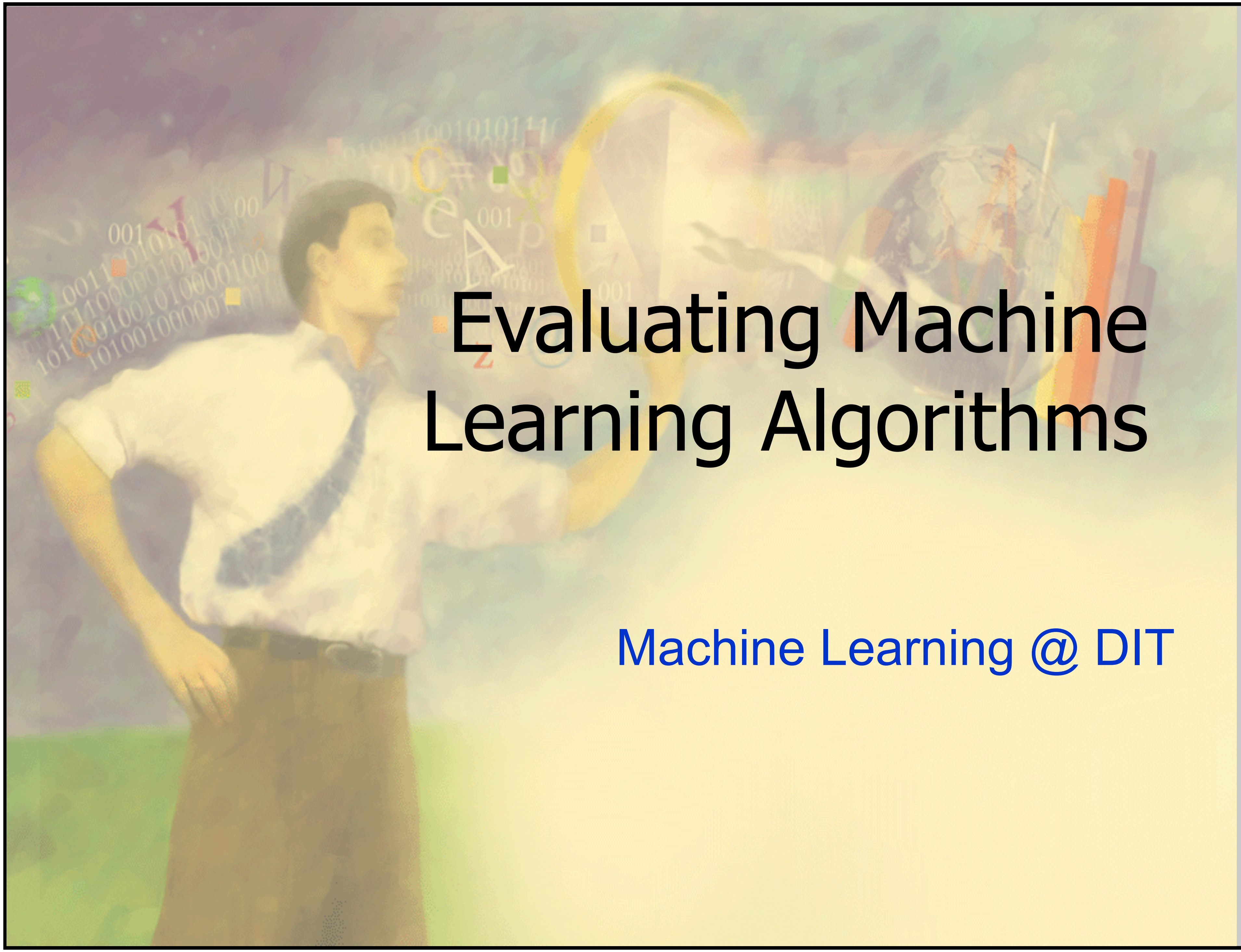


Evaluating Machine Learning Algorithms

Machine Learning @ DIT

Evaluating Classification Accuracy

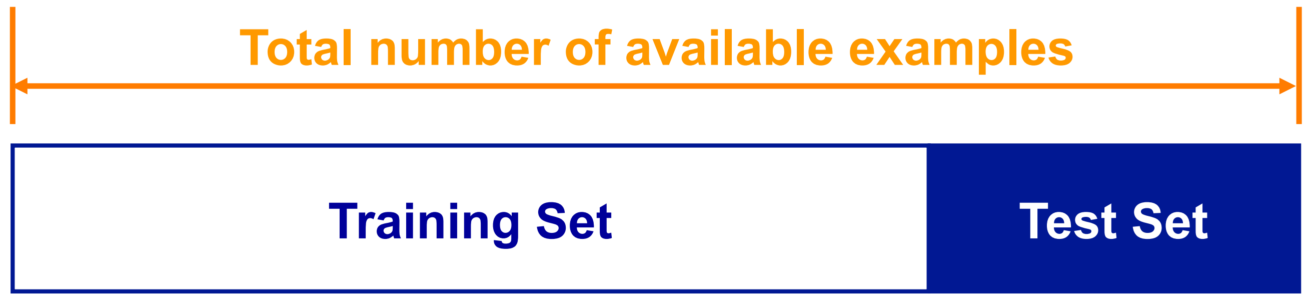
During development, and in testing before deploying a classifier in the wild, we need to be able to quantify the performance of the classifier

- How accurate is the classifier?
- When the classifier is wrong, how is it wrong?

Useful to decide on which classifier (which parameters) to use and to estimate what the performance of the system will be

Hold-Out Testing Sets

Split the available data into a *training set* and a *test set*



Train the classifier in the training set and evaluate based on the test set

Classifier Accuracy

The **accuracy** of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier

- Often also referred to as **recognition rate**
- **Error rate** (or **misclassification rate**) is the opposite of accuracy

Example

Expected	Predicted
Y	Y
N	Y
Y	Y
Y	Y
N	N
N	N
Y	Y
Y	N
N	N
Y	Y
N	Y
Y	Y
Y	N
N	N
Y	Y
N	N
Y	Y
Y	N
N	Y
N	N

False Positives Vs False Negatives

While it is useful to generate the simple accuracy of a model, sometimes we need more

When is the model wrong?

- False positives vs false negatives
- Related to type I and type II errors in statistics

Often there is a different cost associated with false positives and false negatives

- Think about diagnosing diseases

Confusion Matrix

Is a device used to illustrate how a model is performing in terms of false positives and false negatives

It gives us more information than a single accuracy figure

It allows us to think about the cost of mistakes

It can be extended to any number of classes

Confusion Matrix

Model Result

A Non-Lapsed	B Lapsed	
True Positive (TP)	False Negative (FN)	A Non-Lapsed
False Positive (FP)	True Negative (TN)	B Lapsed

Expected Result

Other Accuracy Measures

Sometimes a simple accuracy measure is not enough

$$\textbf{Model Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

$$\textbf{Missclassification Rate} = \frac{FP + FN}{TP + FP + TN + FN}$$

$$\textbf{Sensitivity}\ \text{(True Positive Rate)} = \frac{TP}{TP + FN}$$

$$\textbf{Specificity (}\text{True Negative Rate)} = \frac{TN}{TN + FP}$$

Measures Beyond Simple Accuracy

- Most classification algorithms actually give us a numeric output which we can use to go further than accuracy:
  - The Mahalanobis Distance
  - The Kolmogorov-Smirnov Distance
  - ROC Curves & Area Under the Curves
  - Gini

Example

Expected	Predicted	**Score**
Y	Y	0.99
N	Y	0.65
Y	Y	0.60
Y	Y	0.58
N	N	0.21
N	N	0.26
Y	Y	0.69
Y	N	0.35
N	N	0.17
Y	Y	0.77
N	Y	0.74
Y	Y	0.89
Y	N	0.01
N	N	0.33
Y	Y	0.53
N	N	0.36
Y	Y	0.73
Y	N	0.19
N	Y	0.63
N	N	0.23

The Mahalanobis Distance

- The mahalanobis distance measures the distance between the mean of two distributions. For example the distance between the Non-Lapsed (A) and the Lapsed (B) predictions in a retention model

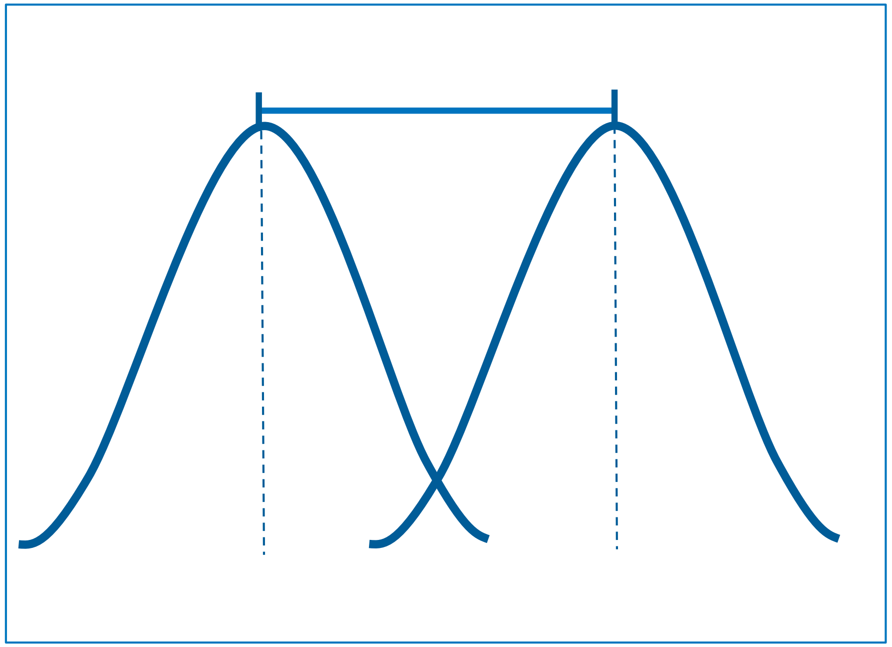
$$M = \frac{|\mu_A - \mu_B|}{\sigma}$$

- Where σ is the pooled standard deviation of the the Non-Lapsed (A) and the Lapsed (B) from their respective means

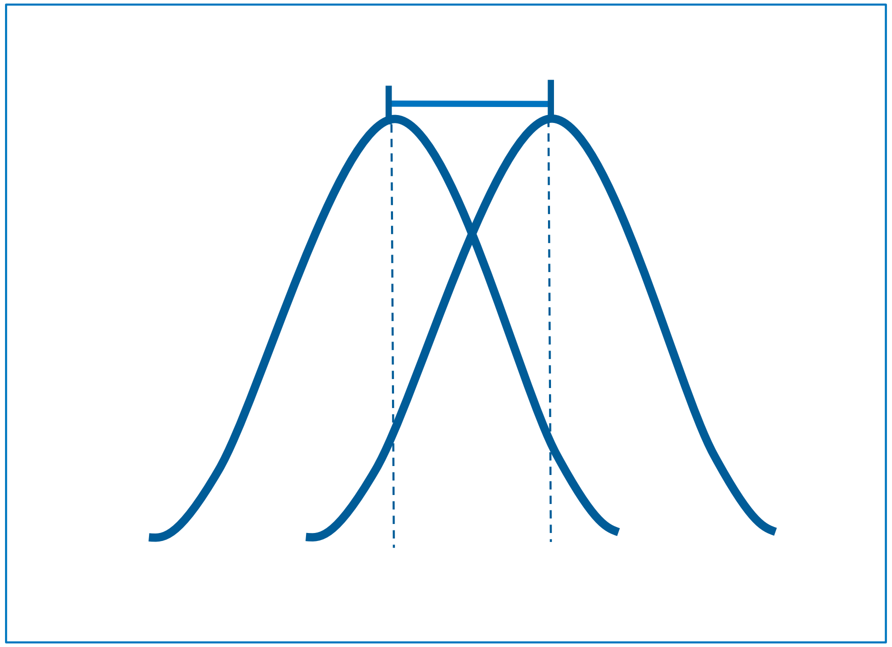
The Mahalanobis Distance

- The mahalanobis distance is a measure of separation.

Good Separation

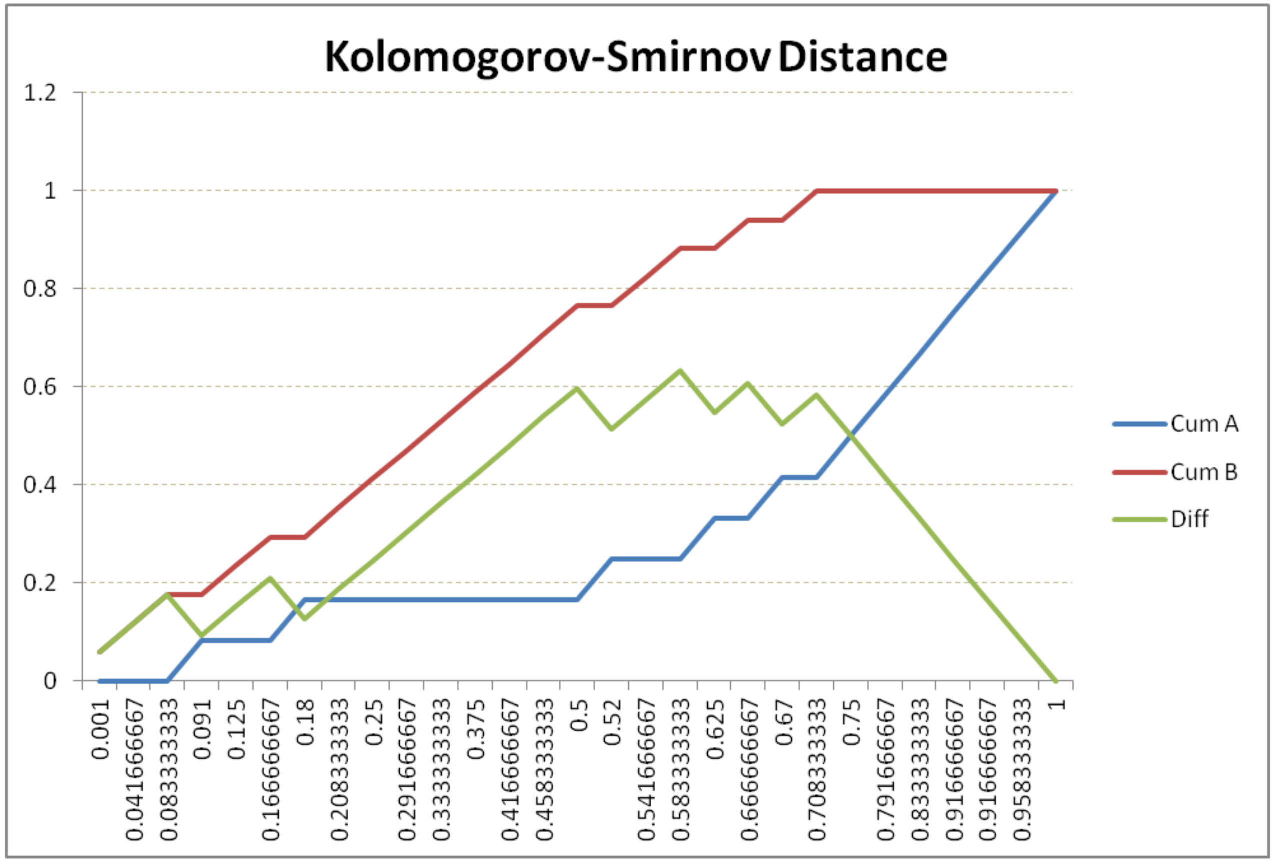


Poor Separation

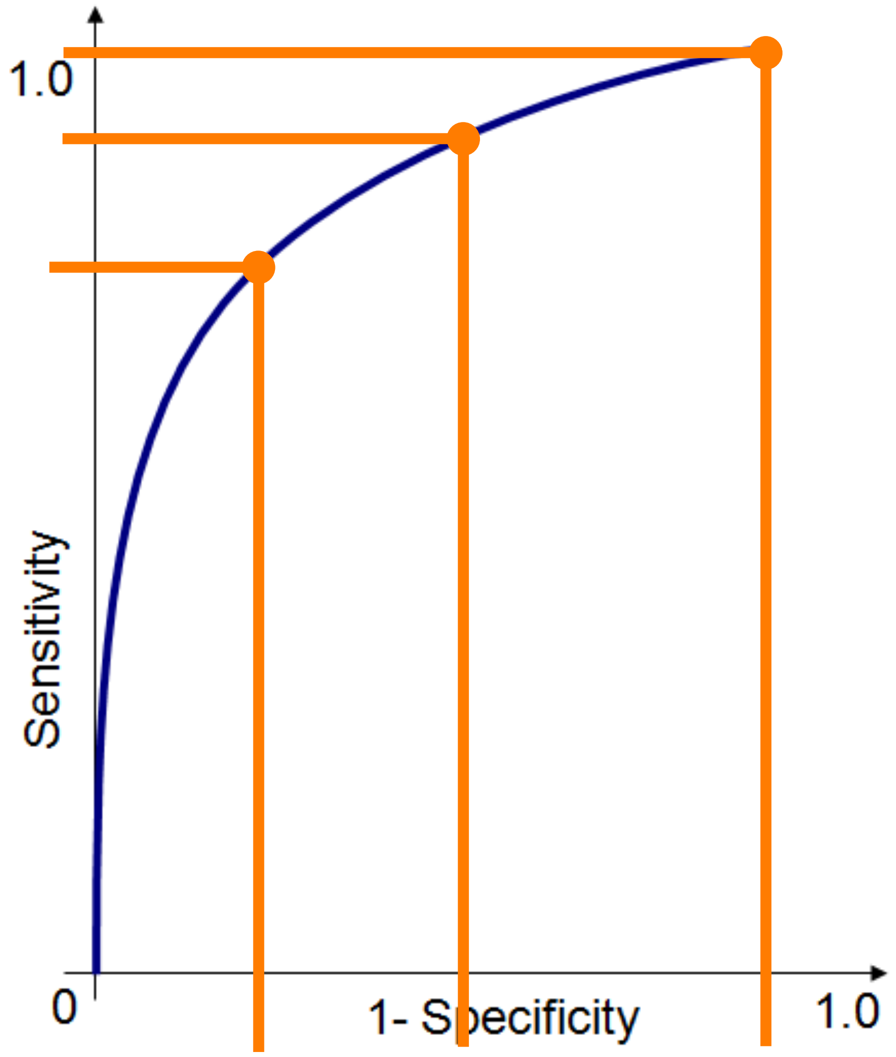


The Kolmogorov-Smirnov Distance

- Another Separation Measure
- Measures the distance between the model outcome distribution.

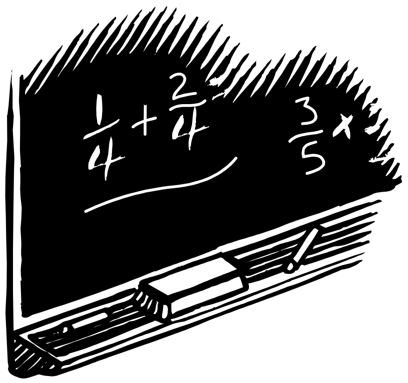


ROC Curves (cont…)

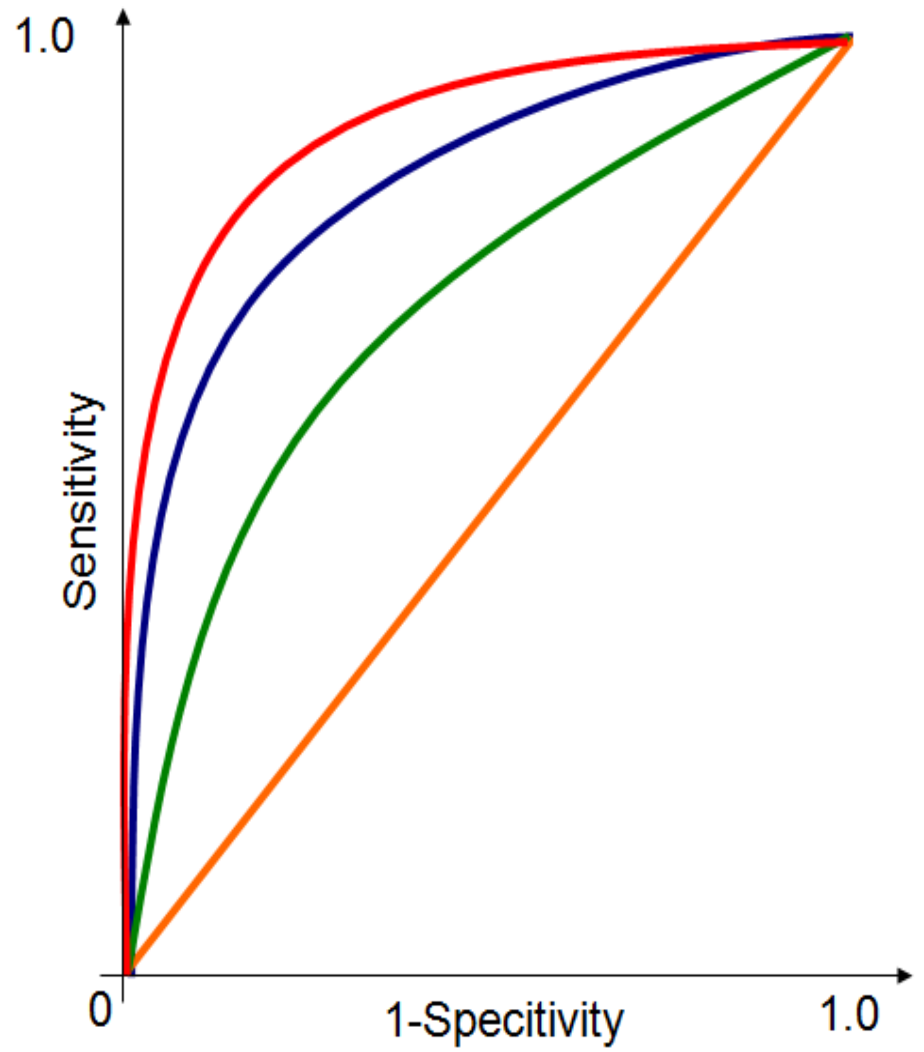


Consider how the relationship between true positives and false positives can change

We need to choose the best operating point



ROC Curves (cont…)



- ROC curves can be used to compare models
- The greater the area under the curve (AUC) the more accurate the model
- ROC index is the area under the ROC curve

Gini- Statistic

- The Gini-Statisic can be calculated from the ROC Curve:

$$\text{Gini} = 2 * \text{AUC} - 1$$

Where the AUC is the area under the ROC curve.

Other Measures

There are any number of other measures that are used to measure model accuracy:

- Cumulative lift
- Akaike's Information Criterion
- Schwarz's Bayesian Criterion

Individuals tend to have their own opinion on which one is best - whatever your customer/ boss wants!

Evaluating Continuous Targets

- When dealing with continuous targets different evaluation measures are required
- The most common is **average squared error** (**ASE**)
- To calculate ASE first calculate the sum of squared errors (SSE) for the whole test set and then divide by the number of examples in the test set, N

$$ASE = SSE/N$$

Types of Validation

- Use a hold-out sample (in-sample testing)
- Validate using an sample not from the training period (Out-of-time)
- Validate using an sample that is selected from a different population than that used to build the model (Out-of-sample)

Evaluating Models

How we do this depends on how much data is available

If there is unlimited data available then there is no problem

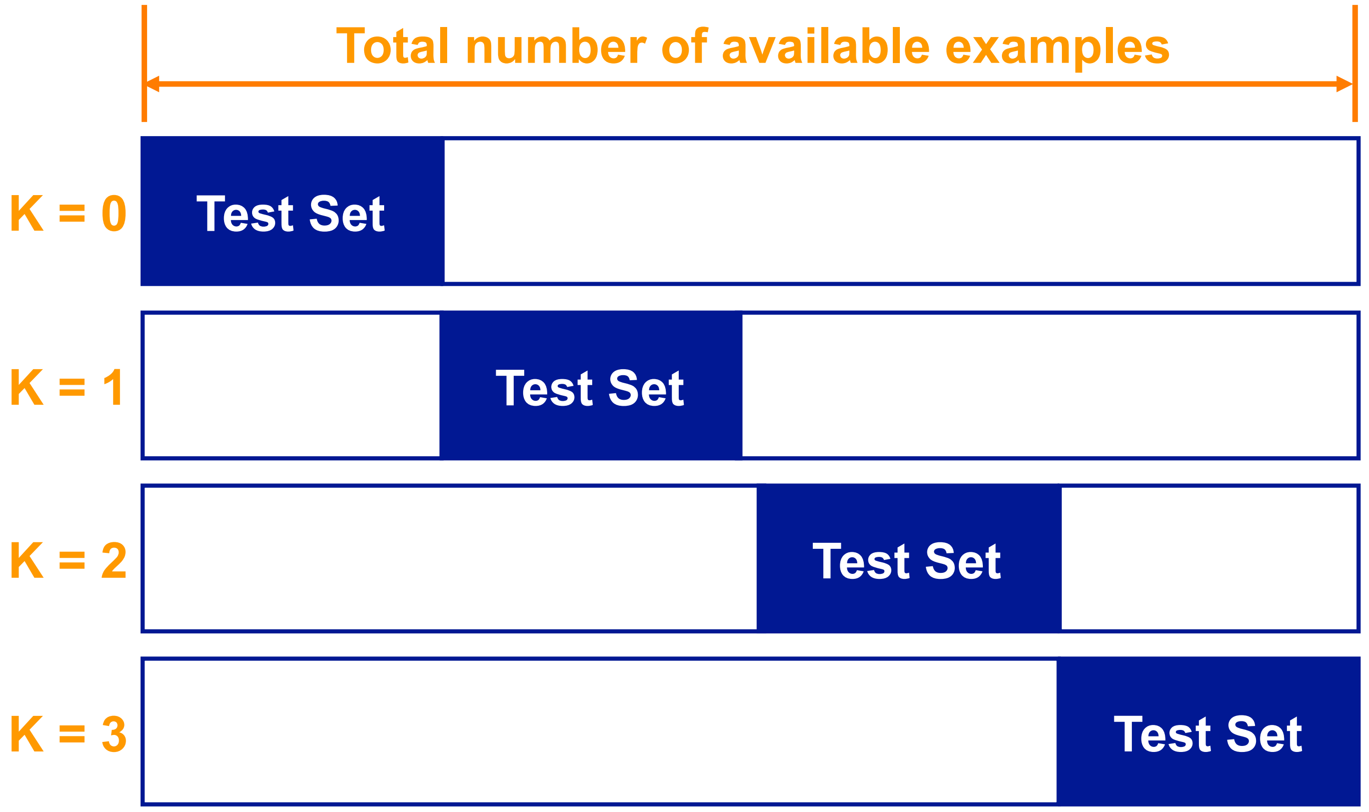
Usually we have less data than we would like so we have to compromise

- Use hold-out testing sets
- Cross validation
  - K-fold cross validation
  - Leave-one-out validation
- Parallel live test

K-Fold Cross Validation

Divide the entire data set into k *folds*

For each of k experiments, use kth fold for testing and everything else for training



K-Fold Cross Validation (cont…)

The accuracy of the system is calculated as the average error across the k folds

The main advantages of k-fold cross validation are that every example is used in testing at some stage and the problem of an *unfortunate split* is avoided

Any value can be used for k

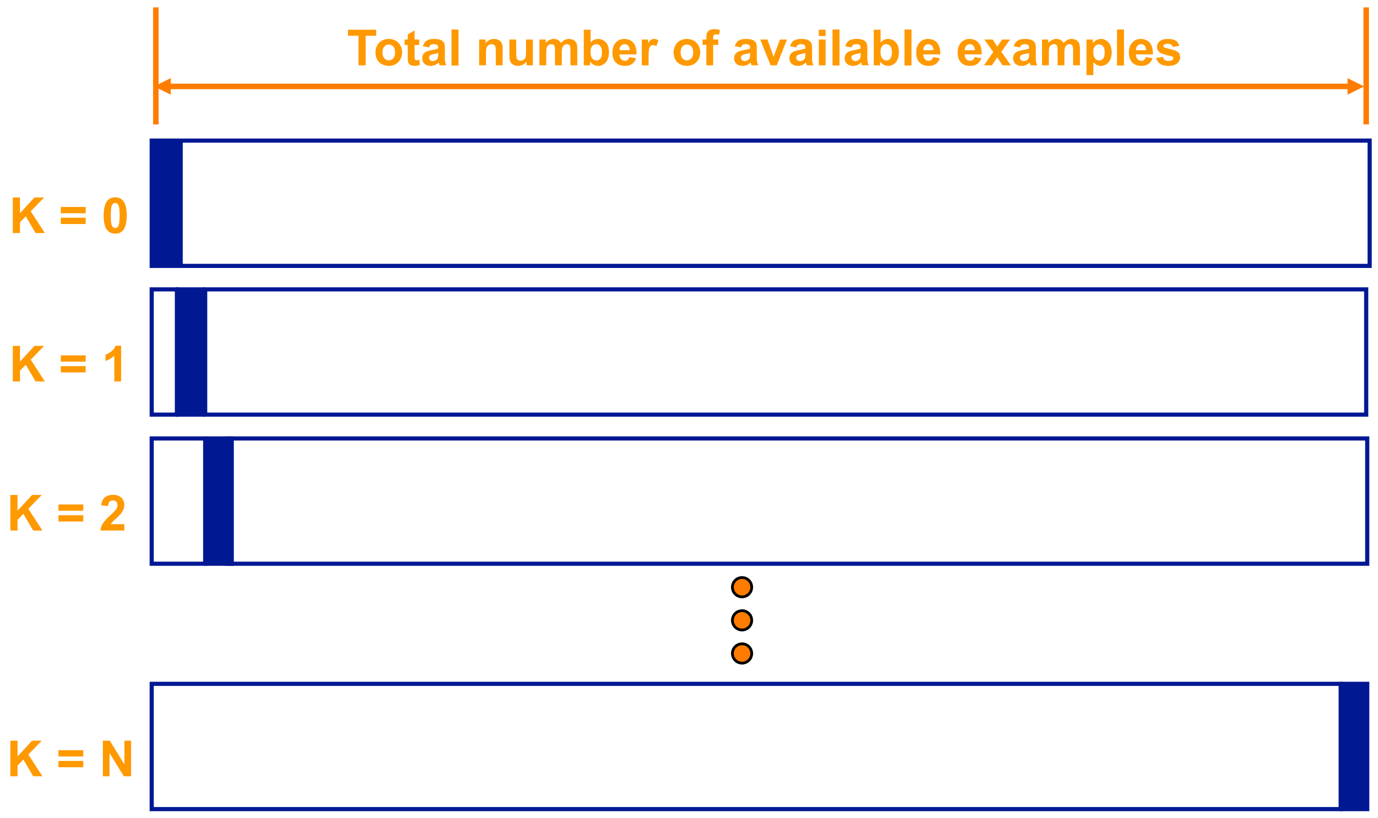
– 10 is most common

– Depends on the data set

Leave-One-Out Cross Validation

Extreme case of k-fold cross validation

With N data examples perform N experiments with N-1 training cases and 1 test case



Ongoing Validation

- Once your model have been deployed you need to check to make sure that the model is still behaving as it was intended
- There are two main areas that you need to consider:
  - Model Accuracy
  - Population Stability

Model Accuracy

- On an ongoing basis you need to check the model is performing as well as it did during development
- Run your accuracy measures (Gini/KS) on an ongoing basis and compare them to same the measures calculated during development
- If these statistics are deteriorating, you will need to investigate and either re-balance (fine -tune) the model or do a full rebuild.

Population Stability

- You need to compare the distributions of the model results from development and the most recent model run
- You need to compare the distributions of the model inputs from development and the most recent model run
- There are a number of methods for doing this:
  - Stability Index Report (Used widely in credit scoring)
  - Kolmogorov-Smirnov measure

Questions?

?