# DATA MINING PROJECT

## CLASSIFICATION AND CLUSTERING OF AIRLINE SATISFACTION DATA

WISSEM REDJEM & PETER EVANS

The airline industry had a revenue of around $800 billion as of 2019, with the vast majority of that revenue accounted for by passenger travel [1]. With almost 300 airlines registered with the International Air Transport Association [2], the field is extremely competitive and the challenge of attracting and retaining customers' business is ever ongoing. Given the gradual nature of the return to air travel in the future as the pandemic subsides, the customer base will be more limited than it was previously, and thus the competition will be even stronger.

As a result, airlines will need to take advantage of every tool available to them to win as much business as possible. Maintaining customers' loyalty to an airline will likely be an essential part of this endeavour. There are many factors that contribute to loyalty, but customer satisfaction has been found to be the one with the largest effect [3, 4]. Thus, finding methods to maximise this satisfaction could be a key part of the success of a given airline.

This report will examine what features of a customer's flight experience contribute most to their satisfaction, using both unsupervised and supervised machine learning techniques. The aim of both of these methods will be to provide valuable business insight, with the former attempting to group customers together and enable effective customer targeting, and the latter producing a model that will be able to predict the satisfaction of an individual customer.

## REVIEW OF RELEVANT LITERATURE

Much of the recent research conducted into customer satisfaction in the airline industry has utilised data available on online review sites, in the case of [5, 6] the site chosen was Skytrax. Both classification and clustering have been undertaken with this online data, with [5] making use of both the review scores submitted to the website and text sentiment analysis of the customers' comments to create a classification decision tree with the Hoeffding algorithm capable of determining if a given passenger was satisfied or not with an accuracy of 84.4%. The unsupervised learning in [6] used the k-means algorithm after PCA dimension reduction to divide the customers into 6 groups, and then produced an estimation model for each cluster in order to compare them. The issue with this approach is that there is not sufficient testing of said estimation models to ensure that the parameters are equivalently performing. Also, the scope of the data was limited in that it was reduced to only airlines within the same 'alliance'.

Similar research has been carried out on data from one-off customer surveys, in [7] for example, passengers on international flights out of Iran were asked to provide their feedback. Here, clustering was again utilised, with a model very similar to the common RFM model (Recency, Frequency, Monetary), assumed to have again been executed with the help of k-means clustering. The aim in this case was not to produce a customer satisfaction classification or regression model but instead to examine in depth which factors contribute to each cluster's satisfaction level, and this was completed using the Kano model. It was determined that the flight crew's appearance and language skills were highly important, as were the customers' cleanliness and flight safety, although these were stated as areas that airlines should focus on improving. However, an on-board visa service was found to be seen as not important.

Additional questionnaire mining was performed in [8] with a decision tree classification method once again used, but this time using the CHAID algorithm which relies on the chi-squared statistic. However, this tree was not tested as a classifier as in [5], instead the tree was just used to cluster the passengers into those with similar satisfaction levels. This did reveal the demographic of least satisfied passengers was men in economy class that were under 24 or over 50.

Finally, we look in slightly more depth at our case study that investigates passenger loyalty through data mining [9]. The motivations for this study were similar to those described in the introduction – to maximise market share in an industry that is struggling. Although the airline industry's struggles in this case were not a result of a global pandemic but instead the decreasing domestic customer base in Taiwan due to its improved rail network. The paper outlines a process to assess a customer's loyalty, summing two scores describing their willingness to fly with the airline again, and to recommend the airline to others. It then builds a decision tree model with the aim of identifying passengers as loyal or not loyal using a broad range of data for each passenger, including demographic information and service and consumption choices. This model was constructed using the C4.5 algorithm, an extension of the ID3 algorithm which uses information gain ratio as its splitting criteria. The study used a 90/10 split of training to testing data, with cross-validation – meaning ten tests were run to determine the model's accuracy each with 90% of the data used to train and 10% used to test, but the 90/10 split was different on each test. This evaluation determined the accuracy to be an average of 82.6% on testing data. The tree divided up the given sample of passengers into 11 distinct groups, 5 loyal and 6 disloyal. Key findings were that the largest group of loyal customers was composed of mostly young married men with tertiary education, and that the main factors differentiating loyal from disloyal customers were "satisfaction with airport service, passenger cabin facilities, and information provision and com plaint resolution services". Such conclusions are useful for airline management to use to target the areas of their service that they can improve with maximum return.

## BUSINESS UNDERSTANDING

As discussed above, satisfaction is a key factor for an airline in retaining their customers. However, understanding how to best secure a customer's satisfaction with their trip is not a simple task. Not only are there a large number of factors that could affect their opinion of the journey (cabin crew, food, flight delays), but a wide demographic of people take flights for many different purposes. Improving the experience is far from a one-size-fits-all approach.

Data mining techniques allow huge volumes of data to be processed to draw useful and insightful patterns. In this report, a large collection of passenger questionnaire surveys will be analysed to help bring some light to the passenger satisfaction problem. The goals with this mining investigation will be to

a) determine distinct groups of airline passengers and what it is that both satisfies, and dissatisfies these groups in order to recognise what changes could be made to please them better

b) find a method to determine the impact on overall satisfaction of any changes made to the flight experience

These objectives can be achieved with two distinct categories of data mining, clustering and classification respectively. The former enables the division of a dataset into similar objects, in this case, the division of passengers into those with similar characteristics and preferences. The latter will enable the construction of a prediction model to determine whether or not a given customer is satisfied, and in doing this enable airlines to see the impact of any changes they make before implementing them.

With regard to resources available, this is an investigation conducted by two individuals over the course of weeks, on personal computers. It is for this reason that the dataset chosen to mine shall be limited in size, as is dictated in the project brief. The mining will be completed in Python, making use of common libraries used for data analysis such as NumPy, Pandas, Scikit-learn and Matplotlib. Python is chosen for its flexibility and wide range of features.

Both classification and clustering parts will test and compare multiple different algorithms to construct models that perform adequately on the given data. For classification, the model will be tested primarily with the accuracy metric, where the objective will be to achieve a similar test level as the studies referenced in the literature review [5, 9], with a percentage in the low 80s. However, it will be important that the model's test accuracy does not drop too significantly from the accuracy achieved on validation data, so that it is known to be robust.

For clustering, the quality of the outcome will mainly be measured by calculating the Silhouette coefficient which assesses the separability of the clusters. A score closer to 1 is preferred as it indicates that the clustering is a good fit for the data.

## DATA UNDERSTANDING AND EXPLORATION

The dataset [10] contains the results of an airline passenger satisfaction survey with several satisfaction factors. Each row represents a single response.

### DATA DESCRIPTION

The dataset is divided into a training dataset of over 100,000 entries and a test dataset of over 25,000 entries. In this project, we only use the test data as it matches the limit set for the rows. It contains a total of 24 features, and a description of each column and its type can be found in Table 1 below:

| Column | Description | Type |
|---|---|---|
| Id | A sequential number identifying each row | Numerical interval |
| Gender | The gender of the passenger: Male or female | Categorical nominal |
| Customer type | The customer type: loyal or disloyal | Categorical nominal |
| Age | The age of the passenger | Numerical ratio |
| Type of Travel | Purpose of the flight: personal or business | Categorical nominal |
| Class | Travel class of the passenger: Business, Eco, or Eco Plus | Categorical ordinal |
| Flight Distance | The flight distance of the passenger's travel | Numerical ratio |
| Inflight Wi-Fi service | Satisfaction level of the inflight Wi-Fi service (0: Not Applicable;1-5) | Numerical ratio |
| Departure/Arrival time convenient | Satisfaction level of Departure/Arrival time convenient | Numerical ratio |
| Ease of Online booking | Satisfaction level of the online booking service | Numerical ratio |
| Gate location | Satisfaction level of the flight's gate location | Numerical ratio |
| Food and drink | Satisfaction level of the food and drink service | Numerical ratio |
| Online boarding | Satisfaction level of the online boarding service | Numerical ratio |
| Seat comfort | Satisfaction level of the seat comfort | Numerical ratio |
| Inflight entertainment | Satisfaction level of the inflight entrainment | Numerical ratio |
| On-board service | Satisfaction level of the on-board service | Numerical ratio |
| Leg room service | Satisfaction level of the leg room service | Numerical ratio |
| Baggage handling | Satisfaction level of the baggage handling | Numerical ratio |
| Check-in service | Satisfaction level of the check-in service | Numerical ratio |

| | | |
|---|---|---|
| Inflight service | Satisfaction level of the inflight service | Numerical ratio |
| Cleanliness | Satisfaction level of the cleanliness | Numerical ratio |
| Departure Delay in Minutes | Minutes delayed when departure | Numerical ratio |
| Arrival Delay in Minutes | Minutes delayed when Arrival | Numerical ratio |
| Satisfaction | Satisfaction level: satisfaction and neutral or dissatisfaction | Nominal ordinal |

**Table 1: Description of the dataset features**

The descriptive statistics for the continuous variables are presented in Table 2, showing the distribution of the data, with the number of non-null entries in each column, the mean and standard deviation, minimum, maximum and quartile values.

| | Age | Flight Distance | Departure Delay in Minutes | Arrival Delay in Minutes |
|---|---|---|---|---|
| **Count** | 25976 | 25976 | 25976 | 25893 |
| **Mean** | 39.62 | 1193.78 | 14.3 | 14.74 |
| **Standard deviation** | 15.13 | 998.68 | 37.42 | 37.51 |
| **Min** | 7 | 31 | 0 | 0 |
| **25%** | 27 | 414 | 0 | 0 |
| **50%** | 40 | 849 | 0 | 0 |
| **75%** | 51 | 1744 | 12 | 13 |
| **Max** | 85 | 4983 | 1128 | 1115 |

**Table 2: Descriptive statistics of the numerical continuous features of the dataset**

## DATA QUALITY ASSESSMENT

We assume that the data is accurate and correct, given that there is no way to verify it. However, the data set contains no duplicate entries and a very limited number of missing values. Specifically, the feature *Delayed arrival in minutes* is the only column with missing data (83 records). From Table 2 it is known that at least 50% of the values are equal to zero. Thus, it is reasonable to replace the missing values with zero.

Since the source of the dataset does not mention the time period in which the data were collected, we have no certainty about the timeliness of the dataset. Nevertheless, most of the characteristics are relevant and related to the purpose of the study. As for sampling, the choice to use only the test dataset might introduce some bias, yet it was made in order to speed up the analysis and comply with the set limit for the number of rows.

## DATA VISUALISATION

Figure 1 categorizes the passengers according to different categorical characteristics with the distribution of their satisfaction. It is clear to see in a) that the distribution of women and men in the dataset is similar and that the proportion of passengers of both genders satisfied by the services is likewise close. In b), we can observe that the dataset counts more loyal passengers having a satisfaction level of nearly equal to half, whereas the disloyal customers are mostly dissatisfied. In c), we can see that the majority of passengers were on business trips and were mostly satisfied with the services offered, while those on personal trips were predominantly dissatisfied. Finally, in d), we can observe that the largest part of the passengers were either in business class or in economy class. Business class passengers were largely satisfied, while economy and economy plus passengers were mainly dissatisfied.

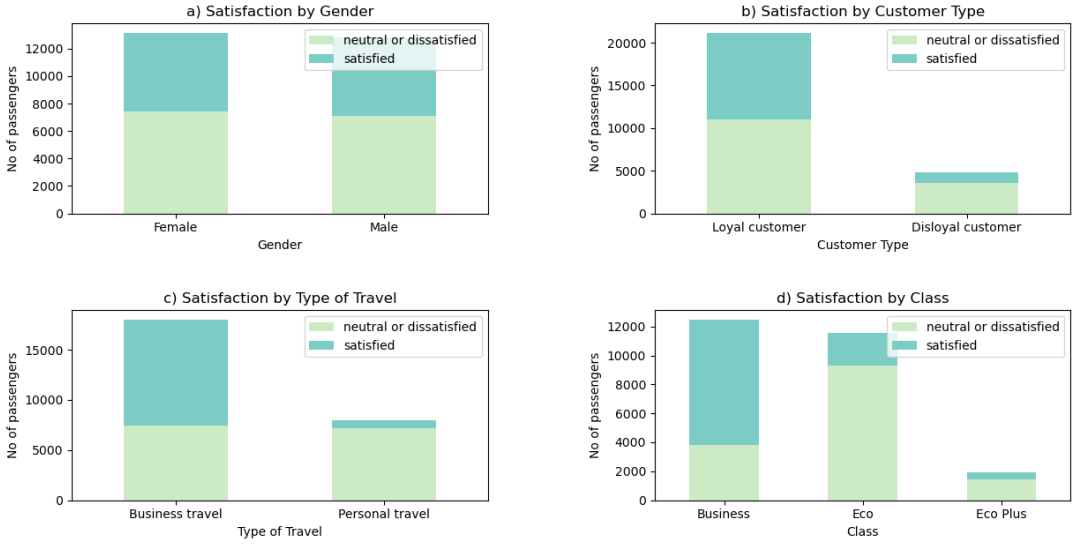


**Figure 1: Satisfaction levels of passengers grouped by gender, customer type, type of travel, and class**

The dataset includes 13 features representing satisfaction levels with the various services offered by the airline. Studying these satisfaction levels is critical in determining which services are successful and which need to be improved. Satisfaction levels are rated from 0 to 5, where 0 signifies not applicable, 1 not satisfied, and 5 very satisfied. Figure 2 shows the distribution of satisfaction levels across the different features.

Overall, the distributions of satisfaction levels in the dataset lean more toward positive ratings of satisfaction, with the middle range being between 3 and 4. 'In-flight service' and 'baggage handling' are the two services with the highest levels of satisfaction, receiving a score above

4 out of 5 from nearly 60% of the passengers. Meanwhile, 'in-flight Wi-Fi' service has the lowest levels of satisfaction, with more than 40% of the passengers rating it below 2 out of 5.

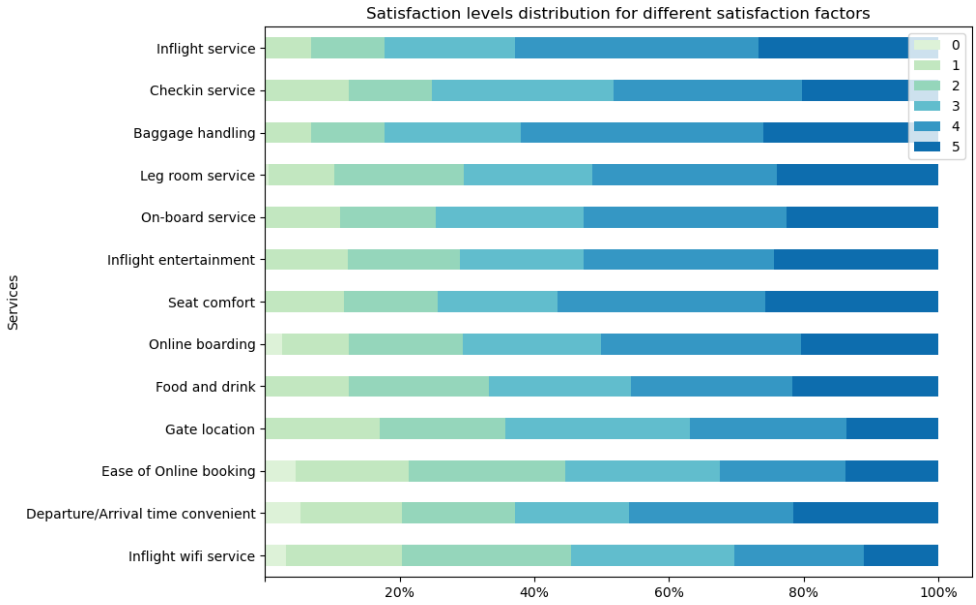


**Figure 2: Distribution of satisfaction levels of different features**

Next, Figure 3 shows the distribution of age (a) and flight distance (b) grouped considering airline passengers' satisfaction. We can observe that the age distribution is relatively similar between the two groups with a somewhat higher median age for satisfied passengers.

On the other hand, the distribution of flight distance between the two groups is very different, showing a higher dispersion among the satisfied passengers, whereas the dissatisfied passengers had mainly shorter flights.

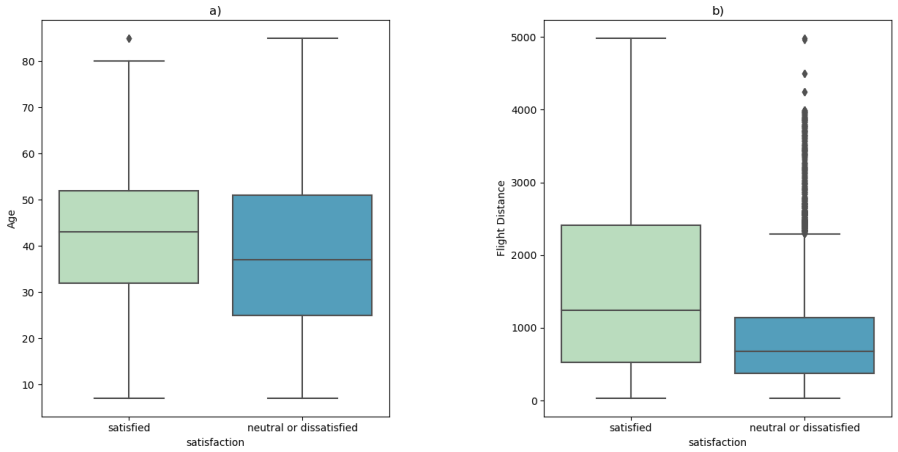


**Figure 3: Boxplots presenting the distribution of the Age and Flight distance grouped by satisfaction**

# CLUSTERING

The purpose of clustering for this application is to uncover similar and meaningful groups of passengers. In this part, three clustering approaches are used to segment the dataset and evaluate their quality. The approaches chosen are K-means, Agglomerative clustering, and DBSCAN.

## DATA PRE-PROCESSING

The pre-processing process for clustering includes the following steps:

- Deleting the *id* column as it does not contribute any information to the dataset,
- Converting the four categorical variables (*Gender*, *Customer Type*, *Type of Travel*, and *Class*) using dummy variables,
- Replacing the missing values in the feature *Delayed arrival in minutes* with zero, assuming that it means no delay was reported.

After the initial pre-processing of the dataset, and conversion of the categorical variables, there are 24 features left. Aside from that, the data sampling was considered when choosing the test data as the dataset, and no additional features were derived.

Following up, two more pre-processing tasks are performed: the scaling of the data and the dimensionality reduction. All these techniques are implemented in Python's library scikit-learn which was used throughout the data mining process.

### DATA SCALING

Most clustering algorithms chosen use distance as a measure of proximity. This makes the algorithms very sensitive to the scales of the dataset features. Indeed, features that are scaled differently do not contribute equally to the model fit and can create bias. For this application, the Min max scaling is used. It transforms the features into the [0,1] range. This method is implemented by scikit-learn MinMaxScalar.

### DIMENSIONALITY REDUCTION

Clustering algorithms such as K-means have a difficult time accurately clustering high dimensional data. Given that the dataset contains 24 features, the curse of dimensionality is likely to be present.

Principal Component Analysis (PCA) is one the most used dimensionality reduction techniques. It enables the reduction of the data from a high dimensional space to a lower dimensional one while retaining most of the variance in information from the original data [11].

The PCA function in Scikit-learn enables the specification of the number of components to be retained. However, this can result in a low explained variation ratio. For example, using 2 or 3 components explains only 34% or 45%, respectively, of the variation in the dataset. Thus, it is important to choose the optimal number of principal components to better represent the data. This can be done by specifying the variation ratio to retain instead. To maintain 90% of the variation in the dataset, 13 principal components must be used.

## GOODNESS-OF-FIT MEASURE

The validation technique used for clustering is the silhouette coefficient. The silhouette coefficient gives a measure of the separability of the clusters, and thus the quality of the clustering approach, which ranges from -1 to 1. A score of 1 signifies that the clustering resulted in good internal variations of clusters that are clearly separated. While a score of 0 means that the clusters are not significantly distinguishable from each other. Finally, -1 indicates that the clusters are formed incorrectly. The silhouette coefficient is computed using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The formula used is (b-a)/max(a, b) [12].

## K-MEANS

K-means is a partition-based algorithm which main constraint is that the number of clusters must be set in advance. Many approaches can be used for this like domain knowledge or using pre-labelled data. However, the dataset at hand does not contain any clues about the number of possible subgroups among the passengers. Thus, the best approach is to use the elbow method.

The elbow method is an iterative statistical method which runs the k-means algorithm multiple times, setting different number of clusters parameters, in order to determine the optimal value. For each iteration, the squared distance of each point to its assigned cluster's centroid is calculated, and the sum of all squared distances gives the clustering error, otherwise known as the inertia. A better cluster has tightly grouped data members, hence a lower sum of squared distance [11]. By running the k-means with different number of clusters and computing the SSE, the line chart in Figure 4 is obtained.

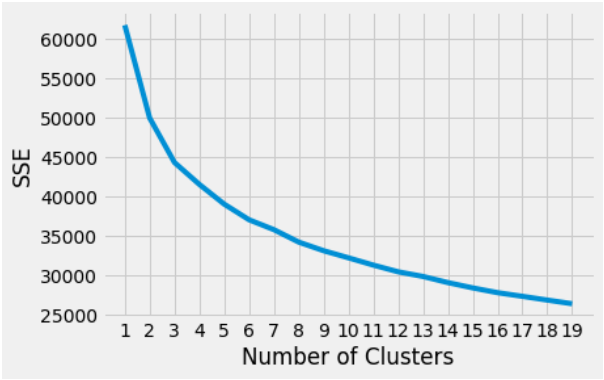


**Figure 4: K-means clustering - elbow method result**

The optimal number of clusters ought to be at the bend in the curve where the sum of the square distances is the lowest. As it may be difficult to identify the elbow of the curve graphically, the python package KneeLocator can be used to programmatically locate it instead. In this case, it finds that the optimal number of clusters is 6.

In order to evaluate the quality of k-means clustering, the algorithm is fitted to both the original data and the PCA data, setting the number of clusters to 6. As can be seen in Table 3, better quality is obtained by adapting the k-means model to the PCA data. The inertia is lower, and the silhouette coefficient is higher. Although the PCA data has a 3% higher silhouette coefficient, the score remains very low (close to zero).

| | Inertia | Silhouette coefficient |
|---|---|---|
| **Original scaled dataset** | 37049.33 | 0.15149 |
| **PCA data** | 33007.88 | 0.18333 |

Table 3: Inertia measure and Silhouette score of the K-means algorithm

By reducing the dataset into 2 principal components, the clusters obtained by the k-means clustering can be observed in Figure 5. The clusters are not very distinguishable from each other. Hence, this might be the reason for the low silhouette score obtained. However, they are reduced to a 2-dimensional space, which impacts the ratio of the explained variation.

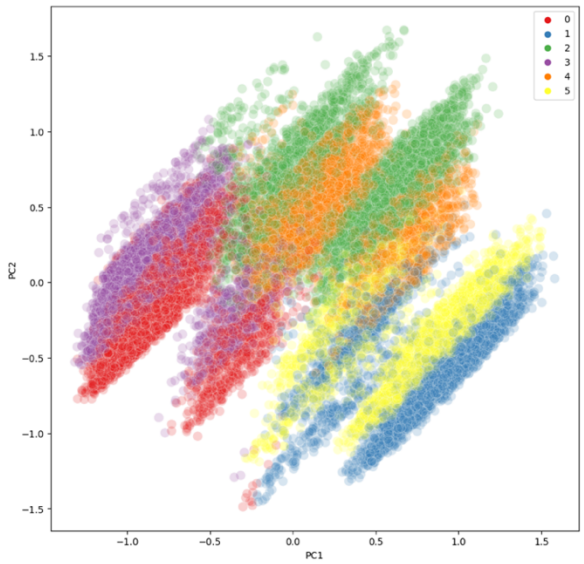


Figure 5: K-means clustering result

## AGGLOMERATIVE CLUSTERING

Agglomerative clustering is a hierarchical clustering that can be visualised with a dendrogram. The latter is used to choose the optimal number of clusters. Figure 6 shows the dendrogram resulting from fitting the PCA data to the agglomerative clustering algorithm. From the dendrogram, and programmatically, it is found that 4 is the optimal number of clusters for this approach.

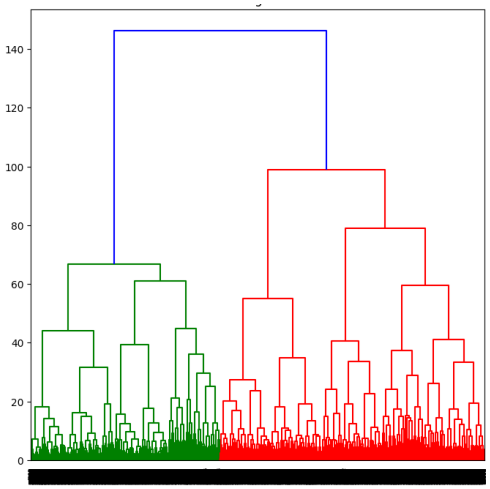


Figure 6: Agglomerative clustering dendrogram

Furthermore, the agglomerative clustering implementation in scikit-learn [13] provides four linkage methods which determine the distance to use when merging clusters:

- Single linkage: minimizes the minimum distance between the closest data objects of a pair of clusters,
- Average linkage: minimizes the average of the distances between all the data objects of a pair of clusters,
- Complete or maximum linkage minimizes the maximum distance between data objects of a pair of clusters,
- Ward: minimizes the sum of squared differences within all clusters.

To choose the optimal linking method, the agglomerative clustering algorithm is performed by setting the number of clusters to 4 and evaluating the quality of each linking method. Table 4 shows the silhouette coefficients obtained for each linking method. The single link gives the lowest silhouette score of 0.00675 while the best performing method is the ward with a score of 0.19573. Nonetheless, this best silhouette score obtained so far remains low.

| Linkage method | Simple | Average | Complete | Ward |
|---|---|---|---|---|
| **Silhouette score** | 0.00675 | 0.16692 | 0.10315 | 0.19573 |

**Table 4: Silhouette scores of different agglomerative clustering linkage methods**

In Figure 7, the dimensionality is reduced to a 2-D space and the clusters resulting from the agglomerative algorithm using the ward linkage are shown. We observe that the 4 clusters obtained are better separated than the clusters from the k-means method.

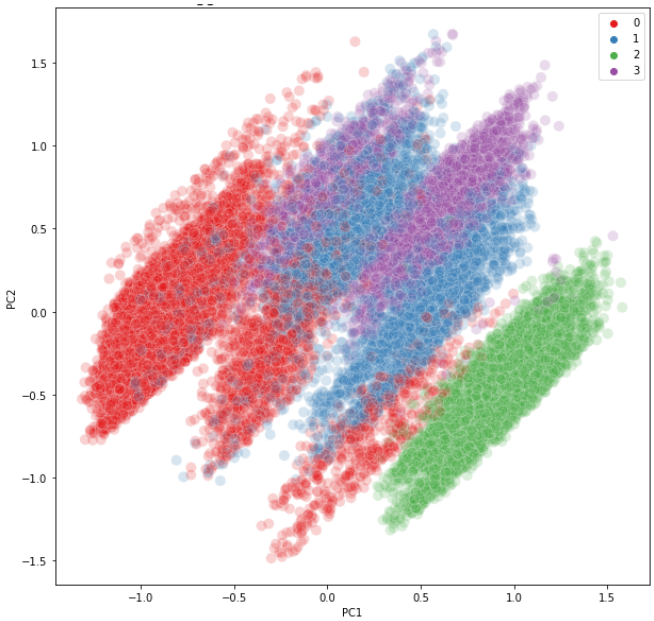


**Figure 7: Agglomerative clustering result**

## DBSCAN

DBSCAN is a density-based clustering approach different from the previous approaches. It does not rely on an optimal number of clusters but requires the setting of the two parameters [14] (1) Eps: which is the radius of a core point data, and (2) MinPts, the minimum number of data points within the radius of a point to be considered a core point.

The quality of DBSCAN depends on a good estimation of these parameters. A method for the estimation is presented in [15]. Although the MinPts value can be chosen based on the domain expertise, it mainly depends on the size of the dataset, and should be greater or equal to the number of features. According to [15], if the number of features is greater than 2, the minimum number of samples should be twice as large. As for the Eps, it can be automatically estimated following the method described in [16] where the average of distances between each point and its selected k-nearest neighbours (such as k = MinPts) is calculated and then the distances obtained are sorted and plotted increasingly to find the optimal value of Eps.

Applying these techniques to our 13 principal components dataset, the MinPts is set to 26 and the k-nearest neighbours algorithm is fitted to the data to estimate the Eps. The resulting averaged distances are plotted in an increasing order and presented in Figure 8. Using the KneeLocator function, the optimal value for Eps is around 0.76344.

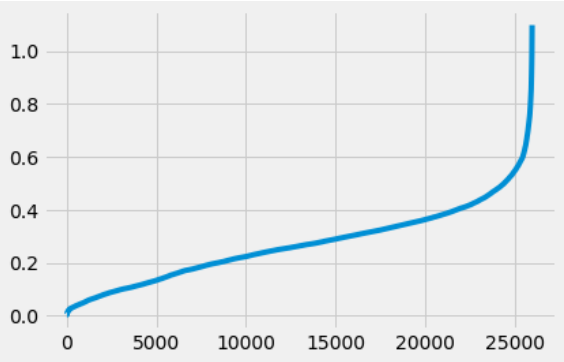


**Figure 8: Estimating the eps parameter of the DBSCAN clustering**

The DBSCAN approach was fitted to the dataset, with the chosen MinPts and Eps, and resulted in a silhouette score equal to 0.2 which is the best so far, although remains low. The algorithm identified 26 clusters apart from the default noise cluster. In an attempt to increase this score, a few more combinations of MinPts and Eps close to the estimated values are tested. However, the obtained silhouette scores remained mostly around 0.2 with the same number of clusters identified.

## MODEL INTERPRETATION

Overall, the clustering quality assessed using the Silhouette coefficient ranged from 0.15 to 0.2, with K-means being the worst performing model. The estimated optimal number of clusters was 6 for K-means and 4 for Agglomerative Clustering, whereas the optimal DSBCAN's parameters identified 26 clusters. Nonetheless, the silhouette coefficients achieved are very low, and all three algorithms with varying parameters failed to increase it to an acceptable score closer to 1.

An interpretation of the results could help to better understand the low score. The model chosen for interpretation is the agglomeration method because it has the second highest silhouette score (0.19573) and fewer clusters (4), which is simpler to make a comparison.

Figure 12 shows the distribution of the passengers between clusters. We can observe that cluster0 is the largest cluster with 41.7% of the passengers.

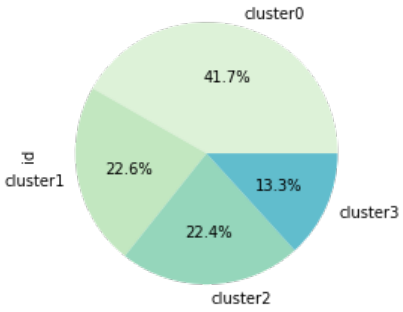


**Figure 9: Distribution of the passengers between clusters**

As for Figures 9, 10, and 11, they show the distributions of the different features among the identified clusters. The distributions of the features *Departure delay in minutes*, *Arrival delay in minutes*, and *Gender* are particularly similar between the clusters. Thus, they are not used when comparing and interpreting the clusters.

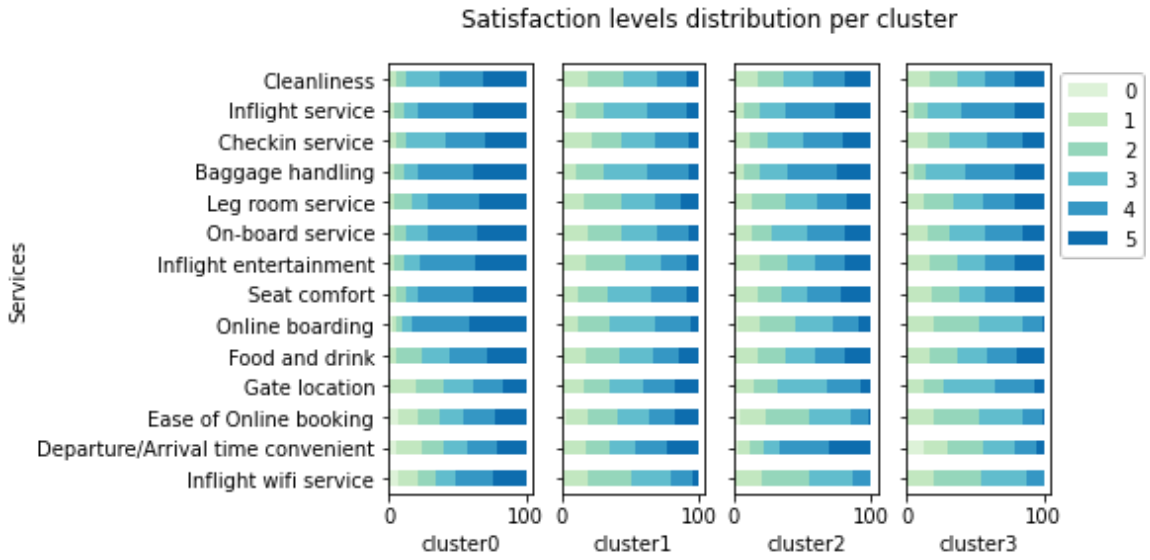


**Figure 10: Satisfaction factors levels distribution per cluster**

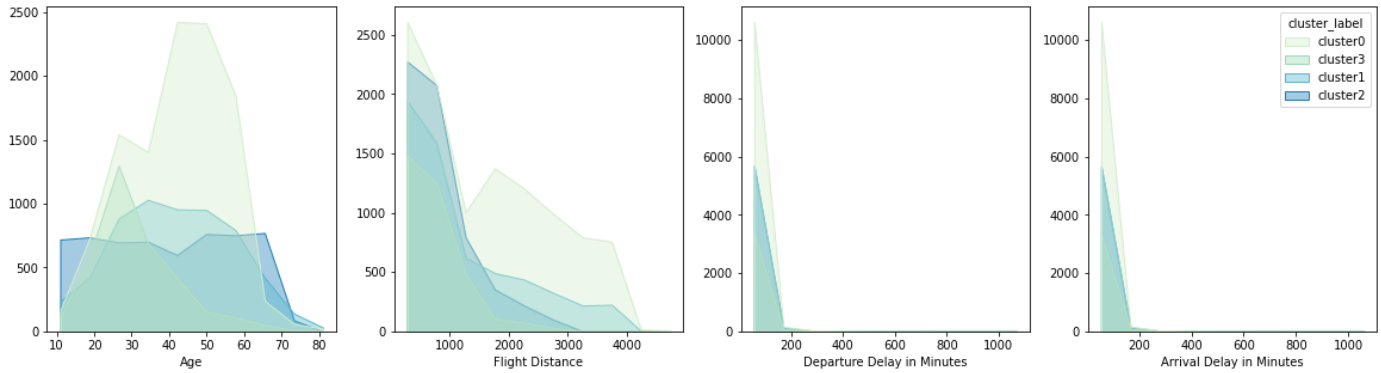


**Figure 11: Continuous features' distribution among clusters**

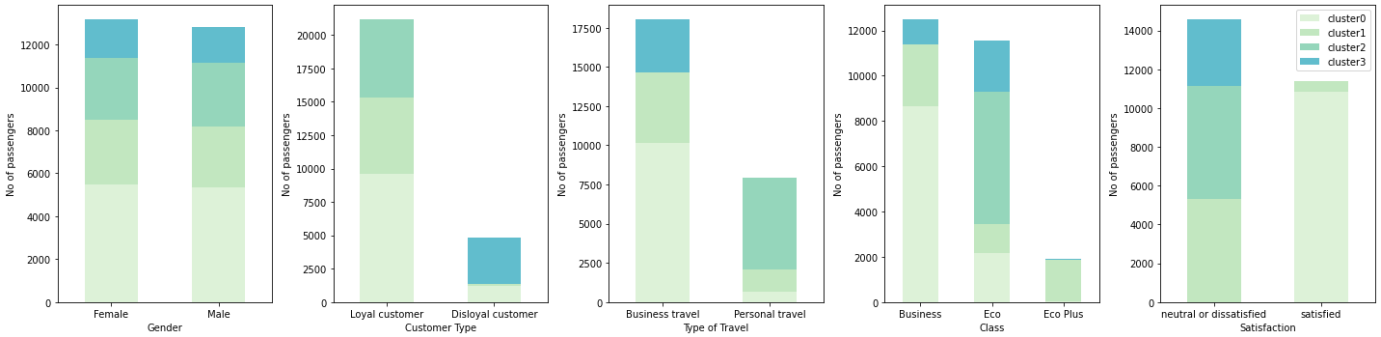


**Figure 12: Categorical features' distribution among clusters**

We can observe from Figures 9, 10, and 11 that:

- Cluster 0 passengers are mostly: happy with the services offered (rating them over 4), loyal business travellers, and overall satisfied with their experience. Their age distribution shows a higher number of middle-aged passengers (between 40 and 60 years old) having the longest flights
- Passengers from cluster 3 are mostly: unhappy with the services offered (rating 3 or less), disloyal customers on business travels, travelling mainly in Business or Eco classes and are dissatisfied or neutral to the experience. They are mainly young adults, between 20 and 30.
- As for clusters 1 and 2, their passengers had a relatively similar feedback about the services and were mostly loyal customers. While cluster 2 passengers travelled on personal basis in Eco class, cluster 1 passengers were mainly traveling for business in either Business or Eco Plus class. However, passengers from both clusters were dissatisfied or neutral about the experience.

Finally, although the performance of the clustering algorithms yielded rather poor silhouette scores, the clustering achieved by the agglomerative method showed a fairly good outcome. The passengers in clusters 0 and 3 were clearly distinguishable from the others.

From a practical point of view, given that the passengers in cluster 0 are mostly loyal to the airline and satisfied with its services, there is nothing more to improve for them. As for the passengers in cluster 3, it would be a challenge to attempt to retain them, since they are mostly dissatisfied with the services. Instead, it is better to focus on the passengers in clusters 1 and 2. Even though they are mostly loyal customers, they are dissatisfied or neutral about their experience. Thus, improving the services they are dissatisfied with, such as in-flight WIFI and online boarding facilities, could increase their appreciation and loyalty to the airline.

# CLASSIFICATION

## DATA PREPROCESSING

Many of the same pre-processing steps applied before clustering are also applied here:

- *Satisfaction* is isolated as the target variable
- The *Customer Type* and *id* columns are removed from the dataset as both possibly would negatively impact the classification model. This is because *Customer Type* is another possible target variable in the data and therefore should not be used as a prediction attribute, and *id* is a totally arbitrary number that can only harm the tested models by overfitting.
- Null values are only present in *Arrival Delay* and are filled with 0 as it is assumed if a delay time was not recorded then there was no delay.
- Categorical values are not compatible with the sklearn library, so these are encoded through of binary encoding for the *Gender*, *Type of Travel*, and *Satisfaction* columns and through dummy variables for *Class*.

The data is then divided into an 80/20 training to test split in order to enable an accurate assessment of any model's performance on unseen data. Both sets are normalized, so that each feature makes an equal contribution in algorithms relying on distances measures. The PCA domain transformation from the clustering section is also completed here with 13 features to use for some classification algorithms.

## CONSTRUCTING MODELS: DECISION TREE

It is very simple to create a decision tree in sklearn, using its DecisionTreeClassifier function. The construction of the tree is completed using either information gain or the Gini impurity index as the splitting metric (the library function runs an optimised version of the CART algorithm, but with the option to switch metrics [17]).

A preliminary tree is created for each of these splitting criterions, and their performances shown in Table 5. It is seen that the training accuracy is 1 for both algorithms, which is an immediate warning sign that overfitting is taking place, something that decision trees are notorious for. Performance between the two different metrics is very similar, with the most noticeable difference being the build and classification times, with information gain quicker in both.

| Criterion | Training Accuracy | Validation Accuracy | Test Accuracy | Build Time (s) | Classification Time (s) |
|---|---|---|---|---|---|
| **Gini Index** | 1.0 | 0.925 | 0.922 | 0.130 | 0.00207 |
| **Information Gain** | 1.0 | 0.927 | 0.925 | 0.111 | 0.00194 |

**Table 5: Performance of ID3 and CART algorithm decision trees. Validation is completed using 10-fold cross-validation.**

Given the drop in accuracy from training to validation and test data, it is clear that overfitting is limiting the performance of the tree on unseen data. This is unsurprising when examining the size of the trees - 1031 leaves for information gain, 1138 for Gini index and each with a maximum depth of 31.

In an attempt to reduce this overfitting, cost complexity pruning is utilised on the training data set with cross-validation in order to determine the ideal method to prune the original trees. The decision tree function in sklearn offers a method to find the pruning path, from which a tree is created for each value of $\alpha$, the tree complexity penalty term. Each of these trees is constructed to minimise the cost function given in Equation 1, where $|T|$ is the number of leaves in the tree, and $R(T)$ is a loss function of the given tree.

$$C_\alpha = R(T) + \alpha|T|$$

**Equation 1**

This process produces hundreds of trees which are all is tested using cross-validation, and the tree which provides the best validation accuracy is chosen for each splitting criterion. The determined optimum $\alpha$ values are 0.000207 for Gini index and 0.000378 for information gain. The performance of these pruned trees is shown in Table 6, and it clear that the pruning has had the desired result – the accuracy achieved in validation and training for both criteria has increased, and as expected this results in a drop in the training accuracy – the overfitting is definitely reduced. As an added benefit the build and classification times are also smaller, which is expected as the trees are smaller than before. In terms of the difference between the two splitting metrics, the difference in accuracy is negligible, so we will move forward with Gini index because of its lower classification time.

| Criterion | Training Accuracy | Validation Accuracy | Test Accuracy | Build Time (s) | Classification Time (s) |
|---|---|---|---|---|---|
| **Gini** | 0.950 | 0.938 | 0.937 | 0.118 | 0.00208 |
| **Information Gain** | 0.952 | 0.937 | 0.938 | 0.113 | 0.00245 |

**Table 6: Performance of CART and ID3 decision trees after cost-complexity pruning.**

To further illustrate the change in performance due to pruning, Figure 13 displays confusion matrices before and after the process. It's clear to see that the true positive rate (sensitivity) has increased, but the true negative rate (specificity) has actually worsened very slightly. For the purposes of this study, we are interested in overall accuracy, which clearly still noticeably improves.

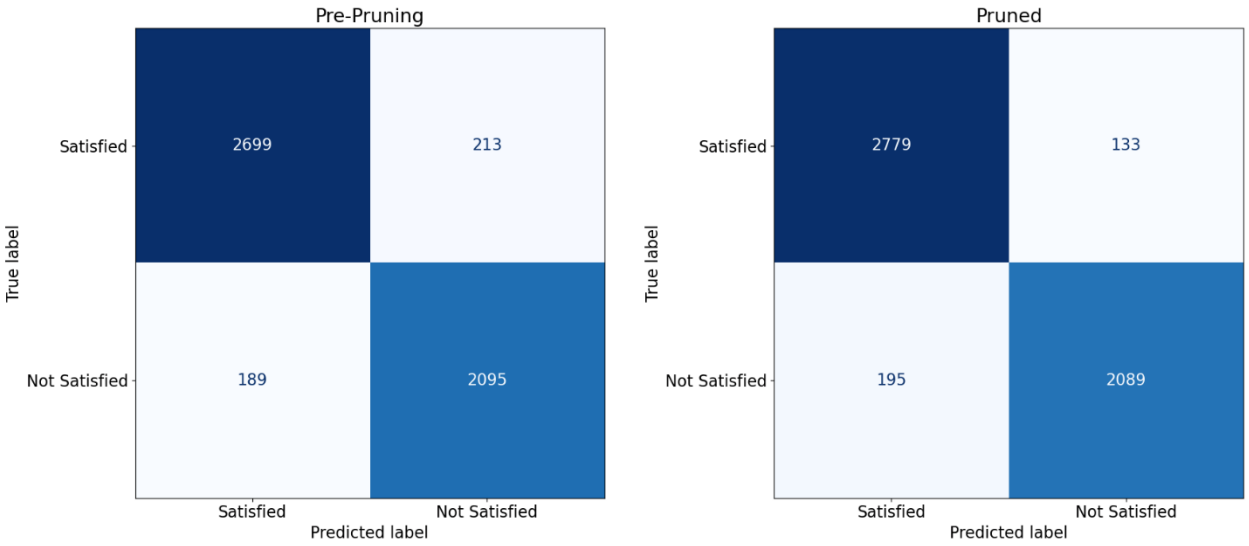


**Figure 13: Confusion matrices for decision tree built with Gini index classifying the test data, before and after cost-complexity pruning.**

## CONSTRUCTING MODELS: K NEAREST NEIGHBOUR

As for the decision tree, the sklearn library provides a function to create a k Nearest Neighbour classifier. However, as for all kNN classifiers, its performance depends very strongly on its hyperparameters – namely the number of nearest neighbours used to make a decision, *k*. In order to test a wide range of model hyperparameters, a grid search technique is used. The parameters investigated are shown in Table 7: Hyperparameters tested in kNN grid search. For a description of each of the distance metrics used, please see APPENDIX A – KNN Distances

Figure 14 shows the results of this grid search, with the best performing combination of hyperparameters determined to be $k = 10$, using the Manhattan distance metric and the inverse of the squared distance as the voting weight. The performance of this specific combination is then found for the test set, and its results shown in Table 8. It's clear that the classification time using kNN is much greater than that of the decision tree, and if classifying large amounts of records, this could be problematic. For that reason, the same grid search process is also completed with the PCA data. It reduces the classification time by about a third at the price of a few percent accuracy.

The training accuracy of 1 could again be a warning of overfitting, but the value of k has been optimised for the best validation score, and techniques such as pruning are not applicable here.

| Hyperparameter | Tested Values |
|---|---|
| **k** | 2 to 100 |
| **Distance Metric** | Euclidean, Manhattan, Chebyshev |
| **Weighting Scheme** | Uniform, Inverse(Distance), Inverse(Distance$^2$) |

**Table 7: Hyperparameters tested in kNN grid search. Note not every k value in range searched, from k=20 each increase was k=10.**

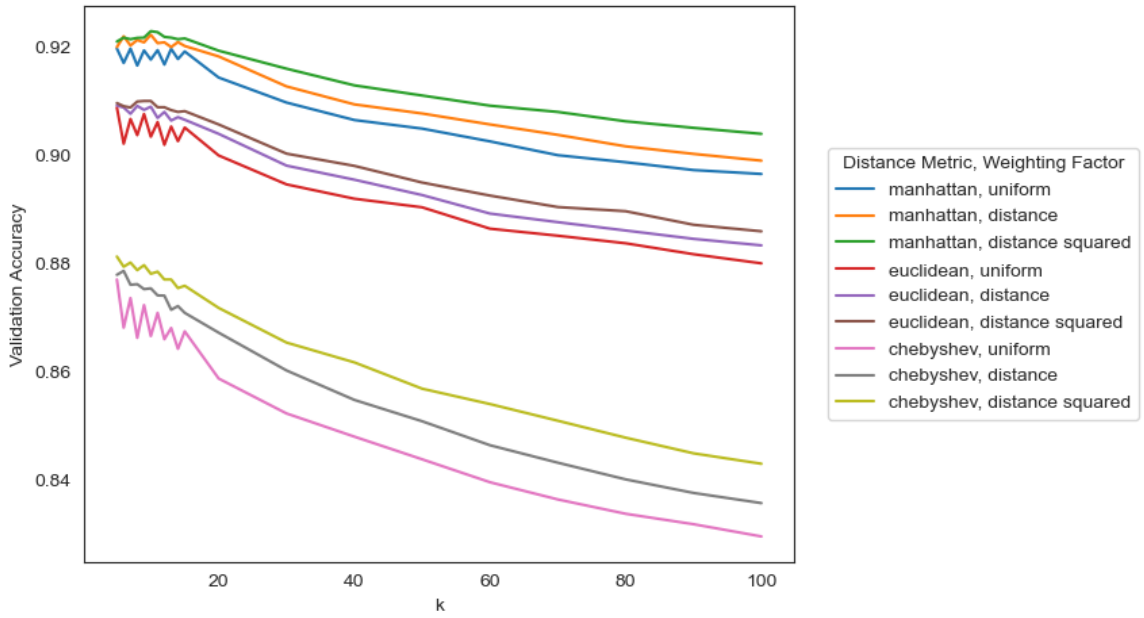


**Figure 14: Validation accuracy using 5-fold cross validation grid search.**

| Method | Training Accuracy | Validation Accuracy | Test Accuracy | Build Time (s) | Classification Time (s) |
|---|---|---|---|---|---|
| **kNN** | 1.0 | 0.923 | 0.918 | 0.0116 | 3.63 |
| **kNN (with PCA)** | 1.0 | 0.898 | 0.892 | 0.0415 | 1.21 |

**Table 8: Performance of kNN classification algorithm, optimised using a 5-fold cross validation grid search.**

## CONSTRUCTING MODELS: NAÏVE BAYES

A naïve Bayesian classifier is also tested, specifically using the Gaussian probability distribution as opposed to a multinomial approach, as some of the variables in the dataset have a well spread distribution (i.e. *Flight Distance*), eliminating the possibility of counting values as is required for the multinomial algorithm. The only hyperparameter requiring optimisation here is the variance smoothing term, added for calculation stability [18], which is chosen through cross-validation as $1E-12$, with the prior distribution determined in the standard way. The performance is below in Table 9.

| Method | Training Accuracy | Validation Accuracy | Test Accuracy | Build Time (s) | Classification Time (s) |
|---|---|---|---|---|---|
| **Gaussian NB** | 0.848 | 0.847 | 0.835 | 0.0170 | 0.00273 |

**Table 9: Performance of Gaussian Naïve Bayes classification algorithm, with variance smoothing parameter of 1E-12 for calculation stability.**

## CONSTRUCTING MODELS: ENSEMBLE (RANDOM FOREST)

Finally, an ensemble method is built to test if it can provide the improved accuracy and reliability that they are often touted to achieve. The standard method of a Random Forest is used, as the decision tree has produced the best results so far, and the method makes use of bagging to ensure that the chance of overfitting is low. It is decided that as each tree in the model only uses a small selection of features, there is no need to prune them as before. The only parameter optimised is therefore the number of trees in the forest, validated every 25 trees and found to reach a maximum performance with 175 trees. Performance is below in Table 10.

| Method | Training Accuracy | Validation Accuracy | Test Accuracy | Build Time (s) | Classification Time (s) |
|---|---|---|---|---|---|
| **Random Forest** | 1.0 | 0.949 | 0.947 | 4.07 | 0.120 |

**Table 10: Performance of Random Forest ensemble method, with 175 trees in the forest each trained on 5 features of a bootstrapped sample.**

## MODEL SELECTION & EVALUATION

In order to compare the models, it's important to consider if the difference in test accuracy is statistically significant, as if not then the better performance seen of one algorithm over the other could just be a result of random chance. To do this, hypothesis testing is utilised with a requirement for 95% significance using the standard deviation of:

$$\sigma = \sqrt{\frac{e_1(1-e_1)}{D_1} + \frac{e_2(1-e_2)}{D_2}}$$ [19],

to find the possible range of values for the difference in the accuracy between two models. If the range includes 0, then the difference is not statistically significant.

Starting with the algorithm with the lowest test accuracy, the Gaussian Naïve Bayesian, we can test it against the others to see if their performance meets this requirement. Then the second lowest test accuracy is tested against those better than it, and so on. It's found that the difference is statistically significant in every case, with 95% confidence. Therefore, we can definitively state that the best model in terms of accuracy is the Random Forest, achieving a 94.7% test accuracy. On the other hand, it takes far longer than any other method to be built, and longer than all others except kNN to classify but, given it's still only 4 seconds, this isn't a problem. Even if the model was extended to the full dataset available in [10], it would not require an unreasonable amount of construction time.

A confidence level can also be applied to the test result, so that it can provide a more complete estimate of real accuracy. Once again using a 95% confidence level, it can be said that the Random Forest will produce a real-world accuracy of between 94.2% and 95.2% on new unseen data, more than meeting the initial model requirements.

## SUMMARY

Multiple data mining approaches have been utilised to achieve the business objectives laid out at the start of this report, with classification being the more successful of the two investigations. Models were produced capable of identifying if passengers would be satisfied overall with their flight experience based upon their survey responses, with the best performing one returning an accuracy between 94.2% and 95.2% using a Random Forest. On the other hand, multiple clustering attempts produced clusters of poor quality, with low silhouette coefficient values, indicating that they are not meaningfully distinguishable. It is concluded that clustering is not an appropriate approach for this dataset.

With regard to future work, clustering will not be pursued as it is quite clear this dataset is not best suited to the technique. Instead, the classification models can be further explored and tested with a larger dataset to obtain better generalized models. A successful model could then be deployed to enable non-technical business executives to use it when making decisions about flight quality.

## LIST OF TABLES

## TABLE OF FIGURES

## WORKS CITED


[1] IATA, "Industry Statistics Fact Sheet," June 2019. [Online]. Available: https://www.iata.org/en/iata-repository/publications/economic-reports/airline-industry-economic-performance---june-2019---data-tables/. [Accessed 21 May 2021].

[2] IATA, "Current Airline Members," [Online]. Available: https://www.iata.org/en/about/members/airline-list/. [Accessed 21 May 2021].

[3] V. Briliana, "Consumer Satisfaction on Airline Passenger Loyalty: Antecedents and Outcomes," *International Journal of Business, Economics and Law,* vol. 16, no. 5, 2018.

[4] E. M. K. P. Y. X. Raphaël K. Akamavi, "Key determinants of passenger loyalty in the low-cost airline business," *Tourism Management,* vol. 46, pp. 528-545, 2015.

[5] D. K. E. L. Emanuel Lacic, "High Enough? Explaining and Predicting Traveler Satisfaction Using Airline Review," *arXiv:1604.00942,* 2016.

[6] "Understanding Customers' Evaluations Through Mining Airline Reviews," *International Journal of Data Mining & Knowledge Management Process,* vol. 6, no. 5, 2015.

[7] S. S. Sahar Tahanisaz, "Evaluation of passenger satisfaction with service quality: A consecutive method applied to the airline industry," *Journal of Air Transport Management,* vol. 83, no. 101764, 2020.

[8] G. N. a. S. Salini, "Customer Satisfaction in the Airline Industry: the Case of British Airways," *Quality and Reliability Engineering International,* vol. 22, p. 581–589, 2006.

[9] J.-Y. W. a. P.-H. Chung, "Retaining Passenger Loyalty through Data Mining: A Case Study of Taiwanese Airlines," *Transportation Journal,* vol. 47, no. 1, pp. 17-29, 2008.

[10] T. Klein, "Airline Passenger Satisfaction," 20 2 2020. [Online]. Available: https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction. [Accessed 21 5 2021].

[11] K. Mysiak, "Explaining K-Means Clustering," 14 Jul 2020. [Online]. Available: https://towardsdatascience.com/explaining-k-means-clustering-5298dc47bad6. [Accessed 13 Jun 2021].

[12] A. Bhardwaj, "Silhoutte Coefficient," 26 May 2020. [Online]. Available: https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c. [Accessed 13 June 2021].

[13] Scikit-Learn, "sklearn.cluster.AgglomerativeClustering," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html?highlight=agglomerative#sklearn.cluster.AgglomerativeClustering. [Accessed 13 June 2021].

[14] P.-N. Tan, M. Steinbach, A. Karpatne and V. Kumar, Introduction to Data Mining (Second Edition), Pearson, 2018.

[15] T. Mullin, "DBSCAN Parameter Estimation Using Python," 10 Jul 2020. [Online]. Available: https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd. [Accessed 13 Jun 2021].

[16] N. Radmah and S. I. Sitanggang, "Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra," *IOP Conference Series: Earth and Environmental Science,* vol. 31, no. 1, 2016.

[17] "User Guide: 1.10. Decision Trees," Scikit-learn, [Online]. Available: https://scikit-learn.org/stable/modules/tree.html#tree. [Accessed 26 05 2021].

[18] Sci-Kit Learn, "Gaussian Naive Bayes (GaussianNB)," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB. [Accessed 10 June 2021].

[19] H. Du, Applied Techniques of Data Mining and Machine Learning, Lecture 8, University of Buckingham, 2021.

[20] T. R. M. D. Randall Wilson, "Value Difference Metrics for Continuously Valued Attributes," in *Proceedings of the International Conference on Artificial Intelligence, Expert Systems and Neural Networks*, 1996.

## APPENDIX A – KNN DISTANCES

The kNN classification method revolves around calculating the distances between all of the data points in the dataset. Thus, the choice of how to calculate these distances can greatly affect the algorithm's performance. 3 metrics are tested in the conducted grid search, the equations for which are below. PEBLS distance, or the modified value distance metric, is not tested, as it does not handle continuous attributes directly [20], and there are multiple continuous attributes in this data for which we do not want to lose information through discretization.

| Metric | Calculation |
|---|---|
| **Euclidean** | $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ |
| **Manhattan** | $\sum_{i=1}^{n}\lvert x_i - y_i\rvert$ |
| **Chebyshev** | $max(\lvert x_i - y_i\rvert)$ |