CXAPP - Space Optimization Challenge

Enhancing Workspace Utilization & Employee Experience through Data-Driven Insights

-

Manish Kumar Vuppugandla

1. Summary

The space optimization challenge revolves around efficiently managing and maximizing the utilization of meeting spaces within an organization. This summary will outline the key aspects of the challenge, including data analysis, assumptions, feature engineering, model selection, and deployment strategies.

Data Analysis:

The challenge begins with a thorough analysis of historical data related to room bookings and meetings. Visualizations and insights are extracted from the data to understand patterns and trends. Notable observations from the analysis include:

- Data Skewness: The data exhibits complex skewness patterns, with a significant concentration of room bookings in the year 2023, particularly from May to August.
- Meeting Durations: Over 95% of bookings have durations of 1-2 hours, with a skewed distribution towards longer durations. This distribution influences model training.
- Building Dominance: A small subset of buildings dominates the bookings, contributing to over 50% of total bookings. Building-specific modeling is needed.
- Amenities Influence: Specific amenities, such as 'Dual Monitors' and 'Height Adjustable Desks,' are popular and account for over 50% of bookings.
- Floor Significance: Floor selection plays a substantial role, with most bookings occurring on lower floors. Contrary to expectations, data shows a preference for lower floors.

Assumptions & Feature Engineering:

To address the space optimization challenge effectively, several assumptions and feature engineering steps are considered:

- IS_CANCELLED Column: 'True' in the "IS_CANCELLED" column indicates a canceled meeting, prompting further investigation to predict meeting cancellations.
- Duration Transformation: The 'duration' of meetings in minutes is transformed into 'counts,' where 1 count represents 30 minutes.
- Duration Limitation: Rows with 'Duration' counts exceeding 24 (equivalent to 12 hours) are restricted.

Model Selection and Deployment:

The choice of machine learning models and deployment strategies is influenced by the data analysis and key observations:

- Data Skewness: Traditional models like *Decision Trees* and *Random Forests* are chosen to handle skewed data.

- Meeting Durations: Models need to capture specific preferences, making decision trees and random forests suitable candidates.

- Building Dominance: Building-specific modeling approaches are essential due to a few dominant buildings.

- Amenities Influence: Feature selection is influenced by popular amenities.

Model Deployment in Production:

Efficient deployment of the chosen machine learning models is crucial for practical application

- Preprocessing: Categorical features, such as amenities, time of day, day of the week, month of the year, and building ID, require preprocessing before feeding into the model.

- User-Friendly Interface: An intuitive front-end interface is essential for user-friendliness and to facilitate preprocessing tasks. Developed a simple front-end where users can select from desired options from a chosen drop-down menu.

- Prediction Interpretation: Predictions correspond to grouping according to the counts, with each count representing a 30-minute time duration. Post-processing steps are needed to convert predictions into meaningful durations.

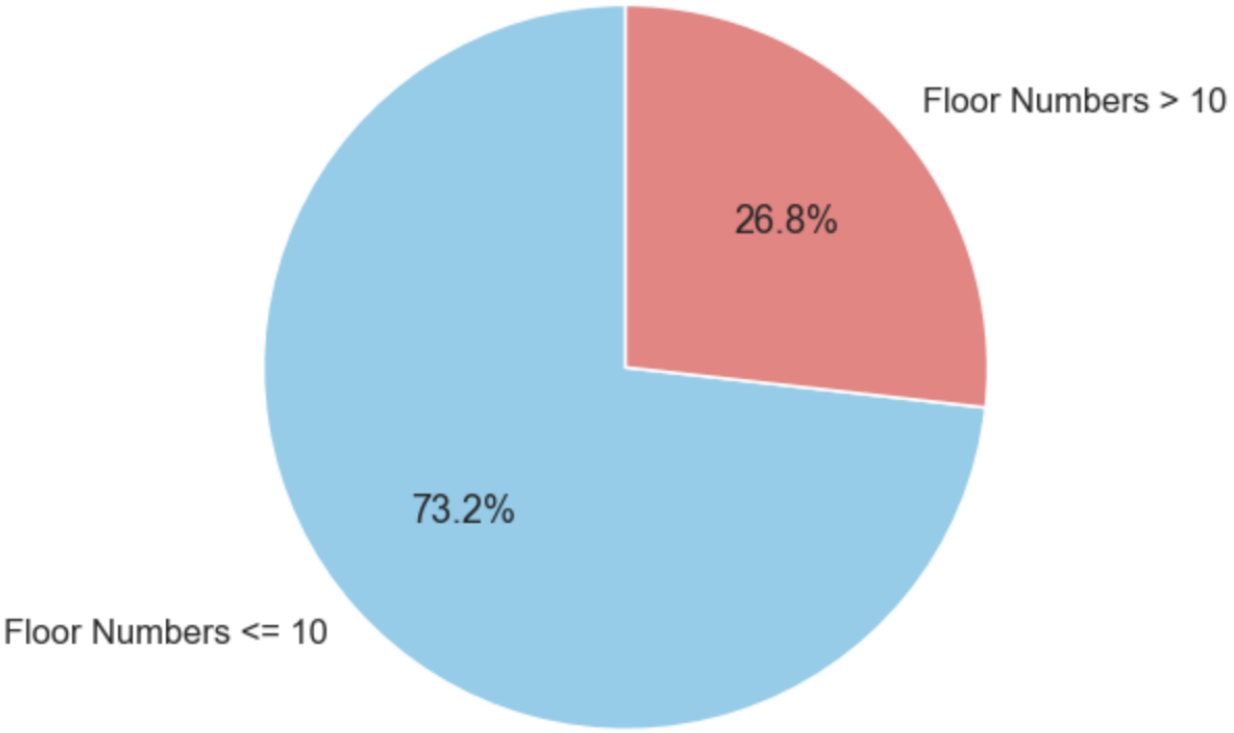
- Integration and Deployment: The model integration into the application involves API calls to handle preprocessing. The pre-trained model can be packaged with the app for deployment on client devices or cloud hosting, enabling website integration.

The space optimization challenge is centered on efficient booking space utilization within an organization. It encompasses data analysis, assumptions, feature engineering, model selection, and deployment strategies. Key observations about data skewness, meeting durations, building dominance, amenities influence, and floor significance guide the choice of models and deployment methods. Successful deployment of machine learning models will result in improved resource allocation, cost savings, and enhanced productivity for the organization.

2. Key Graphs & Figures

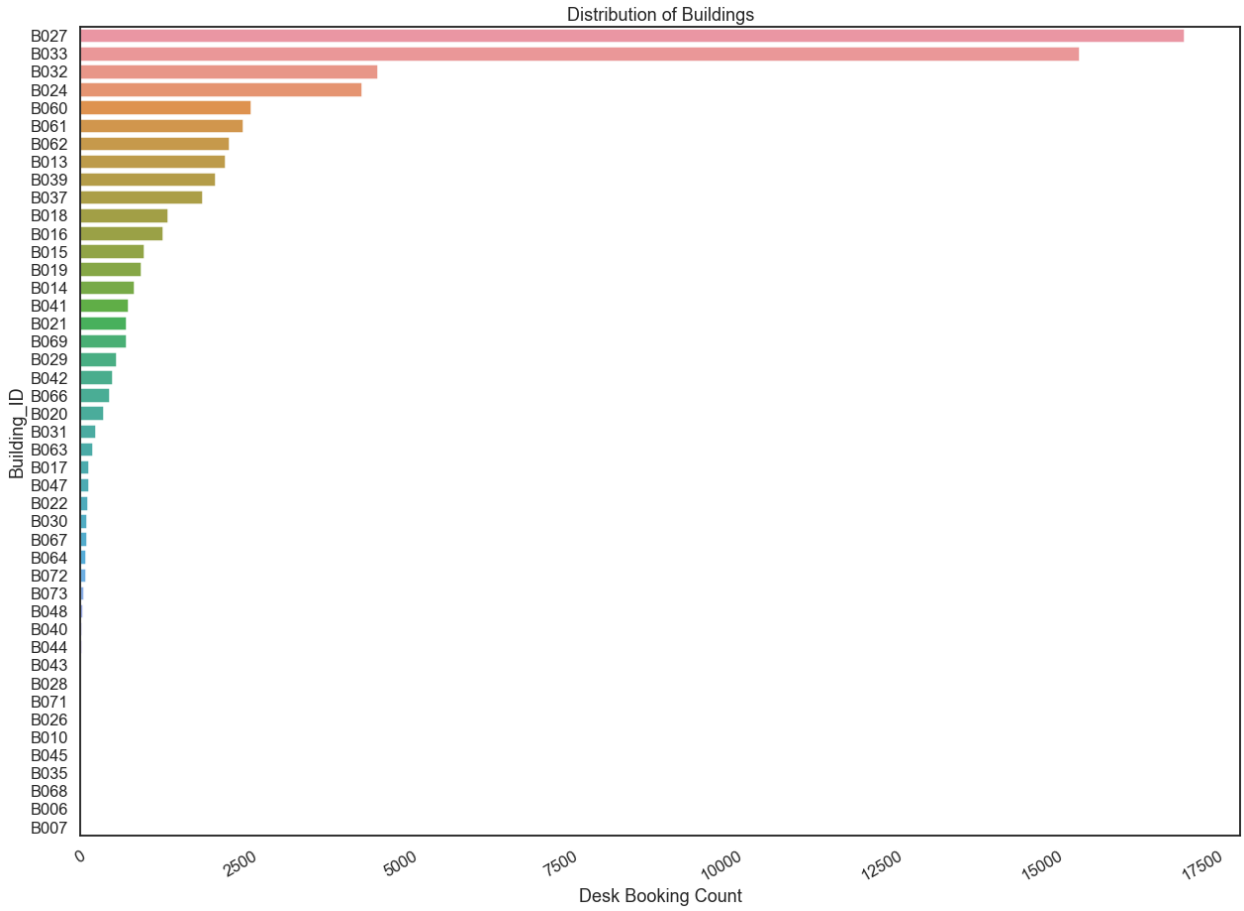
This section showcases important graphs and figures that offer visual representations of the data analysis, aimed at providing a clearer understanding of the data's patterns and trends.

Fig 2. a



- The pie chart visually represents the distribution of floor numbers in the dataset, with two main categories: "Floor Numbers <= 10" and "Floor Numbers > 10."
    - The chart indicates a noticeable correlation between the floor number and the number of scheduled meetings, suggesting that certain floors are more popular choices for meetings.
    - Interestingly, a significant majority of the meetings, approximately 73.2%, are concentrated on floors below the 10th floor.
    - Moreover, more than half of all meetings, exceeding 50%, are scheduled on floors below the 5th floor, indicating a strong preference for lower floors when booking meeting rooms.
    - Contrary to the common expectation that people prefer higher floors and better views, the data shows a notable preference for booking meeting rooms on lower floors.

Fig 2. b



- This horizontal bar plot illustrates the distribution of desk booking counts across different building IDs, with the buildings labeled on the y-axis and the corresponding booking counts on the x-axis.
- It provides a clear visual representation of the variation in booking activity among different buildings, with Building IDs as the categorical variable and Desk Booking Count as the quantitative measure.
    - The plot shows a consistent pattern in building popularity, with approximately 10 buildings encompassing nearly 80% of the total data points.
    - Specifically, only 5 of these buildings are responsible for approximately 50% of all the desk bookings made, highlighting their substantial contribution to the overall booking activity.

In conclusion, these key graphs and figures provide a comprehensive overview of the data analysis, including meeting distribution patterns, building popularity, location analysis, floor utilization, meeting room capacity analysis, and amenities trends.

These insights will inform decision-making and resource optimization strategies.

3. Organization

3.1. Preliminary Data Exploration and Insights (EDA)

Two historical datasets have been provided: room_booking_log and desk_booking_log. However, these datasets differ significantly in size. Desk booking data contains over 400,000 data points, while room booking data consists of only 70,000 data points, making the desk booking data approximately six times larger than the room booking data.

Due to this substantial discrepancy in data size, it is not advisable to combine these datasets for any analysis, whether it's preliminary data analysis or advanced machine learning modeling. Combining them could lead to misleading results, as patterns from one dataset may not accurately reflect real-world scenarios when applied to the other dataset.

Consequently, the data analysis was conducted separately for each of the provided datasets.

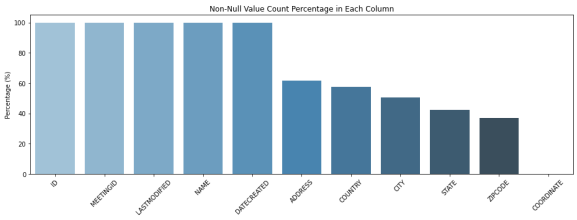
These are the data frames that are currently available for use:

1. Building_df - contains data related to buildings, likely including information about their attributes and characteristics.
2. Floor_df - holds data concerning building floors, potentially detailing floor layouts or configurations.
3. Room_df - contains information about individual rooms within the buildings, possibly including room specifications or features.
4. Desk_df - contains data about individual desks, which may include desk attributes or locations.
5. Rule_df - likely contains rules or regulations related to room or desk bookings.
6. Neighborhood_df - contains information about neighborhood characteristics or attributes, possibly relevant for location-based analysis.
7. EntitlementGroup_df - contains data related to entitlement groups, potentially of access or privileges.
8. RoomBookingLog_df - logs or records room booking activities, providing insights into booking history.
9. DeskBookingLog_df - logs or records desk booking activities, offering insights into desk reservation history.

A more in-depth examination of room booking trends:

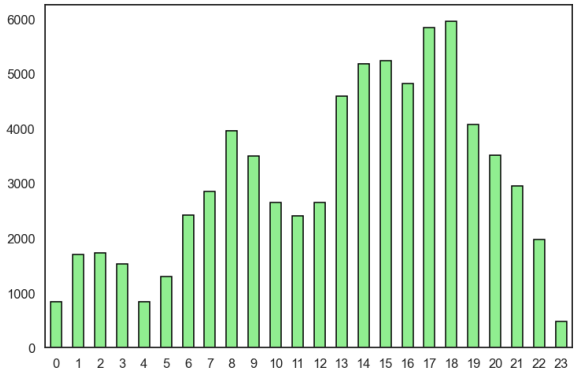
I. RoomBookingLog_df ->

Fig 3.1.I.a



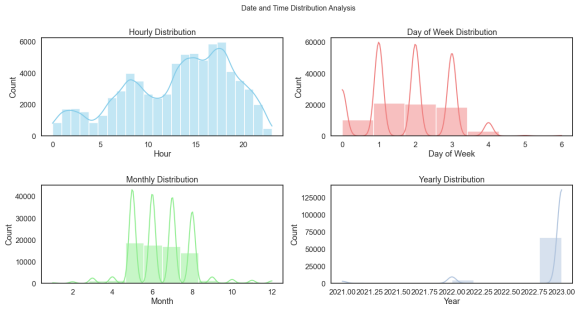
- This bar plot allows for a quick assessment of data completeness in the *"Building_df"* dataset, highlighting columns with a high percentage of non-null values and those with a significant amount of missing data, providing valuable insights into data quality and potential areas for data imputation or cleaning.
- It appears that the *"coordinate"* column is empty, and most of the different aspect columns related to *"location"* have more than half of their values missing or empty.

Fig 3.1.I.b: Room Bookings count vs. Start-Time of the day



- Point 1: On the x-axis of the plot, we have hours of the day, ranging from 0 (midnight) to 23 (11 PM). The y-axis represents the count of room bookings that started during each respective hour.
- Point 2: The plot displays a bar chart, where each bar corresponds to a specific hour, indicating the frequency of room bookings starting at that hour (24-hour format). It allows us to visually analyze the distribution of room booking start times throughout the day.
    - The majority of meetings are typically scheduled during daytime hours, specifically between 11 AM and 7 PM.

Fig 3.1.I.c



1. Hourly Distribution Plot:
    - Illustrates hourly room booking patterns.
    - Indicates peak bookings during daytime hours, especially in the late morning and early afternoon.
2. Day of Week Distribution Plot:
    - Shows room booking distribution across days of the week.
    - Highlights higher bookings on weekdays, particularly Monday to Thursday, with a notable decrease on weekends.
3. Monthly Distribution Plot:
    - Visualizes monthly room booking patterns.
    - Reveals variations in bookings throughout the year, with potential peaks in specific months related to seasons or events.
4. Yearly Distribution Plot:
    - Presents room booking distribution yearly.
    - Helps identify long-term trends or anomalies in room bookings over the past 3 years.

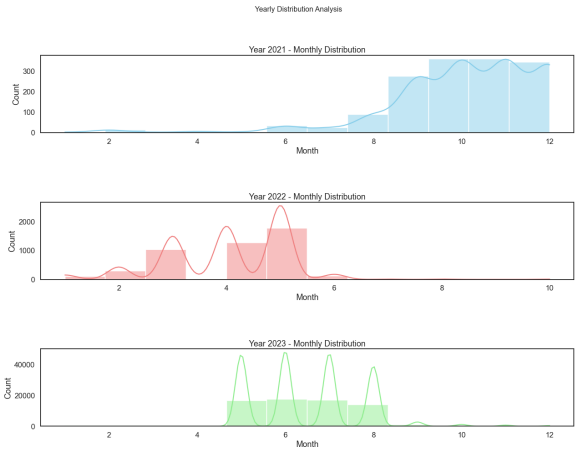
Observations:

- When analyzing the data grouped by the hour of the day, it becomes evident that the period from 12 p.m. to 8 p.m.

experiences the highest frequency of meetings.

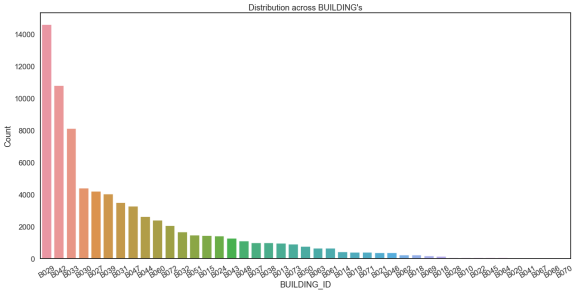
- Tuesday emerges as the most favored day of the week for meetings, closely followed by Wednesday and Thursday. Conversely, weekends exhibit the lowest meeting activity.
- In terms of monthly patterns, May stands out as the busiest month for meetings, with continued high activity from June through August.

Fig 3.1.I.d



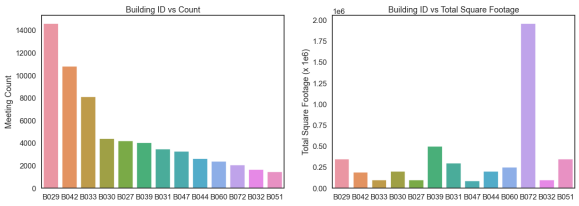
- It appears that the year 2023 contributes to nearly 90% of the total room booking data.

Fig 3.1.I.e



- Visualization of Building Distribution: The plot displays the distribution of room bookings across different buildings. It uses a count plot to show the number of bookings in each building, with buildings ordered by the count of bookings in descending order. The x-axis represents the BUILDING_ID, while the y-axis represents the count of bookings, providing a clear overview of how bookings are distributed among various buildings.
- Insight into Room Booking Patterns: By observing this plot, you can quickly identify which buildings have the highest and lowest booking frequencies. This information is valuable for understanding room booking patterns within the dataset and can help in making informed decisions or strategies related to room allocation or utilization across different buildings.
    - Notably, just five buildings account for 50% of the total meetings. This information can help in optimizing building resources.

Fig 3.1.I.f



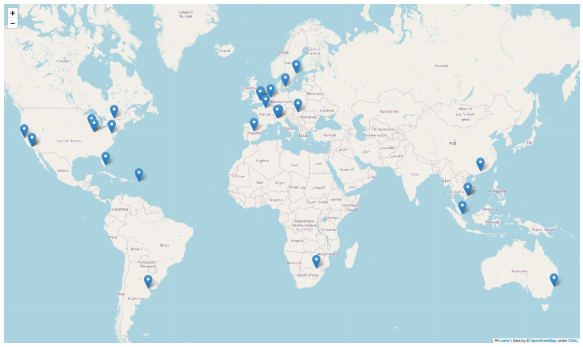
1. The first subplot illustrates the distribution of building IDs based on the count of meetings held in each building, showing variations in meeting frequency across different buildings.
2. The second subplot, presents the relationship between building IDs and the total square footage of the buildings, with each bar representing the square footage in millions, offering insights into the varying sizes of the buildings in the dataset.

Observations:

1. Thirteen buildings account for over 80% of the dataset, indicating that a small number of buildings are heavily used for meetings or bookings.
2. Interestingly, there doesn't appear to be a strong correlation between the popularity of a building, as indicated by the number of meetings scheduled, and the square footage of the building. In other words, larger buildings don't

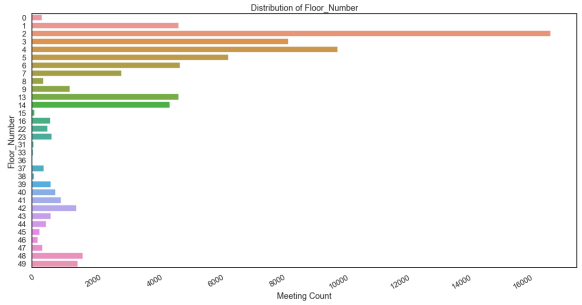
necessarily have more meetings scheduled, and vice versa.

Fig 3.1.I.g



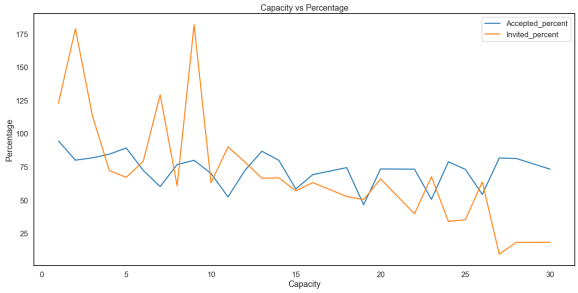
- When mapping the buildings to assess whether the location has any impact on booking distributions or patterns, there doesn't appear to be any discernible relationship at a glance.

Fig 3.1.I.h



- This horizontal bar plot visually represents the distribution of meeting counts across different floor numbers. It provides a clear view of which floors have the highest and lowest meeting activities.
- The plot's x-axis shows the meeting count, while the y-axis displays the floor numbers. It helps identify the most frequently used and less utilized floors for meetings, aiding in space utilization analysis.
    a. There is a noticeable relationship between the floor number and the frequency of scheduled meetings, indicating a pattern in meeting room preferences.
    b. The majority, approximately 73.2%, of meetings occur on floors below the 10th floor, suggesting a concentration of meeting activity on lower floors.
    c. Furthermore, over half, or more than 50%, of meetings are concentrated on floors below the 5th floor, highlighting a significant preference for lower-level meeting spaces.

Fig 3.1.I.i



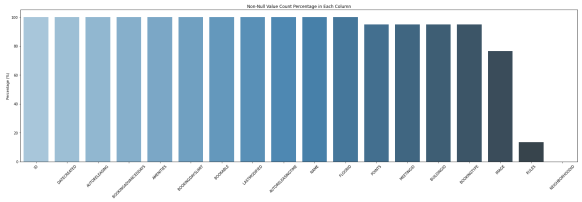
- This line plot illustrates the relationship between meeting room capacity (up to 30 people) and two important percentages: "Accepted_percent" and "Invited_percent." It allows us to observe how these percentages change with varying room sizes.
- The plot showcases the trend that as meeting room capacity increases, both the "Accepted_percent" and "Invited_percent" tend to decrease, indicating that smaller rooms are more likely to have higher acceptance rates and a larger proportion of invited attendees attending the meetings.
    - Meeting rooms with a capacity of fewer than 30 people are the most popular, and they tend to have an average occupancy rate of approximately 75%, indicating efficient utilization of smaller meeting spaces.
    - Interestingly, there doesn't appear to be a strong correlation between the number of invited attendees and the number of attendees who

participate in the meeting, irrespective of the meeting room's capacity. This suggests that factors other than room size play a significant role in determining attendee turnout.

A more in-depth examination of desk booking trends:

II. DeskBookingLog_df ->

Fig 3.1.II.a

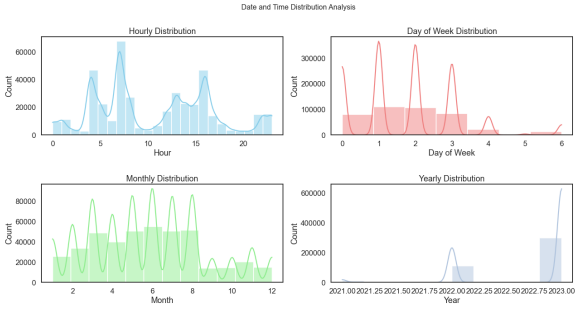


- This bar plot visually represents the percentage of non-null values in each column of the dataset, with columns sorted in descending order based on their non-null value percentages. It offers insights into data completeness across different attributes of the Desk.
- The plot's y-axis shows the percentage of non-null values, while the x-axis displays the column names. It helps identify which columns have a higher proportion of missing data and which ones are more complete, aiding in data quality assessment.
    - The columns labeled "Rules" and "NeighborhoodId" appear to be empty, indicating a lack of data or information in these two attributes.
    - These columns may not provide any meaningful insights or analysis due to their absence of data.

'AUTORELEASING' - column:

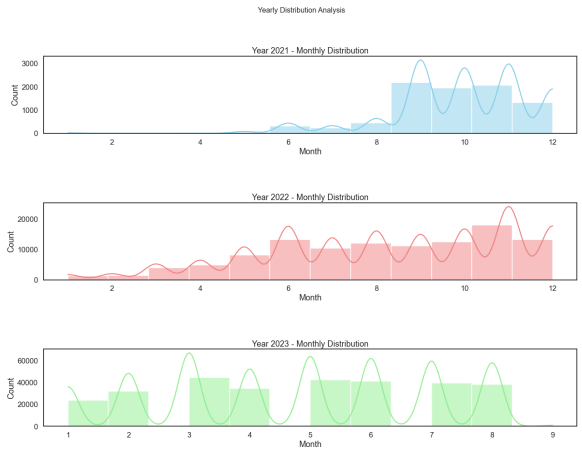
- A significant majority, approximately 99.6%, of the bookings are set to not auto-release the desks, indicating a strong preference for manual release.
- Consequently, this preference for manual release results in a skewed distribution of booking durations, with a tendency towards longer booking periods, as there is a lower likelihood of automatic desk release.

Fig 3.1.II.b



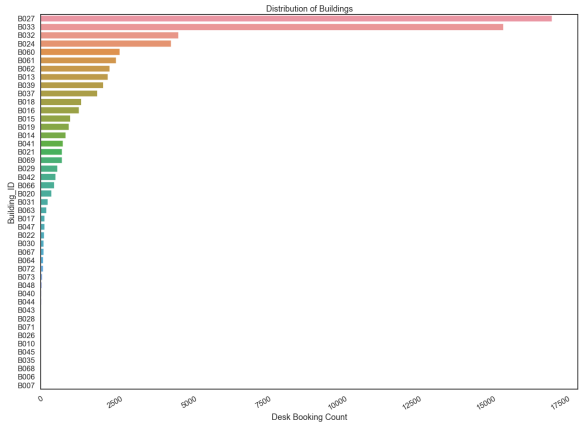
1. Hourly Distribution Plot:
    - Illustrates hourly room booking patterns.
    - Indicates peak bookings during daytime hours, especially in the late morning and early afternoon.
2. Day of Week Distribution Plot:
    - Shows room booking distribution across days of the week.
    - Highlights higher bookings on weekdays, particularly Monday to Thursday, with a notable decrease on weekends.
3. Monthly Distribution Plot:
    - Visualizes monthly room booking patterns.
    - Reveals variations in bookings throughout the year, with potential peaks in specific months related to seasons or events.
4. Yearly Distribution Plot:
    - Presents room booking distribution yearly.
    - Helps identify long-term trends or anomalies in room bookings over the past 3 years.

Fig 3.1.II.c



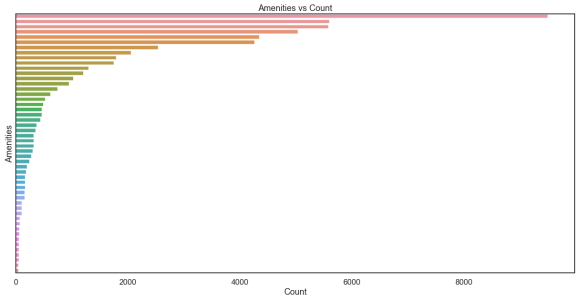
- This set of three subplots presents the yearly distribution analysis of meeting activities for the years 2021, 2022, and 2023. Each subplot depicts the monthly distribution of meetings, with different colors representing each year. The histograms with kernel density estimates (KDE) reveal patterns and variations in meeting frequency over the months.
- The plots allow us to discern how meeting patterns change across the three years, identifying trends and seasonal fluctuations in meeting scheduling for each year.
    - It appears that the year 2023 contributes to nearly 90% of the total room booking data.

Fig 3.1.II.d



- This horizontal bar plot provides insights into the distribution of desk booking activities across different buildings. It visually represents which buildings have the highest and lowest desk booking counts, helping to identify patterns in workspace utilization.
- The x-axis displays the desk booking count, while the y-axis shows the building IDs. It offers a clear overview of the variations in desk booking demand among different buildings, which can be valuable for workspace management and resource allocation decisions.
    - The distribution of desk bookings in buildings follows a similar pattern to the popularity of buildings observed in the previous analysis, reaffirming the consistent demand for workspace in specific buildings.
    - Remarkably, only five buildings account for approximately 50% of the total desk bookings, indicating a concentration of booking activity in a select few buildings and potentially suggesting areas for focused resource allocation or optimization.

Fig 3.1.II.e



- This bar plot displays the top 50 amenity groups by their count, providing a clear visual representation of the most frequently occurring amenities in the dataset.
- By showcasing the number of times each amenity group appears, this plot helps identify the most popular amenities and their relative prevalence in the data, aiding in understanding the preferences of the users or visitors.

A more detailed examination of AMENITIES:

Fig 3.1.a

```python
#Sorting the dictionary by values
Amenities = dict(sorted(Amenities.items(), key=lambda item: item[1], reverse=True))
total_count = sum(Amenities.values())
```

Python

```python
# Initialize a new dictionary to store frequently appearing amenities
frequently_appearing_amenities = {}
current_count = 0

# Iterate through the sorted amenities and add them to the new dictionary
for amenity in Amenities:
    frequently_appearing_amenities[amenity] = Amenities[amenity]
    current_count += Amenities[amenity]
    # Stop when 80% of the counts are reached
    if current_count / total_count >= 0.8:
        break

# Calculate the number of frequently appearing amenities, their total count
num_frequently_appearing_amenities = len(frequently_appearing_amenities)
total_frequently_appearing_amenities_count = sum(frequently_appearing_amenities.values())

print('Number of frequently appearing amenities:', num_frequently_appearing_amenities)
print('Percentage of frequently appearing amenities:', round(total_frequently_appearing_amenities_count / total_count * 100, 2), '%')
```

Python

- The code sorts a dictionary of amenities by their counts in descending order and extracts the top 80% of frequently appearing amenities, resulting in 12 amenities.
- These frequently appearing amenities account for 80.42% of the total counts, suggesting that a small set of amenities is heavily utilized.
- The top 7 amenity groups alone represent over 50% of all desk bookings, highlighting their significant popularity.

Fig 3.1.b

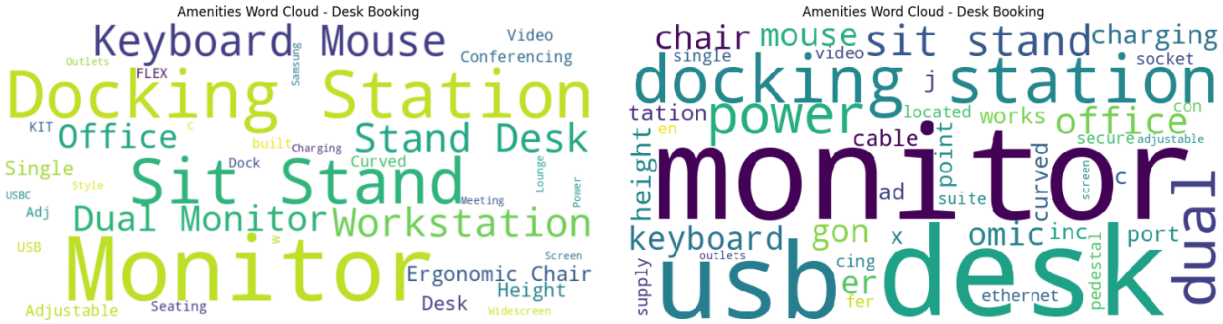
```python
def clean_key(key):
    # Remove square brackets, newline characters, and double quotes
    cleaned_key = key.replace('[', '').replace(']', '').replace('\n', '').replace('"', '').strip()
    # Split the cleaned key into individual words
    words = cleaned_key.split(', ')
    return words
```

Python

- The "clean_key" function removes square brackets, newlines, and double quotes from amenity names, and the cleaned amenities are stored in a new dictionary with their respective counts.
- The cleaned amenity names include items like "Dual Monitors" and "Height-Adjustable Desk," indicating common preferences among users.

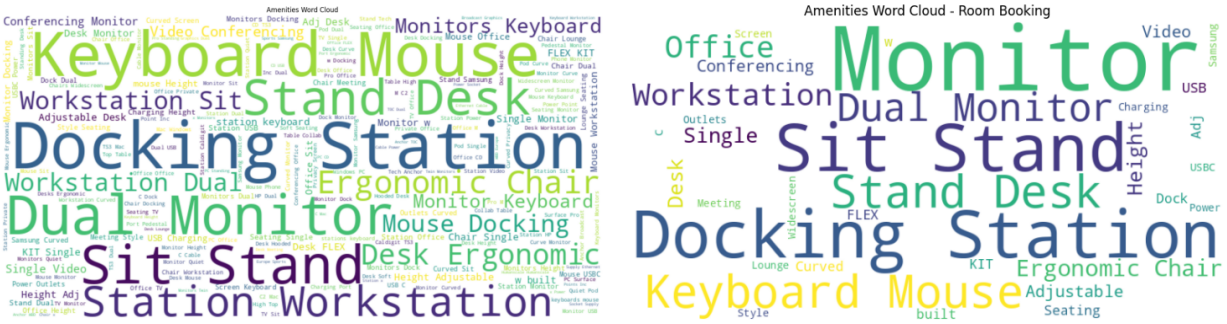
- A word cloud has been created using the most frequently appearing amenities in desk booking data, where the size of each amenity word in the cloud corresponds to its frequency of appearance.

Fig 3.1.c



- A word cloud has been created using the most frequently appearing amenities in room booking data, where the size of each amenity word in the cloud corresponds to its frequency of appearance.

Fig 3.1.d



- This analysis shows that popular amenities are consistent across both desk and room booking data, suggesting consistent preferences among users for certain amenities.

3.2. Assumptions

The "True" value in the "IS_CANCELLED" column, accounting for approximately *82.46%* of the total entries, suggests that a significant majority of the meetings were indeed canceled. However, it raises a crucial question: Should this "True" value indicate meeting cancellations, further prompting the need for investigation to predict meeting attendance? At present, we are making this assumption, but it merits further examination. i.e.,

Is the "True" value in the "IS_CANCELLED" column indicative of a canceled meeting?

- If it does, it warrants further investigation to predict the probability of a booked meeting being canceled.
- However, for the time being, we are assuming the opposite until further clarification.

3.3. Data Preprocessing

- The 'START_TM' and 'END_TM' columns in the dataset are converted from their original format to the datetime format, making them suitable for time-based analysis.
- A new data frame, 'data_cleaned,' is created by filtering the original data to include only records where the 'START_TM' year falls within the range of 2021, 2022, and 2023. This step ensures that only relevant data up to the year 2023 is considered for analysis and demand prediction.
- Notably, the data from the year 2024 and beyond is excluded as it is considered future data and is not relevant to the current analysis.
- This approach is chosen because the majority of the data, approximately 90%, are concentrated in the year 2023. Therefore, for the current analysis, only the data from the year 2023 is being utilized.

Handling Outliers by Including Data up to the 95th Percentile:

- To address outliers in the dataset, a decision has been made to limit the data considered for analysis up to the 95th percentile.
- This means that records beyond the 95th percentile, which are typically considered extreme values or outliers, will not be included in the analysis.
- By adopting this approach, the analysis focuses on the majority of the data while effectively excluding the most extreme data points, which can help improve the robustness and reliability of the analysis results.

Feature Engineering:

- A new feature called 'duration' is created to represent the duration of meetings in minutes.
- Additionally, the 'duration' column is further processed to convert it into 'counts,' where each count corresponds to a 30-minute interval.

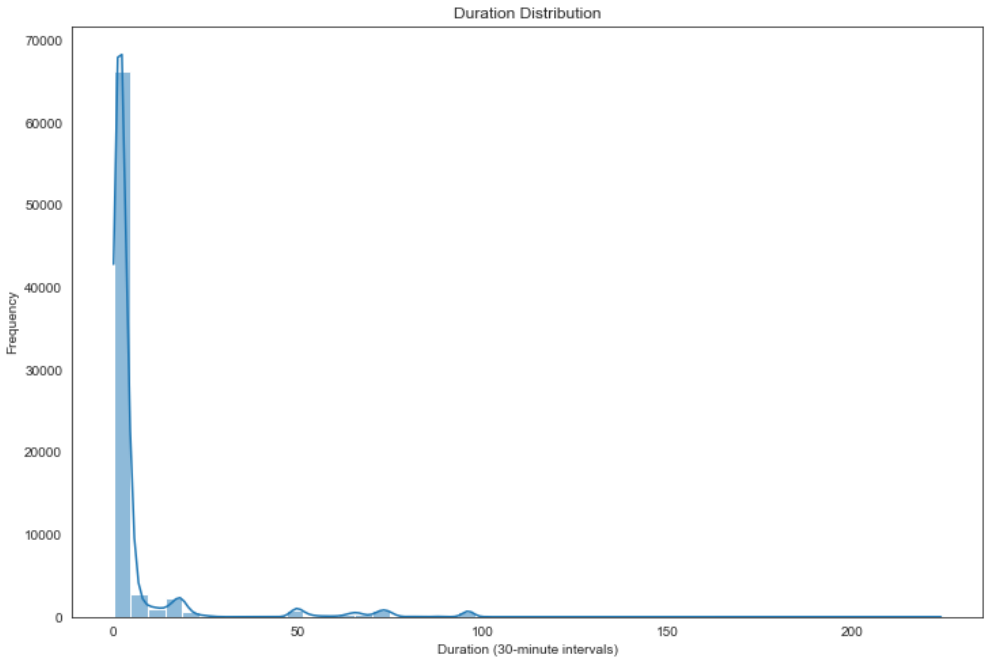
Fig 3.3.a

```python
# Define a function to calculate duration
def calculate_duration(row):
    # If END_TM is on the next day, set it to midnight
    if row['END_TM'].date() > row['START_TM'].date():
        row['END_TM'] = row['END_TM'].replace(hour=23, minute=59, second=59)
    # Calculate duration in 30-minute intervals
    duration_seconds = (row['END_TM'] - row['START_TM']).total_seconds()
    return int(np.ceil(duration_seconds / (30 * 60)))
```

Python

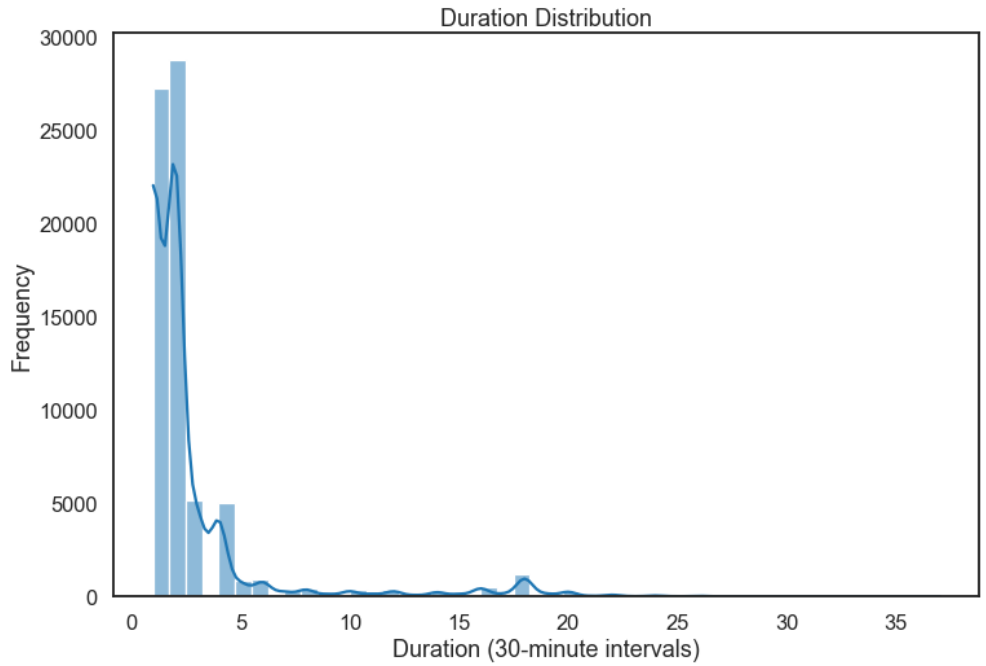
- A function, 'calculate_duration' is defined to perform this calculation:
    - If the 'END_TM' of a meeting extends to the next day, it is adjusted to midnight.
    - The duration is then calculated by finding the difference in seconds between 'START_TM' and 'END_TM,' and this duration is divided by 30 minutes (1800 seconds) to determine the 'counts'.

Fig 3.3.b



- The histogram plot illustrates the distribution of booking durations, with a clear concentration of bookings lasting between 1 and 24 hours (2 to 48 30-minute intervals).

Fig 3.3.c



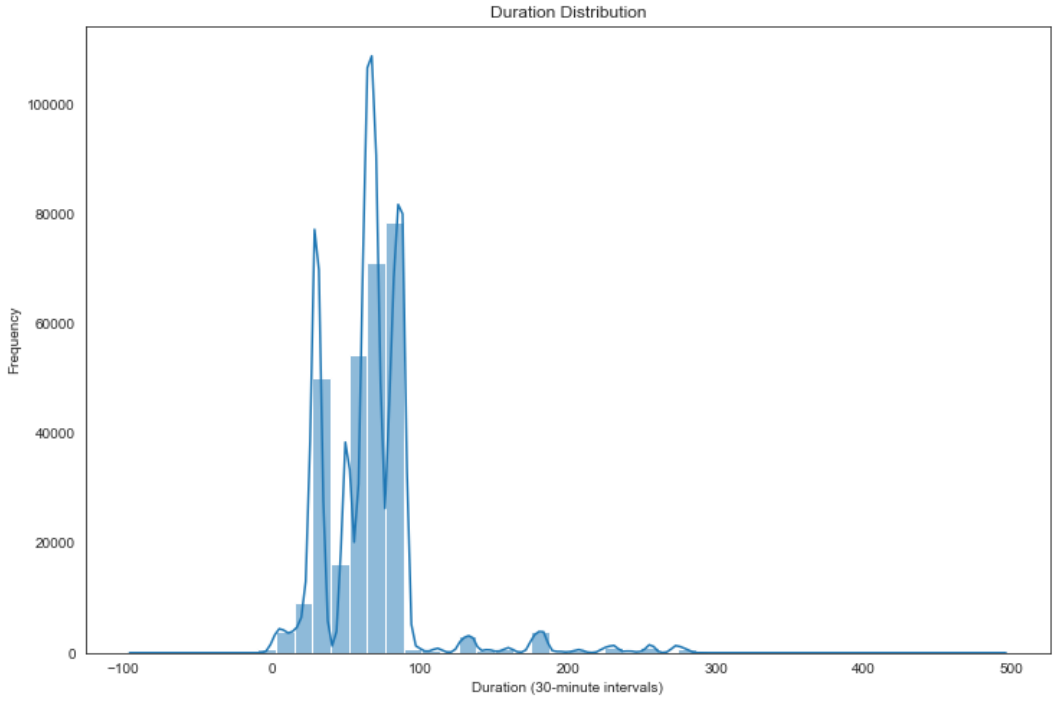
- The data has been cleaned to remove outliers and focus on valid durations, ensuring that only bookings within this 1 to 24-hour range are considered for further analysis.
    a. The majority of meetings are scheduled for 1 hour. (2 counts X 30 min)
    b. The most favored meeting durations fall within the 1-2 hour and 2-3 hour ranges, indicating that these time slots are highly sought after by users.

- The resulting histogram provides a more focused view of meeting duration distribution, allowing for a better understanding of the typical meeting lengths while effectively handling and mitigating the influence of outliers in the analysis.

The data preprocessing steps, including datetime conversion and duration calculation, have been consistently applied to both room booking and desk booking scenarios.

- The constraint to consider data only up to the year 2023 has been enforced to maintain relevance and exclude future data.

Fig 3.3.d

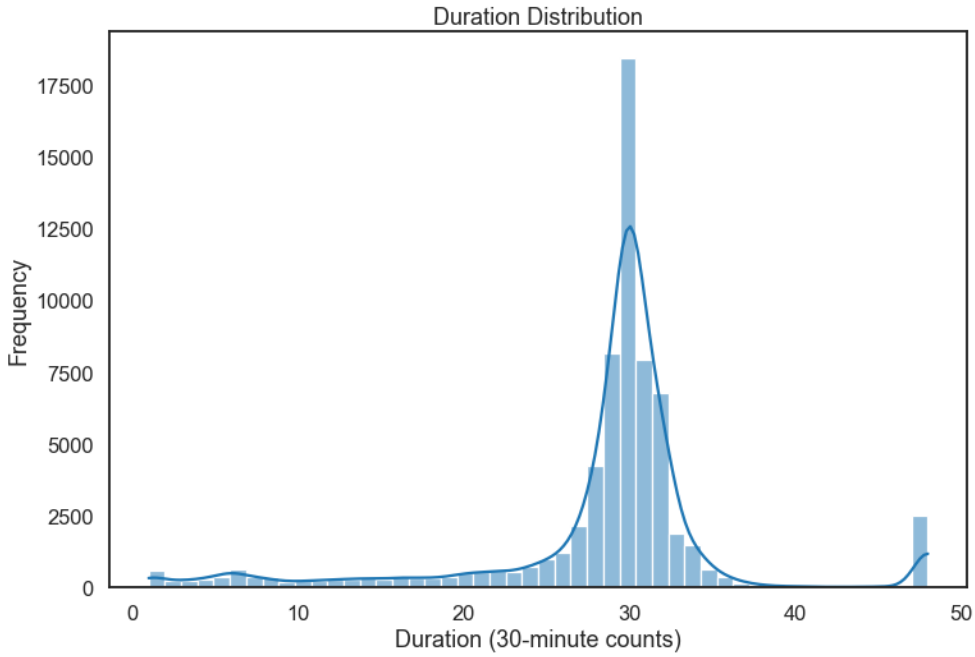


- Outliers in the form of meeting durations exceeding 24 hours (48 intervals of 30 minutes each) have been removed to ensure that extreme values do not skew the analysis for both room and desk bookings. This approach ensures data consistency and reliability across scenarios.

Observations:

1. The majority of bookings are scheduled for 15 hours, equivalent to 30 counts in 30-minute intervals.
2. This observation supports our hypothesis that most desks may not have an automatic release mechanism, leading to longer booking durations. Consequently, the distribution of booking durations is skewed towards the right, with a preference for extended booking periods.

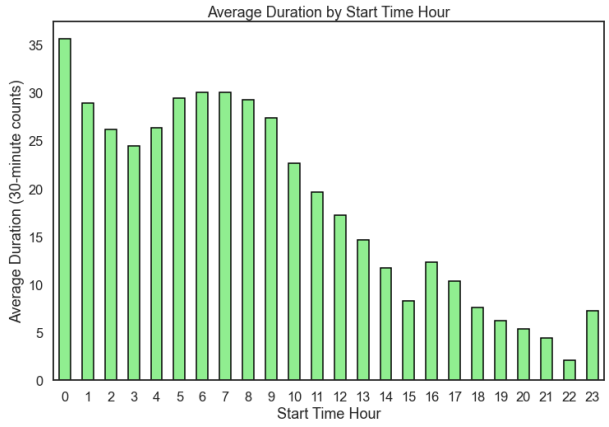
Fig 3.3.e



The following bar plot displays the average booking duration, measured in 30-minute counts, for each hour of the day when bookings start. It offers a clear visualization of how the mean duration varies throughout the day, revealing potential patterns in booking behavior based on the starting hour.

- The plot highlights any notable trends in meeting or desk usage duration, allowing for insights into preferred booking times and the average length of bookings at different hours.

Fig 3.3.f



- This bar plot depicts the average duration of desk bookings based on their start times, organized into hourly intervals.
- It reveals that desk bookings tend to peak early in the morning, around 5-6 AM, and decrease as the day progresses. This pattern suggests a preference for longer desk usage in the morning hours.

3.4. Feature Engineering

The following transformations have been applied to extract time-related features from the 'START_TM' column:

Fig 3.4.a

```python
data_cleaned['time_of_day'] = data_cleaned['START_TM'].dt.hour
data_cleaned['day_of_week'] = data_cleaned['START_TM'].dt.dayofweek
data_cleaned['month_of_year'] = data_cleaned['START_TM'].dt.month
```

- The 'time_of_day' column is created, which captures the hour of the day when the booking or meeting starts. It's obtained by extracting the hour component from the 'START_TM' column.
- The 'day_of_week' column is generated to represent the day of the week when the booking or meeting takes place. It is derived by extracting the day of the week (0 for Monday, 1 for Tuesday, and so on) from the 'START_TM' column.
- Similarly, the 'month_of_year' column is created to capture the month in which the booking or meeting occurs. It is obtained by extracting the month component from the 'START_TM' column.

These transformations provide additional time-related information that can be useful for analyzing and understanding booking patterns and trends.

The process involved in calculating and filtering building IDs that account for 90% of the value counts:

Fig 3.4.b

```python
# Calculate the BuildingID counts
building_counts = data['BUILDINGID'].value_counts().sort_values(ascending=False)

# Sort the counts in descending order
sorted_building_counts = building_counts.sort_values(ascending=False)

# Calculate the cumulative sum of the counts
cumulative_counts = sorted_building_counts.cumsum()

# Find the threshold for 90% of the total counts
total_counts = cumulative_counts.iloc[-1]
threshold = 0.8 * total_counts

# Filter the building IDs that account for 90% of the counts
selected_building_ids = cumulative_counts[cumulative_counts <= threshold].index

print("Building IDs accounting for 90% of the value counts:", len(selected_building_ids))
```

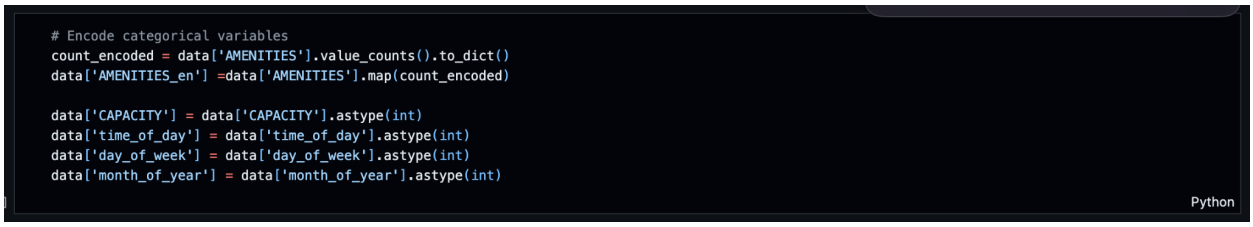
```
Building IDs accounting for 90% of the value counts: 11
```

1. Building counts are calculated, indicating how many times each building ID appears in the dataset.
2. The building counts are sorted in descending order, creating 'sorted_building_counts' to identify the most frequent building IDs.
3. Cumulative counts are computed by taking the cumulative sum of 'sorted_building_counts,' showing the cumulative contribution of each building ID to the total counts.

4. A threshold is set at 80% of the total counts ('threshold'), representing the target cumulative contribution.
5. Building IDs that account for 90% of the total counts are filtered and stored in 'selected_building_ids.'
6. The final step prints the number of selected building IDs that collectively make up 90% of the total value counts in the dataset.

Encoding Categorical Variables:

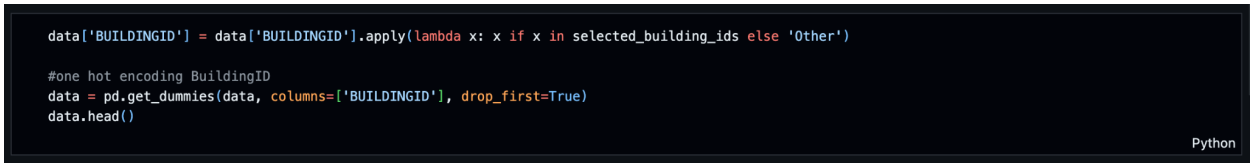
Fig 3.4.c



- A count-based encoding scheme is applied to the 'AMENITIES' column, where the frequency of each amenity combination is counted and stored in a dictionary named 'count_encoded.' This encoding assigns a numerical value to each unique amenity combination based on its frequency in the dataset, creating a new 'AMENITIES_en' column with these numerical values.
- The 'CAPACITY' column is converted to integer data type, ensuring that the capacity values are treated as whole numbers for analysis.
- 'time_of_day,' 'day_of_week,' and 'month_of_year' columns are also converted to integer data types, making these categorical features compatible with numerical analysis methods. This transformation allows for the inclusion of time-related and day-related information as numerical variables in the dataset.

One-Hot encoding:

Fig 3.4.d



- A lambda function is applied to the 'BUILDINGID' column, which replaces building IDs not included in the 'selected_building_ids' list with 'Other.' This consolidation reduces the number of unique building IDs.
- One-hot encoding is performed on the 'BUILDINGID' column, generating binary columns for each building ID. The 'drop_first=True' parameter is used to drop the first binary column to prevent multicollinearity issues.
- As a result, the 'BUILDINGID' column has been transformed into a set of binary columns, each representing a specific building, while the 'Other' category accounts for building IDs not included in the 'selected_building_ids' list.

More Categorical variables are encoded using a count-based approach for the 'AMENITIES' column in the dataset. The process involves the following:

- A dictionary named 'count_encoded' is created to store the counts of each unique amenity combination within the 'AMENITIES' column.
- The 'AMENITIES' column is mapped to its corresponding counts from the 'count_encoded' dictionary, resulting in a new column labeled 'AMENITIES_en.'
- This encoding scheme assigns numerical values to each unique amenity combination based on their frequency in the dataset, allowing for the inclusion of this categorical information in numerical analysis and modeling.

3.5. Use Case Selection and Feature Selection

Use Case Selection and Feature Selection:

As an observer, it was noticed that the typical duration for each desk booking is approximately 15 hours. However, upon closer examination, it was discovered that the 'auto releasing' column, related to desk properties, indicated that desks are seldom auto-released if no user action is taken.

- This resulted in 99% of desk booking data having a 'false' value for auto-releasing. This potential skew led to concerns about data accuracy, as there was limited information about the actual usage of these desk bookings.
- To address this, the decision was made to focus only on predicting demand for room bookings, as the desk booking data might not be suitable for meaningful analysis due to potential inflation.

Use Case Selection:

- The chosen use case is to predict the duration or demand for room bookings.

Feature Selection:

- Feature selection was based on observed patterns and clear boundaries in the data, resulting in the selection of specific features:

Fig 3.5.a

```
data_cleaned.drop(columns=['START_TM', 'BUILDING_ID', 'FLOOR_ID', 'MEETING_ID'], inplace=True)

data = pd.merge(data_cleaned, Room_df, left_on='ROOM_ID', right_on='ID', how='left')
data.drop(columns=['ID_y', 'DATECREATED', 'MEETINGID', 'NAME', 'FLOORID', 'POINTS', 'BOOKABLE'], inplace=True)
data.rename(columns={'ID_x': 'ID'}, inplace=True)

data.columns
```

Python

```
Index(['ID', 'ROOM_ID', 'Duration', 'time_of_day', 'day_of_week',
       'month_of_year', 'BUILDINGID', 'CAPACITY', 'LASTMODIFIED', 'AMENITIES'],
      dtype='object')
```

Fig 3.5.b

- ['AMENITIES_en', 'time_of_day', 'day_of_week', 'month_of_year', 'CAPACITY', and various 'BUILDINGID' binary columns representing specific buildings, with 'BUILDINGID_Other' for the remaining buildings.]

These features are expected to be relevant and influential in predicting room booking duration or demand.

3.6. Model Selection and Model Training

Key Observations and Model Selection

- Data Skewness: The dataset exhibited complex skewness patterns, with a significant concentration of room booking data in the year 2023, particularly from May to August. This skewness had to be considered during modeling.
- Meeting Durations: A large majority of bookings had durations of 1-2 hours, but there was a long right tail in the duration distributions. This skewed distribution of meeting durations posed challenges for model training and prediction.
- Building Dominance: A small subset of buildings dominated the dataset, with just five buildings accounting for over 50% of total bookings. Building-specific modeling approaches were deemed necessary.
- Amenities Influence: Certain amenities, such as 'Dual Monitors,' 'Height Adjustable Desks,' 'Docking Stations,' and 'Keyboard and mouse,' collectively contributed to over 50% of bookings. Understanding the popularity of specific amenities was crucial for feature selection.
- Floor Significance: Floor selection played a substantial role, with floors 1 to 5 combining to account for more than 50% of bookings. Contrary to expectations, roughly 75% of bookings occurred on floors numbered less than 10.

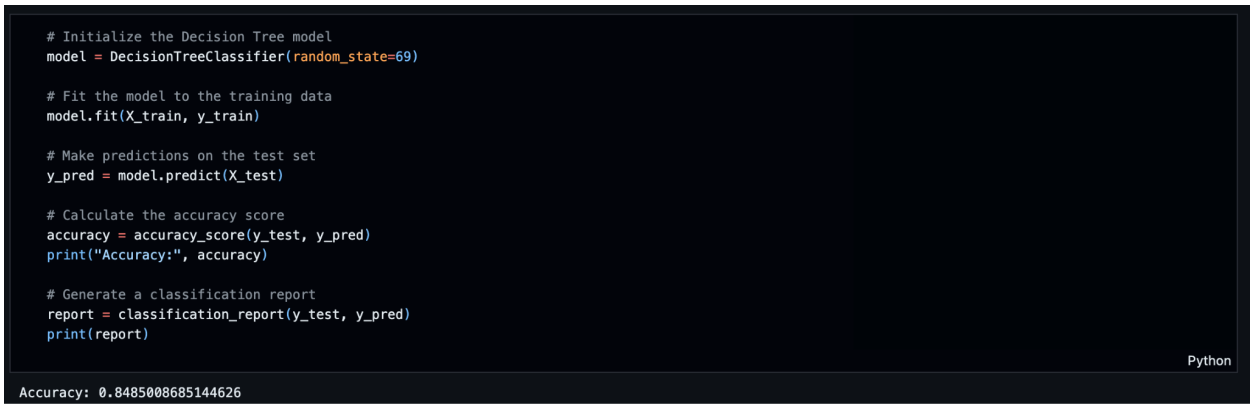
Given these observations and the clear boundaries in feature popularity, traditional machine learning models, specifically decision trees and random forests, were chosen as suitable candidates. These models could handle the skewed data and capture specific preferences and patterns in the dataset.

Model Selection:

1. Decision Tree Model:

- DecisionTreeClassifier with a random state of 69 was initialized.
- The model was fitted to the training data.
- Predictions were made on the test set.
- Accuracy was calculated and reported.
- A classification report was generated to evaluate model performance.

Fig 3.6.a



2. Random Forest Model:
   - RandomForestClassifier with 100 estimators, a random state of 69, and entropy as the criterion was initialized.
   - The model was fitted to the training data.
   - Predictions were made on the test set.
   - Accuracy was calculated and reported.
   - A classification report was generated to evaluate model performance.

Fig 3.6.b

These models were chosen based on their ability to handle the data characteristics observed in the dataset, such as skewness, building dominance, amenities influence, and floor significance. They were trained and evaluated to determine their effectiveness in predicting room booking durations or demand.

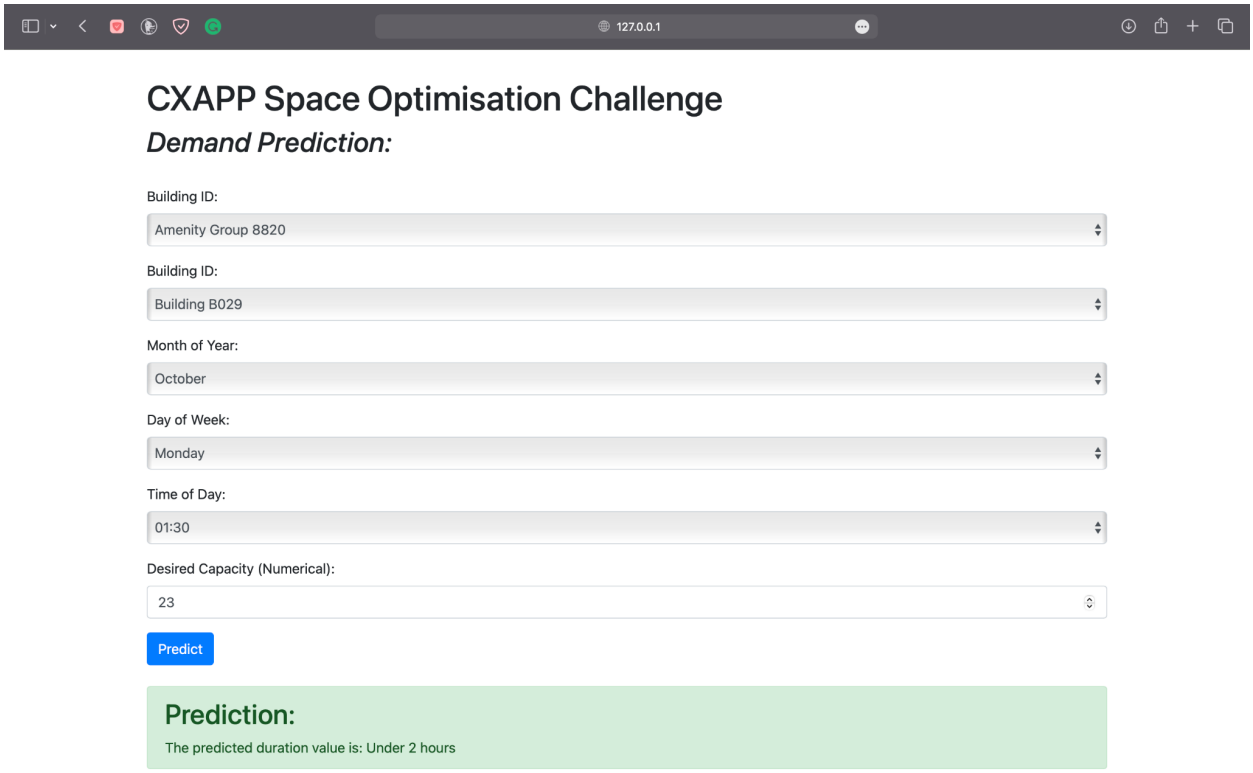
3.7. API Development

API for demand prediction using a machine learning model is developed using FastAPI, a modern Python web framework. Here are the key components of this API development:

- FastAPI Setup: The FastAPI application is created and initialized.
- Templates for HTML: The Jinja2Templates module is used to render HTML templates for the web interface.
- Model Loading: The pre-trained Random Forest model is loaded from a pickled file (random_forest_model.pkl) into the rf_model variable.
- UI / Front-End: The API includes an HTML form that accepts user input for predicting demand. The form fields include amenities, building ID, month of the year, day of the week, time of day, and desired capacity.
- Feature Preprocessing: User input is processed to prepare features for prediction. This includes encoding categorical variables and formatting time-related inputs.
- Prediction: The loaded Random Forest model is used to make predictions based on the provided input features.
- Result Rendering: The prediction result is rendered back to the user on the web page. It displays the predicted duration value.
- JavaScript Interaction: JavaScript is used to toggle visibility for the time input field based on user selection. If the user selects a specific time of day from the dropdown, the numeric input field is hidden.
- Styling: Bootstrap is used to style the HTML elements, providing a clean and user-friendly interface.
- API Deployment: Once the FastAPI application is configured and the API routes are defined, it can be deployed on a web server for public access.

Overall, this FastAPI application provides a user-friendly web interface for predicting demand based on various input parameters. Users can interact with the API by providing their preferences, and the model will return a prediction for the duration of their room booking. This demonstrates how to integrate machine learning models into web applications for real-world use cases.

Fig 3.7.a



3.8. Model Evaluation, Trade-Offs, and Future Scope

Model Evaluation:

- The model achieved an accuracy of 88% in predicting booking categories, classifying them into three groups based on duration.

```
Accuracy: 0.88
              precision    recall  f1-score   support

           A       0.41      0.18      0.25      1443
           B       0.28      0.13      0.18       104
           z       0.90      0.97      0.93     11694

    accuracy                           0.88     13241
   macro avg       0.53      0.43      0.45     13241
weighted avg       0.84      0.88      0.85     13241
```

- Real-world deployment is essential to validate its performance further.
- Fine-tuning can be implemented on the client side to gather feedback and improve predictions using methods like natural language processing (NLP).

Trade-Offs:

- Due to time constraints, amenities groups were directly encoded, leading to the loss of semantic correlation between similar amenities.
- Resource limitations influenced the choice of a random forest model and the grouping of capacities and durations into categories.
- A major trade-off is the limited utilization of room-specific information due to a lack of detailed data, making rooms an index for the model.

Future Scope:

- Addressing trade-offs can enhance the model's capabilities, making it a powerful prediction and recommendation engine.
- The model can provide insights into patterns, resource optimization, anomaly detection, and monetization opportunities.
- Implementing premium data analytics services can offer a valuable revenue stream.
- Solving the optimization challenge by saving resources and gaining domain knowledge is a long-term goal.

Overall, the model has the potential to provide valuable insights, optimize resource usage, and contribute to a deeper understanding of the domain, with monetization opportunities in mind.

THANK YOU

GitHub link: https://github.com/ManishKumarV/Space-Optimization
Email: vuppugandlamanish@gmail.com