Name: stephen williams

Batch code:

Submission date: 3/28/2022

Submitted to data glacier

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'))
```

index.html | Release Notes: 1.76.2 | app.py | requirements.txt

app.py > Python > predict

```python
import numpy as np
from flask import Flask, request,render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)
    output = "yes"
    if prediction == 0:
        output = "no"

    return render_template('index.html', prediction_text='survived {}'.format(output))

if __name__ == "__main__":
    app.run(debug=True)
```


Did they survive the titanic?
Fare
Age
sex: 1 is male 2 0 is female
Predict


Did they survive the titanic?
Fare
Age
sex: 1 is male 2 0 is female
Predict
survived yes