CS500-Data Science Tools and Technique
Model Evaluation

Bahar Ali
PhD Scholar,
National University Of Computer and Emerging Sciences,
Peshawar.

Model Evaluation

Central Question:

How good is a model at
classifying unseen records?

4.1 Metrics for Model Evaluation

- How to measure the performance of a model?

4.2 Methods for Model Evaluation

- How to obtain reliable estimates?

Metrics for Model Evaluation

- Focus on the predictive capability of a model
- Rather than how much time it takes to classify records or build models.

Confusion Matrix

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	True Positives	False Negatives
	Class=No	False Positives	True Negatives

Accuracy and Error Rate

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

$$\text{Error Rate} = 1 - Accuracy$$

	PREDICTED CLASS		
		Class= Yes	Class= No
ACTUAL CLASS	Class= Yes	TP 25	FN 4
	Class= No	FP 6	TN 15

$$\text{Acc} = \frac{25 + 15}{25 + 15 + 6 + 4} = 0.80$$

The Class Imbalance Problem

- Sometimes, classes have very unequal frequency
  - Fraud detection: 98% transactions OK, 2% fraud
  - E-commerce: 99% surfers don’t buy, 1% buy
  - Intruder detection: 99.99% of the users are no intruders
  - Security: >99.99% of Pakistani are not terrorists
- The class of interest is commonly called the positive class, and the rest negative classes.
- Consider a 2-class problem
  - Number of negative examples = 9990
    Number of positive examples = 10
  - If model predicts all examples to belong to the negative class, the accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any positive example.

Precision and Recall

Alternative: Use measures from information retrieval which are biased towards the positive class.

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Majority

$$p = \frac{TP}{TP + FP} \qquad r = \frac{TP}{TP + FN}$$

Precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

Precision and Recall – A Problematic Case

	Classified Positive	Classified Negative
Actual Positive	1	99
Actual Negative	0	1000

- This confusion matrix gives us
  - precision p = 100%
  - recall r = 1%
- because we only classified one positive example correctly and no negative examples wrongly.
- Thus, we want a measure that
  1. combines precision and recall and
  2. is large if both values are large.

F_1-Measure

- F_1-score combines precision and recall into one measure.

$$F_1 = \frac{2pr}{p+r}$$

F_1-score is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{p}+\frac{1}{r}}$$

- The harmonic mean of two numbers tends to be closer to the smaller of the two.
- Thus, for the F_1-score to be large, both p and r must be large.

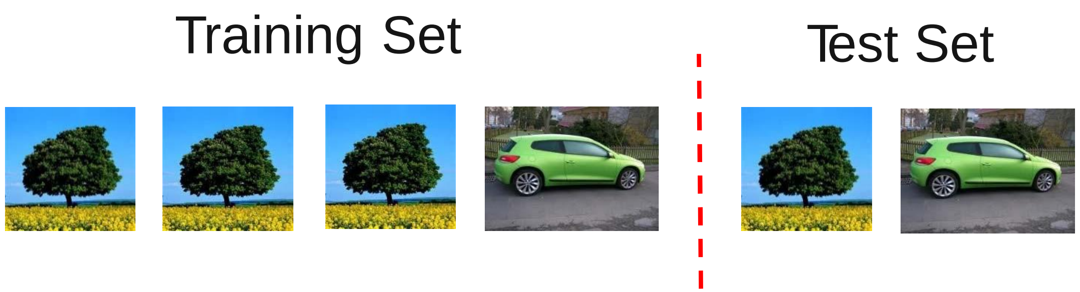
Methods for Model Evaluation

– Methods for estimating the performance measures discussed:

1. Holdout Method
2. Random Subsampling
3. Cross Validation

Holdout Method

- The *holdout* method reserves a certain amount of the labeled data for testing and uses the remainder for training.
- Usually: One third for testing, the rest for training



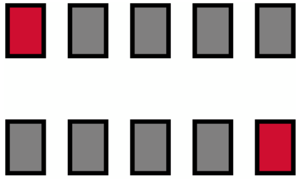
- For unbalanced datasets, random samples might not be representative
  - few or no records of the minority class/classes
- *Stratified sample:* Sample each class independently, so that records of the minority class are present in each sample.

Random Subsampling

– Holdout estimate can be made more reliable by repeating the process with different subsamples

- In each iteration, a certain proportion is randomly selected for training
- The error rates on the different iterations are averaged

Cross-Validation



- *Cross-validation* avoids overlapping test sets
  - First step: data is split into *k* subsets of equal size
  - Second step: each subset in turn is used for testing and the remainder for training
- This is called *k-fold cross-validation*
- The error estimates are averaged to yield an overall error estimate
- Frequently used: k = 10 (90% training, 10% testing)
  - Why ten? Experiments have shown that this is the good choice to get an accurate estimate and still use as much data as possible for training.
- Often the subsets are generated using stratified sampling

Evaluation Summary

- Performance Metrics
  - Use accuracy
  - If interesting class is infrequent, use precision, recall and F1
- Estimation
  - Use cross-validation
  - If dataset is large and computation takes to too much time, use holdout method