Get Into Data Science:

#2 Supervised Machine Learning Demystified

Led by Dr Glenn Amundsen, Senior Data Scientist at Royal Mail

Free | Thursday 6th December 6.30pm - 9pm | Farringdon

LOGIKK
Providing the future

Royal Mail

Computacenter

Get Into Data Science:
A Free Workshop Series

#1
~~Introduction to Interactive Data Visualisation~~

#2
Supervised Machine Learning Demystified

#3
Digital Image Processing

#4
Natural Language Processing for Humans

#5
Modelling the Future with Time Series

#6
Building Recommender Systems

Search for Royal Mail on Meetup

LOGIKK
Providing the future

Royal Mail

Computacenter

Data Science team @ Royal Mail

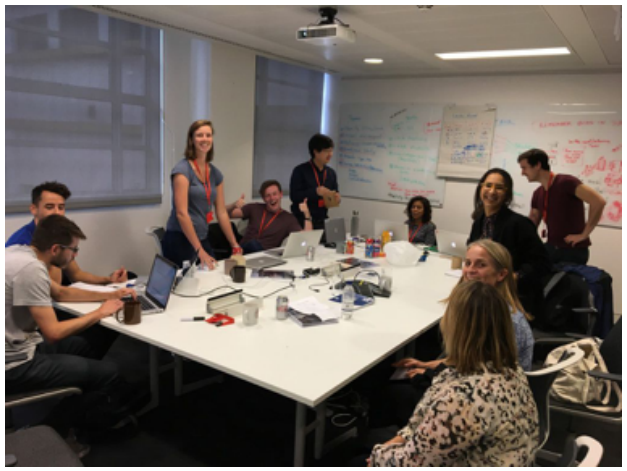
14 countries

8

16

PhD	13	54%
MSc	6	25%
BSc	3	13%
A/L	2	8%

Many backgrounds



Agenda

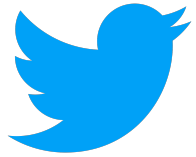
18:30 Welcome

19:00 Introduction to Supervised Machine Learning

19:30 Applying Supervised Machine Learning

20:15 Hands-on Exercise - (Optional)

20:30 Networking + Pizza



#RMdatasci

Supervised Learning Demystified

using random forests

Royal Mail Data Science Workshop 6th December 2018

Glenn Amundsen

Royal Mail

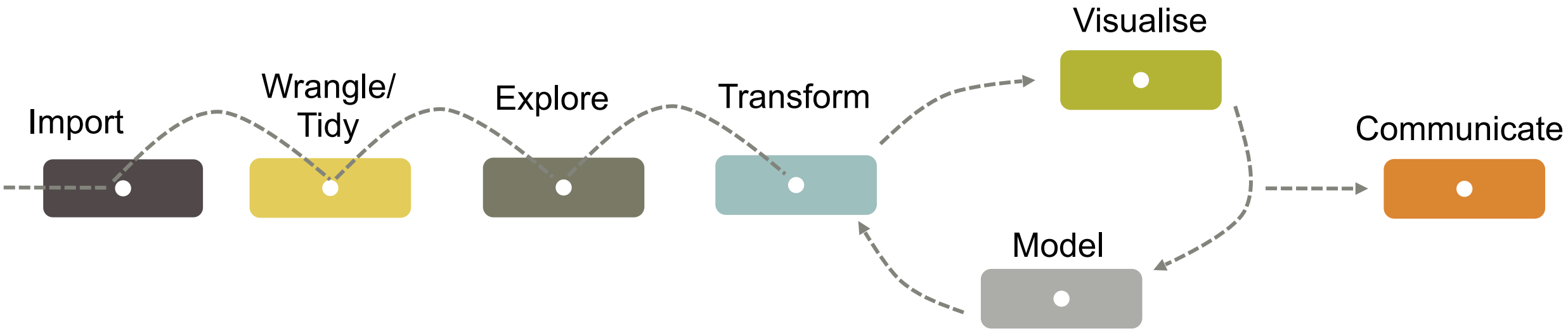
#RMdatasci

Goals for this workshop

1. Understand the basics of machine learning
2. Learn about Random Forests
3. Solve a ML classification problem

Introduction and General concepts

The Data Science process

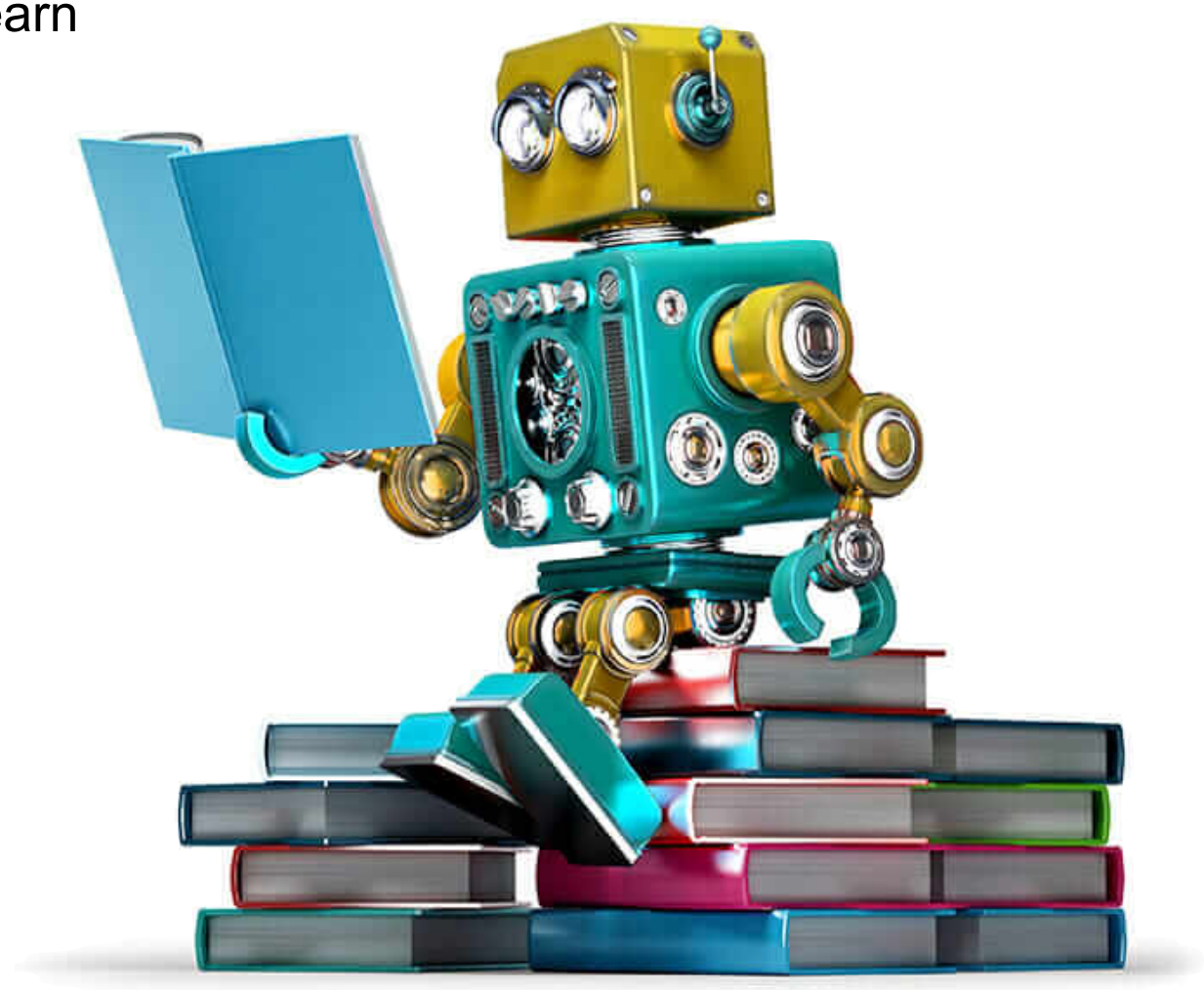


Building a Machine Learning Model

http://r4ds.had.co.nz/introduction.html

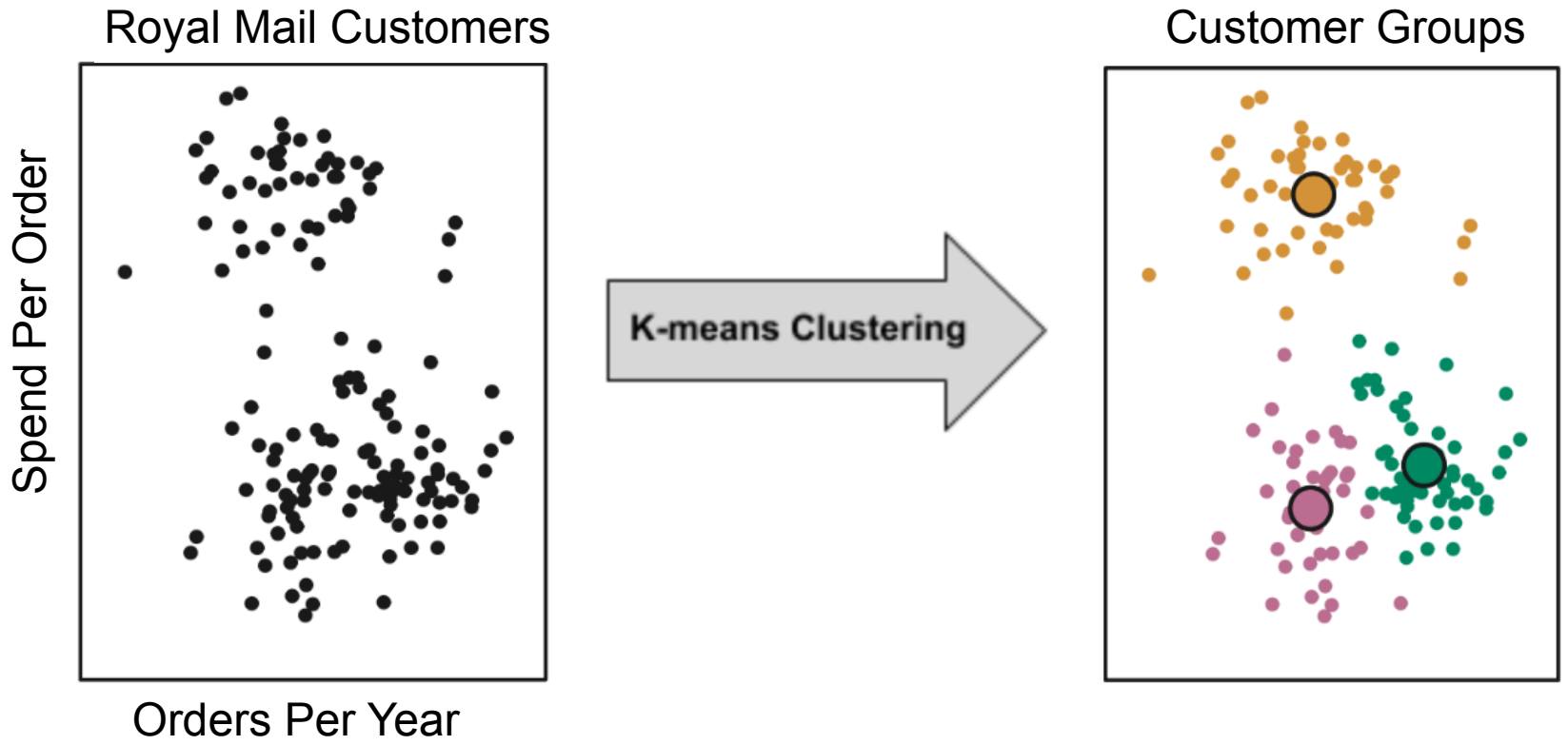
What is Machine Learning?

- A broad term that covers a class of algorithms that learn *something* from data
- Clustering, Classification, Regression
- From very simple (linear regression) to very complicated (neural networks)



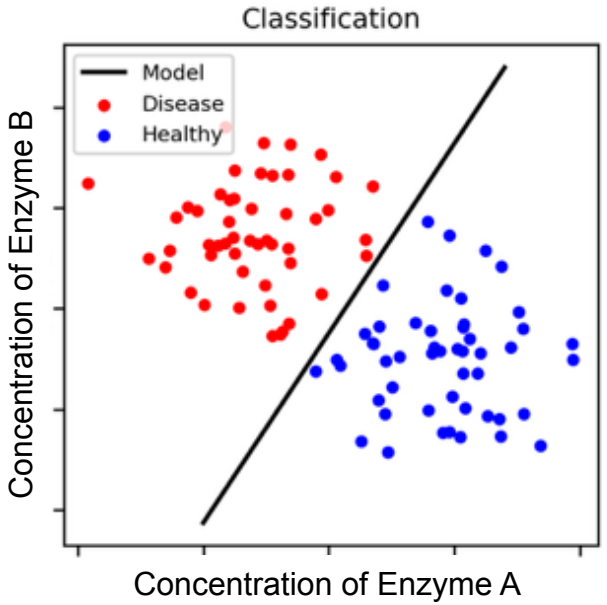
Unsupervised Learning

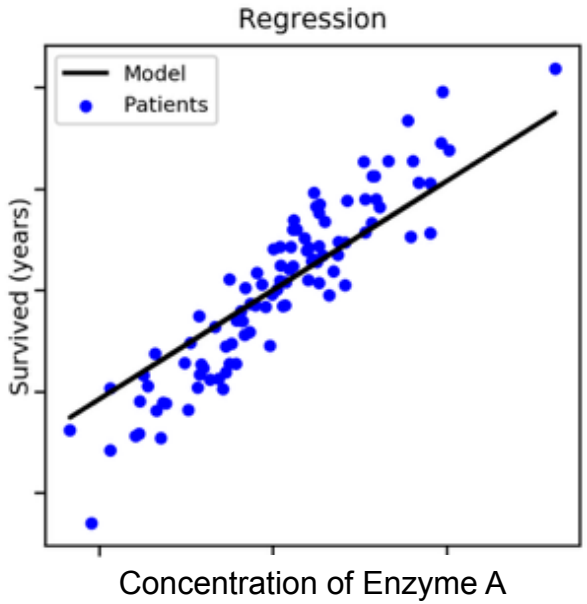
- No Labels, Just learn from the structure of the data
- Use cases: document clustering, customer segmentation, anomaly detection
- Example algorithms: K-means, hierarchical clustering, local outlier detection



Supervised Learning

- **Need to have some labeled data**
- **Learn to predict labels of new, unlabelled, data**
- **Classification: Predict categorical variables**
  - Use cases: predicting presence of disease, *direction* of stock price movement, sentiment of @RoyalMail tweets
  - Example algorithms: Logistic Regression, Random Forests, K-nearest neighbours
- **Regression: predict numerical variables**
  - Use cases: Predicting survival years of patient, *value* of stock price, time a package will arrive
  - Example algorithms: Linear Regression, Random Forests, Support Vector Machines





Applying Machine Learning

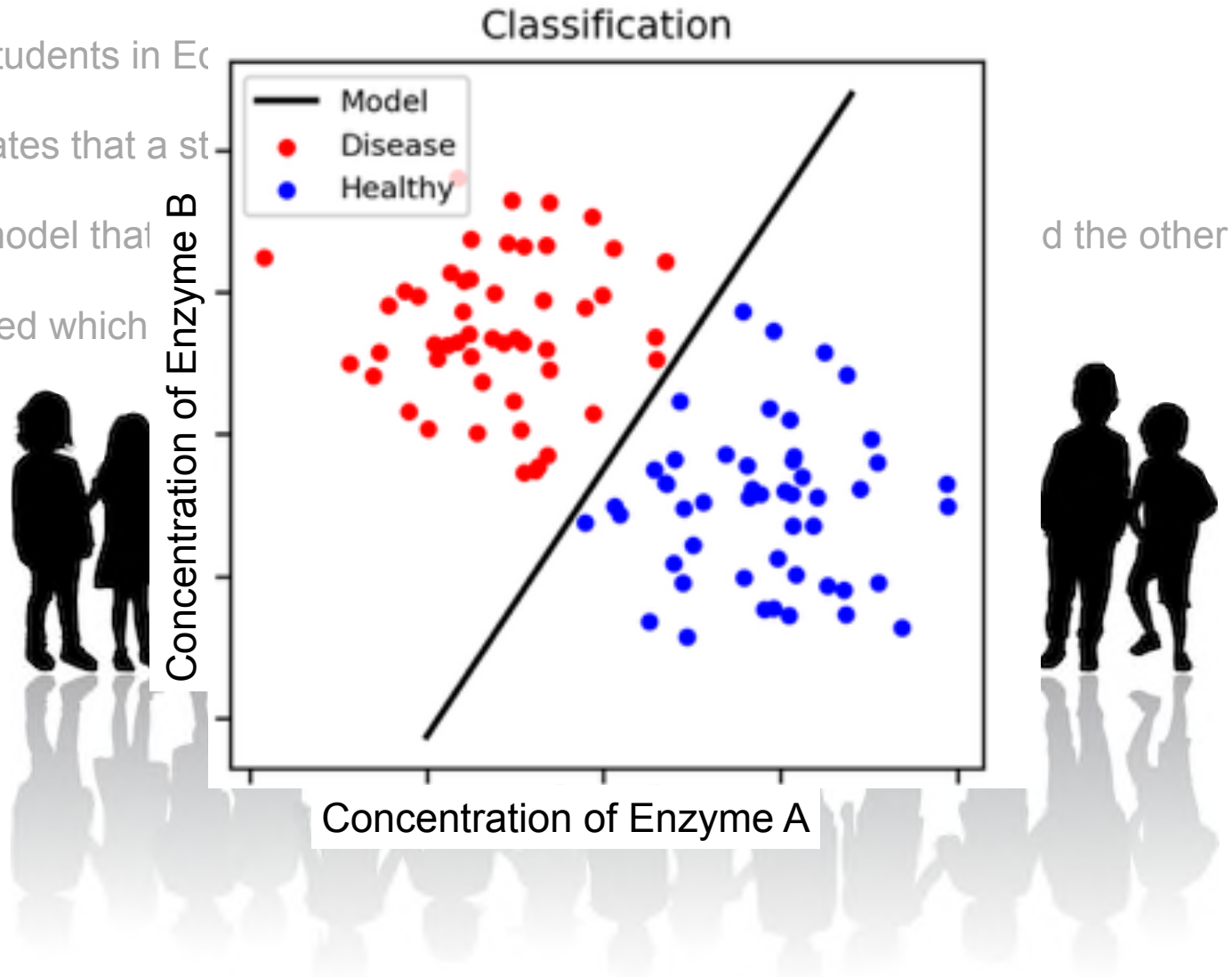
Repo available here: **https://github.com/royalmail-datascience/Meetup2_SupervisedLearning**

The Problem

- 180,000 people are reported missing in the UK every year (80,000 children)
- We want to be able to predict if a student is going missing
- The data: Survey of students in Edinburgh asking a broad range of questions
  - One question indicates that a student has run away (i.e. gone missing)
- Build a classification model that uses the running away question as target labels and the other questions as variables
- Charity is also interested which features are most important

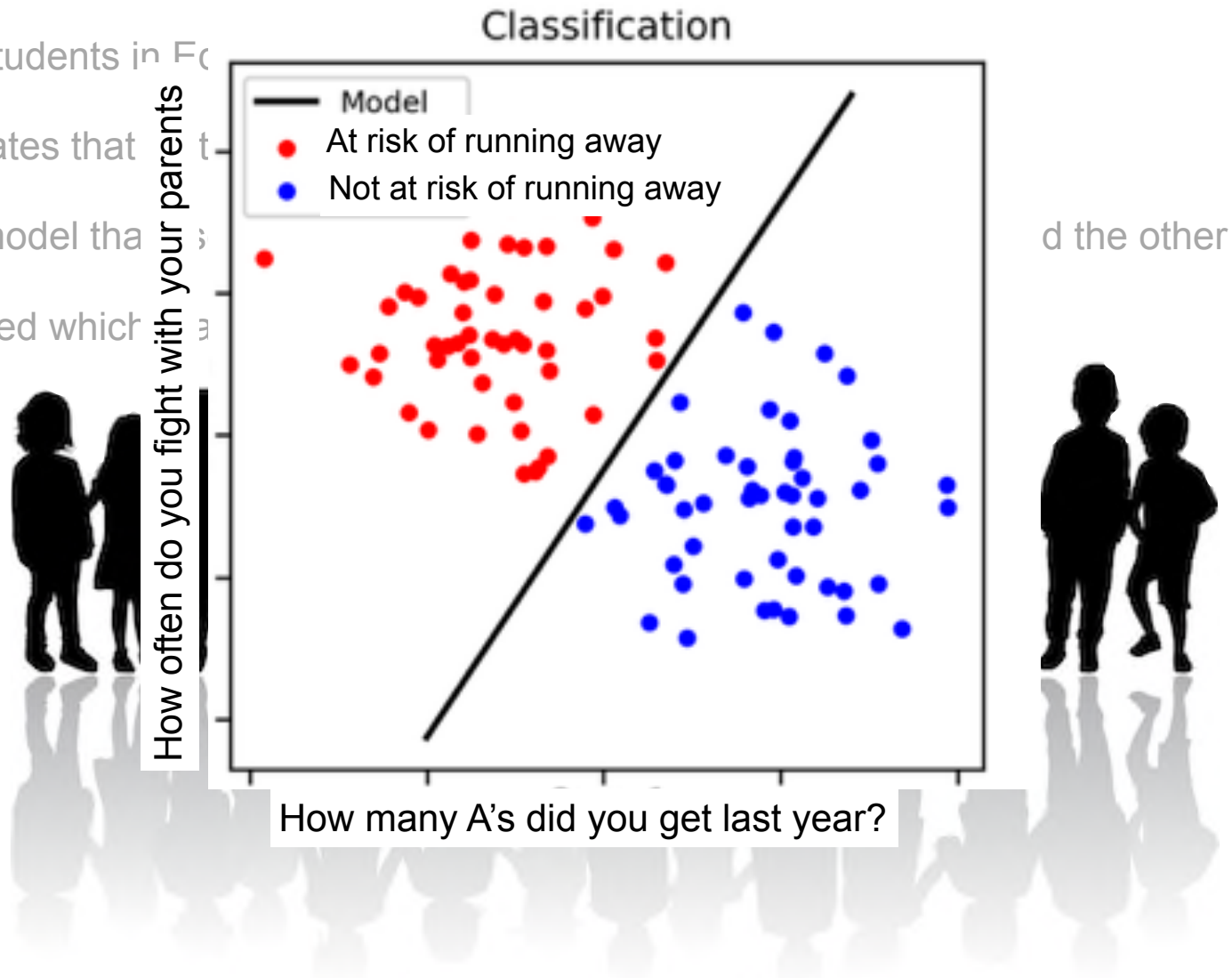
The Problem

- 180,000 people are reported missing in the UK every year (80,000 children)
- We want to be able to predict if a student is going missing
- The data: Survey of students in Ec
  - One question indicates that a st
- Build a classification model that d the other questions as variables
- Charity is also interested which



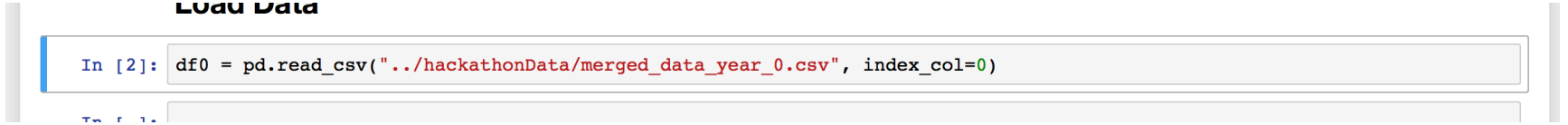
The Problem

- 180,000 people are reported missing in the UK every year (80,000 children)
- We want to be able to predict if a student is going missing
- The data: Survey of students in F
  - One question indicates that t
- Build a classification model tha s d the other questions as variables
- Charity is also interested which a

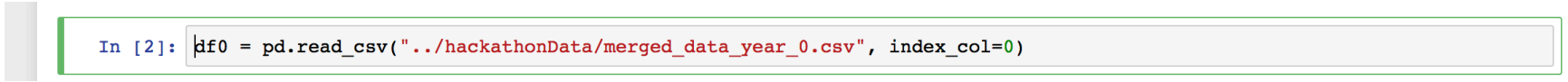


Jupiter Notebooks

- **Shift-return** to execute a cell
- Two modes:
  - **Blue** = command mode



    - Here: **up/down** to navigate cells, **a/b** for new cell above/below, **c** to copy cell **v** to paste cell
    - **Return** to get to edit mode
  - **Green** = edit mode



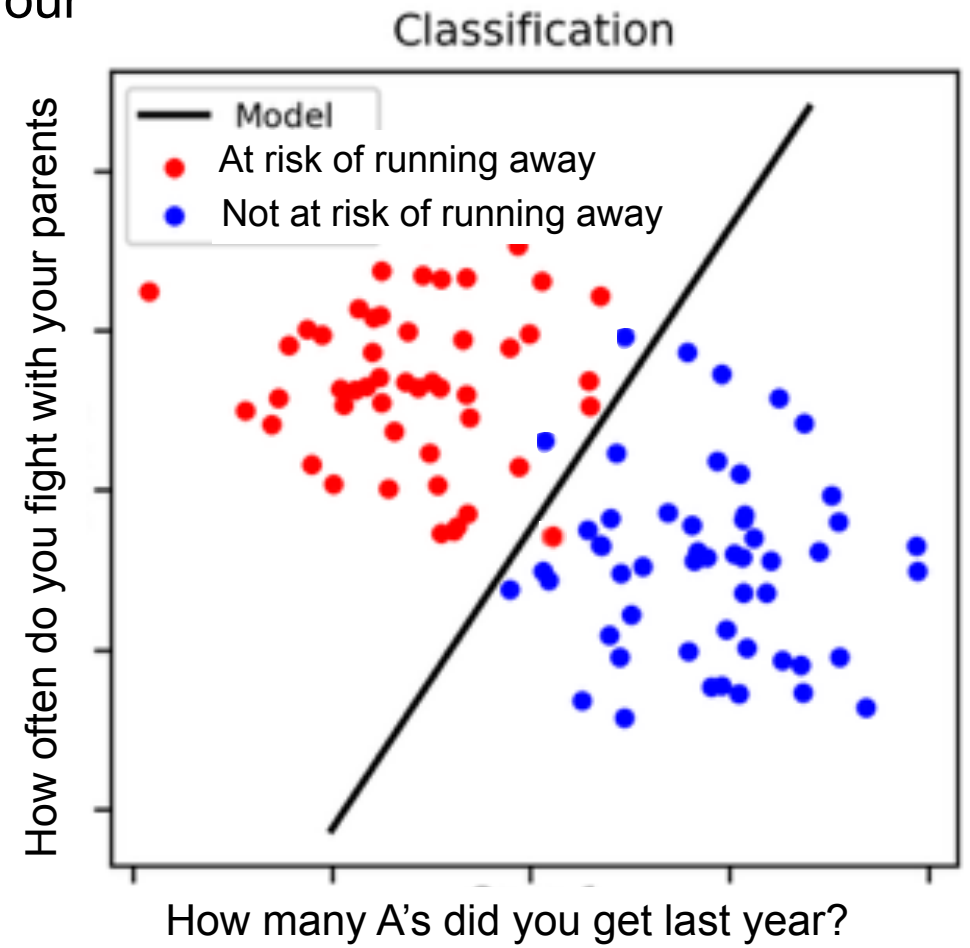
    - **Tab** for auto complete
    - **Shift-tab** for documentation
    - **Esc** to go to command mode

Go to Notebook

- **Shift-return** to execute a cell
- Two modes:
  - **Blue** = command mode
    - Here: **up/down** navigates, **a/b** new cells, **c/v** copy/paste
    - **Return** to get to edit mode
  - **Green** = edit mode
    - **Tab** for auto complete
    - **Shift-tab** for documentation
    - **Esc** to go to command mode

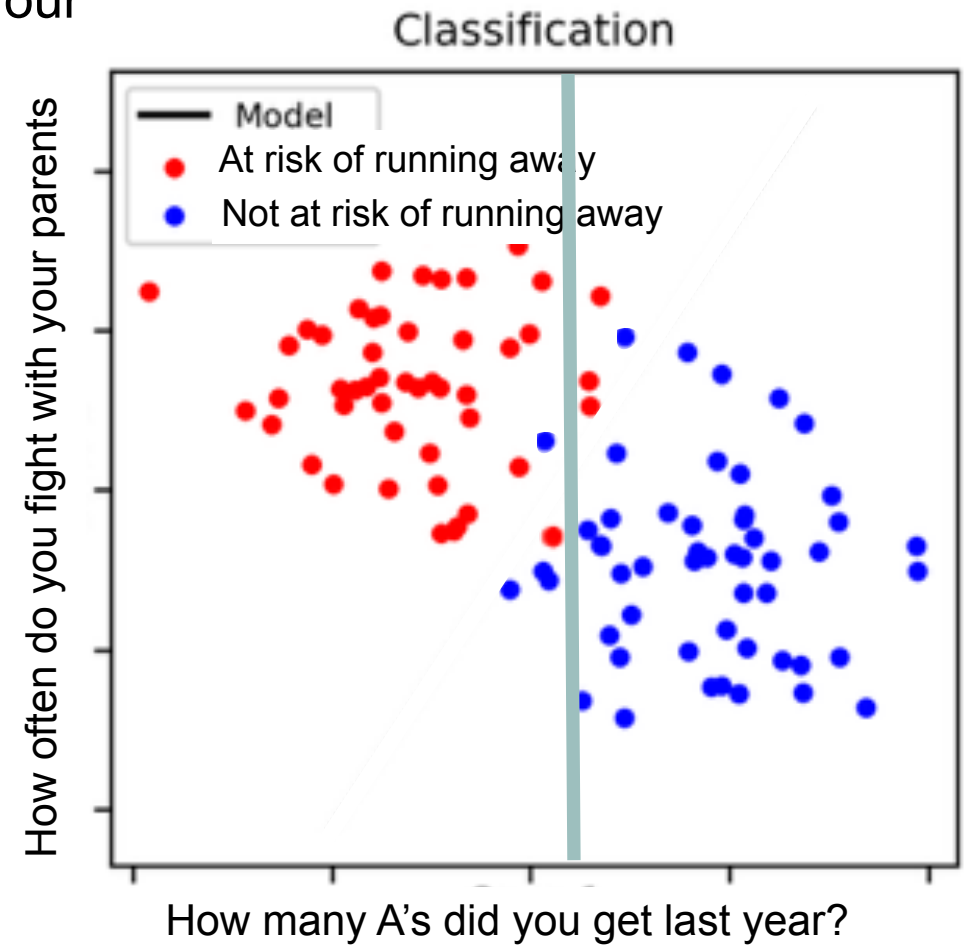
Training a model

- Use existing labeled data to find a mathematical equation, or set of rules on the other variables, that separates your data
  - “Labels" = “Target” = “y”
  - “Variables” = “Features” = “X”



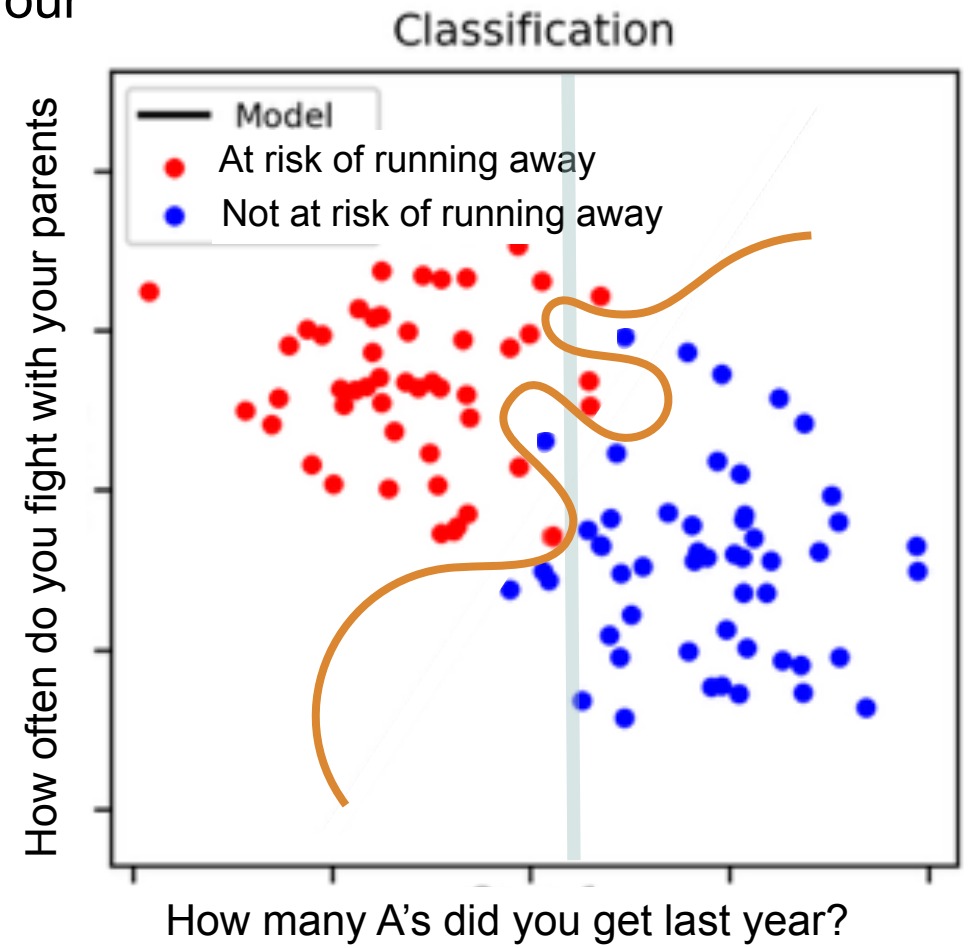
Training a model

- Use existing labeled data to find a mathematical equation, or set of rules on the other variables, that separates your data
  - “Labels" = “Target” = “y”
  - “Variables” = “Features” = “X”
- **Under-fitting**



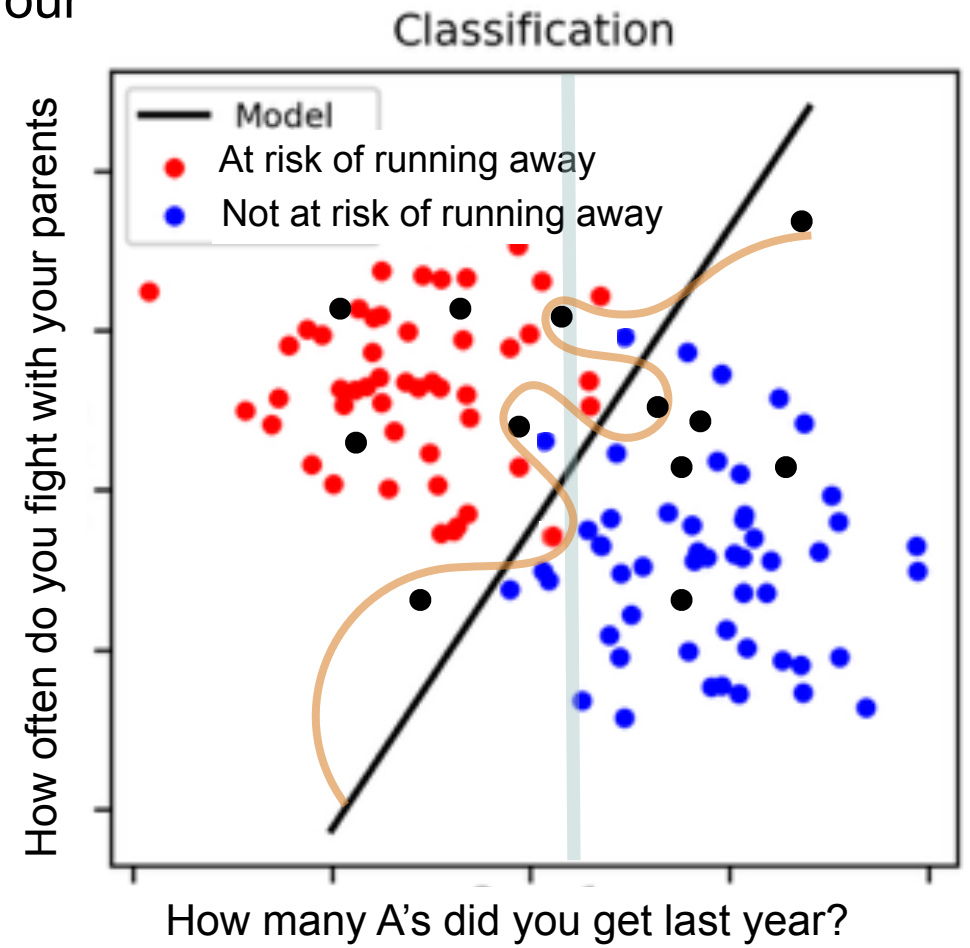
Training a model

- Use existing labeled data to find a mathematical equation, or set of rules on the other variables, that separates your data
  - “Labels" = “Target” = “y”
  - “Variables” = “Features” = “X”
- **Under-fitting**
- **Over-fitting**



Training a model

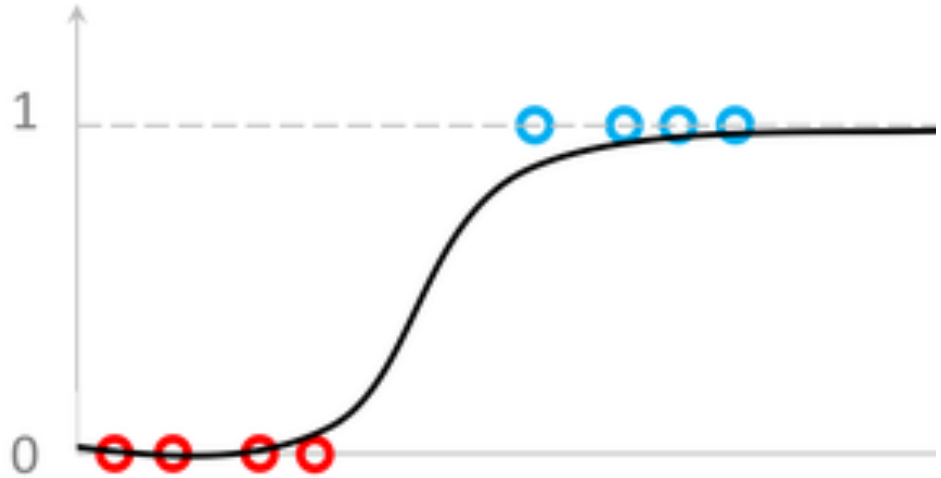
- Use existing labeled data to find a mathematical equation, or set of rules on the other variables, that separates your data
  - “Labels" = “Target” = “y”
  - “Variables” = “Features” = “X”
- **Under-fitting**
- **Over-fitting**
- Test your model on new data



Logistic Regression

- Basic classification ML algorithm
- Analog to linear regression
  - Replace linear function with logistic function

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$



- Predicts the probability that you will be in class 1

Go to Notebook

- **Shift-return** to execute a cell
- Two modes:
  - **Blue** = command mode
    - Here: **up/down** navigates, **a/b** new cells, **c/v** copy/paste
    - **Return** to get to edit mode
  - **Green** = edit mode
    - **Tab** for auto complete
    - **Shift-tab** for documentation
    - **Esc** to go to command mode

Model evaluation

Model evaluation

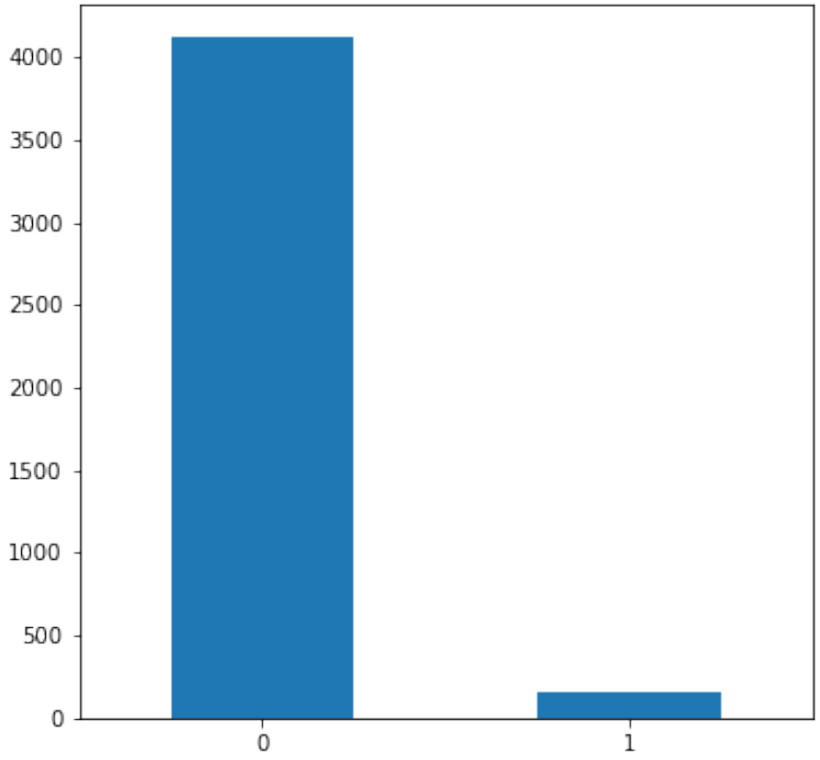
- **Precision**: **Percentage of positive predictions that were actually positive**
  - Percentage of students we said would run away that actually did
  - When we say someone is going to runaway, what is the probability that they actually will
- **Recall**: **Percentage of actually positives that were predicted as positives**
  - From students that run away, what percentage can we predict?
  - How much of the problem are we addressing?
-
  -
  -

Model evaluation

- **Precision**: **Percentage of positive predictions that were actually positive**
  - Percentage of people we said would run away that actually did
  - When we say someone is going to runaway, what is the probability that they actually will
- **Recall**: **Percentage of actually positives that were predicted as positives**
  - Percentage of people that did runaway that we predicted would run away
  - How much of the problem are we addressing?
- Need to think about the use case when deciding on a metric
  - Prioritise Recall: In this case we care more about not missing children that are going to run away
  - Still need to watch Precision, otherwise prediction doesn't mean anything

Unbalanced Data

- When one class of your data dominates the other
- If we labeled everything with 0 we get 96% accuracy
- ML models trying to minimise errors will tend to do this unless we do something to prevent it



Addressing unbalanced data

- Check if it's actually a problem
- Get more data
- Re-sample your data
  - Under Sample
  - Over Sample
    - Synthetic Minority Over-sampling Technique (SMOTE)
- Consider a different algorithm

Addressing unbalanced data

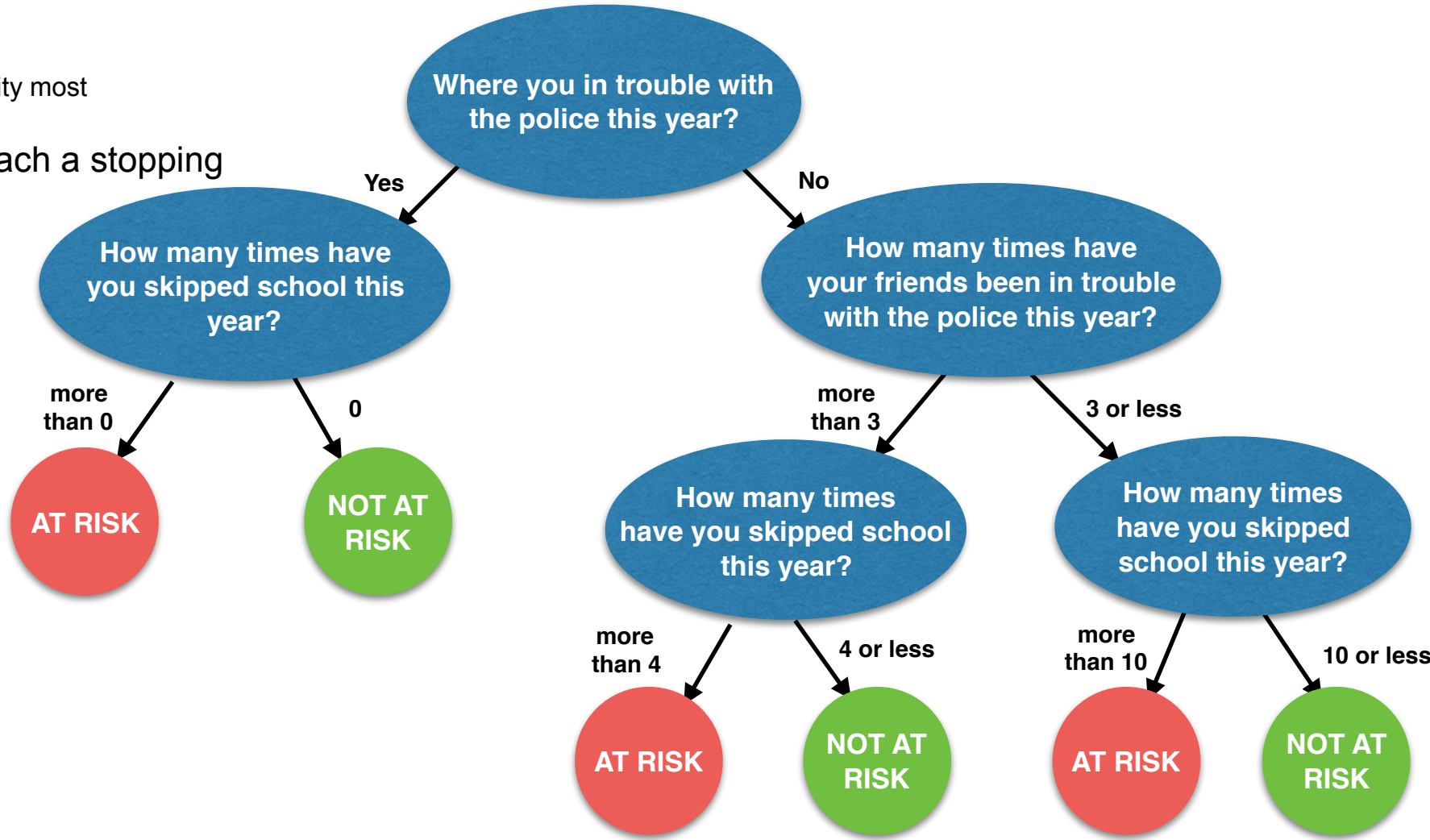
- Check if it's actually a problem
- Get more data
- Re-sample your data
  - Under Sample
  - Over Sample
    - Synthetic Minority Over-sampling Technique (SMOTE)
- Consider a different algorithm

Go to Notebook

- **Shift-return** to execute a cell
- Two modes:
  - **Blue** = command mode
    - Here: **up/down** navigates, **a/b** new cells, **c/v** copy/paste
    - **Return** to get to edit mode
  - **Green** = edit mode
    - **Tab** for auto complete
    - **Shift-tab** for documentation
    - **Esc** to go to command mode

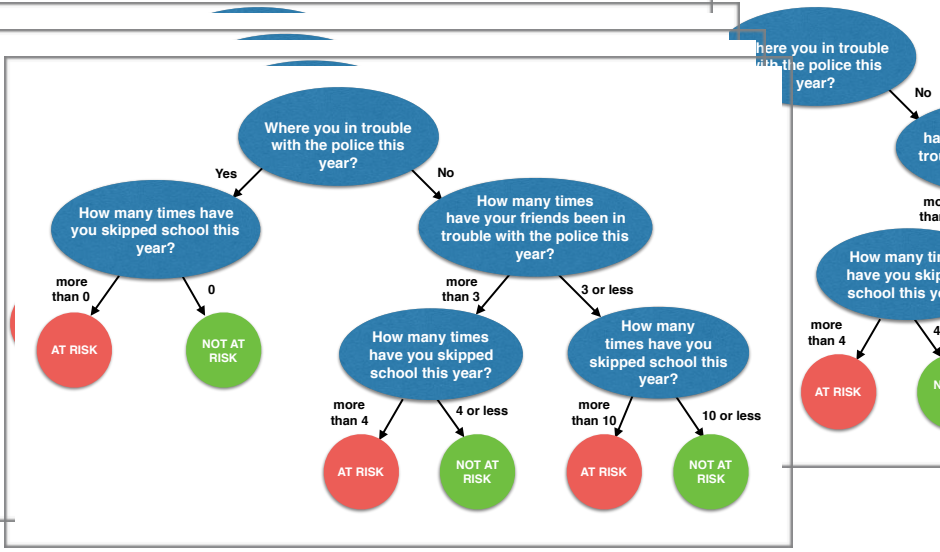
Decision trees

- Way of classifying data
- With each new branch to pick the rule that maximises the split of data
  - At each node pick rule that reduces impurity most
- Continue splitting data until you reach a stopping point
  - Depends on size of tree and leaf purity



Random Forests

- Single decision tree is at risk of over fitting
  - Radom forests build may decision trees
    - Each one is from a random subset of data points
    - Each split of the tree can only use a random subset of the features
    - Final modal takes the average decision from all decision trees
- Advantages:
  - Robust against over fitting
  - Feature importance
  - Less assumptions about data you are putting in



Go to Notebook

- **Shift-return** to execute a cell
- Two modes:
  - **Blue** = command mode
    - Here: **up/down** navigates, **a/b** new cells, **c/v** copy/paste
    - **Return** to get to edit mode
  - **Green** = edit mode
    - **Tab** for auto complete
    - **Shift-tab** for documentation
    - **Esc** to go to command mode

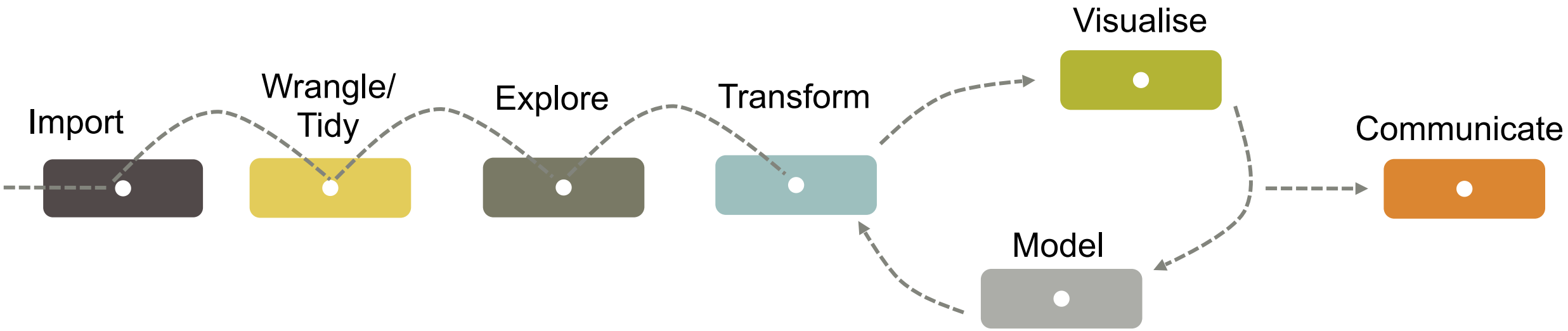
Random Forest Hyper-parameters

- Hyper parameters are parameters that affect *how* a machine learning model learns
- Google is your friend
  - “sklearn random forest”
- Some important features
    - **n_estimators:** The number of trees to use in your random forest
    - **max_features:** The size of the random set of features to choose from at each split
    - When to stop:
      - **max_depth -** How many levels of splits do you allow
      - **min_samples_split -** Nodes less samples than this value will be a leaf
      - **min_samples_leaf -** A split is only allowed if each side of the split leaves at least this many samples
      - **min_impurity_decrease -** Split is only performed if it decreases impurity by at least this much
      - **…**
    - Several others

Setting Hyper-parameters

- As you work on different problems you will start to develop an intuition about how to set the parameters
- Play with them and see how each effects your results
  - What is your precision? Recall?
  - Are you over fitting? Under fitting?
- Sklearn's CVGridSearch
  - Loops through all parameters using cross validation and selects the best set
  - Won't help you learn!
- Google is your friend

The Data Science process



Building a Machine Learning Model

http://r4ds.had.co.nz/introduction.html

Feature Importance

- The charity is very interested in knowing which features are most important for predicting a child's risk of going missing
- Random Forests provide feature importance, by calculating how much impurity reduction each variable accounts for

Go to Notebook

- **Shift-return** to execute a cell
- Two modes:
  - Blue = command mode
    - Here: **up/down** navigates, **a/b** new cells, **c/v** copy/paste
    - **Return** to get to edit mode
  - Green = edit mode
    - **Tab** for auto complete
    - **Shift-tab** for documentation
    - **Esc** to go to command mode

Next steps

- Tune Model
  - Random forest hyper parameters
  - Method of handling imbalance
  - Make sure you save data to do a final test once you have finished tuning
- Try different models
  - Gradient Boosting
  - Support Vector Classifier (SVC)
- Bring in data from other years
- Try to do some feature engineering

Exercise

Repo available here: **https://github.com/royalmail-datascience/Meetup2_SupervisedLearning**

Exercises

1. Vary hyper parameters and plot precision and recall as a function of that parameter
    1.1. n_estimators
    1.2. max_features
    1.3. min_samples_leaf
    1.4. Any other you want to play with
2. Try another classifier of your choice (e.g. Gradient boosting or SVC)

Thanks!

Slides and solution to exercises will be made available on the repo after the workshop

amundsen.glenn@royalmail.com for any questions!