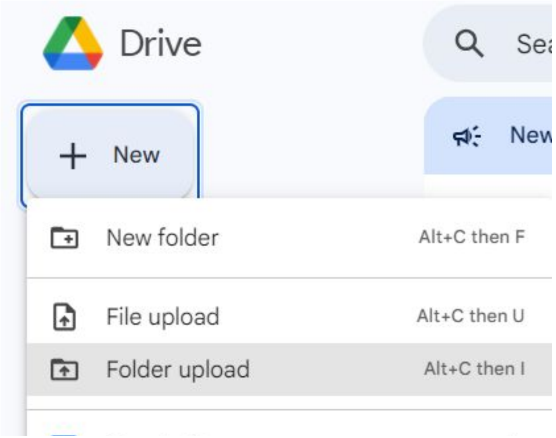
TA hours

HW1
Colab experiment : Find the top Youtubers

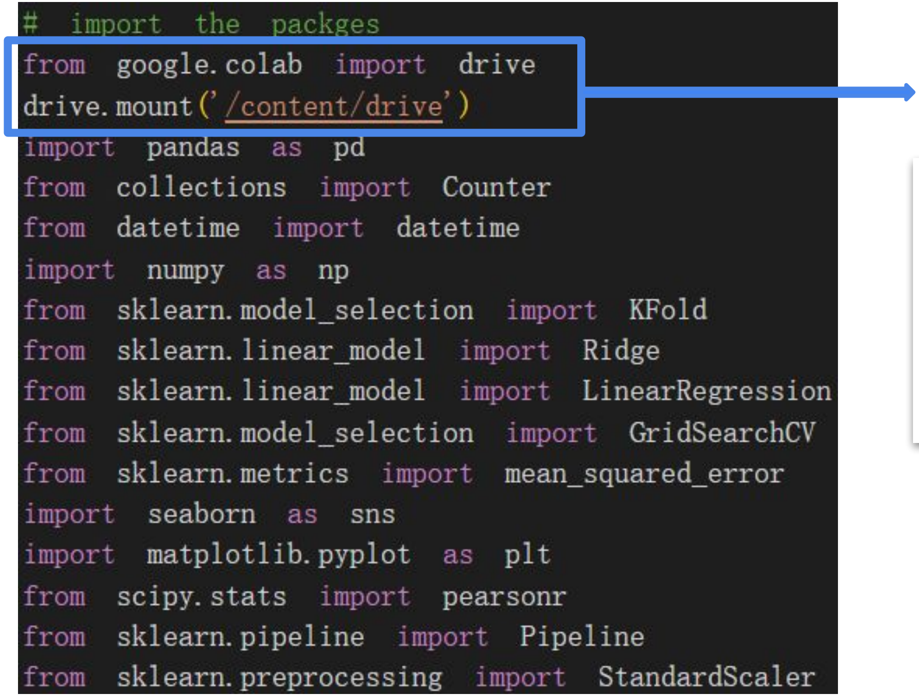
2024.09.10

Colab Setup

- Download the week2_colab folder from NTU COOL
- Upload the folder to your Google Drive
- Utilize Google Colab to open the week2_colab_homework.ipynb file

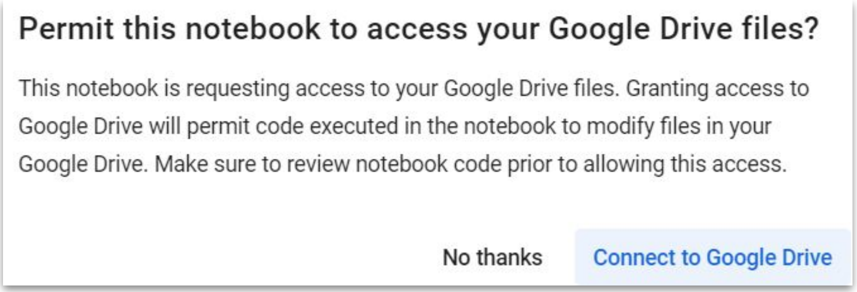


Colab Setup



```
# import the packges
from google.colab import drive
drive.mount(’/content/drive’)
import pandas as pd
from collections import Counter
from datetime import datetime
import numpy as np
from sklearn.model_selection import KFold
from sklearn.linear_model import Ridge
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import pearsonr
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
```

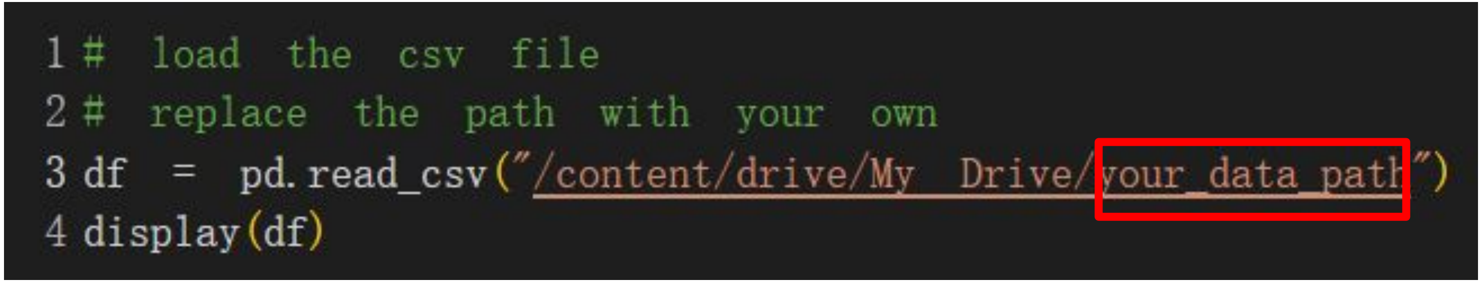
- Get permission to your drive.



Permit this notebook to access your Google Drive files?

This notebook is requesting access to your Google Drive files. Granting access to Google Drive will permit code executed in the notebook to modify files in your Google Drive. Make sure to review notebook code prior to allowing this access.

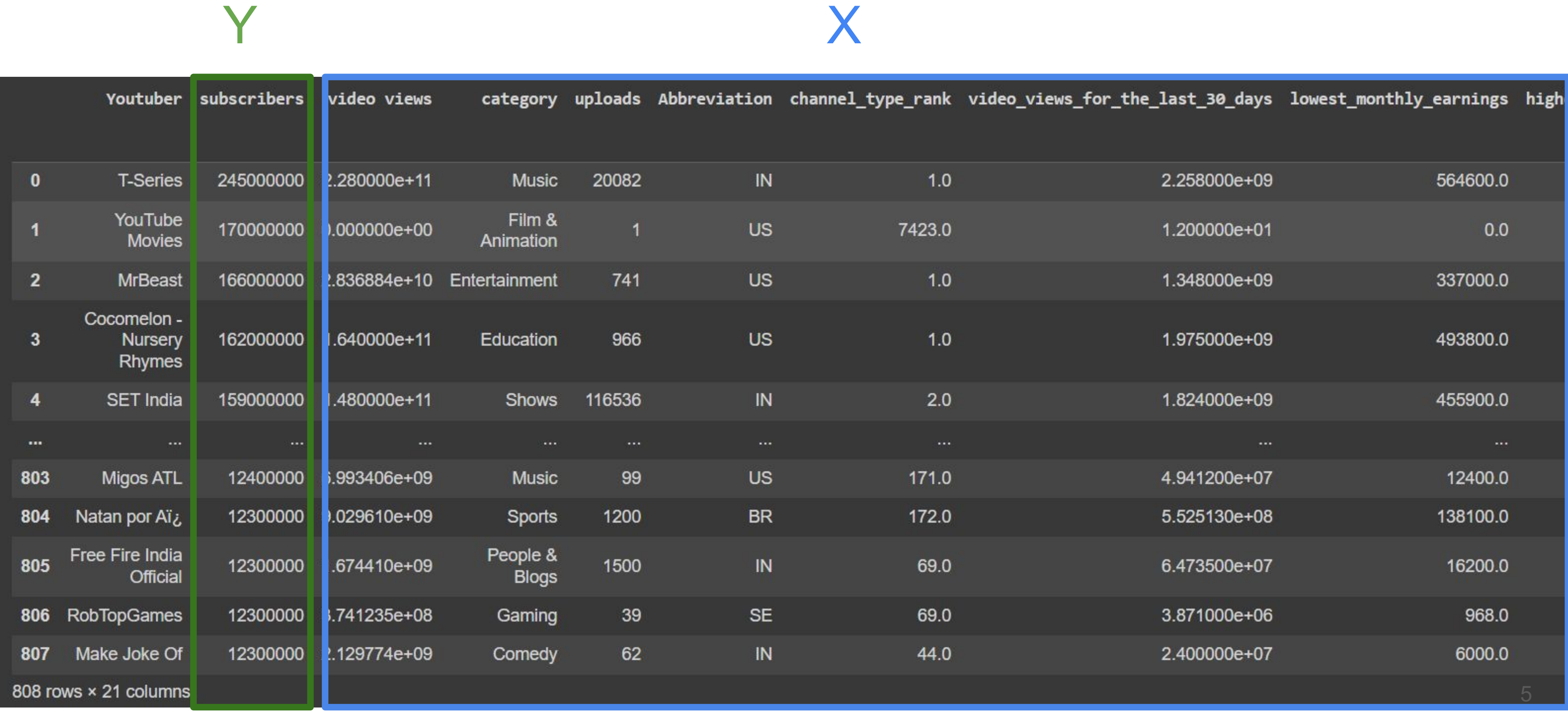
Load the dataset



```
# load the csv file
# replace the path with your own
df = pd.read_csv("/content/drive/My Drive/your_data_path")
display(df)
```

Caution ! changed to your own path

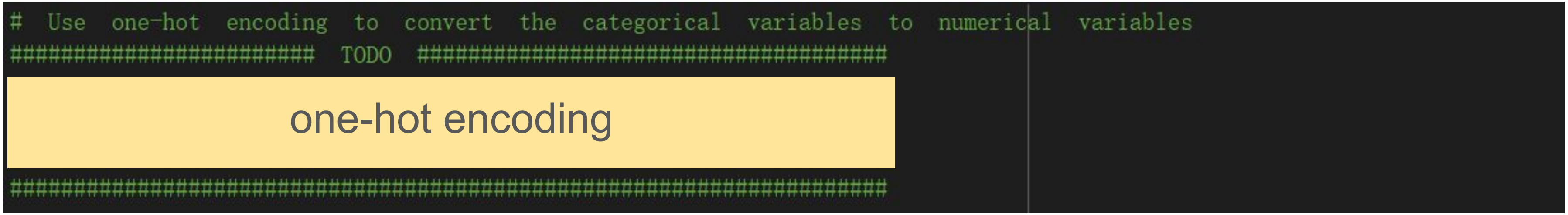
- /content/drive/My Drive / your_data_path



	Youtuber	subscribers	video views	category	uploads	Abbreviation	channel_type_rank	video_views_for_the_last_30_days	lowest_monthly_earnings	high
0	T-Series	245000000	2.280000e+11	Music	20082	IN	1.0	2.258000e+09	564600.0	
1	YouTube Movies	170000000	0.000000e+00	Film & Animation	1	US	7423.0	1.200000e+01	0.0	
2	MrBeast	166000000	2.836884e+10	Entertainment	741	US	1.0	1.348000e+09	337000.0	
3	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	966	US	1.0	1.975000e+09	493800.0	
4	SET India	159000000	1.480000e+11	Shows	116536	IN	2.0	1.824000e+09	455900.0	
...	...	...	...	...	...	...	...	...	...	
803	Migos ATL	12400000	6.993406e+09	Music	99	US	171.0	4.941200e+07	12400.0	
804	Natan por Aï¿	12300000	9.029610e+09	Sports	1200	BR	172.0	5.525130e+08	138100.0	
805	Free Fire India Official	12300000	1.674410e+09	People & Blogs	1500	IN	69.0	6.473500e+07	16200.0	
806	RobTopGames	12300000	3.741235e+08	Gaming	39	SE	69.0	3.871000e+06	968.0	
807	Make Joke Of	12300000	2.129774e+09	Comedy	62	IN	44.0	2.400000e+07	6000.0	

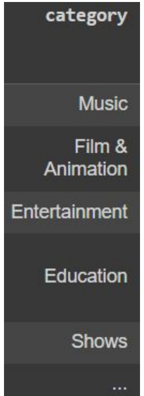
808 rows × 21 columns

TODO - Using one-hot encoding to preprocess the data

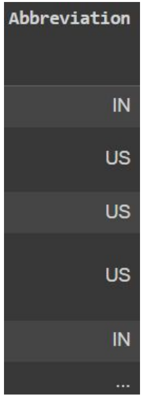


```
# Use one-hot encoding to convert the categorical variables to numerical variables
####################### TODO ####################################
one-hot encoding
##################################################################
```

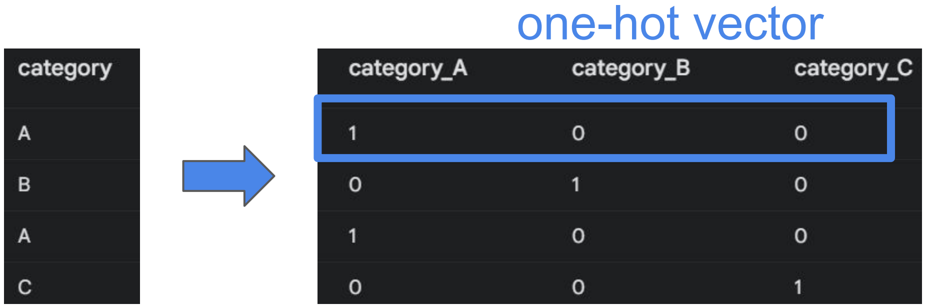
- Use one-hot encoding to convert the categorical variables(category, abbreviation) to numerical variables



category
Music
Film & Animation
Entertainment
Education
Shows
...



Abbreviation
IN
US
US
US
IN
...



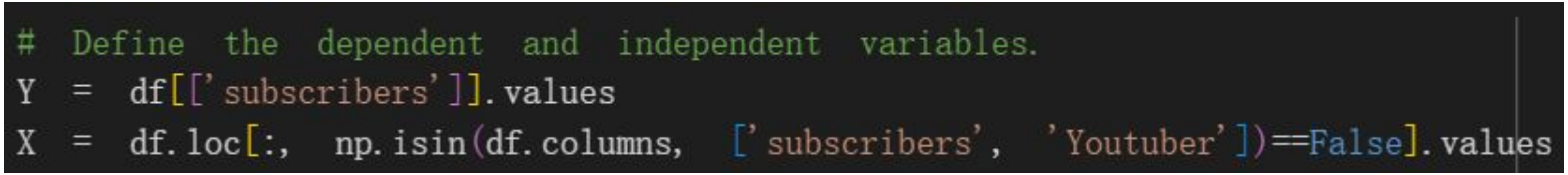
one-hot vector

category
A
B
A
C

category_A	category_B	category_C
1	0	0
0	1	0
1	0	0
0	0	1

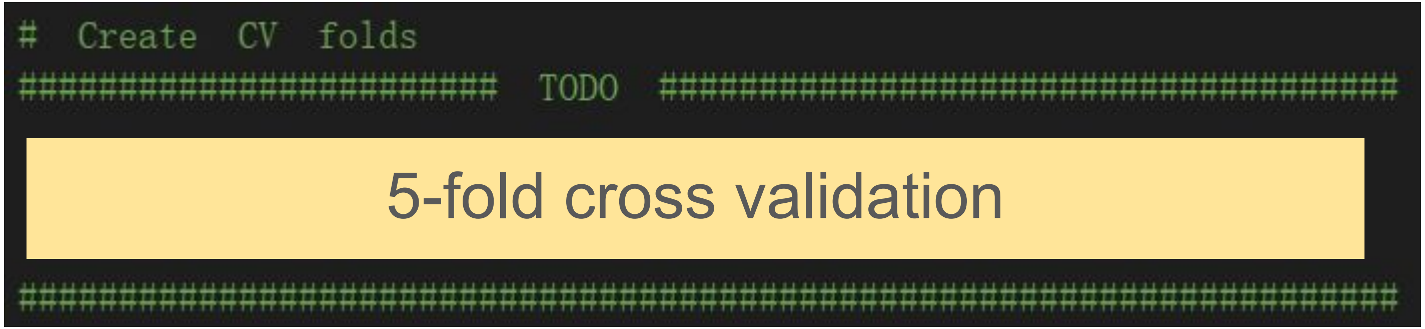
Dependent and Independent variables

- You can take this as a baseline and explore other possibilities with different X and Y values.



```
# Define the dependent and independent variables.
Y = df[[’subscribers’]].values
X = df.loc[:, np.isin(df.columns, [’subscribers’, ’Youtuber’])==False].values
```

TODO - Separate training and testing set with K-fold

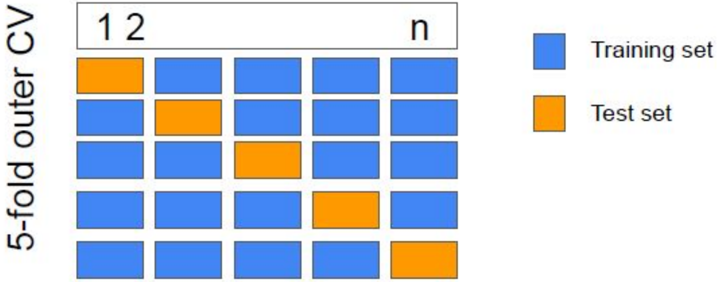


```
# Create CV folds
######################### TODO #######################################
```

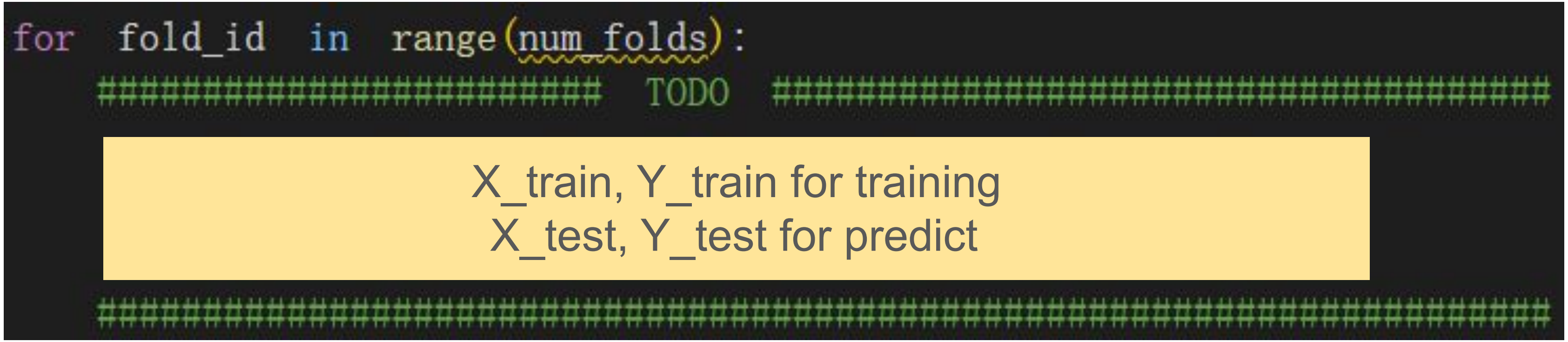
5-fold cross validation

```
######################################################################
```

- Divide the dataset into k-fold, and store the training data index and test data index of each fold.
- Reference: Kfold function from Sklearn



TODO - Complete the training step in each fold

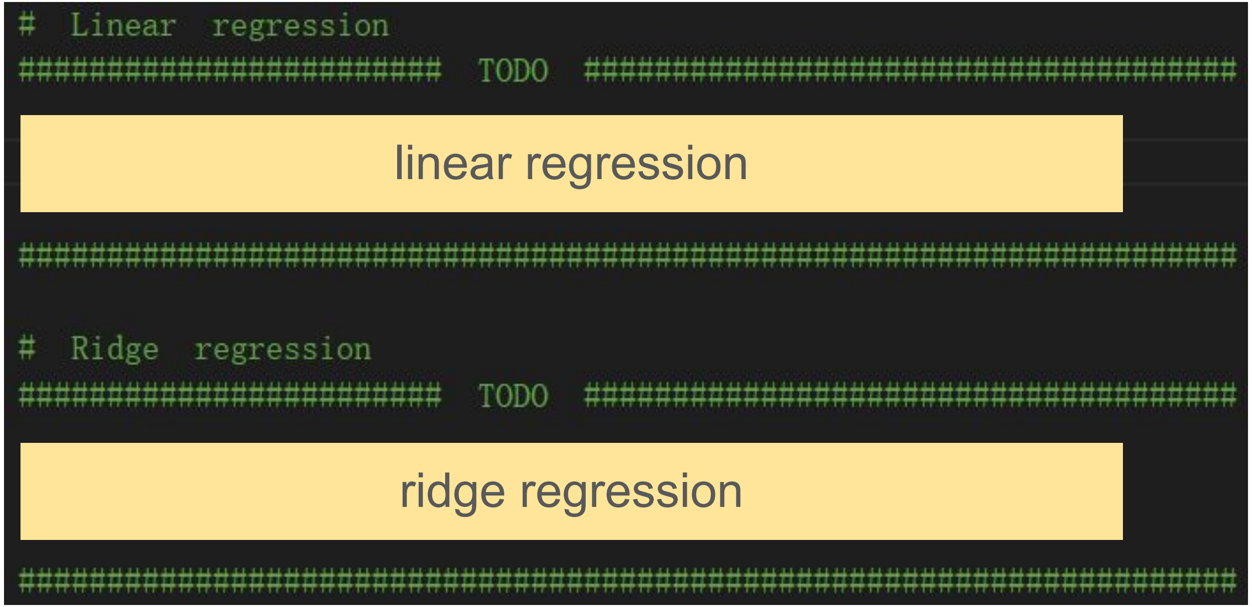


```
for  fold_id  in  range(num_folds):
    ########################  TODO  #####################################

    ######################################################################
```

X_train, Y_train for training
X_test, Y_test for predict

- According to the folding index of each fold obtained previously, the data can be divided into a training set and a test set.

TODO - Implement linear regression && ridge regression



```
#  Linear  regression
########################  TODO  #####################################
```

linear regression

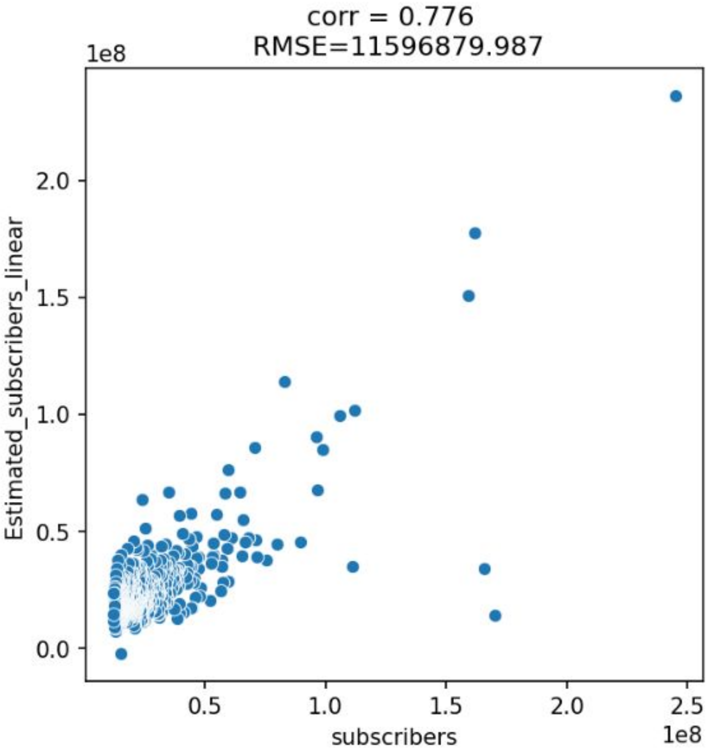
```
#####################################################################

#  Ridge  regression
########################  TODO  #####################################
```

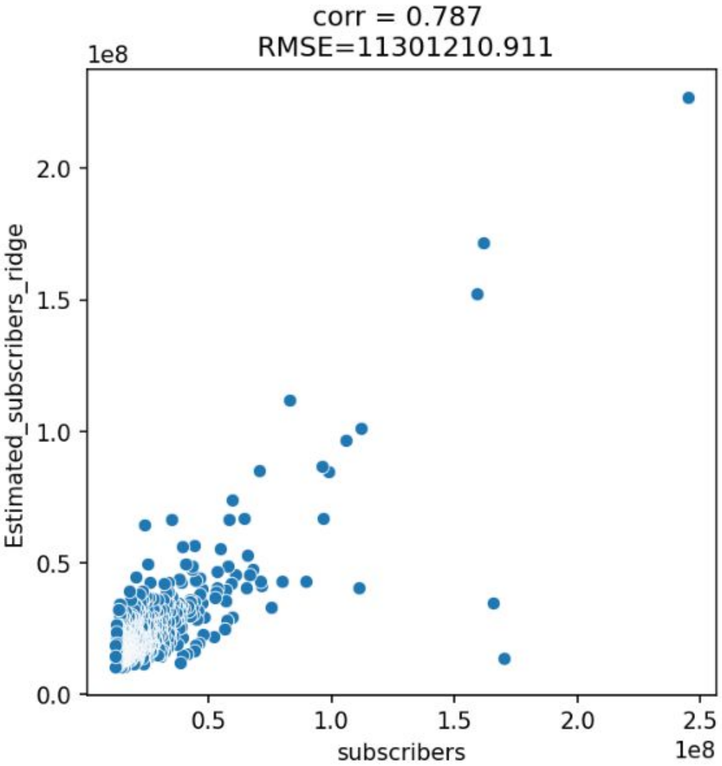
ridge regression

```
#####################################################################
```

- Reference: linear regression, ridge regression function from Sklearn

The reference result



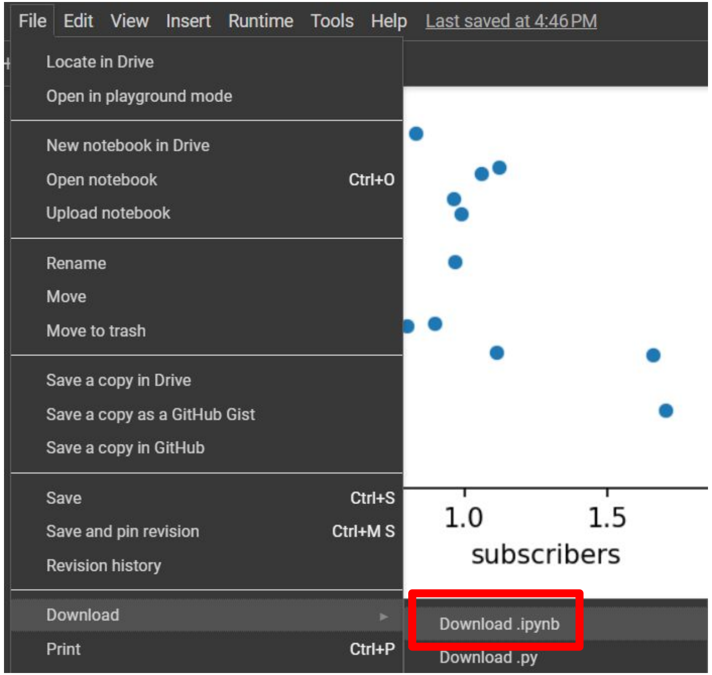


Conclusion

1. Using one hot encoding to preprocess the data.
2. Separate training and testing set with Kfold.
3. Complete the training step in each fold.
4. Implement linear regression && ridge regression.
5. Description of the methodology
6. Conclusions and discussions

Reminder

- After executing your code, download the .ipynb file and submit it to NTU COOL
- HW1 Deadline : 2024/9/16 23:59:59


File
Edit
View
Insert
Runtime
Tools
Help
Last saved at 4:46 PM
Locate in Drive
Open in playground mode
New notebook in Drive
Open notebook
Ctrl+O
Upload notebook
Rename
Move
Move to trash
Save a copy in Drive
Save a copy as a GitHub Gist
Save a copy in GitHub
Save
Ctrl+S
Save and pin revision
Ctrl+M S
Revision history
Download
Print
Ctrl+P
Download .ipynb
Download .py
1.0
1.5
subscribers