Data Mining for Business Analytics

Ch. 02 Overview of the Data Mining Process

2023.03.13.

Contents

2.1 Introduction

데이터 모델링 과정

Define purpose	Obtain data	Explore & clean data	Determine DM task	Choose DM Methods	Apply methods, select final model	Evaluate performance	Deploy

Business Analytics 핵심 요소

예측 분석(Predictive Analytics)

- 분류(Classification) / 예측(Prediction)

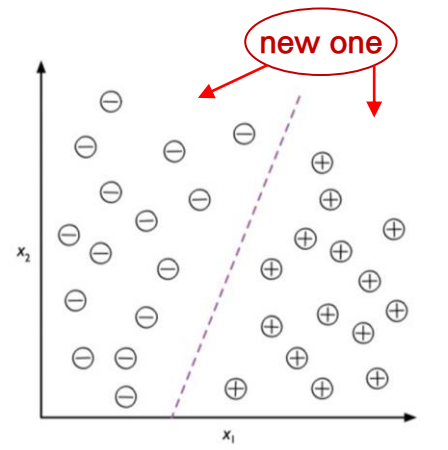
데이터 마이닝에서 많이 쓰이는 데이터베이스 기법

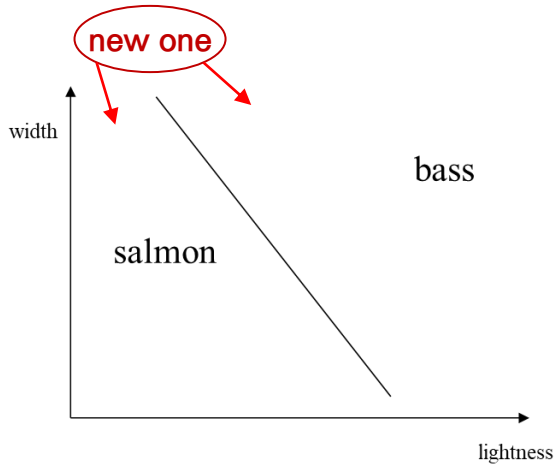
- OLAP(Online Analytic Processing)과 SQL(Structured Query Language)
- 예) 신용카드 고객 중에서 특정 지역에 살고, 연간지출액이 2만 달러를 넘고, 자기 주택을 소유하고 있고, 적어도 95%는 결제일을 맞춘 고객을 찾는 문제

2.2 Core Ideas in Data Mining

Classification

- 데이터 분석의 가장 많이 다루는 문제
- 데이터를 집단으로 구분하기 위함
- “응답” or “응답하지 않음” / “정상” or “사기” / “정상” or “고장” / “회복” or “진행중” or “사망”

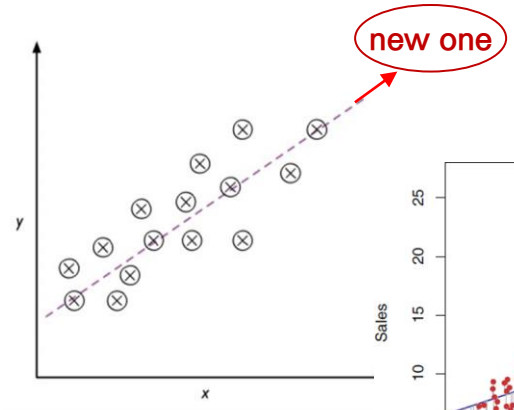


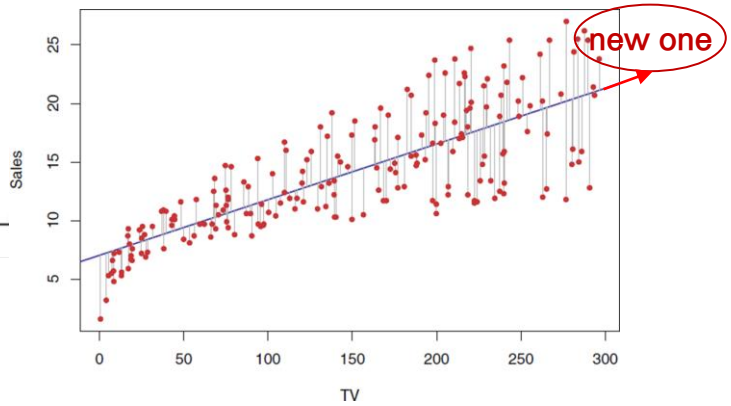


Prediction

- 예측하고자 하는 변수가 숫자로 표현된 연속형 변수일 경우 예측 문제로 분류

※ 예측하고자 하는 변수가 범주형인 경우 분류 문제

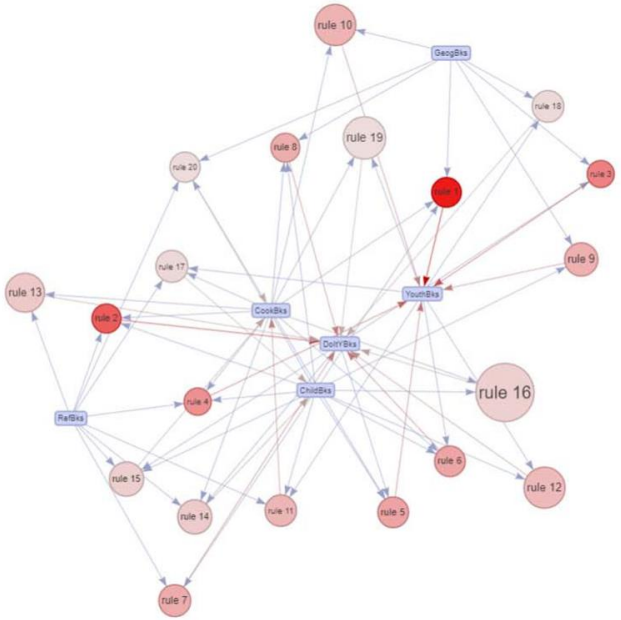




2.2 Core Ideas in Data Mining

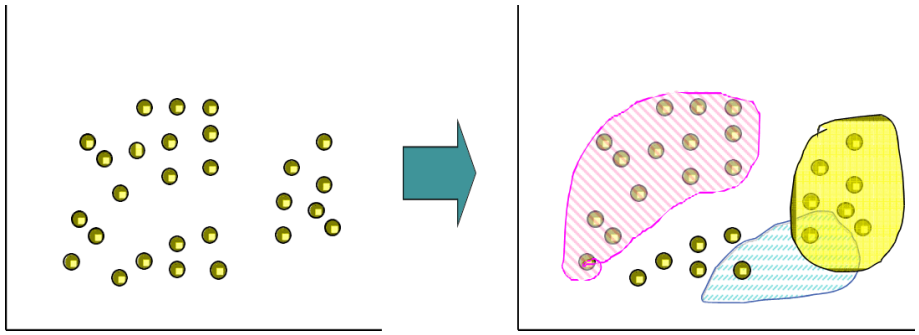
Association Rules and Recommendation Systems

- 대량의 고객거래 데이터베이스는 구매항목들 간의 연관성, 즉 어떤 항목이 어떤 항목과 관련 되는지에 대한 분석
- Collaborative Filtering: 개개인의 포괄적인 과거 구매정보와 다른 사람들의 구매정보를 이용하여 개개인의 구매성향을 예측하는 추천 시스템



Cluster Analysis (as Data reduction method)

- 비지도학습으로 데이터를 동질적인 군집들로 세분화
- 측정치들로 구성된 레코드로부터 측정값들이 유사한 레코드의 모임 또는 군집을 형성하기 위해 사용



2.2 Core Ideas in Data Mining

Dimension Reduction

차원 축소

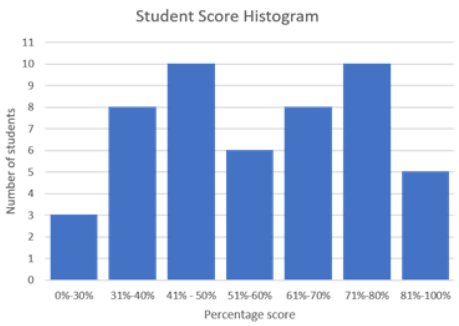
- 변수의 개수를 줄이는 과정
- 지도 학습 전에 수행하며, 예측 성능을 향상시키고 해석의 용이성 증대 목적

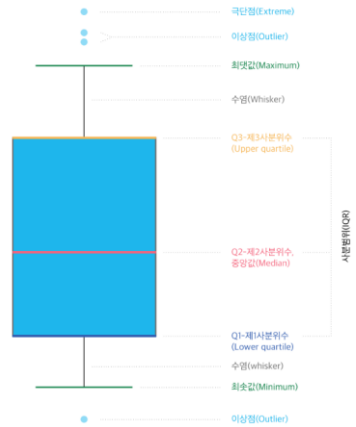
Data Exploration

- 데이터의 전반에 관한 이해와 이상치 탐지 목적
- 수치적 혹은 시각적으로 데이터를 요약하는 방법론 사용

Data Visualization

- 수치형 변수: 히스토그램(histogram)과 상자그림(boxplot)을 이용하여, 변수값의 분포를 파악하고 극단치(outliers)를 찾음
- 범주형 변수: 차트(charts)와 원형 차트(pie charts)를 이용. 변수 간의 가능한 관계들, 관계유형 그리고 극단치를 찾기 위해 한 쌍의 수치형 변수에 대한 산점도(scatter plots)을 조사





2.2 Core Ideas in Data Mining

Supervised Learning

- 분류 또는 예측하고자 하는 변수가 존재할 경우 이를 labeled data(결과값)로 놓고 예측변수와의 관계를 통해 모델링
- 학습 데이터를 이용하여 예측변수와 출력변수 간의 관계를 학습, 훈련함
- 학습 데이터로부터 모델이 구축되면 결과를 알고 있는 검증 데이터를 사용하여 모델의 성능을 평가하고 다른 방법론과 비교
- 다수의 모델을 비교하고자 할 경우, 평가 데이터를 통해 성능을 비교, 평가함
- 최종적으로 검증이 끝난 모델은 종속변수의 값을 모르는 미래의 데이터 예측에 사용
- 예) 단순 선형 회귀분석, 판별 분석, 역전파 신경망

Unsupervised Learning

- 예측 또는 분류를 위해 필요한 출력변수가 없는 경우에 사용되는 알고리즘
- 예) 연관 규칙, 차원 축소 기법, 군집 분석 등

2.3 The Steps in Data Mining

Step 1. Develop an understanding of the purpose of the data mining project

- 데이터 마이닝 프로젝트의 목적을 정확히 설정

Step 2. Obtain the dataset to be used in the analytics

- 분석에 필요한 데이터 획득
- 대량의 데이터베이스에서 무작위 표본 추출 또는 서로 다른 데이터베이스에서 별도로 추출하여 통합

Step 3. Explore, clean, and preprocess the data

- 데이터의 탐색, 정제, 전처리 데이터가 방대해질 수록 문제를 찾기 어려움.
- 데이터가 타당한 조건에 있는지를 검증: 결측치, 극단치 처리, 변수 간의 관계를 산점도 등으로 검토, 변수에 대한 정의, 측정단위, 측정 기간 등에 대한 일관성 체크

Step 4. Reduce the data dimension, if necessary

- 필요시 데이터 차원 축소 필요할 경우 활용
- 불필요한 변수의 제거, 변수 값의 변환, 새로운 변수의 생성 등

2.3 The Steps in Data Mining

Step 5. Determine the data mining task (classification, prediction, clustering, etc.)

- 데이터 마이닝 문제 결정(분류, 예측, 군집 등)

Step 6. Partition the data (for supervised tasks)

- 데이터 분할(지도학습의 경우): 학습(training) / 검증(validation) / 평가(test)

Step 7. Choose the data mining techniques to be used (regression, neural nets, hierarchical clustering, and so on)

- 사용할 데이터 마이닝 기법 선택(회귀분석, 인공신경망, 계층 군집 등)

Step 8. Use algorithms to perform the task

- 데이터 마이닝 프로세스의 여러 단계를 반복적으로 수행하여 가장 좋은 알고리즘 탐색
- 변수의 조합 시도, 알고리즘의 셋팅값 변경

2.3 The Steps in Data Mining

Step 9. Interpret the results of the algorithms

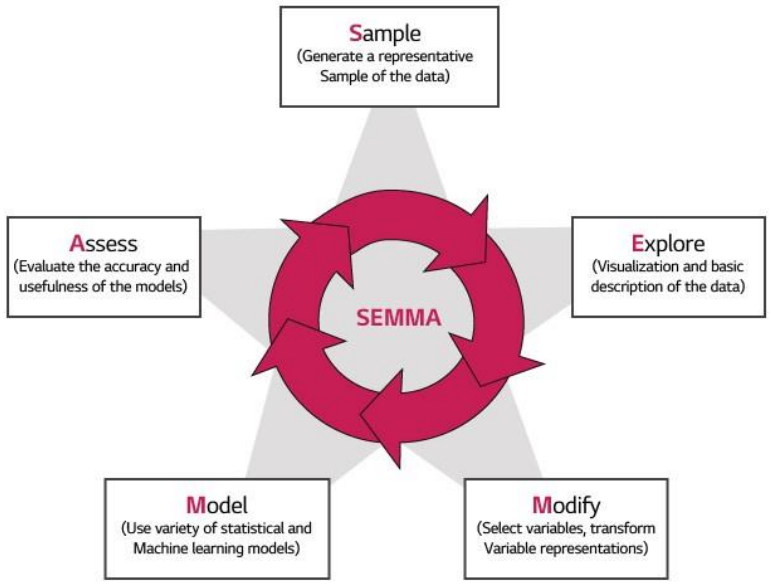
- 가장 효율적인 알고리즘을 찾아내고, 검증 데이터를 이용하여 구축된 알고리즘의 성능을 평가

Step 10. Deploy the model

- 구축된 모델을 운용시스템에 탑재하여 실제 의사결정에 적용하는 단계

cf. SEMMA, methodology by SAS

- 표본 추출(Sampling): Training / Validation / Test data set
- 탐색 (Exploration): 데이터에 포함된 변수들이 어떠한 분포를 하고 있으며, 변수들 간의 관계가 어떠한 것인지를 파악
- 변환 (Modification) 및 변수 선정: 분석 목적에 적합한 형태로 변환
- 모델링 (Modeling): 문제의 특성에 맞는 적절한 기법을 통해서 모형 개발
- 평가(Assessment): 검증 데이터(validation data set)를 통해 여러 종류의 모형을 비교평가



2.4 Preliminary Steps

Predicting Home Values in the West Roxbury Neighborhood

- 보스턴시에서 공개한 보스턴 지역 부동산 평가 데이터
- 2014년 보스턴 남서부 웨스트 록스베리 지역의 단독주택 정보 (14개 변수 / 5,802개 주택)
- 결과값: Total Value(주택 가격)

300 이하 이상 분류해도 예측임. 정렬이 없기 때문.

TOTAL VALUE	TAX	LOT SQFT	YR BUILT	GROSS AREA	LIVING AREA	FLOORS	ROOMS	BEDROOMS	FULL BATH	HALF BATH	KITCHEN	FIREPLACE	REMODEL
344.2	4330	9965	1880	2436	1352	2	6	3	1	1	1	0	None
412.6	5190	6590	1945	3108	1976	2	10	4	2	1	1	0	Recent
330.1	4152	7500	1890	2294	1371	2	8	4	1	1	1	0	None
498.6	6272	13773	1957	5032	2608	1	9	5	1	1	1	1	None
331.5	4170	5000	1910	2370	1438	2	7	3	2	0	1	0	None

변수	설명
TOTAL VALUE	주택가격(단위: 1,000달러)
TAX	세금(단위: 달러)
LOT SQFT	총 부지 면적(단위: 제곱피트)
YR BUILT	건축 연도
GROSS AREA	총 바닥 면적
LIVING AREA	주거공간 면적
FLOORS	층 개수

변수	설명
ROOMS	방 개수
BEDROOMS	침실 개수
FULL BATH	욕실 개수
HALF BATH	보조욕실 개수
KITCHEN	주방 개수
FIREPLACE	벽난로 개수
REMODEL	리모델링 시점(최근/오래전/안 함)

2.4 Preliminary Steps

[실습] Table 2.4

Sampling from a Database

- 대개, 일부 레코드만 가지고 algorithm을 수행하고자 함
- Computing power의 한계로 레코드나 variable의 개수에서 limitation을 가짐
  딥러닝(GPU) 영향이 크고, DM(CPU, RAM) 영향이 큼
- 수백, 수천 개 정도의 레코드로도 정확한 model 수립이 가능

Oversampling Rare Events in Classification Tasks

1. 관심 대상 데이터가 희귀할 경우, 즉 정상 데이터는 많으나 비정상 데이터는 부족할 경우 → 샘플링시 소수 클래스에 보다 큰 가중치를 주어 클래스 관측치 수에 균형을 맞출 수 있음
2. 각 클래스의 오분류에 가중치를 주어 해결 가능

- 모델을 수립하는 데에 아무 정보를 주지 않는 수많은 non-rare event를 포함하는 sampling
  - ✓ 이 관심 대상 event에 대해 overweight하게 됨
  - ✓ Rare event를 찾기 위한 비용도 발생
  - ✓ 응답하지 않은 사람을 응답자로 잘못 분류하는 비용과, 응답자를 찾아내는 비용 간의 균형 필요

2.4 Preliminary Steps

Preprocessing and Cleaning the Data

(1) Type of Variables

- numerical or text (or character)
- continuous(연속형: 대개 주어진 범위 내의 실수), integer(오직 정수 값), or categorical(범주형: 일정 범위의 값을 하나로 범주로 가정)
- categorical: numerical (1,2,3) or text (현금결제, 비현금결제, 파산)
  - ✓ unordered (순위정보를 갖지 않는 범주형): “nominal” (명목형 변수) – Asia, Europe, North America
  - ✓ ordered (순위정보를 갖는 범주형): “ordinal” (순위형 변수) – high, medium, low
- 각 방법마다 적용가능 variable에 제한이 있을 수 있음(ex. Naïve Bayes 는 categorical)

[All columns data types]

```
TOTAL_VALUE      float64
TAX                int64
LOT_SQFT           int64
YR_BUILT           int64
GROSS_AREA         int64
LIVING_AREA        int64
FLOORS           float64
ROOMS              int64
BEDROOMS           int64
FULL_BATH          int64
HALF_BATH          int64
KITCHEN            int64
FIREPLACE          int64
REMODEL         category
dtype: object
```

2.4 Preliminary Steps

[실습] Table 2.6

Preprocessing and Cleaning the Data

변수	설명
REMODEL	리모델링 시점(최근/오래전/안 함)

(2) Handling Categorical Variables

- 범주형 변수가 순위정보를 갖고 있는 경우(연령 구간, 신용등급 등): 연속형 변수인 것처럼 변수를 있는 그대로 사용
- 범주형 변수가 명목형인 경우: 이진 분류의 더미변수로 분할 사용
  - ✓ 학생: 예/아니오 무직: 예/아니오

[Dummy variables]

	REMODEL_Old	REMODEL_Recent
0	0	0
1	0	1
2	0	0
3	0	0
4	0	0

(3) Variable Selection

- 더 많은 변수의 활용이 더 나은 결과를 보장하지 않음 → 꼭 필요한 변수의 사용이 바람직함
- 많은 변수를 모델에 사용하는 경우 변수 간의 상관관계 파악의 복잡성이 증가
  - ✓ 예를 들어 종속변수 Y와 하나의 변수 X의 상관관계를 알기 위해 15개의 데이터로 충분할 수 있음
  - ✓ 만약 15개의 변수 X를 사용한다면 이보다 훨씬 많은 데이터 필요

2.4 Preliminary Steps

Preprocessing and Cleaning the Data

(4) How Many Variables and How Much Data?

- 경험에 의한 법칙(rules of thumb): 모든 예측변수는 각각 10개의 레코드 필요
- Delmaster and Hancock(2001, p. 68): 최소한 6*m*p 개의 레코드 필요(m: 클래스의 수, p는 변수의 개수)
- 도메인 지식을 가지고 있는 사람으로부터의 정보는 변수의 포함 여부를 결정할 때 중요하며 모델의 정확도를 높이고 오차를 줄일 수 있음

(5) Outliers

- 대부분의 데이터로부터 멀리 떨어진 값들은 극단치(Outliers)로 불리운다.
- Rule of thumb: 평균으로부터 표준편차의 3배보다 더 멀리 떨어져 있는 값은 극단치에 해당한다
- 잘못된 데이터의 경우: 도메인 지식이나 상식을 활용하여 제거(체온 50도 등)
- 극단치가 적을 경우 이를 제거
- 각 column 별로 sorting하여 outlier 탐색, 또는 max-min value 검토

2.4 Preliminary Steps

[실습] Table 2.7

Preprocessing and Cleaning the Data

(6) Missing Values

- 결측값을 갖는 레코드의 수가 적다면 그 레코드는 제외하고 분석
- 변수의 개수가 많다면 단순 삭제에 문제가 있음
- 결측값을 데이터에서 차지하는 비율이 낮더라도 많은 레코드에 영향
  - ✓ ex. 30개 변수, 5%의 결측값 → 거의 80%의 데이터 삭제 (주어진 레코드가 결측값을 가지고 있지 않을 확률: $0.95^{30} = 0.215$)
- 결측값을 다른 변수의 해당 값들을 기반으로 대체하는 방법 (평균, 중앙값 등)
- 변수의 중요도를 측정하여 판단: 해당 변수가 예측에 미치는 영향을 판단하여 삭제 또는 대체 여부 결정

```
Number of rows with valid BEDROOMS values before: 5802
Number of rows with valid BEDROOMS values after setting to NAN: 5792

Number of rows after removing rows with missing values: 5792
```

```
medianBedrooms = housing_df['BEDROOMS'].median()
housing_df.BEDROOMS = housing_df.BEDROOMS.fillna(value=medianBedrooms)
```

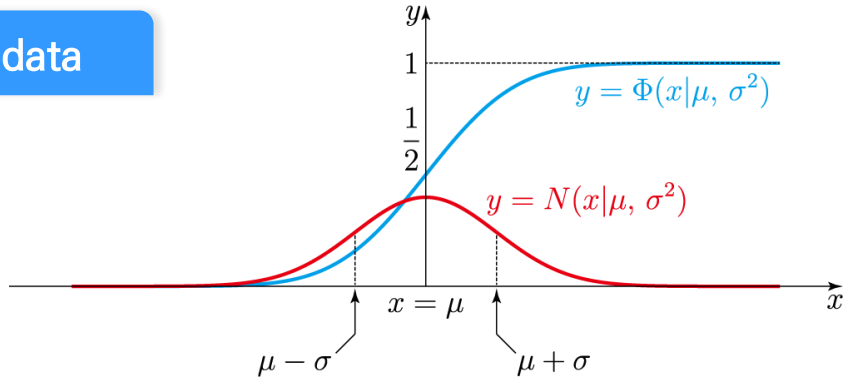
```
Number of rows with valid BEDROOMS values after filling NA values: 5802
```

2.4 Preliminary Steps

Preprocessing and Cleaning the Data

(7) Normalizing (Standardizing) and Rescaling data

- 정규화(Standardizing): 변수의 분포를 정규분포(Normal Distribution or Gaussian Distribution)로 변환하는 과정
  - ✓ z = (X – μ) / σ (μ: 평균, σ: 표준편차)
  - ✓ z-score: 평균으로부터 벗어난 표준편차의 수



정규분포 곡선 / 누적분포 곡선

- 표준화(Normalizing): 모든 변수의 스케일을 동일하게 변환하는 과정
- 변수의 스케일이 [0, 1] 안에 존재하도록 변환

$$x' = \frac{x - \mu}{x_{max} - x_{min}}$$

※ 정규화나 표준화가 필요한 이유

✓ 예를 들어 군집분석의 경우, 한 변수의 달러 단위가 1,000이고 나머지 변수의 단위가 10이라면 달러 변수가 전체 거리 계산을 지배할 수 있음

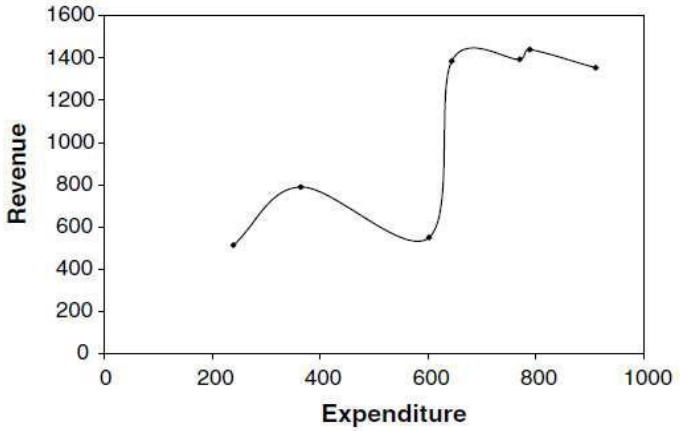
$$d = \sqrt{(1{,}345 - 12)^2 + (3{,}357 - 36)^2 + \cdots}$$

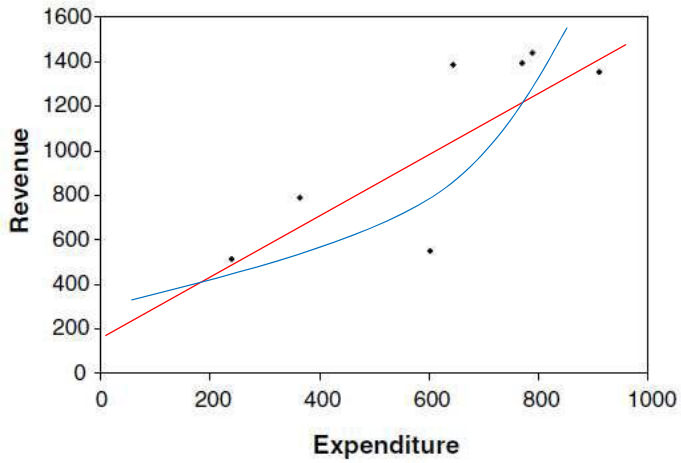
$$d = \sqrt{(13 - 3.5)^2 + (17 - 2.1)^2 + \cdots}$$

2.5 Predictive Power and Overfitting

Overfitting

- 예) 광고비 지출과 매출
  - ✓ 위 곡선: 현재의 데이터를 정확히 설명. 하지만 미래의 데이터가 항상 현재 데이터의 형태를 따른다는 보장은 없음 → 현재 데이터에 과적합(overfitting)되어 있음
  - ✓ 아래 그림: 미래의 데이터를 더 잘 예측할 것으로 보임
- 과적합(Overfitting)
  - ✓ 학습용 데이터에 완전히 적합
  - ✓ 학습용 집합에서 잡음(noise)도 모형화하기 때문에 평가용 집합에서 전체 오차는 일반적으로 증가



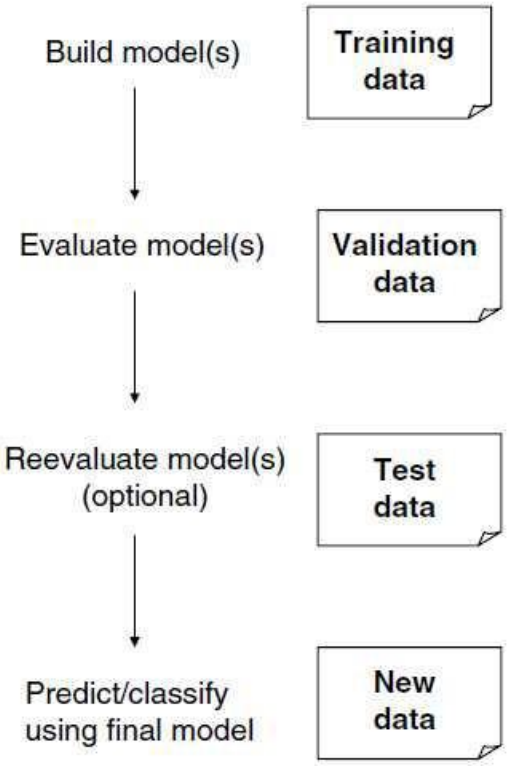


2.5 Predictive Power and Overfitting

Creation and Use of Data Partitions

- 데이터 분할
  - ✓ 미리 정해진 비율에 따라 랜덤하게 분할
  - ✓ 사용자가 관측값 마다 어느 부분에 속할지 지정하여 분할

- 학습 데이터(Training Partition)
  - ✓ 일반적으로 가장 크기가 큰 집합으로 모델을 구축하기 위해 사용되는 데이터
  - ✓ 여러 모델을 개발할 경우, 일반적으로 동일한 학습 데이터 사용
- 검증 데이터(Validation partition) 튜닝시 많이 사용
  - ✓ 때때로 평가 데이터(test partition)으로 불리며 모형을 비교하여 가장 좋은 모형을 선택하기 위해 각각의 모형의 성과를 검증하기 위해 사용
  - ✓ 어떤 알고리즘(예를 들어 분류와 회귀나무)에서는 모형을 조율하고 향상시키기 위해 자동화된 방식으로 사용 가능
- 테스트 데이터(Test partition)
  - ✓ 예비 또는 평가 데이터 집합(holdout or evaluation partition)으로 불리며 모델 구축에 전혀 사용되지 않은 새로운 데이터 → 모델의 성능 평가에 사용



2.5 Predictive Power and Overfitting

[실습] Table 2.9

Creation and Use of Data Partitions

ex) 데이터 분할

```
# Split the dataset into training (60%) and validation (40%) sets
trainData= housing_df.sample(frac=0.6, random_state=1)
# assign rows that are not already in the training set, into validation
validData = housing_df.drop(trainData.index)
```

Training : (3481, 15) Validation : (2321, 15)

```
# randomly sample 50% of the row IDs for training
trainData = housing_df.sample(frac=0.5, random_state=1)
# sample 30% of the row IDs into the validation set, drawing only from records
# not already in the training set; 60% of 50% is 30%
validData = housing_df.drop(trainData.index).sample(frac=0.6, random_state=1)
# the remaining 20% rows serve as test
testData = housing_df.drop(trainData.index).drop(validData.index)
```

Training : (2901, 15) Validation : (1741, 15) Test : (1160, 15)

- Cross-Validation(교차 검증)
  - ✓ 데이터의 양이 적을 경우 분할 방식이 적합하지 않음
  - ✓ 전체 데이터를 5개(k=5)의 중첩되지 않은 폴드(fold)로 나누어 각 폴드는 20%의 데이터를 담고 있게 분할
  - ✓ k-1 부분: 모델 구축 / 나머지 부분: 검증 활용
  - ✓ 이 작업을 k번 수행, 최종적으로 k개의 검증 부분으로부터 구한 예측값을 평균하여 모델의 성능 평가

2.6 Building a Predictive Model

[실습] Table 2.11

Step 1. Determine the purpose

- 웨스트 록스베리 지역 주택가격 예측

Step 2. Obtain the Data

- 2014년 웨스트 록스베리 지역 주택 데이터 사용
- 데이터셋이 작아 샘플링 필요 없음

Step 3. Explore, clean, and preprocess the data

- TAX 변수 날려야함.
  - ✓ 세금 변수의 경우, 주택가격을 알아야 알 수 있으므로, 미래의 주택 가격을 예측하기 위해서는 세금 변수를 사용할 수 없다.
- 이상치(outlier)
  - ✓ FLOOR 변수의 모든 다른 값들이 1과 2 사이의 값이므로, 15는 잘못된 것으로 판단
- 범주형 변수(REMODEL)에 대해 더미변수 생성(최근/오래전/안 함)

FLOORS	ROOMS
15	8
2	10
1.5	6
1	6

TOTAL VALUE	TAX	LOT SQFT	YR BUILT	GROSS AREA	LIVING AREA	FLOORS	ROOMS	BEDROOMS	FULL BATH	HALF BATH	KITCHEN	FIREPLACE	REMODEL
344.2	4330	9965	1880	2436	1352	2	6	3	1	1	1	0	None
412.6	5190	6590	1945	3108	1976	2	10	4	2	1	1	0	Recent
330.1	4152	7500	1890	2294	1371	2	8	4	1	1	1	0	None
498.6	6272	13773	1957	5032	2608	1	9	5	1	1	1	1	None
331.5	4170	5000	1910	2370	1438	2	7	3	2	0	1	0	None

2.6 Building a Predictive Model

[실습] Table 2.11

Step 4. Reduce the data dimension

- 12개의 변수밖에 없으므로 차원축소의 필요성 없음
- PCA(주성분 분석)을 통해 상관성이 높은 다음 변수들을 적은 수의 변수로 축약 가능(e.g. LIVING AREA, ROOMS, BEDROOMS, BATH, HALF BATH)

Step 5. Determine the data mining task

- Task: 13개의 변수를 가지고 웨스트 록스베리 지역 주택 가격 예측(지도학습)
- 사용 변수: TAX를 제외한 모든 연속형 변수와 범주형 변수인 REMODEL

TOTAL_VALUE	TAX	LOT_SQFT	YR_BUILT	GROSS_AREA	LIVING_AREA	FLOORS	ROOMS	BEDROOMS	FULL_BATH	HALF_BATH	KITCHEN	FIREPLACE	REMODEL_Old	REMODEL_Recent
344.2	4330	9965	1880	2436	1352	2.0	6	3	1	1	1	0	0	0
412.6	5190	6590	1945	3108	1976	2.0	10	4	2	1	1	0	0	1
330.1	4152	7500	1890	2294	1371	2.0	8	4	1	1	1	0	0	0

2.6 Building a Predictive Model

[실습] Table 2.11

Step 6. Partition the data(for supervised task)

```
# randomly sample 50% of the row IDs for training
trainData = housing_df.sample(frac=0.5, random_state=1)
# sample 30% of the row IDs into the validation set, drawing only from records
# not already in the training set; 60% of 50% is 30%
validData = housing_df.drop(trainData.index).sample(frac=0.6, random_state=1)
# the remaining 20% rows serve as test
testData = housing_df.drop(trainData.index).drop(validData.index)
```

```
Training : (2901, 15)
Validation : (1741, 15)
Test : (1160, 15)
```

Step 7. Choose the technique

- 다중회귀분석(Multiple linear regression)

```
model = LinearRegression()
model.fit(train_X, train_y)
```

2.6 Building a Predictive Model

[실습] Table 2.11, 13

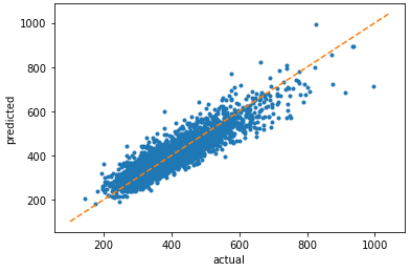
Step 8. Use the algorithm to perform the task

- 잔차(residuals): 실제값과 예측값의 차이 오차
- 예측값(Predicted value): Fitted value로 불리기도 함

- 평균오차(Mean Error): 잔차들의 단순 평균값
- 평균 제곱근 오차(Root Mean Squared Error): 평균제곱 오차값의 제곱근
- Training과 Validation set의 RMSE 값이 유사
  → 모델이 과적합되지 않음을 시사함

[Predicted value in training data]

	TOTAL_VALUE	predicted	residual
2024	392.0	387.726258	4.273742
5140	476.3	430.785540	45.514460
5259	367.4	384.042952	-16.642952
421	350.3	369.005551	-18.705551
1401	348.1	314.725722	33.374278

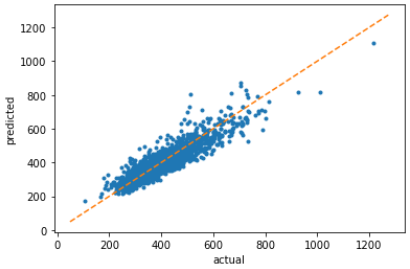


Regression statistics

Mean Error (ME)	-0.0000
Root Mean Squared Error (RMSE)	43.0306
Mean Absolute Error (MAE)	32.6042
Mean Percentage Error (MPE)	-1.1116
Mean Absolute Percentage Error (MAPE)	8.4886

[Predicted value in validation data]

	TOTAL_VALUE	predicted	residual
1822	462.0	406.946377	55.053623
1998	370.4	362.888928	7.511072
5126	407.4	390.287208	17.112792
808	316.1	382.470203	-66.370203
4034	393.2	434.334998	-41.134998



Regression statistics

Mean Error (ME)	-0.1463
Root Mean Squared Error (RMSE)	42.7292
Mean Absolute Error (MAE)	31.9663
Mean Percentage Error (MPE)	-1.0884
Mean Absolute Percentage Error (MAPE)	8.3283

2.6 Building a Predictive Model

[실습] Table 2.14

Step 9. Interpret the result

- 다른 기법과 성능 비교
- 다양한 모델에 대한 다양한 변수 선택 방법 적용

Step 10. Deploy the model

- 점수화(scoring): 새로운 데이터 값 예측

새로운 데이터 (아래)

```
new_data = pd.DataFrame({
    'LOT_SQFT': [4200, 6444, 5035],
    'YR_BUILT': [1960, 1940, 1925],
    'GROSS_AREA': [2670, 2886, 3264],
    'LIVING_AREA': [1710, 1474, 1523],
    'FLOORS': [2.0, 1.5, 1.9],
    'ROOMS': [10, 6, 6],
    'BEDROOMS': [4, 3, 2],
    'FULL_BATH': [1, 1, 1],
    'HALF_BATH': [1, 1, 0],
    'KITCHEN': [1, 1, 1],
    'FIREPLACE': [1, 1, 0],
    'REMODEL_Old': [0, 0, 0],
    'REMODEL_Recent': [0, 0, 1],
})
```

```
   LOT_SQFT  YR_BUILT  GROSS_AREA  LIVING_AREA  FLOORS  ROOMS  BEDROOMS  ₩
0      4200      1960        2670         1710     2.0     10         4
1      6444      1940        2886         1474     1.5      6         3
2      5035      1925        3264         1523     1.9      6         2

   FULL_BATH  HALF_BATH  KITCHEN  FIREPLACE  REMODEL_Old  REMODEL_Recent
0          1          1        1          1            0               0
1          1          1        1          1            0               0
2          1          0        1          0            0               1
Predictions:  [384.47210285 378.06696706 386.01773842]
```