CLASSIFICATION REPORT

Our team:
Bonavita Luca, 964220
Borgonovo Giorgio, 971456
Cirignoni Niccolò, 968373

1. Introduction

In this report we will discuss the development steps of a model for predicting whether a future customer of the website Yoyo.com will be satisfied or not, according to historical customer opinions and the characteristics of their purchases. The report will stick to the steps that the team followed along the development of the final algorithms, which will be ultimately evaluated according to their f1-scores and overfitting levels. We followed an iterative approach, trying to use different versions of the dataset (given by different educated decisions on how to treat missing values and outliers) to see which results they generated, and then decided for its best version according to the scores given by each format.

2. Preliminary Cleaning

2.1. Elimination of null variables

To begin with, we searched for missing values, and we found 3868 null values for the “Age” attribute. As there was no expert available to perform an inspection, and we were not able to give any meaning to the values, there remained two main strategies to treat them: eliminate the rows or substitute the missing values.

Concerning the second method, we tried an automatic replacement approach by taking the mean and standard deviation for both the distributions of age (without missing values) for satisfied and not satisfied customers. Then we substituted the null values with a random normal distribution based on these metrics, and we noticed that the distributions of age for 0s and 1s (separated) were similar to the ones without null values, so it seemed to be a useful transformation. However, when we tested algorithms on this data, despite the higher number of observations to fit the model, they resulted worse compared to the correspondent algorithms run on the data in which missing values were eliminated; despite their generalization capability increased, the f1-scores were worse. We think that the assumption of normality was too strong, making these “artificial” data not so reliable for our analysis. For this reason, we chose to eliminate the null values and work with less rows.

2.2. Elimination of irrelevant variables

Looking at the variables, we firstly analysed the variable ‘id’ to check if there were rows with the same codes, i.e., different purchases made by the same customer, which could have allowed interesting analyses (e.g., the effect of repurchasing on satisfaction). We found just two rows with the same code, but with incoherent data: one is a 37-year-old woman, while the other one is a 25-year-old male; therefore, we considered it just an error, and kept both observations. Being all the ids different, the variable is not relevant for the prediction, so we decided to drop it.

3. Data Exploration and Cleaning

3.1. Categorical variables

To begin with, we compared the distribution of the categorical variables distinguishing between satisfied and unsatisfied customers. We observed that, among the 4 categories (“Gender”, “Customer Type”, “NewUsed” and “Category”), only “Customer Type” seemed to be relevant as it was the only variable that showed a difference in the distribution of the target among its values, “premium” customers look like to be more satisfied than the “not premium” ones. Consequently, this was the only variable kept in the following steps and a relative dummy variable was created.

3.2. Numerical variables

Then, we looked at the numerical variables by performing firstly a distribution analysis, then a univariate analysis and finally a bivariate analysis. In the first one, we noticed that: “Age” seemed to have a normal distribution (with values greater than 18), “Price” and the two delays variables had a power index distribution, and the survey variables had a quite normal distribution but very asymmetric with a mean near 4. The transformation of these distributions will be treated in the following paragraph. In the univariate analysis, we noticed that the variables showed variegated distributions according to the satisfaction of customers. In particular:

- In “**Age**”, both young and old customers appeared to be less satisfied, while middle-aged customer were prevalently satisfied.
- In “**Price**”, the higher the product price, the higher the satisfaction
- In almost all the **survey** variables, a higher score translated into a higher satisfaction, but we also noticed some strange variables behaviours. In “Product Description Accuracy” the trend was counterintuitive, as higher scores brought to lower satisfaction. Another atypical variable is “Manufacturer Sustainability” in which there was a strong peak of dissatisfaction for medium-high scores (both these results are deeply described in “Appendix B”, where information coming from univariate analysis and algorithms are merged to generate some business-related insights).
- In the **delays** variables, as expected, a lower number of days of delay brings to an higher satisfaction

3.3. Standardization

Concerning the normalization and scaling of the data, we tried different options to better perform it. We found as the best the normalization function “Box-Cox” which optimizes this process by finding the best parameter to transform data. For this reason, we decided to apply this transformation to all our variables. After this, we performed a scaling to reach a normal distribution with mean 0 and standard deviation 1 for all the variables. Both the scaling and lambdas (the Box-Cox parameters) were saved in order to make the prediction for the second delivery of the assignment faster.

3.4. PCA

In this phase, we performed some principal component analyses to see if it was possible to reduce the dimensionality (thus also the computational effort needed) without losing too much variability, while still being able to interpret the newly generated PCs. First, we tried to compute a PCA on the whole dataset, but we got a flat distribution of explained variabilities without any predominant ones. Nevertheless, we tried to explain the meaning of each component, but it was very hard to find meaningful interpretation to them. Thus, we decided to perform this analysis only with some specific classes of variables (e.g., survey-related variables, delay-related variables).

Starting with the survey-related attributes, we found the same results as for the whole dataset. After that, we tried PCA with the two delays-related variables ("Shipment Delay in Days" and "Arrival Delay in Days"); thanks to their very high correlation, we obtained a first PC which explained more than 90% of variability, thus we kept only that principal component. On top of that, were able to give it an interpretation: since it referred to delays in the expedition process, we named it "Delays".

3.5. Outliers Elimination and Split train-test

Following the previous analysis, we noticed from the box-and-whiskers plots some possible outliers for the categories: "Product description accuracy", "Helpfulness of reviews and ratings" and "Delays" (very low grades in the formers and very high delays in the latter). Considering their low numerosity and low relevance (assessed by looking at the target distribution in the univariate analysis) we decided to try to cut them off and check if the models' outcomes were improving. More specifically, we decided to try to eliminate them just in the training set, so right after the training-test separation. This way, the test set could provide a solid proxy of reality, while the training set could generate better generalized models. After testing it with our algorithms – despite a lower overfitting – we noticed a slight deterioration of the f1-test score with respect to the base case, so we decided to keep the outliers and go back to the base case.

In terms of splitting training and test set, we decided to use a 70-30 split due to the huge amount of data.

4. Models

At this point, we were ready to start the predictive models' construction. We divided this part into 3 main phases:

1) **Functions definition**: as the process of tuning the models' parameters is very repetitive, we defined two functions, namely "hyperp_search" and "roc", with the goal of making the code more readable and the process faster. The first function preforms a gridsearch (given some models and its parameters) and prints its results (f1 and confusion matrix), while the second one plots the ROC curve and the AUC parameter. It is worth mentioning that to choose k in the k-fold CV, we saw that the leave-one-out method was too computationally tough given the high number of

observations of the dataset, but at the same time we wanted to get the most stable model possible; for this reason, we analysed results with different values of k for different algorithms and found in K = 3 a good trade-off between them. In addition, we noticed in most models that higher values of K lead to a significant difference between the average f1-test of the cross validation and the f1 computed on the test set.

2) **Models' parameters tuning:** when functions were ready, we used them to examine how different algorithms performed by trying out different parameters values. For every algorithm we followed the same methodology: first, we set very wide ranges of parameters to feed the hyperp_search function grossly understanding which assortments of values brought to adequate results, both in terms of f1 score and overfitting; then, we changed the parameters one by one within narrower ranges to meticulously tune them. A table containing the best parameters we used can be found in Appendix A.

3) **Stacking and Voting classifiers**: the goal of phase 2 was to perfect the algorithms' results to compare how they perform with respect to each other, to lay the foundations for building two new classifiers able to consider the prediction of more than one algorithm at the same time. Among the various kinds we found two types of ensembled algorithms that, giving the previous models with their respective best parameters as an imput, obtained a slightly better f1 scores than the ones provided by the single ones. The first one, the stacking, considered all the 8 previous models, while for the second, the voting, we discovered that it performed better with only the best 3 models obtained in the second phase. In the end the final choice for the prediction algorithm was the stacking classifier with an F1 score of 0.846 over the voting classifier with 0.842.

5. Appendixes

Appendix A:

Algorithm	**F1_test**	**Delta F1_test and F1_Train**	**Prec**	**Rec**	**Classification Accuracy**	**AUC**
k-nn	0.800	0.025	0.828	0.775	0.833	0.9
Tree	0.791	0.032	0.820	0.766	0.826	0.90
Naive Bayes	0.705	0.006	0.699	0.711	0.743	0.82
LogReg	0.717	0.005	0.743	0.692	0.764	0.83
SVM	0.813	0.033	0.863	0.768	0.847	0.91*
Neural Networks	0.839	0.021	0.864	0.815	0.865	0.93
Random Forest	0.802	0.023	0.884	0.730	0.841	0.91
AdaBoost	0.832	0.031	0.875	0.793	0.861	0.91
Stacking	0.846	0.028	0.854	0.839	0.868	0.91
Weighted, Soft Voting Classifier	0.842	0.32	0.877	0.808	0.868	0.93

We consider the delta between the f1_train and the f1_test as the overfitting proxy, which we tried to keep below 0.0035, an arbitrary threshold. All the values of the f1_test in the table are lower than that threshold. Computing the AUC for SVM has resulted (unexpectedly) too hard for our machines, and therefore we left the empty cell on the table. However, for parameters close to the ones that were finally picked, AUC for SVM was around 0.91.*

Appendix B: Interpretation of some results obtained

With logistic regression, classification trees, and random forest we were able to interpret some results relatively to the importance of single attributes to predict "Satisfaction". All three models agree on finding "**Helpfulness of reviews and ratings**" as the most important variable to predict the target value, and "**Customer Insurance**", "**Additional Options**", and "**Product Description Accuracy**" as slightly worse attributes, but still very relevant for the prediction. More specifically, as shown by the Logistic Regression weights, "Customer Insurance" and "Additional Options" contribute positively to the probability of an observation to be classified as "Satisfied", while "Product Description Accuracy" contributes negatively to it. This is coherent with the univariate analysis graphs, which show that the deeper the product description, the more likely were customers to be unsatisfied. The same coherence between these results can be found for the other three abovementioned variables as well, for which if bad grades were given in the survey by customers, they were likely to be unsatisfied, and vice versa. In fact, customers that found wide additional options, a good insurance, and good reviews on the products were more likely to be satisfied.
The variables that impacted the least the final prediction model are "**Delays**" (created with the PCA) and "**Manufacturing Sustainability**", which, coherently with the univariate analysis, do not bring relevant insights on whether the customers were satisfied or not, as their distributions distinguished for 0s and 1s are pretty much the same.