Validation and Interpretation

Model Evaluation: Performance Metrics
Recap: Cross – Validation

Key Idea:

*Building on repeated holdout, enforce splits that place each data point in the test data only once.*
Recap: Leave-one-Out Cross-Validation \( \text{LOOCV} \)

- The simplest way to ensure test sets do not overlap is to evaluate each data point as it’s own test set.
- Number of iterations equals the number of instances.
  - Classifier trained \( n \) times, where \( n \) is number of training instances.
- Typically used when number of instances is small, as it provides the maximal amount of training data.
Recap: Leave More Than One Out

- To provide more robust estimation of error we would like to increase the number of points in each test dataset
  - Whereas: Leave one out: $N$ training sessions on $N - 1$ points each

- We can expand into $k$ (usually 5 or 10) equal disjoint subsets or *folds*.
  - e.g., 10 folds, 1 fold (10% of data) used for testing, 9 folds (90% of data) used for training
  - Testing fold rotates each iteration with $\frac{N}{k}$ training sessions on $N - K$ points each

- Average results over all 10-folds for error estimate.
  - To further reduce variance in estimates, may repeat.

5-fold Cross-Validation Example

Test Data ← Training Data →

- Iteration 1
- Iteration 2
- Iteration 3
- Iteration 4
- Iteration 5

Total Dataset ←→
So You’ve Run Cross Validation:

How do we quantify model error?

\[ E_{cv} = \frac{1}{3} (e_1 + e_2 + e_3) \]

Remember at each iteration (fold) we must compute an error for the test set, which is averaged to provide an overall estimate of model performance. However, we must now define, what is e?
Defining Error

The measures of error are drastically different between regression and classification models

- **Regression models:**
  - Provide a means to predict a continuous outcome.

- **Classification models:**
  - Provide a means to predict a binary outcome.
Defining Error

The measures of error are drastically different between regression and classification models

- **Regression models:**
  - Provide a means to predict a continuous outcome.
    - *Linear Regression*
    - *Linear Mixed Effects*
    - *Cox Regression*

- **Classification models:**
  - Provide a means to predict a binary outcome.
    - *Logistic Regression*
    - *Logistic Mixed effects*
Regression Models Evaluation

• Regression models are defined as those which predict a continuous target value \((y)\)
  – I.e. number of days admitted to the hospital

• Thus measurement of error can be done using a notion of **distance**

• This distance is defined as that between the true value, and the predicted value from our model
  – In much the same way as we compute residuals on the training data to assess the model assumptions
  – We know the true value of the “unseen” data, as we artificially hold it out in our (cross) validation frameworks
Regression Model – Metrics

Although an immense set of distance metrics could quantify the regression error, there exist 2 common measures used in practice.

**Root Mean Square Error (RMSE)**

\[
RMSE = \sqrt{\frac{\sum_{j=1}^{n} (y_j - \hat{y}_j)^2}{n}}
\]

- RMSE gives a relatively high weight to large errors (outliers may be problematic)
- In some situations RMSE is known to increase with number of observations.
  - Comparisons between different sample sizes difficult

**Mean Absolute Error (MAE)**

\[
MAE = \frac{\sum_{j=1}^{n} |y_j - \hat{y}_j|}{n}
\]

- More robust (less sensitive) to outliers
- Differences easily interpretable
  - Difference from 0-5, is half 0-10
    - This is 4x smaller with regard to RMSE

- Both metrics can range from 0 to \(\infty\) and are indifferent to the direction of errors
- Both measures provide the result in the *same* units as \(y\).
  - This is extremely useful for interpretation
    - i.e. average 0.3 days off for a length of stay prediction
Classification Model Evaluation

- Classification models are defined as those which are used to predict a discrete target value
  - In our case, we have studied binary classification:
    - Having a tumor or not
    - Dying in the hospital

- In this way, distance becomes less meaningful
  - Can be correct (predict 0, truly 0 / predict 1, truly 1)
  - Or incorrect (predict 1, truly 0 / predict 0, truly 1)

- As a result we look to an entirely different evaluation framework to gain deeper insight into model performance.
  - One focused on quantifying the 4 possible prediction combinations listed above

- Note this same paradigm is seen all across healthcare, including:
  - Diagnostic test accuracy (does the patient truly have a disease vs. the test result)
Classification Metrics: The Confusion Matrix

- In setting up the evaluation of a classification model, data are often organized in what is known as a confusion matrix.
  - Pretty much the same as the contingency tables we have seen multiple times this semester.
Classification Metrics: 
*The Confusion Matrix*

The quadrants are used to define each of the four possible states:

- **TP**: True Positive
  - Positive instance correctly classified

- **FP**: False Positive
  - Negative instance incorrectly classified

- **TN**: True Negative
  - Negative instance correctly classified

- **FN**: False Negative
  - Positive instance incorrectly classified
Classification Metrics: The Confusion Matrix

Note:
It is simple to extend the confusion matrix to more than 2 categories

However for this class, we are focusing on binary classification problem, thus the 2x2 is sufficient

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>37</td>
<td>3</td>
<td>7</td>
<td>Σa = 47</td>
</tr>
<tr>
<td>b</td>
<td>9</td>
<td>25</td>
<td>5</td>
<td>Σb = 39</td>
</tr>
<tr>
<td>c</td>
<td>11</td>
<td>2</td>
<td>43</td>
<td>Σc = 56</td>
</tr>
<tr>
<td>Total</td>
<td>57</td>
<td>30</td>
<td>55</td>
<td>N = 142</td>
</tr>
</tbody>
</table>
Accuracy

• The most common and simplest to compute classification metric
  – The total number correctly predicted divided by the number of predictions made

\[
\frac{TP + TN}{TP + FP + TN + FN}
\]
Limitations of Accuracy

Imagine this is our dataset:
- Blue items represent patients who die from a rare disease
- Yellow items are patients who live

As we are interested in health analytics we build a model to predict patients who will die from the disease
Limitations of Accuracy

Imagine this is our dataset:
- Blue items represent patients who die from a rare disease
- Yellow items are patients who live

As we are interested in health analytics we build a model to predict patients who will die from the disease.

We decide to use accuracy as our metric, and after cross validation select a model that gives us 90% accuracy.

However when we check the model predictions we see the following:

What happened?
Limitations of Accuracy

Imagine this is our dataset:
- Blue items represent patients who die from a rare disease
- Yellow items are patients who live

As we are interested in health analytics we build a model to predict patients who will die from the disease.

We decide to use accuracy as our metric, and after cross validation select a model that gives us 90% accuracy.
- However when we check the model predictions we see the following:

Although the model is accurate, it is useless as it never predicts a patient to have the disease in question.

Thus we can turn to a set of metrics designed to aid in the interpretation of imbalanced class.

It is also not uncommon to evaluate data with respect to each class.
Precision and Recall

Recall

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

- Recall can be thought of as a model’s ability to find all the data points of interest in a dataset.
  - In the rare disease example, true positives are correctly identified disease cases, and false negatives would be individuals the model labels as not diseased that actually were.
- As recall ↑, false negatives (FN) ↓

Precision & Regression

Statistical Methods

Data Preprocessing

Data Understanding

Preliminaries

Validation & Interpretation

Precision

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

While recall expresses the ability to find all relevant instances, precision expresses the proportion of the data our model says was relevant actually were relevant.

\[
\begin{array}{c|c|c|c|c}
\text{Actual Class} & \text{Positive} & \text{Negative} \\
\hline
\text{Positive} & TP & FP \\
\hline
\text{Negative} & FN & TN \\
\end{array}
\]
Recall

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

- Recall can be thought of as a model’s ability to find all the data points of interest in a dataset.
  - In the rare disease example, true positives are correctly identified disease cases, and false negatives would be individuals the model labels as not diseased that actually were.
- As recall ↑, false negatives (FN) ↓

Precision

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

- While recall expresses the ability to find all relevant instances, precision expresses the proportion of the data our model says was relevant actually were relevant.
- As precision ↑, false positives (FP) ↓
Combining Precision and Recall $F_1$

- In some situations, we might know that we want to maximize either recall or precision at the expense of the other metric.
  - In preliminary disease screening of patients, we would probably want a recall near 1.0
    - Find all patients who actually have the disease
    - We can accept a low precision if the cost of the follow-up examination is not significant.

- However, in cases where we want to find an optimal blend of precision and recall we can combine the two metrics using what is called the F1 score.
Combining Precision and Recall

\[ F_1 = \frac{2rp}{r + p} = \frac{2 \times TP}{2 \times TP + FP + FN} \]

- As \( F_1 \) measure ↑, false positives (FP) and false negatives (FN) ↓

- A classifier with a precision of 1.0 and a recall of 0.0 has a simple average of 0.5 but an F1 score of 0

- Essentially weighted average of precision and recall.
  - Actually harmonic mean
Quick Aside: Diagnostic Tests
Sensitivity and Specificity

**Sensitivity**
\[
\text{Sensitivity} = \frac{TP}{TP + FN}
\]
- Conditional probability of predicting/testing positive given the truth is positive
  - Ability to identify true positives

**Specificity**
\[
\text{Specificity} = \frac{TN}{TN + FP}
\]
- Conditional probability of predicting/testing negative given the truth is negative
  - Ability to identify true negatives
Positive / Negative Predicted Value

**Positive Predicted Value**

\[ \frac{TP}{TP + FP} \]

- Conditional probability the truth is positive given predicted/tested positive

**Negative Predicted Value**

\[ \frac{TN}{TN + FN} \]

- Conditional probability the truth is negative given predicted/tested negative
Understanding the Differences

• *Sensitivity and specificity* are *characteristics of the test*.  
  – The population does not affect the results.

• *Positive and negative predictive values* are *influenced by the prevalence* of disease in the population that is being tested.

• As prevalence increases, PPV increases, NPV decreases  
  – If we test in a high prevalence setting, it is more likely that persons who test positive truly have disease than if the test is performed in a population with low prevalence.
Practical Uses

• **Positive/Negative predictive values:**
  – Imagine that you have just received the results of a screening test. If the test was positive, the patient will want to know the probability that they really have the disease, i.e., how worried should they be?
  – Or if they are negative how reassured should the patient be? What is the probability that they are disease free?

• **Sensitivity / Specificity**
  – Sensitivity: Ability of the test in identifying disease in people who truly have the disease
  – Specificity: Ability of the test in correctly classifying truly non-diseased people.
Classifier Outcomes

• Remember, many classification models actually predict a probability of a class

• In order to determine the final outcome we specify a specific threshold at which to draw our decision boundary
  – For the figure to the right, all points over 0.5 are classified as 1, otherwise 0

• However this presents a potential source of error (or an opportunity) to better improve our models
Thinking Back: Types of Error

- **Type 1 error: false positive**
  - Incorrect prediction of negative instance as positive
  - Error of commission

- **Type 2 error: false negative**
  - Incorrect prediction of positive instance as negative
  - Error of omission
Thinking Back: Types of Error

- As we shift the red line, our values in each of the 4 confusion matrix quadrants can change dramatically, allowing us to identify an optimal split...

**But where?**
Thinking Back: Types of Error

- As we shift the red line our values in each of the 4 confusion matrix quadrants can change dramatically, allowing us to identify an optimal split.
  
  Why not try them all!
The Receiver Operator Curve (ROC)

- The notion of testing a classifier's performance across a range of thresholds is captured by what is known as a Receiver Operator Curve (ROC)
  - A ROC curve is a graphical plot that illustrates performance of a binary classifier as its threshold is varied.
Constructing a Curve

- We begin by collecting the true class of all data in the test set.

- Each instance is labeled with the class and the score (probability) provided by the classifier.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Class</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>positive</td>
<td>.4</td>
</tr>
<tr>
<td>2</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>3</td>
<td>positive</td>
<td>.6</td>
</tr>
<tr>
<td>4</td>
<td>positive</td>
<td>.4</td>
</tr>
<tr>
<td>5</td>
<td>negative</td>
<td>.5</td>
</tr>
<tr>
<td>6</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>7</td>
<td>positive</td>
<td>.3</td>
</tr>
<tr>
<td>8</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>9</td>
<td>positive</td>
<td>.5</td>
</tr>
<tr>
<td>10</td>
<td>positive</td>
<td>.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>Class</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>positive</td>
<td>.6</td>
</tr>
<tr>
<td>12</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>13</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>14</td>
<td>positive</td>
<td>.5</td>
</tr>
<tr>
<td>15</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>16</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>17</td>
<td>positive</td>
<td>.3</td>
</tr>
<tr>
<td>18</td>
<td>positive</td>
<td>.2</td>
</tr>
<tr>
<td>19</td>
<td>negative</td>
<td>.1</td>
</tr>
<tr>
<td>20</td>
<td>negative</td>
<td>.1</td>
</tr>
</tbody>
</table>

Example adopted from: https://m.youtube.com/watch?v=sWAJsiVh1Gg
## Constructing a Curve

<table>
<thead>
<tr>
<th>Instance</th>
<th>Class</th>
<th>Score</th>
<th>Instance</th>
<th>Class</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>positive</td>
<td>.6</td>
<td>8</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>10</td>
<td>positive</td>
<td>.6</td>
<td>13</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>11</td>
<td>positive</td>
<td>.6</td>
<td>16</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>5</td>
<td>negative</td>
<td>.5</td>
<td>17</td>
<td>positive</td>
<td>.3</td>
</tr>
<tr>
<td>9</td>
<td>positive</td>
<td>.5</td>
<td>6</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>14</td>
<td>positive</td>
<td>.5</td>
<td>12</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>1</td>
<td>positive</td>
<td>.4</td>
<td>15</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>4</td>
<td>positive</td>
<td>.4</td>
<td>18</td>
<td>positive</td>
<td>.2</td>
</tr>
<tr>
<td>2</td>
<td>negative</td>
<td>.3</td>
<td>19</td>
<td>negative</td>
<td>.1</td>
</tr>
<tr>
<td>7</td>
<td>positive</td>
<td>.3</td>
<td>20</td>
<td>negative</td>
<td>.1</td>
</tr>
</tbody>
</table>

- The instances are then sorted from high to low probability.
Generating ROC Curves

Threshold of .6
TP = 3/10, FP = 0/10
Generating ROC Curves

Threshold of .6
TP = 3/10, FP = 0/10
### Generating ROC Curves

<table>
<thead>
<tr>
<th>Instance</th>
<th>Class</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>positive</td>
<td>0.6</td>
</tr>
<tr>
<td>10</td>
<td>positive</td>
<td>0.6</td>
</tr>
<tr>
<td>11</td>
<td>positive</td>
<td>0.6</td>
</tr>
<tr>
<td>5</td>
<td>negative</td>
<td>0.5</td>
</tr>
<tr>
<td>9</td>
<td>positive</td>
<td>0.5</td>
</tr>
<tr>
<td>14</td>
<td>positive</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>positive</td>
<td>0.4</td>
</tr>
<tr>
<td>4</td>
<td>positive</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>negative</td>
<td>0.3</td>
</tr>
<tr>
<td>7</td>
<td>positive</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>Class</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>negative</td>
<td>0.3</td>
</tr>
<tr>
<td>13</td>
<td>negative</td>
<td>0.3</td>
</tr>
<tr>
<td>16</td>
<td>negative</td>
<td>0.3</td>
</tr>
<tr>
<td>17</td>
<td>positive</td>
<td>0.3</td>
</tr>
<tr>
<td>6</td>
<td>negative</td>
<td>0.2</td>
</tr>
<tr>
<td>12</td>
<td>negative</td>
<td>0.2</td>
</tr>
<tr>
<td>15</td>
<td>negative</td>
<td>0.2</td>
</tr>
<tr>
<td>18</td>
<td>positive</td>
<td>0.2</td>
</tr>
<tr>
<td>19</td>
<td>negative</td>
<td>0.1</td>
</tr>
<tr>
<td>20</td>
<td>negative</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Threshold of .5
TP = 5/10, FP = 1/10
Generating ROC Curves

<table>
<thead>
<tr>
<th>Instance</th>
<th>Class</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>positive</td>
<td>.6</td>
</tr>
<tr>
<td>10</td>
<td>positive</td>
<td>.6</td>
</tr>
<tr>
<td>11</td>
<td>positive</td>
<td>.6</td>
</tr>
<tr>
<td>5</td>
<td>negative</td>
<td>.5</td>
</tr>
<tr>
<td>9</td>
<td>positive</td>
<td>.5</td>
</tr>
<tr>
<td>14</td>
<td>positive</td>
<td>.5</td>
</tr>
<tr>
<td>1</td>
<td>positive</td>
<td>.4</td>
</tr>
<tr>
<td>4</td>
<td>positive</td>
<td>.4</td>
</tr>
<tr>
<td>2</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>7</td>
<td>positive</td>
<td>.3</td>
</tr>
<tr>
<td>8</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>13</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>16</td>
<td>negative</td>
<td>.3</td>
</tr>
<tr>
<td>17</td>
<td>positive</td>
<td>.3</td>
</tr>
<tr>
<td>6</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>12</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>15</td>
<td>negative</td>
<td>.2</td>
</tr>
<tr>
<td>18</td>
<td>positive</td>
<td>.2</td>
</tr>
<tr>
<td>19</td>
<td>negative</td>
<td>.1</td>
</tr>
<tr>
<td>20</td>
<td>negative</td>
<td>.1</td>
</tr>
</tbody>
</table>

Complete Sweep
Comparing Classifiers in ROC Space

Case 1: $FPR(A) > FPR(B)$

Case 1: $FPR(A) = FPR(B)$
Area Under the ROC Curve (AUROC)

Taking this one step further, we can integrate under each curve to summarize model performance across all possible thresholds.

- Area under the curve (AUC) is equal to the probability that a classifier will rank a subject with disease higher than a healthy subject.
  - Assuming 'positive' ranks higher than 'negative'.
This was the final lecture of the course’s main content

• So far you have
  – Data Understanding, Cleaning, and Transformation
  – Statistical Methods
    • Comparing 2+ groups with parametric or non-parametric groups
  – Modeling
    • Regression, Classification, Survival Analysis
  – Validation
    • Evaluation frameworks and metrics
This was the final lecture of the course’s main content

- Next few weeks:
  - Special Topics: Clinical Text Analysis
  - Special Topics: Data Privacy and IRB
  - Case study:
    - Here you will be using all three parts of this class to explore some data, create and validate some models in an effort to answer the question:
      - Are patients more likely to die when admitted to the ICU on a weekend?