Oleksandr Romanko, Ph.D.
Associate Director, Financial Risk Quantitative Research, SS&C Algorithmics
Adjunct Professor, University of Toronto

MIE1624H – Introduction to Data Science and Analytics

Lecture 7 – Machine Learning

University of Toronto
October 31, 2023

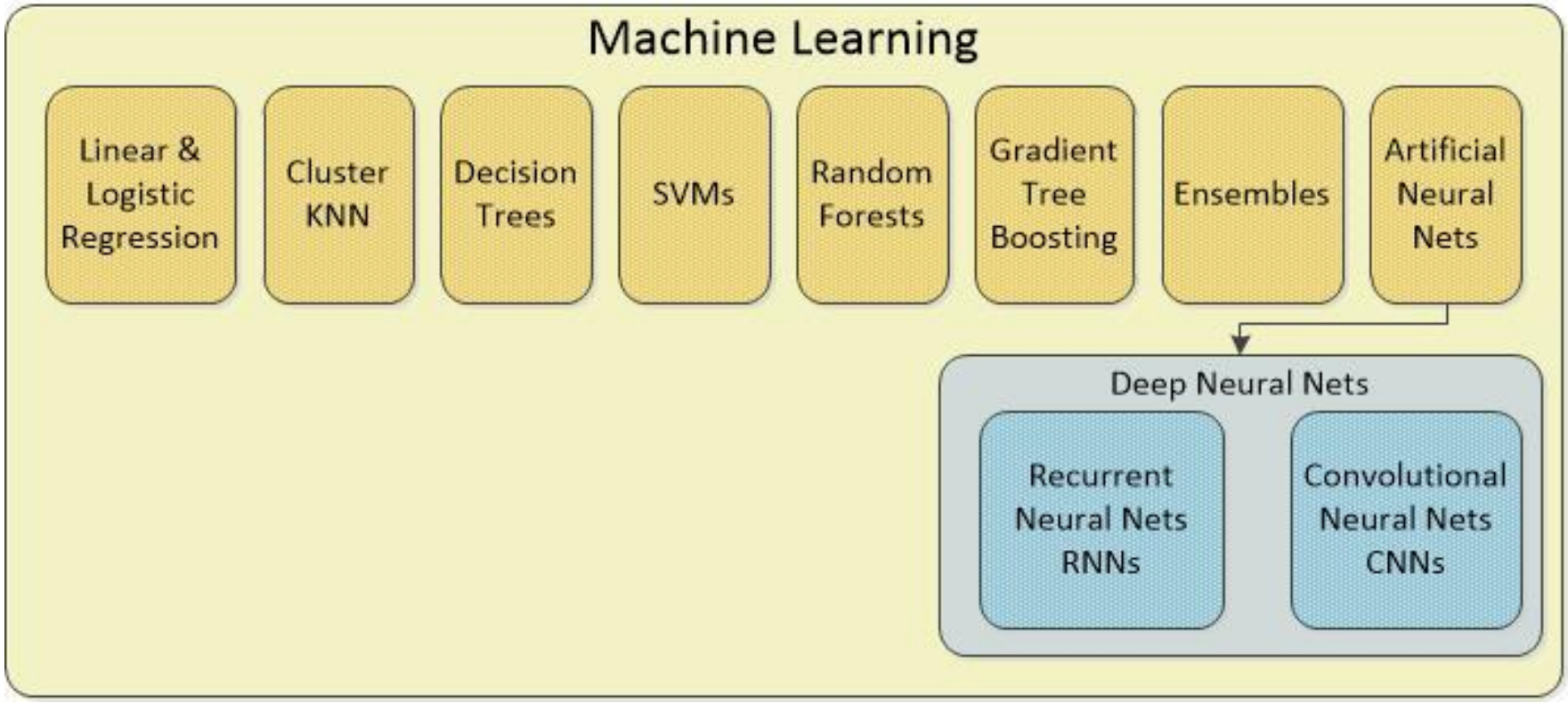
Machine learning

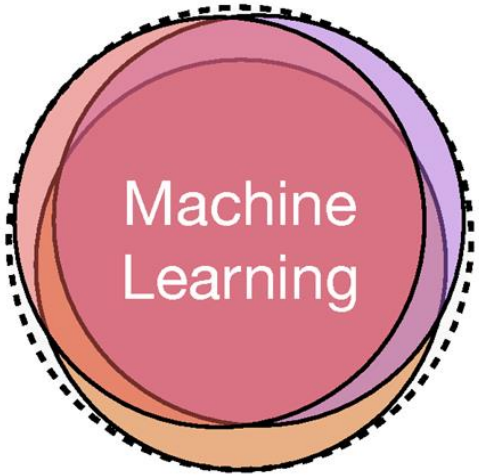
Machine learning gives computers the ability to learn without being explicitly programmed

- **Supervised learning**: decision trees, ensembles (bagging, boosting, random forests), k-NN, linear regression, Naive Bayes, neural networks, logistic regression, SVM
  - Classification
  - Regression (prediction)
- **Unsupervised learning**: k-means, c-means, hierarchical clustering, DBSCAN
  - Clustering
  - Dimensionality reduction (PCA, LDA, factor analysis, t-SNE)
  - Association rules (market basked analysis)
- **Reinforcement learning**
  - Dynamic programming
- **Neural nets**: deep learning, multilayer perceptron, recurrent neural network (RNN), convolutional neural network (CNN), generative adversarial network (GAN)

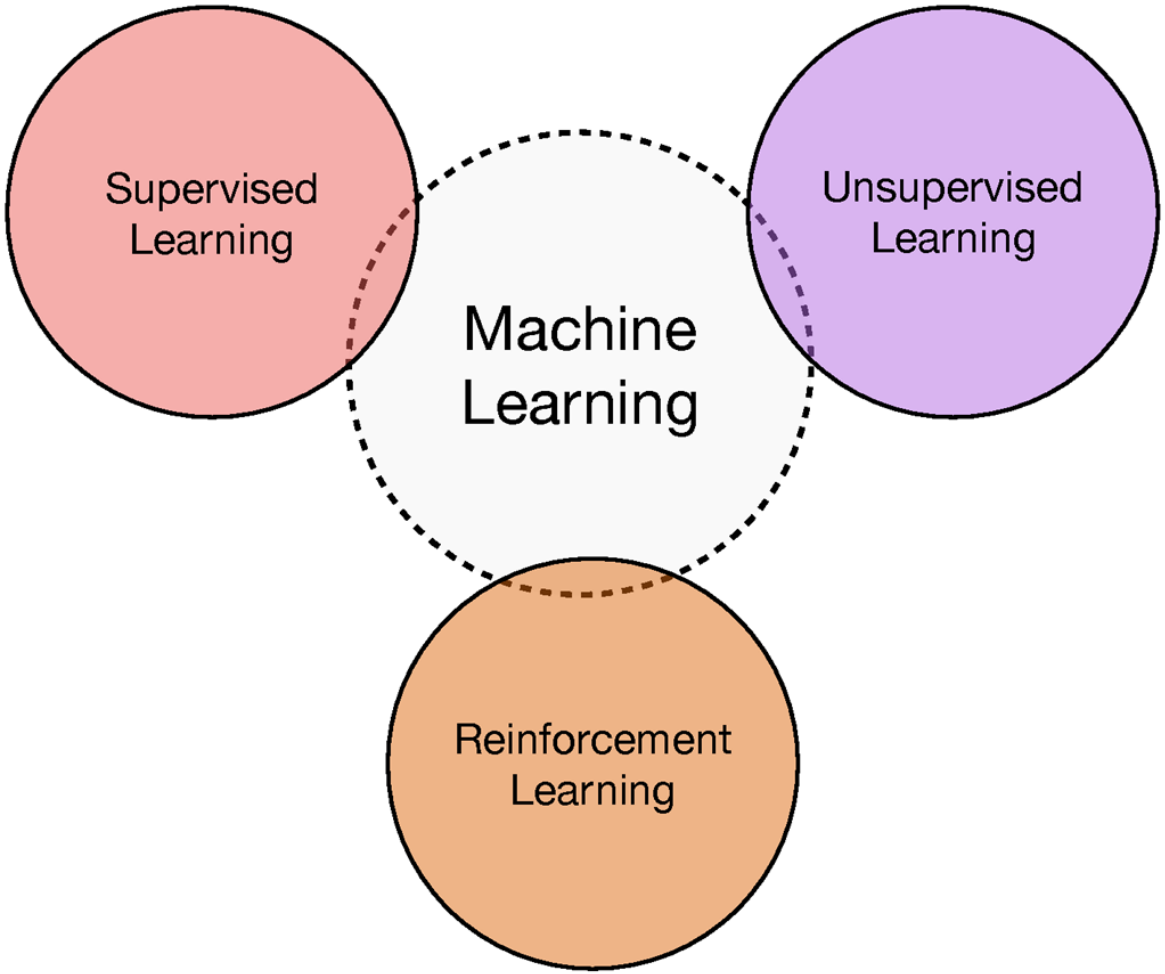
Machine learning

Machine learning gives computers the ability to learn without being explicitly programmed

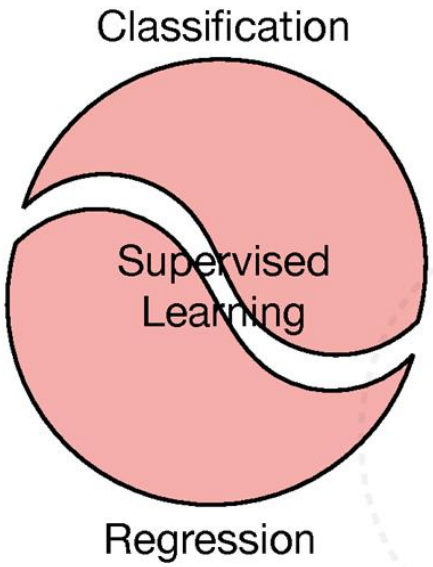




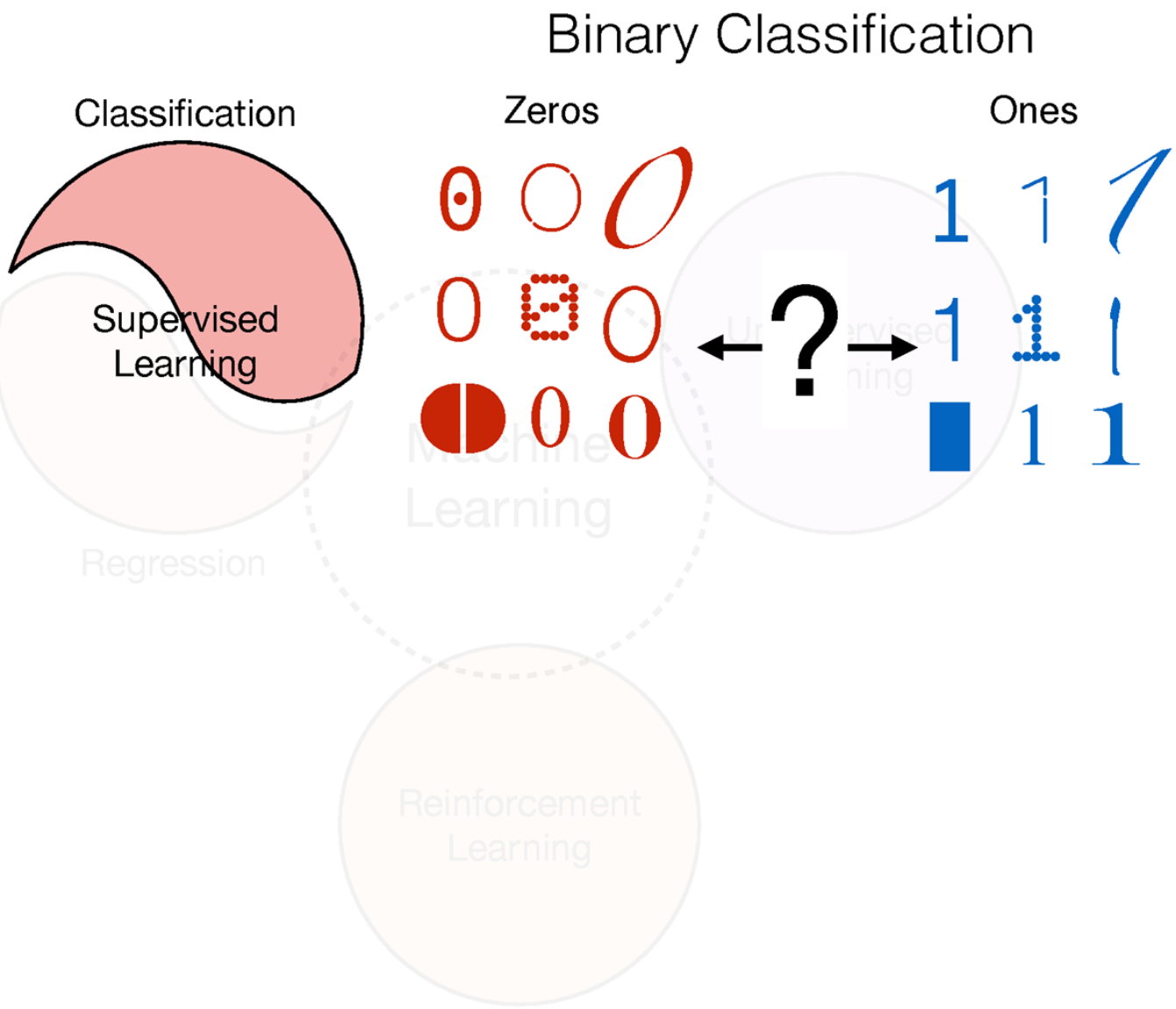
Source: Dima Fishman, Intro to Machine Learning



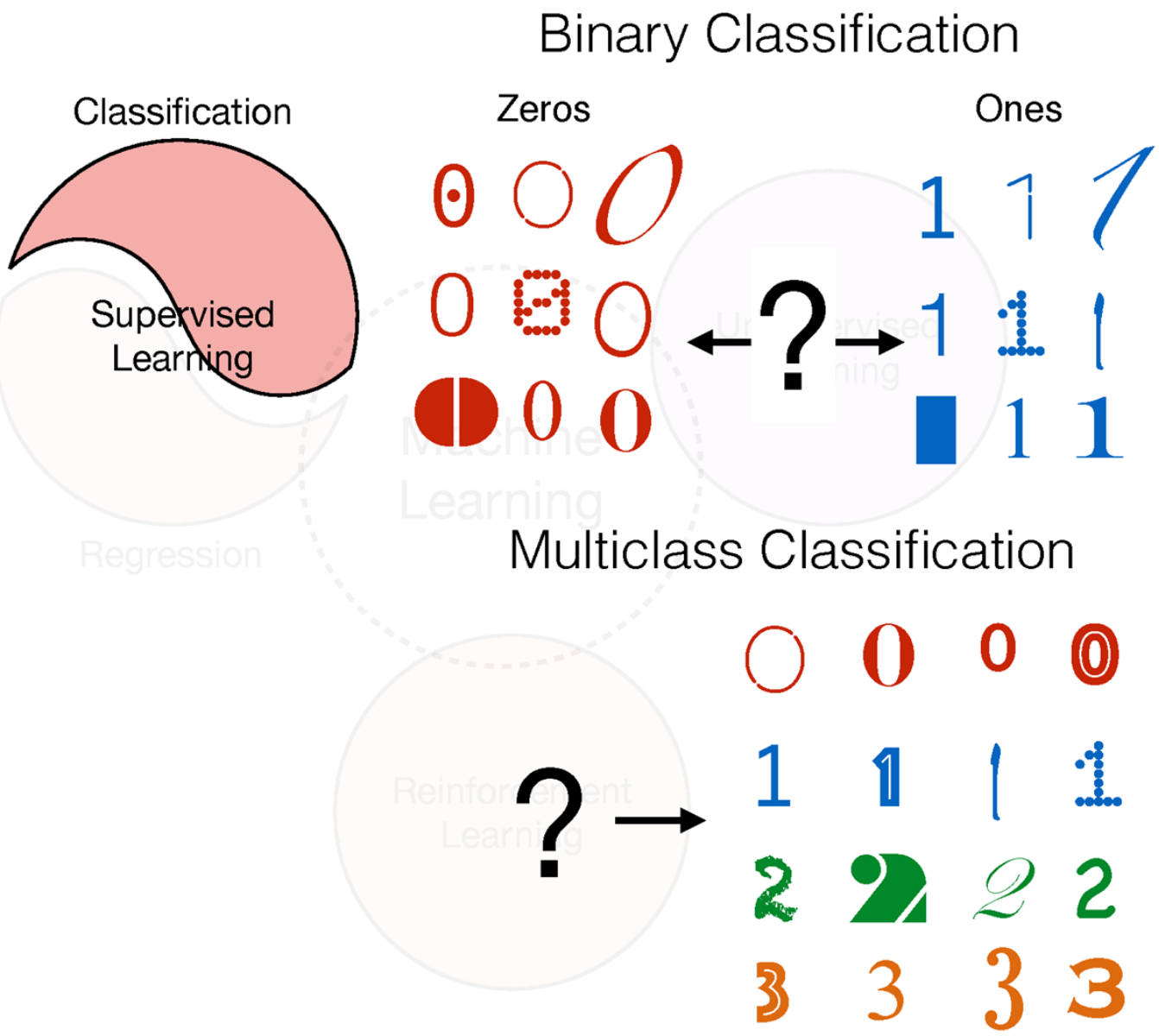
Source: Dima Fishman, Intro to Machine Learning



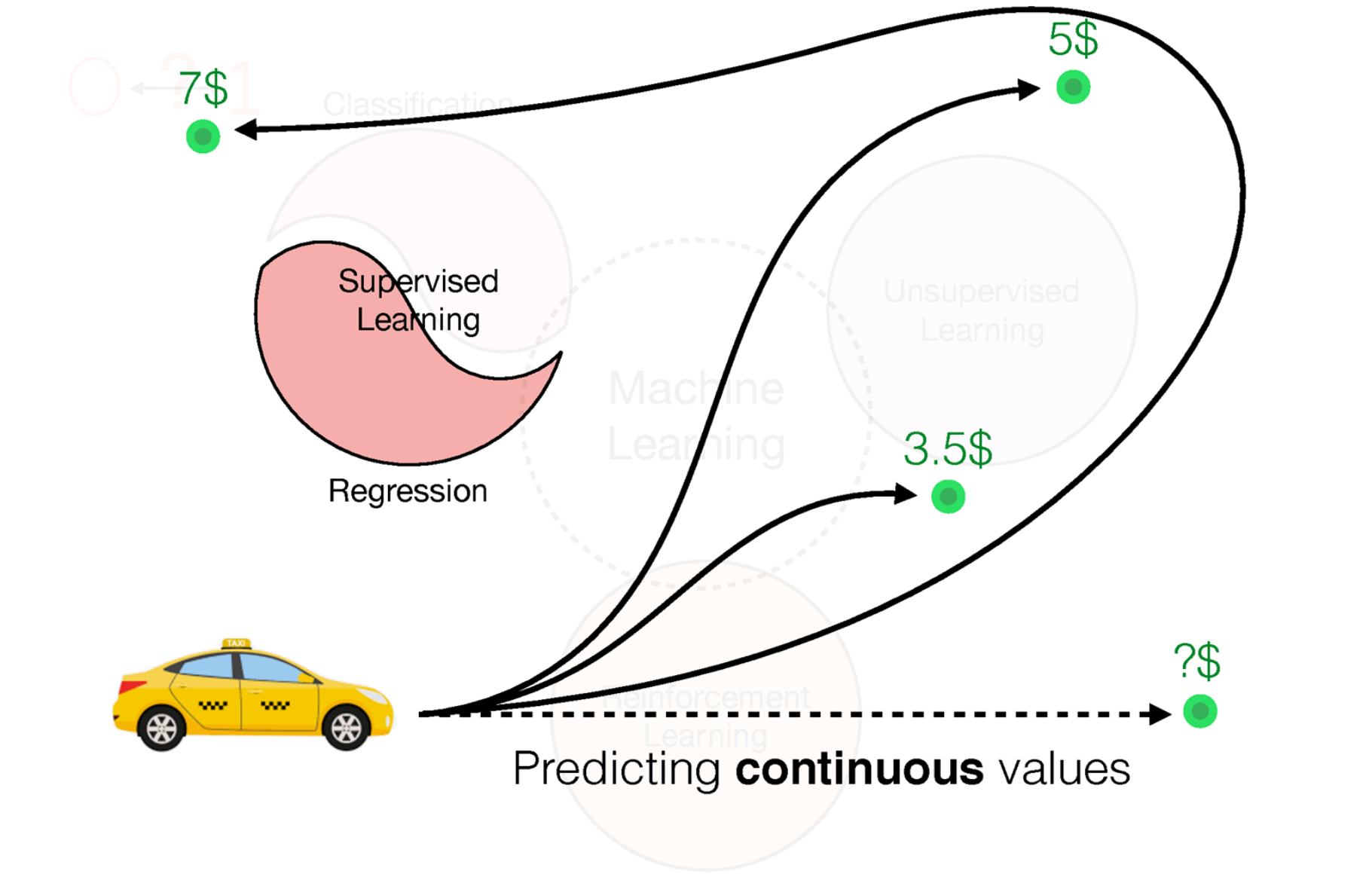
Source: Dima Fishman, Intro to Machine Learning



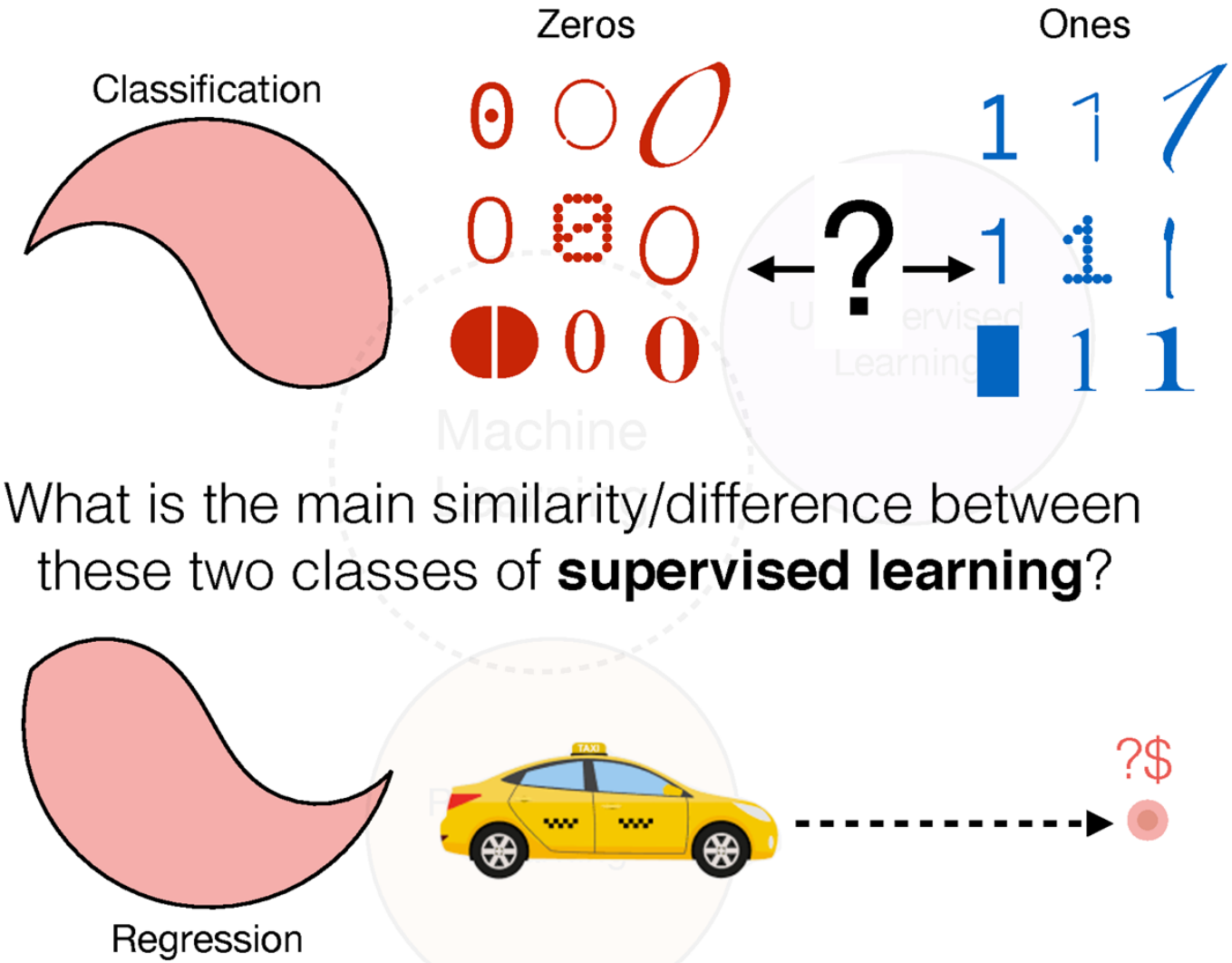
Source: Dima Fishman, Intro to Machine Learning



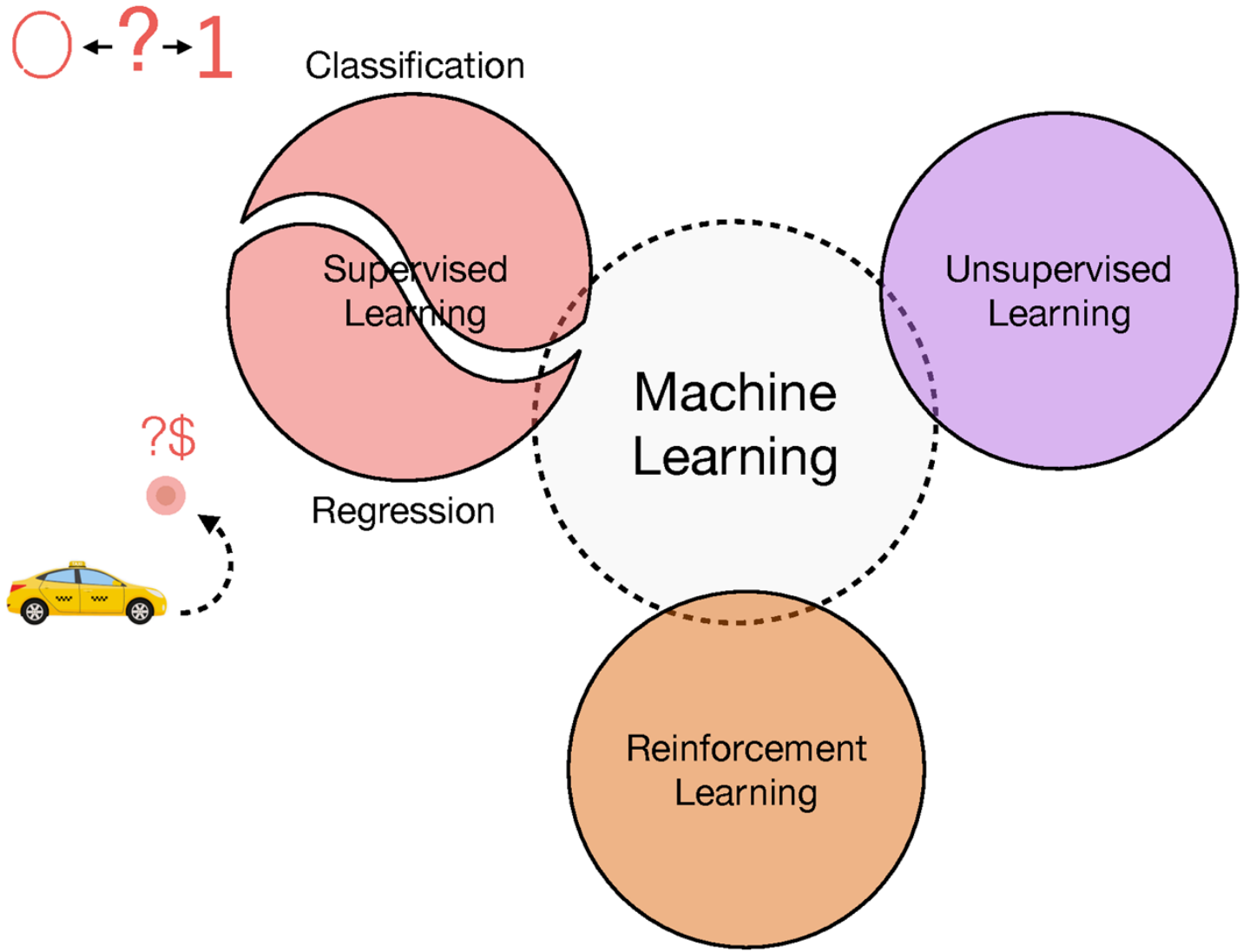
Source: Dima Fishman, Intro to Machine Learning



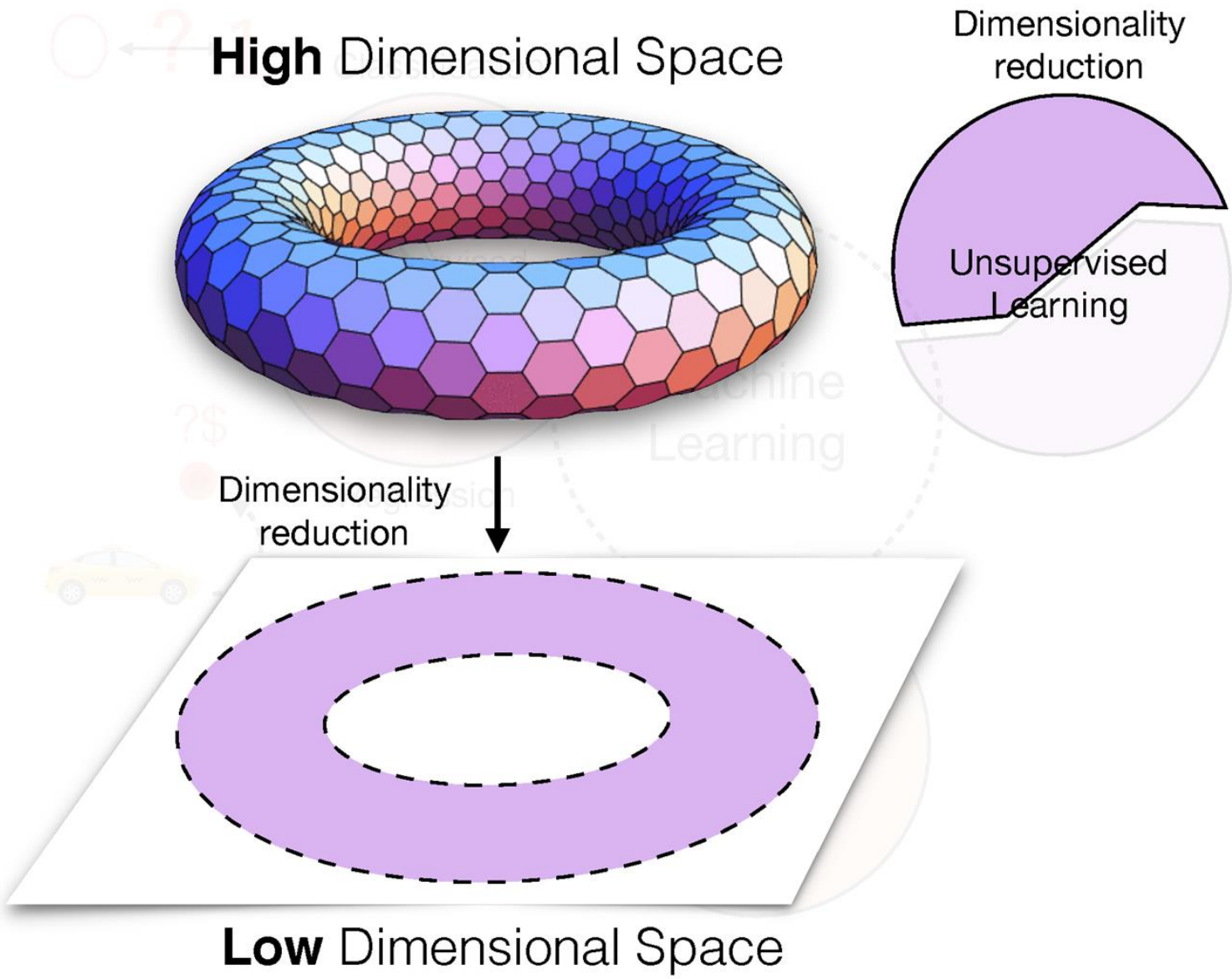
Source: Dima Fishman, Intro to Machine Learning



Source: Dima Fishman, Intro to Machine Learning

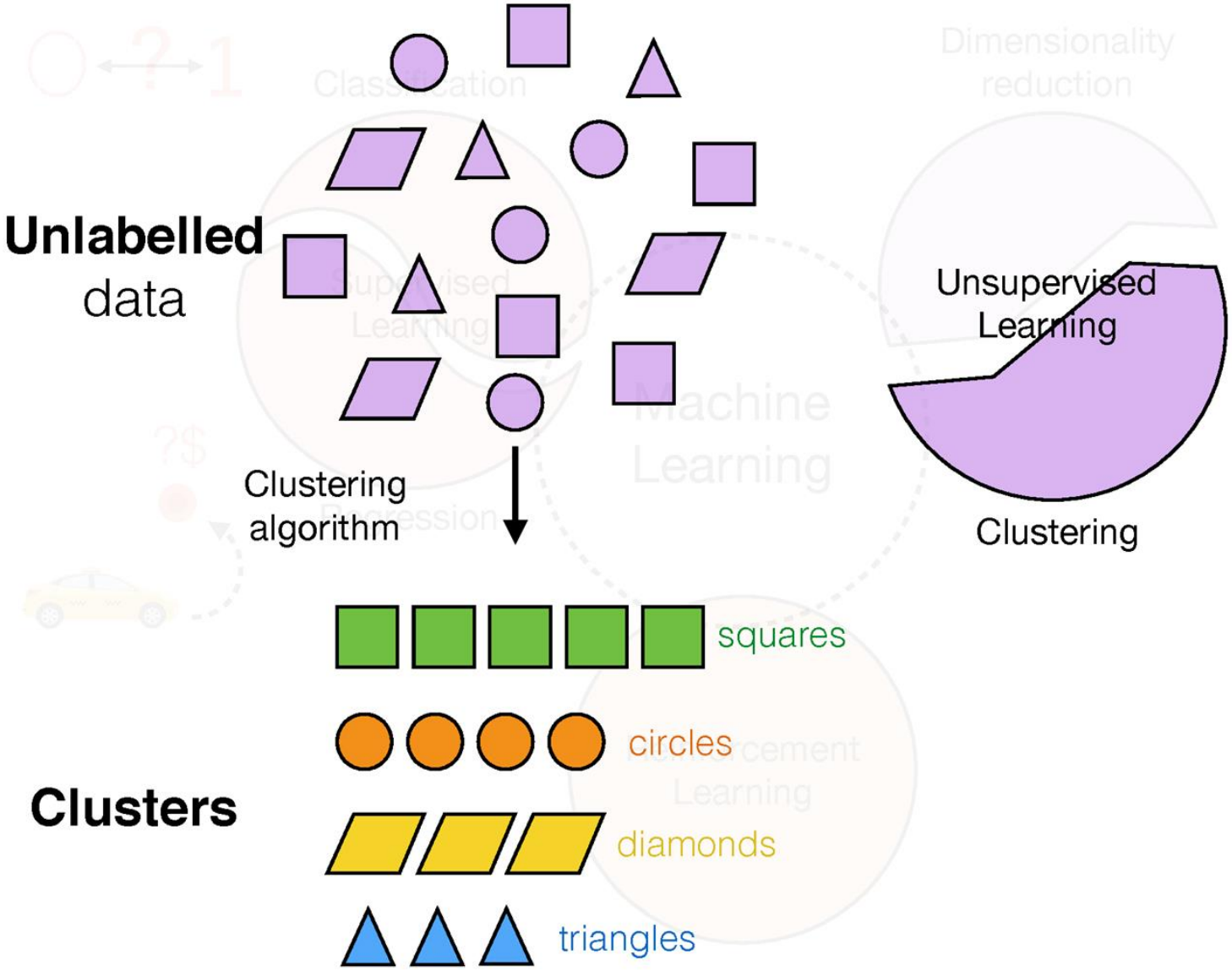


Source: Dima Fishman, Intro to Machine Learning

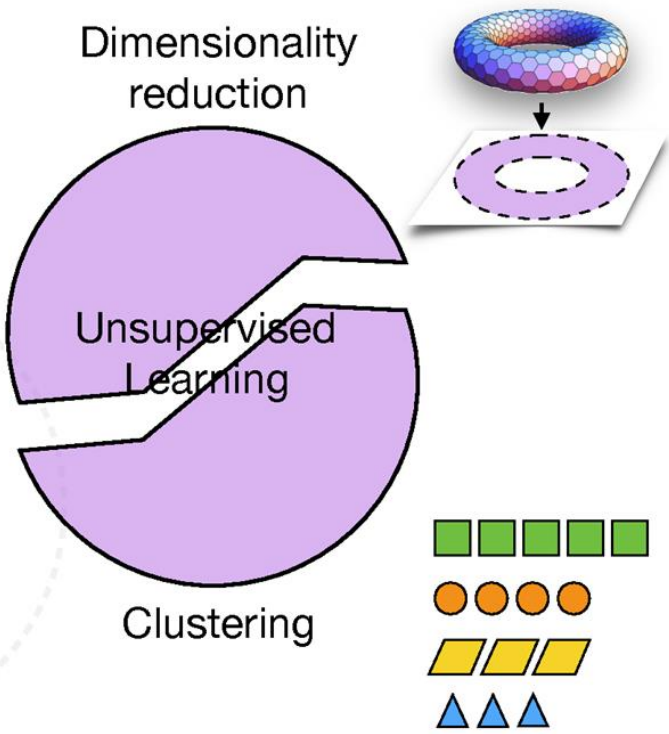


Source: Dima Fishman, Intro to Machine Learning

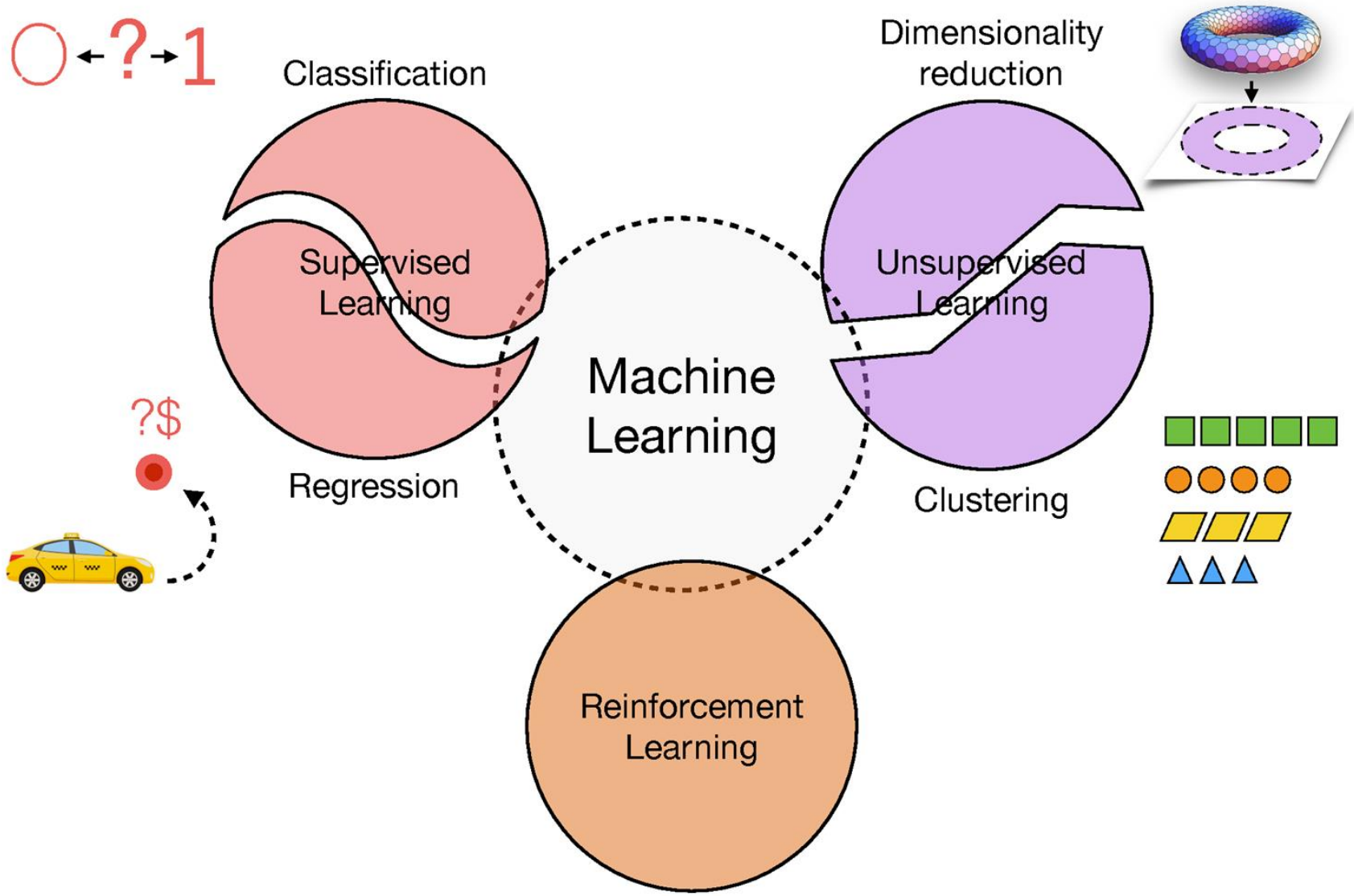
Torus image credit: https://mathematica.stackexchange.com/questions/39879/create-a-torus-with-a-hexagonal-mesh-for-3d-printing



Source: Dima Fishman, Intro to Machine Learning



Source: Dima Fishman, Intro to Machine Learning



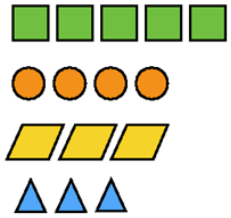
Source: Dima Fishman, Intro to Machine Learning

Classification

Supervised Learning

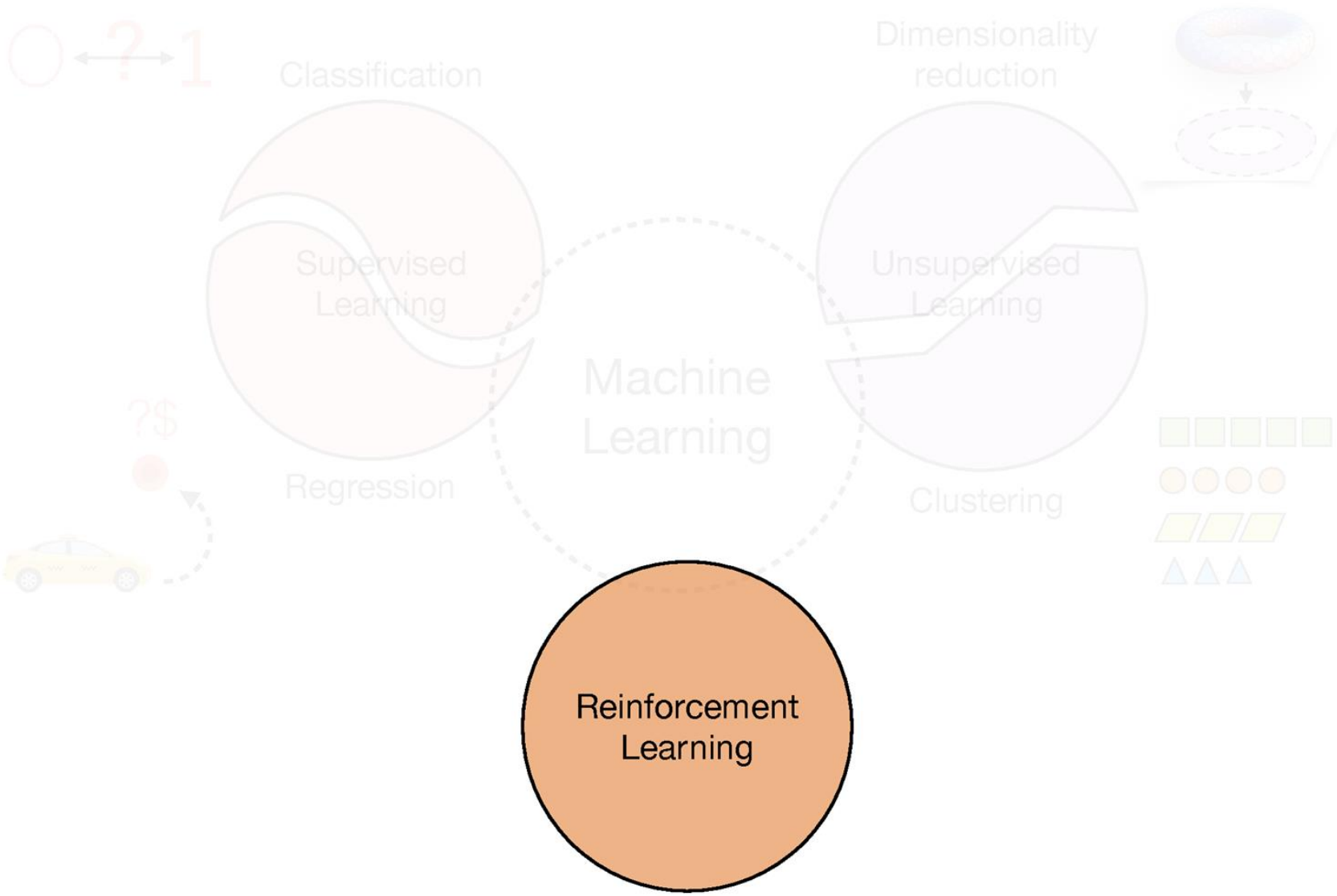
Unsupervised Learning

Clustering

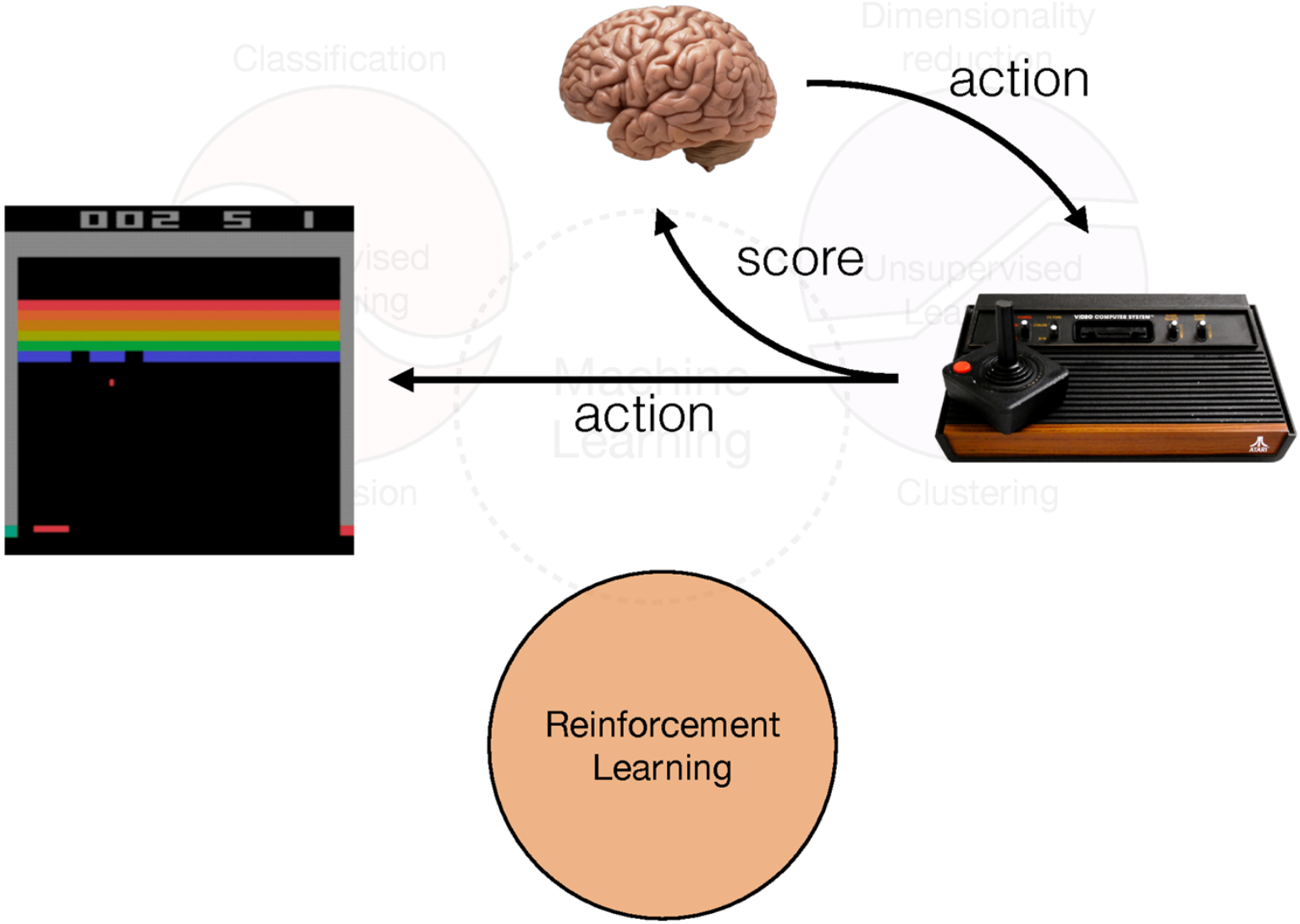


What is the main similarity/difference between **classification** and **clustering**?

Source: Dima Fishman, Intro to Machine Learning

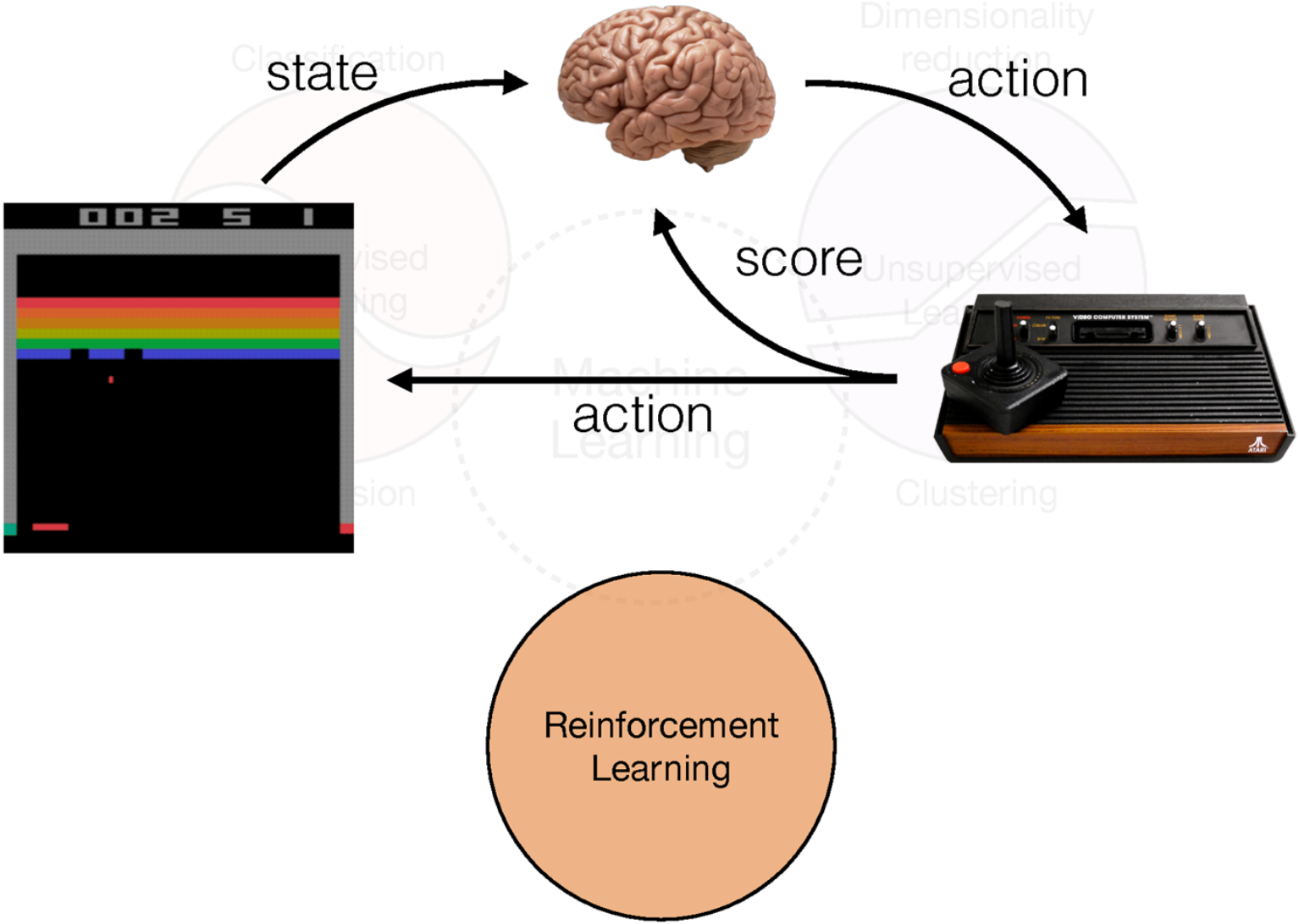


Source: Dima Fishman, Intro to Machine Learning



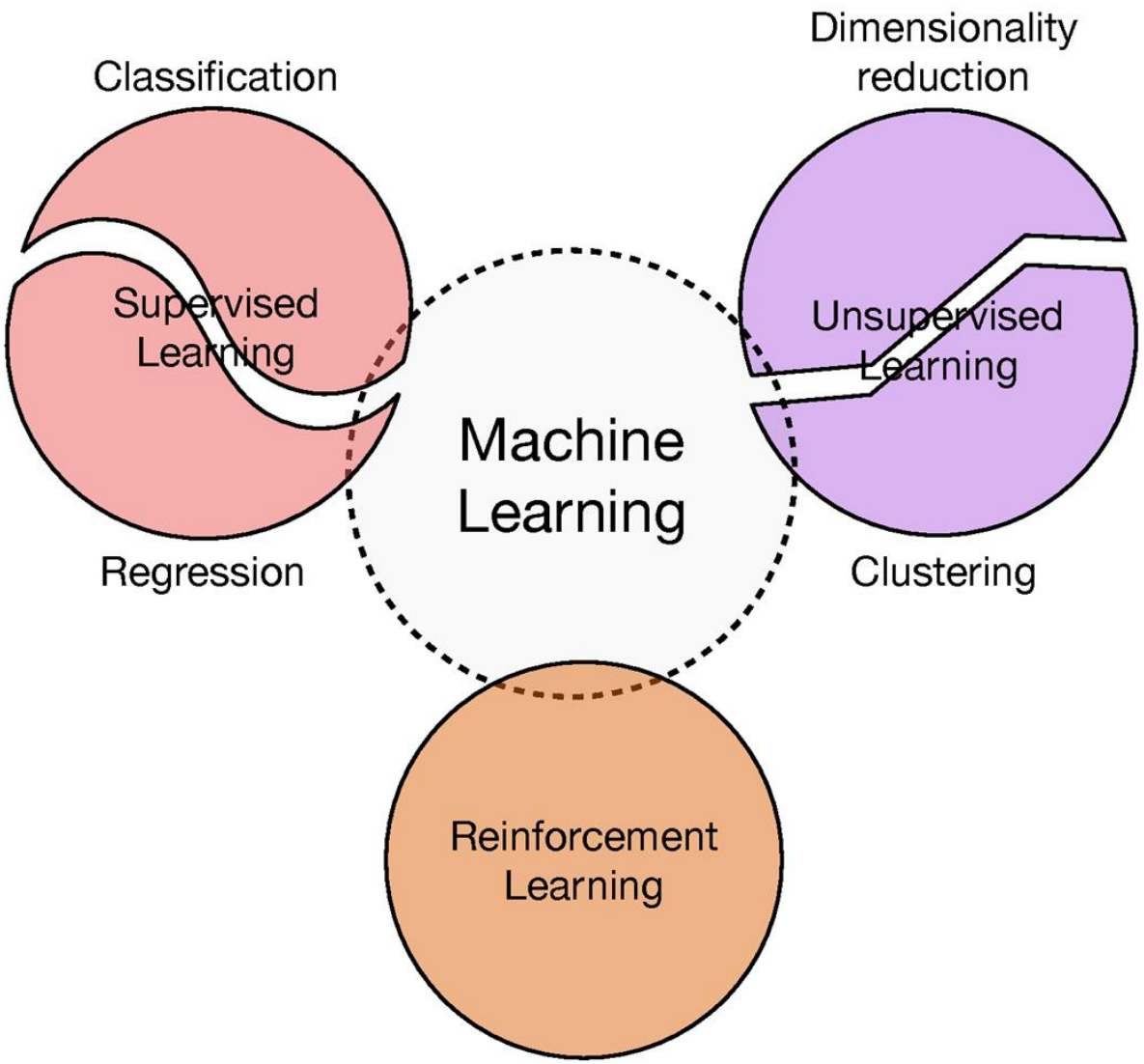
Inspired by https://keon.io/deep-q-learning/

Source: Dima Fishman, Intro to Machine Learning



Inspired by https://keon.io/deep-q-learning/

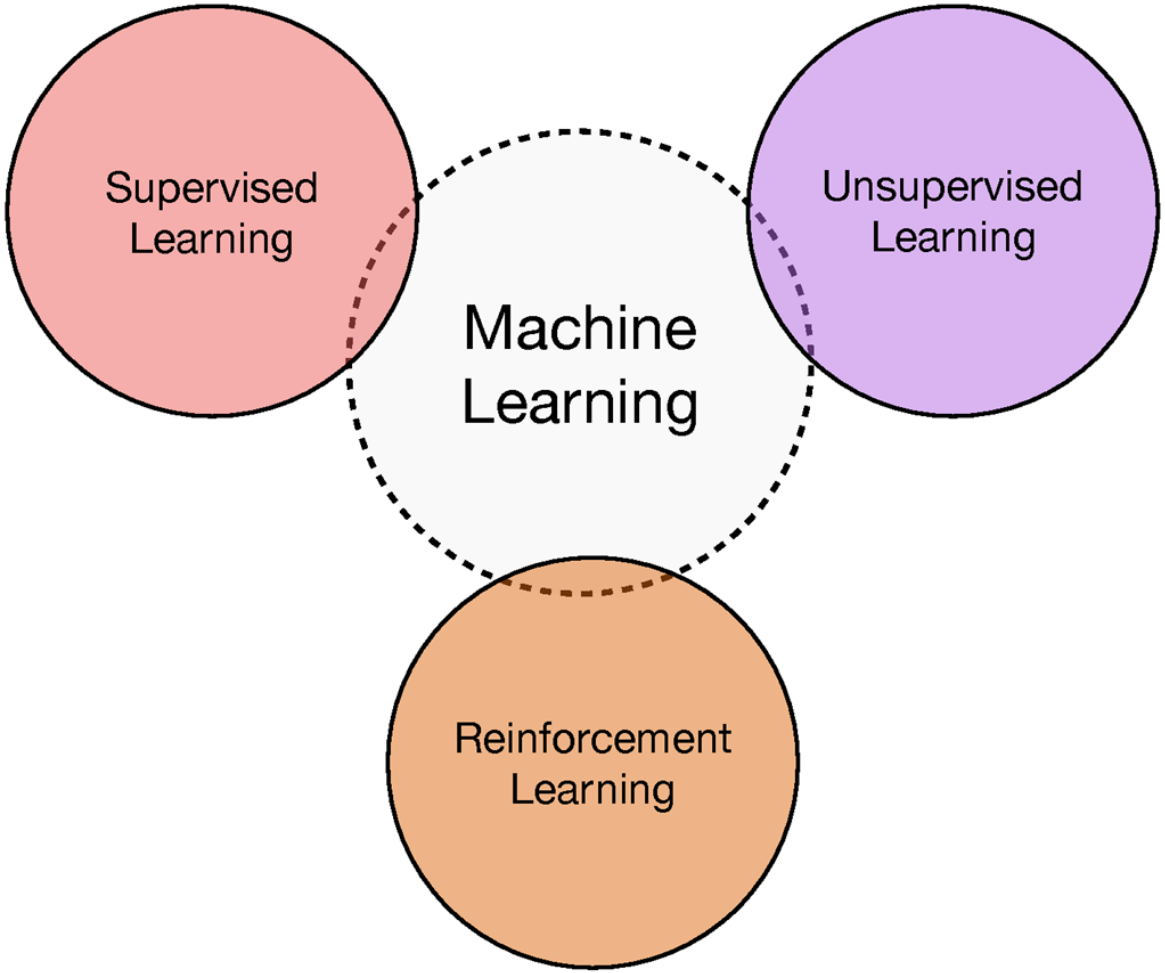
Source: Dima Fishman, Intro to Machine Learning



Source: Dima Fishman, Intro to Machine Learning

Lecture #5, #7, #8

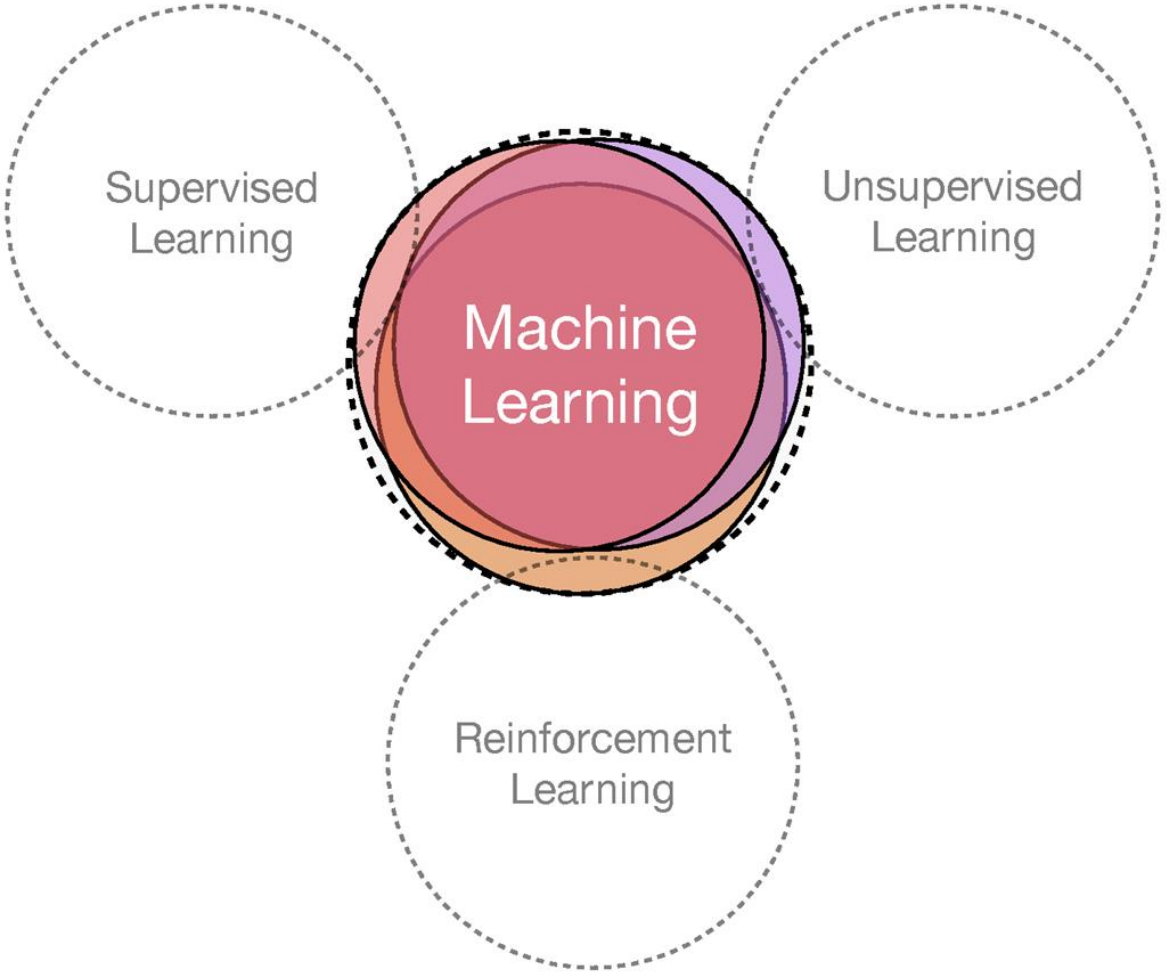
Lecture #8



Source: Dima Fishman, Intro to Machine Learning

Lecture #5, #7, #8

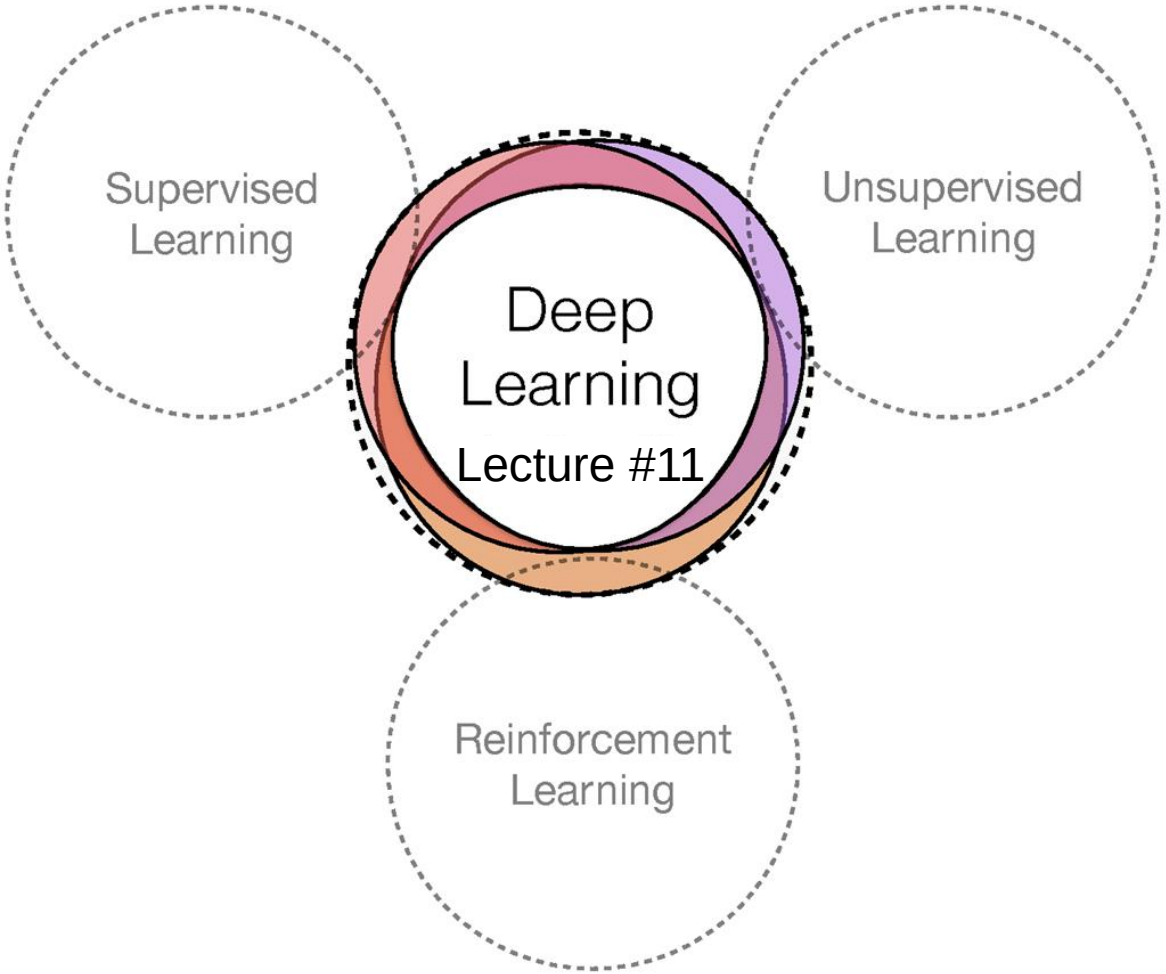
Lecture #8



Source: Dima Fishman, Intro to Machine Learning

Lecture #5, #7, #8

Lecture #8



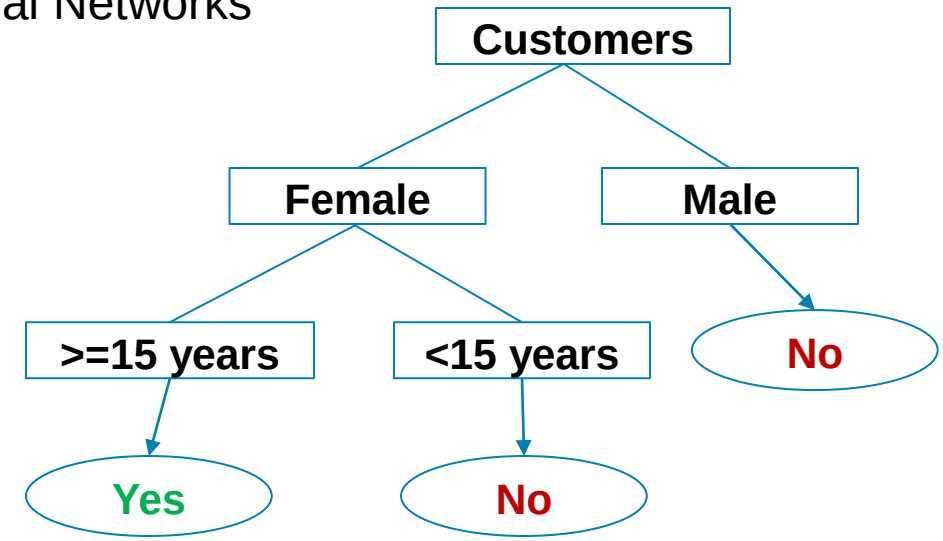
Source: Dima Fishman, Intro to Machine Learning

Supervised Machine Learning – Decision Trees

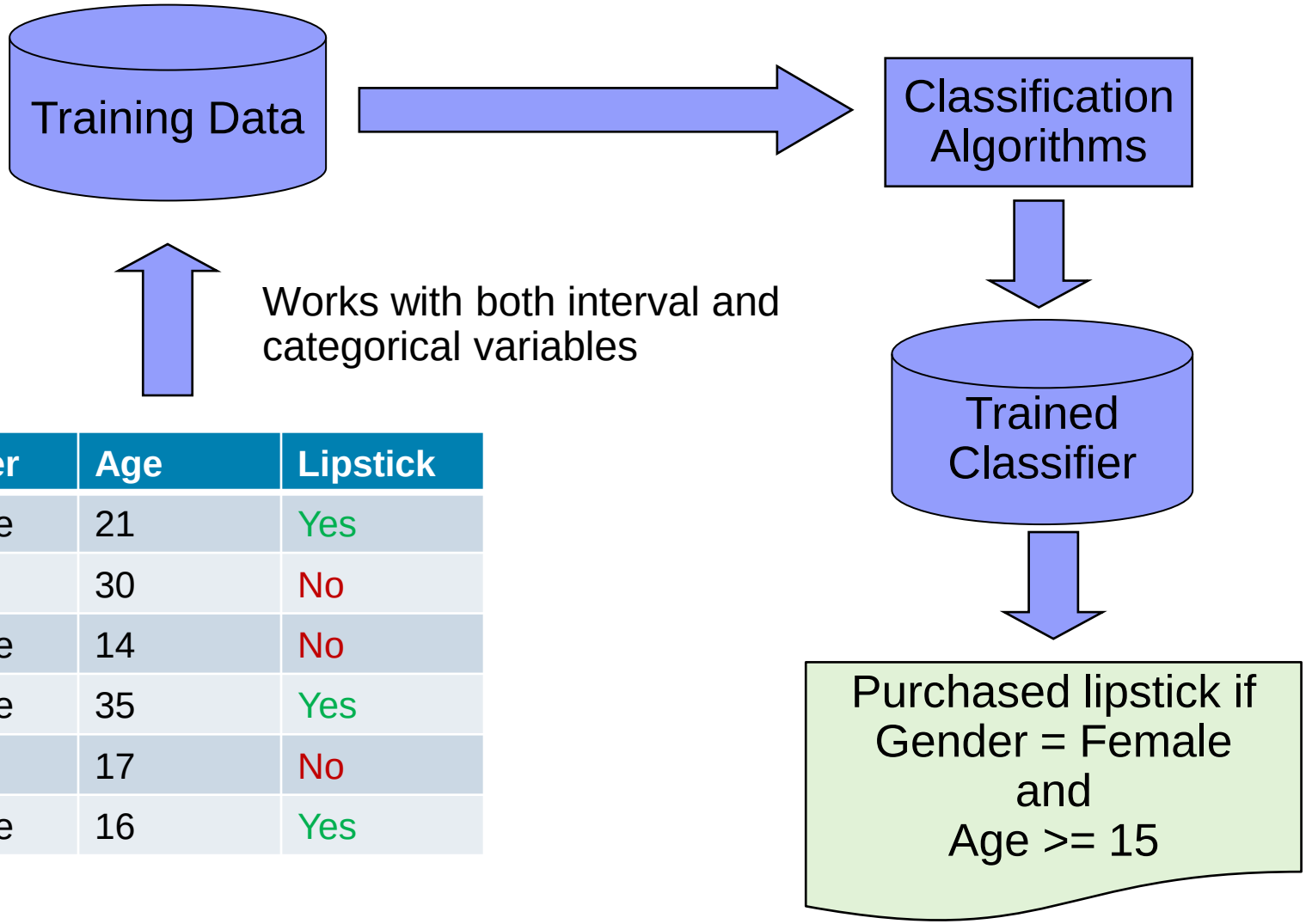
Classification

- Classification is a supervised learning technique, which maps data into predefined classes or groups
- Training set contains a set of records, where one of the records indicates class
- Modeling objective is to assign a class variable to all of the records, using attributes of other variables to predict a class
- Data is divided into test / train, where “train” is used to build the model and “test” is used to validate the accuracy of classification
- Typical techniques: Decision Trees, Neural Networks

Gender	Age	Lipstick
Female	21	Yes
Male	30	No
Female	14	No
Female	35	Yes
Male	17	No
Female	16	Yes



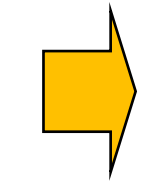
Classification: creating model



Gender	Age	Lipstick
Female	21	Yes
Male	30	No
Female	14	No
Female	35	Yes
Male	17	No
Female	16	Yes

Classification: applying rules

Gender	Age	Lipstick
Female	27	?
Male	55	?
Female	47	?
Male	39	?
Female	27	?
Male	19	?



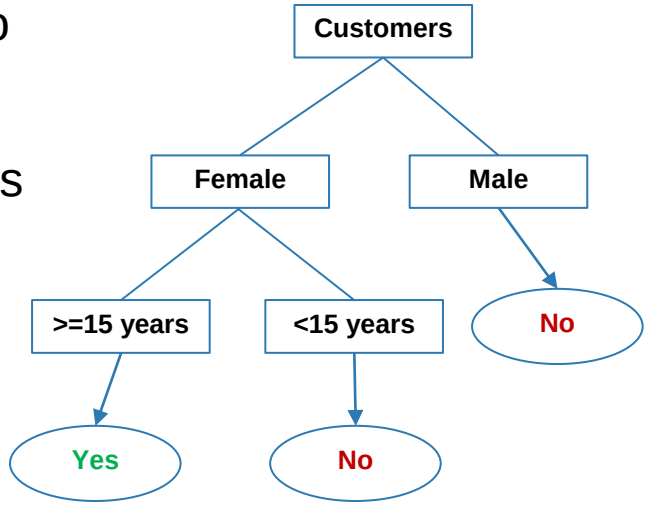
Apply Scoring

Gender	Age	Lipstick
Female	27	P Yes
Male	55	P No
Female	47	P Yes
Male	39	P No
Female	27	P Yes
Male	19	P No

If
Gender = Female
and
Age >= 15 then
Purchase lipstick = YES

Decision (classification) trees

- A tree can be "learned" by splitting the source set into subsets based on an attribute value test
- Tree partitions samples into mutually exclusive groups by selecting the best splitting attribute, one group for each terminal node
- The process is repeated recursively for each derived subset, until the stopping criteria is reached



- Works with both interval and categorical variables
- No need to normalize the data
- Intuitive if-then rules are easy to extract and apply
- Best applied to binary outcomes

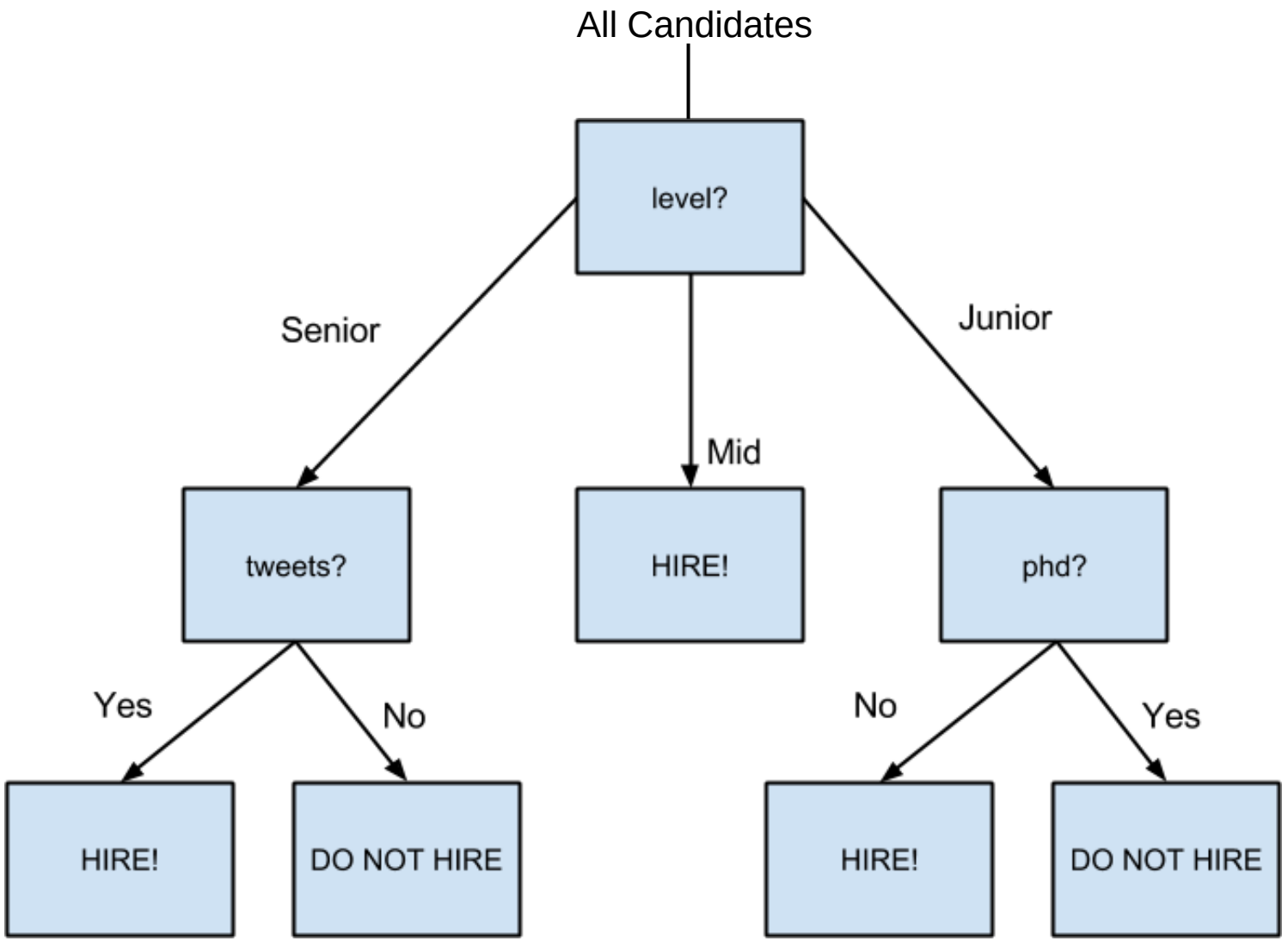
- Decision trees can be used to support multiple modeling objectives
  - Customer segmentation
  - Investment / portfolio decisions
  - Issuing a credit card or loan
  - Medical patient / disease classification

Decision (classification) trees

```
inputs = [
    ({'level':'Senior', 'lang':'Java', 'tweets':'no', 'phd':'no'},    False),
    ({'level':'Senior', 'lang':'Java', 'tweets':'no', 'phd':'yes'},   False),
    ({'level':'Mid', 'lang':'Python', 'tweets':'no', 'phd':'no'},      True),
    ({'level':'Junior', 'lang':'Python', 'tweets':'no', 'phd':'no'},   True),
    ({'level':'Junior', 'lang':'R', 'tweets':'yes', 'phd':'no'},       True),
    ({'level':'Junior', 'lang':'R', 'tweets':'yes', 'phd':'yes'},     False),
    ({'level':'Mid', 'lang':'R', 'tweets':'yes', 'phd':'yes'},         True),
    ({'level':'Senior', 'lang':'Python', 'tweets':'no', 'phd':'no'},  False),
    ({'level':'Senior', 'lang':'R', 'tweets':'yes', 'phd':'no'},       True),
    ({'level':'Junior', 'lang':'Python', 'tweets':'yes', 'phd':'no'},  True),
    ({'level':'Senior', 'lang':'Python', 'tweets':'yes', 'phd':'yes'}, True),
    ({'level':'Mid', 'lang':'Python', 'tweets':'no', 'phd':'yes'},     True),
    ({'level':'Mid', 'lang':'Java', 'tweets':'yes', 'phd':'no'},       True),
    ({'level':'Junior', 'lang':'Python', 'tweets':'no', 'phd':'yes'}, False)
]
```

```
('level',
 {'Junior': ('phd', {'no': True, 'yes': False}),
  'Mid': True,
  'Senior': ('tweets', {'no': False, 'yes': True})})
```

Decision (classification) trees



Root node

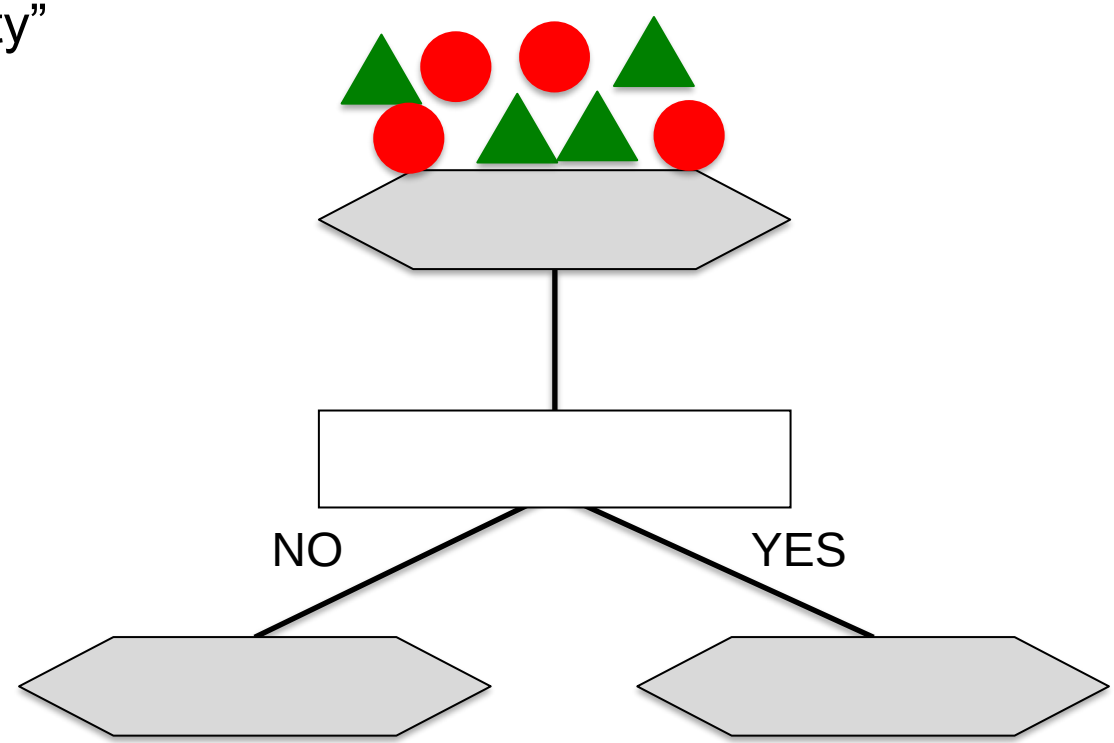
Decision node

Branches

Leaf nodes

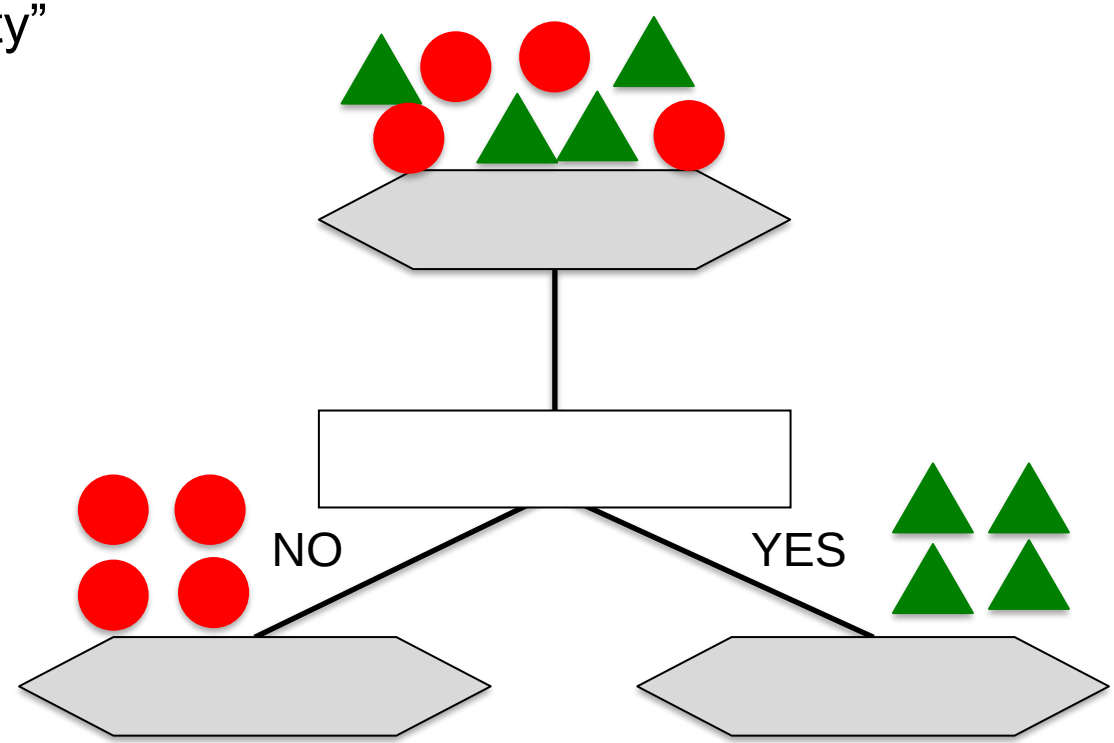
Understanding decision trees

- Decision trees are built using recursive partitioning to classify the data
- The algorithm chooses the most predictive feature to split the data on
- "Predictiveness" is based on decrease in entropy (gain in information) or "impurity"



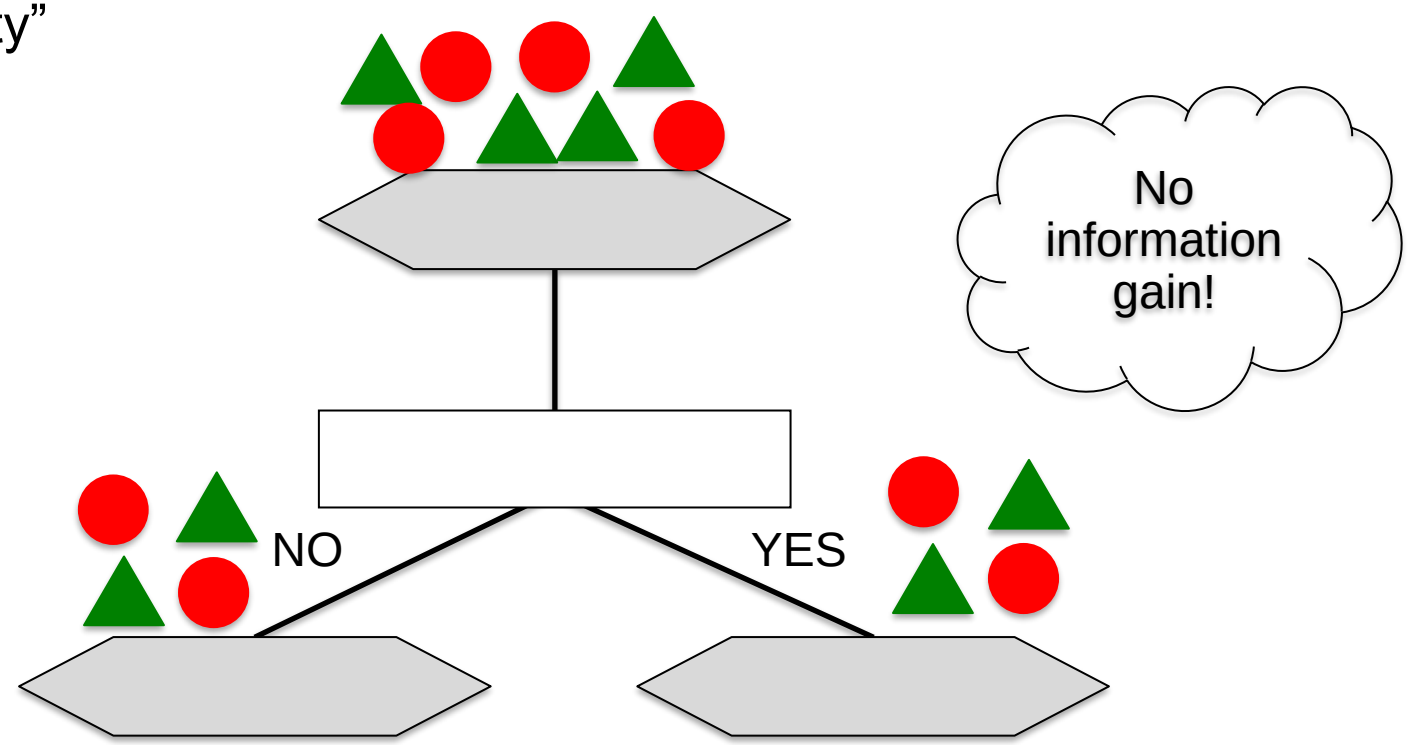
Understanding decision trees

- Decision trees are built using recursive partitioning to classify the data
- The algorithm chooses the most predictive feature to split the data on
- "Predictiveness" is based on decrease in entropy (gain in information) or "impurity"



Understanding decision trees

- Decision trees are built using recursive partitioning to classify the data
- The algorithm chooses the most predictive feature to split the data on
- "Predictiveness" is based on decrease in entropy (gain in information) or "impurity"



Understanding decision trees

- Root node partitions the data using the feature that provides the most information gain
- Information gain tells us how important a given attribute of the feature vectors is:

$$\text{Information Gain} = \text{entropy(parent)} \text{ - average entropy(children)}$$

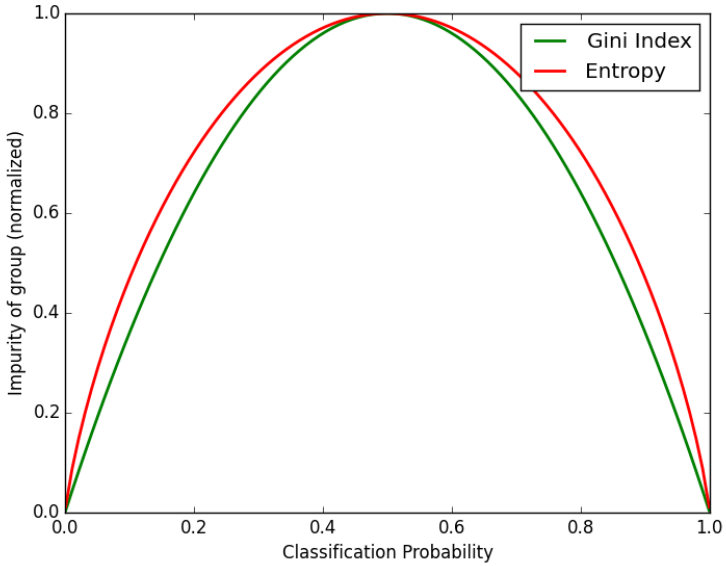
- Entropy is a common measure of target class impurity (*i* is each of the target classes, p_i is proportion of the number of elements in class 0 or 1):

$$\text{Entropy} = \sum_i -p_i \log_2 p_i$$

- Gini Index is another measure of impurity:

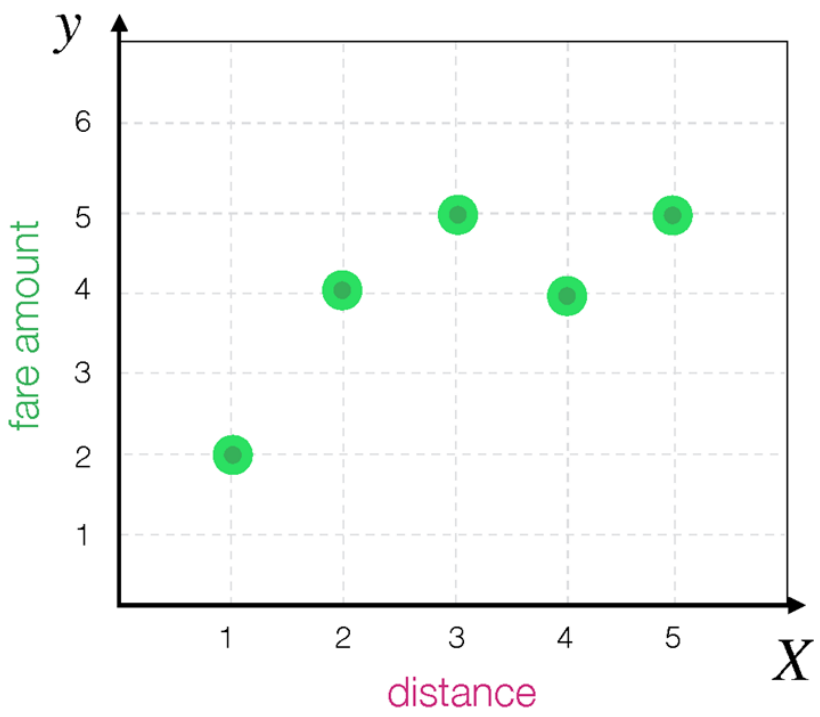
$$\text{Gini} = 1 - \sum_i p_i^2$$

Gini impurity is computationally faster as it doesn't require calculating logarithmic functions, though in reality which of the two methods is used rarely makes too much of a difference.



Decision Tree Algorithm

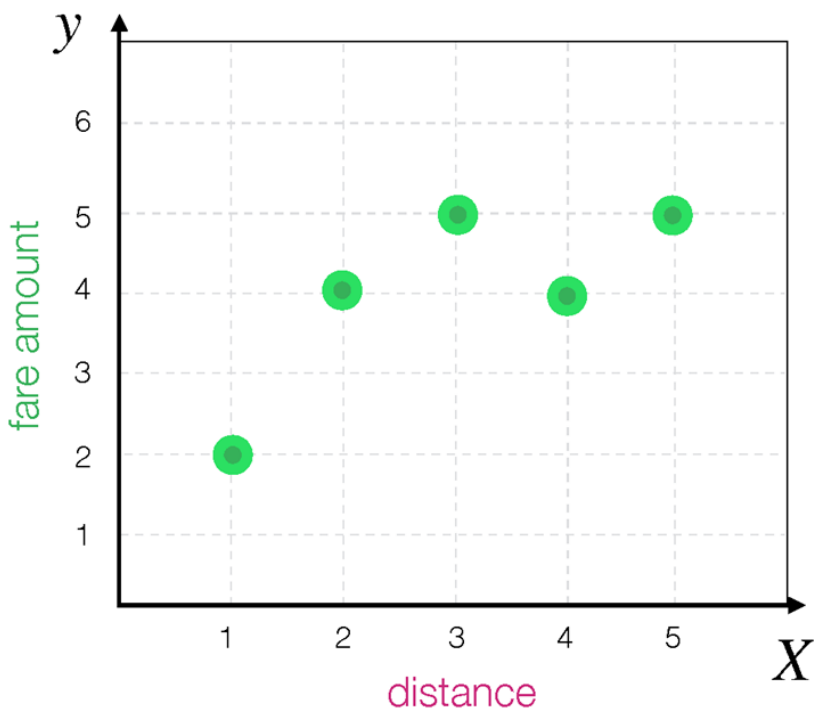
By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**



Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

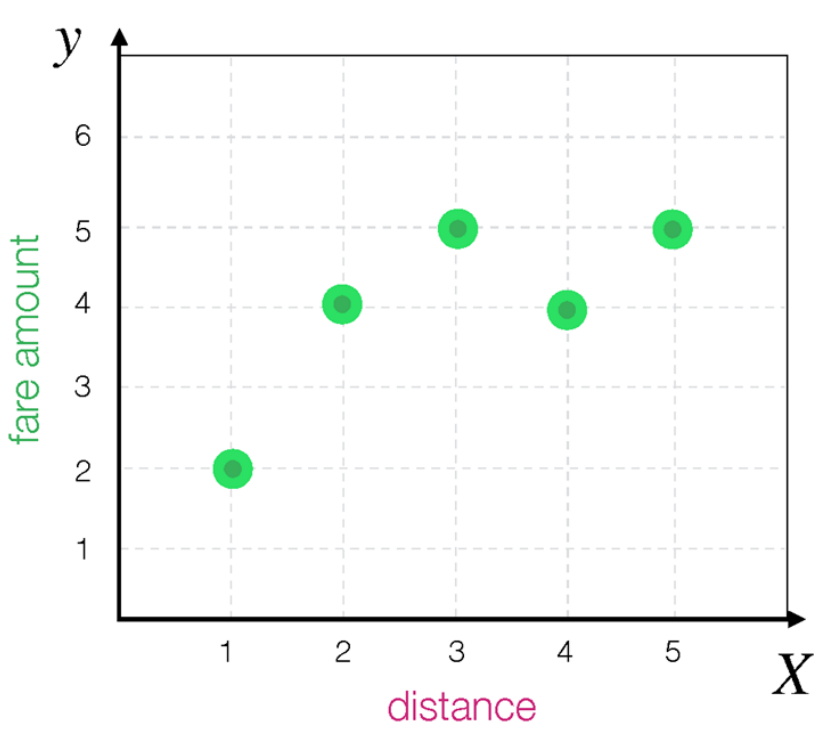


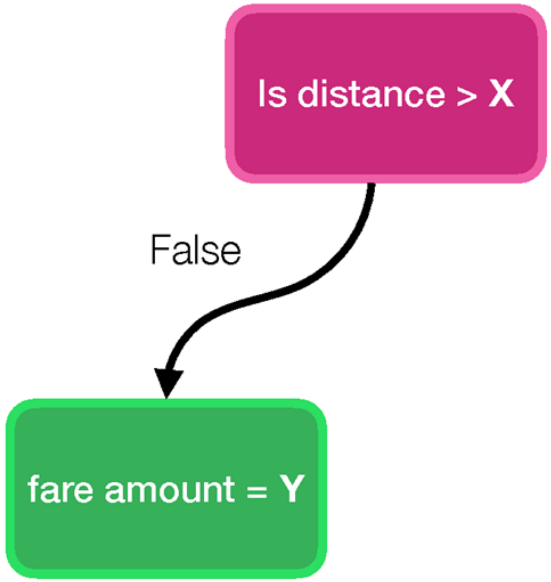
Is distance > X

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

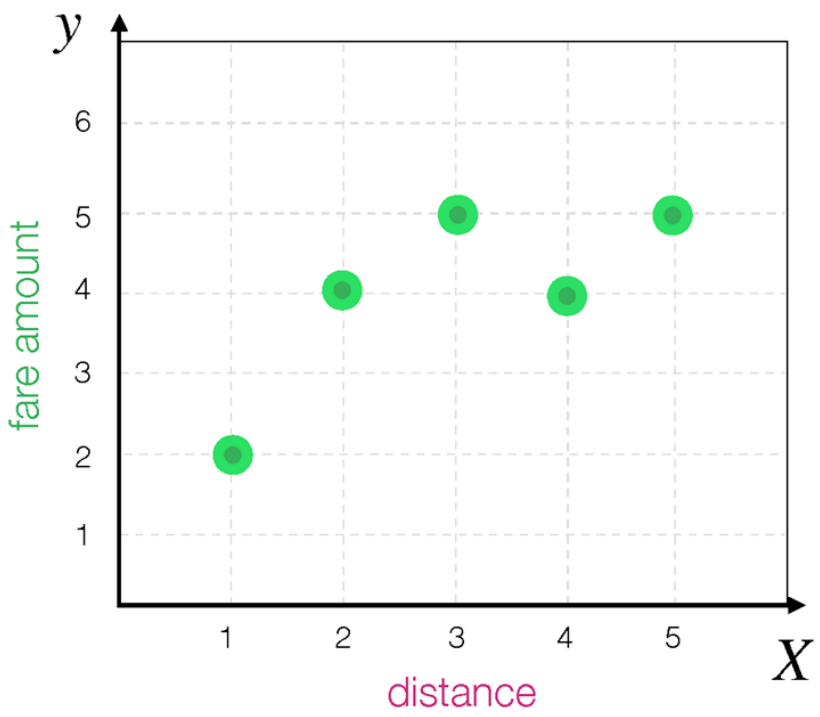


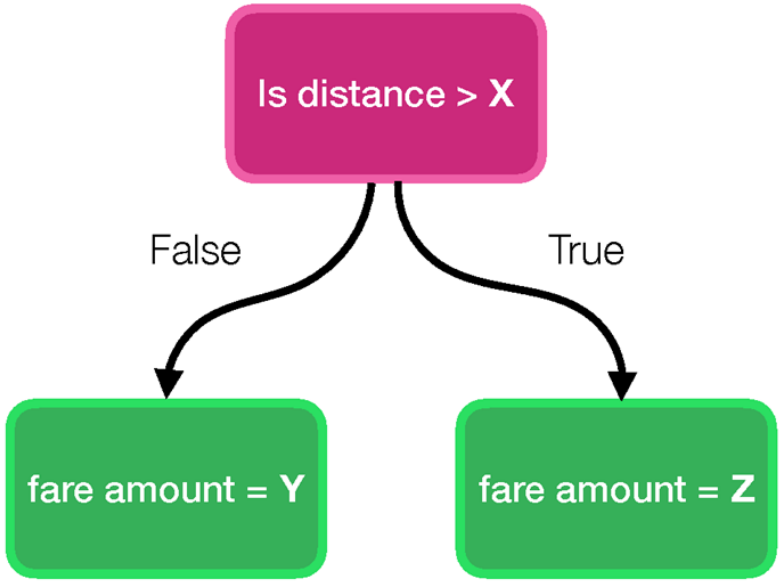


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

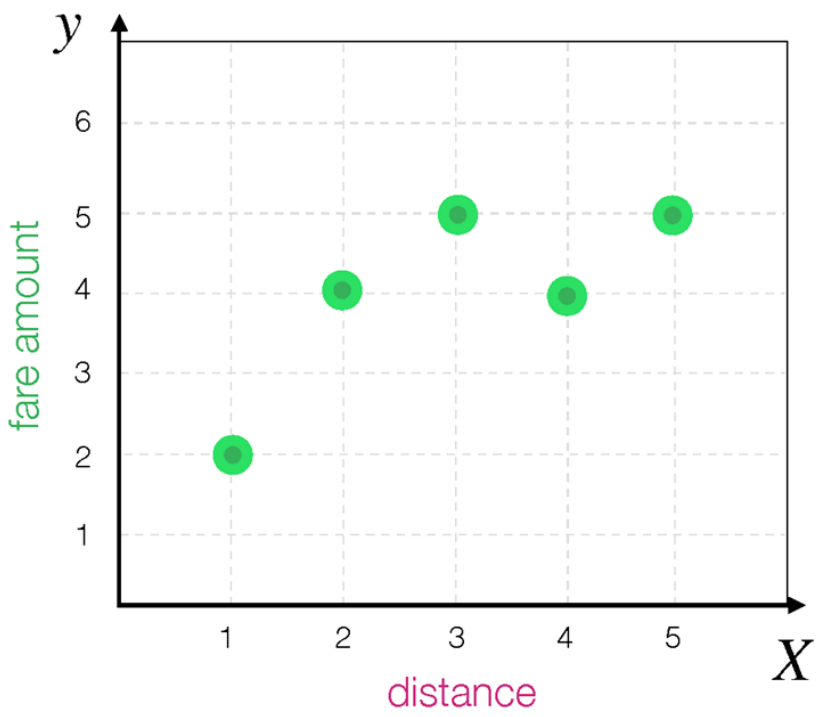


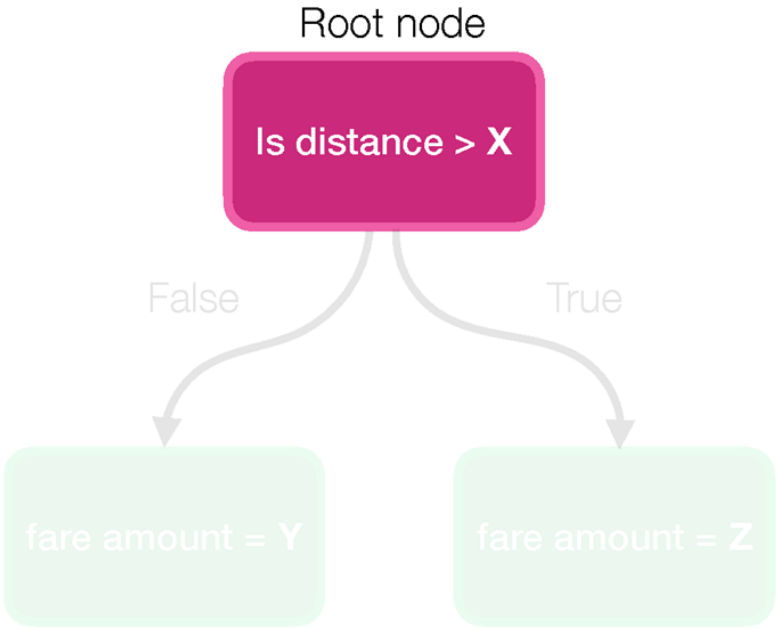


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

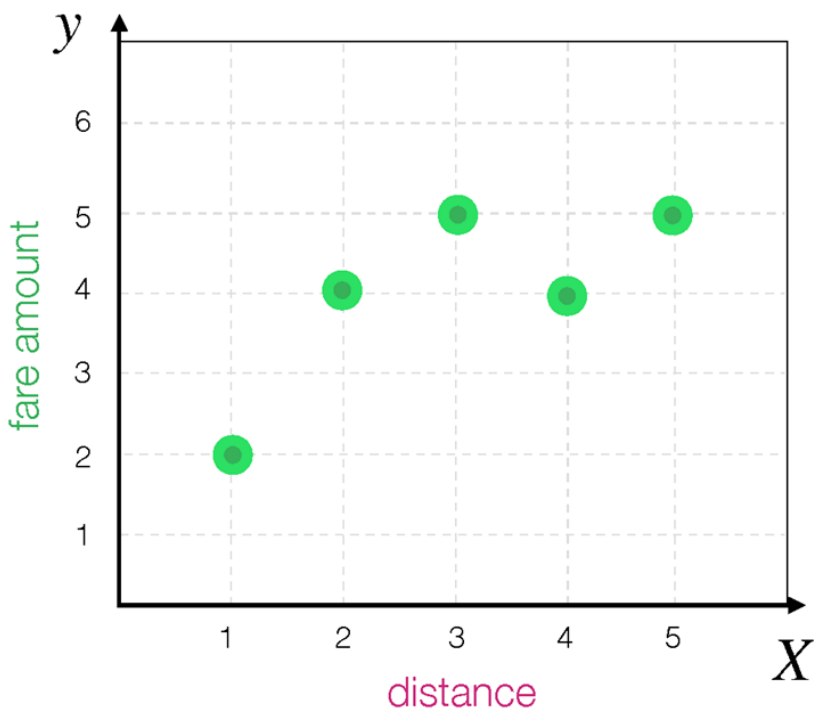


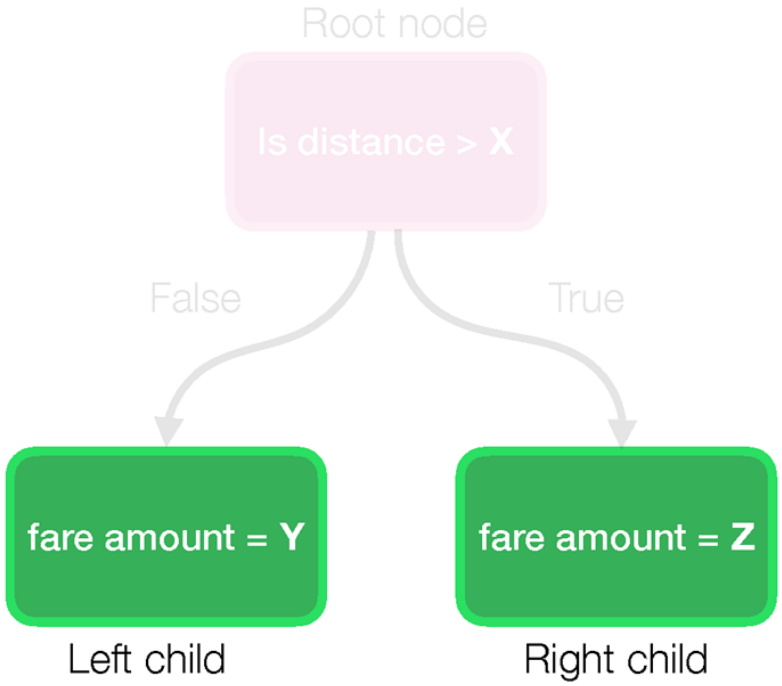


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

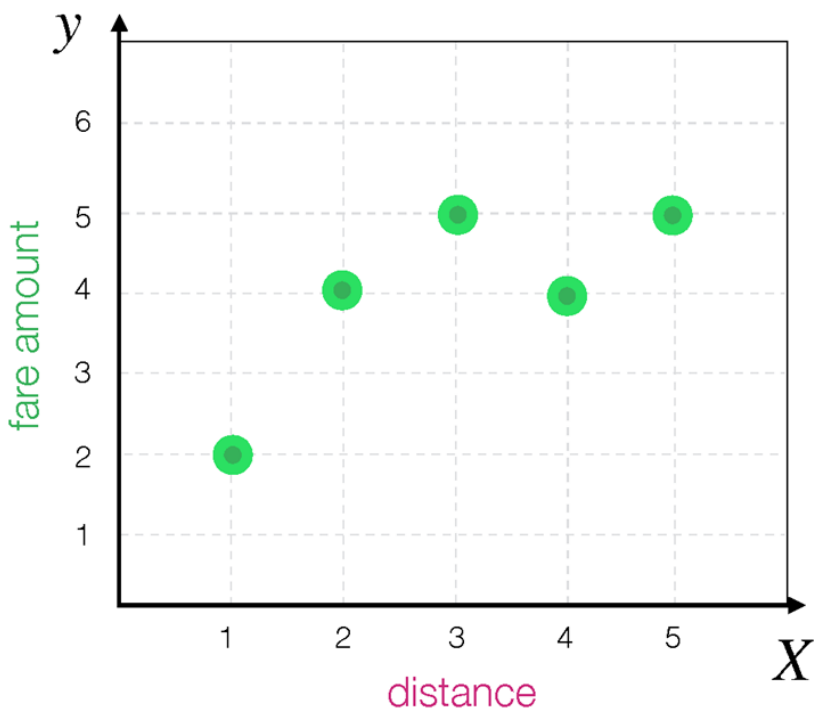


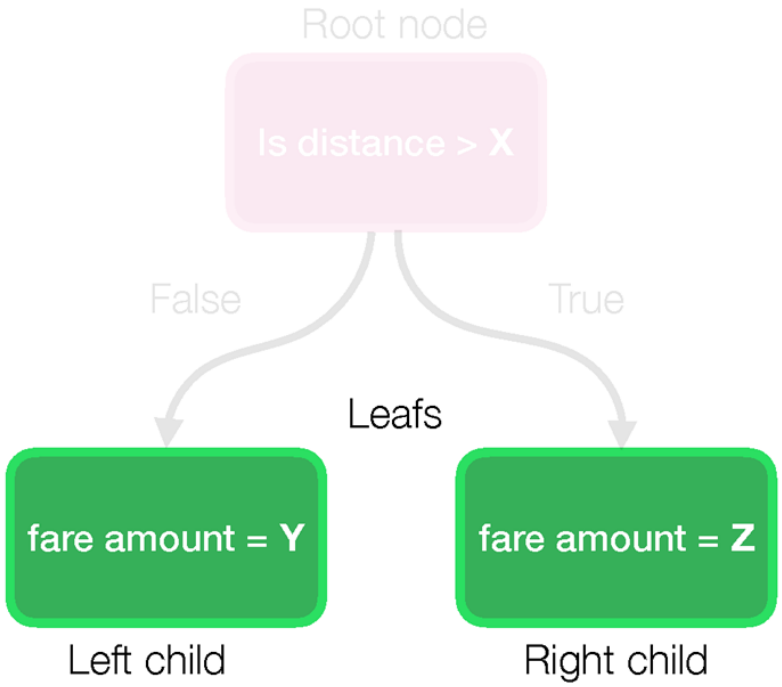


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

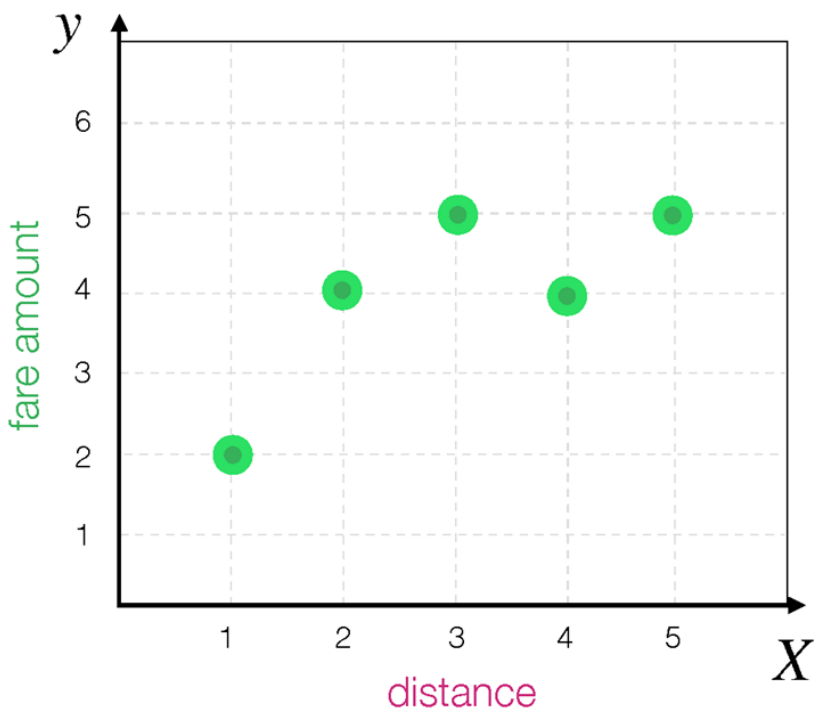


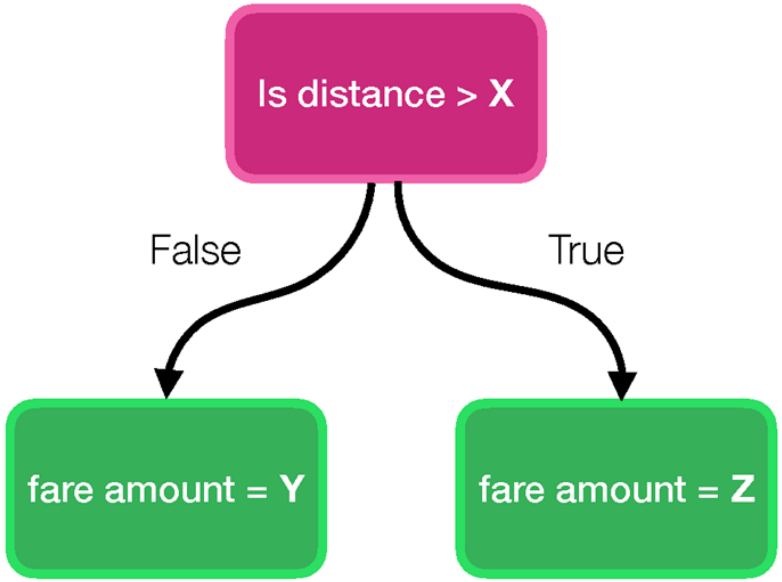


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

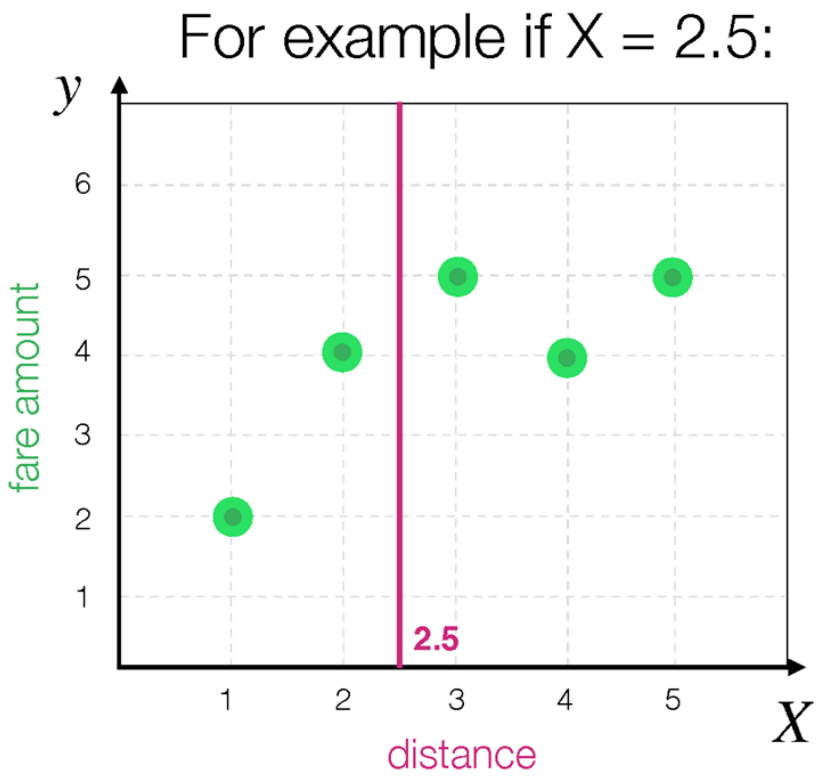


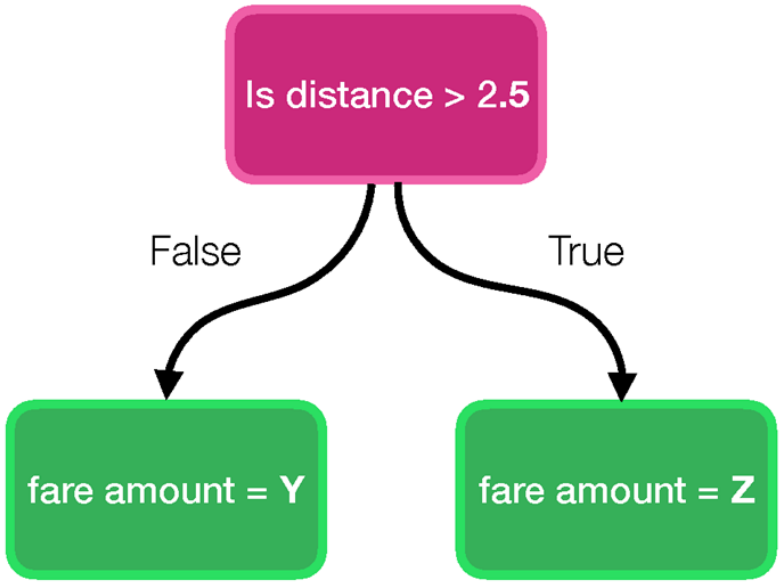


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Here, **X** may correspond to any vertical line.

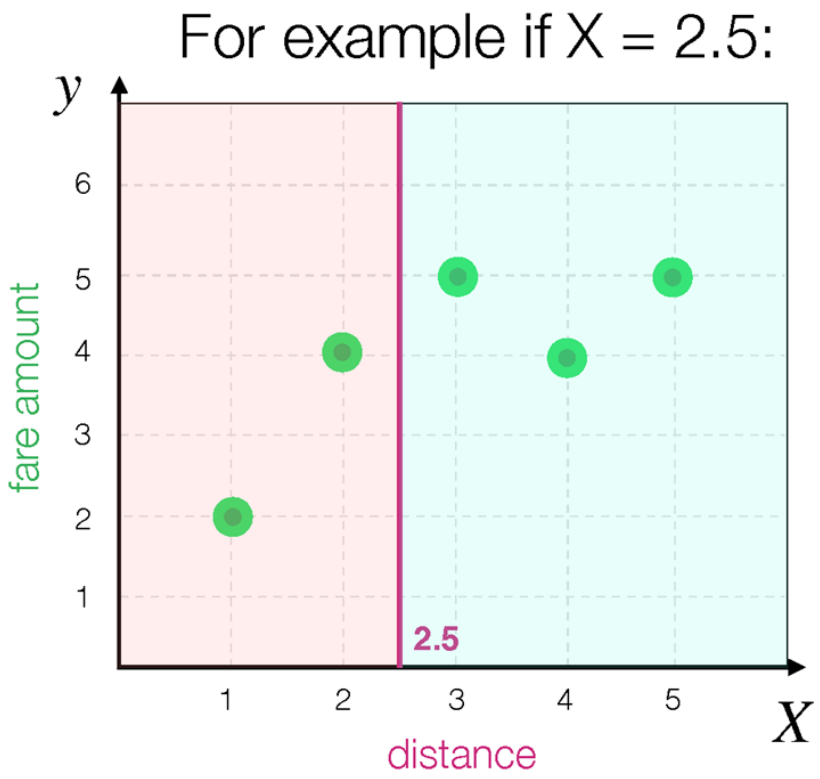


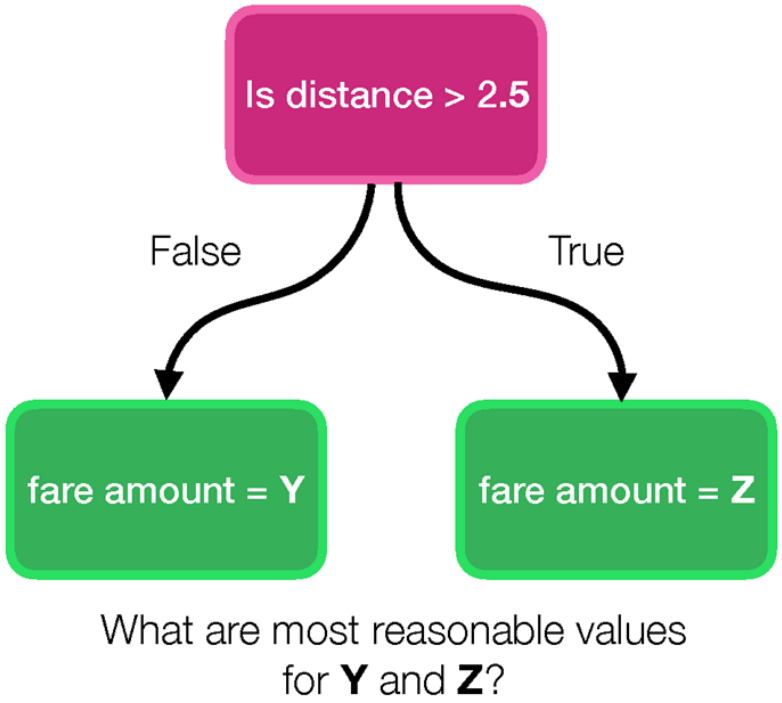


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Here, **X** may correspond to any vertical line.



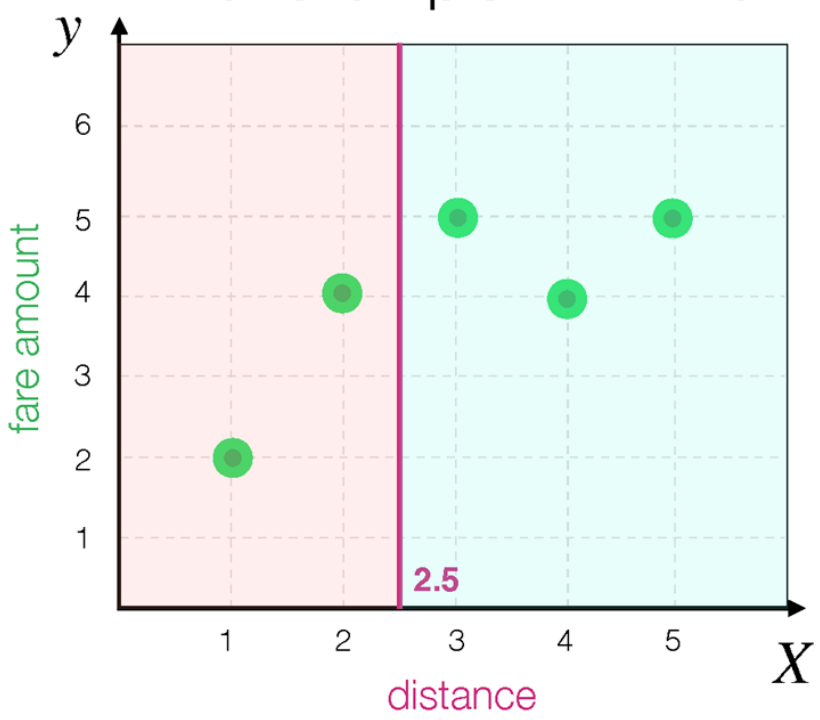


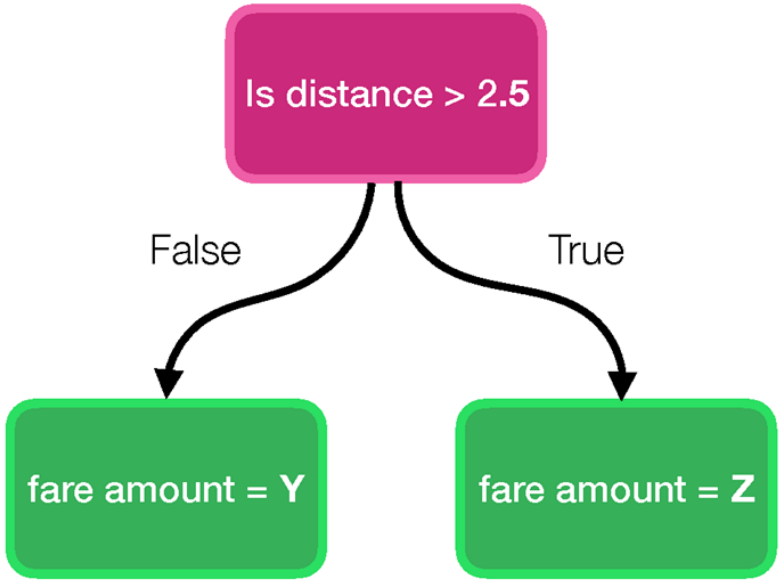
Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Here, **X** may correspond to any vertical line.

For example if X = 2.5:





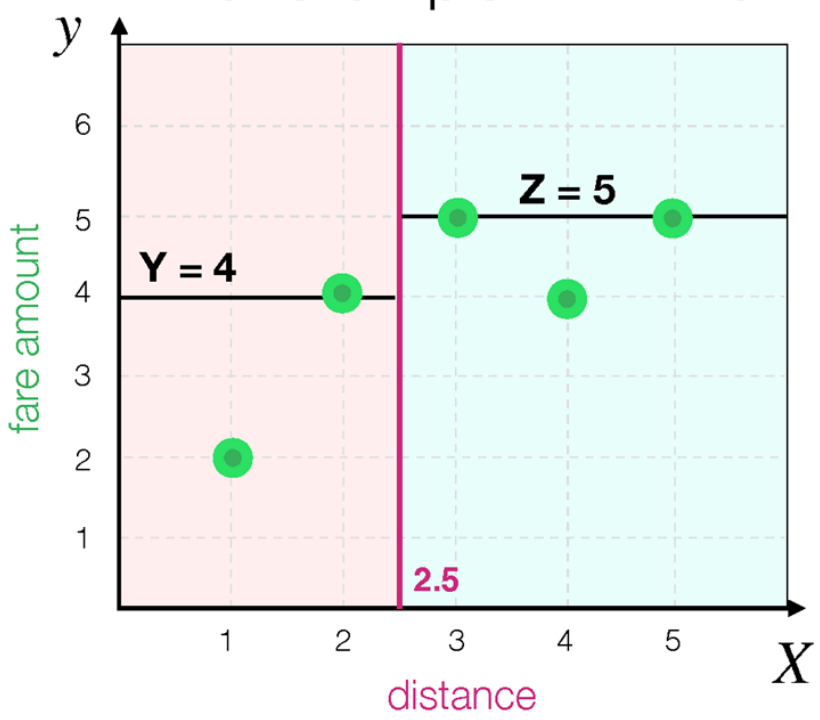
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

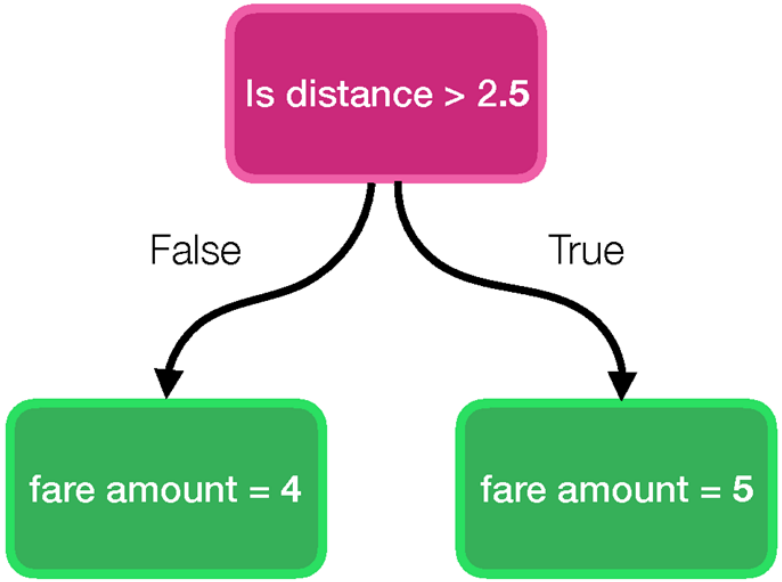
Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

What would be MSE if **Y = 4** and **Z = 5**?

For example if X = 2.5:





What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

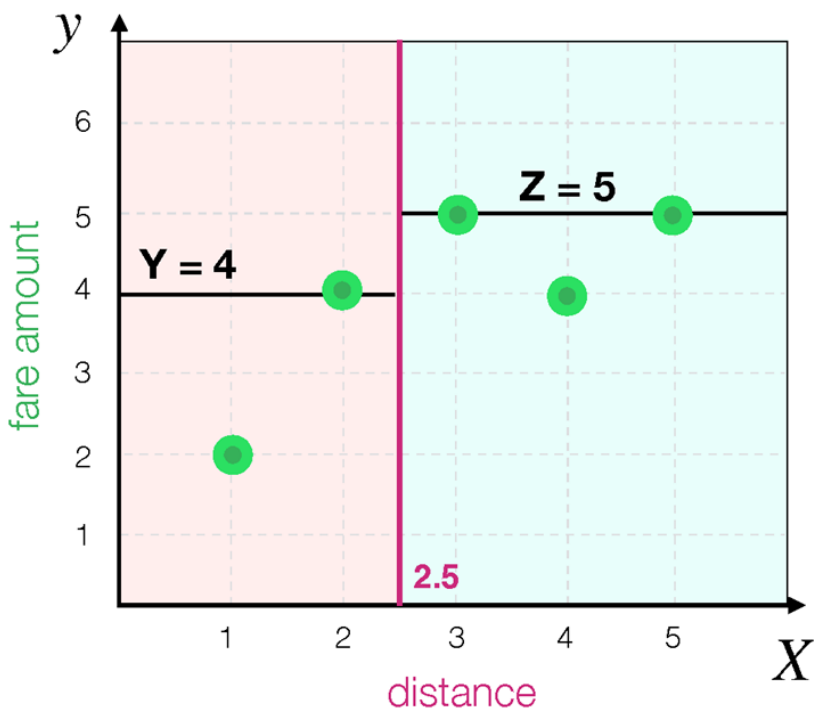
Source: Dima Fishman, Supervised Machine Learning

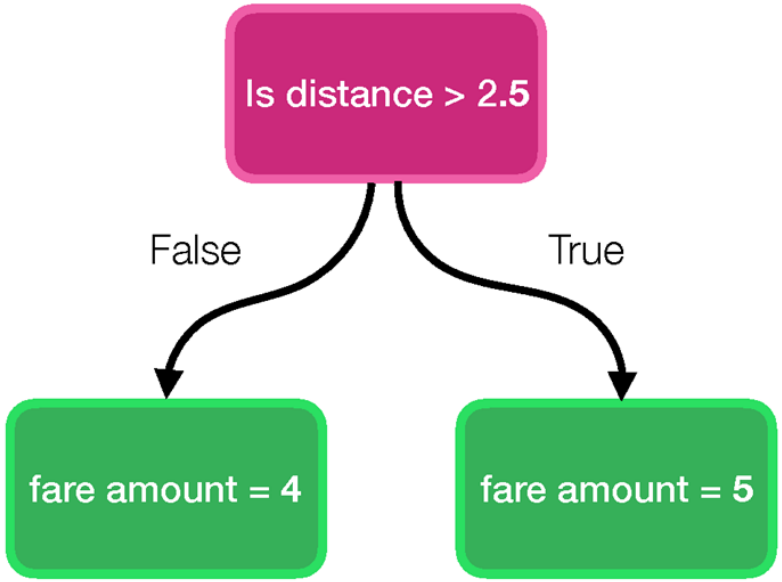
Decision Tree Algorithm

real value

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

predicted value



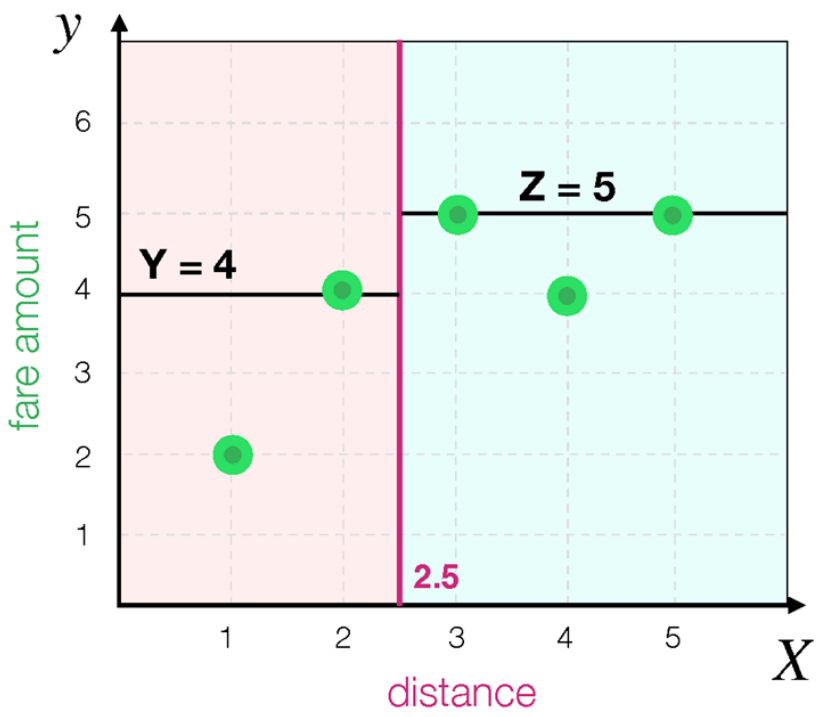


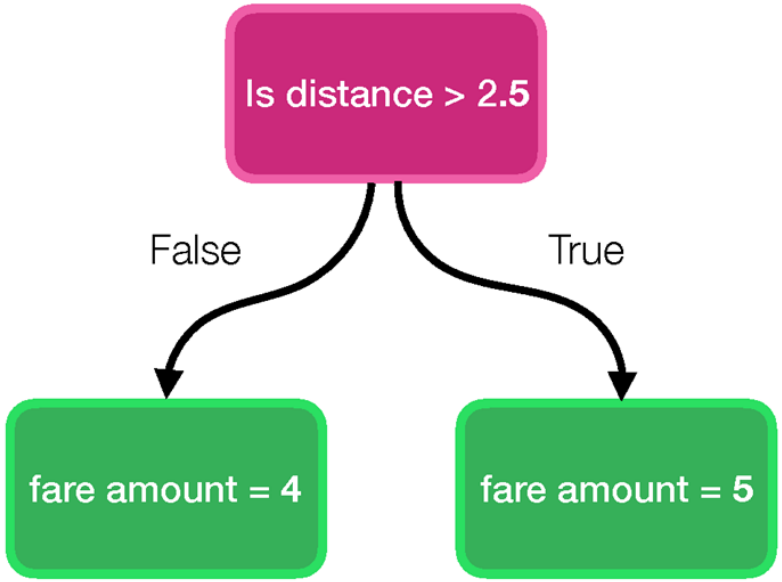
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + (y_3 - \hat{y}_3)^2 + (y_4 - \hat{y}_4)^2 + (y_5 - \hat{y}_5)^2}{5}$$



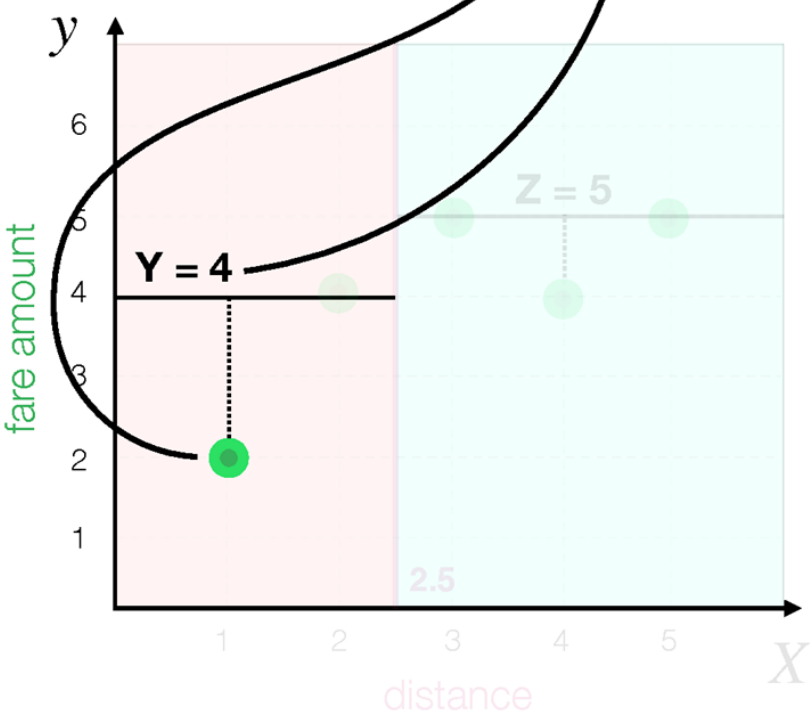


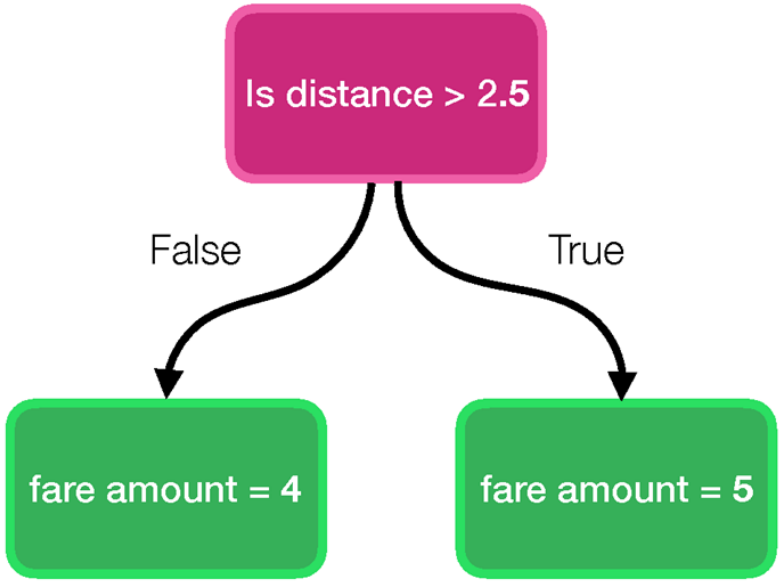
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(2-4)^2 + (y_2 - \hat{y}_2)^2 + (y_3 - \hat{y}_3)^2 + (y_4 - \hat{y}_4)^2 + (y_5 - \hat{y}_5)^2}{5}$$



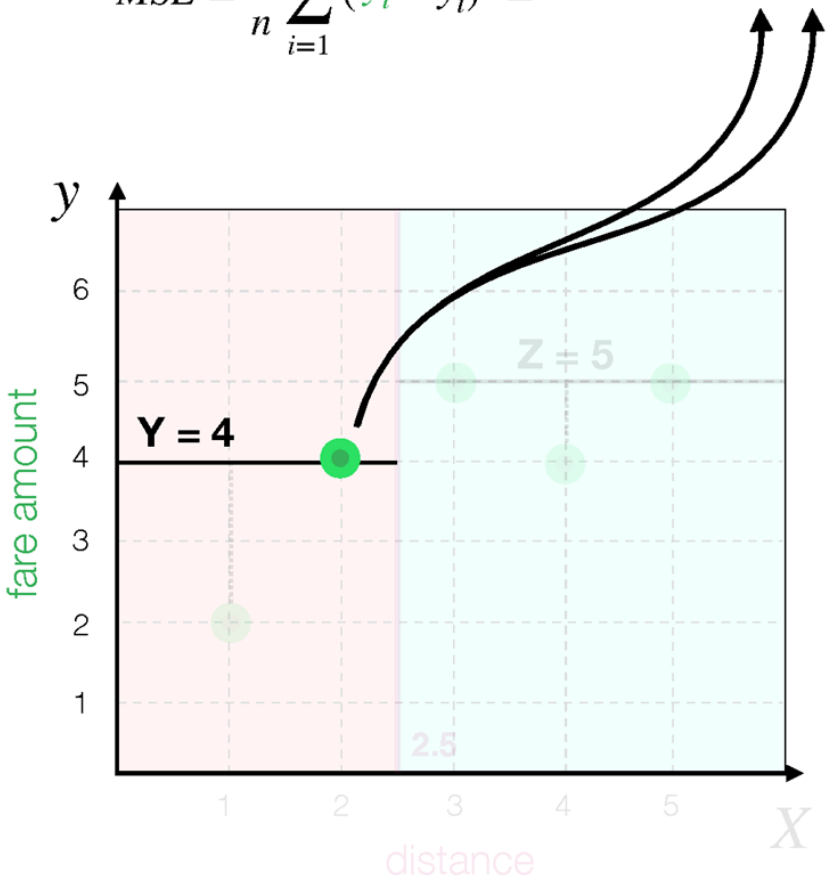


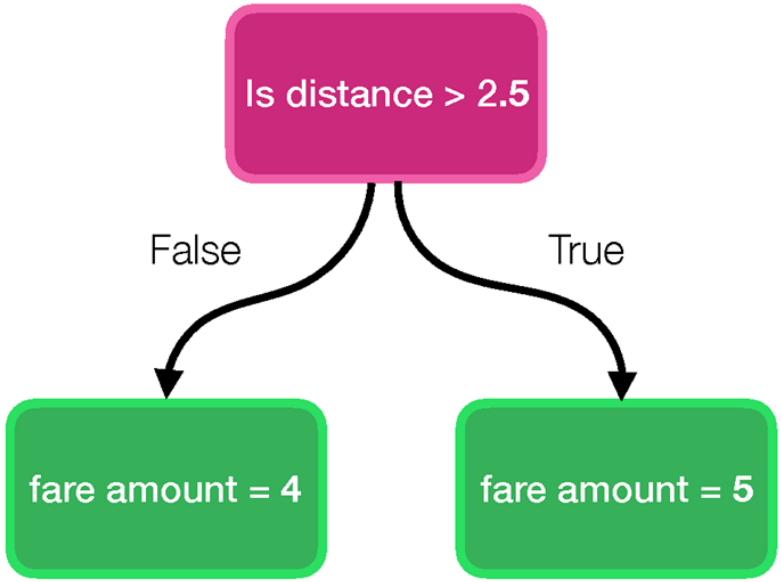
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(2-4)^2 + (4-4)^2 + (y_3 - \hat{y}_3)^2 + (y_4 - \hat{y}_4)^2 + (y_5 - \hat{y}_5)^2}{5}$$



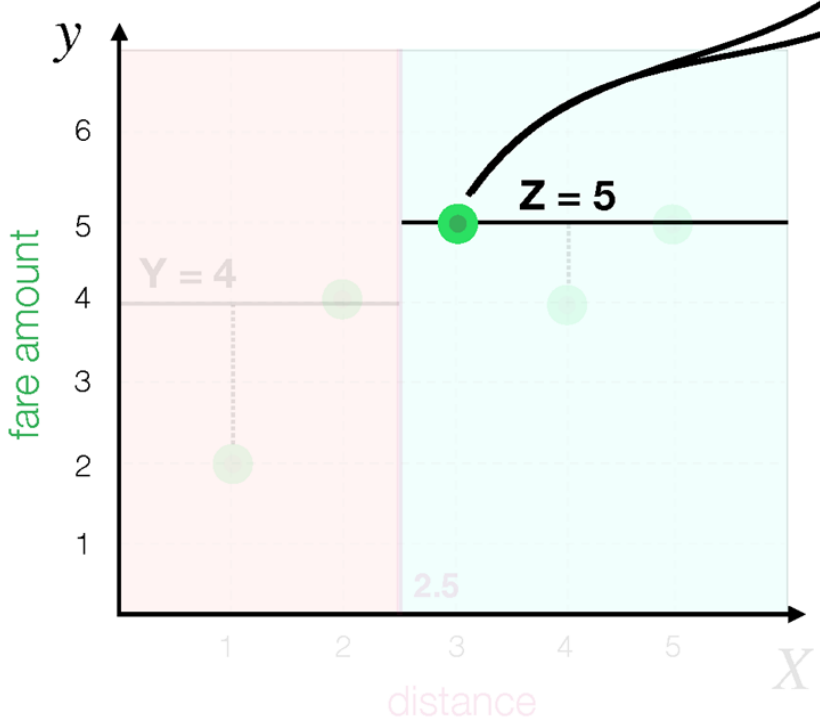


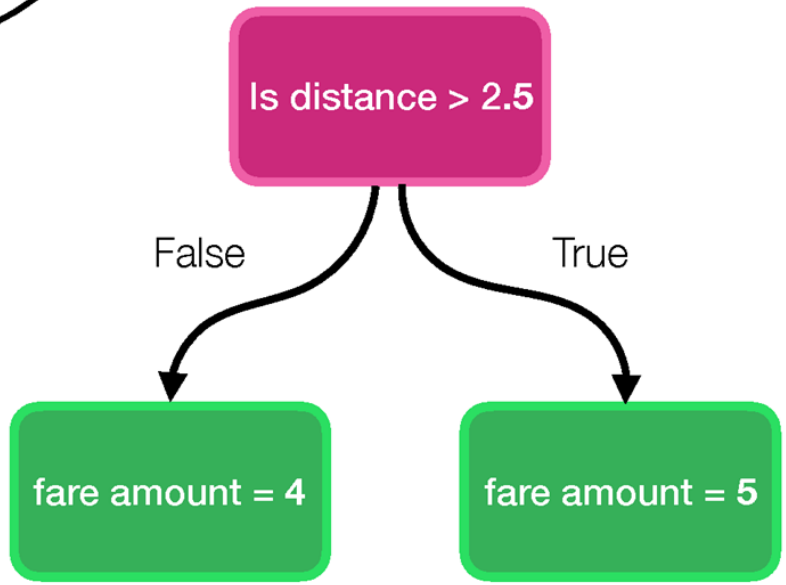
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(2-4)^2 + (4-4)^2 + (5-5)^2 + (y_4 - \hat{y}_4)^2 + (y_5 - \hat{y}_5)^2}{5}$$



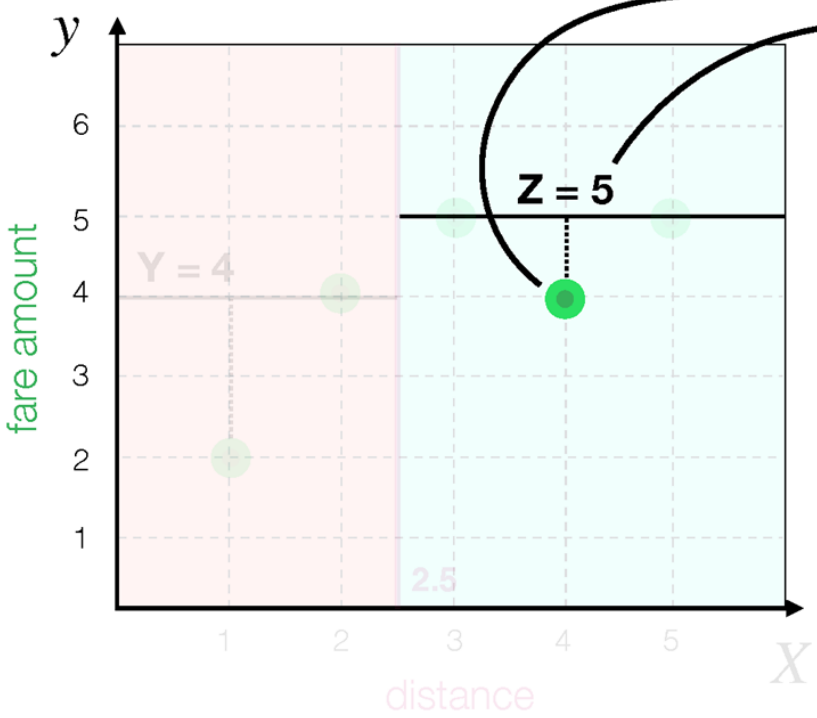


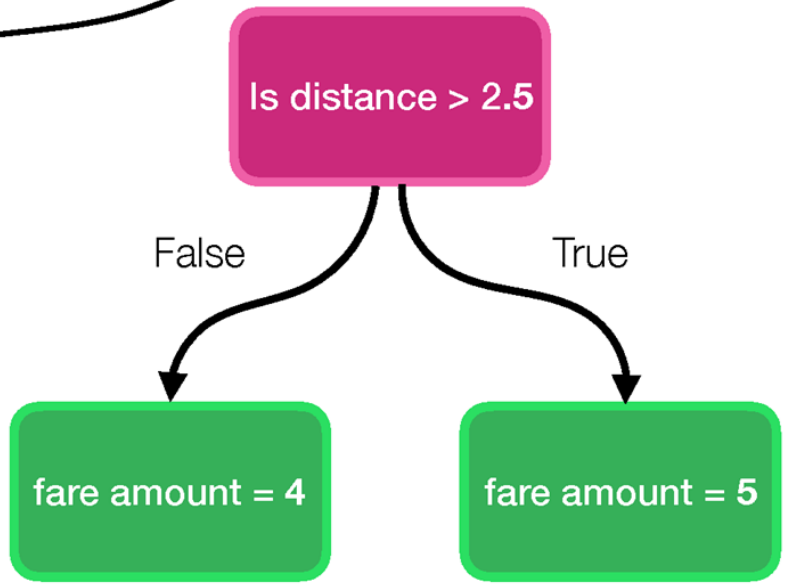
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(2-4)^2 + (4-4)^2 + (5-5)^2 + (4-5)^2 + (y_5 - \hat{y}_5)^2}{5}$$



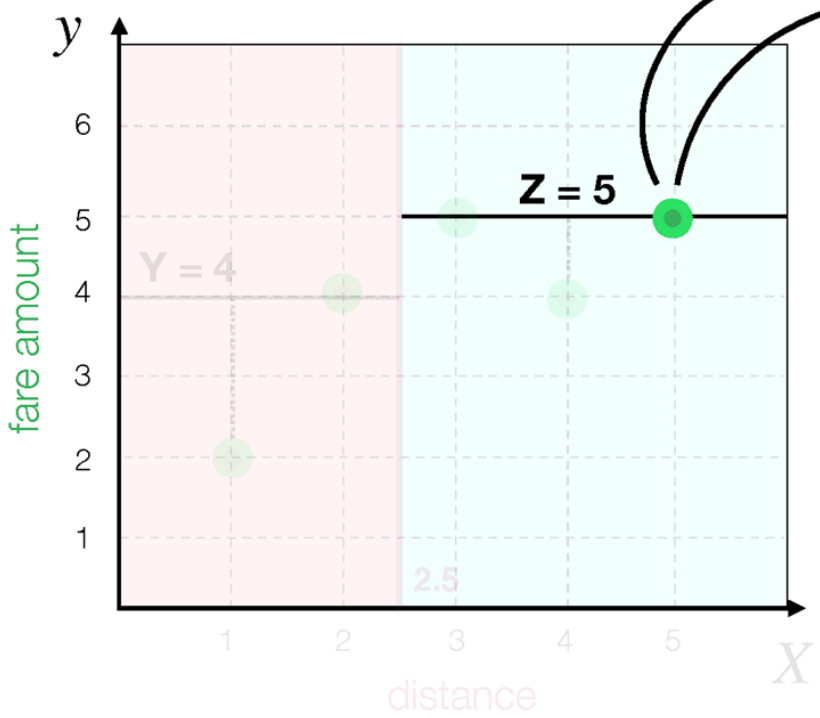


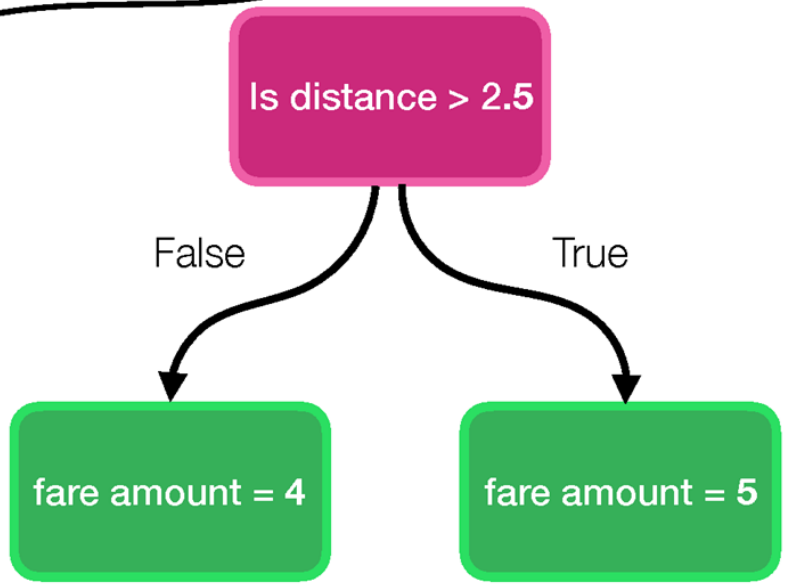
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(2-4)^2 + (4-4)^2 + (5-5)^2 + (4-5)^2 + (5-5)^2}{5}$$



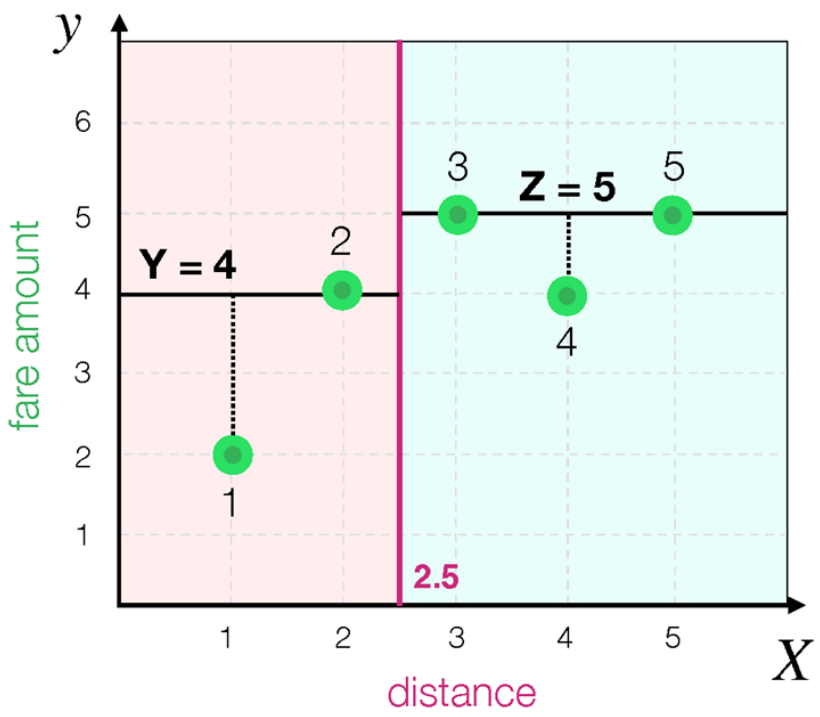


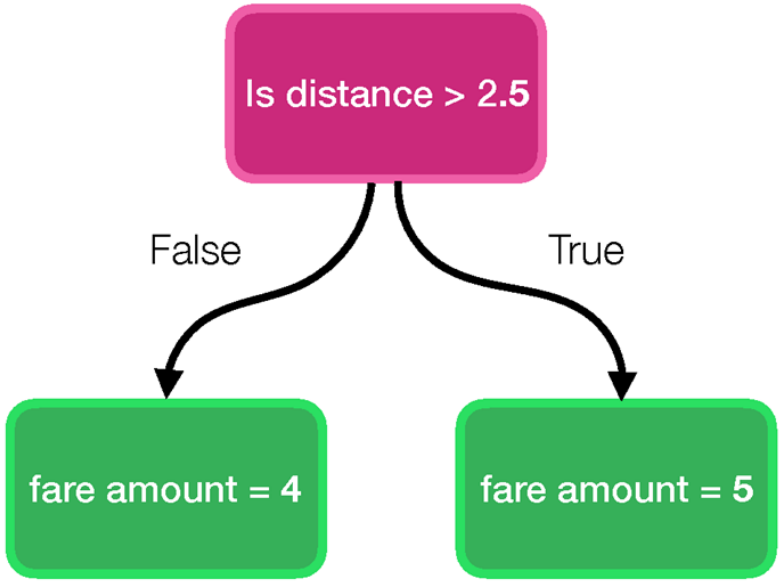
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(-2)^2 + (0)^2 + (0)^2 + (-1)^2 + (0)^2}{5}$$



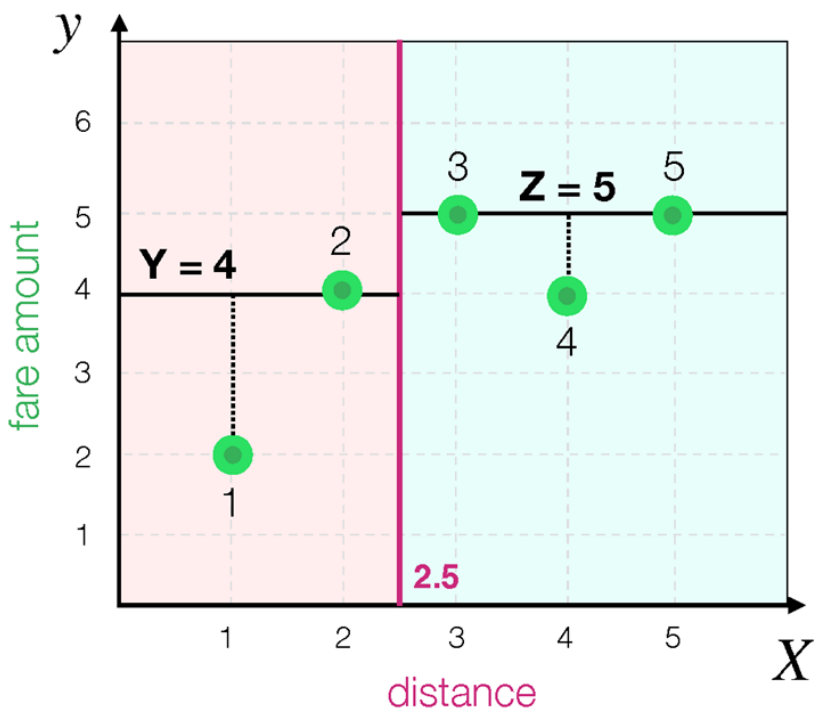


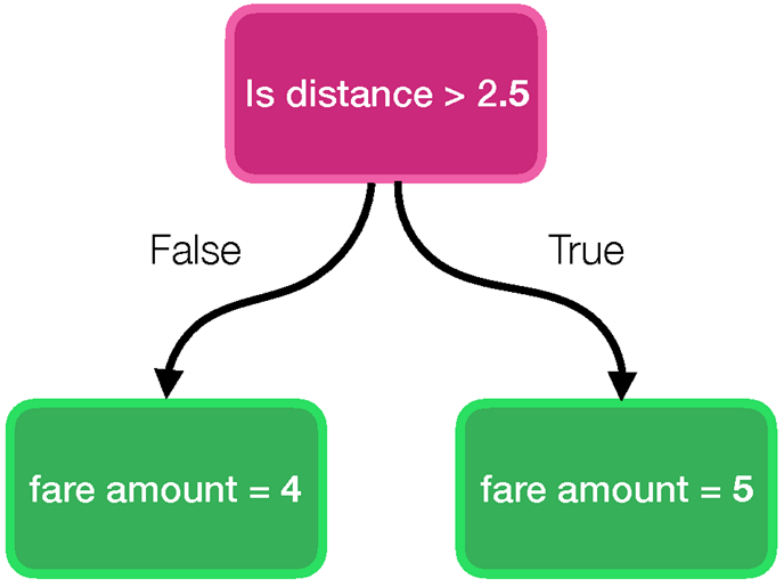
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{4+0+0+1+0}{5}$$



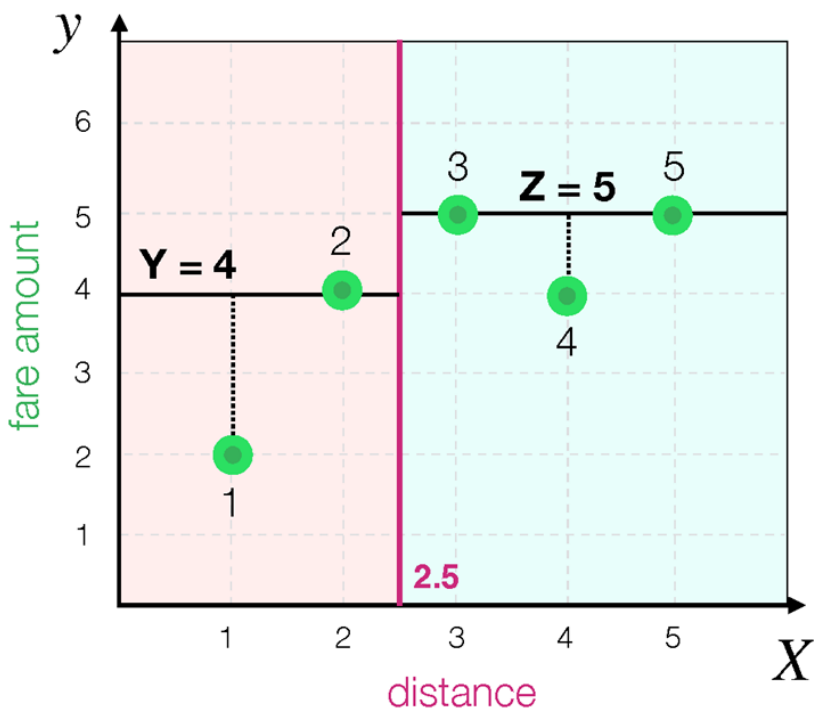


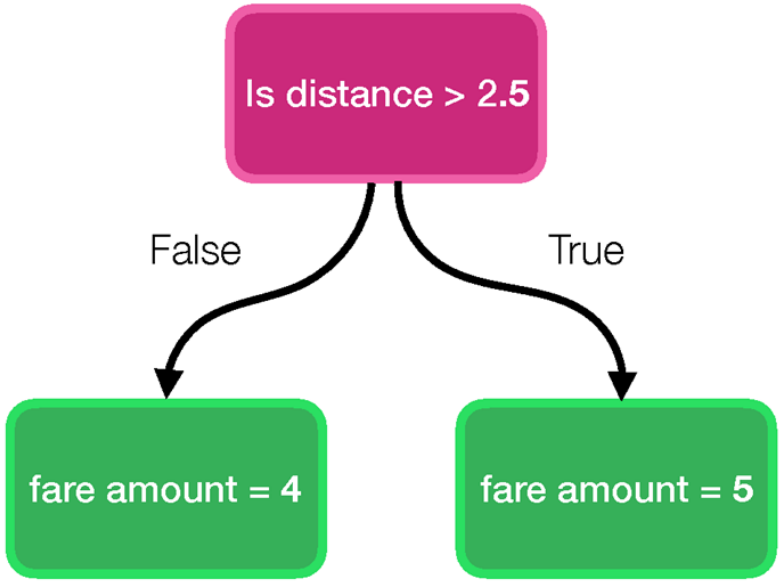
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{5}{5}$$



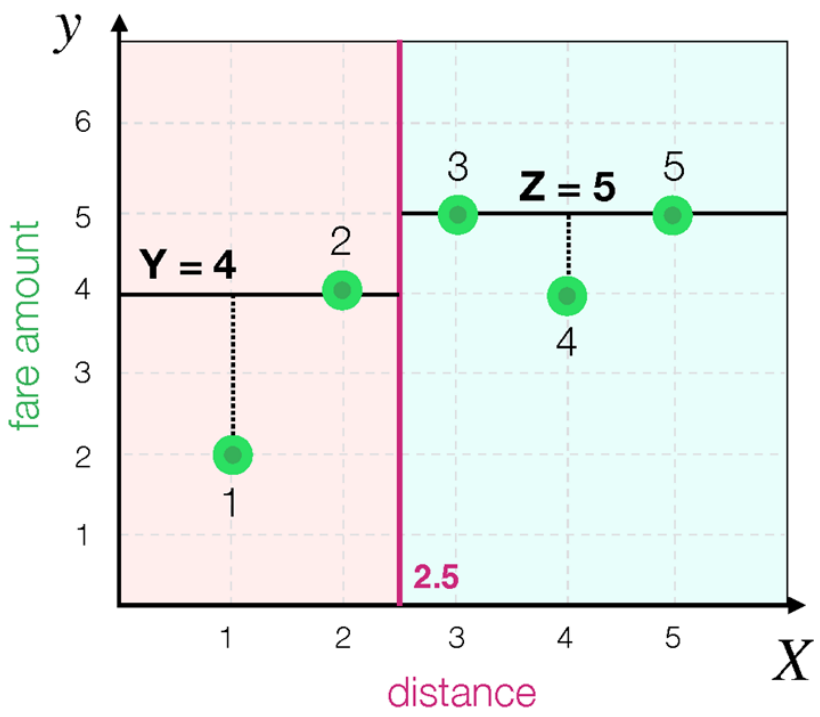


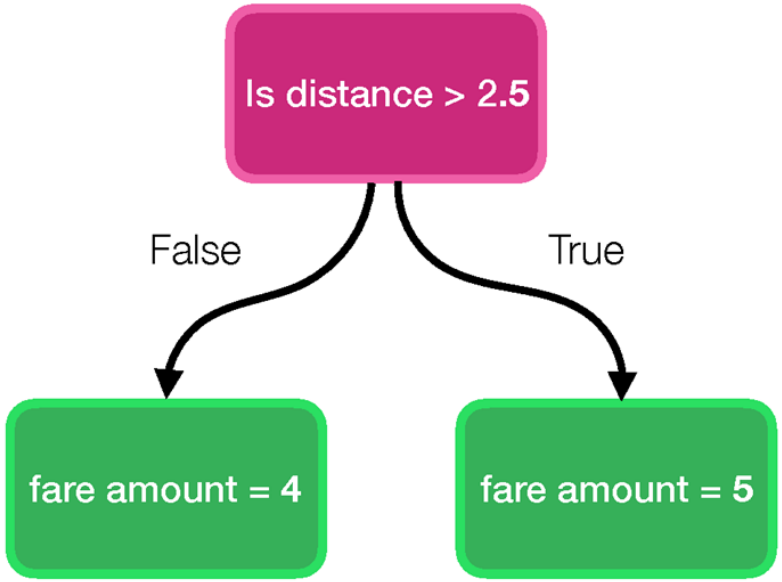
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = 1$$



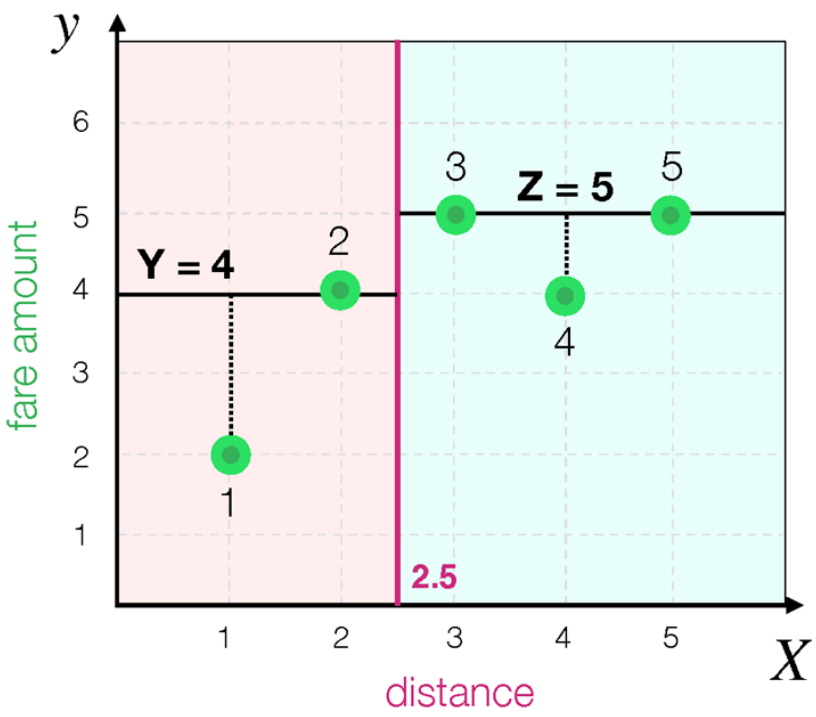


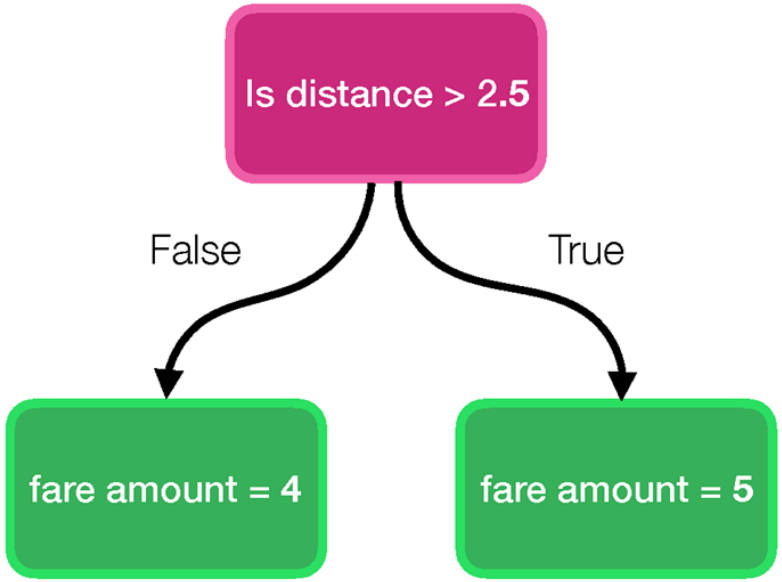
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = 1$ so, if **X** = **2.5**, **Y** = **4** and **Z** = **5**, MSE is **1**





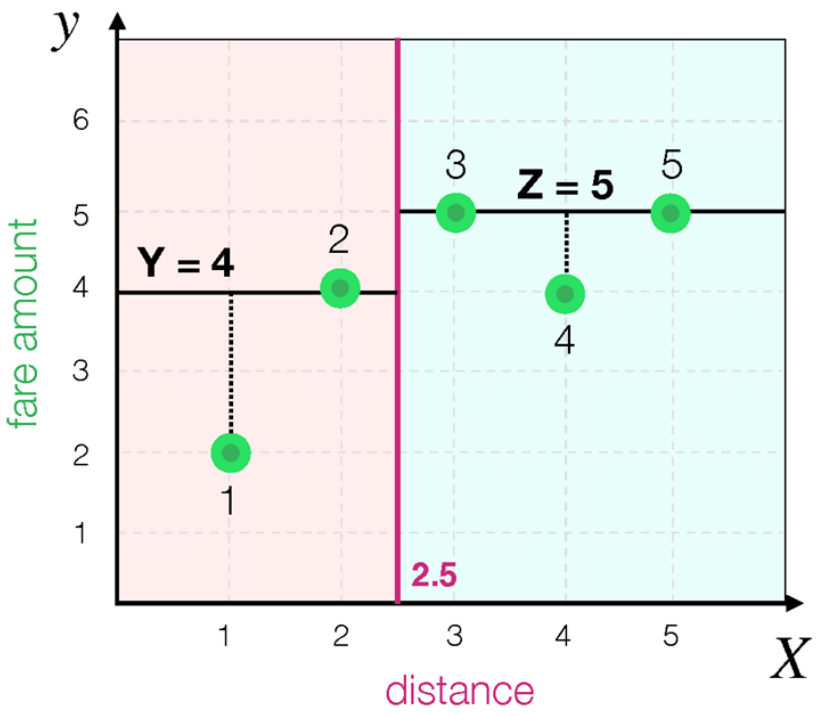
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

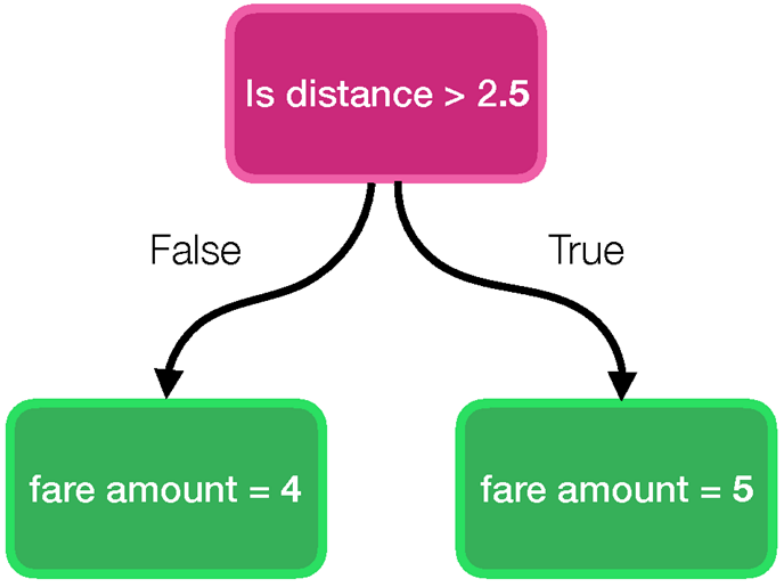
Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = 1$ so, if **X** = **2.5**, **Y** = **4** and **Z** = **5**, MSE is **1**

Can we find better **Y** and **Z**?



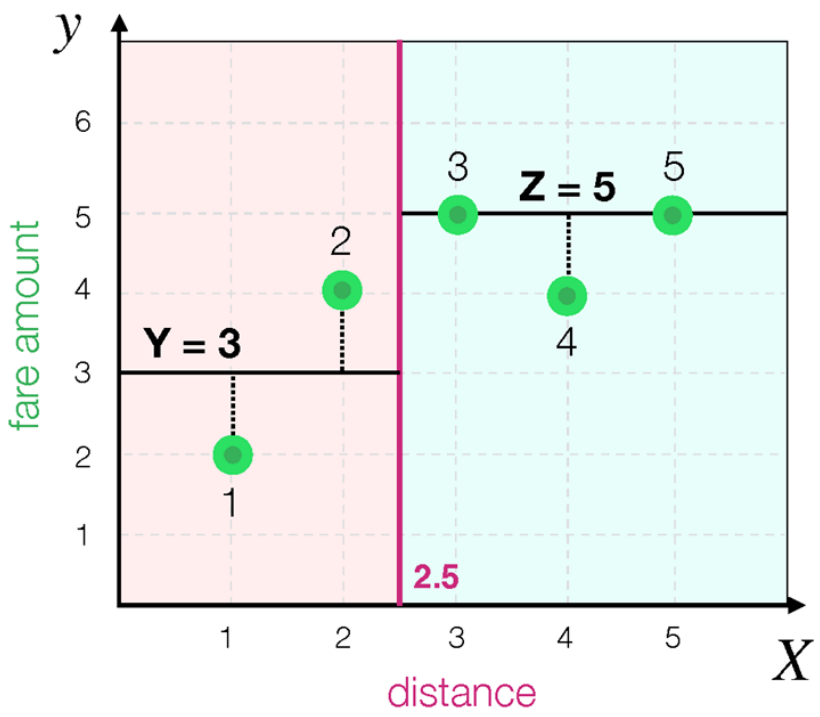


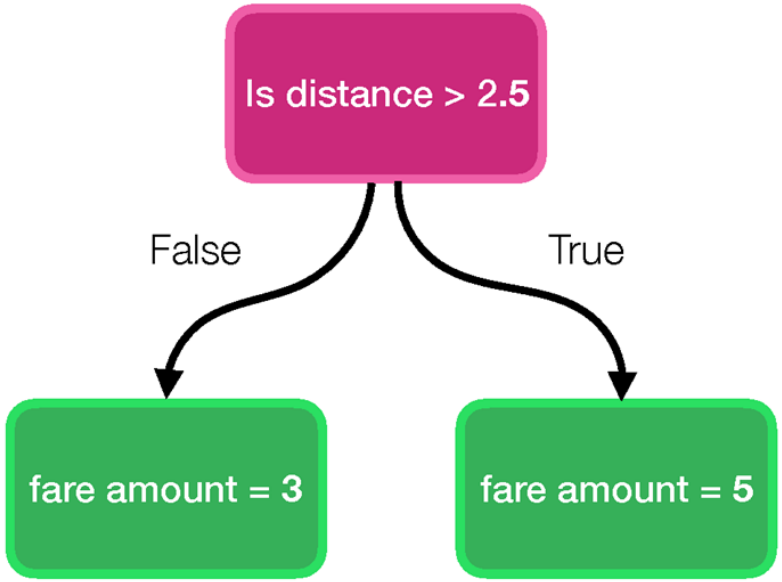
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + (y_3 - \hat{y}_3)^2 + (y_4 - \hat{y}_4)^2 + (y_5 - \hat{y}_5)^2}{5}$$



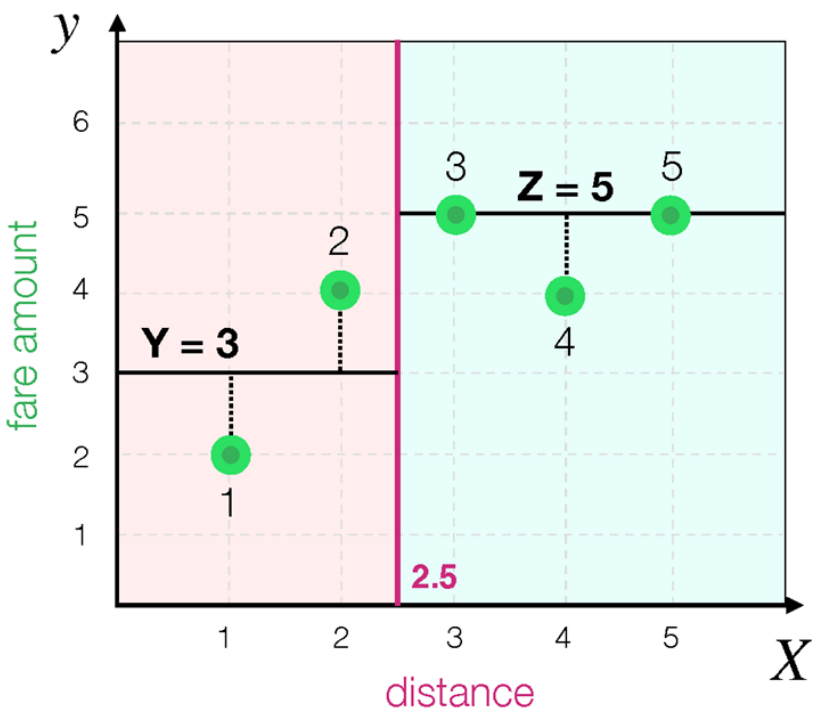


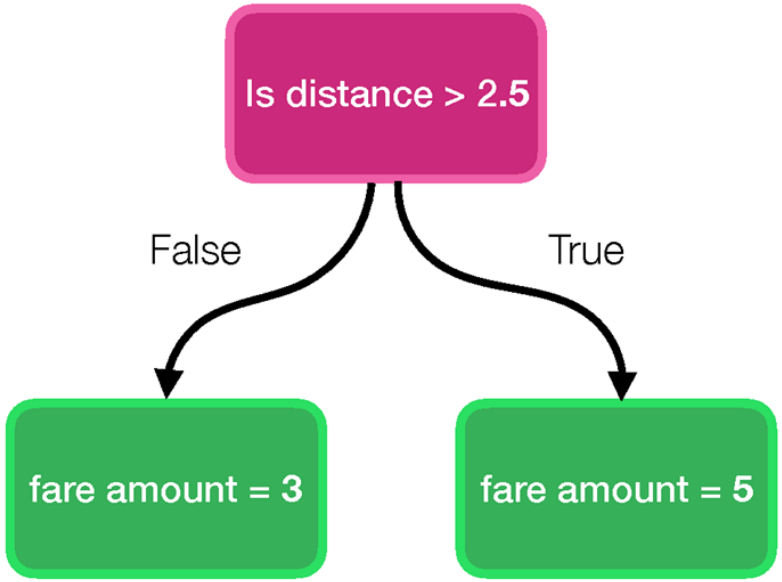
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(2-3)^2 + (4-3)^2 + (5-5)^2 + (4-5)^2 + (5-5)^2}{5}$$



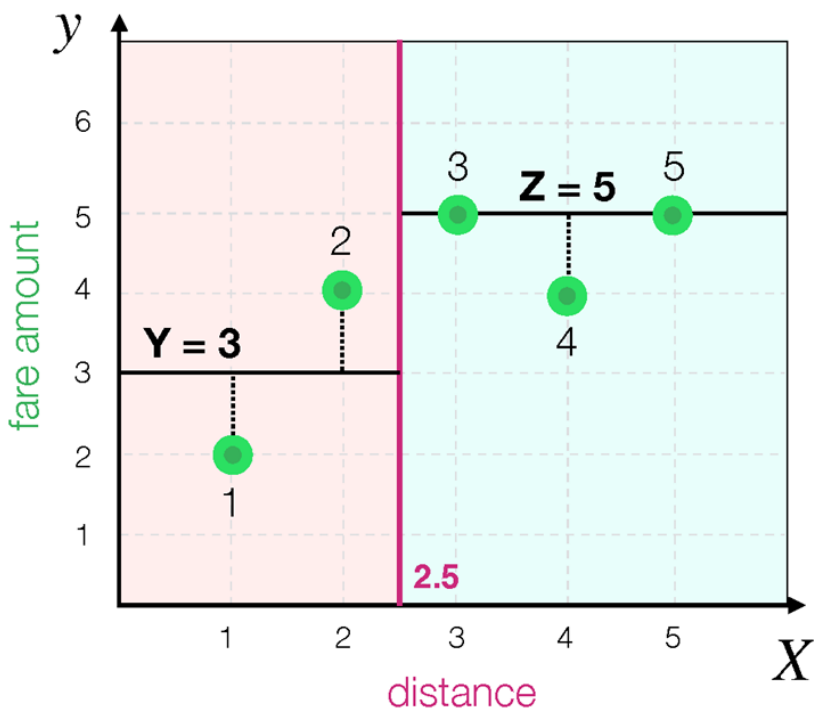


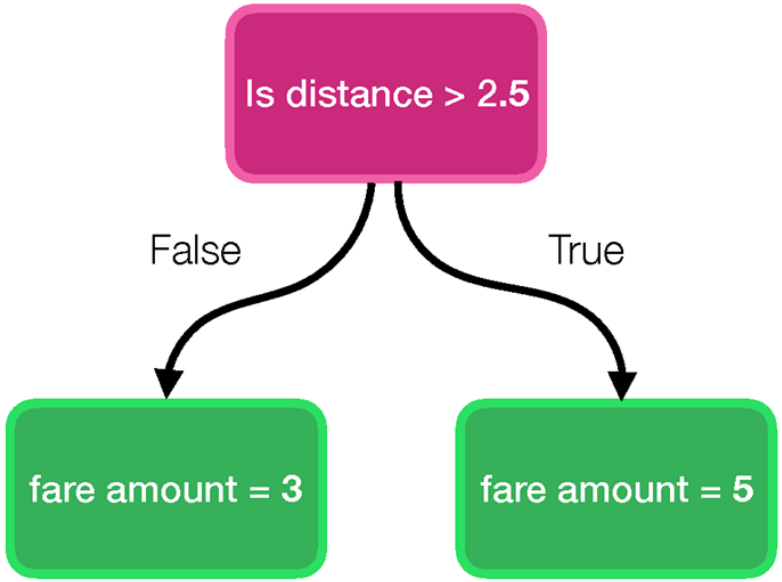
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1+1+0+1+0}{5}$$



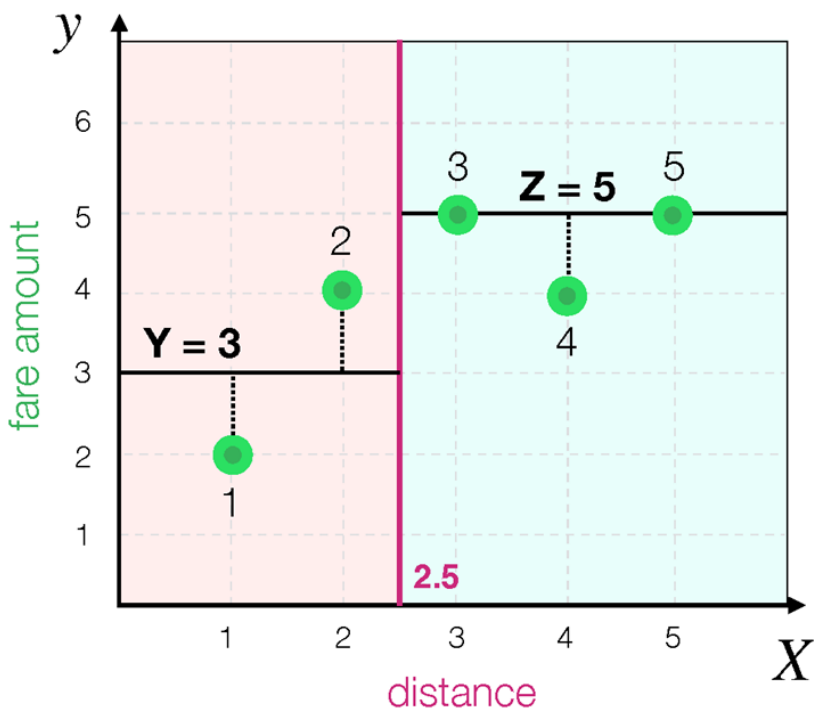


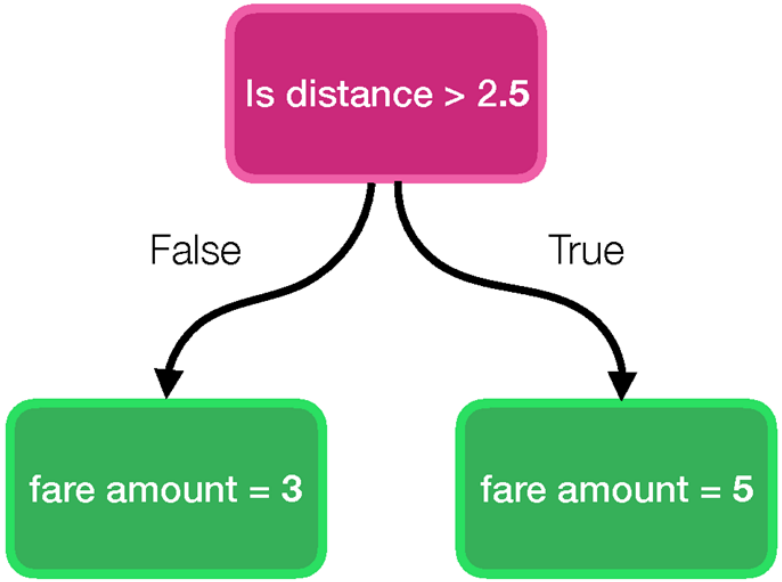
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{3}{5} = 0.6$$





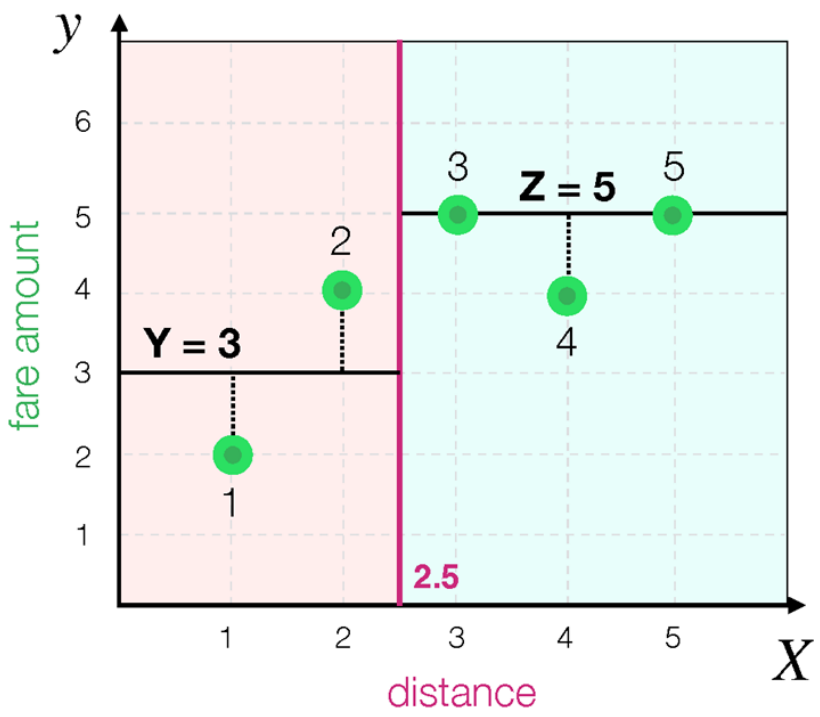
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

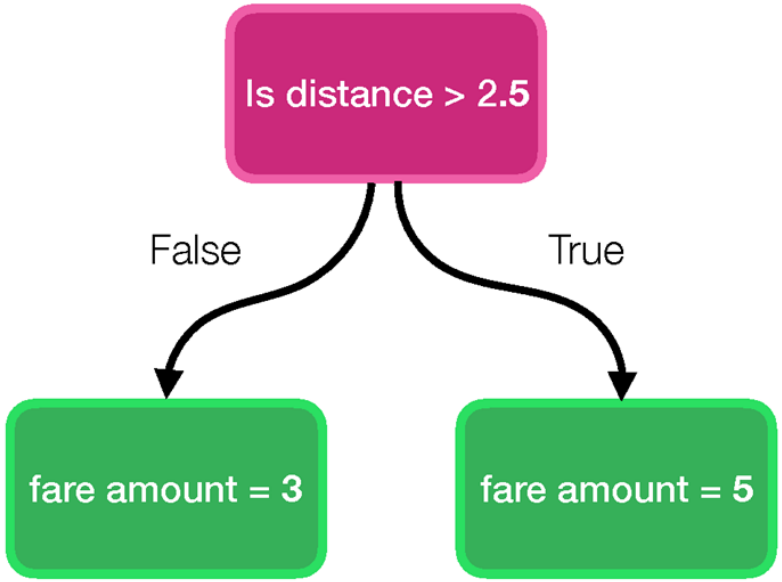
Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{3}{5} = 0.6$$

so, if **X** = **2.5**, **Y** = **3** and **Z** = **5**, MSE is **0.6**



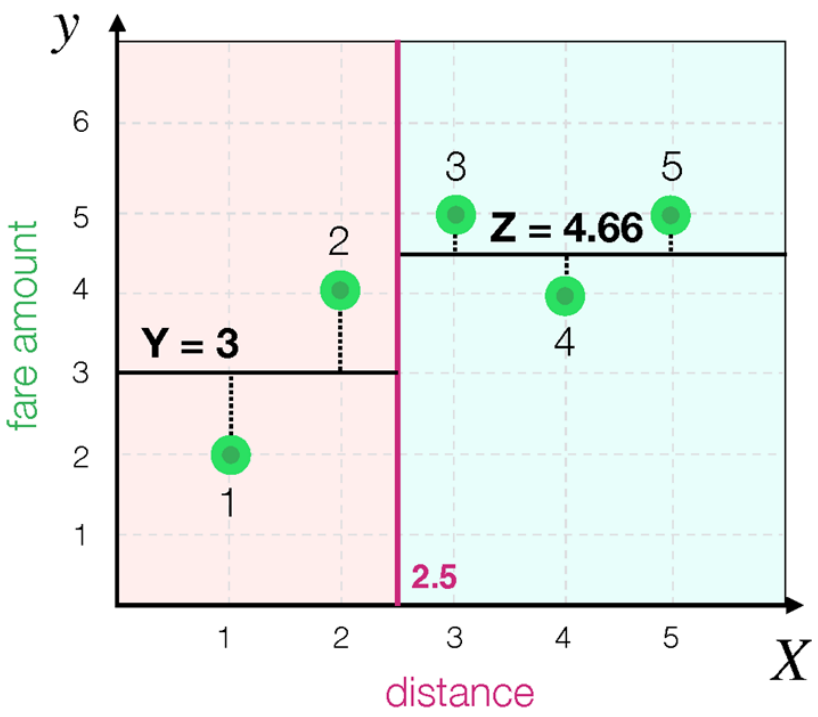


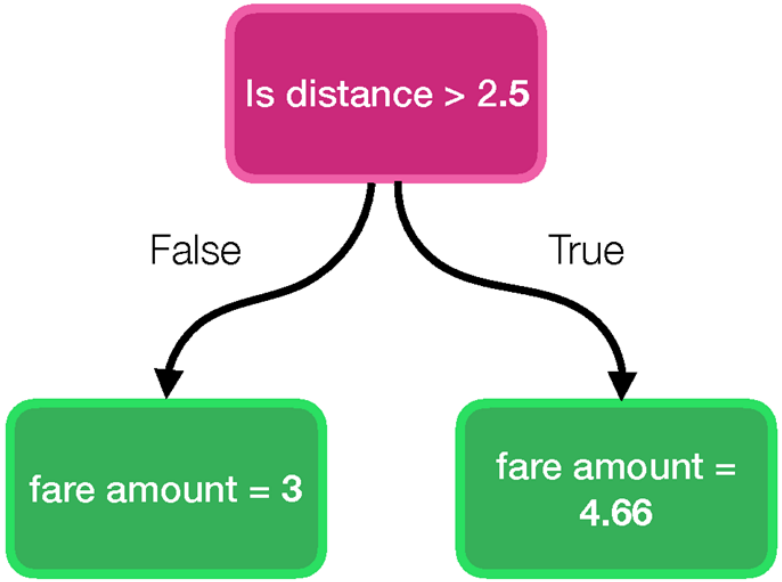
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{(2-3)^2 + (4-3)^2 + (5-4.66)^2 + (4-4.66)^2 + (5-4.66)^2}{5}$$



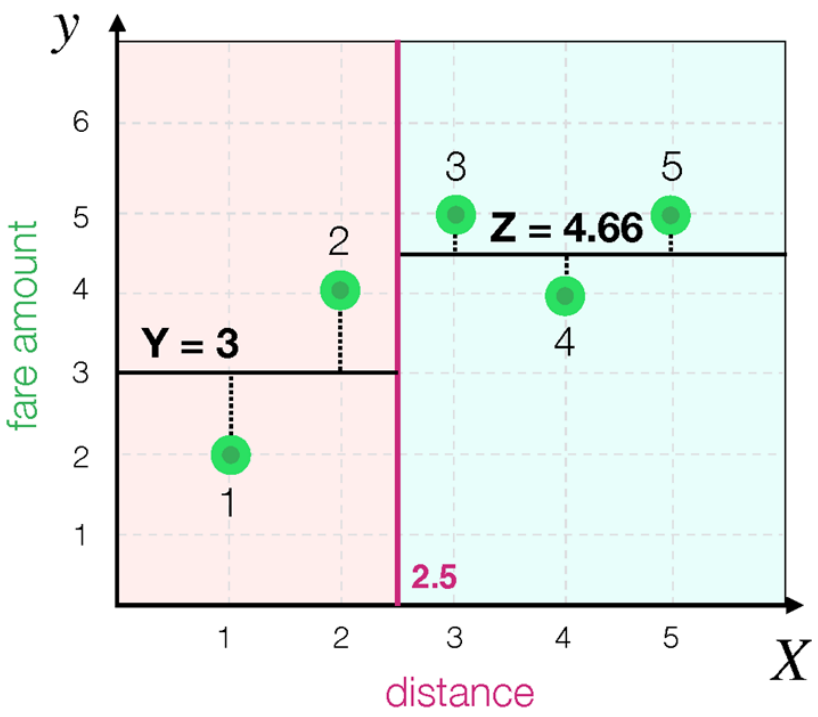


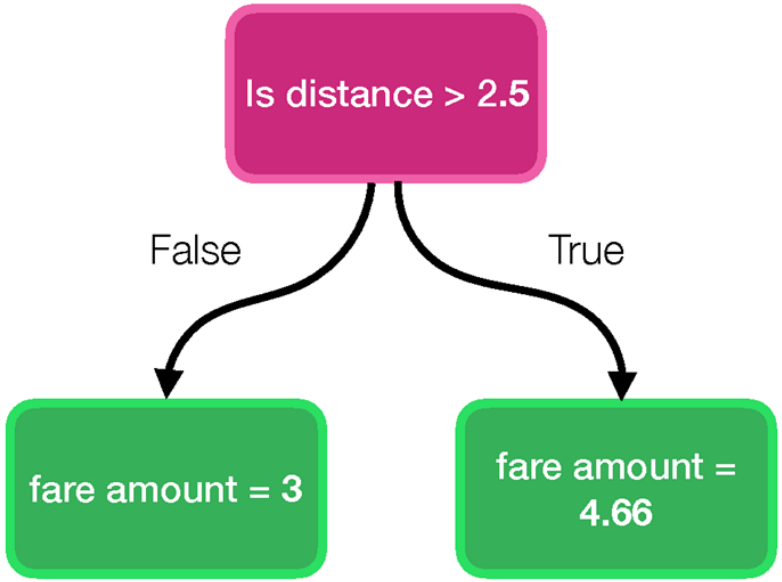
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1 + 1 + 0.12 + 0.43 + 0.12}{5}$$





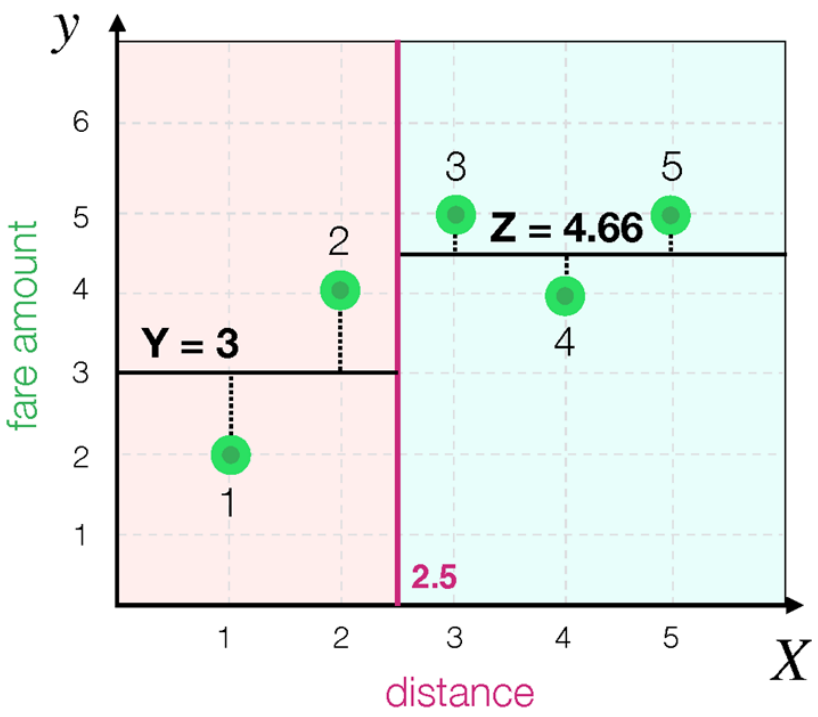
What are most reasonable values for **Y** and **Z (that minimise total MSE)**?

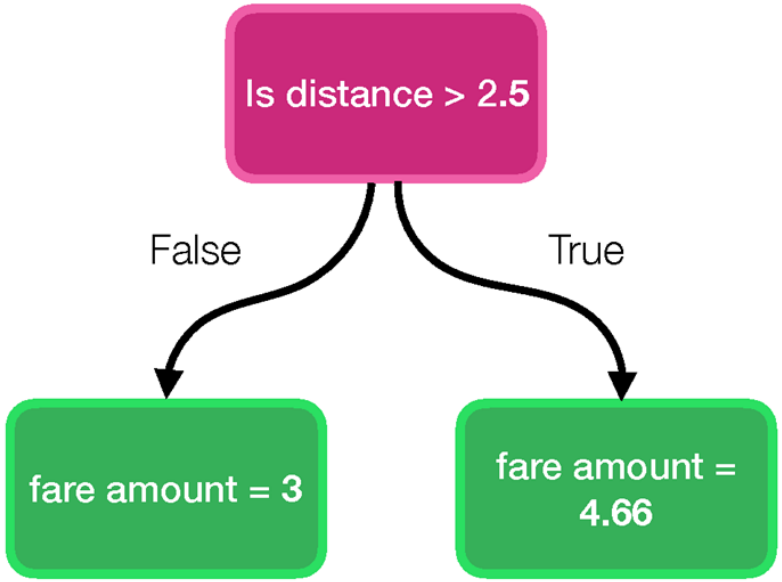
Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{2.67}{5} = 0.53$$

so, if **Y = 3** and **Z = 4.66**,
MSE is **smallest**





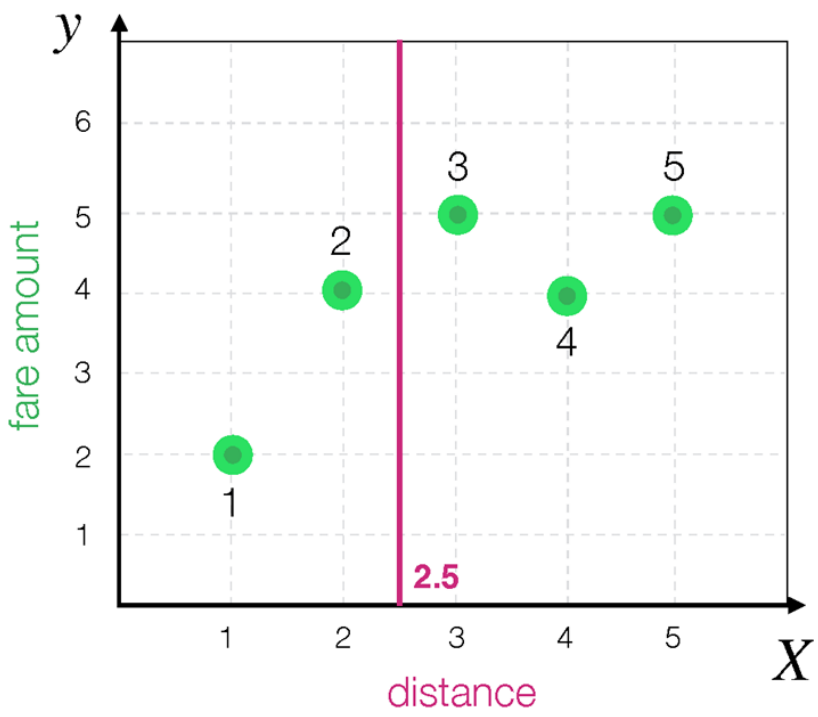
Are we happy?

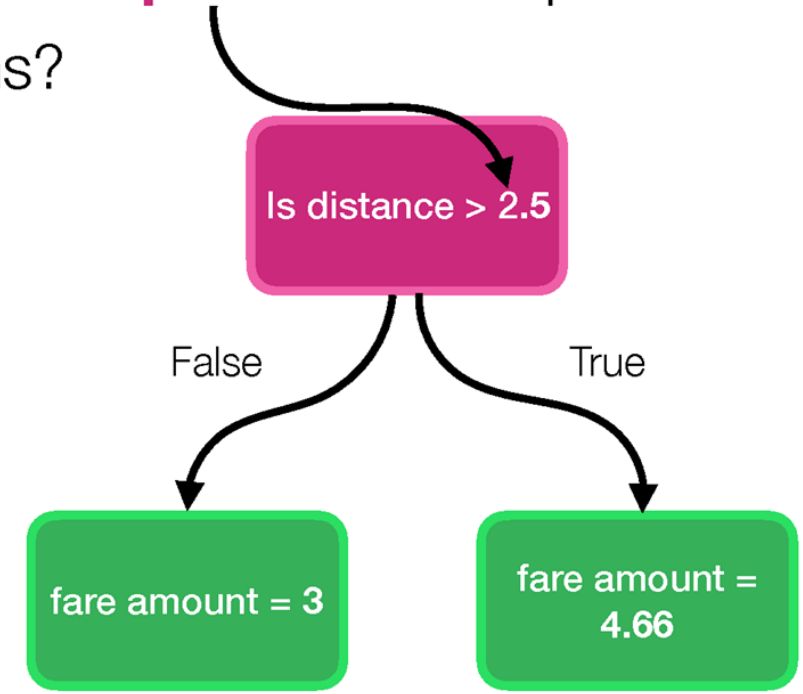
Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Hold on, how did we choose this **split** on the first place?

Maybe there are better options?

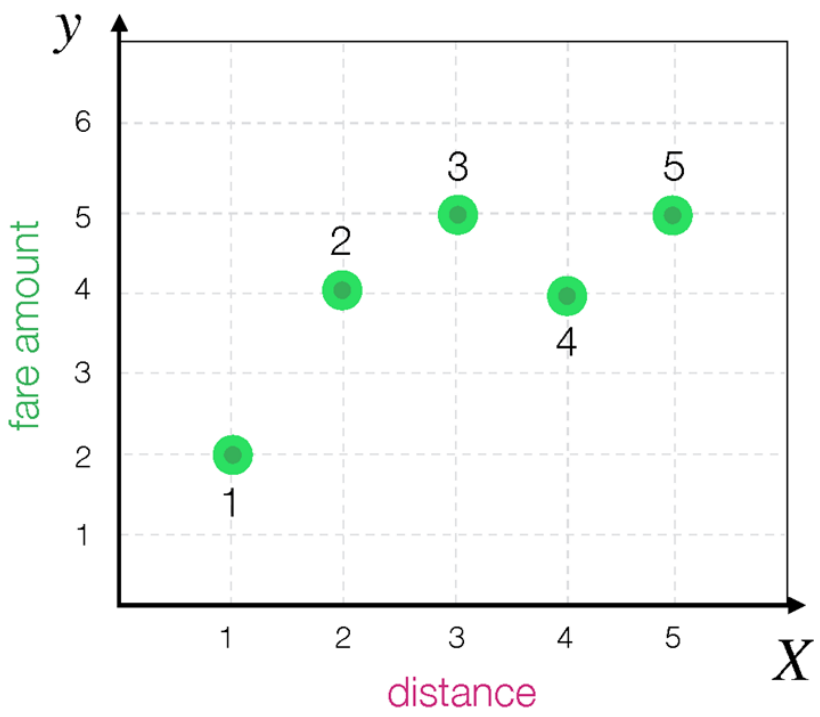


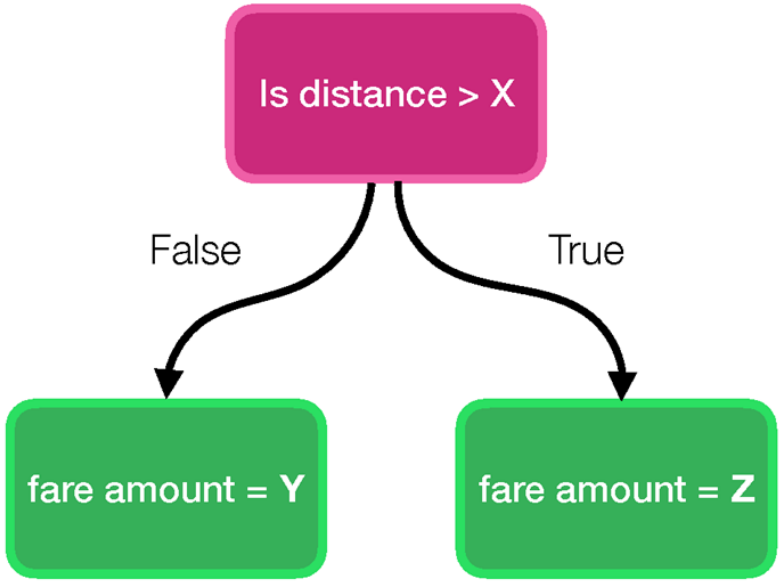


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

What are the possible **split** options in this case?

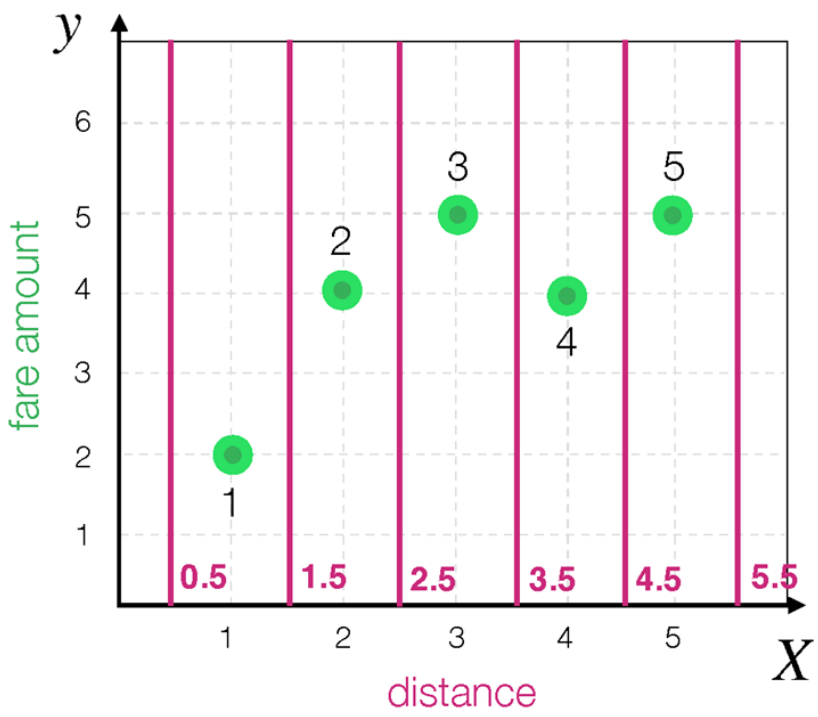


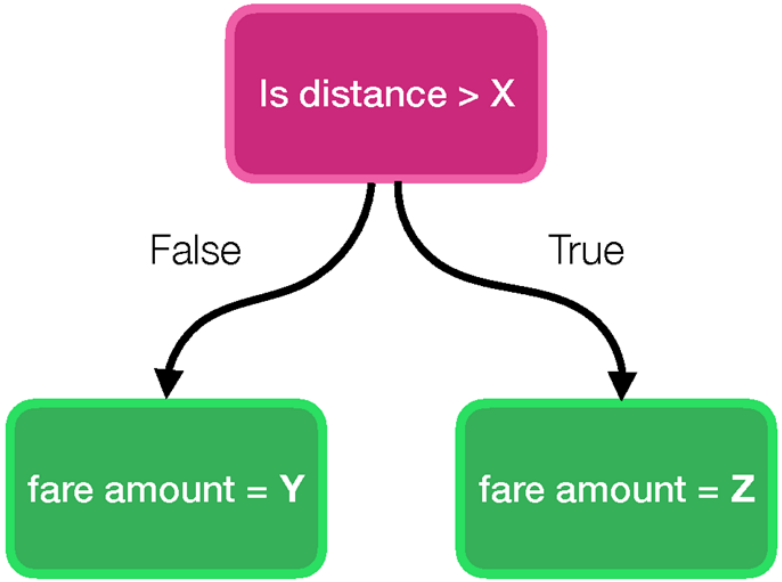


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

What are the possible **split** options in this case?

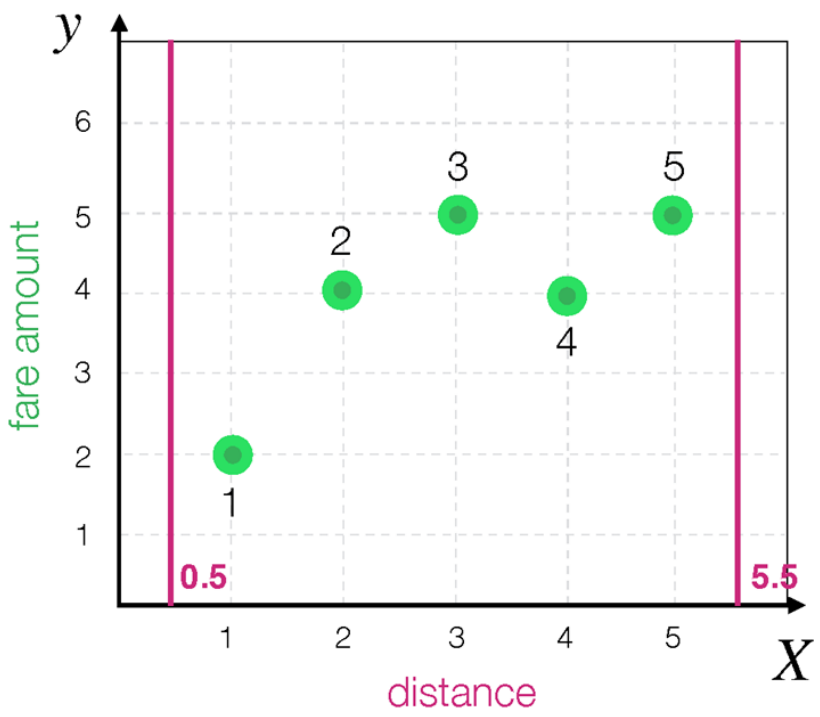


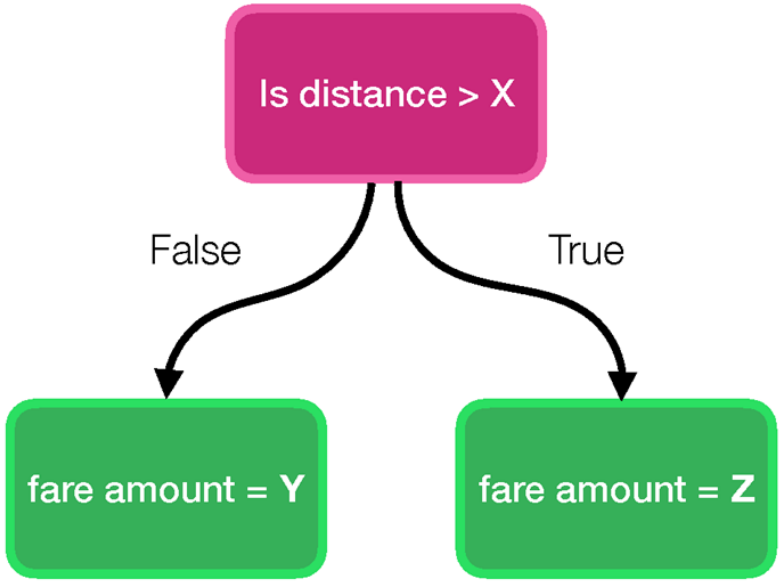


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Are these meaningful?

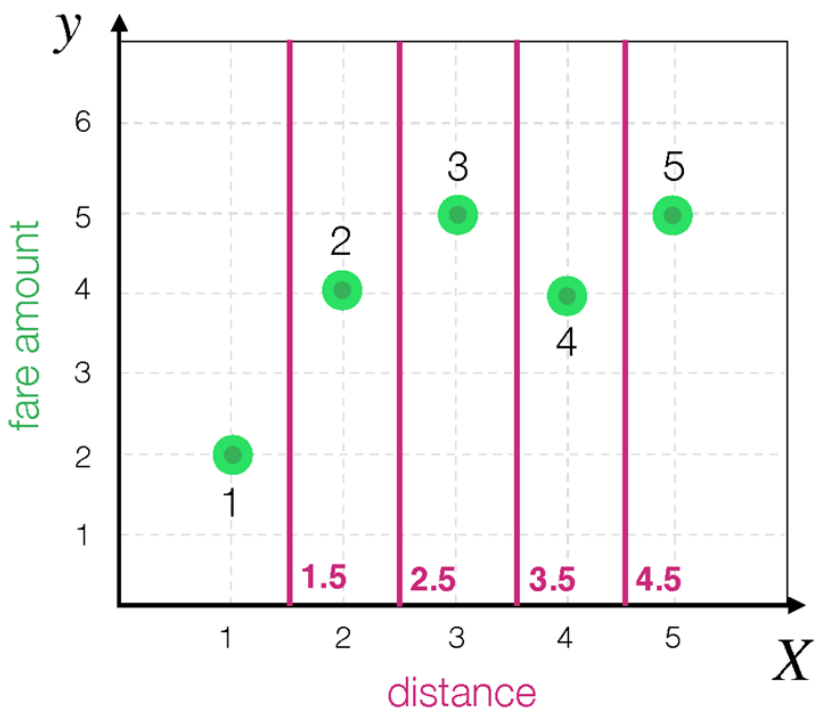


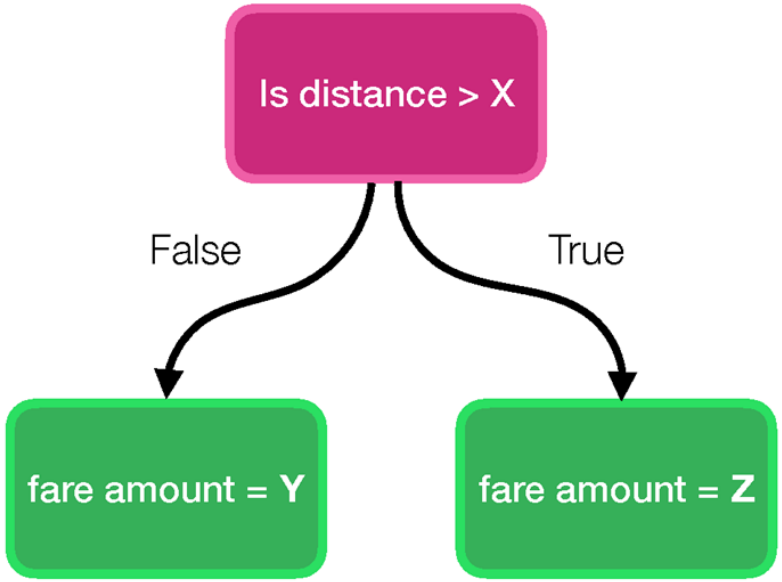


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

How to compare remaining?

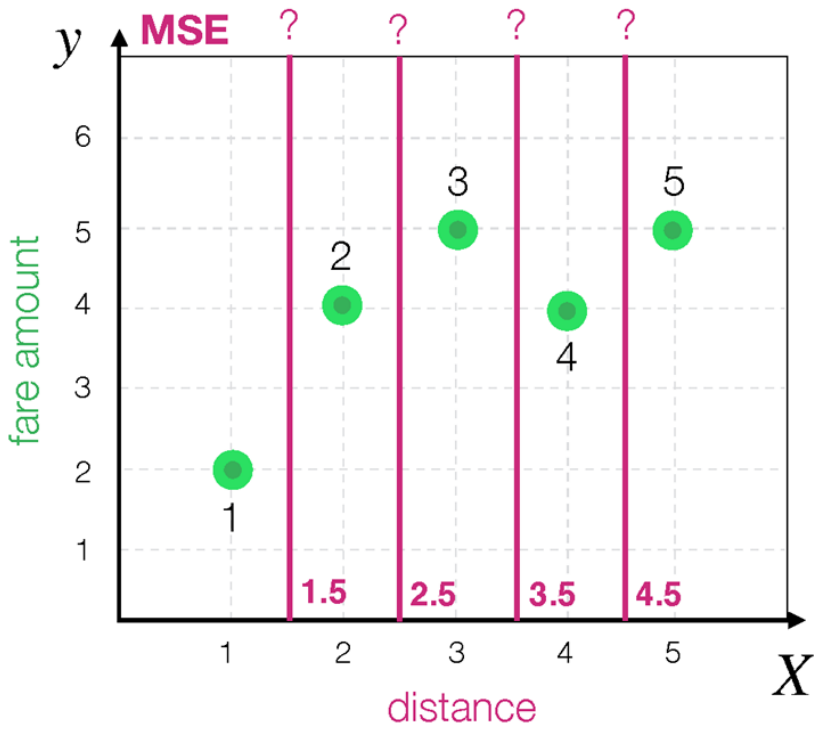


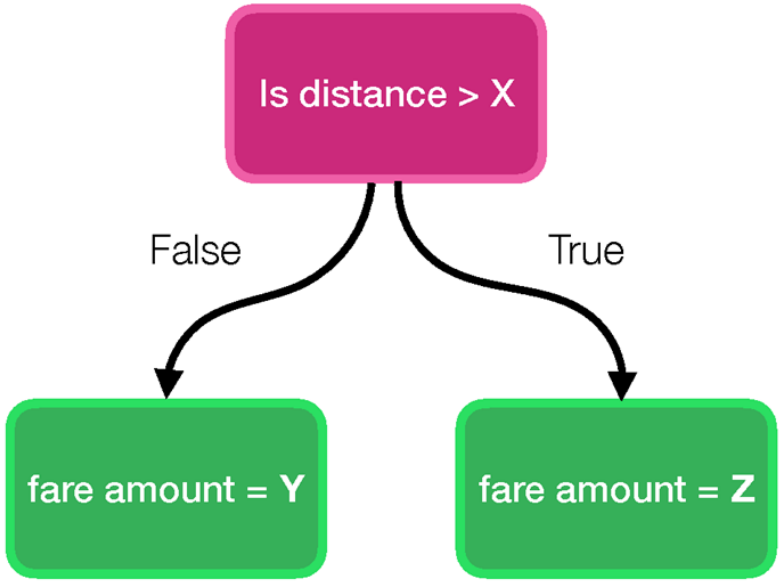


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

How to compare remaining?
For each one we can compute MSE

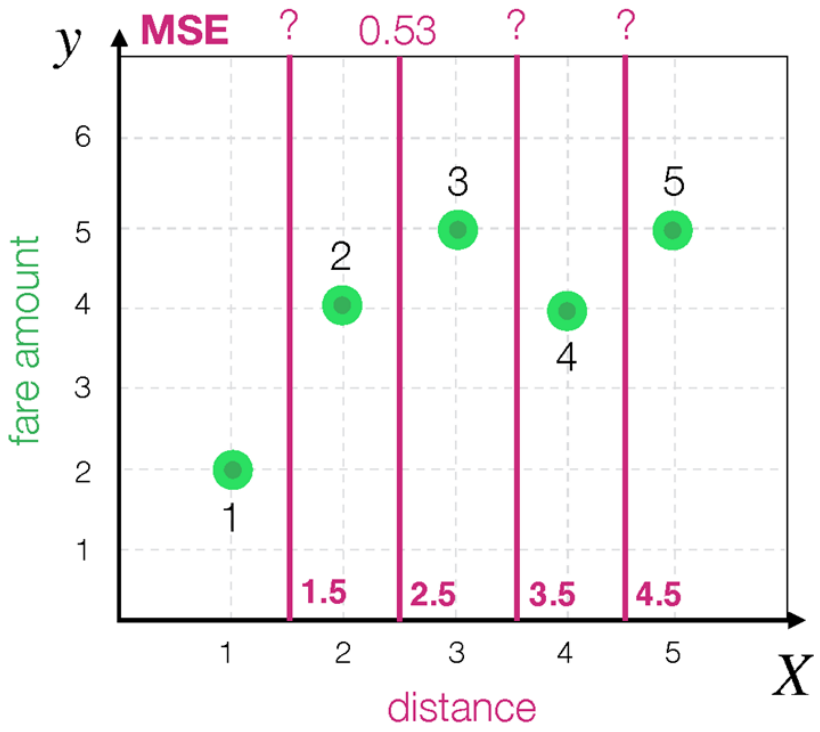


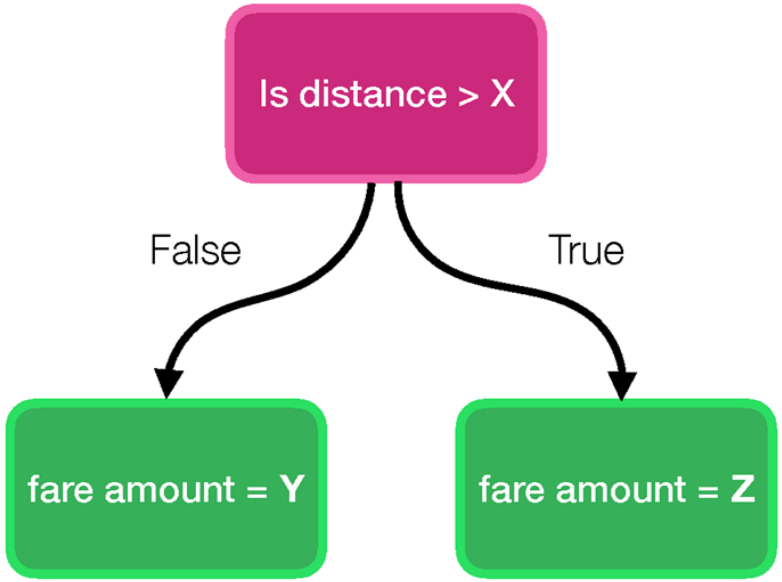


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

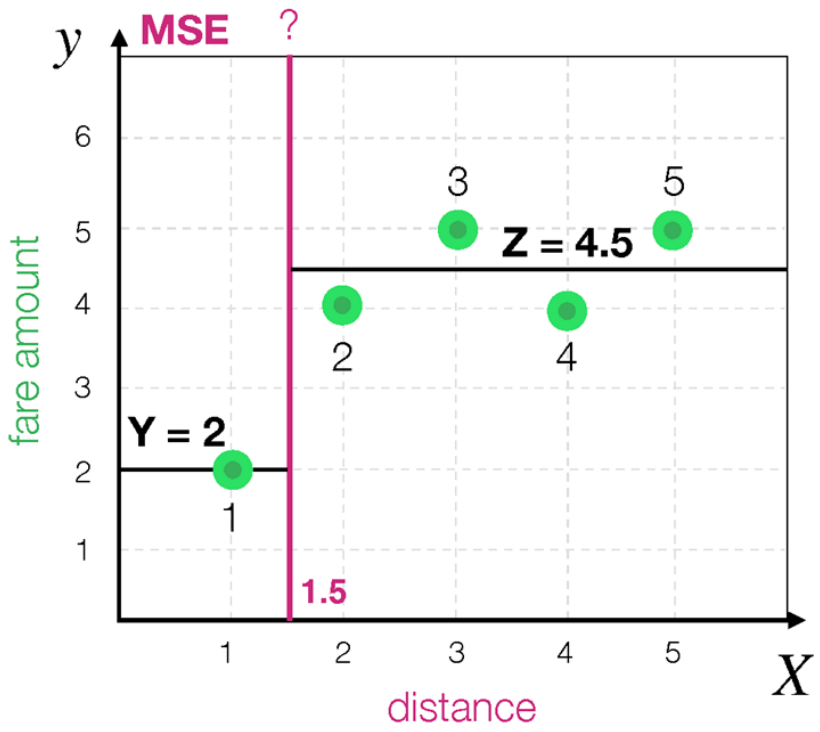
How to compare remaining?
For each one we can compute MSE

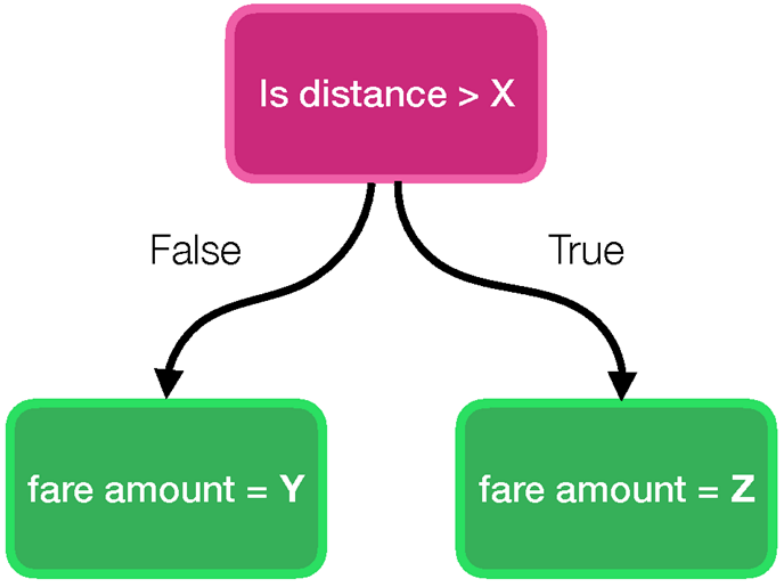




Source: Dima Fishman, Supervised Machine Learning

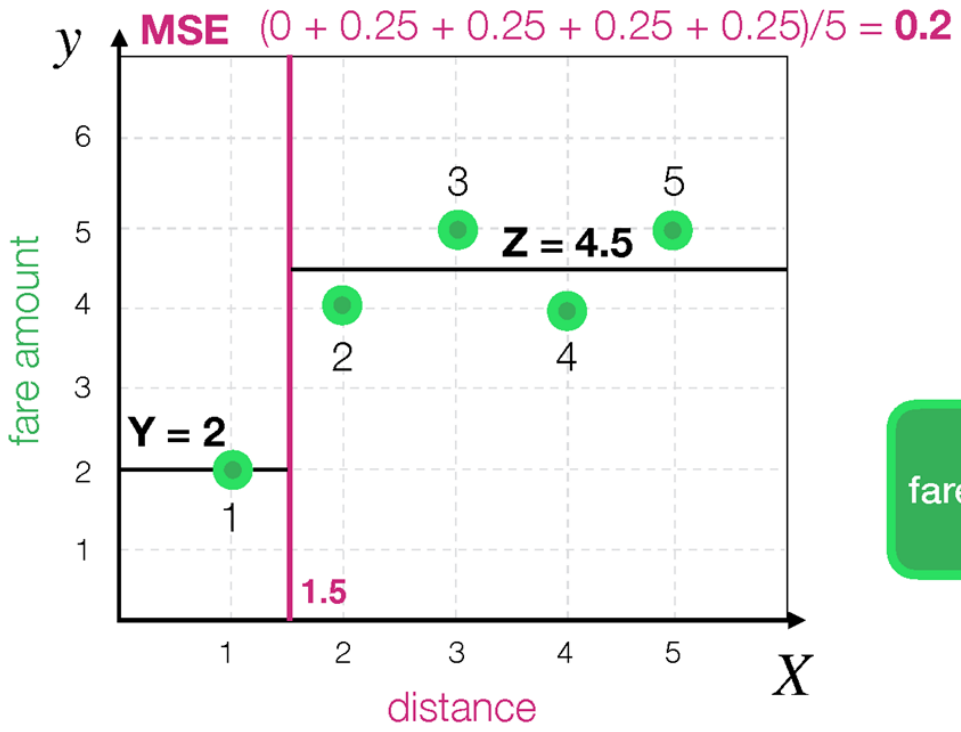
Decision Tree Algorithm

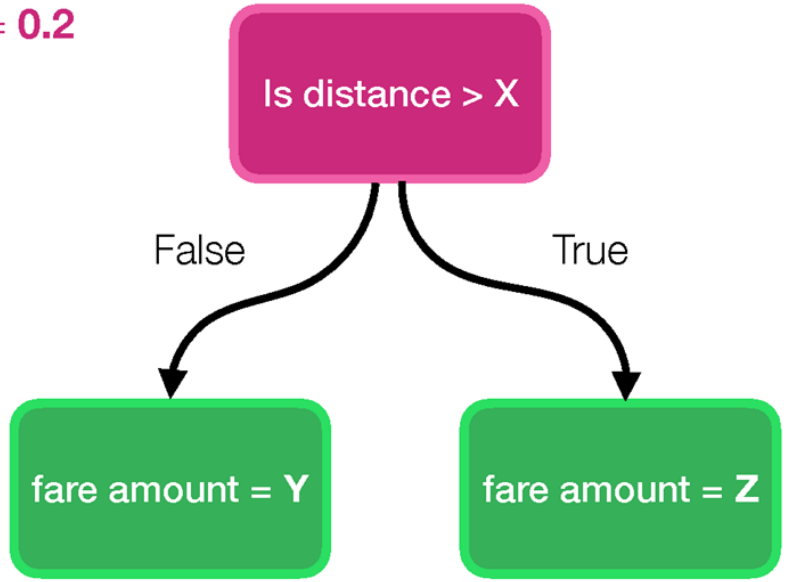




Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

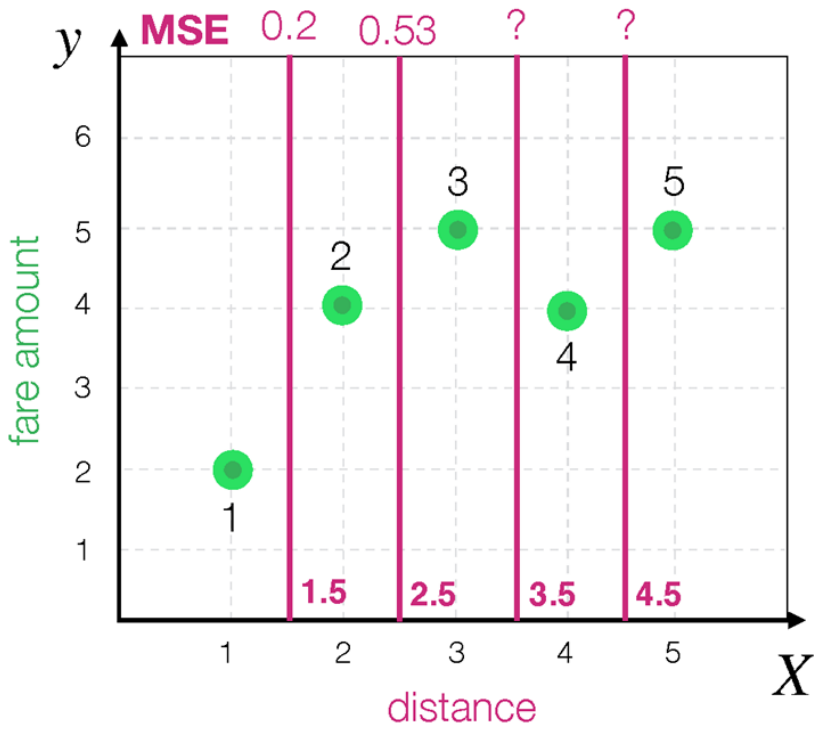


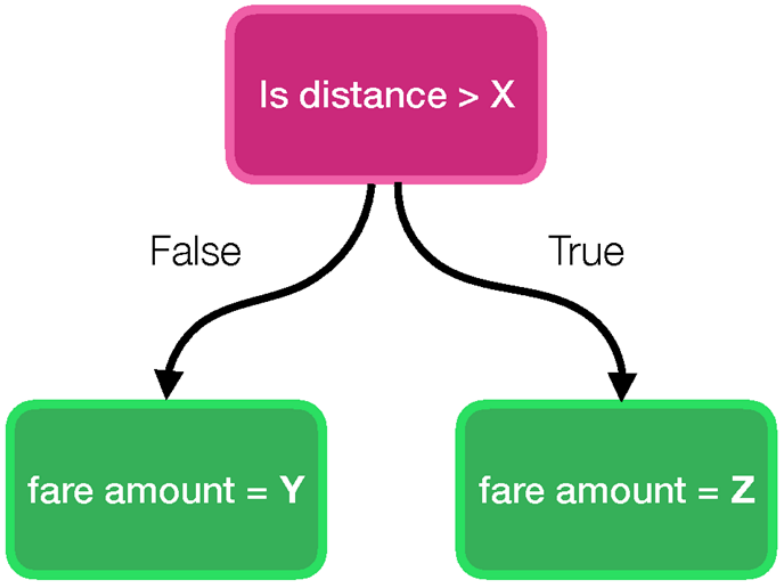


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

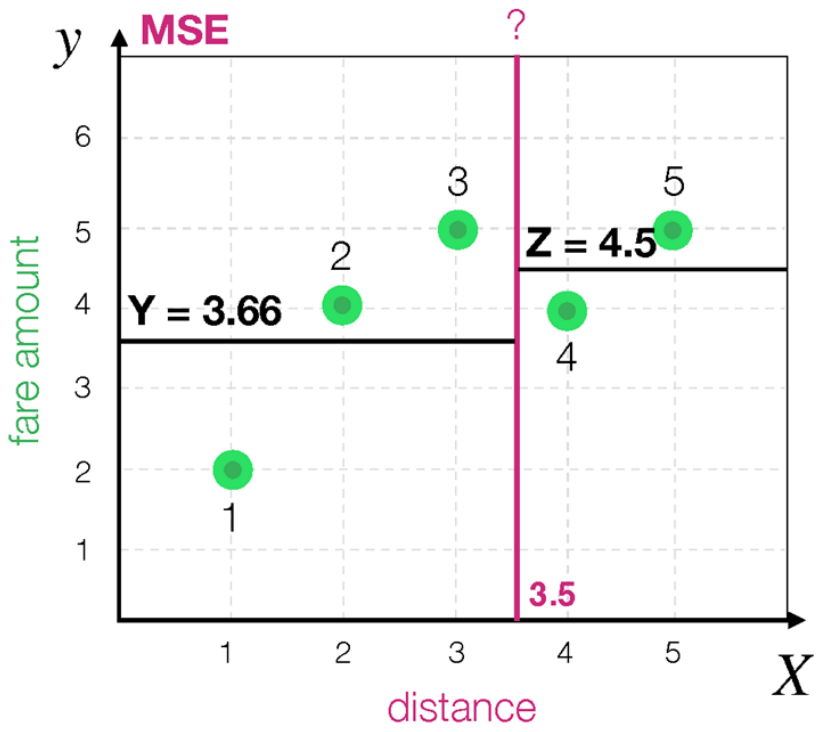
How to compare remaining?
For each one we can compute MSE

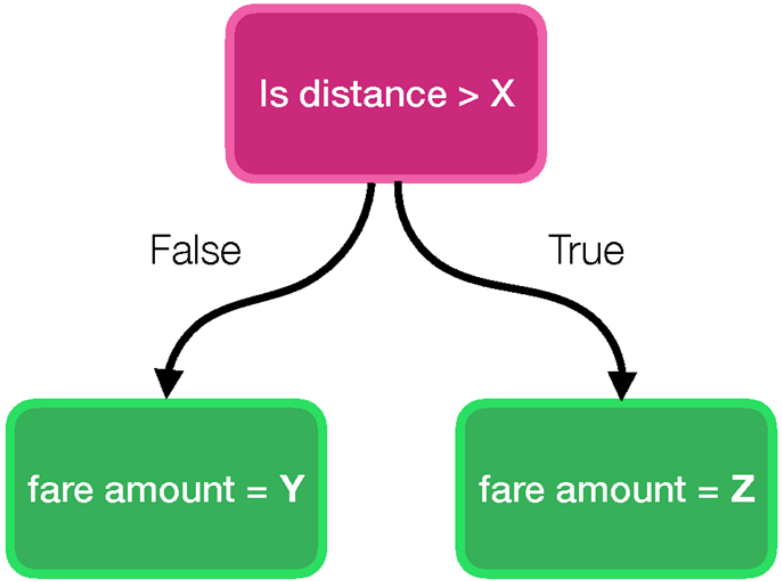




Source: Dima Fishman, Supervised Machine Learning

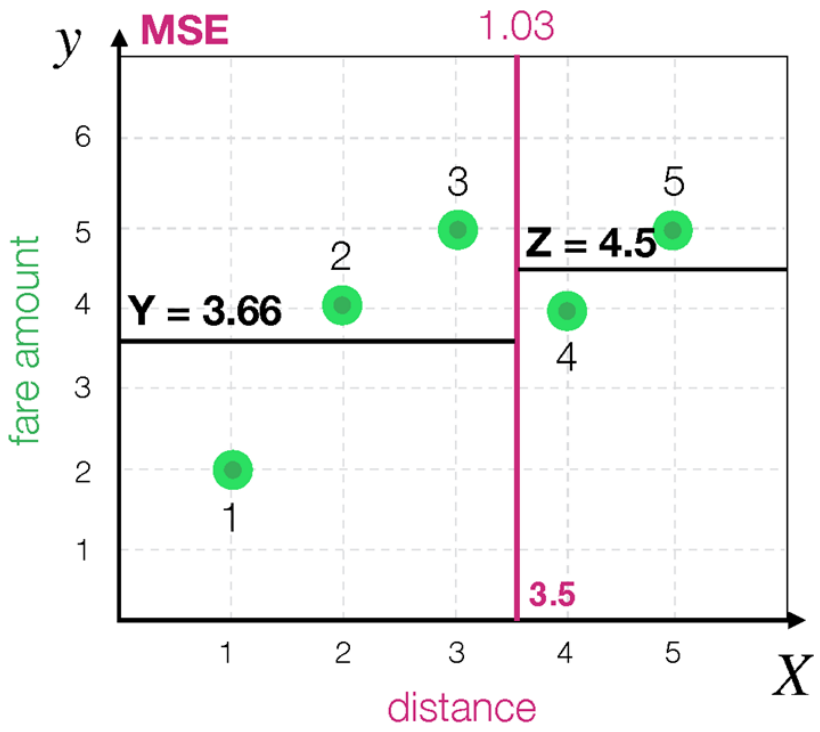
Decision Tree Algorithm

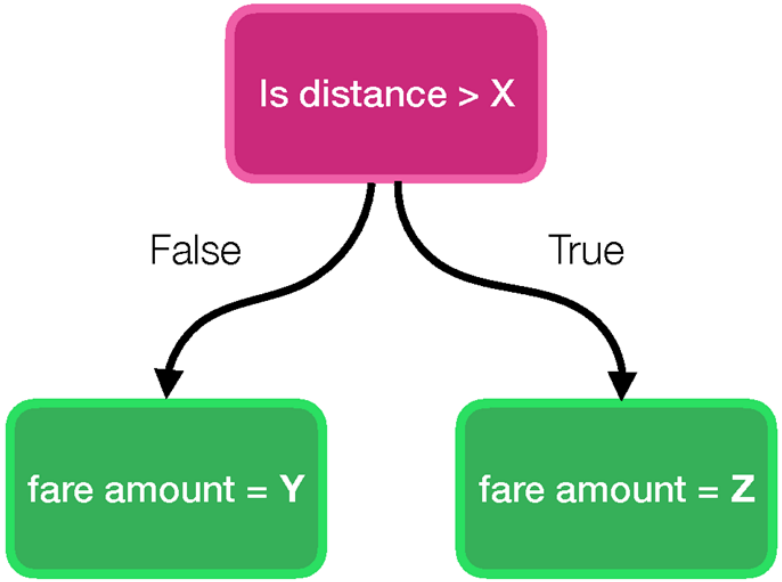




Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

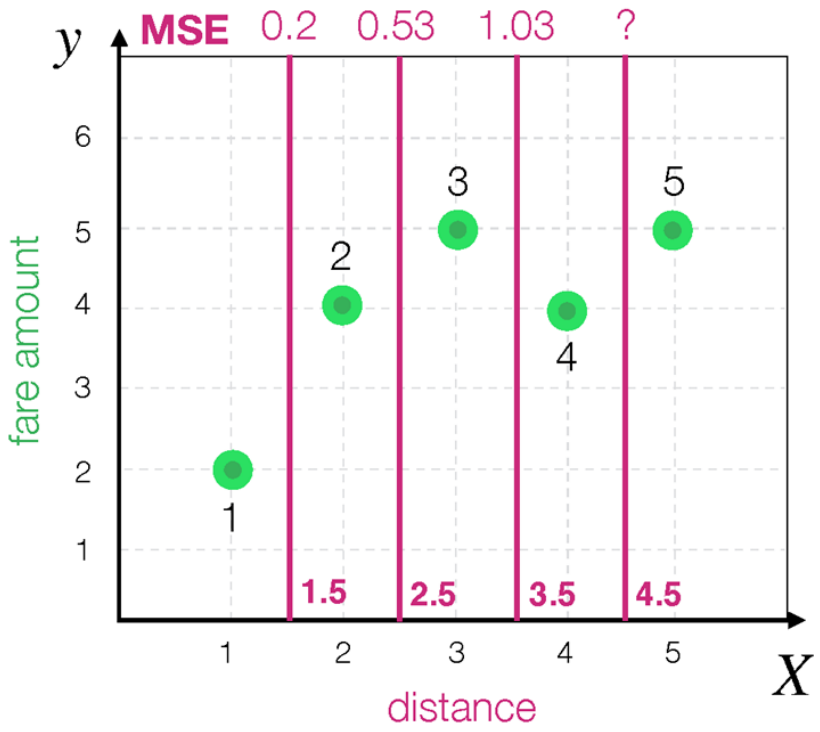


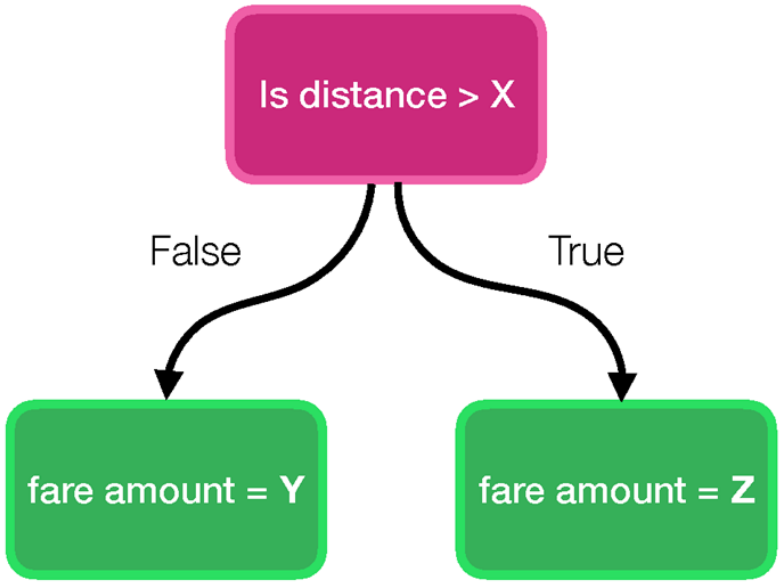


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

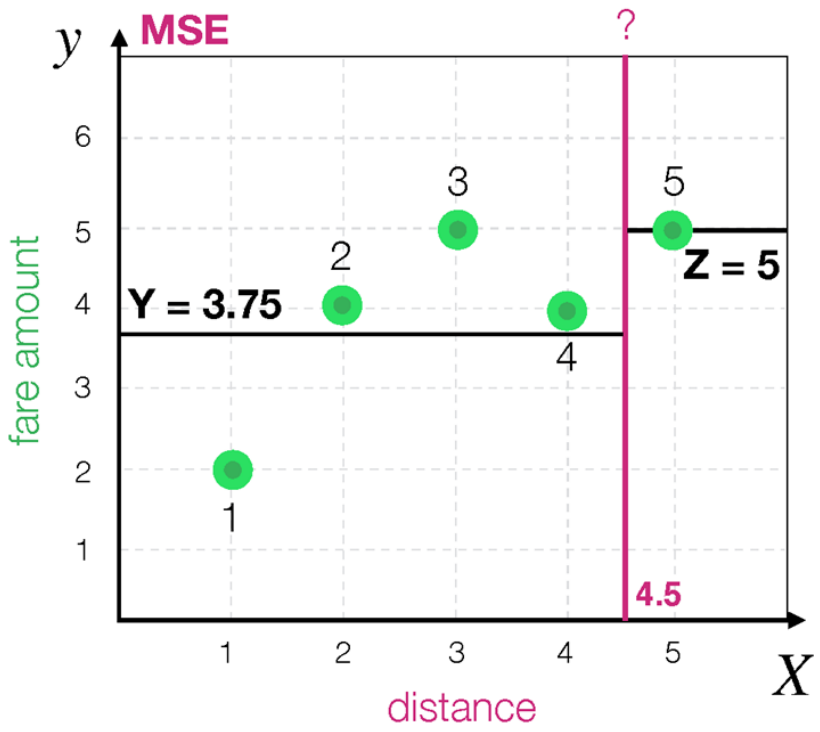
How to compare remaining?
For each one we can compute MSE

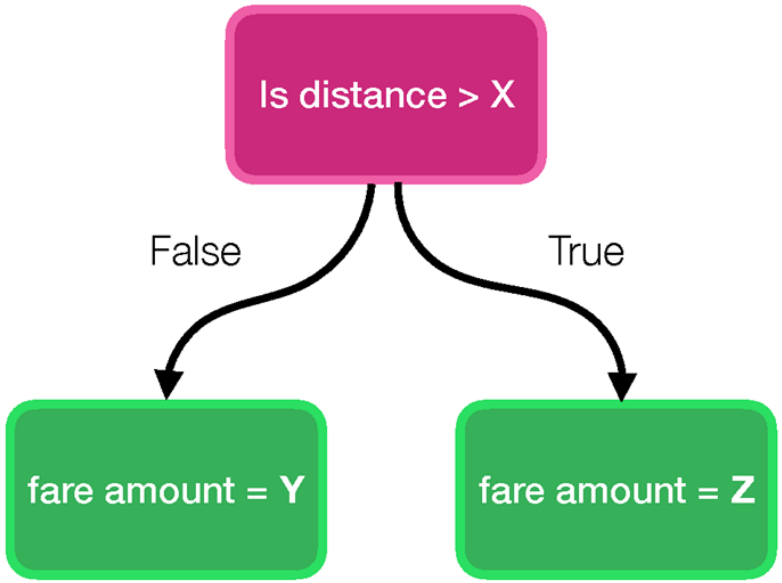




Source: Dima Fishman, Supervised Machine Learning

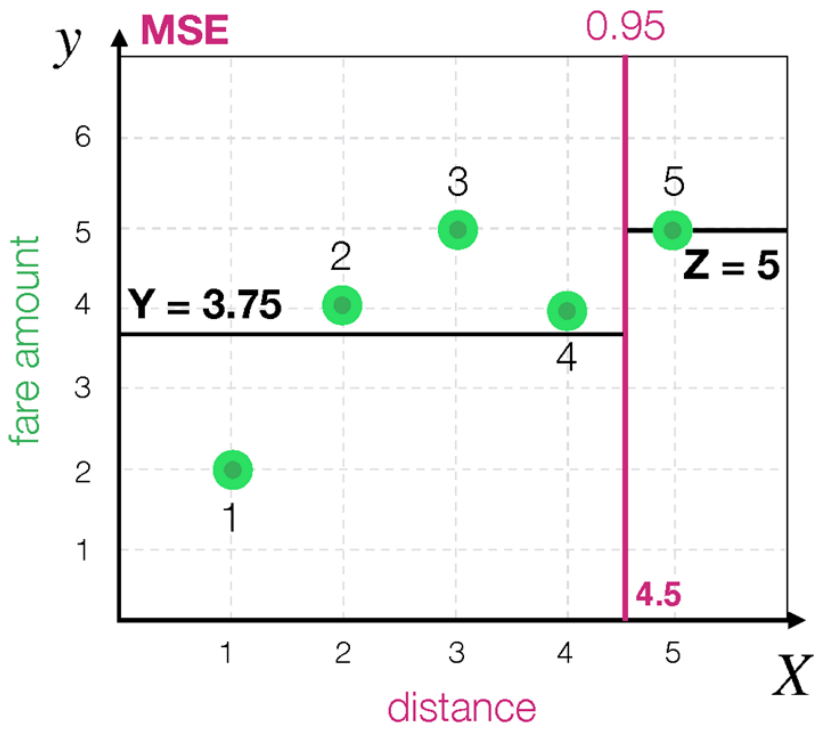
Decision Tree Algorithm

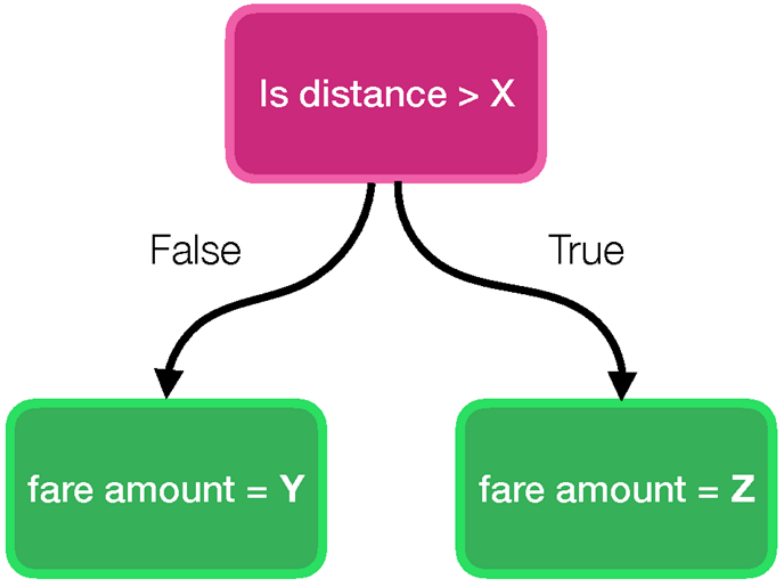




Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

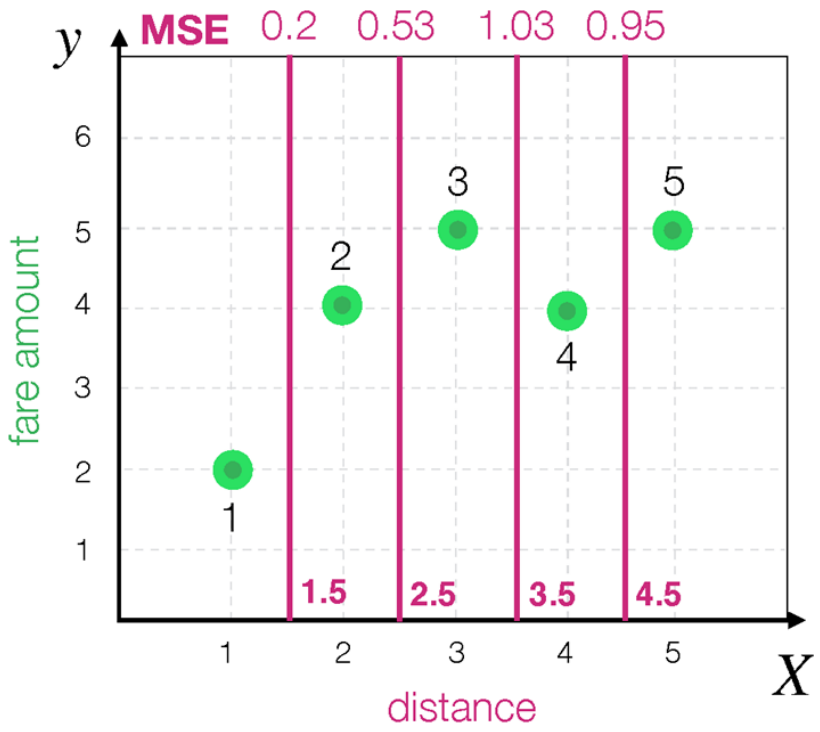


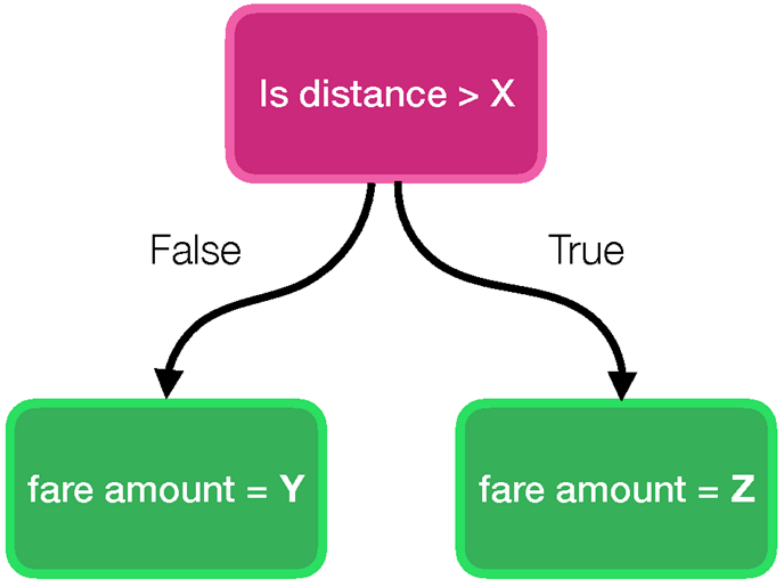


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

How to compare remaining?
For each one we can compute MSE

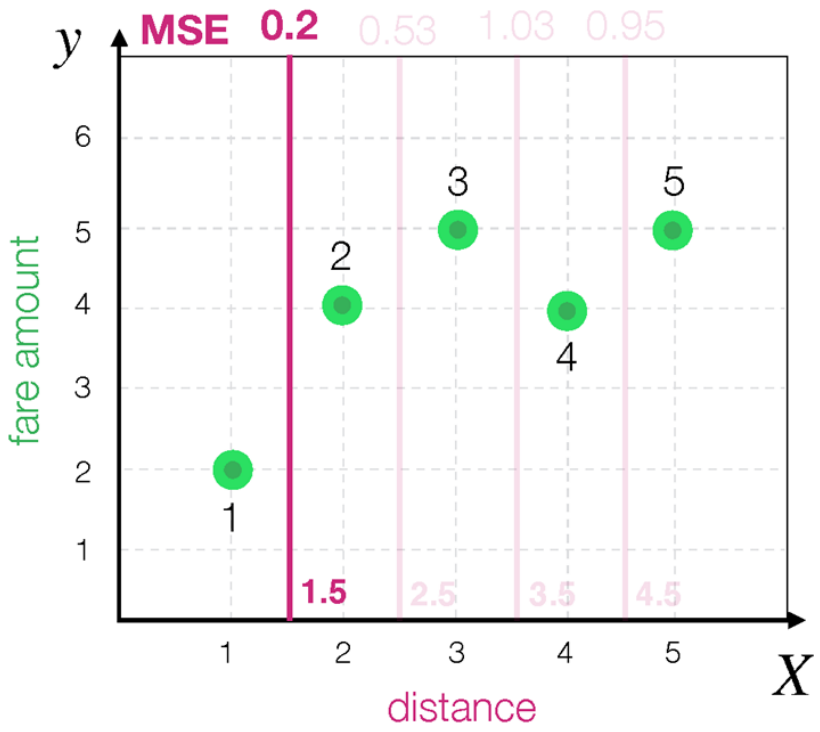


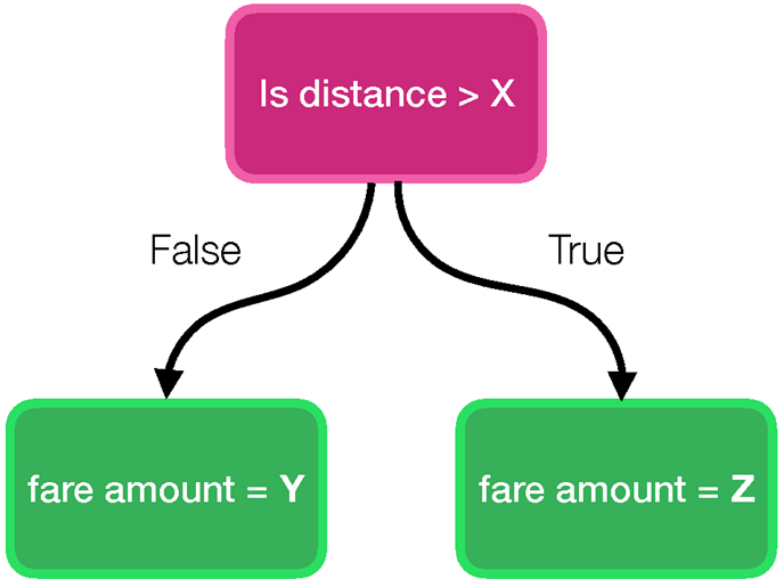


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

We choose the **split** that minimises total MSE

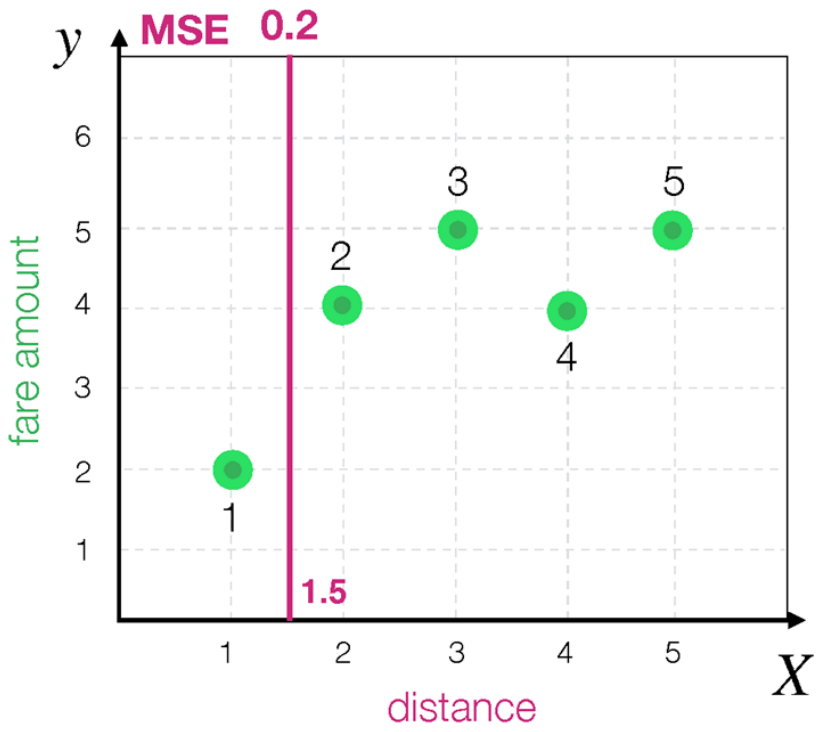


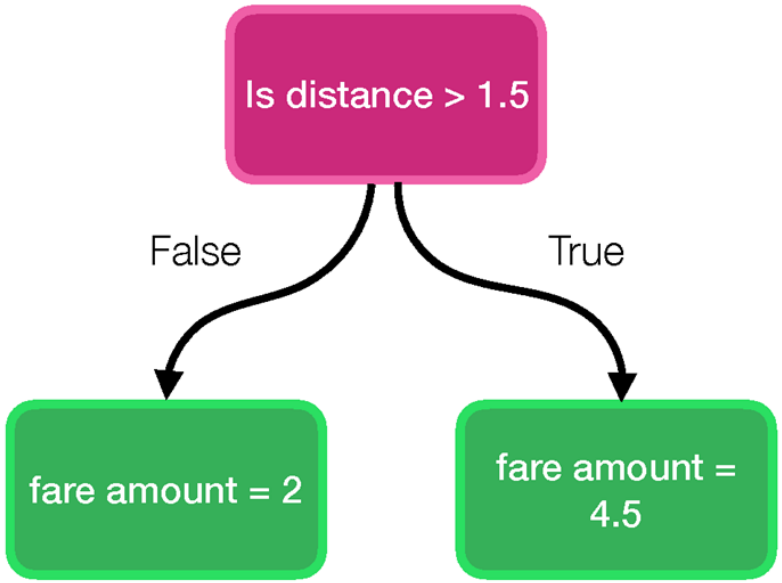


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Thus, the resulting tree:

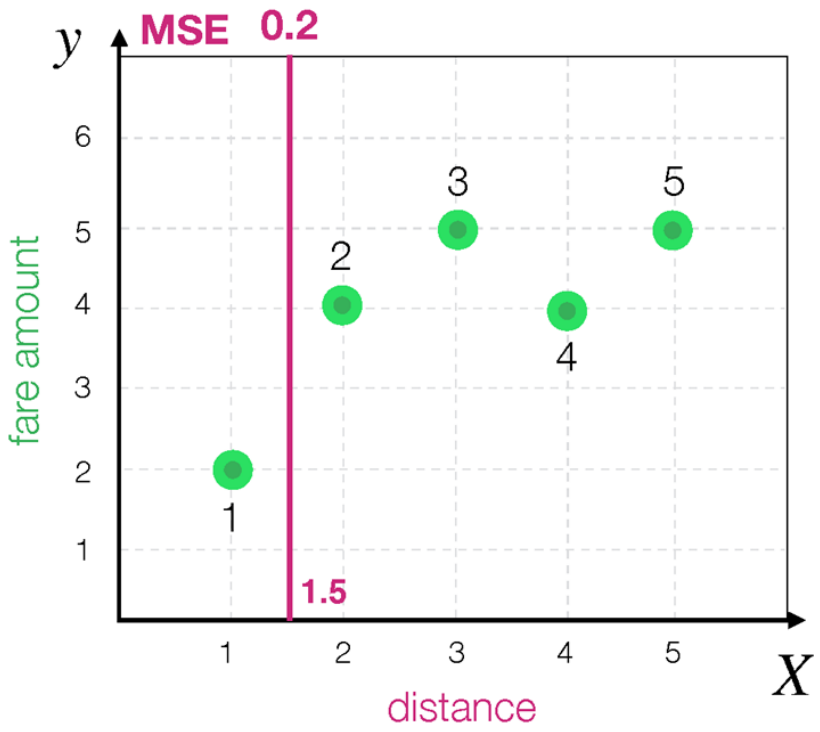


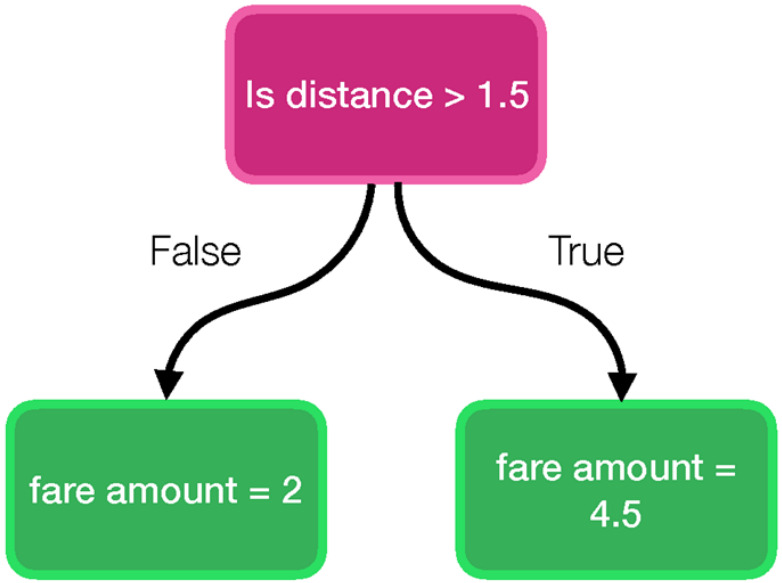


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Can we make our decision tree more accurate?

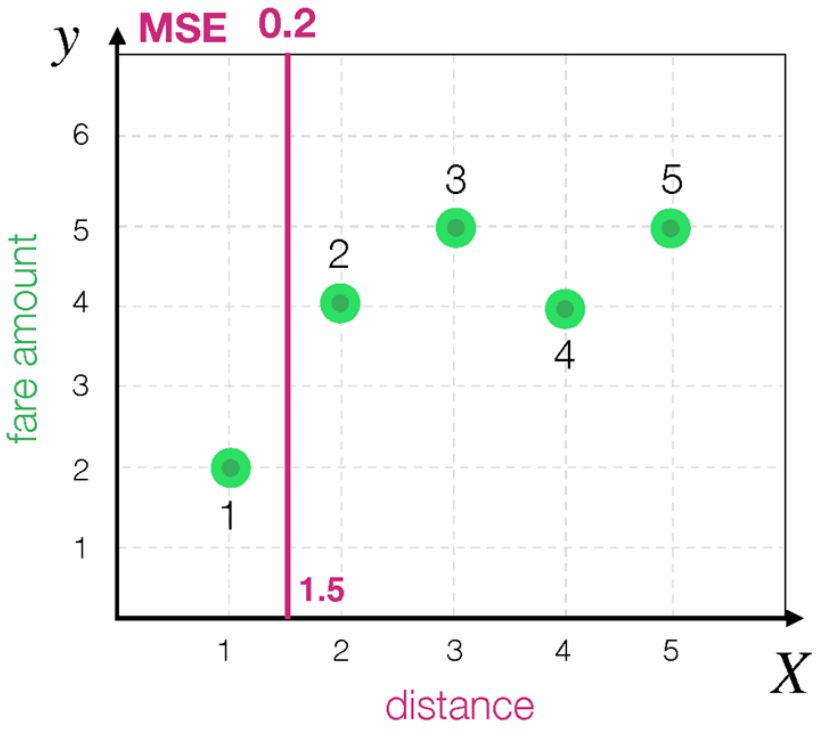


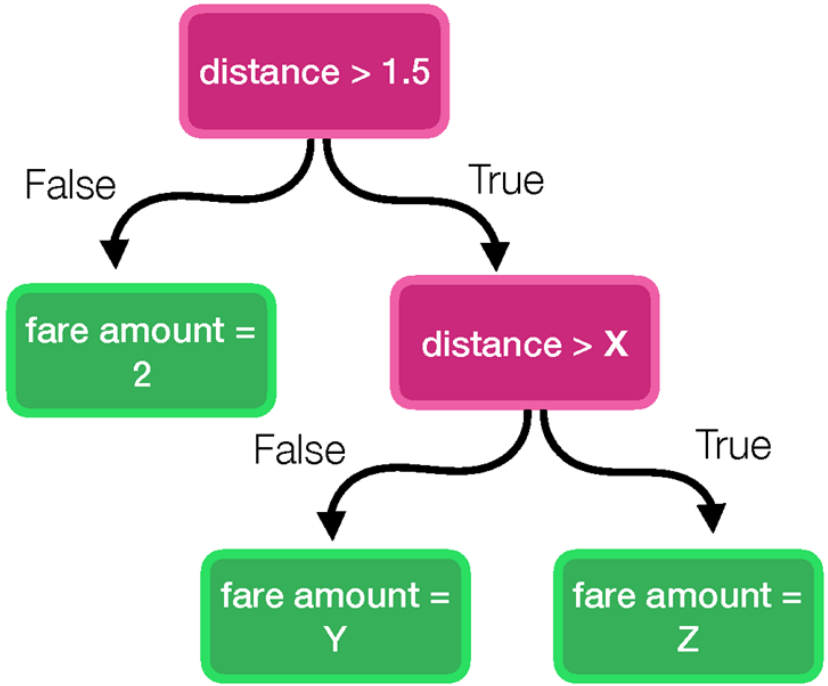


Source: Dima Fishman, Supervised Machine Learning

Decision Tree Algorithm

Can we make our decision tree more accurate?
Yes, by going **deeper**!





Source: Dima Fishman, Supervised Machine Learning

Characteristics of decision trees

Pros	Cons
Easy to interpret	Easy to overfit or underfit the model
Can handle numeric or categorical features	Cannot model interactions between features
Can handle missing data	Large trees can be difficult to interpret
Uses only the most important features	
Can be used on very large or small data	

A tree stops growing at a node when…

- pure or nearly pure
- no remaining variables on which to further subset the data
- the tree has grown to a preselected size limit

Bias-variance tradeoff

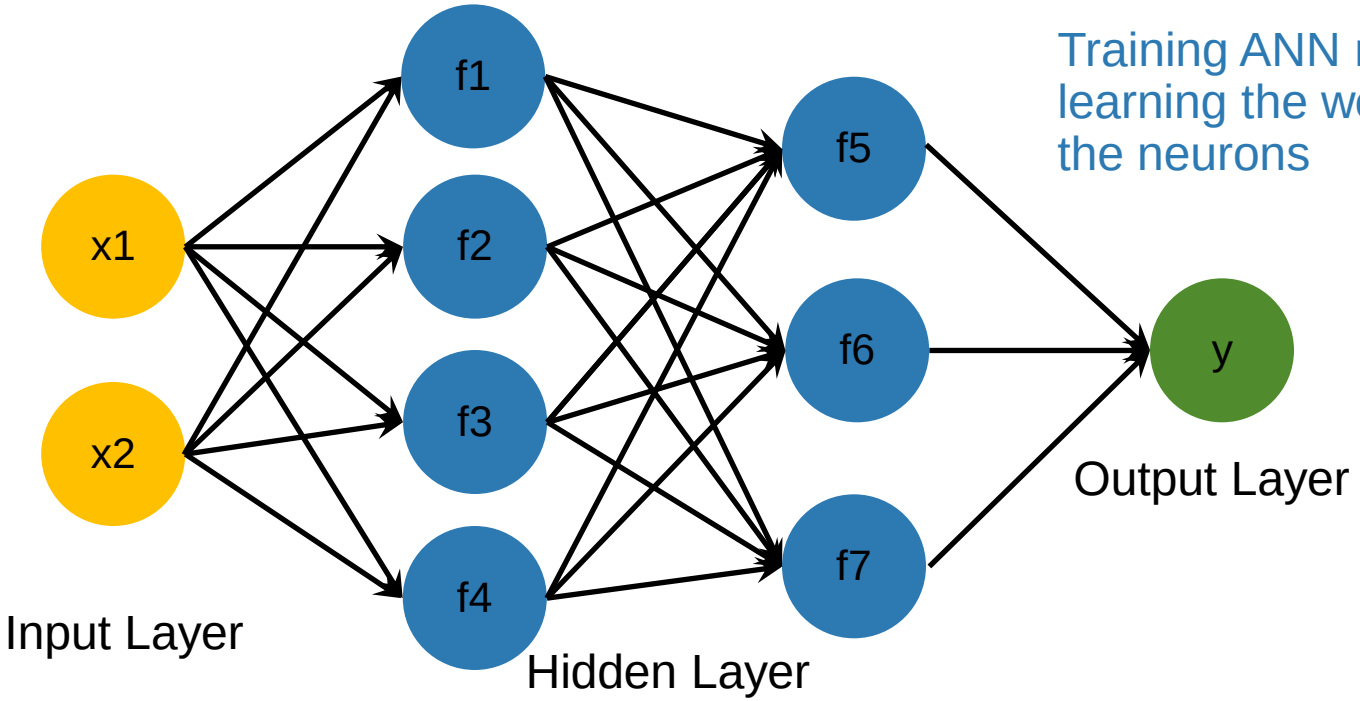
error = bias + variance

- **Bias-variance tradeoff** is the problem of simultaneously minimizing two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:
  - **Bias** is error from erroneous assumptions in the learning algorithm, high bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting)
  - **Variance** is error from sensitivity to small fluctuations in the training set, high variance can cause overfitting, i.e., modeling the random noise in the training data, rather than the intended outputs
- **Ensemble tree methods**:
  - Gradient Boosting (GBoost) is based on weak learners (high bias, low variance). In terms of decision trees, weak learners are shallow trees, sometimes even as small as decision stumps (trees with two leaves). Boosting reduces error mainly by reducing bias.
  - Random Forest uses fully grown decision trees (low bias, high variance). It tackles the error reduction task by reducing variance. The trees are made uncorrelated to maximize the decrease in variance, but the algorithm cannot reduce bias (which is slightly higher than the bias of an individual tree in the forest).

📖 Supervised Machine Learning – Neural Networks

Artificial neural networks (ANNs)

- Based loosely on computer models of how brains work
- Model is an assembly of inter-connected neurons (nodes) and weighted links
- Each neuron applies a nonlinear function to its inputs to produce an output
- Output node sums up each of its input value according to the weights of its links
- Used for classification, pattern recognition, speech recognition
- “Black Box” model – no explanatory power, very hard to interpret the results
- ANNs with more than one hidden layer are called Deep Learning networks



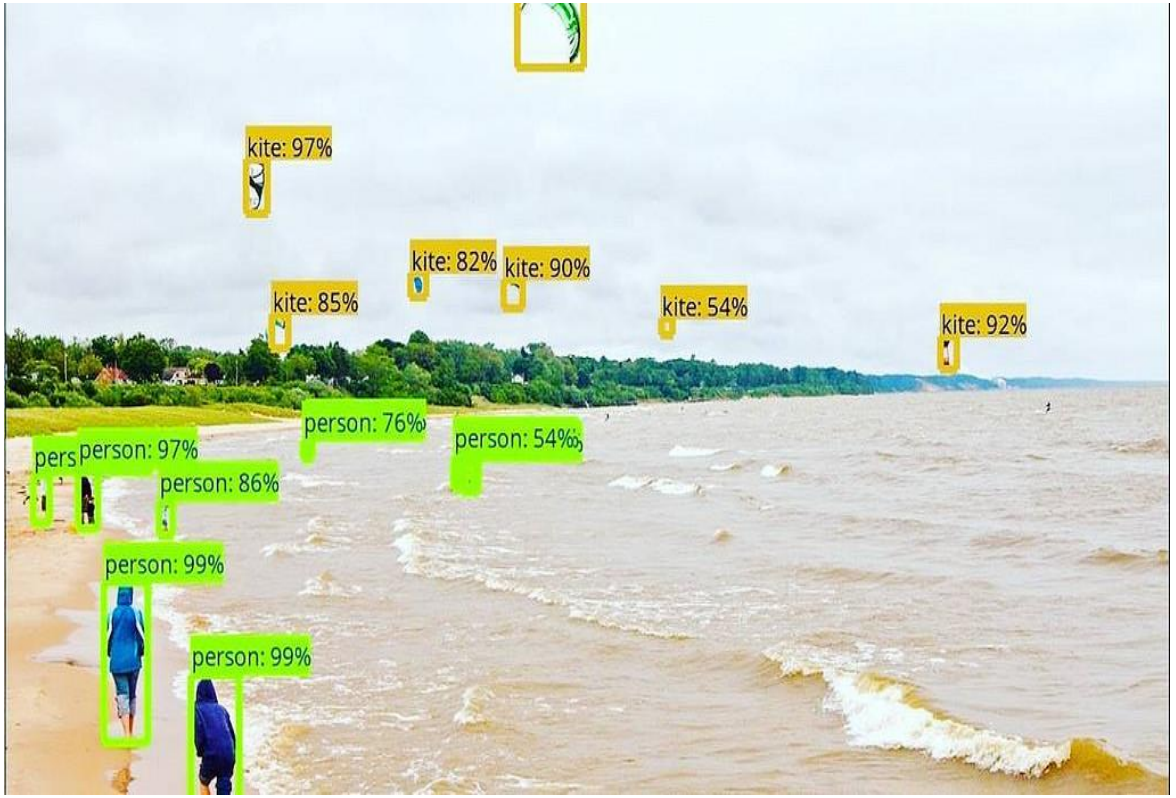
Applications of Deep Learning – Computer Vision

- **Image classification**
- Object detection
- Type: Feedforward Neural Network
- Type: Convolutional Neural Network

Source: CognitiveClass.ai

Applications of Deep Learning – Computer Vision

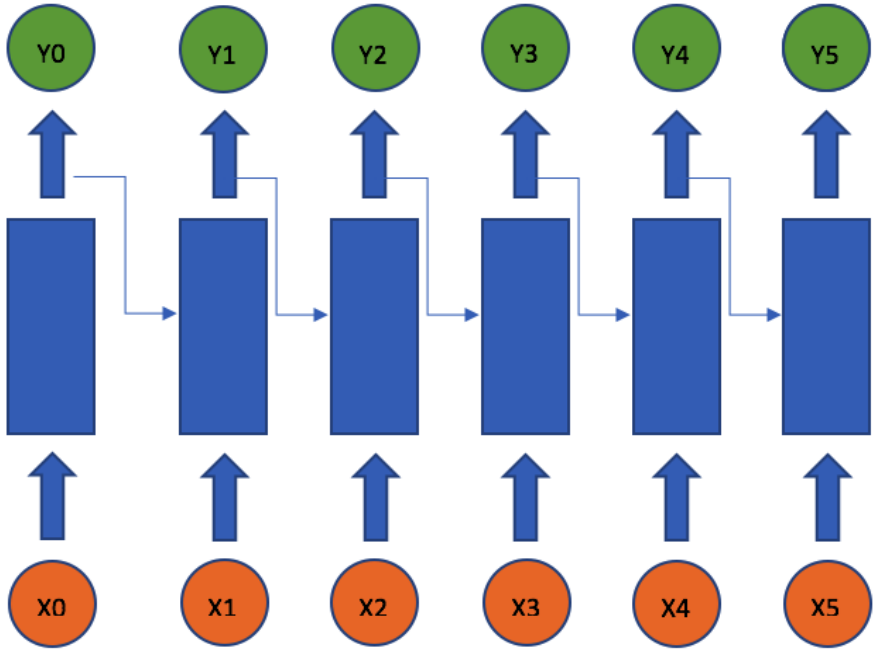
- Image classification
- **Object detection**
- Type: Feedforward Neural Network
- Type: Convolutional Neural Network



Source: CognitiveClass.ai

Applications of Deep Learning – Natural Language Processing and Time Series Analysis

- Speech recognition
- Automatic machine translation
- Time series prediction
- Type: Recurrent Neural Network

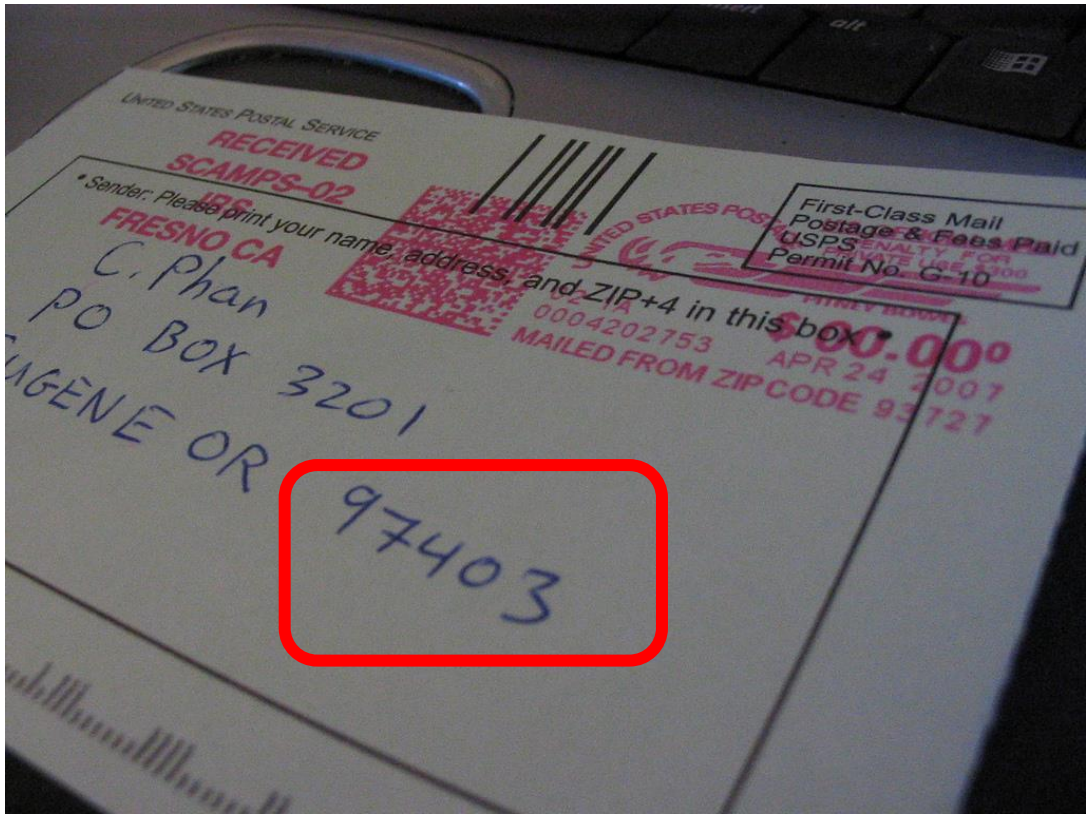


https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f

Source: CognitiveClass.ai

Applications in Computer Vision

- Application: automatically classifying **postal codes**



Source: CognitiveClass.ai

Target variables in MNIST dataset – multi-class classification

$y = 0$ $y = 1$ $y = 2$ ... $y = 9$

Source: CognitiveClass.ai

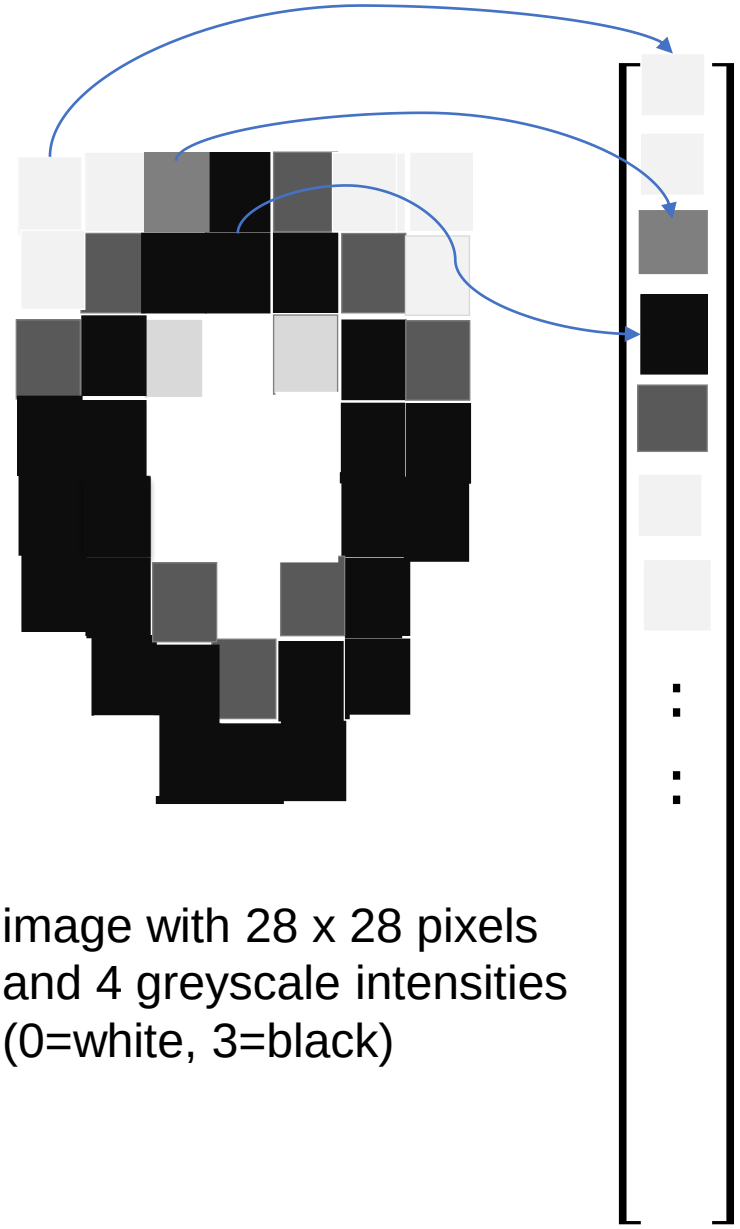


image with 28 x 28 pixels
and 4 greyscale intensities
(0=white, 3=black)

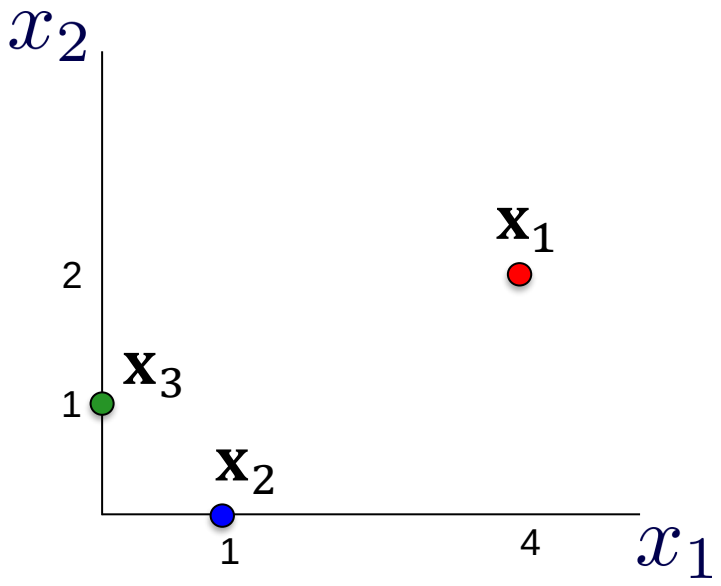
$$\mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 2 \\ \vdots \\ \vdots \end{bmatrix} \quad 784$$

Source: CognitiveClass.ai

Features and targets: 3-class example with 2 features

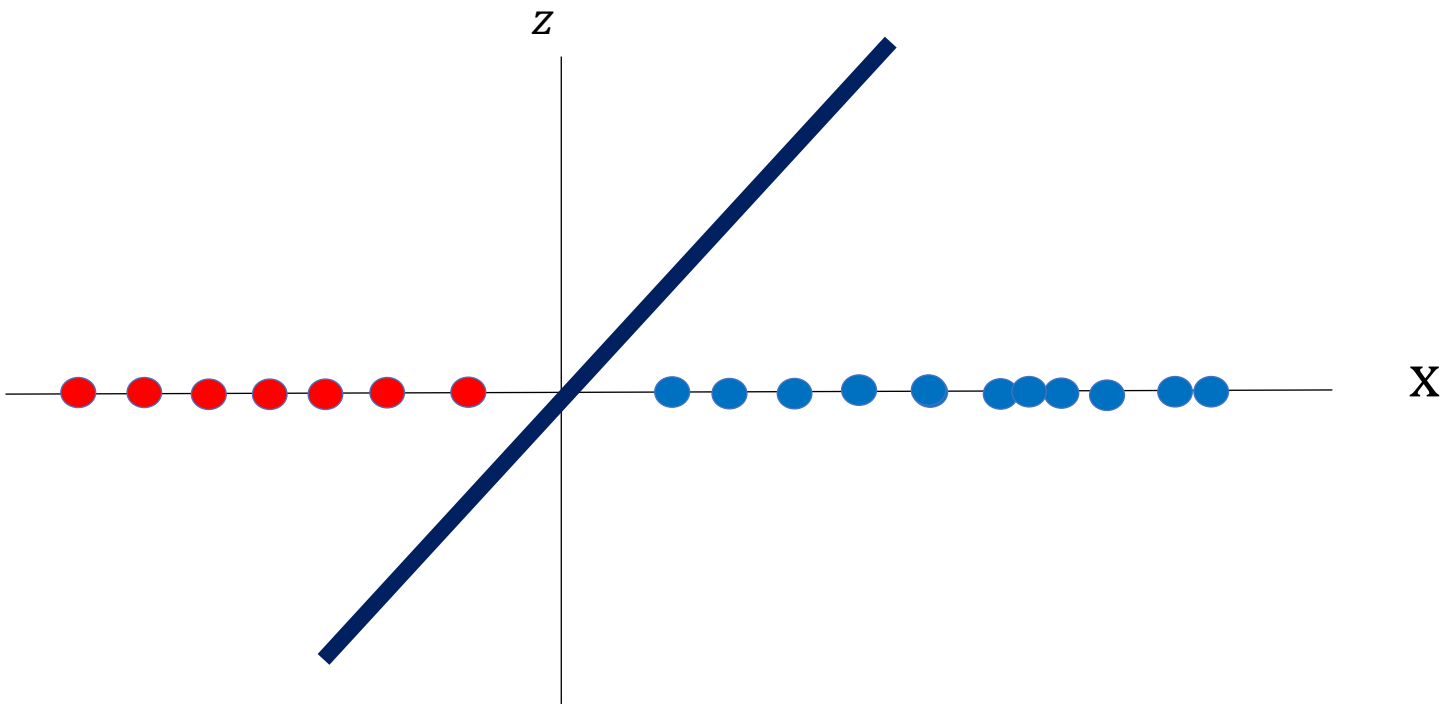
Colors are used to indicates the class

1. $(y_1 = 0, \mathbf{x}_1 = [4, 2])$
2. $(y_2 = 1, \mathbf{x}_2 = [1, 0])$
3. $(y_3 = 2, \mathbf{x}_3 = [0, 1])$



Source: CognitiveClass.ai

Linear classifier



Source: CognitiveClass.ai

Linear classifier

- The equation of a line in 1 dimension is given by:

$$wx + b$$

- This generalizes in D dimensions to :

$$\boldsymbol{w}^T\mathbf{x} + b$$

- Let's see what happens for different values of $\mathbf{x}$

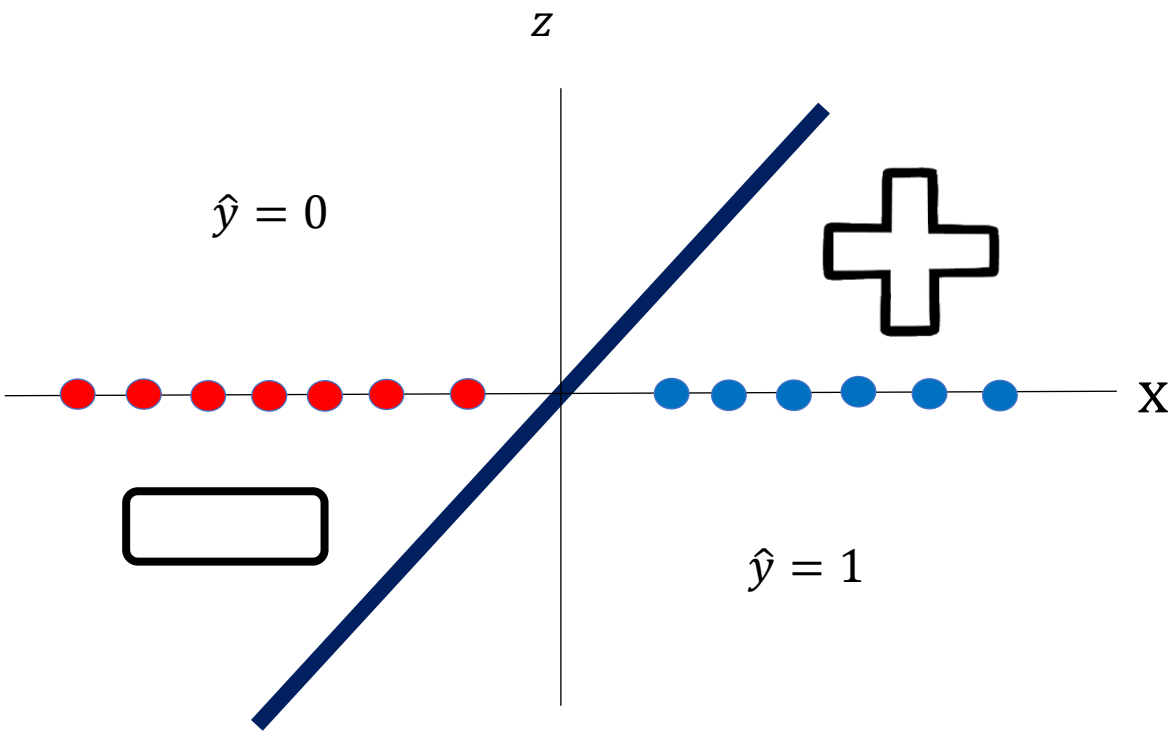
Source: CognitiveClass.ai

Linear classifier

- Consider the following data set
- If we can separate the data with a line, we can use that line to classify the sample

$$z = wx + b$$

$$= 1x - 1$$



Source: CognitiveClass.ai

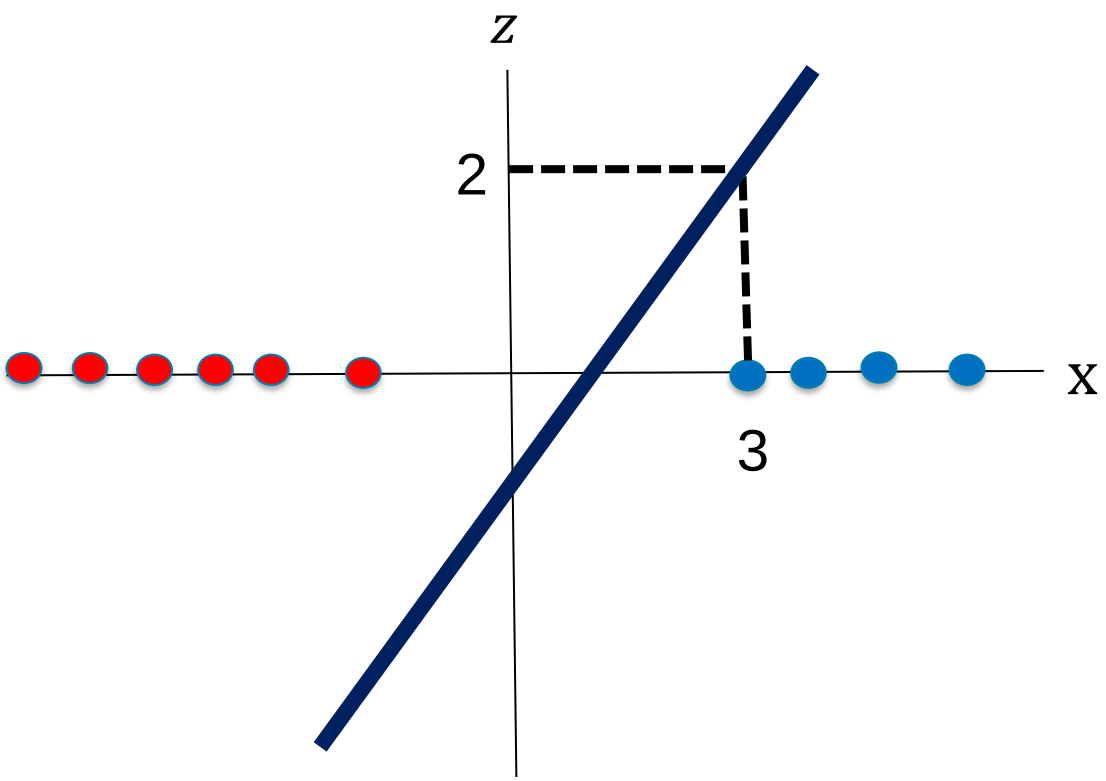
Linear classifier

- If x is on the left side of the line we get a positive number

$$z = 1x - 1$$

$$z = 1(3) - 1$$

$$= 2$$

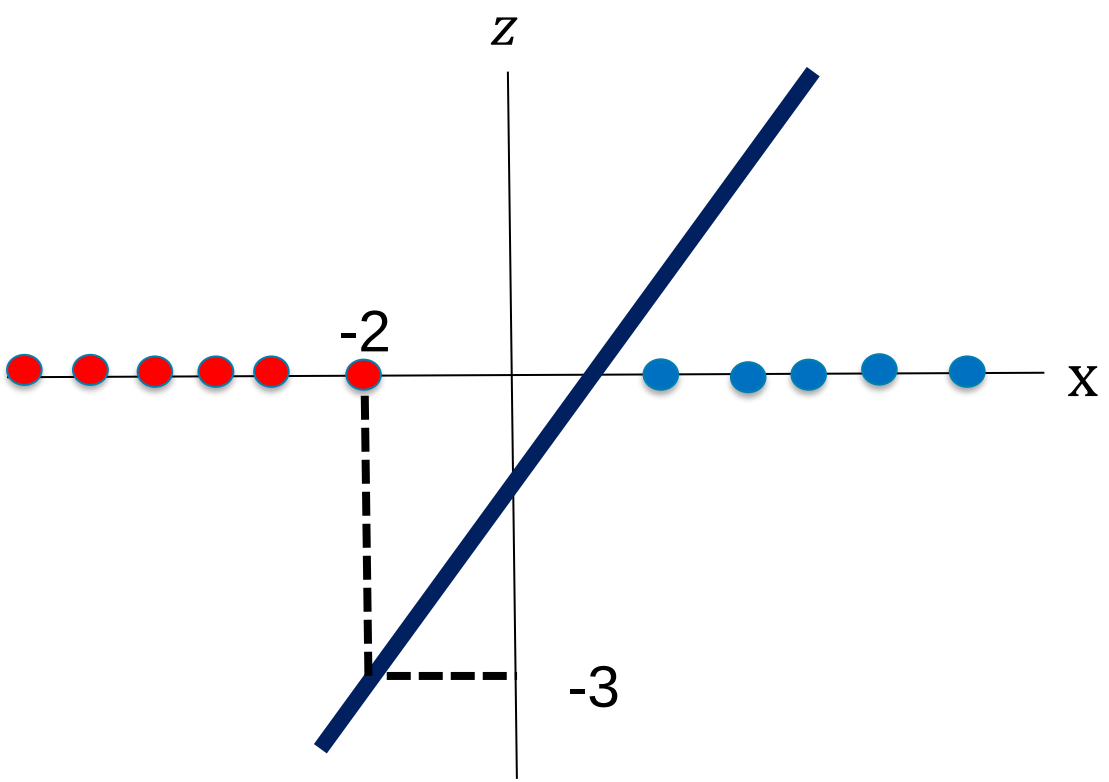


Source: CognitiveClass.ai

Linear classifier

- If x is on the right side of the line we get a positive number

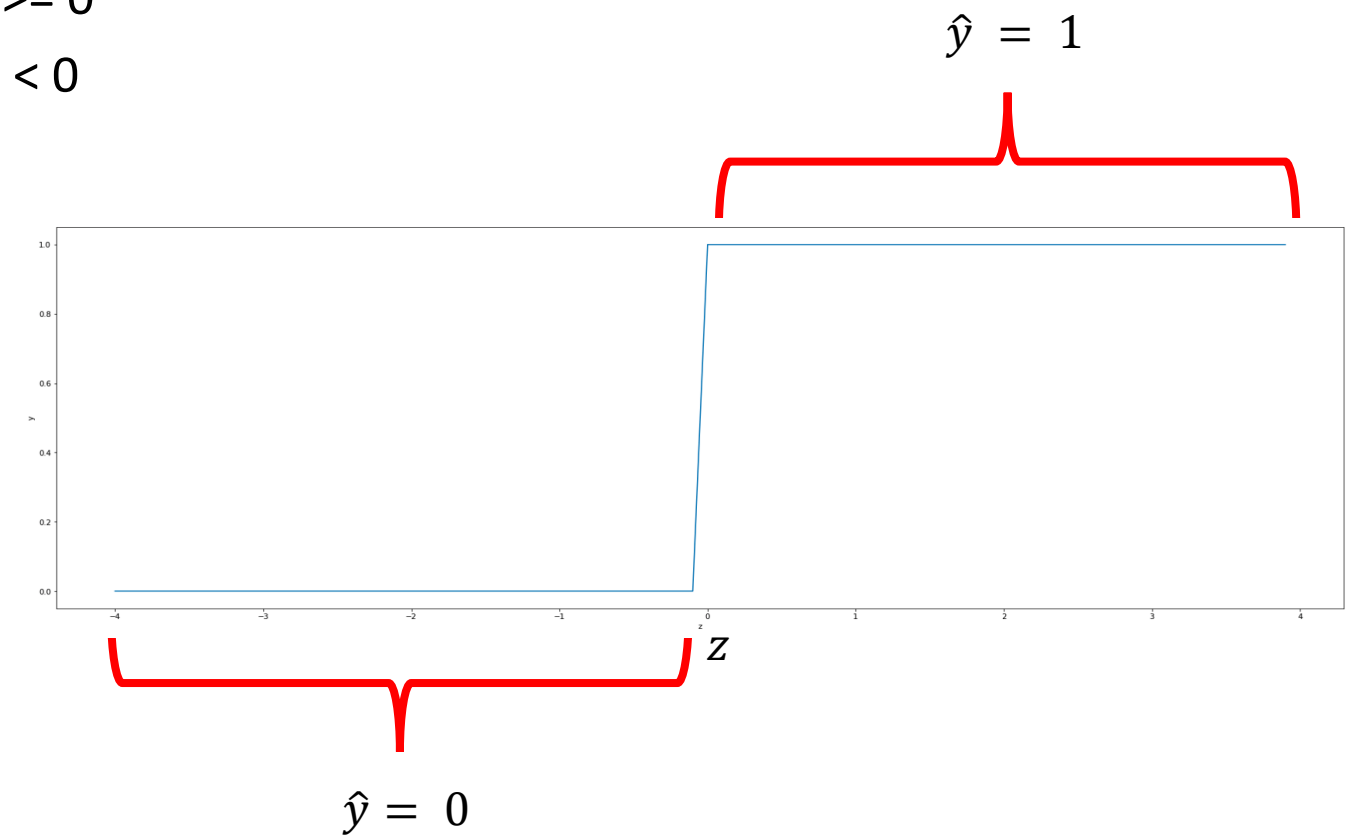
$$z = 1x - 1$$
$$z = 1(-2) - 1$$
$$= -3$$



Source: CognitiveClass.ai

Linear classifiers

- If we use the line to calculate the class of a point, it always returns a positive or negative number, such as 3, -2, and so on.
- But, we need class 0 and class 1. How we can convert the numbers into 0 and 1?
- $\hat{y} = 1$, if $z >= 0$
- $\hat{y} = 0$, if $z < 0$



Source: CognitiveClass.ai

Linear classifier

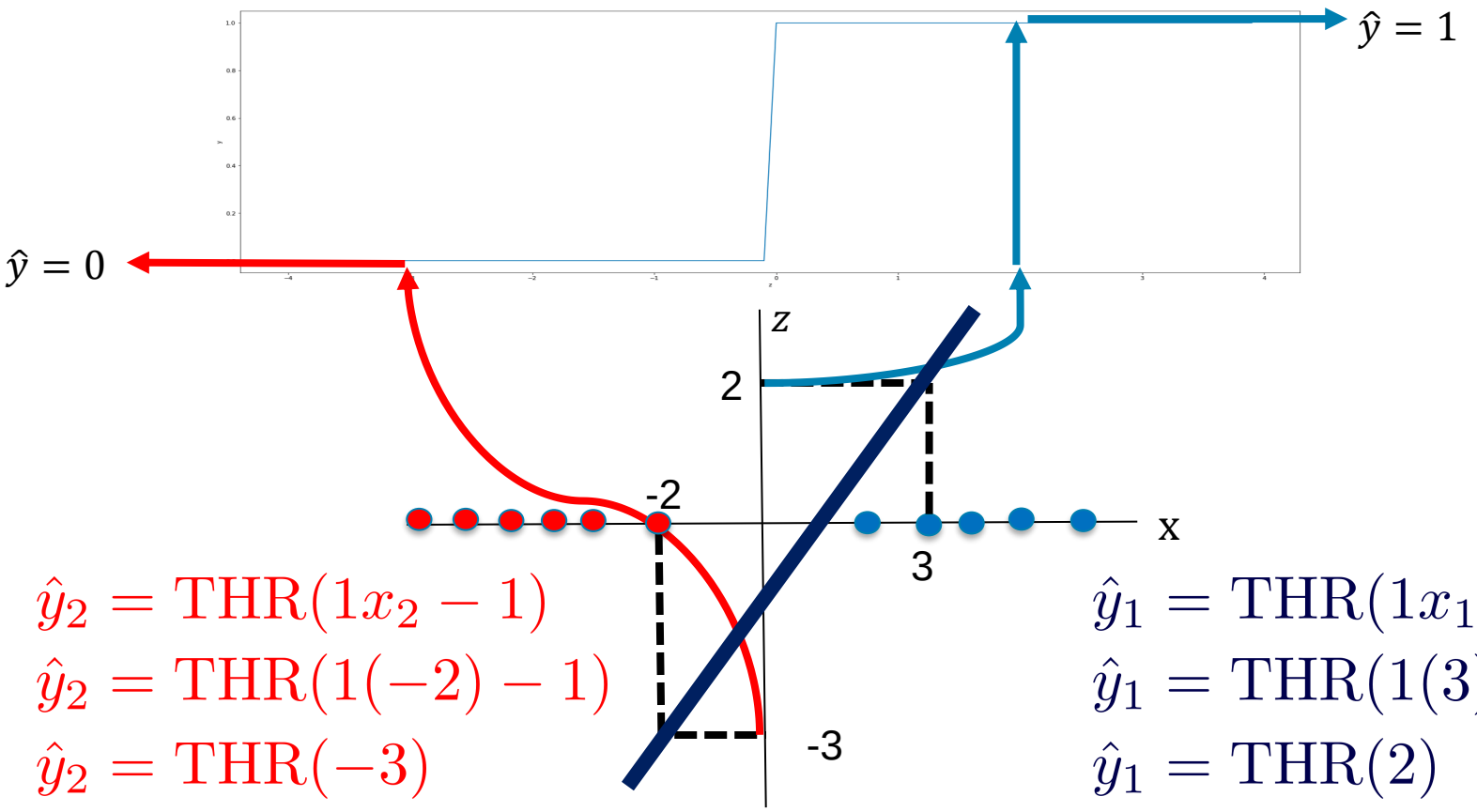
$$\hat{y}_n = f(x_n)$$

parameters: w, b

$$\hat{y}_n = \mathrm{THR}(wx_n + b)$$

Source: CognitiveClass.ai

Linear classifier: Threshold Function

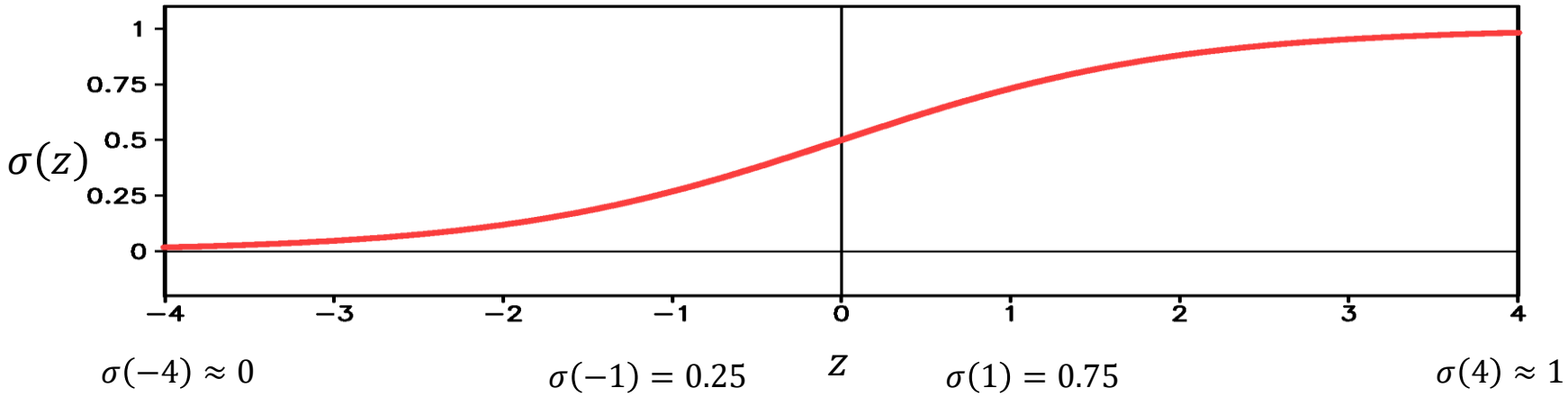


$$\hat{y}_2 = \text{THR}(1x_2 - 1)$$
$$\hat{y}_2 = \text{THR}(1(-2) - 1)$$
$$\hat{y}_2 = \text{THR}(-3)$$
$$\hat{y}_2 = 0$$

$$\hat{y}_1 = \text{THR}(1x_1 - 1)$$
$$\hat{y}_1 = \text{THR}(1(3) - 1)$$
$$\hat{y}_1 = \text{THR}(2)$$
$$\hat{y}_1 = 1$$

Source: CognitiveClass.ai

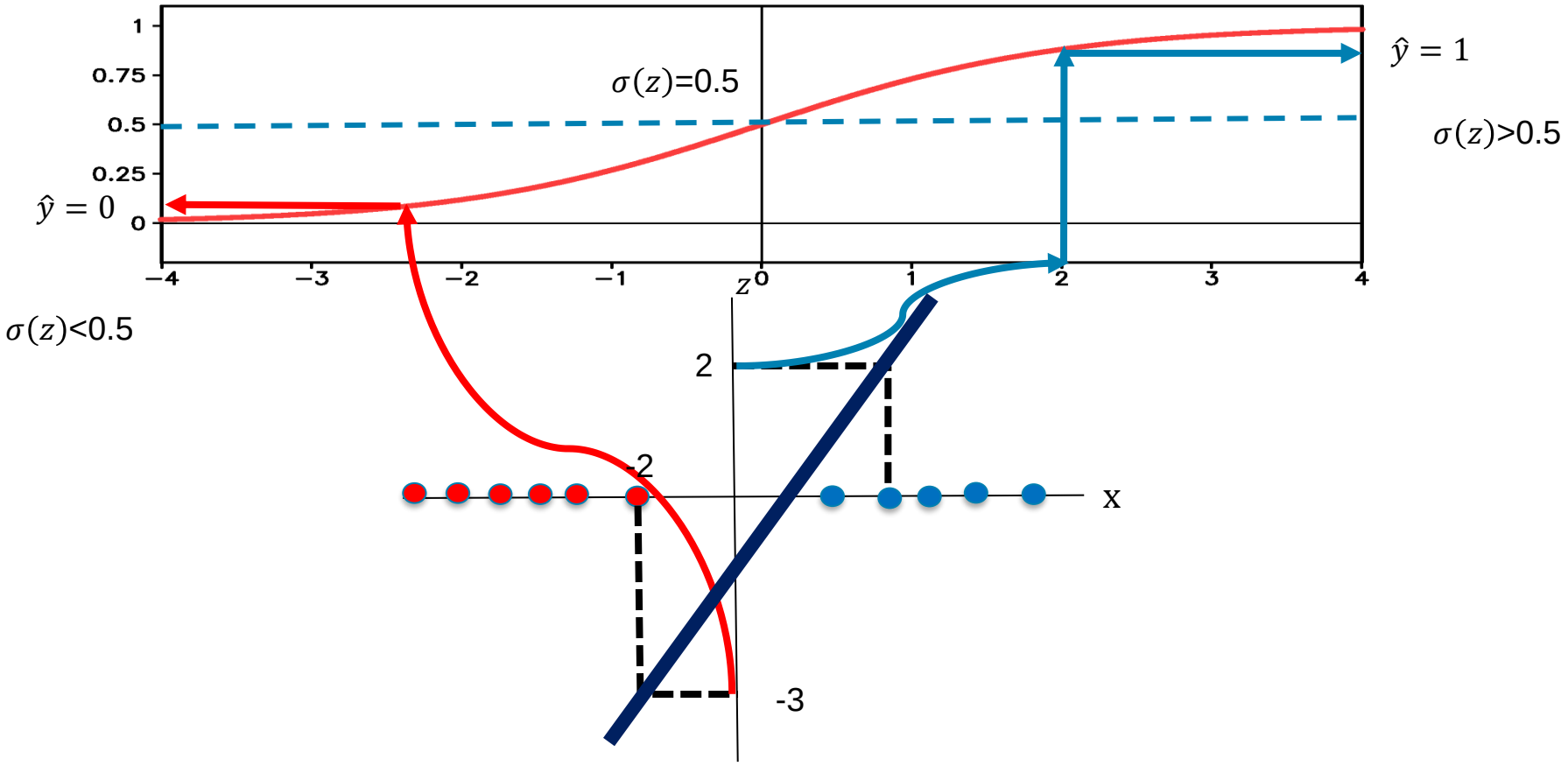
Logistic function



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Source: CognitiveClass.ai

Linear classifier: Logistic Regression

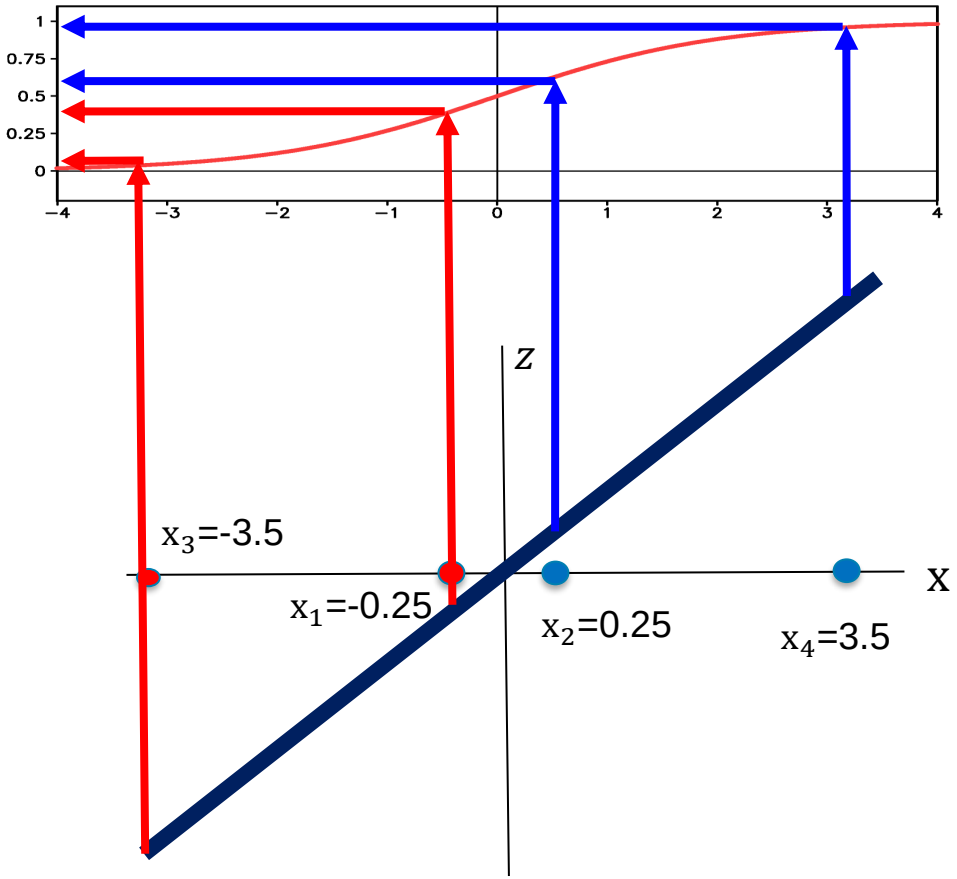


Source: CognitiveClass.ai

Linear classifier: Logistic Regression

- Samples are near the line, we get values close to 0.5
- If they are far, we get values close to 0 or 1

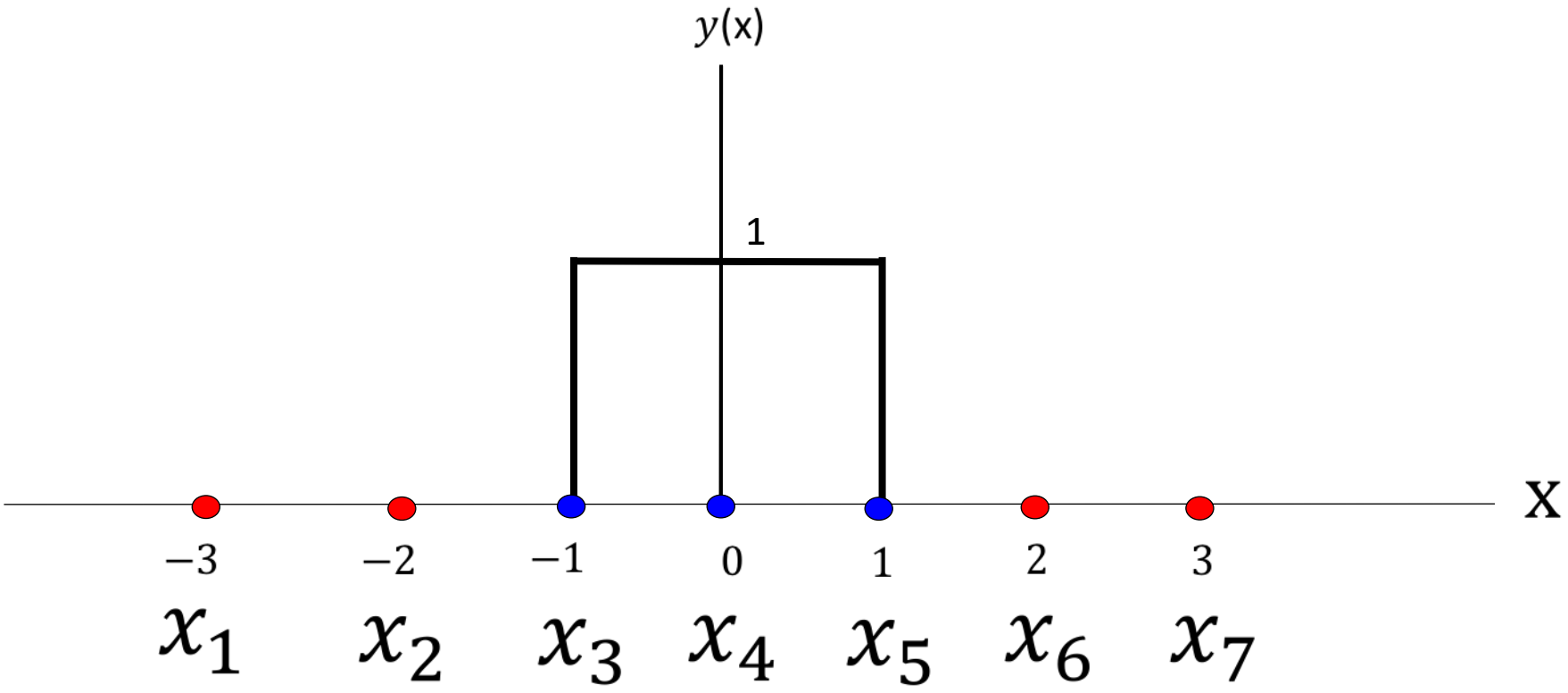
x	y	$\sigma(z)$	$\hat{y}$
x_1	0	0. 44	0
x_2	1	0.56	1
x_3	0	0. 02	0
x_4	1	0.98	1



Source: CognitiveClass.ai

Linear classifiers vs. neural networks – example

- It is helpful to view the sample y as a decision function of x



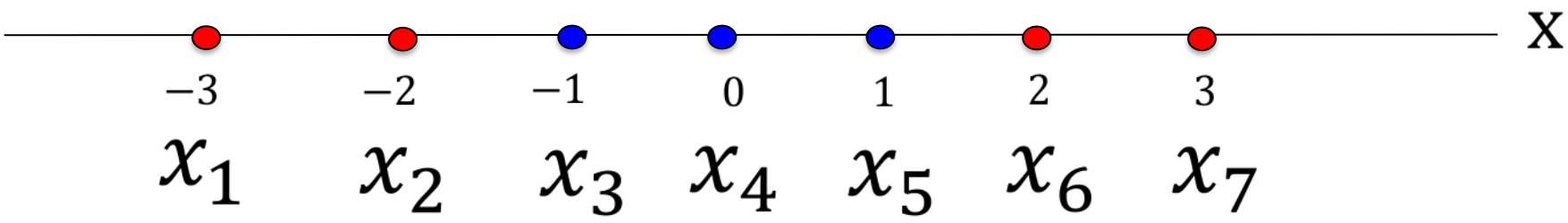
Source: CognitiveClass.ai

Linear classifiers vs. neural networks – example

Colors are used to indicates the class

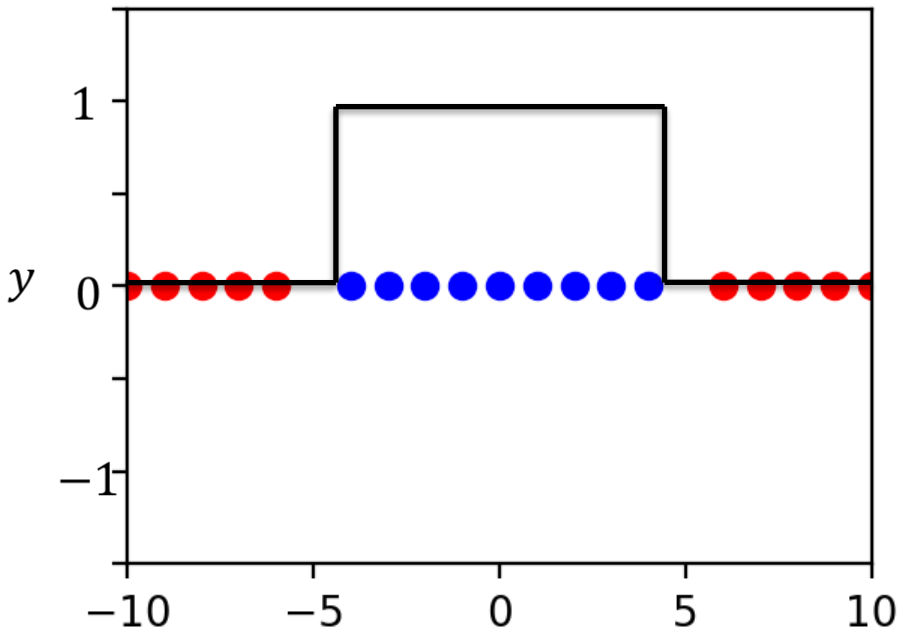
1. $y_1 = 0, x_1 = -3$
2. $y_2 = 0, x_2 = -2$
3. $y_3 = 1, x_3 = -1$
4. $y_4 = 1, x_4 = 0$
5. $y_5 = 1, x_5 = 1$
6. $y_6 = 0, x_6 = 2$
7. $y_7 = 0, x_7 = 3$

This dataset is not linearly separable



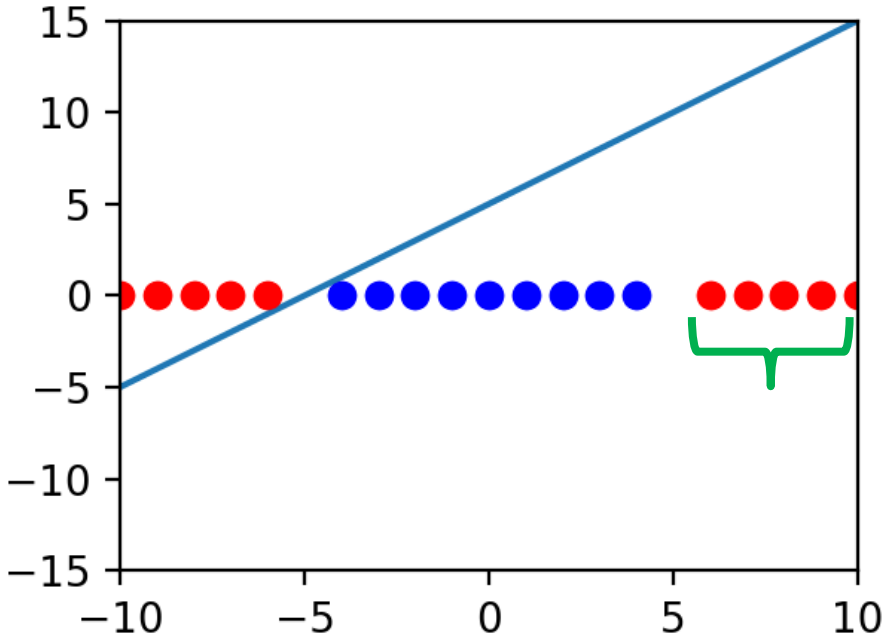
Source: CognitiveClass.ai

Linear classifiers vs. neural networks



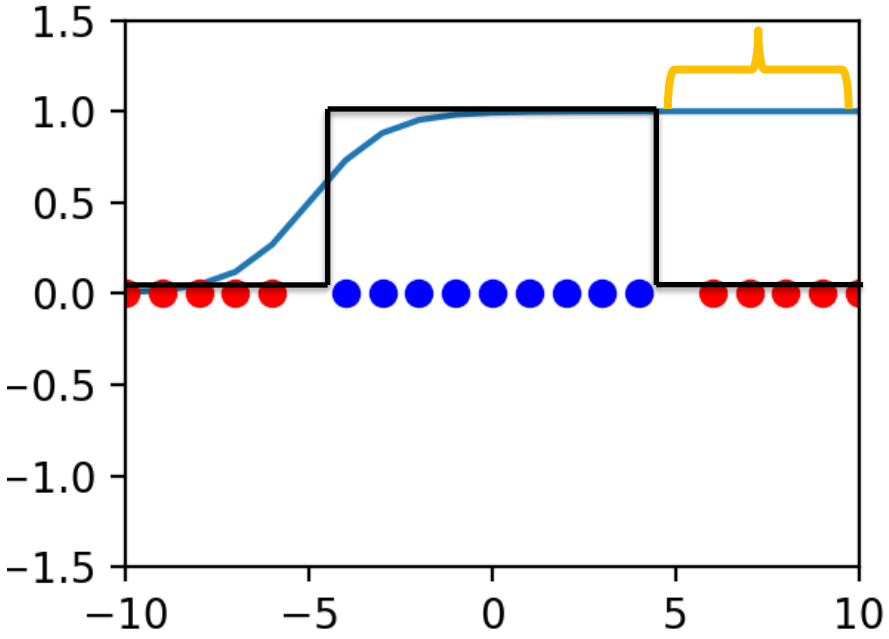
Source: CognitiveClass.ai

Neural networks



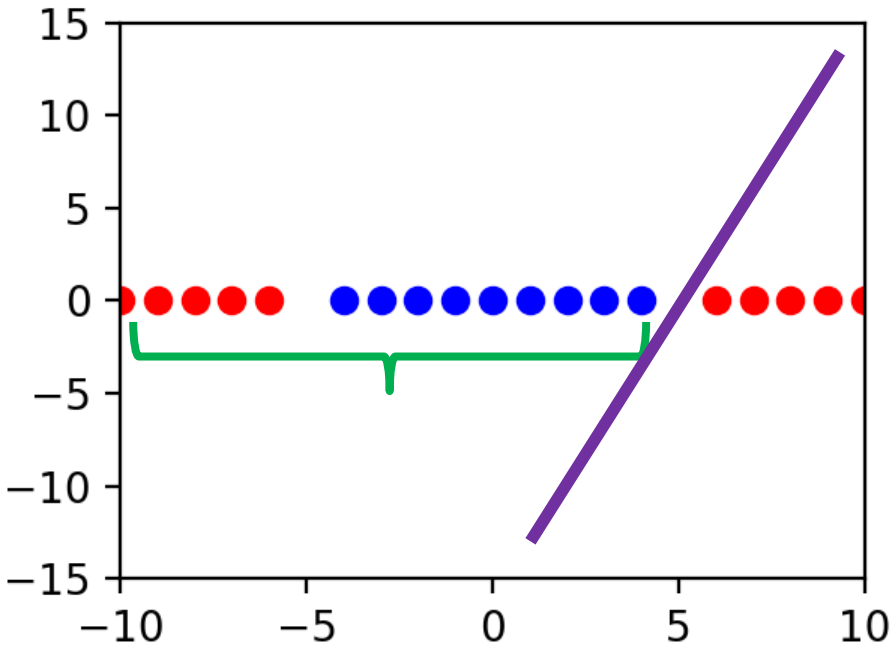
Source: CognitiveClass.ai

Neural networks



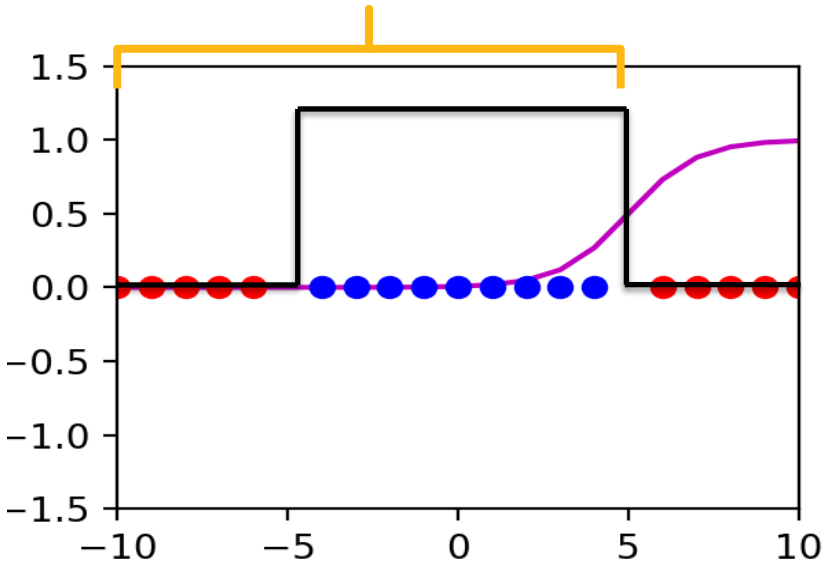
Source: CognitiveClass.ai

Neural networks



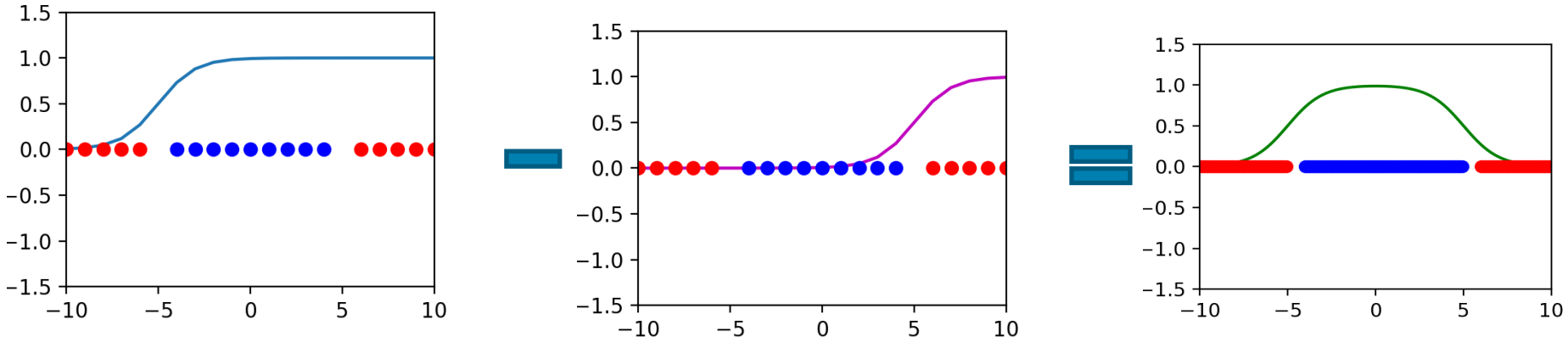
Source: CognitiveClass.ai

Neural networks



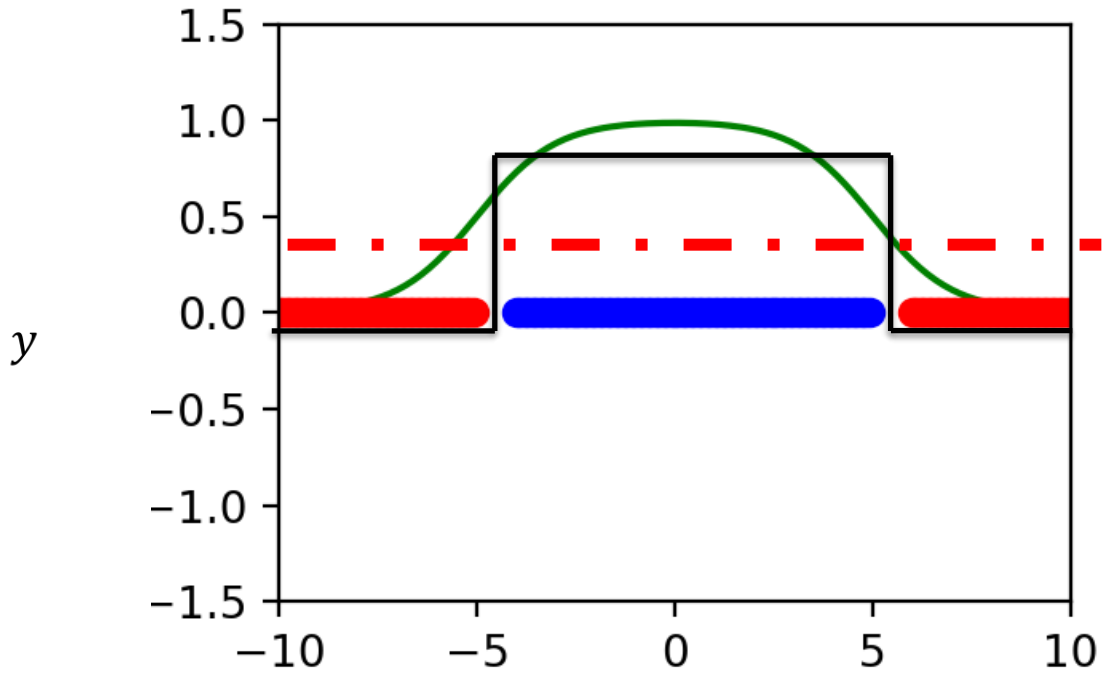
Source: CognitiveClass.ai

Neural networks

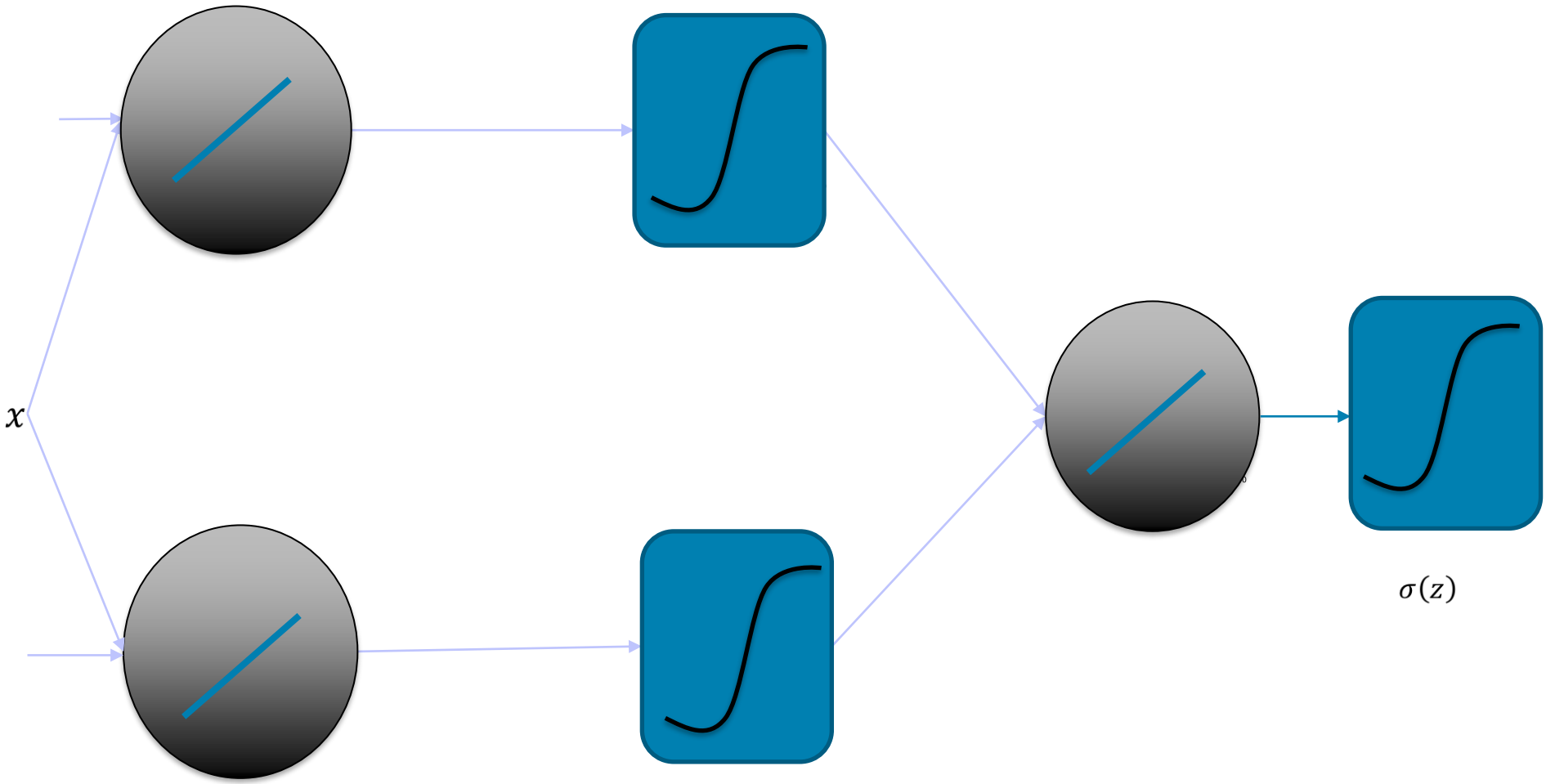


Source: CognitiveClass.ai

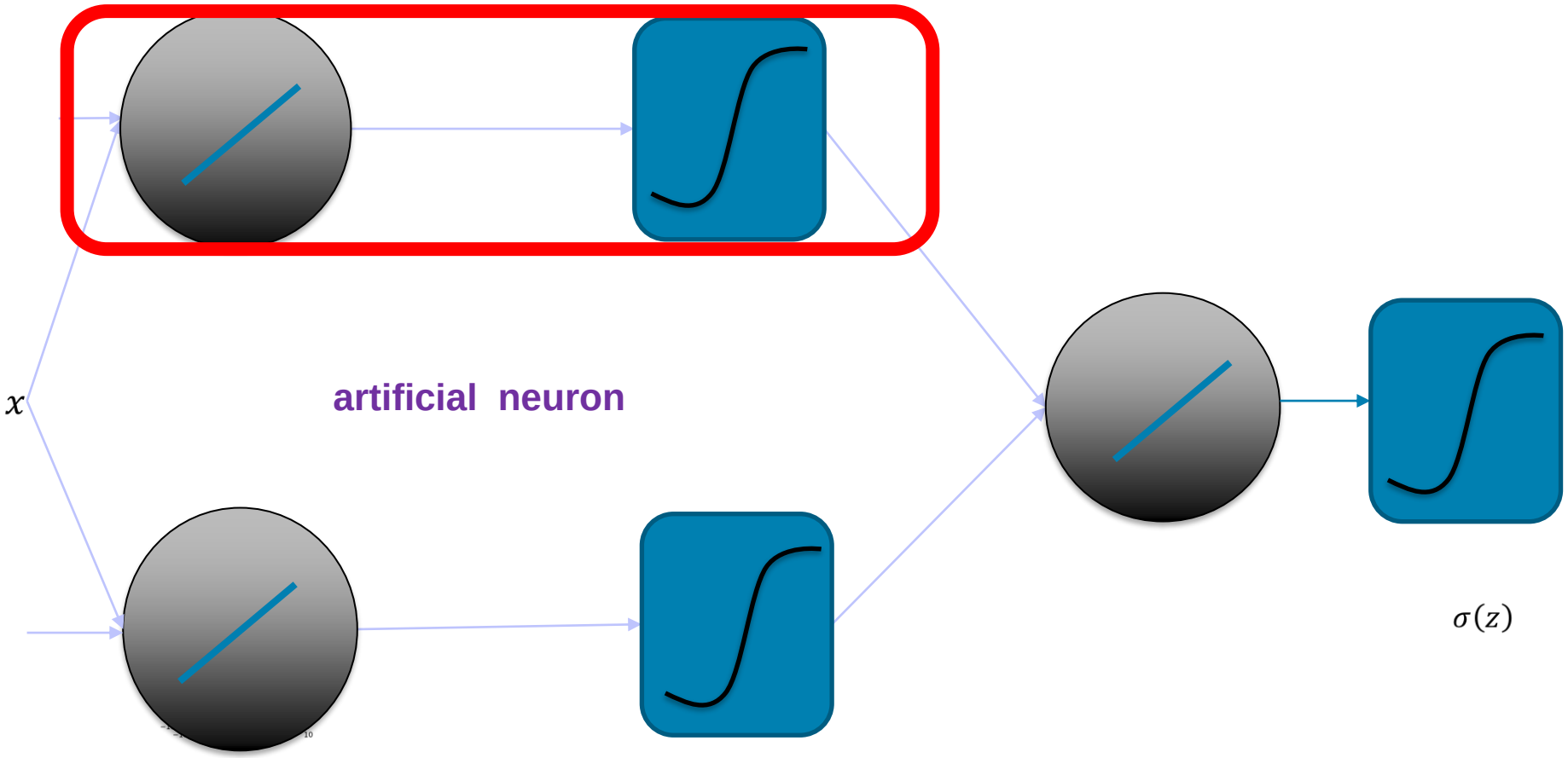
Neural networks



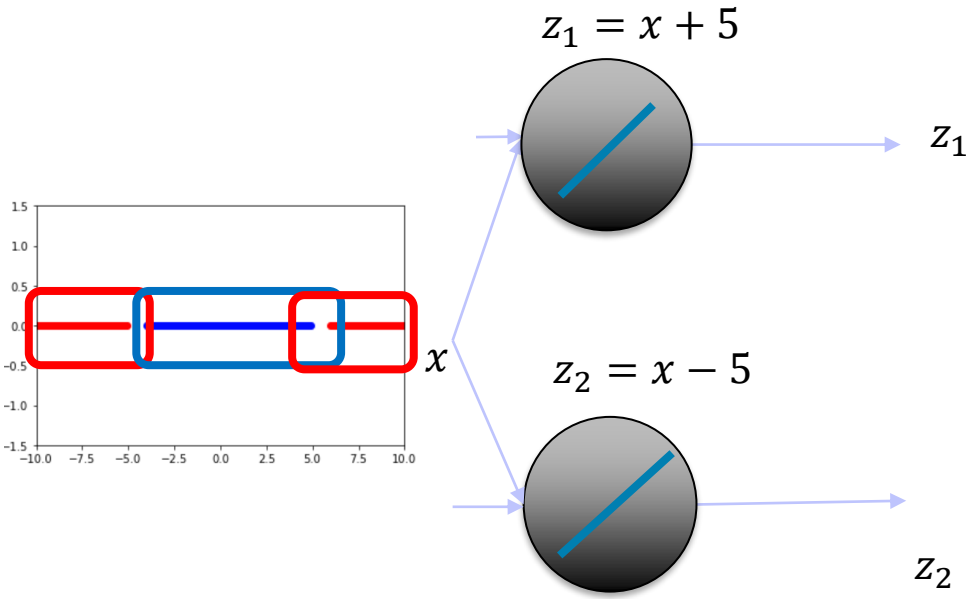
Source: CognitiveClass.ai

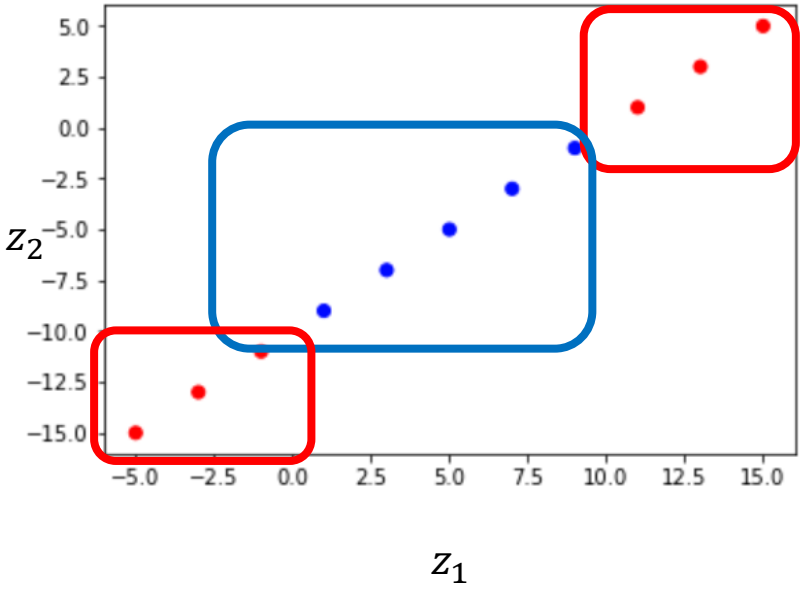


Source: CognitiveClass.ai

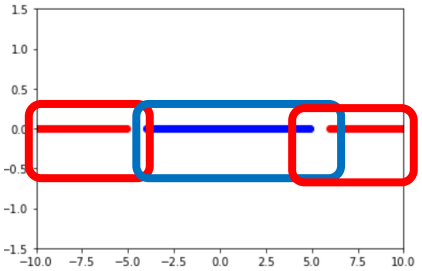


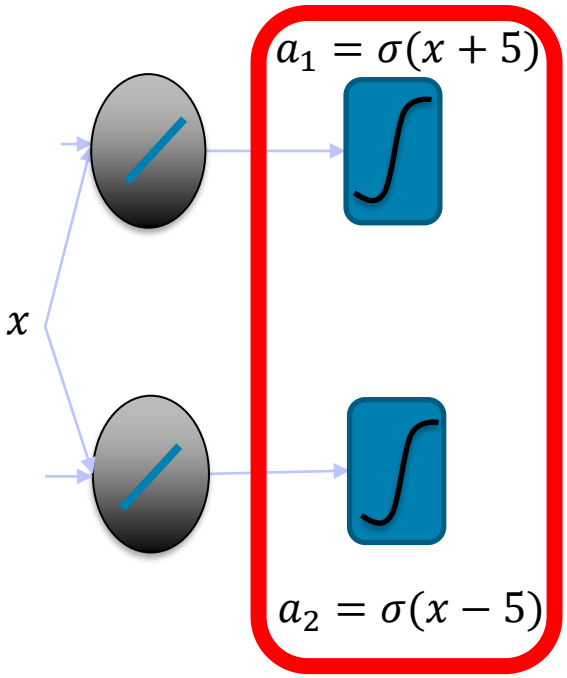
Source: CognitiveClass.ai

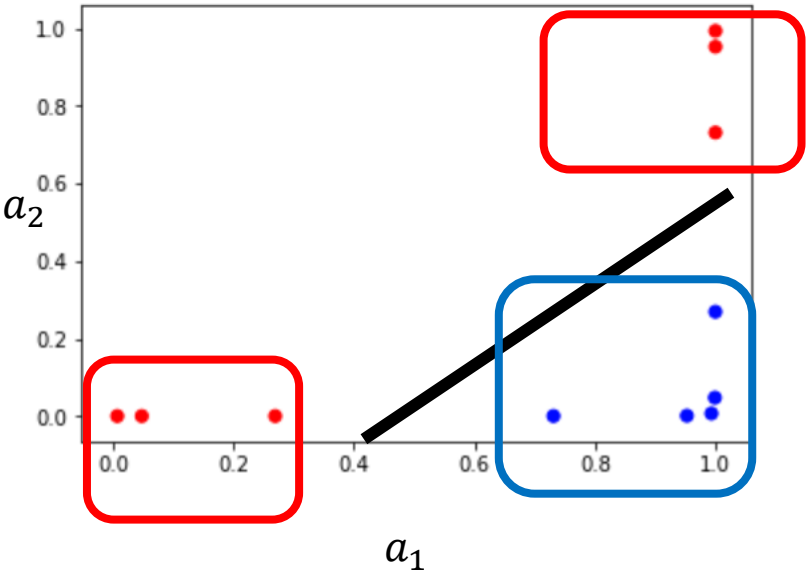




Source: CognitiveClass.ai





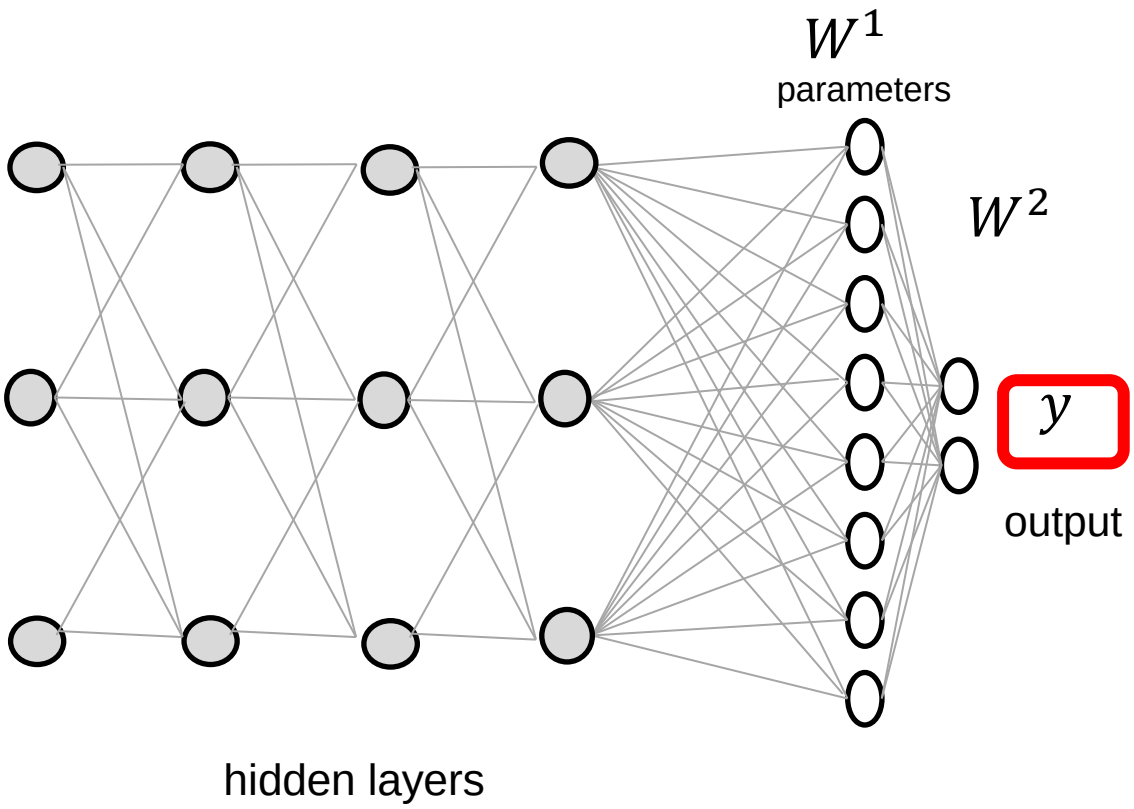


activation

Source: CognitiveClass.ai

Neural networks and tesnors

input



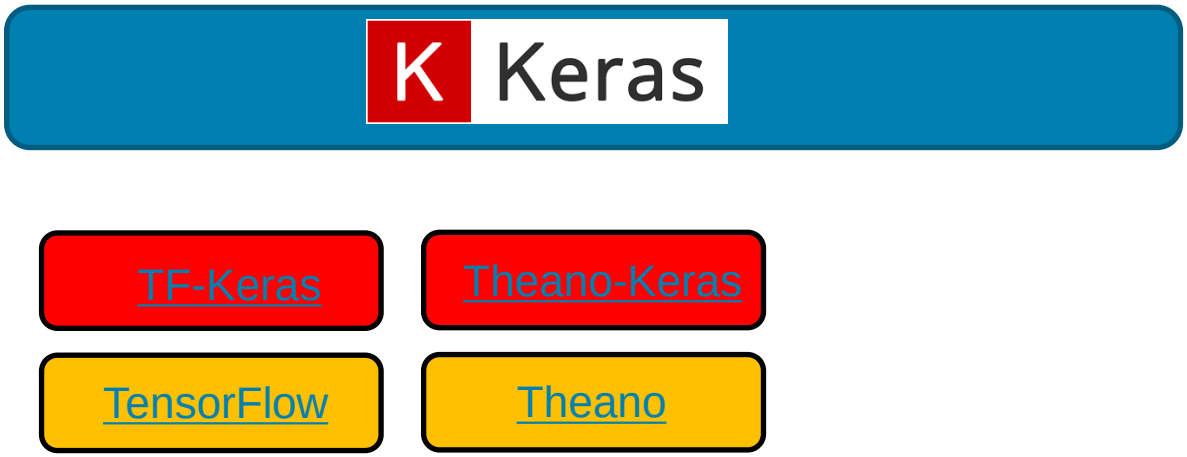
Source: CognitiveClass.ai

TensorFlow	PyTorch
TensorFlow is used in large scale production and deployment	Versatility of this Pythonic framework allows researchers to test out ideas with almost zero friction

The Battle: TensorFlow vs. PyTorch
by Eitan Rosenzvaig

Source: CognitiveClass.ai

Interface



Source: CognitiveClass.ai

Keras	PyTorch
Keras may be easier to get into and experiment with standard layers, in a plug & play spirit	Low-level environment for experimentation, giving the user more freedom to write custom layers and look under the hood of numerical optimization tasks.

Keras or PyTorch as your first deep learning framework: by Piotr Migdal and Rafał Jakubanis

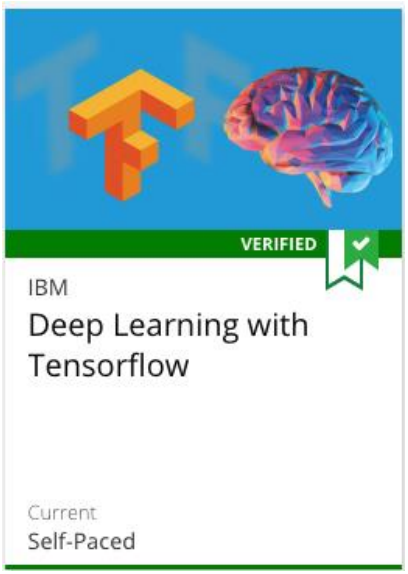
Source: CognitiveClass.ai

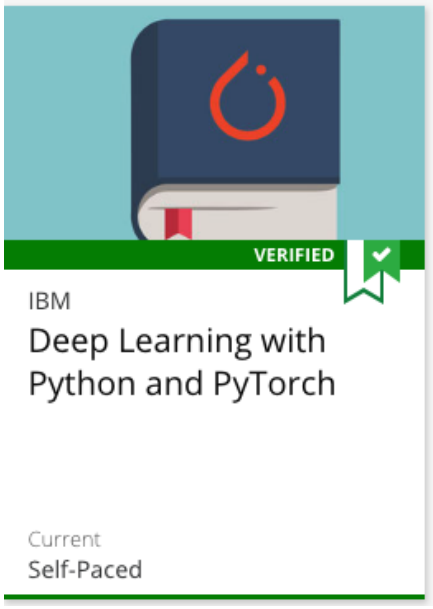
Deep Learning on EdX

Keras

TensorFlow

PyTorch





https://www.edx.org/professional-certificate/ibm-deep-learning

Source: CognitiveClass.ai