Project Report: Customer Satisfaction Prediction

1. Project Overview

This project focuses on predicting customer satisfaction ratings based on various service-related parameters from a customer support dataset. The ultimate goal is to identify key factors influencing customer sentiment and provide actionable insights to enhance customer experience.

2. Dataset Description

The dataset includes 8,469 rows and 17 columns. Key features include:

- Demographics: Customer Name, Age, Gender
- Product Info: Product Purchased, Date of Purchase
- Support Ticket Info: Ticket Type, Subject, Description, Status, Priority, Channel
- Target Variable: Customer Satisfaction Rating (1-5 scale)

3. Data Cleaning & Preprocessing

- Major missing data handled using forward/backward fill or median imputation.
- Columns like 'Resolution', 'First Response Time', and 'Customer Satisfaction Rating' were processed accordingly.
- Label encoding applied to categorical variables like 'Resolution', 'Ticket Priority', and 'Ticket Channel'.

4. Exploratory Data Analysis (EDA)

- Visualized rating distribution, age, gender, ticket status and subject.
- Top issues and channel usage insights.
- Weak correlations observed between numerical features.

5. Feature Engineering

Selected features: 'Resolution', 'Ticket Priority', 'Ticket Channel'.

Target: 'Customer Satisfaction Rating'

StandardScaler used for normalization.

6. Model Building

Project Report: Customer Satisfaction Prediction

Algorithm: Random Forest Classifier (class_weight='balanced')

Split: 70% train, 30% test

7. Model Evaluation

- Accuracy: 56%
- Precision and recall high for class 3.0; poor for others.
- Confusion matrix confirms class imbalance.
- Feature importance: Resolution > Ticket Channel > Ticket Priority

8. Key Takeaways

- Bias towards rating 3.0 due to class imbalance.
- Text-based features could enhance prediction.
- Data balancing and advanced models are future steps.

9. Recommendations

- Apply SMOTE or similar techniques.
- Use NLP on 'Ticket Description'.
- Tune model hyperparameters.

10. Conclusion

A foundational model predicting customer satisfaction using structured data. Performance improvements needed for practical deployment.

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder,StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

In [5]:
```
df=pd.read_csv('customer.csv')
df.head()
```

Out[5]:

	Ticket ID	Customer Name	Customer Email	Customer Age	Customer Gender	Product Purchased	Date of Purchase	Ticket Type	Ticket Subject	Ticket Desc
0	1	Marisa Obrien	carrollallison@example.com	32	Other	GoPro Hero	2021-03-22	Technical issue	Product setup	I'm having a {product_pur
1	2	Jessica Rios	clarkeashley@example.com	42	Female	LG Smart TV	2021-05-22	Technical issue	Peripheral compatibility	I'm having a {product_pur
2	3	Christopher Robbins	gonzalestracy@example.com	48	Other	Dell XPS	2020-07-14	Technical issue	Network problem	I'm facing a p {product_pur
3	4	Christina Dillon	bradleyolson@example.org	27	Female	Microsoft Office	2020-11-13	Billing inquiry	Account access	I'm having a {product_pur
4	5	Alexander Carroll	bradleymark@example.com	67	Female	Autodesk AutoCAD	2020-02-04	Billing inquiry	Data loss	I'm having a {product_pur

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
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
```

In [11]:
```
df.shape
```

Out[11]:
```
(8469, 17)
```

In [19]:
```
df['Resolution']=df['Resolution'].bfill().ffill()
df.head()
```

Out[19]:

	Ticket ID	Customer Name	Customer Email	Customer Age	Customer Gender	Product Purchased	Date of Purchase	Ticket Type	Ticket Subject	Ticket Desc
0	1	Marisa Obrien	carrollallison@example.com	32	Other	GoPro Hero	2021-03-22	Technical issue	Product setup	I'm having a {product_pur
1	2	Jessica Rios	clarkeashley@example.com	42	Female	LG Smart TV	2021-05-22	Technical issue	Peripheral compatibility	I'm having a {product_pur
2	3	Christopher Robbins	gonzalestracy@example.com	48	Other	Dell XPS	2020-07-14	Technical issue	Network problem	I'm facing a p {product_pur
3	4	Christina Dillon	bradleyolson@example.org	27	Female	Microsoft Office	2020-11-13	Billing inquiry	Account access	I'm having a {product_pur
4	5	Alexander Carroll	bradleymark@example.com	67	Female	Autodesk AutoCAD	2020-02-04	Billing inquiry	Data loss	I'm having a {product_pur

In [21]:
```
df['First Response Time'] = pd.to_datetime(df['First Response Time'], errors='coerce')
df['Time to Resolution'] = pd.to_datetime(df['Time to Resolution'], errors='coerce')
```

```
# Fill missing 'First Response Time' and 'Time to Resolution' with median timestamp
df['First Response Time'].fillna(df['First Response Time'].median(), inplace=True)
df['Time to Resolution'].fillna(df['Time to Resolution'].median(), inplace=True)

# Fill missing Customer Satisfaction Rating with median rating
df['Customer Satisfaction Rating'].fillna(df['Customer Satisfaction Rating'].median(), inplace=True)
```

```
/var/folders/pm/cnlmdnjj5g1ct4r7rrx83vnr0000gn/T/ipykernel_1031/518815202.py:5: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which
we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or
df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['First Response Time'].fillna(df['First Response Time'].median(), inplace=True)
/var/folders/pm/cnlmdnjj5g1ct4r7rrx83vnr0000gn/T/ipykernel_1031/518815202.py:6: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which
we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or
df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['Time to Resolution'].fillna(df['Time to Resolution'].median(), inplace=True)
/var/folders/pm/cnlmdnjj5g1ct4r7rrx83vnr0000gn/T/ipykernel_1031/518815202.py:9: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which
we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or
df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['Customer Satisfaction Rating'].fillna(df['Customer Satisfaction Rating'].median(), inplace=True)
```

In [25]:
```
df.isnull().sum()
```

Out[25]:
```
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                      0
Ticket Priority                 0
Ticket Channel                  0
First Response Time             0
Time to Resolution              0
Customer Satisfaction Rating    0
dtype: int64
```

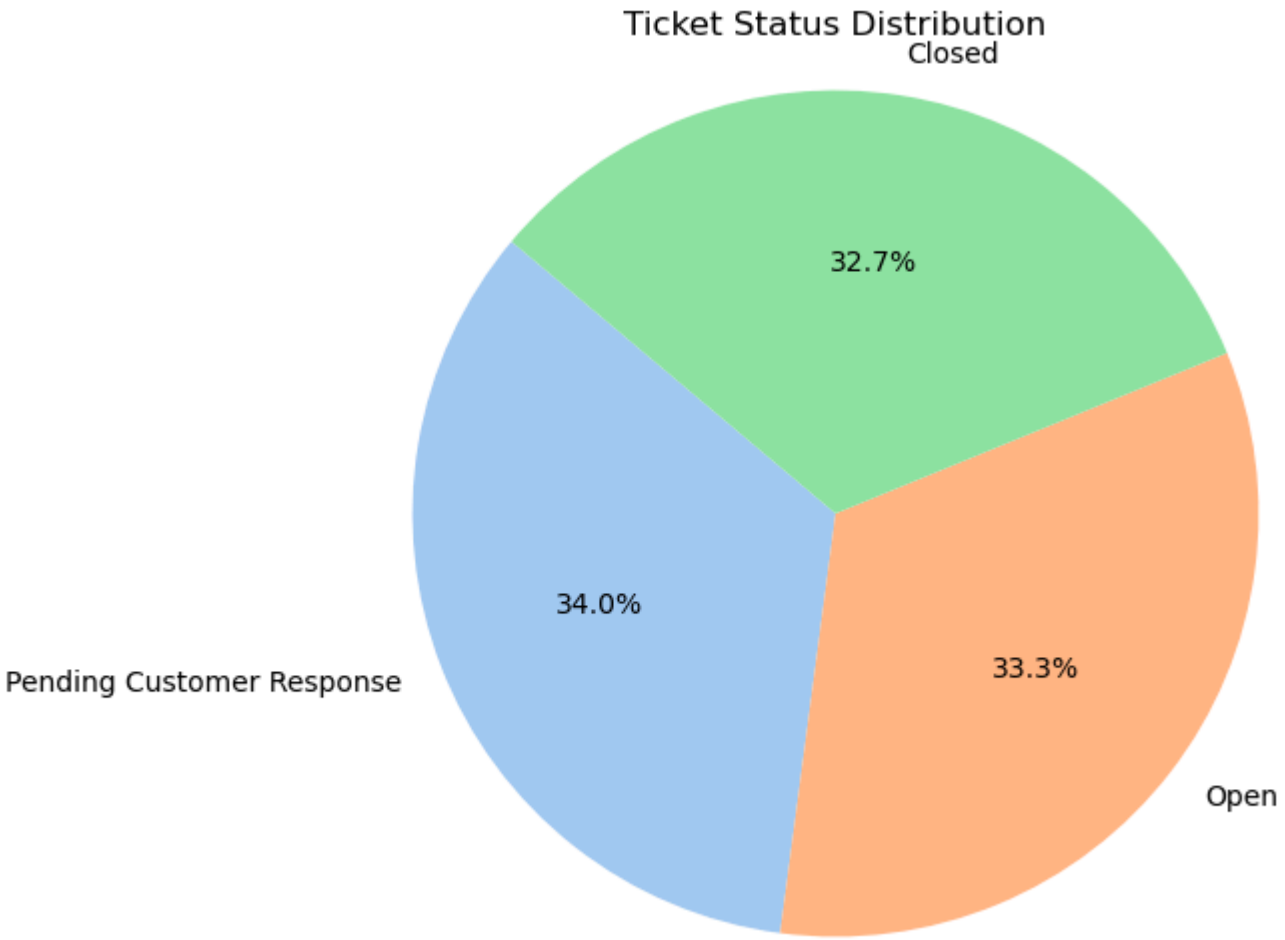
Customer Satisfaction Rating Distribution

In [30]:
```
plt.figure(figsize=(8, 6))
sns.histplot(df['Customer Satisfaction Rating'], bins=5, kde=True, color='skyblue')
plt.title('Customer Satisfaction Rating Distribution')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```

Ticket Status Distribution

In [35]:
```
ticket_status = df['Ticket Status'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(ticket_status, labels=ticket_status.index, autopct='%1.1f%%', startangle=140, colors=sns.color_palette('pas
plt.title('Ticket Status Distribution')
```

```
plt.axis('equal')
plt.show()
```

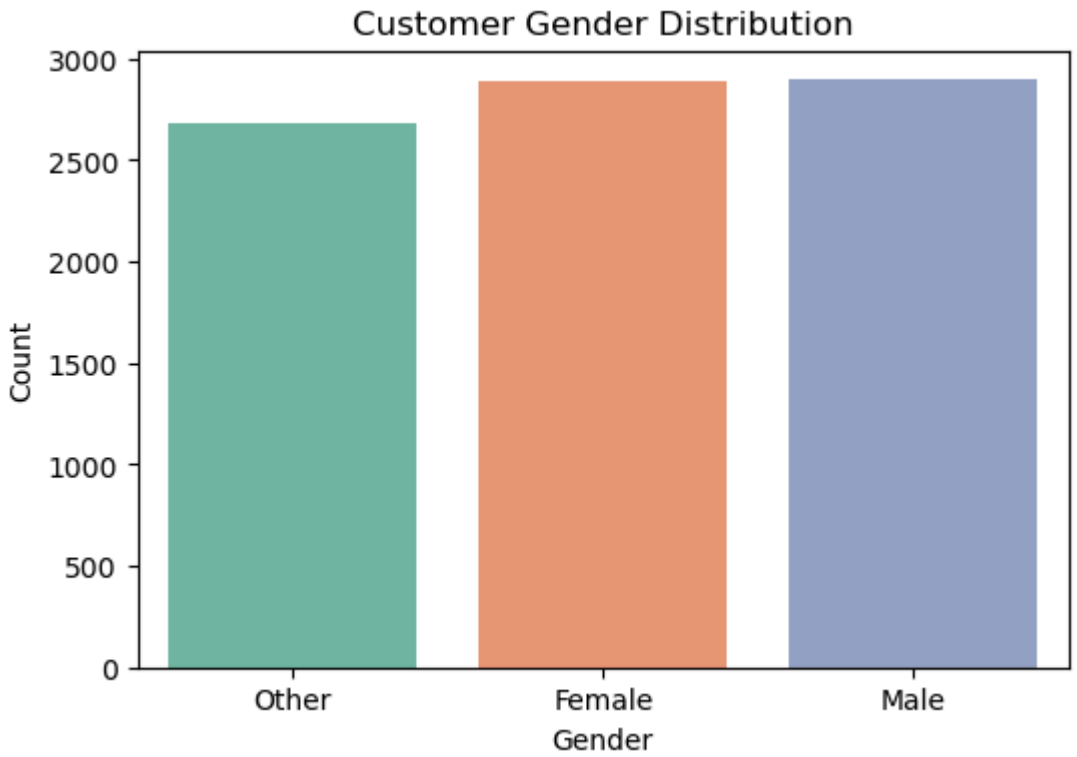


Customer Gender Distribution

In [40]:
```
plt.figure(figsize=(6, 4))
sns.countplot(x='Customer Gender', data=df, palette='Set2')
plt.title('Customer Gender Distribution')
plt.xlabel('Gender')
```

```
plt.ylabel('Count')
plt.show()
```

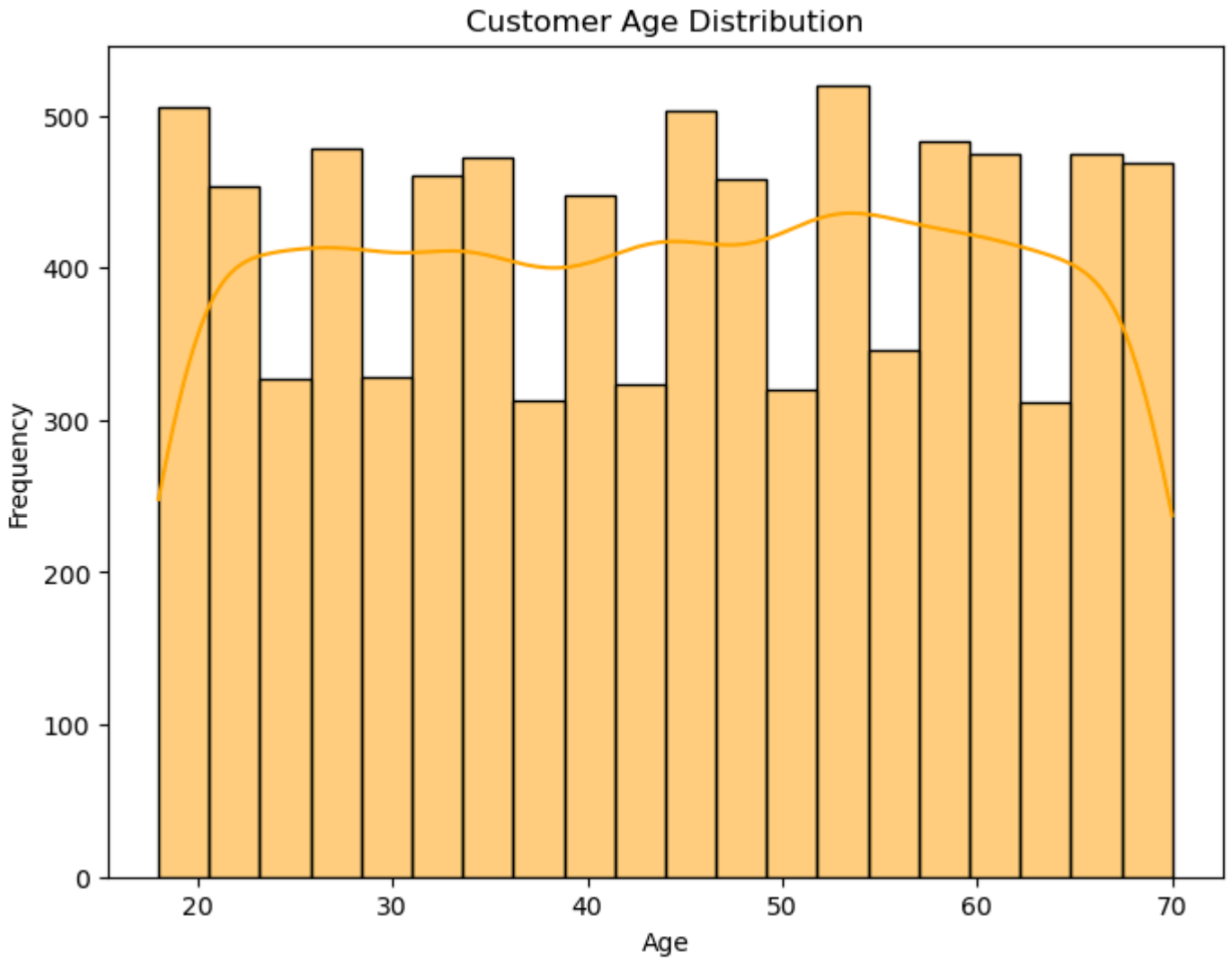
```
/var/folders/pm/cnlmdnjj5g1ct4r7rrx83vnr0000gn/T/ipykernel_1031/1215856890.py:2: FutureWarning:

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.countplot(x='Customer Gender', data=df, palette='Set2')
```



Customer Age Distribution

In [47]:
```
plt.figure(figsize=(8, 6))
sns.histplot(df['Customer Age'], bins=20, kde=True, color='orange')
plt.title('Customer Age Distribution')
plt.xlabel('Age')
```

```
plt.ylabel('Frequency')
plt.show()
```

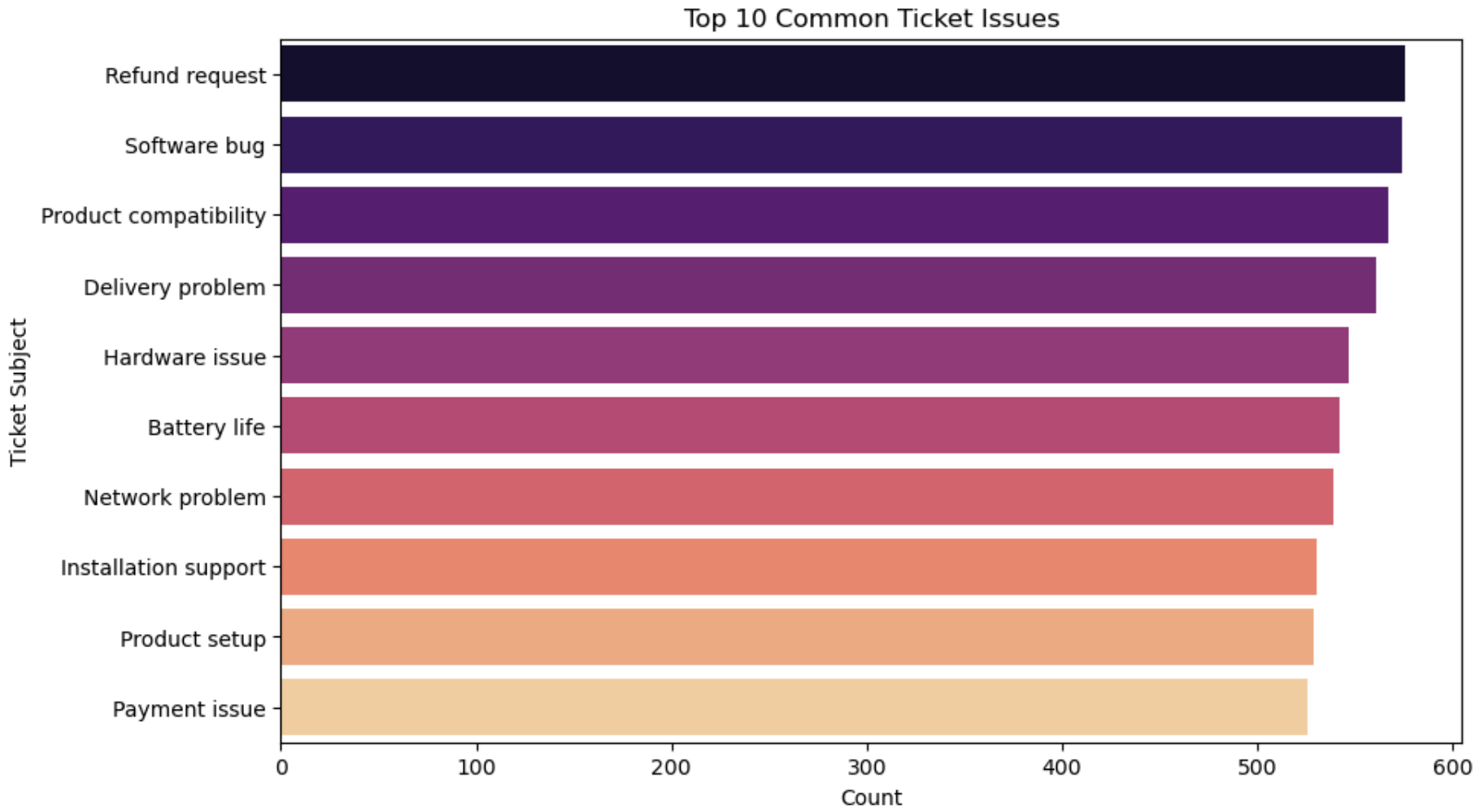


Top 10 Common Issues (Ticket Subjects)

In [52]:
```
top_issues = df['Ticket Subject'].value_counts().head(10)
plt.figure(figsize=(10, 6))
```

```
sns.barplot(y=top_issues.index, x=top_issues.values, palette='magma')
plt.title('Top 10 Common Ticket Issues')
plt.xlabel('Count')
plt.ylabel('Ticket Subject')
plt.show()
```

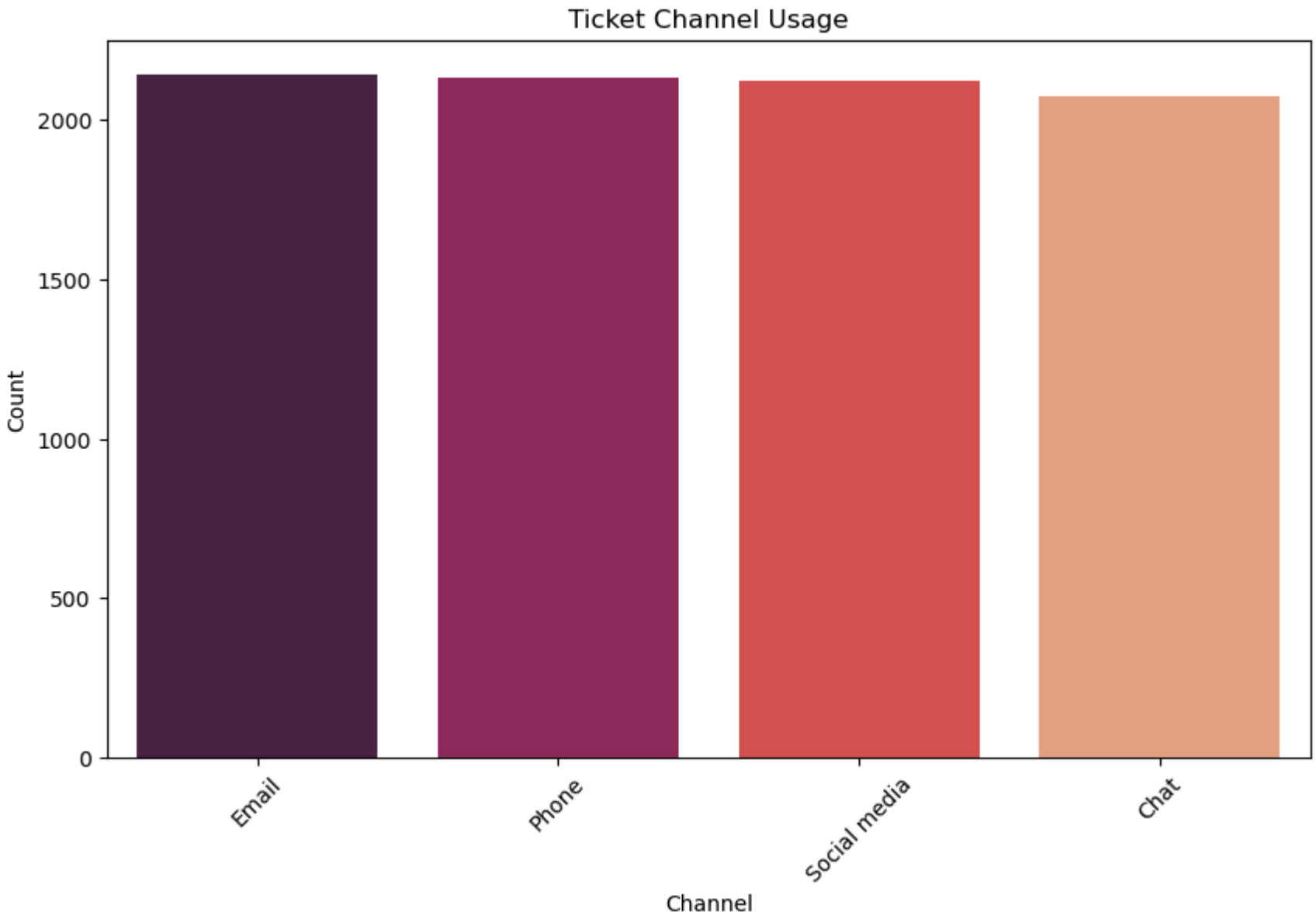
```
/var/folders/pm/cnlmdnjj5g1ct4r7rrx83vnr0000gn/T/ipykernel_1031/4145585260.py:3: FutureWarning:

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(y=top_issues.index, x=top_issues.values, palette='magma')
```



Ticket Channels Used

In [57]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='Ticket Channel', data=df, order=df['Ticket Channel'].value_counts().index, palette='rocket')
plt.title('Ticket Channel Usage')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

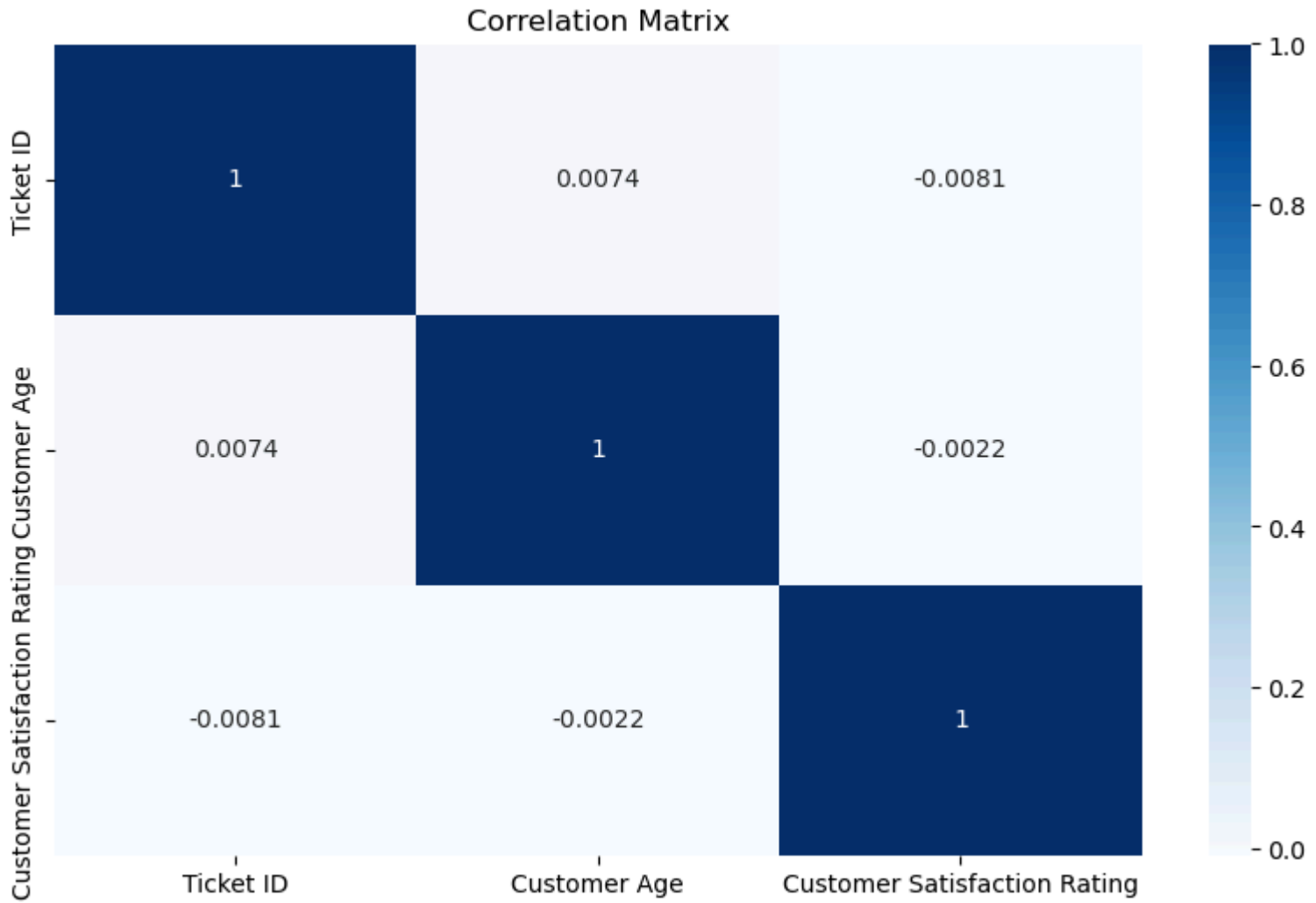
```
/var/folders/pm/cnlmdnjj5g1ct4r7rrx83vnr0000gn/T/ipykernel_1031/2820793133.py:2: FutureWarning:

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.countplot(x='Ticket Channel', data=df, order=df['Ticket Channel'].value_counts().index, palette='rocket')
```



Correlation Heatmap (Numerical Features)

In [62]:
```
plt.figure(figsize=(10, 6))
sns.heatmap(df.select_dtypes(include=['int64', 'float64']).corr(), annot=True, cmap='Blues')
```

```
plt.title('Correlation Matrix')
plt.show()
```



In [ ]:
```
#df['Resolution Duration (hrs)'] = (df['Time to Resolution'] - df['First Response Time']).dt.total_seconds() / 3600
```

In [92]:
```
label_encoders = {}
for col in ['Resolution', 'Ticket Priority', 'Ticket Channel']:
    le = LabelEncoder()
```

```
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
```

In [94]:
```
df.head()
```

Out[94]:

	Ticket ID	Customer Name	Customer Email	Customer Age	Customer Gender	Product Purchased	Date of Purchase	Ticket Type	Ticket Subject	Ticket Desc
0	1	Marisa Obrien	carrollallison@example.com	32	Other	GoPro Hero	2021-03-22	Technical issue	Product setup	I'm having a {product_pur
1	2	Jessica Rios	clarkeashley@example.com	42	Female	LG Smart TV	2021-05-22	Technical issue	Peripheral compatibility	I'm having a {product_pur
2	3	Christopher Robbins	gonzalestracy@example.com	48	Other	Dell XPS	2020-07-14	Technical issue	Network problem	I'm facing a p {product_pur
3	4	Christina Dillon	bradleyolson@example.org	27	Female	Microsoft Office	2020-11-13	Billing inquiry	Account access	I'm having a {product_pur
4	5	Alexander Carroll	bradleymark@example.com	67	Female	Autodesk AutoCAD	2020-02-04	Billing inquiry	Data loss	I'm having a {product_pur

In [102…
```
X = df[['Resolution', 'Ticket Priority', 'Ticket Channel']]
y = df['Customer Satisfaction Rating']
```

In [104…
```
print(X.dtypes)
```

```
Resolution         int64
Ticket Priority    int64
Ticket Channel     int64
dtype: object
```

In [106…
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

Out[106… ((5928, 3), (2541, 3), (5928,), (2541,))

In [108…
```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [153…
```
model = RandomForestClassifier(class_weight='balanced',random_state=42)
model.fit(X_train_scaled, y_train)


y_pred = model.predict(X_test_scaled)
```

In [155…
```
from sklearn.metrics import confusion_matrix,classification_report
```

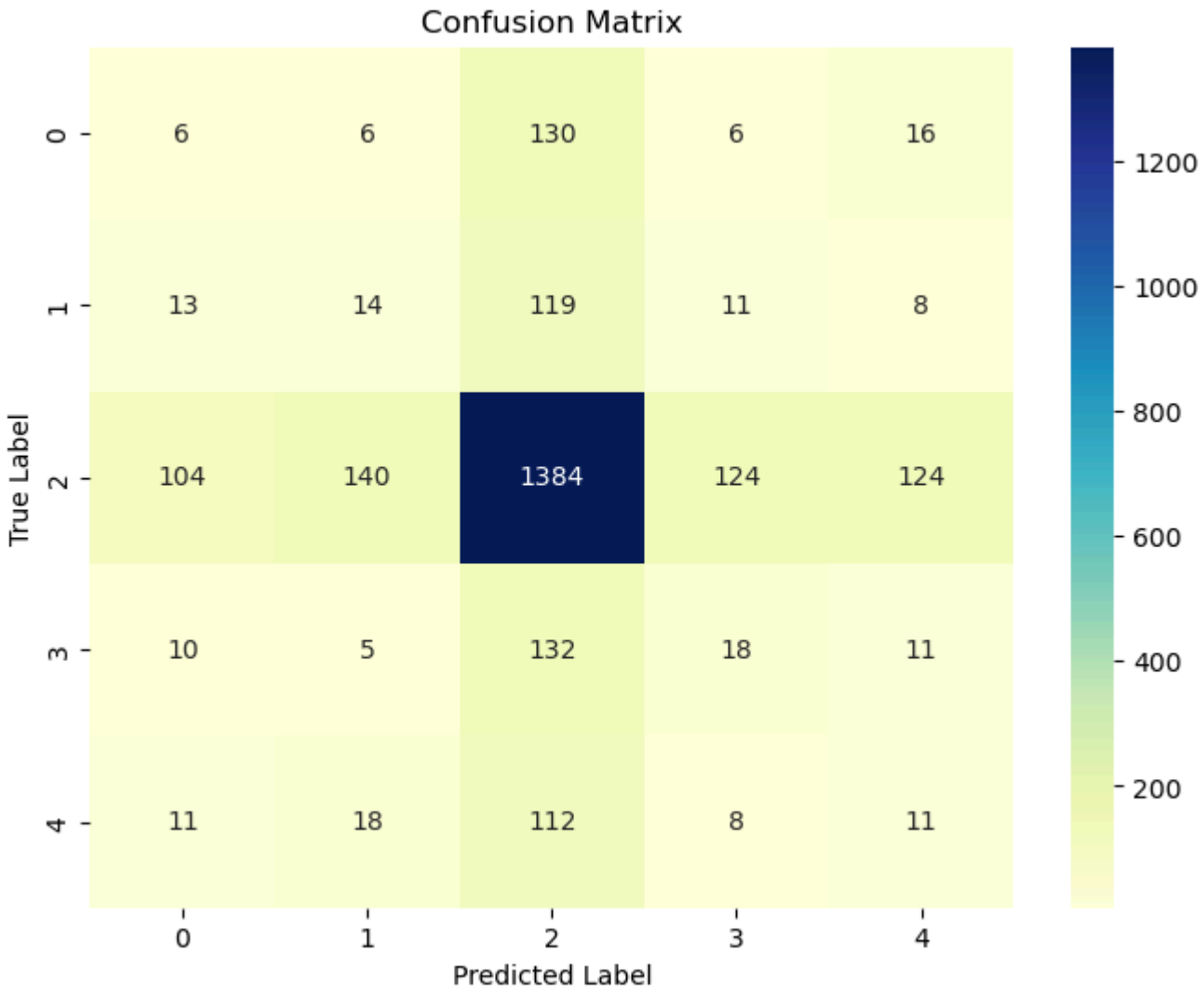
In [157…
```
cr=classification_report(y_test,y_pred)
```

In [159…
```
print(cr)
```

```
              precision    recall  f1-score   support

         1.0       0.04      0.04      0.04       164
         2.0       0.08      0.08      0.08       165
         3.0       0.74      0.74      0.74      1876
         4.0       0.11      0.10      0.10       176
         5.0       0.06      0.07      0.07       160

    accuracy                           0.56      2541
   macro avg       0.21      0.21      0.21      2541
weighted avg       0.56      0.56      0.56      2541
```

In [161…
```
cm = confusion_matrix(y_test, y_pred)

# Plot with a different color palette
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='YlGnBu')  # Try cmap='coolwarm', 'magma', 'viridis', etc.
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
```

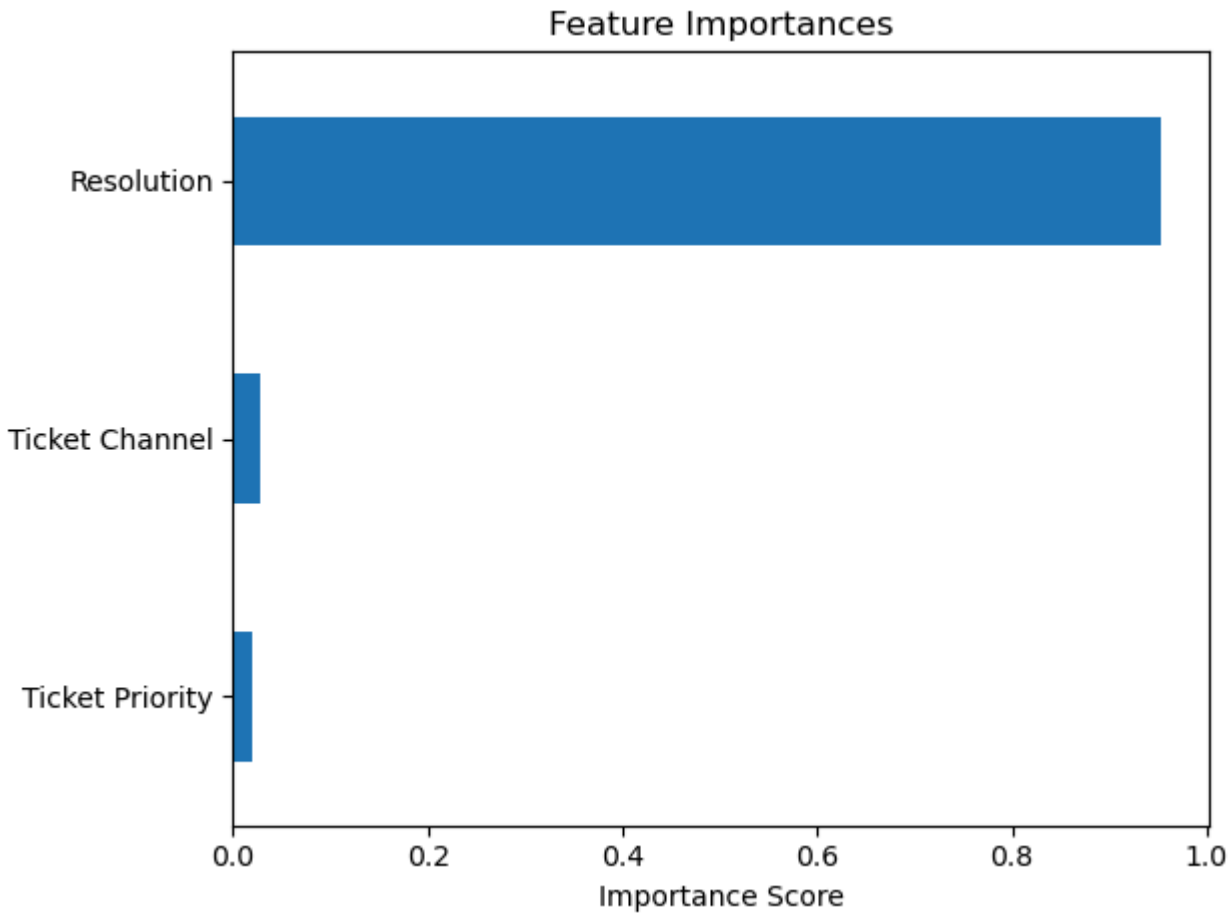
```
plt.ylabel('True Label')
plt.show()
```



In [163…

```
importances = pd.Series(model.feature_importances_, index=X.columns)
importances.sort_values().plot(kind='barh', title='Feature Importances')
plt.xlabel('Importance Score')
```

```
plt.tight_layout()
plt.show()
```



In [ ]: