HOTEL PREDICTION PROJECT

Project Report

by

G1 - Group 1

Anna Choo Xin Yi

Wesley Djingga

Xie Jianlong

Filbert

Zhang Jieyuan

3rd Apr 2022

[redacted]

1.0 Introduction

Maximizing profit is a known goal for all businesses and having 100% occupancy is an ideal situation for hotel owners to optimize their revenue. However, this goal is unattainable due to last-minute cancellations and no-shows. Therefore, hotels engage in profit maximizing strategy such as overbooking where hotel accepts more customers' booking than the number of rooms available.

2.0 Preliminary Literature Review and Problem Statement

While overbooking is a common revenue management practice to minimize losses from late cancellations and no-shows, it can cause many repercussions customers were turned away due to lack of available rooms.

1. *<u>Compensation</u>*
   Based on a literature review, approximately 56% of the customers expects different sorts of compensation when they are being turned away, e.g. a free night. (Hwang & Wen, 2009)
2. *<u>Bad customer experience</u>*
   When customers perceive being turned away as unfairness towards them, it affects customer loyalty. As customer loyalty encourages commitment to repurchase the odds and costs, bad customer experience renders lack of loyalty, and in turns loss in profit. (Kimes & Wirtz, 2003; Tanford, 2016)
3. *<u>Bad reputation</u>*
   Perceived unfairness triggers the emotional side of humans when they feel anger and disappointed. A research study found that 1 customer is capable to influence an average of 12 people. This leads to a loss in both monetary and reputations in the long run. (Bowen and Shoemaker, 1998)

3.0 Objectives

After literature review, the aim of this project is to predict hotel booking cancellation using machine learning (ML) with features of the customers being taken into consideration during the prediction. It enhances the overbooking strategy as hotels could accurately assign the number of rooms to be overbooked.

4.0 Dataset

This project uses "Hotel Booking Demand Datasets". (Antonio et al., 2019) It consist of data from 2 hotels (H1 and H2) with approximately 119,000 rows and 30 columns. The ML model will be trained by 80% of the data and tested by the remaining 20%.

5.0 EDA

The data is being split into 63% not cancelled and 37% cancelled (Figure 21). Among the features explored, the top 3 market segments contributed to the highest cancellation rate.(Figure 22). The number of special requests made (Figure 23) and returned customers (Figure 24) are found to be related to the cancelling rate too.

6.0 Feature Engineering

Feature engineering is an essential pre-processing step that helps to improve the results of the models. Henceforth, the following feature engineering was being performed after reviewing EDA:

1. *Excluded*: (a) Unmeaningful data (*Figure 25*), (b) Highly correlated data (Figure 26), (c) Data that cause quasi separation issue (Figure 27) and (d) Rows of data that are considered as outliers (Figure 28)
2. *Scaling:* (a) StandardScaler for Logistic Regression, K-Nearest Neighbor and Neural Network, (b) MinMaxScaler for Multinomial Naïve Bayes as input values should be non-negative due to Multinomial Distribution, and (c) Scaling was not required for Random Forest as it is a tree-based model, and Gaussian Naïve Bayes as the model calculates the mean and standard deviation of each feature as well as the distance from the centre of distribution.
3. *One-hot-encoding on categorical features*

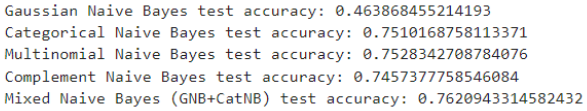
7.0 Models

7.1 Naïve Bayes

Naïve Bayes classifiers are a type of probabilistic classifiers that assume their features to be independent of each other.

Hyperparameter Tuning: (Alpha)
Naïve Bayes performance delta shows insignificant improvement despite hyperparameter tuning. Thus, alpha value is set to default 1 to prevent zero probability.

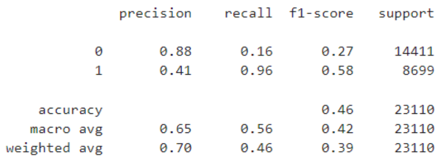
Default Results: As shown in Figure 1: Naive Bayes test accuracy with common, the result of Gaussian Naïve Bayes was less than 50%. Hence, further investigation was done on the classification report.



```
Gaussian Naive Bayes test accuracy: 0.463868455214193
Categorical Naive Bayes test accuracy: 0.7510168758113371
Multinomial Naive Bayes test accuracy: 0.7528342708784076
Complement Naive Bayes test accuracy: 0.7457377758546084
Mixed Naive Bayes (GNB+CatNB) test accuracy: 0.7620943314582432
```

Figure 1: Naive Bayes test accuracy with common feature engineered data

Investigation on Gaussian Naïve Bayes
Based on Figure 2, the model shows to have higher count of false negative. Since Naïve Bayes are generally better with categorical variables, it might be affected by the 14 numerical features in the dataset. Another reason might be caused by the shift in mean due to outliers and one-sided numerical. As distance of the values to the centre of distribution is inaccurate, the model predicts more 1 than 0 when the actual data has more 0 than 1.



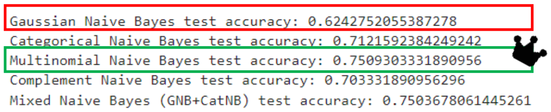
```
              precision    recall  f1-score   support

           0       0.88      0.16      0.27     14411
           1       0.41      0.96      0.58      8699

    accuracy                           0.46     23110
   macro avg       0.65      0.56      0.42     23110
weighted avg       0.70      0.46      0.39     23110
```

Figure 2: Classification report of Gaussian Naive Bayes

Solution
The following additional feature engineering was implemented to push for better results:

1. *Added derivative features* (*Figure 29*)
2. *Dropped the added derivative features that contains the following issues*: (a) Highly correlated features (*Figure 30*) and (b) Aggregated Representative due to extremely low count (*Figure 31*)
3. *Conversion of features to binary value* (*Figure 32*)

Final Results



```
Gaussian Naive Bayes test accuracy: 0.6242752055387278
Categorical Naive Bayes test accuracy: 0.7121592384249242
Multinomial Naive Bayes test accuracy: 0.7509303331890956
Complement Naive Bayes test accuracy: 0.703331890956296
Mixed Naive Bayes (GNB+CatNB) test accuracy: 0.7503678061445261
```

Figure 3: Naive Bayes test accuracy with additional feature engineered data

With the additional feature engineering, the result improved for Gaussian Naïve Bayes (Figure 3: Naive Bayes test accuracy with additional feature engineered data). Although Gaussian Naïve Bayes improved quite significantly, the performance of the other Naïve Bayes models is relatively similar. All in all, it shows a better overall performance as compared to the previous results before the additional feature engineering.
Out of the five models, since Multinomial Naïve Bayes is relatively simpler and perform marginally better, this model is being picked as the best representative for Naïve Bayes model.

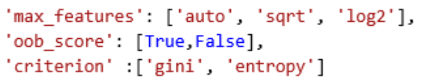
7.2 Random Forest

Random forest is made up of a huge number of individual decision trees that work together as an ensemble.

Default Results

The accuracy result before hyperparameter tuning is 0.87.

Hyperparameter Tuning: max_features, oob_score, criterion
Random forests are known for overfitting data as the depth and complexity of the trees increases. Therefore, a two-step approach was adopted for hyperparameter tuning. First, *GridSearchCV* to identify the best categorical hyperparameter (Figure 4).



```
'max_features': ['auto', 'sqrt', 'log2'],
'oob_score': [True,False],
'criterion' :['gini', 'entropy']
```

Figure 4: Set up for gridsearchcv

Second, optimize other numerical hyperparameters (Figure 5).

The best combination:

1. {'criterion': 'gini', 'max_features': 'auto', 'oob_score': True}
2. Max Depth = 20
3. Min Sample Split = 20
4. Max Terminal Nodes = 40
5. nEstimator = 30
6. Min Sample Leaf = 20

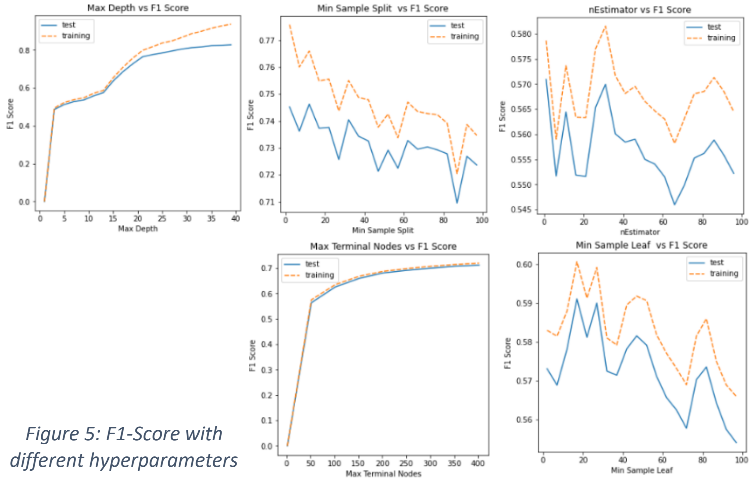


Figure 5: F1-Score with different hyperparameters

Final Result

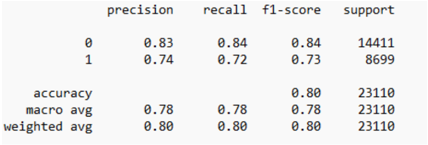
The accuracy result of the pruned random forest is 0.81. Even though the final result after hyperparameter tuning is lower than the default result, the final result is chosen to be the best representation for Random Forest as the default result might be due to overfitting of data.

7.3 K- Nearest Neighbour (KNN)

KNN is an instance-based learning model which does not create any discriminative function from the training data set. It is also non-parametric with no assumptions on the distribution of data.

Default Result

Using the default parameter in sklearn for KNN (n_neighbors =5, weight =“uniform”), KNN predicts the test set with an accuracy of 79.7% with the following classification report (Figure 6).



```
              precision    recall  f1-score   support

           0       0.83      0.84      0.84     14411
           1       0.74      0.72      0.73      8699

    accuracy                           0.80     23110
   macro avg       0.78      0.78      0.78     23110
weighted avg       0.80      0.80      0.80     23110
```

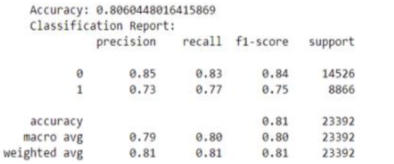
Figure 6: Classification report for KNN before hyperparameter tuning

Down Sampling

The accuracy declined to 76.7% when the a down sampled dataset (number of cancellation equal to number of non-cancellation) was fit into the model.

Hyperparameter Tuning: weight of neighbor, n_neighbors

Only odd numbers are tested for n_neighbors as it is a binary classification model. When weight= “uniform”, the best accuracy and f1 score occurred at n_neighbors = 1 (Figure 7: weight = "uniform", n_neighbors = 1). When weight= “distance”, the f1 score reaches its peak at n_neighbors = 17 (Figure 8).



```
Accuracy: 0.8060448016415869
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.83      0.84     14526
           1       0.73      0.77      0.75      8866

    accuracy                           0.81     23392
   macro avg       0.79      0.80      0.80     23392
weighted avg       0.81      0.81      0.81     23392
```

Figure 7: weight = "uniform", n_neighbors = 1

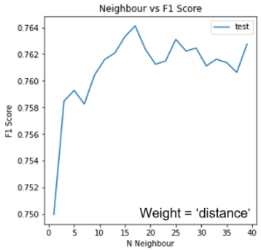


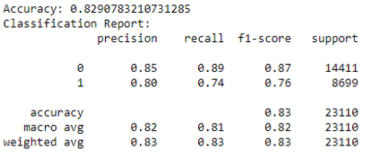
Figure 8: N Neighbor vs F1 Score

The best combination:

1. n_neighbors = 17
2. weights = 'distance'

Final Result

The highest accuracy result of 0.83 was achieved after hyperparameter tuning, hence, the best representative of KNN (Figure 9).



```
Accuracy: 0.8290783210731285
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.89      0.87     14411
           1       0.80      0.74      0.76      8699

    accuracy                           0.83     23110
   macro avg       0.82      0.81      0.82     23110
weighted avg       0.83      0.83      0.83     23110
```

Figure 9: weight = "distance", n_neighbors = 17

7.4 Logistic Regression

Logistic regression uses linear regression as a base model and applies sigmoid function to transform it into a classifier model.

Hyperparameter Tuning: solver, penalty, C

Grid search cross validation was used to look for the best penalty and solver with 5 numbers of k-fold cross validation and accuracy as the scoring criteria (Figure 10: Log Loss Function of Different C). With the best penalty and solver parameters, inverse log loss function of each C value was then plotted to determine best C value (Figure 10: Log Loss Function of Different C).

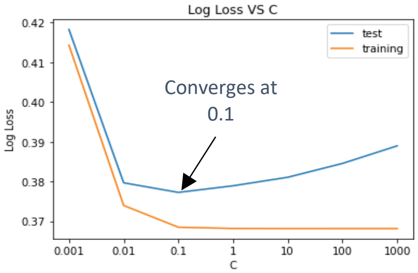
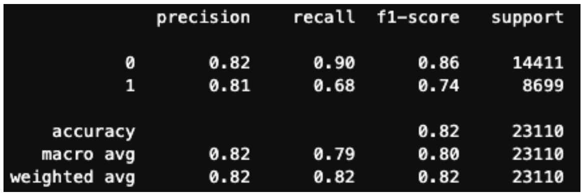


Figure 10: Log Loss Function of Different C

Final Result

The accuracy results 0.82 was achieved after hyperparameter tuning, hence, the best representation of Logistic Regression model.



```
              precision    recall  f1-score   support

           0       0.82      0.90      0.86     14411
           1       0.81      0.68      0.74      8699

    accuracy                           0.82     23110
   macro avg       0.82      0.79      0.80     23110
weighted avg       0.82      0.82      0.82     23110
```

Figure 11: Classification report for Logistic Regression

7.5 Neural Network (NN)

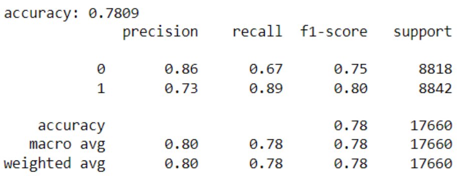
Neural network is a model inspired by network of biological neurons that constitute of nodes and layers.

Hyperparameter Tuning: optimizer, Learning rate, batch size, Epoch

Stochastic gradient descent was the first optimizer tested with the model, however, the accuracy rate was not as high as when adam was used (Figure 12: Final Classification Report for NN).

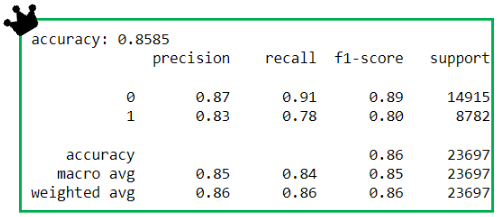
The best combination:

1. Learning rate = 0.01
2. Batch size = 1024
3. Optimizer = Adam
4. Epoch = 100



```
accuracy: 0.7809
              precision    recall  f1-score   support

           0       0.86      0.67      0.75      8818
           1       0.73      0.89      0.80      8842

    accuracy                           0.78     17660
   macro avg       0.80      0.78      0.78     17660
weighted avg       0.80      0.78      0.78     17660
```

Figure 13: Classification report for SGD optimizer



```
accuracy: 0.8585
              precision    recall  f1-score   support

           0       0.87      0.91      0.89     14915
           1       0.83      0.78      0.80      8782

    accuracy                           0.86     23697
   macro avg       0.85      0.84      0.85     23697
weighted avg       0.86      0.86      0.86     23697
```

Figure 12: Final Classification Report for NN

Final Result

The accuracy result of the NN eventually reached 0.86 (Figure 13: Classification report for SGD optimizer) with 3 hidden layers and dropout as regularization to prevent overfitting. Hence, the best representative model for NN.

7.6 Ensemble Learning (Majority Voting)

In an attempt to create a superior machine learning model, ensemble learning model was created based on majority voting of all previous machine learning models trained i.e. Random Forest, K-Nearest Neighbors, Logistic Regression, Multinomial Naïve Bayes and Neural Network.

Default Result

Machine Learning Models	Accuracy	Precision	Recall	F1
Ensemble	0.85	0.88	0.70	0.78

Figure 14: Classification Report for Ensemble Learning Model

Using the ensemble of five machine learning models built, the meta model predicted the test set with an accuracy of 85% and f1 score of 78%. To improve our ensemble model, error correlation between the predictions of each machine learning models was analysed.

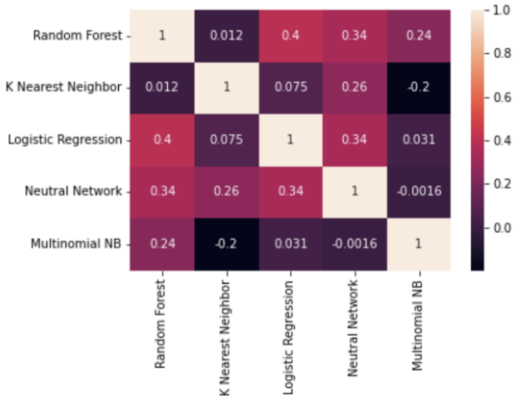
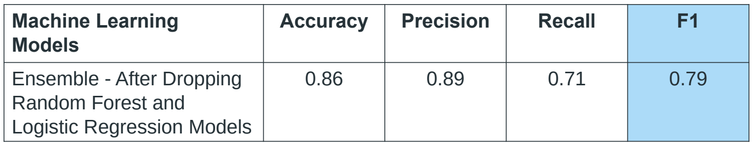


Figure 15 Error Correlation of the Machine Learning Models

As Neural Network is the strongest machine learning model, it is used as the benchmark model. The analysis aimed to point out the machine learning models that have high error correlation with the strongest machine learning model and exclude them from the ensemble model. As a result, whenever Neural Network makes wrong prediction, the other models would not make the same mistake. Hence, it will improve the overall ensemble model performance. As seen from Figure 15 Error Correlation of the Machine Learning Models, Neural Network has higher error correlation with Random Forest and Logistic Regression Models compared to the other models. Thus, these two models were dropped from the ensemble models.

After Dropping Two Higher Error Correlated Models

After dropping two higher error correlated models, the overall performance of the meta model improved by 0.01.



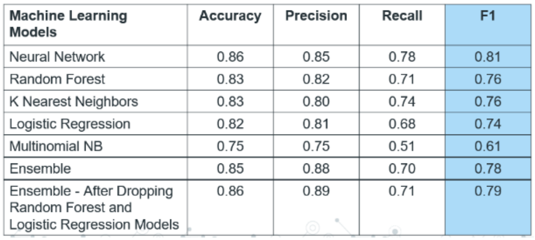
Machine Learning Models	Accuracy	Precision	Recall	F1
Ensemble - After Dropping Random Forest and Logistic Regression Models	0.86	0.89	0.71	0.79

Figure 16: Ensemble Learning Result After Dropping Two Models

8.0 Model Comparison

8.1 Model Results Comparison

Based on the model comparison result in Figure 17: Model Comparison Results, four out of five base models as well as ensemble model resulted in an accuracy of more than 80%. Since both precision and recall are of equal importance, F1 was given the priority because it elegantly sums up the predictive performance. With that, NN has the highest F1 as compared to the rest.



Machine Learning Models	Accuracy	Precision	Recall	F1
Neural Network	0.86	0.85	0.78	0.81
Random Forest	0.83	0.82	0.71	0.76
K Nearest Neighbors	0.83	0.80	0.74	0.76
Logistic Regression	0.82	0.81	0.68	0.74
Multinomial NB	0.75	0.75	0.51	0.61
Ensemble	0.85	0.88	0.70	0.78
Ensemble - After Dropping Random Forest and Logistic Regression Models	0.86	0.89	0.71	0.79

Figure 17: Model Comparison Results

8.2 ROC Comparison

The ROC plots true positive rate (TPR) against false positive rate (FPR) to show the performance of a classification model at all classification thresholds. Figure 18: ROC Curve of Base Model shows the ROC curve of the base five models in this project.

At a glance, the top three ROC curves (Neural Network, Logistic Regression and Random Forest) look similar, however Neural Network performed slightly better as it has higher Area Under Curve (AUC) of 0.938 in comparison to the other models.

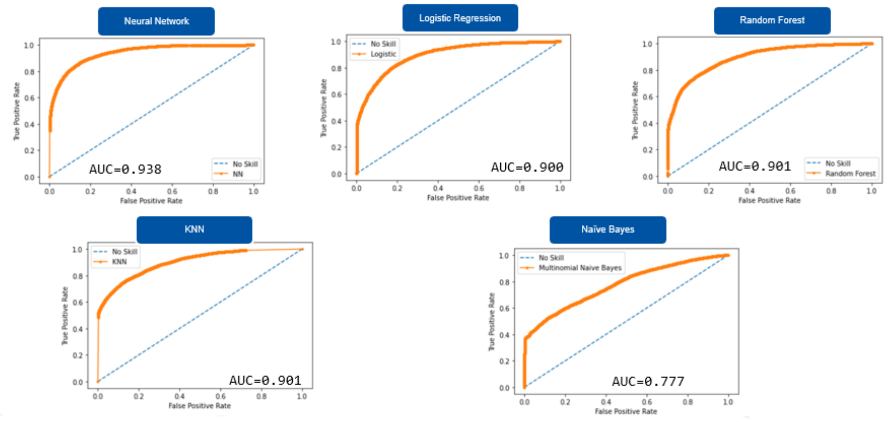


Figure 18: ROC Curve of Base Model

9.0 Feature Importance

Top 10 important features based on Random Forest, Naïve Bayes and Logistic Model.

Top 10 important features for cancellation		
Logistic Regression	**Naïve Bayes**	**Random Forest**
Previous Cancellation	Deposit Type_ Non Refund	Deposit Type_ Non Refund
Deposit Type_ Non Refund	PreviousCancellations	Country_PRT
Country_PRT	AssignedRoomType_L	LeadTime
Agent_9	DaysInWaitingList	TotalSpecialRequest
Assigned Room Type A	MarketSegment_Groups	Deposit Type_ No Deposit
LeadTime	Meal_FB	Agent_9
Agent_240	LeadTime	Previous Cancellation
CustomerType_Transient	ArrivalDateWeekNumber_25	CustomerType_Transient
Hotel_H2	ArrivalDateWeekNumber_18	Booking Changes
MarketSegment_Group	AssignedRoomType_A	Market Segment Direct

Figure 19 Top 10 Feature Importance of Logistic Regression, Naive Bayes and Random Forest

Suggestions:

1. Previous Cancellation: Some of the customers might be price sensitive or like to make comparison with other hotels after booking, hence, they might switch when a better alternative appears.
2. Country_PRT: When the customer is from Portuguese, the customer is likely traveling from other regions of Portugal. When traveling within the same country, the travelling plan may be subjected to more changes given the close proximity.
3. Lead Time: Longer lead time to arrival date also tend to be positively associated with cancellation as plans tend to be subjected to more changes when it is planned way ahead.

4. Agent 9: Hotel can further investigate potential reason for high probability of cancellation for this agent. It might be poor service or other factor that result in Agent 9 being one of the most important features for cancellation.
5. CustomerType_Transient: Transient customers tends to cancel their bookings.

10.0 Implication for Business

The current method to forecast the hotel booking cancellation is based on the historical rate of hotel booking cancellation. This method is deemed unfavorable as the cancelation rate of the hotel booking would change due to seasonality and other factors. The use of machine learning would be a better predictor as the model would also evolve as the hotel booking features evolve.

11.0 Simulation and Demo

To validate the machine learning model, 1000 simulations were done in comparing the hotel booking cancellation results based on historical data versus machine learning. The simulation showed that for every 100 room, the gut feeling (which was based on historical data) showed higher no of overbooked and underbooked rooms as compared to those of machine learning.

For every 100 room	**Gut Feeling**	**Machine Learning**
No of room overbooked	11.21	1.31
No of room underbooked	6.19	5.32

Figure 20 Comparison of Nos of Overbooked and Underbooked Hotel Rooms

It can be seen that by using gut feeling strategy, it resulted in huge errors in the predicted results which would lead to potential revenue loss. On the other hand, the machine learning showed consistently lower error in the predicted results for both overbooked and underbooked rooms which validate our proposed method of using machine learning to predict hotel booking cancellation.

12.0 Conclusion

Hotel overbooking is a common strategy used by hotel manager to achieve the highest possible occupancy rate to maximise the business revenue. The default method to predict the hotel booking cancellation was by using historical data. However, this strategy resulted in many repercussions due to the inaccuracy of hotel booking prediction. Therefore, there is a need for a better method to predict the hotel booking cancellation to maximise the business revenue.

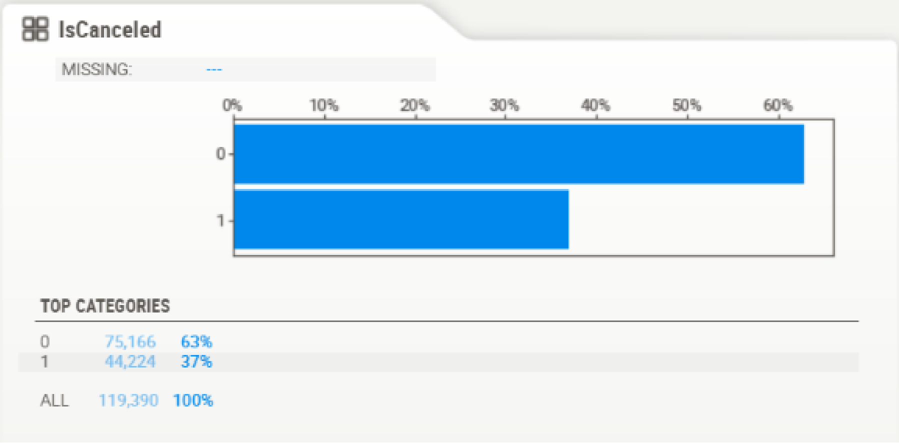
Five different types of machine learning models such as Naïve Bayes, Random Forest, Logistic Regression, K-Nearest Neighbors and Neural Network were trained to predict the hotel booking cancellation. The result showed Neural Network was the best machine learning model to be deployed to predict the hotel booking cancellation.

Ensemble learning model based on majority voting of the five different types of machine learning model trained was created in an attempt to create superior machine learning model. However, it did not perform better as the ensemble model performance was dragged down by the weaker models

1000 simulations were conducted to compare the hotel booking cancellation prediction between historical rate and machine learning model performance. The result showed a consistent lower error in machine learning model prediction. Thus, it was validated that deploying the machine learning model would minimize the nos of hotel room overbooked and it will maximize the hotel booking capacity. Hence, it will improve the customers' experience and improve the hotel profit in the long run.

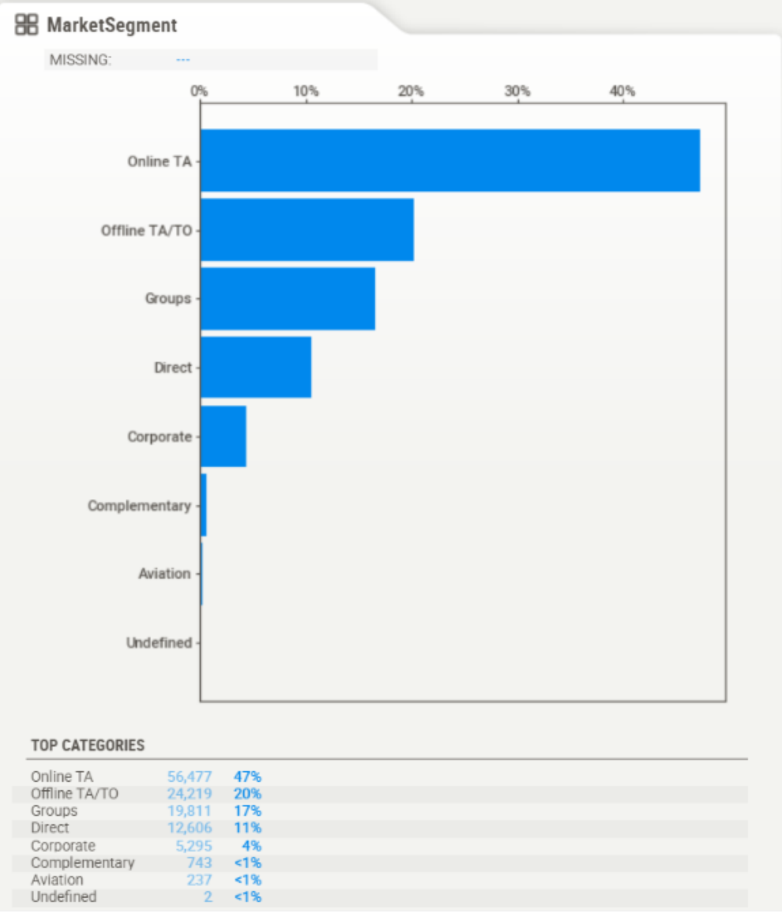
APPENDIX

EDA



TOP CATEGORIES

0	75,166	63%
1	44,224	37%
ALL	119,390	100%



TOP CATEGORIES

Online TA	56,477	47%
Offline TA/TO	24,219	20%
Groups	19,811	17%
Direct	12,606	11%
Corporate	5,295	4%
Complementary	743	<1%
Aviation	237	<1%
Undefined	2	<1%

Figure 21: Market Segment

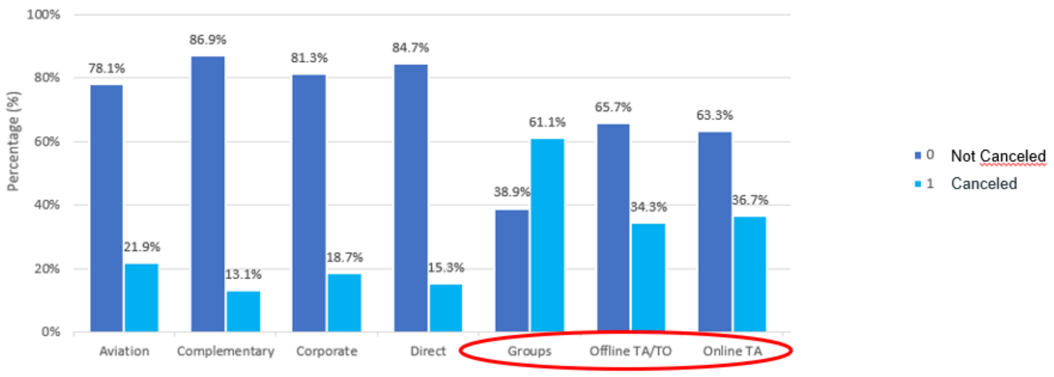


Figure 22: Cancellation rate by Market Segment

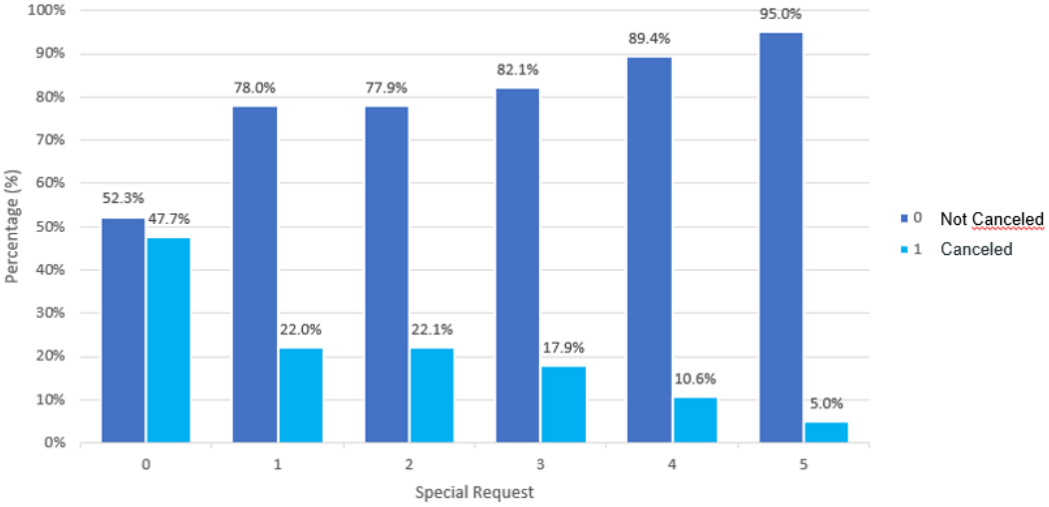


Figure 23: Cancellation Rate by Number of Special Requests

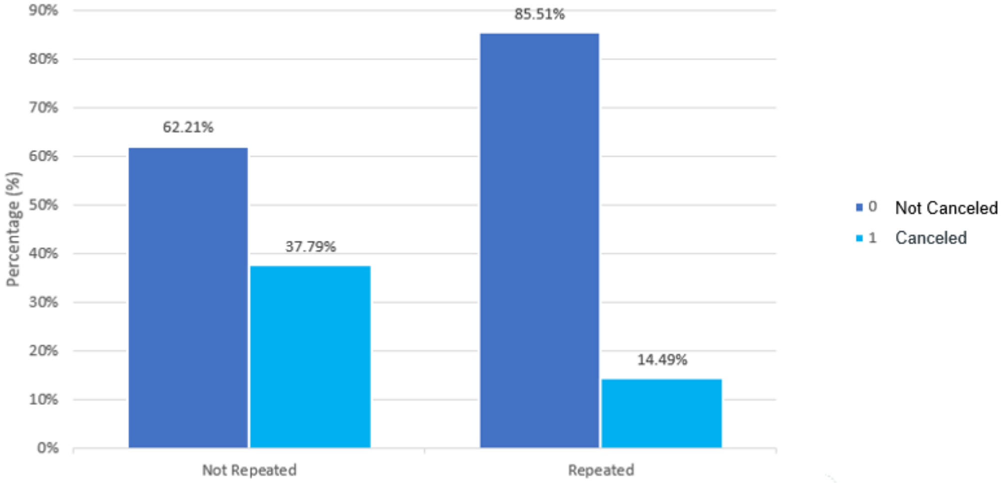
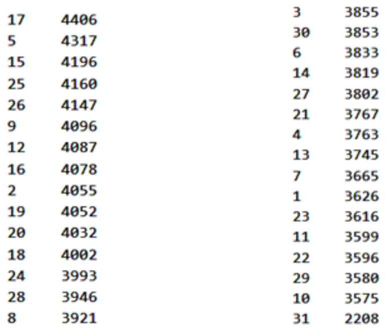


Figure 24: Cancellation Rate by Returned Customers

Feature Engineering

Example of unmeaningful features:

- *E.g. ArrivalDateYear, ArrivalDateDayOfMonth*



```
17      4406
5       4317
15      4196
25      4160
26      4147
9       4096
12      4087
16      4078
2       4055
19      4052
20      4032
18      4002
24      3993
28      3946
8       3921
3       3855
30      3853
6       3833
14      3819
27      3802
21      3767
4       3763
13      3745
7       3665
1       3626
23      3616
11      3599
22      3596
29      3580
10      3575
31      2208
```

Figure 25: ArrivalDateDayOfMonth

Example of highly correlated features:

- *E.g. ArrivalDateMonth*

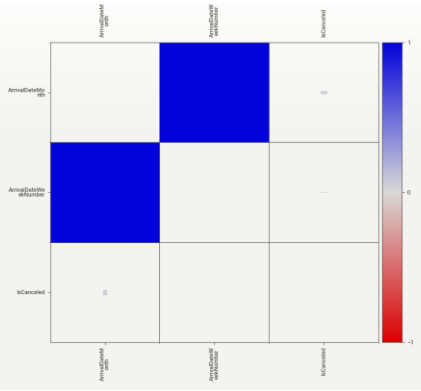
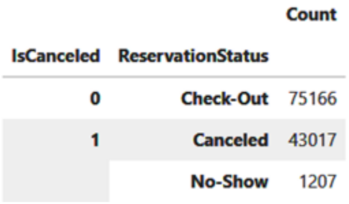


Figure 26: Correlation matrix of ArrivalDateMonth and ArrivalDateWeekNumber

Example of quasi separation issue:

- *E.g. ReservationStatusDate, ReservationStatus, RequiredCarParkingSpaces*



		Count
IsCanceled	**ReservationStatus**	
0	**Check-Out**	75166
1	**Canceled**	43017
	No-Show	1207

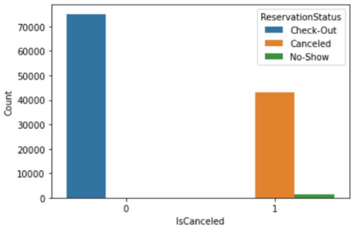
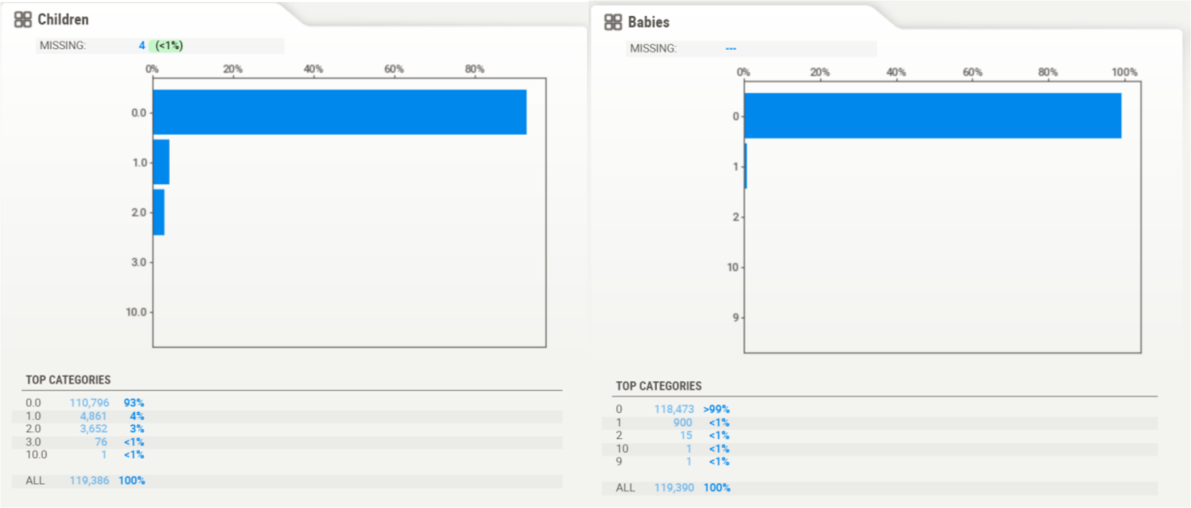


Figure 27: ReservationStatus with value Check-Out is having target (IsCanceled) value as 0 for all rows

Example of outliers:

- *E.g. Average Daily Rate, Adults, StaysInWeekNights, Babies, Children*



TOP CATEGORIES

0.0	110,796	93%
1.0	4,861	4%
2.0	3,652	3%
3.0	76	<1%
10.0	1	<1%
ALL	119,386	100%

TOP CATEGORIES

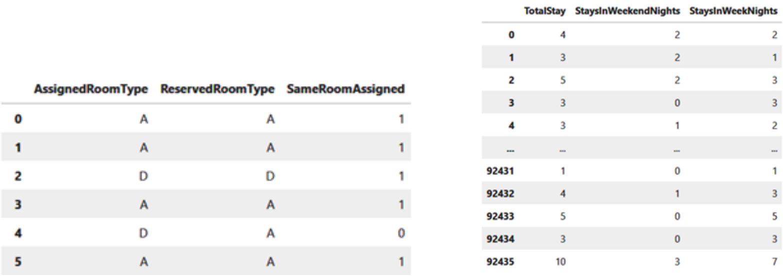
0	118,473	>99%
1	900	<1%
2	15	<1%
10	1	<1%
9	1	<1%
ALL	119,390	100%

Figure 28: Outliers for children (left) and babies (right) features

Model – Naïve Bayes

Example of new additional feature:

- New feature 'SameRoomAssigned' derived from 'ReservedRoomType' and 'AssignedRoomType'



	AssignedRoomType	ReservedRoomType	SameRoomAssigned
0	A	A	1
1	A	A	1
2	D	D	1
3	A	A	1
4	D	A	0
5	A	A	1

	TotalStay	StaysInWeekendNights	StaysInWeekNights
0	4	2	2
1	3	2	1
2	5	2	3
3	3	0	3
4	3	1	2
...	...	...	...
92431	1	0	1
92432	4	1	3
92433	5	0	5
92434	3	0	3
92435	10	3	7

Figure 29: Example of new feature such as AssignedRoomType (left) and TotalStay (right)

Example of highly correlated feature:

- New feature 'TotalStay' derived from 'StayInWeekNights' and 'StayInWeekendNights'

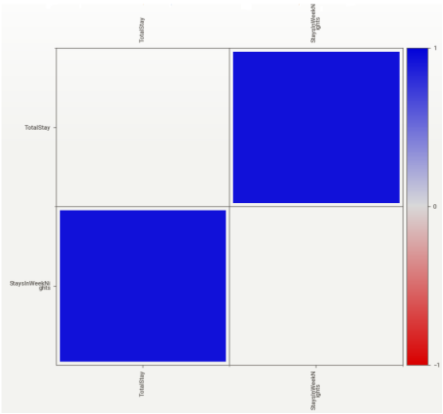
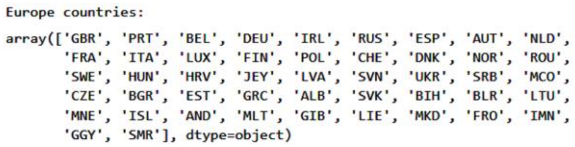


Figure 30: Correlation matrix of StaysInWeekNights and TotalStay

Example of aggregated representative:

- New feature ‘Region’ derived from ‘Country’ by referring to ISO-3166



```
Europe countries:
array(['GBR', 'PRT', 'BEL', 'DEU', 'IRL', 'RUS', 'ESP', 'AUT', 'NLD',
       'FRA', 'ITA', 'LUX', 'FIN', 'POL', 'CHE', 'DNK', 'NOR', 'ROU',
       'SWE', 'HUN', 'HRV', 'JEY', 'LVA', 'SVN', 'UKR', 'SRB', 'MCO',
       'CZE', 'BGR', 'EST', 'GRC', 'ALB', 'SVK', 'BIH', 'BLR', 'LTU',
       'MNE', 'ISL', 'AND', 'MLT', 'GIB', 'LIE', 'MKD', 'FRO', 'IMN',
       'GGY', 'SMR'], dtype=object)
```

Figure 31: Countries represented by Europe region

Example of binary conversion of a feature:

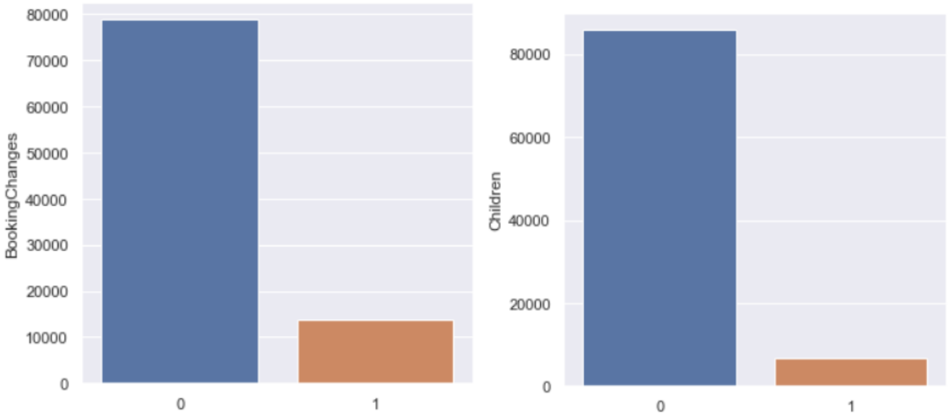


Figure 32: Features which are converted to binary such as BookingChanges (left) and Children (right)

References

All factual material that is not original must be accompanied by a reference to its source. Please use the APA citation style. There are also citation management tools such as EndNote or Zotero, etc. Please contact SMU Libraries (library@smu.edu.sg) if you need assistance with citation management tools.

Antonio, N., de Almeida, A., & Nunes, L. (2019). Hotel booking demand datasets. *Data in Brief*, *22*, 41-49. https://doi.org/https://doi.org/10.1016/j.dib.2018.11.126

Hwang, J., & Wen, L. (2009). The effect of perceived fairness toward hotel overbooking and compensation practices on customer loyalty. *International Journal of Contemporary Hospitality Management*, *21*(6), 659-675. https://doi.org/10.1108/09596110910975945

Kimes, S. E., & Wirtz, J. (2003). Has Revenue Management become Acceptable?: Findings from an International Study on the Perceived Fairness of Rate Fences. *Journal of Service Research*, *6*(2), 125-135. https://doi.org/10.1177/1094670503257038

Tanford, S. (2016). Antecedents and Outcomes of Hospitality Loyalty: A Meta-Analysis. *Cornell Hospitality Quarterly*, *57*(2), 122-137. https://doi.org/10.1177/1938965516640121