Week 18 - Evaluating Supervised Learning Algorithms

Regression Metrics

Regression Metrics

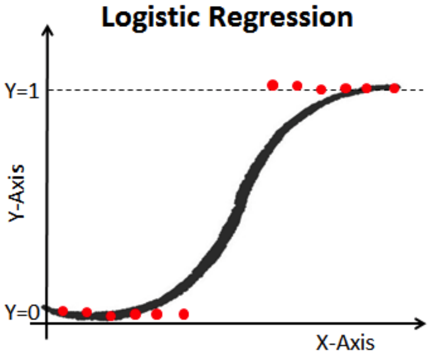
1. R-Squared: the proportion of the variance in the model predictions that is predictable based on the input values.

2. Mean Squared Error (MSE): takes the difference between the predicted value and the actual value, squares it, then takes the average (mean) of those values.

3. Root Mean Squared Error (RMSE): The square root of the MSE.

Classification Metrics

Revision: Logistic Regression



Logistic Regression line represents the likelihood that a particular outcome will occur based on the features provided. Assignment to the two classes is based on the probability cutoff (usually 0.5).

Revision: Logistic Regression

Let's take the case of disease classification: we have a dataset in which the features are patient symptoms, and the outcome is whether or not they have a particular disease (true/false outcome).

Thus, our possible outcomes will be

- 1 - patient has the disease
- 0 - patient is healthy

Divide data into training and test set, run the model, and obtain predicted values for all of the observations in the test set to compare them to the real outcomes.

How can we evaluate the classification?

Commonly used metric:

Accuracy - ratio of total correct predictions to the total number of predictions made.

What's the problem with that?

How can we evaluate the classification?

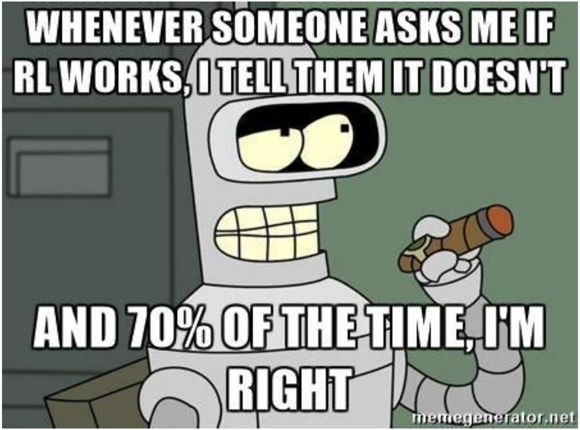
Commonly used metric:

Accuracy - ratio of total correct predictions to the total number of predictions made.

What's the problem with that?

Imagine we are investigating a rare disease that only 1% of patient have. Now we make a model which always predicts "healthy" for every user - we will classify 99 % of observations correctly. Does that make it a good model?

Accuracy


WHENEVER SOMEONE ASKS ME IF RL WORKS, I TELL THEM IT DOESN'T
AND 70% OF THE TIME, I'M RIGHT
memegenerator.net

Correctly Classified Observations

1. Count the number of correctly predicted positive cases we have: for how many sick patients did our algorithm correctly predict they had the disease? This number is known as the number of **true positives**.

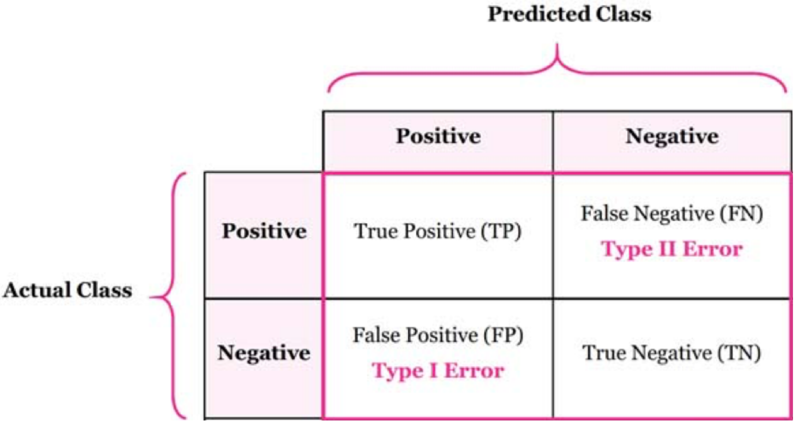
2. Calculate the other half of this number, the number of healthy patients the algorithm correctly predicted to be healthy, which are the **true negatives**.

Incorrectly Classified Observations

1. Calculate those sick patients who we incorrectly classified as healthy. Because these individuals are actually sick and we predicted them to be healthy (the negative class, meaning they are not sick), we call this group **false negatives**.

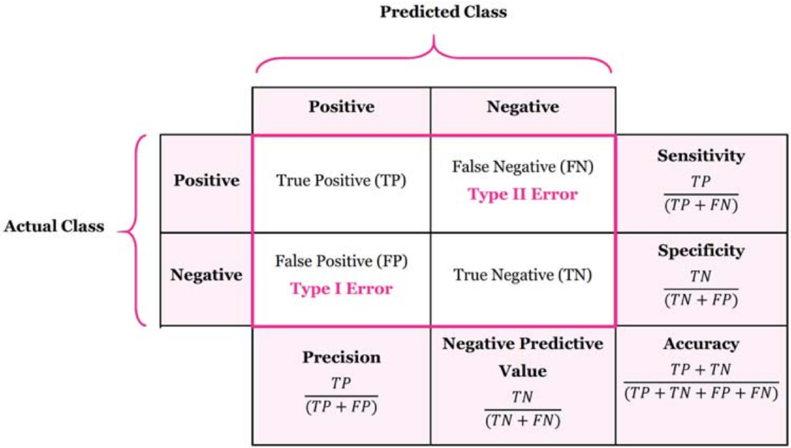
2. Those healthy patients whom our algorithm classified as sick form a part of the group of false positives, as we incorrectly classified them into the positive class.

Confusion Matrix



		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)

Confusion Matrix



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Recall

- Recall is also called Sensitivity or True Positive Rate
- It is the number of true positives divided by all those results whose true value belongs to the positive class (the sum of true positives and false negatives).

$$\text{Recall} = TP/(TP + FN)$$

- Intuitively, recall is the number of relevant items (actually sick patients) that were correctly identified.

Precision

- Precision is also called Positive Predictive Value
- It is the total number of true positives divided by all those values we predicted to be positive (the sum of true positives and false positives).
$$\text{Precision} = TP/(TP + FP)$$
- In our example, our algorithm would achieve 100% precision if all of those patients it predicted to be sick, were actually sick, no matter how many sick people we might have misclassified as healthy.

F-1 Score

The F-1 Score is the harmonic mean of Precision and Recall.

$$\text{F1 score} = \frac{2 * (\text{Precision * Recall})}{(\text{Precision} + \text{Recall})}$$

Example:

Precision $= 0.8$, Recall $= 0.4$

$$\text{F-1 Score} = \frac{2 * 0.8 * 0.4}{0.8 + 0.4} = 0.533$$

The harmonic mean penalizes if one of the measures is low!

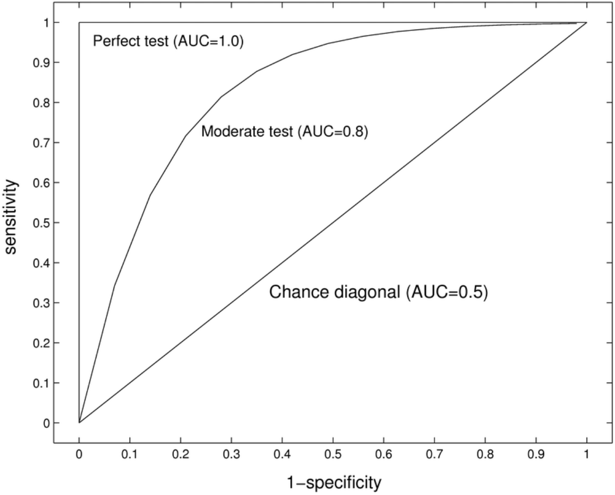
AUC - ROC

- AUC - ROC stands for ”Area Under the Curve - Receiver Operating Characteristics” and is a graphical method for evaluating algorithms

- It plots the True Positive Rate (Recall or Sensitivity) against the False Positive Rate. The AUC-ROC measures performance by

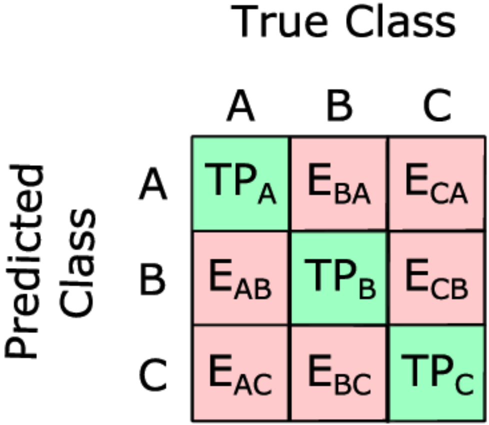
  calculating the area under the curve created by plotting these two values against each other, and is bounded between 0 and 1.

AUC - ROC



Perfect test (AUC=1.0)
Moderate test (AUC=0.8)
Chance diagonal (AUC=0.5)
sensitivity
1−specificity
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Confusion Matrix for Multiclass Problems



		True Class		
		A	B	C
Predicted Class	A	TP_A	E_{BA}	E_{CA}
	B	E_{AB}	TP_B	E_{CB}
	C	E_{AC}	E_{BC}	TP_C

Recall for Multiclass Problems

- Remember, Recall is the number of true positives divided by all those results whose true value belongs to the positive class (such as the number of sick patients correctly identified)

- For multiclass problems, we can calculate this for each class, and then find an average

$$Recall = TP_A/(TP_A + FN_A)$$

- For example, for class A, TP is the green square TP_A and (TP + FN) is the sum across all fields where the **true class** is A.

Precision for Multiclass Problems

- Remember, Precision is the total number of true positives divided by all those values we predicted to be positive (such as how many patients with a positive test result are actually sick)

- For multiclass problems, we can calculate this for each class, and then find an average

$$Recall = TP_A/(TP_A + FP_A)$$

- For example, for class A, TP is the green square TP_A and (TP + FP) is the sum across all fields where the **predicted class** is A.