ECON-UB 232 Data Bootcamp

Airline Passenger Satisfaction: A Classification Approach

Derico Dehnielsen (dd3022) & Hakjoon Lee (hl3257)

Introduction

Air travel is one of the important industries, and airline passenger satisfaction is the main key to the success of airlines. Customer satisfaction is an essential indicator of the performance of the airline industry and can have a huge impact on the brand reputation and revenue of an airline. Understanding what features of passenger satisfaction is more valuable for airlines to provide a better customer experience, retain existing customers, and attract new ones. The dataset contains information on passenger's surveys, such as demographics, flight purpose, flight class, flight distance, and their satisfaction ratings on various variables of the flight factors. Even we, as international students, have taken flight many times using different airlines. Our satisfaction also changes depending on various features of the airlines. By examining the relationship between passenger satisfaction and the various attributes of travel, we focus on contributing to the understanding of passenger preferences. Therefore, our primary goal for this project is to train different models of classification algorithms and build one that would give the most accurate results in predicting passengers' satisfaction from the explanatory variables available in the dataset.

Data Description

We used the dataset "Airline Passenger Satisfaction " from Kaggle for the project.

- **Information of Data**

Gender: Gender of the passengers (Female, Male)

Customer Type: The customer type (Loyal customer, disloyal customer)

Age: The actual age of the passengers

Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel)

Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus)

Flight distance: The flight distance of this journey

Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5)

Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient

Ease of Online booking: Satisfaction level of online booking

Gate location: Satisfaction level of Gate location

Food and drink: Satisfaction level of Food and drink

Online boarding: Satisfaction level of online boarding

Seat comfort: Satisfaction level of Seat comfort

Inflight entertainment: Satisfaction level of inflight entertainment

On-board service: Satisfaction level of On-board service

Leg room service: Satisfaction level of Leg room service

Baggage handling: Satisfaction level of baggage handling

Check-in service: Satisfaction level of Check-in service

Inflight service: Satisfaction level of inflight service

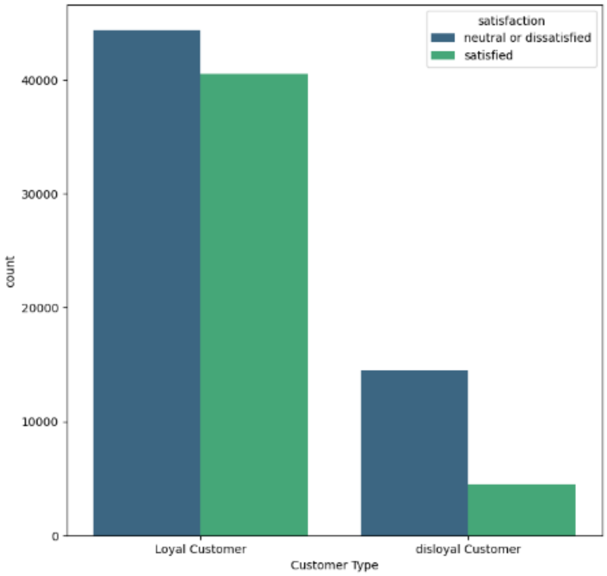
Cleanliness: Satisfaction level of Cleanliness

Departure Delay in Minutes: Minutes delayed when departure

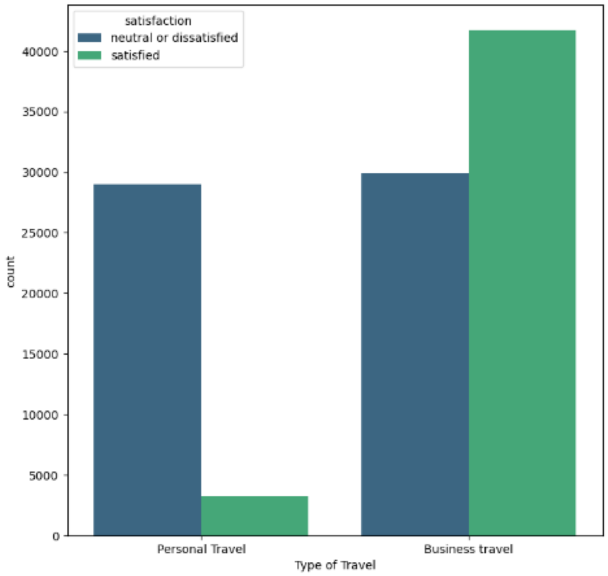
Arrival Delay in Minutes: Minutes delayed when Arrival

Satisfaction: Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

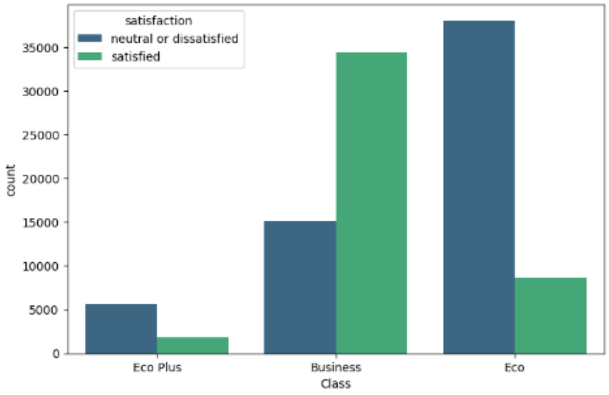
Exploratory Data Analysis/Visualization



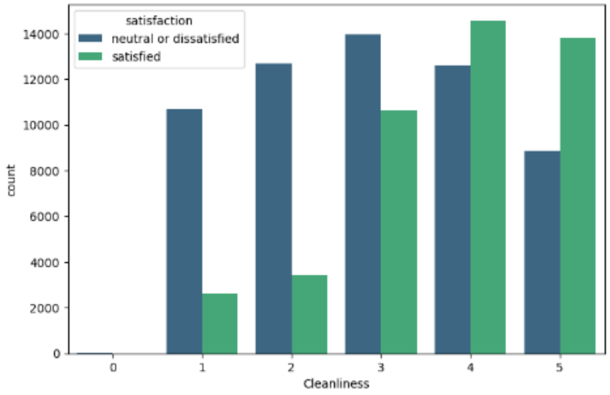
<Customer Type>



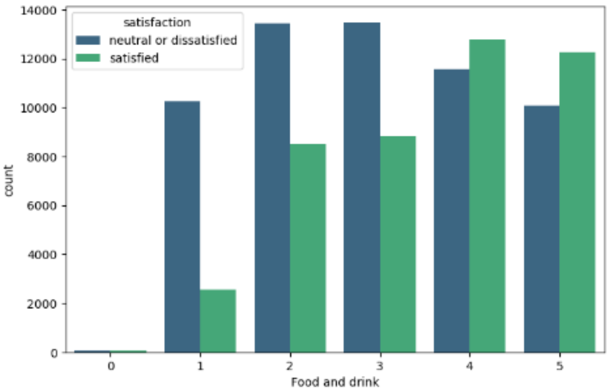
<Type of Travel>

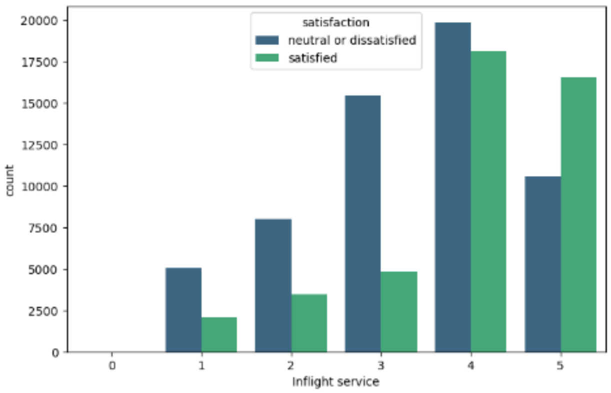


<Class>



<Cleanliness>

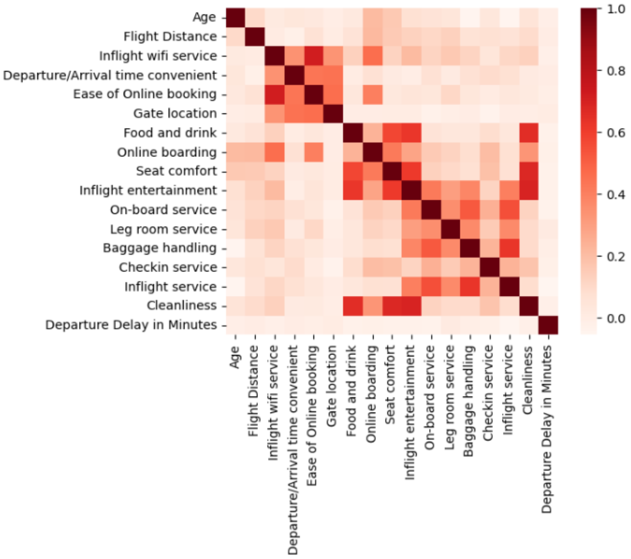




<Food and drink>

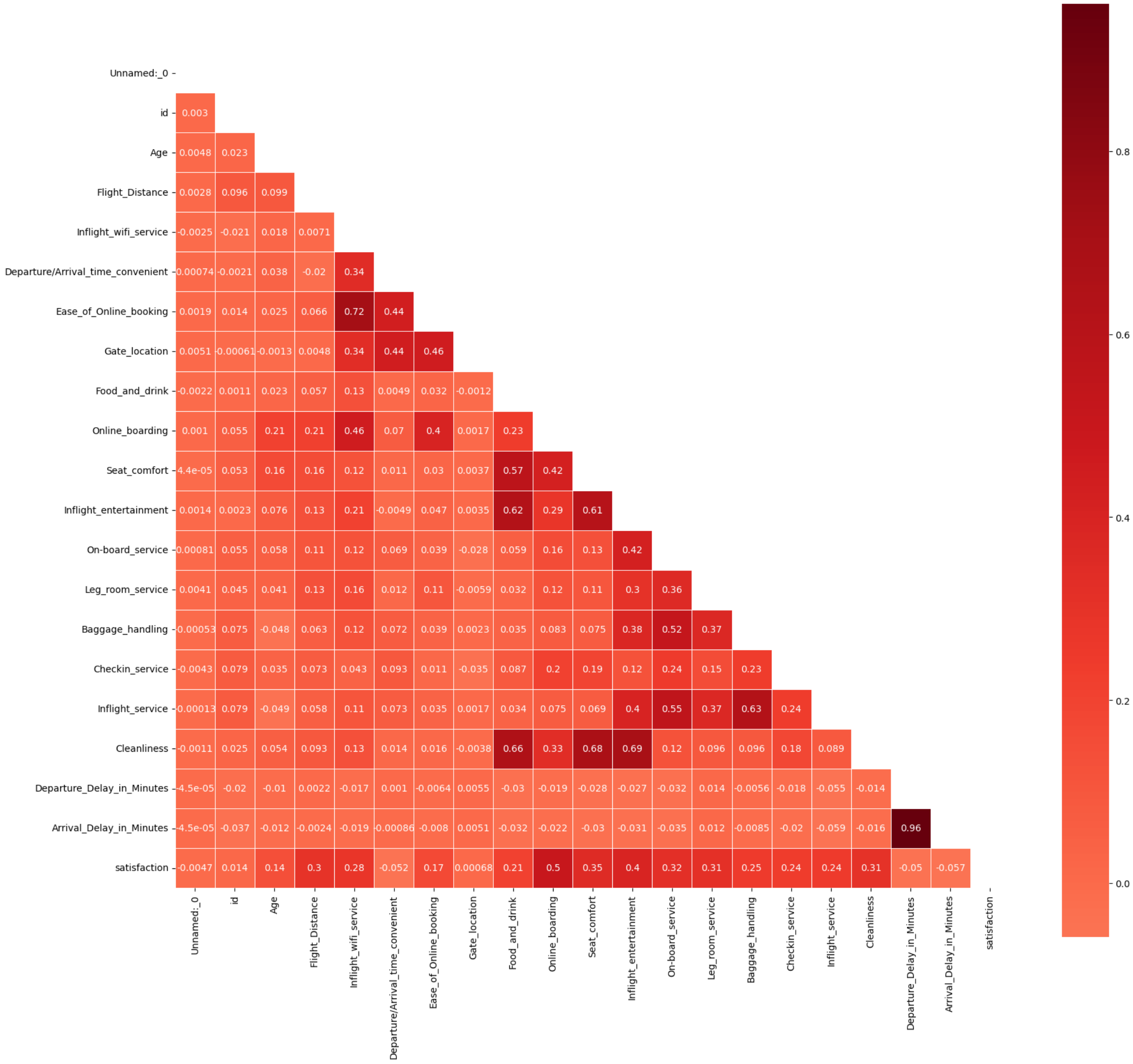
a<Inflight Service>

To begin with, we transformed the original dataset, whose process is described in more detail in the data processing part below, and some figures are visualized to be analyzed before we focus on training various models. From the graphs, we assume that the proportion of loyal customers who are satisfied are larger than the proportion of disloyal customers. The passengers for business travel tend to be more satisfied than those who are for personal travel. The passengers in business class were more likely to be satisfied than those who were in Eco or Eco plus. And the passengers who gave 4 or 5 ratings on "Cleanliness" were satisfied with the airline.



<Matrix1. Correlation matrix between categorical variables>

From Matrix 1, we can easily notice strong correlation between “Inflight wifi service & Ease of online booking”, “Food and drink & Cleanliness ”, “Seat comfort cleanliness”, “Inflight entertainment & Cleanliness” and etc.



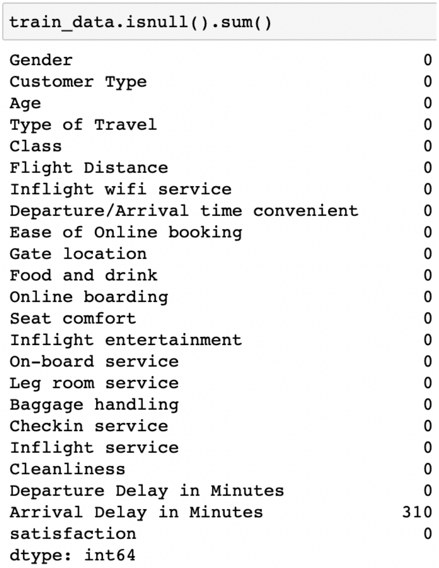
	Unnamed:_0	id	Age	Flight_Distance	Inflight_wifi_service	Departure/Arrival_time_convenient	Ease_of_Online_booking	Gate_location	Food_and_drink	Online_boarding	Seat_comfort	Inflight_entertainment	On-board_service	Leg_room_service	Baggage_handling	Checkin_service	Inflight_service	Cleanliness	Departure_Delay_in_Minutes	Arrival_Delay_in_Minutes	satisfaction
Unnamed:_0																					
id	0.003																				
Age	0.0048	0.023																			
Flight_Distance	0.0028	0.096	0.099																		
Inflight_wifi_service	-0.0025	-0.021	0.018	0.0071																	
Departure/Arrival_time_convenient	0.00074	-0.0021	0.038	-0.02	0.34																
Ease_of_Online_booking	0.0019	0.014	0.025	0.066	0.72	0.44															
Gate_location	0.0051	-0.00061	-0.0013	0.0048	0.34	0.44	0.46														
Food_and_drink	-0.0022	0.0011	0.023	0.057	0.13	0.0049	0.032	-0.0012													
Online_boarding	0.001	0.055	0.21	0.21	0.46	0.07	0.4	0.0017	0.23												
Seat_comfort	4.4e-05	0.053	0.16	0.16	0.12	0.011	0.03	0.0037	0.57	0.42											
Inflight_entertainment	0.0014	0.0023	0.076	0.13	0.21	-0.0049	0.047	0.0035	0.62	0.29	0.61										
On-board_service	0.00081	0.055	0.058	0.11	0.12	0.069	0.039	-0.028	0.059	0.16	0.13	0.42									
Leg_room_service	0.0041	0.045	0.041	0.13	0.16	0.012	0.11	-0.0059	0.032	0.12	0.11	0.3	0.36								
Baggage_handling	0.00053	0.075	-0.048	0.063	0.12	0.072	0.039	0.0023	0.035	0.083	0.075	0.38	0.52	0.37							
Checkin_service	-0.0043	0.079	0.035	0.073	0.043	0.093	0.011	-0.035	0.087	0.2	0.19	0.12	0.24	0.15	0.23						
Inflight_service	0.00013	0.079	-0.049	0.058	0.11	0.073	0.035	0.0017	0.034	0.075	0.069	0.4	0.55	0.37	0.63	0.24					
Cleanliness	-0.0011	0.025	0.054	0.093	0.13	0.014	0.016	-0.0038	0.66	0.33	0.68	0.69	0.12	0.096	0.096	0.18	0.089				
Departure_Delay_in_Minutes	4.5e-05	-0.02	-0.01	0.0022	-0.017	0.001	-0.0064	0.0055	-0.03	-0.019	-0.028	-0.027	-0.032	0.014	-0.0056	-0.018	-0.055	-0.014			
Arrival_Delay_in_Minutes	4.5e-05	-0.037	-0.012	-0.0024	-0.019	-0.00086	-0.008	0.0051	-0.032	-0.022	-0.03	-0.031	-0.035	0.012	-0.0085	-0.02	-0.059	-0.016	0.96		
satisfaction	-0.0047	0.014	0.14	0.3	0.28	-0.052	0.17	0.00068	0.21	0.5	0.35	0.4	0.32	0.31	0.25	0.24	0.24	0.31	-0.05	-0.057	

<Matrix 2. Correlation between each categorical variable and Satisfaction>

To understand the relationship between the variables and the passenger satisfaction, we draw a combined correlation matrix. Since we are predicting passenger satisfaction, computing correlation between each categorical variable and satisfaction is more effective compared to Matrix1. However, it is clear that there are limited conditions to predict corretly passenger satisfaction from these tools. The prediction for the passenger satisfaction needs to be computed considering all the variables for the accuracy. So, we are set to train different classification models and then pick and build upon the best model to achieve our goal.

Data Preprocessing

One problem we found in the dataset is null values that appear in one of the columns; Arrival delay in minutes. Although the portion that is null is quite small, about 3%, we decided to treat these observations so they do not go to waste (see Figure 3). We filled the null values with the average delay in minutes which comes out to be 15 minutes.



```
train_data.isnull().sum()
```

```
Gender                                 0
Customer Type                          0
Age                                    0
Type of Travel                         0
Class                                  0
Flight Distance                        0
Inflight wifi service                  0
Departure/Arrival time convenient      0
Ease of Online booking                 0
Gate location                          0
Food and drink                         0
Online boarding                        0
Seat comfort                           0
Inflight entertainment                 0
On-board service                       0
Leg room service                       0
Baggage handling                       0
Checkin service                        0
Inflight service                       0
Cleanliness                            0
Departure Delay in Minutes             0
Arrival Delay in Minutes             310
satisfaction                           0
dtype: int64
```

Figure 3. Number of null observations per column

For the numerical data in the dataset, we decided to scale the data using the standard scaler. This is because the range of values on different variables varies a lot. For instance, flight distance is mostly in the thousands while age and other satisfaction level variables are not.

Scaling the data makes the numbers closer to each other, which later can help the model to understand the data better and assign more accurate weights to each variable.

The original dataset also contains a lot of categorical data that we have to manipulate, which includes 'Gender', 'Customer Type', 'Type of Travel' , 'Class', and 'satisfaction' variables. If we do not manipulate the dataset, its trained model would not be as accurate since we are not able to put numerical value to the variables. There are two possible ways to treat categorical variables; one is to assign a number to each category in each variable, and another is to assign one and zero whenever we see that the category is seen in the observation. We decided to use the latter option, meaning that we have to have all the categories set as columns too. Another reason why we choose the second option is that not all categories have a linear relationship with one another. For instance, passengers' seat class, whether it is economy class, economy plus class, or business class, the change from one class to another might not be linear. So, we think that it is more fair if each category has an assigned weight individually. We preprocessed the original data with OneHotEncoder from scikit-learn and after transforming the data, we have a total of 26 explanatory variable columns.

Models

We trained a wide range of classification models. Our strategy is to train the model on a wide range of algorithms with no hyperparameter tuning and then, based on the accuracy we got, we will optimize three algorithms with best accuracy. Finally, we will compile these classifier algorithms into one voting classifier algorithm from which we can further manipulate the decision threshold. The list of classification algorithms and why we include them in this list are as follows:

1. Logistic Regression: One of the most widely used and a traditional yet efficient classification model.
2. SGDClassifier: We can benefit from its efficiency in fitting large scale datasets.
3. K-Neighbors Classifier: This is a non-parametric algorithm and it is particularly useful for classification problems where the decision boundary is highly irregular or nonlinear.
4. Decision Tree Classifier: This algorithm is easy to understand and interpret, making it a useful tool for exploring the relationship between features and the target variable.
5. Random Forest Classifier: This is an ensemble learning algorithm that combines multiple decision trees to improve performance and reduce overfitting. It is particularly useful for high-dimensional datasets with many features.
6. Linear Discriminant Analysis: This is a statistical algorithm that works well for problems with multiple classes. It is particularly useful for problems where the data is normally distributed and the classes are well-separated.
7. Naive Bayes Classifier: This is a probabilistic algorithm that works well for problems with many features and relatively small datasets. It is particularly useful for text classification problems.
8. Support Vector Machine: This is a powerful algorithm that can handle high-dimensional datasets and nonlinear decision boundaries. It is particularly useful for problems with a small number of features and a large number of observations.
9. LGBM Classifier: This is a fast and scalable gradient boosting algorithm that works well for large datasets with many features.

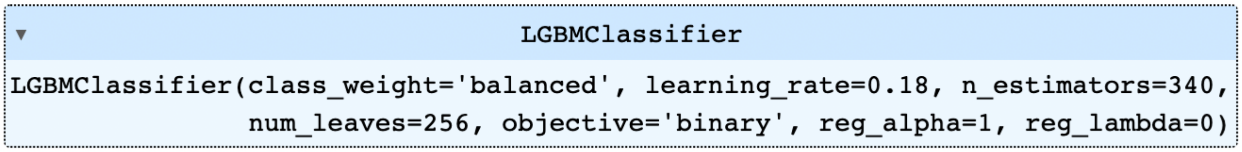
10. XGBoost Classifier: This is another gradient boosting algorithm that is particularly useful for problems with imbalanced datasets or noisy data. It is also highly customizable, with many hyperparameters that can be tuned to optimize performance.

Results and Interpretation

After training these models on the training data without any hyperparameter tuning, these are the accuracy score (how many correct predictions out of all instances) and the standard deviation of the score that we got from the cross validation process.

Model	**Accuracy Score**	**Standard Deviation**
Logistic Regression	0.875	0.003
Stochastic Gradient Descent	0.872	0.004
K-Neighbors Classifier	0.928	0.002
Decision Tree Classifier	0.944	0.002
Random Forest Classifier	0.961	0.001
Linear Discriminant Analysis	0.871	0.003
Naive-Bayes Classifier	0.849	0.003
Support Vector Machine	0.952	0.001
Light GBM Classifier	0.962	0.001
XGBoost Classifier	0.961	0.0

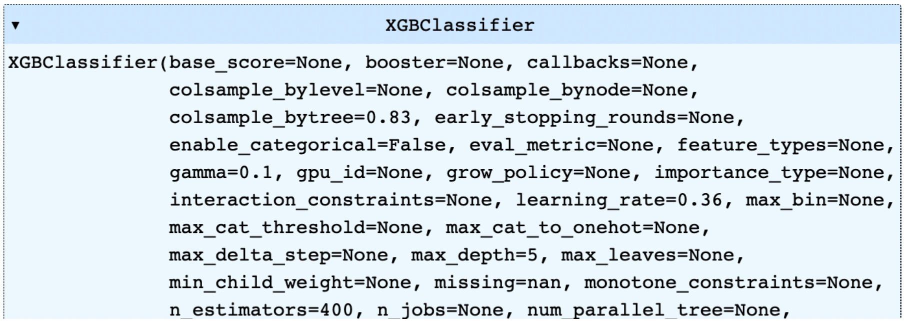
After getting all the results, the top three models with the highest accuracy score came out to be Light GBM Classifier, XGBoost Classifier, and the random forest classifier. So the next step is to optimize the hyperparameters with the training data so that it has a better result. First, for Light GBM Classifier, the hyperparameters that we want to modify are *max_depth_xgb, gamma_xgb, learning_rate_xgb, n_estimators_xgb, colsample_bytree_xgb, reg_alpha, and reg_lambda*. By assigning an array of possible values into these hyperparameters and run it through the RandomizedSearchCV, we can get the best performing hyperparameter for Light GBM Classifier on our training dataset based on the accuracy score. Figure 3 shows the final hyperparameters for the Light GBM Classifier that we will use.



```
▾                              LGBMClassifier
LGBMClassifier(class_weight='balanced', learning_rate=0.18, n_estimators=340,
               num_leaves=256, objective='binary', reg_alpha=1, reg_lambda=0)
```

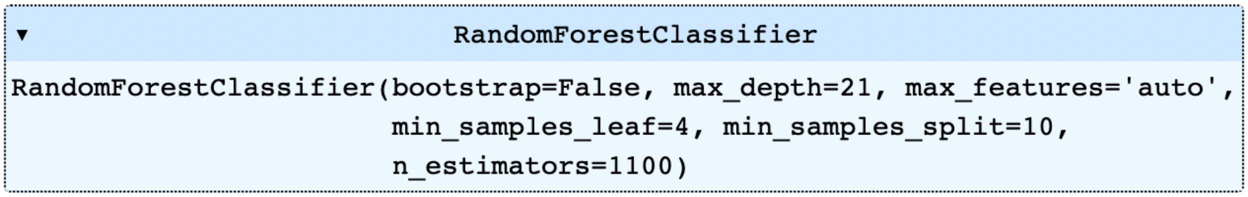
Figure 3. Light GBM Classifier Hyperparameters

Secondly, with similar steps, we also optimized the XGBoost Classifier algorithm and the random forest algorithm. Figure 4 and 5 below shows the final model based on the training data it was trained on.



```
▾                              XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=0.83, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=0.1, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.36, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=5, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=400, n_jobs=None, num_parallel_tree=None,
```

Figure 4. XGBoost Classifier Hyperparameters



```
▼                            RandomForestClassifier
RandomForestClassifier(bootstrap=False, max_depth=21, max_features='auto',
                       min_samples_leaf=4, min_samples_split=10,
                       n_estimators=1100)
```

Figure 5. Random Forest Classifier Hyperparameters

The next step is to combine all these three algorithms into one model called the voting algorithm. Voting algorithm is a soft voting/majority rule algorithm based on the models we input into the algorithm. For instance, if we are comparing three models, where each model gives a predicted probability of classifying the instance in class 1 of 0.8, 0.9, and 0.2 respectively, the average of these three predicted probability is 0.63 (see Figure 6). So the voting classifier will classify that instance as class 1.

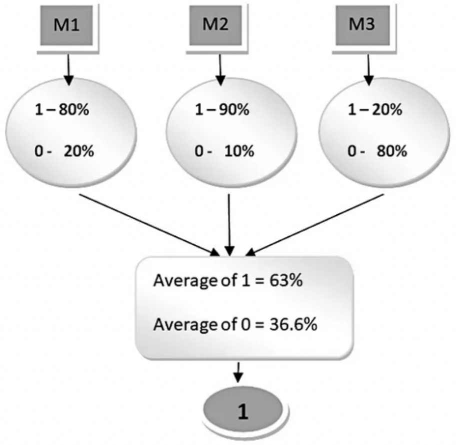


Figure 6. Soft Voting Classifier (Source: Medium, Satyam Kumar)

With the test dataset that we have not touched since the beginning of this project, we tested all four of our algorithms with the test data to see how it performs and whether the voting classifier does a better job relative to the other models.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Light GBM Classifier	0.9608	0.9647	0.9453	0.9549
XGBoost Classifier	0.9598	0.9637	0.9440	0.9537
Random Forest Classifier	0.9618	0.9697	0.9426	0.9560
Voting Classifier	0.9620	0.9689	0.9438	0.9562

From these matrices that we used to compare the different models, we found out that the voting classifier did the best job in terms of its accuracy and F1 score (highlighted in green). We included the F1 score here because it gives a more balanced representation of the precision and recall score. In addition to that, we also plotted the ROC curve as seen in the figure below as well as the AUC score for the prediction which came out to be 0.9947. This indicates that our model can almost perfectly distinguish airlines passengers' satisfaction based on the variables as presented in this dataset.

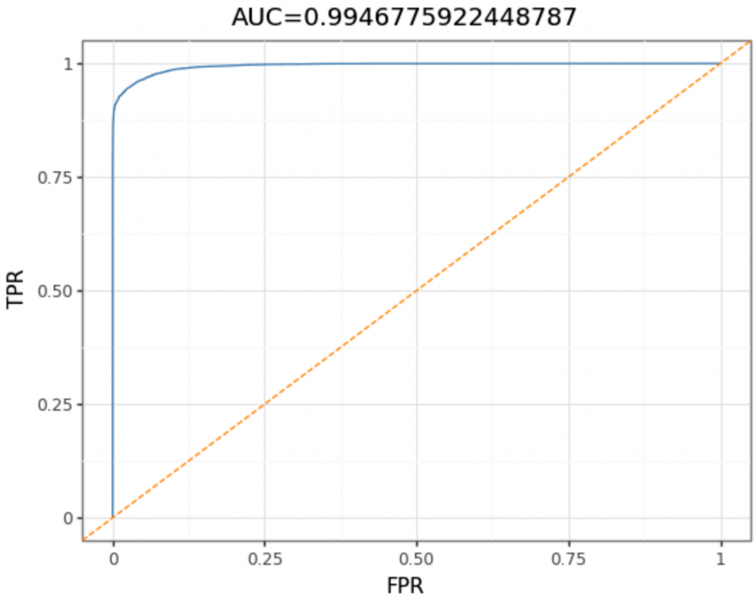


Figure 7. ROC Curve for Voting Classification

Conclusion

The many different explanatory variables, their quantifiable characteristics, and the large size makes this dataset an ideal and interesting one for machine learning exploration, specifically classification task. Our voting classification model which compiles our pre-modelled Light GBM Classifier, XGBoost Classifier, and Random Forest Classifier, outperformed the other models when trained individually. Although it is only a slight improvement from the individual model, the voting classifier algorithm was able to achieve 96.2% accuracy and 0.956 F1-score. This method has the clear advantage that all analyses can be conducted so long as you have access to customers' surveys regarding their flight experiences. While we would assume that doing the same analysis with deep learning would make a better result, the result we got is more than good enough considering the time and power needed to train and test a neural network model with a large size dataset. So for future research possibilities, it would be interesting to look into some deep learning algorithms to predict airline passengers' satisfaction. Even though it might not be a significant difference or improvement to our current model, neural network algorithms are more robust to outliers which may come out in surveys like this dataset.

Reference

Kumar, Satyam. "Use Voting Classifier to Improve the Performance of Your ML Model." *Medium*, 1 Nov. 2021, towardsdatascience.com/use-voting-classifier-to-improve-the-performance-of-your-ml-model-805345f9de0e.