Hotel Booking Optimization

CHEME6880 / SYSEN5880 Project Report

Zhiming Xie (zx284) , Jaeha Kim(jk2894)

Cornell University

Abstract

Hotel booking process is the key business component in the hospitality industry. However, there are some cases in which customers cancel their bookings. This directly leads to the hotel's profit loss. To prevent this profit loss, it is important to get the proper amount of overbookings for booking cancellation based on precise cancellation prediction. If hotels could make an accurate prediction of which customers are going to cancel their bookings and get overbookings, they can maximize their profits. For this goal, in this project we developed a prediction engine that can predict a booking cancellation using “the hotel booking demand datasets”. This dataset collected 119,390 booking data with 31 variables including 14 categorical variables. We compared six different machine learning models and tested dimension reduction using Principal Component Analysis (PCA). As a result, we developed an XGBoost which can predict a hotel booking cancellation with 0.87 test accuracy.

1. Introduction

1.1 Importance

Hotel booking is ubiquitous in our daily life. With the dropping frequencies of Covid-19 lockdown, local governments encourage more people to travel so as to boost the local hospitality industry. According to TourismReview, hotel reservations were currently on an upward trend with a 46% global occupancy rate in April 2021 []. This was a significant increase compared to a low of 13% in the same month in 2020. Leave alone that the occupancy rate in January 2022 had increased up to 47.8%. With the increase in the number of hotel bookings, our team's common goals are to find out the optimal prediction models to address the following problem.

1.2 Application / Problem

Predicting the possibility of each booking cancellation is a major risk management issue in the hospitality industry. In the hotel booking process, the most important goal for hotels is minimizing cancellation rates. No-show and cancellation is related to the hotels' profit and to minimize this, hotels get some portion of extra bookings. However, it is difficult to decide how many extra bookings they need and it is directly related to predicting the cancellation rate. Moreover, it is also important to decrease the number of cancellations. Therefore the problem we addressed on this project is developing a prediction model for the hotel booking cancellation based on "the hotel booking demand datasets" []. This problem is also related to our question of which model will best fit a binary classification problem using the high-dimensional dataset including multiple categorical variables.

1.3 Hypothesis

The hotel booking dataset is made up of 119,390 booking data with 31 variables. Among the 31 variables 14 variables are using categorical values. Our problem is to predict the “Is Canceled” output which is a binary classification problem (“is canceled”: 1, “or not”: 0). There are two major issues in our dataset. First, the dataset is high-dimensional. Second, 50% of the original features are categorical values with two to twelve unique values. With these issues, our hypothesis is that the dimension reduction process will contribute to a prediction model performance improvement. In addition, there are some missing and invalid values. Moreover, based on the interaction scatter plot results of every two variables, we concluded that there is not a significant outlier in our dataset but there are some noises.

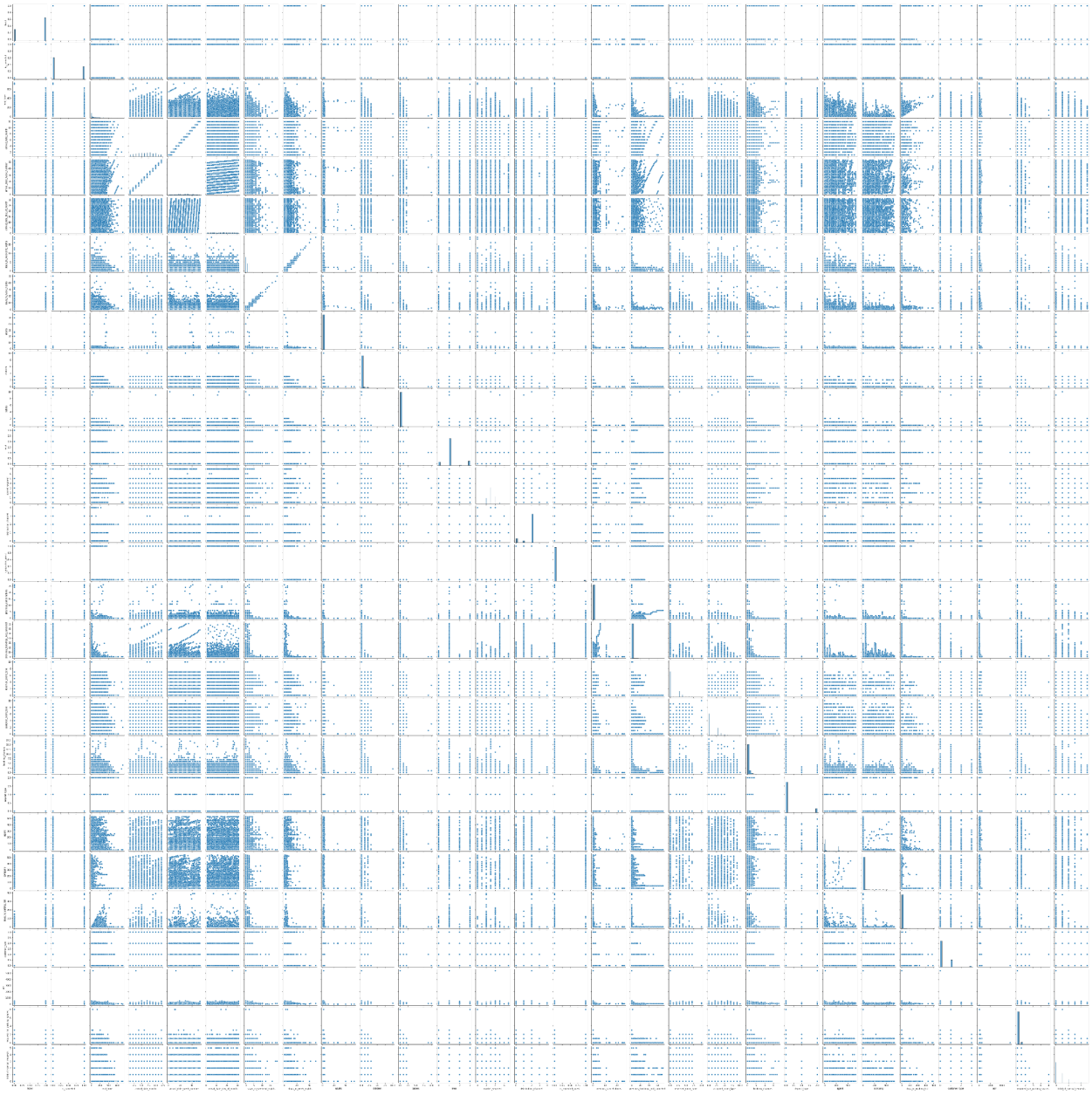

Figure 1. Interaction results with the pairs plots

2.Background of Hotel Booking Demand Dataset

2.1 Problem Description

Predicting booking cancellation is vital to the hospitality industry. It is an important factor in allocating human and financial resources to revive the local economy in the aftermath of the global pandemic. Our team tested how accurate prediction can be enabled by using the customer profiles, booking timeline, and travel type datasets. There is a lot of data produced in the booking process such as booking database, transaction data, and booking change log. However, the databases are usually segmented by multiple stakeholders. Nevertheless, in 2019, Nuno Antonio, Ana de Almeida, and Luis Nunes found this problem and integrated these segmented databases into the "Hotel Booking Demand Dataset" [].

2.2 Dataset

This dataset describes 40,060 booking observations on resort hotels and 79,330 booking observations on city hotels with 31 variables. The data is collected on bookings arrived between the July of 2015 and August 2017 []. Since the data was collected before Covid-19 outbreak, it didn't include Covid-19 related variables. There are 14 categorical variables among 31 variables. More detailed descriptions on the variables are presented in Table 1. The 31 variables can be grouped into guest information, booking requirement, room type, booking update, guest previous record, booking timeline, market distribution, and cancellation. With this dataset, we developed a prediction model for cancellation using the rest variable groups.

Table 1. Overview of 31 dimensions in the hotel booking demand dataset

Variable (Type)	Description	Variable (Type)	Description
ADR (Numeric)	Average Daily Rate (Price)	**Adults (Integer)**	Number of adults
Babies (integer)	Number of babies	**Children (Integer)**	Number of children
Country (Categorical)	Country of Origin	**CustomerType (Categorical)**	Contract / Group / Transient (not contract and not group) / Transient-Party (associated with other transient booking)
Meal (Categorical)	Undefined/SC: No meal package BB: Bed & Breakfast HB: Half board (breakfast and one other meal) FB: Full board (breakfast, lunch, and dinner)	**Required Car Parking Spaces (Integer)**	Number of car parking spaces required by the customer
Reserved Room Type (Categorical)	Code of room type reserved	**Assigned Room Type (Categorical)**	Code for the type of room assigned
Total Of Special Requests (Integer)	Number of special requests made by the customer (twin bed or high floor)	**Booking Changes (Integer)**	Number of changes/ amendments made to the booking from the moment of check-in or cancellation
Is Repeated Guest (Categorical)	Repeated guest (1) or not (0)		
Previous Cancellations (Integer)	Number of previous bookings that were canceled by the customer	**PreviousBookings_Not Canceled (integer)**	Number of previous bookings not canceled by the customer
Stays In Weekend Nights (Integer)	Number of weekend nights the guest stayed or booked	**Stays In Week Nights (Integer)**	Number of week nights the guest stayed or booked
Arrival Date Day Of Month (Integer)	Day of the Month of the arrival date	**Arrival Date Month (Categorical)**	Month of arrival date with 12 categories (January to December)
Arrival Date Year (Integer)	Year of arrival date	**Arrival Date Week Number (integer)**	Week number of the arrival date
Lead Time (Integer)	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date	**Days In Waiting List (Integer)**	Number of days the booking was in the waiting list before it was confirmed to the customer
Reservation Status Date (Date)	Date at which the last status was set.		
Agent (Categorical)	ID of the travel agency that made the booking	**Company (Categorical)**	ID of the company/entity that made the booking or responsible for paying the booking
Distribution Channel (Categorical)	Booking distribution channel (TA: travel agents, TO: Tour Operators)	**Market Segment (Categorical)**	Market segment destination (TA: travel agents, TO: Tour Operators)
Deposit Type (Categorical)	No Deposit Non Refund Refundable		

Reservation Status (Categorical)	Canceled Check-out No-show	**IsCanceled (Categorical)**	Canceled(1) Or not (0)

3.Method

3.1 Feature Engineering

- *Drop invalid data.* In the common hotel booking practices, hotels only allow customers who are over 18 years old to check in or to make reservations in person []. Therefore, we find that if there are cases where the adult number is none or zero, the baby and child number are non-zero. Another invalid data case could be that the adult number is zero, and booking still exists. Both cases are invalid according to the hotel booking practices and we remove them from the dataset.

- *Remove overlapping and irrelevant features.* The features that overlap with other features should be removed from the dataset to reduce feature dimensions. For example, 'reservation_status' overlaps with 'is_cancelled', in our project, we only consider the cancellation status. Thus we drop the 'reservation_status' column. Other features such as 'country'. 'arrival_date_year' and 'reservation_status_date' are not relevant to the topic we discuss, hotel booking cancellation. As a result, we removed those columns as well.

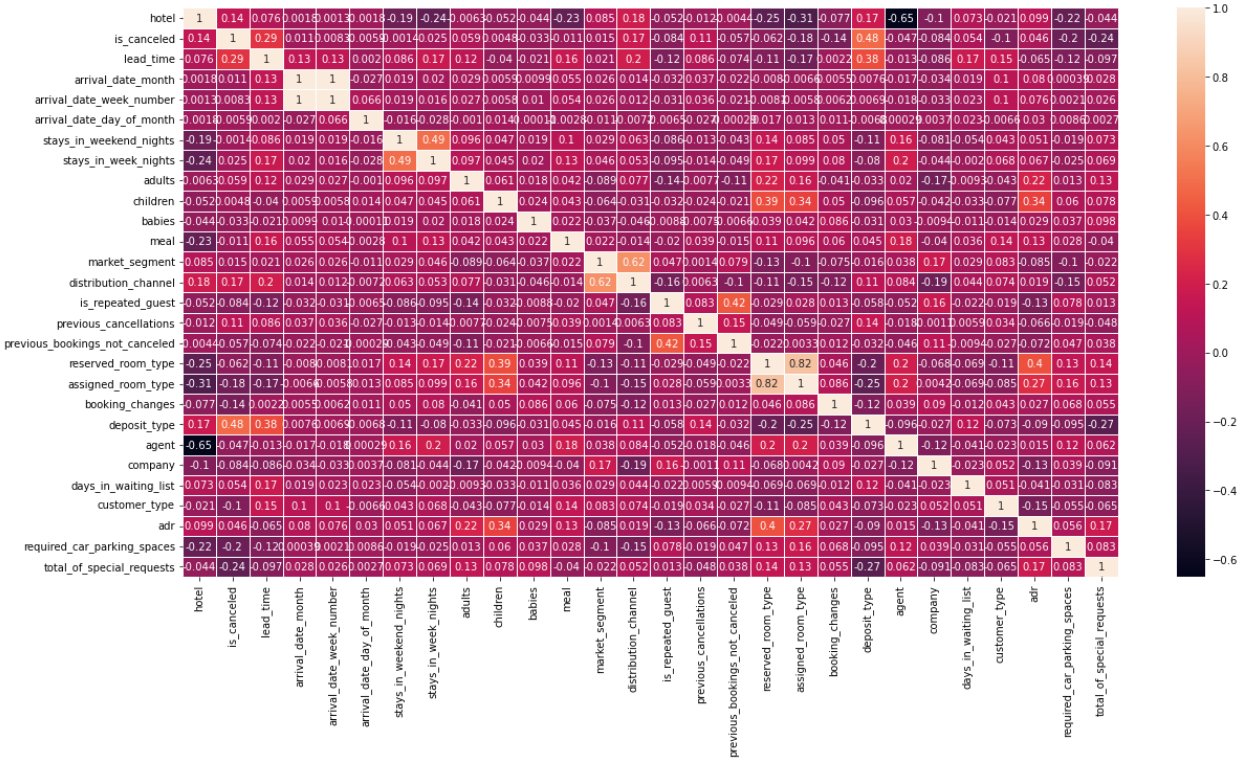

Figure 2. Correlation analysis result

- *Convert categorical features to integers* Categorical features are also crucial in model fitting and training. To utilize those features, we need to convert them into numerical features which serves as a stepping stone for Principal Component Analysis. There are various ways to encode categorical variables. One way is one-hot encoding. But this method increases the dataset dimension dramatically, we converted the string categorical value into ordinal integers. The table below demonstrates all the categorical values that are converted into their corresponding numerical values.

- Replace Null values with default value, 0. There are 16,340 null values in [‘agent’], 112,593 null values in [‘company’], 4 null values in [children’]. Since those features have categorical values, we choose to replace Null with 0 instead of mean values.

- *Principal Component Analysis (PCA)* PCA is implemented in this project to reduce the 26-dimension of the dataset to 6-dimension. Referring to the graph below, these 6 Principal Components cover 99.5% variance of the dataset. The reason for this dimension reduction process is to improve model training performance at a low cost of its accuracy. Less computational power is required to compute a 6-dimension dataset than 26-dimension dataset.

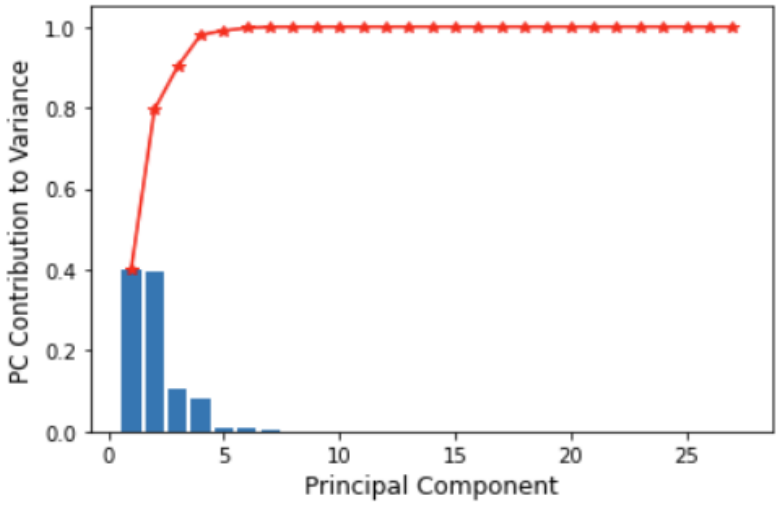

Figure 3. Principal component analysis result

3.2 Data Splitting

The dataset is split into train and test datasets. We allocate 80% of the dataset for the training data and 20% for the testing data. The data split is done on both pre-processed data without principal component analysis and with principal component analysis. The reason for generating

two sets of data is to compare the accuracy of each machine learning model trained on both PCA applied methods and non-PCA applied methods.

3.3 ML Models Training

There are a total of six machine learning models we choose to train on our dataset. These models are all suitable for classification problems. The dataset we obtain from Kaggle consists of 31 variables, and 26 of the variables will be X input features and 1 variable, ‘ is_Canceled’ will be used for the label. In our project, we use a cross validation function - RandomSearchCV to perform hyper-parameter tuning.

Logistic Regression

In this problem, the label of the y values is 1 (canceled) or 0 (not canceled). For this binary classification problem, we first choose the basic classification model, logistic regression. It is a supervised learning method and supports categorizing data into discrete classes, analyzing independent and dependent variables’ relationships. We choose this model based on the advantages of simplicity of implementation. It can be used on high-dimension dataset, although Logistic Regression does not work well on non-linear problems.

Decision Tree

Decision Tree generates a flowchart like tree classification. It represents a tree diagram which helps us to visualize the significance of each feature. Splitting the dataset into subsets based upon the entropy values or the Gini impurity in a recursive manner until entropy does not improve further. Based on the understanding of Decision Tree’s weakness that it is prone to errors with high dimensional dataset, we assumed that Decision Tree model on non-PCA data will have 10%- 15% lower accuracy than that on PCA data.

Random Forest

Random Forest consists of multiple decision trees; its resultant variance is low since the output is based on multiple decision trees rather than one decision tree. Moreover, it repeatedly draws random samples from the dataset with replacement and forms various decision trees. This technique avoids overfitting and high variance. Therefore, Random Forest works well on our 26-dimension, categorical, non-linear dataset.

Ada Boost

Adaboost is an ensemble learning algorithm. However, it is made up of weak learning algorithms. The training data used for training decision stumps contains few data samples assigned higher weights owing to misclassification of those data set in the previous decision stump. Misclassified data samples with higher weight results will get sampled in the new data sample. This iterative ensemble method builds a stronger classifier by combining multiple poor-performing classifiers. In our case, there may exist several weaker learners such as “Baby”, “Breakfast”, and etc. Adaboost comes in handy by reducing the loss function, but Adaboost is sensitive to outliers as it puts more weight on misclassified data which will result in overfitting.

Gradient Boosting

Similar to Adaboost, it combines weak learning algorithms by fitting into the pseudo-residuals of the previous predictor. It is an additive model; the residuals are captured in a tree-by-tree manner which also covers the maximum variance of the data set. A learning rate is introduced to this algorithm to identify negative gradients and move in the opposite direction to minimize the loss. Gradient Boosting is more flexible than Adaboost, it can optimize more on different loss functions than Adaboost does.

XgBoost

XgBoost is a more regularized version of Gradient Boosting. It implements L1 and L2 regularization, which boosts up the model generalization abilities, and offers higher speed computation than Gradient Boost.

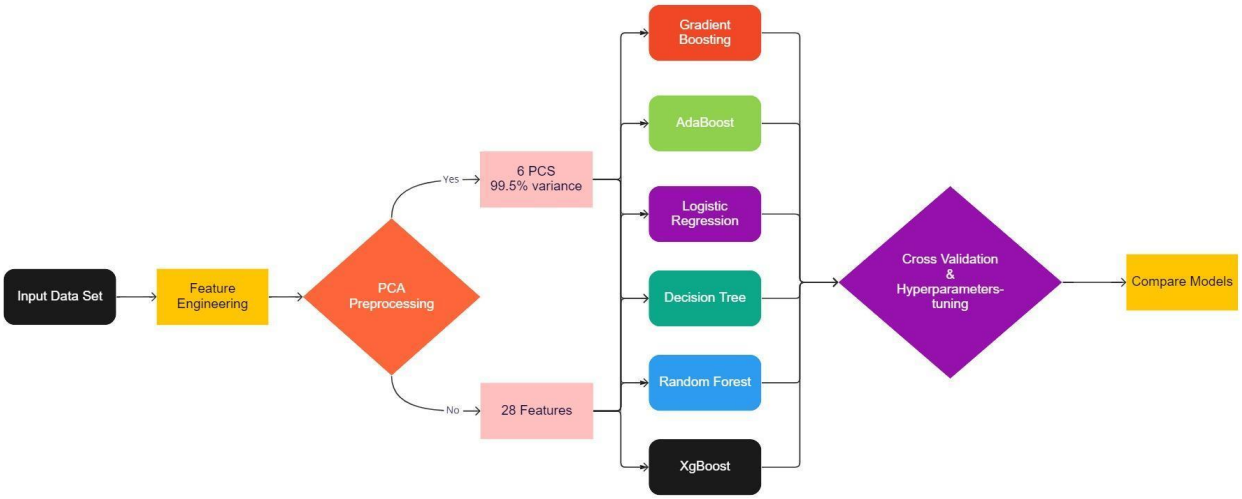

Figure 4. Model development flow chart

4. Results & Discussion

Six machine learning models are used to train the dataset with PCA (6-dimension) and without PCA (26-dimension) and we compared these twelve different models. Six models we used are *Logistic Regression, Decision Tree*, *Random Forest, Ada Boost*, *Gradient Boosting*, and XgBoost.

For hyperparameter tuning, we used the cross validation (RandomSearch CV) to increase a model's accuracy. As the outcome, we get a 'liblinear' solver, L1 penalty, 5000 maximum_iteration, 1.6237 for inverse of regularization strength (c) for the Logistic Regression

model with 0.7980 score without PCA implementation. With PCA implementation, we get the ‘liblinear’ solver, L1 penalty, 100 maximum_iteration, 11.2883 for c for the Logistic Regression model with 0.6675 score.

In the Decision Tree model, we get Gini impurity function for measuring a split quality, None for maximum depth, 8 maximum_features, and 6 minimum samples_leaf. This model gives 0.8266 score without PCA implementation. With PCA implementation, we get the Gini impurity function for measuring a split quality, None for maximum depth, 5 maximum_features, and 2 minimum samples_leafmodel with 0.7733 score.

In the Random Forest model, we get 19 trees (n_estimators), 0.0141 for minimum number of samples for a leaf node, and 0.3317 for maximum number of features. This model gives 0.8239 score without PCA implementation. With PCA implementation, we get 144 trees, 0.0176 for minimum number of samples for a leaf node, and 0.2692 for maximum number of features with 0.7211 score.

In the Adaboost model, we get 200 maximum number of estimators for termination (n_estimator), and 0.1 learning rate. This model gives 0.8142 score without PCA implementation. With PCA implementation, we also get 200 maximum number of estimators for termination, and 0.1 learning rate with 0.6764 score.

In the Gradient Boosting model, we get 0.7142 learning rate, deviance loss function, 5 max depth, 170 boosting stages (n_estimators), and 0.5523 for subsampling fraction for fitting each base learner (subsample). This model gives 0.8388 score without PCA implementation. With PCA implementation, we get 0.1584 learning rate, deviance loss function, 6 max depth, 765 boosting stages, and 0.6702 for subsample with 0.7899 score.

In the XGBoost model, we get 7 minimum weight for a child (min_child_weight), 15 maximum depth, 0.3 learning rate, 0.1 minimum loss reduction(gamma), and 0.7 for tree subsampling ratio (colsample_bytree). This model gives 0.8625 score without PCA implementation. With PCA implementation, we get 1 minimum weight for a child, 12 maximum depth, 0.3 learning rate, 0.4 minimum loss reduction, and 0.5 for tree subsampling ratio with 0.7949 score.

According to the comparison of these 12 hyperparameter tuned models, XgBoost model obtains the highest accuracy (0.8625) with PCA and without PCA as the chart below shows. To our expectation, XgBoost performs the best owing to its two advantageous characteristics: L1&L2 regularization and parallelized tree building.

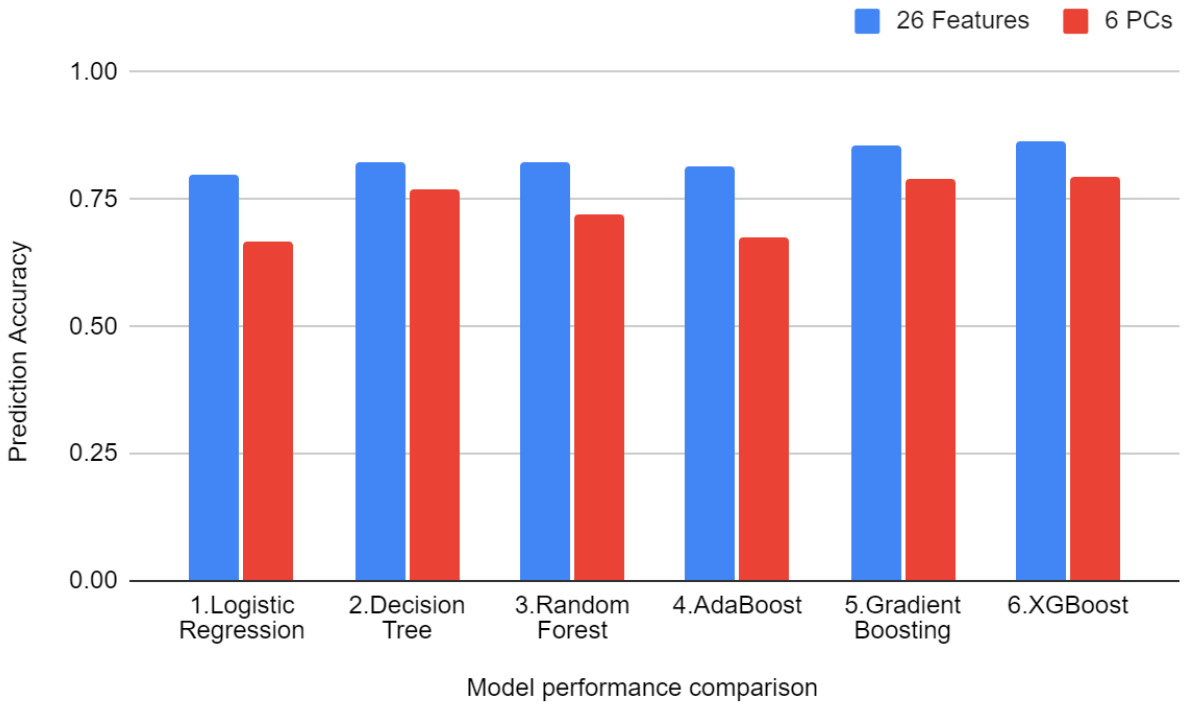

Figure 5. Prediction model accuracy comparison on 26-dimensional dataset and PCA applied 6-dimensional dataset

On the other hand, we assumed that the dimension reduction method would provide a better performing model in our classification problem. However, in all six cases, non-PCA models perform better than PCA models. The reason might be that most of the variables in the hotel booking demand datasets are converted from categorical values. This study shows an example that dimension reduction is not appropriate for categorical variables. Furthermore, the encoding method of the categorical values to integers might be not appropriate in our variables. The data type conversion could be done with one-hot encoding which we avoided for the curse of dimensionality.

5. Conclusion

Using the hotel booking demand datasets that collected 119,390 booking observations with 31 variables, we develop a booking cancellation prediction model. In the model development process, six machine learning models are selected for this binary classification problem. There are two major comparisons in the model selection. First we compare six different models and we compare PCA applied models with the original models. It generates twelve different models which tuned hyperparameters with 5-fold cross validation. Among six models, XGBoost outperforms the other five models. When we compare PCA applied models with others, models which do not apply PCA perform better with higher accuracy. The XGBoost without PCA application works best with 0.8625 prediction accuracy on the test data.

6. Recommendations

From the discussion above, we find out that PCA could not perform better on a dataset with categorical variables. Moreover, these categorical values can be converted with one-hot encoding if they are not ordinal values. We also recommend dropping [‘days_ in_waiting_list’,

'assigned_room_type', 'arrival_date_day_of_month'] because these variables have low correlation values with the label according to Figure 2.

In addition, Artificial Neural Networks can be tested using our dataset with two hidden layers. After the final presentation, we additionally trained ANN. We use the ReLu activation function for two hidden layers because of its reduced likelihood of the gradient to vanish. The output layer we recommend to use the sigmoid function since there are two class labels in output, 0 or 1. Sigmoid will be a good fit since its range is from 0 to 1 in a reversed S shape. Figure 6 shows the training accuracy and the cross validation accuracy from the trained ANN model. ANN performs extremely well on the dataset with only 40 epochs. The training loss drops below 0.05 while the training and cross validation accuracy increases above 0.9905. At last, other advanced or faster machine learning models such as LightGBM could have been used.

-

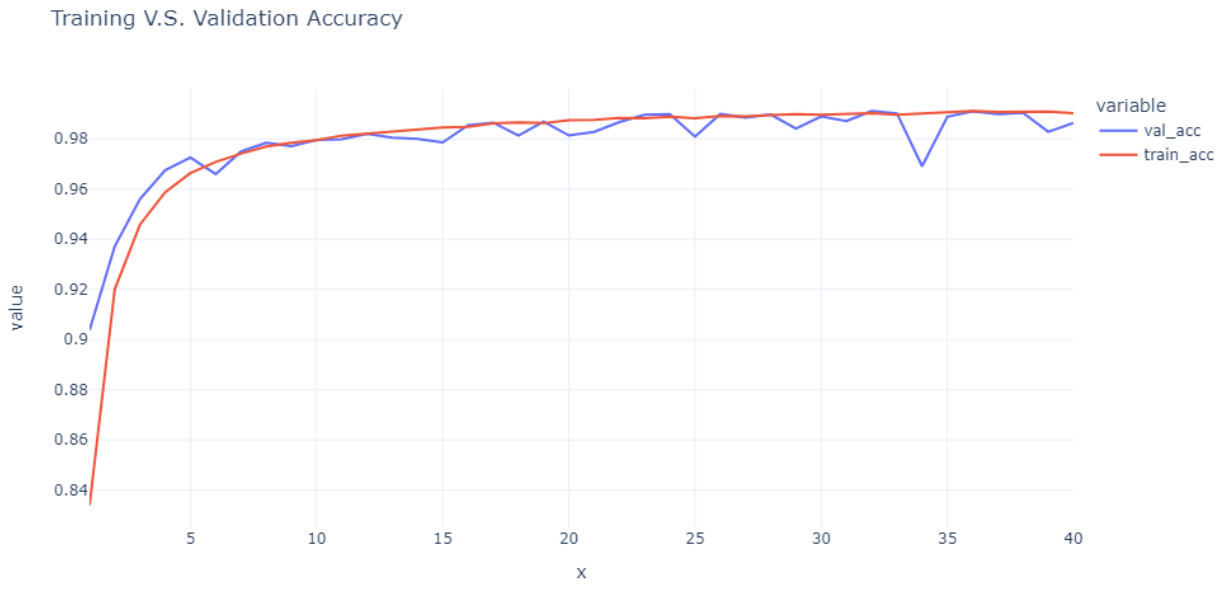

Figure 6. Training v.s. validation accuracy of ANN model with different epochs

7.References

Europe Hotel Satisfaction Score

: https://www.kaggle.com/datasets/ishansingh88/europe-hotel-satisfaction-score

Reference Project

: https://www.kaggle.com/code/steinerhslu/europe-hotel-review-gioele

Hotel Booking Prediction

: https://www.kaggle.com/code/niteshyadav3103/hotel-booking-prediction-99-5-acc

Dataset Source, Feature Table

: Antonio, Nuno, Ana de Almeida, and Luis Nunes. 2019. "Hotel Booking Demand Datasets." *Data in Brief* 22 (February): 41–49. https://doi.org/10.1016/j.dib.2018.11.126.