Data

Machine learning and data

- Machine learning is learning from data
  - Such approaches are often called "data driven"
- We need to understand about data:
  - How to represent it
  - Where it comes from
  - Things we do with it before feeding it to ML algorithms

Data types

- **Discrete values**: "yes" / "no", "red" / "white" / "blue"
  - Serve as outputs of **classification**
  - Does not imply ordering
  - We sometimes represent them with integers
    - but be careful: make sure you don't perform arithmetic operations and $<$ comparisons with them
  - **One hot** representation: eg. "[0, 0, 1.0]" represents blue
- **Integer values**: counts of units
  - It is often useful to convert them to floating point

- **Floating point numbers**:
    - Inputs and output of **regression** problems
    - We often **normalize** them - bringing their range to [0.0, 1.0] with 0.0 being the minimum value and 1.0 being the maximum

- **Boolean values as floating point**
  - It is customary to represent false with 0.0 and true with 1.0
  - Intermediate values allow to represent probability, uncertainty, degrees of confidence, etc.
- **Date/Time**
  - Should be able to compare it!
  - Often represented as an integer number of seconds from a specific moment in time.

Complex data types

- **Fixed length arrays**
  - When they are short, you can see them as equivalents of structures in programming languages
  - Can be added, scaled
- **Time series**
  - Eg. price of a stock on the stock market
  - Eg. temperature readings of a patient

Complex data types cont'd

- **Text**
  - Usually, variable lenght list of characters
- **Voice / Sound**
  - Usually, variable lenght list of samples
- **Images**
  - Can be seen as an array of pixels (x, y) or a tensor with 3 values per pixel (x, y, color)
  - Values usually "normalized" to [0,1].
    - Careful: many external formats represent them as 0...255
  - Very often, we need to bring them to the same size before we can do something useful with them...
- **Video**
  - Sequences of images

Features

- Often data comes in form of records or rows:
  - eg. students in the class.
  - one row per students
- Each row contains a set of different data which we call **features**:
  - name, UCF id, date of birth, homework-1-submitted, points, etc.
  - We often have some kind of interpretation of the meaning.
- It is possible to create new features:
  - current-date - date of birth, rounded down to closest integer gives me "age"
  - age can give me boolean "legally-allowed-to-drink-alcohol"
  - points in class * age / ucf-id --> new feature of dubious usefulness

Choosing features

- More features are not always better
  - Unnecessary features slow down learning
  - They might confuse the machine learning system
  - Depends on the algorithm
  - At minimum: more features / more data / more work collecting it.
- **Engineering features** used to be a significant part of ML.
- Some ML algorithms can learn which features to pay attention to and which to ignore.
- Other ML algorithms can learn to extract their own features.

Where do you get your data from?

- Collect it from the problem you are studying by careful sampling and measurement.
  - Eg. choose 20 representative Covid patients, and measure their temperature etc.
- Extract from logs
  - Use the hospitals' records of Covid patients and recorded temperatures.
- Big data:
  - If you are Google collect all the messages of type "is a 105F fever means I have Covid" and try to infer how many have indeed Covid etc.
- Quality of data gets worse as the quantity increases.

Public datasets

- There are many publicly available datasets covering various topics
- Some of them can be used to train useful machine learning models
  - But you have to be ready to use the features the creator of the dataset found useful to collect
- Commonly used as learning tools and in competitions
- Others are used to compare and validate datasets
- Kaggle is a website for ML competitions
  - Many publicly available datasets

Private datasets

- For many companies, the data they collected and own are an important part of their business proposition

Preliminary exploration of the data

- Before you dump your data in a machine learning algorithms, **look at it**
- Check maximum / minimum / average values
- Check for unusually high and low values
- What are the data types?
- How many unique values are for a feature? Which are the most popular values?
    - Can you make sense of it? Plot a histogram!
    - Eg. most popular selling price for a house = 0
        - Why?
- Plot the numerical values in several ways.

Preprocessing and cleaning the data

- Many datasets are imperfect
    - Missing features - no information
    - Wrong / outlier features - with no relationship to true data
    - Noisy features - related to the true data, but with some kind of noise added
- Decisions must be made about what to do with each of these
    - Missing features, or data of the wrong type can trigger software failures
    - Outlier features (eg house price = 1 trillion dollars) can yield bad results, even if there are few of them

Possible cleaning choices

- Identify outliers
  - For instance, stipulate an acceptable range
- One choice: drop records with missing / outlier data
- Other choice: data imputation
  - Replace the data with an estimated value based on other available information
  - Average of the feature, most frequent value for the feature etc.
- Noise reduction techniques for different data types...
  - Sound, image etc.

What do we want to predict?

- Decide on your machine learning **output**
- Data type, encoding format, range
- Do you have supervised labels for it?

Choosing features

- Often it is advisable to choose the features that enter into your ML algorithm
  - It is is surely irrelevant, don't include it.
  - Predicting survival by patient id...
- Many algorithms force you to use specific data formats, eg. collections of floats.
- We will say more about this when discussing specific algorithms