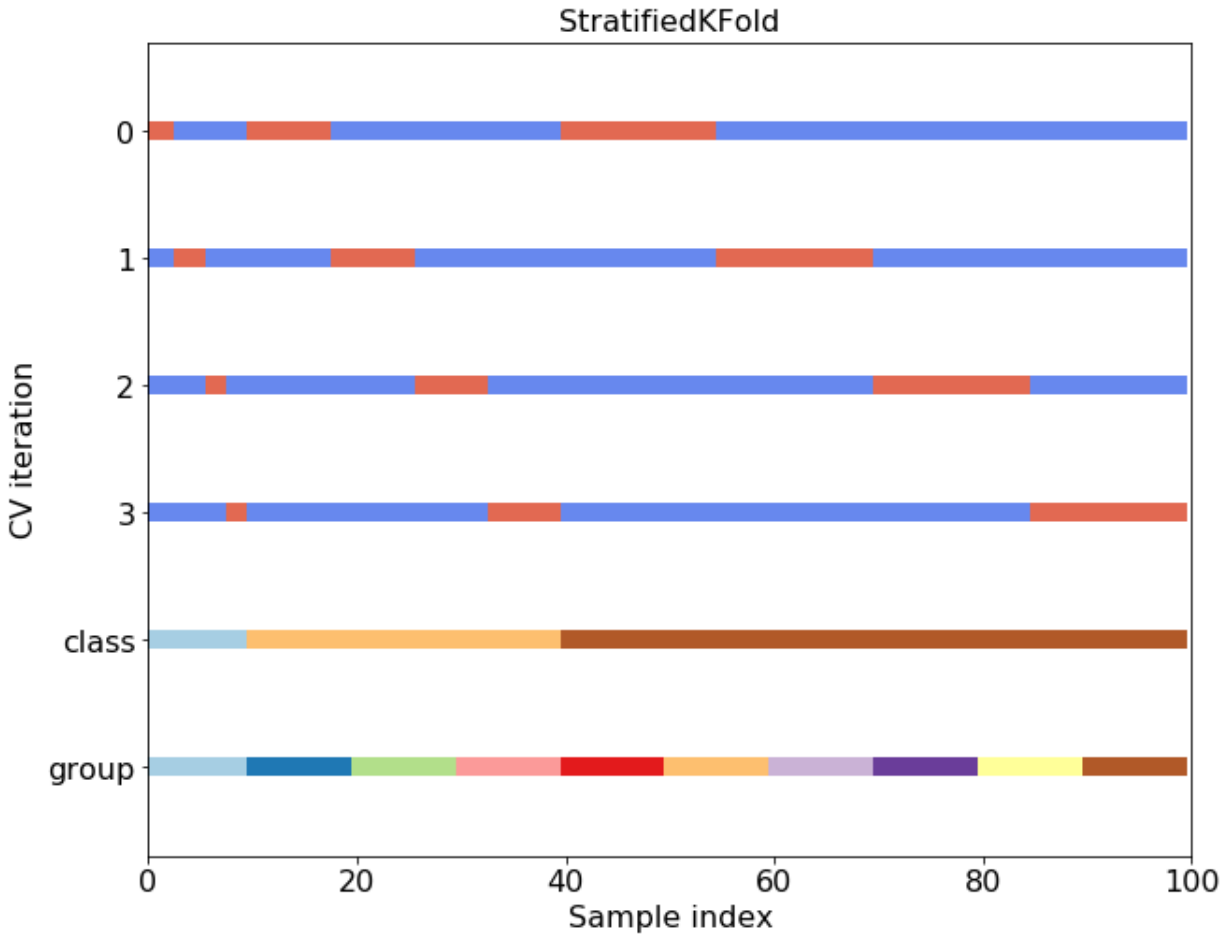
Visualizing Cross-validation



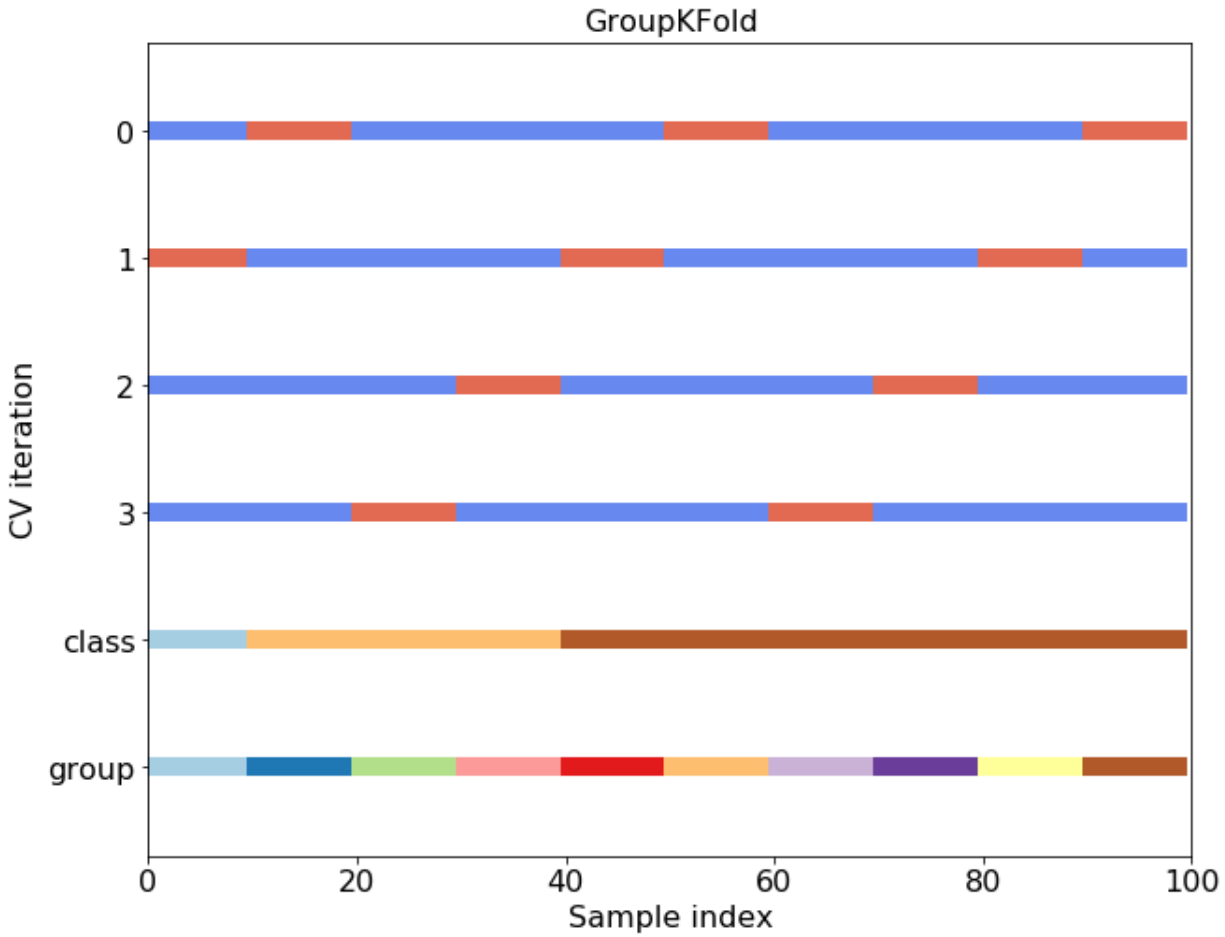
Visualizing Cross-validation



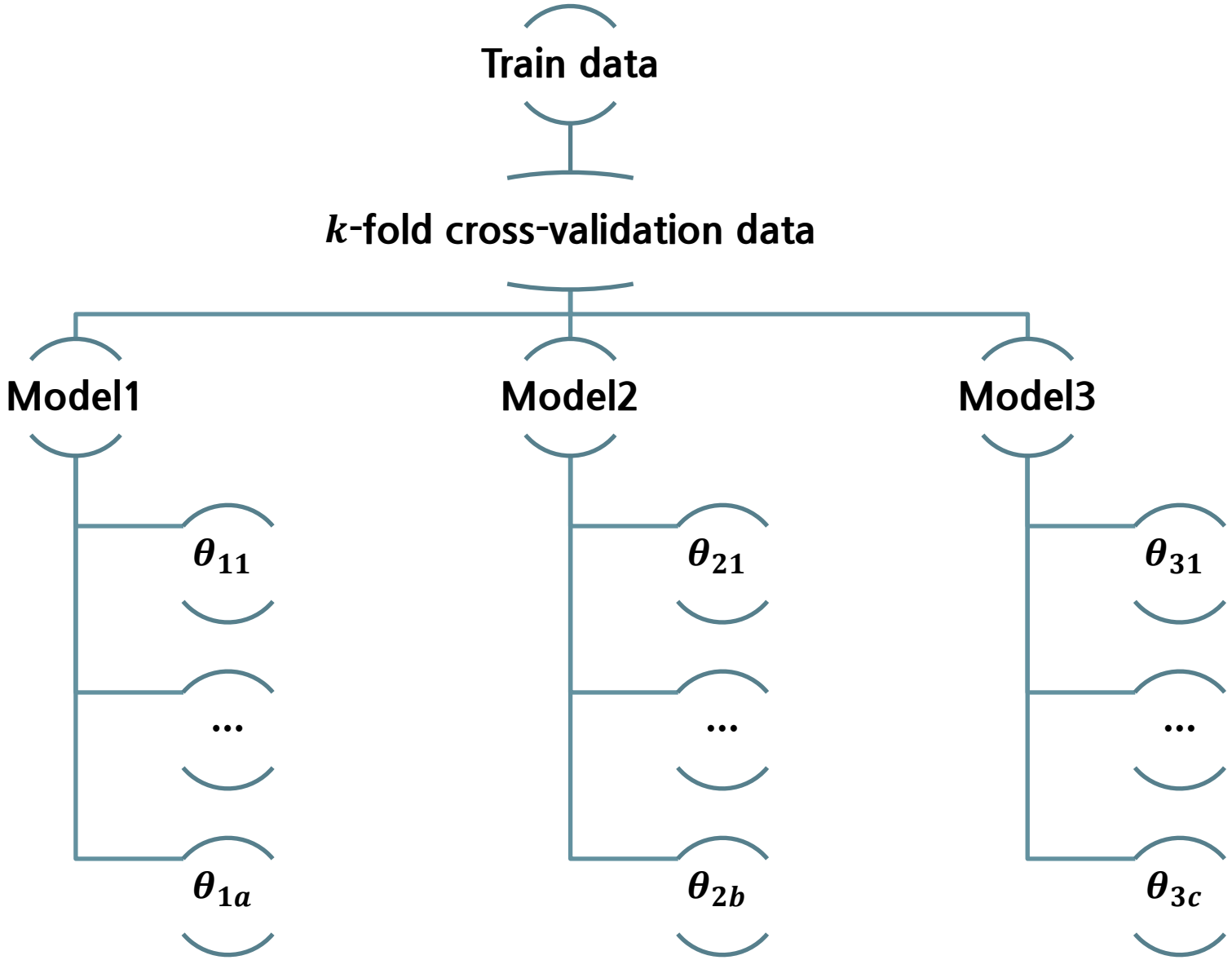
Visualizing Cross-validation



Visualizing Cross-validation



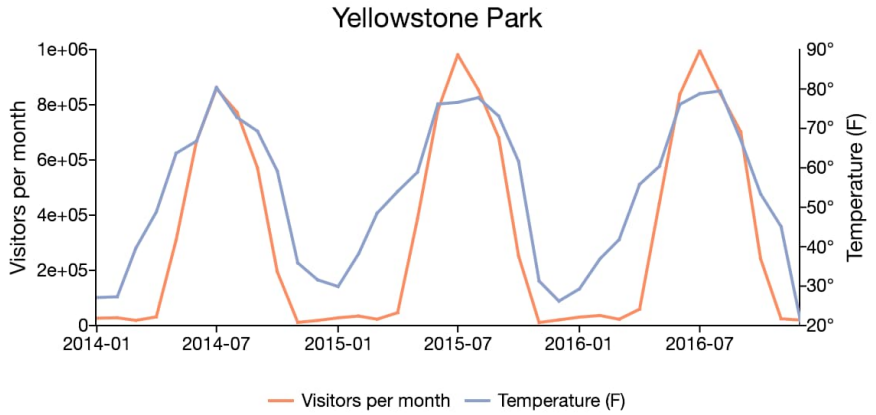
Model Selection



Time Series Analysis

Time Series

- A time series is a series of data points indexed (or listed or graphed) in time order
  - A sequence taken at successive equally spaced points in time
  - Ex) the daily closing values of stocks, height of ocean tides, birth rates of years, weight tracking, monthly sunspot observations



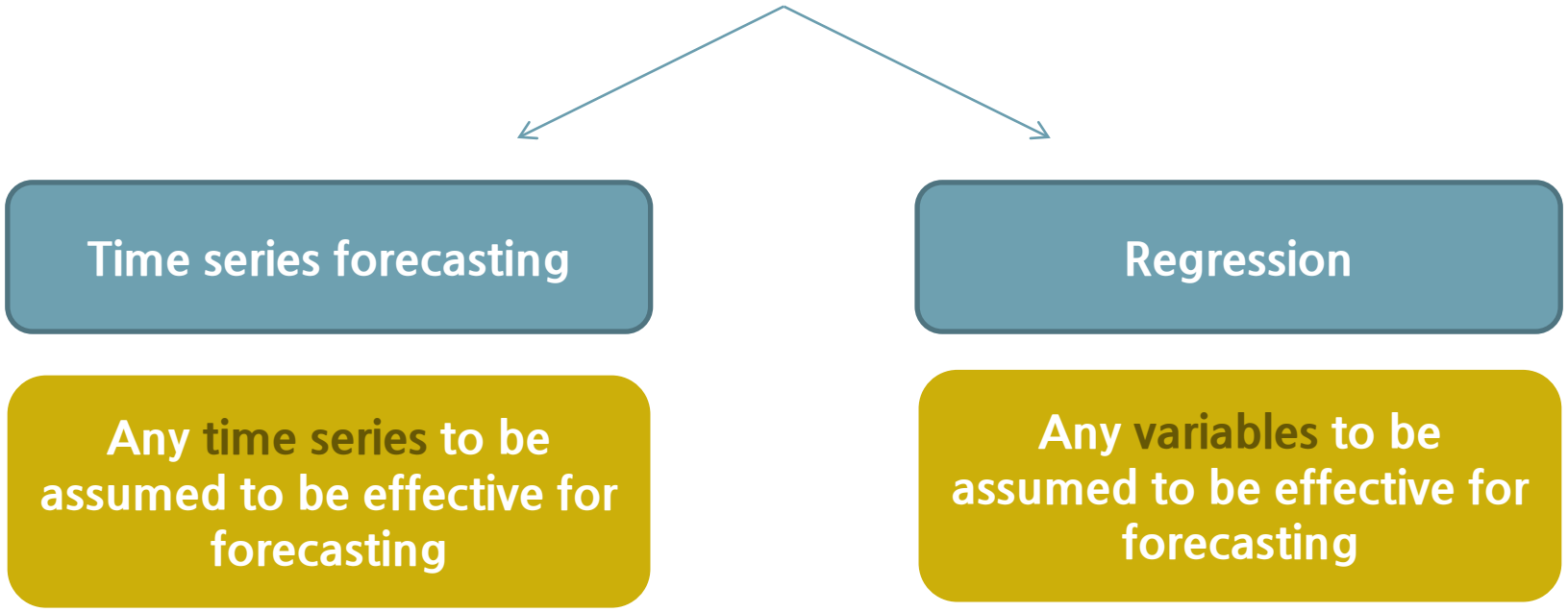
- **Time series analysis** comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data
- **Time series forecasting** is the use of a model to predict future values based on previously observed values.

Why Sales Forecasting is Important?

- Sale Planning
  - When your sales make their forecasts, they are also planning their future activities, providing each of them with a business plan for managing their territory
- Inventory Controls
  - The more accurate the sales forecast, the better prepared your company will be to manage its inventory, avoiding both overstock and stock-out situations.
- Supply Chain Management
  - When you can predict demand and manage production more efficiently, you also have better control over your supply chain
- Financial Planning
  - Anticipating sales gives you the information you need to predict revenue and profit
- Price Stability
  - With solid forecasting, the good levels of inventories that you maintain will prevent the need for panic sales to rid your business of excess merchandise
- Marketing
  - Sales forecasting gives marketing an advanced look at future sales and offers the opportunity to schedule promotions if it appears sales will be weak

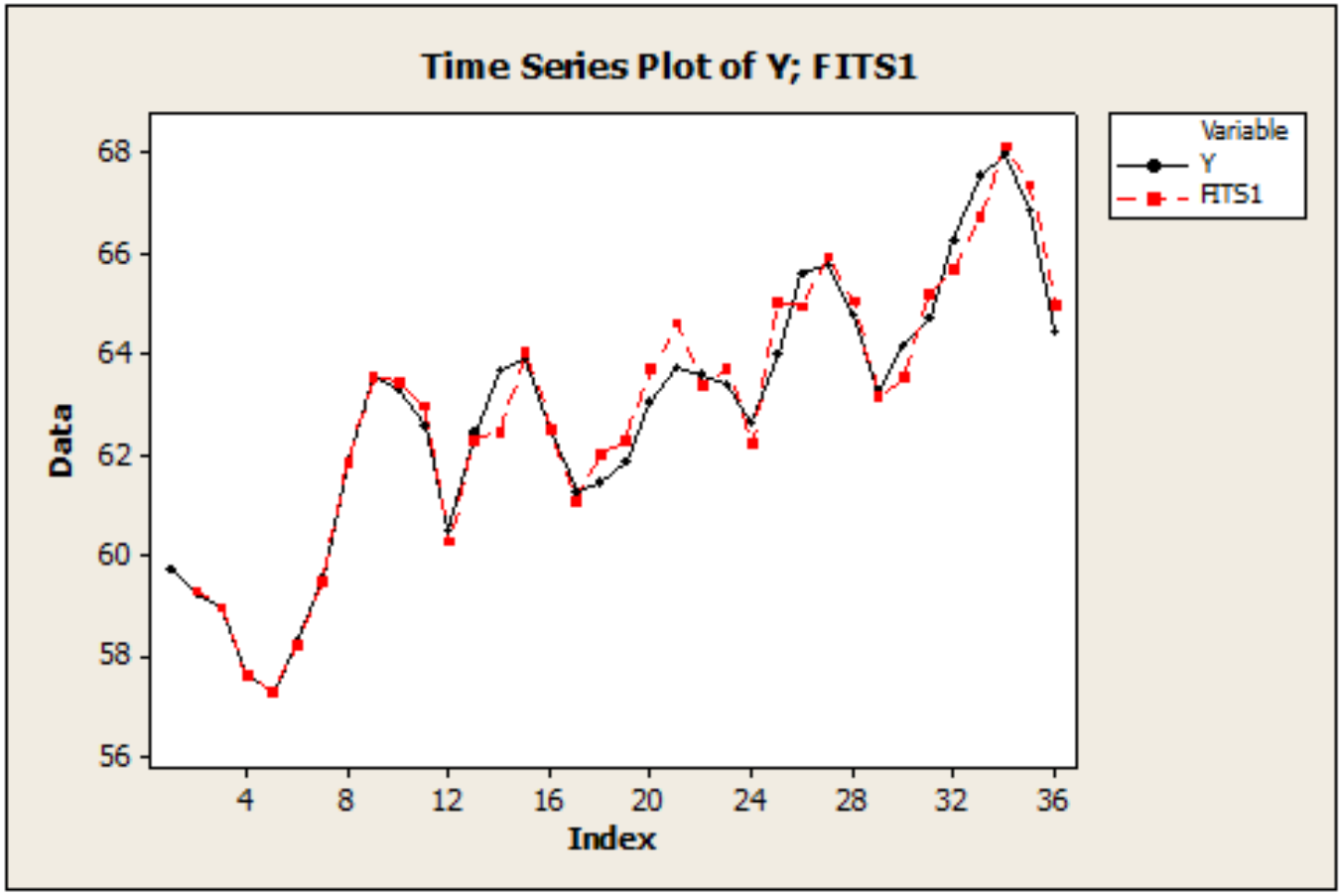
Define the Problem

- Target: sales of [something]
  - Numeric and continuous



Example of Time Series Forecasting

- Curve fitting
  - The process of constructing a curve, or mathematical function, that has the best fit to a series of data points



Autoregressive Model

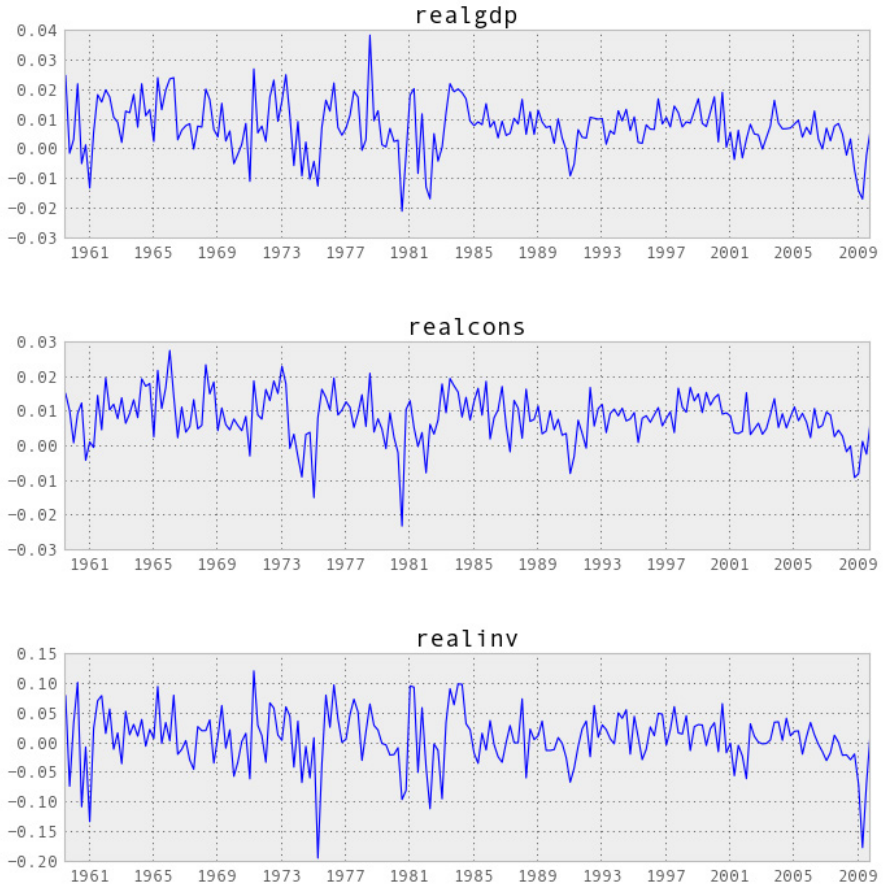
- The autoregressive model specifies that the output variable depends linearly on its own previous values and on a **stochastic term**
  - Stochastic: imperfectly predictable
  - A stochastic process is also called as random process
- $AR(p)$ model

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \epsilon_t$$

  - φ_i: parameters of the model
  - c: constant
  - ϵ_t: noise

Autoregressive Model

- Vector autoregression (VAR)
  - Capture the linear interdependencies among multiple time series
  - Generalize the univariate autoregressive model (AR model) by allowing for more than one evolving variable



Autoregressive Model

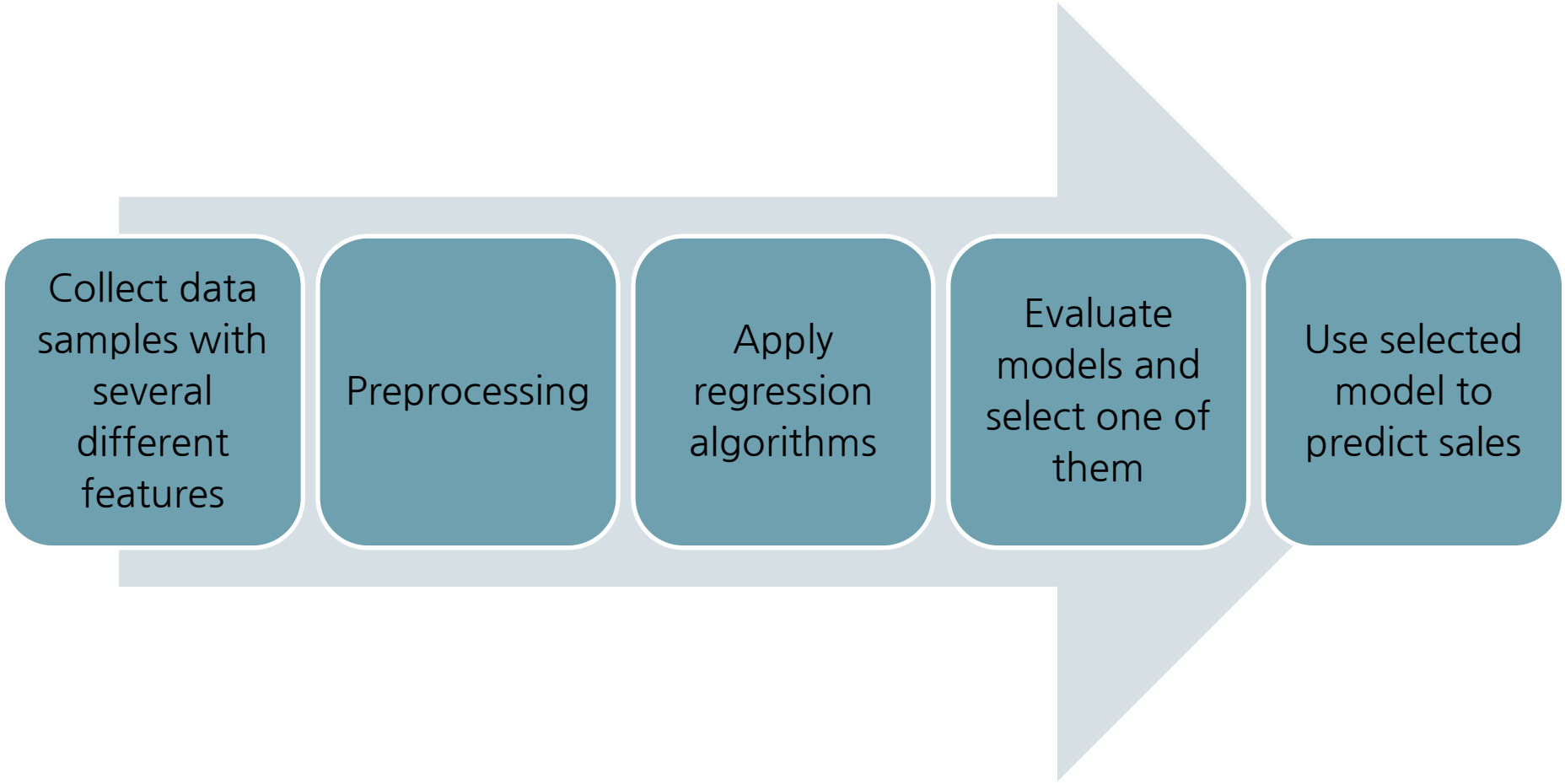
- $VAR(p)$
  - Consist of k variables (time series) $\rightarrow X_t, \epsilon_t, c$ are $k \times 1$ vector, φ_i is $k \times k$ matrix

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \epsilon_t$$

$$X_t = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ x_{t,3} \\ \vdots \\ x_{t,k} \end{bmatrix}, \epsilon_t = \begin{bmatrix} e_{t,1} \\ e_{t,2} \\ e_{t,3} \\ \vdots \\ e_{t,k} \end{bmatrix}, c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_k \end{bmatrix}, \varphi_i = \begin{bmatrix} a_{i,11} & \cdots & a_{i,1k} \\ \vdots & \ddots & \vdots \\ a_{i,k1} & \cdots & a_{i,kk} \end{bmatrix}$$

Approach to Predict Sales by Regression

- Assumption
  - Some time-invariant factors also affect sales
- Process of building regression model and its utilization



Programming Exercise

Split Dataset

- Split dataset

```
from sklearn.model_selection import train_test_split
from sklearn import datasets

diabetes = datasets.load_diabetes()
X = diabetes.data
y = diabetes.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Split Dataset

- Compare $k - NN$ models varying with k
  - k: 5 or 7

```
from sklearn.neighbors import KNeighborsRegressor
```

  - change random_state when splitting data
    - 100 or 200
  - Compare R^2

- Which k is better?

Cross-validation

- Generate cross-validation sets
  - n: the number of observations in train data
  - n_splits: k

```
import numpy as np
from sklearn.model_selection import KFold

X = ["a", "b", "c", "d"]
kf = KFold(n_splits=2)
for train, test in kf.split(X):
    print("%s %s" % (train, test))
```

  - KFold divide data set into train and validation set in order of row number
    - If data set is sorted by classes, you have to set “shuffle=True” to randomly divide set or use **ShuffleSplit**

Cross-validation

- Generate cross-validation sets
  - If you want divide train and validation with keeping output ratios, use **StratifiedKFold**

```
from sklearn.model_selection import StratifiedKFold

X = np.ones(10)
y = [0, 0, 0, 0, 1, 1, 1, 1, 1, 1]
skf = StratifiedKFold(n_splits=3)
for train, test in skf.split(X, y):
    print("%s %s" % (train, test))
```

    - Additionally, use label information
    - If you want to shuffle samples within each class, set “shuffle=True” or use **StratifiedShuffleSplit**

Cross-validation

- Generate cross-validation sets
  - **GroupKFold**

```
from sklearn.model_selection import GroupKFold

X = [0.1, 0.2, 2.2, 2.4, 2.3, 4.55, 5.8, 8.8, 9, 10]
y = ["a", "b", "b", "b", "c", "c", "c", "d", "d", "d"]
groups = [1, 1, 1, 2, 2, 2, 3, 3, 3, 3]

gkf = GroupKFold(n_splits=3)
for train, test in gkf.split(X, y, groups=groups):
    print("%s %s" % (train, test))
```

    - Additionally, use group information
    - If you want to shuffle samples within each group, set “shuffle=True” or use **GroupShuffleSplit**

Parameter Selection

- Using the digits dataset, determine parameter C of support vector classifier based on accuracy

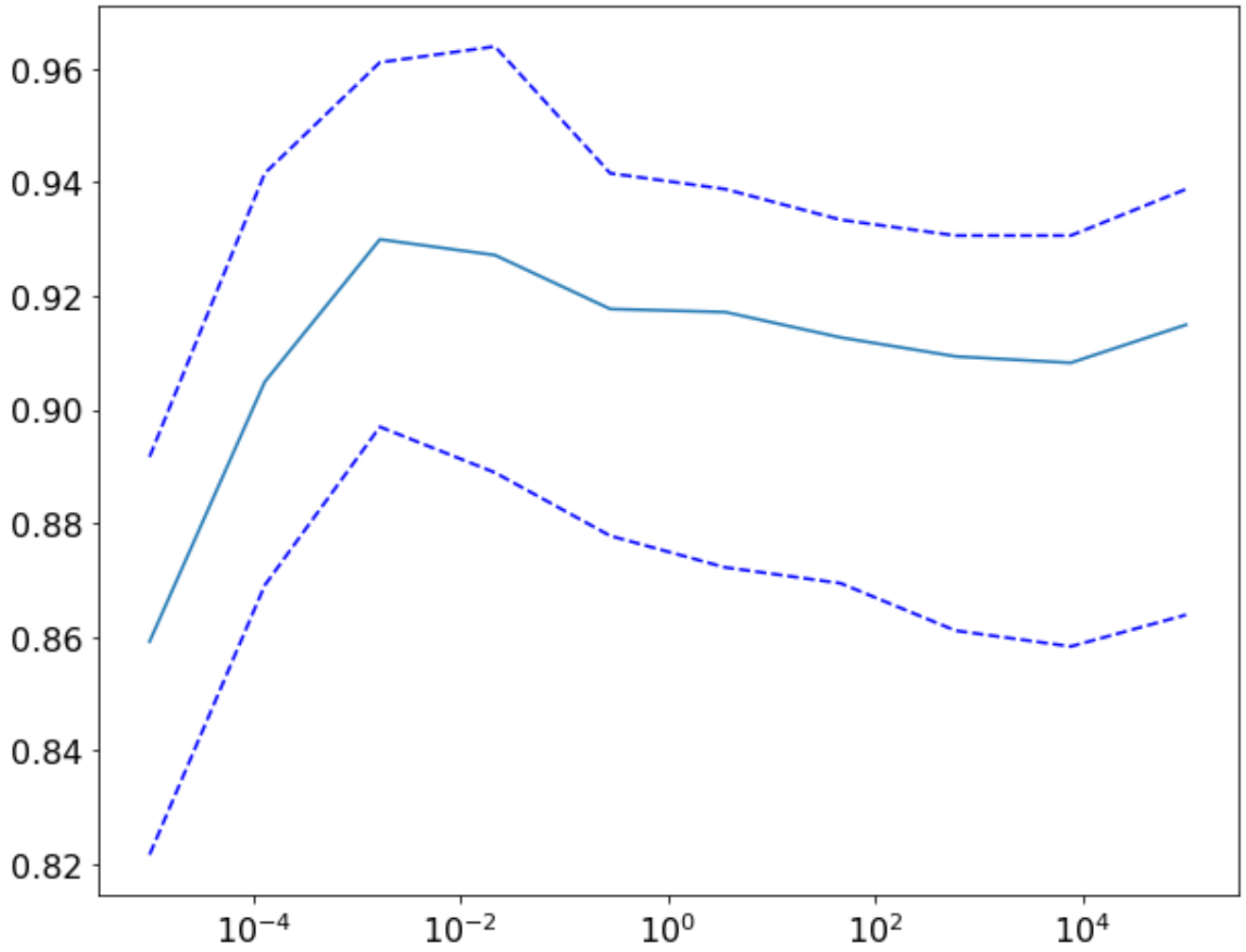
```
import numpy as np
from sklearn import datasets
from sklearn.linear_model import LogisticRegression

digits = datasets.load_digits()
X = digits.data
y = digits.target

C_s = np.logspace(-10, 0, 10)
```

Parameter Selection

- Cross-validation score for different C



Next Week

Rossmann Store Sales Forecasting

- Rossmann
  - The company which operates over 3,000 drug stores in 7 European countries

- The purpose of the problem
  - Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance
  - Predict 6 weeks of daily sales for 1,115 stores located across Germany
  - Ref: https://www.kaggle.com/c/rossmann-store-sales

Provided Data Set

- Train
  - Historical data including Sales
- Test
  - Historical data excluding Sales
- Store
  - Supplemental information about the stores

Data Description

- Variable description (Train)
  - Store: a unique Id for each store
  - DayofWeek
  - Date
  - **Sales: the turnover for any given day (this is what you are predicting)**
  - Customers: the number of customers on a given day
  - Open: an indicator for whether the store was open
    0 = closed, 1 = open
  - Promo: indicates whether a store is running a promo on that day
  - StateHoliday: indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends.
    a = public holiday, b = Easter holiday, c = Christmas, 0 = None
  - SchoolHoliday: indicates if the (Store, Date) was affected by the closure of public schools

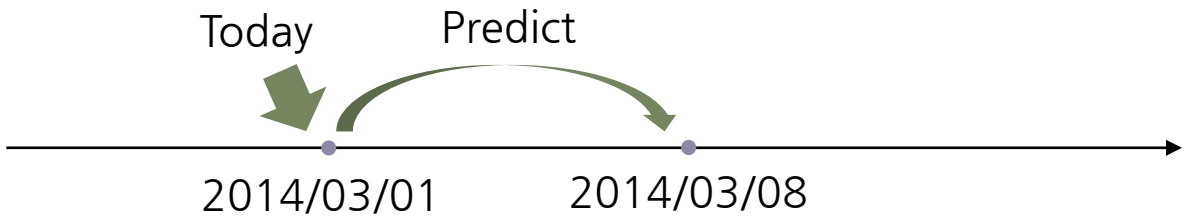
Data Description

- Variable description (Store)
  - StoreType: differentiates between 4 different store models (a, b, c, d)
  - Assortment: describes an assortment level
    a = basic, b = extra, c = extended
  - CompetitionDistance: distance in meters to the nearest competitor store
  - CompetitionOpenSince[Month/Year]: gives the approximate year and month of the time the nearest competitor was opened
  - Promo2: a continuing and consecutive promotion for some stores
    0 = store is not participating, 1 = store is participating
  - Promo2Since[Year/Week]: describes the year and calendar week when the store started participating in Promo2
  - PromoInterval: describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g.
    "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

Assignment

Assignment 04

- The purpose
  - Forecast sales of stores after one week



    - You can use information from a week ago

- Split data
  - Training set: use samples in 2013 and 2014
  - Validation set: use samples in 2015

Assignment 04

- Select explanatory variables to predict sales
  - List variables used in learning
    - Explain why
- Select learning methods to predict sales
  - At least two
    - It is possible to find the best parameter using cross-validation
  - compare the trained models using validation set (samples in 2015)
- Summarize procedures and results
  - Describe preprocessing steps
  - Explain the results from different algorithms