# Course Notes

## Machine Learning 101

### What Is Machine Learning

* The goal of machine learning is to get machines to act more like humans.

### AI/Machine Learning/Data Science

* AI:
  + A human intelligence exhibited by machines.
  + Narrow AI:
    - Machines can be as good if not better at certain narrow tasks.
    - Can only do one thing well. Like spot heart disease from an Xray or win a game of chess.
  + General AI:
    - Can do multiple things like humans.
    - Currently we are a long way from this.
* Machine Learning:
  + A subset of AI.
  + An approach to try and achieve artificial intelligence through systems that can find patterns in data.
  + The science of getting computers to act without being specifically programmed.
* Deep Learning:
  + In the past may have been called deep neural networks although it is not really akin to a neural network like the brain.
  + A technique for implementing machine learning.
  + Can be thought of as a type of algorithm.
* Data Science:
  + Role of Data Science expert and Machine Learning expert can overlap.

### How Did We Get Here

* We started getting so much data that we needed smarter ways to use and learn from that data.
* The amount of data in the world is doubling every two years.
* A lot of data generated is unused.
* Machine Learning specialists are now needed to turn data that is unused into useful data.

### Types of Machine Learning

* Machine Learning is about predicting results based on incoming data
* Supervised Learning:
  + A subset of ML.
  + Think of it as a csv file with rows and columns labelled.
  + Classification: Does this test item fit the class we are looking for? Is this an apple or a pear.
  + Regression: Hire engineer based on inputs like age, experience, skillset etc.
  + Supervised Learning has a right and wrong.
* Unsupervised Learning:
  + Subset of ML.
  + For data that does not have labels or cannot be labelled.
  + No right or wrong. We don’t put in an input and get an output like, “This is a pear because it looks like the pears in the model.” Instead the LLM does what it wants and tells us what we need to know. We are not supervising it and telling it what to do. We are letting it go off by itself, play, try things, and come back and tell us what we should know, what might be valuable or interesting.
  + Clustering:
    - We give the AI or LLM a bunch of data and it gives us groups or results rather than what we specifically asked for.
    - We could give the AI/LLM a million YouTube videos and ask us to tell us what it finds. It will put things in groups or clusters and tell us something like, “People really love videos on cats.”
  + Association Rule Learning:
    - Associating different things to predict an outcome. Like associating multiple inputs about a customer and having the AI predict what that customer will want to buy.
* Reinforcement Learning:
  + Subset of Machine Learning.
  + Teaching machines through trial and error. Through rewards and punishment.
  + EG A program learning a game until it gets the highest score. Think the AI moving a plank in a game to bounce a ball back up to get points. At first the ball goes down and it get’s no points, a punishment. The AI then decides what if I move the plank to where the ball is going. It does and gets points, reward. The ball speeds up and goes down, punishment. AI decides I must move faster. It does and get’s more points, reward. And so on till it is an expert. It could do this in chess and learn what patters lead to rewards more often and what patters lead to loss or punishment more often.

### What is Machine Learning Round 2

* Using an algorithm/computer program to learn about patterns in data. Then take that algorithm and what it has learned and use it to make predictions about the future when given similar data.
* Machine Learning algorithms are also called models.
* A normal algorithm starts with the inputs and a set of instruction and we use it to get an output.
* A ML algorithm starts with the inputs and outputs and figures out the set of instructions or the algorithm that we can then use with similar inputs in the future.
* Machine Learning algorithms might try to figure out the instructions to get from inputs to the desired output 1000’s of times before it finds the right set of instructions.
* Data Analysis:
  + Looking at a set of data and gaining an understanding of it by looking at different examples. We might communicate our findings in graphs or charts.
* Data Science:
  + Running experiments on a set of data with the hope of finding actionable insight in it.
  + Machine Learning might be considered a part of data science.

Machine Learning 101 – Section Review Task. What is Machine Learning in 5 sentences or less: Machine Learning is giving a computer an input, our data, and telling it our desired output, the result we expect, and having the computer come up with a set of instructions or algorithm to get from the input to the output. We can then use this algorithm on similar sets of data inputs to predict what the output might be. An example of this could be taking in labelled data of 1000’s of images of a good running action, our input, and each image could be labelled with ‘good’ or ‘bad’, our output. We give this data to the AI, and it creates an algorithm or set of instructions to determine whether a running action is good or bad. We could then pass it an image of a running action and have it tell us if it is good or bad. Sometimes the output is not specifically defined and instead we want the LLM or AI to come up with an algorithm to give us what it thinks is useful. An example would be giving it a million YouTube videos and asking it what is interesting about this data. Lots of people love cats apparently.

## Machine Learning and Data Science Framework

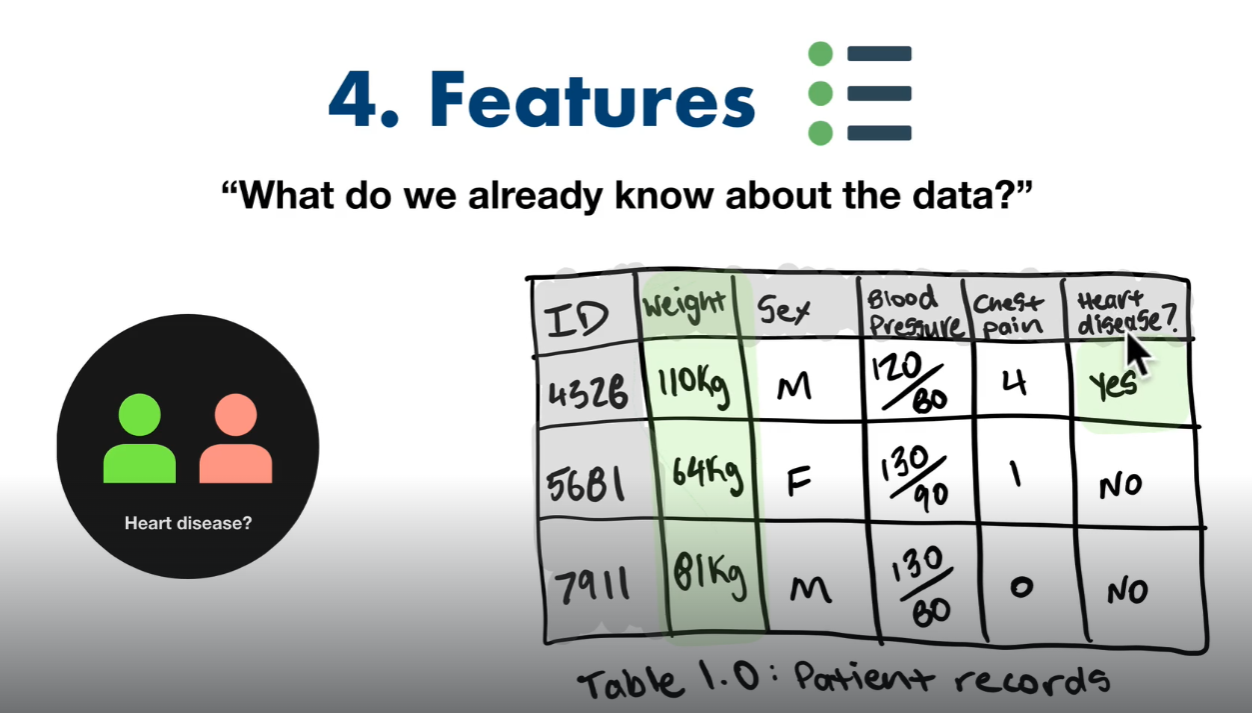
### Overview

* A Machine Learning and Data Science Framework is something you can use for future projects.

### Introducing Our Framework

* Machine Learning comes in 3 parts:
  + Data collection
    - Data we need to create our model.
  + Data modelling
    - Taking a data set and applying machine learning algorithm’s to find insights on that dataset.
    - The steps to do this are:
      * 1. Problem definition: What problem are we trying to solve?
      * 2. Data: What data do we have?
      * 3. Evaluation: What defines success?
      * 4. Features: What features should we model?
      * 5. Modelling: What kind of model should we use?
      * 6. Experiments: What have we tried/what else can we try?
  + Deployment
    - Once we created out model through the 6 step process above we can then deploy our model to users. This maybe through an application, through an API or some other way.
* Tips for success in Machine Learning and Data Science:
  + Write lots of code rather than sitting around overthinking the process.
  + Make mistakes faster rather than thy to make things perfect.
  + Build more projects faster building on what others have done before, using tools others have already created and shared. As opposed to building everything from scratch.
  + Learn what matters and get’s us to a point we can provide practical value in a job or an application for people as opposed to learning every little thing before we get started.

### 6 Step Machine Learning Framework

* A framework is a guide to approaching a problem using ML.
* It allows us to break down the task into smaller steps.
* The framework:
  + 1. Problem definition:
    - What problem are we trying to solve?
    - Is it a supervised or unsupervised problem?
    - Is it a classification or regression problem?
  + 2. Data:
    - ML consists of using algorithms to find patterns in data. So data is vital to a ML project.
    - What data do we have?
    - Is it structured data?
      * Rows and columns.
      * What you’d typically find in a spreadsheet.
    - Unstructured data?
      * Such as images or audio
    - We need to know what kind of data we have before we can start making decision on how to use ML with it.
  + 3. Evaluation:
    - What defines success?
    - Aiming to perfect your model could lead to getting a little bit better each time you improve it but end up taking forever to get to say 99% successful. Perfect model likely does not exist.
    - It is better to aim for a level of success that will solve your problem well enough to have real practical impact. An 80% success rate could well be enough to improve business bottom line significantly.
    - EG For this real estate ML project to be feasible we need a model that can predict the cost of houses with a 95% accuracy.
  + 4. Features:
    - 
    - What features should we model?
    - What do we already know about the data?
    - EG for predicting whether someone has heart disease, you might use their body weight as a feature. Since body weight is a number, it is called a numerical feature. Data may show that when someone is over X weight their likelihood of have heart disease increases by X%.
    - There are different types of features such as categorical or derived.
    - Features can be thought of as columns in a table?
    - A ML algorithm’s goal is to turn features into patterns to make predictions.
  + 5. Modelling:
    - What kind of model should we use?
    - Many of the most useful ML algorithm’s have already been coded for us.
  + 6. Experiments:
    - What have we tried/what else can we try/How can we improve?
    - Through experimenting we might find that the data we have is not suited to our problem.
    - We might find that our model does not get the success we require.
    - As we cycle through this framework remember it is a guide. As we experiment and improve we might find it useful to go back and improve some steps but not need to improve every step.
    - The problem might be poor data or not enough data. We can add more data and then test again to see if we better meet our success metric without going through all steps the framework on each cycle. On another cycle we might hypothesize that we need different or more features and then we can test. perhaps feasibility needs to be looked again to see if a slightly lower success rate is feasible and makes business sense.

### Types of Machine Learning Problems – 1. Problem Definition

* What problem are we trying to solve?
* When shouldn’t you use machine learning?
  + If a simple hand coded instruction program will do the job, then do this.
  + Other than these simple examples where you know the instructions and parameters you can find value in ML.
* Main Types of ML:
  + Match out business problem to the main types of ML problem that fits.
  + Supervised Learning:
    - You have data and labels. ML algorithm tries to use the data to predict a label.
    - A model might be trained on a set of health data like age, weight, and waist size. The Labelled result in this data set might be live past 80 and the data yes or no. The model would then take a new person’s data and predict whether they will live past 80.
    - If the algorithm gets it wrong, it corrects itself and tries again. This act of correction is why it is called supervised.
    - So, we show the model what examples of what correct results are. The algorithm then comes up with an algorithm to predict the result for future similar data.
      * Who supervises and corrects the algorithm/model. Is it a human or does the model correct itself? If so, how?
    - The main types of Supervised Learning are:
      * Classification:
        + Is this example, this data, one thing or another?
        + Binary Classification is two options. EG Heart Disease or not?
        + Multi-class classification is more than two options. EG What dog breed is this?
      * Regression:
        + Trying to predict a number. Sometimes called a continuous number, which simply means a number which can go up or down.
        + EG How much will this house sell for? Or, How many people will buy my product?
  + Unsupervised Learning:
    - No labelled target.
    - Pass data to the ML algorithm and ask it if it can find anything interesting?
    - Or pass data to the ML algorithm, like purchase history, and ask it to predict what else the user might want to buy.
    - Clustering:
      * Putting groups of similar examples together
      * Recommending what music someone might like based on what they have listened to before often start out as unsupervised learning problems.
  + Transfer Learning:
    - Uses what one model has learned to create a model for a similar problem.
    - For example, take a model that is used to identify different car types, then fine tune that model to identify different plane types.
    - This is valuable because training a ML algorithm model, which means letting it find all the patterns in data, can be a very expensive task.
    - To find patterns in data a ML algorithm has to perform millions of calculations.
    - In our example the car type model may have already learned what grass, trees and cement paths look like and can discount them from the part of the image, the car, we actually want. A new model for a plan would also find this useful so if it can have that learning transferred to it then it does not have the expensive task of learning it all again.
  + Reinforcement Learning:
    - Rewarding a ML algorithm if it wins and punishing it if it loses.
    - EG If the model wins then it get’s +1, loses -1. It is told its goal is to get the highest score possible.
    - Yet to find it’s way into common use and practical application very often.
* Matching our problem to a ML Problem:
  + Supervised learning:
    - I know my inputs and outputs.
    - EG input: scan of heart, output: does patient of heart disease.
    - EG input: house data like rooms, bathrooms, garages etc, output: house price.
  + Unsupervised learning:
    - I’m not sure of the outputs but I have the inputs.
    - EG input: customer purchase history. AI can then group this data in ways that might provide useful insights.
  + Transfer learning:
    - I think my problem is similar to an ML problem someone else has already solved.

### Types of Data

* Structured:
  + Things rows and columns, like a csv file.
  + EG Customer purchase transactions
  + EG Patient records like weight, blood pressure, chest pain and do they have heart disease.
* Unstructured:
  + Images, audio, videos, even emails.
  + These can be turned into numbers to create structure but typically one example of the data can be very different to the next.
  + EG a pic of a dog sitting can look very different to a pic of a dog lying down. Look at a dog from the front when it’s body is hidden behind its head could look have AI saying dogs look like circles. Then look from above it might say dogs look like rectangles.
* Static Data
  + Can be structured or unstructured.
  + Does not change over time.
  + Csv is a common example of static data format
  + More data the better, the better chance of finding patterns.
  + What someone purchased on x date won’t change. It is hard set data.
* Streaming Data
  + Can be structured or unstructured.
  + Data that is constantly changed over time.
  + ML algorithm model might start using static data but then move to working with a dataset that is constantly being updated when we move to production.
* A common data science workflow:
  + Open a csv file of our static data >
  + In a Jupyter notebook which is a tool for building machine learning project >
  + Explore the data, perform data analysis using pandas (a python library for data analysis) and making visualisations such as graphs >
  + Comparing different data points using matplotlib >
  + Build machine learning model using scikit learn >

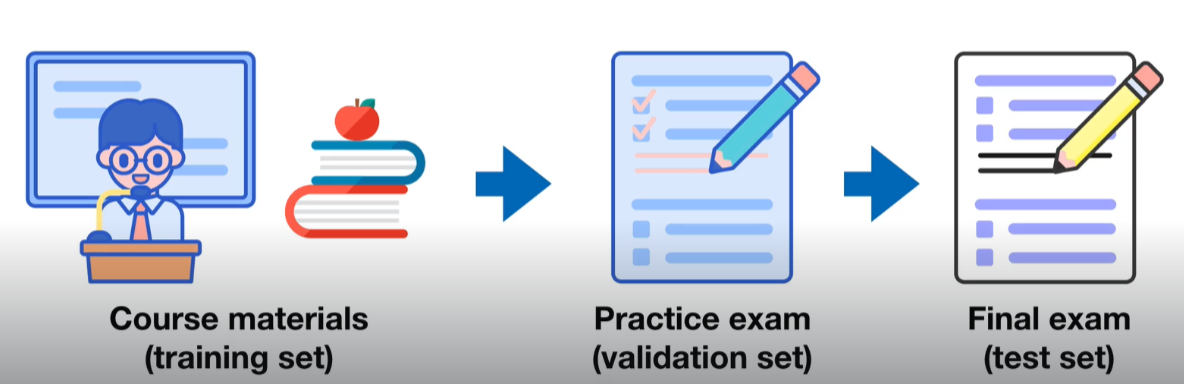
### Types of Evaluation

* What defines success for us?
* Evaluation metric: how well our ML algorithm predicts the future.
* If it is a safety issue or life and death you likely want a high level of accuracy.
* For a business decision it might be lower as long it can be good enough to improve overall profit.
* There are different evaluation metrics for different problems.
* For Classification measure accuracy, precision and recall.
* For Regression measure Mean absolute error (MAE), Mean squared error (MSE), and Root mean squared error (RMSE). You’ll use these to limit how different the predicted number is to a current example. Like the predicted car price compared to the current sale price.
* For recommendation measure Precision at K.
* Evaluation metrics can change as a project goes on and we learn more.

### Features in Data

* What do we already know about the data?
* Features = different forms of data.
* The different forms of data within structured and unstructured data.
* Features = variables. Features can be called feature variables.
* EG weight, sex, heart rate, chest pain? Are features of the data.
* They are the feature variables that are used to predict the target variable.
* There are different kinds of feature variables:
  + Numerical features: Like a number
  + Categorical features: Means one thing or another, like sex, is or is not categories.
  + Derived features: When someone looks at the data and creates a new feature using the existing features. An example is “Visit in last year?” could be derived from patient history visit timestamps. This is called feature engineering.
* Unstructured data has features too they are just less obvious.
* Examples might be “Number or rectangles coming out of dogs body”, or “number of circles on dogs head” An ML model figures them out these features on its own. We don’t have to come up with them.
* A feature works best in an ML algorithm if many of the samples have it. If only 10% of the records in our data have a value for a certain feature it is not a very useful feature.
* This is called feature coverage. 100% feature coverage is ideal. If very low feature coverage either remove feature or try and fill in the data.

### Modelling

* What machine learning model should we use?
* Choose our 3 Data sets:
  + Modelling can be broken down into three parts:
    - Choosing and training a model.
    - Tuning a model.
    - Model comparison.
  + 3 Sets
    - 
    - The most important concept in machine learning.
    - Split data into three different sets:
      * Training – train model on this.
      * Validation – tune model on this.
      * Test – test and compare our different models on this.
  + Generalization: The ability for a machine learning model to perform well on data it hasn’t seen before. Consider the example above where a student (our model) learns on data – training set. The model can then answer questions in the final exam that it has never seen before.
  + For our model to become great at predicting data, you want to avoid it becoming a memorization machine. It is important that our 3 data sets are different.
  + If we have 100 records we might shuffle them into a random order then take 70-80% for training, 10-15% to test and fine tune on and then 10-15% to test on. These percentages are standard practice.
  + Next step, once we have split the data, Is we would choose a model, feed it our data for training. Then we will see if we can improve on results in the validation set. You can then test on test set.
* Choose a model
  + There are many prebuilt models so you don’t have to come up with your own ML algorithm.
  + What kind of ML algorithm suits what kind of problem.
  + Some models work better on different types of data.
  + Working with structured data:
    - Decision Tree models work best.
    - Examples are Random Forest and gradient boosting algorithms like CatBoost and XGBoost.
  + Working with unstructured data:
    - Deep Learning (some call it neural networks) and Transfer learning tend to work best.
  + Choose model > Train model.
    - Goal is to line up the inputs and outputs
    - Goal: minimise time between experiments.
      * One way to do this is train on say 10% of the Training Dataset first and see how it goes.
      * You could start with a more simple model to start and see if it gets the results we are look for. Deep Learning models for example take longer to train than other models.
      * ML is HIGHLY iterative so we want to minimize the time between iterations. We might ask is it worth spending 3 hours on training to get say 94% accuracy when we can get say 92% accuracy in half the time spent on a different model.
  + Try fail, try fail a lot.
  + Start small and build up. Start with a smaller dataset and simpler model first to get practical results then increase complexity as needed.
* Tune our model:
  + Model can be tuned for different types of data.
  + Ideally test on the Validation data set. If not use the training data.
  + Many models can have different hyper parameters that can be adjusted. Think temp dial on an oven that we might need to adjust to get a better result on each iteration or attempt at cooking.
  + Different types of models have different hyper parameters.
  + EG a Random Forest model will allow you to adjust the number of trees.
  + EG a Deep Learning model will allow you to adjust the number of layers.
* Compare our models:
  + How will our model perform in the real world?
  + Test out our model on the Test set of data.
  + This is the final exam.
  + This will be the final step and give us an indication of performance once deployed in production.
  + This is a good way to see how it generalizes or performs on similar data it has not ever seen before.
  + A good model will yield similar results on the Training, Validation, and Test sets.
  + A slight decrease in performance on the test set when compared to the training set is not uncommon.
  + Underfitting: When the Training set performance is significantly lower than the test set. This is cause for concern.
    - Data mismatch: Occurs when the data you are testing on is too different from the data you are training on. Such as having different features in the test data to the training data.
    - Combat underfitting by:
      * Trying a more advanced model.
      * Increasing the model hyperparameters.
      * Reduce the number of features. To many features can lead to a model struggling to find patters in them.
      * Train the model for longer on more data.
  + Overfitting: When the Test set performance is higher than the Training set performance. This is also cause for concern.
    - Can happen when we see data leakage, which is when data from the training set ends up in the test set.
    - To reduce overfitting:
      * Collect more data. More data means the model might not be able to find all patterns and will need to generalize?
      * Try a less advanced model. Less common.
  + Overfitting and underfitting are both examples of a model not being able to generalize well.
  + When comparing different models make sure they were trained on the same dataset.
  + Consider how long a all factors not just accuracy when deciding which model is best for your needs. Is the accuracy of a model slightly better but takes too long to predict? Is the training time of a model so large for just a 2% better accuracy that iterating through the process to get better results would be untenable?

### Experimentation

* Machine learning projects become tool matching projects.
* Look at a project at the start and work out which tools you need for each step.
* How a typical project might work:
  + Define Problem: Client comes to you with a set of data and asks you to find insights in it. There might be some back and forth as you come up with a clear problem to solve. Step 1.
  + Data Analysis: Then we will do Data Analysis. This is steps 2, 3, and 4 in our process. We have a look at out data, come up with an evaluation metric that makes sense and look at what features of the data we have to deal with.
  + Build ML model: Then we build our machine learning model using the features we found in the data to predict some target. Step 5.
  + At this point we have a working model but the client may ask what can we do better? So we go back and start experimenting. We might try a different model, try with different data, more data, tinker with the hyperparameters of our current model. Do we need better data, could it better labelled? Is there to many records that don’t have data in some fields/features? Is the data inaccurate? Is our evaluation metric to unreasonable? Do we need the model to be that accurate or is it still very financially beneficial with a lower accuracy? These and many more options could be where we try things.

### Tools We Will Use

* Setting up computer for ML and Data Science (creating the workshed):
  + Anaconda:
    - The hardware store of data science and machine learning tools.
    - Whole project tool
  + Jupyter Notebooks:
    - To write python code and communicate our work.
    - Jupyter is the tool used to manage our overall project?
    - Whole project tool
  + Pandas:
    - Data analysis tool.
  + Matplotlib:
    - Data analysis tool.
  + NumPy:
    - Data analysis tool.
  + Tensor Flow:
    - Building machine learning models.
  + PyTorch:
    - Building machine learning models.
  + SciKit Learn:
    - Building machine learning models.
  + Dmlc XGBoost:
    - Building machine learning models.
  + CatBoost:
    - Building machine learning models.
* It is not as important to know the intricacies of each of these tools, to know all the functions of these libraries, as it is to know which tool to use for which type of problem.

## NumPy

* Is a library.
* Similar to Python Lists
* One of the most used libraries when it comes to machine learning.
* 2 Main reasons we want to use NumPy:
  + Much faster than Python Lists. We do a lot of computation in machine learning so being able to do things faster is vital.
  + NumPy helps us turn data into numbers that computers can understand?
* Backbone of all numerical computing in Python.
* NumPy stands for Numerical Python.
* Turns data into a series of numbers.
* A machine learning algorithm will then learn the patterns in those numbers.
* NumPy used across almost the entire ML/DS pipeline.
* It’s fast due to optimisations written in C code.
* Can use these optimisations using Python.
* Backbone of other Python scientific packages. EG is basis of the pandas package.