Travel Demand Prediction Using Machine Learning

```
import pandas as pd
import seaborn as sns #makes it easy to plot Heatmaps
import matplotlib.pyplot as plt
from datetime import datetime

#Module - 1
DATA_FILE = "user_inquiries.csv"

def save_inquiry(data):
    try:
        df = pd.read_csv(DATA_FILE)
        df = pd.concat([df, pd.DataFrame([data])], ignore_index=True)
    except FileNotFoundError:
        df = pd.DataFrame([data])
    df.to_csv(DATA_FILE, index=False)



def get_user_input():
    print("\nPlease enter your travel inquiry details:")
    age = input("Age: ")
    gender = input("Gender (Male/Female/Other): ")
    occupation = input("Occupation: ")
    location = input("Travel Location: ")
    travel_date = input("Travel Date (YYYY-MM-DD): ")
    group_size = input("Group Size: ")
    inquiry_date = datetime.today().date().strftime('%Y-%m-%d')

    # Basic validation could be added here if needed
```

```
    return {
        "Age": age,
        "Gender": gender,
        "Occupation": occupation,
        "Location": location,
        "TravelDate": travel_date,
        "GroupSize": group_size,
        "InquiryDate": inquiry_date
    }

def show_heatmap():
    try:
        df = pd.read_csv(DATA_FILE)
    except FileNotFoundError:
        print("No inquiry data found. Please submit some inquiries first.")
        return

    df['InquiryDate'] = pd.to_datetime(df['InquiryDate']).dt.date.astype(str)
    heatmap_data = df.pivot_table(index='Location', columns='InquiryDate', aggfunc='size', fill_value=0)

    plt.figure(figsize=(12, 6))
    sns.heatmap(heatmap_data, annot=True, fmt='d', cmap='Blues')
    plt.title("Daily User Travel Inquiries Heatmap")
    plt.ylabel("Location")
    plt.xlabel("Inquiry Date")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
#Module -2
SEARCH_FILE = "search_behavior.csv"
```

```
def save_search_behavior(data):
    try:
        df = pd.read_csv(SEARCH_FILE)
        df = pd.concat([df, pd.DataFrame([data])], ignore_index=True)
    except FileNotFoundError:
        df = pd.DataFrame([data])
    df.to_csv(SEARCH_FILE, index=False)

def get_search_input():
    print("\nPlease enter search details:")
    keyword = input("Search Keyword: ")
    filters = input("Filters applied (e.g., price,weather,distance): ")
    time_spent = input("Time spent on search (in seconds): ")
    clicked = input("Clicked Destination: ")
    search_date = datetime.today().date().strftime('%Y-%m-%d')

    return {
        "SearchKeyword": keyword,
        "Filters": filters,
        "TimeSpent": time_spent,
        "ClickedDestination": clicked,
        "SearchDate": search_date
    }

def show_search_insights():
    try:
        df = pd.read_csv(SEARCH_FILE)
    except FileNotFoundError:
        print("No search data found. Please perform some searches first.")
```

```
        return

    # Most clicked destinations
    plt.figure(figsize=(10, 5))
    df['ClickedDestination'].value_counts().plot(kind='bar', color='skyblue')
    plt.title("Most Clicked Destinations")
    plt.xlabel("Destination")
    plt.ylabel("Clicks")
    plt.tight_layout()
    plt.show()


def show_search_analytics():
    try:
        df = pd.read_csv("search_behavior.csv")
    except FileNotFoundError:
        print("No search data available.")
        return

    # Prepare data
    df['SearchDate'] = pd.to_datetime(df['SearchDate']).dt.date
    top_keywords = df['SearchKeyword'].value_counts().head(5)
    top_destinations = df['ClickedDestination'].value_counts().head(5)
    search_counts_by_date = df['SearchDate'].value_counts().sort_index()
    filters_series = df['Filters'].str.split(',', expand=True).stack().str.strip()
    top_filters = filters_series.value_counts().head(5)

    # Create dashboard
    fig, axs = plt.subplots(2, 2, figsize=(14, 10))
    fig.suptitle("Travel Search Analytics Dashboard", fontsize=16, fontweight='bold', color='darkblue')
```

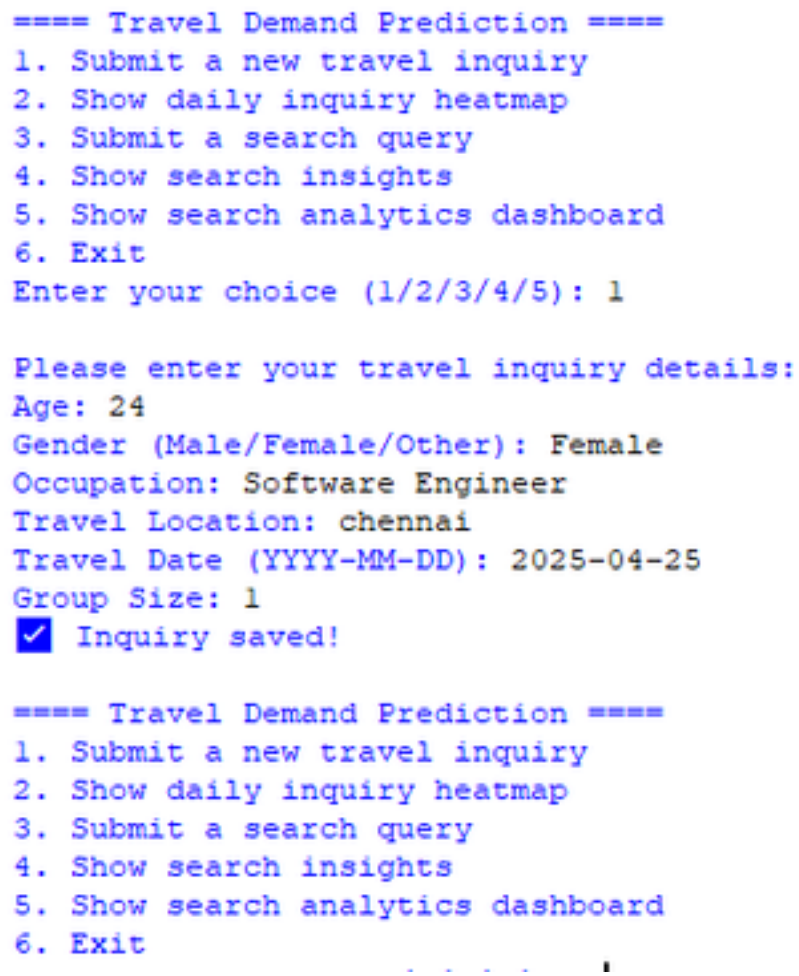
```
    # Bar Chart - Top Search Keywords
    axs[0, 0].bar(top_keywords.index, top_keywords.values, color='skyblue')
    axs[0, 0].set_title("Top Search Keywords", fontsize=12)
    axs[0, 0].set_ylabel("Frequency")
    axs[0, 0].tick_params(axis='x', rotation=45)

    # Bar Chart - Clicked Destinations
    axs[0, 1].bar(top_destinations.index, top_destinations.values, color='mediumseagreen')
    axs[0, 1].set_title("Top Clicked Destinations", fontsize=12)
    axs[0, 1].set_ylabel("Clicks")
    axs[0, 1].tick_params(axis='x', rotation=45)

    # Line Chart - Search Count Per Day
    axs[1, 0].plot(search_counts_by_date.index.astype(str), search_counts_by_date.values, marker='o',
color='coral')
    axs[1, 0].set_title("Searches Per Day", fontsize=12)
    axs[1, 0].set_ylabel("Number of Searches")
    axs[1, 0].tick_params(axis='x', rotation=45)

    # Horizontal Bar - Most Used Filters
    axs[1, 1].barh(top_filters.index[::-1], top_filters.values[::-1], color='orchid')
    axs[1, 1].set_title("Most Used Filters", fontsize=12)
    axs[1, 1].set_xlabel("Usage Count")

    plt.tight_layout(rect=[0, 0.03, 1, 0.95])  # Adjust layout to fit title
    plt.show()
def main():
    while True:
        print("\n==== Travel Demand Prediction ====")
```

```
print("1. Submit a new travel inquiry")
print("2. Show daily inquiry heatmap")
print("3. Submit a search query")
print("4. Show search insights")
print("5. Show search analytics dashboard")
print("6. Exit")
choice = input("Enter your choice (1/2/3/4/5): ")

if choice == "1":
    data = get_user_input()
    save_inquiry(data)
    print("✅ Inquiry saved!")
elif choice == "2":
    show_heatmap()
elif choice == "3":
    data = get_search_input()
    save_search_behavior(data)
    print("✅ Search behavior recorded!")
elif choice == "4":
    show_search_insights()
elif choice == "5":
    show_search_analytics()

elif choice == "6":
    print("Goodbye!")
    break

else:
    print("Invalid choice. Please enter 1, 2 ,3,4 or 5.")
```

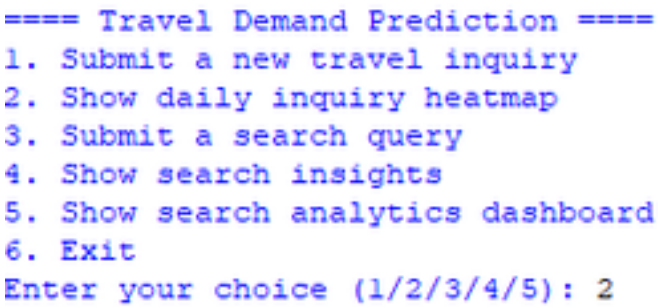
```
if __name__ == "__main__":
    main()
```

Outputs:

Step 1:



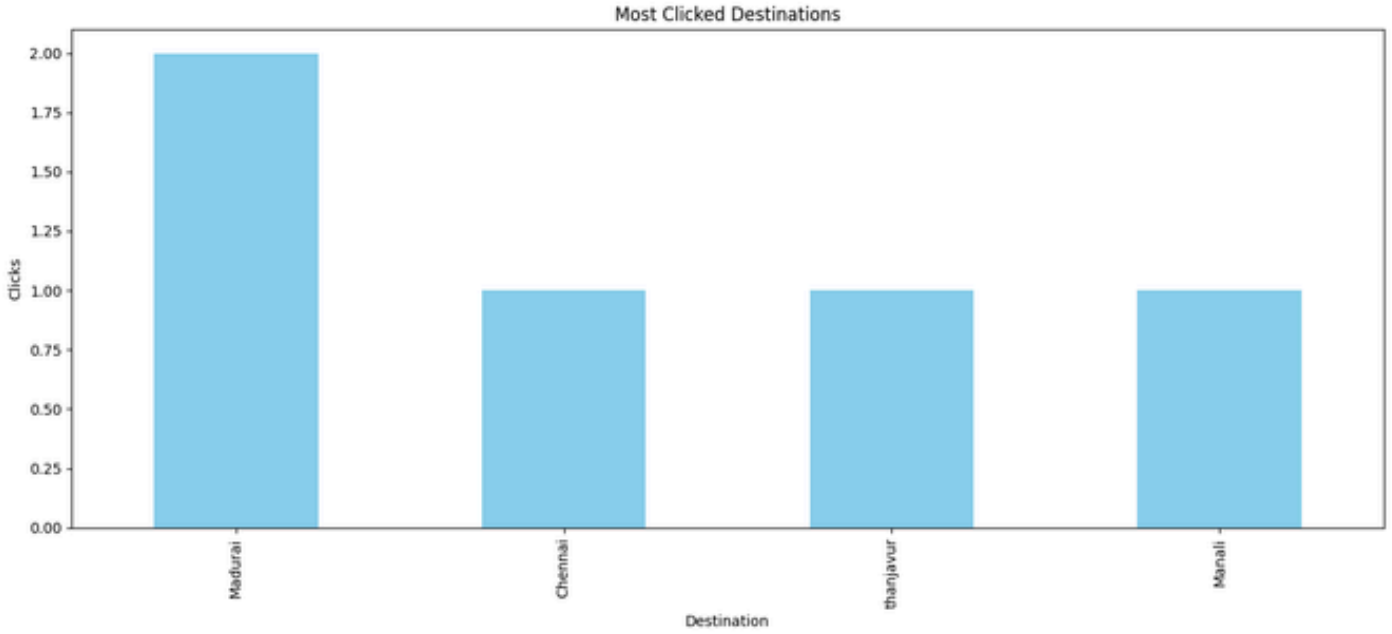
Step 2:



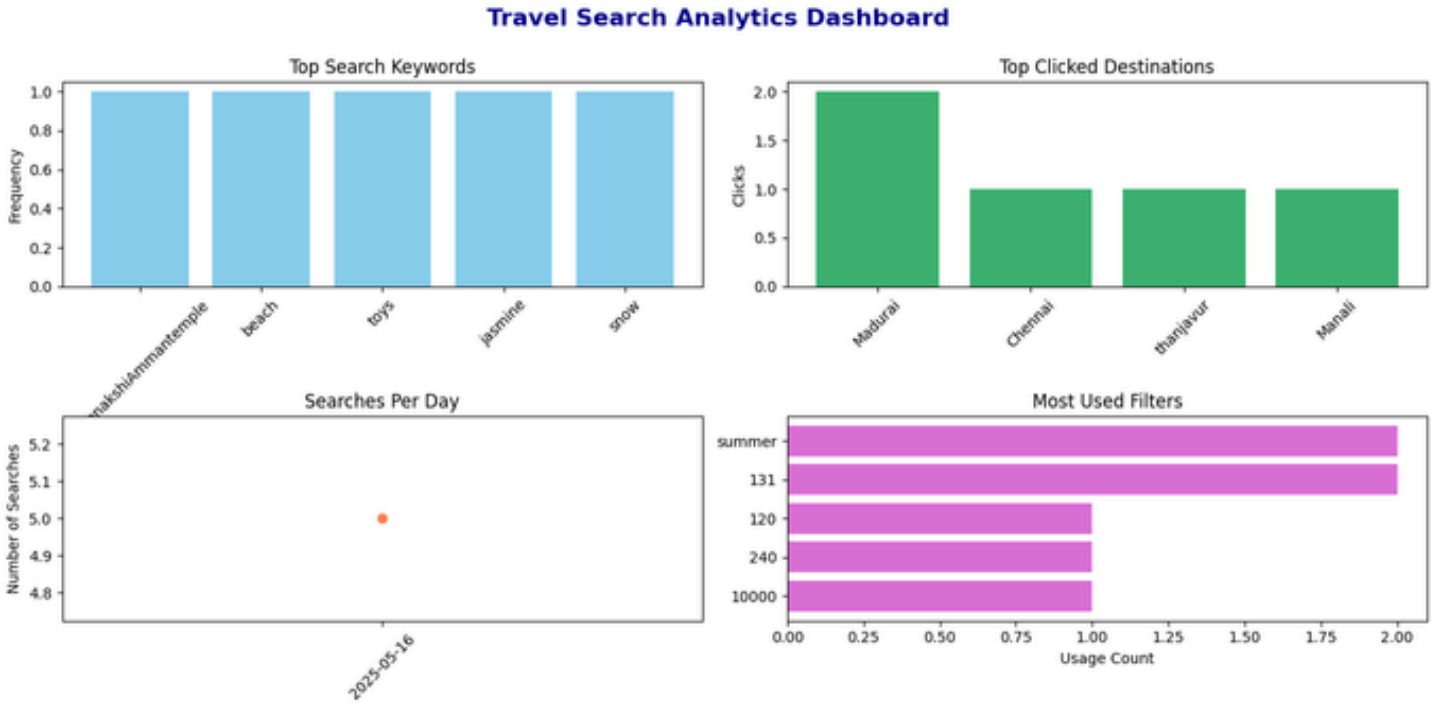
Step 3:

```
==== Travel Demand Prediction ====
1. Submit a new travel inquiry
2. Show daily inquiry heatmap
3. Submit a search query
4. Show search insights
5. Show search analytics dashboard
6. Exit
Enter your choice (1/2/3/4/5): 3

Please enter search details:
Search Keyword: snow
Filters applied (e.g., price,weather,distance): 10000
Time spent on search (in seconds): 40
Clicked Destination: Manali
✅ Search behavior recorded!
```

Step 4:


Most Clicked Destinations
Clicks
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Madurai
Chennai
thanjavur
Manali
Destination

Step 5:


Travel Search Analytics Dashboard
Top Search Keywords
Frequency
1.0
0.8
0.6
0.4
0.2
0.0
beach
toys
jasmine
snow
Top Clicked Destinations
Clicks
2.0
1.5
1.0
0.5
0.0
Madurai
Chennai
thanjavur
Manali
Searches Per Day
Number of Searches
5.2
5.1
5.0
4.9
4.8
2025-05-16
Most Used Filters
summer
131
120
240
10000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Usage Count