UNITED AIRLINES

SKYHACK

2.0

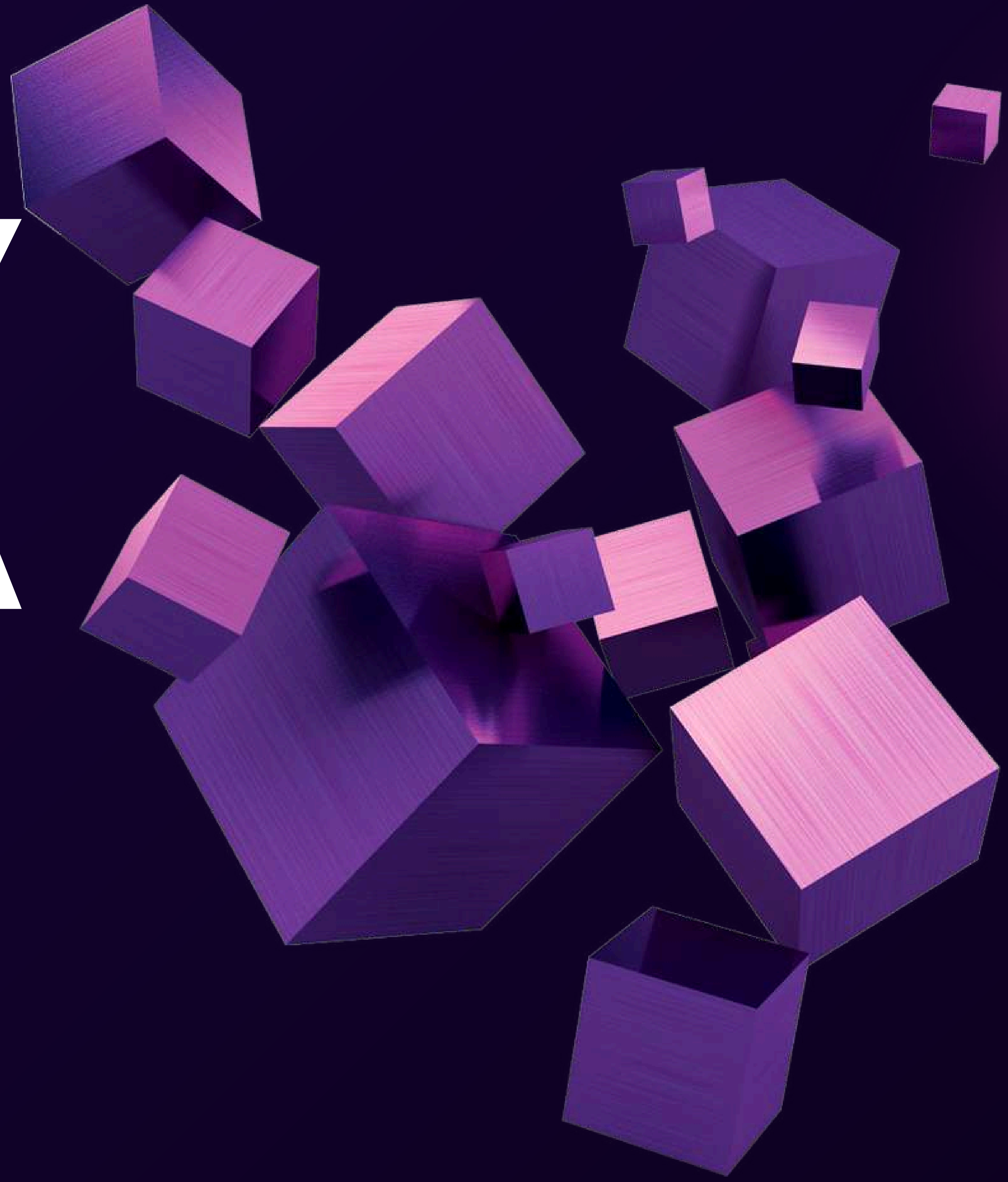
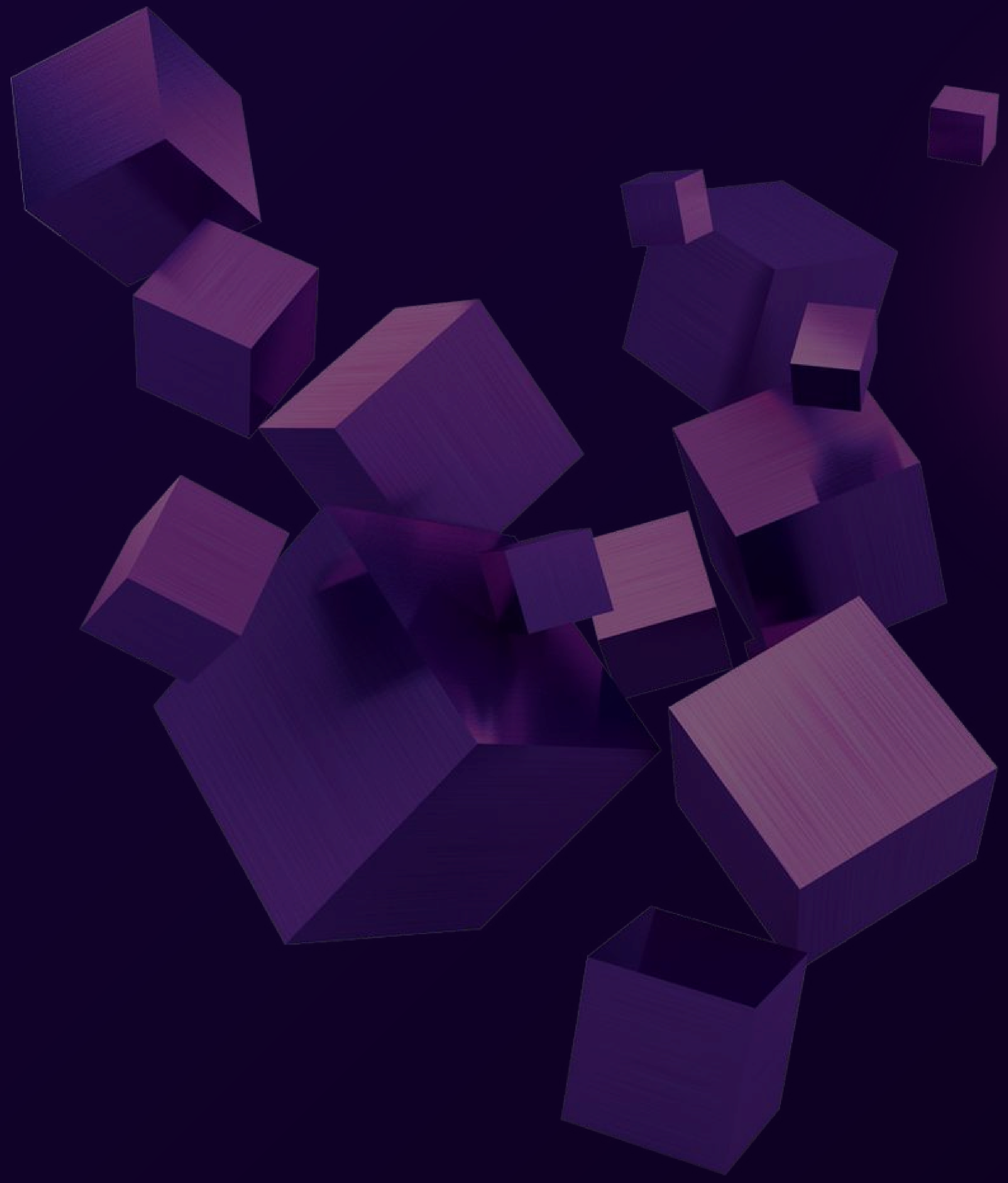


TABLE OF CONTENT

PROBLEM STATEMENT & BACKGROUND
DATA DESCRIPTION
DELIVERABLES
CONCEPTS USED
EXPLORATORY DATA ANALYSIS AND DATA-PREPROCESSING
ROOT CAUSE ANALYSIS
SENTIMENT ANALYSIS AND NLP MODELLING
ML MODELS TRAINING
OUTLIERS ANALYSIS
INSIGHTS
RECOMMENDATIONS



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).

PROBLEM STATEMENT

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.

BACKGROUND

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.

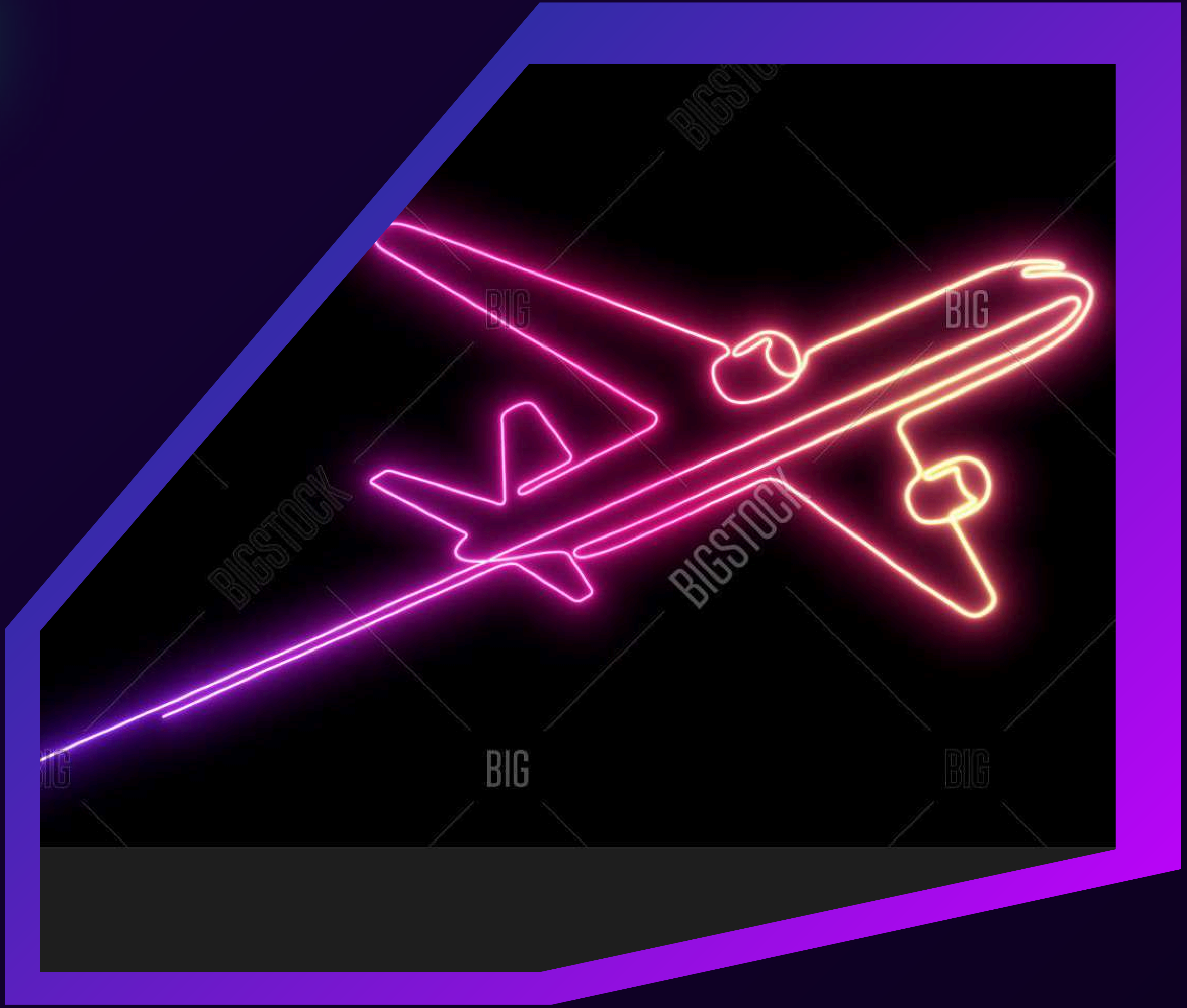
DATA DESCRIPTION

01 **CALLS**
CALLS-TIMELINES AND TRANSCRIPTS OF THE CALL

02 **CUSTOMERS**
ELITE LEVELS OF THE CUSTOMER

03 **REASONS**
PRIMARY REASON FOR THE CALL

04 **SENTIMENT STATISTICS**
AVERAGE SENTIMENTS AND AVERAGE SILENT PERCENTAGE



LINK TO THE DATASETS

DELIVERABLES

UNITED AIRLINES

01

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?

02

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.

03

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.

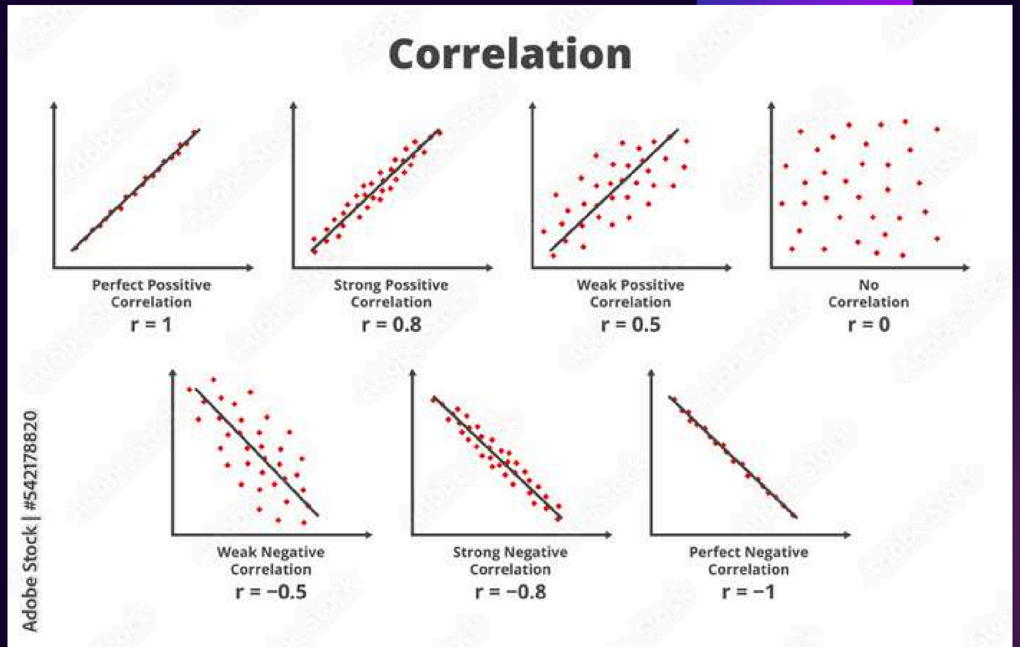
CONCEPTS

EXPLORATORY DATA ANALYSIS,DATA PREPROCESSING AND ROOT CAUSE ANALYSIS

Exploratory Data Analysis (EDA) is a critical step in the data analysis process, where analysts visually and statistically examine datasets to uncover patterns, anomalies, and insights. By using techniques such as data visualization and summary statistics, EDA helps identify trends and relationships within the data, guiding further analysis. Root Cause Analysis (RCA), on the other hand, is a method used to identify the underlying reasons for a problem or issue. By systematically investigating the causes of a problem, RCA helps organizations implement effective solutions and prevent recurrence, ultimately improving decision-making and operational efficiency. Together, EDA and RCA form a powerful approach to understanding data and driving informed actions.

NATURAL LANGUAGE PROCESSING

NLP, short for Natural Language Processing, would be that part of artificial intelligence concerned with enabling computers to be able to interact with humans in their language. It employs algorithms and models that enable machines to understand, interpret, and produce human language in a meaningful and contextually relevant sense. NLP empowers loads of applications-the change in how we interact with technology and then it turns out to be a means to process lots of textual data-in the form of chatbots, translation services, and sentiment analysis.



CORRELATION

Data correlation analysis is the study of the relationship between two or more variables to determine how closely they are associated with each other. This statistical method therefore shows the direction and strength of the relationship observed and sometimes runs on the principles of correlation coefficients such as Pearson's and Spearman's. If the correlation is positive, that is, when the value of the first variable increases, then the value of the second variable also tends to increase. However, a negative correlation suggests an inverse relationship, where an increase in the one variable results in a decrease of the other.

MERGING ALL DATSETS INTO A SINGLE CSV FILE

```
[2]: calls = pd.read_csv('calls.csv')
     customers = pd.read_csv('customers.csv')
     reason = pd.read_csv('reason.csv')
     senti_stats = pd.read_csv('sentiment_statistics.csv')
```

Merging the calls and customers csv ¶

```
[3]: calls_customers = calls.merge(customers,how='inner',on='customer_id')
```

Merging reason.csv with calls_customers

```
[4]: df = calls_customers.merge(reason,how='left',on='call_id')
```

Merging all the csv files together

```
[5]: data = df.merge(senti_stats, how='inner',on='call_id')
```

```
[6]: data.isnull().sum()
```

```
[6]: call_id                        0
     customer_id                    0
     agent_id_x                     0
     call_start_datetime            0
     agent_assigned_datetime        0
     call_end_datetime              0
     call_transcript                0
     customer_name                  0
     elite_level_code           25767
     primary_call_reason         5157
     agent_id_y                     0
     agent_tone                   217
     customer_tone                  0
     average_sentiment            109
     silence_percent_average        0
     dtype: int64
```

EXPLORATORY DATA ANALYSIS (EDA)

01

Understanding data

02

Checking for missing data

03

Clubbing columns with multiple categories

04

Checking for outliers

05

Label encoding

MISSING VALUE HANDLING & ONE-HOT ENCODING

```python
# Fill missing values for 'agent_tone' with mode
data['agent_tone'].fillna(data['agent_tone'].mode(), inplace=True)
```

```python
# Drop rows where 'primary_call_reason' (target) is missing
data.dropna(subset=['primary_call_reason'], inplace=True)
```

```python
# Handling missing values
# Impute 'elite_level_code' with mode (most frequent value)
data['elite_level_code'].fillna(data['elite_level_code'].mode()[0], inplace=True)
```

```python
# Fill missing values for 'average_sentiment' with mode
data['average_sentiment'].fillna(data['average_sentiment'].mean(), inplace=True)
```

```python
# Convert categorical columns (agent_tone, customer_tone) into one-hot encoding
categorical_cols = ['agent_tone', 'customer_tone']
data = pd.get_dummies(data, columns=categorical_cols, drop_first=True)
```

BEFORE

```python
data.isnull().sum()
```

```
call_id                      0
customer_id                  0
agent_id                     0
call_start_datetime          0
agent_assigned_datetime      0
call_end_datetime            0
call_transcript              0
customer_name                0
elite_level_code         25767
primary_call_reason       5157
agent_tone                 217
customer_tone                0
average_sentiment          109
silence_percent_average      0
waiting_time                 0
handling_time                0
dtype: int64
```

AFTER

```python
# Check if there are any remaining missing values
data.isnull().sum()
```

```
call_id                    0
customer_id                0
agent_id                   0
call_start_datetime        0
agent_assigned_datetime    0
call_end_datetime          0
call_transcript            0
customer_name              0
elite_level_code           0
primary_call_reason        0
agent_tone                 0
customer_tone              0
average_sentiment          0
silence_percent_average    0
waiting_time               0
handling_time              0
dtype: int64
```

OUTLIERS HANDLING

```
# IQR calculation for handling time
Q1 = data['handling_time'].quantile(0.25)
Q3 = data['handling_time'].quantile(0.75)
IQR = Q3 - Q1
```

```
IQR
```

```
11.0
```

```
#Extracting the outliers from the dataframe
outliers = data[(data['handling_time'] < (Q1 - 1.5*IQR)) | (data['handling_time'] > (Q3 + 1.5*IQR))]
```

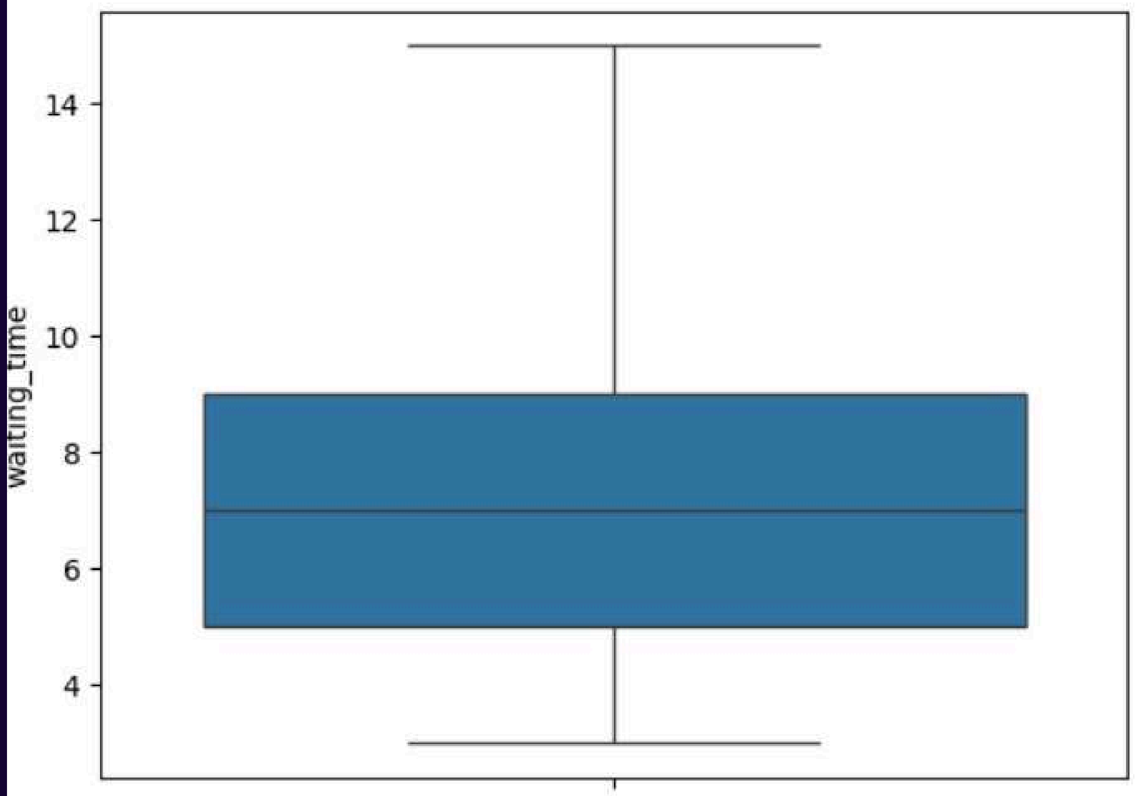
```
data = data.drop(outliers.index,axis=0)
```

The outliers in the feature handling_time are removed from the data using IQR range method.

WAITING TIME HAS NO OUTLIERS

```
ns.boxplot(y='waiting_time',data=data)
```

<Axes: ylabel='waiting_time'>

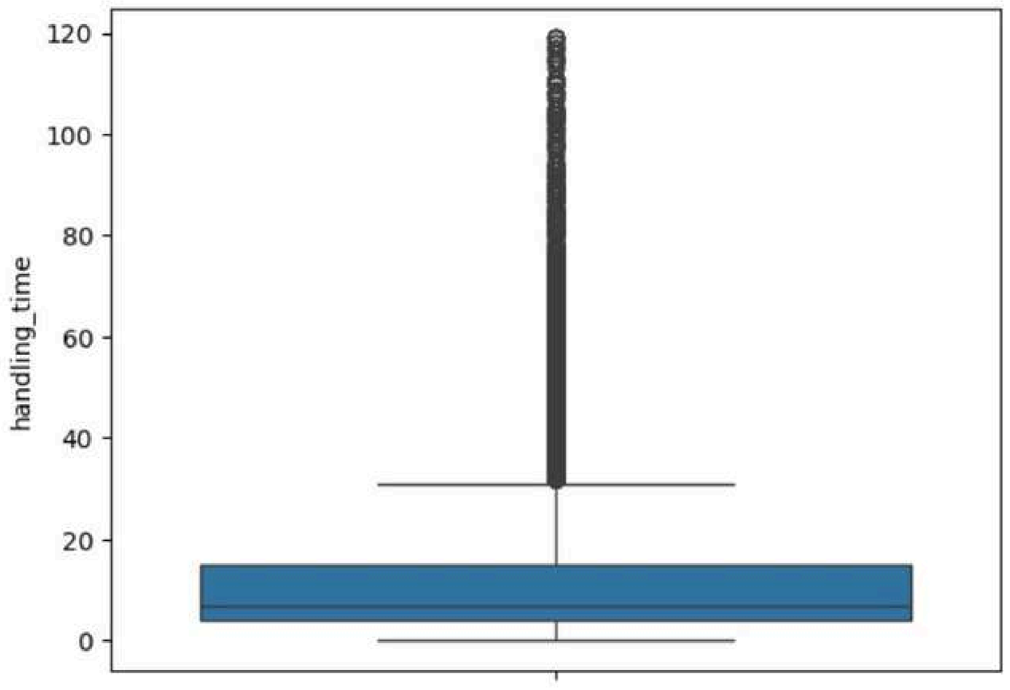


HANDLING TIME DATA HAS A LOT OF OUTLIERS

BEFORE

```
sns.boxplot(y='handling_time',data=data)
```

<Axes: ylabel='handling_time'>

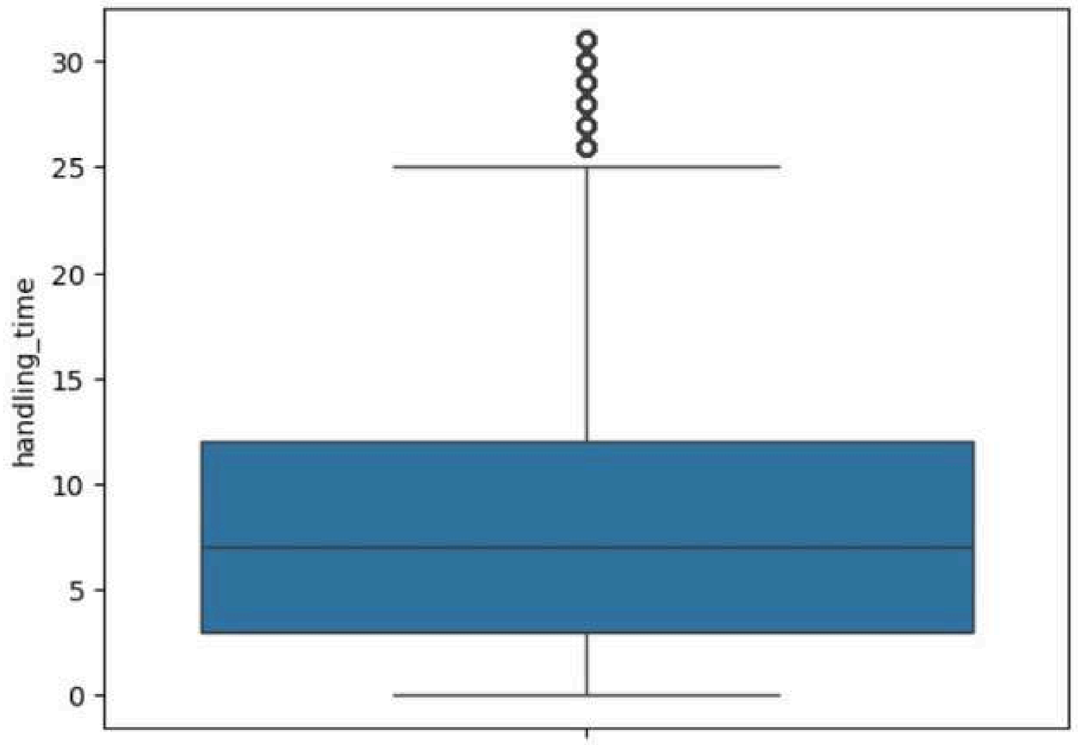


Handling time has a lot of outliers.

AFTER

```
sns.boxplot(y='handling_time',data=data)
```

<Axes: ylabel='handling_time'>



AHT(AVERAGE HANDLING TIME) & AST(AVERAGE SPEED TO ANSWER)

```
#Computing Total waiting time
data['call_start_datetime'] = pd.to_datetime(data['call_start_datetime'])
data['agent_assigned_datetime'] = pd.to_datetime(data['agent_assigned_datetime'])
```

```
#Time difference calculation
data['time_difference'] = data['agent_assigned_datetime'] - data['call_start_datetime']
```

```
#Converting time in minutes
data['waiting_time'] = data['time_difference'].dt.total_seconds() / 60
```

```
data = data.drop('time_difference',axis=1)
```

```
#AST (Average Speed to Answer):
#Time spent by the customer in queue till the agent answers the call
AST = sum(data['waiting_time'])/calls.shape[0]
```

```
AST
```

```
7.284458988998747
```

The Average Speed to Answer Calls is approximately 7.284 minutes.

```
#Computing Total handling time
data['agent_assigned_datetime'] = pd.to_datetime(data['agent_assigned_datetime'])
data['call_end_datetime'] = pd.to_datetime(data['call_end_datetime'])
```

```
#Time difference calculation
data['time_difference'] = data['call_end_datetime'] - data['agent_assigned_datetim
```

```
#Converting time in minutes
data['handling_time'] = data['time_difference'].dt.total_seconds() / 60
```

```
data = data.drop('time_difference',axis=1)
```

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

```
AHT = sum(data['handling_time'])/calls.shape[0]
```

```
AHT
```

```
11.61747667455786
```

BEFORE REMOVING OUTLIERS;

AHT = 11.617 MINUTES

AST= 7.284 MINUTES

AVERAGE CALL DURATION = 18.902 MINUTES

AFTER REMOVING OUTLIERS;

AHT = 8.863 MINUTES

AST= 7.284 MINUTES

AVERAGE CALL DURATION = 16.152 MINUTES

```
count    61953.000000
mean         8.862832
std          7.144852
min          0.000000
25%          3.000000
50%          7.000000
75%         12.000000
max         31.000000
Name: handling_time, dtype: float64
```

The Average call handling time(after removing outliers) is approximately 8.863 minutes.

The Average call duration time(after removing outliers) is approximately 16.152 minutes.

```
Average_call_duration = AHT+AST
```

```
Average_call_duration
```

```
18.901935663556607
```

The Average call duration time is approximately 18.902 minutes.

```python
# Calculate Percentage Difference Between AHT for Most and Least Frequent Call Reasons
# Get AHT for each call_reason
aht_by_reason = data.groupby('primary_call_reason')['handling_time'].mean().reset_index()
most_frequent_reason = data['primary_call_reason'].value_counts().idxmax()
least_frequent_reason = data['primary_call_reason'].value_counts().idxmin()
```

```python
# Calculate the AHT for the most and least frequent call reasons
most_frequent_aht = aht_by_reason[aht_by_reason['primary_call_reason'] == most_frequent_reason]['handling_time'].values[0]
least_frequent_aht = aht_by_reason[aht_by_reason['primary_call_reason'] == least_frequent_reason]['handling_time'].values[0]
```

```python
# Calculate percentage difference
percentage_difference = ((most_frequent_aht - least_frequent_aht) / least_frequent_aht) * 100
```

```python
print(f"Average Handling Time for Most Frequent Call Reason ({most_frequent_reason}): {most_frequent_aht:.2f}")
print(f"Average Handling Time for Least Frequent Call Reason ({least_frequent_reason}): {least_frequent_aht:.2f}")
print(f"Percentage Difference in AHT: {percentage_difference:.2f}%")
```

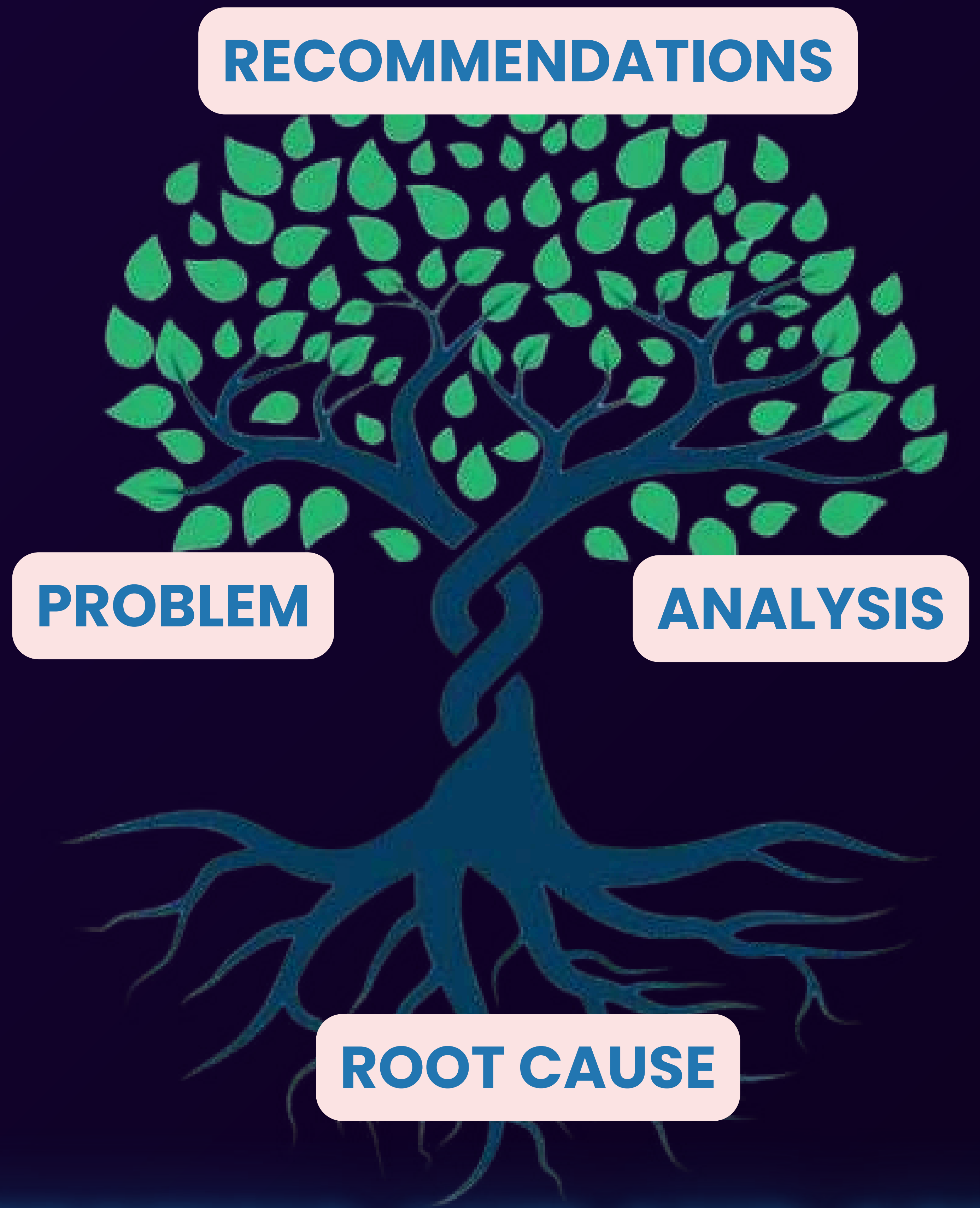
```
Average Handling Time for Most Frequent Call Reason (IRROPS): 10.01
Average Handling Time for Least Frequent Call Reason (UnaccompaniedMinor): 7.86
Percentage Difference in AHT: 27.30%
```

AHT for Most Frequent Call Reason (IRROPS): 10.01 MINUTES

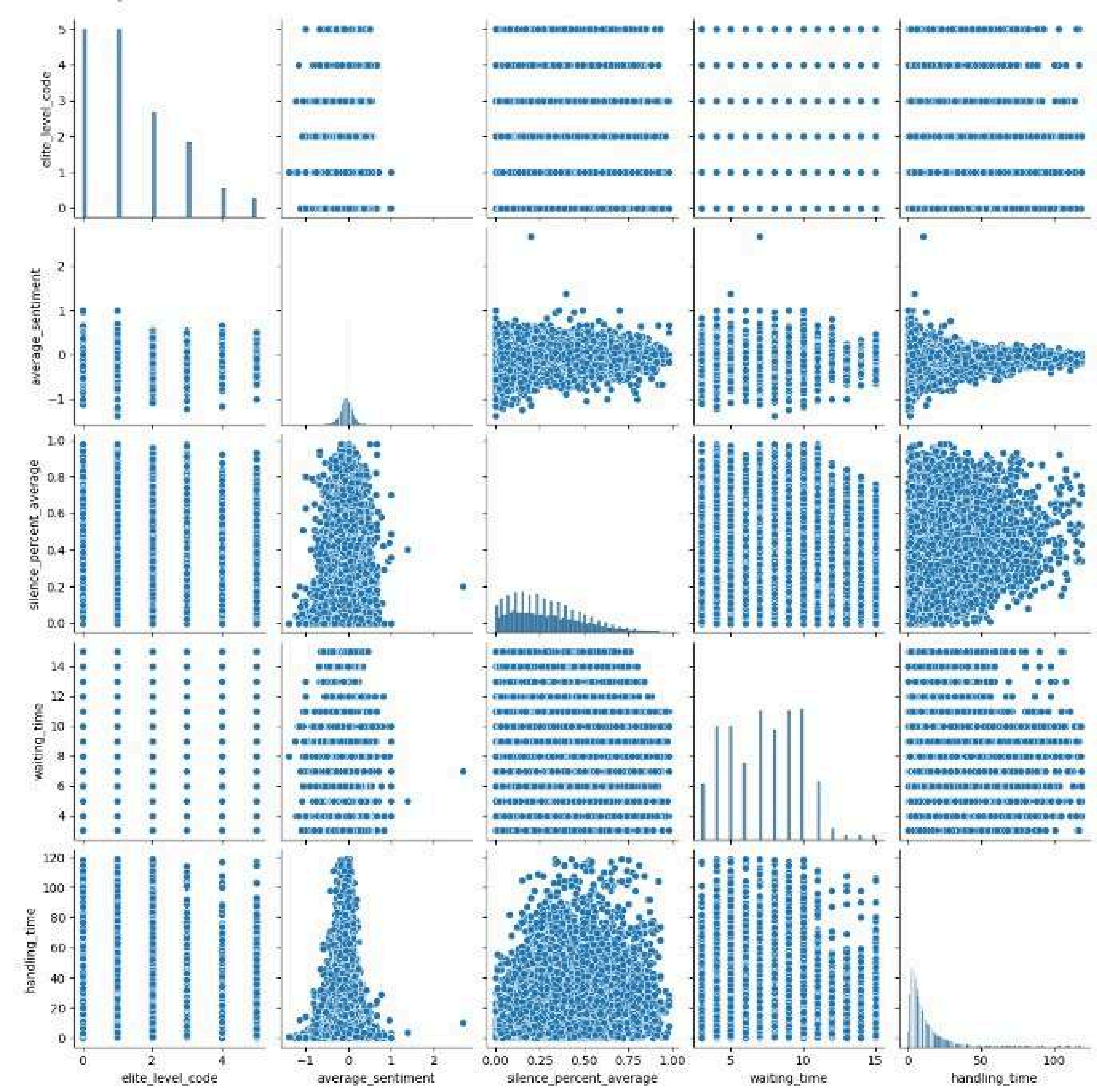
AHT for Least Frequent Call Reason (UnaccompaniedMinor): 7.86 MINUTES

PERCENTAGE DIFFERENCE IN AHT: 27.30%

ROOT CAUSE ANALYSIS



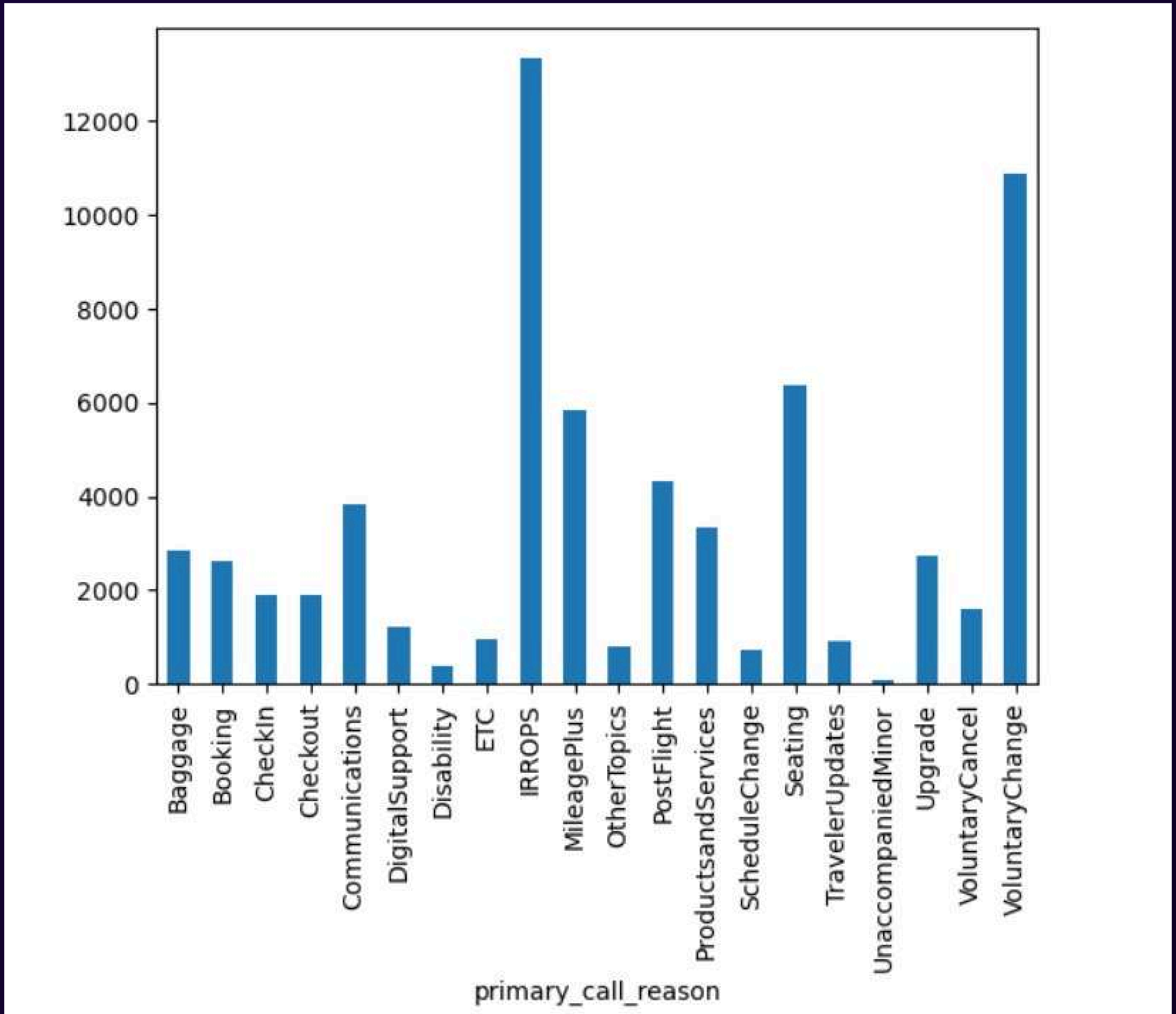
TO IDENTIFY KEY FACTORS AFFECTING AHT AND AST



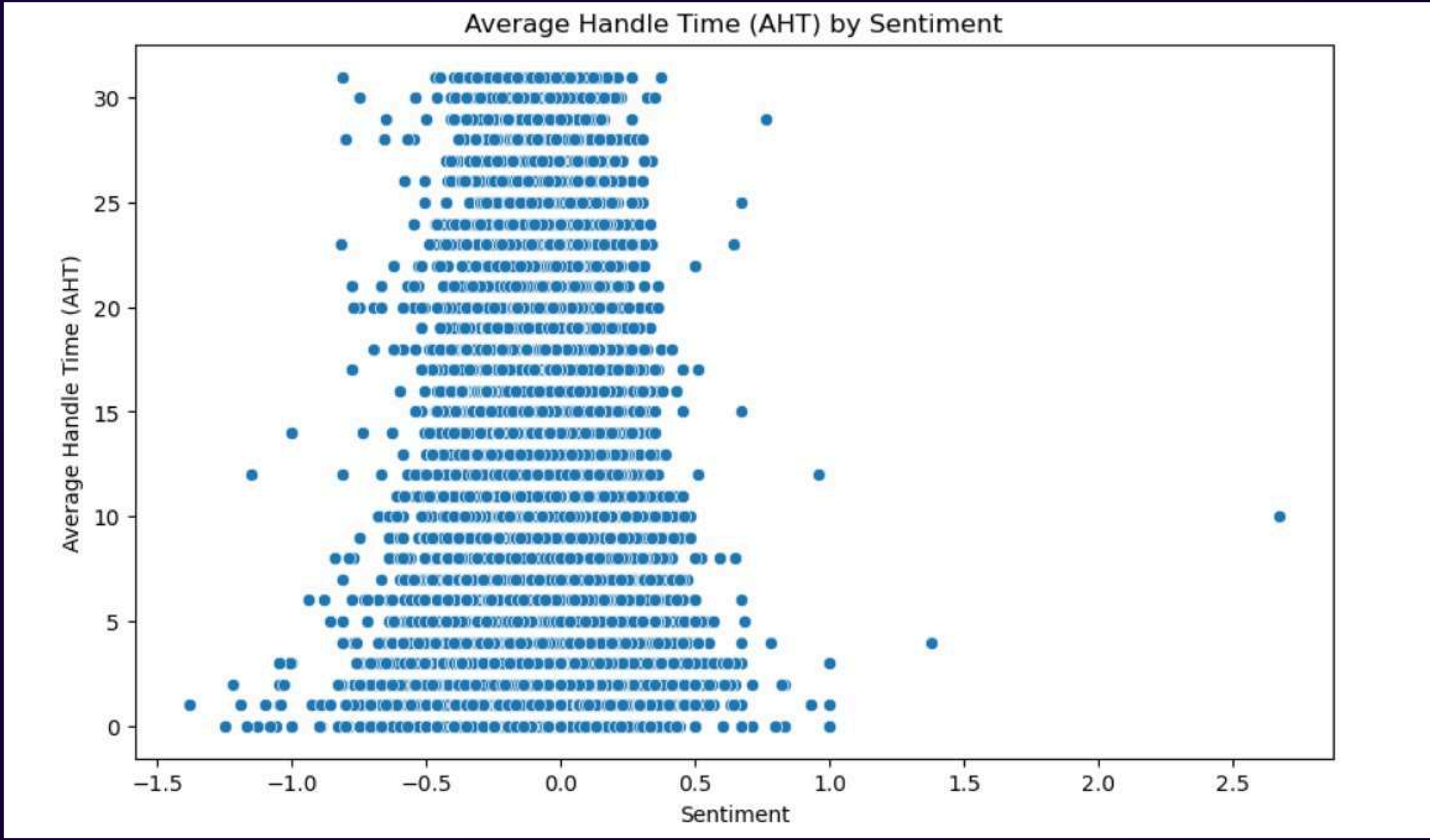
PAIR PLOT FOR ALL THE VARIABLES

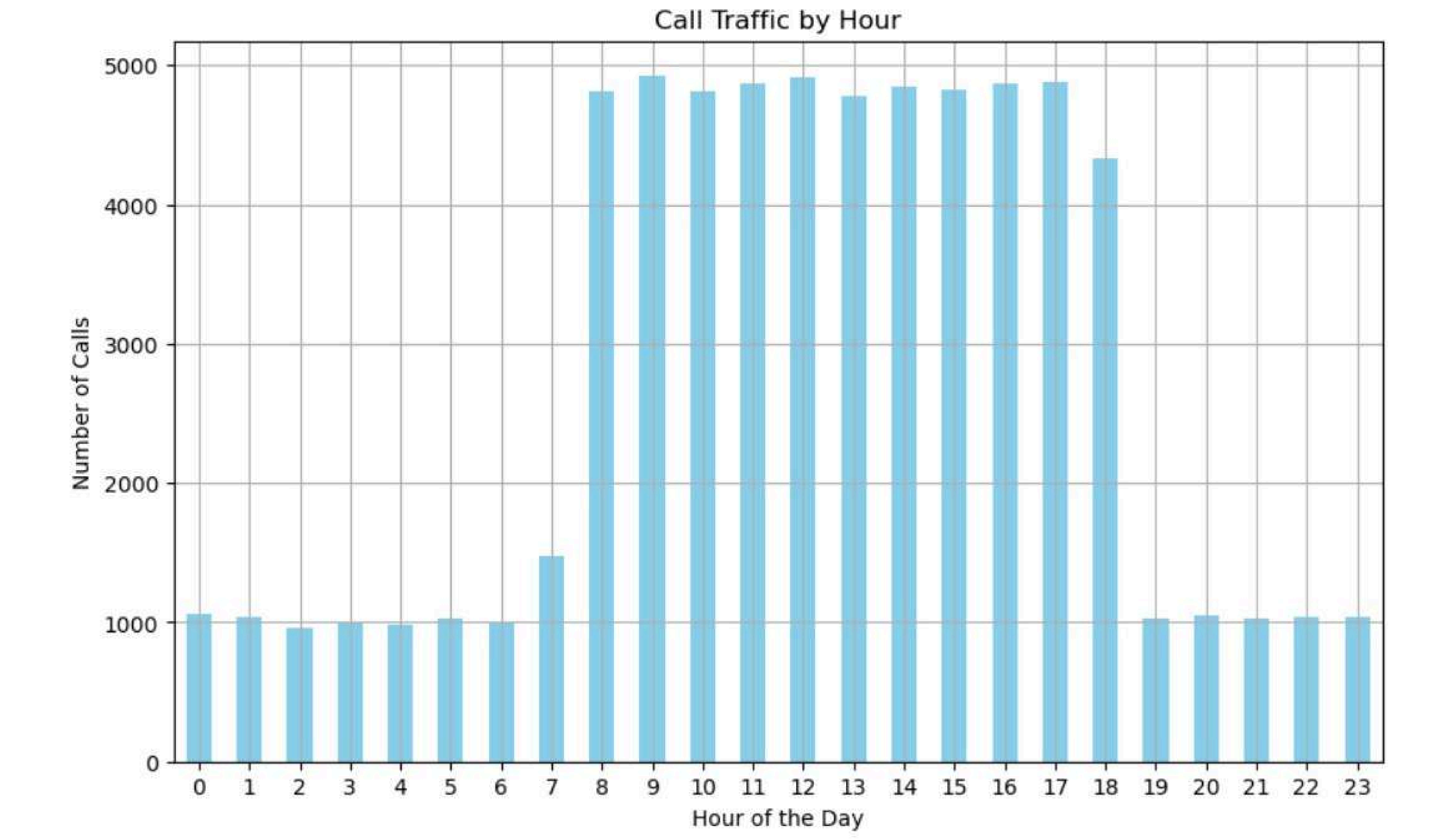
HEAT MAP FOR ALL THE FEATURES

BAR GRAPH OF PRIMARY CALL REASON



SCATTER PLOT FOR AVERAGE SENTIMENT VS AHT

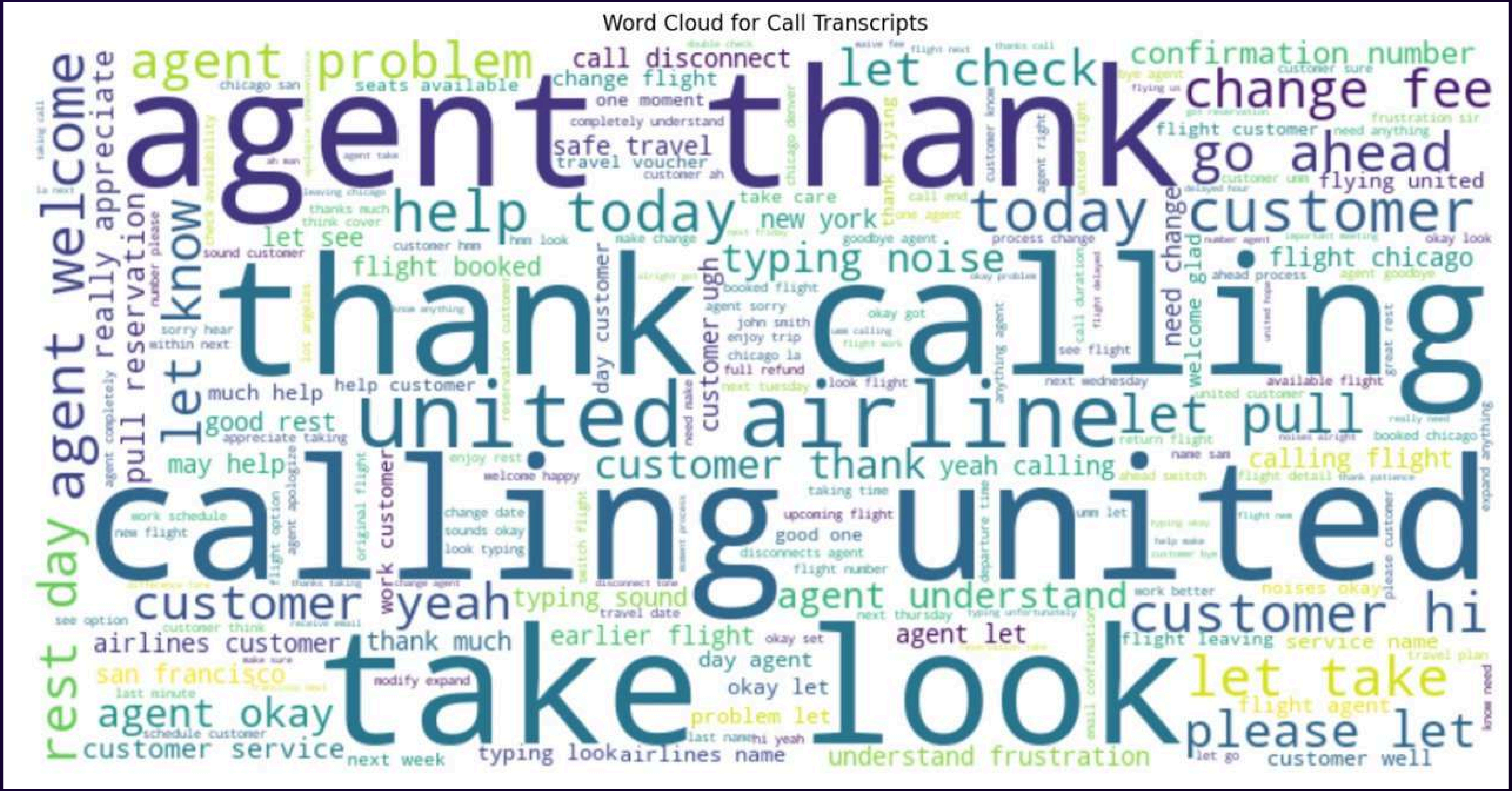




PEAK HOURS OF CALL TRAFFIC IS FROM - 8AM TO 6PM

MAJOR WORDS AFFECTING OUR TARGET VARIABLES USING SENTIMENT ANANLYSIS AND NLP

WORD CLOUD


Word Cloud for Call Transcripts
agent thank
thank calling
calling united
take look
united airline
agent welcome
agent problem
call disconnect
let check
confirmation number
change fee
go ahead
customer
today
help today
typing noise
let pull
customer thank
customer hi
agent understand
customer yeah
let take
please let
customer service
agent okay
san francisco
rest day
let know
really appreciate
pull reservation
understand frustration

TOP WORDS PER CLUSTER

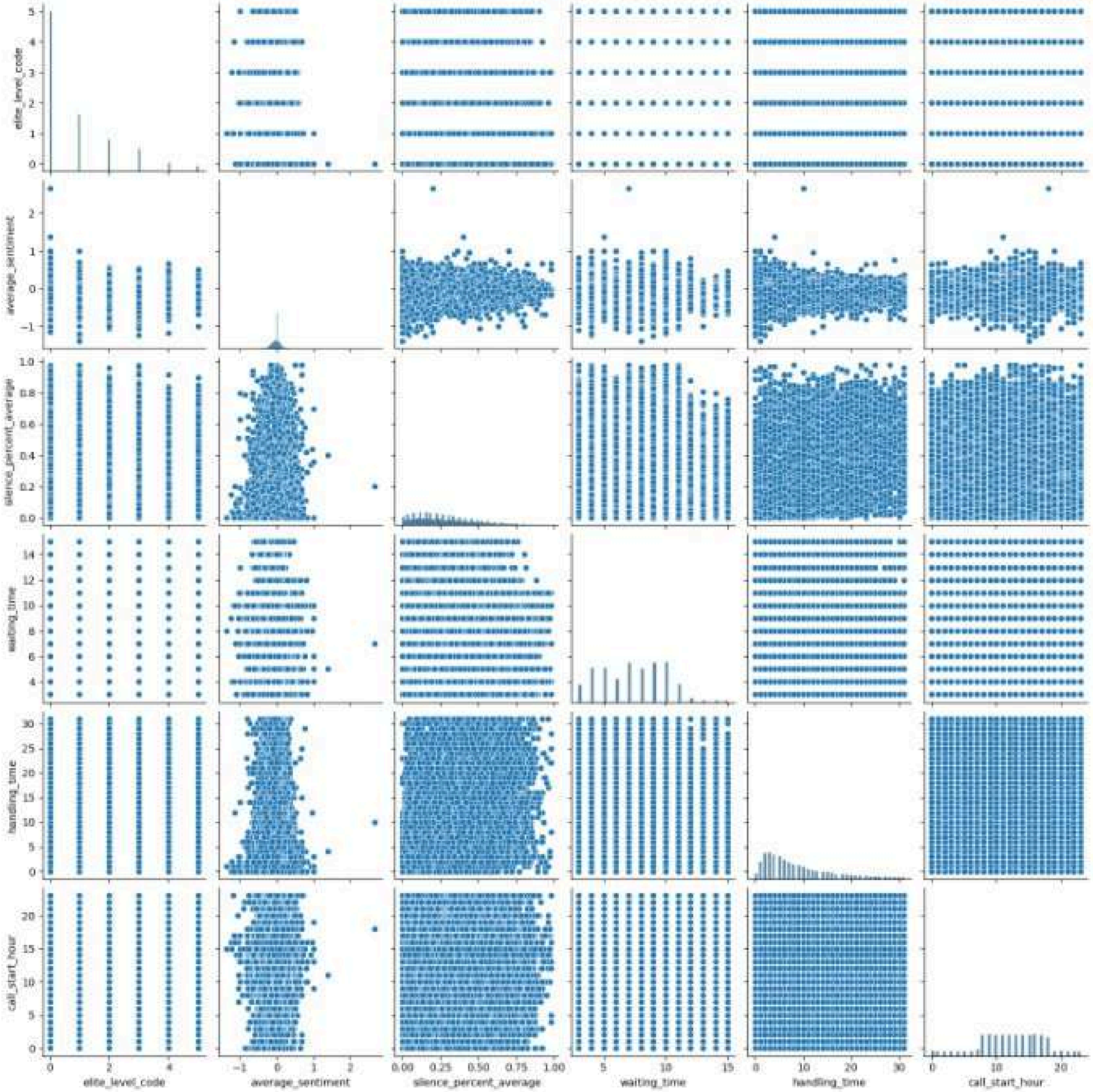
```python
print("Top words per cluster:")
order_centroids = kmeans.cluster_centers_.argsort()[:, ::-1]
terms = tfidf_vectorizer.get_feature_names_out()
for i in range(5):  # Adjust cluster number based on the results
    print(f"Cluster {i}:")
    print(" ".join([terms[ind] for ind in order_centroids[i, :10]]))
```

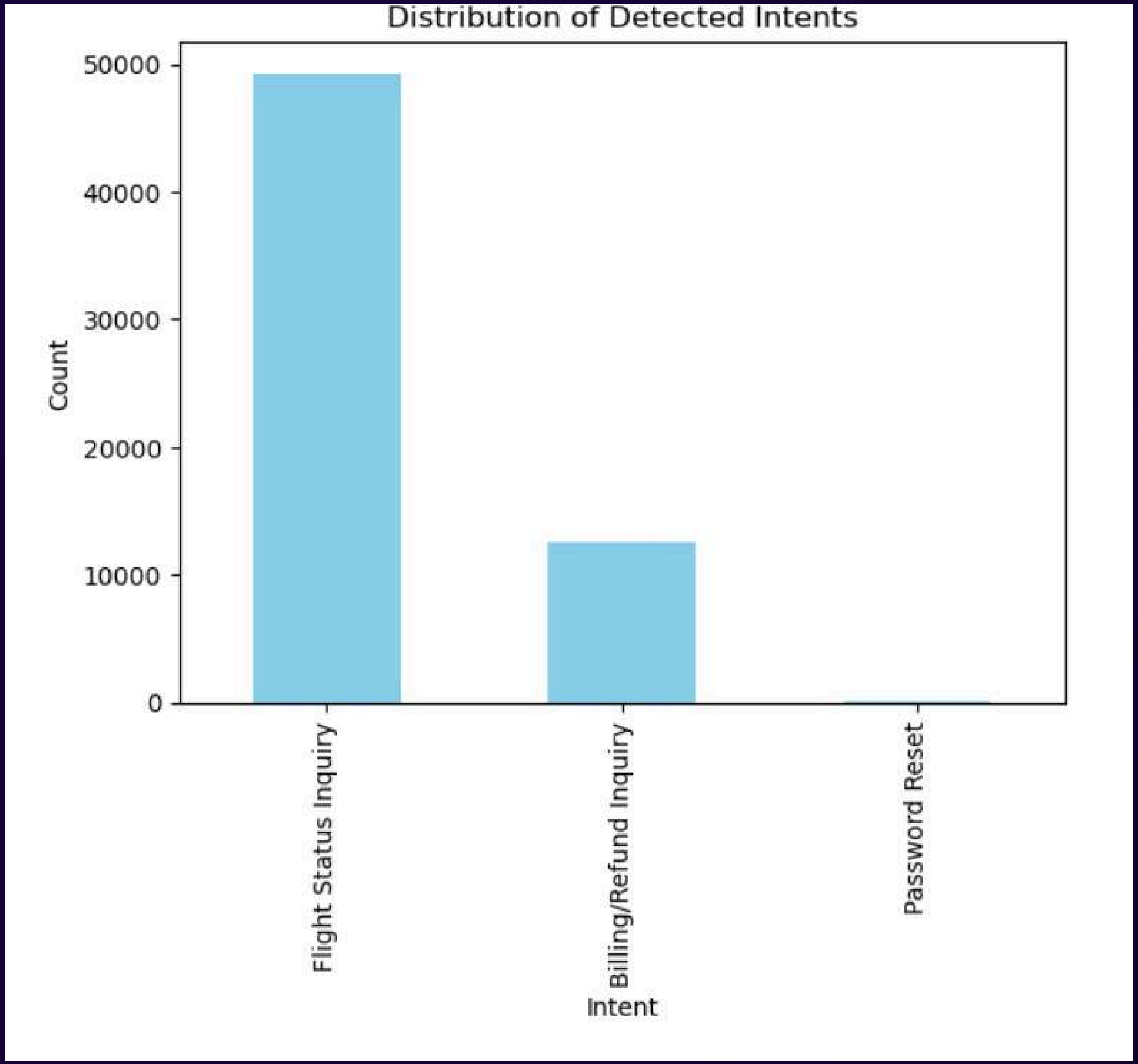
```
Top words per cluster:
Cluster 0:
customer agent flight delay refund experience voucher delays let united
Cluster 1:
flight agent change customer let fee would work help need
Cluster 2:
return change agent flight customer saturday date fee let sunday
Cluster 3:
flight agent customer let wanted time check seat help next
Cluster 4:
flight agent customer get let tomorrow sir delay meeting like
```

PAIRPLOT OF FEATURES AFTER REMOVING THE OUTLIERS

```
plt.figure(figsize=(10,6))
sns.pairplot(data=data_for_pairplot_outliers_removed)
```

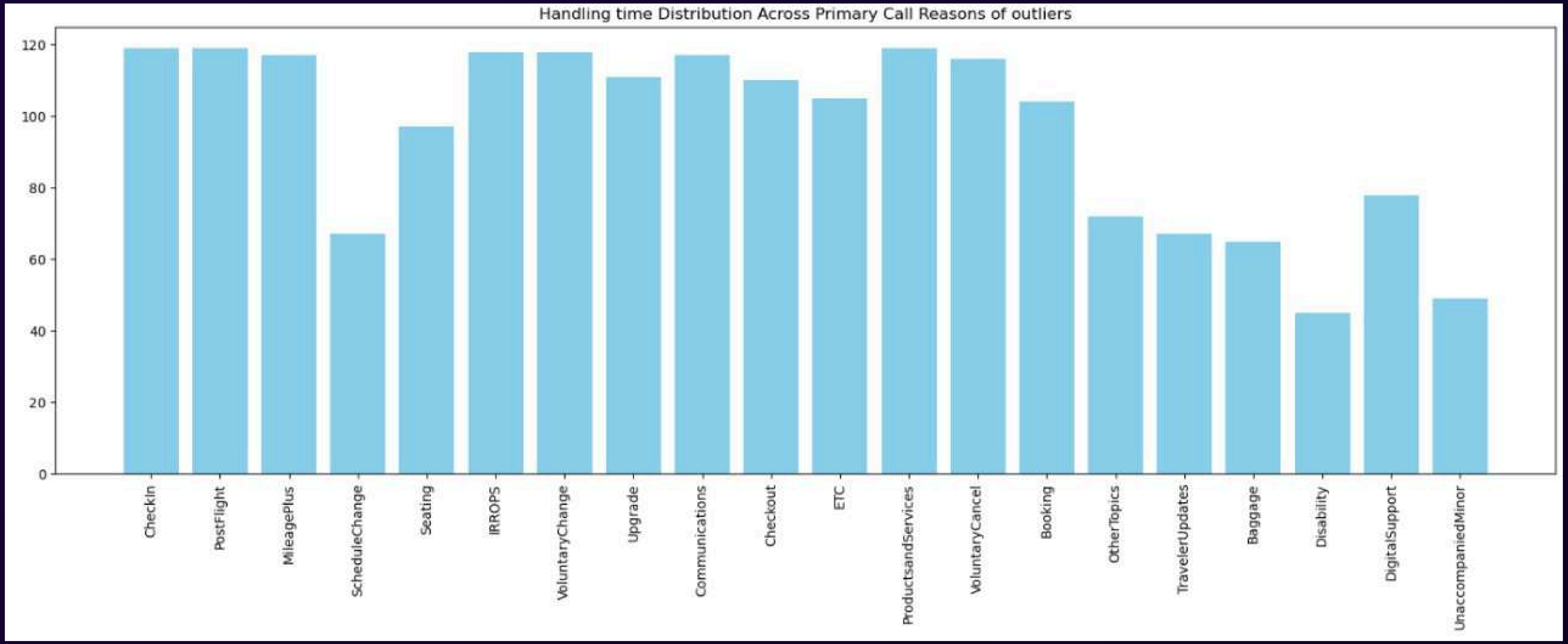
```
<seaborn.axisgrid.PairGrid at 0x1a942647238>
<Figure size 1000x600 with 0 Axes>
```



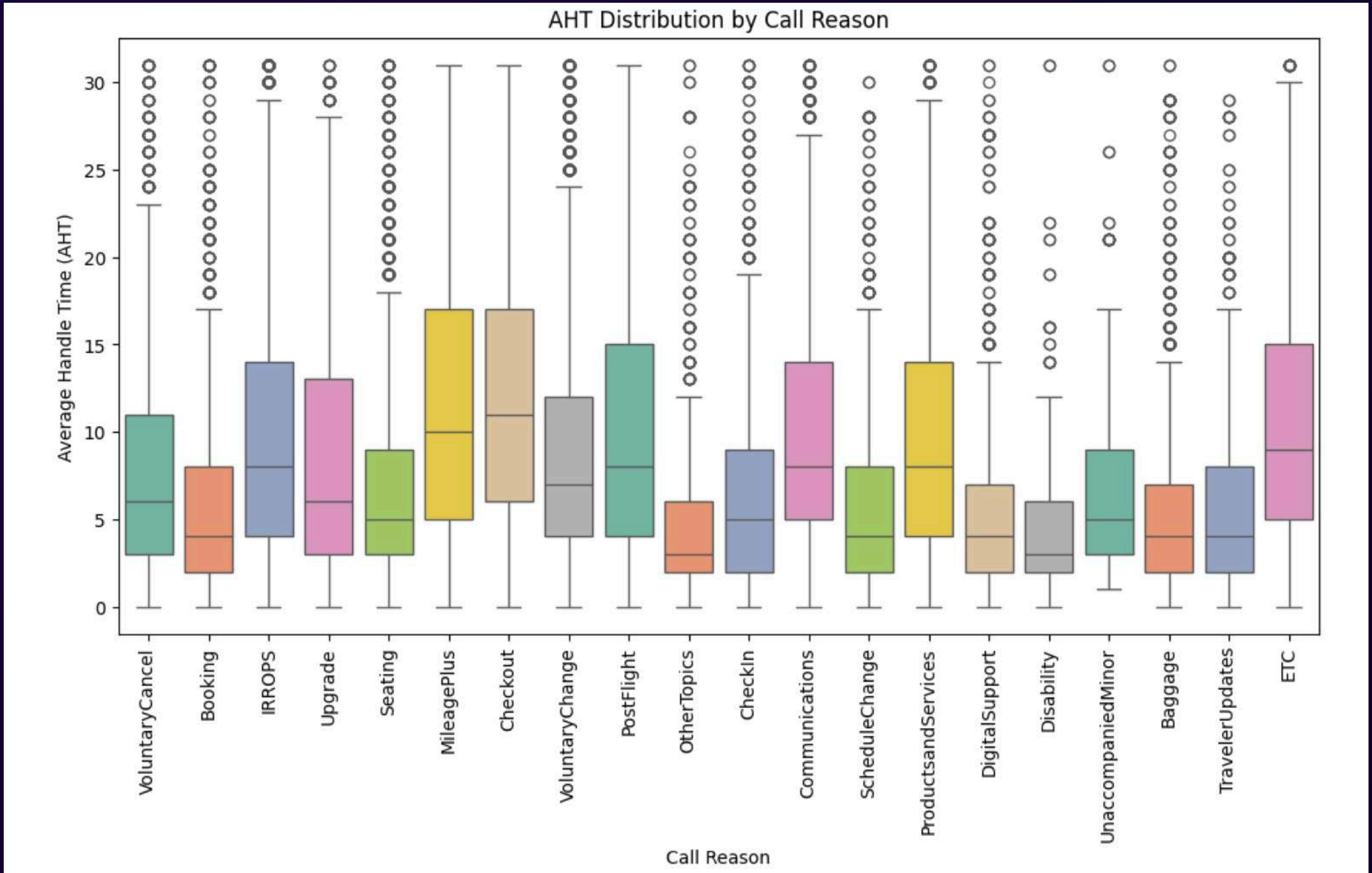


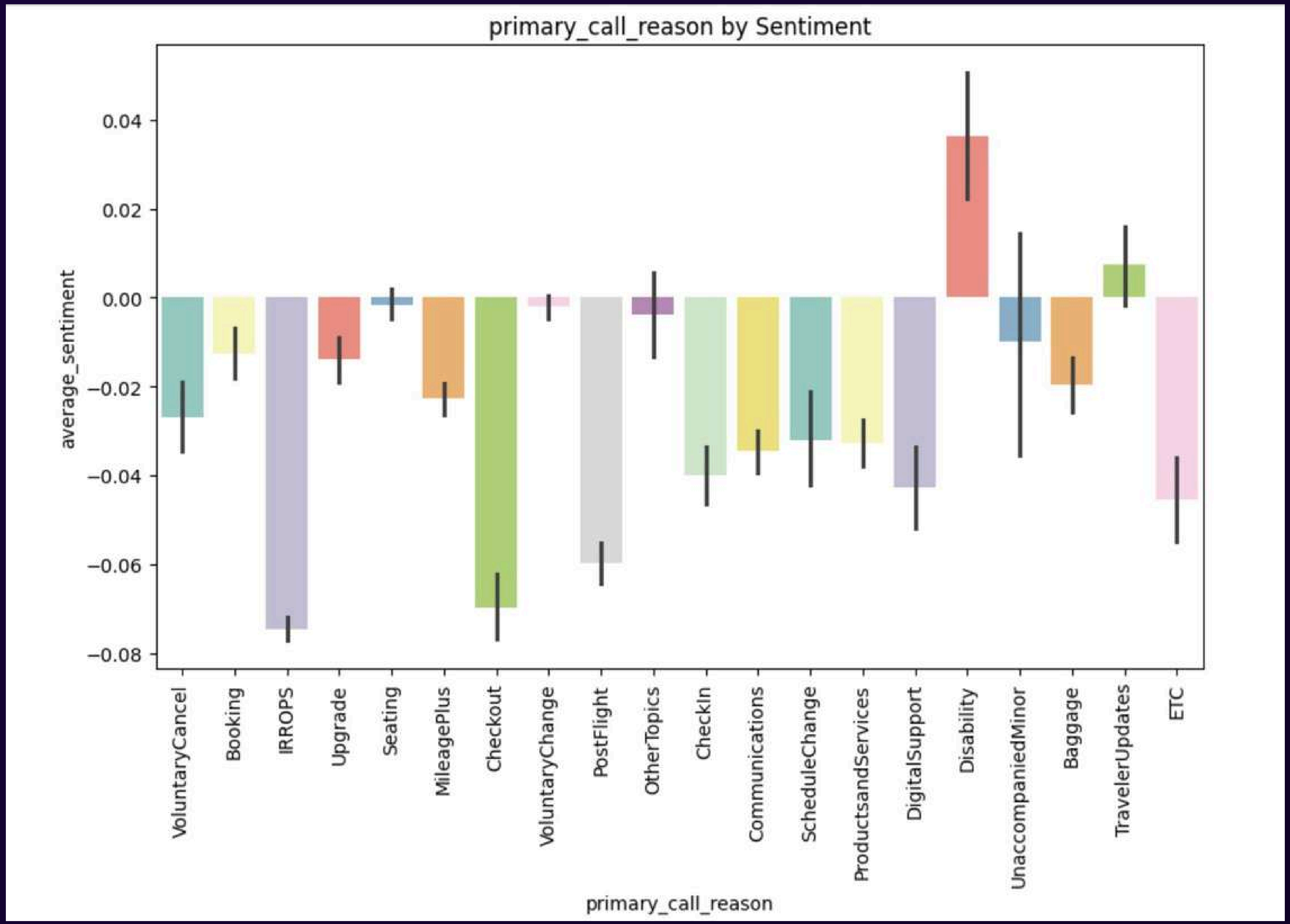
ON THE BASIS OF GIVEN PRIMARY CALL REASONS, WE USED NLP AND WE GOT THE DISTRIBUTION OF DETECTED INTENTS AS SHOWN IN THE DIAGRAM

HOW CALL HANDLING TIME VARIES ACROSS DIFFERENT CALL REASONS FOR THE OUTLIERS



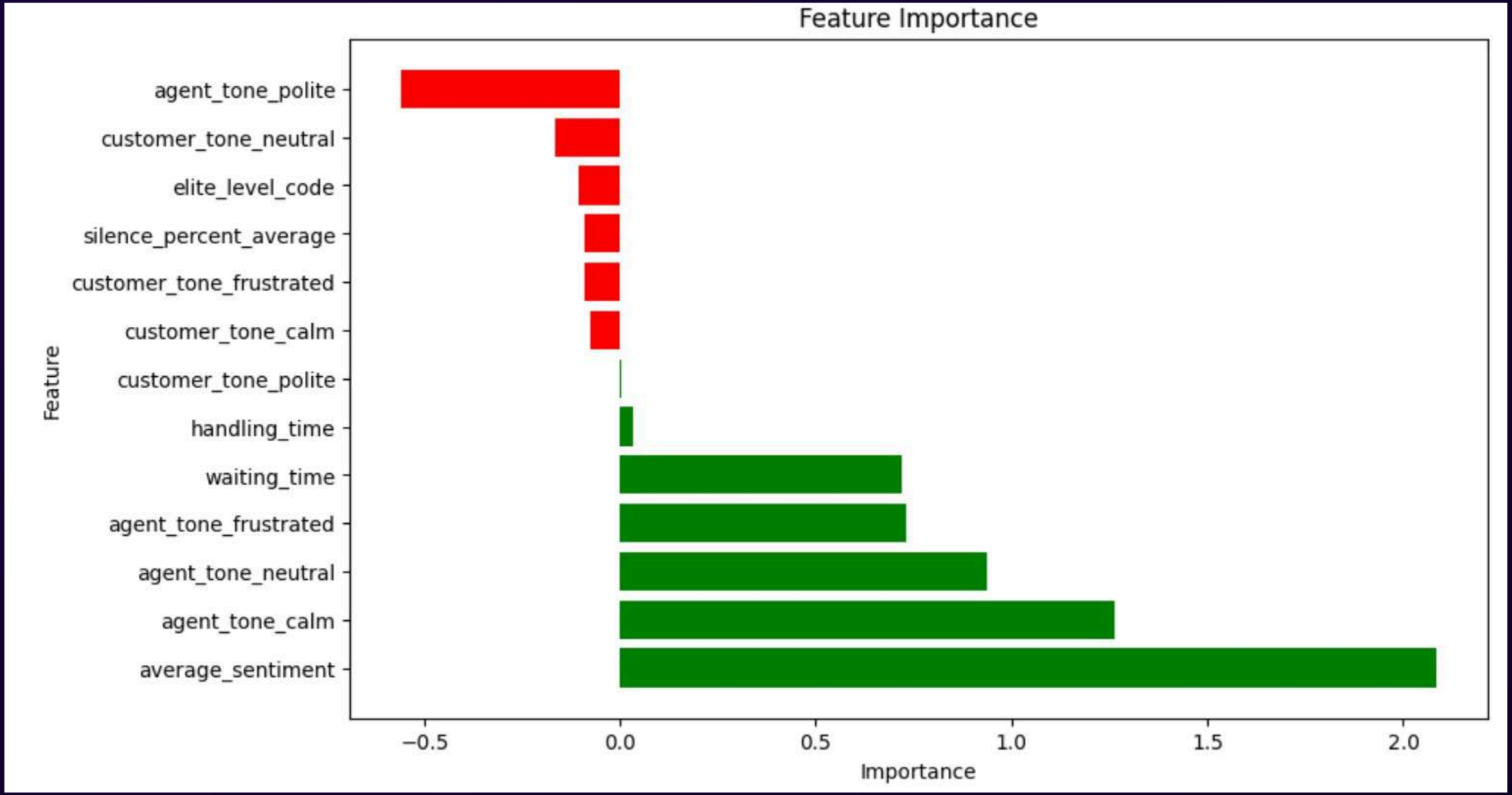
DISTRIBUTION OF CALL HANDLING TIME ACROSS DIFFERENT CALL REASONS



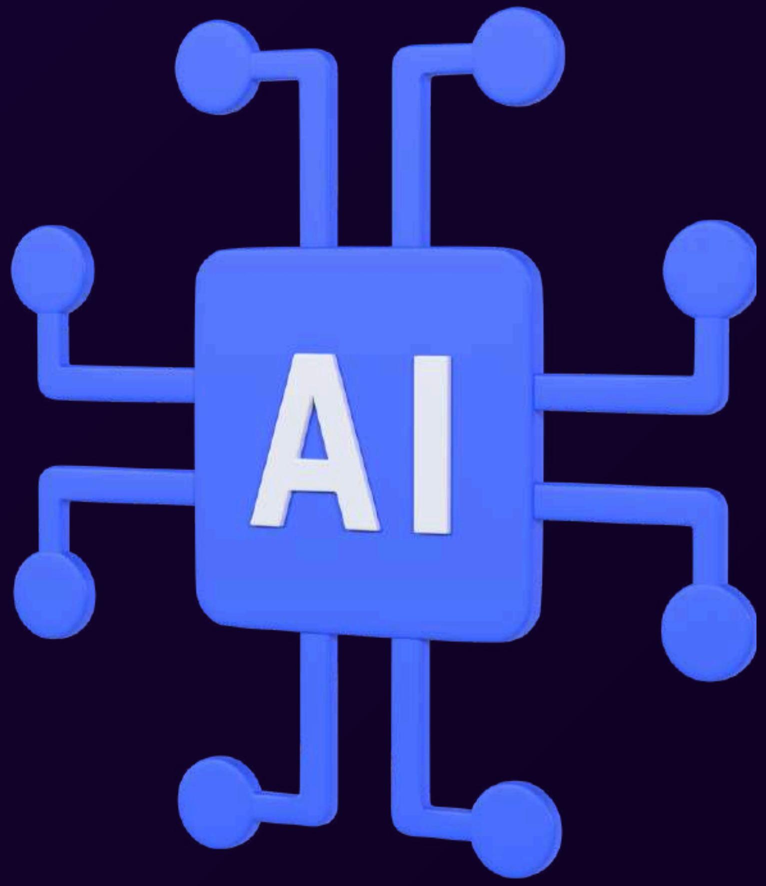


BAR DIAGRAM OF AVERAGE SENTIMENT ACROSS VARIOUS CALL REASONS

FEATURE IMPORTANCE OF THE NUMERICAL FEATURES IN PREDICTING PRIMARY CALL REASONS



TRAINING ML MODELS



TRAINING MULTINOMIAL LOGISTIC REGRESSION MODEL AND PUBLISHING THE CLASSIFICATION REPORT

```
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(max_features=1000)),  # Limit to top 1000 features
    ('classifier', RandomForestClassifier(n_estimators=100, random_state=42))
])
```

```
X_train, X_test, y_train, y_test = train_test_split(data['cleaned_transcript'], data['encoded_call_reason'],
                                                    test_size=0.2, random_state=42)
```

```
vectorizer = TfidfVectorizer(max_features=1000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Train the multinomial logistic regression model
model = LogisticRegression(multi_class='multinomial', solver='saga', max_iter=1000)
model.fit(X_train_tfidf, y_train)

# Prediction
y_pred = model.predict(X_test_tfidf)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.1977241546283593
Classification Report:
               precision    recall  f1-score   support

           0       0.29      0.01      0.01       567
           1       0.12      0.00      0.00       528
           2       0.00      0.00      0.00       329
           3       0.00      0.00      0.00       348
           4       0.00      0.00      0.00       692
           5       0.00      0.00      0.00       226
           6       0.00      0.00      0.00        64
           7       0.00      0.00      0.00       164
           8       0.22      0.65      0.33      2414
           9       0.09      0.02      0.04       987
          10       0.00      0.00      0.00       163
          11       0.13      0.03      0.05       739
          12       0.05      0.01      0.01       639
          13       0.00      0.00      0.00       134
          14       0.12      0.03      0.05      1306
          15       0.00      0.00      0.00       194
          16       0.00      0.00      0.00        25
          17       0.00      0.00      0.00       496
          18       0.00      0.00      0.00       287
          19       0.18      0.37      0.24      2089

    accuracy                           0.20     12391
   macro avg       0.06      0.06      0.04     12391
weighted avg       0.12      0.20      0.12     12391
```

TRAINING RANDOM FOREST MODEL WITH TFIDVECTORIZER

Communications	0.00	0.00	0.00	692
DigitalSupport	0.00	0.00	0.00	226
Disability	0.00	0.00	0.00	64
ETC	0.00	0.00	0.00	164
IRROPS	0.21	0.72	0.32	2414
MileagePlus	0.15	0.01	0.02	987
OtherTopics	0.00	0.00	0.00	163
PostFlight	0.07	0.00	0.00	739
ProductsandServices	0.20	0.00	0.00	639
ScheduleChange	0.00	0.00	0.00	134
Seating	0.11	0.01	0.02	1306
TravelerUpdates	0.00	0.00	0.00	194
UnaccompaniedMinor	0.00	0.00	0.00	25
Upgrade	0.00	0.00	0.00	496

[75]:
```
# Model Training
pipeline.fit(X_train, y_train)
```

[75]:
```
Pipeline
  TfidfVectorizer
  RandomForestClassifier
```

[76]:
```
# Model Evaluation
y_pred = pipeline.predict(X_test)
```

[77]:
```
# Classification report and confusion matrix
print(classification_report(y_test, y_pred, target_names=label_encoder.classes_))
confusion_mtx = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", confusion_mtx)
```

INSIGHTS & FINDINGS

- THE AVERAGE HANDLING TIME HAS A LOT OF OUTLIERS WHICH WAS LEADING ITS MEAN TO QUITE HIGH VALUE BUT AFTER REMOVING THEM WE HAVE AHT = 8.863 MINUTES

- AHT FOR MOST FREQUENT CALL REASON (IRROPS): 10.01 MINUTES
  AHT FOR LEAST FREQUENT CALL REASON (UNACCOMPANIEDMINOR): 7.86 MINUTES
  PERCENTAGE DIFFERENCE IN AHT: 27.30%

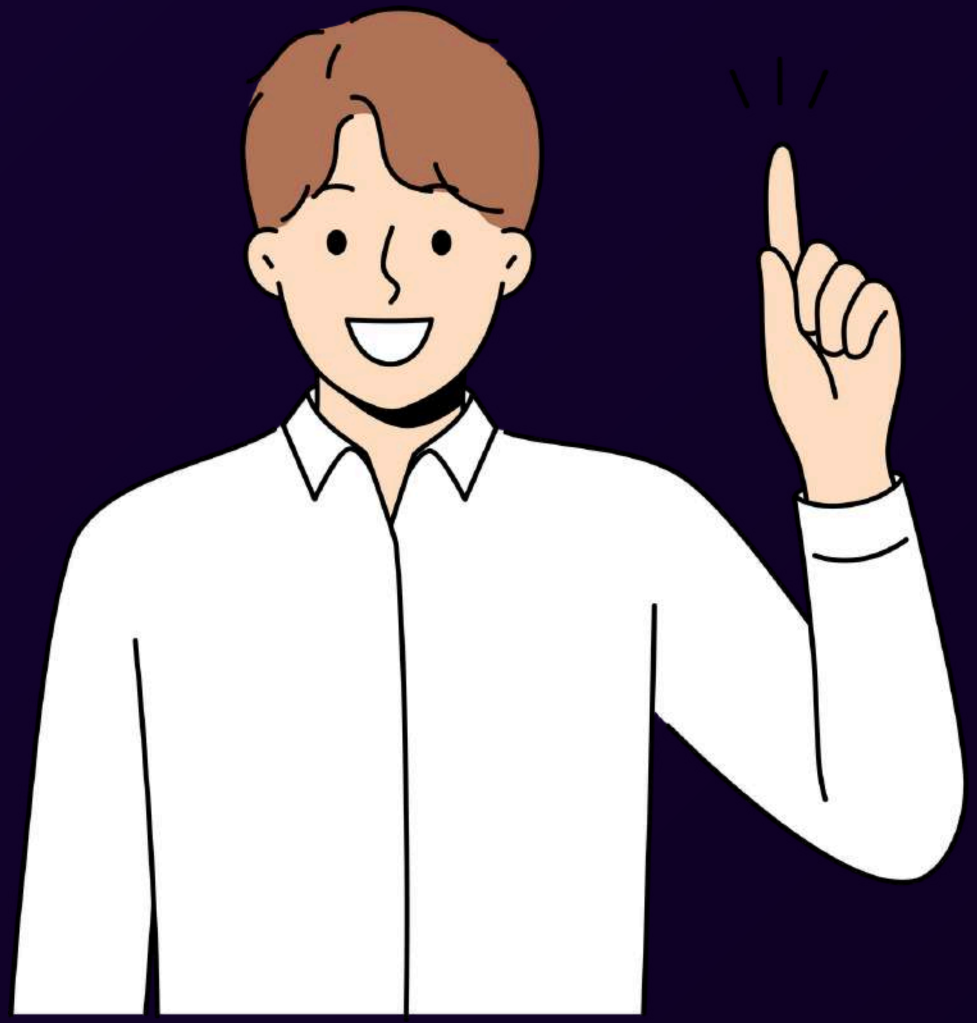
- SILENCE_PERCENTAGE_AVERAGE IS CORRELATED WITH HANDLING TIME WITH CORRELATION COEFFICIENT BEING 0.42.

- PRIMARY CALL REASONS ARE IRROPS WITH AROUND 13K PLUS CUSTOMERS AND VOLUNTARY CHANGE OF 10K PLUS CUSTOMERS.

- PEAK HOURS OF CALLING TRAFFIC ARE FROM 8AM TO 6PM

- ONLY DISABILITY AND TRAVELER UPDATE HAD AVERAGE SENTIMENT IN POSITIVE.
- FEATURE IMPORTANCE DIAGRAM COCLUDES THAT AVERAGE SENTIMENT IS HIGHLY CORRELATED WITH PRIMARY CALL REASON.
- TOP WORDS PER CLUSTER AFTER PERFORMING K MEANS CLUSTERING ARE DELAY, REFUND, EXPERIENCE, VOUCHER, FEE, SEAT
- AFTER FITTING THE NLP MODEL WE DETECTED THE INTENT OF CALLING PERSON AND WE GOT FLIGHT STATUS ENQUIRY AS THE DETECTING INTENT OF AROUND 50K PEOPLE.

RECOMMENDATIONS!!



AUTOMATE BILLING INQUIRIES IN THE IVR. IDENTIFIED 12629 CASES.

PROVIDE REAL-TIME FLIGHT INFORMATION IN THE IVR. IDENTIFIED 49286 CASES.

IMPLEMENT A SELF-SERVICE OPTION FOR PASSWORD RESET. IDENTIFIED 38 CASES

AHT AND AST OPTIMIZATION: AGENTS WITH LONGER HANDLING TIMES OR CALL REASONS THAT TEND TO EXTEND AHT SHPULD BE PRIORITIZED FOR TRAINING OR PROCESS IMROVEMENTS.

IVR SELF SERVICE OPTIONS : LOOK FOR FREQUENT CALL REASONS WITH LOW SENTIMENT SCORES TO IDENTIFY CASES THAT COULD BE AUTOMATED.

SENTIMENT CORRELATIONS : CALLS WIHT LOWER SENTIMENT SCORES AND HIGHER SILENCE PERCENTAGES MAY INDICATE AREAS WHERE IVR CAN BE MORE EFFECTIVE, REDUCING AGENT INVOLVEMENT.

SINCE 8AM TO 6PM HAS HIGHEST CALL TRAFFIC SO MORE WORKFORCE AND LOGISTICS SHOULD BE DEPLOYED FOR THAT REASONS.

TEAM NAME-404 KILLERS

WE ARE THE 404 KILLERS!!!!!!

TEAM MEMBERS-

1. SUSHANTA DUTTA
2. SAMRENDRA

SIGNING OFF