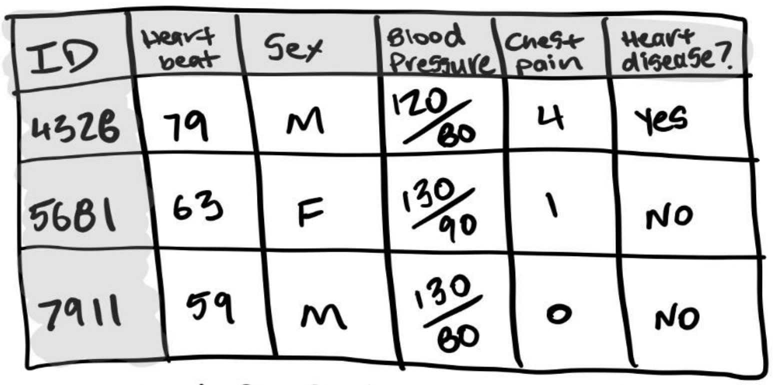
2. Data—If machine learning is getting insights out of data, what data do you have?

- The data you have or need to collect will depend on the problem you want to solve.

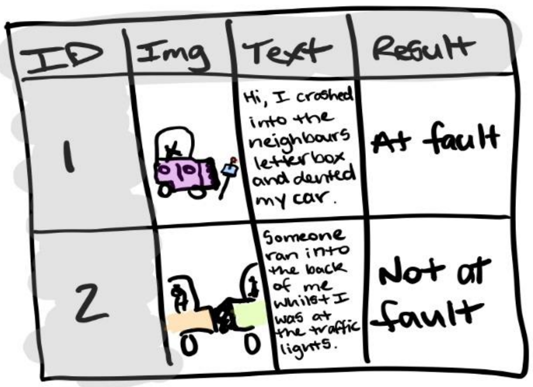
If you already have data, it's likely it will be in one of two forms. Structured or unstructured. Within each of these, you have static or streaming data.

- **Structured data**
  - Think a table of rows and columns, an Excel spreadsheet of customer transactions, a database of patient records. Columns can be numerical, such as average heart rate, categorical, such as sex, or ordinal, such as chest pain intensity.
- **Unstructured data**
  - Anything not immediately able to be put into row and column format, images, audio files, natural language text.
- **Static data**
  - Existing historical data which is unlikely to change. Your companies customer purchase history is a good example.
- **Streaming data**
  - Data which is constantly updated, older records may be changed, newer records are constantly being added.



ID	Heart beat	Sex	Blood Pressure	Chest pain	Heart disease?
4326	79	M	120/80	4	Yes
5681	63	F	130/90	1	No
7911	59	M	130/80	0	No

Table 1.0: Patient records

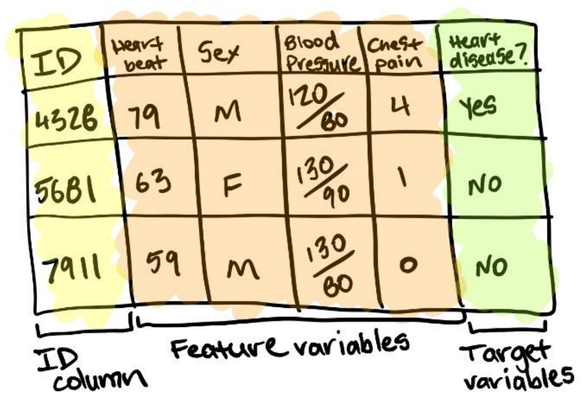


ID	Img	Text	Result
1		Hi, I crashed into the neighbours letterbox and dented my car.	At fault
2		Someone ran into the back of me whilst I was at the traffic lights.	Not at fault

Table 2.0: Car insurance claims

- Two examples of structured data with different kinds of data within it. Table 1.0 has numerical and categorical data. Table 2.0 has unstructured data with images and natural language text but is presented in a structured manner.

❖ Two examples of structured data with different kinds of data within it. Table 1.0 has numerical and categorical data. Table 2.0 has unstructured data with images and natural language text but is presented in a structured manner.


ID column
Feature variables
Target variables

ID	Heart beat	Sex	Blood Pressure	Chest pain	Heart disease?
4326	79	M	120/80	4	Yes
5681	63	F	130/90	1	NO
7911	59	M	130/80	0	NO

- ❖ Table 1.0 broken into ID column (yellow, not used for building machine learning model), feature variables (orange) and target variables (green). A machine learning model finds the patterns in the feature variables and predicts the target variables.

▪ For **unsupervised learning**, you won't have labels. But you'll still want to find patterns. Meaning, grouping together similar samples and finding samples which are outliers.

▪ For **transfer learning**, your problem stays a supervised learning problem, except you're leveraging the patterns machine learning algorithms have learned from other data sources separate from your own.

▪ **Remember**, if you're using a customers data to improve your business or to offer them a better service, it's important to let them know. This is why you see "this site uses cookies" popups everywhere. The website uses how you browse the site, likely along with some kind of machine learning to improve their offering.

3. Evaluation—What defines success? Is a 95% accurate machine learning model good enough?

- A 95% accurate model may sound pretty good for predicting who's at fault in an insurance claim. But for predicting heart disease, you'll likely want better results.

Other things you should take into consideration for classification problems.

- **False negatives—**
  - The model predicts negative, actually positive. In some cases, like email spam prediction, false negatives aren't too much to worry about. But if a self-driving cars computer vision system predicts no pedestrian when there was one, this is not good.
- **False positives—**
  - Model predicts positive, actually negative. Predicting someone has heart disease when they don't, might seem okay. Better to be safe right? Not if it negatively affects the person's lifestyle or sets them on a treatment plan they don't need.
- **True negatives—**
  - Model predicts negative, actually negative. This is good.
- **True positives—**
  - Model predicts positive, actually positive. This is good.
- **Precision—**
  - What proportion of positive predictions were actually correct? A model that produces no false positives has a precision of 1.0.
- **Recall—**
  - What proportion of actual positives were predicted correctly? A model that produces no false negatives has a recall of 1.0.
- **F1 score—**
  - A combination of precision and recall. The closer to 1.0, the better.
- **Receiver operating characteristic (ROC) curve & Area under the curve (AUC)—**
  - The ROC curve is a plot comparing true positive and false positive rate. The AUC metric is the area under the ROC curve. A model whose predictions are 100% wrong has an AUC of 0.0, one whose predictions are 100% right has an AUC of 1.0.
  - **For regression problems (where you want to predict number)**
    - you'll want to minimize the difference between what your model predicts and what the actual value is. If you're trying

to predict the price a house will sell for, you'll want your model to get as close as possible to the actual price. To do this, use MAE or RMSE.

- **Mean absolute error (MAE)—**
  - The average difference between your model's predictions and the actual numbers.
- **Root means square error (RMSE)—**
  - The square root of the average of squared differences between your model's predictions and the actual numbers.
- ❖ Use RMSE if you want large errors to be more significant. Such as, predicting a house to be sold at $300,000 instead of $200,000 and being off by $100,000 is more than twice as bad as being off by $50,000. Or MAE if being off by $100,000 is twice as bad as being off by $50,000.

- ❖ Let's say you're trying to recommend customers products on your online store. You have historical purchase data from 2010–2019. You could build a model on the 2010–2018 data and then use it to predict 2019 purchases. Then it becomes a classification problem because you're trying to classify whether someone is likely to buy an item.

4. Features—What features does your data have and which can you use to build your model?

- Not all data is the same. And when you hear someone referring to features, they're referring to different kinds of data within data.
- The three main types of features are **categorical**, **continuous** (or numerical) and **derived.**
- **Categorical features—One or the other(s)**.
  - For example, in our heart disease problem, the sex of the patient. Or for an online store, whether or not someone has made a purchase or not.

- **Continuous (or numerical) features—**
  - A numerical value such as average heart rate or the number of times logged in.
- **Derived features—**
  - Features you create from the data.

Often referred to as feature engineering. Feature engineering is how a subject matter expert takes their knowledge and encodes it into the data. You might combine the number of times logged in with timestamps to make a feature called time since last login. Or turn dates from numbers into “is a weekday (yes)” and “is a weekday (no)”.

- ❖ Text, images and almost anything you can imagine can also be a feature. Regardless, they all get turned into numbers before a machine learning algorithm can model them.
- ▪ **Some important things to remember when it comes to features.**
- **Keep them the same during experimentation (training) and production (testing)—**
- ➢ A machine learning model should be trained on features which represent as close as possible to what it will be used for in a real system.
- **Work with subject matter experts—**
- ➢ What do you already know about the problem, how can that influence what features you use? Let your machine learning engineers and data scientists know this.
- **Are they worth it?—**
- ➢ If only 10% of your samples have a feature, is it worth incorporating it in a model? Have a preference for features with the most coverage. The ones where lots of samples have data for.
- **Perfect equals broken—**
- ➢ If your model is achieving perfect performance, you’ve likely got feature leakage somewhere. Which means the data your model has trained on is being used to test it. No model is perfect.

.