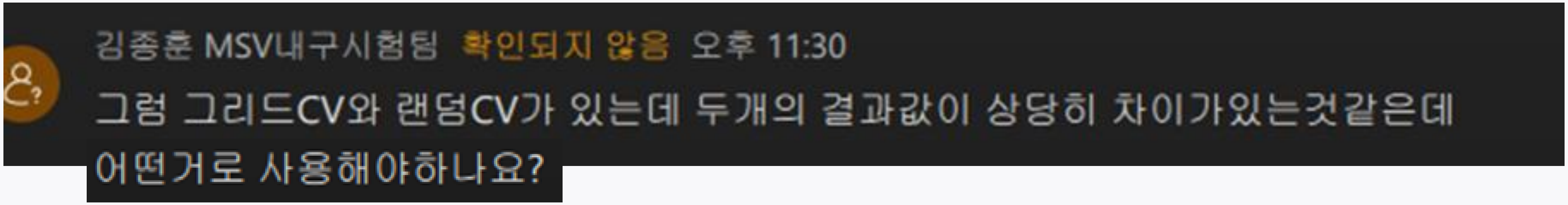
Mini-project overview

Lecture 16

Changho Suh

January 29, 2024

Remaining questions from the last lecture



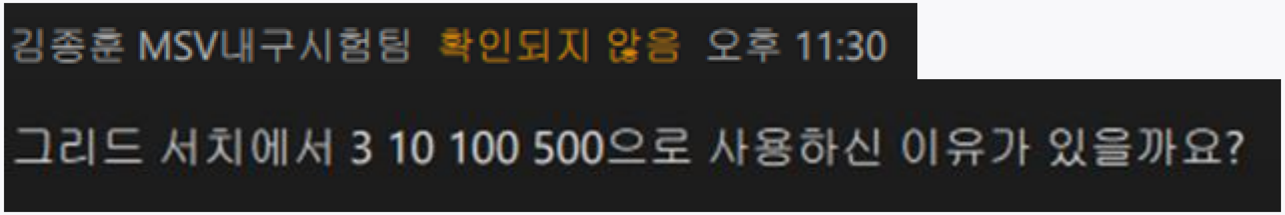
김종훈 MSV내구시험팀 확인되지 않음 오후 11:30

그럼 그리드CV와 랜덤CV가 있는데 두개의 결과값이 상당히 차이가있는것같은데 어떤거로 사용해야하나요?

처음: RandomizedSearchCV를 통해 성능이 좋은 영역을 rough하게 search

나중: 해당영역에서 GridSearchCV를 통해 fine-search

Remaining questions from the last lecture

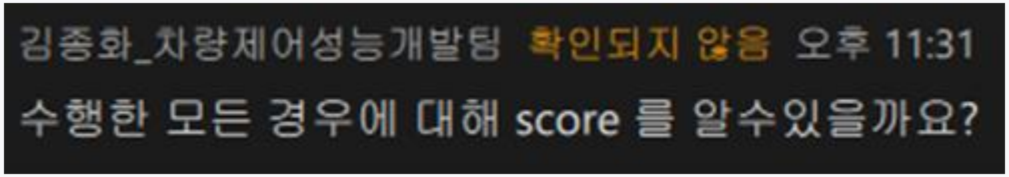


김종훈 MSV내구시험팀 확인되지 않음 오후 11:30

그리드 서치에서 3 10 100 500으로 사용하신 이유가 있을까요?

Default값=100 을 기준으로 잡은 한 가지 예시일 뿐

Remaining questions from the last lecture



김종화_차량제어성능개발팀 확인되지 않음 오후 11:31

수행한 모든 경우에 대해 score 를 알수있을까요?

알 수 있음. 오늘 오후 세션시 보는 방법을 학습.

Plan for Week 2

1. Mini-project # 1 (Day 6):

   Learn how to: architect a machine learning project;
   do data processing (via pandas);
   improve a model.

2. Mini-project # 2 (Day 7):

   Learn how to: do data processing for RNN;
   improve a model.

3. Proposal (Days 8,9,10):

   Learn how to write a proposal.

   Write your own proposal and do rehearsal.

   Deliver final presentation.

Mini-project #1

(Day 6)

Guideline for a machine learning project

Four steps illustrated via:

“**STAR**” method

Will explain the STAR method while describing the contents of mini-project #1.

1. Situation

Describe the **s**ituation:

1. Project context (배경)

2. Challenge (도전적 과제)

1. Situation: Project context

Mini-project #1 is about:

Vehicle manufacturing

There are many car options.

The vehicle test time varies significantly across different options.

1. Situation: Challenge

The challenge that we will address is about:

Time taken for testing a vehicle

Some car options require long test time.

→ Incurs significant cost and environmental issues.

Hence: Crucial to figure out car options taking short test time.

2. Task: Come up with an ML task

Task: Predict vehicle test time

Will consider an ML task that Mercedes-Benz took into account:

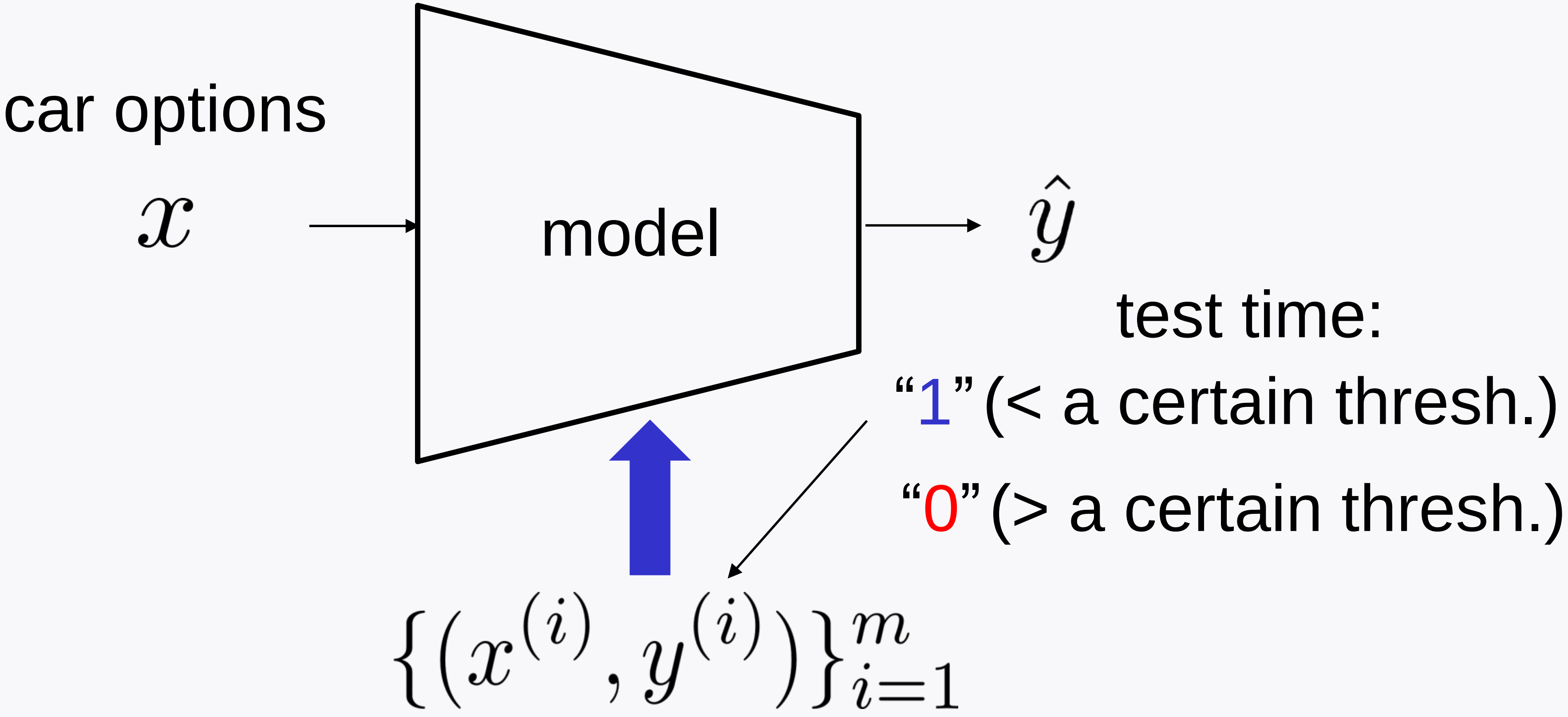
A **binary classifier:**

Outputs 1 if test time < a certain threshold

0 if test time > a certain threshold

2. Task: ML model

Test-time prediction:



car options

x

model

$\hat{y}$

test time:

“1” (< a certain thresh.)

“0” (> a certain thresh.)

$$\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$$

3. Activity: Describe activities

3.1. Dataset:

What is *m*? What are (*x*, *y*) and their size?

How to collect if not prepared yet?

3.2. Model:

Choose a model (LS, LR, DNN, CNN, RNN, RF etc)

Explain the rationale behind the choice.

3.3. Target performance:

Accuracy (classification)   RMSE (regression)

Explain the rationale behind the choice using domain knowledge

3.1. Activity: Dataset

Source: Mercedes-Benz provides **4209** examples

Data: anonymized **376** car options

Label: test time

Raw data is in **csv** file.

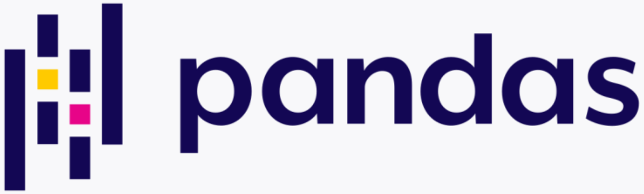
mercedes_test.csv

x_1	…	x_{376}	y
K	…	at	130.81
K	…	av	88.53
az	…	n	76.26

Will learn how to **load data from csv file**

3.1. Activity: How to load MB dataset

To this end, will employ:



3.1. Activity: Pre-processing

pandas for data manipulation & analysis

x_1	…	y
K	…	130.81
K	…	88.53

x1_aa	…	x1_K	…	x1_z	…	y
0	…	1	…	0	…	0
0	…	1	…	0	…	1

1. One hot encoding of categorical data $\in \{aa, \ldots, z\}$

2. Binary label w/ a median threshold

3.1. Activity: Data frame → numpy array

NumPy

for large, multi-dimensional arrays and matrices

x1_aa	…	x1_K	…	y
0	…	1	…	0
0	…	1	…	1

$$X = \begin{pmatrix} 0 & \cdots & 1 & \cdots \\ 0 & \cdots & 1 & \cdots \end{pmatrix}$$

$$y = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

3.1. Activity: Train/val/test sets

# of examples: **4209**

Recall: It belongs to the middle range:

$$1,000 \leq m \leq 10,000$$

Hence: Will take the following split ratio:

8:1:1

3.2. Activity: Model selection

Will try three models for exercise purpose:

LS, **LR** and **DNN**

3.3. Activity: Target performance

LS: ~89% accuracy (w/ regularization)

LR: ~89% (w/ regularization)

3.3. Activity: Target performance

DNN: ~89% (w/ some techniques)

Techniques that we will apply:

- Regularization, early stopping
- He’s initialization
- Hyperparameter search w/ cross validation:
  - # of layers, learning rate, …

4. Result: Describe results (impacts)

Explain what your ML model can do:

1. Quantitatively: In terms of numbers (money)

2. Qualitatively: In many other aspects

Three more things to learn from MP1

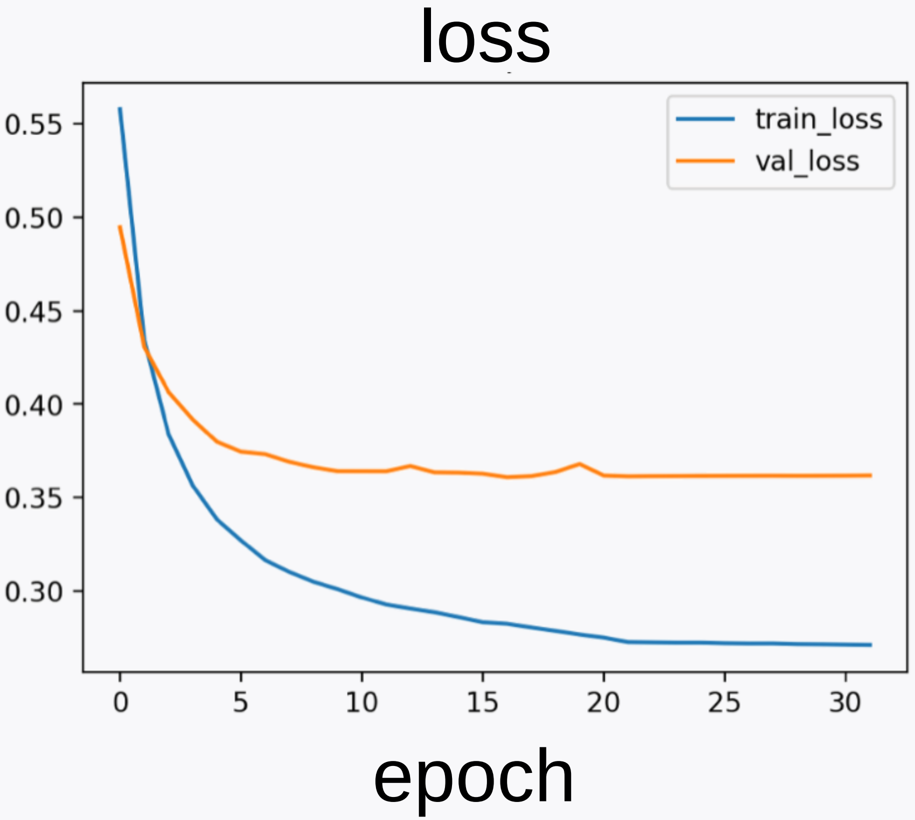
1. Plotting

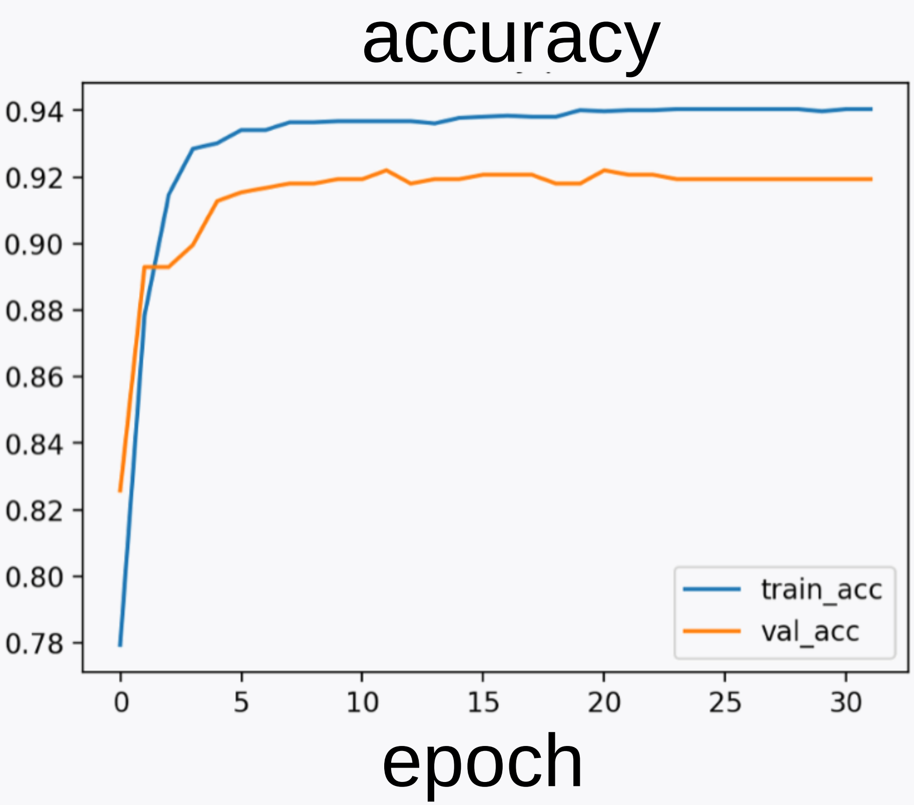
2. Logging

   Track intermediate results

3. Saving

Plotting


loss
train_loss
val_loss
0.55
0.50
0.45
0.40
0.35
0.30
0
5
10
15
20
25
30
epoch


accuracy
0.94
0.92
0.90
0.88
0.86
0.84
0.82
0.80
0.78
train_acc
val_acc
0
5
10
15
20
25
30
epoch

Logging: Track intermediate results

Example:

(LR) regularization_factor=0.9 → acc: 89%

(DNN) learning rate: 0.01, batch size: 512 → acc: 91%

To this end, will use **pandas**.

Saving

Save parameters of trained models.

Sklearn models:

```
from joblib import dump
```

Keras models:

```
model.save()
```

Look ahead

Figure out how mini-project #2 is organized.