Recap

09 June 2023 19:56

Linear Reg case study

Cross validation

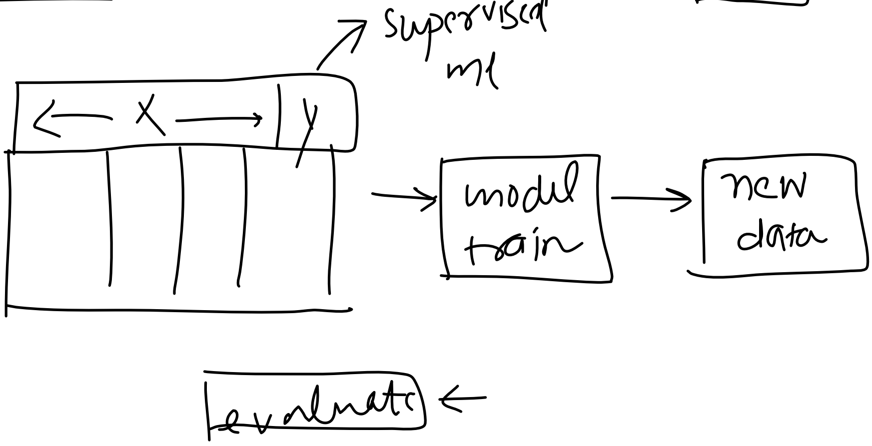
Hyper parameter

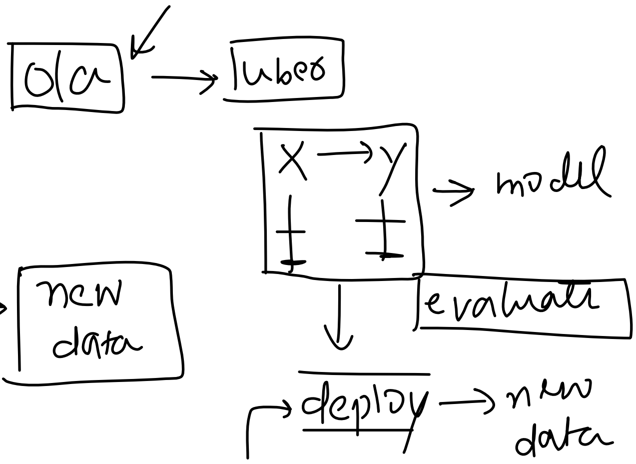
model selection

The Problem

10 June 2023 12:31

Cross val → why



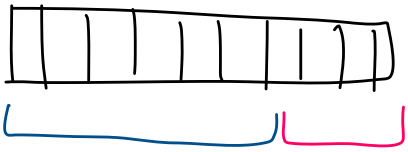


The Hold-out Approach

10 June 2023 12:31

→ train_test_split

1000 customers



0.75 0.25

training test

1) shuffle

2) train_test

train → model

y_pred

y_true

model evaluation

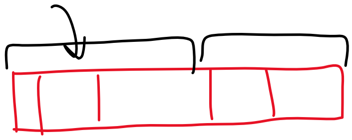
crossvalidation

Problem with Hold-out Approach

10 June 2023 12:32

more data better models

High ml



shuffle

random_state = 1, 2, 3

1. Variability: The performance of the model can be very sensitive to how the data is divided into training and testing sets. If the split is unfortunate, the training set may not be representative of the overall distribution of data, or the test set might contain unusually easy or difficult examples. This leads to high variance in the estimation of the model's performance.

2. Data inefficiency: The holdout method only uses a portion of the data for training and a different portion for testing. This means that the model doesn't get to learn from all available data, which can be particularly problematic if the dataset is small.

acc → 95

acc → 99, 91, 81

3. Bias in performance estimation: If some classes or patterns are over- or under-represented in the training set or the test set due to the random split, it can lead to a biased performance estimation. High bias ← data leakage

4-5 cities

4. Less reliable for hyperparameter tuning: If the holdout method is used for hyperparameter tuning, there's a risk of overfitting to the test set because information might leak from the test set into the model. This means that the model's performance on the test set might be overly optimistic and not representative of its performance on unseen data.

1

reliable

variance

100% 80%

80% training

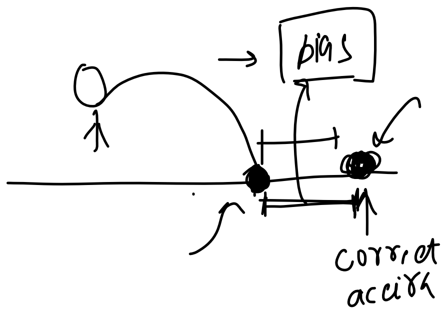
20% percent

0.72, 0.76, 0.77 best → model deploy

Bias variance trade off

80% → bias ↑↓

loss → $\boxed{\text{bias}}$ + var + noise



Why is hold-out approach used then?

10 June 2023 12:32

52 62 82 92 81.9 87.2 87.4

smaller

1. Simplicity: The holdout method is straightforward and easy to understand. You simply divide your data into two sets: a training set and a test set. This simplicity makes it appealing, especially for initial exploratory analysis or simple projects.
2. Computational Efficiency: The holdout method is computationally less intensive than methods like k-fold cross-validation. In k-fold cross-validation, you need to train and test your model k times, which can be computationally expensive, especially for large datasets or complex models. With the holdout method, you only train the model once.
3. Large Datasets: For very large datasets, even a small proportion of the data may be sufficient to form a representative test set. In these cases, the holdout method can work quite well.

→ project

→ model

→ cross-vali → multiple model

large

large dataset → high variance random

↳ diminates

1L → 10 sample

1000 samplez

pop avg height

↑ more data

1L row → random

↳ change

Cross Validation

10 June 2023 12:33

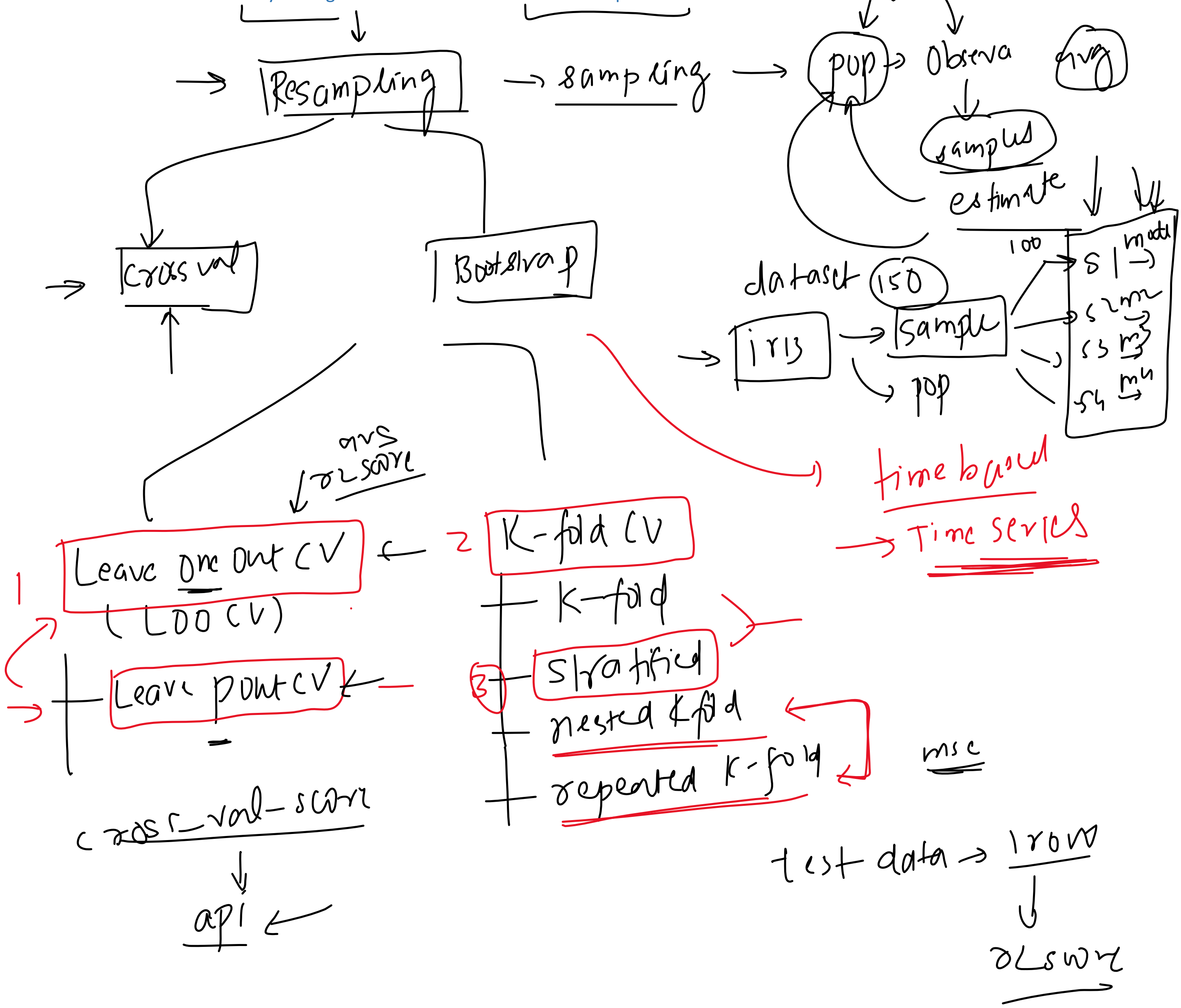
given → f1, f2, f3

test

avg sal

final

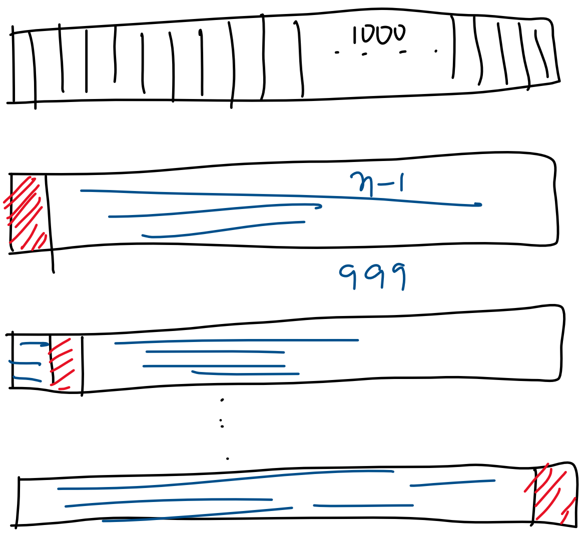
The idea of cross-validation is to divide the data into several subsets or "folds". The model is then trained on some of these subsets and tested, on the remaining ones. This process is repeated multiple times, with different subsets used for training and validation each time. The results from each round are usually averaged to estimate the model's overall performance.



Leave One Out Cross Validation (LOOCV)

10 June 2023 12:33

dataset → 1000 rows (n) → models (1000) → model evaluation



→ 999 rows → training → m1
1 row → test

999 rows → m2
1 row

m1000

LOOCV

accuracy

$m_1 + m_2 \dots + m_{1000}$ → avg → final accuracy score

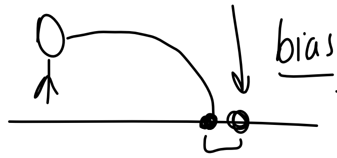
Advantages:

1. Use of Data: LOOCV uses almost all of the data for training, which can be beneficial in situations where the dataset is small and every data point is valuable.
2. Less Bias: Since each iteration of validation is performed on just one data point, LOOCV is less biased than other methods, such as k-fold cross-validation. The validation process is less dependent on the random partitioning of data.
3. No Randomness: There's no randomness in the train/test split, so the evaluation is stable, without variation in the results due to different random splits. (no shuffling)

→ LOOCV → n-1 row

80% data → high bias

$\frac{n-1}{n}$ % → higher → reduce bias



bias → reduce

Disadvantages:

1. Computational Expense: LOOCV requires fitting the model N times, which can be computationally expensive and time-consuming for large datasets.
2. High Variance: LOOCV can lead to higher variance in the model performance since the training sets in all iterations are very similar to each other.
3. Inappropriate Performance Metric: Performance metrics like R^2 are not appropriate to be used with LOOCV as they are not defined when the validation set only has one sample.
4. Not Ideal for Imbalanced Data: In classification problems, if you have imbalanced classes, LOOCV may not provide a reliable estimate of model performance because the single validation sample in each iteration may not

50000 row data → 50k models

accuracy { lot of time }

4. Not Ideal for Imbalanced Data: In classification problems, if you have imbalanced classes, LOOCV may not provide a reliable estimate of model performance because the single validation sample in each iteration may not be representative of the overall class distribution.

When to use:

1. Small datasets: LOOCV is most beneficial when you have a limited amount of data. With small datasets, you want to use as much data as possible for training to get a reliable model, which is exactly what LOOCV offers by using all but one data point for training.
2. ~~Balanced datasets~~: LOOCV might not perform well on imbalanced datasets, especially in classification problems, because the training set might end up missing some classes. Thus, it's more appropriate to use LOOCV when you have a balanced dataset.
3. Need for less biased performance estimate: Since LOOCV uses nearly all the data for training, it gives a less biased estimate of model performance compared to other methods like k-fold cross-validation.

model
var → bias
low high
LOOCV

accuracy { lot of time, lot of space }
1 data point
over
accura → variance
1000 → 990 10
Y N
990 test sets
N component
540-460
Y N

K-Fold Cross Validation

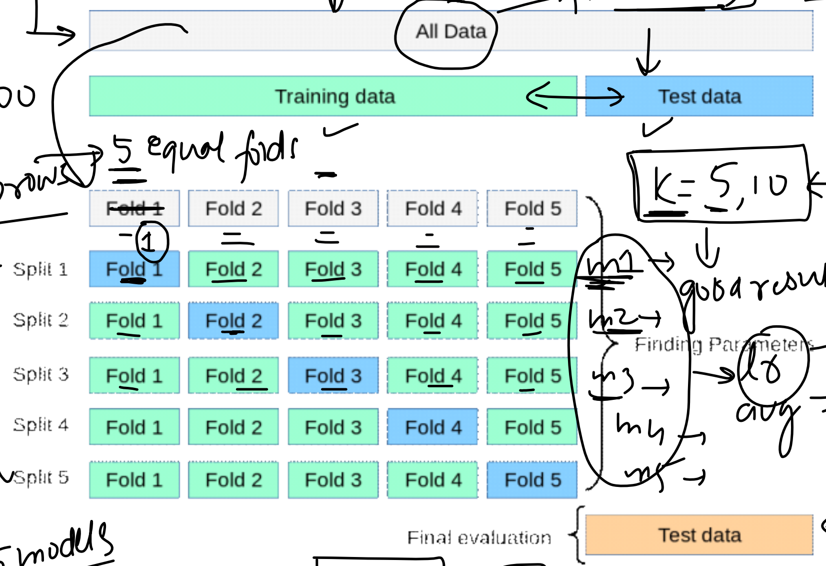
10 June 2023 12:34

→ most used cv technique general variant

pure wrcepts

Linear Reg → case study

Data leaks



100 ↓ 20 rows → 5 equal folds

K = 5, 10

m1 → m2 → m3 → m4 → m5 →

good results → model

avg → avg accr

unseen data

5 models

1 one each fold

n rows → n models → n-1 row training

1 testing

avg

→ Cross validation → 1/2

→ Hyperparameter

Data Leakage

→ Evaluate

→ Hold out set → train → test

→ problems

→ use

→ Cross validation costly

→ LOOCV

what

draw

advant

→ K fold K-1 fold train high bias

n-1 rows training → bias low

Advantages of K-Fold Cross Validation: ← K-fold → variant

1. Reduction of Variance: By averaging over k different partitions, the variance of the performance estimate is reduced. This is beneficial because it means that the performance estimate is less sensitive to the particular random partitioning of the data.

→ LOOCV → no. of rows 10000

2. Computationally Inexpensive: Take less time and space in comparison to LOOCV

→ K=5,10 → 5 models

Disadvantages of K-Fold Cross Validation:

1. Potential for High Bias: If k is too small, there could be high bias if the test set is not representative of the overall population.

2. May not work well with Imbalanced Classes: If the data has imbalanced classes, there's a risk that in the partitioning, some of the folds might not contain any samples of the minority class, which can lead to misleading performance metrics.

1 fold → multiple

r2 → 1 row → vary

1000 → 5 fold, 200 rows / fold

2. May not work well with imbalanced classes: If the data has imbalanced classes, there's a risk that in the partitioning, some of the folds might not contain any samples of the minority class, which can lead to misleading performance metrics.

↳ classification

Vanilla K fold → 95% Y 5% N → 100 row 5 fold CV

95Y 5N

multiple

1000 → 5 fold 200 rows/fold

variance

fo rows

hold-out variance

> K-fold

When to use:

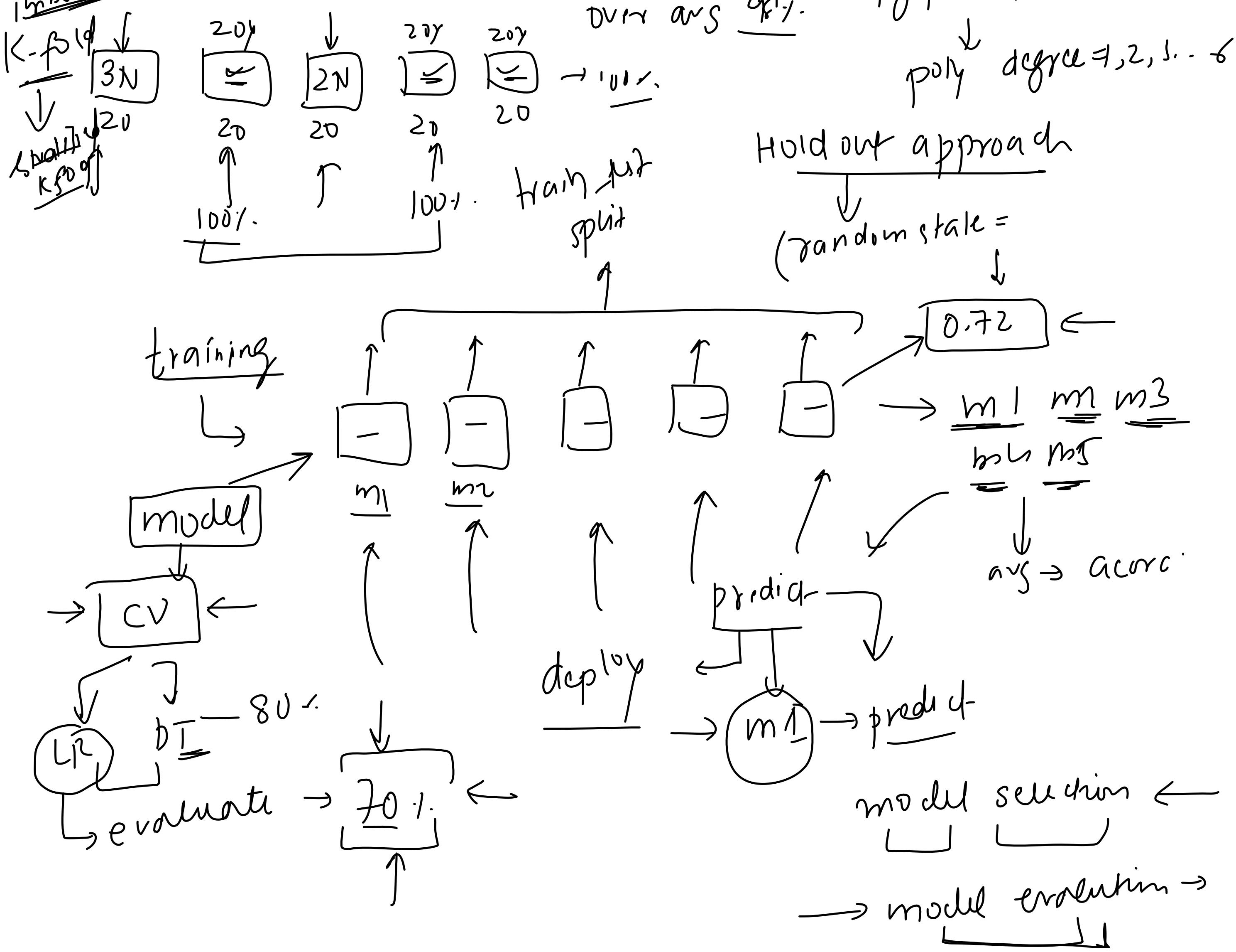
1. When you have a sufficiently large dataset: K-Fold Cross Validation requires the model to be trained K times, so it can be computationally intensive. However, if your dataset is large enough, this increased computational cost can be justified by the more reliable estimate of model performance.

2. When your data is evenly distributed: K-Fold Cross Validation works best when the data is evenly distributed. If your dataset is imbalanced (i.e., one class has significantly more samples than another), it might be better to use a technique like Stratified K-Fold Cross Validation, which aims to ensure each fold is a good representative of the whole dataset.

imbalanced

mpg | horsepower

poly degree = 1, 2, 3 . . 6



Stratified K-Fold CV

10 June 2023 12:34

95% Y 5% N

95% Y 5% N

Imbalanced

classification

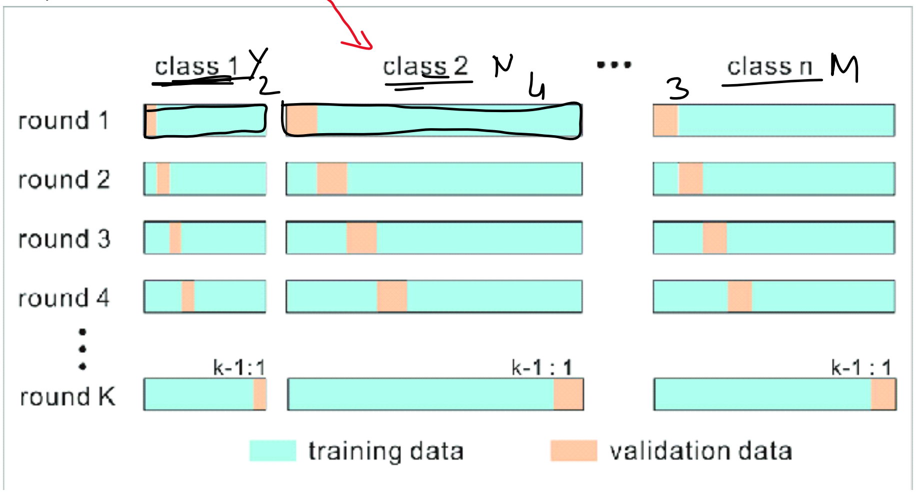
Y 95%

N 5%

K-fold

100%

avg



1, 2, 3

2 4 3

Class is not

realistic → avg → realistic value

m1 → 70% 5 num

unseen data → test data

mL → 72%

unseen data → test