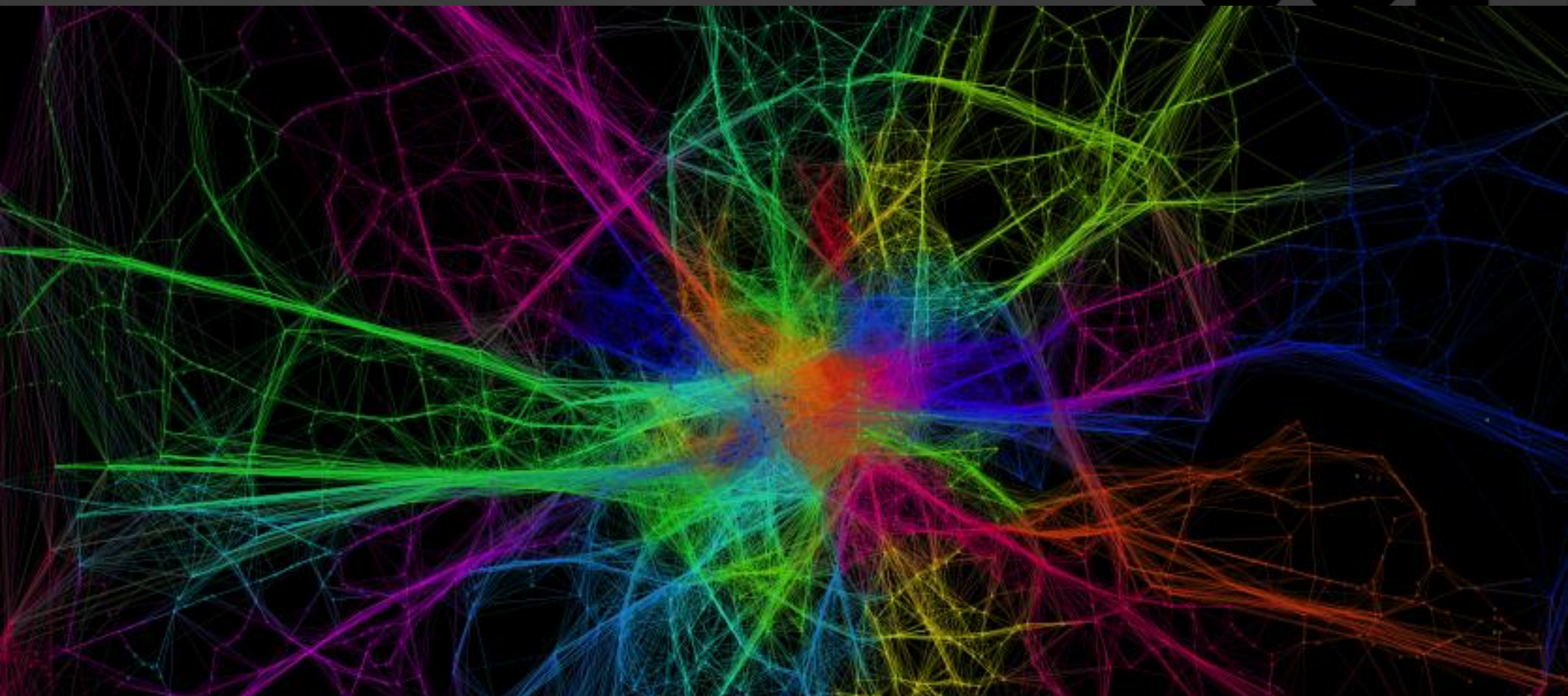


Classification

CASA0006: 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

Recap
What we already know

Handling and cleaning data

Database, SQL, Python Pandas/Sklearn

Clustering analysis

Kmeans, DBSCAN, hierarchical clustering

Regression analysis

Linear regression, VIF, Lasso, CART, RF, GBDT

Classification analysis (this week's topic)

Data Analysis
Picking an Approach

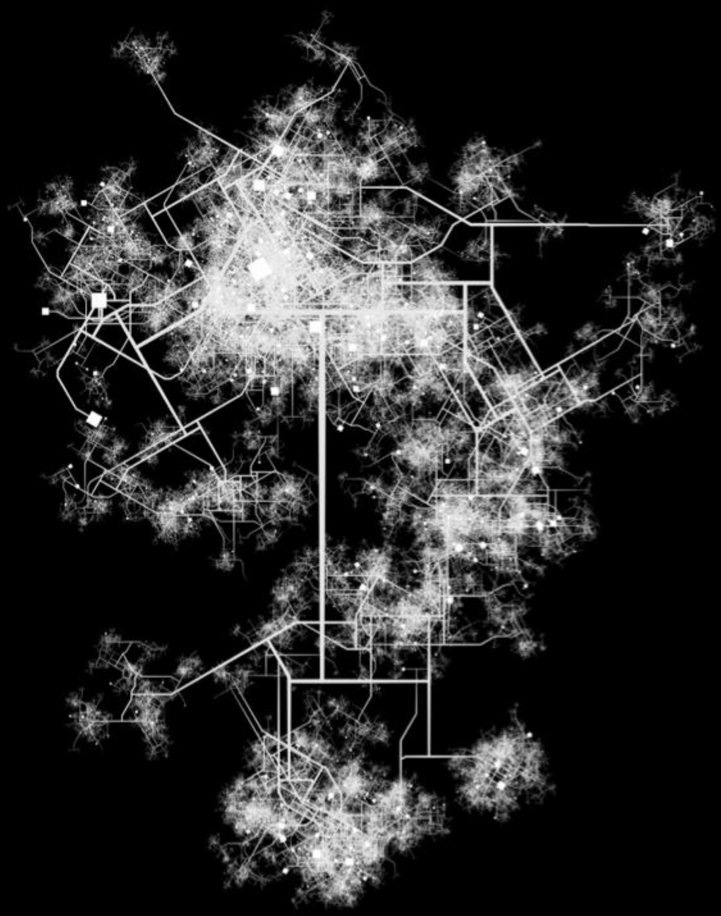
Input Dataset ➡

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



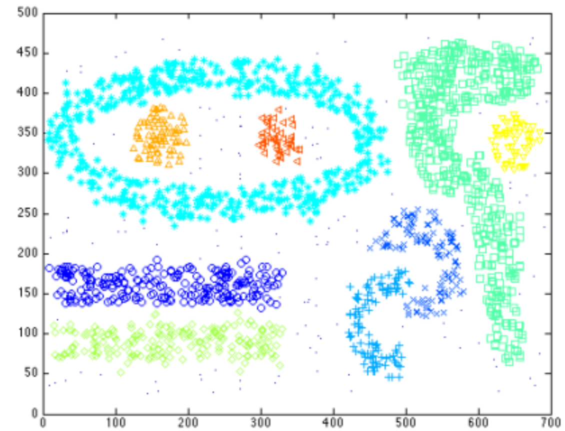
1. Overview of classification
2. CART for classification
3. RF and GBDT for classification
4. Logistic regression
5. ANN

Classification

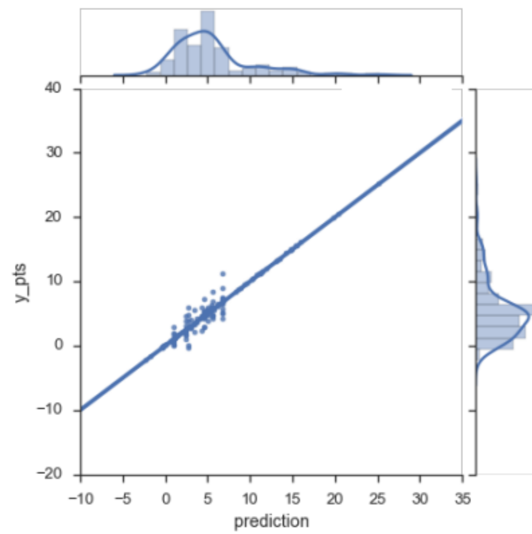
UCL

Unsupervised Learning

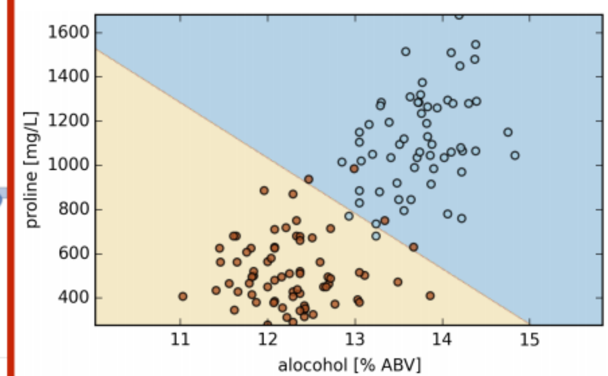
Supervised Learning



Clustering:
[DBSCAN on a toy dataset]



Regression:
[Soccer Fantasy Score prediction]



Classification:
[SVM on 2 classes of the Wine dataset]

Today's topic

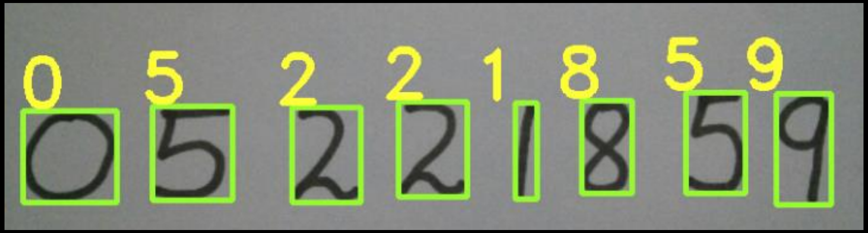
No labels

Continuous Y as labels

Discrete Y as labels

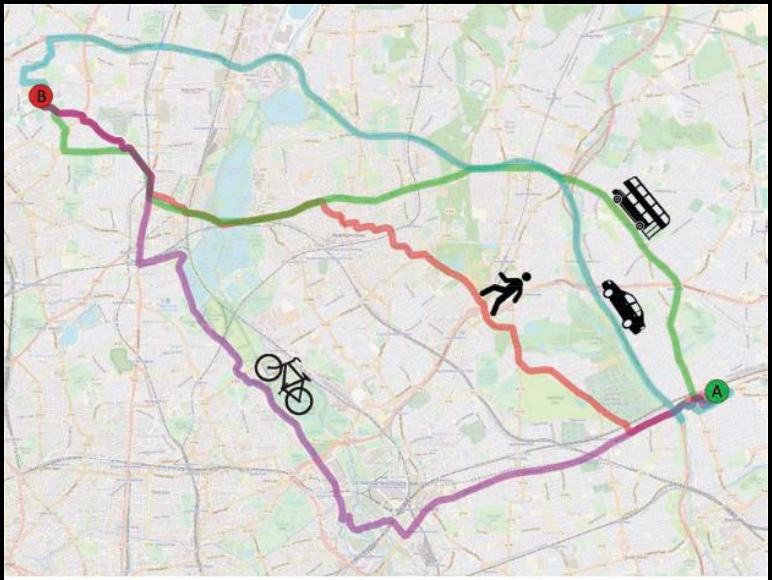
http://archive.ics.uci.edu/ml/datasets/Wine

Classification Examples



Digit Recognition

Spam Filtering



Transport Mode Detection
(from travel survey)

Categories of classification

- Number of classes
  - Two-class
  - Multiple-class (more complications)
- Predicting labels or probabilities
  - Hard classifier (only output one class, or a single class with probability 1 and all other classes with probability 0)
  - Soft classifier (output probabilities of different classes. The class with the largest prob is the output label)

Example: travel mode prediction based on travel surveys

	Walk	Transit	Driving	Cycling
Hard Classifier	0	**1**	0	0
Soft Classifier	0.2	**0.5**	0.1	0.2

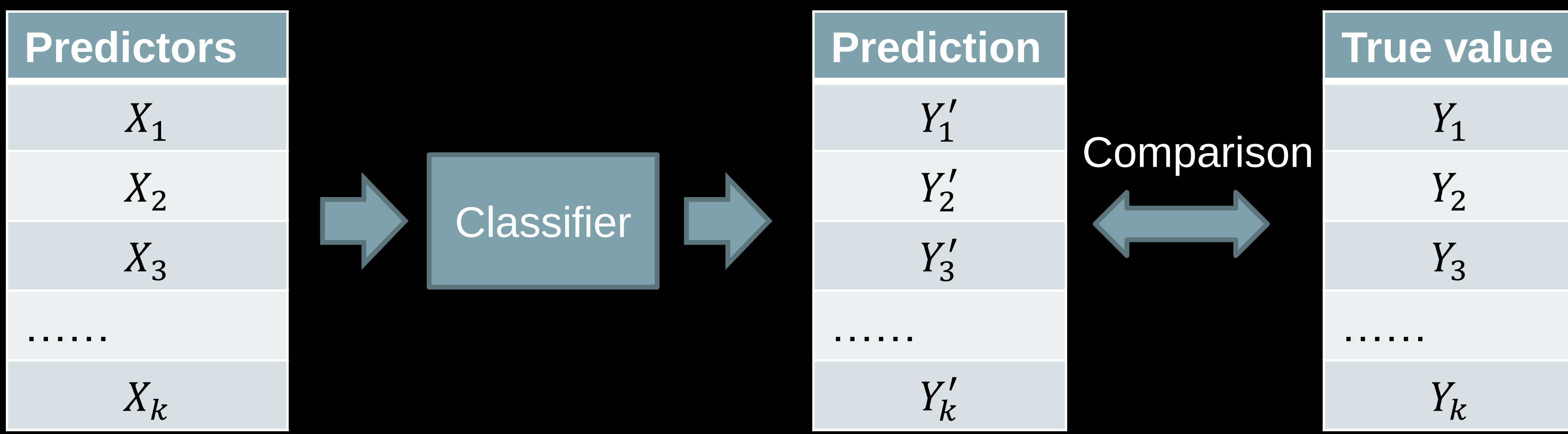
Classification vs. regression

- Similarity
  - Workflow (Train-test split, cross validation)
  - Methods (CART, RF, GBDT, ANN)
- Difference

	Target output	Metric
Regression	(continuous) number	R^2, RMSE
Classification	(discrete) class label	Accuracy, Recall, F1 score, etc.

Performance metrics

Example: predicting travel modes as one of four modes



How accurate are the predictions?

- For a single record: the prediction is TRUE or FALSE
- For many records: confusion matrix

https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Performance metrics

Confusion Matrix: comparing predicted against observed classes

Two-class (e.g. Driving vs. Not-driving)

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Performance metrics

Comparing predicted against observed classes

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Confusion Matrix

Classification Accuracy

Proportion correctly classified

$$\frac{Correct}{Correct + Incorrect} = \frac{\#tp + \#tn}{\#tp + \#tn + \#fp + \#fn}$$

Precision

How many positive predictions are correctly classified?

$$\frac{\#tp}{\#tp + \#fp}$$

Range of [0,1]

Recall

How many positive classes are correctly classified?

$$\frac{\#tp}{\#tp + \#fn}$$

Range of [0,1]

F1

A balance between precision and recall, takes beta attribute which weights precision or recall (usually beta = 1)

$$(1 + \beta^2)\frac{precision \; * \; recall}{\beta^2 precision + recall}$$

Range of [0,1]

Performance metrics

Q1: What is the conflict between precision and recall?

A: In many cases, improving one of them would lead to degrading of the other.

S1: predict most as ‘negative’ – maximise prec

Predicted

Actual

	Positive	Negative
Positive	1	99
Negative	0	100

Precision = 1
Recall = 0.01

S2: predict most as ‘positive’ – maximise recall

Predicted

Actual

	Positive	Negative
Positive	100	0
Negative	99	1

Precision = 0.5
Recall = 1

Performance metrics

Q2: why is F1 score a tradeoff between prec and recall?

A: the F1 score combines these two metrics into a single metric; It has a value between min(prec, recall) and max(prec, recall).

S1: predict most as 'negative' – maximise prec

Predicted

Actual

	Positive	Negative
Positive	1	99
Negative	0	100

Precision = 1
Recall = 0.01
F1 = 0.02

S2: predict most as 'positive' – maximise recall

Predicted

Actual

	Positive	Negative
Positive	100	0
Negative	99	1

Precision = 0.5
Recall = 1
F1 = 0.67

Performance metrics

Q3: What is the problem of accuracy (or why are other metrics needed?)

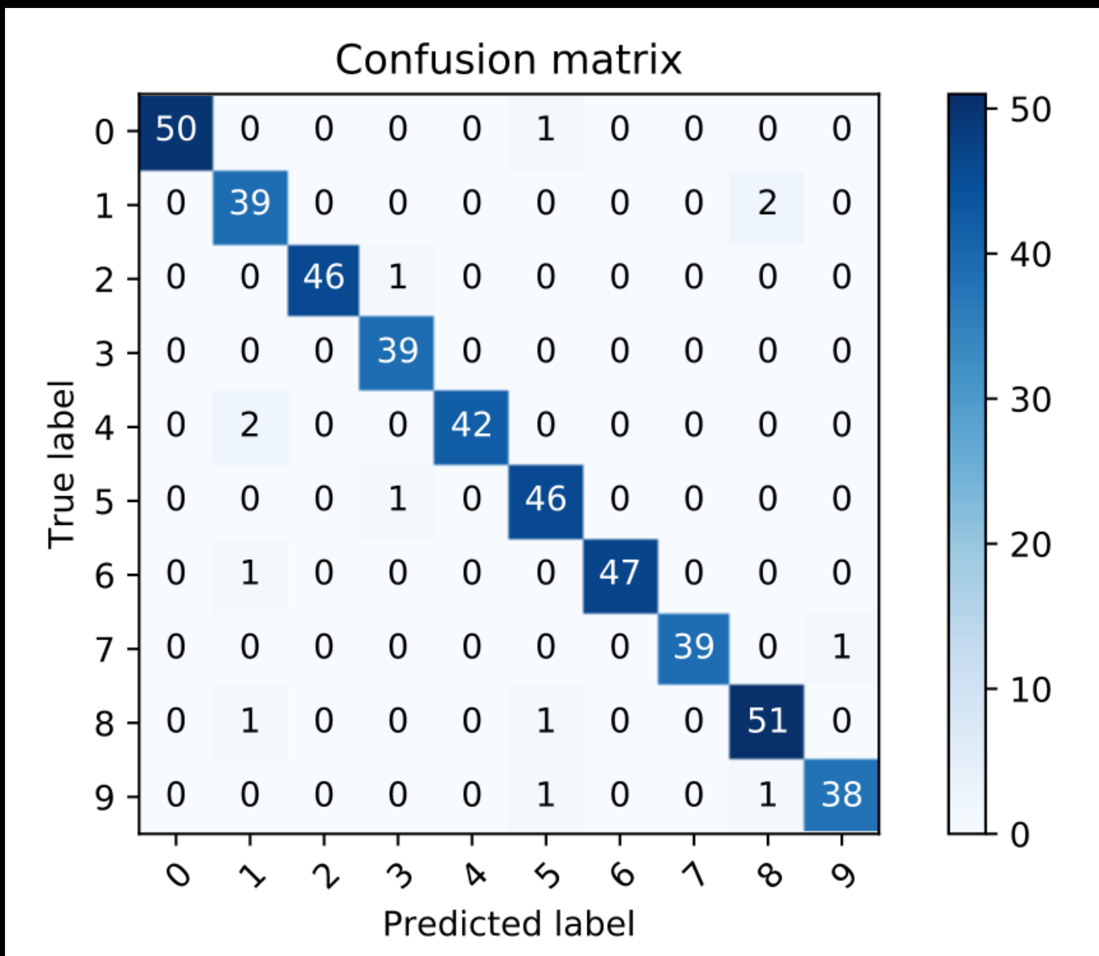
Accuracy does not tell details of ‘false positive’ or ‘false negative’, which are important for some applications (think about pregnancy test or Covid test)

Accuracy paradox: Accuracy might be not useful when the class distribution is highly imbalanced. For example, when predicting mode with actual 99% driving and 1% cycling, a ‘trivial’ predictor that simply predicts ‘driving’ will have very high accuracy, but this predictor is not useful.

Suggestion: when presenting classification results, you could present both accuracy and F1 score.

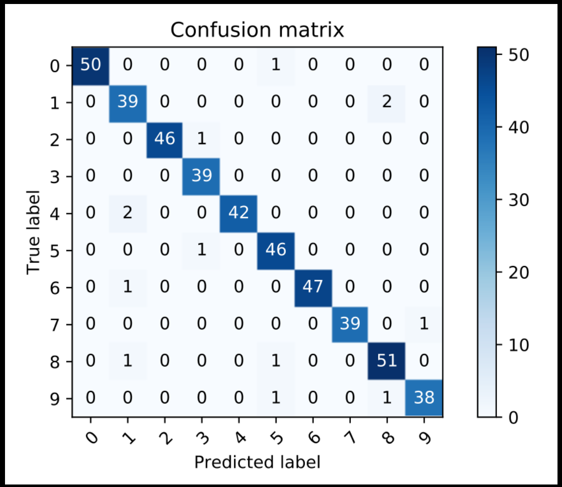
Performance metrics

Multiple-class problem (K classes)



- Numbers on the diagonal line are the TP for each class
- n_{ij} is the number of instances with actual class i that are classified as class j.
- When reading a confusion matrix, it is important to know meaning of rows and columns (rows as true label, or predicted label?)

Performance metrics



Multiple-class problem (K classes)

Classification Accuracy $\dfrac{\sum_i n_{ii}}{\sum_i \sum_j n_{ij}}$

Precision* $\text{average}(Prec_1, Prec_2, \ldots, Prec_K)$

Recall* $\text{average}(Rec_1, Rec_2, \ldots, Rec_K)$

F1* $\text{average}(F1_1, \mathrm{F1}_2, \ldots, F1_K)$

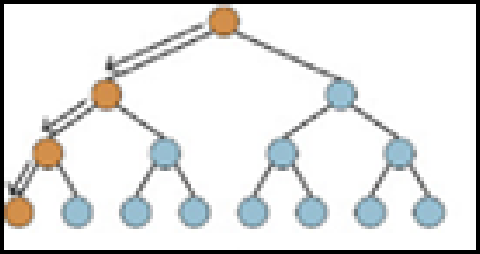
* This is called macro average of precision/recall/F1. Other ways of calculating these scores include 'micro' and 'weighted'. See here.

Classification trees

Classification trees vs. decision trees

- Similarity: overall idea, hyperparameters, feature importance, etc.
- Difference

	Cost function of split	Value of a node	Prediction
Regression	Weighted sum of MSE	Mean of all records on this node	A number
Classification	Weighted sum of Gini impurity	Majority class	A class or probability distribution over classes

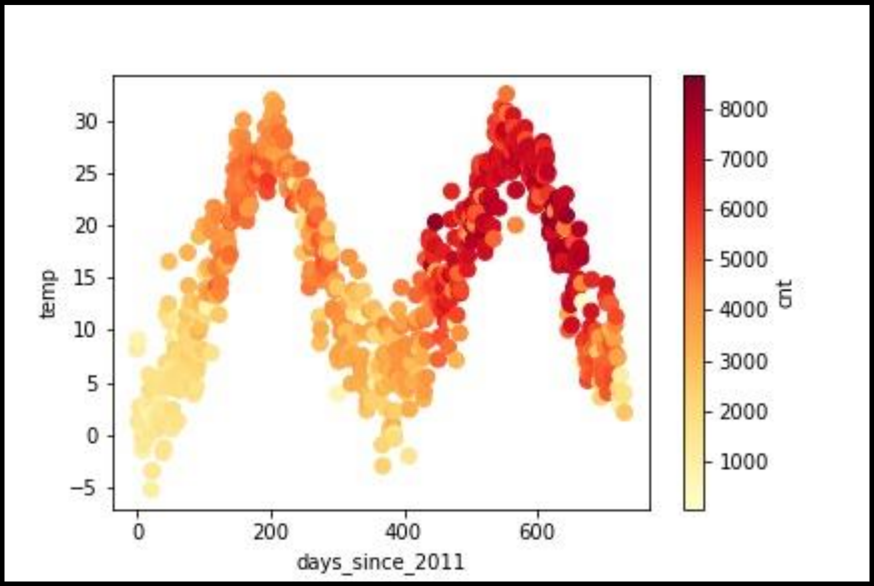


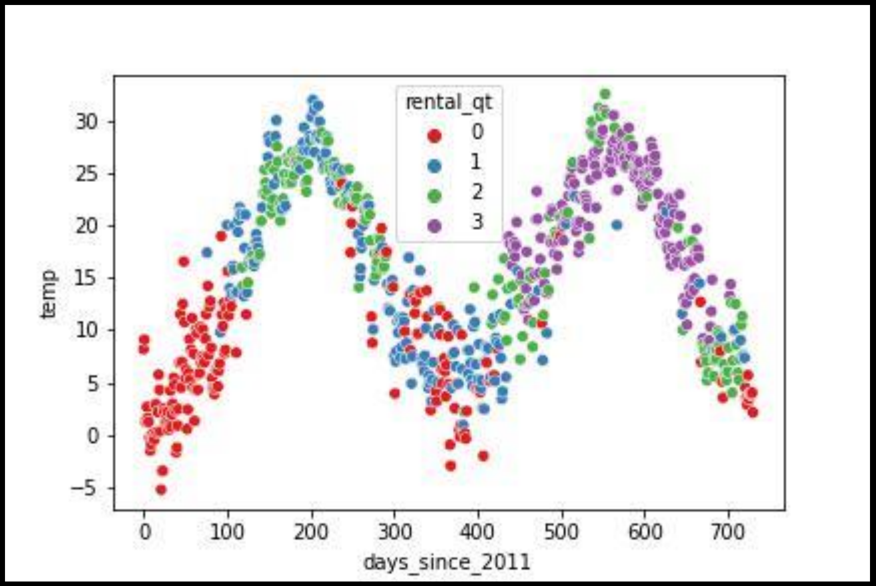
CART

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

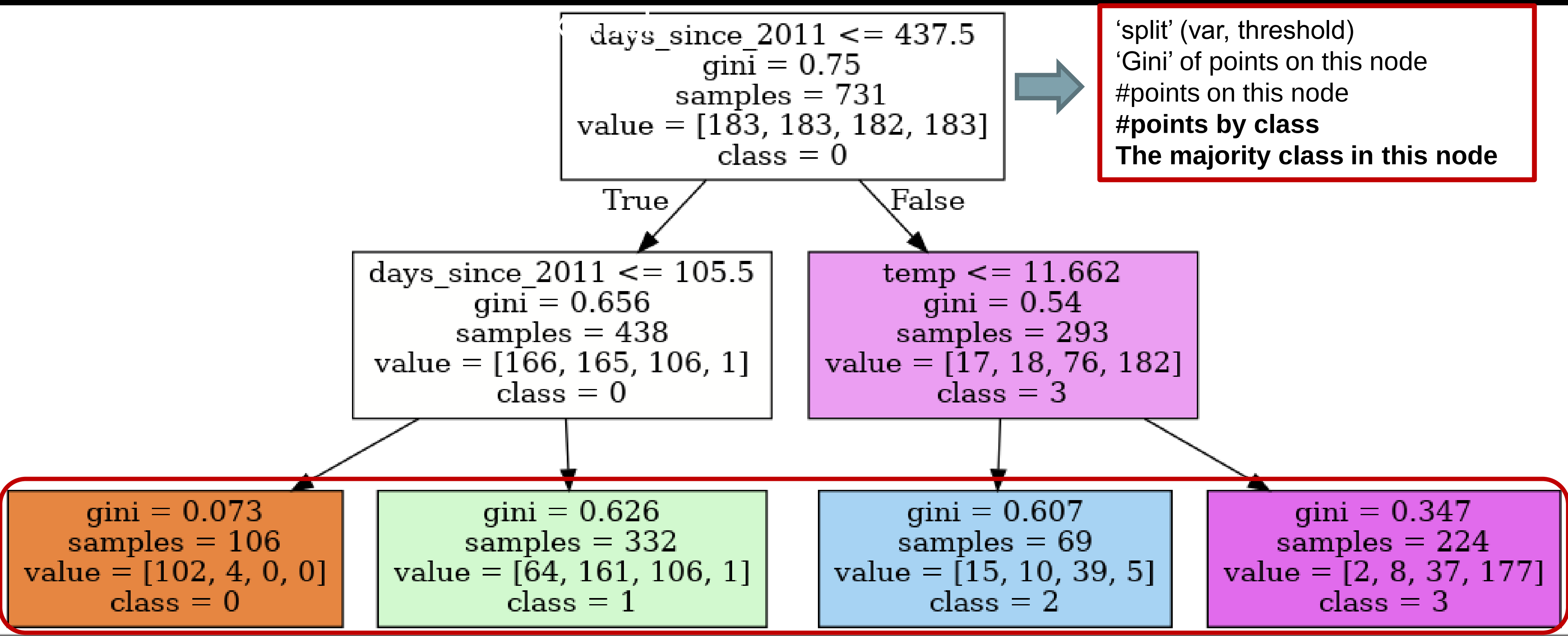
- Transformed into a classification problem using the quantile of bike rental (0,25,50,75,100)

Classification (4 classes)





CART

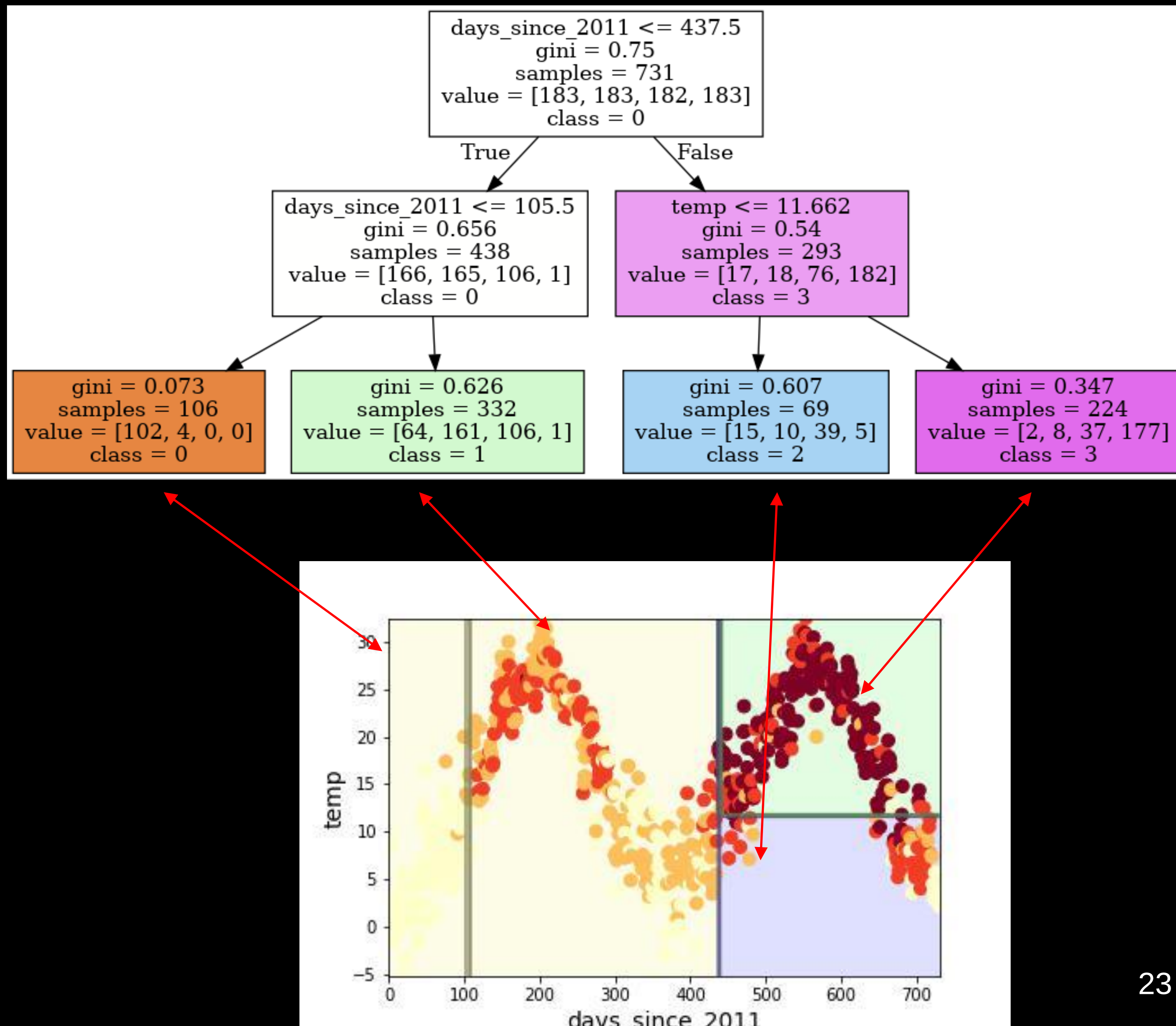


Leaf node. If a data point fall into a leaf, then it is predicted as the 'majority class' of this leaf.

The prediction can be either a label or a prob distribution on four classes

Another visualisation of this CART

CART

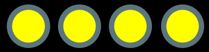


Gini impurity

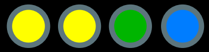
- Choose the best split that maximises the increase of purity (compared to before split), or minimises the decrease of ***Gini impurity***

- ***Gini impurity:*** measures the impurity of a group containing different classes (where p_i is the probability of a class)

$$I_G(p) = \sum_{i=1}^{J} p_i(1 - p_i)$$



- Gini = 0. (if and only if only one class in the set)



- Gini = 0.5*(1-0.5) + 0.25*(1-0.25) + 0.25*(1-0.25) = 0.625

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 Gini impurity)

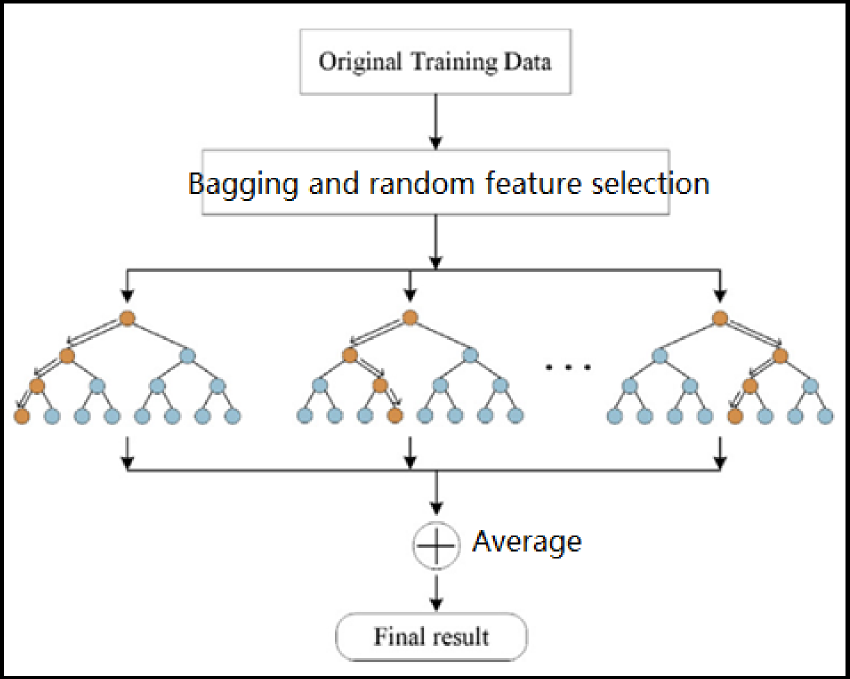
$$J(k, t_k) = \frac{m_{\text{left}}}{m} Gini_{\text{left}} + \frac{m_{\text{right}}}{m} Gini_{\text{right}}$$

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

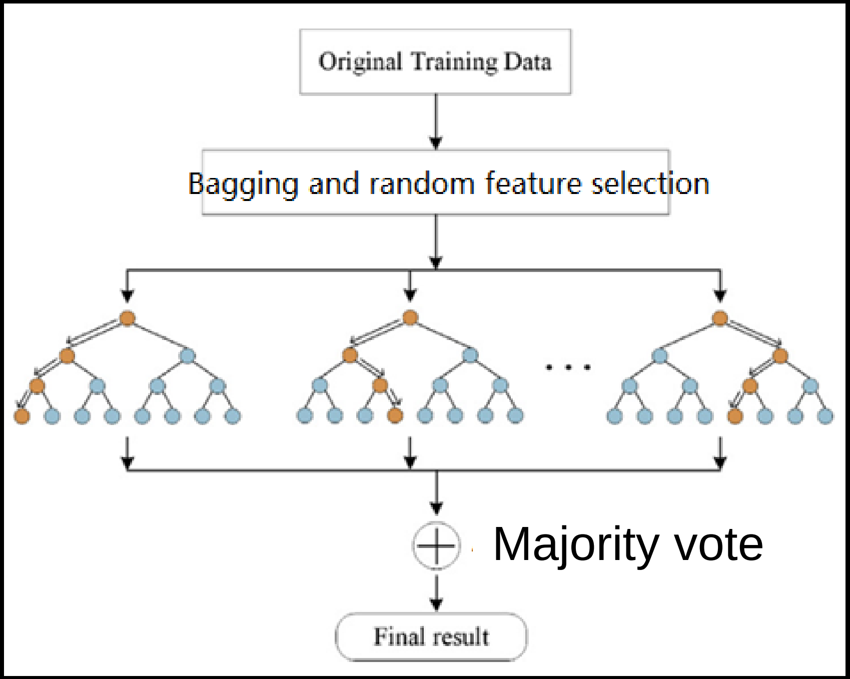
RF and GBDT for classification

RF for classification

Regression



Classification



The output can be a predicted class or a prob distribution over classes (from the vote of the trees)

Logistic regression

Logistic Regression (or logit)

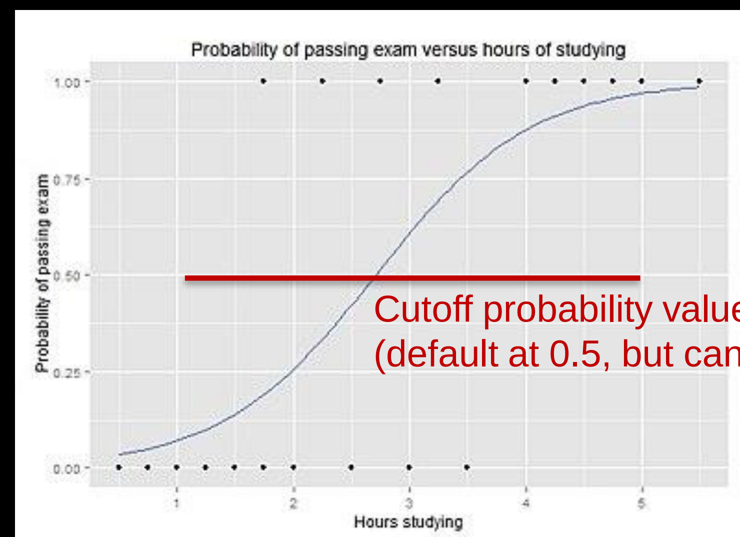
Use a logistic function to model a binary response variable.

The output can be either the probability that this record belongs to a class or the predicted class label

Logistic function: $y = \dfrac{exp(\sum_{i=0}^{n} \beta_i x_i)}{1 + exp(\sum_{i=0}^{n} \beta_i x_i)}$

$\boxed{x} \rightarrow \boxed{y} \rightarrow \boxed{\text{binary class}}$

(-∞,+∞) [0,1] 0 or 1



Logistic Regression (or logit regression)

Pros
Simple; easy to understand; used widely (discrete choice models in econometrics); can model non-linear data relationship

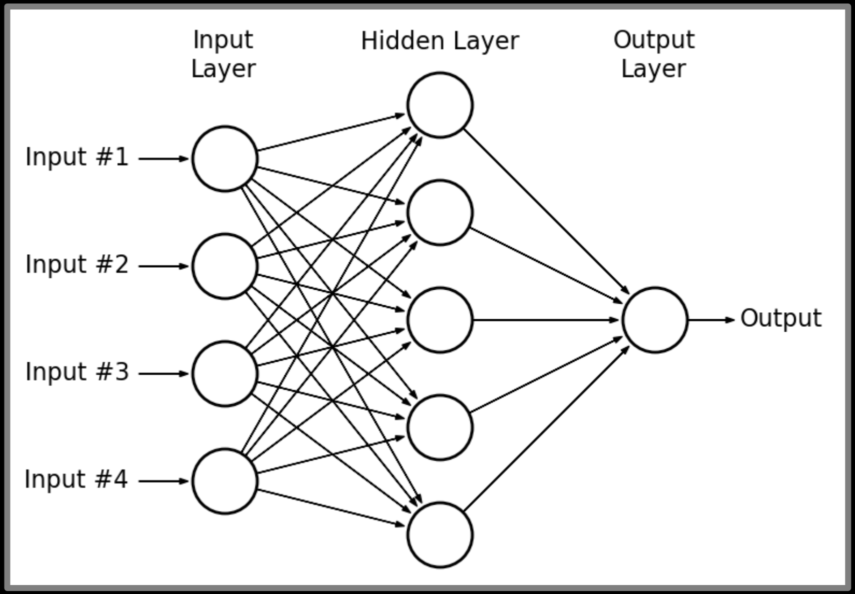
Cons
Subject to variable selection and multicollinearity (like linear regression)

Extensions
- ***Binomial***: fail *vs* pass
- ***Multinomial*** (more than two classes): bus *vs* car *vs* cycling
- ***Ordinal*** (ordered multiple categories): very satisfied *vs* somewhat satistified *vs* neural *vs* somewhat dissatisfied vs very dissatisfied

Artificial neural network

Artificial Neural Networks

- Informal: ANN can be thought of as a multilayer logistic regression
- A logistic regression is the simplest ANN, with no hidden layer.
- Each unit in the NN is called a neuron



Artificial Neural Networks

Calculating value of a neuron: two-step process

1. Weighted sum of neurons in previous layers
2. Non-linear activation function (e.g. logistic)

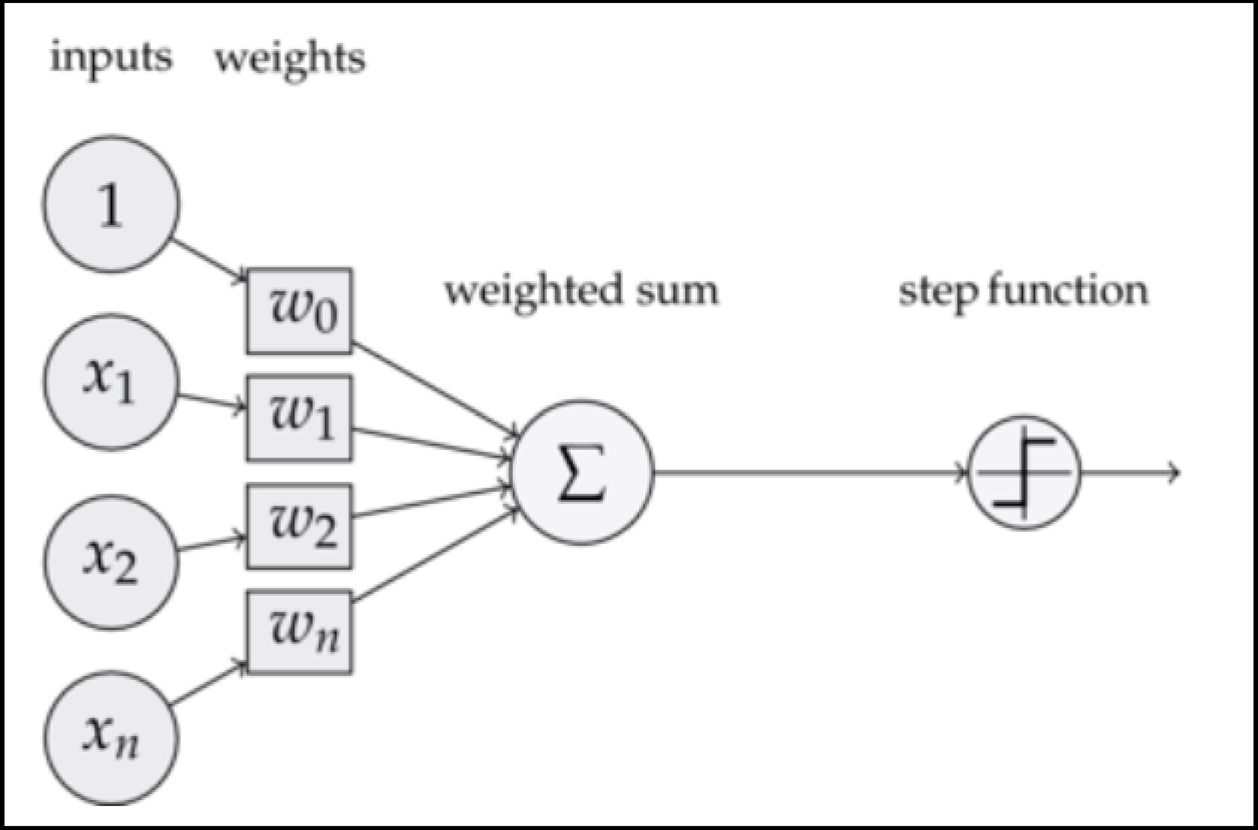


Image Credit

Artificial Neural Networks

Examples of activation function (note that sigmoid is same as logistic)

Step

$$a(z) = \begin{cases} 0, & \text{if } z < 0 \\ 1, & \text{if } z \geq 0 \end{cases}$$

Sigmoid $a(z) = \frac{1}{1+\exp(-z)}$

Hyperboic tangent $a(z) = \tanh(z)$

Rectified linear unit (ReLU) $a(z) = \max(0, z)$

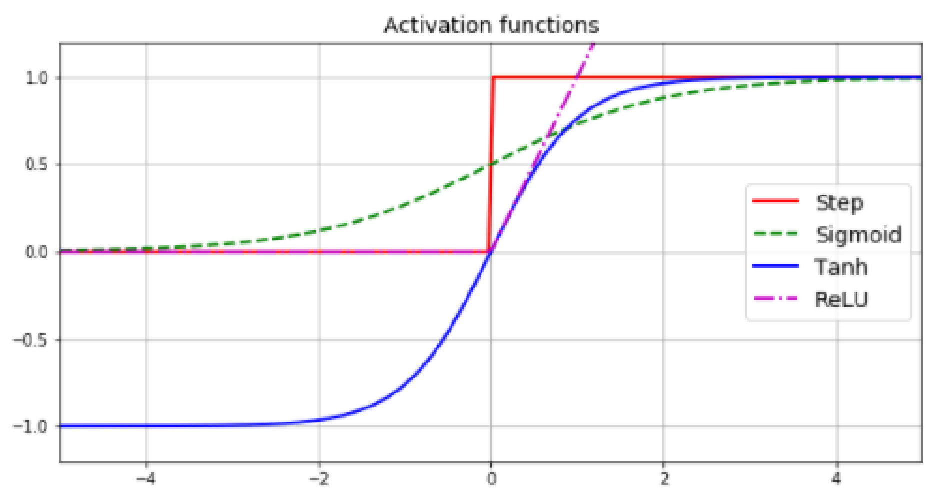
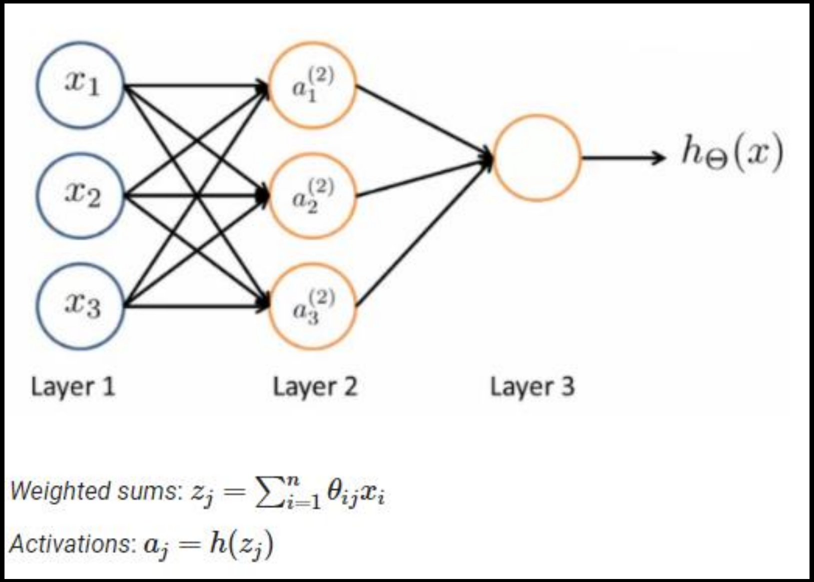


Image credit

Artificial Neural Networks

NNs are constructed by multiple layers of neurons



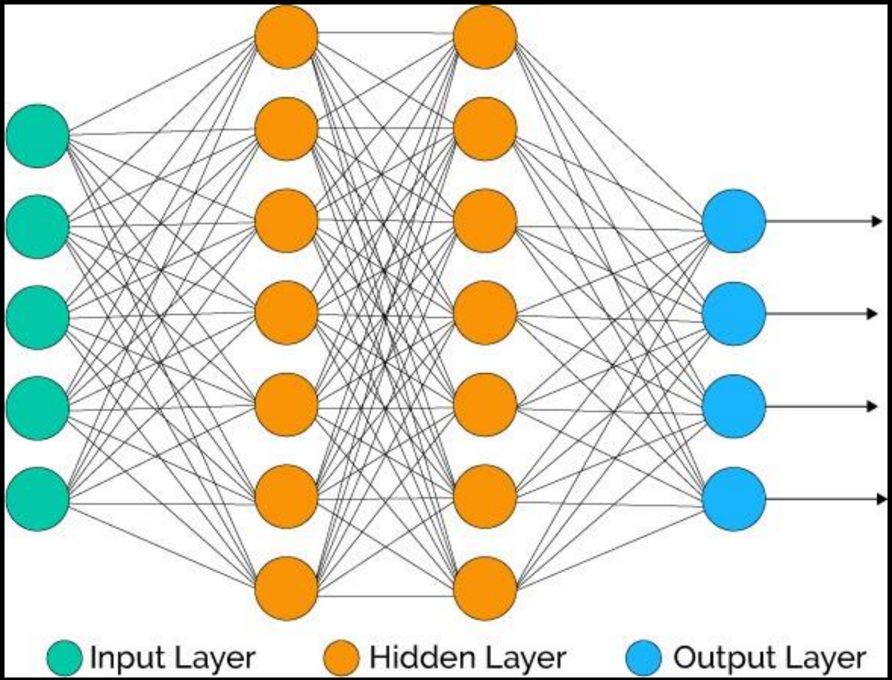
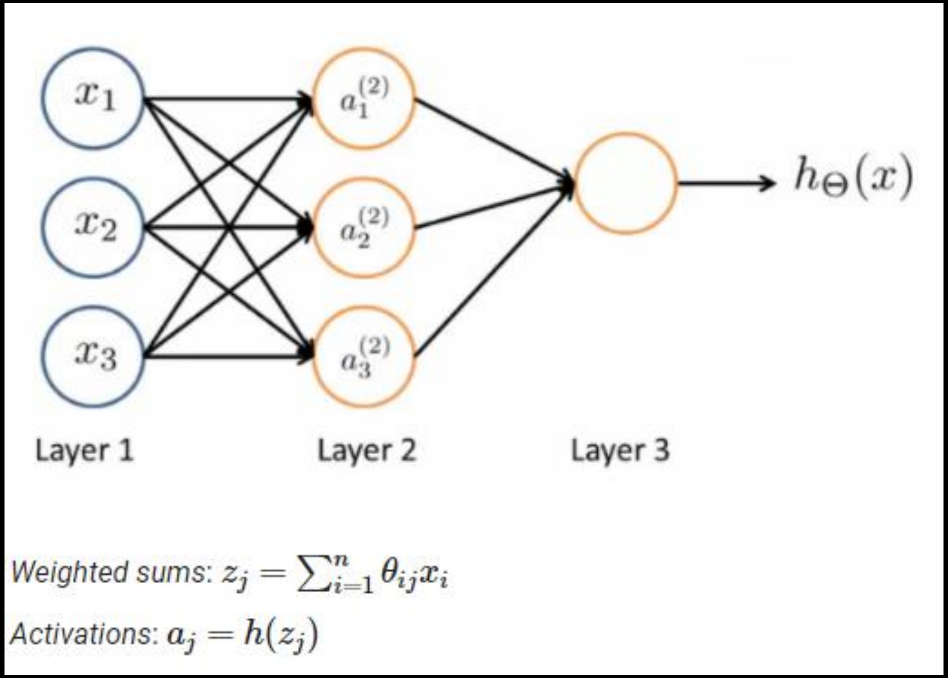


Image credit

Artificial Neural Networks

What are the parameters of ANN, and what are the hyperparameters?



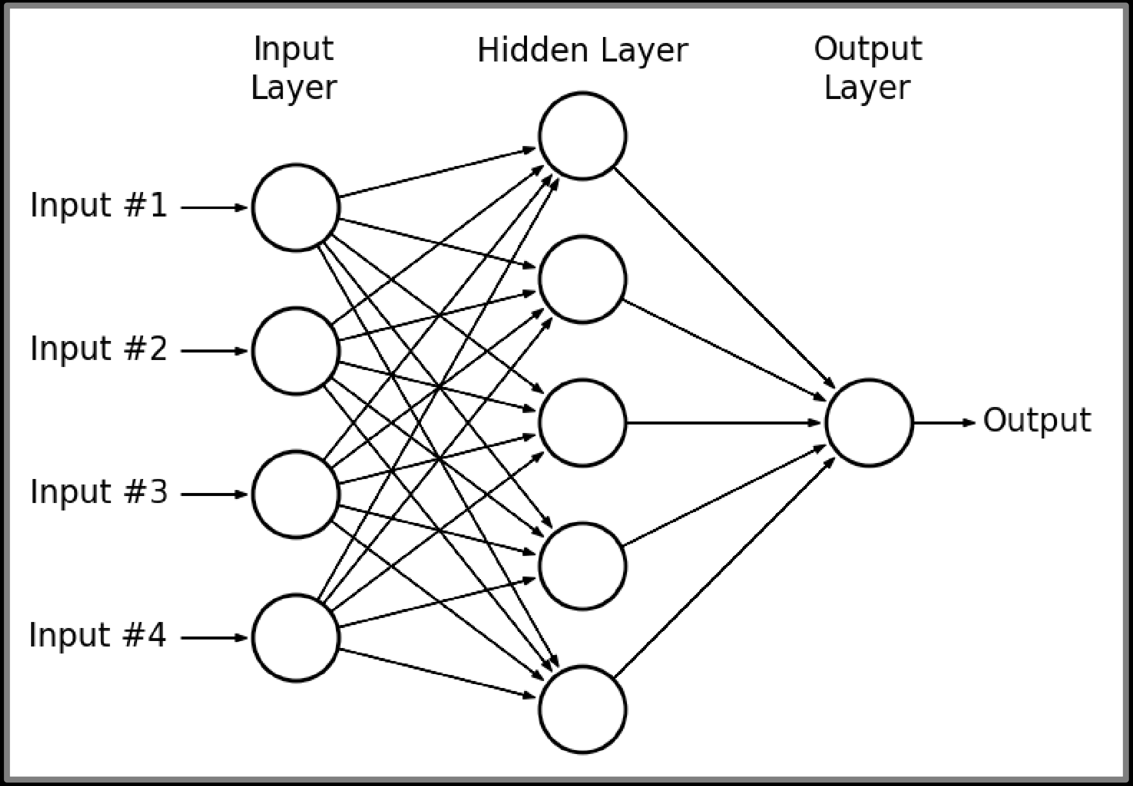
- Hyperparameters
  - # hidden layers
  - # neurons per layer
  - Activation function
  - etc.
- Parameters (learnt during model training)
  - The weights θ_{ij}

Image credit

The power of ANNs

- Universal approximation theorem: a feedforward network can accurately approximate any continuous function from one finite dimensional space to another, **given enough hidden units** (Hornik et al. 1989, Cybenko 1989).
- Therefore, ANNs have the potential to be universal approximators.
- However - universal approximation theorem does not provide any guarantee that training finds this representation. Subject to model tuning and computational power.
- In addition, there are many variants of ANN that are well-suited for different types of data (tableau, image, time series, etc.) without requiring data transformation. This is called end-to-end learning. Will discuss in the lecture of ‘unstructured data’.

Using ANN for regression



Output layer

- Only one unit (corresponding to the y variable)
- No activation function is needed for output layer

Using ANN for classification

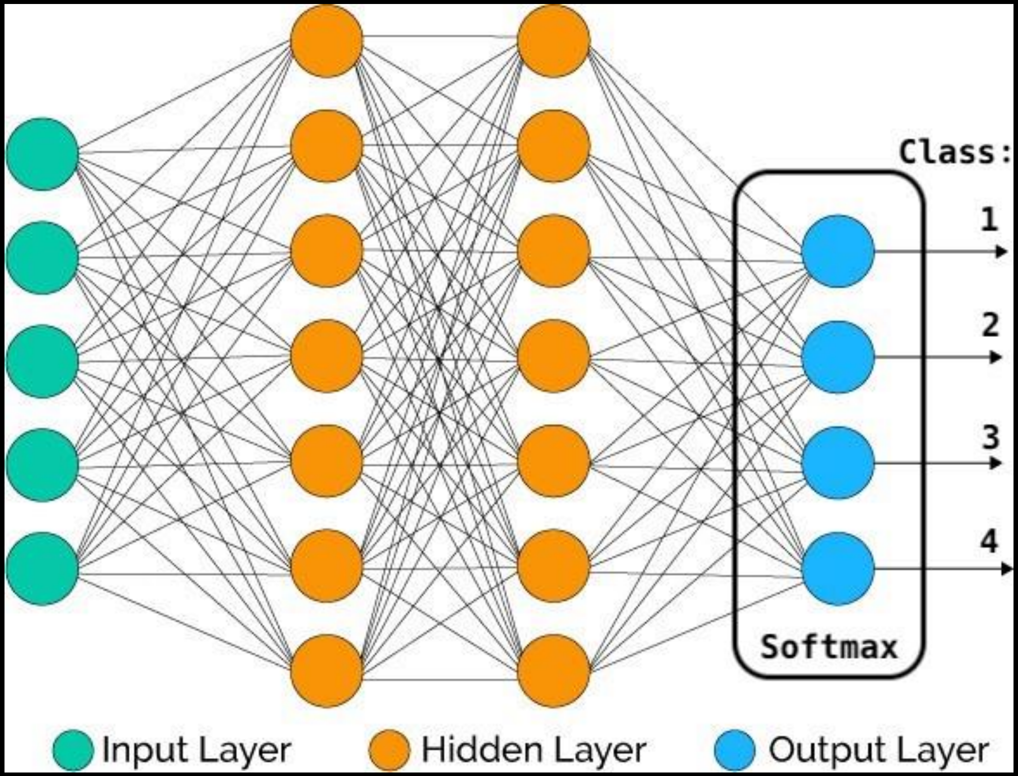


Image credit

Output layer

- # units equals # classes
- The activation function is softmax function (which maps predictions to probability)
- The output is the probability distribution over classes

Softmax function

$$\hat{p}_j = \frac{\exp(a_j)}{\sum_{j'} \exp(a_{j'})}.$$

such that

- $\sum_j \hat{p}_j = 1$
- $0 \leq \hat{p}_j \leq 1$

Summary of regression

- **Classification:** similarity and difference from regression
- **Performance metrics**: accuracy, precision, recall, F1 score
- Classification methods
- CART/RF/GBDT
- Logistic regression
- ANN

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

Papers on travel mode prediction

- Zhao, X., Yan, X., Yu, A., & Van Hentenryck, P. (2020). Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models. *Travel Behaviour and Society*, *20*, 22–35. https://doi.org/10.1016/j.tbs.2020.02.003
- Wang, S., Wang, Q., & Zhao, J. (2020). Deep neural networks for choice analysis: Extracting complete economic information for interpretation. *Transportation Research Part C: Emerging Technologies*, *118*(December 2018), 102701. https://doi.org/10.1016/j.trc.2020.102701
- Wang, S., Mo, B., & Zhao, J. (2020). Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions. *Transportation Research Part C: Emerging Technologies*, *112*, 234–251. https://doi.org/10.1016/j.trc.2020.01.012

Workshop

- This workshop will focus on using classification 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.

Thank You

Questions?

Huanfa Chen

huanfa.chen@ucl.ac.uk