Introduction to Classification

Daniel Lawson University of Bristol

Lecture 01.2 (v4.0.0)

Chihuahua or muffin?

Signposting

- Most of machine learning is modern statistics.
  - The main distinction is about purpose: whether developed for prediction, or for understanding.
- Today is about the framing of the problem.

Types of machine learning

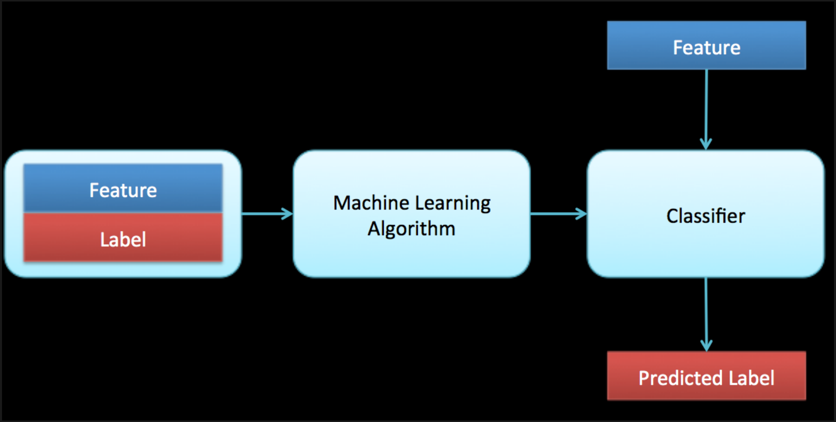
- Unsupervised: **no labels**. For example,
  - Clustering
  - Dimensionality reduction
  - Smoothing
- Supervised: exploits **labels**. For example,
  - Classification
  - Regression
- Other types:
  - Semi-supervised: **some labels** are available
  - Active: can **choose which labels** to obtain
  - Reinforcement: **reward based**. explore vs exploit?
  - etc.

Magic of Supervised Learning

- By **cross-validation**, splitting data into:
  - **Train**: What we learn parameters on,
  - **Calibration**: What we learn hyperparameters on,
  - **Test**: What we report performance on,
- Machine Learning can be (almost?) automated -
  - If you can define a suitable **label** for supervision,
  - And if you can find **test data** that look like your real problem.

Classification

- Machine Learning classification is about how to make good predictions of **classes** based on previous experience of how **features** relate to **classes**.

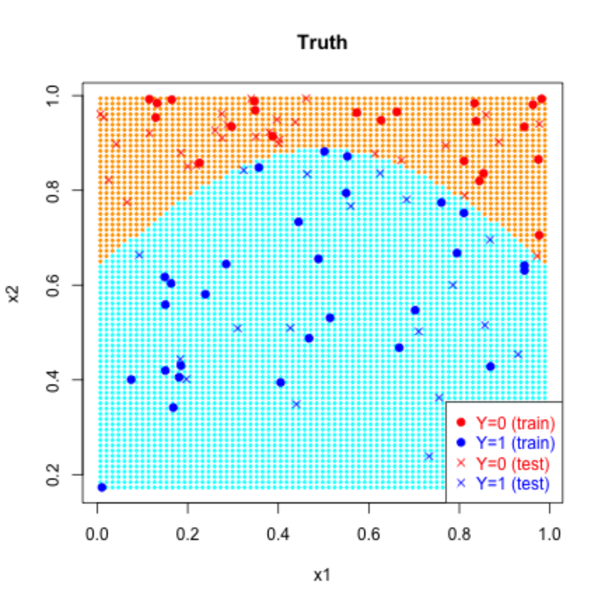


Examples of classification

Classification is broadly the "detection, recognition, recall of **prior experience**".

- **Spam** filtering (spam/not spam)
- **Face detection** (image classification)
- **Speech recognition**
- **Handwriting recognition**
- **Turing test** ... though that is human, not machine!
- **Next token** prediction for LLMs...

Classification



From Regression to Classification

- **Linear regression** is obtaining solutions to:

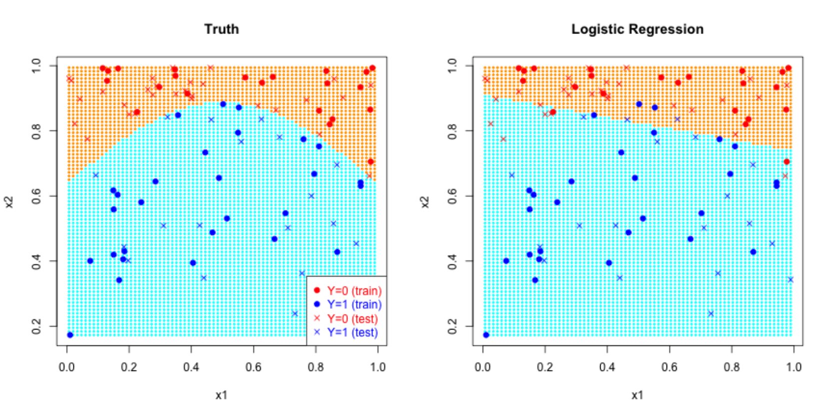
$$y_i = \vec{x}_i \cdot \beta + e_i$$

- Definitions: $\vec{x}_i$ *and* β *are vectors of length* p*;* e_i *are residuals whose squared-sum is minimised.*
- **Logistic regression** estimates the probability that a binary outcome y is 1:

$$\mathrm{logit}(p(y_i)) = \ln\left(\frac{p(y_i)}{1 - p(y_i)}\right) = \vec{x}_i \cdot \beta + e_i$$

- with $y_i \sim \mathrm{Bern}(p(y_i))$. The prediction is the **log-odds** ratio, with values > 0 predicting a 1 and values < 0 predicting a 0.

Logistic Regression example



Classification Performance (naive)

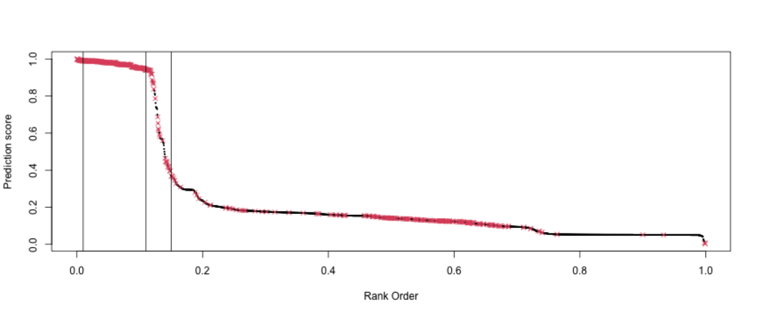
Consider a **best guess** classifier, i.e. we output a predicted class only.

- Evaluate performance on **test** data, to prevent over-fitting.
- **Accuracy** is the proportion of correctly classified points.
  - But... what is the **baseline**? If there are imbalanced classes?
- Classifier performance is understood through a **Confusion Matrix**, a comparison between:
  - Ground truth class, and
  - Predicted classes.
- For binary classes, summarise using (true/false)(positive/negative) outcomes.

Classification Performance

- Most predictors are continuous scores, allowing **ranking**
  - e.g. Logistic regression log-odds
  - But... are these **probabilities?**
- Scores are **ordered**. So we can choose a threshold to control the total proportion of **positive predictions**.
  - This provides a **relationship** between **Positive Claims** and **True Positives**.
  - Multiclass classification contains nuance

Classification Performance Thresholding

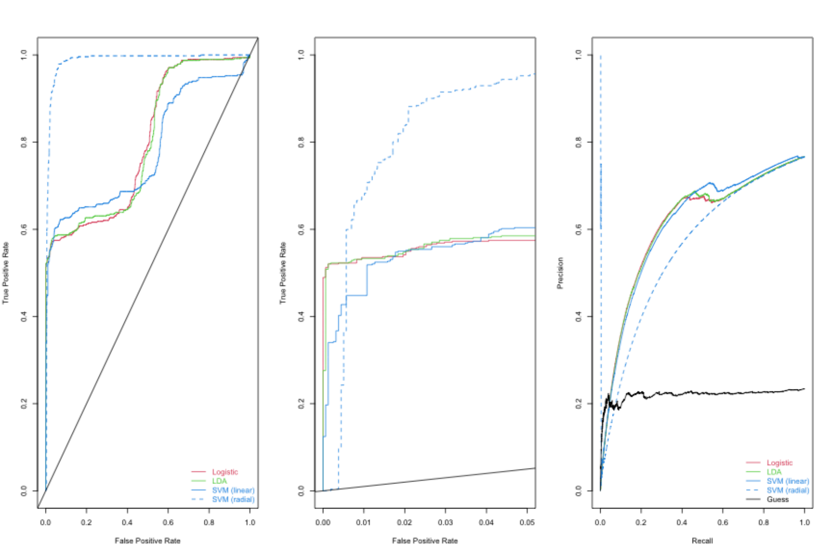


	$Y = 1$	$Y = 0$	Condition
$\hat{Y} = 1$	TP	FP	Prediction positive
$\hat{Y} = 0$	FN	TN	Prediction negative
Claim	Truth positive	Truth negative	

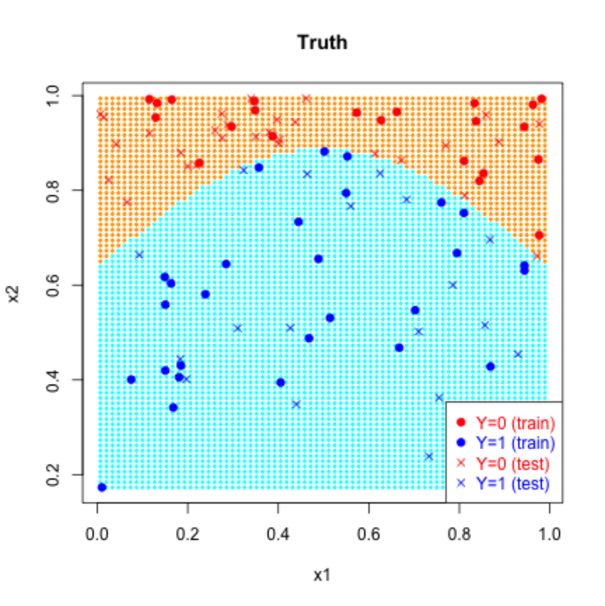
Classification Performance Representations

- There are many ways to represent performance
- The **Receiver-Operator-Curve (ROC)** is the most popular, as it holds regardless of the true distribution of the data.
  - X-axis: False Positive Rate (FPR) $= P(\hat{Y} = 1|Y = 0)$
  - Y-axis: True Positive Rate (TPR) $= P(\hat{Y} = 1|Y = 1)$
  - The **Area Under the Curve (AUC)** is a measure of Accuracy (0.5=guessing, 1=perfect).
  - We need to care about the region of the ROC curve that matters.
- The **Precision-Recall curve** is appropriate when we care specifically about positive cases:
  - X-axis: Precision $= P(Y = 1|\hat{Y} = 1)$
  - Y-axis: Recall=TPR $= P(\hat{Y} = 1|Y = 1)$

ROC/PR Curve Example



Classification



K-Nearest Neighbour classification

- We later introduce **K-NN**.
  - Data in $\mathcal{R}^d$
  - Make a choice of distance function, e.g. Euclidean

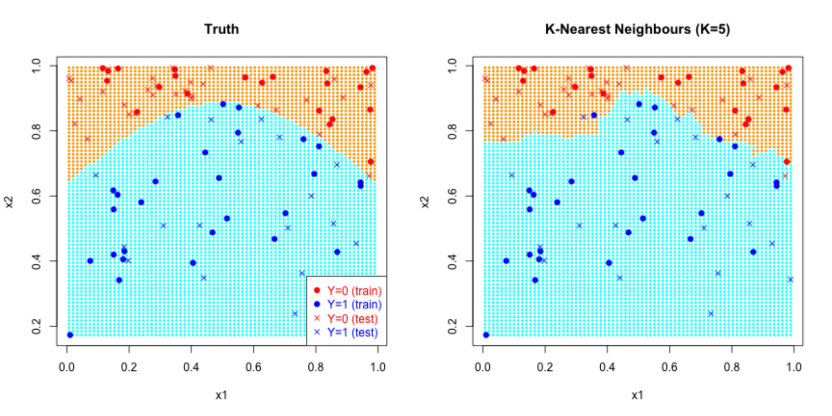
$$d(x, y) = ||x - y||_2$$

  - Obtained the K nearest neighbours of points
- Classifier: **majority vote** of the training labels of all k neighbours.
  - But... How do you choose k?
- A naive implementation scales poorly with N, but an approximate lookup can control complexity.
  - See also: Condensed nearest neighbor[1] approaches to reduce the amount of data required at the classification stage.

---

[1] Hart P, The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory 18 (1968) 515-516. doi: 10.1109/TIT.1968.1054155

K-Nearest Neighbour example



Wrapup

- These are **baseline models**
- **Logistic regression** is the go-to straw man classifier in machine learning:
  - It is a natural predictive model
  - It does surprisingly well in many settings
- **k-NN** is the interpolation method to beat
  - **Neighbourhoods** are always competitive, but are costly at test time
  - How should we measure "close"?
- The magic is in the **sampling**:
  - train, calibration, **test**

Portfolio:

- Portfolio Question 1.2.1: What metrics are better than accuracy? In what sense are they reliable, and do they relate to uncertainty?

References:

- **References** for classification basics:
  - Stack Exchange Discussion of ROC vs PR curves.
  - Davis and Goadrich, "The Relationship Between Precision-Recall and ROC Curves", ICML 2006.
  - Rob Schapire's ML Classification features a Batman Example...
  - Chapter 4 of The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Friedman, Hastie and Tibshirani).
- k-Nearest Neighbours:
  - Chapter 13.3 of The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Friedman, Hastie and Tibshirani).
- Linear Discriminant Analysis:
  - Sebastian Raschka's PCA vs LDA article with Python Examples
  - Chapter 4.3 of The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Friedman, Hastie and Tibshirani).
- SVMs:
  - Jason Weston's SVMs tutorial
  - e1071 Package for SVMs in R
  - Chapter 12 of The Elements of Statistical Learning: Data