

```
In [16]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
In [4]: # Load the data
data = pd.read_csv('customer_support_tickets.csv')

# View first few rows
print(data.head())

# Check data types and missing values
print(data.info())
print(data.describe())
```

| | 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 |

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 8469 entries, 0 to 8468
```

```
Data columns (total 17 columns):
```

| # | Column | Non-Null Count | Dtype |
|---|----------------|----------------|--------|
| 0 | Ticket ID | 8469 non-null | int64 |
| 1 | Customer Name | 8469 non-null | object |
| 2 | Customer Email | 8469 non-null | object |

```

3  Customer Age                8469 non-null  int64
4  Customer Gender             8469 non-null  object
5  Product Purchased           8469 non-null  object
6  Date of Purchase            8469 non-null  object
7  Ticket Type                 8469 non-null  object
8  Ticket Subject              8469 non-null  object
9  Ticket Description           8469 non-null  object
10 Ticket Status                8469 non-null  object
11 Resolution                  2769 non-null  object
12 Ticket Priority              8469 non-null  object
13 Ticket Channel               8469 non-null  object
14 First Response Time          5650 non-null  object
15 Time to Resolution           2769 non-null  object
16 Customer Satisfaction Rating 2769 non-null  float64

```

dtypes: float64(1), int64(2), object(14)

memory usage: 1.1+ MB

None

| | 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 |

In []: *#Handle Missing Values*

```

data.replace([np.inf, -np.inf], np.nan, inplace=True)

# Drop rows where satisfaction rating is missing (target column)
data = data.dropna(subset=['Customer Satisfaction Rating'])

# Optionally fill or drop other missing values
data = data.dropna()

```

In []: *#Encode Categorical Variables*

```

label_cols = ['Customer Gender', 'Product Purchased', 'Ticket Type',
              'Ticket Status', 'Ticket Priority', 'Ticket Channel']

label_encoders = {}
for col in label_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

```

In [7]: *#Feature Engineering*

```

X = data[['Customer Age', 'Customer Gender', 'Product Purchased',
          'Ticket Type', 'Ticket Status', 'Ticket Priority', 'Ticket Channel']]
y = data['Customer Satisfaction Rating']

```

In [8]: *#Train and Test Split*

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42)
```

In [9]: *#Feature Scaling*

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [10]: *#Model Building*

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

Out[10]:

RandomForestClassifier

RandomForestClassifier(random_state=42)

In [11]: *#Model Evaluation*

```
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

Accuracy: 0.19855595667870035

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1.0 | 0.17 | 0.17 | 0.17 | 168 |
| 2.0 | 0.17 | 0.17 | 0.17 | 174 |
| 3.0 | 0.24 | 0.25 | 0.25 | 175 |
| 4.0 | 0.21 | 0.20 | 0.20 | 162 |
| 5.0 | 0.20 | 0.21 | 0.20 | 152 |
| accuracy | | | 0.20 | 831 |
| macro avg | 0.20 | 0.20 | 0.20 | 831 |
| weighted avg | 0.20 | 0.20 | 0.20 | 831 |

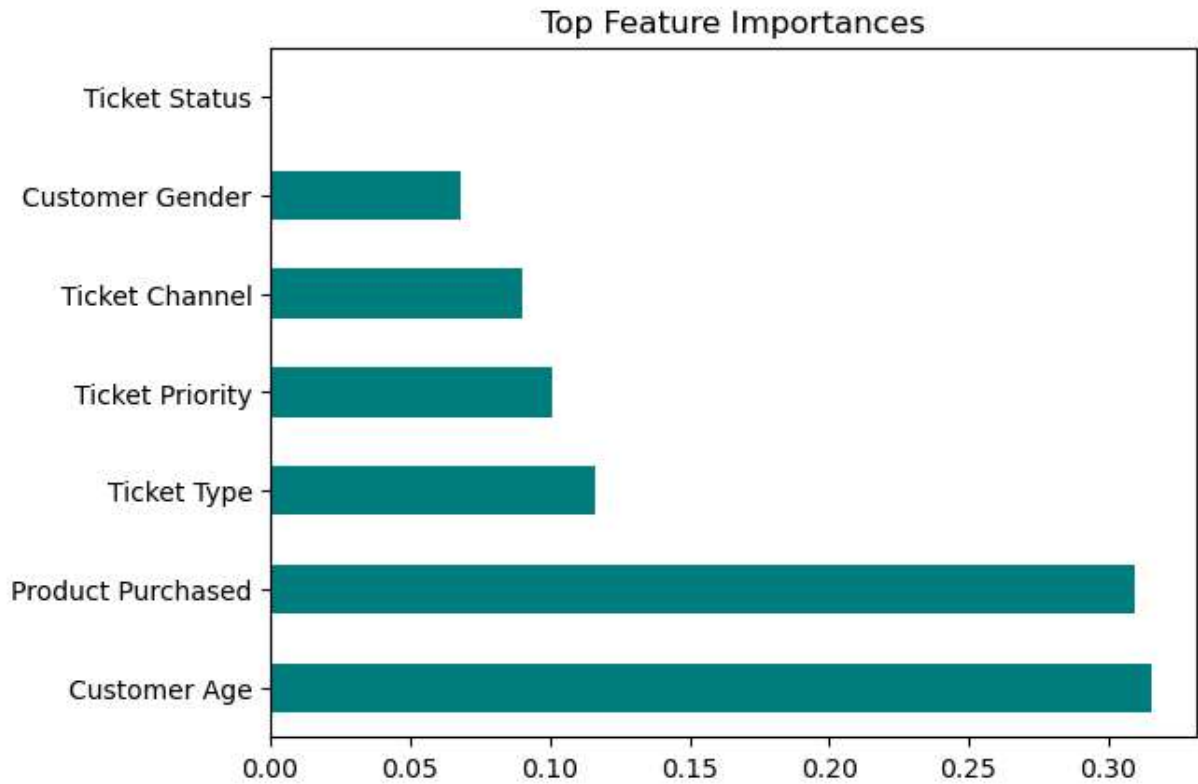
Confusion Matrix:

```
[[28 38 33 31 38]
 [38 29 48 31 28]
 [35 33 44 32 31]
 [37 34 27 32 32]
 [28 34 31 27 32]]
```

In [12]: *#Feature Importance Visualization*

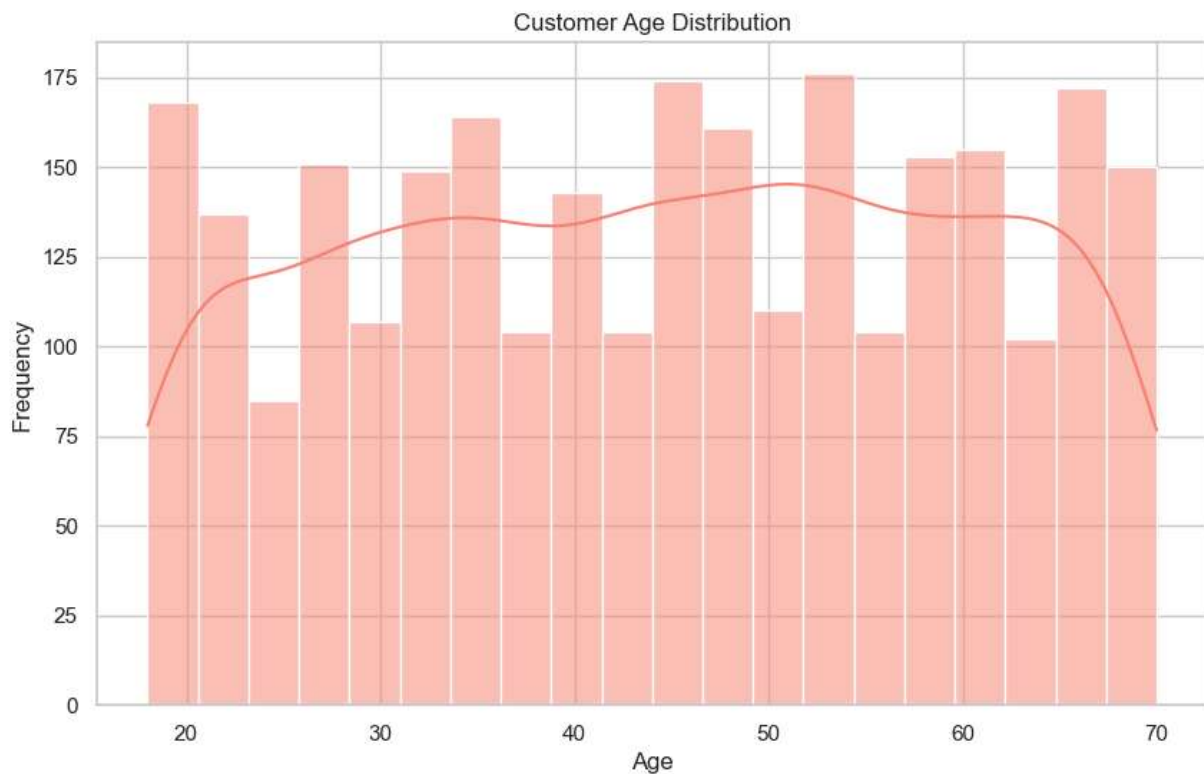
```
feature_importances = pd.Series(model.feature_importances_, index=X.columns)
feature_importances.nlargest(10).plot(kind='barh', color='teal')
```

```
plt.title('Top Feature Importances')  
plt.show()
```



In [14]: *#Age Distribution*

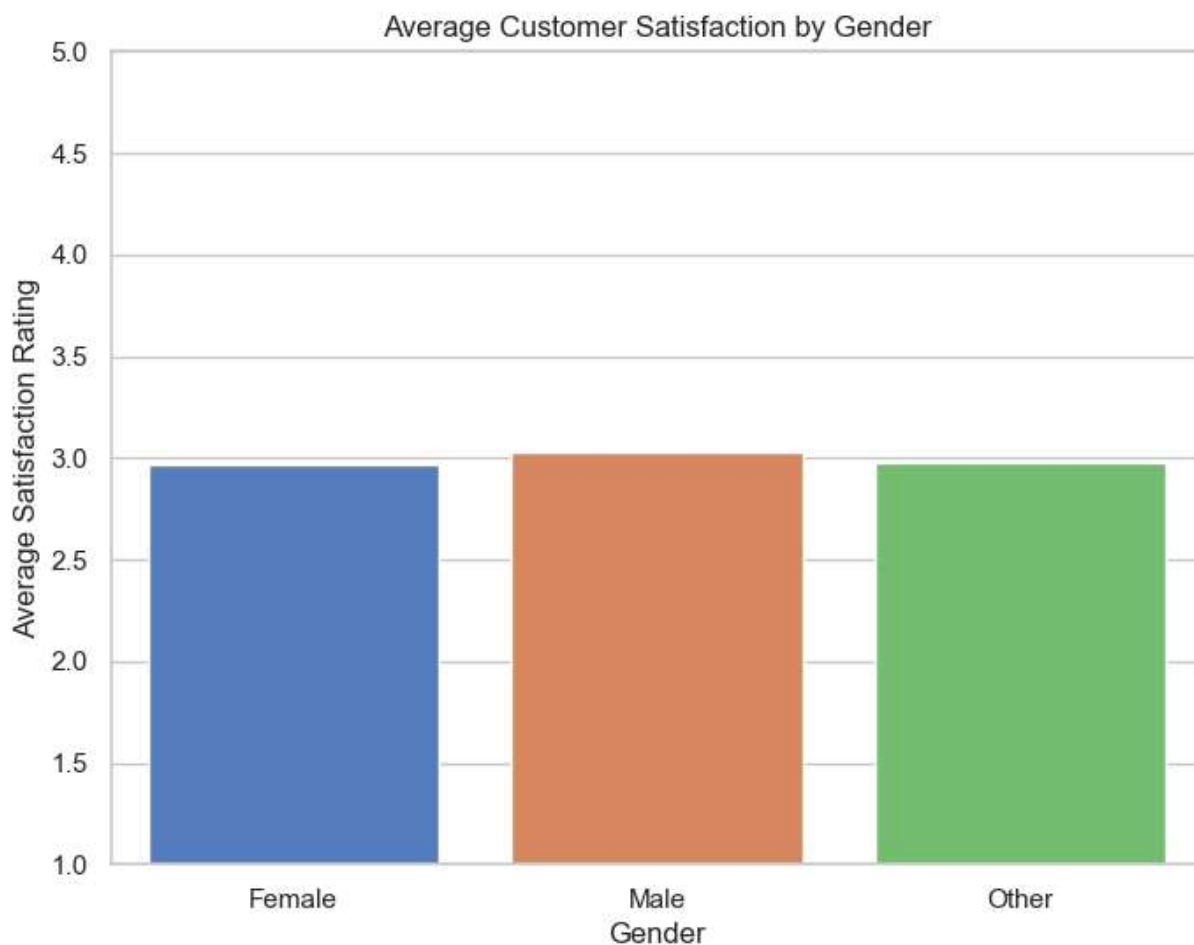
```
plt.figure(figsize=(10, 6))  
sns.histplot(data['Customer Age'], bins=20, kde=True, color='salmon')  
plt.title('Customer Age Distribution')  
plt.xlabel('Age')  
plt.ylabel('Frequency')  
plt.show()
```



In [17]: *#Satisfaction Ratings By Gender*

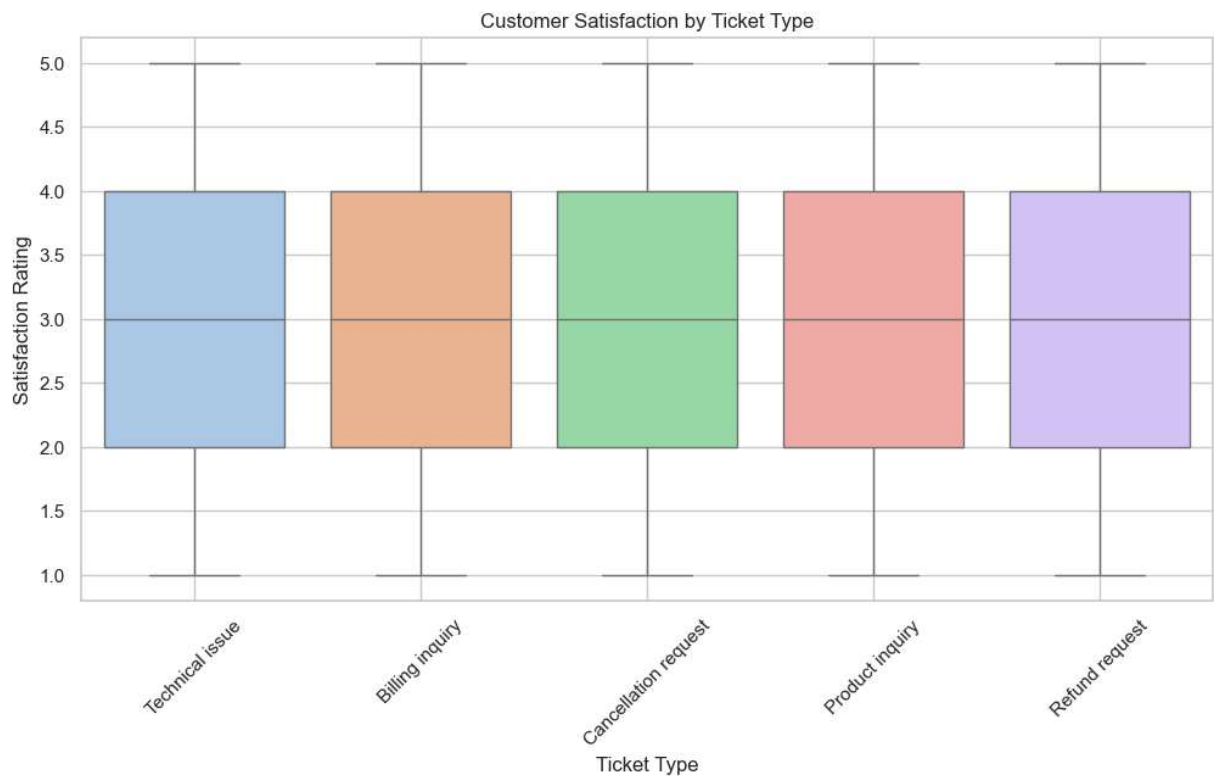
```
avg_satisfaction = data.groupby('Customer Gender')['Customer Satisfaction Rating'].

plt.figure(figsize=(8, 6))
sns.barplot(x='Customer Gender', y='Customer Satisfaction Rating', data=avg_satisfa
plt.title('Average Customer Satisfaction by Gender')
plt.xlabel('Gender')
plt.ylabel('Average Satisfaction Rating')
plt.ylim(1, 5) # Rating scale is 1 to 5
plt.show()
```



In [18]: *#Ticket Type vs Satisfaction*

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Ticket Type', y='Customer Satisfaction Rating', data=data, palette='
plt.title('Customer Satisfaction by Ticket Type')
plt.xlabel('Ticket Type')
plt.ylabel('Satisfaction Rating')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



```
In [19]: #Ticket Trends Overtime

# Ensure the date is parsed
data['YearMonth'] = data['Date of Purchase'].dt.to_period('M')

# Group by Year-Month
ticket_trends = data.groupby('YearMonth').size()

plt.figure(figsize=(12, 6))
ticket_trends.plot(kind='line', marker='o', color='steelblue')
plt.title('Customer Support Ticket Trends Over Time')
plt.xlabel('Year-Month')
plt.ylabel('Number of Tickets')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```