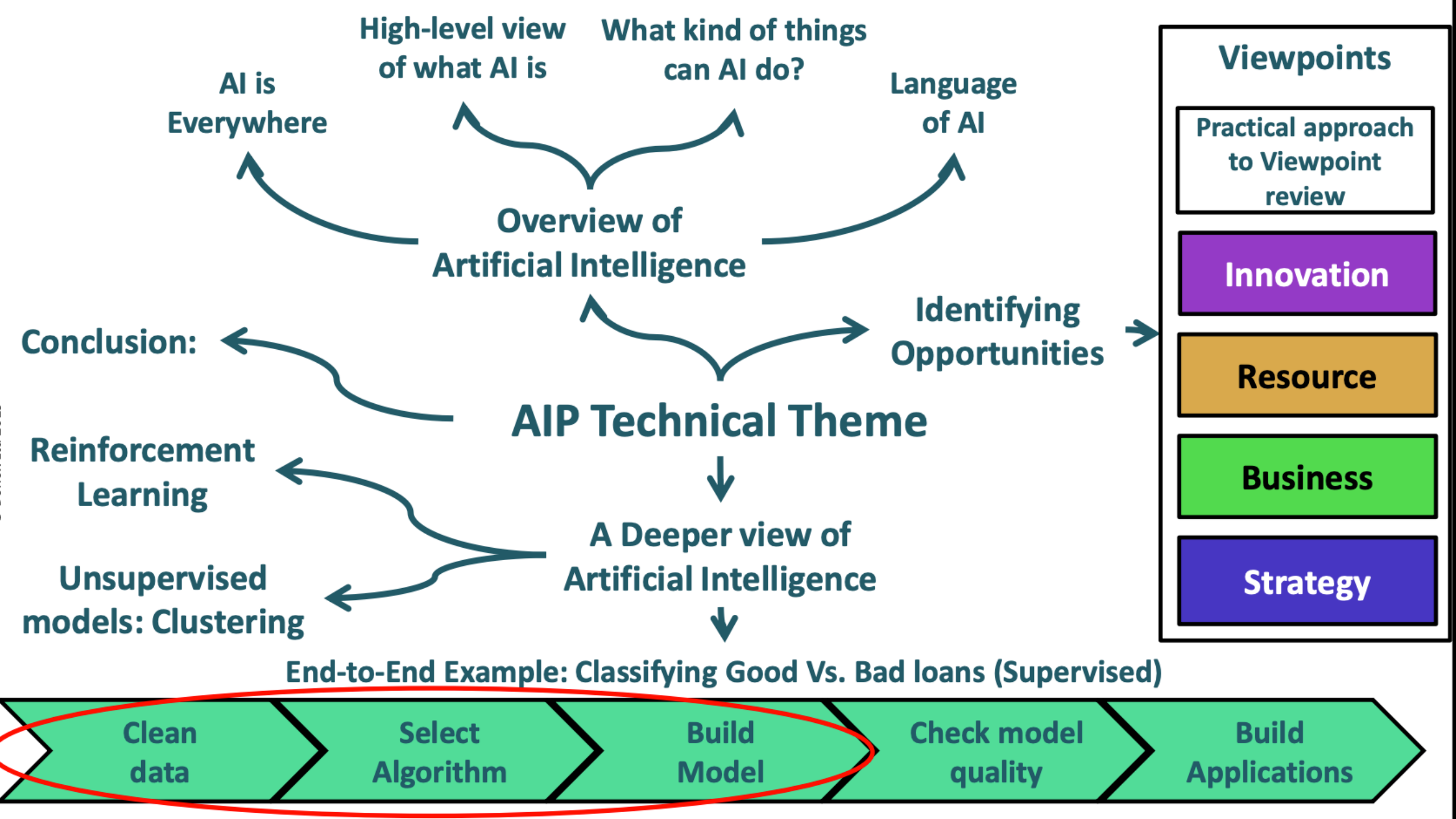
immerse
EDUCATION

Supervised Learning

Elizabeth Savochkina | 6th June

Shape of a Day

Registration 9.00am to 9.30am
Supervised Learning 9.30am to 11.30am
Break 11.30am to 13.30 pm
Hands on: RapidMiner, Python and Jupyter 13.30am to 15.30pm


High-level view of what AI is
What kind of things can AI do?
AI is Everywhere
Language of AI
Overview of Artificial Intelligence
Conclusion:
Identifying Opportunities
AIP Technical Theme
Reinforcement Learning
A Deeper view of Artificial Intelligence
Unsupervised models: Clustering
Viewpoints
Practical approach to Viewpoint review
Innovation
Resource
Business
Strategy
End-to-End Example: Classifying Good Vs. Bad loans (Supervised)
Clean data
Select Algorithm
Build Model
Check model quality
Build Applications

End-to-End Example: Classifying Good Vs. Bad loans

Credit Scoring

Why?

It's a high-value problem .. Lending money to people who are likely to default is not fair on lender or borrower

Opportunity?

Significant amount of historic data

Data covers many features

Historic data is labelled

Method?

Build a model that categorises new borrowers based on the probability they will default on their loan

Data Source: The German Credit Data

Creditability	Account Balance	Duration Of Credit	Payment Status of Previous	Purpose	Credit Amount	Value Savings Stocks	Length of current employment	Instalment per cent	Sex and Marital Status	Guarantors	Duration in Current address	Most valuable available	Age
1	1	18	4	2	1049	1	2	4	2	1	4	2	21
1		9	4	0	2799	1	3	2	3	1	2	1	36
1		12	2	9	841	2	4	2	2	1	4	1	23
1					2122	1	3	3	3	1			39
1					2171	1	3	4	3	1			38
1					2241	1	2	1	3	1			
1					3398	1	4	1	3	1			39
1					1361	1	2	2	3	1			40
1					1098					1	4	3	65
1					3758					1	4	4	23
1	1	11	4	0	3905					1	2	1	36
1	1	30	4	1	6187					1	4	3	24
1	1	6	4	3	1957					1	4	3	31
1	2	48	3	10	7582					1	4	4	31
1	1	18	2	3	1936					1	4	3	23
1	1	6	2	3	2647	3	3	2	3	1	3	1	44
1	1	11	4	0	3939	1		1	3	1	2	1	40
1	2	18	2	3	3213	3	2	1	4	1	3	1	25
1	2	36	4	3	2337	1	5	4	3	1	4	1	36
1	4	11	4	0	7228	1	3	1	3	1	4	2	39

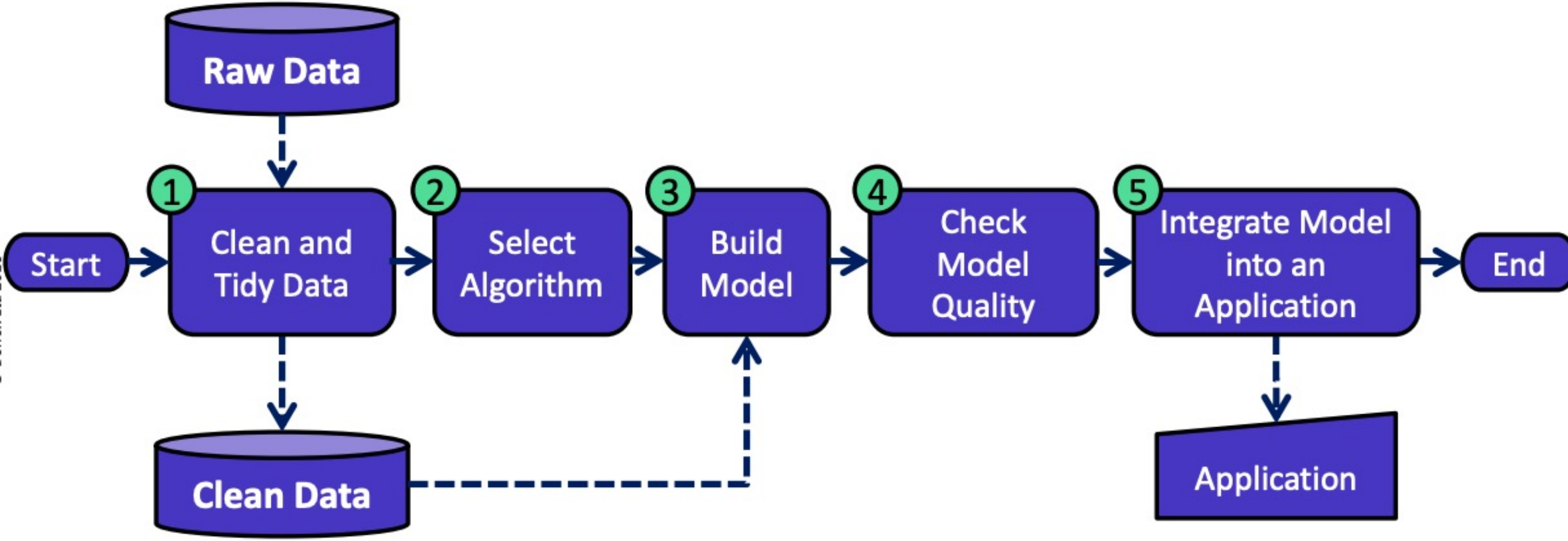
Label in the first column
1 : Good loan
0: Default

20 Features

1000 Observations
700 good loans
300 default

https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

Typical ML Workflow



Step 1 : Data Cleansing

Step 1 : Data Cleansing

1 Clean and Tidy Data

- Most data will be incomplete and have errors of various types
- We could just delete rows with missing data or errors ..
- But this is very wasteful in data .. You tend to throw away a lot of useful information
- Deleting data can also give biased results

ID	Name	Age	Credit Score	Loan Amount	Marriage Status
1	Rob Collins	28	920	27,228	Single
2	Dave Smith	43	634	45,673	Married
3	Sue James	32	0	41,813	Divorced
4	Ankit Gupta		720	23,645	Single
5	Kurt Godel	51	890	33,437	Single
6	Isaac Newton	420	786	39,739	Bereaved

Data Cleansing : Many Strategies ..

1 Clean and Tidy Data

Replace missing value with 'worst case' value

Replace missing value average of other legal values

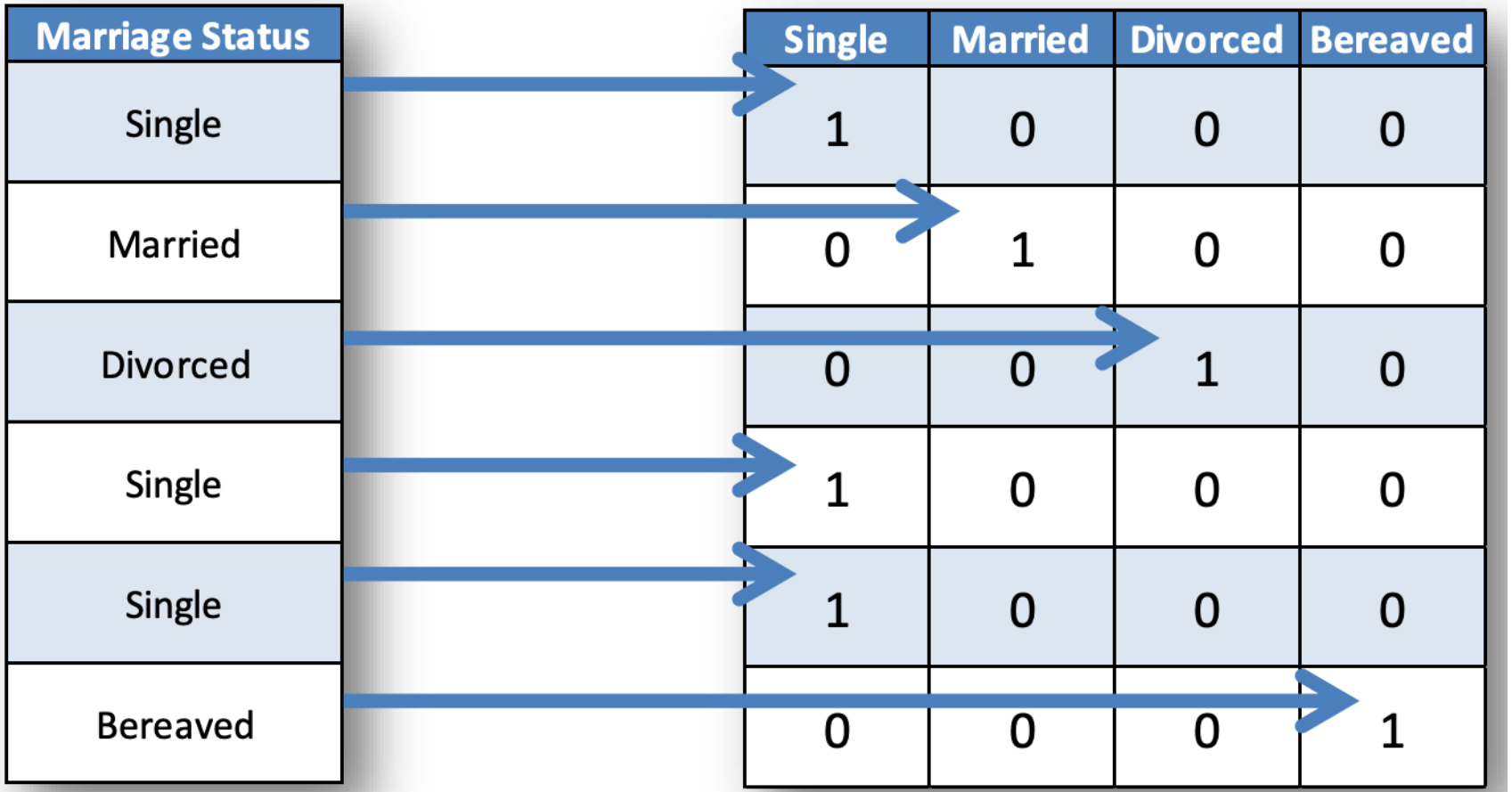
Alert analyst with a warning message to enable source data to be checked

ID	Name	Age	Credit Score	Loan Amount	Marriage Status
1	Rob Collins	28	920	27,228	Single
2	Dave Smith	43	634	45,673	Married
3	Sue James	32	500	41,813	Divorced
4	Ankit Gupta	38.5	720	23,645	Single
5	Kurt Godel	51	890	33,437	Single
6	Isaac Newton	420	786	39,739	Bereaved

Turning Classifications into Numbers

1 Clean and Tidy Data

- Many of the ML models depend on having numerical data rather than classifications such as “Single”
- Classifiers can be translated into ‘1-hot encoding’



Marriage Status
Single
Married
Divorced
Single
Single
Bereaved

Single	Married	Divorced	Bereaved
1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	0
1	0	0	0
0	0	0	1

*Binary Classification

Data Cleaning: Practical Example

German Credit Data : Data Source

1 Clean and Tidy Data

- The data in this section is publically available at the UCI Machine Learning Repository

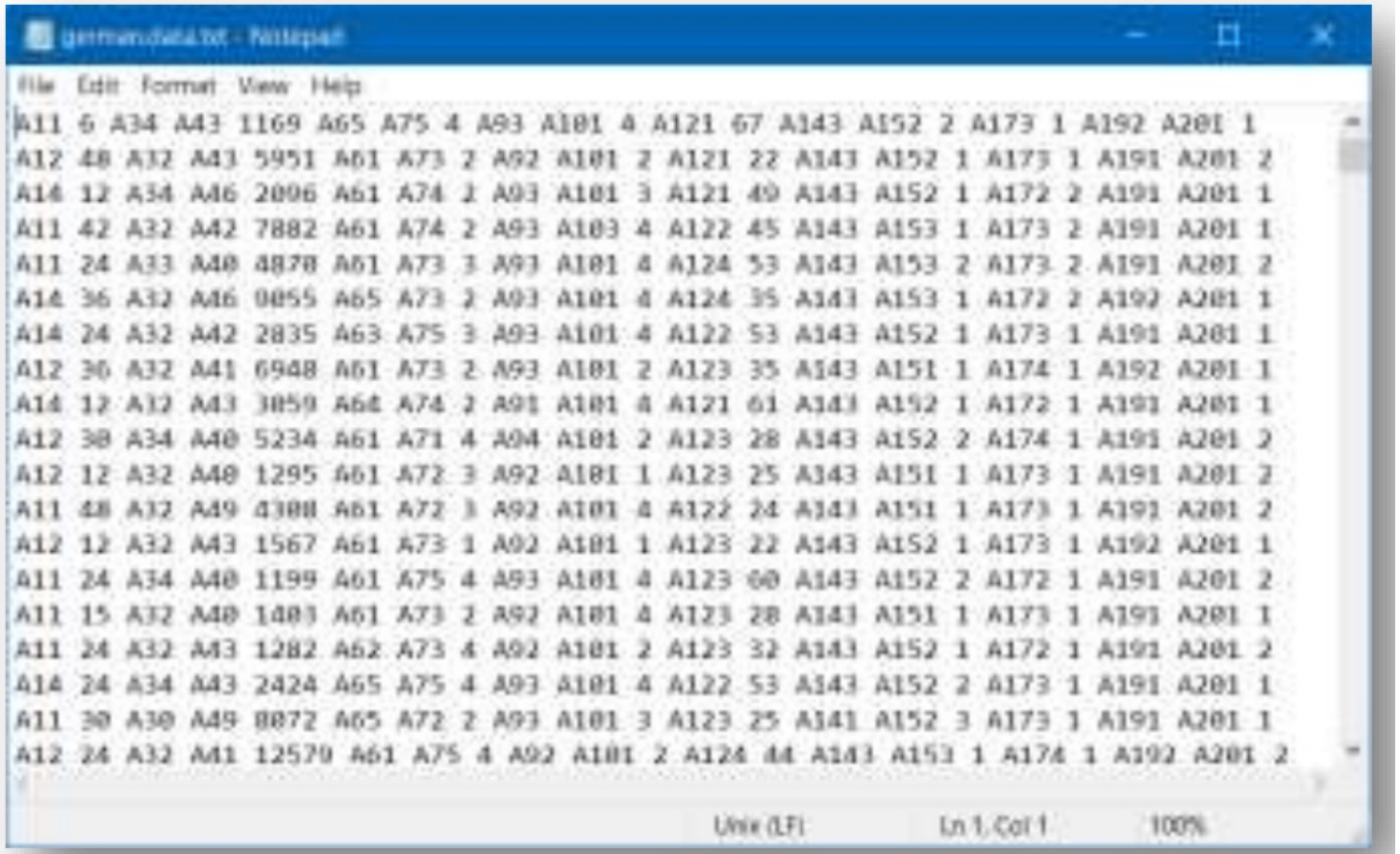
https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

https://bit.ly/1K3bcku

Raw Data

1 Clean and Tidy Data

- The data comes as a text file, with coded values
- This needs some tidying before it can be used to build models



Data Description

1 Clean and Tidy Data

- The data also comes with a descriptive file to help you interpret each column

Attribute description for German credit data

Attribute 1: (qualitative)
Status of existing checking account
A11 : ... < 0 DM
A12 : 0 <= ... < 200 DM
A13 : ... >= 200 DM /
salary assignments for at least 1 year
A14 : no checking account

Attribute 2: (numerical)
Duration in month

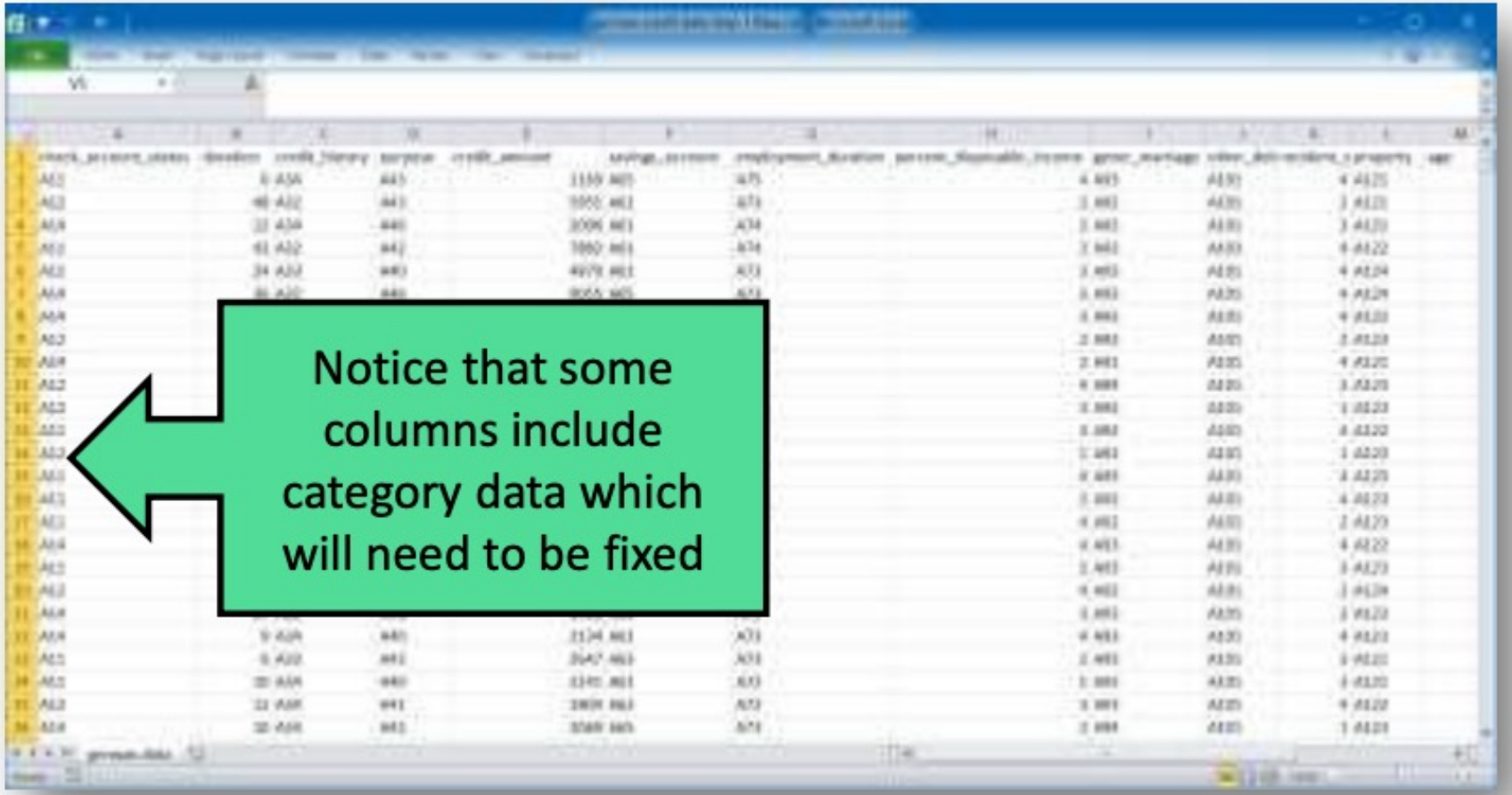
Attribute 3: (qualitative)
credit history
A30 : no credits taken/
all credits paid back duly
A31 : all credits at this bank paid back duly
A32 : existing credits paid back duly till now
A33 : delay in paying off in the past
A34 : critical account/
other credits existing (not at this bank)

Attribute 4: (qualitative)
Purpose
A40 : car (new)
A41 : car (used)
A42 : furniture/equipment
A43 : radio/television
A44 : domestic appliances
A45 : repairs
A46 : education
A47 : vacation - does not exist?
A48 : retraining
A49 : business
A410 : others

Column (Feature) Names

1 Clean and Tidy Data

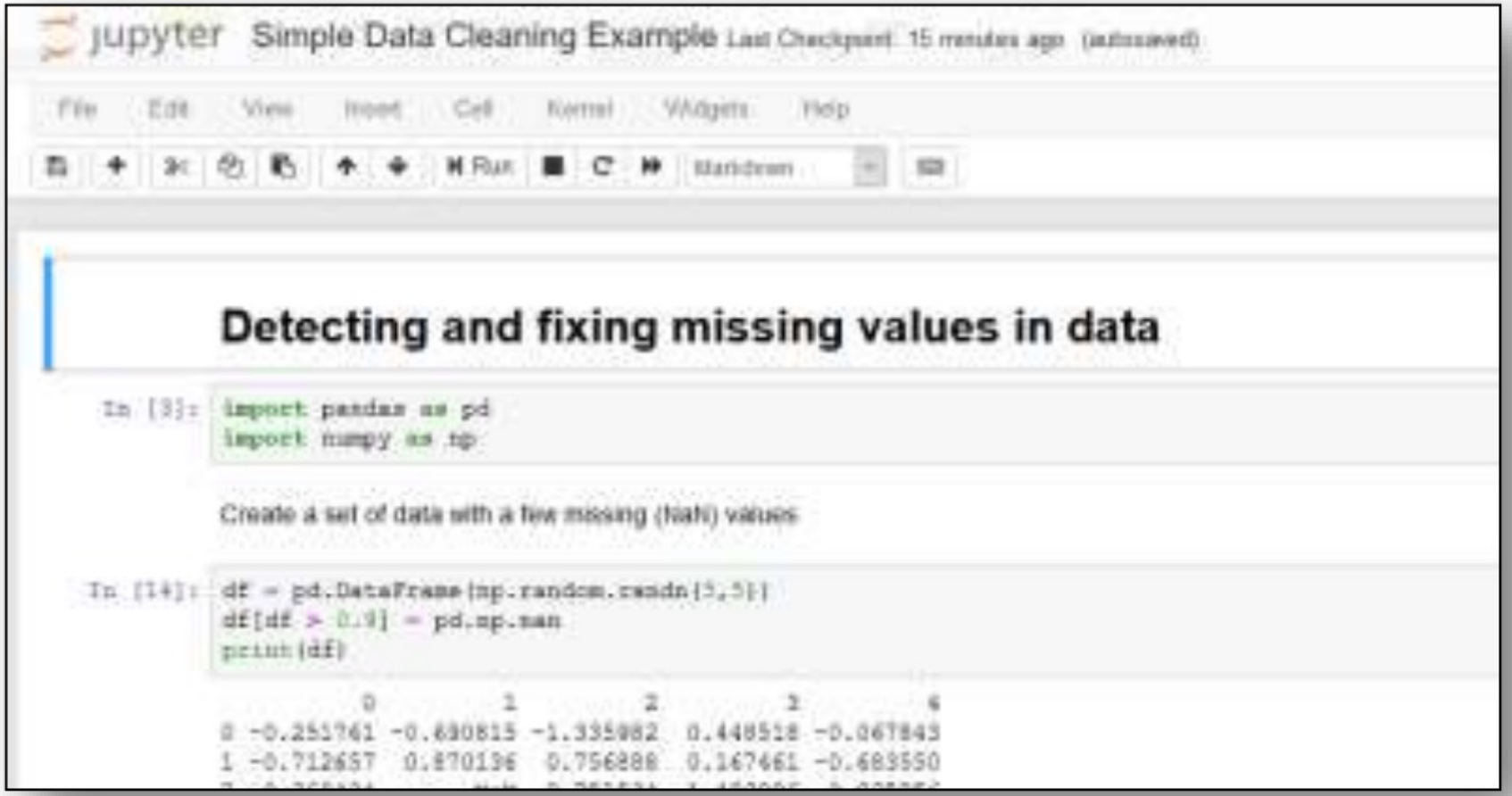
- I first used Excel to add a row at the top of the file containing names for each column (feature)
- Then I saved as a comma-separated variable file



Data Cleansing in Python

1 Clean and Tidy Data

- Python and the various libraries include many functions that support data cleansing,
- For example, detecting and repairing
  - Missing values
  - Zeros
  - Out-of-range values

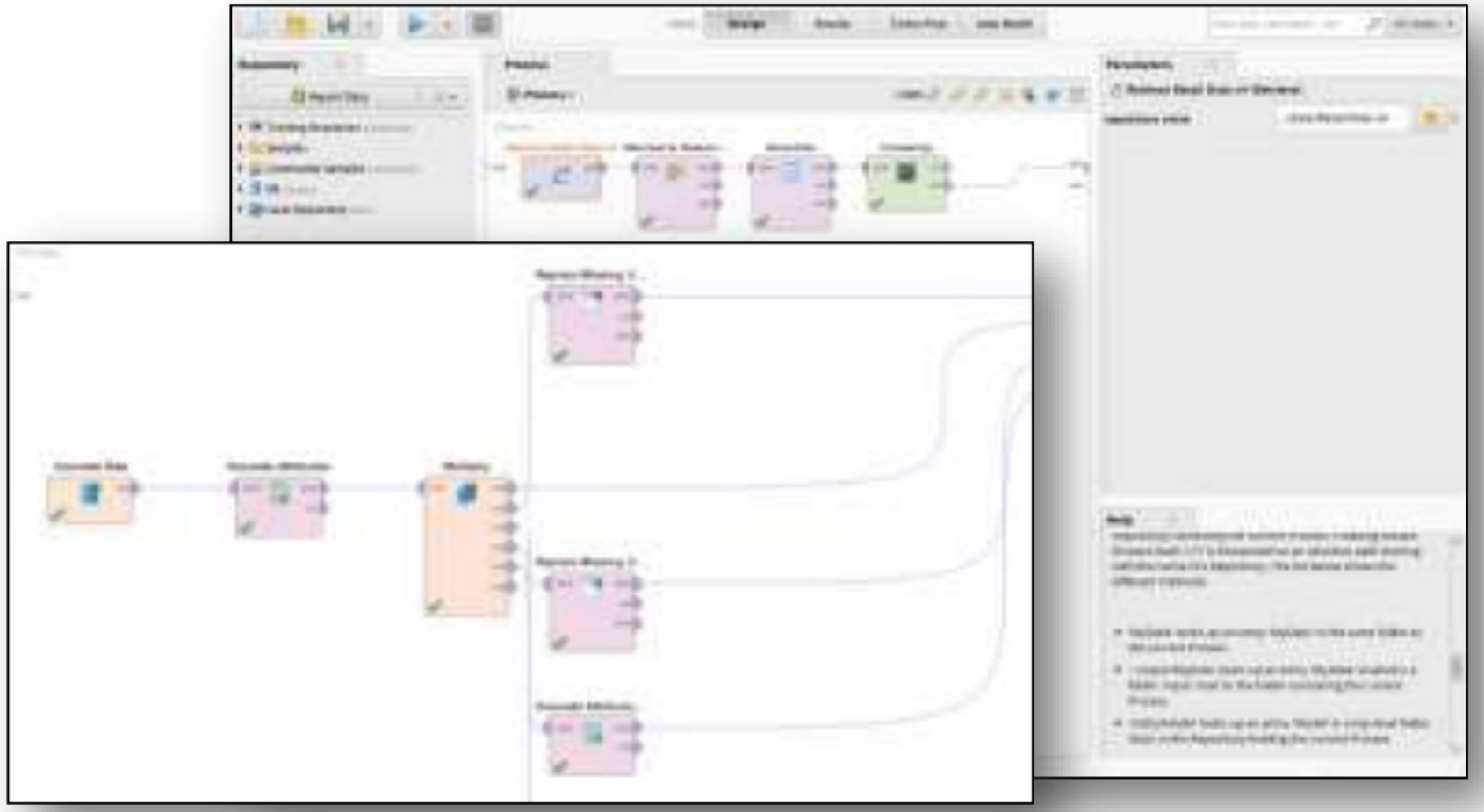


Demonstration: 03-01 – Loading and Cleaning Data”

Fix Missing Data in RapidMiner

1 Clean and Tidy Data

- RapidMiner also provides these functions ..
- ..but presented as graphic blocks rather than textual computer code

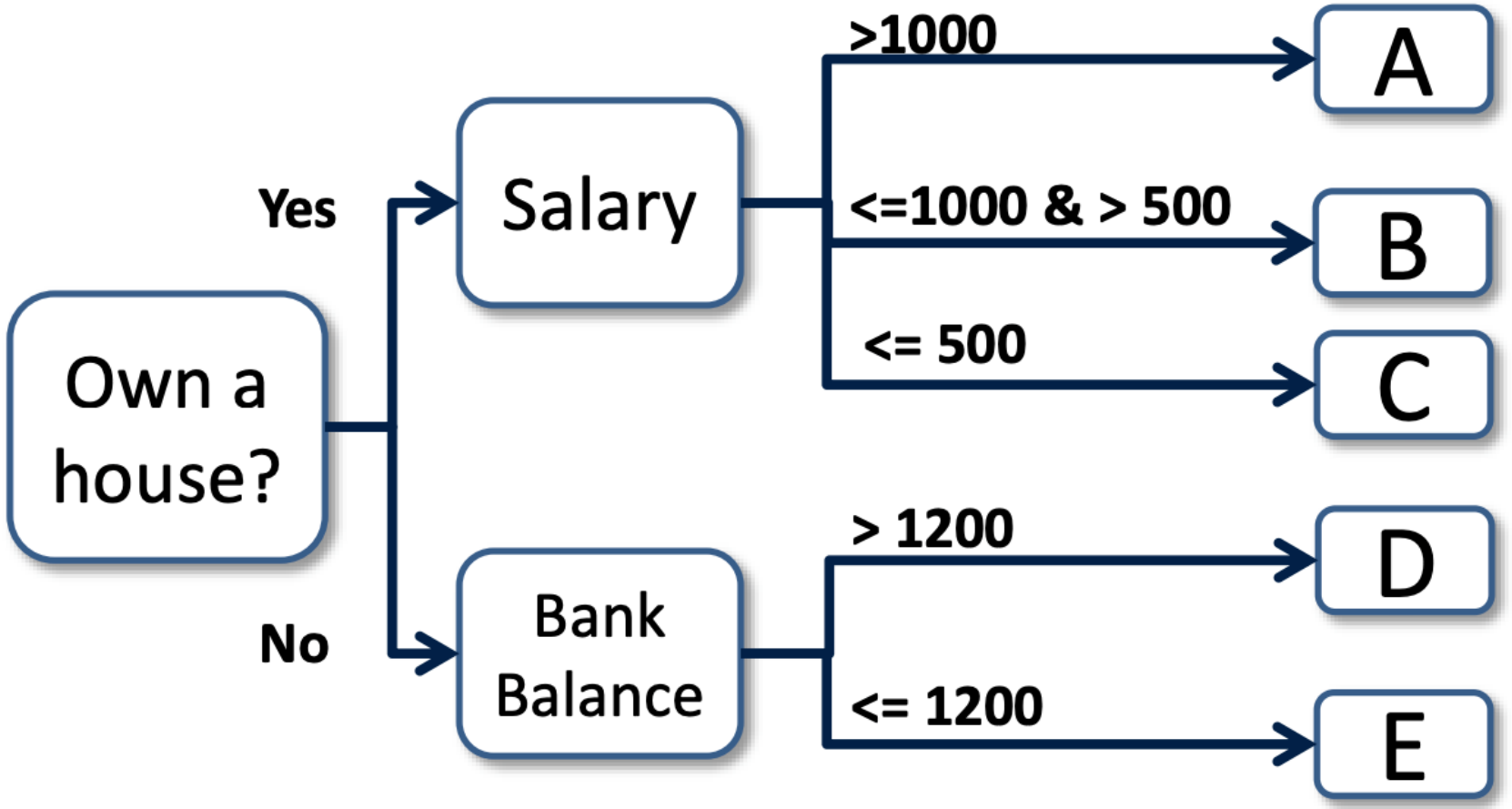


Step 2 : Selecting an Algorithm

Classification using Decision Trees

2 Select Algorithm

- The Decision Tree algorithm analyses data and builds rules based on previous examples
- Each decision step is selected by the algorithm to generate the most 'pure' set at the next level down

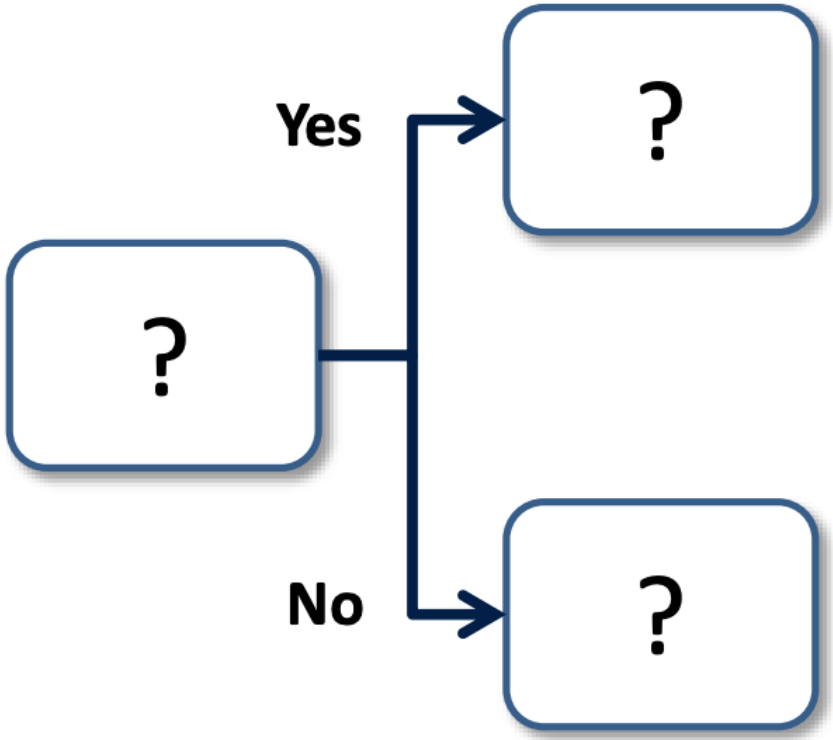


Decision Trees - Good and Bad

2 Select Algorithm

Good

- Simple to use
- Easy to explain the resulting logic (transparent)
- Works with ‘category’ data as well as numbers



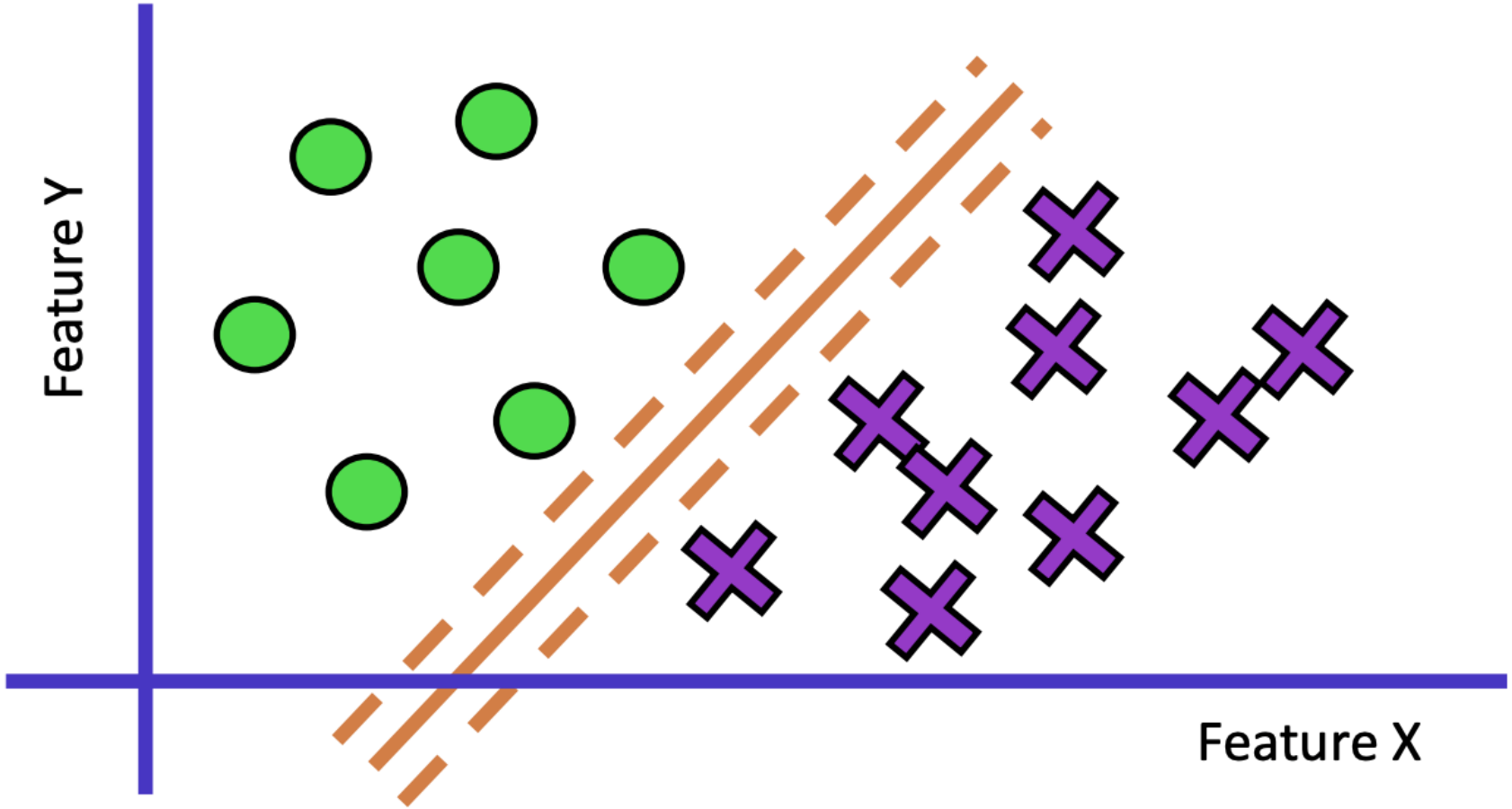
Bad

- Trees can grow very deep
- Tendency to ‘over-fit’ the data
  - I.e. result is too restricted and does not work well with new data

Support Vector Machine (SVM)

2 Select Algorithm

- Wiggle a line around until it neatly divides two classes with the biggest possible gap
- If there is more than two dimensions this will need to be a ‘plane’



Logistic Regression

2 Select Algorithm

- Create an ‘S’ shaped curve that fits known data points
- Curve runs from 0% to 100% and represents the probability of the item being in a particular classification

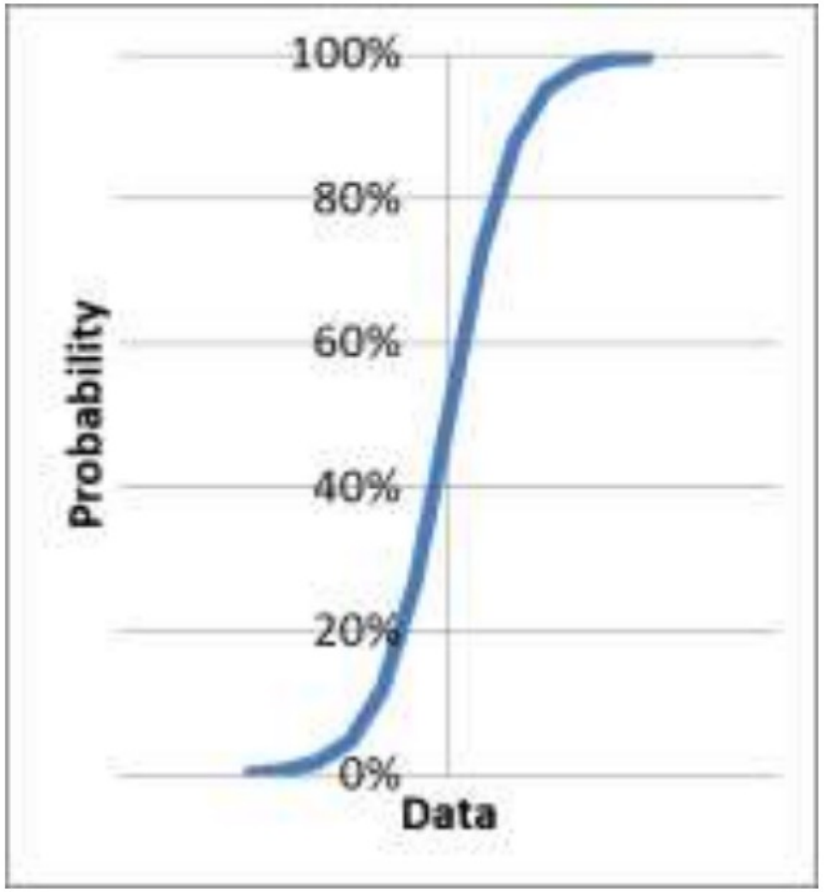
- Estimates the probability of an event occurring (A or not A), based on a given dataset of independent variables

Logistic Regression – Good and Bad

2 Select Algorithm

Good

- Gives a probability of the item being in a classification
- Better than simple ‘Yes’ / No result as we can set our own level of risk of getting the wrong answer

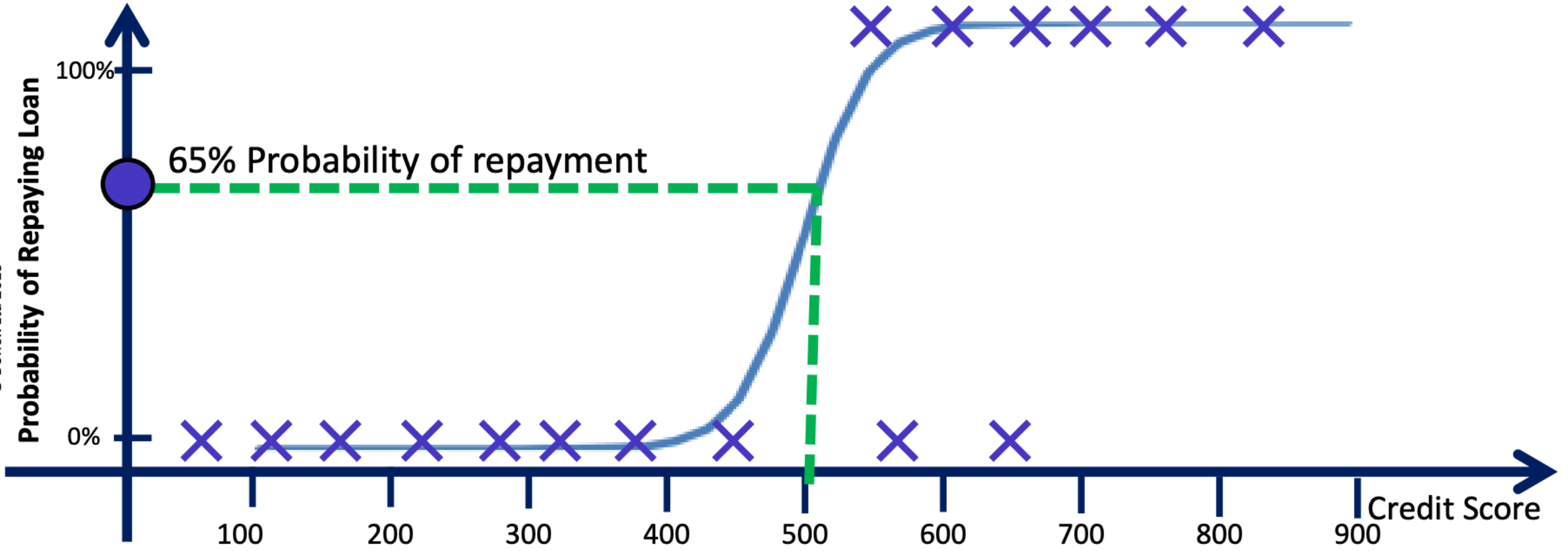


Bad

- Does not work well with lots (1000s) of different features (dimensions)
- Not so good with category data inputs

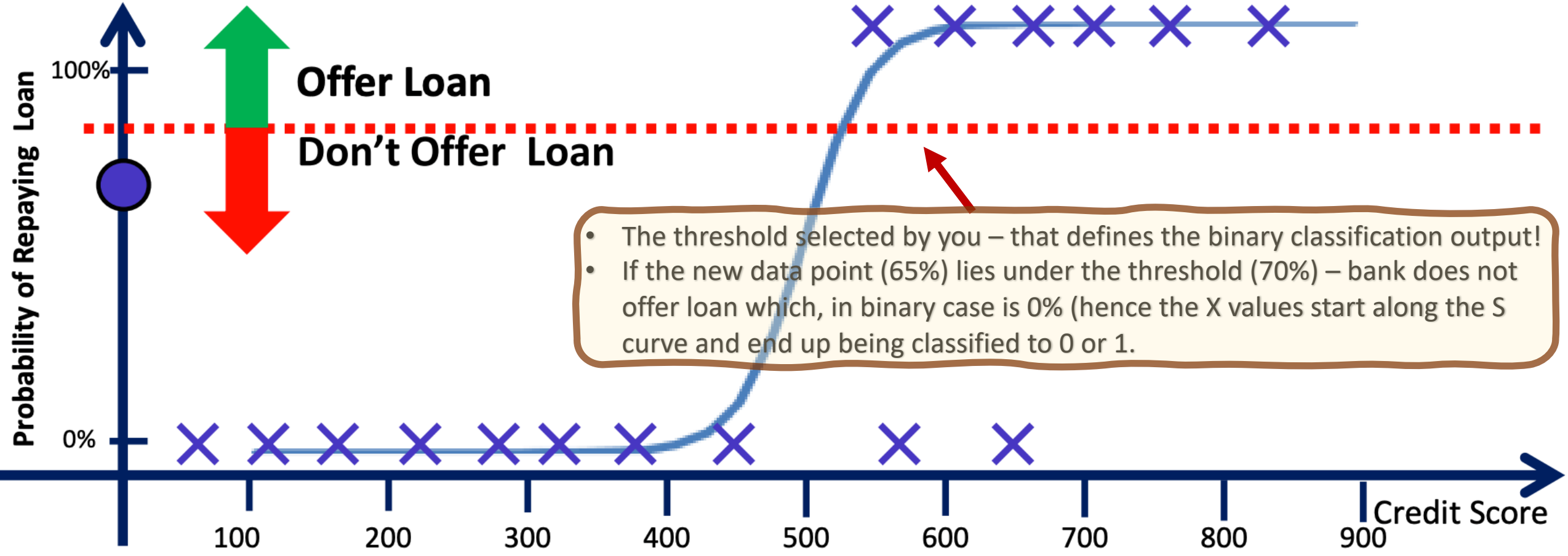
Logistic Regression to Predict Probability

2 Select Algorithm



Setting the Decision Threshold

2 Select Algorithm



- The threshold selected by you – that defines the binary classification output!
- If the new data point (65%) lies under the threshold (70%) – bank does not offer loan which, in binary case is 0% (hence the X values start along the S curve and end up being classified to 0 or 1.

Step 3 : Building a Model

Building the Model

3 Build Model

- Building models is made significantly easier as a result of a huge range of pre-built tools and libraries
- You can draw on the resources of companies such as Google and Facebook who have made their software libraries publically available for free

For Example .. Credit Analysis

3 Build Model

Get the data from a file into your program

```
german_credit_data = pd.read_csv("german_credit.csv")
```

Separate out the 'result' column (label) indicating if the person defaulted on their loan

```
X = german_credit_data.loc[:,'Account_Balance':'Foreign_Worker']
Y = german_credit_data['Creditability']
```

Create the tool that is used to build the model – in this case, from an existing library of solvers

```
logModel = LogisticRegression(solver='liblinear')
```

Build the model!

```
logModel = logModel.fit(X,Y)
```

It Can't be that Easy!

3 Build Model

- It isn't – it would be misleading to say it was
- But some powerful models are surprisingly easy, given the available tools
- Bespoke ML and more complex models are, of course, difficult!

PhD+ in Mathematics Data Science / ML / AI	**Creating new types of ML. Building libraries. Building ML for new and unusual applications.**
MSc in Data Science / ML / AI	**Building sophisticated models from more complex data sets. Using advanced tools including Neural Networks. Tuning and optimising models**
Experienced Programmer with some ML training	**Building and running 'real' models using existing ML tools and libraries. Building simpler ML applications**
Primary school level use of Python	**Building and running simple models based on many tutorials and worked examples on the Internet**

Model Building
Practical Example

Step 3 : Build the Model

3 Build Model

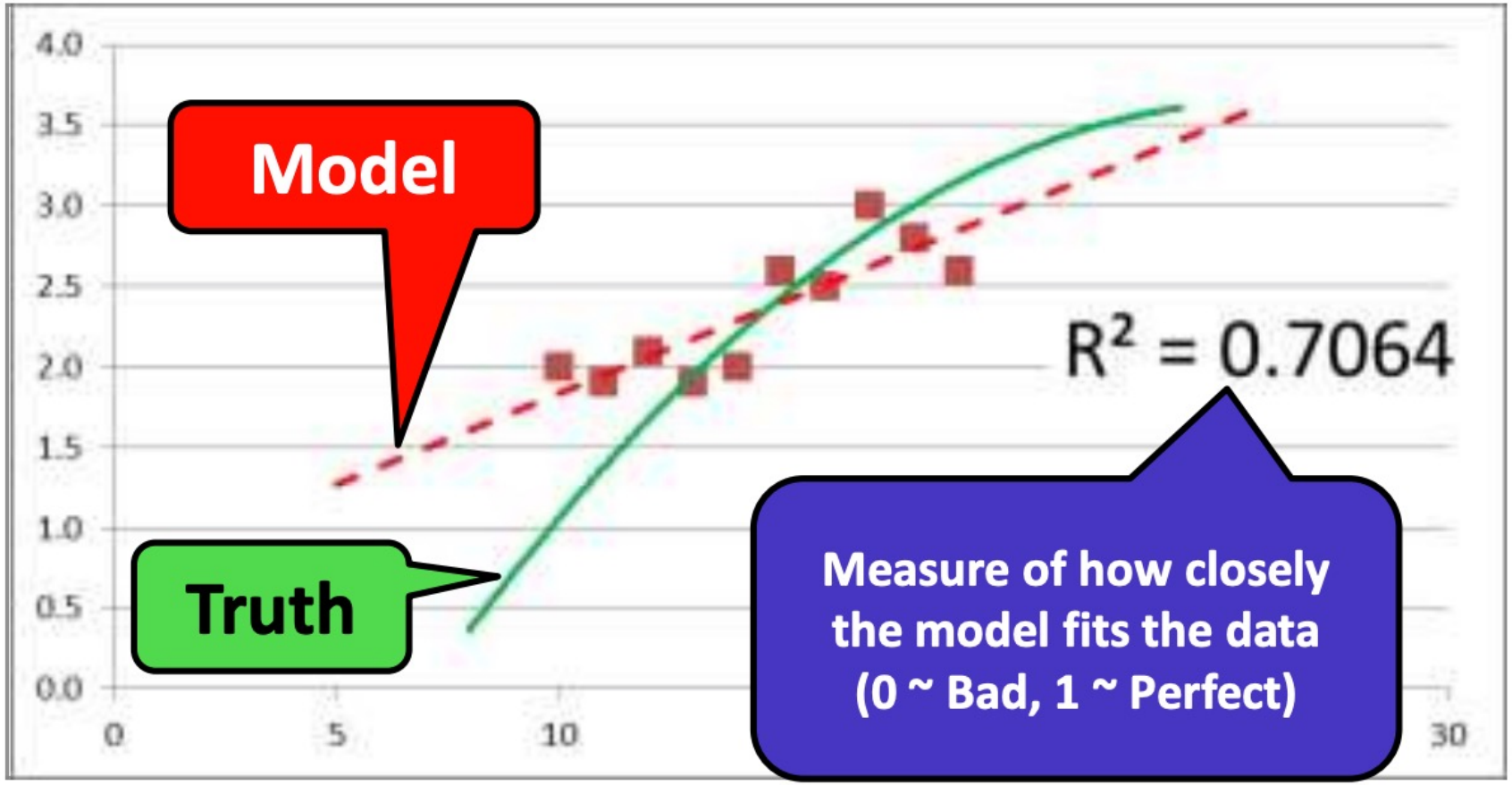
- Model building can be relatively straight-forward for many types of ML model due to the powerful libraries and tools that are publically available

Demonstration: "03-02 – Building the Model"

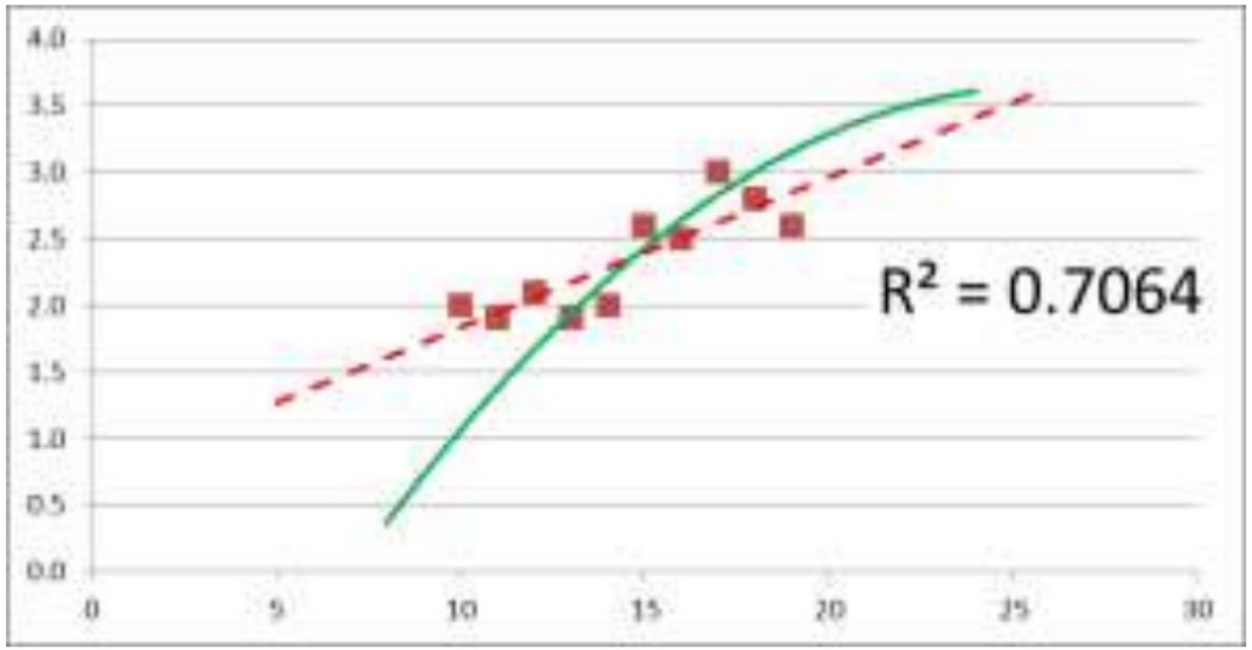
When Models go Bad

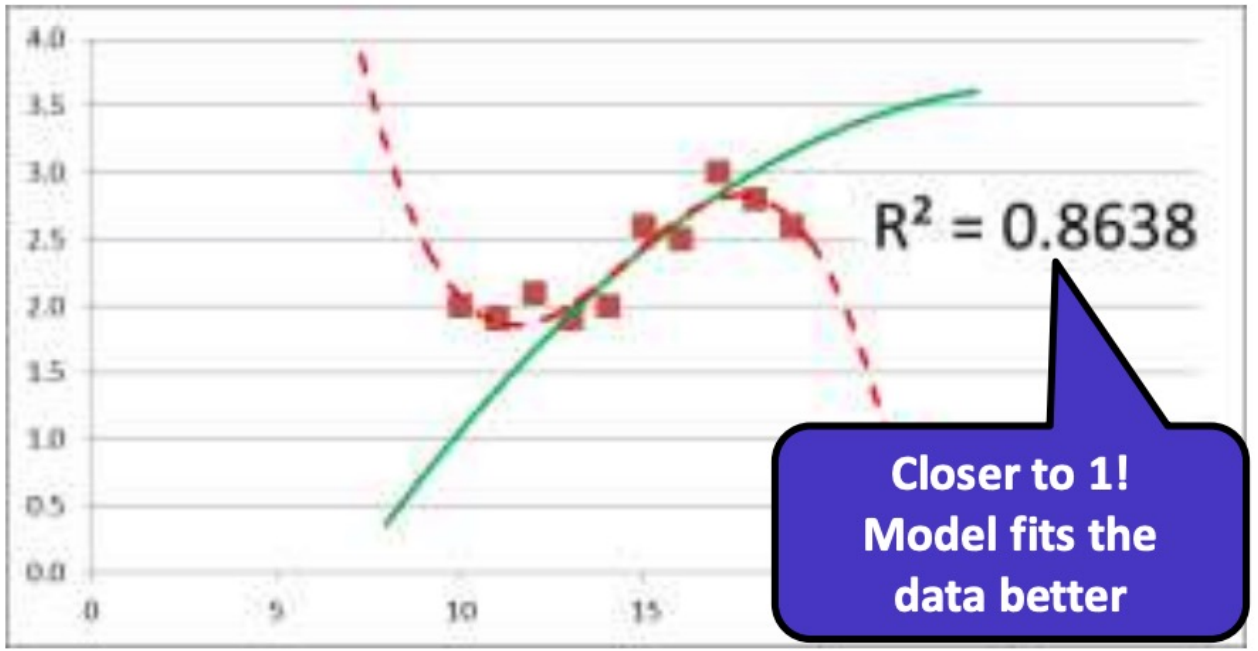
Fitting a Straight Line to the Data

	Y	
X	Truth	Measurement
10	1.0	2
11	1.4	1.9
12	1.7	2.1
13	2.0	1.9
14	2.2	2
15	2.4	2.6
16	2.6	2.5
17	2.8	3
18	3.0	2.8
19	3.2	2.6



Let's Try a More 'Sophisticated' Model



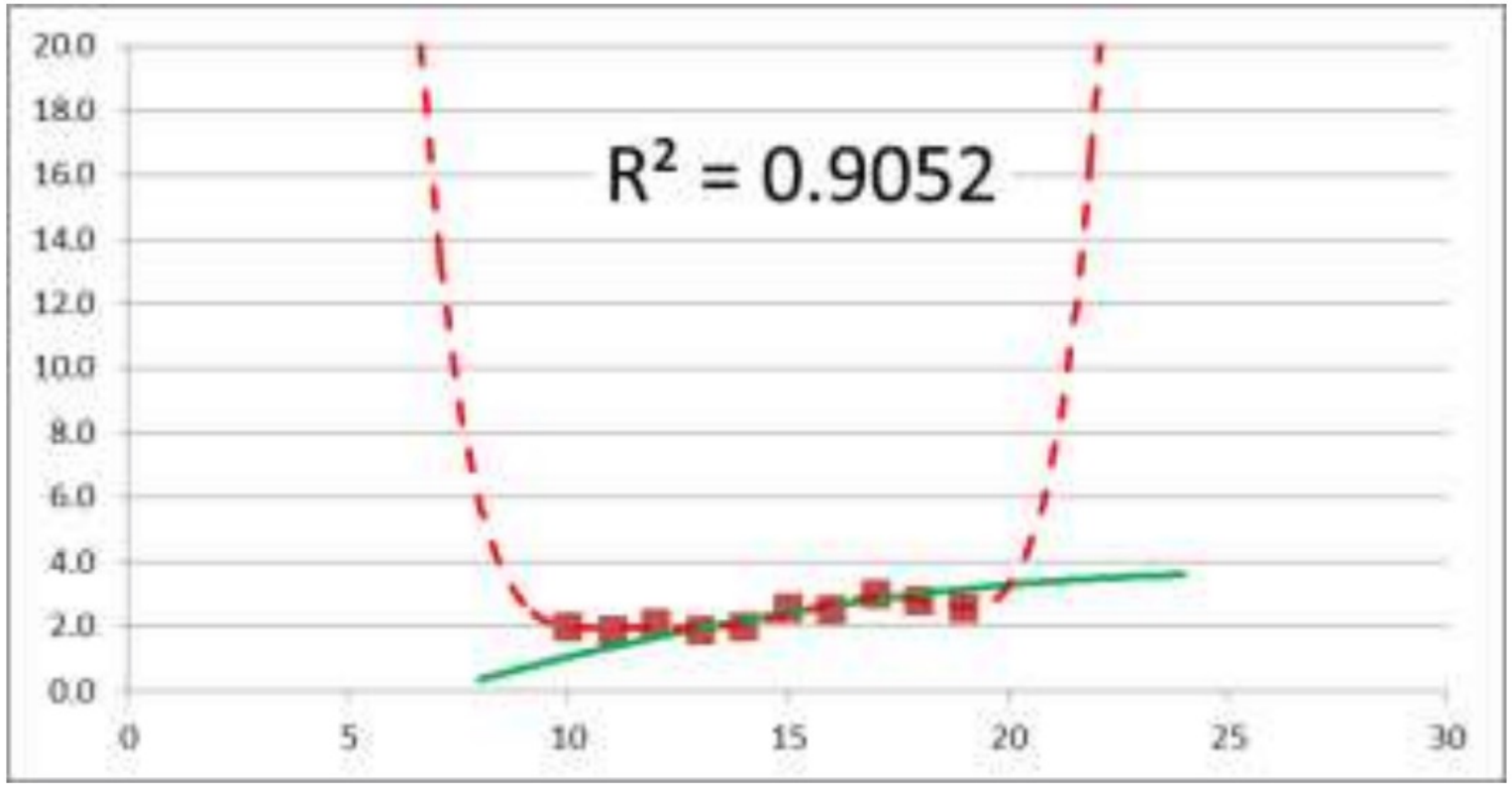


$$Y = A.x + K$$

$$Y = A.x + B.x^2 + C.x^3 + K$$

Let's Try a Really 'Sophisticated' Model!

- Our R2 value is now close to 1 …
  - The line very closely matches our data points
- But do we have a 'good' model?
- Not really, .. It does not seem to do a good job of predicting



$$Y = A.x + B.x^2 + C.x^3 + D.x^4 + E.x^5 + G.x^6 + K$$

Over-Fitting

- This phenomena is called ‘over-fitting’ – and is a reoccurring problem in data-driven models
- The model learns specific examples …
  - .. but fails to generalise to new data ..
  - .. and is therefore not useful for prediction
- We will learn how to detect this by evaluating model quality in the next session.