Cracking the cancellation code: Factors that drive reservation drop offs

INN Hotels for for Data Analysis Supervised Learning Classification

08/11/2023

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

- Based on the previous hotel's room reservation data, we attempted to identify the factors that influence the cancellation of reservations so that hotel managers could be prepared in advance to replace the reservation if it was cancelled.

- Our analysis shows the highest important factors are the **lead time**, **online reservation, price of the room** and the **number of the special request** of the guests.

- According to our findings, the following characteristics increase the likelihood of a reservation being cancelled:
  - **Lead time:** if the guest made the reservation more than 151 days in advance
  - **Online reservation**: if the reservation was made online
  - **Price:** the higher the price of the room, the more likely the reservation will be cancelled.
  - **Number of special request:** if the guest has few or no special requests

Business Problem Overview and Solution Approach

- A significant number of hotel reservations are cancelled or no-shows, which is a revenue-diminishing factor for hotels to deal with. Cancellations result in losses for hotels, with last-minute cancellations resulting in the **loss of resources** (revenue), **additional costs of distribution channels, renting the room at a cheaper price**, and **human resources to make arrangements for the guests**.

- We created a model using a Machine Learning algorithm that can predict whether a new reservation will result in cancellations or not based on various features of the reservation. When a new reservation is made, the model analyzes whether the reservation will be cancelled or not, and the system can use this prediction to set different policies if the guest wishes to cancel the reservation later.

- In this case, the guests who truly want the room would make the reservation, and we would cancel the reservations that were suspected of being cancelled later.

EDA Results

- On average, reservations for rooms are made **85 days before** the check in.
- The hotel charges an average of **103 dollars** per day to rent a room.
- In some cases, rooms have been rented for free through **complementary** and **online** markets.
- The majority of the rooms have been rented by **couples or families of three**.
- The majority of the guests did not require parking.
- **Meal plan 1 is the most popular**, with almost **no demand for Meal Plan 3**.
- The most popular room type is 1, while room type 3 has a low level of popularity.
- In order, there is a higher demand for renting rooms in **August, September, and October**.
- The majority of the guests had no special requests.
- It is expected that **30%** of the reservations will be cancelled.

Link to Appendix slide on data background check

EDA Results

- The majority of rooms rented through the **complementary market are free**, and the **most expensive** rooms are those rented through the **online market.**
- The reservation made through **complementary** will not be cancelled, whereas cancellations are possible in other cases.
- If a guest has more than 3 special requests, they **will not cancel their reservation**.
- Lower expected price for guests with **no special requests** compared to those with special requests
- The **higher the price** of the rooms, the **greater the likelihood** of cancellation.
- If the reservation is made far in advance of the check-in date, it is **very likely** that the reservation will **be cancelled.**
- Travelling with families does not appear to affect the booking status.
- Cancellations are more likely for bookings of more than **10 days**.
- **Recurring guests** almost never cancel their reservations.

Link to Appendix slide on data background check

EDA Results

- The **busiest months** for reservations are August and October.
- A reservation for **January** has almost no chance of being cancelled.
- Rooms are **cheapest** in the early months, and the **most expensive** between May and September.

Link to Appendix slide on data background check

Data Preprocessing

- The used dataset contained no duplicated or missing values.

- Using boxplots, we discovered that the majority of the features had outliers, but because they appeared error-free, we did not change their values. We only dealt with the outliers in **'avg_price_per_room'** feature and replaced prices greater than **500** with **179**.

- We encoded our categorical features using the **one_hot_encoding** technique during feature engineering to prepare our data for fitting to the machine learning algorithm.

- We used **70%** of the original data as training data and **30%** to test the generated model's performance.

Model Performance Summary

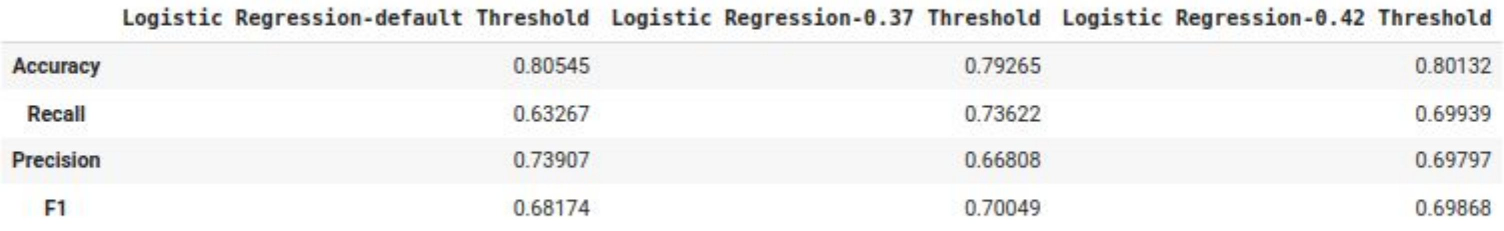
- In order to build our model, we used logistic regression and decision trees. We will present our findings for models generated using both of the algorithms.

- **Logistic Regression:**
  - Initially, we had 27 predictor variables after encoding the categorical values. We used **21 variables** after removing variables with p-values greater than 0.05 iteratively.
  - For Logistic Regression, the most important features for distinguishing the target variables for each observation were '**arrival_year**,' **no_of_previous_cancellations**,' and '**type_of_meal_plan_Not Selected**.'
  - We used the **AUC-ROC** technique to adjust the threshold for making predictions and generated two more models with these settings.

Model Performance Summary

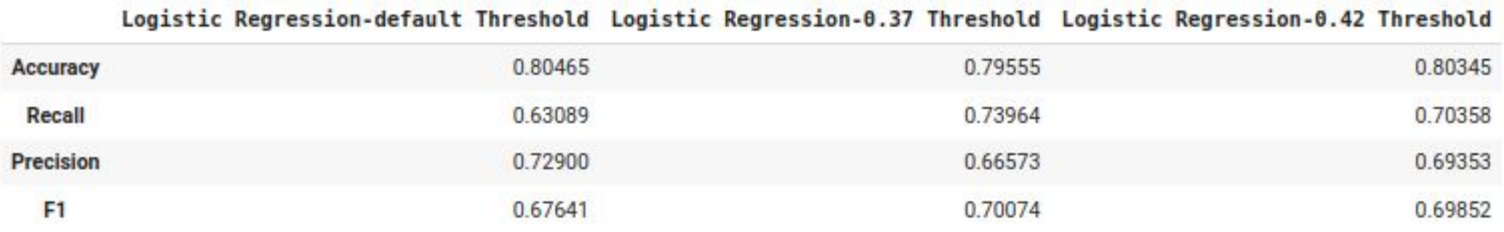
- **Logistic Regression:** We generated various models under various conditions, and their training and test performance can be seen in the tables below:

Training



	Logistic Regression-default Threshold	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold
Accuracy	0.80545	0.79265	0.80132
Recall	0.63267	0.73622	0.69939
Precision	0.73907	0.66808	0.69797
F1	0.68174	0.70049	0.69868

Test



	Logistic Regression-default Threshold	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold
Accuracy	0.80465	0.79555	0.80345
Recall	0.63089	0.73964	0.70358
Precision	0.72900	0.66573	0.69353
F1	0.67641	0.70074	0.69852

Model Performance Summary

- **Logistic Regression:**
    - In our problem, we were attempting to maximize the generated model's F1-score. Among the various models we generated, the model generated by **logistic regression with a threshold of 0.37** is our best model, with an **F1-score of 0.7** for both training and test sets.
    - This result also demonstrates that our model is **well generalized** because it performs the same on both the training and test sets.

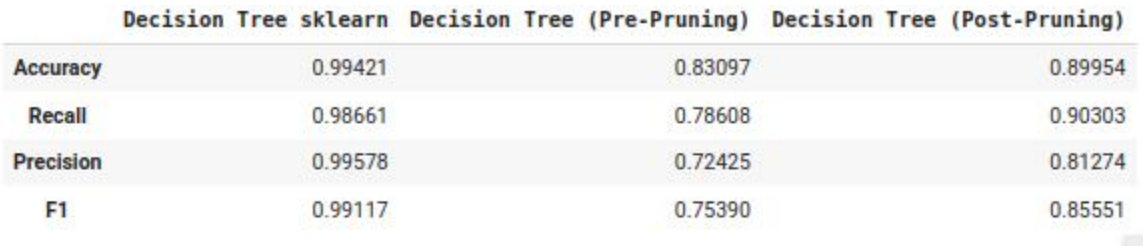
Model Performance Summary

- **Decision Tree:**
    - Using the decision tree's default configurations, we identified 18 features that are more important in making predictions.
    - We pruned our trees using **pre-pruning** and **post-pruning** techniques. The following models have been identified as the best:
        - **Pre-pruning:** our best model had the following parameters after using hyperparameter tuning and Cross Validation techniques:
            - Max_depth = 6
            - Max_leaf_nodes = 50
            - Min_samples_split = 10
            - #Used features: 10
        - **Post_pruning:** We determined the best **ccp_alpha** parameter by using the cost complexity to prune the tree as ‘0.00012267633155167043’
            - #Used features : 14

Model Performance Summary

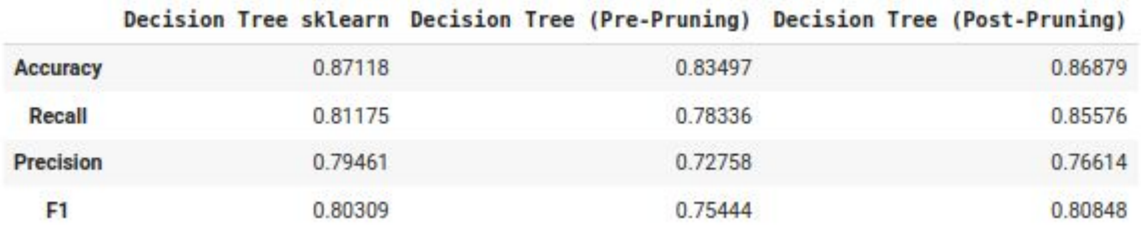
- **Decision Tree:** We generated different models using the decision tree's default parameter, pre-pruning the tree, and post-pruning the tree, and their training and test performance are shown in the tables below:

Training



	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.99421	0.83097	0.89954
Recall	0.98661	0.78608	0.90303
Precision	0.99578	0.72425	0.81274
F1	0.99117	0.75390	0.85551

Test



	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.87118	0.83497	0.86879
Recall	0.81175	0.78336	0.85576
Precision	0.79461	0.72758	0.76614
F1	0.80309	0.75444	0.80848

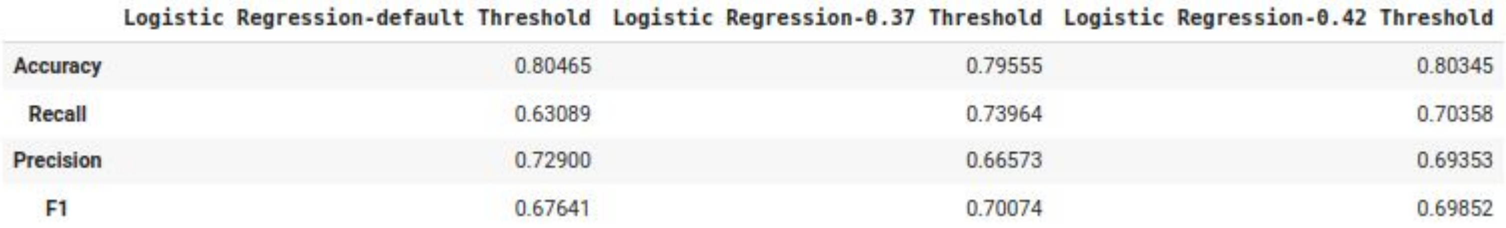
Model Performance Summary

- **Decision Tree:**
  - We **overfit** the training data by using the default decision tree parameters, as we have a high performance on the training set but a lower performance on the test set.
  - Pruning techniques appear to help the model generalize without overfitting the training data (which was the goal of pruning the tree).
  - We would choose the **best model** as the model generated using the post-pruning technique because it has the **highest F1-score** of all and is **well generalized** as it performs similarly on the training and test sets.

Model Performance Summary

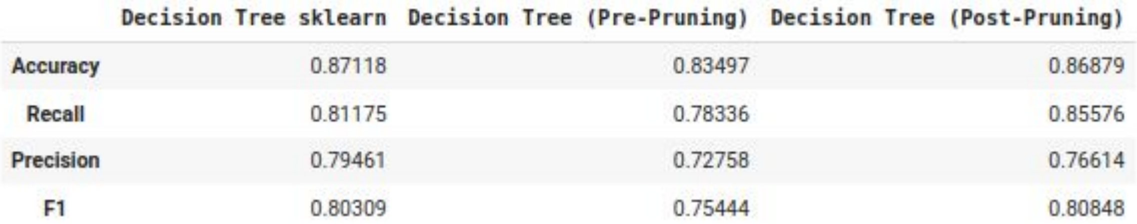
- **Logistic Regression vs Decision tree:** Following is a comparison of the models generated by logistic regression and decision tree:

Logistic Regression



	Logistic Regression-default Threshold	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold
Accuracy	0.80465	0.79555	0.80345
Recall	0.63089	0.73964	0.70358
Precision	0.72900	0.66573	0.69353
F1	0.67641	0.70074	0.69852

Decision Tree



	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.87118	0.83497	0.86879
Recall	0.81175	0.78336	0.85576
Precision	0.79461	0.72758	0.76614
F1	0.80309	0.75444	0.80848

- The use of a decision tree results in a higher F1-score. So, among all of the models produced, we would select the model produced by **post-pruning the tree.**

APPENDIX

Data Background and Contents

- INN Hotels Group operates a hotel chain in Portugal; they are experiencing high booking cancellation rates and have sought data-driven solutions. We must create a model to assist them based on the data provided, which includes some attributes of their previous customers' booking details.
- The data set contains **18 booking-related attributes**, including but not limited to the following:
  - **no_of_adults:** Number of adults
  - **no_of_children**: Number of Children
  - **no_of_weekend_nights**: Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
  - **no_of_week_nights**: Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
  - **type_of_meal_plan:** Type of meal plan booked by the customer
  - **room_type_reserved**: Type of room reserved by the customer.
  - **lead_time:** Number of days between the date of booking and the arrival date
  - **market_segment_type**: Market segment designation.
  - **repeated_guest**: Is the customer a repeated guest?
  - ...

Model Building - Logistic Regression

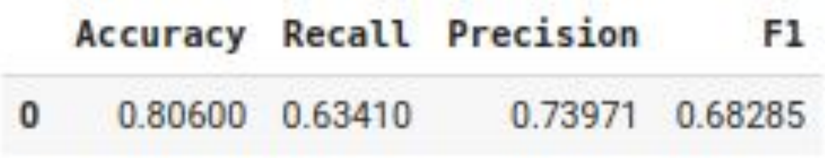
- Initially, Logistic Regression employed 27 predictor variables. We know that logistic regression requires the independent variables to have little or no multicollinearity. We used the variance inflation factor (VIF) to assess the multicollinearity of the variables and eliminated those with p-values greater than 0.05.
  - We eventually settled on **21 variables**, which aided the **model's convergence**.

- Checking the odds of the variables, we concluded that the most important features for distinguishing the target variables for each observation were **'arrival_year,' 'no_of_previous_cancellations,' and 'type_of_meal_plan_Not Selected.'**

Model Building - Logistic Regression

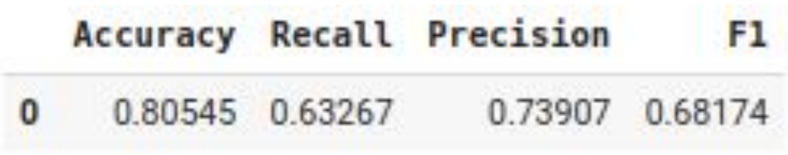
- Here is a comparison of model performance before and after checking for multicollinearity:

Before



	Accuracy	Recall	Precision	F1
0	0.80600	0.63410	0.73971	0.68285

After

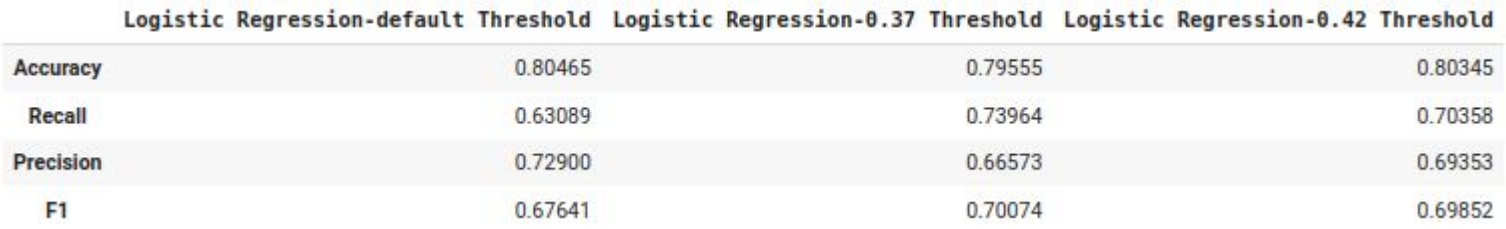


	Accuracy	Recall	Precision	F1
0	0.80545	0.63267	0.73907	0.68174

- We see no improvement in the model's performance over the training set, **but in the later case, the model converged while it did not in the other.**

Model Performance Evaluation and Improvement - Logistic Regression

- The table below shows the model performance over the test set when the classification threshold is varied:



	Logistic Regression-default Threshold	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold
Accuracy	0.80465	0.79555	0.80345
Recall	0.63089	0.73964	0.70358
Precision	0.72900	0.66573	0.69353
F1	0.67641	0.70074	0.69852

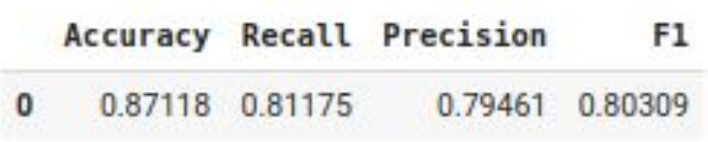
- We can conclude that adjusting the classification threshold using the AUC-ROC technique **did help us improve the model's overall performance**. The classification threshold of **0.37** was determined to be the best.

Model Building - Decision Tree

- To train a Decision Tree model, we first encoded the categorical variables in the dataset. The model was then fitted to the training data.

- Decision Trees would select the one with the **highest importance** at each level based on the importance of the features in each split (as measured by the impurity of the feature).

- We obtained the following performance of the generated model over the test set using the default parameters of the decision tree algorithm:



	Accuracy	Recall	Precision	F1
0	0.87118	0.81175	0.79461	0.80309

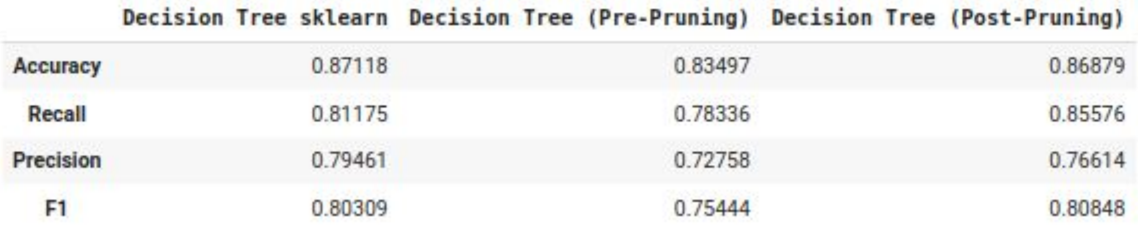
- The generated **model performs well**, especially when compared to the model generated using Logistic Regression.

Model Performance Evaluation and Improvement - Decision Tree

- In order to prune our decision tree model, we used **pre- and post-pruning** techniques. **Pre-pruning** will attempt to set the model's hyper-parameters prior to generating the model, while **post-pruning** will attempt to prune the tree after the model has been generated.

- We identified 10 variables that were important enough to be chosen for pre-pruning and 14 for post-pruning.

- The model's performance across the test set in the three versions can be summarized as follows:



	Decision Tree sklearn	Decision Tree (Pre-Pruning)	Decision Tree (Post-Pruning)
Accuracy	0.87118	0.83497	0.86879
Recall	0.81175	0.78336	0.85576
Precision	0.79461	0.72758	0.76614
F1	0.80309	0.75444	0.80848

- The model generated using the **post-pruning** technique has roughly the same performance as the model generated using the **default parameters**, but the **latter model is preferred** because it is **more generalized** when we compare training performance versus test performance.

Slide Header

- Please add any other pointers (if needed)

Great Learning

Happy Learning !