STAT5003

Week 6 : Cross Validation

Jaslene Lin
The University of Sydney

THE UNIVERSITY OF SYDNEY

Readings and R functions covered

Important

- **Introduction to Statistical Learning**
  - ➠ Cross validation covered in Chapter 5
- **R** functions
- `caret::createDataPartition`
  - ➠ `caret::train`
  - ➠ `caret::confusionMatrix`
  - ➠ `pROC::roc`
  - ➠ `pROC::auc`

Resampling Methods

📊 Resampling Methods in Statistical Learning

- **Resampling methods** are techniques that involve repeatedly drawing samples from a training data and refitting the model on each (re)sample in order to obtain additional information about the fitted model.

🧪 Two Main Resampling Methods

1. **Cross-Validation (CV)**
    - Used to estimate **test error** and **select tuning parameters**
2. **Bootstrap**
    - Used primarily to **assess variability**, e.g., standard errors and confidence intervals

Training error vs test error

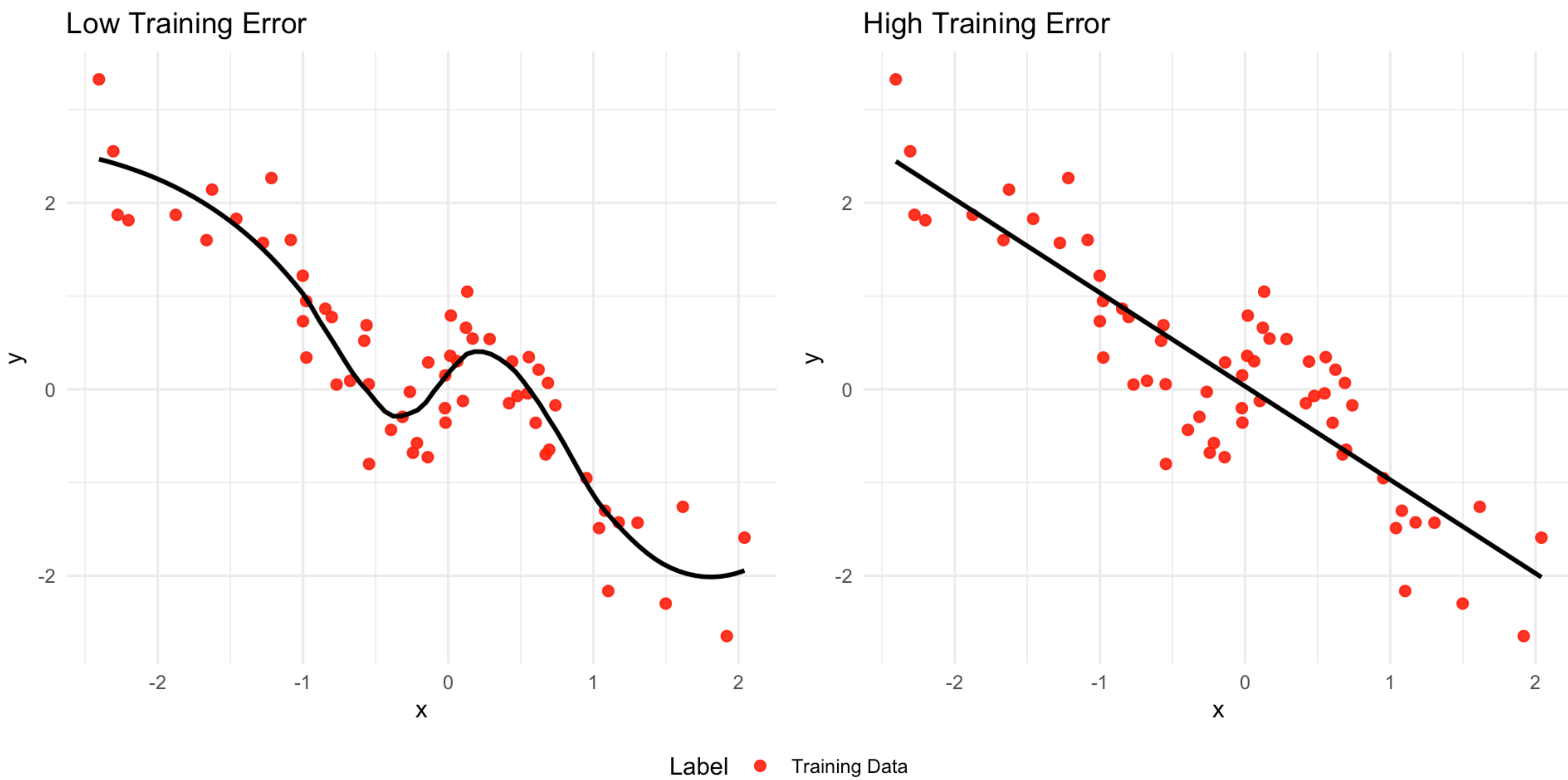
Training error vs test error

Training error is the performance metric applied to the observations used to train the model.

Test error is the average error when applying a model to predict the response on new (test) observations that were not used in the training of the model.

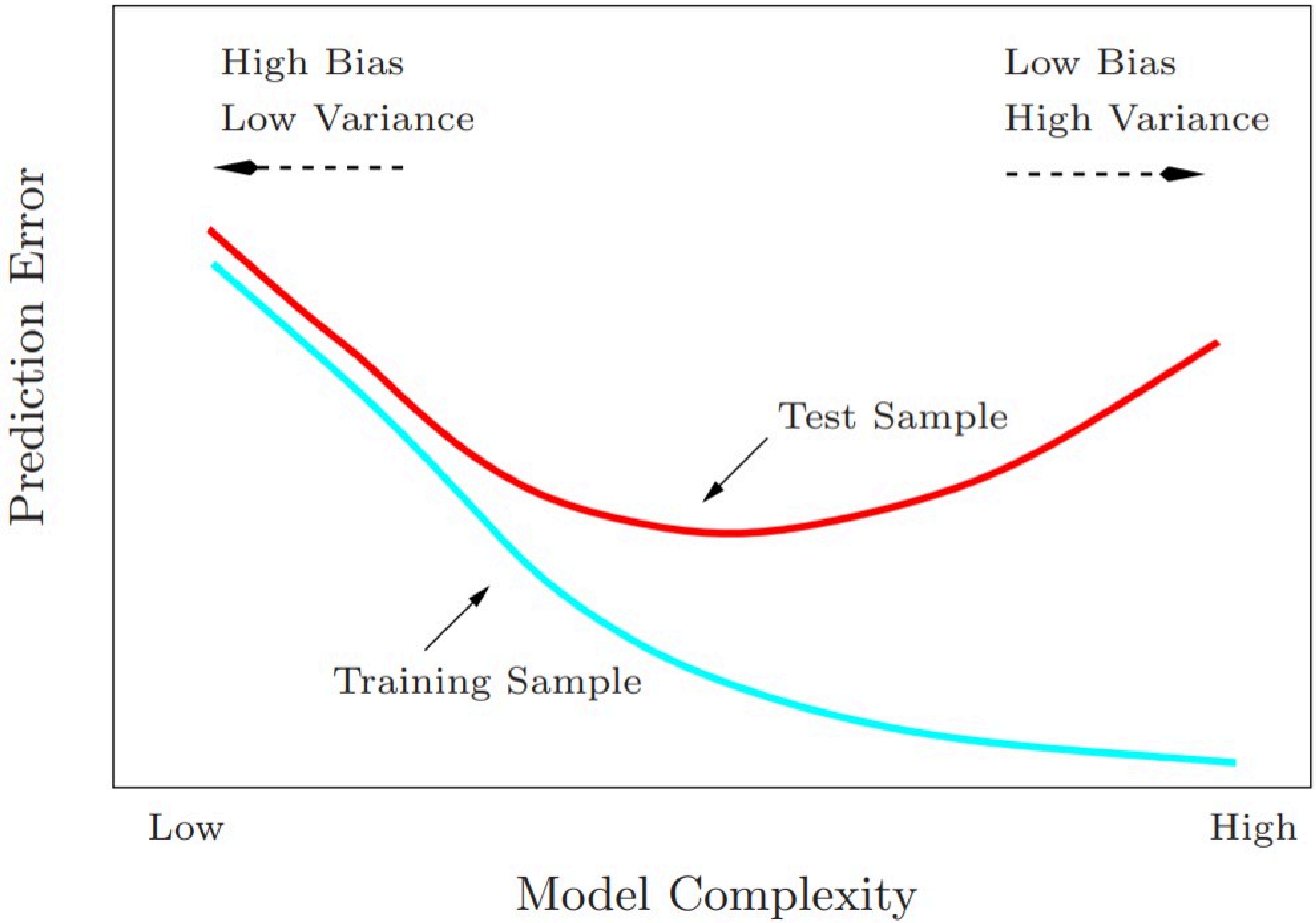
- Training error is usually very different in magnitude to the test error.
    - ➠ Training error can **underestimate** the test error.

Pick the better model



Pick the better model

Training error versus Test error



Strategies to estimate the test error

Desirable but not attainable (gold standard)

```
-   Use a large designated test set (often not available)
```

Adjust the training error to estimate the test error

```
-   Common to add a penalty term to the model
    -   Bayesian information criterion (BIC)
    -   Adjusted $R^2$
```

Cross validation

```
-   Remove or hold out a subset of observations (test set) and use
    the rest to train the model
-   Assess model performance on the test set
```

Test set approach

- Here, we randomly divide the available set of samples into two:
  - ➠ a training set
  - ➠ a test set
- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the test set
- The resulting test-set error provides an estimate of the test error
- Typically assessed using
  - ➠ **MSE** in the case of a quantitative response (regression)
  - ➠ **Misclassification rate** in the case of a qualitative (discrete) response (classification)

Example of the training and test split

- Random split of the data into two halves
    - The left is the training indices
    - The right is the test indices

Drawbacks of test set approach

- The estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the test set.
- In the test set approach, only a subset of the observations are used to fit the model.
    - ➠ This suggests that the test set error may tend to **overestimate** the test error for the model fit on the entire data set.

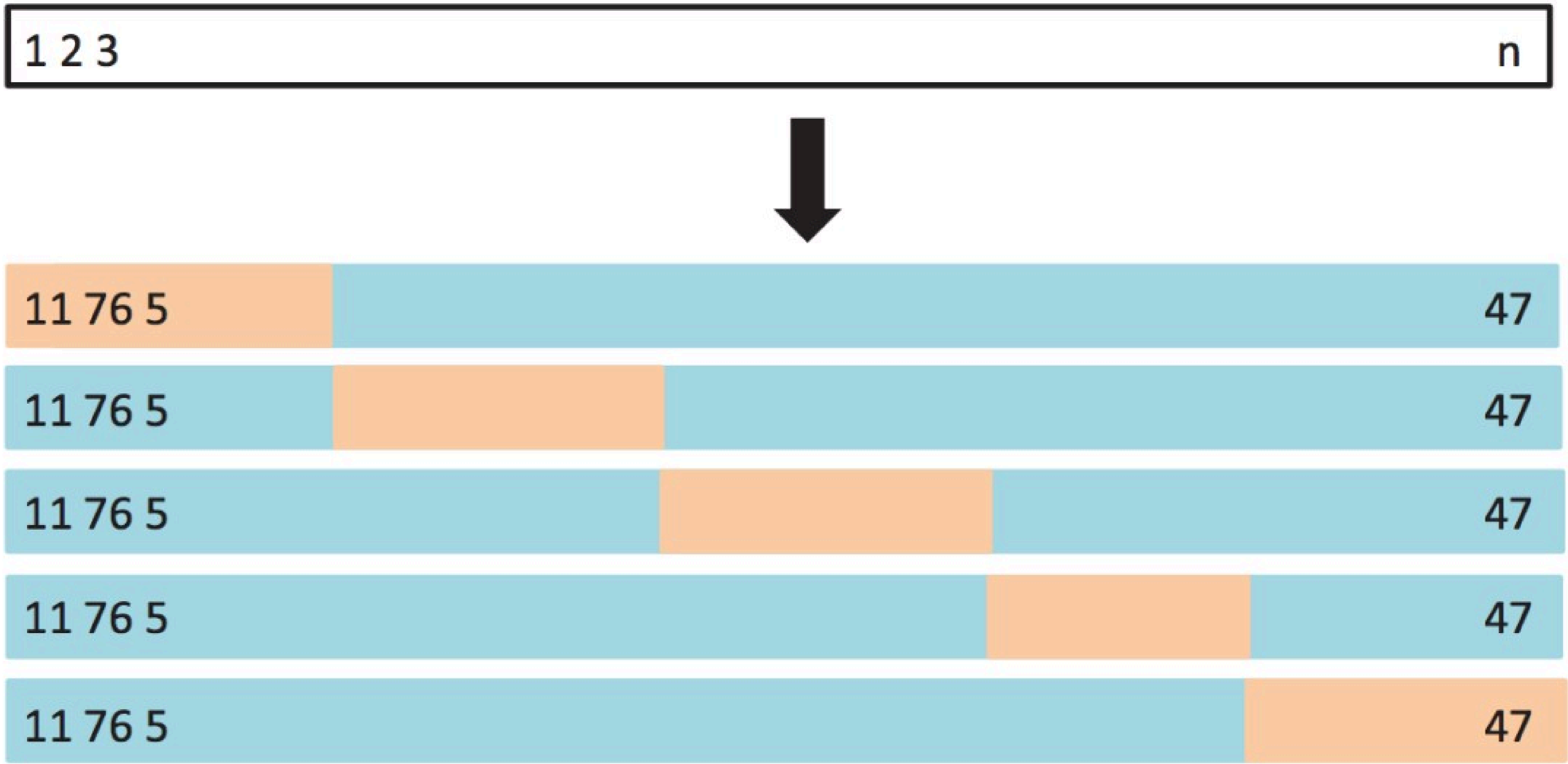
k-fold and repeated cross validation

k-fold cross validation

- Widely used approach for estimating test error.
  - ➟ Estimates can be used to select best model, and to give an idea of the test error of the final chosen model.
- Idea is to randomly divide the data into k equal-sized parts and the procedure is repeated k times:
  - ➟ each time, the first fold is treated as a test set, and the method is fit on the remaining $k-1$ folds.
  - ➟ the MSE_i is computed at each iteration.
- This process results in k estimates of the test error, MSE_1, MSE_2, MSE_3, ..., MSE_k
  - ➟ the k-fold CV estimate is computed by averaging these values

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i$$

Example: 5-fold

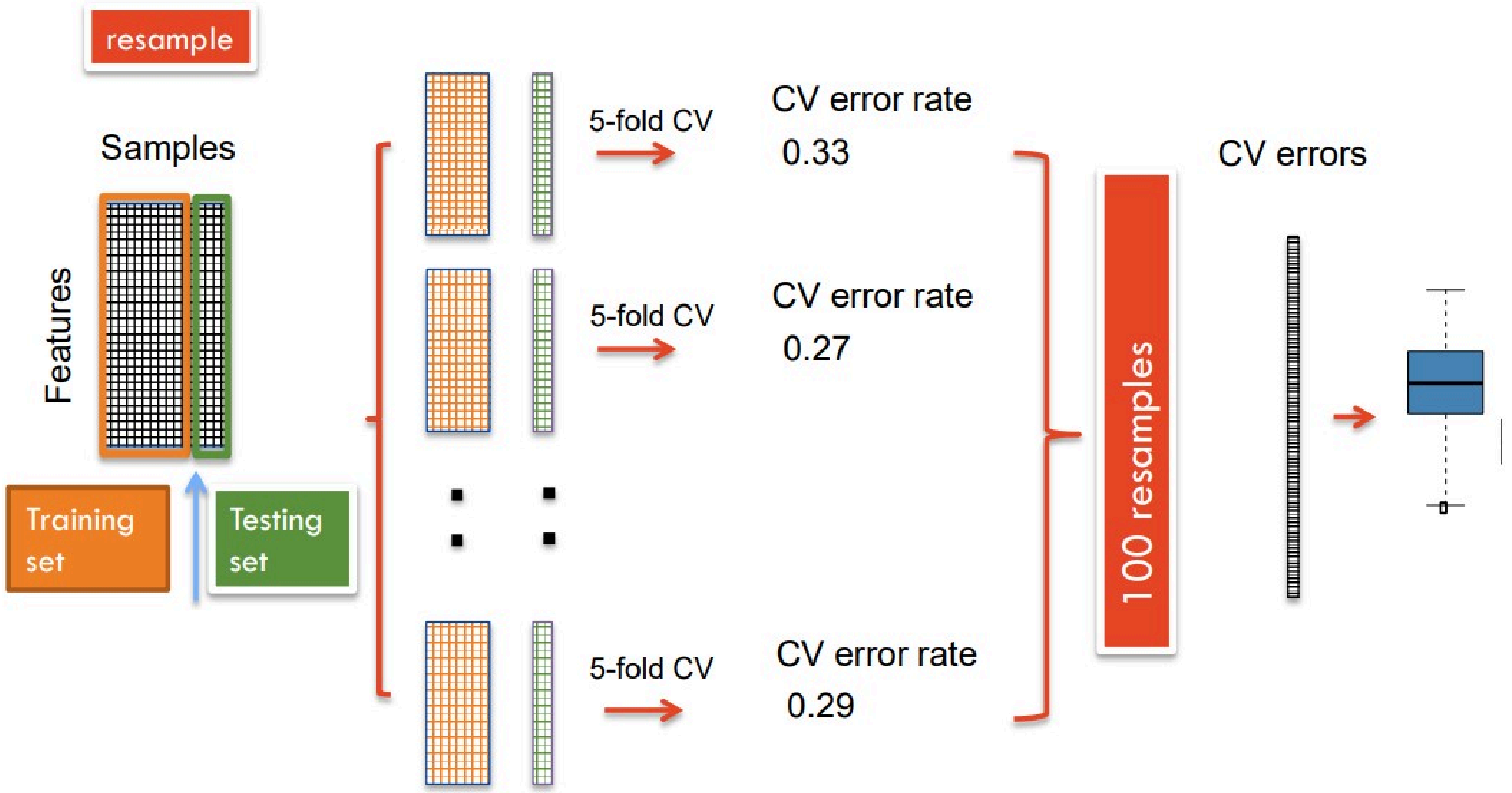


Repeated Cross Validation

It repeats the k-fold CV process multiple times, each with different random splits.

This helps to: provide a less biased CV test error estimate. provide the variance of the CV error.

It comes with **a computational cost**.



❓ Cross-Validation Example: What's Wrong Here?

Scenario:

- You are working with a **high-dimensional dataset** (many more features than observations).
- All variables are **numeric**, and you need to perform **dimension reduction** before modeling.

CV Procedure Used:

1. Compute the **correlation** between each predictor and the response variable **using the entire dataset**.
2. Select the **top 50 variables** with the highest correlation.
3. Use these 50 variables as features and perform **k-fold cross-validation** to evaluate the model.

⚠️ Does this approach seem okay?

- All steps *seem* logical, but something subtle is going wrong...

🚫 What’s the Problem with This Approach?

Data Leakage (Information Leak)

- You’re using the **entire dataset**, including the **test sets**, to select the top 50 features.
- This means the test set is influencing the training process — even before **the model sees it**.

🎯 Why This Is a Problem:

- Violates a key principle of cross-validation:
  🔒 **The test data should be kept completely separate** from training and preprocessing.
- Leads to an **optimistic estimate** of test error.
- In high-dimensional settings, this can cause **severe overfitting** to noise.

✅ Correct Approach:

- Perform **feature selection inside each fold**, using **only the training portion**.
- Then evaluate the model on the test set with those selected features.

✅ Correct Approach:

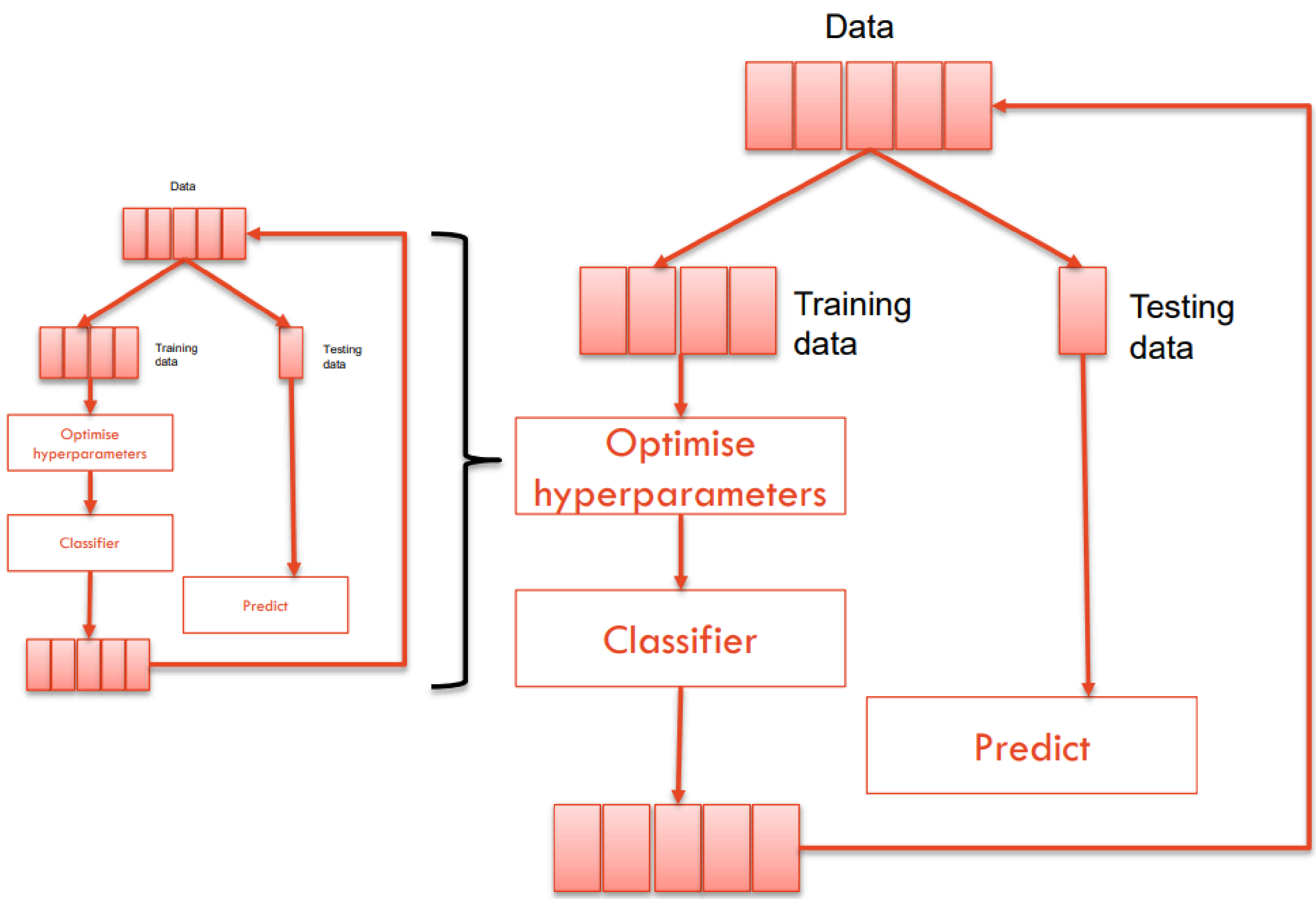
- Split the dataset into k folds
- For each iteration,
  - ➟ Perform **feature selection inside each fold**, using **(k-1) folds**.
  - ➟ Train your model using the selected variables above.
  - ➟ Run your model and record accuracy against the k^{th} fold.

Other information leakage to check

- Other things you should do it within with CV loop
  - ➟ Feature selection
  - ➟ Hyperparameter optimization
  - ➟ Missing data imputation

Nested cross validation

Assessing model performance while tuning hyperparameters



Model Selection

- The reason for doing cross-validation is to evaluate the different models by estimating their performance on unseen data
- Example: If you need to choose between kNN, LDA, logistic regression, and SVM, then you can run each of these classification algorithms with cross-validation, and pick the one with the highest CV accuracy
- But then, you can go back to use all the data to build a final model

Classification evaluation metrics

Confusion Matrix

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

- True positive: positive class and predicted to be positive class
- False positive: negative class but predicted to be positive class
- False negative: positive class but predicted to be negative class
- True negative: negative class and predicted to be negative class

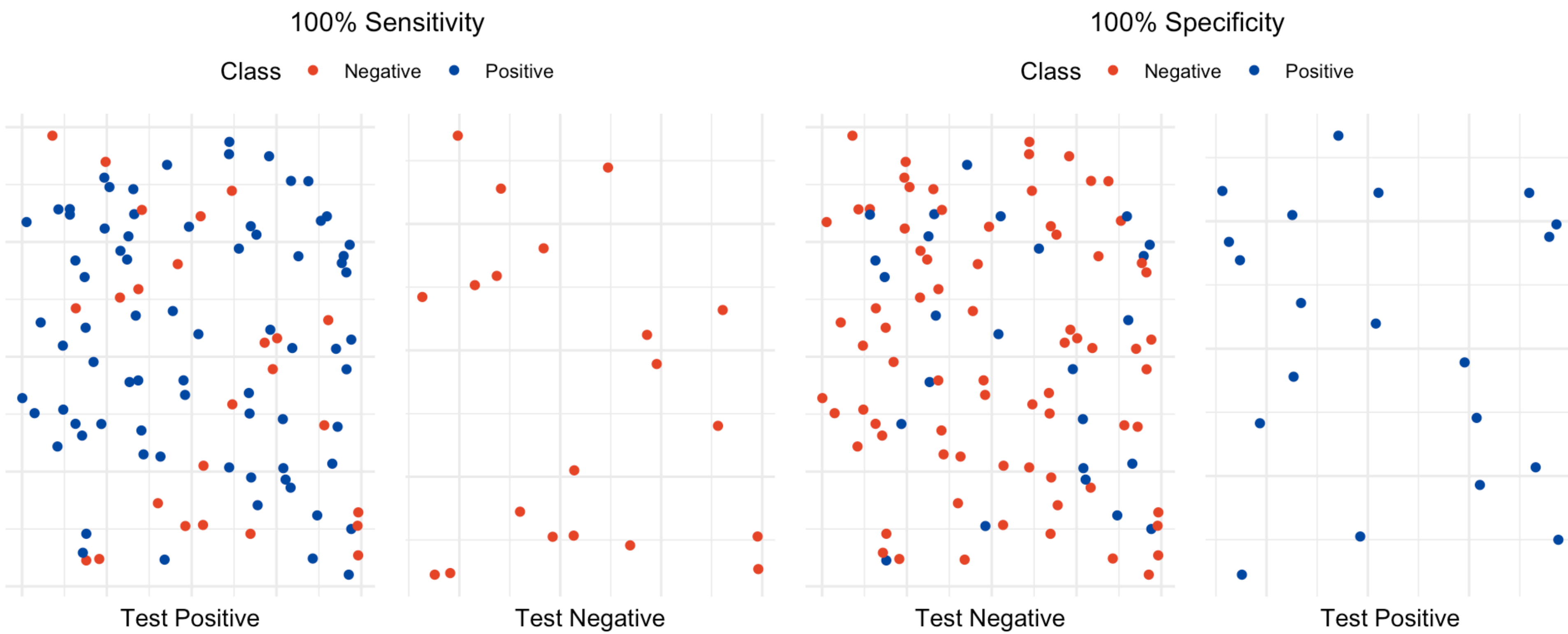
Accuracy

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Observations}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Limitations of Accuracy

Better Alternatives

Metrics that account for class imbalance and error type:

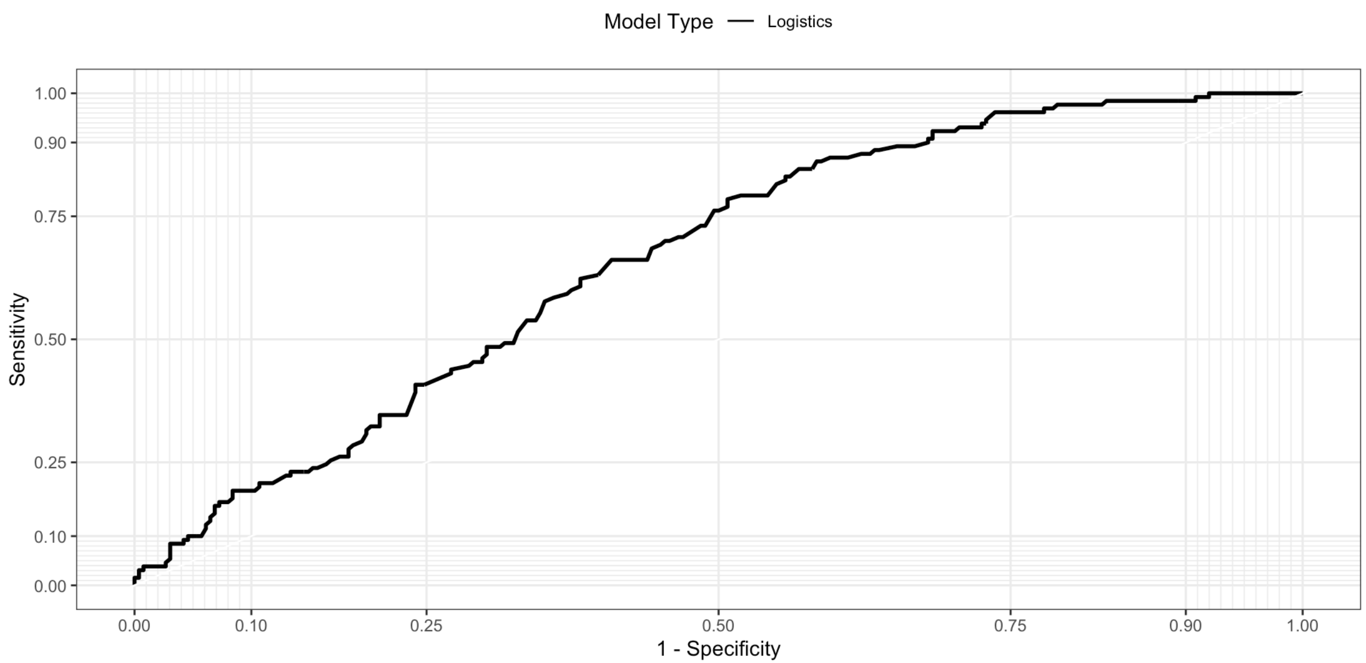


- Sensitivity (Recall) $= \frac{\mathrm{TP}}{\mathrm{TP+FN}} = \frac{\mathrm{TP}}{\mathrm{P}}$
- Specificity $= \frac{\mathrm{TN}}{\mathrm{TN+FP}} = \frac{\mathrm{TN}}{\mathrm{N}}$
- Precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}}$
- $F_1 = \frac{2\ \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$ (Harmonic mean)
- $\mathrm{GM} = \sqrt{\mathrm{Precision} \times \mathrm{Recall}}$ (Geometric mean)

📈 Receiver Operating Characteristic (ROC) Curve

- The **ROC curve** is a graphical tool used to evaluate the performance of a **binary classifier**.

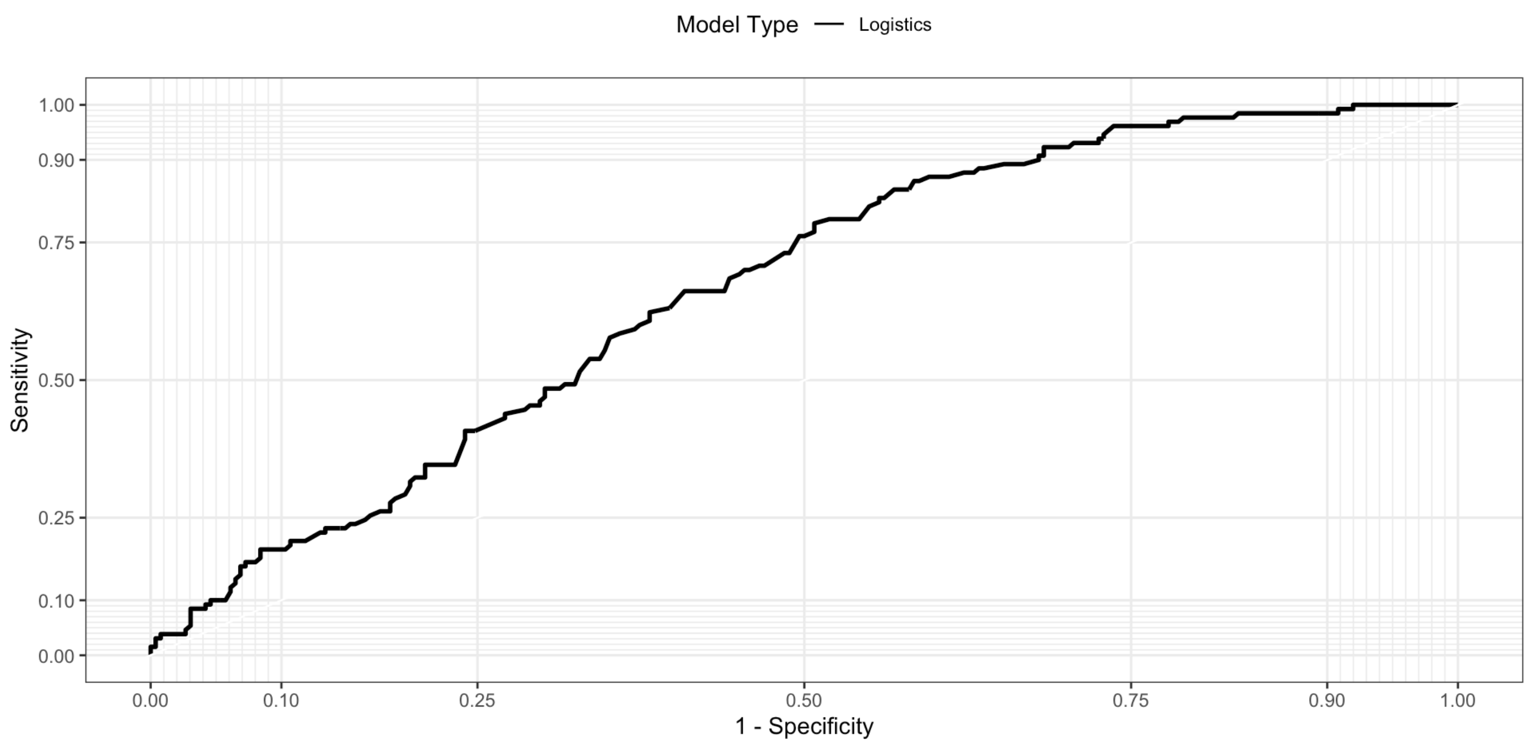
`PimaIndiansDiabetes`



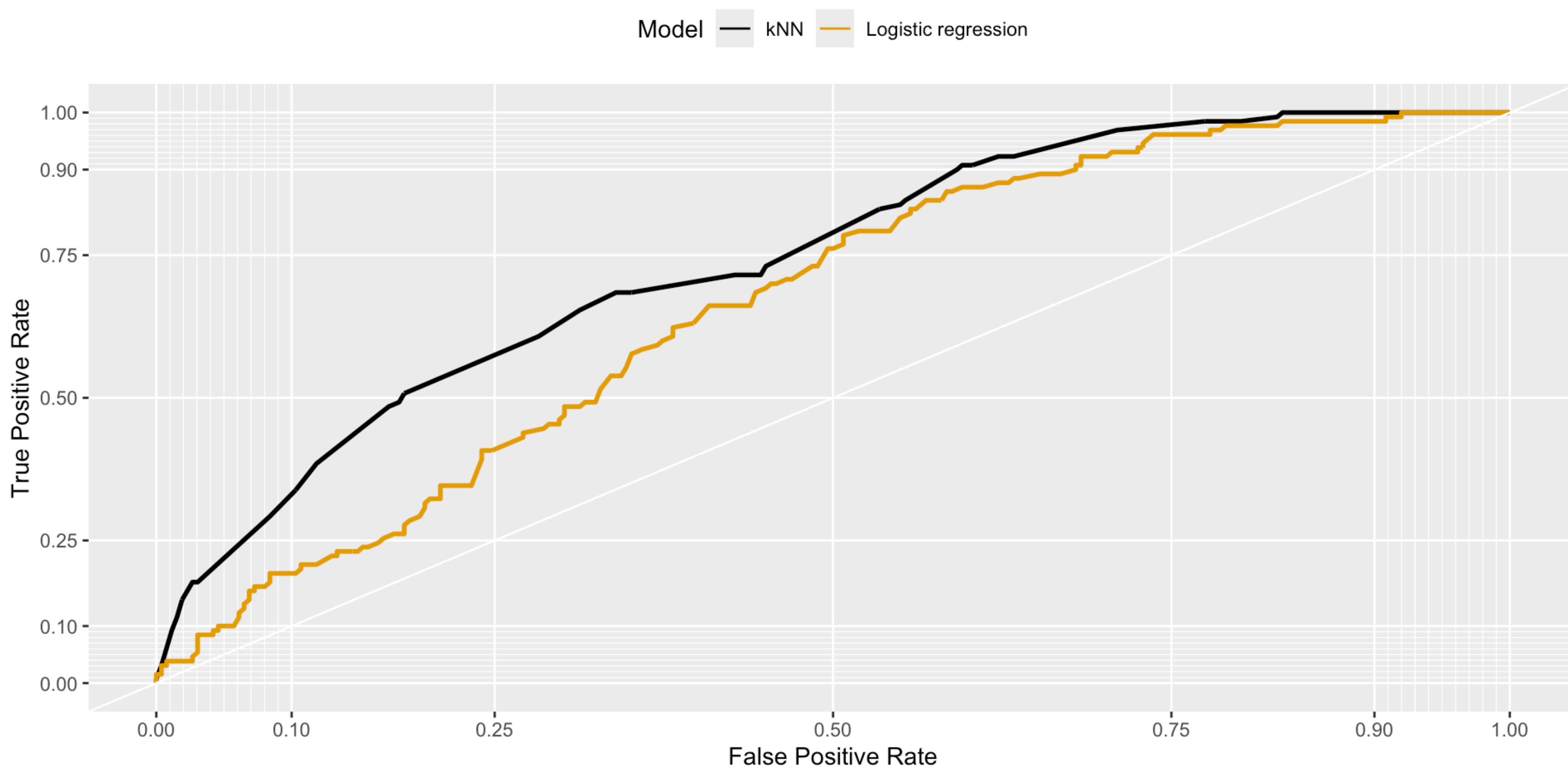
- It plots the **Sensitivity** against the **False Positive Rate (1 - Specificity)** at **various threshold settings**.
- Each point on the ROC curve corresponds to a different **decision threshold**.
- The closer the curve is to the **top-left corner**, the better the model.

Why Use the ROC Curve?

- Helps compare **classification models**.
- Useful when classes are **imbalanced**.
- The **Area Under the Curve (AUC)** summarises the overall performance:
  - AUC = 1: Perfect model
  - AUC = 0.5: No better than random guessing



Comparing ROC curves



Multi-class classfication

- We have discussed performance measures for binary classification
- Performance measured like precision, recall, F_1, and AUC may be generalised to classification problems with more than 2 labels
    - ➠ Useful for your group project
    - ➠ A kaggle blog for reading (code in Python)