

Model Deployment on Streamlit Cloud

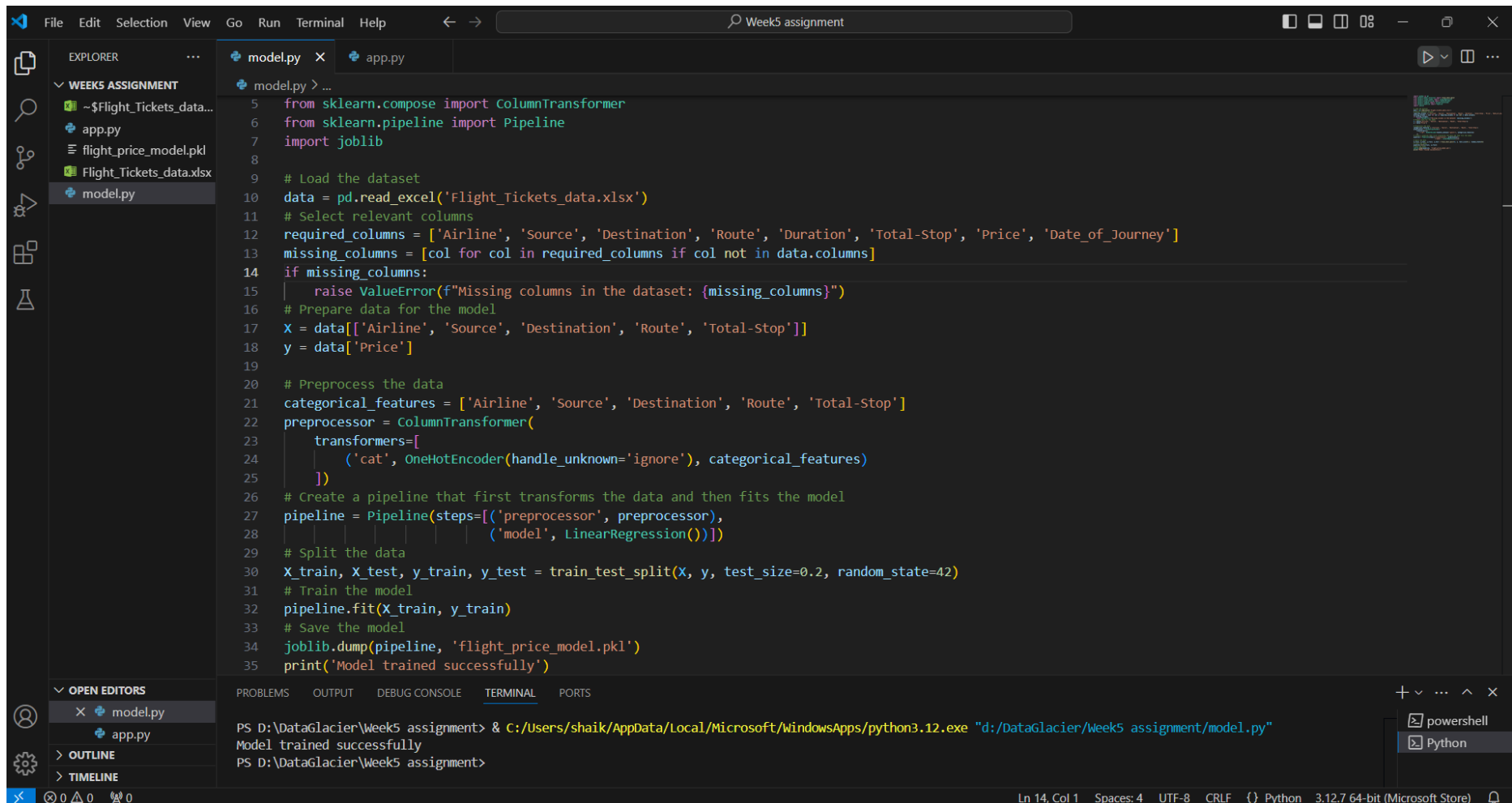
Name: Amrin Shaikh

Batch code: LISUM38

Submission date:10/10/2024

Submission to: Data Glacier

1. **Select Toy Data and Build the Model:** Here's a simplified example using a toy dataset (Flight_Tickets_data.xlsx) for predicting a value (e.g., price).



The screenshot displays a Visual Studio Code editor window titled "Week5 assignment". The Explorer panel on the left shows a project structure with a "WEEKS ASSIGNMENT" folder containing files like "Flight_Tickets_data.xlsx", "flight_price_model.pkl", "app.py", and "model.py". The main editor area shows the content of "model.py", which is a Python script for training a linear regression model. The script imports necessary libraries, loads the dataset, preprocesses the data using a ColumnTransformer, splits it into training and testing sets, trains the model, and saves it as "flight_price_model.pkl". The bottom panel shows the TERMINAL output, indicating that the model was trained successfully. The status bar at the bottom indicates the file is at line 14, column 1, using UTF-8 encoding and CRLF line endings, and is a Python 3.12.7 64-bit file.

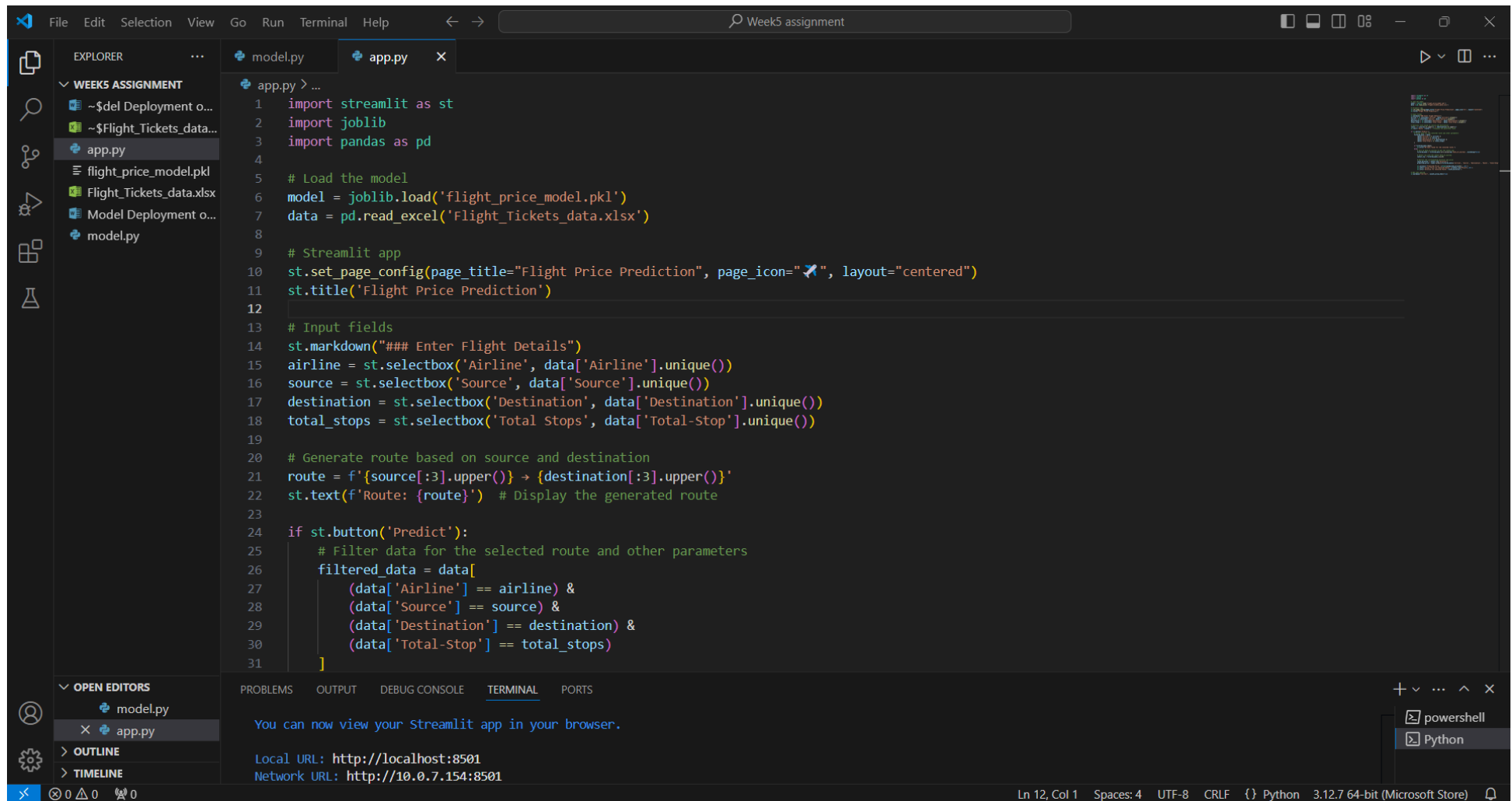
```
5 from sklearn.compose import ColumnTransformer
6 from sklearn.pipeline import Pipeline
7 import joblib
8
9 # Load the dataset
10 data = pd.read_excel('Flight_Tickets_data.xlsx')
11 # Select relevant columns
12 required_columns = ['Airline', 'Source', 'Destination', 'Route', 'Duration', 'Total-Stop', 'Price', 'Date_of_Journey']
13 missing_columns = [col for col in required_columns if col not in data.columns]
14 if missing_columns:
15     raise ValueError(f"Missing columns in the dataset: {missing_columns}")
16 # Prepare data for the model
17 X = data[['Airline', 'Source', 'Destination', 'Route', 'Total-Stop']]
18 y = data['Price']
19
20 # Preprocess the data
21 categorical_features = ['Airline', 'Source', 'Destination', 'Route', 'Total-Stop']
22 preprocessor = ColumnTransformer(
23     transformers=[
24         ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
25     ])
26 # Create a pipeline that first transforms the data and then fits the model
27 pipeline = Pipeline(steps=[('preprocessor', preprocessor),
28                             ('model', LinearRegression())])
29 # Split the data
30 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
31 # Train the model
32 pipeline.fit(X_train, y_train)
33 # Save the model
34 joblib.dump(pipeline, 'flight_price_model.pkl')
35 print('Model trained successfully')
```

PS D:\DataGlacier\Week5 assignment> & C:/Users/shaik/AppData/Local/Microsoft/WindowsApps/python3.12.exe "d:/DataGlacier/Week5 assignment/model.py"

Model trained successfully

PS D:\DataGlacier\Week5 assignment>

2. Create app.py for Model Deployment:



```
File Edit Selection View Go Run Terminal Help
Week5 assignment

EXPLORER
WEEK5 ASSIGNMENT
~$del Deployment o...
~$Flight_Tickets_data...
app.py
flight_price_model.pkl
Flight_Tickets_data.xlsx
Model Deployment o...
model.py

app.py > ...
1 import streamlit as st
2 import joblib
3 import pandas as pd
4
5 # Load the model
6 model = joblib.load('flight_price_model.pkl')
7 data = pd.read_excel('Flight_Tickets_data.xlsx')
8
9 # Streamlit app
10 st.set_page_config(page_title="Flight Price Prediction", page_icon="✈️", layout="centered")
11 st.title('Flight Price Prediction')
12
13 # Input fields
14 st.markdown("### Enter Flight Details")
15 airline = st.selectbox('Airline', data['Airline'].unique())
16 source = st.selectbox('Source', data['Source'].unique())
17 destination = st.selectbox('Destination', data['Destination'].unique())
18 total_stops = st.selectbox('Total Stops', data['Total-Stops'].unique())
19
20 # Generate route based on source and destination
21 route = f'{source[:3].upper()} → {destination[:3].upper()}'
22 st.text(f'Route: {route}') # Display the generated route
23
24 if st.button('Predict'):
25     # Filter data for the selected route and other parameters
26     filtered_data = data[
27         (data['Airline'] == airline) &
28         (data['Source'] == source) &
29         (data['Destination'] == destination) &
30         (data['Total-Stops'] == total_stops)
31     ]

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://10.0.7.154:8501

+ ... ^ x
powershell
Python

Ln 12, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.7 64-bit (Microsoft Store)
```

3. Testing Deployment:

The screenshot shows a web browser window with the title "Flight Price Prediction" and the address bar displaying "localhost:8501". The application interface is dark-themed and features the following elements:

- Header:** A "Deploy" button with a dropdown menu icon in the top right corner.
- Main Title:** "Flight Price Prediction" in large, bold, white text.
- Form Section:** Titled "Enter Flight Details" in white text, containing four dropdown menus:
 - Airline:** Set to "Air India".
 - Source:** Set to "Delhi".
 - Destination:** Set to "Cochin".
 - Total Stops:** Set to "1.0".
- Route Summary:** Below the form, it displays "Route: DEL → COC".
- Action Button:** A "Predict" button located at the bottom of the form.

4. Input / select Values:

Flight Price Prediction

Enter Flight Details

Airline

Air India

Multiple carriers

Air India

Jet Airways

Air Asia

IndiGo

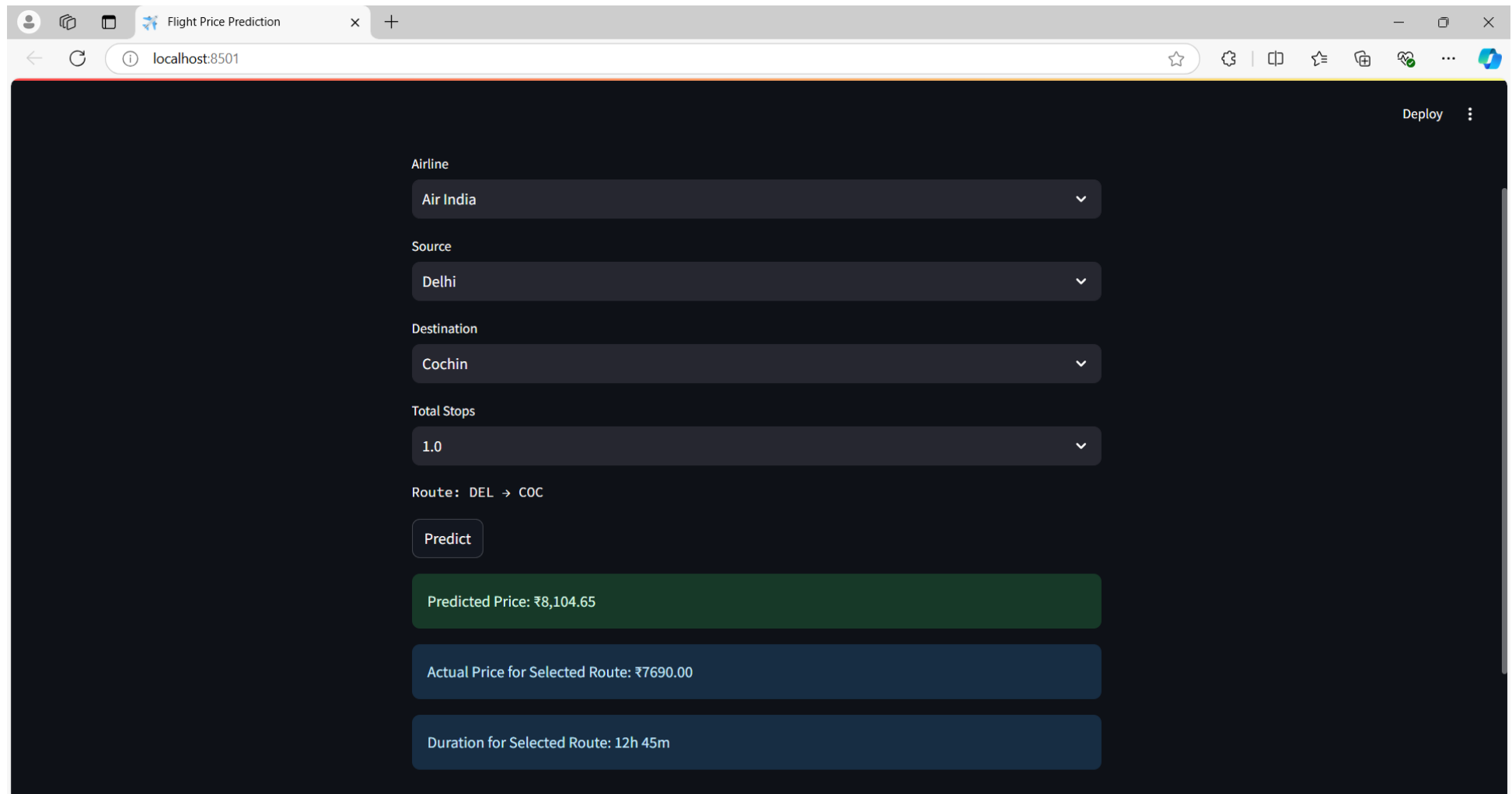
Multiple carriers Premium economy

GoAir

Predict

Deploy

5. Predicted Price with other information:



The screenshot shows a web browser window with the title "Flight Price Prediction" and the address bar displaying "localhost:8501". The application interface is dark-themed and includes a "Deploy" button in the top right corner. The main form contains four dropdown menus for "Airline", "Source", "Destination", and "Total Stops", with values "Air India", "Delhi", "Cochin", and "1.0" respectively. Below these is a "Route: DEL → COC" label and a "Predict" button. The results are displayed in three horizontal bars: a green bar for "Predicted Price: ₹8,104.65", a dark blue bar for "Actual Price for Selected Route: ₹7690.00", and another dark blue bar for "Duration for Selected Route: 12h 45m".

Deploy

Airline

Air India

Source

Delhi

Destination

Cochin

Total Stops

1.0

Route: DEL → COC

Predict

Predicted Price: ₹8,104.65

Actual Price for Selected Route: ₹7690.00

Duration for Selected Route: 12h 45m

6. Streamlit Deployment:

Deploy an app · Streamlit

Model-deployment-on-cloud/ap

https://share.streamlit.io/deploy

shaikhAmrin02

My appsMy profileExploreDiscuss

[← Back](#)

Deploy an app

Repository ⓘ

Paste GitHub URL

ShaikhAmrin02/Model-deployment-on-cloud

Branch

main

Main file path

app.py

App URL (optional)

model-deployment-on-cloud-ztba5jfvb4jyxpsjqvih2s

.streamlit.app

Domain is available

Advanced settings

Deploy!

7. Deployed on Streamlit Cloud:

