

웹 애플리케이션

# Steamlit

류영표 강사

ryp1662@gmail.com

Copyright © “Youngpyo Ryu” All Rights Reserved.

This document was created for the exclusive use of “Youngpyo Ryu”.

It must not be passed on to third parties except with the explicit prior consent of “Youngpyo Ryu”.



# 류영표

Youngpyo Ryu

동국대학교 수학과/응용수학 석사수료

前 Upstage AI X 네이버 부스트 캠프 AI tech 1~ 6기 멘토

前 Innovation on Quantum & CT(IQCT) 이사

前 한국파스퇴르연구소 Image Mining 인턴(Deep learning)

前 (주)셈웨어(수학컨텐츠, 데이터 분석 개발 및 연구인턴)

## 강의 경력

- 현대자동차 연구원 강의 (인공지능/머신러닝/딥러닝/강화학습)
- (주) 모두의연구소 Aiffel 1기 퍼실리테이터(인공지능 교육)
- 인공지능 자연어처리(NLP) 기업데이터 분석 전문가 양성과정 멘토
- 공공데이터 청년 인턴 / SW공개개발자대회 멘토
- 고려대학교 선도대학 소속 30명 딥러닝 집중 강의
- 이젠 종로 아카데미(파이썬, ADSP 강사)
- 최적화된 도구(R/파이썬)을 활용한 애널리스트 양성과정(국비과정) 강사
- 한화, 하나금융사, 한전 KDN, IBK 기업은행 교육
- 인공지능 신뢰성 확보를 위한 실무 전문가 자문 활동
- 인공지능 학습용 데이터 구축 사업 품질검증 전문가 자문 활동
- 보건 · 바이오 AI활용 S/W개발 및 응용전문가 양성과정 강사
- Upstage AI X KT 융합기술원 기업교육 모델최적화 담당 조교

## 주요 프로젝트 및 기타사항

- 의료 영상을 이용한 갑상선암 연관 유전변이 예측 플랫폼 개발 고도화
- 개인 맞춤형 당뇨병 예방·관리 인공지능 시스템 개발 및 고도화(안정화)
- 폐플라스틱 이미지 객체 검출 경진대회 3위
- 인공지능(AI)기반 데이터 사이언티스트 전문가 양성과정 1기 수료
- 제 1회 산업 수학 스터디 그룹 (질병에 영향을 미치는 유전자 정보 분석)
- 제 4,5회 산업 수학 스터디 그룹 (피부암, 유방암 분류)



# Streamlit

- Python으로 작성된 코드만으로도 웹 애플리케이션을 쉽게 만들 수 있는 오픈 소스 프레임워크.
- 데이터 과학자, 엔지니어, 연구자들이 데이터 애플리케이션을 손쉽게 구축하고 공유할 수 있도록 설계됨.
- 장점 : 복잡한 웹 개발 지식 없이도 직관적인 방식으로 애플리케이션을 만들 수 있는 점.
- 코드 한줄로 인터랙티브한 웹 인터페이스 구축 / Pandas, Matplotlib, Plotly 등과 손쉬운 통합
- 설치 명령어 : `pip install streamlit`



# Streamlit

- 간편성 : 복잡한 프론트엔드 지식이 없어도 웹 앱을 만들 수 있습니다.
- 반응형 : 자동으로 업데이트되는 위젯을 제공하여 데이터와 시각화의 상호작용을 쉽게 구현할 수 있음.
- 데이터 통합 : 주요 데이터 분석 및 시각화 라이브러리(ex. Pandas, Matplotlib, Plotly)와의 통합이 용이.
- 빠른 프로토타이핑 : 원하는 대로 앱의 모양과 기능을 빠르게 바꿔 볼 수 있어 프로토타이핑에 이상적.

\* 프로토타이핑 : 본격적인 제품 개발에 들어가기 전에 시제품이나 견본품을 제작하고 검증하는 과정.

# Steamlit 단점

## 1. 낮은 확장성 (Scalability)

- 단일 서버 기반 → 사용자 수가 많아질수록 성능 저하 가능성

## 2.. 보안 기능 부족 (Security)

- 기본 보안 미지원 → 민감한 데이터 저장 시 별도 보안 조치 필요

## 3. 커스터마이징 한계 (Customization)




- 빠른 개발은 가능하지만, 세세한 UI 커스터마이징은 제약 있음

## 4. 통합성 부족 (Integration)

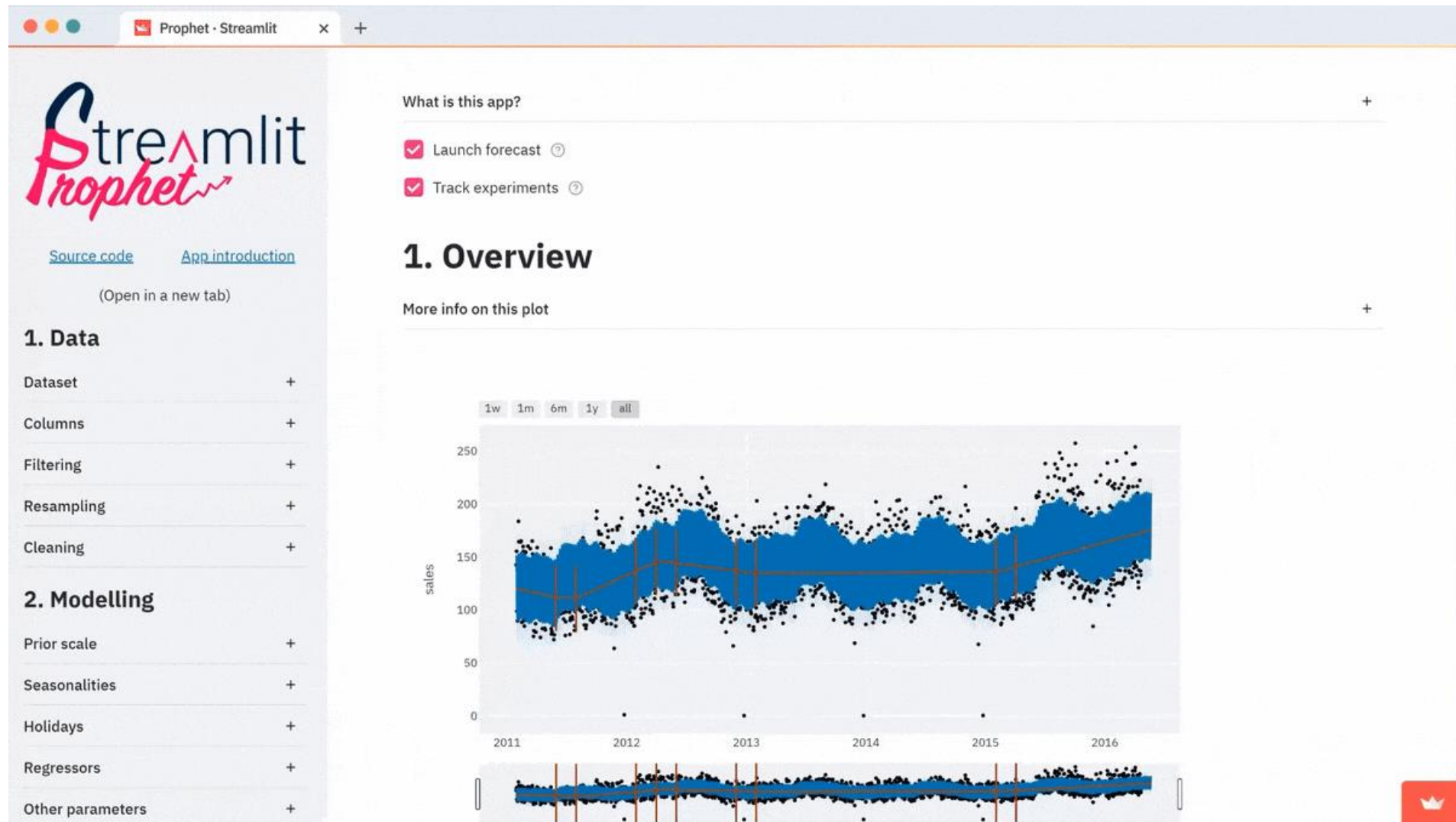
- 외부 라이브러리나 프레임워크와 통합이 다소 어려움

## 5. 생태계 미성숙 (Immature Ecosystem)

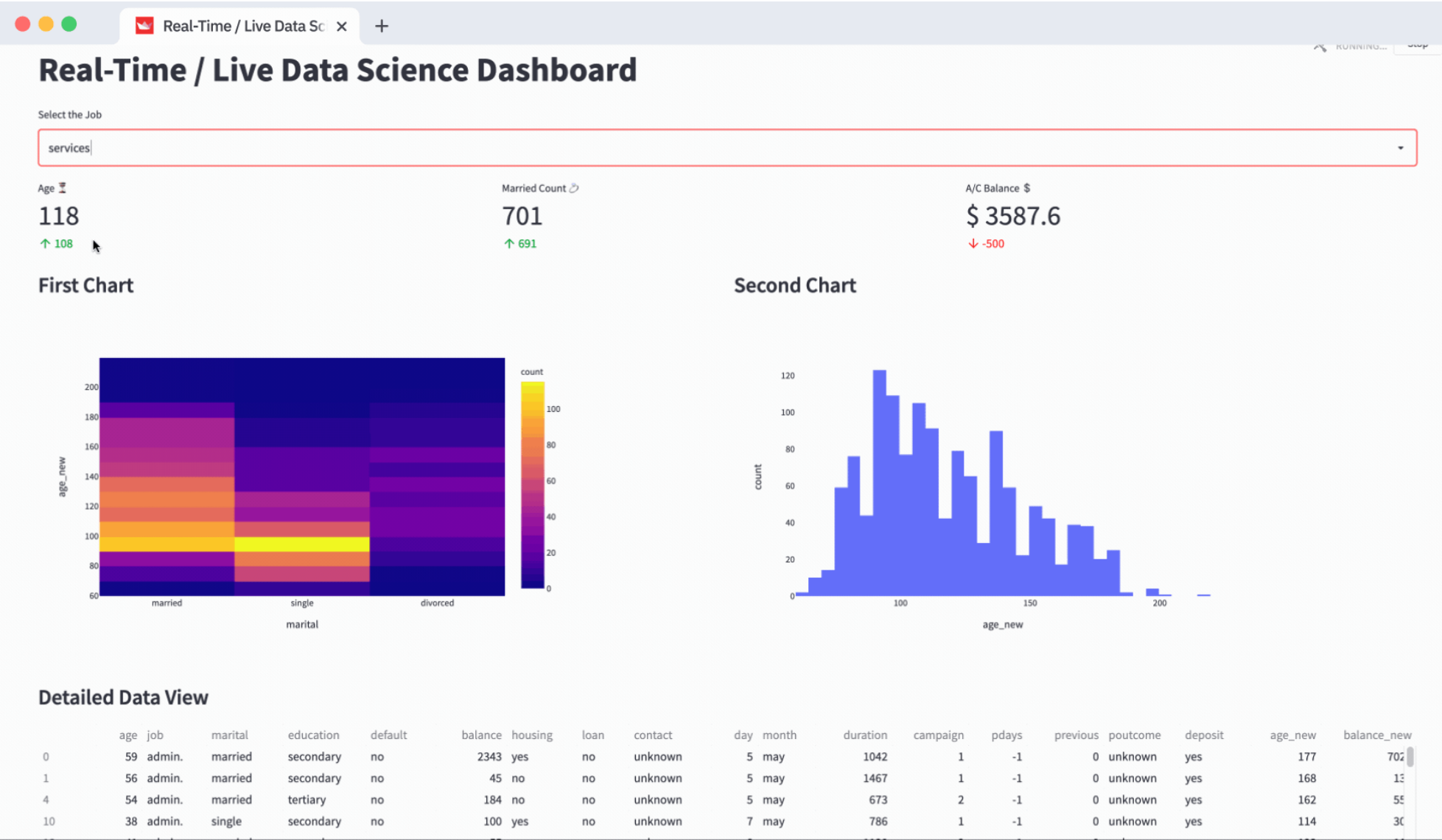
- 비교적 신생 프레임워크 → 일부 패키지/리소스 부족

구분	 Streamlit	 django	 Flask
사용 용이성	높음 (최소한의 코딩)	중간 (높은 학습 곡선)	중간 (웹 개발 이해 필요)
웹 개발 지식 필요 여부	X	O	O
완비된 툴 제공	X (데이터 과학 및 머신 러닝 응용 프로그램에 초점)	O (ORM, 관리자 패널, 인증 지원 등 포함)	X (최소주의 디자인, 라이브러리로 확장 가능)
유연성	중간 (복잡한 웹 애플리케이션 부적합)	높음 (복잡한 웹 애플리케이션 적합)	높음 (원하는 도구와 라이브러리 선택가능)
단순한 웹 애플리케이션에 적합	O	X (과잉)	O

# Steamlit\_활용사례









# Streamlit\_시작하기


```
PS C:\Users\FirstZone 01\Documents> python -m streamlit hello

Welcome to Streamlit. Check out our demo in your browser.


Local URL: http://localhost:8506
Network URL: http://192.168.1.58:8506


Ready to create your own Python apps super quickly?
Head over to https://docs.streamlit.io


May you create awesome apps!
```

 Hello

 DataFrame demo

 Plotting demo

 Mapping demo

 Animation demo

## Welcome to Streamlit! 🙌

Streamlit is an open-source app framework built specifically for machine learning and data science projects. 📁 Select a demo from the sidebar to see some examples of what Streamlit can do!

### Want to learn more?

- Check out [streamlit.io](https://streamlit.io)
- Jump into our [documentation](https://docs.streamlit.io)
- Ask a question in our [community forums](https://discuss.streamlit.io)

### See more complex demos

- Use a neural net to [analyze the Udacity Self-driving Car Image Dataset](#)
- Explore a [New York City rideshare dataset](#)

# Streamlit\_시작하기

```
Main.py X
Main.py
1 import streamlit as st
2 st.title("Streamlit Introduction")
3 st.write("Hello, world!")
```

```
C:\Users\FirstZone 01\Documents>python -m streamlit run Main.py
```

Welcome to Streamlit!

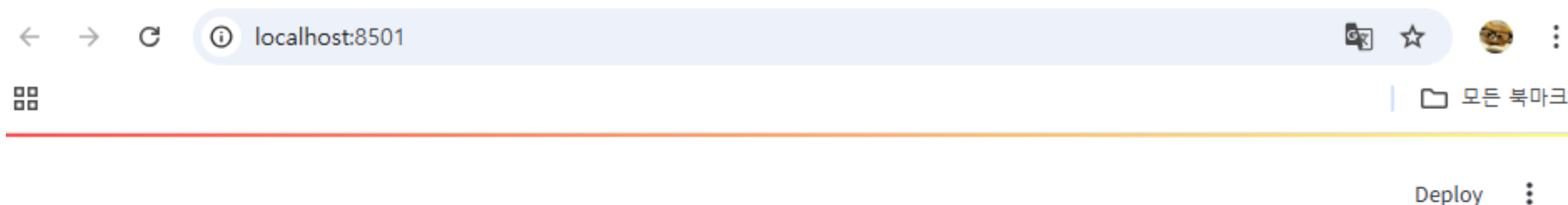
If you'd like to receive helpful onboarding emails, news, offers, promotions, and the occasional swag, please enter your email address below. Otherwise, leave this field blank.

Email: ryp1662@gmail.com

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.1.58:8501>



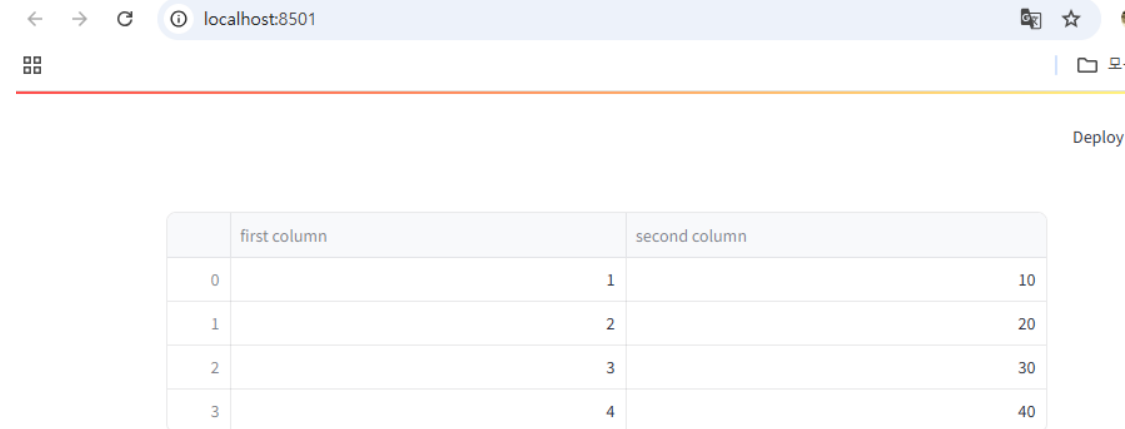
## Streamlit Introduction

Hello, world!

# Streamlit

- 데이터를 표시하는 다양한 방법
  - Streamlit 메서드를 호출하지 않고도 'df' 만 입력해도 앱에 표시할 수 있음.
    - Streamlit이 자체 라인에서 변수나 리터럴 값을 볼 때마다 st.write()를 사용하여 이를 자동으로 앱에 쓰기 때문.

```
"""  
# My first app  
Here's our first attempt at using data to create a table:  
"""  
  
import streamlit as st  
import pandas as pd  
df = pd.DataFrame({  
    'first column': [1, 2, 3, 4],  
    'second column': [10, 20, 30, 40]  
})  
  
df # st.write(df) 와 동일하게 작동함.
```



	first column	second column
0	1	10
1	2	20
2	3	30
3	4	40

# Streamlit

- 데이터를 표시하는 다양한 방법
  - 텍스트, 데이터, Matplotlib 등 거의 모든 것을 st.write()에 전달할 수 있음.
  - Streamlit이 이를 파악하고 올바른 방식으로 렌더링

```
import streamlit as st
import pandas as pd

st.write("Here's our first attempt at using data to create a table:")
st.write(pd.DataFrame({
    'first column': [1, 2, 3, 4],
    'second column': [10, 20, 30, 40]
}))

# df = pd.DataFrame({
#     'first column': [1, 2, 3, 4],
#     'second column': [10, 20, 30, 40]
# })

#st.dataframe(df) # 대화형 테이블
#st.table(df) # 정적 테이블
```

- 렌더링(Rendering): 데이터를 사용자에게 보여줄 수 있는 형태로 변환하여 웹 브라우저에서 시각적으로 표현하는 과정을 의미함.

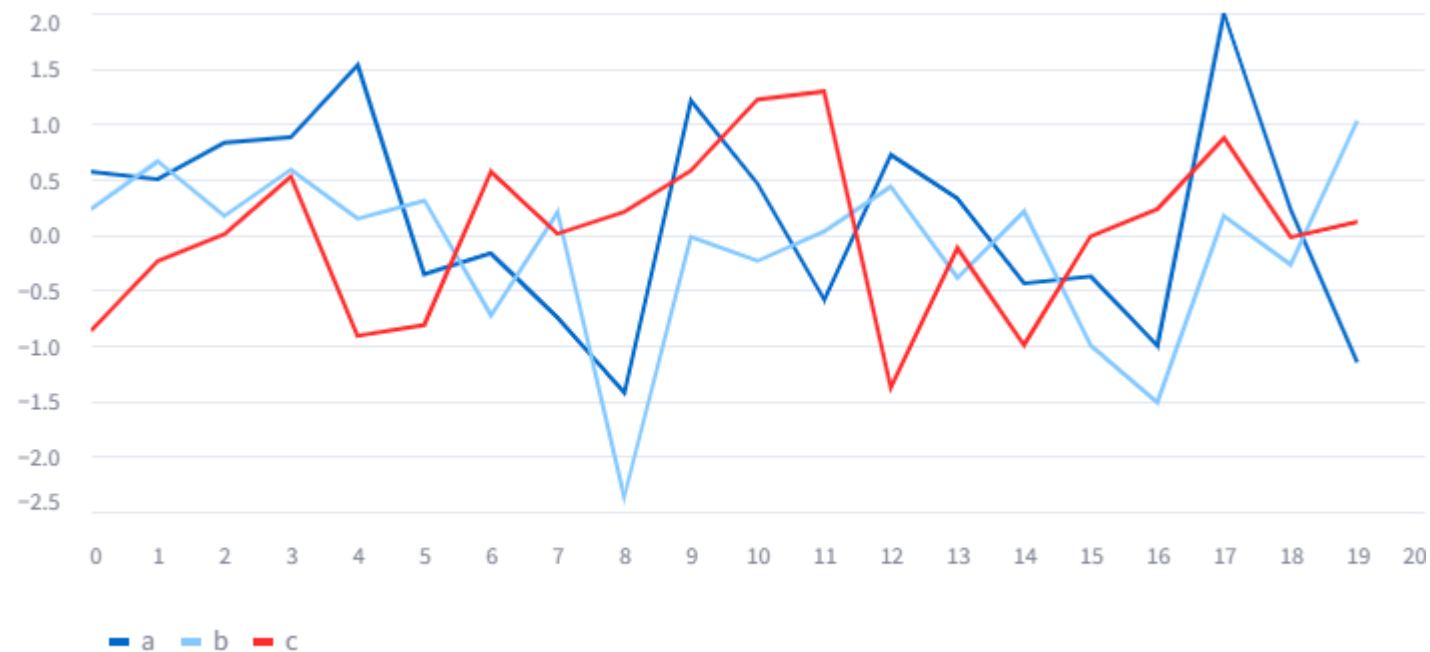
# Streamlit

- 차트와 지도 그리기
  - Line chart

```
import streamlit as st
import numpy as np
import pandas as pd

chart_data = pd.DataFrame(
    np.random.randn(20, 3),
    columns=['a', 'b', 'c'])

st.line_chart(chart_data)
```



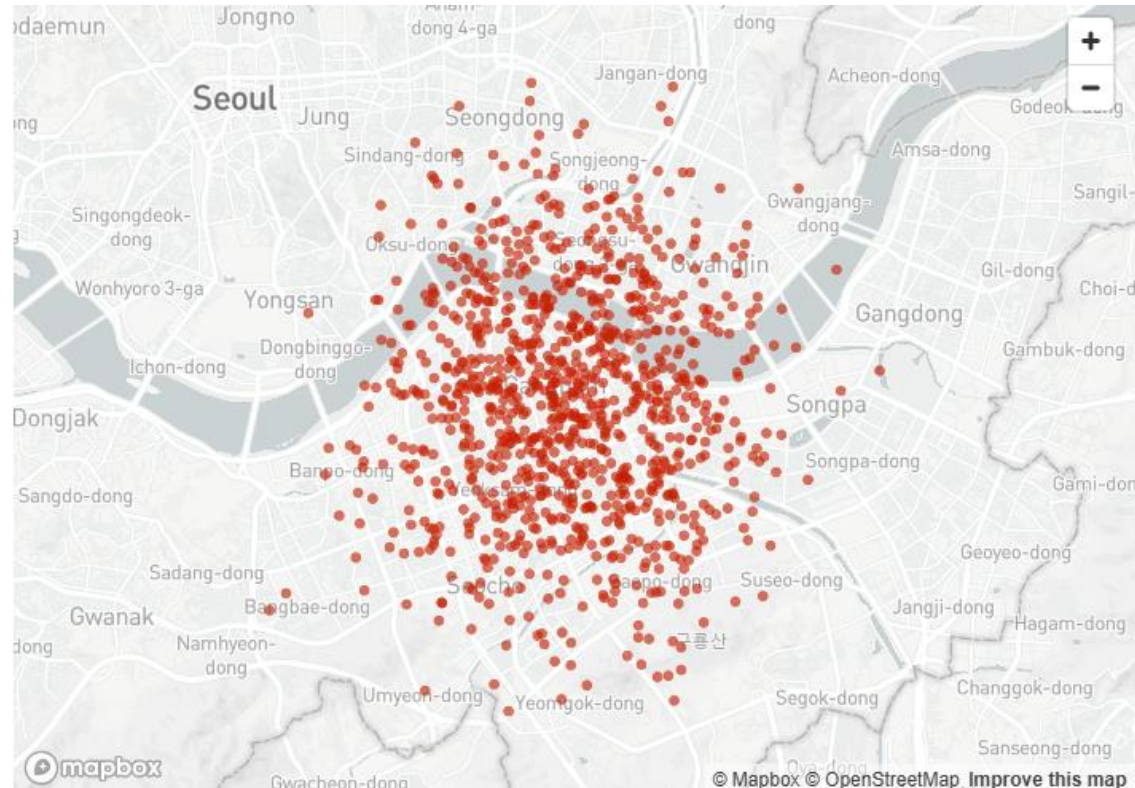
# Streamlit

- 차트와 지도 그리기
  - 지도(샘플 데이터 생성 후 강남 지도 그리기)

```
import streamlit as st
import numpy as np
import pandas as pd

map_data = pd.DataFrame(
    np.random.randn(1000, 2) / [50, 50] + [37.514575, 127.0495556],
    columns=['lat', 'lon'])

st.map(map_data)
```



- 위젯
  - 데이터나 모델을 탐색하려는 상태로 설정한 경우 즉, 사용자의 입력을 받을 경우 `st.slider()`, `st.button()`, `st.selectbox()` 같은 위젯 추가 가능.

```
import streamlit as st
x = st.slider('x') # 📌 this is a widget
st.write(x, 'squared is', x * x)
```

- 처음 실행 시 위 앱은 "0 squared is 0"이라는 텍스트 표시.
- 사용자가 위젯과 상호 작용할 때마다 Streamlit은 단순히 스크립트를 위에서 아래로 다시 실행하여 위젯의 현재 상태를 프로세스의 변수에 할당.
- 예를 들어, 사용자가 슬라이더를 위치 10으로 이동하면 Streamlit은 위의 코드를 다시 실행하고 이에 따라 `x`를 10으로 설정.
- 앱 화면도 "10 squared is 100"이라는 텍스트 변경됨.



# Streamlit

- 체크박스를 사용하여 데이터 표시 / 숨기기
  - 체크박스를 이용해 앱에서 특정 차트나 섹션을 숨기거나 표시할 수 있음.
  - `st.checkbox()` 위젯 사용.
  - Checkbox에 check가 되어 있을 때는 조건문 실행, check가 해제되어 있을 때는 조건문 실행하지 않음.

```
import streamlit as st
import numpy as np
import pandas as pd

if st.checkbox('Show dataframe'):
    chart_data = pd.DataFrame(
        np.random.randn(20, 3),
        columns=['a', 'b', 'c'])

    chart_data
```

☐ Show dataframe

# Streamlit

- option을 위해 select box 사용
  - 선택지 안에서 선택하게 하려면 st.selectbox 사용

```
import streamlit as st
import pandas as pd

option = st.selectbox(
    'Which number do you like best?',
    [1, 2, 3, 4, 5])

'You selected: ', option
```

Which number do you like best?

1

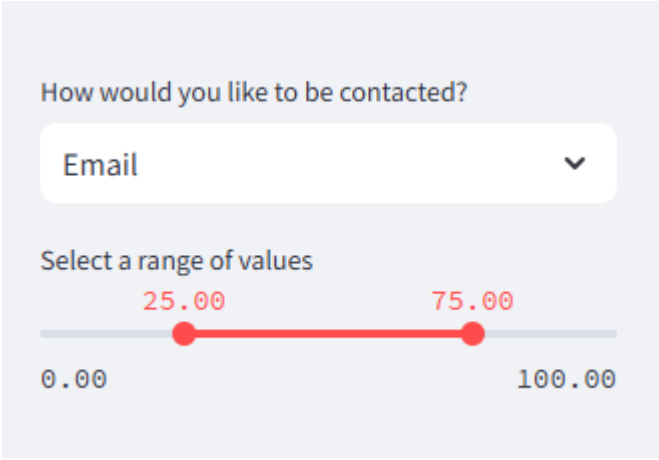
You selected: 1

- Layout
  - ‘st.sidebar’를 사용하여 왼쪽 패널 사이드바에서 위젯을 쉽게 구성할 수 있음.
  - ‘st.sidebar’에 전달된 각 요소는 왼쪽에 고정되어 사용자가 UI 컨트롤에 계속 액세스하면서 앱 콘텐츠를 집중할 수 있음.
  - 예를 들어, 사이드바에 select box와 slider를 추가하려면 ‘st.slider’ 및 ‘st.selectbox’ 대신 ‘st.sidebar.slider’ 및 ‘st.sidebar.selectbox’ 사용.

```
import streamlit as st

# Add a selectbox to the sidebar:
add_selectbox = st.sidebar.selectbox(
    'How would you like to be contacted?',
    ('Email', 'Home phone', 'Mobile phone')
)

# Add a slider to the sidebar:
add_slider = st.sidebar.slider(
    'Select a range of values',
    0.0, 100.0, (25.0, 75.0)
)
```



The screenshot displays a sidebar with two interactive widgets. The top widget is a selectbox titled "How would you like to be contacted?" with a dropdown menu currently showing "Email". The bottom widget is a slider titled "Select a range of values" with a range from 0.00 to 100.00. The slider has red markers and labels at 25.00 and 75.00, indicating the selected range.

# Streamlit

- Layout
  - ‘st.sidebar’를 사용하여 왼쪽 패널 사이드바에서 위젯을 쉽게 구성할 수 있음.
  - ‘st.sidebar’에 전달된 각 요소는 왼쪽에 고정되어 사용자가 UI 컨트롤에 계속 액세스하면서 앱 콘텐츠를 집중할 수 있음.
  - 예를 들어, 사이드바에 select box와 slider를 추가하려면 ‘st.slider’ 및 ‘st.selectbox’ 대신 ‘st.sidebar.slider’ 및 ‘st.sidebar.selectbox’ 사용.

```
import streamlit as st

left_column, right_column = st.columns(2)
# You can use a column just like st.sidebar:
left_column.button('Press me!')

# Or even better, call Streamlit functions inside a "with" block:
with right_column:
    chosen = st.radio(
        'Sorting hat',
        ("Gryffindor", "Ravenclaw", "Hufflepuff", "Slytherin"))
    st.write(f"You are in {chosen} house!")
```

Press me!

Sorting hat

- ☒ Gryffindor
- ☐ Ravenclaw
- ☐ Hufflepuff
- ☐ Slytherin

You are in Gryffindor house!

```
# Streamlit 기본 앱 예제
import streamlit as st

# 제목 표시
st.title('나의 첫 Streamlit 앱')

# 간단한 텍스트 출력
st.write('안녕하세요! Streamlit으로 쉽게 웹 앱을 만들 수 있습니다.')

# 버튼 추가 및 버튼 클릭 시 텍스트 출력
if st.button('클릭해 보세요'):
    st.write('버튼이 클릭되었습니다!')

# 슬라이더로 사용자 입력 받기
age = st.slider('나이를 선택하세요:', 0, 100, 25)
st.write('선택한 나이는', age, '세 입니다.')

# 데이터프레임 표시하기
import pandas as pd

data = {'이름': ['철수', '영희', '민수'],
        '나이': [25, 30, 28],
        '성별': ['남', '여', '남']}
df = pd.DataFrame(data)
st.write('데이터프레임:', df)

# 사이드바 추가
st.sidebar.title('사이드바 메뉴')
option = st.sidebar.selectbox('메뉴를 선택하세요', ['홈', '데이터 보기', '설정'])
st.write('선택한 메뉴:', option)
```

## 사이드바 메뉴

메뉴를 선택하세요

홈

▼

## 나의 첫 Streamlit 앱

안녕하세요! Streamlit으로 쉽게 웹 앱을 만들 수 있습니다.

클릭해 보세요

나이를 선택하세요:



선택한 나이는 25 세 입니다.

데이터프레임:

	이름	나이	성별
0	철수	25	남
1	영희	30	여
2	민수	28	남

선택한 메뉴: 홈

```
# Streamlit 기본 앱 예제
import streamlit as st

# 제목 표시
st.title('나의 첫 Streamlit 앱')

# 간단한 텍스트 출력
st.write('안녕하세요! Streamlit으로 쉽게 웹 앱을 만들 수 있습니다.')

# 버튼 추가 및 버튼 클릭 시 텍스트 출력
if st.button('클릭해 보세요'):
    st.write('버튼이 클릭되었습니다!')

# 슬라이더로 사용자 입력 받기
age = st.slider('나이를 선택하세요:', 0, 100, 25)
st.write('선택한 나이는', age, '세 입니다.')

# 데이터프레임 표시하기
import pandas as pd

data = {'이름': ['철수', '영희', '민수'],
        '나이': [25, 30, 28],
        '성별': ['남', '여', '남']}
df = pd.DataFrame(data)
st.write('데이터프레임:', df)

# 사이드바 추가
st.sidebar.title('사이드바 메뉴')
option = st.sidebar.selectbox('메뉴를 선택하세요', ['홈', '데이터 보기', '설정'])
st.write('선택한 메뉴:', option)
```

## 사이드바 메뉴

메뉴를 선택하세요

홈

▼

## 나의 첫 Streamlit 앱

안녕하세요! Streamlit으로 쉽게 웹 앱을 만들 수 있습니다.

클릭해 보세요

나이를 선택하세요:



선택한 나이는 25 세 입니다.

데이터프레임:

	이름	나이	성별
0	철수	25	남
1	영희	30	여
2	민수	28	남

선택한 메뉴: 홈

# Streamlit

```
# Streamlit 체계적으로 배우기 위한 전체 코드 예제

import streamlit as st
import pandas as pd
import numpy as np

# 앱의 타이틀과 설명
st.title('Streamlit 체계적 학습 가이드')
st.write('이 앱을 통해 Streamlit의 기본부터 다양한 기능까지 체계적으로 배울 수 있습니다.')

# 섹션 1: 기본 텍스트 및 타이틀 표시
st.header('1. 기본 텍스트 및 타이틀 표시하기')
st.subheader('부제목 표시하기')
st.write('일반 텍스트는 st.write로 간단히 표시합니다.')

# 섹션 2: 인터랙티브 버튼과 체크박스
st.header('2. 인터랙티브 버튼과 체크박스 사용하기')
if st.button('클릭하세요'):
    st.success('버튼을 클릭했습니다!')

check = st.checkbox('체크박스 선택하기')
if check:
    st.info('체크박스가 선택되었습니다.')

# 섹션 3: 사용자 입력 위젯 (슬라이더, 선택박스 등)
st.header('3. 사용자 입력 위젯 활용하기')
age = st.slider('나이 선택:', 0, 100, 25)
color = st.selectbox('좋아하는 색상을 고르세요:', ['빨강', '파랑', '초록'])
st.write(f'나이는 {age}세, 좋아하는 색상은 {color}입니다.')

# 섹션 4: 데이터프레임과 차트 시각화하기
st.header('4. 데이터프레임과 차트 시각화하기')
data = pd.DataFrame(
    np.random.randn(20, 3),
    columns=['A', 'B', 'C']
)
st.write('데이터프레임 예시:', data)
st.line_chart(data)
```

```
# 섹션 4: 데이터프레임과 차트 시각화하기
st.header('4. 데이터프레임과 차트 시각화하기')
data = pd.DataFrame(
    np.random.randn(20, 3),
    columns=['A', 'B', 'C']
)
st.write('데이터프레임 예시:', data)
st.line_chart(data)

# 섹션 5: 사이드바 및 레이아웃 관리하기
st.header('5. 사이드바 및 레이아웃 관리하기')
st.sidebar.title('사이드바 메뉴')
menu = st.sidebar.radio('메뉴 선택', ['홈', '데이터 분석', '설정'])
st.write('선택한 메뉴:', menu)

col1, col2 = st.columns(2)
with col1:
    st.write('왼쪽 컬럼 내용입니다.')
with col2:
    st.write('오른쪽 컬럼 내용입니다.')

# 섹션 6: 파일 업로드와 이미지 표시
st.header('6. 파일 업로드와 이미지 표시하기')
uploaded_file = st.file_uploader('파일 업로드')
if uploaded_file:
    st.image(uploaded_file)

# 섹션 7: 고급 예제 (세션 상태 유지)
st.header('7. 고급 예제: 세션 상태 유지하기')

if 'counter' not in st.session_state:
    st.session_state.counter = 0

increment = st.button('카운터 증가')
if increment:
    st.session_state.counter += 1

st.write('현재 카운터 값:', st.session_state.counter)
```



## Streamlit 체계적 학습 가이드 ↻

이 앱을 통해 Streamlit의 기본부터 다양한 기능까지 체계적으로 배울 수 있습니다.

### 1. 기본 텍스트 및 타이틀 표시하기

#### 부제목 표시하기

일반 텍스트는 `st.write`로 간단히 표시합니다.

### 2. 인터랙티브 버튼과 체크박스 사용하기

클릭하세요

☐ 체크박스 선택하기

### 3. 사용자 입력 위젯 활용하기

나이 선택:



좋아하는 색상을 고르세요:

빨강 ▼

나이는 25세, 좋아하는 색상은 빨강입니다.

## 4. 데이터프레임과 차트 시각화하기

데이터프레임 예시:

	A	B	C
0	-0.6606	-0.3601	-0.4202
1	-0.2441	-0.4177	1.7431
2	1.1774	0.3331	-1.0509
3	-0.1848	-0.0316	-0.1385
4	-0.0354	1.2214	0.5148
5	-0.2513	0.1431	0.7552
6	-0.6595	1.2669	-0.0492
7	-0.351	1.2538	-0.7046
8	0.445	0.4906	-0.336
9	-0.6386	0.5761	-0.5356



## 5. 사이드바 및 레이아웃 관리

선택한 메뉴: 홈

왼쪽 컬럼 내용입니다.

오른쪽 컬럼 내용입니다.

## 6. 파일 업로드와 이미지 표시하기

파일 업로드



Drag and drop file here

Limit 200MB per file

Browse files

## 7. 고급 예제: 세션 상태 유지하기

카운터 증가

현재 카운터 값: 0

# Streamlit

```
# 고급 Streamlit 앱 예제 (머신러닝 모델을 활용한 데이터 분석 앱)
```

```
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# 앱 타이틀
st.title('고급 Streamlit 앱: 머신러닝 데이터 분석')
```

```
# 데이터 로딩
@st.cache_data
def load_data():
    iris = load_iris()
    df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
    df['species'] = pd.Categorical.from_codes(iris.target, iris.target_names)
    return df
```

```
df = load_data()
st.write('아이리스 데이터셋:', df.head())
```

```
# 사용자 입력 위젯
st.sidebar.header('모델 파라미터 설정')
n_estimators = st.sidebar.slider('나무의 개수 (n_estimators):', min_value=10, max_value=200, value=50)
max_depth = st.sidebar.slider('나무의 최대 깊이 (max_depth):', min_value=1, max_value=20, value=5)
test_size = st.sidebar.slider('테스트 데이터 비율:', min_value=0.1, max_value=0.5, value=0.2)
```

```
# 데이터 분리
X = df.iloc[:, :-1]
y = df['species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=42)
```

```
# 모델 학습
model = RandomForestClassifier(n_estimators=n_estimators, max_depth=max_depth, random_state=42)
model.fit(X_train, y_train)
```

```
# 예측 및 평가
predictions = model.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
cm = confusion_matrix(y_test, predictions, labels=model.classes_)
```

```
# 결과 표시
st.subheader('모델 성능 평가')
st.write(f'정확도(Accuracy): {accuracy:.2f}')
```

```
fig, ax = plt.subplots()
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=model.classes_, yticklabels=model.classes_, ax=ax)
plt.ylabel('실제 값')
plt.xlabel('예측 값')
st.pyplot(fig)
```

```
# 피쳐 중요도 시각화
st.subheader('피쳐 중요도')
importances = model.feature_importances_
feature_names = X.columns
fig, ax = plt.subplots()
sns.barplot(x=importances, y=feature_names, ax=ax)
plt.xlabel('중요도')
st.pyplot(fig)
```

```
# 인터랙티브 예측
st.subheader('새로운 데이터로 예측하기')
sepal_length = st.number_input('꽃받침 길이(sepal length, cm)', min_value=0.0, max_value=10.0, value=5.1)
sepal_width = st.number_input('꽃받침 너비(sepal width, cm)', min_value=0.0, max_value=10.0, value=3.5)
petal_length = st.number_input('꽃잎 길이(petal length, cm)', min_value=0.0, max_value=10.0, value=1.4)
petal_width = st.number_input('꽃잎 너비(petal width, cm)', min_value=0.0, max_value=10.0, value=0.2)
```

```
if st.button('예측하기'):
    new_data = np.array([[sepal_length, sepal_width, petal_length, petal_width]])
    pred = model.predict(new_data)
    st.write('예측된 품종은:', pred[0])
```

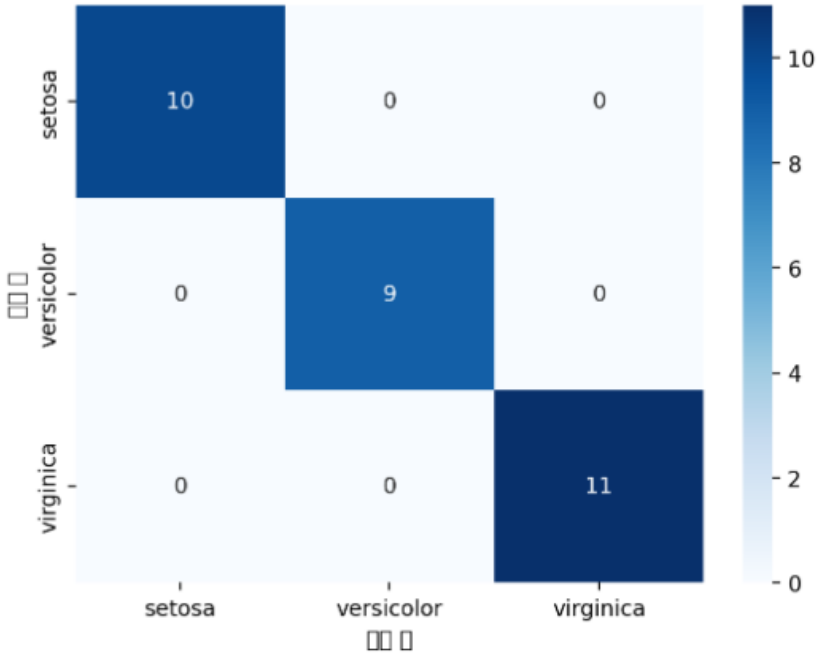
## 고급 Streamlit 앱: 머신러닝 데이터 분석

아이리스 데이터셋:

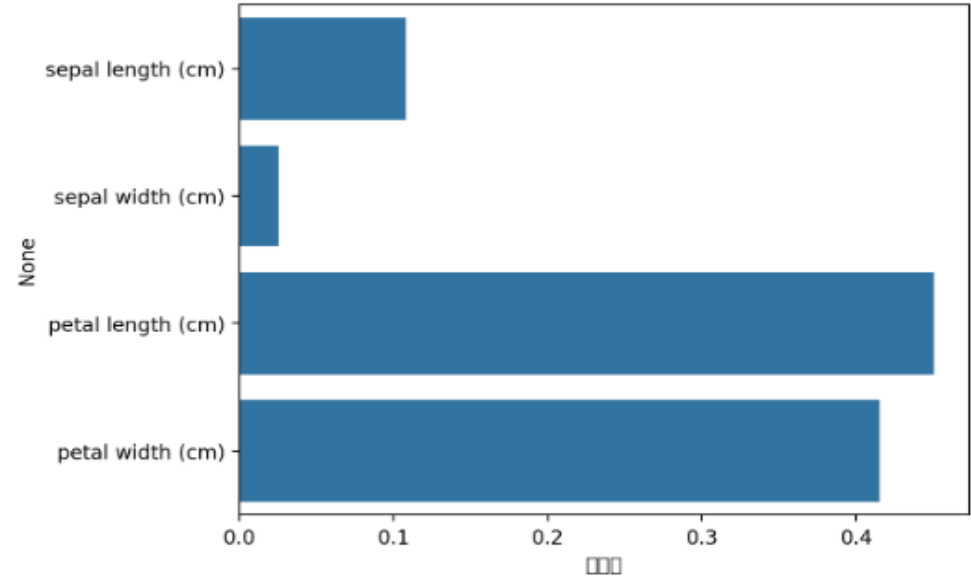
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5	3.6	1.4	0.2	setosa

## 모델 성능 평가

정확도(Accuracy): 1.00



## 피쳐 중요도



## 새로운 데이터로 예측하기

꽃받침 길이(sepal length, cm)

5.10 - +

꽃받침 너비(sepal width, cm)

3.50 - +

꽃잎 길이(petal length, cm)

1.40 - +

꽃잎 너비(petal width, cm)

0.20 - +

예측하기

```
import streamlit as st
import pandas as pd
import numpy as np

st.title('Uber pickups in NYC')

DATE_COLUMN = 'date/time'
DATA_URL = ('https://s3-us-west-2.amazonaws.com/'
            'streamlit-demo-data/uber-raw-data-sep14.csv.gz')

@st.cache
def load_data(nrows):
    data = pd.read_csv(DATA_URL, nrows=nrows)
    lowercase = lambda x: str(x).lower()
    data.rename(lowercase, axis='columns', inplace=True)
    data[DATE_COLUMN] = pd.to_datetime(data[DATE_COLUMN])
    return data

data_load_state = st.text('Loading data...')
data = load_data(10000)
data_load_state.text("Done! (using st.cache)")

if st.checkbox('Show raw data'):
    st.subheader('Raw data')
    st.write(data)

st.subheader('Number of pickups by hour')
hist_values = np.histogram(data[DATE_COLUMN].dt.hour, bins=24, range=(0,24))[0]
st.bar_chart(hist_values)

hour_to_filter = st.slider('hour', 0, 23, 17)
filtered_data = data[data[DATE_COLUMN].dt.hour == hour_to_filter]

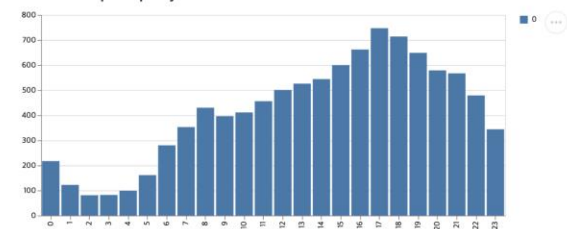
st.subheader('Map of all pickups at %s:00' % hour_to_filter)
st.map(filtered_data)
```

## Uber pickups in NYC

Done! (using st.cache)

☐ Show raw data

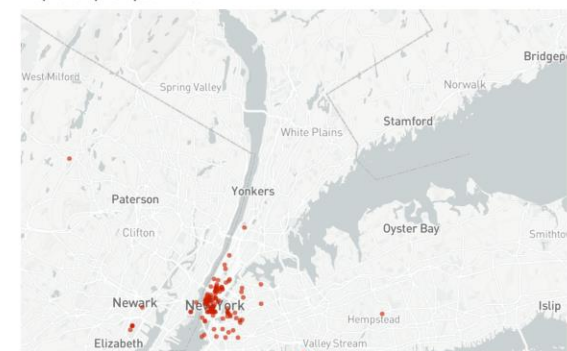
Number of pickups by hour



hour



Map of all pickups at 1:00



# Streamlit

```
import streamlit as st
import pandas as pd
import numpy as np

# 앱 타이틀
st.title('Streamlit 고급 기능 실습')
st.write('다양한 Streamlit 기능을 삼도 있게 배우기 위한 앱입니다.')

# 섹션 1: 인터랙티브 위젯 활용
st.header('1. 인터랙티브 위젯 활용하기')
name = st.text_input('이름을 입력하세요:')
age = st.number_input('나이를 입력하세요:', min_value=0, max_value=120)
if st.button('입력 완료'):
    st.success(f'{name}님의 나이는 {age}세 입니다.')

# 섹션 2: 고급 레이아웃 구성
st.header('2. 고급 레이아웃 구성하기')
col1, col2, col3 = st.columns(3)
with col1:
    st.subheader('컬럼 1')
    st.write('첫 번째 컬럼입니다.')
with col2:
    st.subheader('컬럼 2')
    st.write('두 번째 컬럼입니다.')
with col3:
    st.subheader('컬럼 3')
    st.write('세 번째 컬럼입니다.')

# 섹션 3: 다양한 시각화 차트
st.header('3. 다양한 데이터 시각화하기')
chart_data = pd.DataFrame(
    np.random.randn(20, 3),
    columns=['A', 'B', 'C']
)
st.line_chart(chart_data)
st.bar_chart(chart_data)

# 섹션 4: 파일 처리 및 다운로드
st.header('4. 파일 업로드 및 다운로드하기')
uploaded_file = st.file_uploader('파일 업로드', type=['csv', 'xlsx'])
if uploaded_file:
    if uploaded_file.name.endswith('.csv'):
        df = pd.read_csv(uploaded_file)
    else:
        df = pd.read_excel(uploaded_file)
    st.dataframe(df)

    csv = df.to_csv(index=False).encode('utf-8')
    st.download_button(
        label='CSV로 다운로드',
        data=csv,
        file_name='data.csv',
        mime='text/csv',
    )
```

```
# 섹션 5: 세션 상태와 고급 상태 관리
st.header('5. 세션 상태 및 고급 상태 관리')
if 'counter' not in st.session_state:
    st.session_state['counter'] = 0

increment = st.button('카운터 증가')
reset = st.button('카운터 리셋')

if increment:
    st.session_state['counter'] += 1
if reset:
    st.session_state['counter'] = 0

st.write('현재 카운터 값:', st.session_state['counter'])

# 섹션 6: 페이지 네비게이션
st.header('6. 멀티페이지 네비게이션 예시')
page = st.sidebar.radio('페이지 선택', ['홈', '분석', '시각화'])

if page == '홈':
    st.write('홈 화면입니다.')
elif page == '분석':
    st.write('데이터 분석 페이지입니다.')
elif page == '시각화':
    st.write('시각화 페이지입니다.')

# 섹션 7: 멀티미디어 활용
st.header('7. 멀티미디어 요소 추가')
st.image('https://streamlit.io/images/brand/streamlit-logo-primary-colormark-darktext.png', width=300)
st.audio('https://www.soundhelix.com/examples/mp3/SoundHelix-Song-1.mp3')

# 섹션 8: 캐싱 및 성능 최적화
st.header('8. 데이터 캐싱 및 성능 최적화')
@st.cache_data
def expensive_computation(n):
    return sum(i**2 for i in range(n))

num = st.slider('계산할 숫자 범위 선택:', 10000, 1000000, step=10000)
result = expensive_computation(num)
st.write(f'결과: {result}')
```



# Thank you.

Steamlit  
ryp1662@gmail.com

Copyright © “Youngpyo Ryu” All Rights Reserved.  
This document was created for the exclusive use of “Youngpyo Ryu”.  
It must not be passed on to third parties except with the explicit prior consent of “Youngpyo Ryu”.