Introduction to Machine Learning for Social Scientists

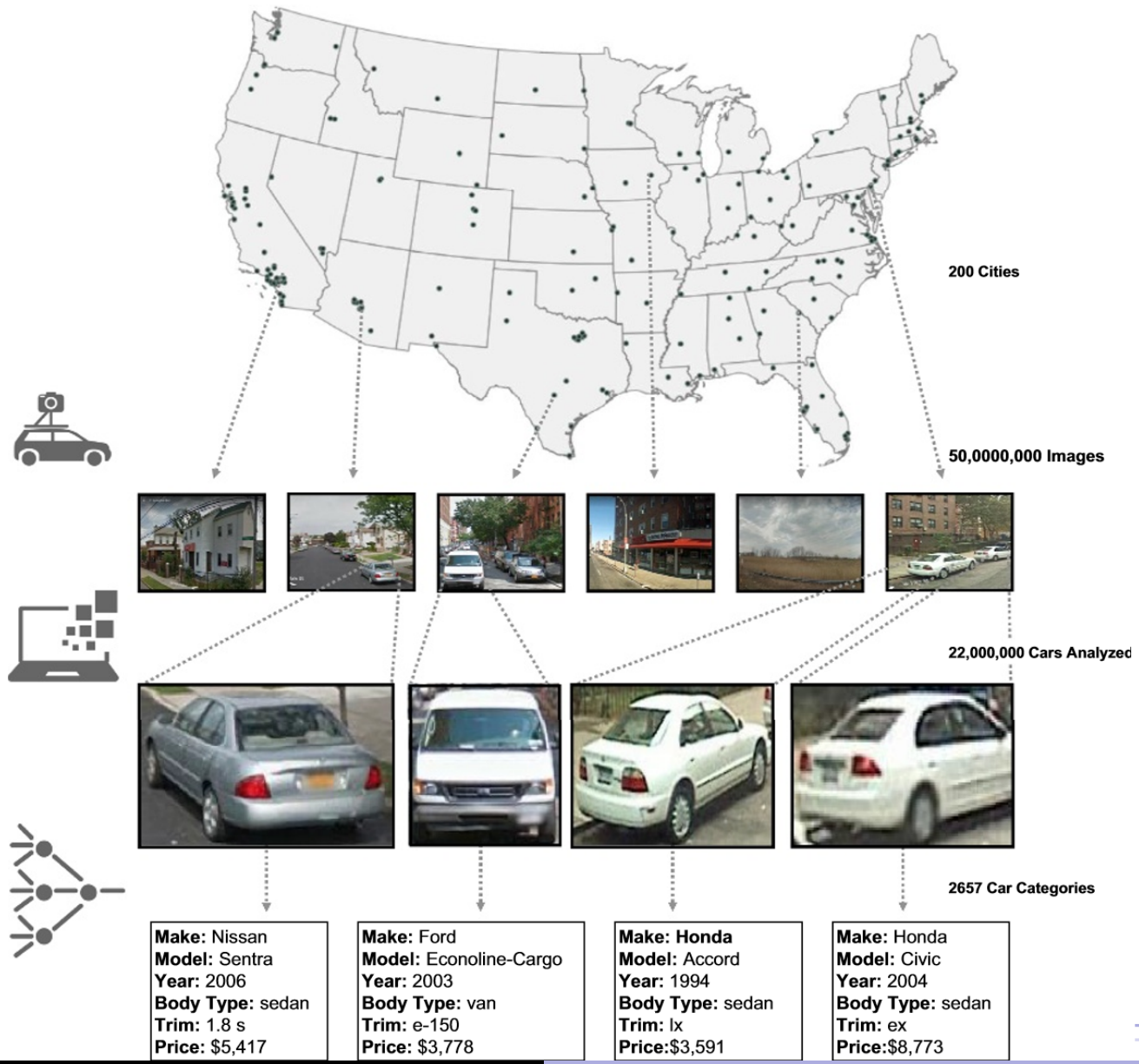
Class 7: More on performance and intro to re-sampling

Edgar Franco Vivanco

Stanford University
Department of Political Science

edgarf1@stanford.edu

Summer 2018


200 Cities
50,0000,000 Images
22,000,000 Cars Analyzed
2657 Car Categories
Make: Nissan
Model: Sentra
Year: 2006
Body Type: sedan
Trim: 1.8 s
Price: $5,417
Make: Ford
Model: Econoline-Cargo
Year: 2003
Body Type: van
Trim: e-150
Price: $3,778
Make: Honda
Model: Accord
Year: 1994
Body Type: sedan
Trim: lx
Price:$3,591
Make: Honda
Model: Civic
Year: 2004
Body Type: sedan
Trim: ex
Price:$8,773

Homework 3: Due Friday July 20th at 11:59pm

Midterm: Monday July 23rd (1:30pm to 2:50pm)

Submit your teams for the final project

- By Wed 25th before class

One submission per team: spokesperson

Today

- Mid-term guidelines
- More on accuracy, precision and recall
- Intro to re-sampling

Midterm structure:

In-class, closed book.

Midterm structure:

In-class, closed book.

- General concepts (25%)
- Regression (25%)
- Classification (25%)
- R (25%)

Midterm structure:

In-class, closed book.

- General concepts (25%)
- Regression (25%)
- Classification (25%)
- R (25%)

No computers! No cellphone! Just a pen

General concepts:

- Inference vs. Prediction
- Supervised vs. Unsupervised
- Regression vs. Classification
- Training, test and validation sets
- Overfitting
- Cost functions, errors
- Models and variables (output, input, n)
- Ability to apply and interpret these concepts

Regression:

- Coefficients
- Prediction
- Limitations of linear model
- Errors
- SSE, RSS

Classification:

- Logistic model
- From probabilities to classes
- Confusion matrix
- Performance measures

R:

- Vector types (numeric, logical, string)
- Other basic objects (matrix, dataframe, list)
- Subset your data ([],$, subset())
- Canned functions: predict, lm, glm, table, ifelse, set.seed, etc.
- Your own functions

A note on pseudocode:

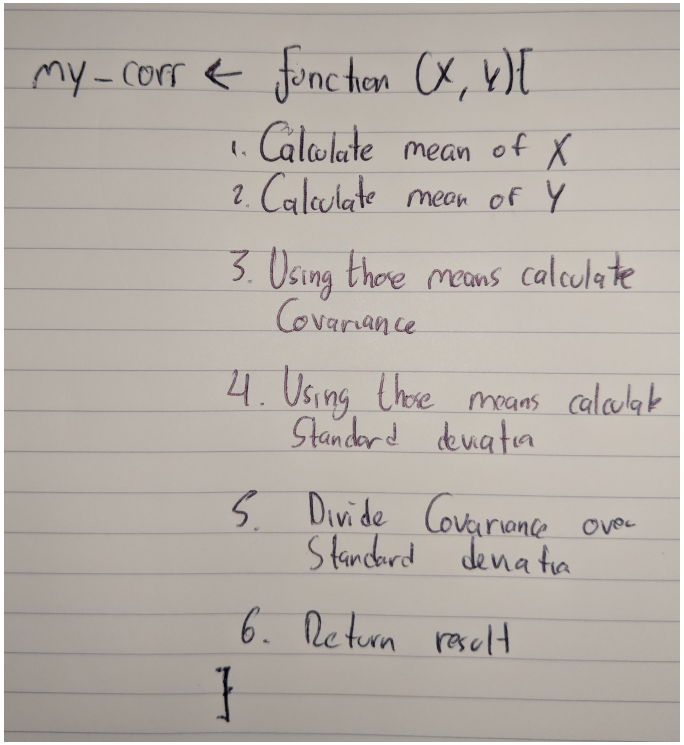
```
#########################
# Function to calculate pearson's correlation
# between two numeric vectors
# Syntax:
# my_cor(x,y):
# x: numeric vector of size n
# y: numeric vector of size n
#####################
my_cor <- function(x,y){
  ## Calculating means
  mean_x <- mean(x)
  mean_y <- mean(y)
  ## Covariance
  cov_x_y  <- sum((x-mean_x) * (y-mean_y))
  ## standard deviations
  sdx <- sum((x-mean_x)^2) * sum((y-mean_y)^2)
  ## Correlation: Cov/SD
  cor_x_y <- cov_x_y/sqrt(sdx)
  ### Return result
  return(cor_x_y)

}
```

A note on pseudocode:

```
##########################
# Function to calculate pearson's correlation
# between two numeric vectors
# Syntax:
# my_cor(x,y):
# x: numeric vector of size n
# y: numeric vector of size n
#####################
my_cor <- function(x,y){
  ## Calculating means
  mean_x <- mean(x)
  mean_y <- mean(y)
  ## Covariance
  cov_x_y  <- sum((x-mean_x) * (y-mean_y))
  ## standard deviations
  sdx <- sum((x-mean_x)^2) * sum((y-mean_y)^2)
  ## Correlation: Cov/SD
  cor_x_y <- cov_x_y/sqrt(sdx)
  ### Return result
  return(cor_x_y)

}
```



A note on calculations:

$$SSE = \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2$$

$Y = c(2.6, 6.6, 5.5, 4, 3)$
$\hat{Y}_i = c(2, 5, 5.5, 4.1, 2.9)$

Where to study?

- Slides
- Code from class
- Section materials
- Homework solutions
- Book (ISL):
  - Chapter 2
  - Chapter 3 (3.1,3.2, 3.6)
  - Chapter 4 (4.1, 4.2, 4.6)
  - Chapter 5 (5.1.1, 5.1.2)
- Extra tutorials

Questions?

Overview

Confusion Matrix

To asses the quality of our data we compare our classifications with the real data or the "gold standard".

Actual \ Guess	Yes	No
Yes		
No		

Confusion Matrix

Actual \ Guess	Yes	No
Yes	True positive	False Negative
No	False Positive	True Negative

Accuracy

Accuracy is the percentage of observations classified correctly.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalseNegative + FalsePositive}$$

Precision

How many items classified as Yes are correctly classified?

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

It is equal to 1 if all the guesses as Yes are actually Yes.

Recall

How many items **that are actually** as Yes are correctly classified? In other words, is the number of correct results divided by the number of results that should have been returned.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

It is equal to 1 if all the actual Yes are classified as Yes.

F-score

Harmonic mean of precision and recall:

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

Example:

Actual \ Guess	Yes	No
Yes	10	15
No	25	100

Accuracy= 10+100 / (10+15+25+100) = 0.73

Example:

Compared to a dumb classifier: All as negative

Actual \ Guess	Yes	No
Yes	0	25
No	0	125

Accuracy= 0+125 / (0+25+0+125) = 0.83

This useless model has zero predictive power, yet better accuracy!

Other Example:

Let's try calculating accuracy for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

Actual \ Guess	Yes	No
Yes	1	8
No	1	90

- Accuracy= 1+90 / (1+90+8+1) = 0.91
- Pretty good, right?

Other Example:

Let's try calculating accuracy for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

Actual \ Guess	Yes	No
Yes	1	8
No	1	90

- Accuracy= 1+90 / (1+90+8+1) = 0.91
- Pretty good, right?
- Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).

Other Example:

Let's try calculating accuracy for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

Actual \ Guess	Yes	No
Yes	1	8
No	1	90

- Accuracy= 1+90 / (1+90+8+1) = 0.91
- Pretty good, right?
- Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).
- Of the 9 malignant tumors, the model only correctly identifies 1 as malignant– a terrible outcome, as 8 out of 9 malignancies go undiagnosed!

Precision and Recall:

Let's try calculating precision and recall for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

Actual \ Guess	Yes	No
Yes	1	8
No	1	90

Precision and Recall:

Let's try calculating precision and recall for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

Actual \ Guess	Yes	No
Yes	1	8
No	1	90

- Precision = 1 / (1+1) = 0.5
- Our model has a precision of 0.5 – in other words, when it predicts a tumor is malignant, it is correct 50% of the time.

Precision and Recall:

Let's try calculating precision and recall for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

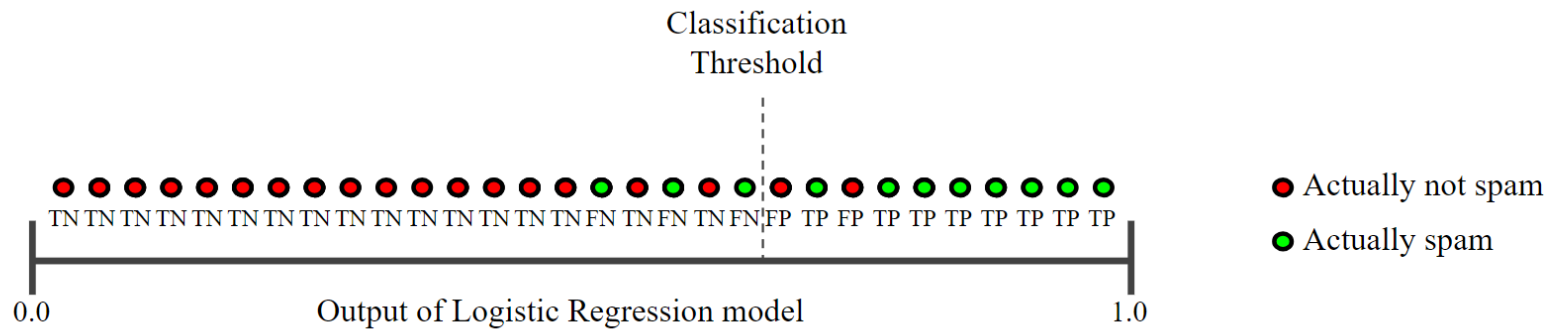
Actual \ Guess	Yes	No
Yes	1	8
No	1	90

- Precision = 1 / (1+1) = 0.5
- Our model has a precision of 0.5 – in other words, when it predicts a tumor is malignant, it is correct 50% of the time.
- Recall = 1/ (1+8) = 0.11
- Our model has a recall of 0.11 – in other words, it correctly identifies 11% of all malignant tumors.

Precision and Recall:

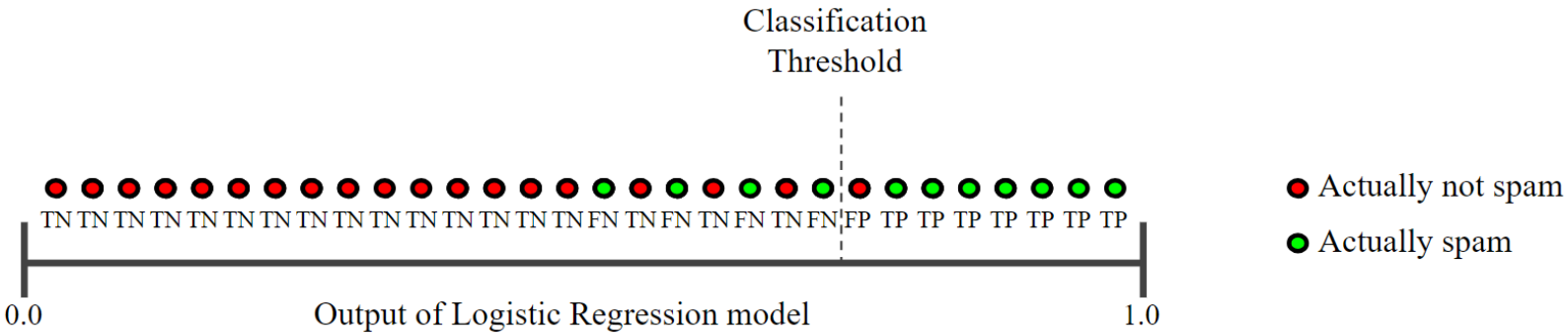
- Precision and Recall are often in tension
- Example: Classifying Spam
- Precision: 8/ (8+2)=.8
- Recall: 8/(8+3)=0.72

Actual \ Guess	Yes	No
Yes	8	3
No	2	17



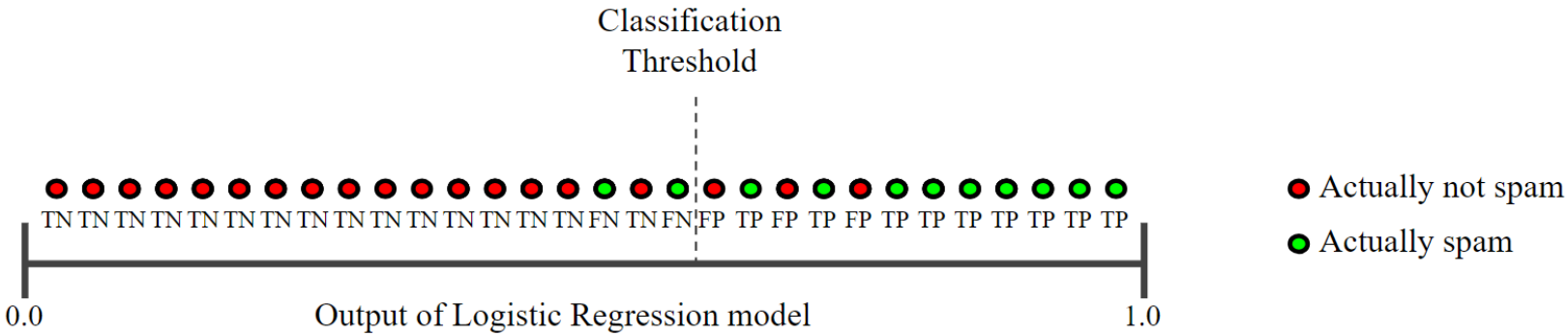
The effect of increasing the classification threshold.:

- The number of false positives decreases, but false negatives increase.
- Precision: 7 / (7+1)= 0.88
- Recall: 7/ (7+4)= 0.64
- As a result, precision increases, while recall decreases.



The effect of decreasing the classification threshold.:

- False positives increase, and false negatives decrease.
- Precision: 9 / (9+3)= 0.75
- Recall: 9/ (9+2)= 0.82
- As a result, this time, precision decreases and recall increases.



Some lessons

- Accuracy alone doesn't tell the full story, in particular when you're working with a class-imbalanced data set.

Some lessons

- Accuracy alone doesn't tell the full story, in particular when you're working with a class-imbalanced data set.
- Usually, to fully evaluate the effectiveness of a model, you must examine both precision and recall.

Some lessons

- Accuracy alone doesn't tell the full story, in particular when you're working with a class-imbalanced data set.
- Usually, to fully evaluate the effectiveness of a model, you must examine both precision and recall.
- There's always a trade-off between precision and recall.

Some lessons

- Accuracy alone doesn't tell the full story, in particular when you're working with a class-imbalanced data set.
- Usually, to fully evaluate the effectiveness of a model, you must examine both precision and recall.
- There's always a trade-off between precision and recall.
- At the end, there is not a best way to evaluate all models.

Some lessons

- Accuracy alone doesn't tell the full story, in particular when you're working with a class-imbalanced data set.
- Usually, to fully evaluate the effectiveness of a model, you must examine both precision and recall.
- There's always a trade-off between precision and recall.
- At the end, there is not a best way to evaluate all models.
- Different metrics give us different (and valuable) insights into how a classification model performs.

Overview

Evaluating fit:

- In sample: dependent variable in training data

Evaluating fit:

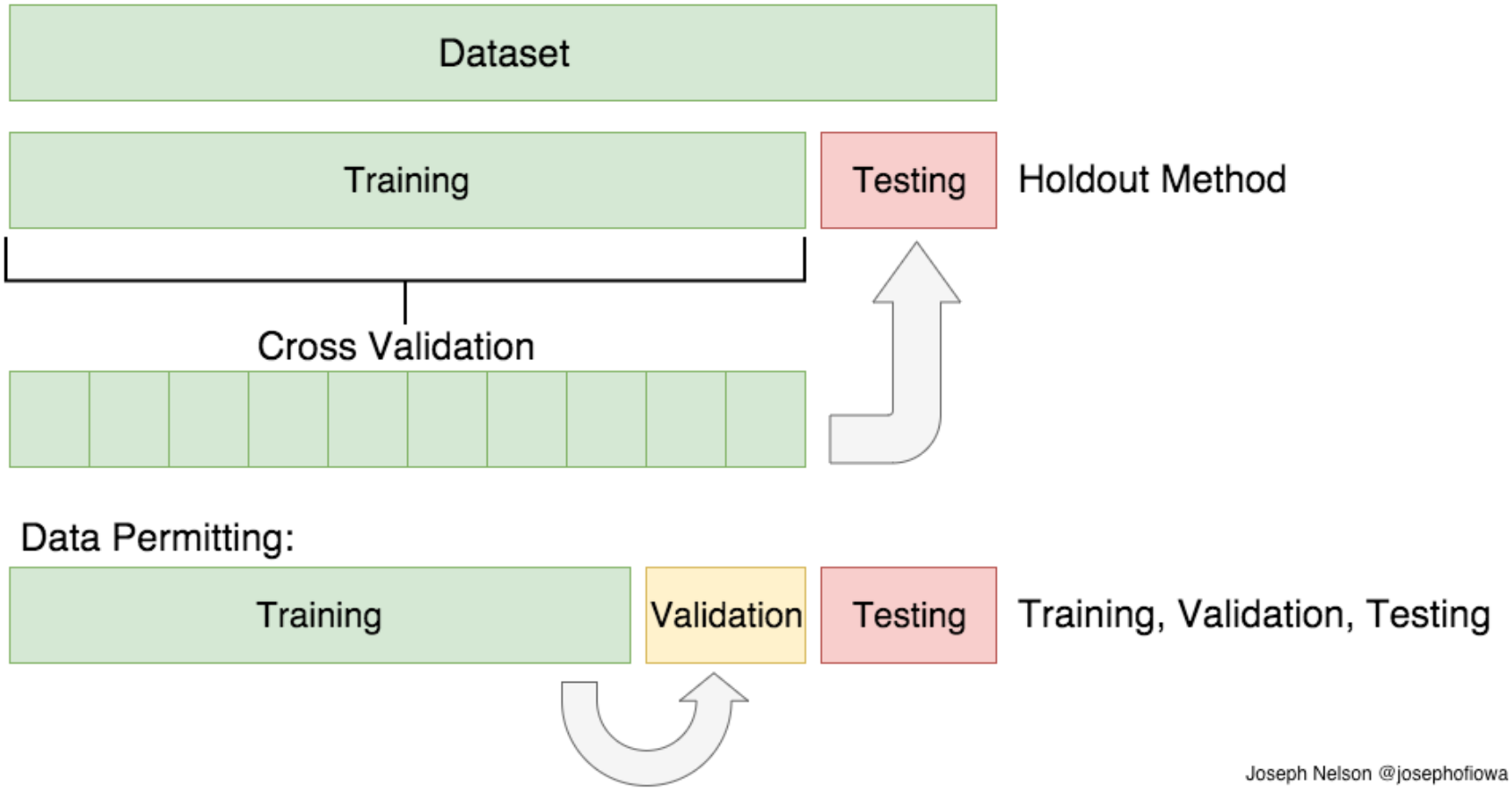
- In sample: dependent variable in training data
- But in general, we do not really care how well the method works on the training data. Rather, we are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.

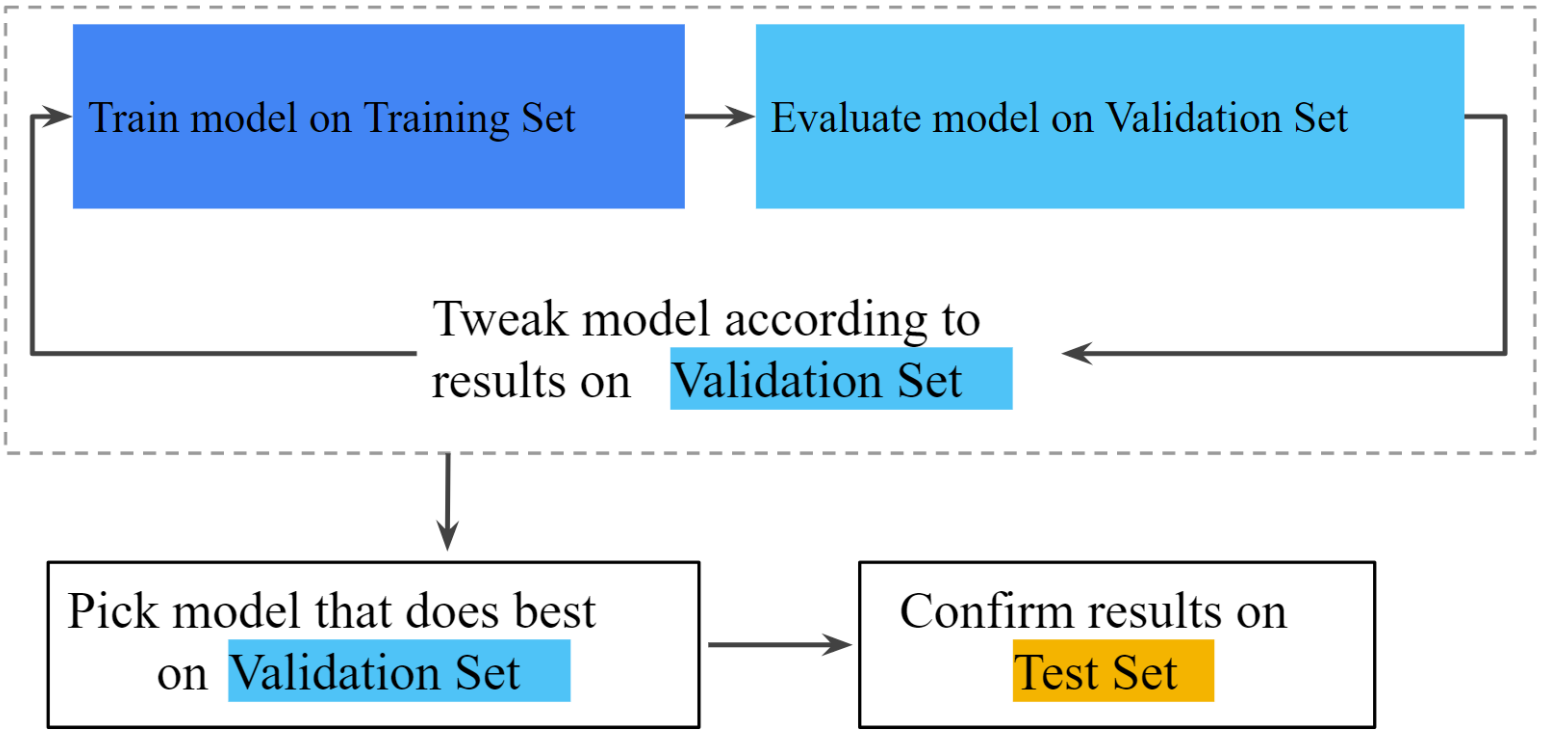
Evaluating fit:

- In sample: dependent variable in training data
- But in general, we do not really care how well the method works on the training data. Rather, we are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.
- Out of sample: held out data, test data

Evaluating fit:

- In sample: dependent variable in training data
- But in general, we do not really care how well the method works on the training data. Rather, we are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.
- Out of sample: held out data, test data
- Best practice: evaluate t with gold standard data


Dataset
Training
Testing
Holdout Method
Cross Validation
Data Permitting:
Training
Validation
Testing
Training, Validation, Testing
Joseph Nelson @josephofiowa


Train model on Training Set
Evaluate model on Validation Set
Tweak model according to results on Validation Set
Pick model that does best on Validation Set
Confirm results on Test Set

Cross-Validation: Some Intuition

Recall Optimal division of data:

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance
- Avoid overfitting and have context specific penalty

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance
- Avoid overfitting and have context specific penalty

Steps

1. Randomly divide the data into training and validation

Steps

1. Randomly divide the data into training and validation
2. Fit the model on the training set

Steps

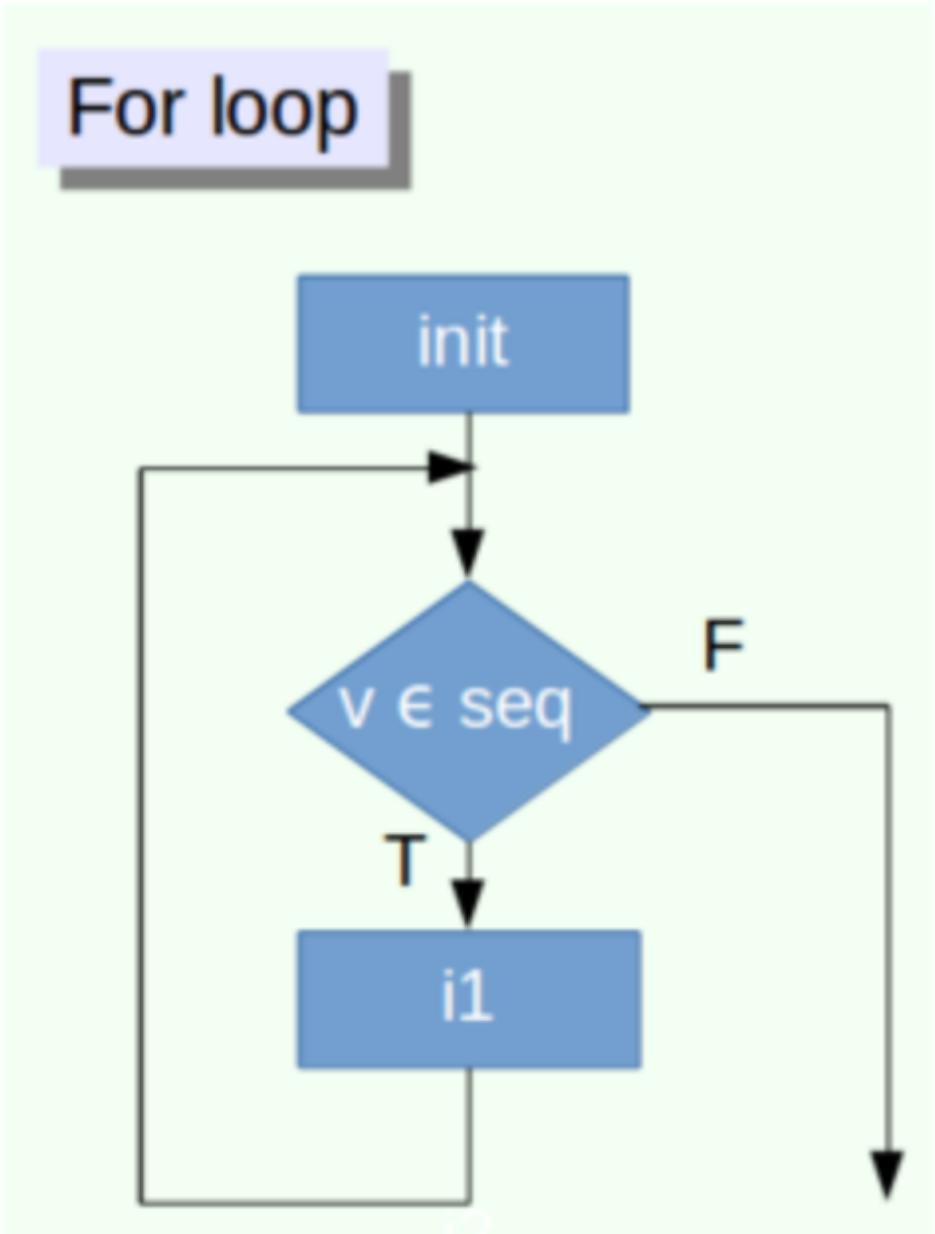
1. Randomly divide the data into training and validation
2. Fit the model on the training set
3. The fitted model is used to predict the observations in the validation set

Steps

1. Randomly divide the data into training and validation
2. Fit the model on the training set
3. The fitted model is used to predict the observations in the validation set
4. The resulting validation set error ratetypically assessed using MSE in the case of a quantitative responseprovides an estimate of the test error rate.

Leave-One-Out Cross-Validation

- Like the validation set approach, LOOCV involves splitting the set of observations into two parts.
- However, instead of creating two subsets of comparable size, a single observation (x1,y1) is used for the validation set, and the remaining observations (x2,y2),...,(xn,yn) make up the training set.
- The statistical learning method is t on the n 1 training observations, and a prediction $\hat{y_1}$ is made for the excluded observation, calculating the error $(y_1 - \hat{y_1})^2$
- We repeat the process n times
- We estimate a mean-cross validation error.


For loop
init
v ∈ seq
F
T
i1

R!

Remember:

- Midterm (Study!)
- Homework
- Submit groups