Airline Passenger Satisfaction analysis

1. Project Overview (what we want to do)

Objective: analyze passenger satisfaction and answer questions such as:

- Count how many passengers selected each satisfaction level (basic task).
- How does satisfaction vary by travel class?
- Which in-flight service features correlate most with satisfaction?
- Does age group or travel type (business vs personal) influence satisfaction?
- How do delays relate to satisfaction?
- For dissatisfied passengers, which features are rated worst?
- Build simple predictive models (decision tree / logistic regression) to identify important features.

Environment & Libraries (and why we use them)

- *pandas* — data loading, cleaning, grouping, aggregation (read_csv, groupby, value_counts, pd.cut, get_dummies).
- *numpy* — numeric operations and preparing arrays.
- *matplotlib / seaborn* — plotting (countplot, barplot, heatmap, scatter).
- *scikit-learn* (DecisionTreeClassifier, LogisticRegression, train_test_split, StandardScaler) — simple models and feature importance.
- *scipy.stats* (chi2_contingency) — chi-square test for categorical associations. These are standard and commonly used in data analysis pipelines.

Project: Airline Passenger Satisfaction (Kaggle Dataset)

1. Import & Load Data

```
import zipfile, pandas as pd
```

```
with zipfile.ZipFile("train.csv.zip") as z:
   with z.open("train.csv") as f:
      df = pd.read_csv(f)

df.head()
```

---

2. Basic Cleaning

```
df['satisfaction_num'] = df['satisfaction'].map({'satisfied':1, 'neutral or dissatisfied':0})


# Age groups
bins=[0,25,40,60,100]
labels=["Youth(<25)","YoungAdult(25-40)","MiddleAge(40-60)","Senior(60+)"]
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels)
```

---

3. Satisfaction Counts

```
import seaborn as sns, matplotlib.pyplot as plt


counts = df['satisfaction'].value_counts().reset_index()
counts.columns = ['satisfaction','count']


sns.barplot(data=counts, x='satisfaction', y='count', hue='satisfaction', palette='viridis', legend=False)
plt.title("Satisfaction Level Counts")
plt.show()
```

---

4. Satisfaction by Class

```
class_satisfaction =
df.groupby(['Class','satisfaction']).size().reset_index(name='count')


sns.barplot(data=class_satisfaction, x='Class', y='count', hue='satisfaction',
palette='Set2')
plt.title("Satisfaction by Class")
plt.show()
```

---

5. Correlation with Service Features

```
service_cols = [
 'Inflight wifi service','Food and drink','Seat comfort','Inflight entertainment',
 'On-board service','Leg room service','Baggage handling','Checkin service',
 'Inflight service','Cleanliness','Online boarding'
]
service_cols = [c for c in service_cols if c in df.columns]


corr =
df[service_cols+['satisfaction_num']].corr()['satisfaction_num'].sort_values(as
cending=False)


sns.barplot(x=corr.index, y=corr.values, palette="coolwarm")
```

```
plt.xticks(rotation=45)
plt.title("Correlation with Satisfaction")
plt.show()
```

---

6. Decision Tree (Feature Importance)

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix

X = df[service_cols].fillna(0)
y = df['satisfaction_num']

X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.25,random_state=42)

clf = DecisionTreeClassifier(max_depth=4, random_state=42)
clf.fit(X_train,y_train)

print(classification_report(y_test, clf.predict(X_test)))

importances = pd.Series(clf.feature_importances_,
index=X.columns).sort_values(ascending=False)
sns.barplot(x=importances.index, y=importances.values, palette="viridis")
plt.xticks(rotation=45)
plt.title("Feature Importances")
plt.show()
```

---

7. Satisfaction by Age Group & Travel Type

```
sns.countplot(data=df, x='AgeGroup', hue='satisfaction', palette='Set2')
plt.title("Satisfaction by Age Group")
plt.show()

sns.countplot(data=df, x='Type of Travel', hue='satisfaction', palette='mako')
plt.title("Satisfaction by Travel Type")
plt.show()
```

---

8. Delay Impact

```
df['DepartureStatus'] = df['Departure Delay in Minutes'].apply(lambda x:'On-time' if x==0 else 'Delayed')

sns.countplot(data=df, x='DepartureStatus', hue='satisfaction', palette='viridis')
plt.title("Satisfaction by Departure Status")
plt.show()
```

---

9. Dissatisfied Passenger — Worst Features

```
dissatisfied = df[df['satisfaction']=="neutral or dissatisfied"]
mean_ratings = dissatisfied[service_cols].mean().sort_values()
```

```
plt.scatter(mean_ratings.values, mean_ratings.index, s=mean_ratings.values*300,
         c=mean_ratings.values, cmap="viridis", alpha=0.75, edgecolor="black")
for i,v in enumerate(mean_ratings.values):
   plt.text(v+0.05, i, f"{v:.2f}", va='center')
plt.title("Worst Rated Features (Dissatisfied)")
plt.xlabel("Average Rating")
plt.show()
```

What We Did (Summary for Notes)

1. Loaded dataset from zip.
2. Cleaned data: added satisfaction_num, created AgeGroup.
3. Counted satisfaction levels & visualized.
4. Compared satisfaction across travel classes.
5. Found correlations of service features with satisfaction.
6. Built Decision Tree for feature importance.
7. Checked age group & travel type impact.
8. Analyzed delays vs satisfaction.
9. For dissatisfied passengers, identified worst-rated features (bubble chart).

Skills Learned

- Data cleaning & grouping (pandas).

- Visualization (seaborn, matplotlib).
- Correlation analysis.
- Decision Tree modeling.
- Insights on satisfaction drivers.