# DM 모델적합개요

모델의 편의와 분산간 상충
모델 성능평가, 최적 모델 선택을 위한 자료 분할
전처리

## 모델의 편의와 분산간 상충 (Bias-Variance Trade-off) 때문에 과소적합과 과대적합사이 최적 적합을 탐색해야 함

■E[모델의 제곱오차] = 모델의 편의 제곱 + 모델의 분산

- 편의: 실제값과 예측값간의 차이. 편의가 크다 = 과소 적합(underfitting)
- 분산: TR내의 변동에 의한 오차. 분산이 크다 = TR내의 노이즈까지 적합= 과적합 (overfitting)
- 편의를 줄이는 방법: 모델을 복잡하게 만들어 실제값과 예측값 차이를 최소화 최근 기법들은 TR에서 잔차를 거의 0으로 만들 수 있음(편의 최소화 가능)
- 분산을 줄이는 방법: 모델을 복잡하게 만들면 편의가 줄어 TR에 너무 충실한 모형을 생성하게 됨. TR이 조금만 바뀌어도 예측값이 크게 바뀌어 변동성이 커짐. 가급적 모형을 단순화하면 분산을 줄일 수 있음
- 편의와 분산간 상충관계

| 편의 | 분산 |
|---|---|
| 실제값과 예측값간의 오차 | TR내의 변동성 때문에 발생하는 오차 |
| $B^2\left(\hat{f}(x:TR)\right) = \left[E\left[\hat{f}(x:TR)\right] - f(x;TR)\right]^2$ | $Var\left[\hat{f}(x:TR)\right]$ $= E[\hat{f}(x:TR) - E[\hat{f}(x:TR)]]^2$ |
| 편의가 크다: 실제값과 예측값간 차이가 크다. 모델 예측값이 안 맞는다<br>원인: 과소적합(TR에 너무 단순한 모델)<br>처리: 모델복잡도를 높힘 | 분산이 크다: 자료가 조금만 바뀌어도 예측값이 많이 변한다<br>원인: 과대적합(TR에 너무 충실한 모델)<br>처리: 모델복잡도를 낮춤 |
| 모델복잡도를 높여 편의를 줄임 | ⇨ 분산이 커지고 예측값 변동이 심함<br>⇨ 과대적합 |
| 과소적합 ⇦ 예측값이 안맞음 ⇦ | ⇦ 모델복잡도를 낮춰 분산을 줄임 |

- 초모수 튜닝: 모델의 편의와 분산간 최적 타협점을 탐색하는 과정
  모델의 제곱오차 기대값이 최소화되는 최적 초모수값 탐색과정
  모델 자신이 추정할 수 없는 모수이므로 자료를 이용하여(분할, 재표본 등)통해 최적값을 탐색해야 함

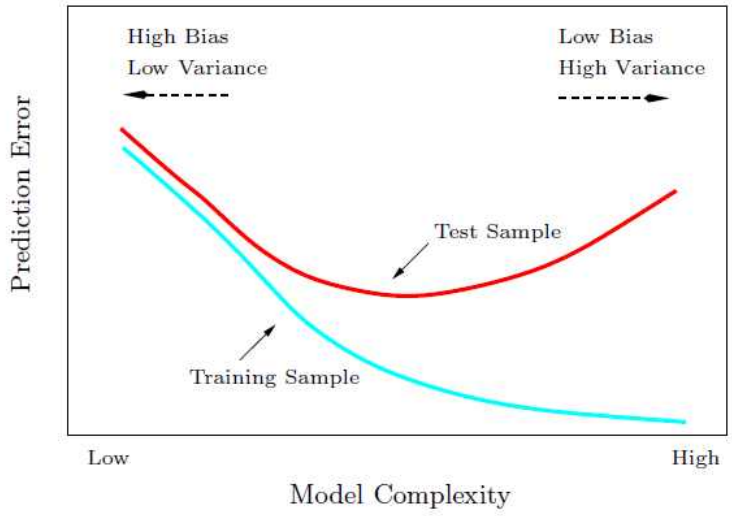



FIGURE 2.11. *Test and training error as a function of model complexity.*

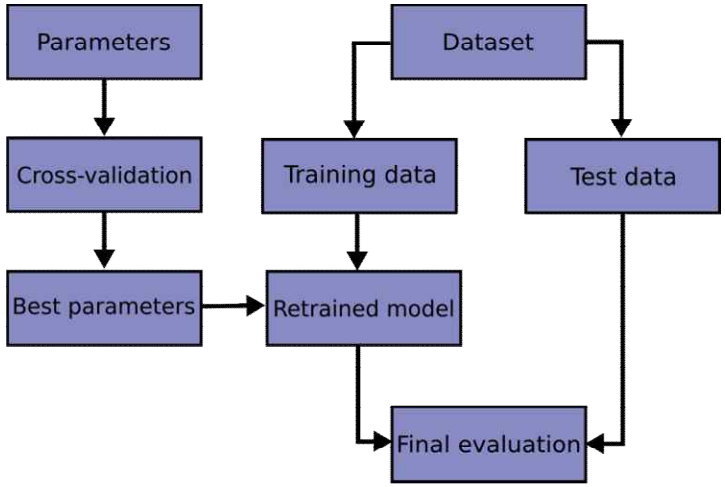
# 모델 성능평가 및 최적 모델 선택을 위한 자료 분할 (DM-DataSplitting.pptx)

■활용 목적에 따른 자료 구분

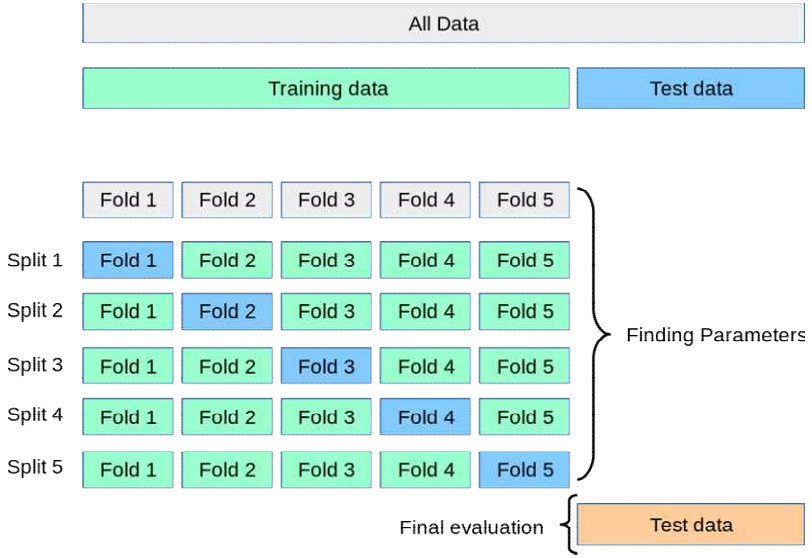
- 적합용 자료(TR, training set, 훈련용 자료): 모델 적합에 사용되는 자료
- 검정용 자료(TS, test set): 적합된 모델의 성능을 평가하는 별도의 자료
- 평가용 자료(VL, validation set): 모델의 초모수를 결정하는데 사용되는 자료

■자료 분할의 목적

- 모델의 객관적 평가: 적합용과 분리된 별도의 검정용 자료로 성능을 평가해야 함
  - 정보 누수(information leak) 주의: 모델 개발과정과 검정용 자료는 완전 분리시켜서 모델에 검정용 자료의 정보가 활용되지 않도록 해야 함
- 모수 튜닝: 모델 복잡도 등을 결정하는 초모수(hyperparameter) 최적값 결정
  - 자료가 충분하면 별도의 VL 를 활용
  - 최근에는 별도의 VL 없이 재표본 방법(교차타당성이나 붓스트랩 표본)을 이용하여 모수를 결정함. 자료 낭비 방지 및 자료 재활용 극대화
  - 과소적합이나 과대적합 방지




| 방법 | 특징 |
|---|---|
| TR/VL/TS 분할 | 선택편의(selection bias) 위험<br>VL을 별도 확보(자료 손실 큼)SAS 4:3:3 분할 |
| TR/TS 분할 | 선택편의 위험 (1회 분할)<br>자료 활용 약함<br>별도 VL이 없음<br>TR을 재표본방법으로 TR/VL로 활용 |
| 반복 TR/TS 분할<br>repeated TR/TS split<br>Monte Carlo CV | TR/TS 분할을 r번 반복<br>선택편의 위험 감소 (TS가 r개 생성됨)<br>안정적인 성능평가<br>모수 튜닝에 활용 가능 |
| k겹 교차타당성<br>k-fold CV | TR을 k개의 폴드(동일크기)로 분할<br>(k-1)개의 폴드는 TR, 1개 폴드는 VL로 활용<br>별도 VL없이 모수 튜닝할 때 사용 |
| 반복 k겹 교차타당성<br>repeated k-fold CV | k겹 교차타당성을 r번 반복<br>안정적인 모수 튜닝<br>계산 시간 오래 걸림 |
| 붓스트랩 표본<br>bootstrap sampling | BS(TR과 크기가 같은 복원추출표본)를 B개 생성<br>TR의 약 63.2%가 BS에 포함됨<br>OOB(Out-Of-Bag): BS에 빠진 자료(36.8%)<br>OOB를 VL로 활용 가능. 모수 튜닝에 활용 가능 |



## 소개 : 과적합, 모델조정, 성능평가 (r4pda-Ch10-apm Ch4 Overfitting and Model Tuning.pptx)

■과적합 (Overfitting, 과대적합)

- 최근 기법들은 자료내 복잡한 관계를 포착할 수 있도록 정교화 됨(모델 용량 확대)
- 최근기법은 자료내 패턴을 과장 또는 자료에 없는 패턴을 허위로 제시할 수 있음
- 과적합: 모델은 일반화 능력이 떨어져 새 자료에 대한 예측 신뢰도를 저하시킴
- 목표: 과적합하지 않는 모델을 선택하고, 그 모델의 성능을 객관적으로 평가 (tuning model parameters and evaluating model performance)

■성능평가의 2가지 측면 : 모델 튜닝과 모델비교

- 모델 튜닝: 특정 모델이 최적 성능을 갖도록 모델의 초모수를 튜닝(평가기준필요)
- 모델 비교: 여러 모델의 성능을 비교하여 최종모델을 선택

■모델 튜닝

- 초모수 (hyperparameter): 적합과정에서 값을 결정할 수 없는 모수
  - 주로 모델의 복잡도를 결정. 초모수값에 따라 모델의 형태가 바뀜. 주어진 자료를 이용하여 초모수의 최적값을 결정해야 함.
  - 예: KNN의 K, SVM의 h, 로지스틱의 컷오프 등
- (타당성) 평가용 자료(validation set): 초모수 결정을 위해 확보한 별도 자료. 최근에는 별도의 VL을 확보하지 않고 재표본방법으로 튜닝하는 추세임

■ 성능 평가

- 다수의 튜닝된 최적모델들의 성능을 객관적으로 평가하여 최종모델 확정
- 검정용 자료(test set): 성능 평가하려면 TR이 아닌 별도의 자료 필요
- 명백한 오류율 (apparent error rate): TR에서의 오류율(성능)
  TR에서 모델을 개발하고, 그 TR로 성능을 평가하면 개발된 모델의 성능을 과대평가하게 됨 (지나치게 낙관적, too optimistic)
- TS가 적으면 객관적인 성능 평가가 어려움 . 반드시 TS를 확보해서 성능평가할 것
- TS가 없으면 TR을 재활용해서 모델튜닝/성능평가해야 함

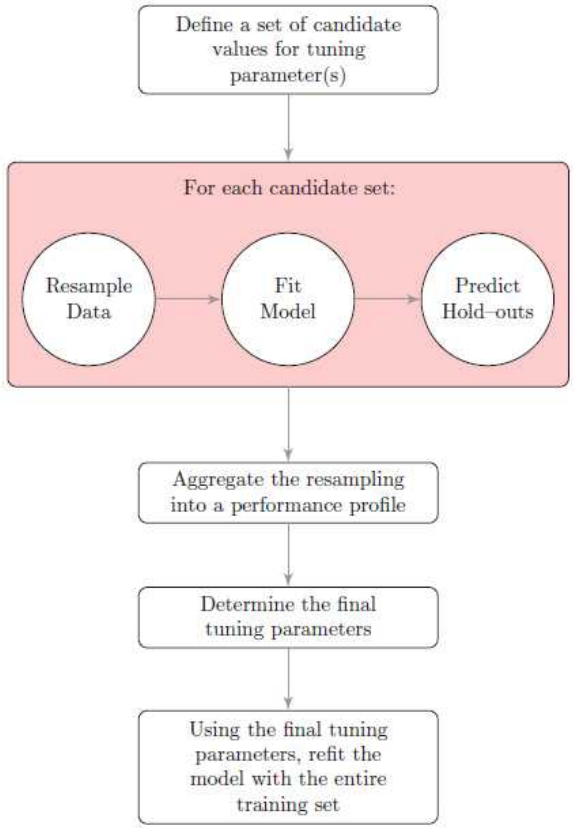



Fig. 4.4: A schematic of the parameter tuning process. An example of a candidate set of tuning parameter values for $K$-nearest neighbors might be odd numbers between 1 and 9. For each of these values, the data would be resampled multiple times to assess model performance for each value

```
Define sets of model parameter values to evaluate
for each parameter set do
    for each resampling iteration do
        Hold–out specific samples
        [Optional] Pre–process the data
        Fit the model on the remainder
        Predict the hold–out samples
    end
    Calculate the average performance across hold–out predictions
end
Determine the optimal parameter set
Fit the final model to all the training data using the optimal parameter set
```

## 자료 분할 (Data Splitting, Data Partition)

■모델 적합의 단계

- 자료 전처리 Pre-processing the predictor data
- 모델 모수의 추정 Estimating model parameters
- 모델 결정 Selecting predictors for the model
- 성능 평가 Evaluating model performance
- 최종 모수 조정 Fine tuning class prediction rules (via ROC etc)

■용도에 따른 자료의 구분

- 적합용 자료(TR, training set, 훈련용 자료): 모델 적합에 사용되는 자료
- 검정용 자료(TS, test set): 적합된 모델의 성능을 평가하는 별도의 자료
- 평가용 자료(VL, validation set): 모델의 초모수를 결정하는데 사용되는 자료

■ 자료의 분할 1: 단순 자료 분할 (랜덤 분할)

- 자료가 많으면 자료를 랜덤하게 TR, VL, TS로 분할하여 사용
  - SAS의 [Data Partition]: 4:3:3으로 랜덤 분할
- 랜덤 분할하므로 TR, VL, TS가 서로 동질적이라고 할 수 있음
- 단점:
  일회성: 어떻게 분할되느냐 따라 결과가 상이. 성능 평가의 분산 평가 불가
  자료 손실: VL, TS가 분석에서 제외됨. 전체 자료가 적으면 적용 불가.
  불균형 자료: 이산형 타겟 분포가 불균형하면 적용 어려움. 예) 사기적발 (1/10만)

■자료의 분할 2: 타겟에 대해 층화 추출 (stratified on target levels)

- 사기 적발 같이 타겟이 불균형한 분포를 가지면 랜덤 분할시 TR과 TS의 타겟 분포가 심하게 다를 수 있음. 예) 남자100, 여자 100,000 (1:1000)
- 이산형 타겟 수준별 층화 랜덤 분할. TR, TS 에서 모두 남녀 비율1:1000로 유지되도록 층화 랜덤 분할 (random sampling within subgroups in target)
- 연속형 타겟: 타겟을 임시로 범주화하고 범주별로 층화 랜덤분할 예) 키를 {작음, 보통, 큼}

■자료의 분할 3: 입력변수를 고려한 분할

- 분할시 입력변수들 값이 특정 영역으로 쏠리지 않도록 해야 함
- 가급적 모든 입력변수들 값이 골고루 TR과 TS에 분포하는 것이 바람직
  예) 운동기구 구매여부 예측시 TR에는 모두 비만, TS 모두 정상체중 고객만 할당되면 결과를 신뢰할 수 없음
- 최대 비유사성 표집(Maximum dissimilarity sampling): 다양한 관측값 (;비유사한 관측값)이 TS에 포함되도록 추출
  타겟 수준별로
  1. 임의의 관측값 xi 를 TS에 할당
  2. xi과 다른 모든 관측값의 거리를 계산하고 가장 거리가 먼 xj를 TS에 할당
  3. {xi, xj}와 다른 모든 관측값의 (군집)거리가 가장 거리가 먼 xk를 TS에 할당
  4. 원하는 TS 크기가 될 때까지 반복

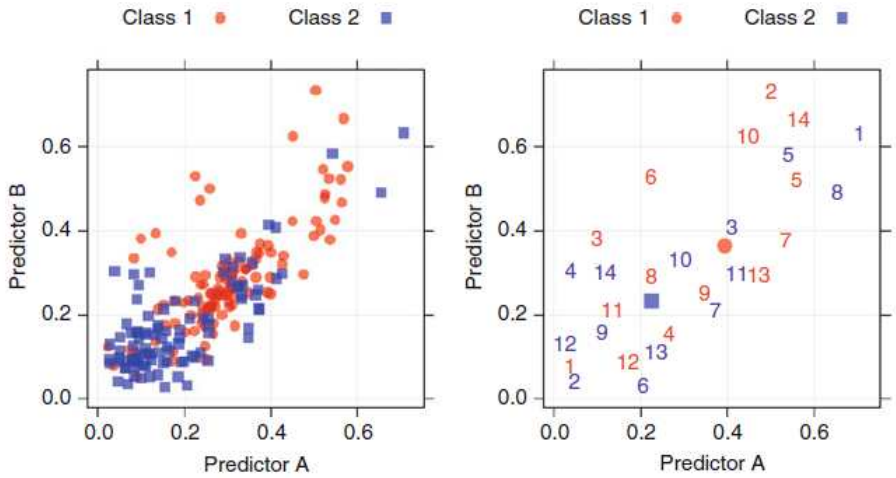
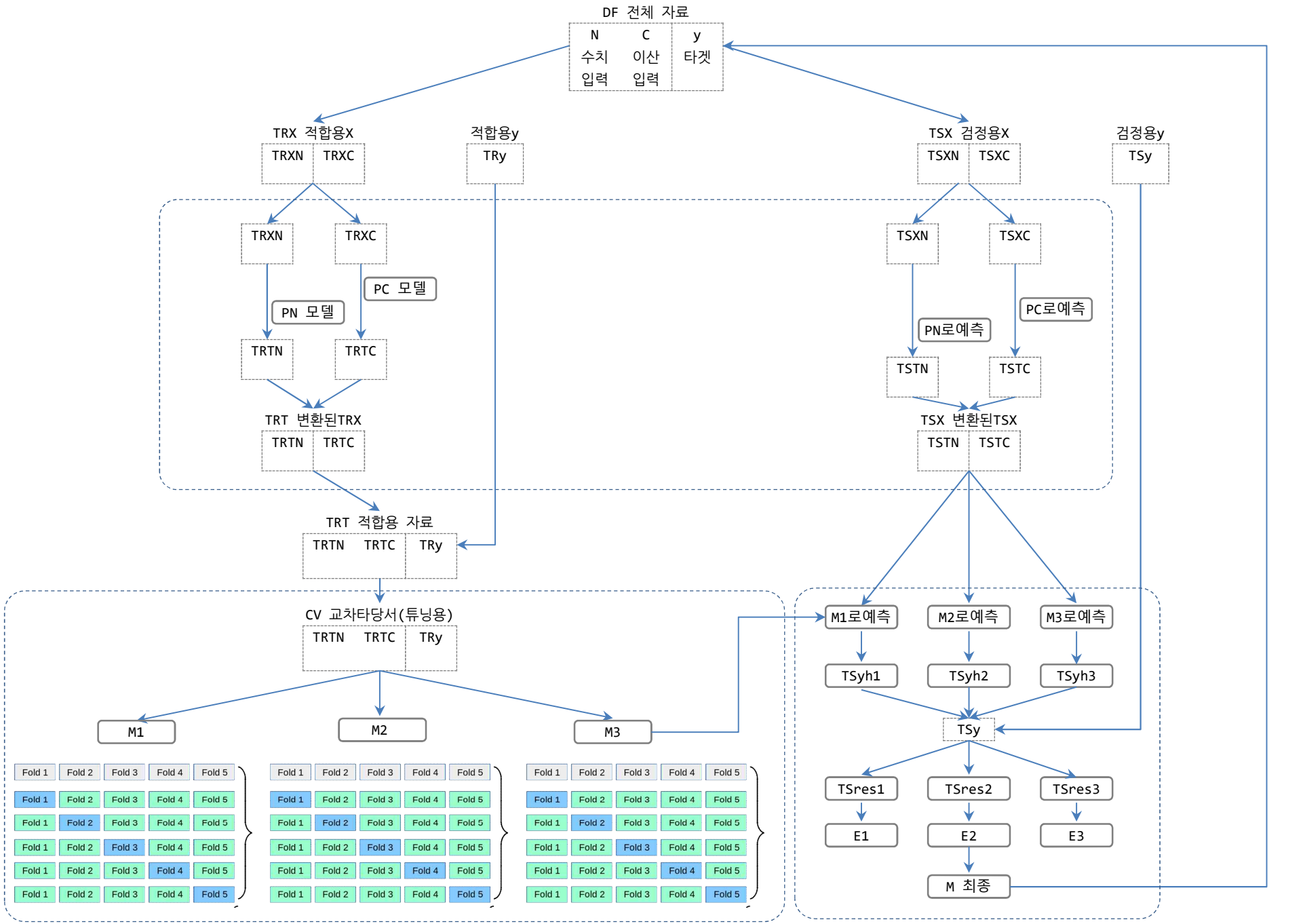



Fig. 4.5: An example of maximum dissimilarity sampling to create a test set. After choosing an initial sample within a class, 14 more samples were added

# 모델개발 개요: SEMMA (Sampling, Explore, Modify, Model, Assess)



## 전처리 (13-10-4-RF-01 Preprocess.pptx)

■전처리

- 전처리(pre-processing): 모형 구축하기 전에 모형 성능을 높이기 자료처리
- 데이터 정제(cleaning), 표준화, 변수변환, 변수추출, 변수선택 등 포함
- caret의 preProcess., recipe의 step_xxx, sklearn의 Pipeline

■전처리 주요내용

- 영분산과 거의 영분산 입력변수의 처리
- 상관된 입력변수 식별: 중복 변수 제거
- 입력변수의 변환: 수학변환(로그, 제곱근 등), 정규성 멱변환, 표준화, 숫자입력 이산화 등
- 기타 전처리

■유의사항

- 자료를 TR, TS로 분할한 뒤 TR을 전처리하고, 동일한 방법으로 TS를 전처리함(정보누수, information leakage 방지)
- 입력변수의 측도에 따라 전처리 방법이 다름
- 입력변수만 전처리함.
- 타겟은 전처리하지 않음 (타겟을 변환해야 한다면 TR,TS 분할전에 할 것)

■패키지별 처리 방식

- 1단계: 전처리 객체 생성
- 2단계: 전처리 객체 적합 (TR을 이용하여 전처리에 필요한 모수값 추정)
- 3단계: 전처리 객체를 적용한 자료 생성 (전처리된 TR, 전처리된 TS 등)
- 예: caret 전처리 방식
  - 1단계 (전처리 객체 생성): pR <- preProcess(TRX, method=..)
  - 2단계(전처리 객체 적합): TRXX <- predict(pR, newdata=TRX)로 사전모수추정
  - 3단계(전처리 객체를 적용한 자료 생성): TSXX <- predict(pR, newdata=TSX)

■전처리 1: 영분산, 거의 영분산 입력 (Zero variance, Near zero variance input)

- 한 개 값만 가지는 변수(영분산 변수) : 변수의 분산이 0. 정보 가치 없음. 제거
- 거의 영분산 변수(Near-Zero Variance input): 가지는 값이 대부분 동일하여 한 값에 쏠려 있는 변수. 분산이 거의 0인 변수
- 영분산, 거의 영분산 변수는 모형 적합을 불안정할 가능성이 높음
- 모수 튜닝을 위한 재표집(교차타당성/붓스트랩)시 거의 영분산이 될 수 있음(튜닝에 과도한 영향을 미칠 수 있음)
- 처리: 거의 영분산 변수는 사전에 파악하여 제거
- 영분산 변수의 식별
  - 빈도비율(FreqRatio)=가장 흔한 값 빈도/두 번째 흔한 값 빈도
    일반적으로 1에 가까우면 안정적.
    매우 불균형이면 큰 값
    예: 남/여=99/1>>1. 성별이 남에 완전히 치우친 상태
    (caret 기준) freqCut=95/5보다 큰 변수 제거
  - 유일값 비율(PercentUnique)=유일값들의 종류수/표본수 x 100 (%)
    다양한 값을 가질수록 100%에 가까워짐. 예) 100명 모두 다른 키면, 100/100
    동일한 값을 가질수록 0%에 가까워짐 예) 100명 모두 같은 키면, 1/100
    (caret 기준) uniqueCut=10%
  - 빈도비율 >freqCut=95/5 AND 유일값 비율<10% 인 변수는 거의 영분산으로 간주 (두 기준 모두 충족하는 변수 제거 권장)
- 예제: caret::mdrr에서 nearZeroVar() 사용하여 거의 영분산 변수 식별
  - saveMetrics= 인자는 세부 정보를 제공하며, 디폴트는 FALSE 임)
  - nearZeroVar() 호출시 saveMetrics=T 지정시 각 변수별 빈도비율과 유일값 비율, zeroVar와 nzv에 대한 논리값 제공
  - (기본값) 유일값비율이 10% 이하이고 빈도비율이 19(95/5)보다 큰 입력변수는 거의 영분산으로 간주

■상관이 높은  입력변수의 식별: 중복 변수의 제거

- 일부 모형은 상관이 높은 입력변수들이 동시에 포함되면 불안정해짐(다중공선성)
- 서로 상관이 높은 입력변수가 있다는 것은 정보가 중복된다는 의미. 중복의 소지가 있는  변수는 제거하여 모형을 안정화/단순화함
- caret::findCorrelation(), caret::FindLinearCombs():  QR분해로 선형종속 포착

■입력변수 변환 1: 표준화(= 중심화 + 척도화), 범위정규화

- 입력변수들의 분산이 비슷할 때 적합이 안정적임. 신경망, 군집분석에서 권장
- 많이 쓰이는 변환: 중심화(centering), 척도화(scaling), 표준화(standardization, 범위 정규화(range normalization)
- 예제: caret:mdrr

■입력변수 변환 2: 대칭성 확보를 위한 변환

- 분포가 한쪽으로 쏠린 입력변수는 적합을 불안정하게 할 수 있음
- BoxCox, YeoJohnson 변환: 원래 타겟의 정규성 확보를 위한 최적 멱변환 방법이지만 caret에서는 입력변수의 대칭성(또는 정규성)을 확보하기 위해 적용

■결측값 대체 (imputation)

- 자료내 결측값을 찾아 대체.  자료의 낭비 방지

| | 단순 대체: 특정값 대체 | |
|---|---|---|
| | 숫자 | 명목 |
| 단일값 대체 | medianImpute 중위수 대체 | modeImpute 최빈값 대체 |
| | meanImpute 평균 대체 | |
| 단일값 대체 -모형 대체 | knnImpute: 결측이 있는 관측치에 대하여 k개의 가장 가까운 이웃을 찾아 이들 값의 평균으로 결측값을 대체<br>단순 대체보다 정교하고, 계산비용도 낮은 편<br>참고: knnImpute시 모든 입력은 표준화됨. 숫자형 입력만 가능 | |
| | bagImpute: 결측을 bagged 트리로 예측하여 대체. 각 입력변수에 대해 다른 모든 입력변수로 bagged tree를 만들어 예측값으로 대체<br>단순 대체보다 정교하고, 숫자형, 명목형 모두 가능. 계산 부담 높음 | |
| 복수값 대체 | 한 개의 결측에 대해 다수의 대체값 제공. 계산 부담 매우 높음 | |

■가변수 생성: OneHotEncoder, One-of-K

- 요인(명목형 입력변수)을 모형에 포함시키는 방법.
- 순서형 : 수준순서에 따라 숫자를 할당하고 연속형으로 취급(OrdinalEncoder)
- 명목형 :

| | Less-than-full Model overparameterization | Full model |
|---|---|---|
| 처리 | 수준수 만큼 가변수 생성 | 수준수-1개의 가변수<br>참조수준: 첫 번째 수준 |
| sklearn | **OneHotEncoder(drop=None)** | OneHotEncoder(drop='first') |
| caret | **DummyVars(~, fullRank=F)** | DummyVars(~, fullRank=T) |
| recipes | step_dummy(.., one_hot=T) | **step_dummy(.., one_hot=F)** |

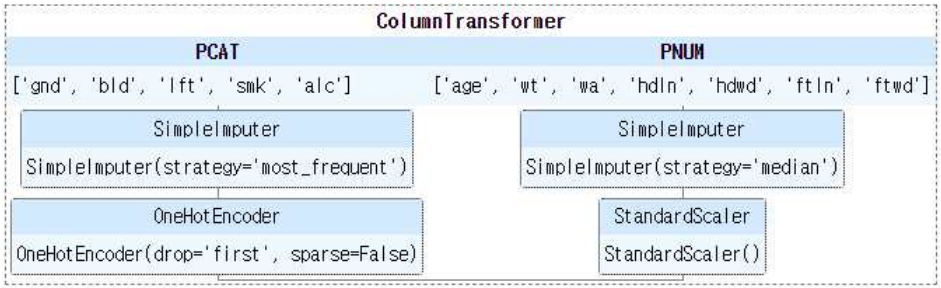
## recipes::step_XXX vs preProcess ( method=c('zv', 'nzv', 'corr', 'YeoJohnson', 'center', 'scale', 'range', 'bagImpute','pca', 'SpatialSign'))

| 적용 순서 | 처리 | recipes:step_xxx | preprocess 옵션 |
|---|---|---|---|
| 영분산 | save=T로 {zeroVar, nzv}를 생성해서 영분산 확인 | nearZeroVar(x, freqCut=95/5, uniqueCut=10, save=F,..)<br>RC %>% step_zv(SEL..) | 'zv',<br>nearZeroVar |
| | 타겟 수준별 nzv변수 제거. naïve Bayes모형시 유용 | checkConditionalX(x, y) | 'conditionalX' |
| 거의 영분산 | save=F면 거의 영분산 변수 인덱스 반환 | nearZeroVar(x, freqCut=95/5, uniqueCut=10, save=F,..)<br>RC %>% step_nzv(SEL.., freq_cut=95/5, unique_cut=10,..) | 'nzv', nearZeroVar |
| 높은 상관<br>공선성 방지 | 상관계수 절대값의 평균이 cutoff 보다 높은 변수 제거 | findCorrelation(x, cutoff=0.9, ..)<br>RC %>% step_corr(SEL.., threshold=0.9,..) | 'corr'<br>findCorrelation |
| 정규성 멱변환<br>최적 멱변환 | YeoJohnson: Box Cox 개선. 모든 숫자 변수 적용가능 | RC %>% step_YeoJohnson(SEL..) | 'YeoJohnson' |
| | Box Cox: 양의 숫자변수만 가능 | RC %>% step_BoxCox(SEL..) | 'BoxCox' |
| 표준화 | 평균중심화 (x-mean(x)) | RC %>% step_center(SEL..) | 'center' |
| | 분산척도화 (x/sd(x)) | RC %>% step_scale(SEL..) | 'scale' |
| | 표준화 | RC %>% step_normalize(SEL..) | |
| 범위 정규화 | 범위정규화 (x – min(x))/range(x) | RC %>% step_range(SEL.., min=0, max=1,..) | 'range' |
| 대체(숫자입력) | 중위수 대체. 부정확, 결측 허용, 신속 | RC %>% step_medianimpute(SEL..) | 'medianImpute' |
| | 평균 대체. 부정확, 결측 허용, 신속 | RC %>% step_meanimpute(SEL..) | 'meanImpute' |
| 대체(명목입력) | 최빈값 대체. 명목형만 가능 | RC %>% step_modeimpute(SEL..) | |
| 대체(모형) | knn 대체. 정확, 결측 허용, 신속, center/scale함 | RC %>% step_knnimpute(SEL.., neighbors=5, ..) | 'knnImpute' |
| | bagged tree 대체(xj ~ .-xj). 정확, 결측 허용, 느림 | RC %>% step_bagimpute(SEL.., trees=25, ..) | 'bagImpute' |
| 차원축소 | 주성분분석. 누적 95%이상 추출. center/scale함 | RC %>% step_pca(SEL.., num_comp=5, threshold=NA, ..) | 'pca', prcomp |
| | 독립성분(첨도, 4차 적률). center/scale함 | RC %>% step_ica(SEL.., num_comp=5, ..) | 'ica', fastICA |
| 이상점 조정 | 공간부호. 입력변수를 단위원에 투사. 이상점의 영향력 축소 | RC %>% step_spatialsign(SEL..) | 'SpatialSign' |
| 가변수 생성 | 명목형 입력처리(one_hot=F면 수준수-1개의 가변수 사용) | RC %>% step_dummy(SEL.., one_hot=F, ..) | dummyVars(formula,<br>fullRank=F) |
| | step_downsample, step_upsample은<br>themis::step_downsample()로 대체 | SEL.. 방법<br>- 입력이름을 직접 지정: starts_with(), ends_with(), contains(), matches()<br>- 변수기능에 따라 지정: all_predictors(), all_outcomes()<br>- 변수측도에 따라 지정:all_numeric(), all_nominal() | |

## 패키지별 처리 방식 비교 예시

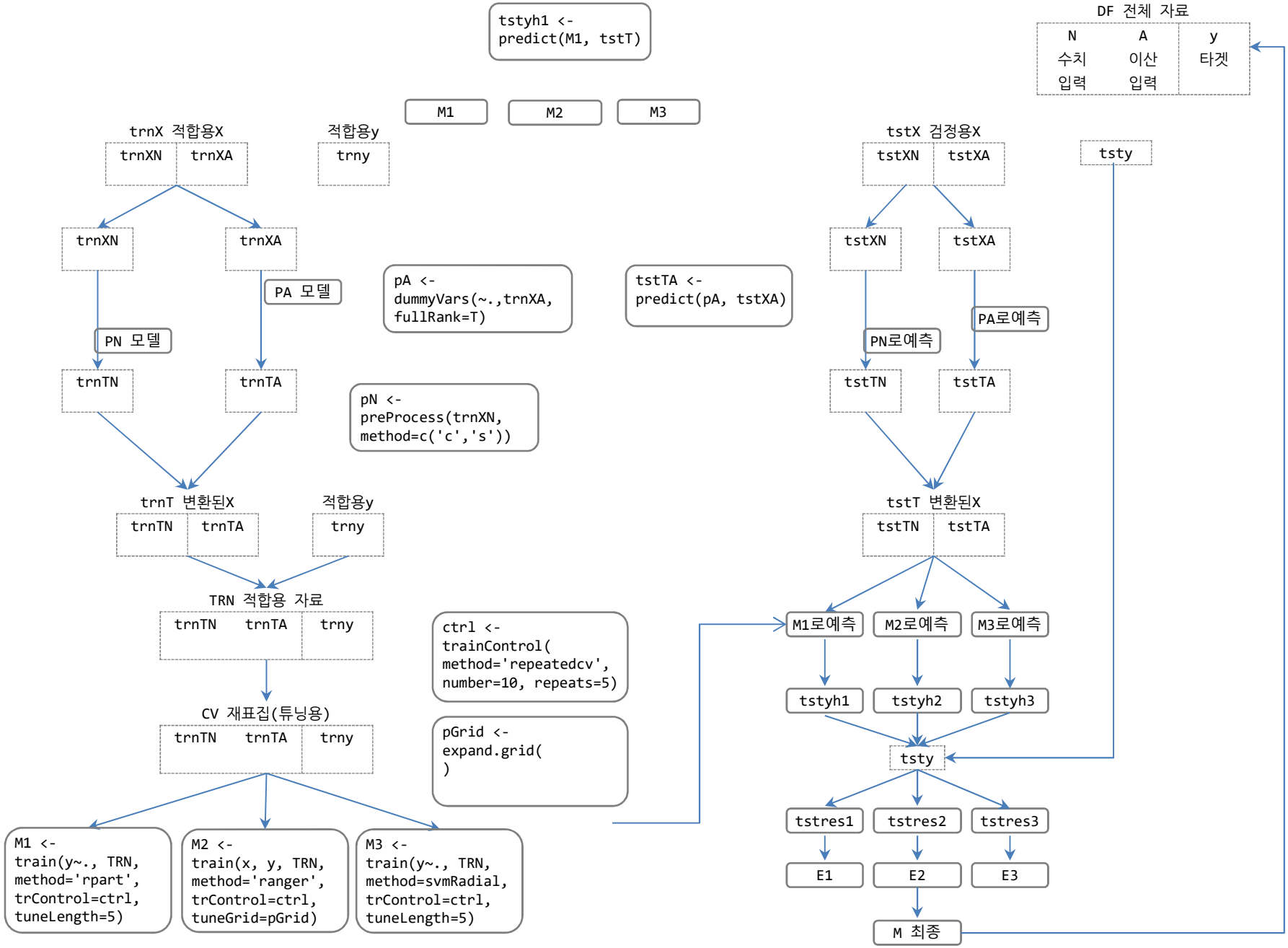
| | recipe방식 |
|---|---|
| 전처리 정의<br><br>전처리<br>사전적합 | RC <- recipe(ht~., data=TR) %>%<br>  step_modeimpute(gnd, bld) %>%<br>  step_medianimpute(all_numeric()) %>%<br>  step_normalize(all_numeric()) %>%<br>  step_dummy(all_nominal())<br><br>pRC <- prep(RC, training=TR) |
| 전처리된 TR | TRT <- bake(pRC, new_data=TR) |
| 전처리된 TS | TST <- bake(pRC, new_data=TS) |

| | caret방식 |
|---|---|
| 전처리 정의<br>전처리 사전적합<br>전처리된 TR | pZ <- preprocess(TRX,<br>  method=c('center','scale'))<br>TRXZ <- predict(pZ, newdata=TRX)<br>pD  <- dummyVars(~gnd+bld, data=TRXZ)<br>TRXZD <- predict(pD, TRXZ) |
| 전처리된 TS | TSXZD <- predict(pD, predict(pZ, newdata=TS)) |




| | sklearn방식 (요인외는 모두 숫자형으로 처리할 수 있도록 요인지정 항상 할 것) |
|---|---|
| 파이프라인<br>정의 | # NUMFEATS = ['age', 'wt', 'wa', 'hdln', 'hdwd', 'ftln', 'ftwd'] # 타겟('ht') 제외<br># CATFEATS = ['gnd', 'bld','lft', 'smk', 'alc' ]<br>CATX = list(TRX.select_dtypes('category').columns)<br>NUMX = list(TRX.select_dtypes('number').columns)<br>PCAT = Pipeline([<br>  ('simpleimputer', SimpleImputer(strategy='most_frequent')),<br>  ('dummy', OneHotEncoder(drop='first', sparse=False))<br>  ])<br>PNUM = Pipeline([<br>  ('simpleimputer', SimpleImputer(strategy='median')),<br>  ('standardscaler', StandardScaler())<br>  ])<br>P = ColumnTransformer(transformers=[('PCAT', PCAT, CATX),<br>  ('PNUM', PNUM, NUMX)]); |
| 파이프라인<br>시각화 | from sklearn import set_config<br>set_config(display='diagram')  # diplays HTML in a jupyter context<br>P |
| 전처리<br>사전적합<br><br>전처리된<br>TRX | P.fit(TRX)<br>TRXT = P.transform(TRX)  # TRXT 는 nd.array임. 변수이름 모두 사라짐<br>COLX = np.append(P.named_transformers_['PCAT']['dummy'].get_feature_names(CATX), NUMX)<br>TRXT = pd.DataFrame(TRXT, columns=COLX) |
| 전처리된<br>TSX | TSXT = P.transform(TSX)<br>TSXT = pd.DataFrame(TSXT, columns=COLX) |

# V2


tstyh1 <-
predict(M1, tstT)
DF 전체 자료
N
수치
입력
A
이산
입력
y
타겟
M1
M2
M3
trnX 적합용X
trnXN
trnXA
적합용y
trny
tstX 검정용X
tstXN
tstXA
tsty
trnXN
trnXA
tstXN
tstXA
PA 모델
pA <-
dummyVars(~.,trnXA,
fullRank=T)
tstTA <-
predict(pA, tstXA)
PA로예측
PN 모델
PN로예측
trnTN
trnTA
tstTN
tstTA
pN <-
preProcess(trnXN,
method=c('c','s'))
trnT 변환된X
trnTN
trnTA
적합용y
trny
tstT 변환된X
tstTN
tstTA
TRN 적합용 자료
trnTN
trnTA
trny
M1로예측
M2로예측
M3로예측
ctrl <-
trainControl(
method='repeatedcv',
number=10, repeats=5)
CV 재표집(튜닝용)
trnTN
trnTA
trny
tstyh1
tstyh2
tstyh3
pGrid <-
expand.grid(
)
tsty
tstres1
tstres2
tstres3
M1 <-
train(y~., TRN,
method='rpart',
trControl=ctrl,
tuneLength=5)
M2 <-
train(x, y, TRN,
method='ranger',
trControl=ctrl,
tuneGrid=pGrid)
M3 <-
train(y~., TRN,
method=svmRadial,
trControl=ctrl,
tuneLength=5)
E1
E2
E3
M 최종