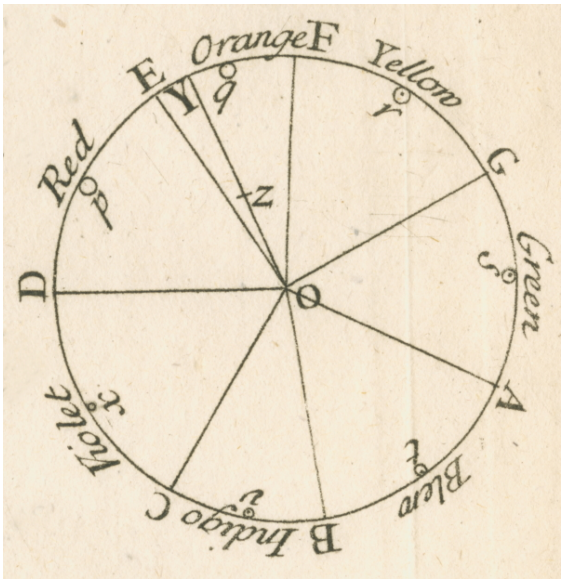
# Overview of machine learning

COMS 4771 Fall 2025

Problem: Create a program that, given a pair of input colors (e.g., blue, yellow), returns the color observed when the input colors are mixed together.



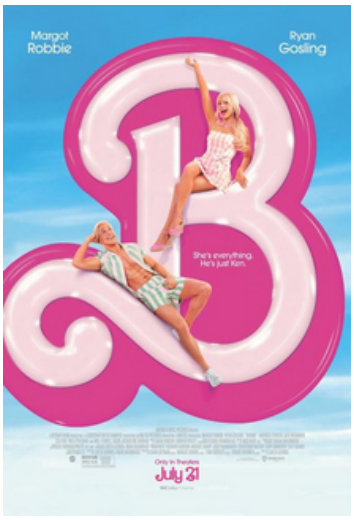

(blue, yellow) $\mapsto$ green

Problem: Create a program that, given an image of a bird, returns the name of the bird species

"Peregrine falcon"

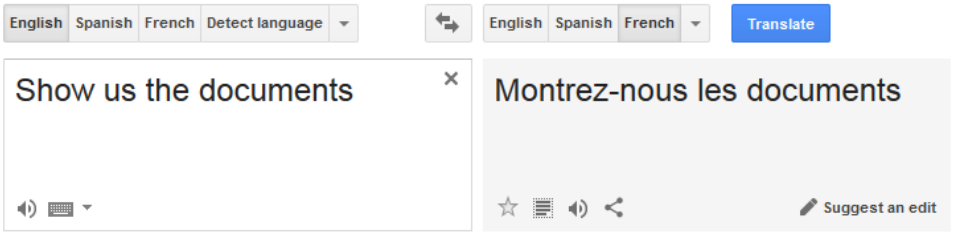


Problem: Create a program that, given a Netflix user and a movie, returns the rating that the user would give to the movie




****

Problem: Create a program that, given a sentence written in English, returns the French translation






Problem: Create a program that, given a list of numbers, returns a list of the same numbers in sorted order

$$(29, 31, 67, 43, 21, 30, 96) \longrightarrow (21, 29, 30, 31, 43, 67, 96)$$

Problem: Create a program that, **given no input at all**, returns the outcome of a fair coin toss **that you will perform after the program completes**.

"heads"



- Which of these problems could you solve "by hand"?

- Which of these problems could be solved at all?

**Basic pipeline for (supervised) machine learning**

- Step 1. Collect training data—collection of input/output pairs
- Step 2. Decide how to represent inputs (feature engineering) and the general "template" of the program (model selection)
- Step 3. Learning algorithm "fills-in" the template using data
- Step 4. Get additional test data—more input/output pairs
- Step 5. Evaluate program on the test data



Feature engineering: how will the input be provided to the program?

- Netflix movie rating:
- Spam filtering:

Model selection: what is the general "template" of the program?

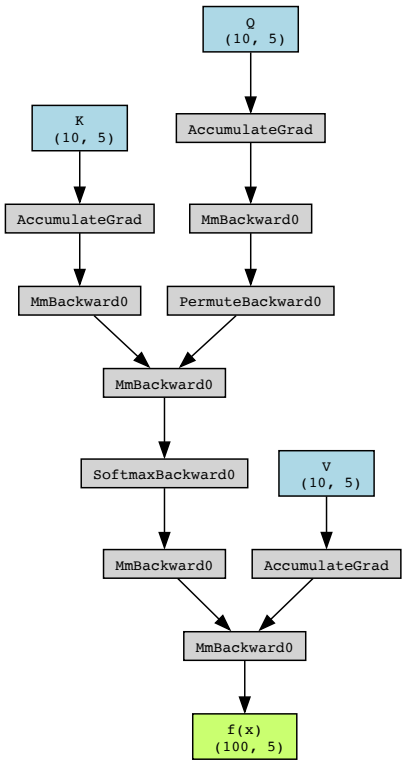
**Require:** age, gender, genre, release year, title

```
if age ≥ 40 then
    if genre = war or genre = western then
        return 4.3
    else
        return 1.5
    end if
else
    if release year > 1998 then
        return 3.5
    else
        return 2.0
    end if
end if
```



$$\Pr(\text{spam} \mid \text{email features} = x) = \frac{1}{1 + \exp(-(w_1 x_1 + w_2 x_2 + b))}$$

```python
def f(x):
  k = x @ K
  q = x @ Q
  a = torch.softmax(k @ q.T, dim=1)
  return a @ x @ V
```






## This course

- Statistical framework for supervised machine learning
  - Basic prediction theory
  - Evaluation criteria (e.g., risk, calibration, bias)
- Algorithmic paradigms for supervised machine learning
  - Memorization and space partitioning
  - Deriving algorithms based on statistical models
  - Numerical optimization
- Some modeling techniques
  - Statistical models
  - Feature maps, kernels, neural networks
  - Regularization

More machine learning

- **Supervised learning**: data available is input/output pairs
- **Unsupervised learning**: do something with just the inputs
- **Online learning**: continuously make predictions over time
- **Reinforcement learning**: make decisions in reactive environment



Beyond machine learning

- **Statistical inference**: quantify uncertainty, determine causality, etc.
- **Artificial intelligence**: efficient knowledge representation and reasoning
- **Learning theory**: mathematical models and stat/comp. limits of learning
- ...