Project Development Phase
Number of functional features included in the solution

Date	04-11-23
Team ID	592499
Project Name	Machine learning model for occupancy rates and demand in hospitality industries

Team Members :

N Nitin : nitin.n2021@vitstudent.ac.in

Mekala Sujan : mekala.sujan2021@vitstudent.ac.in

The number of functional features to include in a machine learning model for predicting occupancy rates and demand in the hospitality industry can vary depending on the specific dataset and business context. However, here are some common functional features that are often included in such models:

1. Historical Occupancy Rates: Historical occupancy rates for the property over time are essential for understanding trends and seasonality.

2. Room Rates: Pricing information for rooms, including daily rates, promotions, and discounts.

3. Date and Time Features:
   - Day of the week
   - Month
   - Year
   - Public holidays and special events
   - Seasonal indicators

4. Weather Data: Weather-related features such as temperature, precipitation, and weather conditions can affect demand.

5. Competitor Data: Information about competitors, their pricing strategies, and occupancy rates in the local area.

6. Marketing and Promotion Data: Data on marketing campaigns, advertising spend, and promotional activities.

7. Local Events: Information about local events, conferences, festivals, or other activities that can influence demand.

8. Booking Lead Time: The number of days between the booking date and the check-in date, as this can impact demand.

9. Length of Stay: Average length of stay for previous guests.

10. Online Reviews and Ratings: Guest reviews and ratings from platforms like TripAdvisor, Yelp, or Google can provide insights into demand trends.

11. Cancellation Rate: The rate at which reservations are canceled, which can affect occupancy.

12. Economic Indicators: Local and national economic factors that may influence travel and hospitality trends.

13. Property Features: Features of the property itself, such as the number of rooms, amenities, and room types.

14. Market Demand: Historical data on overall market demand in the hospitality industry in the local area.

15. Booking Channels: Information on the channels through which bookings are made, such as direct website bookings, online travel agencies (OTAs), or phone reservations.

16. Customer Segmentation: Information about different customer segments, such as business travelers, leisure travelers, or group bookings.

17. Historical Data on Special Packages: Data on packages and deals offered by the hotel in the past.

18. Location Data: Information about the property's location, proximity to attractions, transportation hubs, and local demand generators.

19. Regulatory Changes: Data on any local regulations or policies that may impact occupancy and demand.

It's important to note that feature selection should be data-driven and based on the correlation of each feature with the target variable (occupancy rates and demand). Feature engineering and selection should be an iterative process, where you assess the importance of each feature and potentially remove or add features as needed to improve the model's predictive accuracy.

Additionally, using techniques such as feature importance analysis, correlation analysis, and domain knowledge can help you identify the most relevant functional features for your specific use case.

In addition to these features, some models may also use more specialized features, such as customer sentiment data from online reviews or social media data.

The number of functional features that are included in a model is a trade-off between accuracy and complexity. More features can lead to more accurate predictions, but they can also make the model more complex and difficult to train and interpret.

Here is an example of a machine learning model for occupancy rates and demand in hospitality industries that includes 19 functional features:

```
import pandas as pd

# Create a list of features
features = ["date", "day_of_week", "month", "year", "is_holiday", "is_weekend",
        "number_of_events", "average_temperature", "average_humidity",
        "average_wind_speed", "average_precipitation", "occupancy_rate"]

# Create a DataFrame
df = pd.DataFrame(features)

# Train a machine learning model
model = ...

# Make predictions
predictions = model.predict(df)
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load the data
df = pd.read_csv("occupancy_rates_and_demand.csv")

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(df[features], df["occupancy_rate"], test_size=0.25)
```

```
# Create a linear regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)

# Calculate the mean absolute percentage error (MAPE)
mape = sum(abs(predictions - y_test) / y_test) / len(y_test) * 100

# Print the MAPE
print(mape)
```

This code will train a linear regression model to predict occupancy rates based on the functional features. The model is then evaluated on the test set using the MAPE metric.

A MAPE of less than 10% is generally considered to be good accuracy. However, the desired level of accuracy will depend on the specific hotel or hospitality business.

Here is an example of how to use the code to predict occupancy rates for a future date:

```
# Create a DataFrame for the future date
future_date_df = pd.DataFrame({"date": ["2023-12-31"]})

# Add the other functional features to the DataFrame
future_date_df["day_of_week"] = future_date_df["date"].dt.dayofweek
future_date_df["month"] = future_date_df["date"].dt.month
future_date_df["year"] = future_date_df["date"].dt.year
future_date_df["is_holiday"] = future_date_df["date"].dt.is_holiday
future_date_df["is_weekend"] = future_date_df["date"].dt.is_weekend

# Predict the occupancy rate for the future date
predicted_occupancy_rate = model.predict(future_date_df)

# Print the predicted occupancy rate
print(predicted_occupancy_rate)
```

This code will print the predicted occupancy rate for the future date, based on the values of the functional features.