

Project Deployment using Streamlit



Outline

- Introduction to Streamlit and its use for building web apps
- Overview of the Streamlit app structure and how it relates to the previously built ML model
- Installing necessary packages and loading the previously trained model
- Building a simple Streamlit app that takes user input, feeds it into the model, and displays the predicted output

Introduction to Streamlit and its use for building web apps

"A dashboard is a **visual display** of the most **important information** needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be **monitored at a glance.**"

– Stephen Few

"A dashboard is a **visual display** of data used to **monitor conditions** and/or **facilitate understanding.**"

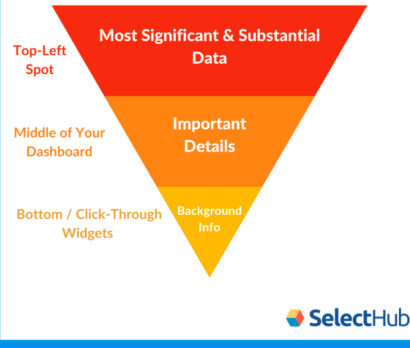
– Big Book of Dashboards



Aturan Umum

1

Inverted Pyramid for Dashboard Design



Inverted pyramid

Prinsip ini merekomendasikan untuk meletakkan item yang paling penting dan merupakan big picture di paling atas dan detail yang menyertainya disusun ke bawah

2



Z-pattern

Prinsip ini didasarkan pada kebiasaan membaca dari mayoritas orang yang membaca dari kiri ke kanan



“Streamlit turns data scripts into shareable web apps in minutes”

1. Build an app in a few lines of code with simple API.
2. Adding a widget is the same as declaring a variable.
3. Used in the world's top data science groups.
4. Compatible with many python library.



Monitoring apps

Tools that display trends and real-time insights.



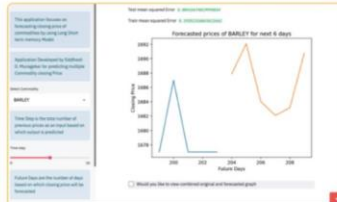
Analysis tools

Apps that use machine learning to analyze new data types.



Detection apps

Tools that use computer vision to detect and track objects.



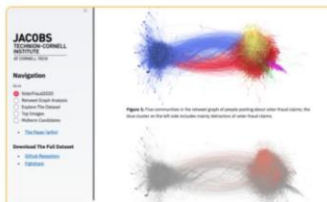
Prediction tools

Apps that predict prices, stock shortages, quality issues, etc.



Explanatory apps

Apps that analyze large datasets and present easy to understand insights.



Interactive tools

Tools that allow you to interact with the data to gain new insights.

Ini memungkinkan kita membuat data apps (termasuk dashboard) menggunakan Python dengan cepat dan penulisan script yang relatif sangat mudah.

Selain itu, user dapat menambahkan fungsionalitas yang dibuat oleh orang lain yang dinamakan components (klik [di sini](#) untuk melihat components)

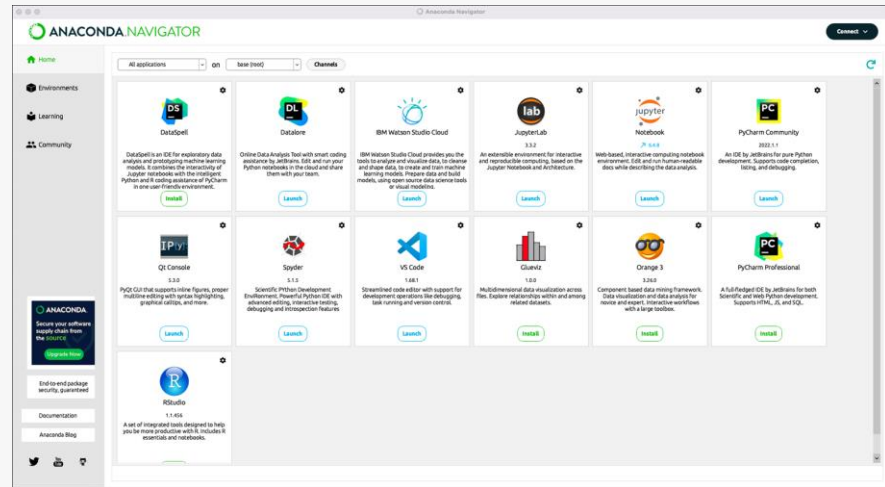
Untuk melihat contoh data apps atau dashboard dapat dilihat [di sini](#)

First of the all

Install (Anaconda)

Download anaconda [here](#)

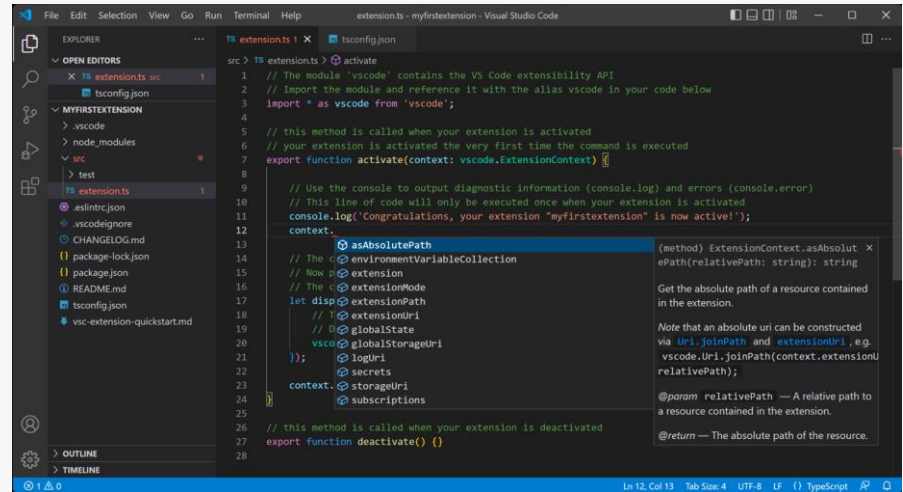
select based on your operating system



Install (VS Code)

Download VS Code [here](#)

select based on your operating system



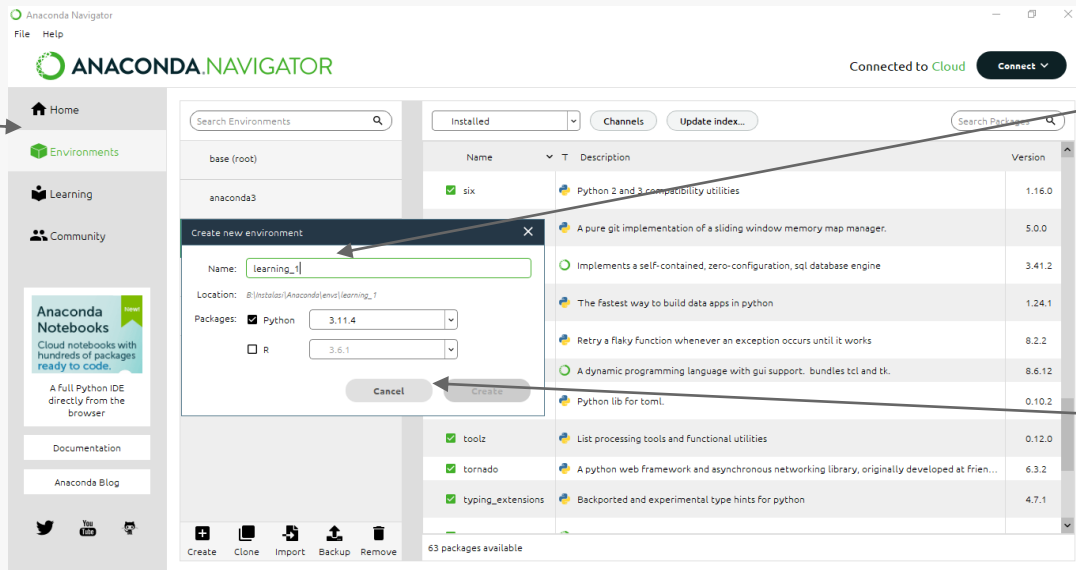
The screenshot shows the Visual Studio Code interface with a project named 'extension.ts - myfirstextension'. The Explorer sidebar on the left shows the file structure: 'src' folder containing 'extension.ts', 'tsconfig.json', and 'test'. The 'src' folder is expanded, showing 'extension.ts' selected. The main editor area displays the content of 'extension.ts', which is a TypeScript file for a VS Code extension. The code includes comments and a function 'activate' that logs a message to the console. A tooltip is visible over the 'asAbsolutePath' method call, providing details about its signature and usage.

```
src > ts extensions.ts > activate
1 // The module "vscode" contains the VS Code extensibility API
2 // Import the module and reference it with the alias vscode in your code below
3 import * as vscode from 'vscode';
4
5 // this method is called when your extension is activated
6 // your extension is activated the very first time the command is executed
7 export function activate(context: vscode.ExtensionContext) {
8
9     // Use the console to output diagnostic information (console.log) and errors (console.error)
10    // This line of code will only be executed once when your extension is activated
11    console.log('Congratulations, your extension "myfirstextension" is now active!');
12
13    context.
14        .asAbsolutePath
15        (method) ExtensionContext.asAbsolutePath X
16        ePath(relativePath: string): string
17
18    Get the absolute path of a resource contained
19    in the extension.
20
21    Note that an absolute uri can be constructed
22    via Uri.joinPath and extensionUri, e.g.
23    vscode.Uri.joinPath(context.extensionUri,
24    relativePath);
25
26    @param relativePath — A relative path to
27    a resource contained in the extension.
28
29    @return — The absolute path of the resource.
30
31    // The c:\environmentVariableCollection
32    // Now p\extension
33    // The c:\extensionMode
34    let disp\extensionPath
35    // !\extensionUri
36    // !\globalState
37    vscode\globalStorageUri
38    });
39    logUri
40    secrets
41    context.\storageUri
42    subscriptions
43
44    // this method is called when your extension is deactivated
45    export function deactivate() {}
46
47
48
```

Preparing virtual environment on Anaconda Navigator

Click on environment

1



2

Click Create

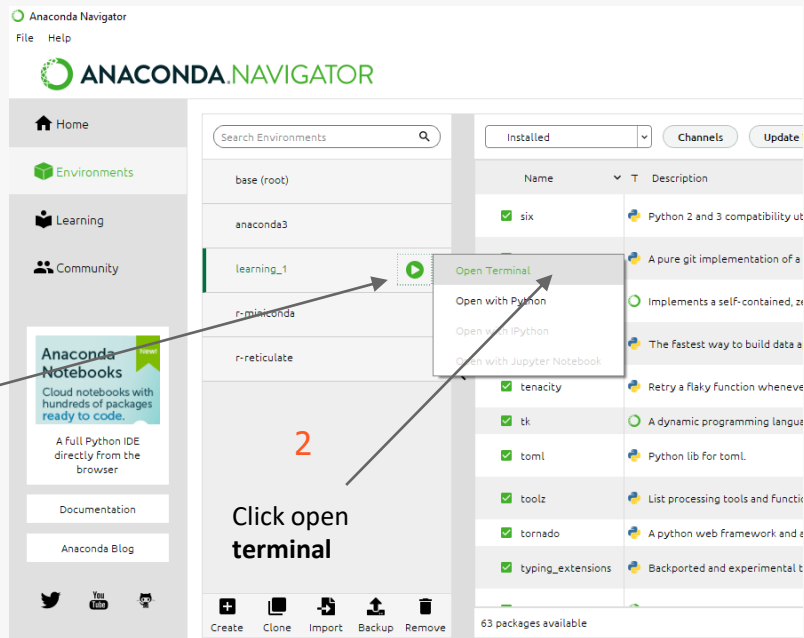
3

Fill the name for the new environment

4

Click Create

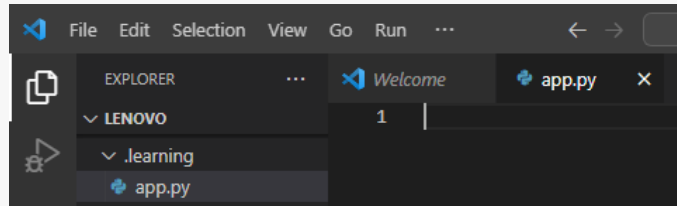
And then.



Make sure we are in the right environment

```
C:\WINDOWS\system32\cmd.exe  
(learning_1) C:\Users\Lenovo>pip install streamlit
```

type **pip install streamlit** on terminal the enter wait for the installation to finish and then type **code .**



Overview of the Streamlit app structure



Want to open about?

Predict my house!



User Input Features

Upload your input CSV file

Drag and drop file here

Limit 200MB per file • CSV

Browse files

Input Manual:

Number of area:

Welcome to my machine learning dashboard

This dashboard created by : [@abelkristanto](#)

This app predicts the **Housing Prices**

Data obtained from the [housing dataset](#).



Konfigurasi Halaman

```
st.set_page_config(page_title=None, page_icon=None,  
layout="centered", initial_sidebar_state="auto", menu_items=None)
```

Sebelum memulai membuat halaman dashboard, user dapat mengatur setting halaman menggunakan fungsi `st.set_page_config` dengan parameter antara lain

1. **page_title**: nama halaman pada bagian atas browser
2. **page_icon**: icon halaman
3. **layout**: tata letak, wide atau centered
4. **initial_sidebar_state**: kondisi awal sidebar (dijelaskan kemudian)
5. **menu_items**: menu tambahan, berupa dictionary dengan nama menu di bagian keys dan link di bagian values

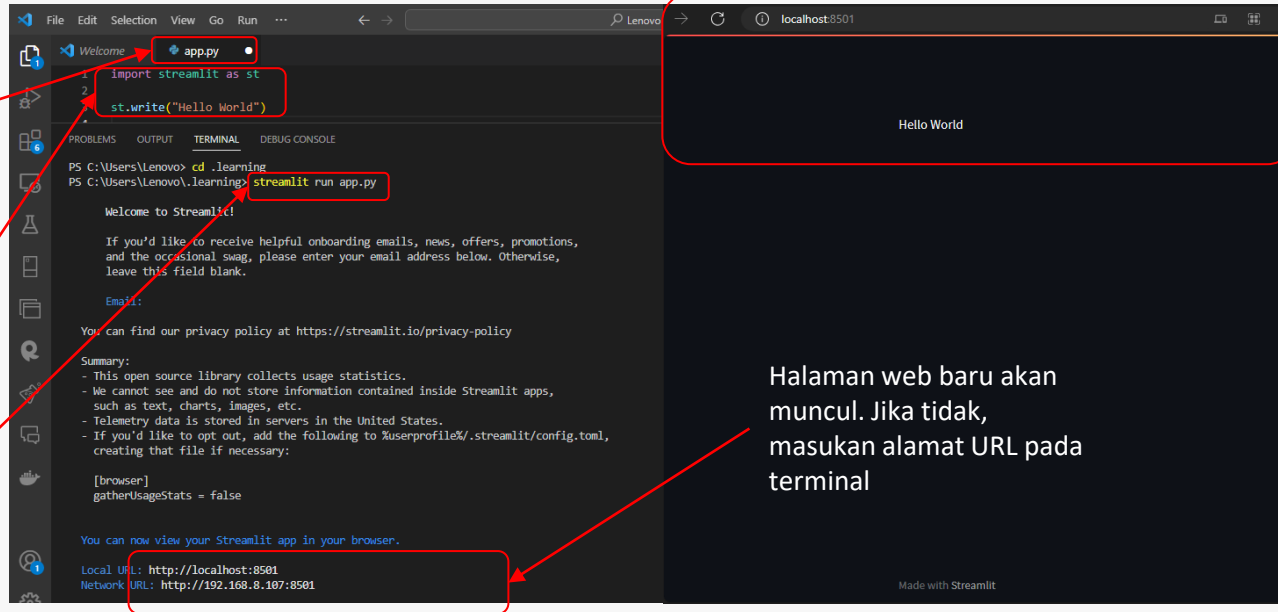


Memulai streamlit

Buat file baru di dalam folder project dengan nama `app.py`

Tulis kode berikut lalu **save**

Eksekusi '`streamlit run app.py`' pada terminal

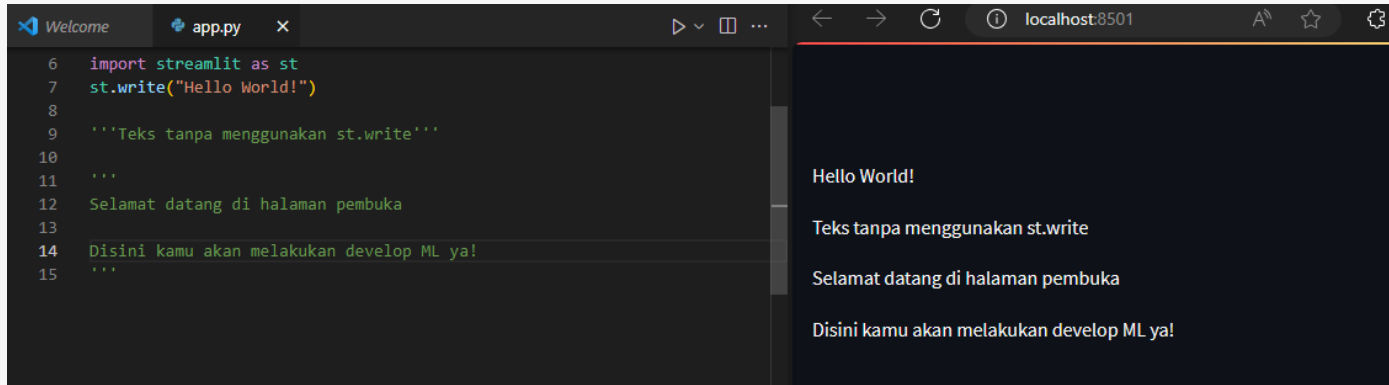


Output yang dihasilkan dari streamlit

Halaman web baru akan muncul. Jika tidak, masukan alamat URL pada terminal

Pastikan selalu save `app.py` setiap melakukan running

Menampilkan teks menggunakan `st.write` dan magic



```
6 import streamlit as st
7 st.write("Hello World!")
8
9 '''Teks tanpa menggunakan st.write'''
10
11 '''
12 Selamat datang di halaman pembuka
13
14 Disini kamu akan melakukan develop ML ya!
15 '''
```

Hello World!

Teks tanpa menggunakan st.write

Selamat datang di halaman pembuka

Disini kamu akan melakukan develop ML ya!

Seperti halnya menampilkan teks dengan metode `print()` di python, menampilkan teks pada streamlit dapat menggunakan `st.write` atau cukup dengan mengapit teks menggunakan triple quote (`'''`)

Menulis teks berbagai format

```
24 # Memasukkan teks dengan metode markdown
25 st.markdown("## Ini ditulis menggunakan _markdown_")
26
27 # Menulis teks dengan style title
28 st.title("Ini adalah judul")
29
30 # Menulis teks dengan style header
31 st.header("Ini juga adalah header")
32
33 # Menulis teks dengan style Subheader
34 st.header("Ini juga adalah subheader")
35
36 # Menulis teks dengan style preformatted-text/teks biasa
37 st.header("Ini juga adalah text biasa")
38
39 # Menulis teks dengan style caption
40 st.caption("Ini juga adalah caption")
41
42 # Menulis teks dengan format code
43 st.code("import streamlit as st")
44
45 # Menulis teks dengan format multi code
46 st.code("""
47 import pandas as pd
48 import streamlit as st # ini untuk memanggil package streamlit
49 """)
50
51 # Menulis teks latex
52 st.latex("x^2 + by + c = 100")
```

ini ditulis menggunakan *markdown*

Ini adalah judul

Ini juga adalah header

Ini juga adalah subheader

Ini juga adalah text biasa

Ini juga adalah caption

```
import streamlit as st
```

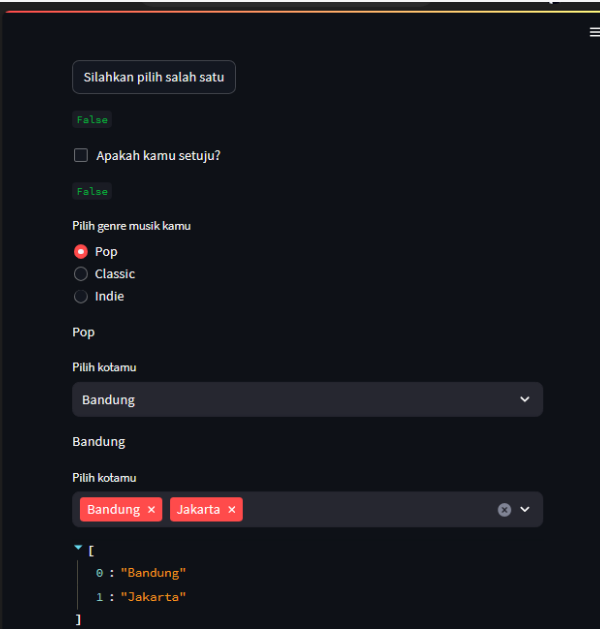
```
import pandas as pd
import streamlit as st # ini untuk memanggil package streamlit
```

$$x^2 + by + c = 100$$

Streamlit memiliki banyak opsi dalam menampilkan teks menjadi beberapa format seperti Judul, Bab, Subbab, teks biasa hingga kode dan LaTeX untuk menulis formula matematika. Selain itu, user juga dapat menambahkan format markdown yang lebih leluasa.

Input widget (1)

```
55 # Output TRUE/FALSE
56 tombol = st.button("Silahkan pilih salah satu")
57 st.write(tombol)
58
59 # Output TRUE/FALSE
60 agree = st.checkbox("Apakah kamu setuju?")
61 st.write(agree)
62
63 # Output 1 item
64 genre = st.radio(
65     "Pilih genre musik kamu",
66     ("Pop", "Classic", "Indie")
67 )
68 st.write(genre)
69
70 # Output 1 item
71 kota = st.selectbox(
72     "Pilih kotamu",
73     ("Bandung", "Jakarta", "Surabaya")
74 )
75 st.write(kota)
76
77 # Output Multiselect
78 multi = st.multiselect(
79     "Pilih kotamu",
80     ("Bandung", "Jakarta", "Surabaya")
81 )
82 st.write(multi)
```



st.button awalnya akan bernilai False lalu akan bernilai True jika diklik

st.checkbox akan bernilai True jika dicentang dan False untuk sebaliknya

st.radio akan memiliki nilai sesuai dengan opsi yang dipilih
st.selectbox akan memiliki nilai sesuai dengan opsi yang dipilih
st.multiselect akan

mengembalikan *list* berisi label yang dipilih

Input widget (2)

```
85 # Output slider
86 weight = st.slider(
87     "Masukkan berat badanmu",
88     min_value=0,
89     max_value=100,
90     step=1,
91     value=60 # Default
92 )
93 st.write(weight)
94
95 # Output item slider
96 size = st.select_slider(
97     "Masukkan ukuran bajumu",
98     ("XS", "S", "M", "L", "XL", "XXL"),
99     value = "M" # Default
100 )
101 st.write(size)
102
103 # Output Numerical
104 numeric = st.number_input(
105     "Masukkan nilai parameter numerik",
106     min_value=0,
107     max_value=100,
108     step=1,
109     value = 10 # Default
110 )
111 st.write(numeric)
112
113 # Output text
114 nama = st.text_input("Masukkan nama kamu")
115 st.write(nama)
116
117 # Output area text
118 tulisan = st.text_area("Tulis komentar disini")
119 st.write(tulisan)
```

st.slider dan **st.number_input**

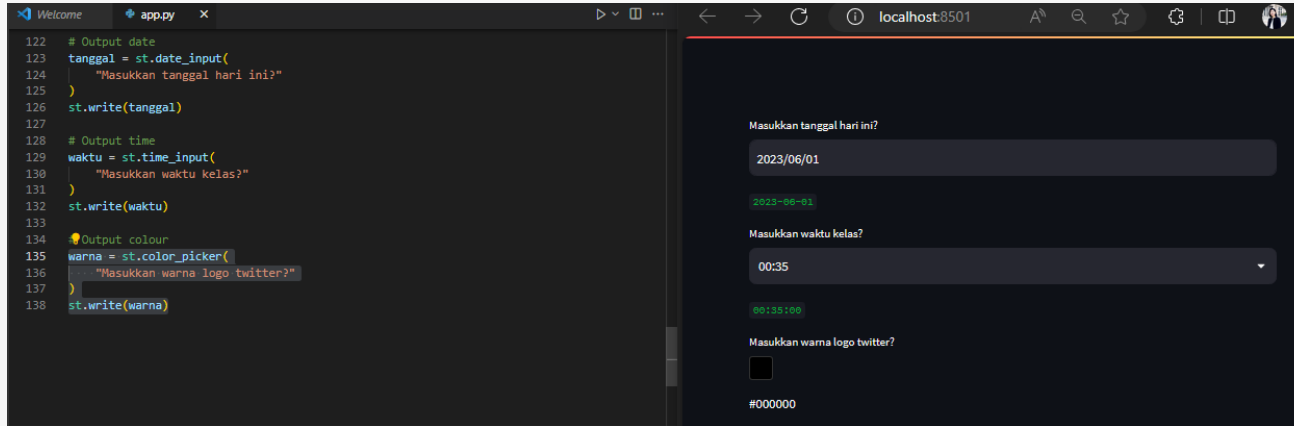
memiliki fungsi yang hampir sama yaitu meminta input berupa angka yang terdapat pada rentang tertentu

st.select_slider digunakan untuk meminta inputan berupa data ordinal seperti ukuran baju, indeks nilai, dsb

st.text_input meminta inputan berupa teks singkat

sedangkan **st.text_area** meminta inputan berupa teks panjang

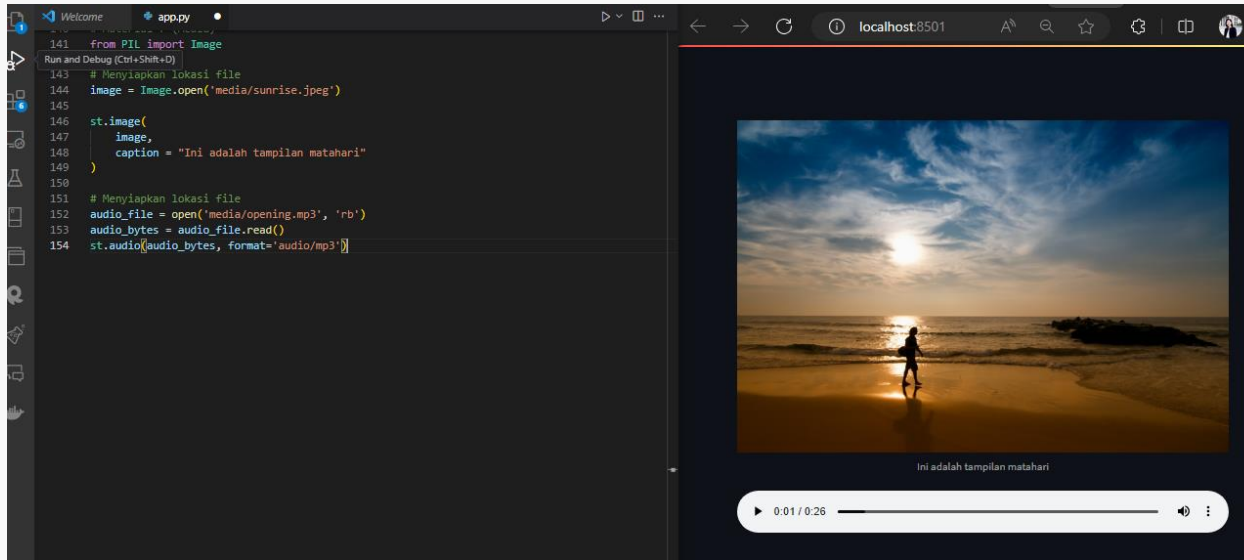
Input widget (3)



```
122 # Output date
123 tanggal = st.date_input(
124     "Masukkan tanggal hari ini?"
125 )
126 st.write(tanggal)
127
128 # Output time
129 waktu = st.time_input(
130     "Masukkan waktu kelas?"
131 )
132 st.write(waktu)
133
134 # Output colour
135 warna = st.color_picker(
136     "Masukkan warna logo twitter?"
137 )
138 st.write(warna)
```

date_input dan **time_input** adalah input widget dalam streamlit yang digunakan untuk meminta inputan berupa tanggal dan waktu **color_picker** membantu user dalam memilih kode warna hexadecimal

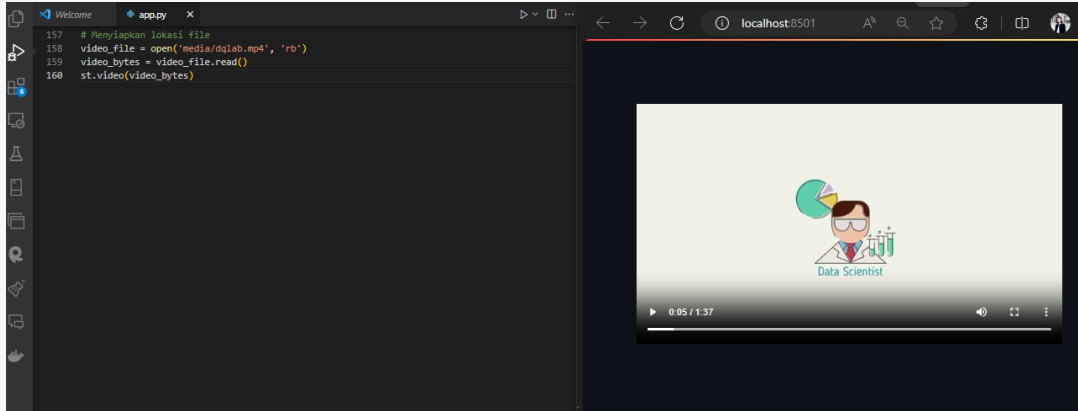
Media (1)



Menampilkan gambar menggunakan streamlit dapat menggunakan library **PIL** dan **st.image**

Untuk dapat menampilkan audio player yang dapat memainkan berbagai macam format audio dapat menggunakan **st.audio**

Media (2)

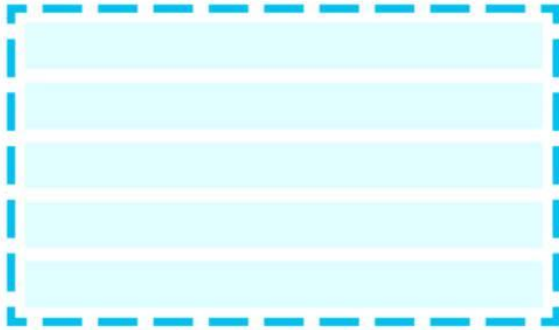


Menampilkan video player
dengan **st.video**

Container

Lorem ipsum dolor sit amet

Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Diam quis enim lobortis scelerisque fermentum dui faucibus in. Pharetra magna ac placerat vestibulum lectus mauris ultrices.



Sem integer vitae justo eget. In egestas erat imperdiet sed euismod nisi porta lorem mollis. Eu feugiat pretium nibh ipsum consequat nisl vel pretium. Elit ut aliquam purus sit amet. Aliquet nibh praesent tristique magna sit. Dapibus ultrices in iaculis nunc.

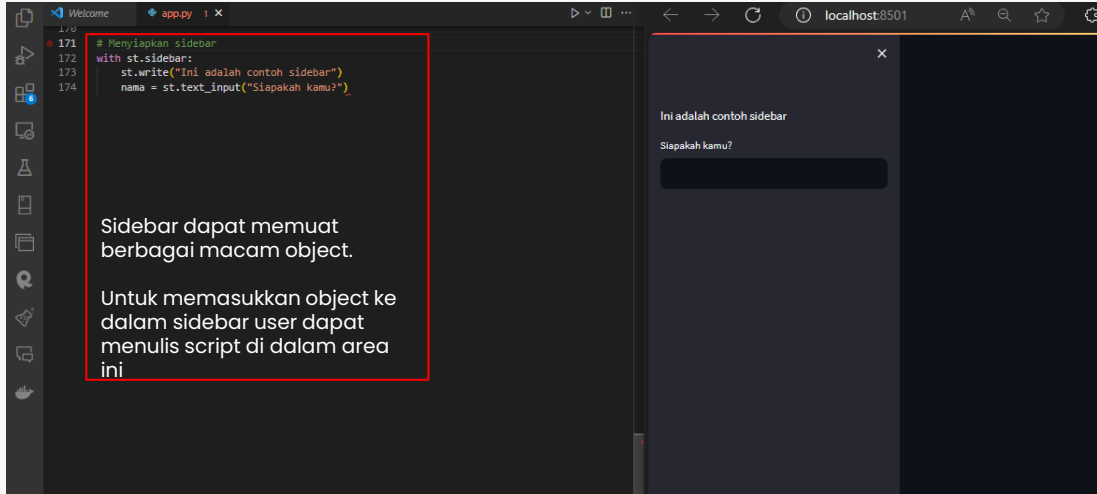
Enim eu turpis egestas pretium aenean pharetra. Nunc sed blandit libero volutpat sed cras ornare arcu. Etiam erat velit scelerisque in. Purus semper eget dui at tellus at urna condimentum mattis. Sapien

```
app.py U x
app.py
1 import streamlit as st
2 from lorem_text import lorem
3
4 with st.container():
5     st.write('ini di dalam kontainer')
6
7 st.write('ini di luar kontainer')
```

st.container adalah element yang berguna untuk mengelola penempatan dan pengelompokkan elemen.

untuk dapat memasukkan elemen ke dalam kontainer cukup masukkan elemen di dalam notasi with dari kontainer

Sidebar



Sidebar adalah elemen dalam streamlit yang berfungsi untuk menambahkan layout berupa sidebar interaktif yang *collapsible* dan memuat object

Columns

```
177 # Membuat 3 kolom dengan sama lebar
178 col1, col2, col3 = st.columns(3)
179
180 with col1:
181     st.title("Kol 1")
182
183 with col2:
184     st.title("Kol 2")
185
186 with col3:
187     st.title("Kol 3")
188
189 # Membuat 2 kolom dengan lebar 1/4 dan 3/4
190 col1, col2 = st.columns([1, 3])
191 with col1:
192     st.title("Kol 1")
193
194 with col2:
195     st.title("Kol 2")
```

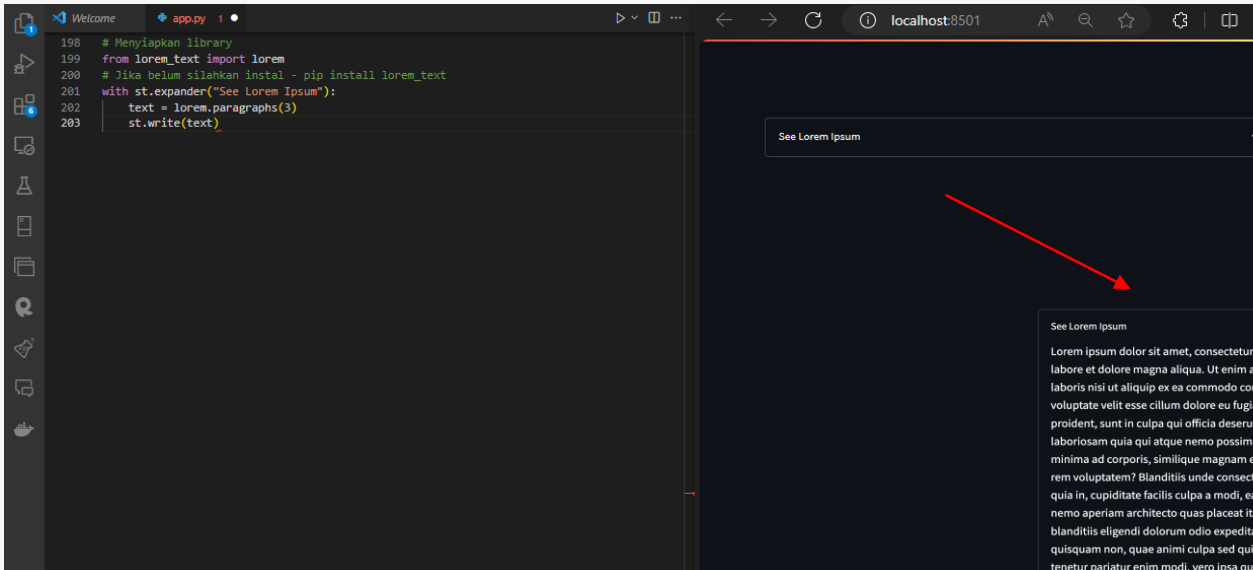
Kol 1 Kol 2 Kol 3

Kol 1 Kol 2

st.columns merupakan container yang tersusun secara horizontal dan berfungsi untuk mengelompokkan object dimana memasukkan n buah kolom sama lebar, cukup masukkan nilai n pada `st.columns(n)`

Untuk memasukkan n buah kolom dengan lebar variatif, masukkan parameter berupa list sepanjang n yang berisi nilai lebar dari kolom secara berurutan. Kolom bisa memuat kolom lagi namun maksimal 1 (one level nesting)

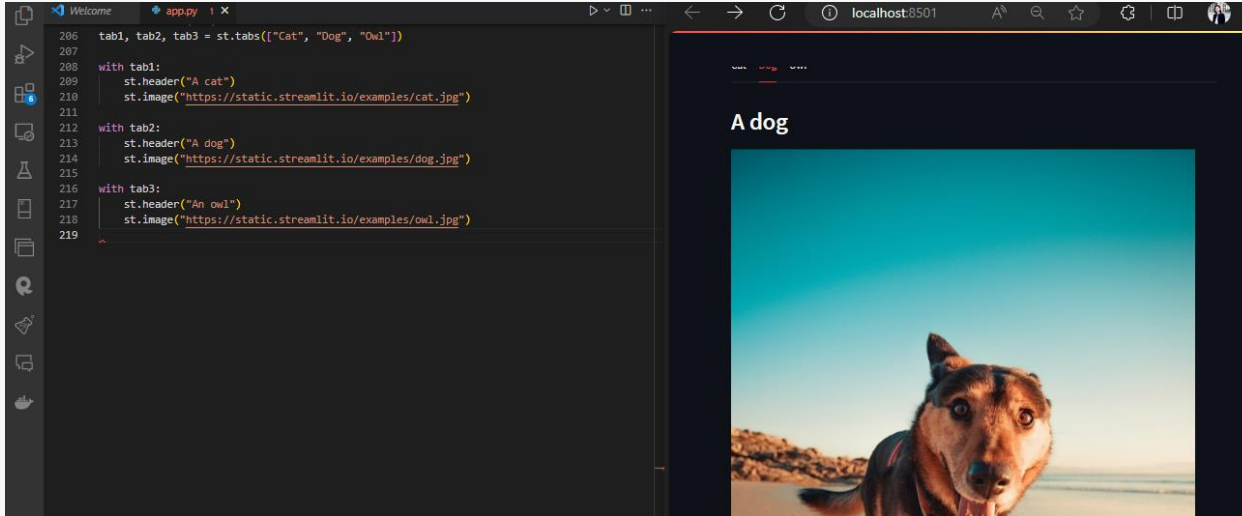
Expander



st.expander adalah elemen kontainer yang collapsible (dapat dibuka/tutup) karena satu jenis kontainer maka st.expander dapat dimasukkan beberapa element

Tidak dapat membuat expander di dalam expander (nested)

Tab



st.tabs adalah kontainer berbentuk tab, memudahkan user untuk bernavigasi di antara group konten yang saling berhubungan.

Untuk membuat n buah tab, berikan parameter pada st.tabs suatu list sepanjang n yang berisi judul untuk masing-masing tab

Installing necessary packages and loading the previously trained model

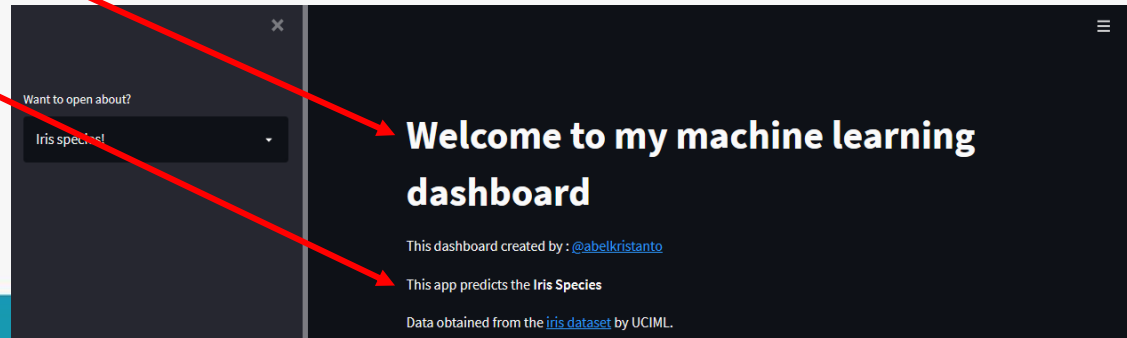
```
1 # Prepare the library
2 import streamlit as st
3 import pandas as pd
4 import pickle
5 import time
6 from PIL import Image

7
8 st.set_page_config(page_title="Halaman Modelling",
9                   layout="wide")
10
11 # Introduction
12 st.write("""
13     # Welcome to my machine learning dashboard
14
15     This dashboard created by : @abelkristanto(https://www.linkedin.com/in/abelkristanto/)
16     """)
17 add_selectitem = st.sidebar.selectbox("Want to open about?", ("Iris species!"))
18 st.write("""
19     This app predicts the Iris Species
20     Data obtained from the iris dataset(https://www.kaggle.com/uciml/iris) by UCIML.
21     """)
```

Menyiapkan library yang digunakan

Membuat halaman dashboard dengan judul Halaman Modelling dan menggunakan layout wide/penuh

Konfigurasi Halaman



```
47 if st.sidebar.button('Predict!'):
48     df = input_df
49     st.write(df)
50     with open("model/model_iris.pkl", 'rb') as file:
51         loaded_model = pickle.load(file)
52
53     prediction = loaded_model.predict(df)
54     result = ['Iris-setosa' if prediction == 0 else ('Iris-versicolor' if prediction == 1 else 'Iris-virginica')]
55     st.subheader('Prediction: ')
56     output = str(result[0])
57
58     with st.spinner('Wait for it...'):
59         time.sleep(4)
60     st.success(f"Prediction of this app is {output}")
```

Menampilkan data dalam bentuk dataframe

Mengimport model yang sudah kita buat sebelumnya

Melakukan prediksi hasil

Menampilkan hasil prediksi

Konfigurasi Modelling

User Input Features:

Upload your input CSV file

Drag and drop file here
Limit 200MB per file • CSV

[Browse files](#)

Input Manual

Sepal Length (cm)
4.30 10.00

Sepal Width (cm)
2.00 5.00

Petal Length (cm)
1.00 9.00

Petal Width (cm)
0.10 5.00


[Predict!](#)

Welcome to my machine learning dashboard

This dashboard created by : [@abelkristanto](#)

This app predicts the Iris Species

Data obtained from the [iris dataset](#) by UCIML.



**Building a simple Streamlit app
that takes user input, feeds it
into the model, and displays
the predicted output**


```
23 # Collects user input features into dataframe
24 st.sidebar.header('User Input Features:')
25 uploaded_file = st.sidebar.file_uploader("Upload your input CSV file", type=["csv"])
26 if uploaded_file is not None:
27     input_df = pd.read_csv(uploaded_file)
28 else:
29     def user_input_features():
30         st.sidebar.header('Input Manual')
31         SepalLengthCm = st.sidebar.slider('Sepal Length (cm)', 4.3,6.5,10.0)
32         SepalWidthCm = st.sidebar.slider('Sepal Width (cm)', 2.0,3.3,5.0)
33         PetalLengthCm = st.sidebar.slider('Petal Length (cm)', 1.0,4.5,9.0)
34         PetalWidthCm = st.sidebar.slider('Petal Width (cm)', 0.1,1.4,5.0)
35         data = {'SepalLengthCm': SepalLengthCm,
36                 'SepalWidthCm': SepalWidthCm,
37                 'PetalLengthCm': PetalLengthCm,
38                 'PetalWidthCm': PetalWidthCm}
39         features = pd.DataFrame(data, index=[0])
40         return features
41 input_df = user_input_features()
```

Menyiapkan inputan
dalam bentuk CSV

Menyiapkan inputan dalam
slider dalam halaman

Konfigurasi Inputan

User Input Features:

Upload your input CSV file

Drag and drop file here
Limit 200MB per file • CSV

[Browse files](#)

Input Manual

Sepal Length (cm)

4.30 10.00

Sepal Width (cm)

2.00 5.00

Petal Length (cm)

1.00 9.00

Petal Width (cm)

0.10 5.00

[Predict!](#)

Welcome to my machine learning dashboard

This dashboard created by : [@abelkristanto](#)

This app predicts the Iris Species

Data obtained from the [iris dataset](#) by UCIML



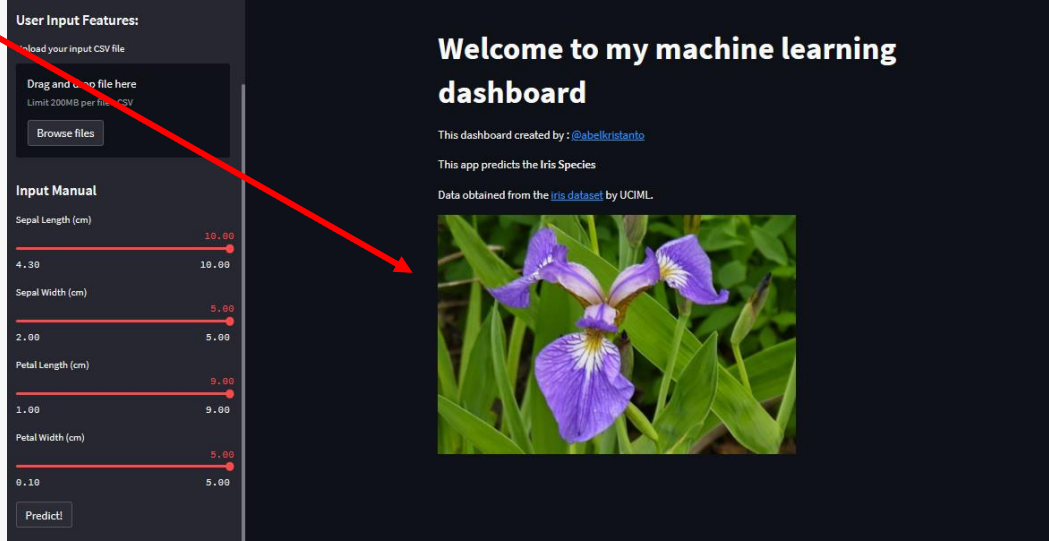
```
43 # Add picture
44 img = Image.open("iris.JPG")
45 st.image(img, width=500)
```

Simpan dahulu file iris.JPG dalam environment yang kamu miliki

Atur ukuran lebar gambar, kamu bisa menambahkan

caption = "Iris Picture"

Menambahkan picture



User Input Features:

Upload your input CSV file

Drag and drop file here

Limit 200MB per file • CSV

Browse files

Input Manual

Sepal Length (cm)



Sepal Width (cm)



Petal Length (cm)



Petal Width (cm)



Predict!

Welcome to my machine learning dashboard

This dashboard created by: [@abelkristanto](#)

This app predicts the Iris Species

Data obtained from the [iris dataset](#) by UCIML**User Input Features:**

Upload your input CSV file

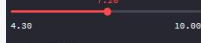
Drag and drop file here

Limit 200MB per file • CSV

Browse files

Input Manual

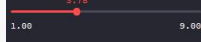
Sepal Length (cm)



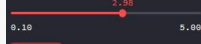
Sepal Width (cm)



Petal Length (cm)



Petal Width (cm)



Predict!

Welcome to my machine learning dashboard

This dashboard created by: [@abelkristanto](#)

This app predicts the Iris Species

Data obtained from the [iris dataset](#) by UCIML

	Sepal.LengthCm	Sepal.WidthCm	Petal.LengthCm	Petal.WidthCm
0	7.2000	3.3600	3.7800	2.5800

Prediction:

Prediction of this app is Iris-versicolor

Before dan After

Thanks!

