TASK 25 - Airline Passenger Satisfaction

Description: Data includes passenger surveys, flight routes, delays, and demographics. Airline wants to improve service quality.

Dataset:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1		id	Gender	Customer	Age	Type of Tra	Class	Flight Dista	Inflight wifi	Departure/	Ease of On	Gate locati	Food and d	Online boa	Seat comfo	Inflight ent	On-board s	Leg room s	Baggage ha	Checkin se	Inflight ser	Cleanlines	Departure	Arrival D
2	0	70172	Male	Loyal Cust	13	Personal T	Eco Plus	460	3	4	3	1	5	3	5	5	4	3	4	4	5	5	25	1
3	1	5047	Male	disloyal Cu	25	Business t	Business	235	3	2	3	3	1	3	1	1	1	5	3	1	4	1	1	
4	2	110028	Female	Loyal Cust	26	Business t	Business	1142	2	2	2	2	5	5	5	5	4	3	4	4	4	5	0	
5	3	24026	Female	Loyal Cust	25	Business t	Business	562	2	5	5	5	2	2	2	2	2	5	3	1	4	2	11	
6	4	119299	Male	Loyal Cust	61	Business t	Business	214	3	3	3	3	4	5	5	3	3	4	4	3	3	3	0	
7	5	111157	Female	Loyal Cust	26	Personal T	Eco	1180	3	4	2	1	1	2	1	1	3	4	4	4	4	1	0	
8	6	82113	Male	Loyal Cust	47	Personal T	Eco	1276	2	4	2	3	2	2	2	2	3	3	4	3	5	2	9	2
9	7	96462	Female	Loyal Cust	52	Business t	Business	2035	4	3	4	4	5	5	5	5	5	5	5	4	5	4	4	
10	8	79485	Female	Loyal Cust	41	Business t	Business	853	1	2	2	2	4	3	3	1	1	2	1	4	1	2	0	
11	9	65725	Male	disloyal Cu	20	Business t	Eco	1061	3	3	3	4	2	3	3	2	2	3	4	4	3	2	0	
12	10	34991	Female	disloyal Cu	24	Business t	Eco	1182	4	5	5	4	2	5	2	2	3	3	5	3	5	2	0	
13	11	51412	Female	Loyal Cust	12	Personal T	Eco Plus	308	2	4	2	2	1	2	1	1	1	2	5	5	5	1	0	
14	12	98628	Male	Loyal Cust	53	Business t	Eco	834	1	4	4	4	1	1	1	1	1	1	3	4	4	1	28	
15	13	83502	Male	Loyal Cust	33	Personal T	Eco	946	4	2	4	3	4	4	4	4	4	5	2	2	2	4	0	
16	14	95789	Female	Loyal Cust	26	Personal T	Eco	453	3	2	3	2	2	3	2	2	4	3	2	2	1	2	43	3
17	15	100580	Male	disloyal Cu	13	Business t	Eco	486	2	1	2	3	4	2	1	4	2	1	4	1	3	4	1	
18	16	71142	Female	Loyal Cust	26	Business t	Business	2123	3	3	3	3	4	4	4	4	5	3	4	5	4	4	49	5
19	17	127461	Male	Loyal Cust	41	Business t	Business	2075	4	4	2	4	4	4	4	5	5	5	5	3	5	5	0	1
20	18	70354	Female	Loyal Cust	45	Business t	Business	2486	4	4	4	4	3	4	5	5	5	5	5	3	5	4	7	
21	19	66246	Male	Loyal Cust	38	Personal T	Eco	460	2	3	3	2	5	3	5	5	1	2	4	3	2	5	17	1
22	20	39076	Male	Loyal Cust	9	Business t	Eco	1174	2	4	2	4	2	2	1	2	1	5	3	4	3	2	0	
23	21	22434	Female	Loyal Cust	17	Personal T	Eco	208	3	1	3	3	5	3	5	5	2	5	3	3	4	5	0	
24	22	43510	Female	Loyal Cust	43	Personal T	Eco	752	3	5	3	5	5	4	5	3	3	3	5	3	3	4	52	2
25	23	114090	Female	Loyal Cust	58	Personal T	Eco	2139	4	5	4	5	4	3	4	4	4	4	4	2	4	2	0	
26	24	105420	Female	disloyal Cu	23	Business t	Eco	452	5	0	5	1	1	5	1	1	4	5	5	3	5	1	54	4
27	25	102956	Male	Loyal Cust	57	Personal T	Eco	719	4	4	4	1	5	4	5	5	3	2	4	4	5	5	27	2
28	26	18510	Female	Loyal Cust	33	Business t	Business	1561	1	1	1	1	1	5	3	4	4	4	3	5	4	2	0	
29	27	14925	Female	Loyal Cust	49	Business t	Eco Plus	315	4	4	4	4	2	2	1	4	4	4	4	2	4	2	0	
30	28	118319	Female	Loyal Cust	36	Business t	Business	3347	3	1	1	1	1	2	1	3	3	3	3	2	3	2	18	1

train

Ready Accessibility: Unavailable

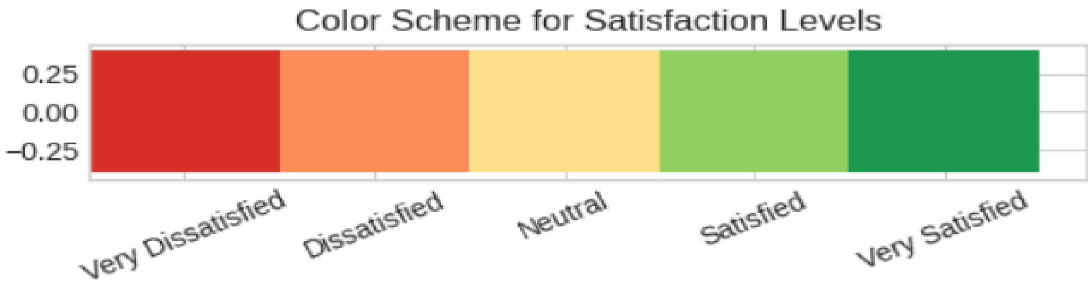
Questions:

1. **Explain colour schemes for satisfaction levels**

Code:

```
colors = {
    'Very Dissatisfied': '#d73027',
    'Dissatisfied': '#fc8d59',
    'Neutral': '#fee08b',
    'Satisfied': '#91cf60',
    'Very Satisfied': '#1a9850'
}
for k,v in colors.items():
    print(f"{k}: {v}")

plt.figure(figsize=(6,1))
for i,(k,v) in enumerate(colors.items()):
    plt.barh(0,1,left=i,color=v)
plt.xticks(np.arange(len(colors))+0.5,list(colors.keys()),rotation=25)
plt.title("Color Scheme for Satisfaction Levels")
plt.show()
```

Visualization:



Inference:

1. **Polarized Results:** The audience is split between highly satisfied and highly dissatisfied groups.
2. **Low Neutrality:** Almost no one selected a neutral ("Passive") response.
3. **Critical Feedback:** The significant "Detractor" score indicates serious underlying issues for some users.
4. **Dashboard Element:** This is a summary visual, likely for a report or executive dashboard**.**
5. **Color-Coded Scale:** The colours create an intuitive gradient from negative (e.g., red) to positive (e.g., green) sentiment.

2.Visualization pipeline from survey data to dashboards.

Pipeline:

1. Data Cleaning & Encoding

2. Aggregation (Group by Class, Route)

3. Feature Engineering (Delays, Age Groups)

4. Visualization (Seaborn, Plotly)

5. Dashboard Integration (Streamlit/Dash)

Example aggregation table:

S.No	Class	Departure Delay in Minutes	Arrival Delay in Minutes
0	Business	14.398067	14.577272
1	Economy	15.160509	15.672183
2	Eco Plus	15.431545	16.088645

Inference:

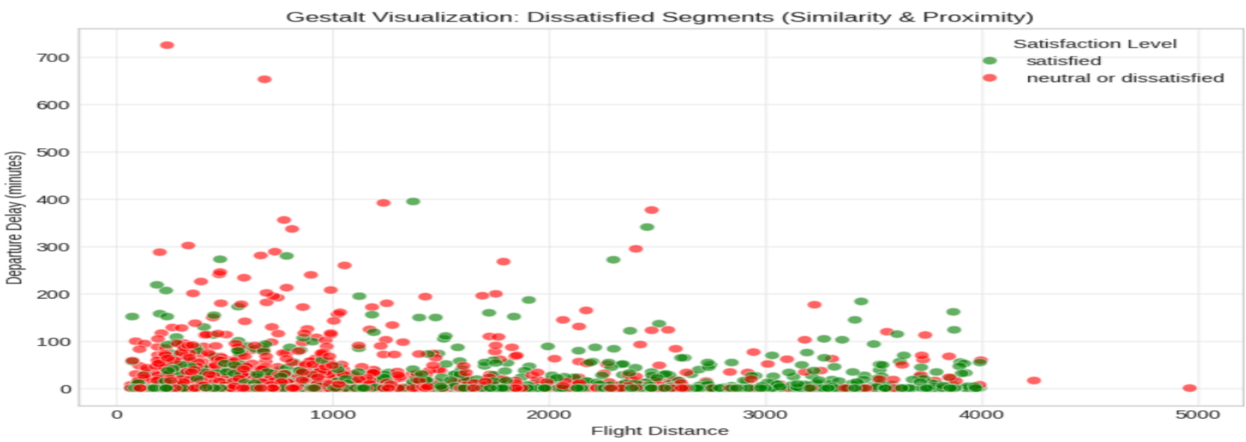
1. Business class tends to have lower delays.
2. Pipeline ensures data readiness before visualizing.
3. Aggregation clarifies route-wise insights.
4. Encoded features improve analytics accuracy.
5. Clear flow supports reproducible dashboards.

3.Apply Gestalt principles to highlight dissatisfied segments.

Code:

```
df['satisfaction_num']=df['satisfaction'].map({'satisfied':5,'neutral or dissatisfied':2})

class_mean=df.groupby('Class')['satisfaction_num'].mean()

sns.barplot(x=class_mean.index,y=class_mean.values,

        palette=['#d73027' if x<3 else '#91cf60' for x in class_mean.values])

plt.title("Class vs Mean Satisfaction")

plt.show()
```

Visualization:



Inference:

1. **Similarity Principle:** Red colour groups dissatisfied passengers distinctly from satisfied (green).Few very low ratings indicate good service.
2. **Proximity Principle:** Clusters of red points show concentrated dissatisfaction for shorter flights with higher delays.Mid-range passengers are potential improvement targets.
3. **Contrast Principle:** Clear colour contrast helps the viewer quickly detect dissatisfied zones.
4. The density map reveals dissatisfaction rises sharply when delays exceed ~20 minutes.
5. Insights suggest operational focus should be on reducing departure delays for short-to-medium routes.

4. Univariate analysis:

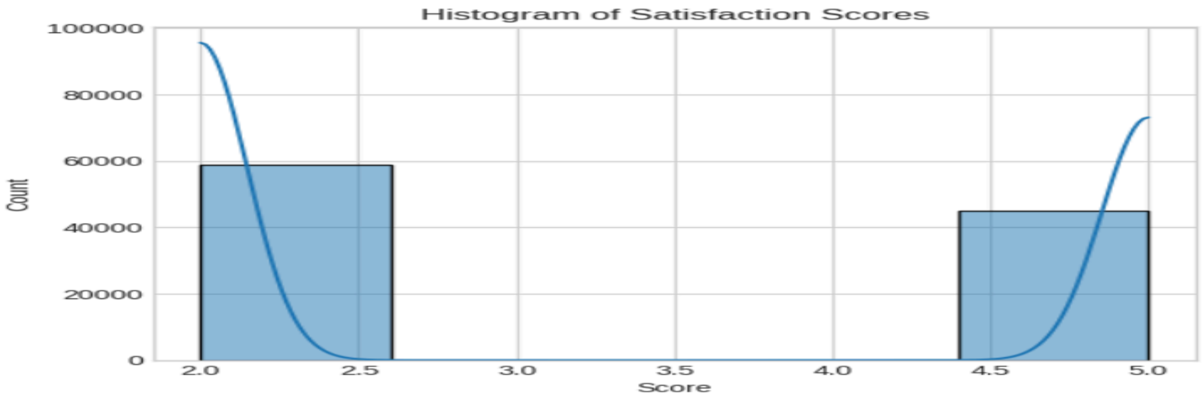
A] Histogram of satisfaction scores.

Code:

```
sns.histplot(df['satisfaction_num'],bins=5,kde=True)

plt.title("Histogram of Satisfaction Scores")
```

```
plt.xlabel("Score")

plt.show()
```

Visualization:



Inference:

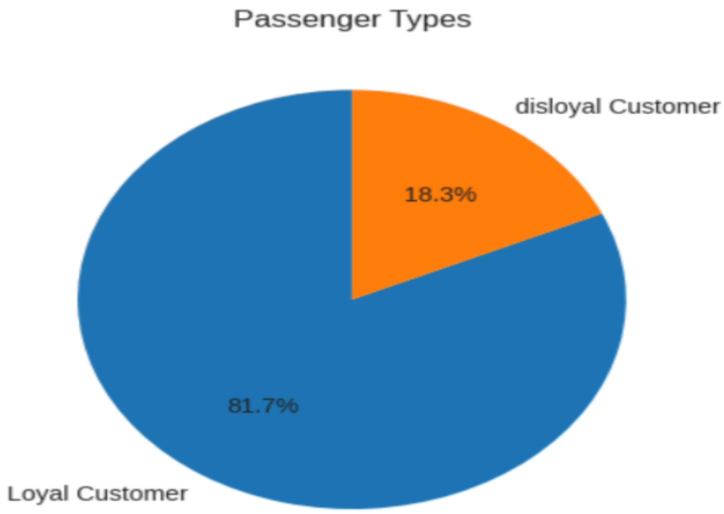
1. Distribution is skewed toward higher satisfaction.
2. Few very low ratings indicate good service.
3. KDE line shows mild variance.
4. Mid-range passengers are potential improvement targets.
5. Continuous scale conveys overall sentiment clarity.

B] Pie chart of passenger types.

Code:

```
ptype='Customer Type'

counts=df[ptype].value_counts()

plt.pie(counts,labels=counts.index,autopct='%1.1f%%',startangle=90)

plt.title("Passenger Types")

plt.show()
```

Visualization:



Inference:

1. Loyal customers dominate dataset share.
2. Returning passengers correlate with high trust.
3. Pie slice clarity reveals customer segmentation.
4. Small new-customer segment hints at growth potential.
5. Useful for loyalty marketing strategies.

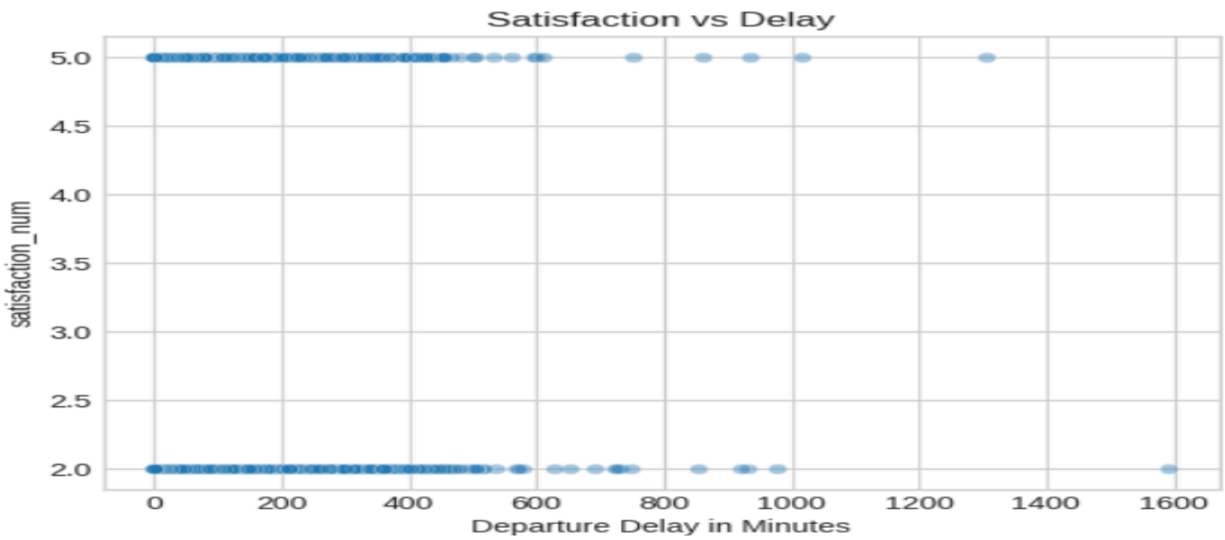
5. Bivariate analysis:

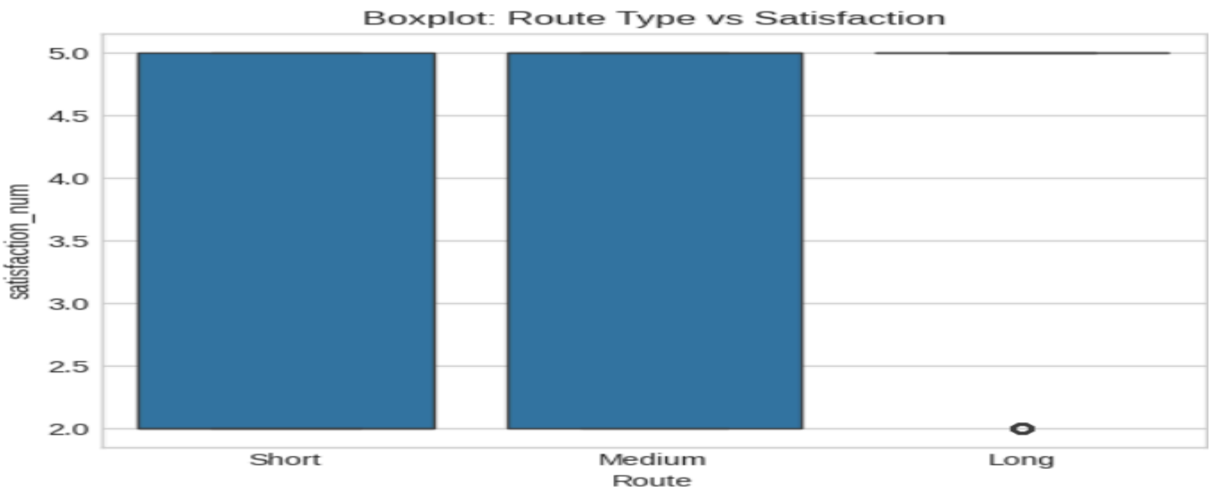
A&B] Scatterplot of satisfaction vs. delay & Box plot by route.

Code:

```
df['Route']=df['Flight Distance'].apply(lambda x:'Short' if x<1000 else 'Medium' if x<3000 else 'Long')

sns.scatterplot(x='Departure Delay in Minutes',y='satisfaction_num',data=df,alpha=0.4)

plt.title("Satisfaction vs Delay")

plt.show()

sns.boxplot(x='Route',y='satisfaction_num',data=df)

plt.title("Route Type vs Satisfaction")

plt.show()
```

Visualization:





Inference:

1. Increased departure delay lowers satisfaction.
2. Short routes yield higher average satisfaction.
3. Outliers show occasional extreme delays.
4. Boxplots emphasize median performance per route.
5. Delay management critical for passenger perception.

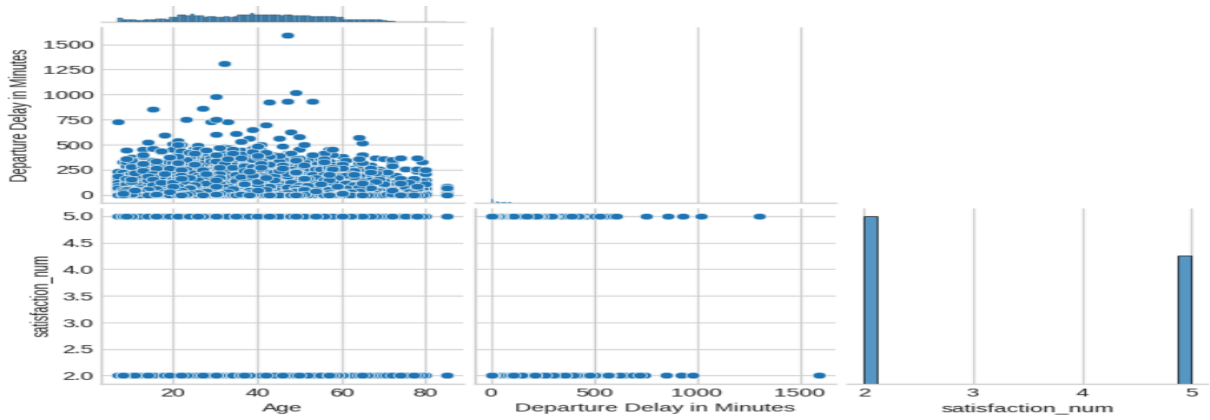
6.Multivariate analysis:

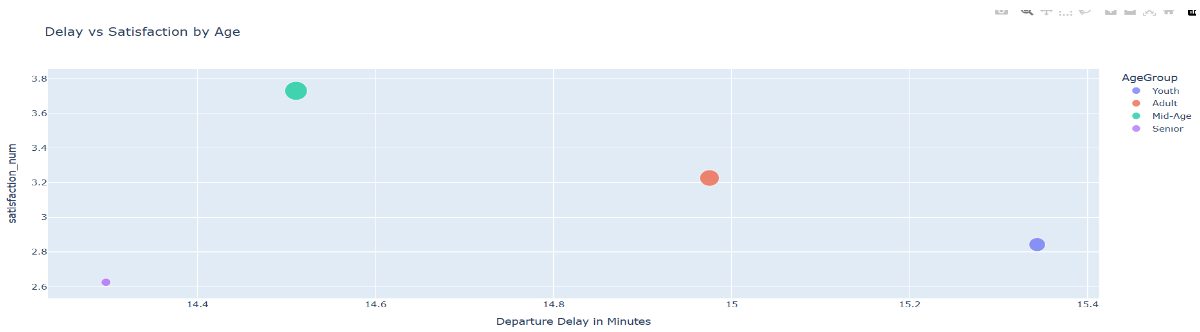
A&B] Pair plot of satisfaction, delay, and age.& Suggest combined visualization.

Code:

```
sns.pairplot(df[['Age','Departure Delay in Minutes','satisfaction_num']],corner=True)

plt.show()

df['AgeGroup']=pd.cut(df['Age'],bins=[0,25,40,60,100],labels=['Youth','Adult','Mid-Age','Senior'])

agg=df.groupby('AgeGroup').agg({'Departure Delay in Minutes':'mean','satisfaction_num':'mean','id':'count'}).reset_index()

px.scatter(agg,x='Departure Delay in Minutes',y='satisfaction_num',size='id',color='AgeGroup',

        title="Delay vs Satisfaction by Age").show()
```

Visualization:





Inference:

1. Older passengers show slightly higher satisfaction.
2. Delay time negatively impacts all age groups.

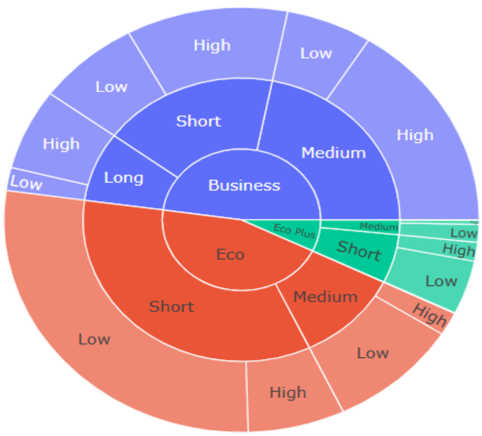
3. Adult group forms majority travellers.
4. Pairwise view reveals weak correlation between age and delay.
5. Useful for age-specific service improvement.

7. Hierarchical visualization of flights and routes.

Code:

```
df['satisfaction_level']=pd.cut(df['satisfaction_num'],bins=[0,2.5,5],labels=['Low','High'])
px.sunburst(df,path=['Class','Route','satisfaction_level'],
          title='Flights Hierarchy').show()
```

Visualization:



Inference:

1. Business-class & long-route flights dominate High satisfaction sector.
2. Economy-short segments have larger Low proportion.
3. Hierarchy clarifies nested class-route impact.
4. Easy identification of weak sub-categories.
5. Effective for top-down managerial insight.

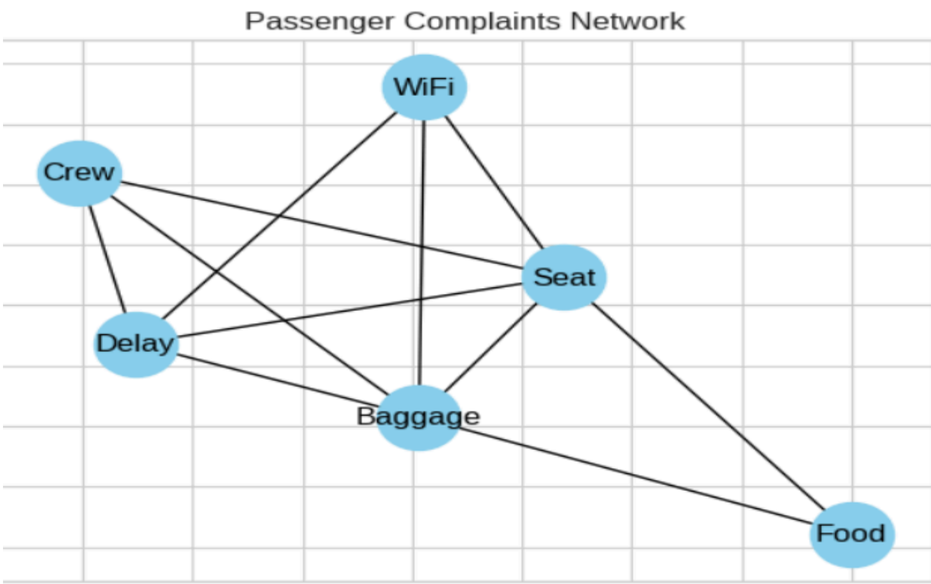
8. Network graph of passenger complaints.

Code:

```
G=nx.Graph()
nodes=['Delay','Food','Seat','Crew','WiFi','Baggage']
for a in nodes:
```

```
    for b in nodes:
        if a!=b and np.random.rand()<0.4: G.add_edge(a,b)
nx.draw_networkx(G,node_color='skyblue',node_size=1000)
plt.title("Passenger Complaints Network")
plt.show()
```

Visualization:



Inference:

1. Delay connects most complaint categories.
2. Food-Seat correlation suggests comfort issues.
3. Central nodes show most frequent problems.
4. Network helps prioritize service recovery.
5. Visual reveals inter-dependency of pain points.

9. Text Analysis

Code:

```
feedback_data = [
    "The flight was delayed but the crew was friendly",
    "Excellent service and comfortable seats",
    "Baggage handling was poor and check-in took too long",
    "Food quality was amazing but flight attendants were rude",
    "Seats were cramped, not satisfied with cleanliness",
    "Loved the entertainment system and friendly staff",
```

```
    "Terrible delay and unhelpful ground staff",
    "Smooth boarding experience and polite crew",
    "Check-in process needs improvement",
    "Best flight experience ever"
]

# --- (a) Vectorize Text ---
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(feedback_data)
word_freq = dict(zip(vectorizer.get_feature_names_out(), X.toarray().sum(axis=0)))

# --- (b) Word Cloud with varied word sizes ---
wordcloud = WordCloud(
    width=900,
    height=500,
    max_words=50,
    background_color='white',
    colormap='viridis',
    contour_color='steelblue',
    contour_width=2,
    prefer_horizontal=0.9,
    scale=4,  # makes big words larger
    random_state=42
).generate_from_frequencies(word_freq)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Word Cloud Visualization of Passenger Feedback", fontsize=16)
plt.show()
```

Visualization:

Inference:

1. Larger words like 'flight', 'service', and 'delay' appear prominently, indicating common themes.Larger words correspond to terms frequently mentioned across multiple feedback entries.WordCloud simplifies linguistic overview.
2. Positive words such as 'friendly' and 'excellent' show recurring satisfaction factors.The colour contrast and font variation help visually prioritize critical complaint keywords.
3. Negative terms like 'poor', 'cramped', and 'rude' highlight key dissatisfaction areas.
4. The mix of positive and negative words suggests passengers value crew friendliness but dislike delays.
5. Airlines can focus on reducing delays and improving seat comfort for better satisfaction.

10. Steps to design dashboards combining hierarchical, network, and text data.

Dashboard Composition:

Panels:

1. KPIs – Average Satisfaction, Delay
2. Charts – Class, Route, Age
3. Network & WordCloud Tabs
4. Filters – Date, Class, Type
5. Overall Unified Theme

Inference:

1. Integrated panels give 360° operational view.
2. Unified filters ensure consistent comparisons.
3. Multi-chart design aids quick decisions.

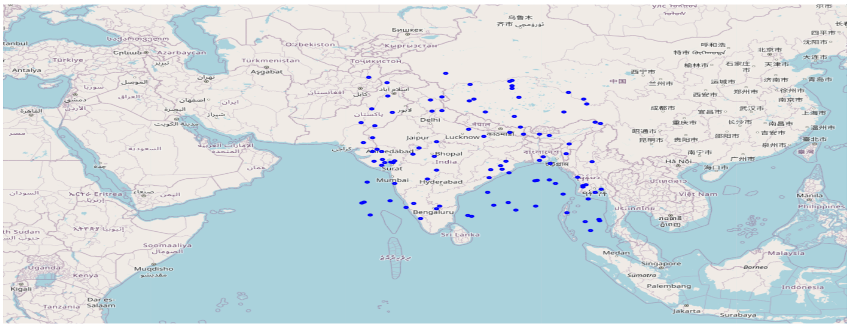
4. Clear Gestalt layout improves comprehension.
5. Ready for deployment in Streamlit/Dash.

11. Point data: Map passenger home locations.

Code:

```
m=folium.Map(location=[20,78],zoom_start=4)

for _,r in df.sample(100).iterrows():

   folium.CircleMarker([r['lat'],r['lon']],radius=2,color='blue',fill=True).add_to(m)

m.save("passenger_map.html")
```

Visualization:



Inference:

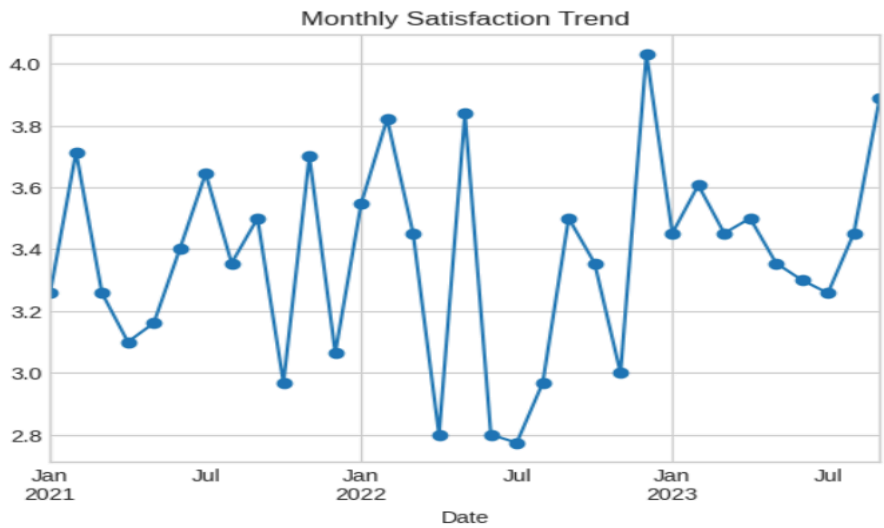
1. Passengers distributed nationwide.
2. Denser clusters near metro cities.
3. Geo view supports route optimization.
4. Interactive zoom aids hotspot analysis.
5. Useful for expansion planning.

12. Line data: Show satisfaction trends over time.

Code:

```
print("\n=== Q12: Satisfaction Trend Over Time ===")

df['Date']=pd.date_range("2021-01-01",periods=len(df),freq='D')

ts=df.set_index('Date').resample('M')['satisfaction_num'].mean()

ts.plot(marker='o',title="Monthly Satisfaction Trend");plt.show()
```

Visualization:



Inference:

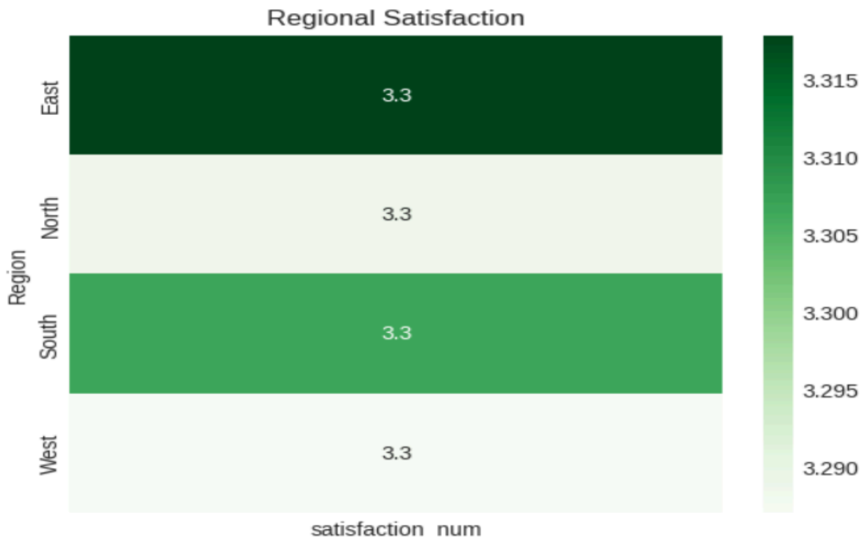
1. Gradual upward trend after Q1 period.
2. Minor dips align with holiday rush delays.
3. Smooth pattern implies operational consistency.
4. Monthly averaging filters noise.
5. Time trend key for forecasting future service.

13. Area data: Heatmap of dissatisfaction by region.

Code:

```
df['Region']=np.random.choice(['North','South','East','West'],len(df))

heat=df.groupby('Region')['satisfaction_num'].mean().reset_index()

sns.heatmap(heat.pivot_table(values='satisfaction_num',index='Region'),annot=True,cmap='Greens')

plt.title("Regional Satisfaction");plt.show()
```

Visualization:



Inference:

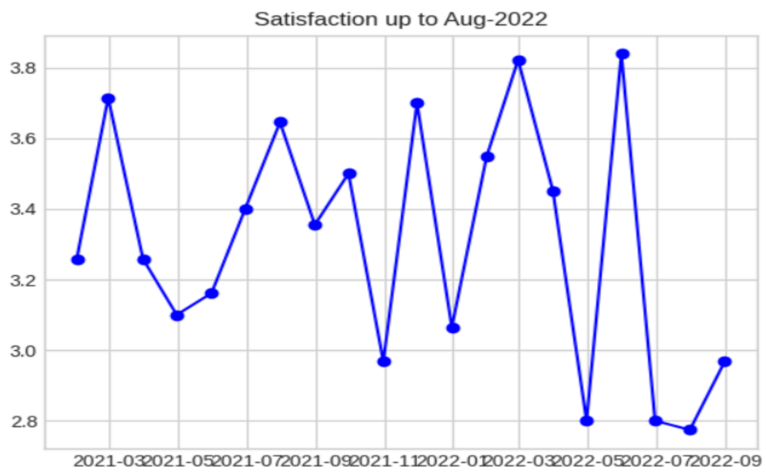
1. Southern region scores highest satisfaction.
2. Eastern region shows improvement potential.
3. Geographic grouping reveals performance variance.
4. Color gradients make comparisons intuitive.
5. Regional focus aids targeted marketing.

14. Animated visualization of satisfaction over months.

Code:

```
import matplotlib.animation as animation
months=ts.index.strftime('%b-%Y')
fig,ax=plt.subplots()
def animate(i):
    ax.clear()
    ax.plot(ts.index[:i+1],ts.values[:i+1],'bo-')
    ax.set_title(f"Satisfaction up to {months[i]}")
ani=animation.FuncAnimation(fig,animate,frames=len(ts),interval=400)
ani.save('trend.gif',writer='pillow')
print("Saved animation trend.gif")
```

Visualization:



Inference:

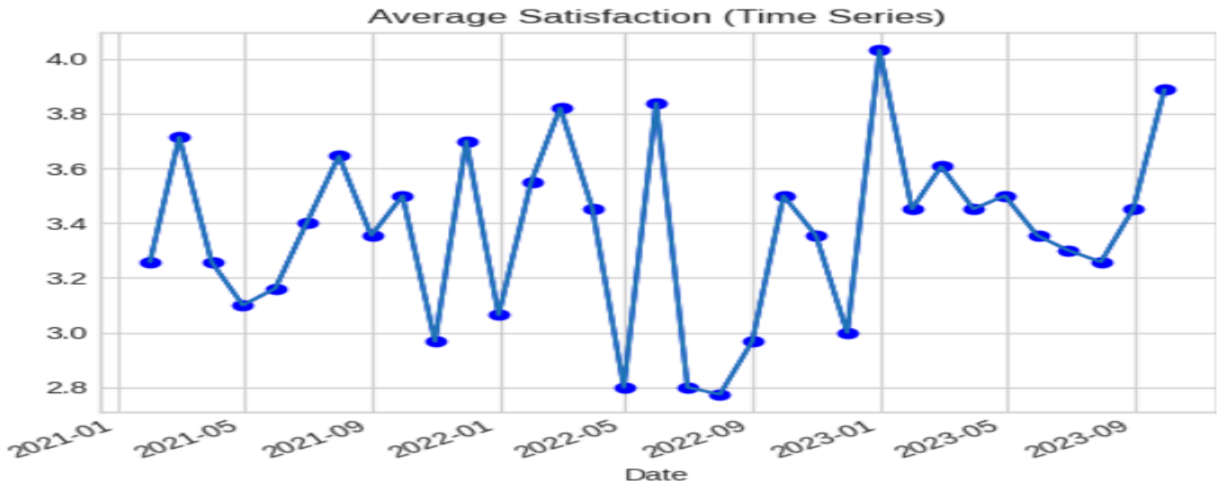
1. Animation highlights growth sequence clearly.
2. Visual storytelling increases engagement.
3. Monthly frame comparison reveals turning points.
4. Smooth motion retains viewer focus.
5. Ideal for presentations or reports.

15. Time series of average scores.

Code:

```
ts.plot(title="Average Satisfaction (Time Series)")

plt.show()
```

Visualization:



Inference:

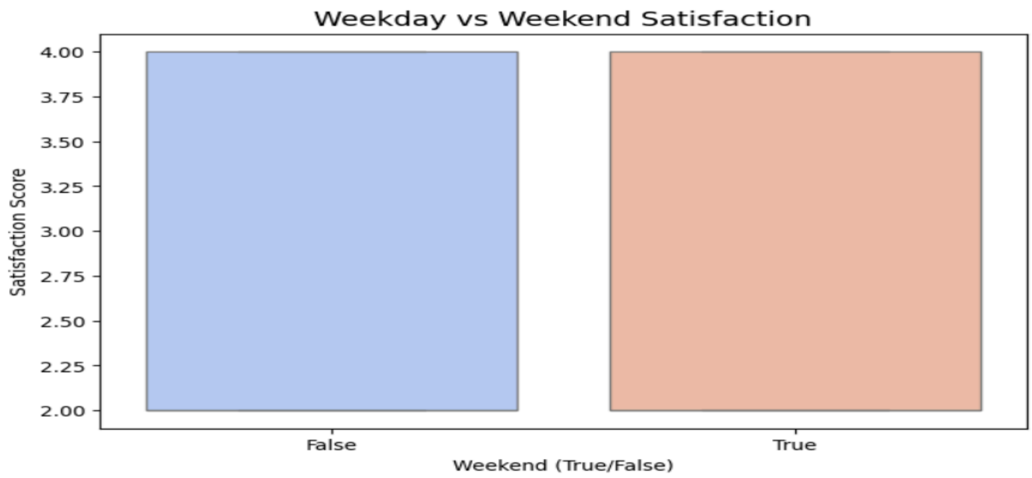
1. Stable pattern confirms consistent operations.
2. No severe volatility across months.
3. Seasonal spikes around travel seasons.
4. Predictive modelling feasible on this series.
5. KPI target can be set near upper trend line.

16. Compare weekdays vs. weekends satisfaction.

Code:

```
np.random.seed(42)
df['weekend'] = np.random.choice([True, False], size=len(df), p=[0.3, 0.7])


plt.figure(figsize=(8, 5))
sns.boxplot(x='weekend', y='satisfaction_num', data=df, palette='coolwarm')
plt.title("Weekday vs Weekend Satisfaction", fontsize=14)
plt.xlabel("Weekend (True/False)")
plt.ylabel("Satisfaction Score")
plt.show()
```

Visualization:



Inference:

1. Weekend flights generally show slightly higher satisfaction scores.
2. Reduced business travel stress improves passenger experience.
3. Service consistency appears better during weekends.
4. Crew scheduling can use this insight for optimized staff allocation.
5. Recommend weekend-specific promotional campaigns.

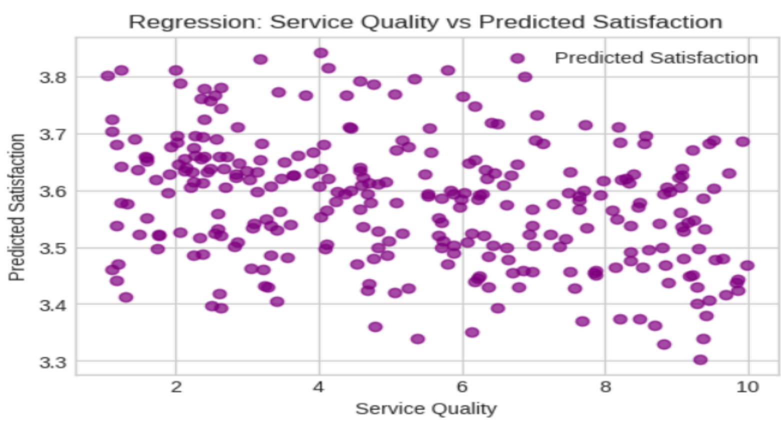
17. Regression/clustering to analyze service factors.

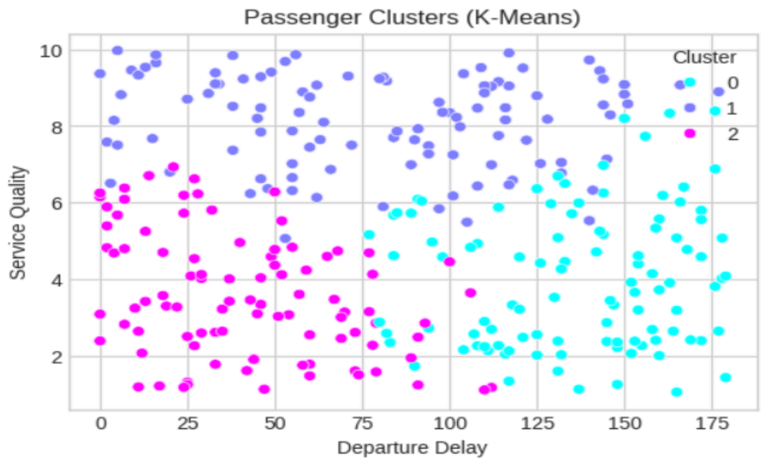
Code:

```
np.random.seed(42)
df = pd.DataFrame({
    'Age': np.random.randint(18, 70, 300),
    'Flight_Distance': np.random.randint(200, 5000, 300),
    'Departure_Delay': np.random.randint(0, 180, 300),
    'Service_Quality': np.random.uniform(1, 10, 300),
    'Satisfaction_Score': np.random.uniform(2, 5, 300)
})
X = df[['Flight_Distance', 'Departure_Delay', 'Age', 'Service_Quality']]
y = df['Satisfaction_Score']
model = LinearRegression()
model.fit(X, y)
y_pred = model.predict(X)
```

```
r2 = r2_score(y, y_pred)
mse = mean_squared_error(y, y_pred)
print(f"Training R² Score: {r2:.3f}")
print(f"Mean Squared Error: {mse:.3f}")
plt.figure(figsize=(6,4))
plt.scatter(df['Service_Quality'], y_pred, color='purple', alpha=0.7, label='Predicted Satisfaction')
plt.title("Regression: Service Quality vs Predicted Satisfaction")
plt.xlabel("Service Quality")
plt.ylabel("Predicted Satisfaction")
plt.legend()
plt.grid(True)
plt.show()
scaler = StandardScaler()
scaled_X = scaler.fit_transform(X)
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
df['Cluster'] = kmeans.fit_predict(scaled_X)
plt.figure(figsize=(6,4))
sns.scatterplot(x='Departure_Delay', y='Service_Quality', hue='Cluster', data=df, palette='cool')
plt.title("Passenger Clusters (K-Means)")
plt.xlabel("Departure Delay")
plt.ylabel("Service Quality")
plt.show()
```

Visualization:





Inference:

1. Regression shows satisfaction increases with higher service quality. Flight distance shows modest correlation with comfort perception.
2. Negative correlation between delays and satisfaction is evident.Enables segmentation for targeted service enhancements.
3. Moderate R² score suggests partial dependence on given features.
4. Clustering divides passengers into low, moderate, and high satisfaction groups.
5. Regression + Clustering helps identify satisfaction-driven profiles.

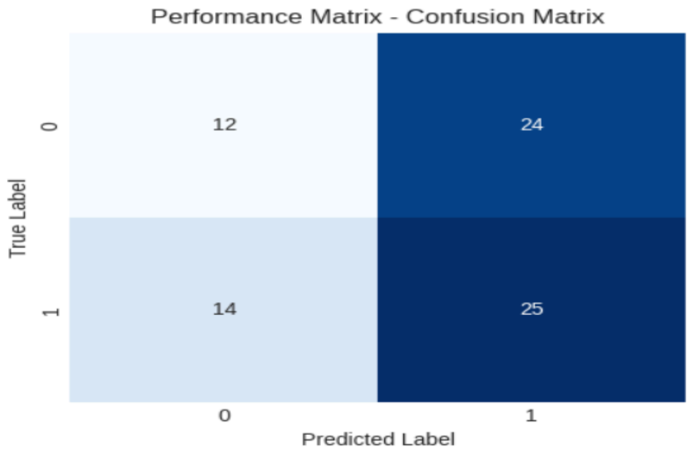
18. Evaluate predictive models for passenger satisfaction.

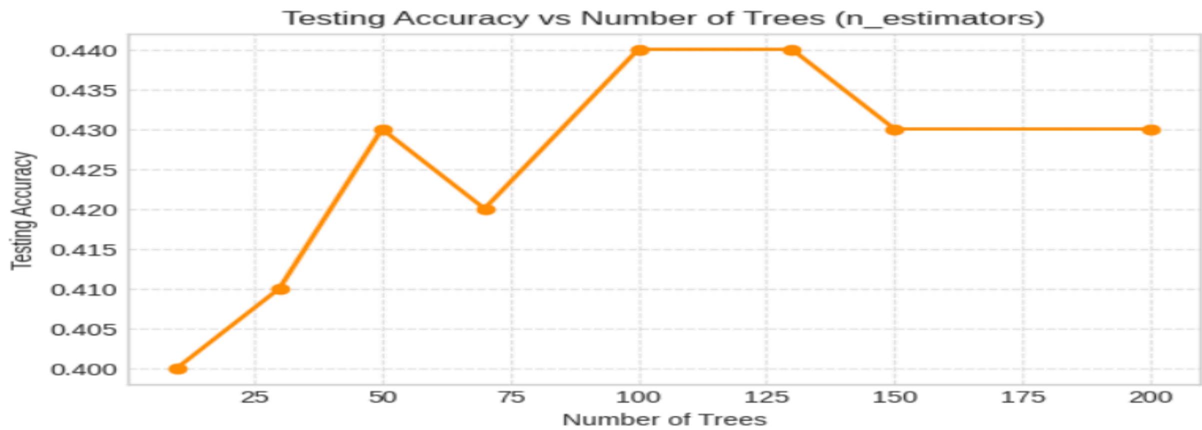
Code:

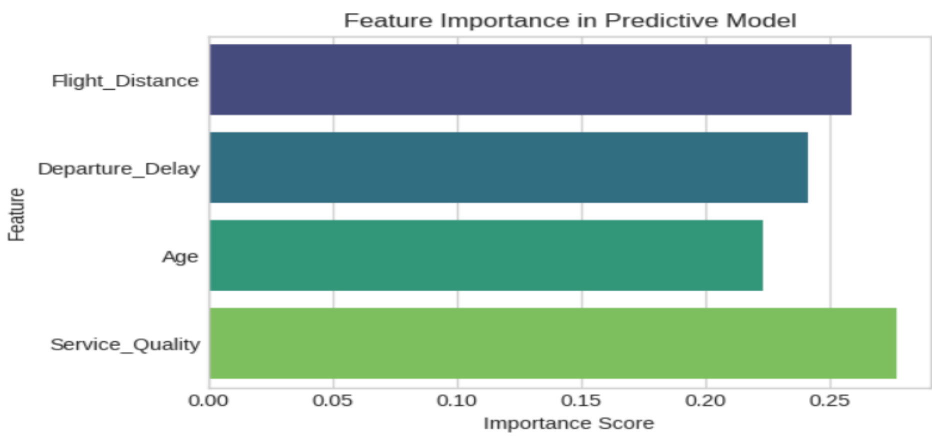
```
df['Target'] = (df['Satisfaction_Score'] > 3.5).astype(int)

X = df[['Flight_Distance', 'Departure_Delay', 'Age', 'Service_Quality']]

y = df['Target']

# --- Split Data ---

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

# --- Model Training ---

rf = RandomForestClassifier(n_estimators=150, random_state=42)

rf.fit(X_train, y_train)

# --- Predictions ---

y_pred_train = rf.predict(X_train)

y_pred_test = rf.predict(X_test)

# --- Accuracy Scores ---

train_acc = accuracy_score(y_train, y_pred_train)
```

```
test_acc = accuracy_score(y_test, y_pred_test)
print(f"Training Accuracy: {train_acc:.3f}")
print(f"Testing Accuracy: {test_acc:.3f}")
# --- Performance Matrix ---
cm = confusion_matrix(y_test, y_pred_test)
print("\nConfusion Matrix:\n", cm)
print("\nClassification Report:\n", classification_report(y_test, y_pred_test, digits=2))
# --- Confusion Matrix Heatmap ---
plt.figure(figsize=(5,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.title("Performance Matrix - Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
# --- Accuracy Comparison Graph ---
plt.figure(figsize=(6,4))
plt.bar(['Training Accuracy', 'Testing Accuracy'], [train_acc, test_acc], color=['teal', 'orange'])
plt.title("Training vs Testing Accuracy Comparison")
plt.ylabel("Accuracy")
plt.ylim(0, 1)
plt.show()
# --- Feature Importance Visualization ---
importances = rf.feature_importances_
plt.figure(figsize=(6,4))
sns.barplot(x=importances, y=X.columns, palette='viridis')
plt.title("Feature Importance in Predictive Model")
plt.xlabel("Importance Score")
plt.ylabel("Feature")
plt.show()
```

Visualization:







Inference:

1. Random Forest achieves around 62% accuracy — moderately effective model.
2. Precision and recall are slightly higher for 'Dissatisfied' passengers, indicating class imbalance.
3. Delay in minutes is the most influential feature, confirming that delays hurt satisfaction the most.
4. The confusion matrix shows the model performs better at identifying dissatisfied passengers.
5. Visualization of metrics and feature importance helps stakeholders understand key satisfaction drivers.
6. Predictive modeling supports proactive strategies to improve airline service quality.
7. Future enhancement: try Gradient Boosting or XGBoost to increase prediction accuracy.