**Section 3: Machine Learning and Data Science Framework**

**# Introducing Our Framework**

* Focus on practical solutions and writing machine learning code
* Steps to learn machine learning
  + Create a framework
  + Match to data science and machine learning tools
  + Learn by doing

**#** [**6 Step Machine Learning Framework**](https://www.mrdbourke.com/a-6-step-field-guide-for-building-machine-learning-projects/)

* Problem definition: What problems are we trying to solve?
  + Supervised or Unsupervised
  + Classification or Regression
* Data: What kind of data do we have?
  + Structured or Unstructured
* Evaluation: What defines success for us?
  + Example: House data -> Machine learning model -> House price
  + Predicted price vs Actual price
* Features: What do we already know about the data?
  + Example: Heart disease? Feature: body weight
  + Turn features such as weight into patterns to make predictions whether a patient has heart disease?
* Modelling: Based on our problem and data, what model should we use?
  + Problem 1 -> Model 1
  + Problem 2 -> Model 2
* Experimentation: How could we improve/what can we try next?

**Types of Machine Learning Problems**

* When shouldn't you use machine learning?
  + When a simple hand-coded instruction based system will work
* Main types of machine learning
  + Supervised Learning
  + Unsupervised Learning
  + Transfer Learning
  + Reinforcement Learning
* Supervised Learning: data and label -> make prediction
  + Classification: Is this example one thing or another?
    - Binary classification = two options
    - Example: heart disease or no heart disease?
    - Multi-class classification = more than two options
  + Regression: Predict a number
    - Example: How much will this house sell for?
    - Example: How many people will buy this app?
* Unsupervised Learning: has data but no labels
  + Existing Data: Purchase history of all customers
  + Scenario: Marketing team want to send out promotion for next summer
  + Question: Do you know who is interested in summer clothes?
  + Process: Apply labels such as Summer or Winter to data
  + Solution: Cluster 1 (Summer) and Cluster 2 (Winter)
* Transfer Learning: leverages what one machine learning model has learned in another machine learning model
  + Example: Predict what dog breed appears in a photo
  + Solution: Find an existing model which is learned to decipher different car types and fine tune it for your task
* Reinforcement Learning: a computer program perform some actions within a defined space and rewarding it for doing it well or punishing it for doing poorly
  + Example: teach a machine learning algorithm to play chess
* Matching your problem
  + Supervised Learning: I know my inputs and outputs
  + Unsupervised Learning: I am not sure of the outputs but I have inputs
  + Transfer Learning: I think my problem may be similar to something else

**Types of Data: What kind of data do we have?**

* Different types of data
  + Structured data: all of the samples have similar format
  + Unstructured data: images and natural language text such as phone calls, videos and audio files
  + Static: doesn't change over time, example: csv
    - More data -> Find patterns -> Predict something in the future
  + Streaming: data which is constantly changed over time
    - Example: predict how a stock price will change based on news headlines
    - News headlines are being updated constantly you'll want to see how they change stocks
* Start on static data and then if your data analysis and machine learning efforts prove to show some insights you'll move towards streaming data when you go to deployment or in production
* A data science workflow
  + open csv file in jupyter notebook (a tool to build machine learning project)
  + perform data analysis with panda (a python library for data analysis)
  + make visualizations such as graphs and comparing different data points with Matplotlib
  + build machine learning model on the data using scikit learn to predict using these patterns

**Types of Evaluation: What defines success for us?**

* 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
* data -> machine learning model -> predict: heart disease? -> accurancy 97.8%
* predicting whether or not a patient has heart disease is an important task so you want a highly accurate model
* Different types of metrics for different problems
  + Classification: accurancy, percision, recall
  + Regression: Mean absolute error (MAE), Mean squared error (MSE), Root mean squared error (RMSE)
  + Recommendation: Precision at K
* Example: Classifying car insurance claims
  + text from car insurance claims -> machine learning model -> predict who caused the accident (person submitting the claim or the other person involved ?) -> min 95% accuracy who caused the accident (allow to get it wrong 1 out of 20 claims)

**Features In Data: 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: predict heart disease problem
  + Features of the data: weight, sex, heart rate
  + They can also be referred to as feature variables
  + We use the feature variables to predict the target variable which is whether a person has heart disease or no.
* Different features of data
  + numerical features: a number like body weight
  + categorical features: sex or whether a patient is a smoker or not
  + derived features: looks at different features of data and creates a new feature / alter existing feature
    - Example: 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.
    - feature engineering: process of deriving features like this out of data
* Unstructured data has features too
  + a little less obvious if you looked at enough images of dogs you'd start to figure out
  + legs: most of these creatures have four shapes coming out of their body
  + eyes: a couple of circles up the front
  + machine learning algorithm figure out what features are there on its own
* What features should you use?
  + a machine learning algorithm learns best when all samples have similar information
  + feature coverage: process of ensuring all samples have similar information

**Modelling Part 1 - 3 sets**

* Based on our problem and data, what model should we use?
* 3 parts to modelling
  + Choosing and training a model
  + Tuning a model
  + Model comparison
* The most important concept in machine learning (the training, validation and test sets or 3 sets)
  + Your data is split into 3 sets
    - training set: train your model on this
    - validation set: tune your model on this
    - test set: test and compare on this
  + at university
    - training set: study course materials
    - validation set: practice exam
    - test set: final exam
  + generalisation: the ability for a machine learning model to perform well on data it has not seen before
* When things go wrong
  + Your professor accidentally sent out the final exam for everyone to practice on
  + when it came time to the actual exam, everyone would have already seen it now
  + Since people know what they should be expecting they go through the exam
  + They answer all the questions with ease and everyone ends up getting top marks
  + Now top marks might appear good but did the students really learn anything or were they just expert memorization machines
  + 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
* split 100 patient records
  + training split: 70 patient records (70-80%)
  + validation split: 15 patient records (10-15%)
  + test split: 15 patient records (10-15%)

**Modelling Part 2 - Choosing**

* Based on our problem and data, what model should we use?
* 3 parts to modelling
  + Choosing and training a model: training data
  + Tuning a model: validation data
  + Model comparison: test data
* Choosing a model
  + Problem 1 -> model 1
  + Problem 2 -> model 2
  + Structured Data: [CatBoost](https://catboost.ai/), [XGBoost](https://github.com/dmlc/xgboost), [Random Forest](https://towardsdatascience.com/understanding-random-forest-58381e0602d2)
  + Unstructured Data: Deep Learning, Transfer Learning
* Training a model
  + inputs: X(data) -> model -> predict outputs: y(label)
  + Goal: minimise time between experiments
    - Experiment 1: inputs -> model 1 -> outputs -> accurancy (87.5%) -> training time (3 min)
    - Experiment 2: inputs -> model 2 -> outputs -> accurancy (91.3%) -> training time (92 min)
    - Experiment 3: inputs -> model 3 -> outputs -> accurancy (94.7%) -> training time (176 min)
  + Things to remember
    - Some models work better than others and different problems
    - Don't be afraid to try things
    - Start small and build up (add complexity) as you need.

**Modelling Part 3 - Tuning**

* Based on our problem and data, what model should we use?
* Example: Random Forest - adjust number of trees: 3, 5
* Example: Neural Networks - adjust number of layers: 2, 3
* Things to remember
  + Machine learning models have hyper parameters you can adjust
  + A model first results are not it's last
  + Tuning can take place on training or validation data sets

**Modelling Part 4 - Comparison**

* How will our model perform in the real world?
* Testing a model
  + Data Set: Training -> Test
  + Performance: 98% -> 96%
* Underfitting (potential)
  + Data Set: Training -> Test
  + Performance: 64% -> 47%
* Overfitting (potential)
  + Data Set: Training -> Test
  + Performance: 93% -> 99%
* Balanced (Goldilocks zone)
* Data leakage -> Training Data overlap Test Data -> Overfitting
* Data mismatch -> Test Data is different to Training Data -> underfitting
* Fixes for underfitting
  + Try a more advanced model
  + Increase model hyperparameters
  + Reduce amount of features
  + Train longer
* Fixes for overfitting
  + Collect more data
  + Try a less advanced model
* Comparing models
  + Experiment 1: inputs -> model 1 -> outputs -> accurancy (87.5%) -> training time (3 min) -> prediction time (0.5 sec)
  + Experiment 2: inputs -> model 2 -> outputs -> accurancy (91.3%) -> training time (92 min) -> prediction time (1 sec)
  + Experiment 3: inputs -> model 3 -> outputs -> accurancy (94.7%) -> training time (176 min) -> prediction time (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 apple
    - Model 1 on dataset 1
    - Model 2 on dataset 1
  + One best performance Metric does not equal the best model

**Experimentation**

* How could we improve / what can we try next?
  + Start with a problem
  + Data Analysis: Data, Evaluation, Features
  + Machine learning modelling: Model 1
  + Experiments: Try model 2
* 6 Step Machine Learning Framework questions
  + Problem definition: What kind of problems you face day to day?
  + Data: What kind of data do you use?
  + Evaluation: What do you measure?
  + Features: What are features of your problems?
  + Modelling: What was the last thing you testing ability on?

**Tools We Will Use**

* Data Science: 6 Step Machine Learning Framework
* Data Science: [Anaconda](https://www.anaconda.com/), [Jupyter Notebook](https://jupyter.org/)
* Data Analysis: Data, Evaluation and Features
* Data Analysis:[pandas](https://pandas.pydata.org/), [Matplotlib](https://matplotlib.org/), [NumPy](https://numpy.org/)
* Machine Learning: Modelling
* Machine Learning: [TensorFlow](https://www.tensorflow.org/), [PyTorch](https://pytorch.org/), [scikit-learn](https://scikit-learn.org/stable/), [XGBoost](https://xgboost.ai/), [CatBoost](https://catboost.ai/)
* [Elements of AI](https://www.elementsofai.com/)