INN Hotels Project

Supervised Learning-Classification

2/17/2023

Contents / Agenda

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

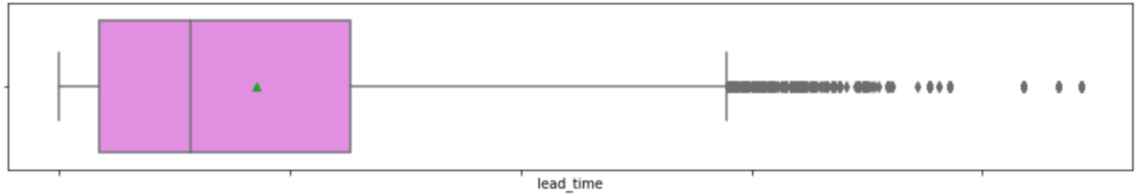
Business Problem Overview and Solution Approach

- INN Hotels, a hotels group in Portugal, is experiencing a surge in booking cancellations and want to utilize their data to create a model which can predict which bookings are likely to be cancelled. To do this, we must analyze the data and decide which factors have a higher influence on booking cancellations and use what we learn to build a predictive model to help determine which bookings are more likely to be canceled in advance as well as help create cancellation and refund policies for the hotels.

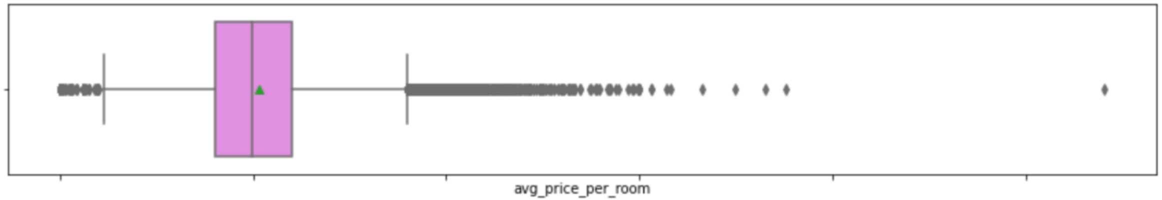
- Bookings being cancelled has many impacts on a hotel such as losing revenue when the room cannot be booked after a last minute cancellation and similarly having to lower prices last minute in order to book a room. These are reasons why INN Hotels is seeking help to benefit their company's well-being,

EDA Results

- The data has a shape of 36275 by 18 with no missing values



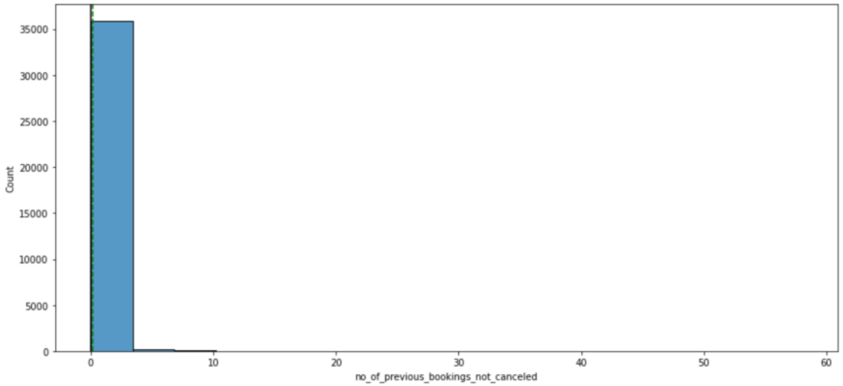
- Above is the histogram for the days between booking and arrival. As you can see, they are mainly small values but have a lot of outliers on the higher end, meaning most bookings are made last-minute.



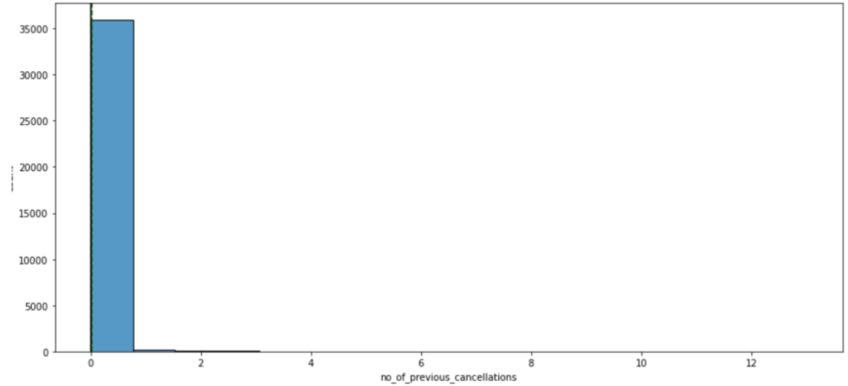
- This is the histogram for the average price per room, which is fairly varied.

EDA Results

- The boxplot to the right shows the number of previous bookings that were not cancelled by the same customer. As we can see, they are usually zero meaning they are either a first time customer or have not cancelled any bookings,

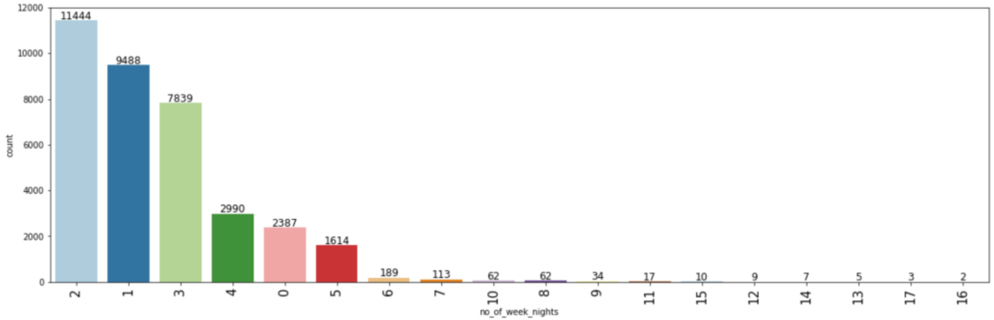


- The boxplot to the right shows the number of previous bookings that were cancelled by the same customer. Again they are mostly at zero.

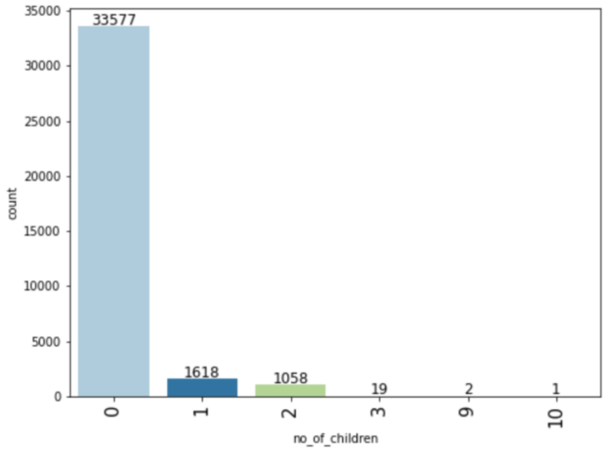


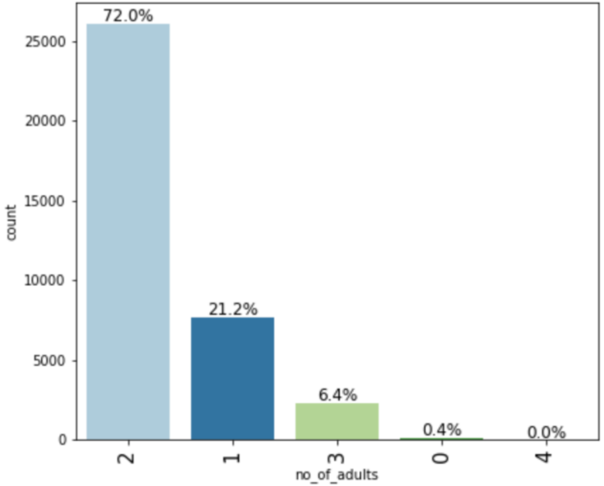
EDA Results

- The barplots to the right shows the number of children and adults per room. This gives us a good idea of what kinds of cutomers are staying at the hotel for example families with children, couples, or single adults.



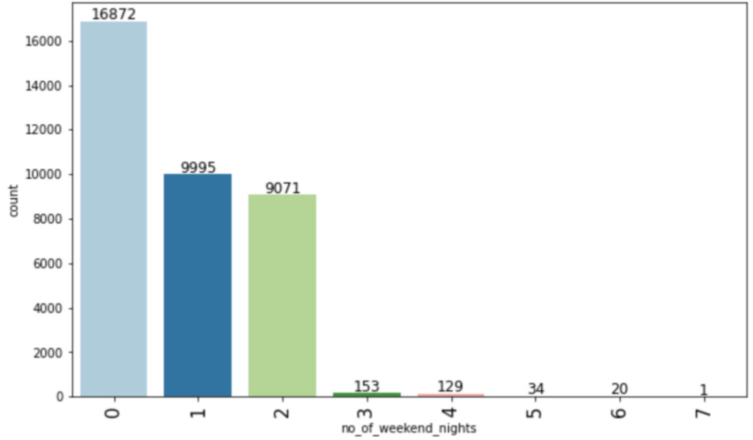
- The barplot above shows the number of weeknights per stay.



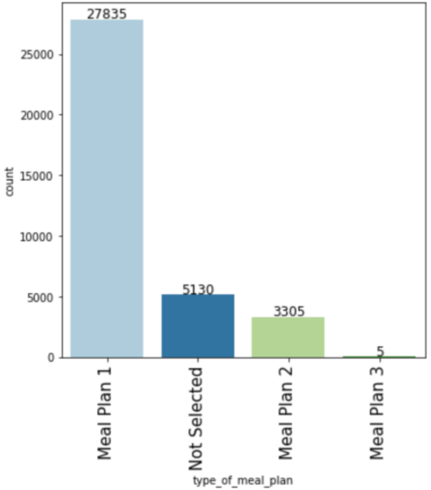


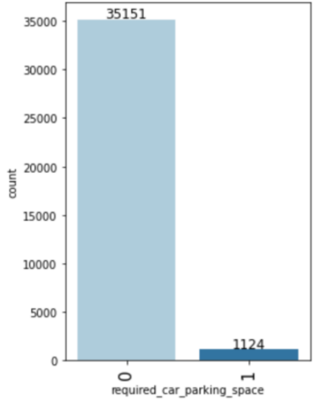
EDA Results

- The barplot to the right shows the number of weekend nights per stay, showing that the highest occuring number is 0 weekend nights possibly meaning more stays are mader on weekdays.

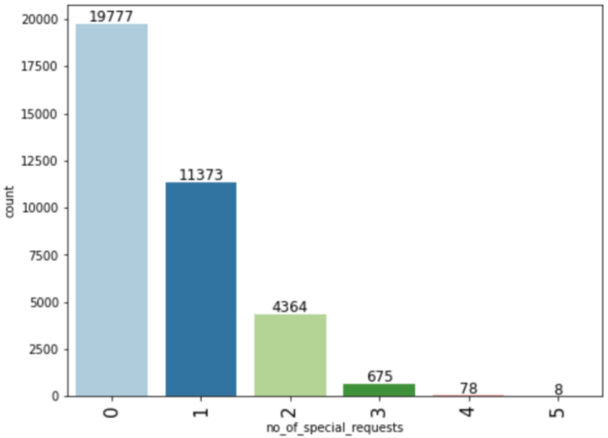


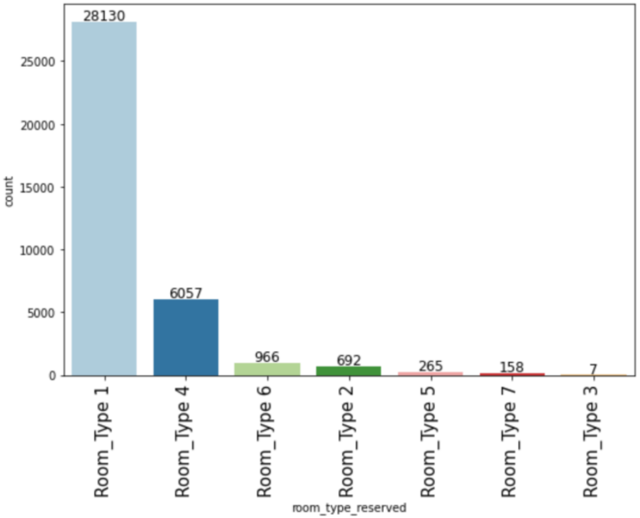
- The barplots to the right show the types of meal plan and number of required car parking spaces.

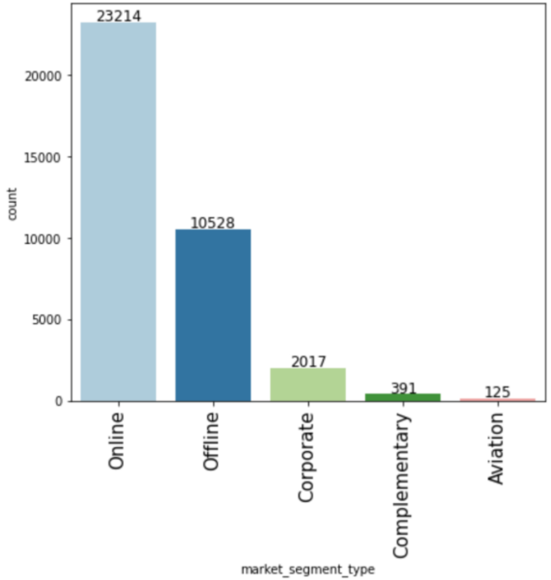




EDA Results



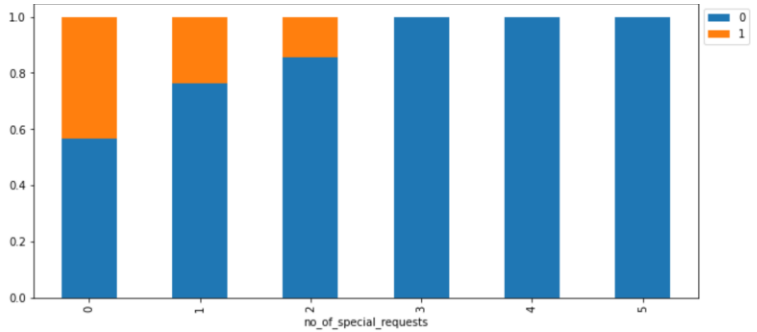




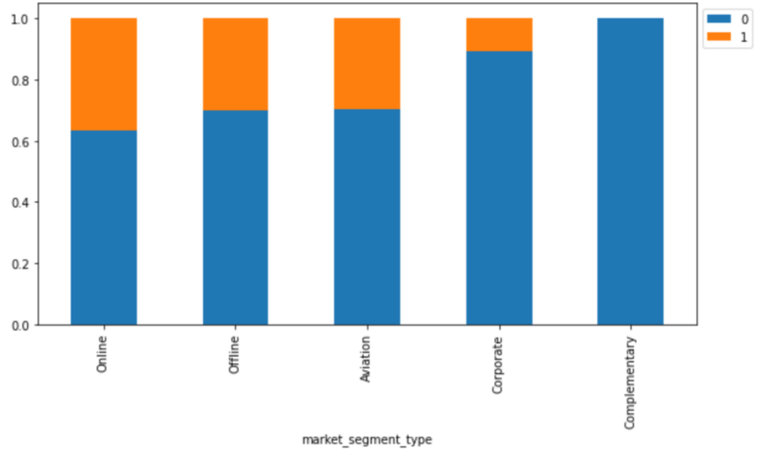
- The barplots above show the number of special requests, room types, and what market they are from. From these, we learn most customers do not make special requests, reserve room type 1, and make the booking online.

EDA Results

- The stacked barplot to the right shows the number of special requests made by a guest and the orange shows the bookings which were cancelled. This shows that the more requests made, the less likely it is for a booking to be cancelled.

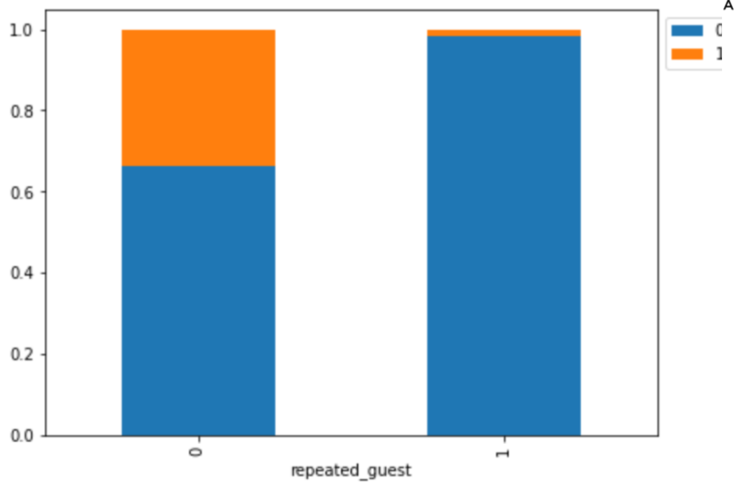


- The stacked barplot to the right shows the market segment type and also which bookings were cancelled in orange. It shows that free bookings are not likely to be cancelled.

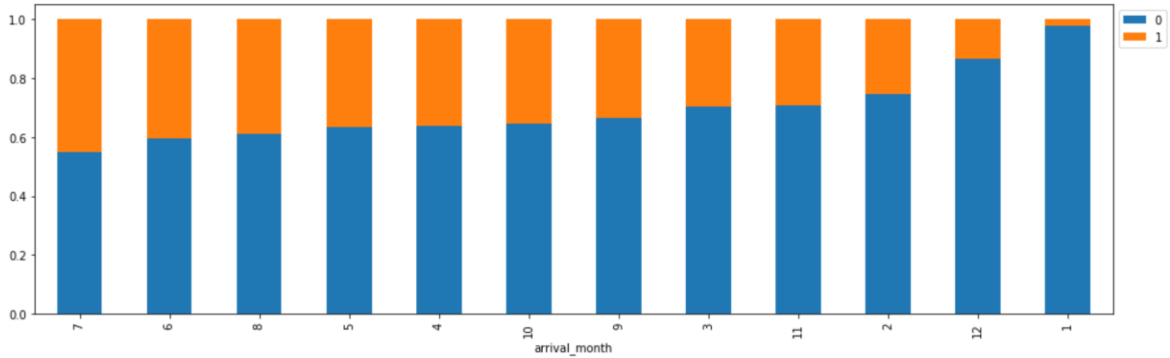


EDA Results

- The stacked barplot to the right shows that repeating guests are less likely to cancel a booking.

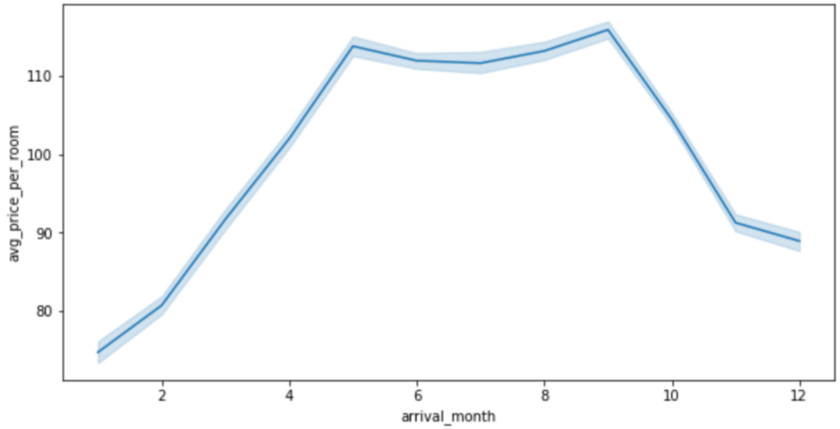


- The stacked barplot below shows that bookings for the beginning or end of the year are less likely to be cancelled.

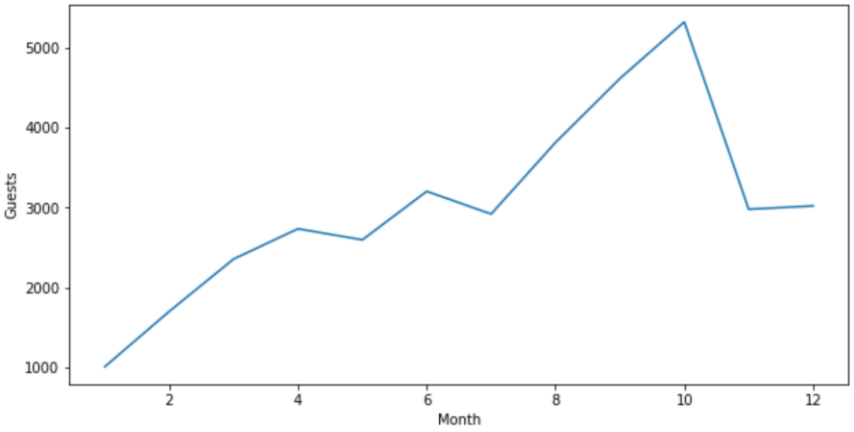


EDA Results

- The lineplot to the right shows that rooms are usually more expensive the middle of the year (summer).

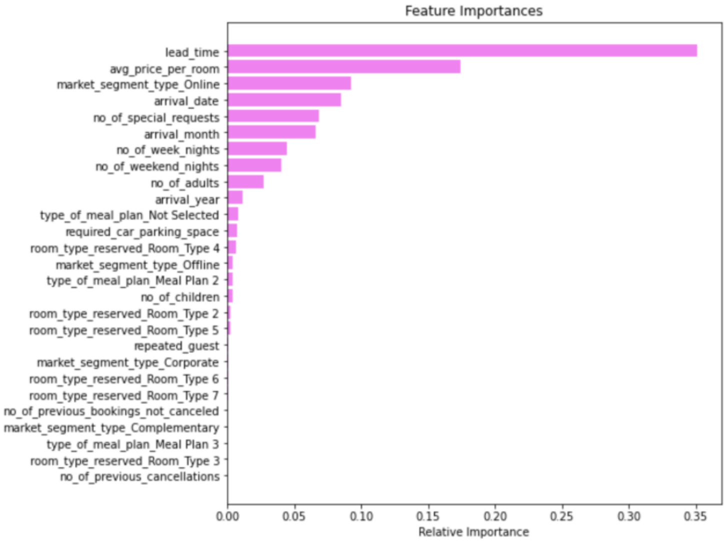


- The lineplot to the right shows that during the year there is a fairly steady increase in the number of guests staying.



Data Preprocessing

- There are no duplicated or missing values!
- The outliers also do not require treating
- Below are the features of most importance for our particular model:



Model Building - Logistic Regression

- Data preparation for modeling: we built a logistic regression model

	coef	std err	z	P>|z|	[0.025	0.975]
const	-922.8266	120.832	-7.637	0.000	-1159.653	-686.000
no_of_adults	0.1137	0.038	3.019	0.003	0.040	0.188
no_of_children	0.1580	0.062	2.544	0.011	0.036	0.280
no_of_weekend_nights	0.1067	0.020	5.395	0.000	0.068	0.145
no_of_week_nights	0.0397	0.012	3.235	0.001	0.016	0.064
required_car_parking_space	-1.5943	0.138	-11.565	0.000	-1.865	-1.324
lead_time	0.0157	0.000	58.863	0.000	0.015	0.016
arrival_year	0.4561	0.060	7.617	0.000	0.339	0.573
arrival_month	-0.0417	0.006	-6.441	0.000	-0.054	-0.029
arrival_date	0.0005	0.002	0.259	0.796	-0.003	0.004
repeated_guest	-2.3472	0.617	-3.806	0.000	-3.556	-1.139
no_of_previous_cancellations	0.2664	0.086	3.108	0.002	0.098	0.434
no_of_previous_bookings_not_canceled	-0.1727	0.153	-1.131	0.258	-0.472	0.127
avg_price_per_room	0.0188	0.001	25.396	0.000	0.017	0.020
no_of_special_requests	-1.4689	0.030	-48.782	0.000	-1.528	-1.410
type_of_meal_plan_Meal Plan 2	0.1756	0.067	2.636	0.008	0.045	0.306
type_of_meal_plan_Meal Plan 3	17.3584	3987.873	0.004	0.997	-7798.729	7833.445
type_of_meal_plan_Not Selected	0.2784	0.053	5.247	0.000	0.174	0.382
room_type_reserved_Room_Type 2	-0.3605	0.131	-2.748	0.006	-0.618	-0.103
room_type_reserved_Room_Type 3	-0.0012	1.310	-0.001	0.999	-2.568	2.566
room_type_reserved_Room_Type 4	-0.2823	0.053	-5.304	0.000	-0.387	-0.178
room_type_reserved_Room_Type 5	-0.7189	0.209	-3.438	0.001	-1.129	-0.309
room_type_reserved_Room_Type 6	-0.9501	0.151	-6.274	0.000	-1.247	-0.653
room_type_reserved_Room_Type 7	-1.4003	0.294	-4.770	0.000	-1.976	-0.825
market_segment_type_Complementary	-40.5976	5.65e+05	-7.19e-05	1.000	-1.11e+06	1.11e+06
market_segment_type_Corporate	-1.1924	0.266	-4.483	0.000	-1.714	-0.671
market_segment_type_Offline	-2.1946	0.255	-8.621	0.000	-2.694	-1.696
market_segment_type_Online	-0.3995	0.251	-1.590	0.112	-0.892	0.093

Logit Regression Results

Dep. Variable:	booking_status	No. Observations:	25392
Model:	Logit	Df Residuals:	25364
Method:	MLE	Df Model:	27
Date:	Fri, 17 Feb 2023	Pseudo R-squ.:	0.3292
Time:	02:13:18	Log-Likelihood:	-10794.
converged:	False	LL-Null:	-16091.
Covariance Type:	nonrobust	LLR p-value:	0.000

Training performance:

	Accuracy	Recall	Precision	F1
0	0.80600	0.63410	0.73971	0.68285

Model Performance Evaluation and Improvement - Logistic Regression

- We then checked VIFs for multicollinearity and dropped the high p-value variables and executed a new regression as well as a confusion matrix. Here are the results:

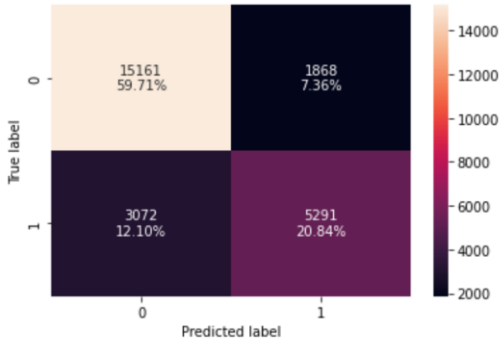
	coef	std err	z	P>|z|	[0.025	0.975]
const	-915.6391	120.471	-7.600	0.000	-1151.758	-679.520
no_of_adults	0.1088	0.037	2.914	0.004	0.036	0.182
no_of_children	0.1531	0.062	2.470	0.014	0.032	0.275
no_of_weekend_nights	0.1086	0.020	5.498	0.000	0.070	0.147
no_of_week_nights	0.0417	0.012	3.399	0.001	0.018	0.066
required_car_parking_space	-1.5947	0.138	-11.564	0.000	-1.865	-1.324
lead_time	0.0157	0.000	59.213	0.000	0.015	0.016
arrival_year	0.4523	0.060	7.576	0.000	0.335	0.569
arrival_month	-0.0425	0.006	-6.591	0.000	-0.055	-0.030
repeated_guest	-2.7367	0.557	-4.916	0.000	-3.828	-1.646
no_of_previous_cancellations	0.2288	0.077	2.983	0.003	0.078	0.379
avg_price_per_room	0.0192	0.001	26.336	0.000	0.018	0.021
no_of_special_requests	-1.4698	0.030	-48.884	0.000	-1.529	-1.411
type_of_meal_plan_Meal Plan 2	0.1642	0.067	2.469	0.014	0.034	0.295
type_of_meal_plan_Not Selected	0.2860	0.053	5.406	0.000	0.182	0.390
room_type_reserved_Room_Type 2	-0.3552	0.131	-2.709	0.007	-0.612	-0.098
room_type_reserved_Room_Type 4	-0.2828	0.053	-5.330	0.000	-0.387	-0.179
room_type_reserved_Room_Type 5	-0.7364	0.208	-3.535	0.000	-1.145	-0.328
room_type_reserved_Room_Type 6	-0.9682	0.151	-6.403	0.000	-1.265	-0.672
room_type_reserved_Room_Type 7	-1.4343	0.293	-4.892	0.000	-2.009	-0.860
market_segment_type_Corporate	-0.7913	0.103	-7.692	0.000	-0.993	-0.590
market_segment_type_Offline	-1.7854	0.052	-34.363	0.000	-1.887	-1.684

Logit Regression Results

Dep. Variable:	booking_status	No. Observations:	25392
Model:	Logit	Df Residuals:	25370
Method:	MLE	Df Model:	21
Date:	Fri, 17 Feb 2023	Pseudo R-squ.:	0.3282
Time:	02:16:22	Log-Likelihood:	-10810.
converged:	True	LL-Null:	-16091.
Covariance Type:	nonrobust	LLR p-value:	0.000

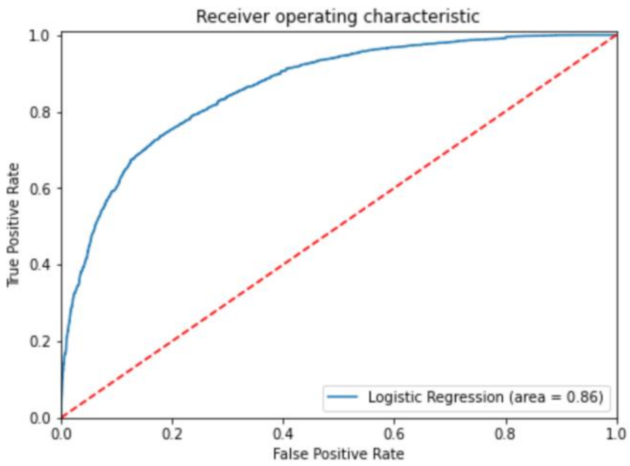
Training performance:

	Accuracy	Recall	Precision	F1
0	0.80545	0.63267	0.73907	0.68174

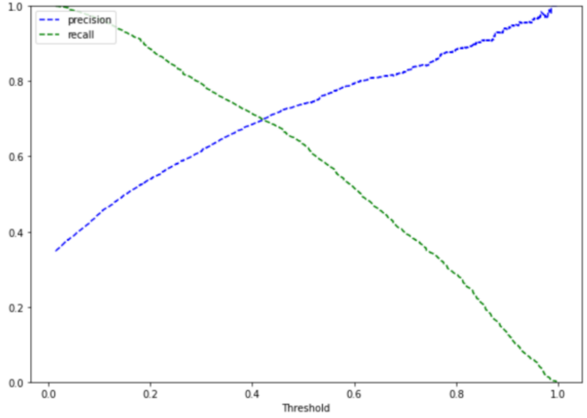


Model Performance Evaluation and Improv n

- The AUC-ROC curve of our model is shown to the right.
- We wanted to improve the recall score by changing the model threshold with this curve. The optimal threshold cutoff is where tpr is high and fpr is low

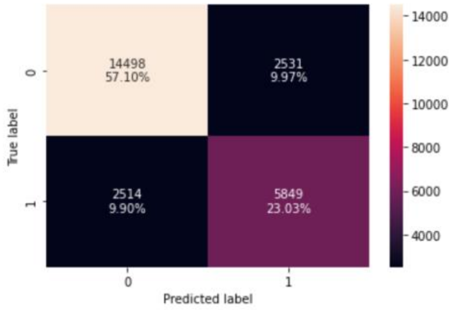


- Using the Precision-Recall curve to the right, we found th optimal threshold for our model to be 0.42



Model Performance Evaluation and Improvement - Logistic Regression

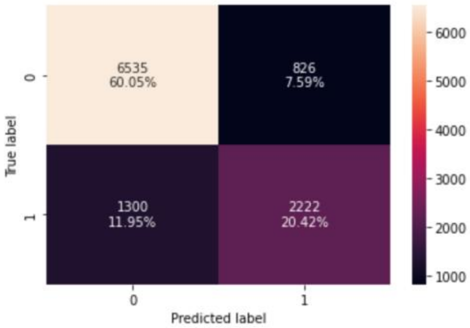
- With the new threshold, we tested both our test and train data. Here are the results for both:




```
[ ] log_reg_model_train_perf_threshold_curve = model_performance_classification_statsmodel
        lg1, X_train1, y_train, threshold=optimal_threshold_curve
    )
    print("Training performance:")
    log_reg_model_train_perf_threshold_curve
```

Training performance:

	Accuracy	Recall	Precision	F1
0	0.80132	0.69939	0.69797	0.69868




```
[ ] log_reg_model_test_perf = model_performance_classification_statsmodels( lg1, X_t

    print("Test performance:")
    log_reg_model_test_perf
```

Test performance:

	Accuracy	Recall	Precision	F1
0	0.80465	0.63089	0.72900	0.67641

Model Performance Evaluation and Improvement - Logistic Regression

- We then compared the performance of testing and training data on both of the thresholds:

Training performance comparison:

	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 performance comparison:

	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

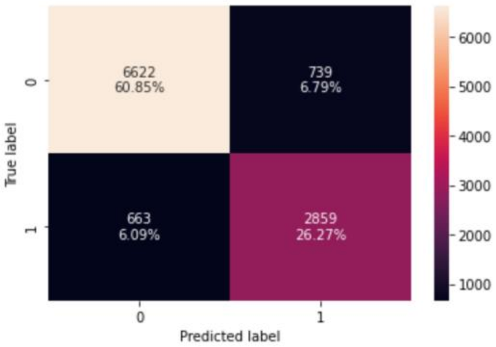
- We concluded that the threshold of 0.42 is preferred.

Model Building - Decision Tree

- First, we split the data:

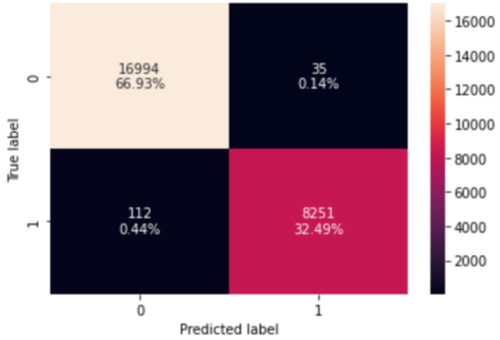
```
Shape of Training set :  (25392, 27)
Shape of test set :  (10883, 27)
Percentage of classes in training set:
0    0.67064
1    0.32936
Name: booking_status, dtype: float64
Percentage of classes in test set:
0    0.67638
1    0.32362
Name: booking_status, dtype: float64
```

- Then, we tested the model on these sets of data:




```
decision_tree_perf_test = model_performance_classificati
decision_tree_perf_test
```

	Accuracy	Recall	Precision	F1
0	0.87118	0.81175	0.79461	0.80309




```
decision_tree_perf_train = model_performance_classific
    model, X_train, y_train
)
decision_tree_perf_train
```

	Accuracy	Recall	Precision	F1
0	0.99421	0.98661	0.99578	0.99117

Model Building - Decision Tree

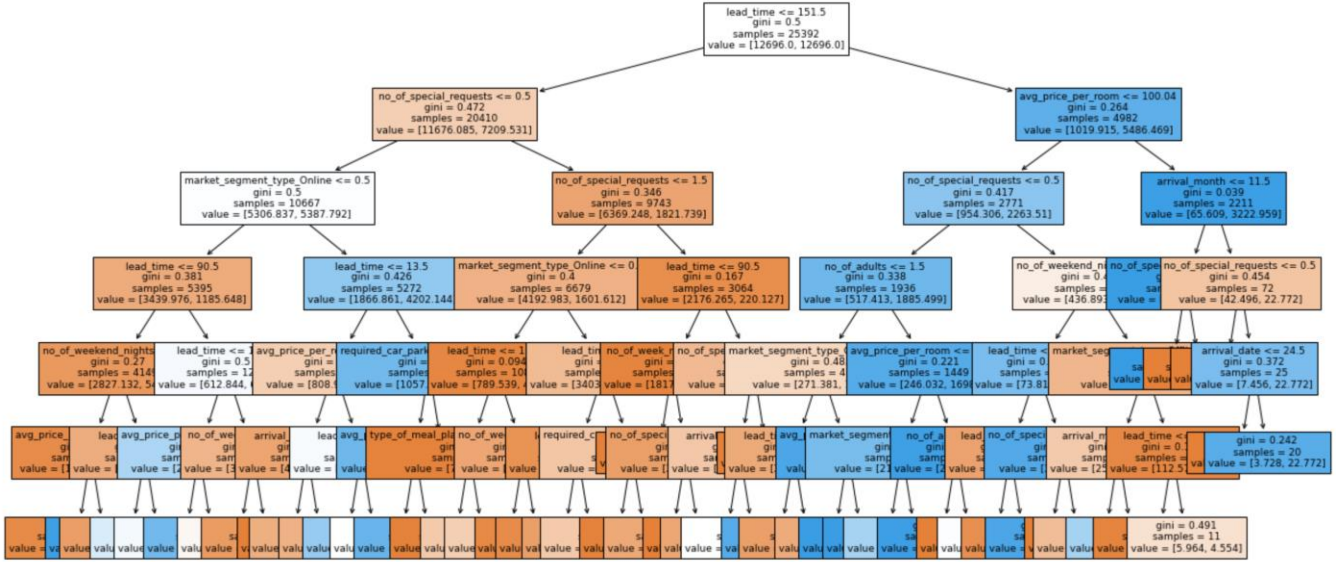
- Pre-pruning: below are the parameters set for the decision tree:

```
# Choose the type of classifier.
estimator = DecisionTreeClassifier(random_state=1, class_weight="balanced")

# Grid of parameters to choose from
parameters = {
    "max_depth": np.arange(2, 7, 2),
    "max_leaf_nodes": [50, 75, 150, 250],
    "min_samples_split": [10, 30, 50, 70],
}

# Type of scoring used to compare parameter combinations
acc_scorer = make_scorer(f1_score)

# Run the grid search
grid_obj = GridSearchCV(estimator, parameters, scoring=acc_scorer, cv=5)
grid_obj = grid_obj.fit(X_train, y_train)

# Set the clf to the best combination of parameters
estimator = grid_obj.best_estimator_

# Fit the best algorithm to the data.
estimator.fit(X_train, y_train)
```

```
DecisionTreeClassifier(class_weight='balanced', max_depth=6, max_leaf_nodes=50,
                       min_samples_split=10, random_state=1)
```

Model Building - Decision Tree

- Next, we began to visualize the tree:



Model Performance Evaluation and Improvement - Decision Tree

- We already have the important features noted in data preprocessing, so we move on to cost-complexity pruning. These are our results:

```
[ ] clf = DecisionTreeClassifier(random_state=1, class_weight="balanced")
    path = clf.cost_complexity_pruning_path(X_train, y_train)
    ccp_alphas, impurities = abs(path.ccp_alphas), path.impurities
```

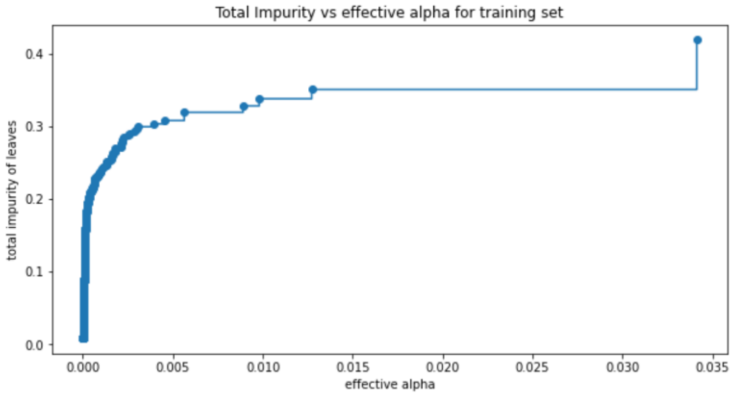
```
pd.DataFrame(path)
```

	ccp_alphas	impurities
0	0.00000	0.00838
1	0.00000	0.00838
2	0.00000	0.00838
3	0.00000	0.00838
4	0.00000	0.00838
...	...	...
1839	0.00890	0.32806
1840	0.00980	0.33786
1841	0.01272	0.35058
1842	0.03412	0.41882
1843	0.08118	0.50000

1844 rows × 2 columns

Model Performance Evaluation and Improvement - Decision Tree

- Next, we train a decision tree using effective alphas
- The last value in ccp_alphas is the alpha value that prunes the whole tree, leaving the tree with one node.



Model Performance Summary

- We want to pick the best models to predict if a booking will be cancelled.
- We want to avoid false negatives and false positives as they are harmful to our predictive models, so we want the highest F1 score in our model for both the logistic regression and decision tree.
- The decision tree post-pruning is the model with the highest F1 (0.80858) and accuracy of 0.86888, recall of 0.85576, and precision of 0.76634. This is our best model.
- As we found, our top features for prediction were lead time, market segment type, avg. price per room, no. of special requests, and arrival month.

Executive Summary

- Insights:
    - Bookings with a lead time of 120 days or more are most likely to be cancelled
    - Online bookings are most likely to be cancelled & complimentary ones are least likely
    - The more special requests, the less likely the booking is to be cancelled
    - Summer bookings are more likely to be cancelled, Winter ones are less likely
    - Bookings with 3 or more guests are more likely to be cancelled

Executive Summary

- Business Recommendations:
    - Converting more rooms to room type 1 would promote more bookings, as this is a popular choice.
    - Removing meal plan 3, the least popular, would save business costs.
    - Giving guests rewards for more stays/less cancellations such as discounts, free parking, etc would promote business.
    - The hotels should implement a cancellation fee for within 24 hours of the booking and for no-shows as well.
    - Guests should not be allowed to make bookings too far in advance (1 year or more).

Happy Learning !