Analysis of Hotel Reservation Data set using Machine learning Techniques

Gillian Kingsbury
MS Business Analytics
University of St Thomas

Danayit Shewamene
MS Data science
University of St Thomas

Mansoo Cho
MS in Data Science
University of St Thomas

Abstract

For a hotel to be profitable it must have a general idea of whether its customers will honor their reservations or cancel. If hotel management has a pre indicator whether a hotel reservation will be cancelled or not they can then better manage their operations. The dataset utilized in this analysis contains hotel reservation data. The dataset consists of 36,275 instances that contain details about a hotel reservation's booking status. The data is sourced from Kaggle; it has 17 total attributes with 5 categorical or binary and 12 numerical. We defined the problem as a binary classification, where the target variable is booking status, which is either be 'cancelled" or "not cancelled." We evaluated several machine learning algorithms for predicting hotel reservation cancellations including Logistic Regression, Random Forest, SVM, KNN, Decision Tree, Naïve Bayes and Voting Classifier Ensemble. We then applied feature reduction techniques, backward selection and PCA, on the data and re-trained the model and assessed the model's performance. We used the scikit-learn library in python to implement these algorithms. The results show that Random Forest has the highest accuracy and thus achieved the best performance. The random forest algorithm had an accuracy of 88.88%. Our feature importance analysis showed that lead time (the number of days between the booking and start of reservation) and average price per room are the two most important indicators of whether a reservation will be honored. This result agrees with our initial hypothesis for this analysis. Our study is limited to one dataset, performing similar studies on various datasets could provide further insights and yield better results for future studies.

Introduction

It is crucial for hotel managers to have a grasp on whether a hotel reservation will be honored or not. This will allow hotels to better manage their operations and boost customer satisfaction. Identifying the features that impact cancellations can help the hotel improve their booking policies and reduce the number of cancellations. Our analysis of the hotel reservation dataset will provide answers to what attributes are a key factor in resulting in a successful booking or cancellation. The information gained from this analysis can benefit hotels when they are offering rooms and can help hotels avoid over or under booking their rooms.

The dataset utilized in this analysis contains hotel reservation data. The dataset consists of 36,275 instances of hotel reservation booking status and its attributes. The data is sourced from Kaggle, it has 17 total attributes with 5 categorical or binary and 12 numerical attributes. This includes features such as the rates of the hotel rooms, booking status and reservation details. Our hypothesis is that the price and the length of stay will have the highest impact on whether customers will end up cancelling their hotel booking. We apply various machine learning methods to build predictive models and to assess their accuracies and overall performance.

Problem Definition

We defined the problem as a binary classification, where the target variable is booking status, which is either 'cancelled" or "not canceled". We were interested in examining which features have meaningful impacts on predicting booking status. Furthermore, we decided to look at the feature importance of length of stay (lead time) and average room price using different machine learning algorithms.

Hypothesis: Would knowing lead time and average price per room help predict booking status?

After some data processing, we obtained 31 features (one-hot encoded, standardized and oversampled with SMOTE algorithm). We were interested in examining two feature engineering techniques to extract meaningful features. We decided to use PCA and backward elimination for feature engineering.

Methodology

We utilized the scikit-learn library in python both to preprocess the dataset and to implement the various machine learning algorithms. One-hot encoding from Pandas library was used to encode the categorical columns. Feature scaling was done to standardize (Standard Scaler from scikit-learn) the features to ensure they were on the same scale. We selected the random state parameter of 42 where applicable to have uniform results. The dataset was split to 70 percent as training and 30 percent as test. To overcome the class imbalance, we utilized SMOTE (from imblearn library) to oversample the minority classes in the training set. Then we applied feature engineering techniques to construct three feature matrices: 1. All feature matrix, 2. PCA feature matrix (scikit-learn library), and 3. Backward elimination matrix (using statsmodels library). We used the following machine learning algorithms from scikit-learn to analyze the data on the three feature matrices we created—Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier, Random Forest Classifier, K-Nearest Neighbors (KNN), Naïve Bayes and Voting Classifier.

Description of data:

The raw, unprocessed data has 17 total attributes with 5 categorical or binary variables and 12 numerical variables.

1. No. of adults (numeric): number of adults per booking
2. No. of children (numeric): number of adults per booking
3. No. of weekend nights (numeric): number of weekend nights booked.
4. No. of weekday nights (numeric): number of weekday nights booked.
5. Type of meal plan (categorical): Meal plans 1 – 3 and no choice
6. Required car parking space (binary): does the customer require parking?
7. Room type (categorical): Room type 1 – 7
8. Lead time (numeric): No. of days between the date of booking and the arrival date
9. Arrival year (numeric)
10. Arrival month (numeric)
11. Arrival day (numeric)
12. Market segment type (categorical): online, offline, aviation, corporate or complementary
13. Repeated guest (binary): Is the customer a repeated guest?
14. No. of previous cancellations (numeric)
15. No. of bookings not cancelled (numeric): number of previous bookings that were not cancelled by the customer prior to the current booking
16. Average price per room (numeric): in euros, average price per day of the reservation.
17. No. of special requests (numeric): Total number of special requests made by the customer.

The target variable is **Booking Status** which is a flag variable indicating if the reservation was cancelled or not.

Data Processing:

1. We label-encoded the target variable such that “not cancelled” is coded as 0; “cancelled” is coded as 1.
2. We decided to create a new variable “arrival season” which may help predict the booking status depending on which season the reservation was booked. We used “arrival month" to extract arrival season information. If arrival month was either 12, 1 or 2, then arrival season was coded as “winter.” Else if arrival month was 3, 4 or 5, then arrival season was coded as “spring. Else if arrival month was 7, 8, or 9, then arrival season was coded as “summer.” If arrival month was 9, 10 or 11, then arrival season was coded as “fall”.
3. “Lead time” already contained redundant information with “arrival year”, “arrival month,” and “arrival day”; we decided to drop “arrival year”, “arrival month,” and “arrival day.”

4. We one-hot encoded the categorical/binary variables "room type," "type of meal plan," market segment type" and "arrival season." (See Appendix – Table A)
5. We split the data into training and test sets with 30% being the test set size.
6. We scaled the data using Standard Scaler (standardization)
7. There was imbalance in the target variable; we addressed the issue of imbalance by using SMOTE resampling method.

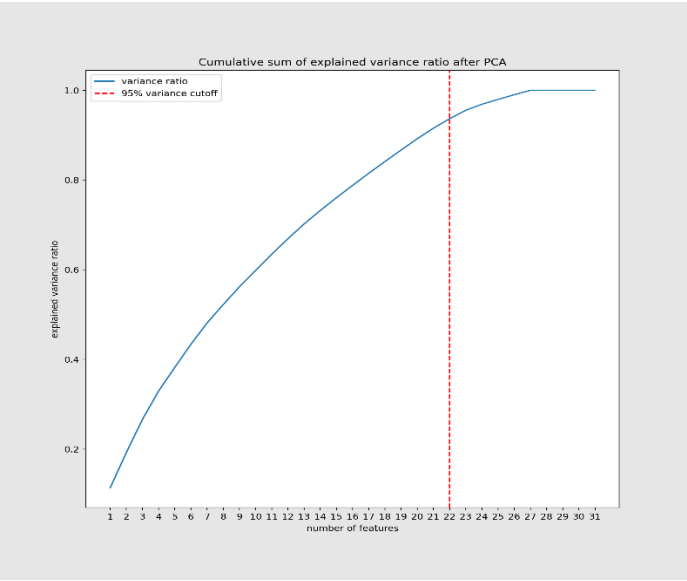
Feature Engineering:

We wanted to compare models based on 2 feature engineering techniques: PCA and backward elimination. We were interested in what feature engineering method would result in the best performance of the models.

We had the following 3 feature matrix to compare model performances: 1. All feature matrix with 31 features, 2. PCA generated feature matrix, and 3. backward eliminated feature matrix.

First, we performed PCA on the training dataset. We looked at the cumulative sum of explained variance by the PCA features. We decided to keep the features that captured 95% of the total variability.

Figure 1: Cumulative sum of explained variance ratio of PCA features



The plot shows that 22 PCA features accounted for 95% of the total variability.

We utilized a backward feature selection algorithm (Backward Elimination) to explore what feature engineering techniques would give the best performance of the models. The model initially was trained on 31 features, using logistic regression, we found that 19 features were meaningful in predicting booking status.

Backward elimination steps:

1. Fit a logistic regression on all 31 features.
2. Remove a feature with the highest p-value (p-value > significance level = 0.05)
3. Fit a logistic regression on remaining features.
4. Repeat steps 2 and 3 until the p-values of all features became significant (significance level = 0.05; we want p-values to be < 0.05).

Variables remaining after backward elimination (19 features): (See Appendix – Table B)

0. x0: intercept
1. x2: no_of_children
2. x3: no_of_weekend_nights
3. x4: no_of_week_nights
4. x5: required_car_parking_space
5. x6: lead_time
6. x7: repeated_guest
7. x8: no_of_previous_cancellations
8. x10: avg_price_per_room
9. x11: no_of_special_requests
10. x12: room_type_reserved_Room_Type 1
11. x16: room_type_reserved_Room_Type 5
12. x17: room_type_reserved_Room_Type 6
13. x18: room_type_reserved_Room_Type 7
14. x22: type_of_meal_plan_Not Selected
15. x23: market_segment_type_Aviation
16. x25: market_segment_type_Corporate
17. x27: market_segment_type_Online
18. x29: arrival_season_spring
19. x31: arrival_season_winter

Analysis

1. *Logistic regression*

    - Logistic regression on all 31 features resulted in the accuracy of 78.01 %.
    - **Logistic regression on 22 PCA features resulted in the accuracy of 78.18 %.**
    - Logistic regression on 19 backward elimination features resulted in the accuracy of 77.95%

Result: Logistic regression on PCA feature matrix resulted in the best performance in predicting booking status.

2. *SVM*

SVM models have a hyper parameter "kernel." SVM uses kernel functions to project data into a higher dimension for analysis. RBF is short for Radial Basis Function and "poly" is a polynomial function. We wanted to compare the kernels rbf, and poly on all three feature matrices and find which one resulted in the highest accuracy.

- **SVM with rbf kernel on all 31 features resulted in the accuracy of 81.641%**
- SVM with rbf kernel on 22 PCA features resulted in the accuracy of 81.099%.
- SVM with rbf kernel on 19 backward elimination features resulted in the accuracy of 79.647%.
- SVM with poly kernel on all 31 features resulted in the accuracy of 81.457%
- SVM with poly kernel on 22 PCA features resulted in the accuracy of 80.952%
- SVM with poly kernel on 19 backward elimination features resulted in the accuracy of 79.436%

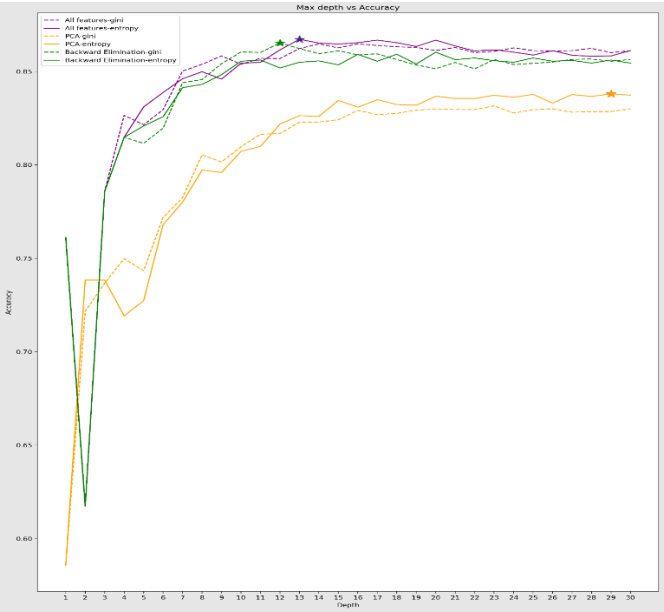
Result: The hyper parameter kernel="rbf" resulted in the best performance on predicting booking status. We chose our hyper parameter for SVM to be "rbf." Also, SVM with rbf on all 31 features resulted in the best performance in predicting booking status.

3. *Decision Trees*

Decision tree algorithm has two hyper parameters "criterion" and "max depth." Decision tree uses certain "criterions" to split data. "Gini" criterion calculates the probability of incorrect classification; "entropy" criterion tries to maximize the information gain after the data split. "Max depth" is a threshold specifying when to stop splitting nodes. We wanted to find the combination of hyper parameters that resulted in the highest accuracy of predicting booking status.

First compare criteria: Gini vs entropy

Figure 2: Accuracy of Decision Tree Classifiers by criterions



Decision tree models were fitted using the 2 criteria: Gini and entropy per feature matrix. The stars mark the highest accuracy per each feature matrix.

- For all feature matrix, criterion = entropy gave us the maximum accuracy
- For 22 PCA features, criterion = entropy gave us the maximum accuracy.
- For 19 backward elimination features, criterion = Gini gave us the maximum accuracy.

Then we wanted to find the max_depth that would give us the highest accuracy on each feature matrix.

Figure 3: Accuracy of Decision Tree Classifiers across max depths

- For all 31 features, max_depth = 13 gave us the maximum accuracy.
- For 22 PCA features, max_depth = 29 gave us the maximum accuracy.
- For 19 backward elimination features, max_depth = 12 gave us the maximum accuracy.

The last step was to output the highest accuracies with our chosen hyper parameters.

- Decision Tree with all features accuracy (max_depth=13, criterion=entropy): 86.724%.
- Decision Tree with 22 features (PCA) accuracy (max_depth=29, criterion=entropy): 83.80%
- **Decision Tree with 19 features (backward elimination) accuracy (max_depth=12, criterion=gini): 88.81%**

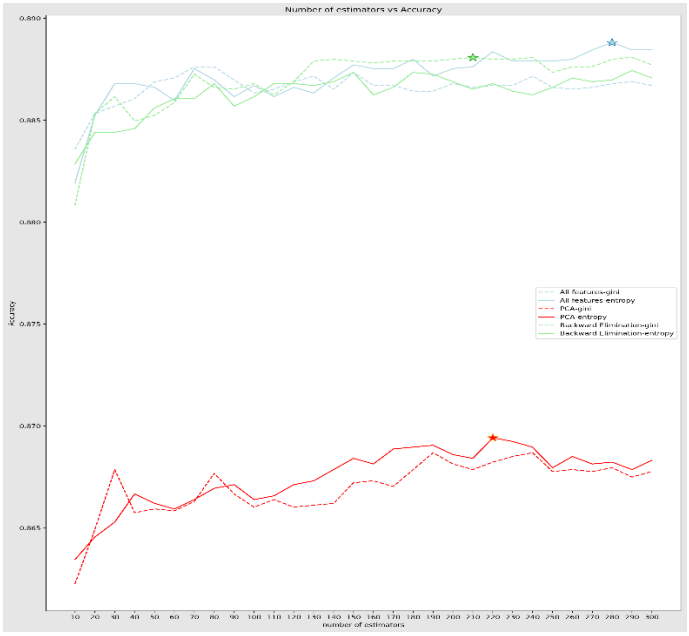
Result: Decision tree algorithm with criterion = Gini, max_depth = 12 on 19 backward elimination features resulted in the best performance on predicting booking status.

4. *Random Forest*

Random forest algorithms have 2 hyper parameters: number of estimators (n_estimators) and criterion. The number of estimators specifies the number of Decision Trees within a Random Forest; criterion is the same as that of Decision Tree's. We wanted to find the combination of hyper parameters that would result in the best accuracy of predicting booking status.

First compare criteria: Gini vs entropy.

Figure 4: Accuracy of Random Forest Classifiers by criterions



Random forest models were fitted using the 2 criteria: Gini and entropy per feature matrix. The stars mark the highest accuracy per each feature matrix.

- For all feature matrix, criterion = entropy gave us the maximum accuracy
- For 22 PCA features, criterion = entropy gave us the maximum accuracy.
- For 19 backward elimination features, criterion = gini gave us the maximum accuracy.

Then we wanted to find the n_estimators that would give us the highest accuracy on each feature matrix.

Figure 5: Accuracy of Random Forest Classifiers across Number of Estimators

- For all 31 features, n_estimators = 280 gave us the maximum accuracy.
- For 22 PCA features, n_estimators = 220 gave us the maximum accuracy.
- For 19 backward elimination features, n_estimators = 210 gave us the maximum accuracy.

The last step was to output the highest accuracies with our chosen hyper parameters.

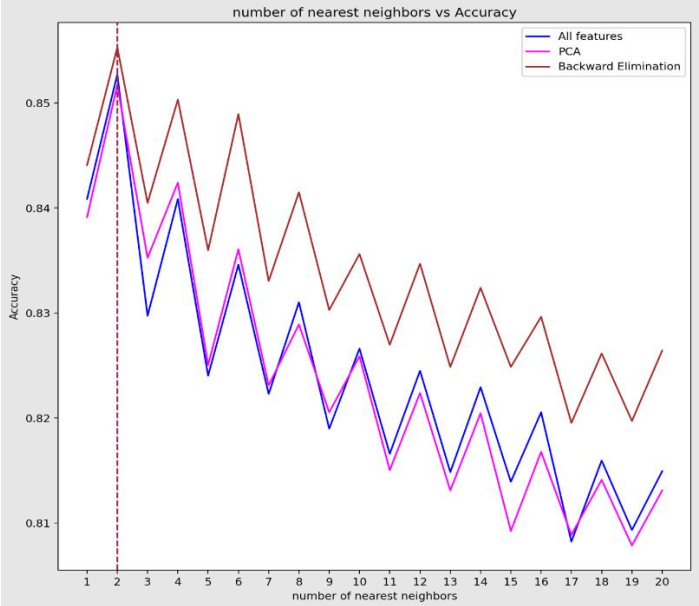
- **Random Forest with all features accuracy (n_estmators=280, criterion=entropy): 88.882%**
- Random Forest with 22 features (PCA) accuracy (n_estmators=220, criterion=entropy): 86.943%
- Random Forest with 19 features (backward elimination) accuracy (n_estmators=210, criterion=gini): 88.808%

Result: Random Forest algorithm with criterion = entropy, n_estimators = 280 on all 31 features resulted in the best performance on predicting booking status.

5. *KNN*

KNN has a hyper parameter number of nearest neighbors (n_neighbors). The number of nearest neighbors specifies the number of neighbors to calculate the distance from. We wanted to find the best value of n_neighbors that would result in the highest accuracy of predicting booking status.

Figure 6: Accuracy of KNN Classifiers by number of nearest neighbors



KNN models were fitted to the number of nearest neighbors ranging from 1 to 20. N_neighbors = 2 resulted in the best accuracies for all 3 feature matrices.

- KNN with all features accuracy (n_neighbors=2): 85.271%
- KNN with 22 features (PCA) accuracy (n_neighbors=2): 85.170%
- **KNN with 19 features (backward elimination) accuracy (n_neighbors=2): 85.528%**

Result: KNN algorithm with n_neighbors = 2 on 19 backward elimination features resulted in the best performance on predicting booking status.

6. *Naïve bayes*

Naïve bayes models were built on all three feature matrices; accuracies were calculated accordingly.

- Naive Bayes accuracy with all 31 features: 40.788%
- **Naive Bayes accuracy with 22 features: 52.449%**
- Naive Bayes accuracy with 19 features: 49.60%

Result: Naïve Bayes algorithm on 19 Backward elimination features resulted in the best performance on predicting booking status.

7. *Voting classifier*

After learning which hyper parameters resulted in the best performance in predicting booking status, we decided to build Voting Classifiers on all three feature matrices. Below is a summary of chosen models for each voting classifier.

1. All feature matrix (31 features)
- Logistic Regression: N/A
- SVM: kernel=rbf
- Decision Tree: max_depth=13, criterion=entropy
- Random Forest: n_estimators = 280, criterion=entropy
- KNN: n_neighbprs = 2
- Naive Bayes: N/A

2. PCA feature matrix (22 features)
- Logistic Regression: N/A
- SVM: kernel=rbf
- Decision Tree: max_depth=29, criterion=entropy
- Random Forest: n_estimators = 220, criterion=entropy
- KNN: n_neighbprs = 2
- Naive Bayes: N/A

3. Backward elimination feature matrix (19 features)
- Logistic Regression: N/A
- SVM: kernel=rbf
- Decision Tree: max_depth=12, criterion=Gini
- Random Forest: n_estimators = 210, criterion=Gini
- KNN: n_neighbprs = 2
- Naive Bayes: N/A

Using these hyper parameter values, we built voting classifiers on all 3 feature matrices.

1. All feature matrix result:

	precision	recall	f1-score	support
0	0.91	0.90	0.91	7317
1	0.81	0.82	0.81	3566
accuracy			0.88	10883
macro avg	0.86	0.86	0.86	10883
weighted avg	0.88	0.88	0.88	10883

2. PCA feature matrix result:

	precision	recall	f1-score	support
0	0.90	0.90	0.90	7317
1	0.80	0.80	0.80	3566
accuracy			0.87	10883
macro avg	0.85	0.85	0.85	10883
weighted avg	0.87	0.87	0.87	10883

3. Backward elimination feature matrix result:

	precision	recall	f1-score	support
0	0.91	0.90	0.91	7317
1	0.80	0.82	0.81	3566
accuracy			0.87	10883
macro avg	0.86	0.86	0.86	10883
weighted avg	0.88	0.87	0.87	10883

Result: Voting classifier on all 31 features gave us an F1 score of 0.91 and accuracy of 0.88. This gave us the best performance in predicting booking status.

(See Appendix Table C for the full summary of analysis results)

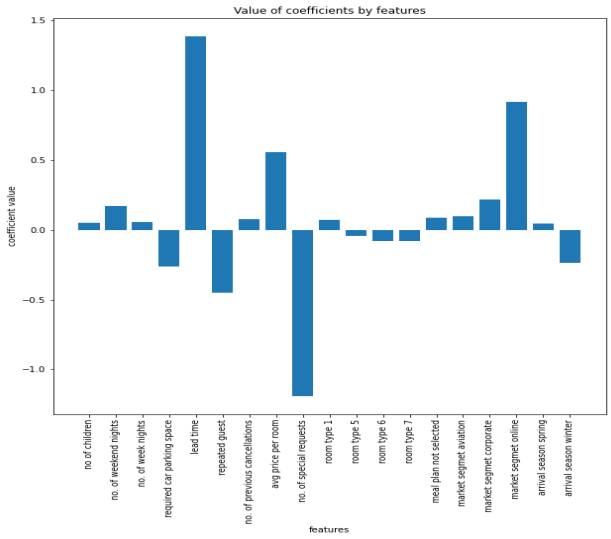
Feature Importance

To answer our hypothesis, we wanted to measure the importance of features on these algorithms: logistic regression, decision tree classifier, random forest classifier. The backward elimination technique resulted in 19 important features including “lead time” and “average price per room.”

1. *Logistic regression*

Feature importance is measured by the model coefficients. Positive coefficients indicate that a feature predicts class 1 (cancelled) and negative coefficient indicates a feature that predicts class 0 (not cancelled). The magnitude of coefficients tells us how much of an impact a feature has on predicting each class.

Figure 7: Feature Coefficient values of Logistic Regression



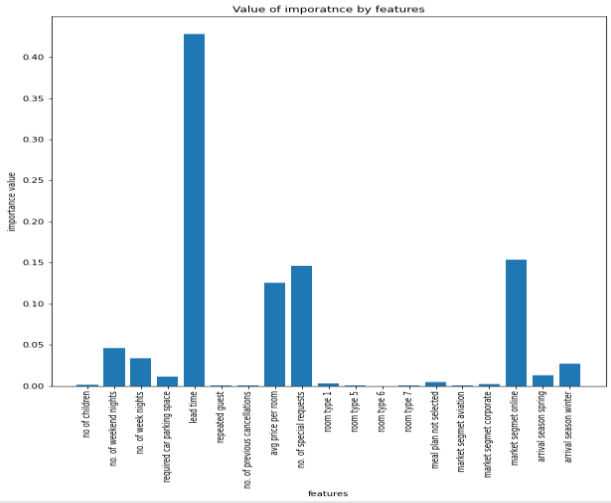
- Lead time has coefficient of 1.386
- Average price per room has coefficient of 0.555

This suggests that Lead Time and Average price per room have significant impact on predicting booking status being canceled.

2. *Decision Tree*

Decision tree classifier calculates feature importance scores which indicates how much impact a feature has on predicting target classes. The values are positive, and it simply measures the degree of impact regardless of class labels.

Figure 8: Feature Importance Score of Decision Tree



- lead time, coefficient: 0.428

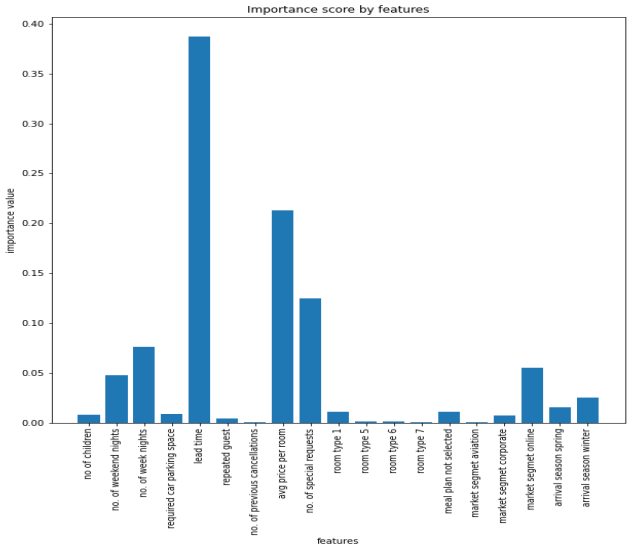
- avg price per room, coefficient: 0.126

The plot shows that lead time has the highest impact on predicting booking status; average price per room has considerable impact but less compared to lead time.

3. *Random Forest*

Random forest classifier also calculates feature importance score, and it has the same property as those from decision tree classifier.

Figure 9: Feature Importance Score of Random Forest



- lead time, importance score: 0.387

- avg price per room, importance score: 0.213

The plot shows that lead time has the highest impact on predicting booking status; average price has the second highest impact.

Conclusion

Our analysis considered several machine learning algorithms to study the hotel reservation status data, our results show that the most important predictors of hotel reservation cancellations are lead time and average price per room. This result agrees with our initial hypothesis for this analysis. We also explored different feature engineering techniques and analyzed their impact on predicting hotel reservations. Hotels have much to gain from a study like this which can give them the tools to predict the cancellation behavior and probabilities of customers so that they can better their policies and manage their resources. Our study is limited to one dataset, performing similar studies on various more

robust datasets could provide further insights and yield better results for future studies.

Reference

1. Brownlee, Jason. "How to Calculate Feature Importance with Python." *MachineLearningMastery.com*, 20 Aug. 2020, https://machinelearningmastery.com/calculate-feature-importance-with-python/
2. "Scikit Learn Documentation" *Scikit*, https://scikit-learn.org/stable/.
3. "Numpy User Guide." *NumPy User Guide - NumPy v1.24 Manual*, https://numpy.org/doc/stable/user/index.html#user.
4. "Pandas User Guide" *User Guide - Pandas 2.0.1 Documentation*, https://pandas.pydata.org/docs/user_guide/index.html#user-guide.
5. Gunjal, Satish. "Decision Tree." *Quality Tech Tutorials*, 9 June 2020, https://satishgunjal.com/decision_tree/.
6. Ellis, Christina. "Max Depth in Random Forests." *Crunching the Data*, 20 Sept. 2022, https://crunchingthedata.com/max-depth-in-random-forests/.

Appendix

(Table A. 31 features after one-hot encoding)

```
Data columns (total 31 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   no_of_adults                          36275 non-null  int64
 1   no_of_children                        36275 non-null  int64
 2   no_of_weekend_nights                  36275 non-null  int64
 3   no_of_week_nights                     36275 non-null  int64
 4   required_car_parking_space            36275 non-null  int64
 5   lead_time                             36275 non-null  int64
 6   repeated_guest                        36275 non-null  int64
 7   no_of_previous_cancellations          36275 non-null  int64
 8   no_of_previous_bookings_not_canceled  36275 non-null  int64
 9   avg_price_per_room                    36275 non-null  float64
 10  no_of_special_requests                36275 non-null  int64
 11  room_type_reserved_Room_Type 1        36275 non-null  uint8
 12  room_type_reserved_Room_Type 2        36275 non-null  uint8
 13  room_type_reserved_Room_Type 3        36275 non-null  uint8
 14  room_type_reserved_Room_Type 4        36275 non-null  uint8
 15  room_type_reserved_Room_Type 5        36275 non-null  uint8
 16  room_type_reserved_Room_Type 6        36275 non-null  uint8
 17  room_type_reserved_Room_Type 7        36275 non-null  uint8
 18  type_of_meal_plan_Meal Plan 1         36275 non-null  uint8
 19  type_of_meal_plan_Meal Plan 2         36275 non-null  uint8
 20  type_of_meal_plan_Meal Plan 3         36275 non-null  uint8
 21  type_of_meal_plan_Not Selected        36275 non-null  uint8
 22  market_segment_type_Aviation          36275 non-null  uint8
 23  market_segment_type_Complementary     36275 non-null  uint8
 24  market_segment_type_Corporate         36275 non-null  uint8
 25  market_segment_type_Offline           36275 non-null  uint8
 26  market_segment_type_Online            36275 non-null  uint8
 27  arrival_season_fall                   36275 non-null  uint8
 28  arrival_season_spring                 36275 non-null  uint8
 29  arrival_season_summer                 36275 non-null  uint8
 30  arrival_season_winter                 36275 non-null  uint8
dtypes: float64(1), int64(10), uint8(20)
```

(Table B. Logistic regression output)

Logit Regression Results

Dep. Variable:	target	No. Observations:	34146
Model:	Logit	Df Residuals:	34126
Method:	MLE	Df Model:	19
Date:	Mon, 24 Apr 2023	Pseudo R-squ.:	0.3403
Time:	18:14:31	Log-Likelihood:	-15615.
converged:	True	LL-Null:	-23668.
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
x0	-0.4701	0.017	-26.969	0.000	-0.504	-0.436
x2	0.0524	0.020	2.667	0.008	0.014	0.091
x3	0.1694	0.014	11.875	0.000	0.141	0.197
x4	0.0563	0.014	3.911	0.000	0.028	0.085
x5	-0.2632	0.018	-14.438	0.000	-0.299	-0.227
x6	1.3867	0.019	72.956	0.000	1.349	1.424
x7	-0.4518	0.063	-7.128	0.000	-0.576	-0.328
x8	0.0797	0.025	3.130	0.002	0.030	0.130
x10	0.5554	0.020	27.217	0.000	0.515	0.595
x11	-1.1917	0.019	-62.320	0.000	-1.229	-1.154
x12	0.0713	0.017	4.226	0.000	0.038	0.104
x16	-0.0445	0.015	-3.005	0.003	-0.074	-0.015
x17	-0.0788	0.020	-3.956	0.000	-0.118	-0.040
x18	-0.0780	0.018	-4.425	0.000	-0.113	-0.043
x22	0.0895	0.015	5.932	0.000	0.060	0.119
x23	0.0968	0.013	7.678	0.000	0.072	0.122
x25	0.2175	0.019	11.275	0.000	0.180	0.255
x27	0.9185	0.020	46.042	0.000	0.879	0.958
x29	0.0447	0.014	3.176	0.001	0.017	0.072
x31	-0.2381	0.017	-13.618	0.000	-0.272	-0.204

(Table C. Model accuracy Results on number of Machine learning algorithms with full and reduced feature matrices)

	Accuracy - all features	Accuracy - Backward elimination	Accuracy - PCA
Logistic Regression	78.02%	77.95%	**78.18%**
SVM	**81.64%**	79.65%	81.10%
KNN	85.27%	**85.53%**	85.17%
Decision Tree	86.72%	**88.81%**	83.80%
Random Forest	**88.88%**	88.81%	86.94%
Naïve Bayes	40.79%	49.60%	**52.45%**
Voting classifier	**88.00%**	87.00%	87.00%