Predicting Average Daily Rate (ADR) and Measuring the Financial Health of the Hospitality Industry

Stuart Toda[1]

stu-13/Predicting_Hotel_Financial_Health (github.com)

[1]Brown University, Data Science Initiative, Providence, Rhode Island

stuart_toda@brown.edu

12/5/2022

1. Introduction

The global hotel industry, worth approximately $1.5 trillion, is a mercurial environment stimulated by human instinct and curiosity to explore every corner of the globe [2]. In recent years, the industry has benefited from technological adaptations as well as socio-economic shifts in consumer behavior. However, with the Covid global pandemic, countries closed their borders and the industry experienced the harsh repercussions of their travel-dependent business model. As a result, it became incredibly important for hotels to predict their financial health and identify factors that contribute to their bottom line.

This project attempts to build a robust regression tool to predict the profitability of hotels through one of their financial metrics, ‘average daily rate’ or the revenue generated by an occupied room. This dataset originated from the data journal article, “Hotel Booking Demand Datasets”, published by members of Instituto Universitário de Lisboa and Instituto de Telecomunicações in Lisbon, Portugal. The information was extracted from the property management systems of a resort hotel in Algarve and a city hotel in Lisbon [3]. It consists of 32 features associated with hotel reservations.

While this project primarily focuses on the revenue generated by rooms, past works are more concerned with the occupancy rate of hotels through hotel cancellations. In one of these studies, the author, Anurag Lahon, adopts 4 classifiers: Logistic Regression, Ridge Regression, Lasso Regression, and Random Forest and achieved 80.4%, 80.9%, 80.6% and 93.9% accuracies respectively [1]. These results indicate that the attributes in the dataset have high predictive power when classifying cancellations through the use of various models. One major takeaway is that feature engineering is an important step to bring out hidden features with high predictive power.

2. Exploratory Data Analysis

The following figures represent a few of the many visualizations created during the dataset's investigation.

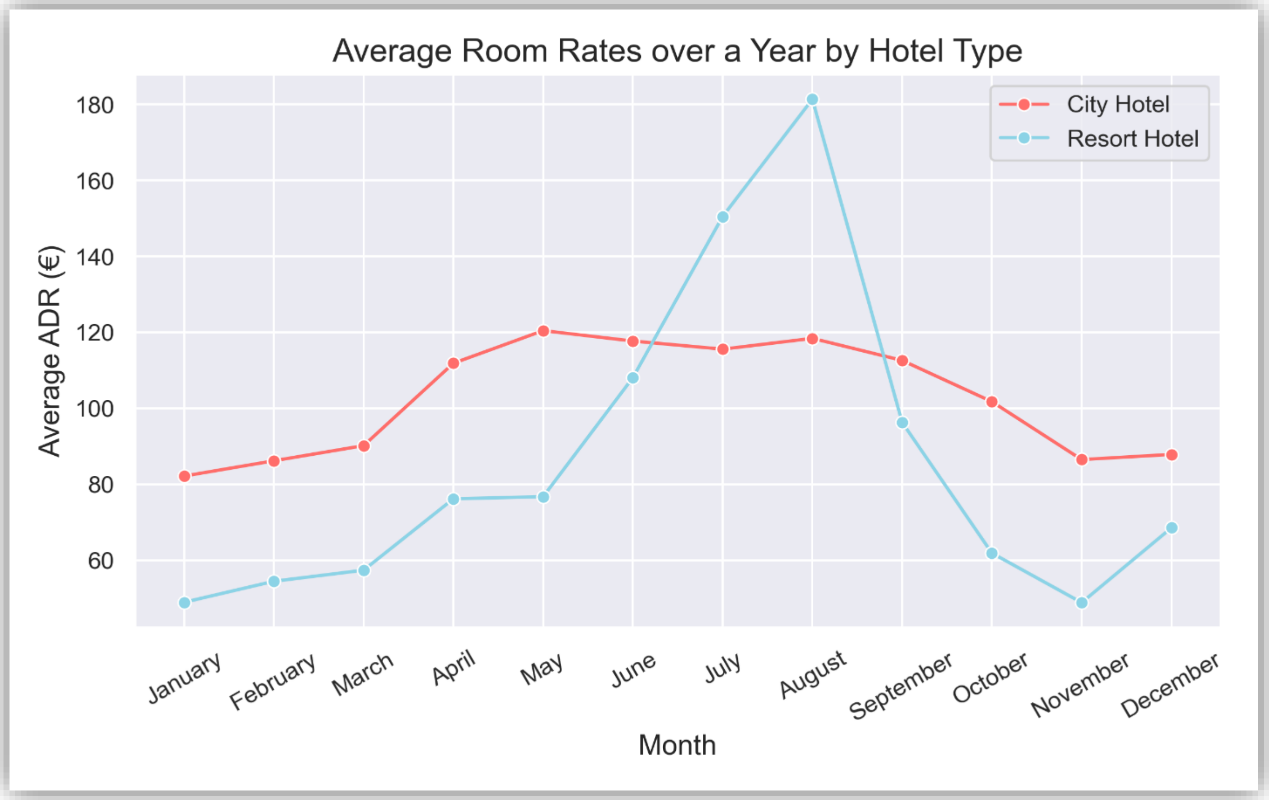


Figure 2.1 Average Room Rate over a Year by Hotel Type

Figure 1 depicts the average of the target variable, ‘adr’, over each month sequentially by hotel type. The average ‘adr’ for the city hotel is observed to be flatter and less susceptible to change over the course of a year. In comparison, the average ‘adr’ for the resort hotel sees a higher jump over the summer months, indicating the impact of seasonality.

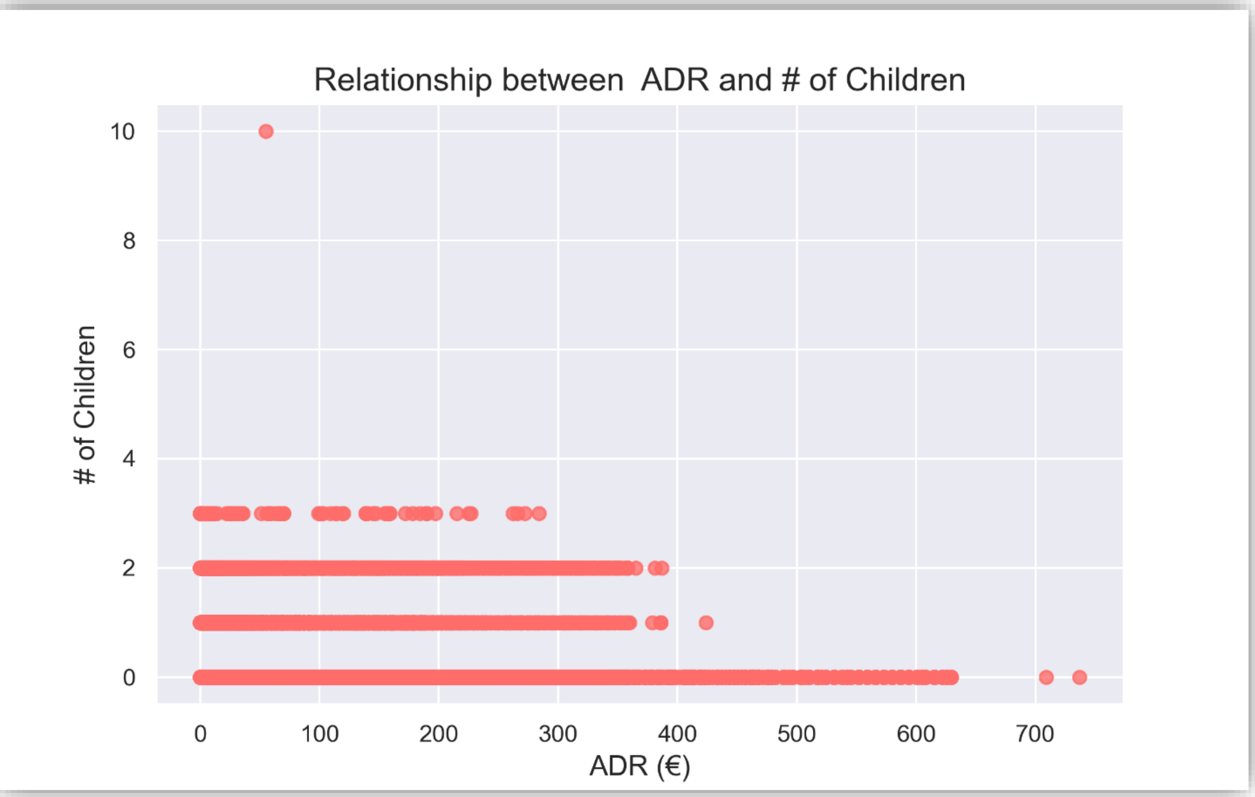


Figure 2.2 Relationship between Average Daily Rate and Number of Children on Reservation

Figure 2 shows the scatter plot of the relationship between the average daily rate and children. Typical correlation shows that more people on a reservation requires a larger room, which leads to a more expensive room rate and revenue lift; this visualization shows the opposite effect. An increase in the number of children on the reservation shows a noticeable decrease in 'adr'

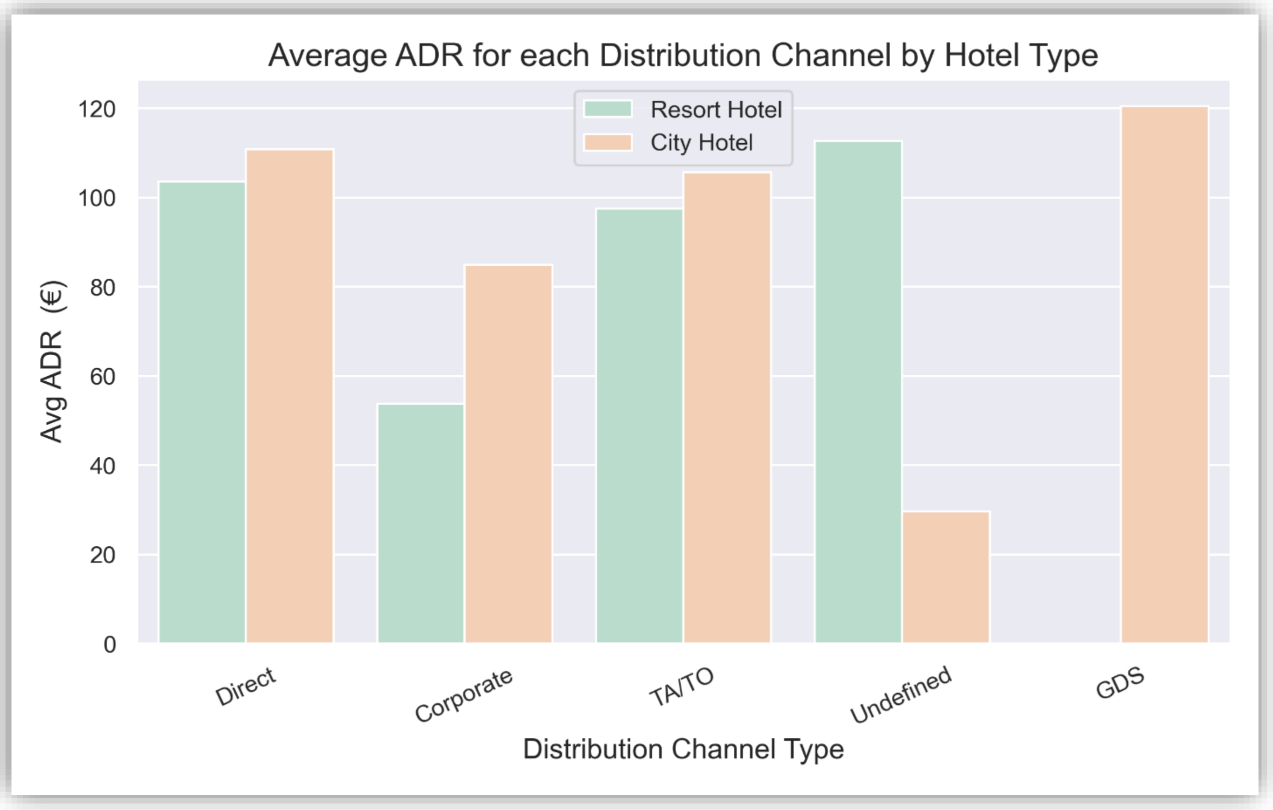


Figure 2.3 Average ADR for each Distribution Channel by Hotel Type

Figure 3 depicts a bar plot of the average 'adr' generated by distribution channel and hotel type. The distribution channels represent the origination of the reservation. In general, city hotels generate higher average 'adr's booked through each distribution channel, except for the undefined category.

3. Methods

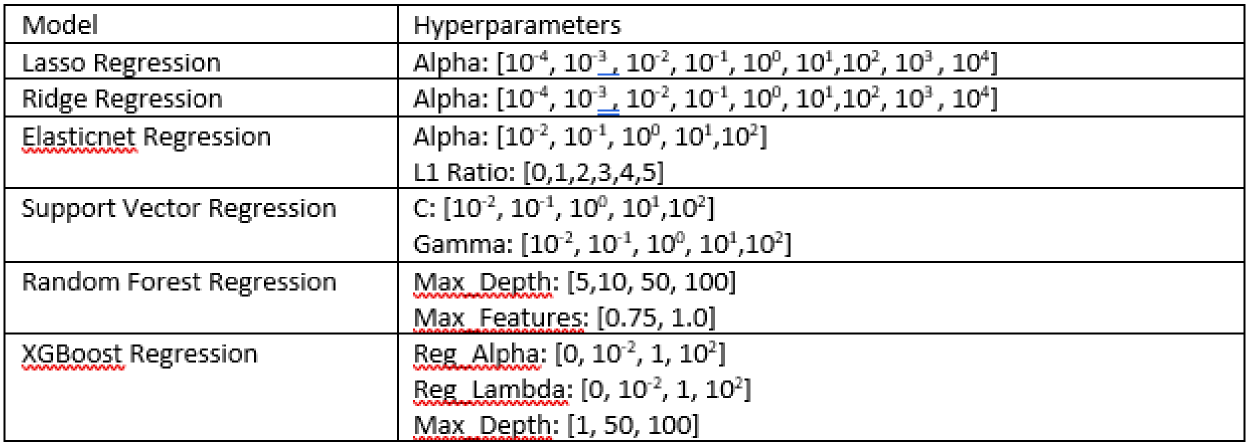
3.1 Splitting and Preprocessing

The dataset was compiled from two different hotel property management systems and was split by hotel type to maintain IID principles. Following the split, the city hotel and resort hotel datasets were duplicated and each identical pair of datasets went through different preprocessing methods based on different machine learning models.

For the models that can't handle missing values, one pair of datasets underwent a series of feature engineering, which included adding new classes for features that intentionally had missing data and cleaning up the remaining column 'Children' to fill in 4 rows with the most common value through simple imputation. For models that can handle missing values, these datasets remain untouched. Next, both pairs of datasets shared the same creation of features, such as Season (e.g. Winter, Spring, etc) and reservation month transformed with sin(x). When month is treated as an ordinal feature, the neighboring property of January and December are not taken into account; the sine transformation leverages the cyclical nature of the sine wave to capture this.

Once data creation was completed, the following encoders were used with their respective data types: OneHotEncoder for categorical features, StandardScaler for numerical and OrdinalEncoder for categorical features with an inherent ordering. The final preprocessed city and resort datasets with no missing values contains 635 and 582 columns, respectively. For the preprocessed city and resort datasets with missing values, each contain 628 and 590 features, respectively.

3.2 Machine Learning Pipeline



Model	Hyperparameters
Lasso Regression	Alpha: [10^{-4}, 10^{-3} , 10^{-2}, 10^{-1}, 10^{0}, 10^{1},10^{2}, 10^{3} , 10^{4}]
Ridge Regression	Alpha: [10^{-4}, 10^{-3} , 10^{-2}, 10^{-1}, 10^{0}, 10^{1},10^{2}, 10^{3} , 10^{4}]
Elasticnet Regression	Alpha: [10^{-2}, 10^{-1}, 10^{0}, 10^{1},10^{2}] L1 Ratio: [0,1,2,3,4,5]
Support Vector Regression	C: [10^{-2}, 10^{-1}, 10^{0}, 10^{1},10^{2}] Gamma: [10^{-2}, 10^{-1}, 10^{0}, 10^{1},10^{2}]
Random Forest Regression	Max_Depth: [5,10, 50, 100] Max_Features: [0.75, 1.0]
XGBoost Regression	Reg_Alpha: [0, 10^{-2}, 1, 10^{2}] Reg_Lambda: [0, 10^{-2}, 1, 10^{2}] Max_Depth: [1, 50, 100]

Figure: 3.2.1: Models with Hyperparameter Ranges

The machine learning models used include Random Forest Regression, XGBoost Regression, Ridge, Lasso, Elasticnet and Support Vector Regression. The hyperparameters and the ranges can be referenced in **Figure: 3.2.1**. For each model, 5 random states were tested and the average Root Mean Squared Error was calculated as the final metric while the standard deviation was identified to take into account the uncertainties in splitting and non-deterministic models.

The main pipeline, ‘MLpipe_KFold_RMSE_prep’, utilized all machine learning models except XGBoost and only employed the datasets with no missing values as its input. With the respective city and resort dataset, training and test sets was created with 80-20 splits. The training sets were then fed into a GridSearchCV to identify the best set of hyperparameters for each model. The estimator parameter was fulfilled by a pipeline containing the column transformer for preprocessing and the machine learning algorithm of interest. The cross-validation method implemented was a 4 KFold, where each validation fold utilized the Root Mean Squared Error as its scoring mechanism to identify the set of parameters that minimized the error.

A second pipeline was adopted for XGBoost because there is no streamlined way to integrate GridSearchCV with XGBoost. The main issue was the difference in the validation set that was used to train the XGBoost model and the validation set selected by GridSearchCV to find the best hyper parameters. Therefore, the second pipeline, ‘xgb_pipe’, was a deconstructed version of GridSearchCV. The best model was decided by manually storing the validation RMSE score and selecting the corresponding best model across 5 random states.

4. Results

4.1 Test Scores

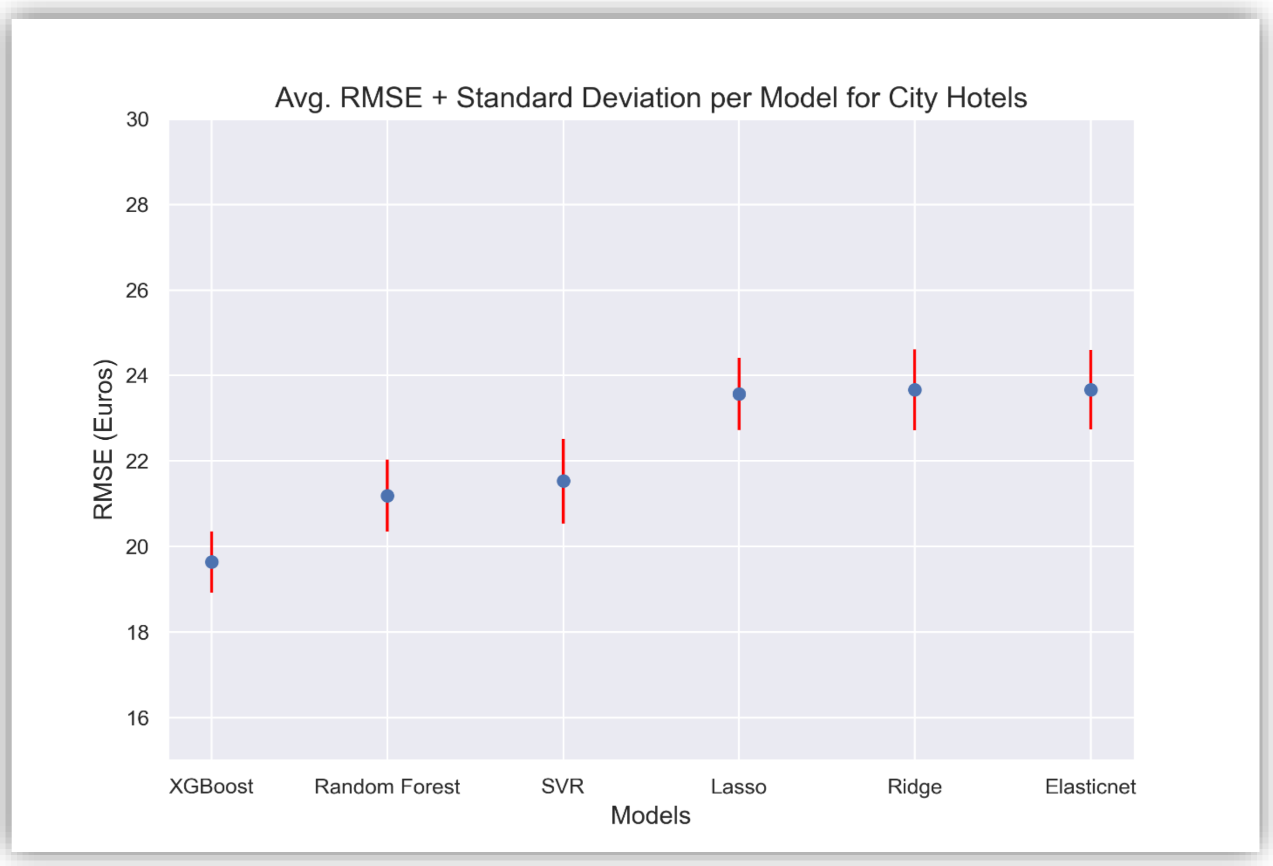
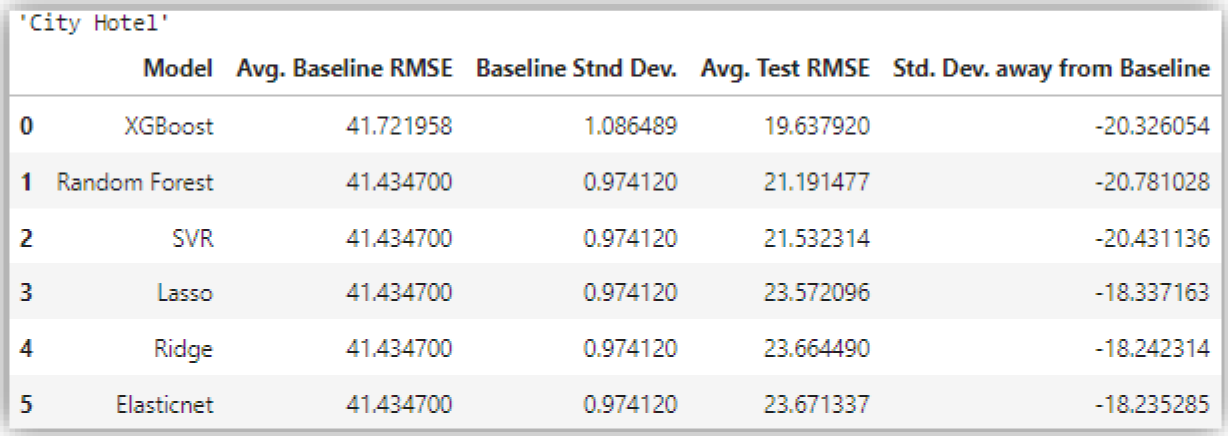


Figure 4.1.1 Average RMSE for City Hotel's ADR for each machine learning model across 5 random states



```
'City Hotel'
```

	Model	Avg. Baseline RMSE	Baseline Stnd Dev.	Avg. Test RMSE	Std. Dev. away from Baseline
0	XGBoost	41.721958	1.086489	19.637920	-20.326054
1	Random Forest	41.434700	0.974120	21.191477	-20.781028
2	SVR	41.434700	0.974120	21.532314	-20.431136
3	Lasso	41.434700	0.974120	23.572096	-18.337163
4	Ridge	41.434700	0.974120	23.664490	-18.242314
5	Elasticnet	41.434700	0.974120	23.671337	-18.235285

Figure 4.1.2 Table of City Hotel’s Average Baseline RMSE, Baseline Score Standard Deviation, Average Test RMSE and the number of standard deviations away from the Baseline RMSE for each model

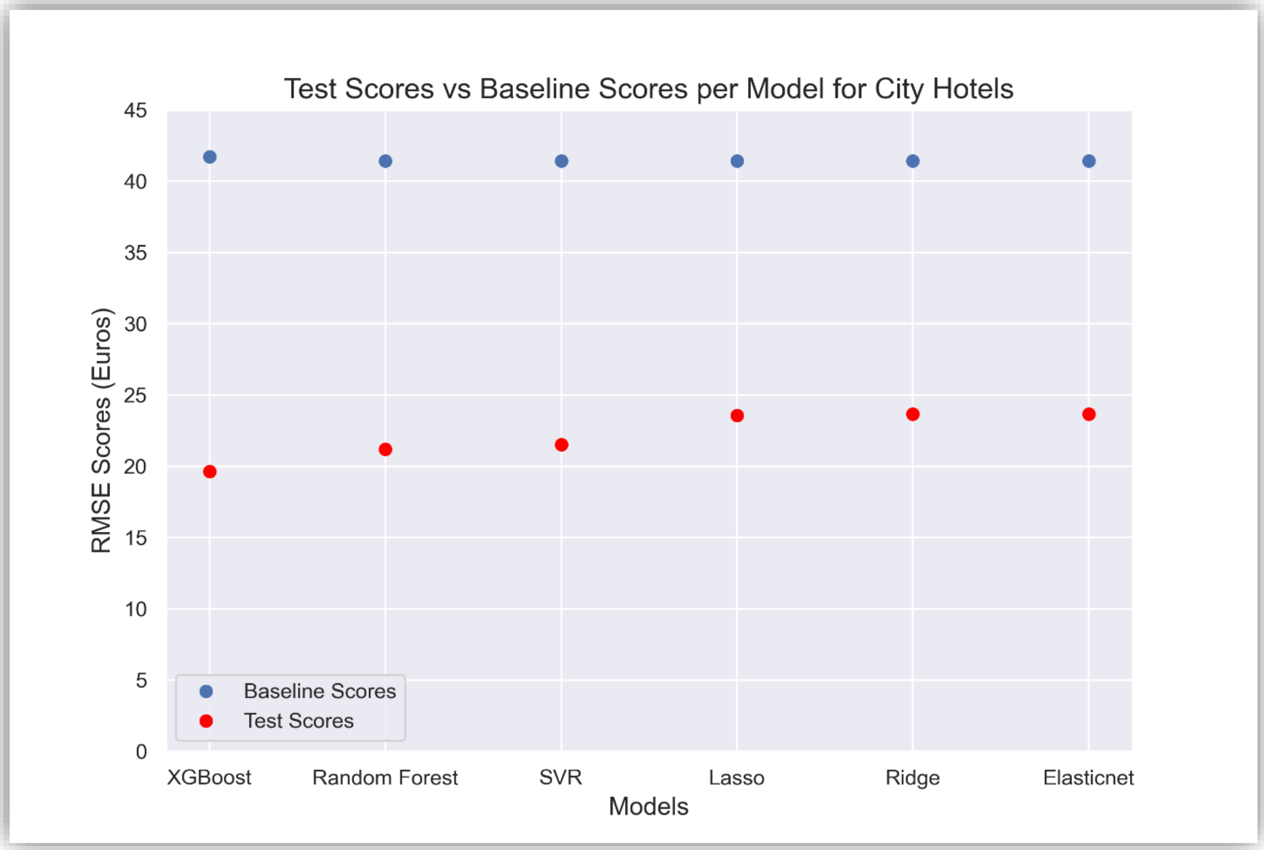


Figure 4.1.3 Average Baseline RMSE vs Average Test RMSE for City Hotels for each machine learning model across 5 different random states

Based on **Figure 4.1.1,** the best performing model for city hotel data is XGBoost Regression with an average RMSE of 19.63 euros and the smallest standard deviation of 0.71. Compared to their respective baseline scores, each model performs significantly better. For the best model, XGBoost, the average baseline RMSE is 41.72 euros with a standard deviation 1.43. Therefore, the average test RMSE is around -20 standard deviations away (**Figure 4.1.2**). A comparison of all models of their test scores and baseline scores can be found in **Figure 4.1.3**.

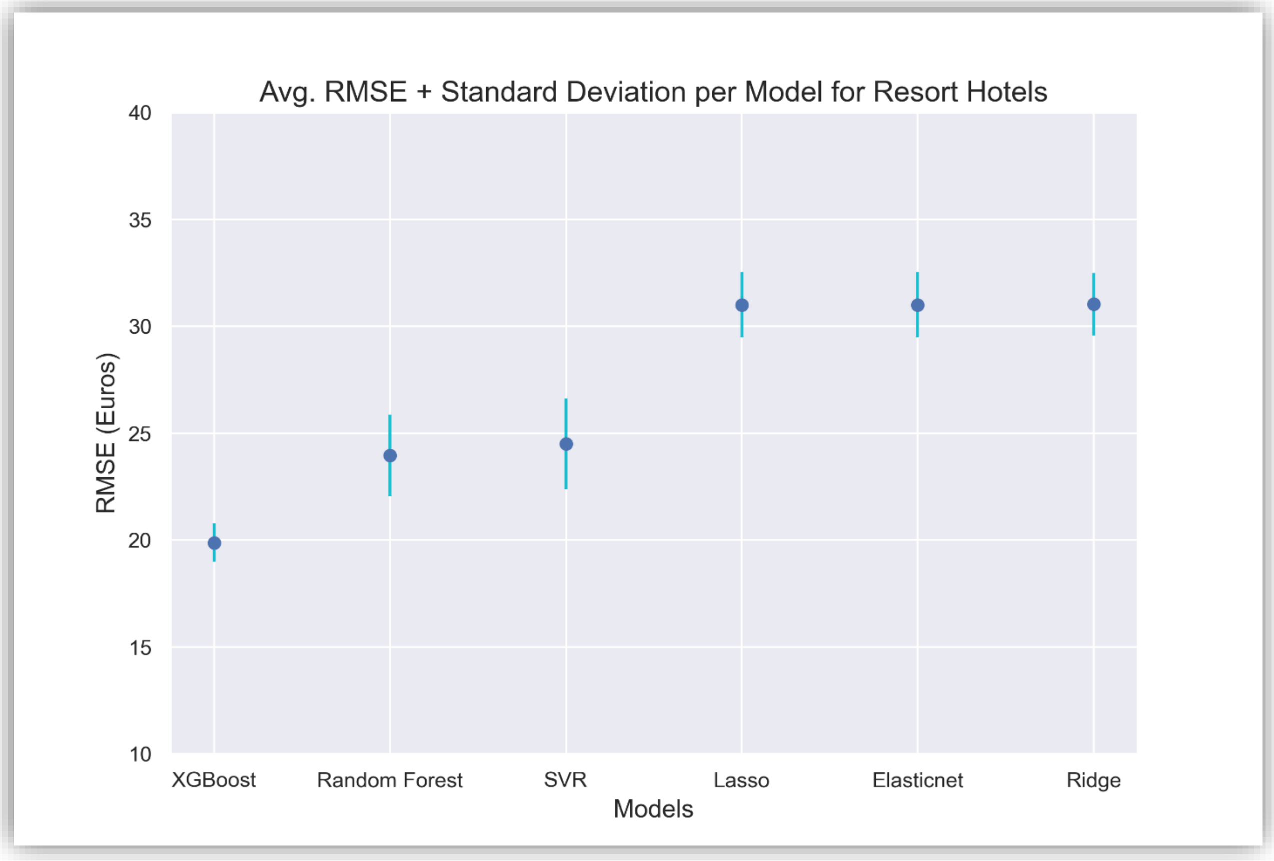
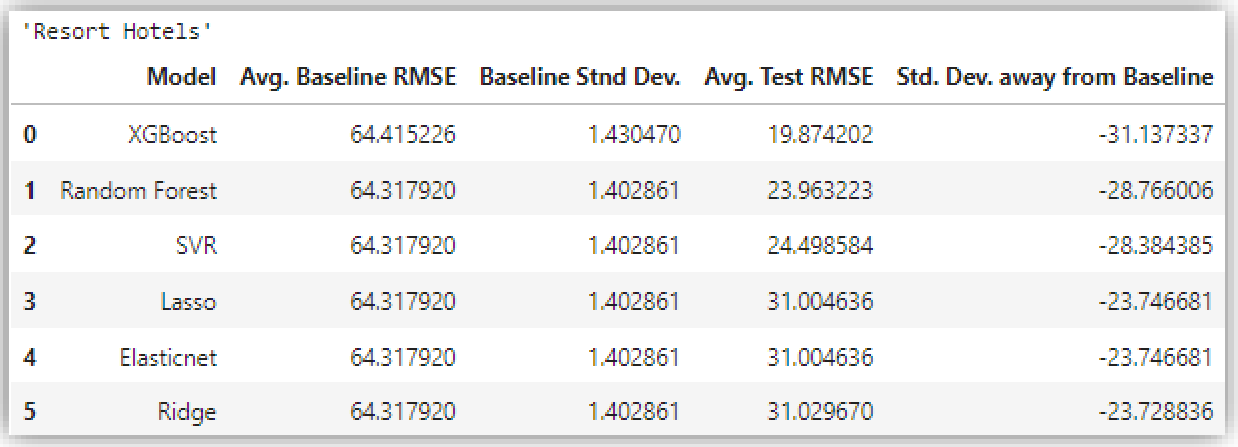


Figure: 4.1.4 Average RMSE for City Hotel’s ADR for each machine learning model across 5 random states



```
'Resort Hotels'
```

	Model	Avg. Baseline RMSE	Baseline Stnd Dev.	Avg. Test RMSE	Std. Dev. away from Baseline
0	XGBoost	64.415226	1.430470	19.874202	-31.137337
1	Random Forest	64.317920	1.402861	23.963223	-28.766006
2	SVR	64.317920	1.402861	24.498584	-28.384385
3	Lasso	64.317920	1.402861	31.004636	-23.746681
4	Elasticnet	64.317920	1.402861	31.004636	-23.746681
5	Ridge	64.317920	1.402861	31.029670	-23.728836

Figure 4.1.5 Table of Resort Hotel's Average Baseline RMSE, Baseline Score Standard Deviation, Average Test RMSE and the number of standard deviations away from the Baseline RMSE for each model

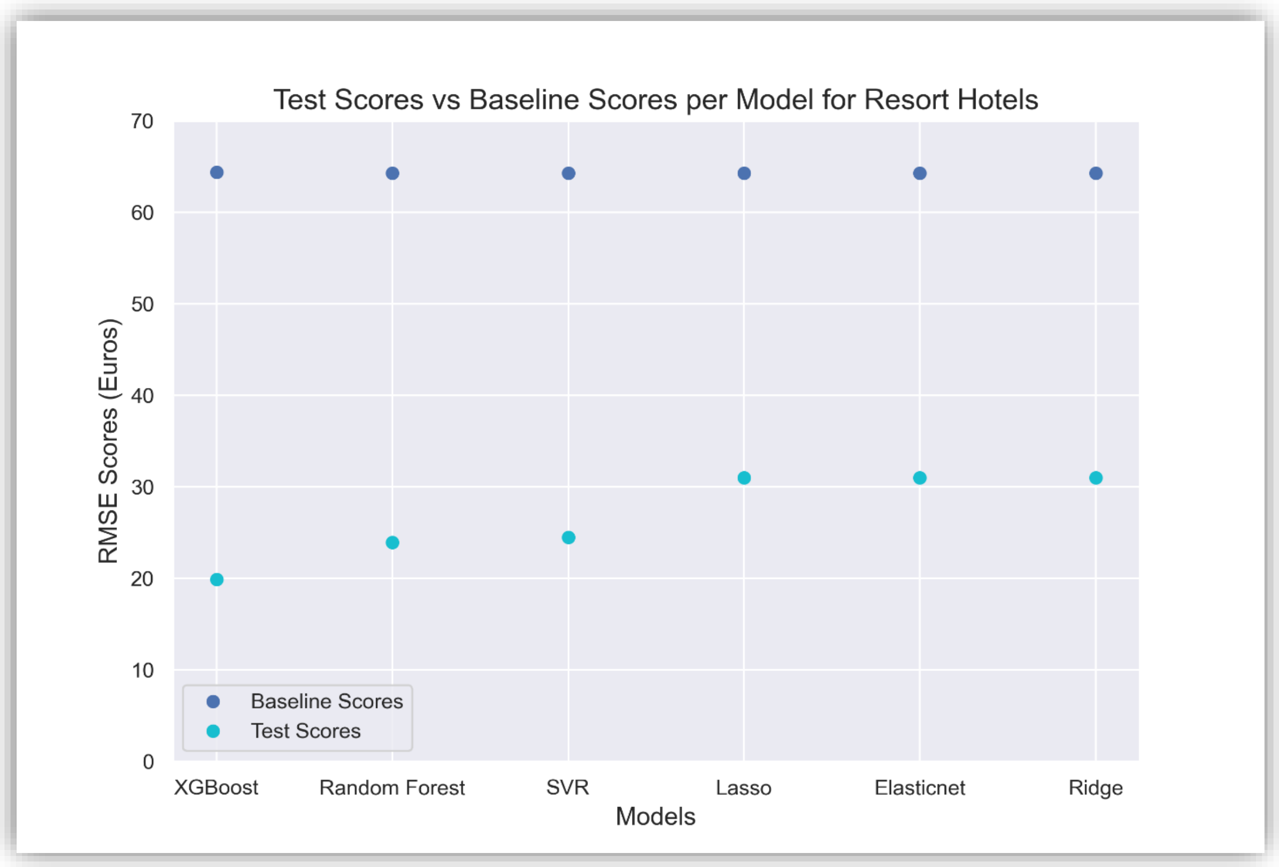


Figure 4.1.6 Average Baseline RMSE vs Average Test RMSE for City Hotels for each machine learning model across 5 random states

For resort hotels, **Figure 4.1.4** indicates that the best performing model is also XGBoost Regression with an average RMSE of 19.87 euros and the smallest standard deviation of 0.71. Furthermore, each model achieved better results than their respective baseline scores. Regarding the best model, XGBoost, the average baseline RMSE is 64.41 euros with a standard deviation 1.43. As a result, the average test RMSE for resort hotels is around -31 standard deviations away (**Figure 4.1.2**). With a higher standard deviation count, the resort hotel XGB model outperformed the city hotel XGB model in terms of performance against their baselines.

4.2 Global Importance

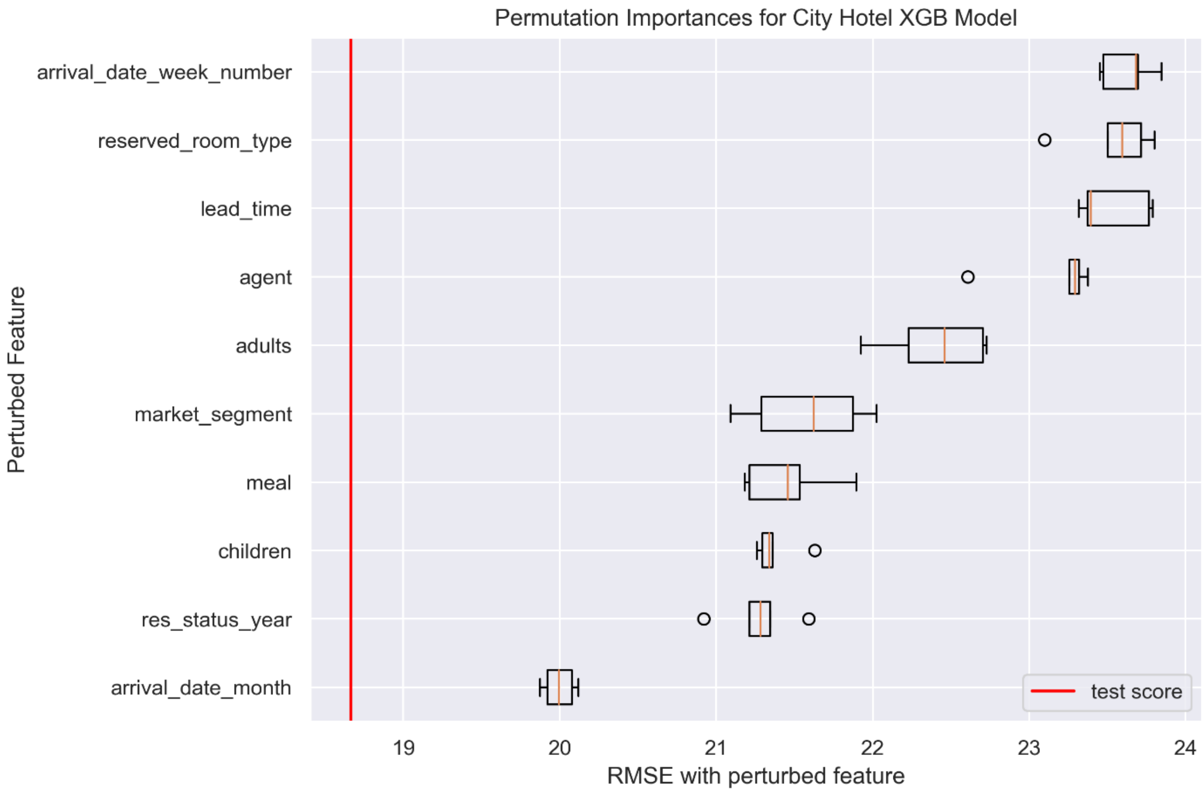


Figure 4.2.1 Permutation Importance for City Hotels using best model, XGBoost

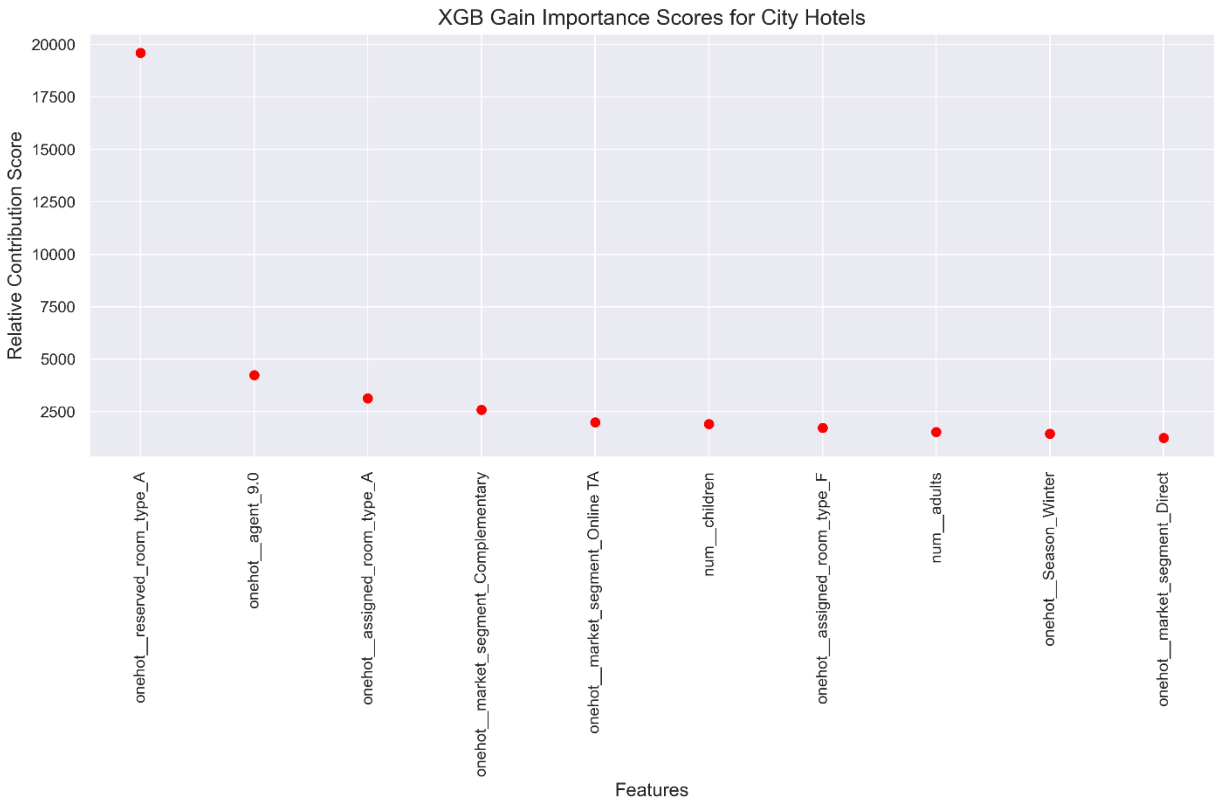


Figure 4.2.2 XGBoost Gain metric for City Hotels

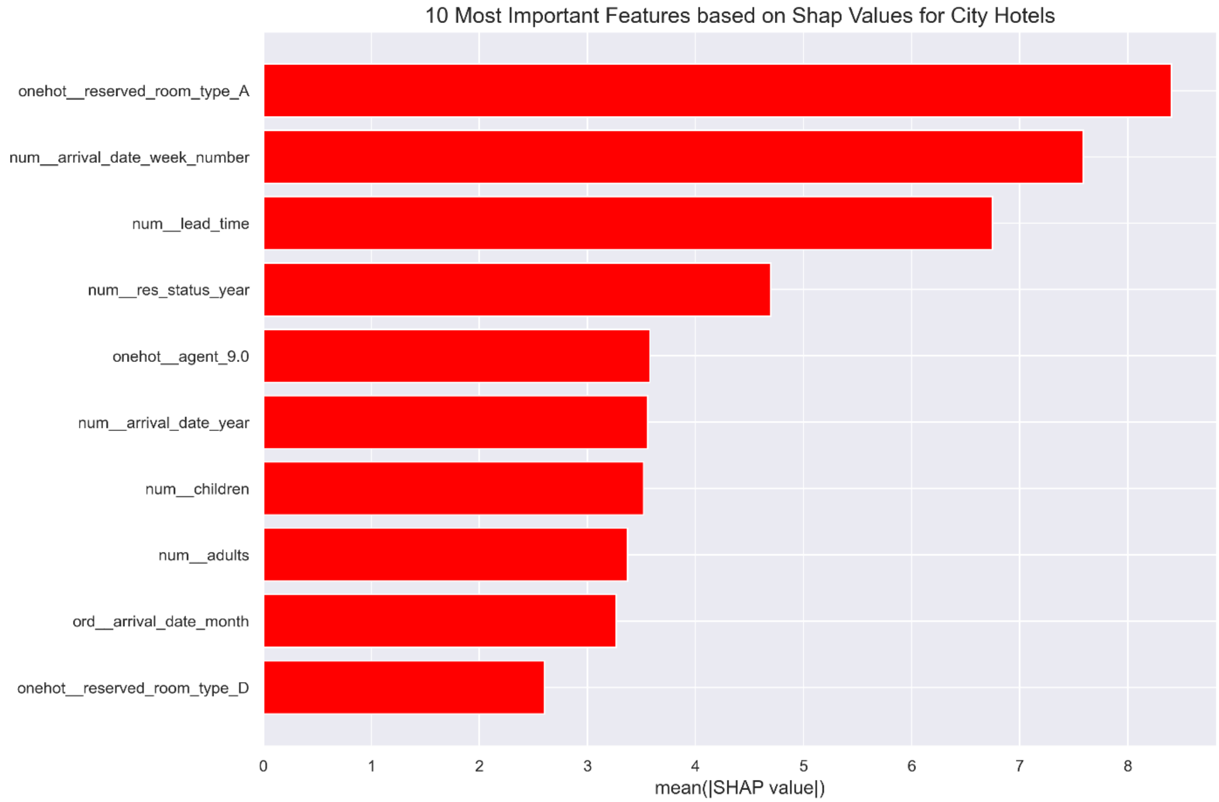


Figure 4.2.3 Global Importance through SHAP values for City Hotels using best model, XGBoost

To visualize the feature importance, 3 different methods were implemented: Permutation Importance, XGB Gain and Shap Values for global importance. All three metrics were computed using the XGBoost model due to its performance relative to other models. Across the results of all three methods, the feature, ‘reserved_room_type_A’/’reserved_room_type’ stood out as it appeared in the top 3 most important features for all methods. Without knowledge on the significance of room type A, it can be assumed that its price may point to room rates as one of the two price extremes: expensive or cheap.

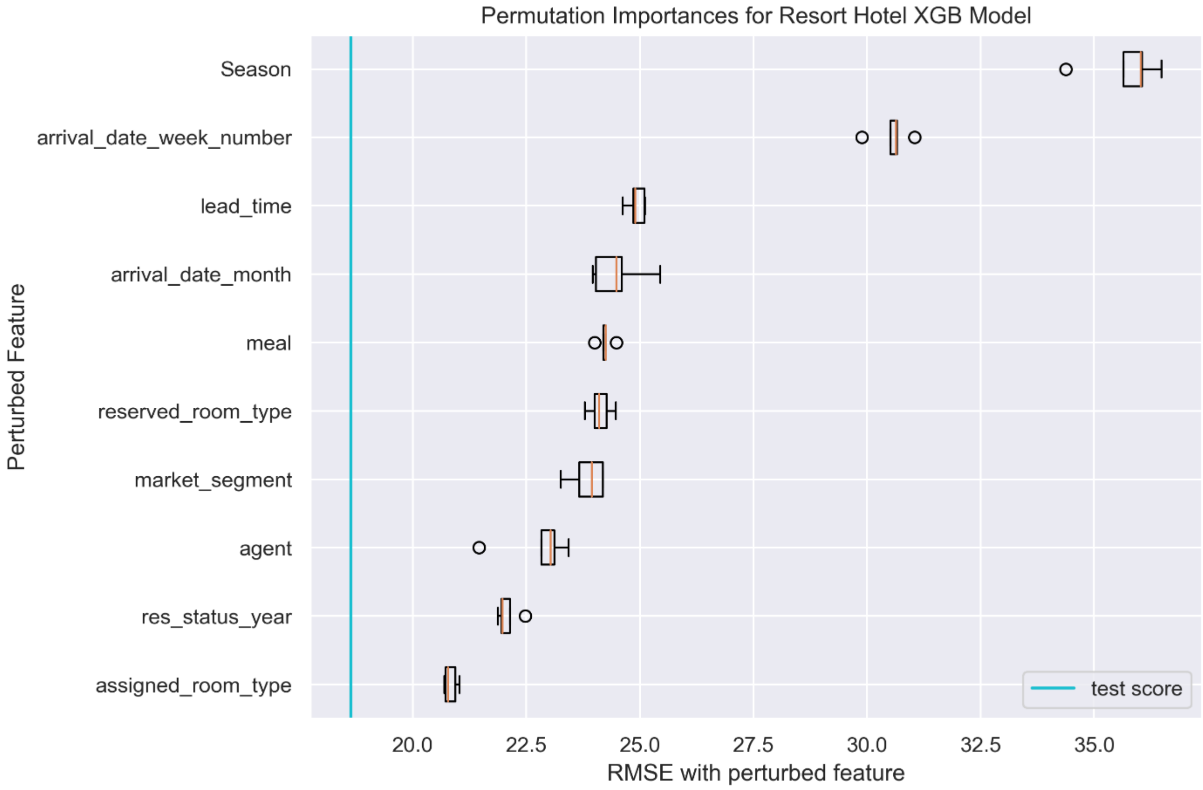


Figure 4.2.4 Permutation Importance for Resort Hotels using best model, XGBoost

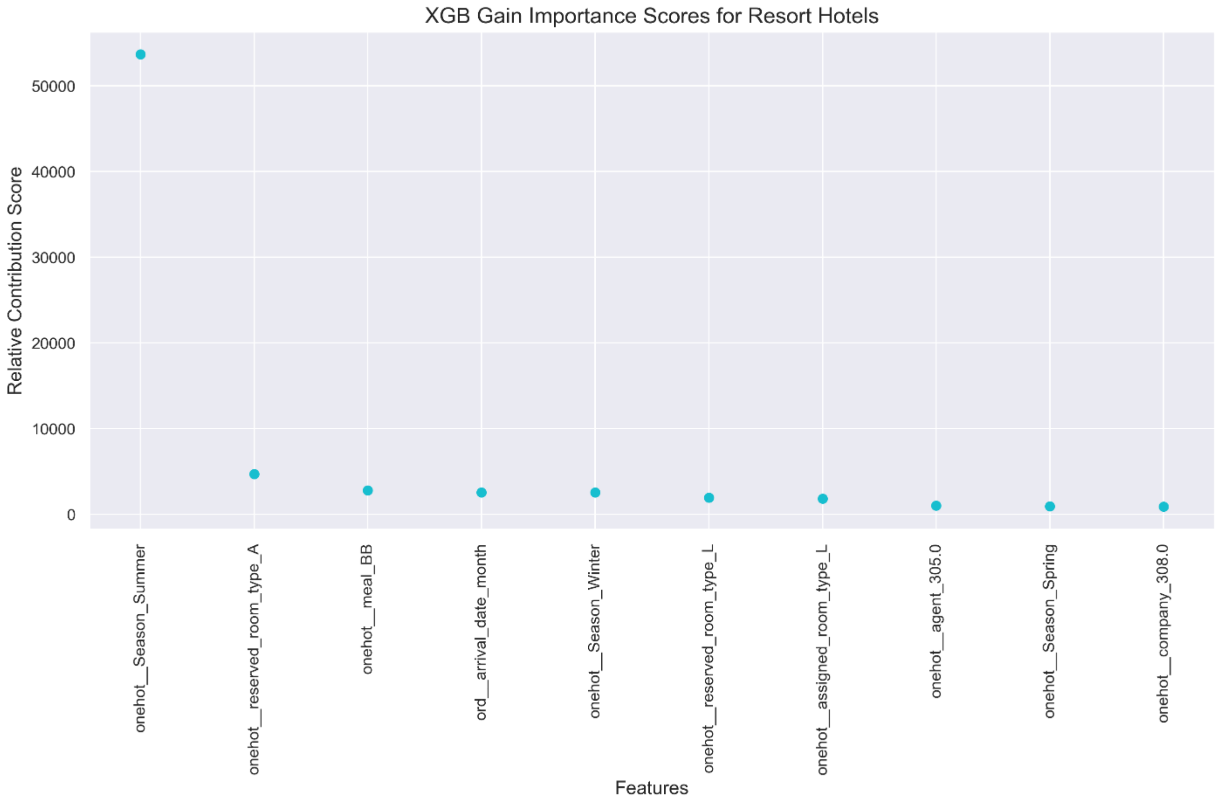


Figure 4.2.5 XGBoost Gain metric for Resort Hotels

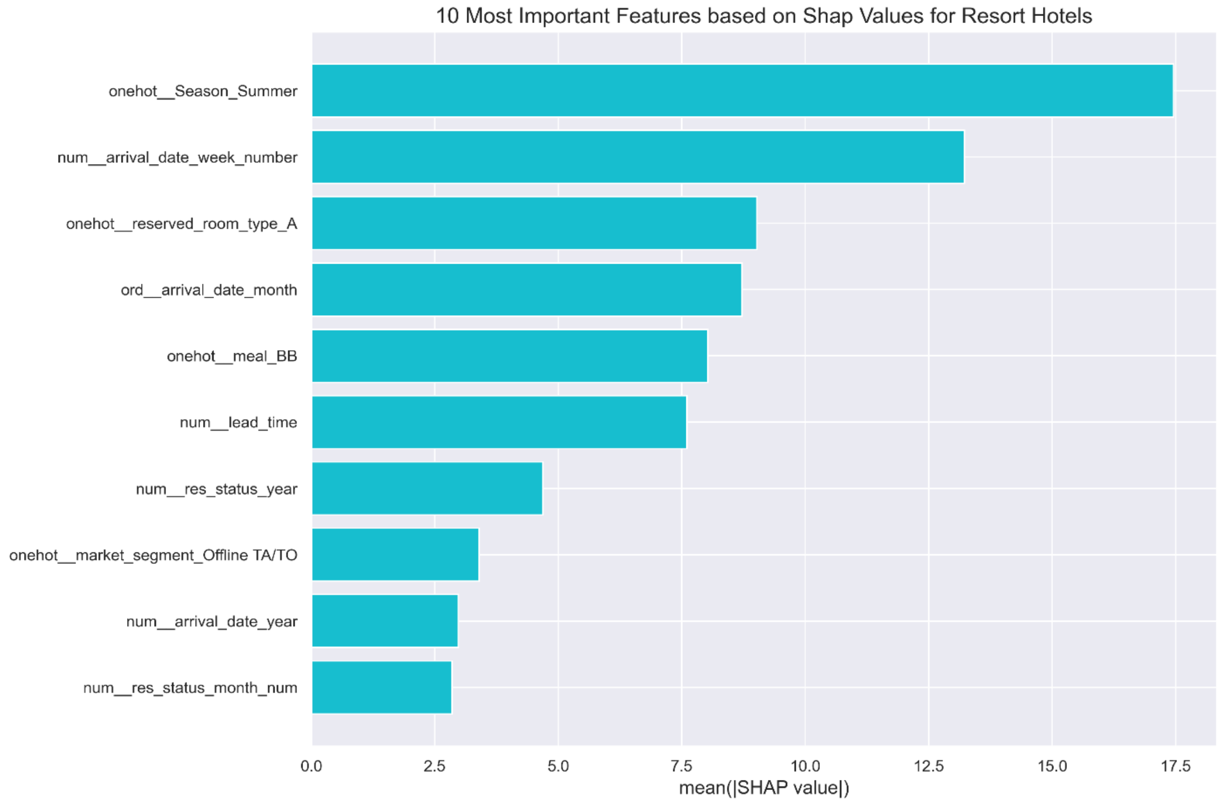


Figure 4.2.6 Global Importance through SHAP values for City Hotels using best model, XGBoost

For resort hotels, all three metrics were computed using the XGBoost model. . The feature, 'Season', specifically Summer was deemed the most important feature across all models. **Figure 2.1** confirms these results since the average adr for resort hotels see a dramatic incline during the months of June, July August.

4.3 Local Importance

The following figures below represent a few force plots created to explain the local SHAP value importance at different index values.

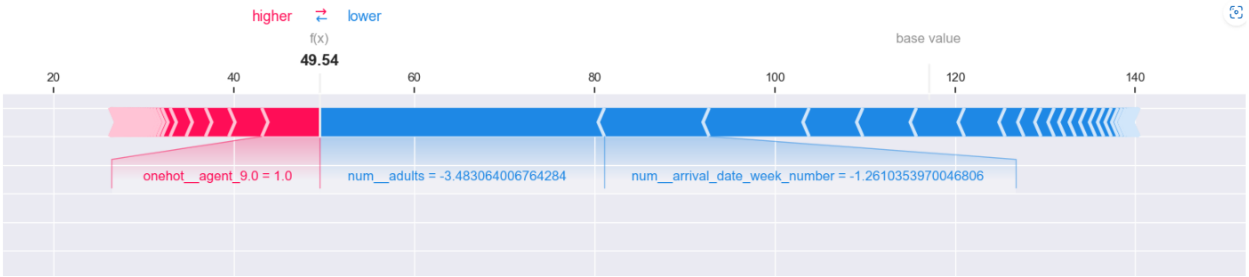


Figure 4.3.1 SHAP value local importance at index: 0 in City Hotel dataset

Based on the preprocessed feature values at index = 0 in the city hotel dataset, the predicted adr is 104.23 euros. One visible feature that positively contributes to this output is ‘onehot__agent_9.0’. Features that negatively contribute to the output include ‘num__adults’ and ‘num__arrival_date_week_number’.

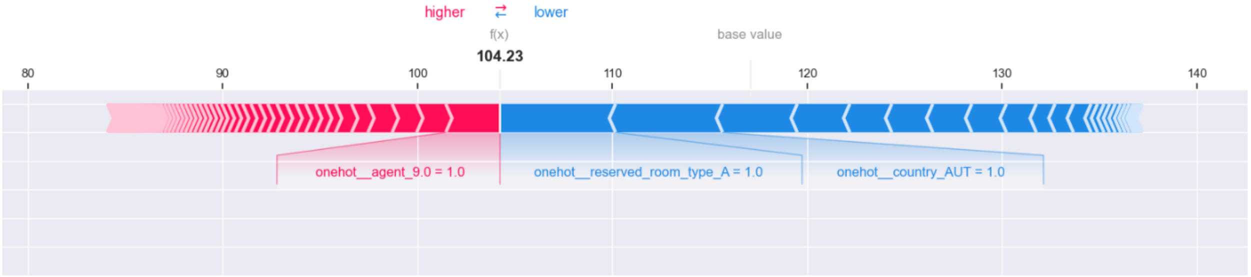


Figure 4.3.2 SHAP value local importance at index: 300 in City Hotel dataset

Based on the preprocessed feature values at index = 300 in the city hotel dataset, the predicted adr is 104.23 euros. One visible feature that positively contributes to this output is ‘onehot__agent_9.0’. Features that negatively contribute to the output include ‘onehot__reserved__room_type_A’ and ‘onehot__country_AUT’.

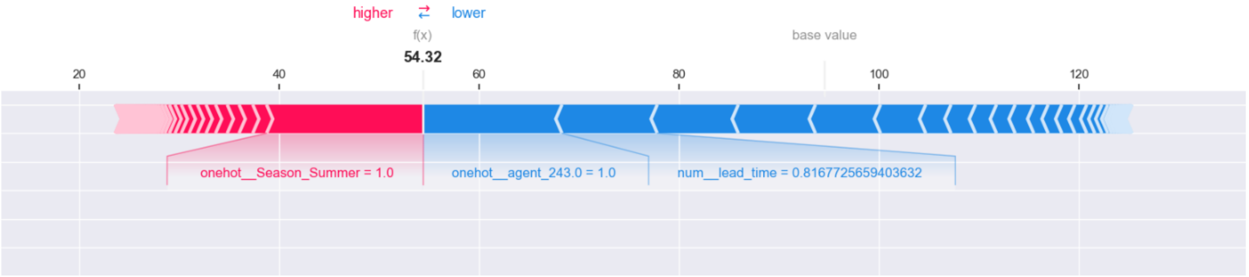


Figure 4.3.3 SHAP value local importance at index: 0 in Resort Hotel dataset

Based on the preprocessed feature values at index = 0 in the resort hotel dataset, the predicted adr is 54.32 euros. One visible feature that positively contributes to this output is ‘onehot__Season_Summer’. Some of the features that negatively contribute to the output include ‘onehot__agent_243.0’ and ‘num__lead_time’.

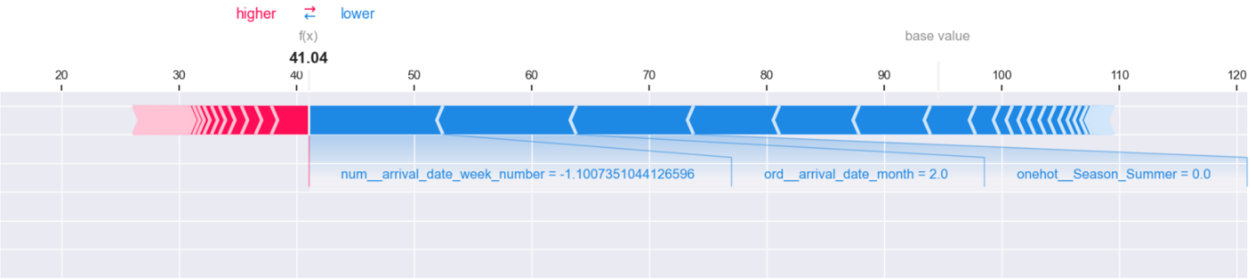


Figure 4.3.2 SHAP value local importance at index: 800 in City Hotel dataset

Based on the preprocessed feature values at index = 800 in the resort hotel dataset, the predicted adr is 41.04 euros. Although there are features that positively contribute to the predicted output, none are visible in the output. Some of the features that negatively contribute to the output include ‘num__arrival_date_week_number’, ‘ord__arrival_date_month’, ‘onehot_Season_Summer’.

5. Outlook

There are a few ways that the models and analysis can be further improved. First, additional data can be added to bolster the predictability of the dataset. The additional data can be split into two groups, first being data related to hotel actions such as promotions and advertising spend. Hotels can leverage both data sets to understand how advertised promotions drive a lift in average daily rate through different customers and regions. Secondly, demographic or psychographic data can be leveraged for consumer centric analyses, such as occupation and disposal income. Greater transparency in certain feature values would allow for more meaningful interpretations that can lead to decisive business decisions. Features like ‘reserved_room_type’ had its original value labels scrubbed in favor of letters of the alphabet, which creates uncertainty on its importance. It becomes difficult to definitively provide concrete and meaningful interpretation of these features.

1. References

Adnan, Dimas. “Predicting a Hotel Booking Demand.” Medium, Towards Data Science, 26 Aug. 2020, https://towardsdatascience.com/predicting-a-hotel-booking-demand-7608a7dbf5a4.

[1] Lahon, Anurag. “Hotel Booking Demand Project Using Different Models.” Medium, Becoming Human: Artificial Intelligence Magazine, 8 May 2020, https://becominghuman.ai/hotel-booking-demand-project-using-different-models-339b7c86235e.

[2] Published by S. Lock, and Jul 5. “Hotel and Resort Industry Market Size Worldwide 2021.” Statista, 5 July 2022, https://www.statista.com/statistics/1186201/hotel-and-resort-industry-market-size-global/.

[3] Antonio, Nuno, et al. “Hotel Booking Demand Datasets.” Data in Brief, vol. 22, 2019, pp. 41–49., https://doi.org/10.1016/j.dib.2018.11.126.

Mostipak, Jesse. “Hotel Booking Demand.” Kaggle, 13 Feb. 2020, https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand.

Soomro, Aaqib Qadeer. “Exploratory Data Analysis of the Hotel Booking Demand with Python.” Medium, Analytics Vidhya, 9 Sept. 2021, https://medium.com/analytics-vidhya/exploratory-data-analysis-of-the-hotel-booking-demand-with-python-200925230106.