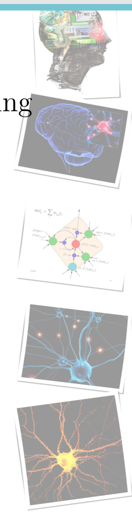
A quick introduction to machine learning

Spyros Samothrakis
Senior Lecturer, IADS
University of Essex
MiSoC

June 22, 2022



Welcome/course contents

- What will this course cover?
  - Day 1: An intro to machine learning (ML)
  - Day 1: ML labs
  - Day 2: An intro to causal inference
  - Day 2: ML and causal inference labs
- Textbooks?
  - Mitchell, T. M. (1997). Machine learning.[1]
  - Bishop, C. M. (2006). Pattern recognition and machine learning. springer.[2]
  - Wasserman, L. (2013). All of statistics: a concise course in statistical inference. Springer Science & Business Media.[3]

[1] `http://www.cs.cmu.edu/~tom/mlbook.html`
[2] `https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/`
[3] `http://www.stat.cmu.edu/~larry/all-of-statistics/index.html`

Better science through data

Hey, Tony, Stewart Tansley, and Kristin M. Tolle. "Jim Gray on eScience: a transformed scientific method." (2009).[4]

- Thousand years ago: empirical branch
  - You observed stuff and you wrote down about it
- Last few hundred years: theoretical branch
  - Equations of gravity, equations of electromagnetism
- Last few decades: computational branch
  - Modelling at the micro level, observing at the macro level
- Today: data exploration
  - Let machines create models using vast amounts of data

[4] `http://languagelog.ldc.upenn.edu/myl/JimGrayOnE-Science.pdf`

Better business through data

- There was a report by Mckinsey

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.[5]

- Urges everyone to monetise "Big Data"
- Use the data provided within your organisation to gain insights
- Has some numbers as to how much this is worth
- Proposes a number of methods, most of them associated with machine learning and databases

[5] `http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation`

Why is it popular now?

- **Algorithms + data + tools**
- Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical science, 16(3), 199-231.[6]
- Anderson, P. W. (1972). More is different. Science, 177(4047), 393-396.[7]
- Pedregosa, et.al. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.[8]

[6] `http://projecteuclid.org/download/pdf_1/euclid.ss/1009213726%20`
[7] `https://www.tkm.kit.edu/downloads/TKM1_2011_more_is_different_PWA.pdf`
[8] `https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf`

So this course covers tools

- ML theory
  - *Supervised learning Regression Classification*
  - Understanding basic modelling
  - Confirming your model is sane
  - Tuning your model
  - **All within a very applied setting**
- Tools
  - Numpy
  - Scikit-learn

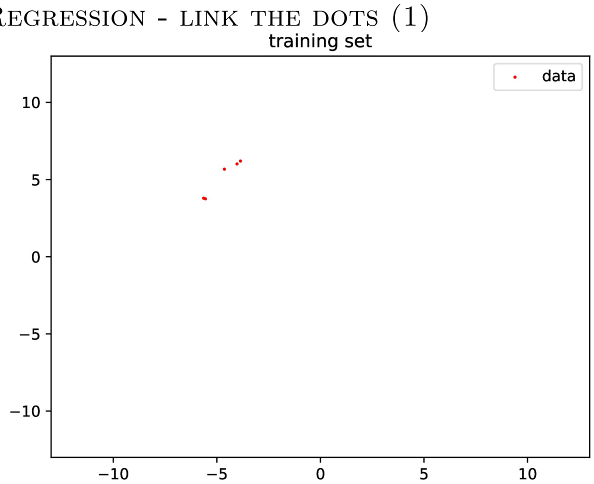
What is supervised learning?

- Imagine someone gives you a group of smokers
  - And asks the question – what is their life expectancy?
- **Completely made up imaginary data**

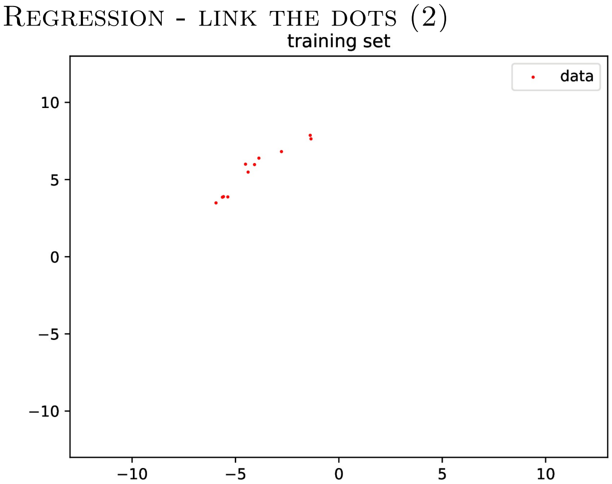
Some abstraction

- We are given inputs $x_0, x_1 ... x_n$ and we are looking to predict y
- Let's plot!

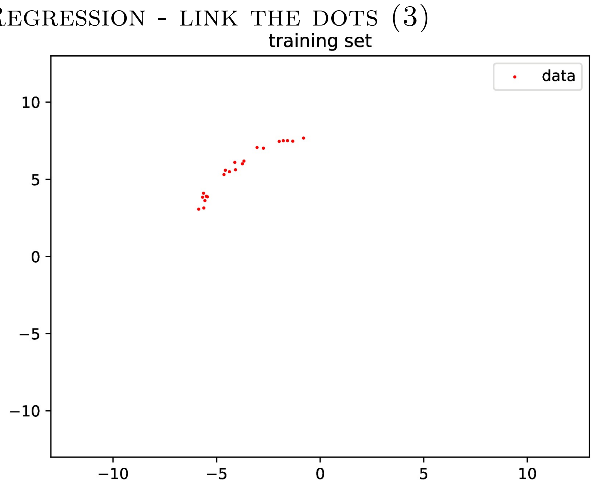
Regression - link the dots (1)



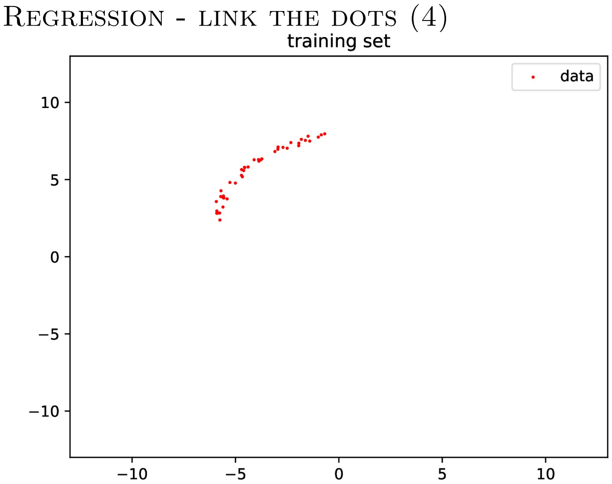
Regression - link the dots (2)



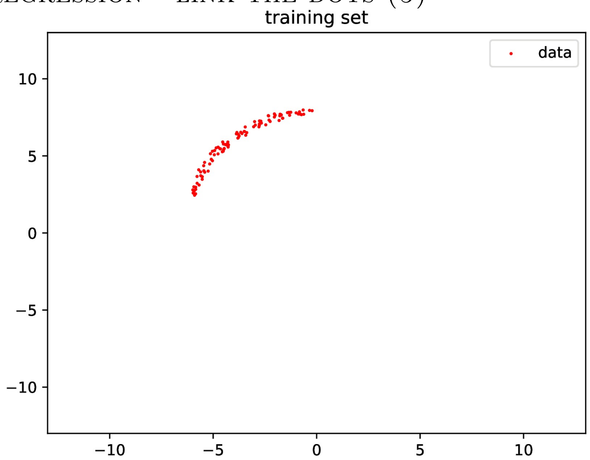
Regression - link the dots (3)



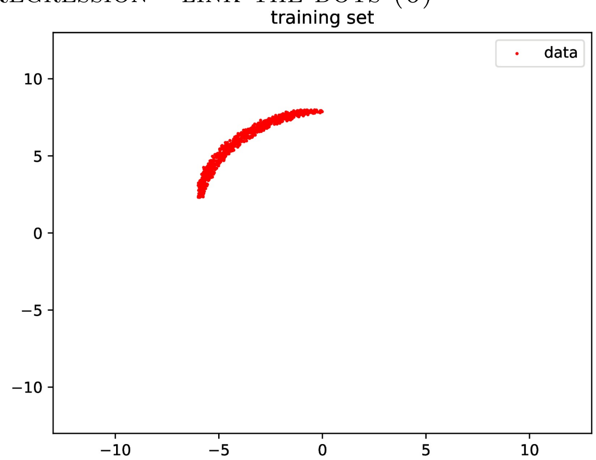
Regression - link the dots (4)



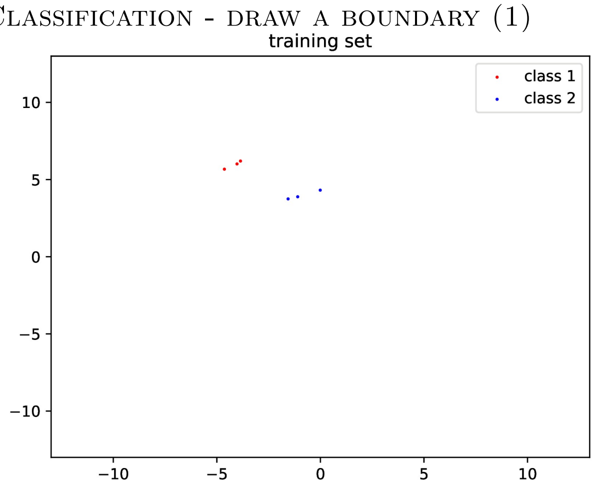
Regression - link the dots (5)



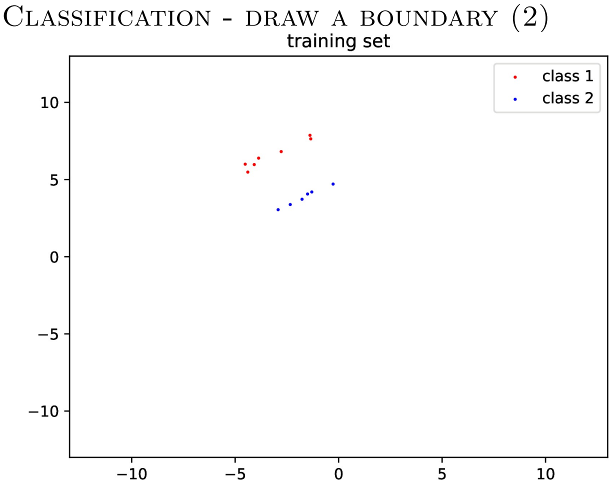
Regression - link the dots (6)



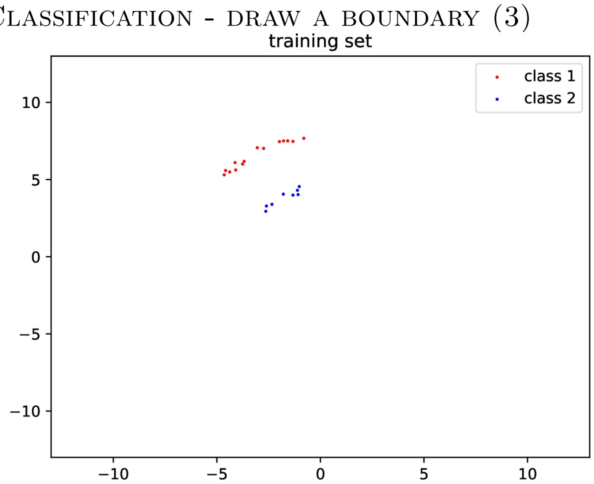
Classification - draw a boundary (1)



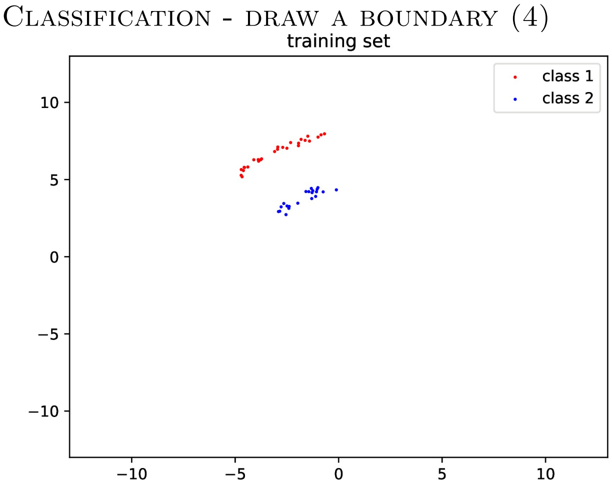
Classification - draw a boundary (2)



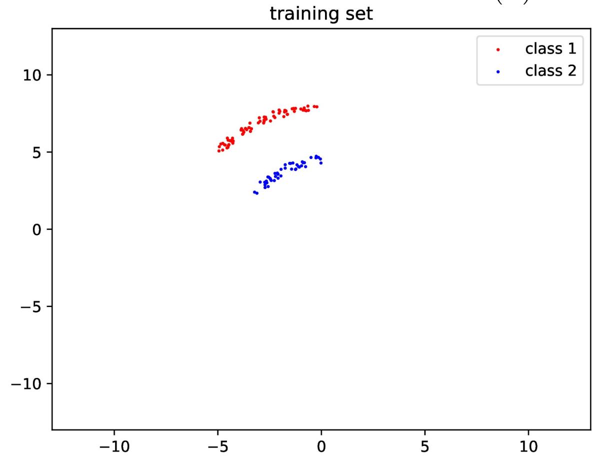
Classification - draw a boundary (3)



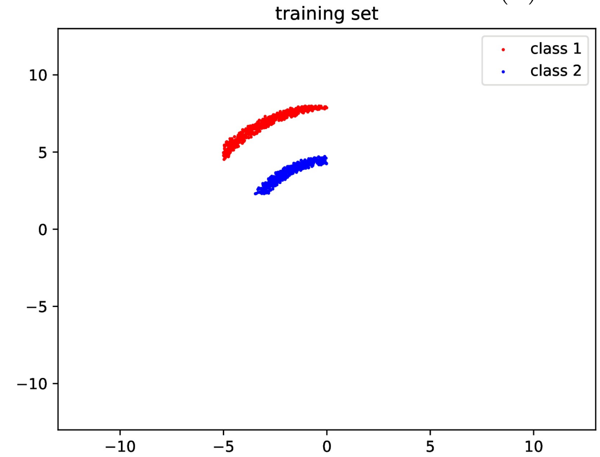
Classification - draw a boundary (4)



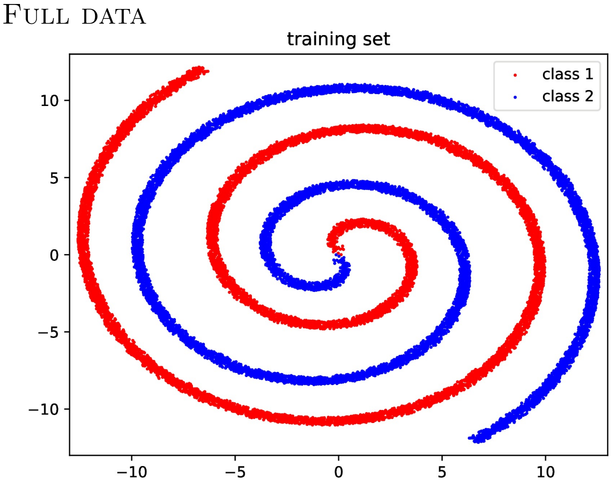
Classification - draw a boundary (5)



Classification - draw a boundary (6)

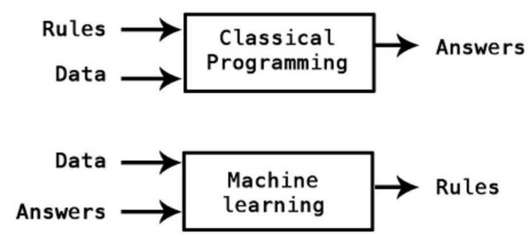


Full data



Intuition (1)

Chollet, F. (2018). Deep learning with Python (Vol. 361). New York: Manning.[9]



[9] `https://www.manning.com/books/deep-learning-with-python`

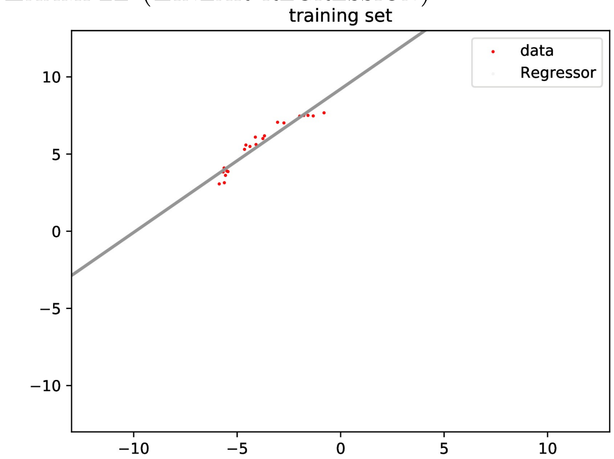
Intuition (2)

- That's it - we are given data, and we need to come up with an algorithm to join it up – but in high dimensions
  - Can can be binary, categorical, real-valued - more on this later
- How well well a function joins the data is called the "loss"
- Multiple solutions exist, so loss function must take into account concepts other than pure fit

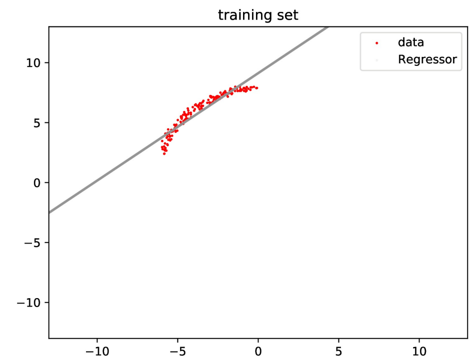
Linear regression

- Linear and logistic regression
  - Logistic regression does classification
- You just assume everything is a line
- $f(x) = wx + b$

Example (Linear regression)

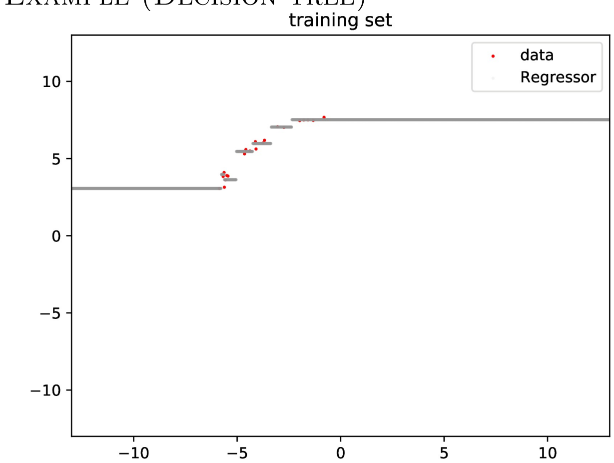


Example (Linear regression)

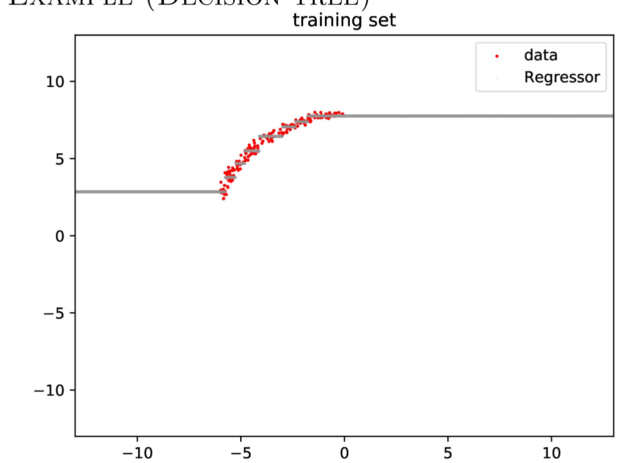


$f(x) = 0.9x + 9.1$

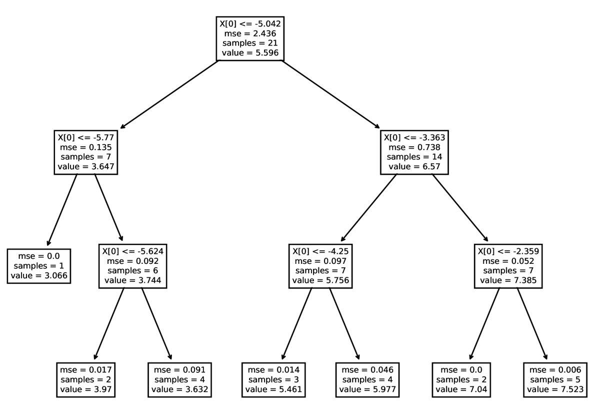
Example (Decision tree)



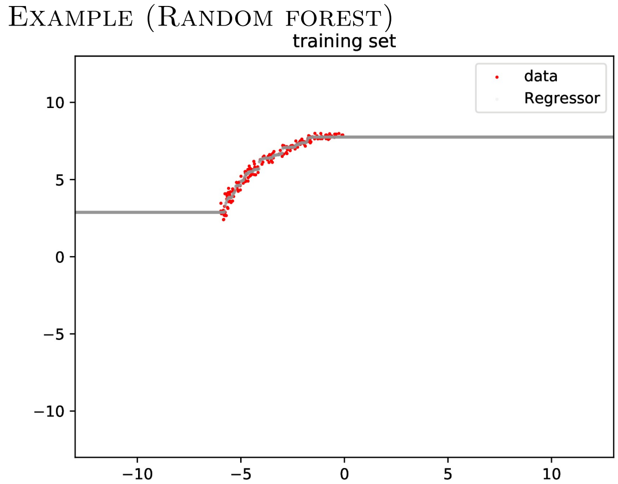
Example (Decision tree)



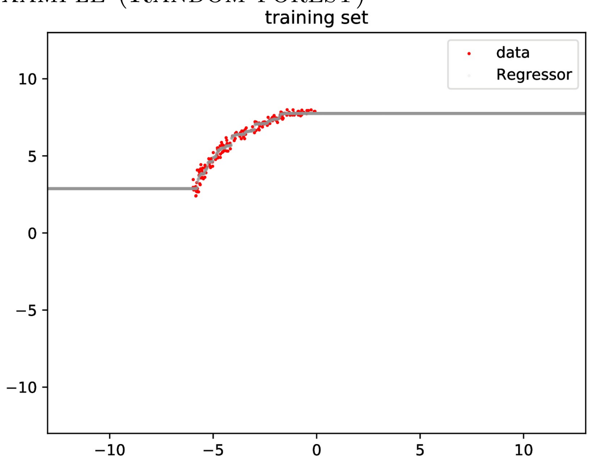
Example (Decision tree — internal)



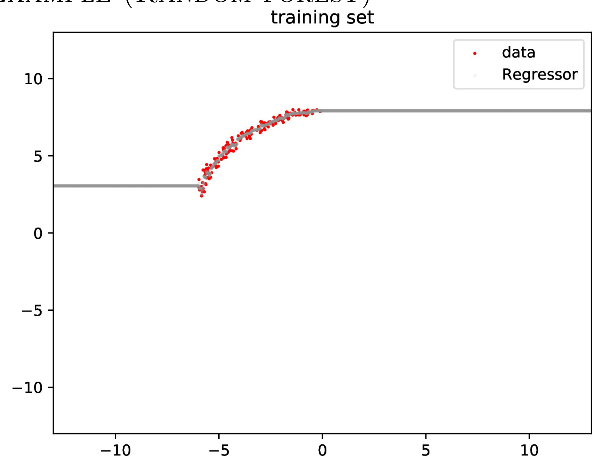
Example (Random forest)



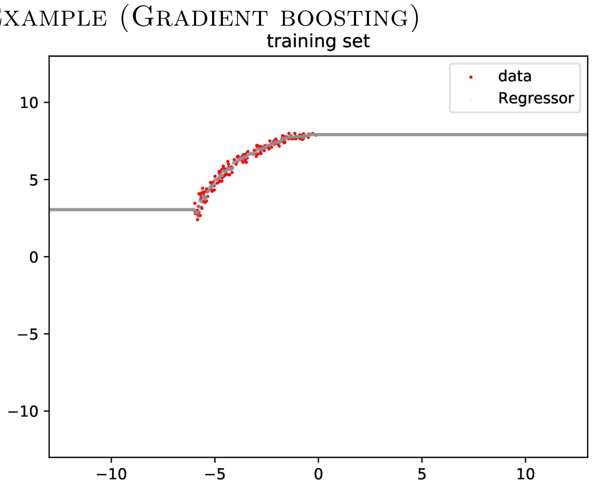
Example (Random forest)



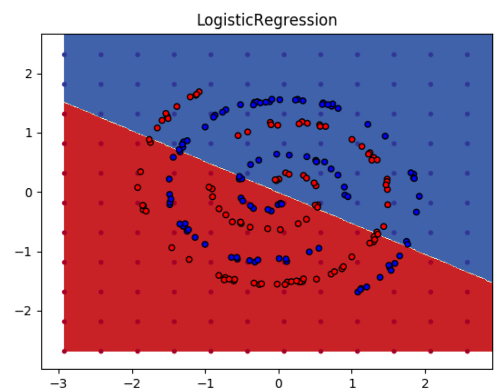
Example (Random forest)



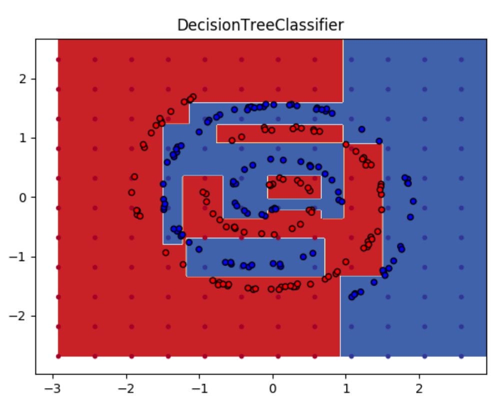
Example (Gradient boosting)



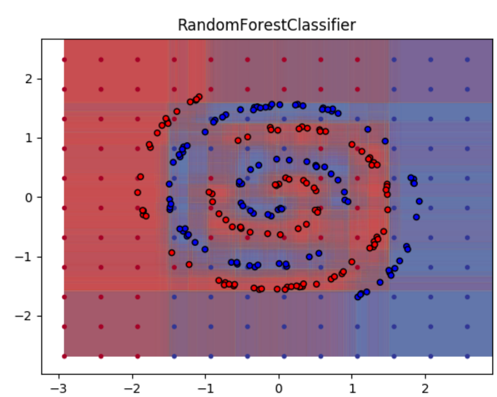
Classification (Logistic regression)



Classification (Decision trees)



Classification (Random forests)



Data dimensionality

- Until now we have seen input data of 1 (for regression) or two (for classification) dimensions
- How about higher dimensional data?
  - Some times data can have millions of features
- Let's examine more high dimensional dataset
- Visualisation becomes harder

Diabetes Classification

Feature	Description
X_0	Pregnancies: Number of times pregnant
X_1	Glucose: Plasma glucose concentration
X_2	BloodPressure: Diastolic blood pressure (mm Hg)
X_3	SkinThickness: Triceps skin fold thickness (mm)
X_4	Insulin: 2-Hour serum insulin (mu U/ml)
X_5	BMI: Body mass index (weight in kg/(height in m)^2)
X_6	DiabetesPedigreeFunction: Diabetes pedigree function
X_7	Age: Age (years)
y	Outcome: Has diabetes (0 or 1)

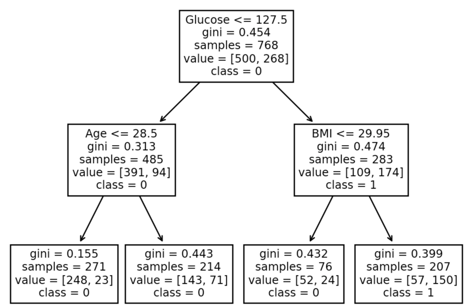
https://www.kaggle.com/mathchi/diabetes-data-set

How does the data look like?

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age
0	6	148	72	35	0	33.60	0.63	50
1	1	85	66	29	0	26.60	0.35	31
2	8	183	64	0	0	23.30	0.67	32
3	1	89	66	23	94	28.10	0.17	21
4	0	137	40	35	168	43.10	2.29	33

	y
0	1
1	0
2	1
3	0
4	1

Decision Tree



Diabetes regression

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. Annals of statistics, 32(2), 407-499.[10]

Feature	Description
X_0	age in years
X_1	sex
X_2	bmi body mass index
X_3	bp average blood pressure
X_4	s1 tc, total serum cholesterol
X_5	s2 ldl, low-density lipoproteins
X_6	s3 hdl, high-density lipoproteins
X_7	s4 tch, total cholesterol / HDL
X_8	s5 ltg, possibly log of serum triglycerides level
X_9	s6 glu, blood sugar level
y	disease progression one year after baseline

[10] https://scikit-learn.org/stable/datasets/toy_dataset.html#diabetes-dataset

Let's see the real data values

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
0	0.04	0.05	0.06	0.02	-0.04	-0.03	-0.04	-0.00	0.02	-0.02
1	-0.00	-0.04	-0.05	-0.03	-0.01	-0.02	0.07	-0.04	-0.07	-0.09
2	0.09	0.05	0.04	-0.01	-0.05	-0.03	-0.03	-0.00	0.00	-0.03
3	-0.09	-0.04	-0.01	-0.04	0.01	0.02	-0.04	0.03	0.02	-0.01
4	0.01	-0.04	-0.04	0.02	0.00	0.02	0.01	-0.00	-0.03	-0.05

"Note: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times n_samples (i.e. the sum of squares of each column totals 1)."

	y
0	151.00
1	75.00
2	141.00
3	206.00
4	135.00

Linear regression

$y = -210x_0 - 5036x_1 + 10916x_2 + 6812x_3 - 16635x_4 10011x_5 + 2121x_6 + 3718x_7 + 15776x8 + 1420x_9 + 152$

Plotting?

Quality assessment

- In lower dimensions, the visualisations we did provided some insights to the quality of our methods
  - This is impossible in higher dimensions
- We need to measure some kind of metric that denotes quality of fit

Metrics

- For regression,
  - Mean Squared Error
  - Mean Absolute Error
- For classification
  - Accuracy
  - Mean Squared Error
  - Cross-entropy loss
  - AUC
- Each one has different benefits, e.g. absolute errors tend to be more robust to outliers

Accuracy

- Each row is now assigned to a class of $y_i \in 0..20$
- Accuracy is the obvious one
  - $accuracy = \frac{1}{N} \sum_{i=0}^{N-1} y_i = \hat{f}(x)$
  - The higher the accuracy the better
- What if the dataset is unbalanced - how informative is accuracy then?
- There are multiple metric functions
  - Use the one appropriate for your problem

Mean Squared Error (MSE)

- Reality is $f(x)$
- Our model is $\hat{f}(x)$ (e.g. a decision tree)
- Sample from the model are $\{y_0 ... y_N\}$
  - $MSE = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \hat{f}(x_i)\right)^2$
- For every possible sample
  - $E\left[\left(y - \hat{f}(x)\right)^2\right]$

Train/validation/test split

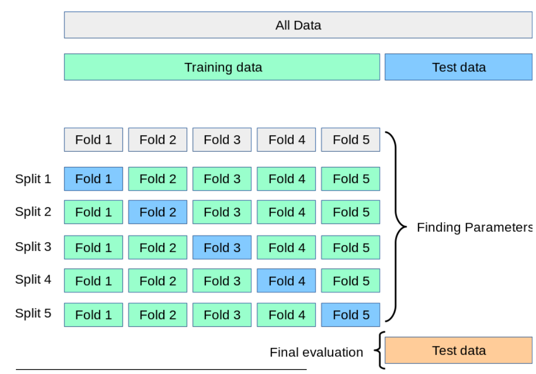
- Basic idea: split your data into three portions
- 1. train, you used that to train your classifier/regressor
- 2. validation, you use that to assess the quality of your method, retraining as you see fit
- 3. test, you report results on this
- Common split is 60%/20%/20%

Cross validation

- How about we split our data into multiple validation sets and find the mean?
- Instead of having just one split train/test split, we can have multiple
- Colloquially goes by names like 5-fold CV, 10-fold CV
- There are multiple ways of doing the sampling to create training/validation sets, we will focus on only one

Pictorial depiction of 5-fold CV

Copied from SKlearns website[11]



[11] `https://scikit-learn.org/stable/_images/grid_search_cross_validation.png`

Why tune?

Hyperparameters

- Called hyperparameters (vs parameters) as they influence how the modelling is done (vs the direct modeling)
  - How many trees?
  - Tree depth?
  - Maximum tree size
  - l2 regularisation?
- vs parameters (e.g. weights in linear regression)

We need to look for optimal parameters

- Computationally expensive
- We can do this either by searching both the classifier/regressor space and their parameters
- Grid search
  - More than one parameter, we exhaustively search

Example using linear regression

alpha	scores	mean	std
0.0001	[2782, 3032, 3226, 3003, 2917]	2992.1772	145.5645
0.0001	[2783, 3032, 3223, 3002, 2920]	2992.0154	143.9139
0.0002	[2785, 3032, 3218, 3001, 2923]	2991.8400	141.7267
0.0007	[2812, 3042, 3186, 3002, 2945]	2997.5634	122.1458
0.0009	[2818, 3042, 3179, 2992, 2946]	2995.3784	117.9862
0.0012	[2827, 3043, 3178, 2978, 2947]	2994.6426	115.5067
0.0037	[2884, 3060, 3190, 2895, 2968]	2999.3816	114.1540
0.0049	[2918, 3079, 3201, 2869, 2985]	3010.3321	118.4097
0.0065	[2938, 3111, 3215, 2856, 3017]	3027.3294	126.2295
0.0085	[2966, 3152, 3219, 2859, 3057]	3050.5713	128.2733
0.0113	[3014, 3212, 3236, 2872, 3113]	3089.2555	134.1712
0.0149	[3028, 3292, 3279, 2918, 3201]	3143.7112	146.9126
0.0196	[3040, 3366, 3358, 2970, 3289]	3204.6848	166.7447
0.0259	[3082, 3493, 3484, 3074, 3435]	3313.4750	193.2530
0.0342	[3206, 3706, 3681, 3237, 3678]	3501.7398	229.0676
0.0452	[3434, 4030, 3972, 3448, 4037]	3784.1217	281.4318
0.0597	[3801, 4573, 4447, 3745, 4545]	4222.0278	369.6680
0.0788	[4401, 5460, 5212, 4299, 5425]	4959.4742	505.7819
0.1040	[5211, 6521, 6262, 5200, 6486]	5935.8770	603.2078
0.1374	[5353, 6521, 6262, 5290, 6486]	5982.4134	547.2524

Wrapping up

- You get data from somewhere
- ML will help you predict certain targets
- Data can be noisy
- You might need to pre-process it
- The more data the better
- Choosing the right classifier/regressor is important
  - Cross-validate and test