Advanced Data Classification

- **Performance measures and the Confusion Matrix**
- **Bias and variance**
- **Other validation approaches (e.g., Cross-Validation)**
- **Class imbalance**
- **Model ensembles**

MIS772
Predictive Analytics

Advanced Data Classification
making classification better

Refer to your textbook by Vijay Kotu and Bala Deshpande, *Data Science: Concepts and Practice*, 2nd ed, Elsevier, 2018.

DEAKIN BUSINESS SCHOOL

AACSB ACCREDITED

EFMD EQUIS ACCREDITED

Interpreting Confusion Matrices

Understanding accuracy, precision & recall

DEAKIN BUSINESS SCHOOL

Accuracy, Recall & Precsion

Actual

Predicted	True Positives	False Positives
	False Negatives	True Negatives

accuracy: 87.39%

	true Low	true High	class precision
pred. Low	67	10	87.01%
pred. High	5	37	88.10%
class recall	93.06%	78.72%	

- **Accuracy**:
  - Proportion of correctly classified data points among the total number of data points in the test set
  - $Accuracy = (TP + TN)/(TP + FP + TN + FN)$ = (67+37)/(67+10+37+5)=0.8739=87.39%
- **Precision**:
  - Proportion of data points that have been correctly classified as belonging to the target class among all the cases that have been classified as belonging to the target class:
  - $Precision = TP/(TP + FP)$. $Precision\_Low$= 67/(67+10)=0. 8701=87.01%; $Precision\_High$=37/(37+5)= 88.10%
- **Recall**:
  - Proportion of data points that have been correctly classified as belonging to the target class among all the cases that are actually in the target class:
  - $Recall = TP/(TP + FN)$. $Recall\_High$=37/(37+10)=78.72%; $Recall\_Low$=67/(67+5)=93.06%

Specificity, f-measure, …

Cross-Validation

Dealing with variance - when doing it twice is better than doing it once

Variance

- ***Variance*** **is the error due to the model sensitivity to small fluctuations in training data, and its inability to generalise to validation data.**

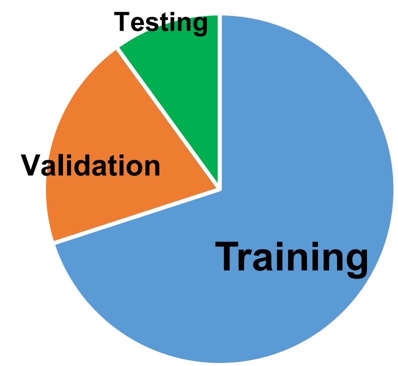
- **The model is *over-fitting* training data when it becomes too sensitive to data variations (commonly *over-trained*), thus learning noise.**

When data is noisy, the model variance will always be high

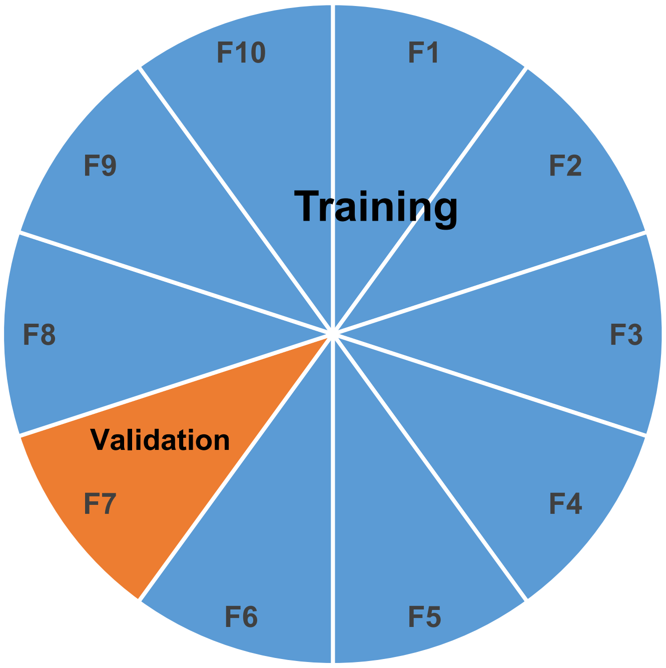
Cross-Validation

- ***Model validation*** **deals mainly with variance problems.**
- ***Holdout method*** **(covered before) validates the model only once and assumes that the training and validations partitions are representative of the population, which may not be correct.**
- ***Cross-validation*** **(CV) method does not make this assumption.**
- **It trains and validates a model using different data samples. Then average performance of all runs is returned. It assumes that all resulting models are similar to that trained using all data.**
- ***k-fold cross-validation*** **splits data randomly into k folds (or parts, e.g. 10). Then k-1 parts are used for model training and 1 part to evaluate it. We do this k times, which ensures that every data point is used in the model training.**
- ***LOOCV*** **(leave one out CV) is a k-fold validation useful for small data sets, which uses n-1 cases for training and 1 for validation. LOOCV may over-fit data.**

Holdout Validation

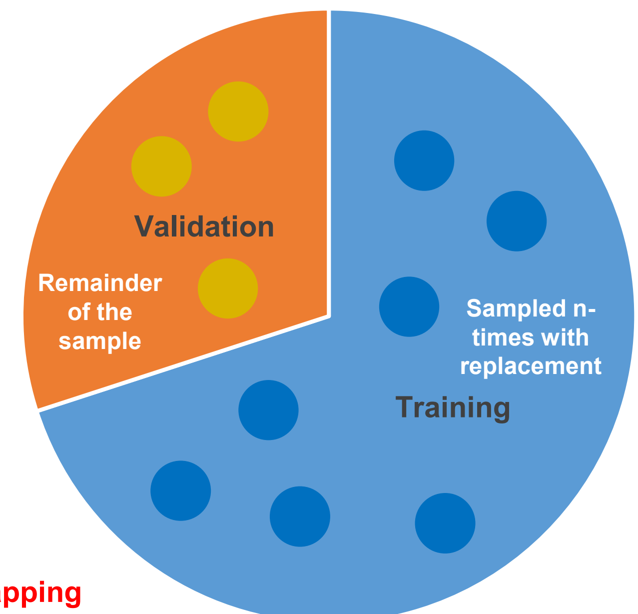


k-Fold Cross-Validation



Bootstrapping

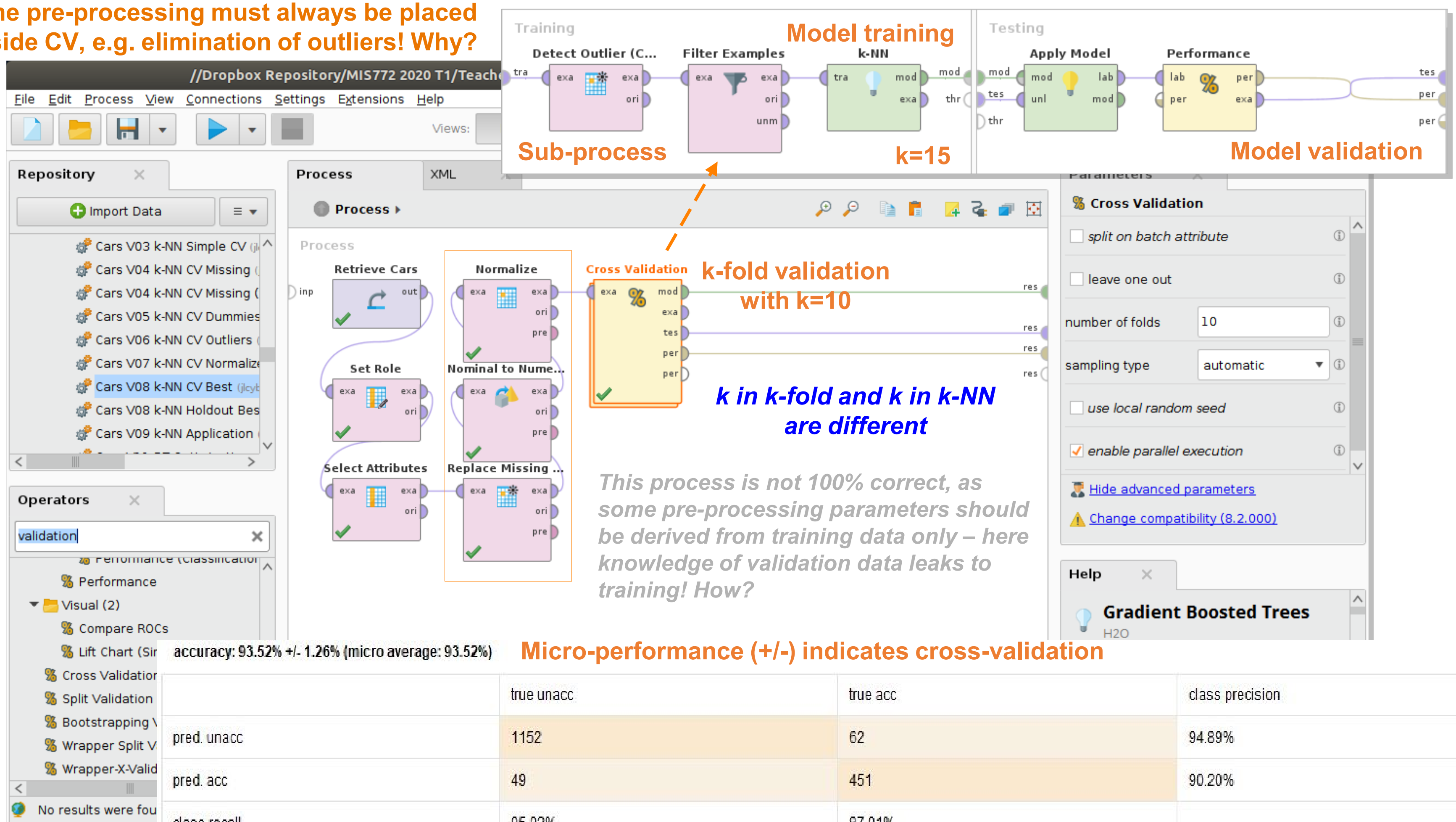
- ***Bootstrapping validation*** **repeated sampling of *k* random examples (with replacement) from a total of *n* examples for training, and the remaining ones used for validation.**
- **On average, in every bootstrap sampling on average only 63.8% of all available data is used in training, and the remaining 36.2% is used for validation.**
- **Because the selection is random, some data will never get used for training and some will never get used for validation.**



Bootstrapping Validation

- **Bootstrap is very useful when we do not have enough data for cross-validation.**
- **It ensures that training and validation data samples are sufficiently large to create and test an unbiased model.**
- **However, because of sampling with replacement, the model tends to over-fit data.**

k-Fold Cross-Validation with RapidMiner



accuracy: 93.52% +/- 1.26% (micro average: 93.52%)

Micro-performance (+/-) indicates cross-validation

	true unacc	true acc	class precision
pred. unacc	1152	62	94.89%
pred. acc	49	451	90.20%
class recall	95.92%	87.91%	

- Data has been loaded
- Label attribute defined
- Predictor attributes selected
- Data has been pre-processed appropriately for the model
- Cross-validation (k-fold or bootstrap) of the model performed
- Performance collected, averaged and reported, e.g. in Confusion Matrix
- Cross-validation performance assessed, e.g. via Accuracy, Kappa, AUC
- Deployment model built using all available data and ready to be applied to new data

Also see KD 8 on Model evaluation

DEAKIN BUSINESS SCHOOL

Balancing Act

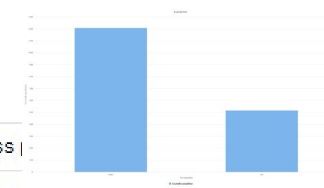
Dealing with the minority class of positive examples

DEAKIN BUSINESS SCHOOL

Problems with Accuracy
Alternative measures

- Often (e.g. fraud detection) we have a *minority class* of *positive* examples (important to us), ***we cannot trust accuracy*** alone as it can be high even though most or all examples are incorrectly classified.
- Instead, we can use *kappa*, which adjusts accuracy based on the distribution of class values.
  Kappa > 0.6 is considered good!
  High accuracy but low kappa is poor!
- Kappa might yield (much) lower values than accuracy.
- Other measures of performance may include *recall* and *precision*, *true / false negative / positive rates*, etc.
- Class recall and precision provide a fine grain model performance.

The label attribute has unbalanced classes



Kappa Performance -
Same confusion table but different measurement

kappa: 0.844 +/- 0.030 (micro average: 0.844)

	true unacc	true acc	class
pred. unacc	1152	62	94.89%
pred. acc	49	451	90.20%
class recall	95.92%	87.91%	

- In cases of *class imbalance*, some classifiers may produce results biased towards the majority class.
- The solution may involve training with a *balanced data sample* by either *over-sampling* the minority class or *under-sampling* the majority class.

- Balancing of training data may lead to a better model.
- Balancing of validation data leads to incorrect accuracy.
- If doing so, accuracy needs to be recalculated.

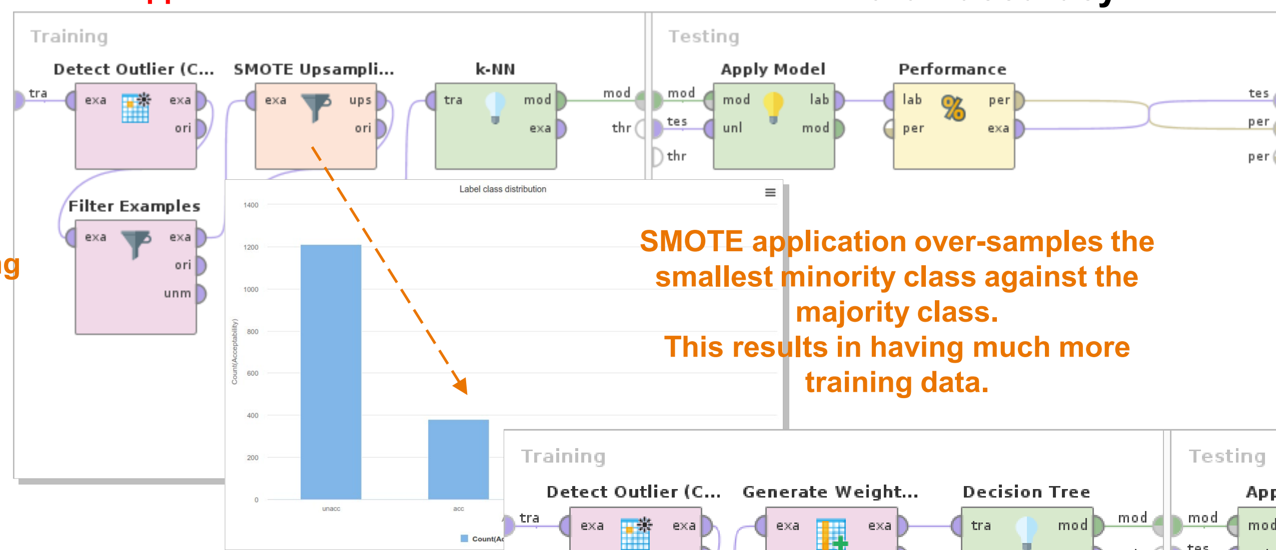
Also see KD 15.5 on
Sampling and balancing data

DEAKIN BUSINESS SCHOOL

Balancing with SMOTE in Cross-Validation

- The easiest way to balance the label classes is to do so only for the model training and to use the unbalanced data for validation
- *SMOTE* (Synthetic Minority Oversampling TEchnique) is one of several operators that helps balancing your data. It creates *synthetic* (not real) data points in the smallest label class.
- As a result we often see the validation performance drop ☹
- *Weighing examples* (by inverse of class frequency) is an alternative approach, which could be used in training, e.g. weights are used (when present) by Decision Trees.
- We can also balance classes by *resampling*, e.g. down-sampling the majority class.
- It is important to study the confusion table to determine if the performance improved in the class of interest rather than overall!
- We could *change the performance measure*, e.g. aim to reduce false-negatives (FN) rather than accuracy.

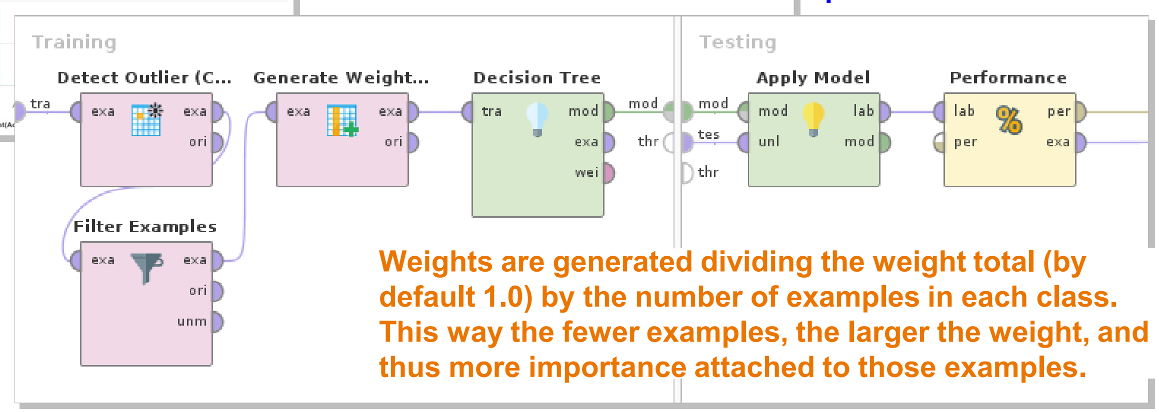
SMOTE Approach



SMOTE applied for model training only!

SMOTE application over-samples the smallest minority class against the majority class.
This results in having much more training data.

In some circumstances, models easily deal with unbalanced samples.
In which case, by balancing data the overall performance may drop!
It all depends on the data!
Always test if sample balancing actually improves the model performance or not.



Weights are generated dividing the weight total (by default 1.0) by the number of examples in each class. This way the fewer examples, the larger the weight, and thus more importance attached to those examples.

Weighing Approach

For sampling and weighing examples see:
https://academy.rapidminer.com/learn/video/sampling-weighting-demo

Playing in an Ensemble

Dealing with bias -
The power of many
over power of one

Bias

- ***Bias*** **is the error resulting from the model's inaccurate assumptions about the population, thus impacting its ability to fit validation data.**
- **Bias may be due to an incorrect model, its parameters or the selection of training data.**
- **The model is *under-fitting* training data when its bias is high, i.e. it is too simple (commonly *under-trained*) to fit validation data.**

When the model is incorrect its bias will always be high

Ensembles

Ensembles **represent the approach called meta-learning, i.e. a method improving model performance by drawing predictions from several models working together.**
All models' *results are returned in aggregate* form, e.g. using the mean or mode.

The ensembles' *value is in the model teamwork*, so that when one model performs poorly on parts of data, regardless how much you train it, other models excel on the same data.
Thus *ensembles reduce bias* of any single model.

Feature Bagging (Random Forests): Model 1, Model 2, Model 3

Bagging (Bootstrap Aggregating): Random Model 3, Random Model 2, Random Model 1

Boosting (AdaBoost or XGBoost): Model 1 → Model 2 → Model 3

Stacking/Voting (on Examples or Features): Model 1, Model 2, Model 3

Row No.	Acceptability	Buying	Maint	Doors	Persons	Lug_Boot	Safety
414	unacc	vhigh	low	5more	2	big	high
415	unacc	vhigh	low	5more	4	small	low
416	unacc	vhigh	low	5more	4	small	med
417	acc	vhigh	low	5more	4	small	high
418	unacc	vhigh	low	5more	4	med	low
419	acc	vhigh	low	5more	4	med	med
420	acc	vhigh	low	5more	4	med	high
421	unacc						low
422	acc						med
423	acc						high
424	unacc						low
425	unacc						med
426	acc						high
427	unacc	vhigh	low	5more	more	med	low
428	acc	vhigh	low	5more	more	med	med
429	acc	vhigh	low	5more	more	med	high
430	unacc	vhigh	low	5more	more	big	low
431	acc	vhigh	low	5more	more	big	med
432	acc	vhigh	low	5more	more	big	high

Strength in variety, e.g. models trained using different algorithms, different sets of model params, different subsets of data, different subsets of attributes.

Also see KD 4.7 on Ensemble learners

More on Ensembles

Bagging (Bootstrap Aggregation) **aims to train and validate multiple models on several random data samples.**

- **Samples with replacement are used for training, to the size equal to that of the entire data set.**
- **The average performance is returned, e.g. as mode or mean.**
- **Rating** 👍👍👍 **(Jacob's humble opinion)**

Random Forests **create many different models, each using a random sample of all attributes. The average performance is returned.**

- **Rating** 👍👍

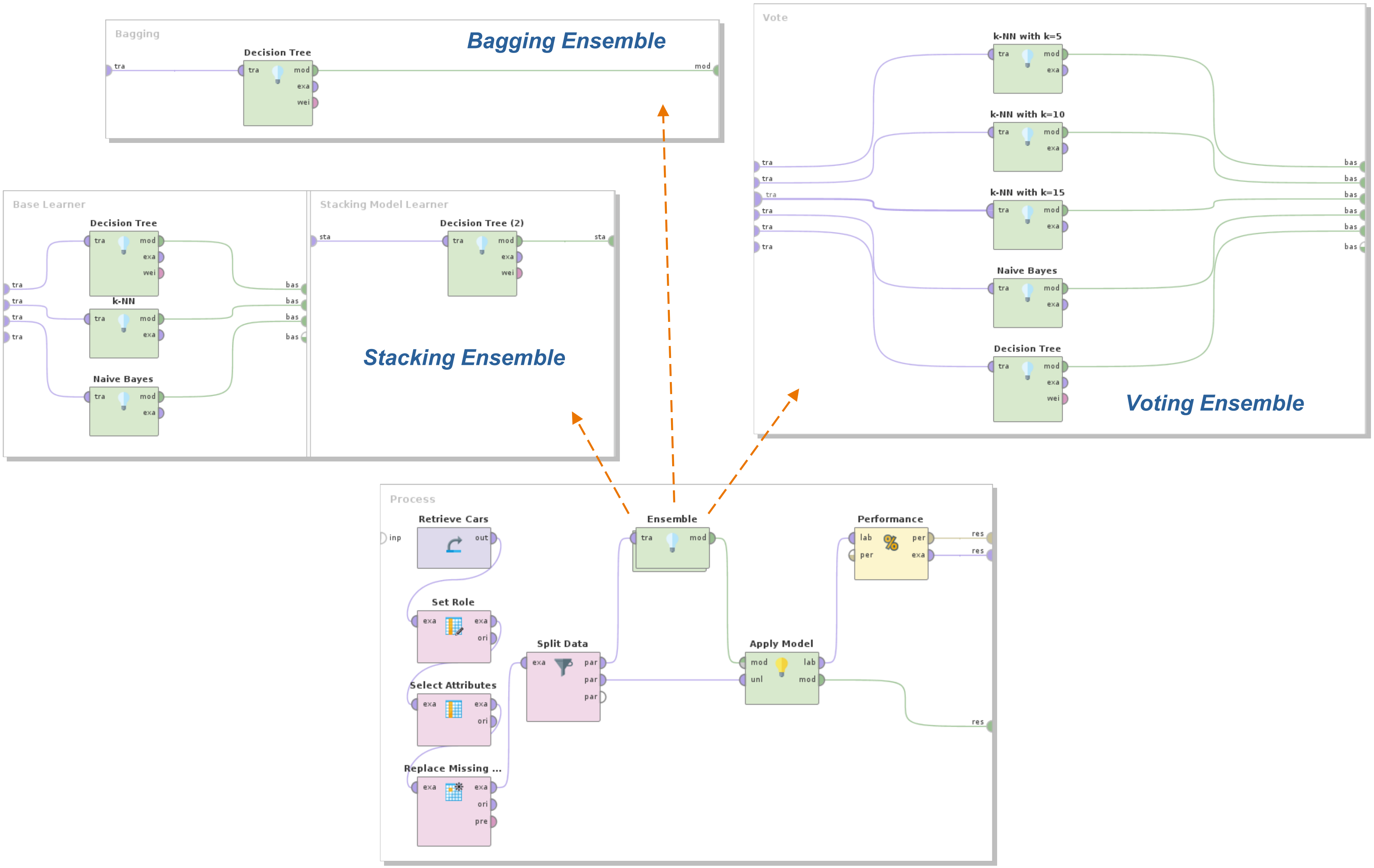
Stacking/Voting **builds multiple models and then determines, e.g., through a higher level learner, which of those models should be used.**
Rating 👍👍👍👍

Boosting **builds models in sequence, where the next model to be added improves the performance of the models in the ensemble, e.g.**

- ***AdaBoost (Adaptive Boosting)*** **identifies and weighs examples on which the previous models failed and fits next models to these examples. Rating** 👍👍👍👍
- ***XGBoost (Gradient Boosting)*** **produces models that predict the residuals (errors) of the previous models to correct their predictions. Rating** 👍👍👍👍👍
- **In both the overall performance is weighed by the performance of models in an ensemble.**

Ensembles in RapidMiner

Some of the most popular ensembles (such as collections of trees) have a default structure, thus are offered as a single operator, e.g. *Random Forest* or *Gradient Boosted Trees*.
Others need a custom definition of their structure.



DEAKIN BUSINESS SCHOOL

Appendix

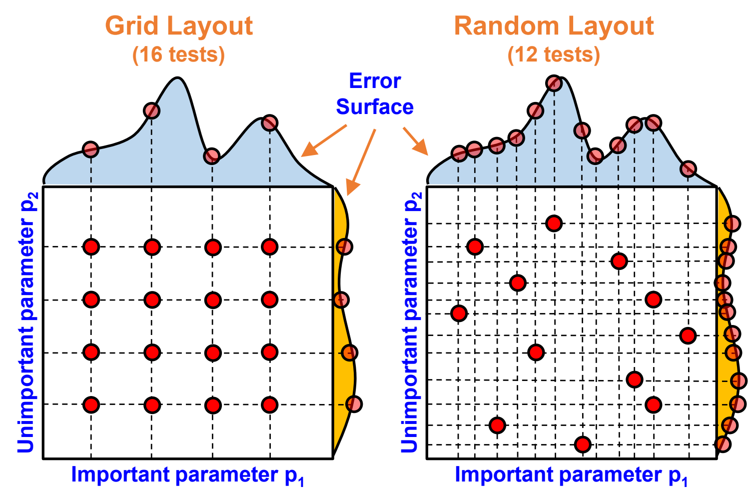
Tuning your instrument

Systematic exploration of options in search of optima

Single Parameter Model Optimisation

- The model performance depends on how well we select the model parameters for the data, e.g.
  - k-NN performance depends on the number of neighbours
  - Decision tree performance depends on several parameters, such as splitting criterion, tree maximum depth or pruning
- We need to *tune the model* by experimenting with its parameters to maximise performance.
- Trial and error is not the best way. A systematic approach is preferred.
- A simple approach is to construct a *loop over a list of values* for a single parameter and then log, chart and review the resulting performance indicators.
- RapidMiner also provides support (via its operators) for the a more thorough simultaneous exploration of multiple model parameters via *grid search*.

- There are two possible ways of exploring multiple parameter values, i.e. with:
  - *Grid search* of parameter values, where for each parameter we supply a list of its possible values and we test the model on all their combinations;
  - *Random grid search*, where test points are generated randomly, each having a combination of (most likely) unique parameter values.



The collection of parameters may include the more important parameters, which may be better at identifying distinguishing features of the error surface than unimportant parameters.

Also see KD 15.6 on Optimisation tools

DEAKIN BUSINESS SCHOOL

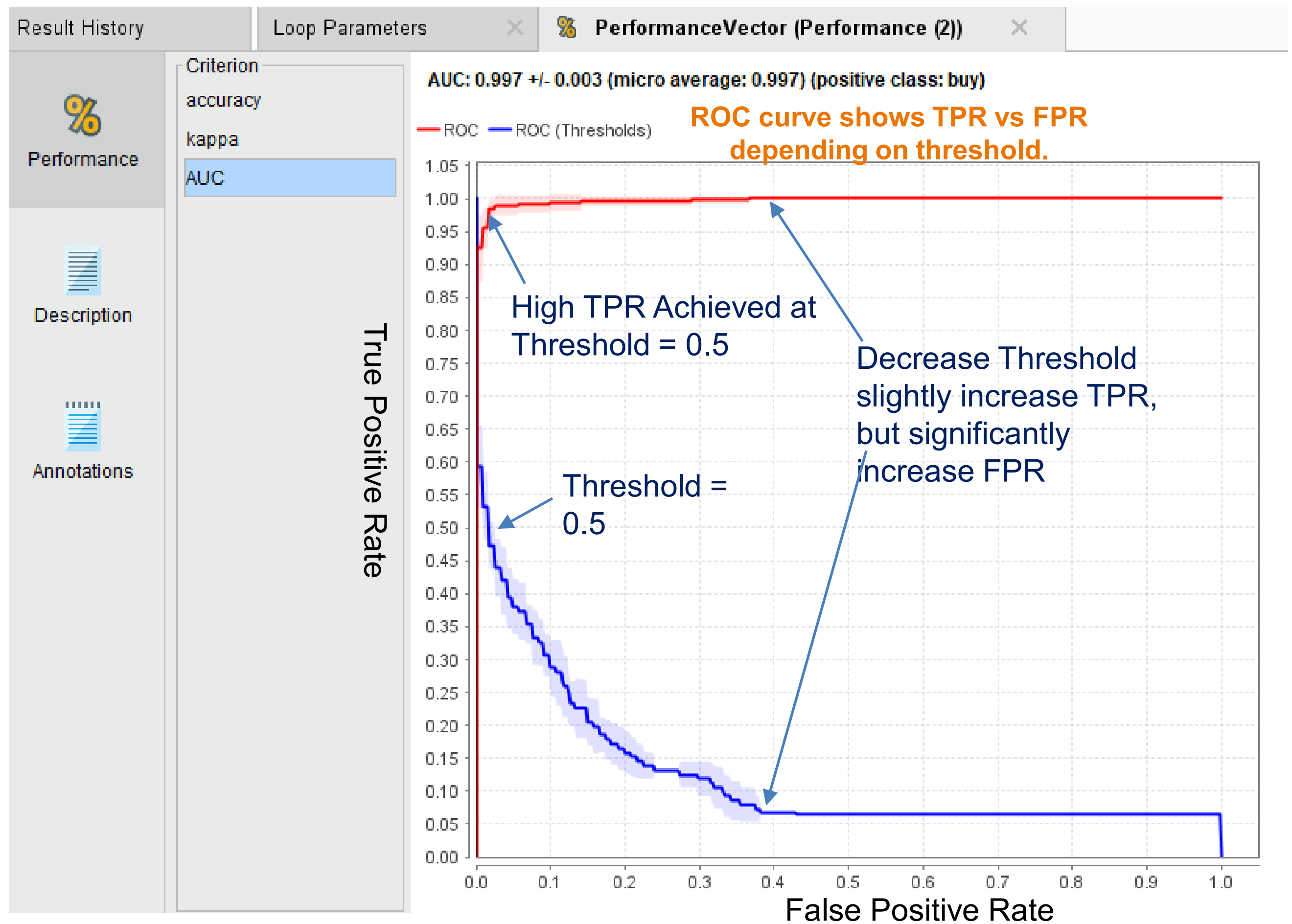
On the threshold

Nothing is done until it is done!

ROC and Area under Curve (AUC)

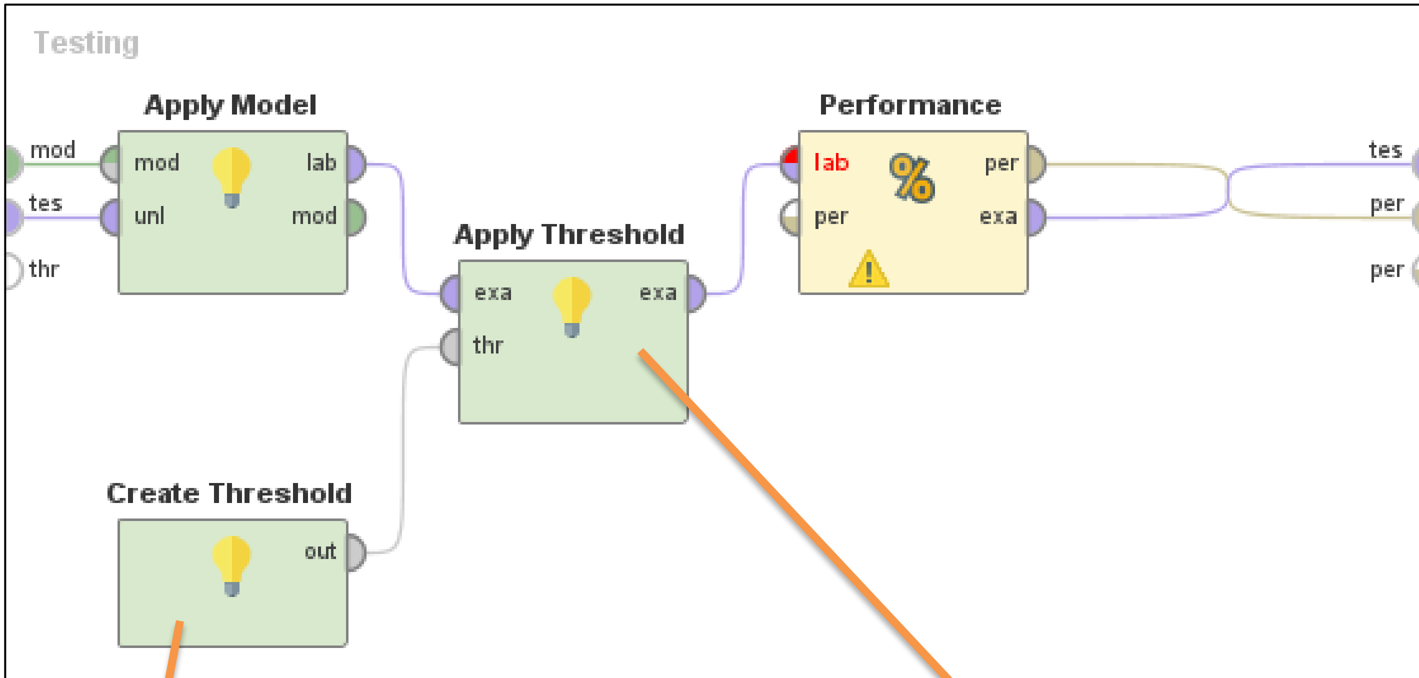
- shows the performance of a binomial classifier at several different class probability thresholds
- visualizes model performances with respect to sensitivity and specificity at the same time
- provides the possibility of finding the best model parameters, especially in the cases of imbalanced data sets

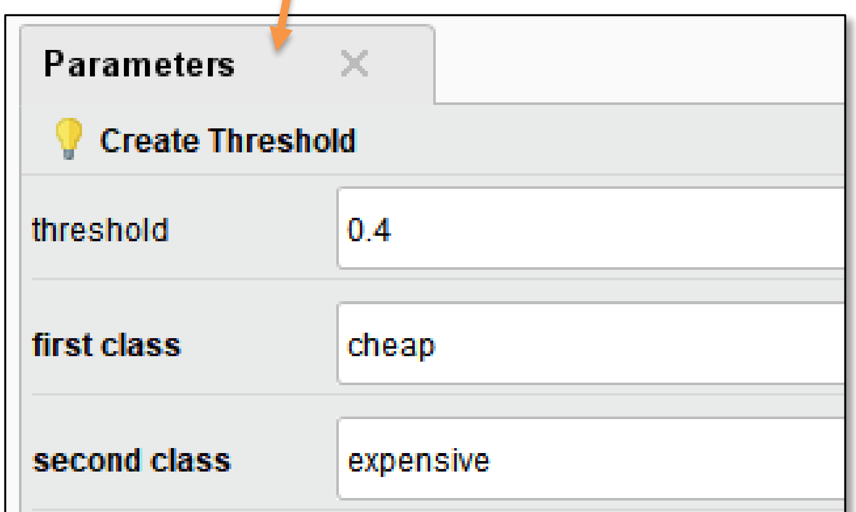
Performance (Binominal Classification)



- The area under any given ROC curve (AUC) summarizes the performance of the classifier model into a single measure

Setting thresholds for classification in RapidMiner (note this is done on the testing data)





"If the confidence for the second class is greater than the given threshold the prediction is set to this class otherwise it is set to the other class."

Summary / Review

- What is variance and bias?
  When is model under/over-fitting data?
  What is model under/over-training?
- Describe k-fold validation.
  Explain bootstrapping validation.
  Explain LOOCV.
- What are the problems with accuracy?
  Explain accuracy vs kappa?
- What is sample balancing?
  Why are we balancing samples?
  Explain minority vs majority class?
  Describe and compare up-sampling and down-sampling?
  What are approaches to sample balancing?
- How can we optimise k-NN?
  How can DT be optimised?
- What is a grid optimisation?
  How is it different from random grid?
  Which one is better?
  Which is supported by RapidMiner?
- What types of model ensembles do you know?
- How can we compare the performance of different ensemble models?
- What precautions need to be taken when comparing performance of ensembles?
- Can a model performance be improved after the model has already been trained?
- What is ROC? What is AUC? Describe.
- What is the role of threshold in classification? How can it be used?

DEAKIN BUSINESS SCHOOL