Machine Learning Review

Basic concepts

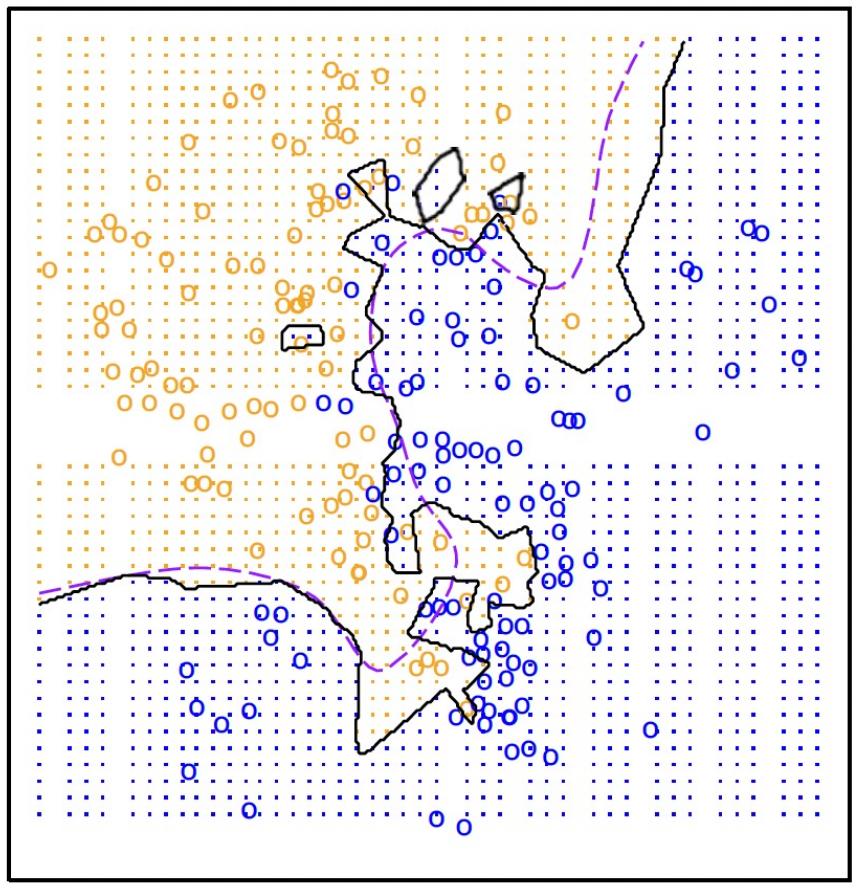
Resampling methods

- Model Assessment: having chosen a final model, estimating its prediction error on new data.
- Model Selection: estimating the performance of different models in order to choose the best one.
- Repeatedly drawing samples from a training set and refitting a model on each sample to obtain more information about the fitted model.
  - Example: Estimate the variability of a linear regression fit by repeatedly drawing different samples from the training data, fitting a regression model to each new sample, and then examining the extent to which the resulting fits differ.

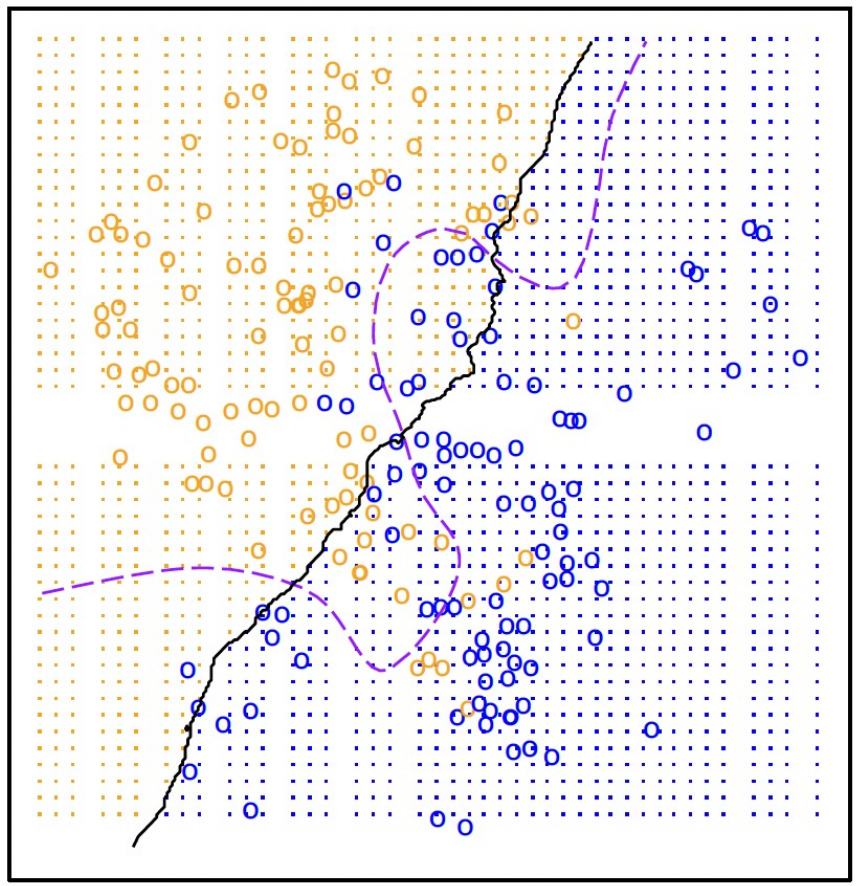
Model assessment

- Generalization: the prediction capability of a model (f) on independent test data
  - Given testing samples (X, Y), and prediction $\widehat{Y} = f\ (x, \theta)$
    - θ is the model (hyper) parameter, e.g. K in KNN
  - Prediction error (or "Loss function"): e.g.
    - Squared error: $L\ (Y, \hat{Y}) = (Y - \hat{Y})^2$
    - Absolute error: $L\ (Y, \hat{Y}) = |Y - \hat{Y}|$
- Note: The training error rate can dramatically underestimate the test error rate.

KNN: K=1

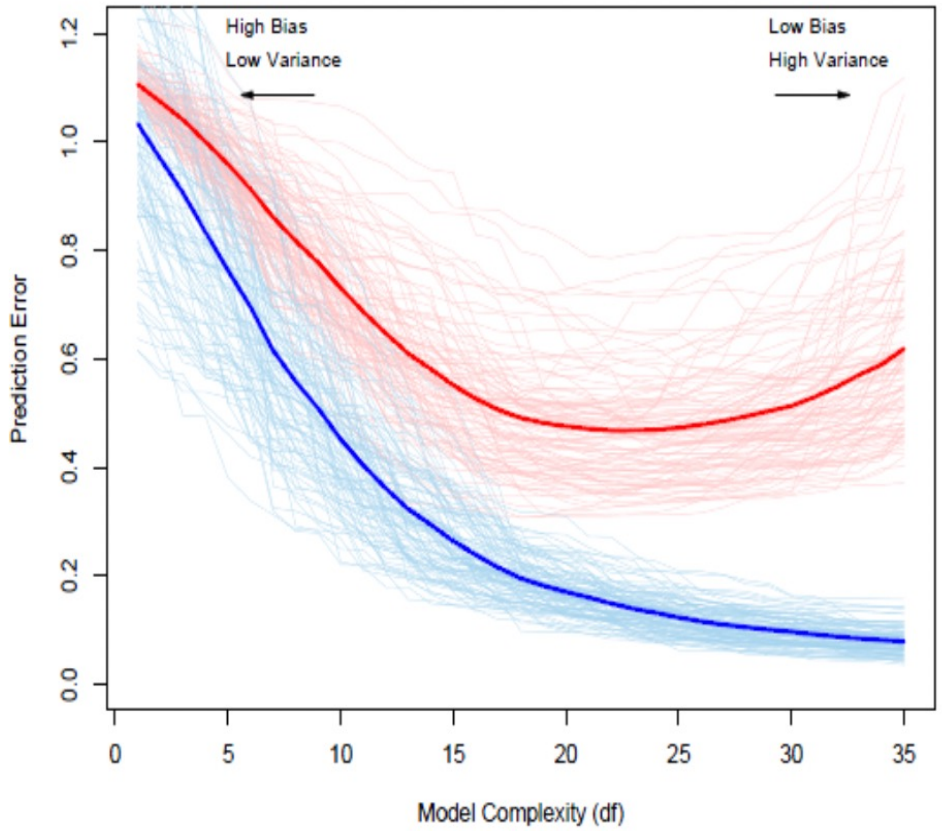


KNN: K=100



Model assessment

- As the model becomes more and more complex, it uses the training data more and is able to adapt to more complicated underlying structures.
- Hence, there is a decrease in bias but an increase in variance.
- Training error consistently decreases with model complexity.
- A model with zero training error is overfit to the training data and will typically generalize poorly.



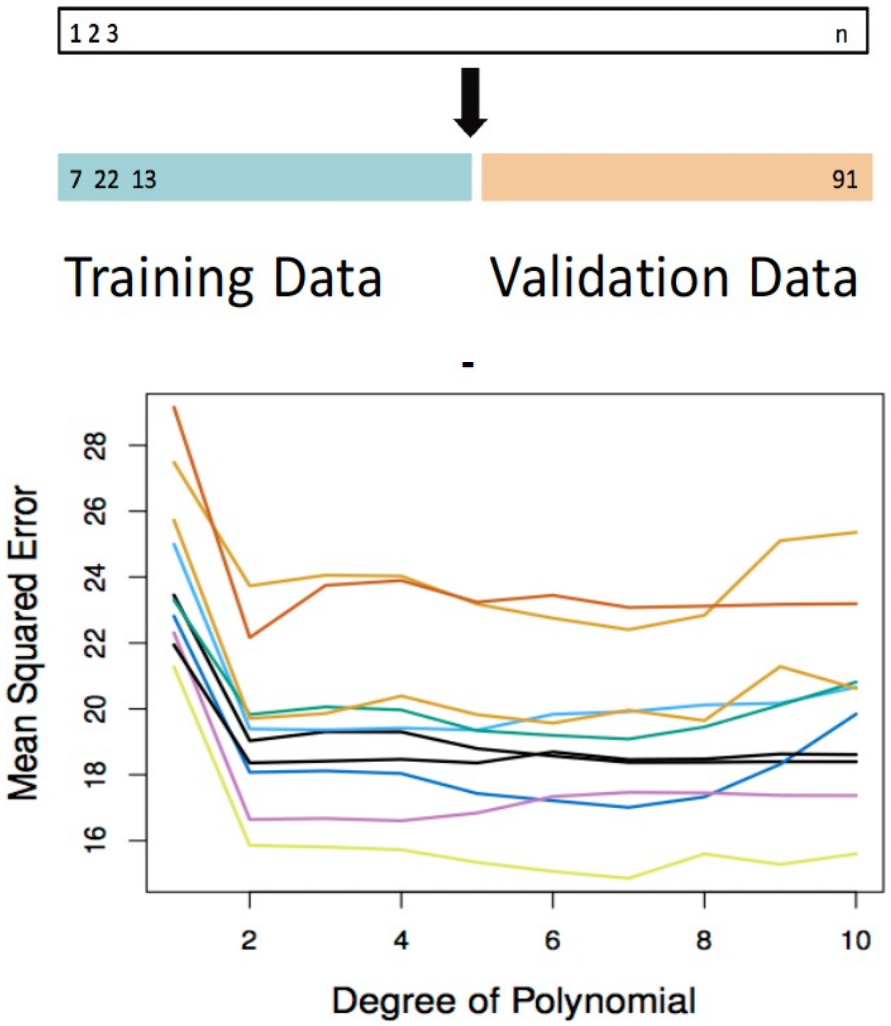
Model assessment

- Data-rich situation: randomly split data into training, validation, and test sets (e.g. 50%, 25%, 25%)
  - **Training** set: fit the model
  - **Validation** set (sometimes called evaluation set):
    - tune model parameters
    - estimate prediction error for model selection
  - **Test** set: assess the prediction error of the final chosen model
    - in machine learning competition, usually a test set is given without truth labels

Model selection

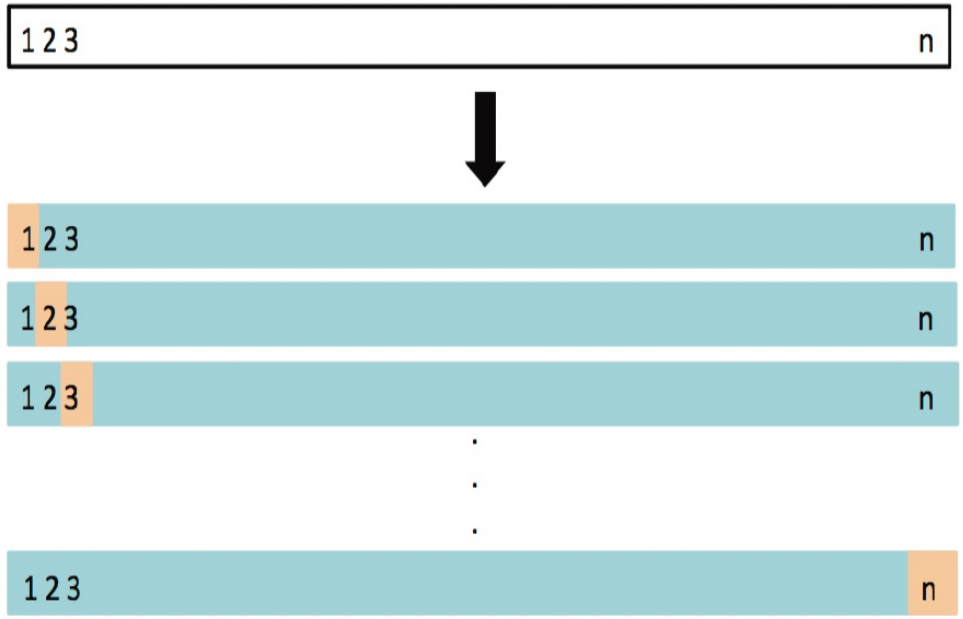
- The most important use of validation is for model selection.
- In almost every learning situation, there are some choices to be made and we need a principled way of making these choices.
- Validation can be used to estimate the out-of-sample error for more than one model.
  - Suppose we have M models.
  - We evaluate each model on the validation set to obtain the validation errors.
  - Then select the model with the lowest validation error.

Validation set



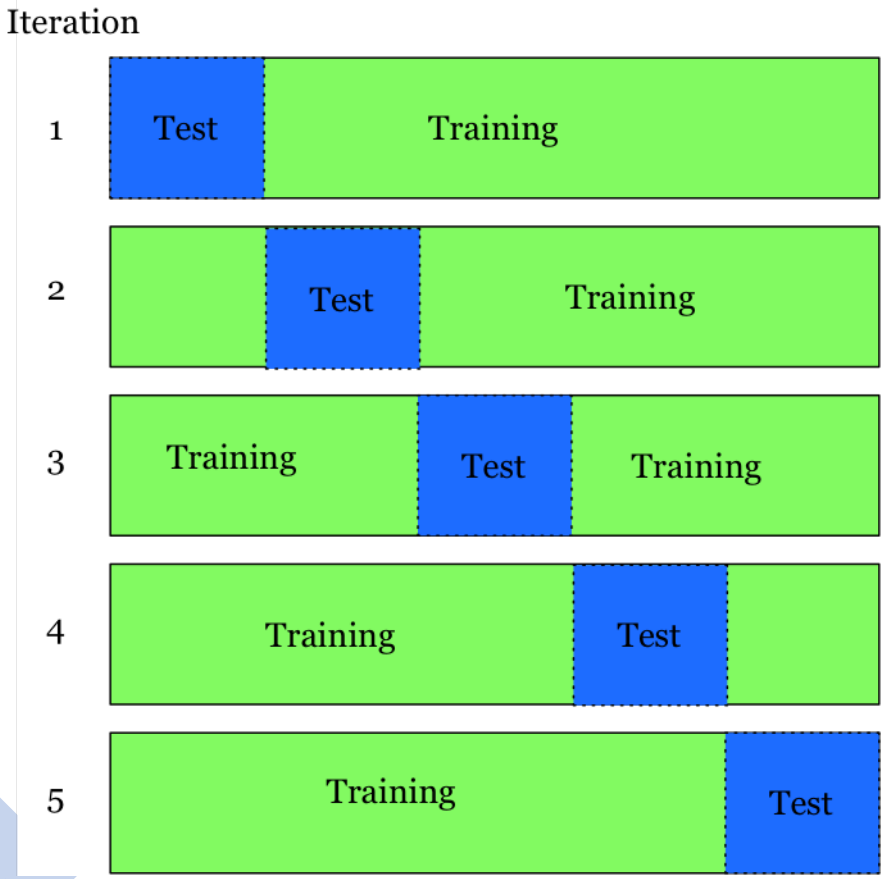
- We can randomly split the data into separate training and validation data sets.
- Then, we use the training data set to build each possible model and select the model that gives the lowest error on the validation set.
  - Validation error can be highly variable.
  - Only a subset of data is used to train the model.
  - Validation error may overestimate the test error.

Leave-One-Out Cross-Validation



- A single observation is used for the validation set and the remaining observations (n – 1) make up the training set.
- Fit the model using the training data set
- Evaluate the model using validation set and compute the corresponding MSE.
- Repeat this process n times, producing n mean squared errors.
- $\text{CV} = \frac{1}{n} \sum_{i=1}^{n} MSE_i$
  - Does not overestimate test error.
  - Yield similar results.
  - Computationally intensive.

K-Fold Cross Validation



- The simplest and most widely used method.
- We randomly divide the data set into K folds (typically K = 5 or 10).
- The first fold is treated as a validation set and fit a model on the remaining K – 1 folds.
- The MSE is computed on the observations in the held-out fold. The process is repeated K times, taking out a different part each time.
- Average the K estimates of the test error.

What to do with CV?

- Average prediction error: $CV(\theta) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, f^{k(i)}(x_i, \theta)\right)$
  - Where:
    - θ: the model parameters. (e.g., K in KNN)
    - $f^{k(i)}$: model fitted on the kth iteration
    - N: number of samples
  - Tune model parameters (θ) to minimize the average prediction error
    - e.g. How to determine K in KNN?
- Select the model with the minimal prediction error (along with θ determined)
- Fit the selected model to all the data!
- How to decide K?

Performance metrics

- **Accuracy** is often used to measure model performance
- $Accurarcy = \frac{\#\ correct\ predictions}{\#\ of\ total\ samples}$
- However, for an imbalanced classification problem where one category represents the overwhelming majority of the data points, accuracy can be a problematic metric
  - e.g. prediction of rare diseases: Assume only 1 patient in every 1M population. What is the accuracy if you predict everyone is healthy?
  - e.g. prediction of earthquakes

Precision, Recall, F-score

- Precision: percentage of true cases among the predicated true cases
- Recall: percentage of true cases that have been retrieved over the total number of true cases
- F-score = $\frac{2*precision*recall}{precision+recall}$

Confusion Matrix

165 Samples in total		Actual	
		YES	NO
PREDICTED	YES	100	10
	NO	5	50

For "YES" group:

precision=?

recall=?

f-score=?

For "NO" group:

precision=?

recall=?

f-score=?

Overall model performance

- precision_macro (or recall_macro or f1_macro) is calculated as:
  - calculate precision for each label
  - average over labels

- precision_micro (or recall_micro or f1_micro): calculates metrics globally regardless of labels.
  - precision_micro = recall_micro = accuracy
  - Note: precision_micro and recall_micro are the same for single label classification

- With imbalanced classes, the difference between these two metrics may be significant, e.g.
  - class 1: 950 samples, precision 90%
  - class 2: 50 samples, precision 70%

Tradeoff between precision and recall

- Is it possible to maximize precision and recall at the same time?
- To increase precision, you only give correct label to those you are highly confident with, but this results in a consequence that not all of the correct cases are sufficiently retrieved.
- To increase recall, you can give label to those you are not confident with, but precision is bad.

Prediction errors

- False Positive: Predict a sample is positive when it is in fact negative.
- False Negative: Predict a sample is negative when it is in fact positive.

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

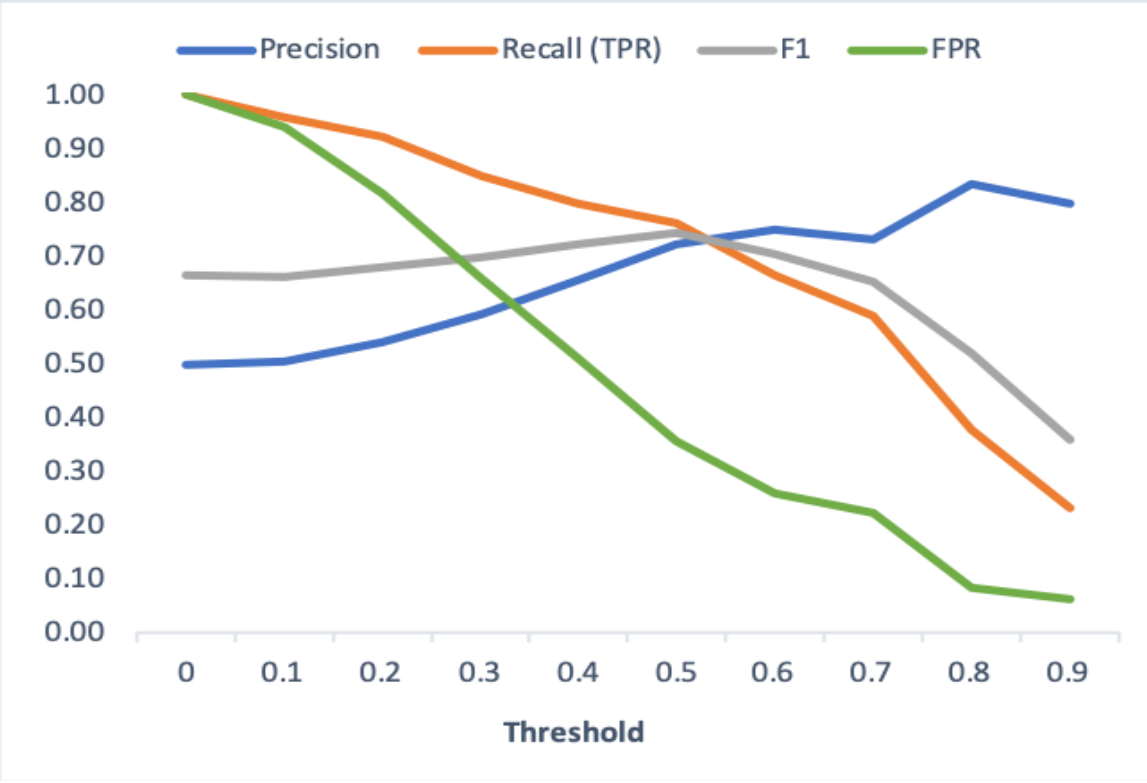
Metrics

- $Precision\ of\ positive\ class = \frac{True\ positive}{true\ positive + false\ positive}$
- $Recall\ of\ positive\ class = \frac{True\ positive}{true\ positive + false\ negative}$
- $True\ positive\ rate\ (TPR) = recall\ of\ positive\ class = sensitivity$
- $False\ positive\ rate\ (FPR) = \frac{False\ positive}{False\ positive + true\ negative} = 1 - specificity = 1 -$ recall of negative class

Two types of prediction problems

- Predict class labels directly, e.g. report the class of the majority of K nearest neighbors in KNN
- Predict a **probability** (i.e. confidence) for each class, e.g. the percentage of neighbors of a class
  - Calibrate a probability threshold to determine class labels for desired performance levels
  - How can the threshold affect precision, recall, true positive rate, and false positive rate? e.g. threshold 0.90 vs. 0.10

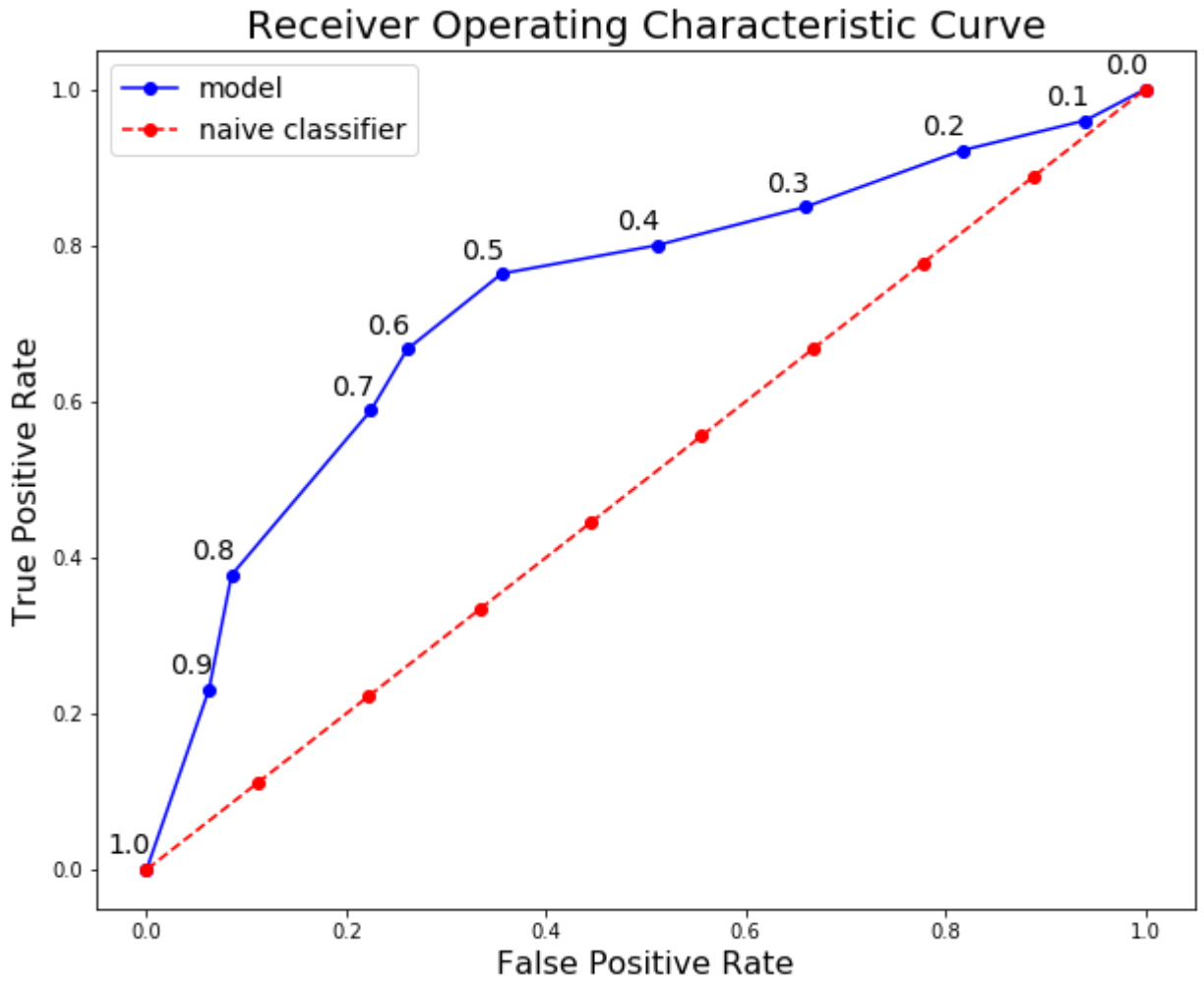
Threshold	TP	FP	TN	FN	Precision	Recall (TPR)	F1	FPR
0	50	50	0	0	0.50	1.00	0.67	1.00
0.1	48	47	3	2	0.51	0.96	0.66	0.94
0.2	47	40	9	4	0.54	0.92	0.68	0.82
0.3	45	31	16	8	0.59	0.85	0.70	0.66
0.4	44	23	22	11	0.66	0.80	0.72	0.51
0.5	42	16	29	13	0.72	0.76	0.74	0.36
0.6	36	12	34	18	0.75	0.67	0.71	0.26
0.7	30	11	38	21	0.73	0.59	0.65	0.22
0.8	20	4	43	33	0.83	0.38	0.52	0.09
0.9	12	3	45	40	0.80	0.23	0.36	0.06



- As the threshold increases, for the positive class, how does each of the metric change?
  - Precision?
  - Recall or true positive rate (TPR)?
  - False positive rate (FPR)?
- As the threshold increases, precision increases but recall decreases, both TPR and FPR decrease. Ideally, we want high TPR and low FPR, but they move in the same direction.

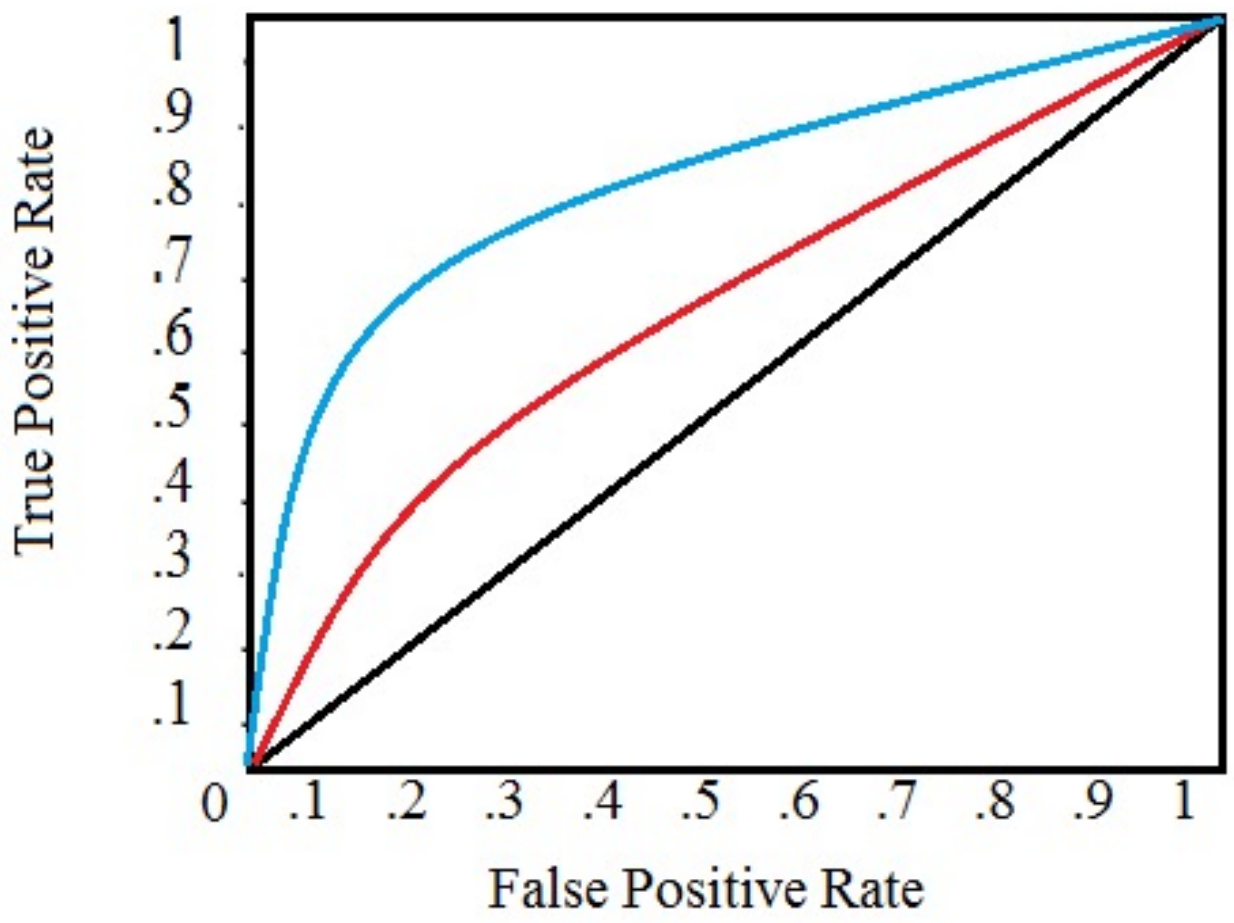
ROC curve

Receiver operating characteristic (ROC) curve: plots the true positive rate (TPR) vs the false positive rate (FPR) as a function of the model's threshold for classifying positives.



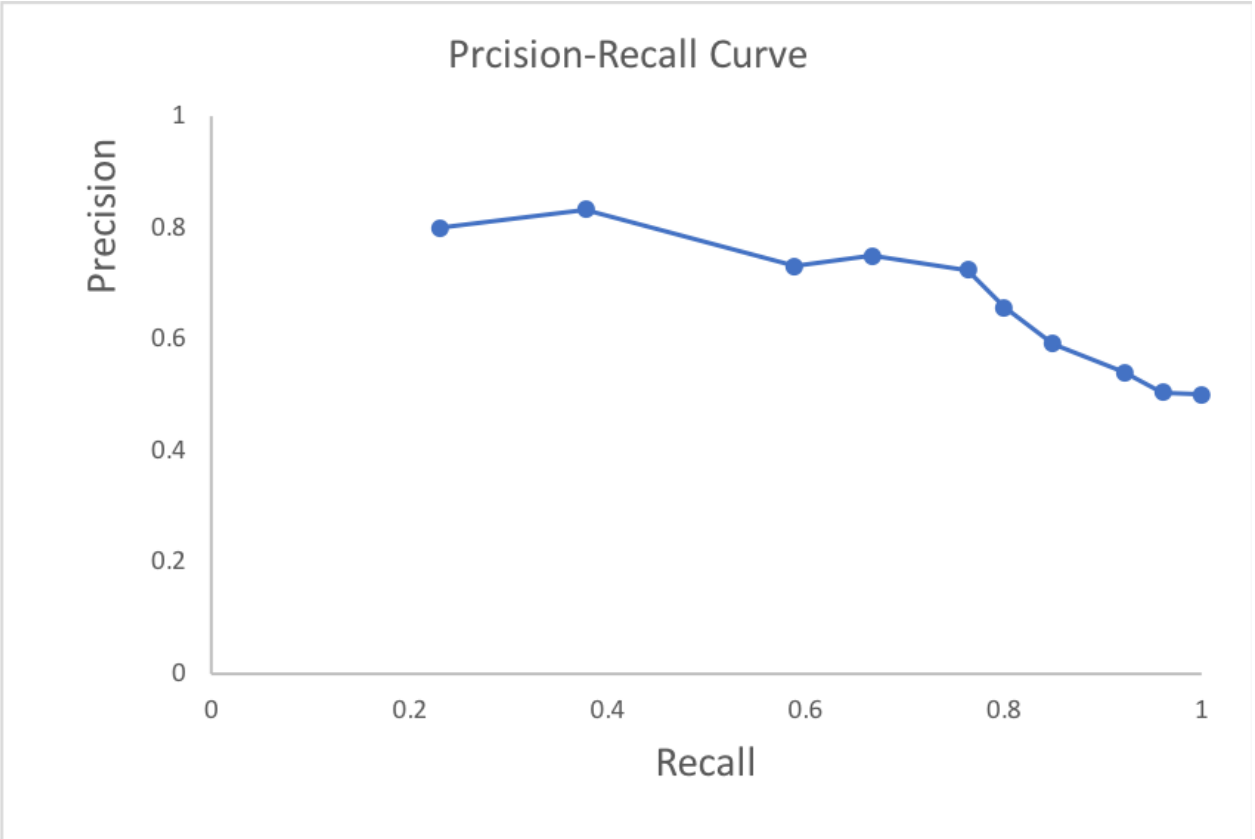
AUC curve

- **Area under the curve (AUC)**: metric to calculate the overall performance of a classification model based on area under the ROC curve
  - Which model is better?
  - How to determine best threshold?



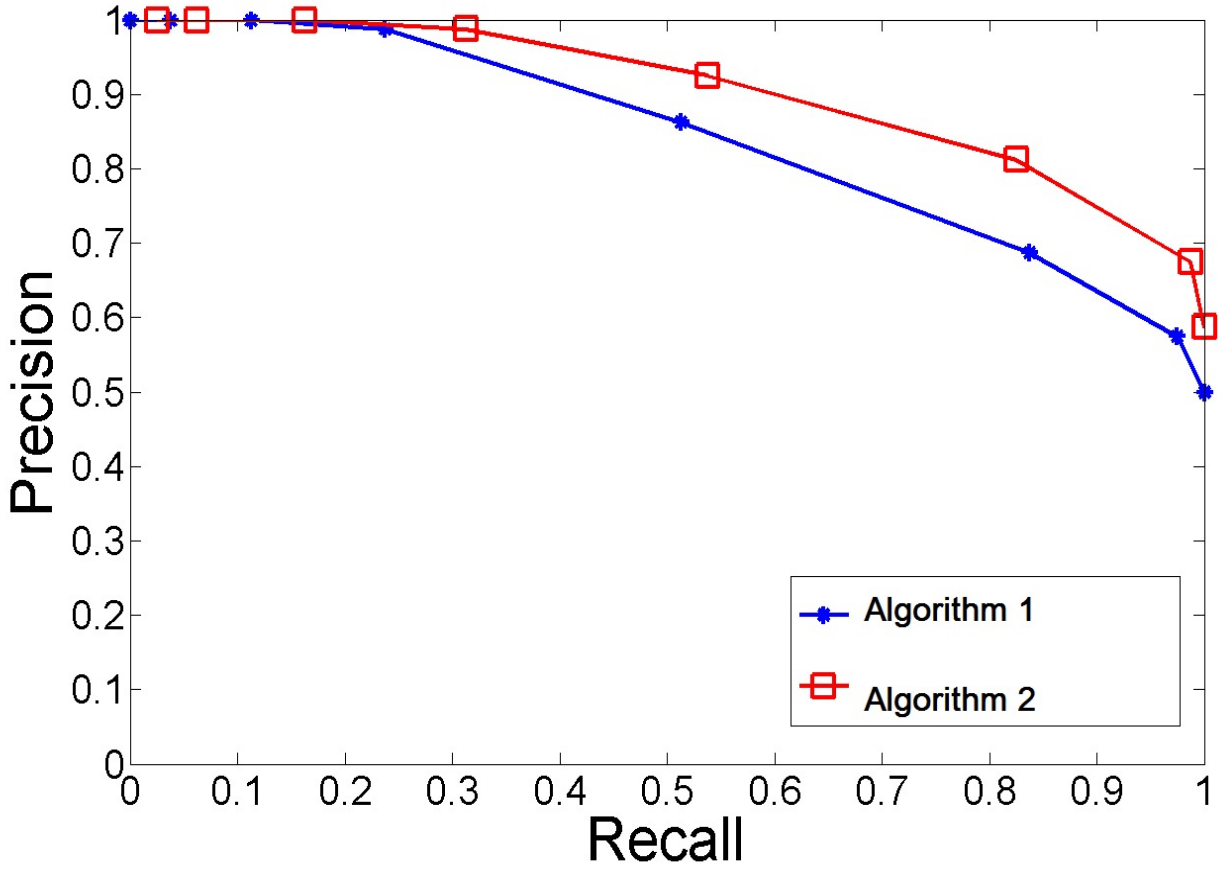
Precision-Recall Curve

- **Precision-Recall Curve** : plot the precision (y-axis) and the recall (x-axis) for different thresholds, much like the ROC curve.

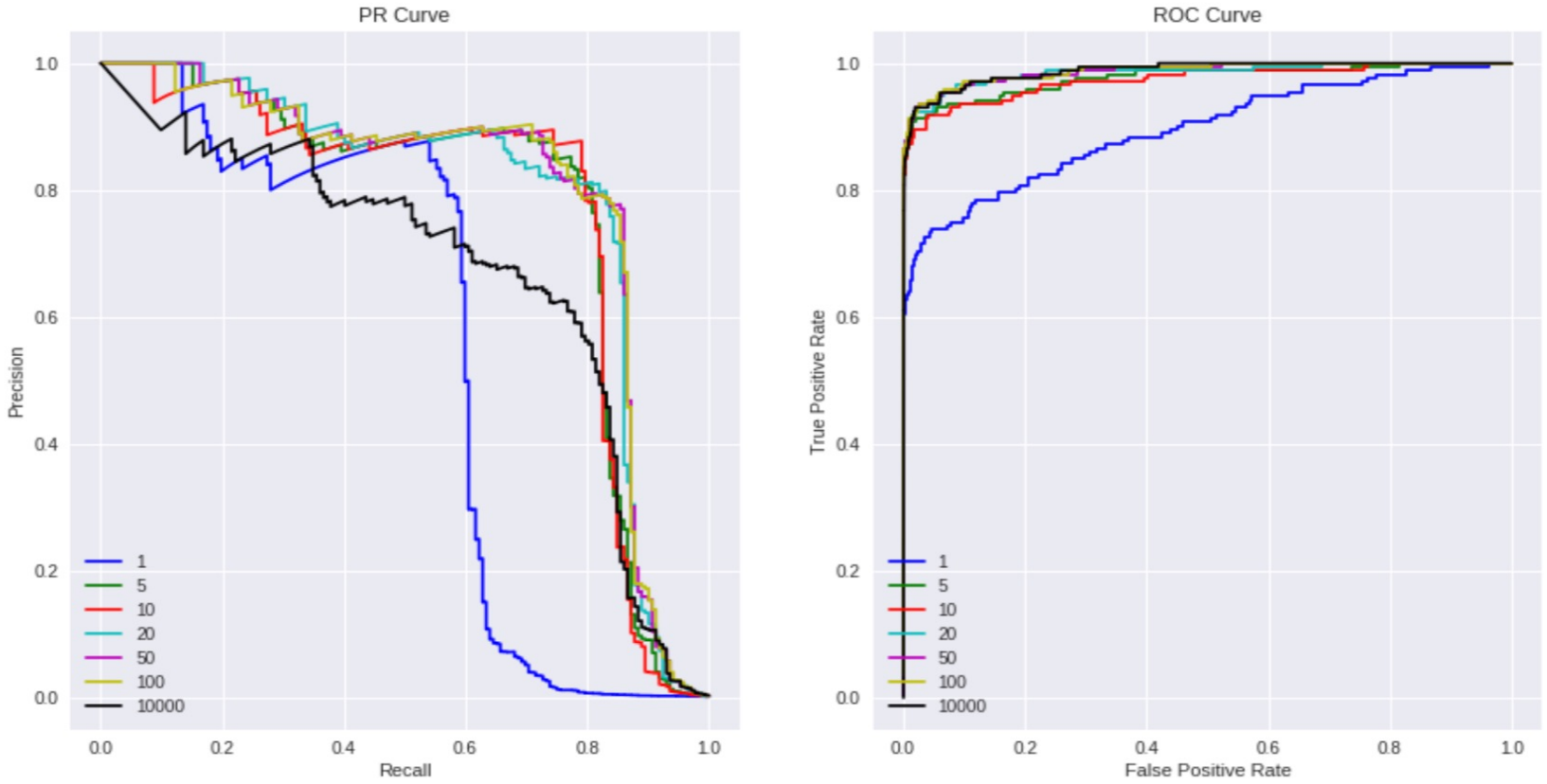


Precision-Recall Curve

- Which algorithm is better?
  - source: https://i.stack.imgur.com/T0kQr.png
- More about ROC and PRC:
  - When the dataset is extremely imbalanced, i.e. the majority is negative samples, a model with high AUC is not necessarily good. For experiment, check https://www.kaggle.com/lct14558/imbalanced-data-why-you-should-not-use-roc-curve
  - For in-depth discussion between ROC and PRC, read Davis, J. and Goadrich, M., The relationship between Precision-Recall and ROC curves. http://pages.cs.wisc.edu/~jdavis/davisgoadrichcamera2.pdf



ROC or PRC for imbalanced data



- Source: https://www.kaggle.com/lct14558/imbalanced-data-why-you-should-not-use-roc-curve