Lecture Title and Date

Lecture 25m8a+b: Supervised Data Mining: Preliminaries + Decision Trees - 02/19/2025

Objectives of the Lecture

- By the end of this lecture, students should be able to:
    - Gain exposure to types of machine learning calculations and understand generally the goals of machine learning techniques
    - Understand the main considerations in machine learning:
        - Understand the bias-variance tradeoff, including the concept of overfitting, and be able to explain its impact on model performance
        - Understand the curse of dimensionality and how complex models with many features may lead to a downgrade in performance
        - Understand how these important considerations influence the way we design models
    - Understand how to assess the performance of data mining models effectively.
        - Emphasis on evaluating models' ability to generalize beyond training data to ensure reliability in practical applications.
        - Understand dataset separation techniques (training/testing sets, cross-validation).
        - Understanding model evaluation metrics (True Positives, True Negatives, sensitivities, specificities).
        - Utilization of graphical tools like the Receiver Operating Characteristic (ROC) curve.

Key Concepts and Definitions

- **Bias-Variance Tradeoff**:
    - Bias: occurs when a model is too simple, leading to underfitting
    - Variance: occurs when a model is too complex, causing it to capture noise in the training data rather than general trends, leading to overfitting
    - Bias-Variance Tradeoff describes the balancing of these two error sources. In a model, we aim to minimize both
- **Curse of Dimensionality**:
  The notion that as the number of features/ dimensions increases, our model sees decreased model performance if the number of data points does not scale accordingly. The takeaway is that simpler models with fewer and well-chosen features often see better performance than complex models.
- **Overfitting**: Occurs when a model learns the noise in the training data instead of underlying patterns, leading to poor generalization.
- **Cross-Validation**: A technique for validating model performance by dividing data multiple ways (e.g., leave-one-out) and averaging the results.
- **Accuracy:** Proportion of true results among total cases examined.

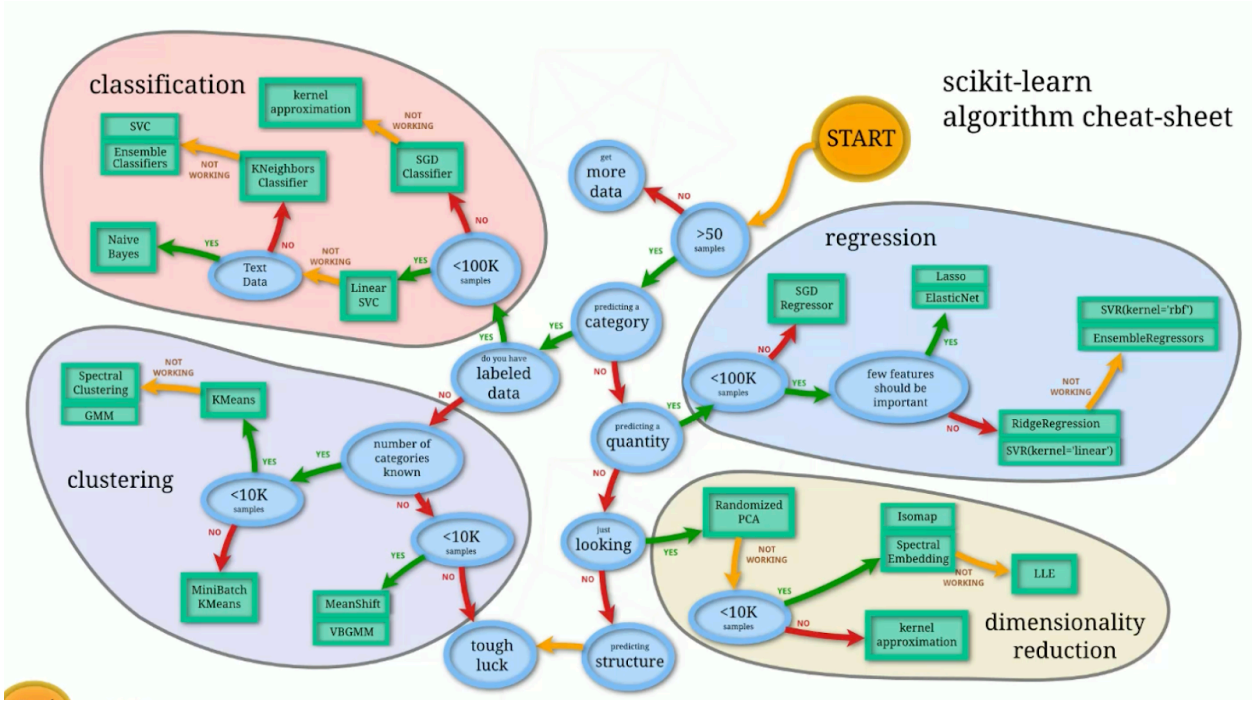
- **Sensitivity:** Proportion of actual positives correctly identified.
- **Specificity:** Proportion of actual negatives correctly identified.
- **Error Rate**: Proportion of incorrect predictions relative to the total predictions.
- **ROC Curve**: A graphical representation to evaluate the performance of a binary classifier. One axis shows sensitivity, and the other shows the false positive rate (1 - specificity).
- **Area Under the Curve (AUC)**: Measures the classifier's ability across different threshold settings; values closer to 1 indicate better performance while 0.5 signals random predictions.

Main Content/Topics

General Overview of Machine Learning (ML):

- **1. What is machine learning?**

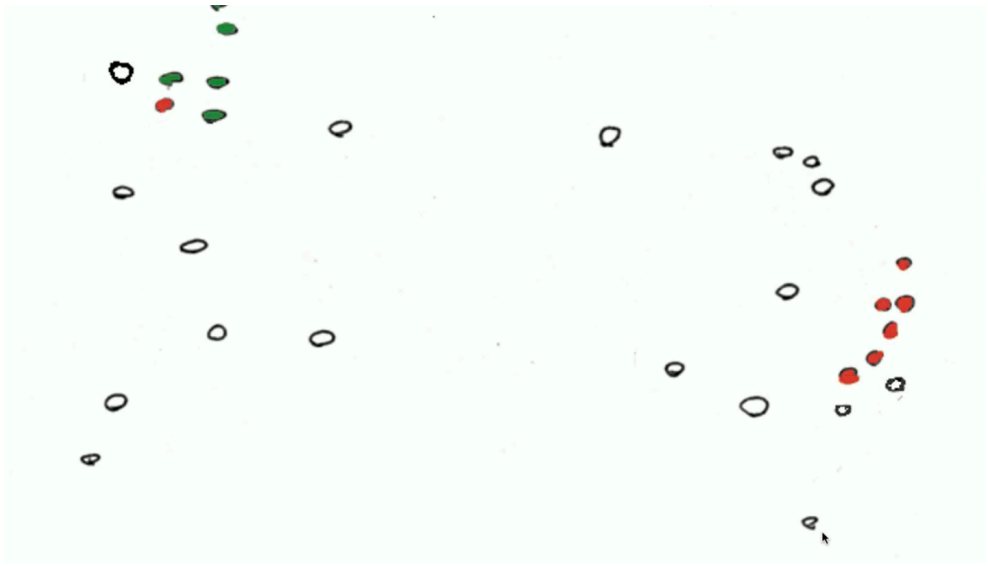
The world of ML is complicated and not completely organized. There are groups of calculations we can do with ML techniques, such as clustering, classification, regression, dimensionality reduction. Clustering and dimension reduction tend to fall into the unsupervised category, while classification and regression tend to fall into the supervised category.



The calculation used also depends on the amount of data, type of data, and interpretability. There are a myriad of types of calculations we can do, but a few broad distinctions in these calculations we can consider, particularly in supervised learning, are:

a) Regression vs Classification – Regression models use quantitative labels, while classification models use categorical labels.
b) Regularized vs Unregularized – Regularization techniques prevent overfitting through penalization of complex models, while unregularized models have no penalty and risk of overfitting.
c) Parametric vs Nonparametric – In parametric models like linear regression, an explicit parametric model is assumed, otherwise the model is non-parametric.
d) Ensemble vs Non Ensemble – Ensemble models combine multiple models to improve performance while non-ensemble models rely on a single model.

When using machine learning techniques, we usually use tables/ matrices to structure data, where columns correspond to different coordinates (or features) and rows correspond to different instances. An example is representing genomic data with “sites along the genome” as our columns and “factors and chromatin modifications (different tissues)” as our rows. One way to then visualize our data matrix is by thinking of each row as a point in an abstract, high-dimensional space. Revisiting the types of calculations we mentioned earlier, many of them have to do with drawing boundaries, clustering, labelling, classifying, and extrapolating points in this high-dimensional data space.



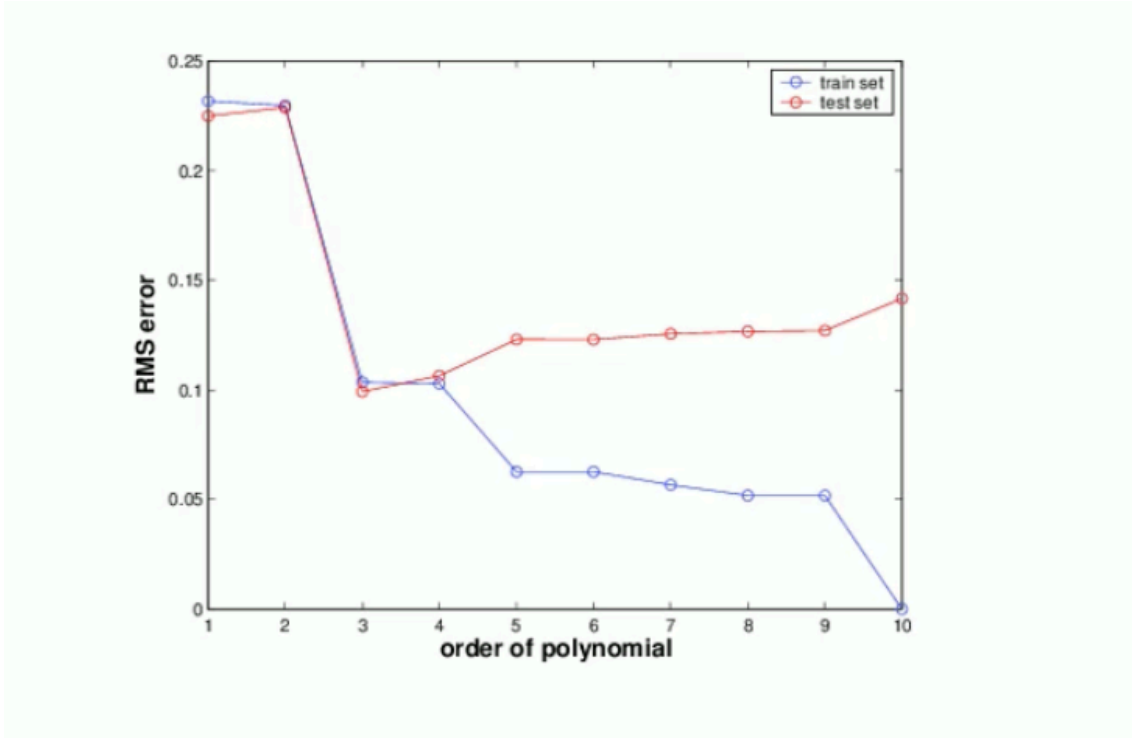
- **2. Considerations in ML**

Model dimensionality and Overfitting, CoD

There are many considerations in ML stemming from how we assess the performance of our ML models. Generally to evaluate performance, we divide our datasets into a **training and**

testing set. We parametrize and build a model on a training set, and test on a testing set (discussed further below). This process becomes complicated when data is limited, and people also must consider if this split between training and testing is representative or biased. Cross validation helps mitigate this issue by performing many potential splits and averaging the results. In addition to training the model, when building models there are different levels of parameters we need to optimize. Although we train a model on a particular training dataset, there are also hyperparameters that we want to fine tune as well. To do this, sometimes we further split the dataset into a **validation set**, which can be used to tune hyperparameters before a final evaluation on the testing set.

Another key consideration when designing ML models is how to avoid overfitting. **Overfitting** occurs when we introduce so many parameters into the model that it memorizes the data; this becomes an issue when we use it to predict new points, as the model has too much variance and is unable to generalize well when introduced to new data. On the other hand, there is too much **bias** in our model when there are too few parameters, causing the model to oversimplify the data relationships. Therefore, we need to balance the **bias-variance tradeoff** – i.e. choosing a model that is not too simple but also not too sensitive. A related concept is Occam's Razor, which encourages us to "accept the simplest explanation that fits the data."



The graph above depicts the bias-variance tradeoff. Observe that initially when the order of polynomials is low (i.e. too much bias), the error is higher. As the order of polynomials increases (i.e. variance increases), we begin to overfit. Error in the training set continues to decrease as the model overfits and memorizes the data, but error in the testing set begins to rise since the generalization capabilities of an overfit model is weak.

An important note is that in addition to overfitting, increasing the number of parameters/ features in the model may lead to a reduction in performance if the number of datapoints is not increased. This is known as the **curse of dimensionality,** which warns that oftentimes simpler models perform better than models with too many features.

- ROC (receiver operating characteristic) plot: graph some notion of error rate (i.e. number of false positives/ negatives) versus some notion of coverage (how many of the known positives that we cover). We can then threshold this score and compare to positive and negative gold standards
  - Example: Breast cancer screening; notion of people that have breast cancer, notion of people that don't have breast cancer
  - Similarly: Screening for terrorists at the airport: small represents terrorist, large represents number of normal people
  - Threshold and make prediction; black dots will be called "positive"; in the two above examples, sensitivity stays the same BUT number of FP dramatically increases; there are more false positives than true positives; this is because majority people in the population do not have cancer/ terrorist
  - But sensitivity and specificity don't change → what changes is positive predictive value…tricky!
  - TLDR: must consider how balanced the dataset is

Data Mining Performance Evaluation

1. **Overview of Evaluation**

Evaluating the performance of data mining approaches is crucial for determining their effectiveness and practical application. By measuring how well models can generalize to **new, unseen data**, practitioners can ensure that the developed models are reliable in real-world scenarios and not merely capturing **noise or overfitting** to a particular dataset.

A common methodology involves splitting datasets into **training and testing sets**, and sometimes also including a **validation set**. The **training set** is used to fit or "teach" the model, while the testing set is reserved strictly for evaluating the final performance of the trained model, providing an unbiased assessment of how it might perform in practice. When included, the validation set allows for further **fine-tuning** of parameters or hyperparameters without compromising the integrity of the testing data.

In addition to these splits, **cross-validation** is widely adopted to gain a more robust and reliable estimation of a model's performance. Under this method, data is divided into multiple subsets or "folds." The model is trained on some of these folds and tested on the remaining one, and this process is rotated to ensure every fold serves as a test set once. By aggregating the results across folds, cross-validation provides a more comprehensive view of how well the model might generalize.

A specialized version of this approach is the **Leave-One-Out Method**, where a single data point is isolated as the test set, and the remaining data points form the training set. The process repeats until every data point in the dataset has been used as the test instance once. While computationally more intensive, leave-one-out cross-validation can sometimes yield more precise insights into the model's performance, especially for smaller datasets.

2. Model Evaluation Metrics

a) Definitions

- **True Positives (TP):** The number of positive cases correctly identified by the model.
- **True Negatives (TN)**: The number of negative cases correctly identified.
- **False Positives (FP)**: The number of negative cases incorrectly classified as positive.
- **False Negatives (FN):** The number of positive cases incorrectly classified as negative.

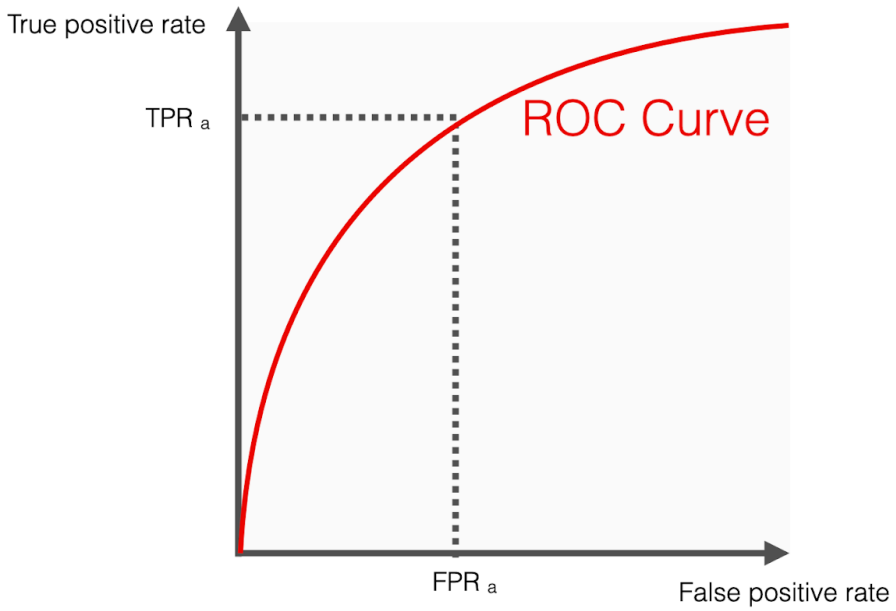
b) Key Metrics

Classification problems revolve around determining whether a given instance belongs to one category or another, and evaluating their effectiveness relies on a variety of metrics. One fundamental measure is Accuracy, which represents the proportion of true results—both true positives and true negatives—out of all predictions made. It provides a quick gauge of overall performance but can sometimes mask issues with unbalanced data.

Metrics such as **Sensitivity** and **Specificity** focus on different types of errors. Sensitivity = (TP / (TP + FN)) measures how effectively the model identifies actual positive instances, while Specificity = (TN / (TN + FP)) gauges how well the model identifies actual negative instances. Another related metric, the **True Positive Rate** = (TP / (TP + FN)), indicates how many of the genuinely positive are predicted positives. Finally, **Error Rate** captures the proportion of incorrect predictions relative to the total predictions, serving as a simple inverse measure to accuracy.

By contrast, regression problems aim to predict continuous values rather than class labels. Here, performance is often evaluated using the **sum of squares error**, which quantifies how far the predictions deviate from the actual values by summing the squared differences. The **root mean square error** goes a step further by taking the square root of the average of the squared differences, thus placing the error in the same units as the predicted variable and often making it more intuitive to interpret.

ROC plot:



A commonly used graphical technique for assessing a binary classifier's performance is the **Receiver Operating Characteristic** (ROC) curve, which plots the **true positive rate** (sensitivity) on the y-axis against the **false positive rate** (1 - specificity) on the x-axis for various threshold settings. By examining this curve, one can evaluate how well a classifier separates positive instances from negative ones as the decision threshold changes. The Area Under the Curve (AUC) then serves as a numerical summary of the classifier's **discriminative power**, reflecting the likelihood that the classifier will correctly rank a random positive instance higher than a random negative instance. An AUC of 1.0 signifies a perfect classifier, while an AUC of 0.5 indicates performance equivalent to random guessing.

c) Unbalanced Dataset

In the context of **unbalanced datasets**, where there is a significant disparity in the size of positive and negative classes, interpreting model performance can become complex. A high specificity might coincide with a low positive predictive value if the positive cases are rare compared to negatives, which is notably evident in cases such as breast cancer detection. In situations where datasets are heavily imbalanced, a model that seems to perform poorly might actually be quite effective, provided it is analyzed through the right lens.

For instance, in breast cancer screening, the population includes both individuals with breast cancer and those without. In this scenario, while the sensitivity (the ability to correctly identify actual cases of cancer) remains stable, the number of false positives (individuals incorrectly identified as having cancer) may substantially increase. This is largely because the majority of individuals in the population do not have breast cancer, thus skewing the results. As a result, although the model effectively identifies many true positive cases, it also flags a significant number of false positives, leading to challenges in interpreting the predictive value accurately.

Similarly, consider the process of screening for potential terrorists at airports. Here again, the majority of the population consists of non-threatening individuals, while only a small fraction may represent actual threats. The concept of a threshold is applied, whereby certain indicators lead to predictions labeled as ‘positive’ (black dots). Although the sensitivity remains unchanged, the increase in false positives is notable. Because most people are normal, the prevalence of false positives can overshadow the true positives identified by the screening process.

In instances where the true number of positive and negative cases remain uncertain, using approximate metrics like **positive predictive value** becomes crucial for evaluating model performance. These metrics provide a means to estimate the model's predictive abilities in scenarios with ambiguous ground truth.

Discussion/Comments

- Discussion Point: In real-world applications, how can we better mitigate the issue of overfitting in our models?
  Possible Answer: One effective approach is to use regularization techniques, such as Lasso or Ridge regression, which can help constrain model complexity. Additionally, employing cross-validation helps ensure the model’s generalization by evaluating it on multiple subsets of data.

- Question: How can we decide the optimal threshold for our model when analyzing an ROC curve?
  Possible Answer: The optimal threshold can be determined based on the specific needs of the application, such as whether false positives or false negatives carry higher costs. Additionally, one can use the point on the ROC curve that is nearest to the top-left corner, which represents the best balance between sensitivity and specificity.

- Question: In the context of unbalanced datasets, what strategies can be employed to improve the predictive value?
  Possible Answer: Some strategies include balancing the dataset through resampling techniques like oversampling the minority class or undersampling the majority class, and using synthetic data generation methods like SMOTE. Another approach is to adjust the classification threshold or use cost-sensitive learning methods that take class imbalance into account.

- Comment: The examples provided about breast cancer screening and airport security really highlight the practical implications of these metrics! What are some real-world consequences of high false positive rates in these scenarios?
  Possible Answer: High false positive rates can lead to unnecessary anxiety and medical procedures in breast cancer screenings, while in airport security, it could result in

unwarranted detentions and loss of public trust. Thus, it is crucial to strike a balance between sensitivity and specificity to minimize adverse effects.

- Question: How can organizations implement the knowledge gained from this lecture to improve data mining efforts?
  Possible Answer: Organizations can create a systematic framework for model evaluation that incorporates the discussed metrics, ensuring ongoing training and validation of models with fresh data. Additionally, fostering interdisciplinary collaboration can enhance understanding of context when interpreting results, ultimately leading to better decision-making.

References ISL/ESL

ISL:

- Relative Chapters:
  - Chapter 4 (4.4 mentioned ROC)
  - Chapter 5 Resampling methods (Cross validation)
- Chapter 4.4:
  The ROC curve is a popular graphic for simultaneously displaying the two types of errors for all possible thresholds. The name “ROC” is historic, ROC curve and comes from communications theory. It is an acronym for receiver operating characteristics.

		True class		
		− or Null	+ or Non-null	Total
Predicted class	− or Null	True Neg. (TN)	False Neg. (FN)	N*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P*
	Total	N	P	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

TABLE 4.7. *Important measures for classification and diagnostic testing, derived from quantities in Table 4.6.*

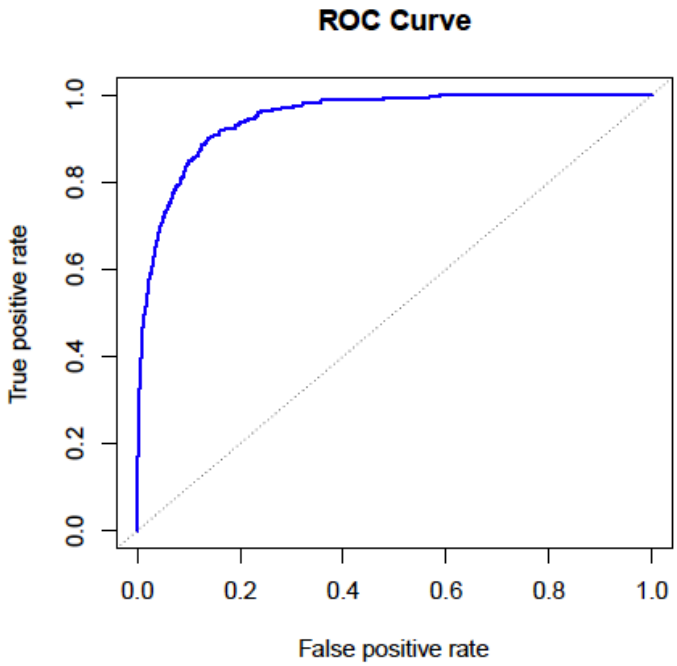


FIGURE 4.8. *A ROC curve for the LDA classifier on the* `Default` *data. It traces out two types of error as we vary the threshold value for the posterior probability of default. The actual thresholds are not shown. The true positive rate is the sensitivity: the fraction of defaulters that are correctly identified, using a given threshold value. The false positive rate is 1-specificity: the fraction of non-defaulters that we classify incorrectly as defaulters, using that same threshold value. The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The dotted line represents the "no information" classifier; this is what we would expect if student status and credit card balance are not associated with probability of default.*

An ideal ROC curve will hug the top left corner, so the larger the area under the ROC curve (AUC) the better the classifier.

- Chapter 5.1 Cross Validation:
- Mentioned several approaches to perform cross validation:
  - Leave-One-Out Cross-Validation (LOOCV): only pick out one instance as test data when splitting training and testing dataset.

    The LOOCV estimate for the test MSE is the average of these n test error estimates:

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i. \qquad (5.1)$$

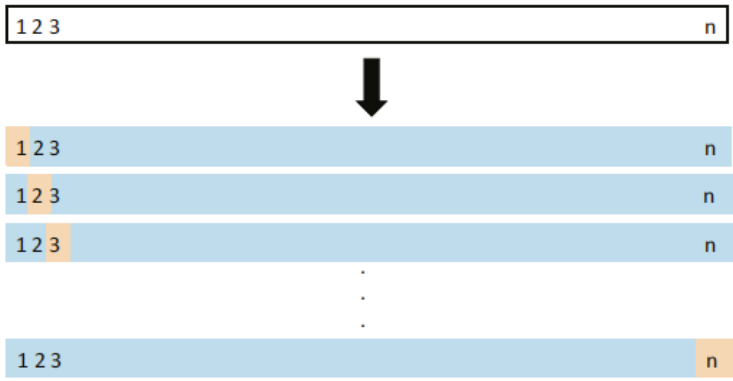


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSEs. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

Can be computationally expensive.

- K-fold cross validation

An alternative to LOOCV is *k-fold CV*. This approach involves randomly dividing the set of observations into k groups, or *folds*, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining $k-1$ folds. The mean squared error, MSE_1, is then computed on the observations in the held-out fold. This procedure is repeated k times; each time, a different group of observations is treated as a validation set. This process results in k estimates of the test error, $\text{MSE}_1, \text{MSE}_2, \ldots, \text{MSE}_k$. The k-fold CV estimate is computed by averaging these values,

k-fold CV

$$\text{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \text{MSE}_i. \tag{5.3}$$

Figure 5.5 illustrates the k-fold CV approach.

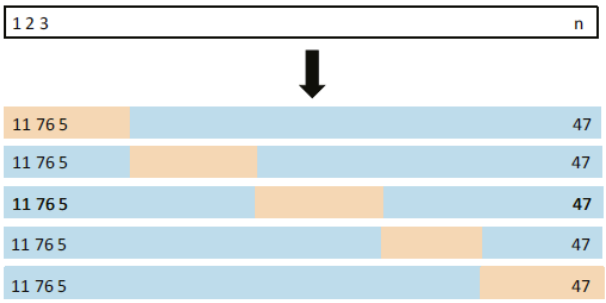


FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

- Bias-variance trade off in cross validation
  Bias: LOOCV performs better than k-fold CV
  Variance: k-fold CV performs better than LOOCV

- ESL:
  - Chapter 7: Model Assessment and Selection (7.3: bias-variance tradeoff)

    Figure 7.1 illustrates the important issue in assessing the ability of a learning method to generalize. Consider first the case of a quantitative or interval scale response. We have a target variable Y, a vector of inputs X, and a prediction model $\hat{f}(X)$ that has been estimated from a training set $\mathcal{T}$. The loss function for measuring errors between Y and $\hat{f}(X)$ is denoted by $L(Y, \hat{f}(X))$. Typical choices are

$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{squared error} \\ |Y - \hat{f}(X)| & \text{absolute error.} \end{cases} \tag{7.1}$$

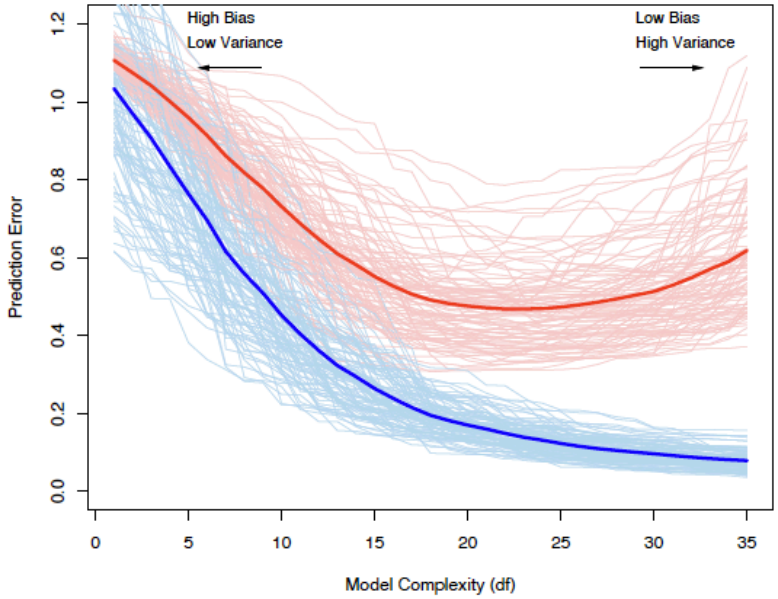
  - 

    FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for* 100 *training sets of size* 50 *each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\text{E}[\overline{\text{err}}]$.