Predictive Modelling Introduction

Dr Liew How Hui

Jan 2022

Definition

Predictive modelling uses “statistics” to build mathematical models which can be used for predicting. (https://en.wikipedia.org/wiki/Predictive_modelling)

Terminologies of similar meaning: **Statistical learning**, **Machine learning**, https://en.wikipedia.org/wiki/Predictive_analytics

Topics (To cover: ✓; Not cover: ✗)

- Supervised Learning Models:
  - Classifiers: kNN ✓, logistic regression ✓, Naive Bayes ✓, LDA ✓, classification trees ✓, Neural Network ✗, ...
  - Regressors: kNN ✓, linear regression ✓ and variations, regression tree ✓, support vector regressor (SVR ✗), ...
- Validation Strategies
- Unsupervised Learning Models:
  - Dimensional Reduction: PCA ✓, ...
  - Clustering: k-Means ✓, HC ✓, ...
  - Anomaly detection ✗
  - Association ✗
  - Autoencoders ✗
- Self-supervised learning ✗
- Reinforcement learning ✗: Value-based, Policy-based, Model-based

May 2020 Final Assessment Q1(b)

Write an essay with no more than 3 pages to **summarise** the various **unsupervised learning models** and **supervised learning models** you learned by using **appropriate mathematical formulation**. Based on what you learned from your assignment and the Internet, suggest **improvements** on this course and propose a good online teaching learning environment. Be warned that non-constructive remarks and insults will receive ZERO mark. (7 marks)

Purpose: You should summarise what you have learned.

Software and Data

Popular data analysis programming languages:

- R + misc libraries: `https://cran.r-project.org/`
- Python (+ Pandas + Sklearn + RPy2): Anaconda Python (`https://www.anaconda.com/products/individual`)
- Java (Weka)
- SQL, NoSQL, etc.

Popular data:

- https://archive.ics.uci.edu/ml/datasets.php
- Kaggle (need registration)

Online Learning Tools

- Microsoft Teams
- WBLE (based on Moodle)
- Lecturer Github Site: `https://liaohaohui.github.io/UECM3993`
- YouTube: E.g. Dr Kilian Weinberger's Machine Learning Lecture, StatQuest, etc.

Reference Books

- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, An Introduction to Statistical Learning: with Applications in R, Springer 2013

  `https://statlearning.com/`
- Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag, 2008

  `https://web.stanford.edu/~hastie/ElemStatLearn/`

Classes & Assessment

The ‘course outcomes’ of this subject are

- CO1: Describe the key concept of statistical learning;
- CO2: Compare statistical models for prediction and estimation through supervised learning;
- CO3: Identify relationship and structures from unlabelled data through unsupervised learning;
- CO4: Demonstrate supervised and unsupervised learning with statistical software;
- CO5: Interpret results from supervised and unsupervised learning.

Classes & Assessment (cont)

Week 1: Practical 1 starts (basic R). There will be 12 practicals. Tutorial 1 starts (11 tutorials). All online???

Week 2: Chinese New Year. No class???

Week 3: Practical 2 (until 12 in Week 13) & Tutorial 2 (until 11 in Week 12) are physical classes.

Week 15: Hari Raya Puasa (finally, there is 1 week break before final exam).

Final Exam (50%): 3 + 1 Questions(?). Tentatively:

- Q1: CO1, Supervised + Unsupervised, 12.5%
- Q2: CO2, Supervised Learning Models, 12.5%
- Q3: CO3, UnSupervised Learning Models, 12.5%
- Q4/Q5: CO5, Supervised + Unsupervised, 12.5%

Classes & Assessment (cont)

Coursework (50%)

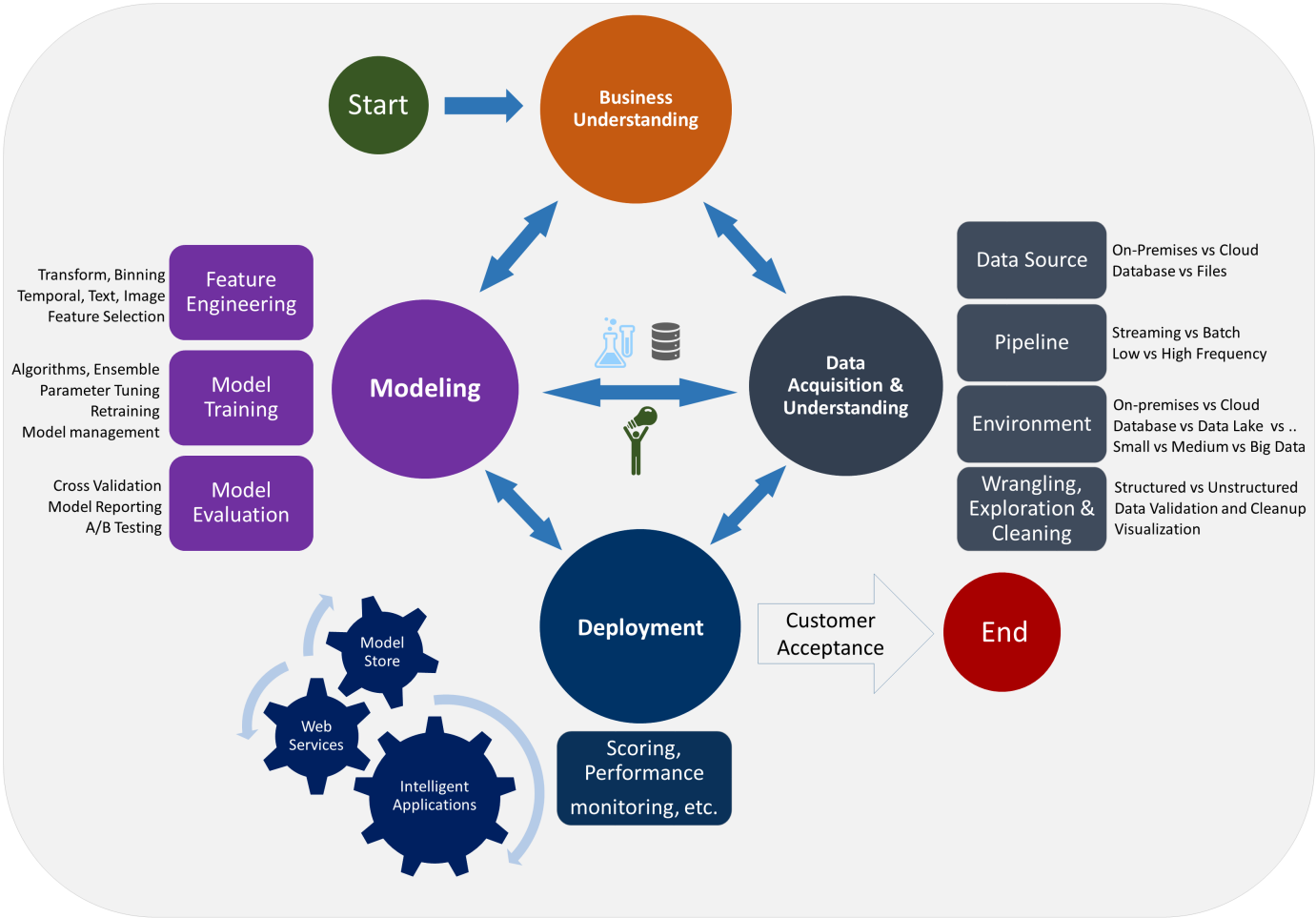
- Physical Practical Quiz (CO4): 12%. **Week 6** Thursday during tutorial class
- Assignment (38%): Report 18% (CO1 + CO2 + CO3) + Programming Code 10% (CO4) + Oral Presentation 10% (CO5, Online, need to include the explanation of the best model's algorithm clearly!!!). Starts from Week 4, ends at Wed Week 11. After that: Oral Presentation.
- Week 1: Start to find assignment group members, 4–7 in a group.

Data Science and CRISP-DM

Data Science / CRISP-DM (Cross Industry Standard Process for Data Mining)

- Business understanding
- Data understanding
- Data preparation / preprocessing
- Modelling
- Evaluation
- Deployment

Microsoft Data Science Lifecycle



https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/lifecycle-business-understanding

Outline

1. Business Understanding
2. Data Understanding
3. Data Preparation / Preprocessing
4. Modelling
5. Evaluation
6. Deployment

Business Understanding

- Define the business goals that the data science techniques can target.
- Find the relevant data that helps you answer the questions

Wrong Direction:

- Many of your seniors apply the predictive models on the data and choose the “best” model forgetting the “goal”: How do the factors influence the target; OR How does the best model help business.

Business Understanding (cont)

Example: Suppose the **goal** is to understand the factors that affect the height of a person.

- Age
- Amount of carbohidrates
- Amount of protein
- Amount of fibre
- Quantity and quality of exercises
- Hours of sleep
- etc.

Business understanding: Which factors are the easiest to perform data collection and more likely to be useful to meet the goal?

Outline

1. Business Understanding
2. **Data Understanding**
3. Data Preparation / Preprocessing
4. Modelling
5. Evaluation
6. Deployment

Data Understanding

Data sources (what lecturer learned from IT visit):

- SQL: Microsoft SQL (most popular), Oracle, PostgresQL, MySQL (used to be very popular in Web mangement), MariaDB, etc.
- SPARQL: https://en.wikipedia.org/wiki/SPARQL, http://spark.apache.org/sql/, https://pypi.org/project/sparkql/
- ...

Whether the real-world data comes from the 'databases' or from Excel files or Internet, they are noisy, i.e. there are missing values, wrong values, etc.

Data Understanding (cont)

To know the data — Exploratory Data Analysis (EDA):

- Data summary (mean, mode, median, etc.)
- Measurement units? E.g. 1 metre or 100 cm?
- Visualisation: histogram, bar plot, etc.
- Data correlation?

To clean up the data:

- Missing values? Wrong values?
- Outliers?

Data Understanding (cont)

Unstructured Data (EDA cannot be used):

- Texts: Reports in Word, PDF; Twitters; etc.
- Images
- Biometric data
- Songs / Lyrics
- Time series: Stock price; Online game control sequence; Industrial robot control sequence; etc.

Observations:

- Most companies usually stored data in structured data format in SQL database using SQL.

Data Understanding (cont)

Structured data (EDA can be used):

- Tabular data containing:
  - Categorical (or nominal) data: Typical example: Gender. R's `factor`; Python's `astype("category")`
  - Ordinal data: Typical example: Student grade. R's `ordered`
  - Numerical data: Typical example: Temperature.
- With some assumptions, we can construct "structured" approximations to unstructured data.

EDA and Data Understanding (cont)

Exploratory data analysis tools for a single data:

- R's `summary`, Python's `describe`
- For (continuous) numerical data: R's `hist` (histogram), `stem` (stem-and-leaf plot)
- For integral data and categorical data, R's `table`.

EDA and Data Understanding (cont)

Exploratory data analysis tools for two data:

- categorical vs categorical: `barplot`, `table`
- categorical vs numerical: `boxplot`
- numerical vs numerical:
  - `cor(x, y, method="pearson")`,
  - (scatter) `plot`

Outline

1. Business Understanding
2. Data Understanding
3. **Data Preparation / Preprocessing**
4. Modelling
5. Evaluation
6. Deployment

Data Preparation / Preprocessing

- Standardisation of datasets — column scaling: `scale(d.f)`
- Normalisation — row scaling
- Non-linear and custom transformation
- Encoding categorical features: E.g. one hot encoding

```
oneh = caret::dummyVars( ~ ., data=df)
final_df = data.frame(predict(oneh, newdata=df))
```

- Discretization: `arules::discretize`
- Imputation of missing values: Can be extremely slow using packages like ‘mice’
- Generating polynomial features: `poly()`

Outline

1. Business Understanding
2. Data Understanding
3. Data Preparation / Preprocessing
4. **Modelling**
5. Evaluation
6. Deployment

Modelling

Modelling means to a mathematical model which fits the observed data:

$$(\boldsymbol{x}_1, y_1),\ (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n).$$

Note that $(\boldsymbol{x}_i, y_i)$ are usually marketing or scientific data obtained through surveys or observations and they are usually stored in a tabular form (with/without missing values which we assumed to have been cleaned).

Note that the $\boldsymbol{x}_i$ are usually called inputs / features / columns / indenpendent variables / etc. and may have more than one components:

$$\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,p}).$$

The y_i are usually called the outputs / targets / response / dependent variable / etc. and is usually one component.

Modelling (cont)

We 'want' to use a function to approximate the data. This leads to

A "Predictive Model" as a "Blackbox"

$$output = model(inputs),$$
$$Y = f(X_1, \cdots, X_p)$$

The function f may be used for

- Prediction: If we have inputs $x_1, \cdots, x_p$ and f, we can "know" a possible output y.
- Inference: Is the model correct? How Y is changing w.r.t X_i? E.g. What factors "improves" sales?

Modelling (cont)

Example (Linear Regression): If the linear regression (a kind of supervised learning method) is used to model the relation between y and x as follows.

y	23.82	47.16	66.66	88.39	110.54
x	1	2	3	4	5
y	131.1	174.15	214.72	233.9	252.14
x	6	8	10	11	12

Predict the value at $x = 7$ using the linear regression model.

[Ans: $\hat{y} = 150.9304$]

Modelling (cont)

The output data y_i can be

- numerical / quantitative: E.g. sales figure
- categorical / qualitative: E.g. success / fail

The "attribute" of the output allows us to classify "models" into

- Regressor: Use to solve regression problems
- Classifier: Use to solve classification problems

Modelling (cont)

The Bayesian approach allows us to classify "models" into

- Discriminative models:

$$\hat{Y} = \text{argmax}_j \, \mathbb{P}(Y = j | X_1, \cdots, X_p)$$

  E.g. linear regression, kNN, logistic model, etc.

- Generative models (only for classifiers?):

$$\hat{Y} = \text{argmax}_j \, \mathbb{P}(X_1, \cdots, X_p | Y = j) \mathbb{P}(Y = j)$$

  E.g. Naive Bayes, LDA, etc.

Modelling (cont)

We usually use a 'family' of models $f_{p_1,\ldots,p_q}$ to approximate the data.

The "number of parameters" of the model allows us to classify "models" into

- Parametric models:
  - Models with **fixed** set of parameters
  - "Training" tries to find the most suitable parameter values to minimise "errors"
- Nonparametric models:
  - Models without fixed set of parameters. Internal "representation" **grows as data increases**.
  - "Training" tries to "fit" the data into the model!

Modelling (cont)

Example: Suppose the output y is governed by the input x following the following equation:

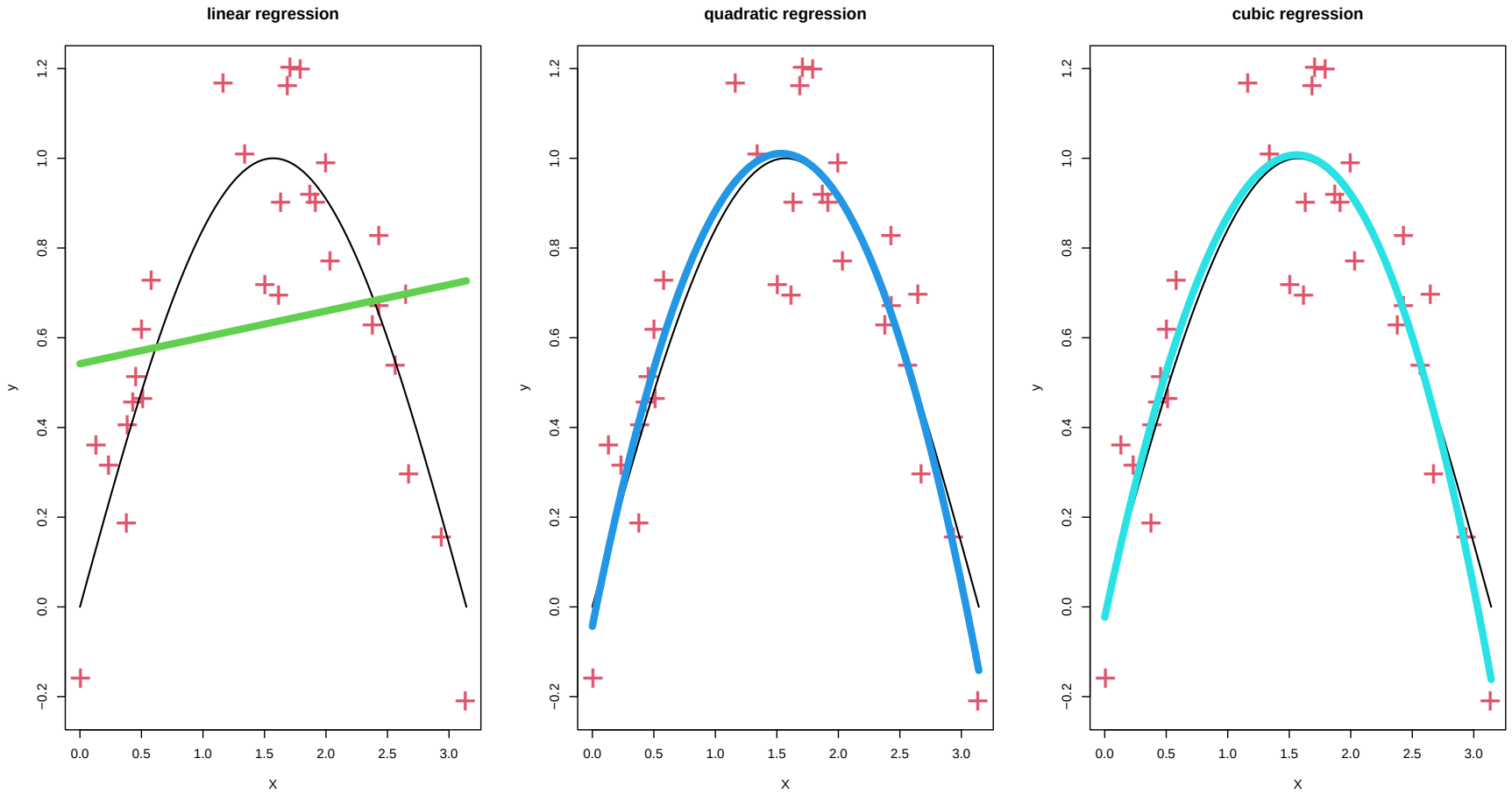
$$y = \sin(x + 2\pi) + R, \quad R \sim Normal(0, 0.2^2)$$

for the range $x \in [0, \pi]$.

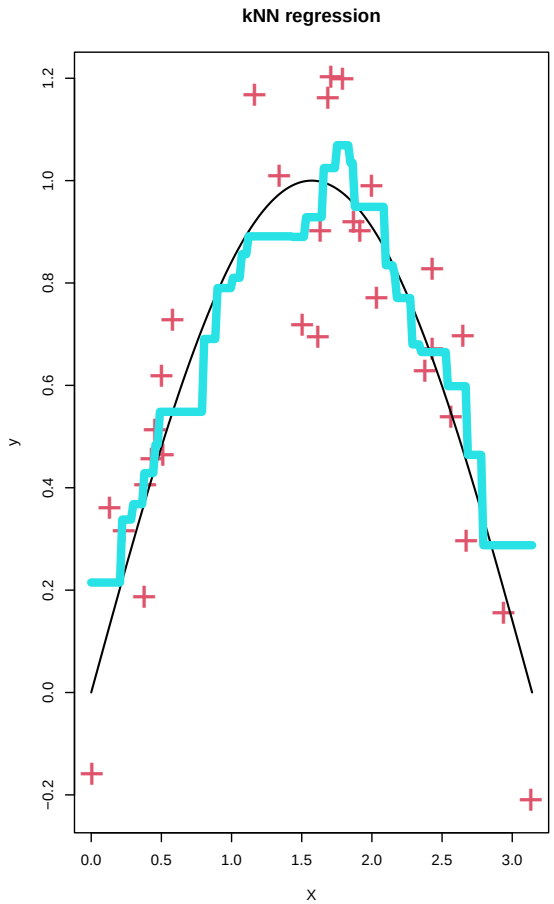
We try the following regression models:

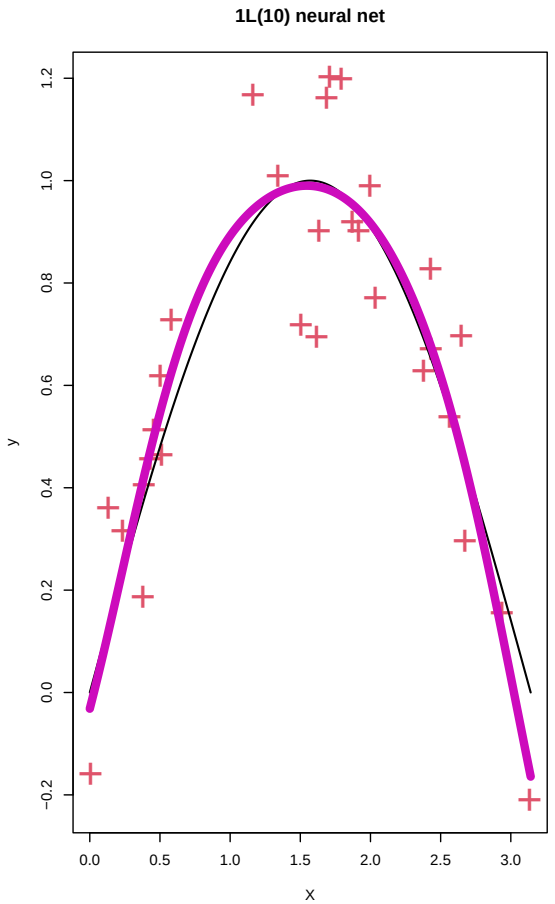
- Linear regression: $y = ax + b + \epsilon$
- Quadratic regression: $y = a_2x^2 + a_1x + b + \epsilon$
- Cubic regression: $y = a_3x^3 + a_2x^2 + a_1x + b + \epsilon$
- kNN (Topic 2)
- Neural Network with 1 hidden layer 10 nodes (= $1 \times 10 + 10 + 10 \times 1 + 1 = 31$) parameters
- Regression tree

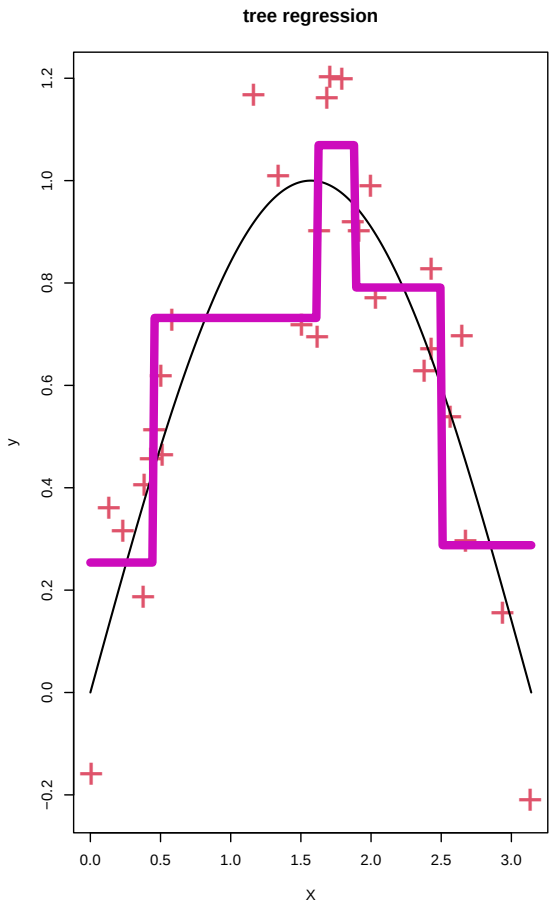
Modelling (cont)



Modelling (cont)







Modelling (cont)

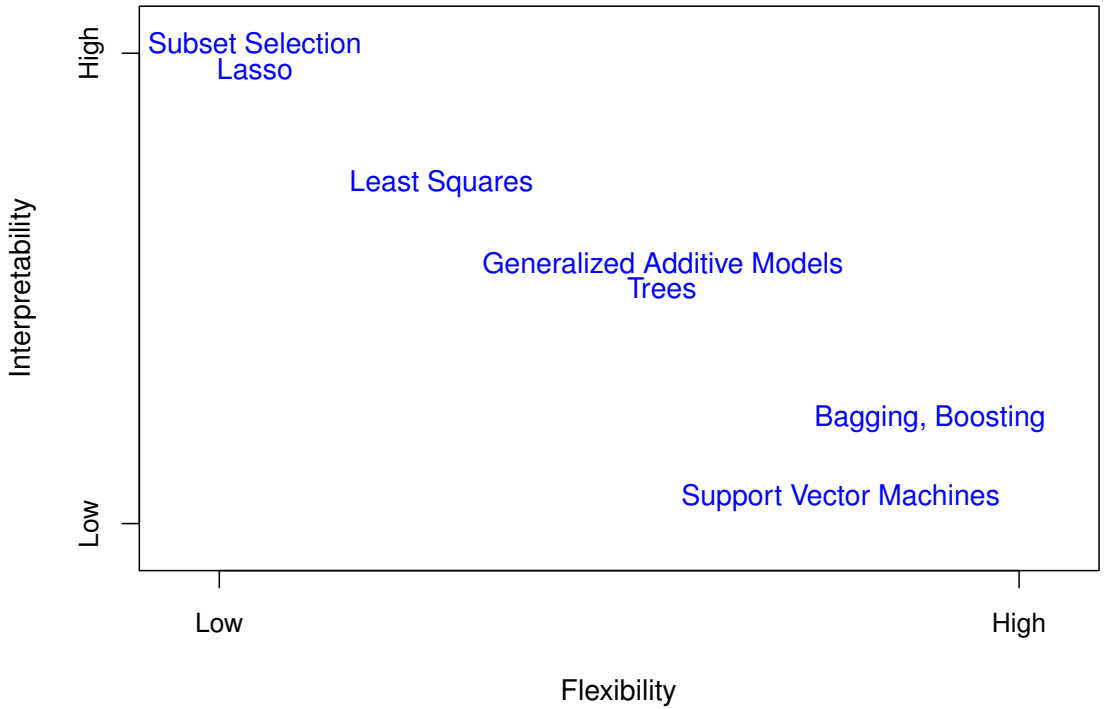
Which model is the best?

Things to consider: Flexibility (more parameters, model more complex) vs Interpretability (less parameters, model simpler)

- Inflexible $\Rightarrow$ Simpler math formula $\Rightarrow$ Poorer Predictibility, better inference(?)
- Flexible $\Rightarrow$ Complicated math formula $\Rightarrow$ Good Predictibility, poorer inference(?)

Modelling (cont)

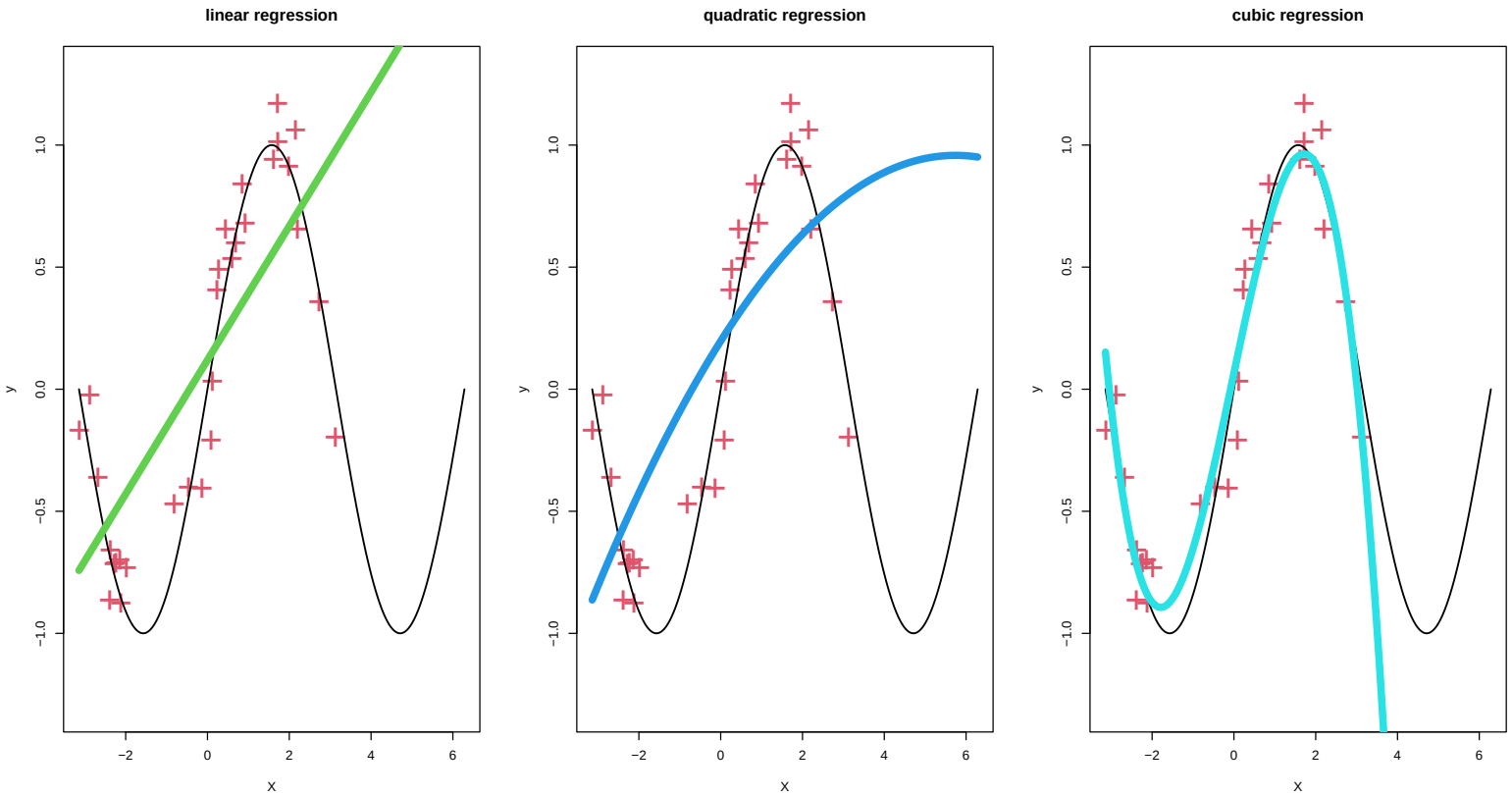
Usually, the interpretability and the flexibility are ‘inverse’ of each other like what the textbook shows:



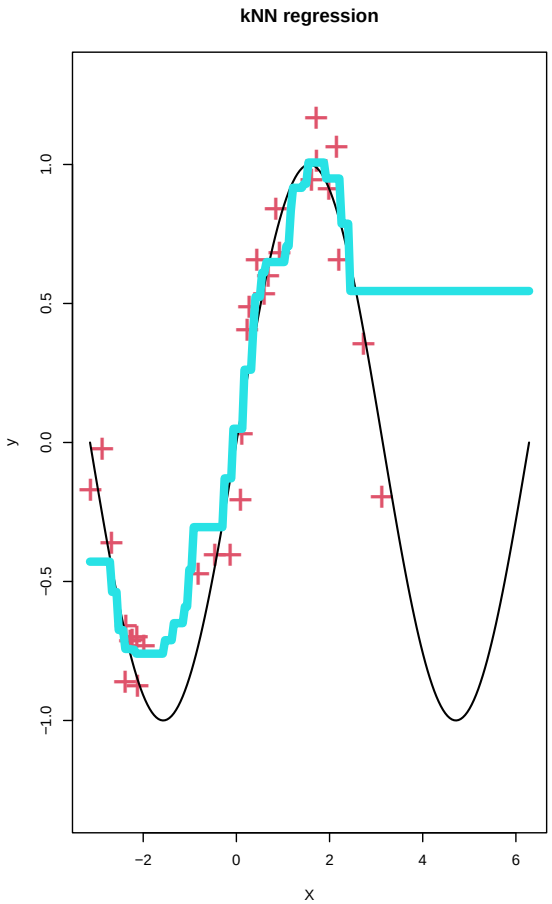
Modelling (cont)

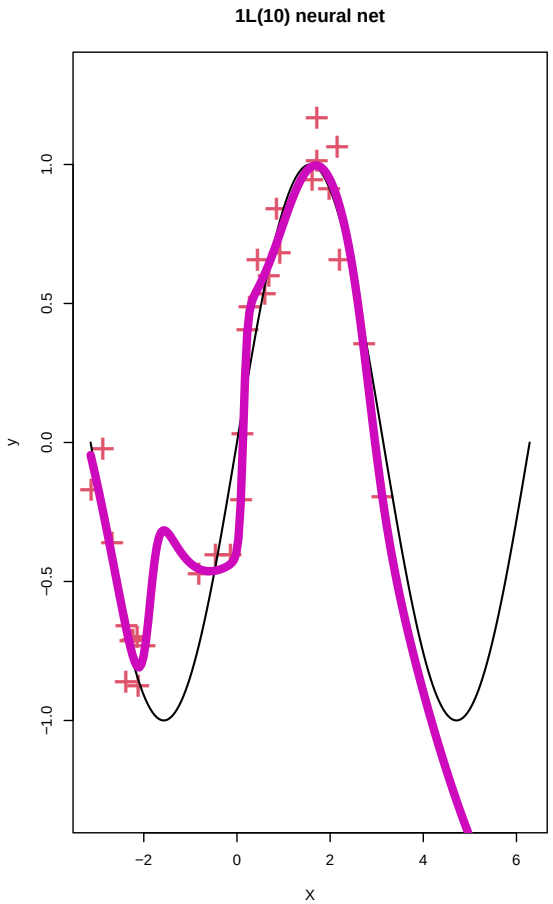
In the previous example, we have looked at the model for the range $x \in [0, \pi]$, now, let us look at $x \in [-\pi, 2\pi]$ with the same formula:

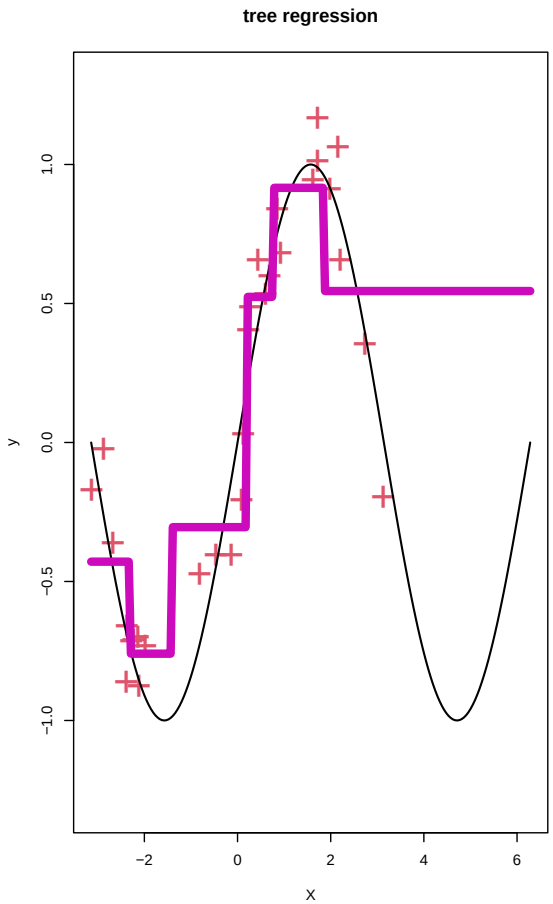
$$y = \sin(x + 2\pi) + R, \quad R \sim Normal(0, 0.2^2).$$



Modelling (cont)



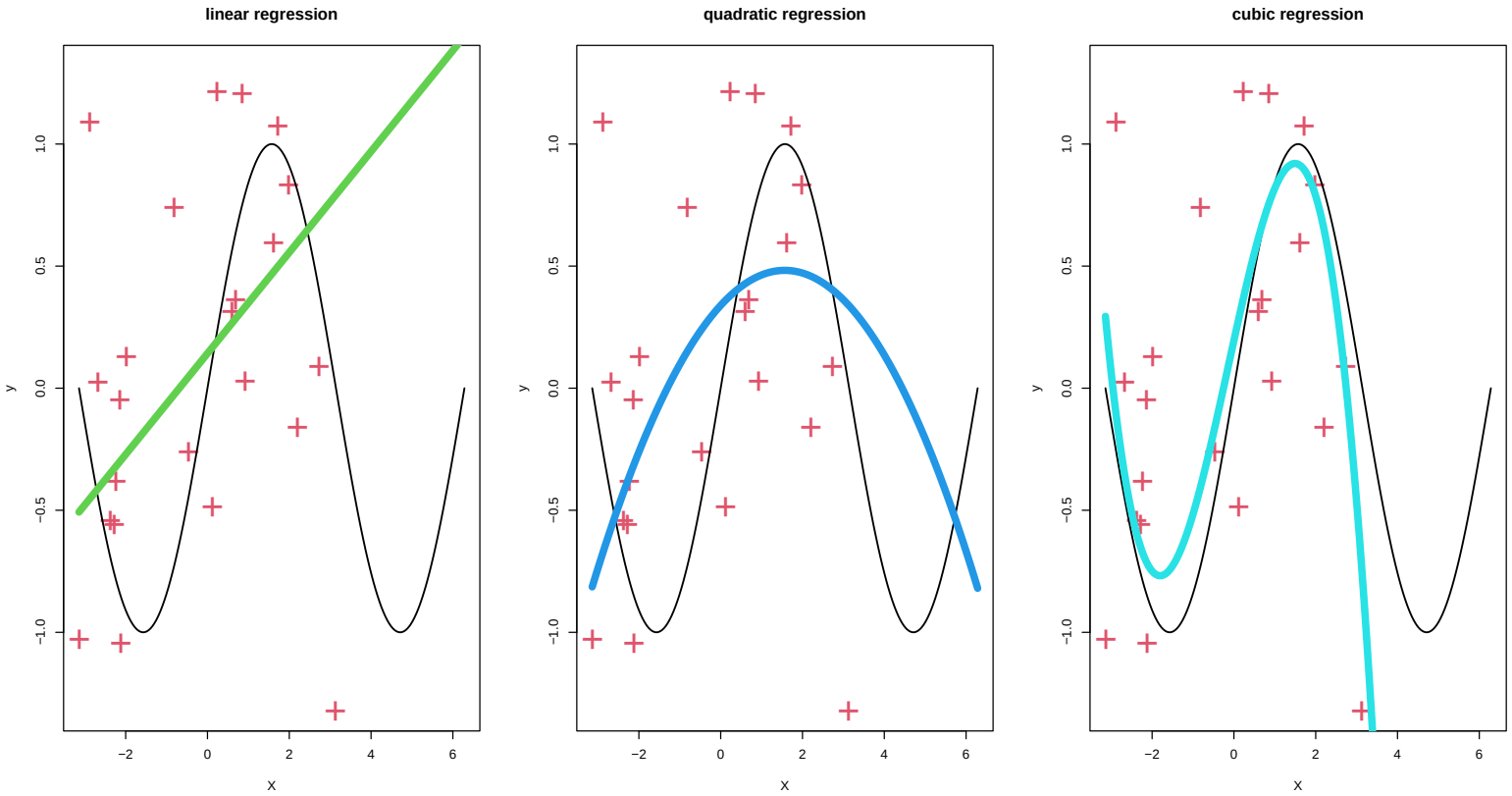




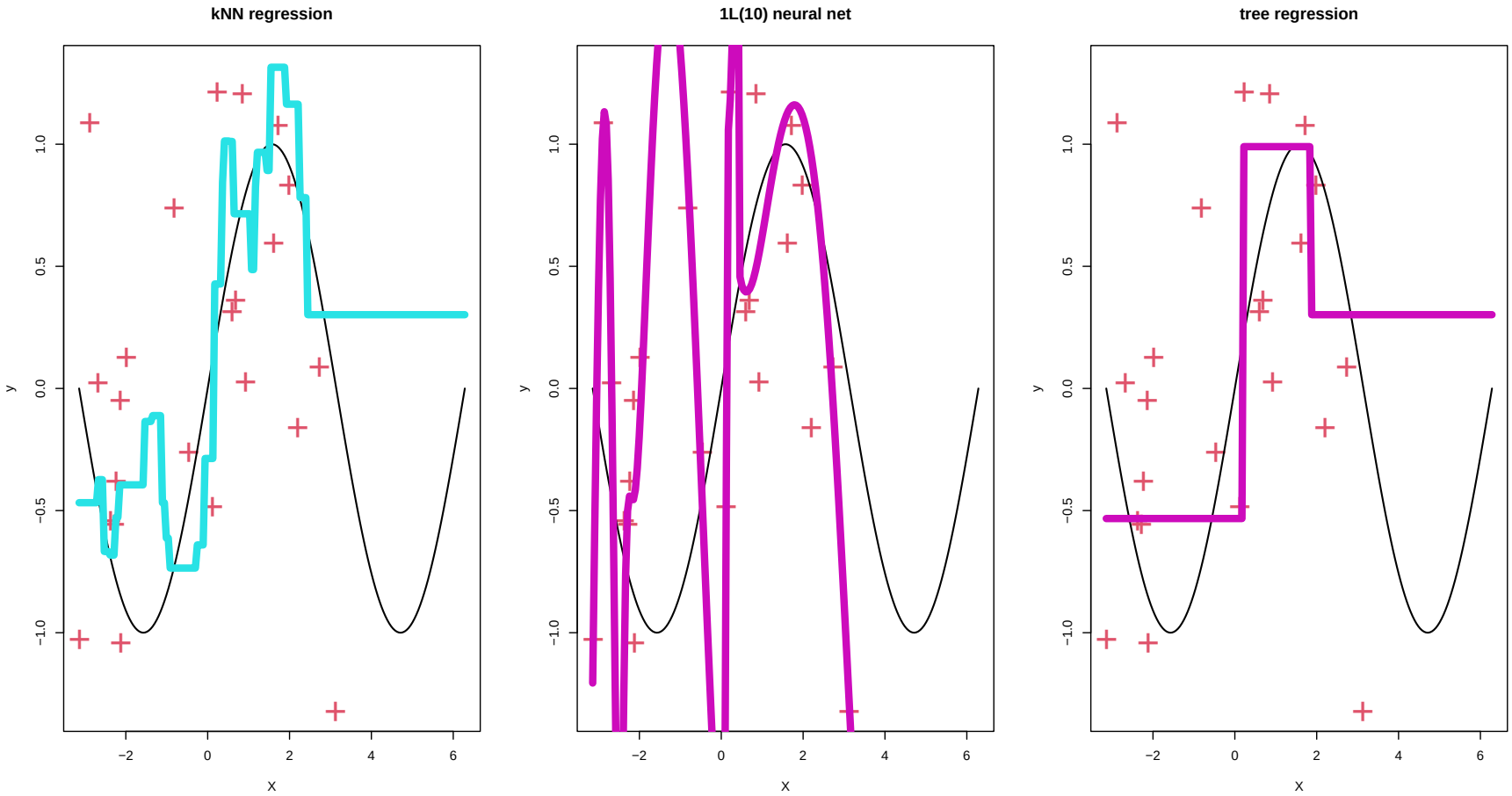
Modelling (cont)

In the previous example, we increase the 'noise' with σ from 0.2 to 1.2:

$$y = \sin(x + 2\pi) + R, \quad R \sim Normal(0, 1.2^2).$$



Modelling (cont)



Neural network is not performing well when the noise is large!

Modelling (cont)

How do we perform modelling in R (and Python)?

Model training and (predict) in R

```
model = lm(y ~ ., data=Xy)
predicted = predict(model, newdata=data.frame(x1=...,x2=...))
```

Model .fit and .predict in Python

```
from sklearn.linear_model import LinearRegression
lrobj = LinearRegression()
model = lrobj.fit(Xy.iloc[:,3:4],Xy.iloc[:,4])
newdata = pd.DataFrame({'x1':..., 'x2':...})
predicted = model.predict(newdata)
```

Outline

1. Business Understanding
2. Data Understanding
3. Data Preparation / Preprocessing
4. Modelling
5. **Evaluation**
6. Deployment

Model Validation / Evaluation

How do you know if the 'trained' model (the coloured lines in the earlier graphs) is **close** the actual function (the black thin line)?

"Loss functions" — measuring the difference between 'predicted values' and 'actual values'.

Regression problem:

- RSS = $\sum_{i=1}^{n}(y_i - f(x_1, \cdots, x_p))^2$
- $R^2 = 1 - \frac{\sum(Y_{actual} - Y_{predicted})^2}{\sum(Y_{actual} - Y_{mean})^2}$

Classification problem:

- Error rate: $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$
- Accuracy: $\frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i)$
- Kappa, ROC, etc.

Model Validation (cont)

Problem with the **theoretical loss function**: If we know the difference between the 'trained model' and the 'actual model'. That means we **know** the 'actual model'. If we know the actual model, why do we need to use the trained model?

In practise, we only have the 'observed data', i.e. $(\boldsymbol{x}_i, y_i)$ and we need to find out what predictive model to use!!!

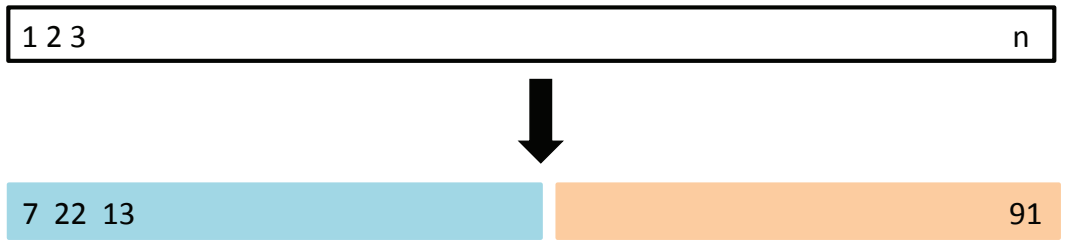
Since we **only** have data and the mathematical models, we need to the data to check if the 'trained' mathematical models are OK.

Model Validation (cont)

Holdout method, **validation set approach**:
Randomly dividing the available set of observations into two parts:

- Training set — to build/fit the model
- Validation/Test set — to test/evaluate the fitted model

Pictorially,



Model Validation (cont)

Holdout method, validation set approach in R:

```
library(datasets)
set.seed(0)
test.index = sample(1:nrow(iris), size=0.4*nrow(iris))
X_y.test  = iris[ test.index, ]
X_y.train = iris[-test.index, ]
library(e1071)
clf = svm(Species ~ ., data = X_y.train, kernel='linear')
predicted = predict(clf, newdata=X_y.test)
conftbl <- table(predicted, X_y.test$Species)
# Accuracy of prediction
sum(diag(conftbl))/sum(conftbl)
```

Model Validation (cont)

Holdout method, validation set approach in Python:

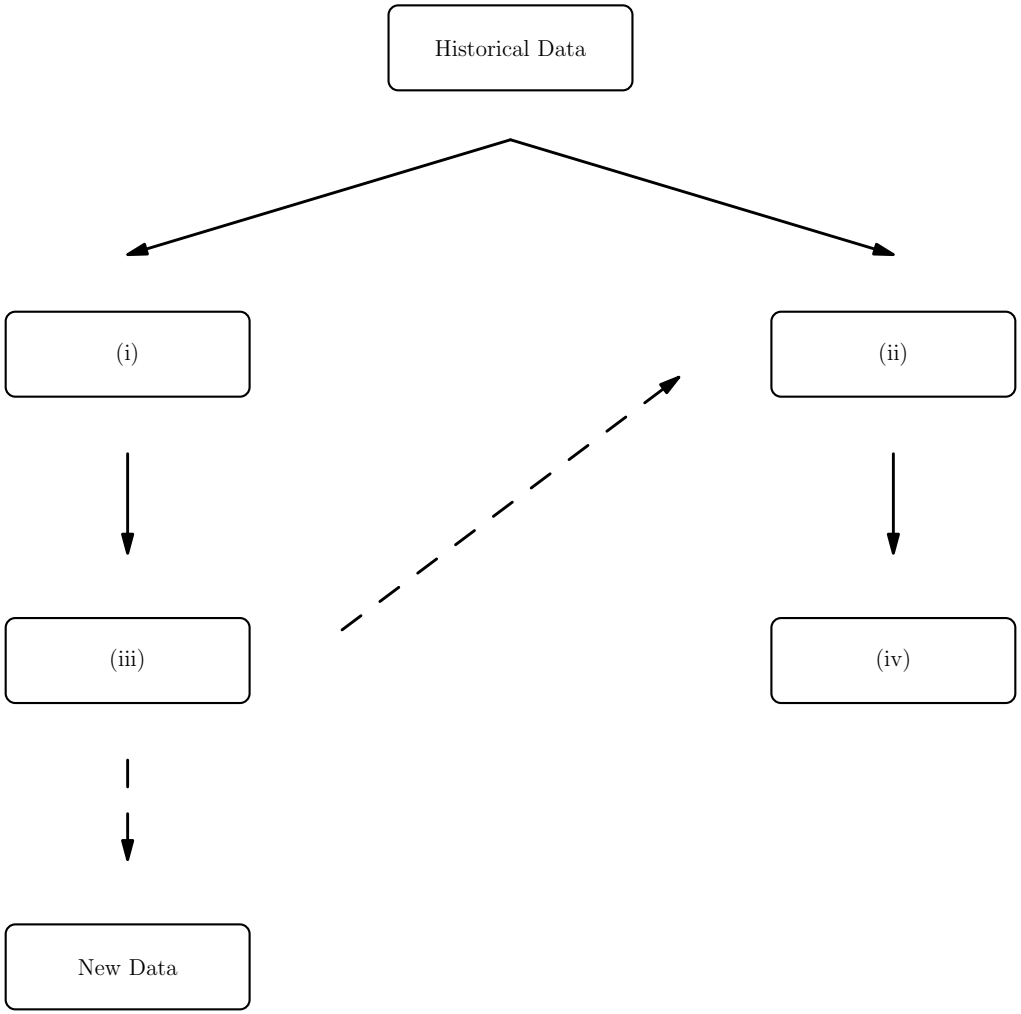
```
from sklearn.model_selection import train_test_split
from sklearn import datasets, svm
X, y = datasets.load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.4, random_state=0)
clf_obj = svm.SVC(kernel='linear', C=1)
clf = clf_obj.fit(X_train, y_train)
# Accuracy of prediction (see Confusion matrix)
clf.score(X_test, y_test)
```

Model Validation (cont)

Example 1.7.1 (Final Exam Jan 2019, Q3)
(a) A predictive model can be built when historical data with known response are presented. The predictive model is then used to predict the response of a new data set with predictors given. Figure Q3(a) shows the process to form a predictive model.
Fill in the blanks (i) to (iv) in Figure Q3(a). State the differences between regression and classification for each step in the process of forming a predictive model.
(b) Give three examples on how statistical learning can help in risk/fraud analytics.

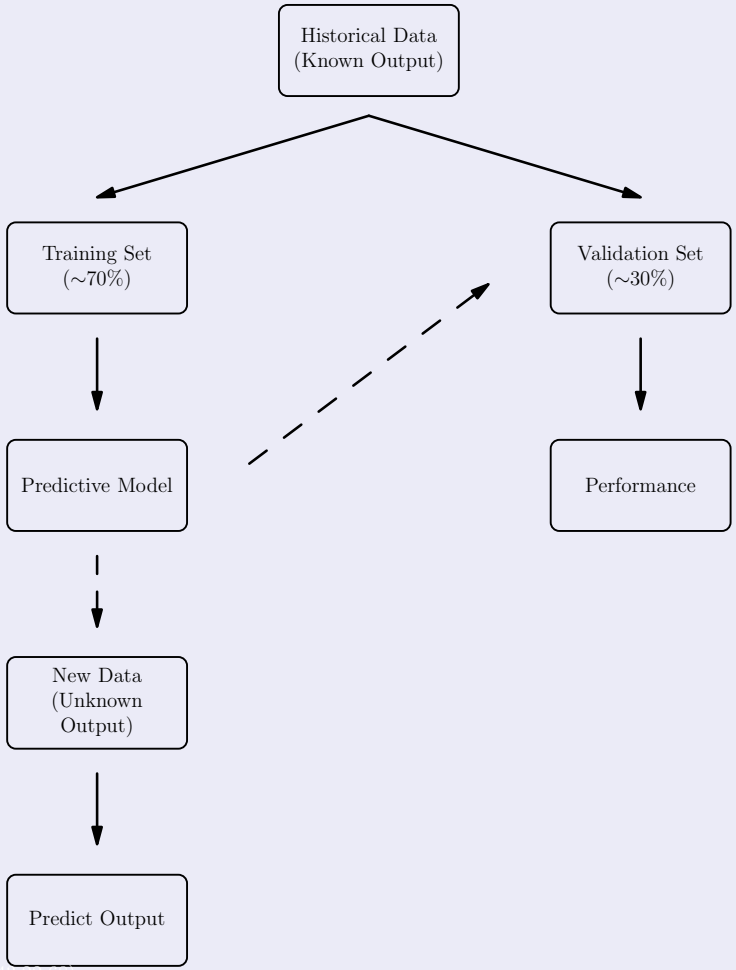
Model Validation (cont)

Figure Q3(a)



Model Validation (cont)

Example 1.7.1 (a) Answer



(54.43,22.68)

Regression vs. classification

	Regression	Classification
Response	Numerical	Categorical
Split	Linear sampling	Stratified sampling
Performance	RSS, R^2	Confusion matrix
Scoring	$\hat{y}(\boldsymbol{x}) \pm$ s.d.	$\mathbb{P}(Y = j \mid \boldsymbol{X} = \boldsymbol{x})$

(b) Answer

- Banking industry uses credit scores to decide if an applicant can get a loan.
- Insurance industry predicts changes of an event to calculate premium.
- Financial institutions predicts frauds in transactions

May 2020 Final Assessment Q1(a)

Describe the differences between regression and classification for each step in the process of forming a predictive model with appropriate justifications.

(3 marks)

Answer

	Regression	Classification
Response	Numerical	Categorical
Split	Linear sampling	Stratified sampling
Performance	RSS, R^2	Confusion matrix
Scoring	$\hat{y}(\boldsymbol{x})\pm$ s.d.	$\mathbb{P}(Y = j\|\boldsymbol{X} = \boldsymbol{x})$

. . . . . [3 marks]

Confusion Matrix

For a classification problem with binary outcomes (only 2 classes), positive (+) and negative (-), the confusion matrix / contingency table can be presented as follows.

		Actual observations		
		Positive (+)	Negative (-)	**Precision**
Predicted	Positive (+)	True Positive Count (TP)	False Positive Count (FP)	Positive Predictive Value (PPV)
	Negative (-)	False Negative Count (FN)	True Negative Count (TN)	Negative Predictive Value (NPV)
	Recall	True Positive Rate (TPR) (Sensitivity)	True Negative Rate (TNR) (Specificity)	Accuracy (ACR)

Confusion Matrix (cont)

Example 1.7.2: A predictive model has been built by using the training set. After predicting the outcome (fraud, not fraud) by implementing the model into validation set, the results are recorded as follow:

- Numbers of customer predicted to be fraud and the prediction is correct = 70
- Numbers of customer predicted to be fraud and the prediction is incorrect = 30
- Numbers of customer predicted not to be fraud and the prediction is correct = 80
- Numbers of customer predicted not to be fraud and the prediction is incorrect = 20

Confusion Matrix (cont)

		True Class	
		Fraud (+)	Not Fraud (-)
Predicted Class	Fraud (+)	70 (TP)	30 (FP)
	Not Fraud (-)	20 (FN)	80 (TN)

Calculate the accuracy measures sensitivity, specificity, PPV, NPV, ACR, FPR, FNR.

Confusion Matrix (cont)

```
lvs   <- c("not fraud", "fraud")  # -> class 1, 2
lvs.r <- c("fraud", "not fraud")  # Show fraud first
truth <- factor(rep(lvs, times=c(110, 90)),
                levels=lvs.r)
pred  <- factor(c(rep(lvs, times=c(80, 30)),
                  rep(lvs, times=c(20, 70))),
                levels=lvs.r)
xtab  <- table(pred, truth)
TPR = xtab[1,1]/sum(xtab[,1])  # Sensitivity
TNR = xtab[2,2]/sum(xtab[,2])  # Specificity
PPV = xtab[1,1]/sum(xtab[1,])
NPV = xtab[2,2]/sum(xtab[2,])
FPR = 1 - TNR
FNR = 1 - TPR
```

Decision Making Example

You have just landed a great analytic job with ACME Inc., one of the largest telecommunication firms in United States. They are having a major problem with customer retention in their wireless business. In the Mid-Atlantic region, 20% of cell phone customers leave when their contracts expire, and it is getting increasingly difficult to acquire new customers.
Since the cell phone market is now saturated, the huge growth in the wireless market has tapered off. Communications companies are now engaged in battles to attract each other's customers while retaining their own. Customers switching from one company to another is called churn.

Decision Making Example (cont)

You and your team have been called in to help understand the problem and to devise a solution. The data set given consists of 2,000,000 observations with 10 predictors. Your team decided to build several predictive models using different methods. The models are then tested with the validation data set. The results of testing are shown as below:

Model	TP	FP	TN	FN
3-Nearest Neighbour	281,609	31,291	263,077	24,023
500-Nearest Neighbour	181,301	51,014	243,354	124,331
LR (all predictors)	243,344	55,194	239,174	62,288
LR (significant predictors)	249,487	61,493	232,875	56,145

Based on the information given, make a decision on the model that is suitable for churn prediction to be proposed to the company. Discuss on your decision. . . . . **Online Class Discussion?**

Confusion Matrix (cont)

Multiclass Example 1.7.4: Use kNN with $k = 5$ (a kind of predictive model) to train on 75% of the iris data and then construct the confusion matrix on the remainder 25% iris data (as test data).

```
library(datasets)
data(iris)
N = nrow(iris)
set.seed(59) #set.seed(9)
train_idx = sample(N, size=0.75*N)
iris_train = iris[ train_idx,]
iris_test  = iris[-train_idx,]

library(class)  # for knn
M = ncol(iris)
iris_predict = knn(train=iris_train[,-M], test=iris_test[,-M],
                   cl=iris_train[,M], k=5)
library(gmodels) # dependencies: gtools, gdata
CrossTable(x=iris_predict, y=iris_test[,M], prop.chisq=FALSE)
```

Confusion Matrix (cont)

	iris_test[, M]			
iris_predict	setosa	versicolor	virginica	Row Total
setosa	14	0	0	14
	1.000	0.000	0.000	0.368
	1.000	0.000	0.000	
	0.368	0.000	0.000	
versicolor	0	10	1	11
	0.000	0.909	0.091	0.289
	0.000	0.909	0.077	
	0.000	0.263	0.026	
virginica	0	1	12	13
	0.000	0.077	0.923	0.342
	0.000	0.091	0.923	
	0.000	0.026	0.316	
Column Total	14	11	13	38
	0.368	0.289	0.342	

ROC Curve

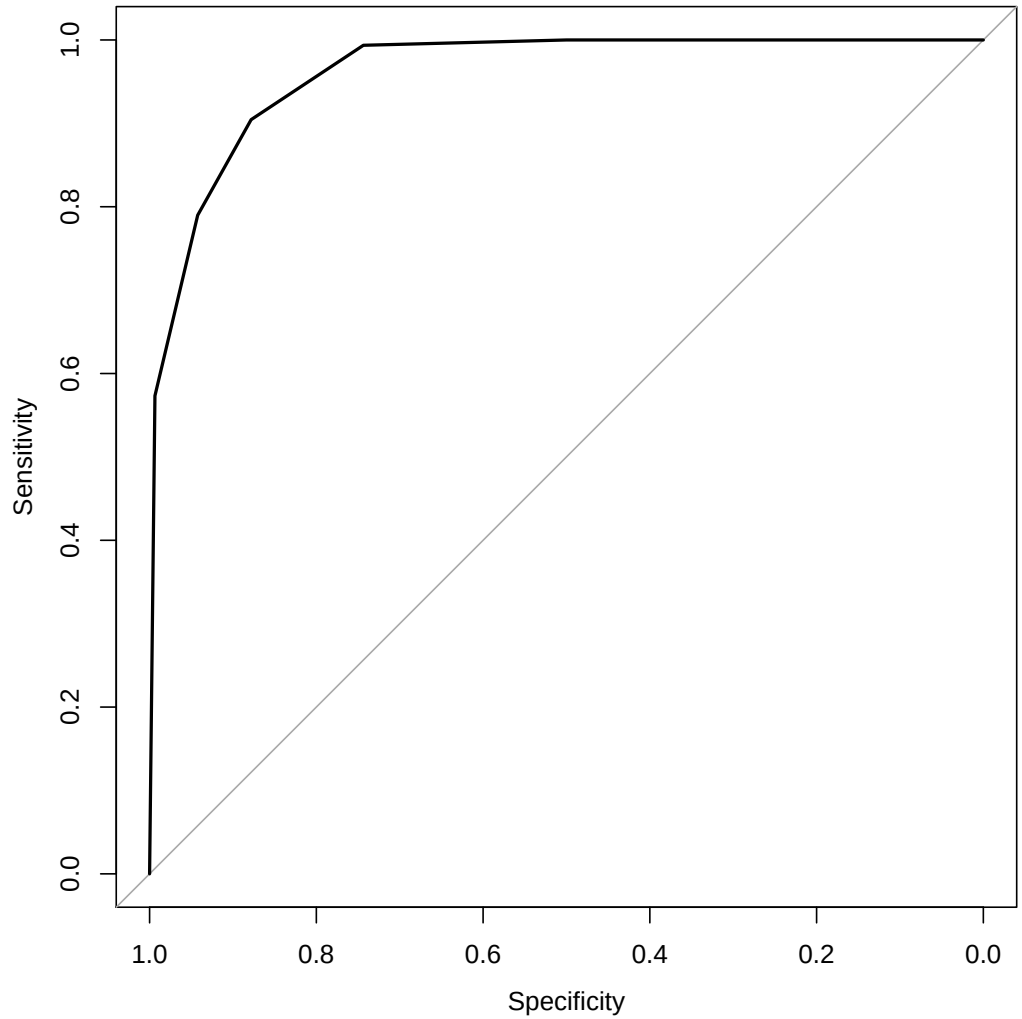
Receiver Operating Characteristic (ROC) curve = "sensitivity" vs "1−specificity" ("TPR vs FPR")
Only for **binary** classifiers.

```
library(ISLR)
Smarket = Smarket[,-1]  # Remove Year
N = nrow(Smarket)

set.seed(59) #set.seed(9)
train_idx = sample(seq(N), size=0.75*N)
Smarket_train = Smarket[ train_idx,]
Smarket_test  = Smarket[-train_idx,]

library(class)  # for knn
M = ncol(Smarket)
Smarket_predict = knn(train=Smarket_train[,-M], test=Smarket_test[,-M],
                cl=Smarket_train[,M], k=5, prob=TRUE)
suppressWarnings(library(pROC))
prob = attr(Smarket_predict,"prob")
prob = ifelse(Smarket_predict=="Up", prob, 1-prob)
proc.obj = roc(Smarket_test[,M], prob, plot=TRUE)
```

ROC Curve (cont)



Note that R's ROC curve sensitivity is 1 to 0 while the theoretical version is FPR(=1−sensitivity), which is 0 to 1.

Unsupervised Learning

When the data has no label or we want to know the distribution of the input and output data, we will be applying the unsupervised learning methods:

- Descriptive Statistics / EDA
- Visualisation → Dashboard
- Clustering (for small dimension?) E.g. k-means, HC
- Dimensionality Reduction (for large dimension?). E.g. PCA
- `https://en.wikipedia.org/wiki/Association_rule_learning`
- `https://en.wikipedia.org/wiki/Anomaly_detection`

Outline

1. Business Understanding
2. Data Understanding
3. Data Preparation / Preprocessing
4. Modelling
5. Evaluation
6. **Deployment**

Deployment

Real-world deployment:

- Spam filtering for emails & SMS
- Intrusion detection
- ???

Designing dashboards:

- Tableau:
- Qlikview:
- Power BI:
- D3.js
- ...

Use Cases

Classification:

- Email → spam / non-spam;
- Tumour → benign / malignant;
- Writing → characters / words;
- Image → label;
- Activity → fraud / non-fraud

Use Cases (cont)

Regression:

- Predicting insurance premiums
- Motor insurance pricing using GLM
- Econometrics
- Signal processing in sensors — industrial process tomography ?

Use Cases (cont)

Unsupervised learning:

- clustering COVID-19 viruses and other coronavirus
- market segmentation
- visualisation
- dimensionality reduction

Real-World Concern

Pros & Cons of AI.

E.g. AI in changing background

- Pro: Video production
- Con: Fake scene: A cooking background can be turned to a murder background.

E.g. AI in removing someone from the scene

- Pro: Video production, e.g. removing “tourists” from a documentary scene.
- Con: “Fake” CCTV.