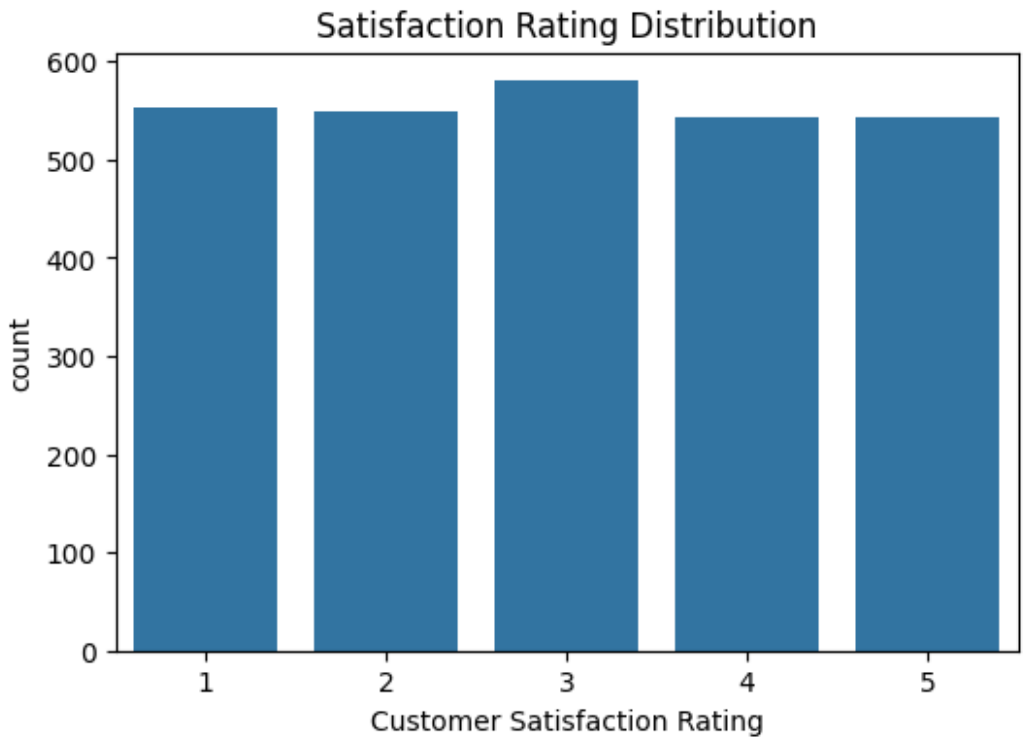
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

cols = [
    'Customer Age','Customer Gender','Product Purchased','Ticket
Type',
    'Ticket Priority','Ticket Channel','Customer Satisfaction Rating'
]

chunk_iter = pd.read_csv(
    '/content/drive/MyDrive/customer_support_tickets.csv',
    usecols=cols,
    chunksize=200_000
)

df = pd.concat(chunk_iter, ignore_index=True)

df = df.dropna(subset=['Customer Satisfaction Rating'])
df['Customer Satisfaction Rating'] = df['Customer Satisfaction
Rating'].astype(int)

plt.figure(figsize=(6,4))
sns.countplot(x='Customer Satisfaction Rating', data=df)
plt.title('Satisfaction Rating Distribution')
plt.show()
```



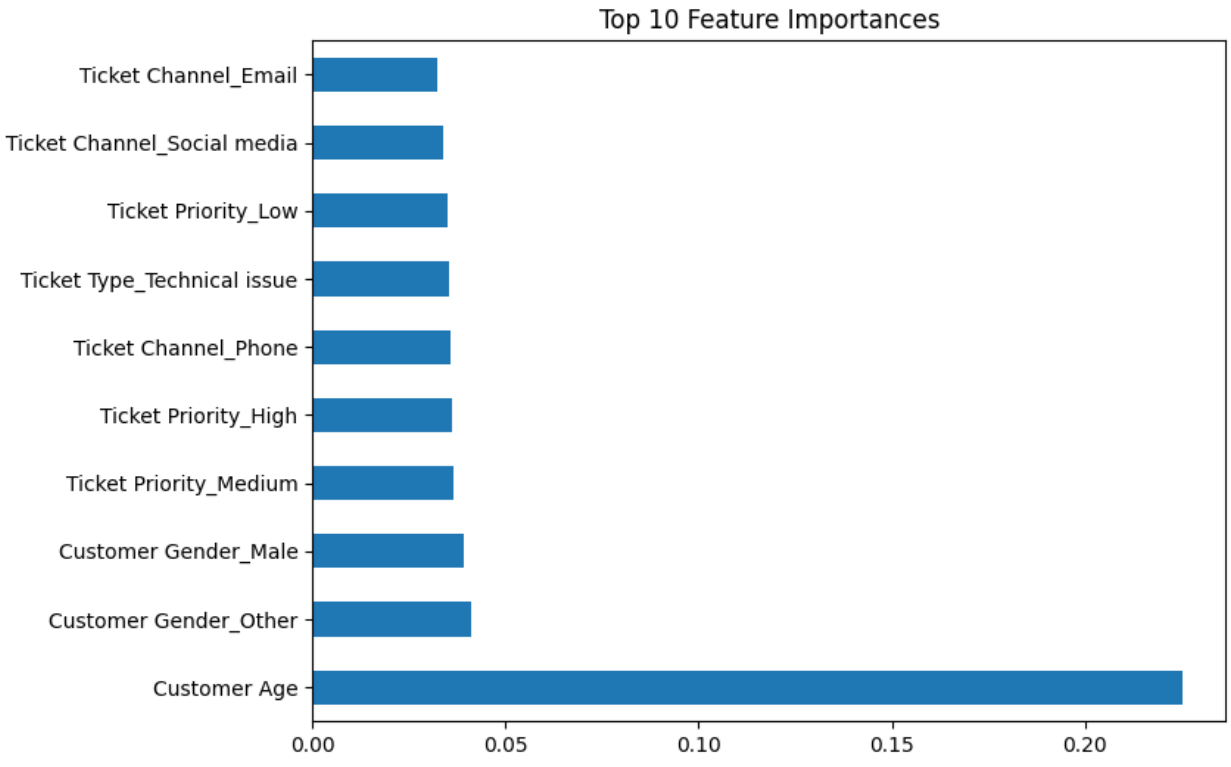

```
plt.figure(figsize=(6,4))
sns.heatmap(
    df[[
      'Customer Age','Customer Satisfaction Rating'
    ]].corr(), annot=True, cmap='coolwarm'
)
plt.title('Correlation Matrix')
plt.show()
```


```
X = df.drop('Customer Satisfaction Rating', axis=1)
y = df['Customer Satisfaction Rating']

numeric_feats = ['Customer Age']
categorical_feats = [
    'Customer Gender','Product Purchased','Ticket Type',
    'Ticket Priority','Ticket Channel'
]

preprocessor = ColumnTransformer([
    ('num', 'passthrough', numeric_feats),
    (
       'cat',
       OneHotEncoder(drop='first'),
       categorical_feats
    )
])

clf = Pipeline([
    ('preproc', preprocessor),
    ('rf', RandomForestClassifier(n_jobs=-1, random_state=42))
])

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

```
print('Accuracy:', accuracy_score(y_test, y_pred))
print('\nConfusion Matrix:\n', confusion_matrix(y_test, y_pred))
print('\nClassification Report:\n', classification_report(y_test,
y_pred))
```

```
Accuracy: 0.2009626955475331

Confusion Matrix:
 [[33 35 41 27 32]
 [27 33 49 33 32]
 [30 27 39 30 49]
 [29 30 37 35 31]
 [28 29 44 24 27]]

Classification Report:
              precision    recall  f1-score   support

           1       0.22      0.20      0.21       168
           2       0.21      0.19      0.20       174
           3       0.19      0.22      0.20       175
           4       0.23      0.22      0.23       162
           5       0.16      0.18      0.17       152

    accuracy                           0.20       831
   macro avg       0.20      0.20      0.20       831
weighted avg       0.20      0.20      0.20       831
```

```
feat_names = (
    numeric_feats +
    list(
      clf.named_steps['preproc']
         .named_transformers_['cat']
         .get_feature_names_out(categorical_feats)
    )
)
importances = clf.named_steps['rf'].feature_importances_
imp = pd.Series(importances, index=feat_names).nlargest(10)

plt.figure(figsize=(8,6))
imp.plot(kind='barh')
plt.title('Top 10 Feature Importances')
plt.show()
```


Top 10 Feature Importances
Ticket Channel_Email
Ticket Channel_Social media
Ticket Priority_Low
Ticket Type_Technical issue
Ticket Channel_Phone
Ticket Priority_High
Ticket Priority_Medium
Customer Gender_Male
Customer Gender_Other
Customer Age
0.00
0.05
0.10
0.15
0.20