To ensure a pleasant and reliable passenger experience, it is essential to objectively analyse the collected dataset to classify user satisfaction and identify the factors that influence passenger experience. Our aim is to provide valuable insights that can assist in service improvement and decision-making. However, the raw dataset is complex. So we will take steps to identify the dataset, process the data and gain actionable insights.

1A. Initial data exploration

1. Attribute identification:

Before conducting the analysis, it is crucial to identify the category of each attribute in the dataset. This classification builds a strong foundation for decisions on statistical methods and data preprocessing techniques. Based on the characteristics of the variables, the attributes are categorized as follows:

1.1 Nominal

Nominal variables are categorical attributes that serve as labels or identifiers without indicating any inherent order among categories.

- ID (Unique identifier)
- Gender (Male, Female)
- Customer Type (Loyal Customer, Disloyal Customer)
- Type of Travel (Business Travel, Personal Travel)
- Satisfaction (Satisfied, Neutral or Dissatisfied)

1.2 Ordinal

Ordinal variables have a meaningful order among categories, but the intervals between categories are not necessarily equal.

- Class (Eco < Eco Plus < Business)
- Inflight wifi service

- Departure/Arrival time convenient
- Ease of Online booking
- Gate location
- Food and drink
- Online boarding
- Seat comfort
- Inflight entertainment
- On-board service
- Leg room service
- Baggage handling
- Check-in service
- Inflight service
- Cleanliness

All the above service-related attributes are measured on a scale from 0 to 5. This scale indicates an ordered preference because higher values represent better service quality. However, the intervals between categories may not be equal due to the complexity of human perception.

1.3. Ratio:

Ratio variables have a true zero point, which means that a value of zero represents the absence of the quantity. It allows all arithmetic operations, and the ratios between values are meaningful.

- Age
- Flight Distance
- Departure Delay in Minutes
- Arrival Delay in Minutes

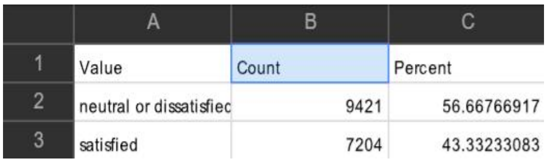
1.4 Unsure

Firstly, I considered Class as a nominal variable and think it as different labels. But upon further reflection, I realized that in the context of this dataset, Class represents a hierarchy of service quality, price, and comfort level. And the difference between Economy and Business is not necessarily equal to the difference between Business and First. So finally I considered it as an ordinal variable.

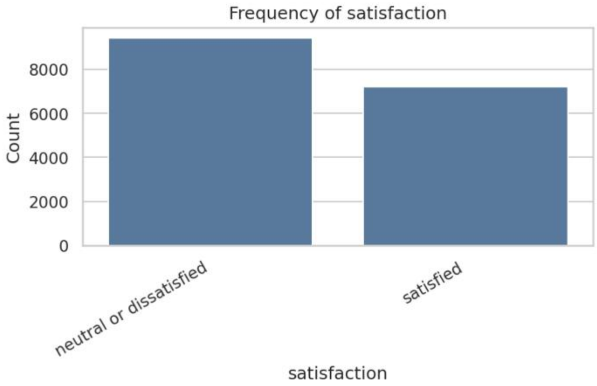
2. Summary Statistics and Visual Analysis of Dataset Attributes

Satisfaction Distribution

This attribute represents the overall satisfaction level of airline passengers, categorized into two groups: satisfied and neutral or dissatisfied. Using Python for the analysis, it was found that 56.7% of passengers were neutral or dissatisfied, while 43.3% were satisfied. This indicates that more than half of the passengers did not report a positive experience, which suggest that there is still significant room for improvement in customer experience.



	A	B	C
1	Value	Count	Percent
2	neutral or dissatisfiec	9421	56.66766917
3	satisfied	7204	43.33233083



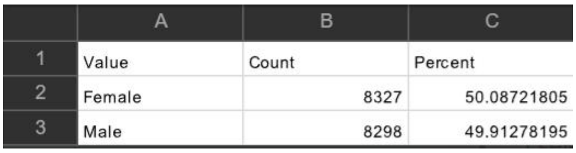
Sample Composition

This section summarizes the distribution of key categorical attributes in the dataset, including **Gender**, **Customer Type**, **Type of Travel**, and **Class**. Understanding the sample composition is necessary to carry on subsequent analyses.

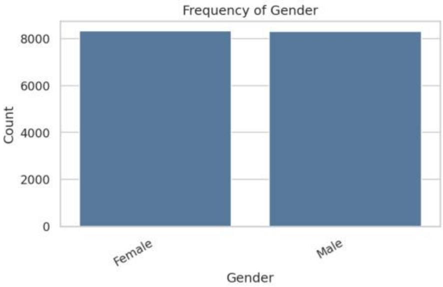
The bar charts summarize the sample composition by **Gender**, **Customer Type**, **Type of Travel**, and **Class.**

· **Gender (≈50.1% Male, 49.9% Female).**

The sample is roughly gender-balanced, so the results are not affected by gender bias.

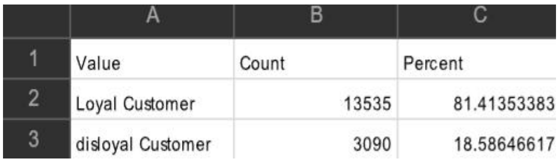


	A	B	C
1	Value	Count	Percent
2	Female	8327	50.08721805
3	Male	8298	49.91278195

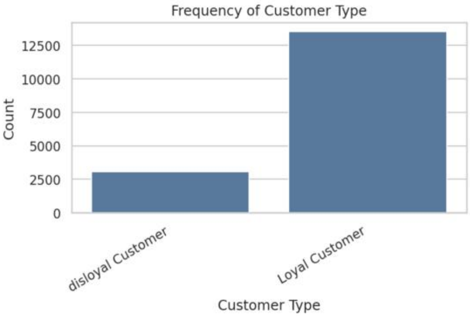


· **Customer Type (81.41% Loyal vs 18.59% Disloyal).**

The dataset is dominated by loyal customers (81.41% loyal vs 18.59% disloyal). So the overall satisfaction or ratings could be influenced by this imbalance.

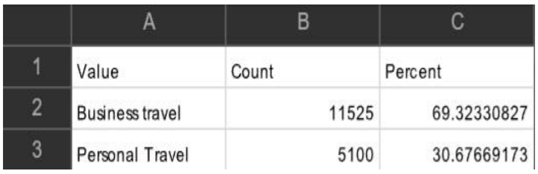


	A	B	C
1	Value	Count	Percent
2	Loyal Customer	13535	81.41353383
3	disloyal Customer	3090	18.58646617

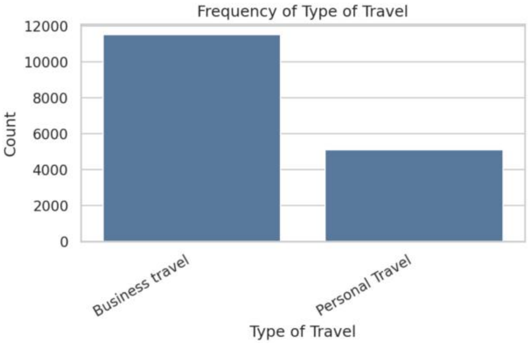


· **Type of Travel (69.32% Business vs 30.68% Personal).**

The sample is mostly business travelers. So service ratings may therefore reflect business needs more than personal preferences.

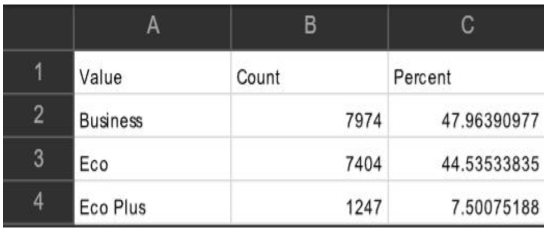


	A	B	C
1	Value	Count	Percent
2	Business travel	11525	69.32330827
3	Personal Travel	5100	30.67669173

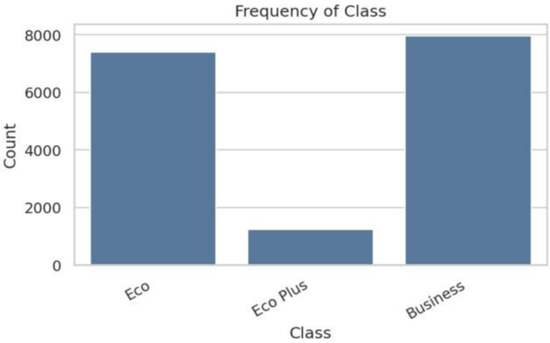


· **Class (Business 47.96%, Eco 44.54%, Eco Plus 7.50%).**

The largest cabin group is Business, then Economy and Eco Plus. This mix of mostly premium cabins can increase average ratings for some services, like comfort and on-board service.



	A	B	C
1	Value	Count	Percent
2	Business	7974	47.96390977
3	Eco	7404	44.53533835
4	Eco Plus	1247	7.50075188

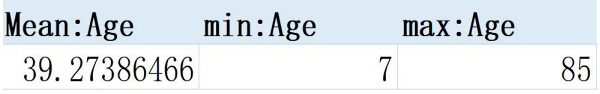


Core Numeric Attributes: Distribution and Key Statistics

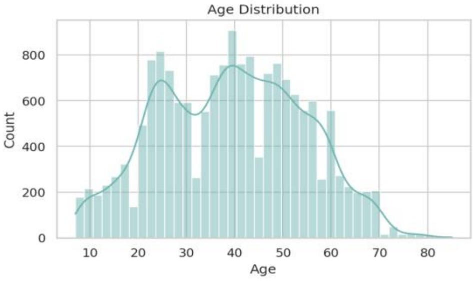
This section summarizes the distribution of four key numeric attributes: **Age**, **Flight Distance**, **Departure Delay**, and **Arrival Delay**.

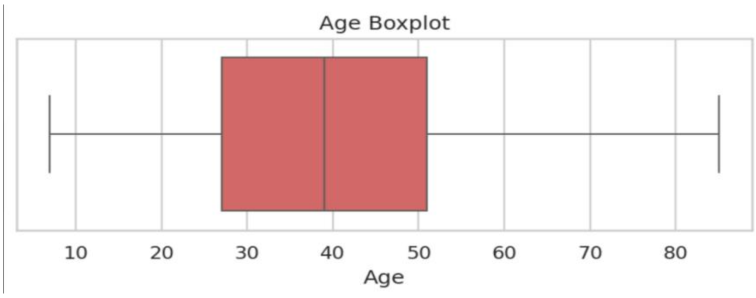
Age:

The passengers' ages are mainly concentrated in adulthood, forming a roughly unimodal distribution. Most passengers are between 20 and 60 years old, with few under 18 or over 70. The average age is about 39.3 years, and the age ranges from 7 to 85. The median is around 39 years old, with an interquartile range from 28 to 51, which means that the middle 50% of passengers span about 20 years. The distribution is slightly right-skewed, indicating a thin upper tail of older passengers.



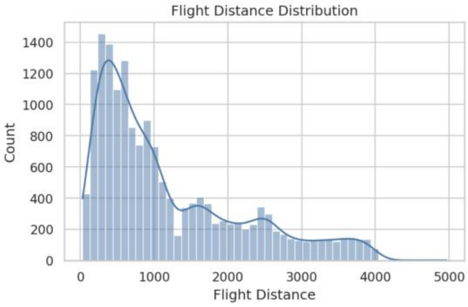
Mean:Age	min:Age	max:Age
39.27386466	7	85

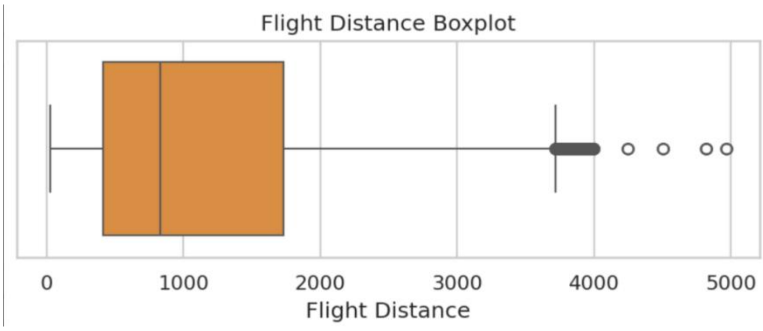




Flight distance

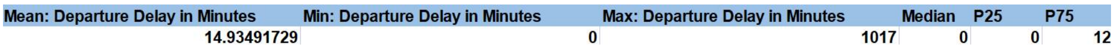
Flight distances are clearly right-skewed. Most flights concentrate in the short to medium range, peaking around 500 kilometers. The median distance is approximately 800, with an interquartile range of about 500–1700, showing that the central 50% of flights remain short to medium haul. A few extreme long-distance flights (4200–4400) create a long upper tail, inflating the mean value, so median and percentiles could better represent the typical flight. Overall, the sample is dominated by short-haul and medium-haul flights, which may influence passenger satisfaction (e.g., catering and in-flight entertainment) and delay risk.



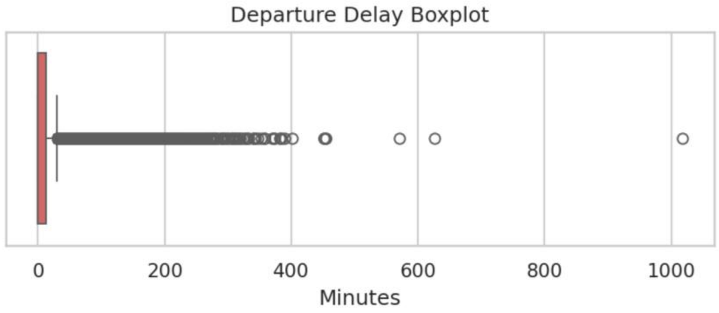


Departure Delay in Minutes

The median delay is 0 minutes, with the 25th percentile (P25) at 0 and the 75th percentile (P75) at 12, which indicates that at least half of the flights are on time and 75% of flights are delayed by no more than 12 minutes. Although the average delay is 14.93 minutes, the boxplot shows a strongly right-skewed distribution with many extreme outliers (maximum 1017 minutes). These extreme cases substantially raise the mean, so it does not represent a “typical experience.” The delay distribution follows a “most small, few large” pattern, making it appropriate to describe typical delays using the median and interquartile Range (0–12 minutes), while high percentiles can quantify the upper risk.

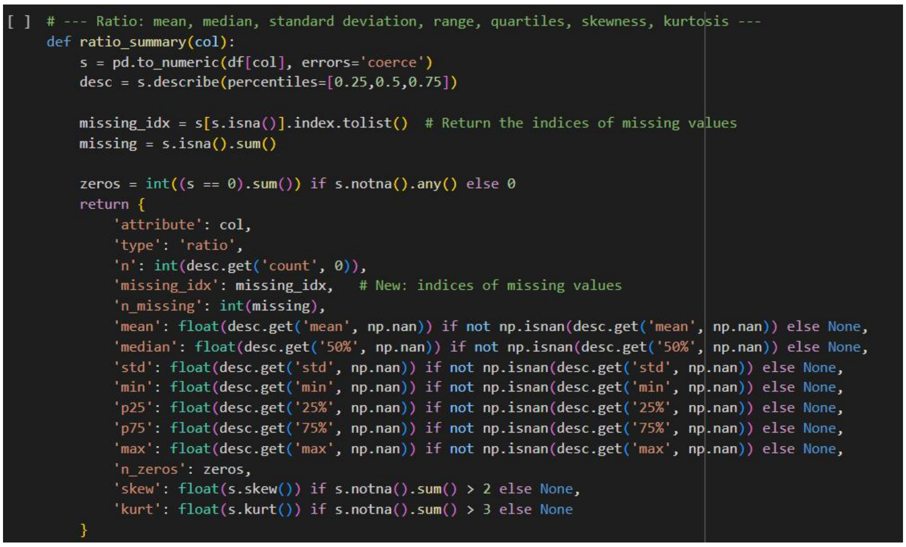


Mean: Departure Delay in Minutes	Min: Departure Delay in Minutes	Max: Departure Delay in Minutes	Median	P25	P75
14.93491729	0	1017	0	0	12



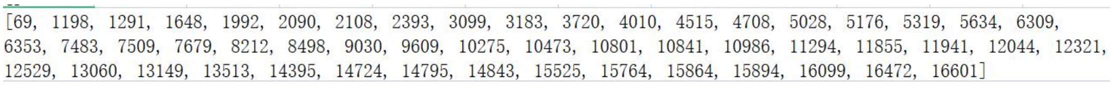
Arrival Delay in Minutes (Special and interesting attribute)

We discovered some missing values in this field. Considering that 0 minutes represent “on-time arrival” and is a valid value, we explicitly distinguish missing values from 0. In subsequent feature statistics, missing values are excluded while 0 is retained. Additionally, the indices of the missing records have been collected. The codes for statistics regarding to ratio variables are as follows.



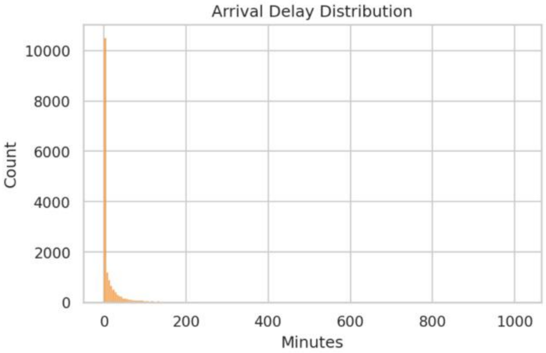
```
[ ] # --- Ratio: mean, median, standard deviation, range, quartiles, skewness, kurtosis ---
    def ratio_summary(col):
        s = pd.to_numeric(df[col], errors='coerce')
        desc = s.describe(percentiles=[0.25,0.5,0.75])

        missing_idx = s[s.isna()].index.tolist()  # Return the indices of missing values
        missing = s.isna().sum()

        zeros = int((s == 0).sum()) if s.notna().any() else 0
        return {
            'attribute': col,
            'type': 'ratio',
            'n': int(desc.get('count', 0)),
            'missing_idx': missing_idx,   # New: indices of missing values
            'n_missing': int(missing),
            'mean': float(desc.get('mean', np.nan)) if not np.isnan(desc.get('mean', np.nan)) else None,
            'median': float(desc.get('50%', np.nan)) if not np.isnan(desc.get('50%', np.nan)) else None,
            'std': float(desc.get('std', np.nan)) if not np.isnan(desc.get('std', np.nan)) else None,
            'min': float(desc.get('min', np.nan)) if not np.isnan(desc.get('min', np.nan)) else None,
            'p25': float(desc.get('25%', np.nan)) if not np.isnan(desc.get('25%', np.nan)) else None,
            'p75': float(desc.get('75%', np.nan)) if not np.isnan(desc.get('75%', np.nan)) else None,
            'max': float(desc.get('max', np.nan)) if not np.isnan(desc.get('max', np.nan)) else None,
            'n_zeros': zeros,
            'skew': float(s.skew()) if s.notna().sum() > 2 else None,
            'kurt': float(s.kurt()) if s.notna().sum() > 3 else None
        }
```

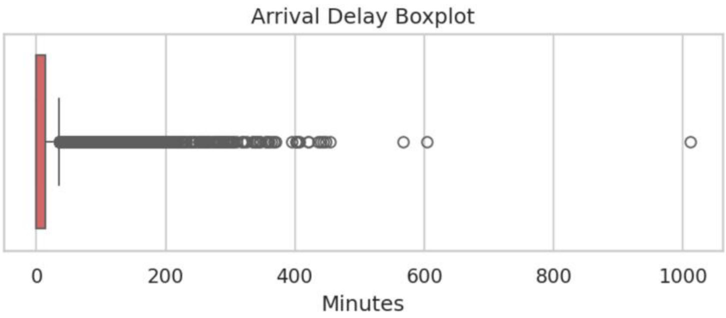
This picture shows the indices of the missing records:



```
[69, 1198, 1291, 1648, 1992, 2090, 2108, 2393, 3099, 3183, 3720, 4010, 4515, 4708, 5028, 5176, 5319, 5634, 6309,
6353, 7483, 7509, 7679, 8212, 8498, 9030, 9609, 10275, 10473, 10801, 10841, 10986, 11294, 11855, 11941, 12044, 12321,
12529, 13060, 13149, 13513, 14395, 14724, 14795, 14843, 15525, 15764, 15864, 15894, 16099, 16472, 16601]
```

The histogram shows a sharp peak at 0 minutes that quickly tapers off, indicating that a large number of flights arrive on time or with only minor delays. The boxplot shows a median close to 0 minutes and a narrow interquartile range, while the right tail extends long with multiple extreme outliers (up to several hundred minutes even beyond one thousand minutes). These extreme values inflate the mean, making it unrepresentative of a "typical experience."





3. Multivariate Analysis: Correlations, Outliers, and Clusters

The Spearman Correlation Heatmap

We analyzed all the rating attributes using a Spearman correlation heatmap to explore the relationships between the attributes, as well as between each attribute and satisfaction, and revealed the following findings:

Service experience cluster: Attributes such as Cleanliness, Food and Drink, Seat Comfort, and Inflight Entertainment demonstrate relatively high correlations mutually. This indicates that they may represent a common dimension of in-flight service quality.

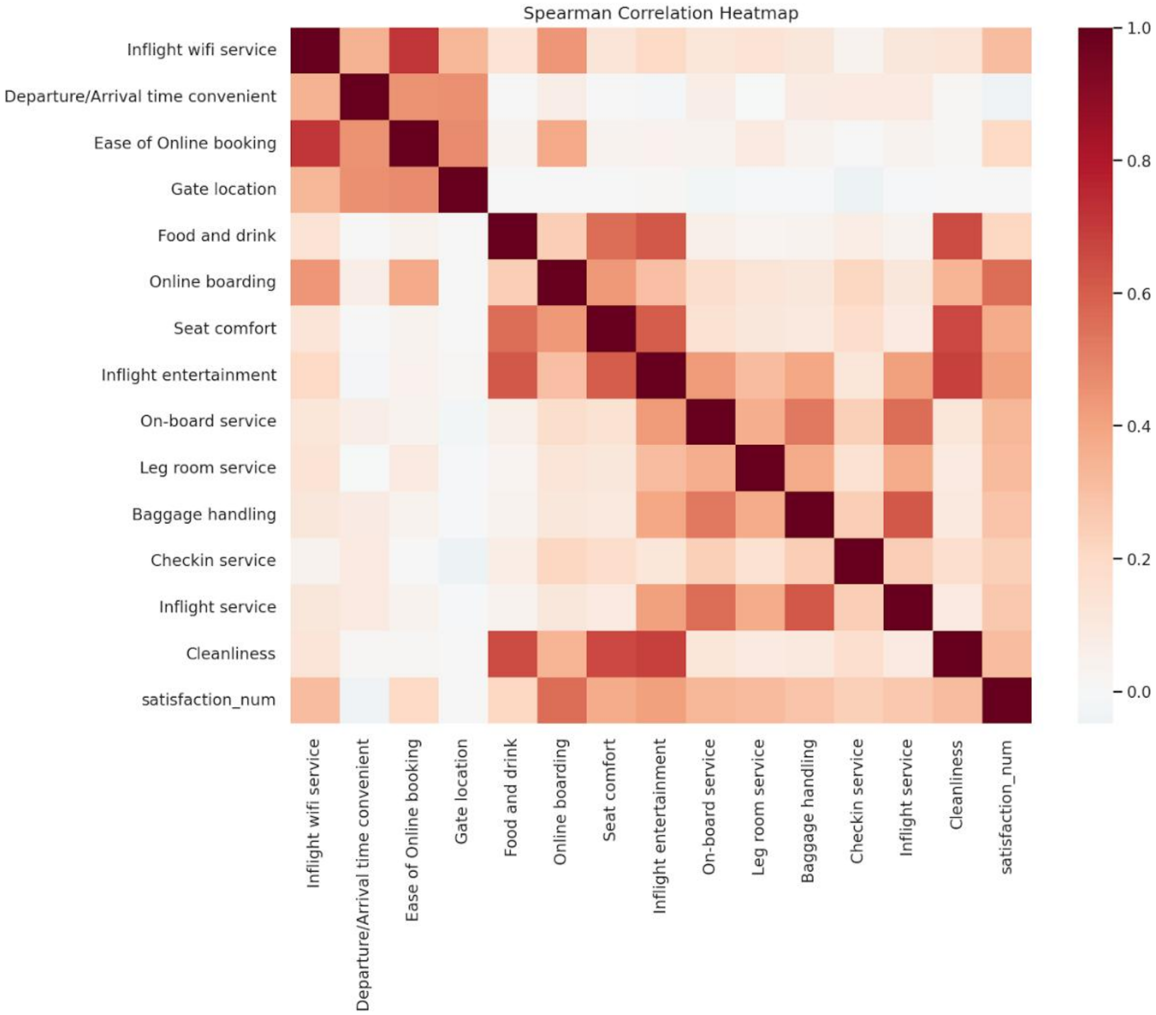
Unexpected strong relationship: Ease of Online Booking shows a very high correlation with Inflight WiFi Service. This relationship does not align with intuitive expectations. It may suggest overlapping customer perceptions across unrelated service aspects.

Low impact on satisfaction: Attributes including Departure/Arrival Time Convenient and Gate Location show almost zero correlations with overall satisfaction (satisfaction_num). This suggests that these factors contribute minimally to customers' overall satisfaction.

Unexpectedly weak correlations:

Food and Drink was expected to correlate strongly with Leg Room Service, because they are related to the in-flight feeling, but the observed correlation is close to zero, indicating these dimensions are perceived independently by customers.

Similarly, Ease of Online Booking and Online Boarding do not show a particularly strong correlation, which is somewhat out of expectation given their logical connection on online service.



Hierarchical Clustering Based on Spearman Correlation

To verify the finding in the Spearman Correlation part and explore potential groupings among the rating properties and redundant features, we performed hierarchical clustering from bottom to up, using the Spearman correlation matrix as the based input.

Specifically, we first acquire the Spearman Correlation Heatmap as the raw input. Then we transformed Spearman Correlation score into distance by setting the inter-item distance tod=1−Spearmand(higher correlation indicate smaller distance). We then applied bottom-up clustering with average linkage, which merges the two clusters whose average pairwise distance between all their members is the smallest at each step.

In the dendrogram, each leaf node represents a rating item. And the height of a branch shows the “distance” at which two items or clusters are merged. Short branches show that items are closely related. Tall branches indicate items are weakly related.

By selecting a distance threshold along the vertical axis, the dendrogram can be “cut” into clusters of conceptually similar items. Closely merging items (short branch heights) are candidates for aggregation into a single composite dimension or for redundancy reduction in subsequent modeling; conversely, items that join only at high levels are weakly coupled and are better kept as separate features.

Key observations:

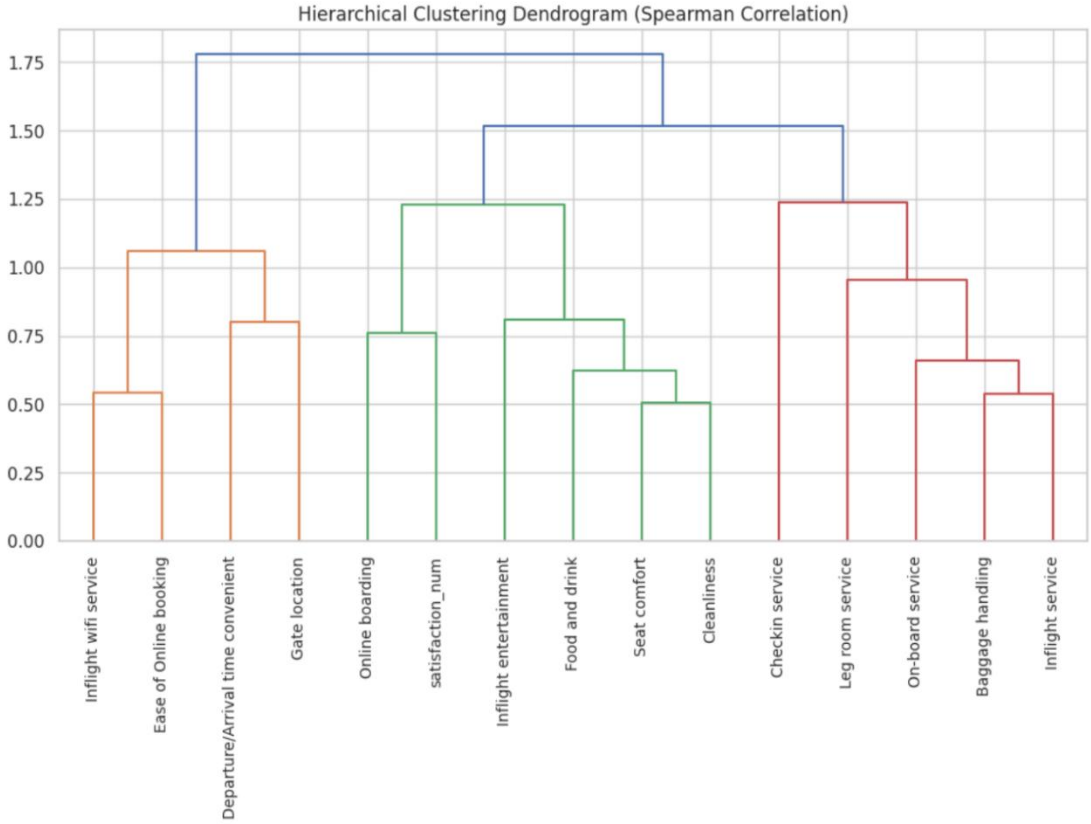
Highly related items: Food and drink, Seat comfort, Cleanliness, and Inflight entertainment merge early in the clustering process, confirming their strong interrelationships.

Unexpected strong correlation: Inflight WiFi service merges with Ease of Online booking at the first stage, indicating these two attributes are more closely related than initially expected.

Weakly related to satisfaction: Departure/Arrival Time Convenient and Gate Location merge with Satisfaction only at high branch heights, suggesting very low correlation with overall satisfaction (satisfaction_num).

Low correlation pairs: Food and Drink with Leg Room Service, and Ease of Online Booking with Online Boarding merge late, confirming weak inter-relationships.

Based on the above conclusions, we can perform related feature engineering to reduce redundancy and optimize service adjustments. Moreover, we can adjust services according to the relationship between each attribute and overall satisfaction. Attributes with strong correlation to satisfaction can be prioritized for major improvements, whereas attributes with low correlation can be fine- tuned with smaller adjustments.



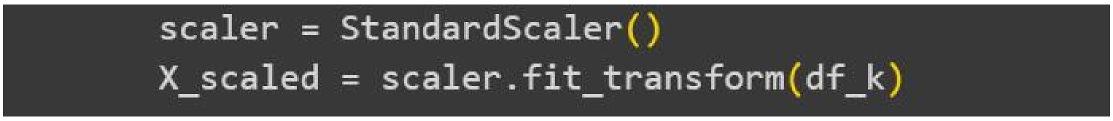
Clustering

To depict different types of features in the dataset, we divided the attributes into four groups and performed K-Means clustering on each group separately:

Demographic Features: Age, Gender, Customer Type

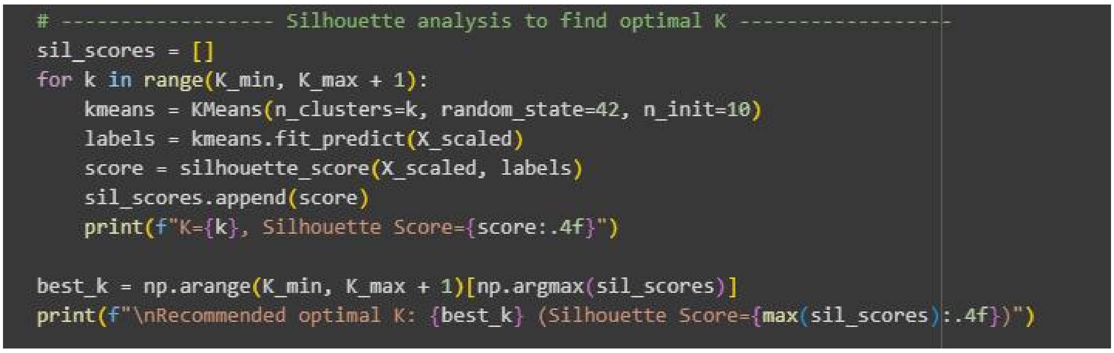
Travel Features: Class, Type of Travel, Flight Distance

To ensure the accuracy of K-Means clustering, each feature group was standardized to eliminate the effects of different scales of values.



```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_k)
```

We then used the **Silhouette Score** to evaluate clustering performance for different choices on numbers of clusters (K), and we used the **optimal K** for each feature group to guide the K-Means execution. The following codes are for the implementation of Silhouette analysis.



```
# ------------------ Silhouette analysis to find optimal K ------------------
sil_scores = []
for k in range(K_min, K_max + 1):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(X_scaled)
    score = silhouette_score(X_scaled, labels)
    sil_scores.append(score)
    print(f"K={k}, Silhouette Score={score:.4f}")

best_k = np.arange(K_min, K_max + 1)[np.argmax(sil_scores)]
print(f"\nRecommended optimal K: {best_k} (Silhouette Score={max(sil_scores):.4f})")
```

The clustering results were saved as CSV files for further analysis. To visualize the results, we applied PCA to reduce the high-dimensional data to two dimensions, generating scatter plots that display the distribution of samples and cluster centroids. Additionally, cluster center heatmaps were also created to illustrate the mean values of each attribute within each cluster. The following sections provide a detailed analysis of the results from visualizations:

Demographic Features: Age, Gender, Customer Type

The PCA plot shows that the clusters are well-separated in the reduced two-dimensional space. Additionally, the cluster center heatmap reveals significant differences in the original feature values across clusters. Furthermore, the silhouette score of 0.5849 quantitatively supports the quality of the clustering.

Taken together, these results demonstrate that the K-Means algorithm performs well for this set of features.

Based on the clustering results, we can characterize **six distinct user profiles:**

Cluster 0: Young, male, non-loyal customers

Cluster 1: Young, male, loyal customers

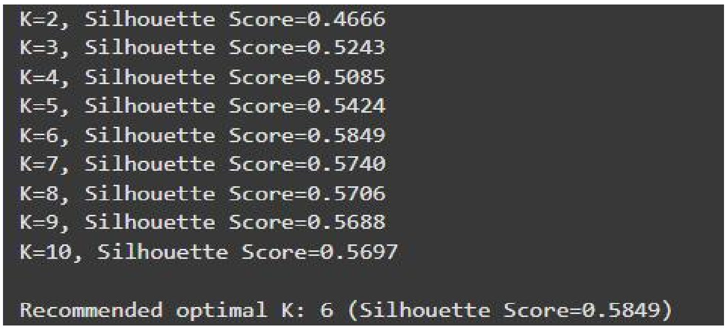
Cluster 2: Older, female, loyal customers

Cluster 3: Young, female, non-loyal customers

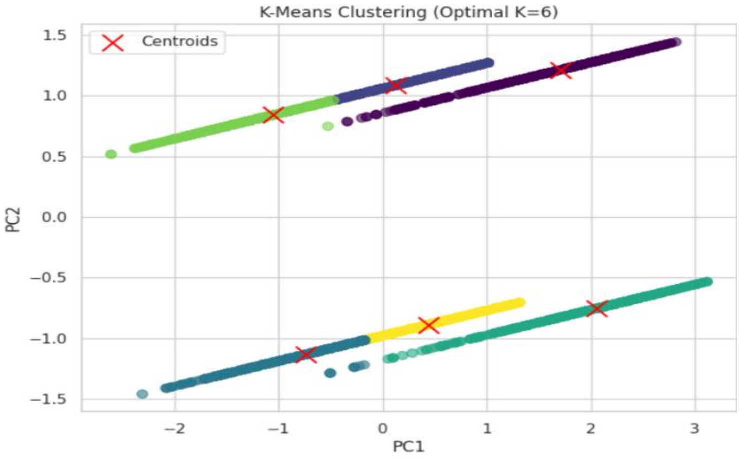
Cluster 4: Older, male, loyal customers

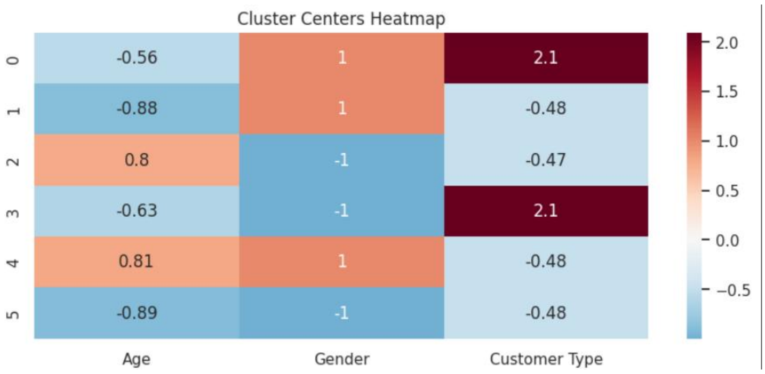
Cluster 5: Young, female, loyal customers

These profiles provide a clear segmentation of the customer base according to age, gender, and loyalty status, allowing for more targeted service strategies and personalized engagement.



```
K=2, Silhouette Score=0.4666
K=3, Silhouette Score=0.5243
K=4, Silhouette Score=0.5085
K=5, Silhouette Score=0.5424
K=6, Silhouette Score=0.5849
K=7, Silhouette Score=0.5740
K=8, Silhouette Score=0.5706
K=9, Silhouette Score=0.5688
K=10, Silhouette Score=0.5697

Recommended optimal K: 6 (Silhouette Score=0.5849)
```





	Age	Gender	Customer Type
0	-0.56	1	2.1
1	-0.88	1	-0.48
2	0.8	-1	-0.47
3	-0.63	-1	2.1
4	0.81	1	-0.48
5	-0.89	-1	-0.48

Travel Features: Class, Type of Travel, Flight Distance

From the cluster center heatmap, we observe some points on single attribute:

Class: Divided into three categories, perfectly matching the original distribution of the attribute.

Type of Travel: Divided into two categories, also consistent with the inherent distribution of data.

Flight Distance: After K-Means clustering, roughly grouped into three ranges: -0.61 to -0.32, 0.76 to 0.81, and a category containing the value 2.2.

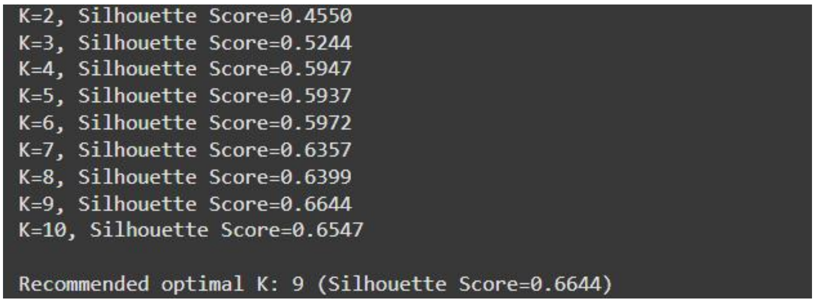
Combining these results, the users' trips can be classified into 9 distinct categories， and the accuracy can be proved by Silhouette Score = 0.6644.

Key Business Insights:

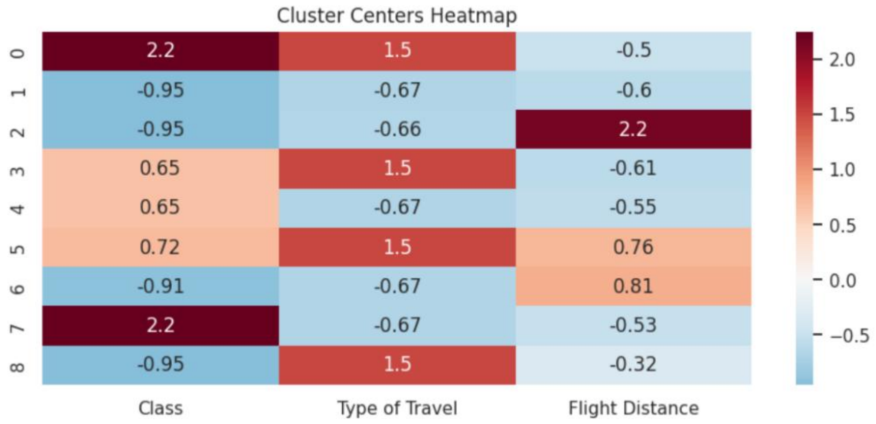
Differences Between Business and Personal Travel: Personal travel is mostly concentrated in short-distance trips, with three out of four relevant distributions falling in the short-distance category. In contrast, business travel is more evenly distributed across short, medium, and long distances.

Cabin Preferences: Business travelers tend to prefer economy class more frequently compared to other traveler segments.

These insights help in understanding travel behavior and can guide service optimization and marketing strategies targeted to different traveler types.



```
K=2, Silhouette Score=0.4550
K=3, Silhouette Score=0.5244
K=4, Silhouette Score=0.5947
K=5, Silhouette Score=0.5937
K=6, Silhouette Score=0.5972
K=7, Silhouette Score=0.6357
K=8, Silhouette Score=0.6399
K=9, Silhouette Score=0.6644
K=10, Silhouette Score=0.6547

Recommended optimal K: 9 (Silhouette Score=0.6644)
```

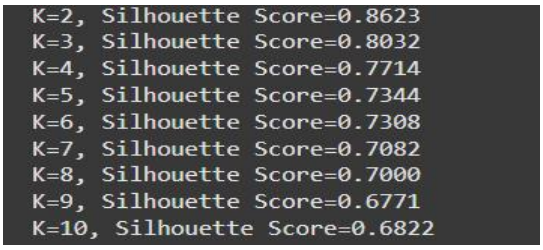


Some interesting attributes

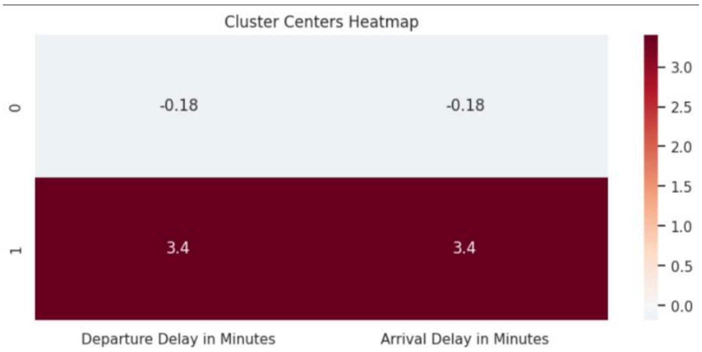
Objective Delay Metrics: Departure Delay in Minutes, Arrival Delay in Minutes

In our clustering analysis using Departure Delay in Minutes and Arrival Delay in Minutes, the results indicated that, within the tested range of K values, the silhouette coefficients remained consistently high and close to 1. This suggests that these two attributes exhibit a strong correlation. Moreover, in the cluster center heatmap, the boundaries between clusters are also clearly distinguishable.

These findings motivated us to conduct further outlier analysis, specifically to explore scenarios where Departure Delay in Minutes is significantly large while Arrival Delay in Minutes remains unexpectedly small.



```
  K=2, Silhouette Score=0.8623
  K=3, Silhouette Score=0.8032
  K=4, Silhouette Score=0.7714
  K=5, Silhouette Score=0.7344
  K=6, Silhouette Score=0.7308
  K=7, Silhouette Score=0.7082
  K=8, Silhouette Score=0.7000
  K=9, Silhouette Score=0.6771
  K=10, Silhouette Score=0.6822
```

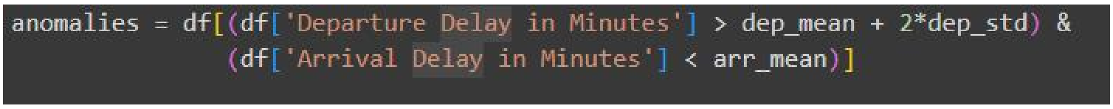


We define the following anomaly detection criteria:

Condition 1: A flight is considered an anomaly if its Departure Delay in Minutes is greater than the mean plus two standard deviations. This identifies flights with exceptionally long departure delays compared to the overall dataset.

Condition 2: At the same time, the Arrival Delay in Minutes must be lower than the overall mean arrival delay.

These two conditions show that although the departure was very late, the arrival time was still earlier than average.



```
anomalies = df[(df['Departure Delay in Minutes'] > dep_mean + 2*dep_std) &
               (df['Arrival Delay in Minutes'] < arr_mean)]
```

We printed the identified anomalies and visualized the distribution of the entire dataset across the two features (Departure Delay in Minutes and Arrival Delay in Minutes). The result, however, returned an empty DataFrame, and the visualization showed no anomalous points.

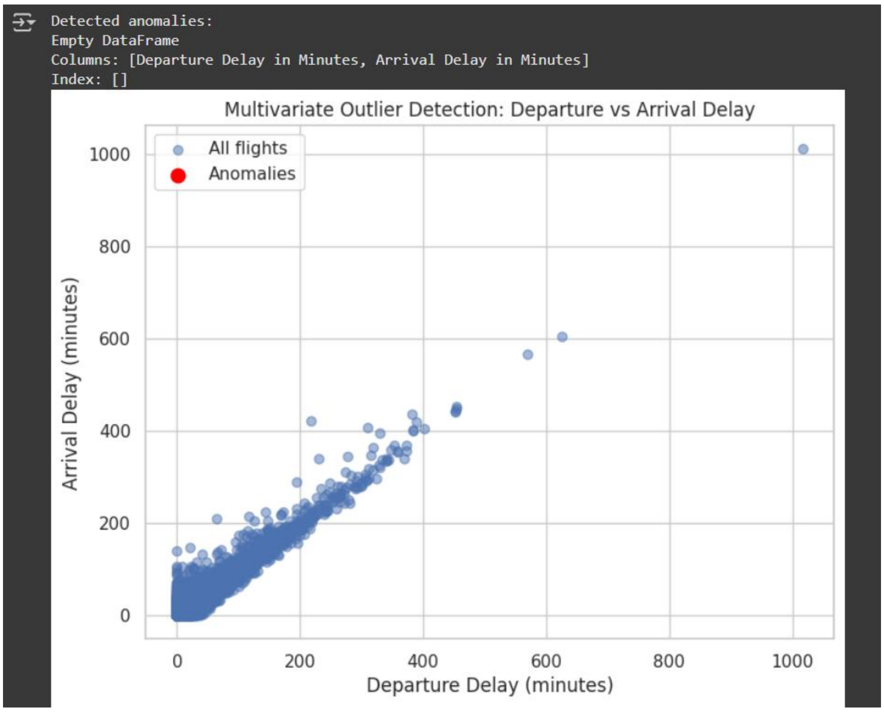
Even with more than 16,000 records in the dataset, no cases met the defined conditions. We further attempted to relax the anomaly criteria, but the results remained the same—no anomalies were detected under the adjusted thresholds.

This outcome highlights several important insights:

Strong Correlation Departure delays almost always translate into arrival delays, indicating a direct and consistent relationship between the two variables.

Absence of Extreme Outliers – No contradictory records were found, confirming the stability and reliability of these attributes.

Operational Implication – Once a departure delay occurs, airlines rarely recover the lost time during the flight, suggesting limited flexibility in compensating for schedule disruptions.



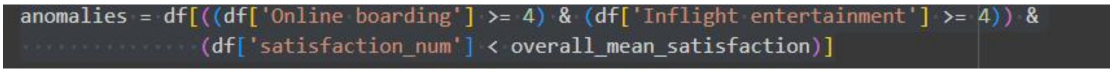
```
Detected anomalies:
Empty DataFrame
Columns: [Departure Delay in Minutes, Arrival Delay in Minutes]
Index: []
```

Multivariate Outlier Detection: Departure vs Arrival Delay
● All flights
● Anomalies
Arrival Delay (minutes)
0
200
400
600
800
1000
Departure Delay (minutes)
0
200
400
600
800
1000

Anomaly Detection Based on Online Boarding and Inflight Entertainment

Based on the previously generated Spearman correlation heatmap and hierarchical clustering results, we identified Online Boarding and Inflight Entertainment as the attributes most strongly correlated with passenger satisfaction. Therefore, we performed anomaly detection focusing on these two features in relation to satisfaction.

The anomaly condition was defined as follows: passengers who rated either Online Boarding ≥ 4 and Inflight Entertainment ≥ 4, but whose satisfaction score was below the global mean satisfaction.



```
anomalies = df[((df['Online boarding'] >= 4) & (df['Inflight entertainment'] >= 4)) &
               (df['satisfaction_num'] < overall_mean_satisfaction)]
```

To better interpret the results, we printed the indices of abnormal points. And after that, we used a Combined Mean Heatmap to visualize the overall distribution of mean satisfaction scores across the feature combinations, while Outlier Scatter Plot was employed specifically to highlight the detected anomalies. Our findings are as follows:

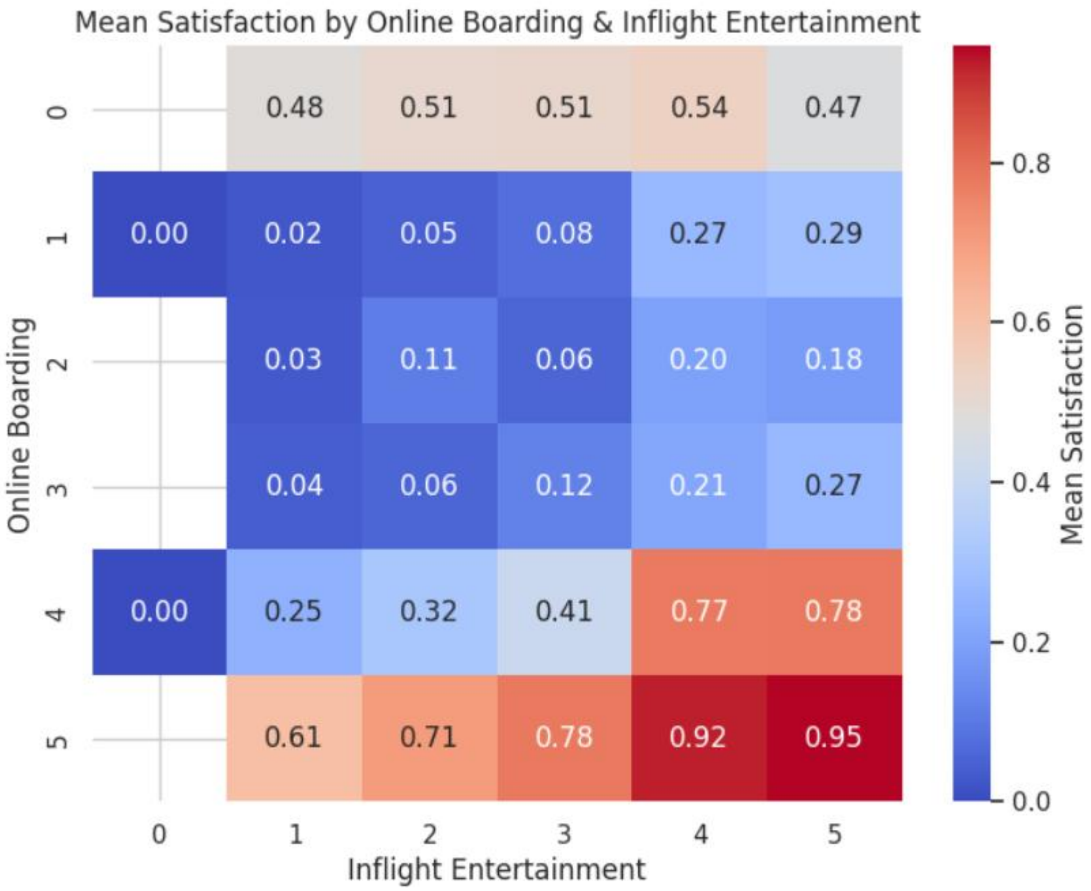
Combined Mean Heatmap

1. Online Boarding Rating Positively Correlates with Satisfaction
Analysis of the dataset indicates that online boarding ratings have a strong positive relationship with passenger satisfaction. When the online boarding rating is between 0 and 3, the average satisfaction remains low (below 0.3) and in some instances approaches zero. Conversely, when the rating reaches 5, satisfaction significantly improves, with the maximum value observed at 0.095. This suggests that the online boarding experience is a factor which can strongly influence overall passenger satisfaction.

2. Inflight Entertainment Has a Weaker Impact on Satisfaction
The effect of inflight entertainment on satisfaction is comparatively weaker. High online boarding ratings combined with higher inflight entertainment ratings can further enhance satisfaction. However, when online boarding ratings are low, even high inflight entertainment scores do not lead to high satisfaction levels. This indicates that inflight entertainment plays a supporting role in improving passenger satisfaction.

3. Interaction Between Online Boarding and Inflight Entertainment
The combined effect of improving both online boarding and inflight entertainment ratings results in the highest satisfaction levels. For example, when both ratings are at the maximum score of 5, the satisfaction score reaches its peak.

4. Anomalies: High Satisfaction Despite Low Online Boarding Ratings
An anomaly is observed in the first row of the heatmap (online boarding rating = 0), where inflight entertainment scores of 0–3 correspond to an average satisfaction that is surprisingly high (around 0.48–0.54).

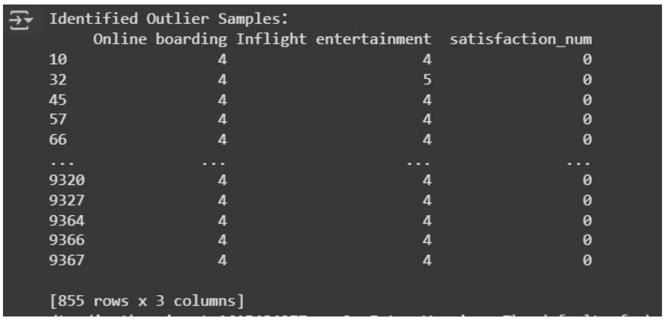


Outlier Scatter Plot

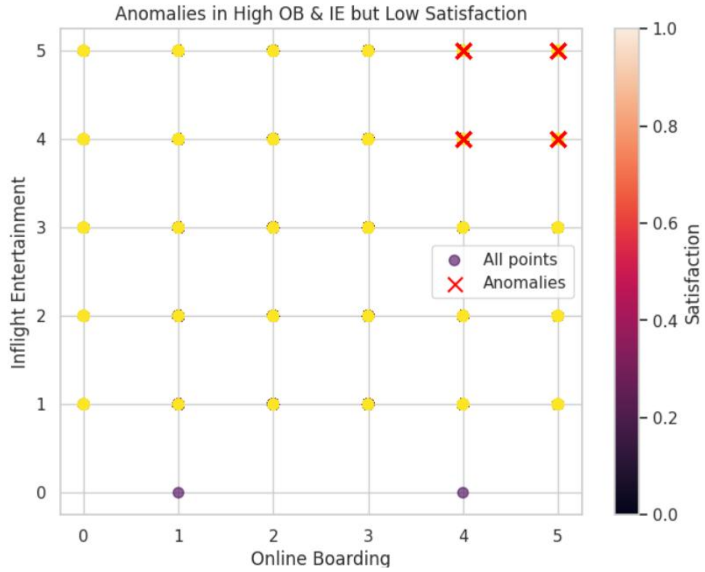
The detected anomalies are concentrated in the high-rating regions for Online Boarding and Inflight Entertainment: specifically, where both ratings are 4 or 5. Normally, these rating combinations should correspond to high satisfaction, yet these points exhibit unexpectedly low satisfaction.

These anomalies contradict the general trend, as Online Boarding and Inflight Entertainment are the two features most strongly correlated with overall satisfaction. This suggests that other factors, which are less correlated with satisfaction, may also influence the satisfaction in some ways.

From the analysis, the majority of samples follow the expected positive correlation between these ratings and satisfaction. Out of 16,624 records, only 855 were identified as anomalies, indicating that such deviations are relatively rare.



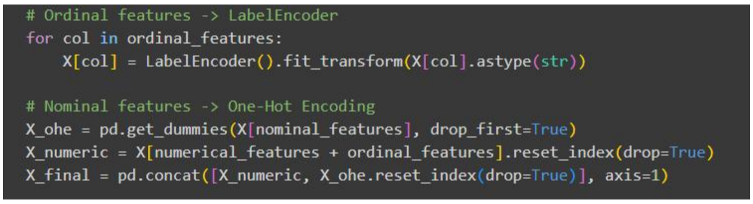
```
Identified Outlier Samples:
      Online boarding  Inflight entertainment  satisfaction_num
10                  4                       4                 0
32                  4                       5                 0
45                  4                       4                 0
57                  4                       4                 0
66                  4                       4                 0
...               ...                     ...               ...
9320                4                       4                 0
9327                4                       4                 0
9364                4                       4                 0
9366                4                       4                 0
9367                4                       4                 0

[855 rows x 3 columns]
```



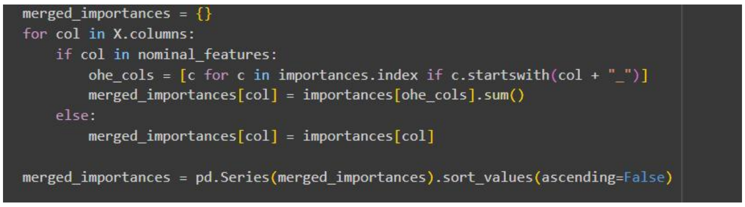
Random Forest Feature Importance Analysis for Passenger Satisfaction

In this modeling exercise, we first consolidated all available features to serve as input for training a Random Forest model. Nominal (categorical) variables in the dataset were preprocessed using one-hot encoding to ensure compatibility with the model.

After training, we ranked the importance of all features based on the Random Forest's inherent feature importance scores. Since one-hot encoding splits nominal variables into multiple binary columns, failing to merge them would scatter their importance across multiple features, making it difficult to interpret the overall contribution of the original variable. To address this, we aggregated the one-hot encoded features back into their original nominal variables, producing a clear and interpretable feature importance visualization. This provides a solid foundation for subsequent analysis.

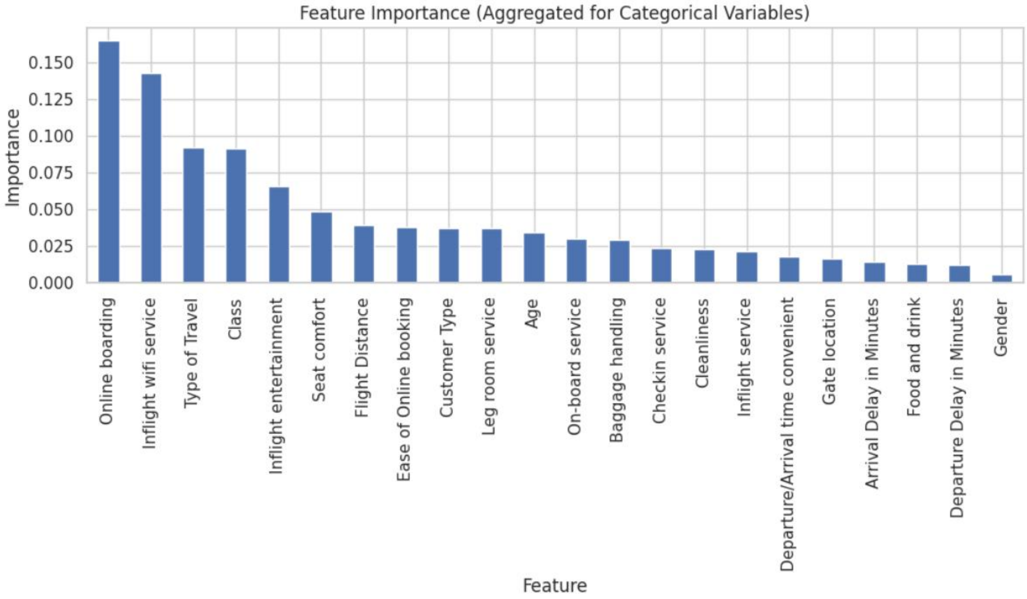


```python
# Ordinal features -> LabelEncoder
for col in ordinal_features:
    X[col] = LabelEncoder().fit_transform(X[col].astype(str))

# Nominal features -> One-Hot Encoding
X_ohe = pd.get_dummies(X[nominal_features], drop_first=True)
X_numeric = X[numerical_features + ordinal_features].reset_index(drop=True)
X_final = pd.concat([X_numeric, X_ohe.reset_index(drop=True)], axis=1)
```



```
merged_importances = {}
for col in X.columns:
    if col in nominal_features:
        ohe_cols = [c for c in importances.index if c.startswith(col + "_")]
        merged_importances[col] = importances[ohe_cols].sum()
    else:
        merged_importances[col] = importances[col]

merged_importances = pd.Series(merged_importances).sort_values(ascending=False)
```

The generated feature importance plot provides a clear and intuitive view of the influence of each feature on satisfaction across the dataset, making the relative contributions of different factors immediately apparent. Moreover, the patterns revealed by this plot are consistent with the earlier findings from the Spearman correlation heatmap and hierarchical clustering analysis, mutually validating and reinforcing the results. This further confirms the reliability and consistency of our analysis, offering a solid foundation for subsequent decision-making aimed at optimizing passenger satisfaction.



1B. Data preprocessing

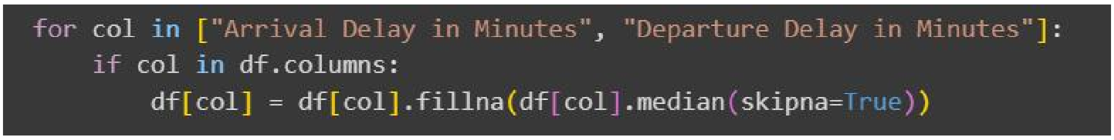
1. Binning techniques

In this part, we need to apply binning to smooth two delay time attributes:

1. Arrival Delay in Minutes
2. Departure Delay in Minutes

1.1 Missing Value Handling

Before binning, missing values need to be addressed. For both delay attributes, if missing values are present, they are filled using the median. Since these two attributes exhibit a long-tail distribution where the mean is easily influenced by extreme values, so we choose the median to offer a more robust representation of the typical value for most samples. And the following is how we carry on this step:



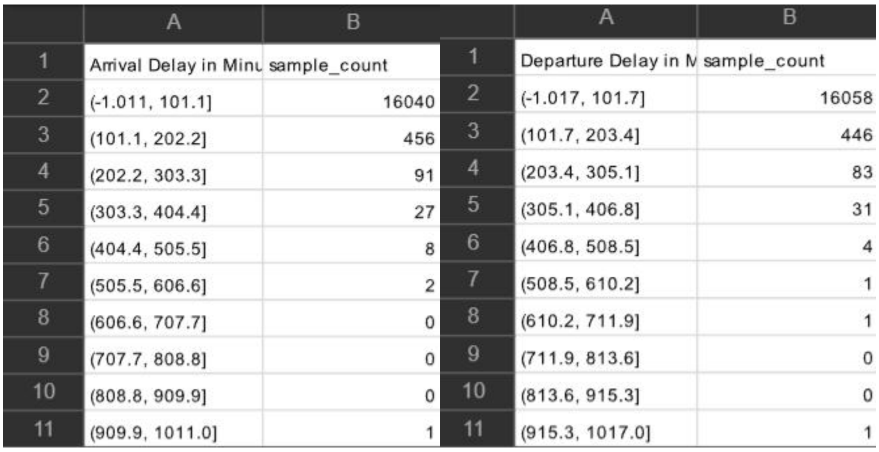
```
for col in ["Arrival Delay in Minutes", "Departure Delay in Minutes"]:
    if col in df.columns:
        df[col] = df[col].fillna(df[col].median(skipna=True))
```

1.2 Binning

1.2.1 Equi-width binning

Equi-width binning distributes all values evenly across intervals of equal width. Since the data for these two attributes is unevenly distributed, with a large number of zero values and a clear long-tail distribution (the maximum value even exceeds 1000), choosing too many bins would result in many empty bins, while too few bins would fail to reflect the data distribution well. Therefore, we chose 10 bins as an initial attempt.

We counted the frequency of each interval after binning, and the results are as follows:



	A	B		A	B
1	Arrival Delay in Minu	sample_count	1	Departure Delay in M	sample_count
2	(-1.011, 101.1]	16040	2	(-1.017, 101.7]	16058
3	(101.1, 202.2]	456	3	(101.7, 203.4]	446
4	(202.2, 303.3]	91	4	(203.4, 305.1]	83
5	(303.3, 404.4]	27	5	(305.1, 406.8]	31
6	(404.4, 505.5]	8	6	(406.8, 508.5]	4
7	(505.5, 606.6]	2	7	(508.5, 610.2]	1
8	(606.6, 707.7]	0	8	(610.2, 711.9]	1
9	(707.7, 808.8]	0	9	(711.9, 813.6]	0
10	(808.8, 909.9]	0	10	(813.6, 915.3]	0
11	(909.9, 1011.0]	1	11	(915.3, 1017.0]	1

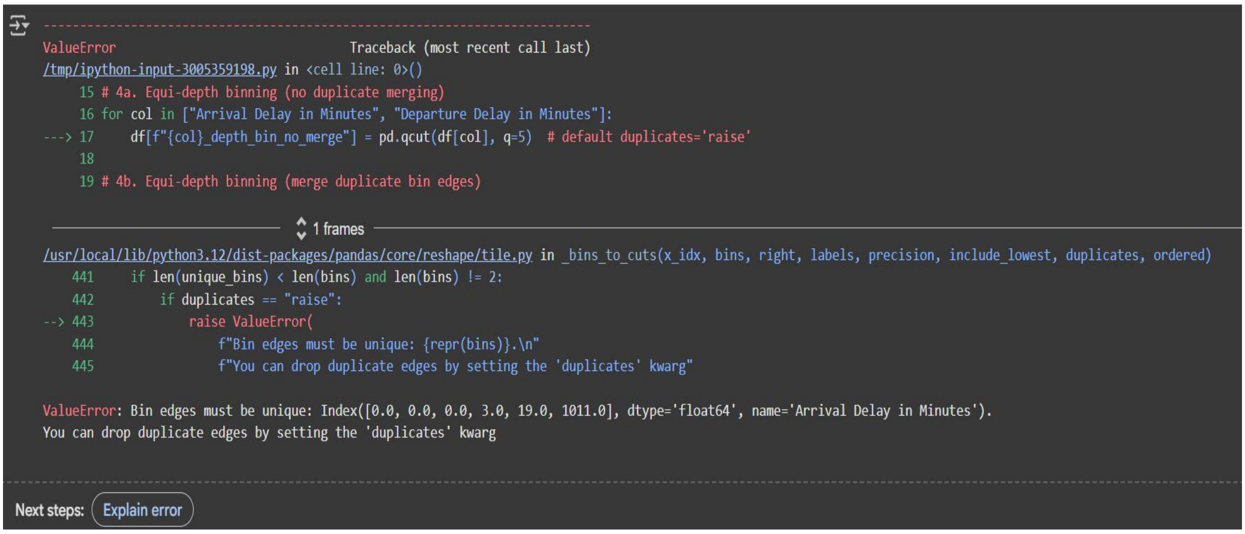
From the results of binning, we could draw some findings. First of all, the samples are highly concentrated in the first bin, accounting for the vast majority (16,040 samples). Secondly, as the delay time increases, the number of samples decreases rapidly, with the later bins containing very few or even zero samples. Finally, the long-tail distribution is evident, which validates the feasibility of using the median to fill in missing values.

In summary, applying equi-width binning to this type of distribution leads to a large concentration of data in the first interval, which is not conducive to data analysis.

1.2.2 Equi-depth binning

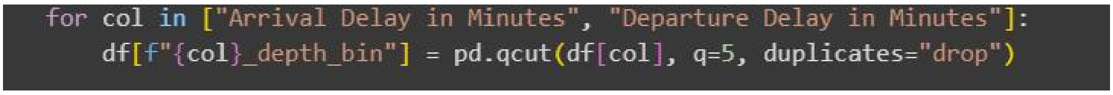
The definition of equal-depth binning is to divide the data based on quantiles, so that each bin contains approximately the same number of samples.

It is worth mentioning that in our first attempt, we did not merge duplicate quantile points, which resulted in the following error. The root cause was that there were a large number of delay values equal to 0, causing many quantile points to overlap, which is not allowed in pandas.



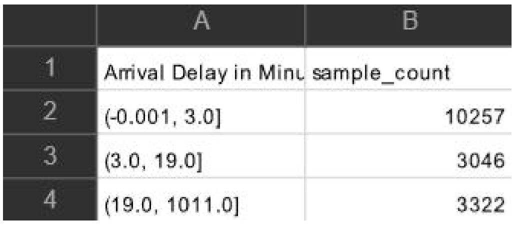
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipython-input-3005359198.py in <cell line: 0>()
     15 # 4a. Equi-depth binning (no duplicate merging)
     16 for col in ["Arrival Delay in Minutes", "Departure Delay in Minutes"]:
---> 17     df[f"{col}_depth_bin_no_merge"] = pd.qcut(df[col], q=5)  # default duplicates='raise'
     18 
     19 # 4b. Equi-depth binning (merge duplicate bin edges)

                                  1 frames

/usr/local/lib/python3.12/dist-packages/pandas/core/reshape/tile.py in _bins_to_cuts(x_idx, bins, right, labels, precision, include_lowest, duplicates, ordered)
    441     if len(unique_bins) < len(bins) and len(bins) != 2:
    442         if duplicates == "raise":
--> 443             raise ValueError(
    444                 f"Bin edges must be unique: {repr(bins)}.\n"
    445                 f"You can drop duplicate edges by setting the 'duplicates' kwarg"

ValueError: Bin edges must be unique: Index([0.0, 0.0, 0.0, 3.0, 19.0, 1011.0], dtype='float64', name='Arrival Delay in Minutes').
You can drop duplicate edges by setting the 'duplicates' kwarg
```

Therefore, in our second attempt, we used duplicates='drop' to automatically merge duplicate quantile points. We chose to create 5 equal-depth bins, but due to the automatic merging, the actual number of bins was fewer than defined.

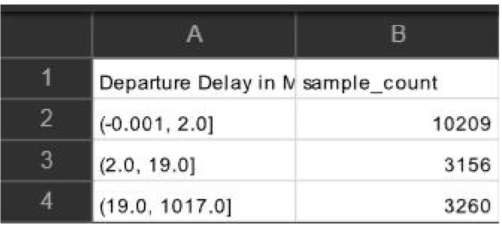


```
for col in ["Arrival Delay in Minutes", "Departure Delay in Minutes"]:
    df[f"{col}_depth_bin"] = pd.qcut(df[col], q=5, duplicates="drop")
```

To display the results, we counted the frequency of each bin, as shown in the figure below. From the results, we can observe that the first bin is approximately an integer multiple of bin 2 and bin 3, and that the sizes of bin 2 and bin 3 are roughly equal. This proves the accuracy of the accuracy of our binning.



	A	B
1	Arrival Delay in Minu	sample_count
2	(-0.001, 3.0]	10257
3	(3.0, 19.0]	3046
4	(19.0, 1011.0]	3322



	A	B
1	Departure Delay in M	sample_count
2	(-0.001, 2.0]	10209
3	(2.0, 19.0]	3156
4	(19.0, 1017.0]	3260

2. Normalization

In this dataset, we applied both Min–Max normalization and Z-Score standardization to the Flight Distance attribute.

Min–Max normalization linearly maps the original values to a fixed range [0,1], using the formula:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where x is the original value, x_{min} is the minimum value while x_{max} is the maximum value of the attribute.

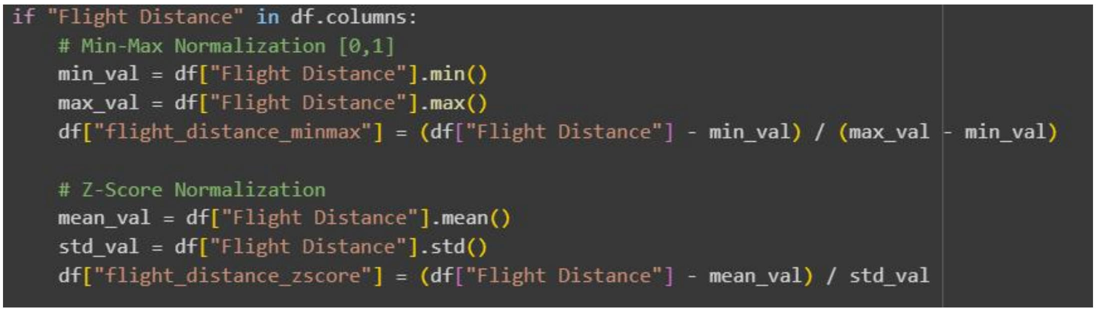
Z-Score standardization transforms the data to have a mean of 0 and a standard deviation of 1 through centering and scaling, using the formula:

$$x' = \frac{x - \mu}{\sigma}$$

where μ is the mean of the attribute, σ is the standard deviation.

The Flight Distance column has the following statistics: minimum = 31, maximum = 4963, median = 834, 25th percentile = 413, 75th percentile = 1734, mean = 1185. Most flights are concentrated in the 400–1700 km range, with a few long-distance flights forming a noticeable long-tail distribution.

Our codes are as follows:



```python
if "Flight Distance" in df.columns:
    # Min-Max Normalization [0,1]
    min_val = df["Flight Distance"].min()
    max_val = df["Flight Distance"].max()
    df["flight_distance_minmax"] = (df["Flight Distance"] - min_val) / (max_val - min_val)

    # Z-Score Normalization
    mean_val = df["Flight Distance"].mean()
    std_val = df["Flight Distance"].std()
    df["flight_distance_zscore"] = (df["Flight Distance"] - mean_val) / std_val
```

Min-Max:

This method compresses all values into the range [0,1], which is beneficial for models and analyses that require a fixed input range. However, since the first 75% of the data is concentrated roughly between 0 and 0.3, and the distribution has a noticeable right-skewed long tail, the resulting normalized data is highly unevenly distributed, which may affect subsequent analysis and modeling.

Z-Score：

This method standardizes the data by centering it around the mean with unit variance. It makes attributes with different scales comparable, but since both the mean and standard deviation are sensitive to outliers, the method itself is not robust to extreme values. As a result, extreme values will produce very large positive or negative Z-scores. Unlike Min-Max normalization, the results are not restricted to a fixed range (e.g., [0,1]), which may be unsuitable in scenarios where bounded input values are required.

3. Discretization

To better understand the distribution of age in the dataset, we discretised the continuous Age attribute into categorical bins using given age ranges. The categories are defined as follows:

Young: ages 21 and younger

Early Adulthood: ages 22–34

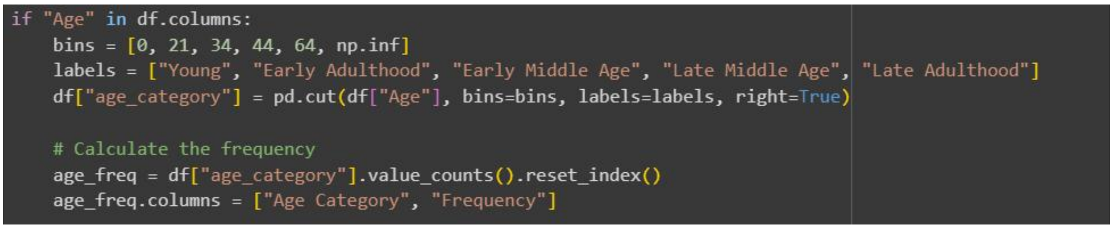
Early Middle Age: ages 35–44

Late Middle Age: ages 45–64

Late Adulthood: ages 65 and older

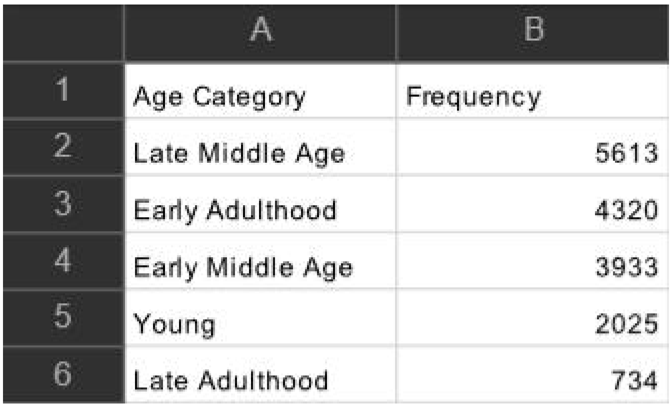
This discretisation was implemented using the pd.cut() function in Python, which allows for binning based on specified intervals. The “right=True” parameter ensures that the bin intervals are right-inclusive, and the labels parameter assigns meaningful names to each bin.

Our codes are as follows:



```
if "Age" in df.columns:
    bins = [0, 21, 34, 44, 64, np.inf]
    labels = ["Young", "Early Adulthood", "Early Middle Age", "Late Middle Age", "Late Adulthood"]
    df["age_category"] = pd.cut(df["Age"], bins=bins, labels=labels, right=True)

    # Calculate the frequency
    age_freq = df["age_category"].value_counts().reset_index()
    age_freq.columns = ["Age Category", "Frequency"]
```

The frequency distribution of each age category in the dataset is as follows:



	A	B
1	Age Category	Frequency
2	Late Middle Age	5613
3	Early Adulthood	4320
4	Early Middle Age	3933
5	Young	2025
6	Late Adulthood	734

This categorisation helps simplify analysis and enables clearer insights into age-related patterns in the data.

4. Satisfaction Attribute Binarisation

To simplify analysis and enable machine learning models to process categorical data, we applied binarisation to the satisfication attribute. This technique converts categorical values into binary numerical values, making them easier to interpret and use in downstream tasks.

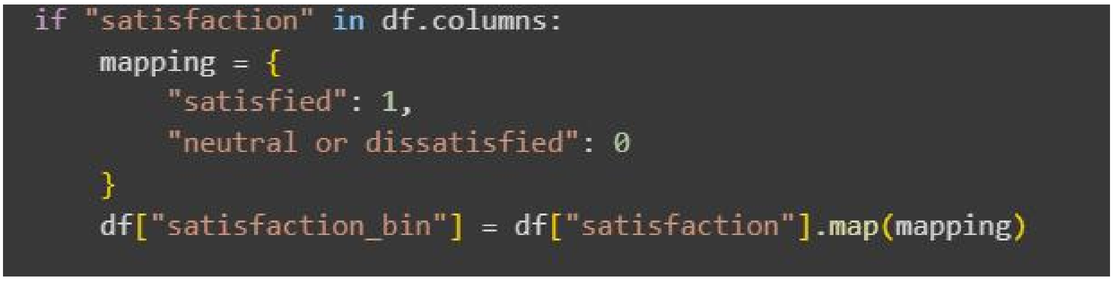
In our dataset, the satisfication variable contains two categories: "satisfied" and "neutral or dissatisfied"

We mapped these categories to binary values using the following logic:

"satisfied" → 1

"neutral or dissatisfied" → 0

This transformation was implemented using the map() function in Python, as shown below:



```
if "satisfaction" in df.columns:
    mapping = {
        "satisfied": 1,
        "neutral or dissatisfied": 0
    }
    df["satisfaction_bin"] = df["satisfaction"].map(mapping)
```

The resulting binary values were stored in a new column named satisfication_bin. This column can now be used for statistical analysis, visualisation, or as input to machine learning models.

1C. Summary of Findings

The analysis reveals several key insights into passenger satisfaction and travel behavior. Overall, 56.7% of passengers reported being neutral or dissatisfied, indicating substantial room for improvement in customer experience.

The sample composition is relatively balanced by gender, but highly skewed toward loyal customers (81.4%) and business travelers (69.3%), with nearly half traveling in Business Class. This composition may bias satisfaction levels toward frequent and premium customers.

Numeric attributes display distinct patterns: Age is centered in adulthood with a slightly right-skewed distribution; Flight Distance shows a long-tailed distribution dominated by short and medium-haul flights; both Departure and Arrival Delays are heavily right-skewed, with most flights on time but a few extreme delays inflating the averages.

Correlation and clustering analysis identify meaningful service dimensions. Cleanliness, Food and Drink, Seat Comfort, and Inflight Entertainment form a strongly correlated group representing in-flight service quality. Ease of Online Booking and Inflight WiFi Service show an unexpectedly strong correlation that warrants further investigation. In contrast, attributes such as Gate Location and Departure/Arrival Time Convenience show negligible influence on satisfaction. Hierarchical clustering validates these relationships and highlights opportunities for feature reduction.

K-Means clustering uncovers distinct passenger profiles: six demographic-based groups and nine travel-based categories. Business travelers demonstrate more balanced distance distributions, while personal travel is concentrated in short-haul flights. Cabin preferences also vary, with business travelers showing a surprising preference for Economy Class in some cases.

Delay analysis confirms a strong dependency between departure and arrival delays, with no evidence of recovery during flights. This highlights an operational limitation in schedule management.

Finally, feature importance analysis indicates that Online Boarding and Inflight Entertainment are the strongest drivers of satisfaction, consistent with correlation-based findings. While most passengers follow this pattern, anomalies exist where high ratings in these areas do not align with high satisfaction, suggesting the presence of additional hidden factors.

These findings provide actionable insights into key service drivers, operational limitations, and customer segmentation, offering a foundation for targeted service improvements and strategic decision-making.

Based on the conclusions above, it is recommended to prioritise improvements in Online Boarding and Inflight Entertainment, as these are the strongest drivers of passenger satisfaction. Additionally, enhancing flight delay management through better scheduling, increased buffer times, and operational adjustments can mitigate the negative impact of departure delays on arrivals. Implementing these measures should lead to higher overall satisfaction, particularly among loyal and business passengers.