**MAJOR PROJECT**

**DOMAIN : DATA SCIENCE**

Name：SUDHEKSHA . B

# 1 ) Import dataset :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings.csv")

data.head()

# 2 ) To check missing values :

print(data.isnull())

print(data.isnull().sum())

# 3 ) Key Features :

Booking Date :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings.csv")

print("Booking Date Analysis:\n")

print("Total bookings:", len(data))

Length of Stay :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings.csv")

data['total\_stay'] = data['stays\_in\_weekend\_nights'] + data['stays\_in\_week\_nights']

print(data['total\_stay'])

Pricing :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings (2).csv")

pricing\_data=data[['arrival\_date\_year','arrival\_date\_month','arrival\_date\_week\_number','arrival\_date\_day\_of\_month','market\_segment','distribution\_channel','deposit\_type','customer\_type','reserved\_room\_type','assigned\_room\_type','previous\_cancellations','previous\_bookings\_not\_canceled','adr']]

print(pricing\_data.head(10))

Demographics :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings (2).csv")

demographics\_data = data[['country', 'adults', 'children', 'babies', 'customer\_type']]

print(demographics\_data.head())

Special Request :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings.csv")

if 'total\_of\_special\_requests' in data.columns:

    print("\nSpecial Requests Distribution:")

    print(data['total\_of\_special\_requests'].value\_counts())

To convert Data fields into proper Date time Format :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings.csv")

data['reservation\_status\_date'] = pd.to\_datetime(data['reservation\_status\_date'], errors='coerce')

if {'arrival\_date\_year', 'arrival\_date\_month', 'arrival\_date\_day\_of\_month'}.issubset(data.columns):

    data['arrival\_date'] = pd.to\_datetime(data['arrival\_date\_year'].astype(str) + '-' +data['arrival\_date\_month'] +data['arrival\_date\_day\_of\_month'].astype(str),

      format='%Y-%B-%d',errors='coerce')

print(data[['reservation\_status\_date', 'arrival\_date']].head())

To check Numerical values are correctly formatted :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings.csv")

numerical\_columns=['stay\_in\_weekend\_nights','stay\_in\_week\_nights','adults','children','babies']

incorrect\_values={}

for col in numerical\_columns:

  if col in data.columns:

    incorrect\_values[col]=data[data[col]<0][col]

    print(incorrect\_values)

# 4 ) Cleaning Data set :

Remove Duplicates :

import pandas as pd

data=pd.read\_csv("/content/hotel bookings.csv")

data\_cleaned = data.drop\_duplicates()

Handling Outliers :

import pandas as pd

data = pd.read\_csv('/content/hotel bookings.csv')

price\_col = 'adr'

Q1 = data[price\_col].quantile(0.25)

Q3 = data[price\_col].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

data\_no\_outliers = data[(data[price\_col] >= lower\_bound) & (data[price\_col] <= upper\_bound)]

print(f"Original size: {data.shape}, After removing outliers: {data\_no\_outliers.shape}")

Drop irrelevant columns :

import pandas as pd

df = pd.read\_csv("/content/hotel bookings.csv")

print("Original Columns:")

print(df.columns.tolist())

irrelevant\_columns = [

    'agent',

    'company',

    'reservation\_status',

    'reservation\_status\_date',

    'assigned\_room\_type',

    'booking\_changes',

    'days\_in\_waiting\_list']

data\_cleaned = data.drop(columns=irrelevant\_columns)

print("\nCleaned Columns:")

print(data\_cleaned.columns.tolist())

# 5) Peak Booking Times :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv("/content/hotel bookings.csv")

month\_order = ['January', 'February', 'March', 'April', 'May', 'June','July', 'August', 'September', 'October', 'November', 'December']

data['arrival\_date\_month'] = pd.Categorical(data['arrival\_date\_month'], categories=month\_order, ordered=True)

plt.figure(figsize=(12, 6))

sns.boxplot(x='arrival\_date\_month', y='lead\_time', data=data)

plt.title('Distribution of Lead Time by Month (Peak Booking Time Indicator)')

plt.xlabel('Arrival Month')

plt.ylabel('Lead Time (days)')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

Optimal Length of stay :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data= pd.read\_csv("/content/hotel bookings.csv")

data['total\_stay'] = data['stays\_in\_weekend\_nights'] + data['stays\_in\_week\_nights']

data= data[data['total\_stay'] > 0]

stay\_vs\_rate = data.groupby('total\_stay')['adr'].mean().reset\_index()

plt.figure(figsize=(10, 6))

sns.lineplot(x='total\_stay', y='adr', data=stay\_vs\_rate)

plt.title("Average Daily Rate vs Total Stay Duration")

plt.xlabel("Total Nights Stayed")

plt.ylabel("Average Daily Rate (ADR)")

plt.tight\_layout()

plt.show()

Factors that drive special request :

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

data = pd.read\_csv("/content/hotel bookings.csv")

print(data.columns)

features = ['customer\_type', 'meal', 'market\_segment', 'deposit\_type','booking\_changes', 'is\_repeated\_guest', 'total\_of\_special\_requests']

data\_filtered = data[features]

Booking Time and Pricing Relationship :

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv("/content/hotel bookings.csv")

if 'lead\_time' not in data.columns or 'adr' not in data.columns:

    raise ValueError("Dataset must contain 'lead\_time' and 'adr' columns.")

grouped = data.groupby('lead\_time')['adr'].mean().reset\_index()

plt.figure(figsize=(12, 6))

sns.lineplot(data=grouped, x='lead\_time', y='adr')

plt.title('Relationship Between Booking Time (Lead Time) and Price (ADR)')

plt.xlabel('Lead Time (Days in Advance)')

plt.ylabel('Average Daily Rate (ADR)')

plt.grid(True)

plt.tight\_layout()

plt.show()

Stay Duration VS Daily rate :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('/content/hotel bookings.csv')

required\_cols = ['stays\_in\_weekend\_nights', 'stays\_in\_week\_nights', 'adr']

for col in required\_cols:

    if col not in data.columns:

        raise ValueError(f"Missing required column: {col}")

data['stay\_duration'] = data['stays\_in\_weekend\_nights'] + data['stays\_in\_week\_nights']

grouped = data.groupby('stay\_duration')['adr'].mean().reset\_index()

plt.figure(figsize=(12, 6))

sns.barplot(data=grouped, x='stay\_duration', y='adr', palette='Blues\_d')

plt.title('Stay Duration vs Average Daily Rate (ADR)')

plt.xlabel('Stay Duration (Total Nights)')

plt.ylabel('Average Daily Rate (ADR)')

plt.xticks(rotation=45)

plt.grid(axis='y')

plt.tight\_layout()

plt.show()

Impact on Booking :

import pandas as pd

data = pd.read\_csv("/content/hotel bookings.csv")

if 'is\_canceled' not in data.columns:

    raise ValueError("Dataset must contain 'is\_canceled' column.")

total\_bookings = len(data)

canceled\_bookings = data['is\_canceled'].sum()

not\_canceled = total\_bookings - canceled\_bookings

print(f"Total Bookings: {total\_bookings}")

print(f"Canceled Bookings: {canceled\_bookings} ({(canceled\_bookings/total\_bookings)\*100:.2f}%)")

print(f"Successful Bookings: {not\_canceled} ({(not\_canceled/total\_bookings)\*100:.2f}%)\n")

features = ['lead\_time', 'previous\_cancellations', 'booking\_changes',

            'total\_of\_special\_requests', 'is\_repeated\_guest']

print("Correlation of numeric features with cancellation:\n")

for feature in features:

    if feature in data.columns:

        corr = data['is\_canceled'].corr(data[feature])

        print(f"   - {feature}: {corr:.2f}")

categorical = ['customer\_type', 'deposit\_type', 'market\_segment', 'distribution\_channel']

# 6) Data Visualisation :

Booking Trends over time :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data=pd.read\_csv("/content/hotel bookings.csv")

data['total\_stay'] = data['stays\_in\_week\_nights'] + data['stays\_in\_weekend\_nights']

data['total\_price'] = data['adr'] \* data['total\_stay']

print(data[['adr', 'total\_stay', 'total\_price', 'reserved\_room\_type', 'meal', 'customer\_type']])

sns.boxplot(data=data, x='arrival\_date\_year', y='total\_price')

plt.xlabel("arrival year")

plt.ylabel("total price")

plt.title("booking trends over time")

plt.show()

Box plot for Price :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

sns.boxplot(data=data, x='hotel', y='adr')

plt.title('box plot on price')

plt.xlabel('Hotel Type')

plt.ylabel('Average Daily Rate (ADR)')

plt.grid(True, axis='y')

plt.tight\_layout()

plt.show()

Heat Map on Seasonal Demand :

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

data = pd.read\_csv('/content/hotel bookings.csv')

required = ['hotel', 'arrival\_date\_month']

for col in required:

    if col not in data.columns:

        raise ValueError(f"Dataset must contain '{col}' column.")

month\_order = ['January', 'February', 'March', 'April', 'May', 'June',

               'July', 'August', 'September', 'October', 'November', 'December']

pivot = data.groupby(['arrival\_date\_month', 'hotel']).size().unstack().reindex(month\_order)

plt.figure(figsize=(10, 6))

sns.heatmap(pivot, annot=True, fmt='d', cmap='Blues')

plt.title('Seasonal Booking Demand by Month')

plt.xlabel('Hotel Type')

plt.ylabel('Month')

plt.tight\_layout()

plt.show()

Bar Chart for Special Request :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('/content/hotel bookings.csv')

special\_req\_counts = data['total\_of\_special\_requests'].value\_counts().sort\_index()

plt.figure(figsize=(8, 5))

sns.barplot(x=special\_req\_counts.index, y=special\_req\_counts.values)

plt.title('Number of Bookings by Special Request Count')

plt.xlabel('Total Special Requests')

plt.ylabel('Number of Bookings')

plt.grid(axis='y')

plt.tight\_layout()

plt.show()

# 7 ) Model Evaluation :

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

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

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.metrics import mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

data = pd.read\_csv('/content/hotel bookings.csv')

data= data[data['adr'].notnull()]

target = 'adr'

features = ['lead\_time', 'arrival\_date\_month', 'stays\_in\_week\_nights', 'stays\_in\_weekend\_nights',

            'reserved\_room\_type', 'customer\_type', 'deposit\_type', 'meal', 'total\_of\_special\_requests']

data= data[features + [target]].dropna()

X = data[features]

y = data[target]

categorical\_features = ['arrival\_date\_month', 'reserved\_room\_type', 'customer\_type', 'deposit\_type', 'meal']

numeric\_features = list(set(features) - set(categorical\_features))

preprocessor = ColumnTransformer(

    transformers=[

        ('num', 'passthrough', numeric\_features),('cat', OneHotEncoder(drop='first'), categorical\_features)])

model = Pipeline(steps=[('preprocessor', preprocessor),  ('regressor', LinearRegression())])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print(" Model Evaluation:")

print(" - R² Score:", r2\_score(y\_test, y\_pred))

print(" - RMSE:", np.sqrt(mean\_squared\_error(y\_test, y\_pred)))

regressor = model.named\_steps['regressor']

feature\_names = numeric\_features+list(model.named\_steps['preprocessor'].transformers\_[1][1].get\_feature\_names\_out(categorical\_features))

coef\_df = pd.DataFrame({'Feature': feature\_names, 'Coefficient': regressor.coef\_})

coef\_df = coef\_df.sort\_values(by='Coefficient', key=abs, ascending=False)

plt.figure(figsize=(10, 6))

plt.barh(coef\_df['Feature'], coef\_df['Coefficient'])

plt.title('Feature Impact on ADR (Price)')

plt.xlabel('Regression Coefficient')

plt.gca().invert\_yaxis()

plt.tight\_layout()

plt.show()

Explanation of the code :

1. **LIBRARY IMPORTS :**

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

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

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.metrics import mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

**# Pandas and numpy :** These libraries are used for data manipulation and numerical operations

**# matplotlib :** it is used for data visualisation

**# sklearn :** used for machine learning tasks,including data splitting,scaling,modelling and evaluation

**SUMMARY :**

**DATA PREPARATION :** cleaning , converting types , feature engineering

**EDA :** Visualise to understand the data

**MODELLING :** splitting data , scaling features , training a regression model

**EVALUATION :** Assessing model performance and feature importance

This code provides a comprehensive analysis on hotel booking dataset,preparing for machine learning tasks and offering insights through visualisation.