Evaluation of Classification Models

Data Science Dojo

Agenda

- Metrics for Evaluation
  - Confusion Matrix
  - Accuracy, Precision, Recall, F1 measure
- Building Robust Machine Learning Models
  - Bias/Variance Tradeoff
- Methods of Evaluation
  - Cross Validation
  - ROC Curve

The Limitations of Accuracy

- Consider a 2-class problem:
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If the model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading!

METRICS FOR EVALUATION

Confusion Matrix

	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class=No
Class=Yes	a	b
Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Confusion Matrix

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{a + d}{a + b + c + d}$$

Precision

$$p = \frac{TP}{TP + FP} = \frac{a}{a + c}$$

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

Recall/Sensitivity

$$r = \frac{TP}{TP + FN} = \frac{a}{a + b}$$

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

F1-Score

$$F1 = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Harmonic mean of precision and recall

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

Specificity

$$S = \frac{TN}{FP + TN} = \frac{d}{c + d}$$

Useful if negative class more important positive

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

WILL MY MODEL BETRAY ME?

Is My Model Really Good?

- My model shows an accuracy of 90% in the **training environment**
- Would the model be 90% accurate in **production environment**?

Perils of Overfitting

Train/Test partition is not enough

Labelled Data

Training Data	Blind Holdout Data
70%	30%

Blind Holdout Data

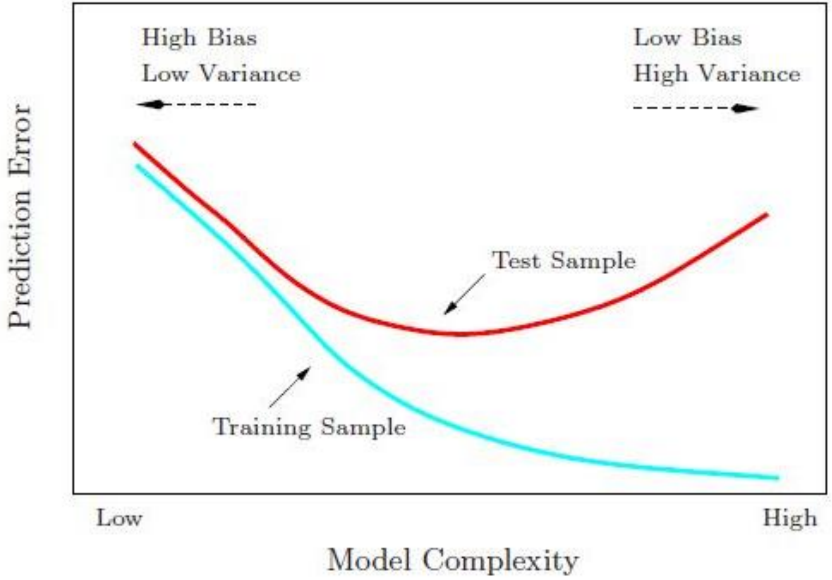
- The person building the model has no access to the holdout data set
- Why do we need to lock this away?

Overfitting

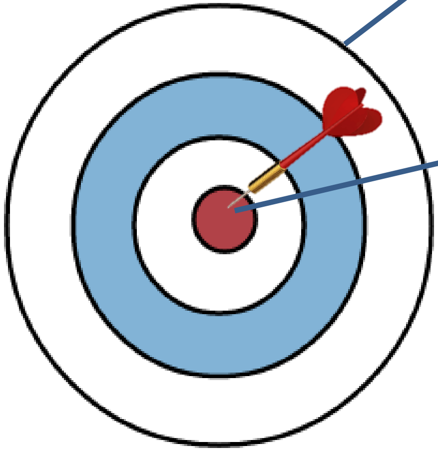
- The gravest and most common sin of machine learning
- Overfitting: learning so much from your data that you memorize it.
  - You do well on training data
  - But don’t do well (or even fail miserably) on test data

Bias/Variance Tradeoff

- You can beat your data to confess anything



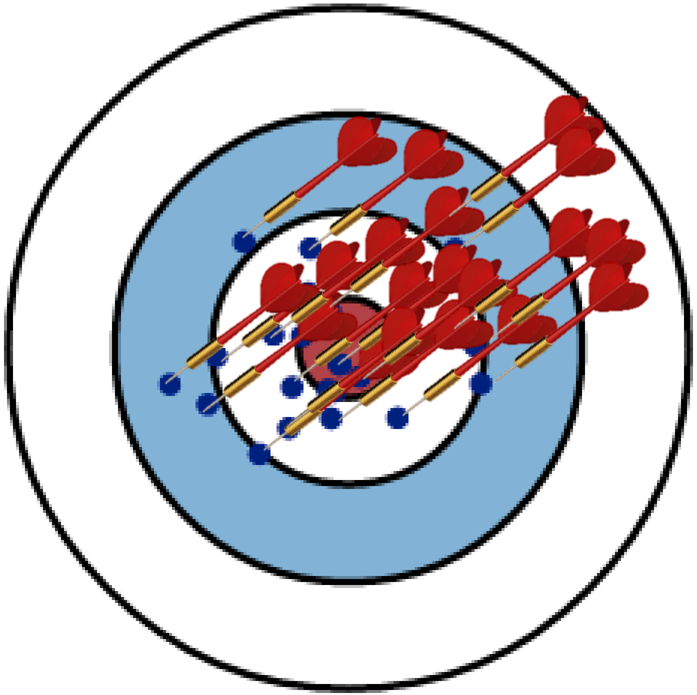
Bias/Variance Trade-off



Each dartboard represents a model

Bullseye is the theoretical best performance (accuracy, precision, recall or something else)

Bias/Variance Trade-off

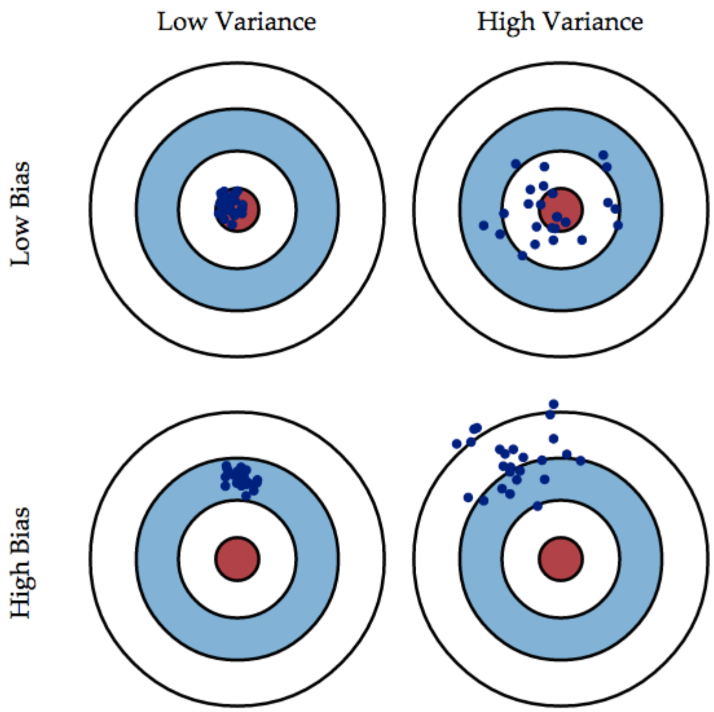


Try several random variations of the test set

Each dart represents a random variation of the test set.

The generation of random numbers is too important to be left to chance.

Bias/Variance Trade-off



METHODS OF EVALUATION

Holdout Set

- 70% for training
- 30% for testing
- 60/40 and 50/50 also possible
- Repeated holdout: Apply 70/30 many times.

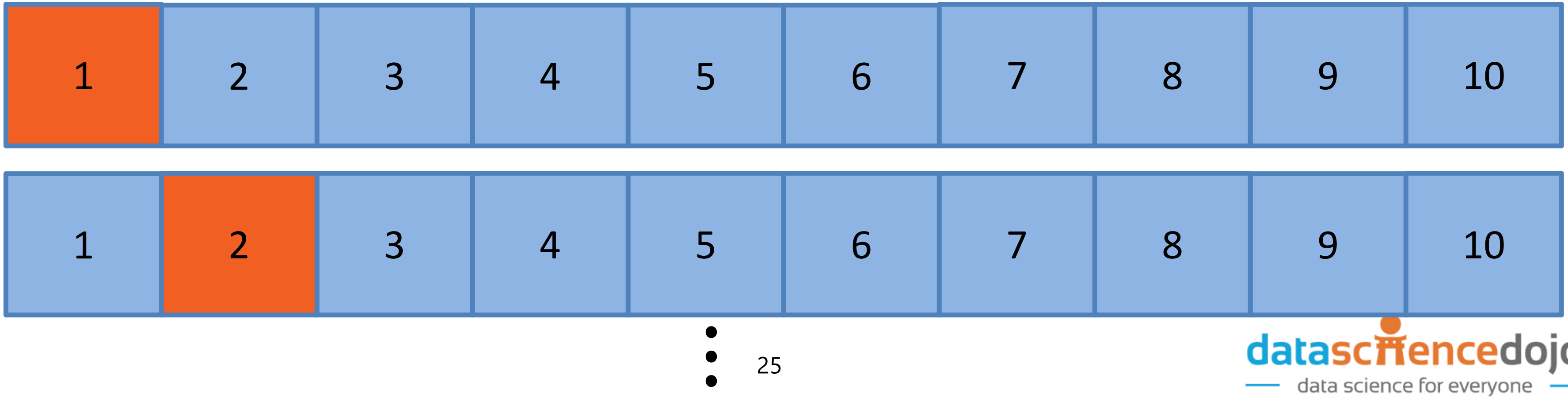
Cross validation

- Very useful tool for evaluation
- Split dataset into random partitions
  - Stratified sample if appropriate

1	2	3	4	5	6	7	8	9	10

Cross validation

- Train model on 2-10, test on 1
- Train (new) model on 1,3-10, test on 2
- Repeat 10 times



Cross validation

- Result: 10 models, labeled by test partition
- Measure bias and variance
- Detect overfitting

	1	2	3	4	5	6	7	8	9	10	Avg	Std
Accuracy	.84	.86	.83	.85	.79	.84	.86	.85	.89	.83	.844	.026
Precision	.79	.78	.81	.79	.85	.76	.82	.71	.75	.76	.782	.040
Recall	.75	.83	.76	.83	.65	.80	.74	.76	.77	.79	.768	.052

Stratified Sampling

- Used with cross validation or holdout set
- Ensures that all partitions have fixed ratio of classes
  - Same ratio as training set
  - If training set is 5% class 1, 95% class 2, so is each partition
- Use with very uneven class distributions
- Avoid when class distribution isn't constant

Bootstrapped Sampling

- Sampling with replacement
- We will discuss this in detail when we get to ensemble methods

ROC CURVE

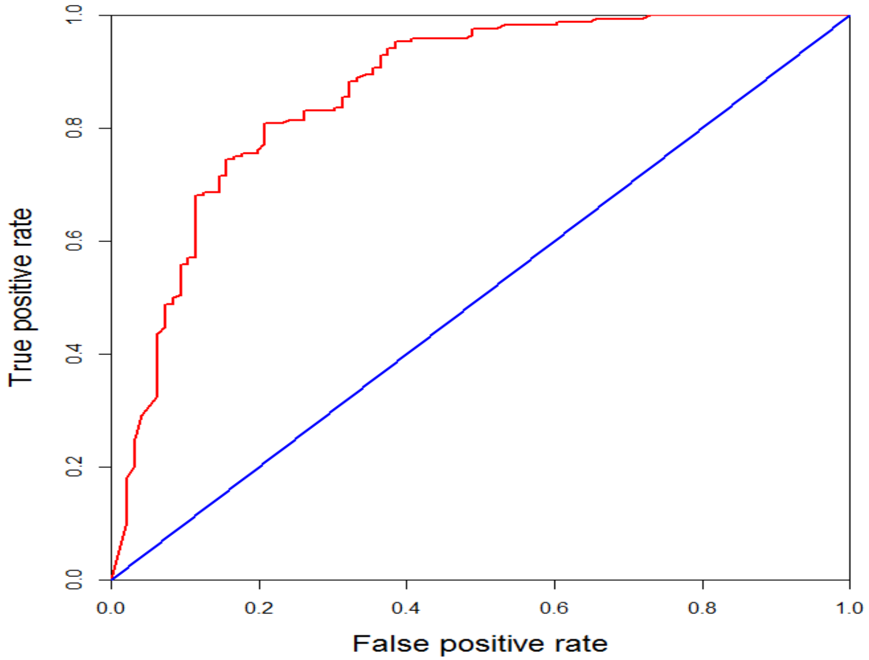
Controlling Precision and Recall

- What if probabilities are reported?
- Threshold
  - The probability value which separates positive predictions from negative predictions
  - Adjusts class label metrics

Pid	**Prediction**	**T=0.5**	**T=0.25**	**T=0.75**
2	.95	Survived	Survived	Survived
3	.86	Survived	Survived	Survived
5	.02	Dead	Dead	Dead
7	.15	Dead	Dead	Dead
13	.48	Dead	Survived	Dead
14	.35	Dead	Survived	Dead
21	.12	Dead	Dead	Dead
24	.01	Dead	Dead	Dead
34	.74	Survived	Survived	Dead
54	.63	Survived	Survived	Dead

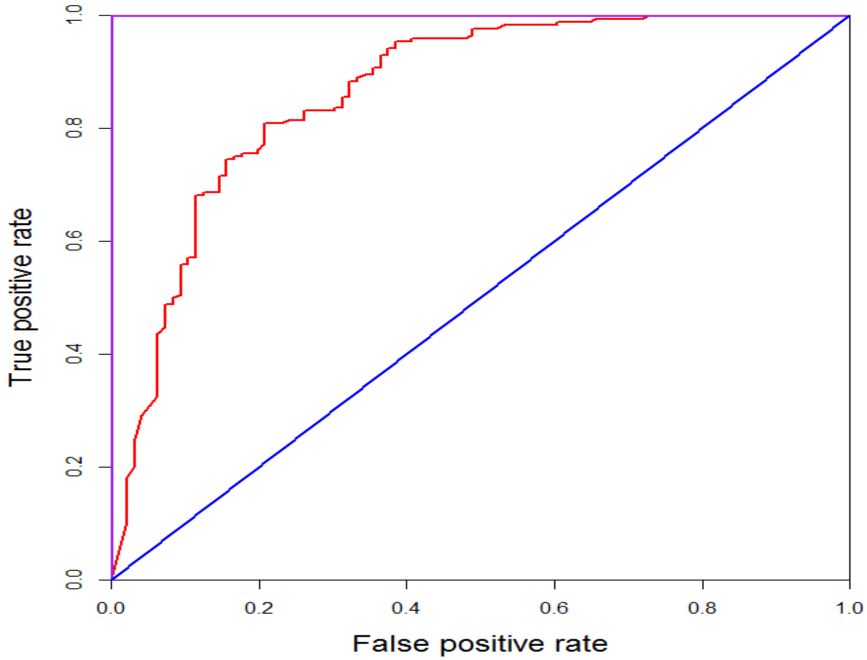
ROC(Receiver Operating Characteristic)

- Developed to analyze noisy signals
- TP on the y-axis vs FP on the x-axis
- Plot points for different threshold values
- Curve represents quality of model *independent* of threshold



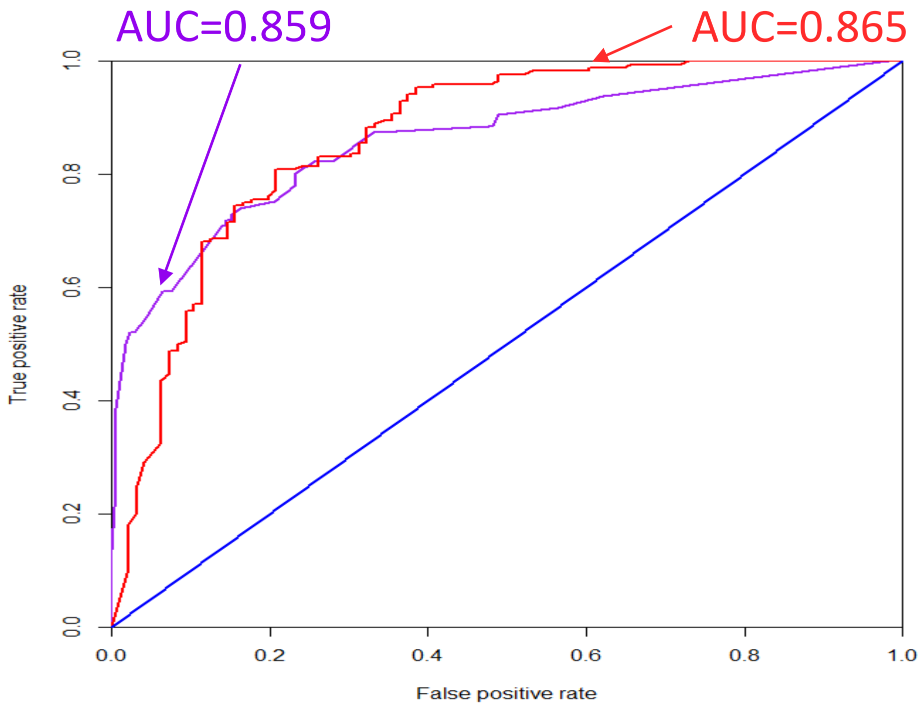
ROC Curve

- Ideal curve (purple)
  - 100% True Positives
  - 0% False Positives
- Random chance (blue)
  - Worst case
- Below diagonal line?
  - Prediction is opposite of the true class



datasciencedojo
data science for everyone

Using ROC for Model Comparison



- No model consistently outperforms the other
  - Purple is better at low thresholds
  - Red is better at high thresholds
- Area Under ROC Curve (AUC)
  - Calculate the area under the curves
  - Compare models directly

QUESTIONS