✈️ Airline Pricing Analysis – Python Portfolio Projcet Step-by-Step Plan

STEP-1: Load Dataset & Basic inspection

1. import Required Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# display settings
pd.set_option('display.max_columns', None)
```

2.Load the Dataset

```
df = pd.read_excel('/airline_price_.xlsx')
```

3.Quick Data inspection

```
df.head()
```

	Ticket_ID	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_l
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	
3	3	Vistara	UK-	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	

```
df.shape
```

```
(300153, 12)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Ticket_ID         300153 non-null  int64  
 1   airline           300153 non-null  object 
 2   flight            300153 non-null  object 
 3   source_city       300153 non-null  object 
 4   departure_time    300153 non-null  object 
 5   stops             300153 non-null  object 
 6   arrival_time      300153 non-null  object 
 7   destination_city  300153 non-null  object 
 8   class             300153 non-null  object 
 9   duration          300153 non-null  float64
 10  days_left         300153 non-null  int64  
 11  price             300153 non-null  int64  
dtypes: float64(1), int64(3), object(8)
memory usage: 27.5+ MB
```

```
df.describe()
```

	Ticket_ID	duration	days_left	price
count	300153.000000	300153.000000	300153.000000	300153.000000
mean	150076.000000	12.221021	26.004751	20889.660523
std	86646.852011	7.191997	13.561004	22697.767366
min	0.000000	0.830000	1.000000	1105.000000
25%	75038.000000	6.830000	15.000000	4783.000000
50%	150076.000000	11.250000	26.000000	7425.000000
75%	225114.000000	16.170000	38.000000	42521.000000
max	300152.000000	49.830000	49.000000	123071.000000

4.Check Missing values

```
df.isnull().sum()
```

	0
Ticket_ID	0
airline	0
flight	0
source_city	0
departure_time	0
stops	0
arrival_time	0
destination_city	0
class	0
duration	0
days_left	0
price	0

dtype: int64

5.Remove Duplicate Records

```
df.duplicated().sum()
df.drop_duplicates(inplace=True)
```

6.Rename Columns (Clean & Professional)

```
df.columns = df.columns.str.lower().str.replace(" ", "_")
df.head()
```

	ticket_id	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_l
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	
3	3	Vistara	UK-	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	

STEP-2: Data Cleaning & Feature Engineering (Python)

1. Handle Date_of_Journey → Datetime

Convert string date into datetime and extract useful features.

```
import datetime

# Assuming today's date as the reference point for calculating the date of journey.
# In a real-world scenario, you might have a 'booking_date' column.
reference_date = datetime.date.today()
df['date_of_journey'] = df['days_left'].apply(lambda x: reference_date + datetime.timedelta(days=int(x)))

# Convert to datetime objects and extract day and month
df['date_of_journey'] = pd.to_datetime(df['date_of_journey'])
df['journey_day'] = df['date_of_journey'].dt.day
df['journey_month'] = df['date_of_journey'].dt.month

df.head()
```

	ticket_id	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_l
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	

2. Clean Duration Column (VERY IMPORTANT)

Example values: 2h 50m,5h,1h 30m Function to convert duration → minutes

```
# The 'duration' column is already in hours as a float. To convert to minutes, multiply by 60.
df['duration_minutes'] = df['duration'] * 60
```

3. Clean Total_Stops

Values:non-stop,1 stop,2 stops

```
df['stops'] = df['stops'].astype(str).str.strip().str.lower().replace({
    'non-stop': 0,
    '1 stop': 1,
    '2 stops': 2,
    '3 stops': 3,
    '4 stops': 4,
    'zero': 0, # Added this line to handle 'zero' string
    'one': 1, # Added this line to handle 'one' string
    'two_or_more': 2 # Handle 'two_or_more' string
})

df['stops'] = df['stops'].astype(int)
```

```
/tmp/ipython-input-2205023048.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a
  df['stops'] = df['stops'].astype(str).str.strip().str.lower().replace({
```

4.Handle Missing values

```
df.isnull().sum()
```

	0
ticket_id	0
airline	0
flight	0
source_city	0
departure_time	0
stops	0
arrival_time	0
destination_city	0
class	0
duration	0
days_left	0
price	0
date_of_journey	0
journey_day	0
journey_month	0
duration_minutes	0

dtype: int64

5. Drop Unnecessary Columns

```
df.drop(['date_of_journey'], axis=1, inplace=True)
```

6.Final Dataset Check

```
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 15 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   ticket_id         300153 non-null  int64  
 1   airline           300153 non-null  object 
 2   flight            300153 non-null  object 
 3   source_city       300153 non-null  object 
 4   departure_time    300153 non-null  object 
 5   stops             300153 non-null  int64  
 6   arrival_time      300153 non-null  object 
 7   destination_city  300153 non-null  object 
 8   class             300153 non-null  object 
 9   duration          300153 non-null  float64
 10  days_left         300153 non-null  int64  
 11  price             300153 non-null  int64  
 12  journey_day       300153 non-null  int32  
 13  journey_month     300153 non-null  int32  
 14  duration_minutes  300153 non-null  float64
dtypes: float64(2), int32(2), int64(4), object(7)
memory usage: 32.1+ MB
```

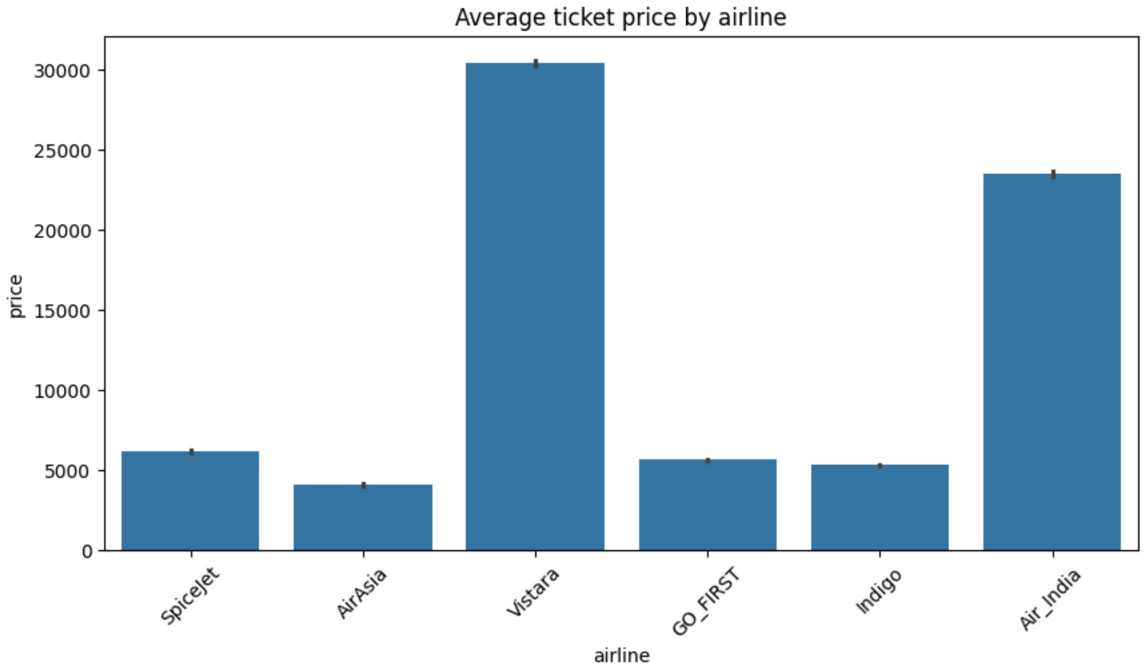
	ticket_id	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_l
0	0	SpiceJet	SG-8709	Delhi	Evening	0	Night	Mumbai	Economy	2.17	
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	0	Morning	Mumbai	Economy	2.33	
2	2	AirAsia	I5-764	Delhi	Early_Morning	0	Early_Morning	Mumbai	Economy	2.17	
3	3	Vistara	UK-995	Delhi	Morning	0	Afternoon	Mumbai	Economy	2.25	
4	4	Vistara	UK-963	Delhi	Morning	0	Morning	Mumbai	Economy	2.33	

📊 STEP 3: Exploratory Data Analysis (EDA)

1. Airline vs Average Ticket Price

Business Question: Which airline is most expensive?

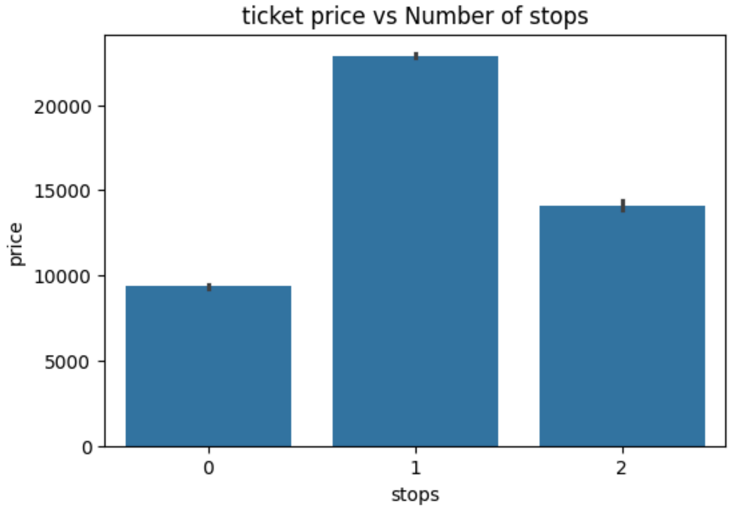
```
plt.figure(figsize=(10,5))
sns.barplot(x='airline', y='price',data=df)
plt.xticks(rotation=45)
plt.title("Average ticket price by airline")
plt.show()
```



2. Total Stops vs Ticket Price

Business Question: Do more stops reduce price?

```
plt.figure(figsize=(6,4))
sns.barplot(x='stops',y='price',data=df)
plt.title("ticket price vs Number of stops")
plt.show()
```



3. Monthly Price Trend

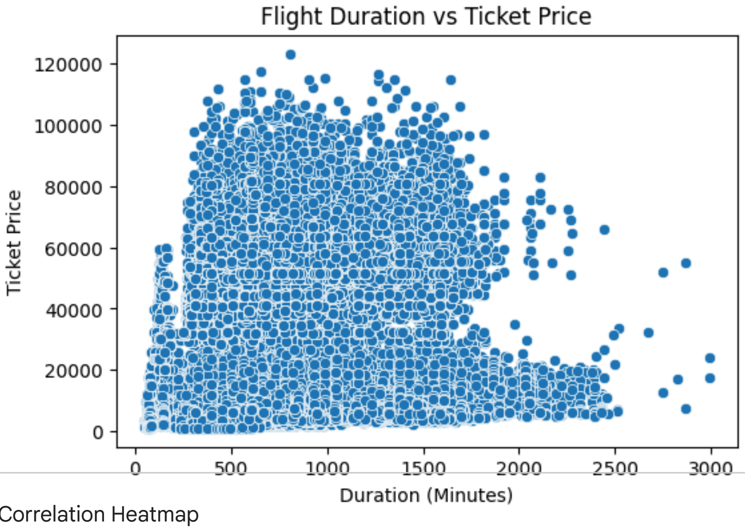
Business Question: Does seasonality affect pricing?

```
plt.figure(figsize=(8,4))
sns.lineplot(x='journey_month',y='price',data=df)
plt.title("monthly trend of flight prices")
plt.xlabel("month")
plt.ylabel("average price")
plt.show()
```

4. Duration vs Price

Business Question: Are longer flights more expensive?

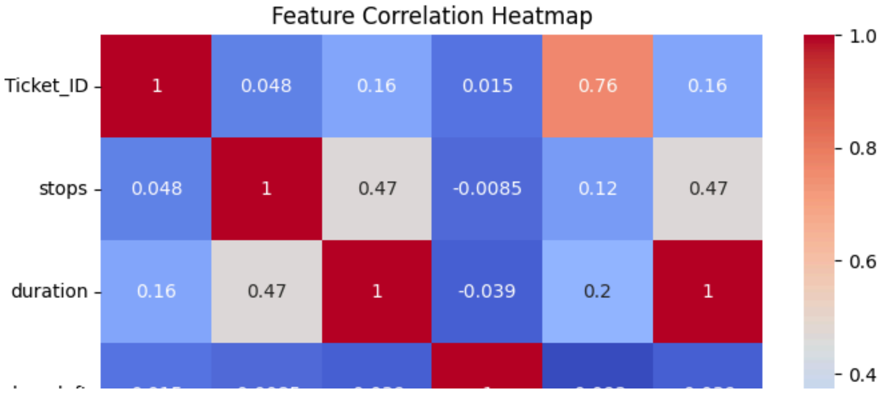
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(6,4))
sns.scatterplot(x='duration_minutes', y='price', data=df)
plt.title("Flight Duration vs Ticket Price")
plt.xlabel("Duration (Minutes)")
plt.ylabel("Ticket Price")
plt.show()
```



5. Correlation Heatmap

Business Question: Which factors influence price the most?

```
import numpy as np # Import numpy to resolve NameError
plt.figure(figsize=(8,6))
sns.heatmap(df.select_dtypes(include=np.number).corr(), annot=True, cmap='coolwarm')
plt.title("Feature Correlation Heatmap")
plt.show()
```


Feature Correlation Heatmap
Ticket_ID
stops
duration