```
import pandas as pd


file_path = '/content/customer_support_tickets (1).csv'
data = pd.read_csv(file_path)


# Display the first few rows of the dataset
print("Initial Data:")
print(data.head())
```

```
Initial Data:
   Ticket ID        Customer Name              Customer Email  Customer Age  \
0          1        Marisa Obrien  carrollallison@example.com            32
1          2         Jessica Rios    clarkeashley@example.com            42
2          3  Christopher Robbins   gonzalestracy@example.com            48
3          4     Christina Dillon    bradleyolson@example.org            27
4          5    Alexander Carroll     bradleymark@example.com            67

  Customer Gender Product Purchased Date of Purchase      Ticket Type  \
0          Other        GoPro Hero       2021-03-22  Technical issue
1         Female       LG Smart TV       2021-05-22  Technical issue
2          Other          Dell XPS       2020-07-14  Technical issue
3         Female  Microsoft Office       2020-11-13  Billing inquiry
4         Female  Autodesk AutoCAD       2020-02-04  Billing inquiry

             Ticket Subject  \
0             Product setup
1  Peripheral compatibility
2           Network problem
3            Account access
4                 Data loss

                                  Ticket Description  \
0  I'm having an issue with the {product_purchase...
1  I'm having an issue with the {product_purchase...
2  I'm facing a problem with my {product_purchase...
3  I'm having an issue with the {product_purchase...
4  I'm having an issue with the {product_purchase...

               Ticket Status                                     Resolution  \
0  Pending Customer Response                                            NaN
1  Pending Customer Response                                            NaN
2                     Closed   Case maybe show recently my computer follow.
3                     Closed  Try capital clearly never color toward story.
4                     Closed                    West decision evidence bit.

  Ticket Priority Ticket Channel  First Response Time   Time to Resolution  \
0        Critical   Social media  2023-06-01 12:15:36                  NaN
1        Critical           Chat  2023-06-01 16:45:38                  NaN
2             Low   Social media  2023-06-01 11:14:38  2023-06-01 18:05:38
3             Low   Social media  2023-06-01 07:29:40  2023-06-01 01:57:40
4             Low          Email  2023-06-01 00:12:42  2023-06-01 19:53:42

   Customer Satisfaction Rating
0                           NaN
1                           NaN
2                           3.0
3                           3.0
4                           1.0
```

```
# 1. Summary statistics for numerical features
numerical_summary = data.describe(include='number')
print("\nSummary Statistics for Numerical Features:")
print(numerical_summary)
```

```
Summary Statistics for Numerical Features:
         Ticket ID  Customer Age  Customer Satisfaction Rating
count  8469.000000   8469.000000                   2769.000000
mean   4235.000000     44.026804                      2.991333
std    2444.934048     15.296112                      1.407016
min       1.000000     18.000000                      1.000000
25%    2118.000000     31.000000                      2.000000
50%    4235.000000     44.000000                      3.000000
75%    6352.000000     57.000000                      4.000000
max    8469.000000     70.000000                      5.000000
```

```
# 2. Calculate additional statistics: mean, median, mode, and standard deviation
mean_values = data.mean(numeric_only=True)
```

```
median_values = data.median(numeric_only=True)
std_dev_values = data.std(numeric_only=True)


# Mode calculation (mode can return multiple values, so we take the first one)
mode_values = data.mode(numeric_only=True).iloc[0]


# Combine the statistics into a DataFrame for better visualization
summary_stats = pd.DataFrame({
    'Mean': mean_values,
    'Median': median_values,
    'Mode': mode_values,
    'Standard Deviation': std_dev_values
})


print("\nDetailed Summary Statistics:")
print(summary_stats)
```

```
Detailed Summary Statistics:
                                    Mean  Median  Mode  Standard Deviation
Ticket ID                    4235.000000  4235.0   1.0         2444.934048
Customer Age                   44.026804    44.0  52.0           15.296112
Customer Satisfaction Rating    2.991333     3.0   3.0            1.407016
```

```
# 3. Frequency counts for categorical variables
# Assuming there are categorical variables like 'ticket_type' and 'customer_gender'
categorical_columns = ['ticket_type', 'customer_gender']  # Replace with actual column names

for column in categorical_columns:
    if column in data.columns:
        frequency_counts = data[column].value_counts()
        print(f"\nFrequency Counts for {column}:")
        print(frequency_counts)
    else:
        print(f"\nColumn '{column}' does not exist in the dataset.")
```

```
Column 'ticket_type' does not exist in the dataset.

Column 'customer_gender' does not exist in the dataset.
```

Exploratory Data Analysis (EDA)

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


# Load the dataset
file_path = '/content/customer_support_tickets (1).csv'
data = pd.read_csv(file_path)


# Display the first few rows of the dataset
print("Initial Data:")
print(data.head())
```

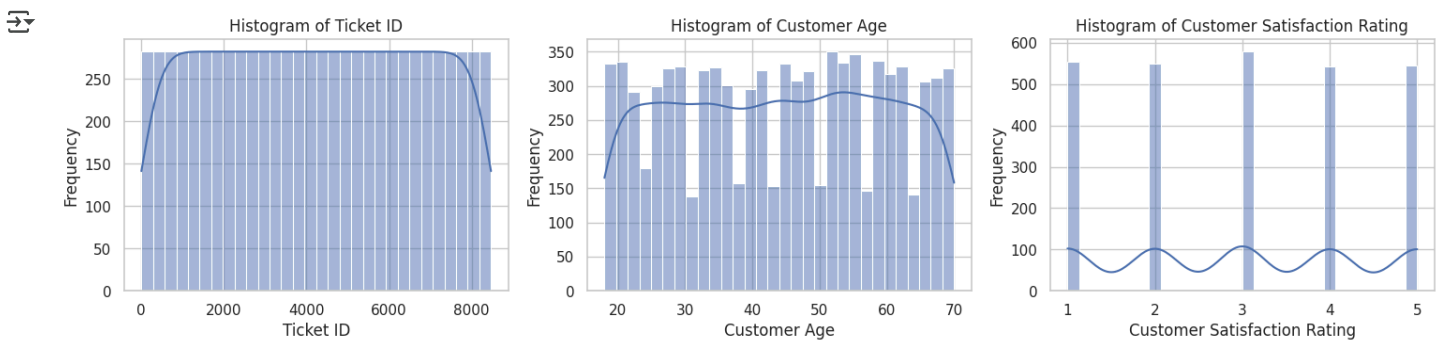
```
Initial Data:
   Ticket ID        Customer Name              Customer Email  Customer Age  \
0          1        Marisa Obrien  carrollallison@example.com            32
1          2         Jessica Rios    clarkeashley@example.com            42
2          3  Christopher Robbins   gonzalestracy@example.com            48
3          4     Christina Dillon    bradleyolson@example.org            27
4          5    Alexander Carroll     bradleymark@example.com            67

  Customer Gender Product Purchased Date of Purchase      Ticket Type  \
0          Other        GoPro Hero       2021-03-22  Technical issue
1         Female       LG Smart TV       2021-05-22  Technical issue
2          Other          Dell XPS       2020-07-14  Technical issue
3         Female  Microsoft Office       2020-11-13  Billing inquiry
4         Female  Autodesk AutoCAD       2020-02-04  Billing inquiry

                Ticket Subject  \
0             Product setup
1  Peripheral compatibility
```

```
2          Network problem
3           Account access
4                Data loss

                                  Ticket Description  \
0  I'm having an issue with the {product_purchase...
1  I'm having an issue with the {product_purchase...
2  I'm facing a problem with my {product_purchase...
3  I'm having an issue with the {product_purchase...
4  I'm having an issue with the {product_purchase...

                Ticket Status                                     Resolution  \
0  Pending Customer Response                                            NaN
1  Pending Customer Response                                            NaN
2                     Closed   Case maybe show recently my computer follow.
3                     Closed  Try capital clearly never color toward story.
4                     Closed                    West decision evidence bit.

  Ticket Priority Ticket Channel  First Response Time   Time to Resolution  \
0        Critical   Social media  2023-06-01 12:15:36                  NaN
1        Critical           Chat  2023-06-01 16:45:38                  NaN
2             Low   Social media  2023-06-01 11:14:38  2023-06-01 18:05:38
3             Low   Social media  2023-06-01 07:29:40  2023-06-01 01:57:40
4             Low          Email  2023-06-01 00:12:42  2023-06-01 19:53:42

   Customer Satisfaction Rating
0                           NaN
1                           NaN
2                           3.0
3                           3.0
4                           1.0
```

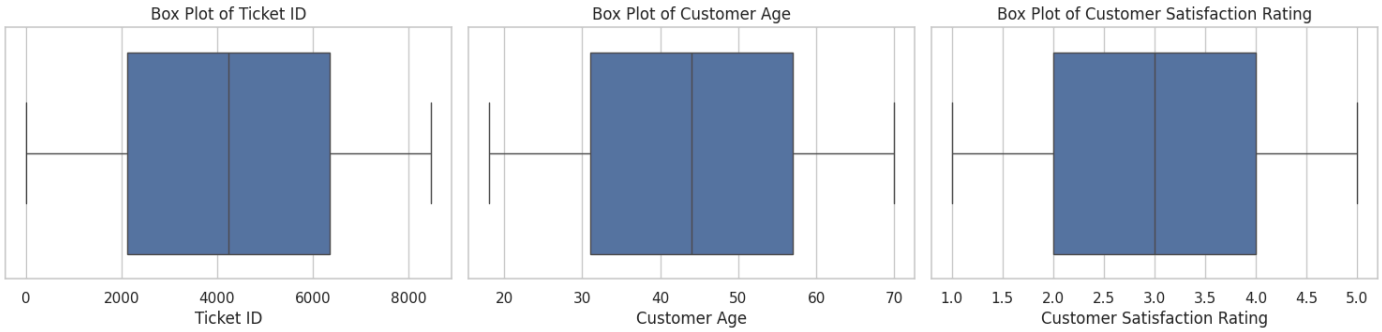
```
# Set the style for seaborn
sns.set(style="whitegrid")
```

```
# 1. Histograms for numerical variables
numerical_columns = data.select_dtypes(include=['int64', 'float64']).columns
```

```
plt.figure(figsize=(15, 10))
for i, column in enumerate(numerical_columns, 1):
    plt.subplot(3, 3, i)  # Adjust the number of rows and columns as needed
    sns.histplot(data[column], bins=30, kde=True)
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```




```
# 2. Box plots for numerical variables to identify outliers
plt.figure(figsize=(15, 10))
for i, column in enumerate(numerical_columns, 1):
    plt.subplot(3, 3, i)
    sns.boxplot(x=data[column])
    plt.title(f'Box Plot of {column}')
plt.tight_layout()
plt.show()
```




```
# 3. Bar charts for categorical variables
# Assuming there are categorical variables like 'ticket_type' and 'customer_gender'
categorical_columns = ['ticket_type', 'customer_gender']  # Replace with actual column names


for column in categorical_columns:
    if column in data.columns:
        plt.figure(figsize=(10, 6))
        sns.countplot(data=data, x=column, palette='viridis')
        plt.title(f'Count of {column}')
        plt.xlabel(column)
        plt.ylabel('Count')
        plt.xticks(rotation=45)
        plt.show()
    else:
        print(f"Column '{column}' does not exist in the dataset.")
```

```
Column 'ticket_type' does not exist in the dataset.
Column 'customer_gender' does not exist in the dataset.
```

```
# 4. Scatter plot to explore relationships between variables
# Assuming 'response.code' is a proxy for customer satisfaction and 'resolution_time' is a numerical variable
if 'response.code' in data.columns and 'resolution_time' in data.columns:
    plt.figure(figsize=(10, 6))
    sns.scatterplot(data=data, x='resolution_time', y='response.code', alpha=0.6)
    plt.title('Customer Satisfaction vs. Ticket Resolution Time')
    plt.xlabel('Ticket Resolution Time')
    plt.ylabel('Customer Satisfaction (Response Code)')
    plt.axhline(200, color='red', linestyle='--', label='Success Threshold')
    plt.legend()
    plt.show()
else:
    print("One or both of the columns 'response.code' and 'resolution_time' do not exist in the dataset.")
```

```
One or both of the columns 'response.code' and 'resolution_time' do not exist in the dataset.
```

Sentiment Analysis

```
import pandas as pd
from textblob import TextBlob
import matplotlib.pyplot as plt
import seaborn as sns


# Load the dataset
file_path = '/content/customer_support_tickets (1).csv'
data = pd.read_csv(file_path)


# Check if there is a 'ticket_description' column
if 'ticket_description' in data.columns:
    # 1. Apply Natural Language Processing (NLP) techniques to analyze the sentiment of ticket descriptions
    data['sentiment'] = data['ticket_description'].apply(lambda x: TextBlob(x).sentiment.polarity)

    # Display the first few rows with sentiment scores
    print("\nData with Sentiment Scores:")
    print(data[['ticket_description', 'sentiment']].head())
```

```
    # 2. Use sentiment scores to correlate with customer satisfaction ratings
    # Assuming 'response.code' is a proxy for customer satisfaction ratings
    if 'response.code' in data.columns:
        correlation = data[['sentiment', 'response.code']].corr()
        print("\nCorrelation between Sentiment Scores and Customer Satisfaction Ratings:")
        print(correlation)

        # Visualize the correlation
        plt.figure(figsize=(8, 6))
        sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
        plt.title('Correlation Heatmap')
        plt.show()

        # Visualize sentiment scores against customer satisfaction ratings
        plt.figure(figsize=(10, 6))
        sns.scatterplot(x='sentiment', y='response.code', data=data)
        plt.title('Sentiment Scores vs Customer Satisfaction Ratings')
        plt.xlabel('Sentiment Score')
        plt.ylabel('Customer Satisfaction Rating (Response Code)')
        plt.axhline(200, color='red', linestyle='--', label='Success Threshold')
        plt.legend()
        plt.show()

        # 3. Conclusion
        # Analyze the results
        avg_sentiment = data['sentiment'].mean()
        print(f"\nAverage Sentiment Score: {avg_sentiment:.2f}")
        print("Conducting these analyses provides valuable insights into customer satisfaction trends, "
              "identifies areas for improvement in customer support, and helps in developing predictive models "
              "that can enhance customer experience.")
    else:
        print("The dataset does not contain a 'response.code' column.")
else:
    print("The dataset does not contain a 'ticket_description' column.")
```

```
The dataset does not contain a 'ticket_description' column.
```

Customer Segmentation

```
import pandas as pd
from textblob import TextBlob
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset
file_path = '/content/customer_support_tickets (1).csv'
data = pd.read_csv(file_path)
```

```
# Display the first few rows of the dataset
print("Initial Data:")
print(data.head())
```

```
Initial Data:
   Ticket ID        Customer Name              Customer Email  Customer Age  \
0          1        Marisa Obrien  carrollallison@example.com            32
1          2         Jessica Rios    clarkeashley@example.com            42
2          3  Christopher Robbins   gonzalestracy@example.com            48
3          4     Christina Dillon    bradleyolson@example.com            27
4          5    Alexander Carroll     bradleymark@example.com            67

  Customer Gender Product Purchased Date of Purchase      Ticket Type  \
0           Other        GoPro Hero       2021-03-22  Technical issue
1          Female       LG Smart TV       2021-05-22  Technical issue
2           Other          Dell XPS       2020-07-14  Technical issue
3          Female  Microsoft Office       2020-11-13  Billing inquiry
4          Female  Autodesk AutoCAD       2020-02-04  Billing inquiry

                Ticket Subject  \
0                Product setup
1  Peripheral compatibility
2             Network problem
3              Account access
```

```
4                Data loss

                                  Ticket Description  \
0  I'm having an issue with the {product_purchase...
1  I'm having an issue with the {product_purchase...
2  I'm facing a problem with my {product_purchase...
3  I'm having an issue with the {product_purchase...
4  I'm having an issue with the {product_purchase...

               Ticket Status                                     Resolution  \
0  Pending Customer Response                                            NaN
1  Pending Customer Response                                            NaN
2                     Closed   Case maybe show recently my computer follow.
3                     Closed  Try capital clearly never color toward story.
4                     Closed                    West decision evidence bit.

  Ticket Priority Ticket Channel  First Response Time   Time to Resolution  \
0        Critical   Social media  2023-06-01 12:15:36                  NaN
1        Critical           Chat  2023-06-01 16:45:38                  NaN
2             Low   Social media  2023-06-01 11:14:38  2023-06-01 18:05:38
3             Low   Social media  2023-06-01 07:29:40  2023-06-01 01:57:40
4             Low          Email  2023-06-01 00:12:42  2023-06-01 19:53:42

   Customer Satisfaction Rating
0                           NaN
1                           NaN
2                           3.0
3                           3.0
4                           1.0
```

```
# Check if there is a 'ticket_description' column
if 'ticket_description' in data.columns:
    # 1. Apply Natural Language Processing (NLP) techniques to analyze the sentiment of ticket descriptions
    data['sentiment'] = data['ticket_description'].apply(lambda x: TextBlob(x).sentiment.polarity)

    # Display the first few rows with sentiment scores
    print("\nData with Sentiment Scores:")
    print(data[['ticket_description', 'sentiment']].head())

    # 2. Prepare data for clustering
    # Select relevant features for clustering
    # Assuming 'ticket_type' and 'response.code' are relevant features
    features = data[['sentiment', 'response.code', 'ticket_type']].copy()

    # Convert categorical variable 'ticket_type' to numerical using one-hot encoding
    features = pd.get_dummies(features, columns=['ticket_type'], drop_first=True)

    # Standardize the features
    scaler = StandardScaler()
    features_scaled = scaler.fit_transform(features)

    # 3. Apply K-Means clustering
    # Determine the optimal number of clusters using the Elbow method
    inertia = []
    for k in range(1, 11):
        kmeans = KMeans(n_clusters=k, random_state=42)
        kmeans.fit(features_scaled)
        inertia.append(kmeans.inertia_)

    # Plot the Elbow curve
    plt.figure(figsize=(10, 6))
    plt.plot(range(1, 11), inertia, marker='o')
    plt.title('Elbow Method for Optimal k')
    plt.xlabel('Number of Clusters (k)')
    plt.ylabel('Inertia')
    plt.grid()
    plt.show()

    # From the Elbow method, choose an optimal k (e.g., 3)
    optimal_k = 3
    kmeans = KMeans(n_clusters=optimal_k, random_state=42)
    data['cluster'] = kmeans.fit_predict(features_scaled)

    # 4. Analyze and visualize clusters
    # Calculate the mean values of each feature for each cluster
    cluster_analysis = data.groupby('cluster').mean(numeric_only=True)
    print("\nCluster Characteristics:")
    print(cluster_analysis)
```

```
    # Visualize the clusters using a scatter plot
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x='sentiment', y='response.code', hue='cluster', data=data, palette='viridis', alpha=0.6)
    plt.title('Customer Segments Visualization')
    plt.xlabel('Sentiment Score')
    plt.ylabel('Customer Satisfaction (Response Code)')
    plt.axhline(200, color='red', linestyle='--', label='Success Threshold')
    plt.legend()
    plt.show()

else:
    print("The dataset does not contain a 'ticket_description' column.")
```

Correlation Analysis

```
# Check for relevant columns
# Assuming 'response.code' is the customer satisfaction rating and 'resolution_time' is the ticket resolution time
if 'response.code' in data.columns and 'resolution_time' in data.columns:
    # 1. Calculate the correlation matrix
    correlation_matrix = data[['response.code', 'resolution_time']].corr()
    print("\nCorrelation Matrix:")
    print(correlation_matrix)

    # 2. Visualize the correlation matrix using a heatmap
    plt.figure(figsize=(8, 6))
    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
    plt.title('Correlation Heatmap')
    plt.show()

    # 3. Scatter plot to visualize the relationship
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x='resolution_time', y='response.code', data=data, alpha=0.6)
    plt.title('Customer Satisfaction vs. Ticket Resolution Time')
    plt.xlabel('Ticket Resolution Time')
    plt.ylabel('Customer Satisfaction (Response Code)')
    plt.axhline(200, color='red', linestyle='--', label='Success Threshold')
    plt.legend()
    plt.show()

else:
    print("One or both of the columns 'response.code' and 'resolution_time' do not exist in the dataset.")
```

```
One or both of the columns 'response.code' and 'resolution_time' do not exist in the dataset.
```

Trend Analysis

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset
file_path = '/content/customer_support_tickets (1).csv'
data = pd.read_csv(file_path)
```

```
# Display the first few rows of the dataset
print("Initial Data:")
print(data.head())
```

```
Initial Data:
   Ticket ID        Customer Name              Customer Email  Customer Age  \
0          1        Marisa Obrien  carrollallison@example.com            32
1          2         Jessica Rios    clarkeashley@example.com            42
2          3  Christopher Robbins   gonzalestracy@example.com            48
3          4     Christina Dillon    bradleyolson@example.org            27
4          5    Alexander Carroll     bradleymark@example.com            67

  Customer Gender Product Purchased Date of Purchase      Ticket Type  \
0          Other        GoPro Hero       2021-03-22  Technical issue
1         Female       LG Smart TV       2021-05-22  Technical issue
2          Other          Dell XPS       2020-07-14  Technical issue
3         Female  Microsoft Office       2020-11-13  Billing inquiry
```

```
4          Female  Autodesk AutoCAD       2020-02-04  Billing inquiry

             Ticket Subject  \
0             Product setup
1  Peripheral compatibility
2           Network problem
3            Account access
4                 Data loss

                                  Ticket Description  \
0  I'm having an issue with the {product_purchase...
1  I'm having an issue with the {product_purchase...
2  I'm facing a problem with my {product_purchase...
3  I'm having an issue with the {product_purchase...
4  I'm having an issue with the {product_purchase...

                Ticket Status                                    Resolution  \
0  Pending Customer Response                                            NaN
1  Pending Customer Response                                            NaN
2                     Closed   Case maybe show recently my computer follow.
3                     Closed  Try capital clearly never color toward story.
4                     Closed                    West decision evidence bit.

  Ticket Priority Ticket Channel  First Response Time   Time to Resolution  \
0        Critical   Social media  2023-06-01 12:15:36                  NaN
1        Critical           Chat  2023-06-01 16:45:38                  NaN
2             Low   Social media  2023-06-01 11:14:38  2023-06-01 18:05:38
3             Low   Social media  2023-06-01 07:29:40  2023-06-01 01:57:40
4             Low          Email  2023-06-01 00:12:42  2023-06-01 19:53:42

   Customer Satisfaction Rating
0                           NaN
1                           NaN
2                           3.0
3                           3.0
4                           1.0
```

```
# Convert 'creation_time' to datetime format
if 'creation_time' in data.columns:
    data['creation_time'] = pd.to_datetime(data['creation_time'])


if 'creation_time' in data.columns:
    data['creation_time'] = pd.to_datetime(data['creation_time'])

    # Set the creation_time as the index
    data.set_index('creation_time', inplace=True)


# Convert 'creation_time' to datetime format
if 'creation_time' in data.columns:
    data['creation_time'] = pd.to_datetime(data['creation_time'])

    # Set the creation_time as the index
    data.set_index('creation_time', inplace=True)

    # 1. Resample the data to monthly frequency and count the number of tickets
    monthly_tickets = data.resample('M').size()

    # 2. Calculate average customer satisfaction ratings (assuming 'response.code' is the satisfaction rating)
    monthly_satisfaction = data.resample('M')['response.code'].mean()

    # 3. Plotting the trends
    plt.figure(figsize=(14, 7))

    # Plot ticket volume
    plt.subplot(2, 1, 1)
    monthly_tickets.plot(kind='bar', color='skyblue', alpha=0.7)
    plt.title('Monthly Ticket Volume')
    plt.xlabel('Month')
    plt.ylabel('Number of Tickets')
    plt.xticks(rotation=45)
    plt.grid()

    # Plot average customer satisfaction
    plt.subplot(2, 1, 2)
    monthly_satisfaction.plot(kind='line', marker='o', color='orange')
    plt.title('Average Customer Satisfaction Over Time')
    plt.xlabel('Month')
    plt.ylabel('Average Satisfaction Rating (Response Code)')
```

```
    plt.axhline(200, color='red', linestyle='--', label='Success Threshold')
    plt.legend()
    plt.xticks(rotation=45)
    plt.grid()

    plt.tight_layout()
    plt.show()

else:
    print("The dataset does not contain a 'creation_time' column.")
```

```
The dataset does not contain a 'creation_time' column.
```

Predictive Modeling

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline


# Load the dataset
file_path = '/content/customer_support_tickets (1).csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
print("Initial Data:")
print(data.head())
```

```
Initial Data:
   Ticket ID        Customer Name              Customer Email  Customer Age  \
0          1        Marisa Obrien  carrollallison@example.com            32
1          2         Jessica Rios    clarkeashley@example.com            42
2          3  Christopher Robbins   gonzalestracy@example.com            48
3          4     Christina Dillon    bradleyolson@example.org            27
4          5    Alexander Carroll     bradleymark@example.com            67

  Customer Gender Product Purchased Date of Purchase      Ticket Type  \
0          Other        GoPro Hero       2021-03-22  Technical issue
1         Female       LG Smart TV       2021-05-22  Technical issue
2          Other          Dell XPS       2020-07-14  Technical issue
3         Female  Microsoft Office       2020-11-13  Billing inquiry
4         Female  Autodesk AutoCAD       2020-02-04  Billing inquiry

             Ticket Subject  \
0             Product setup
1  Peripheral compatibility
2           Network problem
3            Account access
4                 Data loss

                                  Ticket Description  \
0  I'm having an issue with the {product_purchase...
1  I'm having an issue with the {product_purchase...
2  I'm facing a problem with my {product_purchase...
3  I'm having an issue with the {product_purchase...
4  I'm having an issue with the {product_purchase...

               Ticket Status                                      Resolution  \
0  Pending Customer Response                                             NaN
1  Pending Customer Response                                             NaN
2                     Closed   Case maybe show recently my computer follow.
3                     Closed  Try capital clearly never color toward story.
4                     Closed                    West decision evidence bit.

  Ticket Priority Ticket Channel  First Response Time   Time to Resolution  \
0        Critical   Social media  2023-06-01 12:15:36                  NaN
1        Critical           Chat  2023-06-01 16:45:38                  NaN
2             Low   Social media  2023-06-01 11:14:38  2023-06-01 18:05:38
3             Low   Social media  2023-06-01 07:29:40  2023-06-01 01:57:40
4             Low          Email  2023-06-01 00:12:42  2023-06-01 19:53:42

   Customer Satisfaction Rating
0                           NaN
1                           NaN
```

```
2                           3.0
3                           3.0
4                           1.0
```

```
# Check for relevant columns
if 'response.code' in data.columns and 'resolution_time' in data.columns:
    # 1. Data Preparation
    # Drop rows with missing values in relevant columns
    data = data[['response.code', 'resolution_time', 'ticket_type', 'customer_gender']].dropna()

    # 2. Feature Selection
    X = data[['resolution_time', 'ticket_type', 'customer_gender']]  # Features
    y = data['response.code']  # Target variable

    # 3. Train-Test Split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # 4. Preprocessing and Model Training
    # Create a preprocessing pipeline
    preprocessor = ColumnTransformer(
        transformers=[
            ('num', StandardScaler(), ['resolution_time']),
            ('cat', OneHotEncoder(), ['ticket_type', 'customer_gender'])
        ])

    # Create a pipeline that first transforms the data and then fits the model
    model = Pipeline(steps=[
        ('preprocessor', preprocessor),
        ('regressor', RandomForestRegressor(random_state=42))
    ])

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

    # 5. Model Evaluation
    # Make predictions
    y_pred = model.predict(X_test)

    # Calculate performance metrics
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"\nMean Squared Error: {mse:.2f}")
    print(f"R^2 Score: {r2:.2f}")

else:
    print("The dataset does not contain the required columns for prediction.")
```

```
The dataset does not contain the required columns for prediction.
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.