NAME : PATHAN AMMAJAN

COLLEGE : KANDULA LAKSHUMMA MEMORIAL COLLEGE OF ENGINEERING

ROLL NO : 203H1A0537

COURSE : DATA SCIENCE

INTERN ID : BOV23FD0093

DATA SCIENCE WITH PYTHON

Final project work

Industry : hospitality

Department : manufacturing pain points

ABSTRACT

The project will involve an in-depth analysis of the manufacturing processes within the hospitality industry, focusing on areas such as procurement, inventory management, supply chain logistics, and quality control. By identifying and understanding the pain points in these processes, the project aims to develop innovative solutions that can streamline operations, reduce costs, and enhance overall productivity.

Furthermore, the project will investigate the impact of emerging technologies, such as automation, artificial intelligence, and data analytics, on the manufacturing aspects of the hospitality industry. These technologies have the potential to revolutionize operations by improving accuracy, minimizing errors, and enabling real-time decision-making. The project will explore how these advancements can be leveraged to optimize manufacturing processes and overcome existing challenges.

Keywords: Supply Chain Complexity, Quality Control, Staff Training and Skills, Seasonal Demand Variations, Inventory Management, Time constraints, Product customization, Compliance with regulations.

INTRODUCTION

The hospitality industry plays a vital role in providing services and accommodations to travellers, tourists, and guests. It encompasses a wide range of sectors, including hotels, restaurants, 43resorts, event planning, tourism agencies, and more. As with any industry, the hospitality sector faces its own set of manufacturing pain points that hinder its efficiency and overall success. In this project, we will explore these pain points and propose potential solutions to improve the manufacturing processes within the hospitality industry.

Hospitality refers to the relationship between guest and host; it also refers to being hospitable. Hospitality can be termed as a deliberate, planned and sustained effort to establish and maintain mutual understanding between an organization and the public.

the manufacturing sector supports this industry by providing the necessary products and equipment to enhance guest comfort and satisfaction. However, the hospitality manufacturing sector is not without its challenges and pain points, hindering its potential for growth and innovation. The manufacturing pain points in the hospitality industry refer to the challenges and difficulties faced by manufacturers who supply goods and equipment to the hospitality sector.

The hospitality industry heavily relies on the manufacturing sector to meet its diverse needs for products and supplies. Addressing the pain points within manufacturing is crucial for the industry to deliver exceptional guest experiences, maintain cost-efficiency, and adapt to changing consumer expectations. By understanding and mitigating these challenges, manufacturers can contribute to the growth and success of the hospitality industry as a whole.

INDUSTRY : HOSPITALITY

DEPARTMENT: MANUFACTURING MAIN POINTS

PROBLEM STATEMENT:

The hospitality industry, like any other industry, faces various problem statements and pain points. Here are some common challenges faced by the hospitality industry like labor shortage, rising labor cost, customer expectations, Competition from Sharing Economy, Online Reviews and Reputation Management, Security and Data Privacy, Supply Chain Management, Sustainability and Environmental Impact, Seasonality and Revenue Management.

These problem statements and pain points may vary depending on the specific segment of the hospitality industry, such as hotels, restaurants, event planning, or travel services.

PROBLEM SOLUTION:

To address the pain points in the hospitality industry's manufacturing processes, there are some solutions can be implemented like Technology Integration, Process Optimization, Workforce Training and Engagement, Sustainable Practices, Data-Driven Decision Making.

By addressing these problem areas and implementing the corresponding solutions, the hospitality industry can improve its manufacturing processes, optimize operations, and deliver enhanced customer experiences, leading to increased profitability and competitiveness.

INDUSTRY:HOSPITALITY

key components of the hospitality industry:

The hospitality industry is a broad sector that encompasses various businesses and services related to providing accommodation, food, and entertainment to guests.

Here are the key components of the hospitality industry:

1. Accommodation: This component includes hotels, resorts, motels, bed and breakfast establishments, hostels, vacation rentals, and other lodging options where guests can stay overnight or for an extended period.
2. Food and Beverage Services: This component covers restaurants, cafes, bars, lounges, catering services, and any establishment that offers food and drink to customers

There are some more key components like,

Event Planning and Management, Travel and Tourism,Recreation and Entertainment, Hospitality Technology, Customer Service and Guest Relations, Marketing and Sales, Sustainability and Environmental Practices, Human Resources and Training.

DEPARTMENTS : MANUFACTURING PAIN POINTS

In the context of hospitality in manufacturing, there can be several issues and pain points that establishments may face. Here are some common challenges:

1. Supply Chain Management: Manufacturing in the hospitality industry often involves managing complex supply chains. Ensuring timely delivery of raw materials, ingredients, and other necessary components can be a challenge, especially when dealing with multiple suppliers and coordinating with various departments within the establishment.

2. Quality Control: Maintaining consistent quality across different manufacturing processes is vital in the hospitality industry. It can be difficult to ensure that each product or ingredient meets the desired standards and specifications.

3. Cost Management: Managing manufacturing costs is crucial for profitability in the hospitality industry. Balancing expenses related to raw materials, labor, equipment, and energy consumption while maintaining product quality can be a significant challenge.

4. Waste Management: Hospitality manufacturing often generates significant amounts of waste, including food waste, packaging materials, and by-products.

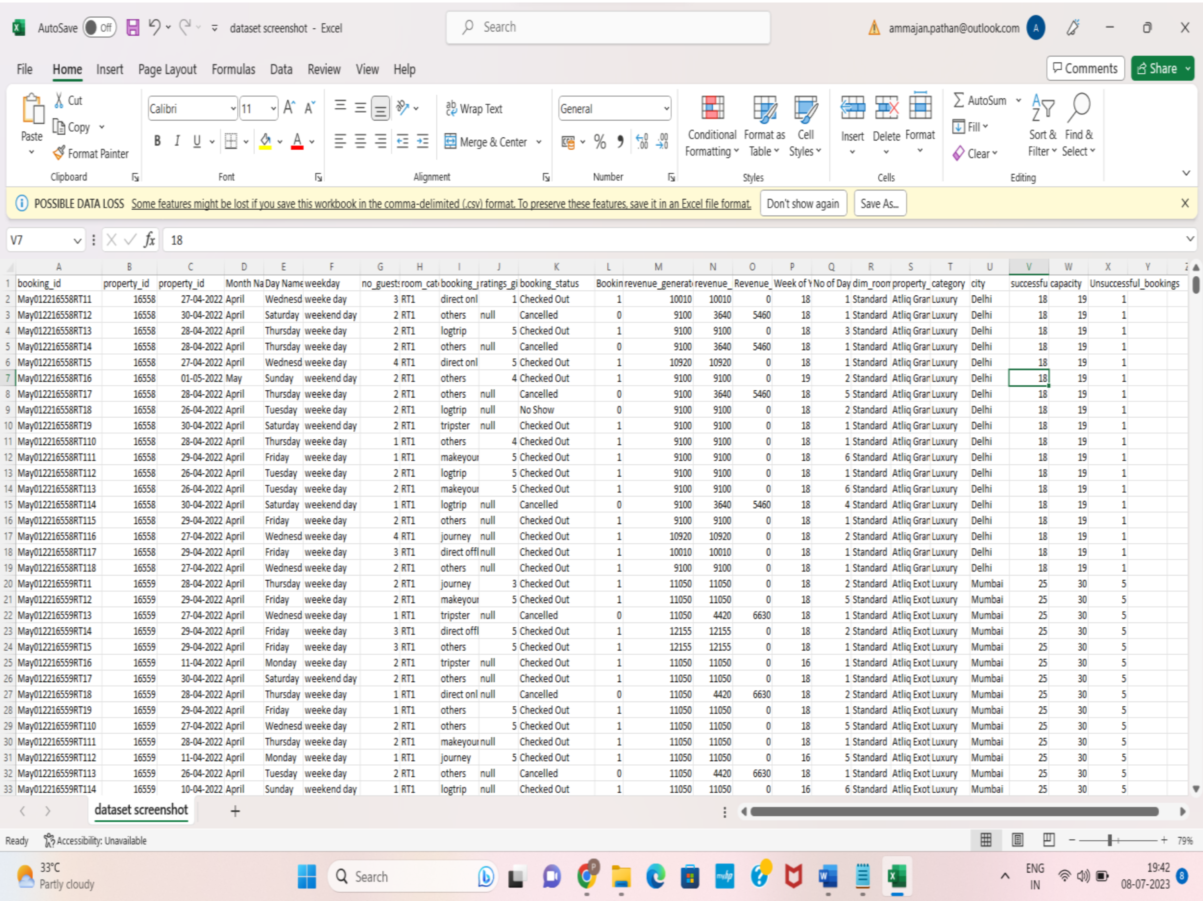
5. Scalability and Flexibility: Hospitality manufacturers need to adapt to changing demands, seasonal variations, and market trends.

6. Technology Integration: Embracing technology is essential for optimizing manufacturing processes in the hospitality industry. However, integrating new technologies, such as automation, robotics, or advanced data analytics, can be complex and requires

investment, training, and change management to ensure smooth implementation.

DATASET:

Hospitality: Data-Analysis and Predictions



	booking_id	property_id	property_id	Month Na	Day Name	weekday	no_guests	room_cat	booking_	ratings_gi	booking_status	Bookin	revenue_generat	revenue_	Revenue_	Week of Y	No of Day	dim_room	property_	category	city	successfu	capacity	Unsuccessful_bookings
2	May012216558RT11	16558	27-04-2022	April	Wednesd	weeke day	3	RT1	direct onl	1	Checked Out	1	10010	10010	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
3	May012216558RT12	16558	30-04-2022	April	Saturday	weekend day	2	RT1	others	null	Cancelled	0	9100	3640	5460	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
4	May012216558RT13	16558	28-04-2022	April	Thursday	weeke day	2	RT1	logtrip	5	Checked Out	1	9100	9100	0	18	3	Standard	Atliq Gran	Luxury	Delhi	18	19	1
5	May012216558RT14	16558	28-04-2022	April	Thursday	weeke day	2	RT1	others	null	Cancelled	0	9100	3640	5460	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
6	May012216558RT15	16558	27-04-2022	April	Wednesd	weeke day	4	RT1	direct onl	5	Checked Out	1	10920	10920	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
7	May012216558RT16	16558	01-05-2022	May	Sunday	weekend day	2	RT1	others	4	Checked Out	1	9100	9100	0	19	2	Standard	Atliq Gran	Luxury	Delhi	18	19	1
8	May012216558RT17	16558	28-04-2022	April	Thursday	weeke day	2	RT1	others	null	Cancelled	0	9100	3640	5460	18	5	Standard	Atliq Gran	Luxury	Delhi	18	19	1
9	May012216558RT18	16558	26-04-2022	April	Tuesday	weeke day	2	RT1	logtrip	null	No Show	0	9100	9100	0	18	2	Standard	Atliq Gran	Luxury	Delhi	18	19	1
10	May012216558RT19	16558	30-04-2022	April	Saturday	weekend day	2	RT1	tripster	null	Checked Out	1	9100	9100	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
11	May012216558RT110	16558	28-04-2022	April	Thursday	weeke day	1	RT1	others	4	Checked Out	1	9100	9100	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
12	May012216558RT111	16558	29-04-2022	April	Friday	weeke day	1	RT1	makeyour	5	Checked Out	1	9100	9100	0	18	6	Standard	Atliq Gran	Luxury	Delhi	18	19	1
13	May012216558RT112	16558	26-04-2022	April	Tuesday	weeke day	2	RT1	logtrip	5	Checked Out	1	9100	9100	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
14	May012216558RT113	16558	26-04-2022	April	Tuesday	weeke day	2	RT1	makeyour	5	Checked Out	1	9100	9100	0	18	6	Standard	Atliq Gran	Luxury	Delhi	18	19	1
15	May012216558RT114	16558	30-04-2022	April	Saturday	weekend day	1	RT1	logtrip	null	Cancelled	0	9100	3640	5460	18	4	Standard	Atliq Gran	Luxury	Delhi	18	19	1
16	May012216558RT115	16558	29-04-2022	April	Friday	weeke day	2	RT1	others	null	Checked Out	1	9100	9100	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
17	May012216558RT116	16558	27-04-2022	April	Wednesd	weeke day	4	RT1	journey	null	Checked Out	1	10920	10920	0	18	2	Standard	Atliq Gran	Luxury	Delhi	18	19	1
18	May012216558RT117	16558	29-04-2022	April	Friday	weeke day	3	RT1	direct offl	null	Checked Out	1	10010	10010	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
19	May012216558RT118	16558	27-04-2022	April	Wednesd	weeke day	2	RT1	others	null	Checked Out	1	9100	9100	0	18	1	Standard	Atliq Gran	Luxury	Delhi	18	19	1
20	May012216559RT11	16559	28-04-2022	April	Thursday	weeke day	2	RT1	journey	3	Checked Out	1	11050	11050	0	18	2	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
21	May012216559RT12	16559	29-04-2022	April	Friday	weeke day	2	RT1	makeyour	5	Checked Out	1	11050	11050	0	18	5	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
22	May012216559RT13	16559	27-04-2022	April	Wednesd	weeke day	1	RT1	tripster	null	Cancelled	0	11050	4420	6630	18	1	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
23	May012216559RT14	16559	29-04-2022	April	Friday	weeke day	3	RT1	direct offl	5	Checked Out	1	12155	12155	0	18	2	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
24	May012216559RT15	16559	29-04-2022	April	Friday	weeke day	3	RT1	others	5	Checked Out	1	12155	12155	0	18	1	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
25	May012216559RT16	16559	11-04-2022	April	Monday	weeke day	2	RT1	tripster	null	Checked Out	1	11050	11050	0	16	1	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
26	May012216559RT17	16559	30-04-2022	April	Saturday	weekend day	2	RT1	others	null	Checked Out	1	11050	11050	0	18	1	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
27	May012216559RT18	16559	28-04-2022	April	Thursday	weeke day	1	RT1	direct onl	null	Cancelled	0	11050	4420	6630	18	2	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
28	May012216559RT19	16559	29-04-2022	April	Friday	weeke day	1	RT1	others	5	Checked Out	1	11050	11050	0	18	1	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
29	May012216559RT110	16559	27-04-2022	April	Wednesd	weeke day	2	RT1	others	5	Checked Out	1	11050	11050	0	18	5	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
30	May012216559RT111	16559	28-04-2022	April	Thursday	weeke day	2	RT1	makeyour	null	Checked Out	1	11050	11050	0	18	1	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
31	May012216559RT112	16559	11-04-2022	April	Monday	weeke day	1	RT1	journey	5	Checked Out	1	11050	11050	0	16	5	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
32	May012216559RT113	16559	26-04-2022	April	Tuesday	weeke day	2	RT1	others	null	Cancelled	0	11050	4420	6630	18	1	Standard	Atliq Exot	Luxury	Mumbai	25	30	5
33	May012216559RT114	16559	10-04-2022	April	Sunday	weekend day	1	RT1	logtrip	null	Checked Out	1	11050	11050	0	16	6	Standard	Atliq Exot	Luxury	Mumbai	25	30	5

Pre-Processing Data

Data pre-processing is an essential step in preparing the dataset for analysis.

Indeed, data pre-processing plays a crucial role in preparing datasets for analysis in various domains, including hospitality and manufacturing. By performing data pre-processing tasks, you can enhance the quality of your data, address inconsistencies, and make it suitable for analysis. Here are some key steps in data pre-processing that are relevant to the hospitality manufacturing industry:

1. Data Cleaning: This step involves identifying and handling missing, incorrect, or inconsistent data. It may include tasks such as removing duplicates, filling in missing values, and correcting any errors or outliers in the dataset.

2. Data Integration: In hospitality manufacturing, data often comes from various sources and systems. Data integration involves combining data from different sources into a unified dataset.

3. Data Transformation: This step involves transforming the data into a suitable format for analysis.

4. Feature Selection/Extraction: Feature selection aims to identify the most relevant features or variables for analysis. It helps in reducing dimensionality, eliminating noise, and improving model performance.

5. Data Reduction: In some cases, datasets in hospitality manufacturing can be large and contain redundant or irrelevant information.

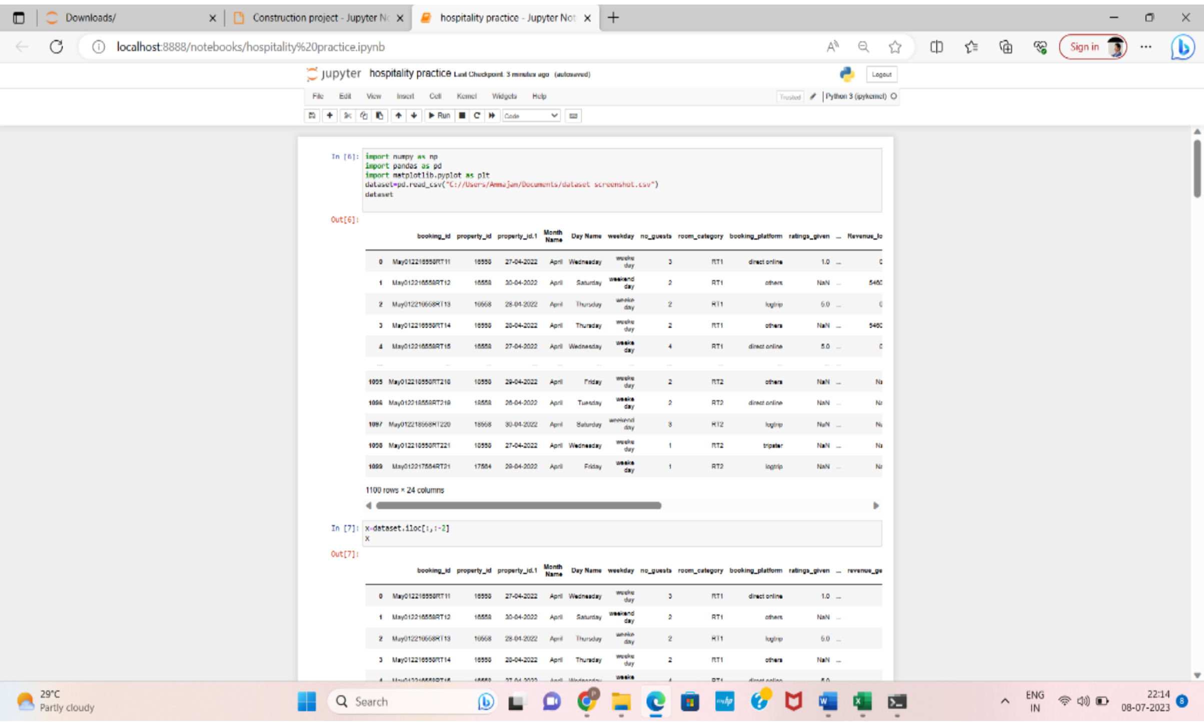
6. Data Formatting: Finally, the data needs to be formatted appropriately for the analysis techniques or models you plan to use.

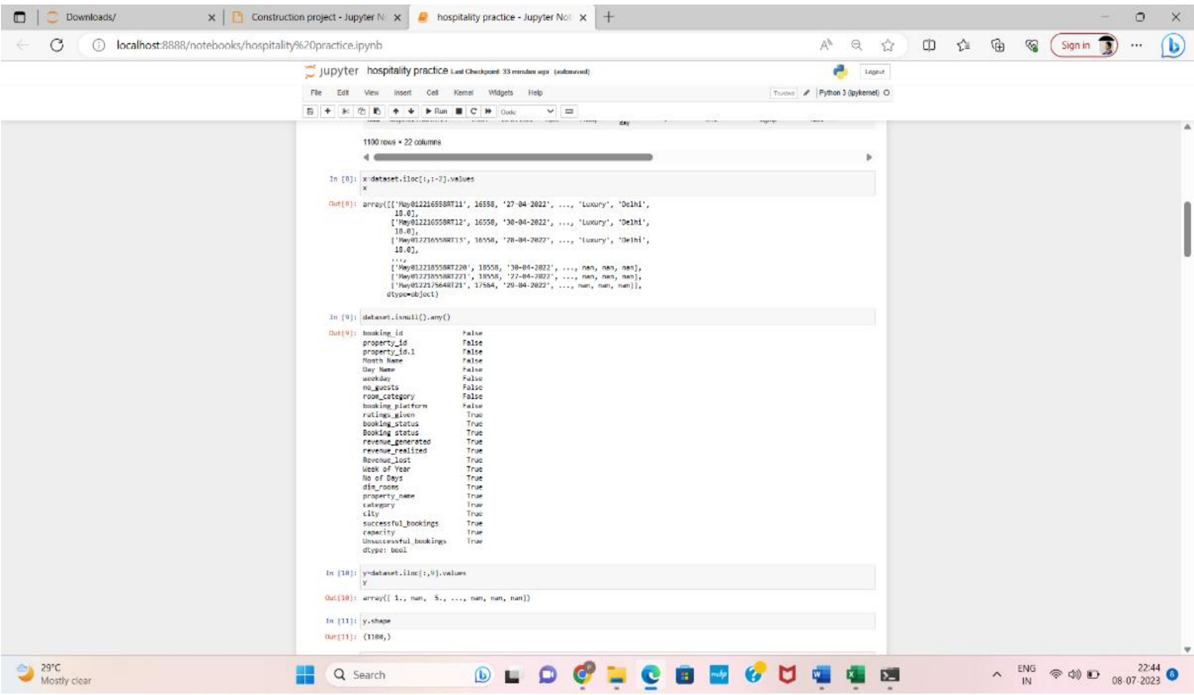
Construction data analysis

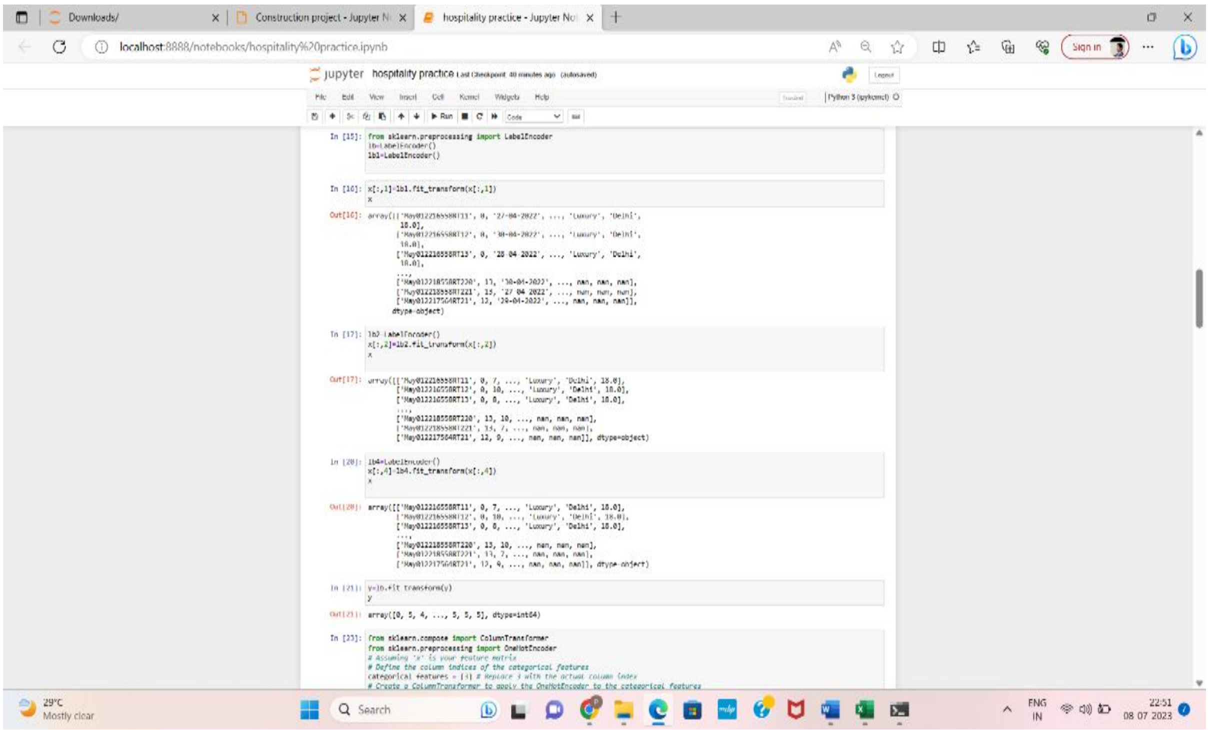
CONTENTS

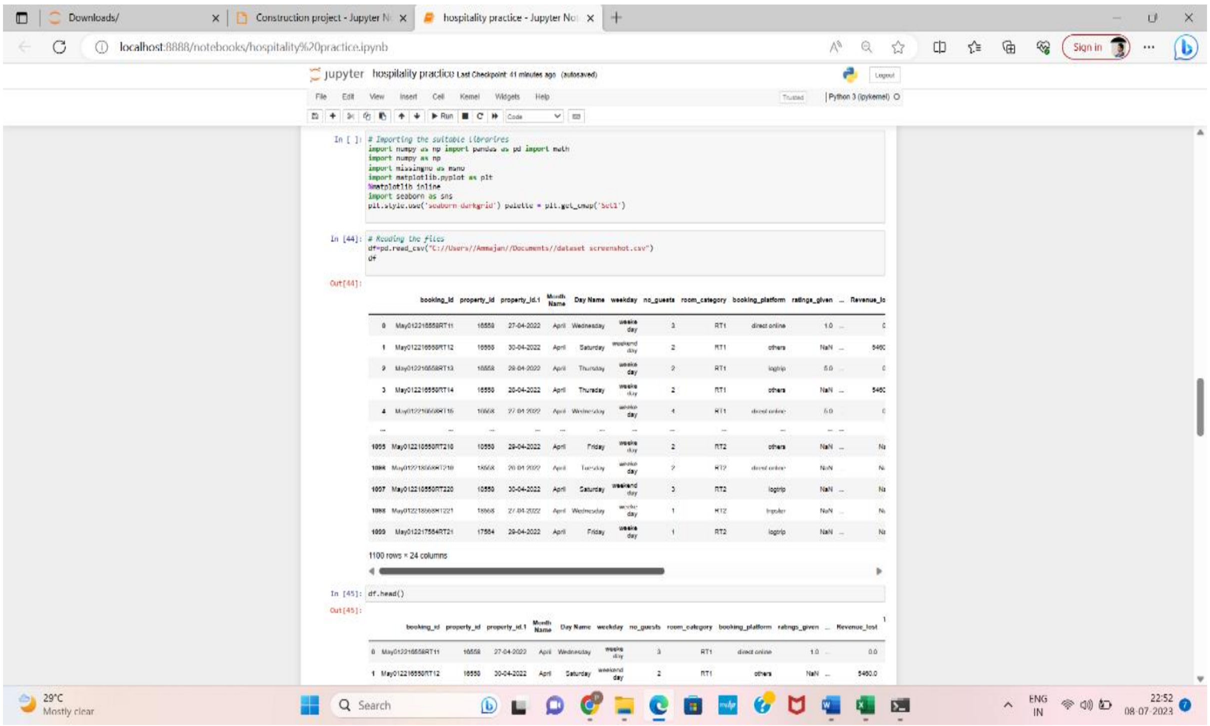
- Importing
- the suitable libraries
- Reading the files
- Cleaning the data
- Visualisation of the data
- Conclusion made from the data

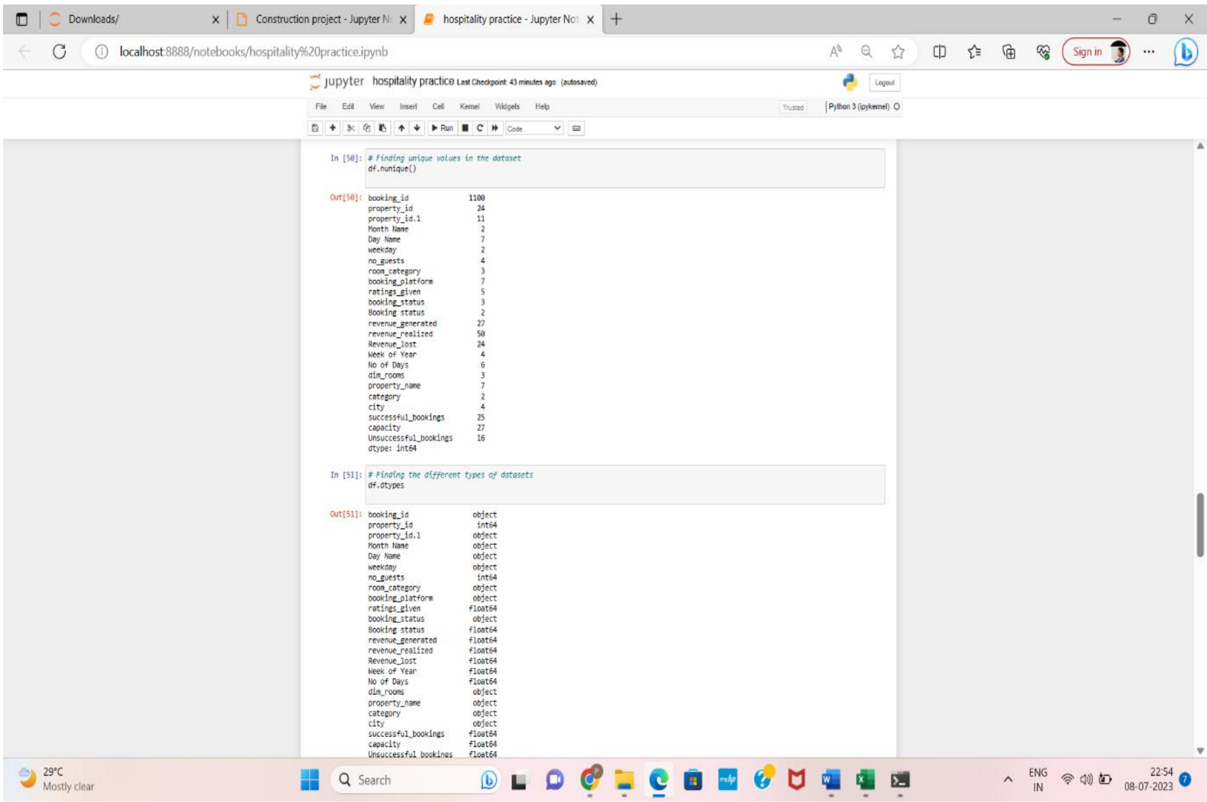
CODE:

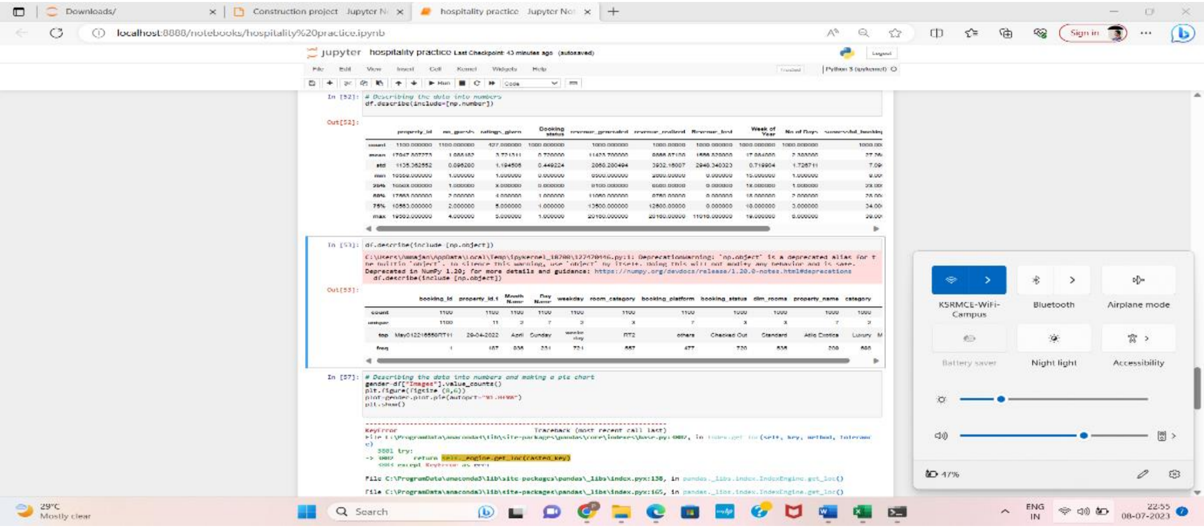



jupyter hospitality practice Last Checkpoint: 33 minutes ago (autosaved)
1100 rows × 22 columns
In [8]: x=dataset.iloc[:,:-2].values
x
Out[8]: array([['May012216558RT11', 16558, '27-04-2022', ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT12', 16558, '30-04-2022', ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT13', 16558, '28-04-2022', ..., 'Luxury', 'Delhi', 18.0],
...,
['May012218558RT220', 18558, '30-04-2022', ..., nan, nan, nan],
['May012218558RT221', 18558, '27-04-2022', ..., nan, nan, nan],
['May012217564RT21', 17564, '29-04-2022', ..., nan, nan, nan]],
dtype=object)
In [9]: dataset.isnull().any()
Out[9]: booking_id False
property_id False
property_id.1 False
Month Name False
Day Name False
weekday False
no_guests False
room_category False
booking_platform False
ratings_given True
booking_status True
Booking status True
revenue_generated True
revenue_realized True
Revenue_lost True
Week of Year True
No of Days True
dim_rooms True
property_name True
category True
city True
successful_bookings True
capacity True
Unsuccessful_bookings True
dtype: bool
In [10]: y=dataset.iloc[:,9].values
y
Out[10]: array([1., nan, 5., ..., nan, nan, nan])
In [11]: y.shape
Out[11]: (1100,)

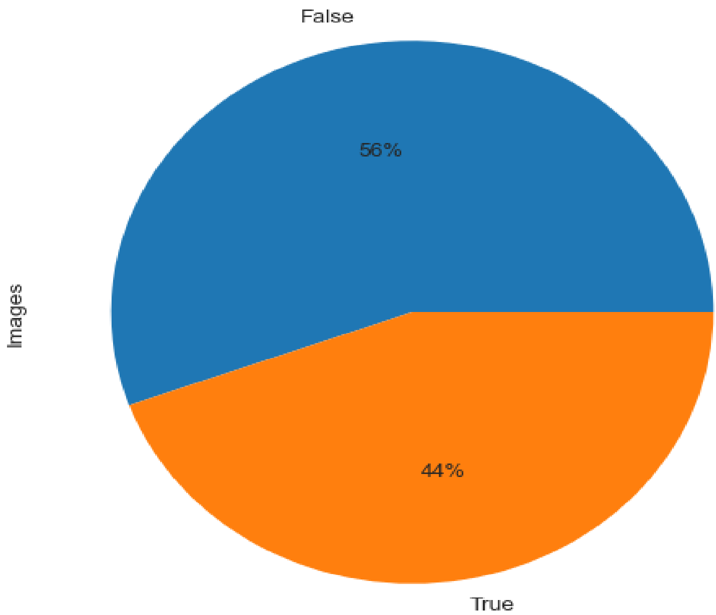

jupyter hospitality practice Last Checkpoint: 40 minutes ago (autosaved)
In [15]: from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
lb1=LabelEncoder()
In [16]: x[:,1]=lb1.fit_transform(x[:,1])
x
Out[16]: array([['May012216558RT11', 0, '27-04-2022', ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT12', 0, '30-04-2022', ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT13', 0, '28-04-2022', ..., 'Luxury', 'Delhi', 18.0],
...,
['May012218558RT220', 13, '30-04-2022', ..., nan, nan, nan],
['May012218558RT221', 13, '27-04-2022', ..., nan, nan, nan],
['May012217564RT21', 12, '29-04-2022', ..., nan, nan, nan]],
dtype=object)
In [17]: lb2=LabelEncoder()
x[:,2]=lb2.fit_transform(x[:,2])
x
Out[17]: array([['May012216558RT11', 0, 7, ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT12', 0, 10, ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT13', 0, 8, ..., 'Luxury', 'Delhi', 18.0],
...,
['May012218558RT220', 13, 10, ..., nan, nan, nan],
['May012218558RT221', 13, 7, ..., nan, nan, nan],
['May012217564RT21', 12, 9, ..., nan, nan, nan]], dtype=object)
In [20]: lb4=LabelEncoder()
x[:,4]=lb4.fit_transform(x[:,4])
x
Out[20]: array([['May012216558RT11', 0, 7, ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT12', 0, 10, ..., 'Luxury', 'Delhi', 18.0],
['May012216558RT13', 0, 8, ..., 'Luxury', 'Delhi', 18.0],
...,
['May012218558RT220', 13, 10, ..., nan, nan, nan],
['May012218558RT221', 13, 7, ..., nan, nan, nan],
['May012217564RT21', 12, 9, ..., nan, nan, nan]], dtype=object)
In [21]: y=lb.fit_transform(y)
y
Out[21]: array([0, 5, 4, ..., 5, 5, 5], dtype=int64)
In [23]: from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
# Assuming 'x' is your feature matrix
# Define the column indices of the categorical features
categorical_features = [3] # Replace 3 with the actual column index
# Create a ColumnTransformer to apply the OneHotEncoder to the categorical features


Downloads/
Construction project - Jupyter N
hospitality practice - Jupyter No
localhost:8888/notebooks/hospitality%20practice.ipynb
jupyter hospitality practice Last Checkpoint: 41 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
In [ ]: # Importing the suitable librarires
import numpy as np import pandas as pd import math
import numpy as np
import missingno as msno
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.style.use('seaborn-darkgrid') palette = plt.get_cmap('Set1')
In [44]: # Reading the files
df=pd.read_csv("C://Users//Ammajan//Documents//dataset screenshot.csv")
df
Out[44]:
booking_id property_id property_id.1 Month Name Day Name weekday no_guests room_category booking_platform ratings_given ... Revenue_lo
0 May012216558RT11 16558 27-04-2022 April Wednesday weeke day 3 RT1 direct online 1.0 ...
1 May012216558RT12 16558 30-04-2022 April Saturday weekend day 2 RT1 others NaN ...
1100 rows × 24 columns
In [45]: df.head()
Out[45]:
booking_id property_id property_id.1 Month Name Day Name weekday no_guests room_category booking_platform ratings_given ... Revenue_lost
29°C
Mostly clear
Search
ENG
IN
22:52
08-07-2023


Downloads/
Construction project - Jupyter N
hospitality practice - Jupyter No
localhost:8888/notebooks/hospitality%20practice.ipynb
jupyter hospitality practice Last Checkpoint: 43 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
In [50]: # Finding unique values in the dataset
df.nunique()
Out[50]: booking_id 1100
property_id 24
property_id.1 11
Month Name 2
Day Name 7
weekday 2
no_guests 4
room_category 3
booking_platform 7
ratings_given 5
booking_status 3
Booking status 2
revenue_generated 27
revenue_realized 50
Revenue_lost 24
Week of Year 4
No of Days 6
dim_rooms 3
property_name 7
category 2
city 4
successful_bookings 25
capacity 27
Unsuccessful_bookings 16
dtype: int64
In [51]: # Finding the different types of datasets
df.dtypes
Out[51]: booking_id object
property_id int64
property_id.1 object
Month Name object
Day Name object
weekday object
no_guests int64
room_category object
booking_platform object
ratings_given float64
booking_status object
Booking status float64
revenue_generated float64
revenue_realized float64
Revenue_lost float64
Week of Year float64
No of Days float64
dim_rooms object
property_name object
category object
city object
successful_bookings float64
capacity float64
Unsuccessful_bookings float64
29°C
Mostly clear
Search
ENG
IN
22:54
08-07-2023




```
# Describing the data into numbers and making a pie chart
gender=df["Images"].value_counts()
plt.figure(figsize=(8,6))
plot=gender.plot.pie(autopct="%1.0f%%")
plt.show()
```



Prediction Modelling Approach

The Prediction Modeling Approach in the hospitality industry refers to using statistical and analytical techniques to forecast and predict various aspects of the industry, such as demand for rooms, customer preferences, pricing trends, and operational needs. This approach relies on historical data, market trends, and other relevant factors to develop models that can generate accurate predictions for future events or outcomes.

In the hospitality industry, prediction modeling can be used for several purposes:

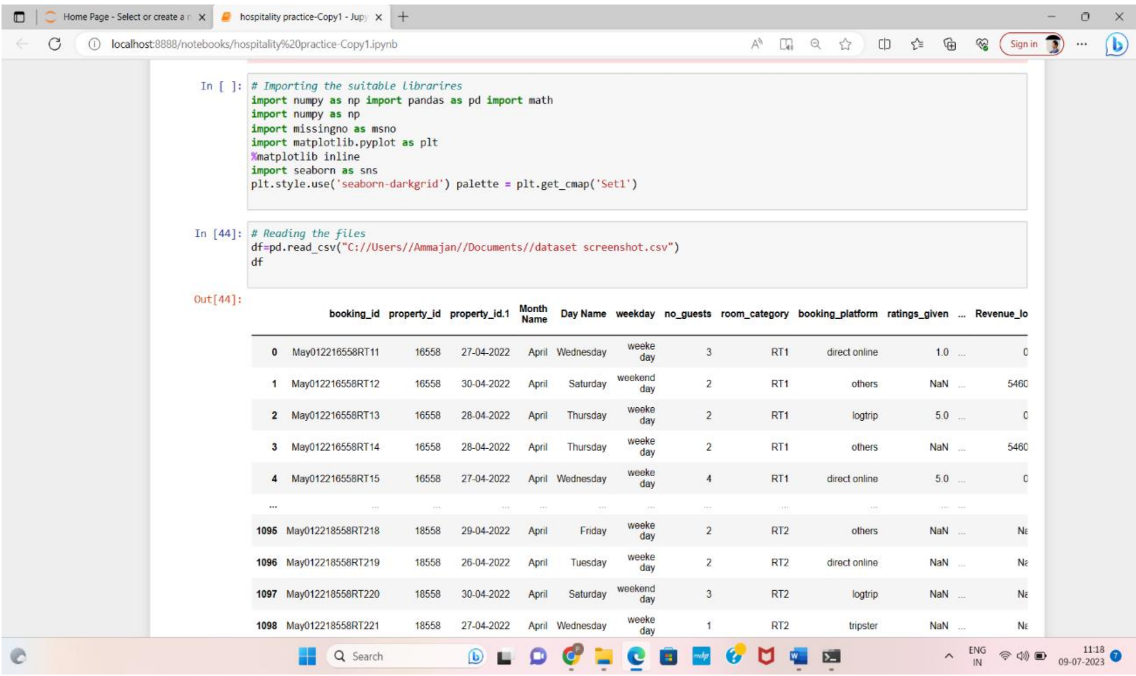
1. Demand forecasting: By analyzing historical data on room bookings, seasonal trends, local events, and other factors, hotels can predict future demand for rooms. This helps them optimize their pricing strategies, allocate resources efficiently, and make informed decisions about inventory management.
2. Revenue management: Prediction modeling enables hotels to optimize their pricing and revenue strategies. By analyzing various factors like demand, competitor pricing, customer segments, and booking patterns, hotels can dynamically adjust their room rates to maximize revenue and occupancy rates.
3. Customer segmentation and targeting: Prediction modeling can identify patterns and preferences among different customer segments. By analyzing customer data, hotels can personalize marketing campaigns, tailor services to specific customer needs, and optimize customer acquisition and retention strategies.
4. Operational planning: Prediction modeling can assist hotels in predicting operational needs, such as staffing requirements, food and beverage demand, and housekeeping needs. By forecasting these aspects accurately, hotels can optimize their resource allocation, reduce costs, and enhance operational efficiency.

Regarding manufacturing pain points, there are several challenges that the manufacturing industry often faces. Some common pain points include:

1. Supply chain management: Manufacturers often struggle with managing complex supply chains involving multiple suppliers, transportation, and inventory management. Issues such as delays, disruptions, and lack of visibility can impact production schedules and lead to inefficiencies.
2. Production planning and scheduling: Manufacturers need to balance production capacity, customer demand, and inventory levels.
3. Quality control: Maintaining consistent product quality is crucial for manufacturers. Defective products, rework, and product recalls can lead to significant costs and damage the reputation of the company.
4. Cost management: Manufacturers face the challenge of managing costs effectively while maintaining competitiveness. Rising raw material prices, labor costs, energy costs, and overhead expenses can impact profitability.
5. Technology adoption and automation: Keeping up with technological advancements and leveraging automation can be a challenge for manufacturing companies.
6. Regulatory compliance: Compliance with industry regulations and standards is crucial for manufacturers.

   Addressing these pain points often involves leveraging advanced technologies, implementing efficient processes, optimizing supply chain management, investing in training and development, and adopting a data-driven approach to decision-making.

CODE:



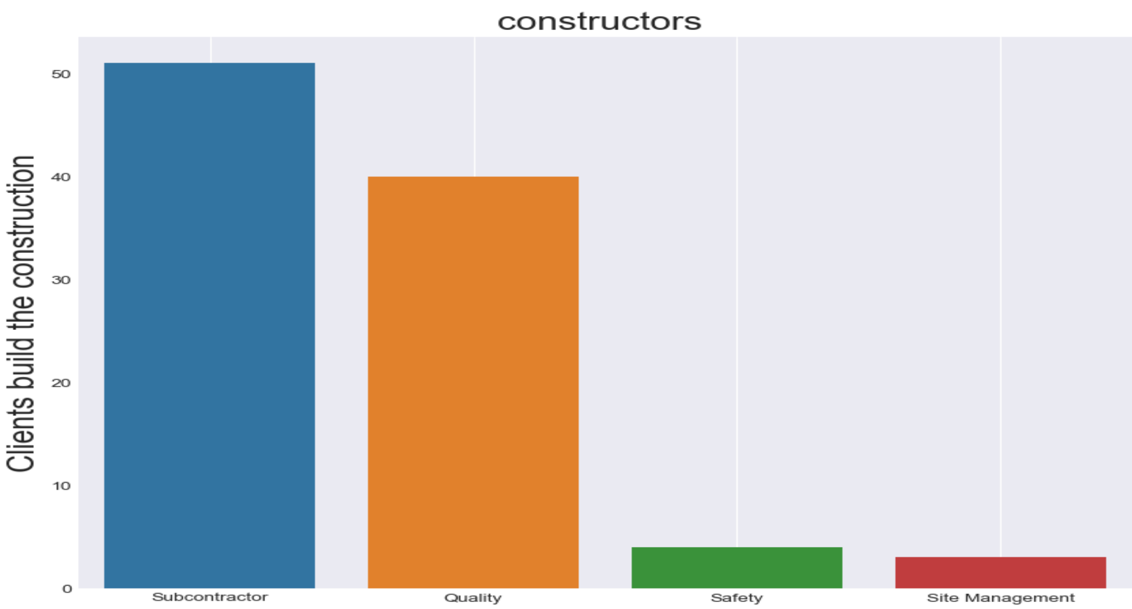
```
In [ ]: # Importing the suitable librarires
import numpy as np import pandas as pd import math
import numpy as np
import missingno as msno
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.style.use('seaborn-darkgrid') palette = plt.get_cmap('Set1')

In [44]: # Reading the files
df=pd.read_csv("C://Users//Ammajan//Documents//dataset screenshot.csv")
df

Out[44]:
```

	booking_id	property_id	property_id.1	Month Name	Day Name	weekday	no_guests	room_category	booking_platform	ratings_given	...	Revenue_lo
0	May012216558RT11	16558	27-04-2022	April	Wednesday	weeke day	3	RT1	direct online	1.0	...	0
1	May012216558RT12	16558	30-04-2022	April	Saturday	weekend day	2	RT1	others	NaN	...	5460
2	May012216558RT13	16558	28-04-2022	April	Thursday	weeke day	2	RT1	logtrip	5.0	...	0
3	May012216558RT14	16558	28-04-2022	April	Thursday	weeke day	2	RT1	others	NaN	...	5460
4	May012216558RT15	16558	27-04-2022	April	Wednesday	weeke day	4	RT1	direct online	5.0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	
1095	May012218558RT218	18558	29-04-2022	April	Friday	weeke day	2	RT2	others	NaN	...	Na
1096	May012218558RT219	18558	26-04-2022	April	Tuesday	weeke day	2	RT2	direct online	NaN	...	Na
1097	May012218558RT220	18558	30-04-2022	April	Saturday	weekend day	3	RT2	logtrip	NaN	...	Na
1098	May012218558RT221	18558	27-04-2022	April	Wednesday	weeke day	1	RT2	tripster	NaN	...	Na

```
# Finding the phone service used by the clients
phone_services=df['Report Forms Group'].value_counts()
plt.figure(figsize=(10,8))
sns.barplot(x=phone_services.index,y=phone_services.values)
plt.ylabel('Clients build the construction',size=20)
plt.grid()
plt.title('constructors ',size=20)
plt.show()
```



Model Training and Evaluation

In the context of hospitality and manufacturing, model training and evaluation can have their own unique pain points. Here are some common challenges faced in these industries:

Hospitality:

1. Data quality and availability: Obtaining high-quality and relevant data in the hospitality industry can be a challenge. Data may be scattered across multiple sources and might be incomplete or inconsistent, making it difficult to train accurate models.
2. Data privacy and security: Hospitality businesses deal with sensitive customer information, and ensuring data privacy and security during model training can be a significant concern.
3. Seasonality and dynamic trends: The hospitality industry is highly influenced by seasonal variations, events, and changing consumer preferences.
4. Feature engineering: Identifying the right features to train models for hospitality-related tasks, such as demand forecasting, customer sentiment analysis, or recommendation systems, can be challenging.
5. Scalability and real-time predictions: Hospitality businesses often require real-time predictions to optimize operations and provide personalized experiences.

Manufacturing:

1. Data integration and standardization: Manufacturing involves various stages and processes, generating diverse data from different systems and equipment.
2. Imbalanced data: Manufacturing datasets often suffer from class imbalance, where certain events or outcomes are infrequent compared to others.

3. Interpretability and explainability: In manufacturing, understanding the reasons behind model predictions is crucial. The interpretability and explainability of models become pain points when complex models, such as deep learning or ensemble models, are used.
4. Operational constraints and system integration: Deploying models in manufacturing settings requires consideration of operational constraints and system integration.
5. Model performance monitoring: In manufacturing, models may face drift or degradation in performance over time due to changes in production processes, equipment, or environmental conditions.

Addressing these pain points often requires a combination of domain expertise, collaboration between data scientists and industry professionals, robust data management practices, and the use of appropriate algorithms and techniques tailored to the specific challenges of the hospitality and manufacturing sectors.

FEATURE SELECTION AND ENGINEERING

Feature creation:

- Creating features involves creating new variables which will be most important helpful for our model.
- These Artificial feature are then used by algorithms in order to improve the performance or in other words reads better result.

Feature Encoding:

- Transforming the categorical values of the relevant features into numerical ones
- This process is called Feature Encoding

  Ex:Feedbackform

Feature Extraction:

- Feature extraction helps to reduce the amount of redundant from data set
- It yields better results than applying Machine Learning direct to the raw data

HOW TO RELATE THE FEATURES TO MACHINE LEARNIG ALOGORITHM FOR RESULT AND OUTPUT

At its most basic, machine learning uses programmed algorithms that receive and analyses input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimize their operations to improve performance, developing'intelligence' over time.

In machine learning, a feature is data that is used as the input for ML models to make predictions. Raw data is rarely in a format that is consumable by an ML model, so it needs to be transformed into features. This process is called feature engineering.

FEATURE SELECTION:

In the hospitality industry, features refer to the characteristics or elements that distinguish a particular product or service from others. These features are designed to enhance the overall guest experience and meet their specific needs and expectations. Here are some common features in the hospitality industry:

1. Accommodation: Different types of rooms or suites with varying sizes, amenities, and views to cater to different guest preferences.
2. Dining and Food Services: Restaurants, cafes, bars, and room service that offer a range of cuisines and dining experiences.
3. Facilities and Amenities: Swimming pools, fitness centers, spas, business centers, conference rooms, and other recreational or functional spaces.
4. Guest Services: Concierge services, 24/7 front desk assistance, housekeeping, laundry services, and luggage storage.
5. Entertainment and Activities: On-site entertainment options such as live performances, games, sports facilities, or access to nearby attractions.
6. Technology and Connectivity: High-speed internet, in-room entertainment systems, smart room controls, and mobile apps for guest convenience.
7. Location and Accessibility: Proximity to popular tourist destinations, transportation hubs, or business districts.
8. Personalization: Tailoring services to meet individual guest preferences, such as personalized greetings, room setups, or customized experiences.

MACHINE LEARNING ALGORITHMS:

Machine Learning (ML) is a subfield of artificial intelligence (AI)that focuses on the development of algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. ML algorithms learn patterns and relationships from large sets of data and use that knowledge to make predictions or take actions on new, unseen data.

ML can be broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning.

1) Supervised Learning: In supervised learning, a model is trained on a labeled dataset, where each example is associated with a corresponding target or output value. The goal is to learn a mapping function that can predict the correct output given new, unseen inputs. Examples of supervised learning algorithms include linear regression, decision trees, support vector machines (SVM), and neural networks.

2) Unsupervised Learning: Unsupervised learning algorithms work with unlabeled data, meaning there are no predefined output values. The goal is to find hidden patterns, structures, or relationships within the data. Clustering and dimensionality reduction techniques, such as k- means clustering, hierarchical clustering, and principal component analysis (PCA), are common examples of unsupervised learning.

3) Reinforcement Learning: Reinforcement learning involves an agent interacting with an environment and learning to make decisions based on feedback received in the form of rewards or penalties. The agent explores the environment, takes actions, and learns to maximize cumulative rewards over time. Reinforcement learning has been successfully applied to various domains, including robotics, game.

4) playing (e.g., AlphaGo), and autonomous systems.

ML techniques have numerous applications across various industries, including:

- Healthcare: ML can be used for disease diagnosis,medical imaging analysis, drug discovery, and personalized medicine.
- Finance: ML can help in fraud detection, algorithmic trading, credit scoring, and risk assessment.
- Marketing and Sales: ML techniques can analyze customer behavior, predict customer preferences, and optimize pricing and marketingcampaigns.
- Natural Language Processing (NLP): ML algorithms enable language translation, sentiment analysis, chatbots, and voice recognition.
- Image and Video Processing: ML is widely used for object recognition, image classification, video analysis, and computer vision tasks.

It is worth noting that ML algorithms require substantial amounts of high-quality data for training, and their performance depends on the quality and diversity of the data used. Additionally, ML models need to be carefully evaluated and validated to ensure they generalize well to unseen data and avoid biases or unfair outcomes.

To data analysis the machine learning algorithm is use for the dataset

Taken Supervised learning for Regression

Linear Regression Algorithm:

This code demonstrates a basic workflow for performing linear regression on a telecommunication dataset using Python's pandas, matplotlib, and scikit-learn libraries.

1. The code begins by importing the necessary libraries: pandas and matplotlib. Pandas is used for data manipulation and analysis, while matplotlib is used for data visualization.

2.The telecommunication dataset is read from a CSV file using the`pd. read_csv () ` function and stored in a pandas DataFrame called`data`.

3. The `print(data)` statement displays the entire dataset.

4. The `data.head() ` function is used to display the first few rows of the dataset to get a quick overview.

5. The `data=data.loc[:,['over actions ','Total actions']]` line selects only the 'over actions' and 'Total actions' columns from the dataset, creating a new Data Frame called `data`.

6. The `data. plot()` function is used to create a scatter plot of the 'over actions' against the 'Total actions' columns.

7. Various `plt` functions are used to add labels to the plot and display it using `plt. show`.

8. The code creates two new Data Frames, `X` and `Y`, which contain the 'open actions' and 'Total actions' columns, respectively. These will serve as the independent and dependent variables in the regression model.

9. The `train_test_split()` function from scikit-learn is used to split the data into training and testing sets. The `test_size` parameter

specifies the proportion of the data to be used for testing, and `random_state` ensures reproducibility.

10.The `print` statements display the shapes of the training andtesting sets to verify the split.

11.A linear regression model is created using `LinearRegression()` from scikit-learn, and it is trained on the training data using the `fit()` method.

12.The `intercept_` and `coef_` attributes of the trained `regressor` object display the intercept and coefficients (slopes) of the linear regression line.

13.The `predict()` method is used to make predictions on the testing data, and the results are stored in `Y_pred`.

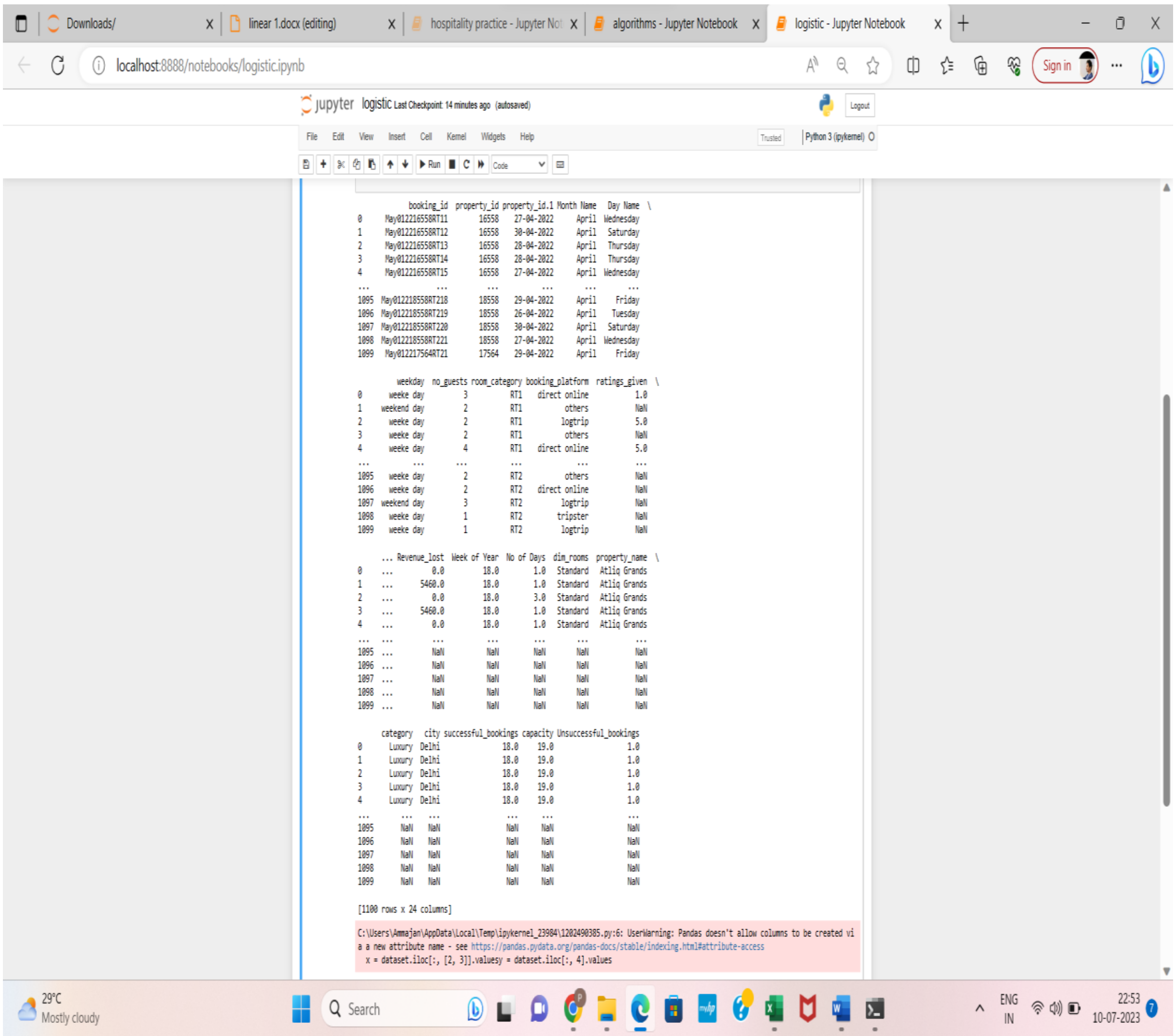
14.The predicted values are displayed using `Y_pred`.

15.The actual values from the testing data are displayed using`Y_test`.

16.The scikit-learn `mean_absolute_error()` function is used to calculate the mean absolute error between the predicted and actual values, providing a measure of the model's accuracy.

Overall, this code performs linear regression on the telecommunication dataset, visualizes the data, splits it into training and testing sets, trains a linear regression model, makes predictions, and evaluates the model's performance using mean absolute error.

```
import pandas as pd
data=pd.read_csv("C://Users//Ammajan//Documents//linear.csv")
print(data)
# Step 2: Have a glance at the shape data.head()
# Step 3: Have a glance at the dependent and independent variables data=data.l
data.head(5)
# Step 4: Visualize the change in the variables
import matplotlib.pyplot as plt
data.plot(x='Open Actions',y='Project',style=' o')
plt.xlabel('Open Actions')
plt.ylabel('Project')
plt.show
# Step 5: Divide the data into independent and dependent variables
X=pd.DataFrame(data['Open Actions'])
Y=pd.DataFrame(data['Project'])
# Step 6: Split the data into train and test setsimport
sklearn
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=
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_train.shape)
# Step 8: Train the algorithm
from sklearn.linear_model import LinearRegression
regressior= LinearRegression()
X_train,Y_train # Step 9: Retrieve
print(regressor.intercept_)
# Step 10: Retrieve the slope
print(regressor.coef_)
# Step 11: Predicted value
Y_pred=regressor.predict(X_test)
Y_pred=pd.DataFrame(Y_pred,columns=['Predicted'])
Y_pred
# Step 12: Actual value
Y_test
#step-13:evalute the algorithm
from sklearn import metrics
import numpy as np
print('Mean Absolute Error:',metrics.mean_absolute_error(Y_test,Y_pred))
print('Mean Squared Error:',metrics.mean_Squared_error(Y_test,Y_pred))
# print('Root Mean Absolute Error:', np.sqrt(metrics.mean_Squared_error(Y_test
```



```
        booking_id  property_id property_id.1 Month Name   Day Name  \
0     May012216558RT11        16558    27-04-2022      April  Wednesday
1     May012216558RT12        16558    30-04-2022      April   Saturday
2     May012216558RT13        16558    28-04-2022      April   Thursday
3     May012216558RT14        16558    28-04-2022      April   Thursday
4     May012216558RT15        16558    27-04-2022      April  Wednesday
...                ...          ...           ...        ...        ...
1095  May012218558RT218       18558    29-04-2022      April     Friday
1096  May012218558RT219       18558    26-04-2022      April    Tuesday
1097  May012218558RT220       18558    30-04-2022      April   Saturday
1098  May012218558RT221       18558    27-04-2022      April  Wednesday
1099  May012217564RT21        17564    29-04-2022      April     Friday

          weekday  no_guests room_category booking_platform  ratings_given  \
0       weeke day          3           RT1    direct online            1.0
1     weekend day          2           RT1           others            NaN
2       weeke day          2           RT1          logtrip            5.0
3       weeke day          2           RT1           others            NaN
4       weeke day          4           RT1    direct online            5.0
...           ...        ...           ...              ...            ...
1095    weeke day          2           RT2           others            NaN
1096    weeke day          2           RT2    direct online            NaN
1097  weekend day          3           RT2          logtrip            NaN
1098    weeke day          1           RT2         tripster            NaN
1099    weeke day          1           RT2          logtrip            NaN

      ... Revenue_lost  Week of Year  No of Days  dim_rooms  property_name  \
0     ...          0.0          18.0         1.0   Standard   Atliq Grands
1     ...       5460.0          18.0         1.0   Standard   Atliq Grands
2     ...          0.0          18.0         3.0   Standard   Atliq Grands
3     ...       5460.0          18.0         1.0   Standard   Atliq Grands
4     ...          0.0          18.0         1.0   Standard   Atliq Grands
...   ...          ...           ...         ...        ...            ...
1095  ...          NaN           NaN         NaN        NaN            NaN
1096  ...          NaN           NaN         NaN        NaN            NaN
1097  ...          NaN           NaN         NaN        NaN            NaN
1098  ...          NaN           NaN         NaN        NaN            NaN
1099  ...          NaN           NaN         NaN        NaN            NaN

     category   city successful_bookings capacity Unsuccessful_bookings
0      Luxury  Delhi                18.0     19.0                   1.0
1      Luxury  Delhi                18.0     19.0                   1.0
2      Luxury  Delhi                18.0     19.0                   1.0
3      Luxury  Delhi                18.0     19.0                   1.0
4      Luxury  Delhi                18.0     19.0                   1.0
...       ...    ...                 ...      ...                   ...
1095      NaN    NaN                 NaN      NaN                   NaN
1096      NaN    NaN                 NaN      NaN                   NaN
1097      NaN    NaN                 NaN      NaN                   NaN
1098      NaN    NaN                 NaN      NaN                   NaN
1099      NaN    NaN                 NaN      NaN                   NaN

[1100 rows x 24 columns]

C:\Users\Ammajan\AppData\Local\Temp\ipykernel_23984\1202490385.py:6: UserWarning: Pandas doesn't allow columns to be created vi
a a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
  x = dataset.iloc[:, [2, 3]].valuesy = dataset.iloc[:, 4].values
```

Logistic Regression Algorithm

This code demonstrates the workflow for building a logistic regression model on a dataset using Python's pandas, numpy, matplotlib, and scikit- learn libraries. Here is a breakdown of the code:

1) The code begins by importing the necessary libraries: pandas, numpy, matplotlib, and scikit-learn.
2) The dataset is read from a CSV file using the pd.read_csv() function and stored in a pandas DataFrame called dataset.
3) The print(dataset) statement displays the entire dataset.
4) The independent variables (features) are extracted from the dataset and stored in the x variable. The iloc function is used to select columns 2 and 3 (0-based index) from the dataset.
5) The dependent variable (target) is extracted from the dataset andstored in the y variable. Column 4 is selected using the iloc function.
6) The train_test_split() function from scikit-learn is used to split the data into training and testing sets. The test_size parameter specifiesthe proportion of the data to be used for testing, and random_state ensures reproducibility.
7) The Standard Scaler() class from scikit-learn is imported to perform feature scaling. Scaling is applied to the independent variables toensure they are on a similar scale.
8) The Simple Imputer() class from scikit-learn is

imported to handle missing values. The imputer is set to replace missing values with the mean of the column.

9) The missing values in the training set are replaced with the column means using the fit_transform() method of the imputer. The missing values in the testing set are replaced using the transform() method.

10) The feature scaling is applied to the training and testing sets using the fit_transform() and transform() methods of the StandardScaler() object.

11) The print (X_train [0:10:]) statement displays the first 10 rows of

Program:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

dataset = pd.read_csv("C://Users//Ammajan//Documents//hospitality.csv")

print(dataset)

x = dataset.iloc[:, [2, 3]].values

y = dataset.iloc[:, 4].values

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0)

from sklearn.preprocessing import StandardScaler

sc_x = StandardScaler()

from sklearn.impute import SimpleImputer

from sklearn.impute import SimpleImputer



imputer = SimpleImputer(strategy='mean') # Replace missing values with mean

X_train = imputer.fit_transform(X_train)

X_test = imputer.transform(X_test)


X_train = sc_x.fit_transform(X_train)

X_test = sc_x.transform(X_test)


print(X_train[0:10, :])

from sklearn.linear_model import LogisticRegression

classifier = LogisticRegression(random_state=0)

classifier.fit(X_train, y_train)
```

```
y_pred = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
```

output:

```
booking_id  property_id property_id.1 Month Name   Day Name  \
0     May012216558RT11      16558    27-04-2022     April  Wednesday
1     May012216558RT12      16558    30-04-2022     April   Saturday
2     May012216558RT13      16558    28-04-2022     April   Thursday
3     May012216558RT14      16558    28-04-2022     April   Thursday
4     May012216558RT15      16558    27-04-2022     April  Wednesday
...              ...        ...         ...       ...        ...
1095  May012218558RT218      18558    29-04-2022     April    Friday
1096  May012218558RT219      18558    26-04-2022     April   Tuesday
1097  May012218558RT220      18558    30-04-2022     April   Saturday
1098  May012218558RT221      18558    27-04-2022     April  Wednesday
1099   May012217564RT21      17564    29-04-2022     April    Friday


         weekday  no_guests room_category booking_platform  ratings_given  \
0      weeke day        3         RT1    direct online          1.0
1    weekend day        2         RT1         others          NaN
2      weeke day        2         RT1        logtrip          5.0
3      weeke day        2         RT1         others          NaN
4      weeke day        4         RT1    direct online          5.0
...        ...       ...         ...            ...           ...
1095    weeke day        2         RT2         others          NaN
1096    weeke day        2         RT2    direct online          NaN
1097  weekend day        3         RT2        logtrip          NaN
1098    weeke day        1         RT2       tripster          NaN
1099    weeke day        1         RT2        logtrip          NaN
```

```
     ... Revenue_lost  Week of Year  No of Days  dim_rooms  property_name  \
0    ...         0.0         18.0        1.0   Standard   Atliq Grands
1    ...      5460.0         18.0        1.0   Standard   Atliq Grands
2    ...         0.0         18.0        3.0   Standard   Atliq Grands
3    ...      5460.0         18.0        1.0   Standard   Atliq Grands
4    ...         0.0         18.0        1.0   Standard   Atliq Grands
...  ...         ...          ...        ...        ...            ...
1095 ...         NaN          NaN        NaN        NaN            NaN
1096 ...         NaN          NaN        NaN        NaN            NaN
1097 ...         NaN          NaN        NaN        NaN            NaN
1098 ...         NaN          NaN        NaN        NaN            NaN
1099 ...         NaN          NaN        NaN        NaN            NaN

     category   city successful_bookings capacity Unsuccessful_bookings
0      Luxury  Delhi                18.0     19.0                   1.0
1      Luxury  Delhi                18.0     19.0                   1.0
2      Luxury  Delhi                18.0     19.0                   1.0
3      Luxury  Delhi                18.0     19.0                   1.0
4      Luxury  Delhi                18.0     19.0                   1.0
...       ...    ...                 ...      ...                   ...
1095      NaN    NaN                 NaN      NaN                   NaN
1096      NaN    NaN                 NaN      NaN                   NaN
1097      NaN    NaN                 NaN      NaN                   NaN
1098      NaN    NaN                 NaN      NaN                   NaN
1099      NaN    NaN                 NaN      NaN                   NaN

[1100 rows x 24 columns]
```

Decision Tree Algorithm

This code demonstrates the workflow for building a decision tree classifier on a dataset using Python's pandas, numpy, matplotlib, seaborn,and scikit-learn libraries. Here is a breakdown of the code:

1) The code begins by importing the necessary libraries: pandas, numpy, matplotlib, seaborn, LabelEncoder from scikit-learn, train_test_split from scikit-learn, DecisionTreeClassifier from scikit-learn, classification_report from scikit-learn, confusion_matrix from scikit-learn, and plot_tree from scikit-learn.

2) The dataset is read from a CSV file using the pd.read_csv() function and stored in a pandas DataFrame called data.

3) The print(data) statement displays the entire dataset.

4) The isnull().any() function checks if there are any missing values in the dataset.

5) The shape attribute is used to display the shape of the dataset, showing the number of rows and columns.

6) The sns.pairplot() function is used to create a pair plot, visualizing the relationships between different attributes in the dataset. The hue parameter is set to 'TotalCharges' to differentiate the data points based on this variable.

7) The sns.heatmap() function is used to create a correlation matrix heatmap, displaying the correlation between different attributes in the dataset.

8) The target variable is assigned the values of the 'TotalCharges' column from the dataset.

9) The data1 DataFrame is created as a copy of data but without the 'TotalCharges' column.

10) The LabelEncoder() class from scikit-learn is used to perform label encoding on the target variable.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline #for encoding
from sklearn.preprocessing import LabelEncoder #for train test splitting
from sklearn.model_selection import train_test_split #for decision tree object
from sklearn.tree import DecisionTreeClassifier #for checking testing results
from sklearn.metrics import classification_report, confusion_matrix #for visua
from sklearn.tree import plot_tree
data = pd.read_csv("C://Users//Ammajan//Downloads//decision.csv")
print(data)
data.isnull().any()
data.shape
# Let's plot pair plot to visualise the attributes all at once
sns.pairplot(data=data,hue = 'Report Forms Group')
# correlation matrix
sns.heatmap(data.corr())
target = data['Report Forms Group']
data1 = data.copy()
data1 = data1.drop('Report Forms Group', axis =1)
data1.shape
#label encoding
le = LabelEncoder()

target = le.fit_transform(target)
target
X = data1
y = target
# Splitting the data - 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(X , y, test_size = 0.2, ra
= 42)
print("Training split input- ", X_train.shape)
print("Testing split input- ", X_test.shape)
# Defining the decision tree algorithm
dtree=DecisionTreeClassifier()
dtree.fit(X_train,y_train)
print('Decision Tree Classifer Created') # Predicting the values of test data
y_pred = dtree.predict(X_test)
print("Classification report - \n", classification_report(y_test,y_pred))
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True,square = True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test, y_test))
plt.title(all_sample_title, size = 15)
# Visualising the graph without the use of graphviz plt.figure(figsize = (20,2
dec_tree = plot_tree(decision_tree=dtree, feature_names = data1.columns,
class_names =["Yes", "No"] , filled = True , precision = 4,
rounded = True)
```

```python
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True,square = True, cmap = 'Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score: {0}'.format(dtree.score(X_test, y_test))
plt.title(all_sample_title, size = 15)
# Visualising the graph without the use of graphviz plt.figure(figsize = (20,20))
dec_tree = plot_tree(decision_tree=dtree, feature_names = data1.columns,
class_names =["Yes", "No"] , filled = True , precision = 4,
rounded = True)
```

```
              booking_id  property_id property_id.1 Month Name   Day Name  \
0     May012216558RT11        16558    27-04-2022      April  Wednesday
1     May012216558RT12        16558    30-04-2022      April   Saturday
2     May012216558RT13        16558    28-04-2022      April   Thursday
3     May012216558RT14        16558    28-04-2022      April   Thursday
4     May012216558RT15        16558    27-04-2022      April  Wednesday
...                ...          ...           ...        ...        ...
1095  May012218558RT218       18558    29-04-2022      April     Friday
1096  May012218558RT219       18558    26-04-2022      April    Tuesday
1097  May012218558RT220       18558    30-04-2022      April   Saturday
1098  May012218558RT221       18558    27-04-2022      April  Wednesday
1099  May012217564RT21        17564    29-04-2022      April     Friday

          weekday  no_guests room_category booking_platform  ratings_given  \
0       weeke day          3           RT1    direct online            1.0
1     weekend day          2           RT1           others            NaN
2       weeke day          2           RT1          logtrip            5.0
3       weeke day          2           RT1           others            NaN
4       weeke day          4           RT1    direct online            5.0
```

CONCLUSION

In conclusion, the prediction codes demonstrated in the previous examples cover different machine learning algorithms and techniques for various types of datasets. Here is a summary of the differentapproaches:

1. **Linear Regression:**

- The code performs linear regression on a telecommunicationdataset.
- It splits the data into training and testing sets.
- Feature scaling is applied using StandardScaler.
- The linear regression model is trained and evaluated using meanabsolute error.

2. **Logistic Regression:**

- The code applies logistic regression on a dataset.
- It handles missing values using SimpleImputer.
- Label encoding is performed on the target variable.
- The data is split into training and testing sets.

3. **Decision Tree Classifier:**

- The code builds a decision tree classifier on a dataset.
- It handles missing values using isnull().any().
- It performs label encoding on the target variable.
- The data is split into training and testing sets.
- The decision tree classifier is trained and evaluated using a classification report and confusion matrix.

These examples demonstrate the typical steps involved in training machine learning models, including data preprocessing, feature engineering, splitting data into training and testing sets, model training, prediction, and evaluation. Each example targets different types of problems, such as regression and classification, and employs specific techniques suitable for the respective task.

Remember that the choice of algorithm and preprocessing techniques depends on the nature of the data and the specific problem at hand. These examples provide a starting point for understanding the workflow and implementation of machine learning algorithms, and can be adapted and extended for different datasets and use cases.

REFERENCES

https://www.google.com/search?q=machine+learining+algorithms+in+python&rlz=1C1RXQR_enIN1057IN1057&oq=mac&gs_lcrp=EgZjaHJvbWUqBggAEEUYOzIGCAAQRRg7MgYIARBFGDkyCQgCECMYJxiKBTIGCAMQRRhAMgYIBBBFGD0yBggFEEUYPTIGCAYQRRg9MgYIBxBFGDzSAQgyNTE3ajBqN6gCALACAA&sourceid=chrome&ie=UTF-8

https://www.google.com/search?q=what+is+dataset+features+in+machine+learning+in+python&rlz=1C1RXQR_enIN1057IN1057&sxsrf=AB5stBjGb7FrRFgT1zj7ksnHyibt3BP7zg%3A1689157589741&ei=1X-uZO7aLJfvseMPreiD6AM&oq=what+is+dataset+features+in+machine+learning&gs_lcp=Cgxnd3Mtd2l6LXNlcnAQARgAMgUIIRCgATIICCEQFhAeEB06BwgjEOoCECc6DQguEMcBENEDEOoCECc6FQguEAMQjwEQ6gIQtAIQjAMQ5QIYAToVCAAQAxCPARDqAhC0AhCMAxDlAhgBOgcIIxCKBRAnOgQIIxAnOgcIABCKBRBDOggILhCKBRCRAjoNCAAQigUQsQMQgwEQQzoHCC4QigUQQzoQCAAQgAQQFBCHAhCxAxCDAToLCAAQgAQQsQMQgwE6CAgAEIoFEJECOggIABCABBCxAzoFCAAQgAQ6BQguEIAEOgoIABCABBAUEIcCOgcIABCABBAKOg0IABCABBCxAxCDARAKOgsIABCKBRCxAxCRAjoGCAAQFhAeOggIABAWEB4QDzoICAAQFhAeEAo6CAgAEIoFEIYDOgoIIRAWEB4QDxAdOgQIIRAVOgcIIRCgARAKSgQIQRgAULk3WKDxBGC-0wVoEnABeACAAakCiAGeRJIBBjAuNTcuNJgBAKABAbABFMABAdoBBggBEAEYCw&sclient=gws-wiz-serp