Thanh Huan Hoang
1691152

Introduction to Machine Learning (COMP90049)
Semester 1 - 2025

1. Introduction

In this project, I attempted to **classify whether a customer is satisfied with their flight** using the dataset from the US Airline passenger satisfaction survey, which is publicly available on Kaggle[1]. It originally contains **103,904 rows** and 25 columns (including the classification column “satisfaction”). Section 3 shows how data cleaning and feature engineering were performed, resulting in **103,594 instances** with **52 features** that will be used for training the models for the classification task.

I chose this dataset because as a travel enthusiast who has set foot in 25 countries before turning 25 (yes, that’s a small flex right there), I pay special attention to the service I receive whenever I board a flight. Understanding what factors contribute to passenger satisfaction is personally meaningful, but also valuable from a broader perspective. Airlines could use such models to determine important aspects of customer satisfaction and improve their services accordingly. This not only enhances the passenger experience but also increases customer retention, strengthens brand loyalty, and supports operational improvements at scale.

2. Literature Review

The dataset “Airline Passenger Satisfaction” was uploaded to Kaggle by TJ Klein (username “teejmahal20”) and last updated on February 21st, 2020. It does not include any information about when, where, or how the data was collected. Many users have asked for the source of the data, but no answer was given. Nonetheless, the dataset has a usability score of 9.41 and has been used by many others (namely more than 400 Kaggle users at the time of writing this report) for analytical and educational purposes, especially for classification tasks, as in this project. Acknowledging the absence of provenance information, no claims are made about the real-world representativeness or generalizability of the findings. The main focus is on the practice of developing and applying classification models to tabular data and evaluating their performance.

Two recent studies also explore methodological and feature choices in this project. *Hong et al. (2023)* applied several machine learning algorithms, which include random forest, k-nearest neighbors (k-NN), logistic regression, and naive bayes, to the same dataset used here. Their work emphasized the importance of feature engineering and categorical one-hot encoding, with Random Forest achieving the best performance across different metrics. They also pointed out that the satisfaction often correlates more with certain aspects such as “Seat comfort”, “Cleanliness”, “Flight distance”,… Their approach aligns with the one taken in this project, which compares multiple supervised learning models after cleaning, preprocessing and transforming data.

Using the same dataset again, *Mirthipati (2024)* further investigated airline customer satisfaction by combining machine learning with causal inference techniques. Data preprocessing steps include handling missing values through median imputation, one-hot encoding of categorical variables, and normalization using min-max scaling. The study implemented different models, including logistic regression and tree-based classifiers, to identify factors affecting satisfaction. Additionally, causal analysis was used to determine the direct effects of service enhancements. They found that improving digital features such as “Online boarding” and “Ease of Online booking could significantly improve overall customer satisfaction. These findings highlight how data-driven models can provide valuable and actionable insights for airline service design.

3. Feature Construction and Preprocessing

Data Exploration: Before starting to preprocess and construct new features, I had a quick **overview of the data** by observing **the number of distinct values** in each column, as well as looking at **histograms of attribute values** for each column, to see whether there was anything out of the ordinary with the data. The number of distinct values for the unnamed column (which is the row index) and the ID column matches the total number of rows, implying that these are unique values and aren’t necessary as features for classification since they don’t bring any significance. The scores for service categories appear to be normal, with all values being integers ranging from 0 to 5. For categorical attributes like “Class” or “Type of Travel,” there are no unexpected values and all of them only have a small number of values. Ages seem to be normally distributed, whereas delays and flight distances are right-skewed. The proportions of satisfaction and neutral/dissatisfaction are 43.33% and 56.67%, respectively. This shows our dataset is pretty balanced, so there’s really no need for resampling.

[1] https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction?select=train.csv

Preprocess and Feature Construction: The first thing that I did was to **remove duplicates** and **drop rows that contain missing values**. While duplicates may indicate common patterns in the population, keeping them could bias the model by overemphasizing frequent cases and reducing generalization. Removing them from the dataset helps the model focus on learning from the diversity of responses rather than memorizing repeated patterns. The number of instances comes down to 103,594, with the percentage satisfaction being 43.34%. This proportion technically stays the same. **Checking for invalid attribute values is unnecessary,** as the histograms indicate there are none. Service scores across different categories are always integers from 0 to 5, and delays and flight distances are positive values. As mentioned above, the **unique values of customer ID and row numbers** do not help with classification and were therefore **removed**.

Binning was used for flight distance, delays, and age, creating new columns named "Flight Distance Category", "Departure Delay Category", "Arrival Delay Category", and "Age Category". According to some US airlines, flight distances can be bucketed into three main categories: short-haul (<700 miles), medium-haul (700-2999 miles)[2], and long-haul (≥3000 miles)[3]. For the departure and arrival delays, according to some airlines and my knowledge, they can be grouped into four main categories: on-time (<15 minutes)[4], minor (15-59 minutes), major (60-179 minutes), and severe (≥180 minutes). [5]Ages are divided into the following groups[6]: <18, 18-24, 25-34, 5-44, 45-54, 55-64, ≥65. The reason why these intervals were determined that way is that there are often compensations or benefits when people travel these distances or when their flights are delayed by certain minutes. Note that the original columns of the attributes, which have numeric values, were not removed when binning was done.

One-hot encoding was then applied to categorical attributes, including the category columns resulting from the binning above. In this step, the original columns were removed. These include "Customer Type", "Type of Travel", "Class", "Gender", "Flight Distance Category", "Departure Delay Category", "Arrival Delay Category", and "Age Category". **Ordinal encoding was considered** for "Class" since there's a clear order: Business > Eco Plus > Eco. But since I am using logistic regression later, ordinal encoding may falsely assume equal distance between categories. That's why one-hot encoding was used instead.

Interaction terms and new features were added. I focus on aspects that, when combined, could potentially affect customers' satisfaction from my point of view. "Average Service Score" is the **mean** score of multiple service-related aspects, providing a general sense of perceived quality. "Total Delay" (**sum** of departure and arrival delay) and "Delay Ratio" (**division**: total delay/flight distance) quantify not just the presence of delays but also their severity relative to flight distance, offering a normalized view of delay inefficiency. Interaction terms, which are **products** of attributes, like "Age × Flight Distance" aim to reflect how travel burden may vary with age, while "Booking × Boarding" captures the online service experience, which is important in a digital world. Similarly, "Checkin × Baggage" and "Comfort × Legroom" were included to represent the ground experience and physical comfort during the flight, both of which are likely to impact satisfaction more strongly in combination than as isolated factors.

Normalization, rescaling, and transformation were applied to all numeric values (except for binary 0 and 1 attributes). To address skewed distributions in the data, a **log transformation** (*log1p*) was used for highly right-skewed attributes like flight distance and delays. This scales down extreme values, reduces the influence of outliers, and makes the data more normally distributed. This is particularly beneficial for models like logistic regression, which assume a linear relationship between attributes and the log-odds of the outcome. All numeric attributes were then **rescaled to a [0, 1] range** using **min-max normalization.** This is important for k-NN, which relies on distance, and helps logistic regression converge more efficiently. Random forest, on the other hand, doesn't require scaling as it can deal with unscaled values.

As a final step, I **convert the nominal values** of the target attribute "satisfaction" **into numeric values**. The categories "neutral or dissatisfied" and "satisfied" are converted into binary values 0 and 1, respectively, since models like logistic regression and k-NN require numeric input. Now, there are **52 attributes**, ready for the classification task.

[2] https://www.usatoday.com/story/travel/roadwarriorvoices/2016/04/22/united-airlines-mileage-plus-points/83365166
[3] https://www.pointhacks.com.au/differences-short-medium-long-haul-flights/
[4] https://www.fly.faa.gov/flyfaa/usmap.jsp?legacy=true
[5] https://www.transportation.gov/airconsumer/airline-cancellation-delay-dashboard
[6] https://www.smartsurvey.com/survey-questions/demographics/age-groups

4. Models and Training

Model Selection: The following three machine learning algorithms were selected: **k-NN (low bias, high variance)**, a non-parametric model, which captures complex patterns but is sensitive to noise; **random forest (moderate bias and variance**), an collections of decision trees, balances bias and variance by reducing overfitting while maintaining good flexibility; **logistic regression (high bias, low variance)**, a linear model performs well when the features and the target is somewhat linear to each other. This diverse selection allows us to compare performance across models with varying complexity and generalization ability. The logistic regression model was instantiated with *max_iter=1000* to ensure convergence during optimization and *solver='liblinear'*, which supports both L1 and L2 regularization. The k-nearest neighbors (k-NN) model used the default constructor. The random forest model was initialized with *random_state=2025* for reproducibility.

Cross-validation: To evaluate model performance fairly, **k-fold cross-validation** with **k=10** was used as the outer loop to split the dataset into training and testing partitions. A random state with the value of 2025 was set for the reproducibility of the results. All **103,594 instances** from the dataset were used for this process.

Hyperparameter Tuning: For each fold, a **nested 3-fold cross-validation** was done within the training data for hyperparameter tuning using *GridSearchCV*. This structure was applied across all three models. Each model had three hyperparameters tuned over a well-defined grid. For logistic regression, I tuned the **regularization strength (C), penalty type (l1 or l2)**, and **intercept fitting**. For k-NN, I tuned the **number of neighbors (k),** the **weighting scheme**, and the **distance metric (p).** For random forest, I only tuned the **tree depth**. During each outer fold, the best hyperparameters were selected based on the **F1-score,** as it provides a balanced measure of model performance by combining precision and recall.

5. Results and Discussion

Performance was evaluated using five metrics: accuracy, precision, recall, F1-score, and ROC AUC. The results were rounded to four decimal places. This procedure ensures robust model comparison while adhering to proper machine learning evaluation practices. The table below lists the hyperparameter values tested, the frequency with which each value was selected as optimal (out of 10 folds), and the average performance across all five metrics.

Model	Hyperparameter(s)	Values Tested (Frequency of Being Best)	Accuracy	Precision	Recall	F1-score	ROC AUC
Logistic regression	C	0.1 (0/10), **1 (7/10)**, 10 (3/10)	0.8804	0.8755	0.8442	0.8595	0.9405
	penalty	**‘l1’ (7/10)**, ‘l2’ (3/10)					
	fit_intercept	**True (7/10)**, False (3/10)					
k-NN	n_neighbors	7 (3/10), **11 (7/10),** 15(0/10)	0.9174	0.9368	0.8681	0.9011	0.9690
	p	1 (10/10), 2(0/10)					
	weights	‘uniform’ (0/10), ‘**distance’ (10/10)**					
Random forest	max_depth	10 (0/10), 20 (1/10), **50 (9/10)**, 100 (0/10), 200 (0/100)	0.9604	0.9691	0.9386	0.9536	0.9931

As shown in the result table above, for hyperparameters with numeric values, the middle value was chosen most frequently out of 10 folds. Interestingly, for all models, ROC AUC was consistently the highest-scoring metric, indicating that each classifier was particularly good at ranking satisfied and dissatisfied passengers even when precision or recall varied. This shows strong discriminative ability, meaning the models effectively separated the two classes, even if the classification threshold could still be tuned for specific use cases.

The three models evaluated displayed different performance levels. Among them, random forest achieved the highest overall performance in all five metrics, with an accuracy of 0.9604 and an F1-score of 0.9536. This is due to its ensemble structure, robustness to overfitting, and ability to capture non-linear relationships and feature interactions, even without heavy preprocessing. Notably, the best *max_depth* value was the middle candidate (50), selected in 9 out of 10 folds. This shows that trees don’t necessarily have to be too deep to perform well.

k-NN ranked second, performed comparatively well (worse than random forest but better than logistic regression in every category), especially in terms of precision (0.9368) and ROC AUC (0.9690). As optimal values, the model landed on *k = 11*, *distance-based weighting*, and *Manhattan distance (p = 1)* in nearly all folds, showing that local structure and absolute feature differences matter in this dataset. The preference for *p = 1* suggests that the model benefited from treating all feature differences linearly, which is often more robust in high-dimensional or heterogeneous feature spaces.

Meanwhile, logistic regression had the lowest scores across all metrics with an F1-score of 0.8595 and an accuracy of 0.8804. This likely happened due to its linear assumptions, which limit its capacity to model complex and non-linear interactions. As for the chosen value for C*,* compared to *C=0.1*, which enforces a stronger regularization and may underfit the model, and *C=10*, which weakens regularization and may overfit the model, *C=1* is the sweet spot in the middle, striking a balanced trade-off between model complexity and generalization. This suggests that a moderate regularization was optimal for this dataset, enough to prevent overfitting while still allowing informative features to contribute to the model. '*l1'* as the most frequently chosen value for *penalty* indicates that the model performs better with sparse feature selection, which may help reduce noise in a wide feature space, and *fit_intercept=True* makes sure that the decision boundary handles non-centered data properly.

Other methods were considered but then excluded. Since I still want to use the whole dataset instead of a subset of it for better results, support vector machines were not included in the end due to their poor scalability with large datasets and their enormous computational cost, especially with nested cross-validation. I tried to train support vector machines, but due to the long runtime, the kernel in my Jupyter Notebook keeps resetting itself, making it impossible for the training and evaluation to be completed. Neural networks were also an option but are known to underperform on structured tabular data. *Shwartz-Ziv and Armon (2022)* stated that they struggle with sparse feature interactions and categorical variables, both of which are naturally handled better by tree-based methods like random forest. The lack of inherent spatial or temporal relationships and the heavy reliance on feature engineering make it hard for neural networks to work well in this case. In contrast, the chosen models offer a more balanced trade-off between performance and practicality, making them well-suited for classification on survey-based tabular data.

6. Conclusion

This project shows all of the necessary steps to predict airline passenger satisfaction using three different supervised machine learning methods: from exploring the data to processing it and choosing and evaluating the models. Through careful preprocessing, feature engineering, and evaluation using various metrics with nested cross-validation and hyperparameter tuning, models with varying bias and variance levels were compared. Random forest proved to be the most effective model overall, achieving high accuracy and generalization performance, followed by k-NN and, lastly, logistic regression, which had the least impressive performance. The results also highlight the importance of proper data preparation, data processing, and hyperparameter tuning in helping the model achieve the highest performance. Knowing what machine learning algorithms to use and not to use based on the dataset and computation resources also plays a crucial role. Understanding how different algorithms respond to the structure and distribution of data allowed for deeper insights into model selection and applicability, showing that predictive modeling in this context can offer valuable guidance for real-world airline service optimization.

7. Bibliography

Hong, A. C. Y., KHAW, K. W., Chew, X., & Yeong, W. C. (2023). Prediction of US airline passenger satisfaction using machine learning algorithms. Data Analytics and Applied Mathematics (DAAM), 7-22.
https://journal.ump.edu.my/daam/article/download/9071/2824/36759

Mirthipati, T. (2024). Enhancing airline customer satisfaction: A machine learning and causal analysis approach. arXiv preprint arXiv:2405.09076.
https://arxiv.org/pdf/2405.09076

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. Information Fusion, 81, 84-90.
https://arxiv.org/abs/2106.03253