untitled10

July 16, 2025

0.0.1 Customer Satisfaction Prediction

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import␣
 ↪mean_absolute_error,mean_squared_error,r2_score,accuracy_score,classification_report,confus
from sklearn.preprocessing import StandardScaler,LabelEncoder
from sklearn.ensemble import RandomForestRegressor,RandomForestClassifier
import random
```

[3]:
```
data = pd.read_csv('customer_support_tickets.csv')
```

[6]:
```
data.head(5)
```

[6]:

	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
```

[8]:
```
# Display basic info about the dataset
data.info()
```

```
<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
```

0.0.2 Data Preprocessing

[6]:
```
# Handling the missing values
data.isnull().sum()
```

[6]:
```
Ticket ID                         0
Customer Name                     0
Customer Email                    0
Customer Age                      0
Customer Gender                   0
Product Purchased                 0
Date of Purchase                  0
Ticket Type                       0
Ticket Subject                    0
Ticket Description                0
Ticket Status                     0
Resolution                     5700
Ticket Priority                   0
Ticket Channel                    0
First Response Time            2819
Time to Resolution             5700
Customer Satisfaction Rating   5700
dtype: int64
```

[8]:
```
data = data.dropna()
```

[28]:
```
# Encoding categorical variable
label_encoders = {}
for column in data.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    data[column] = label_encoders[column].fit_transform(data[column])
```

[30]:
```
# Define feature and Target variable
X = data.drop(['Customer Satisfaction Rating'],axis=1)
y = data['Customer Satisfaction Rating']
```

[34]:
```
# Split the dataset
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.
↪3,random_state=42)
```

[36]:
```
# Feature Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[42]:
```
# Model Building
# Train a random forest Classifier
rfc = RandomForestClassifier(random_state=42)
rfc.fit(X_train,y_train)
```

[42]:
```
RandomForestClassifier(random_state=42)
```

[44]:
```
# Predict on the test set
y_pred = rfc.predict(X_test)
```

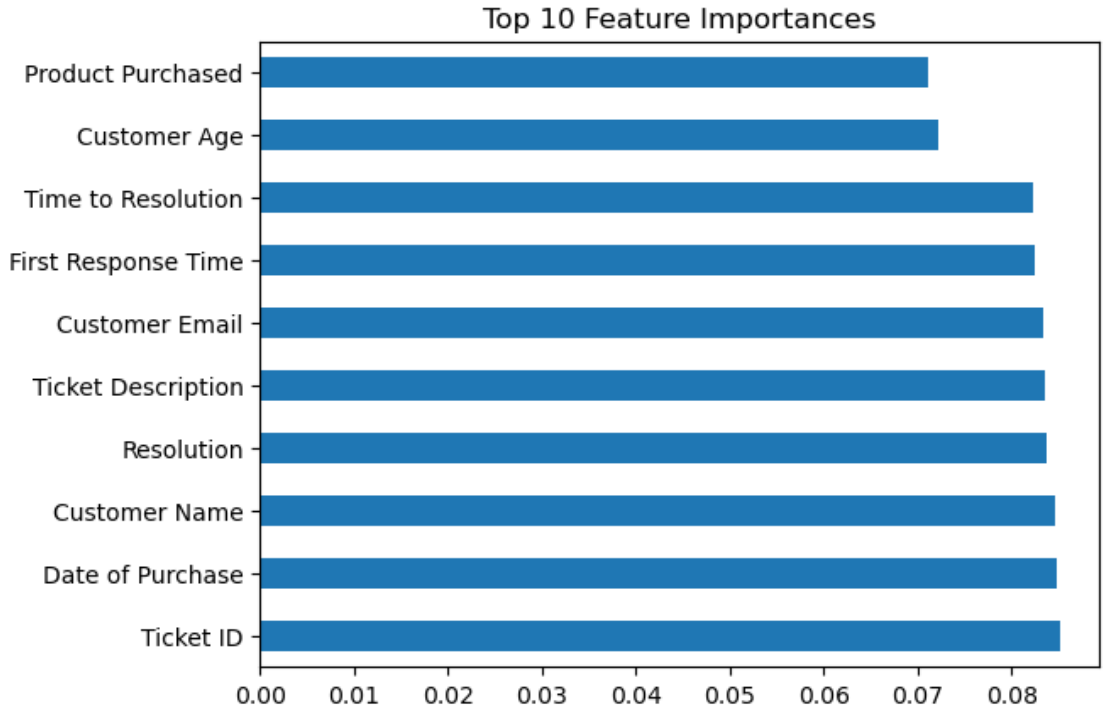
[50]:
```
# Model Evalution
print("Accuracy",accuracy_score(y_test,y_pred))
print("Classification report:\n",classification_report(y_test,y_pred))
```

```
Accuracy 0.2286401925391095
Classification report:
               precision    recall  f1-score   support

         1.0       0.24      0.24      0.24       168
         2.0       0.22      0.20      0.21       174
         3.0       0.25      0.29      0.26       175
         4.0       0.25      0.22      0.23       162
         5.0       0.19      0.20      0.19       152

    accuracy                           0.23       831
   macro avg       0.23      0.23      0.23       831
weighted avg       0.23      0.23      0.23       831
```

[54]:
```
print("Confussion Matrix:\n",confusion_matrix(y_test,y_pred))
```

```
Confussion Matrix:
 [[40 27 45 24 32]
 [38 35 51 22 28]
 [37 32 50 17 39]
 [32 37 29 35 29]
 [23 27 29 43 30]]
```

[56]:
```
# Feature Importance
feature_importances = pd.Series(rfc.feature_importances_,index=X.columns)
feature_importances.nlargest(10).plot(kind='barh')
plt.title('Top 10 Feature Importances')
plt.show()
```



[60]:
```
data.describe()
```

[60]:

	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

[62]:
```
data.columns
```

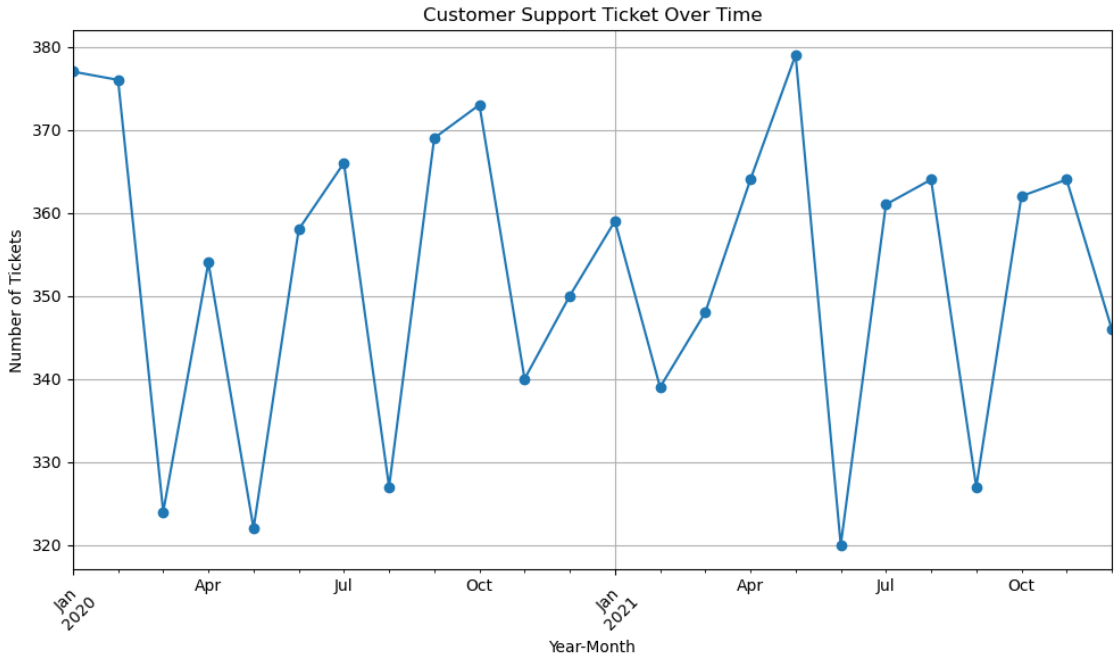
[62]:
```
Index(['Ticket ID', 'Customer Name', 'Customer Email', 'Customer Age',
       'Customer Gender', 'Product Purchased', 'Date of Purchase',
       'Ticket Type', 'Ticket Subject', 'Ticket Description', 'Ticket Status',
       'Resolution', 'Ticket Priority', 'Ticket Channel',
```

```
       'First Response Time', 'Time to Resolution',
       'Customer Satisfaction Rating'],
      dtype='object')
```

[66]:
```
# Analyze customer support ticket trends
common_issues = data['Ticket Subject'].value_counts().head(10)
print('Top 10 Common Issues')
print(common_issues)
```

```
Top 10 Common Issues
Ticket Subject
Refund request           576
Software bug             574
Product compatibility    567
Delivery problem         561
Hardware issue           547
Battery life             542
Network problem          539
Installation support     530
Product setup            529
Payment issue            526
Name: count, dtype: int64
```

[68]:
```
# Ploting ticket trends over time
data['Date of Purchase'] = pd.to_datetime(data['Date of Purchase'])
data['YearMonth'] = data['Date of Purchase'].dt.to_period('M')
ticket_trends = data.groupby('YearMonth').size()
plt.figure(figsize=(10,6))
ticket_trends.plot(kind='line',marker='o')
plt.title('Customer Support Ticket Over Time')
plt.xlabel('Year-Month')
plt.ylabel('Number of Tickets')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



[70]:
```
# Segment customers
# Segment base on ticket type
ticket_type_segmentation = data.groupby('Ticket Type').size()
print("\nSegmentation based on ticket types")
print(ticket_type_segmentation)
```

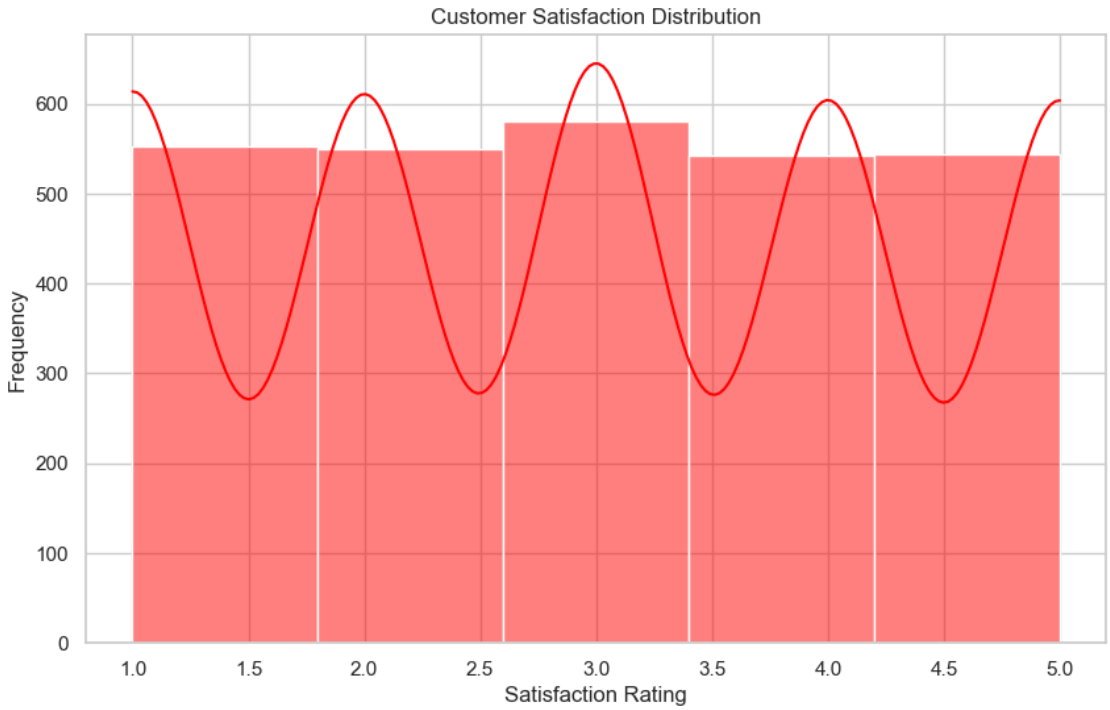
```
Segmentation based on ticket types
Ticket Type
Billing inquiry         1634
Cancellation request    1695
Product inquiry         1641
Refund request          1752
Technical issue         1747
dtype: int64
```

[72]:
```
# Segment based on satisfaction levels
satisfaction_segmentation = data.groupby('Customer Satisfaction Rating').size()
print("\nSegmentation based on Customer Satisfaction Levels")
print(satisfaction_segmentation)
```

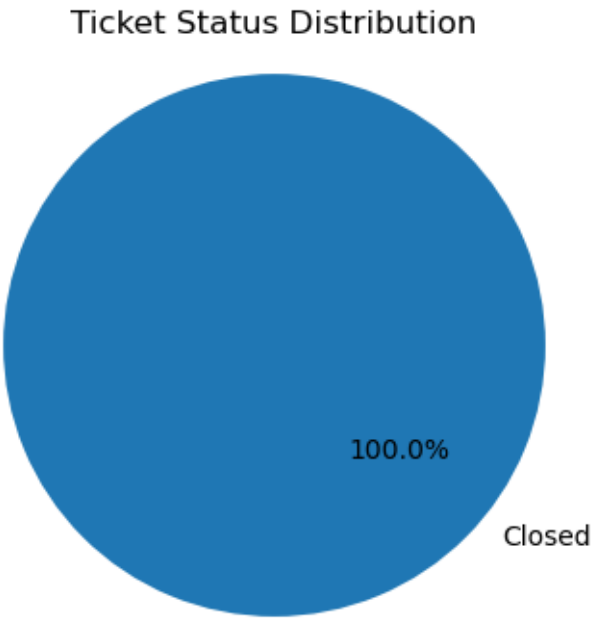
```
Segmentation based on Customer Satisfaction Levels
Customer Satisfaction Rating
1.0    553
2.0    549
```

```
3.0    580
4.0    543
5.0    544
dtype: int64
```

[76]:
```
# Set up the ploting aesthtics
sns.set(style="whitegrid")

# Customer Satisfaction Distribution
plt.figure(figsize=(10,6))
sns.histplot(data['Customer Satisfaction Rating'],bins=5,kde=True,color='red')
plt.title('Customer Satisfaction Distribution')
plt.xlabel('Satisfaction Rating')
plt.ylabel('Frequency')
plt.show()
```



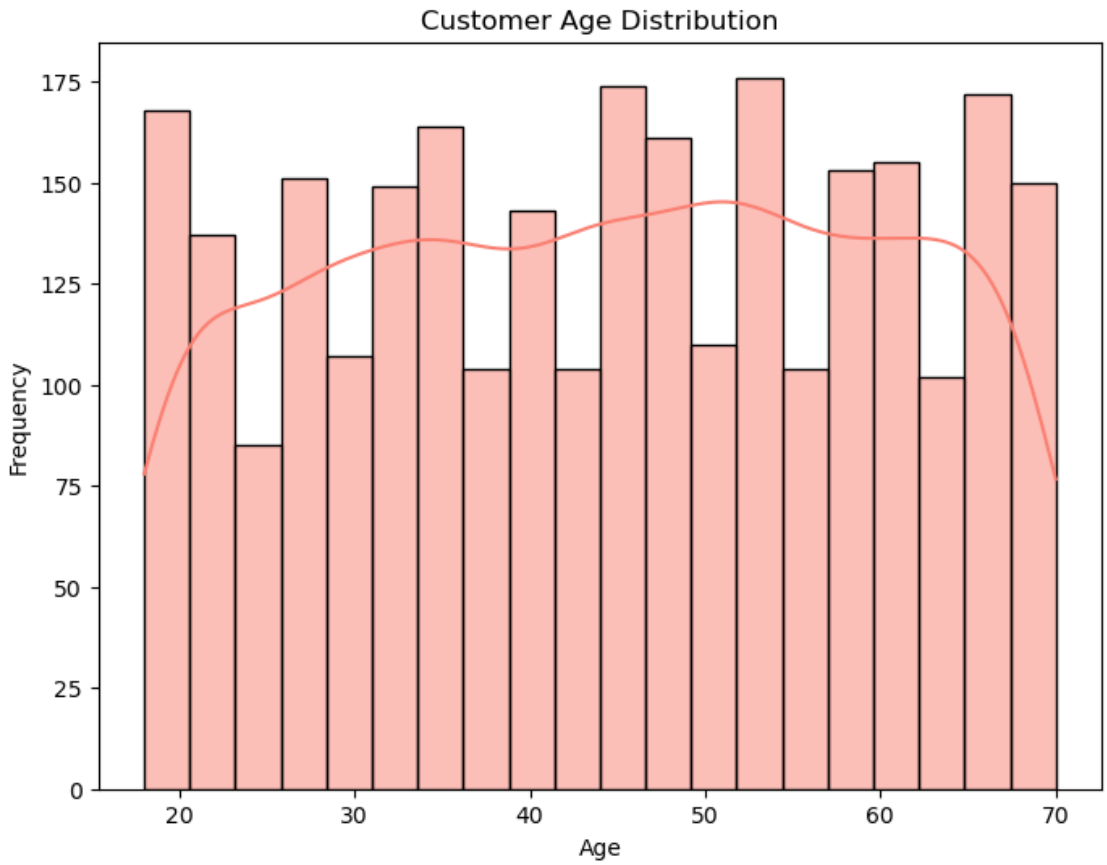
[15]:
```
# Ticket Status Distribution
ticket_status_dis = data['Ticket Status'].value_counts()
plt.figure(figsize=(4,4))
plt.pie(ticket_status_dis,labels=ticket_status_dis.index,autopct='%1.
  ↪1f%%',startangle=140)
plt.title('Ticket Status Distribution')
plt.axis('equal')
plt.show()
```



[13]:
```
data['Ticket Status'].value_counts()
```

[13]:
```
Ticket Status
Closed    2769
Name: count, dtype: int64
```

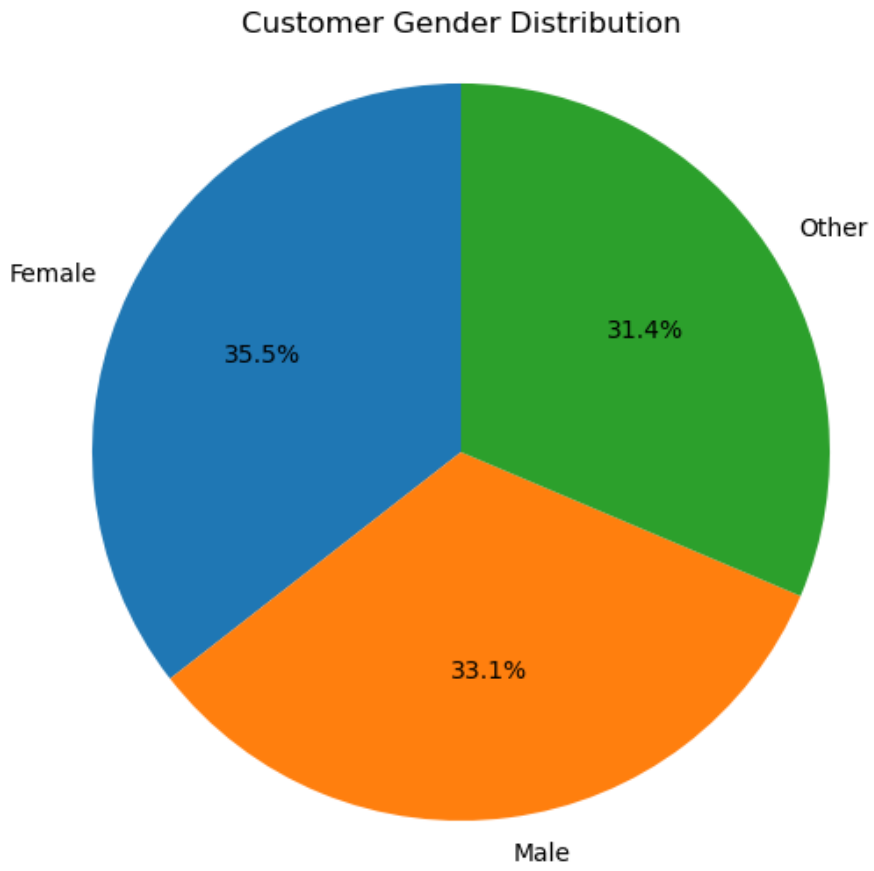
[17]:
```
# Customer Age distribution
plt.figure(figsize=(8,6))
sns.histplot(data['Customer Age'],bins=20,kde=True,color='salmon')
plt.title('Customer Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



[19]:
```
data['Customer Gender'].value_counts()
```

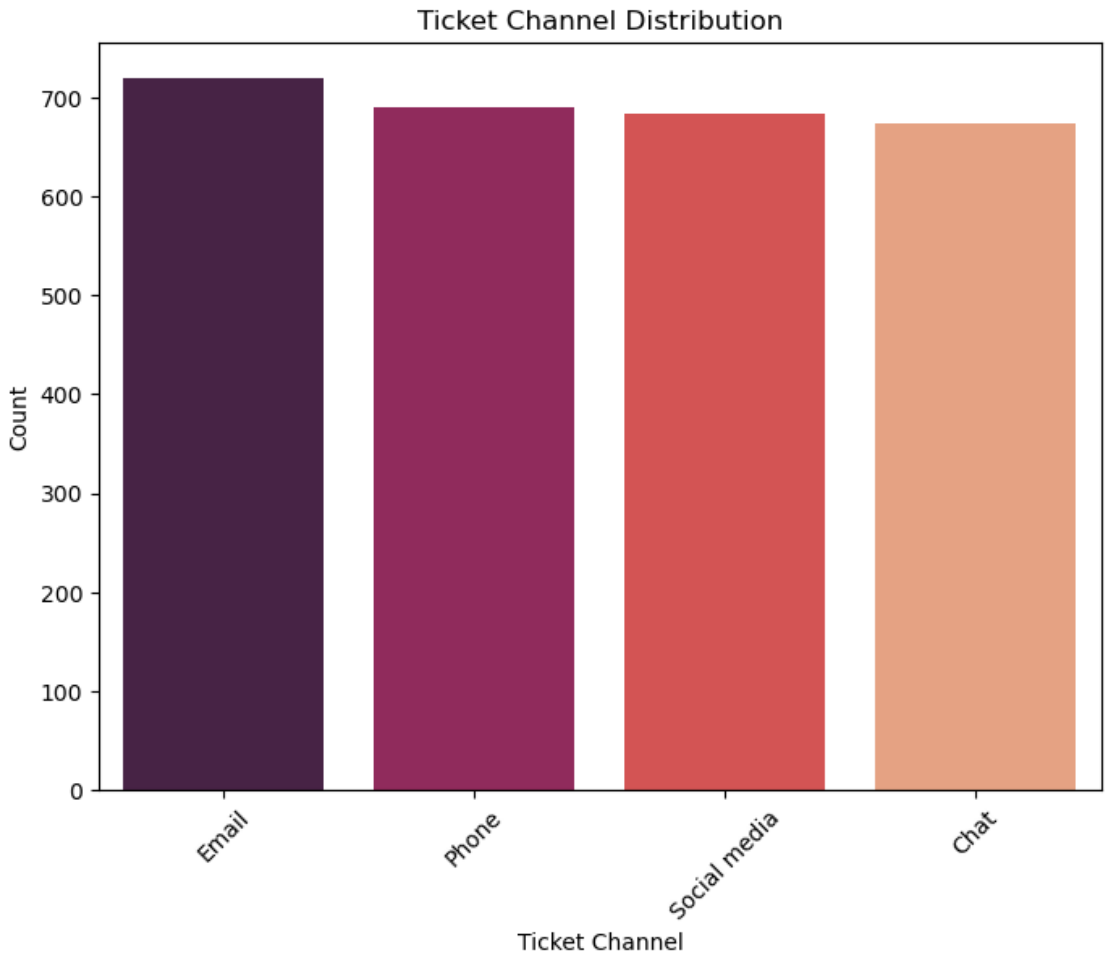
[19]:
```
Customer Gender
Female    984
Male      916
Other     869
Name: count, dtype: int64
```

[21]:
```
# Customer Gender Distribution
customer_gender_dis = data['Customer Gender'].value_counts()
plt.figure(figsize=(8,6))
plt.pie(customer_gender_dis,labels=customer_gender_dis.index,autopct='%1.
  ↪1f%%',startangle=90)
plt.title('Customer Gender Distribution')
plt.axis('equal')
plt.show()
```



[27]:
```
# Ticket Channel Distribution
ticket_channel_dis = data['Ticket Channel'].value_counts()
plt.figure(figsize=(8,6))
sns.barplot(x=ticket_channel_dis.index,y=ticket_channel_dis,palette='rocket')
plt.title('Ticket Channel Distribution')
plt.xlabel('Ticket Channel')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

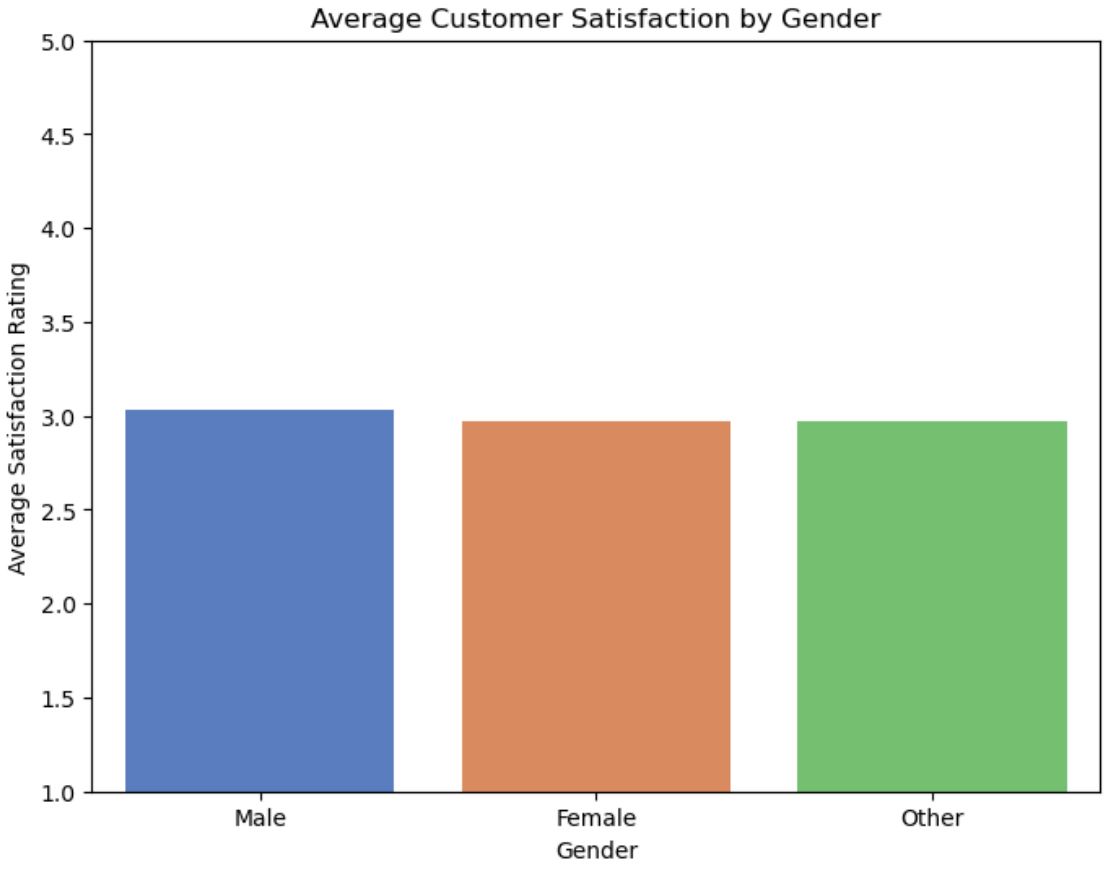
```
C:\Users\aa\AppData\Local\Temp\ipykernel_6840\1268846635.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=ticket_channel_dis.index,y=ticket_channel_dis,palette='rocket')
```



[29]:
```
# Average Customer Satisfaction By Gender
average_satisfaction = data.groupby('Customer Gender')['Customer Satisfaction␣
 ↪Rating'].mean().reset_index()
plt.figure(figsize=(8,6))
sns.barplot(x='Customer Gender',y='Customer Satisfaction␣
 ↪Rating',data=average_satisfaction,palette='muted',order=['Male','Female','Other'])
plt.title('Average Customer Satisfaction by Gender')
plt.xlabel('Gender')
plt.ylabel('Average Satisfaction Rating')
plt.ylim(1,5)
plt.show()
```

```
C:\Users\aa\AppData\Local\Temp\ipykernel_6840\1565998600.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

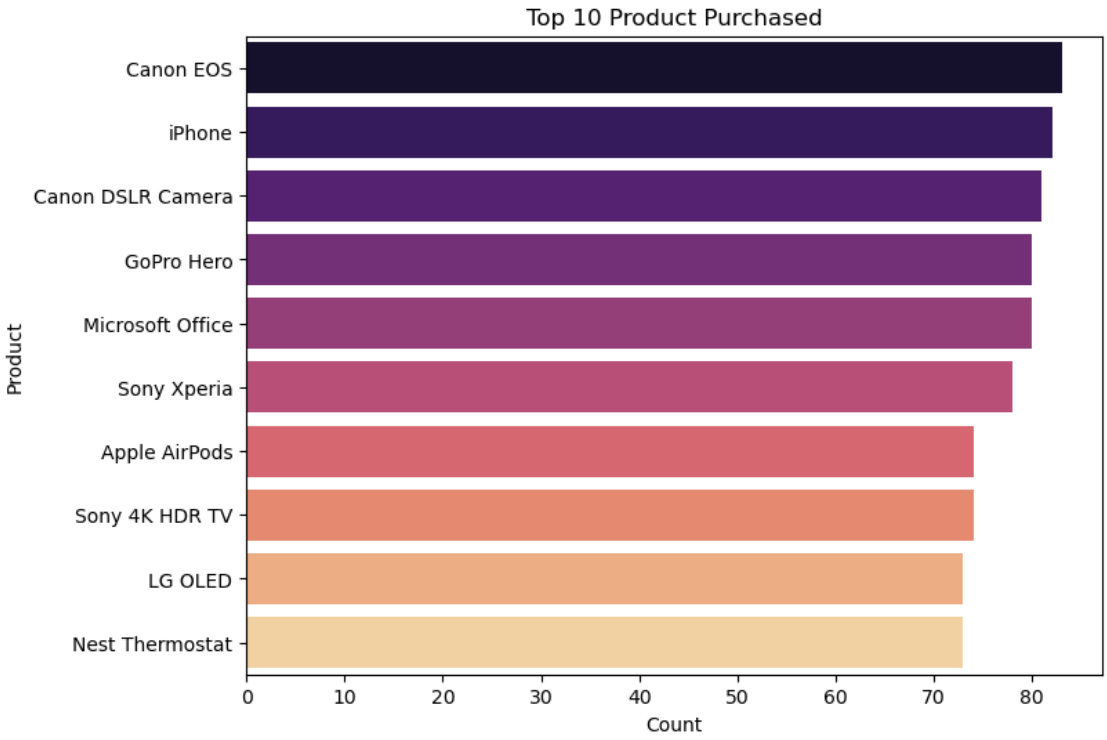
```
  sns.barplot(x='Customer Gender',y='Customer Satisfaction Rating',data=average_
satisfaction,palette='muted',order=['Male','Female','Other'])
```



[31]:
```
# Product Purchased Distribution
plt.figure(figsize=(8,6))
product_purchased_dis = data['Product Purchased'].value_counts().head(10)
sns.barplot(x=product_purchased_dis,y=product_purchased_dis.
  ↪index,palette='magma')
plt.title('Top 10 Product Purchased')
plt.xlabel('Count')
plt.ylabel('Product')
plt.show()
```

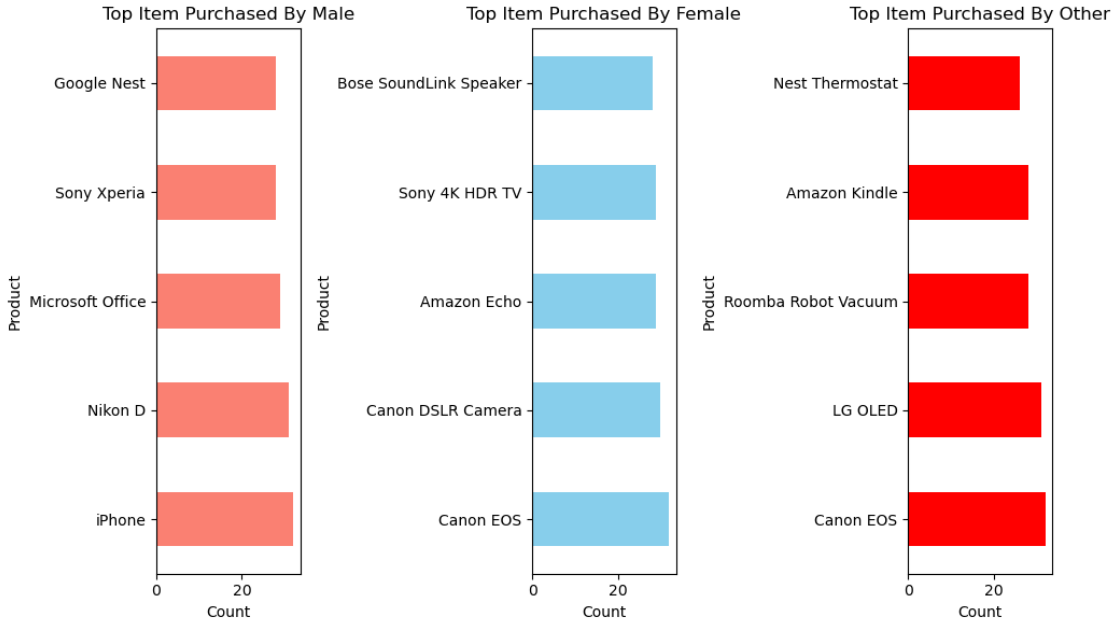
```
C:\Users\aa\AppData\Local\Temp\ipykernel_6840\4046523498.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.
```

```
  sns.barplot(x=product_purchased_dis,y=product_purchased_dis.index,palette='mag
ma')
```

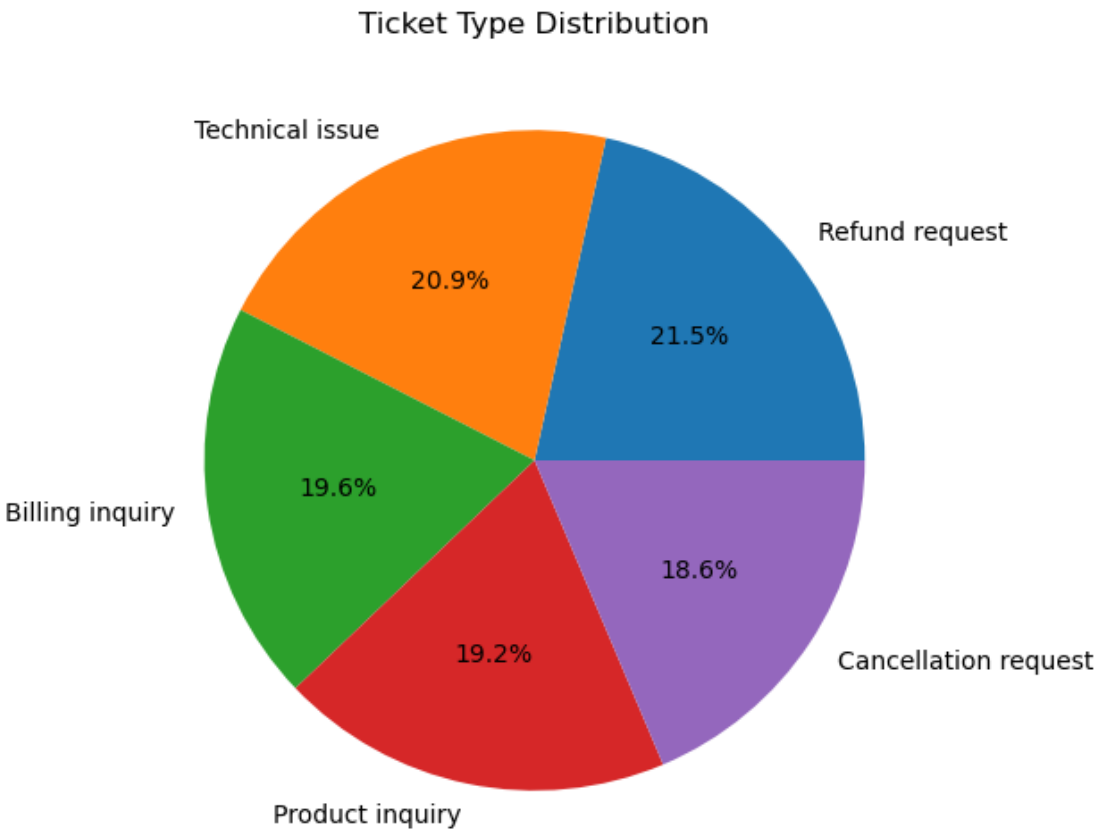


[12]:
```
# Top Items Purchased by Gender
# Top Item Purchased By Male
plt.figure(figsize=(10,6))
plt.subplot(1,3,1)
top_item_male = data[data['Customer Gender']=='Male']['Product Purchased'].
 ↪value_counts().head(5)
top_item_male.plot(kind='barh',color='salmon')
plt.title('Top Item Purchased By Male')
plt.xlabel('Count')
plt.ylabel('Product')

# Top Item Purchased By Female

plt.subplot(1,3,2)
top_item_female = data[data['Customer Gender']=='Female']['Product Purchased'].
 ↪value_counts().head(5)
top_item_female.plot(kind='barh',color='skyblue')
plt.title('Top Item Purchased By Female')
plt.xlabel('Count')
plt.ylabel('Product')
```

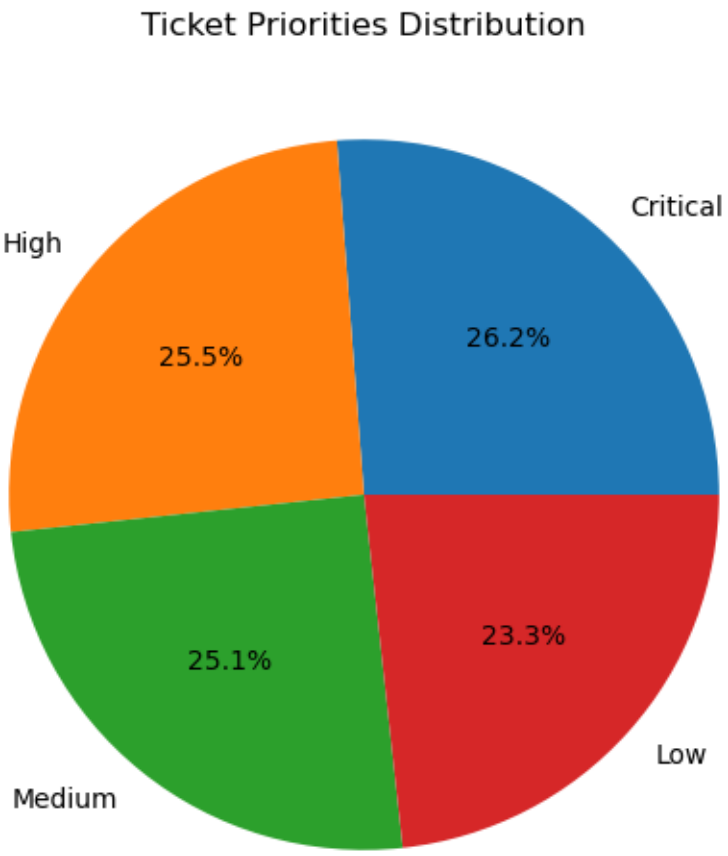
```
# Top Item Purchased By Other

plt.subplot(1,3,3)
top_item_other = data[data['Customer Gender']=='Other']['Product Purchased'].
 ↪value_counts().head(5)
top_item_other.plot(kind='barh',color='red')
plt.title('Top Item Purchased By Other')
plt.xlabel('Count')
plt.ylabel('Product')
plt.tight_layout()
plt.show()
```



[14]:
```
# Count Ticket Type
ticket_type_dis = data['Ticket Type'].value_counts()

# Plot
plt.figure(figsize=(8,6))
ticket_type_dis.plot(kind='pie',autopct='%1.1f%%',color=['red','green','blue'])
plt.title('Ticket Type Distribution')
plt.ylabel('')
plt.show()
```



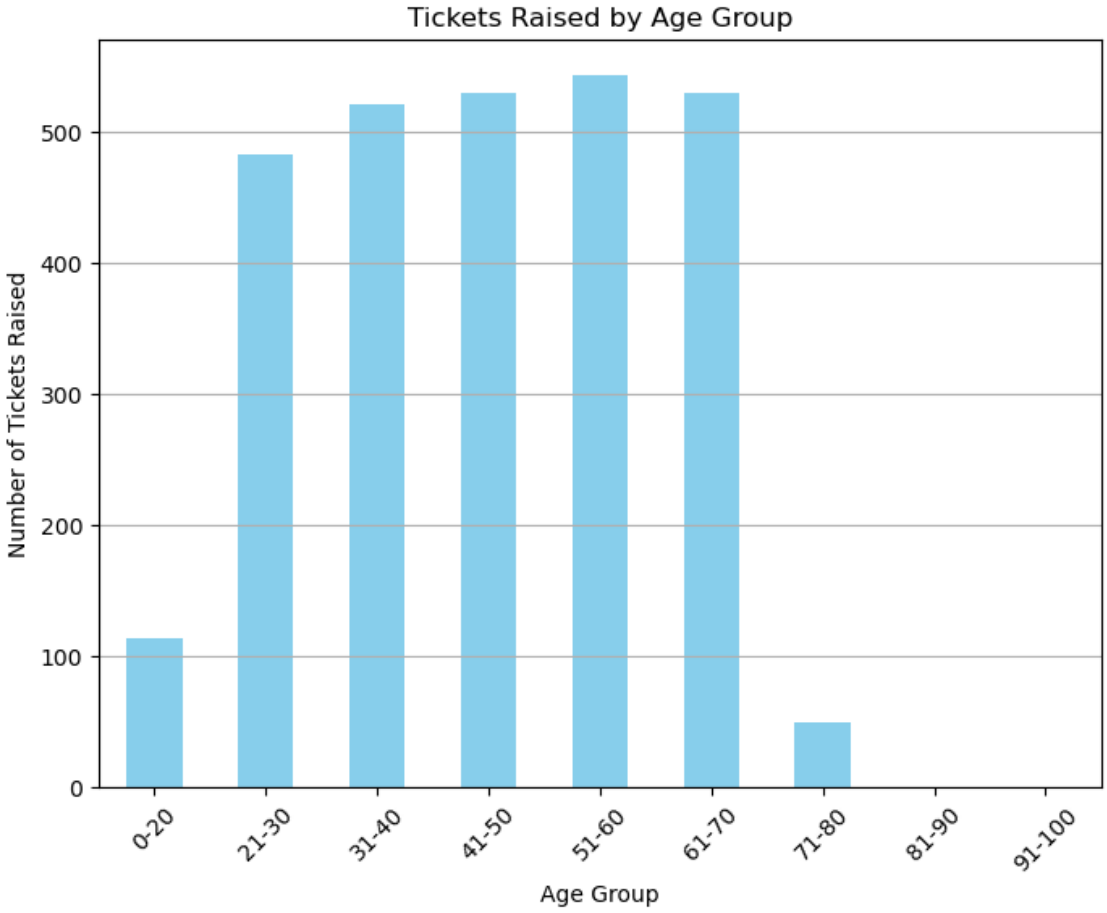
[18]:
```
# Count Ticket Priorities
ticket_priorities_dis = data['Ticket Priority'].value_counts()

# Plot
plt.figure(figsize=(8,6))
ticket_priorities_dis.plot(kind='pie',autopct='%1.
 ↪1f%%',color=['lightblue','lightgreen','lightsalmon','skyblue'])
plt.title('Ticket Priorities Distribution')
plt.ylabel('')
plt.show()
```



[20]:
```
# Define Age Group
bins = [0,20,30,40,50,60,70,80,90,100]
labels =␣
 ↪['0-20','21-30','31-40','41-50','51-60','61-70','71-80','81-90','91-100']

# Categorize Customer into Age Group
data['Age Group'] = pd.cut(data['Customer␣
 ↪Age'],bins=bins,labels=labels,right=False)
# Calculate number of tickets raised by each age group
tickets_by_age_group = data.groupby('Age Group').size()
# Plot
plt.figure(figsize=(8,6))
tickets_by_age_group.plot(kind='bar',color='skyblue')
plt.title('Tickets Raised by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Number of Tickets Raised')
plt.xticks(rotation=45)
```

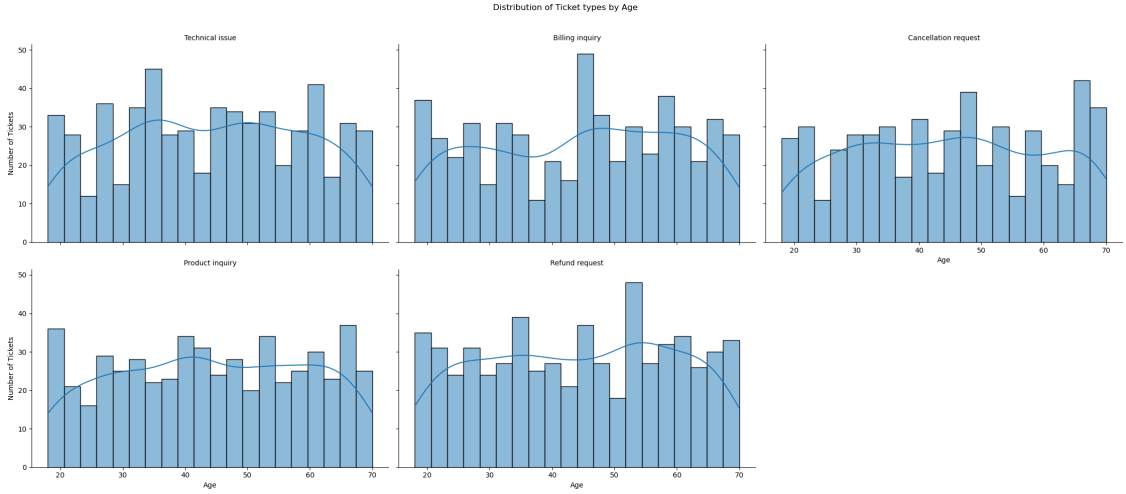
```
plt.grid(axis='y')
plt.show()
```

```
C:\Users\aa\AppData\Local\Temp\ipykernel_9764\1665756363.py:8: FutureWarning:
The default of observed=False is deprecated and will be changed to True in a
future version of pandas. Pass observed=False to retain current behavior or
observed=True to adopt the future default and silence this warning.
  tickets_by_age_group = data.groupby('Age Group').size()
```



[24]:
```
# Replace inf values with NAN
data.replace([np.inf,-np.inf],np.nan,inplace=True)
# Create a facet grid for each ticket type
g = sns.FacetGrid(data,col='Ticket Type',col_wrap=3,height=5,aspect=1.5)
g.map(sns.histplot,'Customer Age',bins=20,kde=True)
# Set title and labels
g.set_titles('{col_name}')
g.set_axis_labels('Age','Number of Tickets')
# Adjust layout
```

```
plt.subplots_adjust(top=0.9)
g.fig.suptitle('Distribution of Ticket types by Age')
plt.show()
```



[26]:
```
data.head(3)
```

[26]:
```
   Ticket ID        Customer Name             Customer Email  Customer Age  \
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  \
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                                 Ticket Description  \
2  Network problem  I'm facing a problem with my {product_purchase…
3   Account access  I'm having an issue with the {product_purchase…
4        Data loss  I'm having an issue with the {product_purchase…

  Ticket Status                                    Resolution  \
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  \
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 Age Group
2                           3.0     41-50
3                           3.0     21-30
4                           1.0     61-70
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: