Airline Passenger Satisfaction Prediction: A Machine Learning Approach to Service Quality Assessment

Mikhail Bocharov
bocharov.md@edu.spbstu.ru

Tamara Goncharova
goncharova.tg@edu.spbstu.ru

Additional collaborators:

Holger Espinola
espinola_rh@edu.spbstu.ru

Vladimir Zaborovsky
vladimir.zaborovsky@spbstu.ru

Vadim Pak
pak_vg@spbstu.ru

Abstract

Airline companies focus on providing the best service to attract and retain customers, but predicting the passenger satisfaction remains diffucult due to the complex interplay of flight and service factors such as delay, travel distance, seat comfort, etc. We propose to utilize the advancements in Machine Learning field by implementing a model that predicts the satisfaction outcome (satisfied or not) for a passenger considering their feedback, flight and service conditions. In this study, we evaluated 2 machine learning algorithms: k-Nearest Neighbours (k-NN) and Random Forest (RF) classifiers using cross-validation on a dataset with 130,000 passenger surveys, with Random Forest giving the superior results, achieving an f1-score of 96.3% on the test data. By analyzing feature importance scores from the Random Forest model, we identify online boarding and inflight Wi-Fi services as the key determinants of passenger satisfaction, offering data-driven guidance for targeted service improvements for airline companies.

1 Introduction

The airline industry is a highly competitive landscape where customer satisfaction is a crucial factor of success. The level of satisfaction is influenced by the quality of operational factors (e.g., flight delays), service quality (e.g., cabin cleanliness), and passenger demographics (e.g., age, gender). Airline companies gather a vast amount of data through surveys that cover various aspects of the flight experience, such as convenience of departure time, leg room, etc. Key satisfaction factors can be identified only when passenger feedback is correctly linked to operational data.

However, the prediction of passenger satisfaction can be a challenge due to the complex interconnected nature of factors forming the passenger's attitude. Traditional methods of analyzing customer feedback can fall short in capturing nonlinear relationships and interactions among these variables.

Machine learning (ML) approach offers a robust mechanism to address such challenges by enabling data-driven classification of passenger satisfaction. By leveraging mostly passenger's feedback the model aims to identify patterns that drive satisfaction (or dissatisfaction). This paper details the implementation and evaluation of Random Forest and k-Nearest Neighbours models to classify whether a passenger will be satisfied based on their feedback, flight details and passenger attributes. Furthermore, we employ the best-performing model (Random Forest) to identify the highest-impact factors influencing satisfaction outcomes, highlighting key areas where service improvements may yield the greatest impact on satisfaction.

2 Methodology

Our research follows a structured machine learning workflow (Figure 1). The pipeline comprises eight key stages:

1. Data Acquisition: Downloading and inspecting the dataset.

2. Data Cleaning: Addressing missing values, duplicate data and outliers.

3. Exploratory Data Analysis (EDA): Visualizing feature distributions and correlations.

4. Data Standardization: Normalizing numerical features and encoding categorical variables.

5. Model Building: Configuring and implementing k-NN and Random Forest.

6. Training & Testing: Cross-validating models and tuning hyperparameters.

7. Model Evaluation & Analysis:

    - Metrics calculation (accuracy, f1-score, etc.) for k-NN and RF.

    - Models perfomance comparison.

    - Concluding models suitability.

8. Insights Generation: Identifying key drivers of passenger satisfaction using trained models.

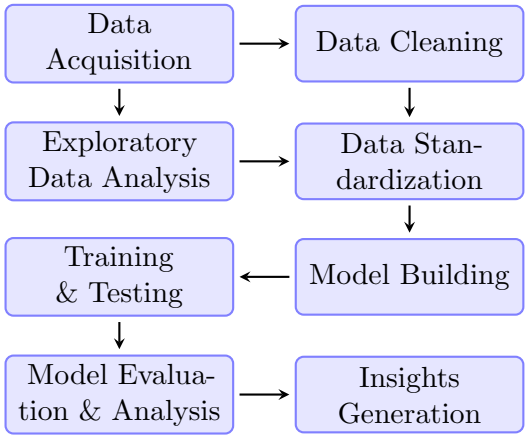

Figure 1: Research Workflow for passenger satisfaction prediction. Arrows indicate sequential dependencies.

2.1 Data acquisition and overview

This study utilizes the "Airline Passenger Satisfaction" dataset sourced from Kaggle (Mahal, 2020). The dataset includes 25 features, combining quantitative metrics (e.g., numerical ratings) and categorical attributes (e.g., customer type, travel class), with the binary target variable reflecting overall passenger satisfaction (satisfied or dissatisfied/neutral). Features capture both operational flight details (e.g., flight distance, class, departure delay) and passenger survey responses evaluating service satisfaction (e.g., seat comfort, in-flight service, cleanliness). Dataset contains 103904 samples and has a data footprint of 15MB. Information about the features can be found in the Table 5 (Appendix A).

2.2 Data preprocessng

To train high-performance models the data must undergo preprocessing. This essential stage ensures data consistency, improved model accuracy, and reduced overfitting by addressing noises, biases and data redundancy. Preprocessing consists of data cleaning, exploratory data analysis (EDA), dataset splitting and data standardization.

2.2.1 Data cleaning

The preprocessing starts by removing samples with missing values, as they constitute a small fraction ($\sim$0.3%) of the dataset. We also eliminate duplicate samples and irrelevant features, such as "ID" and "unnamed column" (row index). These steps ensure the models learn from a prepresentative dataset, focusing on meaningful connections rather than noise or duplicated data.

We also detect outliers to mitigate the potential data skewness using quantile method. We identify values that fall outside the interval between the 2.275% and 97.775% percentile range (capturing 95% of the data). Our analysis reveals that no feature exceeds a 5% threshold, indicating negligible impact from extreme values.

2.2.2 Exploratory Data Analysis

We explore numerical and categorical features to analyze data distributions. This involves plotting frequency histograms and pie charts. They are shown in Figure 6 (Appendix B) and Figure 7 (Appendix B) respectively.

Following initial data cleaning, we examine feature relationships by constructing a correlation matrix. It is depicted in Figure 5 (Appendix B). The analysis reveals a very high correlation (0.97) between delayed arrival and departure times. To mitigate multicollinearity, we remove one of these features (departure delay in this case).

2.2.3 Dataset splitting

After initial preprocessing, we split data into training (80%) and testing (20%) subsets. For our dataset, a 20% test set provides sufficient samples to reliably evaluate model performance while

preserving training data volume.

2.2.4 Data Standardization

To effectively train a model it is necessary to re-scale data in order to avoid disparities that can distort model training. We address this by using Standard Scaler and Robust Scaler to normalize numerical features.

Standard Scaler is applied to features approximating a Gaussian distribution (e.g. age, flight distance). It scales features to have a mean of 0 and a standard deviation of 1, ensuring consistent feature weighting. Scaled value (SV) is computed using the formula (Kherdekar & Naik, 2024):

$$SV = \frac{X - \mu}{\sigma}$$

where

- X is the current value;
- μ is the mean of X;
- σ is the standard deviaton of X.

Robust Scaler is used for features that have a skewed distribution (e.g., inflight Wi-Fi service, online boarding). It scales values using quartiles by the formula:

$$SV = \frac{X - Q_2}{IQR}$$

where

- Q_1, Q_2, Q_3 are the 25th, 50th(median) and 75th quartiles respectively;
- $IQR = Q_3 - Q_1$ is the interquartile range representing the spread of the central 50% of the data.

Categorical features (e.g., type of travel, customer type) are converted into numeric format using one-hot encoding. This method creates a binary (0/1) column for each class of the categorical feature, such that exactly one column is active (1). One-hot encoding avoids the false ordering relationships that occur with integer encoding (Brownlee, 2020).

2.3 Machine Learning Algorithms

For our binary classification task, we implement and compare 2 models: k-Nearest Neighbours and Random Forest.

2.3.1 k-Nearest Neighbours

The k-Nearest Neighbors (k-NN) algorithm classifies data samples by identifying the most frequent class among the **k** closest neighbour samples in a multi-dimensional feature space. Unlike parametric models (e.g., linear algorithms) k-NN operates without prior assumptions about underlying data distributions, enabling it to adapt organically to complex patterns. (Cover & Hart, 1967).

Given the dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ with N samples and d features, where $x_i \in R^d$ are feature vectors and $y_i \in \{1, 2.., K\}$ are corresponing class labels, the distance between a query (test) sample x_{test} and a training sample x_i can be calculated using the Minkowski Distance (Cunningham & Delany, 2021):

$$dist(x_{test}, x_i) = (\sum_{j=1}^{d} |x_{test,j} - x_{i,j}|^p)^{\frac{1}{p}}$$

where $x_{test,j}$ and $x_{i,j}$ represent the j-th feature of corresponding sample. p defines the norm:

- $p = 1$: Manhattan distance (L1);
- $p = 2$: Euclidean distance (L2);

After calculating distances we choose k nearest neighbours and assign the majority class label to the test point x_{test}. Closer neighbors can receive higher weights, depending on the chosen hyperparameter. In this study, we evaluate both uniform

and distance-based weighting schemas using grid search cross-validation. The distance-based weighting achieved statistically superior cross-validation scores, suggesting that closer neighbors, which reflect more similar passenger experiences, are more informative for predicting satisfaction.

The suitabilty of k-NN model is based on two factors:

- Local Similarity Assumption: passenger satisfaction is often influenced by combinations of closely related factors (e.g., class + seat comfort). k-NN leverages feature similarity to identify passengers with comparable experiences.

- Handling Non-Linearity: Unlike linear models, k-NN makes no assumptions about data distribution, making it robust to complex interactions between variables, which are likely to exist on the dataset with 24 features.

Since k-NN is a query-time bruteforce algorithm, it comes with high computational costs and is not recommended for use with big datasets. However, some optimization can be done in order to reduce the total number of required compares (Beygelzimer et al., 2006). In this study, we use k-d trees and ball trees to speed up neighbour searches.

For hyperparameter tuning we use 5-fold GridSearchCV which systematically evaluates (Sohil et al., 2021, pp. 181–185) combinations of k, distance metrics, neighbor-search algorithms and neighbour weightings. The results are shown in the Table 1.

Hyperparameter	**Tested values**	**Optimal value**
Neighbours	3, 5, 7	7
Neigbour weighting	uniform, distance-based	distance-based
Neighbour search algorithm	ball tree, k-d tree, brute force	ball tree
Metric (norm)	manhattan (L1), euclidean (L2)	manhattan (L1)

Table 1: k-NN hyperparameters

2.3.2 Random Forest

Random Forest is an ensemble learning method that aggregates predictions from multiple decision trees to reduce overfitting and improve generalization. The algorithm constructs a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, 2, ..., T\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors. Each tree casts a unit vote for the most popular class at input $\mathbf{x}$ (Breiman, 2001).

When building trees algorithm's goal is to maximize the reduction in so-called Gini-impurity (G) to make tree node as clear as possible (Louppe, 2015) .

Let's assume we have K classes $(k_1, ..., k_K)$:

$$G = 1 - \sum_{c=1}^{K} p_c^2$$

where

- p_c is a fraction of samples in the tree node belonging to class c;
- If all samples belong to one class (pure node), $G = 0$;
- If samples are evenly mixed (impure node), G approaches 1.

As was said before, decision tree aims to increase clarity for each split (Louppe, 2015, pp. 51–53) by

maximizing ΔG calculated using the formula:

$$\Delta G = G_{parent} - \left(\frac{N_{left}}{N_{parent}} G_{left} + \frac{N_{right}}{N_{parent}} G_{right} \right)$$

where

- G_{parent}: Gini-impurity of parent node before split;
- N_{parent}: total number of samples in the parent node;
- $N_{left}(N_{right})$: number of samples in the left(right) child node after splitting;
- $G_{left}(G_{right})$: Gini-impurity of the left(right) node after splitting;

Given an ensemble of T classifiers (trees) $h_1(\mathbf{x}), ..., h_T(\mathbf{x})$ the final prediction is aggregated as:

$$\hat{y} = \arg \max_{c \in \{1,...,K\}} \sum_{t=1}^{T} I(h_t(\mathbf{x}_{test}) = c)$$

where

- $h_t(x_{test})$ is the predicted class from the t-th tree for the test sample x_{test};
- $\arg\max_c$ selects the class c with the highest vote count;
- I$(\cdot)$ is the indicator function.

Advantages of Random Forest include:

- Feature Robustness: The dataset contains 24 features of mixed-type (e.g., numerical: flight distance; ordinal: service ratings; categorical: travel class). Random Forest inherently handles these without extensive preprocessing.
- Non-Linear Decision Boundaries: Decision trees in the ensemble split data based on thresholds (e.g., "departure delay ≥ 30 mins"), effectively modeling rules that reflect real-world passenger decision-making (e.g., dissatisfaction triggered by long delays).
- Resistance to overfitting: By aggregating predictions from multiple decorrelated decision trees – each trained on bootstrapped data with random feature subsets – the chances of overfitting are relatively small compared to other models (Sohil et al., 2021, pp. 186–193).

A notable drawback of RandomForest is the large size of the saved model file, which arises from storing the structure of hundreds of decision trees (including split rules, node thresholds, and feature indices). While this does not impact training or prediction accuracy, it may become a serious problem when deploying such models in production (e.g., in mobile applications).

The defined hyperparameters for RandomForest using GridSearch cross-validation are shown in the Table 2:

Hyperparameter	**Tested values**	**Optimal value**
Number of estimators	10, 50, 100	100
Maximum tree depth	10, 20, None	None
Min samples to split a node	2, 5, 10	2
Min samples to form a leaf	1, 2, 4	1
Features considered when finding the best split	$\sqrt{d}$, $\log_2 d$	$\sqrt{d}$

Table 2: Random forest hyperparameters

2.4 Infrastructure

The models were developed on a system with the following specifications:

- CPU: Intel(R) Core(TM) i5-7200U;
- RAM: 8 GB

We used Python 3.12.9 for this project. The software environment for machine learning:

- Data Processing libraries - Pandas 2.2.3, NumPy 2.2.0, Matplotlib 3.9.2, Seaborn

- Machine Learning library - Scikit-learn 1.6
- Development Environment - Jupyter Notebook.

3 Results

3.1 Model evaluation and comparison

The training time for each model: k-NN took 4,745 seconds (∼1.3 h) to train, while Random Forest completed training in 2,141 seconds (∼36 min), reflecting k-NN's higher computational demand from intensive distance calculations caused by the dataset size.

The main method for assessing the quality of models is the calculation of metrics that help identify their advantages and disadvantages. The following metrics were used in this study to evaluate the quality of classifiers: accuracy, precision, recall, specificity and F1-Score. The resulting metrics for k-NN and Random Forest are shown in Table 3 and Table 4 respectively.

	Accuracy	Precision	Recall	Specificity	f1-score
Mean-metrics	0.93	0.932	0.926	0.926	0.928

Table 3: Evaluation Results for the k-NN Classifier

	Accuracy	Precision	Recall	Specificity	f1-score
Mean-metrics	0.964	0.965	0.962	0.961	0.963

Table 4: Evaluation Results for the RF Classifier

The comparison of models' metrics is shown in Figure 2.

Both models' weights were saved in .pkl format for persistence. The k-NN model occupies approximately 16.4 MB of disk space, while the Random Forest requires up to 68.7 MB.

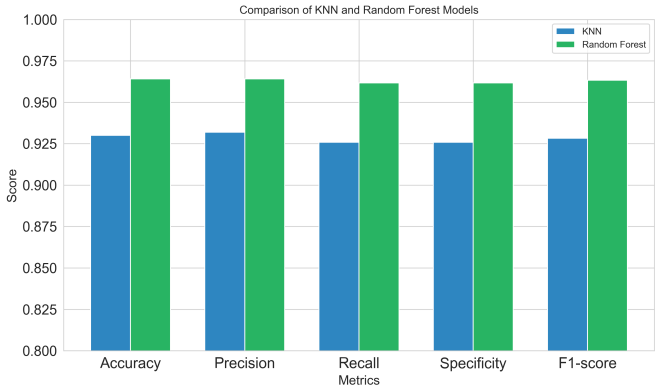

Figure 2: Comparative Performance of k-NN and RF classifiers

Summing up, 4 remarkable points of discussion about models evaluation:

1. Random Forest performs slightly better across all metrics, 0.5% improvement on average.

2. Random Forest is significantly faster to train, (2141 sec. vs 3791 sec. for k-NN), making it preferable for tasks with large amounts of samples.

3. The Random Forest model storage footprint (68.7 MB) is four times larger than the k-NN model (16.4 MB), reflecting a trade-off between performance and practicality. For applications prioritizing storage efficiency or deployment in resource-constrained environments (e.g., mobile applications), k-NN offers a lightweight alternative for a small lose in predictive power.

4. Both models demonstrate comparable and stable values in all metrics, indicating good generalization to test data. The reason behind similarity lies in the close nature of so-called "neighbourhood" algorithms (Lin & Jeon, 2006).

3.2 Feature insghts

Considering RF superior performance, we utilize it to find out which services or flight details affect the passengers' overall satisfaction level the most. Top 10 most influential factors by Gini index of importance are shown in Figure 3.

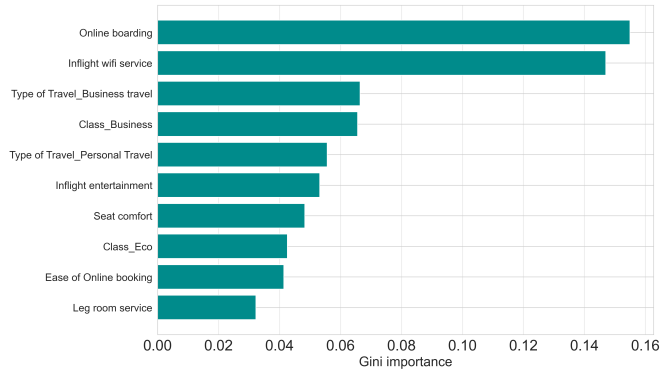

Figure 3: Random Forest measurement of features importance

As revealed by the analysis, online boarding and inflight WiFi service are the highest impact factors driving passenger satisfaction. These findings show the growing importance of seamless digital experiences in air travel, suggesting that airline companies prioritizing these areas are likely to increase the expected satisfaction level of customers.

We also detect the least impactful features. They are shown in Figure 4.

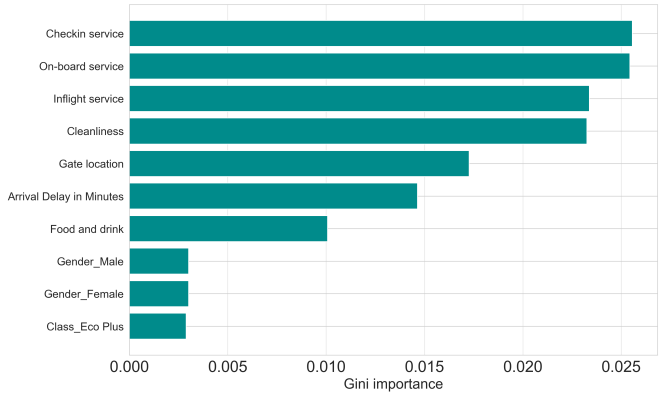

Figure 4: Factors with lowest feature importance

Many other services (e.g., on-board service, inglight service) have much less impact on passengers in terms of overall satisfaction. This analysis data can be used in order to identify the high-cost, low-impact services for optimizing operational cost or rearranging resources to more impactful areas (e.g., wifi service).

Conclusion and Future Work

In this study we implement and evaluatek-Nearest Neighbours (k-NN) and Random Forest classifiers to predict airline passenger satisfaction.

While both Random Forest (RF) and k-NN achieve strong performance (96% F1-score), RF demonstrates slight superiority (0.5% increase across all metrics) alongside significantly faster training times (2,141s vs. 4,745s for k-NN). However, practical deployment introduces trade-offs: RF's large serialized size (4 times larger than k-NN's) requires more disk space for deployment, which poses a significant challenge for resource-constrained systems, whereas k-NN's training complexity hinders scalability. This highlights a critical engineering balance between accuracy, latency, and resource constraints.

The analysis of models identifies online boarding

experience and inflight WiFi service quality as by far the foremost drivers of passenger satisfaction. These findings underscore the growing importance of digital convenience in air travel. Airlines can leverage these insights to strategically prioritize high-impact services while reallocating resources away from less influential offerings (e.g., food and drink, inflight service). Focusing on delivering exceptional performance in these critical areas can be used to:

- Enhance core offerings through digital boarding platforms with fast and reliable connectivity, directly addressing modern traveler expectations. This will ultimately boost customer retention and attract new travelers seeking for better service.
- Optimize operational budgets by deprioritizing underperforming services.
- Develop systems for high-impact services that can be monetized, such as tiered WiFi subscriptions (e.g., basic vs. high-speed streaming plans).

Considering these trade-offs of each model, future studies could explore gradient boosting (e.g, XGBoost, LightGBM) models or neural networks in order to define the optimal method.

References

Beygelzimer, A., Kakade, S., & Langford, J. (2006). Cover trees for nearest neighbor, 97–104. https://doi.org/10.1145/1143844.1143857

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 1–10. https://doi.org/10.1023/A:1010933404324

Brownlee, J. (2020). *Data preparation for machine learning: Data cleaning, feature selection, and data transforms in python*. Machine Learning Mastery.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Cunningham, P., & Delany, S. J. (2021). K-nearest neighbour classifiers - a tutorial. *ACM Computing Surveys*, *54*(6), 128:1–128:4. https://doi.org/10.1145/3459665

Kherdekar, V. A., & Naik, S. (2024). Scaling max absolute scaling. *Smart Trends in Computing and Communications: Proceedings of SmartCom 2024*, *Volume 5*, 110.

Lin, Y., & Jeon, Y. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, *101*(474), 578–590. https://doi.org/10.1198/016214505000001230

Louppe, G. (2015). Understanding random forests: From theory to practice, 40–45, 51–53. https://doi.org/https://doi.org/10.48550/arXiv.1407.7502

Mahal, T. J. (2020). Airline passenger satisfaction. https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

Sohil, F., Sohail, M., & Shabbir, J. (2021, September). *An introduction to statistical learning with applications in r* (Vol. 6). https://doi.org/10.1080/24754269.2021.1980261

Appendices

Appendix A

Table 5: Dataset features

Feature	**Description and possible values**
Unnamed Column	Row Number (integer value)
ID	Flight ID (integer value)
Gender	Gender of the passenger (Female, Male)
Customer Type	Type of customer (Loyal, Disloyal)
Age	Actual age of the passenger (integer value)
Type of Travel	Purpose of travel (Personal, Business)
Class	Travel class (Business, Eco, Eco Plus)
Flight Distance	Flight distance in miles
Inflight Wifi Service	Satisfaction level (0-5)
Time Convenience	Satisfaction with departure/arrival timing (0–5)
Ease of Online Booking	Satisfaction with online booking (0–5)
Gate Location	Satisfaction with gate accessibility (0–5)
Food and Drink	Satisfaction with meal quality (0–5)
Online Boarding	Satisfaction with online boarding process (0–5)
Seat Comfort	Satisfaction with seat ergonomics (0–5)
Inflight Entertainment	Satisfaction with entertainment options (0–5)
On-board Service	Satisfaction with cabin crew service (0–5)
Leg Room Service	Satisfaction with legroom space (0–5)
Baggage Handling	Satisfaction with baggage handling (0–5)
Check-in Service	Satisfaction with check-in process (0–5)
Inflight Service	Satisfaction with overall inflight service (0–5)
Cleanliness	Satisfaction with cabin/tray cleanliness (0–5)
Departure Delay	Departure delay duration (in minutes)
Arrival Delay	Arrival delay duration (in minutes)
Satisfaction	Target variable: Overall satisfaction level (Satisfied, Neutral/Dissatisfied)

Appendix B

Figure 5: Correlation matrix

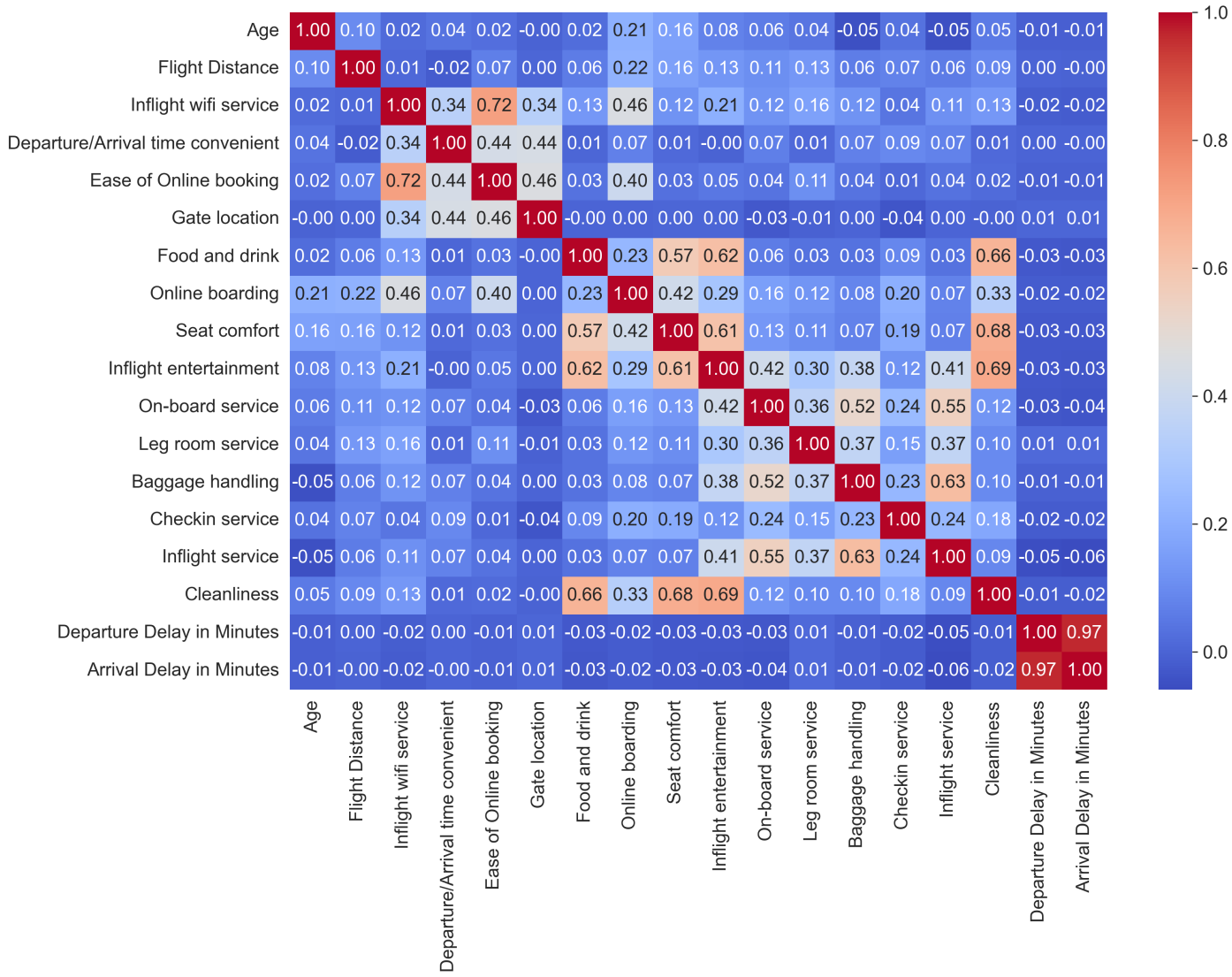

Figure 6: Histograms for numerical features

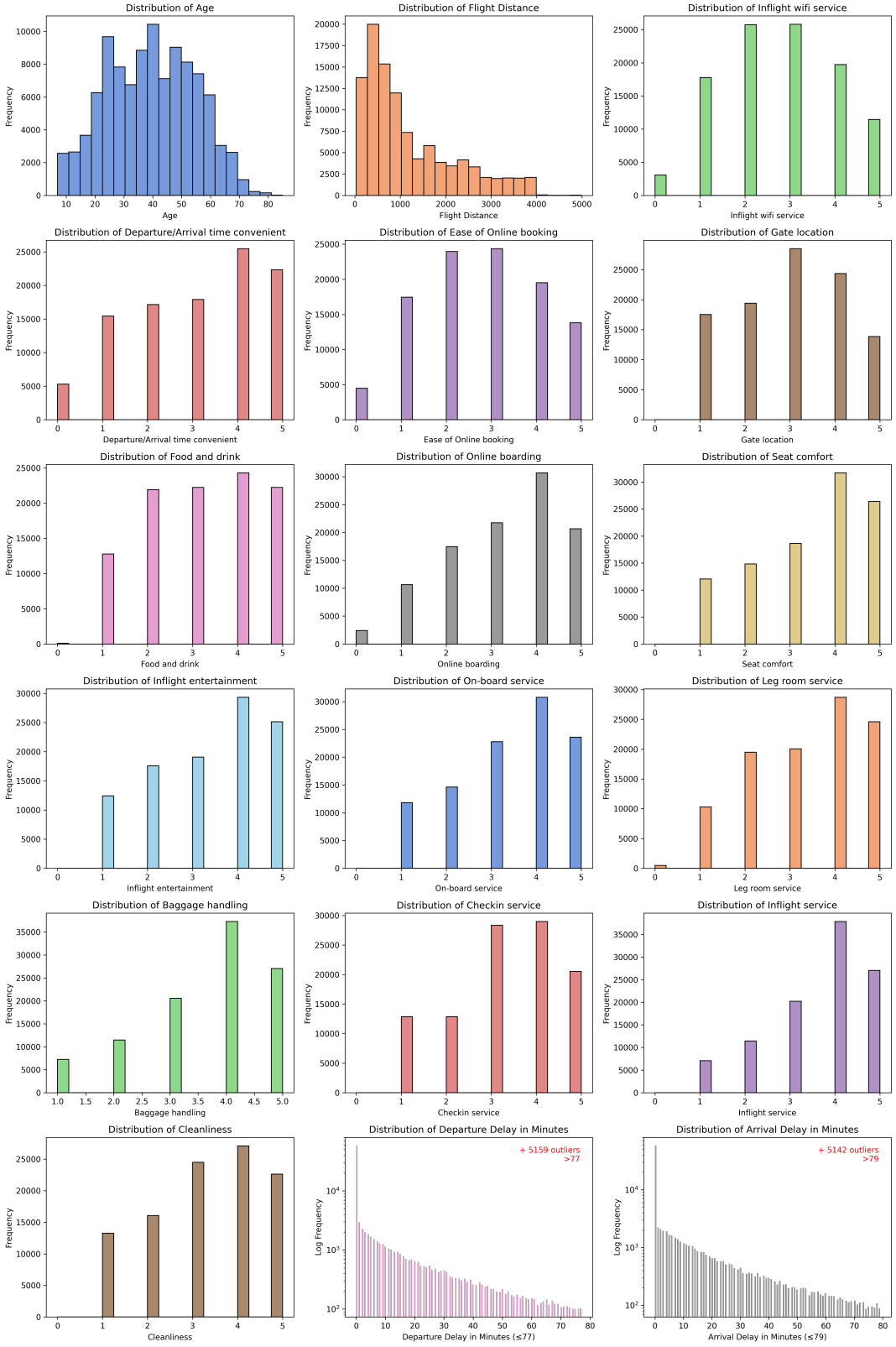

Figure 7: Pie charts for categorical features