**Not Suite Dreams: An Analysis of Hotel Bookings and the Tendency for Patrons to Cancel**

Jacob Redmon, [jredmon@bellarmine.edu](mailto:jredmon@bellarmine.edu)

Christina Porter, [cporter@bellarmine.edu](mailto:cporter@bellarmine.edu)

**ABSTRACT**

In this project, we attempted to train, validate, and predict whether a hotel booking would show as cancelled using various values included in a dataset gathered on hotel bookings by Nuno Antonio, Ana Almeida, and Luis Nunes. We were able to use three models. Overall, the worst predictor was whether a party had children and the best predictor was the time of the hotel stay, and the worst predictor was the amount of children or babies that were included within a party.

1. **INTRODUCTION**

This project uses the k-nearest neighbor machine learning model to determine what factors, if any, can be used to anticipate the cancellation of a hotel booking by a patron.

1. **BACKGROUND**
   1. *Data Set Description*

The data set we used was data on hotel bookings from two hotels in Portugal, “originally from the article Hotel Booking Demand Datasets, written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019.” It, “contains booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things.”

We chose to use this data set because of its nice spread of categorical and numerical columns. The data was originally in two data sets, one for the resort hotel and the other for the city hotel. The original use of the data was for research in revenue management, machine learning, and data mining. The cleaned version of the data can be found here: <https://www.kaggle.com/jessemostipak/hotel-booking-demand>

* 1. *Machine Learning Model*

For this project, we considered using regression, a decision tree, random forest, or K-Nearest Neighbor. Ultimately, we ended up choosing to use the K-Nearest Method because there were so many factors included within this dataset that it was hard to create a good analysis using just the quantitative variables. The K-Nearest Neighbor is a supervised machine learning algorithm that uses data that has already been labeled to predict a value based on unlabeled data. For this method, we assume that an output determined by a set of factors will produce a similar result given the same set of factors again. Thus, if these values are plotted, these values should theoretically be clustered on a graph. Thus, by plotting an unknown value and searching for the closest points around it, or neighbors, we should be able to predict the outcome of the variable.

1. **EXPLORATORY ANALYSIS**

This data set contains 102894 samples with 31 columns, one of which we removed due to lack of data.

Some peculiar findings: the minimum value in the ‘adr’ column (Average Daily Rate) is -6, which must be an entry error as the rest of the row looks normal, the longest booking in advance was 629 days, and the date of arrival was more likely to be the 30th or 31st of any given month. Also, there is not much correlation in the data by looking at the heatmap of the correlation matrix in Figure 1. The most correlated columns were ‘stays\_in\_weekend\_nights’, ‘stays\_in\_week\_nights’, which is unsurprising. There was nothing notable in the histograms for the numerical data.

Shape

Description automatically generated

**Figure 1: Heatmap of the correlation matrix for the dataset’s categorical variables**

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| hotel | object |
| is\_canceled | int64 |
| lead\_time | int64 |
| arrival\_date\_year | int64 |
| arrival\_date\_month | object |
| arrival\_date\_week\_number | int64 |
| arrival\_date\_day\_of\_month | int64 |
| stays\_in\_weekend\_nights | int64 |
| stays\_in\_week\_nights | int64 |
| adults | int64 |
| children | float64 |
| babies | int64 |
| meal | object |
| country | object |
| market\_segment | object |
| distribution\_channel | object |
| is\_repeated\_guest | int64 |
| previous\_cancellations | int64 |
| previous\_bookings\_not\_canceled | int64 |
| reserved\_room\_type | object |
| assigned\_room\_type | object |
| booking\_changes | int64 |
| deposit\_type | object |
| agent | float64 |
| days\_in\_waiting\_list | int64 |
| customer\_type | object |
| adr | float64 |
| required\_car\_parking\_spaces | int64 |
| total\_of\_special\_requests | int64 |
| reservation\_status | object |
| reservation\_status\_date | object |

1. **METHODS**

In this section, we will discuss how we prepared the data for k Nearest Neighbor by completing several different experiments using changes with the split between train, validate, and test using different parameters provided within the hotel bookings dataset.

* 1. *Data Preparation*

When we first started using this dataset, we first had to see what types of data values were contained in each column. While some columns such as is\_repeated\_guest, deposit\_type, or arrival\_date\_day\_of\_month were self-explanatory, some columns such as agent, reserved\_room\_type, or meal didn’t contain a detailed description on how they were categorized. For example, for the reserved\_room\_types, we expected room types classified by whether or not the room was a suite or what type of bed was in the room. Instead, the unique room type values consisted of A, B, C, D, E, F, G, and H, with no indicator to what these values meant. While these values could have been used as-is, we steered cleared of these values because we wouldn’t be able to analyze these values further.

We were able to complete the first eight experiments using quantitative variables; however, to do an experiment with qualitative variables, we had to find a way to turn the categorical variables to numerical because sklearn does not handle categorical values. To do this, we were able to use a dummy set by separating the unique values in each column to their own columns and coding them with true or false. Using this method, if we wanted to stick with using columns from our original dataset, we ended up using more columns to split our experiment with because every column that represented a unique value had to be represented.

* 1. *Experimental Design*

Table 2 describes each experiment that was run. Our goal was to use different parameters in each experiment because we wanted to see if there was a combination of columns that aided our classification more.

Table 2: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | Deciding cancellation by the number of adults, children, and babies with 50/25/25 split for train, validate, and test with n=5 NN. |
| 2 | Deciding cancellation by the number of adults, children, and babies with 70/15/15 split for train, validate, and test with n=5 NN. |
| 3 | Deciding cancellation by the number of adults, children, and babies with 50/25/25 split for train, validate, and test with n=20 NN. |
| 4 | Deciding cancellation by the number of adults, children, and babies with 70/15/15 split for train, validate, and test with n=5 NN. |
| 5 | Deciding cancellation by the number of days scheduled in advanced, length of stay in weeknights, and length of stay in weekend nights with 50/25/25 split for train, validate, and test with n=5 NN. |
| 6 | Deciding cancellation by the number of days scheduled in advanced, length of stay in weeknights, and length of stay in weekend nights with 70/15/15 split for train, validate, and test with n=5 NN. |
| 7 | Deciding cancellation by the number of days scheduled in advanced, length of stay in weeknights, and length of stay in weekend nights with 50/25/25 split for train, validate, and test with n=20 NN. |
| 8 | Deciding cancellation by the number of days scheduled in advanced, length of stay in weeknights, and length of stay in weekend nights with 70/15/15 split for train, validate, and test with n=20 NN. |
| 9 | Deciding cancellation by the customer type, deposit type, and hotel type with 50/25/25 split for train, validate, and test with n = 5 NN. |
| 10 | Deciding cancellation by the customer type, deposit type, and hotel type with 70/15/15 split for train, validate, and test with n = 5 NN. |
| 11 | Deciding cancellation by the customer type, deposit type, and hotel type with 50/25/25 split for train, validate, and test with n = 20 NN. |
| 12 | Deciding cancellation by the customer type, deposit type, and hotel type with 70/15/15 split for train, validate, and test with n = 20 NN. |

* 1. *Tools Used*

The following tools were used for this analysis: Python v3.8.5 running the Anaconda 2020.11 environment via Jupyter Notebooks v6.2.0 for Windows was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 1.1.3, Numpy 1.19.2, Matplotlib 3.3.2, Seaborn 0.11.0, SKLearn 0.20.0, and scipy 1.5.2. These tools were used because they provided the necessary tools to clean, visualize, and train a model with our dataset.

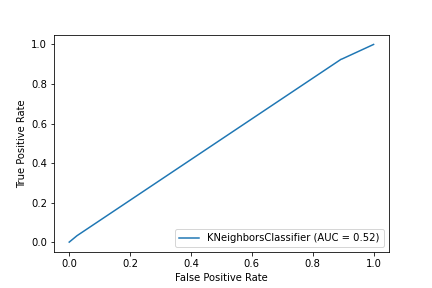
1. **RESULTS**
   1. *Classification Measures*
      1. Experiment 1  
         

Figure 2: ROC Curve for Experiment 1

* + 1. Experiment 2  
       *Chart, line chart

       Description automatically generated*

Figure 3: ROC Curve for Experiment 2

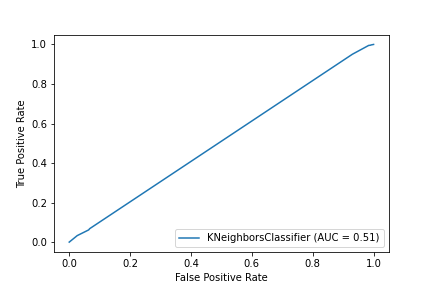
* + 1. Experiment 3  
       

Figure 4: ROC Curve for Experiment 3

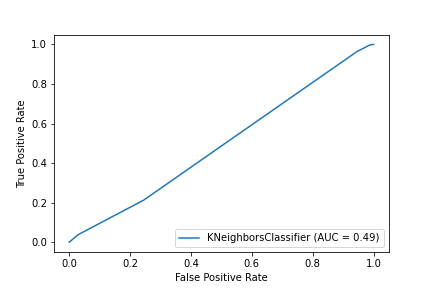
* + 1. Experiment 4  
       **

Figure 5: ROC Curve for Experiment 4

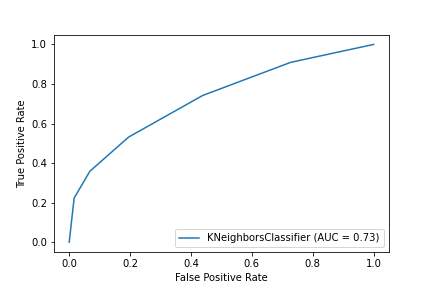
* + 1. Experiment 5  
       **

Figure 6: ROC Curve for Experiment 5

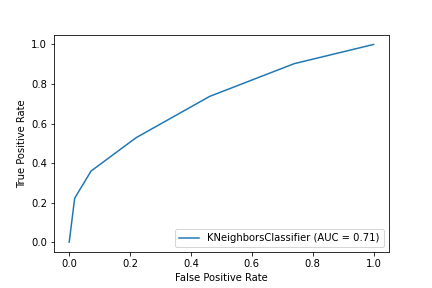
* + 1. Experiment 6  
       **

Figure 7: ROC Curve for Experiment 6

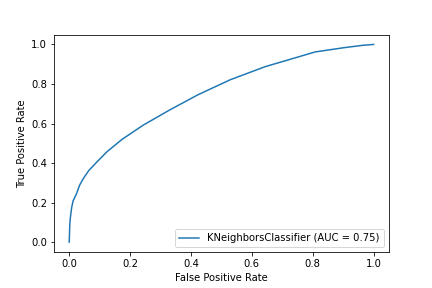
* + 1. Experiment 7  
       **

Figure 8: ROC Curve for Experiment 7

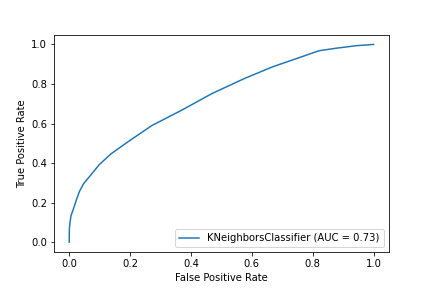
* + 1. Experiment 8  
       **

Figure 9: ROC Curve for Experiment 8

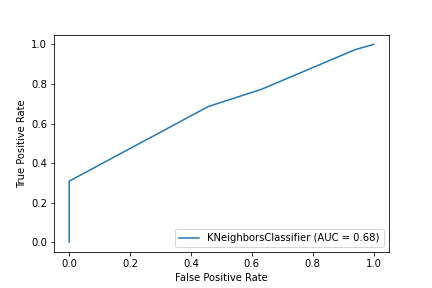
* + 1. Experiment 9  
       **

Figure 10: ROC Curve for Experiment 9

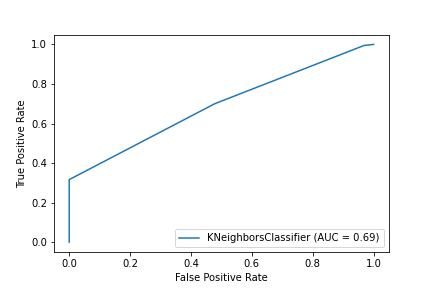
* + 1. Experiment 10  
       **

Figure 11: ROC Curve for Experiment 10

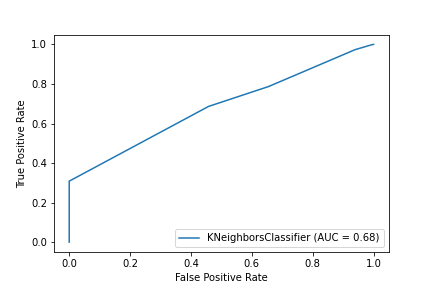
* + 1. Experiment 11  
       **

Figure 12: ROC Curve for Experiment 11

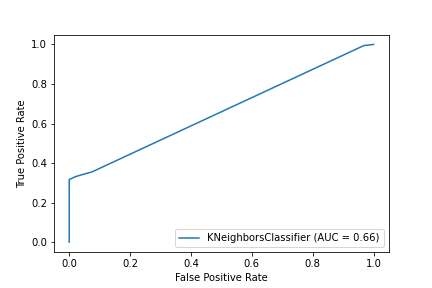
* + 1. Experiment 12  
       **

Figure 13: ROC Curve for Experiment 12

* 1. *Discussion of Results*

The worst model trained was the first model, which consisted of experiments one, two, three and four. As displayed in Figure 2, Figure 3, Figure 4, and Figure 5, our model was split half and half with an area under the ROC curve approximately at .5. Thus, this model was not able to discern between a true positive and a false positive. This means that using the age range in a group or the number of children are not good predictors to determine whether a hotel booking is likely to be cancelled or not. On our initial hypothesis, we thought that the number of children or babies would influence the cancellation of hotel bookings because children have the capability to throw a wrench in plans and quite possibly may cause trips to be cancelled; however, this does not seem to be the case. This could likely be because a lot of planned trips occur either without children or the probability of cancellation due to the number or presence of children aren’t as high as we originally thought.

The third model was slightly better, consisting of experiments nine, 10, 11, and 12. Displayed in Figure 10, Figure 11, Figure 12, and Figure 13, the curve on our model was moved slightly upwards and consisted of slightly more true positive values than false positive values. Yet, I still would not use these categories as accurate predictors to determine the likelihood of a cancelled hotel booking. There wasn’t a set organization to the columns included in this model. Instead of choosing a single factor, we used general booking information to create our model. We believe this caused a split in the decision making that prevented the ROC from growing.

Our best trained model was the second, consisting of experiments five, six, seven, and eight. In this model, we used categories involving time, consisting of the amount of time the hotel was booked before the trip and the length of the stay on either weekends or weekdays. Displayed in Figure 6, Figure 7, Figure 8, and Figure 9, we have decently nice curves with the area under the ROC curve of around .75. Out of these, the best experiment was seven, where we trained 50% of the data and used a value of 20 for k. Out of each of the four experiments in the second mode, experiment 7 being the most accurate makes the most sense because this is a significantly large dataset. Using a large training value risks the possibility of overtraining the data. Additionally, using a larger k value gives a proportionally larger comparison value for a larger dataset.

* 1. *Problems Encountered*

This project went relatively smoothly. We didn’t have any major issues beyond normal cleaning challenges. There was one column in the dataset, company, which was mostly empty, so the column had to be removed. Additionally, we previously discussed some challenges with not knowing the meaning or significance of several columns within the data. One of the most challenging parts of this project was creating the dummy dataset and figuring out how to sift through the additional columns because we ended up having over 200 columns once we created columns based off every unique value in our categorical columns. Not only did this made modeling categorical values difficult, but it was also inefficient and takes up significantly more memory.

* 1. *Limitations of Implementation*

We believe the data we chose for this project worked decently well. There was a good mix of quantitative and qualitative variables and believe that this model is very useful for attempting to predict human behavior, which is one of the largest uses of predictive modeling today. Out of all the values that were included within this dataset, we believe that predicting a cancellation of hotel bookings may not be super useful because there isn’t much that could be used with this data in terms of oncoming bookings but could be suggestive as to new amenities the hotel could offer. For practical use, we might recommend doing a similar model with repeated guests because it could predict potential repeat clientele. Additionally, predicting whether a hotel booking is cancelled could be erratic because most cancellations usually occur due to unforeseen circumstances, which aren’t accounted for in this data. With this project, using the number of booking changes or the number of previous bookings cancelled would be better factors with predicting whether a current booking will be cancelled or not.

* 1. *Improvements/Future Work*

With the models that we used, we would be curious to see the changes in the ROC curve with less prediction or more prediction. Additionally, since this is a large dataset, increasing the value of k could also create some significant changes with the dataset. Additionally, there is a possibility that using the K-nearest neighbor method was not the best approach for this dataset. Since these values are potentially random, we may have gotten better results using a decision tree or even an unsupervised learning method.

1. **CONCLUSION**

In this project, we used the Hotel Booking Demand dataset written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019 to train, validate, and predict whether a hotel booking would show as cancelled using various factors in three different models across 12 different experiments. The first model predicted whether a hotel booking would be cancelled based off the number of adults and children were within the party. The second model predicted a cancellation based off the number of days a booking was created in advance of a trip and the length of a stay, and the final model used random blanket categorical variables that described the hotel stay. Overall, the worst predictor was whether or not a party had children and the best predictor was the time of the hotel stay. While there were some challenges with creating models for this dataset, this project went overall well even though we were not able to find a super accurate predictor of the cancellation of hotel bookings. For the future, we would have continued testing variable categories, split, and k values in this data to find a better predictor for values.

**REFERENCES**

Nuno Antonio, Ana de Almeida, Luis Nunes, *Hotel booking demand datasets*, Data in Brief, Volume 22, 2019, Pages 41-49, ISSN 2352-3409, <https://doi.org/10.1016/j.dib.2018.11.126>. *(https://www.sciencedirect.com/science/article/pii/S2352340918315191)*