Project Name - Customer Satisfaction Prediction (ML _ FA _ DA projects)(Part 1)

Project Type - Data Analysis

Industry - Unified Mentor

Contribution - Individual

Member Name - Hare Krishana Mishra

Task - 1

Project Summary -

Project Description:

The Customer Support Analysis project involves exploring and analyzing a customer support ticket dataset to identify common issues, track support trends, and segment customers based on ticket characteristics. The dataset includes details such as customer demographics, product purchased, type of support request, resolution status, priority levels, and satisfaction ratings. Through data visualization and segmentation, the project helps uncover key patterns in customer service operations, providing valuable insights for process improvement and decision-making.

Objective:

The objective of this project is to analyze customer support tickets to improve service quality and efficiency. Specifically, the project aims to:

- Identify the most frequent support issues.
- Track ticket trends over time.
- Understand customer demographics and behavior.
- Segment customers based on ticket types and satisfaction ratings.
- Provide insights for optimizing ticket resolution processes.

Key Project Details:

Domain: Data Analytics / Exploratory Data Analysis (EDA)

Difficulty Level: Advanced

Tools & Technologies: Python, Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Jupyter Notebook, VS Code, SQL, Excel

Dataset: Customer Support Ticket Dataset containing fields like Ticket ID, Customer Age, Gender, Product Purchased, Ticket Type, Status, Resolution, Priority, Channel, First Response Time, Time to Resolution, and Customer Satisfaction Rating.

Key Steps:

- Data Preprocessing and Cleaning
- Exploratory Data Analysis (EDA)
- Visualizing Ticket Trends and Common Issues
- Segmentation by Ticket Type and Satisfaction Rating
- Analysis of Demographics and Support Channels

Use Cases: Identifying key service pain points, monitoring ticket trends, improving resource allocation, and enhancing the customer experience.

Let's Begin:-

Data Preprocessing

In [ ]:

```
# Importing necessary libraries
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 StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusior
```

In [ ]:

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

Out[ ]:

	Ticket ID	Customer Name	Customer Email	Customer Age	Customer Gender	P Pur
0	1	Marisa Obrien	carrollallison@example.com	32	Other	Gol
1	2	Jessica Rios	clarkeashley@example.com	42	Female	L
2	3	Christopher Robbins	gonzalestracy@example.com	48	Other	I
3	4	Christina Dillon	bradleyolson@example.org	27	Female	M
4	5	Alexander Carroll	bradleymark@example.com	67	Female	A A
...	...	...	...	...	...	
8464	8465	David Todd	adam28@example.net	22	Female	L
8465	8466	Lori Davis	russell68@example.com	27	Female	So
8466	8467	Michelle Kelley	ashley83@example.org	57	Female	
8467	8468	Steven Rodriguez	fpowell@example.org	54	Male	Pla
8468	8469	Steven Davis MD	lori20@example.net	53	Other	Phi

8469 rows × 17 columns

In [ ]:

```
# Display basic info about the dataset
print(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
None
```

In [ ]:

```
# Data Preprocessing
# Handling missing values
data = data.dropna()
```

In [ ]:

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

In [ ]:

```
# Define features and target variable
X = data.drop(['Customer Email', 'Customer Satisfaction Rating'], axis=1)
y = data['Customer Satisfaction Rating']
```

In [ ]:

```
# Splitting the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,ranc
```

In [ ]:

```
from sklearn.preprocessing import LabelEncoder, StandardScaler

# Make a copy to avoid altering the original data
df = data.copy()

# Drop columns that are not useful for prediction (IDs, names, emails, free-
df = df.drop(['Ticket ID', 'Customer Name', 'Customer Email', 'Ticket Subjec

# Encode categorical columns
label_encoders = {}
for col in df.select_dtypes(include=['object']).columns:
    label_encoders[col] = LabelEncoder()
```

```
    df[col] = label_encoders[col].fit_transform(df[col].astype(str))

# Define features and target
X = df.drop('Customer Satisfaction Rating', axis=1)
y = df['Customer Satisfaction Rating']

# Train-test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, rar

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [ ]:

```
# Remove rows where target is NaN
df = df.dropna(subset=['Customer Satisfaction Rating'])

# Now split again
X = df.drop('Customer Satisfaction Rating', axis=1)
y = df['Customer Satisfaction Rating']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, rar

# Scale features
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train the model
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state=42)
rfc.fit(X_train, y_train)
```

Out[ ]:

```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [ ]:

```
# Predict on the test set
y_pred = rfc.predict(X_test)
```

In [ ]:

```
# Model Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

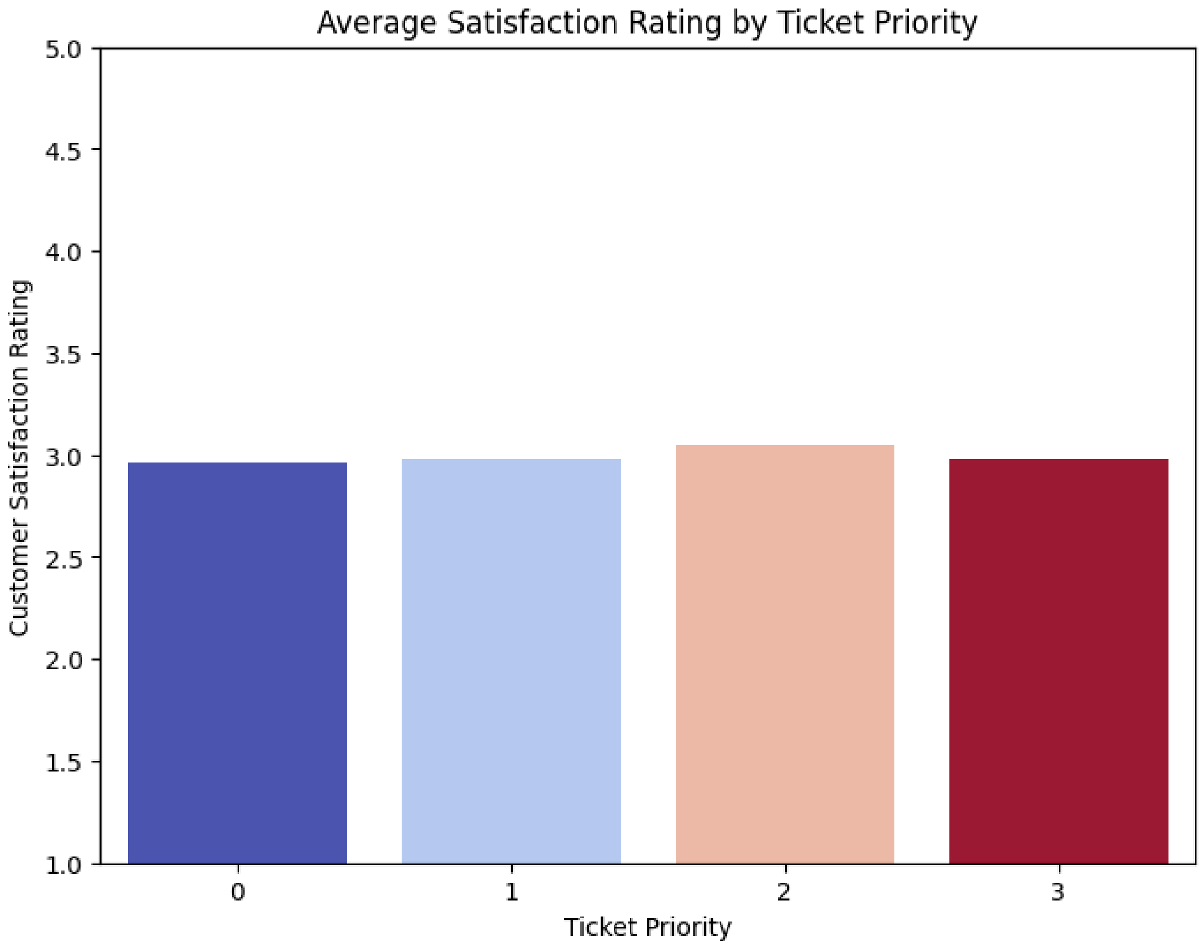
```
Accuracy: 0.2069795427196149
Classification Report:
               precision    recall  f1-score   support

         1.0       0.21      0.19      0.20       168
         2.0       0.20      0.19      0.19       174
         3.0       0.23      0.25      0.24       175
         4.0       0.18      0.17      0.17       162
         5.0       0.21      0.24      0.22       152

    accuracy                           0.21       831
   macro avg       0.21      0.21      0.21       831
weighted avg       0.21      0.21      0.21       831
```

Exploratory Data Analysis (EDA)

Average Customer Satisfaction Score Across Ticket Priority Levels

In [ ]:

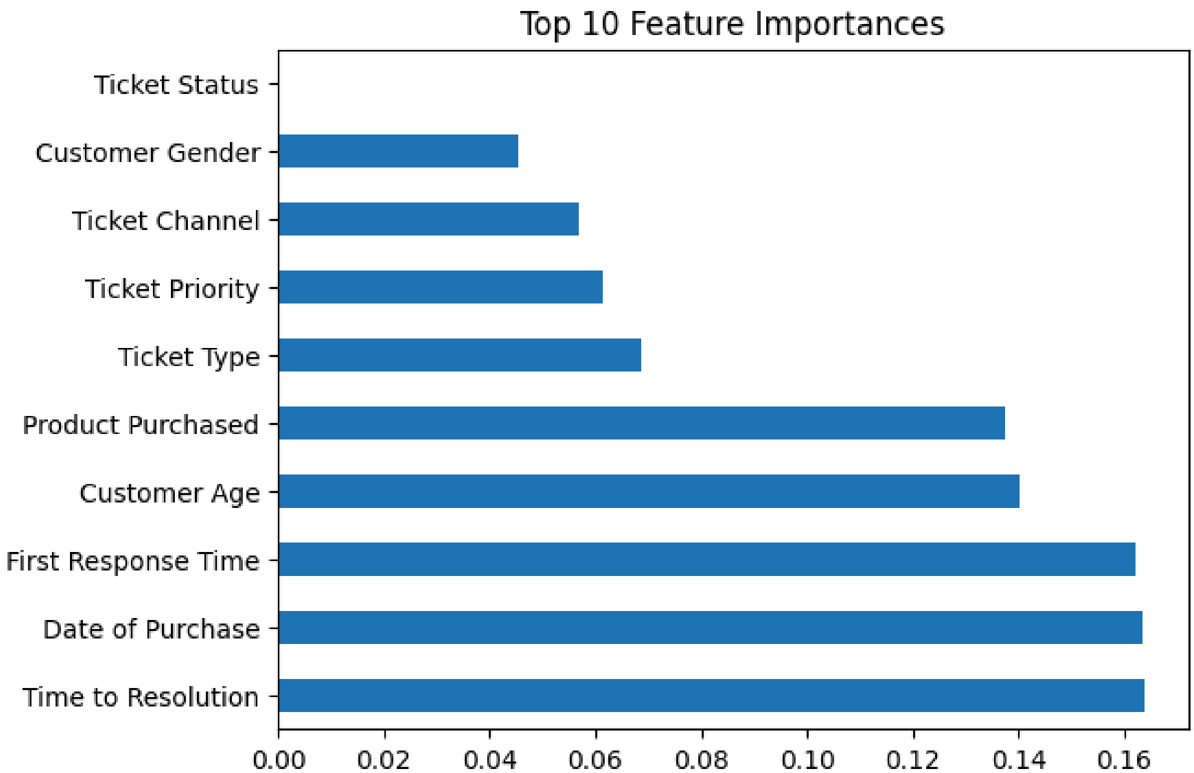
```
plt.figure(figsize=(8,6))
avg_satisfaction_priority = df.groupby('Ticket Priority')['Customer Satisfac

sns.barplot(
    x='Ticket Priority',
    y='Customer Satisfaction Rating',
    hue='Ticket Priority',  # same as x
    data=avg_satisfaction_priority,
    palette='coolwarm',
    legend=False  # hides redundant legend
)

plt.title('Average Satisfaction Rating by Ticket Priority')
plt.ylim(1,5)
plt.show()
```



Top 10 Most Influential Features Driving Customer Satisfaction

In [ ]:
```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
# Visualization of Results
# 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()
```

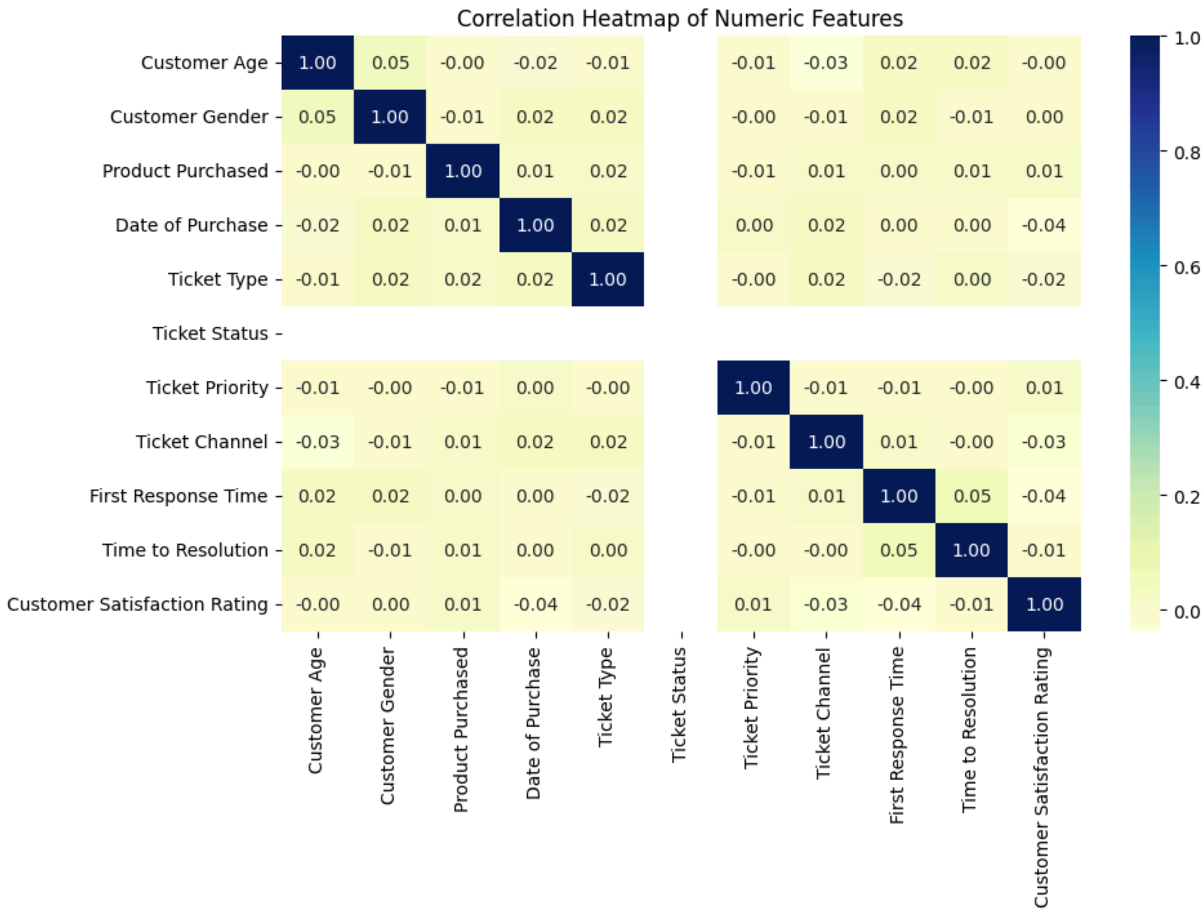
```
Confusion Matrix:
 [[32 34 40 33 29]
 [36 33 47 28 30]
 [36 31 44 30 34]
 [29 35 31 27 40]
 [21 33 32 30 36]]
```



Correlation Heatmap: Relationships Between Numeric Features

In [ ]:

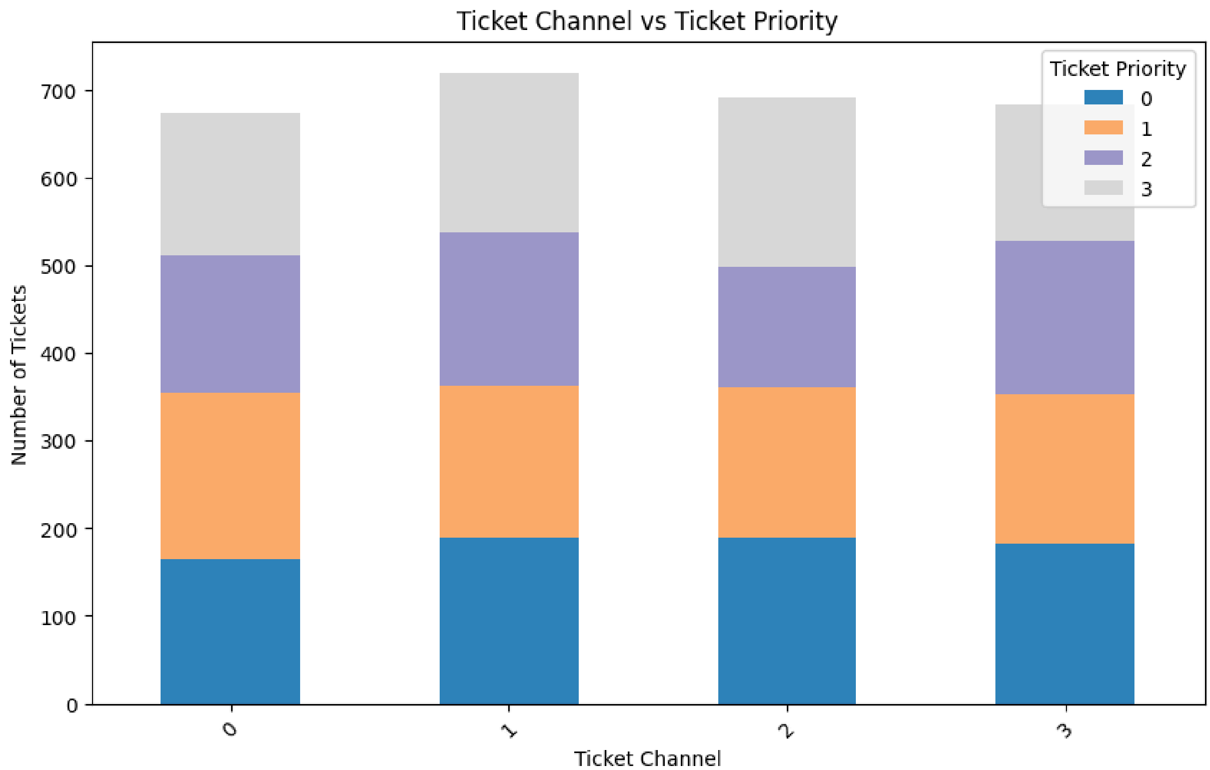
```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap='YlGnBu', fmt=".2f")
plt.title('Correlation Heatmap of Numeric Features')
plt.show()
```



Ticket Volume by Channel and Priority Level

In [ ]:

```
channel_priority = df.groupby(['Ticket Channel','Ticket Priority']).size().

channel_priority.plot(kind='bar', stacked=True, figsize=(10,6), colormap='ta
plt.title('Ticket Channel vs Ticket Priority')
plt.ylabel('Number of Tickets')
plt.xticks(rotation=45)
plt.show()
```


Ticket Channel vs Ticket Priority
Ticket Priority
0
1
2
3
0
100
200
300
400
500
600
700
Number of Tickets
0
1
2
3
Ticket Channel