Artificial Intelligence

Confusion Matrix
for classification algorithms

Confusion Matrix (CM) ?

- An instrument (a table) to assess the outputs of a classification algorithm
  - For binary and...
  - ...for n>2 classes problems
- Rows and columns are both labelled with the existing classes' names
  - The rows represent the "true values"
  - The columns represent the "predicted values"
  - Each cell @ (row, col) counts the number of instances where the actual class in row was predicted as the class in the col
- The diagonal (left-to-right, top-to-bottom) measures then number of instances where the predictions matched the correct true values
  - If there is "no confusion", i.e. true==predicted for all the instances, then all cells *not* in the diagonal will be 0 (zero):
  - A cell(i,j) with i!=j measures the number of instances where:
    - the true classification value is at i's label,
    - but the predicted class is wrong at j's label

Confusion Matrix (CM) - Binary example

- Email Spam (S) or NOT Spam (NS) ?
- y_true: [S, NS, S, NS, S, NS, NS, S]
- y_pred: [S, NS, NS, NS, S, S, NS, S]
- True Negatives: NS correctly classified as NS @ (0,0)
- True Positives: S correctly classified as S @ (1,1)
- False Negatives: S incorrectly classified as NS @ (1,0)
- False Positives: NS incorrectly classified as S @ (0,1)

		Predicted classes	
		NS (0)	S (1)
True classes	NS (0)	3	1
	S (1)	1	3

CM - Binary Classification - Metrics in [0,1]

- Accuracy: correct predictions, relative to all predictions
  - Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision: correct positive predictions, relative to all positive preds
  - Precision = TP / (TP + FP)
- Recall or sensitivity: positive preds, relative to actual positive samples
  - Recall = TP / (TP + FN)
  - Notice: TP+FN is the total number of actual positive samples
  - Interpretation: is the model's ability to capture positive cases
  - Close to 1: almost all positives cases were correctly predicted
- F1 score: the harmonic mean of Precision and Recall
  - Harmonic mean of n parcels = n / (1/parcel 1) + ... + (1/parcel n)
  - F1 = 2 * (Precision * Recall) / (Precision + Recall)
  - Interpretation: useful for comparing models; higher values mean better at capturing positive cases while minimizing FN
  - Close to 1: *both* high precision + high recall

Confusion Matrix - 3-classes example

- Cat, dog or fox?
- y_true: [C, D, F, F, C, D, D, C]
- y_pred: [C, F, D, F, C, C, D, C]
- Overall Accuracy = fraction of correct predictions
- Precision, Recall and F1 score per class
- Macro averages = avg of precision, recall, F1, per class
- Aggregated-averages = average metrics, for all classes

		Predicted classes		
		Cat	Dog	Fox
True classes	Cat	3	0	0
	Dog	1	1	1
	Fox	0	1	1

CM - Multi-class Classification - Metrics

- Overall Accuracy: fraction of correct predictions, relative to all predictions
  - Numerator = for each class C, sum the TP for C
  - Denominator = total number of instances
  - Accuracy = Numerator / Denominator
- Precision per class: positive predictions, relative to all positive predictions, for the class
  - Precision for C = TP for C / (TP for C + FP for C)
- Recall per class or sensitivity: fraction of positive preds, relative to actual positive samples, for the class
  - Recall for C = TP for C / (TP for C + FN for C)
- F1 score per class: the harmonic mean of Precision and Recall
  - F1 = 2 / (1/P + 1/R)
  - F1 = 2 * (Precision for C * Recall for C) / (Precision for C + Recall for C)

CM - Multi-class Classification - Metrics

- Ac = Accuracy for class c
  - (TPc + TNc) / (TPc + TNc + FPc + FNc)
- Pc = Precision for class c
  - TPc / (TPc + FPc)
- Rc = Recall for class c
  - TPc / (TPc + FNc)
- F1-score for class c
  - 2 * (Pc * Rc) / (Pc + Rc)

CM - am_confusion_matrix tools

- At: https://github.com/amsm/am_confusion_matrix
- There is a set of tools for creating the confusion matrix
  - From simple lists of labels and their corresponding predictions
- It includes functions that compute the
  - True Positives
  - False Positives
  - False Negatives
  - True Negatives
  - All per-class and with notation that discriminates the errors
- And the metrics
  - Accuracy
  - Precision,
  - Recall
  - F1-score
  - All with some explainability, including step-by-step calculations

References

- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
- https://github.com/amsm/am_confusion_matrix