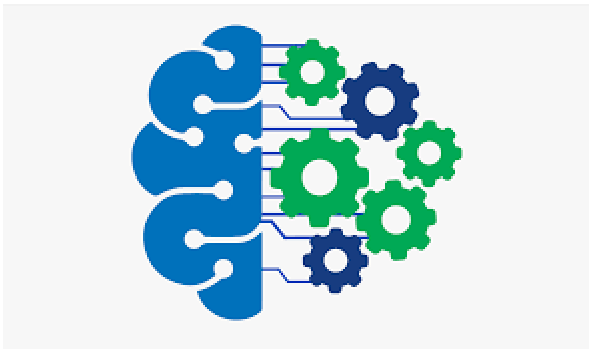


Machine learning

GETTING FIMILLIER WTH ML

Muhammad Naqeeb | ML | July 31, 2021

Contents

Machine Learning

- Machine learning is a field of computer science that aims to teach computers how to learn and act without being explicitly programmed.
- Machine learning is the study of computer algorithms that improve automatically through experience and by the use of data.
- Machine learning is the process of teaching a computer system how to make accurate predictions when fed data.

Those predictions could be answering whether a piece of fruit in a photo is a banana or an apple, spotting people crossing the road in front of a self-driving car, whether the use of the word book in a sentence relates to a paperback or a hotel reservation, whether an email is spam, or recognizing speech accurately enough to generate captions for a YouTube video.

Artificial intelligence (AI)

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions.

Artificial intelligence which simply means a human intelligence exhibited by machines and A.I. is a machine that acts like a human.

Narrow A.I.

That a machines can be just as good or even better than humans at specific tasks.

For example, detecting heart disease from images or at a game of Go or chess or Starcraft and other video games.

Narrow A.I. that we currently have simply means those machines can only do one thing really well they can't be like humans and have multiple abilities.

General A.I.

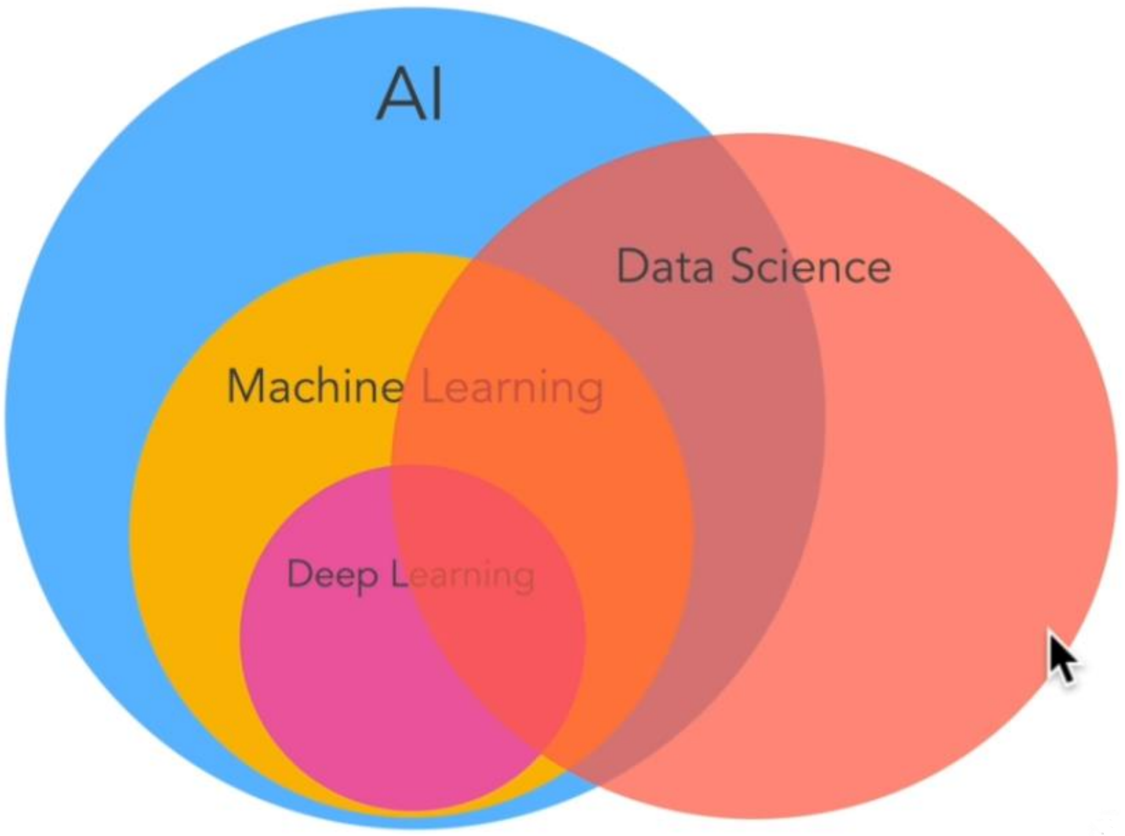
Data Science

Data science is the field of applying advanced analytics techniques and scientific principles to extract valuable information from data for business decision-making, strategic planning and other uses.

Deep Learning

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge

- Deep learning is an AI function that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions.
- Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.



Introduction to Machine Learning

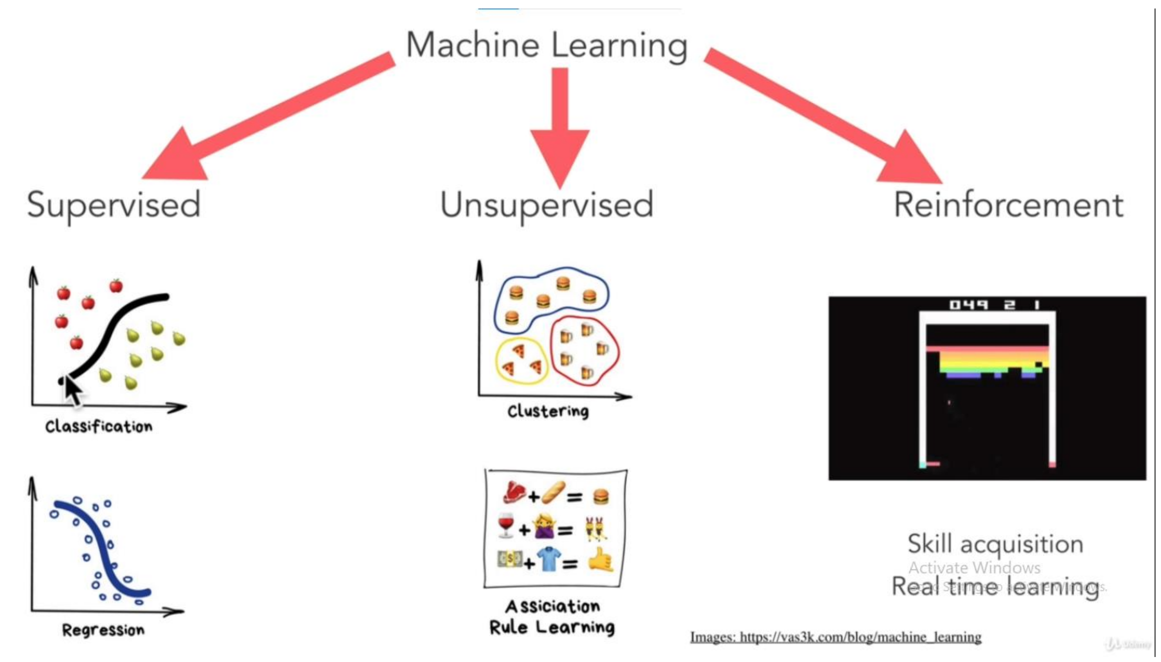
A great Website here is by Google called **teachable machine**. You can built machine learning model in this website.

In machine learning you basically provide a data set, it can be photos or any kind of data sets which you provide to your machine. Then you train your machine for certain output. We trained the computer and we create a model that can predict for us based on that data.

There is another Website a **machine learning playground**.

We give inputs to machines and the machine decides and draws a line to figure out what we should predict for a future input.

Types of Machine Learning



At the end of the day the only reason that we care about machine learning is that we're able to use machines to predict results based on incoming data.

Some of the machine learning categories that you often see and keep in mind what I said that is machine learning is simply about predicting results based on incoming data. And all these subcategories simply do that.

Supervised Learning

In this supervised learning. The data that we received already has CATEGORIES.

THINK OF IT AS A CSC/Excel files with rows and columns label.

We have labeled data and a test data that is label so we know if our function is right or wrong.

So in a supervised learning scenario we can do things like ***classification*** to decide is this an apple. Or is this a paper machine learning model simply draws a line to decide Hey this is an apple and this is a pair or we might do something called **regression** instead of classification based on inputs.

For example predicting stock prices another way that we might use supervised learning is for example to hire engineers based on inputs based on years of experience based on maybe age maybe where they live what type of computers they have all these sorts of inputs that are labeled can be used in a supervised learning system to decide should I hire this engineer or should I not.

Un-supervised Learning

Now sometimes we have data that doesn't have labels and this is called on supervised learning.

Again think of it as a CSP/Excel file without perhaps the column names labeled sometimes with things like ***clustering.*** We need to create these groups or at least the machine to create these groups. For example we give it a bunch of data points and then the machine decides oh this is a group.

This is a group and this is a group or we can have something like ***association rule*** learning where we associate different things to predict what a customer perhaps might buy in the future when groups don't exist. We call it unsupervised learning.

We can't tell the machine that they are right or wrong like we can when we do apples versus pears since there are no true categories but we let the machines just create these categories for us.

Reinforcement learning

Reinforcement learning is really interesting because it's all about teaching machines through trial and error through rewards and punishment so the program simply learns a game by playing it millions of times until well it gets the highest score it doesn't know what it's doing at first but then it tries to maximize the score and eventually figures out that hey maybe I should try and move where the ball is coming. So this is seen for skill acquisition or real time learning and you see it a lot in machine learning programs that play for example video games.

The key thing all of these what they're doing is trying to learn from the data that it receives and predict something.

Machine learning in a nutshell

Computers are really good at processing large amounts of data. Data that we're creating more and more of machine learning lets computers make decisions about data.

It lets computers learn from data and they make predictions and decisions what product does a YouTube recommend, Does this person have a heart disease, Is this e-mail spam all use machine learning.

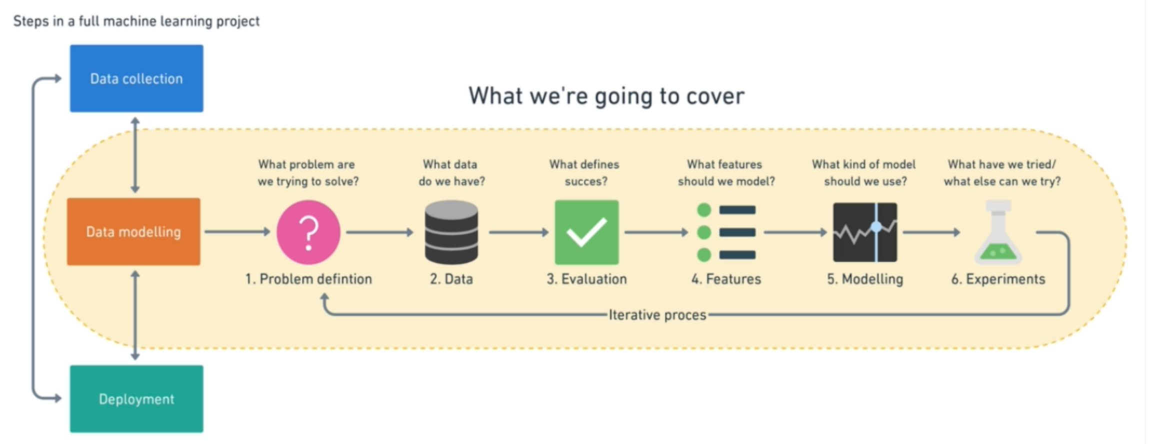
So we essentially let computers decide for us.

But machine learning at the end of the day is just a general term for when computers learned from data. It allows computers to do tasks that in the past required humans and make our lives hopefully easier.

6 Step Machine Learning Framework

Machine learning projects can cover many different topics.

It's important to design a framework you can use to approach different kinds of problems. So when you come up against a problem you can refer back to this field guide and go.



Steps in a full machine learning project
Data collection
What we're going to cover
Data modelling
Deployment
What problem are we trying to solve?
1. Problem defintion
What data do we have?
2. Data
What defines succes?
3. Evaluation
What features should we model?
4. Features
What kind of model should we use?
5. Modelling
What have we tried/ what else can we try?
6. Experiments
Iterative proces

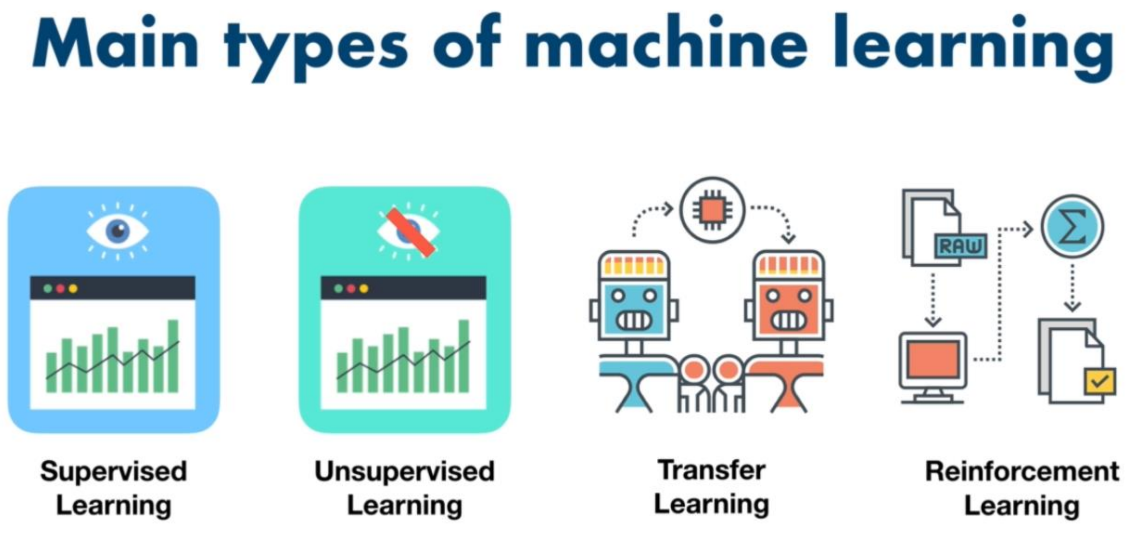
1) Problem Identifying

1. Problem definition
?
"What problem are we trying to solve?"

First step is identifying the problem we're trying to solve as a machine learning problem.

Before we get into different types of machine learning problems it's important to note machine learning isn't the solution to every problem.

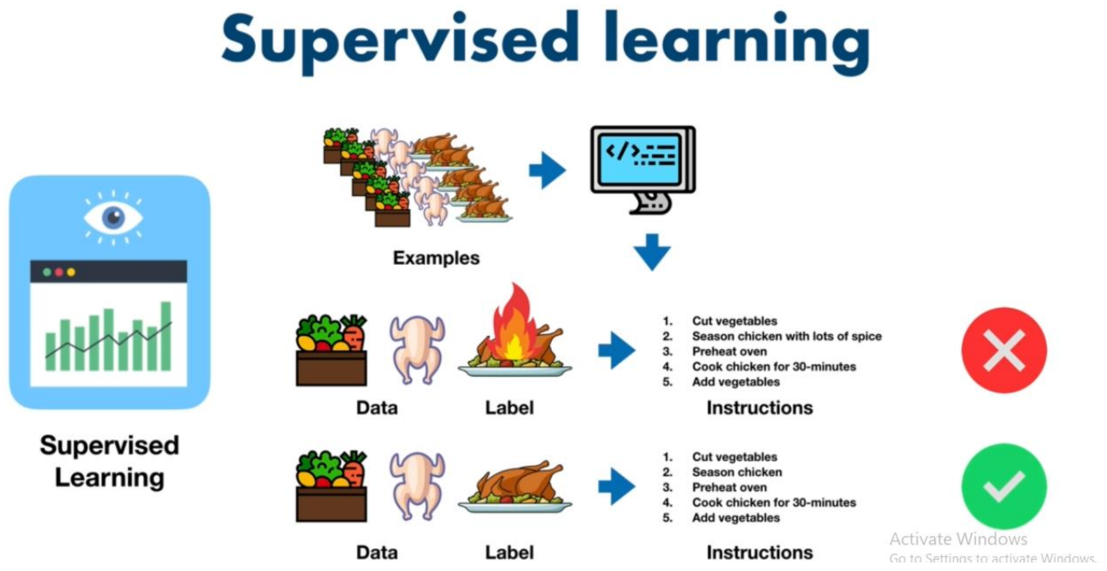
Main types of machine learning problem are **supervised learning**, **unsupervised learning**, **transfer learning** and **reinforcement learning.**



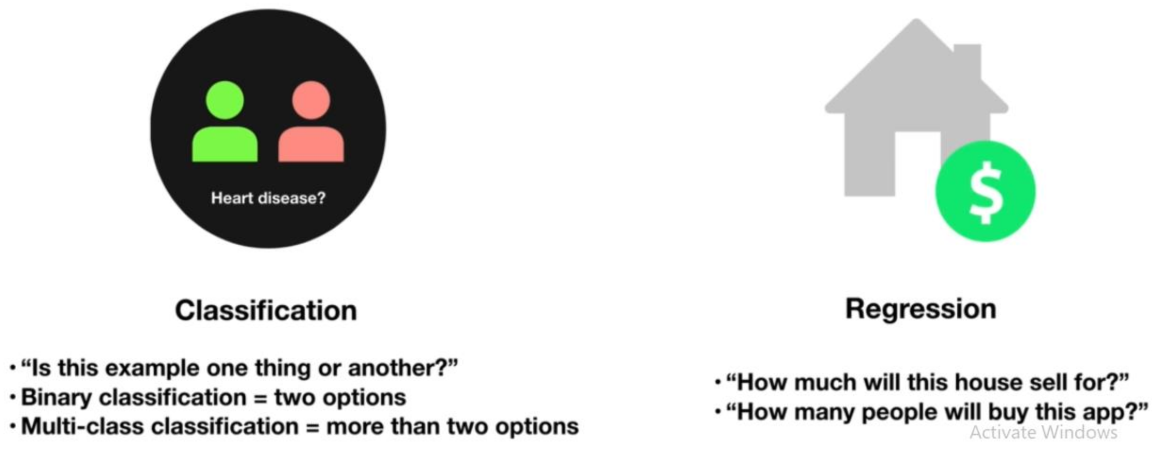
Supervised Learning

In supervised learning you have data and labels, a machine learning algorithm tries to use the data to predict a label. If it guesses the label wrong the algorithm corrects itself and tries again. This act of correction is why it's called supervised.

A supervised learning algorithm repeats this process over and over and over again trying to get better.



The main types of supervised learning problems a **classification** and **regression**


Heart disease?
Classification
• "Is this example one thing or another?"
• Binary classification = two options
• Multi-class classification = more than two options
Regression
• "How much will this house sell for?"
• "How many people will buy this app?"
Activate Windows

Classification

classification involves predicting if something is one thing or another such as if you wanted to predict whether or not a patient had heart disease or not based on their medical records or what type of dog was in an image if there are only two options. It's called **binary classification**.

If there are more than two options it's called **multi class classification.**

Regression problem

Regression problems involve trying to predict a number you might hear it referred to as a continuous number as well which just means a number which can go up or down a classical regression problem

Basics/classical Regression problem is trying to predict the sale price of a house based on things like number of rooms the area it's in, how many bathrooms it has or trying to predict how many people will buy a new app based on Web site visits and clicks.

Unsupervised Learning

Unsupervised learning has data but no labels. Unsupervised learning is a type of algorithm that learns patterns from untagged data. Unsupervised learning uses machine learning algorithms to analyze and cluster(putting groups of similar examples together) unlabeled data sets.

Recommendation problems such as recommending what music someone should listen to based on their previous music choices often start out as unsupervised learning problems.

Transfer Learning

transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

In transfer learning, the learning of new tasks relies on a previously learned tasks.

Reinforcement Learning

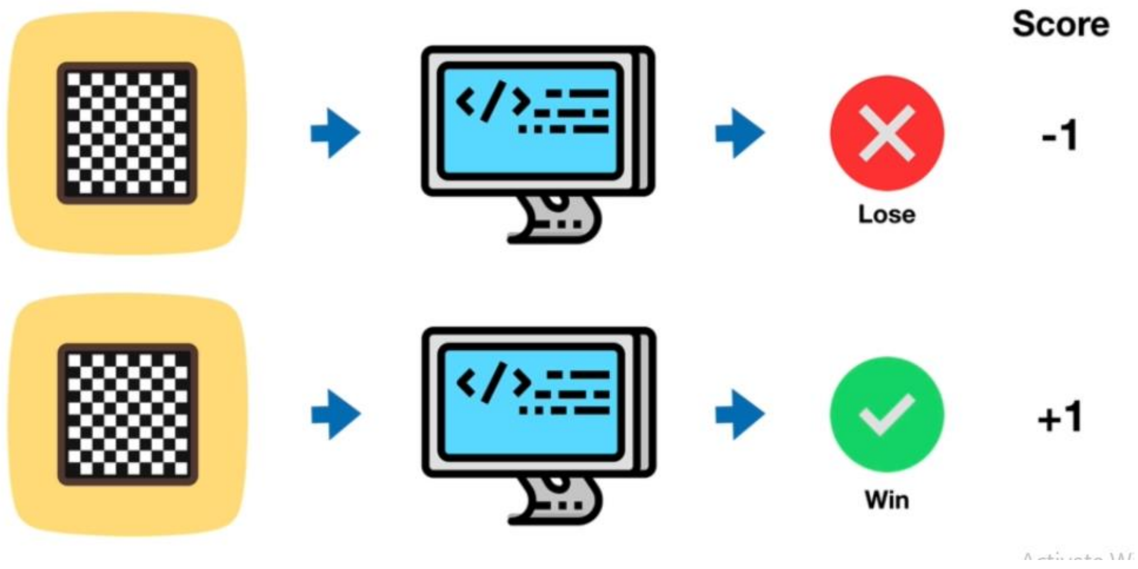
Reinforcement learning involves having a computer program perform some actions within a defined space and rewarding it for doing it well or punishing it for doing poorly.

A good example is teaching a machine learning algorithm to play chess.

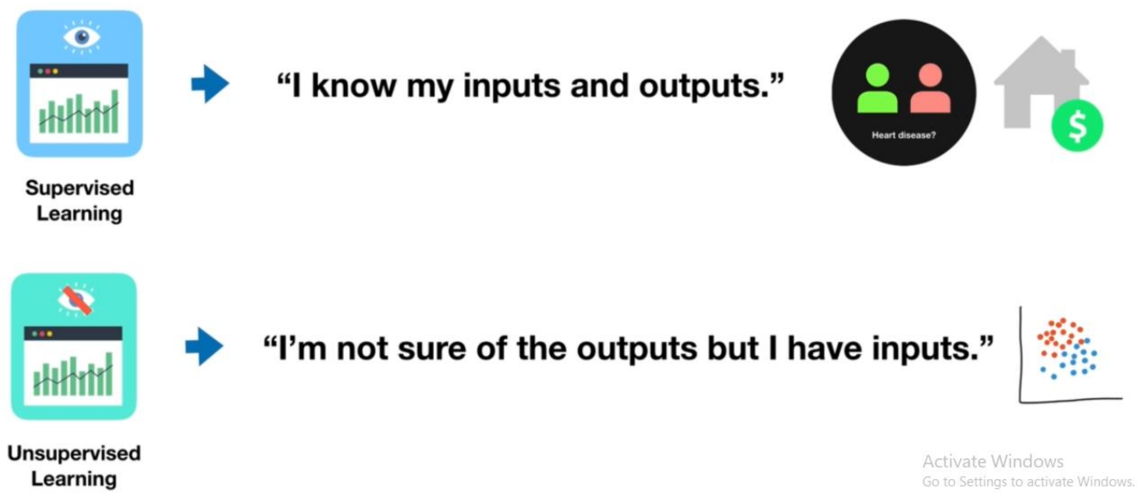
The chess board is a divine space and actions are moving pieces. And when I say punishment or reward these things could be as simple as updating a score with plus one if it wins a negative one if it loses, the machine linings algorithms goal could be to maximize the score.

Reinforcement Learning is what was used for deep mines Alpha go to become the best go.

Reinforcement learning

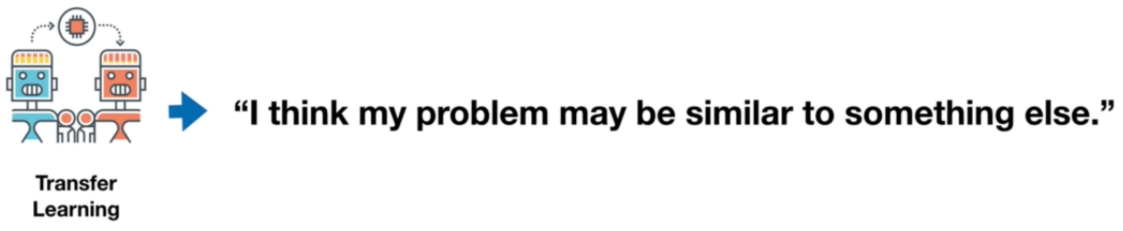


Matching your problem



"I know my inputs and outputs."

"I'm not sure of the outputs but I have inputs."

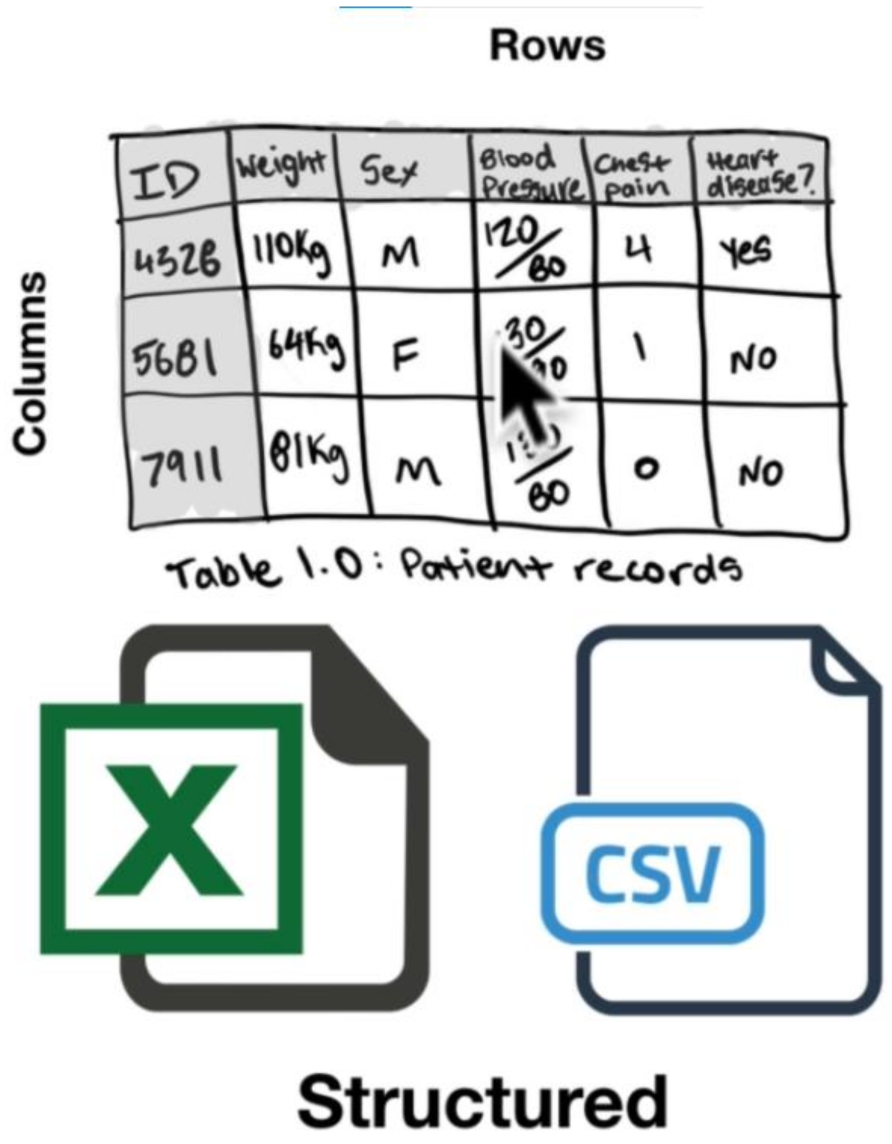


2) Data

Data comes in many different shapes and sizes but the main two types are structured and unstructured.

Structured data

Structured data is something you'd expect to see in an excel file such as rows and columns of different patient medical records and whether or not they have heart disease or not or customer purchase transactions. It's called structured data because all of the samples the different patient records are typically in similar format meaning one column might contain numbers of a certain type such as average blood pressure or sags or weight of a patient and another column might have whether they have chest pain or not and what the level of intensity is.



ID	Weight	Sex	Blood Pressure	Chest pain	Heart disease?
4326	110Kg	M	120/80	4	Yes
5681	64Kg	F	130/[illegible]0	1	NO
7911	81Kg	M	1[illegible]/80	0	NO

Table 1.0: Patient records

Unstructured data

Unstructured data are things like images natural language text such as transcribed phone calls videos and audio files although we can turn these into numbers and create structure.

Now within these two data types there's static and streaming data

Static data

static data is data which doesn't change over time.

You may have a spreadsheet of patient records in a dot CSP format which stands for commas Separated Values which simply means all of the different data is in one file separated by commas.

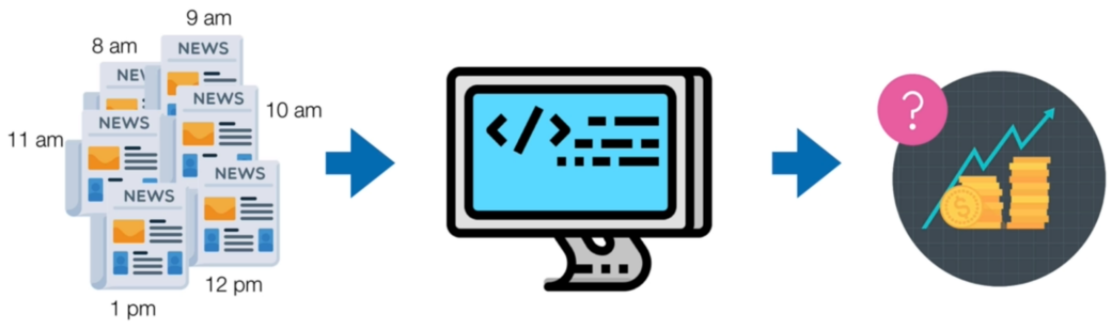
Now CSB is one of the most common types of static data formats.

"There's a saying The more data the better."

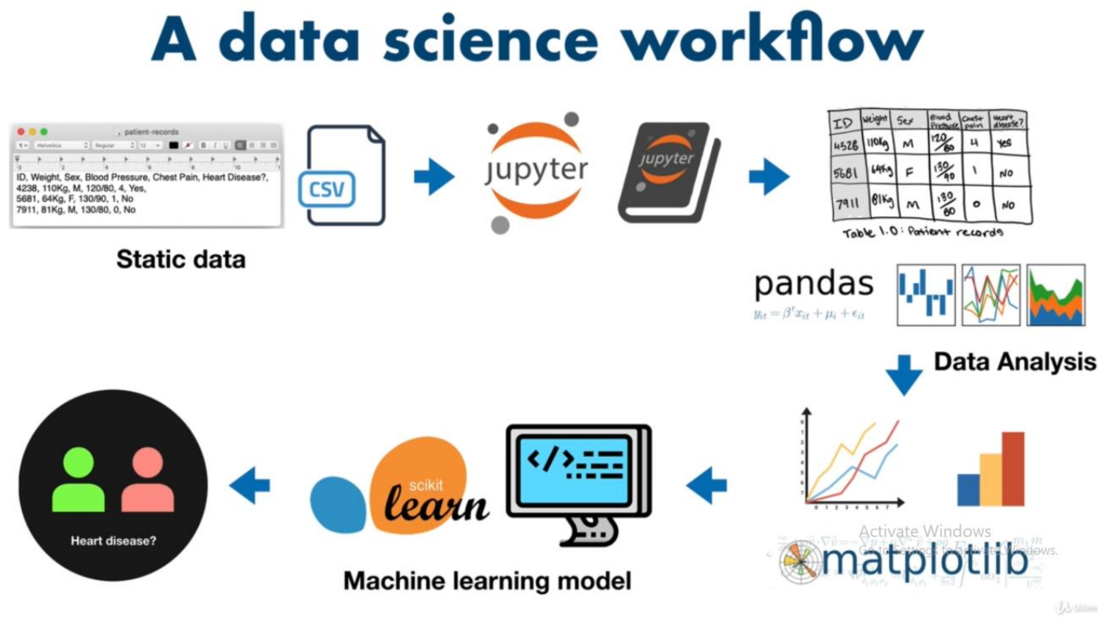
Streaming Data

Streaming data is data which is constantly changed over time.

For example say you wanted to predict how a stock price will change based on news headlines you'll be working with streaming data.



Streaming



```
ID, Weight, Sex, Blood Pressure, Chest Pain, Heart Disease?,
4238, 110Kg, M, 120/80, 4, Yes,
5681, 64Kg, F, 130/90, 1, No
7911, 81Kg, M, 130/80, 0, No
```

ID	Weight	Sex	Blood Pressure	Chest pain	Heart disease?
4328	110Kg	M	120/80	4	Yes
5681	64Kg	F	130/90	1	NO
7911	81Kg	M	130/80	0	NO

Table 1.0: Patient records

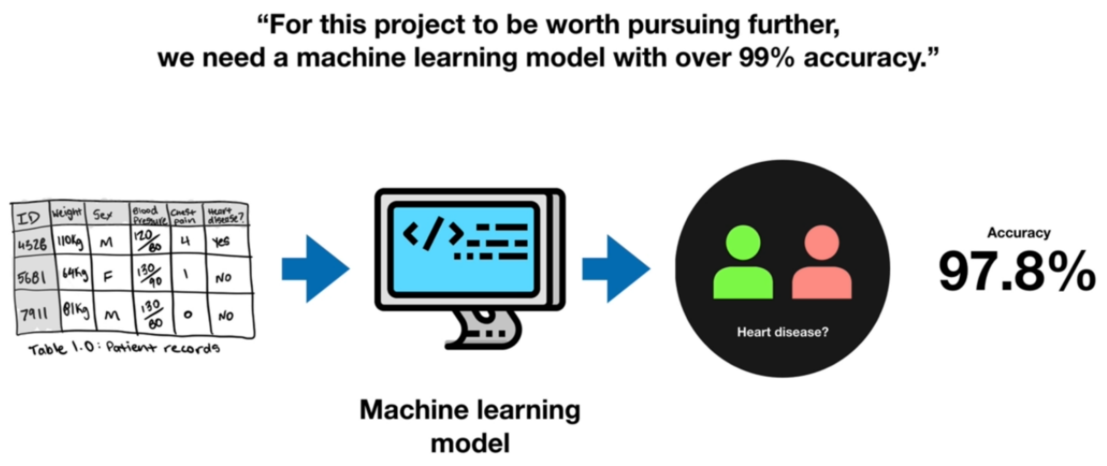
3) Evaluation

3. Evaluation ✓

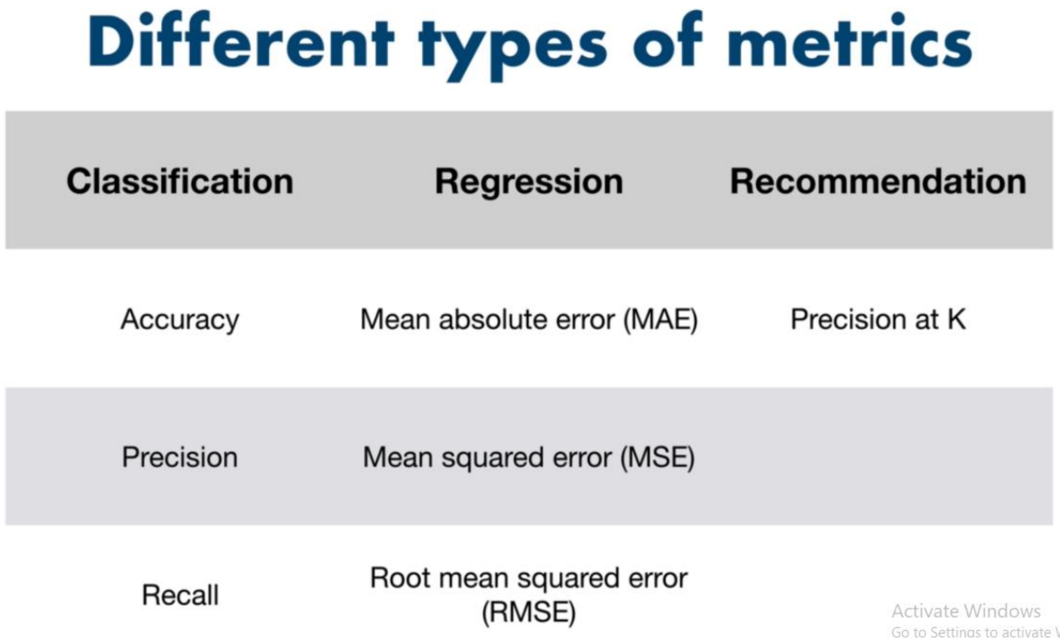
"What defines success for us?"

An evaluation metric is a measure of how well a machine learning algorithm predicts the future.

For example if your problem is to use patient medical records to classify whether someone has heart disease or not you might start by saying for this project to be valuable we need a machine learning model with over 99% accuracy because predicting whether or not a patient has heart disease is an important task. So you'll want a highly accurate model.


"For this project to be worth pursuing further, we need a machine learning model with over 99% accuracy."
Table 1.0: Patient records
Machine learning model
Heart disease?
Accuracy
97.8%

ID	Weight	Sex	Blood Pressure	Chest pain	Heart disease?
4328	110Kg	M	120/80	4	Yes
5681	64Kg	F	130/90	1	No
7911	81Kg	M	130/80	0	No



Different types of metrics

Classification	Regression	Recommendation
Accuracy	Mean absolute error (MAE)	Precision at K
Precision	Mean squared error (MSE)	
Recall	Root mean squared error (RMSE)	

4) Features

4. Features

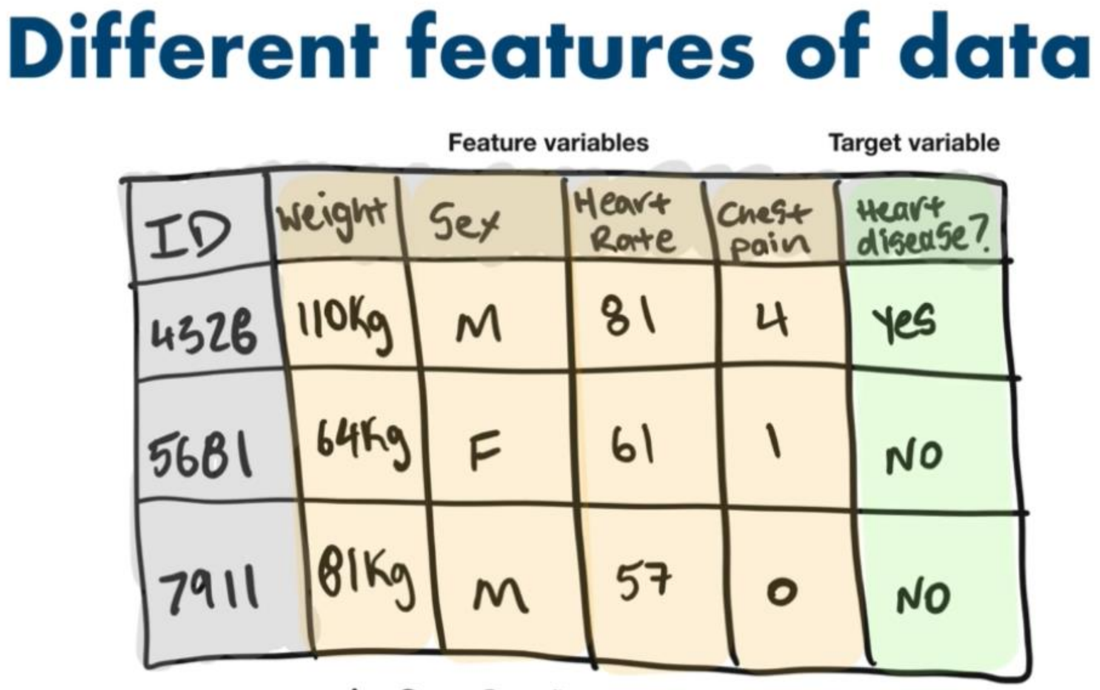
"What do we already know about the data?"

Features is another word for different forms of data. Features refers to the different forms of data within structured or unstructured data.

For example let's go back to our predicting heart disease problem we might want to see if things such as a person's body weight, their sex, their average resting heart rate and their chest pain rating can be used to predict if they have heart disease or not.

These three things a patient's body weight, sex, average resting heart rate and chest pain are features of the data that could also be referred to as feature variables.

In other words we want to use the feature variables to predict the target variables which is whether or not a person has heart disease or no.



Different features of data

	Feature variables				Target variable
ID	Weight	Sex	Heart Rate	Chest pain	Heart disease?
4326	110Kg	M	81	4	Yes
5681	64Kg	F	61	1	NO
7911	81Kg	M	57	0	NO

Table 1.0: Patient records

Now when it comes to feature variables again there are different kinds.

Numerical features

numerical which means a number like body weight.

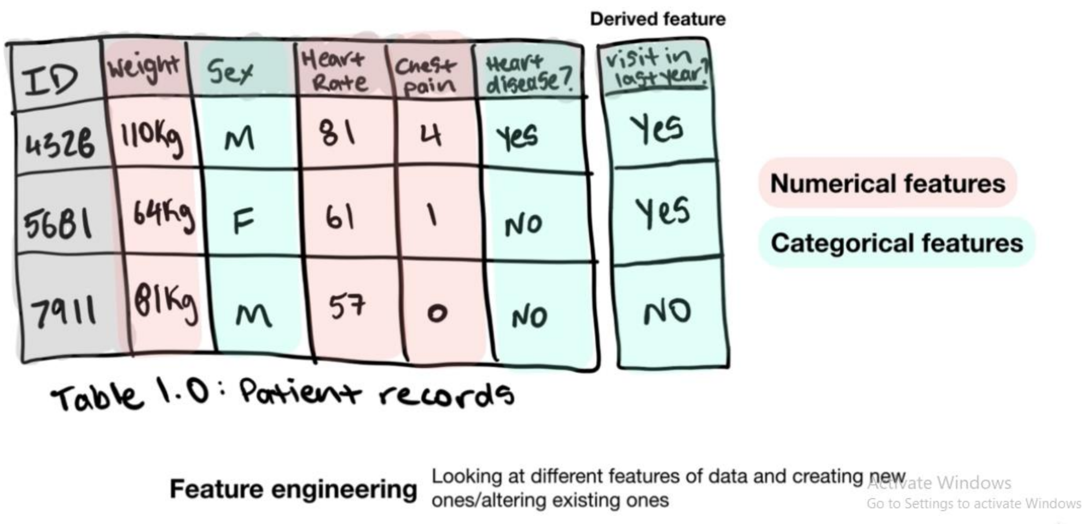
Categorical features

There's categorical which means one thing or another like sex or whether a patient is a smoker or not.

Derived features

derived which is when someone like yourself looks at the data and creates a new feature using the existing ones.

For example you might look at someone's hospital visit history timestamps and if they've had a visit in the last year you could make a categorical feature called visited in last year. If someone had visited in the last year they would get true. If not they would get false.



Derived feature

ID	Weight	Sex	Heart Rate	Chest pain	Heart disease?	visit in last year?
4328	110Kg	M	81	4	Yes	Yes
5681	64Kg	F	61	1	NO	Yes
7911	81Kg	M	57	0	NO	NO

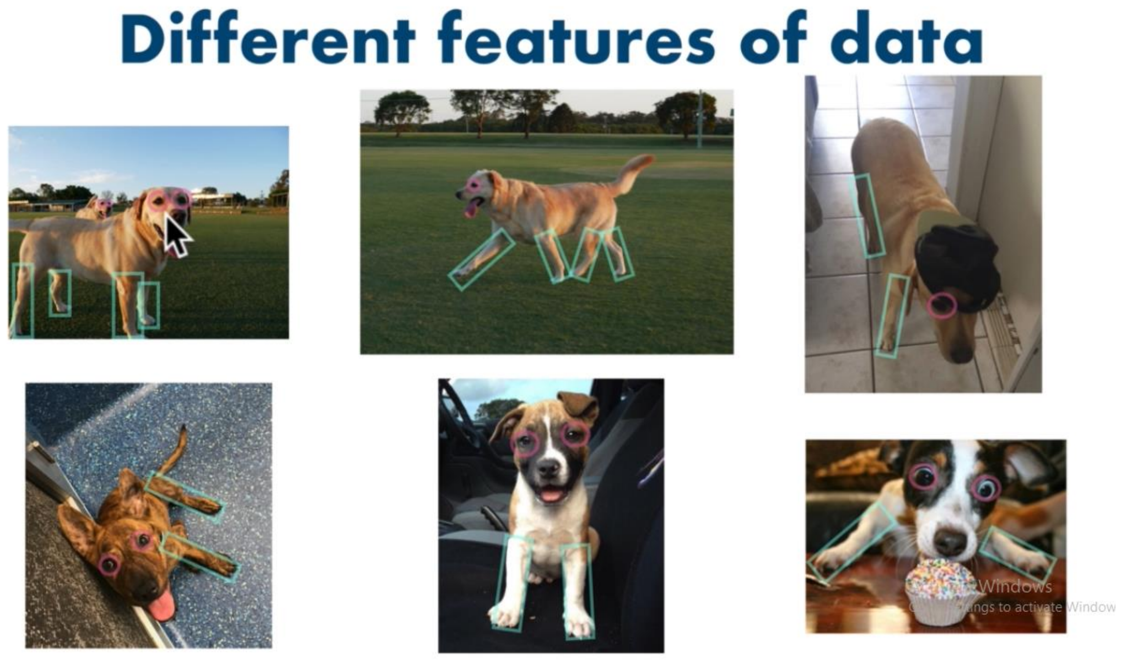
Table 1.0: Patient records

Numerical features
Categorical features

Feature engineering Looking at different features of data and creating new ones/altering existing ones

Heart disease example is structured but unstructured data has features too.

They're just a little less obvious if you looked at enough images of dogs you'd start to figure out Most of these creatures have four shapes coming out of their body, their legs and a couple of circles up the front their eyes as a machine learning algorithm looks at different images. It would start to learn these different shapes and much more and figure out how different pictures are similar or different to each other.


Different features of data

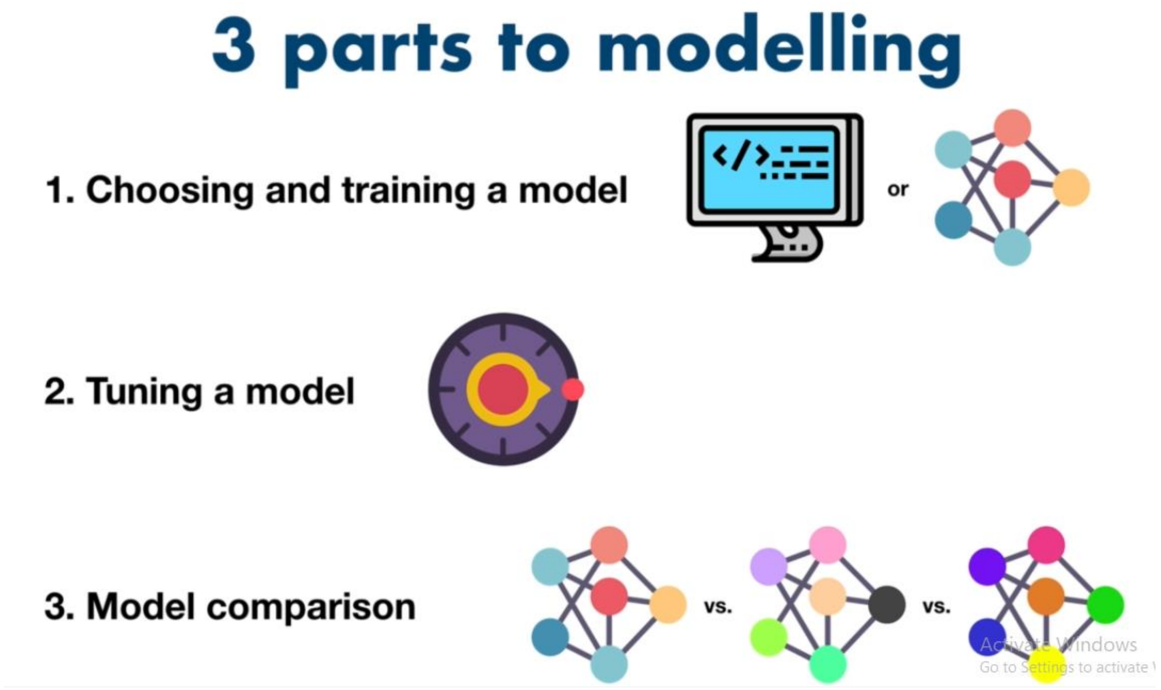
- Feature works best within a machine learning algorithm, if many of the samples have it.
- Machine learning algorithm learns best when all samples have similar information.

5) Modelling

5. Modelling Part 1 — 3 sets

"Based on our problem and data, what model should we use?"

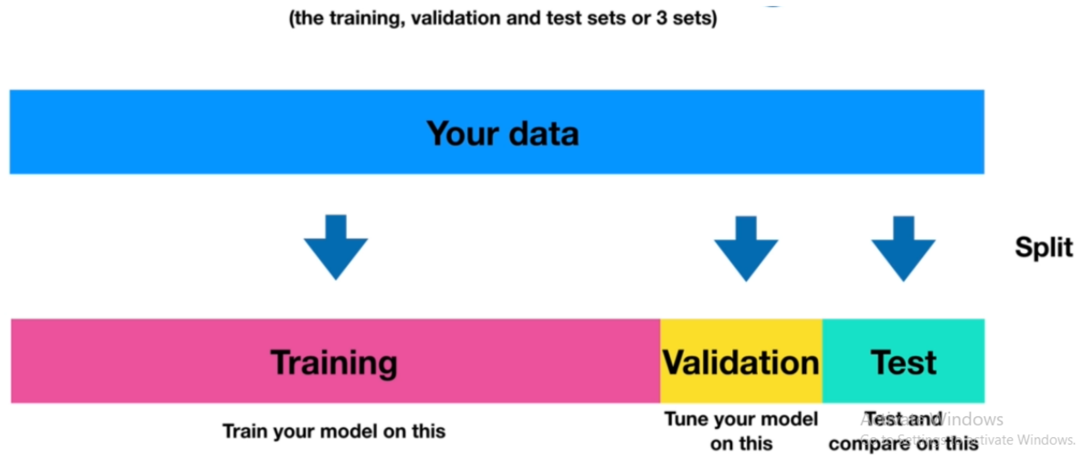
Modelling can be broken down into three parts choosing and training a model, churning a model and model comparison



Part one of modelling is most important concept in machine learning. The **train**, **validation** and **test splits** or commonly referred to as three sets.

Now since you want to be using machine learning models to gain insights on some data to predict the future it's important to test how well they would go and do in the real world.

To do this you split your data into three different sets a **training set** to train your model on, a **validation set** to choosing your model on & a **test set** to test and compare your different models.

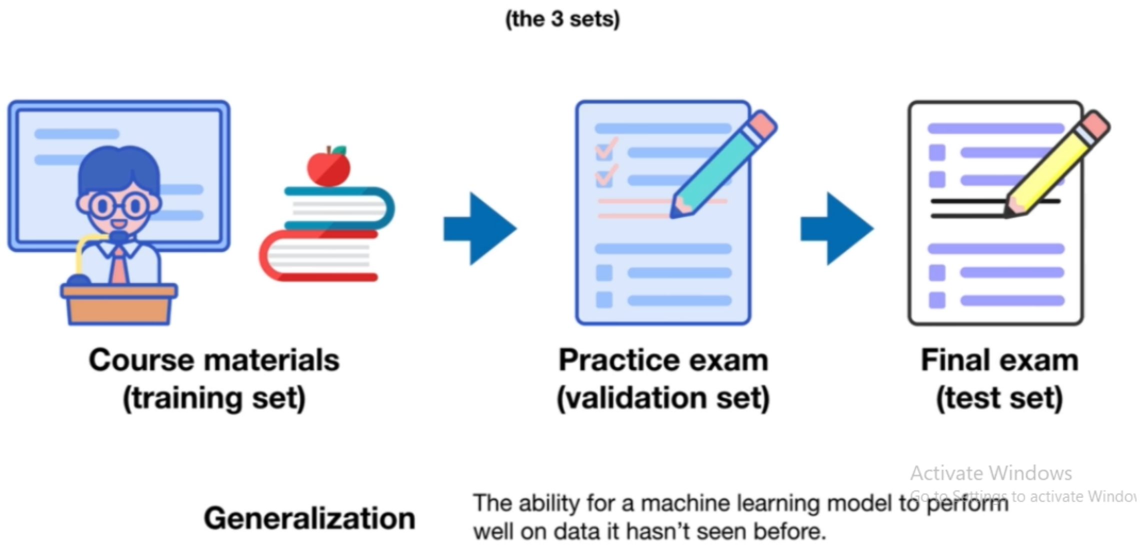


why is this important.

Think of it like this when you're at university you might study the Course materials all through the semester then before the final exam You might see how you could improve your knowledge on a practice exam.

After doing well on the practice exam you're confident you'll do well on the final exam when you take the final exam. And although some of the problems you've never seen before you're able to adapt the knowledge you've learned from the study materials to the slightly different but similar questions on the final exam. Because of this you pass the final exam with great marks.

This adaptation that you had from the course materials and practice exams to the final exam is referred to in machine learning as a generalisation or the ability for a machine learning model to perform well on data it hasn't seen before because of what it's learned on another dataset.

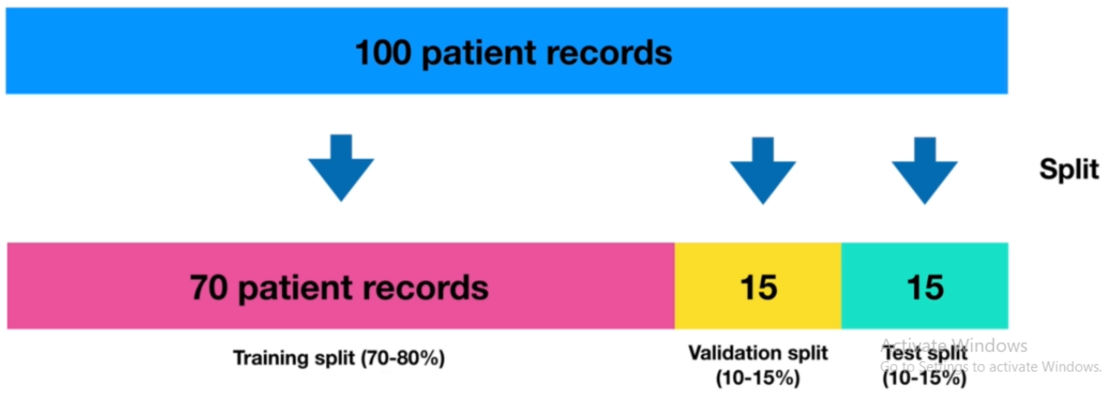


The ability for a machine learning model to perform well on data it hasn't seen before.

For your machine learning models to be valuable at predicting something in the future on unseen data you'll want to avoid them becoming memorization machines. This is where training validation and test splits come in.

In our heart disease example let's say there were 100 patients. One way to create these splits is to shuffle these patients. Then select 70 percent for training which would mean that would be about 70 patient records. And 15 percent for validation and 15 percent for testing which means to be 70 patients in the training set. 15 patients in the validation split and 15 patients in the test split.

(the 3 sets)

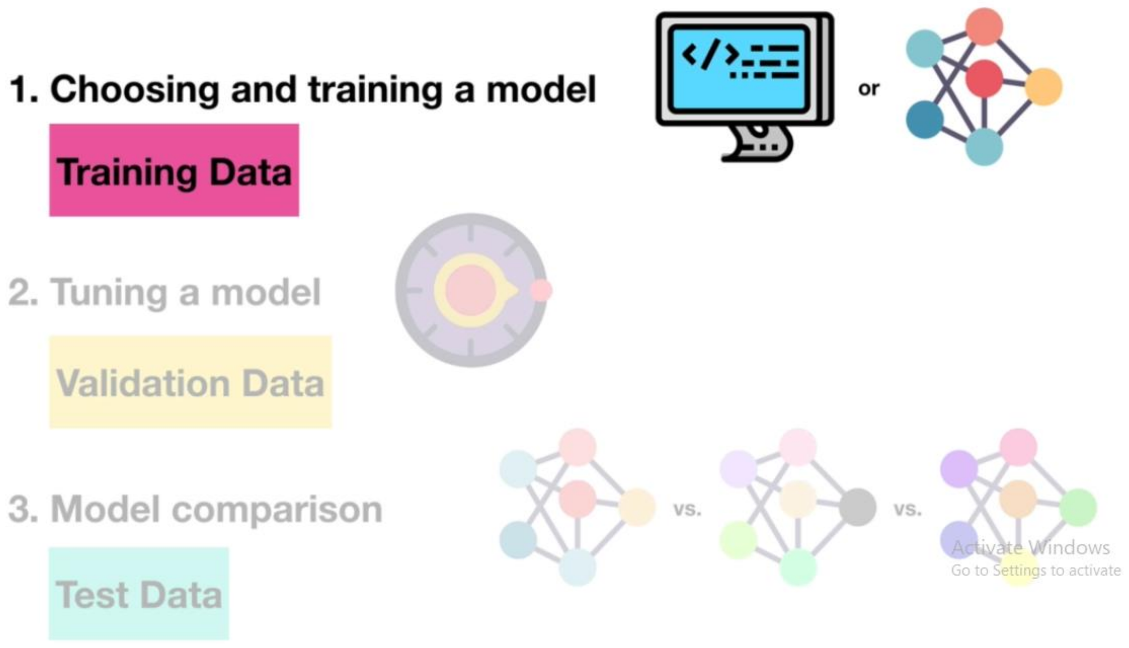


Important to remember is that all three of these sets a separate during training the model never sees the validation split or the test split or vice verse.

- ***Choosing***

5. Modelling Part 2 — Choosing
"Based on our problem and data, what model should we use?"

You choose a model and train it on your training data.

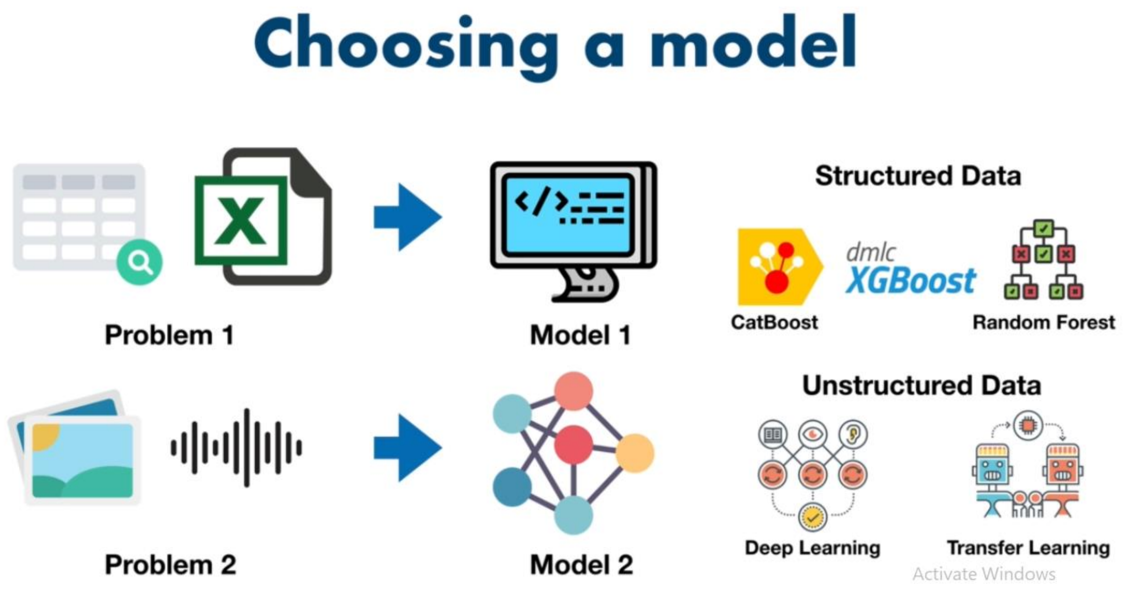


Unlike creating your own algorithms from scratch there are many pre-built machine learning models which you can take advantage of when you first begin.

Your main goal will be knowing what kind of machine learning algorithm to use with what kind of problem, this is because some algorithms work better than others on different types of data.

For now a tidbit to remember is if you're working with structured data decision trees such as **random forest** and **gradient boosting** algorithms like **cat boost in x g boost** tend to work best.

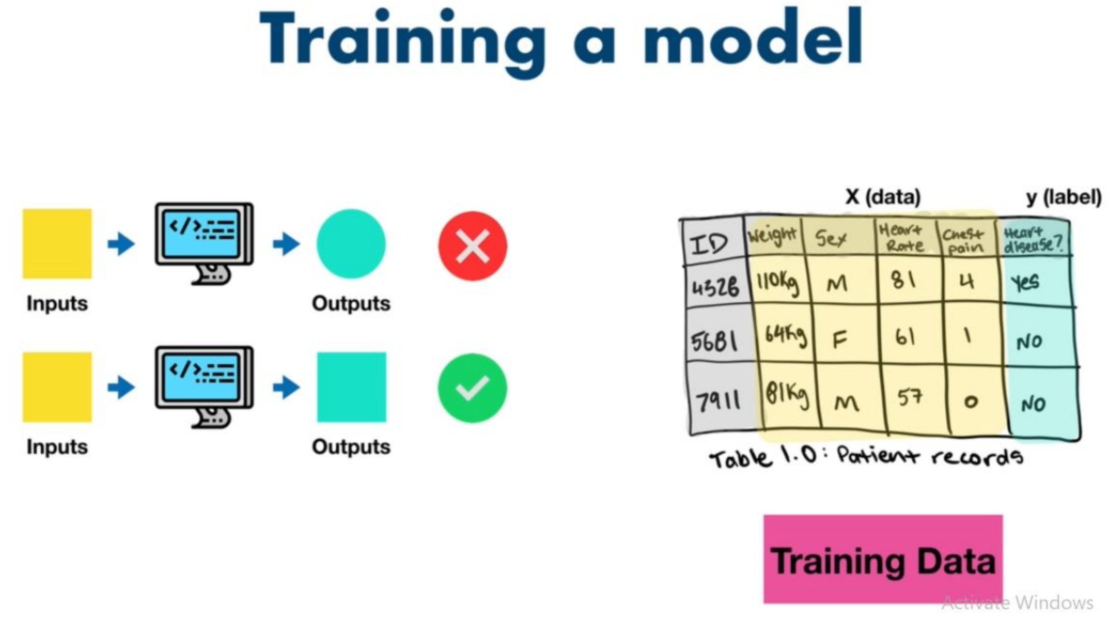
And if you're working with unstructured **data deep learning neural networks** and **transfer learning** tend to work best.



Once you've chosen a model your next step is to train. The main goal here will be to line up the inputs and outputs.

For example in our heart disease problem we want our model to look at the feature variables the inputs and then find the patterns and use them to predict the target variable.

Remember training a model takes place on the training data split. This is where your model learns the course material. We don't want to let our models cheat and see the final exam before they do their study.



ID	Weight	Sex	Heart Rate	Chest pain	Heart disease?
4326	110Kg	M	81	4	Yes
5681	64Kg	F	61	1	No
7911	81Kg	M	57	0	No

Table 1.0: Patient records

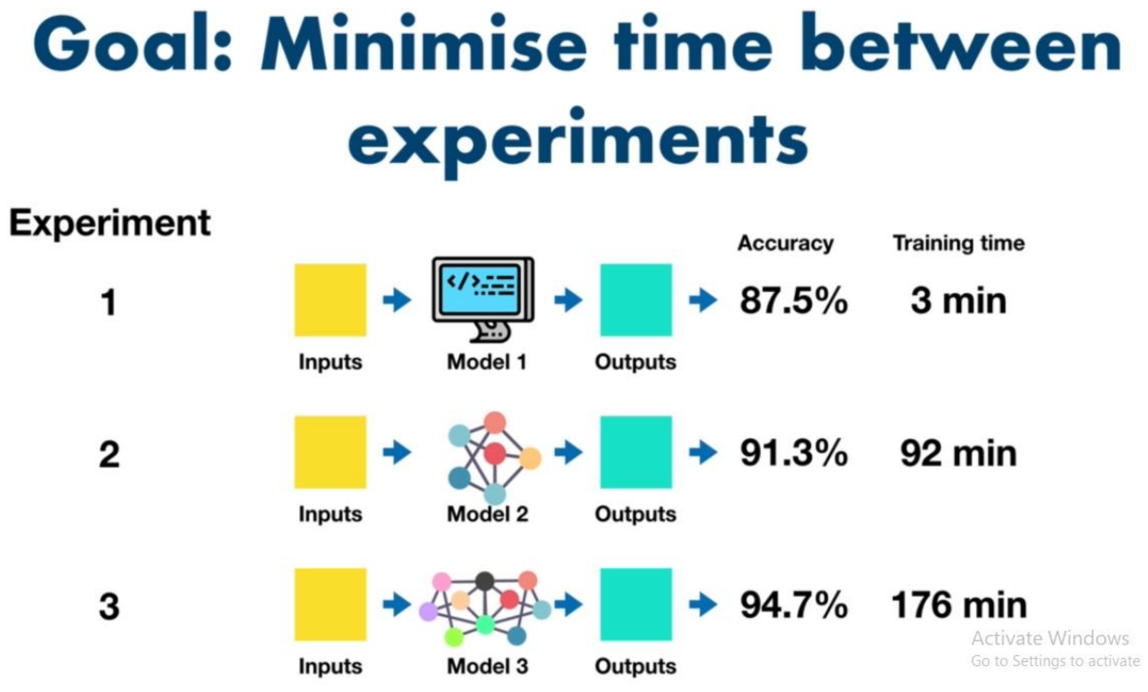
Depending on how much data you have and how complex your model is training may take a while.

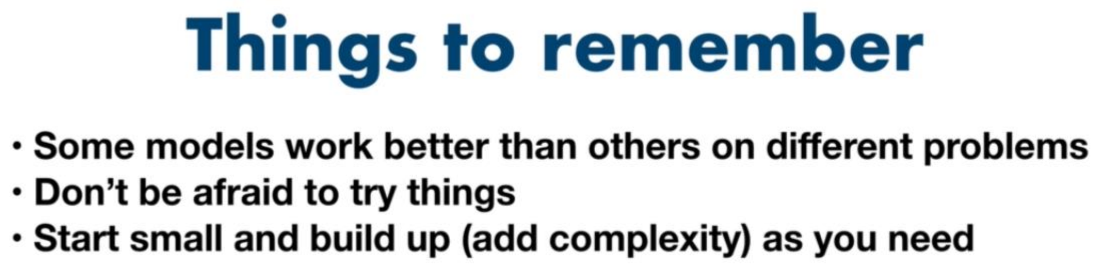
One of your biggest goals when training a model is to minimize the times between experiments. So, sometimes this will mean to use a small portion of your data first.

For example, if your training dataset had 100000 examples you might start training a model with only the first 10000 and see how it goes.

You might also decide to use a less complicated model to begin with deep model such as no networks generally take longer to train than other kinds of models.

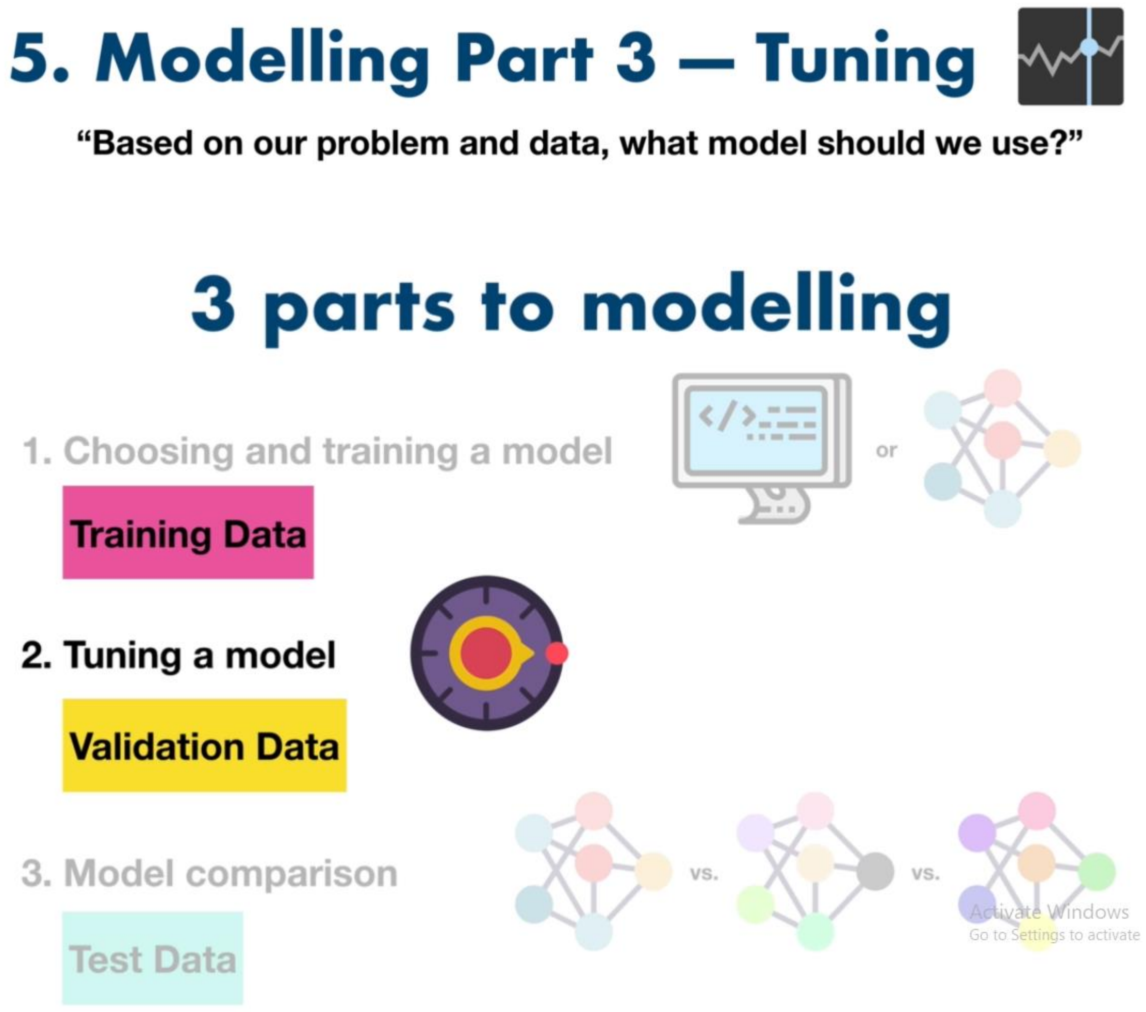
We want to minimize experimentation time that we can go from step 1 to step 2 to step 3.




Things to remember

- Some models work better than others on different problems
- Don’t be afraid to try things
- Start small and build up (add complexity) as you need

- ***Tuning***



5. Modelling Part 3 — Tuning

"Based on our problem and data, what model should we use?"

3 parts to modelling

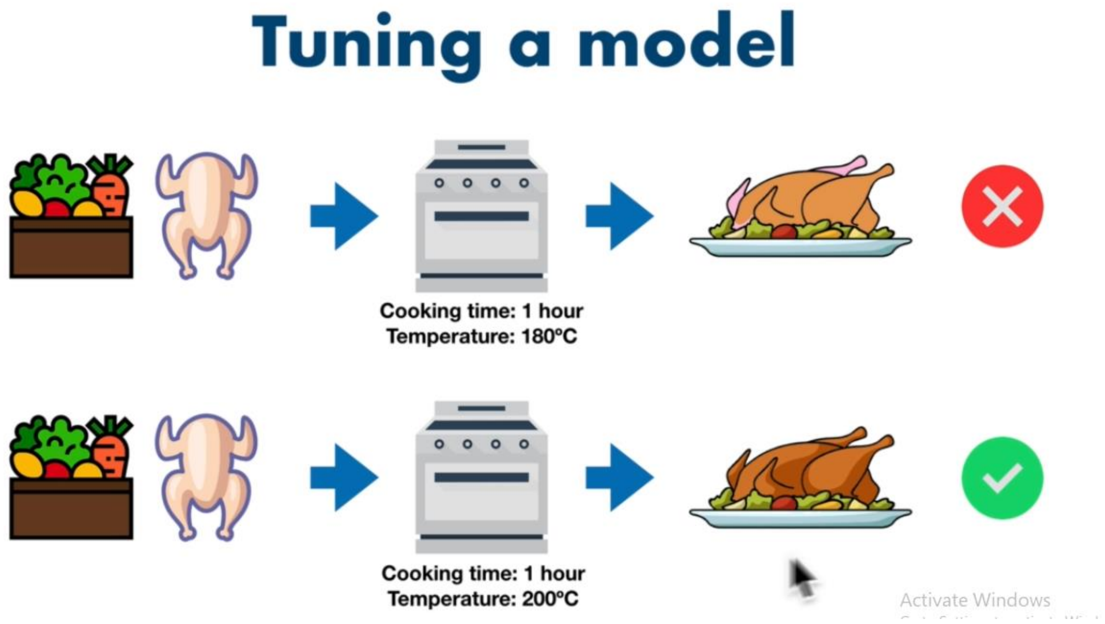
1. Choosing and training a model
   - Training Data
2. Tuning a model
   - Validation Data
3. Model comparison
   - Test Data

The next step is to try and improve it like what we said in last lesson.

How a car can be tuned for different styles of driving a model can be tuned for different types of data. Specifically your data usually this training will take place on a validation data split.

However if you don't have access to a validation set due to how you've done your training validation and test data split it can also happen on the training data on many models have different hyper parameters which can be adjusted.

If you make 180` grains chicken doesn't turn out well you'd take it up 200 degrees. Chicken does turn out well.



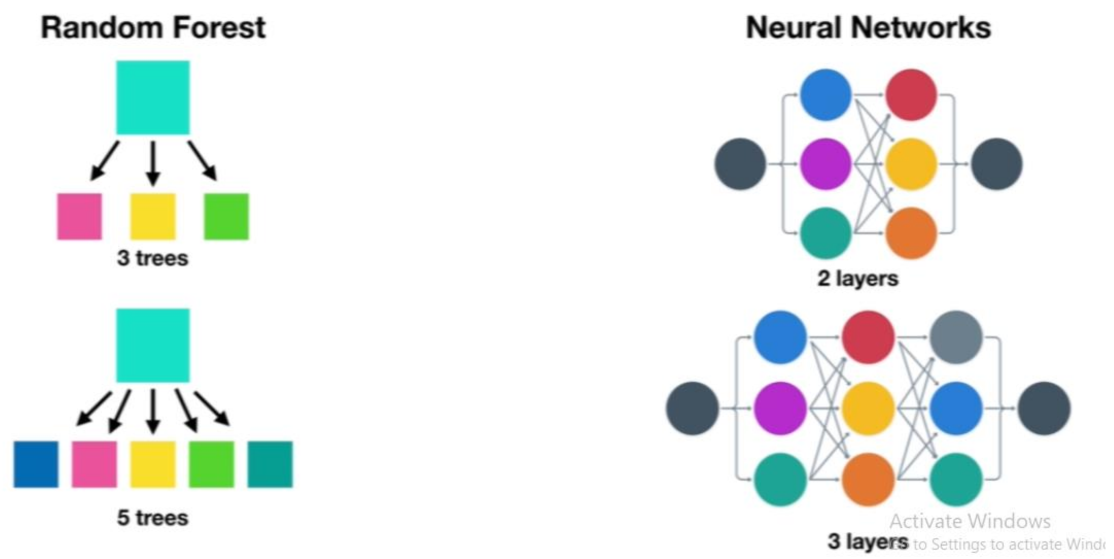
Depending on what kind of model you're using will depend on what kind of hyper properties you can chew.

For example, a random forest will allow you to adjust the number of trees in this one we've got three trees. So, we've got five trees and we'll have a look at the random forest.This is a type of machine learning algorithm.

A no network will allow you to adjust the number of layers.

You can adjust on different kinds of algorithms.

Tuning a model



The main things for you to remember are machine learning models have hyper parameters you can adjust. However, depending on what model you're going to use, the hyper parameters will be different.

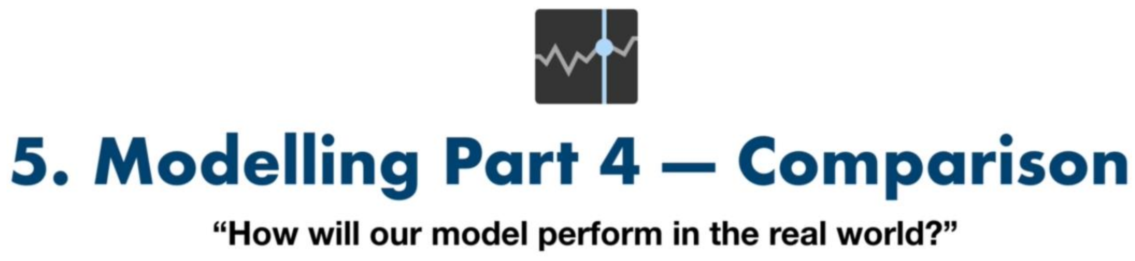
The goal of tuning hybrid parameters is to improve your model's performance

You should do model tuning on your training and or validation sets meaning you train a model using an initial set of hyper-parameters.

Things to remember

- **Machine learning models have hyperparameters you can adjust**
- **A models first results aren't its last**
- **Tuning can take place on training or validation data sets**

- ***Modelling***



5. Modelling Part 4 — Comparison

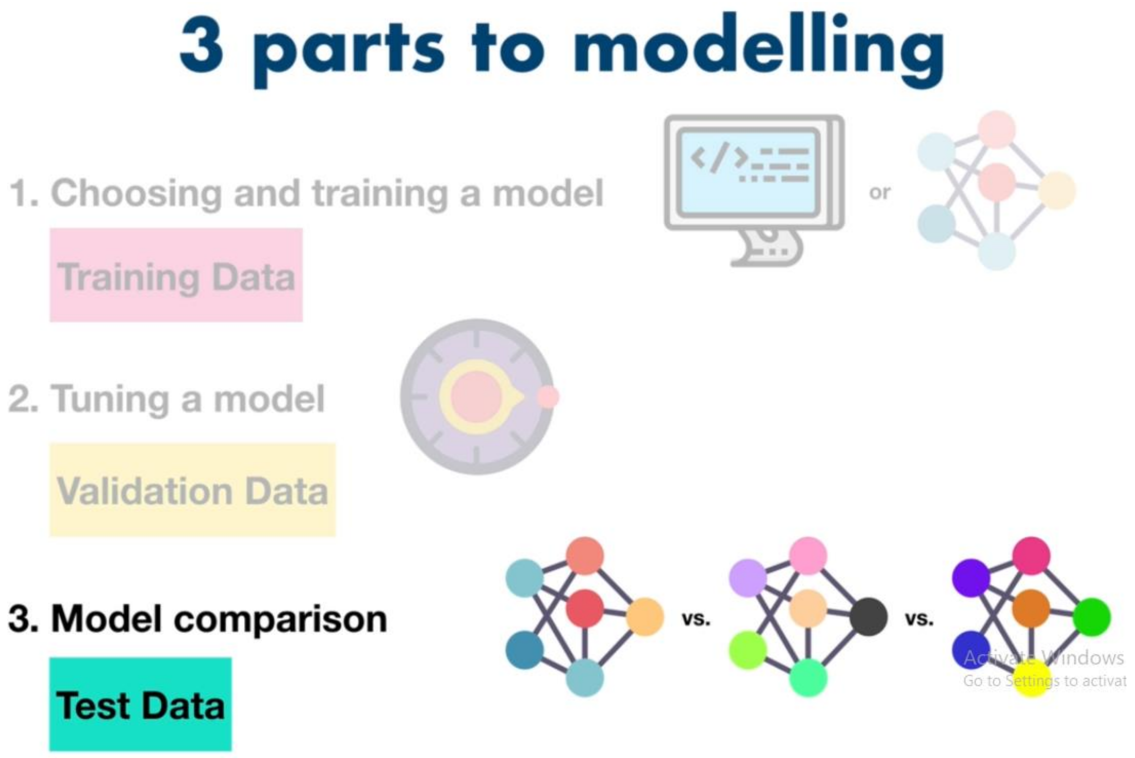
"How will our model perform in the real world?"

So, after you've tuned and proved your models performance through hyper parameter tuning it's time to see how it performs on the tests set, tests set is like the final exam for machine learning models.

If you've created your data splits correctly it should give you an indication on how your model will perform once deployed in production.

Since your model has never seen data in the test set evaluating your model on it is a good way to see how it generalizes.

Generalizing mean adapts to data it hasn't seen before such as how heart disease prediction machine learning model would perform at classifying whether a patient has heart disease or not, on a patient who wasn't in our original data set.



A good model will yield similar results on the **training**, **validation** and **test sets**.

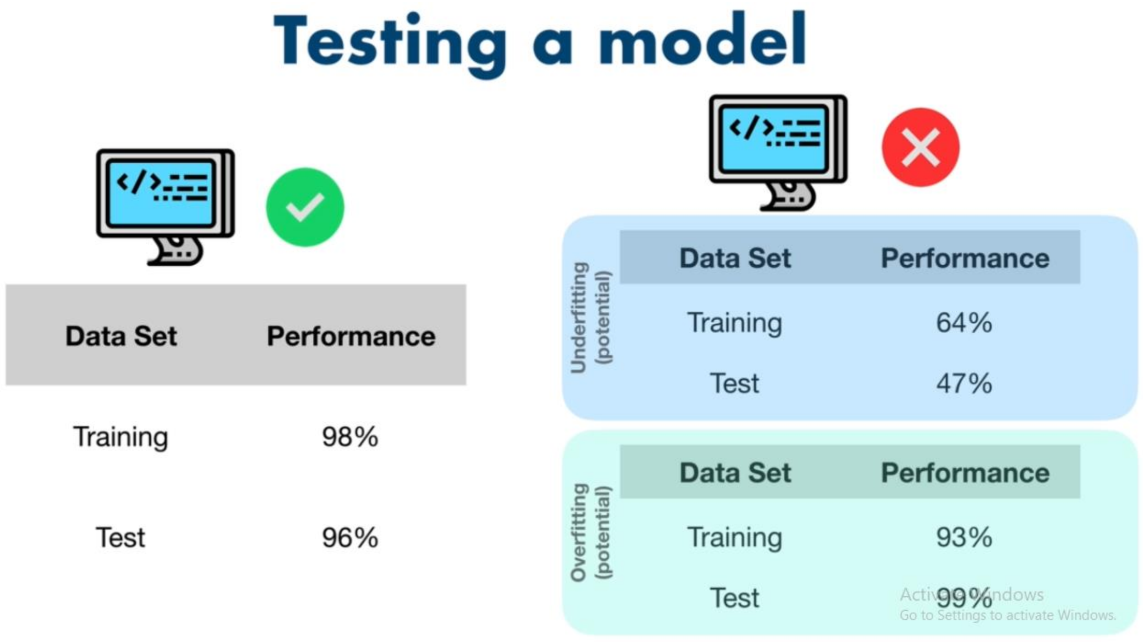
And it's not uncommon to see a slight decline in performance from the model on the training and validation set to the test set.

For example, your model might achieve 98 percent accuracy on the training dataset and 96 percent accuracy on the test set.

What you should be worried about is if the training set performance is dramatically higher than the test set also known as ***under fitting.***

And if the test set performance is higher than the training set performance also known as ***overfishing***

Overfishing and Underfunding are both examples of a model not being able to generalize well which is what we don't want.


Testing a model

Data Set	Performance
Training	98%
Test	96%

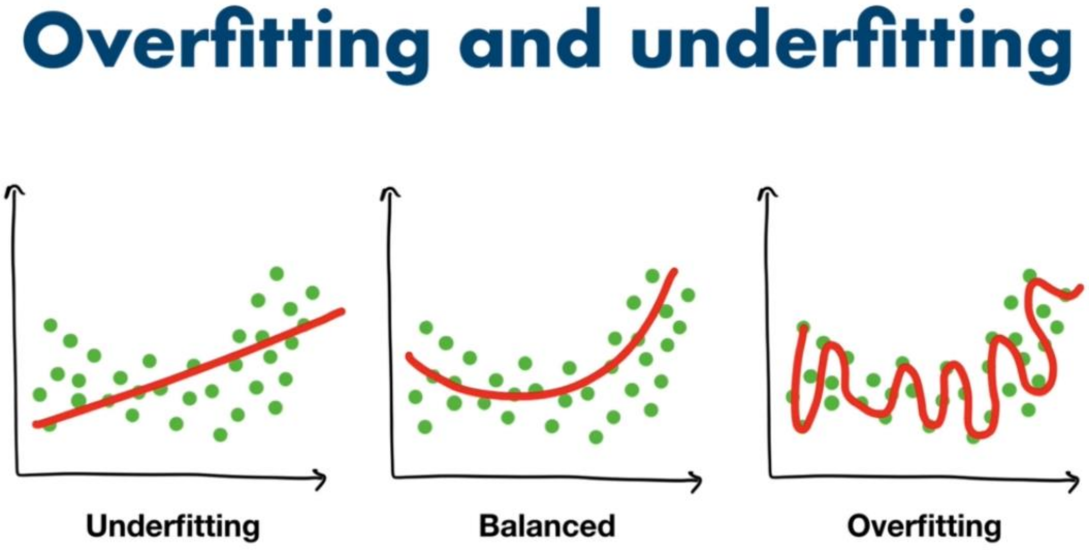
Underfitting (potential)

Data Set	Performance
Training	64%
Test	47%

Overfitting (potential)

Data Set	Performance
Training	93%
Test	99%

The ideal model shows up in the goldilocks zone it fits just right not too well but not too poorly.



There are several reasons why Underfitting & Overfitting can happen but the main ones are **data leakage** and **data mismatch**.

Data leakage

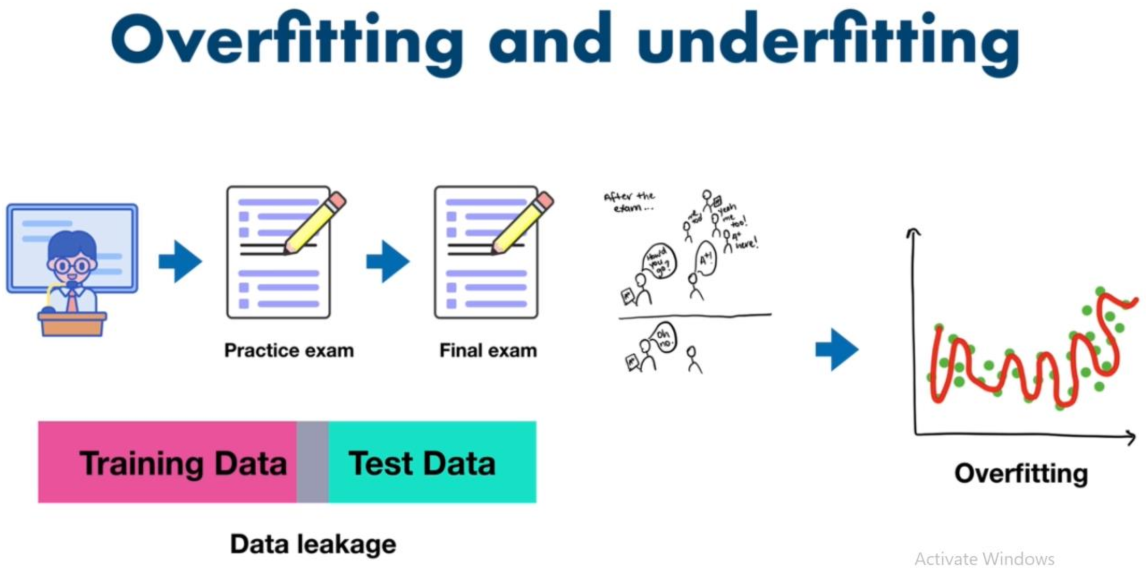
Data leakage happens when some of your test data leaks into your training data.

This often results in **overfishing** or a model doing better on the test set then on the training dataset

It's like to have a look at the final exam or everyone had to look at the final exam as the practice exam. So, you're machine learning model has just learned what it's about to be test on.

So when it comes time to modelling it just learned it way too well and it starts to fit the data. And this is why it's important to do your splits correctly and ensure that machine learning model training happens only on the training data set. Validation and model tuning happens only on the validation or training data set and that testing and model comparison happens on the test data set.

For machine learning the test dataset is used as an indication of how well your model will generalize in the real world. So, you want to avoid data leakage.

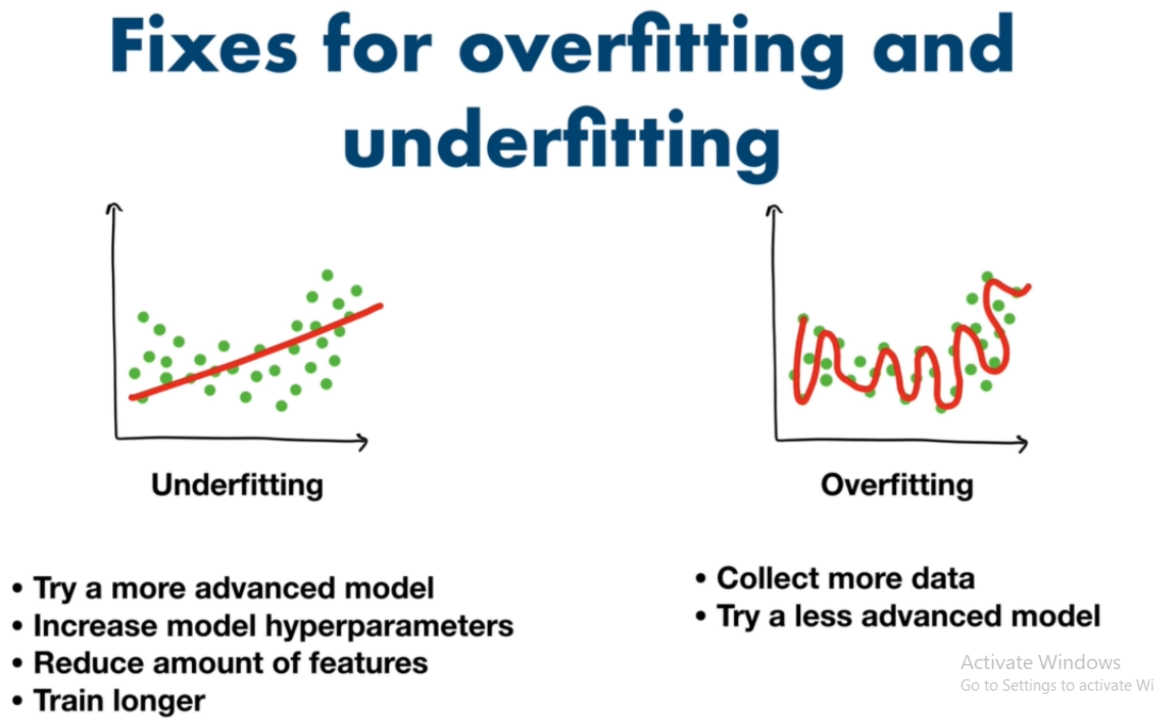


data mismatch

Data mismatch happens when the data you're testing on is different to the data you're training on. Such as having different features in the training data to the test data.

Having this kind of mismatch can lead to models performing poorly on test data compared to training data.

This is why it's important to ensure that training is done on the same kind of data as you'll be testing on. And as close as possible to what you'll be using in your future applications



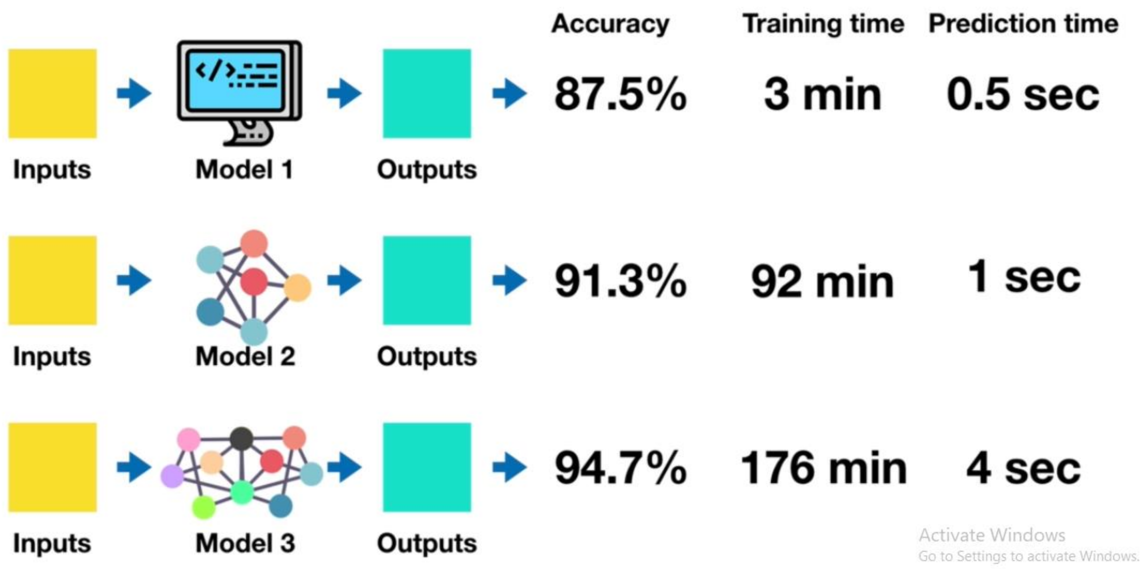
Fixes for overfitting and underfitting

- Try a more advanced model
- Increase model hyperparameters
- Reduce amount of features
- Train longer

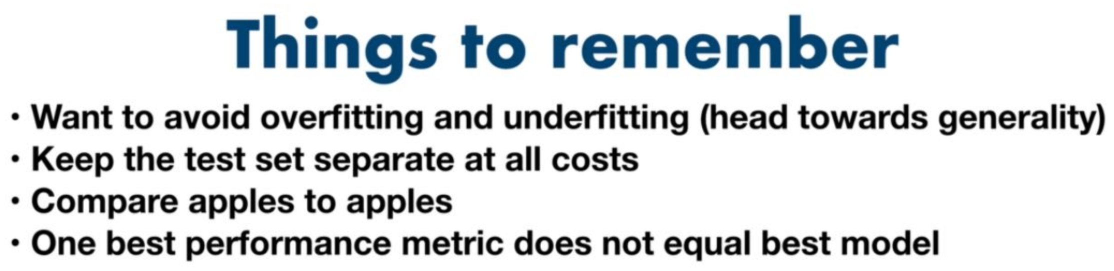
- Collect more data
- Try a less advanced model

Finally when comparing two different models to each other. It's important to ensure you're comparing apples with apples and oranges with oranges.

For example, Model 2 trained on dataset 1 vs. Model 3 trained on data set 1.



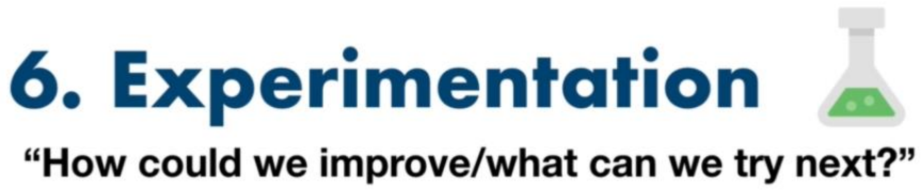
			Accuracy	Training time	Prediction time
Inputs	Model 1	Outputs	87.5%	3 min	0.5 sec
Inputs	Model 2	Outputs	91.3%	92 min	1 sec
Inputs	Model 3	Outputs	94.7%	176 min	4 sec



Things to remember

- Want to avoid overfitting and underfitting (head towards generality)
- Keep the test set separate at all costs
- Compare apples to apples
- One best performance metric does not equal best model

6) Experimentation



6. Experimentation

"How could we improve/what can we try next?"

You might try a different model.

We got model to see how it goes and it performs pretty well.

If that didn't work when you reported it is like oh maybe you could do a little bit better.

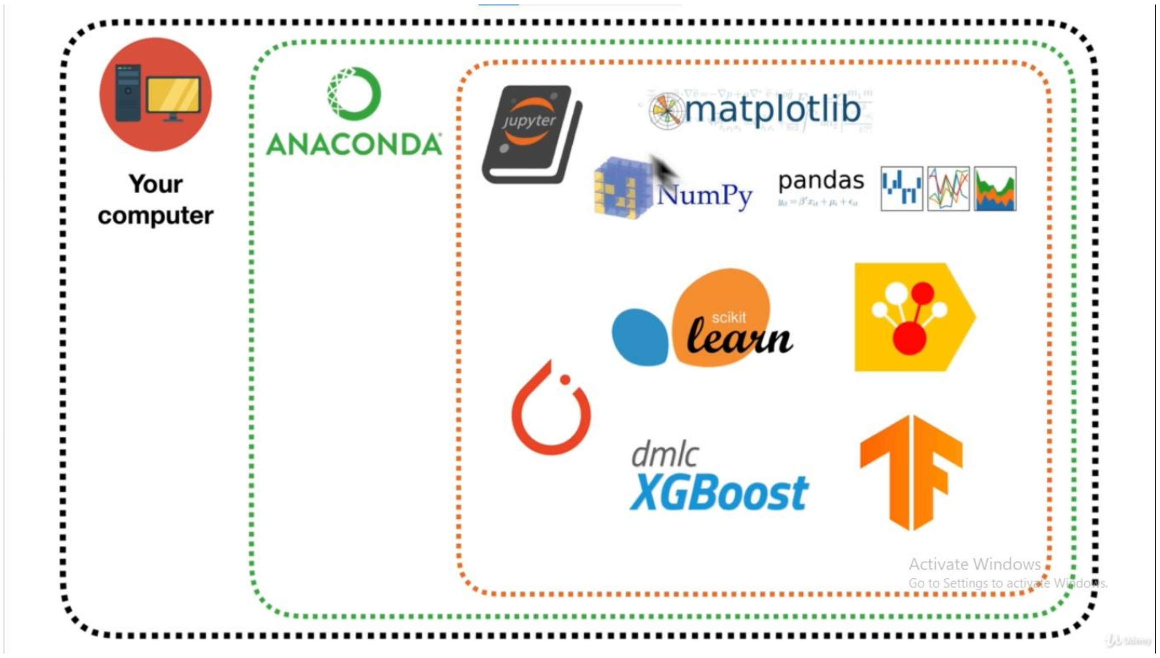
You might try vary the inputs slightly and change your desired outputs and try a new model again.

-------------------------------------Creating a framework Ends --

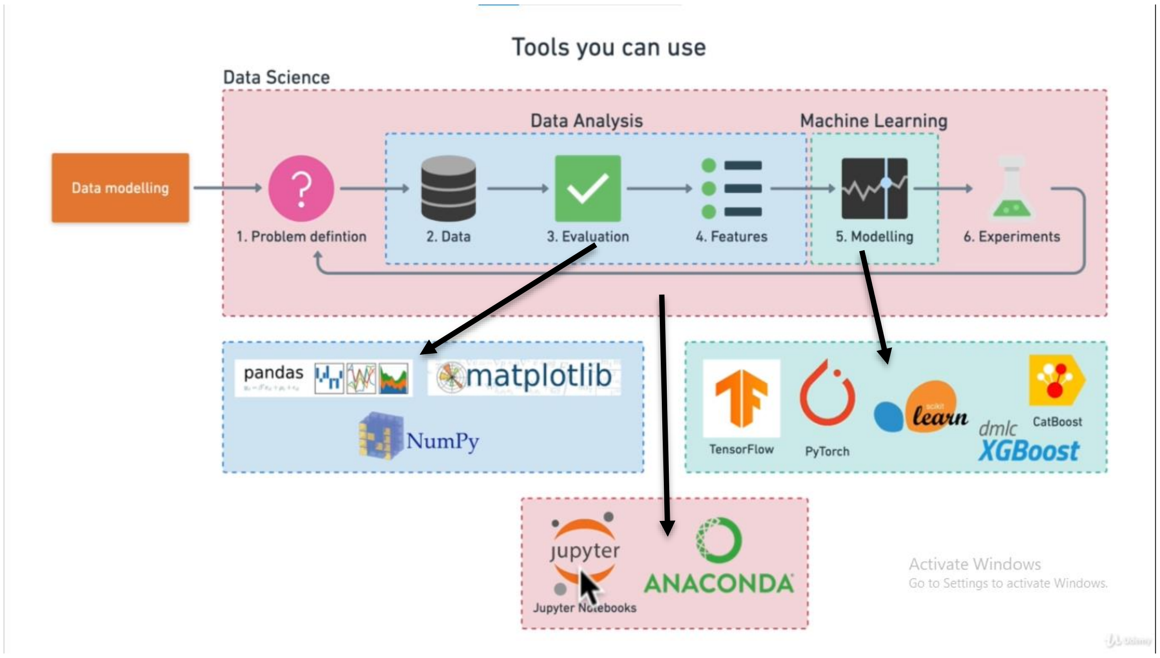
Tools We Will Use

Anaconda which is like the hardware store of data science and machine learning tools.

We're going to install **Anaconda** and then once you've got Anaconda getting all of these other machine learning tools like **Jupiter notebooks, pi torch, numpy, matplotlib, pandas, cat boost ex G boost, and tensor flow**. Once you've got Anaconda getting these tools that we need to apply to that process we've just created that framework we've created will be an absolute breeze.

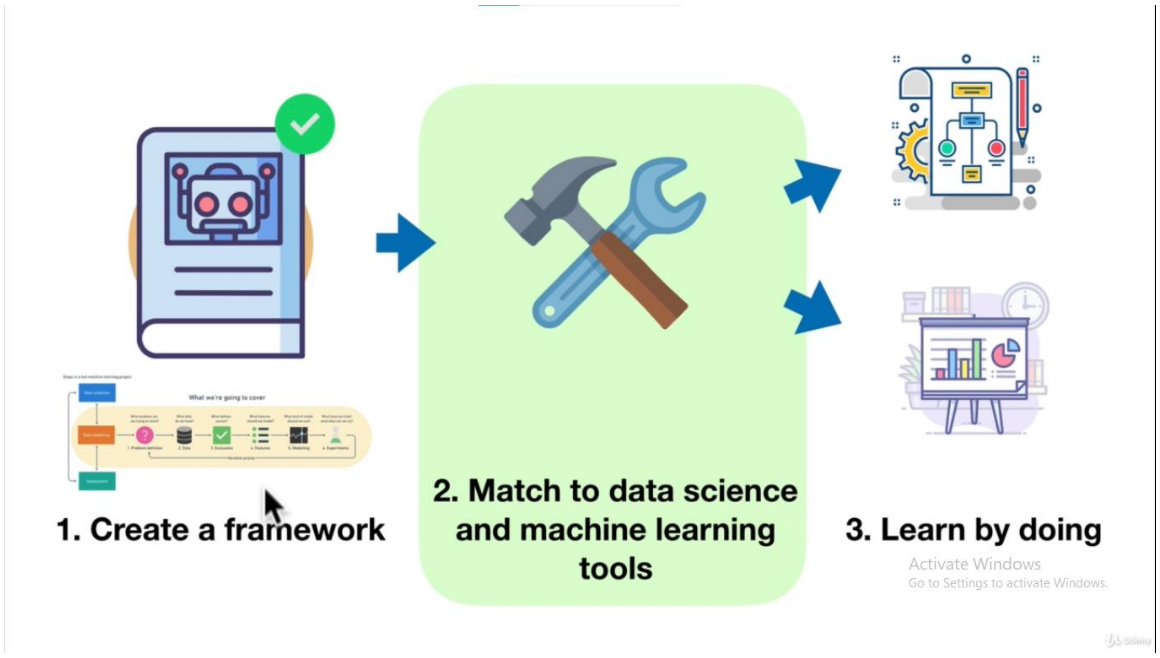


So what will this look like if we wanted to attach it to our process.



Color code for each of are framework process and their respective tools.

Introducing Our Tools


1. Create a framework
2. Match to data science and machine learning tools
3. Learn by doing
Activate Windows
Go to Settings to activate Windows.

There are many different tools that we could use.

So we need something to help manage our tools for us.

To do so, you can use **Anaconda, mini conda** and **conda.**

You can think of **Anaconda** and **mini-Conda** as the hardware store and workbench of data scientists and **Conda** as your personal assistant.

Anaconda and **mini conda** are software distributions.

So much like when you download an app from the App Store. It comes with code that other people have written which you can use on your computer or smartphone.

The same goes for these two. They come with code that other people have written which we can take advantage of on our computers for our own machine learning projects.

collections of code written by other people are often referred to as packages or tools.

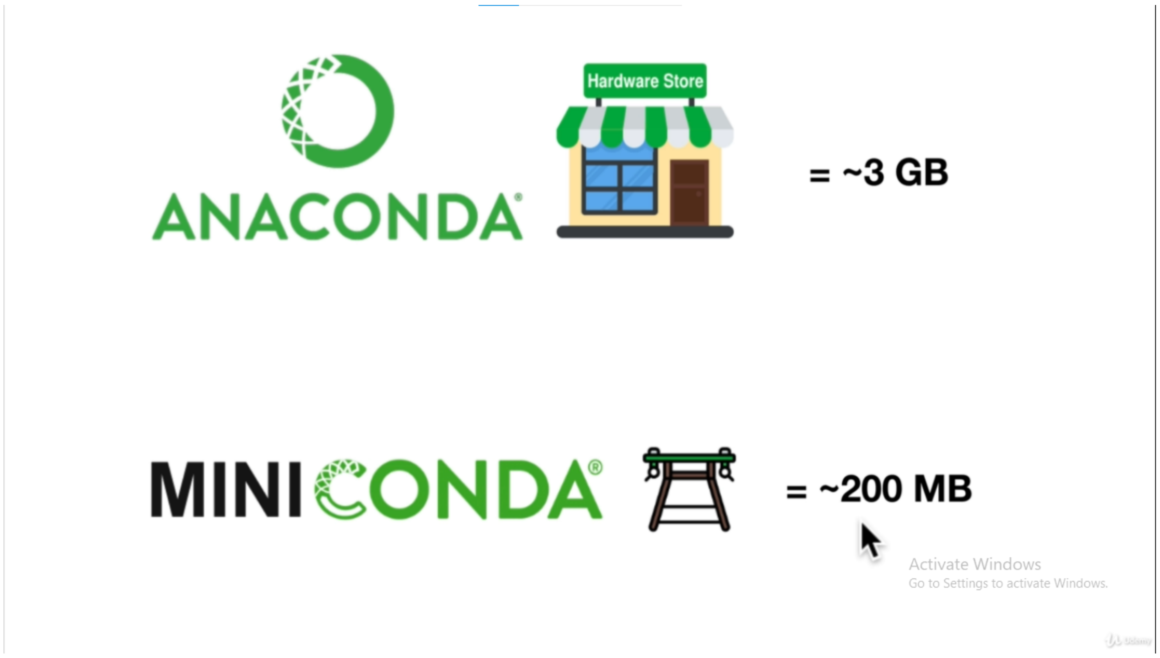
Now downloading anaconda into your computer requires around 3GB or so of space and it installs all of the major data science and machine learning packages.

However, remember, Anaconda is like the hardware store of data science tools. This means it's got almost every data science tool you can imagine. And many of them don't get used.

This is where mini conda comes in.

Mini conda to remember is that the data scientists workbench it starts with the minimum requirements to get started such as Python and a few others.

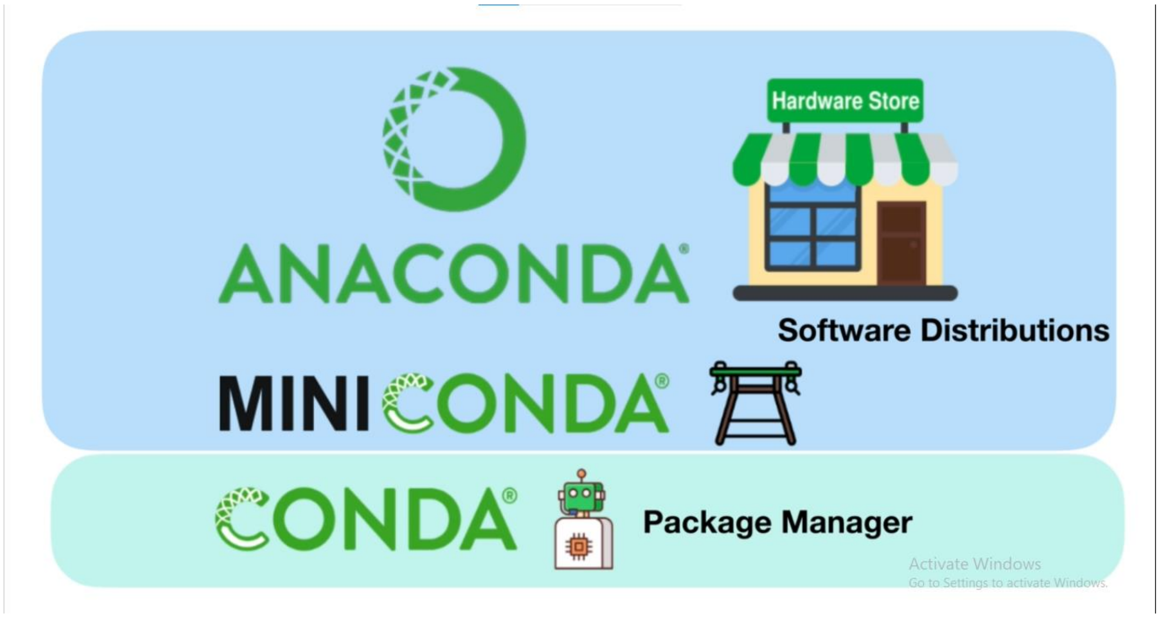
And now because of this it takes up up to about 10 times this space on your computer. Coming in and around 200MB so to save space and to get started we're going to use mini Conda.



Conda

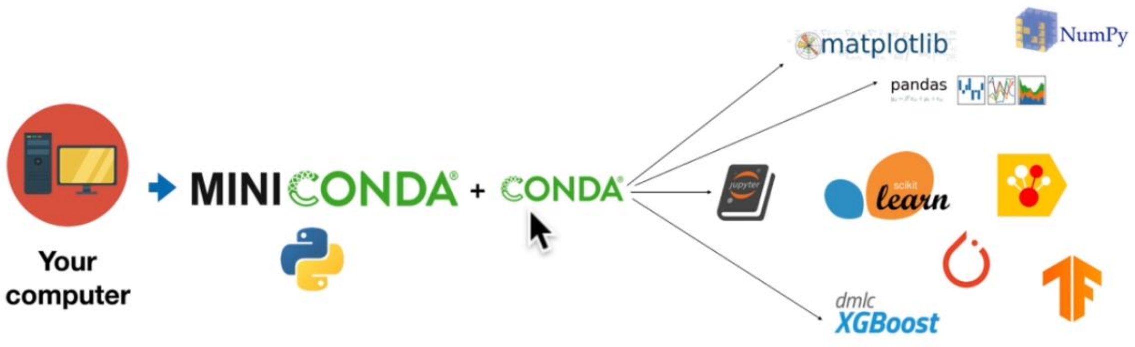
If Anaconda and mini Conda are like the hardware store and workbench and data scientists **conda** is like a data scientists personal assistant.

It can help you set up your workbench or the new tools when you need making a copy of what tools you're using to share with someone else and more formally Anaconda and mini conda are known as software distributions and **Conda** is known as a package manager.



Remember how a package is a collection of code someone else's written which you can use. Well conda helps you download install and manage these packages when you download Anaconda or mini Conda and install it to your computer. Conda allows us to do is install other tools

We use Conda the package manager to install these other packages

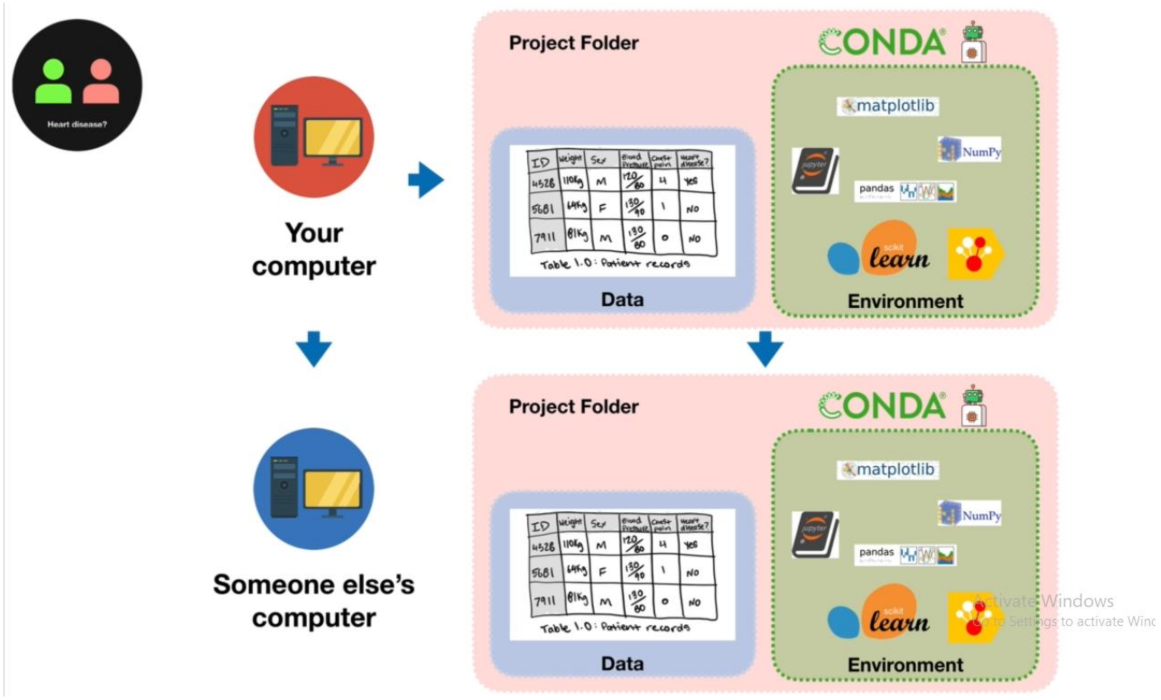


Conda environment

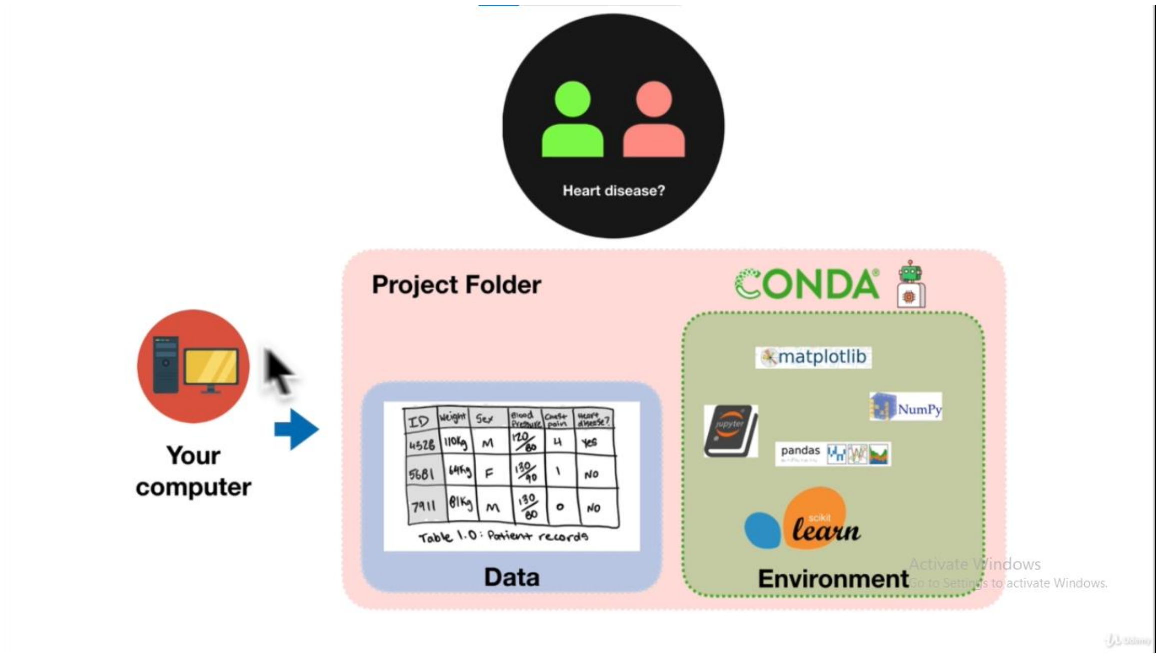
Conda is what you use to create this environment. It install and update the tools that you need.

If someone else wanted to start working on what you're working on with you instead of setting up their own workbench from scratch. Conda allows you to share with them the exact same set of tools and packages you've been using so they can get started straight away.

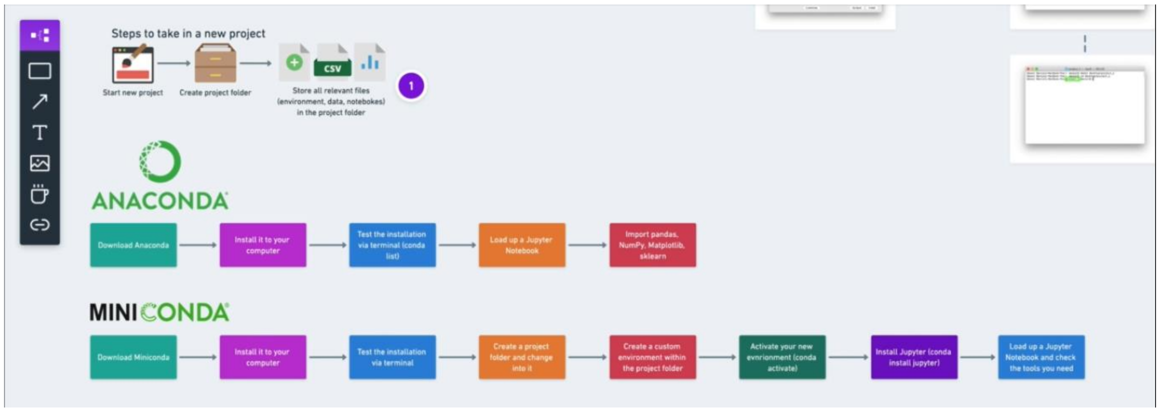
Condo helps to fix that dependency issue. It ensures that you can create this environment that's easily shareable and update able that you can share with someone else.



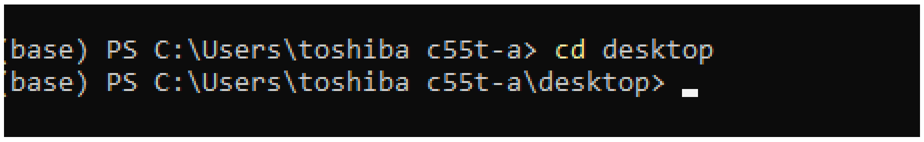
Setup for a particular project


Heart disease?
Your computer
Project Folder
CONDA
matplotlib
NumPy
jupyter
pandas
learn
Data
Environment
ID | weight | Sex | Blood Pressure | Chest pain | Heart disease?
4528 | 110kg | M | 120/80 | 4 | Yes
5681 | 64kg | F | 130/90 | 1 | No
7911 | 81kg | M | 130/80 | 0 | No
Table 1.0: Patient records
Activate Windows
Go to Settings to activate Windows.

Getting started

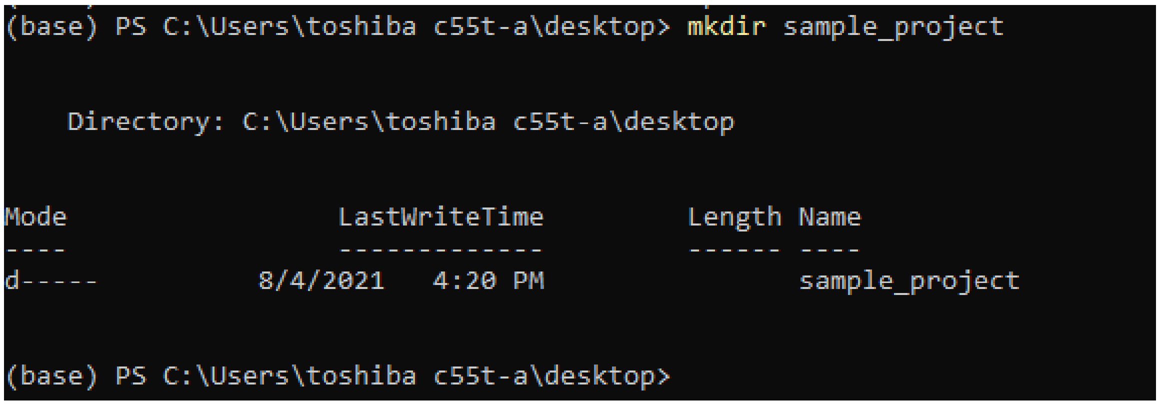

Steps to take in a new project
Start new project
Create project folder
CSV
Store all relevant files (environment, data, notebokes) in the project folder
1
ANACONDA
Download Anaconda
Install it to your computer
Test the installation via terminal (conda list)
Load up a Jupyter Notebook
Import pandas, NumPy, Matplotlib, sklearn
MINICONDA
Download Miniconda
Install it to your computer
Test the installation via terminal
Create a project folder and change into it
Create a custom environment within the project folder
Activate your new evnrionment (conda activate)
Install Jupyter (conda install jupyter)
Load up a Jupyter Notebook and check the tools you need

- Download mini-conda online
- Install mini-conda in your computer
- Open mini-conda prompt/terminal
- Change director to desktop



```
(base) PS C:\Users\toshiba c55t-a> cd desktop
(base) PS C:\Users\toshiba c55t-a\desktop> _
```

- Create a new folder / make directory



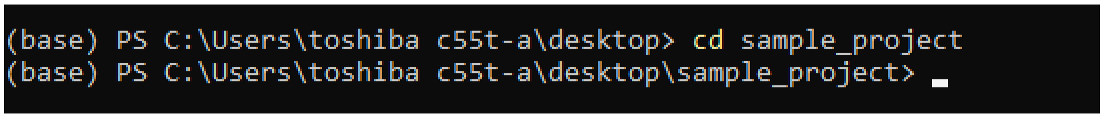
```
(base) PS C:\Users\toshiba c55t-a\desktop> mkdir sample_project


    Directory: C:\Users\toshiba c55t-a\desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          8/4/2021   4:20 PM                sample_project


(base) PS C:\Users\toshiba c55t-a\desktop>
```

- Change directory to newly created folder



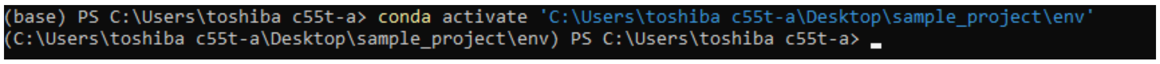
```
(base) PS C:\Users\toshiba c55t-a\desktop> cd sample_project
(base) PS C:\Users\toshiba c55t-a\desktop\sample_project>
```

- **Hint**: Use **dir** command to see all the files in the current folder

- **Create a custom environment within the project folder / use conda to create environment**

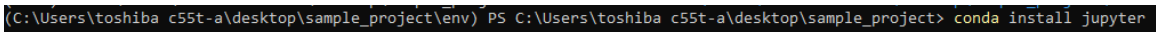
- Run the following command to install the packages:: **conda create --prefix ./env pandas numpy matplotlib scikit-learn**

- Now we have to activate the environment which we created, to access all the tools in the environment: **conda activate 'C:\Users\toshiba c55t-a\Desktop\sample_project\env'**



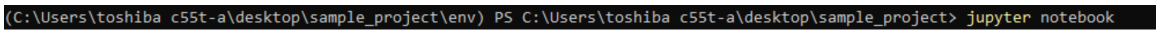
```
(base) PS C:\Users\toshiba c55t-a> conda activate 'C:\Users\toshiba c55t-a\Desktop\sample_project\env'
(C:\Users\toshiba c55t-a\Desktop\sample_project\env) PS C:\Users\toshiba c55t-a>
```

- We firstly forget to install jupyter so, installing it… **conda install jupyter**



```
(C:\Users\toshiba c55t-a\desktop\sample_project\env) PS C:\Users\toshiba c55t-a\desktop\sample_project> conda install jupyter
```

- Load/ open jupyter nootbook **: jupyter notebook** # this will open jupyter notebook in browser



```
(C:\Users\toshiba c55t-a\desktop\sample_project\env) PS C:\Users\toshiba c55t-a\desktop\sample_project> jupyter notebook
```

- When jupyter notebook is loaded, check the tool needed

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import sklearn

- To deactivate the environment maybe you wanted to get out and create a new environment. So, we're gonna type in **condat deactivate** what this will do is take us back to base.

- Now why would we want to do that. Well maybe we want to create a new environment.Maybe you want to work on another project such as sample project 2 and that uses a different environment.