(14th March, Week 3)

Cross-Validation (CV)

Building ontop to hyper parameter discovery methods, we did a crude 70/30 split of training and testing data, and within the training data we split again via 70/30 ratio the training data set to surface what we call the validation set.

This splitting is interesting because it feels like we're suboptimally training a model on only 70% of the data. Is there a method in which we can train on more then that, whilst still being able to self refer an accuracy measure? A long wind up to **yes - indeed we can sir**.

Welcome to folding

We parition our data set into a k-fold. Where we proceed to approach an iteration, specifically iterating k times... each iteration we mark a unique to other iteration fold as the iterations "validation" set. We train on the other $k-1$ iterations and validate on the "validation" set of that iteration.

Each iteration provides some "score" that we in combination average out for our final evaluation of the model.

The higher the fold, the more accurately representing the run's score is of what it would be "in real application". Usually you will see leave one out, 10, 5, and 3 as your k parameter...

Classifier Evaluation Metrics

Accuracy; $\frac{TP+FP}{all}$ is not always enough as a evaluation metric of a classifier. Perhaps you have a model target of proportion of false positives or target proportion of false negatives etc...

0.0.1 Confusion Matrices

Given m classes, an entry $CM_{i,j}$ in a confusion matrix indicates the number of tuples in class i that where labeled by the classifier as j.

Note; FP is often called type I error - reject the true null hypothesis and FP is often called type II error - reject the false null hypothesis.

0.0.2 Other Metrics

- Precision/Exactness: $P = \frac{TP}{TP+FP}$
- Recall/Completeness: $P = \frac{TP}{TP+FN}$

Inverse relationship between precision and recall.

0.0.3 F-Measure, F-Score

Harmonic mean of precision and recall...

$$F_b = (1 + \beta^2)\frac{P_{\text{precision}} \cdot P_{\text{recall}}}{(\beta^2 \cdot P_{\text{precision}}) + P_{\text{recall}}}$$

- In general, it is the weighted measure of precision & recall
- β is a weight - assign β times as much weight to recall as to precision.

precision is more important, so choose higher β and vice versa re: recall

Now this F measure, (known in ”scikit-learn” as ”f1”) can be viewed as a way of tackling heavily skewed data. Example; a dataset that is made 95% ”falsy”. A ”good” classifier may be more weighted towards one that can provide reliable visibility on those curious cases that make up 5% of the dataset. As this provides contrast to a well behaving mode/baseline classifier (as it'd have 95% accuracy), an f-measure may be more favourable.

0.0.4 ROC Curves

ROC (Receiver Operating Characteristics) curves: for visual comparision of models. Originated from signal detection theory...

Applicable when your classification model is trimmed in a way where it provides ”estimate probabilities” for wether a classification is suitable or not for a sample. You need this as this ROC curve is drawn as you shift the ”threshold” needed for a classifiers estimate to be turned into a deemed classification. Used to tune this threshold.

Shows the trade-off between the true positive rate TPR and false positive rate FPR...the area under the ROC curve (AUROC) is a measure of the accuracy of the model and the AUROC of different classification models can be compared.