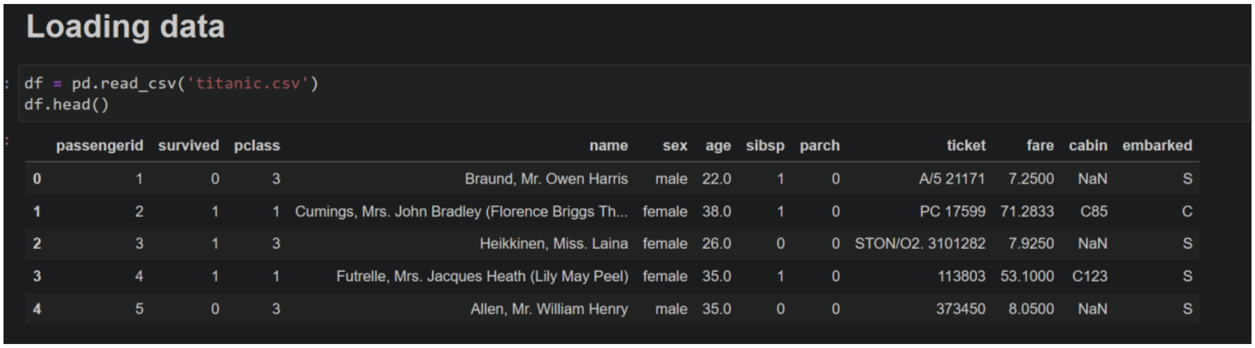
Name: stephen williams

Batch code: LISUM19

Submission date: 4/4/2022

Submitted to data glacier

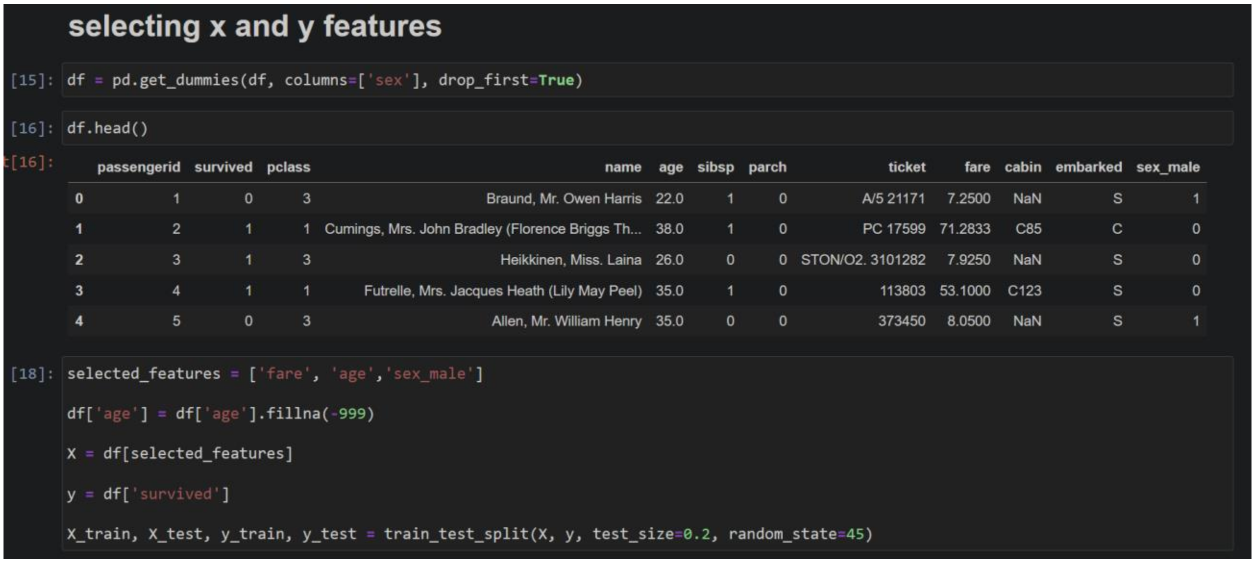
Website: https://svalentinow-streamlit-titanic-model-app-jh5mff.streamlit.app/



Loading data

```
df = pd.read_csv('titanic.csv')
df.head()
```

	passengerid	survived	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S



selecting x and y features

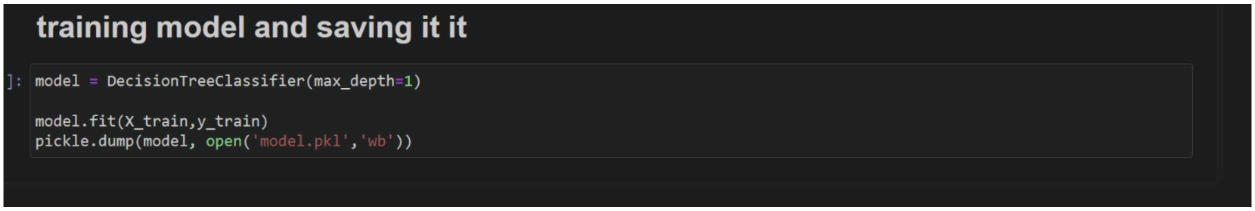
```
[15]: df = pd.get_dummies(df, columns=['sex'], drop_first=True)
```

```
[16]: df.head()
```

[16]:

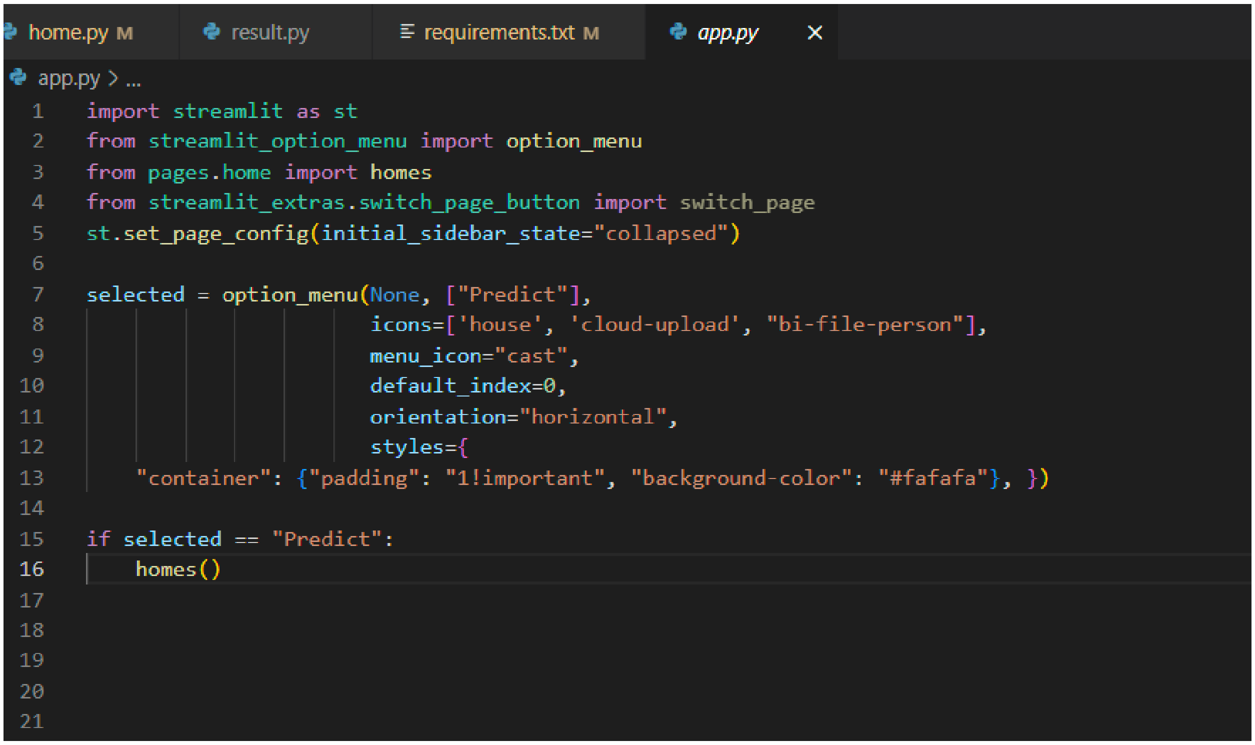
	passengerid	survived	pclass	name	age	sibsp	parch	ticket	fare	cabin	embarked	sex_male
0	1	0	3	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.2500	NaN	S	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	0	PC 17599	71.2833	C85	C	0
2	3	1	3	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.1000	C123	S	0
4	5	0	3	Allen, Mr. William Henry	35.0	0	0	373450	8.0500	NaN	S	1

```
[18]: selected_features = ['fare', 'age','sex_male']

      df['age'] = df['age'].fillna(-999)

      X = df[selected_features]

      y = df['survived']

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=45)
```

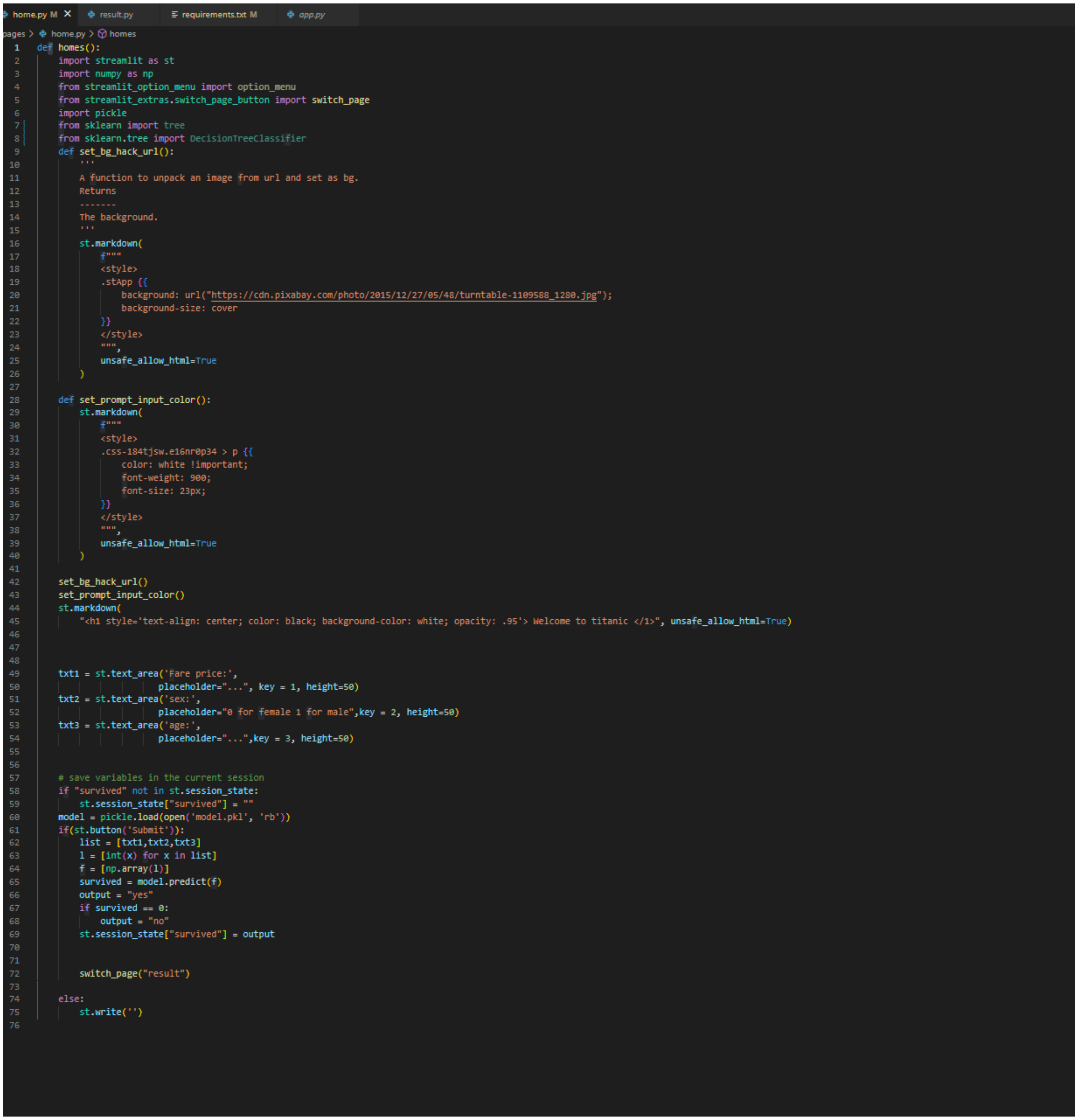


training model and saving it it

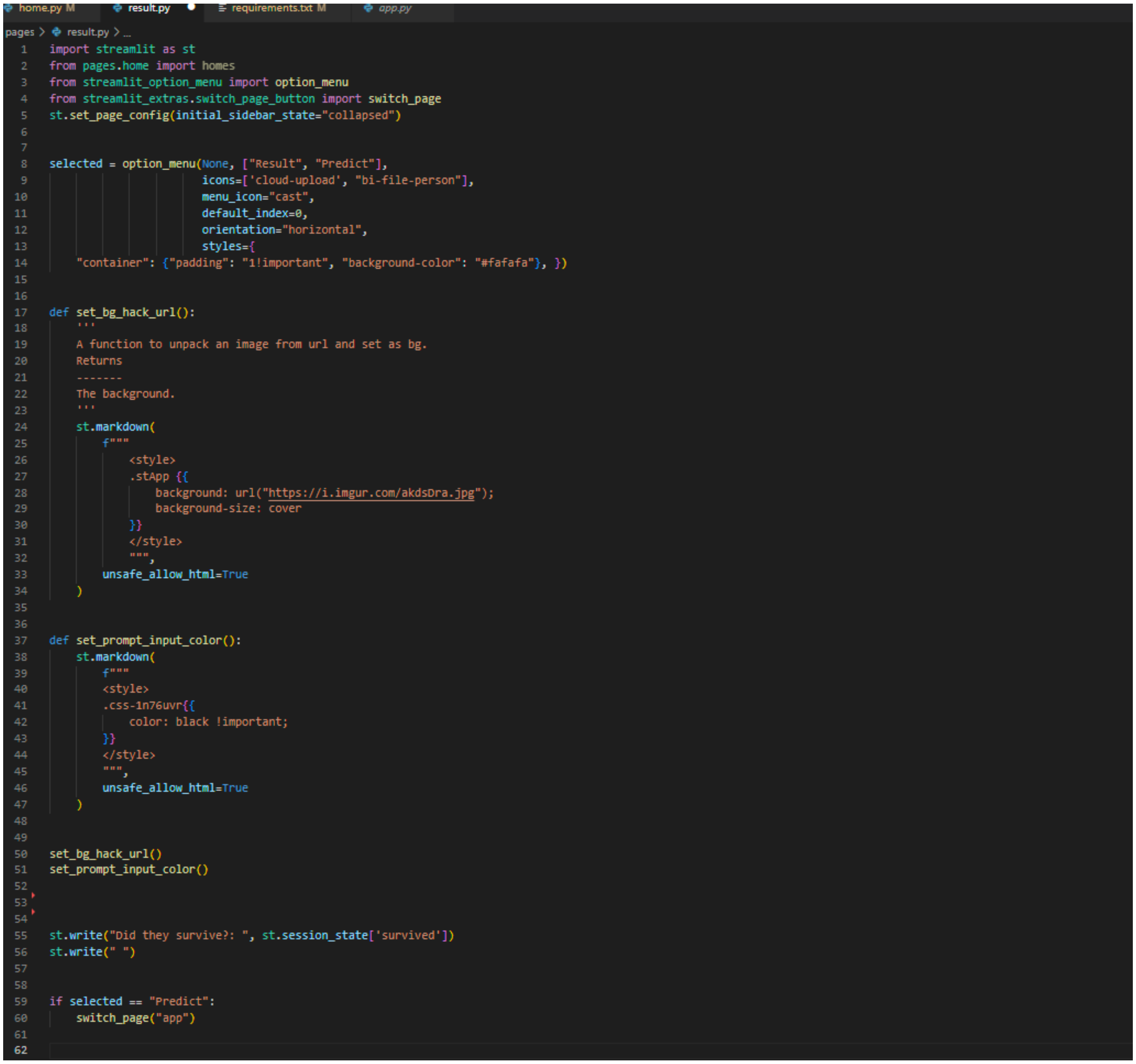
```
model = DecisionTreeClassifier(max_depth=1)

model.fit(X_train,y_train)
pickle.dump(model, open('model.pkl','wb'))
```



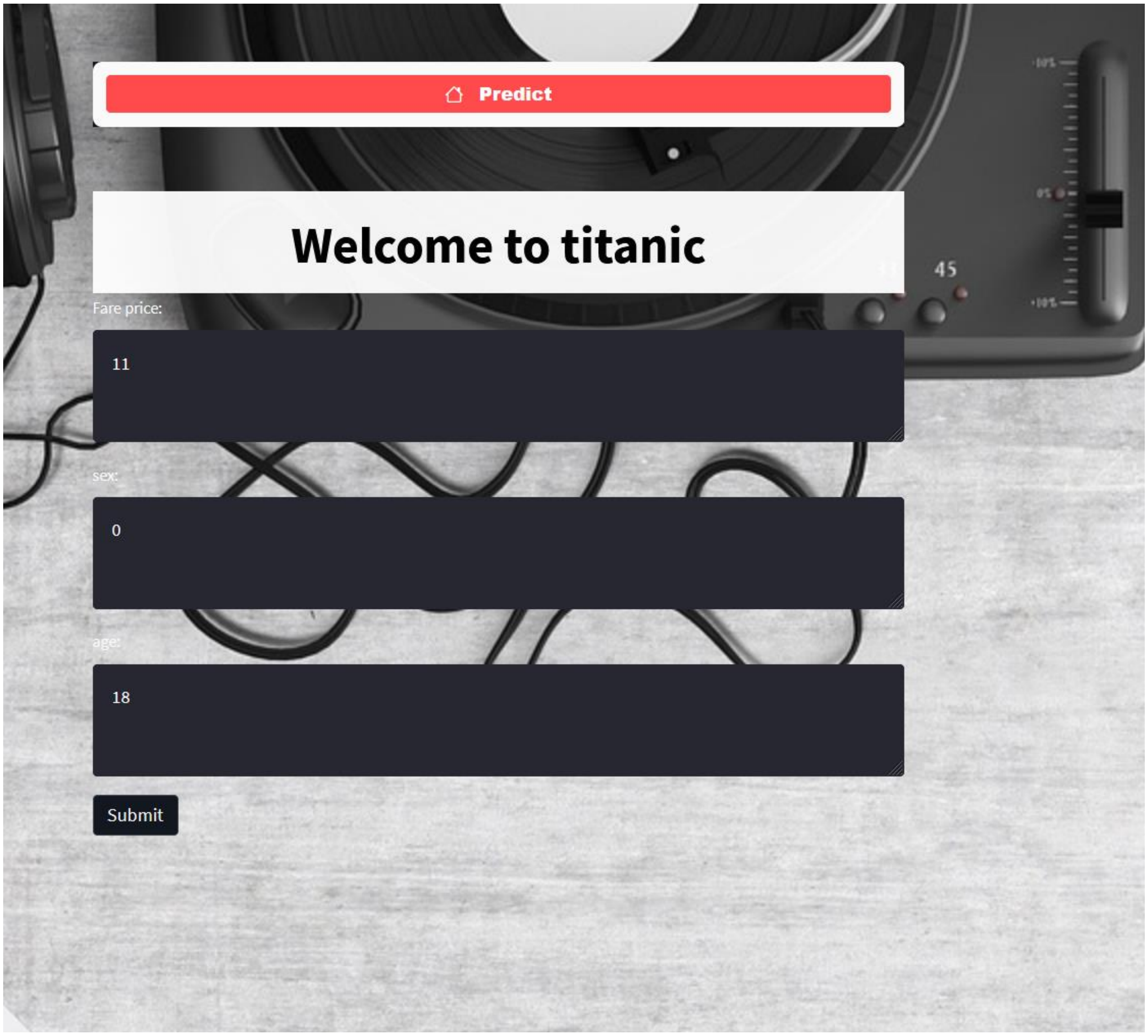
```python
import streamlit as st
from streamlit_option_menu import option_menu
from pages.home import homes
from streamlit_extras.switch_page_button import switch_page
st.set_page_config(initial_sidebar_state="collapsed")

selected = option_menu(None, ["Predict"],
                       icons=['house', 'cloud-upload', "bi-file-person"],
                       menu_icon="cast",
                       default_index=0,
                       orientation="horizontal",
                       styles={
    "container": {"padding": "1!important", "background-color": "#fafafa"}, })

if selected == "Predict":
    homes()
```



```python
def homes():
    import streamlit as st
    import numpy as np
    from streamlit_option_menu import option_menu
    from streamlit_extras.switch_page_button import switch_page
    import pickle
    from sklearn import tree
    from sklearn.tree import DecisionTreeClassifier
    def set_bg_hack_url():
        '''
        A function to unpack an image from url and set as bg.
        Returns
        -------
        The background.
        '''
        st.markdown(
            f"""
            <style>
            .stApp {{
                background: url("https://cdn.pixabay.com/photo/2015/12/27/05/48/turntable-1109588_1280.jpg");
                background-size: cover
            }}
            </style>
            """,
            unsafe_allow_html=True
        )

    def set_prompt_input_color():
        st.markdown(
            f"""
            <style>
            .css-184tjsw.e16nr0p34 > p {{
                color: white !important;
                font-weight: 900;
                font-size: 23px;
            }}
            </style>
            """,
            unsafe_allow_html=True
        )

    set_bg_hack_url()
    set_prompt_input_color()
    st.markdown(
        "<h1 style='text-align: center; color: black; background-color: white; opacity: .95'> Welcome to titanic </1>", unsafe_allow_html=True)



    txt1 = st.text_area('Fare price:',
                        placeholder="...", key = 1, height=50)
    txt2 = st.text_area('sex:',
                        placeholder="0 for female 1 for male",key = 2, height=50)
    txt3 = st.text_area('age:',
                        placeholder="...",key = 3, height=50)


    # save variables in the current session
    if "survived" not in st.session_state:
        st.session_state["survived"] = ""
    model = pickle.load(open('model.pkl', 'rb'))
    if(st.button('Submit')):
        list = [txt1,txt2,txt3]
        l = [int(x) for x in list]
        f = [np.array(l)]
        survived = model.predict(f)
        output = "yes"
        if survived == 0:
            output = "no"
        st.session_state["survived"] = output


        switch_page("result")

    else:
        st.write('')

```



```python
import streamlit as st
from pages.home import homes
from streamlit_option_menu import option_menu
from streamlit_extras.switch_page_button import switch_page
st.set_page_config(initial_sidebar_state="collapsed")


selected = option_menu(None, ["Result", "Predict"],
                       icons=['cloud-upload', "bi-file-person"],
                       menu_icon="cast",
                       default_index=0,
                       orientation="horizontal",
                       styles={
    "container": {"padding": "1!important", "background-color": "#fafafa"}, })


def set_bg_hack_url():
    '''
    A function to unpack an image from url and set as bg.
    Returns
    -------
    The background.
    '''
    st.markdown(
        f"""
            <style>
            .stApp {{
                background: url("https://i.imgur.com/akdsDra.jpg");
                background-size: cover
            }}
            </style>
            """,
        unsafe_allow_html=True
    )


def set_prompt_input_color():
    st.markdown(
        f"""
        <style>
        .css-1n76uvr{{
            color: black !important;
        }}
        </style>
        """,
        unsafe_allow_html=True
    )


set_bg_hack_url()
set_prompt_input_color()



st.write("Did they survive?: ", st.session_state['survived'])
st.write(" ")


if selected == "Predict":
    switch_page("app")

```



Welcome to titanic

Fare price:

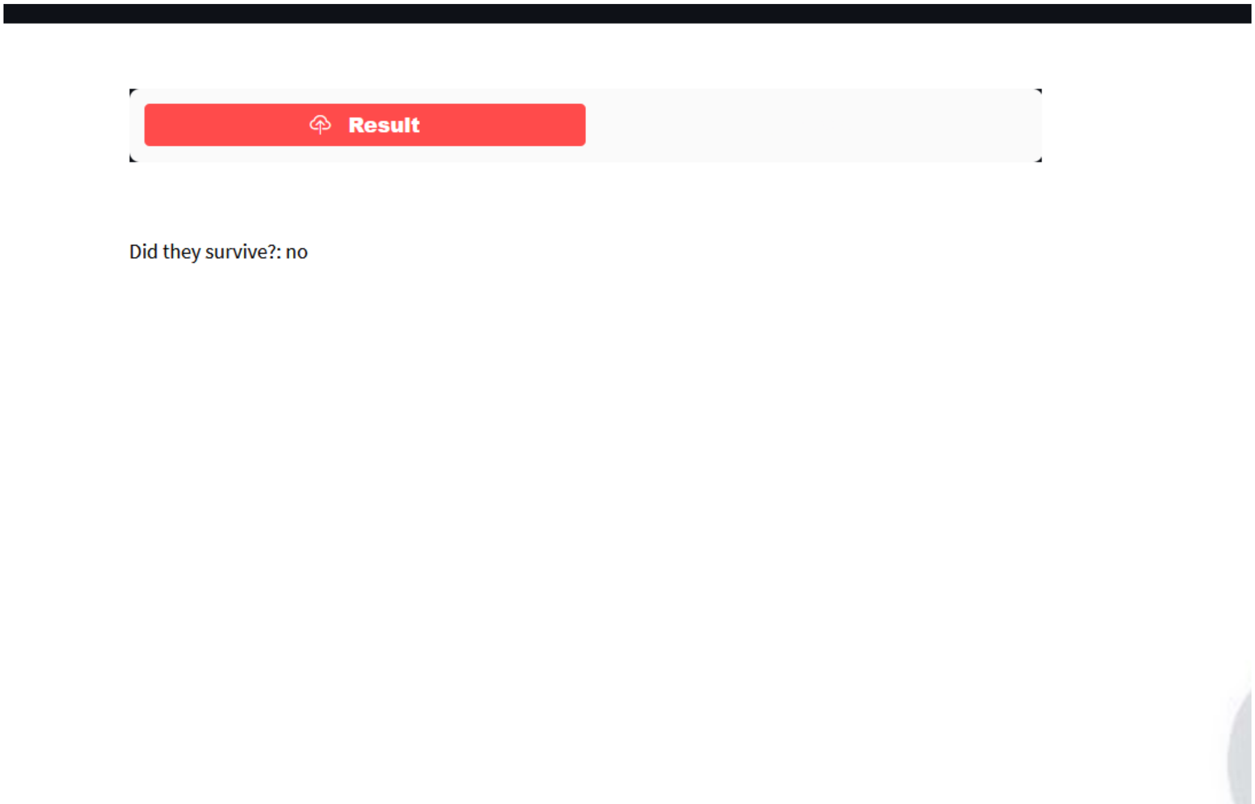
11

sex:

0

age:

18



Did they survive?: no