goML Part 1

[Class Materials]

https://bit.ly/goML_Jan2021

DAY 2

Mr Seow Khee Wei / Mr Shubham Khare

Warm up!

Step 1: Go to the following url

https://bit.ly/3mLg2PG

Step 2: facilitator will walk you through the following question

1) **Write down two key learnings from lessons covered yesterday**

5 mins

Introduction

Name
Shubham KHARE (Ricky)

Email
shubham_khare@rp.edu.sg

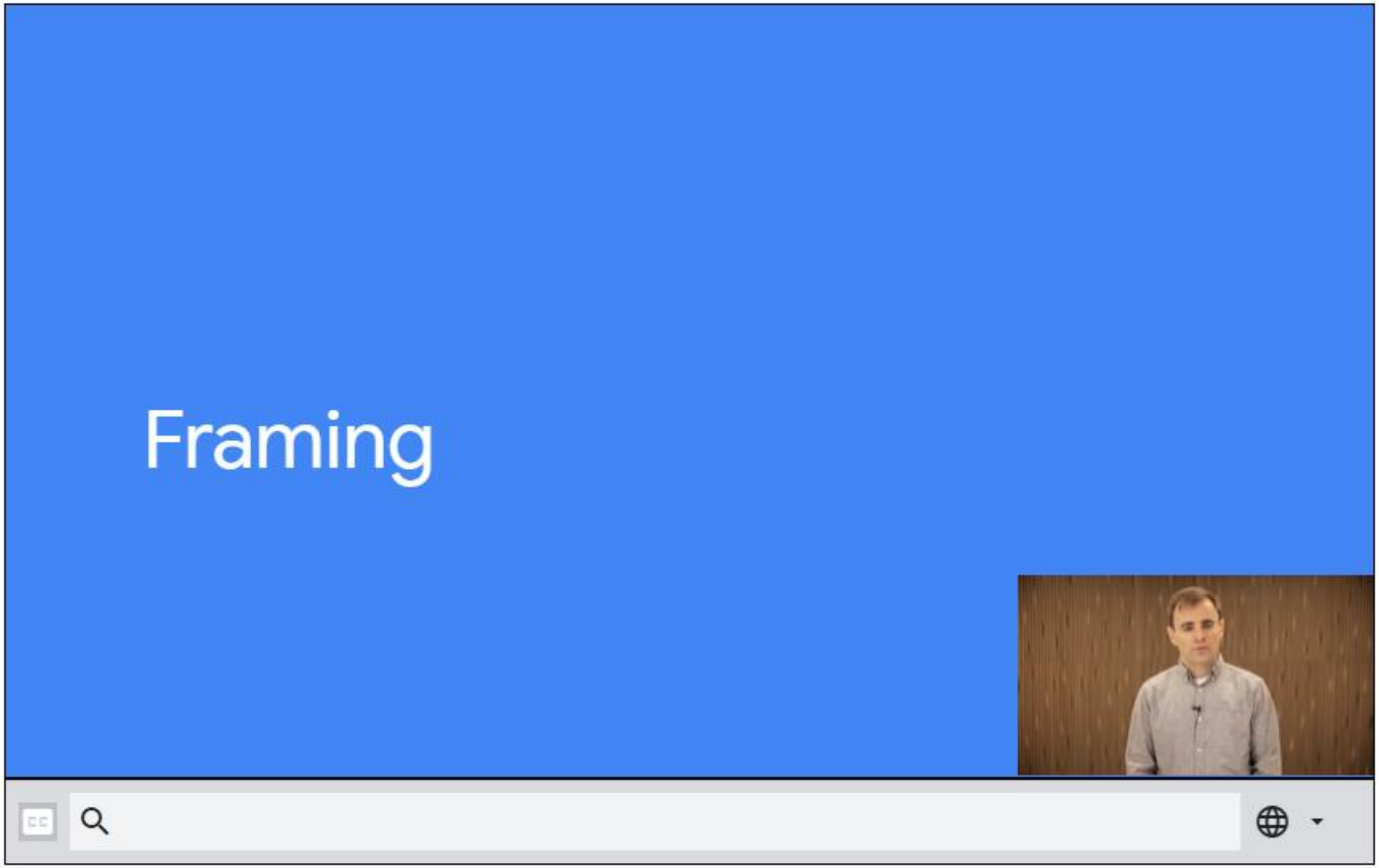
Telegram
@ricky_sk

Experience
AI Project Staff and AI Trainer
Data Scientist
Data Analyst
Project Engineer

Programmes

Day 1	What is Data? Types of Machine learning - Supervised and unsupervised learning Understand the common learning methods and how they are used to solve problems. What they can do and what they cannot do Use case sharing Activity – Numpy	Machine learning Workflow Develop a reusable pipeline for project workflow Data Visualization, Preparation and Cleaning Preparing your data for machine learning including feature engineering Activity – Pandas, Matplotlib, Seaborn, data prep
Day 2	Regression techniques - Training a Regression Model using Linear Regression - Training a Regression Model using Neural Network - HDB resale price predictor Classification techniques Activity – - Classification with Logistic Regression - Classification with Neural Network - Multi Class Classification	Model improvement Improve the performance of any model using simple hyperparameter tuning Activity: HyperParameter Create successful projects that matters Brainstorming to find your ML use case ML Project Checklist **Quiz**

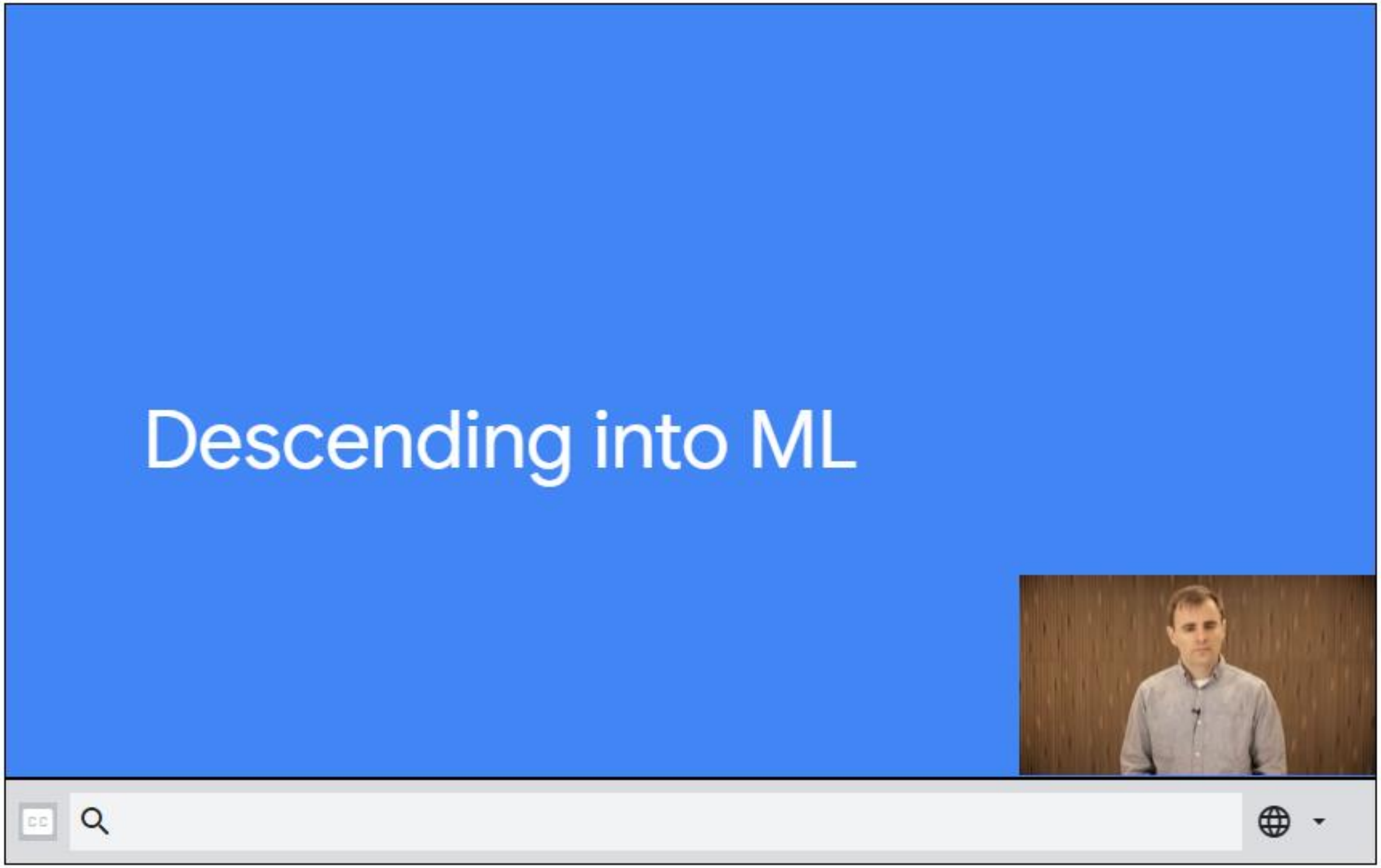
Recap



https://developers.google.com/machine-learning/crash-course/framing/video-lecture?hl=en

Regression

Regression



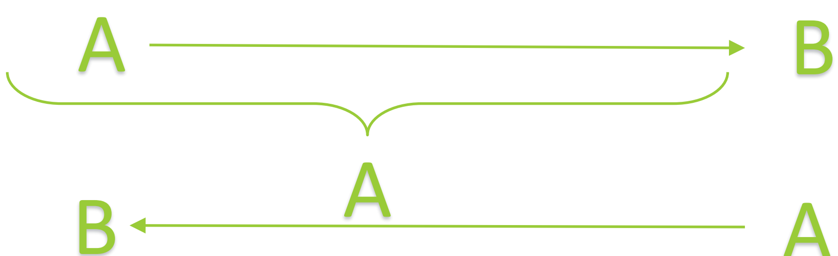
https://developers.google.com/machine-learning/crash-course/descending-into-ml/video-lecture?hl=en

Regression

- In simplest form, a linear regression is the analysis of **the relationships between two or more variables**
- It is also a statistical measure to determine the **strength of relationship** between one **dependent variable (Y)** with another set of changing variables called **independent variables (X1, X2, …)**
- Regression analysis is widely used for **prediction** and **forecasting**

Predicting a continuous number

Floor area sqm (Sqm)	Storey range	Street name	Remaining lease	Resale price ($)
131	01 TO 03	TAMPINES ST 45	74 years 01 month	540,000
120	10 TO 12	TAMPINES ST 71	74 years 11 months	578,888
119	04 TO 06	TAMPINES ST 71	74 years 10 months	545,000
121	01 TO 03	TAMPINES ST 83	66 years 10 months	520,000
146	04 TO 06	TAMPINES AVE 5	65 years 06 months	768,000
146	07 TO 09	TAMPINES AVE 5	66 years 05 months	755,000
148	04 TO 06	TAMPINES ST 11	63 years 09 months	655,000



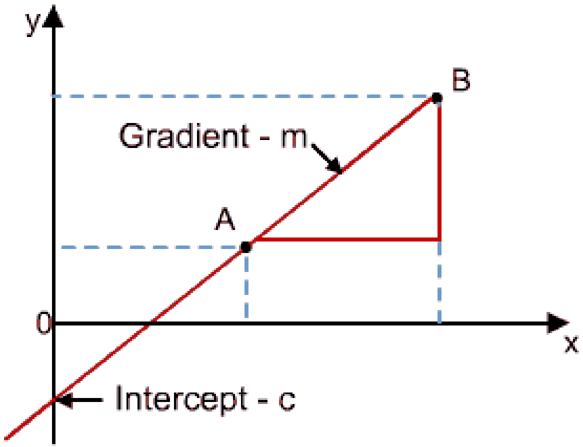
Data from data.gov.sg

Regression

- Independent variables, also called predictor or explanatory variables are regarded as inputs to an equation – usually denoted by X.
- Dependent variables, also called response variables are those values that change as a consequence of changes in other values in the equation – usually denoted by Y
- A simple linear model

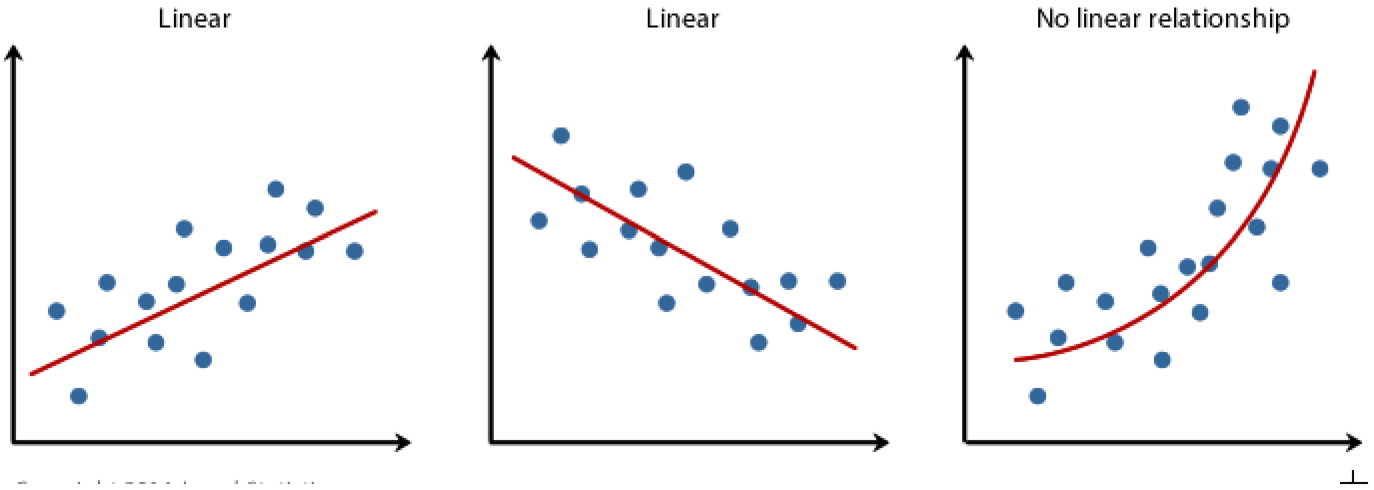
  Y = mX + c

  - Y = dependent variable
  - X = independent variable
  - m = slope of the line
  - c = y-intercept



Regression

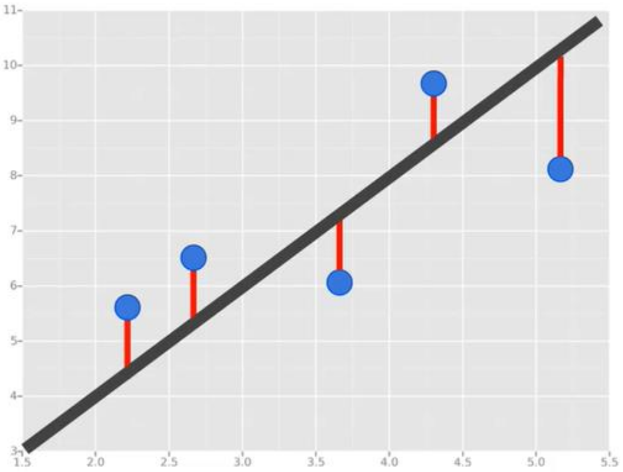
- Linear Regression has a straight-line relationship
- Non-linear regression implies a curved relationship
  - Logarithmic relationship



Regression

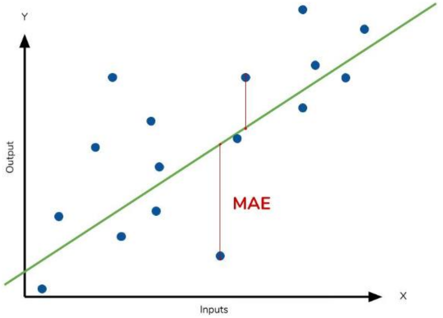
- Goal in regression is to draw a line that is as close as possible to every data point in our analysis.
- In linear regression, it is to minimise the vertical distance between all the data points and our line.
- Methods: **sum of square errors**, **least squares error**

- In Least Squares Method, we find a line of best fit by minimising the sum of squares of the residuals.
- The residual is the difference between the data point (y-value) and the fitted line.

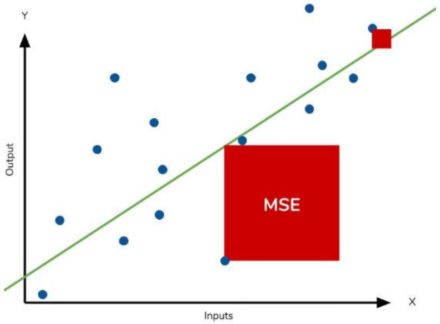


Regression – Error Metrics

- **MAE (Mean Absolute Error)**
  - Absolute difference between the data and model prediction
  - Does not indicate underperformance or overperformance(under or over actual data)
  - Small MAE suggests good prediction. MAE of 0 is a perfect model / predictor



- **MSE (Mean Square Error)**
  - Always larger than MAE (since we square the errors)
  - For comparing across models (reg1. vs reg2.) rather than within model
  - Outliers means adding quadratically to the MSE. Large values on one model shows signs of outliers



- **RMSE (Root Mean Square Error)**
  - Square root of MSE
  - Unit is back to the original and makes interpretation easier.
  - RMSE is standard deviation of the residuals (prediction errors).
  - Shows how spread out the errors are

Ref: www.dataquest.io/blog/understanding-regression-error-metrics

Shape of Data

- A data set is split into Training and Testing data sets.

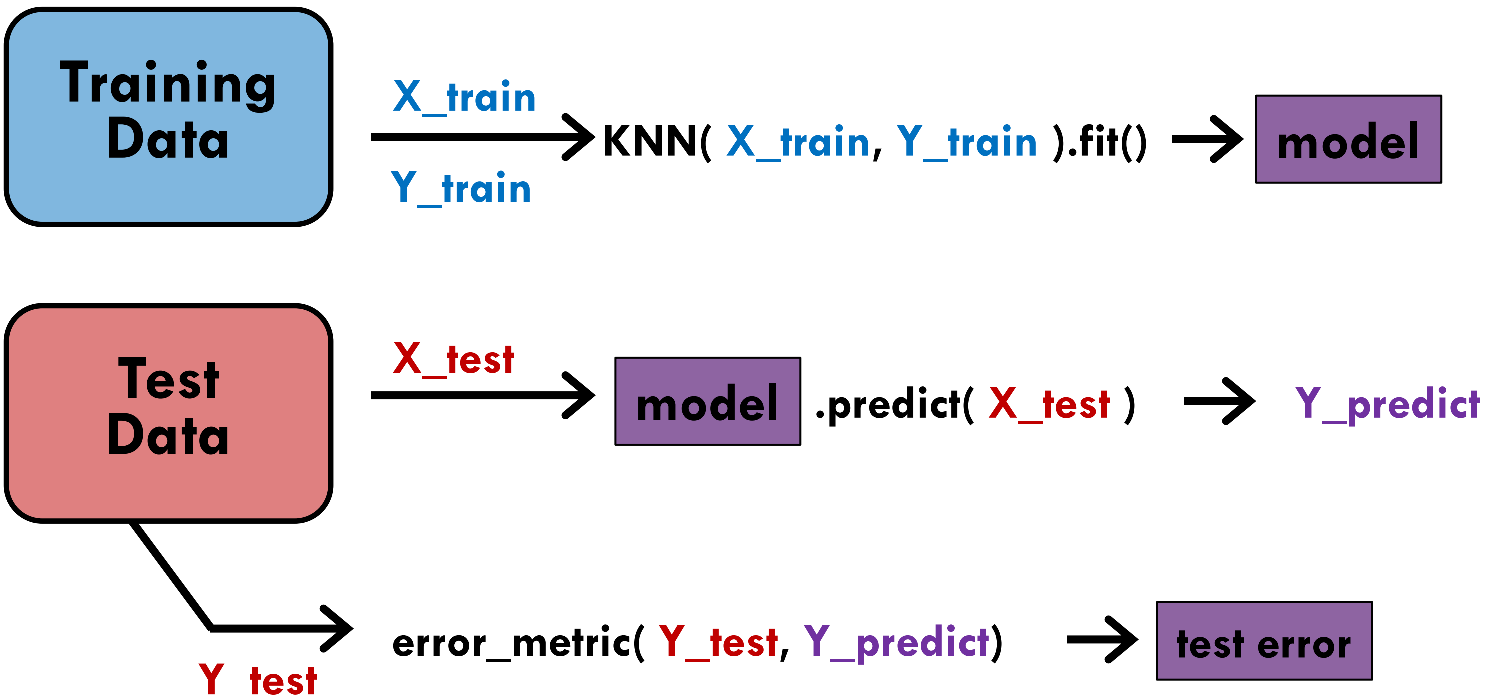
	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Training Data

Test Data

Shape of Data

- Obtaining the Performance Score.



Shape of Data

- Learning Curve
  - More than one test set: **Cross-Validation**

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Training Data 1

Validation Data 1

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Training Data 2

Validation Data 2

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Validation Data 3

Training Data 3

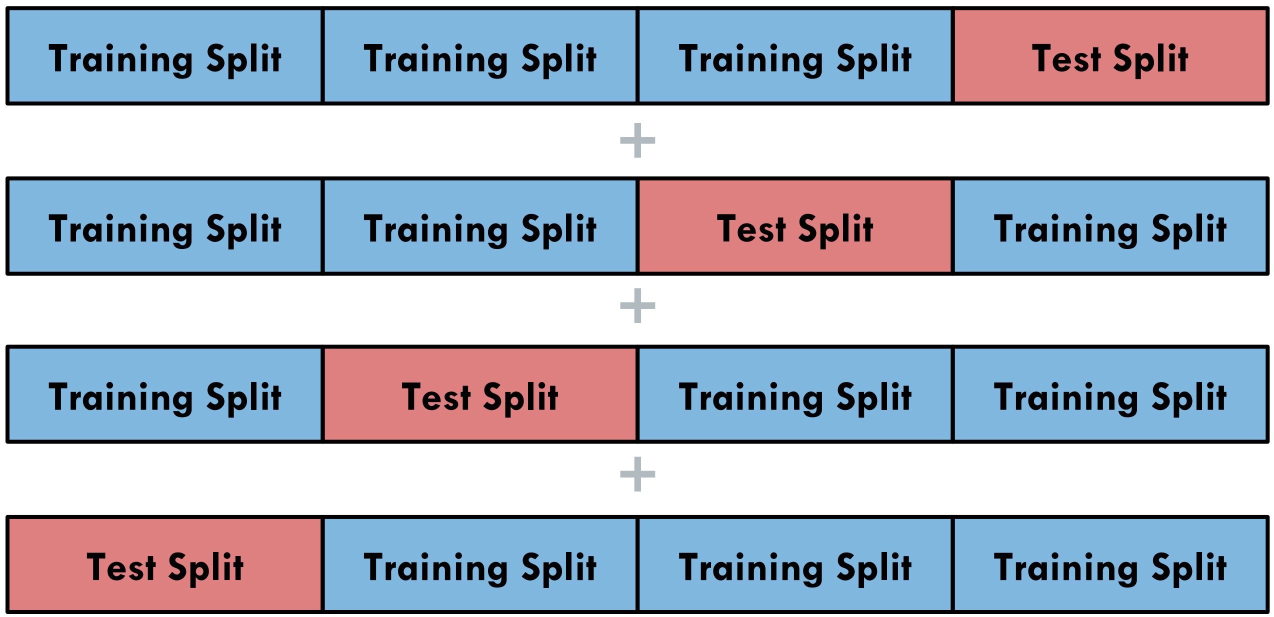
	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Validation Data 4

Training Data 4

Shape of Data

- Learning Curve
  - More than one test set: **Cross-Validation**



Average cross validation results.

Evaluation

The best way to evaluate the performance of an algorithm would be to make predictions for new data to which you already know the answers.

The second-best way is to use clever techniques from statistics called **resampling methods** that allow you to make accurate estimates for how well your algorithm will perform on new data.

1) Split into Train and Test Sets
2) K-fold Cross-Validation (divides the dataset into k folds. Stratified is to ensure that each fold of dataset has the same proportion of observations with a given label.)
3) Leave-one Out Cross-Validation
4) Repeated Random Test-Train Splits

https://medium.datadriveninvestor.com/k-fold-and-other-cross-validation-techniques-6c03a2563f1e

What Techniques to Use When

- Generally, **k-fold cross-validation is the gold standard** for evaluating the performance of a machine learning algorithm on unseen data with k set to 3, 5, or 10.
- Using a train/test split is good for speed when using a slow algorithm and produces performance estimates with lower bias when using large datasets.
- Techniques like leave-one-out cross-validation and repeated random splits can be useful intermediates when trying to balance variance in the estimated performance, model training speed and dataset size.

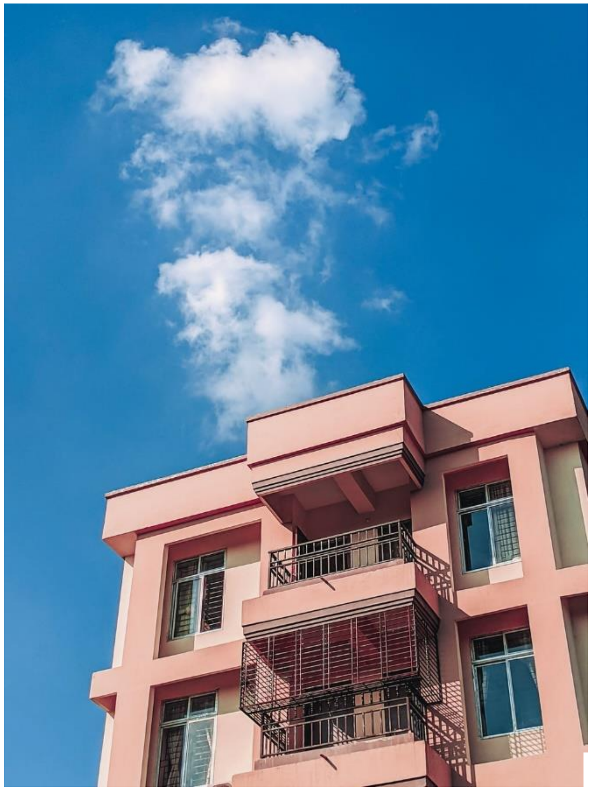
- The best advice is to experiment and find a technique for your problem that is fast and produces reasonable estimates of performance that you can use to make decisions. If in doubt, use 10-fold cross-validation.

Sample datasets for regression

- https://www.telusinternational.com/articles/10-open-datasets-for-linear-regression
- https://archive.ics.uci.edu/ml/datasets.php?format=&task=reg&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=table
- https://data.world/datasets/regression
- https://data.gov.sg/

Activity 2.1

- Training a Regression Model using Linear Regression



sq feet	num bedrooms	num bathrooms	sale price
785	2	2	170461
1477	2	2	271651
712	1	1	139912

Bedrooms	Bathrooms	Sq. Feet	Sale Price
3	2	2000	???

Exercise:
- Predict how much a 1500 sqft, 3 bedrooms with 3 bathrooms cost.

Keras pandas $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$ scikit learn TensorFlow

Step 1:
Watch and listen to the instructor's demonstration

15 mins

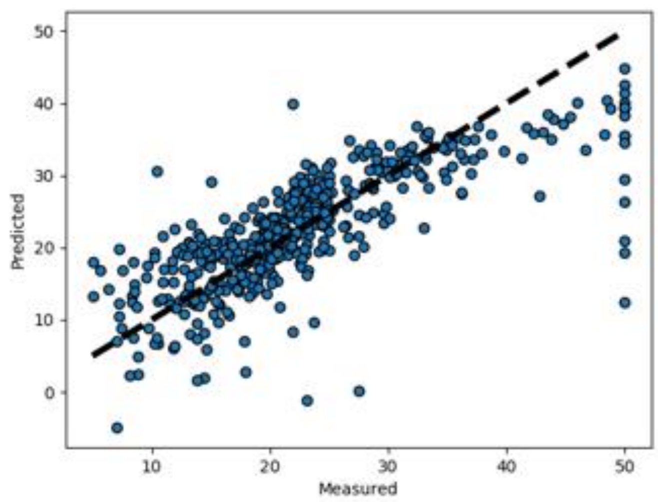
Step 2:
Work through the activities
Target to finish by 10:10

15 mins

Individual Activity

More Regression Algorithms

- **Linear Regression**
  - Linear regression assumes that the input variables have a Gaussian distribution. It is also assumed that input variables are relevant to the output variable and that they are not highly correlated with each other (a problem called collinearity).



More Regression Algorithms

- **Ridge Regression**
  - an extension of linear regression where the loss function is modified to minimize the complexity of the model measured as the sum squared value of the coefficient values (also called the L2-norm)
  - performs better against data that doesn't exactly follow the same pattern as the data the model was trained on.
  - A low alpha value can lead to over-fitting, whereas a high alpha value can lead to under-fitting.

More Regression Algorithms

- **LASSO regression**
  - The Least Absolute Shrinkage and Selection Operator (or LASSO for short) is a modification of linear regression, like ridge regression, where the loss function is modified to minimize the complexity of the model measured as the sum absolute value of the coefficient values (also called the L1-norm).

*L1-norm is also known as least absolute deviations (LAD), least absolute errors (LAE). It is basically minimizing the sum of the absolute differences between the target value and the estimated values

* L2-norm is also known as least squares. It is basically minimizing the sum of the square of the differences between the target value and the estimated values.

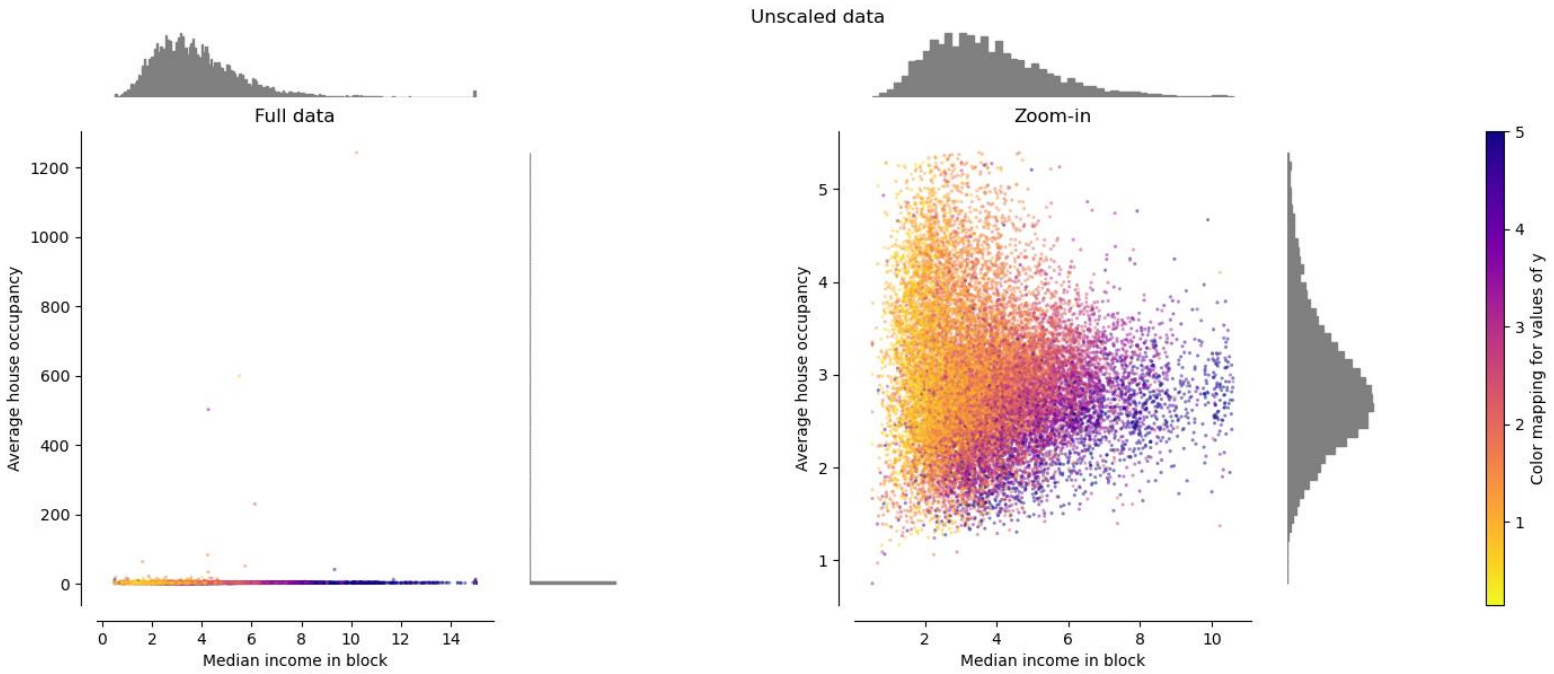
More Regression Algorithms

- **Elastic Net Regression**
  - a form of regularization regression that combines the properties of both Ridge Regression and LASSO regression. It seeks to minimize the complexity of the regression model (magnitude and number of regression coefficients) by penalizing the model using both the L2-norm (sum squared coefficient values) and the L1-norm (sum absolute coefficient values).

More Regression Algorithms

- **K-Nearest Neighbours**
  - The k-Nearest Neighbours algorithm (or KNN) locates the k most similar instances in the training dataset for a new data instance. From the k neighbours, a mean or median output variable is taken as the prediction.
- **Classification and Regression Trees**
  - use the training data to select the best points to split the data in order to minimize a cost metric. The default cost metric for regression decision trees is the mean squared error.
- **Support Vector Machines**
  - Support Vector Machines (SVM) were developed for binary classification. The technique has been extended for the prediction real-valued problems called Support Vector Regression (SVR).

Prepare Data – Data Transforms

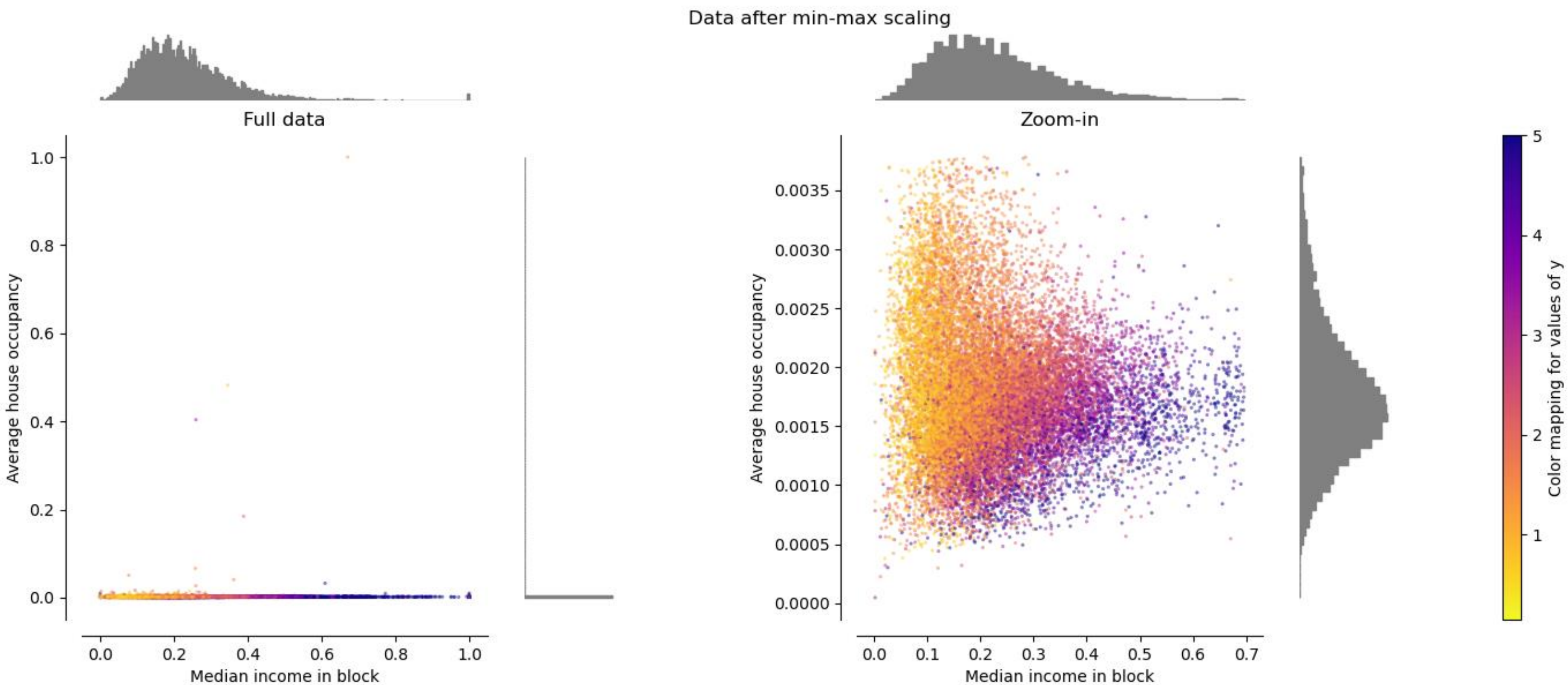


California Housing dataset

https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset

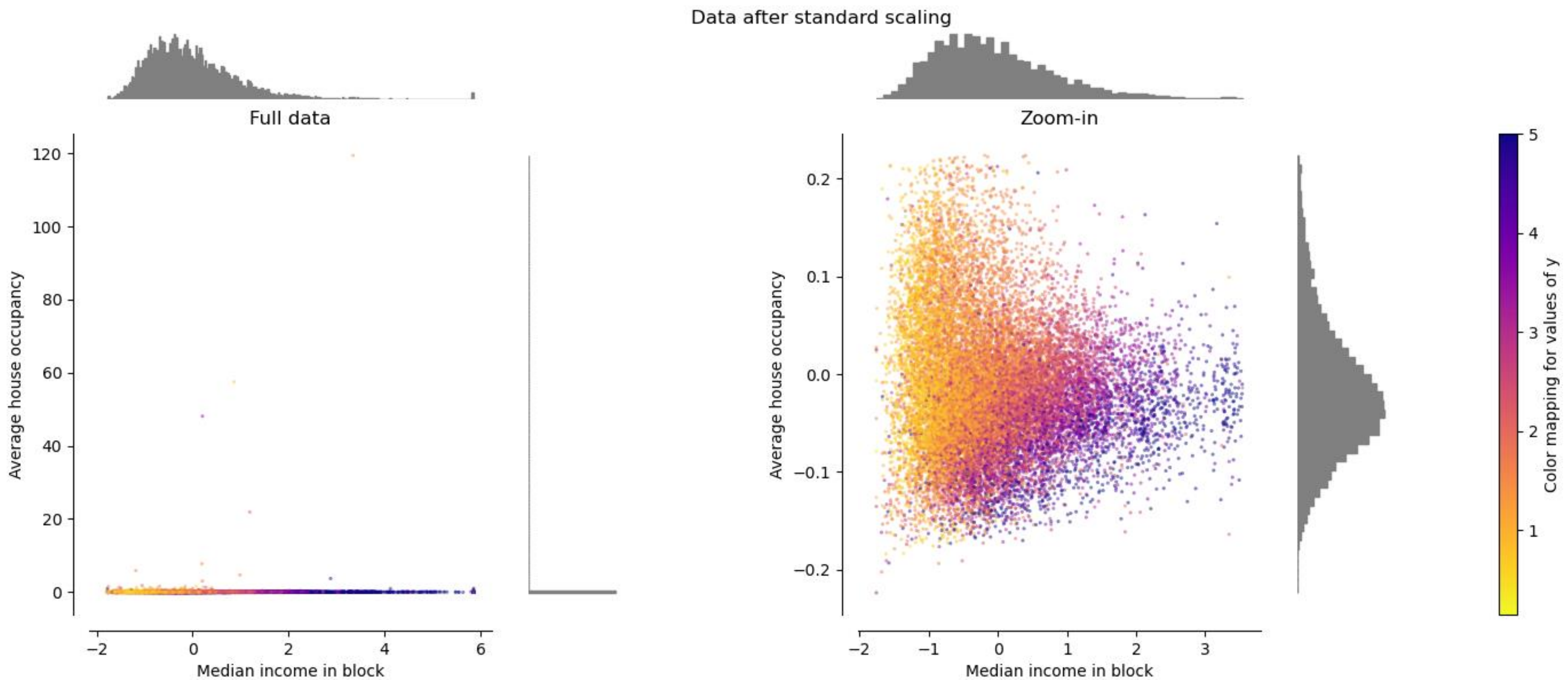
Prepare Data – Data Transforms

- **Rescale Data**
  - Often referred to as normalization and attributes are often rescaled into the range between 0 and 1
  - ***useful for algorithms that weight inputs like regression and neural networks and algorithms that use distance measures like k-Nearest Neighbours***
  - Use MinMaxScaler class



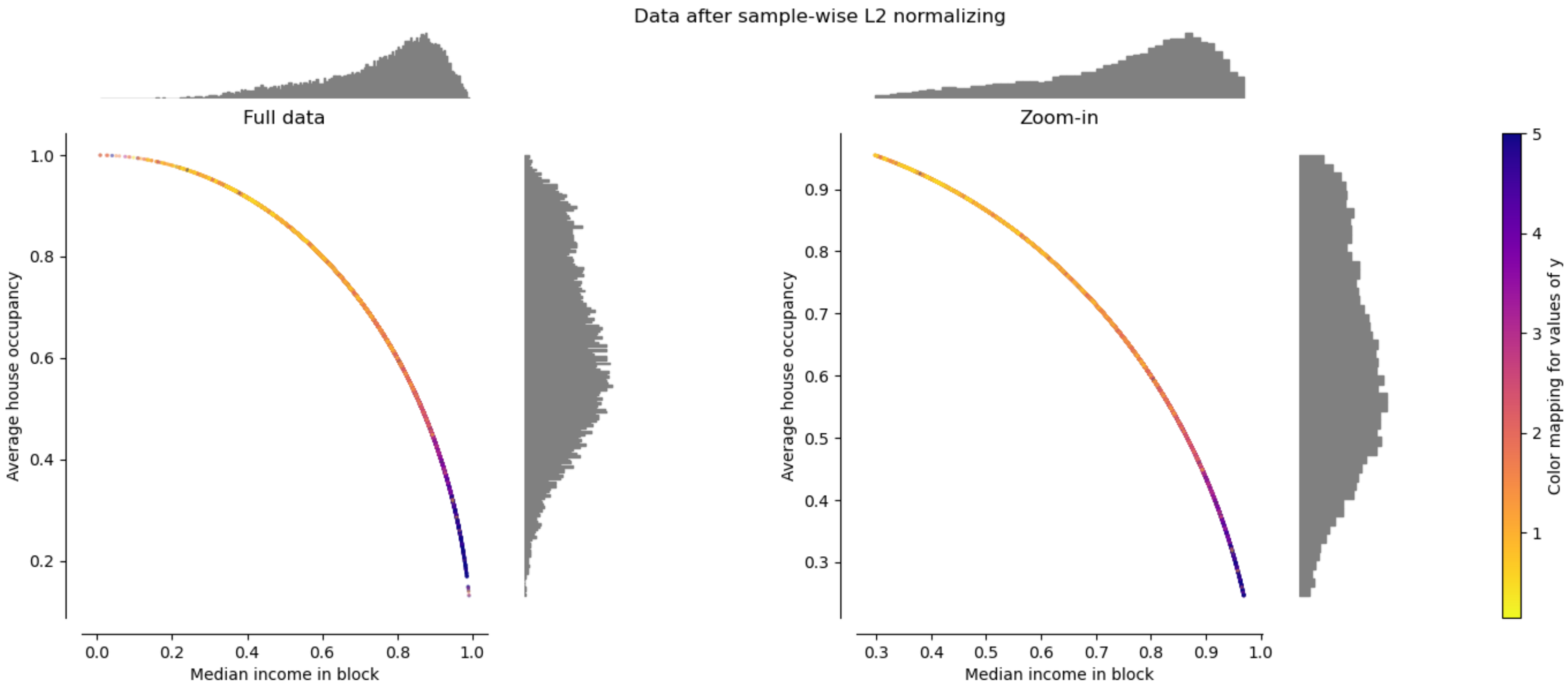
Prepare Data – Data Transforms

- **Standardise Data**
  - ***useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1 (removes the mean and scales the data to unit variance)***
  - Use StandardScaler class
  - Industry's go-to algorithm



Prepare Data – Data Transforms

- **Normalise Data**
  - rescaling each observation (**row**) to have a length of 1 (called a unit norm or a vector with the length of 1 in linear algebra)
  - ***useful for algorithms that use distance measures like k-Nearest Neighbours***
  - Use Normalizer class



Prepare Data – Data Transforms

- **Binarise Data**
  - All values above the threshold are marked 1 and all equal to or below are marked as 0
  - It is also ***useful when feature engineering and you want to add new features that indicate something meaningful.***
  - Use Binarizer class

References

- **Should I normalize/standardize/rescale**
  - http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-16.html
- **Scale, Standardize, or Normalize with Scikit-Learn**
  - https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02
- **Compare the effect of different scalers on data with outliers**
  - https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html

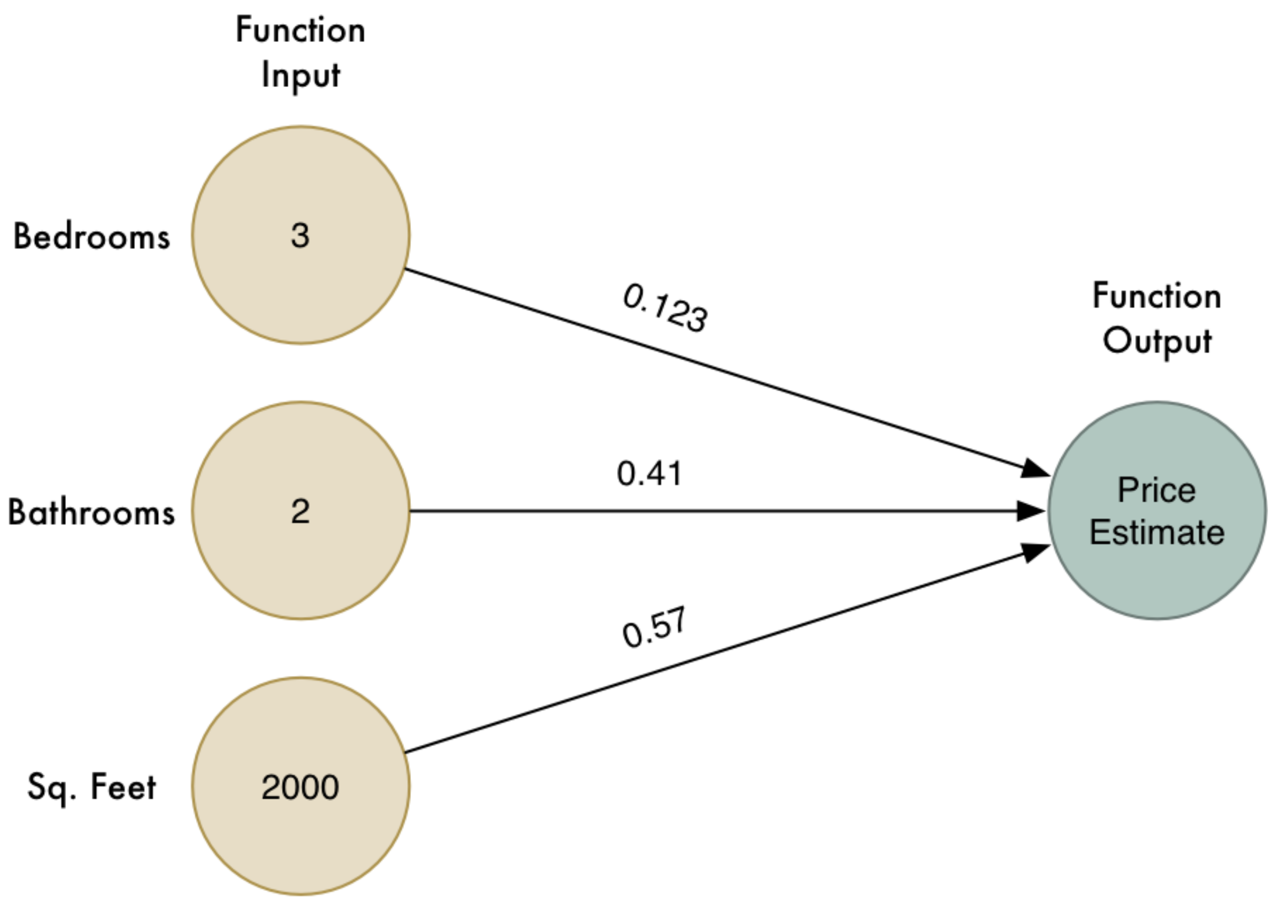
15 Mins Break

What is a Neural Network?

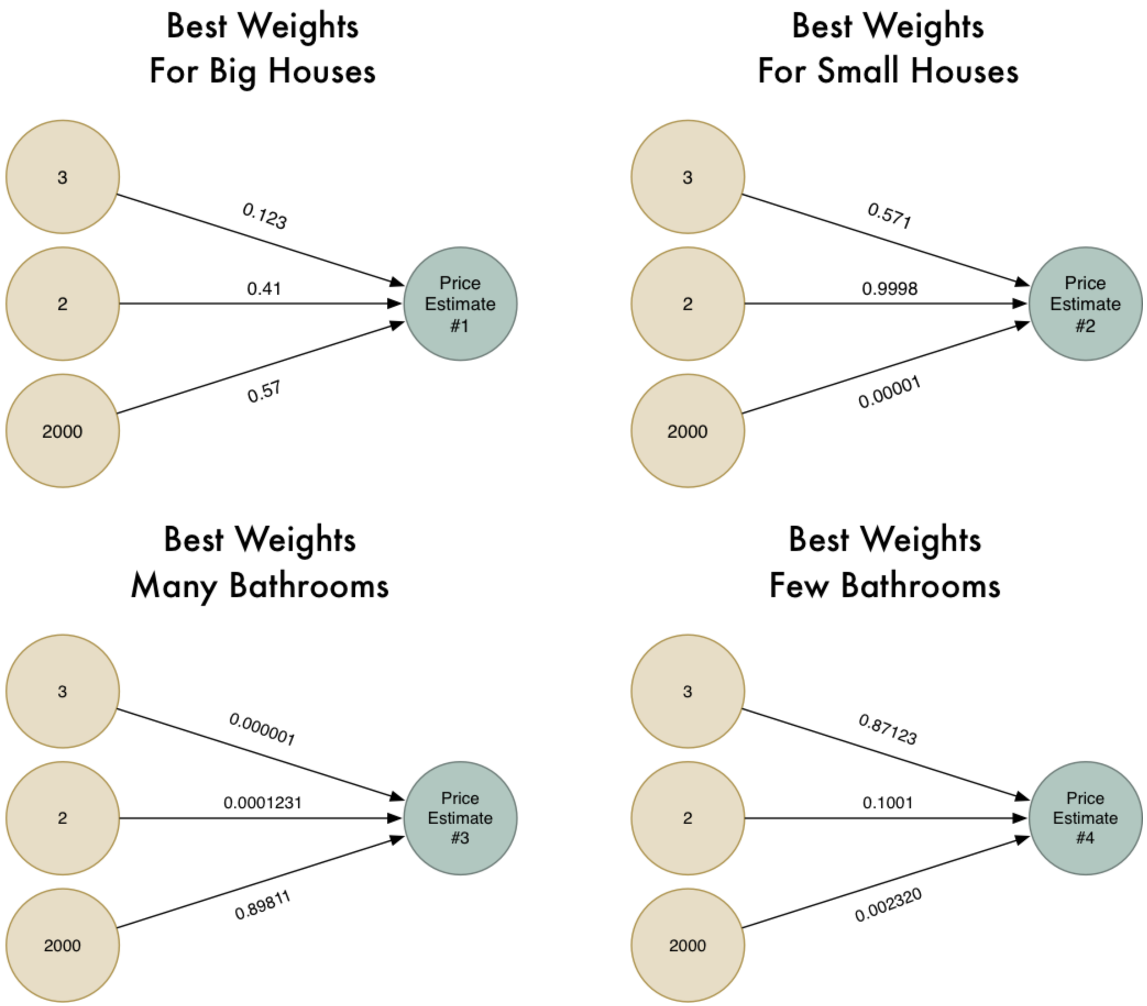
Is the previous activity, we created a simple estimation function:

```
def sales_price(bedrooms, bathrooms, sqft):
    price = 0

    # a little pinch of this
      price += bedrooms * 0.123

      # maybe a handful of this
      price += sqft * 0.56

      # a little extra salt for
      price += 201.23432095

      return price
```

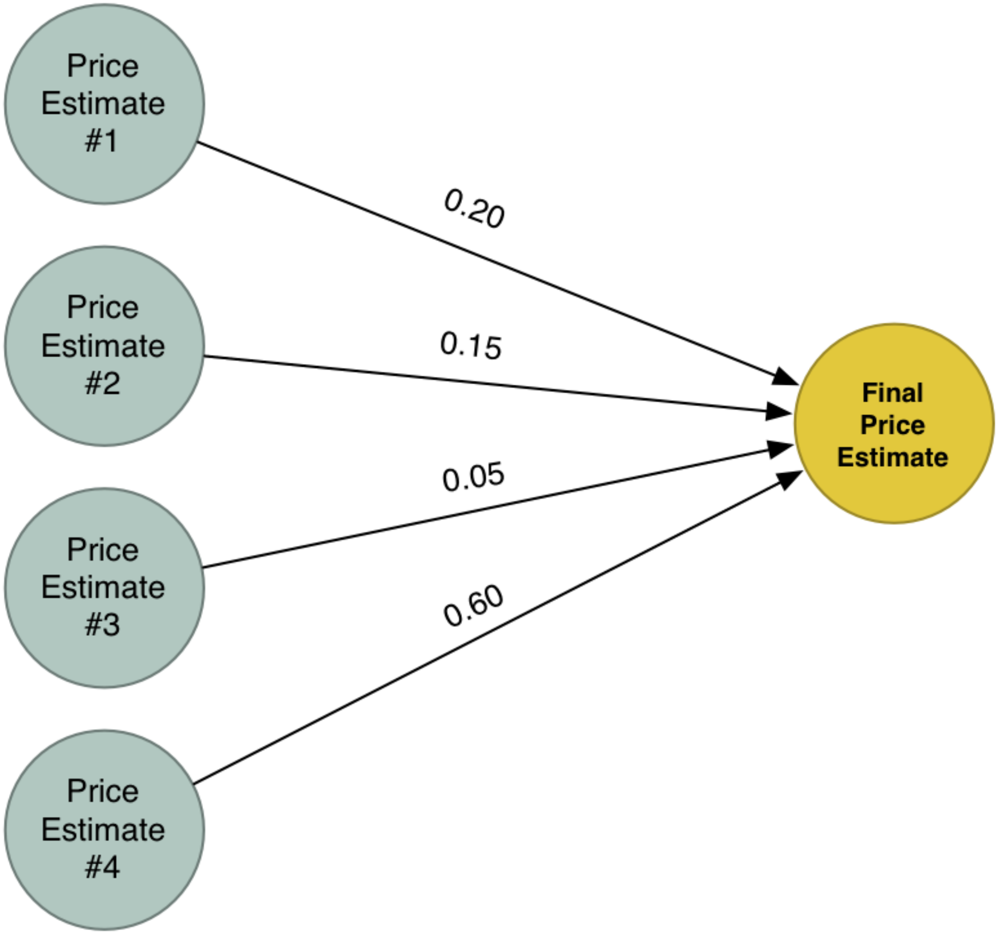
What is a Neural Network?



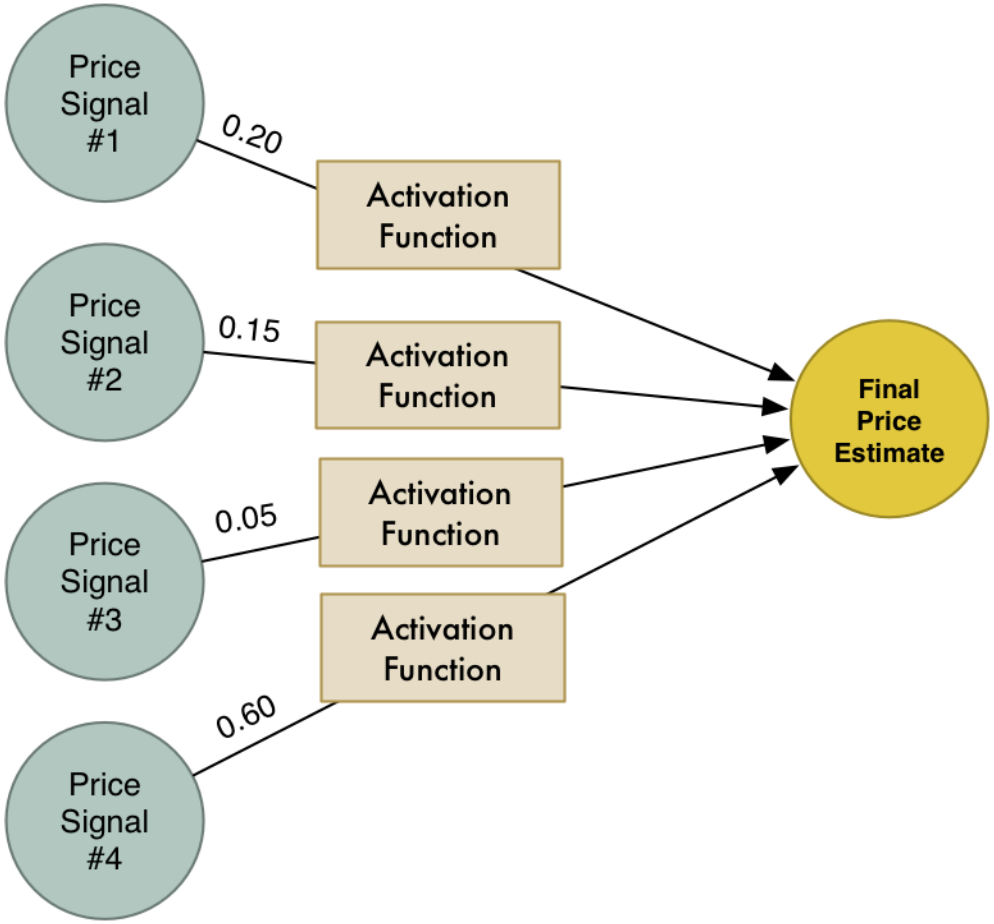
What is a Neural Network?



What is a Neural Network?



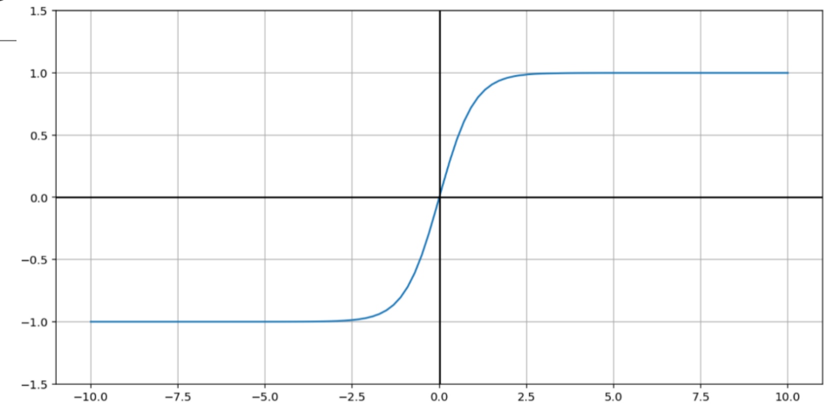
What is a Neural Network?



Activation Functions

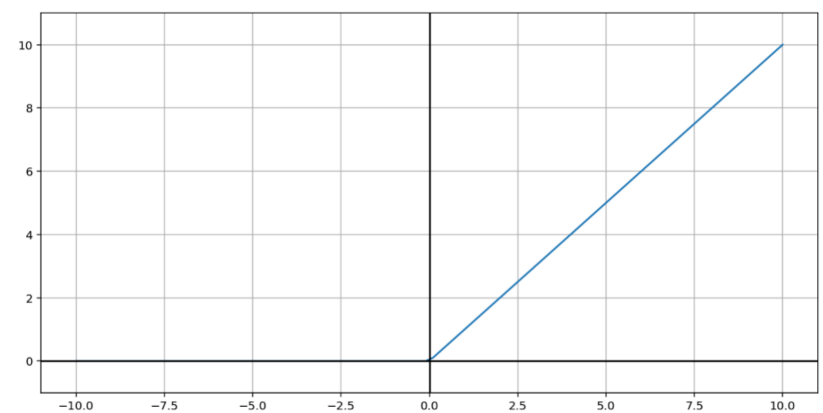
- Hyperbolic tangent function

$$tanh(z) = \frac{\sinh(z)}{\cosh(z)} = \frac{e^{2x} - 1}{e^{2x} + 1}$$



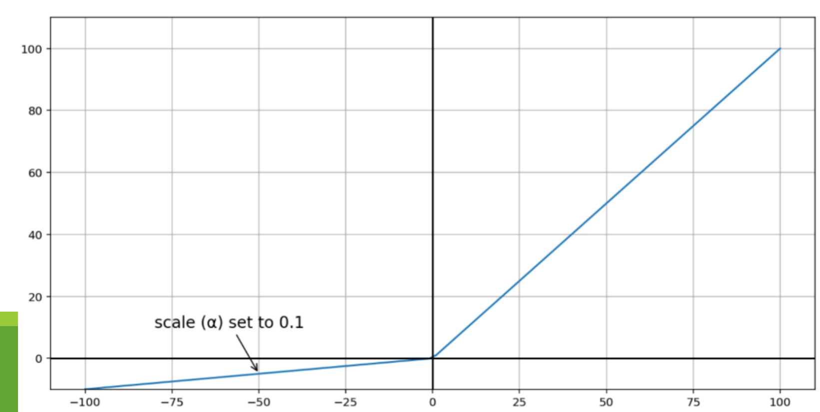
- Rectified Linear Unit (ReLU)

$$ReLU(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases}$$

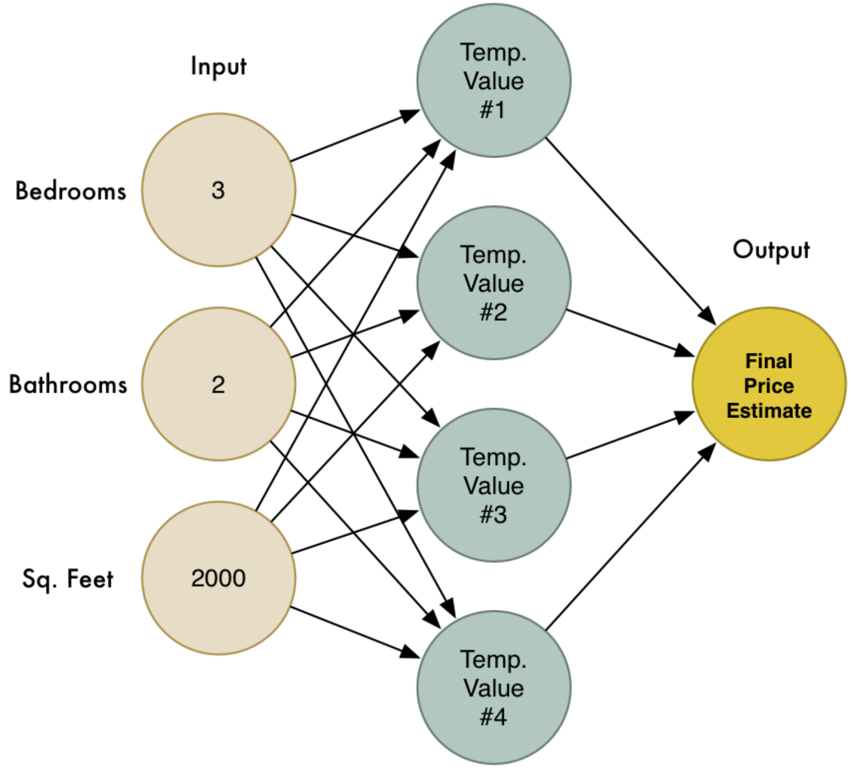


- "Leaky" Rectified Linear Unit (LReLU)

$$LReLU(z) = \begin{cases} \alpha z, & z < 0 \\ z, & z \geq 0 \end{cases}$$



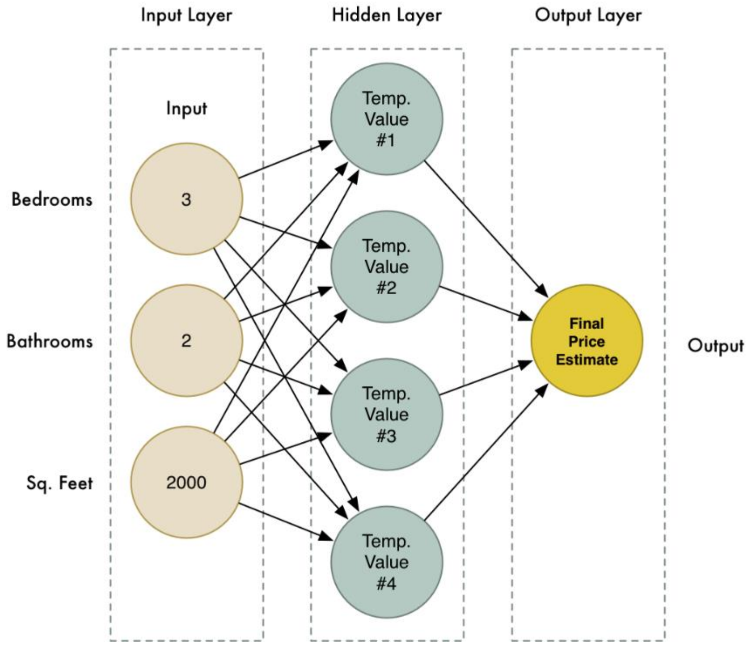
What is a Neural Network?



This is a neural network! Each node knows how to take in a set of inputs, apply a unique set of weights to them, and calculate an output value. Those output values pass through a non-linear activation function and a set of weights that makes each signal contribute more or less to the final value.

Each node is pretty simple by itself. But by chaining together lots of these nodes, we can model things that are too complicated to be modelled by one single neuron.

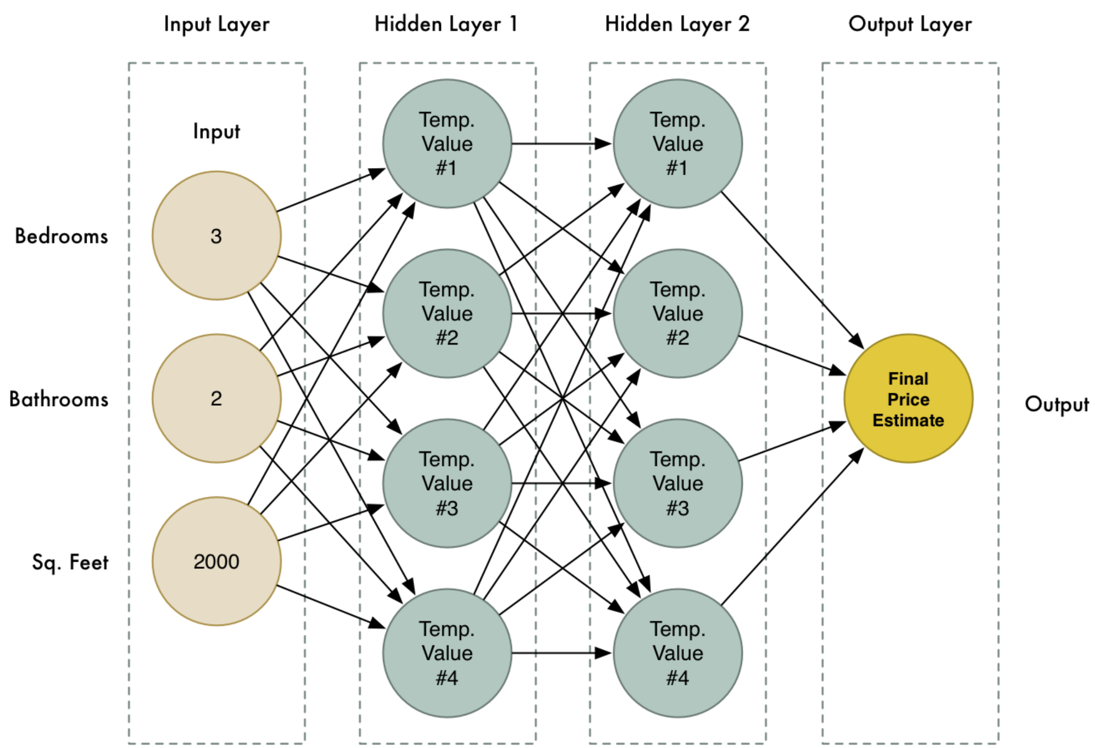
Stacking more layers



- The Input Layer has three input nodes where we pass in the input values for the current house.
- The Hidden Layer has four nodes, so it creates four price estimate signals based on the input values.
- The Output Layer has a single node, so it produces a single output.

Stacking more layers

- To make the neural network capable of modelling more complex relationships, we can add more hidden layers to it:



NN Require Feature Scaling

- Because neural networks take the initial input values as signals and recombine them over and over to produce a final value, **it's really important that the input signals are all numbers that are roughly the same size**.

Bedrooms	Bathrooms	Sq. Feet	Sale Price
1	1	400	$59,000
3	2	2000	$250,000
10	6	9500	$4,320,000

We'll just scale the data proportionally so the smallest values in each column are zero and the largest are one.
Here is the data again after we scale it:

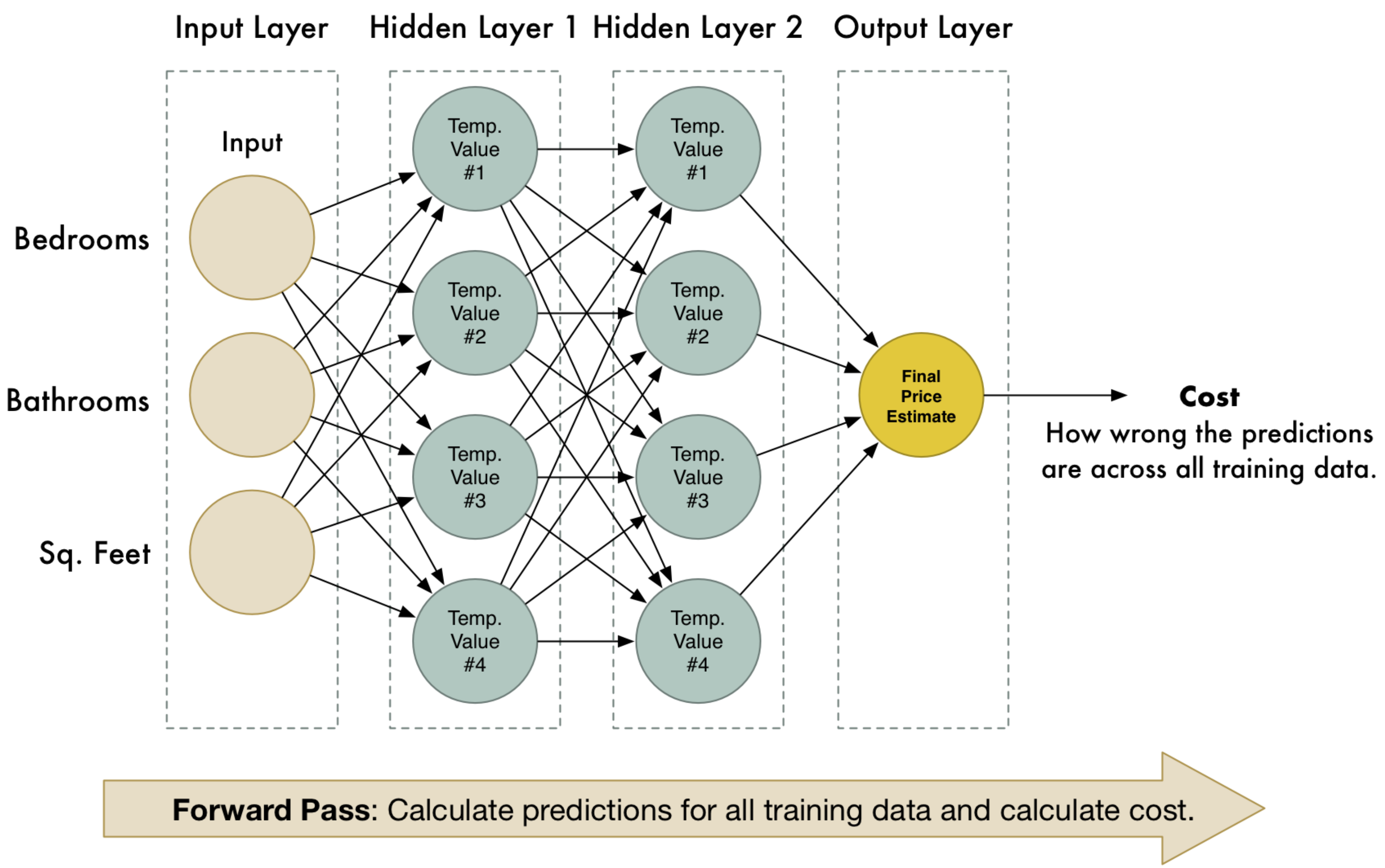
Bedrooms	Bathrooms	Sq. Feet	Sale Price
0.000	0.0	0.0000	0.0000
0.333	0.1	0.2197	0.0586
1.000	1.0	1.0000	1.0000

How to Train a NN

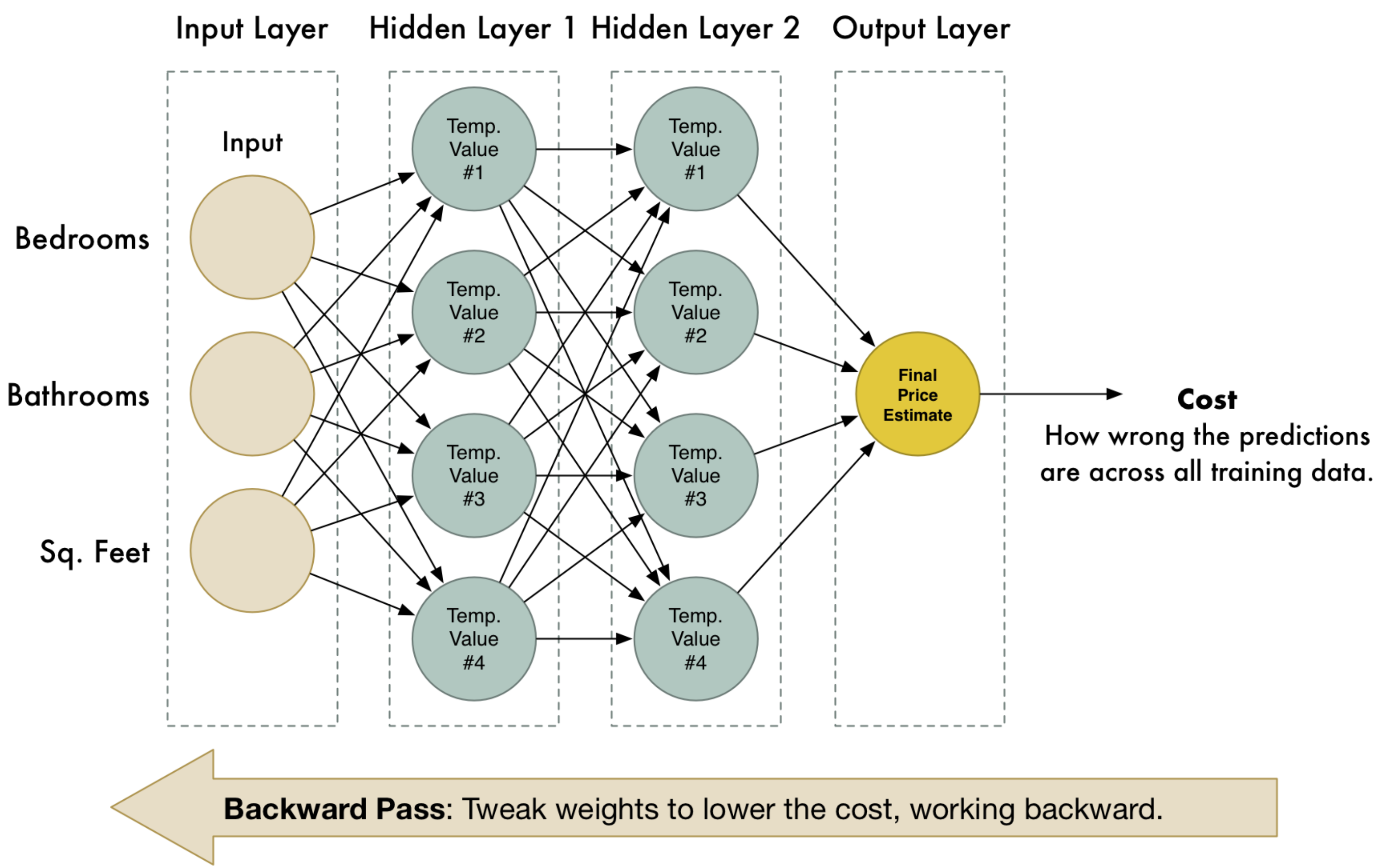
In the last section, we talked about how we find a good value for each of the weights in a linear regression model using a multi-step process:

- Set all of the weights in the model to an initial value like 1.0
- Run all of the training data through the model to get a prediction for each row of the training data.
- Use a cost function (also called a loss function) to calculate how wrong our predictions are across all the training data rows.
- Use the gradient descent algorithm to repeatedly tweak the weights in our model, which in turn adjusts our predictions with the goal of getting the cost as low as possible.

Forward pass

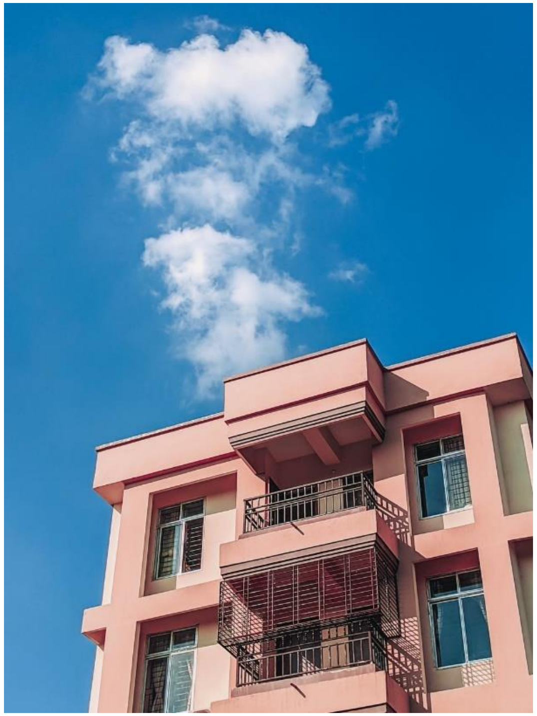


Backward pass



Activity 2.2

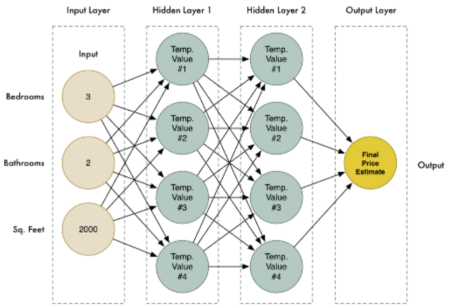
Use tensorflow & keras to train and use a neural network model to estimate the price of a house



sq feet	num bedrooms	num bathrooms	sale price
785	2	2	170461
1477	2	2	271651
712	1	1	139912

Bedrooms	Bathrooms	Sq. Feet	Sale Price
3	2	2000	???

Exercise:
- Add a hidden layer with 100 nodes. Did the performance improve?



Step 1:
Watch and listen to the instructor's demonstration

20 mins

Step 2:
Work through the activities

Target to finish by 11:20

Individual Activity

10 mins

Activity 2.3 - HDB price predictor

Exercises:

- Add different regression algorithms and evaluate performances

Target to finish by 11:50

Step 1:
Watch and listen to the instructor's demonstration

15 mins

Step 2:
Work through the activities

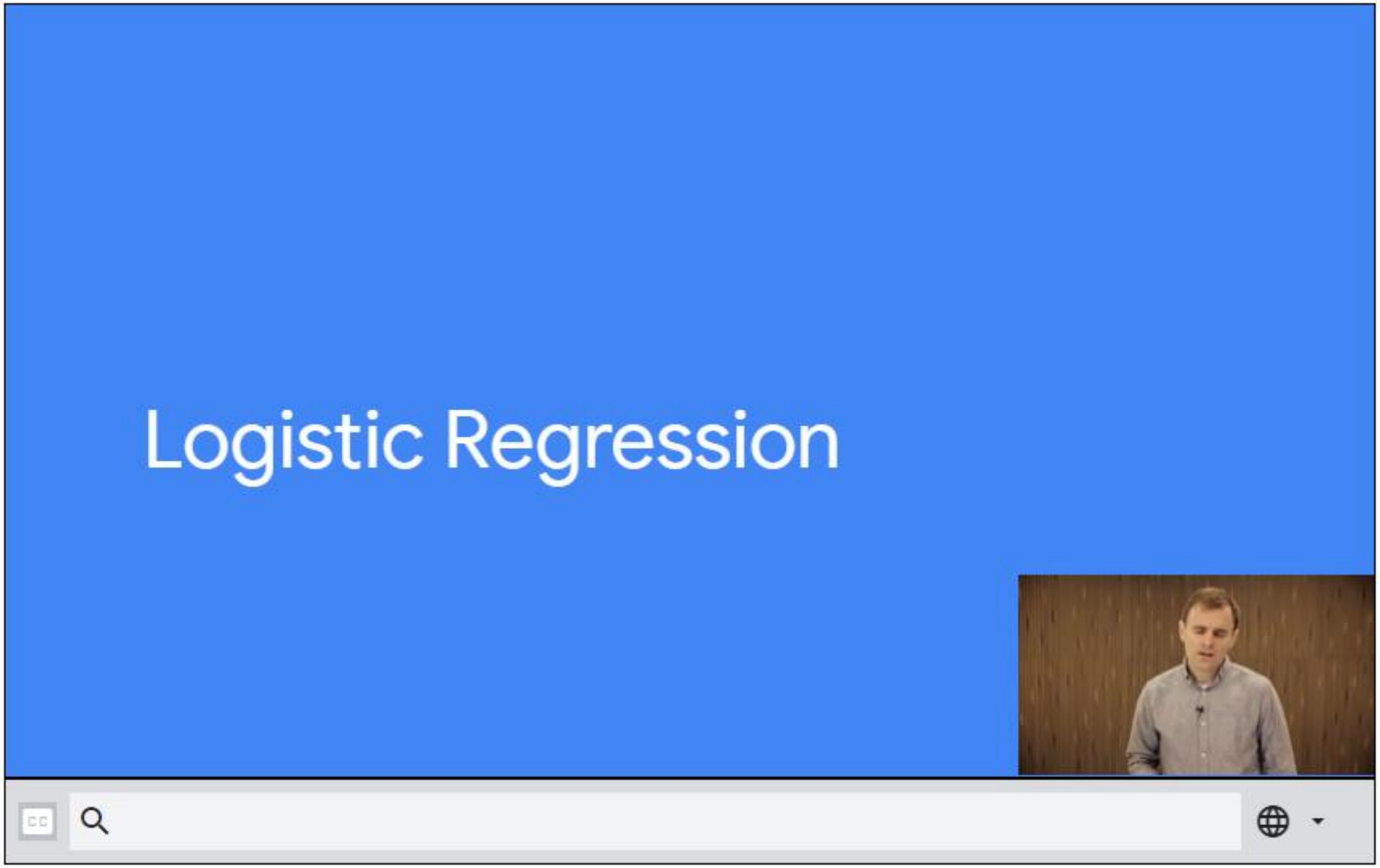
15 mins

Individual Activity

Classification

Logistic Regression

- Logistic Regression
  - used as a method for classification.
  - measures relationship between a binary dependent variable and one or more independent variables.
  - used for predicting the probability of occurrence of an event.
- Some examples of binary classifications models are:
  - Loan Default (yes/no)
  - Cancer or not-cancer
  - Spam vs non-spam



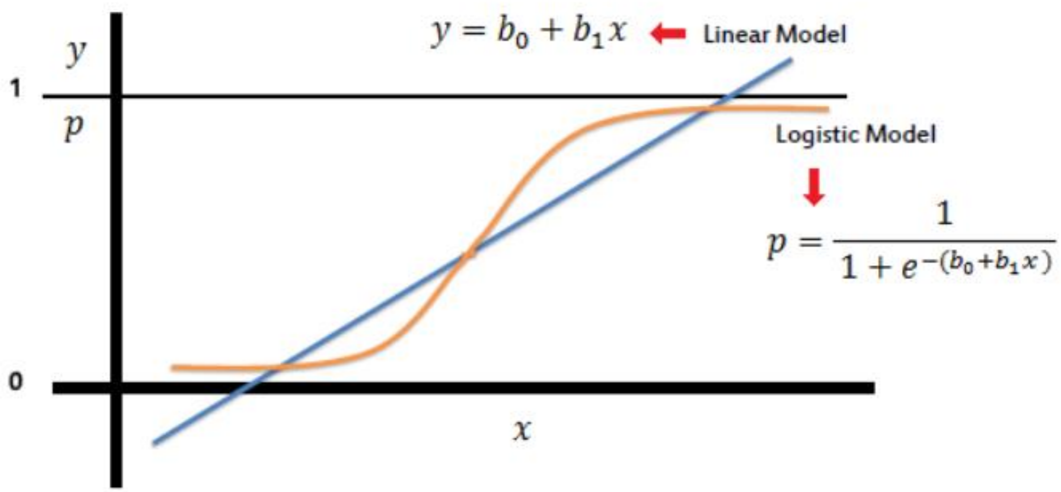
https://developers.google.com/machine-learning/crash-course/logistic-regression/video-lecture

Logistic Regression

- A simple Binary Logistic Regression model

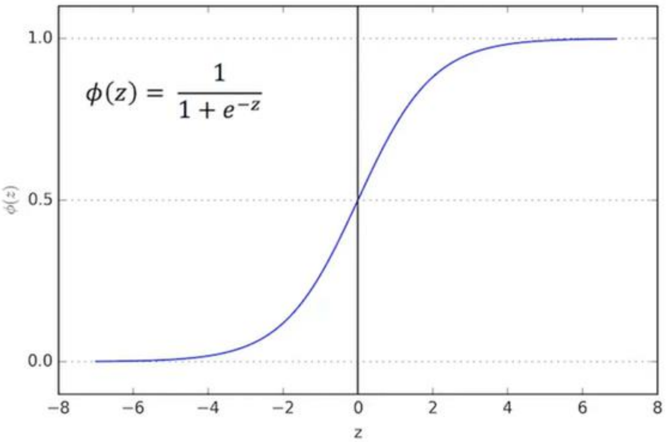
  $Y = b_1X + b_0$

  Y = dependent variable (returns a 0 or a 1)
  - X = independent variable
  - b_0 = constant
  - b_1 = slope or steepness of the curve



Logistic Regression

- Effectively we want to transform our linear regression to a logistic function, one that returns either a 0 or a 1.
- We use the Sigmoid Function for this purpose.
- The Sigmoid or Logistic Function takes in any value and outputs it to be between 0 and 1.



Logistic Regression - evaluating

- How reliable is our prediction?
- Use confusion matrix to evaluate classification models

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Confusion Matrix

- Testing for the presence of a disease
- Terms:
  - True Positives (TP)
  - True Negatives (TN)
  - False Positives (FP) – Type 1 error
  - False Negatives (FN) – Type 2 error

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

- Accuracy: Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate**: Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"

Ref: www.dataschool.io/simple-guide-to-confusion-matrix-terminology

Confusion Matrix

- **True Positive Rate** (Recall/Sensitivity)
  - When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

- Precision:
  - When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/110 = 0.91

- **True Negative Rate** (Specificity)
  - When it's actually no, how often does it predict no?
  - TN/actual no = 50/60 = 0.83

Ref: www.dataschool.io/simple-guide-to-confusion-matrix-terminology

Confusion Matrix

- F1 Score
  - To only optimise recall, our algorithm will predict most examples to belong to the positive class, resulting in many false positives and, hence, low precision.
  - If we try to optimise precision, our model will predict very few examples as positive results, but recall will be very low.
  - To take both precision and recall into account, we can use F1 Score

$$\text{F1 score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Ref: www.dataschool.io/simple-guide-to-confusion-matrix-terminology

Sample datasets for classification

- https://archive.ics.uci.edu/ml/datasets.php?format=&task=cla&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=list
- https://data.world/datasets/classification
- https://datasetsearch.research.google.com/search?query=classification&docid=L2cvMTFqOWJ3cGhoMQ%3D%3D
- https://www.datasetlist.com/

60 mins

Lunch Break

Lunch break 12:15 - 13:15

Activity 2.4 – Classification with Logistic Regression

ATTRIBUTE	DESCRIPTION	VALUE
Preg	Number of pregnancies	[0 – 17]
Plas	Plasma glucose concentration in an oral glucose tolerance test	[0-199]
Pres	Diastolic blood pressure	[0-122]
Skin	Triceps skin fold thickness	[0-99]
Insu	2-Hour serum insulin	[0-846]
Mass	Body mass index	[0-67]
Pedi	Diabetes pedigree function	[0-2.45]
Age	Age of an individual	[21-81]
class	Tested positive / negative	(0,1)

Exercises:
Is a patient with the following data diabetic?
preg:7, plas: 132, pres: 80,
skin: 30, test: 0, mass: 45.5, pedi: 0.547, age: 45

Target to finish by 13:50

Step 1:
Watch and listen to the instructor's demonstration

15 mins

Step 2:
Work through the activities

Individual Activity

15 mins

Keras Model in Scikit Learn

- Keras models can be used in scikit-learn by wrapping them with the **KerasClassifier** or KerasRegressor class.
- To use these wrappers you must define a function that creates and returns your Keras sequential model, then pass this function to the build_fn argument when constructing the KerasClassifier class.

```
def create_model():
   ...
   return model

model = KerasClassifier(build_fn=create_model)
```

- The constructor for the KerasClassifier class can take default arguments that are passed on to the calls to model.fit(), such as the number of epochs and the batch size.

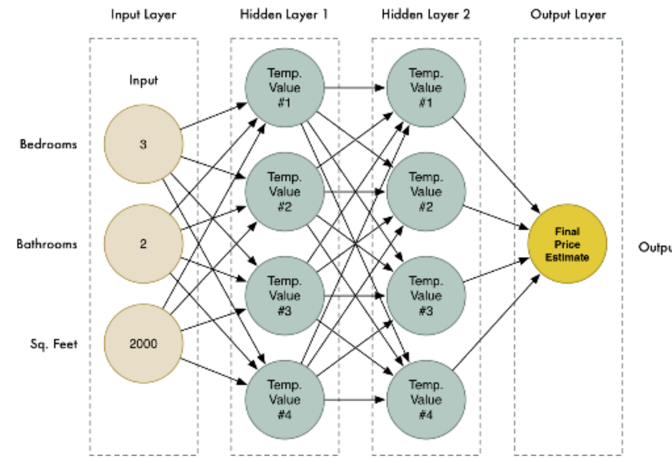
```
def create_model():
   ...
   return model

model = KerasClassifier(build_fn=create_model, epochs=10)
```

Create Model

```
# Function to create model, required for KerasClassifier
def create_model():
  # create model
  model = Sequential()
  model.add(Dense(12, input_dim=8, activation='relu'))
  model.add(Dense(8, activation='relu'))
  model.add(Dense(1, activation='sigmoid'))
  # Compile model
  model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
  model.summary()
  return model

```

Activity 2.5 – Classification with Neural Network

ATTRIBUTE	DESCRIPTION	VALUE
Preg	Number of pregnancies	[0 – 17]
Plas	Plasma glucose concentration in an oral glucose tolerance test	[0-199]
Pres	Diastolic blood pressure	[0-122]
Skin	Triceps skin fold thickness	[0-99]
Insu	2-Hour serum insulin	[0-846]
Mass	Body mass index	[0-67]
Pedi	Diabetes pedigree function	[0-2.45]
Age	Age of an individual	[21-81]
class	Tested positive / negative	(0,1)



Exercises:
Modify the hidden layers with additional nodes. Did the performance improve?

Target to finish by 14:15

Step 1:
Watch and listen to the instructor's demonstration

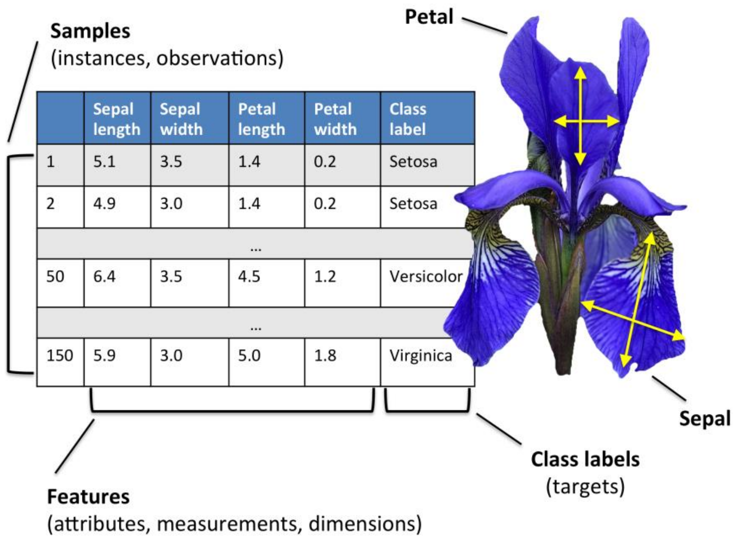
15 mins

Step 2:
Work through the activities

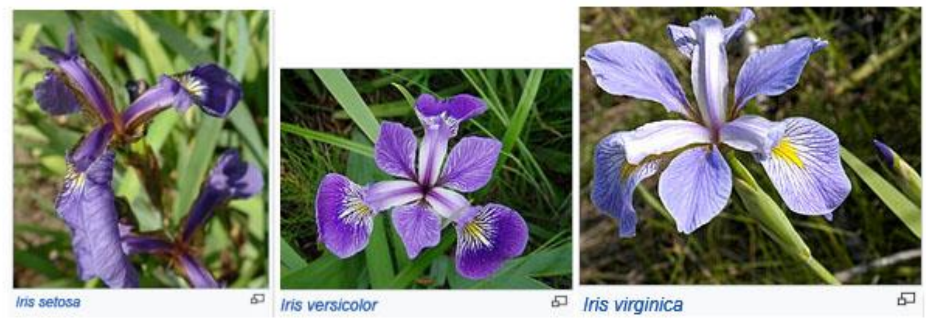
Individual Activity

15 mins

Activity 2.6 – Multi Class Classification



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Exercises:
Modify the hidden layers with additional nodes.
Did the performance improve?

Step 1:
Watch and listen to the instructor's demonstration

10 mins

Target to finish by X:XX

Step 2:
- Do on your own

30 mins

Individual Activity

Activity 2.7 – Classifier Use Case

Dataset:
https://www.kaggle.com/c/titanic/data

Exercises:
Use a different classifier and compare the results

Keras pandas
scikit learn
TensorFlow

Step 1:
Watch and listen to the instructor's demonstration

10 mins

Target to finish by 14:55

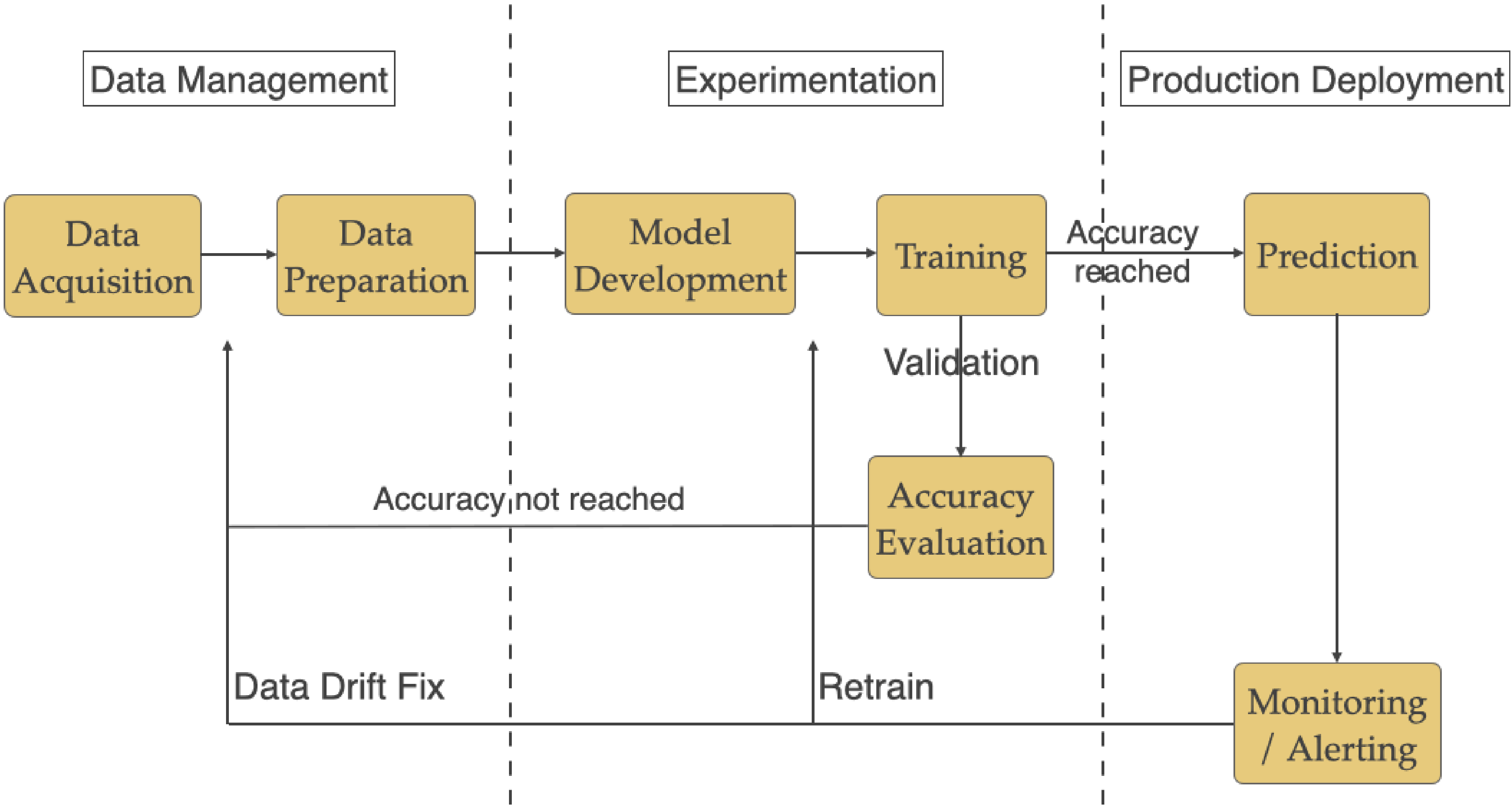
Step 2:
- Do on your own

30 mins

Individual Activity

15 Mins Break

Machine Learning workflow



Model Improvement

What are hyperparameters?

- In machine/deep learning, a hyperparameter is a parameter whose value is used to control the learning process
- Hyperparameters can be classified as
  - **model hyperparameters** - cannot be inferred while fitting the machine to the training set because they refer to the model selection task. E.g. topology and size of a neural network
  - **algorithm hyperparameters** - in principle have no influence on the performance of the model but affect the speed and quality of the learning process

Number of Neurons in the Hidden Layer

Batch Size
Epochs
Optimization Algorithm
Learning Rate and Momentum
Network Weight Initialization
Neuron Activation Function
Dropout Regularization

Keras Model in Scikit Learn

- Keras models can be used in scikit-learn by wrapping them with the **KerasClassifier** or KerasRegressor class.
- To use these wrappers you must define a function that creates and returns your Keras sequential model, then pass this function to the build_fn argument when constructing the KerasClassifier class.

```
def create_model():
    ...
    return model

model = KerasClassifier(build_fn=create_model)
```

- The constructor for the KerasClassifier class can take default arguments that are passed on to the calls to model.fit(), such as the number of epochs and the batch size.

```
def create_model():
    ...
    return model

model = KerasClassifier(build_fn=create_model, epochs=10)
```

How to Use Grid Search in scikit-learn

- Grid search is a model hyperparameter optimization technique.
- In scikit-learn this technique is provided in the GridSearchCV class.
- When constructing this class you must provide a dictionary of hyperparameters to evaluate in the param_grid argument. This is a map of the model parameter name and an array of values to try.
- By default, accuracy is the score that is optimized, but other scores can be specified in the score argument of the GridSearchCV constructor.
- By default, the grid search will only use one thread. By setting the n_jobs argument in the GridSearchCV constructor to -1, the process will use all cores on your machine. Depending on your Keras backend, this may interfere with the main neural network training process.

How to Use Grid Search in scikit-learn

- The **GridSearchCV** process will then construct and evaluate one model for each combination of parameters.
- Cross validation is used to evaluate each individual model and the default of 3-fold cross validation is used, although this can be overridden by specifying the **cv** argument to the **GridSearchCV** constructor.
- an example of defining a simple grid search:

```
param_grid = dict(epochs=[10,20,30])
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X, Y)
```

- Once completed, you can access the outcome of the grid search in the result object returned from grid.fit().
- The best_score_ member provides access to the best score observed during the optimization procedure and the best_params_ describes the combination of parameters that achieved the best results.
- http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html#sklearn.grid_search.GridSearchCV

How to Tune Batch Size and Number of Epochs

- The batch size in iterative gradient descent is the number of patterns shown to the network before the weights are updated. It is also an optimization in the training of the network, defining how many patterns to read at a time and keep in memory.
- The number of epochs is the number of times that the entire training dataset is shown to the network during training. Some networks are sensitive to the batch size, such as recurrent neural networks and Convolutional Neural Networks.

```
# create model
model = KerasClassifier(build_fn=create_model, verbose=1)
# define the grid search parameters
batch_size = [10, 20, 40, 60, 80, 100]
epochs = [10, 50, 100]
param_grid = dict(batch_size=batch_size, epochs=epochs)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X, Y,verbose=1)
```

How to Tune Network Weight Initialization

- Neural network weight initialization used to be simple: use small random values.
- Now there is a suite of different techniques to choose from. Keras provides a laundry list http://keras.io/initializations/
- In this example, we will look at tuning the selection of network weight initialization by evaluating all of the available techniques.
- We will use the same weight initialization method on each layer. Ideally, it may be better to use different weight initialization schemes according to the activation function used on each layer. In the example below we use rectifier for the hidden layer. We use sigmoid for the output layer because the predictions are binary

How to Tune Network Weight Initialization

```
# Function to create model, required for KerasClassifier
def create_model(init_mode='uniform'):
    # create model
    model = Sequential()
    model.add(Dense(12, input_dim=8, kernel_initializer=init_mode, activation='relu'))
    model.add(Dense(1, kernel_initializer=init_mode, activation='sigmoid'))
    # Compile model
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

```
# create model
model = KerasClassifier(build_fn=create_model, epochs=100, batch_size=10, verbose=1)

# define the grid search parameters
init_mode = ['uniform', 'lecun_uniform', 'normal', 'zero', 'glorot_normal', 'glorot_uniform', 'he_normal', 'he_uniform']

param_grid = dict(init_mode=init_mode)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X, Y, verbose=1)

```

How to Tune the Training Optimization Algorithm

- Keras offers a suite of different state-of-the-art optimization algorithms.
- Most of the time you will choose one training optimization algorithm from some prior experience and instead focus on tuning its parameters on your problem
- suite of optimization algorithms supported by the Keras API:
  - http://keras.io/optimizers/
    - SGD
    - RMSprop
    - Adam
    - Adadelta
    - Adagrad
    - Adamax
    - Nadam
    - Ftrl

How to Tune the Training Optimization Algorithm

```
# Function to create model, required for KerasClassifier
def create_model(optimizer='adam'):
    # create model
    model = Sequential()
    model.add(Dense(12, input_dim=8, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    # Compile model
    model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model
```

```
# create model
model = KerasClassifier(build_fn=create_model, epochs=100, batch_size=10, verbose=1)

# define the grid search parameters
optimizer = ['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']

param_grid = dict(optimizer=optimizer)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X, Y, verbose=1)
```

Activity 2.8 – Hyperparameter tuning for Linear Regression

sq feet	num bedrooms	num bathrooms	sale price
785	2	2	170461
1477	2	2	271651
712	1	1	139912

Bedrooms	Bathrooms	Sq. Feet	Sale Price
3	2	2000	???

Exercise:

Step 1:
Watch and listen to the instructor's demonstration

15 mins

Step 2:
Work through the activities

Individual Activity

15 mins

Activity 2.9 – Hyperparameter tuning for Classifier

ATTRIBUTE	DESCRIPTION	VALUE
Preg	Number of pregnancies	[0 – 17]
Plas	Plasma glucose concentration in an oral glucose tolerance test	[0-199]
Pres	Diastolic blood pressure	[0-122]
Skin	Triceps skin fold thickness	[0-99]
Insu	2-Hour serum insulin	[0-846]
Mass	Body mass index	[0-67]
Pedi	Diabetes pedigree function	[0-2.45]
Age	Age of an individual	[21-81]
class	Tested positive / negative	(0,1)

Step 1:
Watch and listen to the instructor's demonstration

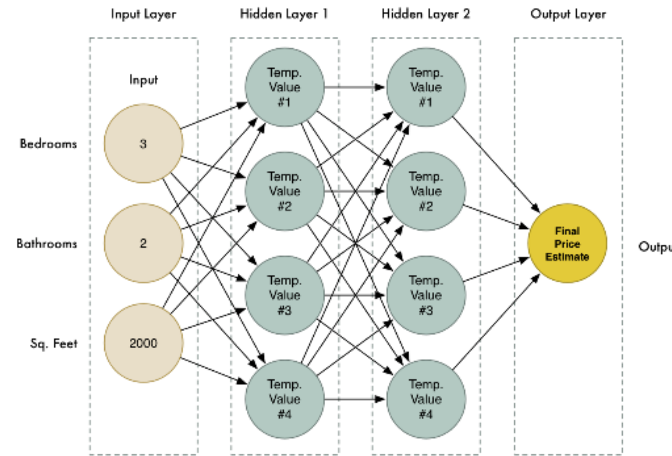
15 mins

Step 2:
Work through the activities

Optional

15 mins

Activity 2.10 – Hyperparameter tuning for Neural Network



Target to finish by 15:45

Step 1:
Watch and listen to the instructor's demonstration

15 mins

Step 2:
Work through the activities

Optional

15 mins

Preparing for Part 2

Create successful AI use cases and projects

01 *Save time → Process automation*

02 *Reduce cost → Predicting the future*

03 *Improve people's lives → at work or at home*

04 *Increase revenue → with predictive marketing*

Where can I start using AI/ML?

✓ Tasks that a typical staff is able to do with "less than one second of thought"

- Examining security video to detect suspicious behaviours
- Finding and eliminating abusive online posts

✓ Not all automation problems require learning.

- Automation without learning is appropriate when the problem is relatively straightforward

✓ Good business problems for ML means any problem that:

- (1) require prediction rather than causal inference; and
- (2) are sufficiently self-contained, or relatively insulated from outside influences.

✓ Task that can afford for some allowance of error

https://hbr.org/2016/11/how-to-tell-if-machine-learning-can-solve-your-business-problem

How Do I decide what is applicable?

- Write down what you do in your job and break apart your activities into:
  - things you do daily or regularly versus things you do sporadically;
  - things that have become second nature versus things that require patient deliberation or lots of thought; and
  - things that are part of a process versus things you do on your own.

- For tasks that you perform regularly, on your own, and that feel automatic . . .
  - identify how many others in your organization do similar tasks and how many people have done this historically.

https://hbr.org/2017/10/how-to-spot-a-machine-learning-opportunity-even-if-you-arent-a-data-scientist

How Do I Decide?

- ✓ Examine the nature of the task.
  - Does it include predicting something or bucketing something into categories?

- ✓ Ask yourself - if 10 colleagues in your organization performed the task, would they all agree on the answer?
  - If humans can’t agree something is true or false, computers can’t reliably transform judgment calls into statistical patterns.

- ✓ How long have people in the organization been doing something similar to this task?
  - If it’s been a long time, has the organization kept a record of successfully completed tasks?
  - If yes, this could be used as a training data set for your supervised learning algorithm.
    - If there is access to clean data, that is helpful as a start
  - If no, you may need to start collecting the data today, and then you can keep a human in the loop to train the algorithm over time.

How Do I Decide?

- Sit down with a data science team and tell them about the task.
  - Walk them through your thought process and tell them what aspects of information you focus on when you complete your task.
  - This will help them determine/confirm if automation is feasible and tease out the aspects of the data that will be most predictive of the desired output.

- Ask yourself, if this were automated :
  - How might that change the product or service you offer to your customers?
  - What is the worst thing that could happen to the business if this were to be automated?
  - What is the worst thing that could happen to the business if the algorithm outputs the wrong answer or an answer with a 65% or 70% accuracy rate?
  - What is the accuracy threshold the business requires to go ahead and automate this task?

Best Practices – Starting out

- Identify the low-hanging fruit
  - Or what is the question you're dying to find the answer for but can't figure out with existing methods (priority problem still and not a toy problem)
- Start supervised learning with a wealth of '<u>historic</u>' but relevant data
  - Don't need to wait to collect months of data before deriving value from AI/ML
  - Choose existing data that is related to a problem so that one can drive ROI with purpose
- Start with clean data, not big data
- Use an available cloud system (Amazon, Google, Microsoft, etc)
  - Use pre-packaged systems if possible versus starting "from-scratch"
- Remember that what AI can do and how it fits into your strategy is the beginning, not the end

Pitfalls to avoid

DON'TS	DO'S
Expect AI to solve everything	Be realistic about what AI can and cannot do given limitations of technology, data and engineering resources.
Hire 2-3 ML engineers and count solely on them to come up with use cases.	Pair engineering talent with business talent and work cross-functionally to find feasible and valuable projects
Expect the AI project to work the first time.	Plan for AI development to be an iterative process, with multiple attempts needed to succeed
Expect traditional planning processes to apply without changes.	Work with AI team to establish timeline estimates, milestones, KPIs, etc
Think you need superstar AI Engineers before you can do anything.	Keep building the team, but get going with the team you have.

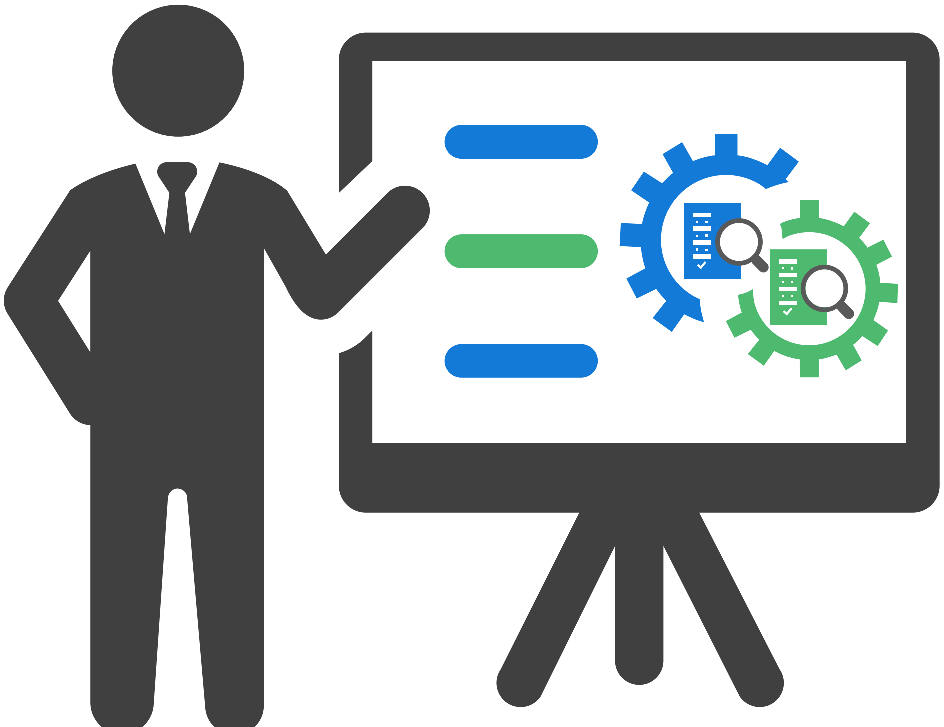
Team Formation

- 2 to a team
- Domain:
  - Structured Data (Regression or Classification)
- To complete proposal template before phase 2 (24-25 Jan 2022)
  - Problem statement
  - Dataset

Quiz

https://bit.ly/3sOo2mF

Q&A



Email
seow_khee_wei@rp.edu.sg
shubham_khare@rp.edu.sg

Telegram
@kwseow
@ricky_sk

Source code:

Thank you

Appendix A – Data Type / Range

Type	Size(Bytes)	Range	Format specifier
short int	2	-32,768 to +32,767	%h
unsigned short int	2	0 to +65,535	%u
unsigned int	4	0 to +4,294,967,295	%u
int	4	-2,147,483,648 to +2,147,483,647	%d
long int	4	-2,147,483,648 to +2,147,483,647	%ld
signed char	1	-128 to +127	%c
unsigned char	1	0 to +255	%c
float	4	3.4 e -38 to 3.4 e 38	%f, %e, %g
double	8	1.7 e -308 to 1.7 e 308	%lf
long double	12	3.4 e -4932 to 3.4 e 4932	%lf

Appendix B – Other Useful Resources

- **Graphviz – rendering dot file to png file**
  - **Install graphviz -** https://graphviz.gitlab.io/download/
  - **Set up system Path to the installed bin directory**
  - **In command or anaconda prompt:**
    - **d:\> dot –Tpng <inputfile.png> -o <outputfile.png>**

- **Reading excel using Pandas**
  - **pd.read_excel()** https://datatofish.com/read_excel/

- **Pandas addition**
  - https://www.geeksforgeeks.org/python-pandas-dataframe-append/
  - https://www.geeksforgeeks.org/adding-new-column-to-existing-dataframe-in-pandas/

- **Pandas dataframe slicing and dicing**
  - https://datacarpentry.org/python-ecology-lesson/03-index-slice-subset/index.html

- **Seaborn pairplot data analysis – GapMinder socioeconomic data**
  - https://github.com/WillKoehrsen/Data-Analysis/blob/master/pairplots/Pair%20Plots.ipynb
  - https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166

Appendix C – Installing Anaconda & Python

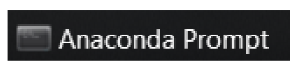
Installing the Anaconda distribution and Python 3

A. Installing on Windows / macOS / Linux

To install on Windows, follow the steps given as follows:

1. First, download the executable from https://www.anaconda.com/distribution/
2. Click on the operating systems needed - 'Windows' or 'macOS' or 'Linux'
3. Download the python 3.7 version
4. 2. Then, launch the Anaconda prompt when completed - this can be found using search from the Windows Start menu

Anaconda prompt is a Windows command prompt with all the environment variables set to point to Anaconda. You are now ready to use your base Python environment.



B. Installing the libraries and packages

1. Launch the Anaconda Prompt and use the 'pip' command.

2. To install jupyter lab library, at the prompt, type: c:\> pip install jupyterlab

3. To install multiple libraries like numpy, panda, matplotlib, at the prompt, type: c:\> pip install numpy panda matplotlib

C. Further References: