

**ABSTRACT**

Being able to predict whether ot not a booking might be cancelled has such a great importance for the profit and survival of an hotel in this new era, where cancellations and changes of a reservation are so easy to perform. In the context of this research, building a predicitve model that is able to foresee if a guest will or will not cancel would be a great help for new investors. In order to achieve this model, data provided by the hotel was preprocessed, and the relevance of the different features was accessed through Recursive Feature Elimination with Cross-Validation. After this selection the model applied was a gradient boosting classifer with an ada boost classifier which achieved an accuracy around 80% on the valiadation set. With this good performance it was possible to conclude that predicting cancellations of an hotel is feasible and that these results might even be increased with a greater training set and more features added.

**KEYWORDS**

Machine Learning; Predictive Modelling; Gradient Boosting Classifier; Ada Boosting Classifiers; Overbooking; Data Science

**INTRODUCTION**

For a little over forty years, Hotel California was a small but well-known hotel in downtown Lisbon. While in the past it used to be a touristic landmark where a room had to be booked months in advance, with the passing of the years, the evolution of modern technologies and the Covid-19 pandemic, cancellations started to arise more frequently, leading to the permanent closure of this iconic space in Portugal’s capital city.

Similarly, over the last decades, hotels from all over the world had to develop strategies to overcome the undesirable effects of new touristic operators, overbooking and annulment policies; most often ending up choosing to apply more severe cancellation fees. Nonetheless, by making use of Machine Learning predictive models, the new investors of Hotel California, planning to open it back to the public in 2025, have decided to create an adequate overbooking strategy to mitigate these consequences.

In the context under study, the predictive model should be able to foresee whether a booking will or will not be canceled, in order to prevent the over or under booking of the hotel. Therefore, averting both the deterioration of the hotel’s reputation and the increase of the costs destined to the reallocation of guests which could not stay at the hotel due to overbooking. 1,2

**BACKGROUND**

Search methods are very useful since they ease the process of try and error required by both feature selection and modelling. The search methods described were studied during this project and are briefly explained above.

**RFECV**

RFECV also known as Recursive Feature Elimination with Cross-Validation is a feature selection WRAPPER method often applied in machine learning projects. It bases its performance on the use of cross-validation to remove, one at a time, the worse feature - in terms of prediction of the target variable of the dataset - of a model. Furthermore, and akin to what happens in the RFE, the features are evaluated at each iteration of the RFECV, depending on the type of problem, by either a classification or regression estimator and a scoring function.

Note that on some occasions, the RFECV only stops when the model’s performance reaches an insignificant value. 3

**SelectKBest**

SelectKBest is a feature selection filter method which has the purpose of selecting a pre-determined k number of best features based on a certain scoring function. Therefore, assessing the statistical significance of the relationship between each feature and the outcome variable.

In summary, through this technique, each variable is evaluated through an evaluation metric (f\_classification or f\_regression) and only the ones which reveal high scores are kept for the final model (to, consequently, be applied to new data). 4

**Mutual Information**

Mutual information (mutual\_info\_classif or mutual\_info\_regression) is a filter selection method that analyses the level of dependency between two variables of the dataset (only selecting the most correlated one in comparison to the target). Additionally, it is typically used with SelectKBest.

This method uses a ranking system, by ranking the best – most informative - k selected variables in comparison to the outcome (be aware that k is the pre-defined and desired number of features in the final ML model).

Furthermore, the quantity of mutual information between two variables can be obtained by the following equation: I(X;Y) = H(X) – H(X|Y), since it represents the “amount of information one can obtain from one random variable given another”.5.

**SelectFromModel**

Another filter method which is frequently used in the process of feature selection is SelectFromModel. It relies on the selection of the most significant variables for the prediction of the target of the dataset; all of this based on a previously defined threshold either of the coefficients (coef\_) or importance of the variables (feature\_importances\_) towards a certain specified model. Only the features which present values above the defined threshold are kept for the final model.4

SelectFromModel is often used with models that have built-in parameter destined to feature selection; for instance: linear models with l1 regularization, decision trees or random forests. 4

Please keep in mind that all the above described feature selection methods can be implemented using packages from the scikit-learn library.

**Extra Trees Classifier**

Also known as extremely randomized trees, this classifier works similarly to random forests such that a random subset of candidates features is selected. However, in this algorithm, the threshold is computed randomly for each candidate feature and the one who performs better (based on the information gain and entropy) is used as a splitting rule. Even if there is a slight increase on bias, it does reduce the variance of the model, enhancing accuracy as well as decreasing execution time. 4,6,7

**Bayes Search / Bayesian Optimization**

Bayesian optimization is, like the name infers, an optimization algorithm that aims to improve black box-based models. It uses prior assessments to update the quality of the model, improving the decision-making process of where to evaluate later. One of its common uses, that was used during this project, is the hyperparameter tunning for models such as neural networks. 8  Opposite of grid search, it does not test all parameter values; instead, a fixed number of parameters is set and sampled. 4

Besides searching methods, several models were also studied in order to achieve the best predictive power possible. Each studied model is briefly explained below.

**Support Vector Classifier (SVC)**

SVC is considered a supervised machine learning technique used for classification problems. The fundamental goal of SVC is the maximization of the distance between two different classes. To understand this algorithm is important to have in mind four basic concepts that will be deeply explained next.9

First concept is called the separating hyperplane and it consists of grouping the observations according to the values of the outcome variable (i.e. in this case: cancelled or not cancelled) in clusters. Then, according to how many dimensions there are, a point, line or plane is defined to separate those clusters. Because it is a general term, that works for a high-dimensional space, it’s commonly referenced as “hyperplane”. (10) Nevertheless, one might wonder where to draw this hyperplane, since multiple options can be considered. In this algorithm, the theory behind this choice is called “The maximum-margin hyperplane”. Briefly explained, this theory suggests that the line (hyperplane) that separates both classes should have a margin. This margin is merely the distance between the line and the nearest expression vector, which in turn becomes the “maximum-margin separating hyperplane”. With this concept, SVM maximizes its ability to predict new and unseen observations. Another important concept in this algorithm is the soft margin. Because not all data can be divided so clearly with a straight line, the need of some “misclassifications” is important. The soft margin allows one or more observations to be grouped on the wrong side of the hyperplane (considered outliers). Since no one intends to have a model full of errors, the number of possible misclassifications is controlled by a parameter defined by the user, resulting in a trade-off between a good margin of error (misclassifications) and the margin size. Finally, the last concept to have in mind is the kernel function. The latter makes it possible for this algorithm to work on a low-dimensional set as it would on a set with higher dimension, however it’s important to have in mind that too many dimensions will lead to overfitting. 9

**Histogram-based Gradient Boosting Classifier (HistGBDT)**

HistGBDT is a relatively recent modification to the original algorithm “Gradient Boosting Decision Tree” (GBDT) which is out of the scope of this project[[1]](#footnote-2). The goal of this new algorithm is to not only improve the predictive power of GBDT but also fasten the process of training in big datasets. These improvements come from grouping the variables in an histogram which will allow for an optimal split without the cost of such computational expense. 11 This algorithm presents some unique advantages over other machine learning models such as the support of missing values (saving time on the pre-processing phase) and a less robust pre-processing stage, so it won’t suffer penalization is a bad pre-processing strategy is applied. 10

**Passive Aggressive Classifier**

Passive Aggressive Classifiers are usually used for large data sets. The name “Passive-aggressive” is due to the fact that the algorithm is considered passive when a correct result is predicted and aggressive when the result is miscalculated. The point of this classifier is to correct the miscalculations by updating the weight vector just enough to predict the input properly and better classify the future observations. 12,13 It is similar to Perceptron such that both reduce dimensionality of the problem, however it allows for a control parameter that balances the size of the margin and the number of misclassifications (as SVC). 4, 14

**Voting Classifier**

To improve predicitions one can make use of emsembling methods such as Voting Classifier. This classifier joins different learning algorithms with the purpose of increasing the predictive power of the model. To create it, the predictions of each classifier are evaluated by the algorithm and each one will receive an amount of votes. Then, this classifier will determine which was the algorithm with the greater number of votes. 15 For this reason, the algorithm used by sklearn is considered a “soft/majority Rule classifier for unfitted estimators”. 4 Since it uses all predicitions made by the different learning algorithms, this “majority voting“ classifier is considered an hard voting classifier. Moreover, by taking in consideration the individual errors of the included classifiers, the accuracy of this ensemble method will also be improved. 15

**METHODOLOGY**

Prior to the beginning of the data exploration and machine learning model implementation, a minor literature review was conducted concerning papers included in the Google Scholar Database and which will be rightfully identified in the References section of this document. Note that only papers which were written in English were considered. The search included Keywords such as “predictive”, “model”,” hotel”, “predicting” and “cancellations”.

Firstly, it is important to mention that the project was developed following a SEMMA framework – Sample, Explore, Modify, Model and Assess - which was chosen due to it being the most appropriate project outline to solve the problem under study. 16

Additionally, it is also of great significance to mention that the totality of the project was developed using python programming language.

**Sample (Data Integration)**

The first step to be performed was the importation of the needed python libraries/packages to develop the project. Only after this was concluded were the csv files imported. These contained the training and testing samples with data from Hotel California, regarding the year of 2016, as well as the sample submission csv file to where the final predictions were exported at the very end of the project.

**Data Exploration and Modification**

Next, the exploration and modification of the data were performed in the same order and way for both the training and testing samples.

Firstly, a copy of both datasets was created to prevent any accidental changes to the original data. Then, the summary statistics, information on the type of variables and possible existence of missing values in the dataset’s features were checked. After that, the two variables who were constant (only had one unique value) were dropped from both datasets.

Furthermore, other pre-processing methods were applied to the data in order to verify the presence of duplicate values (for both observations of a single column and complete rows); due to the fact that it had no duplicates, “BookingID” was transformed into an index column in both datasets. In addition, the verification of the presence and treatment of data inconsistencies was also conducted. The most relevant cases were: first-time guests which already had previous reservations – transformed into true FirstTimeGuests – and not first-time guests but also with no previous reservations – converted into new first time guests.

Afterwards, the creation of variables to ease the feature selection and modelling process was conducted; division of the training data into predictors (X) and outcome variable (y); observations of the predictors (X) were separated into training and validation sets (X\_train, y\_train, X\_val,y\_val) - considering 20% as validation data (random state = 42).

Finally, some graphics were plotted as a way to better visualize the data and possibly identify patterns and relationships between the several features of the dataset.

**Feature Selection**

In advance of the feature selection being initialized the numerical features of both training and validation sets, as well as of the testing sample were scaled. For this step, three scaling methods were initially used (MinMax, Standard, Robust). However, after some trials it was verified that the one which returned the best results was Robust Scaling, most certainly because it is able to confer robustness against possibly existing outliers in the dataset. Nevertheless, this allowed all numerical values of the datasets to be measured through the same scale and be rightfully compared to each other.

On a first stage, the feature selection was conducted only with previously learned methods, such as: RFE, Lasso Regression and Spearman Correlation Matrix. It was also in this phase that it was noticed that, in the final set of variables, there were still highly correlated features that were being included. Therefore, this process was redone by removing, on a first occasion, the highly correlated variables from the datasets (using the coefficients resulting from Extra Trees Classifier). Then new feature selection filter methods were applied, such as RFECV, SelectKBest/Mutual Information and SelectFromModel (previously explained in the Background section of this document).

In summary, when the comparison between the outputs of all these methods were compared the one which resulted in the best accuracy score was the RFECV set of features. Therefore, this was the considered one for the final set of features to be applied in the model.

Finally, the variables with only the best features for the prediction of the outcome “Canceled” were created (X\_train\_out, X\_val\_out,X\_rb\_out,test\_rb\_out).

**Model**

The main goal of this project is “ to identify the machine learning algorithm that is best-suited for the problem at hand; thus, we want to compare different algorithms, selecting the best-performing one as well as the best performing model from the algorithm’s hypothesis space”17. Hence, the first step understood which type of models could be applied to this classification problem of predicting the values of the target variable of the dataset – “Canceled”.

It is of great importance to mention that the hyperparameter optimization carried out in this project was performed “to increase the predictive performance by tweaking the learning algorithm and selecting the best performing model from a given hypothesis space.” 18 Due to the fact that Grid Search (GS) “makes a complete search over a given subset of the hyperparameters space of the training algorithm”17, and Random Search (RS) “overrides the complete selection of all combinations by their random selection” 18 the latter is several degrees less complex than the GS hyperparameter tuning technique. Hence, this stage of the modelling phase was completed using Random Searches.

In order to reach the nearest to the best and then the final predictive model, models such as: Neural Networks (Multi-Layer Perceptron classifiers), Decision Trees, KNN Classifiers and Supported Vector Machines (SVC) were created and then tested with the feature selected training dataset. Other modelling techniques like ensemble – namely: Random Forests, Voting, Stacking, Bagging, Hist Gradient Boosting, Gradient Boosting and Ada Boosting Classifiers were also applied.

After almost sixty attempts, the final predictive model was found. The models were evaluated through a metrics function which will be briefly explained below and the amount of overfitting the model revealed itself to have was also taken into account. This final model is a combination of a Gradient Boosting with an Ada Boosting Classifier.

**Assess**

The performance of all created models, as well as of the final one was assessed using a metrics function that makes use of a classification report and a confusion matrix for both the training and validation datasets. Subsequently, if a model revealed itself to have an accuracy (in the f1-score) over 78% the predictions would be applied to a test sample and submitted on Kaggle to reach a more truthful performance of the model under creation.

**RESULTS**

In what concerns data exploration, neither missing values nor duplicates were found. Several data inconsistencies were found during the exploration process of the data. However, with the information provided to the authors, only two of them could be rightfully treated. 21 Guests belonged to the group of first-time guests even though they had already stayed in the hotel before; as so, they were not considered as first-time guests by the authors. The opposite was also verified, guests that had never done a reservation were not considered first-time guests and so this was treated too by converting them into first-time guests. Descriptive statistics were accessed and can be found in table 1 and 2 (appendix).

Still in the pre-processing stage two variables (“ArrivalYear” and “CompanyReservation”) were excluded right away since they were constant features (all observations had the same value) which do not add any relevant information to the future predictive model.

Due to the fact that all variables were considered numerical, all the tested algorithms managed to include all of the variables of the datasets.

The first feature selection method applied was, as mentioned before, Spearman Correlation, and its results can be found in figure 1 (in the appendix). In the following table, the most correlated variables are presented.

|  |  |  |
| --- | --- | --- |
| Variables | | Spearman Correlation Coefficient |
| FloorAssigned | FloorReserved | 0.81 |
| PreviousReservations | PreviousStays | 0.9 |
| CountryOfOriginHDI (Year-1) | CountryofOriginAvgIncomeEuros (Year-1) | 0.96 |
| CountryOfOriginHDI (Year-1) | CountryofOriginAvgIncomeEuros (Year-2) | 0.96 |
| ArrivalWeekNumber | ArrivalMonth | 1.0 |
| DailyRateEuros | DailyRateUSD | 1.0 |
| CountryofOriginAvgIncomeEuros (Year-2) | CountryofOriginAvgIncomeEuros (Year-1) | 1.0 |
| PreviousReservation | FirstTimeGuest | -1.0 |
| PreviousStays | FirstTimeGuest | -0.89 |

As it was mentioned above, in order to decide which of the highly correlated variables would be deleted from the dataset the Extra-Trees Classifier was computed. It considered the following variables as not relevant to be included in the predictive model: “ArrivalMonth”, “FirstTimeGuest”, “CountryOfOriginAvgIncomeEuros(Year-2)”, “DailyRateUSD”, “PreviousStays”, “FloorReserved”, “PreviousStays”, “CountryOfOriginAvgIncomeEuros(Year-1)”.

Text

Description automatically generated with medium confidenceAfterwards, the other types of feature selection were applied and in the figure below it is possible to see their returned results.

Figure 1: Outputs of Feature Selection Methods

The final features to be included in the model and as explained previously resulted from the outputs of the RFECV feature selection filter method. Therefore, that selection can be observed in the list below:

- ArrivalWeekNumber ,

- ArrivalDayOfMonth,

- ArrivalHour,

- WeekendStays,

- WeekdayStays,

- Adults,

- Children,

- Babies,

- AffiliatedCustomer,

- PreviousReservations,

- PreviousCancellations,

- DaysUntilConfirmation,

- OnlineReservation,

- BookingChanges,

- BookingToArrivalDays,

- ParkingSpacesBooked,

- SpecialRequests,

- PartOfGroup,

- OrderedMealsPerDay,

- FloorAssigned,

- DailyRateEuros,

- %PaidinAdvance,

- CountryofOriginHDI (Year-1).

Finally, the final model – Gradient Boosting Classifier with an Ada Boosting Classifier – resulted in accuracy scores of 0.89 for the validation dataset and 0.80 for the training dataset. Thefore, showing quite a little bit of overfitting. Nevertheless, it was still the model which returned the best result in terms of Kaggle submission, f1\_score of 0,79377.

**DISCUSSION**

Just like Antonio et. Al. (2017), feature selection was a key role in the process of creating a good predictive model. The mean F1 score in the test set achieved was 79% meaning that the classifier performed well in terms of both precision and recall. This results confirm that the possibility of predicting whether or not a guest will cancel with high probability, is feasible. This will allow that the new investors make some adjustments in order to avoid a cancelation such as offer a discount or a service. 2

Notice that this model functions well for this hotel but the same might not be true for another hotel with different kind of guests; reinforcing the importance of feature selection.

After several tries, best predictive model was considered the one who grouped a Gradient Boosting Classifier with an Ada Boost Classifier, achieving a good accuracy on the validation set as well as a good average F1 score in the test set. Even though the best model achieved did present quite a good predictive power, it also produced a little bit of overfit that was very hard to diminish. A smaller training set and existence of noise and outliers might be the cause of this overfit, even though these factors were out of the control by the authors until a certain point. Data shall be retrieved as properly as possible and in a large scale in order to produce a good predictive model and so it is important for the hotels who want to use machine learning techniques to be aware of that.

This research proved that for this hotel and with the data provided, it is possible to achieve a model with good predictive power. It would be interesting to apply the same model for a larger database, maybe integrating more data from other hotel, and check its performance. It would also be interesting to include more distinct features retrieved by the hotel, such as existence of a view in the room reserved, season, type of room reserved (single, double, suite), price of the room, allowance to other parts of the hotel such as the pool or spam nationality of the guest (to compare with the country where the hotel is) , among others, and test whether it would make a difference on the performance.

**CONCLUSION**

With the application of Machine Learning methods it was possible to achieve the aim of this research. By applying WRAPPER methods the identification of the relevant features was achieved and played a key role in the performance of the final model. With the model achieved the assumption made in the beginning of this research about the likelihood of a cancellation was proven feasible. More research with different hotels and features is still necessary in order to achieve more modularity.

**REFERENCES**

1. António, N., Almeida, A., Nunes, L. (2019). Predictive models of hotel booking cancellation: a semi-automated analysis of the literature. *Tourism and Management Studies*. 15. 7-21. 10.18089/tms.2019.15011.
2. Antonio N, Almeida A, Nunes L. (2017) Predicting hotel booking cancellations to decrease uncertainty and increase revenue. *Tourism & Management Studies* 13(2) 25-39
3. Misra P., Singh Yadav A. (2020) Improving the Classification Accuracy using Recursive Feature Elimination with Cross-Validation. *International Journal on Emerging Technologies.* 11(3): 659-665(2020)
4. Pedregosa *et al* , [Scikit-learn: Machine Learning in Python](https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html)*.* *JMLR 12*, pp. 2825-2830, 2011.
5. Witten, H., Frank E., Hall, M., Pal., C. (2016) Data Mining: Practical Machine Learning Tools and Techniques. (4th ed.) *Morgan Kaufmann Series in Data Management Systems.*
6. Baby D., Devaraj S., Hemanth, J., Anishin Raj M. (2021) Leukocyte classification based on feature selection using extra trees classifier: A transfer learning approach. *Turkish Journal of Electrical Engineering and Computer Sciences* 29 2742-2757
7. Lanjewar, M.G., Parab, J., Shaikh, A.Y., Sequeira, M. *(2022).* CNN with machine learning approaches using ExtraTreesClassifier and MRMR feature selection techniques to detect liver diseases on cloud. *Cluster Comput* doi.org/10.1007/s10586-022-03752-7
8. Ma, L., Cui, J., Yang, B. (2019) Deep Neural Architecture Search with Deep Graph Bayesian Optimization, *IEEE/WIC/ACM International Conference on Web Intelligence (WI)* pp. 500-507.
9. Noble, W. (2006). What is a support vector machine?. *Computational Biology*. 24(12), 1565–1567. doi:10.1038/nbt1206-1565
10. Ong, Y.J., Zhou, Y., Baracaldo, N., Ludwig, H. (2020) Adaptive Histogram-Based Gradient Boosted Trees for Federated Learning. *CoRR abs*. 2012.06670
11. Kashifi, M., Ahmad, I. (2022). Efficient Histogram-Based Gradient Boosting Approach for Accident Severity Prediction With Multisource Data*. Transportation Research Record*, 2676(6), 236–258.
12. Matsushima, S., Shimizu, N., Yoshida, K., Ninomiya, T., Nakagawa, H. (2010). Exact Passive-Aggressive Algorithm for Multiclass Classification Using Support Class. 303-314. 10.1137/1.9781611972801.27.
13. Gupta, S., Meel, P. (2021). Fake News Detection Using Passive-Aggressive Classifier. In: Ranganathan, G., Chen, J., Rocha, Á. (eds) Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems, vol 145. *Springer*, *Singapore*. doi.org/10.1007/978-981-15-7345-3\_13
14. Chang, C., Lee, Y., Pao, H. (2010) A Passive-Aggressive Algorithm for Semi-supervised Learning, 2010 *International Conference on Technologies and Applications of Artificial Intelligenc.* pp. 335-341, doi: 10.1109/TAAI.2010.61.
15. Kumar, U., Nikhil, M., and Sumangali, K. (2017) Prediction of breast cancer using voting classifier technique, *2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, pp. 108-114, doi: 10.1109/ICSTM.2017.8089135.
16. Shafique, U., Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International journal of innovation and scientific research*, 12, 217-222.
17. Petro L., Liashchynskyi , (2019) P. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *ArXiv*. 1912.06059.
18. Raschka, S. (2020) Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv*, 10 November 2020. http://arxiv.org/abs/1811.12808.

**APPENDIX**

Table 1: Test Set Descriptive statistics

Table 2: Train Set Descriptive Statistics



Chart

Description automatically generated

Figure 2: Spearman Correlation Matrix

1. Explained on practical classes [↑](#footnote-ref-2)