3장 구현을 위한 도구

❖ 개발 환경 & 언어
❖ Numpy
❖ Pandas
❖ Kaggle API

Python Tutorial

- 파이썬 자습서
  - https://docs.python.org/ko/3/tutorial/index.html

- Jump to Python (wikidocs)
  - https://wikidocs.net/book/1

구글 CoLab

- 구글 코랩 (Google Colab) 설치와 GPU 사용
  - https://www.youtube.com/watch?v=vRu77RmGD-M

- 구글 코랩(Colab) 사용법
  - https://www.youtube.com/watch?v=v19SzGMOd2c
  - https://www.youtube.com/watch?v=mlI1g26lJQM
  - https://colab.research.google.com/notebooks/intro.ipynb
  - https://www.youtube.com/watch?v=wb4F1aeZtRA
  - https://youtu.be/inN8seMm7UI
  - 김태영님 블로그 : https://tykimos.github.io/2019/01/22/colab_getting_started/
  - 파일을 업로드, 다운로드 하는 방법: http://www.dreamy.pe.kr/zbxe/CodeClip/3769485

  - 학습_플랫폼_및_라이브러리_소개.ipynb

Numpy & Pandas

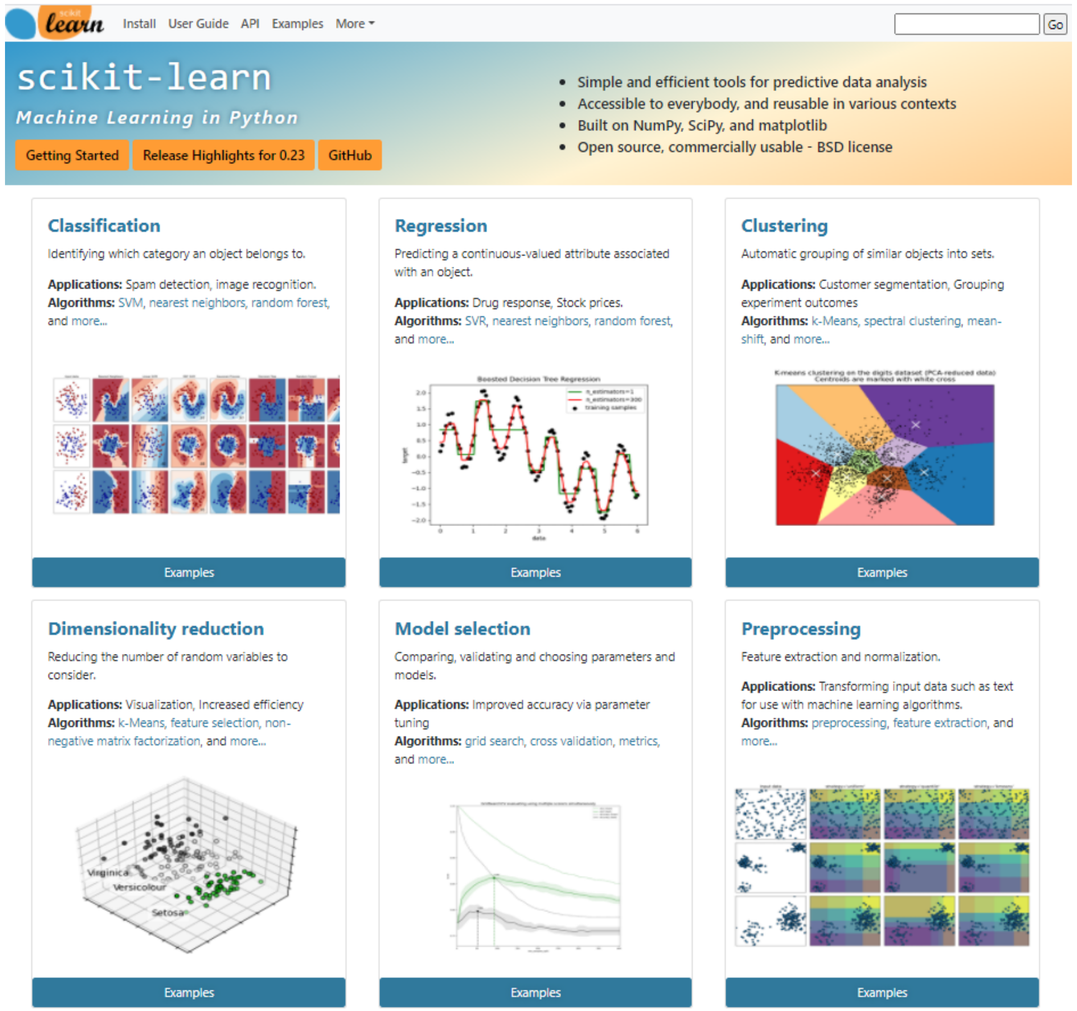
- Numpy
  - http://aikorea.org/cs231n/python-numpy-tutorial/
  - https://datascienceschool.net/intro.html
  - NumPy_기초.ipynb
  - Numpy_고급.ipynb

- Pandas

- Machine learning, deep learning book landscape (박해선)
  - https://www.youtube.com/watch?v=WHn5My6dN7c

Scikit-Learn (1)

- https://scikit-learn.org/stable/
- https://www.youtube.com/watch?v=eVxGhCRN-xA (1:48:00)



scikit-learn

Machine Learning in Python

Getting Started | Release Highlights for 0.23 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...

Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...

Examples

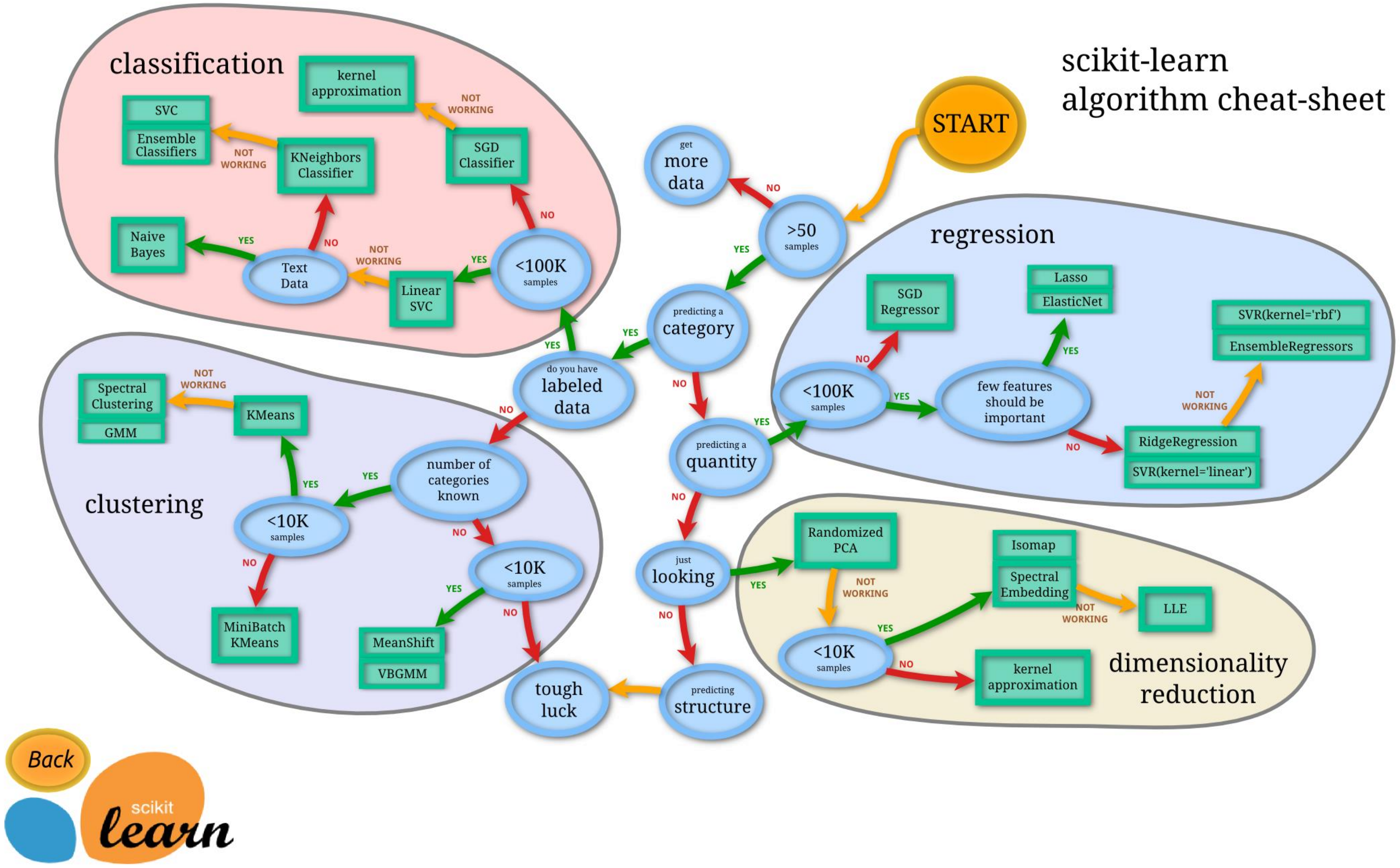
Preprocessing

Feature extraction and normalization.

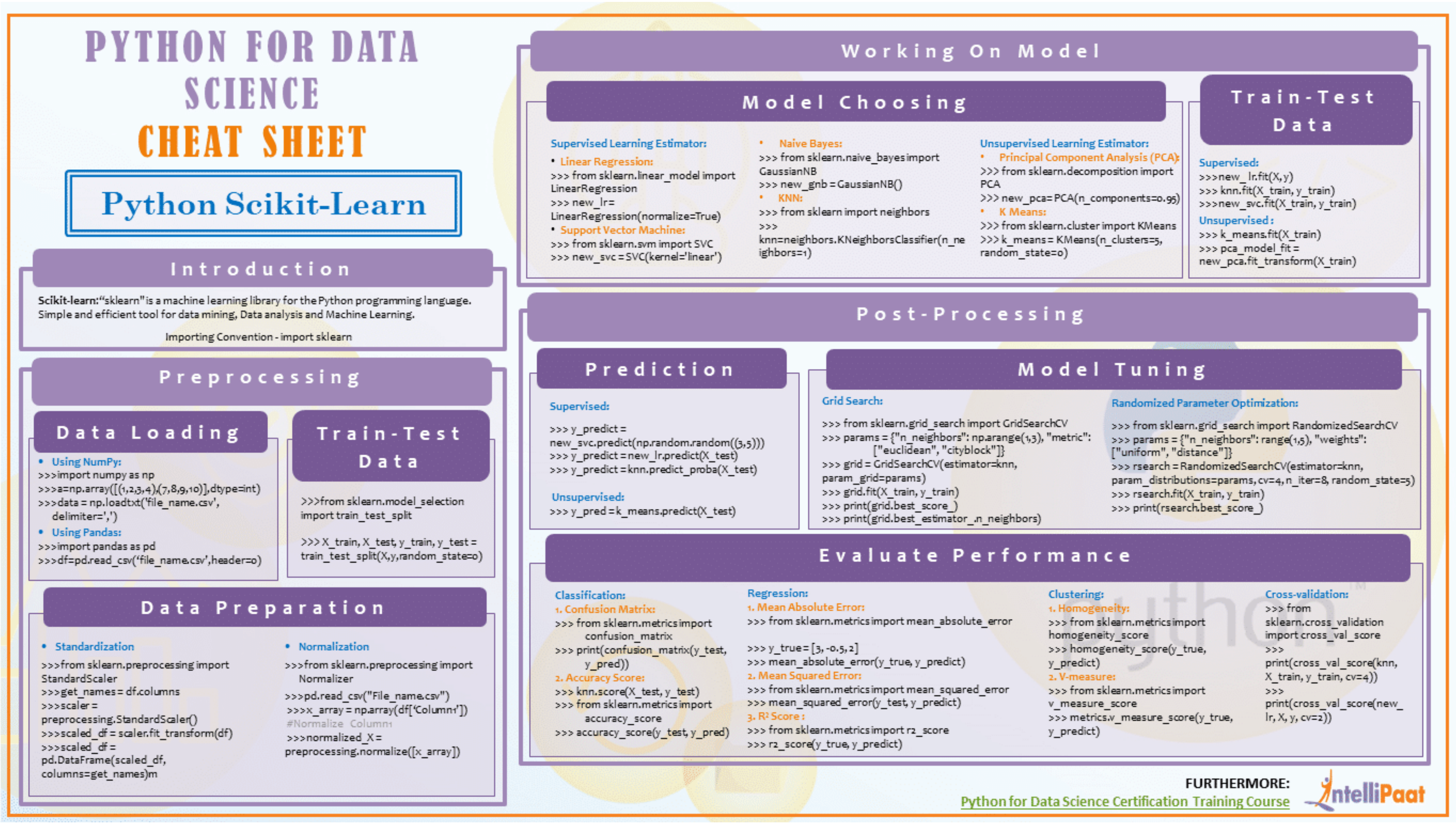
Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: preprocessing, feature extraction, and more...

Examples

Scikit-Learn (2)



Scikit-Learn (3)



PYTHON FOR DATA SCIENCE
CHEAT SHEET

Python Scikit-Learn

Introduction

Scikit-learn:"sklearn" is a machine learning library for the Python programming language. Simple and efficient tool for data mining, Data analysis and Machine Learning.

Importing Convention - import sklearn

Preprocessing

Data Loading

- Using NumPy:

```
>>>import numpy as np
>>>a=np.array([(1,2,3,4),(7,8,9,10)],dtype=int)
>>>data = np.loadtxt('file_name.csv',
   delimiter=',')
```

- Using Pandas:

```
>>>import pandas as pd
>>>df=pd.read_csv('file_name.csv',header=0)
```

Train-Test Data

```
>>>from sklearn.model_selection
import train_test_split

>>> X_train, X_test, y_train, y_test =
train_test_split(X,y,random_state=0)
```

Data Preparation

- Standardization

```
>>>from sklearn.preprocessing import
StandardScaler
>>>get_names = df.columns
>>>scaler =
preprocessing.StandardScaler()
>>>scaled_df = scaler.fit_transform(df)
>>>scaled_df =
pd.DataFrame(scaled_df,
columns=get_names)m
```

- Normalization

```
>>>from sklearn.preprocessing import
   Normalizer
>>>pd.read_csv("File_name.csv")
 >>>x_array = np.array(df['Column1'])
 #Normalize  Column1
 >>>normalized_X =
preprocessing.normalize([x_array])
```

Working On Model

Model Choosing

Supervised Learning Estimator:

- Linear Regression:

```
>>> from sklearn.linear_model import
LinearRegression
>>> new_lr =
LinearRegression(normalize=True)
```

- Support Vector Machine:

```
>>> from sklearn.svm import SVC
>>> new_svc = SVC(kernel='linear')
```

- Naive Bayes:

```
>>> from sklearn.naive_bayes import
GaussianNB
>>> new_gnb = GaussianNB()
```

- KNN:

```
>>> from sklearn import neighbors
>>>
knn=neighbors.KNeighborsClassifier(n_ne
ighbors=1)
```

Unsupervised Learning Estimator:

- Principal Component Analysis (PCA):

```
>>> from sklearn.decomposition import
PCA
>>> new_pca= PCA(n_components=0.95)
```

- K Means:

```
>>> from sklearn.cluster import KMeans
>>> k_means = KMeans(n_clusters=5,
random_state=0)
```

Train-Test Data

Supervised:

```
>>>new_ lr.fit(X, y)
>>> knn.fit(X_train, y_train)
>>>new_svc.fit(X_train, y_train)
```

Unsupervised :

```
>>> k_means.fit(X_train)
>>> pca_model_fit =
new_pca.fit_transform(X_train)
```

Post-Processing

Prediction

Supervised:

```
>>> y_predict =
new_svc.predict(np.random.random((3,5)))
>>> y_predict = new_lr.predict(X_test)
>>> y_predict = knn.predict_proba(X_test)
```

Unsupervised:

```
>>> y_pred = k_means.predict(X_test)
```

Model Tuning

Grid Search:

```
>>> from sklearn.grid_search import GridSearchCV
>>> params = {"n_neighbors": np.arange(1,3), "metric":
          ["euclidean", "cityblock"]}
>>> grid = GridSearchCV(estimator=knn,
param_grid=params)
>>> grid.fit(X_train, y_train)
>>> print(grid.best_score_)
>>> print(grid.best_estimator_.n_neighbors)
```

Randomized Parameter Optimization:

```
>>> from sklearn.grid_search import RandomizedSearchCV
>>> params = {"n_neighbors": range(1,5), "weights":
["uniform", "distance"]}
>>> rsearch = RandomizedSearchCV(estimator=knn,
param_distributions=params, cv=4, n_iter=8, random_state=5)
>>> rsearch.fit(X_train, y_train)
>>> print(rsearch.best_score_)
```

Evaluate Performance

Classification:

1. Confusion Matrix:

```
>>> from sklearn.metrics import
     confusion_matrix
>>> print(confusion_matrix(y_test,
     y_pred))
```

2. Accuracy Score:

```
>>> knn.score(X_test, y_test)
>>> from sklearn.metrics import
     accuracy_score
>>> accuracy_score(y_test, y_pred)
```

Regression:

1. Mean Absolute Error:

```
>>> from sklearn.metrics import mean_absolute_error

>>> y_true = [3, -0.5, 2]
>>> mean_absolute_error(y_true, y_predict)
```

2. Mean Squared Error:

```
>>> from sklearn.metrics import mean_squared_error
>>> mean_squared_error(y_test, y_predict)
```

3. R² Score :

```
>>> from sklearn.metrics import r2_score
>>> r2_score(y_true, y_predict)
```

Clustering:

1. Homogeneity:

```
>>> from sklearn.metrics import
homogeneity_score
>>> homogeneity_score(y_true,
y_predict)
```

2. V-measure:

```
>>> from sklearn.metrics import
v_measure_score
>>> metrics.v_measure_score(y_true,
y_predict)
```

Cross-validation:

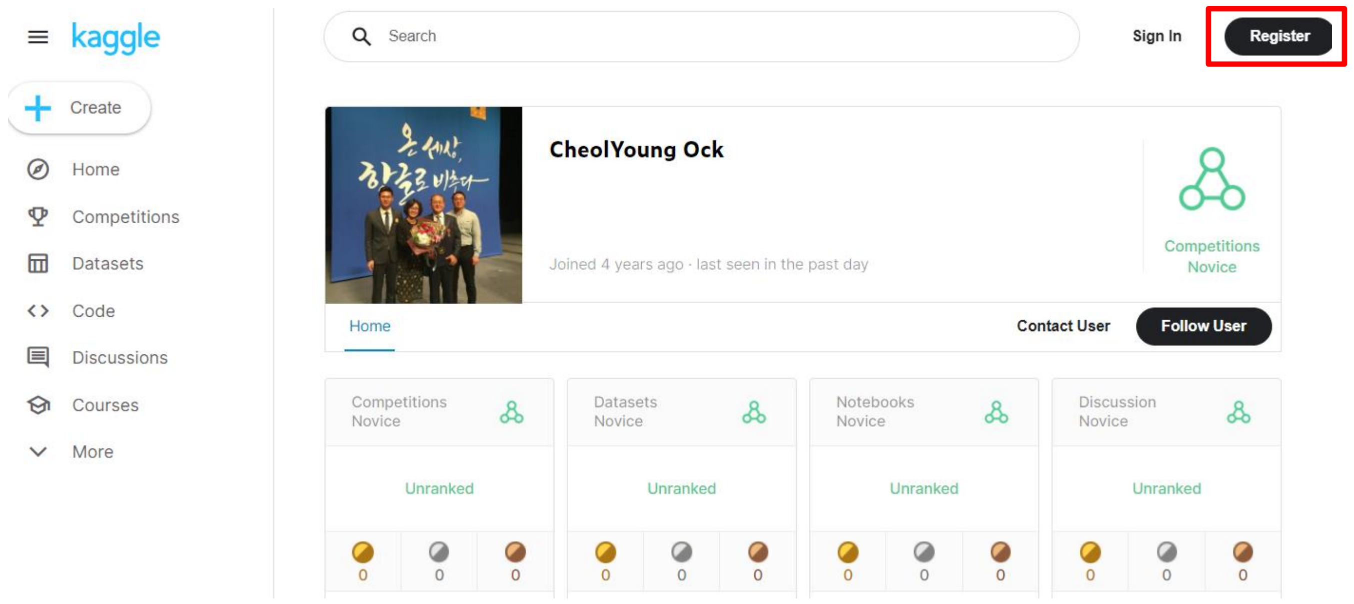
```
>>> from
sklearn.cross_validation
import cross_val_score
>>>
print(cross_val_score(knn,
X_train, y_train, cv=4))
>>>
print(cross_val_score(new_
lr, X, y, cv=2))
```

FURTHERMORE:
Python for Data Science Certification Training Course

IntelliPaat

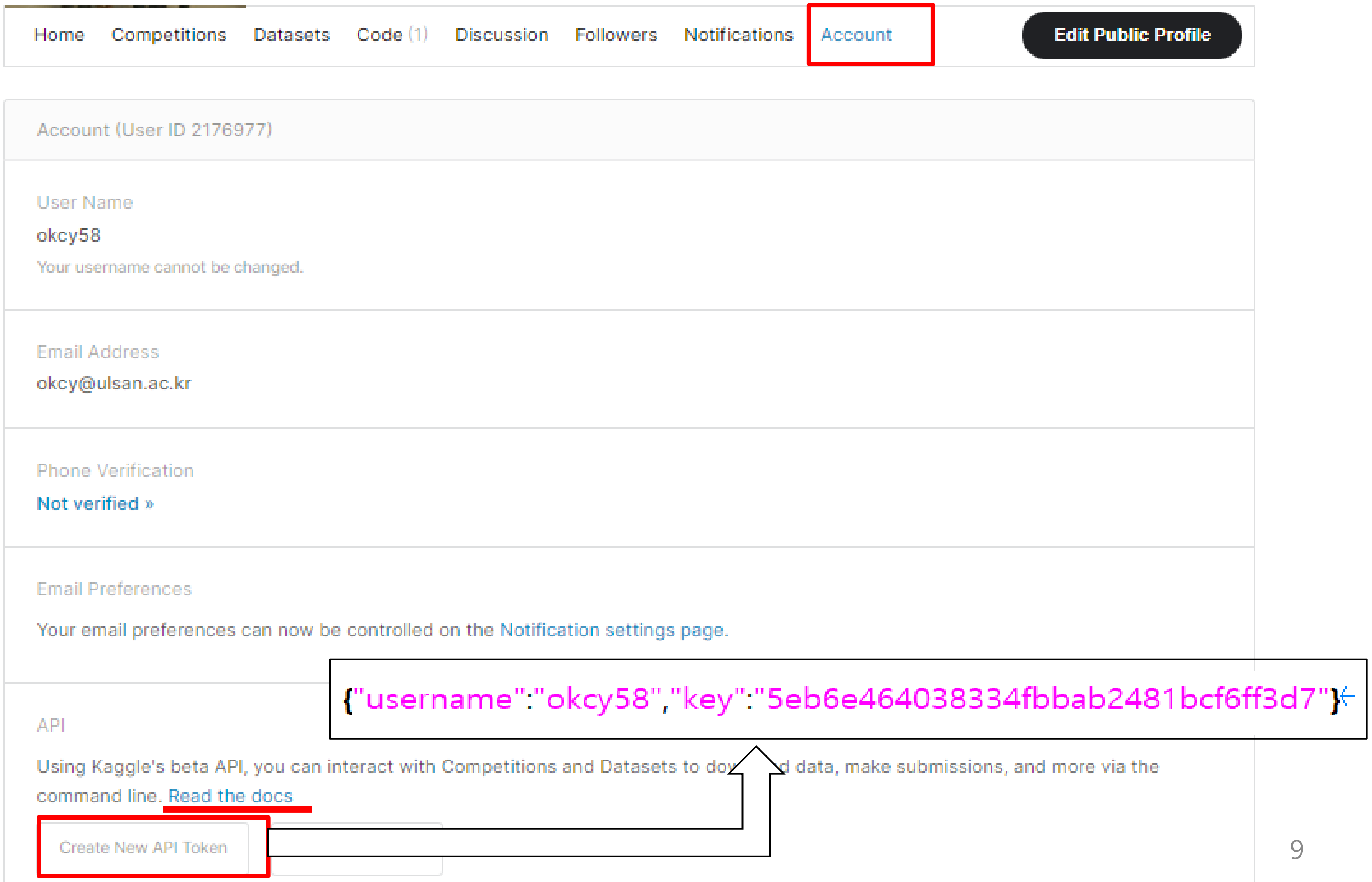
Kaggle API (1)

- https://www.kaggle.com/
  - 등록



Kaggle API (2)

- API 토큰 생성



Account (User ID 2176977)

User Name
okcy58
Your username cannot be changed.

Email Address
okcy@ulsan.ac.kr

Phone Verification
Not verified »

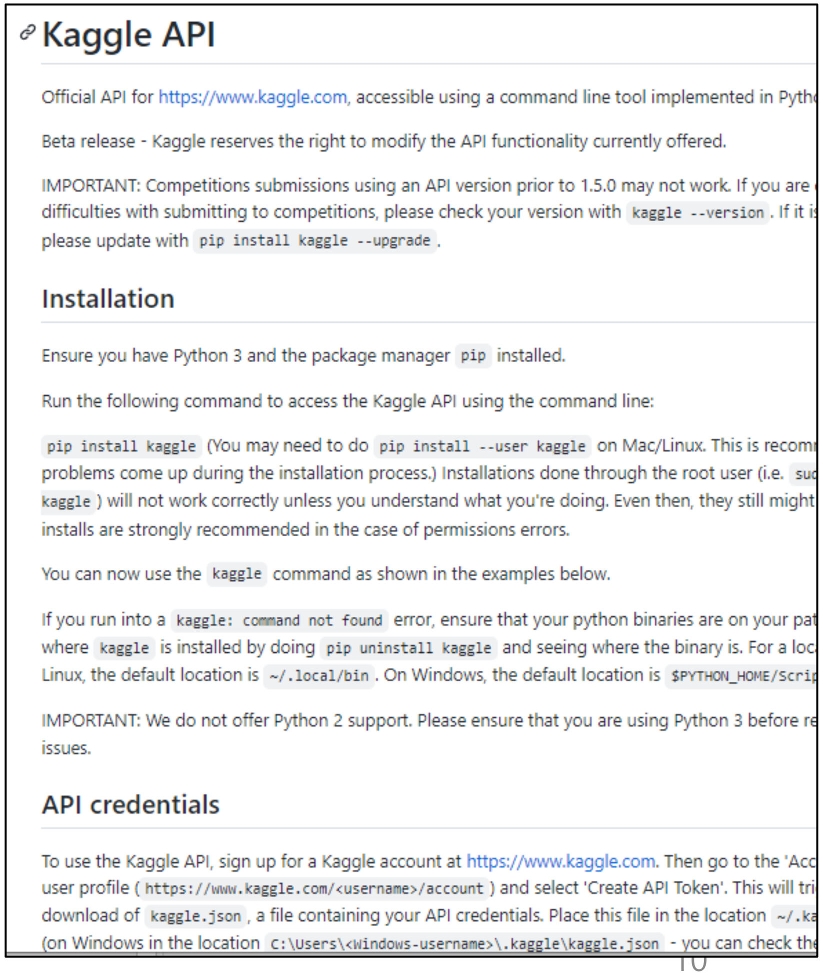
Email Preferences
Your email preferences can now be controlled on the Notification settings page.

API
Using Kaggle's beta API, you can interact with Competitions and Datasets to dow[...]d data, make submissions, and more via the command line. Read the docs

```
{"username":"okcy58","key":"5eb6e464038334fbbab2481bcf6ff3d7"}
```

Kaggle API (3)

❖ Kaggle 라이브러리 설치
- https://github.com/Kaggle/kaggle-api
- pip install Kaggle
- API credentials
  - C:₩Users₩okcy₩.kaggle₩kaggle.json



Kaggle API

Official API for https://www.kaggle.com, accessible using a command line tool implemented in Pytho

Beta release - Kaggle reserves the right to modify the API functionality currently offered.

IMPORTANT: Competitions submissions using an API version prior to 1.5.0 may not work. If you are
difficulties with submitting to competitions, please check your version with `kaggle --version`. If it i
please update with `pip install kaggle --upgrade`.

Installation

Ensure you have Python 3 and the package manager `pip` installed.

Run the following command to access the Kaggle API using the command line:

`pip install kaggle` (You may need to do `pip install --user kaggle` on Mac/Linux. This is recomm
problems come up during the installation process.) Installations done through the root user (i.e. `su`
`kaggle`) will not work correctly unless you understand what you're doing. Even then, they still might
installs are strongly recommended in the case of permissions errors.

You can now use the `kaggle` command as shown in the examples below.

If you run into a `kaggle: command not found` error, ensure that your python binaries are on your pat
where `kaggle` is installed by doing `pip uninstall kaggle` and seeing where the binary is. For a loc
Linux, the default location is `~/.local/bin`. On Windows, the default location is `$PYTHON_HOME/Scrip`

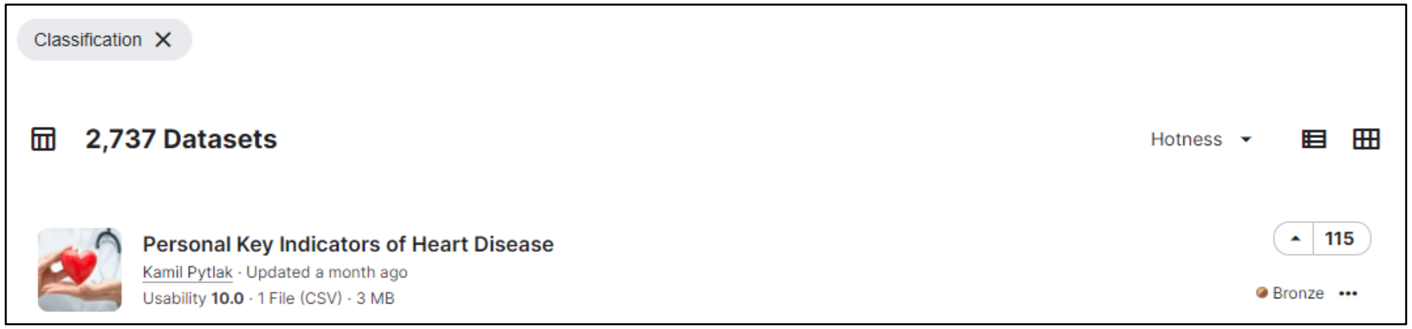
IMPORTANT: We do not offer Python 2 support. Please ensure that you are using Python 3 before re
issues.

API credentials

To use the Kaggle API, sign up for a Kaggle account at https://www.kaggle.com. Then go to the 'Acc
user profile (`https://www.kaggle.com/<username>/account`) and select 'Create API Token'. This will tri
download of `kaggle.json`, a file containing your API credentials. Place this file in the location `~/.ka`
(on Windows in the location `C:\Users\<Windows-username>\.kaggle\kaggle.json` - you can check the

Kaggle API (4)

- Copy API command



Classification ✕

2,737 Datasets

Hotness

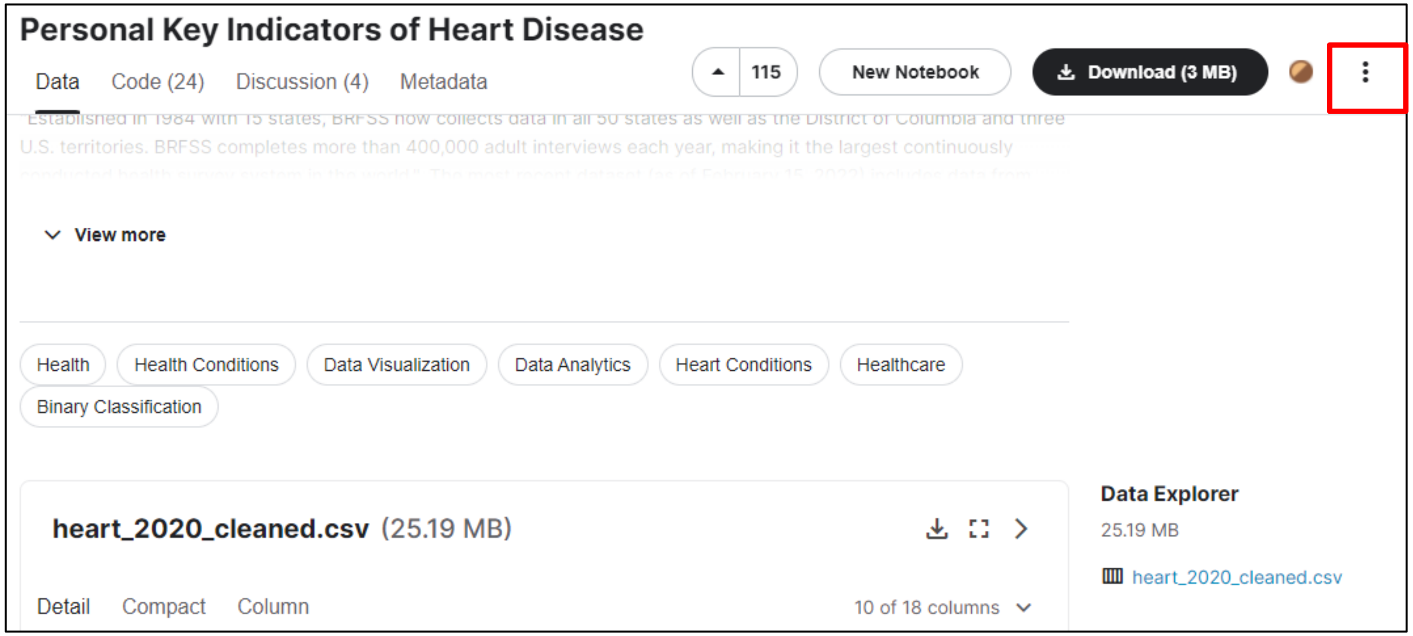
Personal Key Indicators of Heart Disease

Kamil Pytlak · Updated a month ago

Usability 10.0 · 1 File (CSV) · 3 MB

115

Bronze •••



Personal Key Indicators of Heart Disease

Data   Code (24)   Discussion (4)   Metadata

115   New Notebook   Download (3 MB)

Established in 1984 with 15 states, BRFSS now collects data in all 50 states as well as the District of Columbia and three U.S. territories. BRFSS completes more than 400,000 adult interviews each year, making it the largest continuously

View more

Health   Health Conditions   Data Visualization   Data Analytics   Heart Conditions   Healthcare   Binary Classification

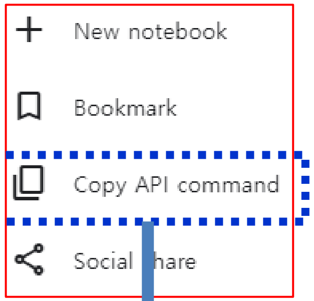
heart_2020_cleaned.csv (25.19 MB)

Detail   Compact   Column   10 of 18 columns

Data Explorer

25.19 MB

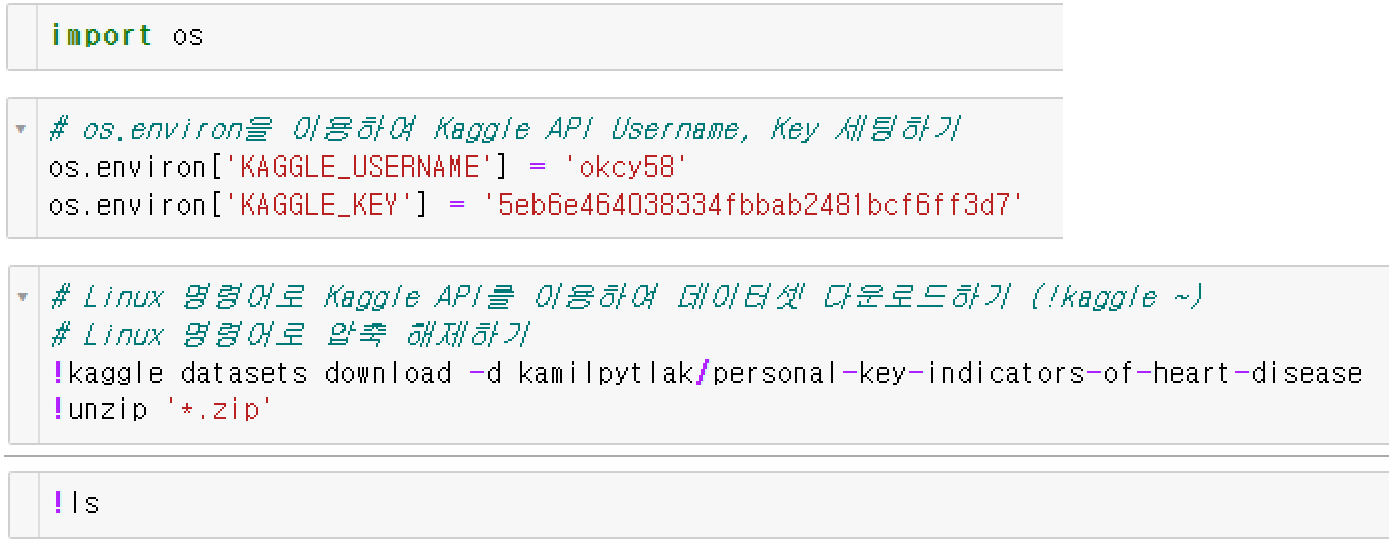
heart_2020_cleaned.csv



```
kaggle datasets download -d kamilpytlak/personal-key-indicators-of-heart-disease
```

Kaggle API (5)

- Colab Notebook에 Kaggle API 세팅하기



```
import os
```

```
# os.environ을 이용하여 Kaggle API Username, Key 세팅하기
os.environ['KAGGLE_USERNAME'] = 'okcy58'
os.environ['KAGGLE_KEY'] = '5eb6e464038334fbbab2481bcf6ff3d7'
```

```
# Linux 명령어로 Kaggle API를 이용하여 데이터셋 다운로드하기 (!kaggle ~)
# Linux 명령어로 압축 해제하기
!kaggle datasets download -d kamilpytlak/personal-key-indicators-of-heart-disease
!unzip '*.zip'
```

```
!ls
```