Hotel Chain Data Analysis and Performance Optimization Project

"Al-Ofoq Al-Lami'" (The Shining Horizon) Hotel Chain

Prepared by: Rinad Almjishai

Date: August 10, 2024

Introduction

"Al-Ofoq Al-Lami'" (The Shining Horizon) is a prestigious hotel chain with multiple properties spread across various locations. In an effort to enhance its operational efficiency and market position, the chain has initiated a comprehensive data analysis project. This project aims to analyze performance metrics across all hotels in the chain, refine marketing strategies, and increase overall revenue.

The primary objectives of this analysis are to:

1. Evaluate hotel performance within the chain
2. Identify areas for improvement in operational efficiency
3. Develop targeted marketing strategies based on data insights
4. Increase overall revenue and profitability for the entire chain

This report presents the methodology, findings, and recommendations derived from our extensive data analysis. The insights gathered will serve as a foundation for strategic decision-making, aimed at optimizing the performance of the chain.

Table of Contents

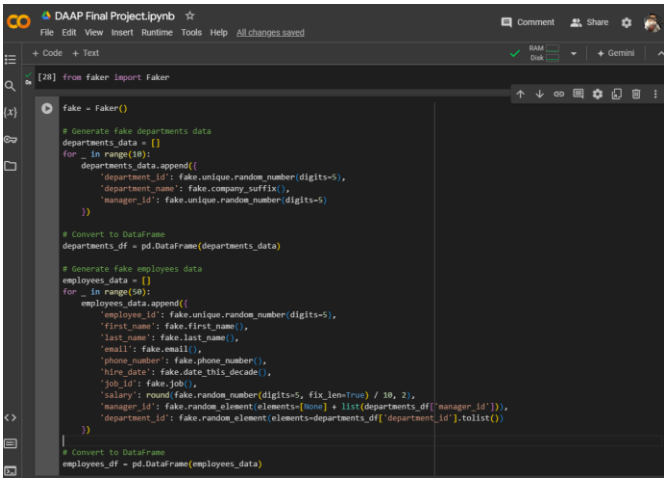
1. Executive Summary
2. Project Overview
    2.1 Background
    2.2 Objectives
    2.3 Scope
3. Data Collection and Preparation
    3.1 Data Sources
    3.2 Data Cleaning and Preprocessing
    3.3 Data Integration
4. Methodology
    4.1 Tools and Technologies Used
    4.2 Data Analysis Techniques
    4.3 Sentiment Analysis Model
5. Data Visualization and Analysis
    5.1 Key Performance Indicators
    5.2 Customer Segmentation
    5.3 Booking Patterns and Trends
    5.4 Room Type Analysis
    5.5 Cancellation Analysis
6. Findings and Insights
7. Extra- Sentiment analysis model

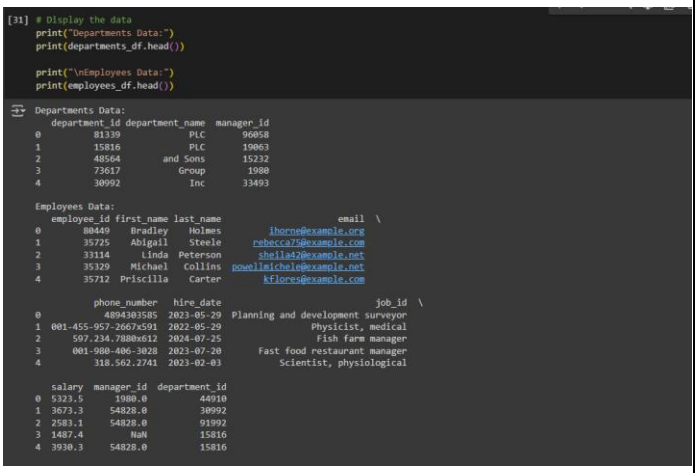
1- Data cleaning, collecting and preparation:

Tools: Google Collab, MySQL Server

In this phase of the Hotel Chain Project, we focused on preparing and cleaning the data to ensure its quality and consistency for analysis. We performed the following steps:

1. Sample Data Creation (Python Faker library):
    - We created sample data for the Employees table to complement the existing booking and review data.

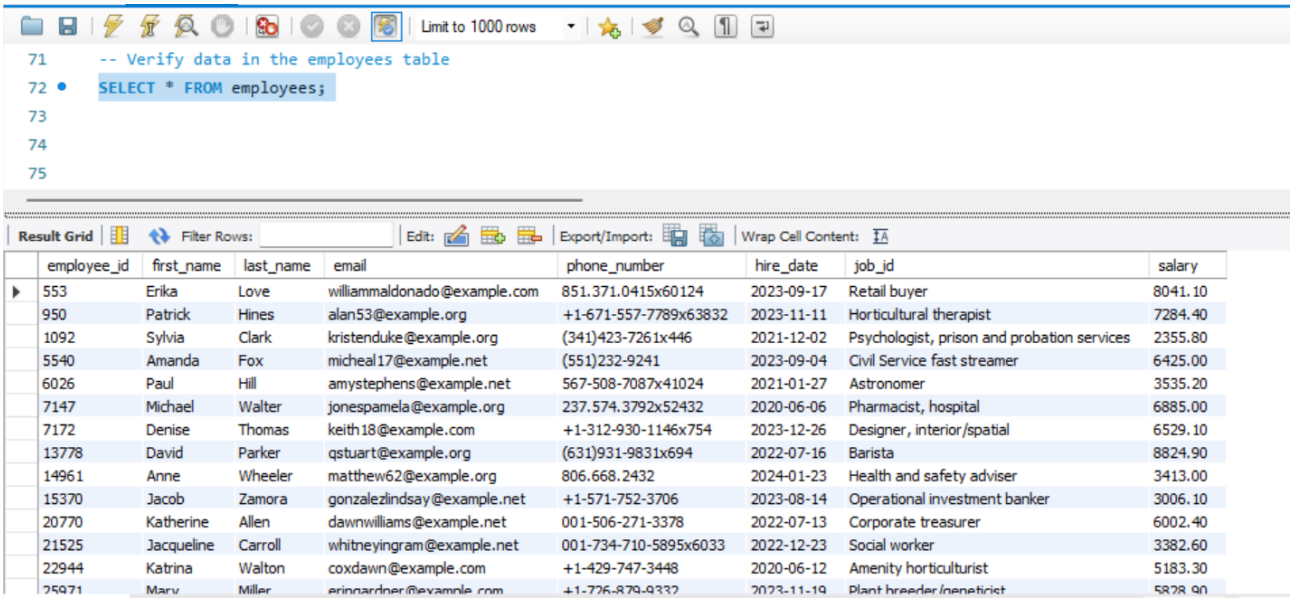




The data is then saved into a CSV file which was loaded into MySQL:

Employees Table Sample:

```
-- Create the employees table
CREATE TABLE IF NOT EXISTS employees (
    employee_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    email VARCHAR(255),
    phone_number VARCHAR(50) NOT NULL,
    hire_date DATE NOT NULL,
    job_id VARCHAR(255),
    salary DECIMAL(10, 2) NOT NULL
);
```



```
71    -- Verify data in the employees table
72 •  SELECT * FROM employees;
73
74
75
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary
553	Erika	Love	williammaldonado@example.com	851.371.0415x60124	2023-09-17	Retail buyer	8041.10
950	Patrick	Hines	alan53@example.org	+1-671-557-7789x63832	2023-11-11	Horticultural therapist	7284.40
1092	Sylvia	Clark	kristenduke@example.org	(341)423-7261x446	2021-12-02	Psychologist, prison and probation services	2355.80
5540	Amanda	Fox	micheal17@example.net	(551)232-9241	2023-09-04	Civil Service fast streamer	6425.00
6026	Paul	Hill	amystephens@example.net	567-508-7087x41024	2021-01-27	Astronomer	3535.20
7147	Michael	Walter	jonespamela@example.org	237.574.3792x52432	2020-06-06	Pharmacist, hospital	6885.00
7172	Denise	Thomas	keith18@example.com	+1-312-930-1146x754	2023-12-26	Designer, interior/spatial	6529.10
13778	David	Parker	qstuart@example.org	(631)931-9831x694	2022-07-16	Barista	8824.90
14961	Anne	Wheeler	matthew62@example.org	806.668.2432	2024-01-23	Health and safety adviser	3413.00
15370	Jacob	Zamora	gonzalezlindsay@example.net	+1-571-752-3706	2023-08-14	Operational investment banker	3006.10
20770	Katherine	Allen	dawnwilliams@example.net	001-506-271-3378	2022-07-13	Corporate treasurer	6002.40
21525	Jacqueline	Carroll	whitneyingram@example.net	001-734-710-5895x6033	2022-12-23	Social worker	3382.60
22944	Katrina	Walton	coxdawn@example.com	+1-429-747-3448	2020-06-12	Amenity horticulturist	5183.30
25971	Mary	Miller	eringardner@example.com	+1-726-879-9332	2023-11-19	Plant breeder/geneticist	5828.90

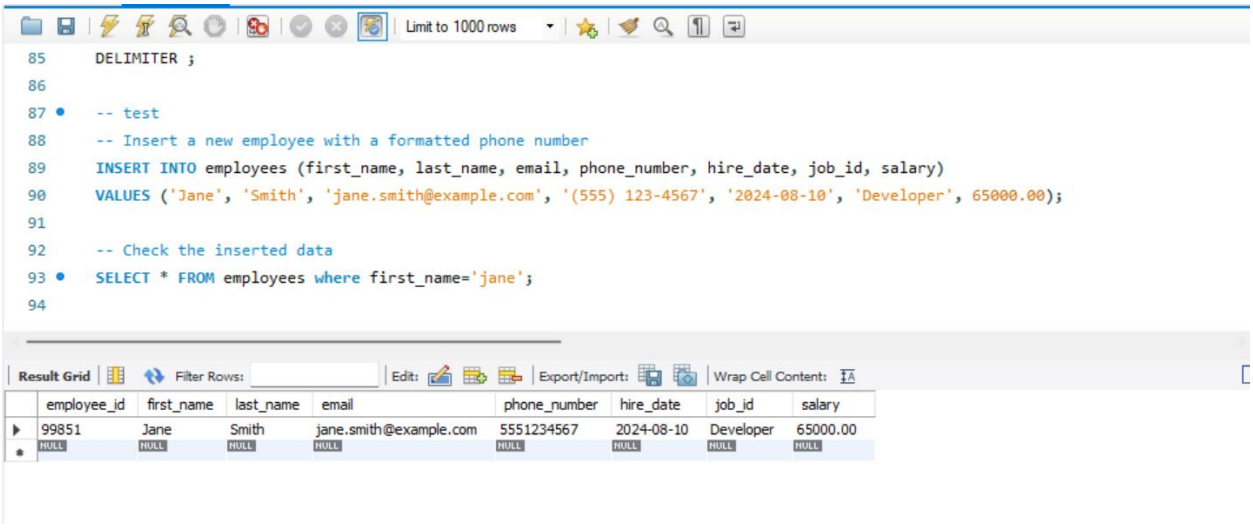
Creating a trigger on the employee table:

The trigger is for formatting employees' phone numbers correctly.

Syntax:

```
CREATE TRIGGER before_employee_insert
BEFORE INSERT ON employees
FOR EACH ROW
BEGIN
    -- Remove non-numeric characters from phone_number
    SET NEW.phone_number = REGEXP_REPLACE(NEW.phone_number, '[^0-9]', '');
END$$
```

Results:



```
85  DELIMITER ;
86
87  -- test
88  -- Insert a new employee with a formatted phone number
89  INSERT INTO employees (first_name, last_name, email, phone_number, hire_date, job_id, salary)
90  VALUES ('Jane', 'Smith', 'jane.smith@example.com', '(555) 123-4567', '2024-08-10', 'Developer', 65000.00);
91
92  -- Check the inserted data
93  SELECT * FROM employees where first_name='jane';
94
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary
99851	Jane	Smith	jane.smith@example.com	5551234567	2024-08-10	Developer	65000.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Booking Data Cleaning:
    - Converted date columns to proper datetime format for easier analysis.(Using excel)
    - Load the data into python for cleaning.

Load the data:

```
[6] import pandas as pd

[7] df=pd.read_csv('hotel_booking.csv')

df.head()
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of
0	Resort Hotel	0	342	2015	July	27	
1	Resort Hotel	0	737	2015	July	27	
2	Resort Hotel	0	7	2015	July	27	
3	Resort Hotel	0	13	2015	July	27	
4	Resort Hotel	0	14	2015	July	27	

5 rows × 36 columns

Handle missing data:

```
# Check for missing values
print(df.isnull().sum())

# Fill missing values for numrics
df.fillna(df.median(numeric_only=True), inplace=True)

# Drop rows with missing values
df.dropna(axis=0, inplace=True)
```

```
hotel                           0
is_canceled                     0
lead_time                       0
arrival_date_year               0
arrival_date_month              0
arrival_date_week_number        0
arrival_date_day_of_month       0
stays_in_weekend_nights         0
stays_in_week_nights            0
adults                          0
children                        0
babies                          0
meal                            0
country                         1
market_segment                  0
distribution_channel            0
is_repeated_guest               0
previous_cancellations          0
previous_bookings_not_canceled  0
reserved_room_type              0
assigned_room_type              0
booking_changes                 0
deposit_type                    0
agent                          96
company                       988
days_in_waiting_list            0
customer_type                   0
adr                             0
required_car_parking_spaces     0
total_of_special_requests       0
reservation_status              0
reservation_status_date         0
name                            0
email                           0
phone-number                    0
credit_card                     0
dtype: int64
```

Results:

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

```
hotel                           0
is_canceled                     0
lead_time                       0
arrival_date_year               0
arrival_date_month              0
arrival_date_week_number        0
arrival_date_day_of_month       0
stays_in_weekend_nights         0
stays_in_week_nights            0
adults                          0
children                        0
babies                          0
meal                            0
country                         0
market_segment                  0
distribution_channel            0
is_repeated_guest               0
previous_cancellations          0
previous_bookings_not_canceled  0
reserved_room_type              0
assigned_room_type              0
booking_changes                 0
deposit_type                    0
agent                           0
company                         0
days_in_waiting_list            0
customer_type                   0
adr                             0
required_car_parking_spaces     0
total_of_special_requests       0
reservation_status              0
reservation_status_date         0
name                            0
email                           0
phone-number                    0
credit_card                     0
dtype: int64
```

Handling duplicates:

Handling duplicates

```
[18] # Drop duplicate rows
     df.drop_duplicates(inplace=True)
```

Feature engineering (create columns that might be useful later)

Feature Engineering

```
[20] df['total_stay_duration'] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
     df['arrival_date_month'] = pd.to_datetime(df['arrival_date_month'], format='%B').dt.month

[21] df.head()
```

ing_spaces	total_of_special_requests	reservation_status	reservation_status_date	name	email	phone-number	credit_card	total_stay_duration
0	0	Check-Out	1/7/2015	Ernest Barnes	Ernest.Barnes31@outlook.com	669-792-1661	**********4322	0
0	0	Check-Out	1/7/2015	Andrea Baker	Andrea_Baker94@aol.com	858-637-6955	**********9157	0
0	0	Check-Out	2/7/2015	Rebecca Parker	Rebecca_Parker@comcast.net	652-885-2745	**********3734	1
0	0	Check-Out	2/7/2015	Laura Murray	Laura_M@gmail.com	364-656-8427	**********5677	1
0	1	Check-Out	3/7/2015	Linda Hines	LHines@verizon.com	713-226-5883	**********5498	2

Renaming columns (Formatting):

```
[23]  # Rename 'phone-number' column to 'phone_number' for consistency
      df.rename(columns={'phone-number': 'phone_number'}, inplace=True)
```

```
df.head()
```

aces	total_of_special_requests	reservation_status	reservation_status_date	name	email	phone_number	credit_card	total_stay_duration
0	0	Check-Out	1/7/2015	Ernest Barnes	Ernest.Barnes31@outlook.com	669-792-1661	**********4322	0
0	0	Check-Out	1/7/2015	Andrea Baker	Andrea_Baker94@aol.com	858-637-6955	**********9157	0
0	0	Check-Out	2/7/2015	Rebecca Parker	Rebecca_Parker@comcast.net	652-885-2745	**********3734	1
0	0	Check-Out	2/7/2015	Laura Murray	Laura_M@gmail.com	364-656-8427	**********5677	1
0	1	Check-Out	3/7/2015	Linda Hines	LHines@verizon.com	713-226-5883	**********5498	2

Save the cleaned data to CSV:

```
[39] df.to_csv('hotels_bbooking_cleaned.csv', index=False)
```

3. Review Data Cleaning:

- Will be used for sentiment analysis

This data cleaning process has prepared our datasets for integration into the SQL database and subsequent analysis in Power BI. The cleaned data will allow for more accurate and meaningful insights into hotel performance, customer satisfaction, and operational efficiency.

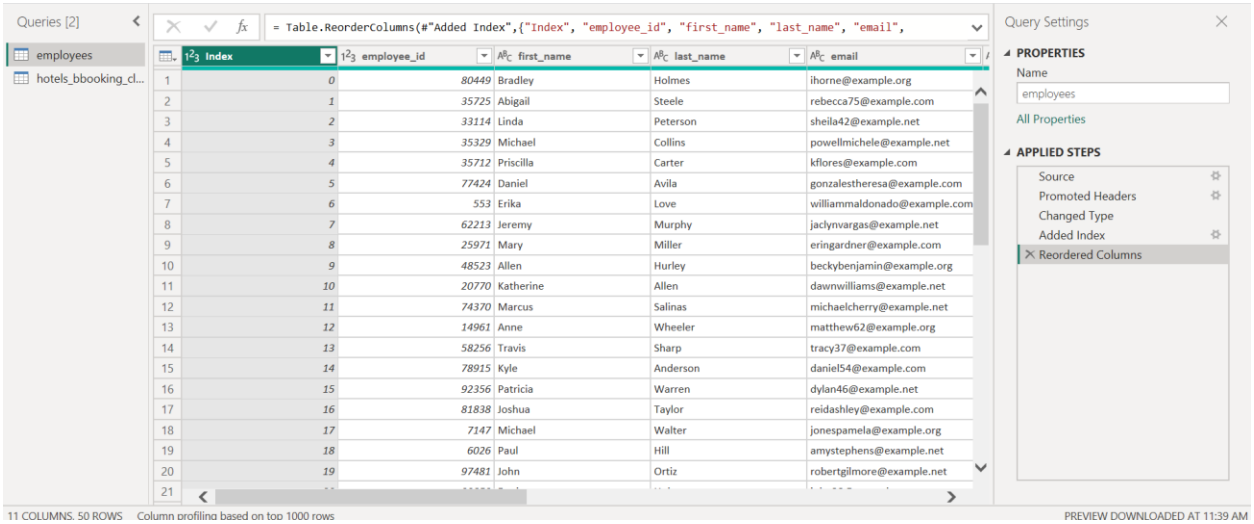
By creating sample data for departments and employees, we've also set the stage for potential analysis of staff performance and departmental efficiency, which could provide valuable insights for hotel management.

2-Data Visualization

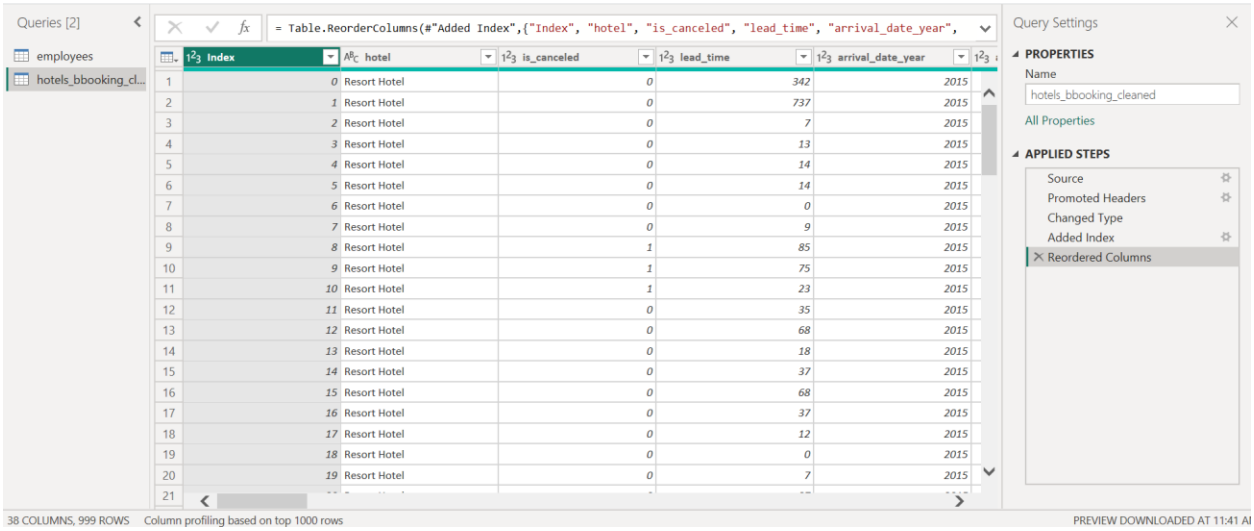
Tools**: Power BI**

After loading the two tables (Cleaned Booking data + Employees) we are going to drill down the tables using power Bi's DAX and Built-in functions:

First, lets add index column to each table, we will use it as a PK to manage the relationship between the tables.



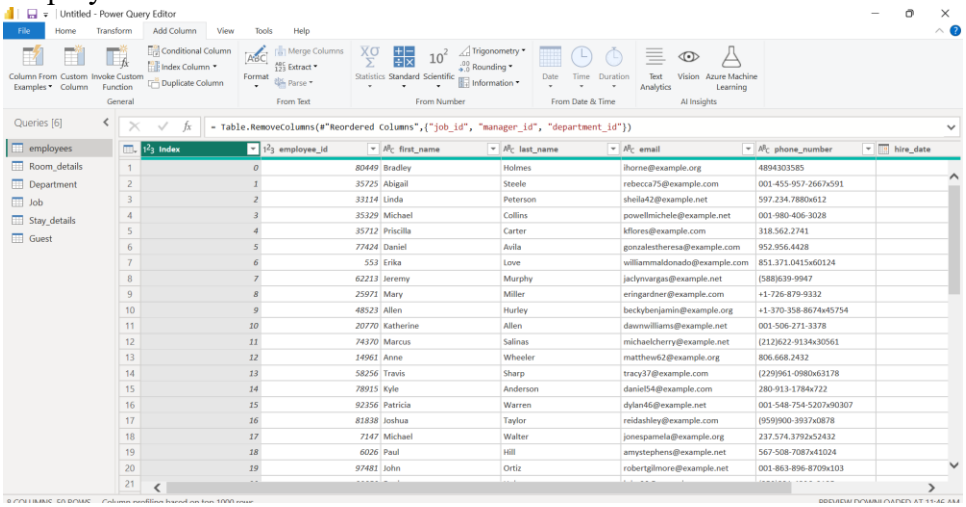
	Index	employee_id	first_name	last_name	email
1	0	80449	Bradley	Holmes	ihorne@example.org
2	1	35725	Abigail	Steele	rebecca75@example.com
3	2	33114	Linda	Peterson	sheila42@example.net
4	3	35329	Michael	Collins	powellmichele@example.net
5	4	35712	Priscilla	Carter	kflores@example.com
6	5	77424	Daniel	Avila	gonzalestheresa@example.com
7	6	553	Erika	Love	williammaldonado@example.com
8	7	62213	Jeremy	Murphy	jaclynvargas@example.net
9	8	25971	Mary	Miller	eringardner@example.com
10	9	48523	Allen	Hurley	beckybenjamin@example.org
11	10	20770	Katherine	Allen	dawnwilliams@example.net
12	11	74370	Marcus	Salinas	michaelcherry@example.net
13	12	14961	Anne	Wheeler	matthew62@example.org
14	13	58256	Travis	Sharp	tracy37@example.com
15	14	78915	Kyle	Anderson	daniel54@example.com
16	15	92356	Patricia	Warren	dylan46@example.net
17	16	81838	Joshua	Taylor	reidashley@example.com
18	17	7147	Michael	Walter	jonespamela@example.org
19	18	6026	Paul	Hill	amystephens@example.net
20	19	97481	John	Ortiz	robertgilmore@example.net
21					



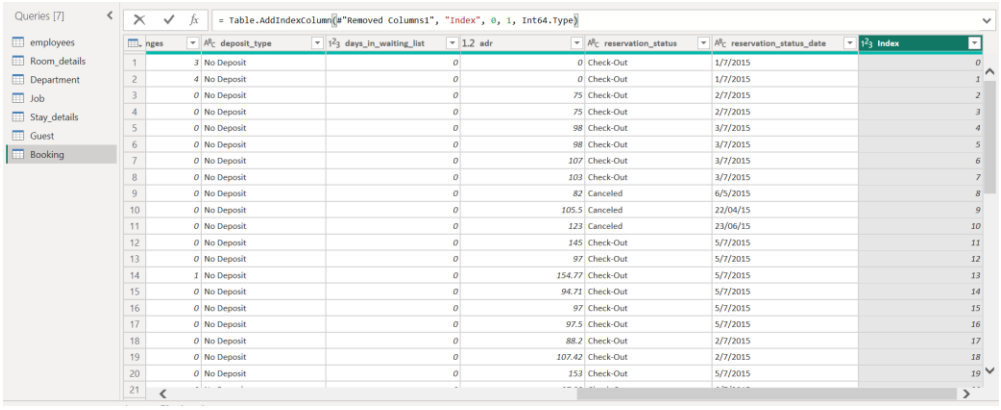
	Index	hotel	is_canceled	lead_time	arrival_date_year
1	0	Resort Hotel	0	342	2015
2	1	Resort Hotel	0	737	2015
3	2	Resort Hotel	0	7	2015
4	3	Resort Hotel	0	13	2015
5	4	Resort Hotel	0	14	2015
6	5	Resort Hotel	0	14	2015
7	6	Resort Hotel	0	0	2015
8	7	Resort Hotel	0	9	2015
9	8	Resort Hotel	1	85	2015
10	9	Resort Hotel	1	75	2015
11	10	Resort Hotel	1	23	2015
12	11	Resort Hotel	0	35	2015
13	12	Resort Hotel	0	68	2015
14	13	Resort Hotel	0	18	2015
15	14	Resort Hotel	0	37	2015
16	15	Resort Hotel	0	68	2015
17	16	Resort Hotel	0	37	2015
18	17	Resort Hotel	0	12	2015
19	18	Resort Hotel	0	0	2015
20	19	Resort Hotel	0	7	2015
21					

The new tables:

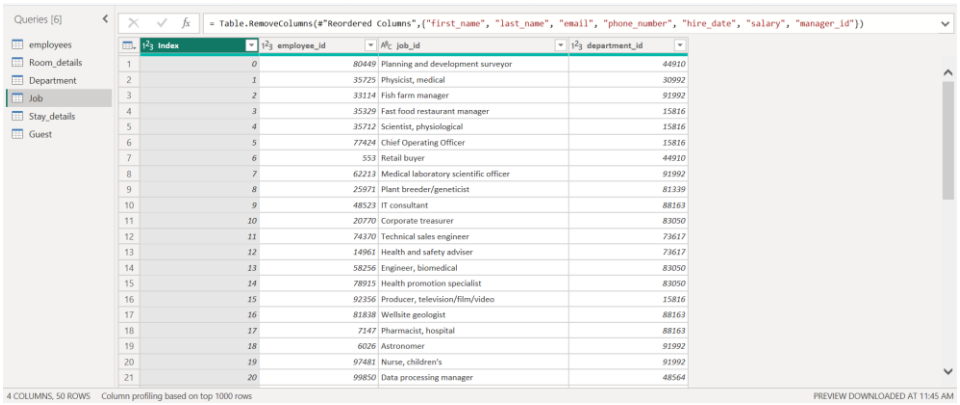
Employee:



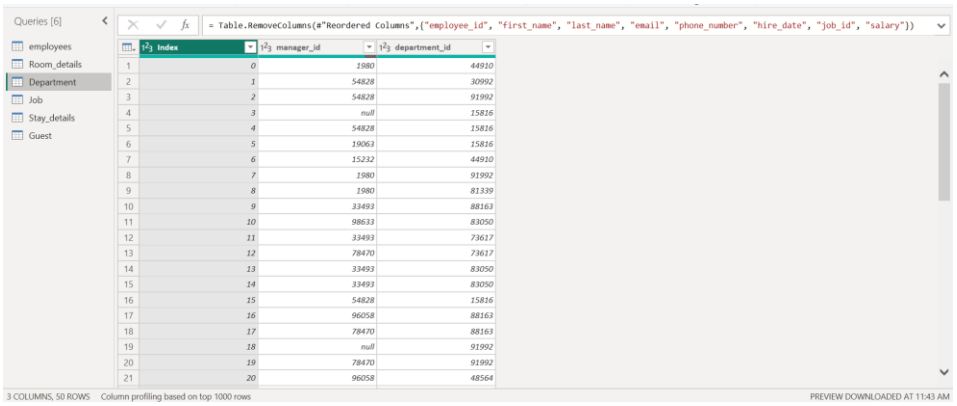
Booking:



Job:

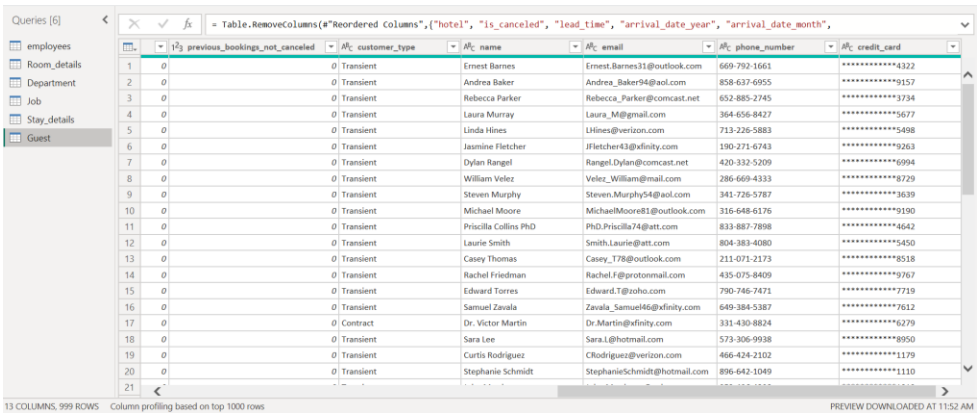


Department:



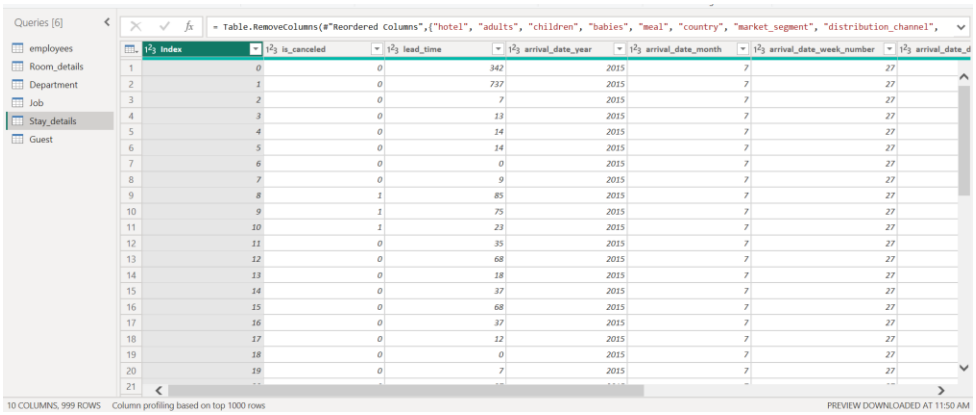
	Index	manager_id	department_id
1	0	1980	44910
2	1	54828	30992
3	2	54828	91992
4	3	null	15816
5	4	54828	15816
6	5	19063	15816
7	6	15232	44910
8	7	1980	91992
9	8	1980	81339
10	9	33493	88163
11	10	98633	83050
12	11	33493	73617
13	12	78470	73617
14	13	33493	83050
15	14	33493	83050
16	15	54828	15816
17	16	96058	88163
18	17	78470	88163
19	18	null	91992
20	19	78470	91992
21	20	96058	48564

Guest:



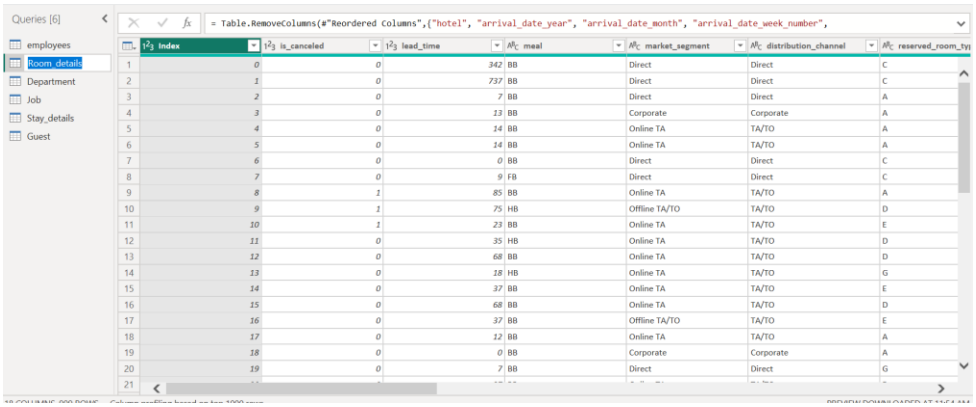
	previous_bookings_not_canceled	customer_type	name	email	phone_number	credit_card
1	0	Transient	Ernest Barnes	Ernest.Barnes31@outlook.com	669-792-1661	**********4322
2	0	Transient	Andrea Baker	Andrea_Baker94@aol.com	858-637-6955	**********9157
3	0	Transient	Rebecca Parker	Rebecca_Parker@comcast.net	652-885-2745	**********3734
4	0	Transient	Laura Murray	Laura_M@gmail.com	364-656-8427	**********5677
5	0	Transient	Linda Hines	LHines@verizon.com	713-226-5883	**********5498
6	0	Transient	Jasmine Fletcher	JFletcher43@xfinity.com	190-271-6743	**********9263
7	0	Transient	Dylan Rangel	Rangel.Dylan@comcast.net	420-332-5209	**********6094
8	0	Transient	William Velez	Velez_William@mail.com	286-669-4333	**********8729
9	0	Transient	Steven Murphy	Steven.Murphy54@aol.com	341-726-5787	**********3639
10	0	Transient	Michael Moore	MichaelMoore81@outlook.com	316-648-6176	**********9190
11	0	Transient	Priscilla Collins PhD	PhD.Priscilla74@att.com	833-887-7898	**********4642
12	0	Transient	Laurie Smith	Smith.Laurie@att.com	804-383-4080	**********5450
13	0	Transient	Casey Thomas	Casey_T78@outlook.com	211-071-2173	**********8518
14	0	Transient	Rachel Friedman	Rachel.F@protonmail.com	435-075-8409	**********9767
15	0	Transient	Edward Torres	Edward.T@zoho.com	790-746-7471	**********7719
16	0	Transient	Samuel Zavala	Zavala_Samuel46@xfinity.com	649-384-5387	**********7612
17	0	Contract	Dr. Victor Martin	Dr.Martin@xfinity.com	331-430-8824	**********6279
18	0	Transient	Sara Lee	Sara.L@hotmail.com	573-306-9938	**********8050
19	0	Transient	Curtis Rodriguez	CRodriguez@verizon.com	466-424-2102	**********1179
20	0	Transient	Stephanie Schmidt	StephanieSchmidt@hotmail.com	896-642-1049	**********1110
21						

Stay_details:



	Index	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_d
1	0	0	342	2015	7	27	
2	1	0	737	2015	7	27	
3	2	0	7	2015	7	27	
4	3	0	13	2015	7	27	
5	4	0	14	2015	7	27	
6	5	0	14	2015	7	27	
7	6	0	0	2015	7	27	
8	7	0	9	2015	7	27	
9	8	1	85	2015	7	27	
10	9	1	75	2015	7	27	
11	10	1	23	2015	7	27	
12	11	0	35	2015	7	27	
13	12	0	68	2015	7	27	
14	13	0	18	2015	7	27	
15	14	0	37	2015	7	27	
16	15	0	68	2015	7	27	
17	16	0	37	2015	7	27	
18	17	0	12	2015	7	27	
19	18	0	0	2015	7	27	
20	19	0	7	2015	7	27	
21							

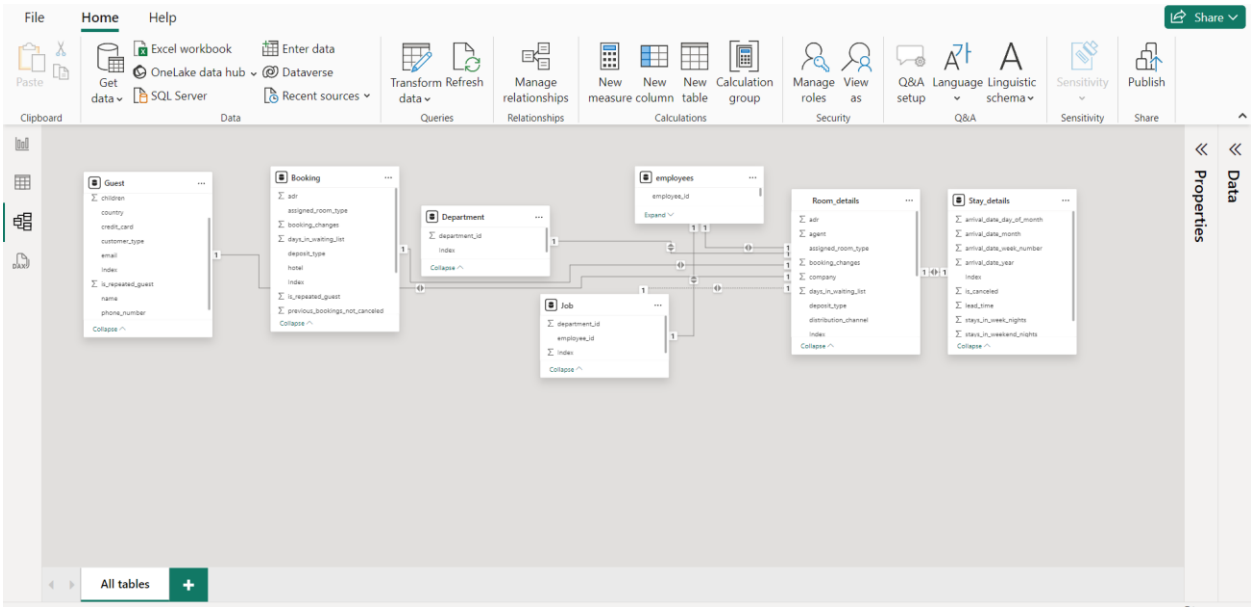
Room_Details:



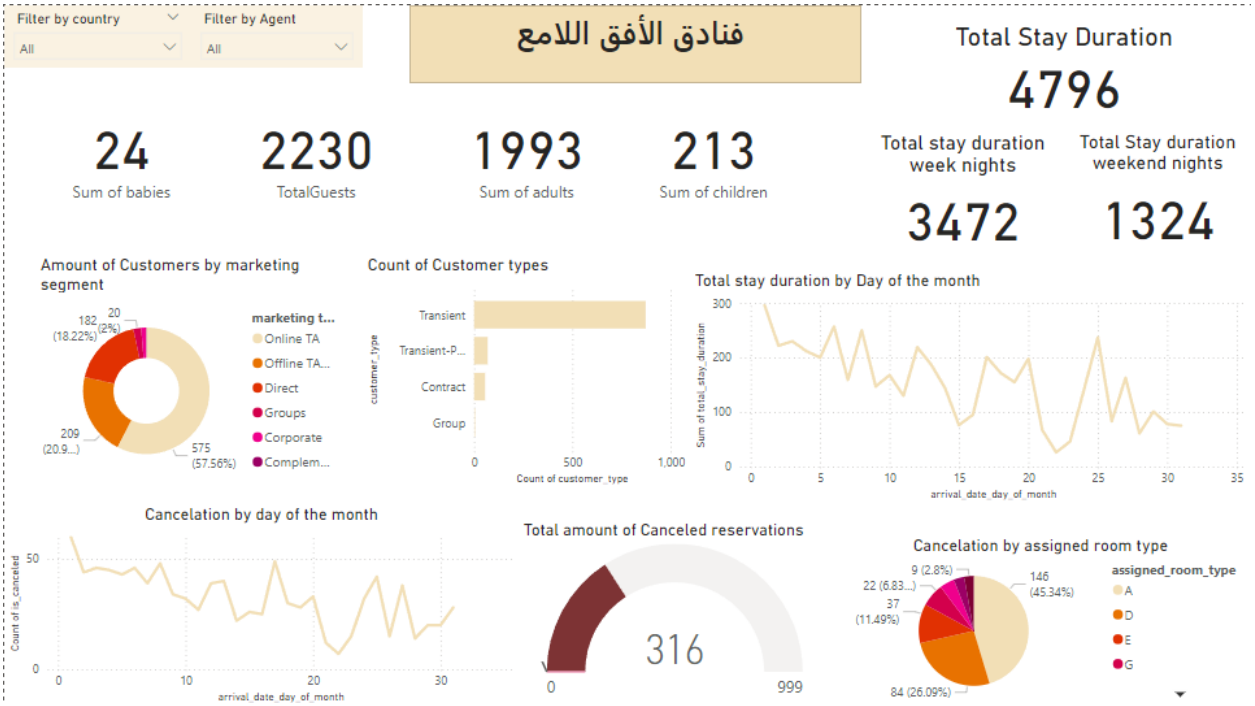
Summary

- **Employee Table:** Contains core employee information.
- **Job Table:** Contains job-related details.
- **Department Table:** Contains department-related details.
- **Booking Table:** Contains booking-related information.
- **Guest Table:** Contains guest-specific details.
- **Stay Details Table:** Contains stay-specific details.
- **Room Details Table:** Contains room-related details.

Model View:


File
Home
Help
Share
Paste
Get data
Excel workbook
OneLake data hub
SQL Server
Enter data
Dataverse
Recent sources
Transform data
Refresh
Manage relationships
New measure
New column
New table
Calculation group
Manage roles
View as
Q&A setup
Language
Linguistic schema
Sensitivity
Publish
Clipboard
Data
Queries
Relationships
Calculations
Security
Q&A
Sensitivity
Share
Guest
Booking
Department
employees
Job
Room_details
Stay_details
Properties
Data
All tables

3- Visualization analysis:



Observations:

1. Correlation between Guest Types and Stay Duration:
    - The total number of guests (2230) is composed of 1993 adults, 213 children, and 24 babies.
    - The total stay duration of 4796 nights is a combination of 3472 weeknight stays and 1324 weekend night stays.
    - This suggests that the adult guests are likely responsible for the majority of the weeknight stays, while the guests with children (213 children + 24 babies) may be contributing more to the weekend night stays.
2. Segmentation and Booking Behavior:
    - The largest marketing segment is "Offline TA" (57.54% of customers), followed by "Direct" (20%) and "Online TA" (13.22%).
    - This implies that the hotel has a strong relationship with offline travel agencies, and a significant portion of its business comes from this channel.

- However, the presence of "Direct" and "Online TA" segments suggests that the hotel is also attracting direct bookings and leveraging online travel agencies.

3. Cancellation Patterns and Room Type:
    - The dashboard shows that the total amount of canceled reservations is 316.
    - Looking at the "Cancellation by assigned room type" chart, we can see that the majority of canceled reservations (84 or 26.58%) are for room type "A", followed by room types "E" (146 or 46.20%) and "D" (45 or 14.24%).
    - This could indicate that certain room types are more prone to cancellations, potentially due to factors like pricing, availability, or guest preferences. Understanding these patterns can help the hotel optimize its room inventory and pricing strategies.
4. Seasonal Trends in Stay Duration:
    - The "Total Stay Duration by Day of the Month" chart shows some fluctuations in the total stay duration throughout the month.
    - These fluctuations could be related to seasonal factors, such as weekends, holidays, or specific events that influence the hotel's occupancy and stay patterns.
    - Analyzing these trends over time can help the hotel anticipate demand and adjust its operations and pricing accordingly.
5. Cancellation Trends and Day of the Month:
    - The "Cancellation by day of the month" chart displays the number of cancellations per day of the month.
    - Identifying any patterns or peaks in cancellations could help the hotel understand the underlying factors driving these cancellations, such as changes in travel plans, last-minute bookings, or specific events.
    - This information can be used to implement proactive measures to reduce cancellations, such as adjusting cancellation policies, improving communication with guests, or offering incentives for guests to maintain their reservations.

4- Extra Credit:

Tools: Google colab

Sentiment analysis model:

In this task, we developed a machine learning application to perform sentiment analysis on hotel reviews using Python in Google Colab. The goal was to classify hotel reviews as either positive (1) or negative (0) based on the text content.

Steps Undertaken:

1- Data Loading and Exploration:

We began by loading the (hotel_Reviews.csv) dataset and exploring its structure using basic pandas functions. This included checking for missing values and reviewing the first few rows of the dataset to understand the content.

2-Data Preprocessing:

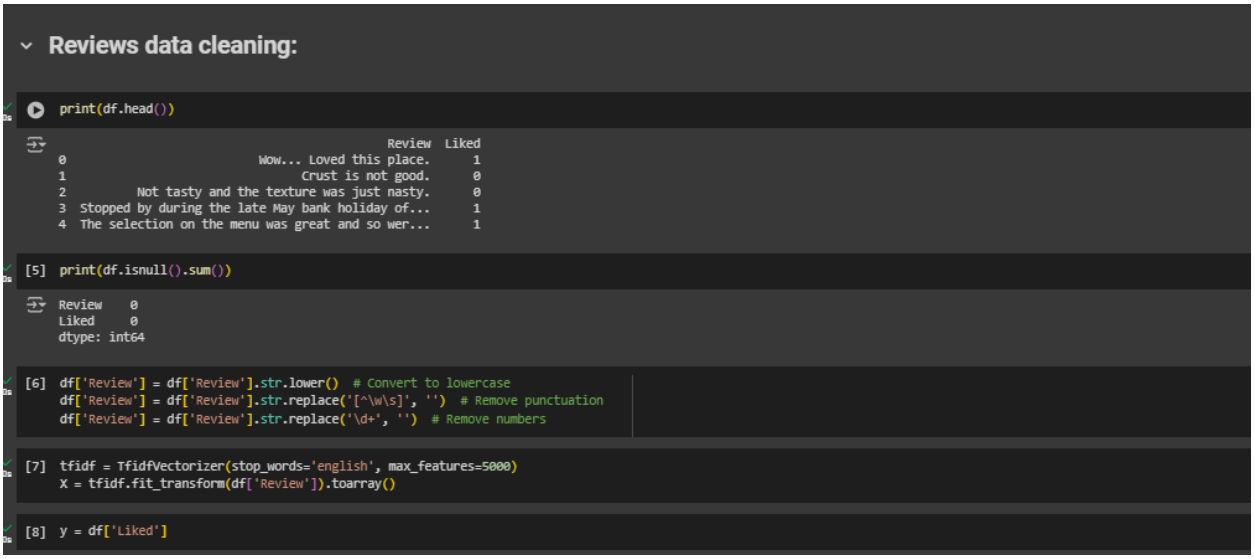
The dataset had a 'Review' column containing the review text and a ‘Liked’ column with binary values (1 for positive, 0 for negative). The text data in the 'Review' column was then preprocessed:

Converted all text to lowercase to ensure uniformity.

Removed punctuation and numbers to focus only on the words that carry meaning.

3- Feature Extraction:

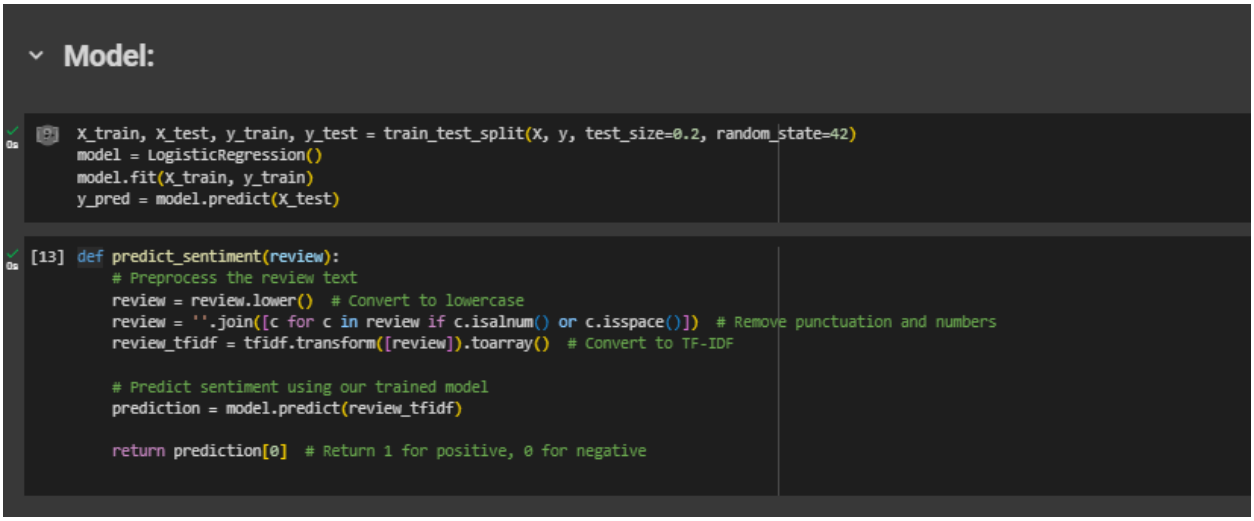
We converted the cleaned text data into numerical form using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This technique transforms the text into a matrix of features that the machine learning model can understand.



4-Model Building and Training:

We split the dataset into training and testing sets using an 80-20 split.

A Logistic Regression model was then trained on the TF-IDF features extracted from the training set.



5-Model Evaluation:

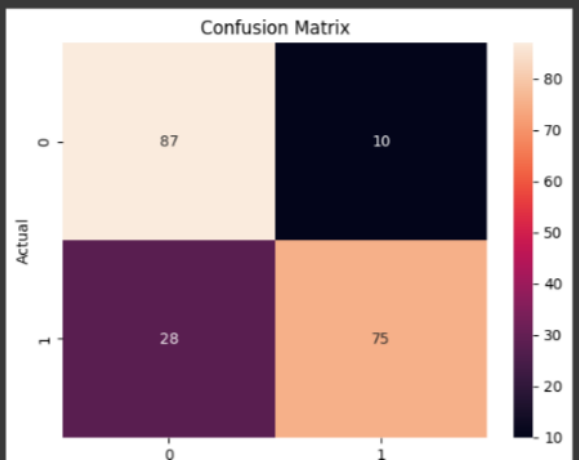
The model's performance was evaluated on the testing set using metrics such as accuracy, classification report (precision, recall, F1-score), and a confusion matrix. These metrics provided insights into how well the model was performing in classifying reviews as positive or negative.

Evaluate

```
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

print('Classification Report:')
print(classification_report(y_test, y_pred))
```

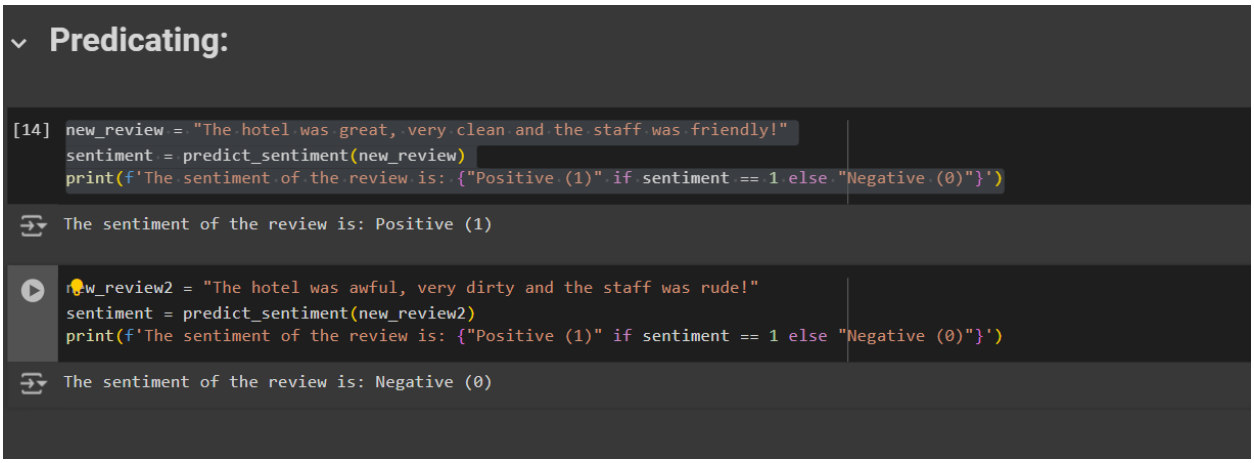
```
Accuracy: 0.81
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.90      0.82        97
           1       0.88      0.73      0.80       103

    accuracy                           0.81       200
   macro avg       0.82      0.81      0.81       200
weighted avg       0.82      0.81      0.81       200
```

```
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



6-Sentiment Prediction for New Reviews:

A custom function, predict sentiment, was created to take a new review as input, preprocess it, and use the trained model to predict its sentiment. This allows for real-time sentiment analysis on new reviews, outputting either a positive (1) or negative (0) classification.



7-Conclusion:

The final product is a machine learning model capable of performing sentiment analysis on hotel reviews, which can be used to predict whether a review is positive or negative. This application could be valuable for understanding customer feedback, improving services, and enhancing overall customer satisfaction in the hospitality industry.