Airline Passenger Satisfaction and Analysis

STAT 432 Final Project Report

Team Members:
Mayank Agarwal (mayanka5)
Yash Bajaj (yashpb2)

Introduction:

The Aviation industry is growing rapidly as airline companies look to retain and acquire new customers. Usually, price is used as the deciding competitive component. In recent years passengers have started focusing more on an airline's services. In this analysis we will find out how the services affect a passenger's satisfaction. To ensure profitability, aviation companies can offer quality services to improve customer satisfaction and create a loyal customer base. This will be rewarding for the airlines in the long run. Airline management can employ a customer driven approach to determine which services are desired the most by passengers.

In our dataset the target variable is "satisfaction" which has two classes: Satisfied (1), Neutral or dissatisfied (0). This is a classic binary classification problem. It is a supervised machine learning task.

Main goals of our analysis:

1. To find the factors that affect a passenger's satisfaction.
2. To predict passenger's satisfaction

Summary:

In our analysis, we first perform exploratory data analysis to visualize our data to find some useful insights about our data. Then we look to find missing values and outliers in our data and handle them using appropriate methods. Feature selection is done using chi-square test and permutation test to select the top features to use for our classification models. Feature engineering methods like label encoding and data normalization techniques are applied. We make use of six classification models in our analysis: Logistic Regression, Decision Trees, K-Nearest neighbors, Random Forests, Ada Boost and XGBoost. Analysis of these models is based on accuracy and AUC score. The factors that affect a passenger's satisfaction the most are determined by feature importance using the model giving us the highest accuracy. Some other techniques like visualization of decision trees and shap values are investigated to better understand how the classification models are predicting satisfaction classes.

Data Source and Description:

The data was sourced from Kaggle
https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

There are two files training and testing containing 25 columns each.
The U.S. Airline Passenger Satisfaction Dataset describes passenger satisfaction by conducting a survey at the airport. The dataset contains customer's information such as age, gender, type of travel, flight distance, class type, departure, and arrival delay. It also contains 14 airline services of customer satisfaction levels ranging from values 0–5. The airline services for measure satisfaction level categories such as inflight wi-fi service, departure/arrival time convenience, ease of online booking, gate location, food and drink, online boarding, seat comfort, inflight entertainment, on-board service,

leg-room service, baggage handling, check-in service, inflight service, and cleanliness.

Literature Review:

For our project, we look at some research papers to further understand how to approach our problem statement.

Investigating airline passenger satisfaction: Data mining method by Tri Noviantoro, Jen-Peng Huang. This paper employs a feature selection method to determine passenger satisfaction. This paper concludes that (1) Online boarding, (2) inflight Wi-Fi service, (3) baggage handling and (4) inflight entertainment are top features to be focused on for an airline to improve passenger satisfaction. This research paper uses classification algorithms such as decision tree, random forest, gradient boosted tree, K-NN, logistic regression, SVM, Naive Bayes and deep learning. The highest accuracy of 95.42% was achieved with the deep learning model. Decision Trees and random forests each achieved an accuracy of 93.46% and 94.41%. SVM achieved the lowest accuracy of 82.80%. Hence, we use Logistic Regression, Decision Trees, K-Nearest neighbors, Random Forests, Ada Boost and XGBoost for analysis. For evaluating the different model's accuracy, ROC curve and F score are used.

The main approach used in this research paper is data preparation by cleaning the data and modeling the data by feature selection. Then the final predictive model is evaluated with other models using the evaluation metrics.

This research also used a 10-fold cross validation strategy to split training and testing data to reduce the effect of bias and sample variability affecting model performance. It used stratified sampling so that the average response values across the partitions are the same.

A fuzzy segmentation analysis of airline passengers in the U.S. based on service satisfaction by Steven Leon and Juan Carlos Martin. In this research the authors used fuzzy clustering methods and ANOVA analysis to find insights into airline passenger satisfaction. The main conclusion we can draw is that an airline can provide high quality services to gain competitive advantage. The results showed that passengers were more satisfied with functional quality (how a service is delivered) than technical quality (what services are delivered). Thus, instead of just increasing the number of services, airlines can leverage the quality and standard of services provided to gain the maximum benefit.

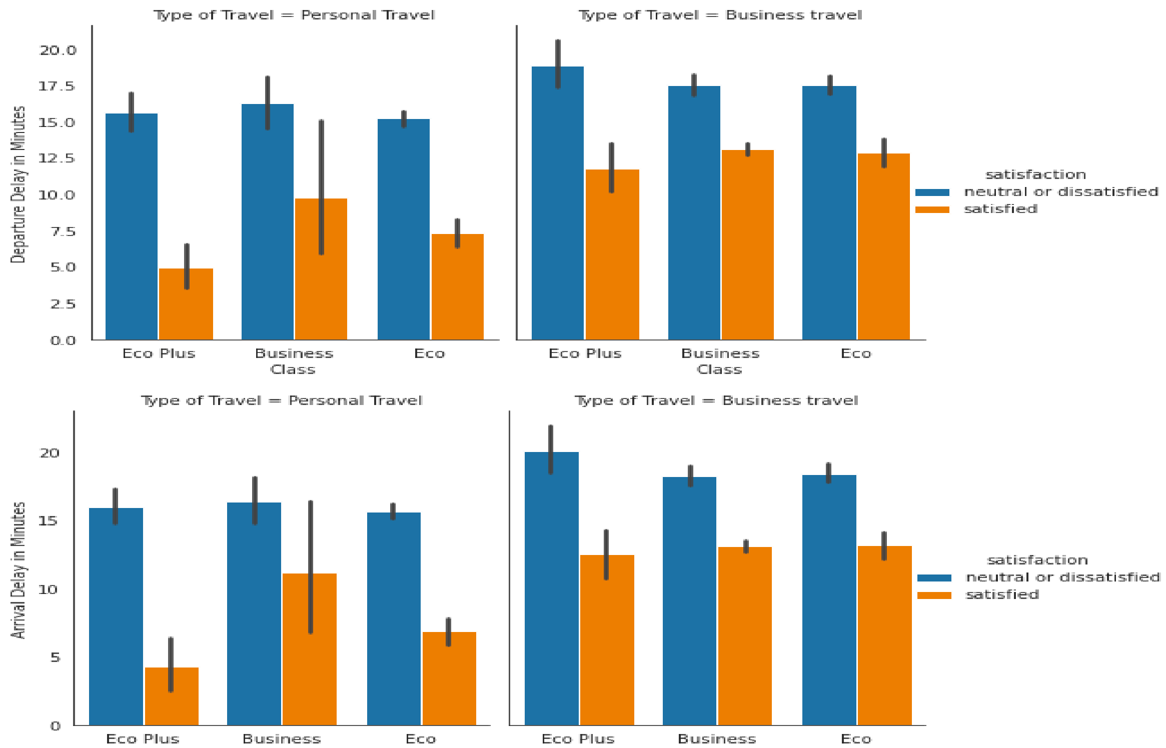
Airline alliance survival analysis: typology, strategy and duration by Sveinn Vidar Gudmundsson and Dawna L Rhoades. This paper explores alliance formation in the international airline industry to determine their survival and duration. Passengers have raised their service quality expectations. For the long-term survival of the airlines, satisfaction of the passengers is critical. Hence in this analysis we are exploring the full-service airline business model and not the low-cost carrier business model.

Generally, sentiment analysis (analyze text, comment) is used to find satisfaction of passengers. In our analysis we are focusing on a machine learning approach to leverage data from surveys to determine the airline services to be prioritized.

Exploratory Data Analysis:

	Neutral or dissatisfied (0)	Satisfied (1)	Total
Train	58879	45025	103904
Test	14573	11403	25976

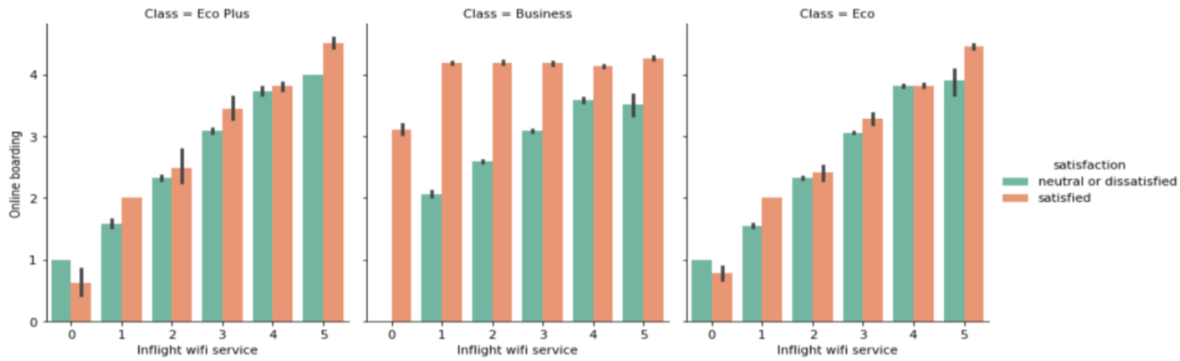
The above Frequency table shows that our data is quite balanced for both training and test datasets.



For personal travel (especially Economy Plus and Economy), the no. of dissatisfied passengers is high when arrival delay in minutes is high. As observed, the level of dissatisfied or neutral passengers is always more than the satisfied passengers whenever there's a delay in arrival or departure time. For business travel in the business class category the no. of satisfied customers is quite high for a long-distance flight. For other combinations, there is an almost equal distribution of satisfied and dissatisfied passengers. For the cleanliness predictor, we observe that as cleanliness increases, the satisfaction of passengers increases.

For both Males and Females, we find that no. of 'Neutral or Dissatisfied' customers are more as compared to 'Satisfied' customers. The same logic applies to 'Loyal' and 'Disloyal' Customers. Also, the no. of Loyal customers is quite higher than 'Disloyal' customers. The early age passengers i.e., 8-38 are more neutral or dissatisfied as compared to satisfied. Passengers aged 39-60 have higher satisfaction levels and then the trend for dissatisfied or neutral is again higher for the remaining higher age group.

For business class, it is observed that all gate locations have a higher no. of dissatisfied passengers when baggage handling is not done perfectly well (rating <= 4). For Economy Plus with gate location 1 and for Economy with gate location 2, when the baggage is handled in a mediocre way (rating in the range 2.0 - 4.0), passengers remain dissatisfied. The maximum no. of satisfied passengers belongs to the category of 4 and 5 rating givers for quality of food and drinks. Below the rating of 4, passengers are mostly dissatisfied. As seat comfort increases, customer satisfaction increases as well.



We see that all 3 classes of passengers are satisfied as to the ease of online boarding increases as well as the presence of an Inflight Wi-Fi service

Correlation Heat Map:

From correlation heat map in appendix, we can observe the following:

The factors affecting the passenger's satisfaction the most are:
Online Boarding, Type of travel, Inflight entertainment, Class, Seat comfort, On-board service, Leg room service, Cleanliness, Inflight Wi-Fi service, Baggage Handling

The features which affect negatively on a passenger's satisfaction are:
Arrival Delay in minutes, Departure Delay in minutes, Departure/Arrival time convenient, Gender, Gate location

Also, from the above correlation plot we see that some of the factors are correlated with one other (multicollinearity):
Arrival delay in minutes and Departure delay in minutes (0.74), Ease on online booking and Inflight Wi-Fi service (0.71), Cleanliness with Inflight entertainment (0.68), Cleanliness with Seat comfort (0.67), Cleanliness with Food and drink (0.65), Inflight service and baggage handling (0.63), Inflight entertainment with Food and drink (0.61)

Feature Engineering:

We have performed label encoding to maintain hierarchical structure of some of our features. This method is beneficial as it provides us with data with low cardinalities. Less dimensions of our features. We have used KNN Imputer to fill missing values for our feature 'Arrival Delay in Minutes' in both train and test data. The main advantage of this method is that we are trying to approximate missing values with points closest to them based on other features. We have outliers in 2 features namely 'Arrival Delay in Minutes' and 'Departure Delay in Minutes' but these are not outliers as most of the values are 0 as there was no arrival/delay. This means that we cannot remove these points as they will be able to determine the level of dissatisfaction among the passengers which can then be used to predict our response.

$$x_{scaled} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

We use MinMax scaler to normalize our data. This is useful when you have a very big quantity in one of the features and have 1's and 0's in another feature. Standardizing our data will help us bring both features to the same scale so that one feature is not assumed to be more important than the other. Another advantage is that this method is very useful in case there are outliers in data. This method also works well with features that don't have perfectly normal distributions.

Feature Selection:

We use Chi-square test and permutation test and select the most important 10 features that will help predict satisfaction for classification models.

The Formula for Chi Square Is

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where:

c = degrees of freedom

O = observed value(s)

E = expected value(s)

A **chi-square test** is used in statistics to test the independence of two events. Given the data of two variables, we can get observed count O and expected count E. Chi-Square measures how expected count E and observed count O deviates from each other.

Weight	Feature
0.1498 ± 0.0028	Inflight wifi service
0.1449 ± 0.0015	Type of Travel
0.0579 ± 0.0007	Customer Type
0.0458 ± 0.0003	Online boarding
0.0297 ± 0.0007	Checkin service
0.0272 ± 0.0006	Baggage handling
0.0245 ± 0.0011	Seat comfort
0.0226 ± 0.0005	Class
0.0206 ± 0.0003	Inflight service
0.0194 ± 0.0006	Cleanliness
0.0124 ± 0.0006	Leg room service
0.0123 ± 0.0004	Age
0.0116 ± 0.0003	On-board service
0.0109 ± 0.0002	Flight Distance
0.0092 ± 0.0004	Inflight entertainment
0.0080 ± 0.0005	Arrival Delay in Minutes
0.0056 ± 0.0003	Ease of Online booking
0.0052 ± 0.0002	Gate location
0.0047 ± 0.0003	Departure Delay in Minutes
0.0037 ± 0.0001	Departure/Arrival time convenient
	... 2 more ...

Permutation Test:

This method helps in feature selection for Blackbox models like Ensemble or deep learning models. The steps for the permutation test are -

1)Randomly shuffle the data in the predictor while keeping the values of other predictors constant.
2) Generate new predictions based on the shuffled values and evaluate the quality of your new predictions.
3) Compute the feature importance score by calculating the decrease in the quality of your new predictions relative to your original predictions

Features for Classification Models: 'Type of Travel', 'Class', 'Flight Distance', 'Inflight Wi-Fi service', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'On-board service', 'Leg room service' and 'Cleanliness'**.**

Model Fitting and Hyperparameter Tuning:

For each of our classification models, we define a parameter grid to tune and get the best parameters. GridSearchCV method is used with 10-fold cross validation with 3 repeats. GridSearchCV tries all possibilities of hyperparameters. Hence the main drawback of this method is that it is computationally expensive. The main metric on which we will evaluate our models is **"Accuracy".** We will also look at classification report, confusion matrix, AUC plot.

Logistic Regression:

Logistic regression is a supervised learning algorithm that evaluates the relationship between the dependent variable and one or more independent variables by estimating the probability using a logistic function. The main benefit of using logistic regression is that it is quick and sufficiently accurate but would not be able to find the complex relations between variables. Our main aim is to minimize error to get the best predicted output. We try to minimize the cost function (quadratic function).

Parameters considered for Hyperparameter Tuning:	**Reasoning:**
Penalty: L1, L2, elasticnet **Solver:** newton-cg, lbfgs, liblinear, saga **C:** 100,10,1,0.1,0.01 $Logistic(p) = \frac{1}{1-e^{-p}}$ To minimize the misclassification rate, prediction is y = 1 when p >= 0.5 and y = 0 when p < 0.5	**L2 penalty** removes a small percentage of weights at each iteration. **Saga solver** is a variant of sag (Stochastic Average gradient descent) and it supports L1 penalty making it suitable for larger datasets. It optimizes the sum of a finite number of smooth convex functions. It is faster than other solvers as it incorporates memory of previous gradient values. **C ($1/\lambda$)** is the inverse regularization parameter, and a low value means that regularization strength is high (giving more weight to complexity penalty)

Classification Report

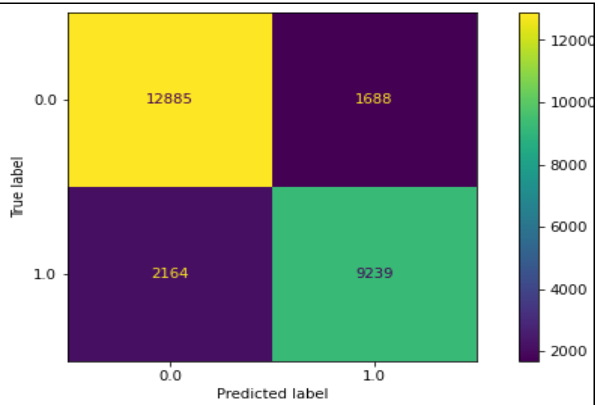
```
              precision    recall  f1-score   support

         0.0     0.8562    0.8842    0.8700     14573
         1.0     0.8455    0.8102    0.8275     11403

    accuracy                         0.8517     25976
   macro avg     0.8509    0.8472    0.8487     25976
weighted avg     0.8515    0.8517    0.8513     25976
```

The main disadvantage of Logistic regression is that it assumes linearity between the dependent and independent variables.

Confusion Matrix



Logistic regression gives Training accuracy of **85.3%** and a testing accuracy of **85.1%.** From the classification report the precision is 0.85 hence our model is correct 85% of the time. The recall is 0.8472 means that these many true positives were found.
ROC Area Under the curve for Logistic model: **0.847** (Appendix)

Decision Trees:

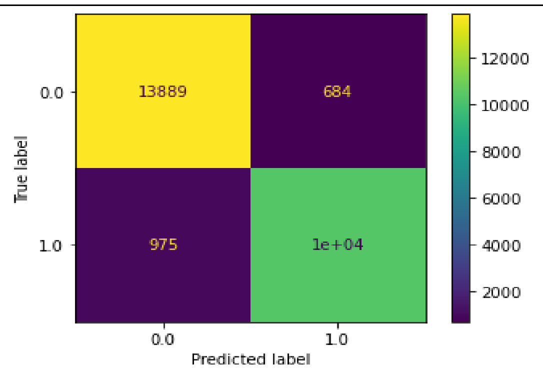
It is a non-parametric supervised learning method used for classification and regression. It infers decision rules from the features to predict the value of a target variable. The main advantage is that it is easy to understand and interpret. We can also visualize the tree to understand our model.
Sometimes decision trees can create overly complex trees that do not generalize the data well causing overfitting. Hence, we can perform hyperparameter tuning by requiring a minimum number of samples required at a leaf node and setting the maximum depth of the tree.

Hyper-parameter tuning grid with best value -	**Reasoning:**
max_depth: 2,3,5,10,20 **min_samples_leaf:** 5,10,20,50,100 **Criterion:** gini, entropy $Gini = 1 - \sum_{i=1}^{n} p^2(c_i)$ $Entropy = \sum_{i=1}^{n} -p(c_i)log_2(p(c_i))$ where $p(c_i)$ is the probability/percentage of class c_i in a node.	**max_depth** controls the depth of the tree. The deeper we allow our tree to grow, the more complex our model will be. Hence, we tune it to make sure our model does not overfit. **min_samples_leaf** is the minimum number of samples required at a leaf node. This makes sure that the model does not overfit by creating a bunch of small branches for one sample each. **Criterion** parameter defines the function that is used to measure the quality of a split. Entropy is an information gain metric measuring the impurity (uncertainty) in a group of observations.

Classification Report

	precision	recall	f1-score	support
0.0	0.9344	0.9531	0.9436	14573
1.0	0.9384	0.9145	0.9263	11403
accuracy			0.9361	25976
macro avg	0.9364	0.9338	0.9350	25976
weighted avg	0.9362	0.9361	0.9360	25976

Confusion Matrix



Decision Trees gives Training accuracy of **93.5%** and a testing accuracy of **93.6%.** From the classification report the precision is 0.93 hence our model is correct 93% of the time. The recall is 0.93 means that these many true positives were found.
ROC area under the curve for decision tree model: **0.9337** (Appendix)

Random Forests:

It is an ensemble supervised learning method which creates multiple decision trees trained on different parts of the same training set with the main aim of reducing variance. The training algorithm applies the method of aggregating bootstrap, or bagging, to tree learners.

$$Y = y_1, \ldots, y_n \text{ and } X = x_1, \ldots, x_n$$

$$\text{for no. of samples selected } b = 1 \text{ to } B, \text{ unseen samples } x' \text{ and regression tree } f_b$$

$$\hat{f} = \frac{\sum_{b=1}^{B} f_b * (x')}{B}$$

Given a training set X with response Y, bagging repeats the process of randomly selecting a sample with replacement of the training set and fits trees to these samples. The predictions can then be done by averaging the predictions from all sets of regression trees fitted on X.

Hyper-parameter tuning grid with best value -

n_estimators = [10, 100, 1000]
max_features = ['sqrt', 'log2']

Reasoning -

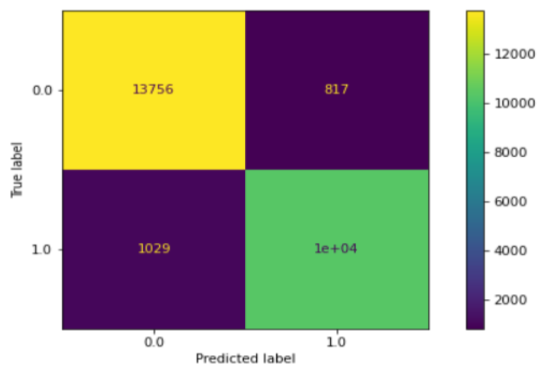
The number of trees (**n_estimators**) that are formed is an important feature for random forest to perform bootstrapping and then averaging the predictions for which we get a value of 100.

max_features are the maximum number of features in an individual tree. Increasing it generally improves model performance, but it decreases the speed of the algorithm. Hence, we choose the optimal number of features. Log2 will take the logarithm of the total number of features in individual run.

Classification Report

	precision	recall	f1-score	support
0.0	0.9304	0.9439	0.9371	14573
1.0	0.9270	0.9098	0.9183	11403
accuracy			0.9289	25976
macro avg	0.9287	0.9268	0.9277	25976
weighted avg	0.9289	0.9289	0.9289	25976

Confusion Matrix



Random Forest gives Training accuracy of **93.46%** and a testing accuracy of **92.89%.**
From the classification report the precision is 0.93 hence our model is correct 93% of the time. The recall is 0.93 means that these many true positives were found.
ROC Area under the curve for Random Forest model: 0.928 (Appendix)

K-Nearest Neighbors:

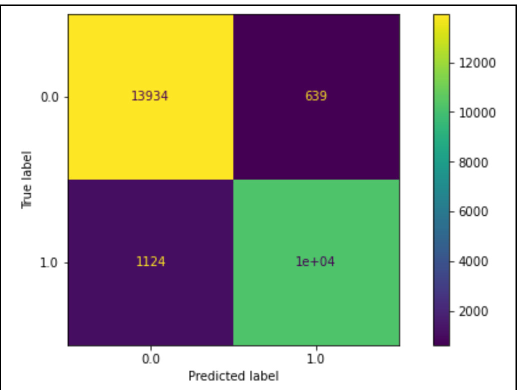
It is a supervised algorithm that compares new data with the nearest neighbor's previous training data. It categorizes a data point according to the classification of its neighbors. The main advantage is that the decision line of the model is flexible and can be non-linear. The parameter k in k-nearest neighbors is the nearest counting neighbor in the majority vote.

Hyper-parameter tuning grid with best value:	**Reasoning:**
n_neighbors = 1,2,3,4,5,6,7,8,9,10 **weights** = uniform, distance **Metric** = euclidean, manhattan, minkowski $d(x, y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$ Formula for calculating Manhattan distance.	When we use uniform **weights** meaning that all points in each neighborhood are weighted equally instead of weighing them by inverse of their distance. **Distance metric** uses distance to find similarities between each element in the dataset. Manhattan distance is a special case of Minkowski distance where only the exponent changes. If the distance is zero then elements are equivalent.

Classification Report

	precision	recall	f1-score	support
0.0	0.9254	0.9562	0.9405	14573
1.0	0.9415	0.9014	0.9210	11403
accuracy			0.9321	25976
macro avg	0.9334	0.9288	0.9308	25976
weighted avg	0.9324	0.9321	0.9319	25976

Confusion Matrix



KNN gives Training accuracy of **93.13%**and a testing accuracy of **93.21%.**
From the classification report the precision is 0.93 hence our model is correct 93% of the time. The recall is 0.92 means that these many true positives were found.
ROC Area under the curve for KNN model: **0.928** (Appendix)

AdaBoost:

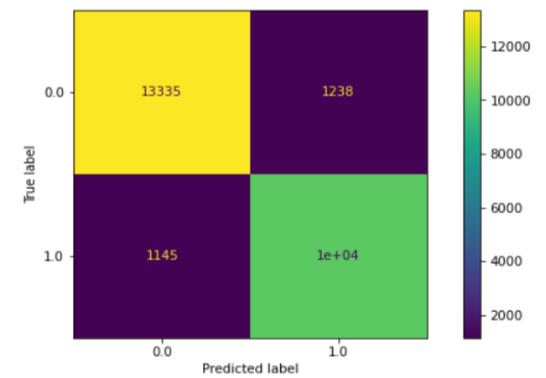
Adaptive Boosting is a very popular boosting technique that aims at combining multiple weak classifiers to build one strong classifier. Here the weak classifier is one that performs better than random guessing, but still performs poorly at designating classes to objects. It is called "best out-of -the-box classifier" as it can be applied on top of any classifier to learn from its shortcomings and propose a more accurate model. The main advantage is that it is easier to use with less need for tweaking parameters. One of the disadvantages of AdaBoost is that it is extremely sensitive to noisy data and outliers. Also, it is much slower than XGBoost.

Hyper-parameter tuning grid with best value -	**Reasoning -**
n_estimators = [10, 50, 100, 500] **learning_rate** = [0.0001, 0.001, 0.01, 0.1, 1.0]	The number of trees (**n_estimators**) that are formed should be high as Ada boost depends on rectifying weak learners and so we get a value of 500. We get the **learning rate** value of 1.0 which determines full contribution and thus it maintains a balance between the contribution of the models and number of trees in the ensemble.

Classification Report

	precision	recall	f1-score	support
0.0	0.9209	0.9150	0.9180	14573
1.0	0.8923	0.8996	0.8959	11403
accuracy			0.9083	25976
macro avg	0.9066	0.9073	0.9070	25976
weighted avg	0.9084	0.9083	0.9083	25976

Confusion Matrix



AdaBoost gives Training accuracy of **91.07%** and a testing accuracy of **90.83%.** From the classification report the precision is 0.9 hence our model is correct 90% of the time. The recall is 0.9 means that these many true positives were found.
ROC Area under the curve for AdaBoost model: **0.9073** (Appendix)

XGBoost:

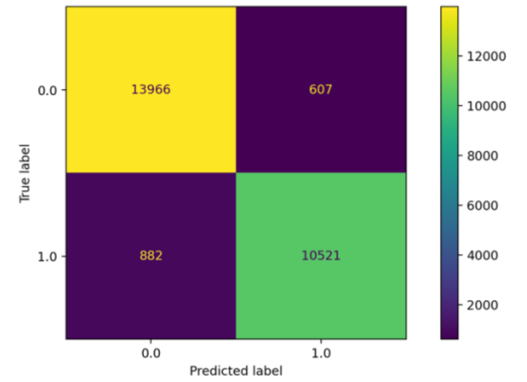
XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. It outperforms other models by regularizing the model, pruning trees using depth first approach, parallelized tree building, handling missing values and providing core computing to obtain results at a faster rate.

Hyper-parameter tuning grid with best value -	**Reasoning -**
gamma:[0,0.1,0.2,0.4,0.8,1.6,3.2,6.4,12.8, 25.6,51.2,102.4, 200] **learning_rate:** [0.01, 0.03, 0.06, 0.1, 0.15, 0.2, 0.25, 0.300000012,0.4, 0.5, 0.6, 0.7] **max_depth:** [5,6,7,8,9,10,11,12,13,14] **n_estimators:** [50,65,80,100,115,130,150] **reg_alpha:** [0,0.1,0.2,0.4,0.8,1.6,3.2,12.8, 25.6,51.2,102.4,200] (L1 regularization) **reg_lambda:** [0,0.1,0.2,0.4,0.8,1.6,3.2,6.4, 12.8,25.6,51.2,102.4,200 (L2 regularization)	In tree-based models, hyper-parameters include maximum depth of the tree, the number of trees to grow, the number of variables to consider when building each tree, the minimum number of samples on a leaf, the fraction of observations used to build a tree and thus based on these hyperparameters we create a grid to get an optimum value to predict our results.

Classification Report

	precision	recall	f1-score	support
0.0	0.94	0.96	0.95	14573
1.0	0.95	0.92	0.93	11403
accuracy			0.94	25976
macro avg	0.94	0.94	0.94	25976
weighted avg	0.94	0.94	0.94	25976

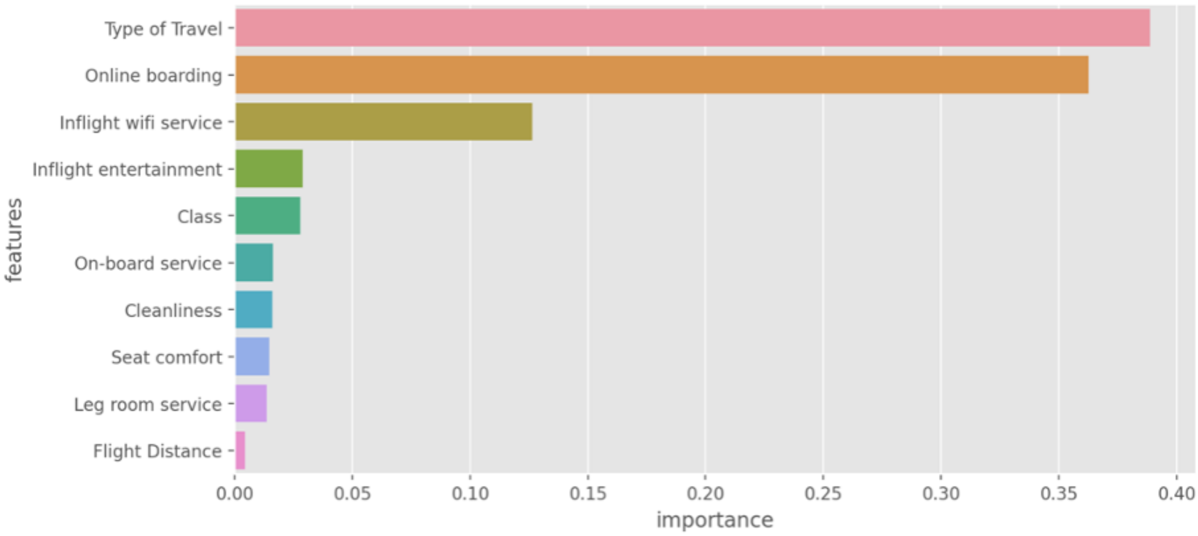
Confusion Matrix



XGBoost gives Training accuracy of **94.31%** and a testing accuracy of **94.27%.** From the classification report the precision is 0.94 hence our model is correct 94% of the time. The recall is 0.94 means that these many true positives were found.
ROC Area under the curve for XGBoost model: **0.9405** (Appendix)

Results and Discussion:

Feature Importance using XGBoost

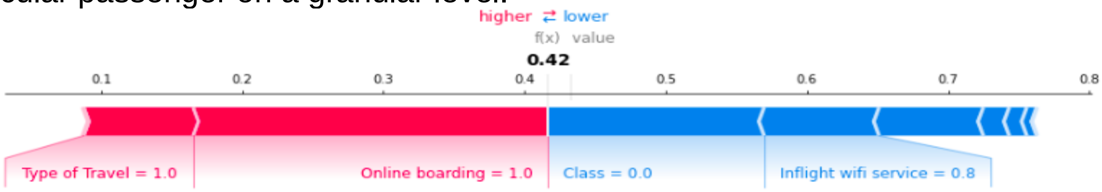


The most important features are **Type of Travel (Business or Personal), Online boarding, Inflight wi-fi service, Inflight entertainment**. All the airlines have very competitive prices and thus we can recommend that airlines should now focus on providing best quality of services as compared to providing a lot of services that are not up to the passenger's expectation. Hence by focusing on these crucial services the airlines can distinguish themselves by creating a loyal customer base which is much more beneficial in the long run

SHAP (Shapley Additive exPlanations) Values:

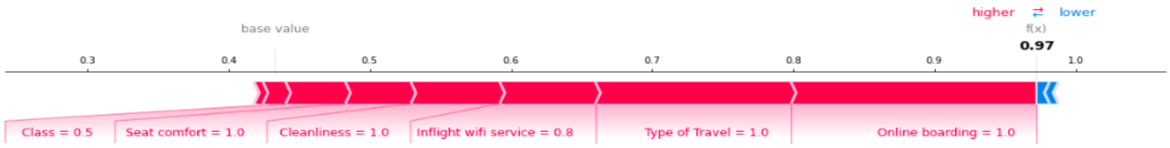
It is a method to explain individual predictions. Shap values are very useful for complex black box models where we lose interpretability of the model. For example, a customer credit approval problem, if a certain customer is denied a credit card and wants to know what factors led to it, we could use shap values to determine the factors for him.

For our problem statement we will try to determine what factors led to satisfaction of a particular passenger on a granular level.



From the above figure, we can see that the **'Type of Travel'** and '**Online boarding rating**' led to a higher probability of 0.42 but the important factors '**Class'** and '**Inflight Wi-Fi service'** stopped our probability from reaching 1**(satisfied**) and thus negatively affected it to reduce the probability by 0.58. Since we get 0.42 as our final probability, our model predicted it as **0**and thus the passenger was **neutral or dissatisfied**. Thus, we can interpret that this **business class customer** had a decent **online boarding** and type of travel by then he was **not satisfied** by class and inflight service

Passenger had poor **online boarding, Inflight experience** making him dissatisfied.



This passenger had very **good online boarding, inflight Wi-Fi service, Cleanliness, Seat comfort and Type of Travel (Business)** making him satisfied. (1)

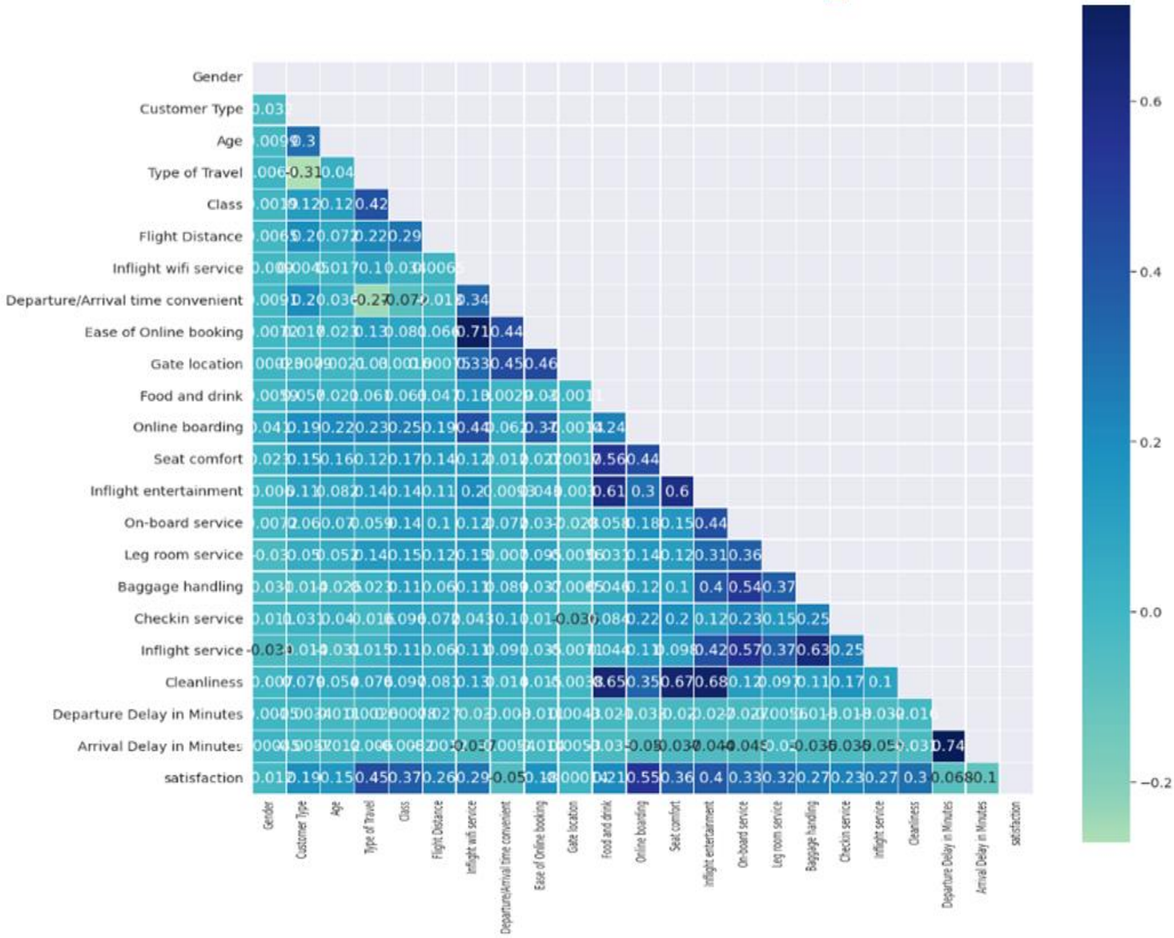
Model	**Training Accuracy (%)**	**Testing Accuracy (%)**
Logistic Regression	85.30	85.17
Decision Trees	93.52	93.61
Random Forest	93.47	92.89
KNN	93.14	93.21
Ada Boost	91.07	90.83
XG Boost	94.31	94.27

Thus, we get the highest training accuracy of **94.31%** and testing accuracy of **94.27%** using the **XGBoost Model**. We use GridSearchCV for calculating the best set of parameters which is computationally very expensive.

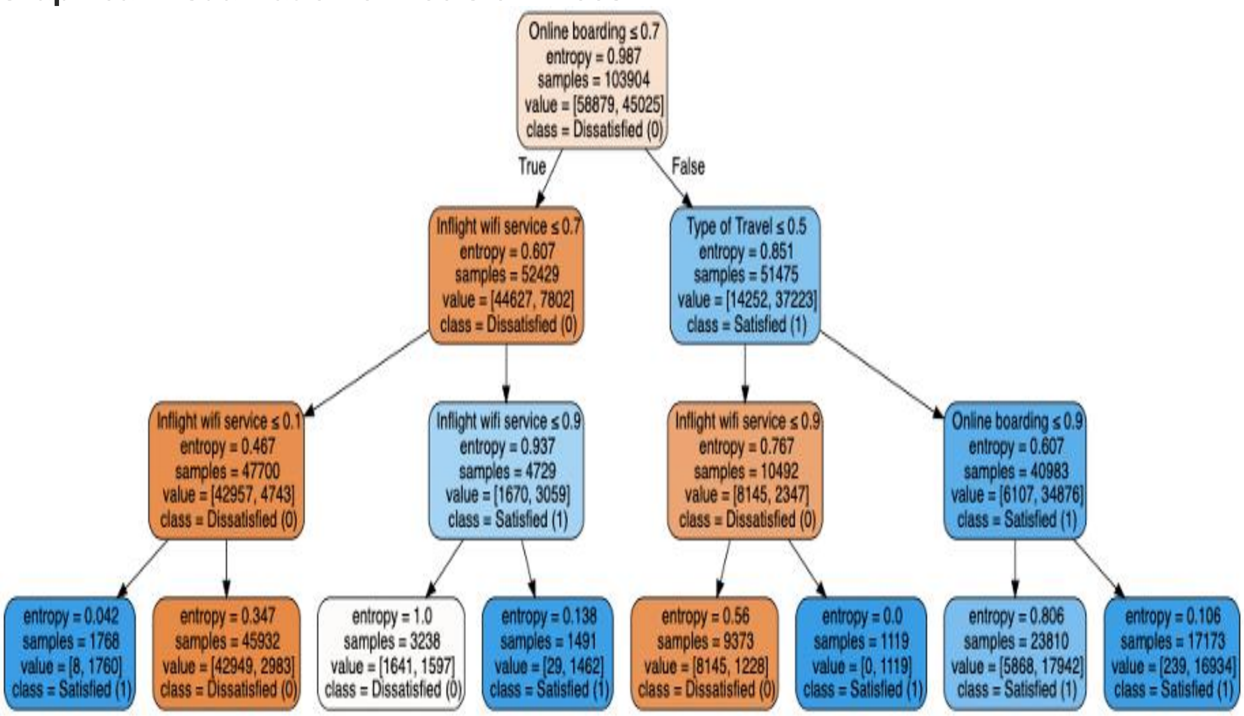
One more important thing to consider is the source of this data as this data could have a **Selection bias** and thus this analysis may only be valid for the US and not applicable worldwide. Also, this data may suffer from **Response bias** as we are unsure about how the survey questionnaire was structured and presented to the passengers. It could have led the passengers to select a particular answer in the survey.

Appendix:

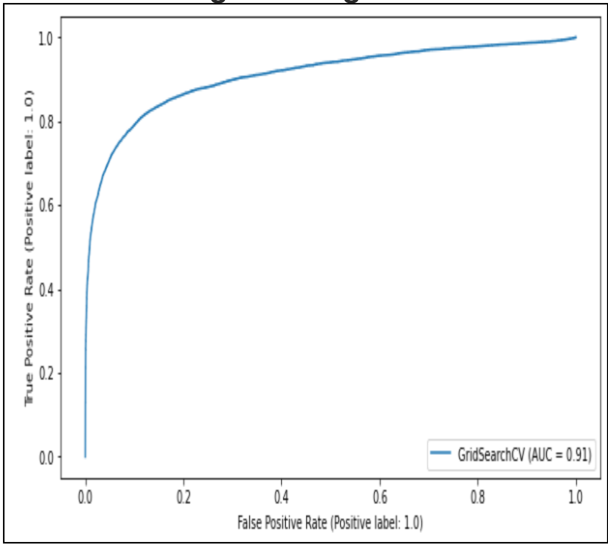
Correlation Heat Map



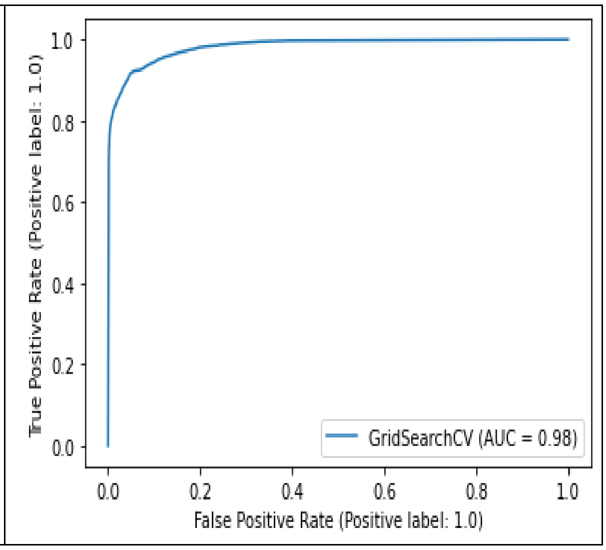
Graphical Visualization of Decision Trees



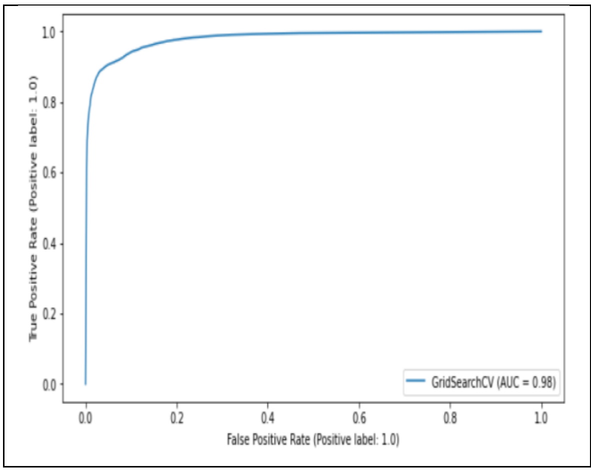
AUC Plot : Logistic Regression



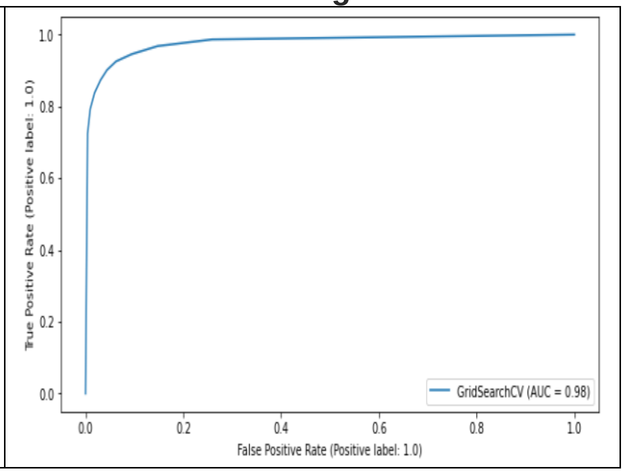
Decision Trees



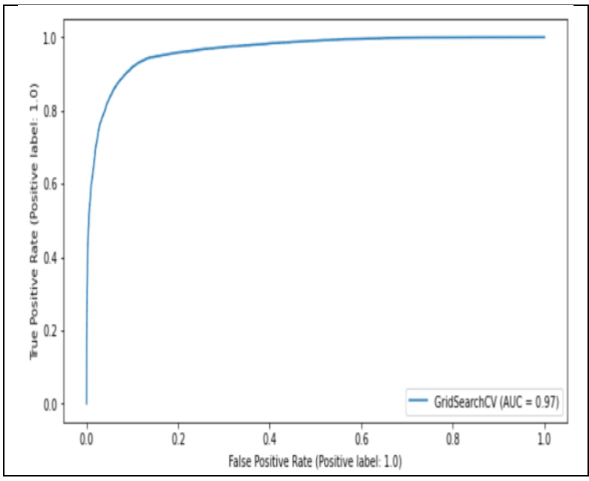
AUC Plot : Random Forests



K-Nearest Neighbors



AUC Plot : AdaBoost



XGBoost