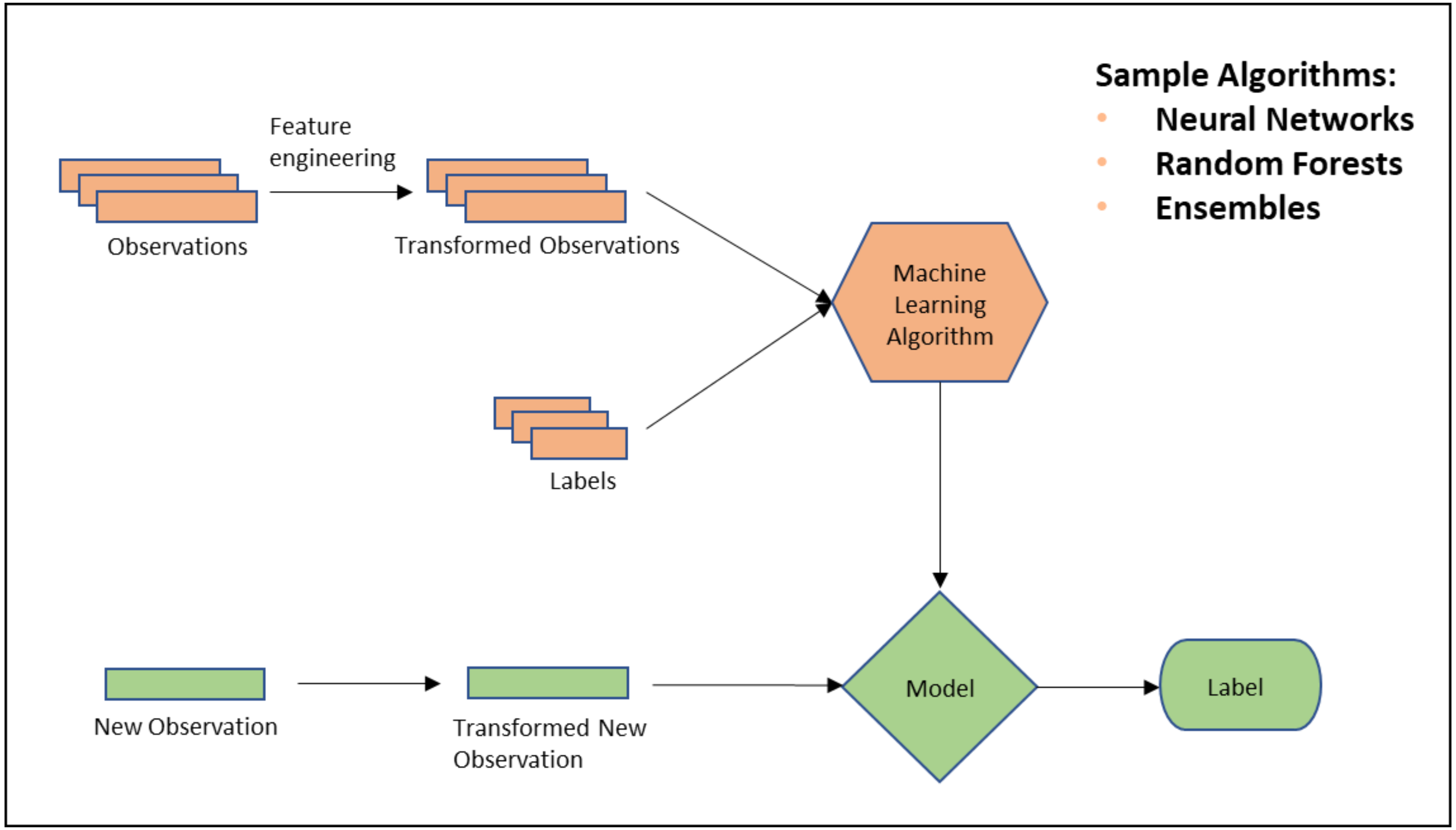
Machine Learning Live
Session #5

Supervised Learning

Supervised learning: each observation is associated with a ***label***

- Try to infer a relationship between the features and labels
- Use the relationship to predict label for a new observation



Classification

- The label (for classification, also called target) is a categorical variable with some number of levels called classes
- Want to predict the class for a new observation

Classification

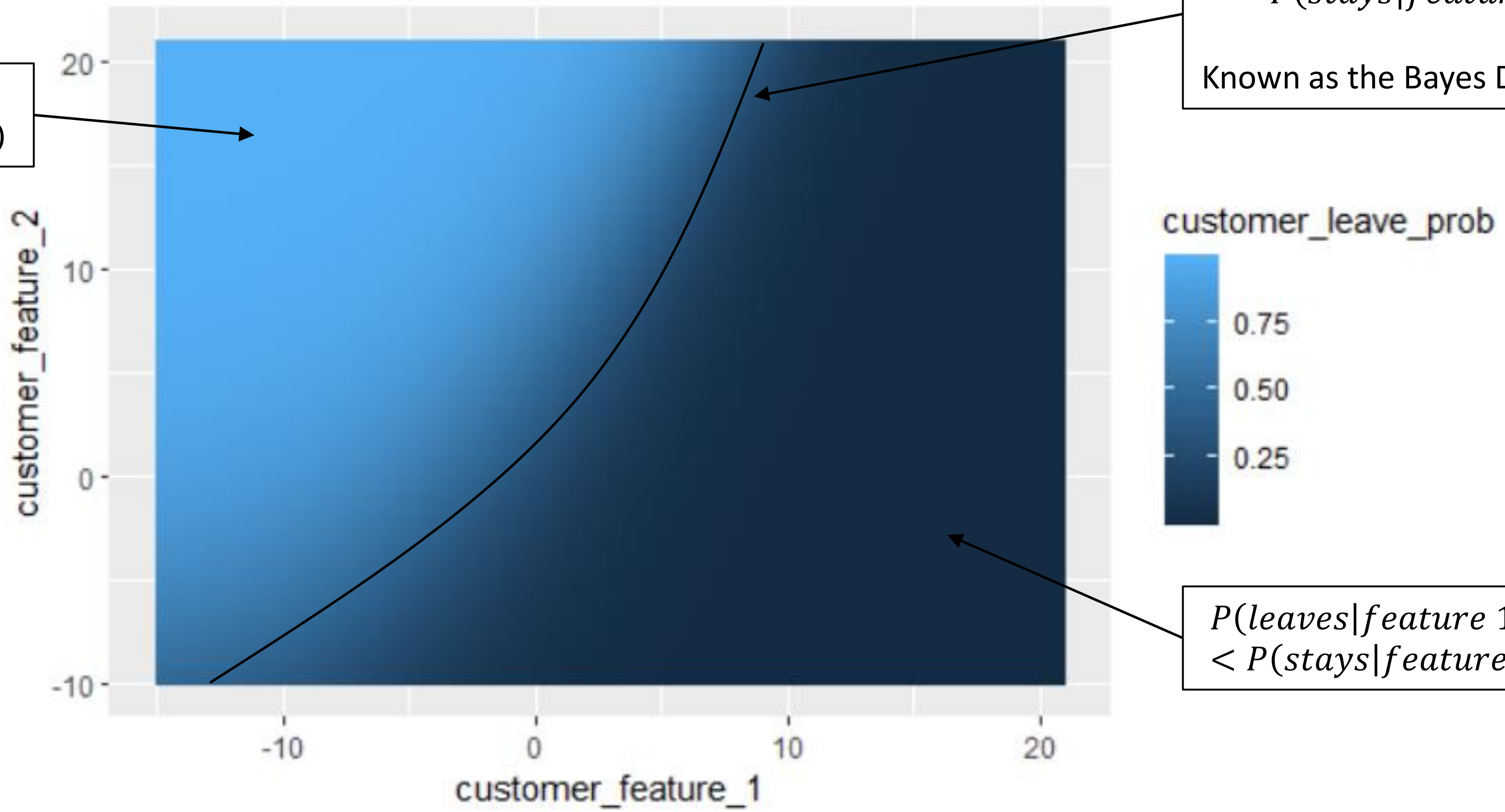
Assume we know the conditional probabilities

$$P(leaves|feature\ 1, feature\ 2)$$

$$P(stays|feature\ 1, feature\ 2)$$

over the entire feature space

$P(leaves|feature\ 1, feature\ 2)$
$> P(stays|feature\ 1, feature\ 2)$



$P(leaves|feature\ 1, feature\ 2)$
$= P(stays|feature\ 1, feature\ 2)$

Known as the Bayes Decision Boundary

$P(leaves|feature\ 1, feature\ 2)$
$< P(stays|feature\ 1, feature\ 2)$

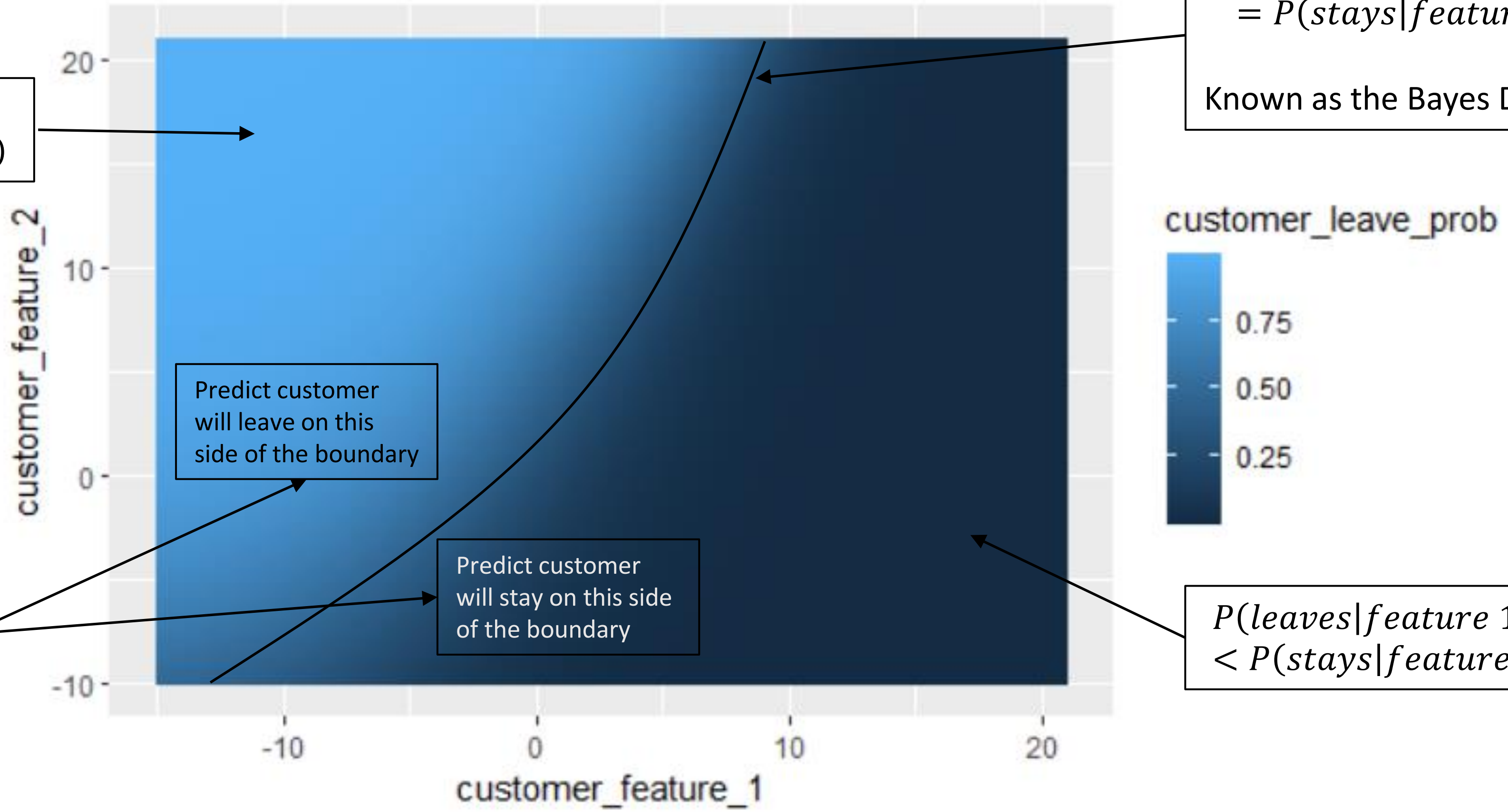
Classification

Assume we know the conditional probabilities

$$P(leaves|feature\ 1, feature\ 2)$$

$$P(stays|feature\ 1, feature\ 2)$$

over the entire feature space



$P(leaves|feature\ 1, feature\ 2) > P(stays|feature\ 1, feature\ 2)$

$P(leaves|feature\ 1, feature\ 2) = P(stays|feature\ 1, feature\ 2)$

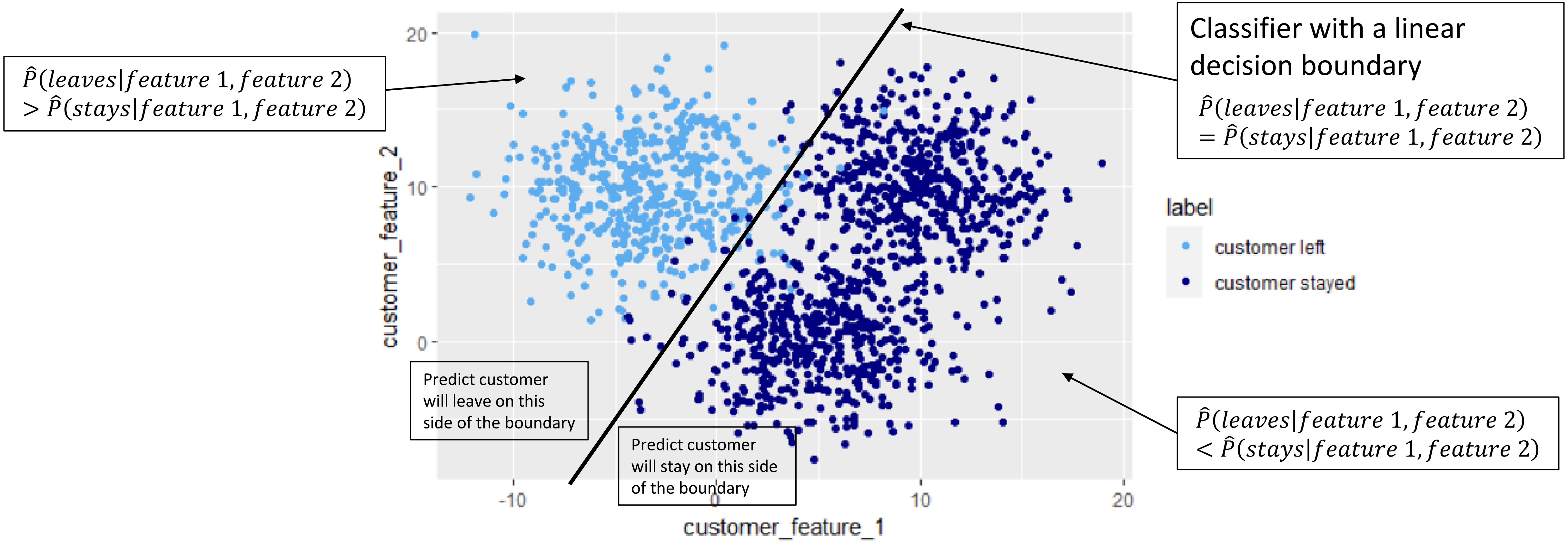
Known as the Bayes Decision Boundary

$P(leaves|feature\ 1, feature\ 2) < P(stays|feature\ 1, feature\ 2)$

The Bayes Classifier

Classification

- In practice, we don't have this information, but we can:
  - Assume there is a conditional probability distribution over the feature space
  - Use a classifier to estimate the conditional probabilities
    - Note, now we have the estimated $\hat{P}$ instead of P



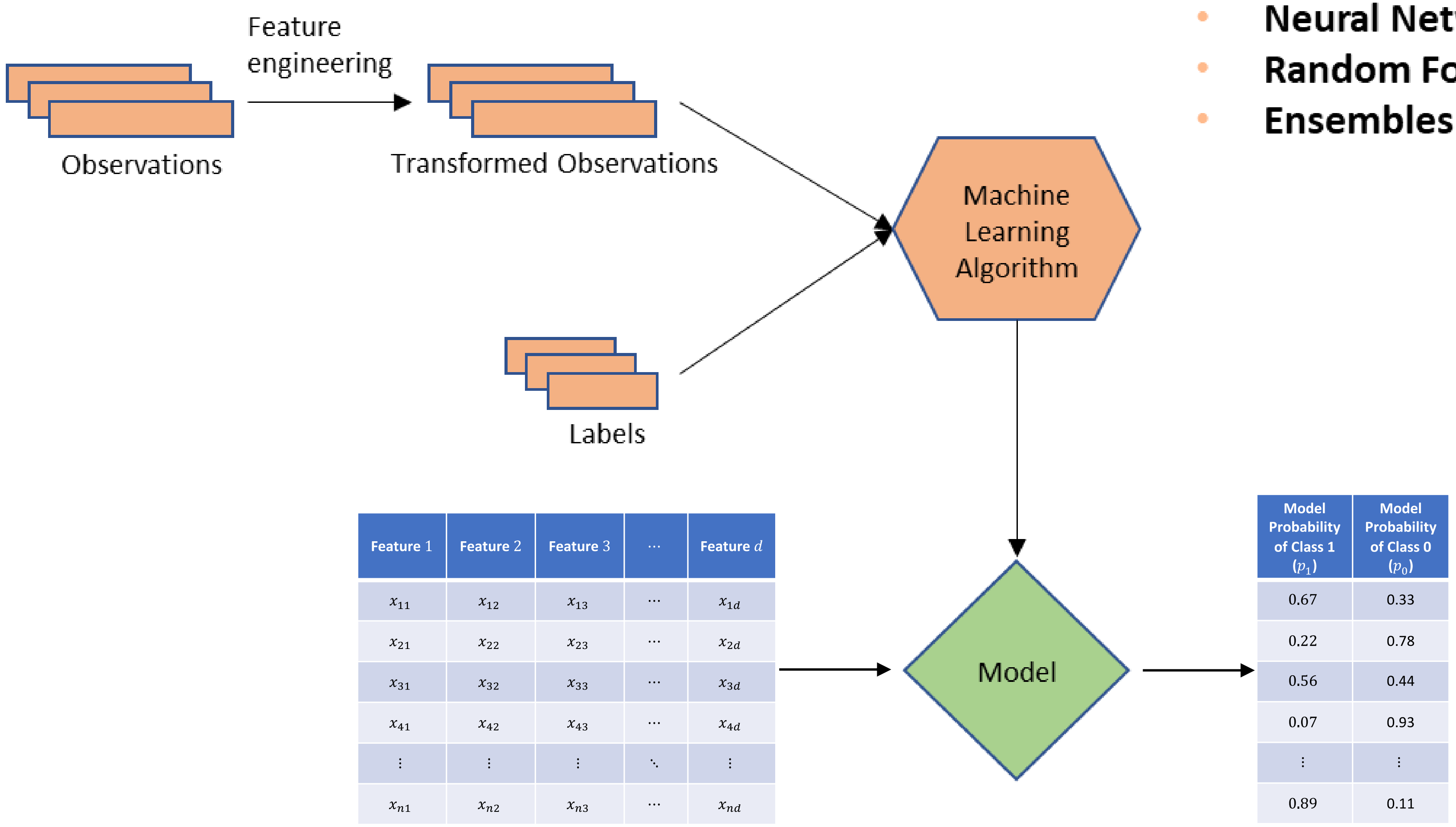
Classification

- Focus on binary classification
  - Target variable has two classes (i.e., levels)
  - Class 1 is called the positive class; class 0 is called the negative class
  - Positive class is the one we are trying to identify
    - E.g., "customer left" in the previous example should be the positive class
      - Want to identify customers that have a high likelihood of leaving
      - Want context such as false positives (the customer has a low chance of leaving, but the model says the chance is high)
      - False positives can be costly → want a model with low rate of false positives
- Extension to multiple classes (> 2) is straightforward and implemented in common languages (R, Python, etc.)

Classification

Sample Algorithms:

- **Neural Networks**
- **Random Forests**
- **Ensembles**



Feature 1	Feature 2	Feature 3	⋯	Feature d
x_{11}	x_{12}	x_{13}	⋯	x_{1d}
x_{21}	x_{22}	x_{23}	⋯	x_{2d}
x_{31}	x_{32}	x_{33}	⋯	x_{3d}
x_{41}	x_{42}	x_{43}	⋯	x_{4d}
⋮	⋮	⋮	⋱	⋮
x_{n1}	x_{n2}	x_{n3}	⋯	x_{nd}

Model Probability of Class 1 (p_1)	Model Probability of Class 0 (p_0)
0.67	0.33
0.22	0.78
0.56	0.44
0.07	0.93
⋮	⋮
0.89	0.11

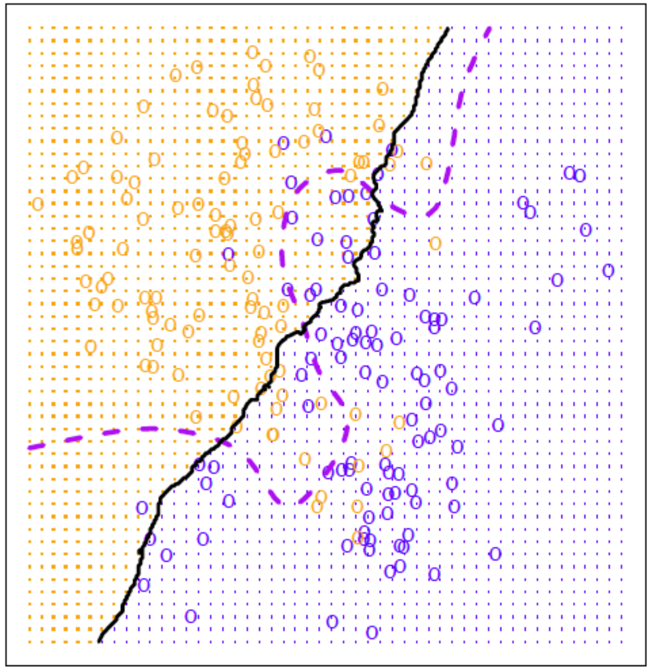
Evaluation Methods for Classification

- Want to evaluate the classifier by assessing how well it predicts the target for new observations
- **Test error** is the average error that results from predicting the target for a new observation
  - Can we calculate the test error on the dataset used to build the model?
  - No, because of overfitting!

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017) *An Introduction to Statistical Learning with Applications in R*. Springer

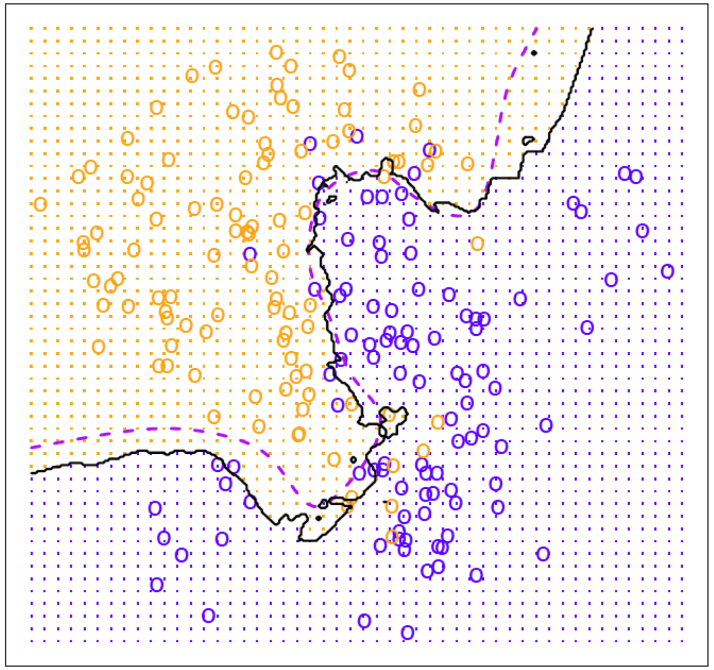
Underfitting vs. Overfitting

Visualizing underfitting and overfitting in classification: a two-dimensional example



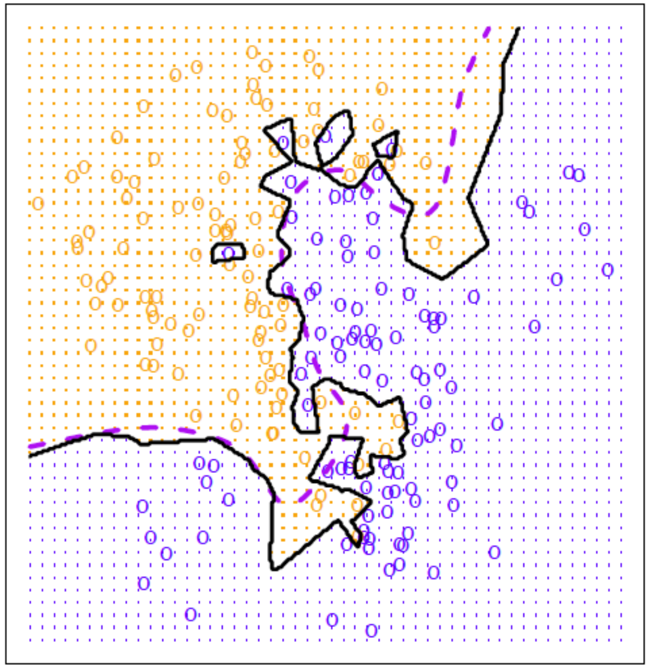
Using 100 nearest neighbors

Underfitting



Using 10 nearest neighbors

Just right!



Using 1 nearest neighbor

Overfitting

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017) *An Introduction to Statistical Learning with Applications in R*. Springer

Evaluation Methods for Classification

- Held-out test set approach
  - Instead of building the model on all available observations, split them into two sets called the training set and the held-out test set (or, simply, test set)
  - Then, build the model using the training set and estimate the test error using the held-out test set
  - Some rules-of-thumb for the split are:
    - 75% training set/25% held-out test set
    - 66% training set/33% held-out test set

Evaluation Methods for Classification

- The focus here is on the evaluation of binary classifiers using a held-out test set
- Main methods for evaluating classifiers are:
  - Accuracy
  - Confusion matrices
  - Receiver operating characteristic (ROC) curves
  - Area under the ROC curve (AUC)
  - Calibration curves

Evaluation Methods for Classification: Accuracy

- Accuracy example for binary classification

Test set →

Feature 1	Feature 2	Feature 3	⋯	Feature d	Target
x_{11}	x_{12}	x_{13}	⋯	x_{1d}	0
x_{21}	x_{22}	x_{23}	⋯	x_{2d}	1
x_{31}	x_{32}	x_{33}	⋯	x_{3d}	1
x_{41}	x_{42}	x_{43}	⋯	x_{4d}	0
⋮	⋮	⋮	⋱	⋮	⋮
$x_{N_{te}1}$	$x_{N_{te}2}$	$x_{N_{te}3}$	⋯	$x_{N_{te}d}$	1

Model Probability of Class 1 (p_1)	Model Probability of Class 0 (p_0)	Model Prediction
0.67	0.33	1
0.22	0.78	0
0.56	0.44	1
0.07	0.93	0
⋮	⋮	⋮
0.89	0.11	1

- $Accuracy = \dfrac{\#\ of\ correct\ predictions\ in\ the\ test\ set}{total\ number\ of\ observations\ in\ the\ test\ set}$
- Why might this be a bad measure of model performance?

Class Imbalance

- What if 98% of the observations in our dataset are of the negative class?
  - Approximately 98% of the training set and 98% of the test set will be of the negative class
  - Model will learn to only predict the negative class
    - Has approximately 98% accuracy $\left(\frac{\#\ of\ correct\ predictions\ in\ the\ test\ set}{total\ number\ of\ observations\ in\ the\ test\ set}\right)$
    - Very bad at identifying the positive class
- Possible solutions:
  - Under-sampling
  - Over-sampling (SMOTE)
  - Penalize a misclassification of the positive class
  - See Chapter 16 of Applied Predictive Modeling for a good discussion of methods

Evaluation Methods for Classification: Confusion Matrices

- Confusion matrices are a common visualization of model performance

Actual Class

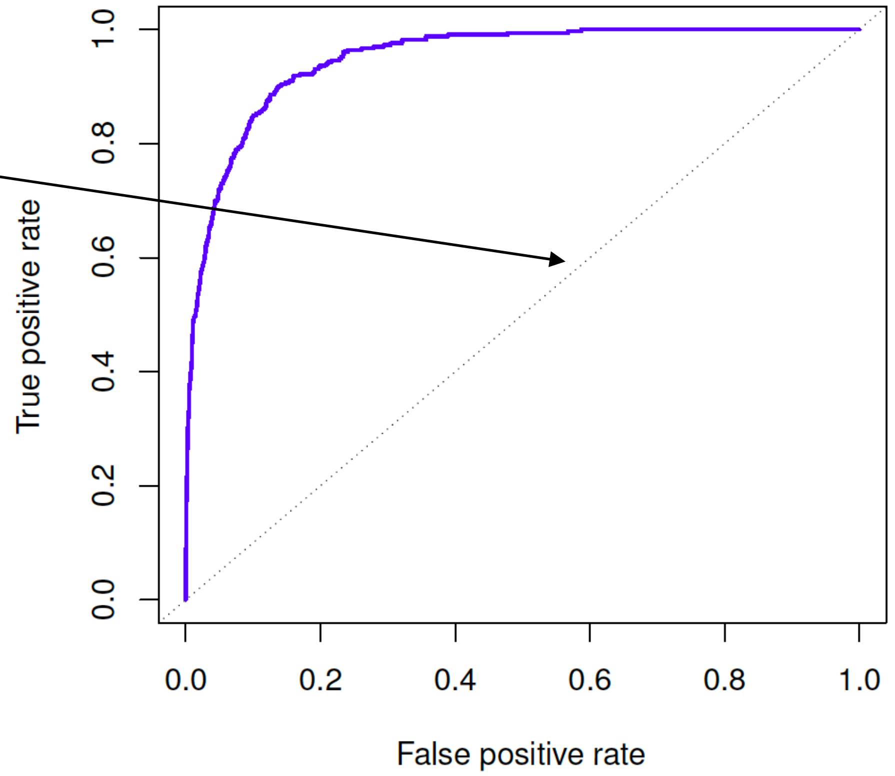
		1	0
Predicted Class	1	True Positive (TP)	False Positive (FP)
	0	False Negative (FN)	True Negative (TN)

  - False positive rate (FPR): $\frac{FP}{FP+TN}$
  - True positive rate (TPR): $\frac{TP}{TP+FN}$

Evaluation Methods for Classification: ROC Curves and AUC

- FPR and TPR can be calculated after applying any threshold $0 \leq t \leq 1$
  - For a threshold t, predict the positive class when $p_1 \geq t$, and the negative class otherwise
- Plotting FPR vs TPR after applying a threshold t ranging from 1 to 0 yields the ROC curve

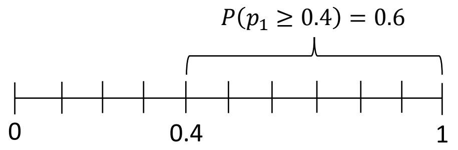
The dashed diagonal line corresponds to random guessing.



- Curves that "hug" the top left corner correspond to good classifiers
- The area under the curve (AUC) is a useful way to summarize the ROC curve.
  - A curve that "hugs" the top left corner will have an AUC close to 1
  - The largest AUC value possible is 1

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017) *An Introduction to Statistical Learning with Applications in R*. Springer

Random Guessing



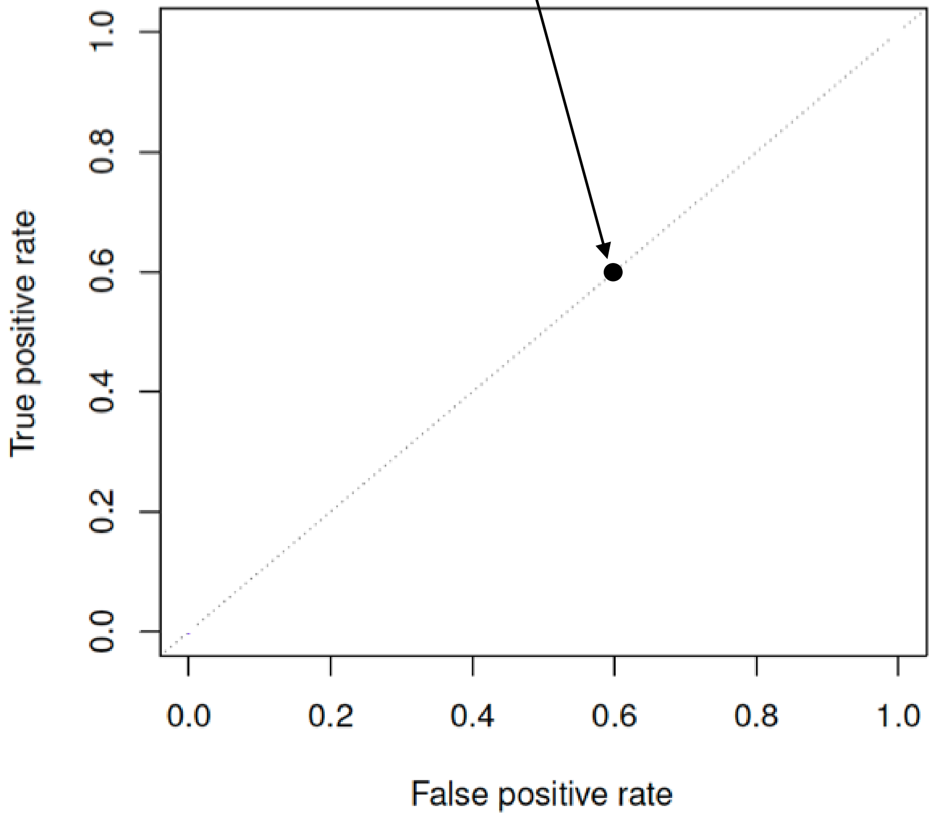
We assign $U(0,1)$ random numbers to the model confidences.

Threshold value of 0.4: when $p_1 \geq 0.4$, predict class 1. Since $p_1 \sim U(0,1)$, we expect 60% of the true negatives to be classified as positives (false positive rate is 0.6) and 60% of the true positives to be classified as positives (true positive rate is 0.6).

Feature 1	Feature 2	Feature 3	$\cdots$	Feature d	Target
$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\cdots$	$x_{1,d}$	0
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\cdots$	$x_{2,d}$	1
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\cdots$	$x_{3,d}$	1
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$\cdots$	$x_{4,d}$	0
$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$\cdots$	$x_{5,d}$	1
$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$\cdots$	$x_{6,d}$	1
$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$\cdots$	$x_{7,d}$	0
$x_{8,1}$	$x_{8,2}$	$x_{8,3}$	$\cdots$	$x_{8,d}$	1
$x_{9,1}$	$x_{9,2}$	$x_{9,3}$	$\cdots$	$x_{9,d}$	0
$x_{10,1}$	$x_{10,2}$	$x_{10,3}$	$\cdots$	$x_{10,d}$	0

Test set

Model Probability of Class 1 (p_1)	Model Probability of Class 0 (p_0)
u_1	$1 - u_1$
u_2	$1 - u_2$
u_3	$1 - u_3$
u_4	$1 - u_4$
u_5	$1 - u_5$
u_6	$1 - u_6$
u_7	$1 - u_7$
u_8	$1 - u_8$
u_9	$1 - u_9$
u_{10}	$1 - u_{10}$



James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017) *An Introduction to Statistical Learning with Applications in R*. Springer

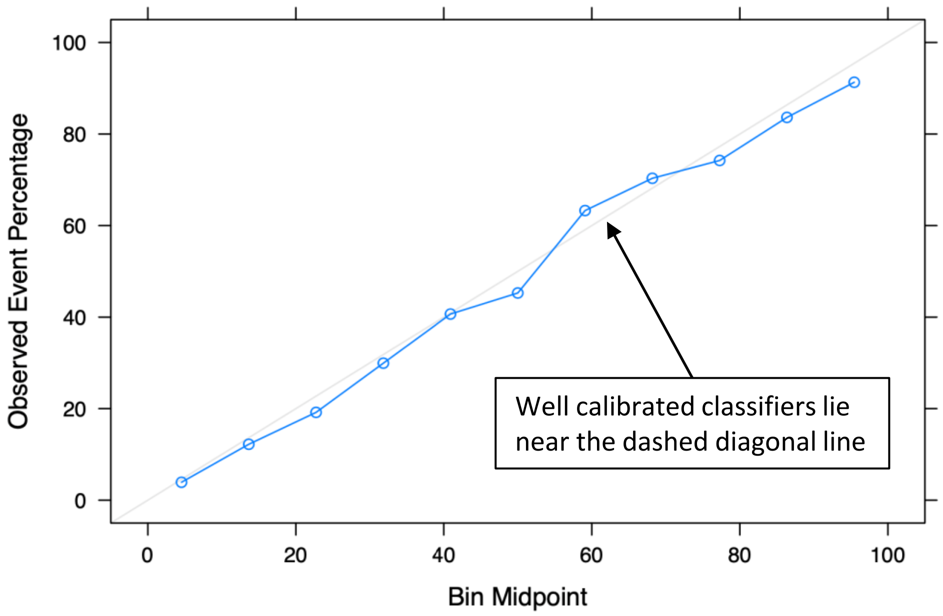
Evaluation Methods for Classification: Calibration Curves

- How well do the predicted probabilities match the actual performance of the model?
- To calculate the calibration curve:
  - Define intervals, e.g., [0,0.1), [0.1,0.2), [0.2,0.3), [0.3,0.4), [0.4,0.5), [0.5,0.6), [0.6,0.7), [0.7,0.8), [0.8,0.9), [0.9,1.0]
  - Assign each test set observation to the interval that contains its p_1 value
  - Calculate $Observed\ Event\ Percentage$: the proportion of test set observations in I of the positive class

Feature 1	Feature 2	Feature 3	⋯	Feature d	Target
x_{11}	x_{12}	x_{13}	⋯	x_{1d}	0
x_{21}	x_{22}	x_{23}	⋯	x_{2d}	1
x_{31}	x_{32}	x_{33}	⋯	x_{3d}	1
x_{41}	x_{42}	x_{43}	⋯	x_{4d}	0
⋮	⋮	⋮	⋱	⋮	⋮
$x_{N_{te}1}$	$x_{N_{te}2}$	$x_{N_{te}3}$	⋯	$x_{N_{te}d}$	1

Test set

Model Probability of Class 1 (p_1)	
0.67	$[0.6, 0.7)$
0.22	$[0.2, 0.3)$
0.56	$[0.5, 0.6)$
0.07	$[0, 0.1]$
⋮	
0.89	$[0.8, 0.9)$



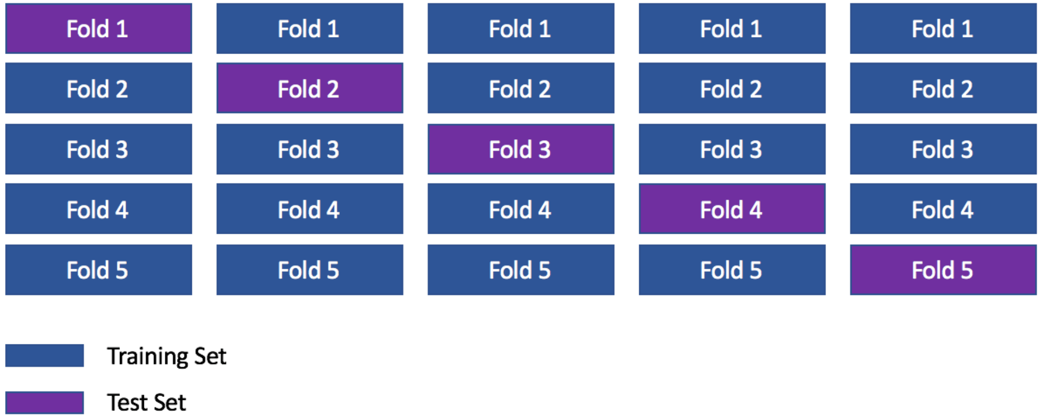
- Plot the $Bin\ Midpoint$ vs $Observed\ Event\ Percentage$

Evaluation Methods for Classification

- Drawbacks of the held-out test set approach
  - Held-out test error may over-estimate the test error that would have resulted from building the model using the entire set of available observations
  - Held-out test error can have high variance, since the observations in the training set can be very different from the observations in the held-out test set
    - Due to an unlucky split (training observations are not representative of the general population)
- k-fold cross-validation is an alternative evaluation approach

Evaluation Methods for Classification

- k-fold cross-validation (CV)
  - Partition the entire set of available observations into k equally-sized groups $G_1, G_2, \dots, G_k$
  - For each group G_i:
    1. Train the model using all other groups $G_1, G_2, \dots, G_{i-1} G_{i+1}, G_{i+2}, \dots, G_k$
    2. Test the model on group G_i



- More data is used for training (as long as k isn't too small)
- Each observation gets a chance to be in the training set AND the test set

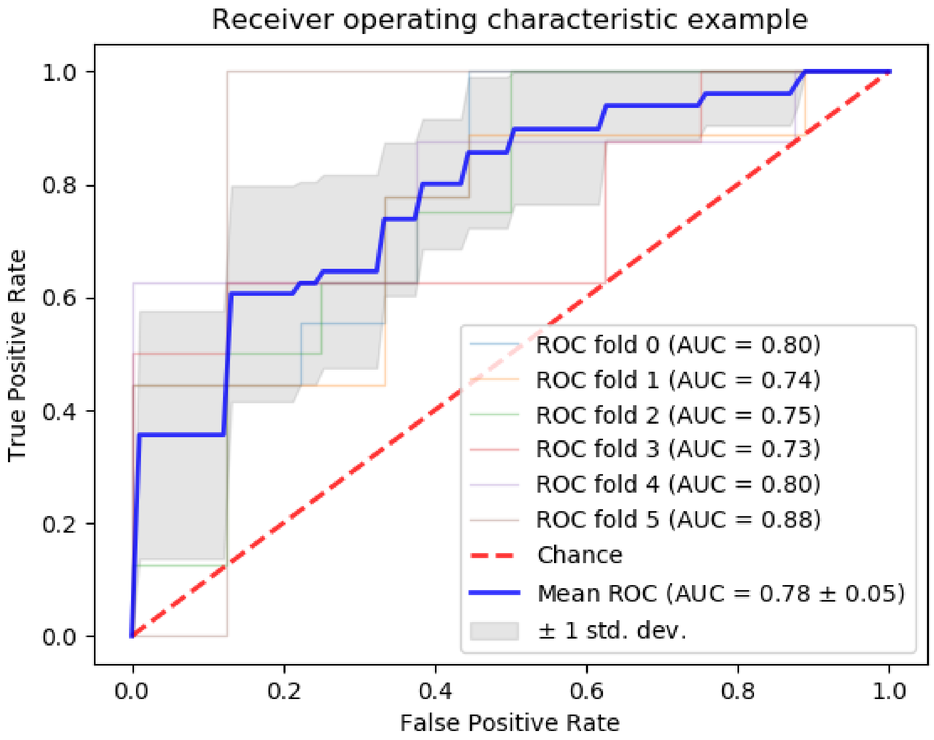
King, R. S. (2019) *Cross-Validation in Scikit-Learn*. https://randalscottking.com/2018/10/cross-validation-in-scikit-learn/

Evaluation Methods for Classification

- 2-fold CV
  - Like the held-out test set approach, often has high bias and can have high variance
- n-fold CV (where n is the number of available observations) is called "leave-one-out CV" (LOOCV)
  - Smallest possible bias for cross-validation, since all but one observation is used for training
  - High variance, since the average is taken over highly correlated estimates
    - The estimates are highly correlated since they are all based on almost the same training set
- $k = 5$ or 10 is typically used and is a good balance of bias and variance
- Main drawback of CV: time consuming

Evaluation Methods for Classification

The evaluation methods we discussed for the held-out test set approach can be applied to CV by averaging



Scikit-Learn Documentation (2019) *Receiver Operating Characteristic (ROC) with Cross-Validation*.