Performance Evaluation

BORUT SLUBAN

DATA MINING AND KNOWLEDGE DISCOVERY

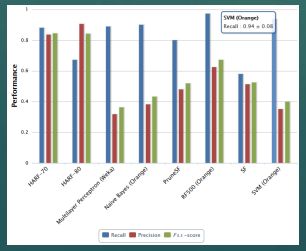
Road map

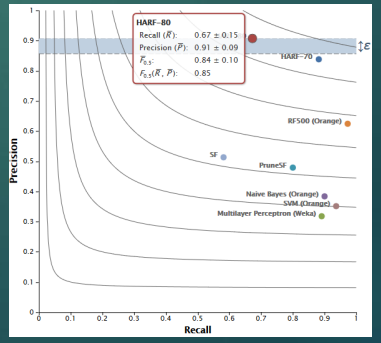
- The data mining problem

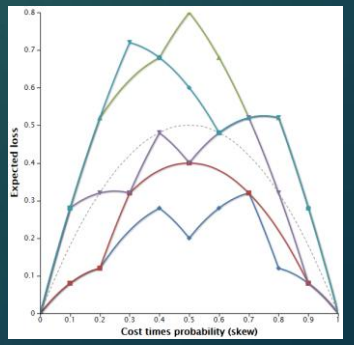
- Got my results ☺ … now what?
- How to measure performance?

age	prescription	astigmatic	tear_rate	lenses [class]
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	soft

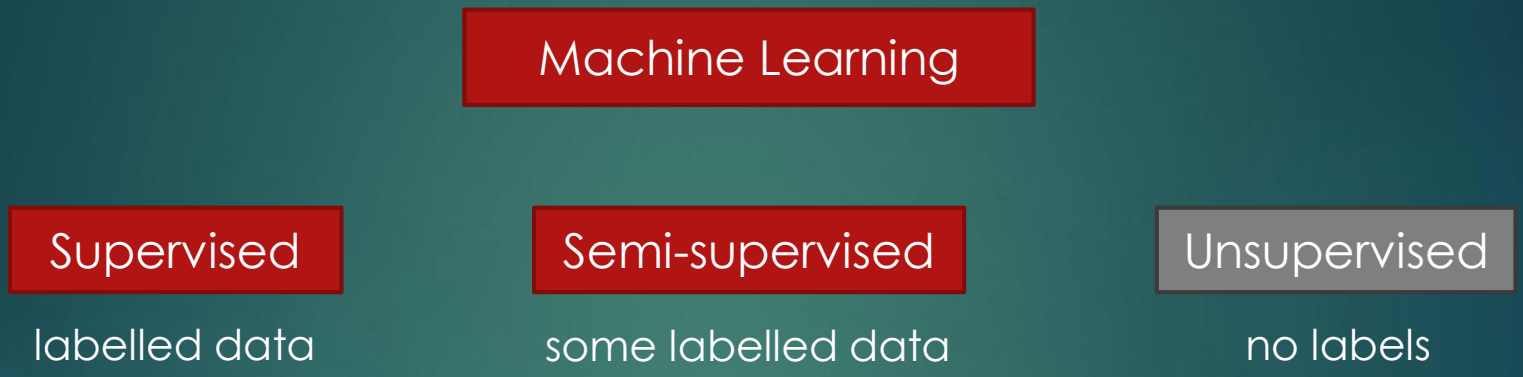
- Visualize performance results







Solving data mining tasks

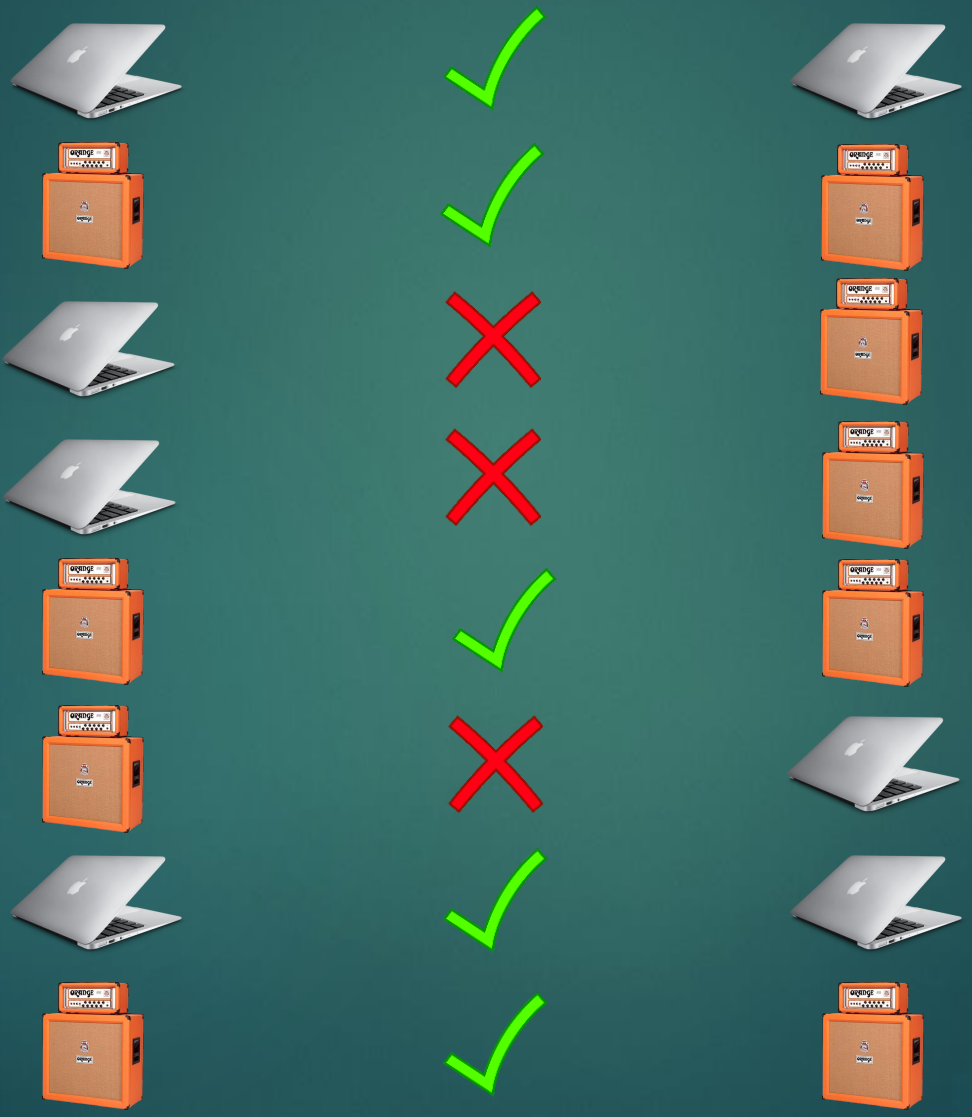


- Train data → Model / Classifier
- Test data → Predictions
  - Labelled data (cross-validation, leave-one-out, etc.)
  - Unlabelled data

How to Measure Performance?

- ***Qualitative*** Evaluation
  - Domain Expert
  - Very demanding (time, cost)
  - Limited in size
- ***Quantitative*** Evaluation
  - Requires known data labels
  - Different performance measures (precision, recall, *F*-measure, …)
  - No size limitations

True/Actual vs Predicted



Confusion Matrix

Predicted

Actual	Positive	Negative
Positive (P)	TP	FN
Negative(N)	FP	TN

$$\text{Accuracy} = \frac{TP+TN}{P+N}$$

$$\text{Precison} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$F = 2\frac{Precision*Recall}{Precision+Recall}$$

Predicted

Actual	Apples	Oranges
Apples	2	2
Oranges	1	3

$$\text{Accuracy} = \frac{5}{8} = 0.625$$

$$\text{Precison} = \frac{2}{3} = 0.67$$

$$\text{Recall} = \frac{2}{4} = 0.5$$

$$F = 0.57$$

$$\text{Precison} = \frac{3}{5} = 0.6$$

$$\text{Recall} = \frac{3}{4} = 0.75$$

$$F = 0.67$$

Scores and Ranks as Predictions


1.0


0.4


0.8


0.6

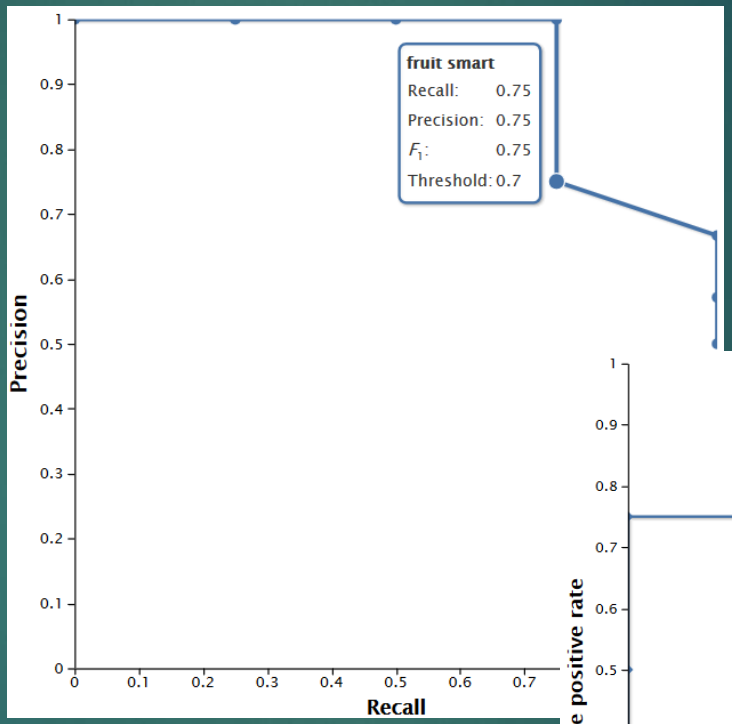

0.7


0.6

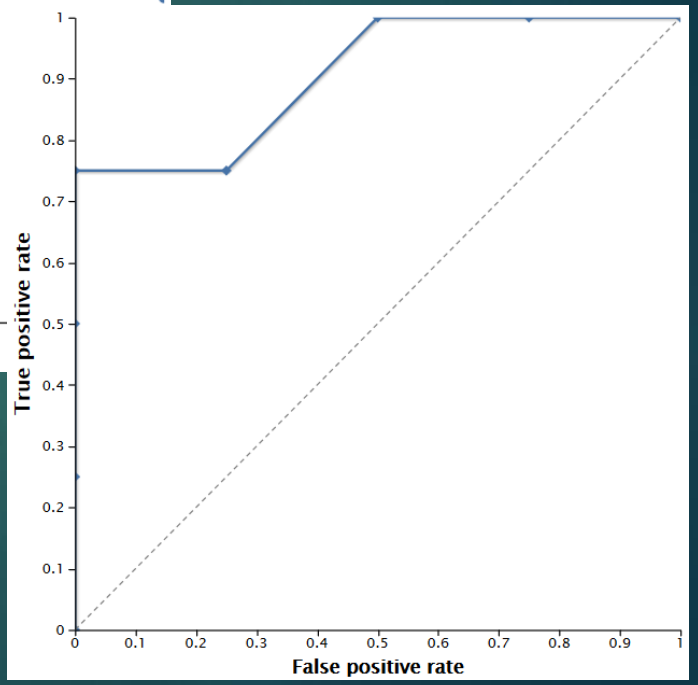

0.9


0.5

Precision-Recall Curve

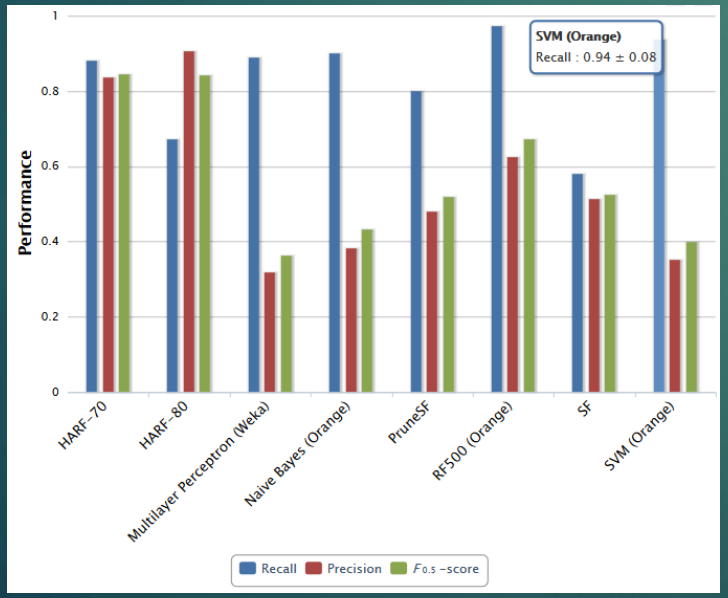


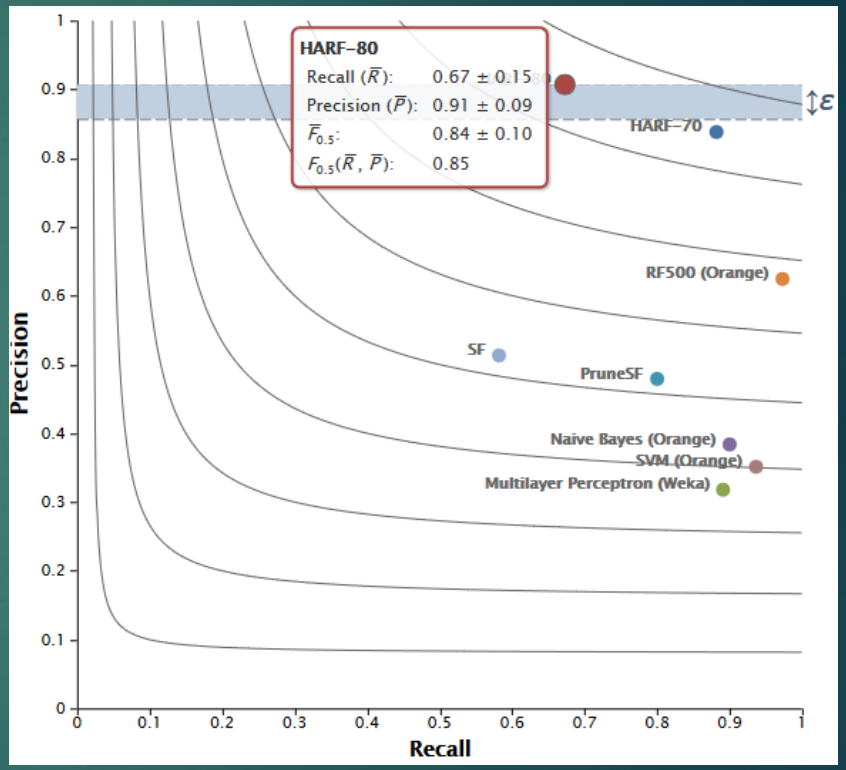
ROC Curve



Presenting results …

Algorithm	Recall	Precision	$F_{0.5}$ score
HARF-70	0.88 ± 0.10	0.84 ± 0.05	0.84 ± 0.05
HARF-80	0.67 ± 0.15	0.91 ± 0.09	0.84 ± 0.10
Multilayer Perceptron (Weka)	0.89 ± 0.08	0.32 ± 0.04	0.36 ± 0.04
Naive Bayes (Orange)	0.90 ± 0.06	0.38 ± 0.05	0.43 ± 0.05
PruneSF	0.80 ± 0.13	0.48 ± 0.04	0.52 ± 0.05
RF500 (Orange)	0.97 ± 0.06	0.62 ± 0.05	0.67 ± 0.05
SF	0.58 ± 0.17	0.51 ± 0.11	0.52 ± 0.12
SVM (Orange)	0.94 ± 0.08	0.35 ± 0.05	0.40 ± 0.06





… using

- ViperChart Platform (http://viper.ijs.si/)
  - Scatter Chart (PR space, ROC space)
  - Curve Charts (Lift, ROC, Precision-Recall, Cost, etc.)
  - Column Charts
- OR
  - Access its API (http://viper.ijs.si/api/)
    from the program language of your choice
  - within Clowdflows (http://www.clowdflows.org/)
  - off-line version ViperChartsPackage (send email to borut.sluban@ijs.si)

Hands-on

- ViperCharts (http://viper.ijs.si/)
- ClowdFlows (http://www.clowdflows.org/)
  - Performance evaluation
    - http://clowdflows.org/workflow/5420/
  - Visual performance evaluation (ViperCharts)
    - http://clowdflows.org/workflow/5418/
- ViperChartsPackage (very BETA, for Python)
- Need help or advice: borut.sluban@ijs.si