Foundation of Machine Learning

Day 3: Machine Learning Deployment

Sieu Tran

Agenda

Day 3 of the Foundation of Machine Learning course will focus on ML deployment

1. Recap of Day 1 and Day 2
2. Introduction to ML Deployment
3. Continuation of Day 1 Coding Assignment
4. Continuation of Day 2 Coding Assignment
5. Group Discussion
6. Classification Problems
7. Day 3 Coding Assignment
8. Closing Words

Recap

Day 1 and Day 2 Coding Assignments Revisit

Working Session 1

Case study:
- United States COVID-19 Cases and Deaths by State over Time

Data:
- Source: https://catalog.data.gov/dataset/united-states-covid-19-cases-and-deaths-by-state-over-time
- Descriptions: https://data.cdc.gov/Case-Surveillance/United-States-COVID-19-Cases-and-Deaths-by-State-o/9mfq-cb36/data

Problems:
- Perform some exploratory analysis:
  - Which states has the most deaths?
  - Which month has the most deaths?
  - Are there any correlations?
  - What does the rate of death look like over time?
- Can you perform dimensional reduction on this dataset?
- Can you predict the rate of death in the upcoming month?
- Consider setting threshold for high/medium/low death rate for classification problem
- Example hypotheses:
  - States near the sea are more effected
  - States that are politically conservative have higher death rates

Get started:
- Python users:
  - Install Anaconda
  - Install Python 3.7
  - Create a virtual environment:
  - *conda create -n yourenvname python=x.x anaconda*
  - Install requirements: *pandas*, *numpy*, and *sklearn* to start

  OR
  - Use Google Collab

- R users:
  - Install RStudios
  - Install requirements: *tidyverse* and *gbm* to start

Working Session 2

Case study:
- Facebook ad and consumer behavior

Data:
- Source: https://www.kaggle.com/datasets/loveall/clicks-conversion-tracking

Problems:
- Read in the data and perform some exploratory analysis
- Create a new column for rate of click (Click/Impression).
- Build a predictive (regression) model to predict the rate of click using a selection of other features
    - Can you perform the train/test/validation split?
    - Can you compute the three/four types of errors we discussed today?
    - Can you look at the residual plots and diagnose the problems your model might be facing?
    - How can you make the model better?

Get started:
- Python users:
    - Install Anaconda
    - Install Python 3.7
    - Create a virtual environment:
    - *conda create -n yourenvname python=x.x anaconda*
    - Install requirements: *pandas*, *numpy*, and *sklearn* to start

  OR
    - Use Google Collab

- R users:
    - Install RStudios
    - Install requirements: *tidyverse* and *gbm* to start

Large Scale Machine Learning

Map-reduce And Data Parallelism

Many learning algorithms can be expressed as computing sums of functions over the training set.

Batch gradient descent

$$\theta_j := \theta_j - \alpha \frac{1}{500} \sum_{i=1}^{500} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

Task Divided

- Machine #1 ...
- Machine #2 ...
- Machine #3 ...
- Machine #4 ...
- Machine #5 ...

Parallel Computing

- Use $\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(100)}, y^{(100)}\right)$
  - $\text{Temp}_j^{(1)} = \sum_{i=1}^{100} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$
- Use $\left(x^{(101)}, y^{(101)}\right), \ldots, \left(x^{(200)}, y^{(200)}\right)$
  - $\text{Temp}_j^{(2)} = \sum_{i=101}^{200} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$
- Use $\left(x^{(201)}, y^{(201)}\right), \ldots, \left(x^{(300)}, y^{(300)}\right)$
  - $\text{Temp}_j^{(3)} = \sum_{i=201}^{300} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$
- Use $\left(x^{(301)}, y^{(301)}\right), \ldots, \left(x^{(400)}, y^{(400)}\right)$
  - $\text{Temp}_j^{(4)} = \sum_{i=301}^{400} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$
- Use $\left(x^{(401)}, y^{(401)}\right), \ldots, \left(x^{(500)}, y^{(500)}\right)$
  - $\text{Temp}_j^{(5)} = \sum_{i=401}^{500} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$

Combine results

$$\theta_j := \theta_j - \alpha \frac{1}{500} \left(\text{Temp}_j^{(1)} + \text{Temp}_j^{(2)} + \text{Temp}_j^{(3)} + \text{Temp}_j^{(4)} + \text{Temp}_j^{(5)}\right)$$

$$(j = 0, \ldots, n)$$

ML Deployment: Docker Example

Example Deployment With Docker

	Phase 1	Phase 2	Phase 3	Phase 4
	Preparation	**Creation**	**Execution**	**Testing**
Specific Steps	**Create Python Program**	**Create Docker File And Image**	**Run Docker Image in Container**	**Test The Python Program Running Within A Container**
Detailed Tasks	• Use your Python IDE to create the API script (e.g., using FastAPI, Flask, etc.) • Create a new Python project • Create a *requirements.txt* file to outline necessary packages and versions • Create a new file: *main.py* in the root folder that outlines the functionality of the application	• Open and edit the file named Dockerfile in the root folder • Build using: *docker build --tag myapp-python-docker*. • This will build the docker image.	• Run with *docker run --publish 5000:5000 myapp-python-docker.*	• Navigate to the URL, e.g., *http://localhost:5000* • The app should return the same results as the Python scripts before. • The main difference is that the same Python web server is now running within a docker container. • Multiple docker containers may be launched on different machines and even on the same machine by overriding the port number.

It’s Coding Time!

Working Session 2

Case study:
- Deploy the models you have built from Day 1 and Day 2

Data:
- Source 1: https://catalog.data.gov/dataset/united-states-covid-19-cases-and-deaths-by-state-over-time
- Source 2: https://www.kaggle.com/datasets/loveall/clicks-conversion-tracking

Problems:
- Can you write them into functions instead of script?
  - Input: data
  - Output: predictions
- Can you clean up the code?
- Can you create the requirements.txt file?
- Can you check each package's versions?
- Can you collect the licenses for each packages and versions?

Get started:
- Python users:
  - Install Anaconda
  - Install Python 3.7
  - Create a virtual environment:
  - *conda create -n yourenvname python=x.x anaconda*
  - Install requirements: *pandas*, *numpy*, and *sklearn* to start

  OR
  - Use Google Collab

- R users:
  - Install RStudios
  - Install requirements: *tidyverse* and *gbm* to start

Closing Notes

Referenced Materials

1. Ng, Andrew. ”Machine Learning By Prof. Andrew Ng”, (2023), GitHub repository, https://github.com/vkosuri/CourseraMachineLearning.
2. Malik, Farhad. “Running Python In Docker Container”. Medium, 03 Oct 2021, https://medium.com/fintechexplained/running-python-in-docker-container-58cda726d574.

