# Streamlit 모듈 설치하기

pip install streamlit

# 앱 폴더 및 파일 만들기

mkdir apps

cd apps

touch app.py

# Streamlit 모듈 import 하고 실행하기

import streamlit as st

# Streamlit 실행하기(터미널에 입력)

streamlit run app.py

# 텍스트 출력

## Title, Header & Subheader

Header와 Subheader를 다음과 같이 달 수 있다. 다만 Header의 경우 subheader 레벨까지만 가능하다. Title, Header, Subheader이 각각 Header, Subheader, Subsubheader인 것으로 인식하면 될 것 같다.

import streamlit as st

## Title

st.title("Streamlit Tutorial")

## Header/Subheader

st.header("This is header")

st.subheader("This is subheader")

## Text

st.text("Hello Streamlit! 이 글은 튜토리얼 입니다.")

## Markdown

Streamlit도 Dash 와 마찬가지로 Markdown을 지원한다.

import streamlit as st

## Markdown syntax

st.markdown("# This is a Markdown title")

st.markdown("## This is a Markdown header")

st.markdown("### This is a Markdown subheader")

st.markdown("- item 1\n"

" - item 1.1\n"

" - item 1.2\n"

"- item 2\n"

"- item 3")

st.markdown("1. item 1\n"

" 1. item 1.1\n"

" 2. item 1.2\n"

"2. item 2\n"

"3. item 3")

## Latex

Latex의 경우 백슬래시(\)를 빈번히 사용되기 때문에, 일반 string 대신 raw string을 붙여주는 편이 좋다.

import streamlit as st

## Latex

st.latex(r"Y = \alpha + \beta X\_i")

## Latex-inline

st.markdown(r"회귀분석에서 잔차식은 다음과 같습니다 $e\_i = y\_i — \hat{y}\_i$")

# 메세지와 에러메세지, 예외처리 메시지

기본적으로 포맷된 메세지 박스 기능을 제공한다.

import streamlit as st

## Error/message text

st.success("Successful")

st.info("Information!")

st.warning("This is a warning")

st.error("This is an error!")

st.exception("NameError('Error name is not defined')")

# 데이터 로딩

## Load data

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

# GitHub에서 아이리스 데이터 다운로드

url = "https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv"

iris\_df = pd.read\_csv(url)

# 데이터프레임과 테이블 출력

데이터를 출력하는 방법에는 3가지 방법이 있다.

* st.table: 입력 테이블 전체를 리턴한다.
* st.dataframe: 적절히 10개의 행을 기준으로 스크롤을 통해 데이터를 관찰 할 수 있고 각 열마다 정렬도 가능하다. 각 테이블의 우측 상단의 확대 버튼을 통해 테이블을 더 크게 볼 수 있다.
* st.write: st.dataframe과 똑같은 결과를 리턴한다.

## Load data

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

# GitHub에서 아이리스 데이터 다운로드

url = "https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv"

iris\_df = pd.read\_csv(url)

## Return table/dataframe

# table

st.table(iris\_df.head())

# dataframe

st.dataframe(iris\_df)

st.write(iris\_df)

st.markdown("\* \* \*")

# 위젯

## st.checkbox – 체크박스

import streamlit as st

## Checkbox

if st.checkbox("Show/Hide"):

st.write("체크박스가 선택되었습니다.")

## st.radio – 라디오버튼

import streamlit as st

## Radio button

status = st.radio("Select status.", ("Active", "Inactive"))

if status == "Active":

st.success("활성화 되었습니다.")

else:

st.warning("비활성화 되었습니다.")

## st.selectbox - 드랍다운 선택

import streamlit as st

## Select Box

occupation = st.selectbox("직군을 선택하세요.", [

"Backend Developer",

"Frontend Developer",

"ML Engineer",

"Engineer",

"Database Administrator",

"Scientist",

"Data Analyst",

"Security Engineer"

])

st.write(" 직군은 ", occupation, " 입니다.")

## st.multiselect - 드랍다운 다중 선택

import streamlit as st

## MultiSelect

location = st.multiselect("선호하는 유투브 채널을 선택하세요.",

(

"운동",

"IT기기",

"브이로그",

"먹방",

"반려동물",

"맛집 리뷰"

)

)

st.write(len(location), "가지를 선택했습니다.")

## st.slider – 슬라이더

import streamlit as st

## Slider

level = st.slider("레벨을 선택하세요.", 1, 5)

## st.button – 버튼

import streamlit as st

## Buttons

if st.button("About"):

st.text("Streamlit을 이용한 튜토리얼입니다.")

## 텍스트 입력

import streamlit as st

# Text Input

first\_name = st.text\_input("Enter Your First Name", "Type Here ...")

if st.button("Submit", key='first\_name'):

result = first\_name.title()

st.success(result)

# Text Area

message = st.text\_area("메세지를 입력하세요.", "Type Here ...")

if st.button("Submit", key='message'):

result = message.title()

st.success(result)

## 날짜와 시간 입력

## Date Input

import streamlit as st

import datetime

today = st.date\_input("날짜를 선택하세요.", datetime.datetime.now())

the\_time = st.time\_input("시간을 입력하세요.", datetime.time())

## 코드와 JSON 출력

with st.echo(): 이하의 코드는 코드블럭으로 출력된다.

import streamlit as st

## Display Raw Code — one line

st.subheader("Display one-line code")

st.code("import numpy as np")

# Display Raw Code — snippet

st.subheader("Display code snippet")

with st.echo():

# 여기서부터 아래의 코드를 출력합니다.

import pandas as pd

df = pd.DataFrame()

## Display JSON

st.subheader("Display JSON")

st.json({"name" : "민수", "gender": "male", "Age": 29})

## 사이드바

st.sidebar에서도 대부분의 위젯을 지원하므로, 다양하게 사이드바를 구성할 수 있다. (단, st.echo, st.spinner, st.write제외)

import streamlit as st

## Sidebars

st.sidebar.header("사이드바 메뉴")

st.sidebar.selectbox("메뉴를 선택하세요.",

["데이터",

"EDA",

"코드"])

## 차트 그리기

Streamlit은 자체 내장된 기본적인 차트 외 matplotlib, plot.ly, altair, vega\_ilte, bokeh, deck\_gl, pydeck, graph\_viz 등 다양한 시각화 패키지를 지원한다.

(Streamlit은 EDA 용도로 많이 사용되는 만큼, 시각화 부분은 따로 다룰 계획이다.)

## Plotting

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

st.subheader("Matplotlib으로 차트 그리기")

# GitHub에서 아이리스 데이터 다운로드

url = "https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv"

iris\_df = pd.read\_csv(url)

## Plotting

st.subheader("Matplotlib으로 차트 그리기")

fig, ax = plt.subplots()

ax = iris\_df[iris\_df['species']=='virginica']['petal\_length'].hist()

st.pyplot(fig)

## LAYOUT

import streamlit as st

st.title("Registeration form")

first, last = st.columns(2)

first.text\_input("First Name")

last.text\_input("Last Name")

email, mob = st.columns([3,1])

email.text\_input("Email ID")

mob.text\_input("Mob Number")

user, pw, pw2 = st.columns(3)

user.text\_input("Username")

pw.text\_input("Password", type="password")

pw2.text\_input("Retype your password", type="password")

ch, bl, sub = st.columns(3)

ch.checkbox("I Agree")

sub.button("Submit")

# 캐싱

import streamlit as st

import pandas as pd

import time

@st.cache\_data

def fetch\_and\_clean\_data(url):

# Fetch data from URL here, and then clean it up.

data = pd.read\_csv(url)

return data

DATA\_URL\_1 = "https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv"

DATA\_URL\_2 = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/raw/titanic.csv"

# 시작 시간 기록

start\_time = time.time()

d1 = fetch\_and\_clean\_data(DATA\_URL\_1)

# Actually executes the function, since this is the first time it was

# encountered.

# 종료 시간 기록

end\_time = time.time()

# 실행 시간 계산

execution\_time = end\_time - start\_time

# 실행 시간 출력

st.write(f"총 실행 시간: {execution\_time:.6f} 초")

st.write(d1.head())

# 시작 시간 기록

start\_time = time.time()

d2 = fetch\_and\_clean\_data(DATA\_URL\_1)

# Does not execute the function. Instead, returns its previously computed

# value. This means that now the data in d1 is the same as in d2.

# 종료 시간 기록

end\_time = time.time()

# 실행 시간 계산

execution\_time = end\_time - start\_time

# 실행 시간 출력

st.write(f"총 실행 시간: {execution\_time:.6f} 초")

st.write(d2.head())

# 시작 시간 기록

start\_time = time.time()

d3 = fetch\_and\_clean\_data(DATA\_URL\_2)

# This is a different URL, so the function executes.

# 종료 시간 기록

end\_time = time.time()

# 실행 시간 계산

execution\_time = end\_time - start\_time

# 실행 시간 출력

st.write(f"총 실행 시간: {execution\_time:.6f} 초")

st.write(d3.head())

# 세션

import streamlit as st

st.title('Counter Example')

# Streamlit runs from top to bottom on every iteraction so

# we check if `count` has already been initialized in st.session\_state.

# If no, then initialize count to 0

# If count is already initialized, don't do anything

if 'count' not in st.session\_state:

st.session\_state.count = 0

# Create a button which will increment the counter

increment = st.button('Increment')

if increment:

st.session\_state.count += 1

# A button to decrement the counter

decrement = st.button('Decrement')

if decrement:

st.session\_state.count -= 1

st.write('Count = ', st.session\_state.count)

# 파일 업로드

import streamlit as st

uploaded\_files = st.file\_uploader("Choose a CSV file", accept\_multiple\_files=True)

for uploaded\_file in uploaded\_files:

bytes\_data = uploaded\_file.read()

st.write("filename:", uploaded\_file.name)

st.write(bytes\_data)

# 예제 - 데이터 프로파일링

import platform

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

from matplotlib import font\_manager, rc

plt.rcParams['axes.unicode\_minus'] = False

if platform.system() == 'Linux':

rc('font', family='NanumGothic')

# 스트림릿 앱 생성

st.title("데이터 프로파일링 실습")

# 파일 업로드 위젯

uploaded\_file = st.file\_uploader("데이터 파일 업로드", type=["csv", "xlsx"])

if uploaded\_file is not None:

# 업로드한 파일을 DataFrame으로 변환

df = pd.read\_csv(uploaded\_file) # 엑셀 파일일 경우 pd.read\_excel 사용

# 데이터 프로파일링

st.header("데이터 미리보기")

st.write(df.head())

st.header("기본 정보")

st.write("행 수:", df.shape[0])

st.write("열 수:", df.shape[1])

st.header("누락된 값")

missing\_data = df.isnull().sum()

st.write(missing\_data)

st.header("중복된 행 수")

duplicated\_rows = df.duplicated().sum()

st.write(duplicated\_rows)

st.header("수치형 데이터 기술 통계량")

numerical\_stats = df.describe()

st.write(numerical\_stats)

st.header("이상치 탐지 (상자 그림)")

plt.figure(figsize=(10, 6))

plt.boxplot(df.select\_dtypes(include=['number']).values)

plt.xticks(range(1, len(df.columns) + 1), df.columns, rotation=45)

plt.title("Outlier detection (box plot)")

st.pyplot(plt)

st.header("데이터 분포 시각화")

column\_to\_plot = st.selectbox("열 선택", df.columns)

plt.figure(figsize=(10, 6))

plt.hist(df[column\_to\_plot], bins=20, edgecolor='k')

plt.xlabel(column\_to\_plot)

plt.ylabel("빈도")

plt.title(f"{column\_to\_plot} Data Distribution")

st.pyplot(plt)

# 예제 - IRIS 데이터 시각화

* CASE 01

import streamlit as st

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

st.set\_option('deprecation.showPyplotGlobalUse', False)

# 아이리스 데이터셋 불러오기

@st.cache\_data

def load\_data():

# GitHub에서 아이리스 데이터 다운로드

url = "https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv"

iris\_df = pd.read\_csv(url)

return iris\_df

iris\_data = load\_data()

# 스트림릿 앱 제목 설정

st.title('아이리스 데이터 시각화')

# 데이터프레임 출력

st.subheader('아이리스 데이터셋')

st.write(iris\_data)

# 품종별 특성 분포 시각화

st.subheader('품종별 특성 분포')

for feature in iris\_data.columns[:-1]:

plt.figure(figsize=(8, 6))

sns.boxplot(x='species', y=feature, data=iris\_data)

plt.title(f'{feature} Distribution')

plt.xlabel('species')

plt.ylabel(feature)

st.pyplot()

# 특성 간 상관 관계 시각화

st.subheader('특성 간 상관 관계')

correlation\_matrix = iris\_data.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

st.pyplot()

# 품종별 특성 산점도 시각화

st.subheader('품종별 특성 산점도')

sns.pairplot(iris\_data, hue='species', diag\_kind='kde')

st.pyplot()

* CASE 02

# 스트림릿 앱에서 Matplotlib의 그림(figure)을 스트림릿에 전달할 때 발생하는 PyplotGlobalUseWarning 경고를 방지하려면 아래와 같이 코드를 수정할 수 있습니다.

import streamlit as st

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# 아이리스 데이터셋 불러오기

# 아이리스 데이터셋 불러오기

@st.cache\_data

def load\_data():

# GitHub에서 아이리스 데이터 다운로드

url = "https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv"

iris\_df = pd.read\_csv(url)

return iris\_df

iris\_data = load\_data()

# 스트림릿 앱 제목 설정

st.title('아이리스 데이터 시각화')

# 데이터프레임 출력

st.subheader('아이리스 데이터셋')

st.write(iris\_data)

# 품종별 특성 분포 시각화

st.subheader('품종별 특성 분포')

for feature in iris\_data.columns[:-1]:

plt.figure(figsize=(8, 6))

sns.boxplot(x='species', y=feature, data=iris\_data)

plt.title(f'{feature} Distribution')

plt.xlabel('species')

plt.ylabel(feature)

st.pyplot(plt)

# 특성 간 상관 관계 시각화

st.subheader('특성 간 상관 관계')

correlation\_matrix = iris\_data.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

ax = sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

st.pyplot(plt)

# 품종별 특성 산점도 시각화

st.subheader('품종별 특성 산점도')

sns.pairplot(iris\_data, hue='species', diag\_kind='kde')

sns.pairplot(iris\_data, hue='species', diag\_kind='kde')

st.pyplot(plt)

# 기타 코드

## 세션 값 테스트 코드

# Sessions Don’t Store Values Forever

# Storing Value

# Associate Session State With Input Widgets

# Sharing Value Between Reruns

# Delete Values From Session State

# rerun 세션 값 유지 테스트 코드

import streamlit as st

input\_var = st.text\_input("enter a name")

st.write(f"Hello, {input\_var}!")

if ("name" not in st.session\_state) and (input\_var != ""):

st.session\_state["name"] = input\_var

st.write("first name you have entered: ")

if "name" in st.session\_state:

st.write(st.session\_state["name"])

st.write(st.session\_state)

# input widget 세션 값 변경 테스트 코드

import streamlit as st

input\_select = st.selectbox("select a food",

options= ("sushi", "steak","burger"),

key="food")

st.write(st.session\_state)