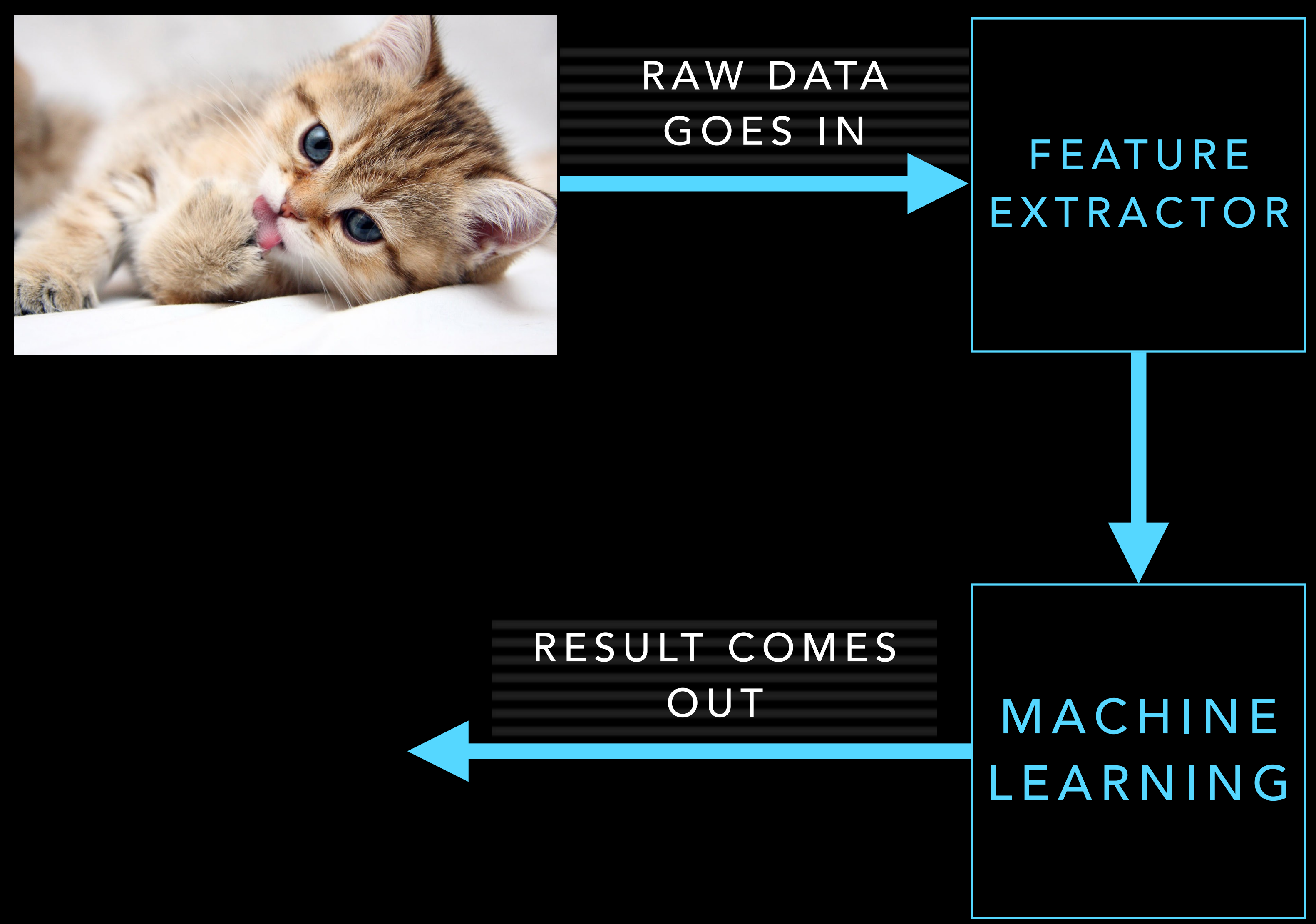
Many applications involving machine learning take the following form:



NUMERIC FEATURES

- We can encode multiple numeric measurements in vector form: [m1, m2, m3, …]
    - If the measurements come from different domains (e.g. windspeed (m/s), power output (kW), etc) we need to be careful to standardise the measurements
        - A difference of 1m/s is very different to a difference of 1kW!

BINARY FEATURES

- For binary features we can build binary vectors [0,1,1,0,…]
- Each element corresponds to something that can be measured in the original data (and is either present or absent)

CATEGORICAL FEATURES

- How can we encode features that can take one value from a set of N>=2 choices?
  - We could choose to represent each value with a numeric index {A->0, B->2, C->3,…}
  - But this is perhaps not the best option unless we believe that A is closer to B than A is to C

- Solution: One Hot Encoding
    - Map each value to a vector in which the values are all 0, except for a single element with a 1 in the position that represents that value.
    - For example, taking the mapping {A->0, B->1, C->2}, the value “B” can be encoded as [0,1,0]
    - All encodings are equally as distant from each other
        - We can think of this encoding as an idealised probability distribution when the possible values are independent of each other

ASIDE: SIMILARITY

- What about if we have prior knowledge about similarities?
    - For example, “Dalmatian” is closer to “Poodle” than it is to “Horse”
    - There are alternative embedding strategies that allow this to be exploited….