Index.html

<!DOCTYPE html>

<html>

<head>

<title>Titanic Survival predictor</title>

<link rel = "stylesheet" href="filename='style.css">

</head>

<body>

<h2>Titanic Survival Predictor</h2>

<form action ="/predict" method="post">

<label>Sex: </label>

<select name="sex">

<option value = "male">Male</option>

<option value = "female">Female</option>

</select><br>

<label>Fare: </label>

<input type = "text" name="fare"><br>

<input type="submit" value="Predict">

</form>

{% if prediction %}

<h3>Result: {{prediction}}</h3>

{% endif %}

</body>

</html>

Style.css

body{

background-color: azure;

font-family: Arial, Helvetica, sans-serif;

color: black;

text-align: center;

padding-top: 50px;

}

h2 {

color: brown;

}

form{

background-color: bisque;

border-radius: 10px;

padding: 20px;

display: inline-block;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

label, select, input{

font-size: 18px;

margin: 10px;

display: block;

}

input[type="text"],select{

padding: 10px;

width:200px;

margin:auto;

}

input[type="submit"]:hover{

background-color: rgb(46, 46, 108);

}

h3{

margin-top: 30px;

color: rgb(40, 217, 16);

}

app.py

from flask import Flask,request,render\_template

import joblib

import numpy as np

app = Flask(\_\_name\_\_)

model = joblib.load('titanic\_model.pkl')

@app.route("/")

def home():

return render\_template('index.html')

@app.route("/predict",methods = ["POST"])

def predict():

sex = 1 if request.form['sex'] == 'female' else 0

fare = float(request.form['fare'])

prediction = model.predict([[sex,fare]])[0]

result = "Survived" if prediction == 1 else 'Did not survive'

return render\_template('index.html',prediction = result)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug = True)

Train\_model.py

import pandas as pd

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

import joblib

#Load the dataset

df = pd.read\_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv")

#Simple preprocessing

df = df[["Sex",'Fare','Survived']].dropna()

df['Sex'] = df["Sex"].map({'male':0,'female':1})

X = df[['Sex','Fare']]

y = df['Survived']

#Train and save the model

model = LogisticRegression()

model.fit(X,y)

joblib.dump(model,'titanic\_model.pkl')

Requirements.txt

Flask

scikit-learn

joblib

Numpy

pandas