CS578 – Interactive and Transparent Machine Learning

Topic: Evaluation

Mustafa Bilgic

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

How do we know if our model is any good?

- What is the performance metric that is relevant for the task at hand?
- How do we use (limited) amount of data to make claims about future performance of the model?
- Given a model and its performance, how do we know if it's any good?

Types of Errors – Classification

- Assume a target/positive class
  - Spam, HasHeartDisease, etc.
- *False positive*
  - Falsely classifying an object as positive
    - E.g., classifying a legitimate email as spam, diagnosing a healthy patient as having heart disease, and so on
  - Also called *Type I* error
- *False negative*
  - Falsely classifying an object as negative
    - E.g., classifying a spam email as not-spam, claiming that a heart-disease patient is healthy, and so on
  - Also called *Type II* error

A Few Performance Measures

- 0/1 loss; error or accuracy
- Precision
- Recall
- F1
- Log-loss
- MSE
- MAE
- RSE

Confusion Matrix

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

ACCURACY

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

$$Accuracy = \frac{Num\ Correct}{Data\ Size} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

$$Precision = \frac{True\ Positive}{Predicted\ Positive} = \frac{TP}{TP + FP}$$

True Positive Rate – Recall – Sensitivity

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

$$TPR = Recall = \frac{True\ Positive}{Actual\ Positive} = \frac{TP}{TP + FN}$$

True Negative Rate – Specificity

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

$$TNR = Specificity = \frac{True\ Negative}{Actual\ Negative} = \frac{TN}{TN + FP}$$

False Positive Rate – Fall-out

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

$$FPR = FallOut = \frac{False\ Positive}{Actual\ Negative} = \frac{FP}{TN + FP}$$

False Negative Rate – Miss Rate

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

$$FNR = Miss\ Rate = \frac{False\ Negative}{Actual\ Positive} = \frac{FN}{TP + FN}$$

F1

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

From Wikipedia

		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	**True positive**	**False Negative** (Type II error)	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	**False Positive** (Type I error)	**True negative**	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Making a Classification Decision

- Given a probabilistic output for an object, say $\langle p, 1-p \rangle$, how do we decide which class to assign to this object?
- The simplest approach is check whether $p > 0.5$ and make a decision accordingly
- This assumes each mistakes (False Positives and False Negatives) are equally costly

Equal Misclassification Costs?

- Which one is worse for you:
  - Delivering a spam email into your Inbox (False Negative), or
  - Delivering a legitimate email into your Spam folder (False Positive)?
- If one is worse than the other, then, should we use 0.5 as the decision threshold or should we adjust it to your preference?

Cost Matrix

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	0	a
	Negative	b	0

Given a probability distribution of $\langle p, 1-p \rangle$ for $\langle Positive, Negative \rangle$ respectively, and given the above cost matrix, under what conditions (in terms of a, b, and p) would you classify an object as *Positive*?

Area Under the Curve (AUC)

- **A**rea **U**nder the **C**urve
- What curve? ROC Curve
  - **R**eceiving **O**perating **C**haracteristic
  - The X axis is False Positive Rate
  - The Y axis is True Positive Rate
  - The curve is plotted by varying the "decision" threshold

AUC EXAMPLE

- Assume 10 actual positives and 20 actual negatives
- Plot the ROC curve and compute the area under it for the following cases:
  - P, P, …, P, N, N, …, N
  - P, N, N, P, N, N, …, P, N, N

MAE, MSE, RSE, AND R^2

- r: true value, g: predicted value, D: dataset, M: the size of the dataset
- MAE
  - $\frac{1}{M}\sum_{d \in D}|r[d] - g[d]|$
- MSE
  - $\frac{1}{M}\sum_{d \in D}(r[d] - g[d])^2$
- RSE
  - $\frac{\sum_{d \in D}(r[d]-g[d])^2}{\sum_{d \in D}(r[d]-\bar{r})^2}$
- R^2
  - 1 – RSE

Splitting the dataset

1. Train-test splits
2. Train-validation-test splits
3. Cross-validation

Train-Test Split

- Randomly split the data into two disjoint sets
- A typical approach: 2/3 for train and 1/3 for test
- Train your model on training data and evaluate it on the test data
  - Use your favorite performance metric
- Report your performance as the expected performance on unseen data
- Caveats:
  - You need a large dataset for this to work
  - You cannot tune your parameters on the test data

Train-Validation-Test Split

- Split your data into three disjoint sets
  - Train, validation, test
- Train your model(s) on the training data
- Evaluate your model(s) on the validation data
- Pick the model that performs best on the validation data
- Test the model on the test data, and report its performance
- Caveat:
  - You need a really big dataset for this to work

Cross-Validation

- Split your data into k disjoint sets
- Each time, one set is the test set and the rest is the training set

Real Life Measures

- Not as clean as the ones we discussed
- Imagine self-driving cars, medical diagnosis, crime prediction, fraud detection, and so on
- Usually, there is not a single performance measure
- Performance is handled on a case-by-case basis; not on an aggregate level