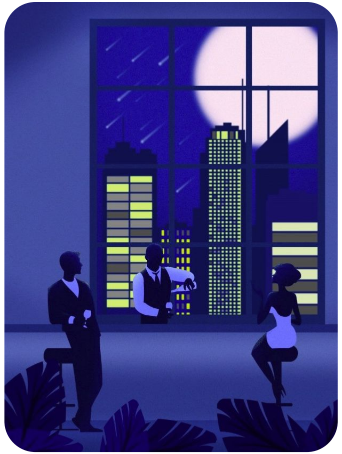
Hotel Revenue Management Analysis and Strategic Action

*An in-depth data-driven analysis to **improve business performance** using Large Language Models (LLMs)*

Team Members:

1. Andreas Lukita (A0221743M) as Gemini
2. Yu Bixun (A0218248B) as ChatGPT
3. Yu Tianyi (A0218288U) as Claude



Phase 1: Business Problem

Yu Tianyi (A0218288U)

Did you know?

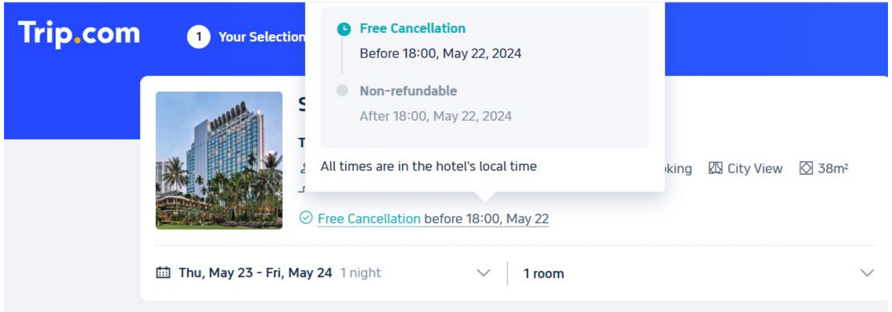
(2024 Global hotel industry statistics)

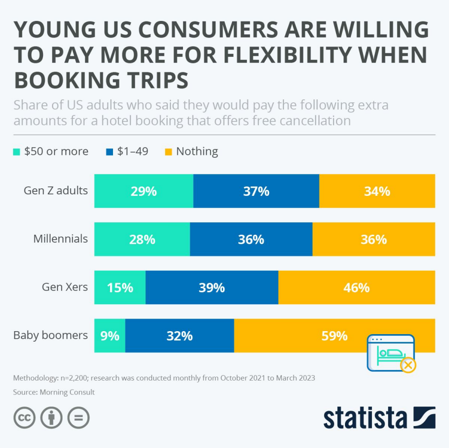
Out of ***5*** *hotel room reservations made by customers,*

1/5 *bookings are cancelled*

Yu Tianyi (A0218288U)

Hotel management are experiencing challenges in revenue management due to increased in booking cancellations.





Sources: Esther Hertzfeld, SiteMinder, Statista

Key Insights

As shown from the infographics, hotel management is having **challenges maintaining their occupancy rate** due to **increased in booking cancellations.**

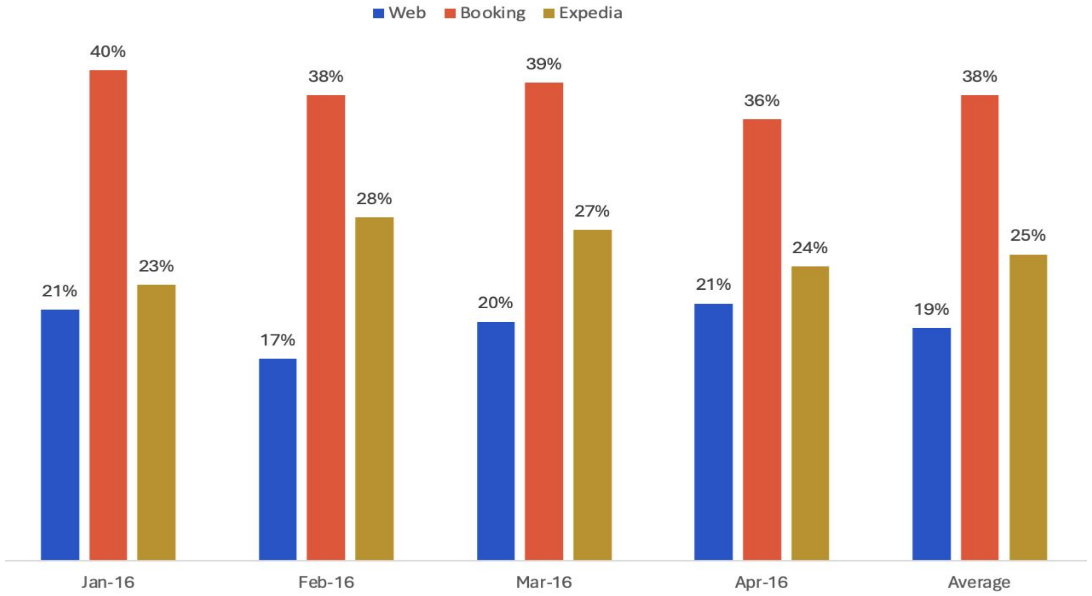
Younger customers value flexibility when booking trips, contributing to the high figure of booking cancellations.

If continue unaddressed, this would inevitably **hurt the management's bottom line**.

Yu Tianyi (A0218288U)

Pinpointing the exact cancellation rates are not simple. However, from online sources, we guesstimate that the figure fluctuates around 28%.

Booking Cancellation Rates across OTAs



Sources: mirai

Key Insights

Data on room prices and availability are **dynamic**. Booking cancellations seem to be affected by many factors.

Proprietary information. Many hotels and OTAs treat cancellation data as confidential.

Data specificity and scope. Statistics across space and time as well as the type of hotel could differ.

Yu Tianyi (A0218288U)

Based on these issues, we suggest that the hotel management adopts the Overbooking strategy to maximise occupancy and hence revenue.

1. What is the trade-off that the management faces between **underbooking** and **overbooking**?

2. What are the **characteristics** and signs of customers who are likely to **cancel** their bookings? Can we identify any pattern?

3. How do we **value-add** to the hotel management and **empower** their decision-making process?

Yu Tianyi (A0218288U)

Let’s skit!

Andreas Lukita (A0221743M)

Andreas Lukita (A0221743M)

Welcome our two newly-hired analysts — John and Thomas

A fresh graduate with little to no experience in DSA, **ChatGPT is his soulmate**

Trained in understanding and processing data, **work collaboratively with LLM**

Andreas Lukita (A0221743M)

Welcome our two newly-hired analysts — John and Thomas

A fresh graduate with little to no experience in DSA,
ChatGPT is his soulmate

Andreas Lukita (A0221743M)

Welcome our two newly-hired analysts — Thomas

Trained in understanding and processing data, **work collaboratively with LLM**

Andreas Lukita (A0221743M)

Pick your side wisely, let's see John and Thomas in action!

Streamlit

GitHub

https://hotel-reservation-dba4714.streamlit.app/

https://github.com/AndreasL7/hotel_reservation

Andreas Lukita (A0221743M)

Phase 2:
Data and Methodology

Yu Tianyi (A0218288U)

The team decided to use these 3 LLMs for purpose of analysis and tackling the problem.

Model	Strengths	Weaknesses
	• Strong performance in math, general knowledge and reasoning • Capable of taking in uploaded code and perform programming in its sandbox environment	• Only contains information up until Apr 2023 (latest update) • May run into internal error in doing the programming
	• Excels in deep analysis, problem-solving, and seamless integration with Google workspace • Somewhat better at explaining and generally give a more friendly conversational tone • Connected to Google Search, thus provides real time data	• Relatively new as compared to the GPT models • Tend to give long-winded instead of direct response
	• Able to understand and comprehend prompts effectively and provides concise explanations • Allows for uploading of excel file for data processing even for the free models with limitations	• Humour slowly fades away, at the first few prompts “clears throat” & “puts on data scientist hat” • Unengaging graphical outputs and visualisations

Yu Tianyi (A0218288U)

The team prompts the LLMs using a set of Primary and Secondary questions

Primary questions

"Perform train test split on my hotel data with test size 0.2 and stratify the target."

"Next, please perform univariate analysis on the dataset using the appropriate visualisation technique."

Yu Tianyi (A0218288U)

The team prompts the LLMs using a set of Primary and Secondary questions

Secondary questions

"Is it possible to drop all the rows with 29 February in 2018 date since there is no 29th February in non-leap years?"

"Here are some observations:
It seems that canceled booking status are more prevalent for lead time above 150 days."

Yu Tianyi (A0218288U)

The team used a set of common questions to prompt the LLMs to get different outputs from all 3 LLMs for the purpose of analysis and to solve the business problem.

Prompts | **Output**

Usage of Persona

You
Hi. Firstly, I need you to become the role of a Data Scientist who will be assisting me in my project in tackling business problems.

ChatGPT
I'm ready to step into the role of a Data Scientist for you! What specific business problems are we looking at, and how can I assist you in your project?

Instruction Prompting with directional stimuli

I need to solve a case study on revenue management for Hotels. The hotel business is trying to maximise their profits from their revenue stream of hotel reservations. Hotel reservations can be cancelled at a low cost or free-of-charge. Please formulate and give me some suitable business problems or problem statements for...

Problem Statements

1. **Overbooking Optimization:**
   - How can we develop a strategy that optimizes overbooking to offset cancellations while minimizing guest inconvenience and negative reputation?

Prompt Completion

You
Hotel Reservations.csv
Spreadsheet

Here is the dataset hotel_reservation.csv

import

```
import pandas as pd

# Load the dataset
df = pd.read_csv('/mnt/data/Hotel Reservations.csv')

# Display the first few rows of the dataframe
df.head()
```

Yu Tianyi (A0218288U)

Phase 3:
Descriptive Analytics

Yu Tianyi (A0218288U)

The team performed data quality check and some univariate analysis to learn more about the data.

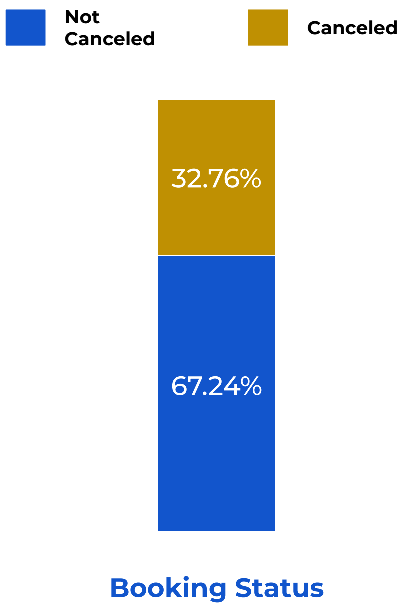
About Dataset

1. Data Quality
a. There are 0 duplicates
b. **No missing values**
c. No signs of abnormal unique values

2. Target Variable
a. Binary variable of ‘Canceled’ or ‘Not Canceled’
b. ‘Canceled’ being **⅓** of dataset, potentially suggests data imbalance

3. Features
a. Contains **17 Features** (excluding Booking_ID and Target Variable)

Target Variable Distribution



Yu Tianyi (A0218288U)

Some of the charts in exploratory analysis generated by LLMs’ code were useful and appropriate for usage.

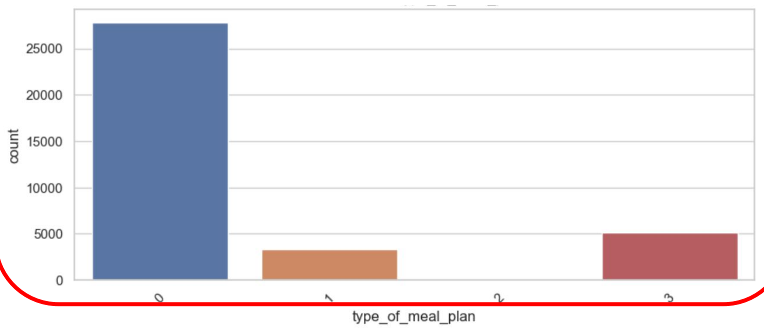
Key Insights

- Most codes provided by LLMs can provide efficiency to progress of project
- Wide variety of graphs can be generated quickly

```
# For categorical variables, we'll display their distribut
categorical_cols = ['type_of_meal_plan', 'required_car_par
                    'market_segment_type', 'repeated_guest

for col in categorical_cols:
    plt.figure(figsize=(10, 4))
    sns.countplot(x=col, data=df)
    plt.title(f'Distribution of {col}')
    plt.xticks(rotation=45)
    plt.show()
```

Charts



Yu Tianyi (A0218288U)

Some of the charts in exploratory analysis generated by LLMs' code were useful and appropriate for usage.

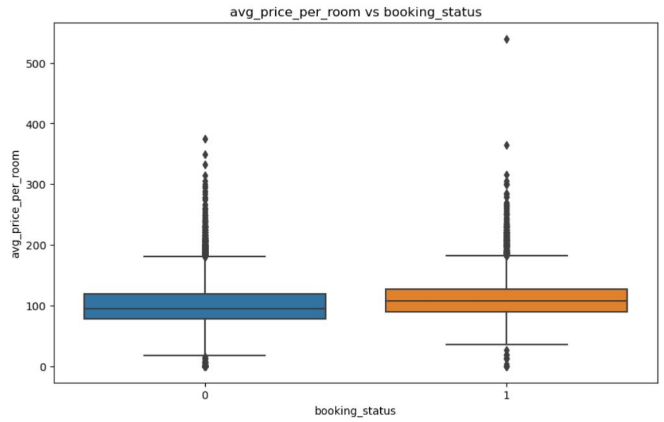
Key Insights

- Most codes provided by LLMs can provide efficiency to progress of project
- Wide variety of graphs can be generated quickly

```
# Bivariate analysis with numerical variables
numerical_cols = ['no_of_adults', 'no_of_children', 'no_of_weekend_nights', 'no_of_week_ni
                  'lead_time', 'no_of_previous_cancellations', 'no_of_previous_bookings_no
                  'avg_price_per_room', 'no_of_special_requests']

print("Relationship between numerical variables and booking_status:")
for col in numerical_cols:
    plt.figure(figsize=(10, 6))
    sns.boxplot(x='booking_status', y=col, data=hotel_data)
    plt.title(f"{col} vs booking_status")
    plt.show()
```

Charts



Yu Tianyi (A0218288U)

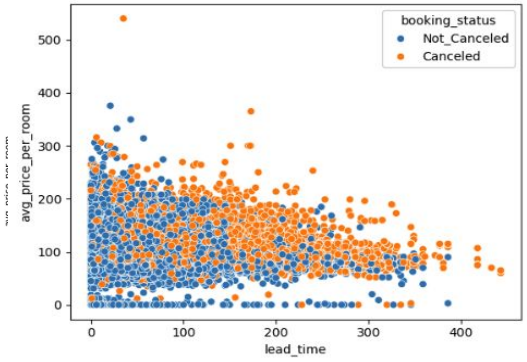
Some of the charts in exploratory analysis generated by LLMs' code were useful and appropriate for usage.

Key Insights

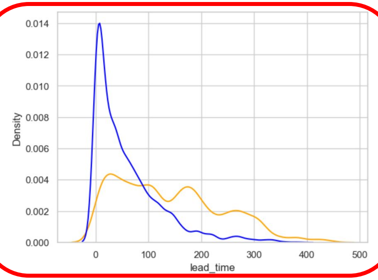
- Most codes provided by LLMs can provide efficiency to progress of project
- Wide variety of graphs can be generated quickly

Charts

```
sns.scatterplot(x='lead_time', y='avg_price_per_room', data=data, hue='booking_status')
plt.show()
```




```
sns.kdeplot(data=data[data['booking_status'] == 'Canceled'], x='lead_time', color='orange')
sns.kdeplot(data=data[data['booking_status'] == 'Not_Canceled'], x='lead_time', color='blue');
```



Yu Tianyi (A0218288U)

Phase 4: Predictive Analytics

Yu Bixun (A0218248B)

In predictive analysis, LLMs were able to generate code to support user in performing the necessary steps to build a working model.

Feature scaling

```python
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Select columns for scaling
numerical_features = ['lead_time', 'avg_price_per_room']

# Initialize scalers (choose one for each feature)
min_max_scaler = MinMaxScaler()
standard_scaler = StandardScaler()

# Fit and transform the data
data[numerical_features] = min_max_scaler.fit_transform(data[numerical_
# OR
data[numerical_features] = standard_scaler.fit_transform(data[numerical
```

Hyperparameter tuning

```python
import xgboost as xgb
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV

# Example XGBoost model (adapt parameters as needed)
xgb_model = xgb.XGBClassifier(objective='binary:logistic', seed=42)

# Parameters to tune (refer to XGBoost documentation for more options)
param_grid = {
    'learning_rate': [0.01, 0.1],
    'max_depth': [3, 6, 10],
    'n_estimators': [50, 100, 200],
    'gamma': [0, 0.1, 0.2]
}

# Choose between GridSearchCV or RandomizedSearchCV
search = GridSearchCV(xgb_model, param_grid, cv=5, scoring='roc_auc')

search.fit(X_train, y_train)

best_model = search.best_estimator_
y_pred = best_model.predict(X_test)

# ... (Calculate evaluation metrics as before)
```

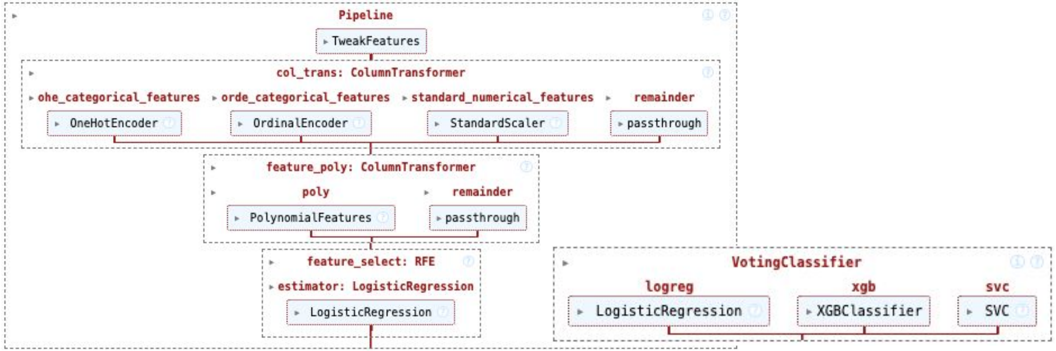
Key Insights

1. Coding for **feature engineering, scaling, transformation**, etc.
2. LLMs are able to recommend well-thought **new features** to enrich the feature space
3. Distinguish between different encoding **techniques**
4. Different types of predictive **models** ranging from logistic regression to XGBoost

Yu Bixun (A0218248B)

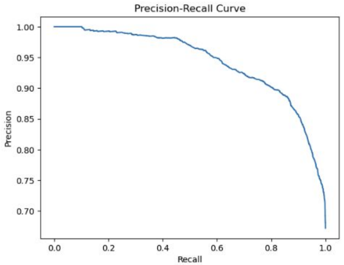
In predictive analysis, LLMs were able to generate code to support user in performing the necessary steps to build a working model.

Pipeline



Model Evaluation

	precision	recall	f1-score	support
Canceled	0.77	0.69	0.73	2376
Not_Canceled	0.86	0.90	0.88	4870
accuracy			0.83	7246
macro avg	0.81	0.80	0.80	7246
weighted avg	0.83	0.83	0.83	7246



Key Insights

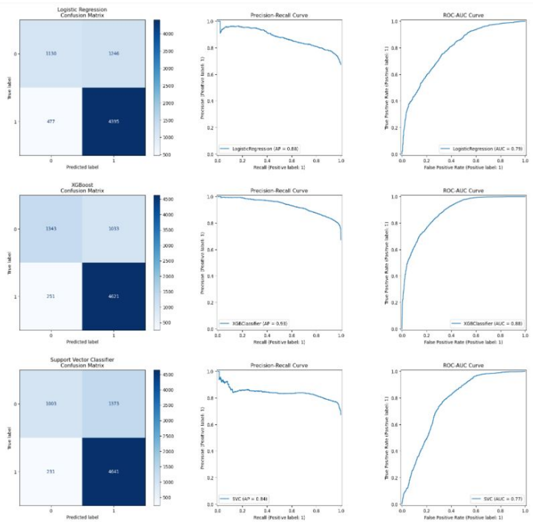
1. While most LLMs do not generally recommend best practice by default, upon prompted, most are able to provide the solution needed

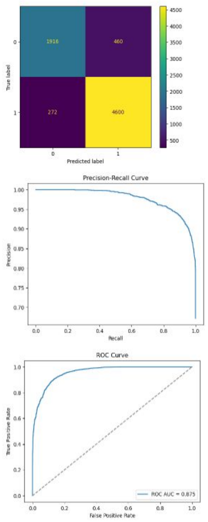
2. Able to recognise the need to focus on **different metrics** to evaluate model's performance, ranging from F1 Score to AUC-ROC

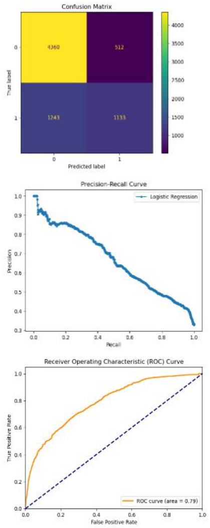
Yu Bixun (A0218248B)

In predictive analysis, LLMs were able to generate code to support user in performing the necessary steps to build a working model.

Exploring different machine learning models







Key Insights

1. Exploring different models allows for **performance comparison** to identify which model is better suited for the specific task at hand

2. Using **directional stimulus prompting** technique to specify the models that we want to explore such as XGBoost and SVC

Yu Bixun (A0218248B)

The team analysed and measure the performance of the predictive models by all 3 LLMs and comparing them to code that is programmed with human intervention.

Logistic Regression Model Performance (Tuned)					
Model	Accuracy	Precision	Recall	F1 Score	AUC Score
	0.76	0.69	0.48	0.56	0.69
	0.81	**0.84**	**0.89**	**0.87**	**0.77**
	0.76	0.78	0.90	0.84	0.79

Yu Bixun (A0218248B)

Phase 5:
Limitation of LLMs

Yu Bixun (A0218248B)

However, there are some limitations of LLMs in tackling coding and data science projects.

Limitations

QN: Are LLMs intelligent and self-aware?

LLMs are not intelligent nor sentient!

“AI is not perfect, we’re all learning along the way… that’s what makes it fun!”

- Lisa Su, CEO of AMD

Yu Bixun (A0218248B)

However, there are some limitations of LLMs in tackling coding and data science projects.

Limitations

1. Non-retention of context

- LLMs may not retain or "remember" context set earlier in conversation
- May impact results of the model later in the project

Used **ordinal encoding** initially but later on uses **one-hot encoding** on 'lead_time_category'

1

```
# Apply ordinal encoding manually using the mapping
data['lead_time_category_encoded'] = data['lead_time_category'].map(lead_time_category_mapping)
```

2

```
categorical_cols = ['type_of_meal_plan',
                    'room_type_reserved',
                    'market_segment_type',
                    'lead_time_category']
```

```
categorical_transformer = OneHotEncoder(handle_unknown='ignore')

# Combining preprocessing steps
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numerical_transformer, numerical_cols),
        ('cat', categorical_transformer, categorical_cols)
    ])
```

Yu Bixun (A0218248B)

However, there are some limitations of LLMs when it comes to performing descriptive analytics.

Limitations

1. Non-retention of context

- In a **worse case** scenario, the LLMs carry on the model training process using the original data even though it has performed the required train-test-split in the previous step
- This results in **data leakage** issue, and easily goes unnoticed. The model would still run, but it significantly **affect the performance**

```
from sklearn.model_selection import train_test_split

# Assuming your target variable is 'booking_status'
X = data.drop('booking_status', axis=1)  # Features
y = data['booking_status']  # Target

# Perform the train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, t

# Check the shapes of the resulting splits to confirm the o
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
# Assuming 'date_combined' is already in datetime format. If not, convert
data['date_combined'] = pd.to_datetime(data['date_combined'])

# 1. is_weekend_arrival
data['is_weekend_arrival'] = data['date_combined'].dt.dayofweek >= 5

# 2. quarter
data['quarter'] = data['date_combined'].dt.quarter

# 3. lead_time_category
data['lead_time_category'] = pd.cut(data['lead_time'],
                                    bins=[-1, 90, 180, float('inf')],
                                    labels=['Within 3 months', 'Within 6 
```

Yu Bixun (A0218248B)

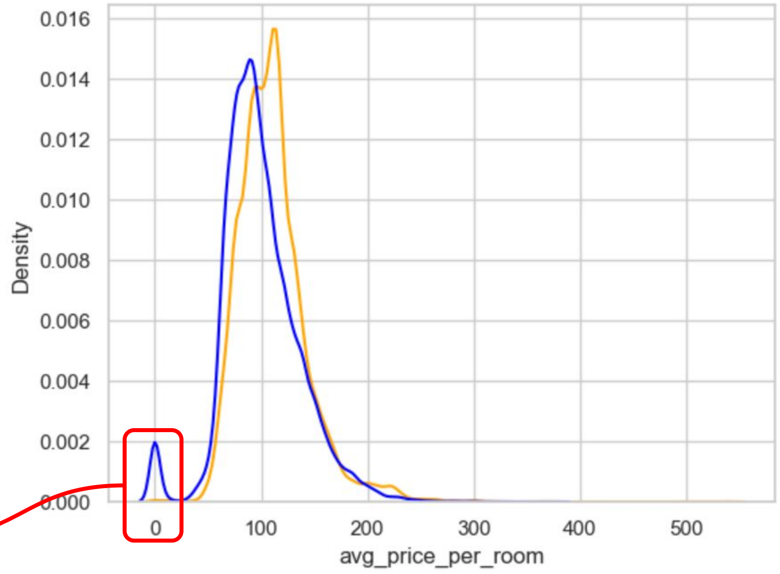
However, there are some limitations of LLMs when it comes to performing descriptive analytics.

Limitations

2. Overlooking unusual and strange data

- Unable to point out and rectify abnormal data or pattern ($0 average price per room)
- This may cause data quality issues if not resolved and impact performance of model
- Does not point out the implications of imbalance data

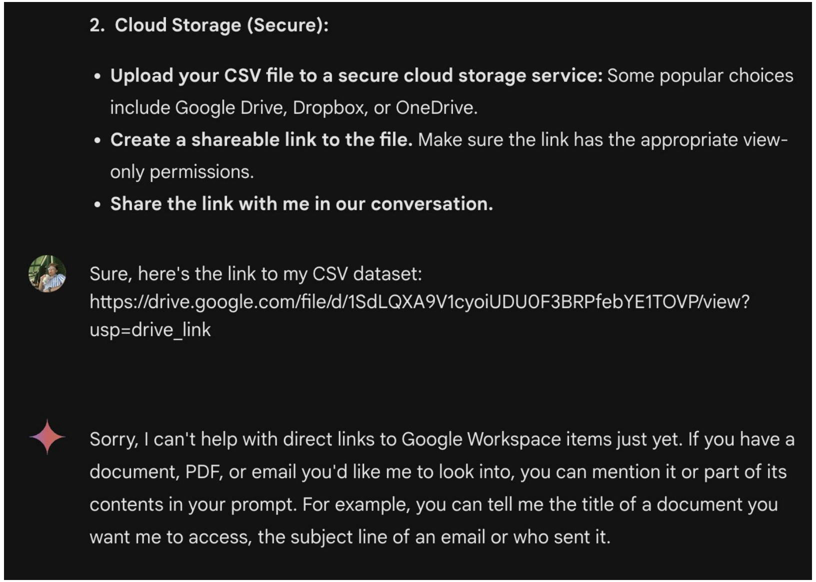
Existence of strange data points like $0 average price per room



Yu Bixun (A0218248B)

However, there are some limitations of LLMs when it comes to performing descriptive analytics.

Limitations



3. Overconfident in its response

- Gemini Advanced acknowledges that currently, user is **unable to directly upload dataset**. It then suggests users to upload the dataset on Google Drive, adjust permissions, and upload the link instead. However, this method fails to work
- Gemini Advanced is confident with its **interpretation of chart images (CV)**, but it produces a completely wrong response

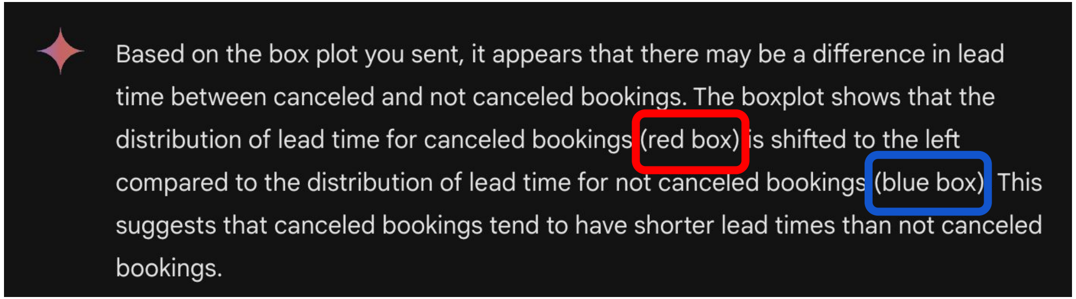
Yu Bixun (A0218248B)

However, there are some limitations of LLMs when it comes to performing descriptive analytics.

Limitations

3. Overconfident in its response

- *“Sure, let's look at the Box Plot that I generated to understand more about the lead time difference between canceled and not_canceled booking!* ***Can you derive any insights from this?****”*


Based on the box plot you sent, it appears that there may be a difference in lead time between canceled and not canceled bookings. The boxplot shows that the distribution of lead time for canceled bookings (red box) is shifted to the left compared to the distribution of lead time for not canceled bookings (blue box). This suggests that canceled bookings tend to have shorter lead times than not canceled bookings.

Yu Bixun (A0218248B)

However, there are some limitations of LLMs when it comes to performing descriptive analytics.

Limitations

4. Produced sub-optimal codes with lots of warnings

- While the LLMs produce codes that generally run, it is trained on **older documentation** with codes that are sometimes **deprecated**. This gives rise to **warnings**
- The LLMs are trained on random github repository. While they produce commonly seen way of writing code, they are **not of best practices** and could lead to **unnoticeable error** along the way *(i.e. not training with Pipeline)*

```
SettingWithCopyWarning:
                SettingWithCopyWarning:
SettingWithCopyWarning:
```

Sources

1. github.com/Aman-4-Real/CodeTemplates
2. stackoverflow.com/questions/69960522/w...
3. github.com/Hari31416/Portfolio

Sources? Sauce or Sus?

Who is Aman and Hari?

Do they write credible code?

What should I be looking out for?

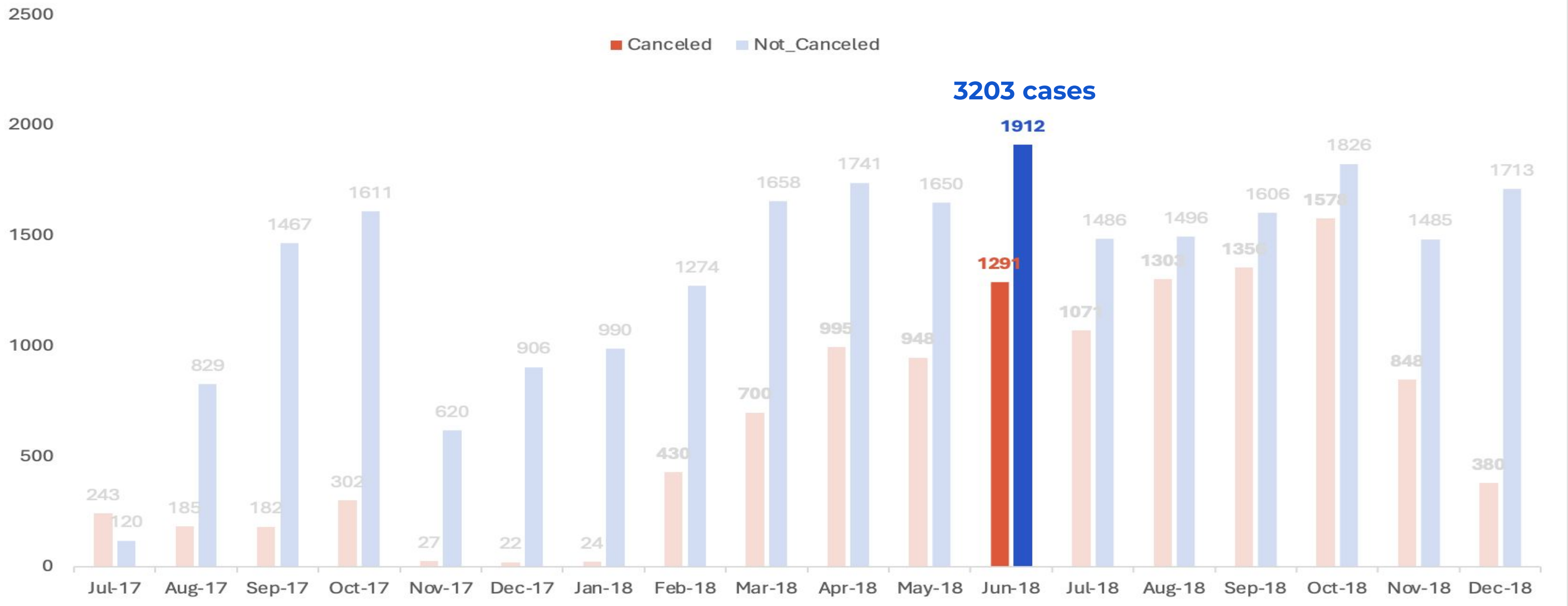
Yu Bixun (A0218248B)

Phase 6:
In-depth Analysis of Cost Strategy

Andreas Lukita (A0221743M)

How do the hotel management judiciously manage the trade-off associated with the overbooking strategy?

Distribution of Canceled vs Non-Canceled Bookings



Andreas Lukita (A0221743M)

Augmenting Prediction: Where Human Intuition Meets LLM Power

Final Model Performance

- ☑ **78.3%** *accuracy*
- ☑ **76.8%** *precision*
- ☑ **97.0%** *recall*
- ☑ **85.8%** *F1-score*
- ☑ **89.2%** *ROC-AUC score*

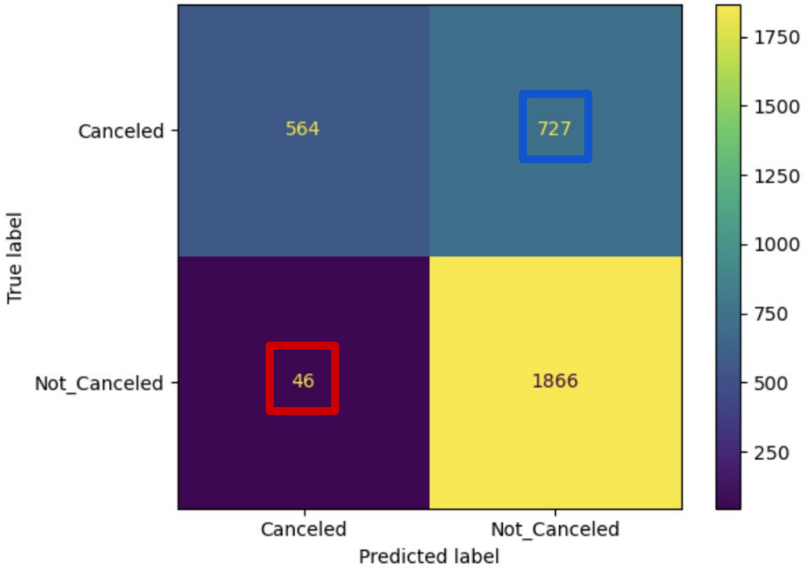
Key Insights

1. **Human-assisted** iteration in model building and hyperparameter tuning
2. Tune model to **minimise False Positive** (predict as cancel but in reality otherwise)
3. Adjustment of **probability decision threshold**

Andreas Lukita (A0221743M)

Understanding False Positive and False Negative

Confusion Matrix



Soft Voting Classifier Model

FN 727

FP 46 *(to minimize)*

***FN**: Our model predicts not canceled, but otherwise*
***FP**: Our model predicts canceled, but otherwise*

Andreas Lukita (A0221743M)

Managing the trade-off associated with the overbooking strategy

Cost of Predicting **False Cancellation (Overbooking)**

€(350) / booking

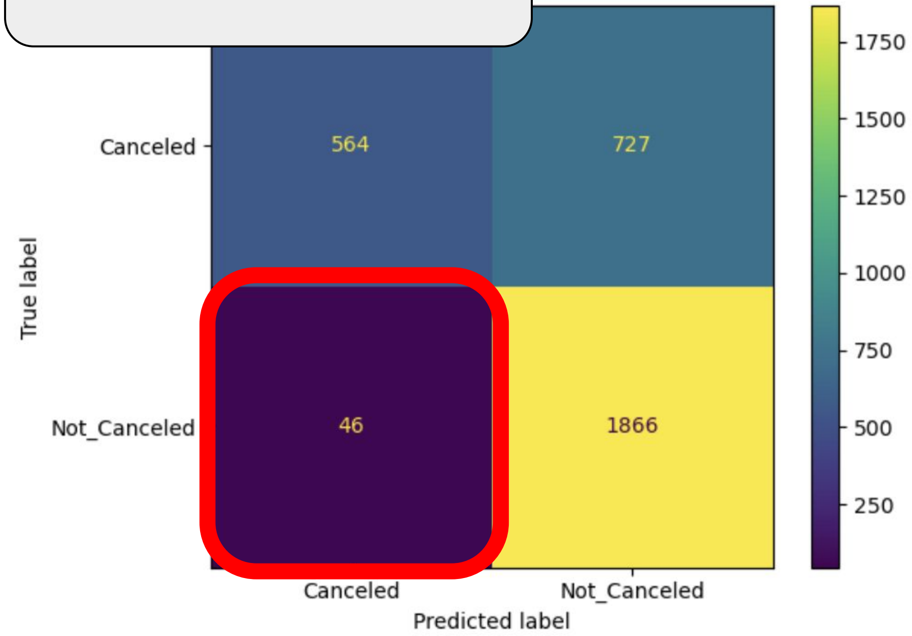
(€150 relocation costs + €200 reputation costs)

Cost of Predicting **False Non-Cancellation (Underbooking)**

€(100) / booking

(€100 no-show costs equivalent to average daily rate)

Aim: Optimise model to reduce False Cancellations!



Andreas Lukita (A0221743M)

AI Model Value Proposition

Total Costs **without** Predictive Model:

€129,100

Total Costs **with** Predictive Model:

€88,800

Total Cost Savings

€40,300 / month

Cost Saving

Our AI Model could potentially save you…

€483,600 Annually

*(**Assuming** monthly average number of bookings are cancellations are constant. Refer to Phase 7 for more detail)*

You're welcome!

Andreas Lukita (A0221743M)

Phase 7: Strategic Recommendation

Andreas Lukita (A0221743M)

What's next?

01

Clustering of Customer Segment

- Further explore if we can categorise customers into a **more specific group** and build a more specific model for this sub-groups to improve performance.

02

Perform Monte-Carlo Simulation

- **Convince the higher-ups** of the stability of the ML model by simulating the booking cancellations to obtain the average incremental cost savings per month from adopting this proposed solution.

03

Model Deployment

- **Test and deploy** the model.
- **Profit analysis** to compare the scenario without and with the AI model.
- **Maintaining** the performance of the model (MLOps).

Andreas Lukita (A0221743M)

End of Presentation