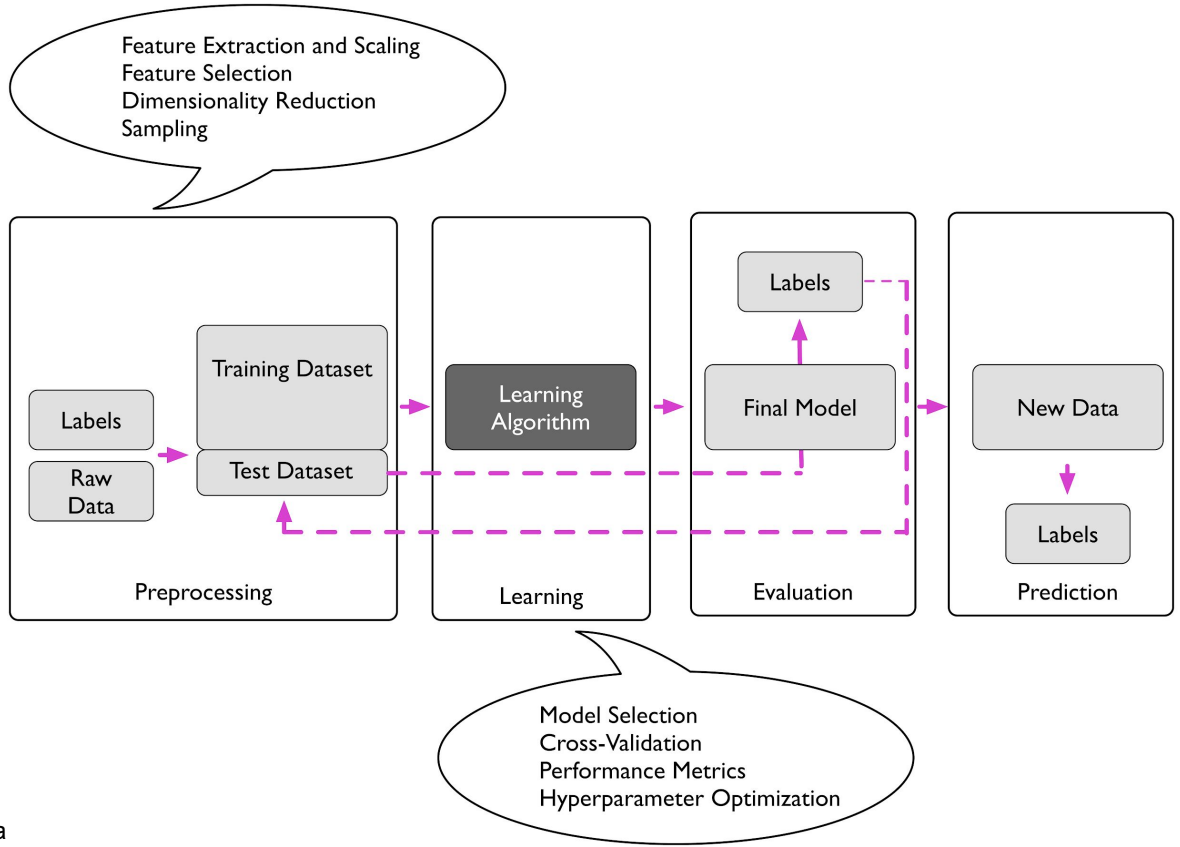
Lecture 15

Model evaluation 2

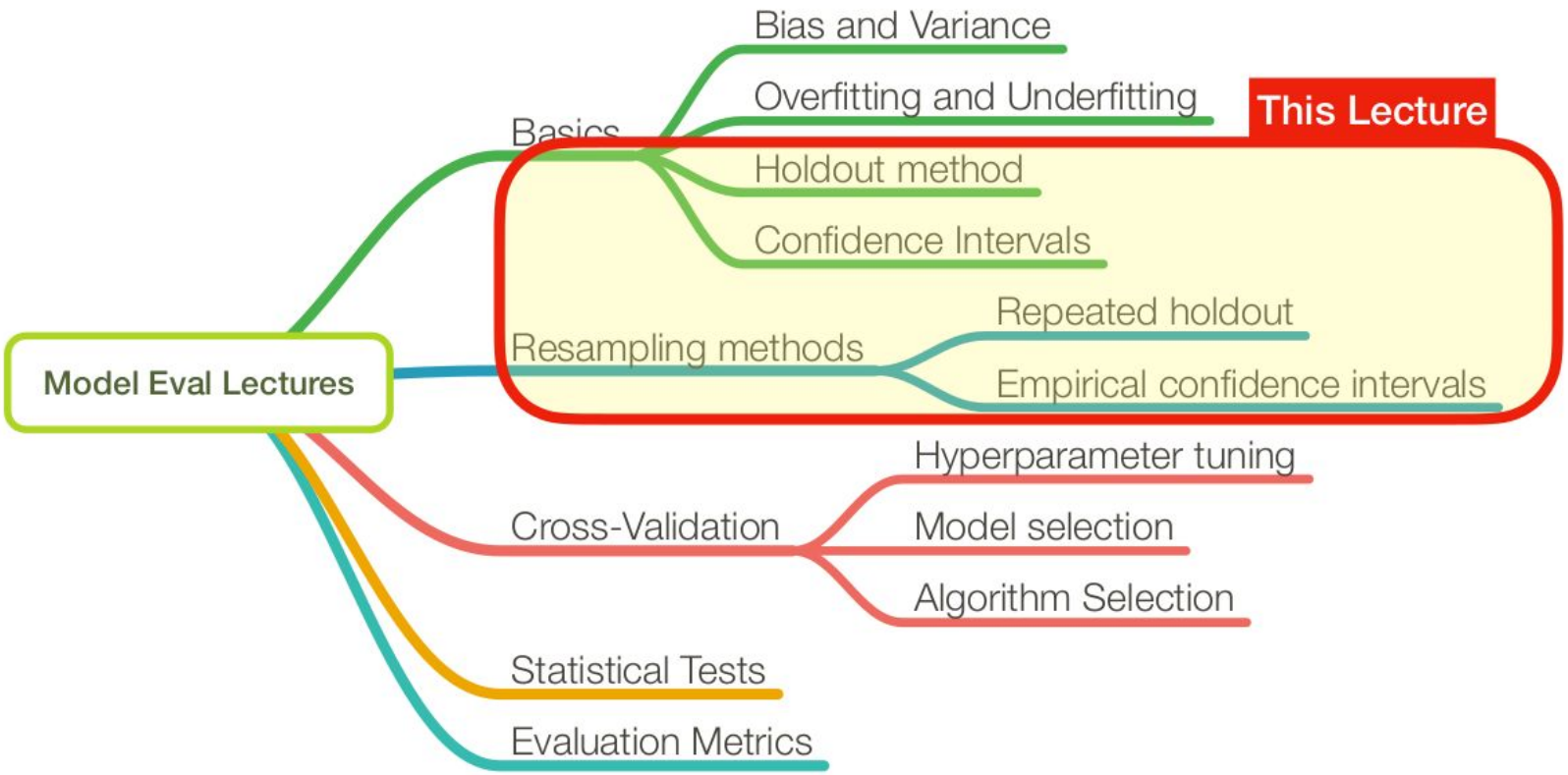
https://github.com/dalcimar/MA28CP-Intro-to-Machine-Learning

UTFPR - Federal University of Technology - Paraná
https://www.dalcimar.com/

Machine learning pipeline



Lecture overview

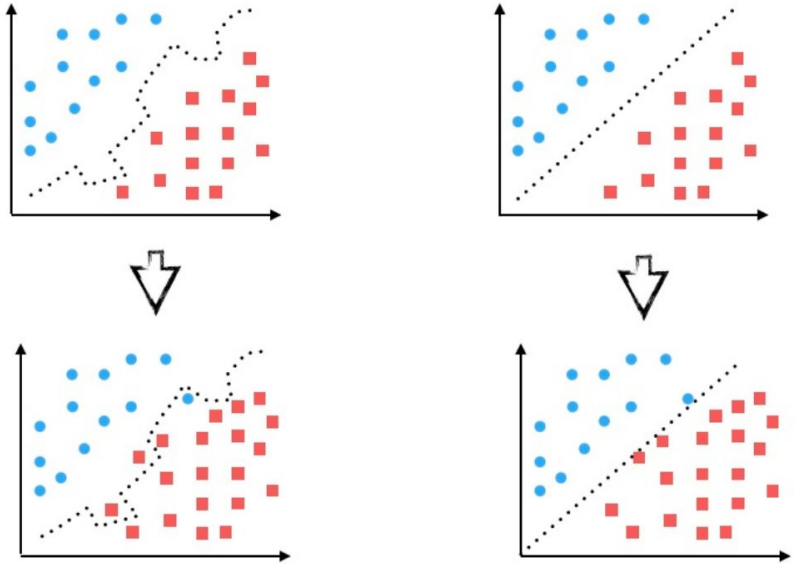


Lecturer Overview

- **Introduction**
- Holdout method for model evaluation
- Holdout method for model selection
- Confidence intervals -- normal approximation
- Resampling & repeated holdout
- Empirical confidence intervals via Bootstrap
- The 0.632 and 0.632+ Bootstrap

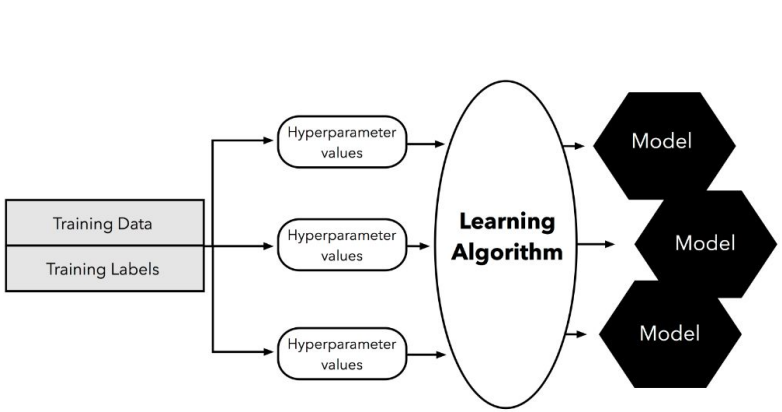
Main points why we evaluate the predictive performance of a model:

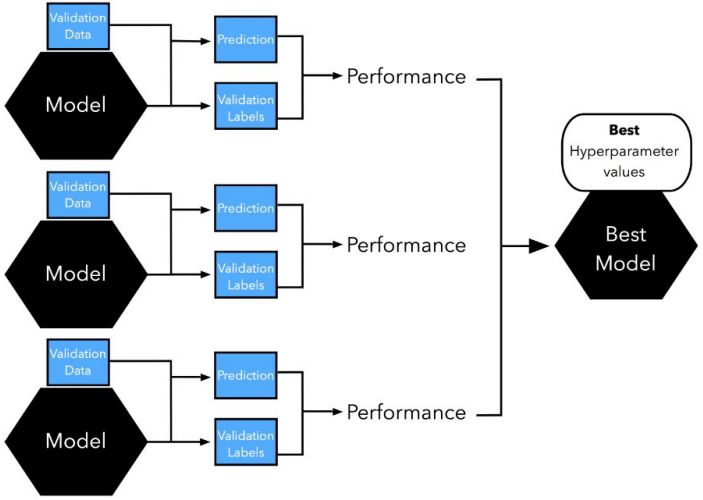
Want to estimate the generalization performance, the predictive performance of our model on future (unseen) data.



Main points why we evaluate the predictive performance of a model:

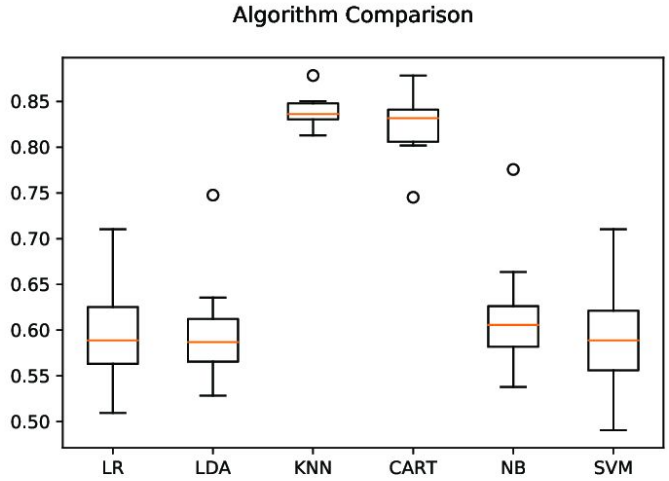
Want to increase the predictive performance by tweaking the learning algorithm and selecting the best performing model from a given hypothesis space.





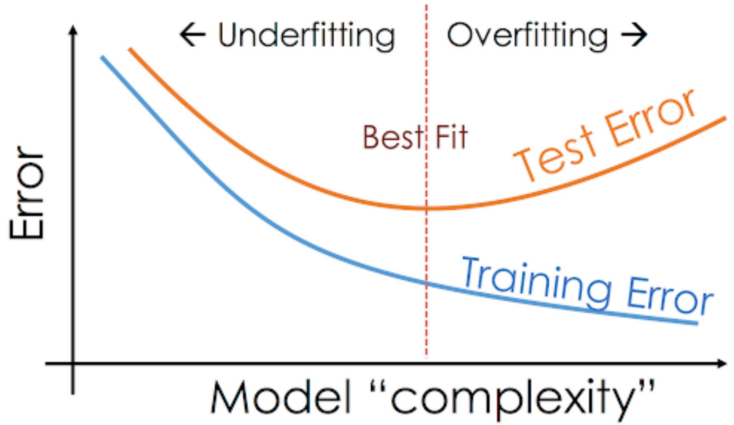
Main points why we evaluate the predictive performance of a model:

Want to identify the ML algorithm that is best-suited for the problem at hand; thus, we want to compare different algorithms, selecting the best-performing one as well as the best performing model from the algorithm's hypothesis space.



Some unfortunate facts about test sets

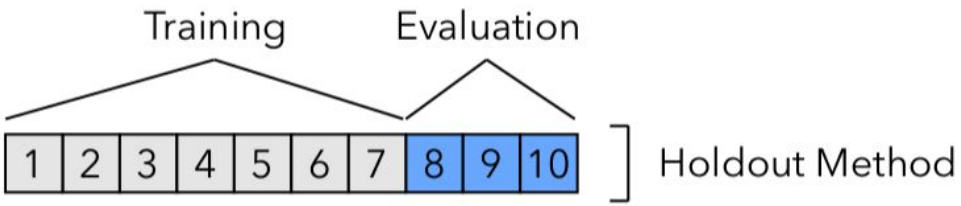
- **Training set** error is an **optimistically biased** estimator of the generalization error
- **Test set** error is an unbiased estimator of the generalization error (test sample and hypothesis chosen independently)
  - In practice, the **test set** error is actually **pessimistically** biased; why?)



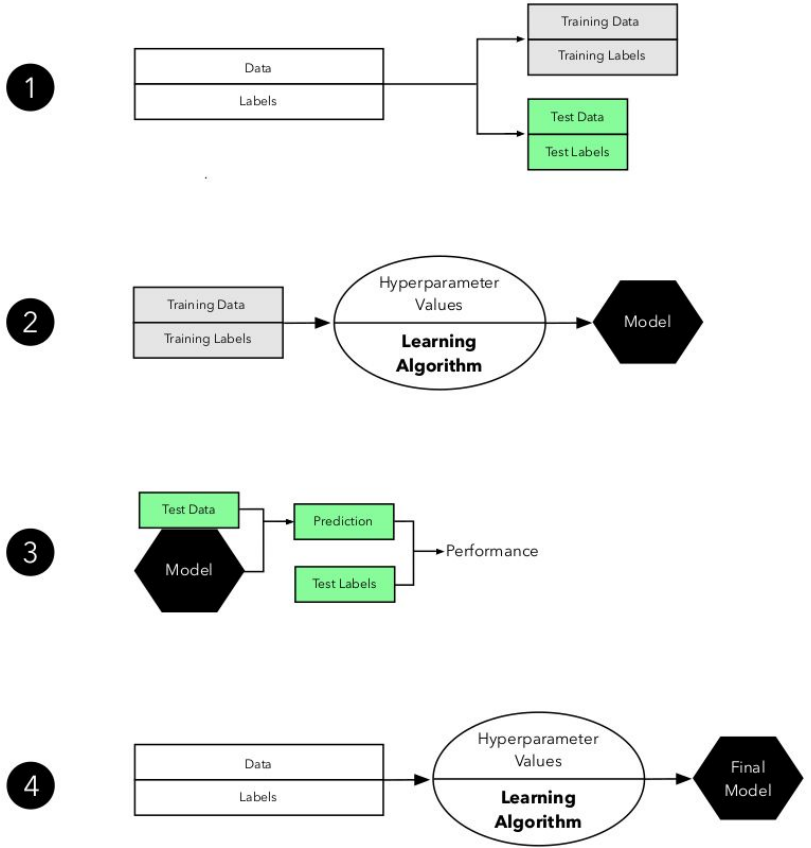
Lecturer Overview

- Introduction
- **Holdout method for model evaluation**
- Holdout method for model selection
- Confidence intervals -- normal approximation
- Resampling & repeated holdout
- Empirical confidence intervals via Bootstrap
- The 0.632 and 0.632+ Bootstrap

Holdout method for model evaluation



Often, using the holdout method is **not a good idea**...

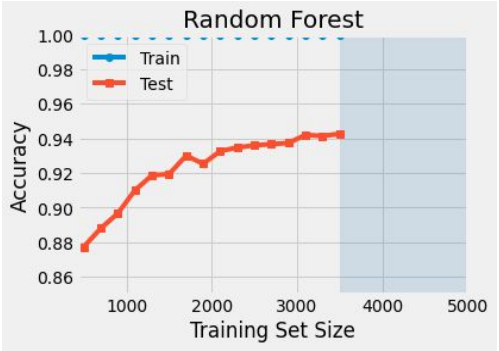


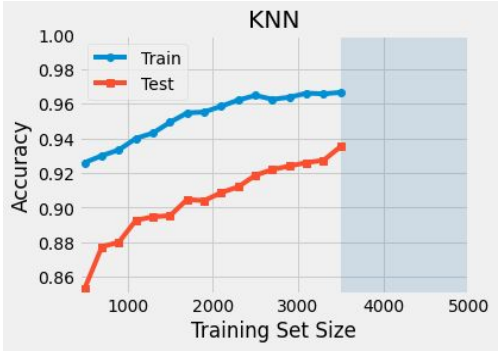
Often using the holdout method is not a good idea...

Test set error as generalization error estimator is **pessimistically biased** (not so bad)

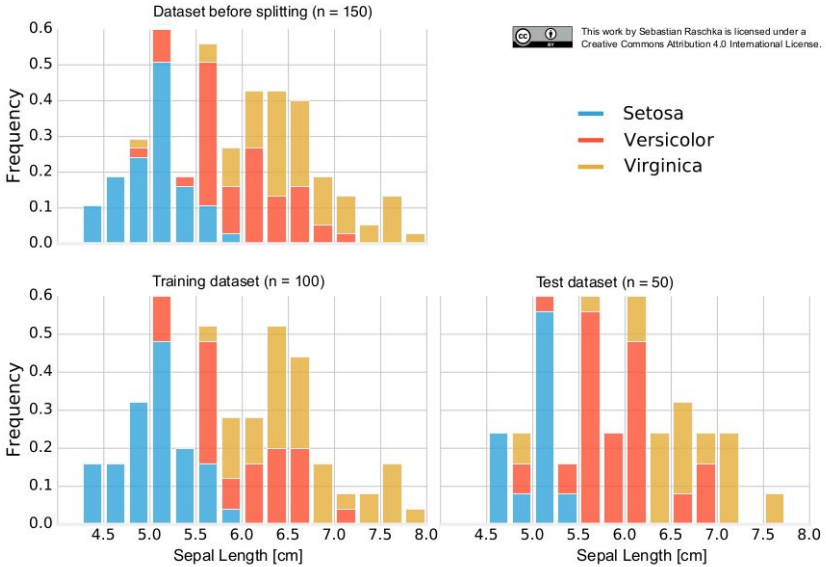
- Suppose we have the following ranking based on accuracy:
  - h2: 75% > h1: 70% > h3: 65%,
- we would still rank them the same way if we add a 10% pessimistic bias:
  - h2: 65% > h1: 60% > h3: 55%.

But it does not account for variance in the training data (bad)





Issues with subsampling (independence violation)



The Iris dataset consists of 50 Setosa, 50 Versicolor, and 50 Virginica flowers; the flower species are distributed uniformly:

- 33.3% Setosa
- 33.3% Versicolor
- 33.3% Virginia

If our random function assigns 2/3 of the flowers (100) to the training set and 1/3 of the flowers (50) to the test set, it may yield the following:

- training set → 38 x Setosa, 28 x Versicolor, 34 x Virginica
- test set → 12 x Setosa, 22 x Versicolor, 16 x Virginica

Often using the holdout method is not a good idea...

Test set errors can also be **optimistically biased**

Do CIFAR-10 Classifiers Generalize to CIFAR-10?

Benjamin Recht, UC Berkeley
Rebecca Roelofs, UC Berkeley
Ludwig Schmidt, MIT
Vaishaal Shankar, UC Berkeley

June 4, 2018

Abstract

Machine learning is currently dominated by largely experimental work focused on improvements in a few key tasks. However, the impressive accuracy numbers of the best performing models are questionable because the same test sets have been used to select these models for multiple years now. To understand the danger of overfitting, we measure the accuracy of CIFAR-10 classifiers by creating a new test set of truly unseen images. Although we ensure that the new test set is as close to the original data distribution as possible, we find a large drop in accuracy (4% to 10%) for a broad range of deep learning models. Yet, more recent models with higher original accuracy show a *smaller* drop and better overall performance, indicating that this drop is likely not due to overfitting based on adaptivity. Instead, we view our results as evidence that current accuracy numbers are brittle and susceptible to even minute natural variations in the data distribution.

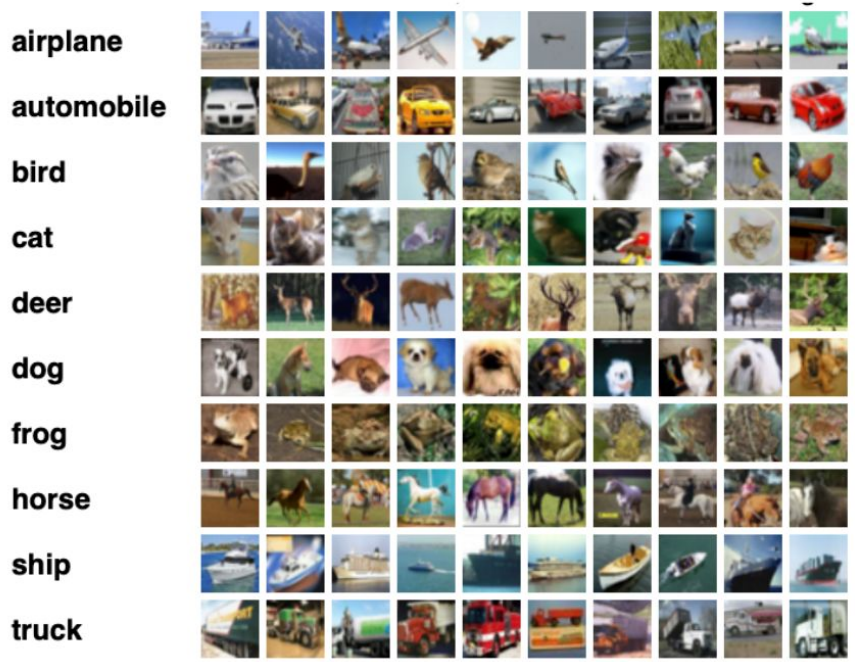
Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2018). Do CIFAR-10 classifiers generalize to CIFAR-10?. *arXiv preprint arXiv:1806.00451*.

https://arxiv.org/abs/1806.00451

The CIFAR-10 dataset

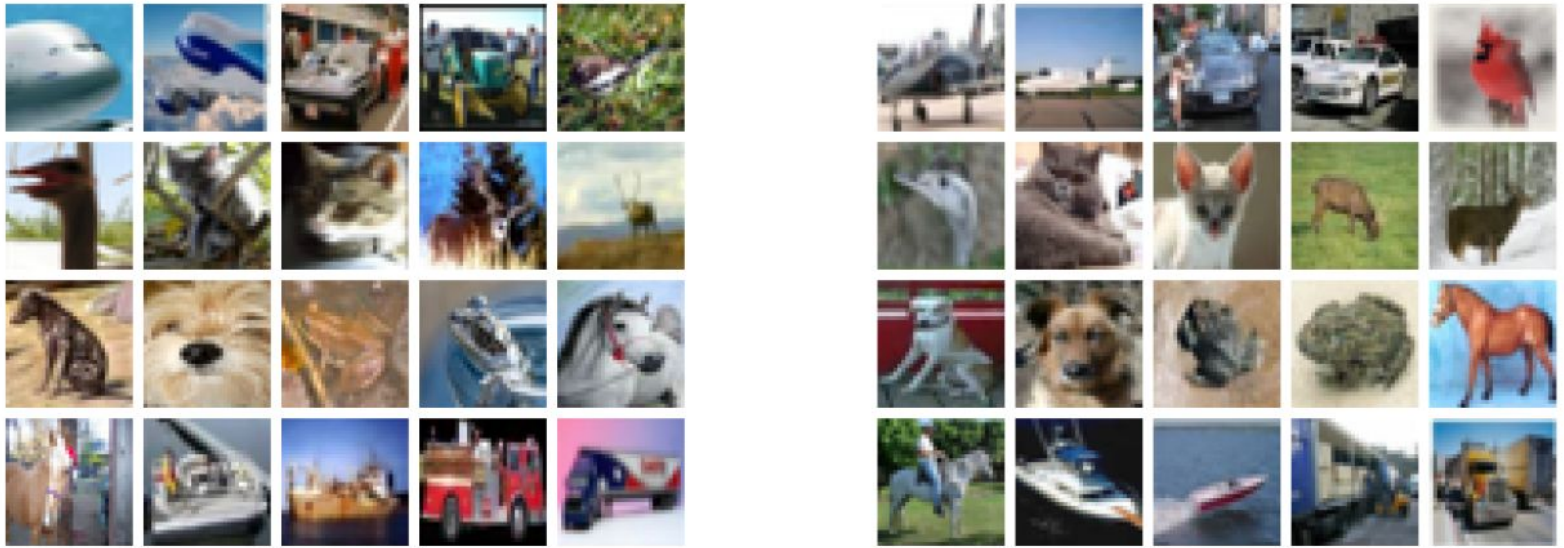
CIFAR -> Canadian Institute For Advanced Research

- 60,000 32x32 color images in 10 classes
- 6,000 images per class
- 50,000 training images and 10,000 test images



https://www.cs.toronto.edu/~kriz/cifar.html

CIFAR-10 test sets



(a) Test Set A

(b) Test Set B

Figure 1: Class-balanced random draws from the new and original test sets.[1]

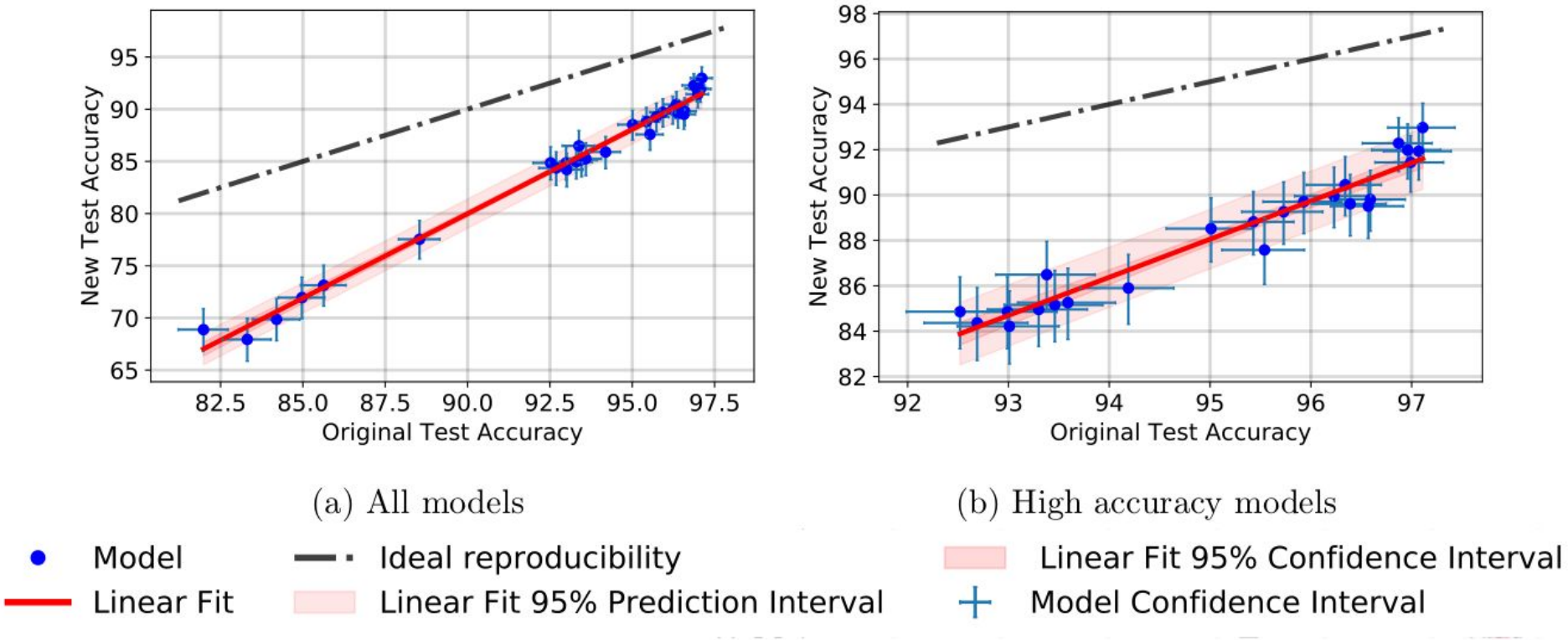


Figure 2: Model accuracy on new test set vs. model accuracy on original test set.

Table 1: Model accuracy on the original CIFAR-10 test set and the new test set, with the gap reported as the difference between the two accuracies. Δ Rank is the relative difference in the ranking from the original test set to the new test set. For example, ΔRank = −2 means a model dropped in the rankings by two positions on the new test set.

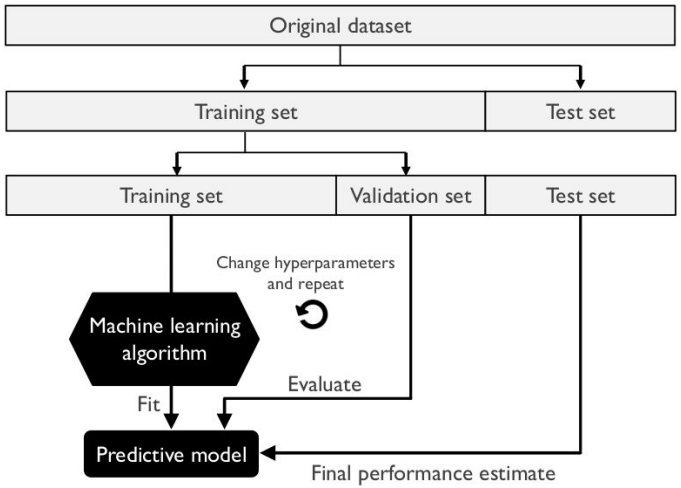
	Original Accuracy	New Accuracy	Gap	Δ Rank
shake_shake_64d_cutout [3, 4]	97.1 [96.8, 97.4]	93.0 [91.8, 94.0]	4.1	0
shake_shake_96d [4]	97.1 [96.7, 97.4]	91.9 [90.7, 93.1]	5.1	-2
shake_shake_64d [4]	97.0 [96.6, 97.3]	91.4 [90.1, 92.6]	5.6	-2
wide_resnet_28_10_cutout [3, 22]	97.0 [96.6, 97.3]	92.0 [90.7, 93.1]	5	+1
shake_drop [21]	96.9 [96.5, 97.2]	92.3 [91.0, 93.4]	4.6	+3
shake_shake_32d [4]	96.6 [96.2, 96.9]	89.8 [88.4, 91.1]	6.8	-2
darc [11]	96.6 [96.2, 96.9]	89.5 [88.1, 90.8]	7.1	-4
resnext_29_4x64d [20]	96.4 [96.0, 96.7]	89.6 [88.2, 90.9]	6.8	-2
pyramidnet_basic_110_270 [6]	96.3 [96.0, 96.7]	90.5 [89.1, 91.7]	5.9	+3
resnext_29_8x64d [20]	96.2 [95.8, 96.6]	90.0 [88.6, 91.2]	6.3	+3
wide_resnet_28_10 [22]	95.9 [95.5, 96.3]	89.7 [88.3, 91.0]	6.2	+2
pyramidnet_basic_110_84 [6]	95.7 [95.3, 96.1]	89.3 [87.8, 90.6]	6.5	0
densenet_BC_100_12 [10]	95.5 [95.1, 95.9]	87.6 [86.1, 89.0]	8	-2
neural_architecture_search [23]	95.4 [95.0, 95.8]	88.8 [87.4, 90.2]	6.6	+1
wide_resnet_tf [22]	95.0 [94.6, 95.4]	88.5 [87.0, 89.9]	6.5	+1
resnet_v2_bottleneck_164 [8]	94.2 [93.7, 94.6]	85.9 [84.3, 87.4]	8.3	-1
vgg16_keras [14, 18]	93.6 [93.1, 94.1]	85.3 [83.6, 86.8]	8.3	-1
resnet_basic_110 [7]	93.5 [93.0, 93.9]	85.2 [83.5, 86.7]	8.3	-1
resnet_v2_basic_110 [8]	93.4 [92.9, 93.9]	86.5 [84.9, 88.0]	6.9	+3
resnet_basic_56 [7]	93.3 [92.8, 93.8]	85.0 [83.3, 86.5]	8.3	0
resnet_basic_44 [7]	93.0 [92.5, 93.5]	84.2 [82.6, 85.8]	8.8	-3
vgg_15_BN_64 [14, 18]	93.0 [92.5, 93.5]	84.9 [83.2, 86.4]	8.1	+1
resnet_preact_tf [7]	92.7 [92.2, 93.2]	84.4 [82.7, 85.9]	8.3	0
resnet_basic_32 [7]	92.5 [92.0, 93.0]	84.9 [83.2, 86.4]	7.7	+3
cudaconvnet [13]	88.5 [87.9, 89.2]	77.5 [75.7, 79.3]	11	0
random_features_256k_aug [2]	85.6 [84.9, 86.3]	73.1 [71.1, 75.1]	12	0
random_features_32k_aug [2]	85.0 [84.3, 85.7]	71.9 [69.9, 73.9]	13	0
random_features_256k [2]	84.2 [83.5, 84.9]	69.9 [67.8, 71.9]	14	0
random_features_32k [2]	83.3 [82.6, 84.0]	67.9 [65.9, 70.0]	15	-1
alexnet_tf	82.0 [81.2, 82.7]	68.9 [66.8, 70.9]	13	+1

Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2018). Do CIFAR-10 classifiers generalize to CIFAR-10?. *arXiv preprint arXiv:1806.00451*.

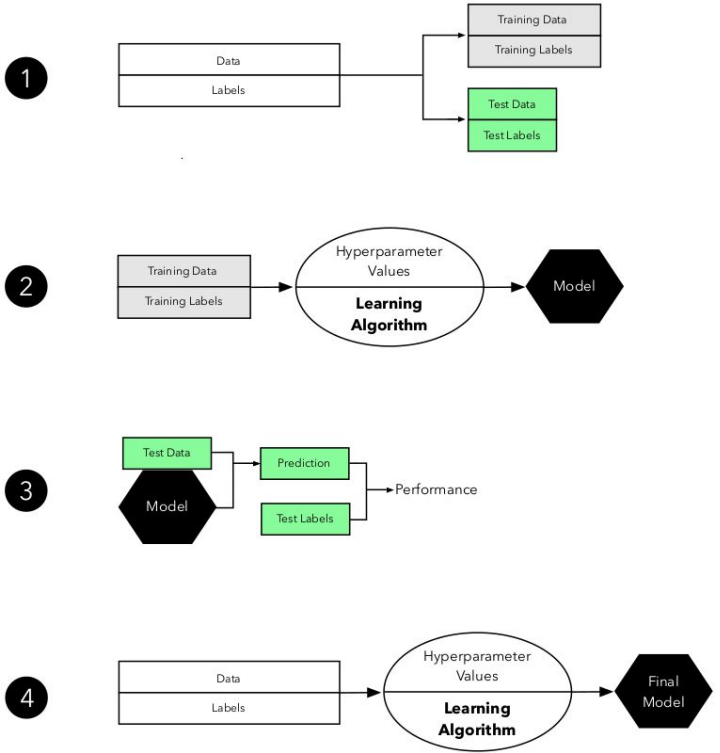
Lecturer Overview

- Introduction
- Holdout method for model evaluation
- **Holdout method for model selection**
- Confidence intervals - normal approximation
- Resampling & repeated holdout
- Empirical confidence intervals via Bootstrap
- The 0.632 and 0.632+ Bootstrap

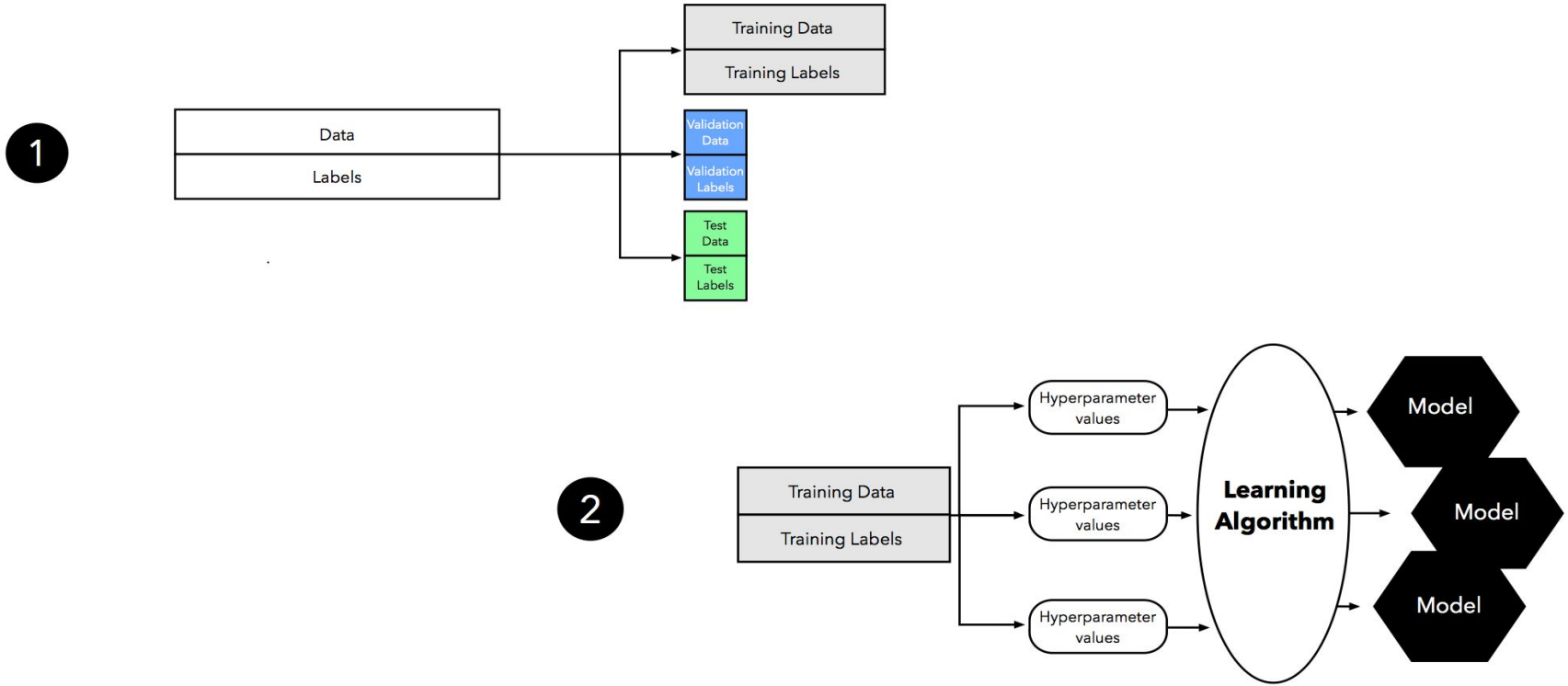
Holdout method for model selection



Holdout method for model evaluation

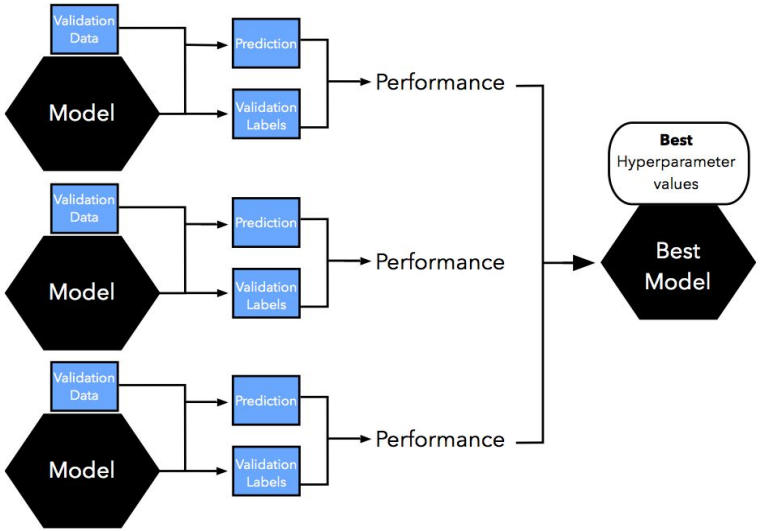


Holdout method for model selection

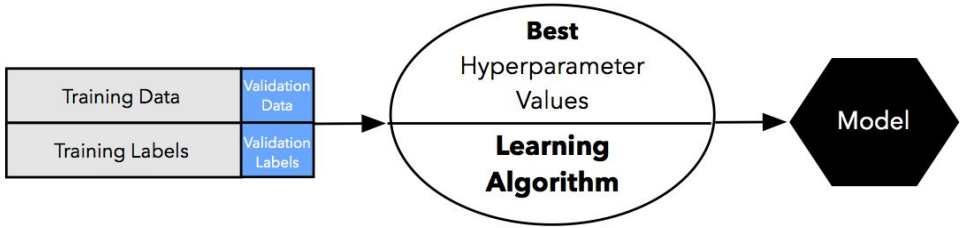


Holdout method for model selection

3

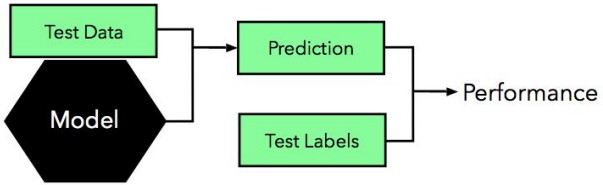


4

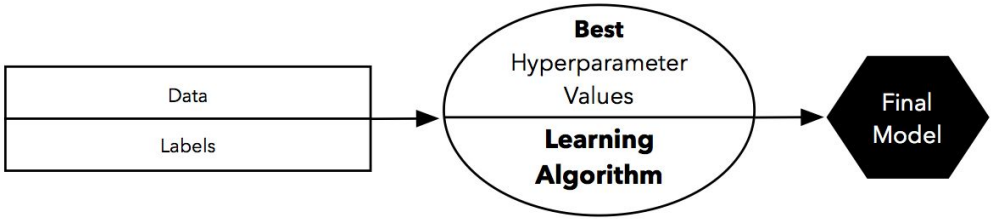


Holdout method for model selection

5



6



Lecturer Overview

- Introduction
- Holdout method for model evaluation
- Holdout method for model selection
- **Confidence intervals - normal approximation**
- Resampling & repeated holdout
- Empirical confidence intervals via Bootstrap
- The 0.632 and 0.632+ Bootstrap