Project 1

COVID-19 Cases Prediction

Objectives

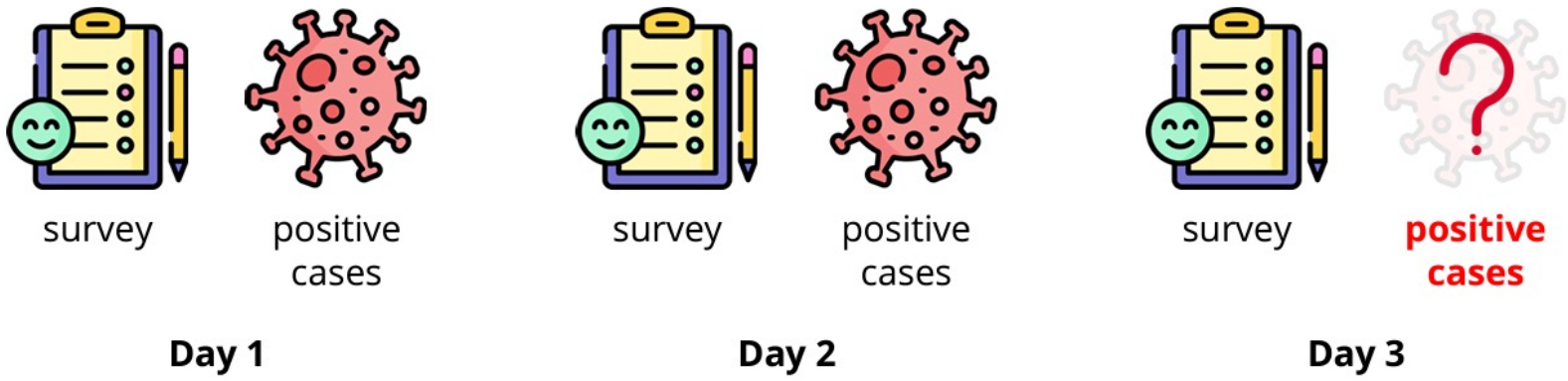
- Solve a regression problem with deep neural networks (DNN).
- Understand basic DNN training tips e.g. hyper-parameter tuning, feature selection, regularization, …
- Get familiar with PyTorch.
- **Due date: Thursday Sep 4, 2025 (23:59)**

Requirement

- Python 3
- **Pytorch:**
  - Documentation: https://pytorch.org/docs/stable/index.html
  - Tutorial: https://www.dataquest.io/blog/pytorch-for-beginners/

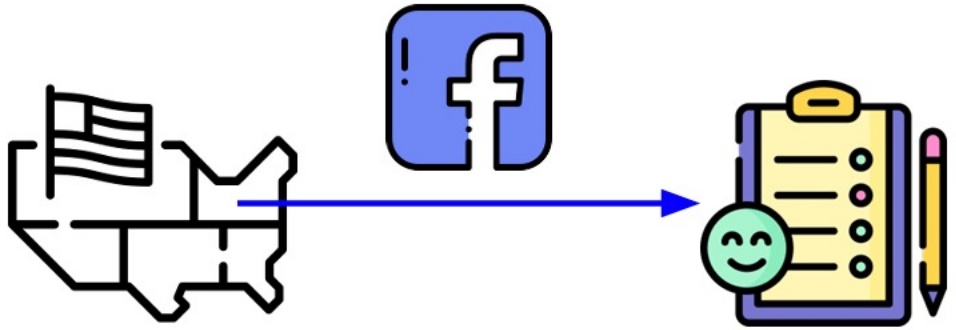
Task Description

- Given survey results in the past 3 days in a specific state in U.S., then predict the percentage of new tested positive cases in the 3rd day.

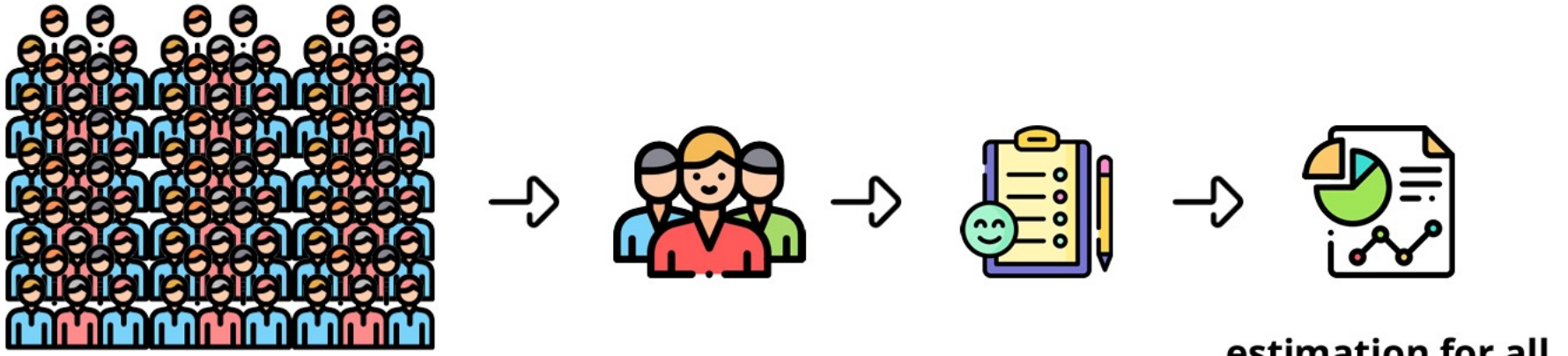


Data -- Delphi's COVID-19 Surveys

- Conducted surveys via facebook (every day & every state)
- Survey: symptoms, COVID-19 testing, social distancing, mental health, demographics, economic effects, ...



Data -- Delphi's COVID-19 Surveys



All population in a certain state of the U.S.

some samples

survey

estimation for all population in that state (data we are using)

Data -- Delphi's COVID-19 Surveys

- **States** (40, encoded to **one-hot** vectors)
  - e.g. `AL, AK, AZ, ...`
- **COVID-like illness** (4)
  - e.g. `cli,ili` (influenza-like illness), ...
- **Behavior Indicators** (8)
  - e.g. `wearing_mask, travel_outside_state, ...`
- **Mental Health Indicators** (5)
  - e.g. `anxious, depressed, ...`
- **Tested Positive Cases** (1)
  - **`tested_positive`** (this is what we want to predict)

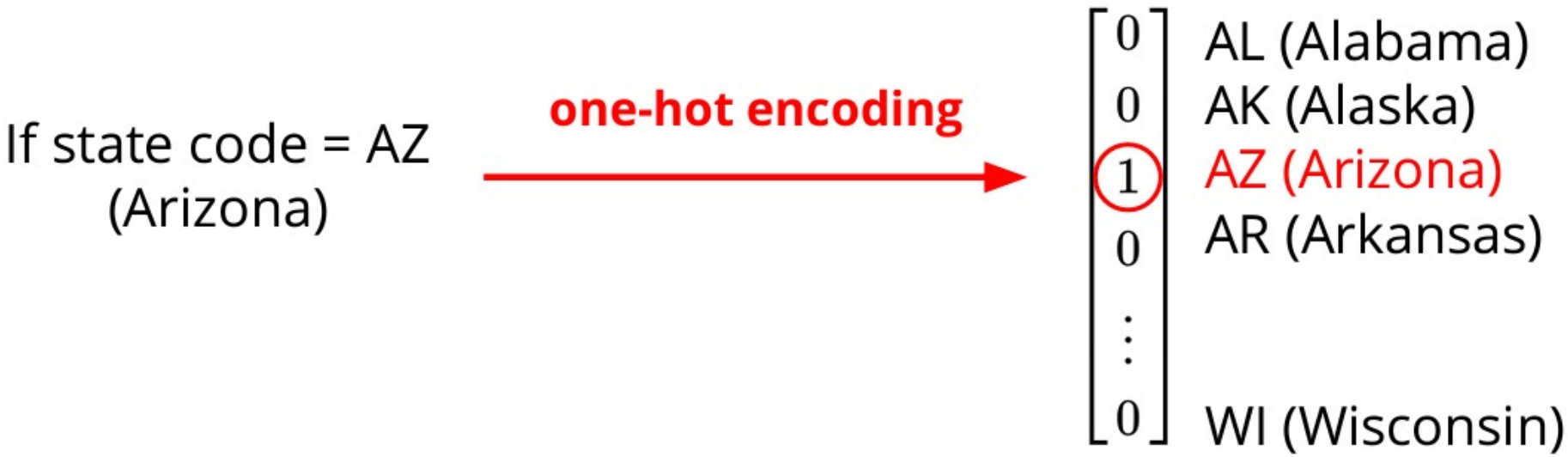
Percentage

Data -- One-hot Vector

- **One-hot vectors**:

  Vectors with **only one element equals to one** while others are zero.

  Usually used to encode discrete values.



Data -- Training

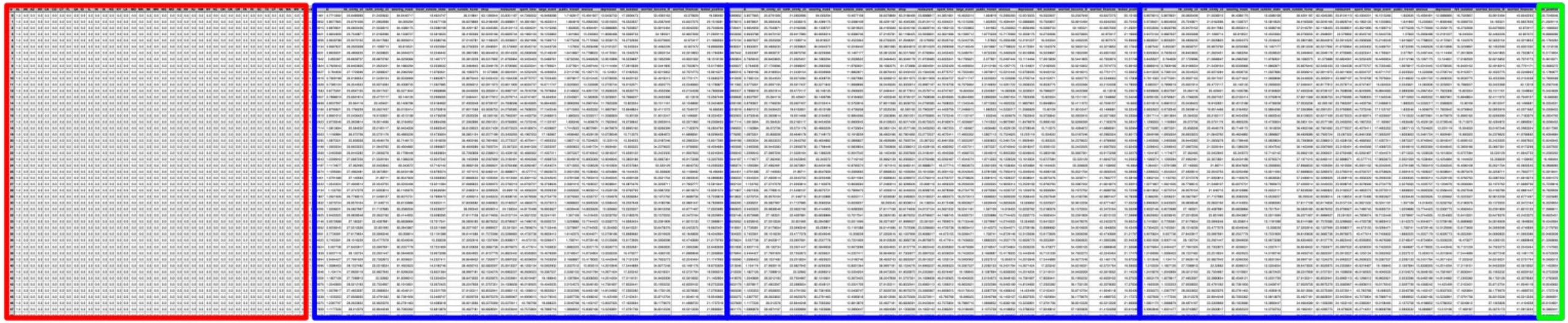
`covid.train.csv` (2700 samples)

state one-hot encoding (40)

Day 1 features (18)

Day 2 features (18)

Day 3 features (18)



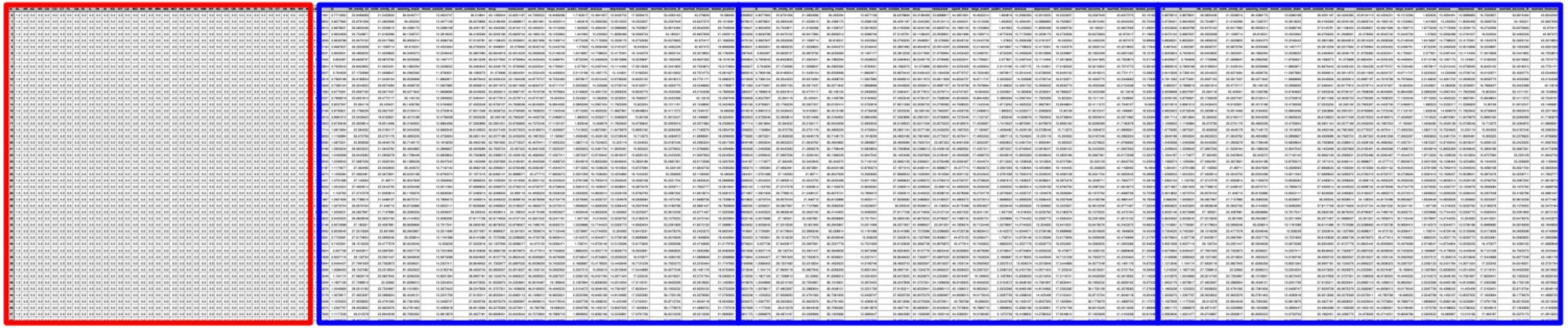
tested positive

1 row = 1 sample

Data -- Testing

`covid.test.csv` (893 samples)

state one-hot encoding (40) | **Day 1 features (18)** | **Day 2 features (18)** | **Day 3 features (17)**



1 row = 1 sample

Evaluation Metric

- **Root Mean Squared Error (RMSE)**

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(f(\mathbf{x}^n) - \hat{y}^n\right)^2}$$

your model

input features
(testing data)

ground truth label
(correct answer)

Hints

- *Feature selection* (what other features are useful?)
- *DNN architecture* (layers? dimension? activation function?)
- *Training* (mini-batch? optimizer? learning rate?)
- *L2 regularization*

Deliverable

- PDF Report (40%)
  - methodology (e.g., network structure, training tips, hyperparameters, etc.)
  - empirical results and evaluation
  - conclusion
- Python Code (60%)
  - Code is required to avoid plagiarism.
- Grading:
  - Total (100):
    - Code (60) + Report (40)