EDA

- Is_repeated_guest: there are no missing values in this variable. Most of the bookings are not repeat guests. Only some of them are repeated customers. I assume that repeated customers are less likely to cancel based on the initial guessing of business.
- previous_cancellations: There are some cases where repeated customers have never canceled a booking reservation. Values = 1/2/3 are reasonable numbers because it is possible for the same customer to cancel multiple times. However 15+ canceled histories may come from agents or group orders. For example, there are 26 customers that canceled 26 times before.
- previous_bookings_not_canceled: 115770 booking history has 0, which means that no customer profiles associated with this booking. It may also mean that a repeated customer cancel all of booking before
- reserved_room_type: Most of the reserved room belongs to type A. D is the second popular room types
- assigned_room_type: Most of the assigned room is A room. There are 14917 assigned rooms that are not the same as the reserved room. We assume that those who are not getting the same will be more likely to cancel the booking
- Booking_changes: The maximum number of booking changes is 21. The average value of booking_changes is 0.22.
- deposit_type: There are three types of deposit method: no deposit, non-refund, refundable. Most of the deposit method is no deposit. By checking their distribution, we found that it is more likely to cancel if a customer chooses a non-refund method, which is an interesting phenomenon.
- agent: There are a lot of missing values in the agent variable. Based on data description, if a booking “Agent” is defined as “NULL”, it means that the booking did not come from a travel agent. In other words, 16340 bookings are not from the agent.

Feature engineering

- Is_repeated_guest: we converted this variable to categorical variables
- booking_changes: We converted this ordinal categorical variable to the numerical variable.
- deposit_type: We converted this variable to categorical variables
- previous_bookings: Instead of having both bookings canceled and not canceled, we take the sum of the two. In other words, previous_bookings = previous cancellation + previous not cancellation.
- got_room_booked: In order to reduce the unnecessary dimensionality, we combine assigned_room_type and reserved_room_type together. In other words, we create a dummy variable that displays 1=customers got the booked room and 0=customers got a different room.
- agent_new: the agent variable contains a lot of missing values. Based on data description, missing values means that this booking is not from the agent. Therefore, we replace the null value with no-agent. However, we did not use this variable because it has too many levels that are more computationally intensive.

- booked_by_agent: In order to reduce the level of categorical variables containing the key information, we create a dummy variable. Yes means that this booking is from an agent.
- agent_risk: we have created a new feature called “agent_risk” which is calculated by the historical rate of cancellations for the agency the booking came through. After careful consideration, we did not use this variable because this variable uses the information from the target variable.

Modeling

- Naive Bayes Classifier:
    - Naive Bayes Classifier is our benchmark model. We used Gaussian Naive Bayes algorithm, which is a probabilistic machine learning algorithm used for classification tasks. It is a variant of the Naive Bayes algorithm that assumes that the features of the dataset are normally distributed. In general, Gaussian Naive Bayes algorithm is simple, fast, and works well for high-dimensional datasets. However, it may not perform well when the independence assumption does not hold or when the features are not normally distributed.
    - For our problem, we conduct grid searches in 10-fold cross validation. By optimizing the hyperparameter in cross validation, our best model is a model with 0.01 var_smoothing. The best cross-validation AUC score is 0.83. After retraining the model with a full train dataset, we predict the unused test dataset. The misclassification rate in unused test dataset is about 0.37. Based on the f1-score on the unused dataset, we got 0.63. The AUC in unused test dataset is only 0.69
- Decision Tree Classifier:
    - A Decision Tree Classifier is a type of machine learning algorithm that is commonly used for classification tasks. It works by recursively splitting the data into subsets based on the values of input features, until a leaf node is reached, which represents the predicted class label for that particular input. The reason why we choose this model is that decision trees are popular for their simplicity, interpretability, and ability to handle both categorical and numerical data. They can also handle missing values and outliers, and are able to capture complex relationships between features.
    - For our problem, we conduct grid searches in 10-fold cross validation. By optimizing the hyperparameter in cross validation, our best model is a model with ccp_alpha=0.001, criterion='entropy', max_depth= 15, max_features=None. The best cross-validation AUC score is 0.92, which is higher than the benchmark mode. After retraining the model with a full train dataset, we predict the unused test dataset. The misclassification rate in unused test dataset is about 0.16, which is better than the benchmark model. Based on the f1-score on unused dataset, we got 0.84, which is a decent value. More importantly, the AUC in unused test dataset is about 0.8190, which overperforms the benchmark model
- XGBoost Classifier:
    - XGBoost Classifier is an ensemble learning algorithm that uses multiple decision trees to make predictions. XGBoost is a variant of gradient boosting. XGBoost

improves on gradient boosting by the following aspects (1) adding regularization to prevent overfitting, (2) adding parallel processing to speed up the training process, (3) providing built-in methods to handle missing values, (4) measuring feature importance.

- For our problem, we conduct grid searches in 10-fold cross validation. By optimizing the hyperparameter in cross validation, our best XGBoost model is a model with {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 200}. The best cross-validation AUC score is 0.87, which is lower than the decision tree model. After retraining the model with a full train dataset, we predict the unused test dataset. The misclassification rate in unused test dataset is about 0.13, which is better than the decision tree model. Based on the f1-score on the unused dataset, we got 0.87, which is a better value than the benchmark model and decision tree model. More importantly, the AUC in the unused test dataset is about 0.8574, which overperforms the benchmark model and the decision tree model.