Society of Actuaries in Ireland

Demystifying Data Science

19th September 2018

Disclaimer

The views expressed in these presentations are those of the presenter(s) and not necessarily of the Society of Actuaries in Ireland

Welcome

- Pedro Ecija Serrano

  Chair, Data Analytics Subcommittee
- First of a series of three presentations

Disclaimer:
The material, content and views in the following presentation are those of the presenter(s).

Demystifying Data Science

- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples

What is Data Science?

“Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to **extract knowledge and insights from data** in various forms”

—Wikipedia

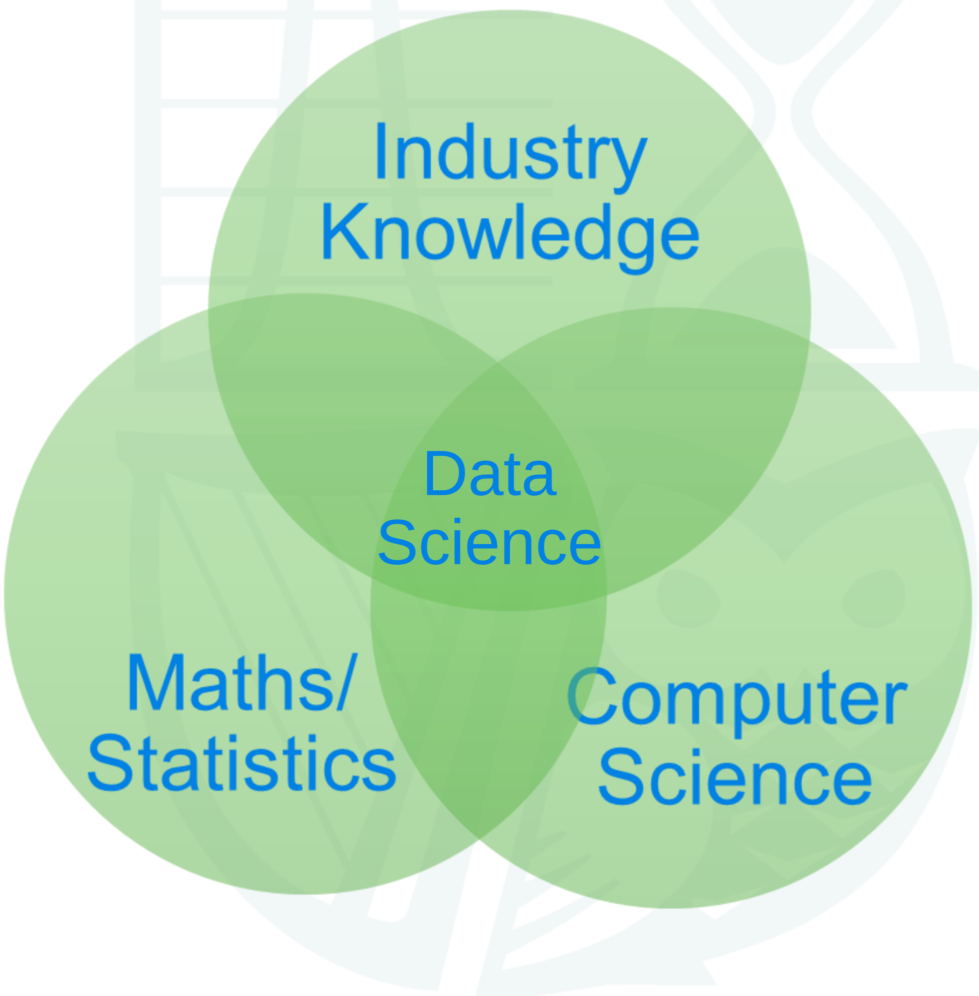
What is Data Science?

“Data science is the study of how to make data-driven decisions”

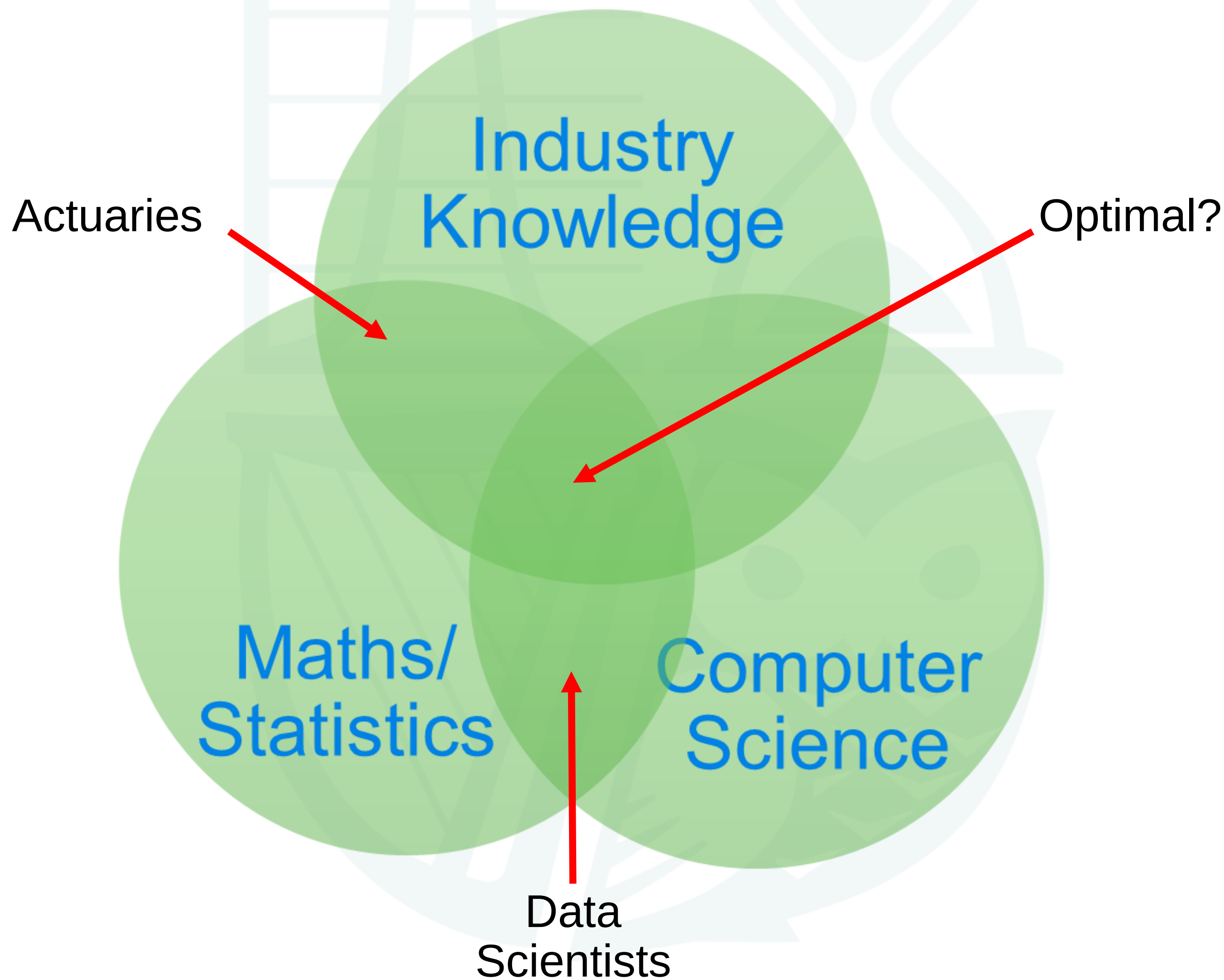
What is Data Science?

The more data you have,
The better your decisions should be

Data Science Map



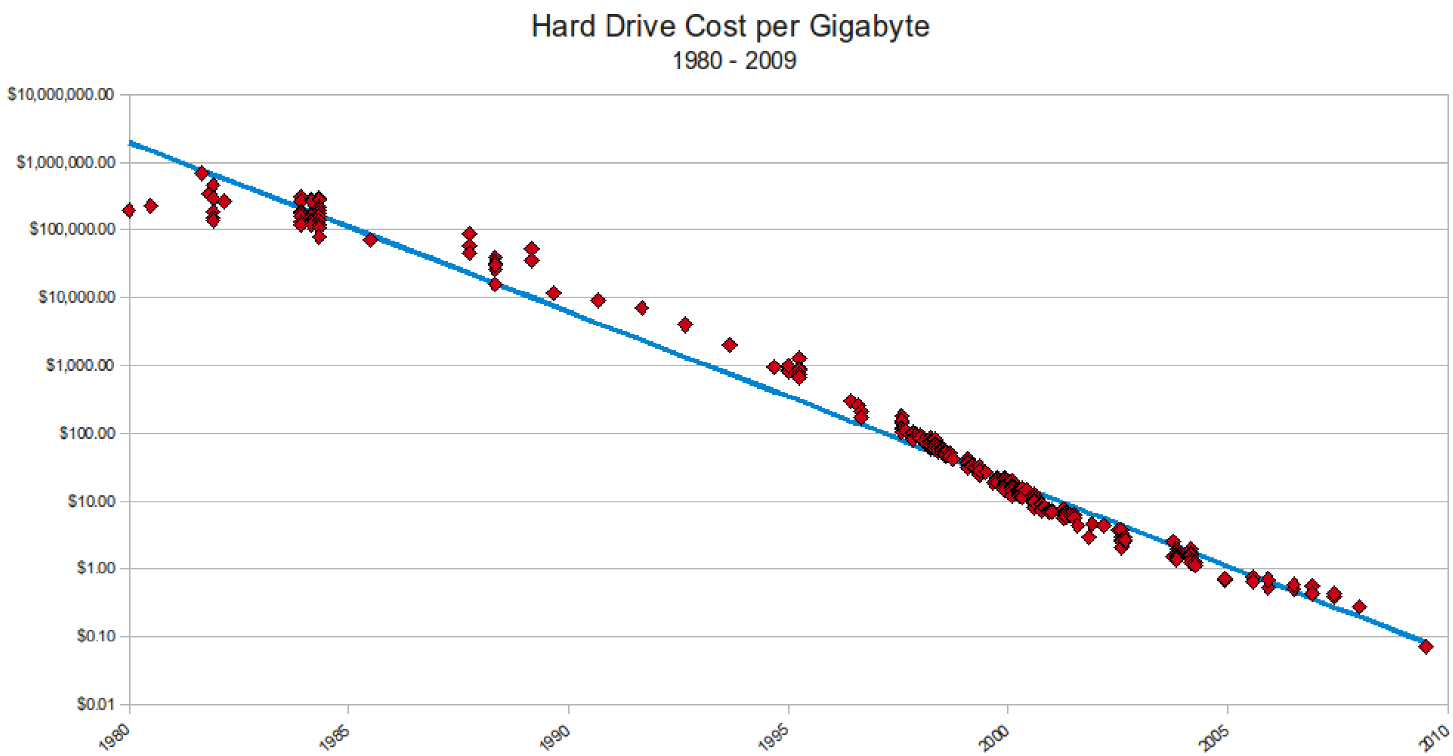
Data Science Map: Insurance Industry



Demystifying Data Science

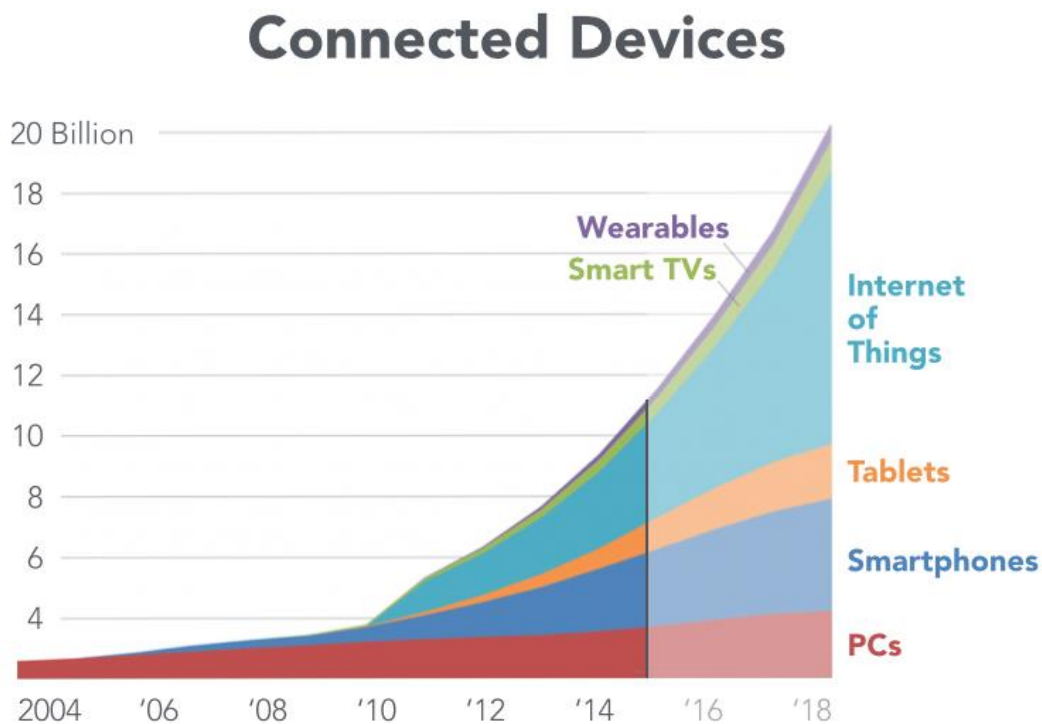
- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples

Data Storage Costs



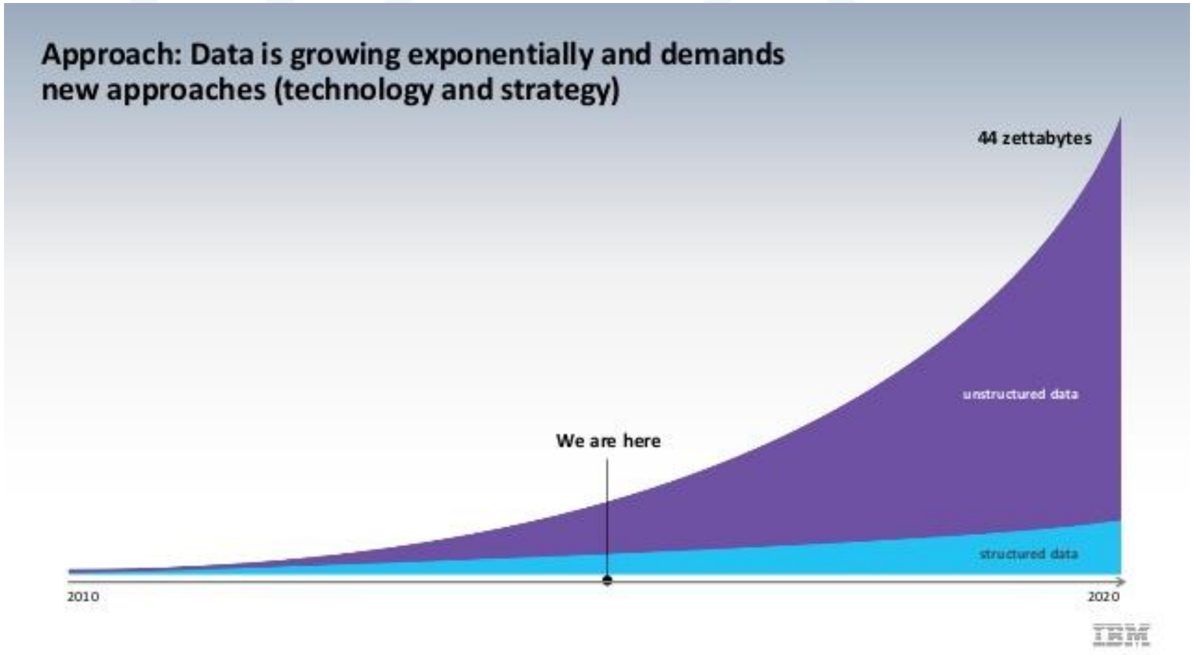
Digitalization

Number of Wifi-Connected Devices

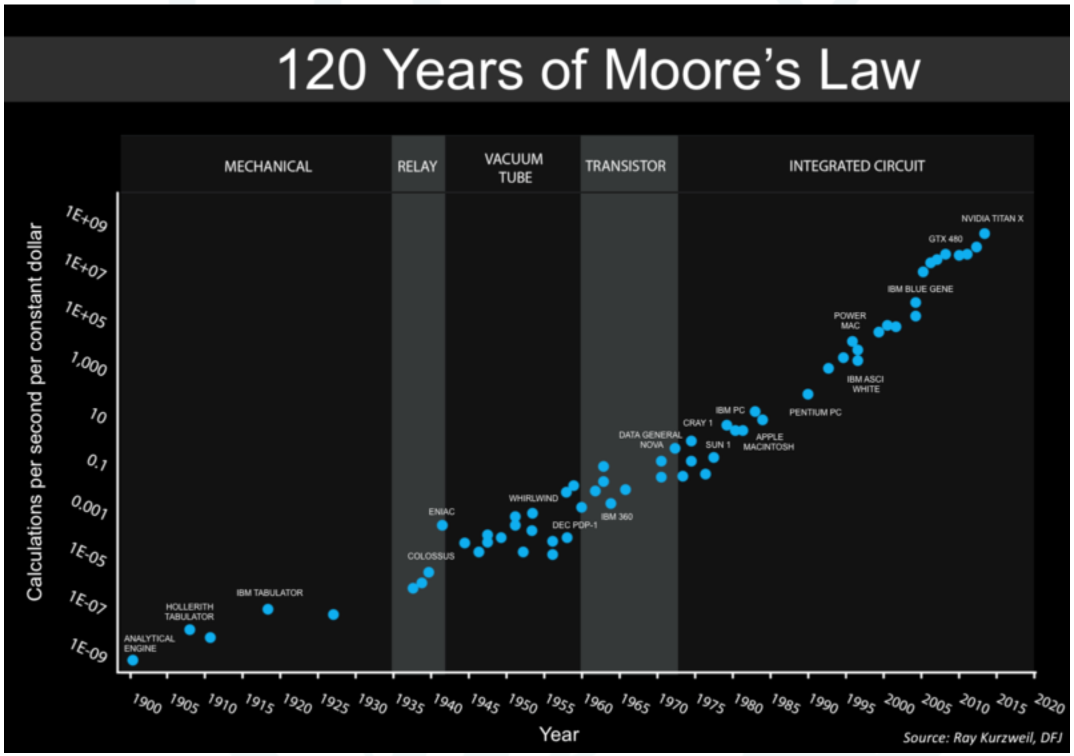


Source: Gartner, IDC, Strategy Analytics, Machina Research, company filings, BII estimates (http://forecastjoy.com/wp-content/uploads/2014/03/deviceforecast.png)

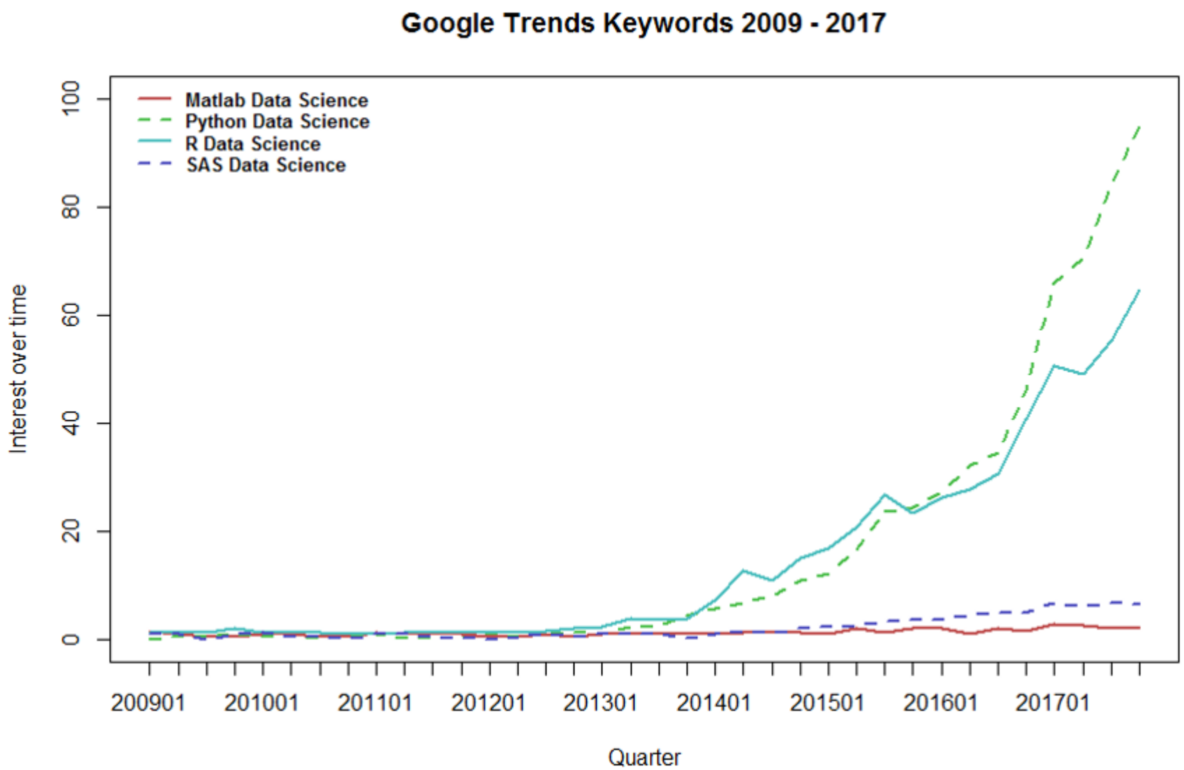
Volume of Data



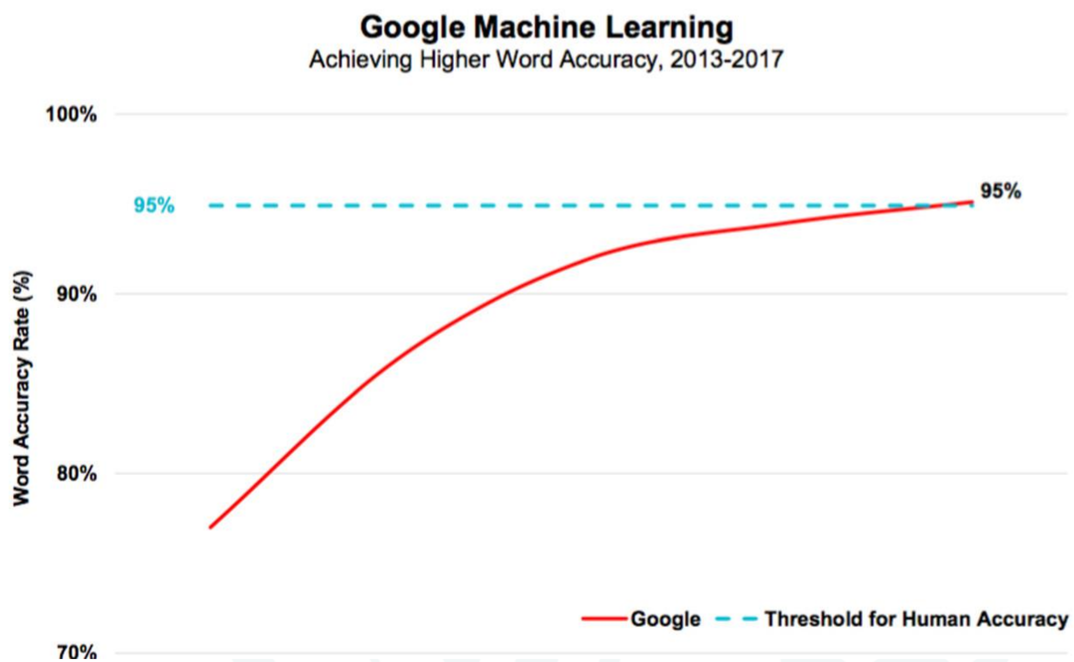
Computer Speeds



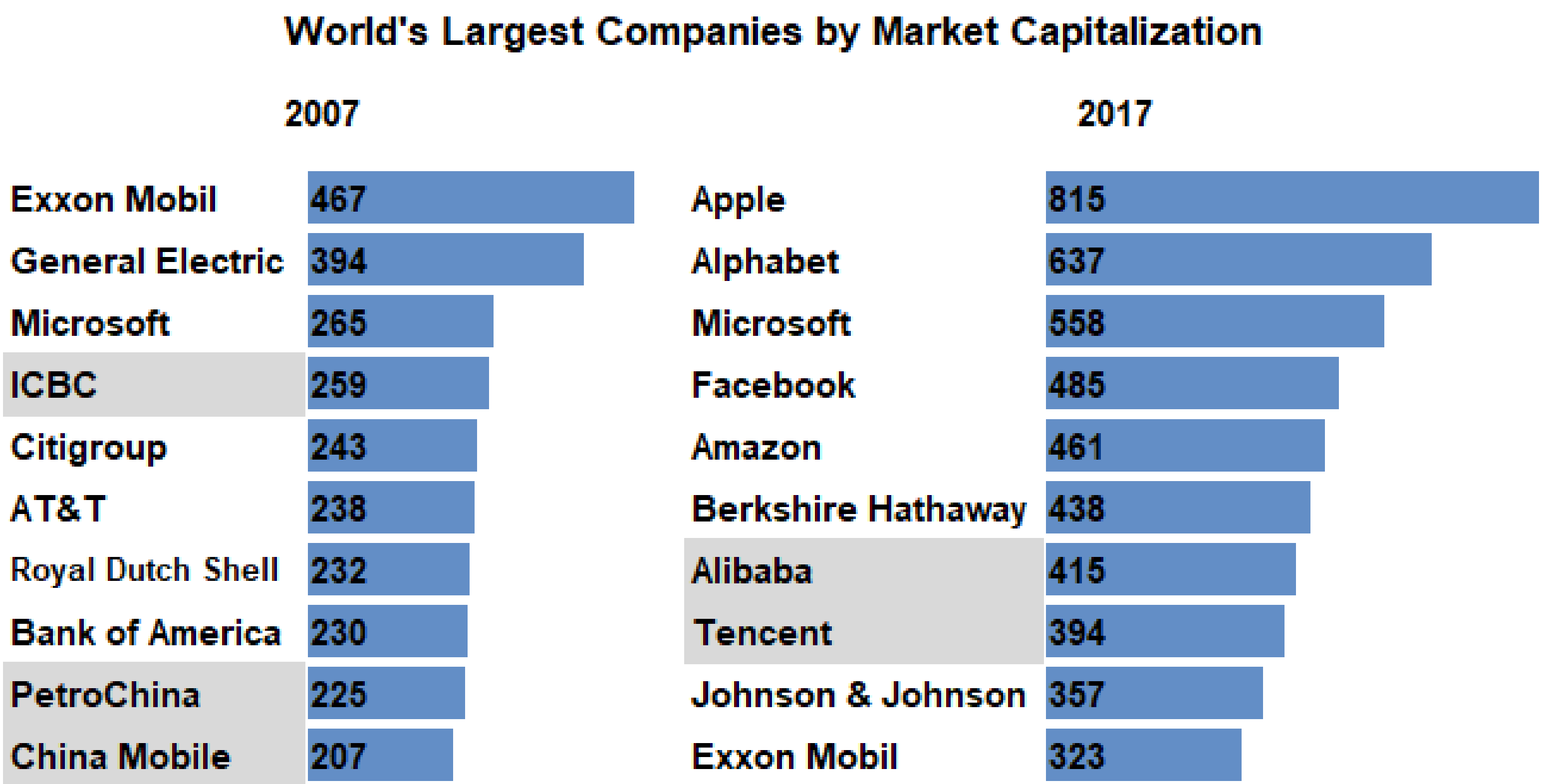
Data Science Tools



Machine Learning



Is Data an Asset?



Why is it a Big Deal Now?

Q: Is data an asset?

A: Yes

Q: How can companies extract value from their data?

A: Data Science

Q: Who will actually analyse this data?

A: Data Scientists

Demystifying Data Science

- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples

Data Science Process

Obtain Data + Develop Plan

↓

Explore Data

↓

Clean + Reformat

↓

Model

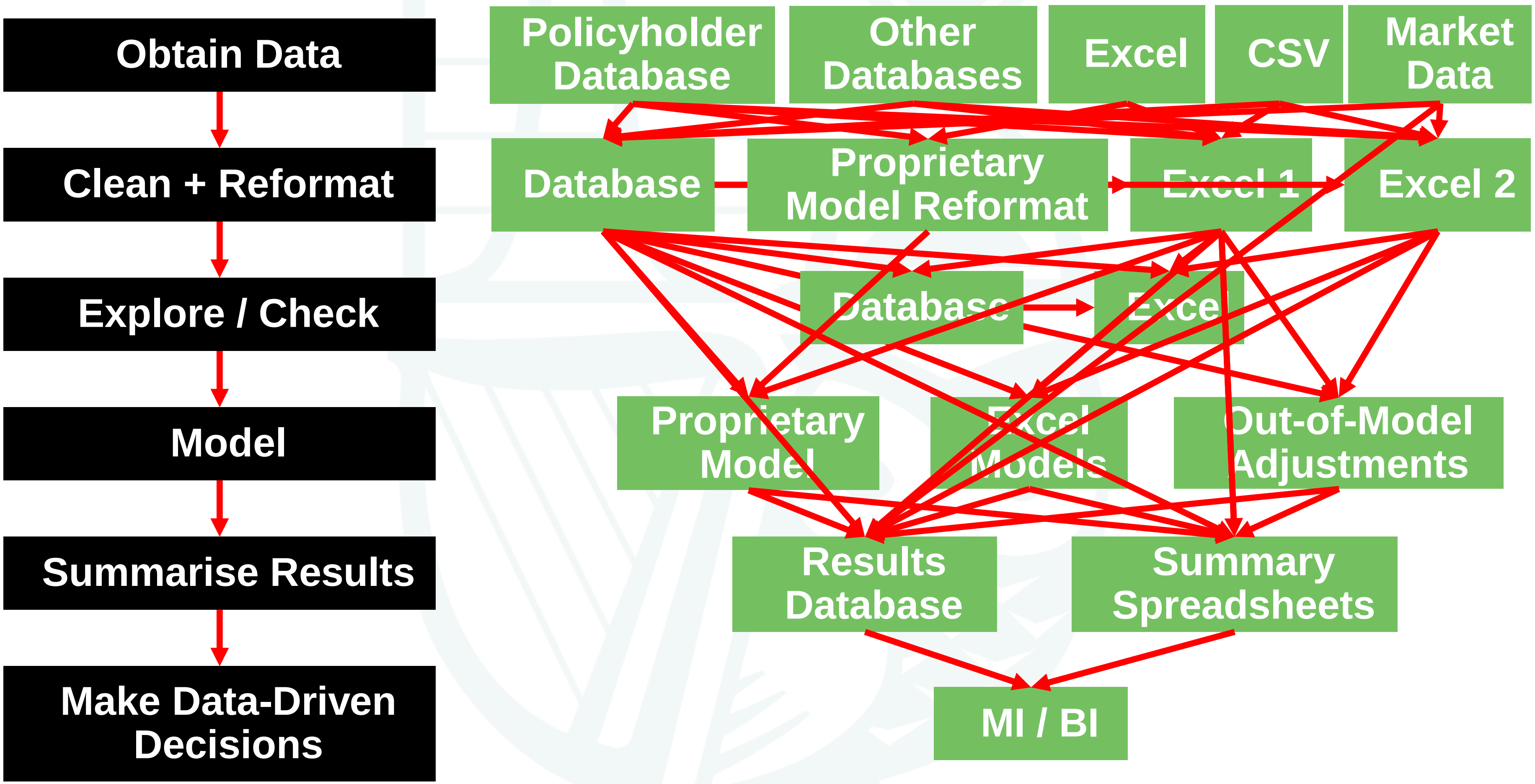
↓

Summarise Results

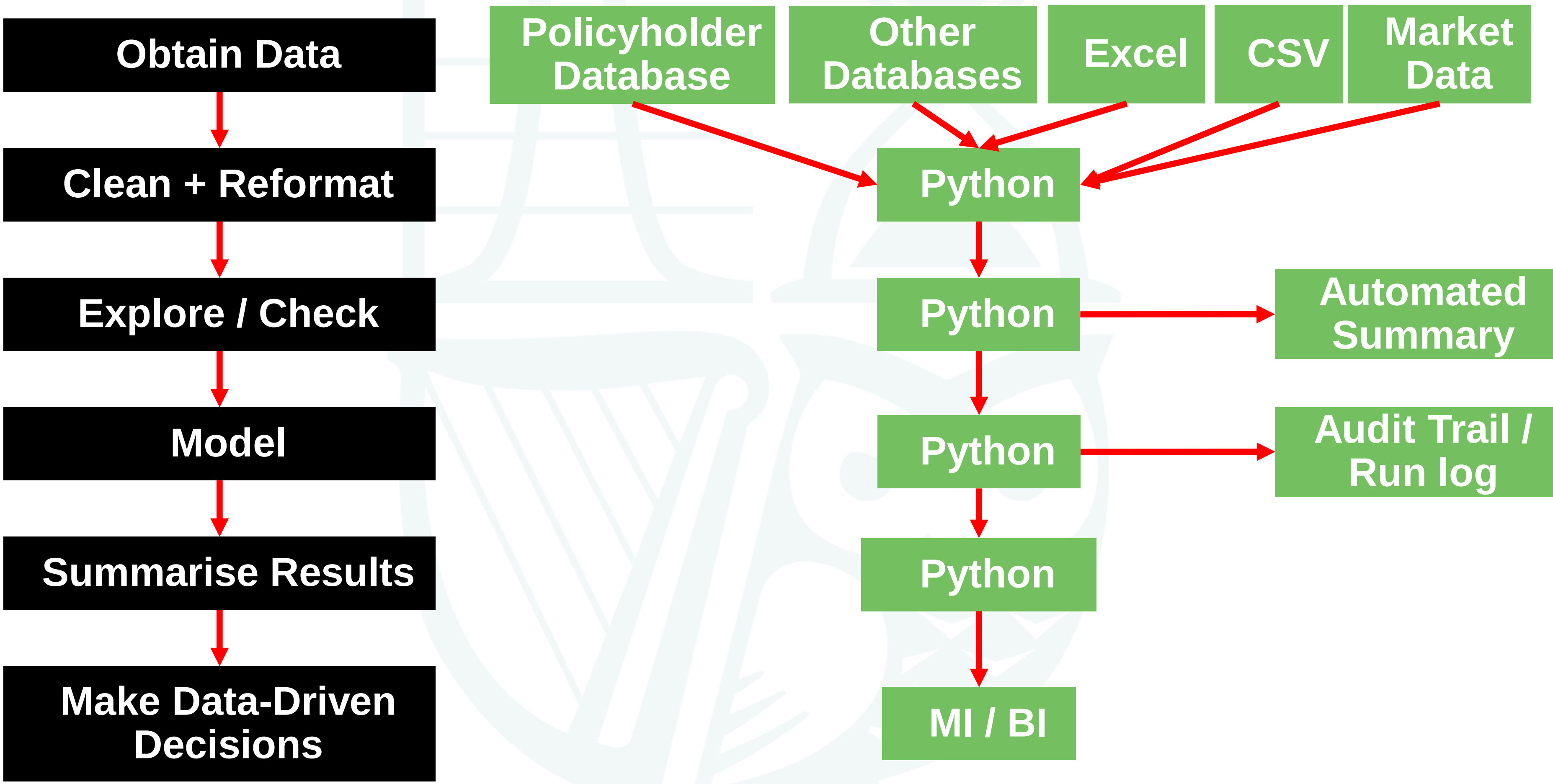
↓

Make Data-Driven Decisions

Traditional Actuarial Process



Data Science Process



Opportunities for Actuaries (1)

- Streamline your processes using open-source data science tools
  - Improve efficiency and reduce time costs
  - Reduced risk of manual error
  - Spend time on value-added work rather than manual labour

Opportunities for Actuaries (2)

- The ultimate wider field?
- Opportunity to drive revenue growth
  - (e.g. using policyholder-level predictive modelling)
- Opportunity to work in different industries
- Powerful new tools to solve real-world problems
- Already familiar with handling data and building complex models
- CDO Roles
- Superstar salaries for top researchers

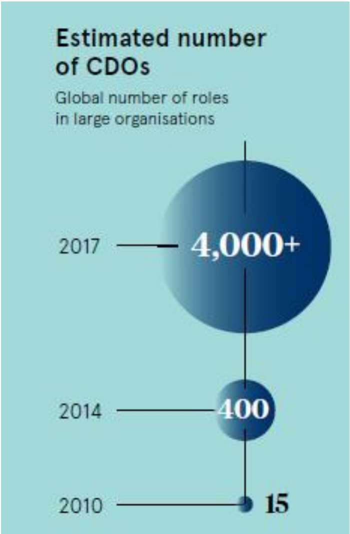
Jobs that pay over $100k

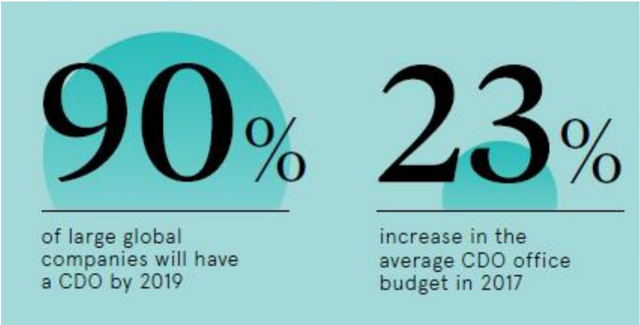
Job title	Average annual salary	Job title	Average annual salary
Neurologist	$217,837	Data scientist	$135,315
Psychiatrist	$194,563	Chief financial officer	$127,887
Anesthesiologist	$173,694	Android developer	$120,971
Radiologist	$168,706	Senior software engineer	$119,791
Physician	$165,391	Full stack developer	$111,709
Dentist	$157,250	Actuary	$111,474
Director of product management	$147,363	Tax manager	$108,515
Surgeon	$140,892	Director of business development	$107,789
Machine learning engineer	$137,332	Architect	$104,080
Vice president of sales	$136,071	Nurse practitioner	$103,233

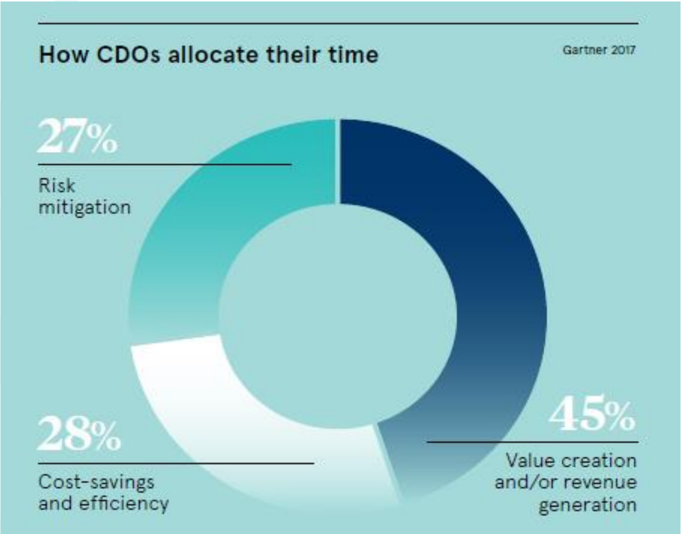
Source: Indeed

Source: Indeed.com, November 2017

Opportunities for Actuaries: Chief Data Officers







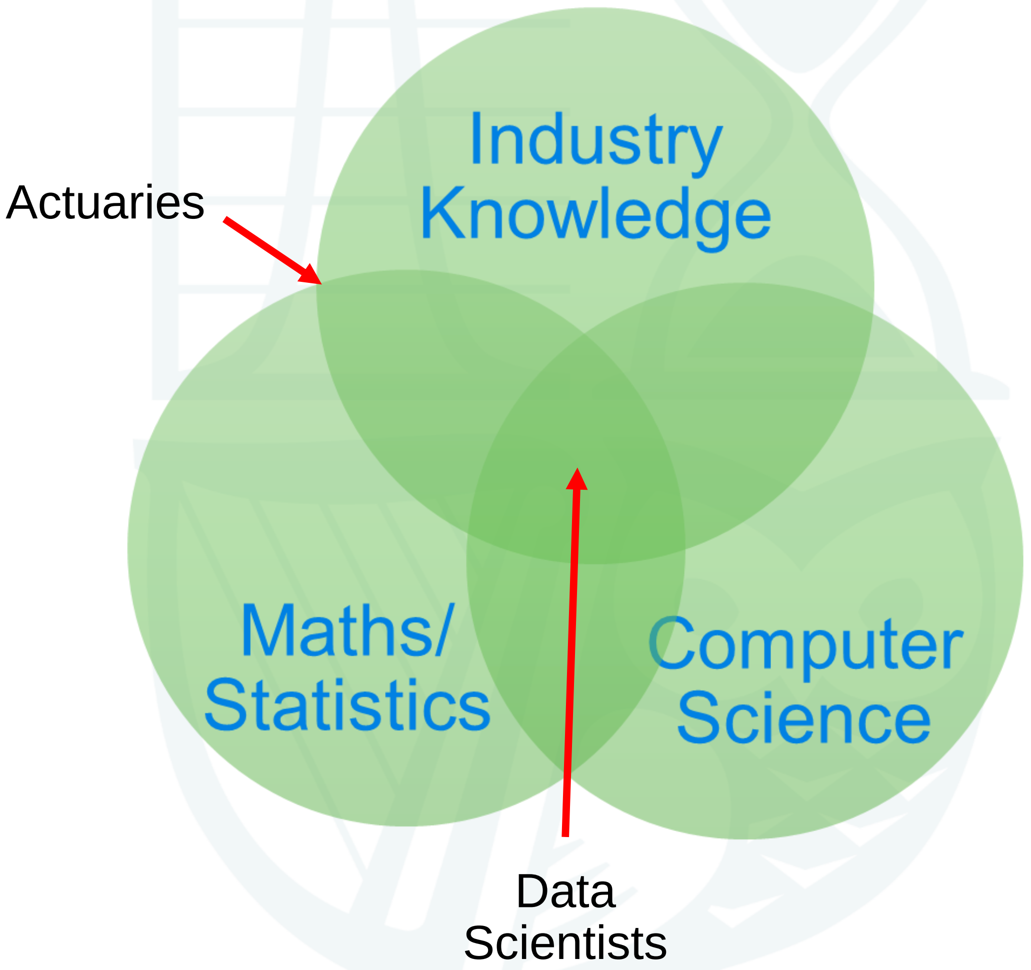
Source: VisualCapitalist.com: The Rise of the Chief Data Officer

Threats for Actuaries

- Increased competition from data scientists
  - Who have strong computer skills
  - Who have powerful predictive models
  - Strong ability to handle data and extract information from the Company's data

  - Particularly for younger actuaries

Threats


Actuaries
Industry Knowledge
Maths/ Statistics
Computer Science
Data Scientists

Threat Mitigation

- Improve data science skills within each actuarial team
  - Mainly by improving computer skills and learning about machine learning models

- Gain access to open-source data science tools at work
  - Overcome internal challenges to open-source software
  - e.g. the IT department might be reluctant to use new software

Opportunities for Companies

- Extract value from their data asset
- Make better data-driven decisions
- Better understanding of risks and opportunities by doing quick, novel analyses of the data
- Streamline operations

Threats for Companies

- New companies could develop massive structural advantages over incumbents?
  - E.g. Amazon have massive structural advantages over traditional retailers

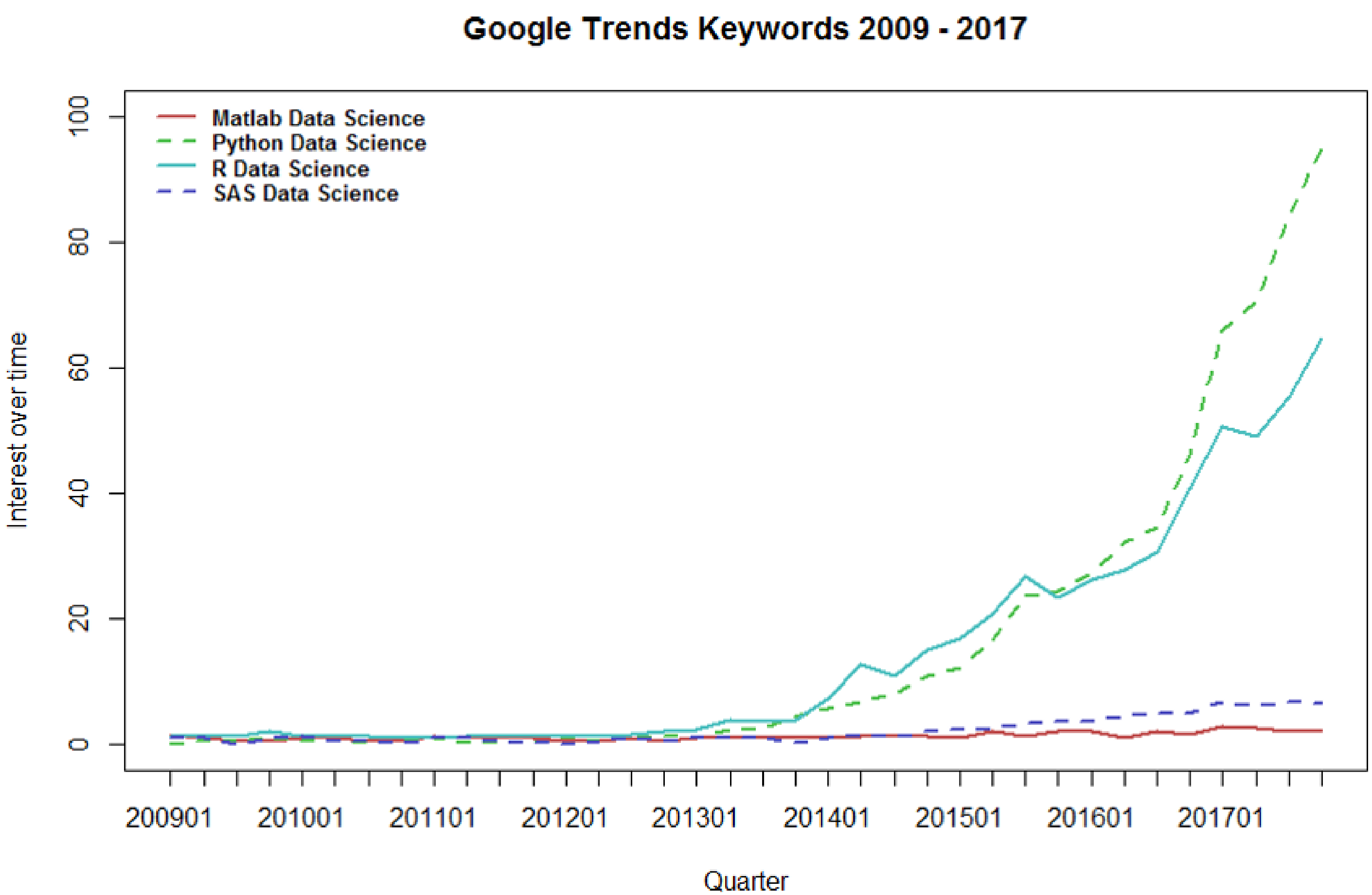
Demystifying Data Science

- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples

Python and R

- Python is a high level, general purpose programming language with readable syntax
- R is a statistical programming language designed by statisticians for statisticians
- Both are widely used for data science
- Both have similar market-leading functionality

Trends


Google Trends Keywords 2009 - 2017
Matlab Data Science
Python Data Science
R Data Science
SAS Data Science
Interest over time
0
20
40
60
80
100
200901
201001
201101
201201
201301
201401
201501
201601
201701
Quarter

Open-Source

Open-source software:

Users have the ability to:

- Run
- Study
- Modify
- Improve
- Copy
- Distribute to anyone and for any purpose

The Python Data Science Stack

- Programming Language
- Numerical and scientific calculations
- Organising data, merging data, doing calculations
- Graphs
- Big Data
- Machine learning
- Artificial intelligence and ultra-fast calculations

Open Source vs Closed Source

	Open Source	Closed Source
Source Code	Open	Hidden
Redistributable?	Yes	No
Modifiable?	Yes	No
Licence and Subscription Fees?	No	Yes
Documentation, Helpdesk and Tutorials	Online (Google / Stackoverflow)	Provided by Provider (for a fee)
Responsiveness to bugs and market	Quick to respond	Depends on Provider
Version Control Systems	Available	Depends on Provider

Open-Source Advantages

- Fast
- Scalable
- Capable of full automation
- No licencing fees
- Auditability
- Flexibility
- Sustainability
- Easy to find or train developers
- Fast Learning Curve

Open-Source Misconceptions

- Not secure
- Too hard to learn
- No documentation / bad documentation
- Not as good as proprietary software

Closed Source Advantages

- It's the Standard / Well Known
- Easier for Unskilled Users
- Guaranteed Support (for a fee)
- Managers prefer buying Software as a Service rather than building own systems?
- Warranties and Indemnity Liability
- Unlikely to Become Obsolete?

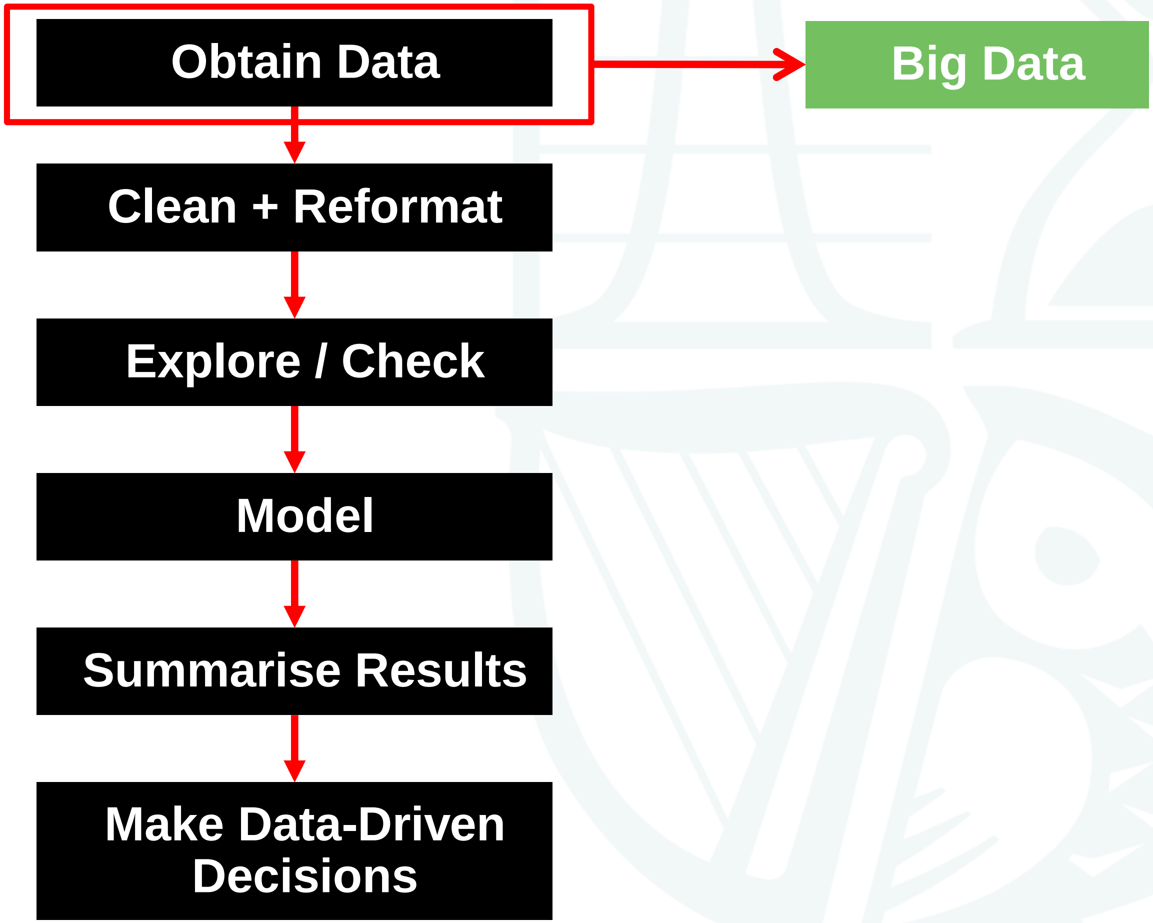
Closed Source Risks

- Expensive
- Restrictive licences
- Lock-in / Capture
- Time-consuming / Hard to learn
- Management Incentives (Planned obsolescence / cash cow)
- Bankruptcy
- Unknown code quality
- Unknown level of security
- No incentive to provide good documentation

Demystifying Data Science

- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples

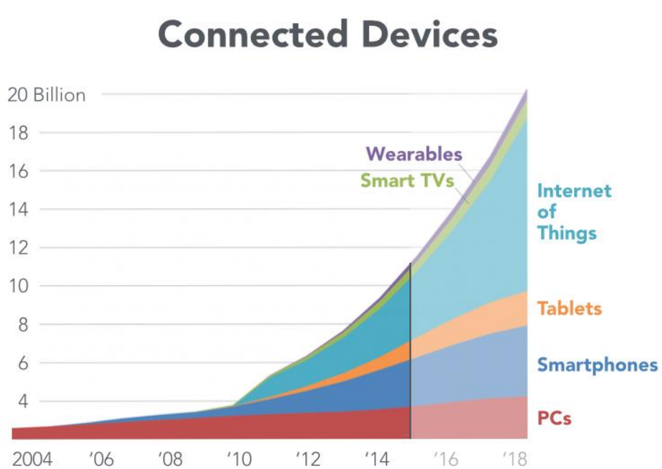
Data Science Process : Buzzwords



Big Data

Big data: data sets that are too big and complex for traditional data processing software

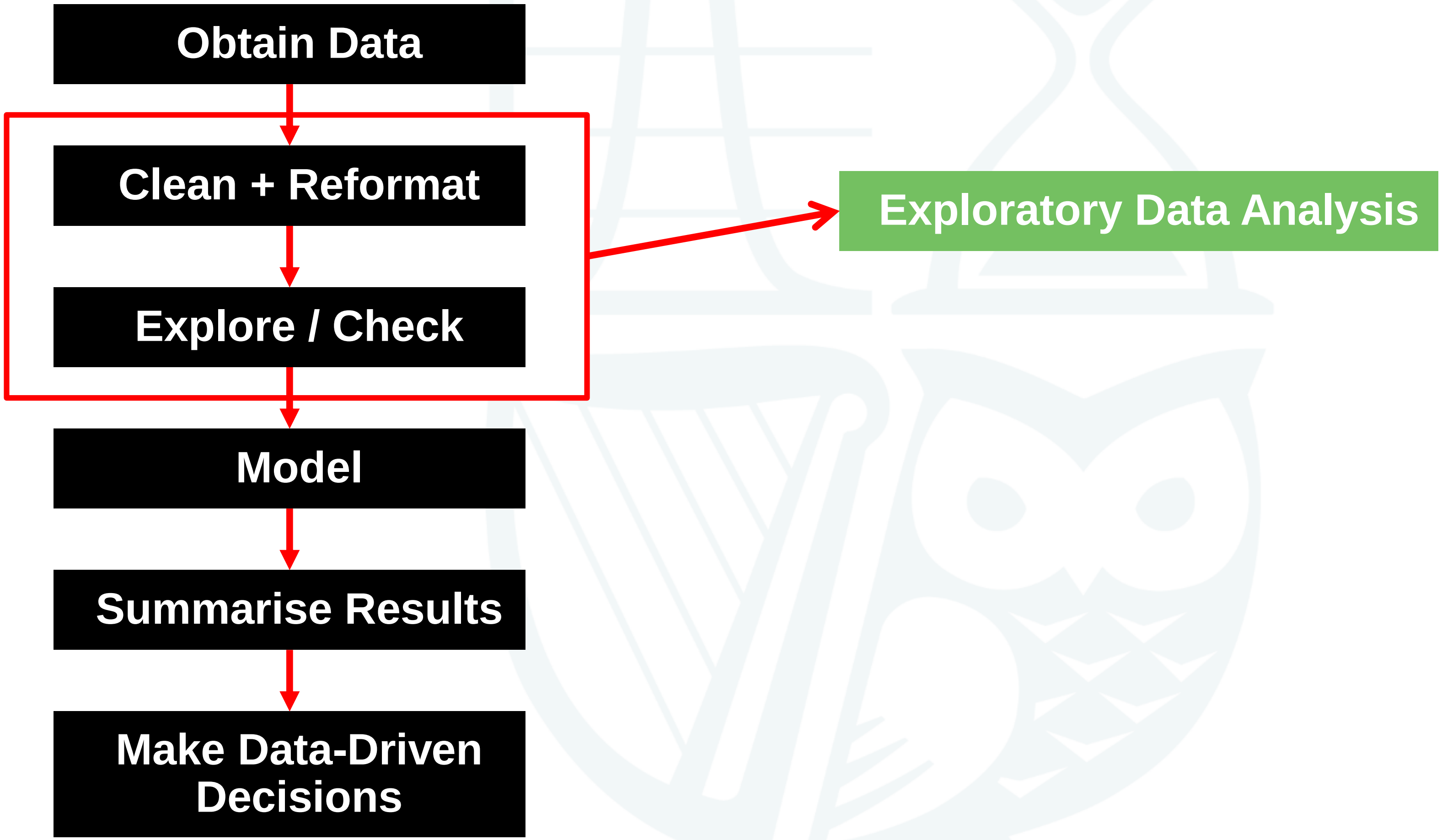
Need to use new software which can distribute the storage and calculations across different machines



Source: Gartner, IDC, Strategy Analytics, Machina Research, company filings, BII estimates (http://forecastjoy.com/wp-content/uploads/2014/03/deviceforecast.png)

hadoop APACHE Spark

Data Science Process



Exploratory Data Analysis

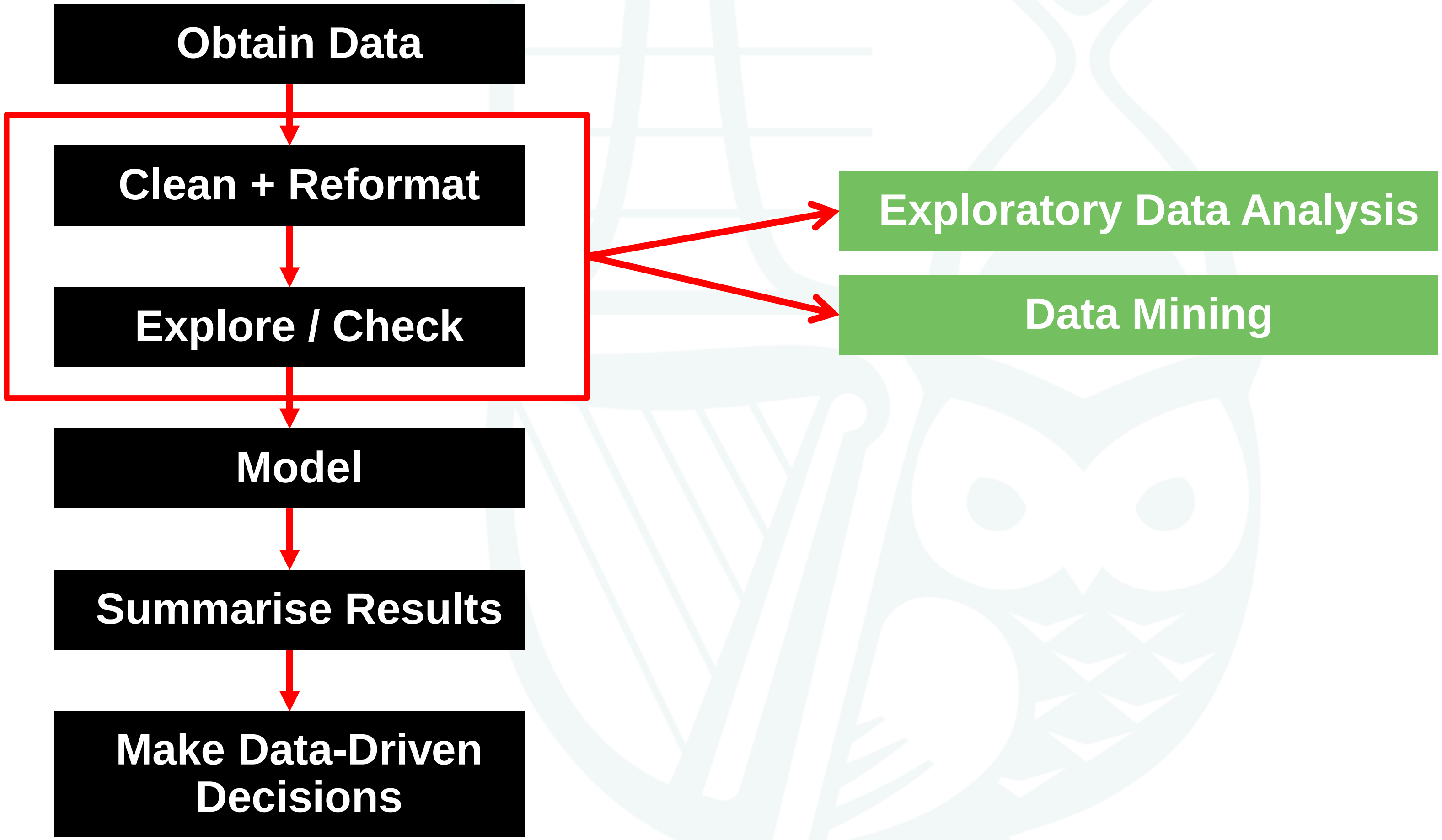
EDA: Analyzing data sets to find their main characteristics

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

matplotlib

Data Science Process

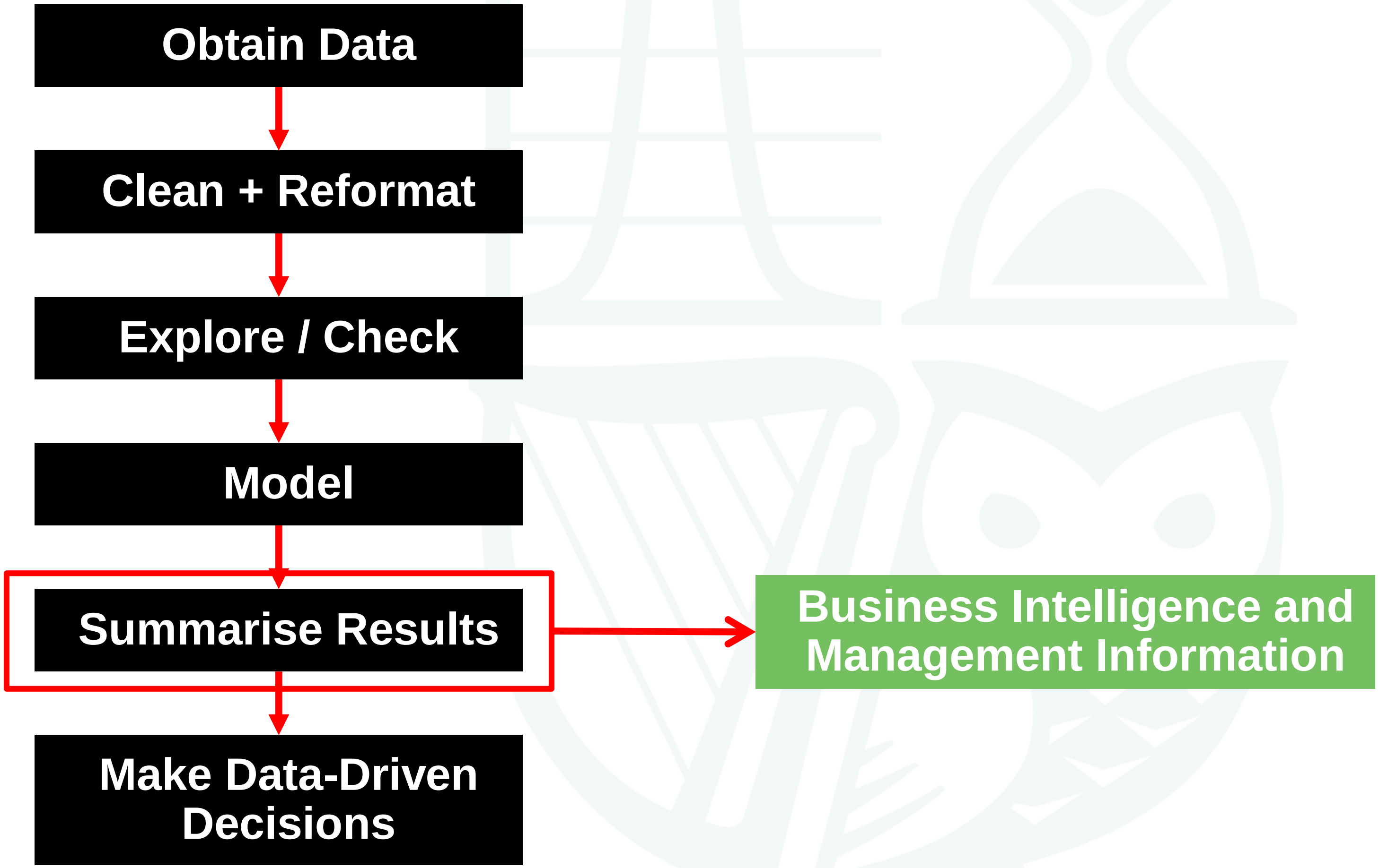


Data Mining

Data Mining is the process of finding patterns and relationships in large datasets

Goal = to extract valuable understandable information from data

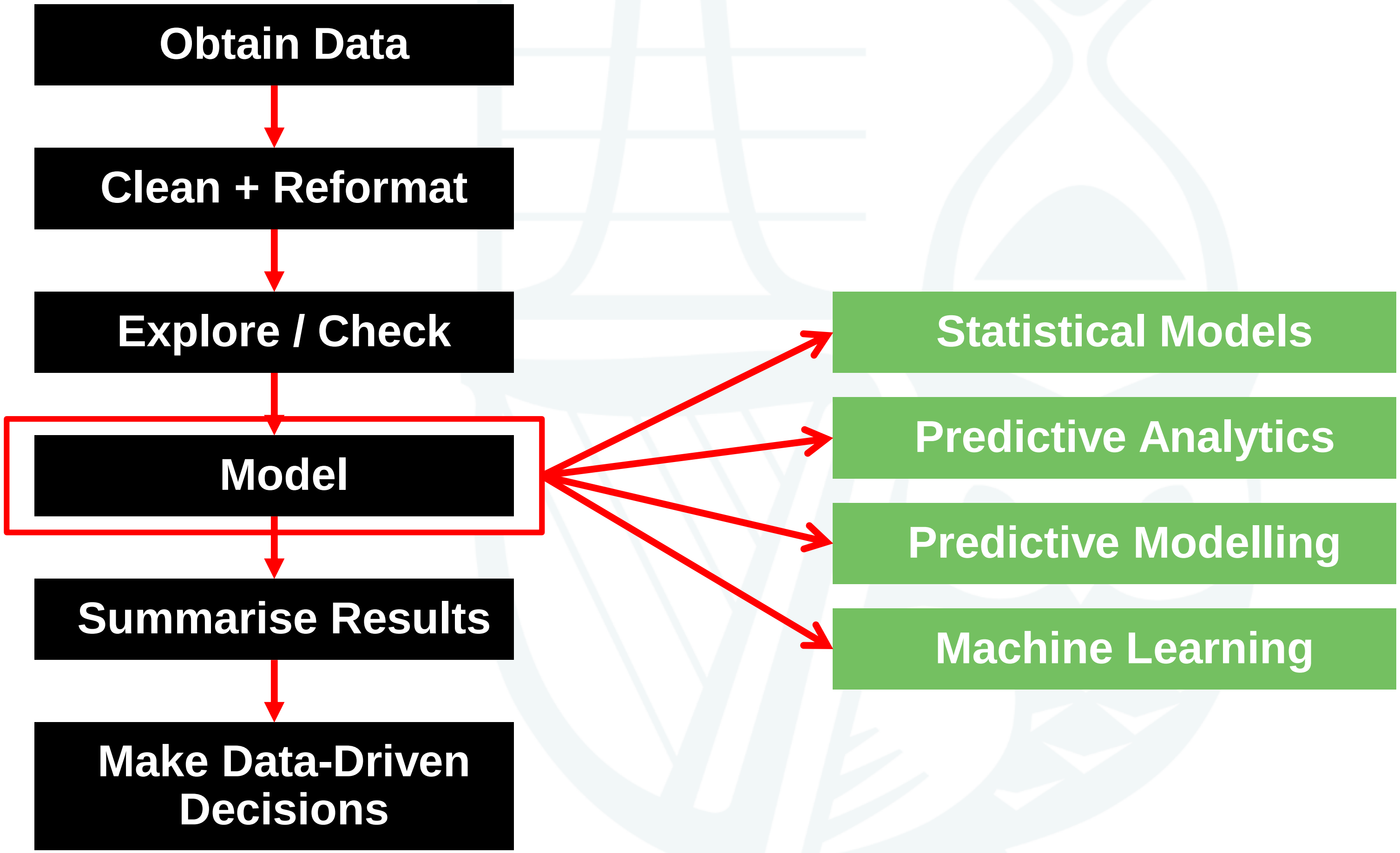
Data Science Process



Business Intelligence and Management Information

Analyzing data and presenting information to help executives make informed business decisions

Data Science Process

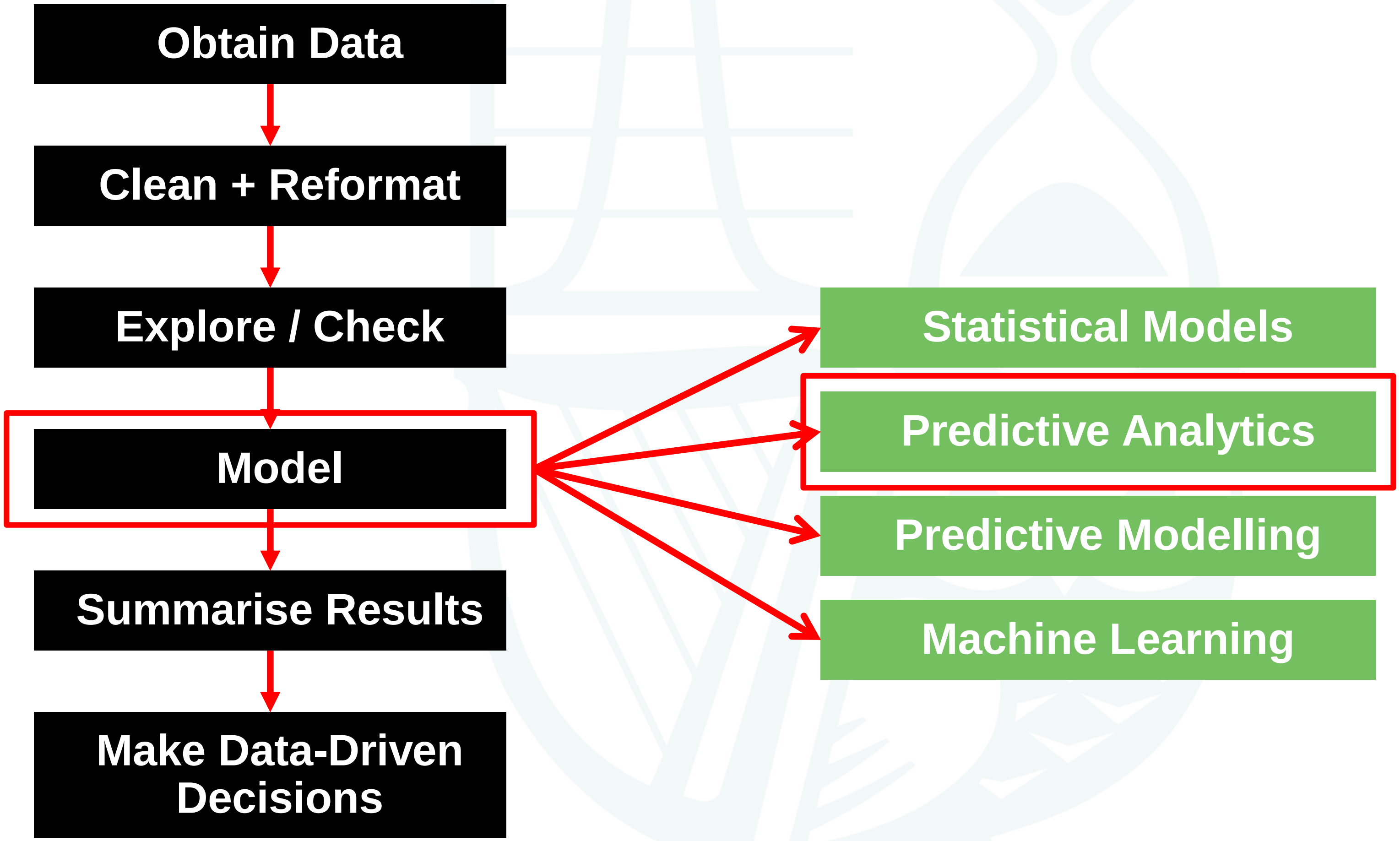


Statistics vs Predictive Analytics vs Machine Learning

Statistics is about data:

- Collection
- Organisation
- Analysis
- Interpretation
- Presentation

Data Science Process



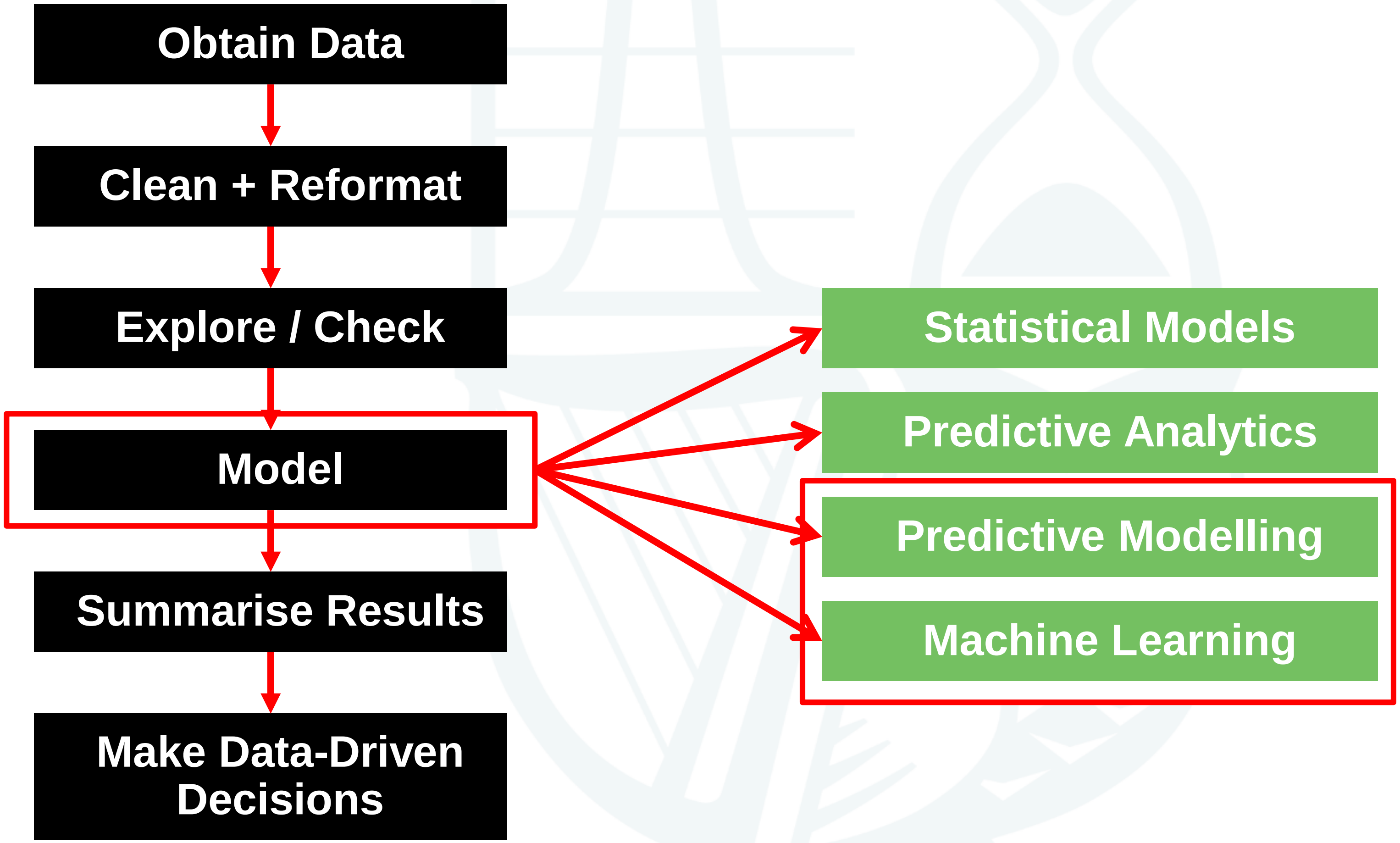
Predictive Analytics

Predictive Analytics is a set of statistical techniques that make predictions about future unknown events

For example:

- Data mining
- Traditional predictive models
- Machine learning models

Data Science Process



Predictive Modelling

Predictive models are models which make predictions about future unknown events.

- Using current and historical data
- Allowing for relationships among many factors
- Make predictions about every example in the dataset
- These predictions can be used to guide decision making

Predictive Modelling

Two main types:

- Traditional predictive models
- Machine learning models

Traditional Predictive Models

Characteristics of traditional predictive models:

- Explainable and interpretable
- Grounded in maths and statistics
- All parameters derived manually using closed form mathematical solutions or simple algorithms
- Lots of manual effort required to build high accuracy models

Machine Learning Models

Machine learning models are predictive models which have the ability to learn from data without being explicitly programmed

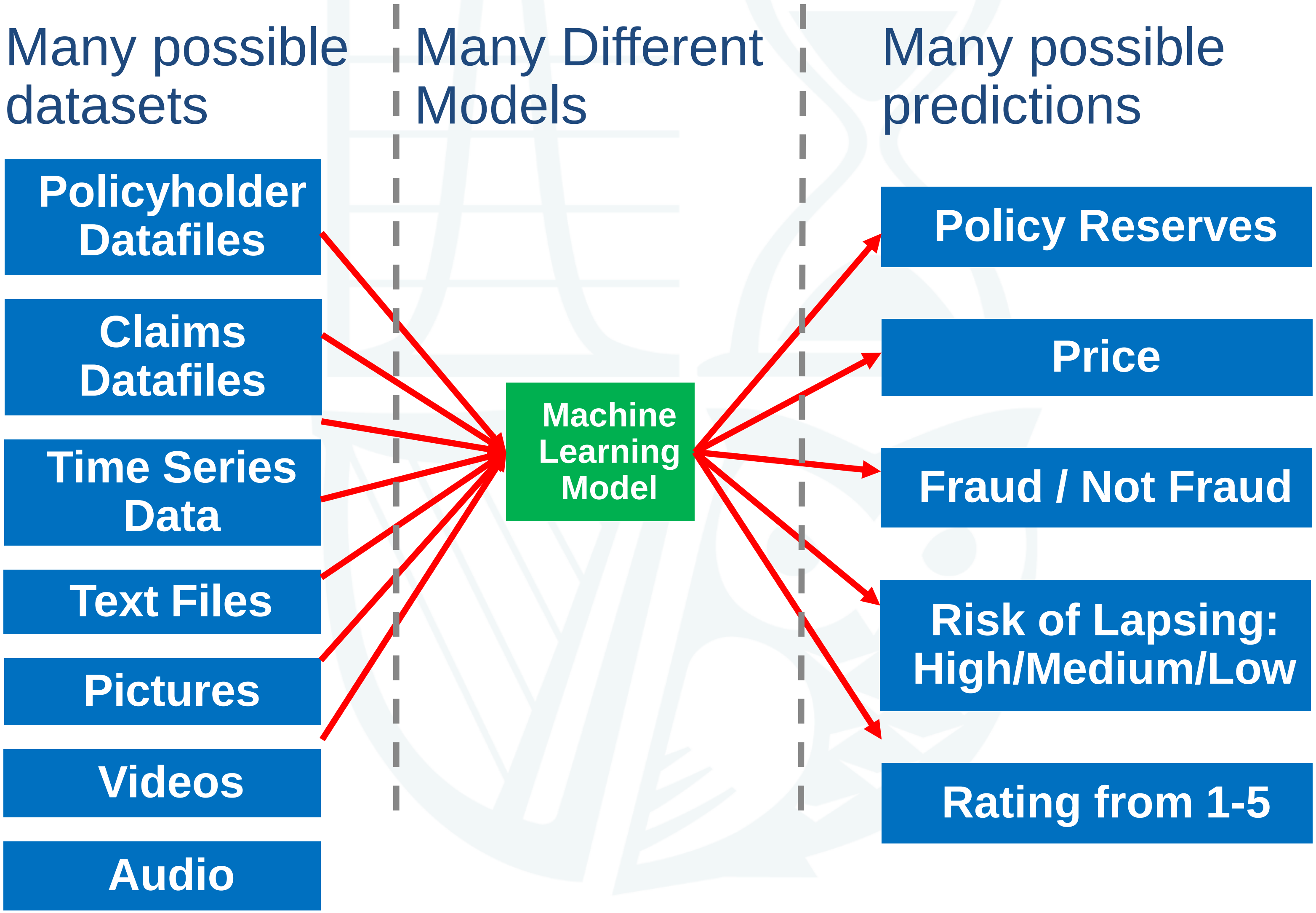
Learning = progressively improving performance on a specific task

Machine Learning Models

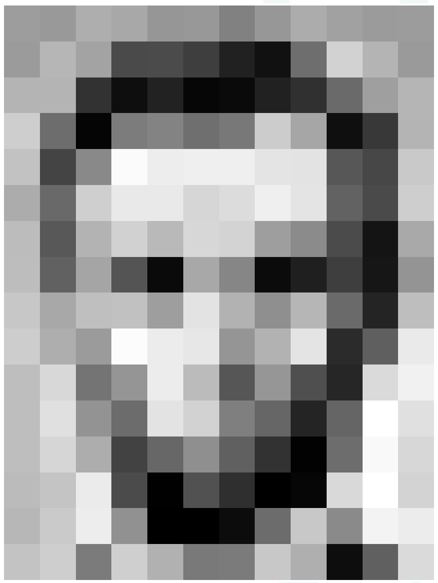
Characteristics of machine-learning models:

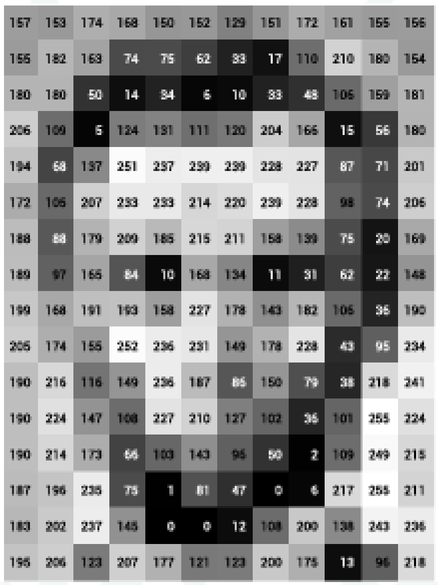
- Automatic
- May be explainable or a black box
- Grounded in computer science
- Most parameters derived automatically using a machine learning algorithm
- Little manual effort required to build high accuracy models

ML Models


Many possible datasets
Many Different Models
Many possible predictions
Policyholder Datafiles
Claims Datafiles
Time Series Data
Text Files
Pictures
Videos
Audio
Machine Learning Model
Policy Reserves
Price
Fraud / Not Fraud
Risk of Lapsing: High/Medium/Low
Rating from 1-5

Digital Photos



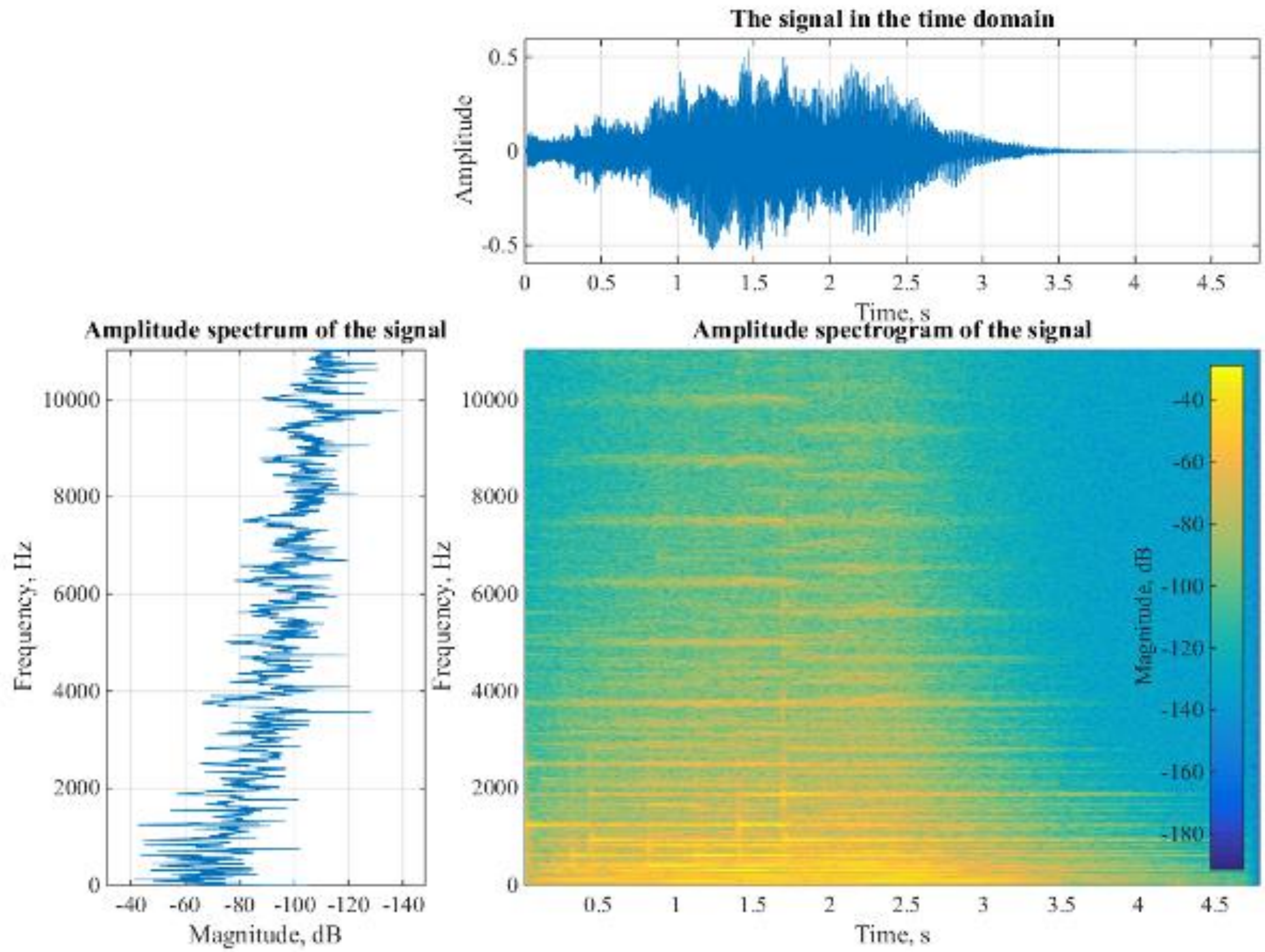


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

- Digital Photos are stored as arrays of numbers

Source: Openframeworks.cc

Digital Audio Files



- Digital Audio files are stored as a time series of arrays
- Each array contains information on pitch and loudness

Source: ch.mathworks.com

Digital Text

- Can be converted to vectors of numbers
  - Glove
  - Word2Vec
  - Word Embeddings

Source: ch.mathworks.com

General Examples of Predictive Models

Self-Driving Cars	Speech-to-text	Fraud Detection	Sales Forecasting
Game Playing	Machine translation	Pricing	Anti-Money Laundering
Reducing Electricity Costs	Chatbots	Credit Risk	Call-Centre Routing
Analysing Satellite Photos	Recommender Systems	Customer Retention	Sentiment Analysis
Reading X-rays	Text-to-Speech	Proxy Models	Geographic Analysis

Example: Machine Translation as Predictive Model

- The model tries to predict what words a human translator would use

Example: Captioning

Train Model

Red dress with White Spots and Black Belt

Red sweater with white stripes on arms and Gingerbread man with Christmas Hat

- The model takes the picture and predicts what the caption should be

Example: Self-Driving Cars

Train Model

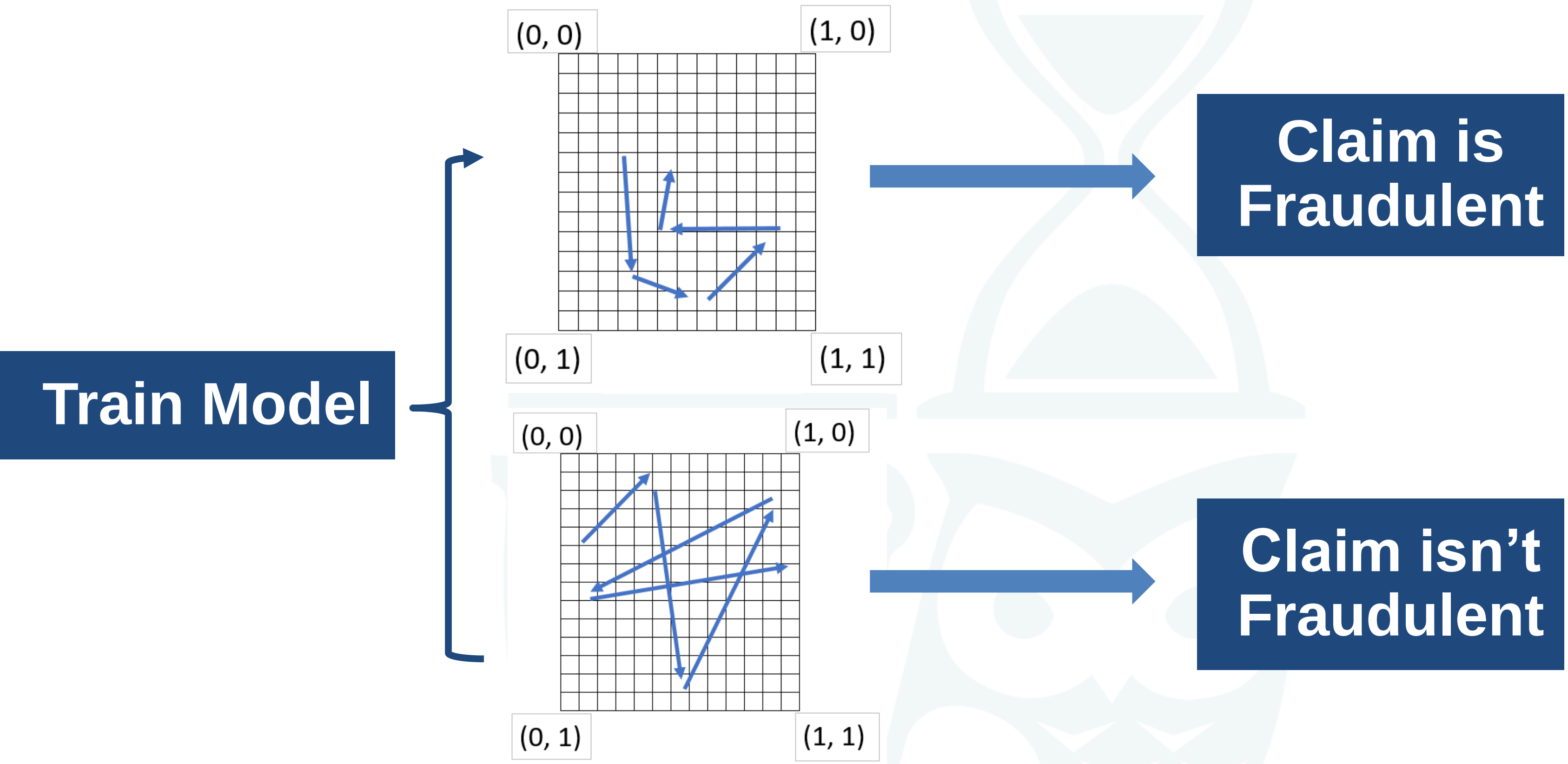
Good Driving

Bad Driving

Model predicts what a good driver would do in the current circumstances

Source: https://clipartxtras.com/

Example: Fraud Detection



The model will predict whether each incoming claim is fraudulent or non-fraudulent

General Examples of Predictive Models

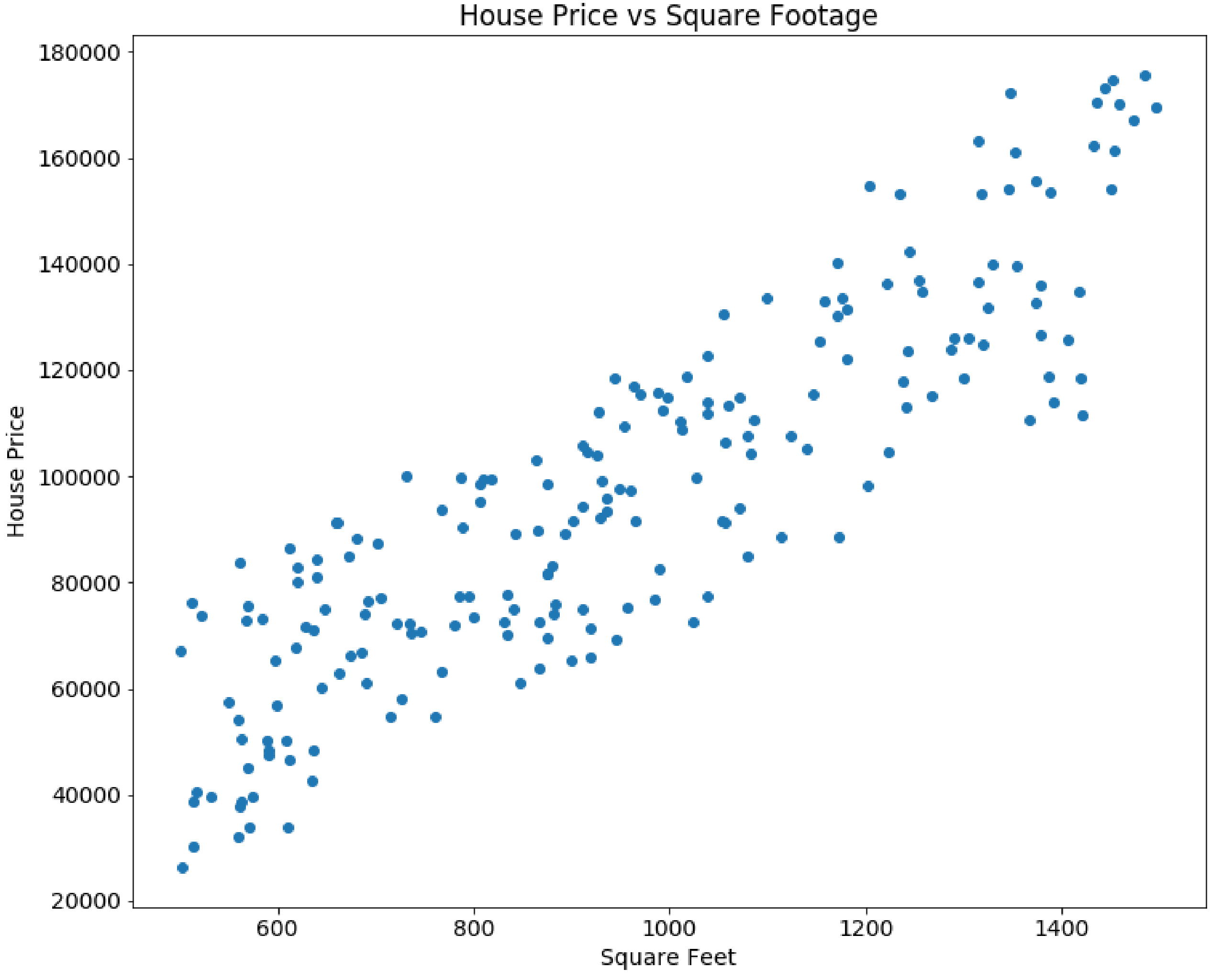
Self-Driving Cars	Speech-to-text	Fraud Detection	Sales Forecasting
Game Playing	Machine translation	Pricing	Anti-Money Laundering
Reducing Electricity Costs	Chatbots	Credit Risk	Call-Centre Routing
Analysing Satellite Photos	Recommender Systems	Customer Retention	Sentiment Analysis
Reading X-rays	Text-to-Speech	Proxy Models	Geographic Analysis

Demystifying Data Science

- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples

Practical Example: Traditional Modelling and Machine Learning

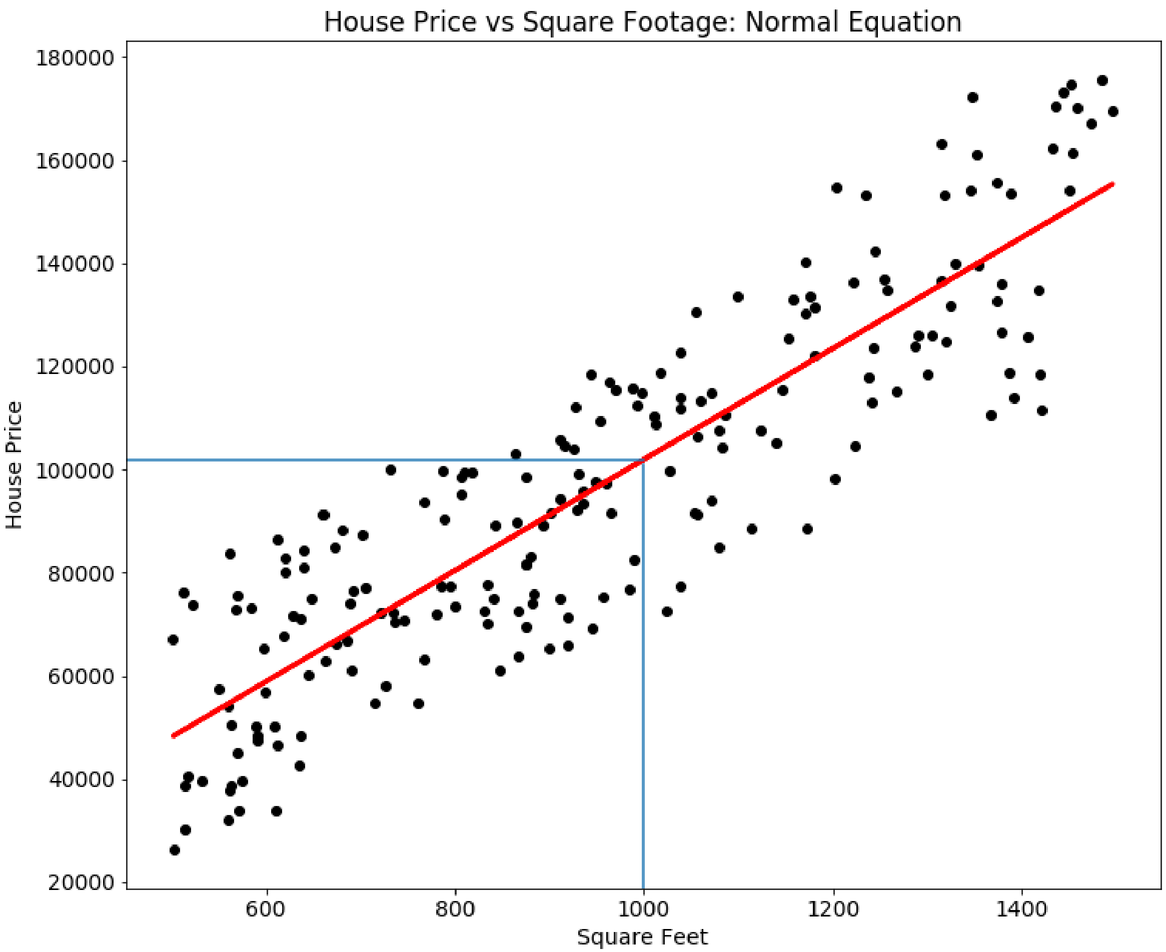
How much is a 1000 square foot house?



Eyeball approach:

Around €90k

Linear Regression Predictive Model



- Linear Regression Model:
  - Price = €101,955
  - Slope = 108
  - Intercept = -5,700
  - MSE = 258 million

- But how do you find the slope and intercept?

Approach 1: Normal Equation

Linear Regression Model:

$$\hat{y} = ax + b = \theta X$$

where:

- $\theta = [a \;\; b]$
- $X = [x \;\; 1]$

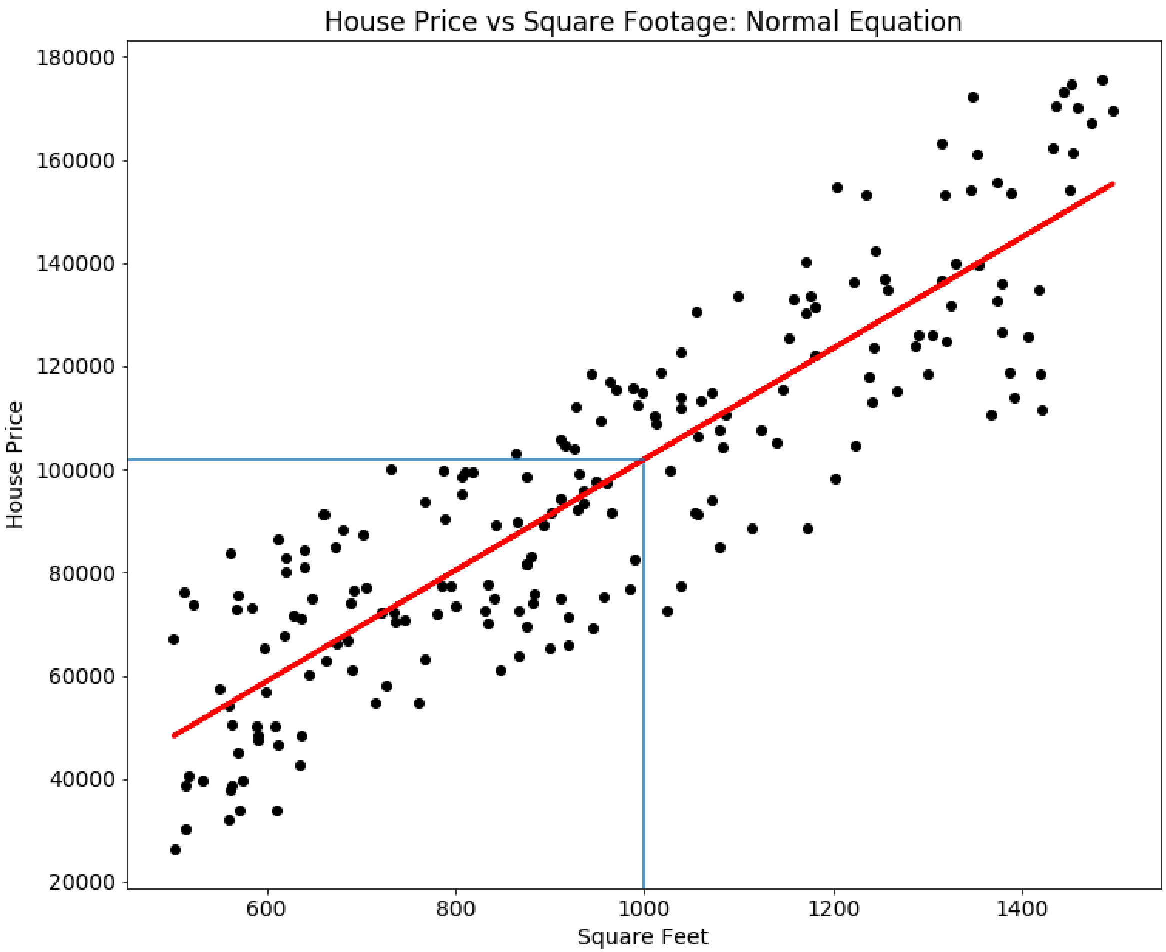
```
theta = (np.linalg.pinv(X.T * X) * X.T) * Y
y_hat = X * theta
```

Choose Loss Function, such as Mean Squared Error

Calculate parameters theta using formula:

$$\theta = (X^T X)^{-1} X^T y$$

Linear Regression Predictive Model



Linear Regression Model:

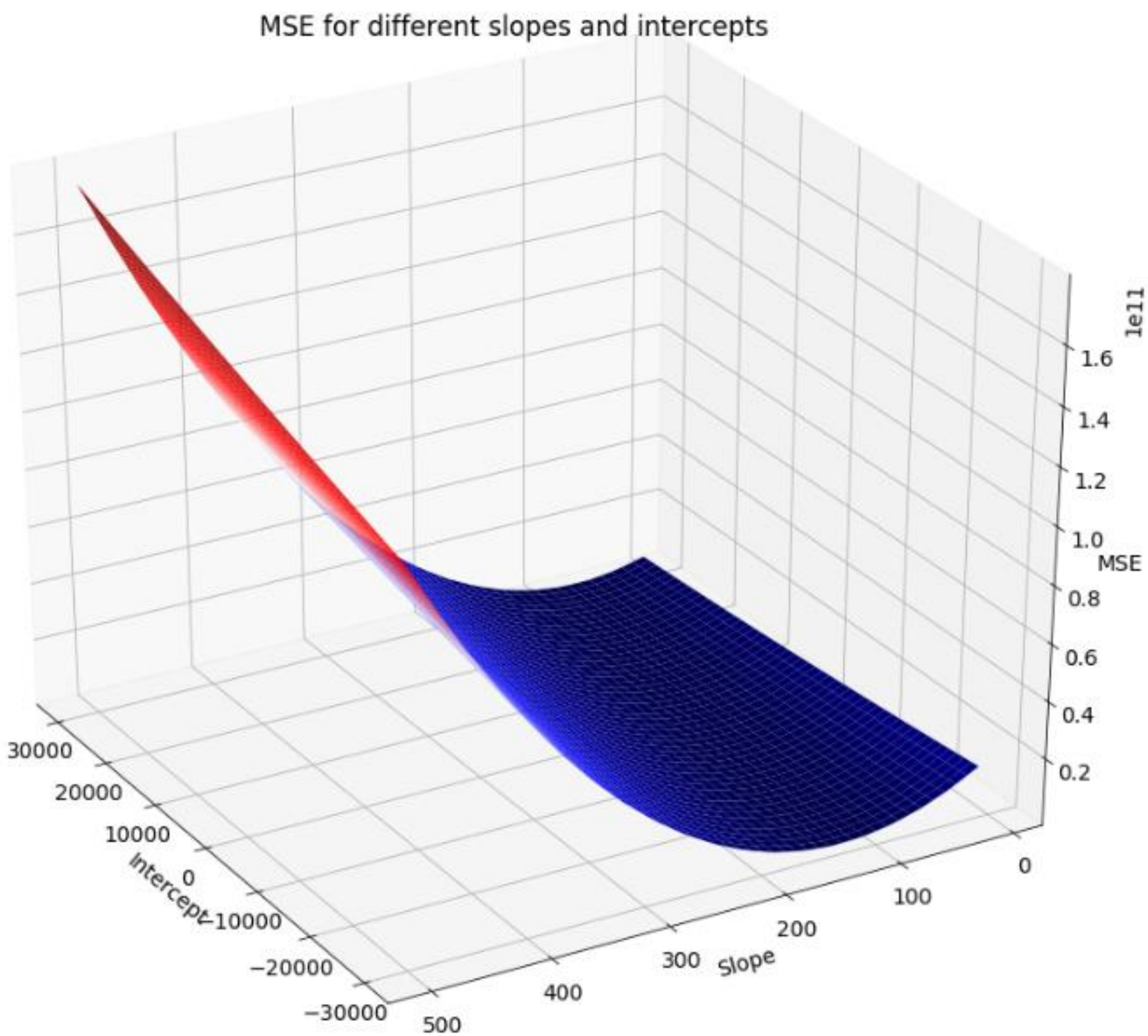
- Price = €101,955
- Slope = 108
- Intercept = -5,700
- MSE = 258 million

Approach 1: Normal Equation

Problem with normal equation:

- Only works if $X^T X$ is invertible
- Doesn’t work on other models
- Doesn’t work well on large datasets

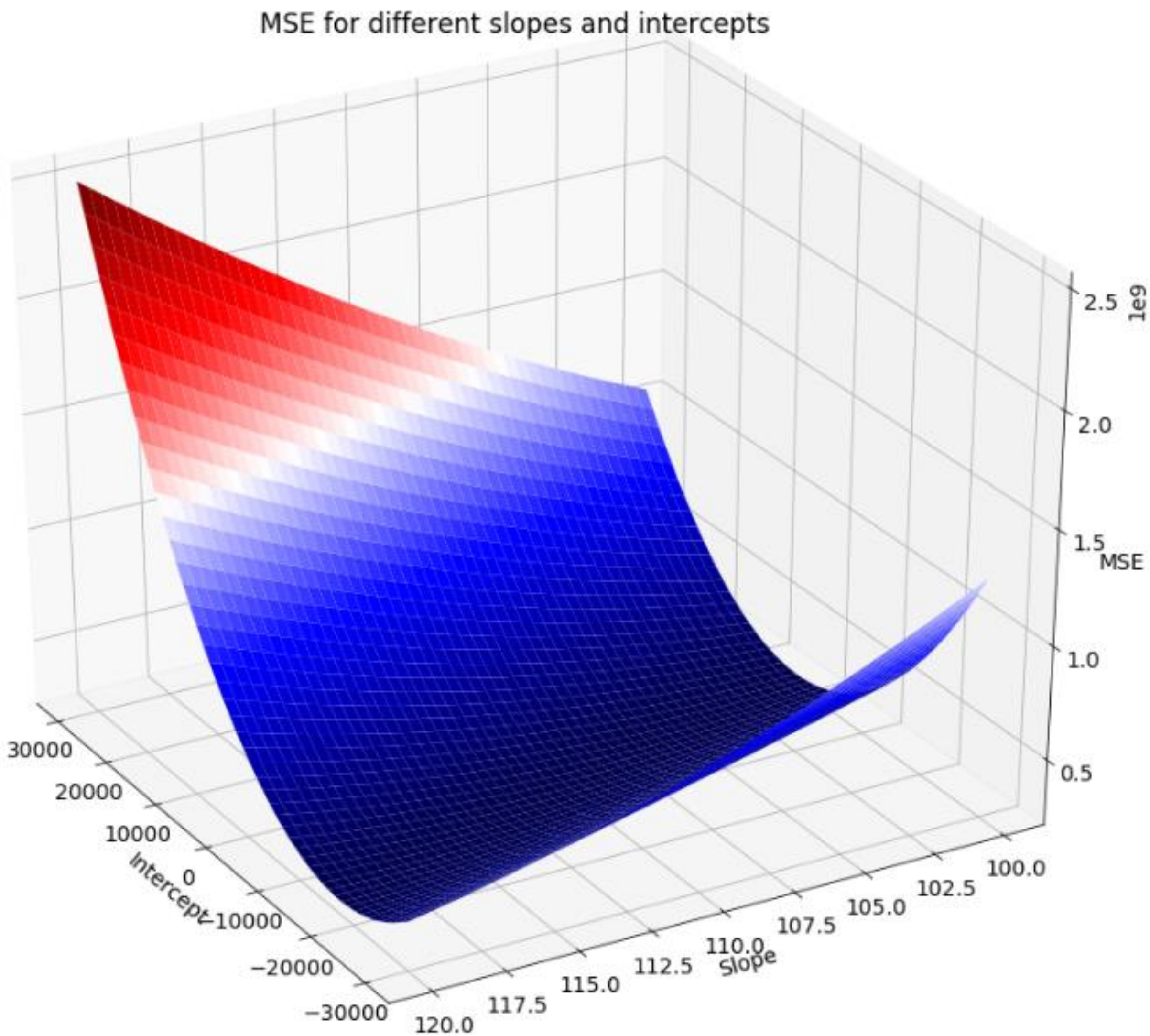
Approach 2: Gridsearch



Point with minimum MSE:

	86
Slopes	113.16
Intercepts	-11,052.63
MSE	261,059,459.22

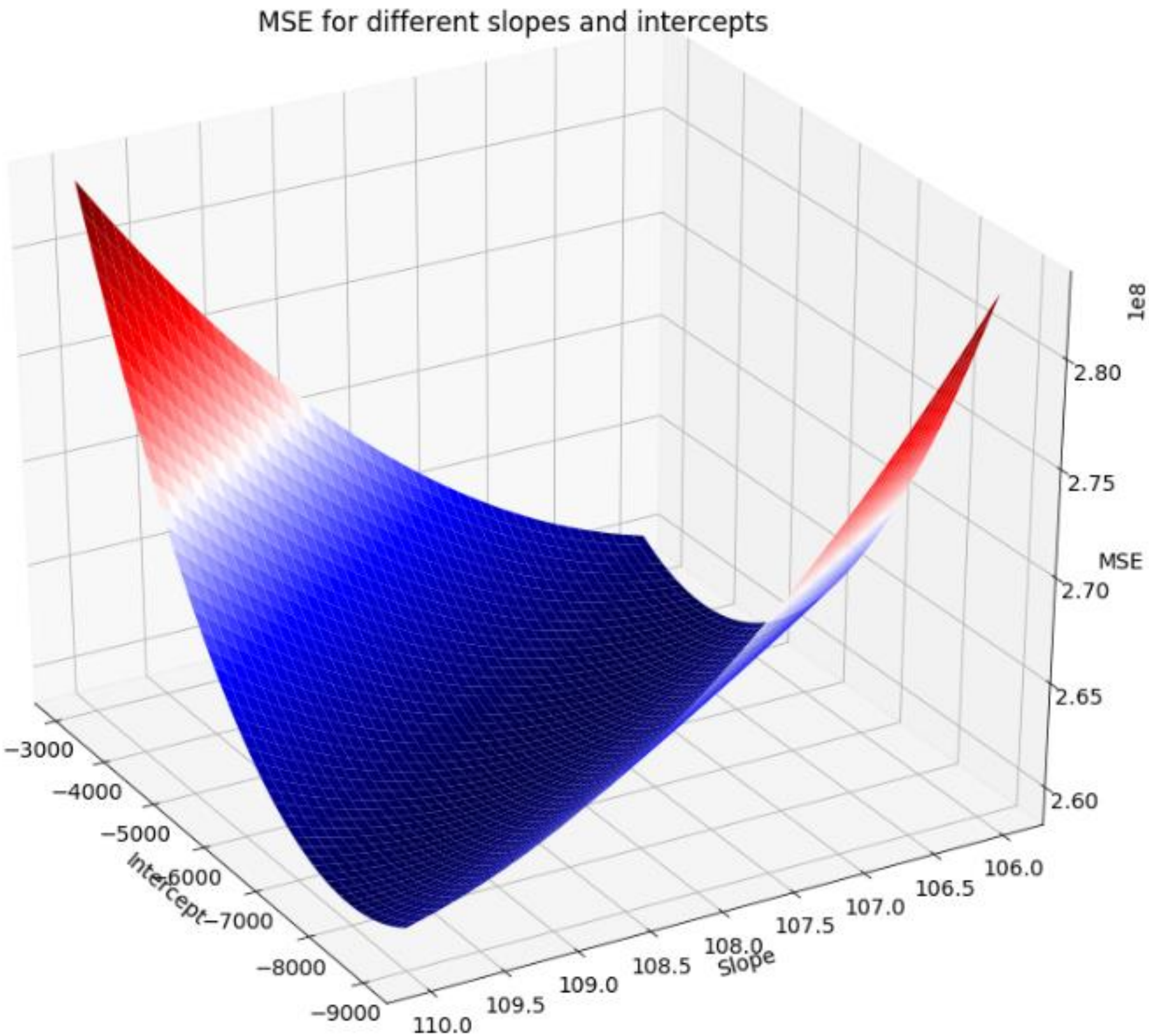
Approach 2: Gridsearch



Point with minimum MSE:

	128
Slopes	106.32
Intercepts	-4,736.84
MSE	258,939,860.54

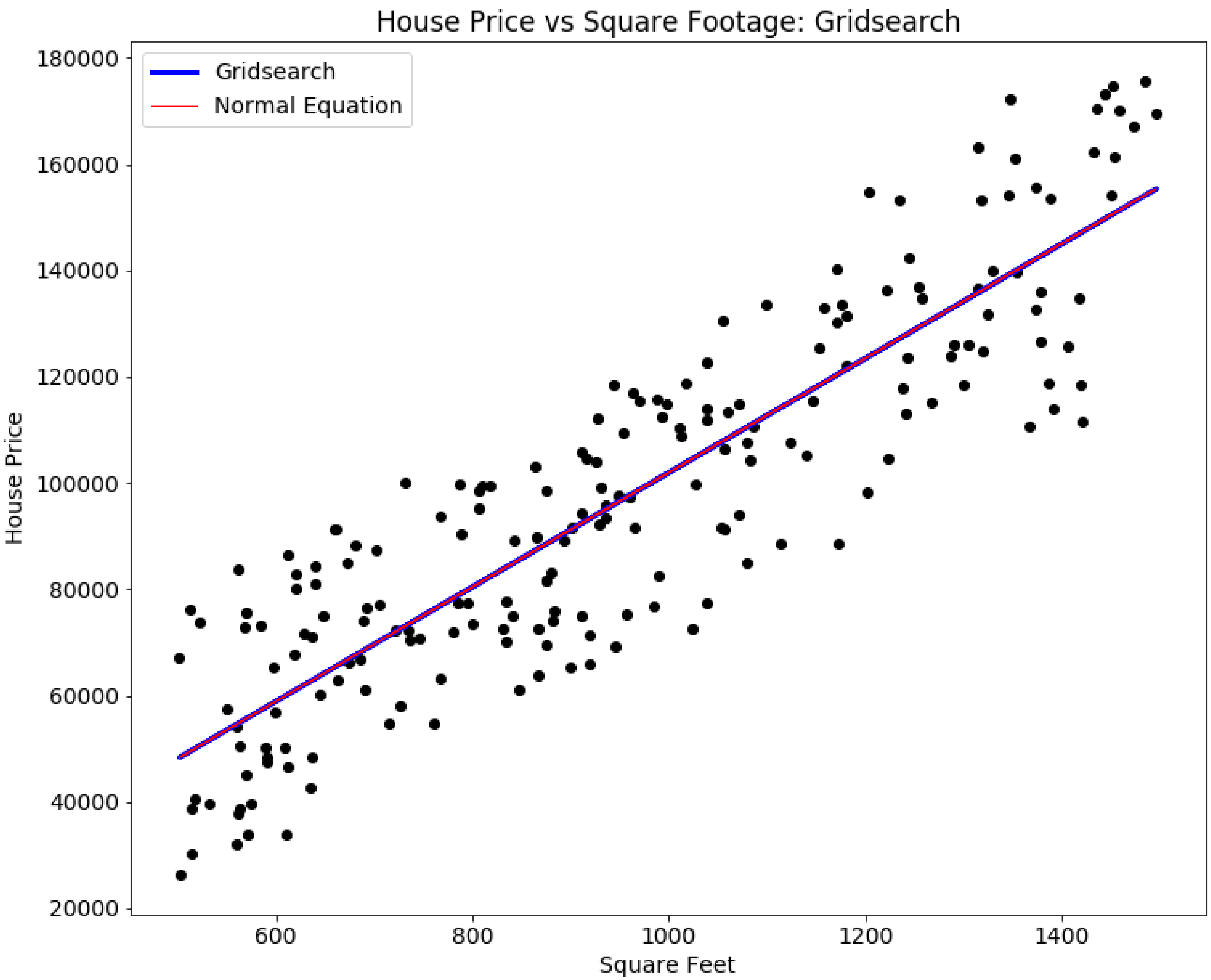
Approach 2: Gridsearch




```
Point with minimum MSE:
```

	1708
Slopes	107.68
Intercepts	-5,743.72
MSE	258,689,013.27

Approach 2: Gridsearch



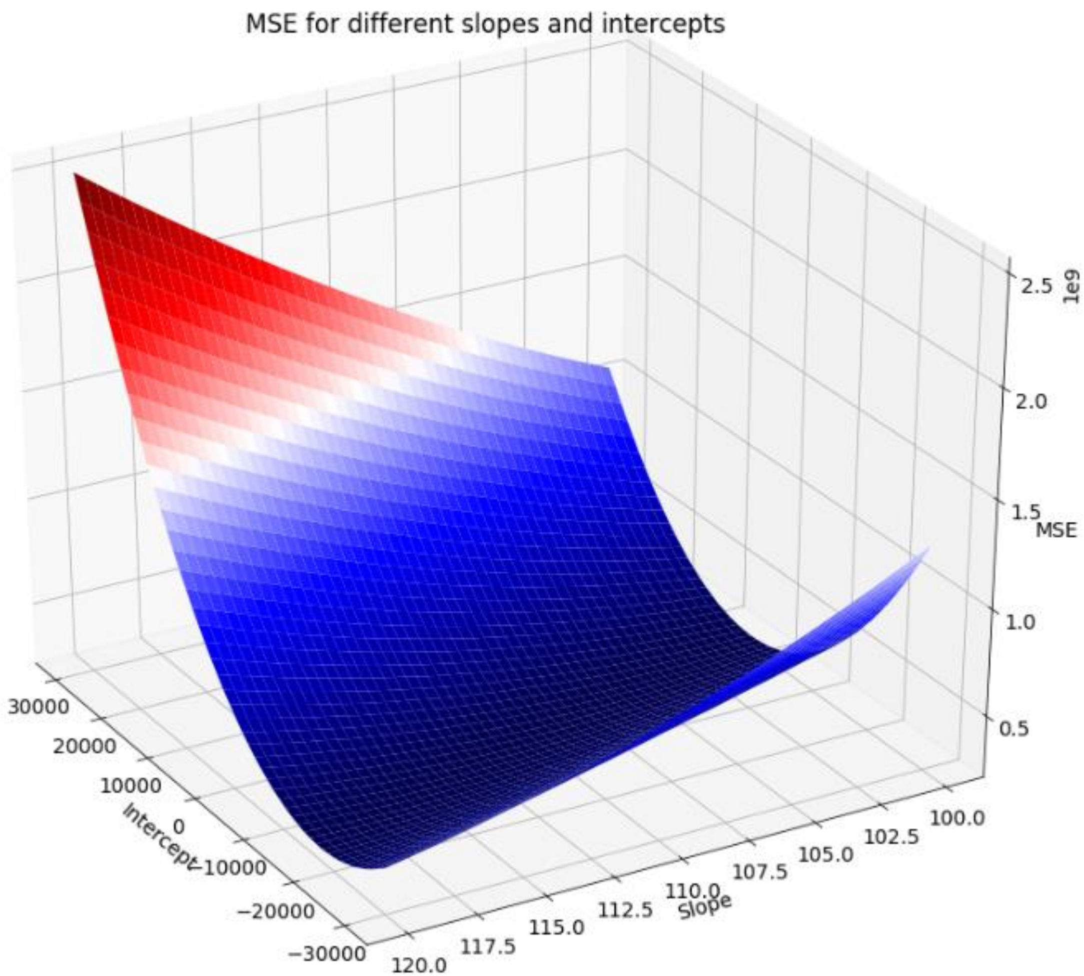
Point with minimum MSE:

	1708
Slopes	107.68
Intercepts	-5,743.72
MSE	258,689,013.27

Approach 2: Gridsearch

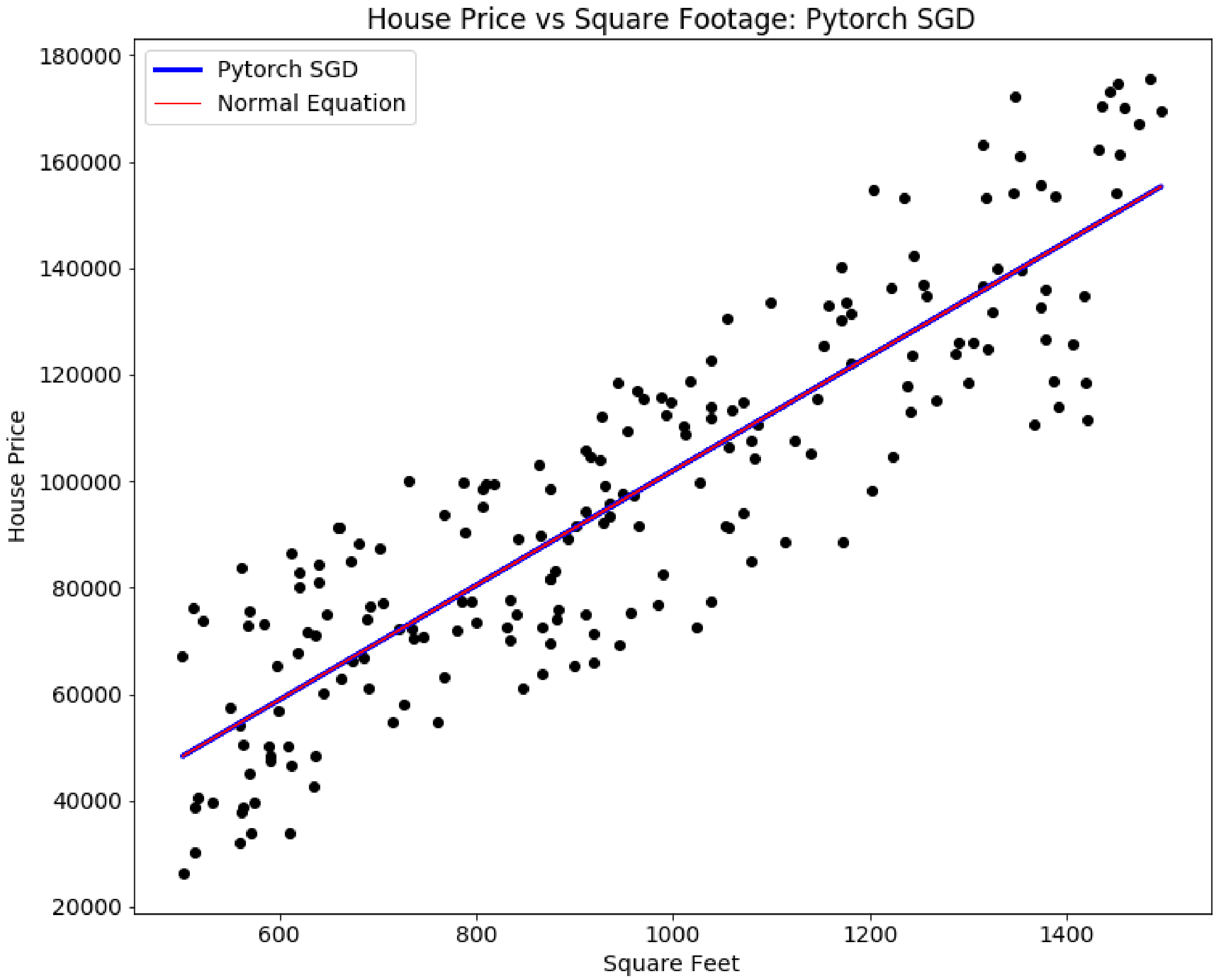
- Problem with gridsearch: Very inefficient
  - Only works for models with a handful of parameters

Approach 3: Stochastic Gradient Descent



1. You don't know the slope and intercept, so randomly choose them
2. Therefore you start at a random point
3. Calculate the slope of the MSE loss surface at that point
4. Take a step downhill
5. Repeat 3 and 4 until you reach the lowest point on the loss surface

Approach 3: Stochastic Gradient Descent



SGD gives exact same answer as Normal Equation in this example

SGD: Python Code

```python
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn   = torch.nn.MSELoss()                # Use Mean-Squared-Error loss metric

for i in range(50000):                        # Take 50k steps downhill
    y_hat = a*x + b                           # Model
    loss = loss_fn(y_hat, y)                  # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                           # Calculate slope of loss surface
    optimizer.step()                          # Step downhill
```

Approach 3: Stochastic Gradient Descent

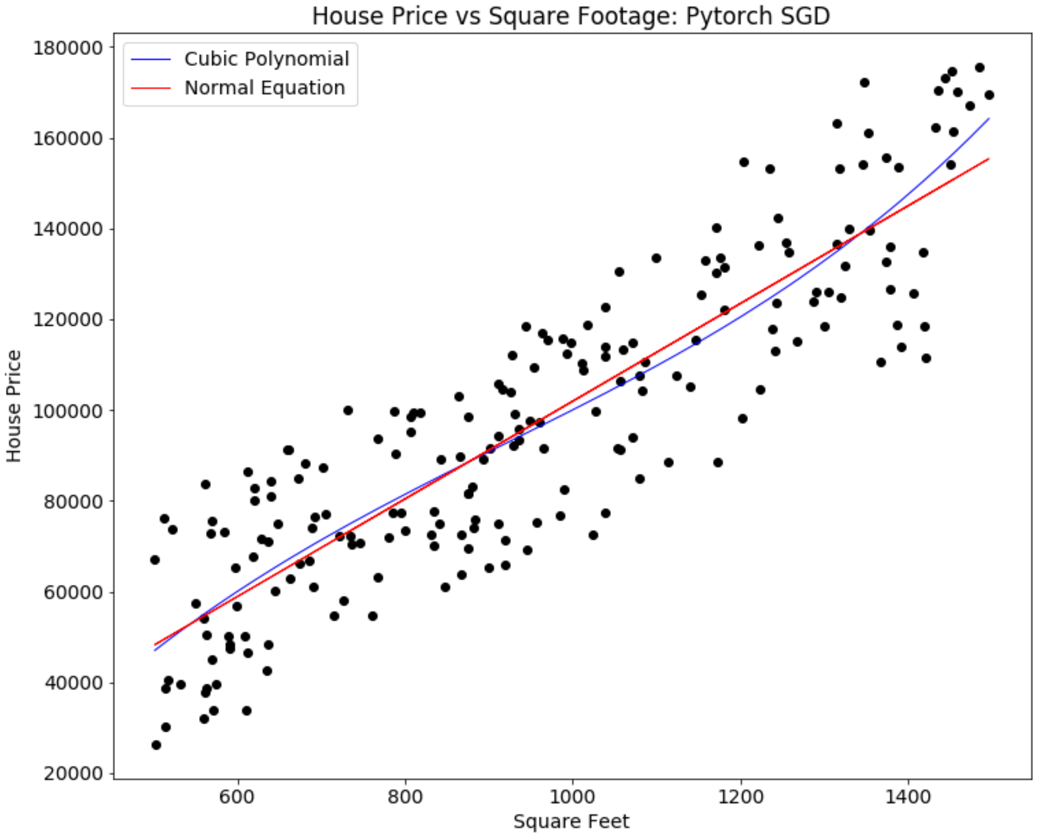
SGD: Cubic Polynomial

```
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)
c = Variable(torch.ones(1,1), requires_grad=True)
d = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b, c, d],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()                         # Use Mean-Squared-Error loss metric

for i in range(100000):                              # Take 100k steps downhill
    y_hat_2 = a*x**3 + b*x**2 + c*x + d              # Model
    loss = loss_fn(y_hat_2, y)                       # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                                  # Calculate slope of MSE loss surface
    optimizer.step()                                 # Step downhill
```

SGD: Cubic Polynomial



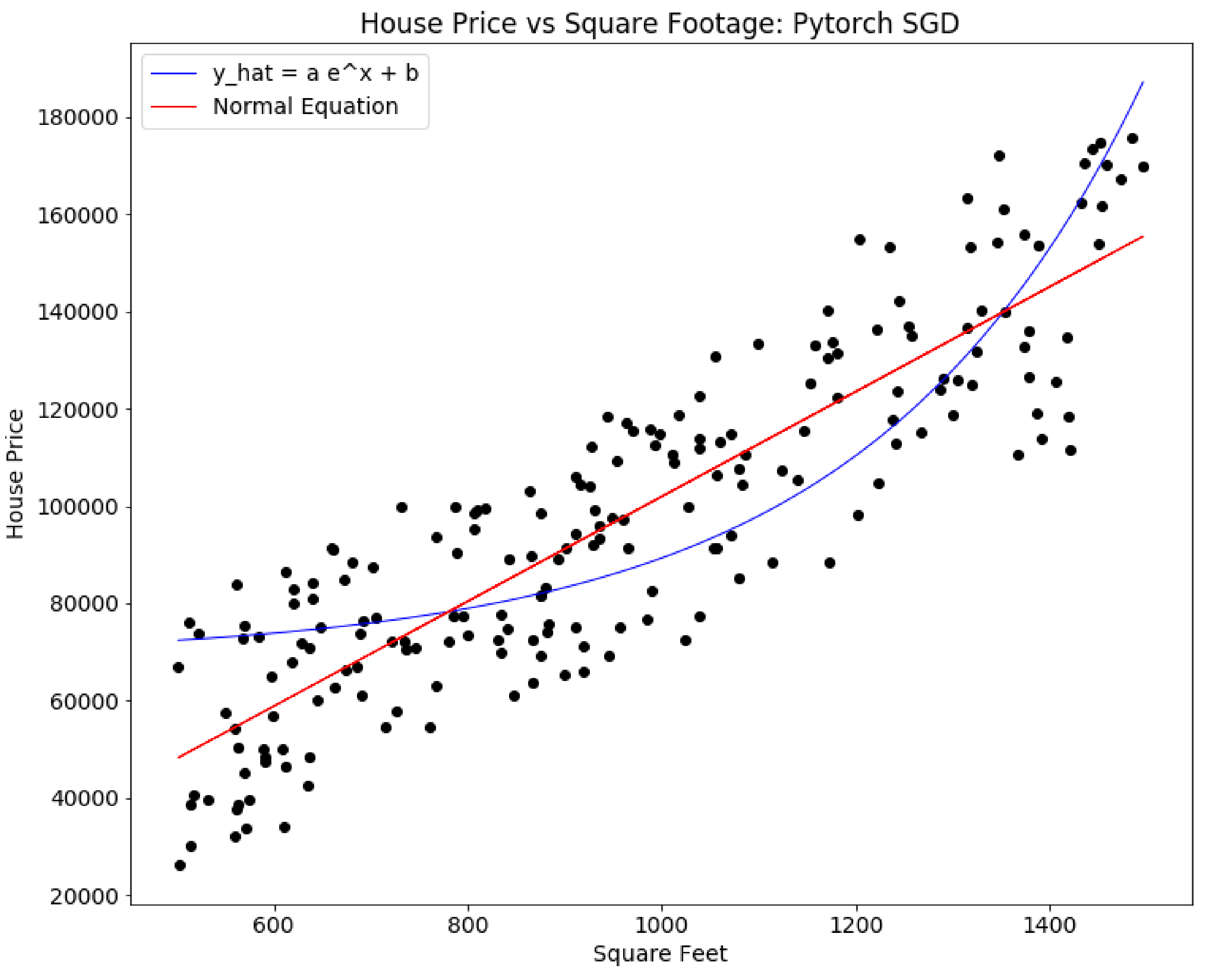
SGD: Exponential Model

```python
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()      # Use Mean-Squared-Error loss metric

for i in range(50000):            # Take 50k steps downhill
    y_hat_2 = a*np.exp(x) + b     # Model
    loss = loss_fn(y_hat_2, y)    # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()               # Calculate slope of MSE loss surface
    optimizer.step()              # Step downhill
```

SGD: Exponential Curve



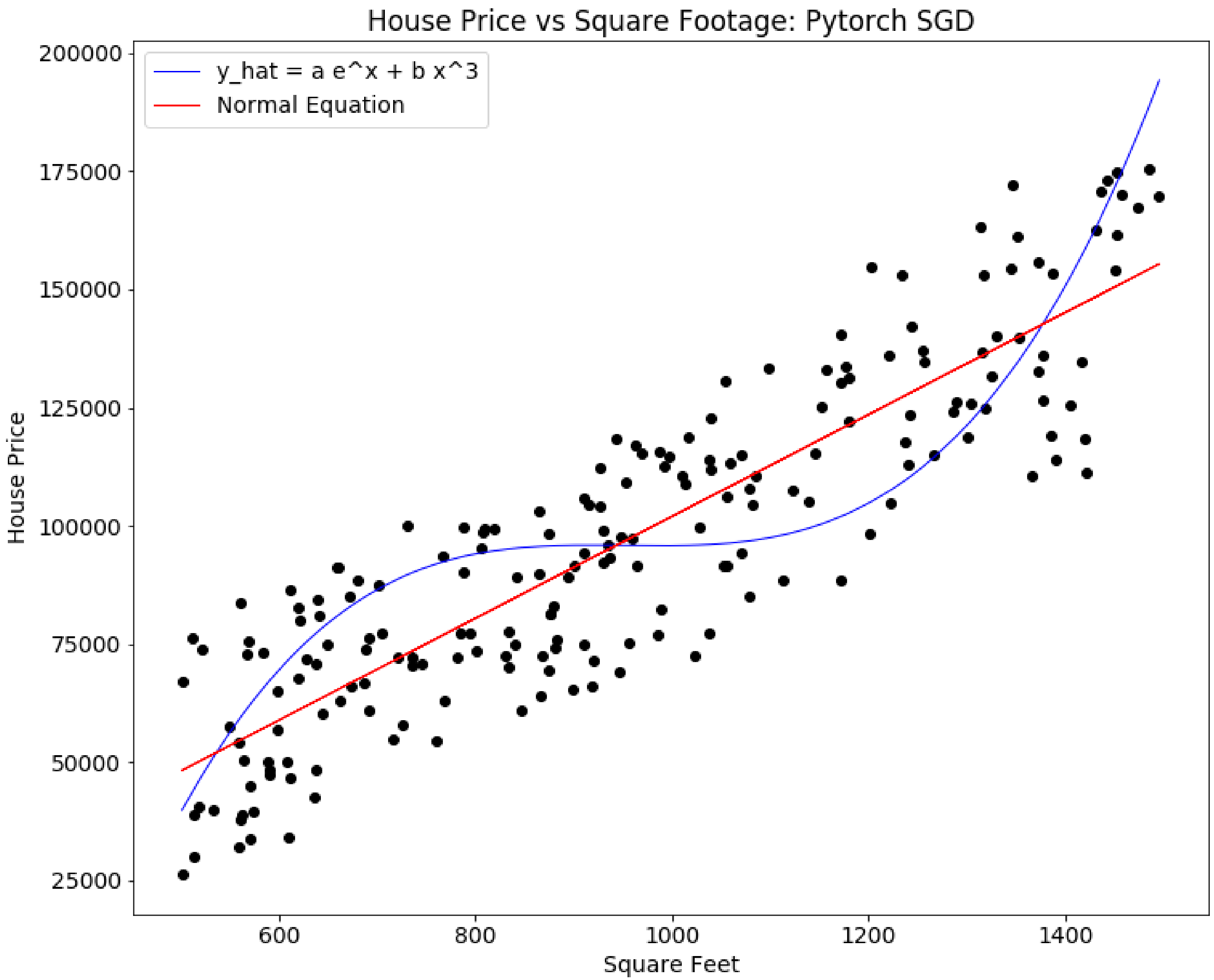
SGD: Exponential Plus Cubic Model

```python
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()                  # Use Mean-Squared-Error loss metric

for i in range(100000):                       # Take 100k steps downhill
    y_hat_2 = a*np.exp(x) + b*x**3            # Model
    loss = loss_fn(y_hat_2, y)                # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                           # Calculate slope of MSE loss surface
    optimizer.step()                          # Step downhill
```

SGD: Exponential Plus Cubic Model



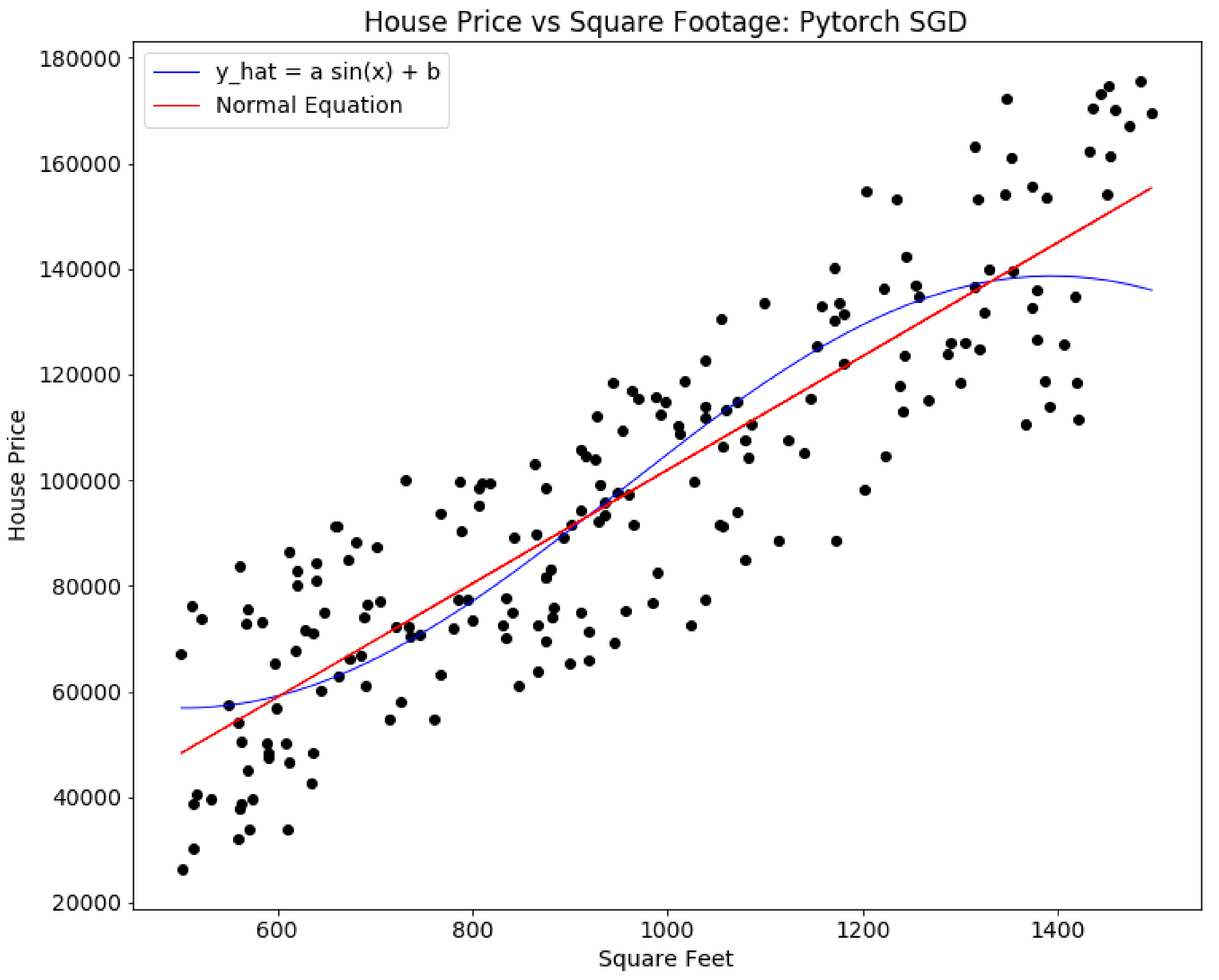
SGD: Sine Regression

```
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # # Use SGD machine-learning algorithm
loss_fn = torch.nn.MSELoss()       # Use Mean-Squared-Error loss metric

for i in range(50000):             # Take 50k steps downhill
    y_hat_2 = a*np.sin(x) + b      # Model
    loss = loss_fn(y_hat_2, y)     # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                # Calculate slope of MSE loss surface
    optimizer.step()               # Step downhill
```

SGD: Python Code



SGD: Mathematical Background

$$MSE = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(ax_i + b - y_i)^2$$

Calculate the partial derivative of the loss function with respect to each of its parameters. This tells you the slope of the loss surface wrt each of parameters.

$$\frac{\partial MSE}{\partial a} = \frac{1}{m}\sum_{i=1}^{m}(ax_i + b - y_i)x_i$$

$$\frac{\partial MSE}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(ax_i + b - y_i)$$

Then move towards the lowest point on the loss surface by taking small steps downslope.

- If the slope is positive, reduce the parameter.
- If the slope is negative, increase the parameter.

SGD: Python Code

```
a = Variable(torch.ones(1,1), requires_grad=True)
b = Variable(torch.ones(1,1), requires_grad=True)

optimizer = torch.optim.SGD([a, b],lr=0.0001) # Use SGD machine-learning algorithm
loss_fn   = torch.nn.MSELoss()                # Use Mean-Squared-Error loss metric

for i in range(50000):                        # Take 50k steps downhill
    y_hat = a*x + b                           # Model
    loss = loss_fn(y_hat, y)                  # Calculate MSE for this particular model
    optimizer.zero_grad()
    loss.backward()                           # Calculate slope of loss surface
    optimizer.step()                          # Step downhill
```

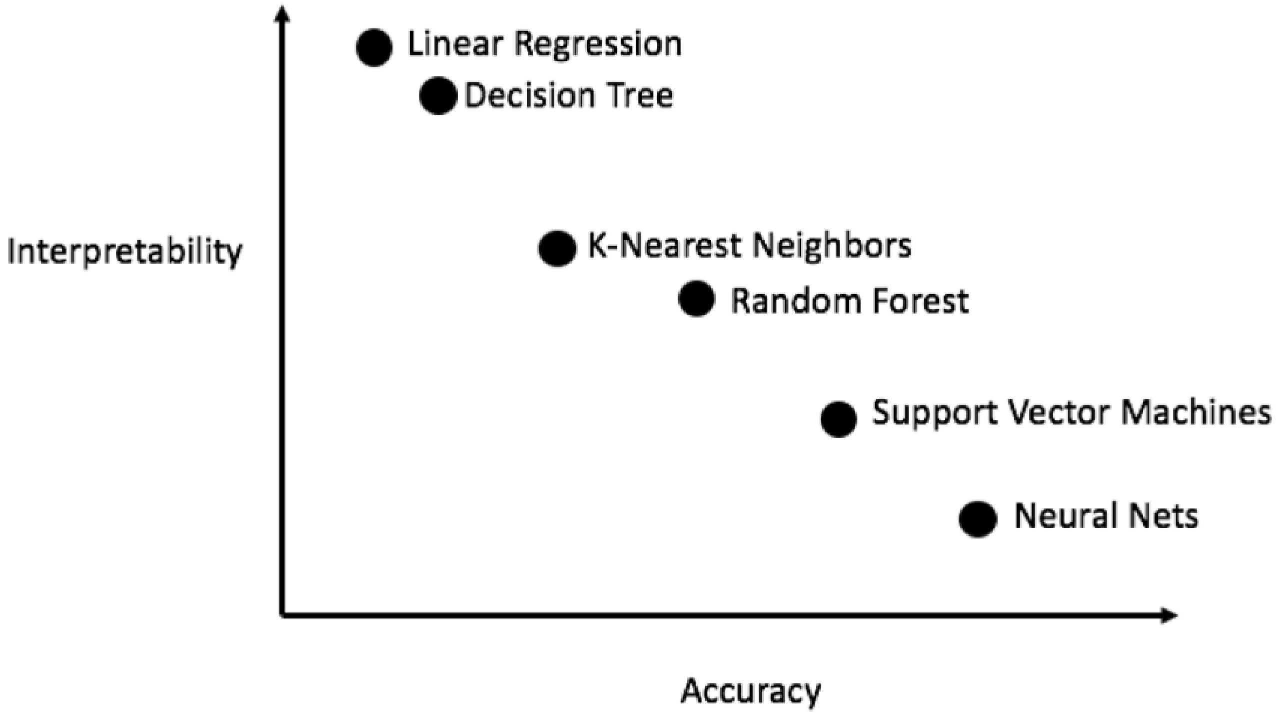
Benefits of SGD

- It is straightforward to calibrate predictive models
- You can build models with thousands of parameters
  - Can work on huge data sets
  - Can achieve human-level accuracy
- You can build models for all different types of data
  - Pictures
  - Videos
  - Audio
  - Text
  - Policyholder datafiles

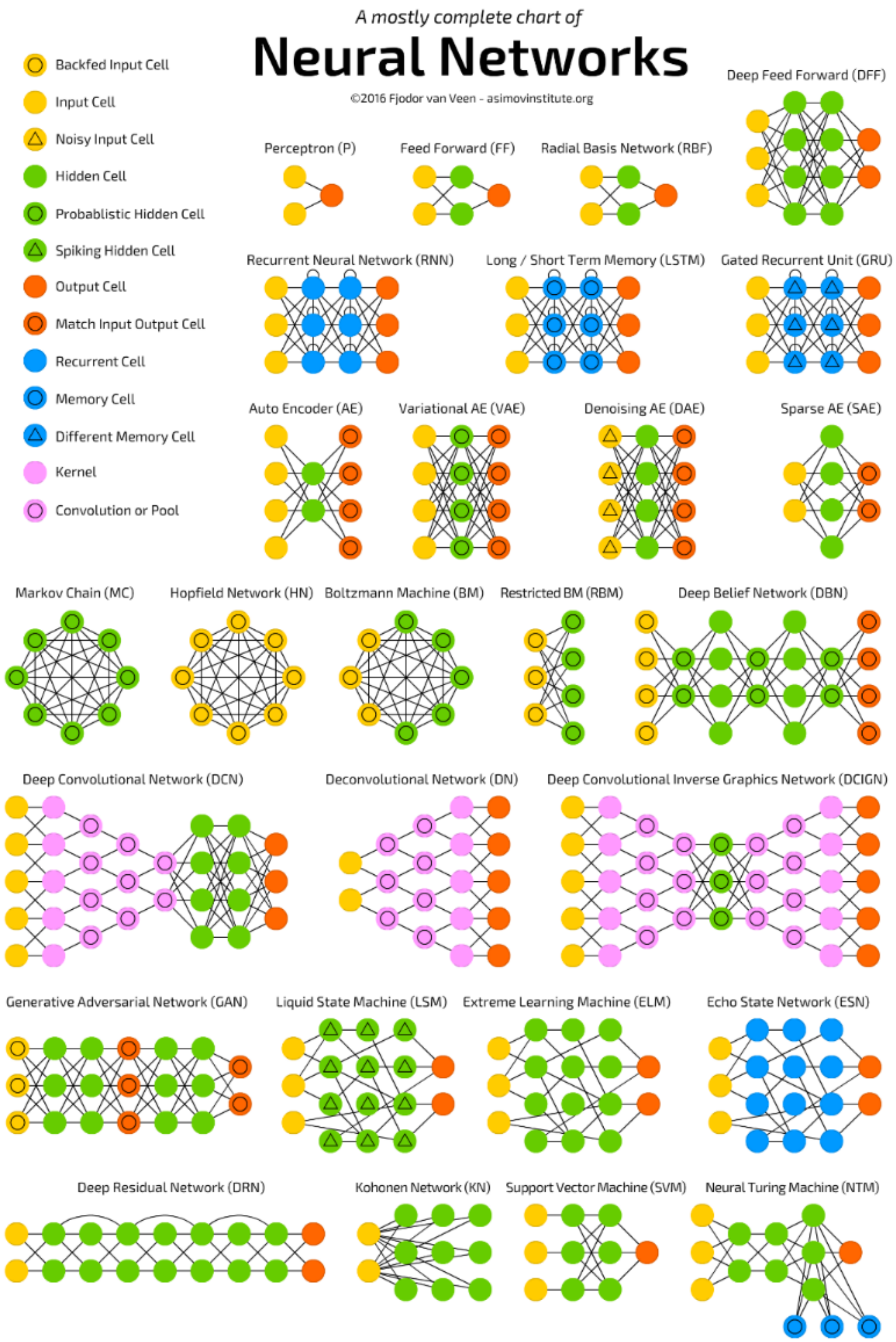
Benefits of SGD

- It works very well in practice
  - You can choose models which are a good fit to the data
  - Rather than choosing models which you are able to fit to the data

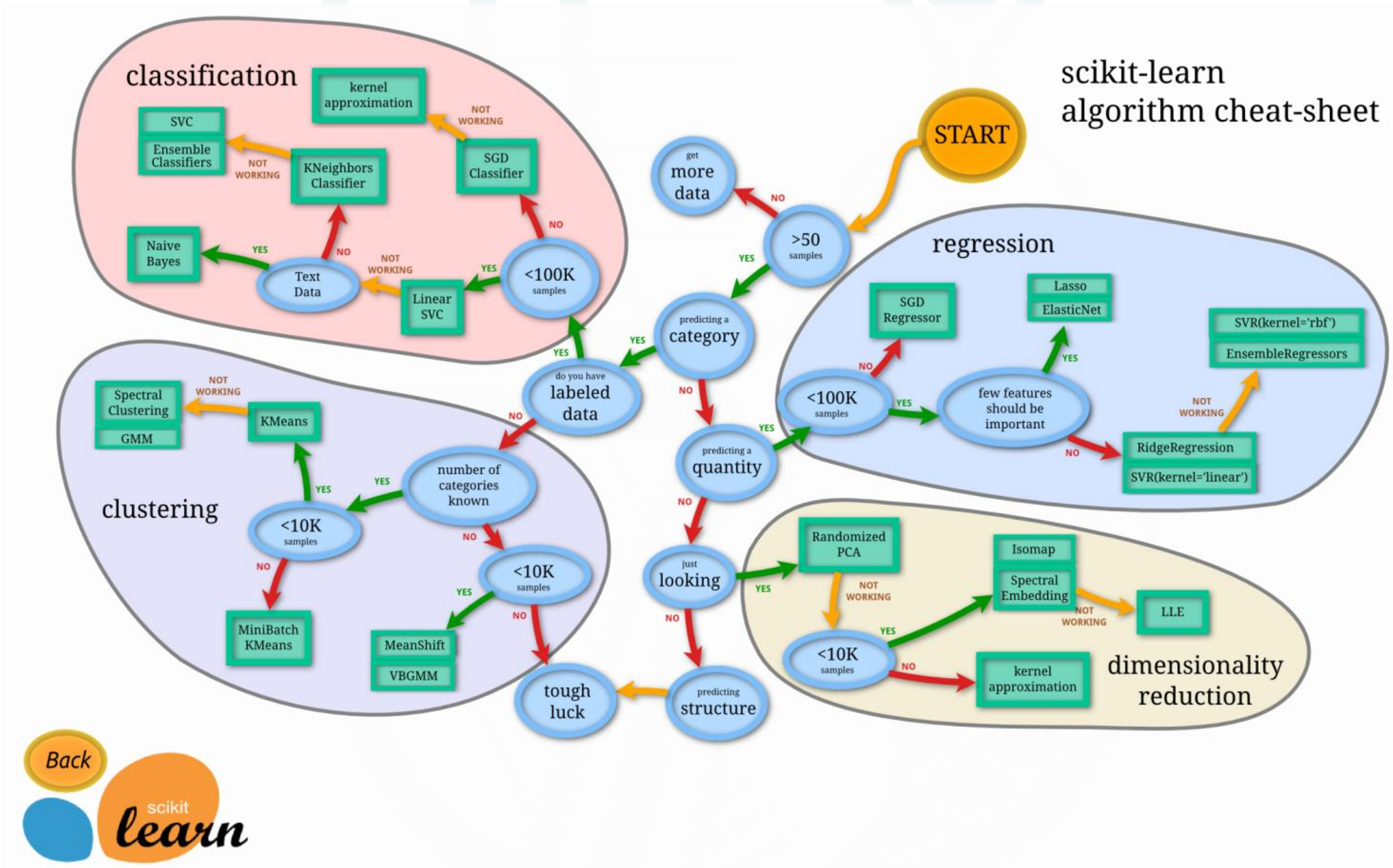
Machine Learning Models



Neural Network Models



Machine Learning Models in Scikit-Learn



Demystifying Data Science

- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples

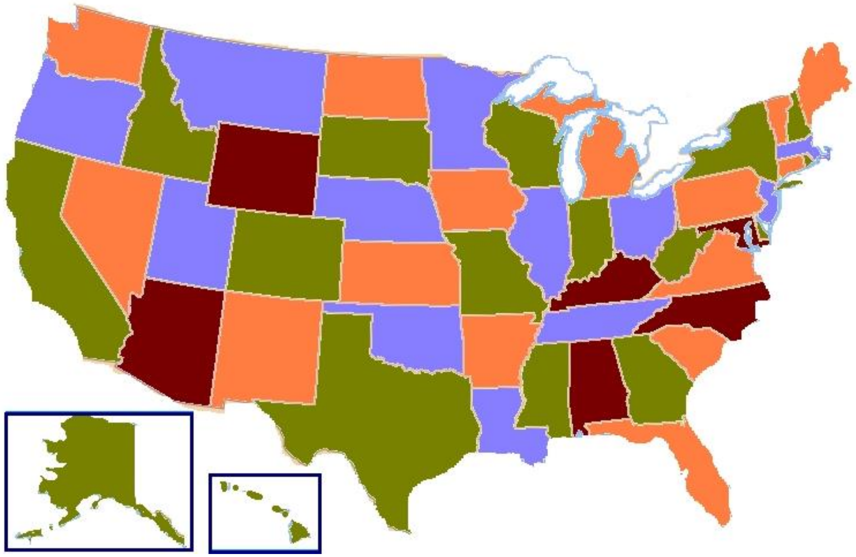
Practical Examples – Getting started

- Big Data
  - More Data
  - More Computing Power
  - More Analysis
- Computers in Actuarial Work
- A Word on Terminology
- Association Rule Mining
- Unsupervised Learning

The role of Computers in Actuarial Work

- Mainframe Systems
- Valuation Software
- Spreadsheets
- A precise answer…
- …given assumptions
- Computers may be able to ‘solve’ problems
- Or at least give valuable insights

Example 1 - Four Colour problem solved



- Proved in 1976
- First major theorem proved by computer

Example 2 - Fermat's Last Theorem solved (almost)

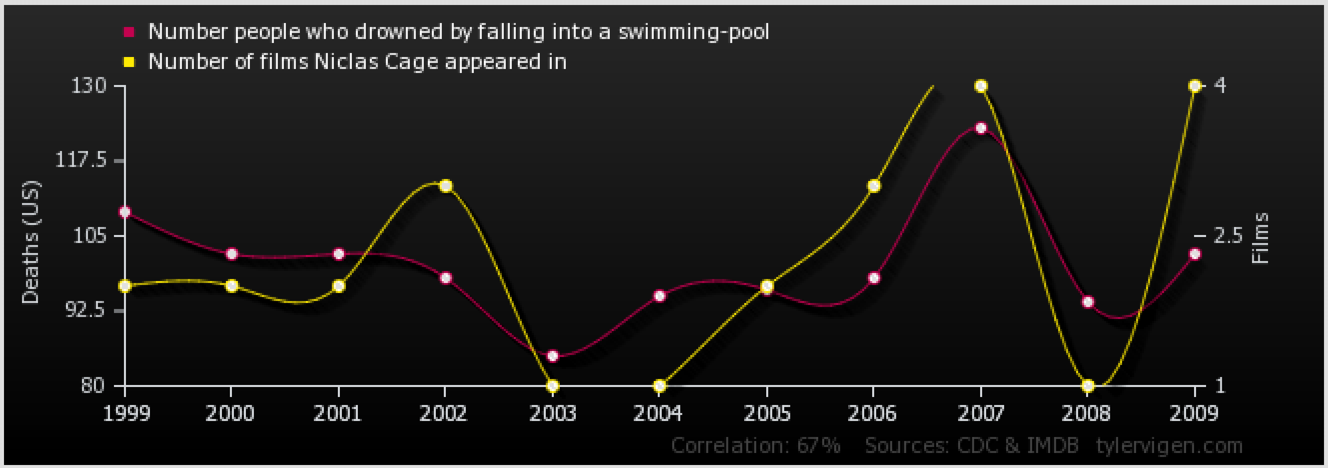
- $x^n+y^n = z^n$
- Solved by computer for all primes up to 4,000,000

Correlation and Causation!

Number people who drowned by falling into a swimming-pool

correlates with

Number of films Nicolas Cage appeared in



Upload this chart to imgur

	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Number people who drowned by falling into a swimming-pool Deaths (US) (CDC)	109	102	102	98	85	95	96	98	123	94	102
Number of films Nicolas Cage appeared in Films (IMDB)	2	2	2	3	1	1	2	3	4	1	4

Correlation: 0.666004

- Results always need to be interpreted!

http://tylervigen.com/spurious-correlations

A word on Terminology

- Actuaries didn't get here first!
- P = A / ä

Periodic Policy Amount =

Bounded Risk Benefit /

Contribution Vector

- Terminology not intuitive...
- ...concepts are

What we're looking to cover

This presentation

- Association Rule Mining (Amazon, Tesco)
- Unsupervised Learning
  - Letting the data tell its own story

Next presentation

- Supervised Learning
  - Where we propose a model

Final presentation

- Deep Learning (Neural Nets)

Association Rule Mining 1

- Purchasing datasets

	Bread	Milk	Eggs	…	Yoghurt	Tuna	Fruit
Customer 1	x						
Customer 2	x	x					x
Customer 3			x			x	
: :		x					
Customer n					x		

- Very very sparse
- Think of Amazon

Association Rule Mining 2

- Of interest, what items occur together?
- As a purchasing dataset will have very sparse data, ideas will be illustrated by a medical dataset
- 240 Patients
- 6 Symptoms

Association Rule Mining Dataset

- Illustrative dataset

	Symptoms					
	1	2	3	4	5	6
Patient 1					x	
Patient 2		x		x		
Patient 3			x	x		x
: :	: :	: :	: :	: :	: :	: :
Patient 240				x	x	
Total	**19**	**157**	**55**	**85**	**58**	**181**

- Less sparse

Association Rule Mining Investigation

- Which symptoms occur together?
- Three key concepts...

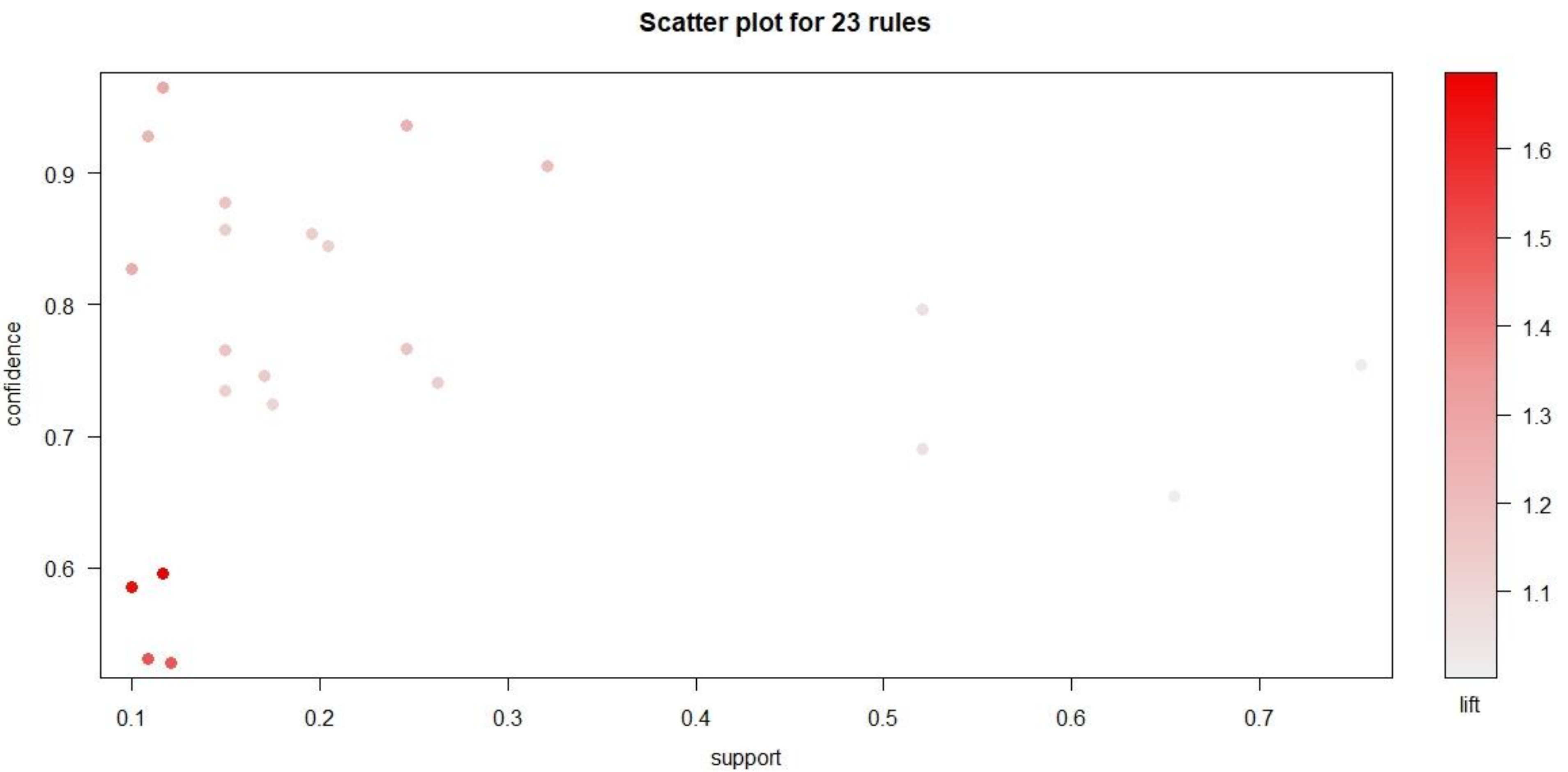
For symptoms A & B

1) **Support** = $P(A \cap B) = P(A,B)$

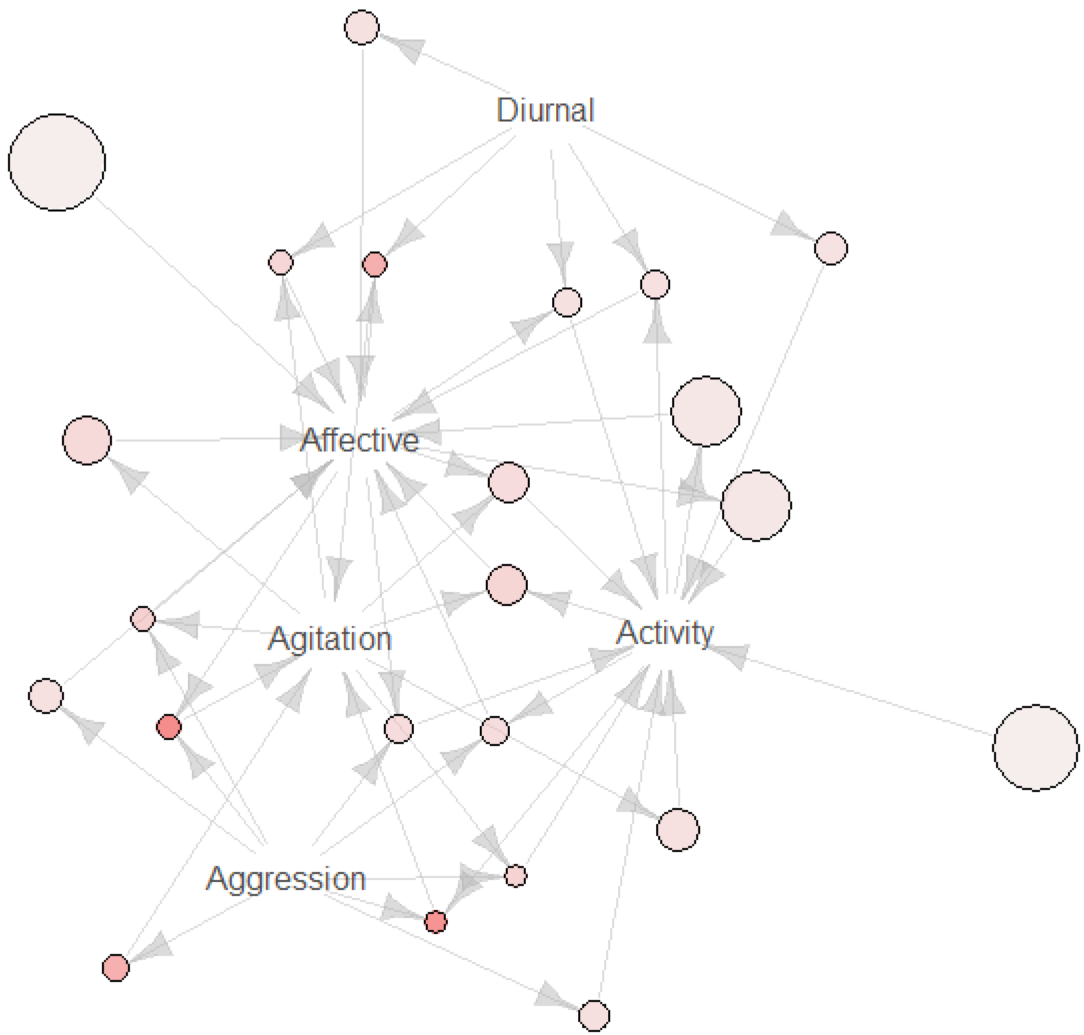
2) **Confidence** = $P(B|A) = P(A,B) / P(A)$

3) **Lift** = $P(A,B) / [P(A).P(B)]$

Association Rule Mining Result 1



Association Rule Mining Result 2



Association Rule Summary

- Concepts are not difficult
- Terminology and visualisation can be confusing at first
- Basic analysis can be enhanced by adding bounds and standardising results
- Very sophisticated algorithms can be developed but speed is an issue

~~___~~

What we’re looking to cover, a reminder

Unsupervised Learning

- No y value, Multiple x values

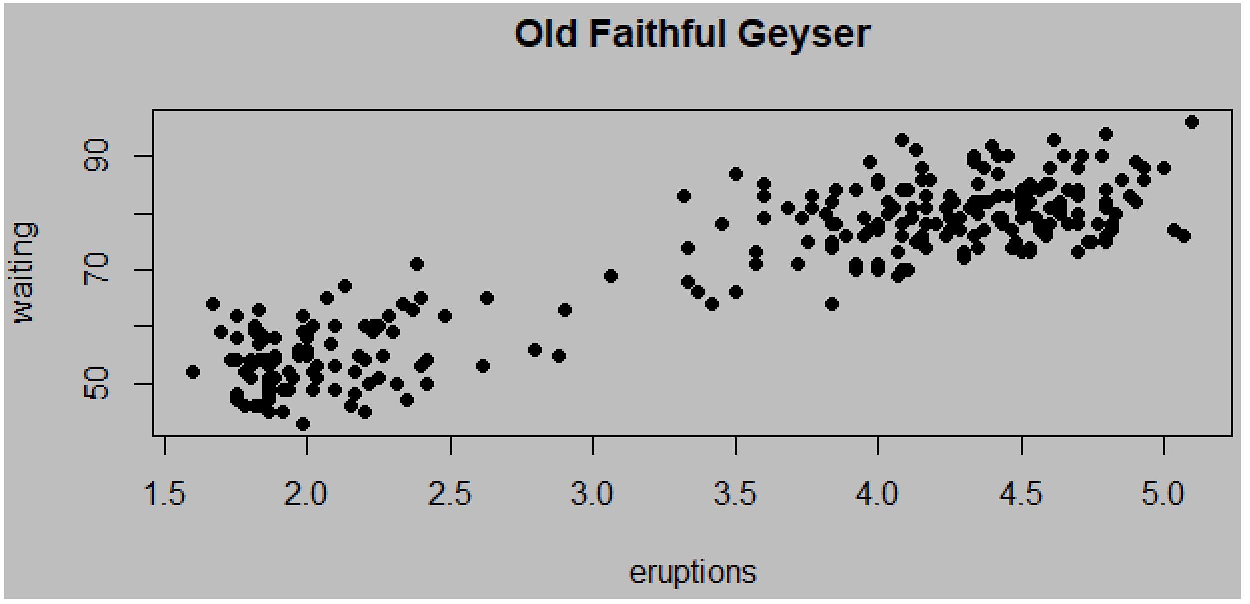
Supervised Learning

- We do have a y value & multiple x values

Unsupervised Learning 1

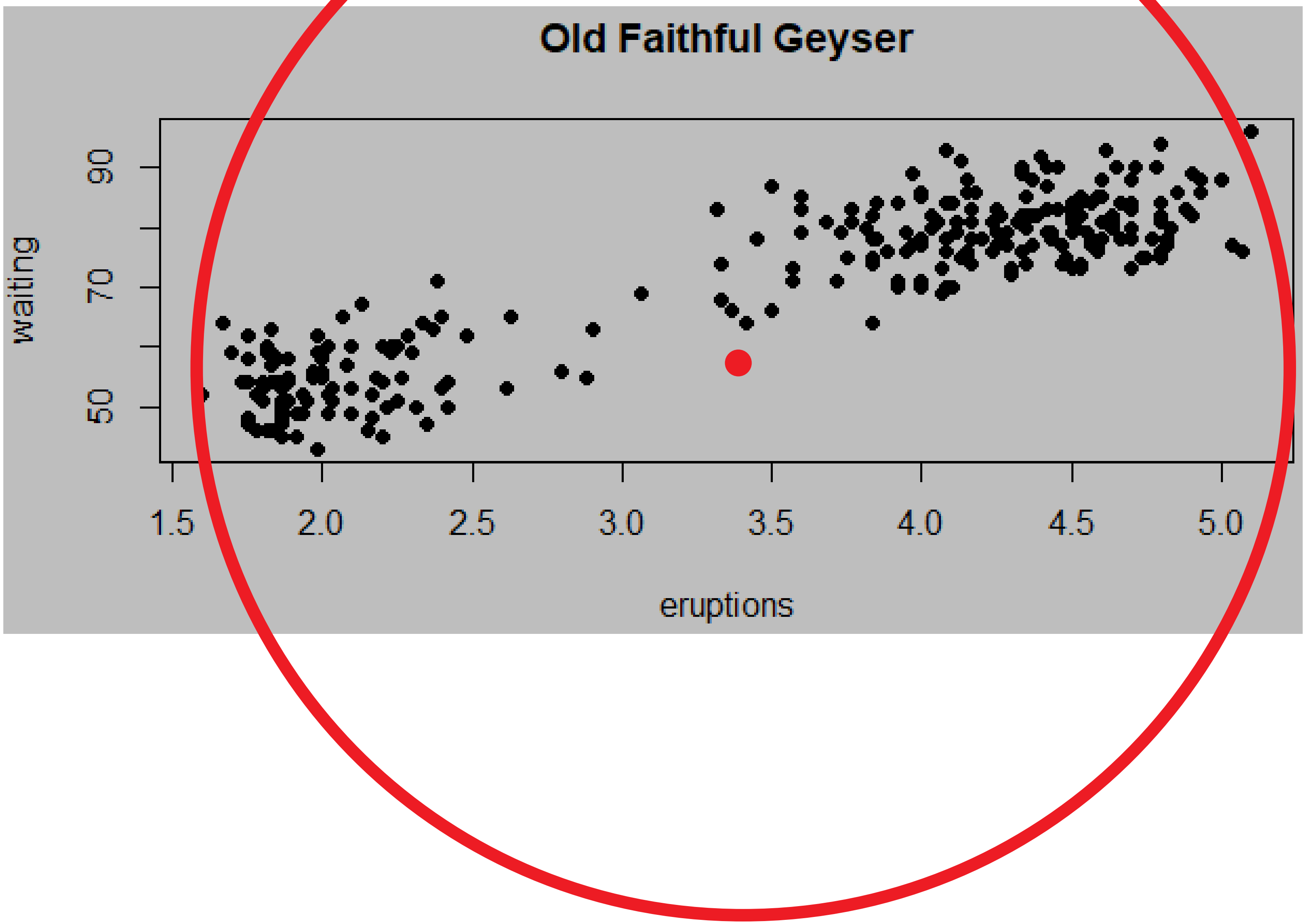
- Old Faithful Geyser
- 272 data points on Waiting & Eruption Times

Unsupervised Learning 2

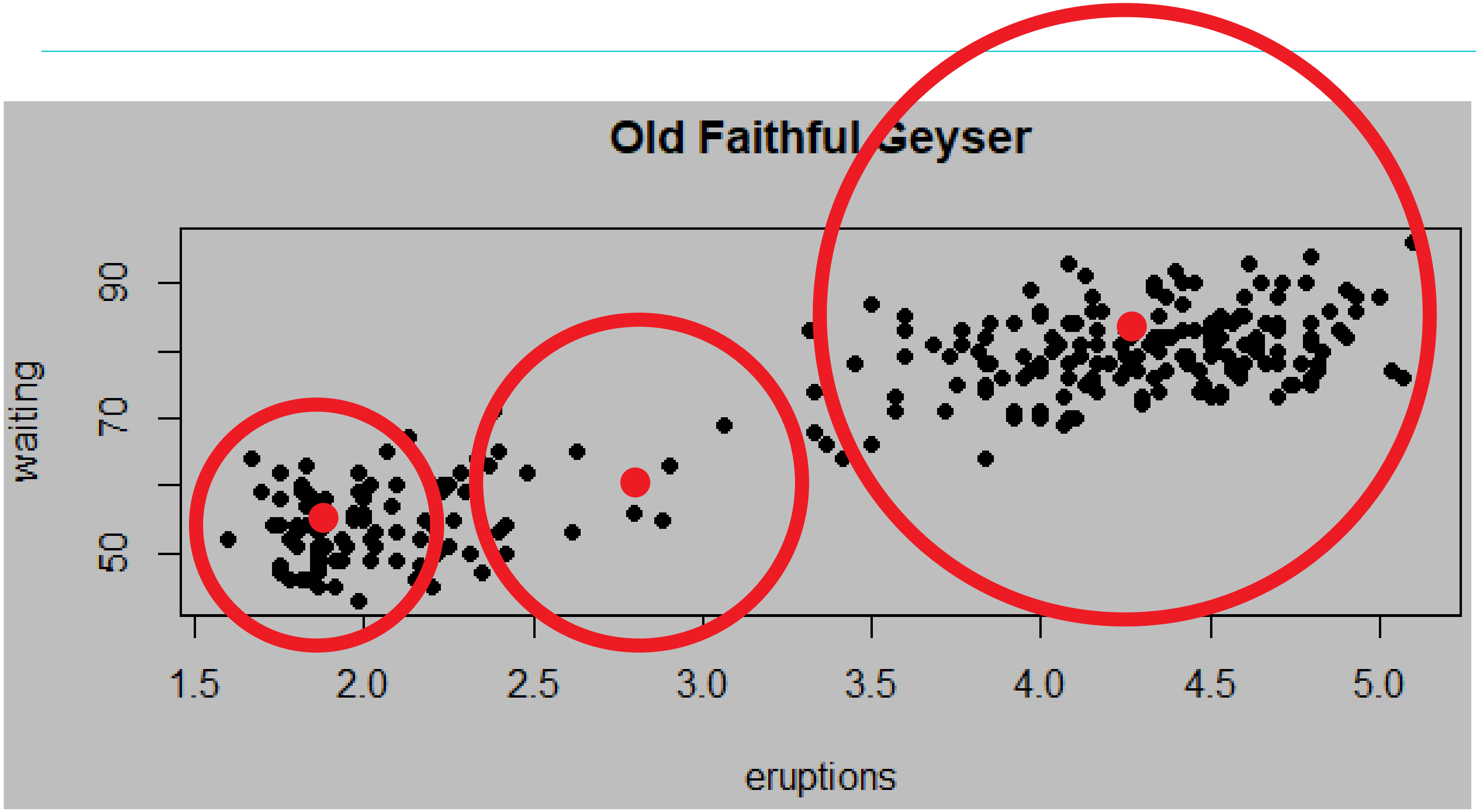


- Old Faithful Geyser
- 272 data points on Waiting & Eruption Times

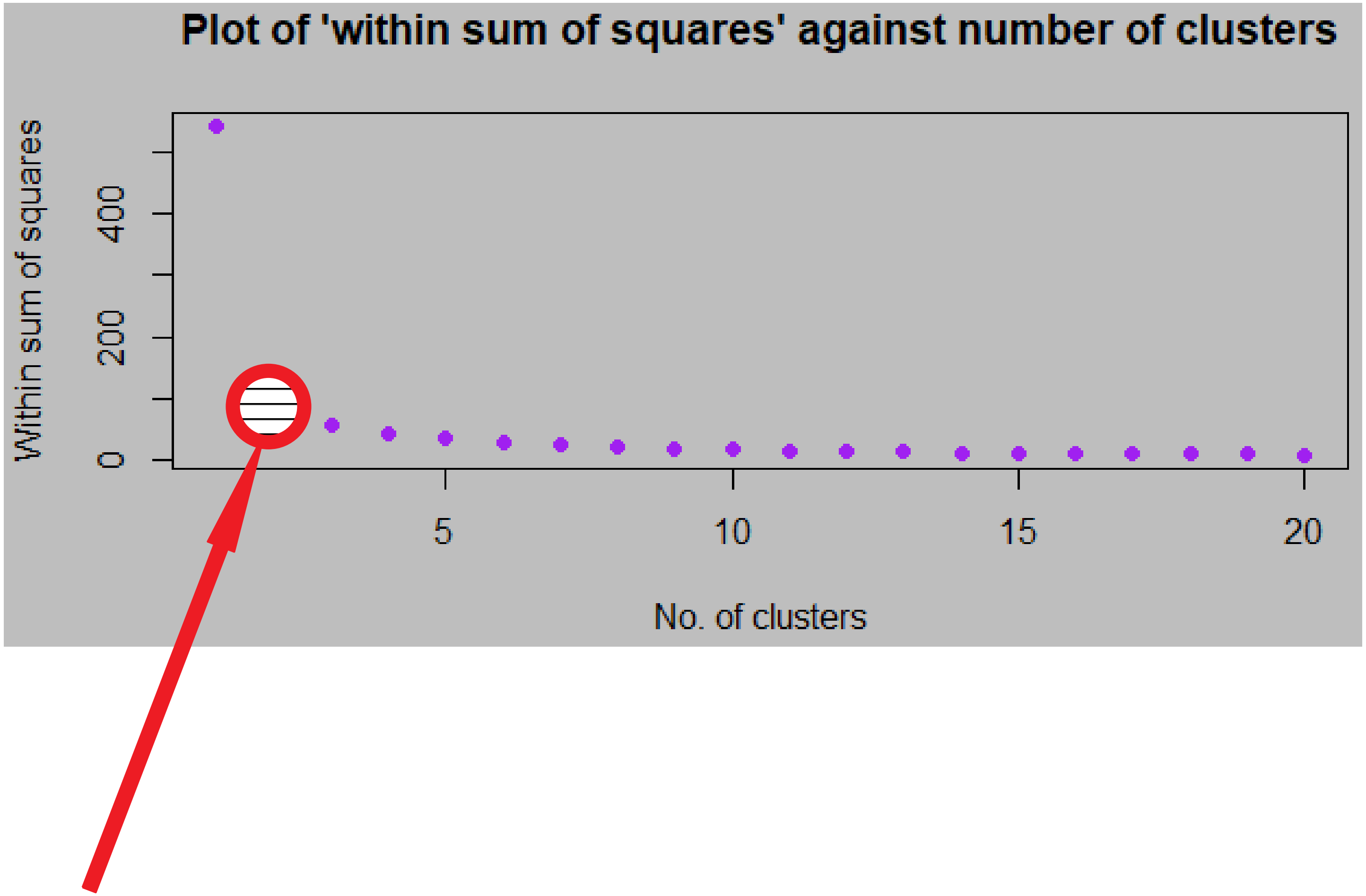
Unsupervised Learning 3



Unsupervised Learning 4



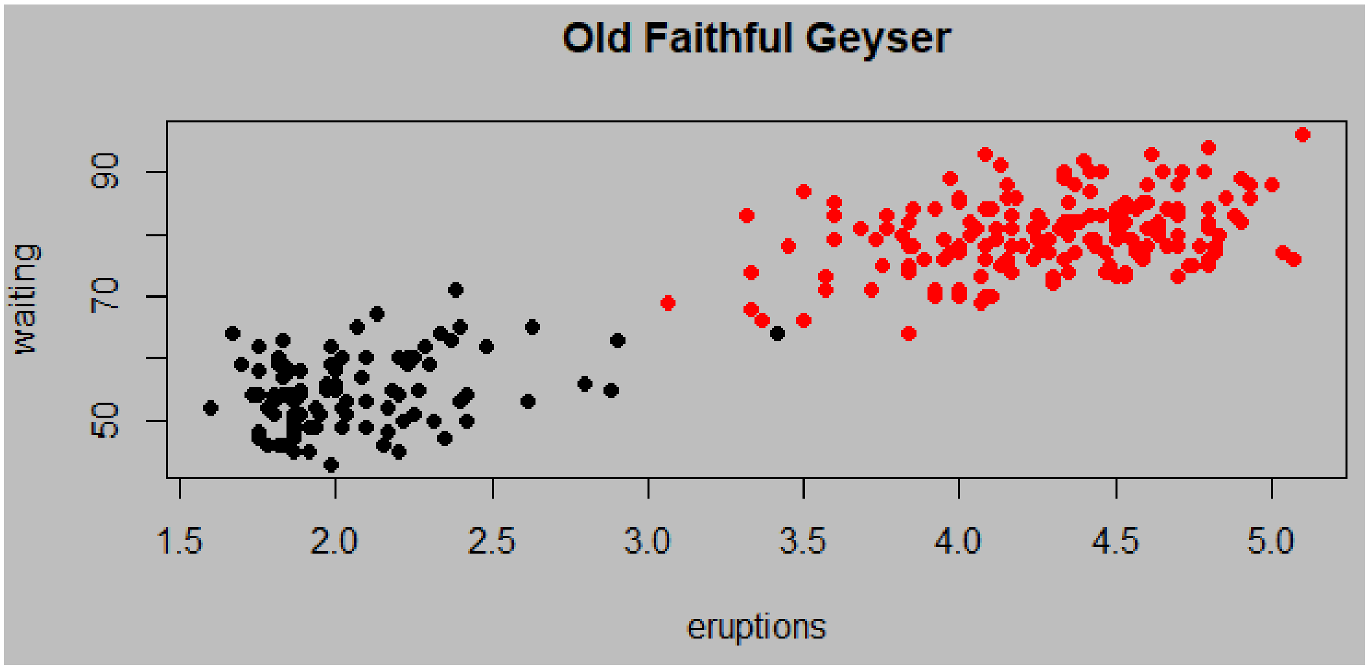
Unsupervised Learning 5



'Elbow'

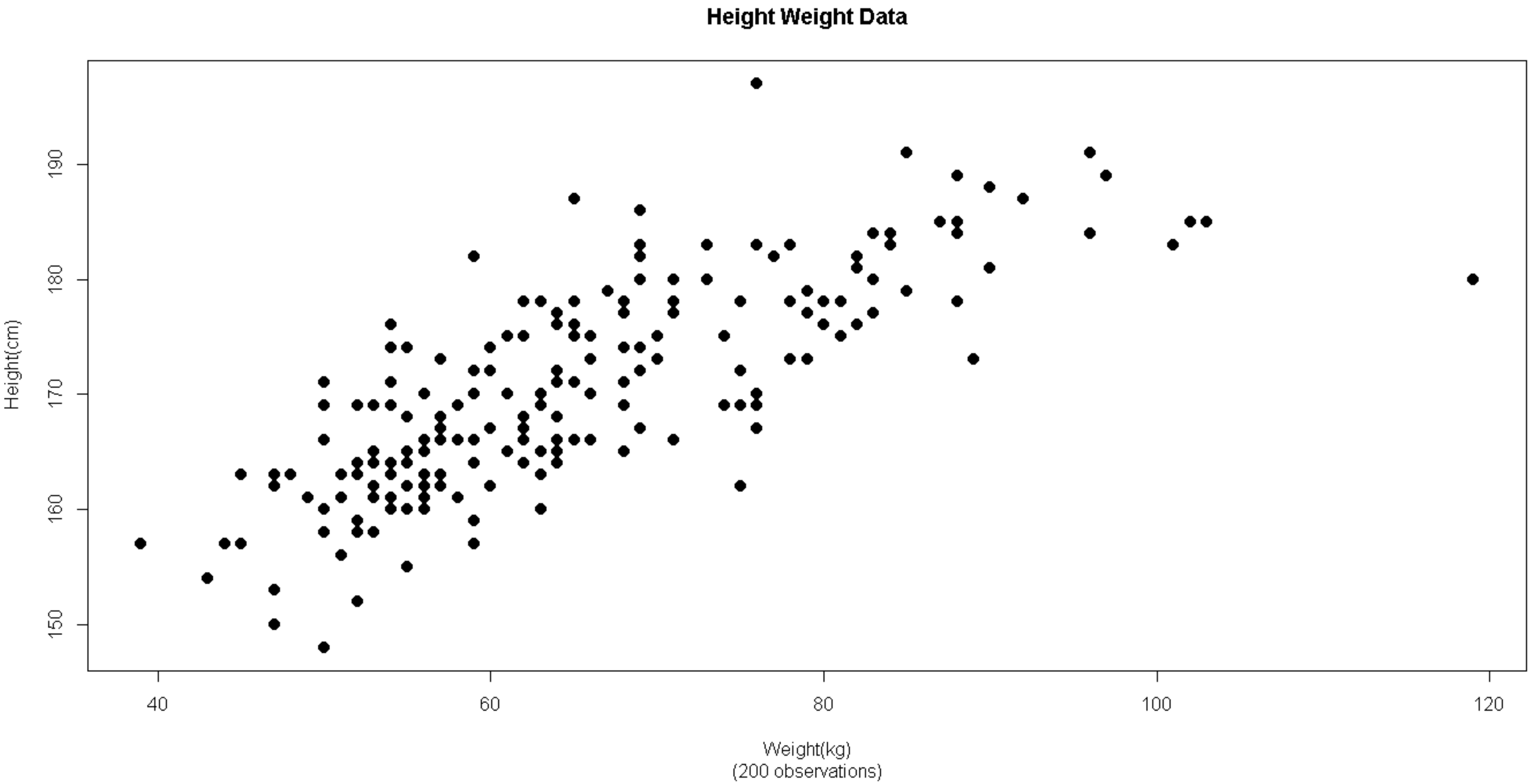
Unsupervised Learning 6

- Resulting Segmentation

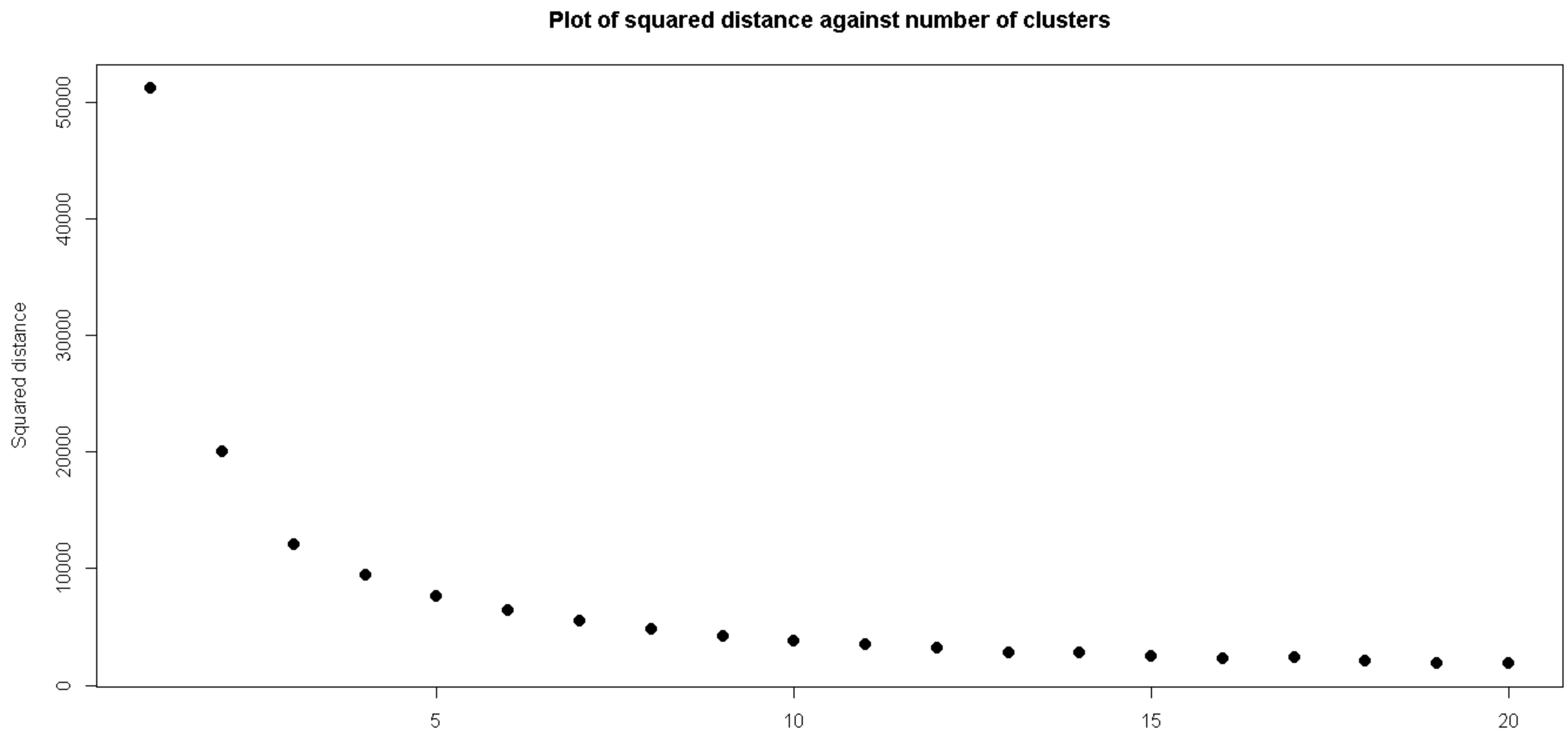


- Can be exploratory or detective

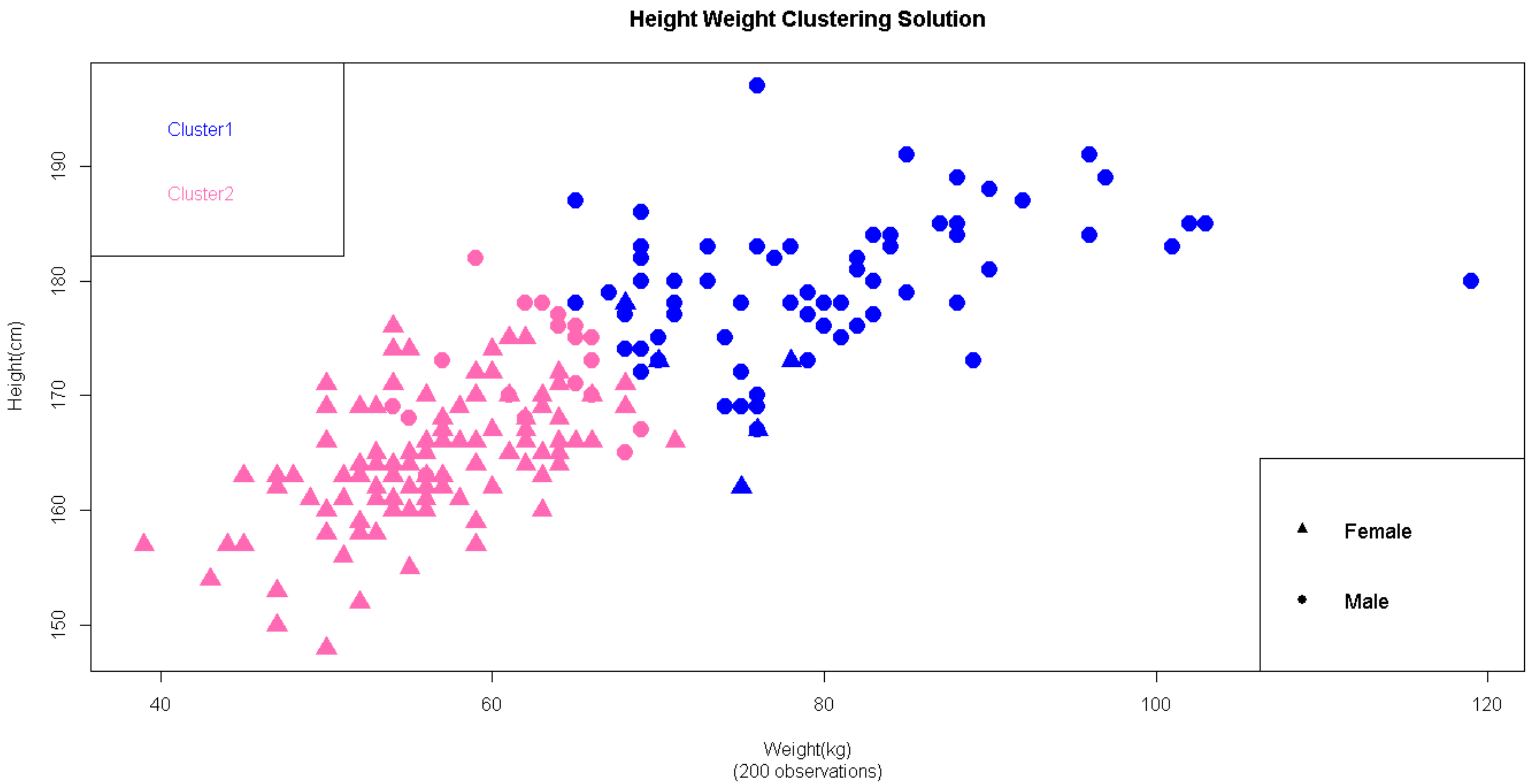
Another Grouping (Clustering) Example 1



Another Grouping (Clustering) Example 2



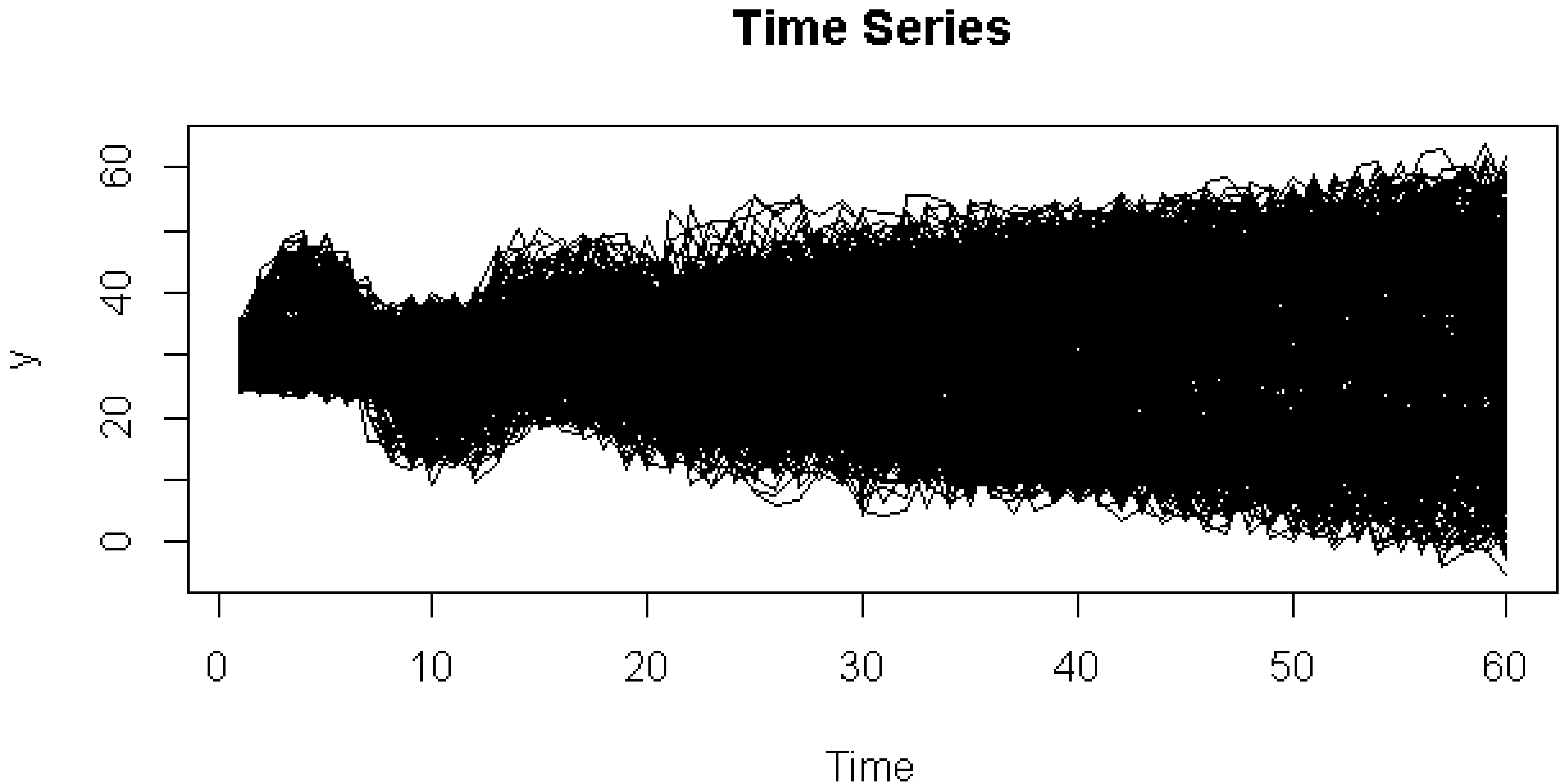
Another Grouping (Clustering) Example 3



Another Grouping (Clustering) Example 4

- Accuracy 88%
- ‘First pass’ result
- Readily implementable
- Methodology generalisable to n dimensions
- Where could this give more insight?
  - Segmentation (Distribution Channel)
  - Any homogeneous group selection
  - Deconstructing portfolios
  - Model point building
  - Outlier identification (Fraud etc.)
  - Trend analysis

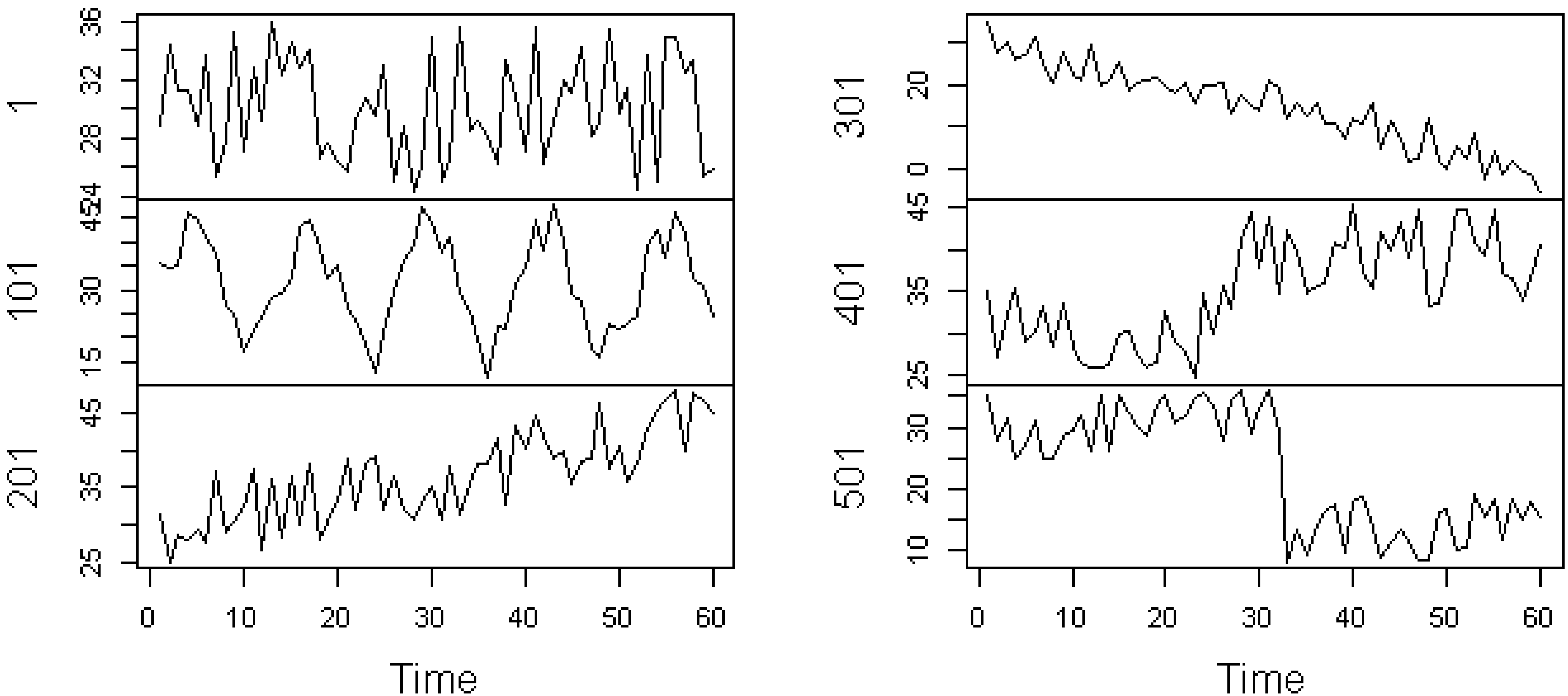
Deconstructing Trend Analysis 1



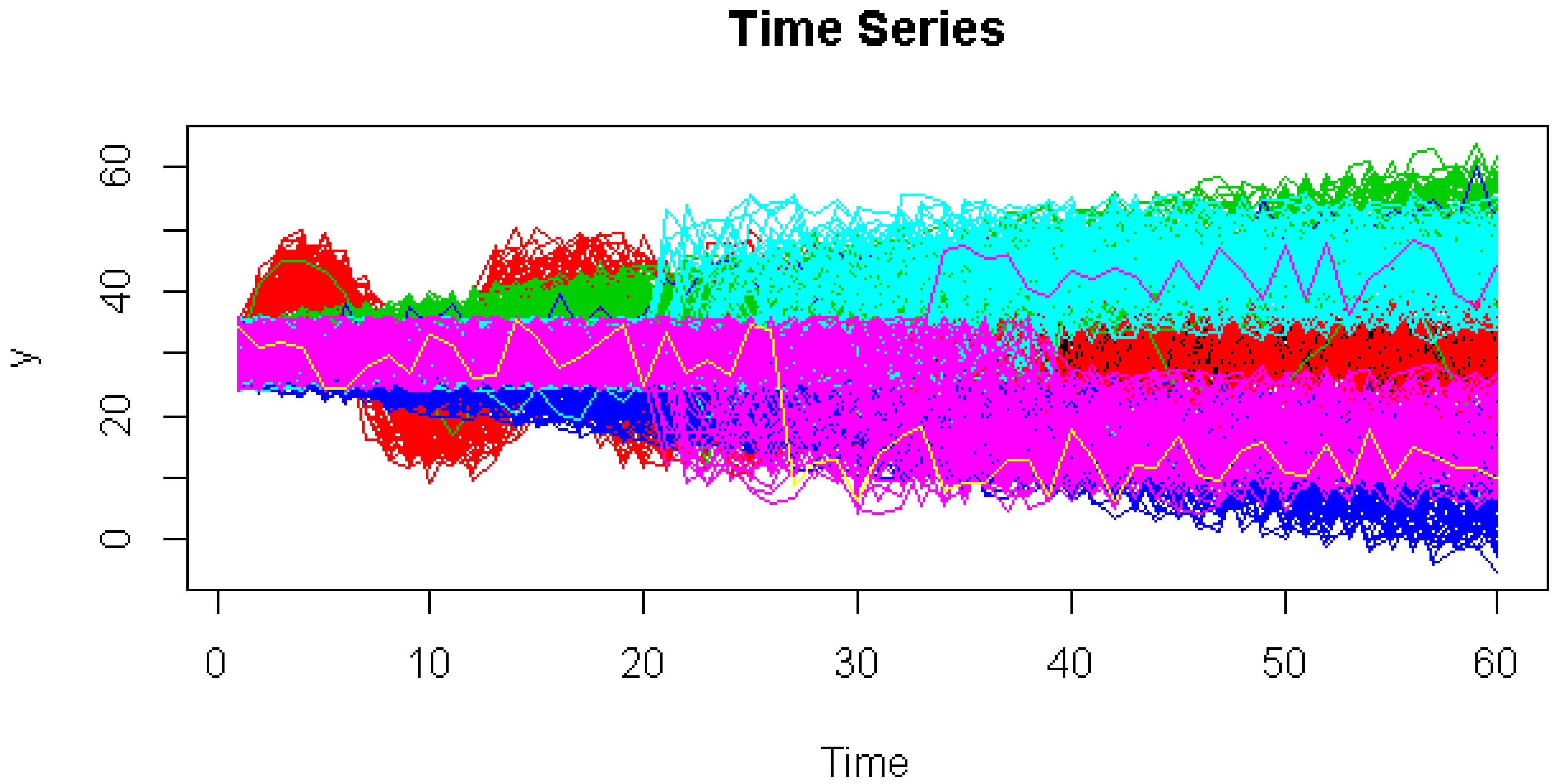
http://www.rdatamining.com/

Deconstructing Trend Analysis 2

- Constructed dataset
- 6 x 100 sub-series



Deconstructing Trend Analysis 3



Deconstructing Trend Analysis 4

Predicted Group

Actual Group	1	2	3	4	5	6
1	97	3	0	0	0	0
2	1	99	0	0	0	0
3	0	0	81	0	19	0
4	0	0	0	63	0	37
5	0	0	16	0	84	0
6	0	0	0	1	0	99

- Accuracy 87%!

Deconstructing Trend Analysis 5

- Accuracy 87%!!!
- Where could this give more insight?
  - Claim rates
  - Seasonal / Selection Effects
  - Investment performance analysis
  - Stochastic model analysis
  - Trend analysis

Unsupervised Learning Summary

- Can help identify patterns in data
- Can help identify homogeneous groups
- Using computer power
- Relatively unsophisticated
- Possible to get answers quickly
- Perfect insight not possible
- Improved understanding may result

Any Questions?

- What is Data Science?
- Why has it Grown So Quickly?
- Opportunities and Threats
- Open Source vs Closed Source
- Buzzwords
- Example: Machine Learning Model
- Practical Examples