

Practical Machine Learning



Practical Machine Learning

Lecture: Introduction to Machine Learning

Ted Scully

Lecturer - Ted Scully

- Senior Lecturer in Computer Science
 - Joined MTU in 2013
 - **Programme Coordinator** for MSc in AI
 - Principal Investigator in Ríomh Research Group
 - Associate Investigator with SFI CONNECT
 - PhD Students (3 Students)
- AI Research and Development Experience
 - Machine Learning
 - Deep Learning
 - Meta-heuristic optimization



Selected Research Projects

- Real-time Detection of Weather Events with Distributed Machine Learning for SmallSat Platforms.
- A framework for PU Learning with GANs.
- Demand side management of water and energy consumption in the Irish dairy industry.
- Optimal battery utilization for micro-grid cost minimization.



Class Rep

- If anyone is interested in acting as a class rep please send me an email with the title Class Rep (ted.scully@mtu.ie).

Electives Modules

- Once you have selected your preferred elective then you should unenroll from the other electives. Please make sure you unenroll yourself before **Friday 1st of October**.

<u>Day/Time</u>	<u>Module</u>	<u>CRN</u>	<u>Self Enrolment Link</u>
Tuesday 09:00	Natural Language Processing	27375	https://cit.instructure.com/enroll/N9BMLJ
Mon 8-10pm	Distributed Ledger Technology	28467	https://cit.instructure.com/enroll/YM8RRN
Wed 6-8pm	Robotics and Autonomous Systems	<u>xxxx</u>	<u>Waiting for link from Lecturer</u>
Wed 6-8pm	Programming Language Design	28470	https://cit.instructure.com/enroll/EWDJ43
Tuesday 6-8pm	Software Agility	28469	https://cit.instructure.com/enroll/AR7ALH

Practical ML - Course Breakdown and Assessment

- Email: ted.scully@mtu.ie
- Weekly Schedule
 - Lectures 2*1hr
 - Discussion Forum
 - Practical labs (commencing in Week 2) – Lab will be given by Mr. Aidan Duggan
- Module is 100% Continuous Assessment.
 - Project 1 - Develop a machine learning model for a real-world problem and perform a comprehensive analysis. (50%).
 - Project 2 - Perform a comparative analysis of a range of machine learning classification algorithms applied to a dataset from an application domain. (50%).
 - **Assessment matrix** will be sent out over the next few days

Content

- Supervised Learning Algorithms.
 - Algorithms such as decision trees, ensemble technique (bagging and boosting, gradient-boosting), instance-based algorithms, naïve bayes, etc.
- Unsupervised Algorithms.
 - Overview of unsupervised learning techniques. Example applications of clustering techniques. Introduction to algorithms such as k-means, k-median, DBScan and hierarchical clustering techniques. Optimization and distortion cost function. Random initialization and methods of selecting number of clusters. Silhouette plots.
- Pre-processing.
 - Application of pre-processing techniques such as outlier detection, feature selection, imputation of missing data, encoding, normalization, deal with imbalanced datasets etc.
- Evaluation and Model Selection.
 - Best practice evaluation techniques such as precision, recall, confusion matrices and ROC curves. Debugging algorithms using validation and learning curves. Cross fold validation. Model selection using hyper parameter optimization.

Resources

We assume that everyone starting this course is able to code in Python and has a reasonable grasp of NumPy and Pandas. We will be using Python 3 (preferably 3.8) as our programming language in this module.

- The following are basic tutorials to get you up and running with the syntax and control structure for Python.
- [Python 3 Tutorial](#) – Clear and focused overview of Python 3 syntax, control structures, data structures etc.
- [Video Python 3 Tutorials](#) – A set of very basic Python 3 video tutorials. More focused on beginners.
- [Automate the Boring Stuff with Python](#) - Learn to Code. If you've ever spent hours renaming files or updating hundreds of spreadsheet cells, you know how tedious tasks like these can be.
-
- The following are a number of accessible tutorials to help you get started with NumPy and Pandas.
- [DataCamp NumPy Tutorial](#) – Accessible and easy to understand tutorial to get started with NumPy
- [NumPy Tutorial](#) – Short overview of NumPy and basic Python data structures. It also covers SciPy (which you don't need) and basic Matplotlib (which you will be covering later in the programme as part of visualization).
- [DataCamp Pandas Tutorial](#) – Short and easy to understand tutorial on using Pandas

Resources - DataCamp Access

I have applied for academic access to DataCamp and you will be receiving an email invitation, which will grant you access.

Please let me know if you don't receive this email before the end of week 1.



DataCamp

Resources

- **Websites**

- [Machine Learning Stanford](#) – Andrew Ng
- [Machine Learning Class \(Washington\)](#) - Pedro Domingos
- [Udacity Machine Learning](#) - Sebastian Thrun (free course)
- [UCI Data Repository](#)
- [Kaggle](#)

- **Books**

- [Hands-On Machine Learning with Scikit-Learn and TensorFlow](#) – Aurelien Geron (2nd Edition)
- [Python Machine Learning](#) - Sebastian Raschka (3rd Edition)
- [Fundamentals of Machine Learning for Predictive Data Analytics](#) – (John Kelleher, Brian MacNamee , Aoife D'Arcy)

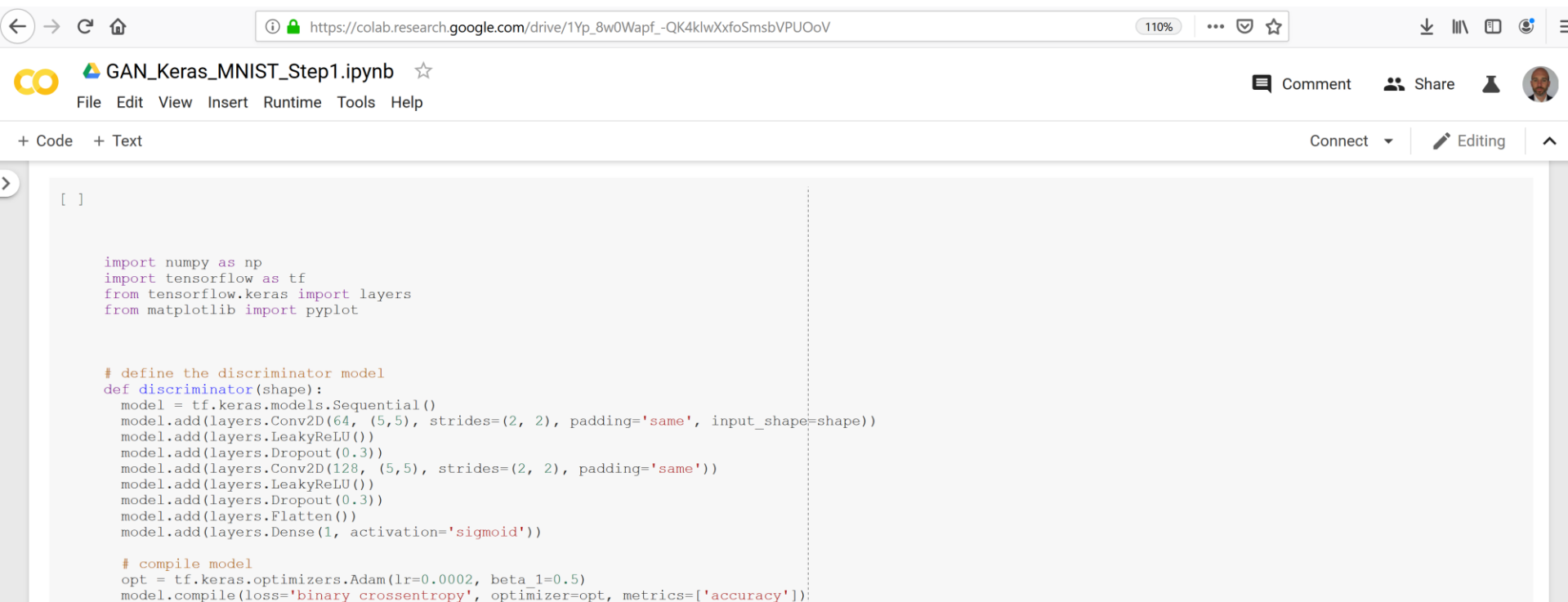
Software Options (1. Anaconda)

- Anaconda is an open-source distribution of Python.
- It comes with a range of essential packages such as NumPy, Pandas, Scikit-Learn and Matplotlib, TensorFlow, PyTorch.
- Spyder IDE or Jupyter Notebook.
- Conda - an open source package management system (isolated environments).
- Download [Anaconda](#) (Python 3.8 version)



Software Options (2. Colab)

- [Google Colab](https://colab.research.google.com/) is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.
- Again comes with essential packages such as Scikit-Learn, NumPy, etc all pre-installed.
- Comes with the option of a free GPU or TPU (not necessary for Practical ML Module)
- **Drawback**. When you connect to a VM runtime, you have a maximum of 12 hours on the VM. You can easily connect to another VM after the 12 hours expires but you will lose access to an data you had in the previous VM.



The screenshot shows a Google Colab notebook titled "GAN_Keras_MNIST_Step1.ipynb". The interface includes a browser address bar with the URL "https://colab.research.google.com/drive/1Yp_8w0Wapf_-QK4klwXxfoSmsbVPUOoV", a toolbar with icons for navigation and settings, and a menu bar with options like File, Edit, View, Insert, Runtime, Tools, and Help. The notebook content shows Python code for defining a discriminator model and compiling it.

```
[ ]

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from matplotlib import pyplot

# define the discriminator model
def discriminator(shape):
    model = tf.keras.models.Sequential()
    model.add(layers.Conv2D(64, (5,5), strides=(2, 2), padding='same', input_shape=shape))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))
    model.add(layers.Conv2D(128, (5,5), strides=(2, 2), padding='same'))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))
    model.add(layers.Flatten())
    model.add(layers.Dense(1, activation='sigmoid'))

# compile model
opt = tf.keras.optimizers.Adam(lr=0.0002, beta_1=0.5)
model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Software Options (2. Colab)

- To use Colab you will need a **Google (GMail)** account.
- Another aspect of Colab is that you can **mount files from your Google Drive.** This allows you to easily access data from the Colab VM instance.
- I have included a short guide to getting started with Google Colab in the Week 1 unit on Canvas.
 - Describes how to create a Colab Notebook from your Google Drive.
 - Mount a data file
 - Open the data file and perform some basic pre-processing on the data file.

Machine Learning

1. Machine Learning is a important **sub-discipline of AI**.
 2. One goal of AI is building programs to perform tasks, which **humans are currently better at**. Machine learning is an avenue that has had success doing exactly that.
- How do you program a computer to:
 - Recognize faces?
 - Identify objects in images
 - Interpret hand written text
 - Interpreting spoken language?
 -

Machine Learning

How do we define Machine Learning?

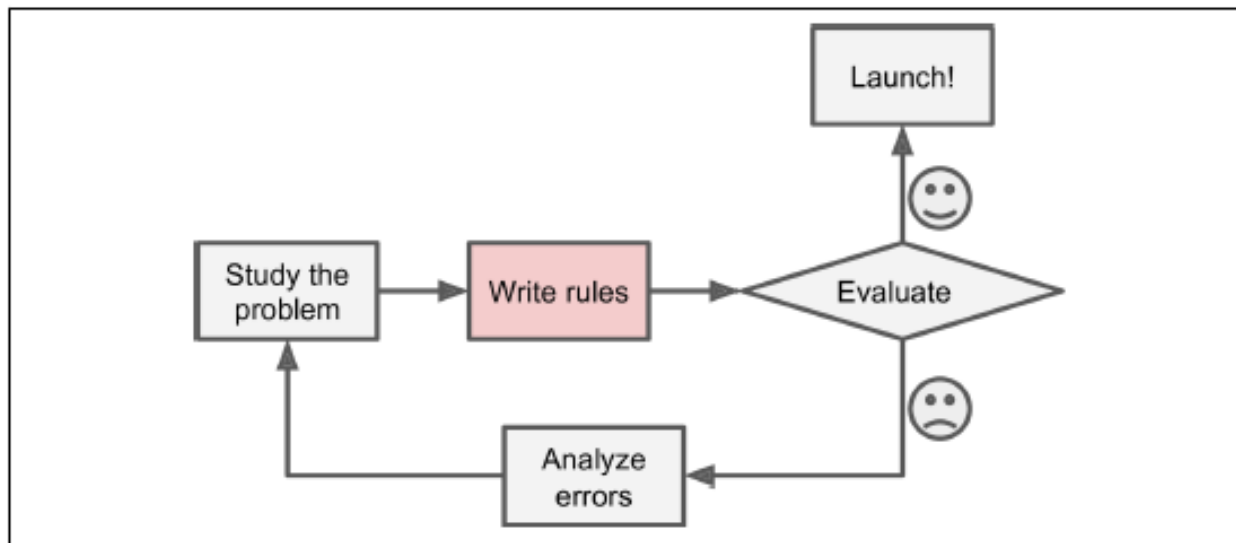
A computer program that will learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .

—Tom Mitchell, 1997

Machine learning (ML) provides a means by which programs can infer new knowledge from observational data.

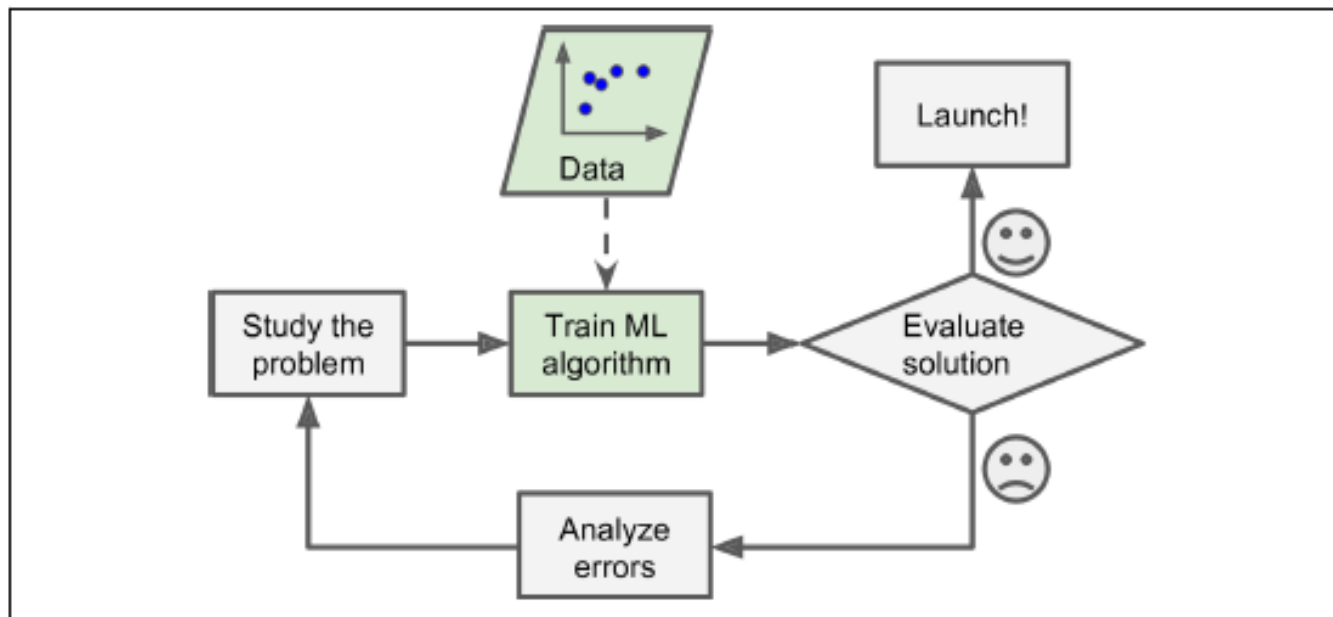
Why Use Machine Learning?

- Let's take a basic problem such as building a **spam filter**.
- We could attempt to build a spam filter using traditional programming techniques.
- First you would look at what spam typically looks like and observe that certain words tend to occur quite frequently in spam.
- You would write a detection algorithm for each of the patterns that you noticed, and your program would flag emails as spam if a number of these patterns are detected.



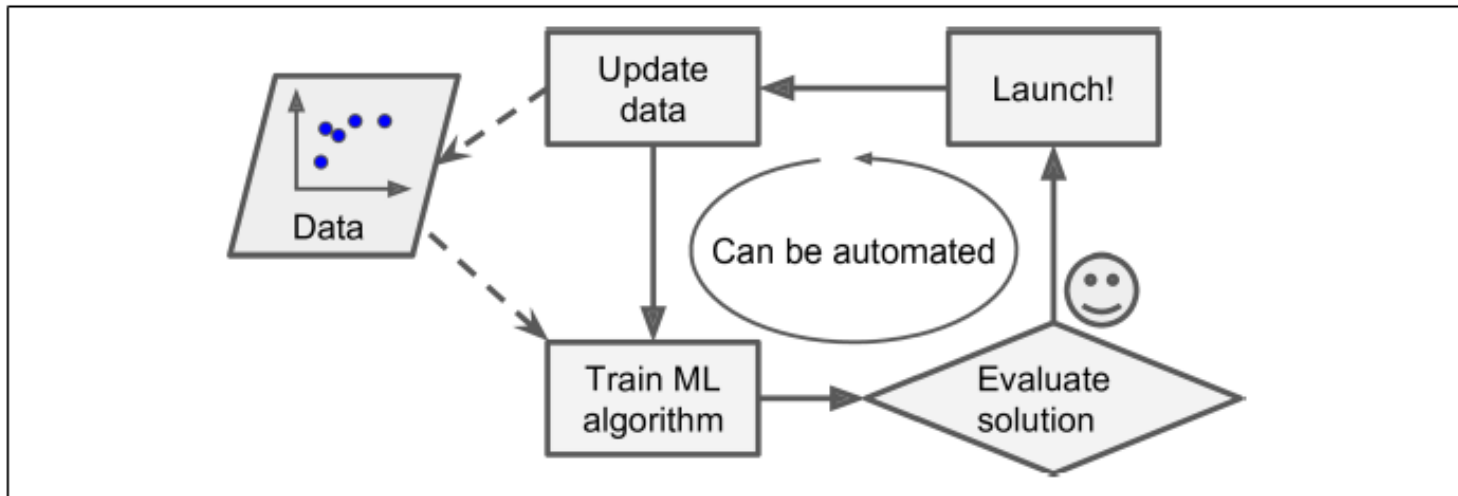
Why Use Machine Learning?

- In contrast, a spam filter based on Machine Learning techniques automatically learns a model (by looking at the words and phrases that are good predictors of spam).
- The program is much shorter, easier to maintain, and most likely more accurate.



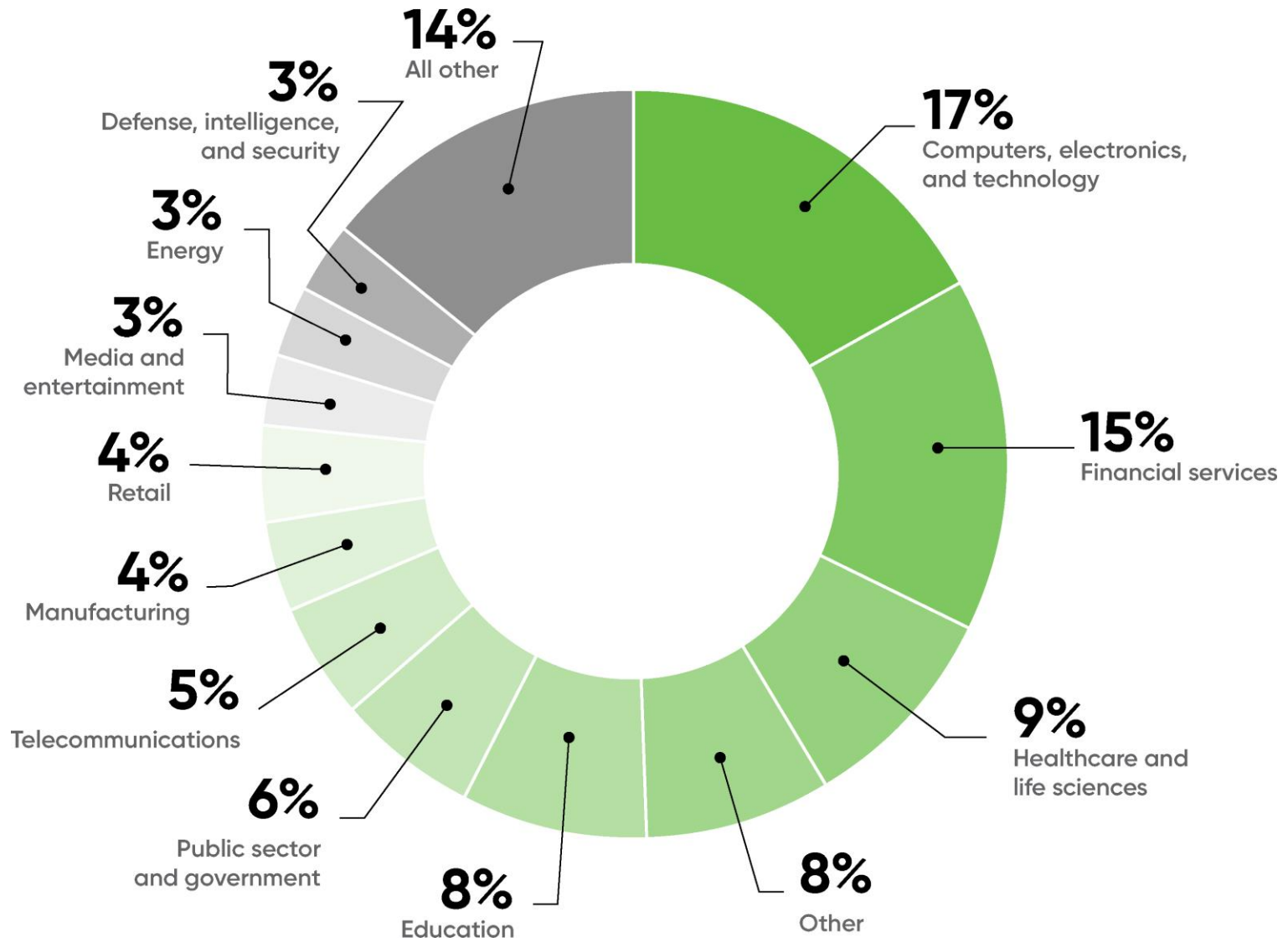
Why Use Machine Learning?

- Building such a system using machine learning also means that we can easily **update our model**.
- It is often necessary to retrain models periodically. This is particularly important in scenarios where there is drift in the data over time.



Applications of Machine Learning

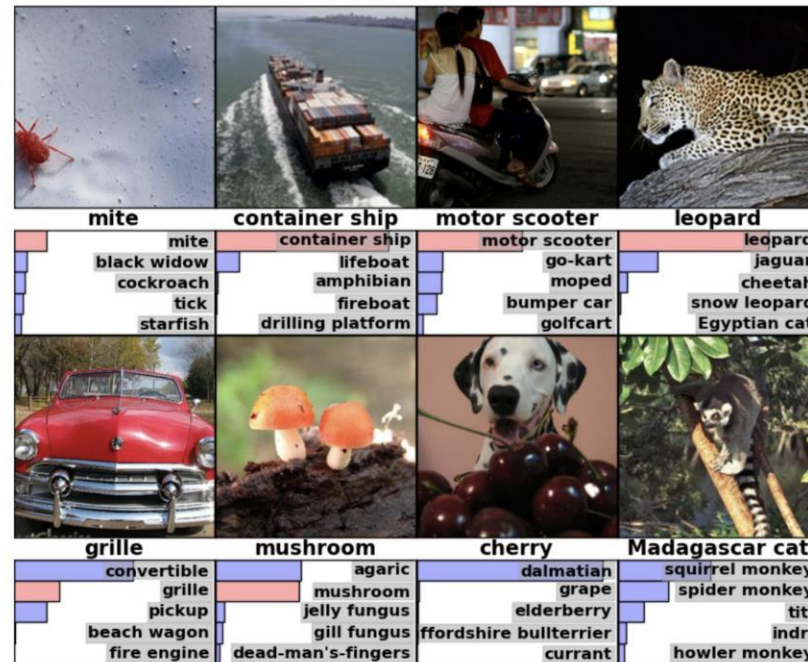
- **Spam:** Host of machine learning algorithms that will learn to classify emails as spam.
- **Natural Language Processing:** Speech recognition, machine translation
- **Recommender Systems:** Netflix, Amazon, Google all use recommender systems (Collaborative and Content Filtering, Marketing).
- **Manufacturing and Robotics:** Manufacturing – Quality inspection, predictive maintenance, etc, Robotics - Recognition of objects , navigation.
- **Commercial/Finance:** Applications include trading agents that interact with the stock markets. Sentiment Analysis, Forecasting and Prediction
- **Navigation:** Research in self-driving cars goes back to early 1990's. From Alvin and Stanley (212 km course, 2005) to Google's Self Driving Car.
- **Medicine:** Medical applications can provide decision support systems for assisting in the diagnosis of patients or identification of particular illnesses.



ImageNet Challenge



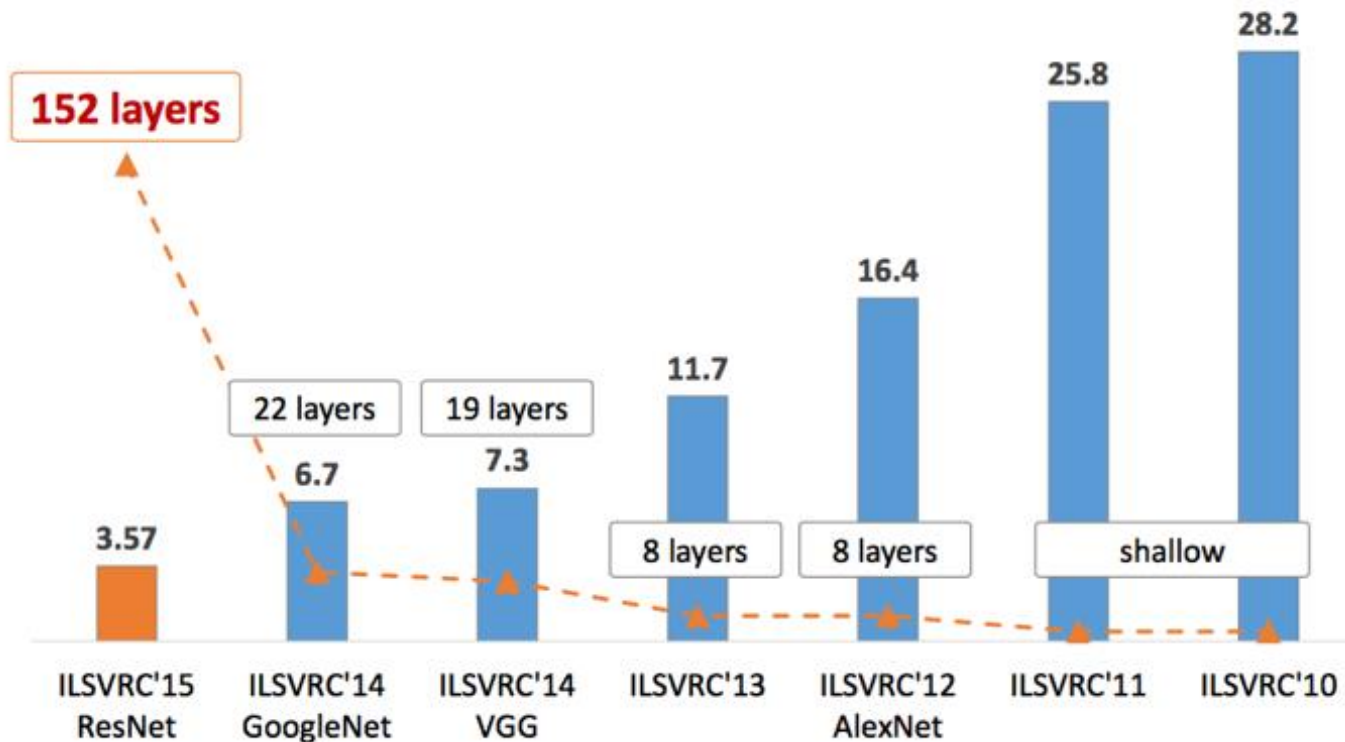
- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.

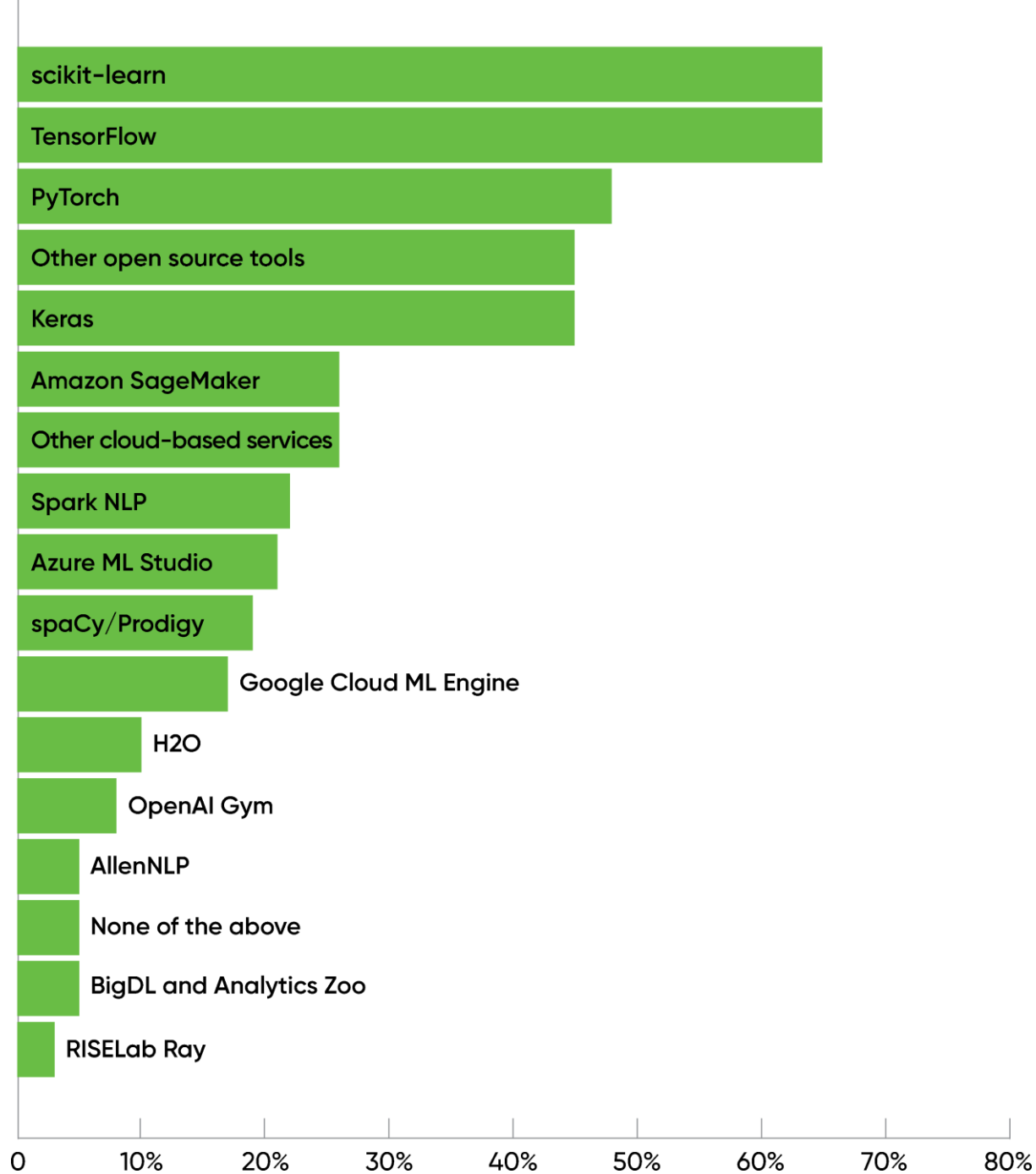


- The annual ImageNet competition began in 2010 where software programs compete to correctly classify and detect objects and scenes.
- In 2012 a submission called **AlexNet** achieved a **top-5 error of 16%**, more than 10.8 percentage points ahead of the runner up.
- As of 2020 it has been cited over 69,990 times.

ImageNet Challenge

- GoogLeNet (also called Inception V1) won the ImageNet competition in 2014.
- ResNet won the ILSVRC 2015 competition with an incredible 3.6% error rate (human performance is 5-10%).
- In 2017, 29 of 38 competing teams got less than 5% wrong.





Basic Terminology Machine Learning Problem

- **Features (often referred to as attributes or variables)** below are Outlook, Temp, Humidity and Windy
- The **Class (Label)** is Play (for regression often referred to as regression target)
- **We refer to an instance as one row from the dataset.**
- **Inference** – Model takes in unseen feature vector and produces a classification.

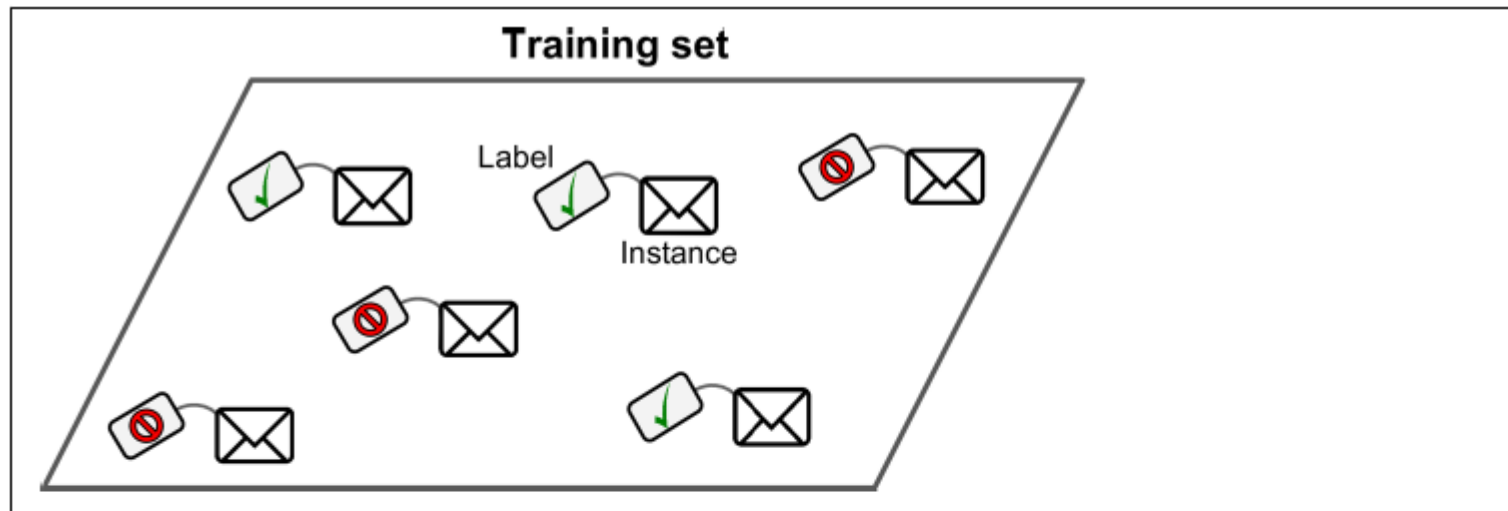
Tennis Dataset					
ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Categories of Machine Learning Algorithms

- Machine learning algorithms can be divided into five main categories
 - Supervised Learning Algorithms
 - Unsupervised Learning Algorithms
 - Semi Supervised Learning Algorithms
 - Self Supervised Machine Learning Algorithms
 - Reinforcement Learning Algorithms

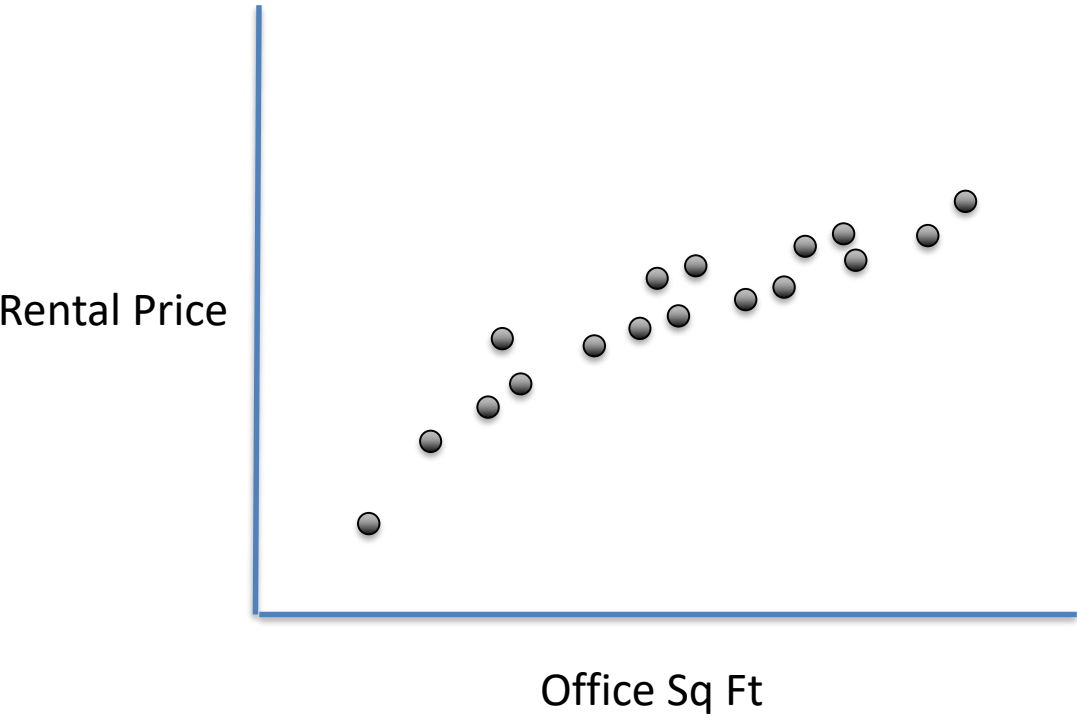
Supervised Learning Algorithms

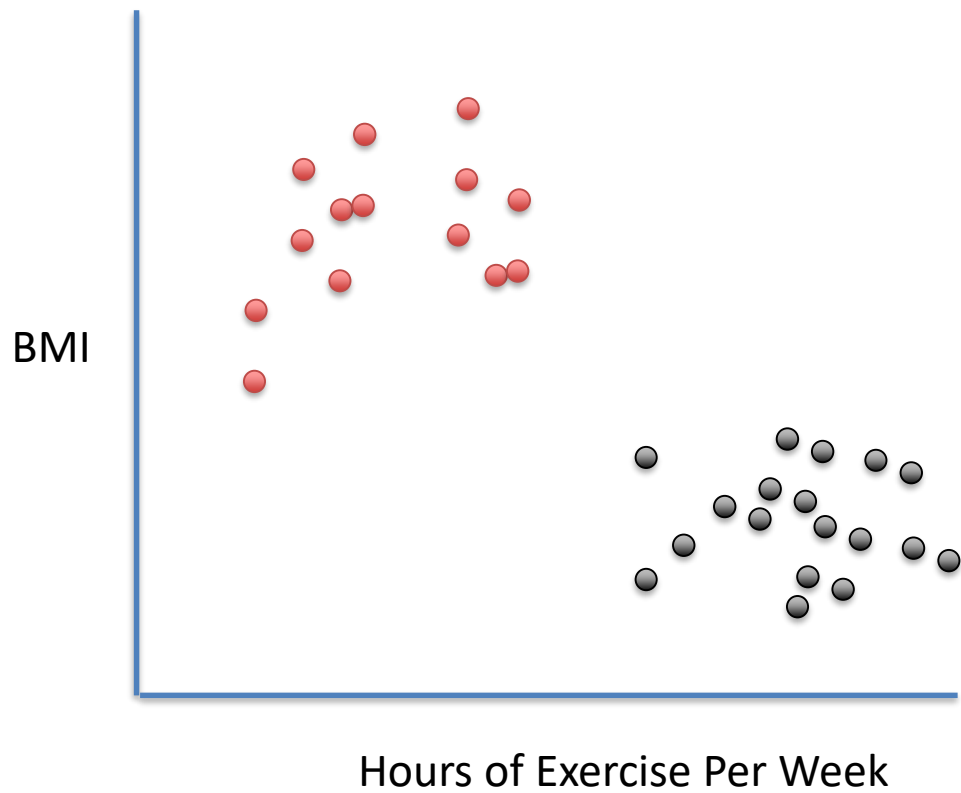
- Supervised learning algorithms used **labelled training data** to learn.
- In other words, the training data you feed to the algorithm includes the desired solutions, called labels.



Supervised Learning Algorithms

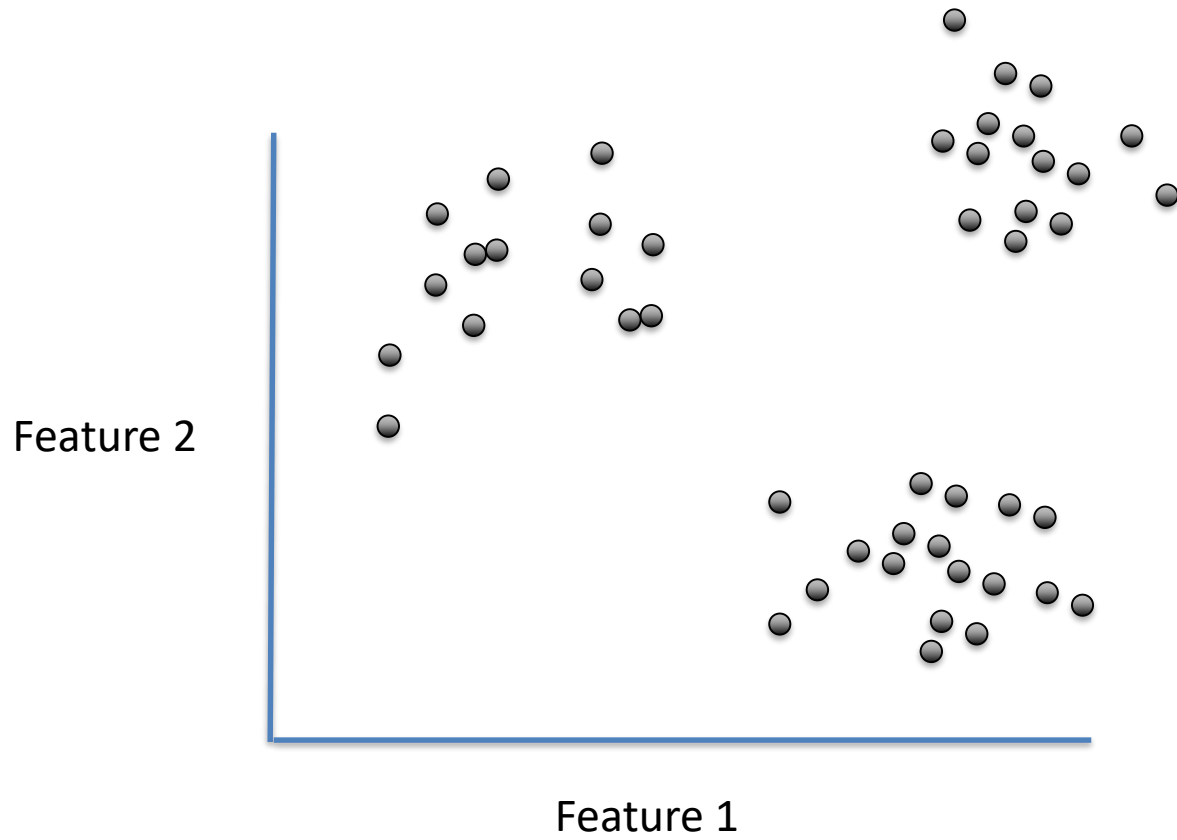
- Supervised learning can be subdivided into either classification or regression algorithms.
- In classification the objective is to correctly **predict the category** that new objects or cases belong to based on specific attributes they have. This decision is based on previous data that you have already observed.
- **Regression** is similar except that rather than predicting a category we want to **predict a numerical value**. For example, predict the concentration of a drug based on a chemical analysis or predict a persons lifespan based on information about their health and lifestyle.

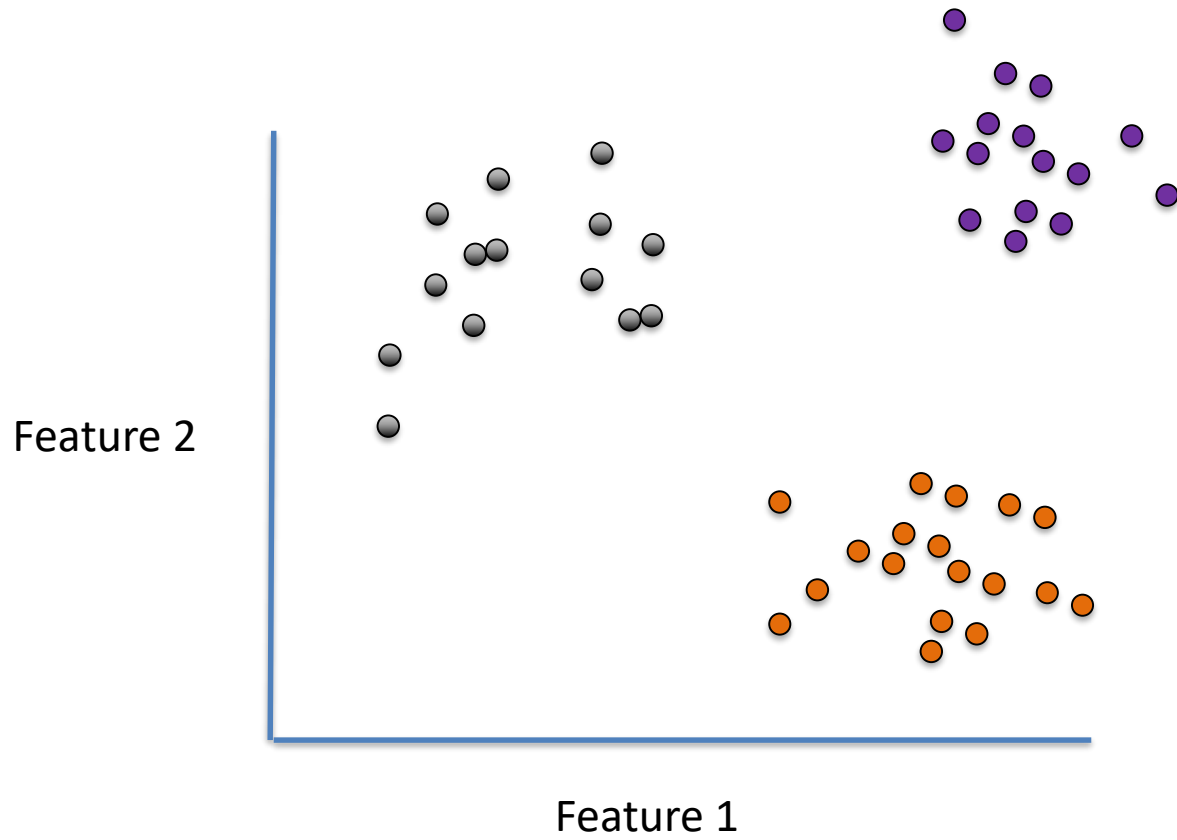




Unsupervised Learning Algorithms

- In unsupervised learning the algorithm is not provided with any labelled training data and **must learn patterns from the data**.
- Unsupervised algorithms seek out **patterns** or **data groupings** in unlabelled data (only features are available).
 - These groups are termed clusters.
 - There are a broad range of clustering machine learning techniques.
 - Example- K Means Clustering (is told in advance how many clusters it should form -- a potentially difficulty)





Applications

- ▶ Google news uses clustering to group new articles with related content. In this case articles related to a the recent whale beaching in Australia

All coverage

 RTE.ie

EU Economy Commissioner confident on new tax deal

Yesterday



THE IRISH TIMES

US multinationals add to pressure on Ireland to agree to a global tax deal

Yesterday



 **breakingnews.ie**

'Constructive' talks on corporate tax reform set to continue, says Taoiseach

5 hours ago



THE IRISH TIMES

Paschal Donohoe takes sitting on the fence to the next level on corporate tax

4 hours ago



 Independent.ie

Donohoe talks tough but Ireland has more to lose outside tax-deal tent

7 hours ago



POLITICO

Ireland seeks tax clarity from US before inking any deal on global minimum rate

15 hours ago



 Reuters

Irish opposition to global tax deal unchanged, watching U.S. closely

20 hours ago · International



T The Times

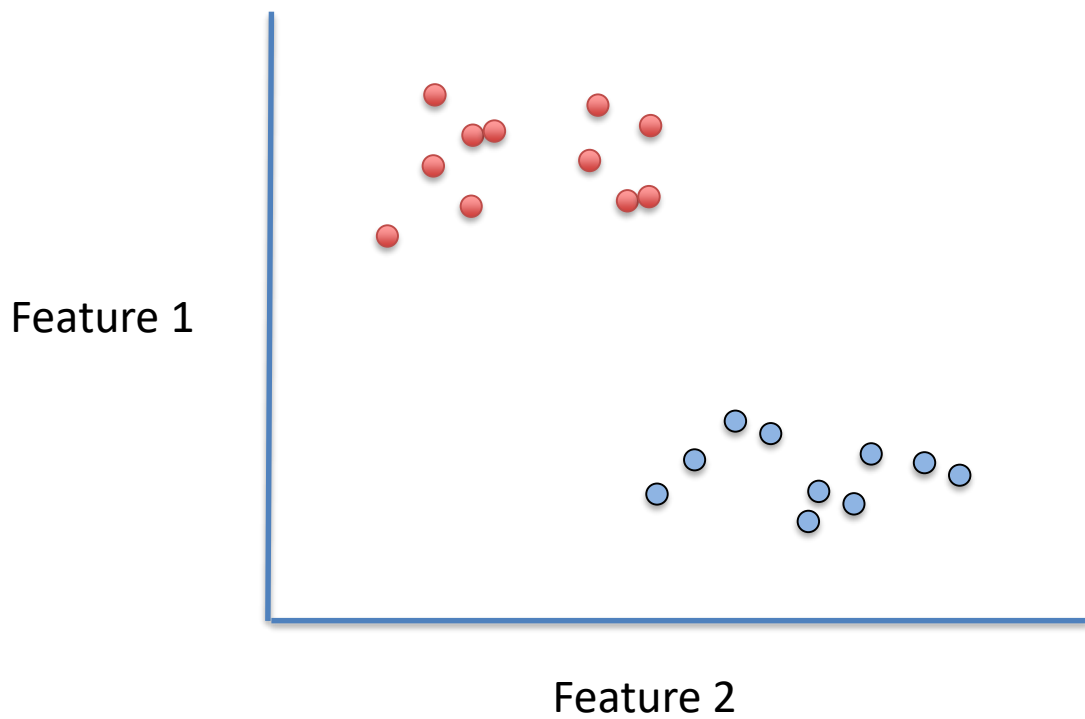
We will only agree OECD 15% corporate tax deal if Ireland benefits, says Donohoe

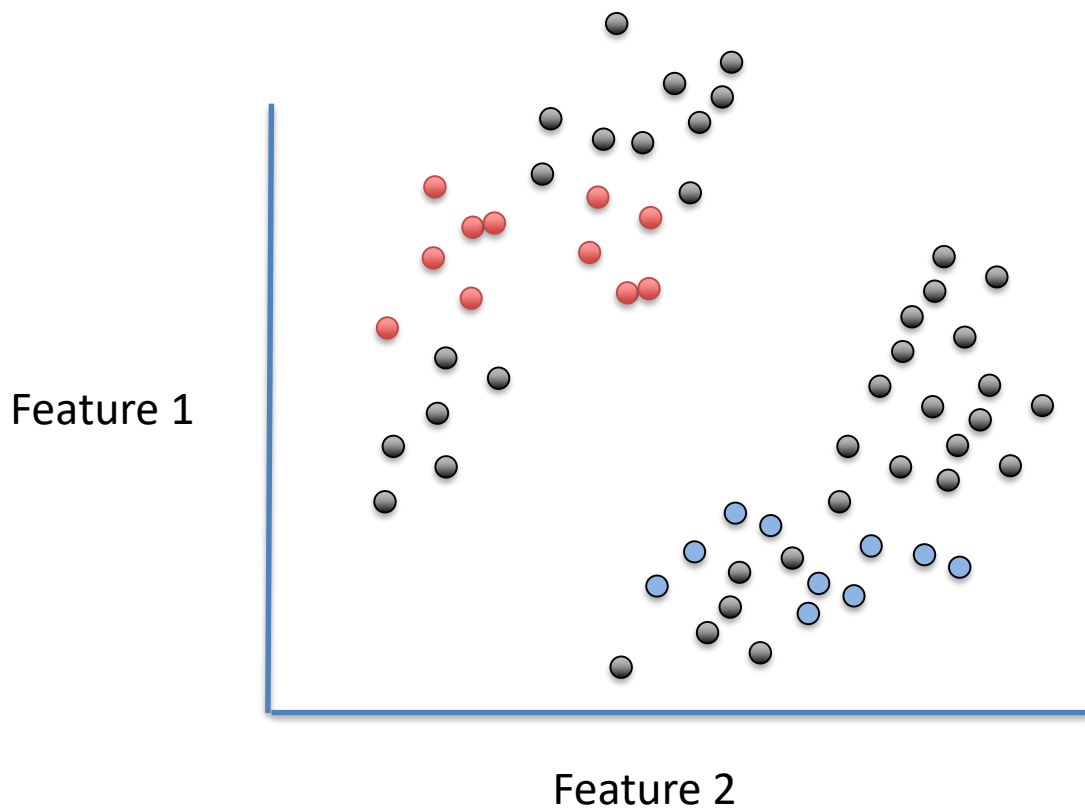
10 hours ago

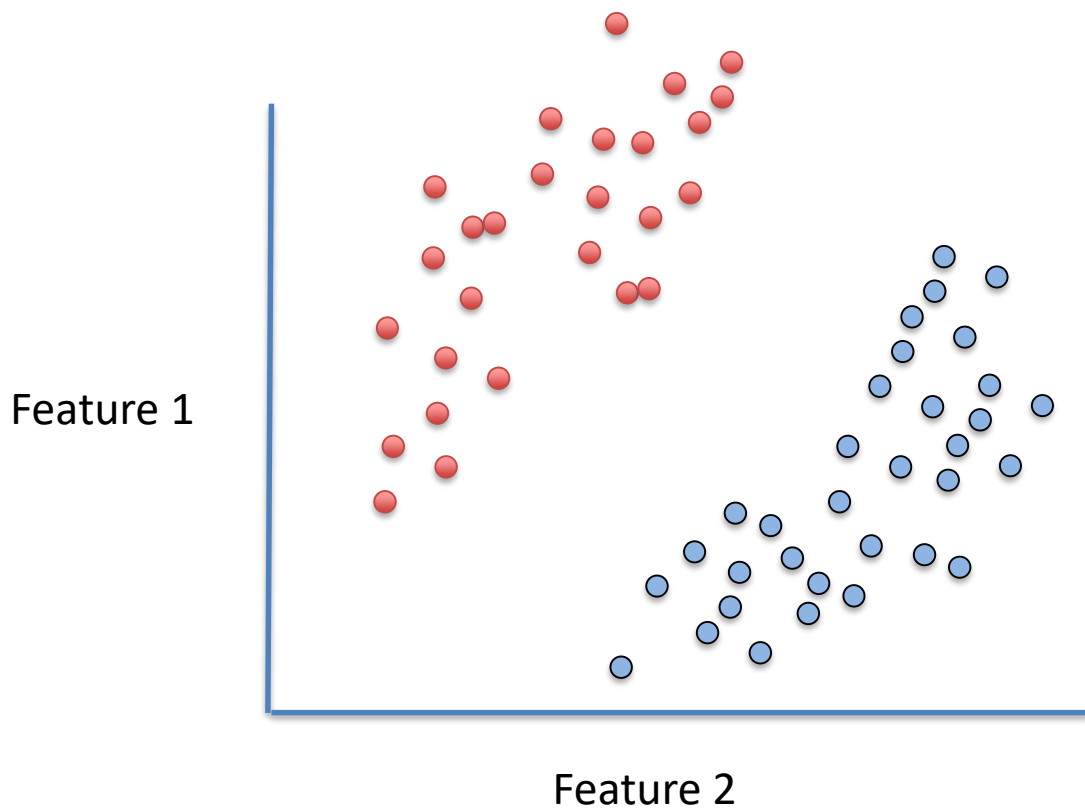


Semi-Supervised Learning

- The semi-supervised learning approach to machine learning **combines** supervised and unsupervised learning techniques.
- Remember supervised learning uses a labelled training set, while unsupervised learning techniques use unlabelled data.
- A semi-supervised approach **utilises both labelled and unlabelled data** for training.
 - Normally a small amount of labelled data is used along with a large amount of unlabelled data







Self Supervised Machine Learning Algorithms

- **Self supervised** learning occurs when we have an **unlabelled** dataset.
- However, the learning process we use is a **supervised learning process**.
- This seems counterintuitive! How can we use supervised learning techniques when the data isn't labelled???
- This is a little more challenging to grasp if you have limited exposure to ML.
- A good example of self supervised learning is a standard **generative adversarial network** (see next few slides).
- [Generative Adversarial Networks can be a little difficult to grasp so don't worry if you don't fully understand this now. We will be covering it in much more detail in the Deep Learning module next semester]

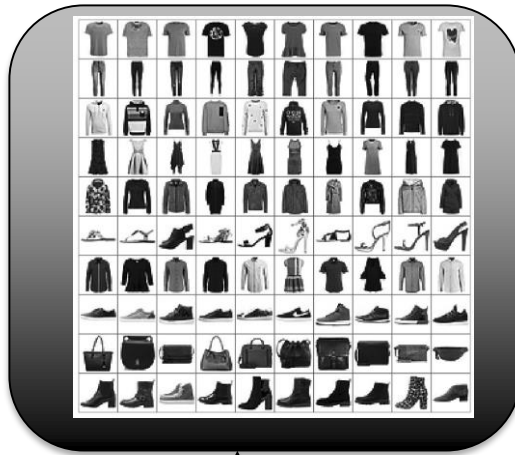
Real Dataset



Real Dataset



Artificial Dataset



Generator Network

The objective of the **generator** is to produce **new images** from the same distribution as the original dataset.

When the generator starts generating images initially they will be very poor. They won't look at all like the images from the real dataset.

We need to teach the generator how to produce authentic looking images.

To do this we will use a **supervised learning** algorithm.

But don't supervised learning algorithms take in labelled data?

Real Dataset



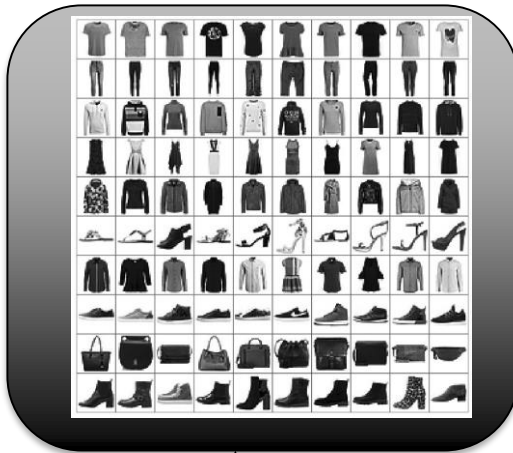
Class Label 1

To do this we will use a supervised learning algorithm. But don't **supervised** learning algorithms take in **labelled** data?

Correct.

So we label all the images in the true dataset as 1 and all the images in the artificial dataset as 0.

Artificial Dataset



Class Label 0

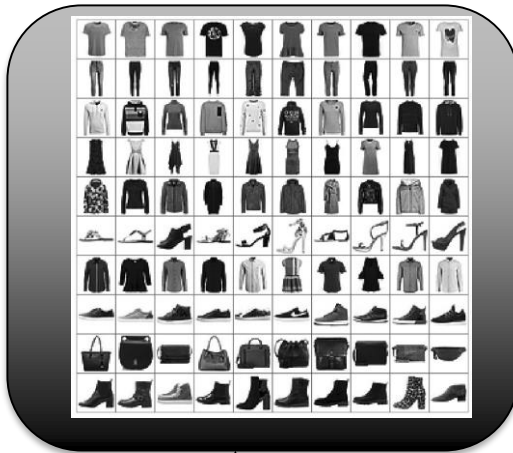
Generator Network

Real Dataset



Class Label 1

Artificial Dataset

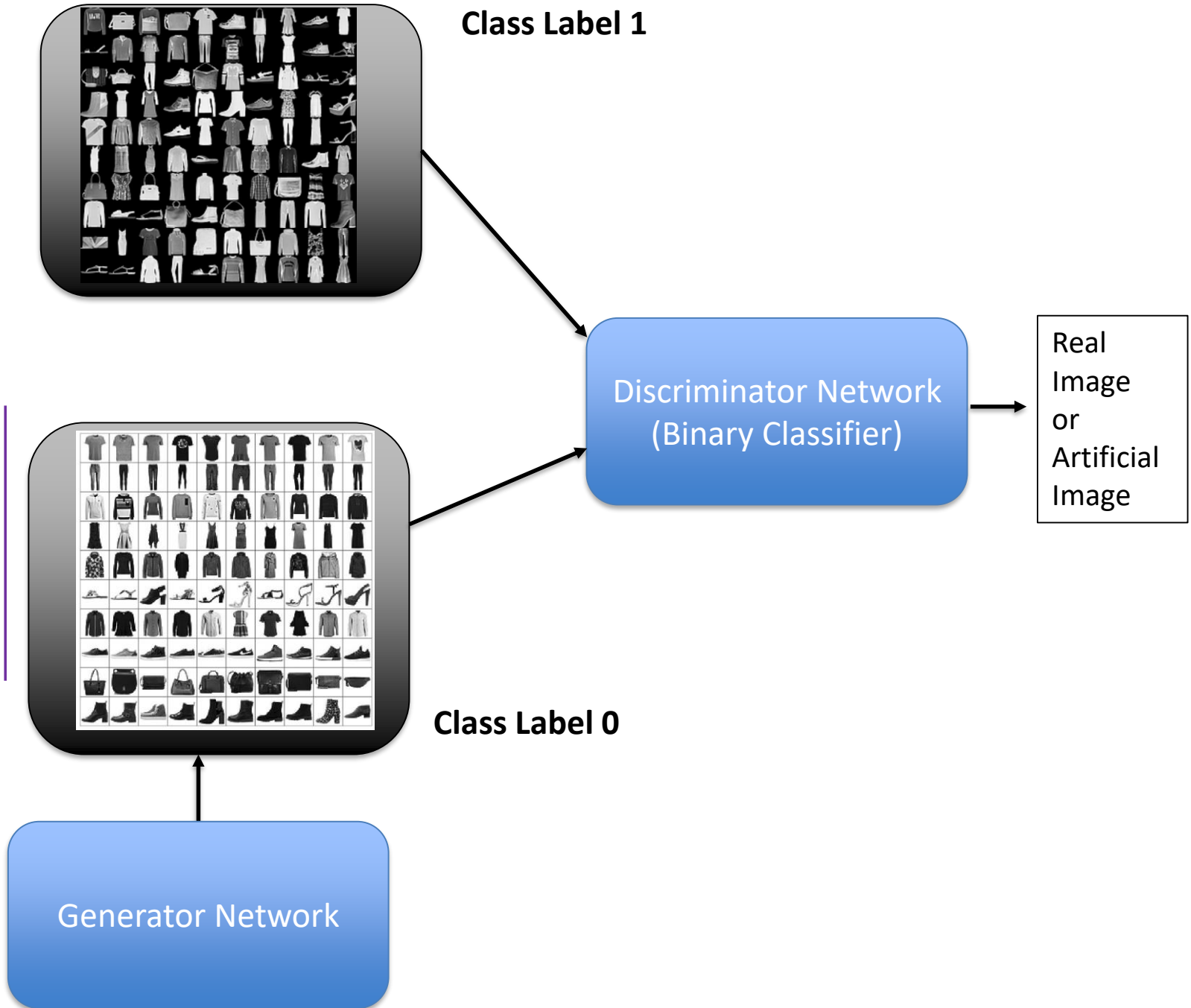


Class Label 0

Discriminator Network
(Binary Classifier)

Real Image
or
Artificial Image

Generator Network

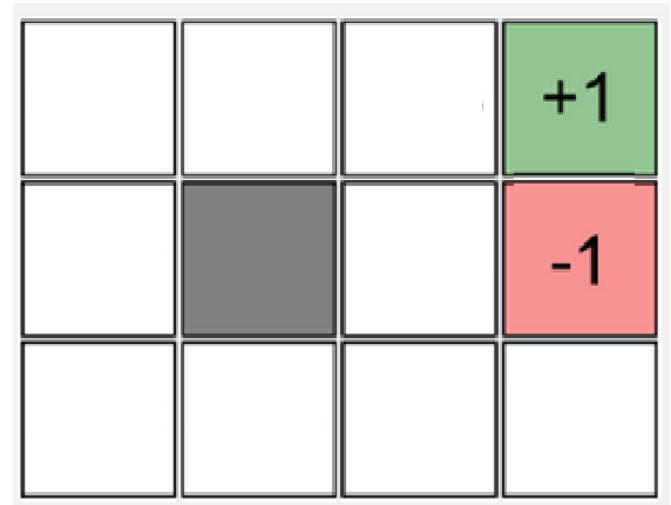


Reinforcement Learning Algorithms

- The objective of Reinforcement learning algorithms is to utilize observed rewards to **learn an optimal (or near-optimal) set of actions (or policy)** for each state in a given environment.
- Unlike most other forms of machine learning the learner **knows nothing about their environment** and just has a set of available actions that it can take.
- The learner is not told which actions to take, but instead must **discover which actions yield the most reward** by trying them in the environment.
- This process continues until it reaches a positive or negative terminal or goal state. It then **positively or negatively reinforces** the actions that led to that state.
- Through repeated interaction with it's environment the agent develops a policy, which **defines what action the agent should choose when it is in a given situation.**

Reinforcement Learning Algorithms

- Take the simple environment on the left. An agent knows nothing about this environment but wants to navigate to the successful state.
- They must start by exploring their environment. They have a certain number of actions available to them. (Up, down, left or right). Each action will take them to a new date on the grid.
- The **intended direction of movement occurs with a probability of 0.8**. With a 0.2 probability you will make a move at right angle to the intended direction. The terminal states have rewards of +1 and -1 respectively.
- **A reinforcement learning program will play this game many times and will develop and optimal policy over time.**



Reinforcement Learning Algorithms

- Take the simple environment on the left. An agent knows nothing about this environment but wants to navigate to the successful state.
- They must start by exploring their environment. They have a certain number of actions available to them. (Up, down, left or right). Each action will take them to a new date on the grid.
- The intended direction of movement occurs with a probability of 0.8. With a 0.2 probability you will make a move at right angle to the intended direction. The terminal states have rewards of +1 and -1 respectively.
- **A reinforcement learning program will play this game many times and will develop and optimal policy over time.**

0.812	0.868	0.918	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

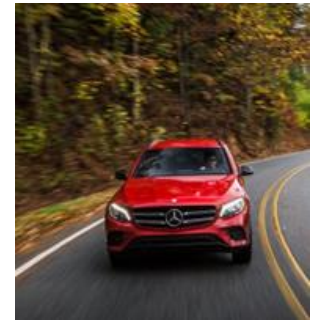
Challenges in Machine Learning

- There are a range of challenges that you may encounter when attempt to build a machine learning model and these can be largely categorized into either **data-based issues** or **model based issues**.
- **Insufficient Amount of Training Data**
 - To work well ML algorithms commonly need quite a lot of data.
 - Even for very simple problems you may often need many hundreds of training examples, and for complex problems such as image or speech recognition you may need **millions** of examples (unless you can reuse parts of an existing model).
 - In some cases poor model performance could be due to a lack of data. It is important to be able to **diagnose** you ML model to determine if a lack of data may improve it's overall level of accuracy.

Challenges in Machine Learning

- **Non-representative Training Data**

- In order to generalize well, it is crucial that your training data be representative of the new cases you want to generalize to.
- By using a non-representative training set, we will train a model that is unlikely to make accurate predictions.
- This is often harder than it sounds: if the sample is too small, you will have sampling noise (i.e., non-representative data as a result of chance), but even very large samples can be non-representative if the sampling method is flawed. This is often referred to as sampling bias.



Challenges in Machine Learning

- Issues with your Data
 - If your training data is full of errors. For example, your training set may contain some features that have little to no relationship with the class you are trying to predict. There could be **outliers in your data or missing values** (e.g., due to poor quality measurements, faulty sensors, etc).
 - It is very common to spend time cleaning up your training data.
 - For example, if some instances are clearly **outliers**, it may help to simply discard them.
 - If some instances are **missing** a few features (e.g., 5% of your customers did not specify their age), you must decide whether you want to ignore this feature altogether, ignore these specific instances with missing value, fill in the missing values.
 - A critical part of the success of a Machine Learning project is coming up with a good set of features to train on. **Feature selection**: selecting the most useful features to train on among existing features.
 - More on this later in the module.

Model Challenges in ML

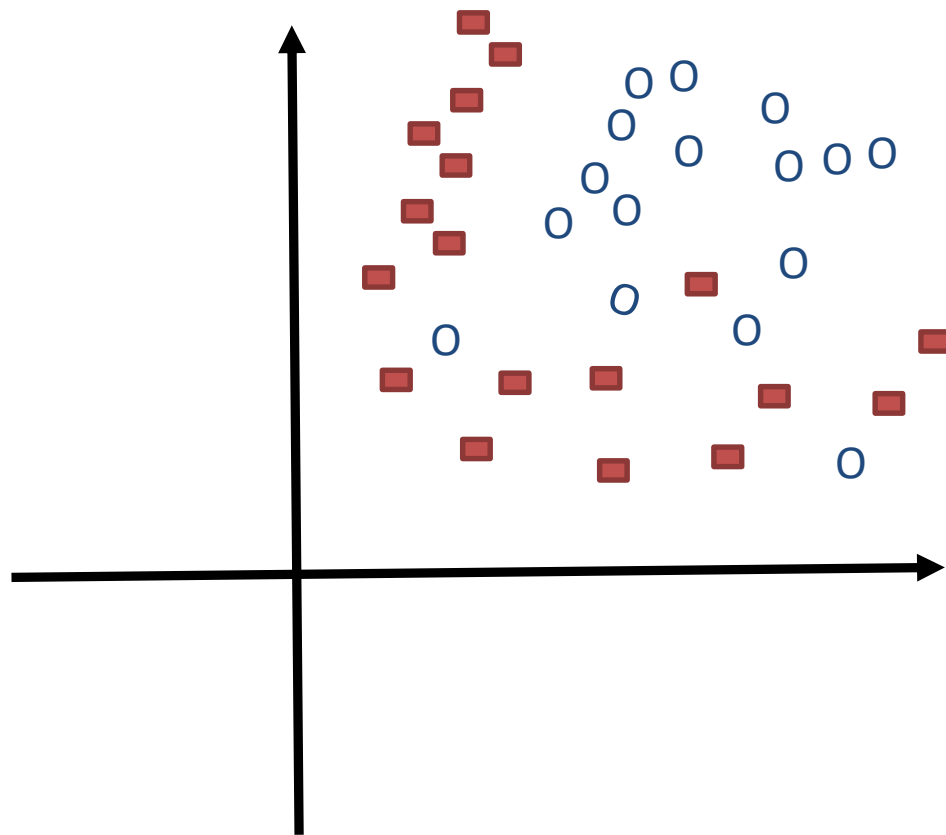
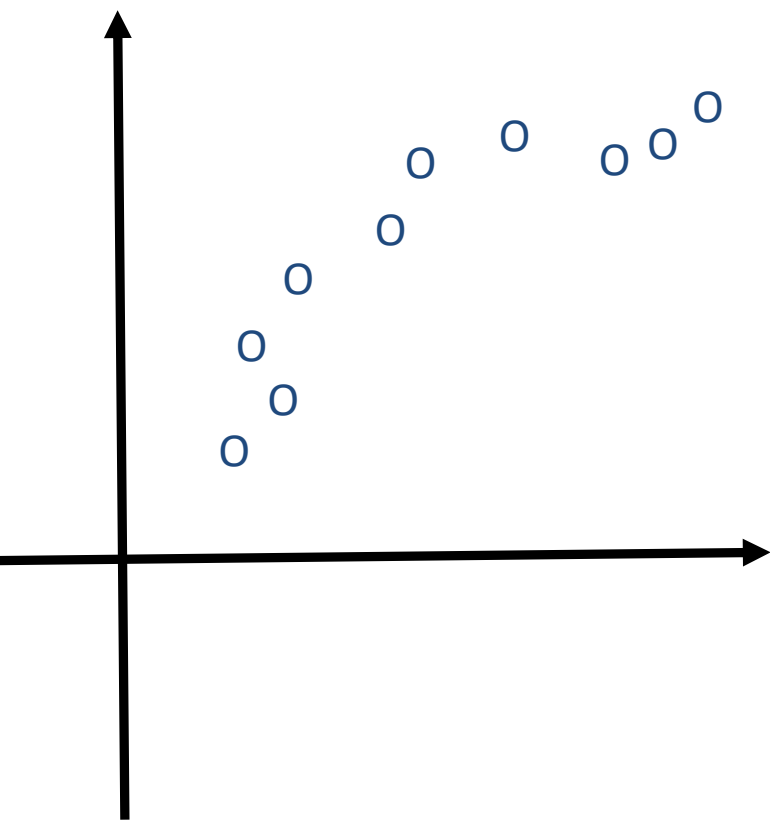
- Issues with your models
- Of course once you address the challenges inherent in the data you must now also tackle the challenges associated with the models.
- There are a broad range of models that we can use. How do we determine **which model to use?**
- Each model has many **parameters** that we can use to tune it's performance. How do we decide on what parameters to set?

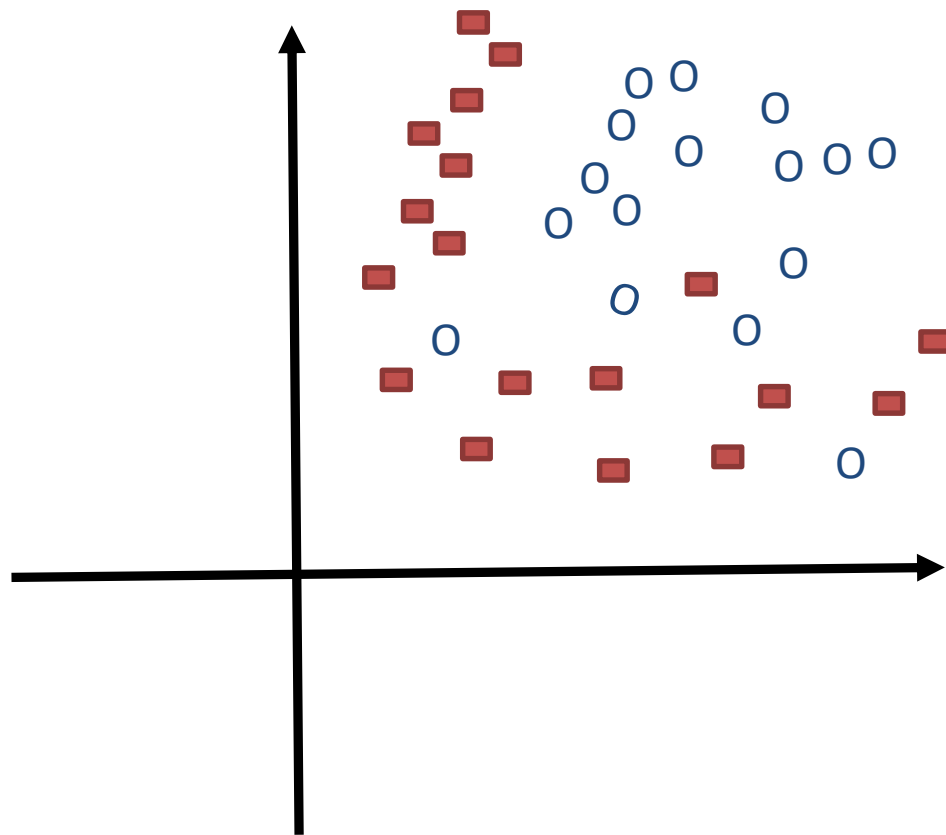
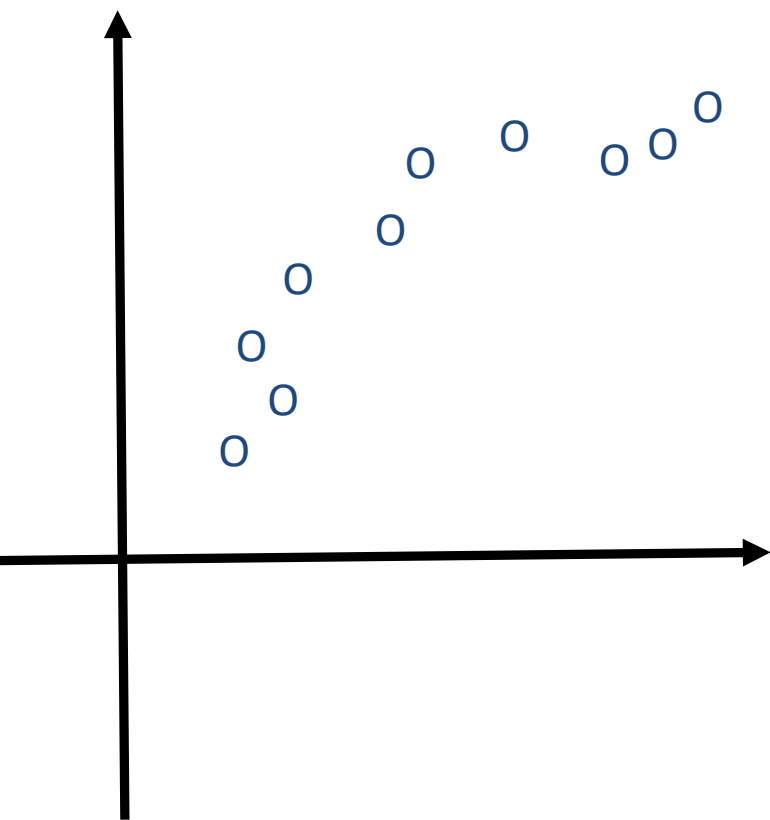
Model Challenges in ML

- When you build a machine learning model you hope that it generalizes well.
- It is very important that you use the **correct methodology** for assessing your model performance. We will talk about this in much more detail later in the module.
- However, one basic rule is that you should **never assess the performance of a model using the data that used to train it.**
- The reason for this is that ML models are very powerful algorithms that can fit the training data very tightly and fail to generalize to unseen data. We refer to this issue as **overfitting**.

Overfitting

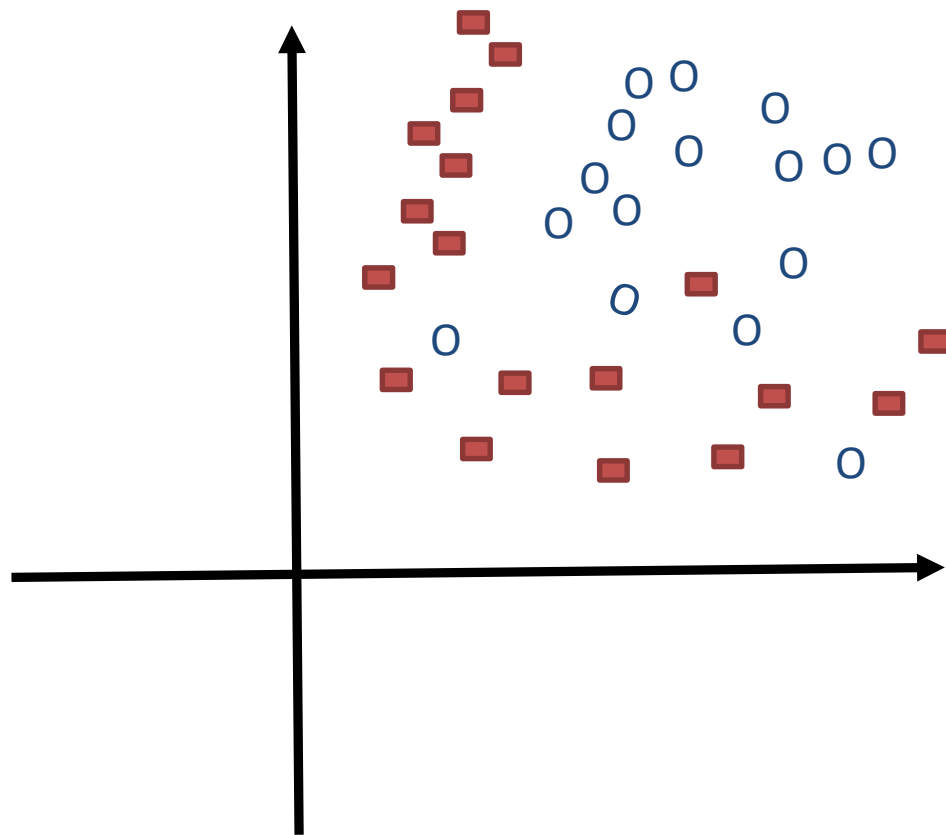
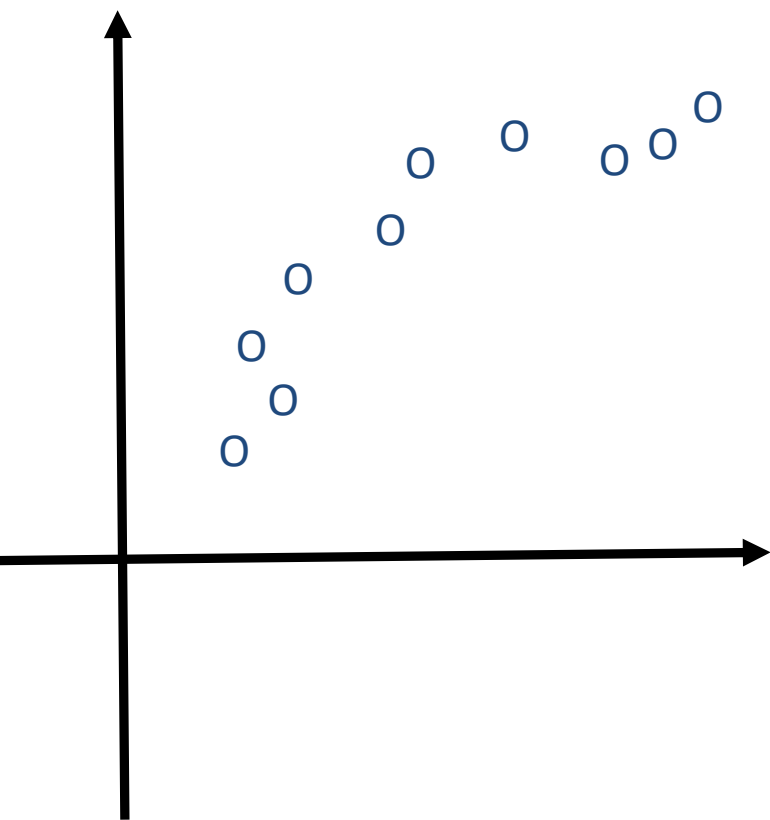
- Overfitting generally occurs when a **model/function is excessively complex and has fit too tightly to the training data.**
- A model/function which has been overfit will generally have poor predictive performance on unseen data (it doesn't generalize well to unseen examples), as it can exaggerate minor fluctuations in the data.
 - A model is typically **trained** by maximizing its performance on some set of training data.
 - However, its overall performance is determined not by its performance on the training data but by its ability to perform well on **unseen data**.
- You can think of this as the difference between memorizing the data and generalizing from the data.





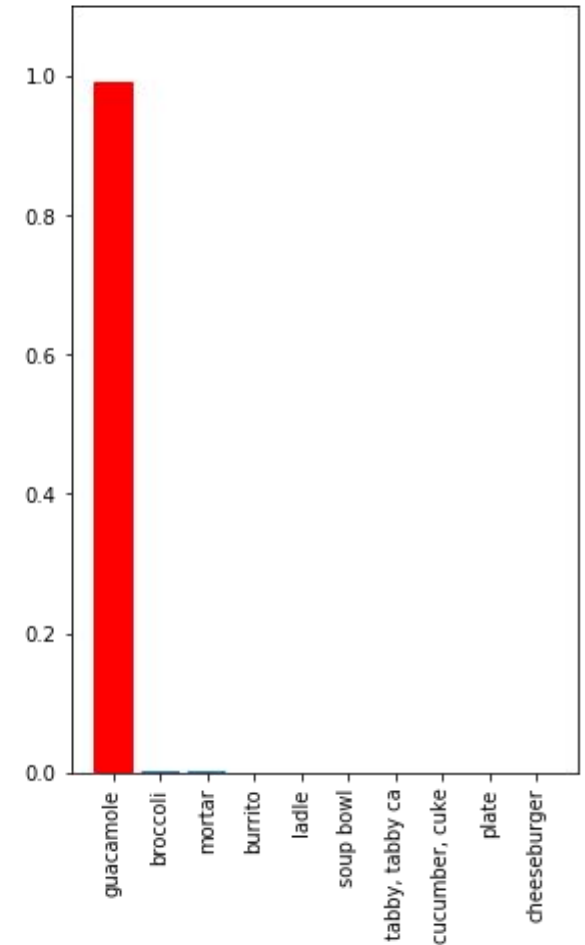
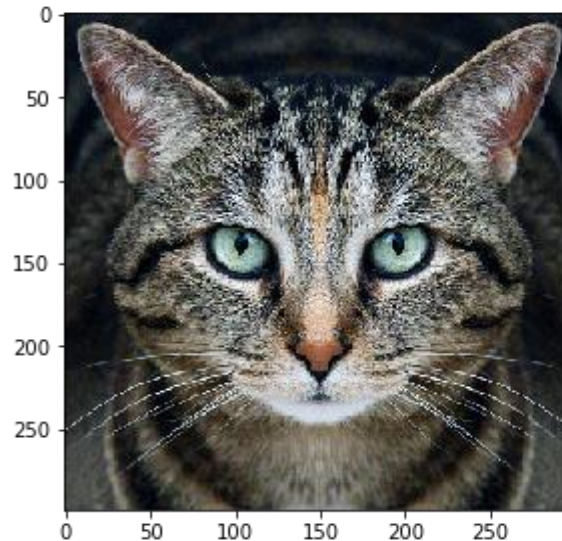
Underfitting

- As you might guess, underfitting is the opposite of overfitting: it occurs when your model is unable to learn the underlying structure of the data.
- Often this issue can be addressed by :
 - Selecting a more **complex model**, with more parameters
 - Feeding **better features** to the learning algorithm (feature engineering)



ML Security Challenge

- While the challenges outlined the previous slides are confined to the ML model and the data there are emergent concerns around the security of ML models.
- For example, research has demonstrated that they can 'fool' deep learning vision systems for perturbing some of the pixels in an image.



[Andrew Ilyas, Logan Engstrom, Anish Athalye, Jessy Lin, Black-box Adversarial Attacks with Limited Queries and Information](#)



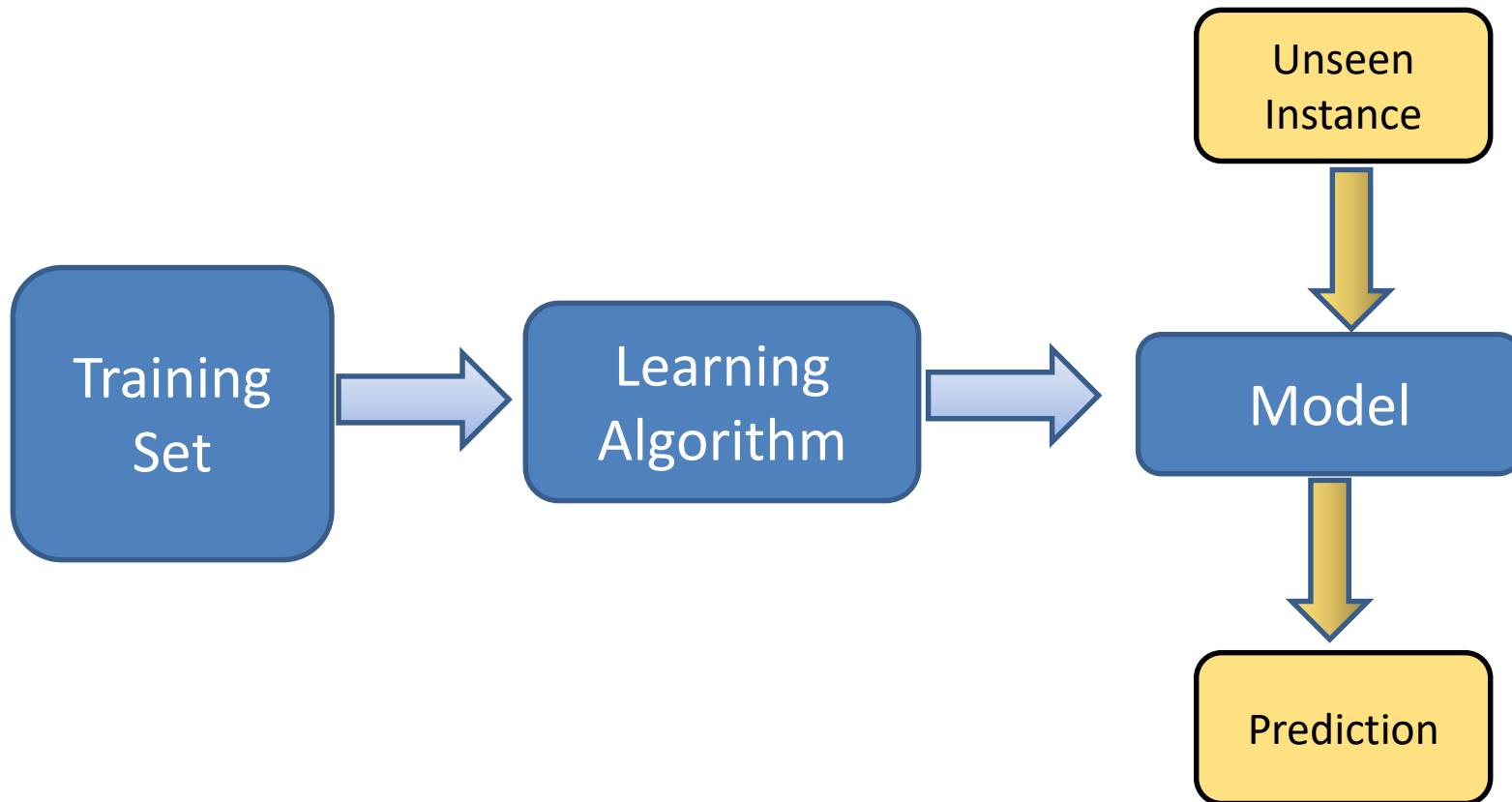
Skiing	91%
Ski	89%
Piste	86%
Mountain Range	86%
Geological Phenomenon	85%
Glacial Landform	84%
Snow	82%
Winter Sport	78%
Ski Pole	75%



Dog	91%
Dog Like Mammal	87%
Snow	84%
Arctic	70%
Winter	67%
Ice	65%
Fun	60%

Summary

- Machine learning (ML) provides a means by which programs can infer new knowledge from observational data.



Different Categories of ML

- *Machine learning algorithms can be divided into five main categories*
- *Supervised Learning Algorithms*
 - *Labelled Data Only*
- *Unsupervised Learning Algorithms*
 - *Unlabelled Data Only*
- *Semi Supervised Learning Algorithms*
 - *Labelled and Unlabelled*
- *Self Supervised Machine Learning Algorithms*
 - *Unlabelled data but supervised techniques*
- *Reinforcement Learning Algorithms*
 - *Learns an optimal policy of actions through interaction with it's environment*

Challenges

- *There are a wide range of challenges you might encounter when developing ML algorithms*
- *Lack of labelled training data*
- *Non representative training data*
- *Which algorithm and parameters do we use?*
- *How to assess a model's performance.*
- *Overfitting and Underfitting*
- *ML Security Challenge*