EXAMPLE: Using ML to detect if certain cc transaction are fraud.

① Assume we have historical data that represents past data of this problem.

Each column represents independent attributes of the transactions.

Merchant Distance	Transaction Amount	Merchant Code (Type)	Is Fraud
5	39.01	1	0 ← FALSE
2	112.81	1	1 ← TRUE
8	4.99	1	0
6	1115.67	1	1
0.5	97.96	4	1
10.11	100	3	0
4	1.15	1	0
29.79	5.87	1	0
45.47	15.93	2	1
4	2500	1	0
0.95	25.66	2	0
21.33	2	5	0

Location of transaction vs. account

② The goal is to access riskiness of new transaction.

↳ EX:

Merchant Distance	Transaction Amount	Merchant Code (Type)
2.44	98.88	2

} This data will be inputs into your FRAUD MODEL

ML will help us find the most relevant subset (similar users/transactions) and then make guesses on that group. This is basically GENERALIZATION & ACTION

CAUTIONARY: THINGS TO CONSIDER WHEN USING ML

Let the application drive the solution, NOT the other way around.

↳ ML is just a tool. Use it if it's the right tool for the job.

To leverage ML, you need the right data

↳ People trained in ML can tell if the company has the right data

↳ No single answer for how much data is enough.

Never underestimate the value of a good heuristic.

With data comes great responsibility.

CASE STUDY: RECOMMENDATION SYSTEMS

There are no single way that these are built ∴ many core algorithms can be used here.

Most companies that use recommander systems have the same user problem to solve.

EX) Youtube recommendation system

↳ PRODUCT GOAL: Ensure every user has access to relevant, informative, and/or entertaining content w/ little friction as possible.

↳ Recommendation engine is just one part of the bigger system.

↳ It acts as another function: takes in data & returning recommendation.

↳ GENERALIZATION: recommending new relevant videos based on videos that similar users have historically found to be relevant.

THE ML TAXONOMY

DATA MATRIX: A structured table consisting of rows & columns. A collection of examples that are either labeled/unlabeled.

	Features			Label
	Merchant Distance	Transaction Amount	Merchant Code (Type)	Is Fraud
Examples	5	39.01	1	0
	2	112.81	1	1
	8	4.99	1	0
	6	1115.67	1	1
	0.5	97.96	4	1
	10.11	100	3	0
	4	1.15	1	0
	29.79	5.87	1	0
	45.47	15.93	2	1
	4	2500	1	0
	0.95	25.66	2	0
	21.33	2	5	0

FEATURES: Input variables. Each individual feature is an attribute that describes the data. (AKA. dimensions)

↳ Features are grouped together as a vector (FEATURE VECTOR) represented by X.

↳ Each individual feature in the vector is represented by: $x_1, x_2, x_3, \ldots, x_n$

LABELS: The attribute we want to predict (AKA. target variable OR output variable)

↳ Represented by y

EXAMPLES: instance of variable. It corresponds to a row in the data matrix. (AKA. datapoint)

↳ LABELED EXAMPLES: contains features [SUPERVISED]

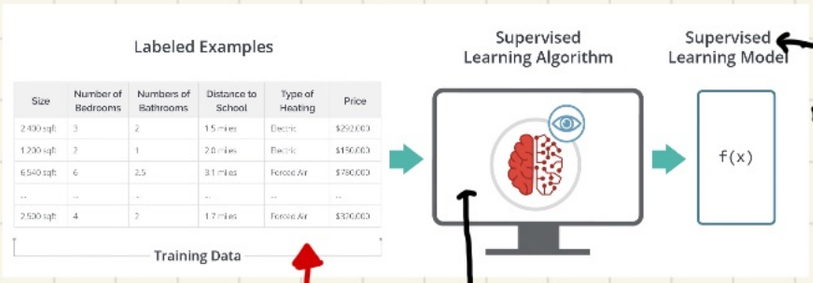
↳ UNLABELED EXAMPLES: contains only features NO labels. [UNSUPERVISED]

SUPERVISED LEARNING

Attempts to discover relationship between features & an associated label for the purpose of future predictions.

- Creates a program [model] that learns from past data to make predictions on similar new data.
- Learns to predict a label

TRAINING: Creating + learning the model.



← A predictive model is built & iteratively improved by using the label as means to check its correctness.

run on training data compromised of labeled examples

enabling the algorithm to gradually learn the relationship between features & labels.

Once training is complete, the result is an optimal model that can be used to make future predictions

MODEL:

A supervised learning model is the learned program that is able to make predictions on new, previously unseen data.

In supervised learning has 2 phases:

1. Training phase = model is built
2. Prediction phase / Inference phase = model is used to make predictions.

UNSUPERVISED LEARNING

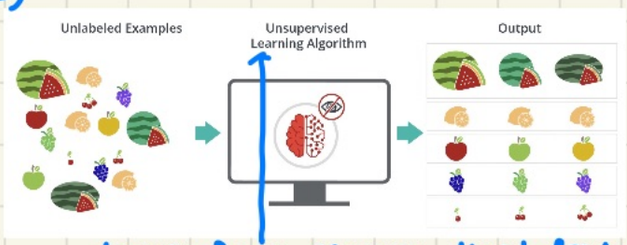
Discovering patterns in data containing labeled examples. It DOES NOT provide the answers (the label).

→ Discovers patterns on its own.
→ Uses unlabeled examples (feature values).
→ Algorithm discovers patterns in data typically by finding examples that are similar to each other & mapping them into different segments / clusters.

Ex) Provided diff images of diff animals. The algorithm determines pictures of humans from 2 legs, 2 arms, walks upright, etc.

CLUSTERING: Unsupervised learning technique that groups subsets of data that are collectively similar to one another based on the similarity of their feature values.

Ex)



Learns from images that fruits come in different sizes & shapes & colors. After learning their characteristics it can categorize the fruits

GENERALIZATION: A model's ability to adapt to new, unseen data that's ≅ to the data that has been trained on.

↳ NOTE: While it's important to make accurate predictions on training data, the ultimate goal is to create a model that can do well on future data

↳ MAKE SURE that our model GENERALIZES to NEW EXAMPLES!

SUPERVISED LEARNING ALGORITHMS:

① **CLASSIFICATION:** The process of recognizing, understanding, and grouping ideas & objects into preset categories or "subpopulations". Use input training data to predict the likelihood of future data that'll fall into predetermined categories.

↳ Labels are discrete or categorical values.

↳ With classifications there are multiple classifications:

↳ **BINARY CLASSIFICATION:** problems where the Q itself is yes/no, T/F, reject/approve.
- We can map each class to "+" or "−" OR 1/0
- EX) Spam mapped to "+" & non spam mapped to "−"

↳ **MULTICLASS CLASSIFICATION:** label we're predicting belongs to 3 or more distinct label values.
- EX) What animal is in given image — cat, dog, rabbit, or frogs

EX) Whether an email is spam or Not [BINARY CLASSIFICATION]

EX) What is the topic of given article [MULTICLASS CLASSIFICATION]

EX) Whether a CC transaction is fraudulent or not [BINARY CLASSIFICATION]

NOTE: When dealing w/ classification problems a label = class label

② **REGRESSION:**

The labels are continuous or any REAL NUMBER.

EX) What will the stock price be tomorrow?

EX) Housing price of new home in NYC.

EX) Minimum & maximum temperature on friday

EXAMPLE: SUPERVISED LEARNING PROBLEM

BUSINESS PROBLEM: We're a new credit lending start up. We know credit lending is a risky business ∴ we want to build a credit scoring system to make better lending decisions for people applying for credit.

⇓ [Translate into machine learning problem]

Build a default prediction model where default is defined as the customer missing 3 consecutive payments in the first 6 months.

⇓ ↳ NOTE: We're precise in defining our default. This will help us when we code it out.

Definition of default can be translated into T/F Q.

∴ with this label we have a binary classification problem.

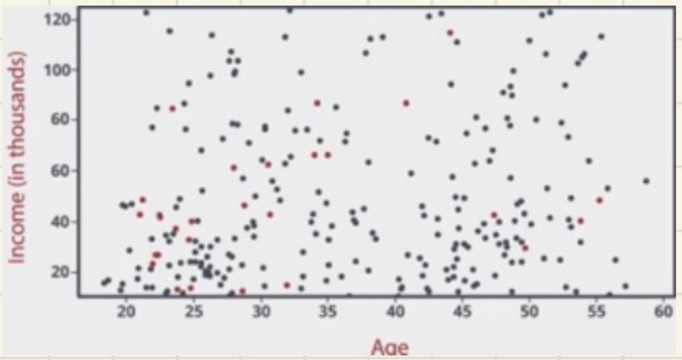
↳ To build our model, we'll need data

↳ If you have no data this is called COLD START PROBLEM. To combat this issue we start the experiment where we give credit to everyone who applies. When they do, we collect their AGE, INCOME as our main predictors.

We then observe for the first 6 months to determine if they're default or not.

Age	Income (in thousands)	y
25	21	0
26	32	0
24	23	0
30	55	0
27	11	0
20	46	0
24	13	1
26	29	0
26	19	0
20	17	0

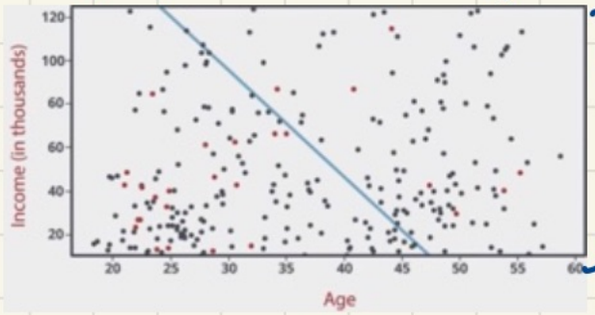
When we plot the data we can observe the relationship between age, income, & defaults:



NOTE: We plotted the default customers as RED vs. BLACK.

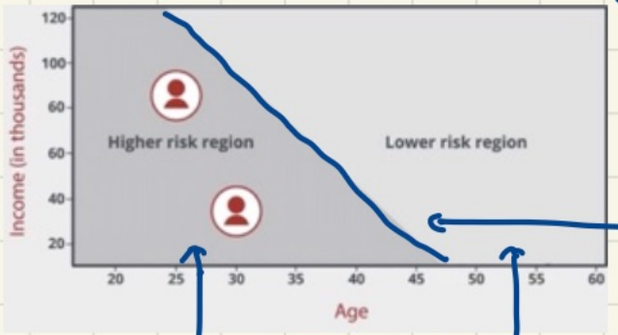
With ML or goal is to GENERALIZE a pattern from observed examples from the data. This means looking beyond the individual data points & trying to find regions on the graph that have higher density of defaulters.

AS AN EXERCISE: Draw a single line that'll best separate the RED from BLACK points.
ML helps us improve from lenders who would draw the lines using huevristic. However there are multiple problems: leaves a good value of people & discriminatory practices.



OPTIMAL Line using ML which helps us ultimately in reducing our error in predicting default incorrectly.

Once we have a model, we can ignore the original training data:



line
This represents our best generalization of the data.
We can use this to solve our business problem.

There is still a continuum between them.

- We can use the distance from the line to set appropriate credit limit.

EXAMPLE: UNSUPERVISED LEARNING PROBLEM

BUSINESS PROBLEM: Assume we have our credit model and are doing well using it to assess lending risk of our single product. But as our company evolves, we realize we're not serving the needs of all our potential customers with just one product.

⇓

NEW BUISNESS PROBLEM: Can we build different lending products that are tailored to specific needs of our different customers & target them at the time of application?

⇓ [TRANSLATE TO ML PROBLEM]

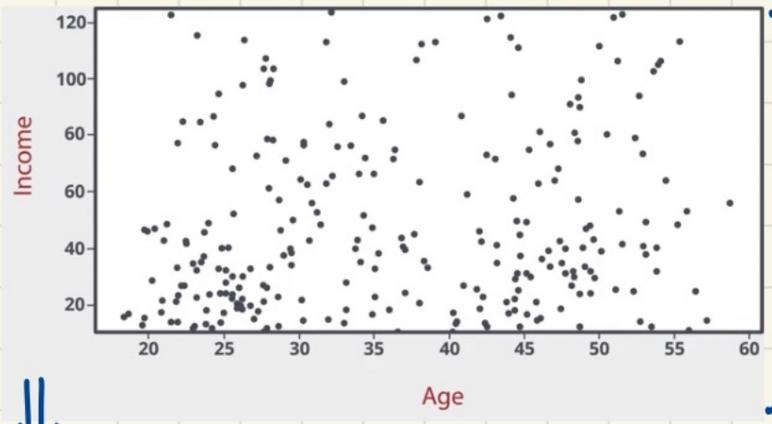
MACHINE LEARNING PROBLEM: Can we identify clusters of similar customers based on their observed attributes at application time.

> **NOTE:** No mention of specific outcome
> Keywords: "cluster" & "similar"
> } Indicates UNSUPERVISED LEARNING

GOAL: Match an applicant to the right product at time of application, we'll need to limit the data we use to what's available at application time.

⇓

Start with some basic visualization:



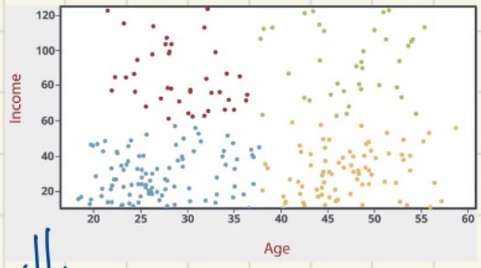
Scatterplot between AGE & INCOME.

We want to segment these points into discrete clusters where members of each cluster are ≅ to each other & ≇ to members outside of the cluster.

⇓

Draw 2 lines to split into 4 segments

∴ using our unsupervised learning clustering algorithms:



Now it's up to us to add meaning & utility to these clusters.

NOTE: In unsupervised we need to add subjective interpretation to derive full benefit.

↳ EX)

↗ Helps humanize segments + inform our product development process.

In many companies will conduct research/surveys in each segments to build strong human connection to them.

⇓

Design product to each specific needs.

THE ML LIFECYCLE

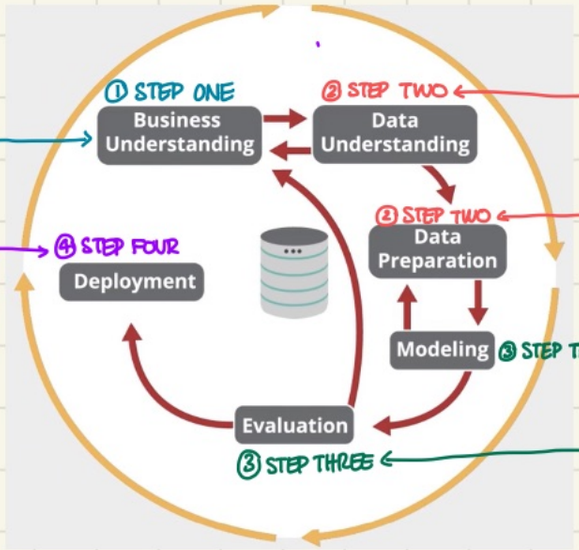
The core of machine learning is a set of computational algorithms driven by a mixture of mathematical techniques from calculus + linear algebra + information theory.

MACHINE LEARNING PROCESS : CRISP-DM

Start by understanding the problem in a business + domain + technical perspective.

When building models in practice we start w/ a written project brief where you document:
- ↳ motivation
- ↳ key constraints
- ↳ Planned approach + timeline

- Differentiates ML engineers from data scientists.
  - ↳ Deployments are left for ML engineers.



Determine if you have the right data.

- When we learn model collections we'll see how intertwined they are

---

SYSTEM RISKS: Applies to all software systems. It's the likelihood for complex & dynamic systems to have failure points.

↳ Up to ML engineers to implement the appropriate tests and monitors to be able to detect & mitigate such problems.

ETHICAL RISKS: The likelihood for large scale & automated decision systems to cause unintented harms.

↳ It's been shown that models that perform well on average can hurt non-majority & historically marginalized groups.

ML PROBLEM FORMULATION

First stage in ML development cycle is: **PROBLEM FORMULATION.**

Questions to consider when handling practice problems:

1. What problem is the model solving & why is it better than a non-ML based solution?
2. What kind of data would you need?
3. How would you solve this without ML?
4. What are the potential risks we should anticipate?

CASE STUDY: RECOMMENDATION SYSTEMS (revisited)

Youtube recommendation system:

Recommendation systems for any type of product typically start with a common data pattern. We call this **USER ITEM MATRIX**

	Item 1	Item 2	Item 3	Item 4	...	Item k
User 1	2	1			...	
User 2		2	4		...	2
User 3	3				...	
User 4	1	2	5	3	...	
User 5	3	2			...	
User 6					...	1
User 7		4	1		...	4
User 8		4	2		...	5
User 9	1				...	
User 10			3	4	...	1
...	...	...	...	...	...	
User N				1	...	4

ROWS = individual users/customers

COLUMNS = items

For our Youtube example:

ROWS = individual videos

← We have entries 1, 2, 3, 4, 5 that can tell us the ratings for how much it's consumed.

The values in the matrix can vary. But they should represent some level of consumption or engagment with specific items.

EX) Binary entries indicating if they viewed the video

EX) Numeric entries indicating ratings or level of consumption.

To make recommendations we can either use supervised/unsupervised approach.

SUPERVISED APPROACH

KEY: identify a good label

Label = entries of the matrix.

↳ If labels are binary ⇒ classification problem

↳ Labels are 1-5 ratings ∴ we can use regression.

- Features of the model will be whatever you have on both the user & the videos.
- For video recommendations we might use prior videos watched, descriptions, genre, video author as features.

UNSUPERVISED APPROACH

Use clustering algorithm & the key is to treat members of the same cluster as the group that'll help select & rank videos from a given user.

EX) When shopping online we have items tailored to show ≅ items to what we've been looking at. This type of recommendation is driven by ITEM BASED CLUSTERING

HYBRID APPROACH

There are often too many items & it would be too $ to run in a brute force way. ∴ we have the hybrid approach

We can use the resulting clusters to filter the billions of video ↓ to just a few thousand. Then we can use supervised model to score the smaller set.

NOTE: Hybrid is used to gain better accuracy of the supervised learning approach w/ the scalability of the unsupervised

THE ML TECH STACK

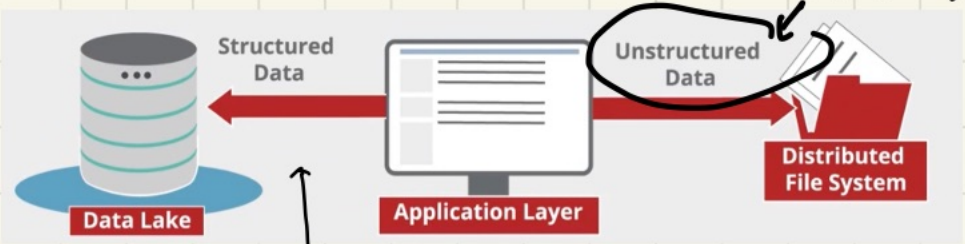
Before building data we need to ask ourselves the following questions:

1. Where do I find my data?
2. Where do I actually build the model?
3. How do I access these different systems?

As a ML engineer you'll most commonly be accessing different data sets in DATA LAKE. Data is sent to data lake through logging processes & SQL updates to databases from application layer.

A general software controlling the entire system.

easier to manage + maintain



Data sent to a database is typically structured meaning it has a well defined schema or format that's the same for every record

Once you know where your data lives we have to study its format. You'll need to access it for analysis, exploration, & modeling.

↳ Usually done in IDE ⟸ main point for accessing data ⟹ Juypter Notebook

---

ML WORKFLOW APPLICATIONS

Data is often stored in Linux-based server.

To access the data directory, you have to use the terminal, navigate the data using typical command line tools.

IMPORTANT: Paths are very important!

Ex) Assume you have a folder in your desktop named myFolder and creat a file in the folder named myFile.txt. The path is:

⇓

"/Users/myUserName/Desktop/myFolder/myFile.txt"