Advanced Institute for Artificial Intelligence

Classification Evaluation

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Receiver Operating Characteristic

Jupyter Notebook: `https://bit.ly/2Uzu8oL`

Credits

- Examples and images taken from:
  - https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226
  - https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
  - https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
  - https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

Quick Recap

Confusion Matrix:

		Actual class	
		P	**N**
Predicted class	**P**	**TP**	FP
	N	FN	**TN**

where P = Positive; N = Negative; TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative

Quick Recap

- Classification problem:
  - Decide to predict the class values
- Predict the probabilities for each class instead
  - Capability to choose and calibrate the threshold for how to interpret the predicted probabilities
  - E.g., the threshold of 0.5:
    - Probability in [0.0, 0.49] is a negative outcome (0), and a
    - Probability in [0.5, 1.0] is a positive outcome (1)
- Threshold can be adjusted to tune the behavior of the model for a specific problem:
  - An example would be to reduce more of one or another type of error

Quick Recap

- Adjusting the threshold:
  - In a diagnosis prediction system, we may be far more concerned with having low false negatives than low false positives.
  - A false negative would mean not warning about a given disease, leading the patient to not worrying about a certain condition
  - A false positive would mean to start a treatment that they don't need
- A common way to compare models that predict probabilities for two-class problems is to use a ROC curve.

ROC Curves

- An ROC curve plots TPR vs. FPR at different classification thresholds.
- Lowering the classification threshold classifies more items as positive
  - **Increases both False Positives and True Positives**
- Conversely, elevating the classification threshold classifies more items as negative
  - Increases both False Negatives and True Negatives
- Example: If it is a cancer classification application you don't want your threshold to be as big as 0.5. Even if a patient has a 0.3 probability of having cancer you would classify him to be 1

ROC Curves

- A useful tool when predicting the probability of a binary outcome is the Receiver Operating Characteristic curve
- Plot of the false positive rate (x-axis) versus the true positive rate (y-axis)
  - Try a number of different candidate threshold values between 0.0 and 1.0
- True positive rate describes how good the model is at predicting the positive class when the actual outcome is positive
  - The true positive rate is also referred to as **sensitivity**
- The false positive rate is also called the **false alarm rate** as it summarizes how often a positive class is predicted when the actual outcome is negative.

Quick Recap

True Positive Rate (TPR) is a synonym for **recall** and is therefore defined as follows:

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

False Positive Rate (FPR) is defined as follows:

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

ROC Curves

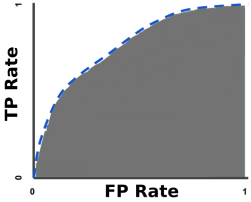
- Very useful tool
- Area Under the Curve (AUC) can be used as a summary of the model skill
- Smaller values on the x-axis of the plot indicate lower false positives and higher true negatives
- Larger values on the y-axis of the plot indicate higher true positives and lower false negatives

ROC Curves

- Good models have curves that bow up to the top left of the plot
- A skillful model will assign a higher probability to a randomly chosen real positive occurrence than a negative occurrence on average

AUC: Area Under the ROC Curve

- AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).
- provides an aggregate measure of performance across all possible classification thresholds
- probability that the model ranks a random positive example more highly than a random negative example



AUC: Area Under the ROC Curve

- AUC ranges from 0 to 1:
  - 100% wrong has an AUC of 0.0; 100% correct has an AUC of 1.0.
- scale-invariant
- in the cost of false negatives vs. false positives, it may be critical to minimize one type of classification error

ROC Curve - Python Code

```python
import numpy as np
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

fpr, tpr, thresholds = roc_curve(y, scores, pos_label=2)
roc_auc = auc(fpr, tpr)
print("AUC: {}".format(roc_auc))
plt.plot(fpr, tpr)
```

Cross Validation

Jupyter Notebook: https://bit.ly/3pDra0K

Credits

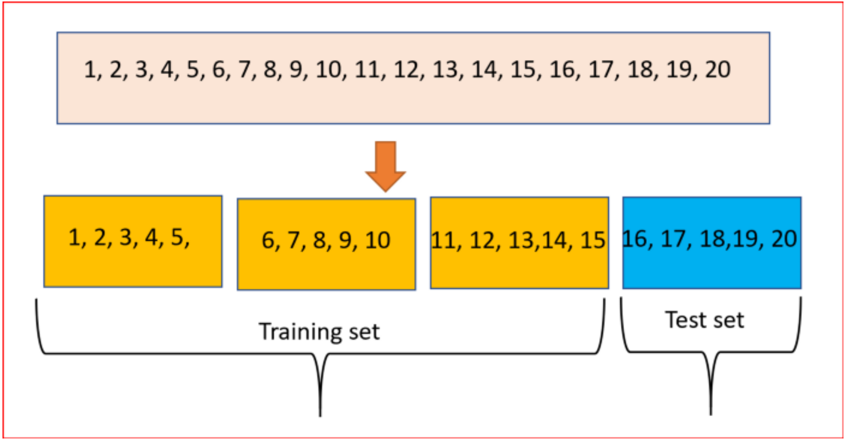
Examples and images taken from:

- https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e
- https://machinelearningmastery.com/k-fold-cross-validation/
- https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/
- https://scikit-learn.org/stable/modules/cross_validation.html

Cross Validation

- Procedure used to estimate the performance of a machine learning algorithm
- Cross-validation involves fitting and evaluating n models
- Provides n estimates of a model's performance on the dataset
  - Use summary statistics such as the mean and standard deviation
  - This score can then be used to compare and ultimately select a model and configuration to use as the "final model" for a dataset

Cross Validation



Cross Validation

Cross Validation error approximates the *True Test* error

$$\boxed{cv_{(k)} = \frac{1}{k}\sum_{1}^{k} MSE_i}$$

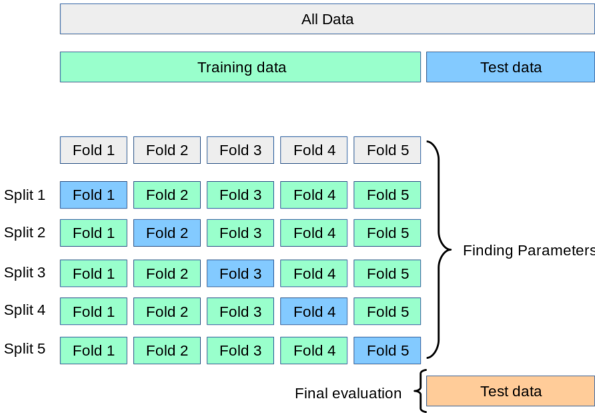
k-Fold Cross Validation

- Resampling procedure used to evaluate machine learning models on a limited data sample
- Simple to understand and generally results in better models than simple train/test split.
- Preparation of the data prior to fitting the model occur on the CV-assigned training dataset
- k=10 is very common in applied machine learning

k-Fold Cross Validation Algorithm

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    1. Take the group as a hold out or test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

k-Fold Cross Validation



k-Fold Cross Validation

- **Advantages:**
  - Computation time is reduced
    - Repeated the process only k times
  - Reduced bias
  - Every data points get to be tested exactly once and is used in training k-1 times
- **Disadvantages:**
  - When compared to simple train/test split, this approach is computationally intensive as the algorithm has to be rerun from scratch k times

k-Fold Cross Validation - Python Code

```python
from numpy import array
from sklearn.model_selection import KFold
# data sample
data = array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6])
# prepare cross validation
kfold = KFold(3)
# enumerate splits
for train, test in kfold.split(data):
  print("train: {}, test: {}".format(data[train], data[test]))
```

Leave One Out Cross Validation - LOOCV

- Divide the data set into two parts:
  - In one part we have a single observation (our test data)
  - The other part, we have all the other observations from the dataset(training data)
- In a dataset with n observations then training data contains n-1 observation and test data contains 1 observation
- This process is iterated for each data point as shown below. Repeating this process n times generates n MSEs.

LOOCV - Python Code

```python
from numpy import array
from sklearn.model_selection import LeaveOneOut
# data sample
data = array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6])
# prepare cross validation
looc = LeaveOneOut()
# enumerate splits
for train, test in looc.split(data):
  print("train: {}, test: {}".format(data[train], data[test]))
```

Leave One Out Cross Validation - LOOCV

- **Advantages:**
  - Far less bias
    - Use the entire dataset for training
  - No randomness in the training/test data
    - Performing LOOCV multiple times will yield same result
- **Disadvantages:**
  - MSE will vary as test data uses a single observation
    - If the data point is an outlier than the variability will be much higher.
  - Execution is expensive as the model has to be fitted n times
    - **Don't Use LOOCV: Large datasets or costly models to fit!**

scikit-learn Pipelines

Jupyter Notebook: https://bit.ly/38TWRw0