I’m constantly fascinated by machine learning and always excited to find new projects for it. But as trendy as ML has become, sometimes a SQL query or IF statement can accomplish the same job as an ML model in much less time. #

Lots of ML projects fail

- Commonly quoted statistic: 87%

why?

- ML is still research -shouldn't aim for 100% success
- But, many are doomed to fail
  -Technically infeasible or poorly scoped
  -Never make the leap to prod
  -Unclear success criteria
  -works, but doesn't solve a big enough problem to be worth the complexity.
- Erodes the boundaries between systems
- Relies on expensive data dependencies
- commonly plagued by system design anti-patterns
- subject to the instability of the external world

before starting an ML project, ask yourself:

- Are we ready to use ML?
  -Do we have a product?
  -Are we collecting data and storing it in a same way?
  -Do we have the right team?
- Do we really need ML to solve this problem?
  -Do we need to solve the problem?
- Is it ethical?

How to pick problems to solve with ML

High impact, low-cost

- High impact problems are likely to be those that address
  -Friction in your product
  -Complex parts of your pipeline

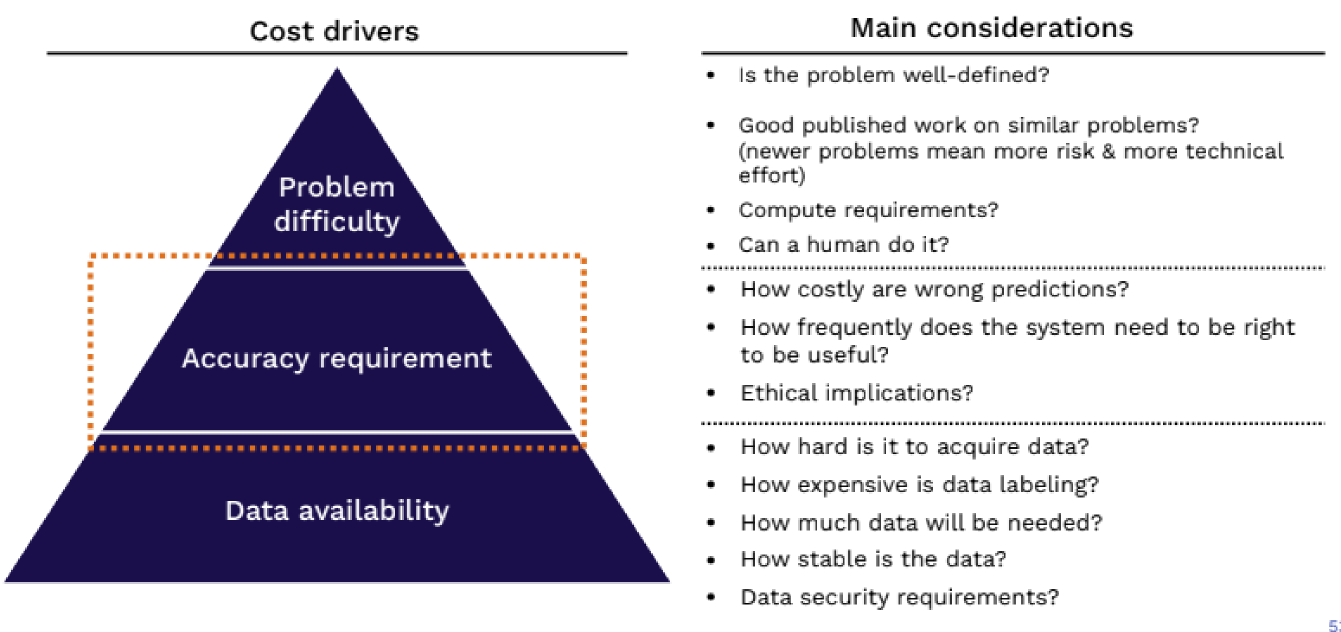
-Places where cheap prediction is valuable
-what other people in your industry are doing

- Low-cost projects are those with data available, and where bad predictions aren't too harmful

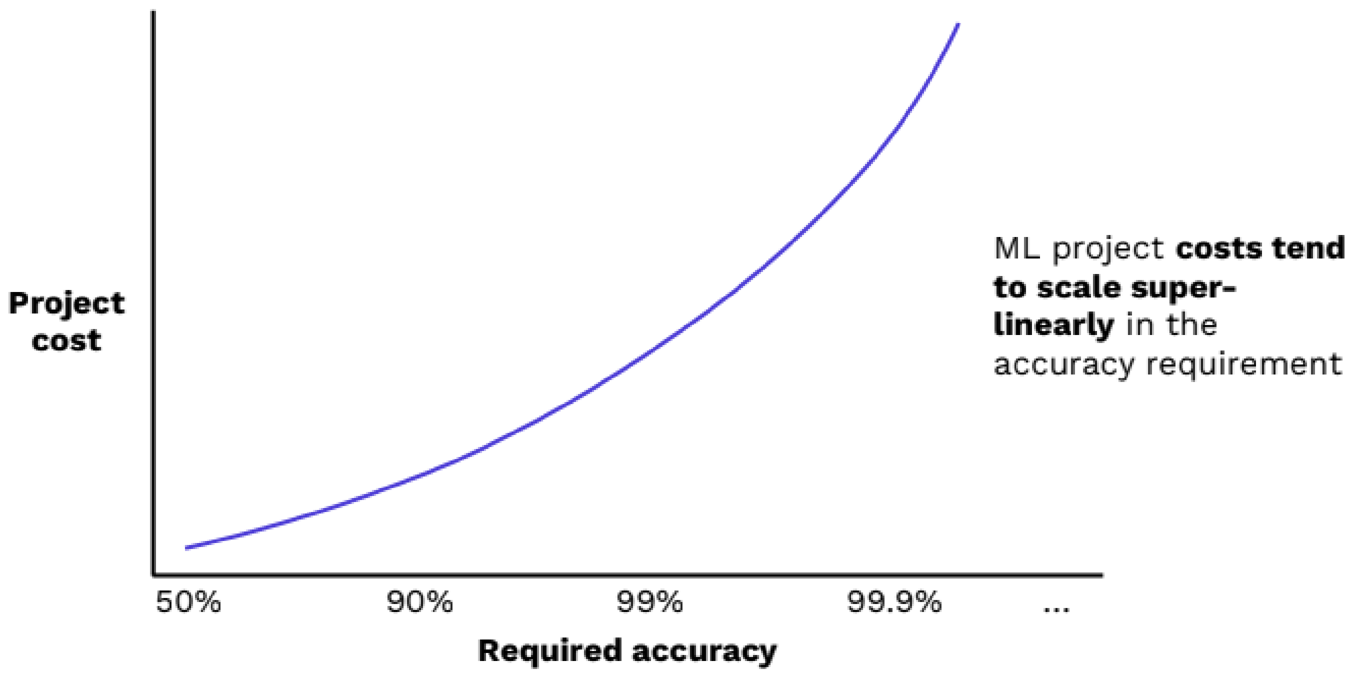
what does ML make economically feasible?

- AI reduces cost of prediction
- Prediction is central for decision making
- cheap prediction means
  -Prediction will be everywhere
  -Even in problems where it was too expensive before
- **Implication**:Look for projects where cheap prediction will have a huge business impact.

Assessing feasibility of ML projects



Why are accuracy requirements so important?



Lifecycle of a ML project