Evaluation of Classification Models

Classifier Evaluation

- Metrics
  - Class Label Based
  - Probability Based
- Methods
  - Bias-Variance Tradeoff
  - Techniques
  - Model Comparison
- Parameter Tuning

Classifier Evaluation

- **Metrics**
  - Class Label Based
  - Probability Based
- Methods
  - Bias-Variance Tradeoff
  - Techniques
  - Model Comparison
- Parameter Tuning

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!

Measuring model performance

- Do you always want your model to be accurate?
- What other metrics are there?

Class Label Metrics

- We will focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

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

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Class Label Metrics

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

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

- Most widely-used metric

Accuracy Alternatives

- Precision
  - $\frac{TP}{TP+FP} = \frac{a}{a+c}$
  - Sensitive to false positives
- Recall/Sensitivity
  - $\frac{TP}{TP+FN} = \frac{a}{a+b}$
  - Sensitive to false negatives
- F1-score
  - $\frac{2rp}{r+p} = \frac{2a}{2a+b+c}$
  - Harmonic average of precision and recall

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

- Specificity
  - $\frac{TN}{FP+TN} = \frac{d}{b+d}$
  - Useful if negative class more important positive

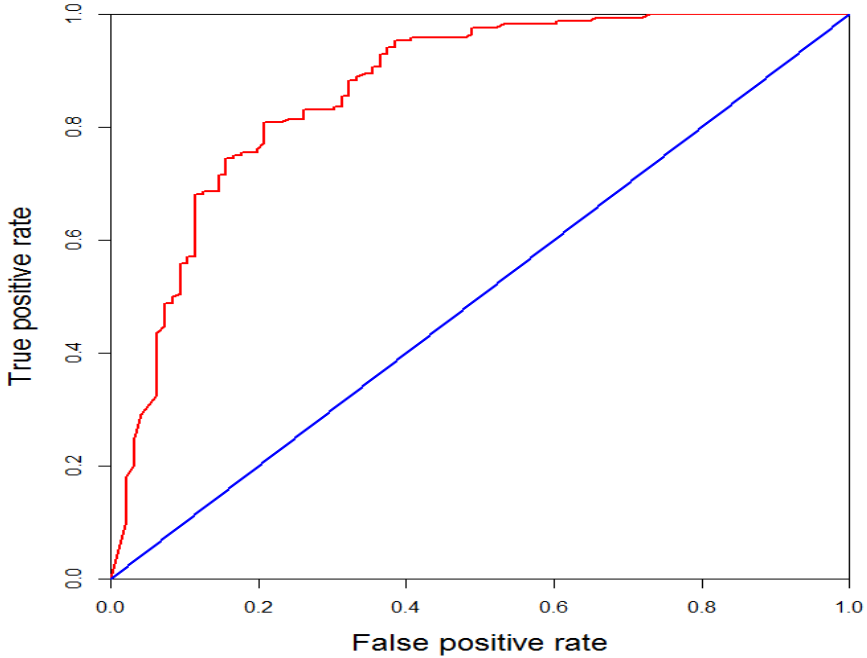
Probability-based Metrics

- 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

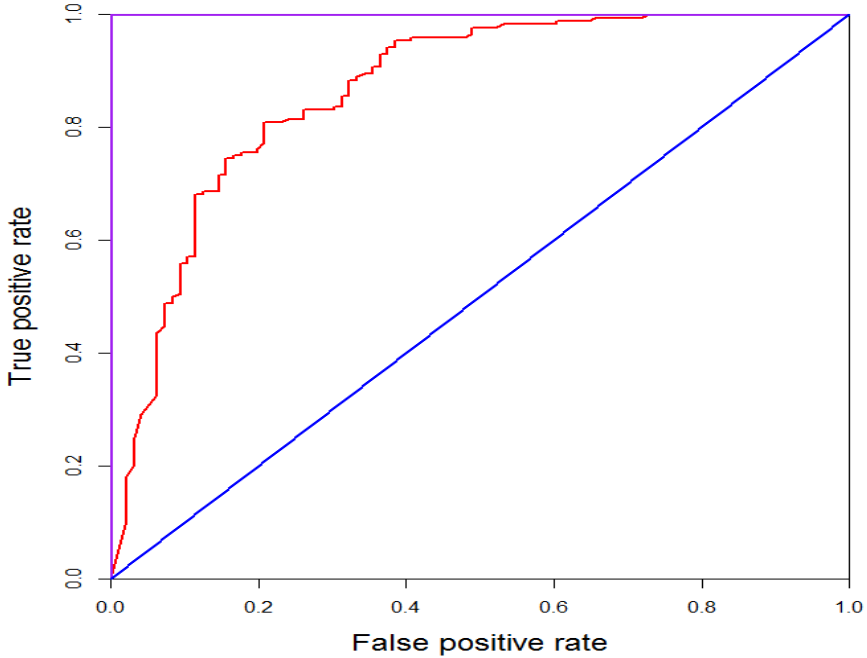
ROC (Receiver Operating Characteristic)

- 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

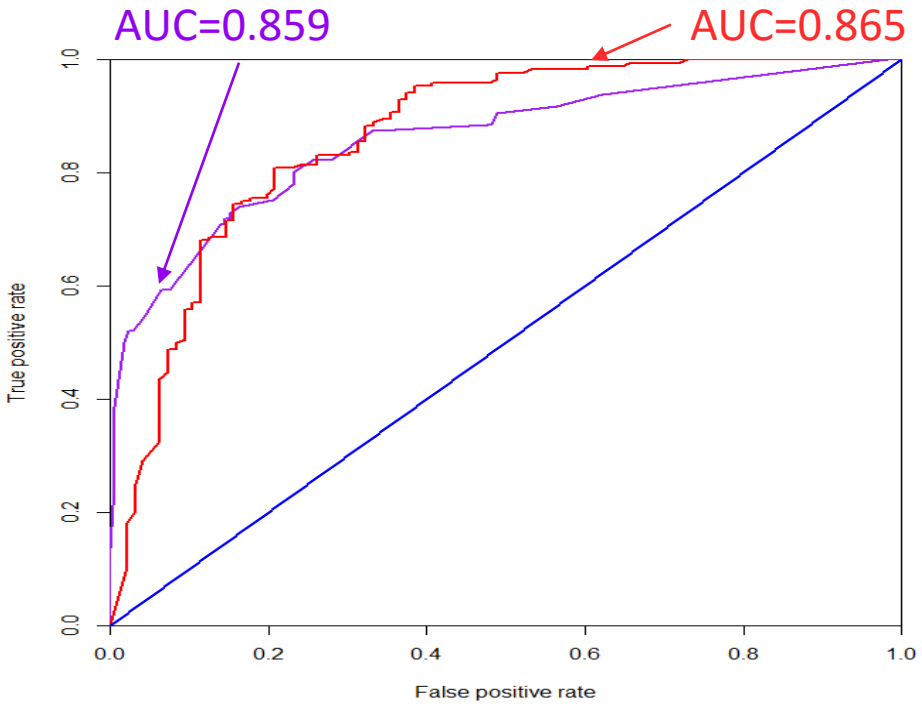


ROC Curve

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



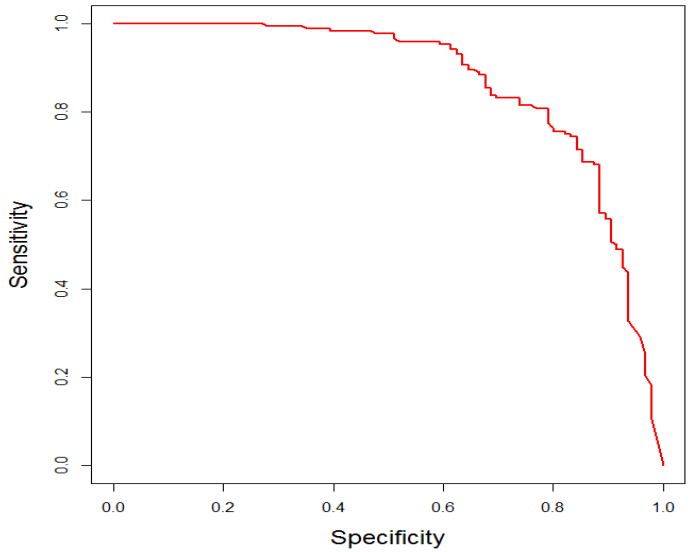
Using ROC for Model Comparison



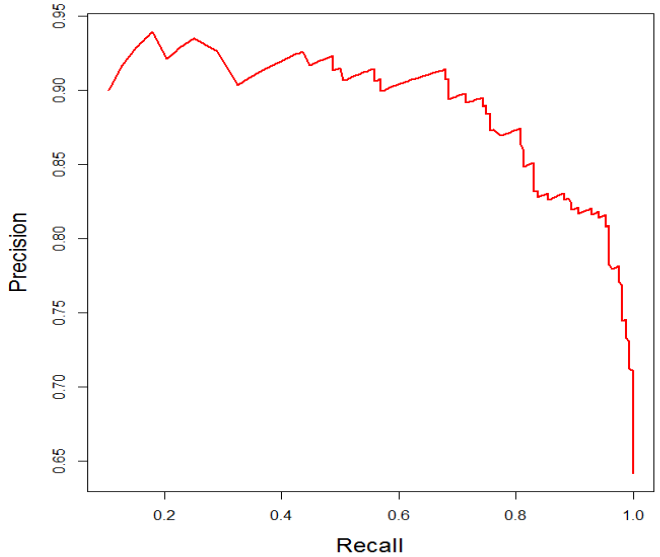
- No model consistently outperforms the other
  - Purple is better at low thresholds
  - Red is better at high thresholds
- Area Under ROC Curve (AUC)
  - Calculate the area under the curves
  - Compare models directly

Other Threshold Curves

Sensitivity/Specificity



Precision/Recall



Multiclass Metrics

- Averaged accuracy, precision, recall, F1
- Multiclass log loss
  - $logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(p_{i,j})$
  - For N test examples and M target levels, where $y_{i,j}$ is 1 if $y_i = j$ and 0 otherwise and $p_{i,j}$ is the predicted probability that object i is of class j

Log Loss Example

True Label	Setosa Prob	Versicolor Prob	Virginica Prob
Setosa	0.972	0.026	0.002
Versicolor	0	0.74	0.26
Versicolor	0.004	0.936	0.06
Virginica	0	0.61	0.39
Virginica	0	0.108	0.892

- $\frac{1}{5}[\log(0.972) + \log(0.74) + \log(0.936) +$

Classifier Evaluation

- Metrics
  - Class Label Based
  - Probability Based
- **Methods**
  - Bias-Variance Tradeoff
  - Techniques
  - Model Comparison
- Parameter Tuning

WILL MY MODEL BETRAY ME?

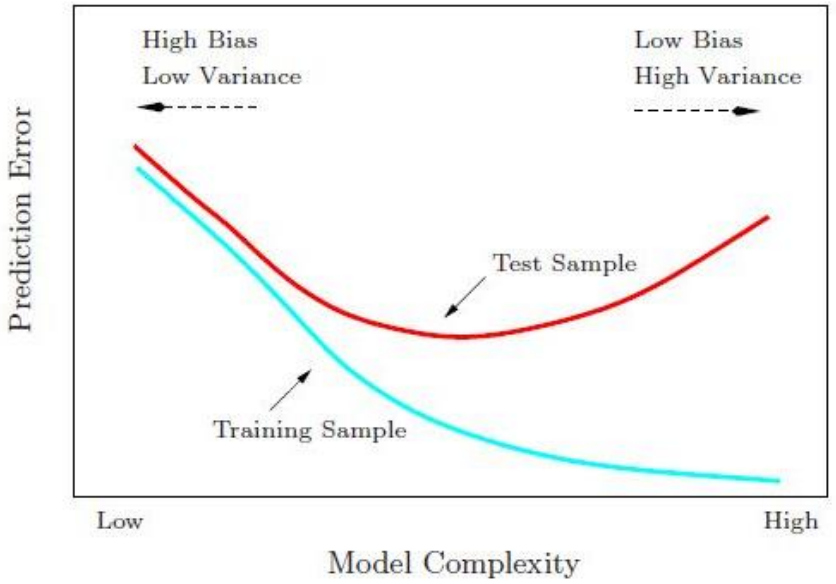
Perils of Overfitting

Overfitting

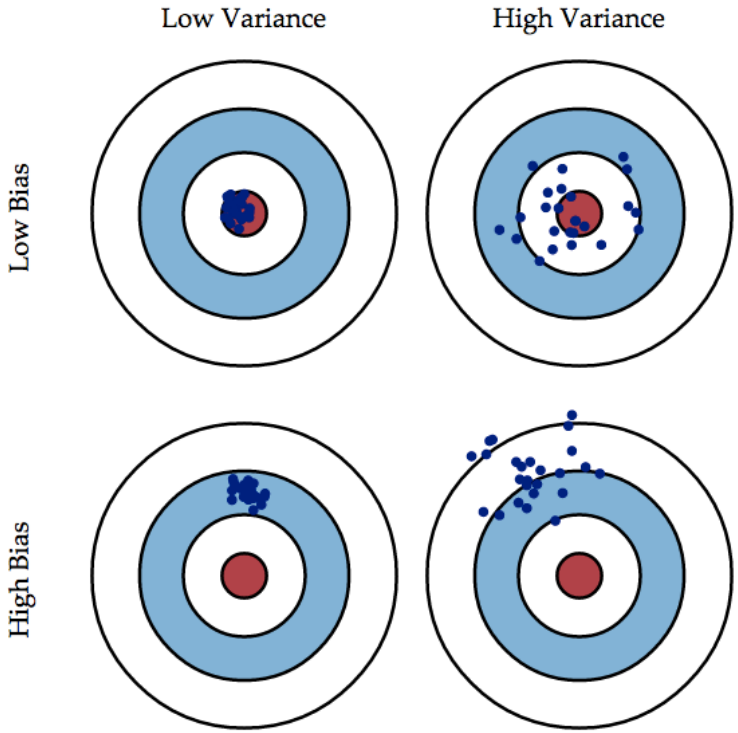
- The gravest and most common sin of machine learning
- Overfitting: learning so much from your data that you memorize it.
  - You do well on training data
  - But don’t do well (or even fail miserably) on test data

Bias/Variance Tradeoff

- You can beat your data to confess anything



Bias/Variance Tradeoff



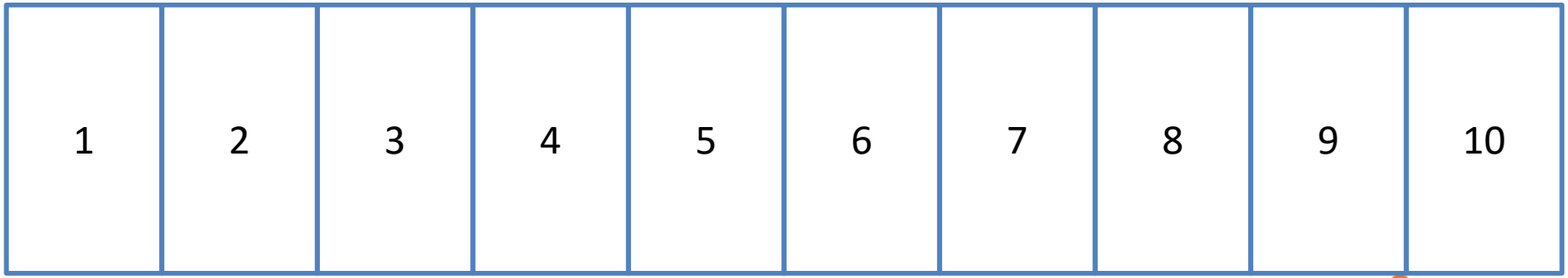
Methods of Evaluation

- Holdout Set
- Cross validation
- Random subsampling
  - "Repeated holdout"
- Stratified sampling
- Bootstrap
  - Sampling with replacement
  - Discuss later (ensemble)

- We've discussed holdout sets, but not the rest.
- Not limited to just one, apply multiple

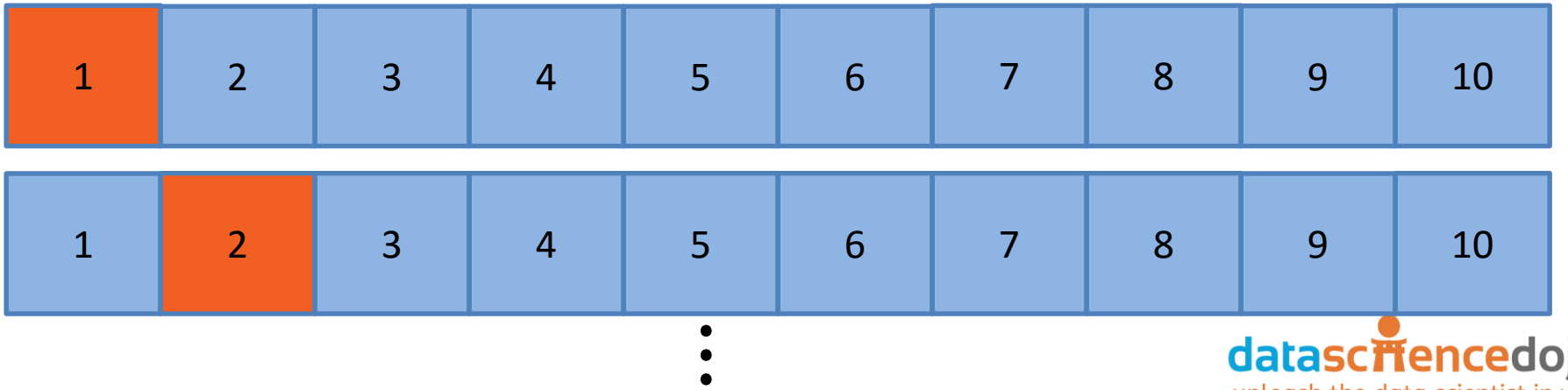
Methods: Cross validation

- Most powerful tool for evaluation
- Split dataset into random partitions
  - Stratified sample if appropriate



Methods: Cross validation

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



Methods: 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

Methods: Repeated Holdout

- Not restricted to 10 partitions, 1 test
- Repeated 70/30 split: 10 partitions, 3 test
- Can go as far as 1 test row each iteration
  - "Leave one out" validation
  - Bad idea!

Methods: Stratified Sampling

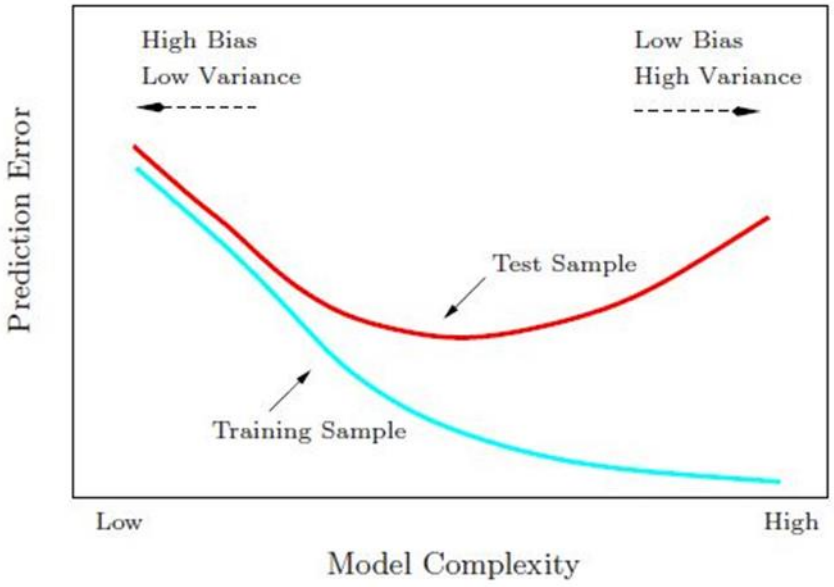
- Used with cross validation or holdout set
- Ensures that all partitions have fixed ratio of classes
  - Same ratio as training set
  - If training set is 5% class 1, 95% class 2, so is each partition
- Use with very uneven class distributions
- Avoid when class distribution isn't constant

You have done everything

- Data is clean
- Missing values, noise etc. are dealt with
- Features engineered
- Right metric has been chosen
- Model is trained
- What is the next step?

Now tune the parameters

- You will tune the parameters until you get the right trade-off between bias and variance
- Compare using cross-validation



QUESTIONS