Airline Passenger Satisfaction Predictive Models

ISOM3360 Project Group 1

CHEN, Hsuan-ching (Cathy) 20825951
NG, Wai Yan (Yanni) 20772790
OEI, Steven 20627517
PUN, Yu Tung (Wendy) 20777283

1. Introduction

In recent years, airline passenger satisfaction has become increasingly vital with the rise of competitors in the industry. It has become crucial for airline companies to pursue and preserve customer loyalty to differentiate themselves from other competitors. Also, correctly predicting passengers' satisfaction with their feedback makes it possible to provide remedial measures timely. In this report, we will use machine learning algorithms to predict customer satisfaction based on a variety of attributes.

The study aims to develop an accurate and reliable binary classification machine learning model that can identify essential characteristics that have an impact on customer satisfaction and forecast it. We begin by analyzing the dataset, and observing distributions in the features. Additionally, we performed feature selection by eliminating features with low correlation to the target variable. We then build four different machine learning models, including Decision Tree, Logistic Regression, Naive Bayes, and Random Forest. The performance is then evaluated based on each model's accuracy score, AUC, and False Positive Rate.

Finally, we present our results and discuss the key findings and major conclusions, including the features used and dropped, the best-performing models, and applications. This study provides insight and suggestions for airline firms to focus on specific elements that could significantly boost customer satisfaction.

2. Data Understanding

1. Link to our dataset: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction
2. Number of records: 103,904
3. Number of attributes: 24 (including the target column)
4. Attribute description:
   a. Features:
      1. ID: unique identifier for each passenger
      2. Gender: Male or Female
      3. Customer Type: Loyal or disloyal customer
      4. Age: Passenger age in years
      5. Type of Travel: Personal or Business
      6. Class: Travel class (Eco, Eco Plus, Business)
      7. Flight Distance: Distance traveled in miles (numerical)
      8. Inflight wifi service: rating (0-5)
      9. Departure/Arrival time convenient: rating (0-5)
      10. Ease of Online booking: rating (0-5)
      11. Gate location: rating (0-5)
      12. Food and drink: rating (0-5)
      13. Online boarding: rating (0-5)
      14. Seat comfort: rating (0-5)
      15. Inflight entertainment: rating (0-5)
      16. On-board service: rating (0-5)
      17. Leg room service: rating (0-5)
      18. Baggage handling: rating (0-5)
      19. Checkin service: rating (0-5)
      20. Inflight service: rating (0-5)
      21. Cleanliness: rating (0-5)
      22. Departure delay in minutes: minutes delayed (numerical)
      23. Arrival delay in minutes: minutes delayed (numerical)

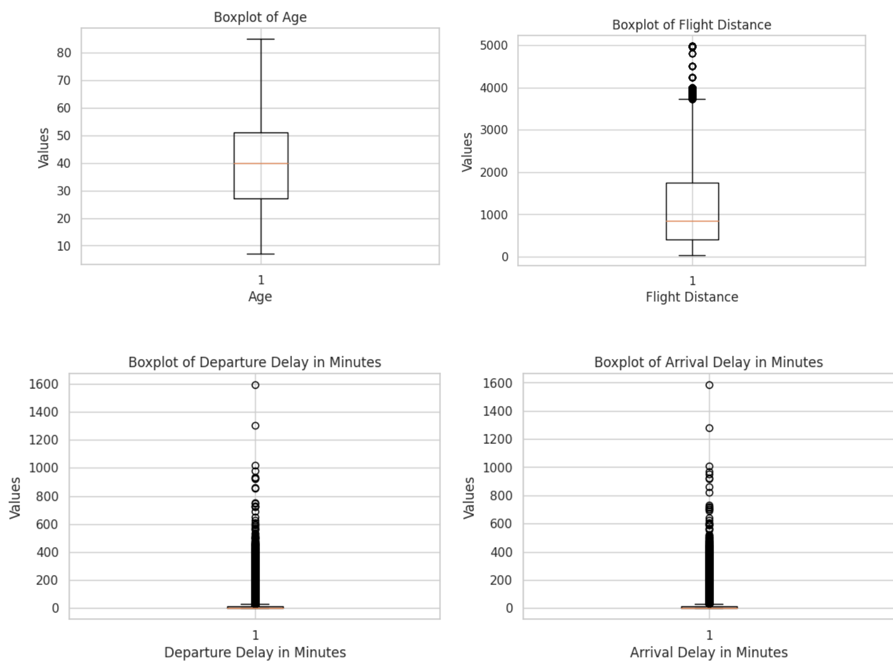
   b. Target:
      1. satisfaction: Satisfaction level (satisfied / neutral or dissatisfied)
5. The table below shows the description of each numerical and ordinal features. On the other hand, our categorical features are: 'Gender', 'Customer Type', 'Type of Travel', 'Class' with the distribution shown with the histograms below.

	mean	std	min	25%	50%	75%	max
Age	39.379706	15.114964	7.0	27.0	40.0	51.0	85.0
Flight Distance	1189.448375	997.147281	31.0	414.0	843.0	1743.0	4983.0
Inflight wifi service	2.729683	1.327829	0.0	2.0	3.0	4.0	5.0
Departure/Arrival time convenient	3.060296	1.525075	0.0	2.0	3.0	4.0	5.0
Ease of Online booking	2.756901	1.398929	0.0	2.0	3.0	4.0	5.0
Gate location	2.976883	1.277621	0.0	2.0	3.0	4.0	5.0
Food and drink	3.202129	1.329533	0.0	2.0	3.0	4.0	5.0
Online boarding	3.250375	1.349509	0.0	2.0	3.0	4.0	5.0
Seat comfort	3.439396	1.319088	0.0	2.0	4.0	5.0	5.0
Inflight entertainment	3.358158	1.332991	0.0	2.0	4.0	4.0	5.0
On-board service	3.382363	1.288354	0.0	2.0	4.0	4.0	5.0
Leg room service	3.351055	1.315605	0.0	2.0	4.0	4.0	5.0
Baggage handling	3.631833	1.180903	1.0	3.0	4.0	5.0	5.0
Checkin service	3.304290	1.265396	0.0	3.0	3.0	4.0	5.0
Inflight service	3.640428	1.175663	0.0	3.0	4.0	5.0	5.0
Cleanliness	3.286351	1.312273	0.0	2.0	3.0	4.0	5.0
Departure Delay in Minutes	14.815618	38.230901	0.0	0.0	0.0	12.0	1592.0
Arrival Delay in Minutes	15.133392	38.649776	0.0	0.0	0.0	13.0	1584.0

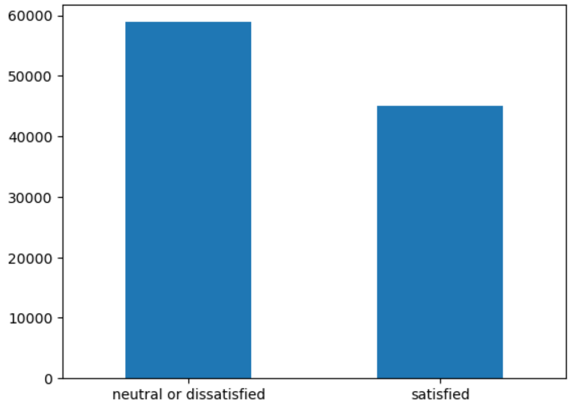
6. Missing values: In our training data, the only feature with missing value is “Arrival Delay in Minutes” where the entry for it is NaN and the number of missing value is 310.

```
Arrival Delay in Minutes    310
dtype: int64
```

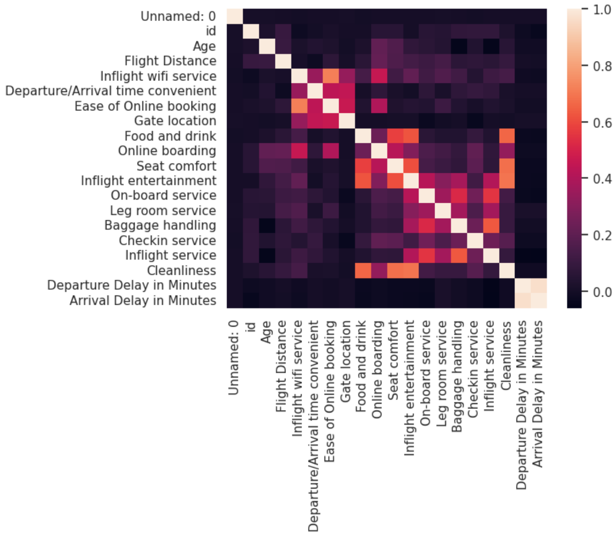
7. Outliers: In this project, boxplots were used to analyze the distribution of the data and identify any outliers that may be present. From the boxplots below, we can see that “Age” doesn’t have any outliers while the others have. However, this might not be the case as this can happen when the data is distributed in such a way that it has a lot of variability, but the median and quartiles are relatively close together.



8. Class imbalance: Our target class is ‘Satisfaction’ which indicates if the passenger is satisfied (‘satisfied’) or not satisfied (‘neutral or dissatisfied’).There is slight imbalance in our target class where 56.67% passengers are neutral or dissatisfied and 43.33% are satisfied.



9. Correlation
    a. The correlation graphs of both numerical and categorical data presented on the figures below.
    b. Departure Delay in Minutes and Arrival Delay in minutes are the only features that are highly correlated.



3. Model Building

We have chosen the Decision tree model, logistic regression model, Naive Bayes model, KNN model, and the Random forest model for model comparison. For each model, we keep the random state constant (42, as it is the most popular value used in general), and we use train test split to tune models separately before getting performance of our best models to do classifier selection.

Decision Tree

Our first step is to test different data preparation (feature engineering) methods. Data preparation methods are summarized below:

	train_df same (after fillna)	data preparation sorted 01	data preparation sorted 02	data preparation sorted 03	data preparation sorted 04	data preparation sorted 05
0	Gender	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding
1	Customer Type	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding
2	Age	standard scalar	standard scalar	standard scalar	minmax	standard scalar
3	Type of Travel	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding
4	Class	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding
5	Flight Distance	Discretization+One-hot encoding	Discretization + One-hot encoding	Discretization + One-hot encoding	Discretization + One-hot encoding	Discretization + One-hot encoding
6	Inflight wifi service	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
7	Departure/Arrival time convenient	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
8	Ease of Online booking	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
9	Gate location	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
10	Food and drink	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
11	Online boarding	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
12	Seat comfort	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
13	Inflight entertainment	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
14	On-board service	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
15	Leg room service	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
16	Baggage handling	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
17	Checkin service	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
18	Inflight service	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
19	Cleanliness	Leave it alone	minmax	One-hot encoding	minmax	standard scalar
20	Departure Delay in Minutes	standard scalar	standard scalar	standard scalar	minmax	standard scalar
21	Arrival Delay in Minutes	standard scalar	standard scalar	standard scalar	minmax	standard scalar
22	satisfaction	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding	One-hot encoding
			(Used in Decision tree 02)		(used in Logistic 04)	(used in Logistic 05)
			(Cathy)			
	Decision tree model (untuned)					
	simple accuracy	0.945199415	0.945262499	0.945199415	0.94528656	0.945454983

Treatment stays the same for [Gender, Customer Type, Type of Travel, Class, Flight Distance, and satisfaction (target)], others vary between methods.
We decided to fit each data preparation method into decision tree model fitting and compare their simple accuracy to find the best dataset, and both 04 and 05 performed well. We had 2 people responsible for the decision tree model fitting with each dataset in parallel.

Decision Tree Model Hyperparameter Tuning (yanni)

After choosing the dataset to use, we move on to hyperparameter tuning. The main trials of the tuning process have been summarized below.

1. Untuned model
2. Tuning the ccp_alpha parameter with GridSearchCV, which finds the best route along the tree.
3. Tuning the criterion parameter with GridSearchCV
4. Another 5 hyperparameter of decision tree classifier has been chosen to tune with GridSearchCV which are max_depth, max_leaf_nodes, min_impurity_decrease, min_samples_leaf, min_samples_split

We have chosen to first tune the model with GridSearchCV and ccp_alpha is because we think finding the best path may be a quick way to tune the best model, but we would like to explore other tuning methods to try to tune out the best model as well, so we then move on to tuning 5 of the most important parameters mentioned in class. Before putting all 5 parameters into GridSearchCV, we first find the possible best parameter range with a few tests rounds first as tuning all parameters requires huge code-running time.
The performances are summarized below.

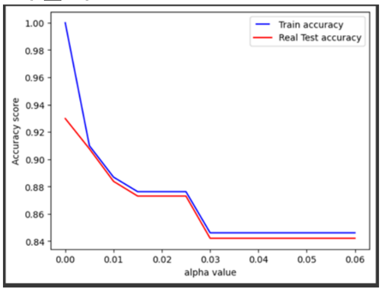
Table 1: Decision tree model 04 Performance evaluation

	Simple accuracy	Cross-val accuracy	cross-val AUC	tree nodes
Untuned	0.944198	0.943101	0.942176	4467
Pruning ccp_alpha (scoring = accuracy+AUC)	0.906125	0.910167	0.951978	23
Pruning criterion (scoring = accuracy)	0.947009	0.946874	0.945815	4043
sep grids 1	0.932515	0.935344	0.985335	303
sep grids 2	0.955709	0.956556	0.981894	879
sep grids 3	0.932515	0.956556	0.985337	303
sep grids 4	0.957865	0.958404	0.984748	879
sep grids 5	0.935787	0.937596	0.986531	879

Table 2: Decision tree model 04 Tuning hyperparameter details

	random_state	Criterion	max_depth	max_leaf_nodes	min_impurity_decrease	min_samples_leaf	min_samples_split	ccp_alpha
Untuned	42	Default	Default	Default	Default	Default	Default	Default
Pruning ccp_alpha	42	Default	Default	Default	Default	Default	Default	0.005
Pruning criterion (s	42	Entropy	Default	Default	Default	Default	Default	Default
sep grids 1	42	Default	925	440	Default	120	16	Default
sep grids 2	42	Default	925	440	Default	Default	Default	Default
sep grids 3	43	Default	925	Default	Default	120	Default	Default
sep grids 4	42	Entropy	925	440	Default	Default	Default	Default
sep grids 5	42	Entropy	925	Default	Default	120	Default	Default

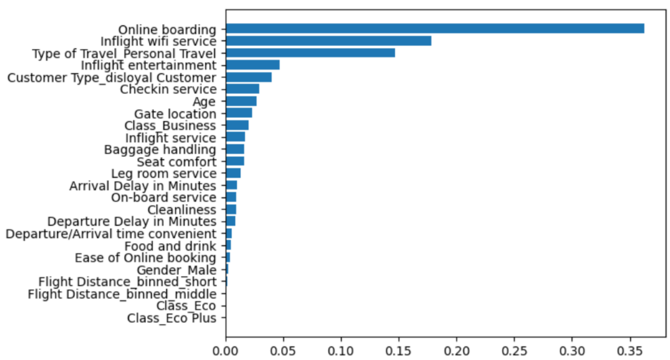
Check Overfitting
Real train vs real test, along ccp_alpha values

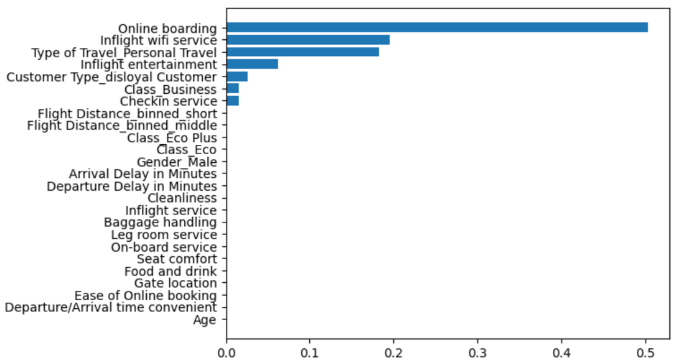


To ensure that the high accuracy scores are not due to overfitting the training set, we check the overfittedness by comparing accuracy scores along parameter values between the sub training and testing set along max_leaf_nodes values, and between the whole training and testing set along ccp_alpha value.
Check overfitting: The results show that there should be no overfitting problem.

Decision Tree Model Feature Selection (Yanni)

We then move on to the attempt of feature importance determination and feature selection. For the decision tree model, sklearn provides the feature-importance feature. (LHS: Untuned DT model, RHS: Tuned DT model)





<u>Fitting into the hyperparameters</u>

- Dropped Dataset from feature selection 4a1 has dropped features of ['Departure/Arrival time convenient', 'Ease of Online booking', 'Cleanliness','Departure Delay in Minutes', 'Arrival Delay in Minutes', 'Gender_Male','Class_Eco', 'Class_Eco Plus', 'Flight Distance_binned_middle', 'Flight Distance_binned_short']

- Dropped Dataset from feature selection 4a2 has dropped features of ['Age','Departure/Arrival time convenient', 'Ease of Online booking', 'Food and drink', 'Seat comfort','On-board service', 'Leg room service', 'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes', 'Gender_Male', 'Class_Eco', 'Class_Eco Plus', 'Flight Distance_binned_middle', 'Flight Distance_binned_short']

After getting the feature importances of each model, we have decided to try to drop 10 and 15 features and fit the two datasets into the decision tree fitting process again. We fit the dimensionally-reduced dataset into our tuned models, train them with the whole training set and evaluate against the real test set from Kaggle, and we all get higher cross validation accuracies and AUC scores.
The results show that by fitting the dataset with the 10 least important features dropped has the highest cross-validation accuracy and AUC score.
The best model from tuning and feature selection is [sep grid 4] with dataset 4a1, with parameter values: Criterion: Entropy, max_depth: 925, max_leaf_nodes: 440, cv accuracy: 96.03% and cv AUC: 0.9913.

Table 3: Decision tree model 04a1	using dropped dataset + whole train set to train + real test set to evaluate							
	random_state	Criterion	max_depth	max_leaf_nodes	min_impurity_decrease	min_samples_leaf	min_samples_split	ccp_alpha
Untuned	42	Default	Default	Default	Default	Default	Default	Default
Pruning ccp_alpha (scoring = accuracy+AUC)	42	Default	Default	Default	Default	Default	Default	0.005
Pruning criterion (scoring = accuracy)	42	Entropy	Default	Default	Default	Default	Default	Default
sep grids 1	42	Default	925	440	Default	120	16	Default
sep grids 2	42	Default	925	440	Default	Default	Default	Default
sep grids 3	43	Default	925	Default	Default	120	Default	Default
sep grids 4	42	Entropy	925	440	Default	Default	Default	Default
sep grids 5	42	Entropy	925	Default	Default	120	Default	Default

Table 4: Decision tree model 04a1	* evaluated with sub train and test set			
	Simple accuracy	Cross-val accuracy	cross-val AUC	tree nodes
Untuned	0.932245	0.946248	0.945789	10129
Pruning ccp_alpha (scoring = accuracy+AUC)	0.907646	0.908011	0.952648	23
Pruning criterion (scoring = accuracy)	0.929165	0.946951	0.946367	9765
sep grids 1	0.932938	0.938414	0.987093	533
sep grids 2	0.942870	0.959665	0.990036	879
sep grids 3	0.932938	0.959665	0.987103	533
sep grids 4	0.940214	0.960290	0.991315	879
	0.932207	0.941427	0.987928	507

Table 5: Decision tree model 4a2	using dropped dataset + whole train set to train + real test set to evaluate							
	random_state	Criterion	max_depth	max_leaf_nodes	min_impurity_decrease	min_samples_leaf	min_samples_split	ccp_alpha
Untuned	42	Default	Default	Default	Default	Default	Default	Default
Pruning ccp_alpha (scoring = accuracy+AUC)	42	Default	Default	Default	Default	Default	Default	0.005
Pruning criterion (scoring = accuracy)	42	Entropy	Default	Default	Default	Default	Default	Default
sep grids 1	42	Default	925	440	Default	120	16	Default
sep grids 2	42	Default	925	440	Default	Default	Default	Default
sep grids 3	43	Default	925	Default	Default	120	Default	Default
sep grids 4	42	Entropy	925	440	Default	Default	Default	Default
sep grids 5	42	Entropy	925	Default	Default	120	Default	Default

Table 6: Decision tree model 4a2	using dropped dataset + whole train set to train + real test set to evaluate			
	Simple accuracy	Cross-val accuracy	cross-val AUC	tree nodes
Untuned	0.940868	0.948972	0.966574	9779
Pruning ccp_alpha (scoring = accuracy+AUC)	0.907646	0.908011	0.952648	23
Pruning criterion (scoring = accuracy)	0.940984	0.949415	0.967181	9447
sep grids 1	0.929319	0.936393	0.985766	533
sep grids 2	0.947682	0.955815	0.989490	879
sep grids 3	0.929319	0.955815	0.985761	533
sep grids 4	0.949415	0.957172	0.990987	879
sep grids 5	0.936480	0.940127	0.990987	879

Decision Tree Model Hyperparameter Tuning (Cathy)

We chose the 05 dataset and divided it into the sub-train dataset to train the decision tree model, and evaluate the performances with the sub-test set, with cross-validation, and with AUC of cross-validation-prediction. In the following hyperparameter tuning process, we tried 3 criteria ['gini', 'entropy', 'log_loss'] respectively. We consider 6 parameters ['max_depth', 'max_leaf_nodes, 'min_samples_split', 'min_samples_leaf', 'min_weight_fraction_leaf', 'min_impurity_decrease']. To better understand each hyperparameter, we found their optimal values when working alone in the DecisionTreeClassifier first before trying different combinations. We narrowed down the possible best range with several rounds of tests from larger steps of range. The results are as below:

	criterion	max_depth	max_leaf_nodes	min_samples_split	min_samples_leaf	min_weight_fraction_leaf	min_impurity_decrease	random_state
optimal value	default = 'gini'	13	375	58	10	default = 0.0	default = 0.0	42
	'entropy'	16	440	93	12	default = 0.0	default = 0.0	42
	'log_loss'	16	440	93	15	default = 0.0	default = 0.0	42

The model performs best when min_weight_fraction_leaf and min_impurity_decrease are at default values under all three criteria, so we will not include them in the following tables. We found that the model performs better when we only consider max_leaf_nodes under all three criteria. Any combination with it leads to a drop in accuracy. Both simple accuracy and cross-validation accuracy of most combinations of parameters range from 0.953 to 0.955, whereas the accuracy of considering max_leaf_nodes only is around 0.9570 to 0.9593. We also found that the optimal values and the performances under {'criterion': 'log_loss'} are highly similar to that of {'criterion': 'entropy'}, and the best results are exactly the same.

criterion									
default = 'gini'	max_depth	max_leaf_nodes	min_samples_split	min_samples_leaf	Simple Accuracy	Cross-val Accuracy	Cross-val AUC	Tree Nodes	vs baseline
# 0 base line	default	default	default	default	0.9454549829	0.9459982291	0.9452651132	5369	-
1	13	default	default	default	0.9541408017	0.9533800431	0.9506758063	1563	risen
2	default	375	default	default	0.9570761754	0.9584327841	0.9557795007	749	risen
3	default	default	58	default	0.9531783841	0.9541018633	0.9517856496	1309	risen
4	default	default	default	10	0.9533708676	0.9527929627	0.9505471243	1947	risen
5	13	375	default	default	0.9553438237	0.9549391746	0.9520070889	749	risen
6	13	default	58	default	0.9535392907	0.9532068063	0.9501884929	743	risen
7	13	default	default	10	0.9534189885	0.9533415460	0.9504040599	1051	risen
8	default	375	58	default	0.9536595929	0.9557668617	0.9530849225	749	risen
9	default	375	default	10	0.9554641259	0.9556609948	0.9529679940	749	risen
10	default	default	58	10	0.9529137193	0.9531394364	0.9506333482	1121	risen
11	13	375	58	default	0.9533708676	0.9532645519	0.9502472837	745	risen
12	default	375	58	10	0.9532986863	0.9536495226	0.9509710670	749	risen
13	13	default	58	10	0.9532265050	0.9524464891	0.9493791392	711	risen
14	13	375	default	10	0.9535392907	0.9533992917	0.9504079788	749	risen
15	13	375	58	10	0.9531783841	0.9524561133	0.9493876312	711	risen

criterion									
'entropy'	max_depth	max_leaf_nodes	min_samples_split	min_samples_leaf	Simple Accuracy	Cross-val Accuracy	Cross-val AUC	Tree Nodes	vs baseline
# 0 base line	default	default	default	default	0.9475723016	0.9478075916	0.9471751616	4853	-
1	16	default	default	default	0.9545257687	0.9530143209	0.9507685696	2237	risen
2	default	440	default	default	0.9592416149	0.9592027256	0.9562811795	879	risen
3	default	default	93	default	0.9544535874	0.9545927010	0.9523938088	821	risen
4	default	default	default	12	0.9541167413	0.9528988297	0.9507581189	1613	risen
5	16	440	default	default	0.9584716809	0.9566137974	0.9538165401	879	risen
6	16	default	93	default	0.9549107358	0.9537746381	0.9514551147	647	risen
7	16	default	default	12	0.9558731534	0.9527255929	0.9503230645	1157	risen
8	default	440	93	default	0.9542851643	0.9555166307	0.9529451321	821	risen
9	default	440	default	12	0.9567633896	0.9559882199	0.9532593347	879	risen
10	default	default	93	12	0.9526490544	0.9536687712	0.9514270265	733	risen
11	16	440	93	default	0.9545979501	0.9537650139	0.9514440097	645	risen
12	default	440	93	12	0.9526490544	0.9541018633	0.9516732928	733	risen
13	16	default	93	12	0.9536114720	0.9528988297	0.9506144067	595	risen
14	16	440	default	12	0.9569799336	0.9543617185	0.9516334427	879	risen
15	16	440	93	12	0.9536114720	0.9528218355	0.9505412449	595	risen

criterion									
'log_loss'	max_depth	max_leaf_nodes	min_samples_split	min_samples_leaf	Simple Accuracy	Cross-val Accuracy	Cross-val AUC	Tree Nodes	vs baseline
# 0 base line	default	default	default	default	0.9475723016	0.9478075916	0.9471751616	4853	-
1	16	default	default	default	0.9545257687	0.9530143209	0.9507685696	2237	risen
2	default	440	default	default	0.9592416149	0.9592027256	0.9562811795	879	risen
3	default	default	93	default	0.9544535874	0.9545927010	0.9523938088	821	risen
4	default	default	default	15	0.9539001973	0.9534666615	0.9512695983	1385	risen
5	16	440	default	default	0.9584716809	0.9566137974	0.9538165401	879	risen
6	16	default	93	default	0.9549107358	0.9537746381	0.9514551147	647	risen
7	16	default	default	15	0.9556566094	0.9528603326	0.9503557251	1037	risen
8	default	440	93	default	0.9542851643	0.9555166307	0.9529451321	821	risen
9	default	440	default	15	0.9560415764	0.9554203881	0.9528236309	879	risen
10	default	default	93	15	0.9523122083	0.9537650139	0.9514335579	723	risen
11	16	440	93	default	0.9545979501	0.9537650139	0.9514440097	645	risen
12	default	440	93	15	0.9523122083	0.9539574992	0.9515067186	723	risen
13	16	default	93	15	0.9531302632	0.9527544657	0.9503929606	591	risen
14	16	440	default	15	0.9557769116	0.9541788574	0.9515086762	879	risen
15	16	440	93	15	0.9531302632	0.9527737142	0.9504125576	591	risen

Decision Tree Model Feature Selection (Cathy)

After deciding the best model, we then move on to the feature selection process. To evaluate the importance of features with different criteria, we used the following four method:

1. Removing the features with low variance
2. Univariate feature selection
3. Recursive feature elimination (REF)
4. Tree-based estimators (impurity-based feature importances)

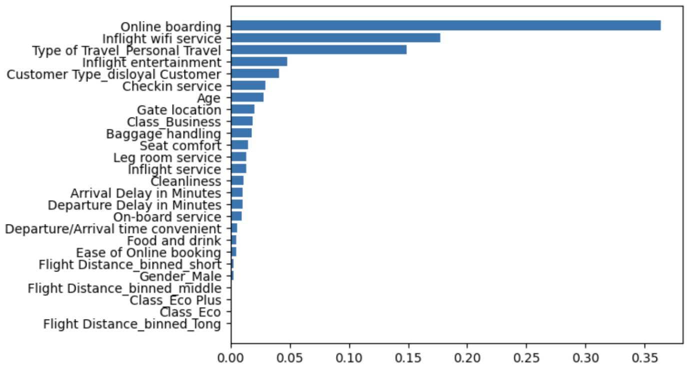
In the tables below, we marked the top 5 important features with yellow background, top 6-10 with orange background, and top 11-20 with purple background. The results of method 1 shown in Table 1 is more inconsistent with the others. Also, the variance for more than ten features are all 1.000010, which makes the discrimination lower, so we focused on method 2-4. In summary, the top 5 mutual important features are as below:

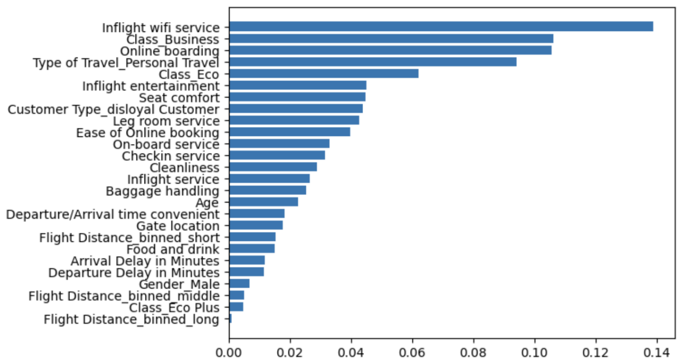
1. Inflight wifi service
2. Online boarding
3. Type of Travel_Personal Travel
4. Class_Business
5. Inflight entertainment

We then looped the number of features n of the above feature selection methods to get the optimal n for the selected model. The model performs better when trained with the top 17 features from univariate feature selection, the top 14 from RFE, and the top 19 from tree-based estimators. The best performance of methods 3 and 4 are better than without dropping any feature, and performs the best with the last method. Thus, we conclude that the best decision tree classifier is that with {‘criterion’: ‘entropy’, ‘max_leaf_nodes’: 440, ‘random_state’: 42}, and trained with top 19 features ['Inflight wifi service', 'Online boarding', 'Type of Travel_Personal Travel', 'Class_Business', 'Class_Eco', 'Inflight entertainment', 'Customer Type_disloyal Customer', 'Seat comfort', 'Ease of Online booking', 'On-board service', 'Baggage handling', 'Cleanliness', 'Leg room service', 'Checkin service', 'Inflight service', 'Age', 'Departure/Arrival time convenient', 'Gate location', 'Flight Distance_binned_short'] generated from tree-based estimators.

	Table 1. Finding important features with different methods	Remove low variance	Univariate (SelectKBest)	Recursive feature elimination		Tree-based estimator (impurity-based)	
#	features	Variance	ranking	feature_importances_	ranking	feature_importances_	ranking
1	Age	1.00001	20	0.02766251	7	0.02272672	16
2	Inflight wifi service	1.00001	11	0.17726932	2	0.13894325	1
3	Departure/Arrival time convenient	1.00001	23	0.00556886	18	0.01837680	17
4	Ease of Online booking	1.00001	18	0.00476826	20	0.03968954	10
5	Gate location	1.00001	26	0.02029665	8	0.01774794	18
6	Food and drink	1.00001	16	0.00487805	19	0.01498315	20
7	Online boarding	1.00001	2	0.36352008	1	0.10575305	3
8	Seat comfort	1.00001	6	0.01509159	11	0.04479739	7
9	Inflight entertainment	1.00001	5	0.04770189	4	0.04500729	6
10	On-board service	1.00001	7	0.00942153	17	0.03314552	11
11	Leg room service	1.00001	8	0.01359945	12	0.04287809	9
12	Baggage handling	1.00001	12	0.01803802	10	0.02536391	15
13	Checkin service	1.00001	14	0.02978305	6	0.03152296	12
14	Inflight service	1.00001	13	0.0133965	13	0.02673017	14
15	Cleanliness	1.00001	9	0.01106076	14	0.02882335	13
16	Departure Delay in Minutes	1.00001	24	0.01003016	16	0.01164295	22
17	Arrival Delay in Minutes	1.00001	22	0.01021186	15	0.01194626	21
18	Gender_Male	0.249947	25	0.00240283	22	0.00687586	23
19	Customer Type_disloyal Customer	0.149308	17	0.04114176	5	0.04395256	8
20	Type of Travel_Personal Travel	0.214044	4	0.14864845	3	0.09412279	4
21	Class_Business	0.249518	1	0.01899995	9	0.10624194	2
22	Class_Eco	0.247491	3	0.00099063	25	0.06213034	5
23	Class_Eco Plus	0.066923	21	0.00111475	24	0.00485738	25
24	Flight Distance_binned_long	0.051043	19	0.00041524	26	0.00110834	26
25	Flight Distance_binned_middle	0.164451	15	0.00143294	23	0.00521081	24
26	Flight Distance_binned_short	0.193102	10	0.00255493	21	0.01542164	19

	Table 2. Comparison between different features selection methods	Variance		Univariate (SelectKBest)	Recursive feature elimination	Tree-based estimator (impurity-based)	Overall	
#	features	top 10	top 17	top 17 (optimal n)	top 14 (optimal n)	top 19 (optimal n)	top 5	
1	Age							
2	Inflight wifi service							
3	Departure/Arrival time convenient							
4	Ease of Online booking							
5	Gate location							
6	Food and drink							
7	Online boarding							
8	Seat comfort							
9	Inflight entertainment							
10	On-board service							
11	Leg room service							
12	Baggage handling							
13	Checkin service							
14	Inflight service							
15	Cleanliness							
16	Departure Delay in Minutes							
17	Arrival Delay in Minutes							
18	Gender_Male							
19	Customer Type_disloyal Customer							
20	Type of Travel_Personal Travel							
21	Class_Business							
22	Class_Eco							
23	Class_Eco Plus							
24	Flight Distance_binned_long							
25	Flight Distance_binned_middle							
26	Flight Distance_binned_short							
								baseline
	Simple Accuracy	0.91872	0.93484	0.957292719	0.959819065	0.960492758	0.92690	0.9592416149
	Cross-val accuracy	0.91887	0.93486	0.958298044	0.959857176	0.960232522	0.92634	0.9592027256
	Cross-val AUC	0.91558	0.93099	0.955077925	0.956743665	0.957409310	0.92300	0.9562811795
	Tree Nodes	879	879	879	879	879	553	879
	vs baseline	dropped	dropped	dropped	risen	risen	dropped	-
	FPR (choose lowest)	0.080669	0.05477	0.0283508739	0.027130886	0.029045261	0.07034	0.0296166444





Logistic Regression

Logistic Regression Model Hyperparameter Tuning

For the logistic regression model, we still use the 04 dataset with train test split and random state = 42 to train, tune and test.

We first tried to use GridSearchCV to find the best parameter choice or value for each of the three parameters: C value (inverse of model complexity), solvers (‘lbfgs’, ‘liblinear’, ‘newton-cg’, ‘newton-cholesky’, ‘sag’, ‘saga’), and penalty (‘l1’, ‘l2’, ‘elasticnet’, None). The results show that for solvers and penalty the best parameters are the default parameters, and the best C value is 110.

Table 1: Logistic fitting 04 sub train and sub test									
	random_state	C	solvers	penalties	Simple Accuracy	Cross validation accuracy	Cross validation AUC	FP	FN
untuned	42	Default	Default = lbfgs	Default = l2	0.875433081	0.874951878	0.926735842		
Grid 1	42	100	Default = lbfgs	Default = l2	0.875304761	0.874910631	0.926539065		
Grid 2	42	110	Default = lbfgs	Default = l2	0.87517644	0.874910631	0.926778686		
		Default = 1.0							

Looking at comparison of accuracy scores along C values between the whole train and real test set, there should be no overfitting problem as well.

Logistic Regression Model Feature Selection

We have used RFE (Recursive Feature Selection) and SFS (Sequential Feature Selection) to select important features for the logistic regression model, then try to select 15 and 10 features to fit into the model again, with selection results summarized below:

	Dataset 04	Decision tree fitting 4b1	Decision tree fitting 4b2
0	Age		dropped
1	Inflight wifi service		
2	Departure/Arrival time convenient	dropped	
3	Ease of Online booking		dropped
4	Gate location	dropped	
5	Food and drink	dropped	dropped
6	Online boarding		
7	Seat comfort	dropped	dropped
8	Inflight entertainment	dropped	
9	On-board service		
10	Leg room service		dropped
11	Baggage handling	dropped	dropped
12	Checkin service		dropped
13	Inflight service		dropped
14	Cleanliness		
15	Departure Delay in Minutes		dropped
16	Arrival Delay in Minutes		dropped
17	Gender_Male	dropped	dropped
18	Customer Type_disloyal Customer		
19	Type of Travel_Personal Travel		
20	Class_Business	dropped	
21	Class_Eco		dropped
22	Class_Eco Plus		dropped
	Flight Distance_binned_middle	dropped	dropped
	Flight Distance_binned_short	dropped	dropped
		15 features left	10 features left

Decision Tree feature selection	Feature Coefficient		RFE 15	RFE 10	RFE 5	RFE 3	SFS 15	SFS 10	SFS 5	SFS 3
Features\Model	Untuned	Tuned								
Age										
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										
Checkin service				✓	✓					
Inflight service				✓						
Cleanliness				✓			✓	✓	✓	
Departure Delay in Minutes				✓						
Arrival Delay in Minutes				✓		✓				
Gender_Male										
Customer Type_disloyal Customer				✓	✓		✓	✓		
Type of Travel_Personal Travel				✓	✓		✓	✓	✓	✓
Class_Business							✓	✓		
Class_Eco										
Class_Eco Plus										
Flight Distance_binned_middle										
Flight Distance_binned_short										
							(set it as 15 and still only 10 selected			

Table 2: Logistic fitting 4b1 whole train to train and real test set to test

		C	solvers	penalties	Simple Accuracy	Cross validation accuracy	Cross validation AUC	FP	FN
untuned	42	Default	Default = lbfgs	Default = l2	0.872651678	0.87362373	0.925701227		
Grid 1	42	100	Default = lbfgs	Default = l2	0.872112719	0.87368148	0.925753644		
Grid 2	42	110	Default = lbfgs	Default = l2	0.872151217	0.87369110	0.925753789		
		Default = 1.0							

Table 3: Logistic fitting 4b2 whole

ummm accuracy keeps dropping tho

		C	solvers	penalties	Simple Accuracy	Cross validation accuracy	Cross validation AUC	FP	FN
untuned	42	Default	Default = lbfgs	Default = l2	0.862411457	0.86613605	0.920981999		
Grid 1	42	100	Default = lbfgs	Default = l2	0.872112719	0.87368148	0.925753644		
Grid 2	42	110	Default = lbfgs	Default = l2	0.872151217	0.87369110	0.925753789		
		Default = 1.0							

as the 4b2 dataset does not increase model accuracy, then lets use the 4a1 dataset which increases model accruacy of decision tree to see if it works

ble 3: Logistic fitting 4a1 whole

		C	solvers	penalties	Simple Accuracy	Cross validation accuracy	Cross validation AUC	FP	FN
untuned	42	Default	Default = lbfgs	Default = l2	0.8704958423	0.8727190484	0.9248905749		
Grid 1	42	100	Default = lbfgs	Default = l2	0.8705343394	0.8726805513	0.9248922905		
Grid 2	42	110	Default = lbfgs	Default = l2	0.8705343394	0.8726805513	0.9248922859		

Logistic regression

features	coefficient	rank
age	-0.1208	
flight distance	-0.0117	
inflight wifi service	0.3922	2
departure time convenient	-0.1238	
ease of online booking	-0.1399	
gate location	0.0273	12
food and drink	-0.0244	
online boarding	0.6109	1
seat comfort	0.0656	9
inflight entertainment	0.0651	10
on-board service	0.3005	4
leg room service	0.2517	5
baggage handling	0.1328	7
checkin service	0.3215	3
inflight service	0.1212	8
cleanliness	0.2198	6
departure delay in min	-0.1681	
gender_male	0.0382	11
Customer Type_disloyal Customer	-2.0141	
Type of Travel_Personal Travel	-2.7038	
Class_Eco	-0.7322	
Class_Eco Plus	-0.8358	

	AUC		Confusion matrix:	
accuracy for model trained by top 10 highest coefficient features	AUC	Confusion matrix:	49763	9116
0.81	0.869		10565	34460
accuracy for model trained by all features	AUC	Confusion matrix:	53261	5618
0.87	0.926		7386	37639

Also, we tried to select 10 most powerful features and retrained the model (file name: Logistic_regression_top_10_coefficient.ipynb) again.
However, when we look at the performance after dropping features for all feature selection methods, all scores are lower. We deduce that it may be due to the fact that the logistic regression model already has penalty for features with low importance, so further dropping features would not improve model performance.

Naïve Bayes

Data preparation methods without any features selection process are summarized at right:

The accuracies are generally low before carrying out any feature selection method. (file: NB_model_label_encoding.ipynb)

Data preparation method	Multinomial Naïve Bayes	Multinomial Naïve Bayes
id	N/A	N/A
gender	one-hot encoding	label encoding
customer type	one-hot encoding	label encoding
age	binned	minmax and then binned
type of travel	one-hot encoding	label encoding
class	one-hot encoding	label encoding
flight distance	binned	minmax and then binned
inflight wifi service	one-hot encoding	originally it is ordinal features, no data cleaning process
departure time convenient	one-hot encoding	originally it is ordinal features, no data cleaning process
ease of online booking	one-hot encoding	originally it is ordinal features, no data cleaning process
gate location	one-hot encoding	originally it is ordinal features, no data cleaning process
food and drink	one-hot encoding	originally it is ordinal features, no data cleaning process
online boarding	one-hot encoding	originally it is ordinal features, no data cleaning process
seat comfort	one-hot encoding	originally it is ordinal features, no data cleaning process
inflight entertainment	one-hot encoding	originally it is ordinal features, no data cleaning process
on-board service	one-hot encoding	originally it is ordinal features, no data cleaning process
leg room service	one-hot encoding	originally it is ordinal features, no data cleaning process
baggage handling	one-hot encoding	originally it is ordinal features, no data cleaning process
checkin service	one-hot encoding	originally it is ordinal features, no data cleaning process
inflight service	one-hot encoding	originally it is ordinal features, no data cleaning process
cleanliness	one-hot encoding	originally it is ordinal features, no data cleaning process
departure delay in minutes	drop	drop
arrival delay in minutes	drop	drop
accuracy	0.51	0.64

Naïve Bayes Model Feature selection

Chi-squared method is adopted to select some features and fit into the Naive Bayes model. Chi-squared method measures the features which are highly correlated/informative to the target variable. Since we have no idea the optimal number of features selected, I fitted the model by using the top 4-11 most important features and recorded the accuracy. If we use the top 5 most important features to fit the model, it generated the highest accuracy 0.8. The models trained by selected features generally have higher accuracy. (file: Data_preparation_feature_selection.ipynb, NB_model_label_encoding_feature_selection.ipynb)

	Chi-squared method	
	NB model feature selection	scores
0	id	423704.8
1	gender	7.9
2	customer type	2990
3	type of travel	14445.7
4	class	13606.9
5	inflight wifi servcie	5422
6	departure time convenient	210.3
7	ease of online booking	2174.5
8	gate location	0.03
9	food and drink	2527.9
10	online boarding	14762
11	seat comfort	6419.3
12	inflight entertainment	8711.2
13	on-board service	5299.3
14	leg room service	5262
15	baggage handling	2448.8
16	checkin service	2808.4
17	inflight service	2362.9
18	cleanliness	5071.4
19	age-binned	411
20	flight distance-binned	8993.5

no. of features	accuracy
11 features	67.65
10	69.53
9	71.6
8	73.9
7	76.36
6	79.14
5	80.39
4	79.9

Feature ID is included because it does matter to the accuracy of the model, if we eliminate ID, the accuracy will become lower.

Model Evaluation

	Model	AUC
0	mnb_test	0.850887
1	mnb	0.852471

Random Forest Model

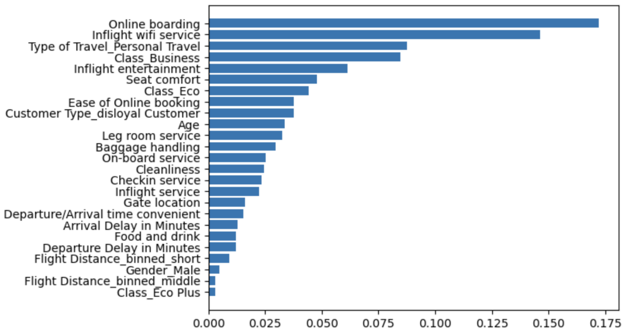
We trained a random forest model on our dataset using the scikit-learn library in Python. The model was trained using “data preparation sorted 4”.

Baseline Result

Before applying any optimizations, we evaluated the performance of the baseline model. The model achieved a cross validation accuracy of 0.96247.

Random Forest Model Feature Selection

In this random forest model, we used rf.feature_importances_ which is calculated using the Mean Decrease Impurity method. For each feature, the importance score is calculated as the sum of the reduction in impurity (measured by Gini impurity or entropy) overall decision trees in the Random Forest, weighted by the number of samples that were split on that feature. We’ve trained models with reduced features, however, the CV accuracy is lower than the baseline result.

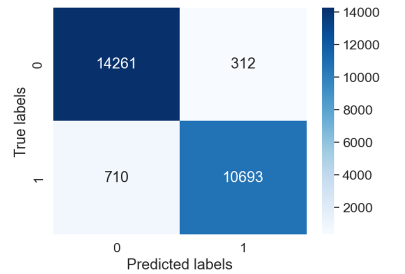


Random Forest Hyperparameter Tuning

To improve the performance of the model, we performed a grid search cross-validation to find the best hyperparameters. We searched over a range of values for the number of estimators and the minimum samples split of the trees. The grid search resulted in the following optimal hyperparameters:

Best hyperparameters: {'min_samples_split': 2, 'n_estimators': 500}

After training the model with the optimal hyperparameters, we evaluated its performance on the training set and test set. The model achieved a cross validation accuracy of 0.96300 which is an increase of 0.053% to the baseline result. For the test set, the model that achieved the confusion matrix is shown at right.



4.Performance Evaluation

Each of our best models	Cross Validation Accuracy	FPR	FNR
Decision tree model (Yanni)	0.9584	0.03027	0.04651
Decision tree model (Cathy)	0.9602	0.02905	0.04746
Logistic regression model (Yanni)	0.8750	0.12970	0.12160
Logistic regression model (Wendy)	0.8705	0.12987	0.12179
Random Forest (Steven)	0.9630	0.02835	0.04743
Multinomial Naive Bayes (Wendy)	0.8039	0.16781	0.23175

Our performance evaluation mainly uses accuracy scores as our dataset does not have class imbalance problems and accuracy does not distort model performance.
We have used simple accuracy, cross validation (cv) accuracy and cv AUC score as the performance evaluation tool when tuning hyperparameters as only one tool may not give the full picture of model performance. During the tuning process, train test split has been used to train and test models, with the train to test proportion being 0.5.
For model selection, we consider two aspects of model performance. The first aspect is the prediction accuracy, where we use cross validation accuracy to choose the best model because the Naive Bayes model does not do classification based on a decision threshold and cannot produce AUC scores.

We also consider cost benefit analysis when selecting the best model. We have identified the cost of False Positive as the cost of losing one customer per each FP point and the cost of False Negative as the cost of retaining unsatisfied customers. We believe that the cost of FP should be significantly higher than that of FN, which is close to zero, so we would choose a model with lower FPR. Therefore, the best classifier is Random Forest Classifier.

5.Conclusion

Our goal in data mining is to identify the determined factors of customer satisfaction in the airline industry and create a reliable binary predictive model. By pinpointing the most important features, airlines can allocate their resources, such as facilities, equipment, labor force, and time spent, more effectively to improve their services. With an accurate predictive model, airlines can quickly respond to customer feedback and take remedial actions if needed. It will increase customer retention rate, moreover, improve the efficiency of operations, allowing airlines to invest more in sustainability. After analyzing the data using four classifiers, we have concluded that the top three important features are "Online boarding", "Inflight wifi service", and "Type of Travel_Personal Travel".

	Feature Selection			Feature selection method
	1	2	3	
Decision tree model (Yanni)	Online boarding	Inflight wifi service	Type of Travel_Personal Travel	Decision tree feature importance
Decision tree model (Cathy)	Inflight wifi service	Online boarding	Type of Travel_Personal Travel	remove low variance + univariate feature selection+ RFE + tree-based estimator
Logistic regression model (Yanni)	Online boarding	Inflight wifi service	Type of Travel_Personal Travel	SFS and RFE
Logistic regression model (Wendy)	Online boarding	Inflight wifi service	Checkin service	Largest coefficient
Random Forest (Steven)	Online boarding	Inflight wifi service	Type of Travel_Personal Travel	Random Forest feature importance (mean decrease impurity)
Multinomial Naive Bayes (Wendy)	ID	Online boarding	Type of Travel	Chi-squared method

In the future, we could explore the effectiveness of new ML models such as Gradient Boosting, Support Vector Machines (SVM), and Neural Networks. These models have shown promise in various applications and may provide superior performance for predicting customer satisfaction in the airline industry. Conducting a cost-benefit analysis would also be valuable. This analysis would assess the costs associated with developing and deploying predictive models against the potential benefits of increased customer loyalty and satisfaction. The results of this analysis could inform decision-making regarding investment in predictive models for improving customer satisfaction.
Potential problems that may arise if an airline company sticks to the conclusions we draw right now is focusing too much on improving the important features we pointed out, ignoring other features. For example, as we now see cleanliness and food and drink as less important features, their quality of cleanliness and food may decline subsequently. Therefore, we suggest airlines to do further ML and other research work to provide services with high and balanced quality, improving their reliability in providing business.