SKYHACK

FROM NETAJI SUBHAS UNIVERISTY OF TECHNOLOGY

PRESENTED BY - SUSHANT SUBHASH KADDU(2021UCS1537) AND SAVAR KAUL (2021UCS1661)

PROBLEM STATEMENT

- As United Airlines continues its journey to become the best airline in the history of aviation, it is crucial to provide world-class customer service, for which one of the key areas of focus is our call center operations. Call centers play a critical role in ensuring customer issues are resolved quickly and efficiently, but we face challenges in improving metrics such as Average Handle Time (AHT) and Average Speed to Answer (AST).
- Your task is to optimize these key call center metrics, helping reduce resolution times and providing faster, more efficient service to our customers. You are required to analyze our existing call center data to identify inefficiencies, determine the drivers of long AHT and AST, and suggest strategies to enhance customer satisfaction, reduce escalations, and improve overall operational efficiency.

- **Background**

  In today’s competitive airline industry, providing efficient and reliable customer service is crucial for customer retention and loyalty. Our call center, which handles customer inquiries, complaints, and service requests, is an essential touchpoint for many of our passengers. However, the growing demand and complexity of services have made it increasingly important to optimize the operations of this critical channel.

- Average Handle Time (AHT) and Average Speed to Answer (AST) are essential metrics that significantly impact call center performance by shaping customer satisfaction and operational efficiency. AHT measures the total time agents spend on each call, from answering to disconnecting, and provides insights into where processes can be streamlined. Reducing AHT without sacrificing quality allows agents to handle more calls with existing resources, improving service levels and controlling costs. Meanwhile, AST tracks how quickly customers reach assistance through self-service tools like IVR systems. A lower AST minimizes customer wait times, enhancing their experience and reducing call abandonment, ultimately supporting a more efficient and customer-friendly operation.
- United Airlines uses a mix of human agents and IVR systems to address customer needs. While IVR systems help automate simple tasks and reduce call volume for human agents, there’s an opportunity to streamline this process and better allocate resources between self-service and agent-based resolutions.

How are AHT and AST Calculated?

- **AHT (Average Handle Time):**
  Time from when the agent picks up the call to when they hang up

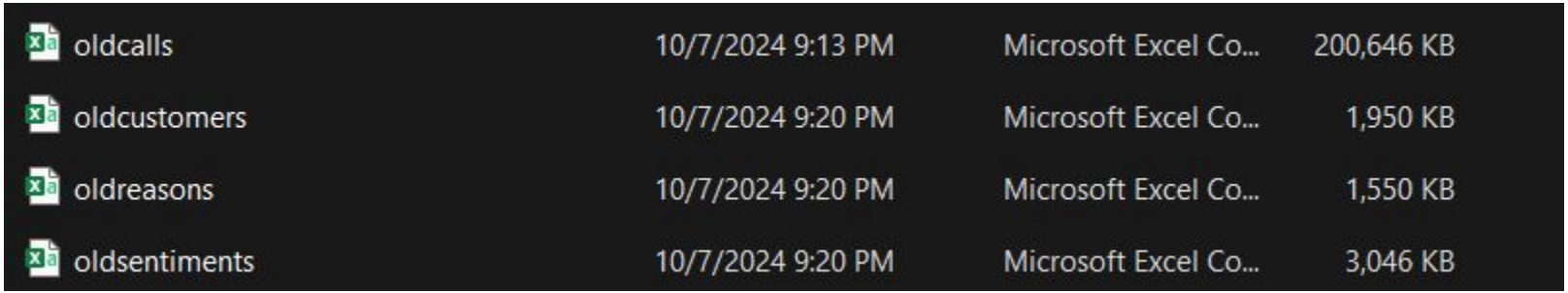
Formula:
AHT = Total Handle Time / Total Number of Calls

- **AST (Average Speed to Answer):**
  Time spent by the customer in queue till the agent answers the call

Formula:
AST = Total Waiting Time / Total Number of Calls

Understanding the data

- 4 .csv files were provided
- We will ensure data quality and consistency across the four provided CSV files for effective analysis.
- We will identify any missing values and fill them with the appropriate statistical measures, such as the mean, median, or mode, depending on the nature of the data.
- We will use Jupyter Notebook software to perform data cleaning and analysis, leveraging libraries such as pandas for data manipulation, NumPy for numerical operations, and Matplotlib or Seaborn for data visualization.



Oldcall.csv file

In [21]:
```
data_call=pd.read_csv('oldcalls.csv')
```

In [22]:
```
data_call.head()
```

Out[22]:

	call_id	customer_id	agent_id	call_start_datetime	agent_assigned_datetime	call_end_datetime	call_transcript
0	4667960400	2033123310	963118	7/31/2024 23:56	8/1/2024 0:03	8/1/2024 0:34	\n\nAgent: Thank you for calling United Airlin...
1	1122072124	8186702651	519057	8/1/2024 0:03	8/1/2024 0:06	8/1/2024 0:18	\n\nAgent: Thank you for calling United Airlin...
2	6834291559	2416856629	158319	7/31/2024 23:59	8/1/2024 0:07	8/1/2024 0:26	\n\nAgent: Thank you for calling United Airlin...
3	2266439882	1154544516	488324	8/1/2024 0:05	8/1/2024 0:10	8/1/2024 0:17	\n\nAgent: Thank you for calling United Airlin...
4	1211603231	5214456437	721730	8/1/2024 0:04	8/1/2024 0:14	8/1/2024 0:23	\n\nAgent: Thank you for calling United Airlin...

In [25]:
```
# Convert call_start_datetime and call_end_datetime to datetime format
data_call['call_start_datetime'] = pd.to_datetime(data_call['call_start_datetime'])
data_call['call_end_datetime'] = pd.to_datetime(data_call['call_end_datetime'])
data_call['agent_assigned_datetime'] = pd.to_datetime(data_call['agent_assigned_datetime'])

# Calculate call duration
data_call['total_customer_duration'] = data_call['call_end_datetime'] - data_call['call_start_datetime']
data_call['customer_wait_time_queue'] = data_call['agent_assigned_datetime'] - data_call['call_start_datetime']
```

In [26]:
```
data_call.head()
```

Out[26]:

	call_id	customer_id	agent_id	call_start_datetime	agent_assigned_datetime	call_end_datetime	total_customer_duration	customer_wait_time_queue
0	4667960400	2033123310	963118	2024-07-31 23:56:00	2024-08-01 00:03:00	2024-08-01 00:34:00	0 days 00:38:00	0 days 00:07:00
1	1122072124	8186702651	519057	2024-08-01 00:03:00	2024-08-01 00:06:00	2024-08-01 00:18:00	0 days 00:15:00	0 days 00:03:00
2	6834291559	2416856629	158319	2024-07-31 23:59:00	2024-08-01 00:07:00	2024-08-01 00:26:00	0 days 00:27:00	0 days 00:08:00
3	2266439882	1154544516	488324	2024-08-01 00:05:00	2024-08-01 00:10:00	2024-08-01 00:17:00	0 days 00:12:00	0 days 00:05:00
4	1211603231	5214456437	721730	2024-08-01 00:04:00	2024-08-01 00:14:00	2024-08-01 00:23:00	0 days 00:19:00	0 days 00:10:00

In [27]:
```
# Export the DataFrame to a CSV file named 'newcalls.csv'
data_call.to_csv('newcalls.csv', index=False)

print("Data exported to newcalls.csv successfully.")
```

```
Data exported to newcalls.csv successfully.
```

Oldsentiments.csv

In [28]:
```
df_sentiments=pd.read_csv('oldsentiments.csv')
```

In [29]:
```
df_sentiments.head()
```

Out[29]:

	call_id	agent_id	agent_tone	customer_tone	average_sentiment	silence_percent_average
0	4667960400	963118	neutral	angry	-0.04	0.39
1	1122072124	519057	calm	neutral	0.02	0.35
2	6834291559	158319	neutral	polite	-0.13	0.32
3	2266439882	488324	neutral	frustrated	-0.20	0.20
4	1211603231	721730	neutral	polite	-0.05	0.35

In [30]:
```
missing_values = df_sentiments.isnull().sum()

# Display the missing values
print("Missing values in each column:")
print(missing_values)
```

```
Missing values in each column:
call_id                   0
agent_id                  0
agent_tone              217
customer_tone             0
average_sentiment       109
silence_percent_average   0
dtype: int64
```

In [31]:
```
df_sentiments['agent_tone'].fillna(df_sentiments['agent_tone'].mode()[0], inplace=True)

# Fill missing values for average_sentiment with median
df_sentiments['average_sentiment'].fillna(df_sentiments['average_sentiment'].median(), inplace=True)
```

In [32]:
```
missing_values = df_sentiments.isnull().sum()

# Display the missing values
print("Missing values in each column:")
print(missing_values)
```

```
Missing values in each column:
call_id                   0
agent_id                  0
agent_tone                0
customer_tone             0
average_sentiment         0
silence_percent_average   0
dtype: int64
```

In [33]:
```
df_sentiments.to_csv('newsentiments.csv', index=False)
```

- Agent tone missing values will be filled with mode since its categorical
- Average sentiment missing values will be filled with median since it's continuous and may have outliers.
- The changes are made a saved in a new file called newsentiments.csv

oldcustomers.csv

In [15]:
```
data_customers=pd.read_csv('oldcustomers.csv')
```

In [16]:
```
data_customers.head()
```

Out[16]:

	customer_id	customer_name	elite_level_code
0	2033123310	Matthew Foster	4.0
1	8186702651	Tammy Walters	NaN
2	2416856629	Jeffery Dixon	NaN
3	1154544516	David Wilkins	2.0
4	5214456437	Elizabeth Daniels	0.0

In [17]:
```
unique_elite_levels = data_customers['elite_level_code'].unique()

# Display the unique values
print(unique_elite_levels)
```

```
[ 4. nan  2.  0.  5.  1.  3.]
```

- The missing values of elite_level_code are being filled with median
- The changes made are saved to a new file called newcustomers.csv

In [19]:
```
median_value = data_customers['elite_level_code'].median()
```

In [20]:
```
data_customers['elite_level_code'].fillna(median_value, inplace=True)

# Optional: Save the updated DataFrame back to a CSV file
data_customers.to_csv('newcustomers.csv',index=False)
```

oldreasons.csv

- In [5]:
```
df = pd.read_csv('oldreasons.csv')
```

In [6]:
```
df.head()
```

Out[6]:

	call_id	primary_call_reason
0	4667960400	Voluntary Cancel
1	1122072124	Booking
2	6834291559	IRROPS
3	2266439882	Upgrade
4	1211603231	Seating

In [7]:
```
num_rows = df.shape[0]
print(f'Number of rows: {num_rows}')
```

```
Number of rows: 66653
```

In [8]:
```
# Get the count of each unique reason in the 'primary_call_reason' column
reason_counts = df['primary_call_reason'].value_counts()
print(reason_counts)
```

```
IRROPS                    13057
Voluntary Change          10291
Seating                    6223
Mileage Plus               5487
Post-Flight                3869
Communications             3779
Products and Services      2792
Upgrade                    2682
Baggage                    2616
Booking                    2589
Checkout                   1840
Check-In                   1490
Voluntary Cancel           1304
Digital Support             996
ETC                         930
Traveler Updates            772
Schedule Change             707
Other Topics                568
Products & Services         476
Disability                  394
```

- `primary_call_reason` column has multiple variations of the same categories due to inconsistent formatting (extra spaces, different cases, etc.)
- To make this data consistent, we should standardize the `primary_call_reason` column by stripping extra spaces and converting the text to lowercase

In [9]:
```
# Standardize the 'primary_call_reason' by stripping extra spaces and converting to title case
df['primary_call_reason'] = df['primary_call_reason'].str.strip().str.title()

# Check the updated unique reasons and their counts
cleaned_reason_counts = df['primary_call_reason'].value_counts()
print(cleaned_reason_counts)
```

```
Irrops                  13311
Voluntary Change        10499
Seating                  6365
Mileage Plus             5587
Post-Flight              3957
Communications           3840
Products And Services    2856
Baggage                  2832
Upgrade                  2738
Booking                  2637
Checkout                 1888
Check-In                 1519
Voluntary Cancel         1329
Digital Support          1014
Etc                       952
Traveler Updates          782
Schedule Change           731
```

- Oldreasons.csv file had less rows compared to the other .datasets that were provided and so to make the number of rows equal to it we did the data distribution of call_id and primary_call_reason
- We generated random call_id values to ensure each entry in the dataset is uniquely identifiable, simulating a realistic dataset structure.
- For the primary_call_reason column, we analyzed the existing distribution of complaint types and created a weighted list. This allowed us to randomly assign reasons to new rows, accurately reflecting the original dataset's patterns.

```
 Create a list of reasons weighted by their counts
easons = []
or reason, count in current_distribution.items():
    reasons.extend([reason] * count)

 Randomly sample 5157 reasons from the weighted list
ew_reasons = random.choices(reasons, k=5157)

 Generate random phone numbers in the format already present in
ef generate_phone_number():
    return f"{random.randint(1000000000, 9999999999)}"  # 10-dig

ew_phone_numbers = [generate_phone_number() for _ in range(5157)

 Create a new DataFrame with the new rows
ew_data = pd.DataFrame({
    'call_id': new_phone_numbers,
    'primary_call_reason': new_reasons
})

 Append the new data to the existing DataFrame
f = pd.concat([df, new_data], ignore_index=True)

 Optional: Reset the index if needed
f.reset_index(drop=True, inplace=True)

 Display the new DataFrame's shape
rint(df.shape)
```

```
# Display the new DataFrame's shape
print(df.shape)
```

```
(71810, 2)
```

In [12]:
```
df.to_csv('newreasons.csv', index=False)

print("DataFrame has been exported to 'newreasons.csv'")
```

```
DataFrame has been exported to 'newreasons.csv'
```

Calculating the total customer duration and the customer wait time in queue in the calls file

In [25]:
```
# Convert call_start_datetime and call_end_datetime to datetime format
data_call['call_start_datetime'] = pd.to_datetime(data_call['call_start_datetime'])
data_call['call_end_datetime'] = pd.to_datetime(data_call['call_end_datetime'])
data_call['agent_assigned_datetime'] = pd.to_datetime(data_call['agent_assigned_datetime'])

# Calculate call duration
data_call['total_customer_duration'] = data_call['call_end_datetime'] - data_call['call_start_datetime']
data_call['customer_wait_time_queue'] = data_call['agent_assigned_datetime'] - data_call['call_start_datetime']
```

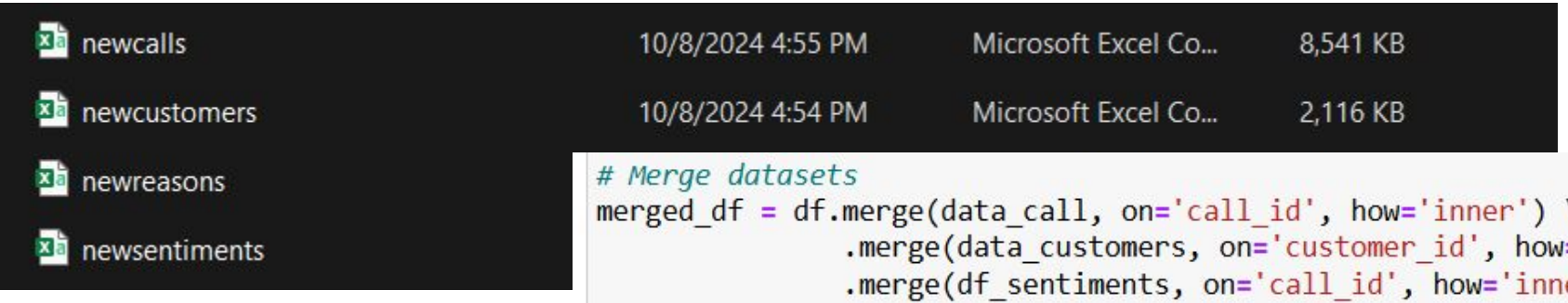
In [26]:
```
data_call.head()
```

Out[26]:

	call_id	customer_id	agent_id	call_start_datetime	agent_assigned_datetime	call_end_datetime	total_customer_duration	customer_wait_time_queue
0	4667960400	2033123310	963118	2024-07-31 23:56:00	2024-08-01 00:03:00	2024-08-01 00:34:00	0 days 00:38:00	0 days 00:07:00
1	1122072124	8186702651	519057	2024-08-01 00:03:00	2024-08-01 00:06:00	2024-08-01 00:18:00	0 days 00:15:00	0 days 00:03:00
2	6834291559	2416856629	158319	2024-07-31 23:59:00	2024-08-01 00:07:00	2024-08-01 00:26:00	0 days 00:27:00	0 days 00:08:00
3	2266439882	1154544516	488324	2024-08-01 00:05:00	2024-08-01 00:10:00	2024-08-01 00:17:00	0 days 00:12:00	0 days 00:05:00
4	1211603231	5214456437	721730	2024-08-01 00:04:00	2024-08-01 00:14:00	2024-08-01 00:23:00	0 days 00:19:00	0 days 00:10:00

The new filtered/cleared .csv files

- After clearing the data from the old files and now that there is consistency among them we will ensure that these new files are merged together to provide the necessary insights.
- We will now merge these new files to create a new file called “united_merged.csv” file and this file will be used further to be worked upon




```
# Merge datasets
merged_df = df.merge(data_call, on='call_id', how='inner') \
              .merge(data_customers, on='customer_id', how='inner') \
              .merge(df_sentiments, on='call_id', how='inner')

# Save to a new CSV file
merged_df.to_csv('united_merged.csv', index=False)
```

```
df=pd.read_csv('united_merged.csv')
```

In [105]:
```
df=pd.read_csv('united_merged.csv')
```

In [106]:
```
df.head()
```

Out[106]:

	call_id	primary_call_reason	customer_id	agent_id_x	call_start_datetime	agent_assigned_datetime	call_end_datetime	total_customer_duration	customer
0	1122072124	Booking	8186702651	519057	2024-08-01 00:03:00	2024-08-01 00:06:00	2024-08-01 00:18:00	00:15:00	
1	6834291559	Irrops	2416856629	158319	2024-07-31 23:59:00	2024-08-01 00:07:00	2024-08-01 00:26:00	00:27:00	
2	2266439882	Upgrade	1154544516	488324	2024-08-01 00:05:00	2024-08-01 00:10:00	2024-08-01 00:17:00	00:12:00	
3	1211603231	Seating	5214456437	721730	2024-08-01 00:04:00	2024-08-01 00:14:00	2024-08-01 00:23:00	00:19:00	
4	5297766997	Mileage Plus	5590154991	817160	2024-08-01 00:11:00	2024-08-01 00:16:00	2024-08-01 00:40:00	00:29:00	

TASK 1

- Long average handle time (AHT) affects both efficiency and customer satisfaction. Explore the factors contributing to extended call durations, such as agent performance, call types, and sentiment. Identify key drivers of long AHT and AST, especially during high volume call periods. Additionally, could you quantify the percentage difference between the average handling time for the most frequent and least frequent call reasons?

Using PostgreSQL

- The united_merged.csv file will be exported in our postgresql table which is called united_table

Data Output | Messages | Notifications

	call_id [PK] bigint	primary_call_reason character varying (255)	customer_id bigint	agent_id_x bigint	call_start_datetime timestamp without time zone	agent_assigned_datetime timestamp without time zone	call_end_datetime timestamp without time zone
1	131642	Seating	103262588	132311	2024-08-04 17:46:00	2024-08-04 17:55:00	2024-08-04 17:59:00
2	325527	Irrops	8994245383	607742	2024-08-18 07:41:00	2024-08-18 07:45:00	2024-08-18 07:50:00
3	526871	Voluntary Change	5769982111	748700	2024-08-10 11:59:00	2024-08-10 12:08:00	2024-08-10 12:15:00
4	634548	Seating	9534089270	542034	2024-08-11 11:15:00	2024-08-11 11:25:00	2024-08-11 11:39:00
5	693921	Voluntary Change	3046806067	633922	2024-08-06 15:58:00	2024-08-06 16:09:00	2024-08-06 16:15:00
6	723257	Mileage Plus	8760640596	717526	2024-08-10 16:21:00	2024-08-10 16:29:00	2024-08-10 16:55:00
7	729176	Irrops	1310505652	675727	2024-08-10 19:41:00	2024-08-10 19:48:00	2024-08-10 20:18:00
8	768800	Communications	2333085875	422417	2024-08-30 18:07:00	2024-08-30 18:10:00	2024-08-30 18:52:00
9	811664	Voluntary Cancel	3800366680	176703	2024-08-31 09:14:00	2024-08-31 09:25:00	2024-08-31 09:32:00
10	1156029	Etc	4598395794	590950	2024-08-04 08:24:00	2024-08-04 08:31:00	2024-08-04 08:39:00
11	1228137	Upgrade	7388532911	153460	2024-08-20 10:25:00	2024-08-20 10:31:00	2024-08-20 10:45:00

AVERAGE HANDLE TIME

Query | Query History

```sql
SELECT
    agent_id_x,
    SUM(EXTRACT(EPOCH FROM (call_end_datetime - agent_assigned_datetime))) AS total_handle_time_in_seconds,
    COUNT(*) AS total_calls,
    SUM(EXTRACT(EPOCH FROM (call_end_datetime - agent_assigned_datetime))) / COUNT(*) AS AHT
FROM
    united_table
GROUP BY
    agent_id_x;

```

Data Output | Messages | Notifications

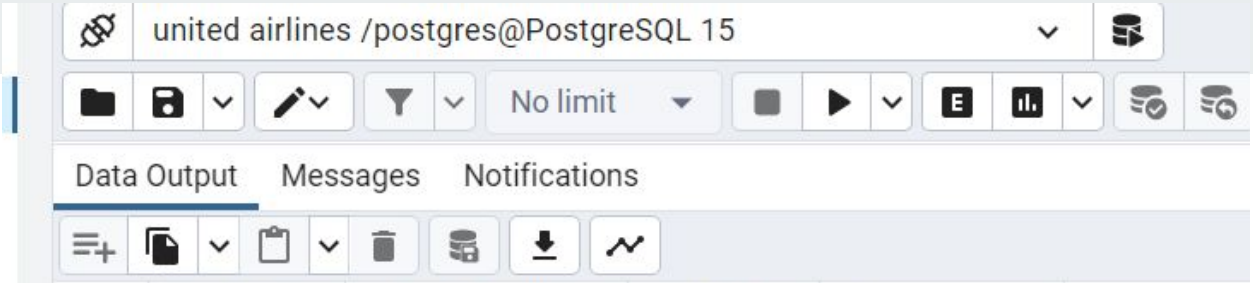
	agent_id_x bigint	total_handle_time numeric	total_calls bigint	aht numeric
1	376343	257760.000000	420	613.7142857142857143
2	693396	33540.000000	37	906.4864864864864865
3	404951	53280.000000	73	729.8630136986301370
4	102574	7200.000000	2	3600.0000000000000000
5	470395	207960.000000	274	758.9781021897810219
6	204674	269100.000000	380	708.1578947368421053
7	664625	41520.000000	71	584.7887323943661972
8	308266	191040.000000	277	689.6750902527075812
9	294450	26640.000000	43	619.5348837209302326
10	814752	249360.000000	367	679.4550408719346049
11	925821	9360.000000	12	780.0000000000000000
12	437196	115680.000000	179	646.2569832402234637
13	388661	138420.000000	220	629.1818181818181818
14	377855	22140.000000	29	763.4482758620689655
15	789268	96360.000000	133	724.5112781954887218
16	926719	88020.000000	133	661.8045112781954887
17	342345	169320.000000	248	682.7419354838709677
18	419382	45300.000000	59	767.7966101694915254
19	817413	92280.000000	125	738.2400000000000000
20	833458	27900.000000	34	820.5882352941176471

AVERAGE SPEED TIME

Query | Query History

```sql
SELECT
    agent_id_x,
    SUM(customer_wait_time_in_queue) AS total_waiting_time,    -- Total Waiting Time
    COUNT(*) AS total_calls,                                   -- Total Number of Calls
    SUM(customer_wait_time_in_queue) / COUNT(*) AS ast         -- Average Speed Time
FROM
    united_table
GROUP BY
    agent_id_x;
```

- Running the query in PostgreSQL



	agent_id_x bigint	total_waiting_time interval	total_calls bigint	ast interval
1	376343	51:36:00	420	00:07:22.285714
2	693396	04:26:00	37	00:07:11.351351
3	404951	08:21:00	73	00:06:51.780822
4	102574	00:16:00	2	00:08:00
5	470395	32:24:00	274	00:07:05.693431
6	204674	46:47:00	380	00:07:23.210526
7	664625	08:33:00	71	00:07:13.521127
8	308266	34:53:00	277	00:07:33.357401
9	294450	05:15:00	43	00:07:19.534884
10	814752	45:38:00	367	00:07:27.629428
11	925821	01:14:00	12	00:06:10
12	437196	21:08:00	179	00:07:05.027933
13	388661	27:27:00	220	00:07:29.181818

PEAK LOAD ANALYSIS- IDENTIFIES THE PEAK TIMES WHEN AHT AND AST ARE HIGHEST, FOCUSING ON HOURLY TRENDS

Query | Query History

```
1  SELECT DATE_TRUNC('hour', call_start_datetime) AS call_hour,
2         AVG(customer_wait_time_in_queue) AS avg_ast,
3         AVG(average_handle_time) AS avg_aht
4  FROM united_table
5  GROUP BY call_hour
6  ORDER BY AVG(average_handle_time) DESC;
7
```

Data Output | Messages | Notifications

	call_hour timestamp without time zone	avg_ast interval	avg_aht interval
1	2024-08-20 22:00:00	00:06:56	00:22:56
2	2024-08-15 05:00:00	00:07:12	00:22:40
3	2024-08-08 00:00:00	00:07:12	00:20:24
4	2024-08-07 05:00:00	00:06:50	00:20:20
5	2024-08-19 01:00:00	00:07:42.857143	00:19:21.428571

SENTIMENTS AND PERFORMANCE- HELPS US IDENTIFY WHEATHER NEGATIVE CUSTOMER SENTIMENTS CORRELATES WITH LONGER CALL TIMES

Query | Query History

```
1  SELECT customer_tone,
2         AVG(customer_wait_time_in_queue) AS avg_ast,
3         AVG(average_handle_time) AS avg_aht
4  FROM united_table
5  GROUP BY customer_tone
6  ORDER BY AVG(average_handle_time) DESC;
7
```

Data Output | Messages | Notifications

	customer_tone character varying (50)	avg_ast interval	avg_aht interval
1	neutral	00:07:17.621251	00:11:46.014028
2	calm	00:07:17.270673	00:11:41.753276
3	angry	00:07:16.876404	00:11:34.049438
4	frustrated	00:07:16.491493	00:11:32.261626
5	polite	00:07:16.597746	00:11:31.68172

IDENTIFY TRENDS IN AHT BASED ON THE TIME OF DAY

Query | Query History

```
1  SELECT EXTRACT(HOUR FROM call_start_datetime) AS hour,
2         AVG(average_handle_time) AS avg_aht
3  FROM united_table
4  GROUP BY hour
5  ORDER BY hour;
```

Data Output | Messages | Notifications

	hour numeric	avg_aht interval
1	0	00:11:25.601504
2	1	00:12:19.54023
3	2	00:11:48.559585
4	3	00:11:44.609218
5	4	00:11:19.979695
6	5	00:11:31.428571
7	6	00:11:44.464286
8	7	00:11:40.216655
9	8	00:11:43.284202
16	15	00:11:32.847078
17	16	00:11:26.178462
18	17	00:11:37.698136
19	18	00:12:03.516382
20	19	00:11:23.009709
21	20	00:11:02.623574
22	21	00:11:29.212828
23	22	00:11:55.402299
24	23	00:11:18.439306

Agent Performance Evaluate agent performance by comparing their AHT and AST. This helps in ranking agents by efficiency and identifying areas for improvement.

Query | Query History

```
1  SELECT agent_id_x,
2         AVG(average_handle_time) AS avg_aht,
3         AVG(customer_wait_time_in_queue) AS avg_ast
4  FROM united_table
5  GROUP BY agent_id_x
6  ORDER BY avg_aht ASC;
```

Data Output | Messages | Graph Visualiser | Notifications

	agent_id_x bigint	avg_aht interval	avg_ast interval
1	547592	00:03:00	00:10:00
2	616988	00:04:45	00:04:15
3	229129	00:05:52.5	00:06:00
4	676262	00:06:30	00:07:30

Call Reasons and Their Impact Analyze the impact of different call reasons on AHT and AST to understand which call types contribute to longer handling times. sql

Query | Query History

```sql
SELECT primary_call_reason,
       AVG(average_handle_time) AS avg_aht,
       AVG(customer_wait_time_in_queue) AS avg_ast
FROM united_table
GROUP BY primary_call_reason
ORDER BY avg_aht DESC;
```

Data Output | Messages | Graph Visualiser | Notifications

	primary_call_reason character varying (255)	avg_aht interval	avg_ast interval
1	Checkout	00:16:56.853814	00:12:04.70339
2	Mileage Plus	00:16:35.573406	00:06:01.015211
3	Etc	00:16:02.89916	00:09:04.285714
4	Post-Flight	00:15:32.896074	00:09:29.431871

Frequent Call Reasons Identify the most frequent and least frequent call reasons, which will help quantify the percentage difference in AHT.

Query | Query History

```
1  SELECT primary_call_reason,
2         COUNT(*) AS call_count
3  FROM united_table
4  GROUP BY primary_call_reason
5  ORDER BY call_count DESC;
6
```

Data Output | Messages | Graph Visualiser | Notifications

	primary_call_reason character varying (255)	call_count bigint
1	Irrops	13311
2	Voluntary Change	10848
3	Seating	6365
4	Mileage Plus	5851
16	Traveler Updates	937
17	Other Topics	818
18	Schedule Change	731
19	Disability	403
20	Unaccompanied Minor	104

Total rows: 20 of 20 | Query complete 00:00:00.062

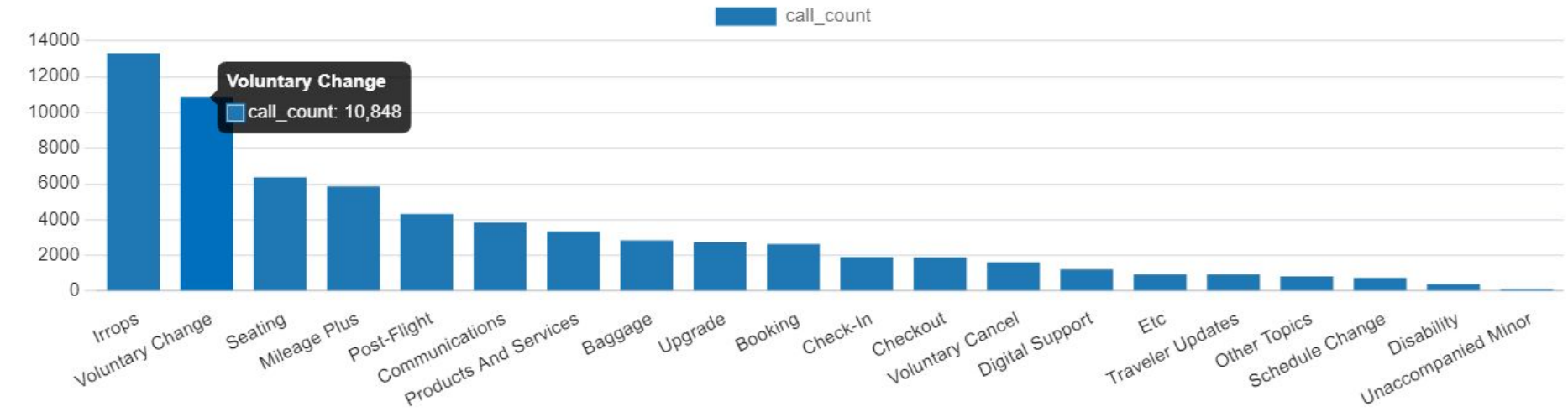
Data Output | Messages | Graph Visualiser | Notifications

Graph Type: Bar Chart

X Axis: primary_call_reason

Y Axis: call_count

Generate



Quantify the percentage difference between the average handling time for the most frequent and least frequent call reasons?.

Query | Query History

```sql
WITH frequent_reason AS (
    SELECT primary_call_reason,
           AVG(average_handle_time) AS avg_aht,
           COUNT(*) AS call_count,
           RANK() OVER (ORDER BY COUNT(*) DESC) AS rank
    FROM united_table
    GROUP BY primary_call_reason
),
aht_values AS (
    SELECT
        MAX(CASE WHEN rank = 1 THEN avg_aht END) AS most_frequent_aht,
        MIN(CASE WHEN rank = (SELECT MAX(rank) FROM frequent_reason) THEN avg_aht END) AS least_frequent_aht
    FROM frequent_reason
)

SELECT
    most_frequent_aht,
    least_frequent_aht,
    (EXTRACT(EPOCH FROM most_frequent_aht) -
     EXTRACT(EPOCH FROM least_frequent_aht)) / EXTRACT(EPOCH FROM least_frequent_aht) * 100 AS percentage_difference
FROM aht_values;
```

Data Output | Messages | Graph Visualiser | Notifications

	most_frequent_aht interval	least_frequent_aht interval	percentage_difference numeric
1	00:13:05.116069	00:08:39.230769	51.20753928201816560700

OVERALL AHT AND AST CALCULATION - Calculate the overall average AHT and AST to provide a benchmark for comparison across agents and call types.

Query | Query History

```
1  SELECT AVG(average_handle_time) AS avg_aht,
2         AVG(customer_wait_time_in_queue) AS avg_ast
3  FROM united_table;
4
```

Data Output | Messages | Graph Visualiser | Notifications

	avg_aht interval	avg_ast interval
1	00:11:37.159275	00:07:16.972334

SENTIMENT ANALYSIS BY CALL REASON

Query | Query History

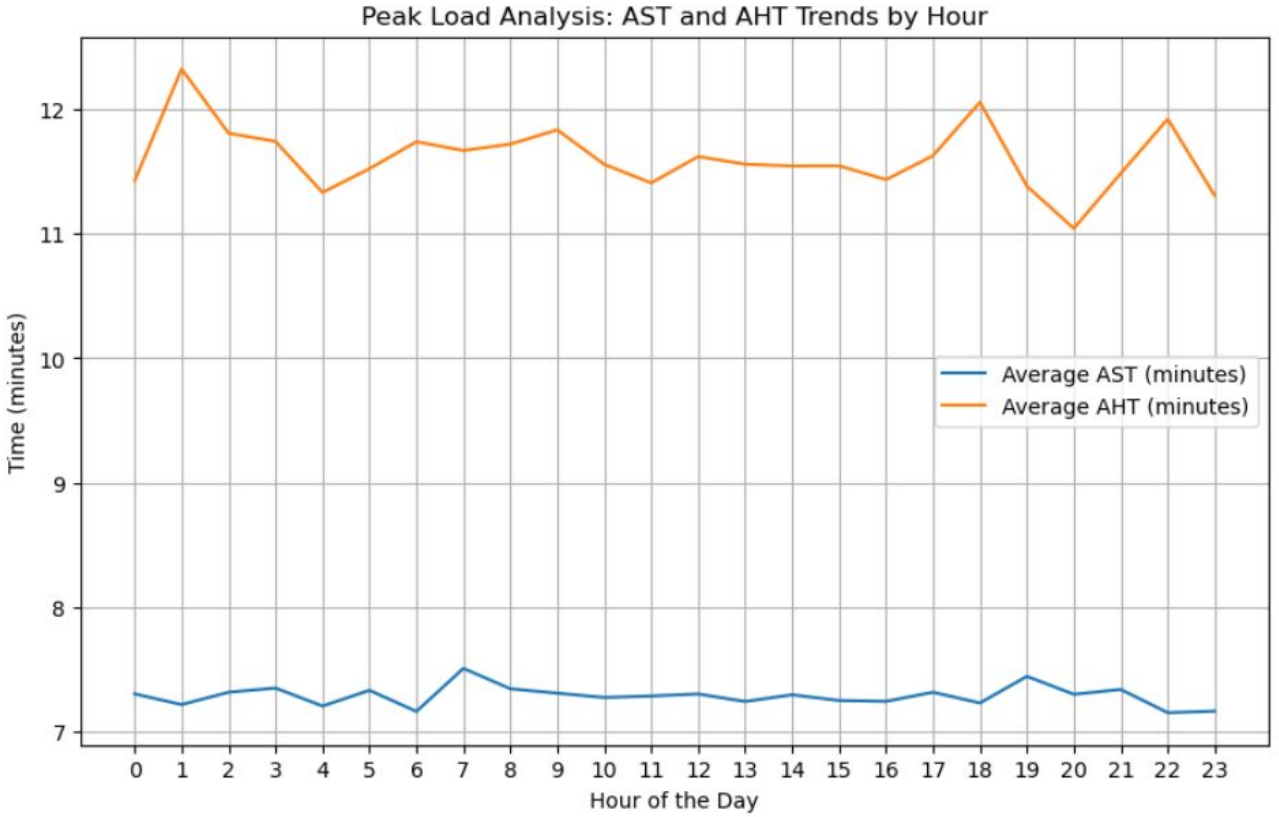
```
1  SELECT primary_call_reason,
2         AVG(average_sentiment) AS avg_sentiment
3  FROM united_table
4  GROUP BY primary_call_reason;
```

Data Output | Messages | Graph Visualiser | Notifications

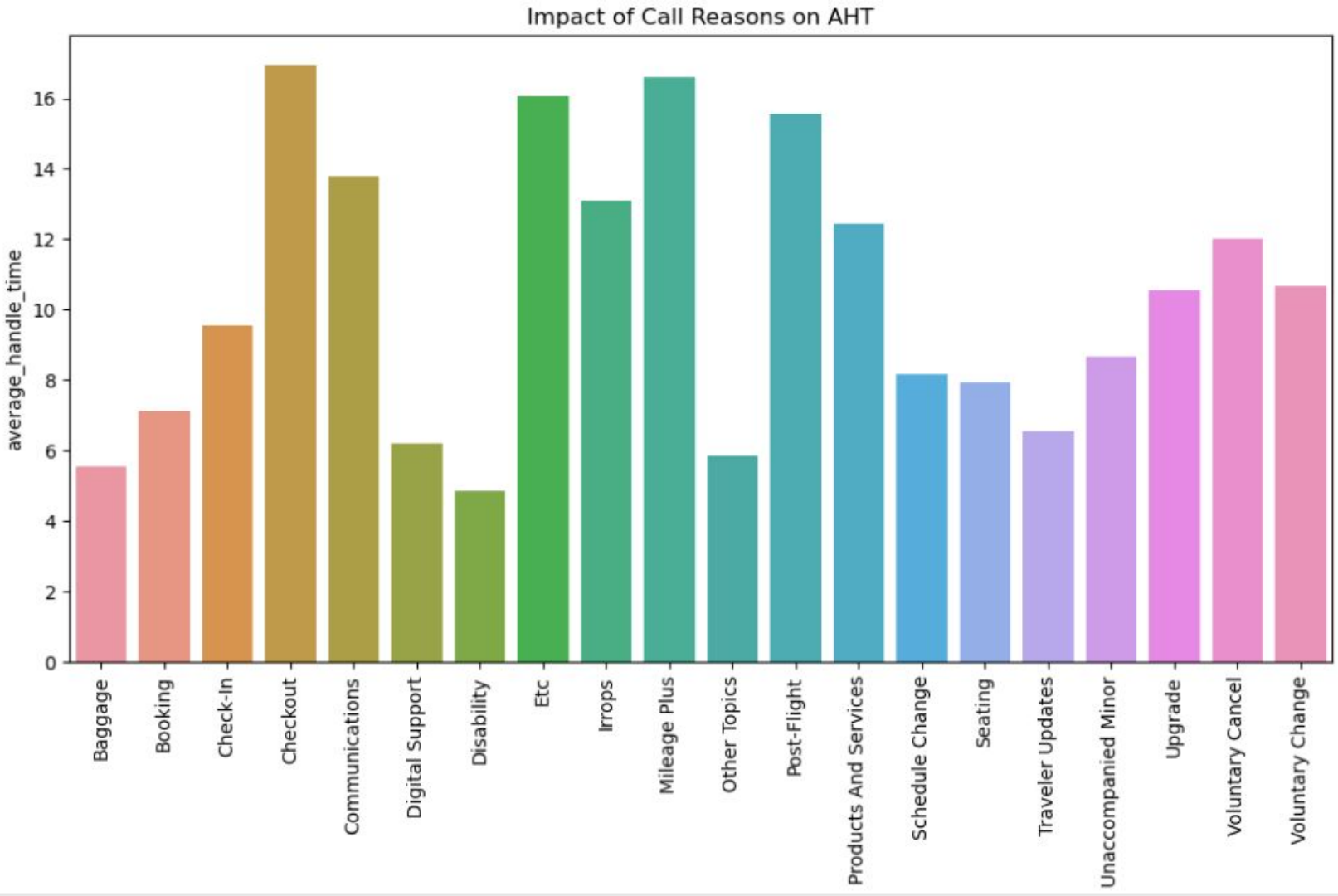
	primary_call_reason character varying (255)	avg_sentiment double precision
1	Voluntary Change	-0.0038449483775811204
2	Digital Support	-0.04204081632653061
3	Products And Services	-0.03440876350540221
4	Unaccompanied Minor	-0.01384615384615385
5	Schedule Change	-0.03384404924760603
6	Etc	-0.04929621848739499
7	Communications	-0.036388020833333416
8	Irrops	-0.07398767936293267
17	Traveler Updates	0.006584845250800432
18	Other Topics	-0.003887530562347187
19	Voluntary Cancel	-0.03150684931506849
20	Upgrade	-0.016011687363038714

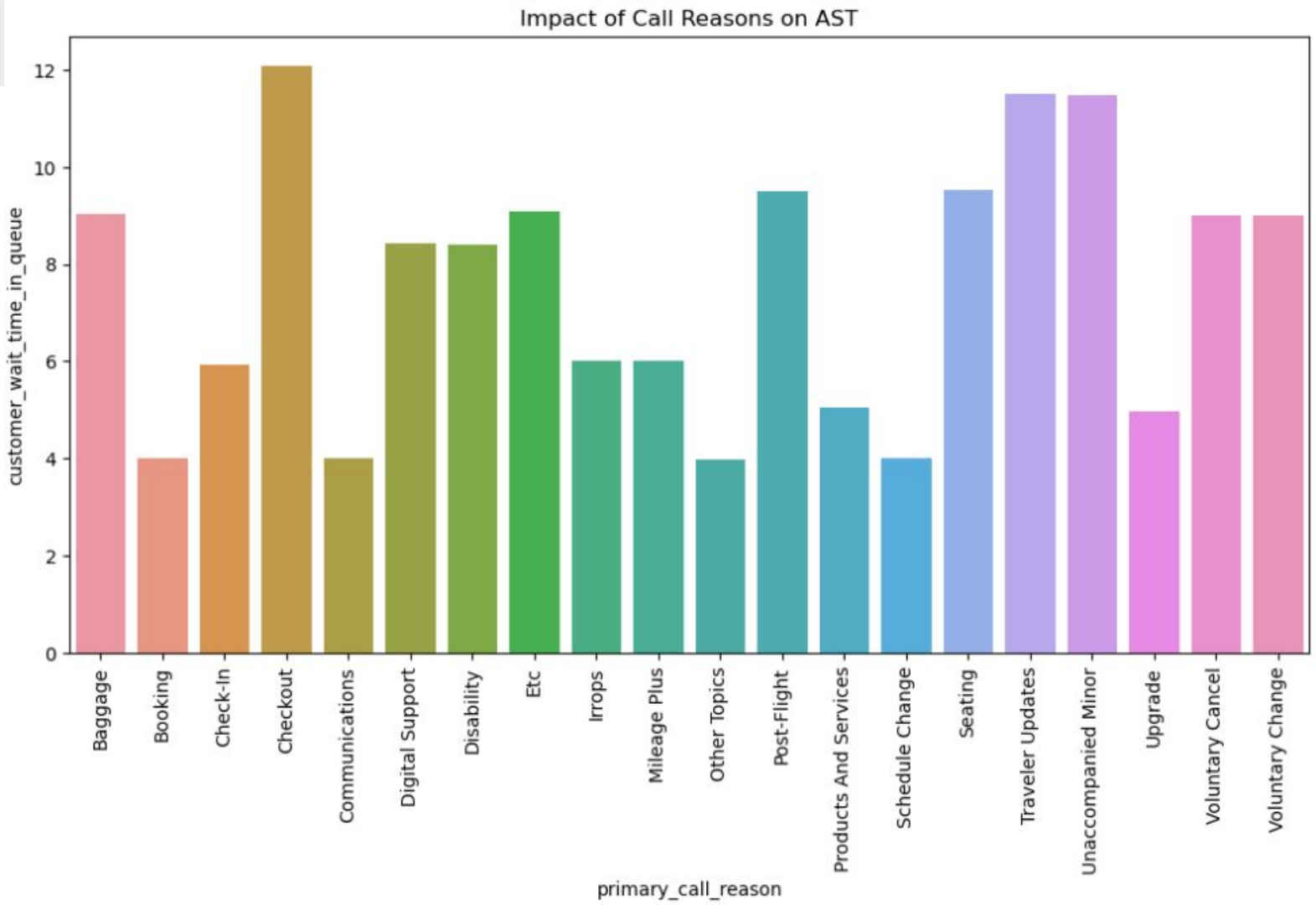
Total rows: 20 of 20 | Query complete 00:00:00.088

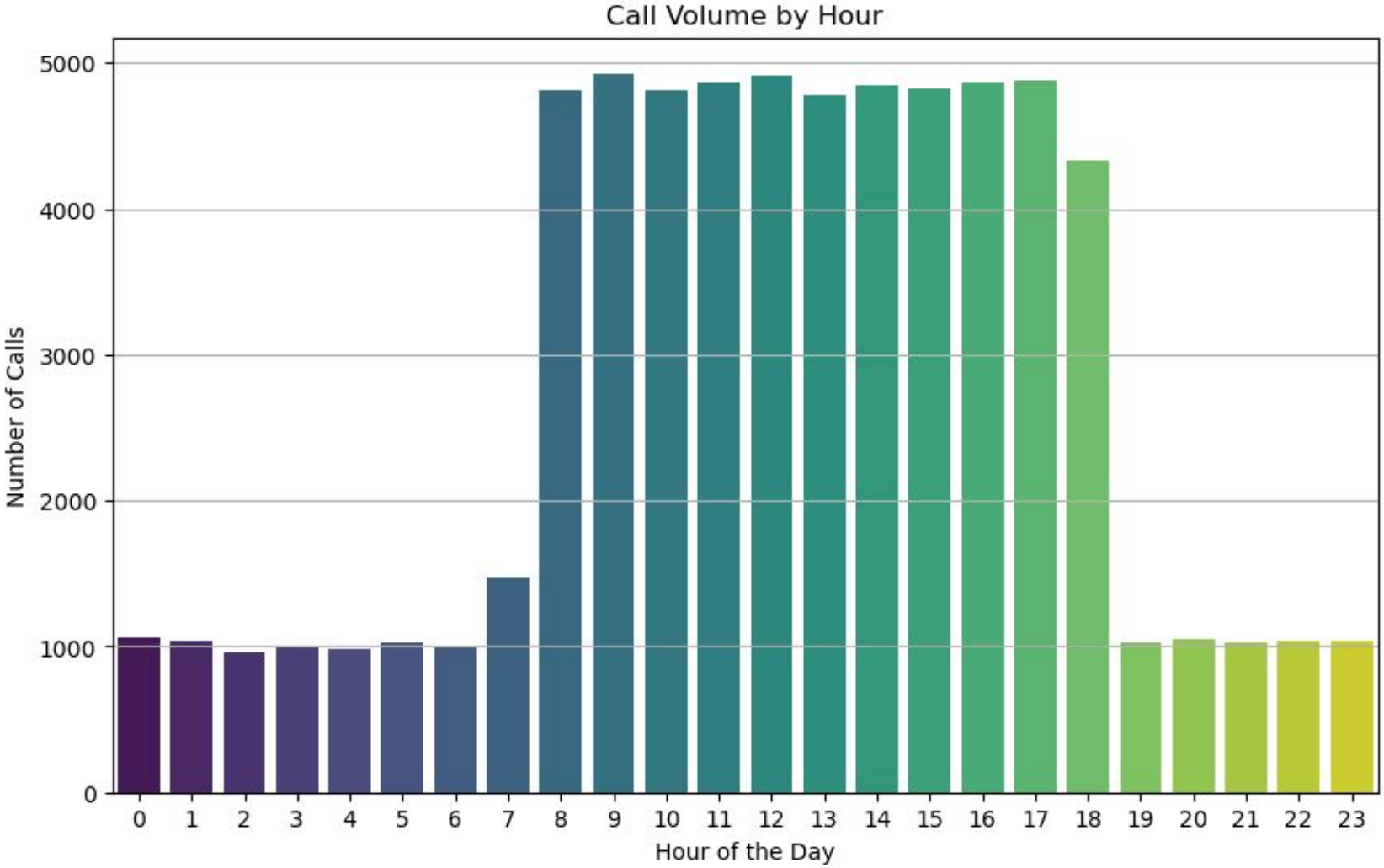
GRAPHS CREATED ON JUPYTER NOTEBOOK (CODE IS IN THE JUPYTER FILE ATTACHED)

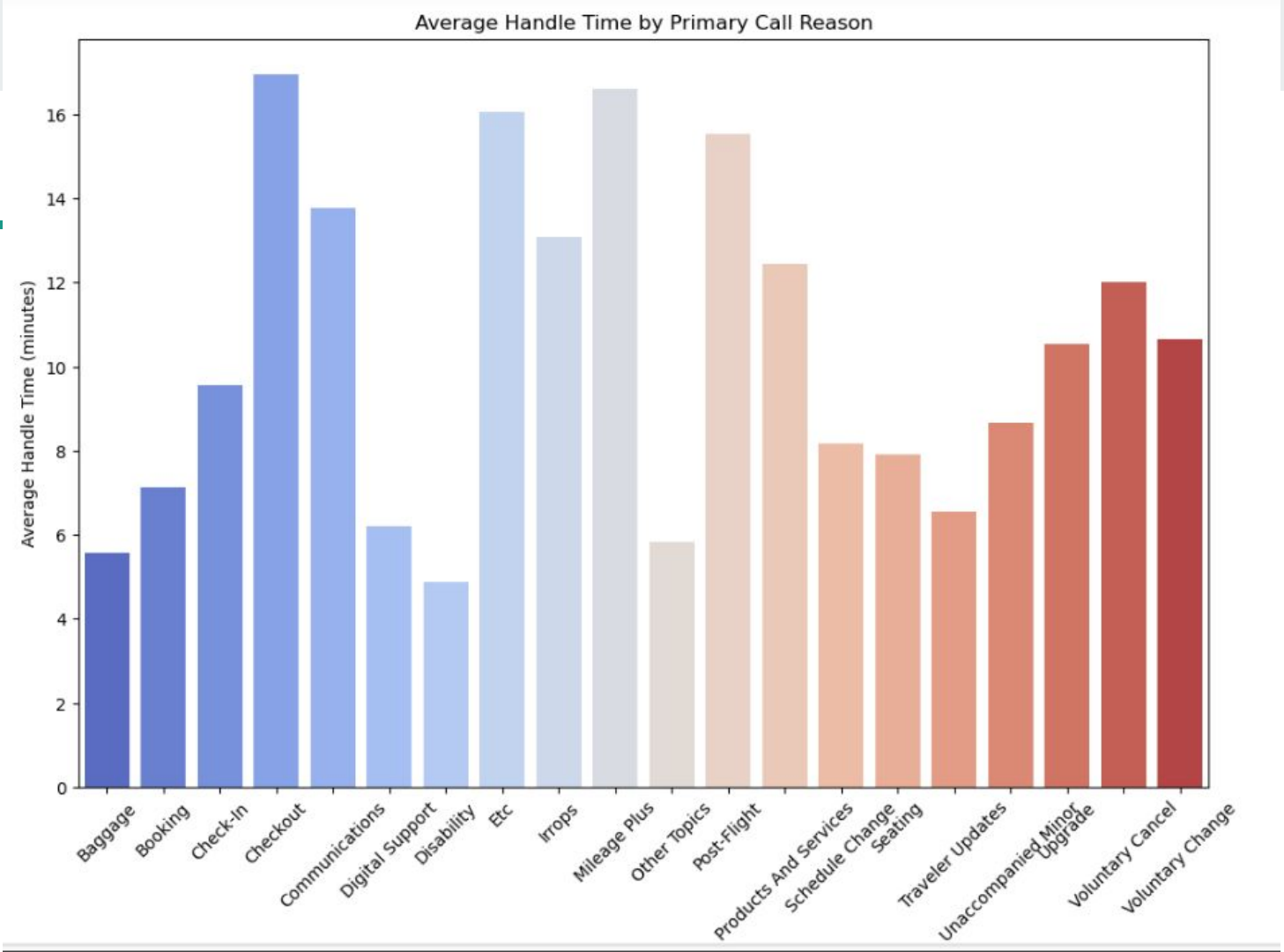


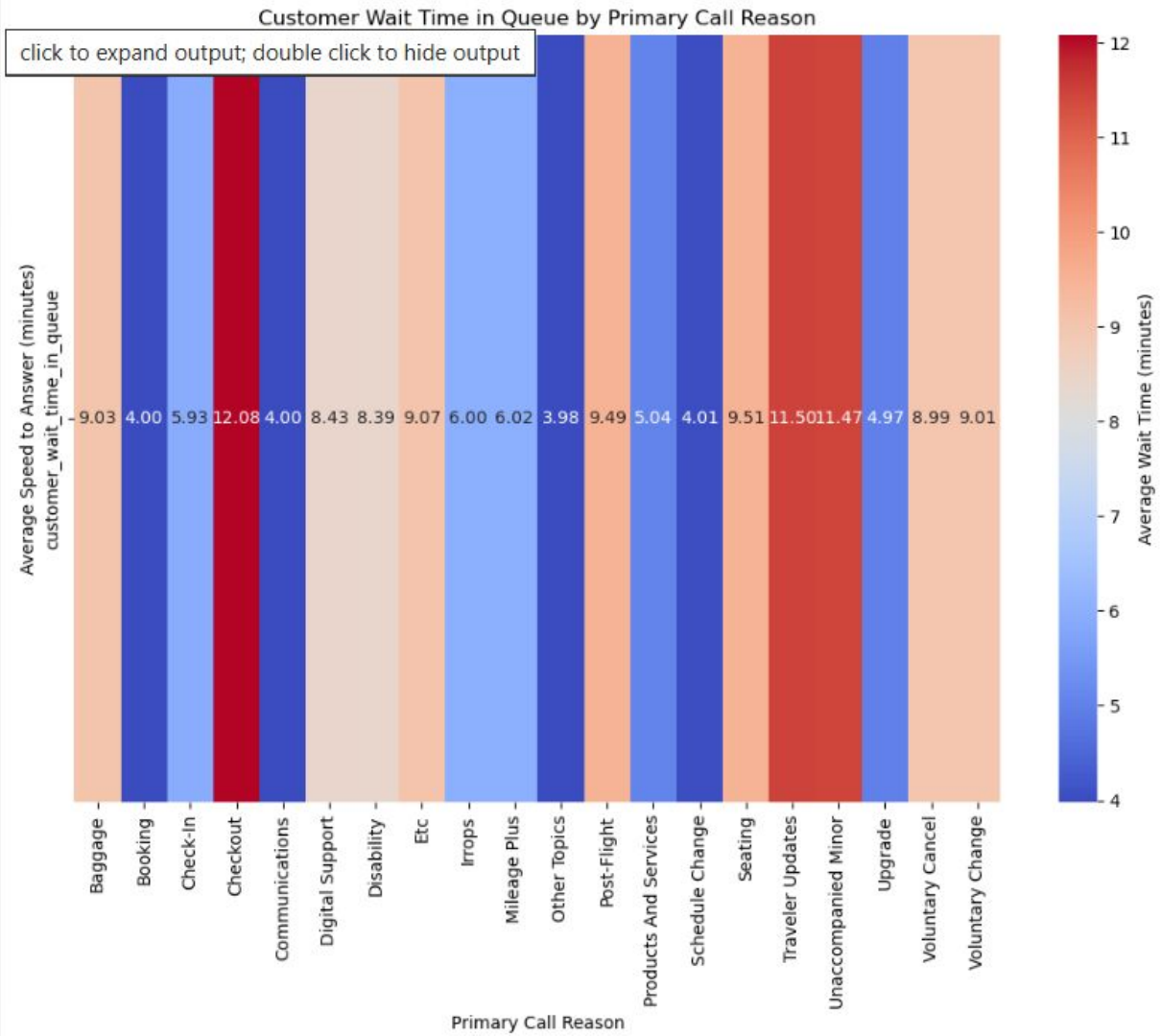

Agent Performance: AHT vs AST
Agent ID
150000
300000
450000
600000
750000
900000
Average AHT (seconds)
Average AST (seconds)
60
50
40
30
20
10
3
4
5
6
7
8
9
10


Impact of Call Reasons on AHT
average_handle_time
0
2
4
6
8
10
12
14
16
Baggage
Booking
Check-In
Checkout
Communications
Digital Support
Disability
Etc
Irrops
Mileage Plus
Other Topics
Post-Flight
Products And Services
Schedule Change
Seating
Traveler Updates
Unaccompanied Minor
Upgrade
Voluntary Cancel
Voluntary Change


Impact of Call Reasons on AST
customer_wait_time_in_queue
0
2
4
6
8
10
12
Baggage
Booking
Check-In
Checkout
Communications
Digital Support
Disability
Etc
Irrops
Mileage Plus
Other Topics
Post-Flight
Products And Services
Schedule Change
Seating
Traveler Updates
Unaccompanied Minor
Upgrade
Voluntary Cancel
Voluntary Change
primary_call_reason


Call Volume by Hour
Number of Calls
Hour of the Day
0
1000
2000
3000
4000
5000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23


Average Handle Time by Primary Call Reason
Average Handle Time (minutes)
0
2
4
6
8
10
12
14
16
Baggage
Booking
Check-In
Checkout
Communications
Digital Support
Disability
Etc
Irrops
Mileage Plus
Other Topics
Post-Flight
Products And Services
Schedule Change
Seating
Traveler Updates
Unaccompanied Minor
Upgrade
Voluntary Cancel
Voluntary Change


Customer Wait Time in Queue by Primary Call Reason
click to expand output; double click to hide output
Average Speed to Answer (minutes)
customer_wait_time_in_queue
9.03
4.00
5.93
12.08
4.00
8.43
8.39
9.07
6.00
6.02
3.98
9.49
5.04
4.01
9.51
11.50
11.47
4.97
8.99
9.01
Baggage
Booking
Check-In
Checkout
Communications
Digital Support
Disability
Etc
Irrops
Mileage Plus
Other Topics
Post-Flight
Products And Services
Schedule Change
Seating
Traveler Updates
Unaccompanied Minor
Upgrade
Voluntary Cancel
Voluntary Change
Primary Call Reason
Average Wait Time (minutes)
12
11
10
9
8
7
6
5
4

TASK 2

- We often observe self-solvable issues unnecessarily escalating to agents, increasing their workload. Analyse the transcripts and call reasons to identify granular reasons associated to recurring problems that could be resolved via self-service options in the IVR system. Propose specific improvements to the IVR options to effectively reduce agent intervention in these cases, along with solid reasoning to support your recommendations.
- To work on task 2 we had merged a new dataset called ‘2merged.xlsx' , it had all the necessary columns required to work on this task and all the unnecessary columns had been dropped and also the data in this dataset was cleaned.

Call transcript column is cleaned and a new column called cleaned transcript is created

In [2]:
```
import pandas as pd
dt=pd.read_excel('2merged.xlsx')
```

In [6]:
```
dt.head()
import re
```

In [7]:
```
pd.set_option('display.max_colwidth', None)  # or use a specific number for maximum width
```

In [8]:
```
def clean_transcript(transcript):
    # Remove newline characters and extra whitespace
    transcript = re.sub(r'\n+', ' ', transcript).strip()
    # Remove speaker identifiers
    transcript = re.sub(r'agent:\s*', '', transcript, flags=re.IGNORECASE)
    transcript = re.sub(r'customer:\s*', '', transcript, flags=re.IGNORECASE)
    # Normalize to lowercase
    transcript = transcript.lower()
    return transcript

# Apply the cleaning function to the 'call_transcript' column
dt['cleaned_transcript'] = dt['call_transcript'].apply(clean_transcript)

# Display the cleaned transcripts
print(dt[['cleaned_transcript']].head())
```

- To clean the transcripts by removing unnecessary characters, normalizing the text to lowercase, and optionally removing speaker identifiers like "Agent:" and "Customer:",

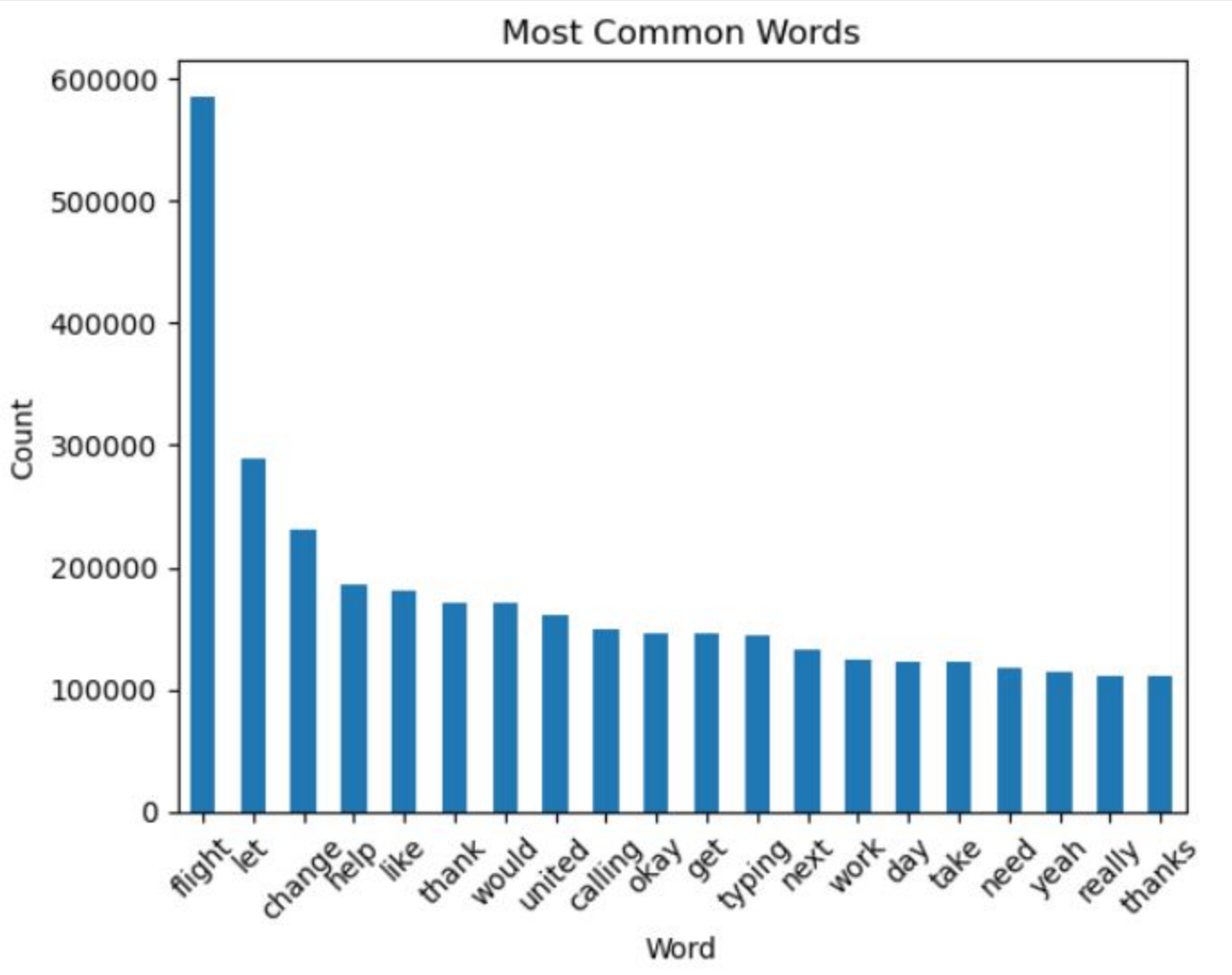
```
cleaned_transcript
0
thank you for calling united airlines customer service, my name is sarah how may i help you? hi, yeah i'm calling because my fl
ight from chicago to new york was delayed by over 3 hours! this is ridiculous, i'm missing important meetings because of this.
i'm so sorry to hear about the delay, that's definitely frustrating. umm, let me pull up your reservation and take a look at wh
at happened. *typing sounds* okay, it looks like there was severe weather in chicago that caused multiple flight cancellations
and delays across the board for all the airlines. ahh shoot, yeah your original flight was scheduled to depart at 2pm but did
n't actually take off until after 5pm. ugh this is such poor planning on united's part, you should have rerouted passengers or
put us on other flights that weren't delayed. now i've wasted a whole day.  i understand your frustration sir, delays are never
fun. let me see what options i have available to help make this up to you. hmm, it looks like i can get you booked on a flight
leaving at 3pm tomorrow that will have you landing in new york by 5pm. i'll waive the change fee and provide you with a travel
voucher for $200 to help cover expenses from the missed meetings today. does that work for your schedule? i guess that's better
than nothing. but you really need to get your act together, this is unacceptable customer service from united. your delays are
costing people time and money.  you're right, we want to do better for our customers. i apologize again for the delay and incon
venience. thank you for flying with us and i hope the rescheduled flight and travel credit help make up for some of the trouble
today. please feel free to reach back out if you have any other issues. thanks for your patience and for understanding - i appr
eciate you taking the time to work through this with me. alright fine. this better not happen again next time i fly united.
i'll be sure to note your record so we can try and avoid any repeats going forward. take care and have a good rest of your day.
yeah, you too. *hangs up* thank you for your call, next caller please. *pretends to answer another call* hi there, thank you fo
r calling united how can i help?
1
thank you for calling united airlines, my name is sam, how can i help you today? hi sam, yeah i'm calling because i need to cha
nge the dates on my upcoming flight. my flight is booked for next thursday but something came up and i need to fly out on monda
y instead. no problem, let me pull up your reservation so i can take a look. can i get your last name and confirmation number p
lease?  sure, last name is smith, confirmation number is asdf456. *makes noise in background* sorry, the kids are being loud ov
er here.  no worries, let me see... okay mr. smith, i've got your reservation here for flight ua128 next thursday. just giving
the availability a check for monday... umm it looks like we do have seats available on a flight leaving earlier that day. the f
are would be an additional $100 each way though. how does that work for you? ah man, an extra $100? that's more than i was hopi
```

In [10]:
```
dt.head()
```

	call_transcript	customer_tone	agent_tone	primary_call_reason	cleaned_transcript
	rlines customer service, my name is Sarah how may calling because my flight from Chicago to New York diculous, I'm missing important meetings because of ut the delay, that's definitely frustrating. Umm, let me look at what happened. Okay, it looks like there was ed multiple flight cancellations and delays across the n your original flight was scheduled to depart at 2pm pm.\n\nCustomer: Ugh this is such poor planning on ed passengers or put us on other flights that weren't \nAgent: I understand your frustration sir, delays are have available to help make this up to you. Hmm, it t leaving at 3pm tomorrow that will have you landing e fee and provide you with a travel voucher for $200 l meetings today. Does that work for your schedule? nothing. But you really need to get your act together, om United. Your delays are costing people time and to do better for our customers. I apologize again for u for flying with us and I hope the rescheduled flight e of the trouble today. Please feel free to reach back s. Thanks for your patience and for understanding - I	angry	neutral	Voluntary Cancel	thank you for calling united airlines customer service, my name is sarah how may i help you? hi, yeah i'm calling because my flight from chicago to new york was delayed by over 3 hours! this is ridiculous, i'm missing important meetings because of this. i'm so sorry to hear about the delay, that's definitely frustrating. umm, let me pull up your reservation and take a look at what happened. *typing sounds* okay, it looks like there was severe weather in chicago that caused multiple flight cancellations and delays across the board for all the airlines. ahh shoot, yeah your original flight was scheduled to depart at 2pm but didn't actually take off until after 5pm. ugh this is such poor planning on united's part, you should have rerouted passengers or put us on other flights that weren't delayed. now i've wasted a whole day. i understand your frustration sir, delays are never fun. let me see what options i have available to help make this up to you. hmm, it looks like i can get you booked on a flight leaving at 3pm tomorrow that will have you landing in new york by 5pm. i'll waive the change fee and provide you with a travel voucher for $200 to help cover expenses from the missed meetings today. does that work for your schedule? i guess that's better than nothing. but you really need to get your act together, this is unacceptable customer service from united. your delays are costing people time and money. you're right, we want to do better for our customers. i apologize again for the delay and inconvenience. thank you for flying with us and i hope the rescheduled flight and travel credit help make up for some of the trouble today. please feel free to reach back out if you have any other issues. thanks for your patience and for understanding - i appreciate you taking the time to work through this

Finding the most common words and their frequency (code is in the jupyter notebook file)

Most common words:
flight: 585915
let: 288452
change: 230728
help: 186627
like: 180177
thank: 171501
would: 170512
united: 160752
calling: 148492
okay: 145644
get: 145136
typing: 143928
next: 131920
work: 124399
day: 123347
take: 121946
need: 117330
yeah: 114338
really: 111302
thanks: 111187


Most Common Words
600000
500000
400000
300000
200000
100000
0
Count
flight
let
change
help
like
thank
would
united
calling
okay
get
typing
next
work
day
take
need
yeah
really
thanks
Word

Finding the contexts from the call transcript column

1. **Transcript Preprocessing**: The transcript was first combined into a single string, splitting it into individual sentences. A list of common keywords and phrases was defined to guide the search for relevant context within the text.
2. **Context Extraction**: For each keyword or phrase, the code scanned the transcript, identified sentences containing those words, and captured a range of sentences around them. The result was a set of contextually relevant sentences, providing insights into how those terms were used.

```python
common_words = [
    'flight', 'change', 'help', 'let', 'like', 'thank', 'would',
    'united', 'calling', 'okay', 'get', 'typing', 'next',
    'work', 'day', 'take', 'need', 'yeah', 'really', 'thanks',
    'change my flight', 'help with booking', 'reschedule',
    'delay', 'customer service', 'travel voucher', 'flight cancellation'
]

# Extract context for each common word
context_results = {word: extract_context(transcript, word) for word in common_words}

# Print the context for each common word
for word, contexts in context_results.items():
    print(f"Context for '{word}':")
    for context in contexts[:10]:
        print(f"- {context}")
    print()
```

```
Context for 'flight':
-
- please let me know if you have any other issues, and thank you for your patience today - i know it hasn't been the travel exp
erience you wanted
- yeah okay, switch me to the 4pm i guess
- hi, yeah umm i'm calling because i have a problem with my flight
- however, we may be able to work with you on applying the value of your ticket toward future travel
- i'm flying from chicago to los angeles on july 15th but i need to change it to the following week on july 22nd
- let me take a look at availability for later dates
- and i'll get you booked on an earlier flight in case that one is delayed, just as a backup
- as for a hotel and meals, due to the weather being the cause of the delay i unfortunately can't provide any vouchers for thos
e
- as for the car rental, let me call around to the rental agencies near lax and see if any of them can have a car ready for you
sooner when you land, just to give you some extra time
```

Context for 'change':
-
- the fare for tomorrow is about $50 higher than what you already paid
- please let me know if you have any other issues, and thank you for your patience today - i know it hasn't been the travel experience you wanted
- however, we may be able to work with you on applying the value of your ticket toward future travel
- i'm flying from chicago to los angeles on july 15th but i need to change it to the following week on july 22nd
- thanks for understanding - i hope you start feeling better soon
- let me take a look at availability for later dates
- i just can't believe you all changed my flight without even notifying me
- umm, ideally around 10 or 11am if there's a flight
- this is really last minute for me and it's out of my control that i need to change it

Context for 'help':
-
- i completely understand the stress of flight delays and risk of missing connections
- ugh i don't know, that's still not very helpful considering i had plans when i was supposed to land
- unfortunately weather is out of our control but i can assure you our crew is working very hard to minimize disruptions
- hi, yeah umm i'm calling because i have a problem with my flight
- thanks for understanding - i hope you start feeling better soon
- *notices short call time* that one went smoothly
- no, i think that addresses my question about the delay
- as for the car rental, let me call around to the rental agencies near lax and see if any of them can have a car ready for you sooner when you land, just to give you some extra time
- sorry for being frustrated before

Context for 'let':
-
- please let me know if you have any other issues, and thank you for your patience today - i know it hasn't been the travel experience you wanted
- yeah okay, switch me to the 4pm i guess
- however, we may be able to work with you on applying the value of your ticket toward future travel
- hi, yeah umm i'm calling because i have a problem with my flight
- i'm flying from chicago to los angeles on july 15th but i need to change it to the following week on july 22nd

Finding keywords which we can use to improve the IVR system

- **Tokenization and Stopword Removal**: The text was tokenized into individual words, and common stopwords (e.g., "and", "the", "is") were removed to focus on meaningful words. Only alphabetic tokens were retained.
- **N-gram Generation**: Bigrams (two-word combinations), trigrams (three-word combinations), and ten-grams (ten-word combinations) were generated from the filtered tokens.
- **Frequency Counting**: The code counted the occurrences of each bigram, trigram, and ten-gram, allowing for the identification of frequently used phrases within the customer service contexts.
- **Results Output**: The counts for bigrams and trigrams were printed, along with the most common ten-grams, providing insights into prevalent phrases and topics in the customer interactions.

```
Bigrams:
('please', 'let'): 4
('let', 'know'): 5
('know', 'issues'): 5
('issues', 'thank'): 4
('thank', 'patience'): 4
('patience', 'today'): 4
('today', 'know'): 4
('know', 'travel'): 4
('travel', 'experience'): 4
('experience', 'wanted'): 4
('wanted', 'yeah'): 2
('yeah', 'okay'): 5
('okay', 'switch'): 5
('switch', 'guess'): 5
('guess', 'hi'): 1
('hi', 'yeah'): 8
('yeah', 'umm'): 8
('umm', 'calling'): 8
('calling', 'problem'): 8
('problem', 'flight'): 8
('flight', 'however'): 1
('however', 'may'): 6
('may', 'able'): 6
('able', 'work'): 6
('work', 'applying'): 6
```

```
Trigrams:
('please', 'let', 'know'): 4
('let', 'know', 'issues'): 5
('know', 'issues', 'thank'): 4
('issues', 'thank', 'patience'): 4
('thank', 'patience', 'today'): 4
('patience', 'today', 'know'): 4
('today', 'know', 'travel'): 4
('know', 'travel', 'experience'): 4
('travel', 'experience', 'wanted'): 4
('experience', 'wanted', 'yeah'): 2
('wanted', 'yeah', 'okay'): 2
('yeah', 'okay', 'switch'): 5
('okay', 'switch', 'guess'): 5
('switch', 'guess', 'hi'): 1
('guess', 'hi', 'yeah'): 1
('hi', 'yeah', 'umm'): 8
('yeah', 'umm', 'calling'): 8
('umm', 'calling', 'problem'): 8
('calling', 'problem', 'flight'): 8
('problem', 'flight', 'however'): 1
('flight', 'however', 'may'): 1
('however', 'may', 'able'): 6
('may', 'able', 'work'): 6
('able', 'work', 'applying'): 6
('work', 'applying', 'value'): 6
('applying', 'value', 'ticket'): 6
('value', 'ticket', 'toward'): 6
```

BIGRAMS AND TRIGRAMS

```
ost Common 10-grams:
(('however', 'may', 'able', 'work', 'applying', 'value', 'ticket', 'toward', 'future', 'travel'), 6), (('flying', 'chicago',
los', 'angeles', 'july', 'need', 'change', 'following', 'week', 'july'), 6), (('please', 'let', 'know', 'issues', 'thank', 'pa
ience', 'today', 'know', 'travel', 'experience'), 4), (('let', 'know', 'issues', 'thank', 'patience', 'today', 'know', 'trave
', 'experience', 'wanted'), 4), (('hotel', 'meals', 'due', 'weather', 'cause', 'delay', 'unfortunately', 'ca', 'provide', 'vou
hers'), 4), (('car', 'rental', 'let', 'call', 'around', 'rental', 'agencies', 'near', 'lax', 'see'), 4), (('rental', 'let', 'c
ll', 'around', 'rental', 'agencies', 'near', 'lax', 'see', 'car'), 4), (('let', 'call', 'around', 'rental', 'agencies', 'nea
', 'lax', 'see', 'car', 'ready'), 4), (('call', 'around', 'rental', 'agencies', 'near', 'lax', 'see', 'car', 'ready', 'soone
'), 4), (('around', 'rental', 'agencies', 'near', 'lax', 'see', 'car', 'ready', 'sooner', 'land'), 4)]
```

- Context coding allows for the identification of specific phrases and terms frequently used by customers. By tagging the context of each interaction, we can capture recurring phrases that may not be immediately apparent without deeper analysis. N-grams:
- By analyzing bigrams, trigrams and 10 grams , we can identify frequently occurring phrases or terms related to specific issues. This helps in pinpointing common customer concerns that lead to agent intervention. Contextual Understanding:
- Analyzing the contexts in which these n-grams occur provides deeper insights into the circumstances surrounding customer inquiries. This allows for a granular understanding of issues, highlighting patterns in customer behavior and needs.
- Context coding involves annotating transcripts with additional information about the circumstances surrounding a call.

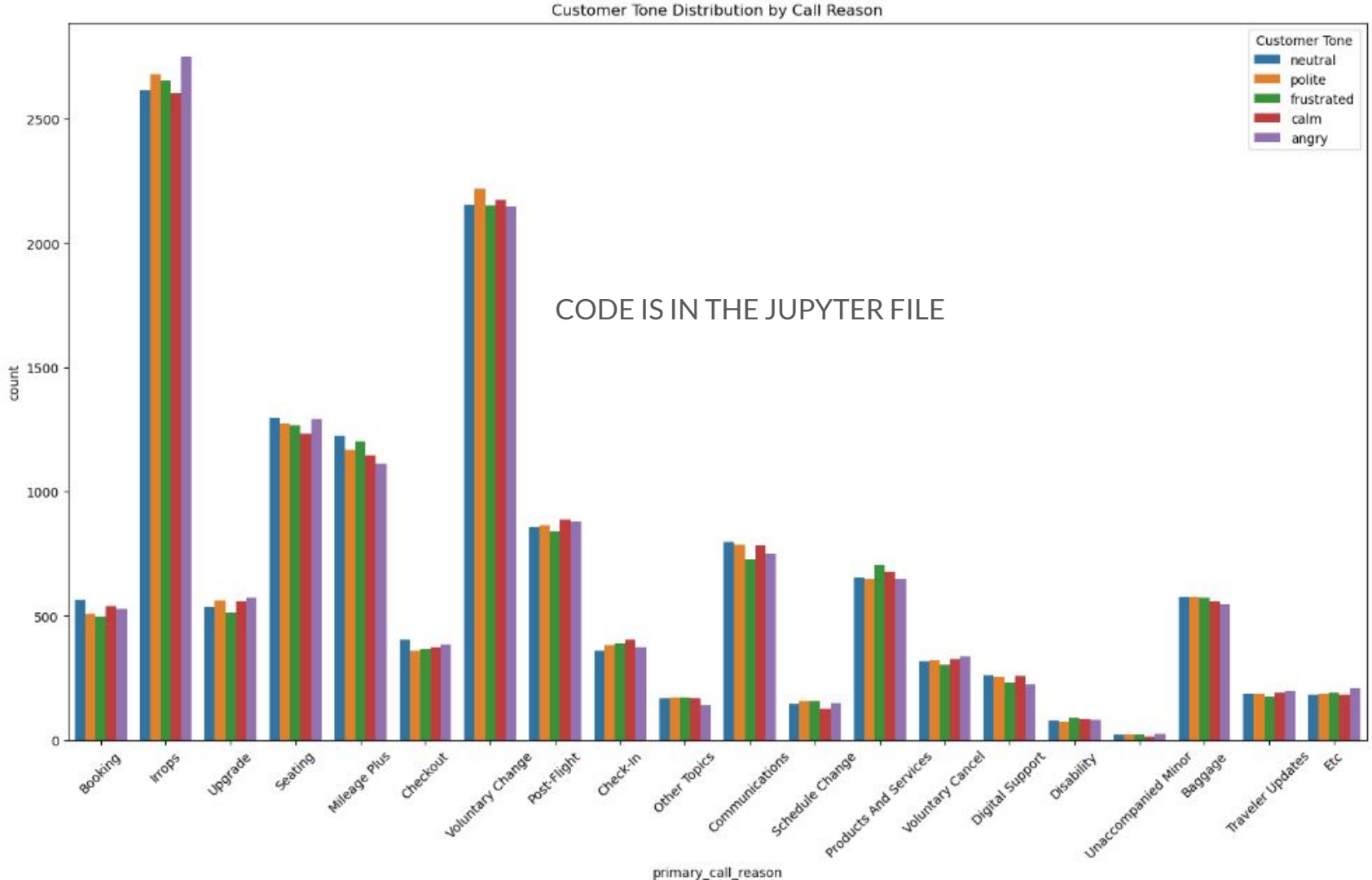
TASK 3

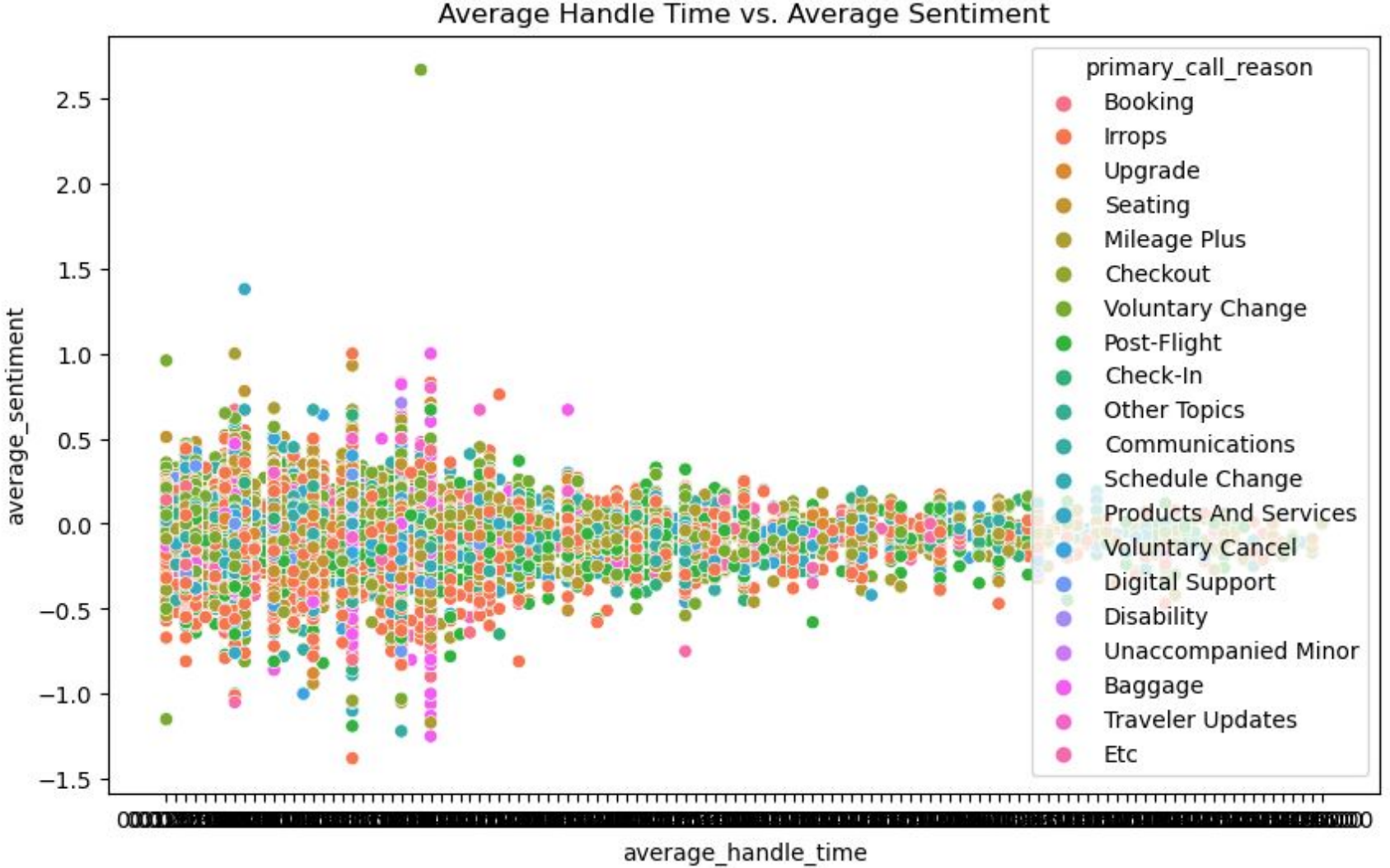
- Understanding the primary reasons for incoming calls is vital for enhancing operational efficiency and improving customer service. Accurately categorizing call reasons enables the call center to streamline processes, reduce manual tagging efforts, and ensure that customers are directed to the appropriate resources. In this context, analyze the dataset to uncover patterns that can assist in understanding and identifying these primary call reasons. Please outline your approach, detailing the data analysis techniques and feature identification methods you plan to use. *Optional task, you may utilize the `test.csv` file to generate and submit your predictions*
- For this task again we will use united_merged.csv file

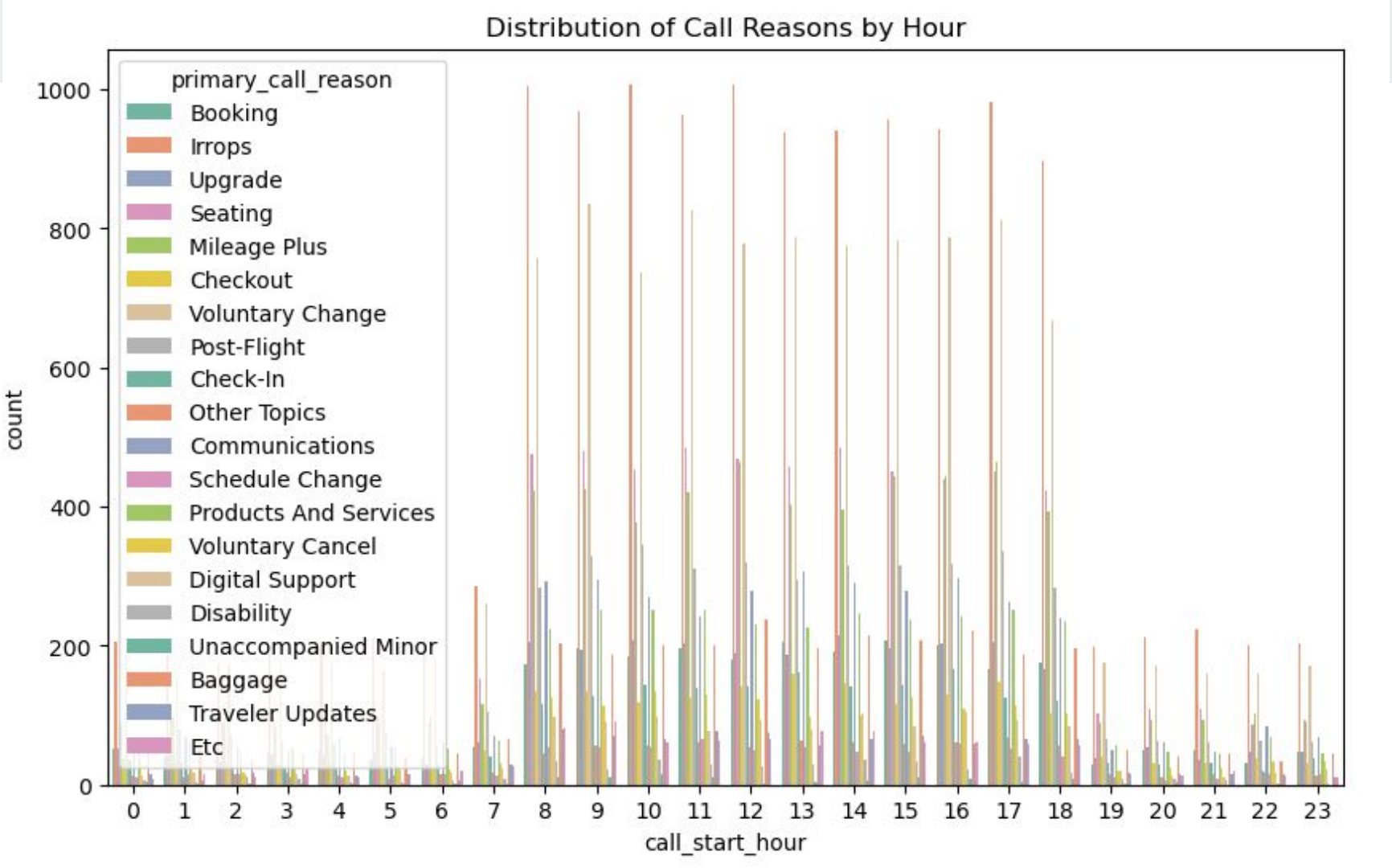
```
md=pd.read_csv('united_merged.csv')
```

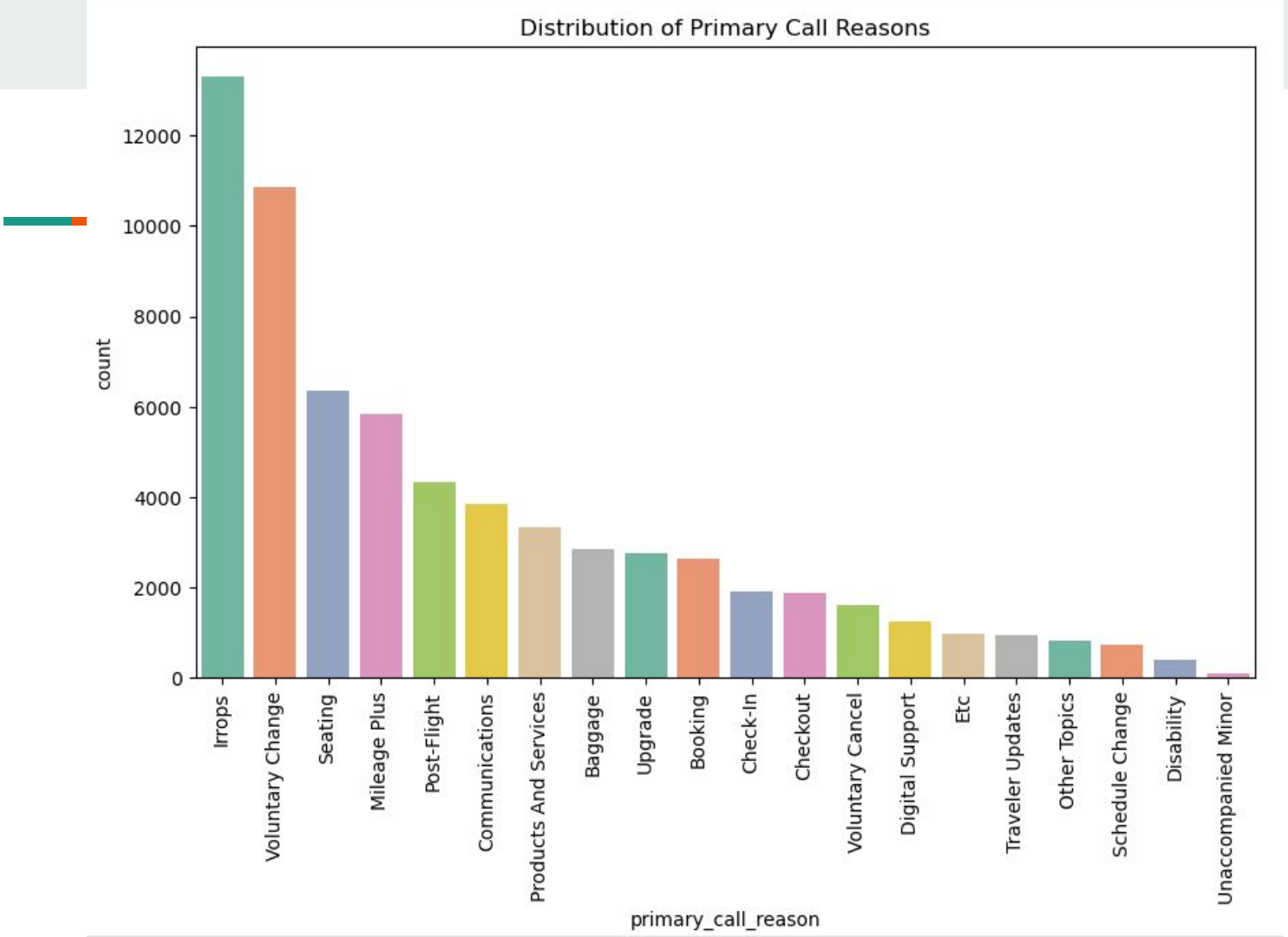
```
md.head()
```

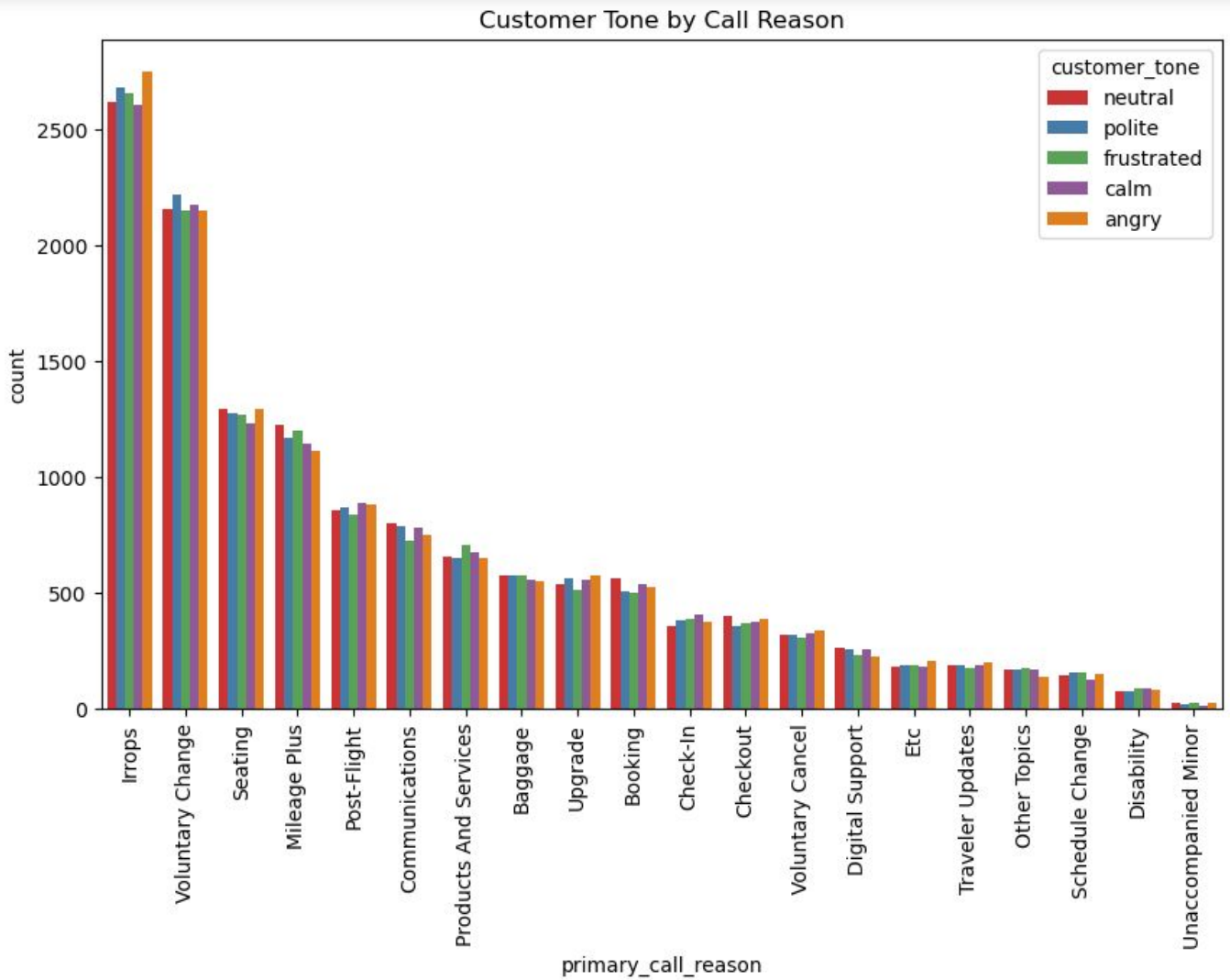
	call_id	primary_call_reason	customer_id	agent_id_x	call_start_datetime	agent_assigned_datetime	call_end_datetime	total_customer_duration	customer
0	1122072124	Booking	8186702651	519057	2024-08-01 00:03:00	2024-08-01 00:06:00	2024-08-01 00:18:00	00:15:00	
1	6834291559	Irrops	2416856629	158319	2024-07-31 23:59:00	2024-08-01 00:07:00	2024-08-01 00:26:00	00:27:00	
2	2266439882	Upgrade	1154544516	488324	2024-08-01 00:05:00	2024-08-01 00:10:00	2024-08-01 00:17:00	00:12:00	
3	1211603231	Seating	5214456437	721730	2024-08-01 00:04:00	2024-08-01 00:14:00	2024-08-01 00:23:00	00:19:00	
4	5297766997	Mileage Plus	5590154991	817160	2024-08-01 00:11:00	2024-08-01 00:16:00	2024-08-01 00:40:00	00:29:00	

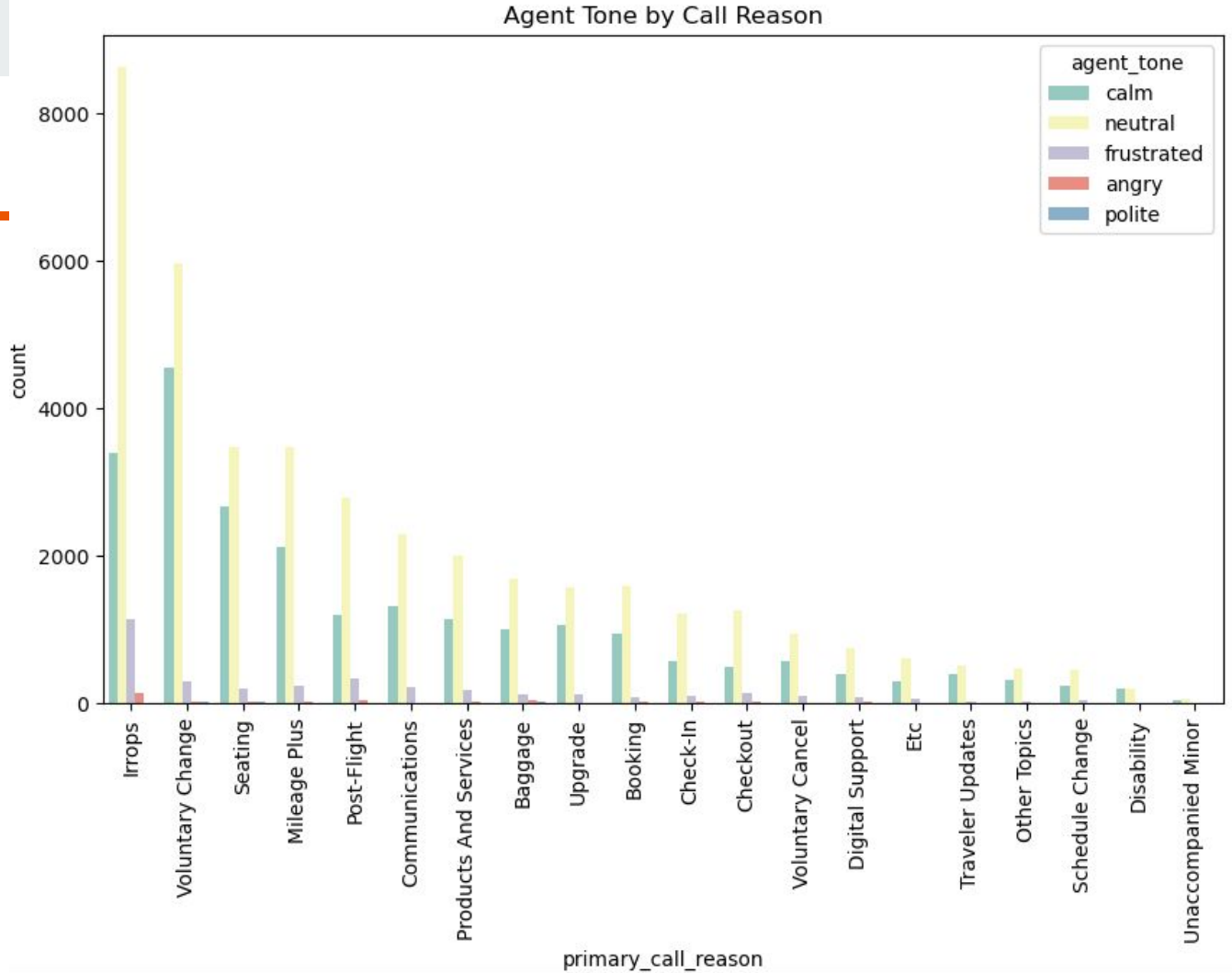

Customer Tone Distribution by Call Reason
Customer Tone
neutral
polite
frustrated
calm
angry
CODE IS IN THE JUPYTER FILE
count
0
500
1000
1500
2000
2500
Booking
Irrops
Upgrade
Seating
Mileage Plus
Checkout
Voluntary Change
Post-Flight
Check-In
Other Topics
Communications
Schedule Change
Products And Services
Voluntary Cancel
Digital Support
Disability
Unaccompanied Minor
Baggage
Traveler Updates
Etc
primary_call_reason

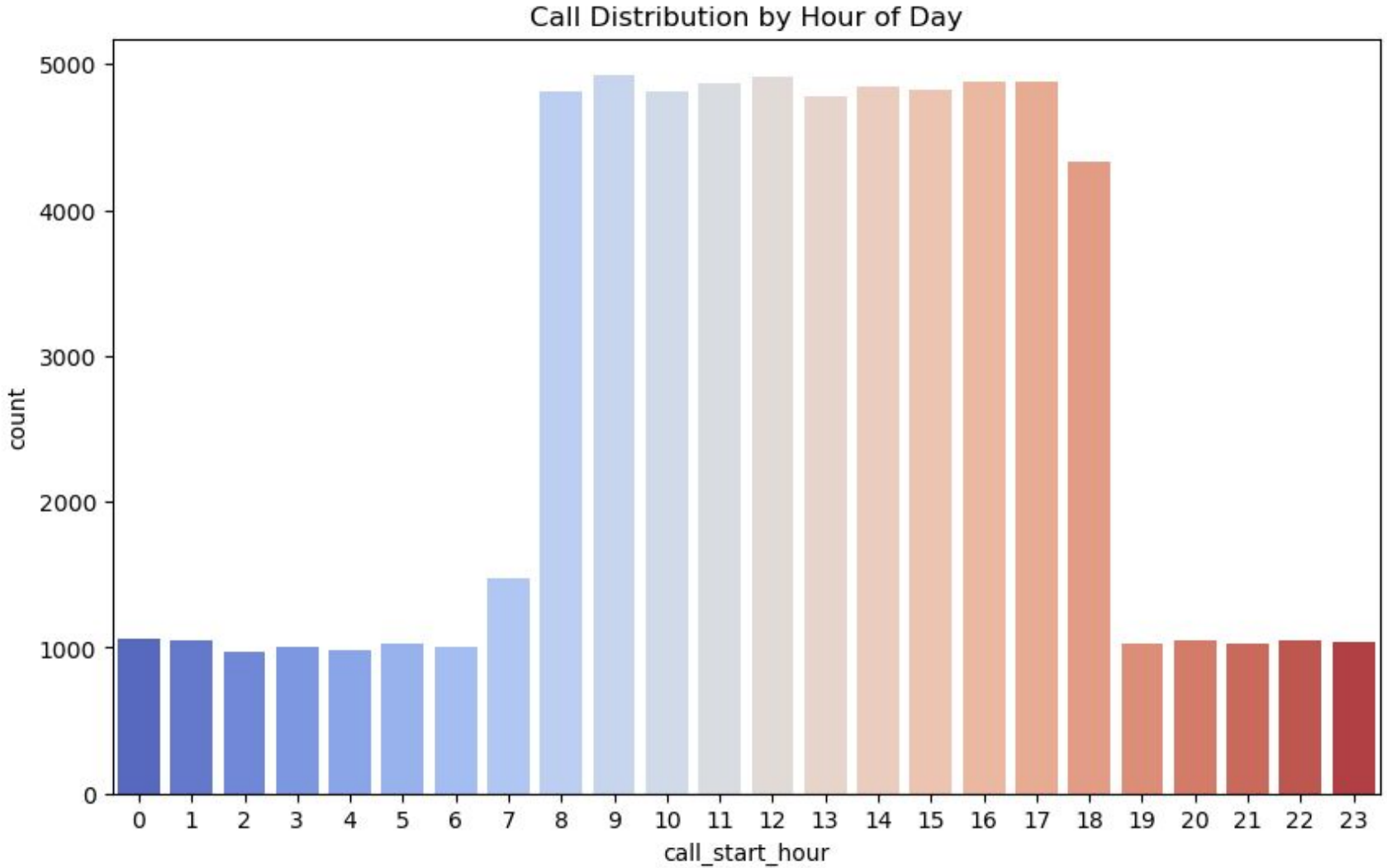

Average Handle Time vs. Average Sentiment
average_sentiment
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
average_handle_time
primary_call_reason
Booking
Irrops
Upgrade
Seating
Mileage Plus
Checkout
Voluntary Change
Post-Flight
Check-In
Other Topics
Communications
Schedule Change
Products And Services
Voluntary Cancel
Digital Support
Disability
Unaccompanied Minor
Baggage
Traveler Updates
Etc

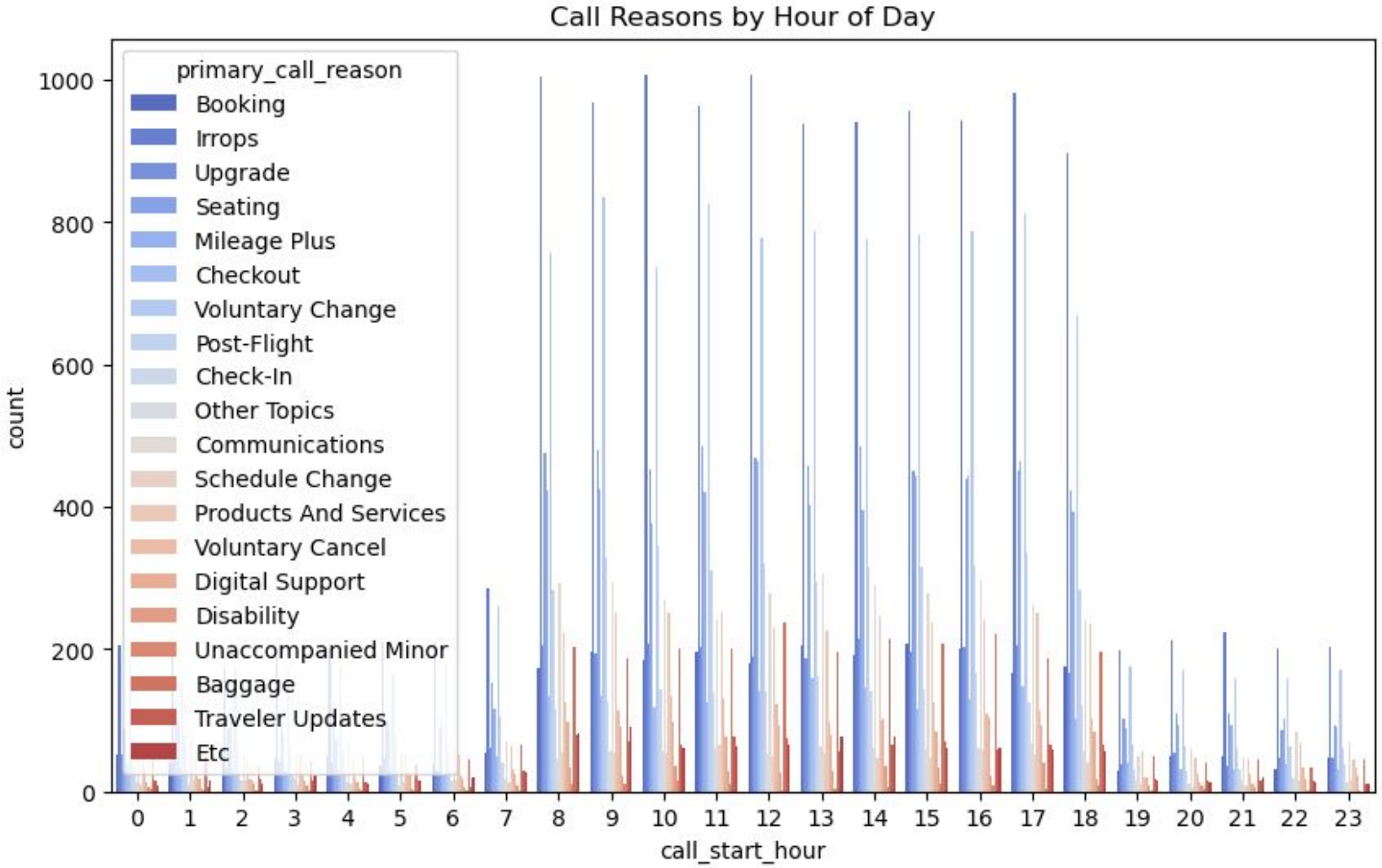

Distribution of Call Reasons by Hour
primary_call_reason
Booking
Irrops
Upgrade
Seating
Mileage Plus
Checkout
Voluntary Change
Post-Flight
Check-In
Other Topics
Communications
Schedule Change
Products And Services
Voluntary Cancel
Digital Support
Disability
Unaccompanied Minor
Baggage
Traveler Updates
Etc
count
0
200
400
600
800
1000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
call_start_hour

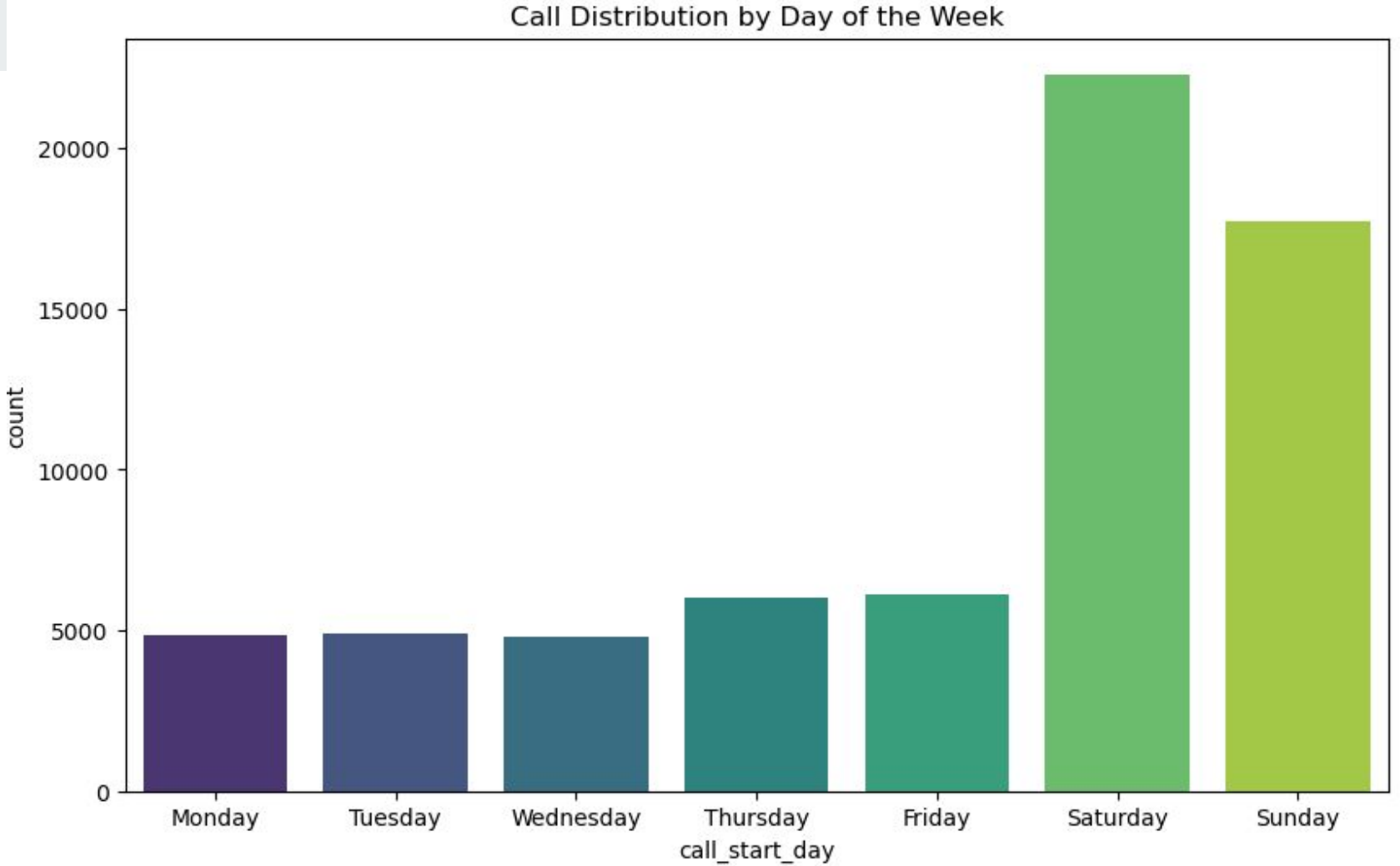

Distribution of Primary Call Reasons
count
12000
10000
8000
6000
4000
2000
0
Irrops
Voluntary Change
Seating
Mileage Plus
Post-Flight
Communications
Products And Services
Baggage
Upgrade
Booking
Check-In
Checkout
Voluntary Cancel
Digital Support
Etc
Traveler Updates
Other Topics
Schedule Change
Disability
Unaccompanied Minor
primary_call_reason

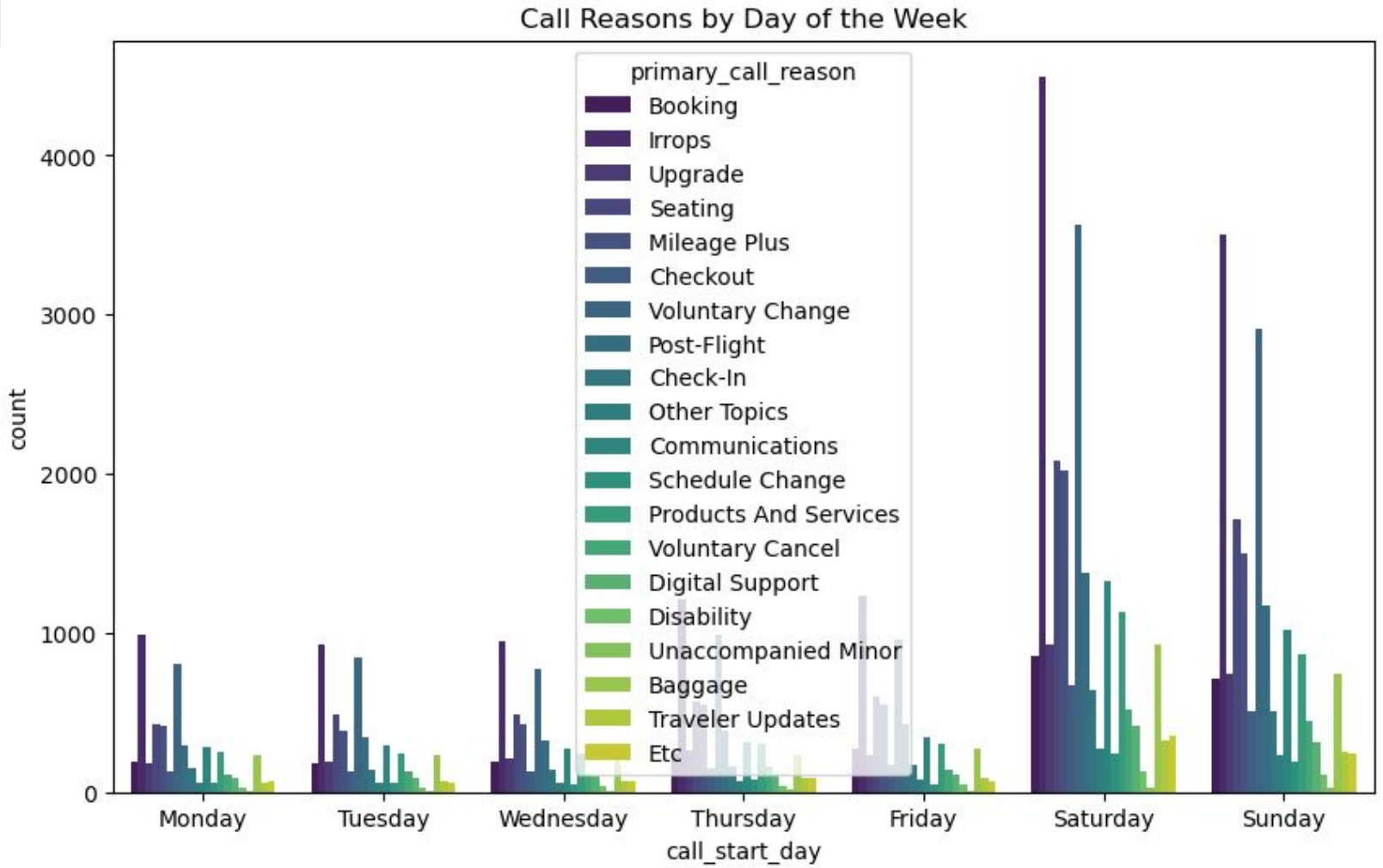

Customer Tone by Call Reason
customer_tone
neutral
polite
frustrated
calm
angry
count
0
500
1000
1500
2000
2500
Irrops
Voluntary Change
Seating
Mileage Plus
Post-Flight
Communications
Products And Services
Baggage
Upgrade
Booking
Check-In
Checkout
Voluntary Cancel
Digital Support
Etc
Traveler Updates
Other Topics
Schedule Change
Disability
Unaccompanied Minor
primary_call_reason

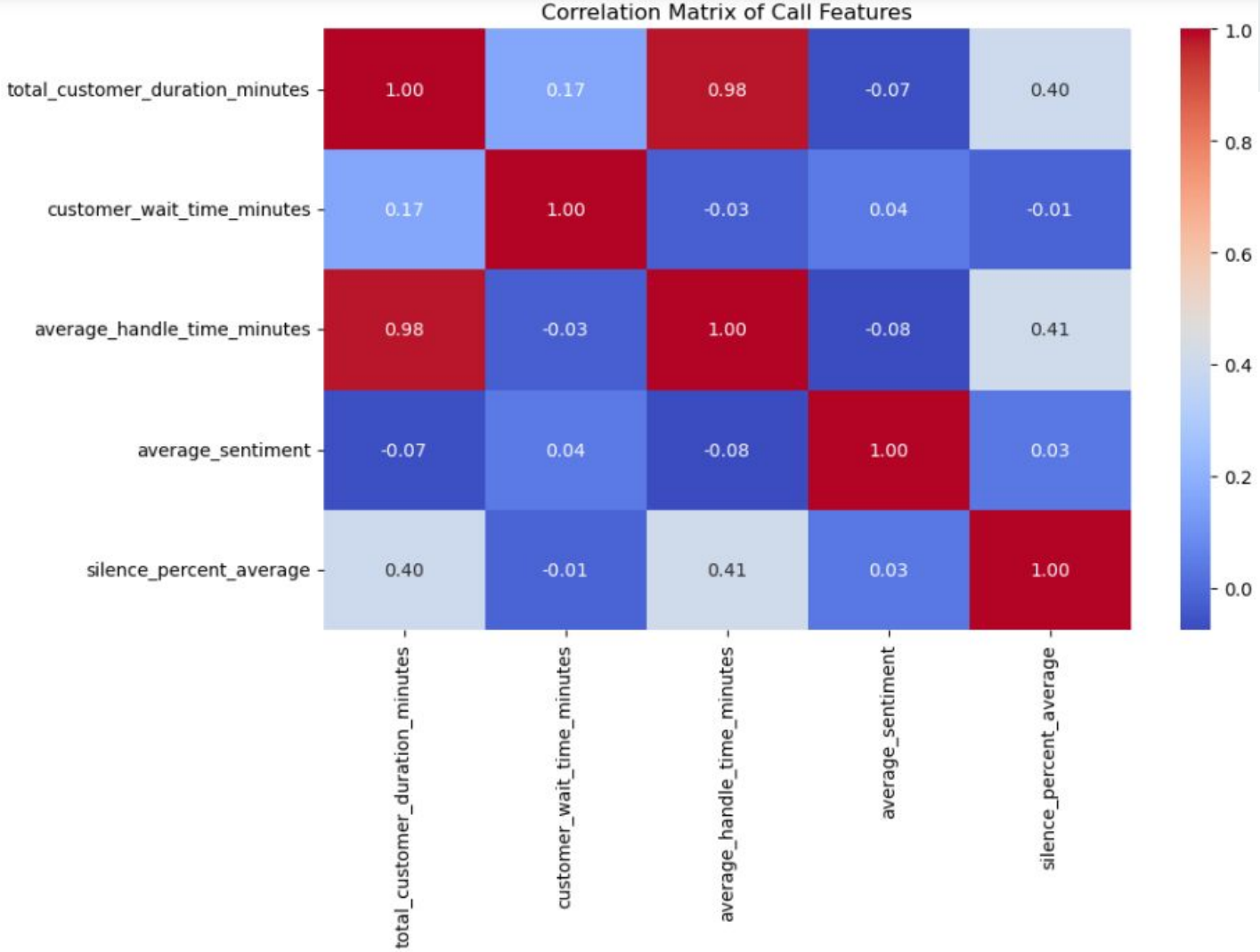

Agent Tone by Call Reason
agent_tone
calm
neutral
frustrated
angry
polite
count
0
2000
4000
6000
8000
Irrops
Voluntary Change
Seating
Mileage Plus
Post-Flight
Communications
Products And Services
Baggage
Upgrade
Booking
Check-In
Checkout
Voluntary Cancel
Digital Support
Etc
Traveler Updates
Other Topics
Schedule Change
Disability
Unaccompanied Minor
primary_call_reason

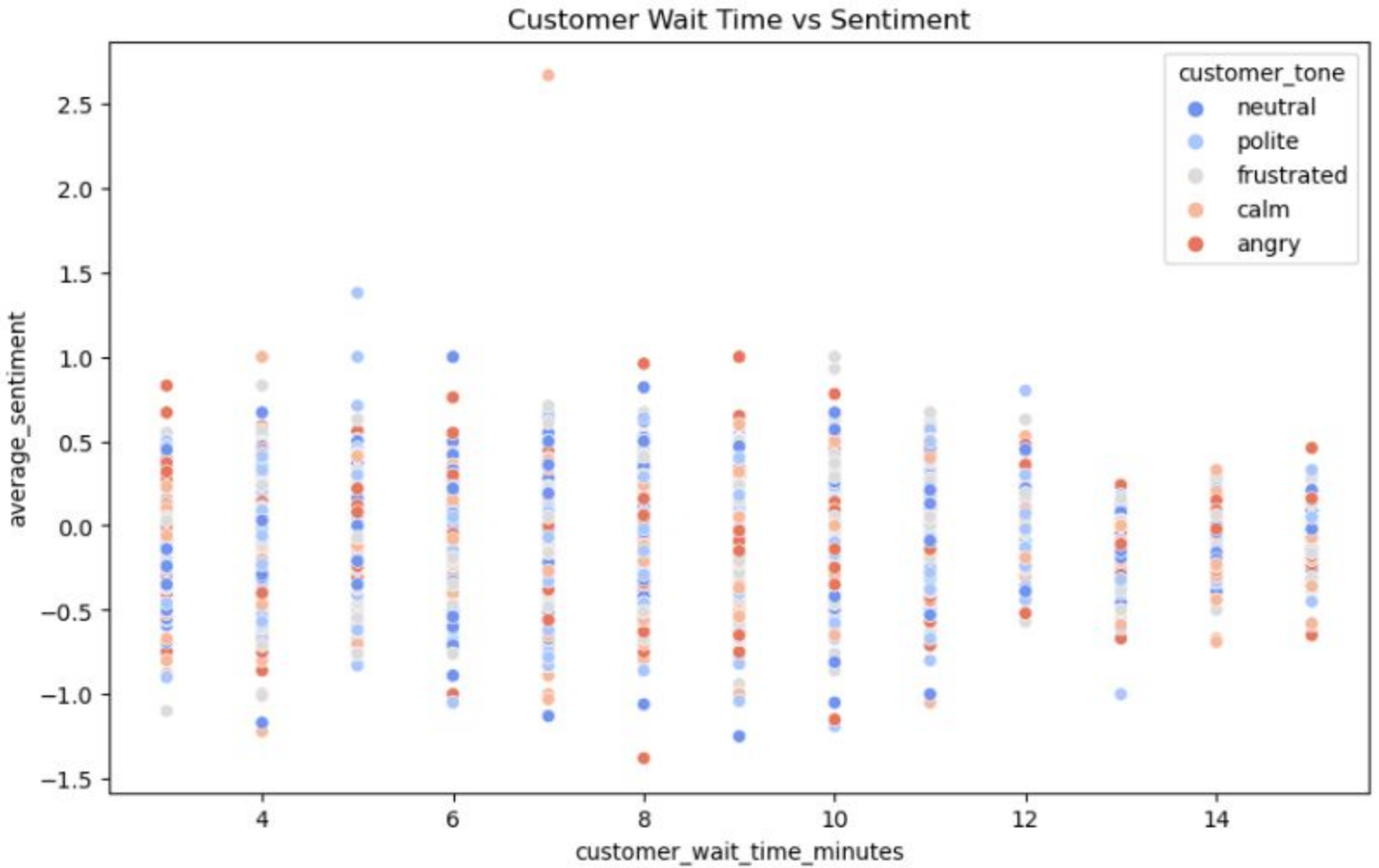

Call Distribution by Hour of Day
count
call_start_hour
0
1000
2000
3000
4000
5000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

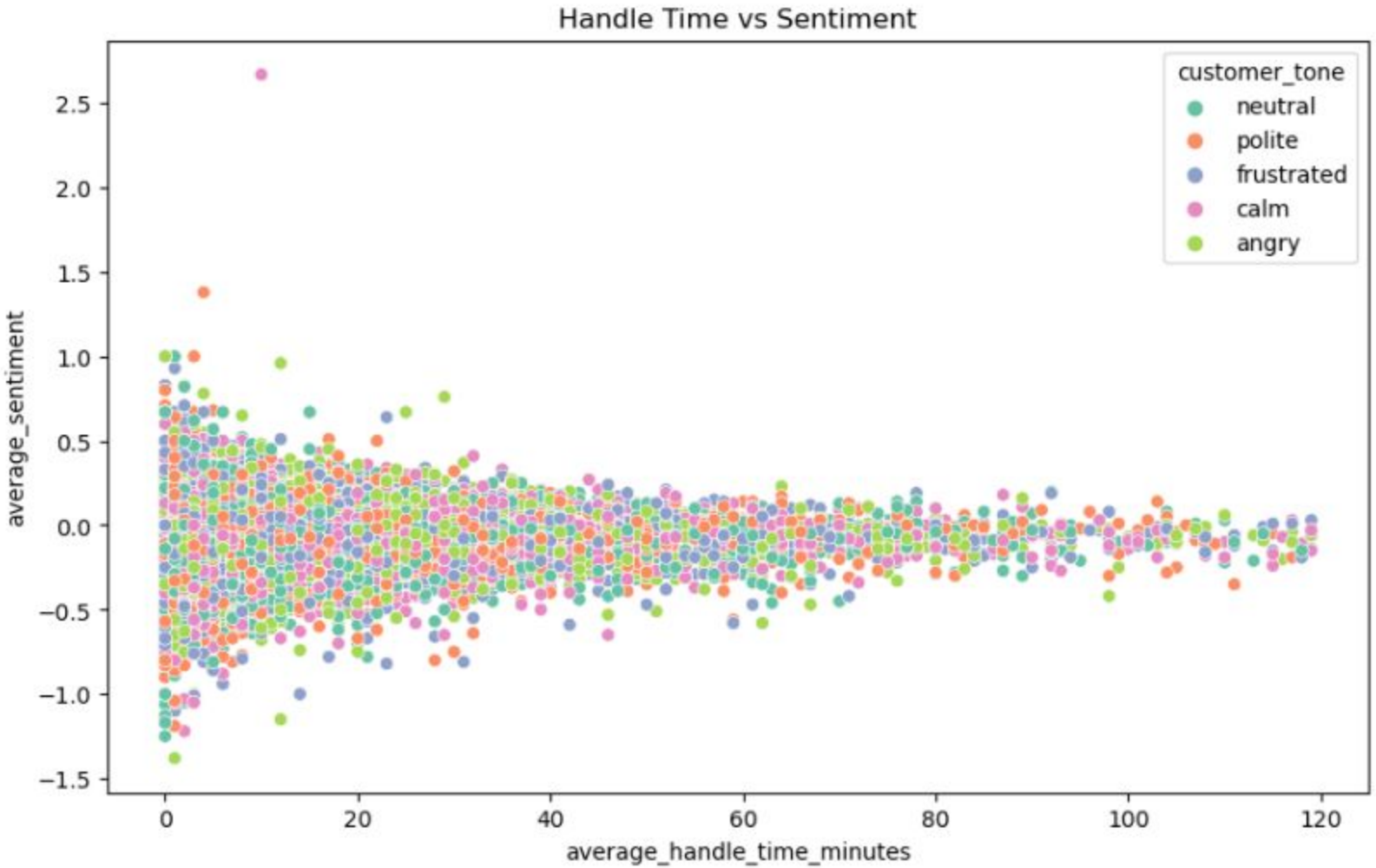

Call Reasons by Hour of Day
primary_call_reason
Booking
Irrops
Upgrade
Seating
Mileage Plus
Checkout
Voluntary Change
Post-Flight
Check-In
Other Topics
Communications
Schedule Change
Products And Services
Voluntary Cancel
Digital Support
Disability
Unaccompanied Minor
Baggage
Traveler Updates
Etc
count
0
200
400
600
800
1000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
call_start_hour

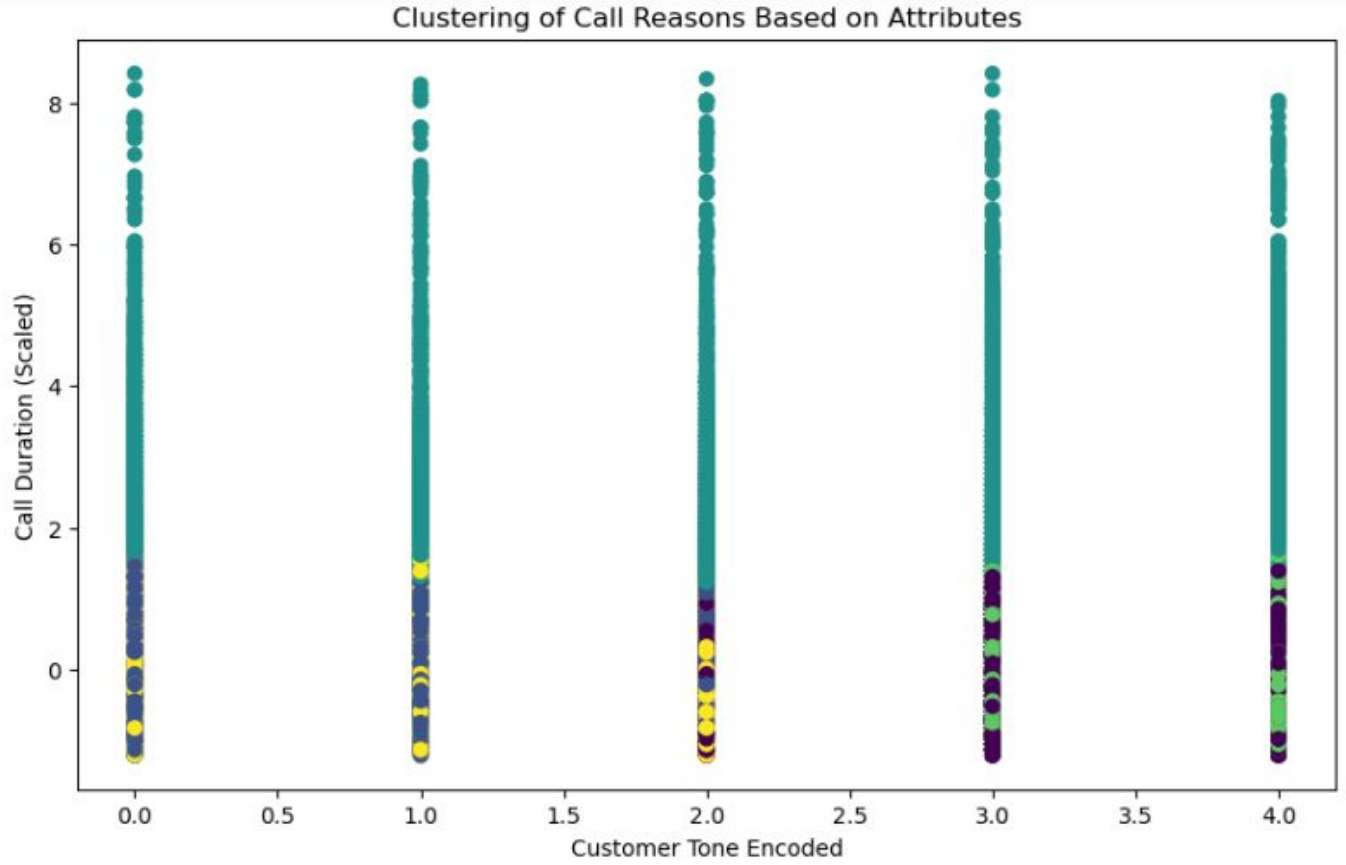

Call Distribution by Day of the Week
count
20000
15000
10000
5000
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
call_start_day


Call Reasons by Day of the Week
primary_call_reason
Booking
Irrops
Upgrade
Seating
Mileage Plus
Checkout
Voluntary Change
Post-Flight
Check-In
Other Topics
Communications
Schedule Change
Products And Services
Voluntary Cancel
Digital Support
Disability
Unaccompanied Minor
Baggage
Traveler Updates
Etc
count
0
1000
2000
3000
4000
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
call_start_day


Correlation Matrix of Call Features
total_customer_duration_minutes
customer_wait_time_minutes
average_handle_time_minutes
average_sentiment
silence_percent_average
1.00
0.17
0.98
-0.07
0.40
0.17
1.00
-0.03
0.04
-0.01
0.98
-0.03
1.00
-0.08
0.41
-0.07
0.04
-0.08
1.00
0.03
0.40
-0.01
0.41
0.03
1.00
1.0
0.8
0.6
0.4
0.2
0.0


Customer Wait Time vs Sentiment
customer_tone
neutral
polite
frustrated
calm
angry
average_sentiment
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
4
6
8
10
12
14
customer_wait_time_minutes


Handle Time vs Sentiment
customer_tone
neutral
polite
frustrated
calm
angry
average_sentiment
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
20
40
60
80
100
120
average_handle_time_minutes



	cluster
0	0
1	1
2	2
3	3
4	4