TECHNOLOGY FUNDAMENTALS FOR BUSINESS ANALYTICS

Jason Kuruzovich

Agenda

- Lab 3
- Missing Data
- Recoding Data/Feature Creation
- Cross Validation
- Lab 5
- Introduction to Kaggle Scripts

Lab 3 Solutions

Missing Data

What are some reasons data might be missing?

Why might missing data be a problem?

What are some reasons data might be missing?

- Missing data can be random
  - Perhaps there is a field where people can put their income in, but it is optional
- Missing data can be linked to the missing value itself
  - Perhaps people with high or low income may be unwilling to report income data
- Missing data can be linked to other observed predictors
  - Demographics may lead people not to answer

Why might missing data be a problem?

“Missing data is a problem because nearly all standard statistical methods presume complete information for all the variables included in the analysis.”

http://www.bu.edu/sph/files/2014/05/Marina-tech-report.pdf

Missing Data Example

Happiness (DV)	Gender	Age	City	Location
9	M	30	Troy	NA
3	NA	31	Boston	Urban
2	F	23	NA	Country
5	M	NA	New York	Urban

For the above example, all records would be dropped if one did an analysis of happiness as a DV with the variety of other independent variables for common models like regression

Missing Data and Languages

- Languages have a specific way of encoding that data is missing
  - R <-NA   is.na()
  - Python (NaN or None) isnull()

How do we deal with missing data?

Simple Solutions (Ignores some data)

- Listwise deletion. Drop records from analysis with missing fields
  - Good: easy (most models will do automatically)
  - Bad: can’t generate predictions where missing, loss of much data
- Create alternate model
  - Good: easy
  - Bad: may need multiple models & there may be some information in missing data that is ignored

How do we deal with missing data?

Data Imputation

- Mean imputation. Easiest solution is to replace each missing value with the mean of the observed value.

Advanced Techniques

- Conditional mean imputation, multiple imputations, & maximum likelihood models use data of known variables to predict the appropriate variable for the missing data

How do we deal with missing data?

- Missing data can limit the ability to incorporate useful data into predictive models
- When doing scientific analysis, there are higher hurdles and one can’t do it to improve results
- In applied analytics, we can more easily try data imputation techniques if it enables us to do prediction

Recoding Data/Feature Extraction

What is a feature?

Feature Extraction in Data Mining

- We have talked about standard data types (string, integer, factor, etc.)
- However, many ways to extract/create new *features* from data
- *Features* are variables likely to be meaningful for data modeling
- *Feature extraction* and *feature selection (which features to include in model)* go together

What are we looking for in “good features”?

[When we perform feature selection, which will we select?]

Is “name” likely to be a good feature in the titanic dataset? Why?

What can we get out of the name field? (Take some time and just open the CSV in Excel)

Feature Selection

- Redundant or irrelevant features can often be disregarded from analyses
- We want encodings of the data that predict the outcome of interest

Example of Feature Extraction

- For a long time, batting average was the most common feature of interest
- Slugging percentage represented a new feature
- https://en.wikipedia.org/wiki/Slugging_percentage

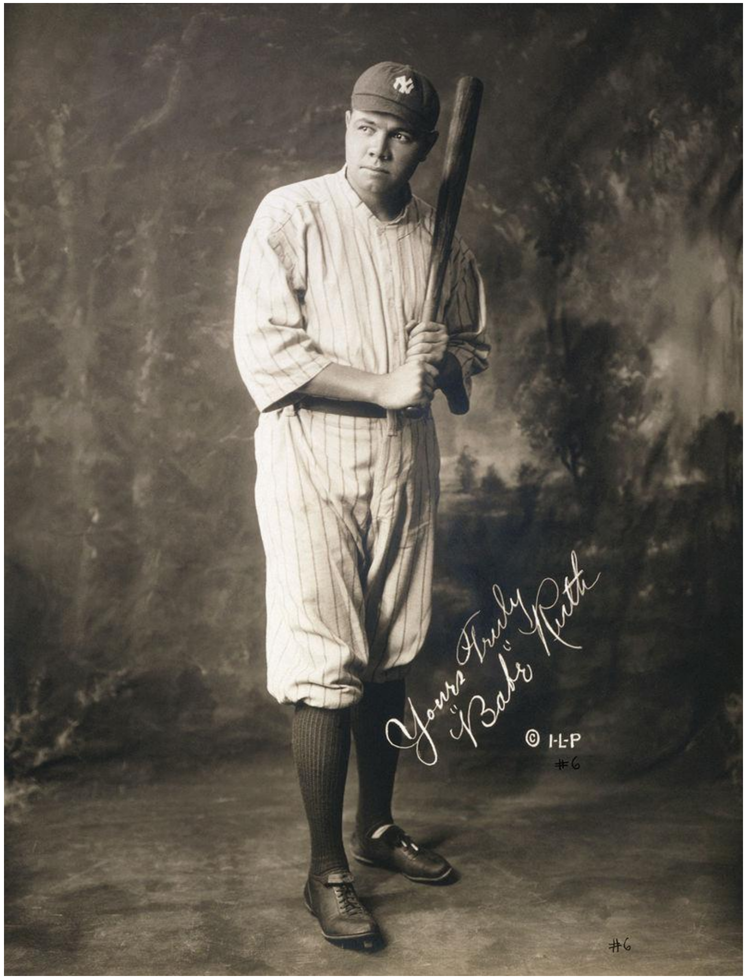


Photo: https://en.wikipedia.org/wiki/Slugging_percentage#/media/File:Babe_Ruth2.jpg

Examples of Feature Extraction

- Who are the first onto the lifeboats of the Titanic?

https://en.wikipedia.org/wiki/Women_and_children_first

Examples of Feature Extraction

- The age variable could be recoded to the following

This could be integer or.

- child = 1 if age <18
- child = 0 if age >=18

factor variable [let's call it stage]

- stage = "child" if age <18
- stage = "adult" if age >=18

Examples of Feature Extraction

- The age variable could be recoded to the following

Factor variable is more flexible to handle multiple categories

- stage = “infant” if age <=2
- stage = “child” if age >2 and age <=12”
- stage = “teen” if age >12 and <18”
- stage = “adult” if age >=18

Examples of Feature Extraction

Modeling Sales -> Imagine we want

- Date ->
  - Year (factor)
  - Month (factor)
  - Day of week (factor)
  - Weekend (binary)
  - End of month (binary)
  - Week in month (factor)

Example of Feature Extraction

For a long time, batting average was the most common feature of interest

Slugging percentage represented a new feature

https://en.wikipedia.org/wiki/Slugging_percentage

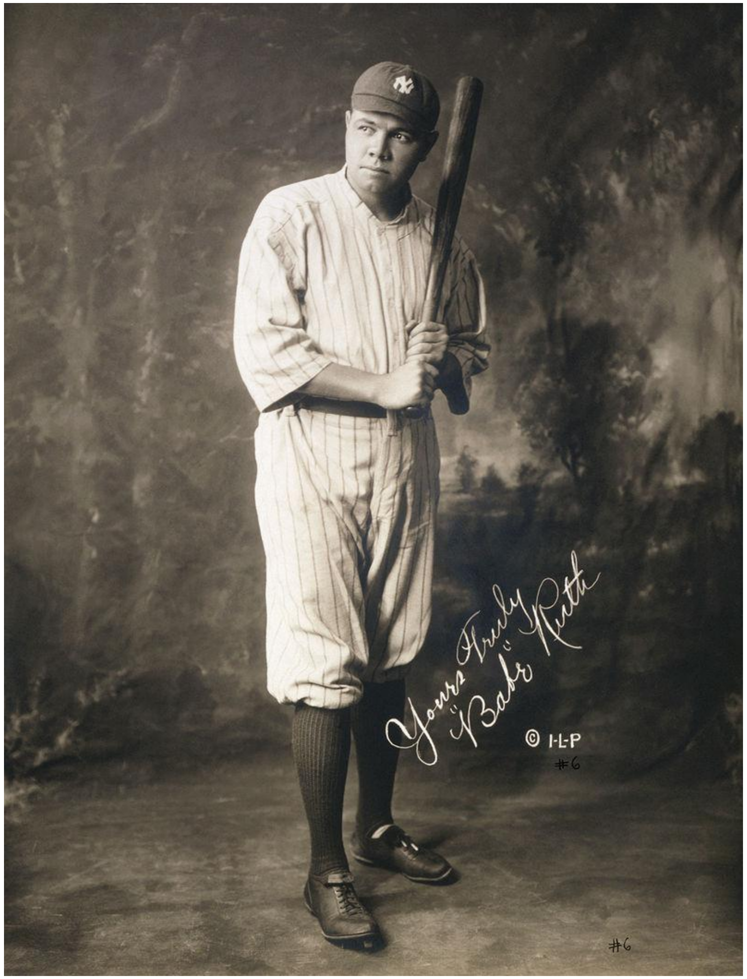


Photo: https://en.wikipedia.org/wiki/Slugging_percentage#/media/File:Babe_Ruth2.jpg

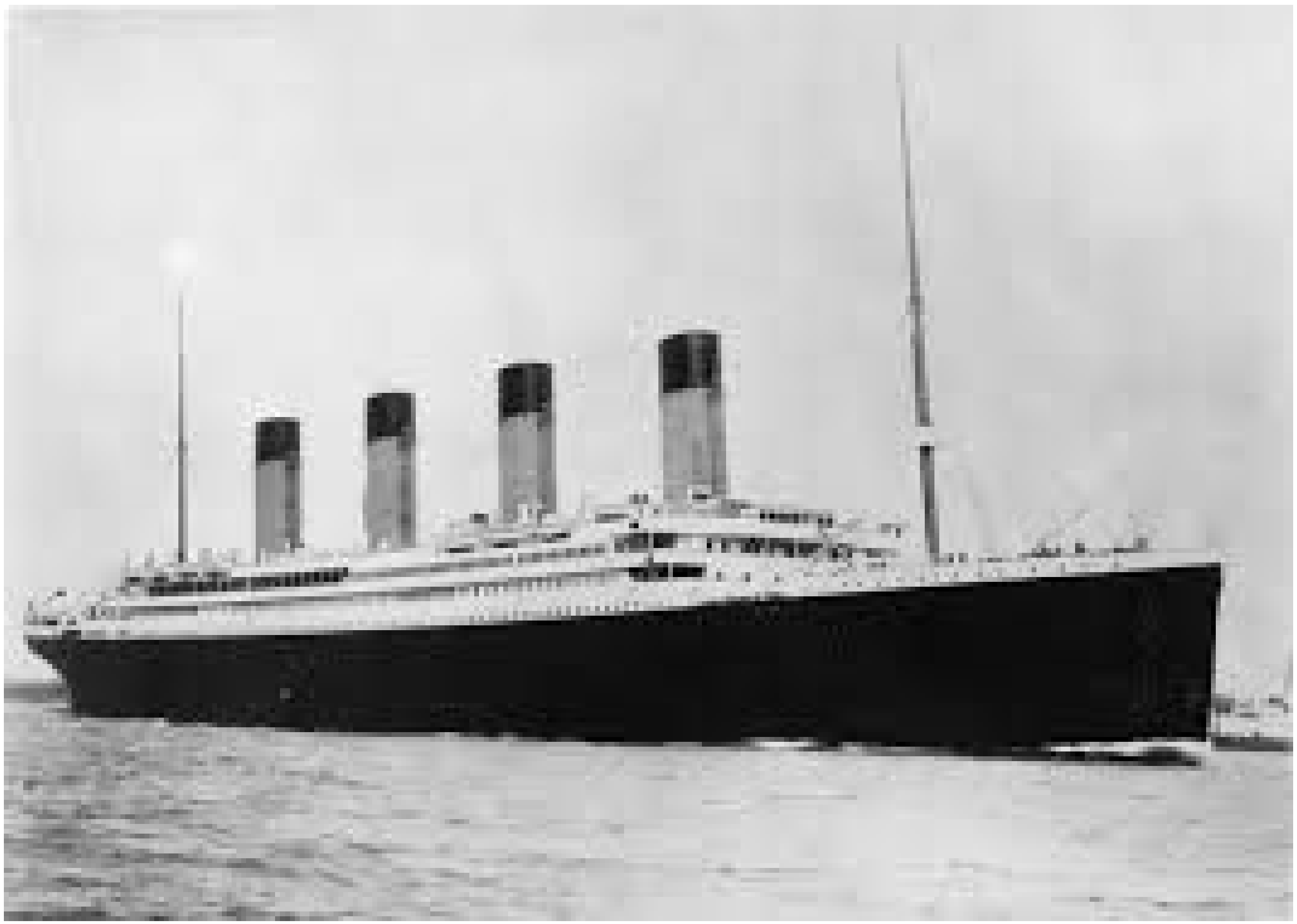
Factor Variables in R

- R takes care of the process of dummy coding variables automatically
- Dummy coding is necessary for categorical and (usually) ordinal variables
- Assigns a binary indicator (dummy variable) to indicate group membership
- For **n exclusive categories** (i.e., you can only be member of 1 category), **you need n-1 dummy variables**
- **Categories << N preferred**

Dummy Coding Example

Color	C1	C2
Red	1	0
Blue	0	1
Red	1	0
Yellow	0	0
Yellow	0	0
Blue	….	….

Feature Extraction in Unstructured Data



Generating Features…

- Conditional statements

- Subset/recode string

In either case, regular expressions can be useful

Recoding with Regular Expressions

- Regular expressions allow us to substitute based on particular patterns
- Works in a variety of languages (Python/R)
- For example, we may want to remove the cabin number and just use the area code of the ship
  - Example: A343 should be recoded to A
  - We can substitute a blank space for all numbers

Regular Expressions

.	The dot matches any single character.
\n	Matches a newline character (or CR+LF combination).
\t	Matches a tab (ASCII 9).
\d	Matches a digit [0-9].
\D	Matches a non-digit.
\w	Matches an alphanumberic character.
\W	Matches a non-alphanumberic character.
\s	Matches a whitespace character.
\S	Matches a non-whitespace character.
\	Use \ to escape special characters. For example, \. matches a dot, and \\ matches a backslash.
^	Match at the beginning of the input string.
$	Match at the end of the input string.
[0-9] All numbers
[a-z] All letters

Regular Expressions

- Return true if a pattern is found in a string, for inclusion as a separate variable
- Substitute for a value in a string, to combine like entities or to remove unnecessary ones

Cross Validation/Sampling Procedure

Sampling Procedures [Cross Validation]

- Used to prevent overfitting of model and/or improving fit
- Many Different Types
  - Holdout Method
  - K-fold Cross Validation
  - Repeated random sub-sampling validation (small n)
  - Leave-one-out cross-validation (small n)

2 Fold Cross Validation/Holdout Method

- For each fold, we randomly assign data points to two sets d_0 and d_1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). We then train on d_0 and test on d_1, followed by training on d_1 and testing on d_0.
- This has the advantage that our training and test sets are both large, and each data point is used for both training and validation on each fold.

Example: 2 Fold Cross Validation/Holdout Method

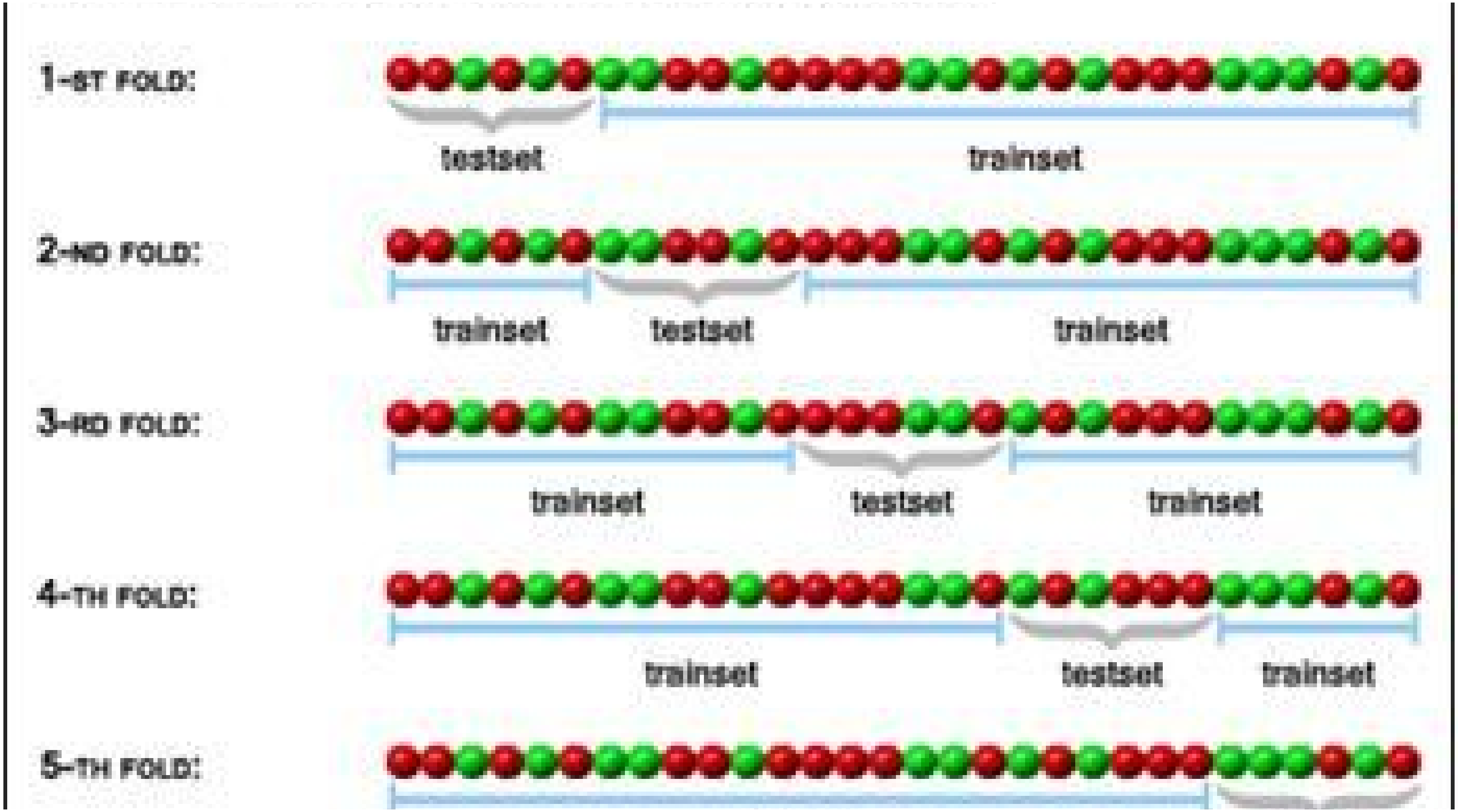
Titanic Dataset (Goal is to predict survival)

- Split sample randomly [DF_1, DF_2]
- Using DF_1 train survival model use the model to predict survival in the DF_1 sample
- Using DF_2 train survival model use the model to predict survival in the DF_1 sample

K-Fold Cross Validation

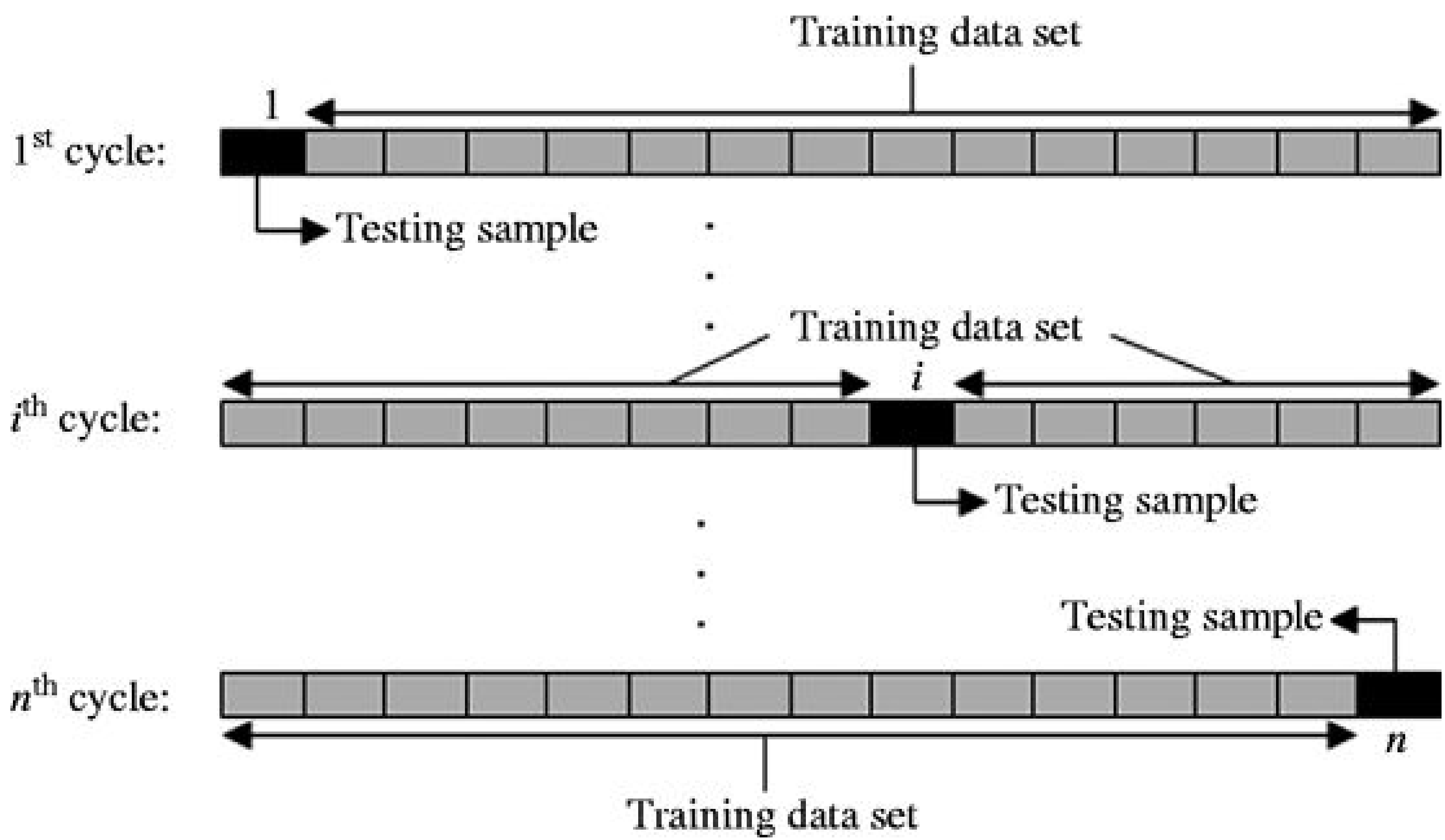
- Data divided into k subsets
- One of the subsets is the test set, the other k-1 sets are the training set
- The average error across k trials is computed

5 Fold Cross Validation



We can increase K to N

Leave-one-out cross-validation



Cross Validation

- 2-fold, 5-fold, N-fold cross validation
- Just different degrees of the same process
- Allows you to run models on subsets of the population

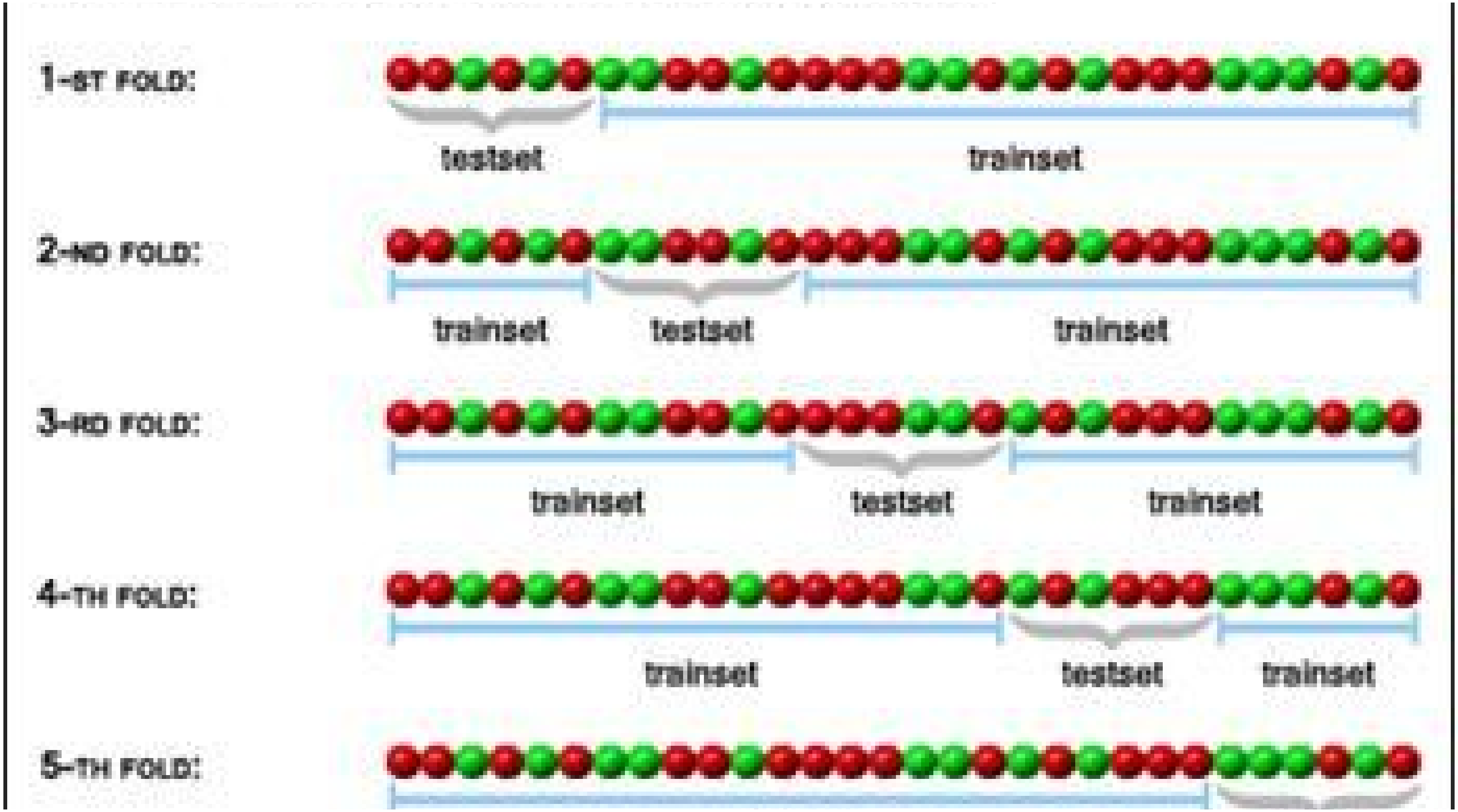
Sampling Procedures [Cross Validation]

- Used to prevent overfitting of model
- Many Different Types
  - Holdout Method
  - K-fold Cross Validation
  - Repeated random sub-sampling validation (small n)
  - Leave-one-out cross-validation (small n)

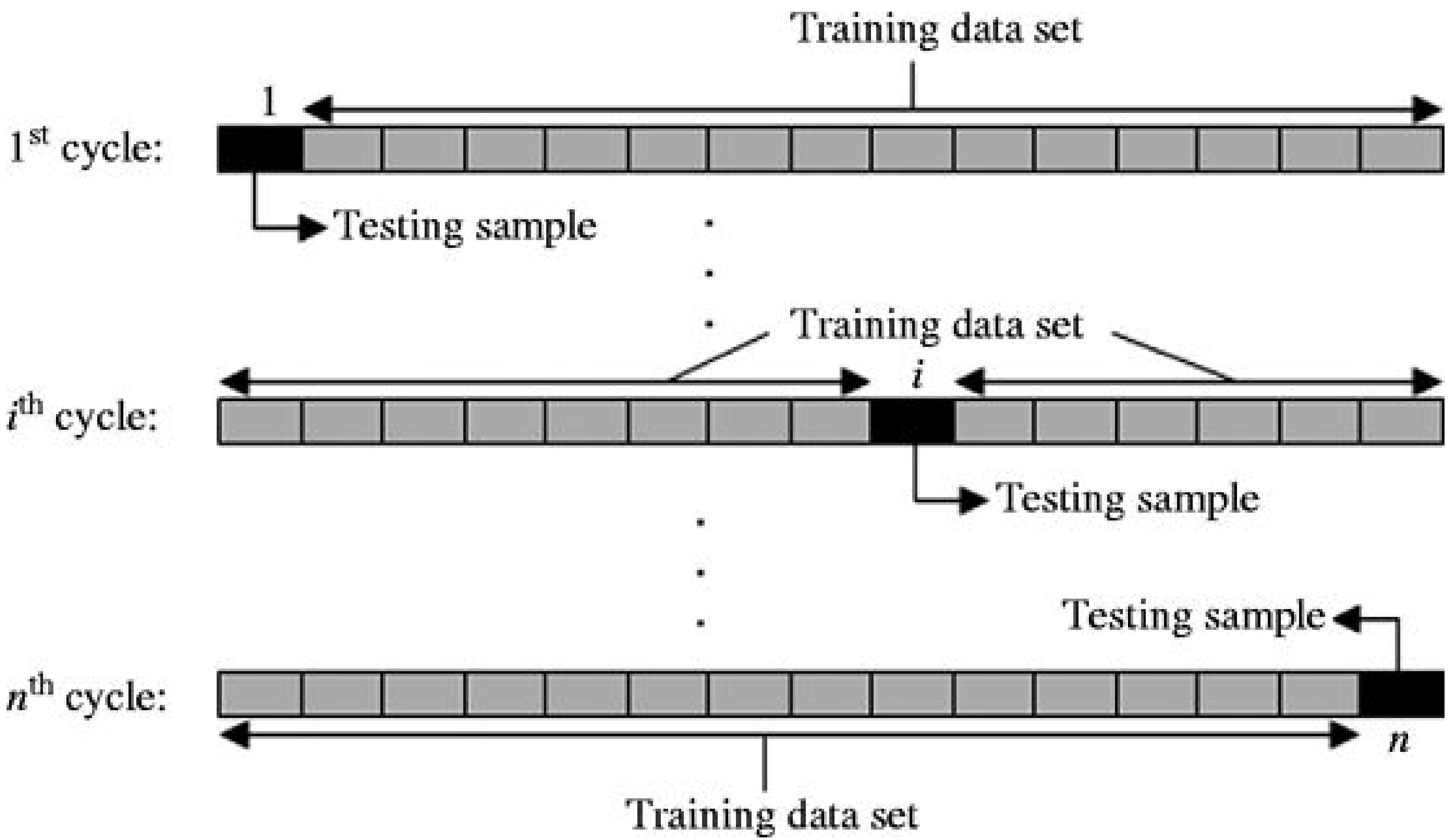
2 Fold Cross Validation/Holdout Method

- For each fold, we randomly assign data points to two sets d_0 and d_1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). We then train on d_0 and test on d_1, followed by training on d_1 and testing on d_0.
- This has the advantage that our training and test sets are both large, and each data point is used for both training and validation on each fold.

5 Fold Cross Validation



Leave-one-out cross-validation



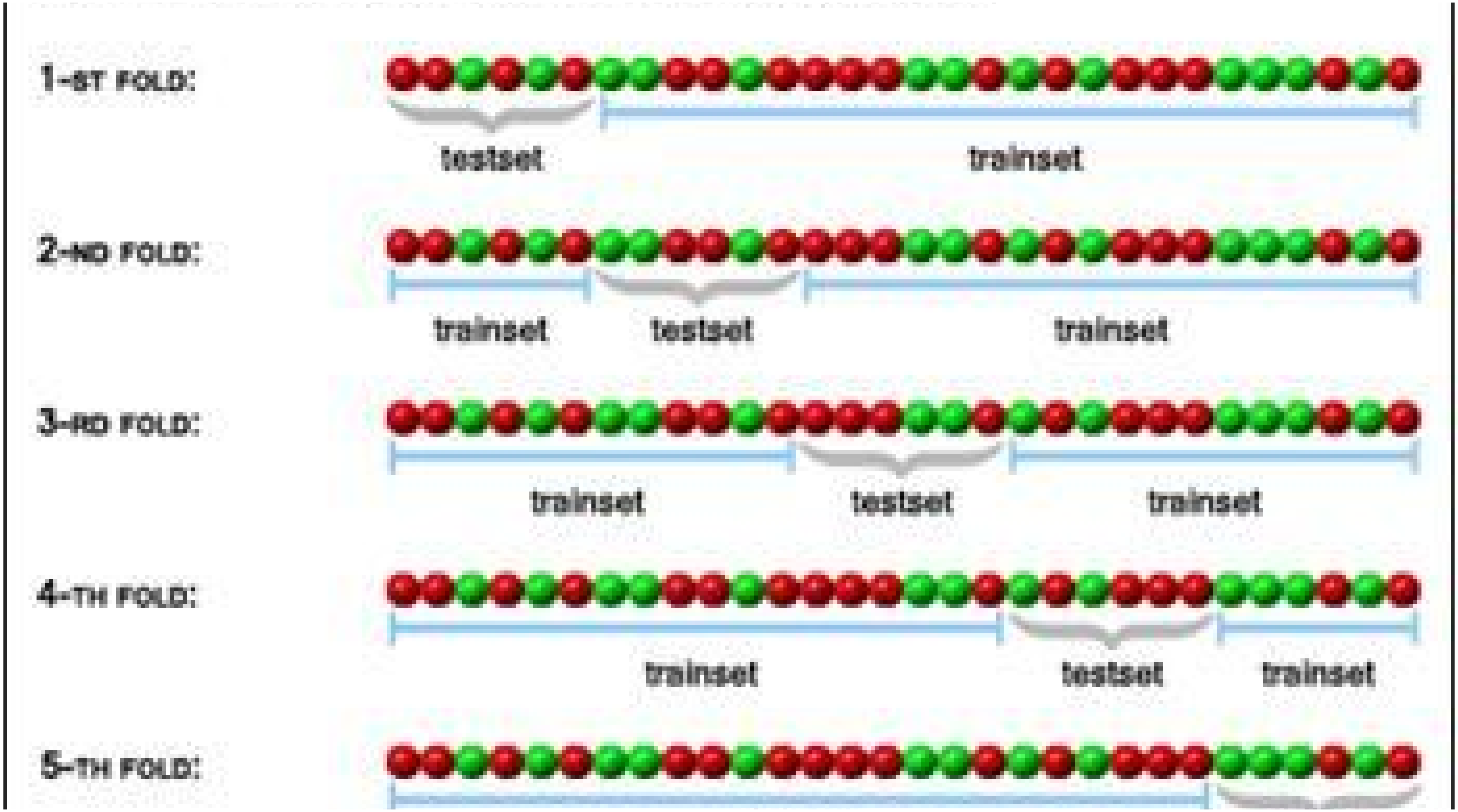
Sampling Procedures [Cross Validation]

- Used to prevent overfitting of model
- Many Different Types
  - Holdout Method
  - K-fold Cross Validation
  - Repeated random sub-sampling validation (small n)
  - Leave-one-out cross-validation (small n)

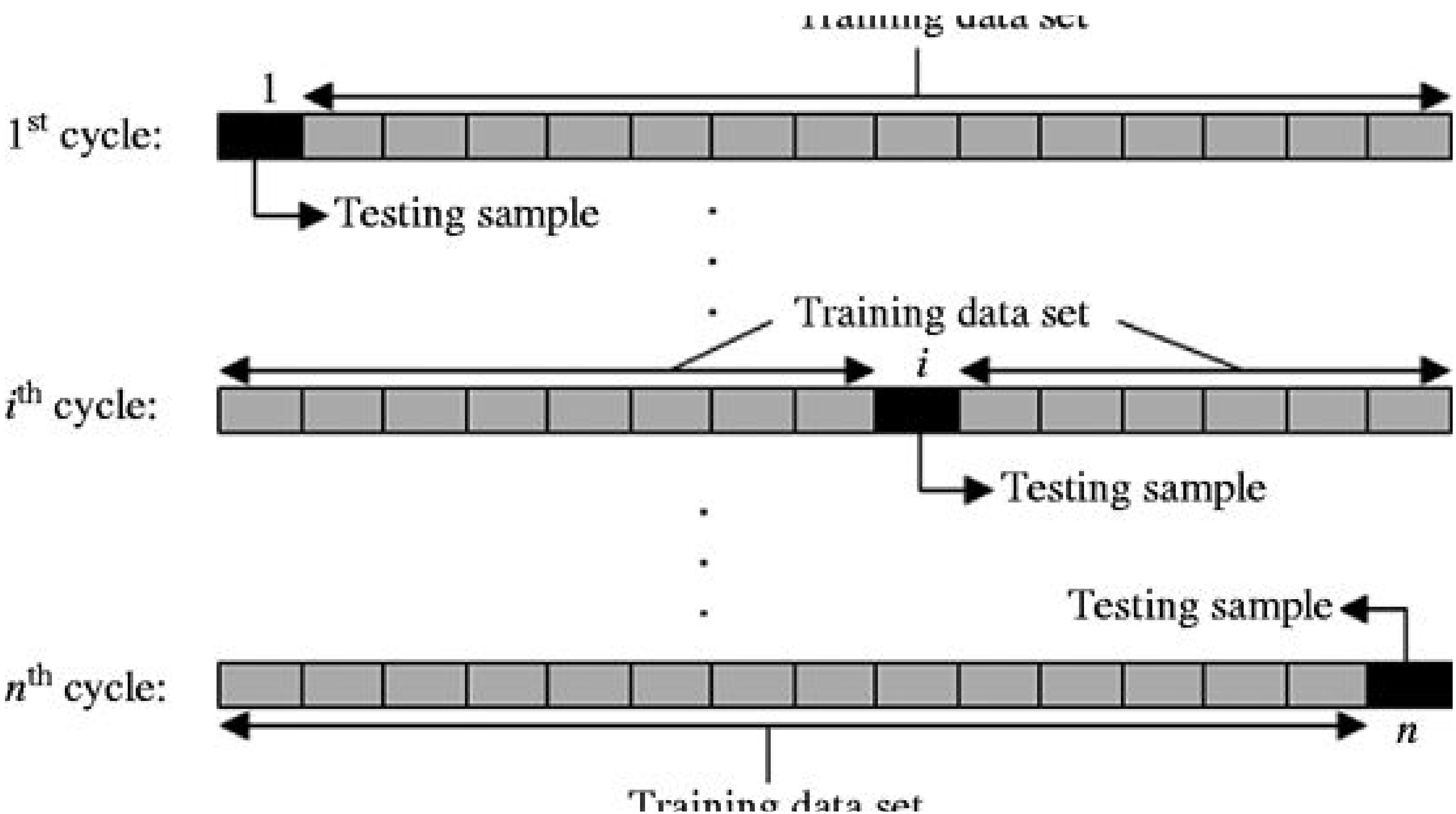
2 Fold Cross Validation/Holdout Method

- For each fold, we randomly assign data points to two sets d_0 and d_1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). We then train on d_0 and test on d_1, followed by training on d_1 and testing on d_0.
- This has the advantage that our training and test sets are both large, and each data point is used for both training and validation on each fold.

5 Fold Cross Validation



Leave-one-out cross-validation



Lab 5

Kaggle

- Post a “note” in Lab 5 with code for feature creation