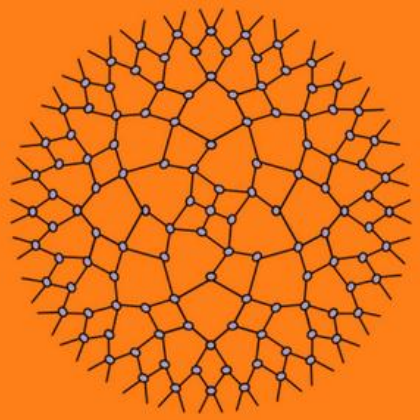
ML Algorithms

NEURAL NETWORKS



Class

A Detailed Look At Neural Networks

Topic

Confusion Matrix;
Comparing Multiple Classifiers;
Generalization;
Bias-Variance Trade-Off

Confusion Matrix

Data:

- **60** cases with **positive** classes
- **15** cases with **negative** classes

	Predicted Positive	Predicted Negative
Actual Positive	40 (True Positive, TP)	20(False Negative, FN)
Actual Negative	10 (False Positive, FP)	5(True Negative, TN)

Confusion Matrix

Data:

- **60** cases with **positive** classes
- **15** cases with **negative** classes

	Predicted Positive	**Predicted Negative**
Actual Positive	40 (True Positive, TP)	20(False Negative, FN)
Actual Negative	10 (False Positive, FP)	5(True Negative, TN)

Confusion Matrix

Data:

- **60** cases with **positive** classes
- **15** cases with **negative** classes

	Predicted Positive	**Predicted Negative**
Actual Positive	40 (True Positive, TP)	20(False Negative, FN)
Actual Negative	10 (False Positive, FP)	5(True Negative, TN)

- Sensitivity = $\frac{TP}{TP+FN} = \frac{40}{40+20} \sim$ 67%

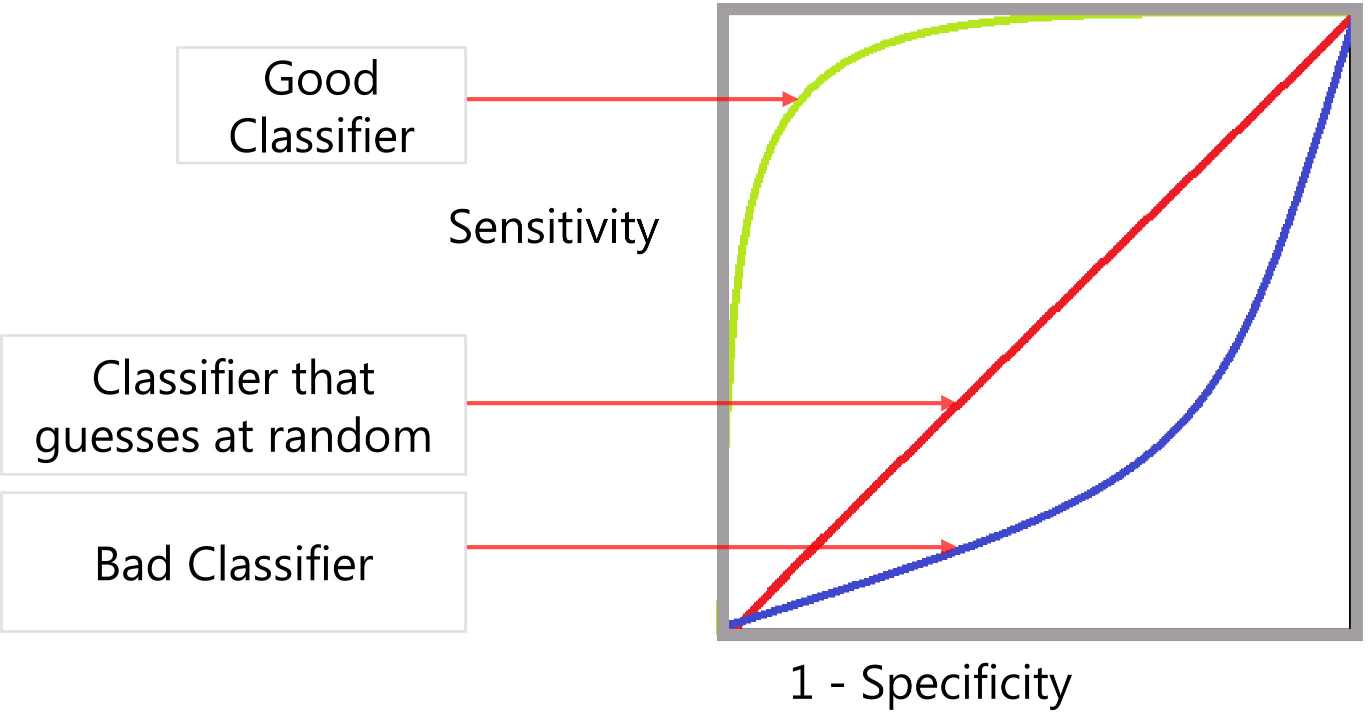
True Positives : Total Positives

- Specificity = $\frac{TN}{FP+TN} = \frac{5}{10+5} \sim$ 33%

True Negatives : Total Negatives

Comparing Multiple Classifiers

Comparing the performance of 2 classifiers



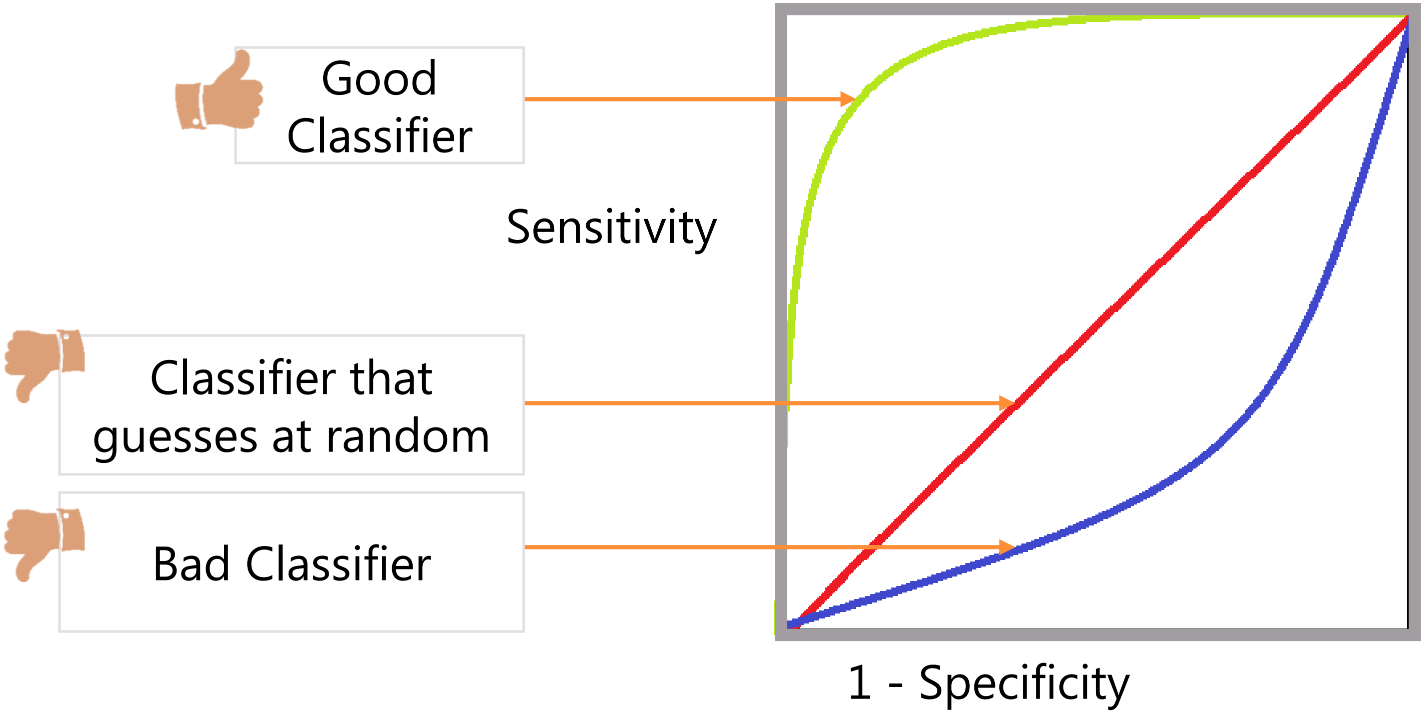
ROC Curves

- Plot Sensitivity or True Positive Rate vs. (1-Specificity) or False Positive Rate for all values of threshold

- **Score**: Output of a decision function
- If it's above/below pre-specified threshold, the predicted class is marked as positive or negative
- Each chosen threshold can have varying values for specificity and sensitivity

Comparing Multiple Classifiers

Comparing the performance of 2 classifiers



ROC Curves

- Plot Sensitivity or True Positive Rate vs. (1-Specificity) or False Positive Rate for all values of threshold

AUC

- Area under curve
- Perfect classifier: AUC of 1
- Higher is better

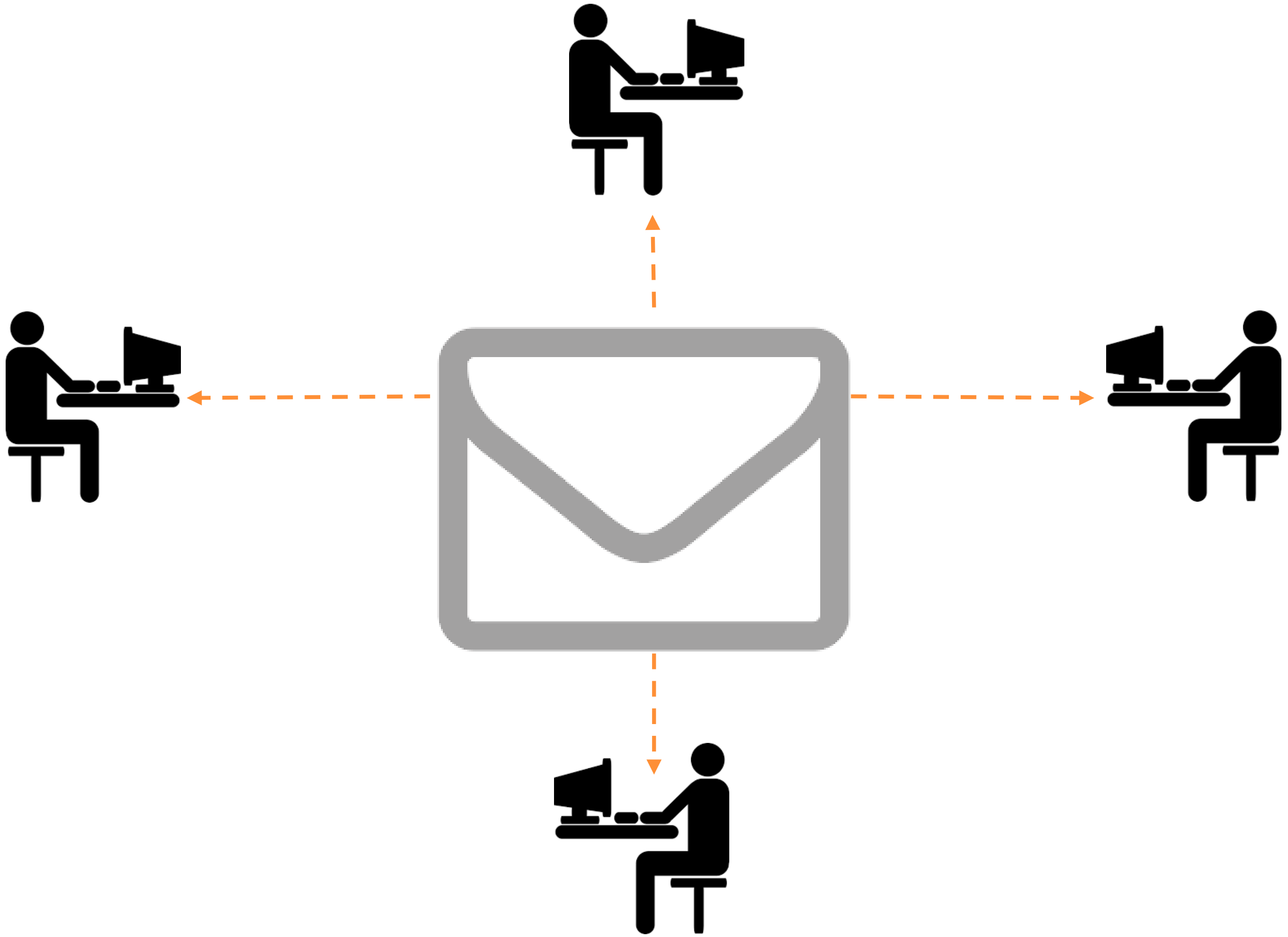
- **Score**: Output of a decision function
- If it's above/below pre-specified threshold, the predicted class is marked as positive or negative
- Each chosen threshold can have varying values for specificity and sensitivity

Samples and Populations

Data used by algorithms are a sample from an infinite universe

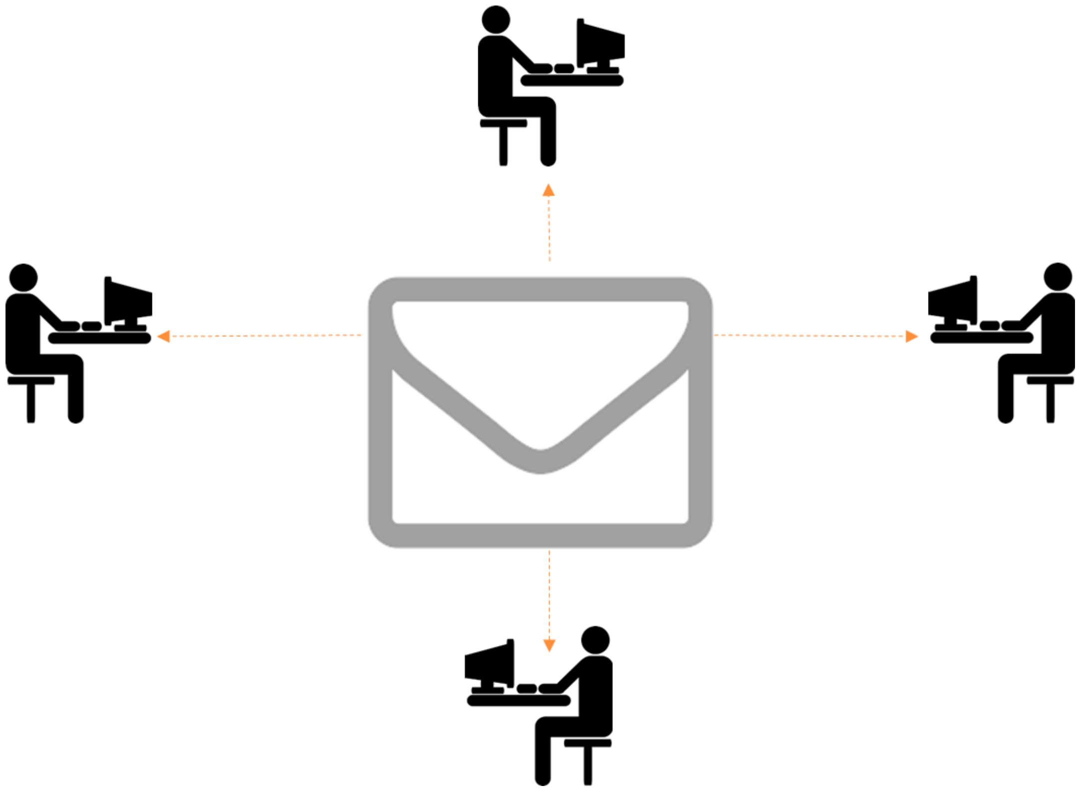
Samples and Populations

Spam email classification example



Samples and Populations

Spam email classification example



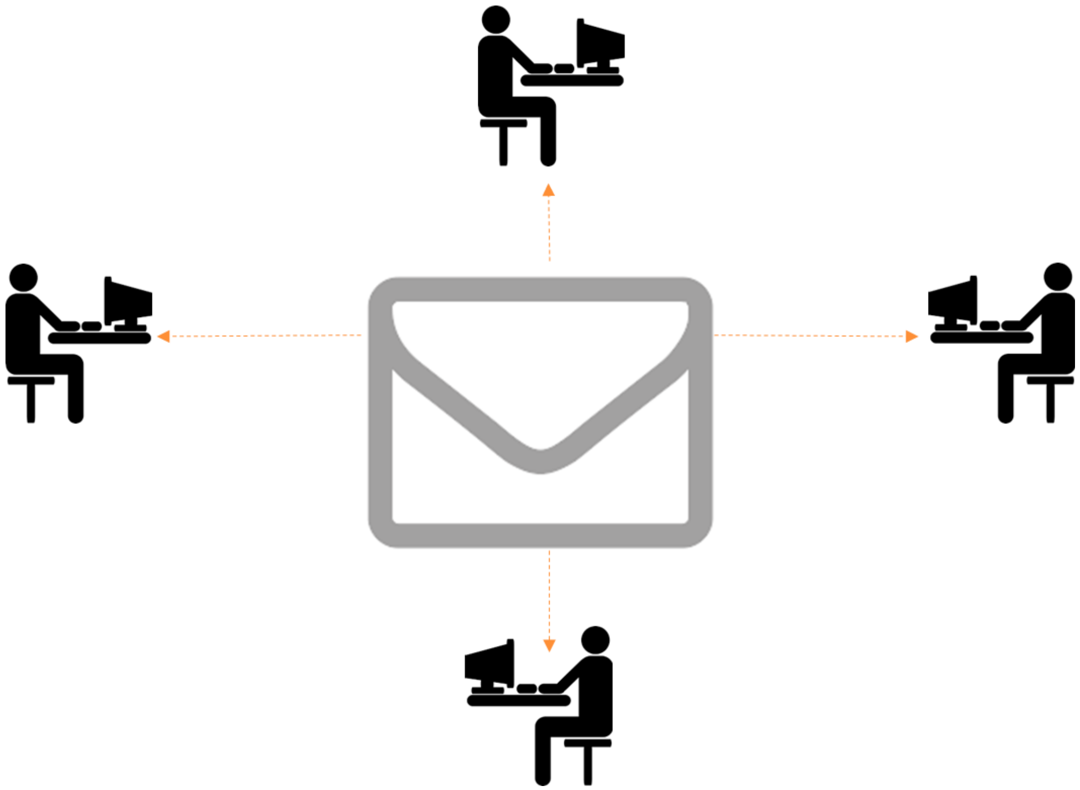
- **Unfeasible**: Collecting and training an algorithm to classify all the data
- **Feasible**: Using a sample of emails as a training dataset

Example

- Credit default dataset
- 29000 credit card users
- Data is a sample of customers

Samples and Populations

Spam email classification example



- Computers utilize sample data to learn algorithm that can predict a specific response
- Litmus test: Applying a trained classifier to unseen or non-sample data
- E.g.: Utilize a classifier trained on 1000 sample emails, to classify previously unseen data
- Viable machine learning models are those that can classify unseen or new data

Samples and Populations

- **Machine learning models' Predictive Problem Objectives**:
  - ❑ To perform well on a training data set
  - ❑ To generalize unseen data sets

- The perfect model (100% success rate with training unseen datasets) is a myth
- Models are subject to a bias-variance trade-off

Bias Variance Tradeoff

- **Bias**=How well an algorithm performs on a training set
- **Variance**=How well the algorithm will perform on unseen data

Generalization

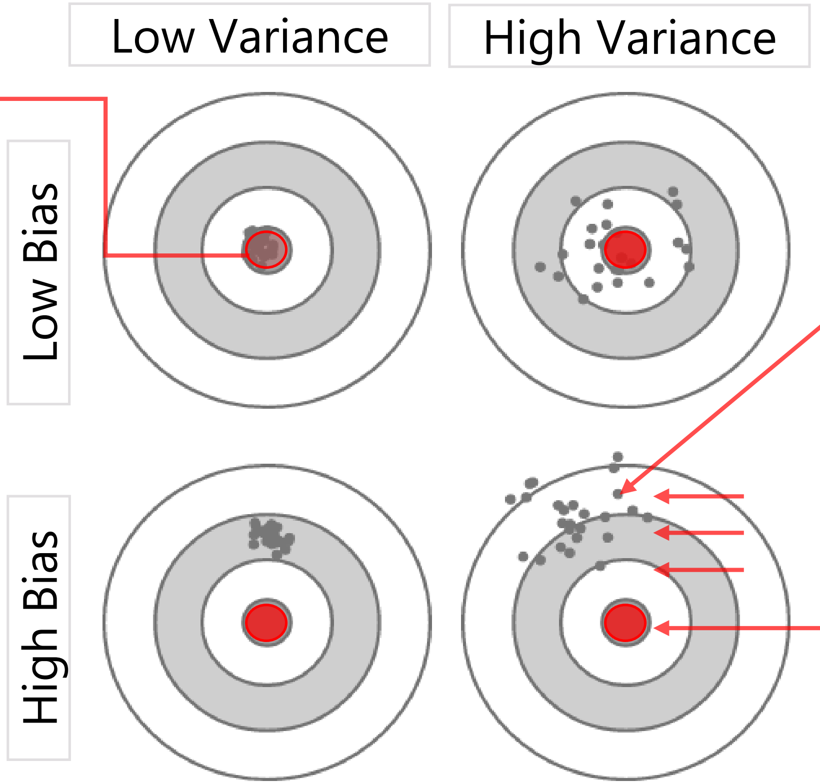
The overfitting problem: When a complicated model fits the training data perfectly but does a terrible job predicting new observations

Generalization

- A model that overfits sample data have low bias but very high variance
- A model that underfits sample data has a high bias (and likely low variance)

Generalization

- A point on the target represents a prediction
- Point placement denotes how far we are off with respect to the true value



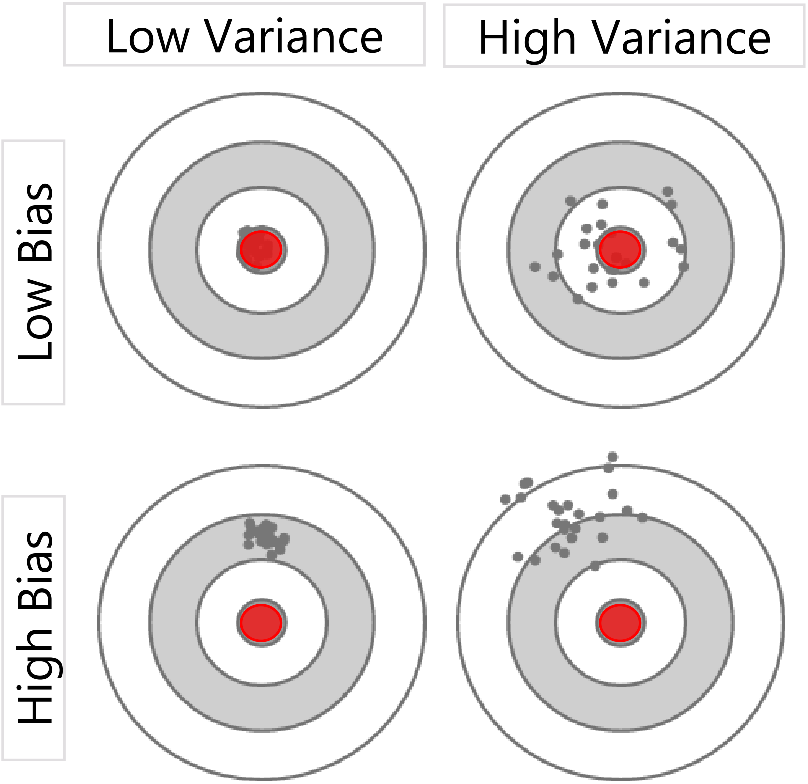
Quality of predictions diminishes the further away from the centre you get

Model that predicts the correct values

http://scott.fortmann-roe.com/docs/BiasVariance.html

Generalization

- Different cloud characteristics indicate the bias-variance trade-off
- A model with a low bias and low variance is desirable



http://scott.fortmann-roe.com/docs/BiasVariance.html

Generalization

Generalization: A model's capacity to perform well on unseen cases

- Must set thresholds
- Only deploy models with a generalization metric that is above threshold value
- The bias-variance trade-off thumb rule: Simpler models tend to generalize better

Some suggestions

- Browse through bias-variance trade-off material online
- Always keep the bias-variance trade-off principle in mind when designing a regression or classification model
- The more complex a model, the more poorly it performs on unseen data

More On Generalization

Building models that generalize well

- Split data into 3 parts train, test & validate (**80-10-10** or **70-20-10**)
- Estimate model parameters on train
- Using these calculate error on test
- Compare errors on test to select final model

More On Generalization

Building models that generalize well

- Once the final model has been selected, predict the validate split and obtain the error
- The error measure computed on the validate split is based on unseen data
- The model was trained on the train split and compared with other models on the test split
- The data in the validate split is completely new to the model
- Therefore, a performance measure on this split functions as a proxy for the model's real world performance
- Other methods of determining error measures that reflect performance on unseen data: K Fold Cross Validation

Bias Variance Trade off: Data Partitioning & Cross Validation

Ensure that

- The model works with the training set and generalizes well to unseen data
- You have a dataset that the model never sees, but whose responses are known

- Randomly partition the original data: training & testing
  - ❑ 80-20 split: 80% of the data in training, 20% in testing
  - ❑ Other options:70-30 or 90-10 splits
  - ❑ Use train for exploratory analysis & model training
  - ❑ Use training data for both
  - ❑ Predict responses using testing split
  - ❑ Testing split acts as a proxy for unseen data
  - ❑ A model that does well on training and testing splits is expected to have low bias and low variance

Bias Variance Trade off: Data Partitioning & Cross Validation

Cross Validation

- Partitioning is random
- Bad training & testing split
- Testing and training splits have varying distribution of features and responses
- Testing split results do not reflect an algorithm's generalization capabilities

Bias Variance Trade off: Data Partitioning & Cross Validation

How it works

- Partition data into K groups
- Step 1: Model is trained on combined data for 2nd to Kth partition, and tested on 1st partition
- Step 2: Model is trained on combined data for 1st and 3rd till the Kth partition, leaving out partition 2
- Second partition is the testing split
- The process continues till all partitions have served as testing splits

Bias Variance Trade off: Data Partitioning & Cross Validation

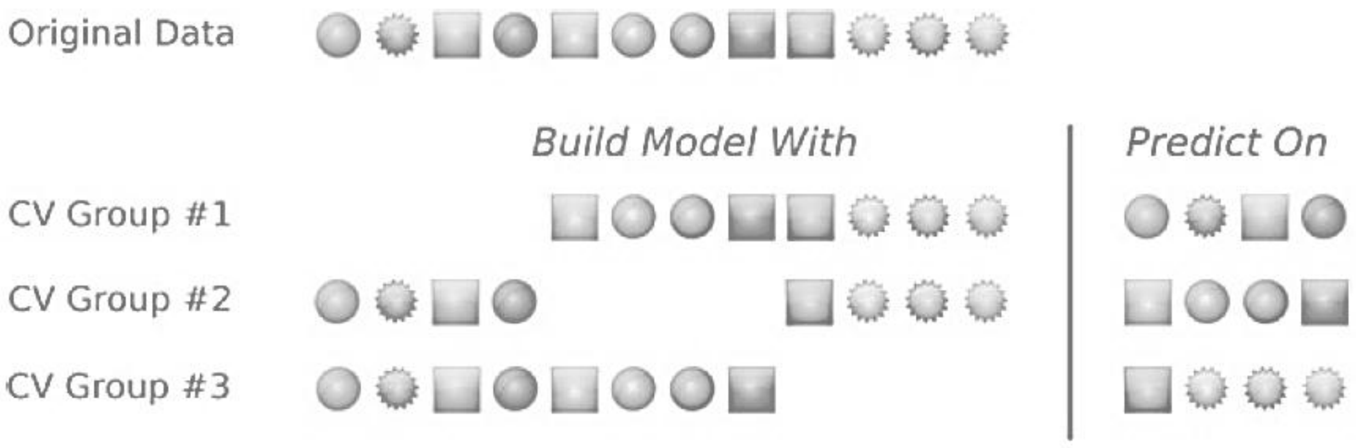
This strategy is called **K-fold cross validation**

Once done, you have a series of performance metrics for each pass of the cross-validation

Each metric is calculated on the partition that was held out as a testing split

A summary of this series of performance metrics act as a proxy for the ability of the model to generalize to unseen datasets

This process cannot be repeated multiple times



Repeated 3-fold cross validation

Image Credit: Applied Predictive Modeling, Kuhn, Johnson

Recap

- Confusion matrix
- Comparing multiple classifiers
- Samples and populations
- Bias variance tradeoff
- Generalization
- More on generalization
- Bias variance trade-off: Data Partitioning & cross validation

JIGSAW ACADEMY

THE ONLINE SCHOOL OF ANALYTICS