University of Milan Bicocca

School of Science

Department of Computer Science, Systems and Communication

Bachelor of Science in Computer Science

Dataset link: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

MACHINE LEARNING

Progetto d'esame

Riccardo Moschi 856243 r.moschi@campus.unimib.it

Luca Poli 852027 l.poli6@campus.unimib.it

Index

1. Introduction

1.1. Description of the target domain and objectives

The target domain of this paper is aero travel; and in particular, customer satisfiability in relation to certain flight metrics.
The research aimed to establish the relationship between customer satisfaction and these influence variables, using a correlation method to test their effect, with the aim of being able to predict the satisfaction of future customers.

1.2. Design choices for creating the data set, any assumptions or assumptions

It was decided to analyze this dataset given the application domain of our knowledge.
We performed a restructuring of the dataset, changing string attributes to numeric, and performed a scaling that allows us to normalize the range of variation of features (features).
For design reasons, it was chosen to use only a sample of data in the dataset, given the enormous computational load it would place on the processor.

2. Dataset

2.1. Preprocessing

After loading the dataset into memory, and before its analysis, the very first preprocessing operations must be performed. Given the "academic" use of the data, and to speed up computation and processing time, we decided to randomly take a sample of 5000 instances (from the approximately 100000).

The preprocessing phase begins by removing unnecessary variables (such as the additional index and id); next, we proceed by moving and renaming the classification target variable so that it is easier to access; then we remove the null rows (since there are few of them, there is no need to impute them); finally, we convert the nonnumeric variables into factors.

2.2. Exploratory analysis

The exploratory analysis begins with the analysis of the distributions of the variables, using box plots and density plots; it then proceeds with the evaluation of the distributions in relation to the target, examining both the frequency of the two classes and the frequency of the data in relation to the class of the target.

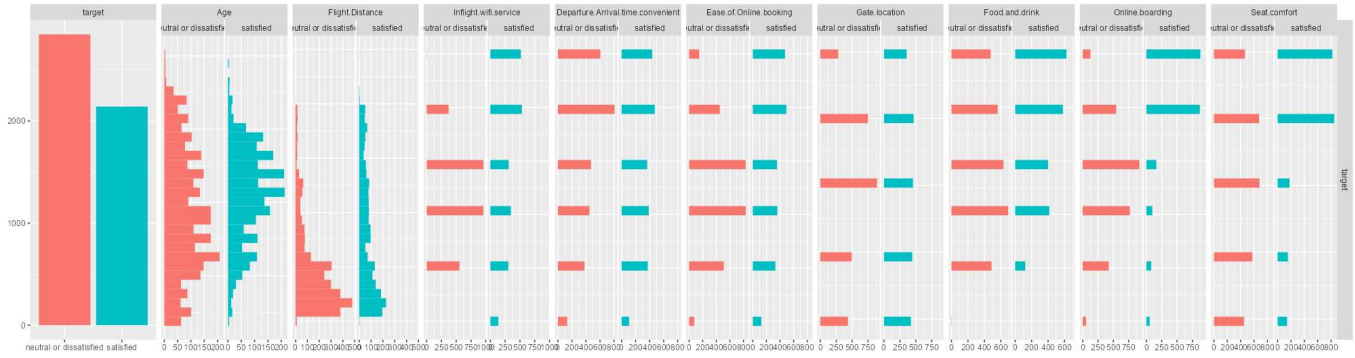


Figure 1: Summary graph of most of the variables in relation to the target

The graph presents the distribution of features in relation to the target, i.e., customer satisfaction or non-satisfaction based on the other data in the dataset, in the form of histograms.

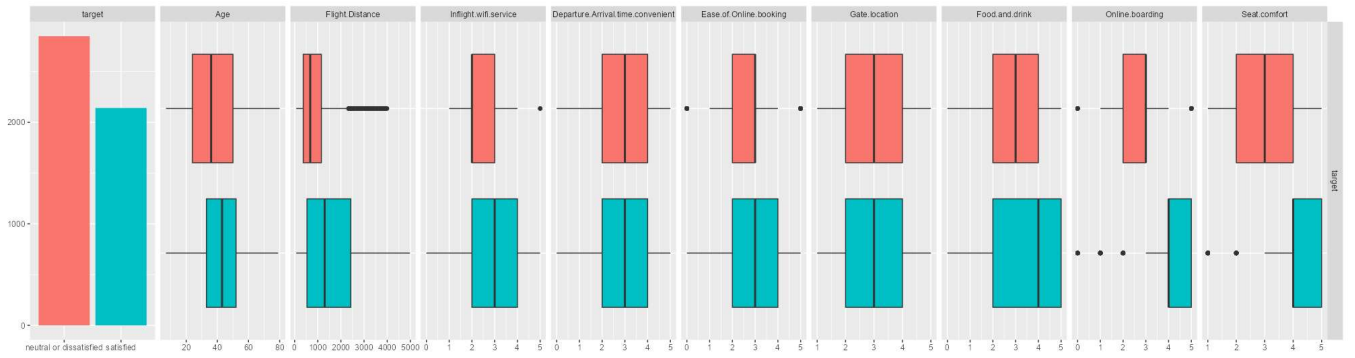


Figure 2: Summary graph of boxplots in relation to the target

The graph shows the distribution of features in relation to the target, in the form of boxplots.
In the following boxplots, it is possible to observe some data that deviate significantly from the rest of the data (outliers).
For example, it can be seen that the attribute "Flight.distance" has left-trending distribution (as also shown in *Figure 1*), and as we expect, it has several outliers as it is likely to take on multiple distance values that are far from the mean.

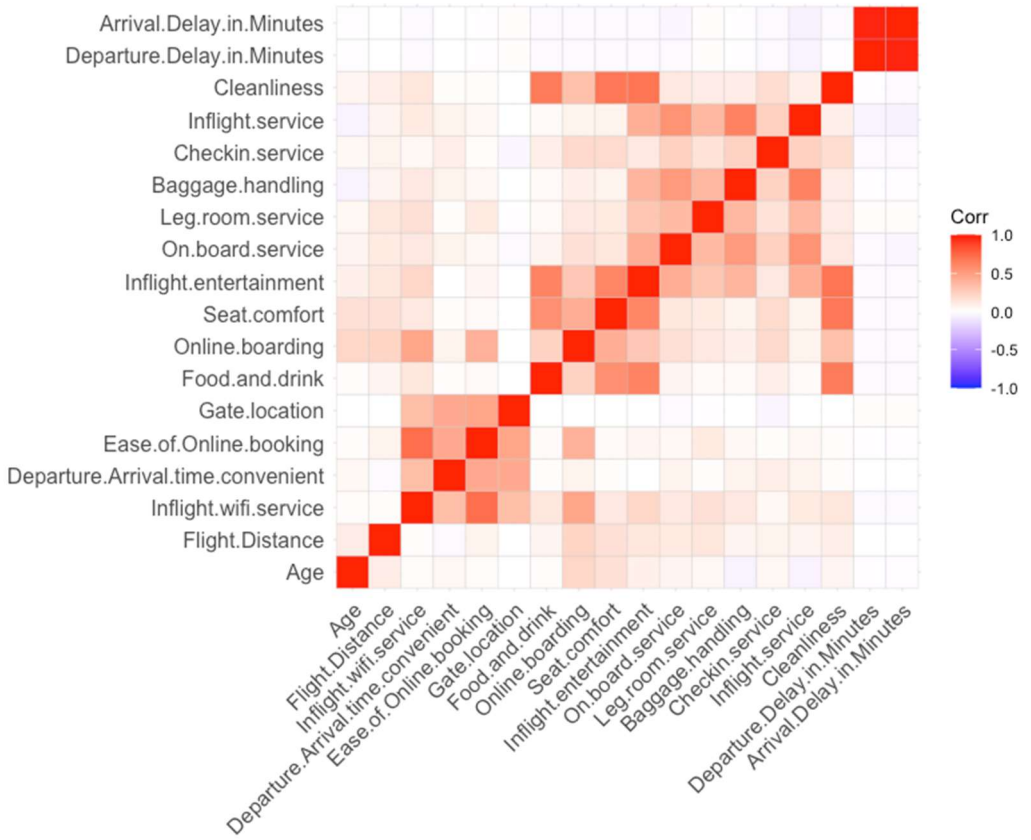


Figure 3: Attribute correlation matrix

The correlation matrix, i.e., the correlation matrix indicates whether two variables are related to each other in any way; in particular, it is possible to observe a good correlation between the features "Cleanliness" and "food.and.drink," "Seat.comfort," "Inflight.entertainment," and especially between "Arrival.Delay.in.Minutes" and "Departure.Delay.in.Minutes," which show the highest link.

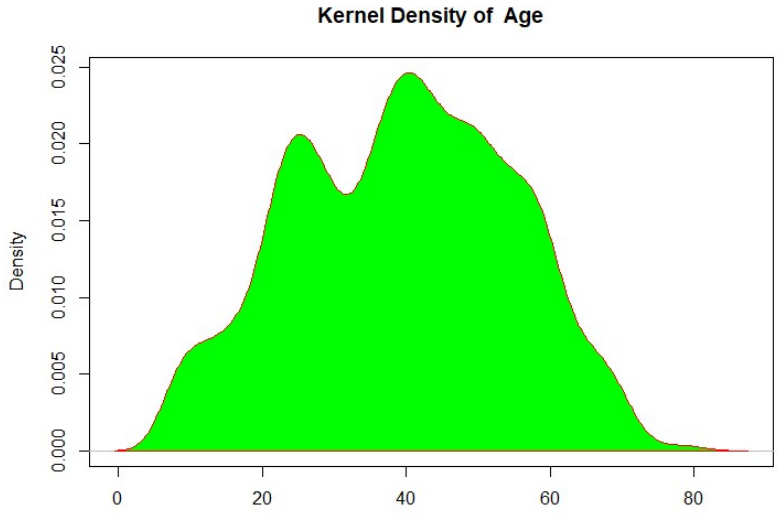


Figure 4: Kernel density of passenger age

It can be seen that the age is more thickened in the center, thus showing a higher frequency on the age around 40.

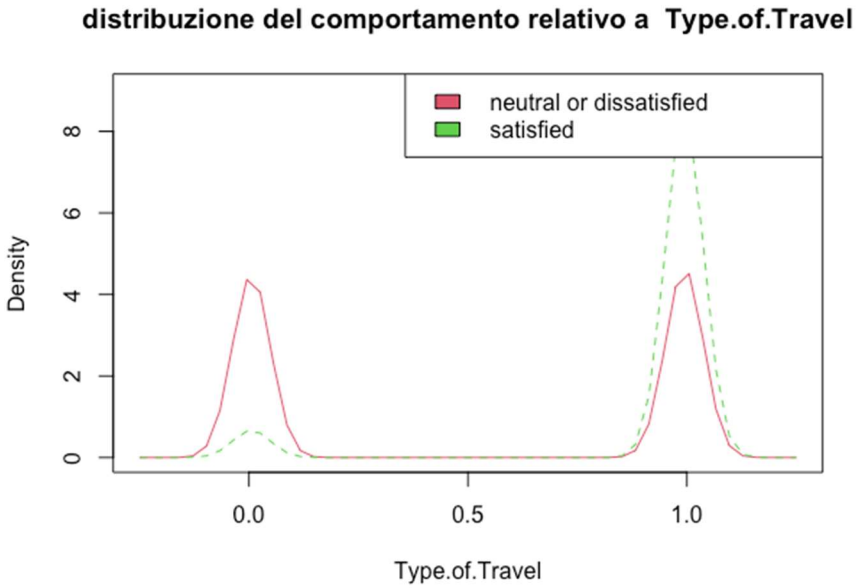


Figure 5: Distribution of behavior with respect to type of business travel (whether for leisure or business).

It is possible to observe the distribution of behavior (satisfied or dissatisfied), in relation to the type of travel (0="Personal Travel",1 =" Business.travel") and contrary to what we might expect, it tends to be that the most liked trips are those for Business.

2.3. Feature Reduction

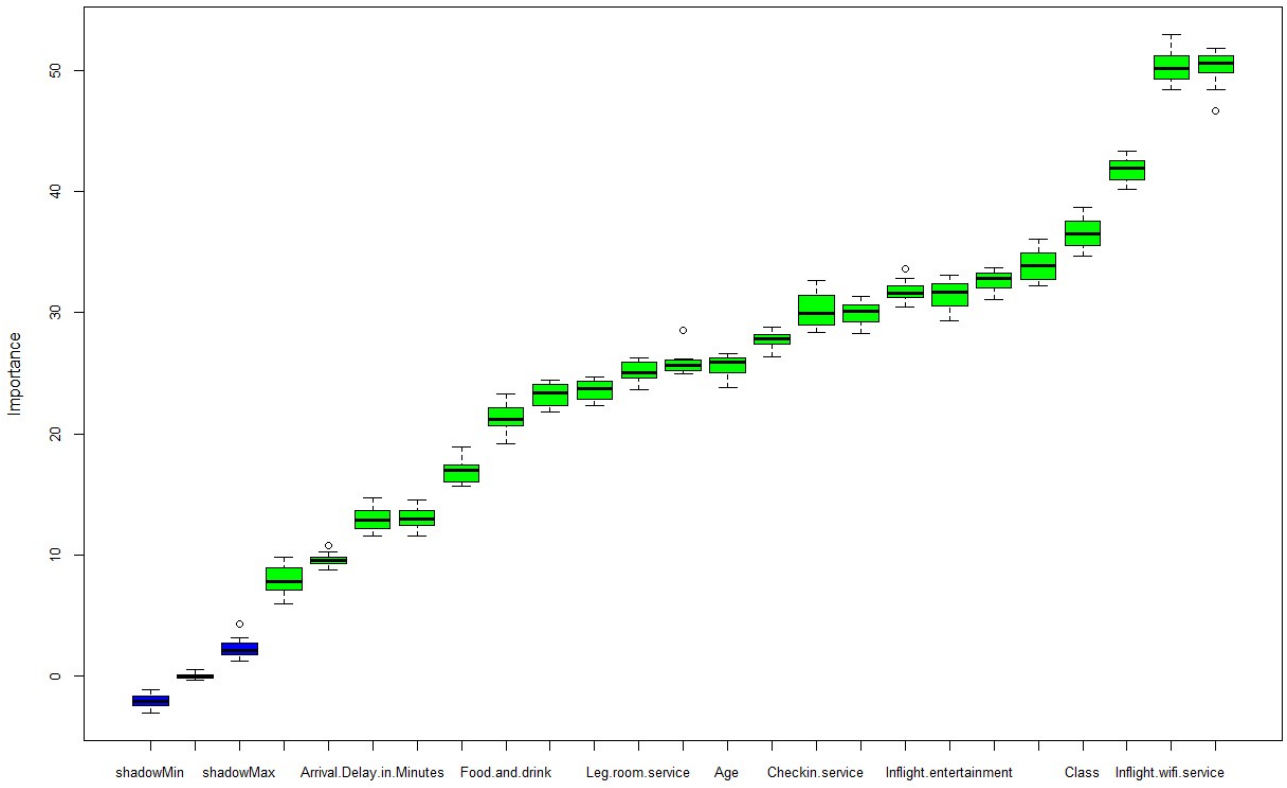
Given the large number of features, we decided to apply two strategies (Boruta and PCA) to try to reduce them.

2.3.1. Feature Selection: Boruta

Boruta is an algorithm that extends Random Forest models to select the best features to retain in the next phase.

The Random Forest algorithm applies bagging techniques to Decision Trees; that is, it extracts random samples from the train set, with different instances and features; then, on each sample it applies a Decision Tree and depending on the split obtained assigns an importance estimate.

Boruta applies a principle similar to Random Forests: first, from each feature, it generates a "synthetic" version of it called the Shadow; then it applies the Random Forest on the extended dataset obtaining the importance estimate; then it goes on to remove all the less important features from the most important shadow; it repeats these two steps until it has only the most important ones (or up to a maximum limit).



Figures 1: Importance estimates for each feature

Inflight.wifi.service	50,270
Online.boarding	50,223
Type.of.Travel	41,790
Class	36,640
Seat.comfort	33,890

Figures 2: Table of features sorted by average importance

As can be seen from the table and graph (showing the distribution of importance for each feature across cycles), the Boruta algorithm does not identify superfluous features (in fact except for shadows, in blue, every feature is in green).

2.3.2. Feature Extraction: PCA

Given the large number of features, a feature extraction strategy, PCA, was implemented in order to extract from the numerical data the main components that most affect the variance of the data.

We decided to choose the first 7 components to achieve our goal, as we wanted to achieve about 74% cumulative variance.

	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	3.83158079	21.2865599	21.28656
Dim.2	2.38775786	13.2653215	34.55188
Dim.3	2.19625476	12.2014153	46.75330
Dim.4	1.96684907	10.9269393	57.68024
Dim.5	1.23864985	6.8813880	64.56162
Dim.6	0.95753884	5.3196602	69.88128
Dim.7	0.91050903	5.0583835	74.93967

Figure 6 Parameters obtained from PCA

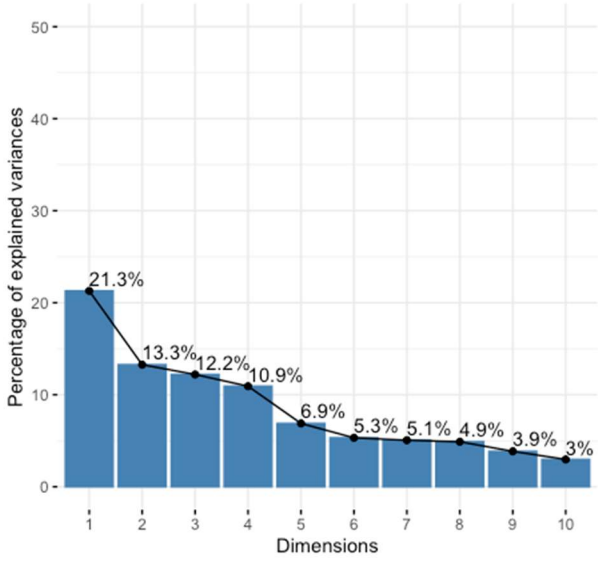


Figure 7 PCA dimensions in graph

Noting the result I will take the first 7 values as I get about 74% cumulative variance and I am satisfied with that data.

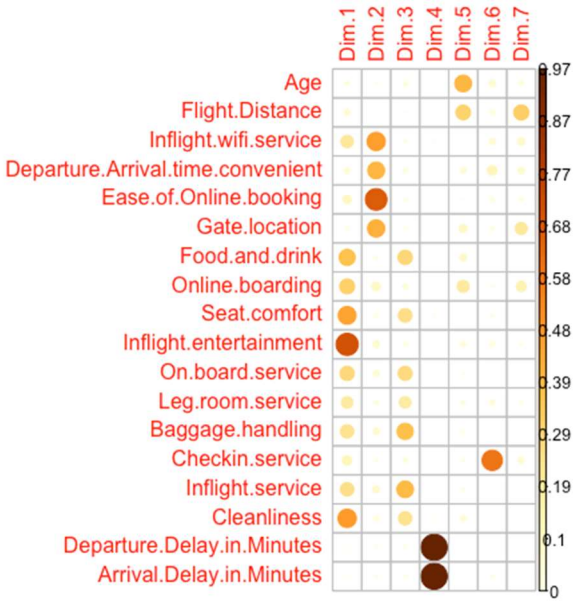


Figure 8 Cos2 of all features at dimension

Thanks to this graph it is possible to visualize the cos2 of the variables on all dimensions, in particular it is possible to observe in dimension 1 and 2 the attributes with higher value, as later analysis will follow.

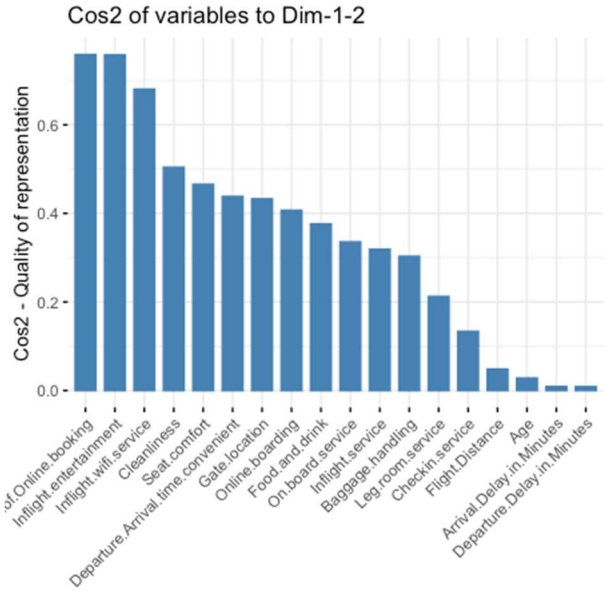


Figure 9 Cos2 of features versus dimension 1-2 of the PCA

It is also possible to create a bar graph on the cos2 of the variables, as it can be seen "Ease.of.Online.booking" turns out to be the feature that contributes the most with variance, compared to the total in dimension 1-2, thus deviating the most from the average. In fact, looking at *Figure 9*, the attributes at the "Dim.1" and "Dim.2" column with the highest variance (colored orange) are precisely the same ones that have higher values in the histogram.

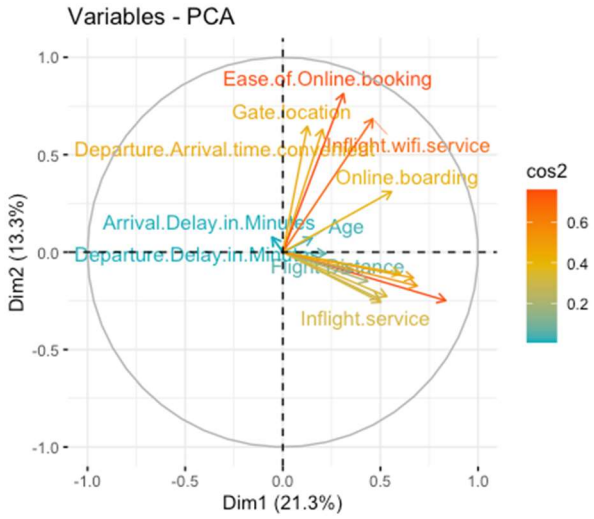


Figure 10 Cos2 represented in the graph

The same attributes, with larger variance, can be observed in this graph as well, in particular "Ease.of.Online.booking," "Inflight.wifi.service" and "Inflight.entertainment" turn out to be the reddest, i.e., with higher cos2.

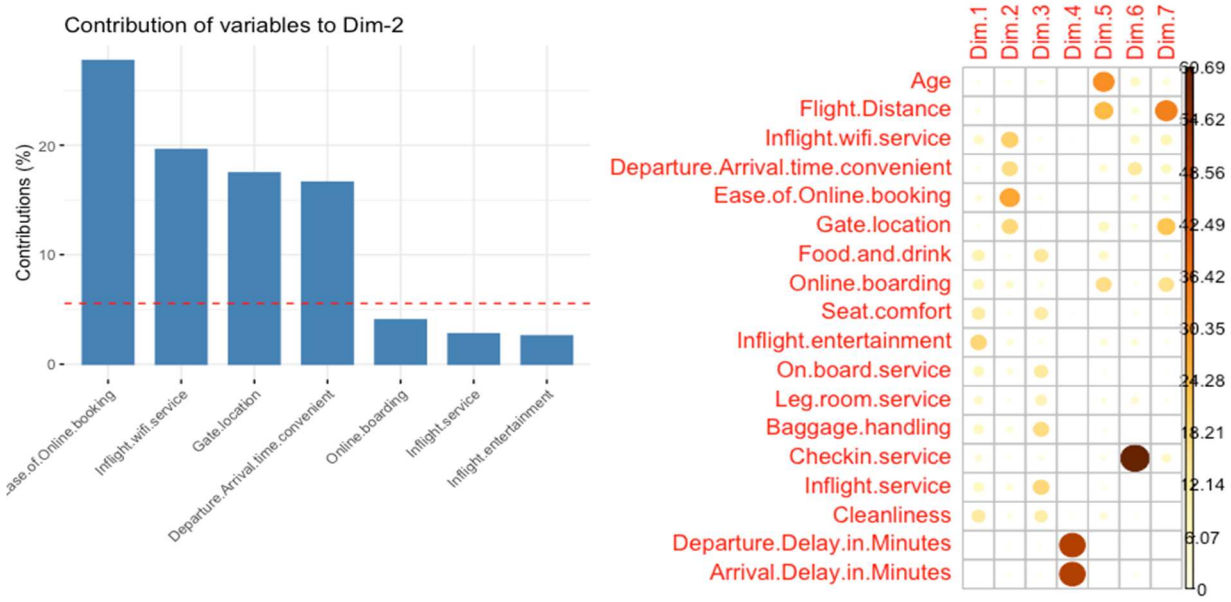


Figure 11 Contribution of individual variables in dimension 2 of the PCA

Starting from the graph on the left we can examine the contribution of the original variable to a given principal component, in this case in dimension 2, which as can be analyzed in the image on the right the feature with the largest contribution turns out to be "Ease.of.Online.booking."

3. Templates

It was decided to use 2 Machine Learning models: Support Vector Machine and Neural Networks; they were chosen for their potential and versatility, then trained by K-Fold Cross Validation with Grid Search (with k equal to 5).

3.1. Neural network

The neural network is an ML model, based on the composition of multiple neurons; each of which performs the weighted summation of its inputs and applies a function to it. The network employed has a deep structure: that is, with hidden layers. The first (noncomputational) layer inputs features into the network, and has as many neurons as variables; the second layer, called "hidden," performs the first computations via the nonlinear logistic function; the third layer, composed of 2 neurons with logistic, performs the last computations and returns the binary class membership probabilities ("satisfied" or "not satisfied"). The logistic function was chosen both because it returns probabilities, and therefore ideal for classification problems; and because, being nonlinear, it allows the two classes to be separated even in nonlinearly separable spaces.

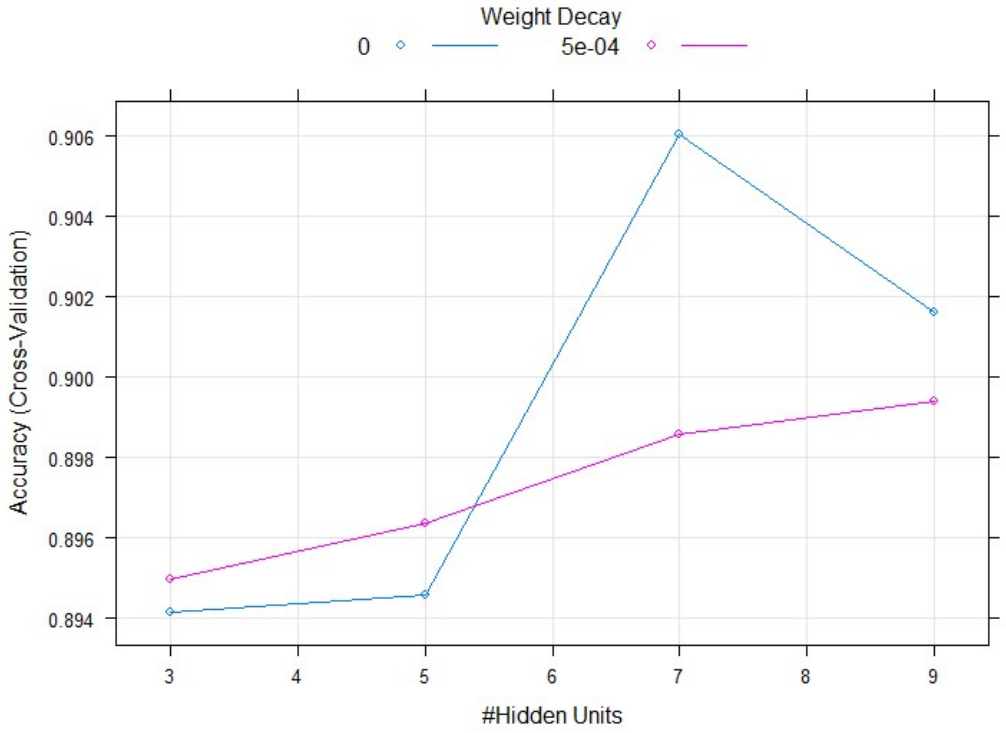


Figure 12 Cross-validation graph of Neural Networks

The network was trained with a K-Fold Cross Validation approach with Grid Search, with k equal to 5; this allows us to obtain more reliable evaluation metrics, because they are obtained by averaging the metrics over the k-folds; and it allows us to vary some hyperparameters to identify the best ones (performing so-called Tuning). The hyperparameters tested are: the number of neurons in the hidden layer, ranging from 3, 5, 7 and 9; and the decay of the weights, which varies between 0 and 0.0005. As can be seen

from the graph, the highest accuracy is obtained with 7 neurons and 0 of Weight Decay; therefore, this model will be the one to be then adopted in the next testing phase.

3.2. SVM

Svm is a type of classification model that uses support vectors to construct a surface separation decision between different classes. The choice of the separation surface is made in a way that maximizes the distance between classes, which means that the model tries to find the separation decision that has the greatest margin. One of the advantages is its ability to perform nonlinear separation using an appropriate kernel.

One important parameter we decided to vary is the cost, which allows us to penalize instances within the margin, improving the robustness of the model with respect to noise in the data.
While for the kernel parameter, we chose to use the Radial Basis Fuction (RBF); as, in addition to being one of the most widely used and efficient, it allows for classifications with different degrees of separability.

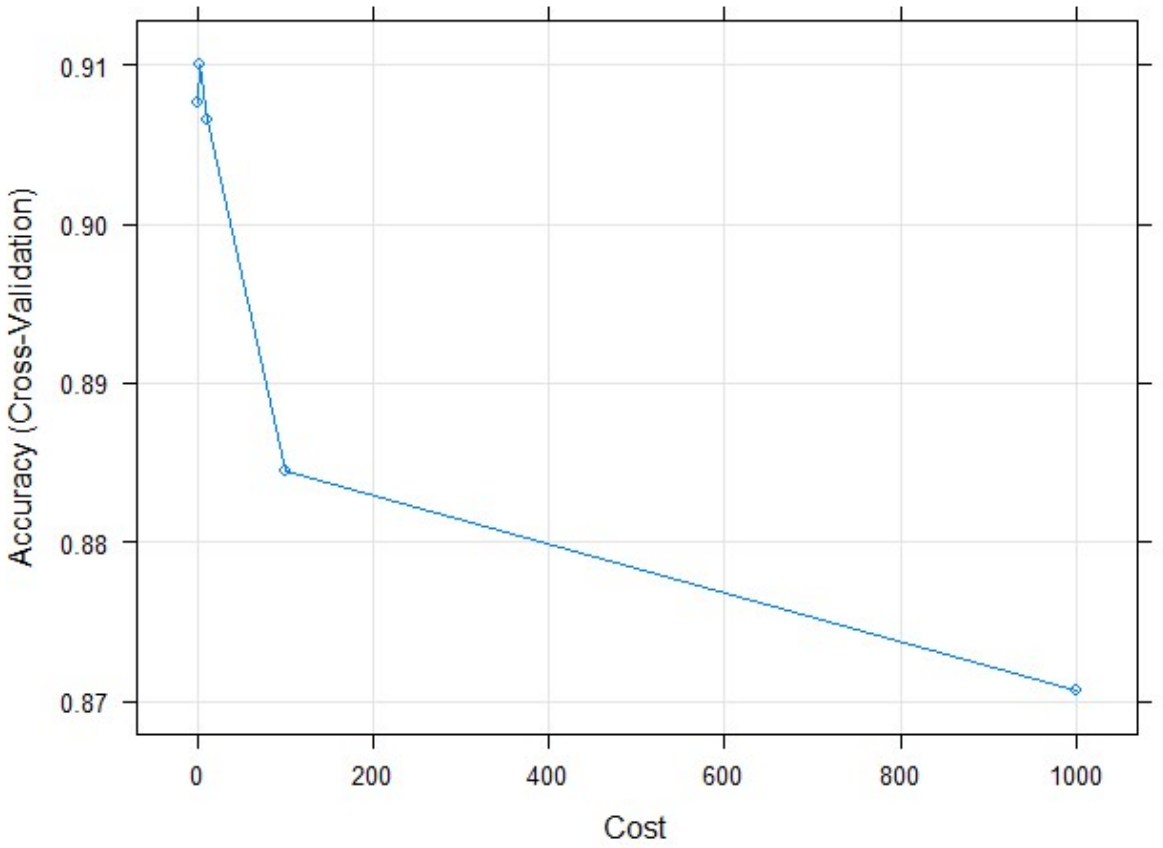


Figure 13 Cross-validation graph of SVM

This graph describes the K-Fold Cross Validation SVM that varies on the cost (from 1 to 1000)

It was decided to set some points: 0.5, 1, 10, 100, 1000; highlighted by dots in the graph. It can be seen that as the cost increases, the accuracy decreases; in fact, avoiding considering data would unbalance the classes, thus decreasing the accuracy.
In particular, it is analyzed from the graph that the value of cost = 1 brought particularly good results, that is, the optimal value, after which the accuracy decreases.

4. Results

The results obtained were calculated on the test data. These were extracted from the relevant dataset, taking a random sample of 500, and then processed similarly to the train data.

4.1. Confusion matrix

From the confusion matrices it is possible to observe a visualization of the results obtained from the models, in particular we will analyze *Precision, Recall* and *F-measure*.

4.1.1. Neural Network

```
                Reference
Prediction     satisfied dissatisfied
  satisfied          199           21
  dissatisfied        24          255

               Accuracy : 0.9098
                 95% CI : (0.8812, 0.9335)
    No Information Rate : 0.5531
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8173

 Mcnemar's Test P-Value : 0.7656

            Sensitivity : 0.8924
            Specificity : 0.9239
         Pos Pred Value : 0.9045
         Neg Pred Value : 0.9140
              Precision : 0.9045
                 Recall : 0.8924
                     F1 : 0.8984
             Prevalence : 0.4469
         Detection Rate : 0.3988
   Detection Prevalence : 0.4409
      Balanced Accuracy : 0.9081

       'Positive' Class : satisfied
```

Figures 3: Confusion Matrix, for the neural network, of predictions on the test set.

As can be seen, the accuracy obtained (of 90 percent) is excellent compared to the No Information Rate of 55 percent; that is, it means that the model is significantly more accurate than the Naive prediction (which, for each instance, simply returns the most frequent class, as it is most likely).

The decision result is very satisfactory, the model was able to correctly classify the observations in that class when it said they belonged to that class; in fact, it has 90% accuracy for the "satified" class, that is, it correctly classified 90% of the observations it said belonged to the "satified" class.

The recall result is also very good, the model has a recall of 89% for the "satisfied" class, meaning that it correctly classified 89% of the observations that actually belong to the "satisfied" class.

4.1.2. SVM

```
                Reference
Prediction     satisfied dissatisfied
  satisfied          200           15
  dissatisfied        23          261

               Accuracy : 0.9238
                 95% CI : (0.897, 0.9455)
    No Information Rate : 0.5531
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8454

 Mcnemar's Test P-Value : 0.2561

            Sensitivity : 0.8969
            Specificity : 0.9457
         Pos Pred Value : 0.9302
         Neg Pred Value : 0.9190
              Precision : 0.9302
                 Recall : 0.8969
                     F1 : 0.9132
             Prevalence : 0.4469
         Detection Rate : 0.4008
   Detection Prevalence : 0.4309
      Balanced Accuracy : 0.9213

       'Positive' Class : satisfied
```

Figures 4: Confusion Matrix, for SVM, of predictions on the test set.

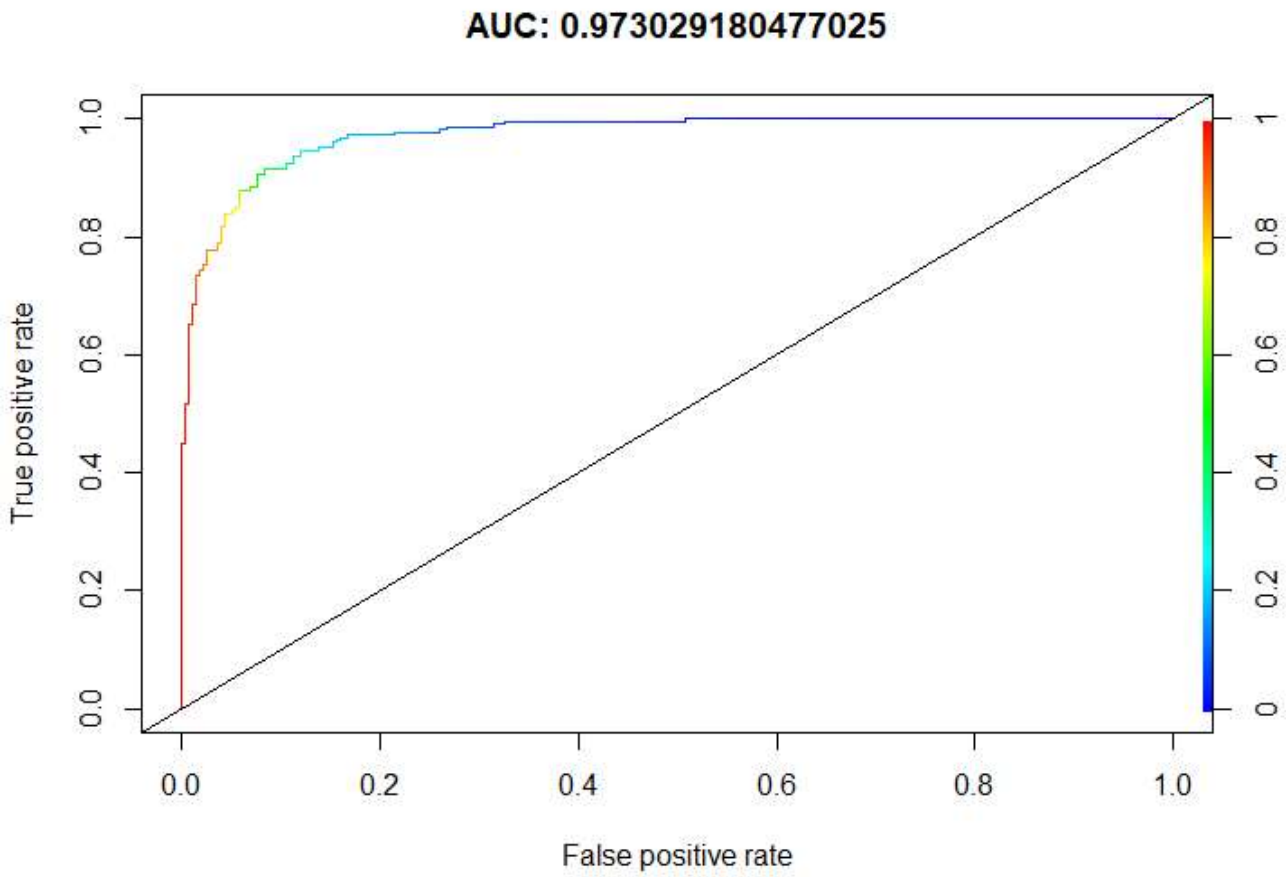
As can be seen, the accuracy obtained (of 92%) is excellent compared to the No Information Rate of 55%; i.e., it means that the model is significantly more accurate than the Naive prediction (which, for each instance, simply returns the most frequent class, as it is most likely).

The decision result is very satisfactory, the model was able to correctly classify the observations in that class when it said they belonged to that class; in fact, it has 93% accuracy for the "satified" class, that is, it correctly classified 93% of the observations it said belonged to the "satified" class.
The recall result is also very good, the model has a recall of 89% for the "satisfied" class, meaning that it correctly classified 89% of the observations that actually belong to the "satisfied" class.
It is interesting to compare the results against the other machine learning model chosen, namely Neural Network, the data are really very similar, this indicates having chosen a good dataset and the most important features for training and prediction.

4.2. Roc Auc

An ROC curve plots TPR (True Positivity Rate) versus FPR (False Positive Rate) at different classification thresholds. By lowering the classification threshold, more positives are classified, thus increasing false positives and true positives.

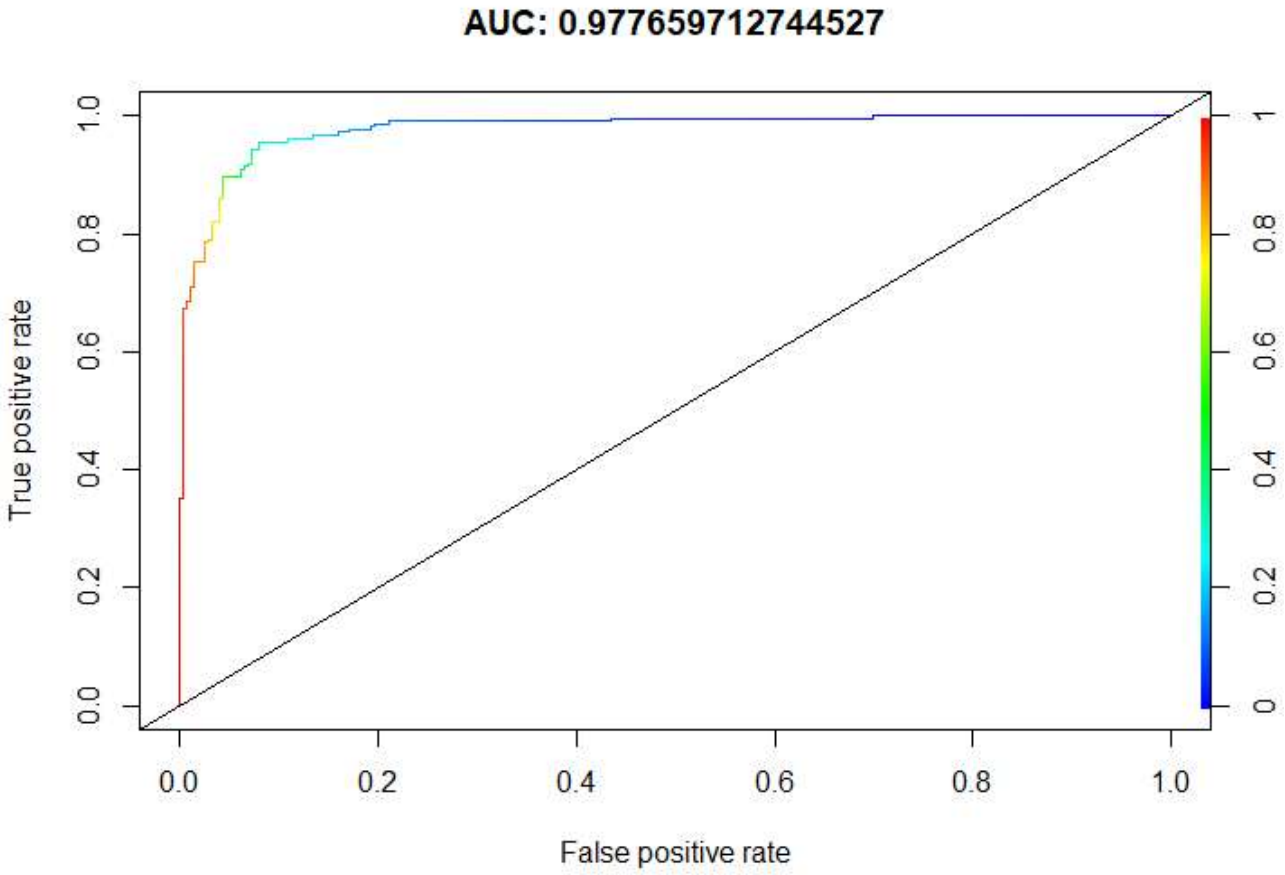
4.2.1. Neural network



Figures 5: Graph showing the performance of the Neural Network on the test set

The graph presented above allows us to show the classification ability of the Neural Network model to make distinctions between the two classes. In this case the value is very close to 1 and is satisfactory.

4.2.2. SVM



Figures 6: Graph showing the performance of SVM on the test set.

The graph presented above allows us to show the classification ability of the SVM model to make distinctions between the two classes. In this case the value is very close to 1 and is satisfactory.

Index of Figures