COMP4388: MACHINE LEARNING

Accuracy Measure

Dr. Radi Jarrar
Department of Computer Science
Birzeit University

Model accuracy

- How does a generated model, *m*, perform on data from domain *D*?
- Which of the generated models, in means of accuracy is best to select given some data from domain *D*?
- How do models produced by some learning algorithm, *A*, perform on data from domain *D*?
- Which of the learning algorithms gives the best model on data from domain *D*?

Model accuracy (2)

- There is a number of approaches that are used to measure the effectiveness of a classification algorithms
- These metrics are useful for evaluating experimental scenarios

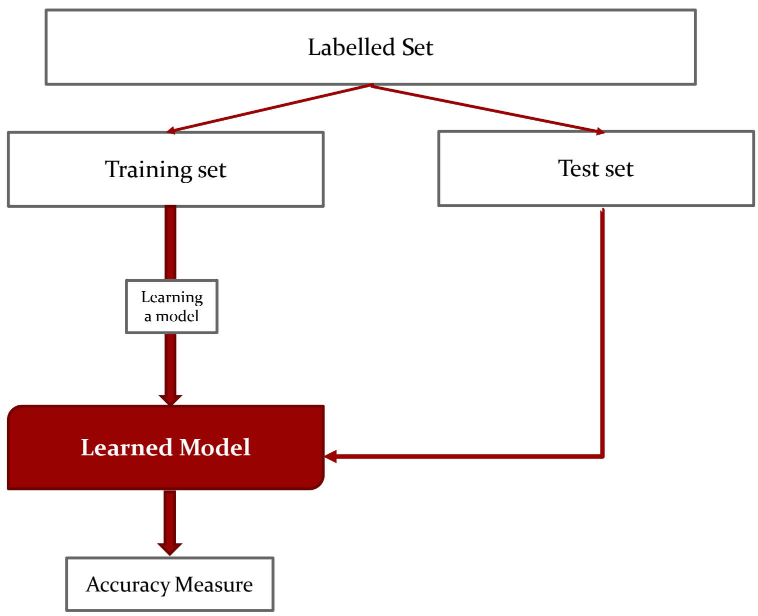
Model accuracy (3)

Regression

- Root Mean Squared Error (RMSE)
- Mean Average Error (MAE)
- R-Square
- Adjusted R-Square

Classification

- Accuracy
- Sensitivity and Specificity
- Precision/Recall/F-score
- Learning Curve
- ROC-AUC curve



EVALUATING REGRESSION MODELS

Evaluating Regression - RMSE

- Root Mean Square Error
- The sample standard deviation of the differences between predicted values and the actual outputs (i.e., the residuals)
- $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(h(x)_i - y_i)^2}$
- The best metric for predicting accuracy for regression
- Simple and present as a default metric for most model

Mean Absolute Error (MAE)

- The average of the absolute difference between the predicted and the actual values
- MAE is a linear score – meaning all individual differences are weighted equally
  - E.g., the difference between 0 and 8 is twice the difference between 0 and 4
- This handles a problem with RMSE as it penalises the higher difference more than MAE (i.e., not very sensitive to outliers)
- $MAE = \frac{1}{N}\sum_{i=1}^{N}|h(x)_i - y_i|$

RMSE vs. MAE

- Case 1: Actual Values = [2,4,6,8] , Predicted Values = [4,6,8,10]
  Case 2: Actual Values = [2,4,6,8] , Predicted Values = [4,6,8,12]
- MAE for case 1 = 2.0, RMSE for case 1 = 2.0
  MAE for case 2 = 2.5, RMSE for case 2 = 2.65
- In general, RMSE will be higher than or equal to MAE
- RMSE is still better to use because the loss function (i.e., cost function) is easier to perform mathematical operations (differentiable)
- If you want to compare two models, MAE is a better choice (easier to interpret and justify)

RMSE vs. MAE

- MAE is more robust to outliers
- MAE minimises the absolute error results in finding the median; whilst RMSE minimises the squared errors over a set of numbers results in finding the mean

R-Squared

- Shows how well features fit a curve or line
- It represents the correlation between the observed outcomes and the predicted outcome values
- $R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - h(x)_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} = 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - h(x)_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2}$
- Notice that the numerator is MSE (i.e., average of squares of the residuals)
- The denominator is the variance in y values
- The higher the MSE the poorer the model
- The higher the R^2 the better the model

Adjusted R-Squared

- The same as R-Squared but it adjusts for the number of terms in a model
- $R^2_{adj} = 1 - \left[\frac{(1-R^2)(N-1)}{N-k-1}\right]$

where N is the total number of observations and k is the number of independent variables

- Adjusted R^2 is always less than or equal to the R^2
- Adjusted R^2 will consider the marginal improvement added by an additional features in the model

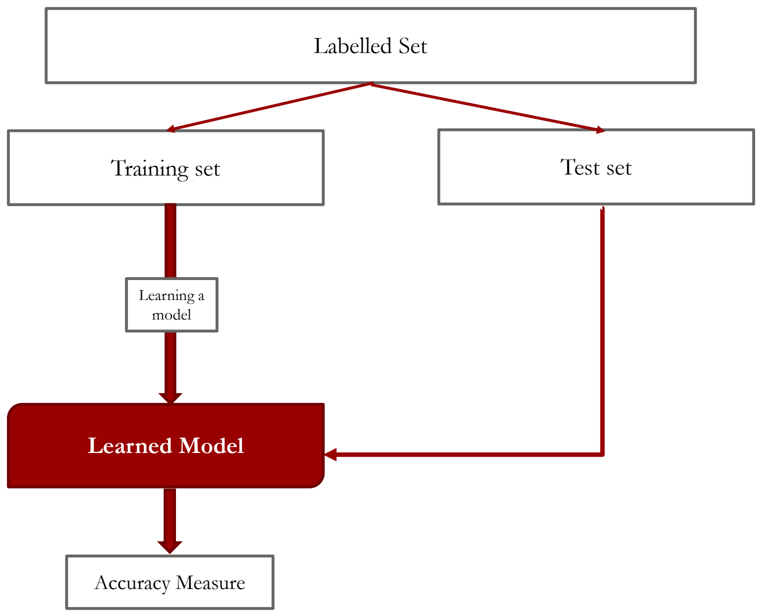
Adjusted R-Squared

- Adjusted R^2 increases if useful features are added and it will decrease if less useful features are added
- However, R^2 increases by increasing features even though the model is not actually improving

Case 1		Case 2			Case 3		
Var1	Y	Var1	Var2	Y	Var1	Var2	Y
x1	y1	x1	2*x1	y1	x1	2*x1+0.1	y1
x2	y2	x2	2*x2	y2	x2	2*x2	y2
x3	y3	x3	2*x3	y3	x3	2*x3 + 0.1	y3
x4	y4	x4	2*x4	y4	x4	2*x4	y4
x5	y5	x5	2*x5	y5	x5	2*x5 + 0.1	y5

	Case 1	Case 2	Case 3
R_squared	0.985	0.985	0.987
Adj_R_squared	0.981	0.971	0.975

EVALUATING CLASSIFICATION MODELS



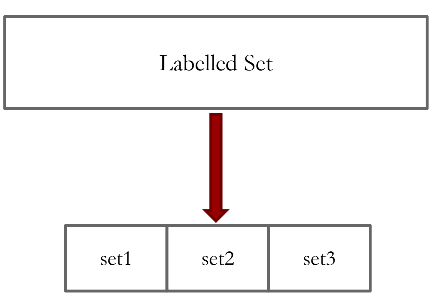
Single Dataset?

- If there is a single dataset, or if the data is small, this will not tell how sensitive accuracy is to a particular training sample
- Larger datasets give better estimations on the accuracy of the model

Cross Validation

- Cross-validation is a technique that is used to avoid overfitting (later in this course)
- In cross-validation, the training dataset is split into a number of folds (subsets) that are used to test the performance of the generated model while the training process is taking place
- Assume a training dataset of 900 records, It can be divided into 3-subsets each of around 300 records namely set1, set2, and set3
- 5-fold and 10-fold cross validations are widely used

Cross Validation (2)



Iteration	Train-on	Test-on

Cross Validation (3)

- E.g., suppose you have 90, using 3-fold cross-validation, estimate the accuracy such that:

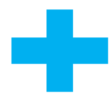
Iteration	Train-on	Test-on	Correctly classified
1	set1, set2	set3	18/30
2	set2, set3	set1	20/30
3	set1, set3	set2	22/30

- Accuracy = 60 / 90 = 0.66 = 66%

Confusion matrix

- The confusion matrix is a well-known method for classification systems
- It contains all information about the actual (the original class label) and the predicted classification assigned by the classification method
- Columns represent predictor's output while the rows represent the actual class labels

Confusion matrix (2)

		Predicted Class	
Actual Class		A	B
	A		
	B		

Confusion matrix (3)

		Predicted Class	
Actual Class		Pos	Neg
	Pos	TP True Positive	FN False Negative
	Neg	FP False Positive	TN True Negative

Confusion matrix (4)

```
(a)  (b)  (c)   <-classified as #confusion matrix
---- ---- ----
 47               (a): class setosa
 41    3          (b): class versicolor
  1   43          (c): class virginica
```


Confusion matrix (5)

- **TP (True positive)** is the number of correct predictions that an instance is positive (classified as class of interest)
- **TN (True negative)** is the number of correct predictions that an instance is negative (not class of interest)
- **FP (False positive)** is the number of incorrect predictions that an instance is negative (incorrectly classified as class of interest)
- **FN (False negative)** is the number of incorrect predictions that an instance is positive (incorrectly classified as not a class of interest)

Accuracy

- The confusion matrix is used to derive a number of performance metrics
- The Accuracy metric measures the proportion of the total number of predictions that were correctly classified
- Used to measure the overall effectiveness of a classifier

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

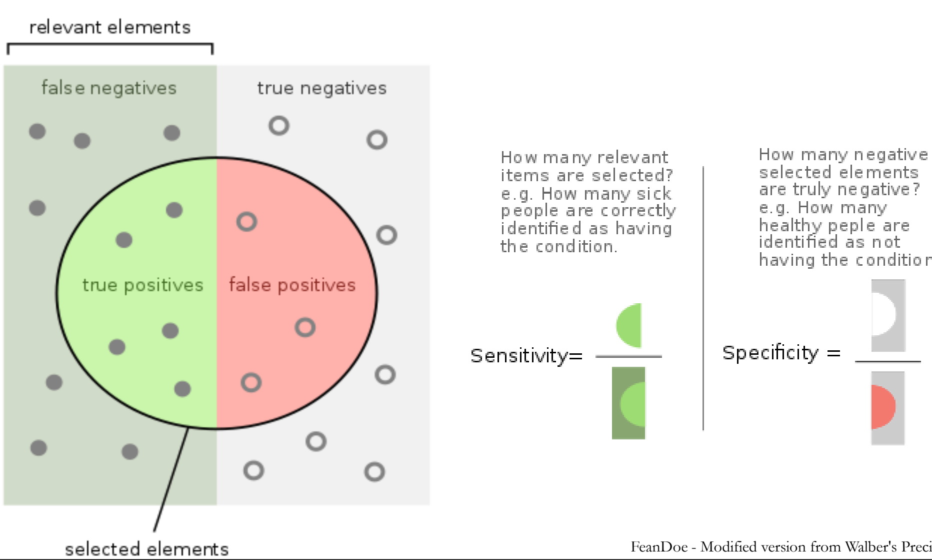
Accuracy (2)

- Is accuracy always good to be used?
  - It is not the best choice when data is imbalanced (i.e., there is a skew in data towards one class)
    - E.g., Is 95% is good when 90% of the data is negative?
  - Cost—Getting a positive wrong costs more than getting a negative wrong
    - E.g., in medical domain, false positives results in wrong tests; however, false negative results in a failure to treat a disease

Error rate

- Is the proportion of incorrectly classified instances
- Error rate = 1 – Accuracy

Sensitivity & Specificity



FeanDoe - Modified version from Walber's Precision and Recall

Sensitivity

- Sensitivity of a model is also called *True Positive Rate*
- It measures the proportion of positive examples that were correctly classified
- E.g., in the health domain, the ability of the model to detect ill patients who have the conditions
- Calculated as the number of true positives (correctly classified) divided by those correctly classified (TP) and those were incorrectly classified (FN)

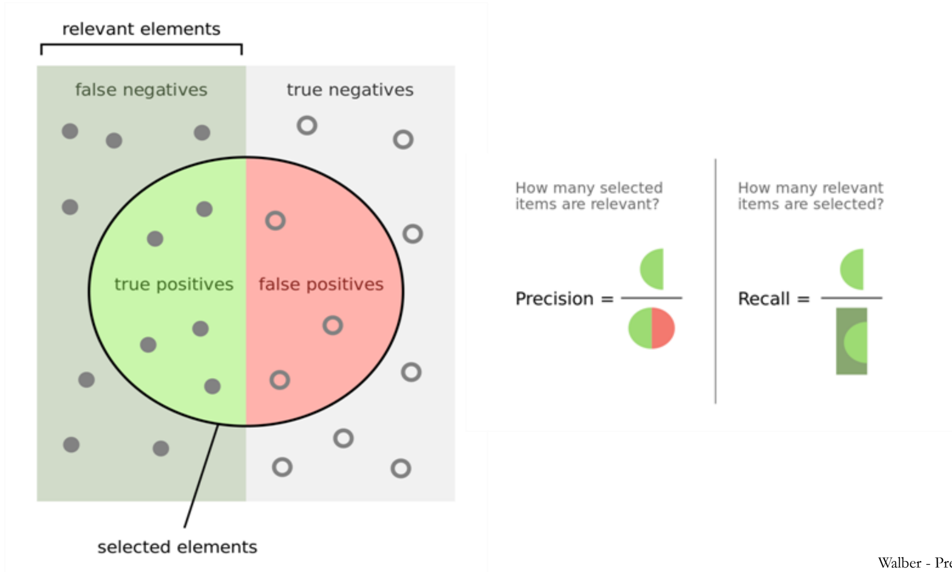
$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity

- Specificity of a model is also called *True Negative Rate*
- It measures the proportion of negative examples that were correctly classified
- E.g., in the health domain, is the proportion of patients with no illness known not to have the disease, who will test negative for it
- Calculated as the number of true negatives divided by the total number of negatives (TN and FP)

$$Specificity = \frac{TN}{TN + FP}$$

Precision & Recall



Walber - Precision and Recall

Precision

- Precision measure the accuracy such that a class has been predicted correctly
- Defines the proportion of positive examples that are correctly classified

$$Precision = \frac{tp}{(tp + fp)}$$

Recall

- Recall measures the completeness of the results (in this context, it is the also the true positive rate or sensitivity)
- It measures the proportion of positive examples that were correctly classified (from the dataset)
- High recall indicates a large portion of positive examples captured in the model

$$Recall = \frac{tp}{(tp + fn)}$$

F-score

- The F-score is a harmonic mean between the precision and recall
- It has the advantage that it combines both the precision and recall in a single value

$$F - score = \frac{2 \times tp}{2 \times tp + fp + fn}$$

Learning Curves

- Learning curves show the effect of the datasets size and the accuracy

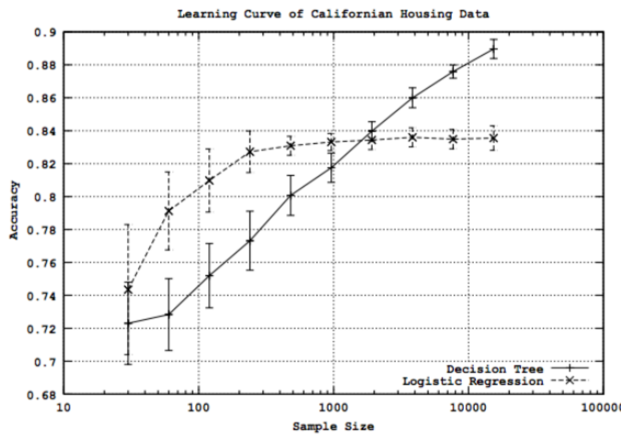


Figure from Perlich et al. *Journal of Machine Learning Research*, 2003

Performance Measure in R

- In R, the package ‘Classification and Regression Training (caret)’ includes many performance measures
- `install.packages(‘caret’)` and `library(caret)`
- A confusion matrix can be shown using cert
- Similar to function table() but the true positive has to be specified
- `confusionMatrix(predicted_type, actual_type, positive = “ClassLabelOfPositive”)`