M3239.005300 Machine Learning & Deep Learning I

Spring 2024

Lab 1

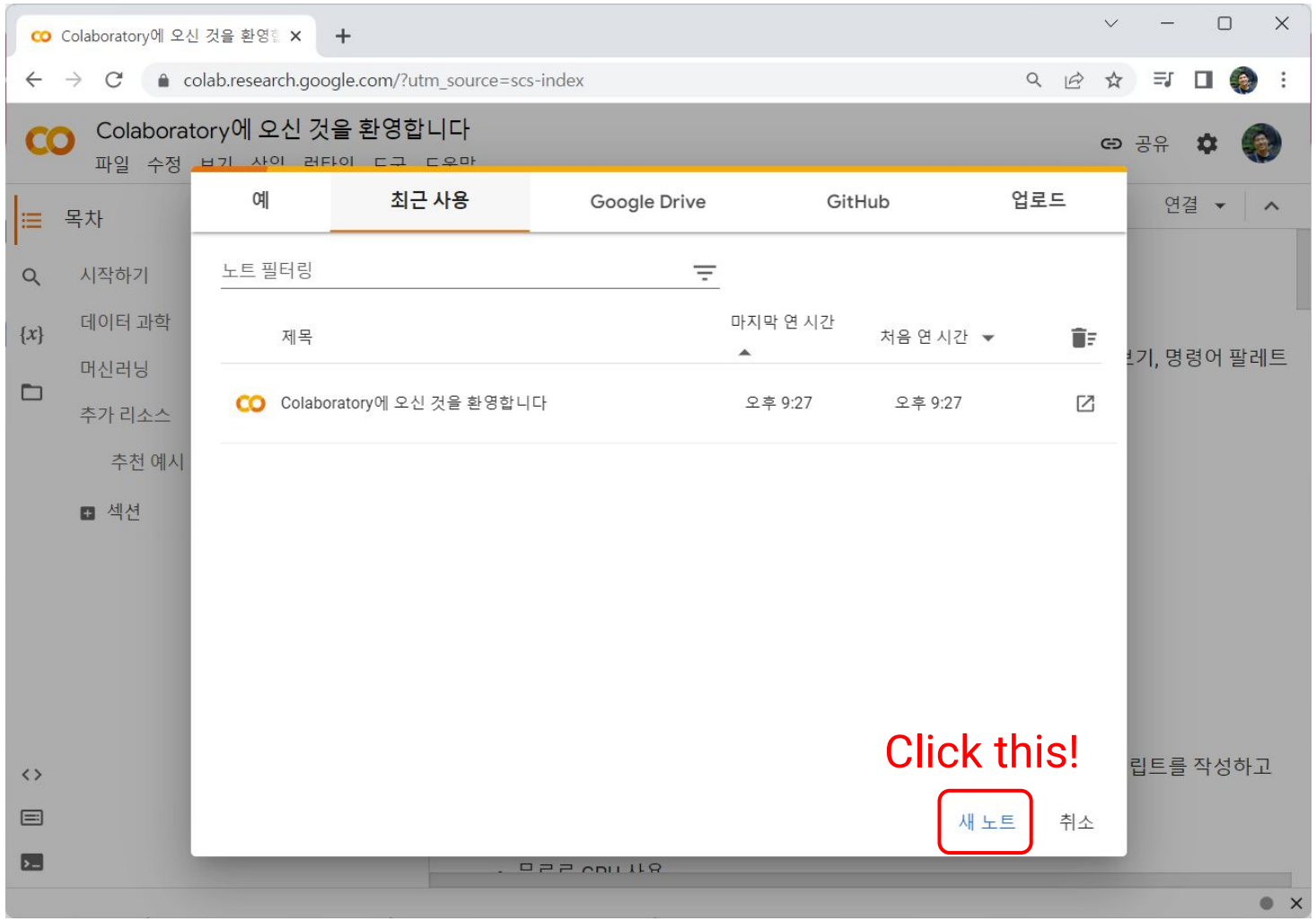
Yujin Jeon

Seoul National University
Graduate School of Data Science

Google Colab Tutorial

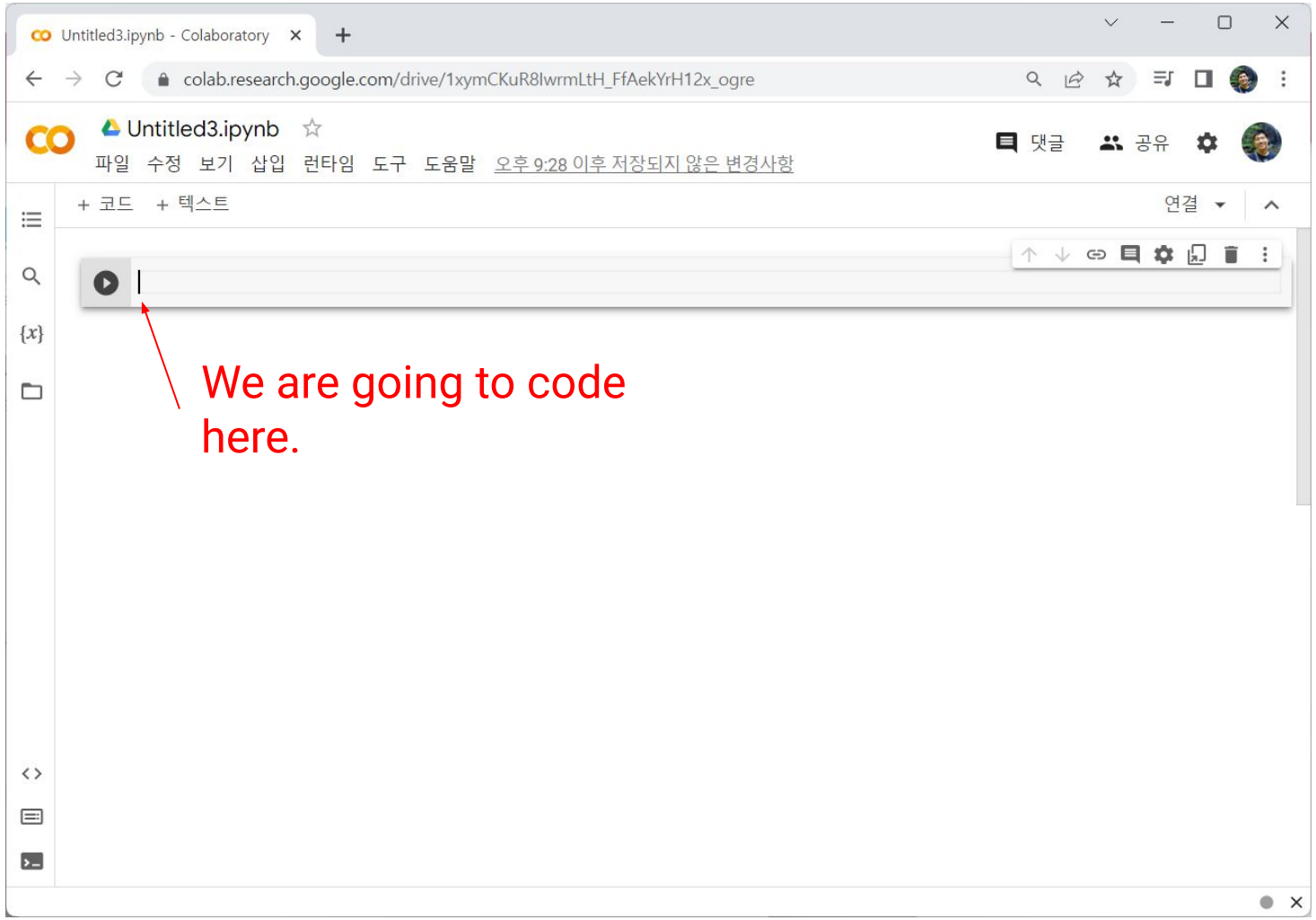
Google Colab

1. Connect to https://colab.research.google.com/



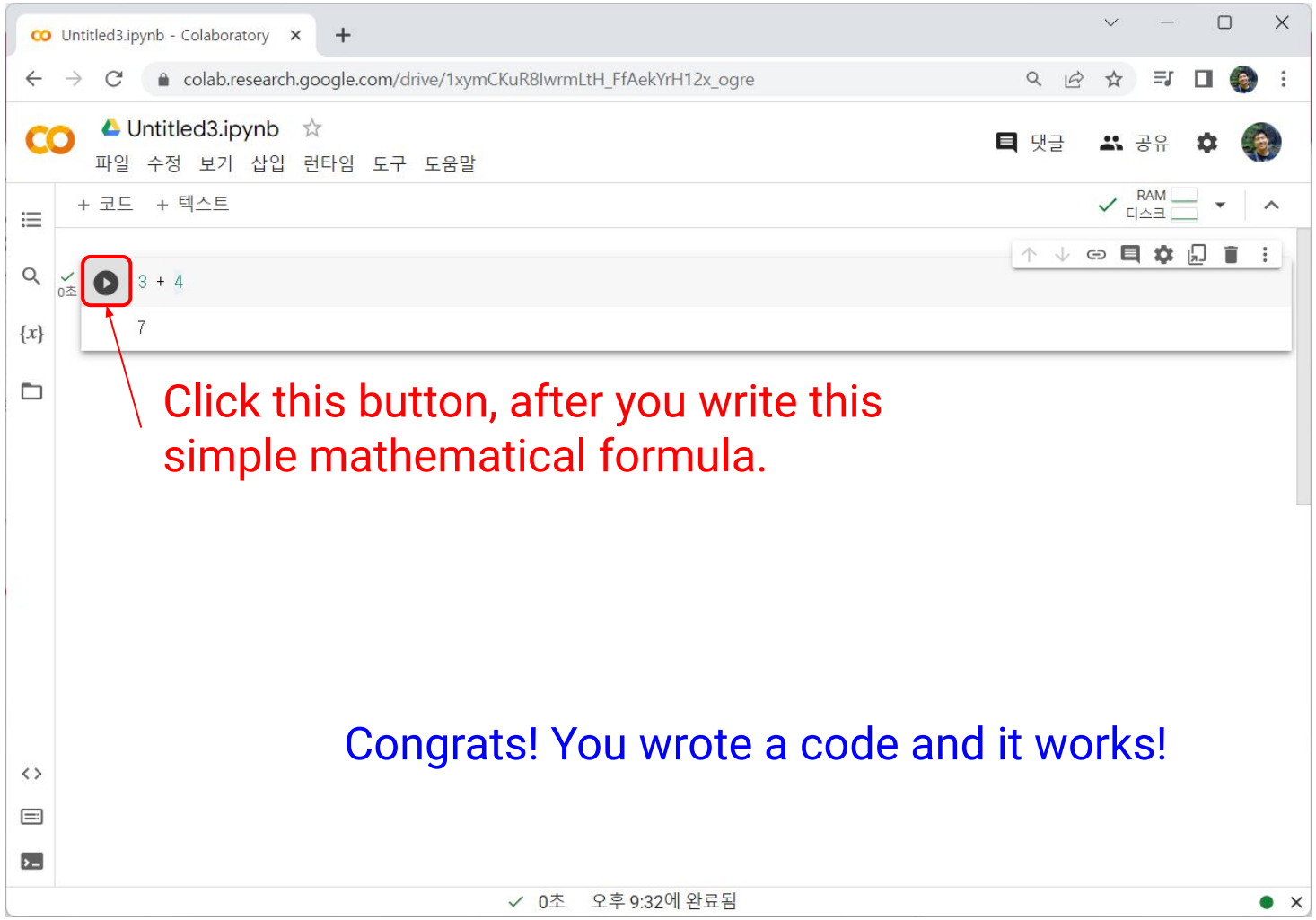
Google Colab

2. A new colab note has been created!



Google Colab

3. Let's try a simple program!



Click this button, after you write this simple mathematical formula.

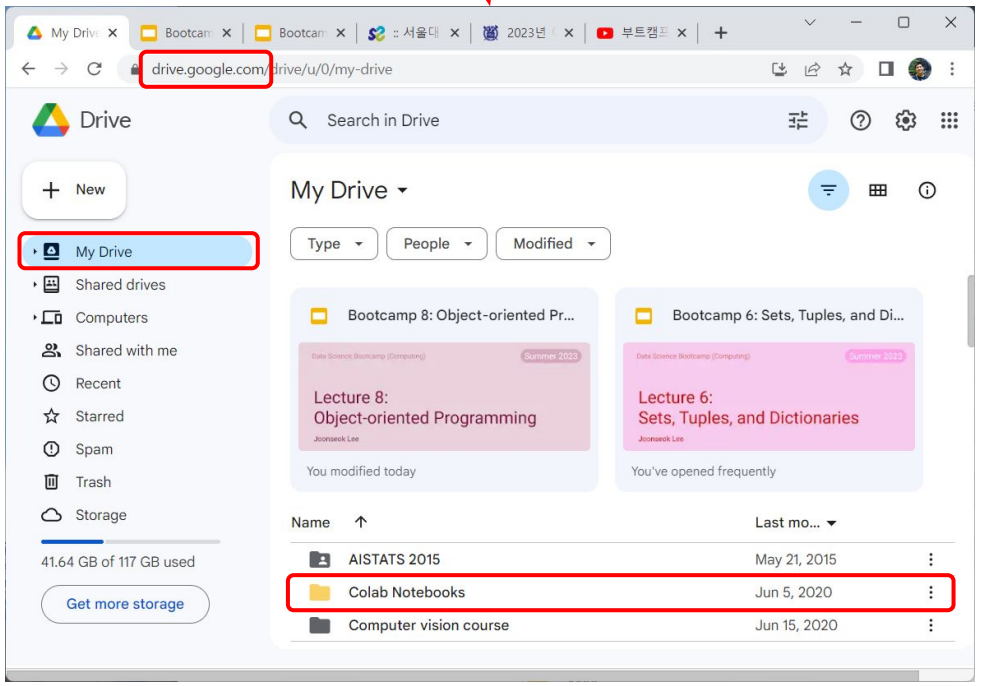
Congrats! You wrote a code and it works!

Google Colab: Opening a File

- Google Colab uses Google Drive as the disk drive that you can read files from or write them to.
  - The default directory for Colab codes is "/content/drive/My Drive/Colab Notebooks".

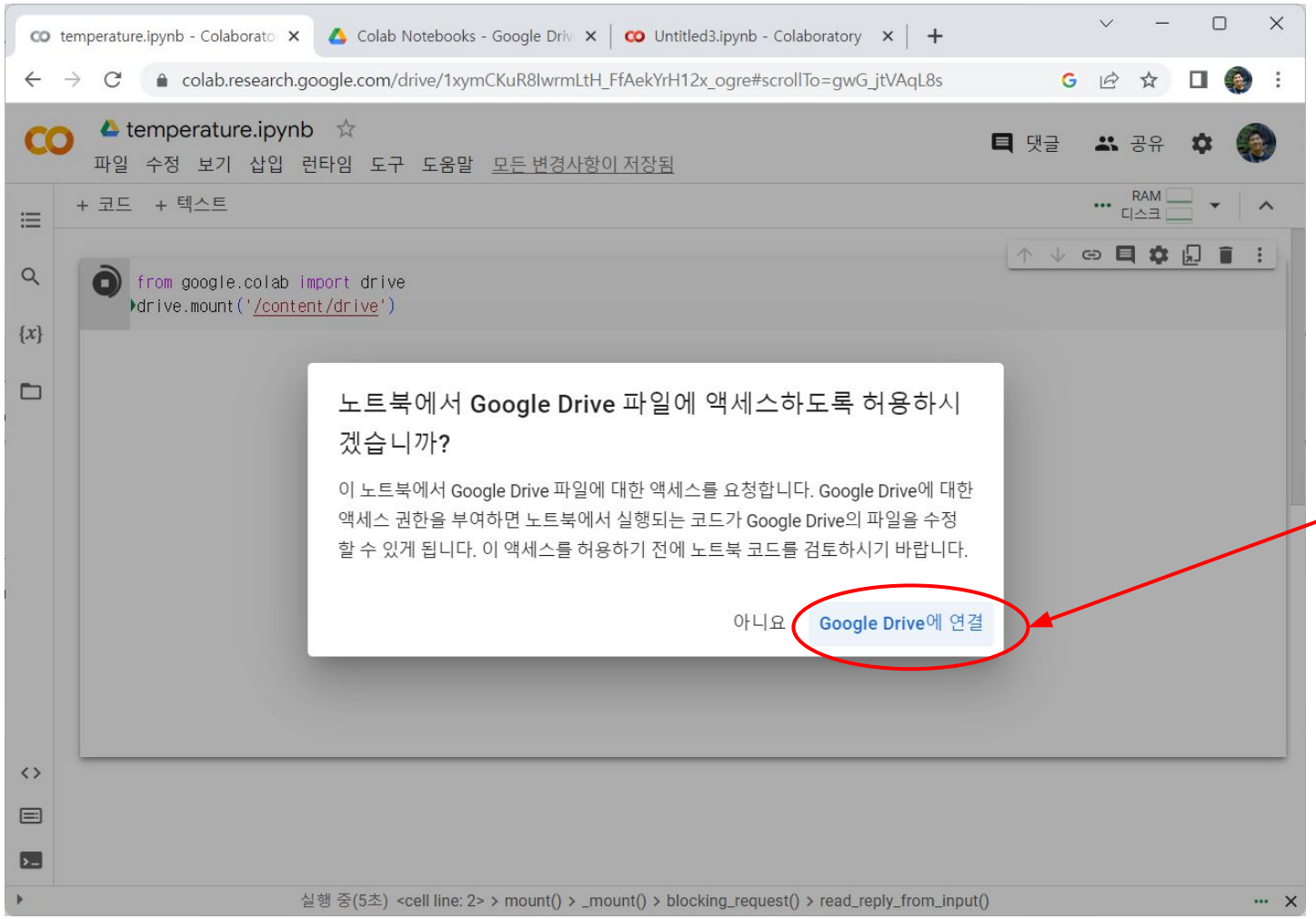
- Let's try to run the following code, to connect to your drive:

```
from google.colab import drive
drive.mount('/content/drive')
```



Google Colab: Opening a File

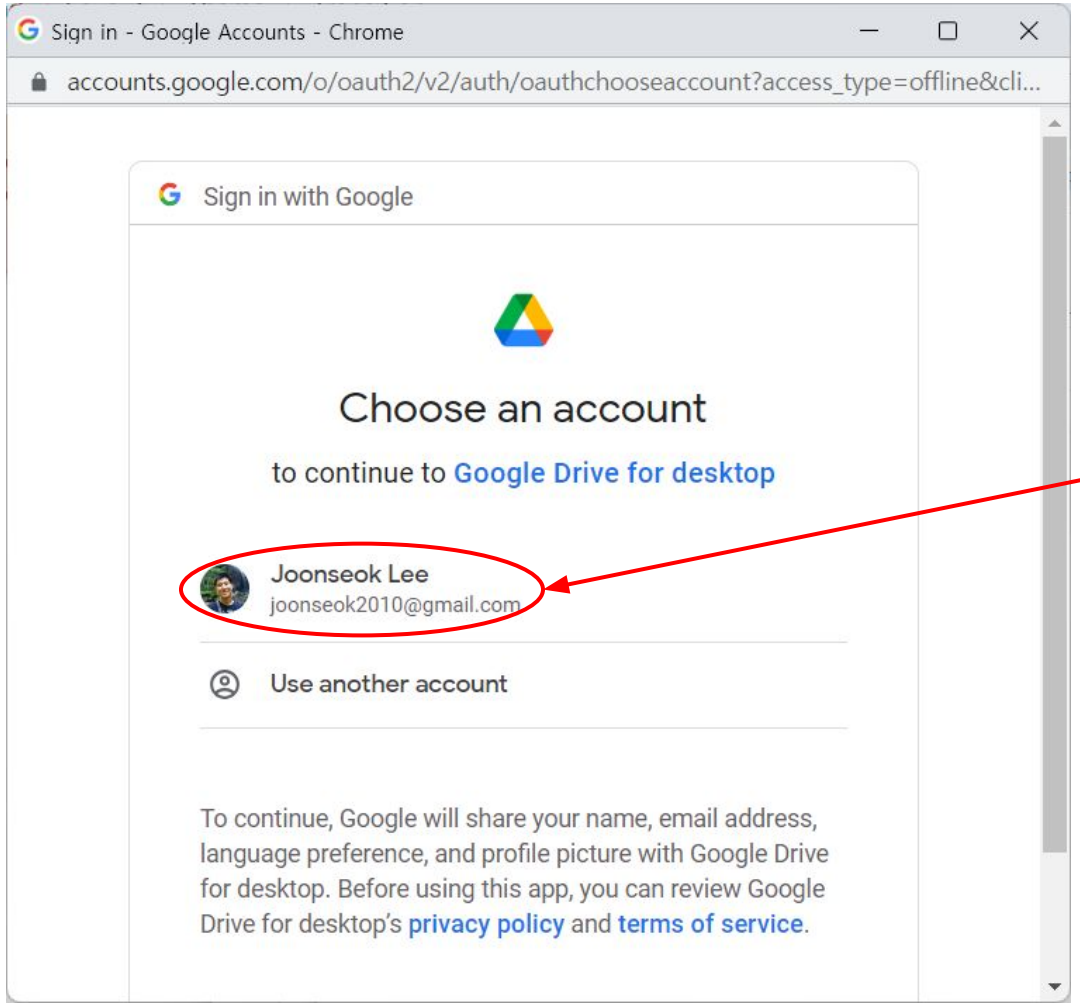
- You will see this screen:



Click here!

Google Colab: Opening a File

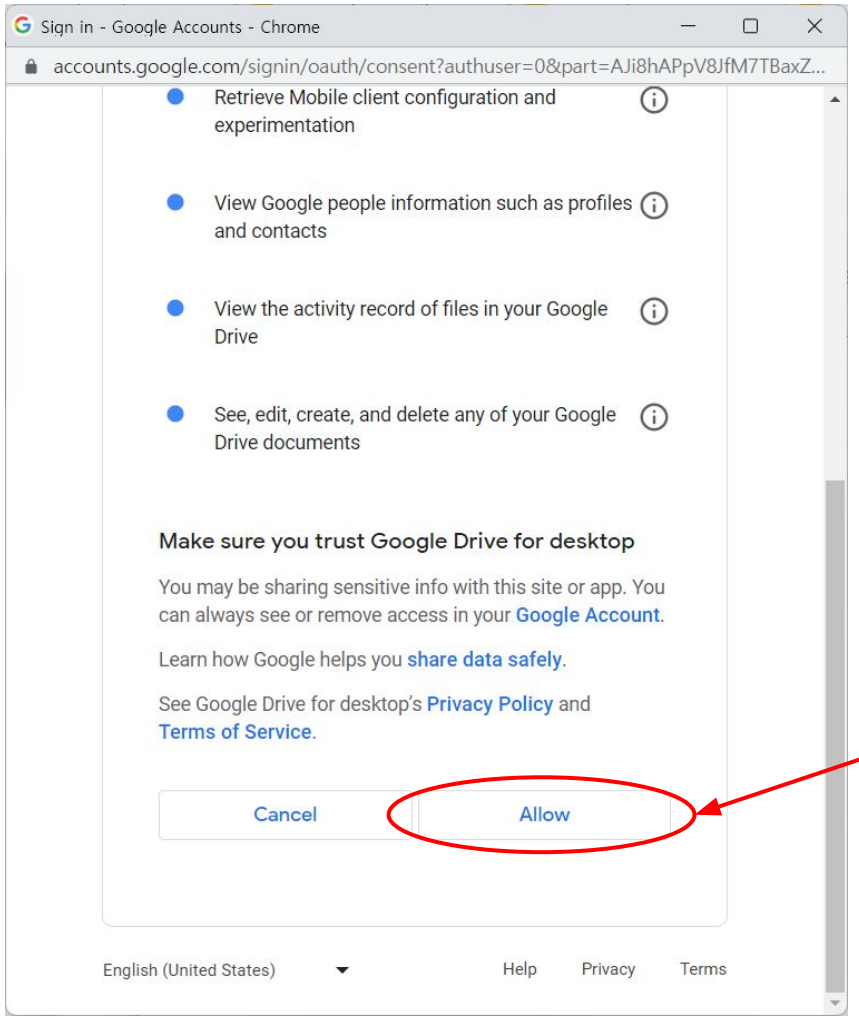
- Then, you will see another window opening:



Choose your account!

Google Colab: Opening a File

- Then, you will see another window opening:

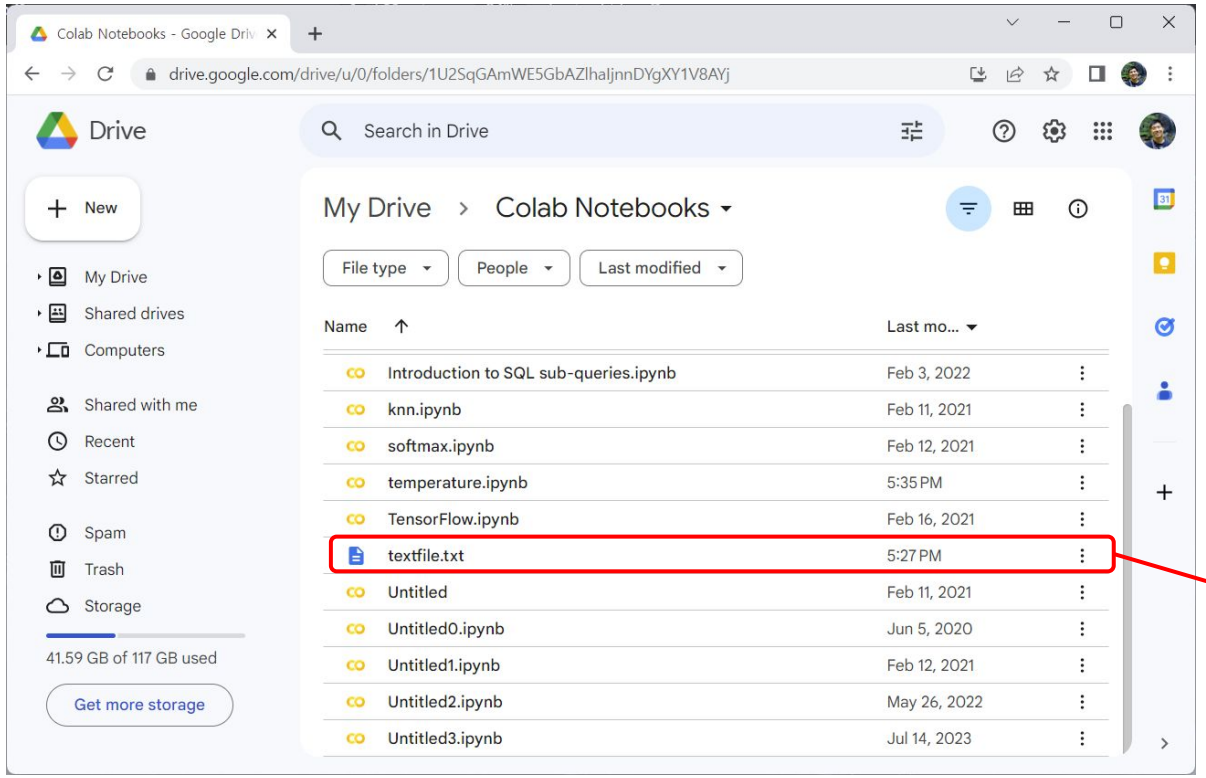


At the bottom, click this!

Google Colab: Opening a File

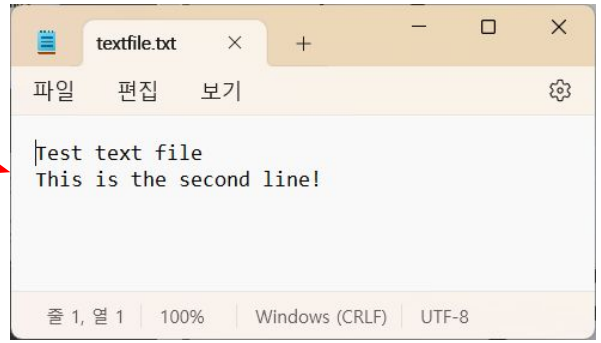
- Now, your drive is mounted to the current Colab notebook. Great!
- Let's move to the target directory by running the following code:

```
%cd '/content/drive/My Drive/Colab Notebooks'
```



You can see this directory on https://drive.google.com.

To demonstrate file reading, let's upload a simple text file on this directory.



Google Colab: Opening a File

- Let's open this file on Colab now.

```
file = open('textfile.txt', 'r')
contents = file.read()
file.close()
print(contents)
```

```
file = open('textfile.txt', 'r')
contents = file.read()
file.close()
print(contents)
```

```
Test text file
This is the second line!
```

Once executed, we can see that the contents of the text file is printed. 😃

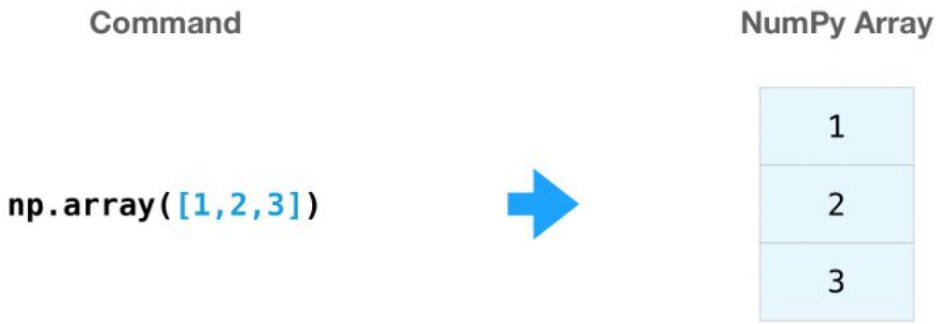
NumPy / Pandas

NumPy / Pandas

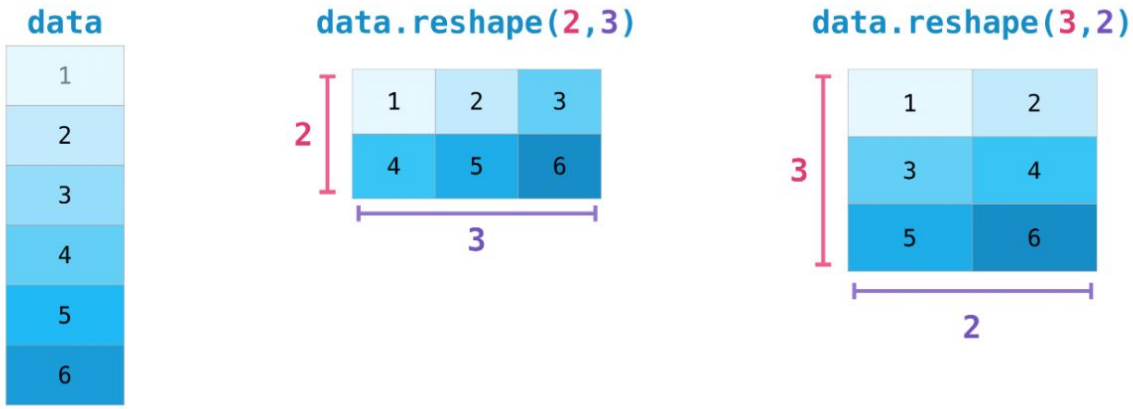
- NumPy
  - A Python library that provides support for large, multi-dimensional **arrays** and **matrices**, along with a collection of **mathematical functions** to operate on these arrays efficiently
- Pandas
  - A Python library used for **data manipulation and analysis**, mainly for structured tabular data such as CSV files

NumPy

- Generating NumPy Arrays
  - `np.array()`
  - `np.arange()`
  - `np.zeros()`
  - `np.ones()`
  - `np.eye()`

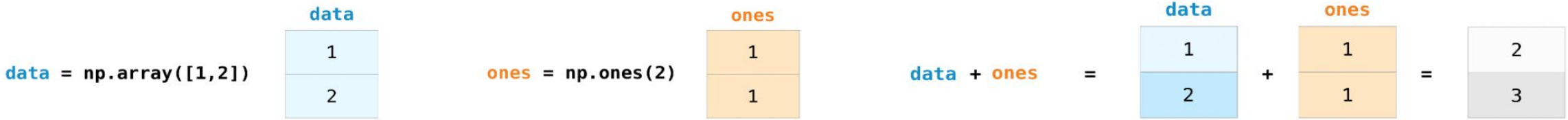


- Array Manipulation
  - Concatenating
  - Indexing and Slicing
  - reshaping
  - transposing



NumPy

- Array Operations
  - `+`, `-`, `*`, `/`
  - `add()`, `substract()`, `multiply()`, `divide()`
  - `dot()`, `sum()`, `prod()`, `inv()`
  - broadcasting



Pandas

- Pandas DataFrames

```
# csv file -> dataframe
df = pd.read_csv("/content/drive/MyDrive/2-1_MLDL/Carseats.csv")
df.head()
```

	Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
0	9.50	138	73	11	276	120	Bad	42	17	Yes	Yes
1	11.22	111	48	16	260	83	Good	65	10	Yes	Yes
2	10.06	113	35	10	269	80	Medium	59	12	Yes	Yes
3	7.40	117	100	4	466	97	Medium	55	14	Yes	Yes
4	4.15	141	64	3	340	128	Bad	38	13	Yes	No

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Sales        400 non-null    float64
 1   CompPrice    400 non-null    int64
 2   Income       400 non-null    int64
 3   Advertising  400 non-null    int64
 4   Population   400 non-null    int64
 5   Price        400 non-null    int64
 6   ShelveLoc    400 non-null    object
 7   Age          400 non-null    int64
 8   Education    400 non-null    int64
 9   Urban        400 non-null    object
 10  US           400 non-null    object
dtypes: float64(1), int64(7), object(3)
memory usage: 34.5+ KB
None
```

```
df.describe()
```

	Sales	CompPrice	Income	Advertising	Population	Price	Age	Education
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	7.496325	124.975000	68.657500	6.635000	264.840000	115.795000	53.322500	13.900000
std	2.824115	15.334512	27.986037	6.650364	147.376436	23.676664	16.200297	2.620528
min	0.000000	77.000000	21.000000	0.000000	10.000000	24.000000	25.000000	10.000000
25%	5.390000	115.000000	42.750000	0.000000	139.000000	100.000000	39.750000	12.000000
50%	7.490000	125.000000	69.000000	5.000000	272.000000	117.000000	54.500000	14.000000
75%	9.320000	135.000000	91.000000	12.000000	398.500000	131.000000	66.000000	16.000000
max	16.270000	175.000000	120.000000	29.000000	509.000000	191.000000	80.000000	18.000000

Machine Learning with Python

Machine Learning with Python

- Linear Regression
- Feature Selection
- Logistic Regression
- Discriminant Analysis

Let’s try them out ! :)

Homework 1

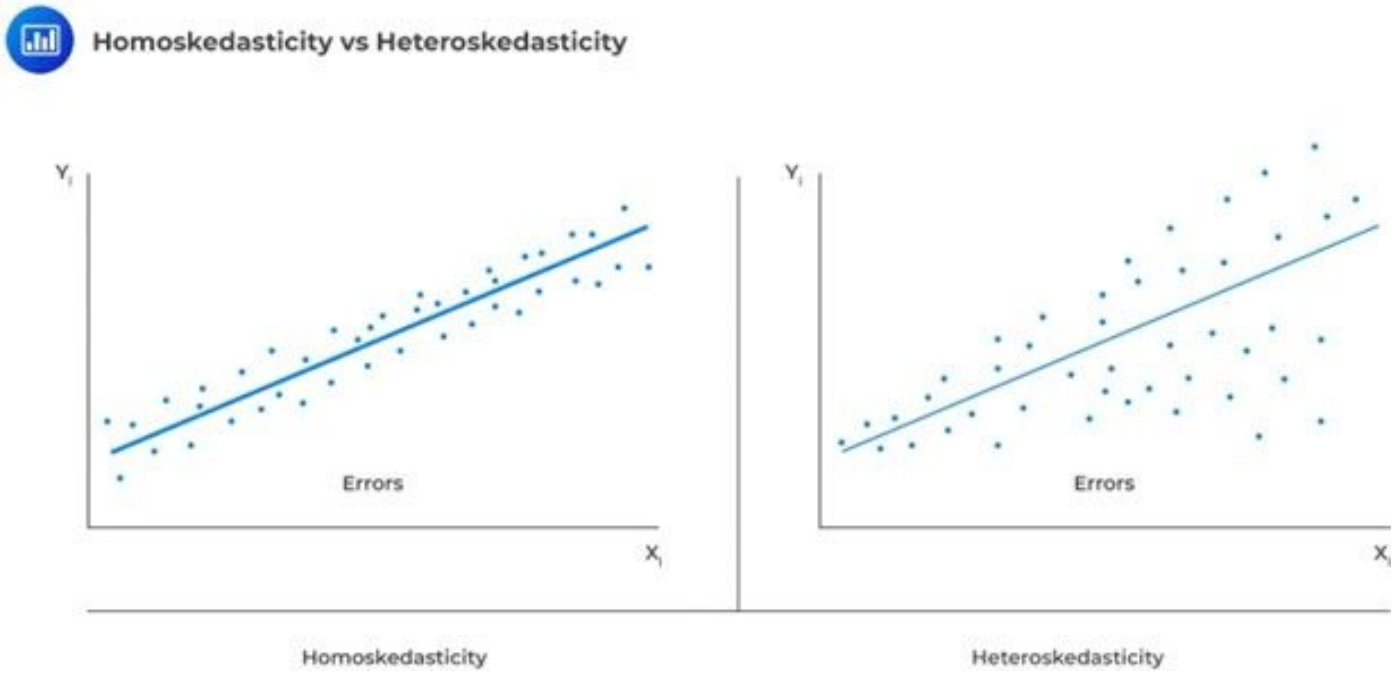
Homework 1

- **Due: 2024/4/11 Thr, 18:00**
- No late submissions allowed!
- Skeleton codes: `hw1_gd.ipynb, hw1_nb.ipynb`
- Allowed to use numpy library
  - Do NOT use any `scikit learn` packages or other equivalent ones that directly implement the question.

- Q1, 2 (MLE, Linear Regression): contents from lecture 3, 4
- Q3 (Gradient Descent): contents from lecture 5
- Q4 (Naive Bayes Classifier): contents from lecture 6
- First ask questions to ETL, then to TA - Chanwoo Kim (chanwoo.kim@snu.ac.kr)

Question 1, 2

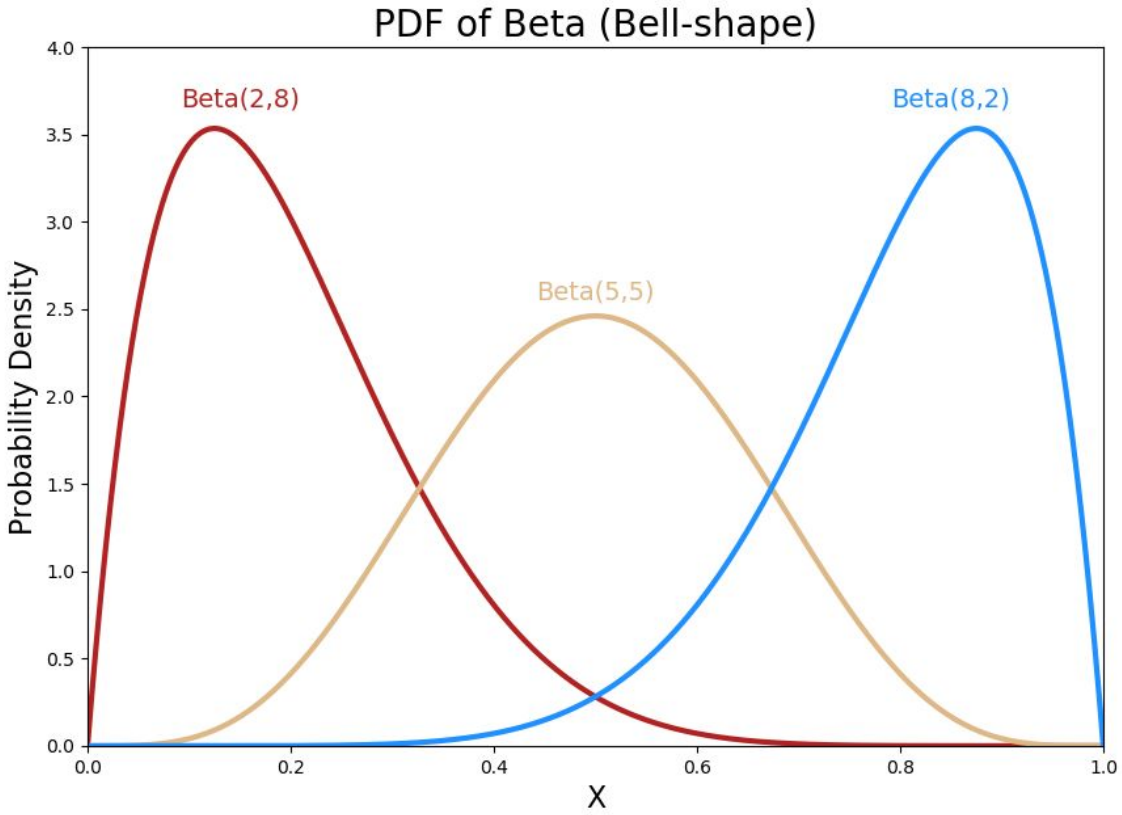
- Q1: MLE for Poisson Distribution, Exponential Distribution
- Q2: MLE for Linear Regression *with Heteroskedasticity (Non-constant variation)*



https://analystprep.com/study-notes/cfa-level-2/quantitative-method/explain-the-types-of-heteroskedasticity-and-how-it-affects-statistical-inference/

Question 3

- Q3: Gradient Descent - MLE for Beta Distribution



https://towardsdatascience.com/beta-distribution-intuition-examples-and-derivation-cf00f4db57af

Question 4

- Q4: Multinomial Naive Bayes Classifier
  - Task: **Log Classification (Text to Class)**

```
2015-10-17 15:45:11,258 INFO [main] org.apache.hadoop.metrics2.impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2015-10-17 15:45:11,399 INFO [main] org.apache.hadoop.metrics2.impl.MetricsSystemImpl: Scheduled snapshot period at 10 second(s).
2015-10-17 15:45:11,399 INFO [main] org.apache.hadoop.metrics2.impl.MetricsSystemImpl: MapTask metrics system started
2015-10-17 15:45:11,430 INFO [main] org.apache.hadoop.mapred.YarnChild: Executing with tokens:
2015-10-17 15:45:11,430 INFO [main] org.apache.hadoop.mapred.YarnChild: Kind: mapreduce.job, Service: job_1445062781478_0015, Ident: (org.apache$
2015-10-17 15:45:11,602 INFO [main] org.apache.hadoop.mapred.YarnChild: Sleeping for 0ms before retrying again. Got null now.
2015-10-17 15:45:12,196 INFO [main] org.apache.hadoop.mapred.YarnChild: mapreduce.cluster.local.dir for child: /tmp/hadoop-msrabi/nm-local-dir/u$
2015-10-17 15:45:12,211 INFO [main] org.apache.hadoop.conf.Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session$
```

Level 0 ~ 3

https://towardsdatascience.com/understand-your-computer-system-using-logs-98139d0b5de1

Question 4

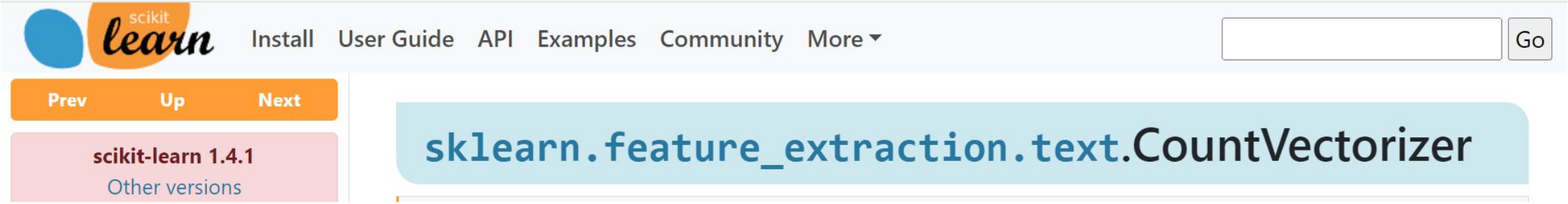
- Q4: Multinomial Naive Bayes Classifier
  - Preprocessing: **Bag of Words (BoW)**

	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0

https://wannabenice.tistory.com/34

Question 4

- Q4: Multinomial Naive Bayes Classifier
  - Preprocessing: **Bag of Words (BoW)** ***(Already Done for You!)***



https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

Question 4

- Q4: Multinomial Naive Bayes Classifier
  - Model: **Multinomial Naive Bayes**

Multinomial Distribution

Parameters	$n > 0$ number of trials (integer) $k > 0$ number of mutually exclusive events (integer) $p_1, \ldots, p_k$ event probabilities, where $p_1 + \cdots + p_k = 1$

e.g., $\mathbf{p}$ of a Dice = (0.05, 0.03, 0.02, 0.7, 0.2, 0.1), n = 10

x_1 = (3, 2, 4, 0, 1, 0) *vs.* x_2 = (0, 0, 0, 5, 3, 2)

https://en.wikipedia.org/wiki/Multinomial_distribution
https://www.researchgate.net/figure/Unusual-dice-used-in-the-experiment-Photo-by-authors_fig3_326849413

Question 4

- Q4: Multinomial Naive Bayes Classifier
  - Model: **Multinomial Naive Bayes**

$\mathbf{p}$ of Dog story= (0.05, 0.03, 0.8, 0, 0.01, 0.07)

$\mathbf{p}$ of Cat story= (0.04, 0.05, 0, 0.8, 0.01, 0.1)

	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0