Today the videos covered an introduction to what machine learning is. On top of that they also talked about different approaches to machine learning based on the different needs of a situation depending on the problem which you are trying to solve.

The process of machine learning is as follow:

Data collection → Data Modeling → Deployment of Model

Data collection:
Consists of the collection of data that will be used by the data modeling to recognize patterns.

Data Modeling:

- **Types of Problems**
    - Classification
        - Using data to determine whether someone is likely to have a heart condition
    - Regression (prediction)
        - Using data to determine whether a stock's price will rise or fall
- **Types of Machine Learning**
    - Supervised
        - Used when you know what the end result should be
            - An example could be if you use a data set of spam emails and regular emails to train a model to recognize the difference (you would be able to tell it if it was right or wrong to help it improve)
    - Unsupervised
        - Used when you don’t know what the end result should be
            - An example of this is if you want to improve the cost effectiveness of a business’ advertising outreach you could using unsupervised learning on large amounts of customer data
                - Doing this would allow you to discover patterns among the customers and separate them into respective groups, using their commonalities to advertise more effectively to those individuals.
    - Transfer
        - Transfer Learning is the act of using an existing models logic to jumpstart your own
            - An example of this is, if you are training a model to learn to recognize a dog you can transfer the logic from a model used to recognize cars because it will enable your model to understand everything that model did for basic things like trees or grass
                - By doing this we reduce the learning curve for our model and allow it to understand what previous models already know without having to go through the same tedious process.

    - Reinforcement
        - Reinforcement learning is the least used type, it is often seen in game playing models as it utilizes a reward system to determine levels of success and the success of changes
            - An example is in a simple video game, if the model is trained to pursue a higher score, each time it does something that improves its score it knows that that thing was correct, over time this allows it to self improve.
- **Types of Data**
    - Structured
        - Structured data is data that you would see represented in something like a spreadsheet, where data is labeled and separated into groups.
    - Unstructured
        - Unstructured data is data that consists of unlabeled things, for example images or audio recordings.
    - Streaming
        - Streaming data is data that is constantly changing, this is the type of data that would often be used in a regression model to make real time predictions about things like stock prices.

- **Evaluation**
    - Deciding what metric you want to measure the success of your model
        - In the case of detecting heart disease you would be judging the model based on its accuracy (example 98% accuracy could be a goal)
        - In the case of discovering which 10 products you should market to each customer this value is completely different than that of the heart disease detection model.

- **Features / Feature Variables**
    - What do we already know about the data?
        - In the case of the heart disease detection model the feature variables in the data would be things like: (weight, age, gender…)
            - **Derived Features** are features which can be derived from data, for example in the heart condition model we could use data of medical history to create a feature showing whether or not they had a checkup within the last year.
        - The target variable is the variable which our model will populate, in this case whether or not an individual has a heart disease.

- **Modeling**
    - Based on our problem and data, what model should we use?
    - Splitting Data

        - Just like how for an exam in school you study the material and then take a practice exam before the real thing, for machine learning their is a similar process
            - An example of this in the heart condition model scenario would be, if your data is comprised of 100 patients, 70 patient files would be used for training (Learning throughout the semester), 15 would be used for validating and fine tuning the model (practice exam), and 15 would be used for the final test of the models effectiveness (exam).
                - The reason we do this is the guarantee that our model is actually learning patterns that it can apply to data it has never seen before successfully, and is not just memorizing.
                - If the data we train the model on is the same as the data we test it on, that would be like a professor giving the real exam as the practice exam, the grades would be inflated.
                - By testing the model on unseen data we get to see how it adapts to real world situations and can apply the knowledge it has obtained thus far to cope with unseen information.
    - Once the data has been split into these 3 categories the process begins
        - Choosing and training model → Tuning model → Model Comparison
    - **Choosing and Training a model**
        - A model is an algorithm, with many models already existing. We don't need to reinvent the wheel, we just need to know which algorithms are best applied to which problems to choose the best model for our situation.
            - For structured data, decision tree models are often used (CatBoost, XGBOOST, Random Forest)
            - For unstructured data, deep learning, neural networks and transfer learning are often used.
    - **Tuning a Model**
        - Based on the model you are using their are certain parameters you can adjust, in the tuning stage you use logic and reasoning to determine which parameters in your model may need to be adjusted to achieve a better result.
    - **Model Comparison**
        - Compare models with models trained on the same data and given the same input
        - Avoid overfitting models (100% accuracy) and underfitting models as they will lead to poor results in testing / real world scenarios.
            - Overfitting is when a model is “too good” and starts to see irrelevant patterns that will hinder its success when applied to real world data / scenarios. (Just because a model is working extremely well on a training data set does not mean that this will translate to real world use)

    - **Experimenting**
        - No model is ever perfect, we should always try new ideas and changes to fine tune and improve our model after its creation.

Conda Environment Setup:
(CONDA DOCUMENTATION:
https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#sharing-an-environment)

- Once Conda is installed use terminal to create a new folder in which we will create an environment
    - To create the environment follow these steps
        - cd into folder
        - conda create --prefix ./env numpy pandas jupyter … (any other packages)
            - If we want to add packages later just use **conda install** inside of the environment
        - Once the environment is created, to use it it must be activated
            - conda activate /Path/To/Environment
            - Once the environment is activated we can type jupyter notebook to launch jupyter notebook in our environment and begin programming with the packages we have included!
        - To exit jupyter notebook just use control c in terminal to kill current process
        - To exit conda environment just use conda deactivate

Common issues:
When creating the environment the capitalization of directories in the path matters, I used different capitalization for some directories in the path at different instances in this process and ended up with two duplicated environments and had to wipe them both and restart.