Data Glacier Virtual Internship Program - Week 4 Submission

Name: Raj Pawar
Batch Code: LISUM44
Submission Date: 25th April 2025
Submitted To: Canvas (GitHub repo link)

1. Task Overview

In Week 4 of the Virtual Internship, the task was to select a toy dataset, train a machine learning model, deploy it using Flask, create a PDF documentation with screenshots, and submit via GitHub.

2. Dataset Used

I have used the Titanic dataset available through the Seaborn library. This dataset contains information about the passengers aboard the Titanic, including age, sex, passenger class, fare paid, and survival status.

3. Model Training

1. Libraries used: pandas, scikit-learn, seaborn
2. Model: RandomForestClassifier
3. Features used: pclass, sex, age, fare, sibsp, parch, embarked
4. Saved the model into titanic_model.pkl using pickle.

[1]:
```
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import pickle

df = sns.load_dataset('titanic')

df.head()
```

[1]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

[2]:
```
df = df.dropna(subset=['age', 'fare', 'embarked', 'sex'])

df['sex'] = df['sex'].map({'male': 0, 'female': 1})
df['embarked'] = df['embarked'].map({'C': 0, 'Q': 1, 'S': 2})

X = df[['pclass', 'sex', 'age', 'fare', 'sibsp', 'parch', 'embarked']]  # Features
y = df['survived']  # Target (what we want to predict)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[3]:
```
from sklearn.ensemble import RandomForestClassifier
import pickle

model = RandomForestClassifier()
model.fit(X_train, y_train)

with open('titanic_model.pkl', 'wb') as file:
    pickle.dump(model, file)

print("Titanic model trained and saved successfully!")
```

```
Titanic model trained and saved successfully!
```

Jupyter notebook showing successful model training and saving ("Titanic model trained and saved successfully!")

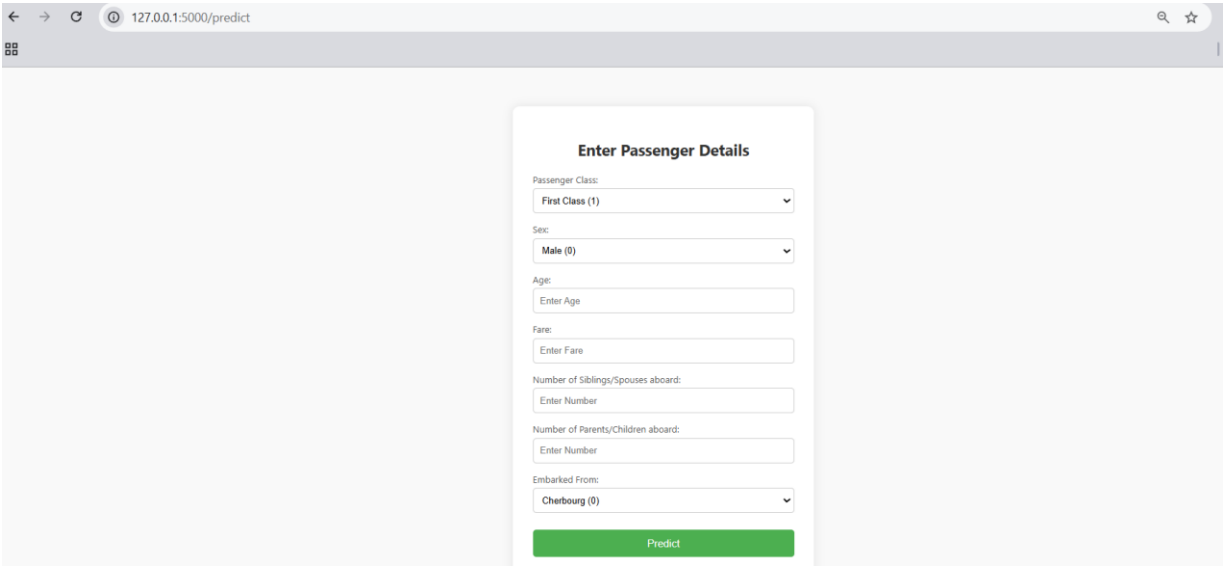
4. Flask Web App Development

I developed a Flask application with a web interface to allow users to input Titanic passenger details and get a prediction whether the passenger would have survived. The web app was styled using CSS for a clean and minimal design.

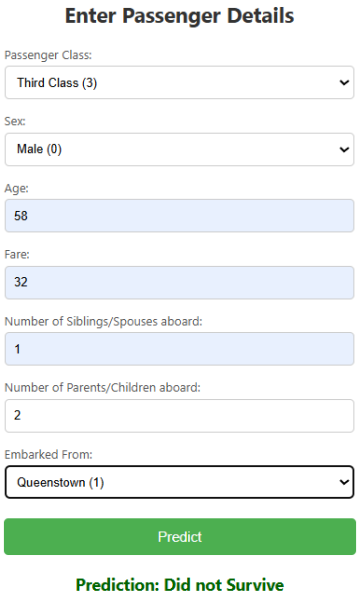
```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Raj\VI - Week 4>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Flask app running in terminal



Home page of the web app



Filled form with input values & Prediction result displayed