COURSE LOGISTICS

- Midterm: 11/8 Wednesday
- HW1 due 9/12 (FAQ: DataFrame -> numpy)
- HW2 out 9/13, due 9/29
- Python workshop session 3 on 9/19 (ML workflow)

DECISION TREES (PART II CONT.)

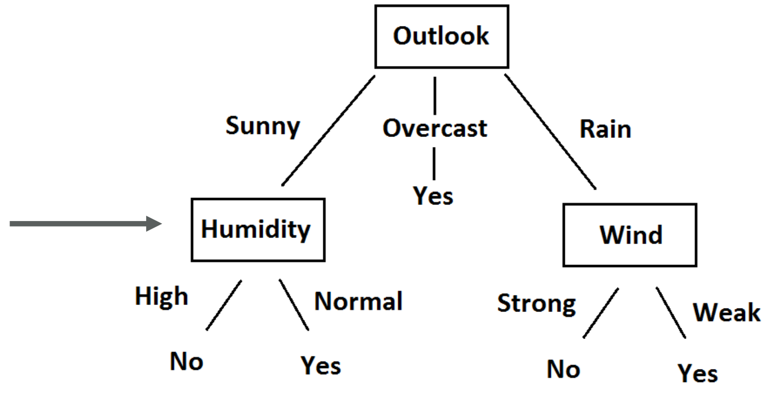
CS 334: Machine Learning

Slides adapted from Intel ML Academy, David Sontag, Luke Zettlermoyer, Carlos Guestrin, Andrew Moore, Mengye Ren, Matthew MacKay, and Yubin Park

REVIEW: DECISION TREE

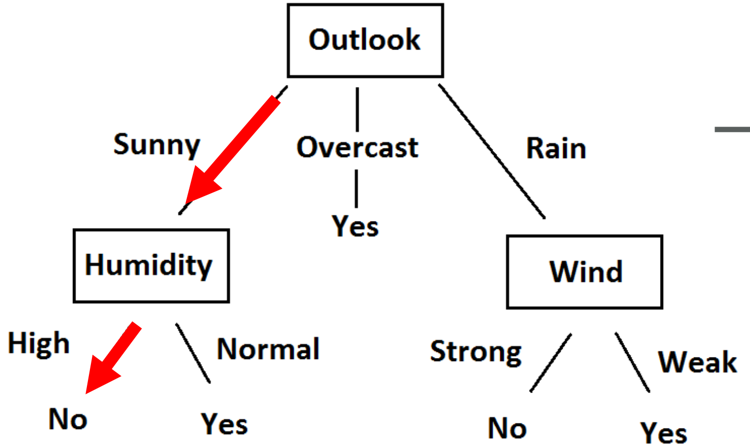
- Training: Build a decision tree from training data

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



- Prediction: Given a new data point, find the path using its features and predict the label at the leaf node

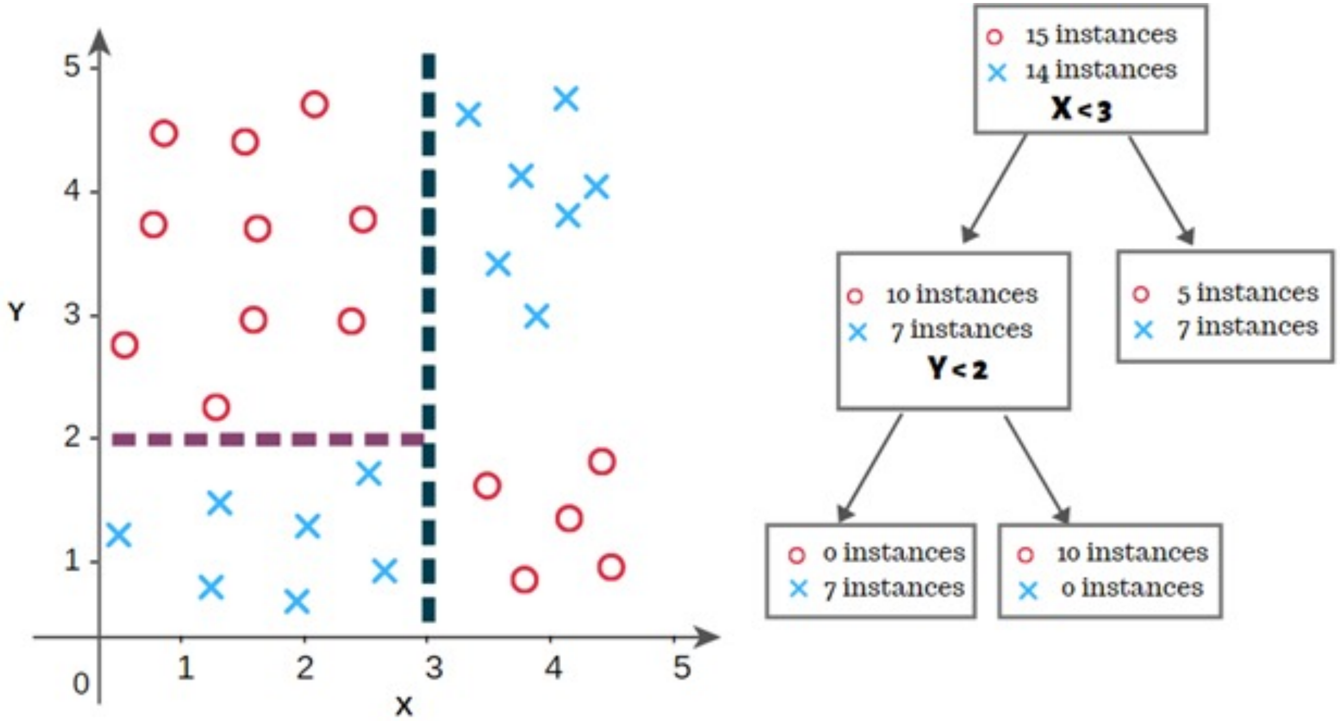
(Sunny, Mild, High, Weak)



$\longrightarrow$ no

REVIEW: HOW TO LEARN THE TREE?

- Recursively create a tree node that splits the current data region into two subregions
  - How to choose the node (splits)?
  - When to stop the tree (how big to grow)?

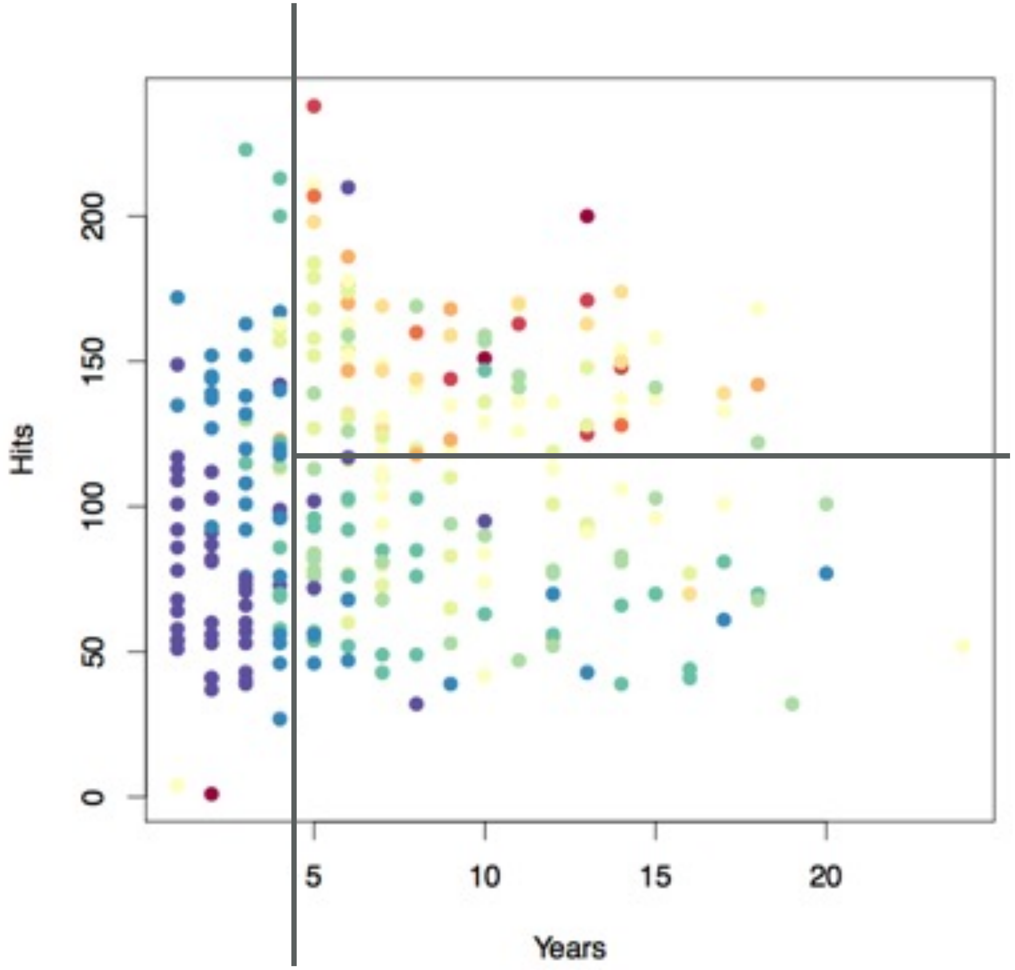


DECISION TREE

- How to build (learn) a decision tree
- Regression vs. classification
- Handling missing attribute values
- Decision tree vs. kNN
- Decision tree in Sklearn

GROUP ACTIVITY

REGRESSION TREES: BASEBALL SALARY PREDICTION



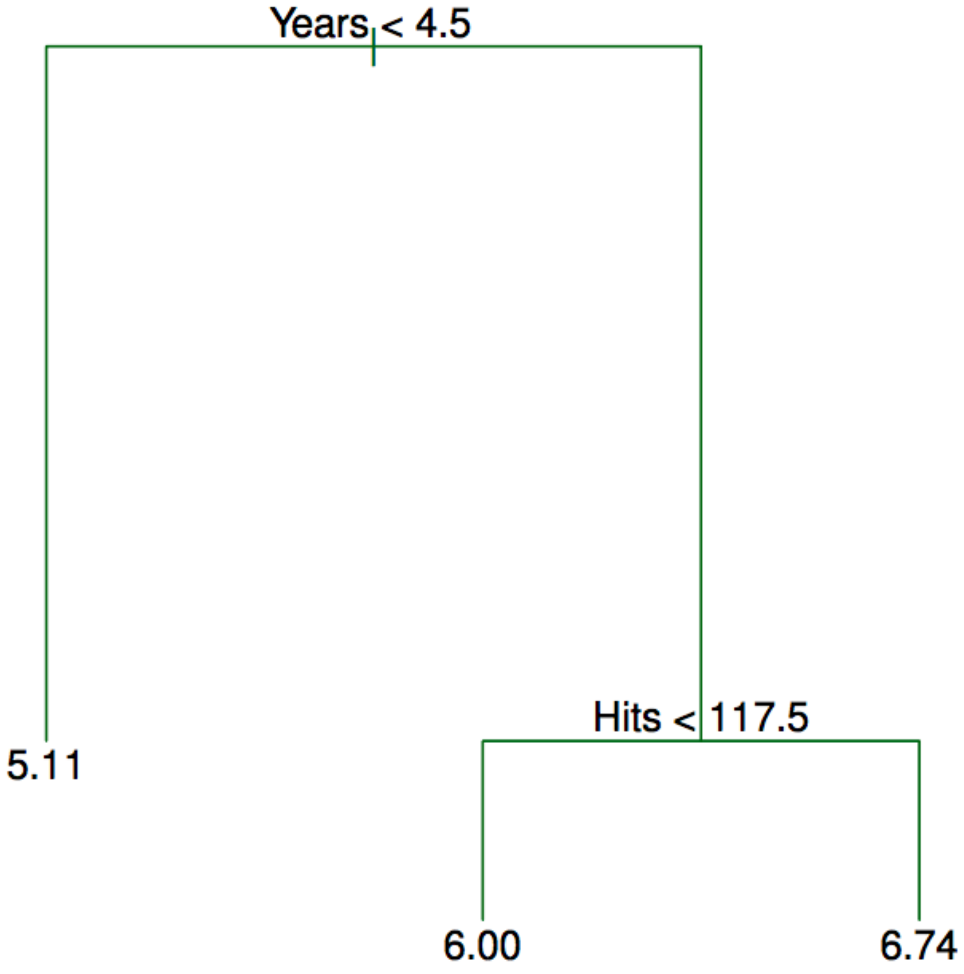
Salary is color-coded from low (blue, green) to high (yellow, red)

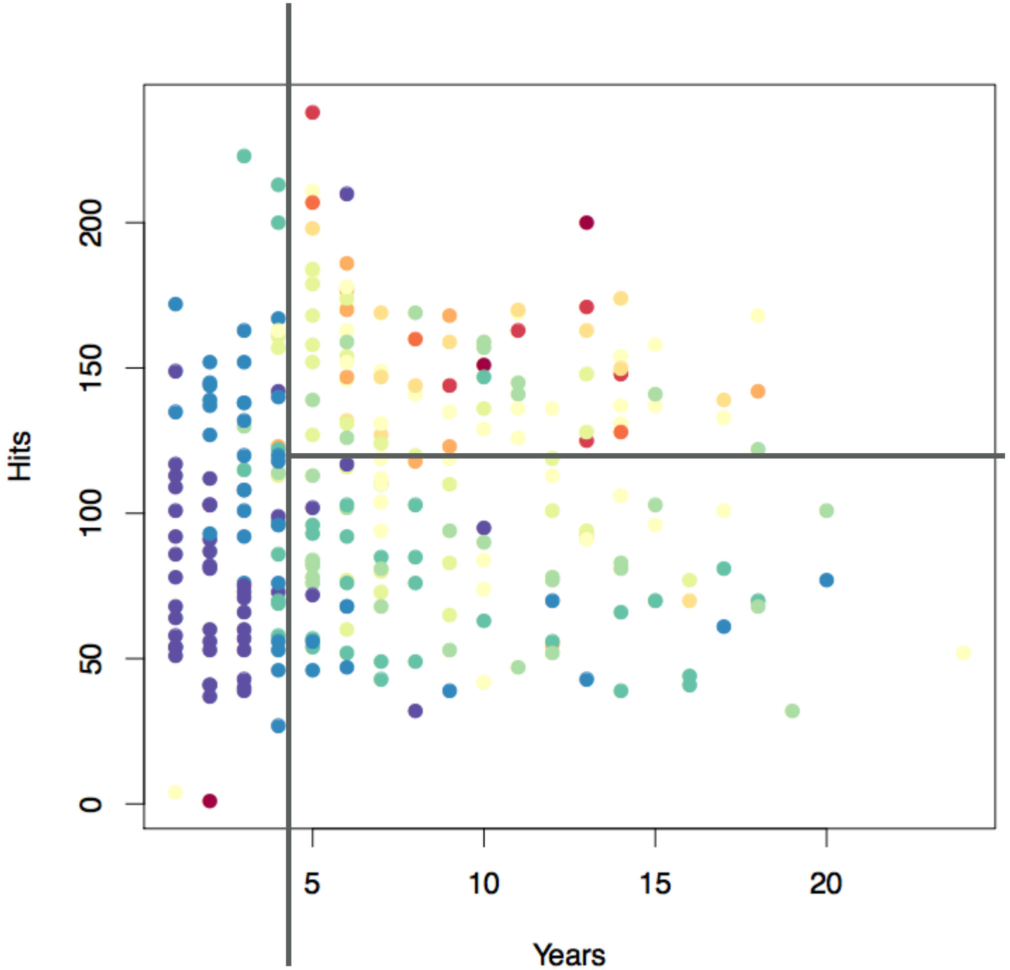
What value to predict at leaf node?
What splitting criteria to use?

REGRESSION TREES

- Similar idea to classification trees
- Each region predicts a single continuous outcome (average) (vs. majority class or probability)
- Splitting criteria: Minimize residual sum of squares (RSS) in regions (vs. entropy or gini) $\sum_{j} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$
  - Others: mean squared error, mean absolute error

EXAMPLE: REGRESSION





DECISION TREE

- How to build (learn) a decision tree
- Regression vs. classification
- Handling missing attribute values
- Decision tree vs. kNN
- Decision tree in Sklearn

DEALING W/ MISSING DATA

- Standard approaches
  - Delete observations with missing features
  - Delete features with missing observations
  - Fill in (impute) missing values with mean, median, zero, or other values

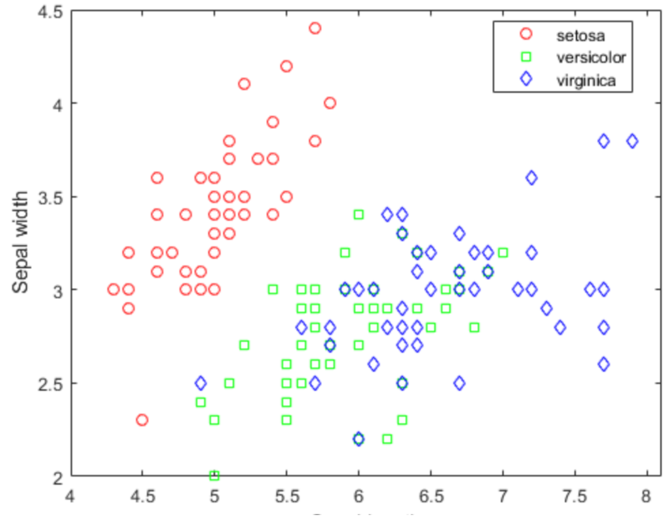
DECISION TREES & MISSING DATA

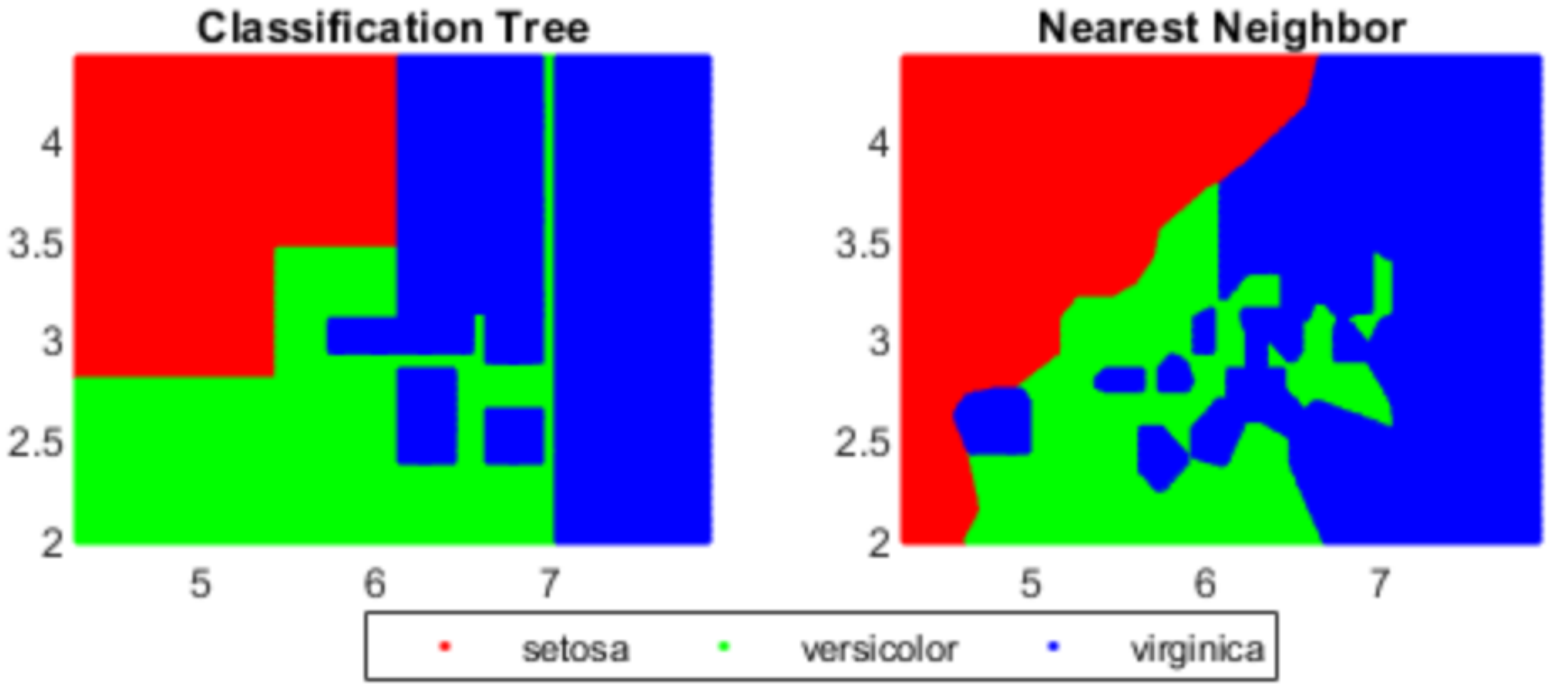
- Categorical Features: Create a new category for missing
  - Associate missing value to each decision node
- Construct surrogate variables (e.g. CART)
  - Create list of surrogate predictors
  - If missing primary, try the surrogate splits in order

DECISION TREE

- How to build (learn) a decision tree
- Regression vs. classification
- Handling missing attribute values
- Decision tree vs. kNN
- Decision tree in Sklearn

DECISION TREE VS KNN





DECISION TREE VS. KNN

- (+) Robust to scale of inputs and missing values
- (+) More interpretable
- (+) Faster to predict values
- (-) Requires more depth to handle complex boundaries

DECISION TREE

- How to build (learn) a decision tree
- Regression vs. classification
- Handling missing attribute values
- Decision tree vs. kNN
- Decision tree in Sklearn

DECISION TREE: SKLEARN

- Import the class containing the classification method
- Create an instance of the class with tree parameters
- Fit the training data and predict the values

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(criterion='gini',
                             max_depth=5)
dtc.fit(xData)
yHat = dtc.predict(xData)
```

https://scikit-learn.org/stable/modules/tree.html

DT-EXAMPLE-TENNIS.IPYNB

HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/1I15-0KKC1GT0LF9LRL6FGEL_QC9VVH33

DT-EXAMPLE-IRIS.IPYNB

HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/18J4NXMPKEQBF_7BOXJXQ4KGSUDSQSM-J

HOMEWORK #2 PREVIEW

- Out **9/13**, Due **9/29 @ 11:59 PM ET**
- 3 questions
  - Q1: Decision tree implementation
  - Q2: Model assessment
  - Q3: Robustness of k-nn and decision tree (model selection)

MODEL ASSESSMENT AND SELECTION

CS 334: Machine Learning

Slides adapted from Joyce Ho, Lee Cooper and Ryan Tibshirani

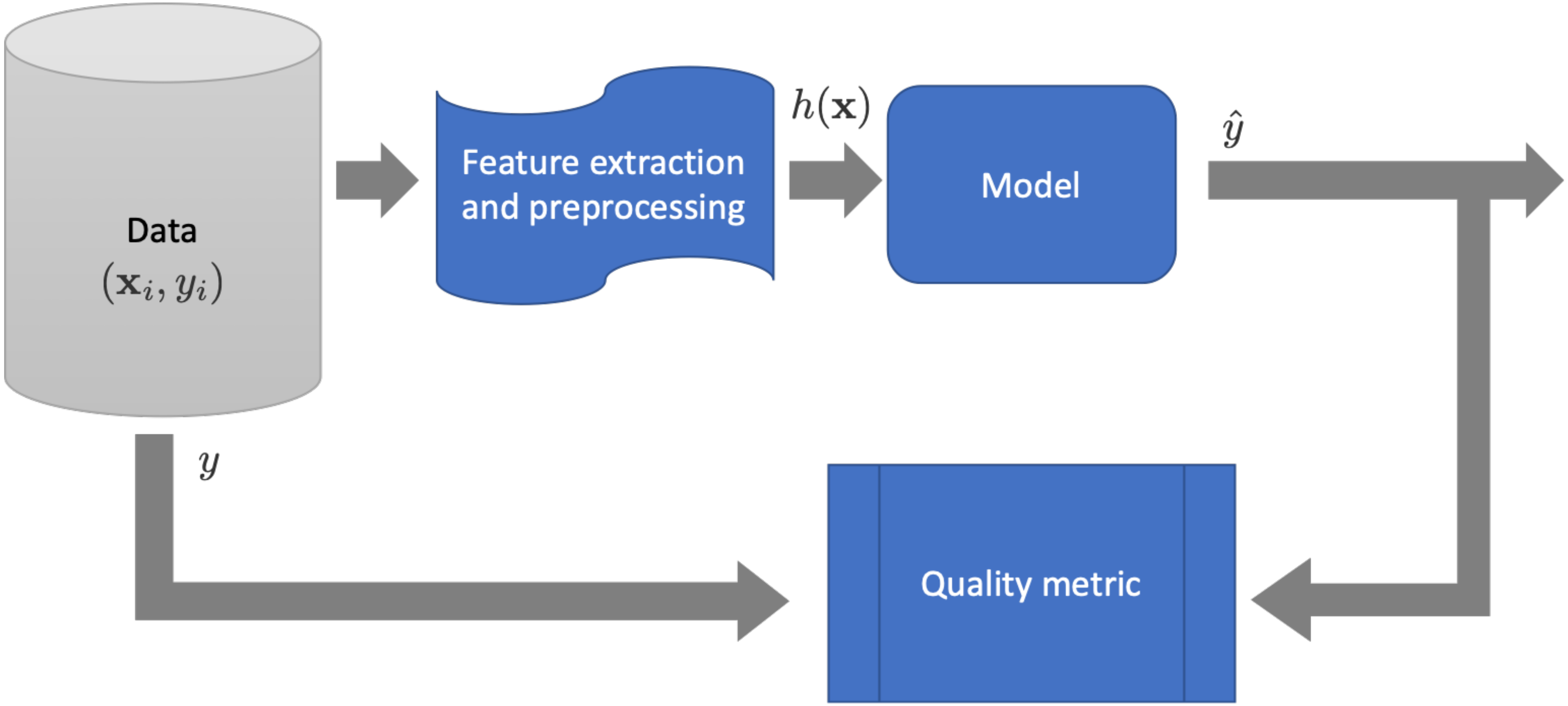
MODEL ASSESSMENT AND MODEL SELECTION

- Model assessment: evaluating a model's performance
- Model selection: selecting the proper level of flexibility for a model (e.g. K for KNN, tree size for decision tree)

“Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.”

–George Box, 1987

MEASURING PERFORMANCE



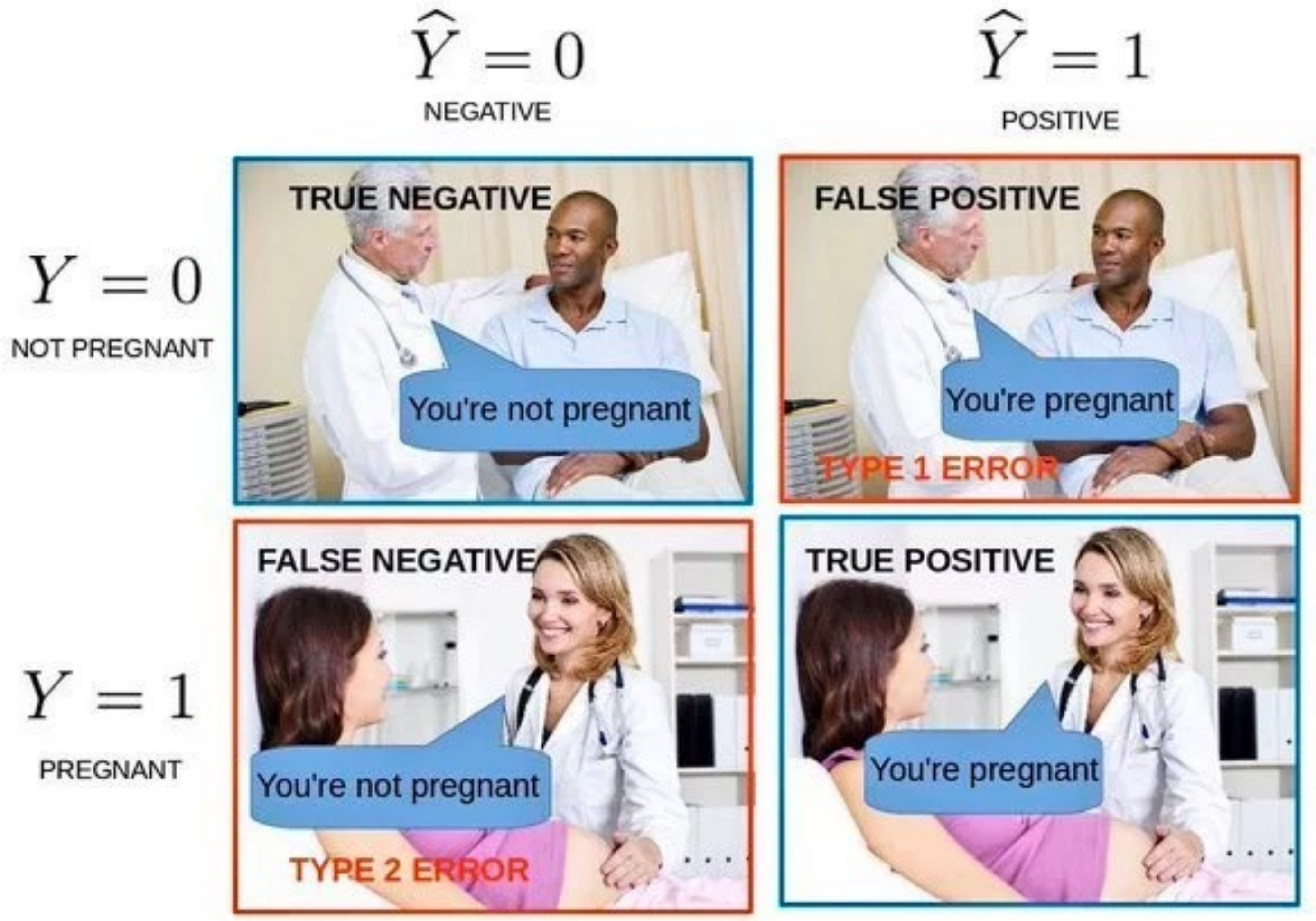
MODEL ASSESSMENT

- Metrics:
  - Classification: Accuracy, precision/recall, AUROC (TPR/FPR), AUPRC (precision/recall)
  - Regression: MSE
- Process: training/test split (holdout), K-fold cross-validation, Monte Carlo cross validation

CONFUSION MATRIX FOR BINARY CLASSIFICATION

	Predicted: (-)	Predicted: (+)
Actual: (-)	True Negative (TN)	False Positive (FP) Type 1 error
Actual: (+)	False Negative (FN) Type 2 error	True Positive (TP)

CONFUSION MATRIX



False Positive (FP)

False Negative (FN)

METRICS

- Accuracy

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

	Predicted: (-)	Predicted: (+)
Actual: (-)	**TN**	FP
Actual: (+)	FN	**TP**

What are the potential problems with accuracy?

PROBLEMS WITH ACCURACY

- Assumes equal cost for both types of error
- Can be inflated when data is imbalanced
  - Accuracy of the base case (predicting dominant class) can be very high

METRICS

- Accuracy

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

- True positive rate, sensitivity, or recall

$$TPR = \frac{TP}{TP + FN}$$

- False positive rate, or 1 - specificity

$$FPR = \frac{FP}{TN + FP}$$

- Positive predictive value, or precision

$$PPV = \frac{TP}{TP + FP}$$

	Predicted: (-)	Predicted: (+)
Actual: (-)	**TN**	FP
Actual: (+)	FN	**TP**

What if I predict everything positive?
Or negative?

SENSITIVITY AND SPECIFICITY: BALANCED ACCURACY

- Tradeoff between sensitivity and specificity

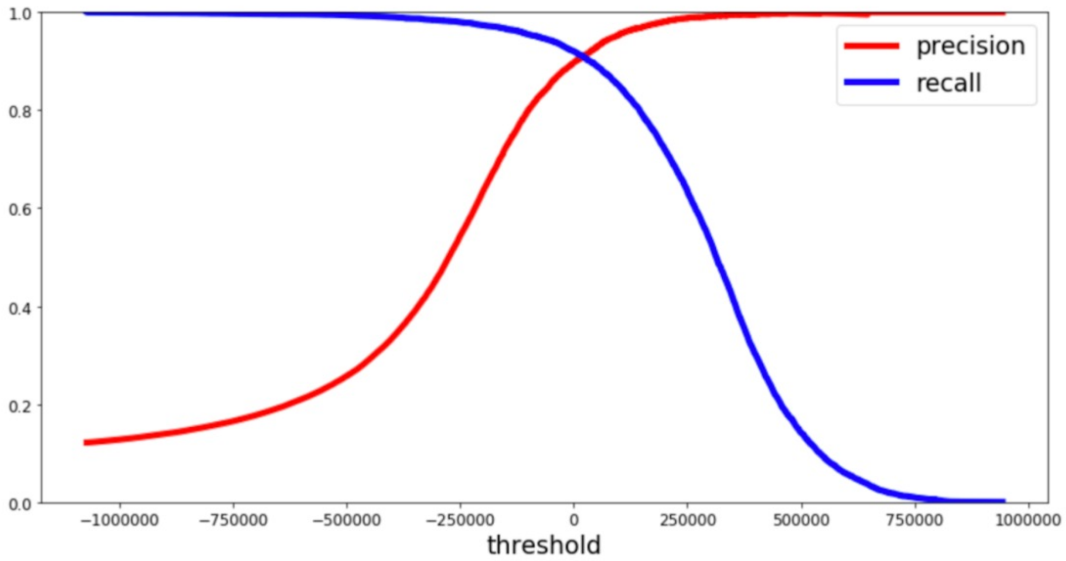
- Balanced accuracy: mean of sensitivity and specificity

$$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$$

PRECISION AND RECALL: F SCORE

- Tradeoff between precision and recall
- F1 Score: Harmonic mean between precision and recall

$$F_1 = \frac{2}{\frac{1}{TPR} + \frac{1}{PPV}}$$
$$= 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$



- General Formula

$$F_\beta = (1+\beta^2)\frac{\text{Precision} * \text{Recall}}{(\beta^2 \text{Precision}) + \text{Recall}}, \beta > 0$$

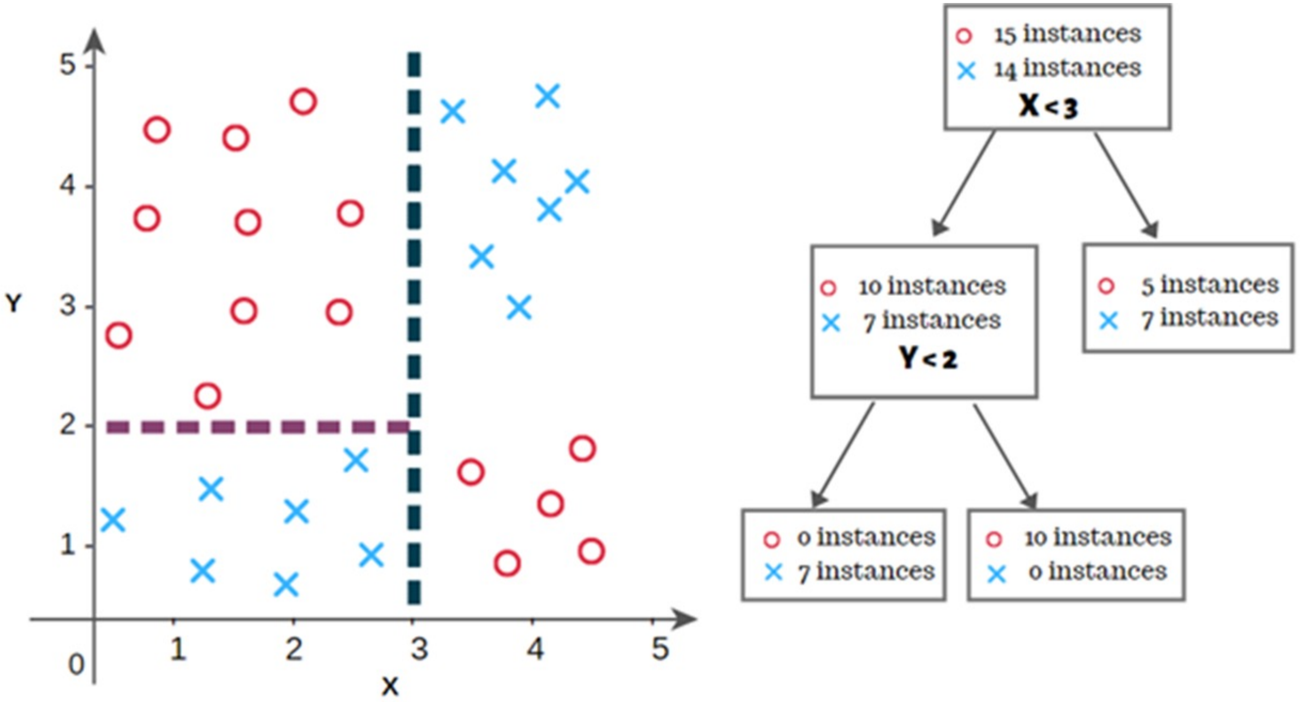
PREDICTED CLASS PROBABILITIES

- Each region R_j contains some subset of training data point
- Predicted probability is just proportion of points in the region belong to class k

$$\hat{p}_g(R_j) = \frac{1}{n_j} \sum_{\mathbf{x}_i \in R_j} \mathbb{1}_{\{y_i = g\}}$$

- Predicted class is the most common class occurring amongst these points

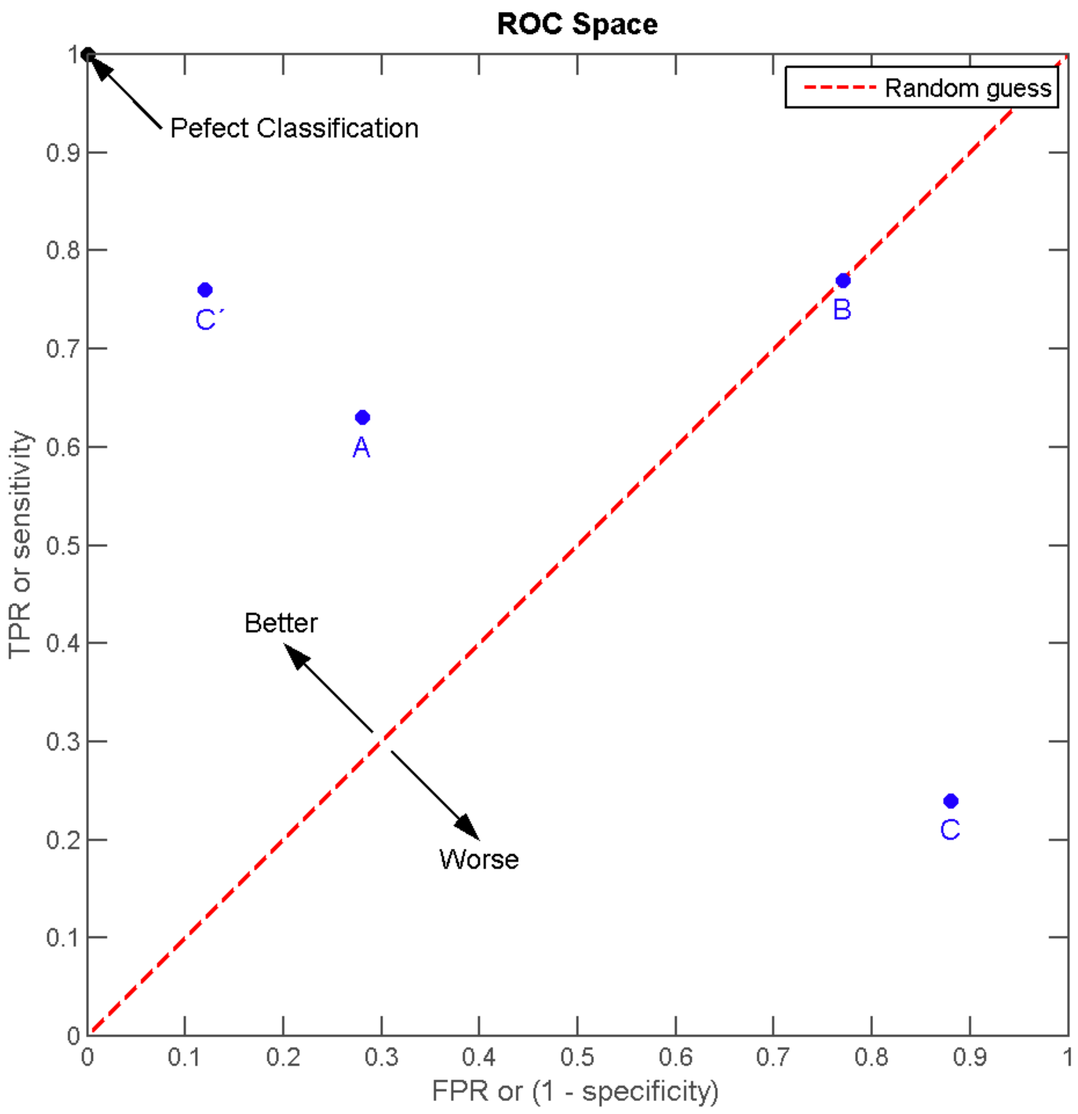
$$g_j = \text{argmax}\ \hat{p}_g(R_j)$$



What if we use different threshold/criteria based on the probability to make class prediction?

RECEIVER OPERATING CHARACTERISTIC CURVE: SPACE

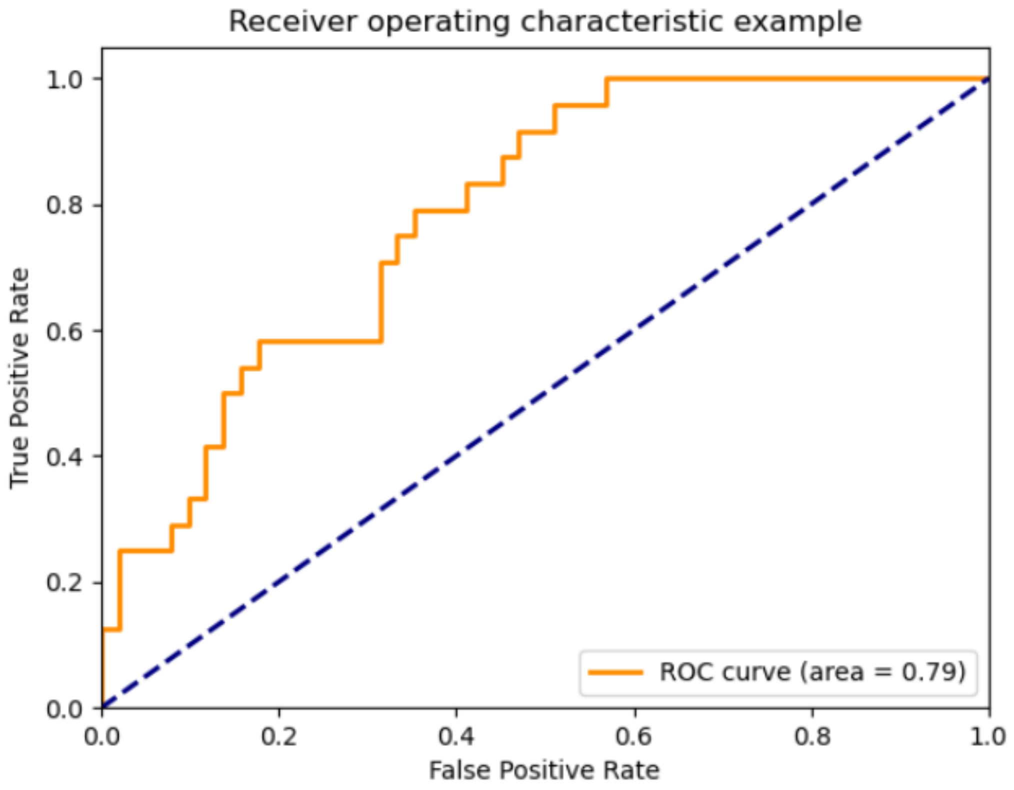
- X axis: FPR (False positive rate)
- Y axis: TPR (True positive rate)
- Each model/threshold corresponds to a FPR-TPR pair (point)
- The top-left (0,1) is the ideal model
- Diagonal represents no-discrimination (randomly predict with probability p)



https://en.wikipedia.org/wiki/Receiver_operating_characteristic

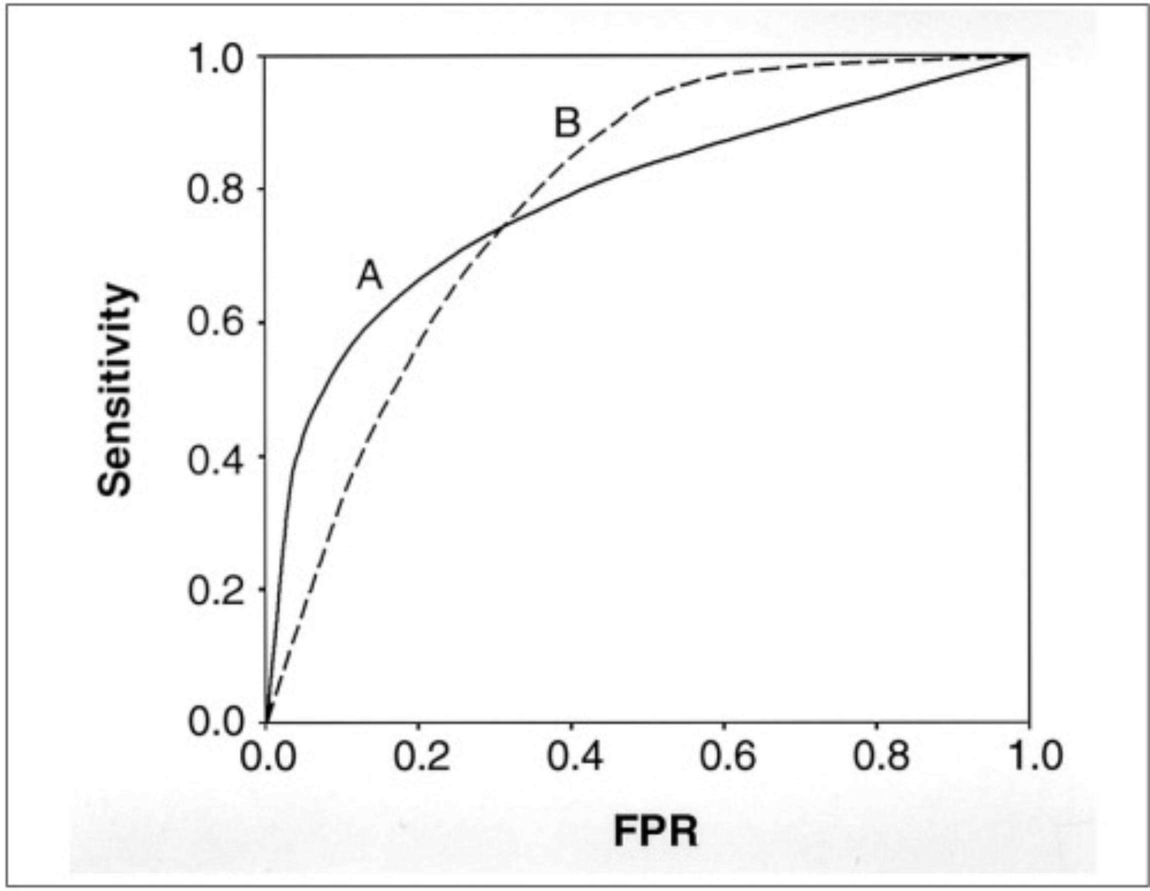
ROC CURVES

- Plot TPR and FPR of a model at various threshold settings
- Each point represents different tradeoff (cost ratio) between FPR and TPR
- Slope is always increasing
- Two non-intersecting curves means one method dominates the other



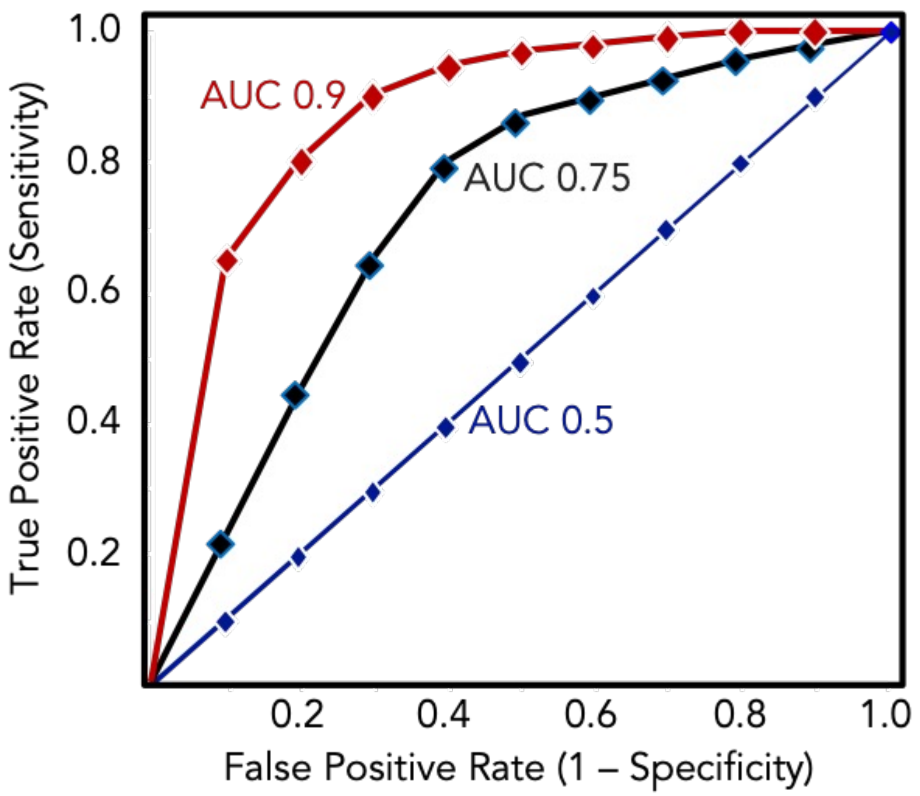
ROC CURVES

- Plot TPR and FPR of a model at various threshold settings
- Each point represents different tradeoff (cost ratio) between FPR and TPR
- Slope is always increasing
- Two non-intersecting curves means one method dominates the other
- What about intersecting curves?



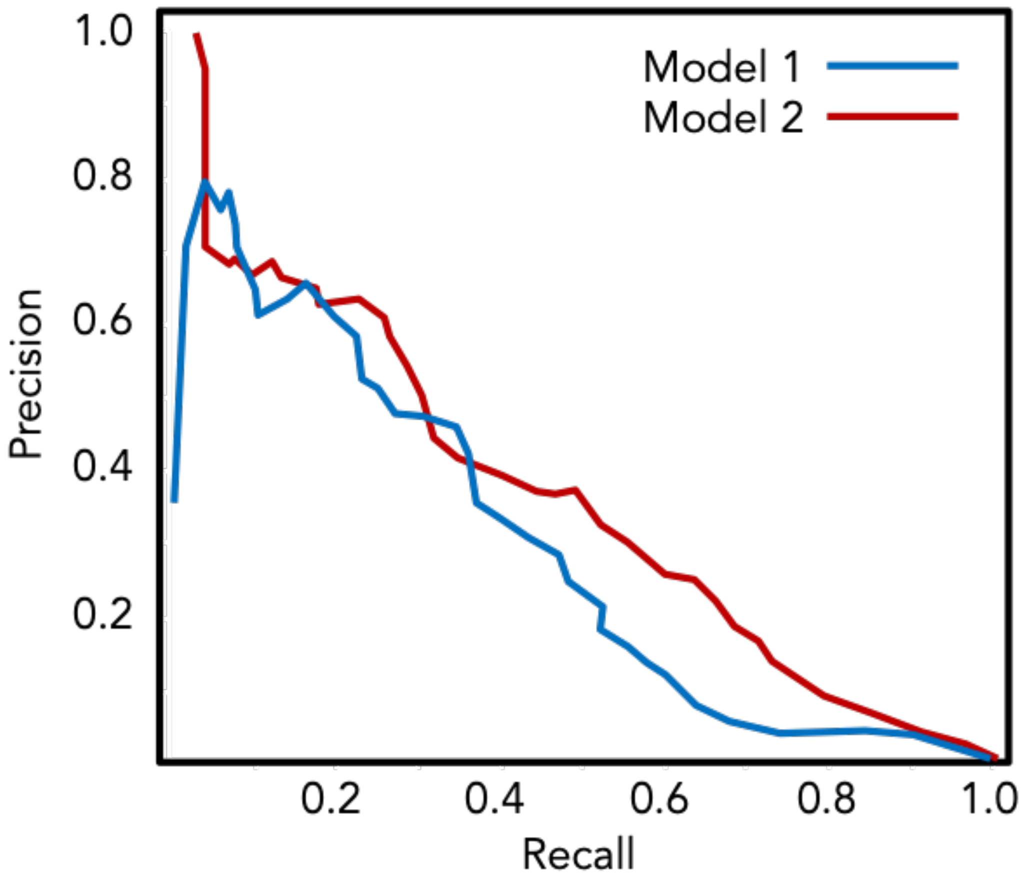
AREA UNDER ROC CURVE (AUC)

- > 0.9: excellent prediction — something potentially fishy, should check for information leakage
- 0.8: good prediction
- 0.5: random prediction
- <0.5: something wrong!



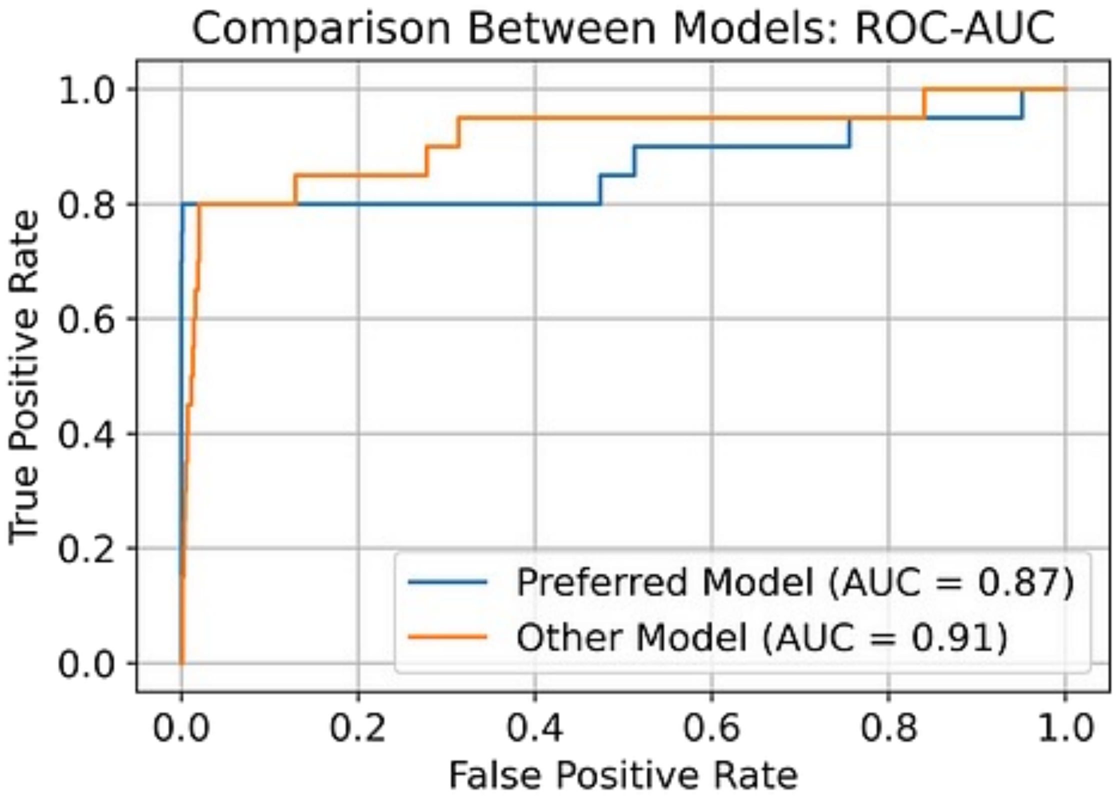
AREA UNDER PRECISION RECALL CURVE (AUPRC)

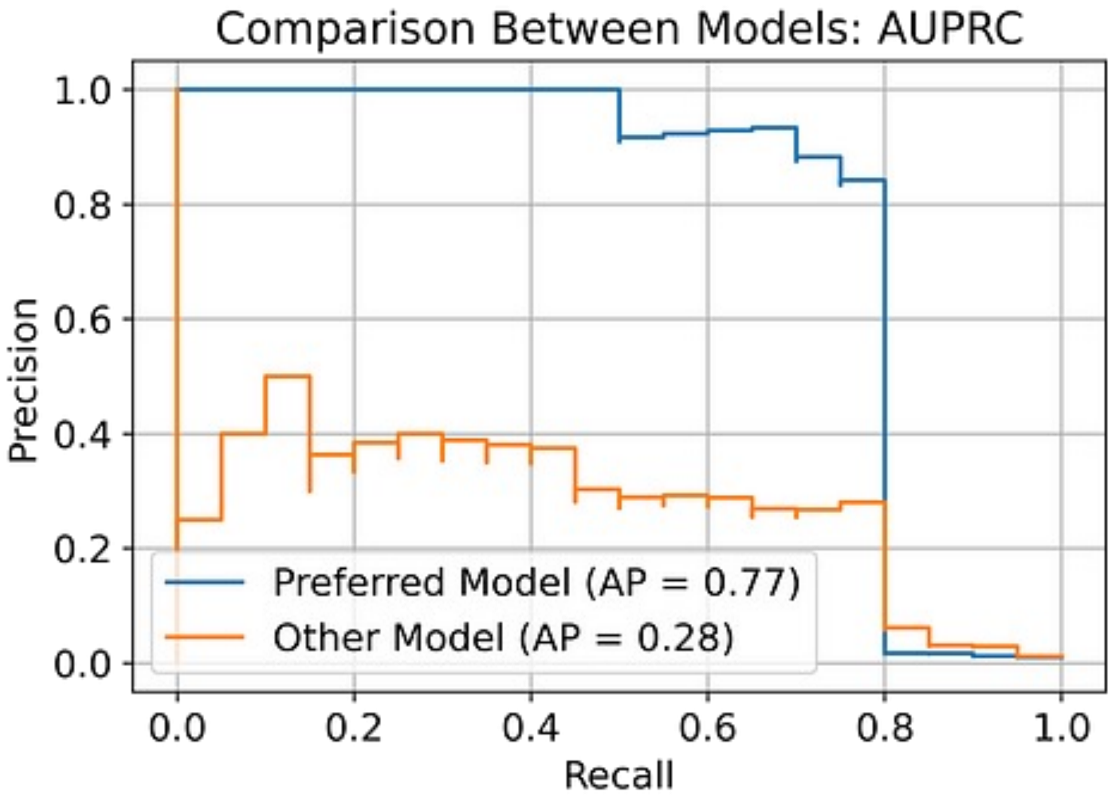
- A high AUPRC represents both high recall and precision
- ROC curves should be used when there are roughly equal numbers of observations for each class.
- Precision-Recall curves should be used when there is a moderate to large class imbalance.



AUROC VS. AUPRC (IMBALANCED DATA)

20 positives; 2000 negatives





https://towardsdatascience.com/imbalanced-data-stop-using-roc-auc-and-use-auprc-instead-46af4910a494

MULTIPLE CLASSES METRICS

- Accuracy:

$$ACC = \frac{TP1 + TP2 + TP3}{\text{Total}}$$

- Macro-average precision:

$$PRE_{\text{macro}} = \frac{PRE_1 + PRE_2 + PRE_3}{3}$$

- Micro-average precision:

$$PRE_{\text{micro}} = \frac{TP1 + TP2 + TP3}{TP1 + TP2 + TP3 + FP1 + FP2 + FP3}$$

	Predicted Class 1	Predicted Class 2	Predicted Class 3
Actual Class 1	TP1		
Actual Class 2		TP2	
Actual Class 3			TP3

(micro-average precision = micro-recall = accuracy)

REGRESSION METRICS

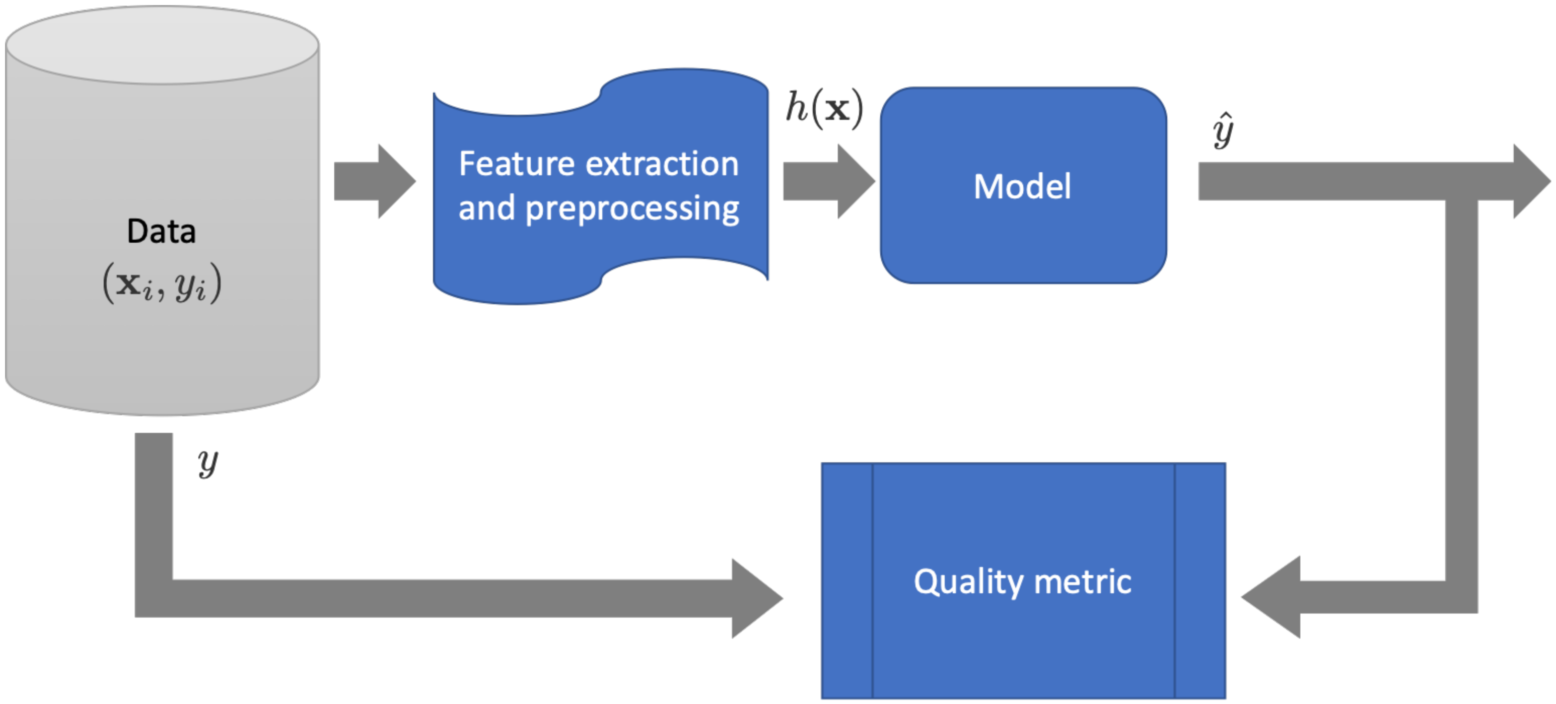
- Mean squared error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

MODEL ASSESSMENT

- Metrics:
  - Classification: Accuracy, precision/recall, AUROC (TPR/FPR), AUPRC (precision/recall)
  - Regression: MSE
- Process: training/test split (holdout), K-fold cross-validation, Monte Carlo cross validation

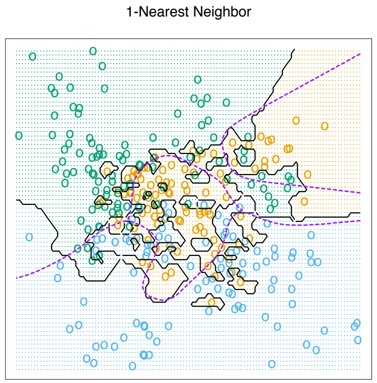
MODEL ASSESSMENT

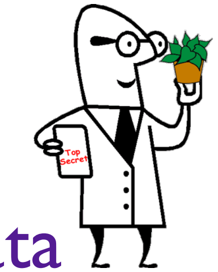


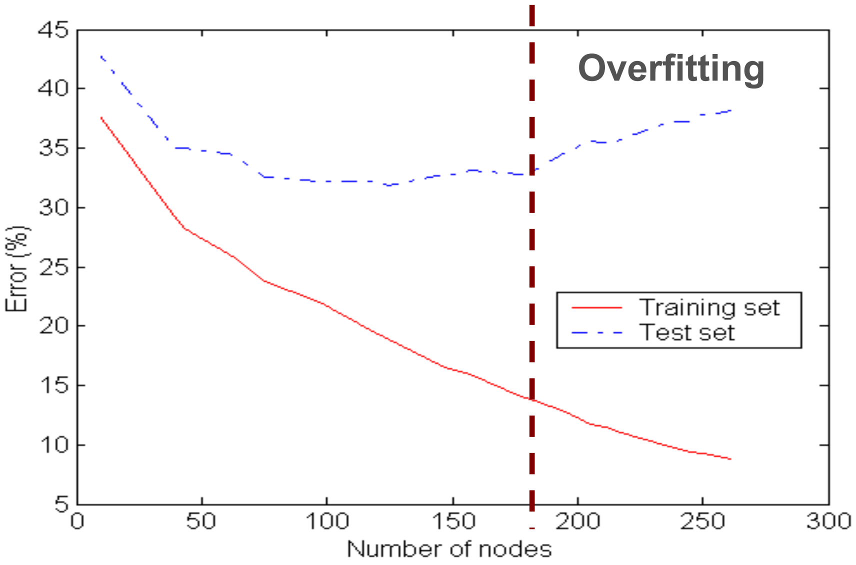
Use all the data to train the model and report the performance on all the data?

THE PITFALL OF TRAINING ERROR

- When the model is overly complex, training error can be 100%
  - kNN when k=1
  - Fully grown tree



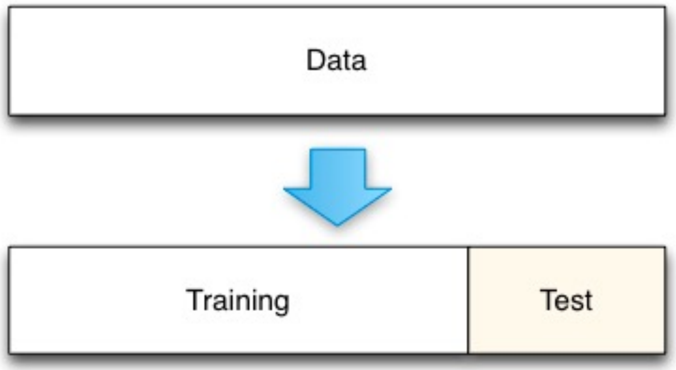




Model "memorizes" the training data
But does not generalize to new data!

HOLDOUT: FORMING A TEST SET

- Hold out some data (i.e., test data) that are not used for training the model
- Proxy for "everything you might see"



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

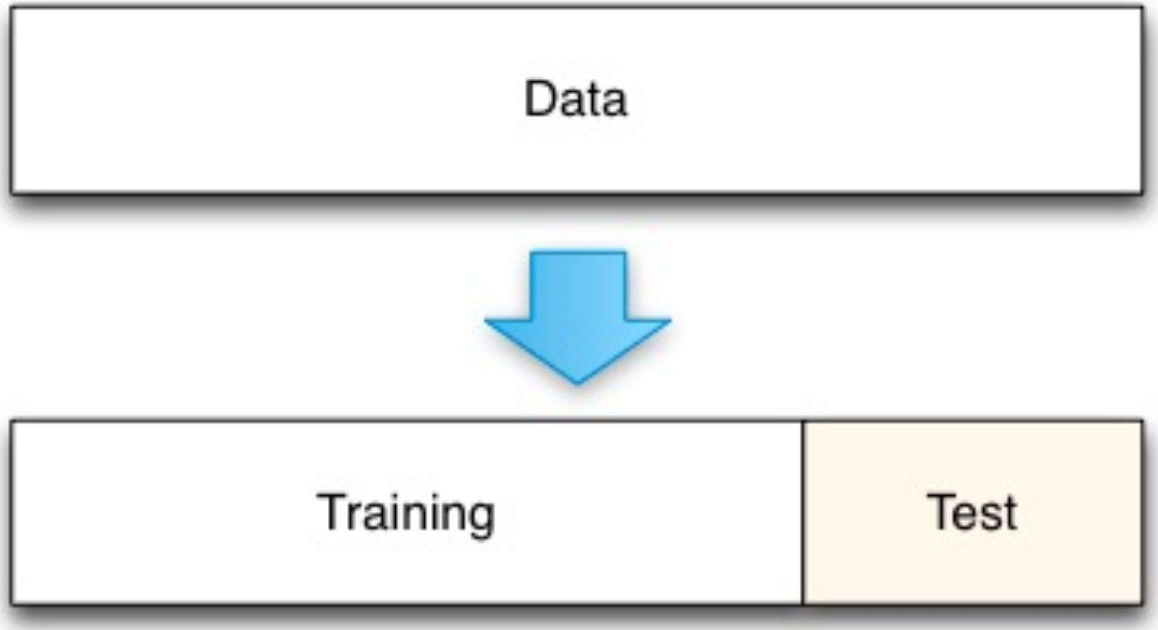
What should the percentage be?

TRAINING/TEST SPLITS

- Too few for training —> unable to properly learn from the data
- Too few for testing —> bad approximation of the true error
- Rule of thumb: Enough test samples to form a reasonable estimate of error

Common split size is 70%-30%

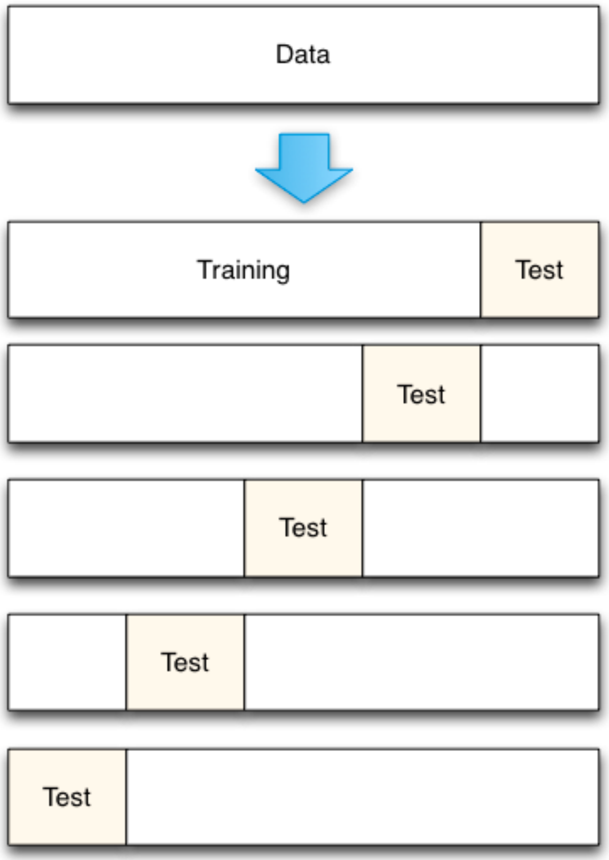
TRAINING/TEST SPLITS



What to do when there isn't enough data? Test data wasted?

K-FOLD CROSS VALIDATION

- Use all the data to train / test
  (but not all at the same time)
- Procedure:
  - Split the training data into K parts or “folds”
  - Train on all but the kth part and validate on the kth part
  - Rotate and report average over K measurements



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

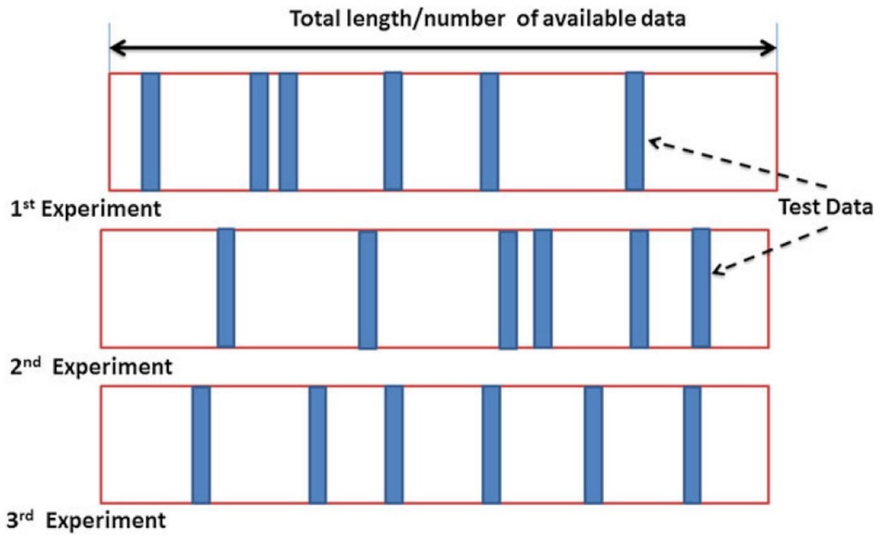
COMMON VALUES OF K

- K = 2 (two-fold cross validation)
- K = 5, 10 (5-fold, 10-fold cross validation) – common practice
- K = N (leave one out cross validation or LOOCV)

Selection is based on how much data you have

MONTE-CARLO CROSS-VALIDATION

- AKA random sub-sampling
  - Randomly select (without replacement) some fraction of your data to form training set
  - Assign rest to test set
  - Repeat multiple times with new partitions
- What are the differences to k-fold cross-validation?

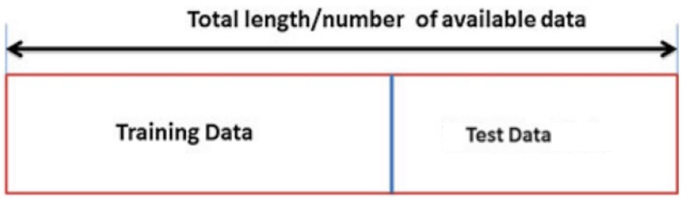


K-FOLD VS MONTE-CARLO

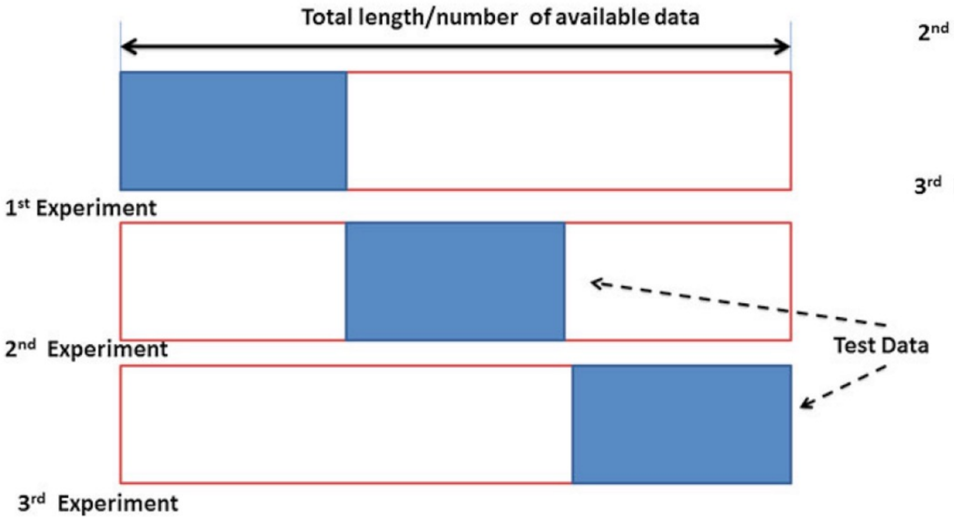
- Cross-validation only explores a few of the possible ways to partition the data
- Monte-Carlo allows you to explore many more possible partitions

ASSESSMENT STRATEGIES

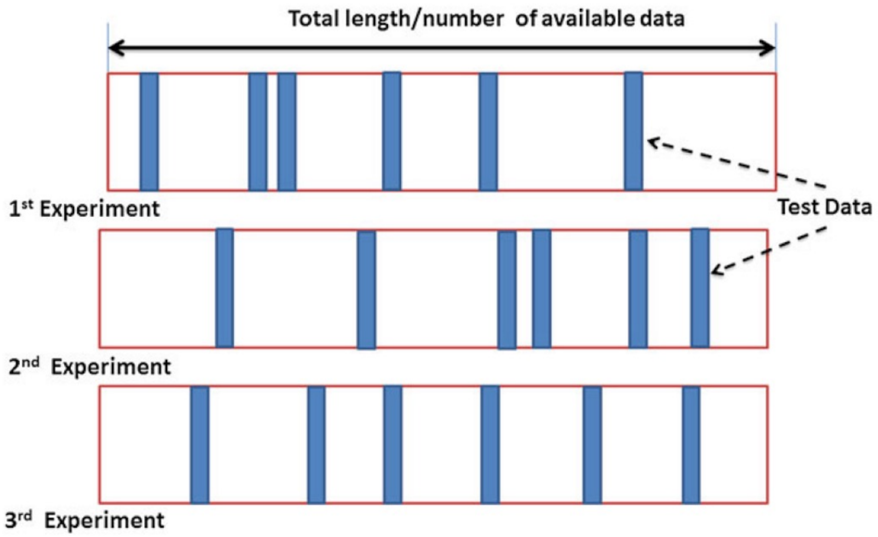
Holdout



k-fold cross-validation



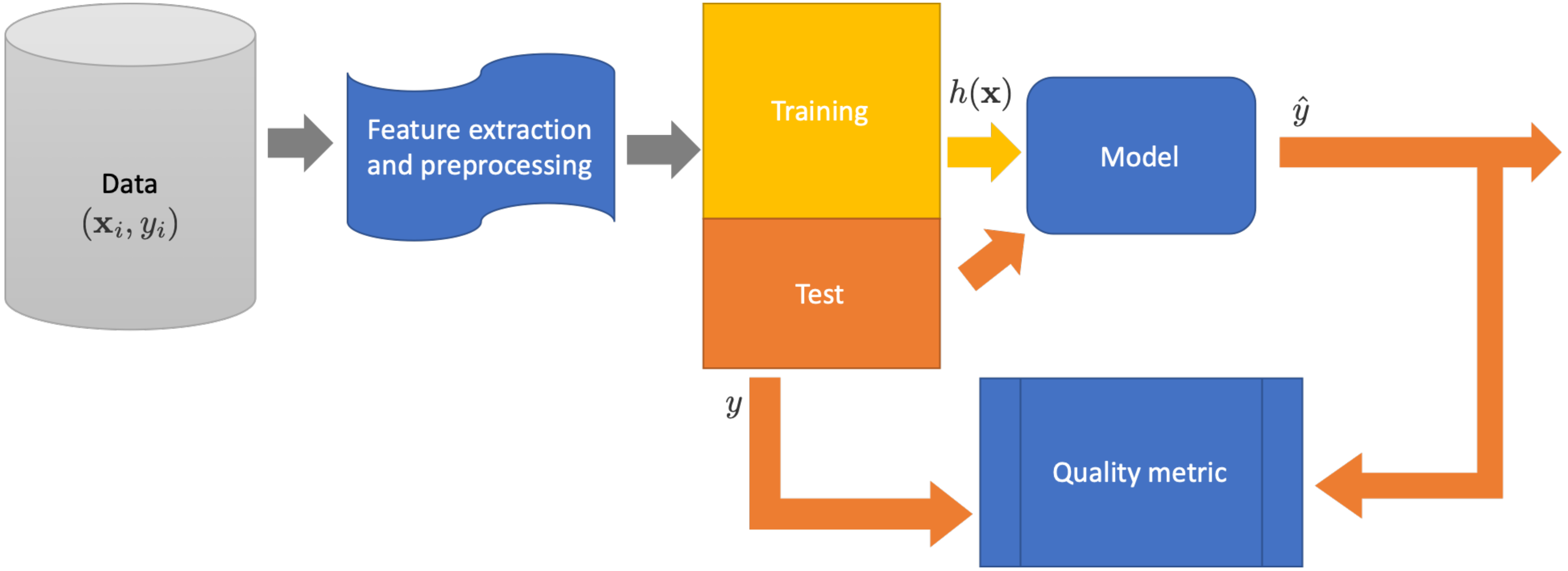
Monte-Carlo cross-validation



Figures 3.6, 3.7, 3.8 (Remesan & Mathew. Hydrological Data Driven Modeling: A Case Study Approach)

MODEL ASSESSMENT: THE PROCESS

- Holdout
- K-fold CV
- Monte-Carlo cross validation



Report the performance on the "test" data

MODEL ASSESSMENT AND MODEL SELECTION

- Model assessment: evaluating a model's performance
- Model selection: selecting the proper level of flexibility for a model (e.g. K for KNN, tree size for decision tree)