



**NAME : PATHAN AMMAJAN**

**ROLL NO. : 203H1A0537**

## COURSE : DATA SCIENCE

## INTERN ID: BOV23FD0093

INDUSTRY : HOSPITALITY

DEPARTMENT: MANUFACTURING MAIN POINTS

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTEXT** | **PAGE.NO** |
| 1 | Abstract |  |
| 2 | Introduction |  |
| 3 | Problem statement and problem solution |  |
| 4 |  |  |
| 5 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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.

Top of Form

**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
3. Travel and Tourism: This aspect involves travel agencies, tour operators, transportation services (such as airlines, cruises, trains, and car rentals), and other businesses that facilitate travel and tourism experiences. It includes providing information, booking accommodations, arranging transportation, and organizing sightseeing activities.

There are some more key components like,

Event Planning and Management, 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**

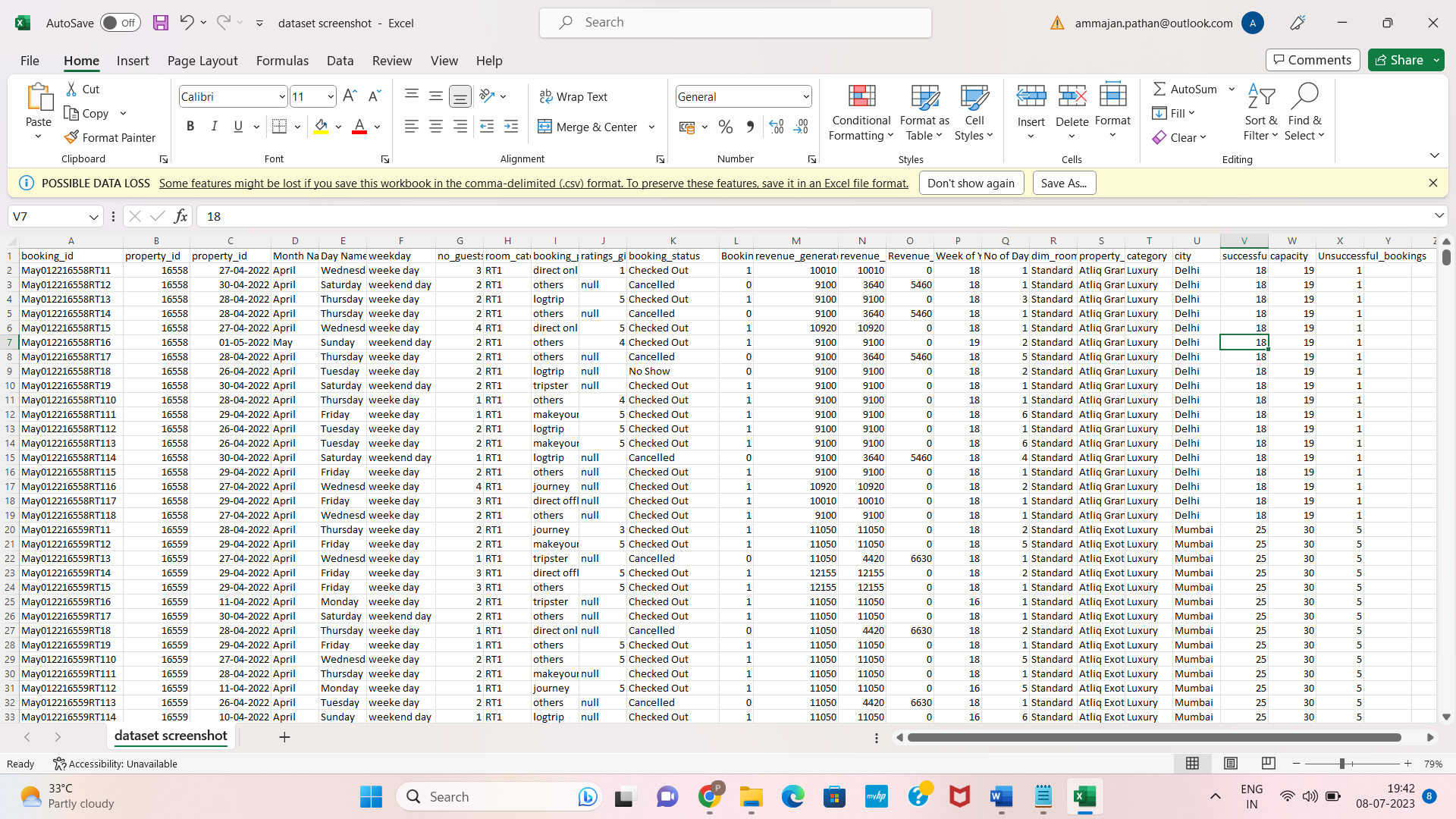
In the context of hospitality in manufacturing, there can be several issues and pain points that establishments may face. Here are some common challenges:

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. Compliance and Regulatory Requirements: Hospitality manufacturers must comply with various regulations and standards, such as food safety regulations, health and safety guidelines, labeling requirements, and environmental regulations.
4. 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.
5. Waste Management: Hospitality manufacturing often generates significant amounts of waste, including food waste, packaging materials, and by-products. Implementing effective waste management strategies, such as recycling, composting, or reducing waste generation, can be challenging and requires careful planning and coordination.
6. Scalability and Flexibility: Hospitality manufacturers need to adapt to changing demands, seasonal variations, and market trends. Scaling up or down production levels to meet fluctuating demands and maintaining flexibility in manufacturing processes can be a challenge.
7. 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**

Consider the Data set taken the Combination of Numerical and Categorical Data Machine Learning Algorithm- To perform Supervised Learning By the Feature Engineer all concept.



**Pre-Processing Data Top of Form**

**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:

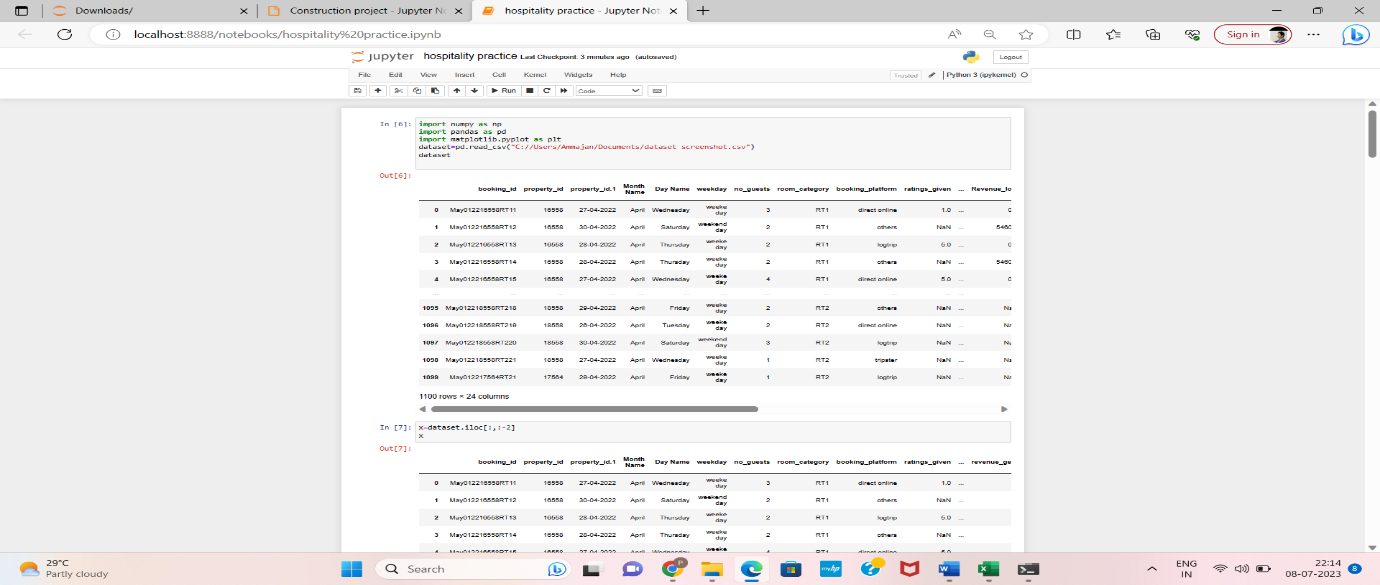
1. Data Cleaning: This step involves identifying and handling missing, incorrect, or inconsistent data. It may include tasks such as removing duplicates, filling in missing values, and correcting any errors or outliers in the dataset.
2. Data Integration: In hospitality manufacturing, data often comes 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. **Top of Form**

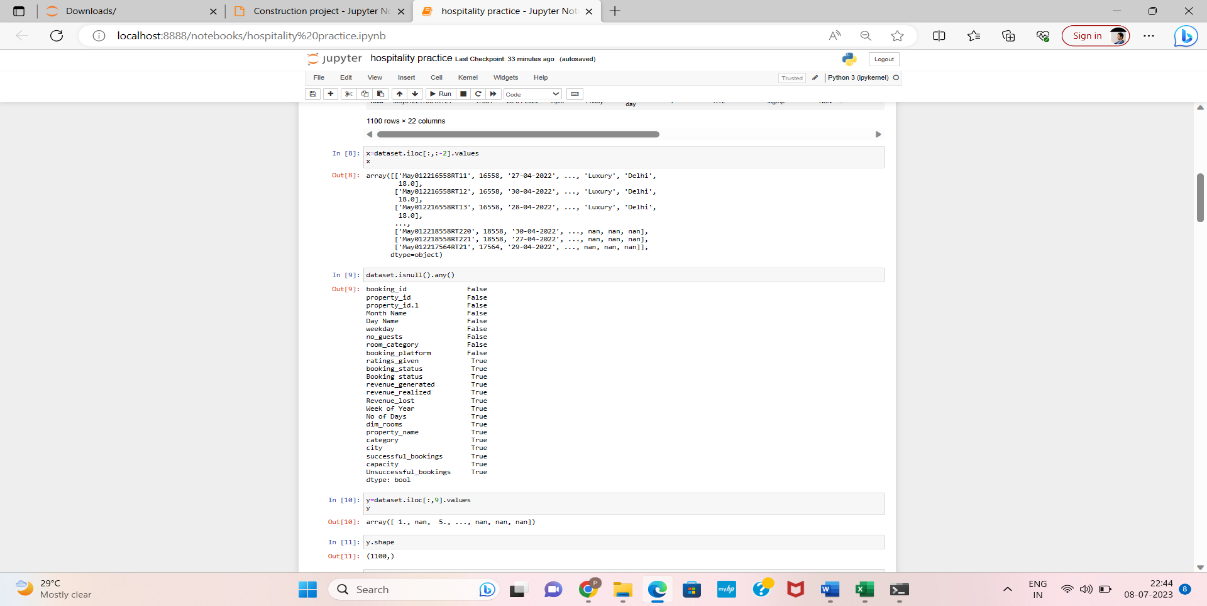
**Construction data analysis**

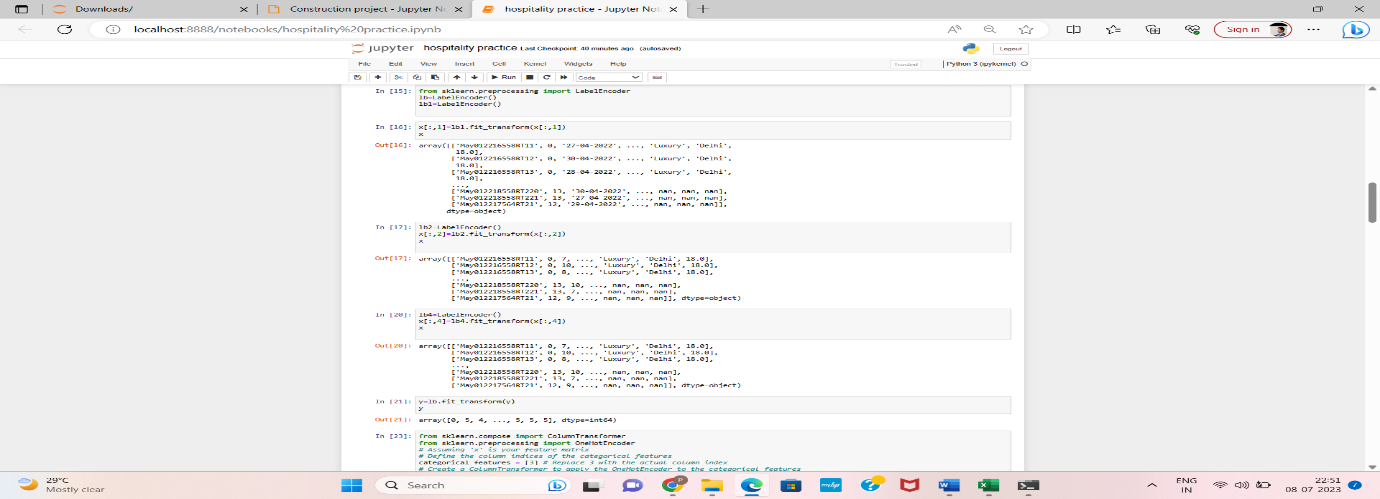
**CONTENTS**

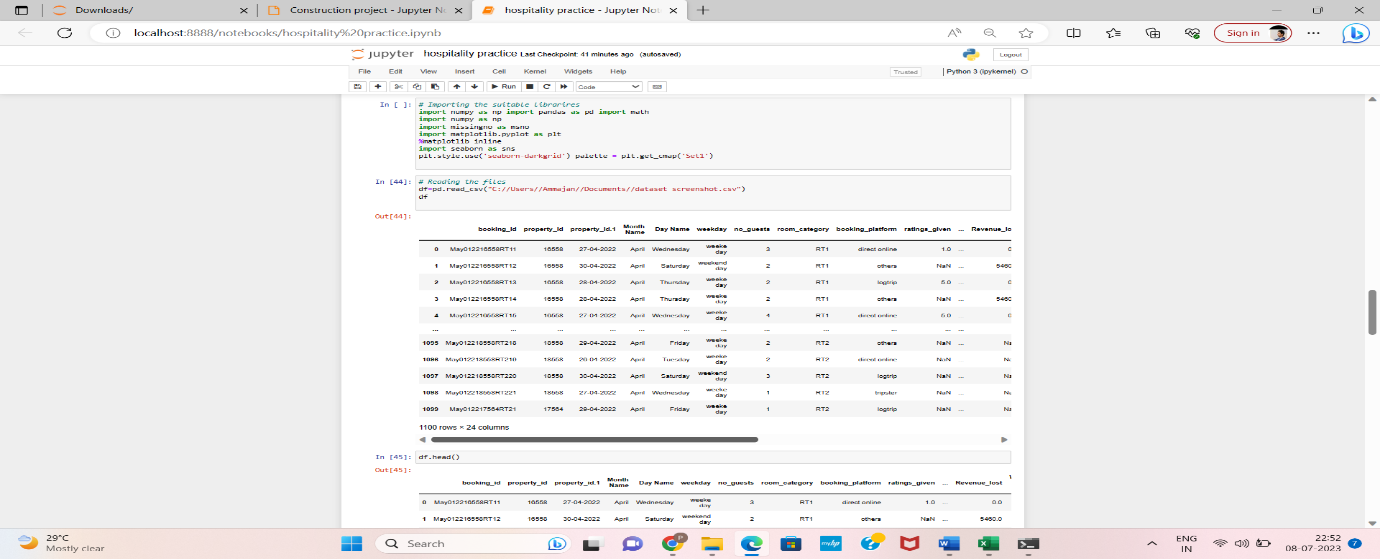
* Importing
* the suitable libraries
* Reading the files
* Cleaning the data
* Visualisation of the data
* Conclusion made from the data

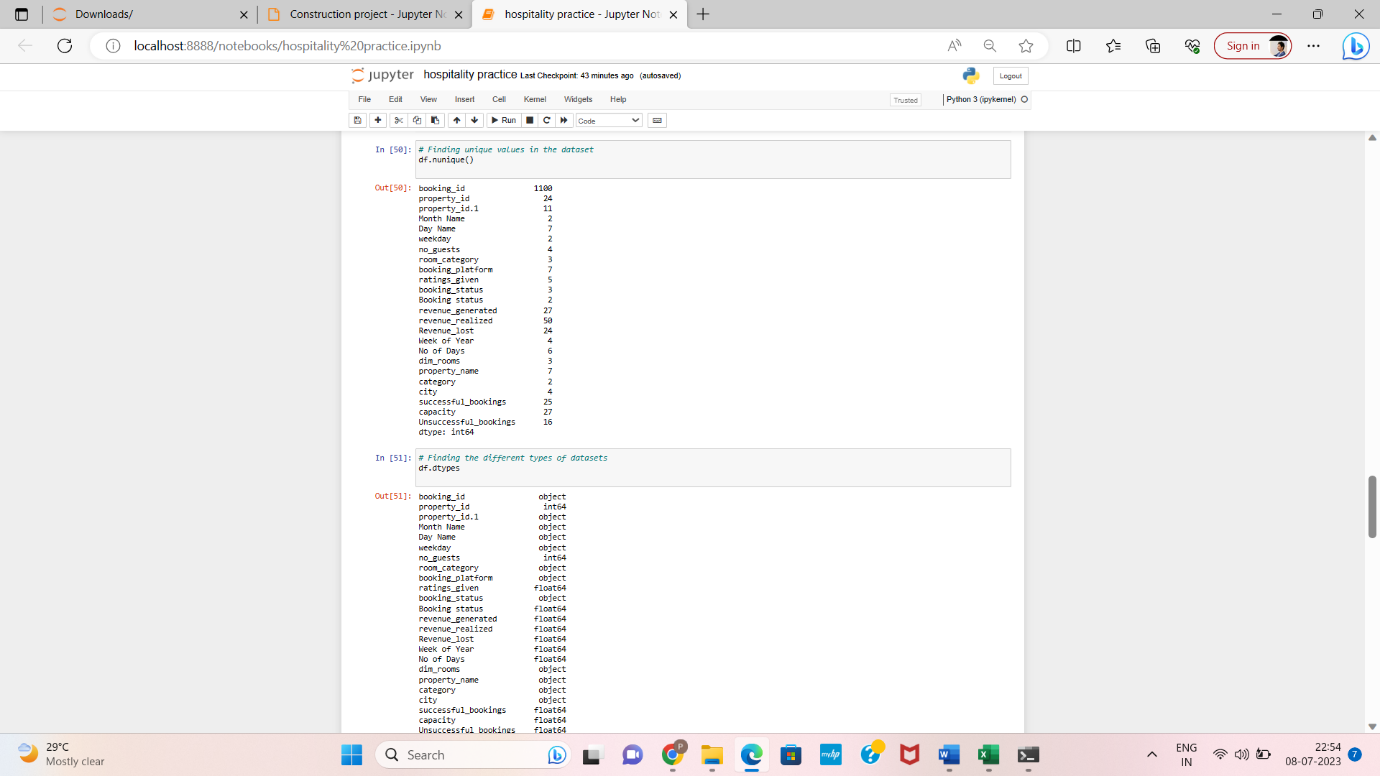
**CODE:**

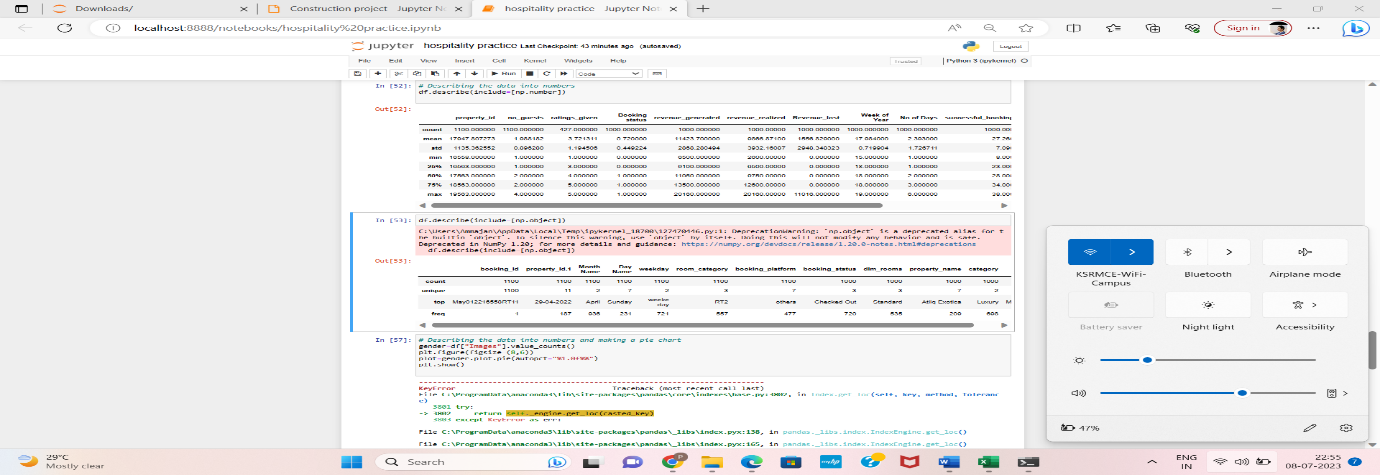
****

****

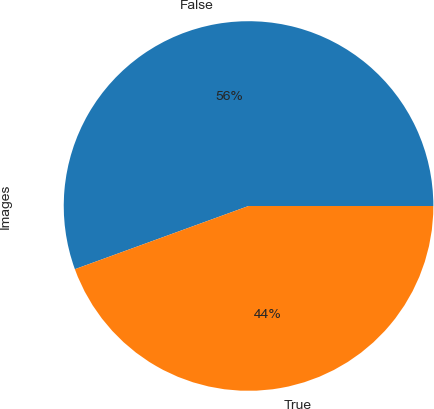
****

****

****

****





**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:

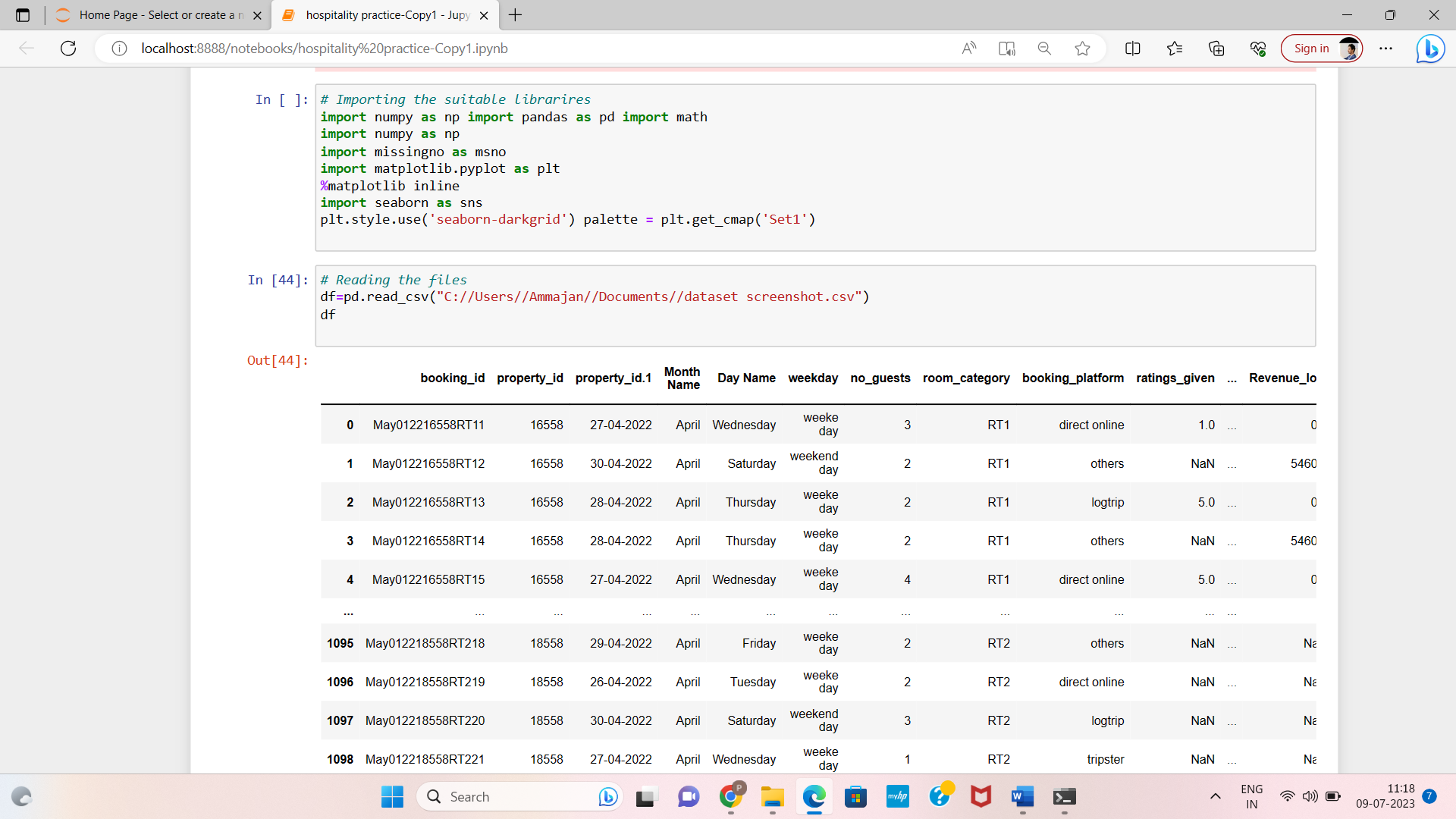
1. Demand forecasting: By analyzing historical data on room bookings, seasonal trends, local events, and other factors, hotels can predict future demand for rooms. This helps them optimize their pricing strategies, allocate resources efficiently, and make informed decisions about inventory management.
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:

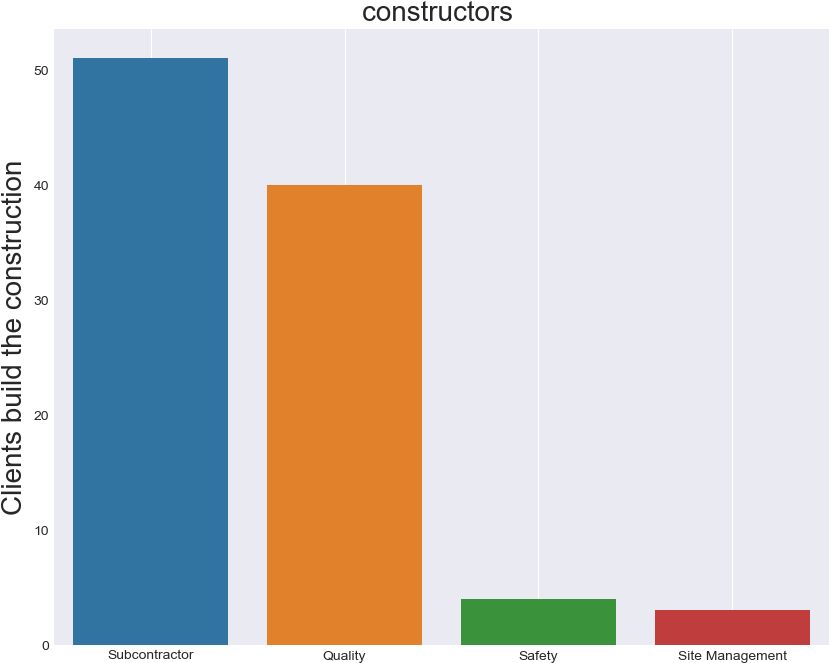
1. Supply chain management: Manufacturers often struggle with managing complex supply chains involving multiple suppliers, transportation, and inventory management. Issues such as delays, disruptions, and lack of visibility can impact production schedules and lead to inefficiencies.
2. Production planning 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:**

****





**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.

Top of Form

**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:**



**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 marketing campaigns.
* 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 and testing 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.

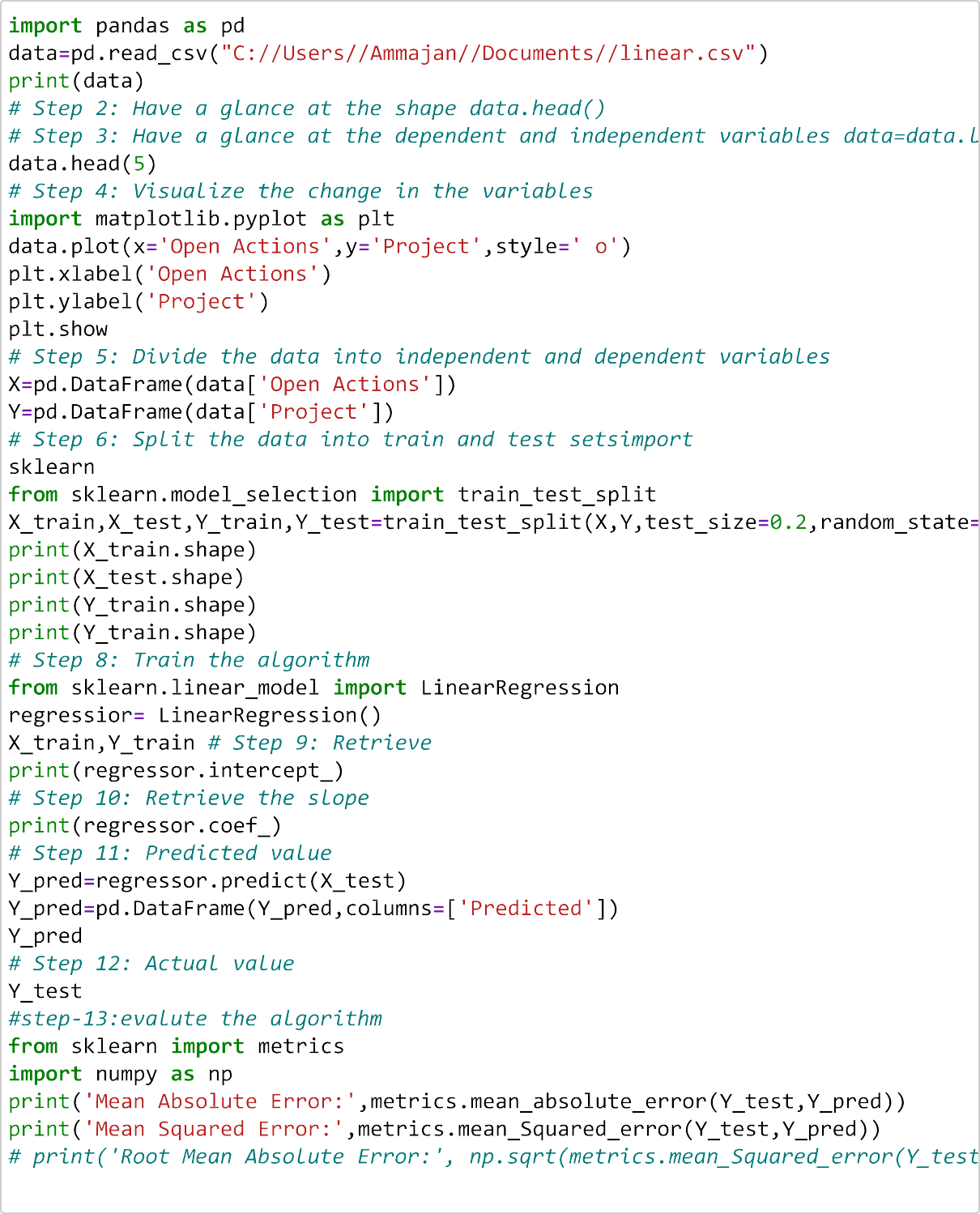
13.The `predict()` method is used to make predictions on the testing data, and the results are stored in `Y\_pred`.

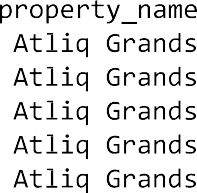
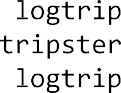
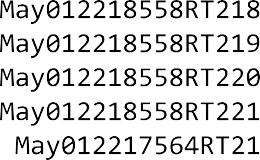
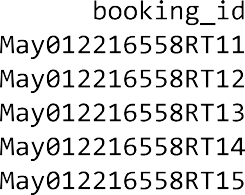
14.The predicted values are displayed using `Y\_pred`.

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.































**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 and stored 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 specifies the 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 to ensure 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

the scaled training set.

1. The LogisticRegression() class from scikit-learn is imported to create a logistic regression model. The random\_state parameter ensures reproducibility.
2. The logistic regression model is trained on the scaled training data using the fit() method.
3. Predictions are made on the testing set using the predict() method, and the results are stored in the y\_pred variable.
4. The confusion\_matrix() function from scikit-learn is used to calculate the confusion matrix, which provides information about the model's performance in terms of true positives, true negatives, false positives, and false negatives. The confusion matrix is stored in the cm variable.

Overall, this code performs logistic regression on the dataset, splits it into training and testing sets, handles missing values, applies feature scaling, trains the logistic regression model, makes predictions, and evaluates the model's performance using the confusion matrix.

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)

​

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.
  11. The independent variables (features) are assigned to the X variable, and the target variable is assigned to the y variable.
  12. The data is split into training and testing sets using the train\_test\_split() function from scikit-learn. The test\_size parameter specifies the proportion of the data to be used for testing, and random\_state ensures reproducibility.
  13. The decision tree classifier is created using the DecisionTreeClassifier() class from scikit-learn, and it is trained on the training data using the fit() method.

Predictions are made on the testing set using the predict() method, and the results are stored in the y\_pred variable.

* 1. The classification report is printed using the classification\_report() function from scikit-learn, which provides precision, recall, F1-score, and support for each class.
  2. The confusion matrix is calculated using the confusion\_matrix() function from scikit-learn, and it is displayed as a heatmap using the sns.heatmap() function.
  3. Finally, the decision tree is visualized using the plot\_tree() function from scikit-learn, which displays the decision tree graphically.

Overall, this code performs decision tree classification on the dataset, handles missing values, performs label encoding, splits the data into training and testing sets, trains the decision tree classifier, makes predictions, evaluates the model's performance using the classification report and confusion matrix, and visualizes the decision tree.





