Science and Technology Facilities Council

Introduction to machine learning II

Associated notebook: https://github.com/keeeto/reading-ml-chemistry/blob/master/01_classification_decision_tree.ipynb

Overview Today(ish)

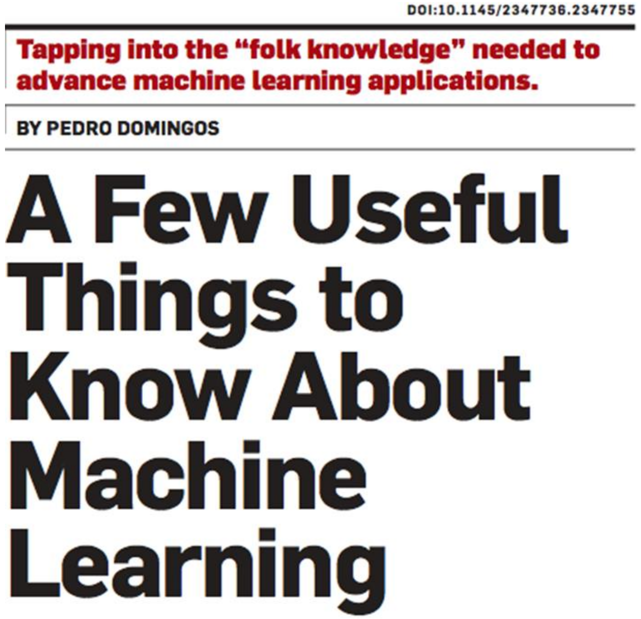
- Decision trees
- Optimisation
- Evaluation
- Overfitting
- Cross-validation
- Ensemble models

Setting up a notebook

- You will need a Google account to do this
- Go to https://colab.research.google.com/
- Search for https://github.com/keeeto/reading-ml-chemistry

A working definition

ML = Representation + Evaluation + Optimisation



DOI:10.1145/2347736.2347755

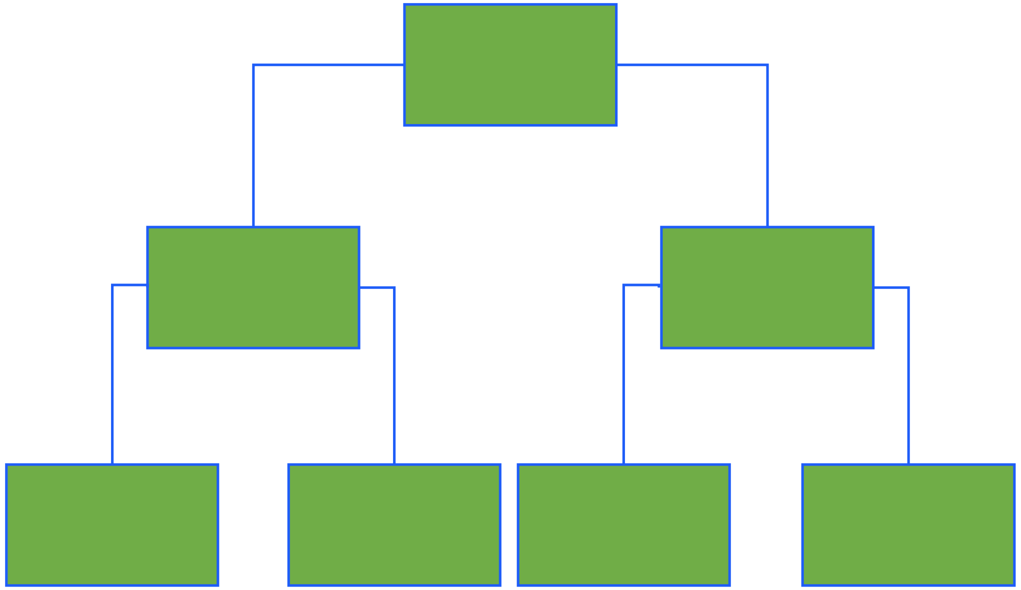
Tapping into the “folk knowledge” needed to advance machine learning applications.

BY PEDRO DOMINGOS

A Few Useful Things to Know About Machine Learning

"A Few Useful Things to Know About Machine Learning" by Pedro Domingos

Decision trees



Data is split by features. E.g. brightness of a pixel
Splits are arranged such that the data splits as evenly as possible at each point.

Decision trees

$$Q_{left}(\theta) = (x, y) | x_f \leq t_j$$
$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta)$$

Data is split according to a threshold value tj.

$$C(Q, \theta) = \frac{n_{left}}{N_j} H(Q_{left}(\theta)) + \frac{n_{right}}{N_j} H(Q_{right}(\theta))$$

The cost of the split is calculated based on some impurity function H() e.g. RMSD of the data.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} C(Q, \theta)$$

The splitting parameters are chosen to minimise C at each split.

Science and
Technology
Facilities Council

Go to notebook

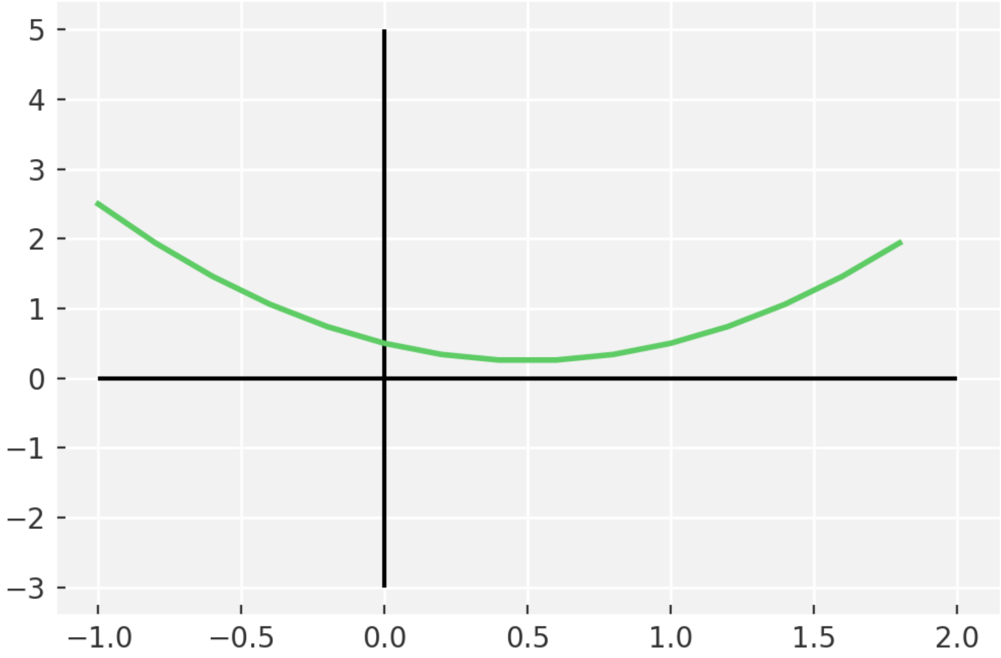
Optimisation/Evaluation

- Evaluation
  - Objective function or scoring function.
  - Distinguish good from bad models.
- Objective function = loss function = cost function
  - Must faithfully represent the “goodness” of a model in a single number

Evaluation metrics

- Mean squared error
- Used in regression
- Square endures a single minimum
- Avoids local minima trapping
- Easy to calculate

$$MSE = \frac{1}{N}\sum(f_i - y_i)^2$$

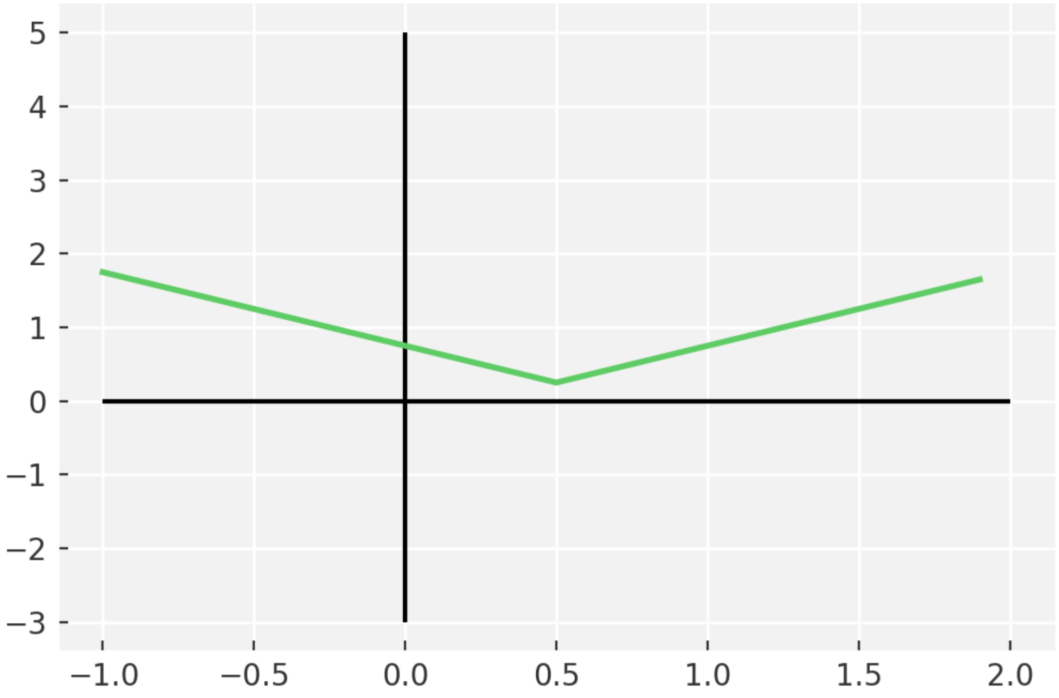


Science and Technology Facilities Council

Evaluation

- Mean Absolute Error
  - Similar to MSE
  - No quadric term
  - More robust to outliers
  - MSE penalises large differences much more than MAE
  - Large gradients close to zero - slow to optimise

$$MAE = \frac{1}{N}\sum |f_i - y_i|$$

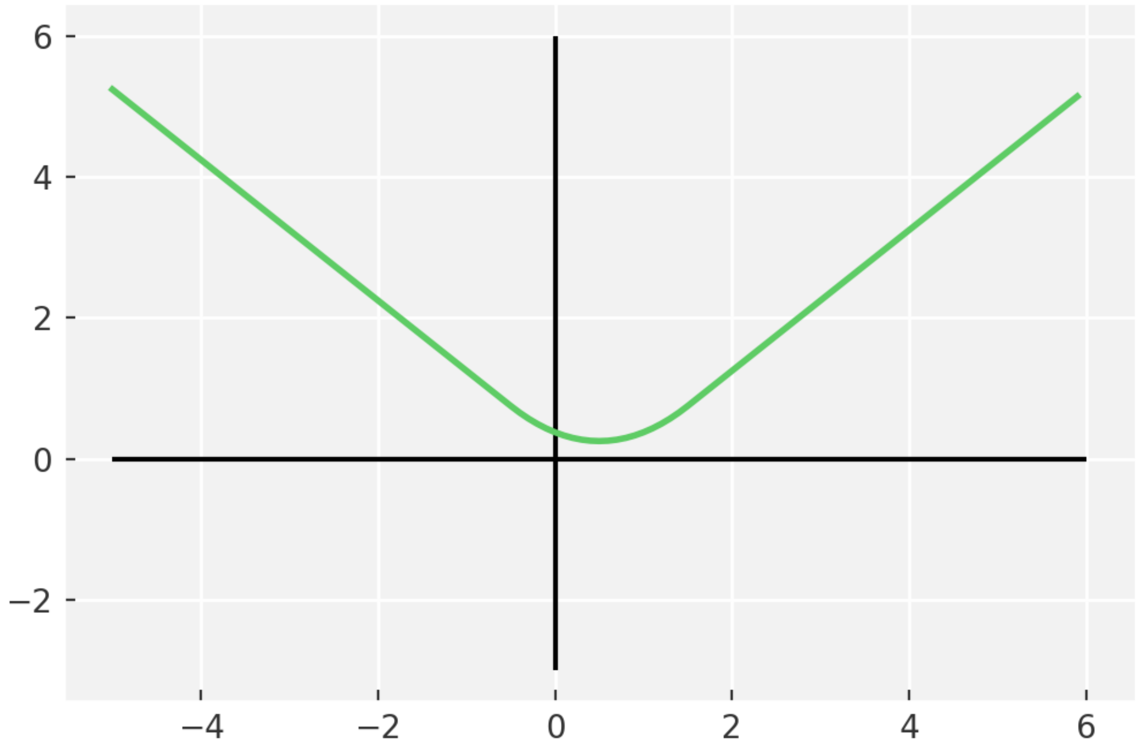


Science and Technology Facilities Council

Evaluation

- Huber loss
- Quadratic close to the minimum
- Linear far from the minimum
- Overcomes problems of MSE and MAE
- More expensive to calculate

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



Evaluation

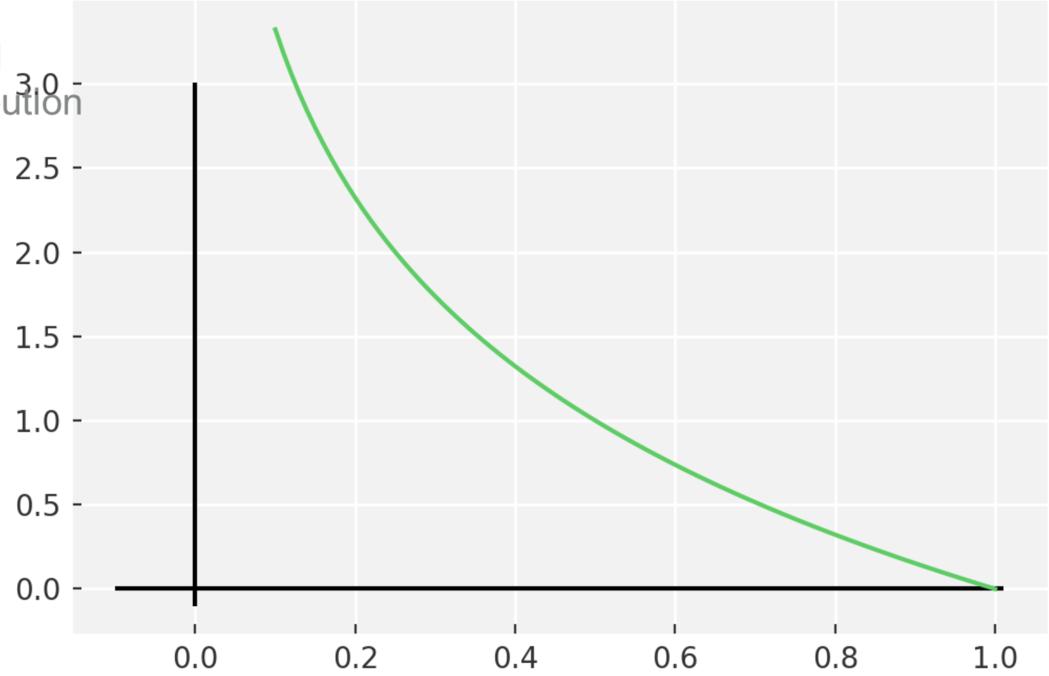
- Cross entropy
- Used for classification problems
- Tells us how similar our model distribution is to the true distribution
- Penalises all errors, but especially those that are most inaccurate

0	0	1	0
0.15	0.25	0.5	0.1

True distribution

$$H(p, q) = -\sum p_i \log_2(q_i)$$

Model distribution

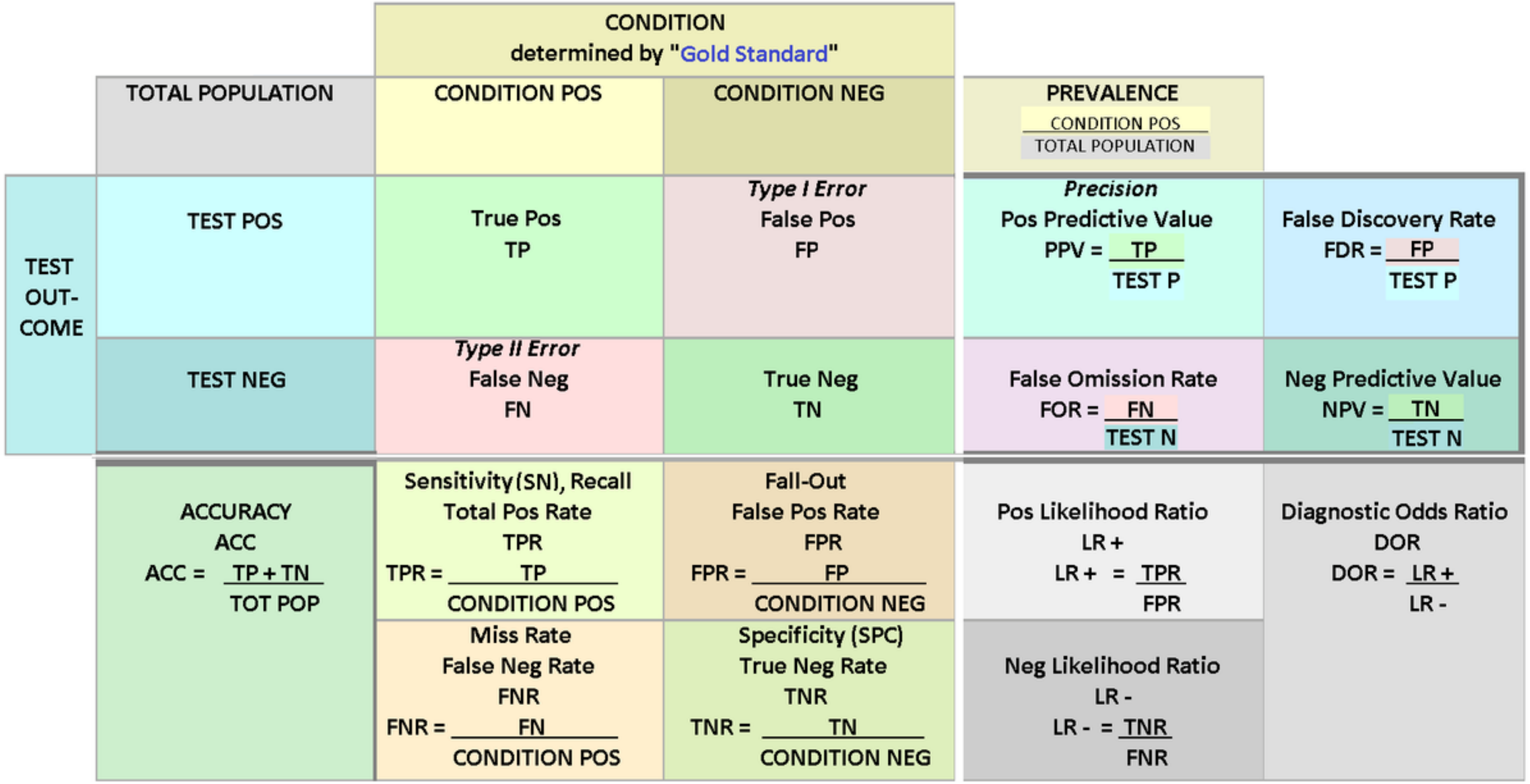


Evaluation

- Hinge loss
- Used for classification
- Does not seek to reproduce the distribution of data
- 0 as long as the classification is correct

$$L = max(0, 1 - \overset{\substack{\text{Label(+/-1)} \\ \downarrow}}{t} \cdot \underset{\substack{\uparrow \\ \text{Prediction}}}{y})$$

Evaluation: Table of confusion



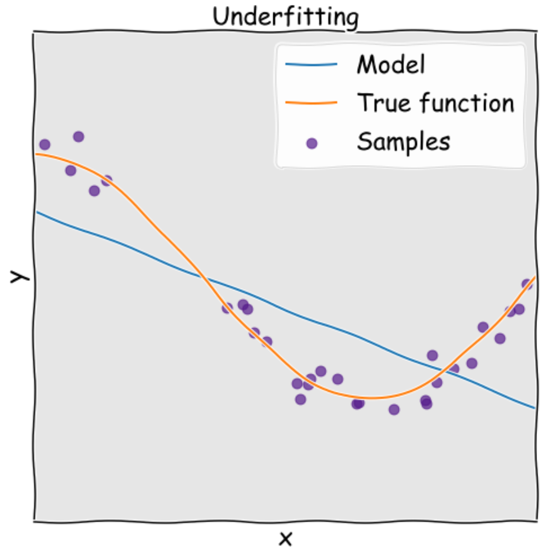
		CONDITION determined by "Gold Standard"			
	TOTAL POPULATION	CONDITION POS	CONDITION NEG	PREVALENCE $\frac{\text{CONDITION POS}}{\text{TOTAL POPULATION}}$	
TEST OUT-COME	TEST POS	True Pos TP	*Type I Error* False Pos FP	*Precision* Pos Predictive Value PPV = $\frac{\text{TP}}{\text{TEST P}}$	False Discovery Rate FDR = $\frac{\text{FP}}{\text{TEST P}}$
	TEST NEG	*Type II Error* False Neg FN	True Neg TN	False Omission Rate FOR = $\frac{\text{FN}}{\text{TEST N}}$	Neg Predictive Value NPV = $\frac{\text{TN}}{\text{TEST N}}$
	ACCURACY ACC ACC = $\frac{\text{TP + TN}}{\text{TOT POP}}$	Sensitivity (SN), Recall Total Pos Rate TPR TPR = $\frac{\text{TP}}{\text{CONDITION POS}}$	Fall-Out False Pos Rate FPR FPR = $\frac{\text{FP}}{\text{CONDITION NEG}}$	Pos Likelihood Ratio LR + LR + = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic Odds Ratio DOR DOR = $\frac{\text{LR +}}{\text{LR -}}$
		Miss Rate False Neg Rate FNR FNR = $\frac{\text{FN}}{\text{CONDITION POS}}$	Specificity (SPC) True Neg Rate TNR TNR = $\frac{\text{TN}}{\text{CONDITION NEG}}$	Neg Likelihood Ratio LR - LR - = $\frac{\text{TNR}}{\text{FNR}}$	

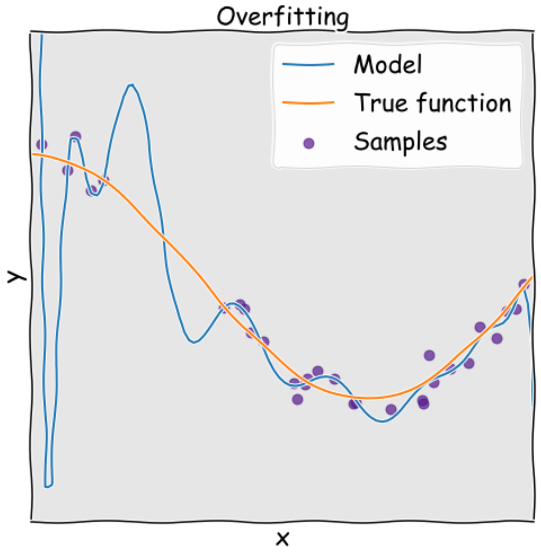
Science and Technology Facilities Council

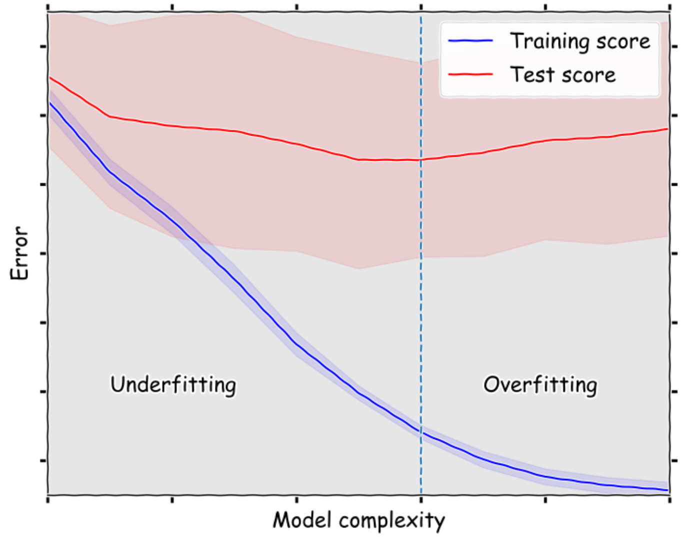
https://en.wikipedia.org/wiki/Confusion_matrix

Evaluation

- Over/under fitting







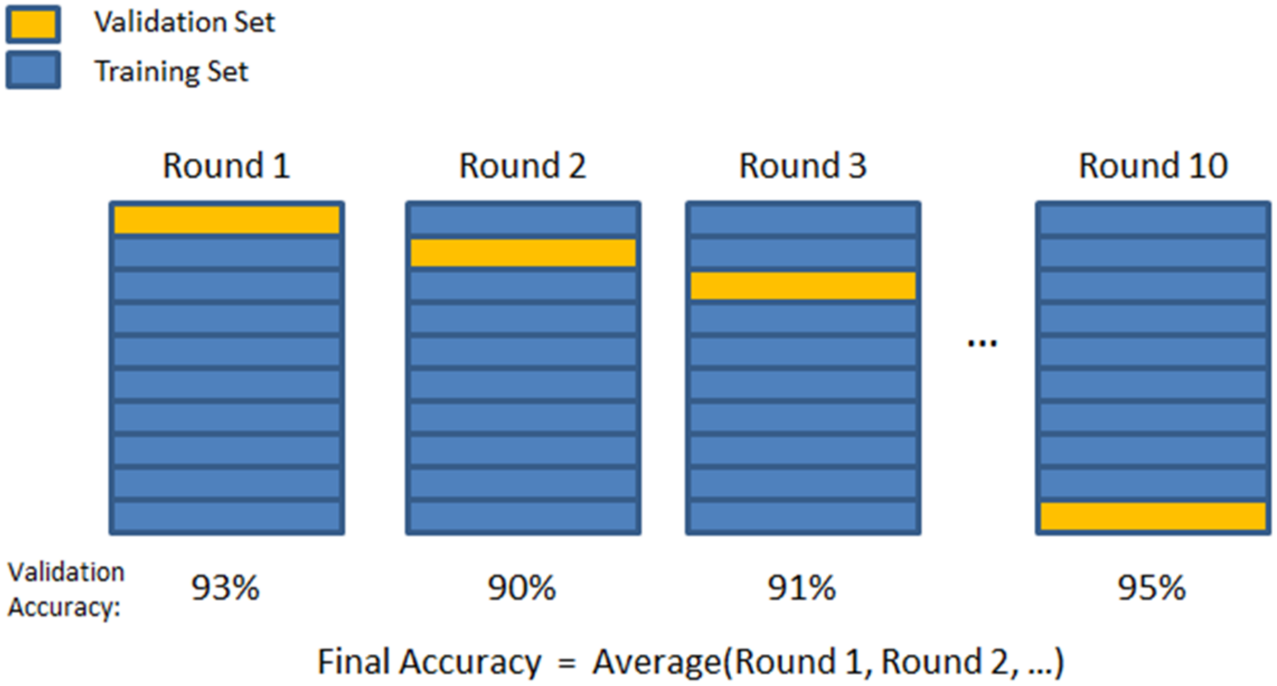
Science and
Technology
Facilities Council

Test and validation sets

- The model must always be validated on data not used for testing
- Often something like 20% of data is used for validation
- Make sure that validation and training distributions are the same

Evaluation

- n-fold cross validation
  - Ensure training/test splits



Building block Cross Validation

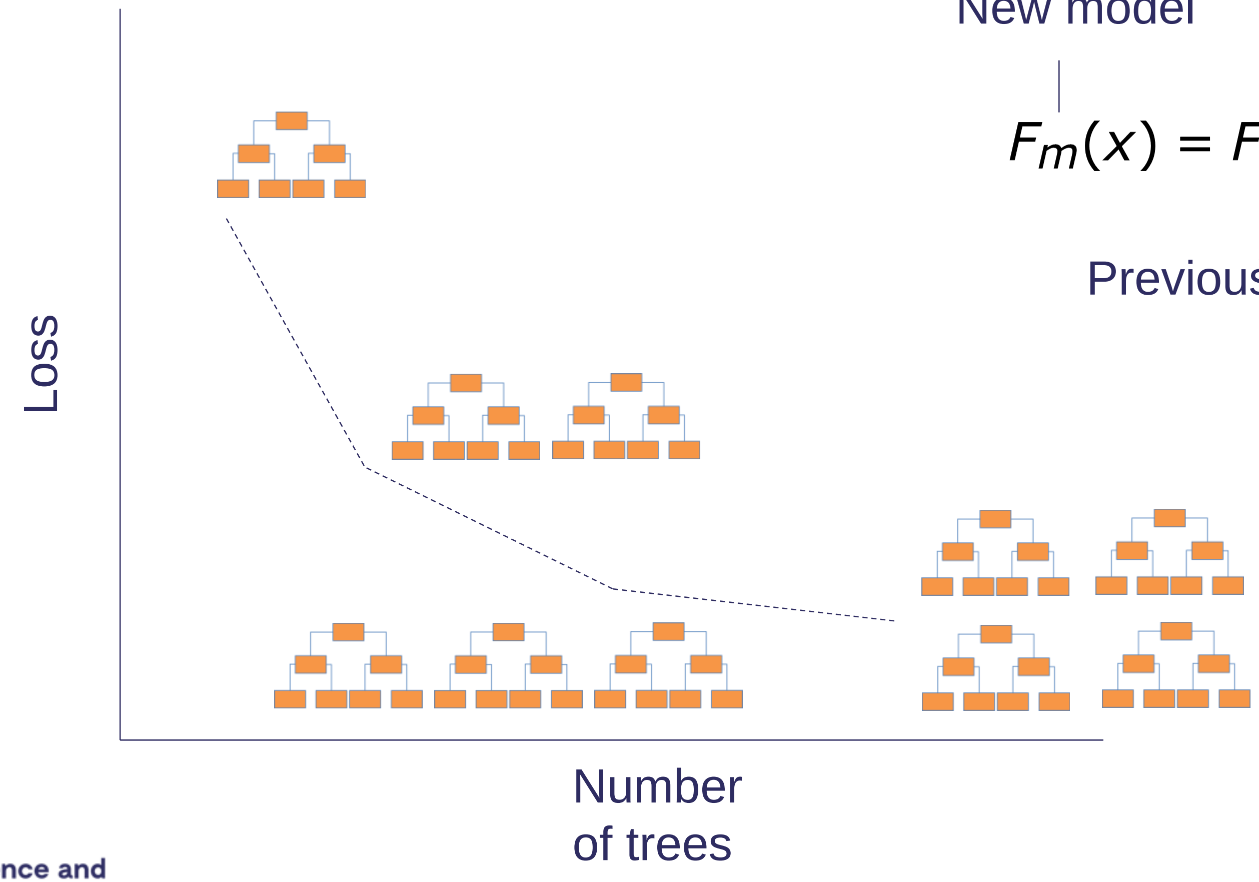
```
from sklearn.model_selection import
cross_val_score
clf = svm.SVC(kernel='linear', C=1)
scores = cross_val_score(clf, X, y, cv=5)
```

Go To Notebook

Boosting + Bagging

- To overcome the limitations of a weak learner we can use booting or bagging.
- Both methods use an ensemble of weak learners to build a strong learner
- Boosting – choose next learner based on the errors of the last learner (gradient boosted decision trees)
- Bagging – stochastically choose next learners (random forests)

Boosted Decision Trees



New model

Learning rate

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

Previous model

Sub model of errors

Building block: Boosted decision tree

```
from sklearn import ensemble

gbr = ensemble.GradientBoostingRegressor(loss='lad', max_depth
= 10, learning_rate = 0.015, min_samples_split = 50,
min_samples_leaf = 1, max_features = len(cols), subsample =
0.9, n_estimators = 300)

gbr.fit(X, y)
```

Go To Notebook

Concept checklist

- Supervised/unsupervised machine learning
- Classical machine learning/deep learning
- Parameters/hyperparameters
- Features and feature engineering
- Decision trees
- Overfitting
- Evaluation/metrics
- Test/train split, cross-validation
- Bagging and boosting

Science and
Technology
Facilities Council

Thank you

Facebook: Science and Technology Facilities Council

Twitter:@STFC_matters

YouTube: Science and Technology Facilities Council