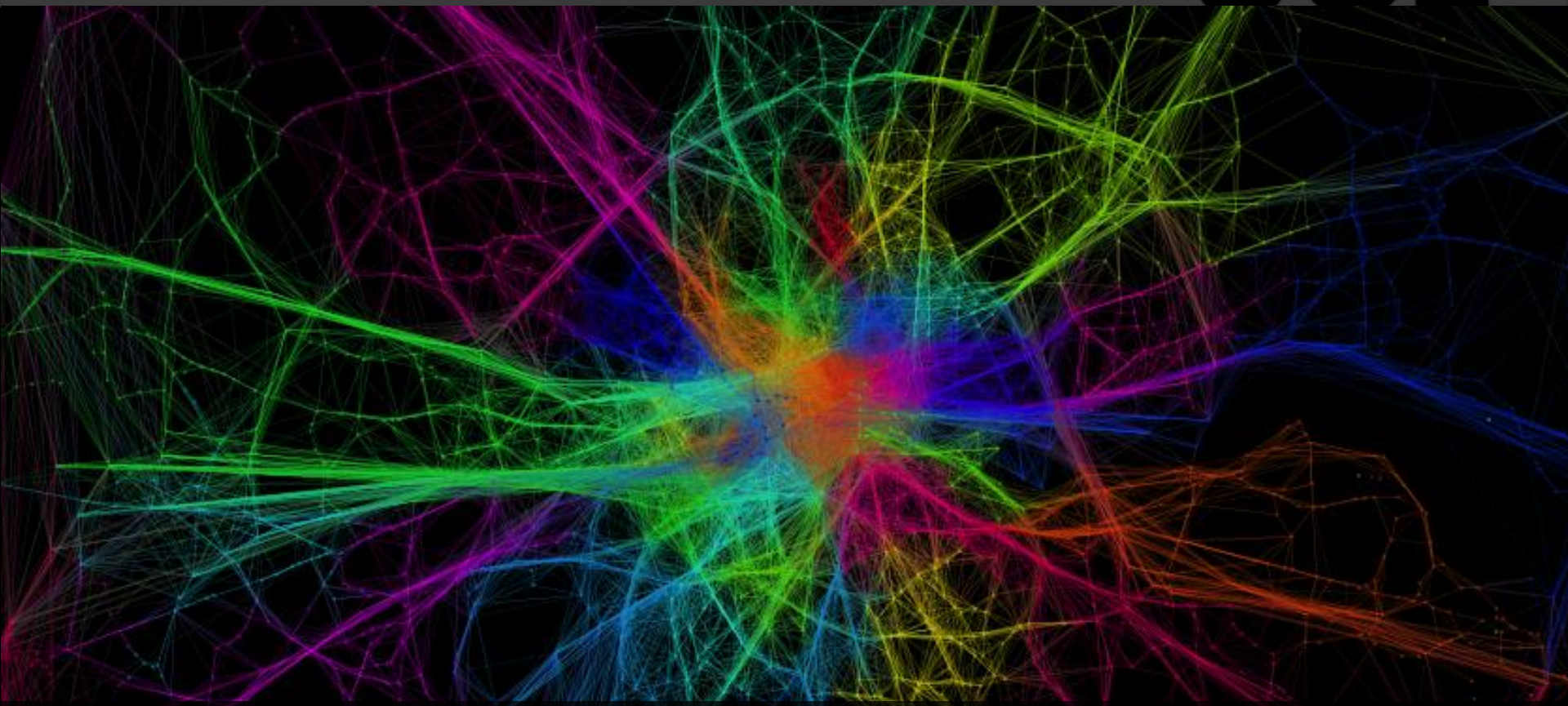


Advanced Regression

CASA0006: Spatial Data Capture and Analysis
CASA0009: Data Science for Spatial Systems

Huanfa Chen

CASA0006

1. Introduction to Databases
2. Introduction to SQL
3. Advanced SQL
4. Data Munging
5. Advanced Clustering
6. Advanced Regression
7. Classification
8. Dimension Reduction
9. Unstructured Data
10. Analysis Workflow

CASA0009

1. Introduction to Databases
2. Introduction to SQL
3. Advanced SQL
4. Data Munging
5. Advanced Clustering
6. Advanced Regression
7. Interactive Viz 1: HTML + CSS
8. Interactive Viz 2: Javascript
9. Server Side Coding: Node.JS
10. Real-time data visualisation

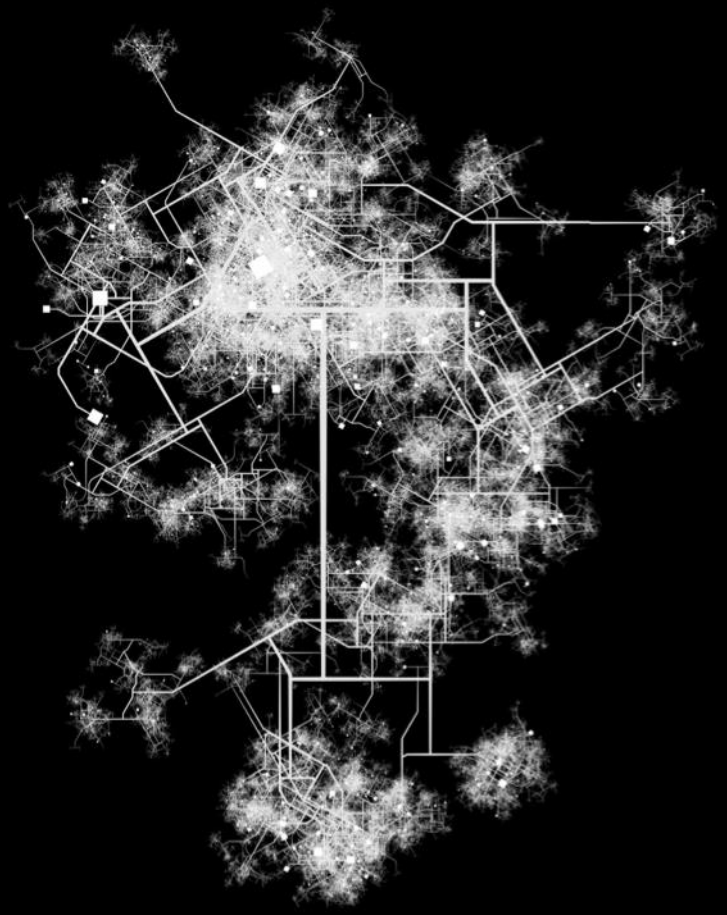
Data Analysis
Picking an Approach

Input Dataset →

Method		Output
Clustering	→	Creation of Groupings
Regression	→	Identify Data Relationships
Classification	→	Identify Discrete Class
Dimensionality Reduction	→	Understand Influential Factors
Association Rule Mining	→	Identify Dependencies
Anomaly Detection	→	Identify Outliers

Unsupervised: no ground truth
Supervised: with ground truth

Outline



1. Overview of ML and regression
2. Recap of linear regression, VIF, Lasso
3. Regression tree
4. Ensemble learning: RF, GBDT
5. Key issues of supervised learning

Acronym

- ML: machine learning
- CART: classification and regression tree
- RMSE: root-mean square error
- SSE: sum of square error
- RF: random forest
- GBDT: gradient boosting decision tree
- FI: feature importance

Overview of ML and regression

Machine learning

"[Machine learning is the] field of study that gives computers the ability to learn without being explicitly programmed." (Authur Samuel, 1959)

“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.“ (Tom Mitchell, 1997)

Three classes of ML

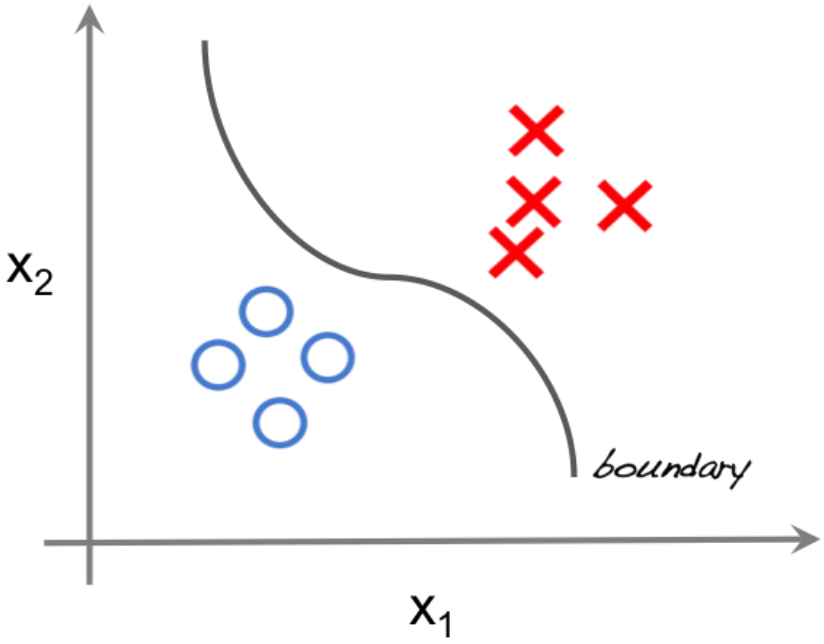
1. **Supervised learning**: learn to predict output given input (based on labelled training data)

2. **Unsupervised learning**: discover hidden representation of input (no labelled training data)

3. **Reinforcement learning**: learn action to maximise payoff

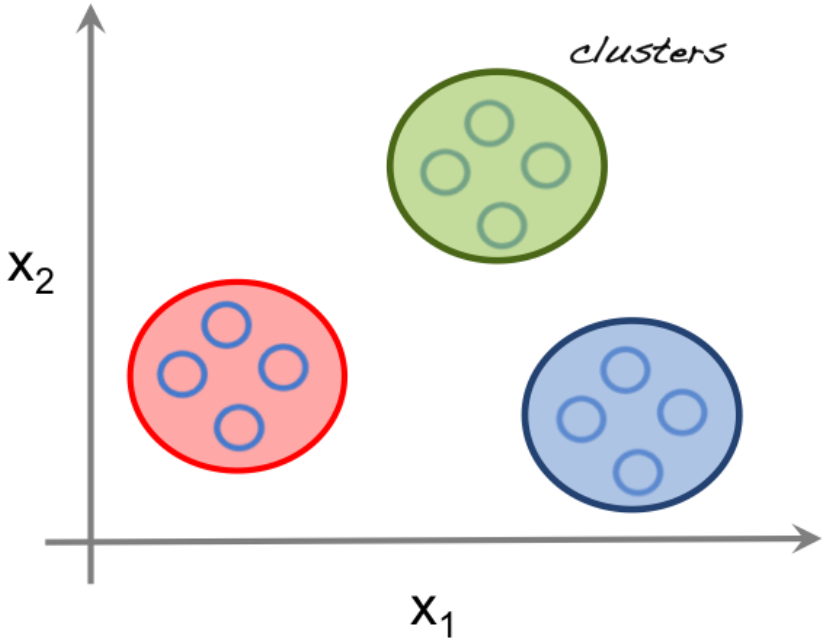
Three classes of ML

Supervised learning



Example: regression, classification

Unsupervised learning



Example: clustering, dimension reduction

Supervised machine learning

- Select a model f, and model y from inputs x as y=f(x, Θ), Θ are the parameters of the model that are learned during training
- Always ask "what is the Θ of this model" when you learn a new model
- What is model training? It means learning Θ from data, aiming to minimise the difference between the predicted and true output of y (e.g. sum of squared error).

Supervised machine learning

Purpose of supervised ML

1. **Prediction**: Predict outcome given data
2. **Inference**: Understand relationship between variables. (Model interpretation)

Supervised machine learning

- Many supervised learning methods are applicable to both regression and classification, including DT, RF, GBDT, ANN.
- Difference

	Target output	Metric
Regression	(continuous) number	R2, RMSE
Classification	(discrete) class label	Accuracy, Recall, F1 score, etc.

Metrics of Regression

	Definition	Range	Trend
R Squared	$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$	(-Inf, 1]	The greater R^2, the higher accuracy
RMSE	$\sqrt{\sum(y_i - \hat{y}_i)^2}$	[0, Inf)	The smaller RMSE, the higher accuracy

NB:

1. By definition, R Square is not the square of a value. A negative R^2 does not mean the software is wrong – it just means the prediction algorithm is not well trained or not suitable.
2. A special case is that linear regression has a R^2 in the range of [0,1].

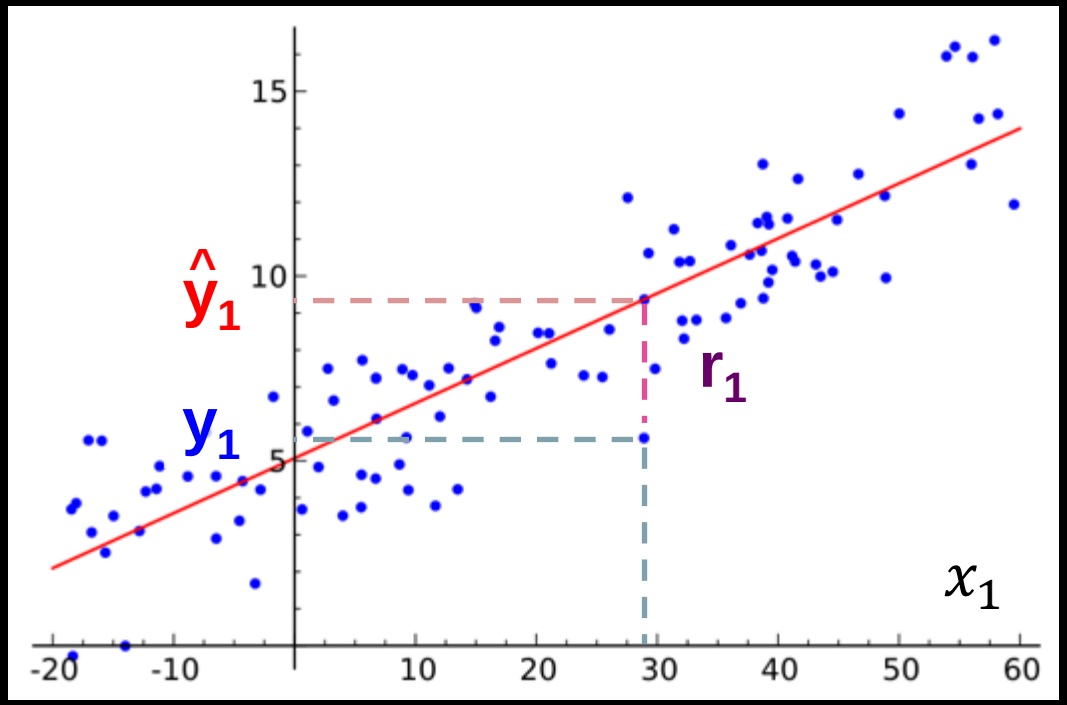
Regression methods

1. Linear regression (recap)
2. Tree-based methods
    1. Classification and regression tree (CART)
    2. Random Forest (RF)
    3. GBDT
3. Artificial neural networks (ANN) (Not covered in this lecture)

NB: tree-based and ANN methods are among the most important regression methods. Other methods exist, such as SVM.

Linear regression

$$\hat{y} = \sum_{i=1}^{k} m_i x_i + c$$



Assumptions

1. Y is a linear combination of x variables
2. X variables are independent from each other.

Parameters to learn

$m_1, m_2, \ldots, m_k, c$

Linear regression: interpretation

Example: predicting the daily number of rented bikes, given weather and calendar information

	Type	Weight
(Intercept)		2399.4
seasonSUMMER	Categorical	899.3
seasonFALL		138.2
seasonWINTER		425.6
holidayHOLIDAY		-686.1
workingdayWORKING DAY		124.9
weathersitMISTY		-379.4
weathersitRAIN/SNOW/STORM		-1901.5
temp	Numerical	110.7
hum		-17.4
windspeed		-42.5
days_since_2011		4.9

- 'workingdayWORKING DAY'

When it is working day, the predicted number of bicycles is 124.9 higher compared to weekend, *given all other features remain fixed*

- 'temp'

An increase of the temperature by 1 degree Celsius increases the predicted number of bicycles by 110.7, *when all other features remain fixed*

https://christophm.github.io/interpretable-ml-book/limo.html#interpretation
https://christophm.github.io/interpretable-ml-book/bike-data.html#bike-data

Linear regression
Advantages

- **Transparent and intuitive**: the prediction being a weighted sum of predictors.
- **Good interpretability**
- **Wide acceptance**: used for inference and predictive modelling in many subjects and fields.
- **Deterministic**: the optimal weights are guaranteed.
- **With a large toolbox**: backed by solid statistical theory, confidence intervals, tests, extensions (e.g., generalised linear model).

Linear regression Problems

- **Multicollinearity**: when some predictors are highly correlated, variance of the coefficient is large and the model becomes unstable and unreliable (solution: VIF, Lasso)
- **Difficulty to account for non-linearity or interaction**: these have to be hand-crafted and explicitly given to the model as an input feature. In the bike case, to account for high-temp and high-humidity weather, we need to create an interaction term temp*hum
- **Low predictive performance**: the relationship that can be learned are restricted, leading to low predictive performance.

Variance Inflation Factors (VIF)

- Given $x_1, x_2, \ldots, x_p$, the VIF for the x_k variable is

$$VIF_k = \frac{1}{1 - R_k^2}$$

where R_k^2 is the R^2 value obtain by regressing the x_k on the remaining x variables:

$$x_k = \sum_{i=1}^{k-1} b_i x_i + \sum_{i=k+1}^{p} b_i x_i$$

- The larger VIF_k, the higher multicollinearity, as x_k can be largely represented by a linear combination of the other variables and thus x_k is redundant.

VIF for variable selection

1. Initialise L as the list of predictor variables (HINT: the response variable is not needed for VIF)
2. Calculate the VIF for each variable in L. (HINT: the order of computing VIF is irrelevant).
3. If the highest VIF is larger than the threshold, remove the corresponding variable from the list L. A threshold of 5 is often used.
4. Repeat Step 2-3, until no VIF is larger than the threshold.
5. Output L.

[Optional] more about VIF:
https://online.stat.psu.edu/stat501/lesson/12/12.4

Lasso
Perform feature selection using a penalty function

Linear regression: $y = \beta x + \beta_0$

Obj: $\min_{\beta,\beta_0} \text{SSE} = \min_{\beta,\beta_0}\{\frac{1}{N}\sum_{i=1}^{N}(y_i - \beta_0 - \beta x_i)^2\}$

With penalty term:

Lasso: $\min_{\beta,\beta_0}\{\text{SSE} + \lambda \sum ||\beta||_1\}$

- $||\beta||_1$ is called L1-norm

e.g. given a linear model y=-3x_1+4x_2+5, $||\beta||_1$=|-3|+|4|=7

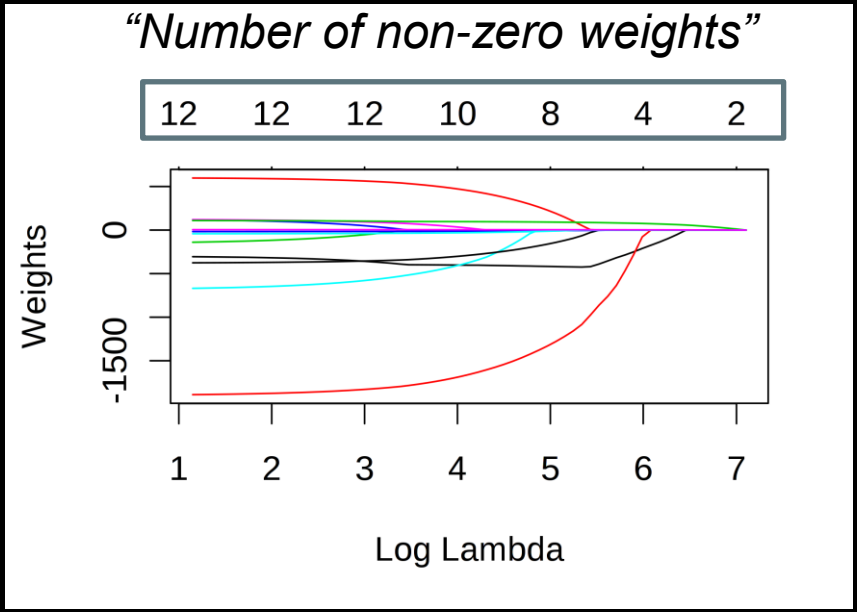
- λ is a hyperparameter controlling the strength of penalty effect.
- As λ increases, the fewer predictors are present in the model, as their weights become zero.
- λ needs to be tuned (using Lasso path).

The methods of ridge and elastic net are similar with Lasso, but the penalty term is a bit different.

Lasso
Perform feature selection using a penalty function

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_0$$

Lasso: $\min_{\beta,\beta_0} \{\text{RSS} + \lambda \sum ||\beta||_1\}$

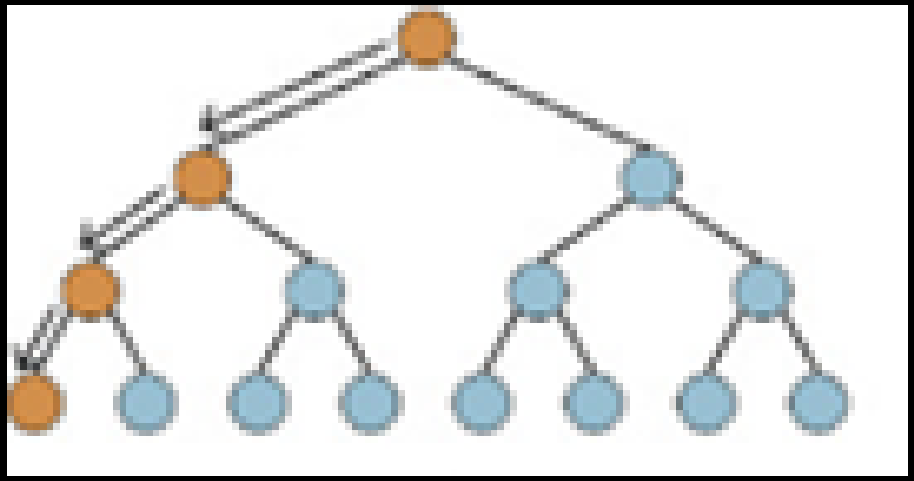


Lasso path

- λ controls the model.
- Each β_i is a function of the lambda, and is represented as a curve in the figure.
- The weight decreases as λ increases.
- Advantages of Lasso
  - It can be automated. λ can be learnt using cross validation
  - It considers all predictors simultaneously.

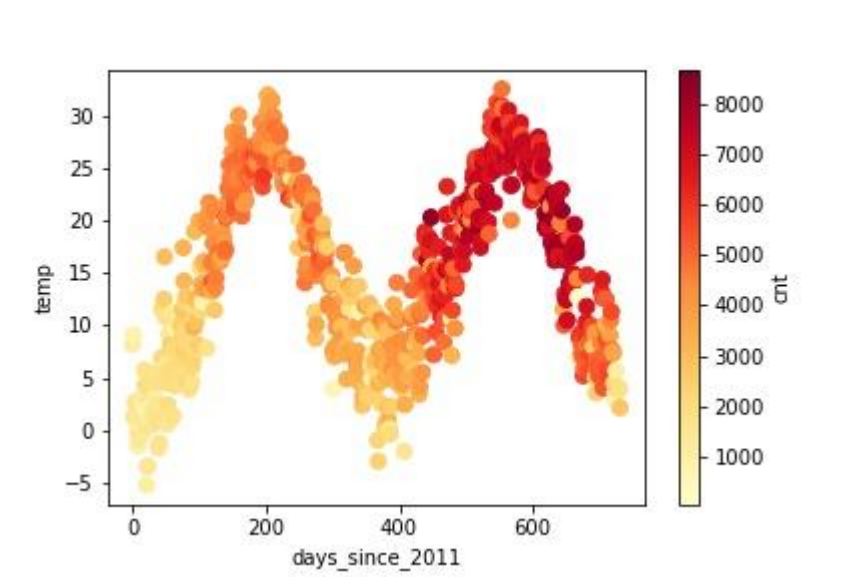
CART

- Simply speaking, CART consists of a flow diagram or a ‘tree’ of decisions about the x variables of a dataset
- Data-driven approach, no assumptions about the data relationship



CART

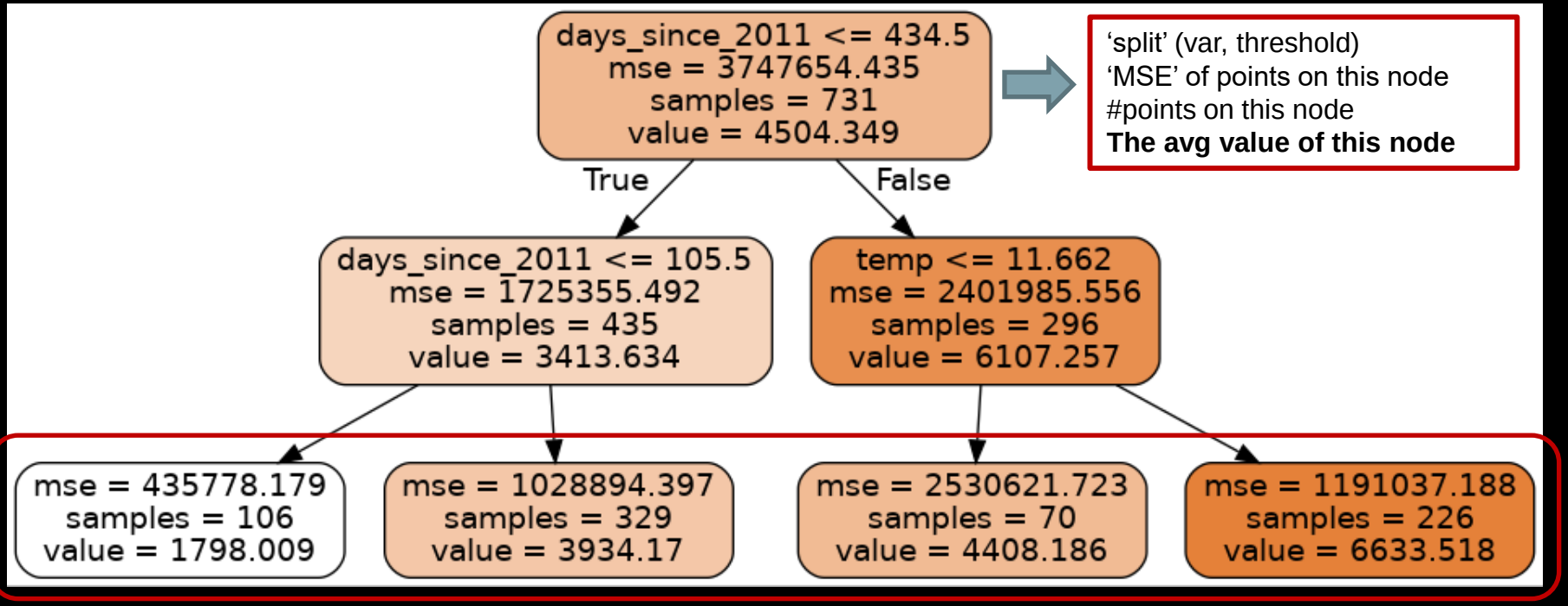
Example: predicting the bike rental using two variables (days_since_2011, temp)



- 731 data points
- Each point is (temp, days_since_2011, bike_rental)
- NB: this is a very simple example of CART. CART is applicable to any dimension of variables

CART

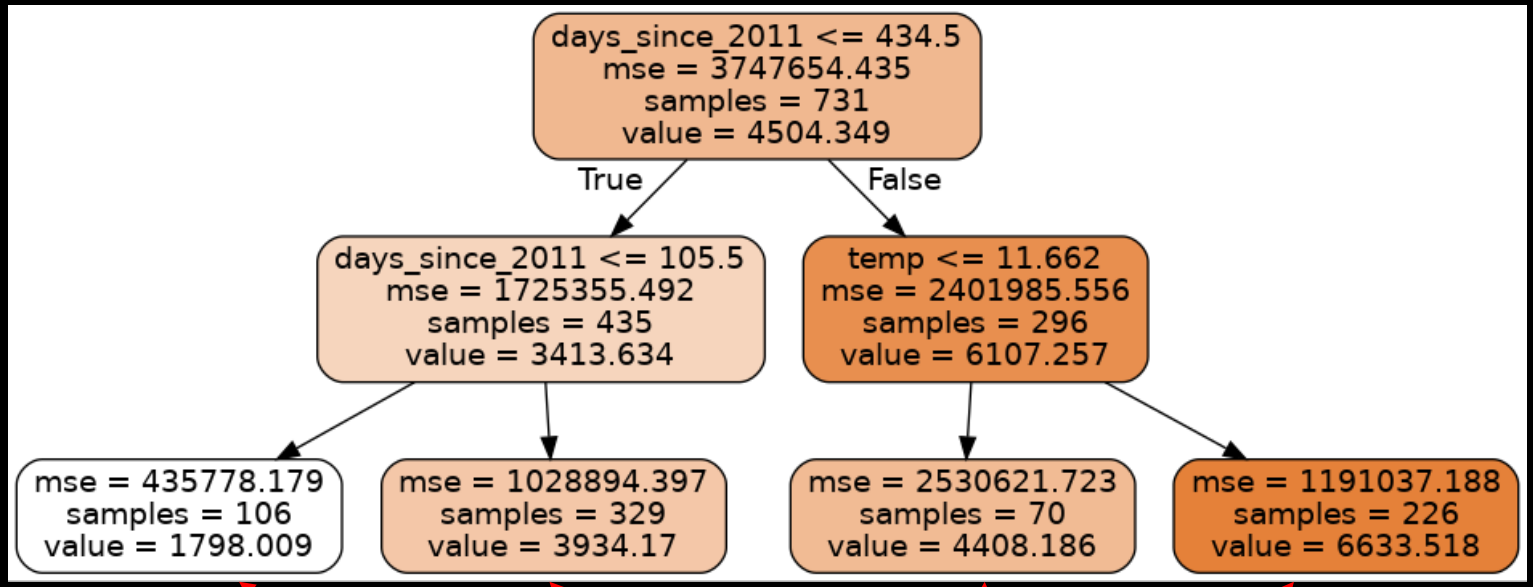
The trained CART for this example

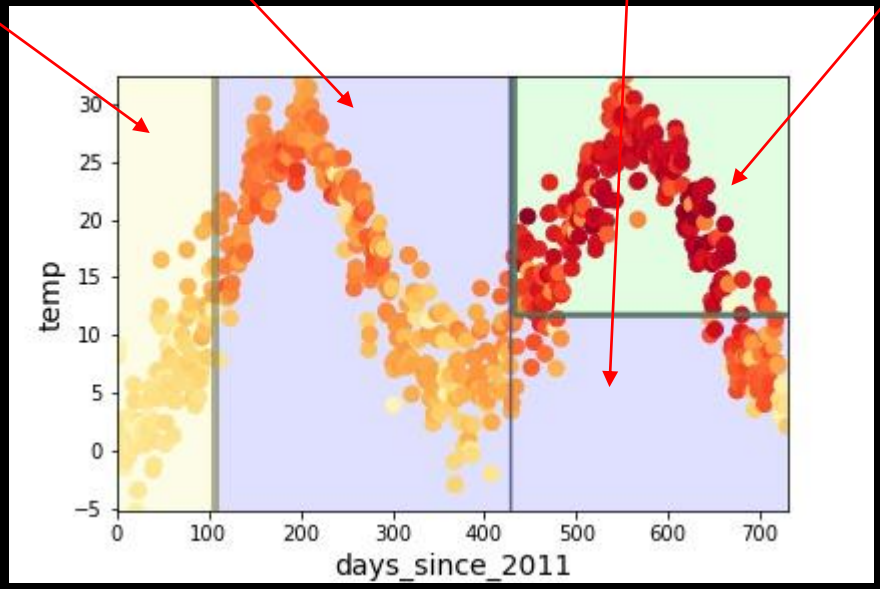


Leaf node. If a data point fall into a leaf, then it is predicted as the 'value' of this leaf.

Another visualisation of this CART

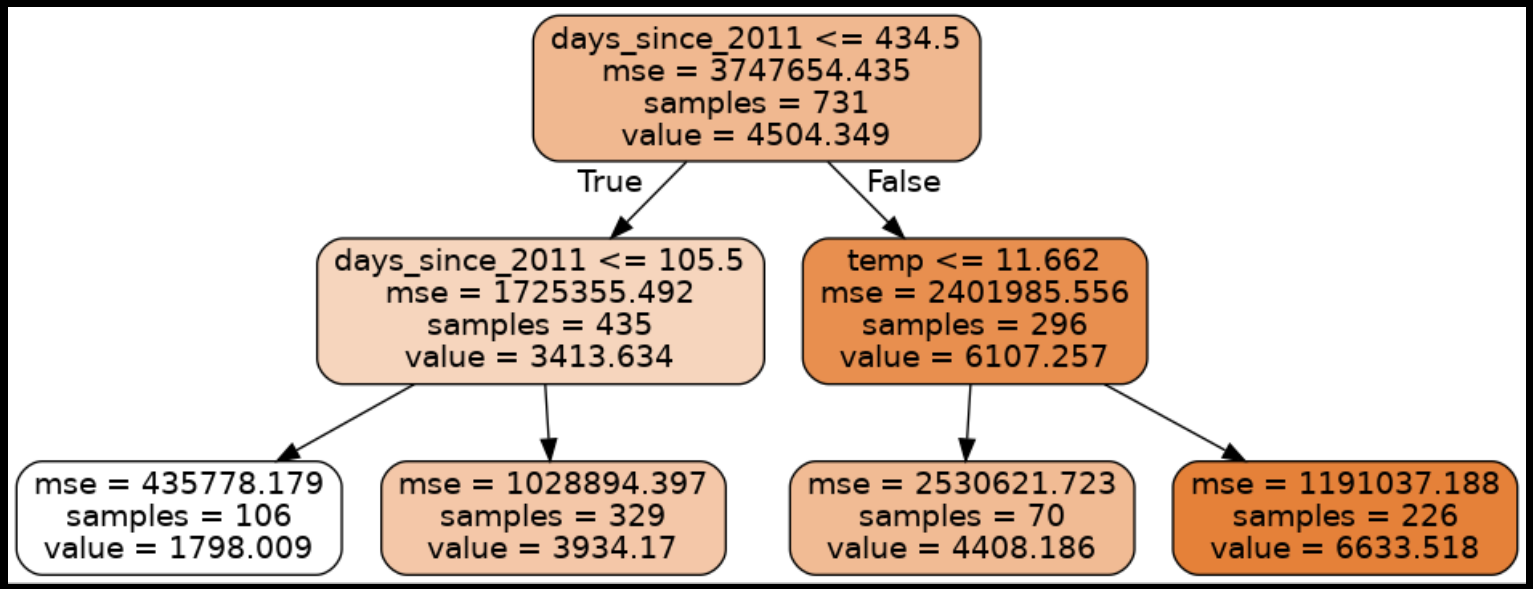
CART





CART

Using CART for prediction



(days_since_2011, temp)

(435, 12): predicted_bike_rental = ??

(434, 12): predicted_bike_rental = ??

CART

- Training of the CART
  - Splits the sample into two subsets using a single variable k at threshold t_k (note only splits into two)
  - Chooses k and t_k by finding pair that minimise the cost function (aka the weighted sum of within-group variation)

$$J(k, t_k) = \frac{m_{\text{left}}}{m} MSE_{\text{left}} + \frac{m_{\text{right}}}{m} MSE_{\text{right}}$$

    - 'left' and 'right' refer to two groups and m_{left} refers to the number of points in group left. $m = m_{\text{left}} + m_{\text{right}}$. MSE is mean square error
  - Repeat the splitting until stop criteria are met

CART

- Stopping criteria of CART (predefined by user)
  - Max tree depth
  - Minimal instances in a node

- Consider a CART as $y=f(x, \Theta)$. What are the Θ?
  - The split (variable and threshold)
  - The leaf node value of the tree

CART

- Advantages of CART
  - **Interpretability**: relatively easy to understand (but a deep tree is not easy to understand)
  - **Flexibility**: no assumptions of data distribution and no transformations needed

- Disadvantages
  - **Lack of smoothness**. Slight changes in the predicators can have a big impact on the response
  - **Tendency of overfitting:** meaning that the tree fits well to the training data but is unable to generalise to new data

- NB
  - CART can be used for both regression and classification
  - The problem of CART will be tackled by RF or XGBoost
  - It is uncommon to use CART to directly make predictions. Rather, CART is used to construct RF or GBDT.

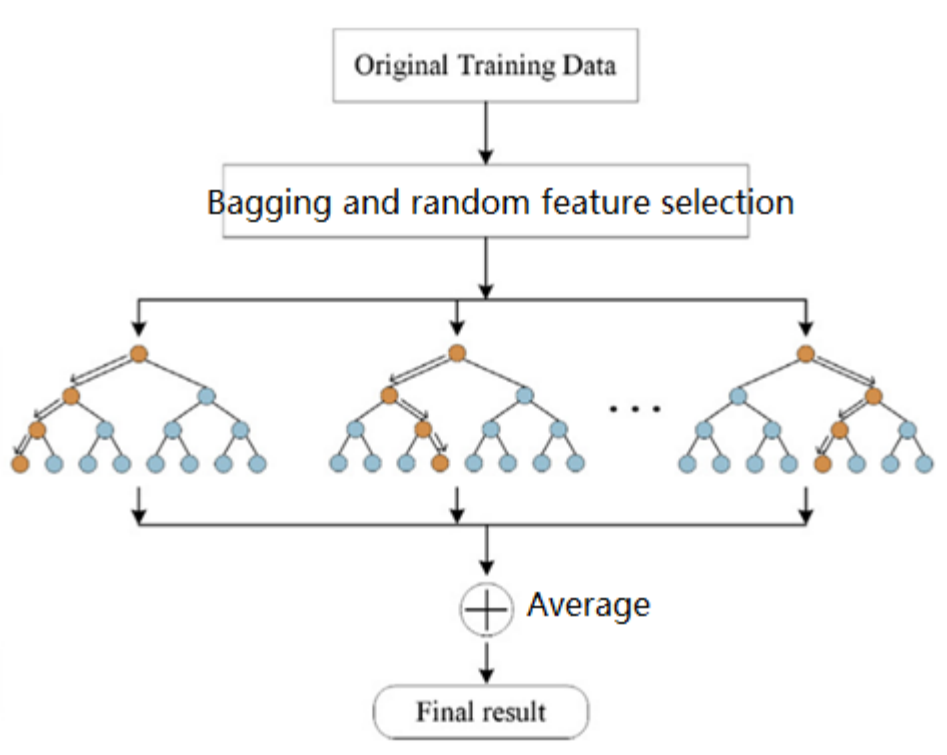
Ensemble learning

- Wisdom of the Crowd
- The average from many predictors may be more accurate any single given predictor
- Even if individual predictors are weak (only slightly better than random), an ensemble can be strong (accurate).
- Group of predictors called an *ensemble*

- CART is a good unit for ensemble learning
  - Training a CART is relatively easy and cheap
  - CART makes no assumptions on input data

- Two common approaches of ensemble learning
  - Bagging (random forest)
  - Boosting (gradient boosting decision tree, GBDT)

Random Forest



- RF is a collection of many different CARTs.
- Given an input, the prediction of RF is a combination (e.g. average) of the output of all trees.

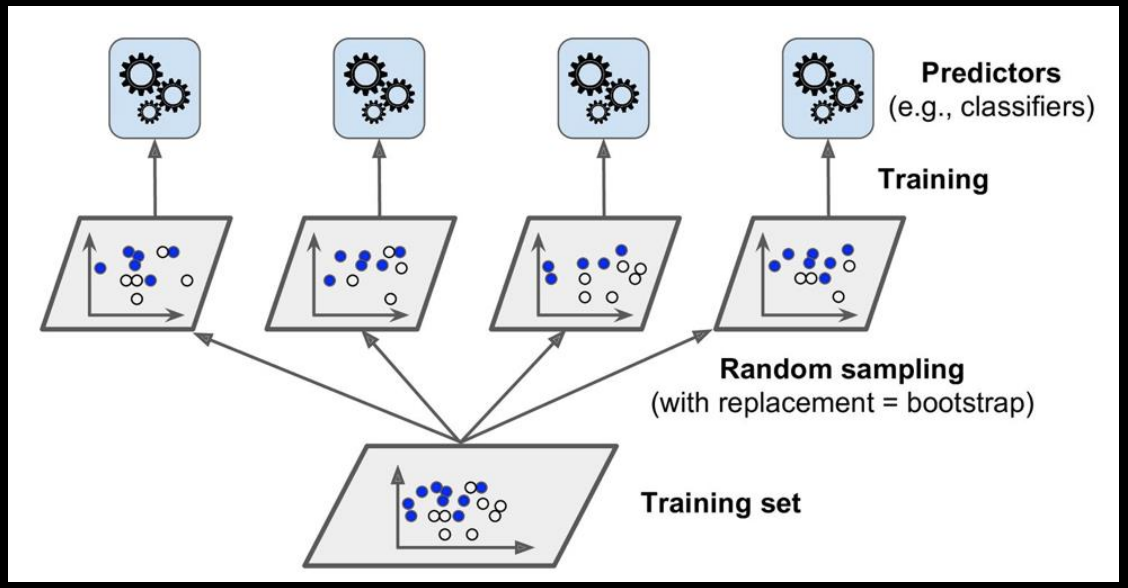
[Amended from image source]

Random Forest

Two techniques to grow different and diverse trees

1. Bagging (short for bootstrap aggregating): sampling instances
2. Random feature selection: sampling features

As each CART sees different training data, the trees are different.



[image source]

Random Forest

Two techniques to grow different and diverse trees

1. Bagging (short for bootstrap aggregating): sampling instances
2. Random feature selection: sampling features

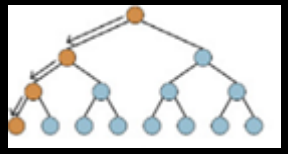
Bootstrap: sampling with replacement. It guarantees that the sample has the same distribution as population; some instances may be sampled repeatedly.

Index	x1	x2	x3
1	2	1.0	2
2	3	1.5	3
3	5	2.0	4
4	4	2.6	6

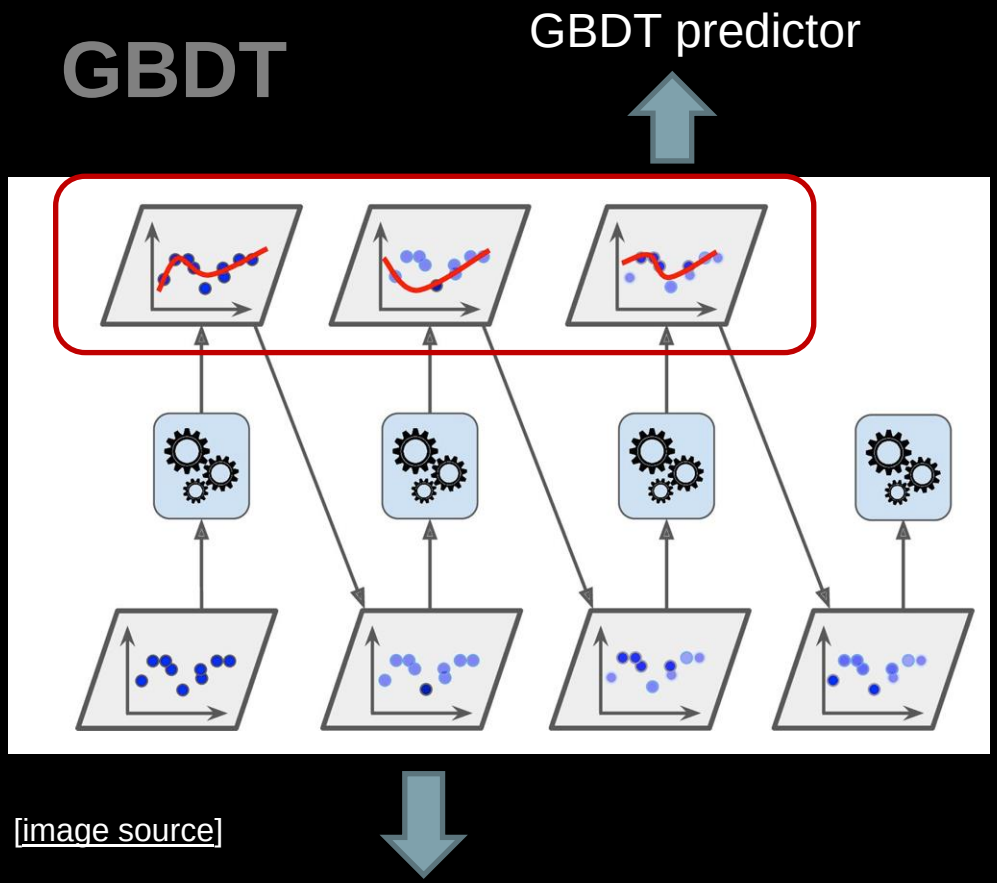
Sampling:
2 features,
7 samples

Index	x1	x3
1	2	2
2	3	3
3	5	4
4	4	6
2	3	3
4	4	6
2	3	3

Train a tree



GBDT



[image source]

A deeper colour means larger residual and then larger weight for the next predictor

- While RF grows trees horizontally (or in parallel), GBDT grows trees vertically (or sequentially)
- A new CART predictor is trained using the residual from the last CART as the weight. It focuses on the inaccurate prediction (with larger residual).
- All trees are combined to form the ensemble (similar to RF)

GBDT

Implementations

- GradientBoostingRegressor from sklearn
  - Good for small projects
- XGBoost (another package)
  - Efficient, robust, Industry-level implementation of GBDT
  - Winner of many data science competitions
  - Highly recommended

- Machine learning = theory + engineering

RF and GBDT

- Advantages
  - No assumptions on data distribution
  - Able to model non-linear relationship and feature interactions
  - Good predictive performance
  - Good generalisation
- Disadvantages
  - Low interpretability: not intuitive, although there are some interpretation methods

Permutation feature importance: interpreting RF/GBDT

- The idea is straightforward. We measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature.
- A feature is "important" if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction.
- In contrast, a feature is "unimportant" if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction.
- This method is model-agnostic
  - Applicable to linear regression, CART, RF, GBDT, etc.
  - Applicable to regression and classification task

Permutation feature importance

[x1,x2,x3]

x1	x2	x3
2	1.0	2
3	1.5	3
5	2.0	4
4	2.6	6

y

y
12
15
16
20

Trained model
$f(x, \Theta)$

1. Estimate the error on the dataset: $e_1=L(y, f([x_1,x_2,x_3]))$
2. Shuffle x1 and get a new dataset $[x_1',x_2,x_3]$
3. Re-estimate the error on the shuffled data $e_2=L(y, f([x_1',x_2,x_3]))$
4. The PFI of x_1 is the difference between e_2 and e_1.
5. Repeat Step 3-4 for x_2 and x_3. You can sort out the important from largest to smallest.

$[x_1',x_2,x_3]$

x1	x2	x3
3	1.0	2
5	1.5	3
4	2.0	4
2	2.6	6

Other interpretation

- Other types of feature importance, such as Gini importance for RF, standardised coefficients for regression. Note that some FI measures are model-specific
- Interpretation of ML is an emerging field
- **Partial dependence plot** shows the marginal effect one or two features have on the predicted outcome of a ML model
- Section 8.1 and 8.5 of this book: https://christophm.github.io/interpretable-ml-book/

Summary of regression

- **Regression** is one type of supervised machine learning. Its target out is a continuous number (compared to a class label of classification)
- **Two metrics**: R^2 and RMSE

- Linear regression: fit linear relationship. Using VIF or Lasso to tackle multicollinearity problem and select variables
- CART: tending to overfitting. Base predictors for RF and GBDT
- Ensemble learning: combine many ‘weak’ predictors
- RF: creating trees horizontally. Two ideas: bagging and random feature selection
- GBDT: creating trees sequentially. Core idea: using residual to reweight data points in the next tree

Summary of regression

	Assumptions	Parameters	Accuracy	Interpretation
Linear model	Many	Few (coefficients and intercept)	Usually low	Easy
RF/GBDT	Few	Many	High	Hard, e.g. PFI

Key issues of supervised learning

Parameters & hyperparameters

- Parameters: $y=f(x, \Theta)$, Θ are the **parameters** of the model that are learned during training. They are not predefined by the user.

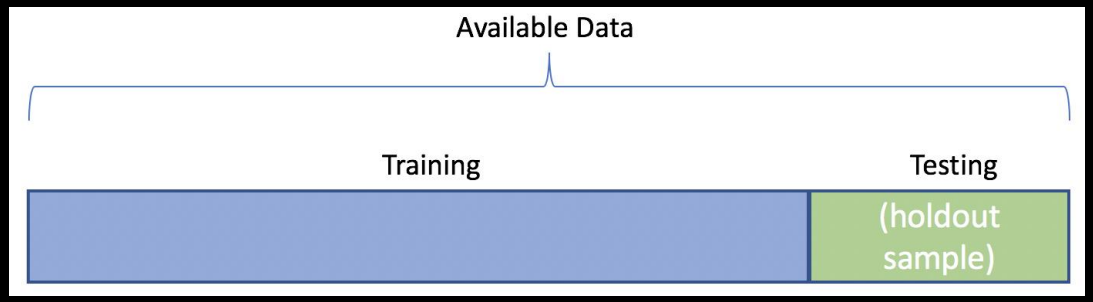
- In ML models, a **hyperparameter** is a 'setting' whose value is used to control the learning process. It should be predefined before model training. Many algorithms provide default value of hyperparameters, but the hyperparameters need to be tuned and optimised by the user

Parameters & hyperparameters

	Parameters	Hyperparameters
Linear model	Coefficient, slope	None
Random forest	The split and the leaf node value of trees	Number of trees, minimum number of instances in a node

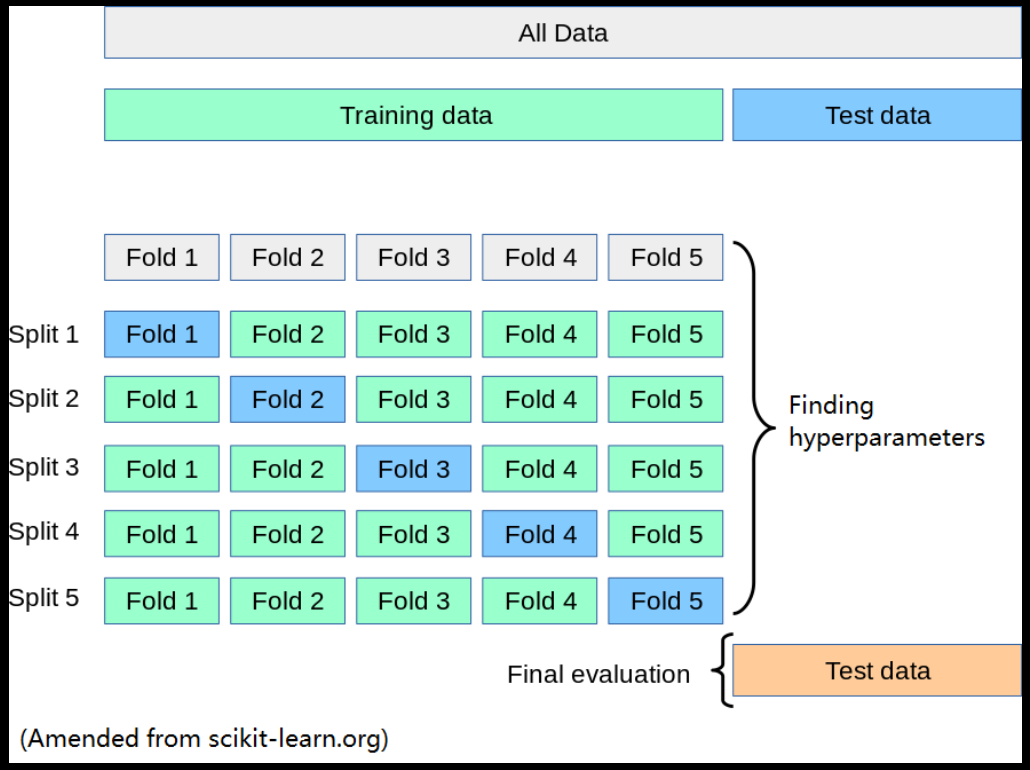
Data split (training & testing)

- A fair evaluation of model performance should be based on training/testing split. Usually, 75% for training and 25% for testing.
- Note that a random split is essential – to avoid selection bias.
- The training data is used to train the model, while the testing data is used to report the model performance
- You can train your model on the training data, and then apply the trained model to the testing data. Then, you can report the accuracy of the testing data to your boss.



Cross validation (CV)

- What if you want to tune the hyperparameters of the model (e.g. number of trees (NT) in a RF)?
- Answer: using cross validation!



Here is an example of 5-fold CV. 5 models will be trained.

Model 1: trained using Fold 2-5, evaluated using Fold 1

Model 2: trained using Fold 1,3-5, evaluated using Fold 2

……

Cross validation (CV)

- Assume you want to tune NT of a RF, from [10, 100, 200]. We will use 5-fold CV.

- To know the Accuracy of NT=10, 5 models will be trained. The accuracy of NT=10 is the mean of accuracy of these 5 models.

- Repeat this step for NT=100, NT=200. Then compare their accuracy. Imagine NT=100 has the highest accuracy, then we will set NT as 100. Then we retrain a RF (called RF_final) with NT=100 using the whole training data.

- Finally – we apply RF_final to the testing data and report its accuracy.

- Question: how many RF models are trained in the above process? (Hint: 3*5+1=16)

Textbooks and tutorials

- VanderPlas, "*Python data science handbook*", O'Reilly, 2017, ISBN 9781491912058 (Example code)
- Geron (2nd Edition), "*Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*", O'Reilly, 2019, ISBN 9781492032649 (Example code)

- Scikit-Learn tutorial, VanderPlas

Thank You

Questions?

Huanfa Chen

huanfa.chen@ucl.ac.uk

Workshop

- This workshop will focus on using regression methods to analyse a multivariate dataset
- You'll continue to use the scikit-learn Python library.
- Download this week's Python Notebook from Moodle, open it in Anaconda and work through