National College of Ireland

National College of Ireland

Project Submission Sheet – 2020/2021

Student Name:	Omkar Ratnoji Tawade		
Student ID:	19232136		
Programme:	MSc. Data Analytics (MSCDAD_A)	**Year:**	2020-21
Module:	Domain Application of Predictive Analytics		
Lecturer:	Vikas Sahni		
Submission Due Date:	04-05-2021		
Project Title:	Predicting Hotel using Machine Learning		
Word Count:	4800 words		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Omkar Ratnoji Tawade

Date: 04-05-2021

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Predicting Hotel Demand using Machine Learning

Omkar Ratnoji Tawade
Data Analytics
National College of Ireland
Dublin, Ireland
x19232136@student.ncirl.ie

***Abstract*—This is a project implementation document on predicting demands of hotel using machine learning techniques. Hotel industry is a growing industry now but there are fluctuations in the demands of hotel based on the factors positive factors like vacation period, any major event in vicinity and negative factors like pandemic, recession etc. In this study, we are using the hotel dataset which consist the past records of all types of booking and we will try to predict the demand of hotel based on number of non-cancelations bookings expected. We will be using extreme gradient boosting algorithm (XGBoost) to build this predictive model.**

Keywords—Hotel Demand, Machine Learning., XGBoost, Prediction

I. RESEARCH AND INVESTIGATION INTO THE APPLICABLE TECHNIQUES

Binru Zhang et. al [1] implemented a forecasting model to predict hotel demand based on LSTM model incorporating internet search index. They designed a logical framework of tourism information search. They categorized the tourism process into three parts pre tour plan which refers to plan made by consumers, In-tour experience which refers to tourism implementation and post tour evaluation refers to the feedback of services they received. They collected data by using scraping. In this process they selected six key words related to the destination and obtained their dataset. While data cleaning, they found that passenger flow information is very sensitive to promotion schemes, emergencies, etc. Further they checked the correlation between their keyword variables and lag variables of predicted variable. Lag variable with maximum correlation coefficient was considered as alternative predictive variable. They used stepwise regression to obtain final predictive variable as well as this step will reduce the model complexity. They implemented four model models based on LSTM, C-LSTM, DBN (Deep Belief Network) and BPNN (Back Propagation Neural Network). Among these four models, LSTM model performed better because it can detect and learn the long-time dynamic information of the time series.

Eleazar C. Sánchez et. al [2] implemented a method to identify critical hotel cancellations using acritical intelligence. In this research, researchers collected the dataset from a four start hotel which is located in centre of Gran Canaria (Spain). The data consists variables like nationality, number of nights, number of weekend days during the stay, and booking status as a predictor variable. After cleaning the data, researches implemented model using C5.0, Support Vector Machine and Artificial Neural Network. All these model were built on the principle of binary classification which classified the type of booking as cancelled booking or not cancelled booking. They used C5.0 algorithm because of its tree like structure which is better for classification purpose. Support vector machine whereas is completely different from C5.0 algorithm as it does not create tree like structures instead of this it find hyperplanes which separates data according to its features. ANN uses the neuron structure to learn the classification. Further they used the ensemble approach in which different models are combined to provide better results. They evaluated the model based on accuracy, precision, specificity, sensitivity and area under curve parameters. They concluded that their ensemble model predicted the cancelation of booking 7 days prior with a great accuracy based on AUC curves.

Naragain Phumchusri et. al [3] implemented a hotel demand forecasting for high frequency and complex seasonality data. Their work presented a forecasting models both time series and causal methods using data of 4 start hotel in Phuket, Thailand. They explored the data and categorized data into four rooms that are Deluxe, Seaview, Pool Access and View. Further they observed seasonality in the data. After confirming the seasonality, researcher used exponential smoothing (Holt-Winter) method on data. They separated data as data for constructing model and other for testing. Further they developed SARIMA model which is an extended version of ARIMA model. They also produced model using BATS and TBATS method. They compared these forecasting models and concluded that holt winter model cannot able to capture changes in hotel daily time series. BATS model explained the seasonality in the data. They also produced the artificial neural network model in which two type of regressor were used that are Independent factors and transformed data. Significance of transformed data was computed. Further different set of variables were tested to find the accuracy of forecasting. Researchers concluded that ANN model performed much better than time series and causal models.

C.Premila Rosy and R. Ponnusamy [4] proposed an intelligent system to support judgmental business forecasting. In this study, researcher considered the case of hotel room demand in hotel revenue management system. Their model comprises advanced room demand forecasting and optimization model that address group reservations. Their model used the reservations data and as well as data of past arrivals which consisted the parameters such as arrival date, reservation date, length of stay, room, type etc. Researchers analyzed the data and extracted factors like seasonality, trend, booking curve, cancelations. Based on these factors they predicted the future bookings. These predicted future bookings were given to optimization model and by analyzing these booking they obtained the realistic prediction for occupancy, arrivals, and revenue in the future. Their forecasting model derived the requirement that was need for optimization model without the need to prior assumptions Optimization model was based on large-scale integer programming which helps the hotel manager by optimizing decision rules for accepting the reservations. Also, their model

was able to generate the recommendation to increase the revenue.

Nuno Antonio et. al [5] implemented a system to predict hotel booking cancellations to avoid uncertainty and to grow the revenue. Researchers used real booking data from four hotel which are located in resort region in Portugal. They implemented their system using CRSIP-DM methodology in which they first studied about the business and found some parameters on which booking demand was highly influenced. They first analysed the relation between lead time and cancelation time over the years, also they found a relations that customers who have cancelations 5 or more time in past are more likely to cancel their bookings. Further they explore the data and considered few scenario before building thee model. Researchers checked the curse of dimensionality scenario as there were many independent variables in the dataset. They verified the correlation between the independent variables. After the data exploration, researchers executed the data preparation in which they performed the mutual feature selection. This help to find or measure the predictor variable impact on the value of dependent variable. They used Microsoft Azure machine learning studio to implement the models. Different models were produced using machine learning classification algorithms like Boosted Decision Tree, Decision Forest, Decision Jungle, Locally Deep Support Vector Machine, and Neural Network. They used k-fold cross validation to evaluate the performance of the model. It avoids the overfitting of the model. They concluded their research by comparing the performance of models. In this comparison, they found that Decision Forest was the best algorithm for 3 out of 4 hotels. Before training the model, researcher used the function Tune model hyper parameters on training data to test with different combination of algorithm's parameter and then to identify the optimum parameter for a particular algorithm.

Misuk Lee [6] proposed a model to forecast the hotel room demand based on advanced booking information . They developed short term forecasting of hotel booking using stochastic approach. They started their research by exploring the booking arrival process and during this exploration they understand that there will be a need to build separate model for each day of week. Further they analysed the high variability in final demand by observing the seasonality in the data. They built their first model based on standard non homogenous Poisson process. In this model they identified an exponential trend. They found a spike in number of bookings as the arrival date comes near. For the second model researcher used negative binomial function. Each customer has given a probability score for booking. After certain amount of booking we will observe a failure or cancelation of booking. This plotted against time and in output we will observe a negative binomial distribution. Researcher further developed third model based on negative multinomial algorithm in which they incorporated inter-temporal correlation which takes consideration that booking process is not independent every day and thus there can be association between late and early bookings. Researchers compared these three model and found that model 1 does not supported the existence of high demand variability and inter temporal correlations whereas by adding few random components in model 2 and model 3 outperformed the model 1. Researchers used the real reservation data of 69 hotel and concluded that their dynamic updating method using the inter temporal correlation can improve the short term forecasting significantly.

Aditi Malvankar [7] et.al implemented a hotel recommendation system using machine learning algorithms like Random Forest, SGD Classifier, XG Boost and Naïve Bayes. They used expedia's hotel recommendation which had two category of features. Geographical, temporal, and search features in one category and latent and destination features in other category. They used principal component analysis because of large number of latent features. Further, they performed ablation experiment to find the important features in the dataset. After preparing the dataset, they computed and plotted the correlation matrix of independent variables to check highly correlated variables. Their outcome variable was a categorical variable which categorized the similar hotels together. After analysing the independent and dependent variables, researchers used the train data to generate different models using algorithms like random forest, stochastic gradient descent classifier, naïve bayes, and xg-boost. They used ensemble learning approach in which these models are combined to solve an intelligent problem. Each model was checked with the test data and outcome of these model were given as five most probable clusters for each user in the test dataset. They also found the problem of data leakage in their study and features such as location of the user, origin-destination distance, hotel market and search destination id were the reasons for data leakage. They proposed a solution to this problem to find cases in training data which have same values in test data features. In their final stage of the study, researcher compare the model based on MAPE values and concluded that XG-boost performed better than other models but their solution of using combination of ensemble learning and data leakage solution model performed effectively compared to other models.

Ting Hu and Ting Song [8] proposed a research on academic forecasting and analysis XG Boost. In this study researchers used result of student of Grade 15 in Nanjing University of Aeronautics and Astronautics. They started their work by identifying the correlation between the subjects. Marks of each subject of each student were the data of this study, so each subject were acted as features. They use the XG Boost algorithm to classify the students based on subject features. They also used XG Boost algorithm to analyze the relevant of the course. They collected the dataset in which result of first three semester were recorded. They observed that XG Boost algorithm computes fast as well as have high accuracy and can learn on less resource. Also, they highlighted the key feature of XG Boost algorithm is that when this algorithm is trained on large dataset then it has ability to perform parallel computing which is not achievable in traditional algorithms. Also, this algorithm uses feature importance degree and evaluation method to calculate the relevance of course.

II. IMPLEMENTATION OF TECHNIQUES

We will start our implementation of our project with data cleaning. In our design report, we did some explanatory analysis of our variables in the project such as bookings distribution across the hotel, number of booking canceled/confirmed distribution, cancelation of bookings across year, month and day, frequency of repeated guests, medium of bookings, and graph between cancelation of bookings against number of days in waiting list which helped us to gain some insights from the data and which will eventually help us to produce a better model. Now we will analyze the data and based on our analysis we will clean the

data or add more features in the data to make model perform better for prediction. There are 32 variables in the dataset among which is_canceled will be our dependent variable. There are 119390 number of observations in the data. Our dependent variable is a categorical variable. The whole dataset contains 18 categorical variables and 14 numeric variables. Since our dataset is too large so there are few areas we should consider before data cleaning. In our initial analysis we have noticed that our dataset consists of total 31994 duplicate rows. We have removed the duplicate rows from the dataset and after removing these duplicate rows the total number of observations became 87396 observations. We have also observed that country variable has high cardinality. High cardinality denotes that those respective variable have high distinct values. This can add problems in modelling. But in our case, it will not have an issue since it is obvious that number of unique countries must be high in numbers. In our initial observation we have found that our dependent variable is_canceled is highly correlated with reservation_status variable.

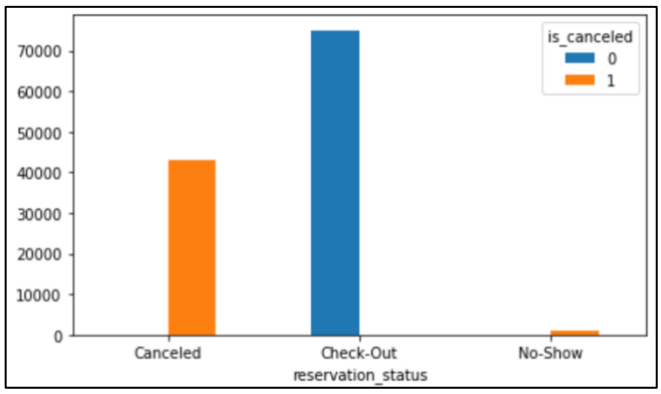


Figure 1. Correlation between is_canceled and reservation_status.

As we can see both variable are almost same, we can drop reservation_status variable from our dataset as it will affect our model to predict better. Due to privacy rights, some cells do not have value and there are 129425 such cells which have null values. Let us analyze the null values in the data by observing the heat map.

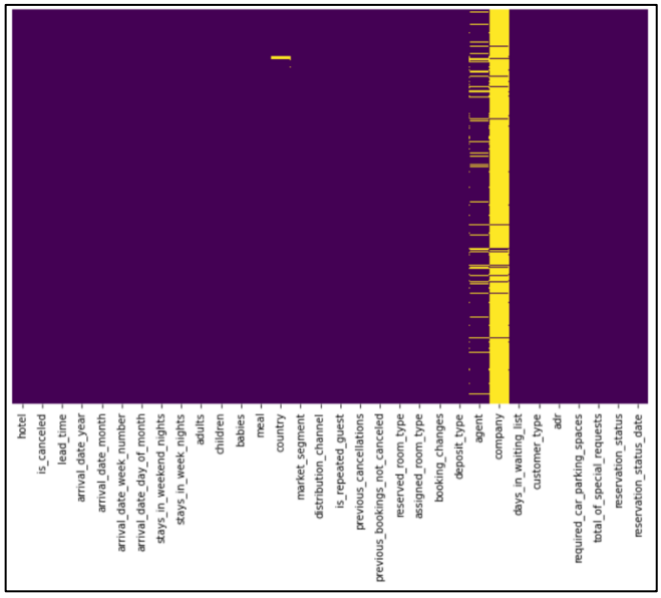


Figure 2. Heatmap to visualise null value in the data.

Based on Figure 2, we can see that there are too many null values in company variable and agent variable. These are null due to data protection. These variables will not impact the result of our model. We have decided to replace null value in company and agent variables by zero.

```
company                        82137
agent                          12193
country                          452
children                           4
lead_time                          0
arrival_date_year                  0
arrival_date_month                 0
arrival_date_week_number           0
is_canceled                        0
market_segment                     0
dtype: int64
```

Figure 3. NA's in the dataset

As we can see in the

```
company                        82137
agent                          12193
country                          452
children                           4
lead_time                          0
arrival_date_year                  0
arrival_date_month                 0
arrival_date_week_number           0
is_canceled                        0
market_segment                     0
dtype: int64
```

Figure 3, there are few variables which have still NA's in it. For country variable, we have decided to replace NA's with most repeating country in the dataset. We have used the mode function to find the most repeating country in the dataset and we have replaced that country with the NA's in the country column. Further, adr_pp which denotes average daily rate per person have 153 null values. In this case also we will be using the mode function to replace NA's in adr_pp column. We have used the mode function because range of adr_pp is large but the values in adr_pp is not evenly distributed. There are outlier in ad_pp column because of this uneven distribution of value and large range. So to avoid any change in the meaning of the data we are using the mode function as we can see that '100' number of guest are repeated more. Children is the last variable in which we can observe few NA's. By observing the frequency distribution of children we have decided to use the mean function to replace the NA's in the children.

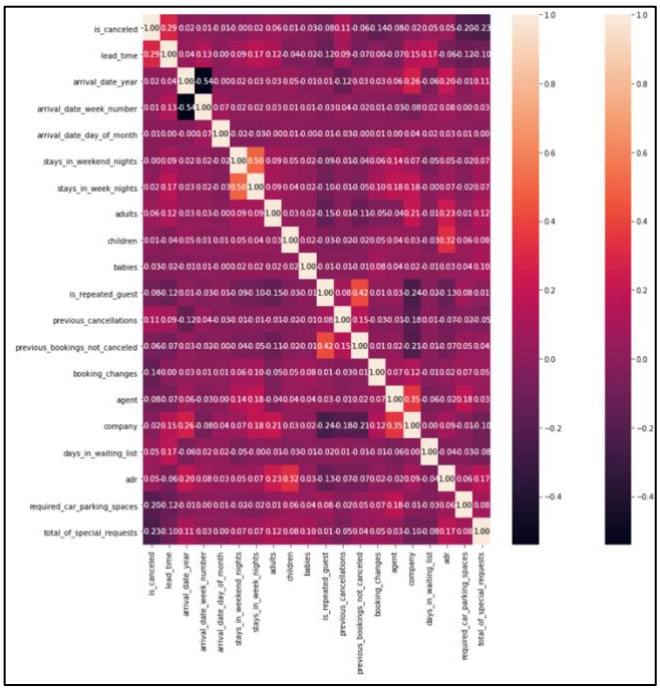


Figure 4. Correlation Plot

After data cleaning, we have implemented the correlation plot. Figure 4 suggest that there no variable which have correlation value above 0.7. So based on this criteria, we can say that there is no correlation present in our dataset. But this correlation plot does not show correlation show between categorical variables. As, we discussed earlier in our initial analysis that reservation_status is almost identical to the is_canceled so there is high correlation between these variables. We also computed a model by incorporating this variable and we got an accuracy of 96% and also the recall rate of the model was 0.93 but it was incorrect as model will be overfitted due to this criteria. We will drop this variable while building the model. Along with this variable, we analysed few variable which does not have significant impact on our dependent variable which are year, week number, day, date, assigned room type, reserved room type, reservation status date, previous cancelations, previous bookings not cancelled. We will be removing these variable from our dataset before model building. Our data cleaning process in now completed and our data is ready for the model building. We have decided to divide the dataset in to three part which is train, validation and test. The purpose of this is to avoid overfitting of the model. Validation data is used while training and it prevent the overfitting of the model whereas the test data dataset will be used to predict the cancelation of bookings which records were not present in the train data. So initially, we have separated the dataset in to two data frames which are for train/val and test with (70:30) ratio. Further we converted the categorical variables data into categorical codes for better analysis. We have created a numpy array for each variable in the dataset as it will help to remove or add variable in the model. By using the stack function of numpy library we have stacked required variable in the numpy array. Further, we used train_test split function on this stacked numpy array and we got the train data of input and output variable, also we got the validation data of input and output variable. While building the model initially, we found that there is class imbalance problem present in our study.

We will now understand the class imbalance problem. Imbalance is nothing but the disproportion or variance in terms of classes. It is present when number of positive classes are less than the negative classes or vice versa.

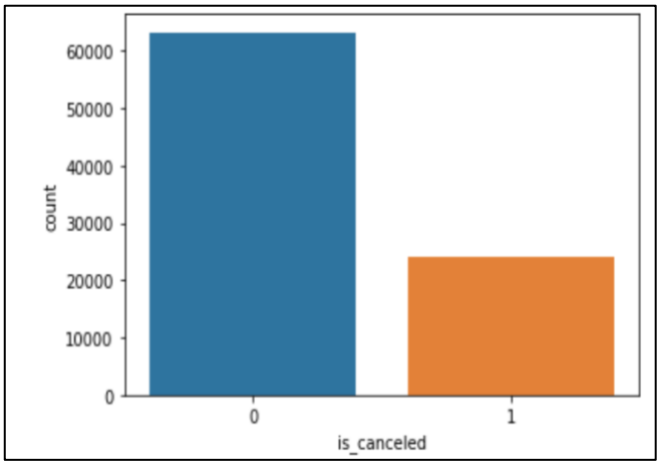


Figure 5. Class Imbalance

As we can see in Figure 5, the proportion of data of cancelled bookings is less than half of the booking which are not cancelled. There is amount of difference of data available for the two classes and this difference is known as class imbalance. We will now understand how class imbalance affect our model. When we develop a prediction model on such data then that predictive model will be dominated by class which have high number of records and in our case, model will be dominated by bookings which are not cancelled. Also the accuracy of the model will be better for predicting the class which have high records and in our case model accuracy will be better for predicting booking which are not cancelled. But our ultimate goal of this project is to predict the cancelation of bookings so the accuracy of predictive model should be better for predicting the booking which is cancelled. We can also confirm the class imbalance problem by just observing the value of sensitivity and specificity value of the prediction model. We will be able to see the large fluctuation among those two values. In extreme gradient boosting algorithm we can resolve the issue of class imbalance problem and can avoid skewing of our model. There is a parameter in extreme gradient boosting algorithm known as scale_pos_weight which is used to increase the scale weights. It will penalise the errors of minor class to a greater extent than on the major class. In our study it will penalise the errors by records who cancelled their hotel booking more than the errors by records who did not cancelled their bookings. Now we will understand how can we implement extreme gradient boosting algorithm. Extreme gradient boosting is a decision tree based ensemble machine learning algorithm that uses a gradient boosting framework. This framework has advantages like regularized boosting which avoids overfitting, can handle the missing values automatically, parallel processing and because of which it is a fastest framework, can cross validate at each iteration, early stopping, finding optimal number of iterations, tree pruning. Although it is very easy to use but the majority of time is invested in tuning the hyper parameter to get a better model and this can be achieved only after experimenting with

hyperparameter values. In our case we used the extreme gradient boosting classifier framework. Further we need to choose the objective function. There are two type of objective function in this framework which are softmax and softprob. Softmax is used to choose one classification which works best among many classification whereas softprob is used to show the probability of each classification. We will be going to use the softprob objective function in our study. Next parameter of extreme gradient boosting is eta which is known as learning rate. It adjust the weight on each step and its default value will be 0.3 and by reducing the learning rate model will able to perform better but it can also make model prone to overfitting. So we should tinker this value accordingly and select the best learning rate for our model. Further parameter of this frame work is max_depth which denotes the maximum depth of tree. Like learning rate this parameter also needs to tune carefully. If we select a small value for depth of tree then it will create a smaller tree for our model and result will not be good whereas if we set a higher number of tree for maximum depth then model will be overfitted. So we should also tinker this value in order to achieve the optimal depth of tree for our model. We will now discuss the hyper parameter tuning in extreme gradient boosting algorithm which will be required in our study to produce a better predictive model as well as to remove problems like class imbalance and to avoid overfitting and underfitting our model. There are almost 17 parameters in extreme gradient boosting classifiers among which only few are essential. We will discuss these few hyper parameter and will see how we implemented the tuning of this hyper parameters in our study. First hyper tunning parameter is subsample, it denotes the amount of data is taken for tree building. It ranges between 0.1 and 1 and its default value is 1 which denotes it take 100% of the data for tree building. If we decrease the value of sub sample it will add regularization effect in our model and when we increase the value of subsample then it will overfit the model. Next hyper tuning parameter is columns by level which denotes number of features considered for each level of the tree. It range between 0.1 and 1 where 1 indicates that all features are consider for each level in the tree. It also behaves similarly like sub sample in which if we increase the value of columns by sample then it will overfit the model. The purpose of using this parameters is for regularisation and which will help to model generalise better. n_estimators is one of the important parameter in extreme gradient boosting which denotes the number of trees in the model. These estimators are built in serial manner such that previous estimator boosts the next estimator. Next hyper tuning parameters will indicate whether our model is better or not. random_state will be used to verify the model performance. If we change the random_state value then model training and test accuracy should not change. If it changes then it indicates that model is not working properly.

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done  25 out of  25 | elapsed:  1.4min finished
RandomizedSearchCV(cv=5, error_score=nan,
                   estimator=XGBClassifier(base_score=0.5, booster='gbtree',
                                           colsample_bylevel=1,
                                           colsample_bynode=1,
                                           colsample_bytree=1, gamma=0,
                                           learning_rate=0.1, max_delta_step=0,
                                           max_depth=3, min_child_weight=1,
                                           missing=None, n_estimators=100,
                                           n_jobs=1, nthread=None,
                                           objective='binary:logistic',
                                           random_state=0, reg_alpha=0,
                                           reg_lambda=1, sc...
                                           verbosity=1),
                   iid='deprecated', n_iter=5, n_jobs=-1,
                   param_distributions={'colsample_bytree': [0.3, 0.4, 0.5,
                                                             0.7],
                                        'gamma': [0.0, 0.1, 0.2, 0.3, 0.4],
                                        'learning_rate': [0.05, 0.1, 0.15, 0.2,
                                                          0.25, 0.3],
                                        'max_depth': [3, 4, 5, 6, 8, 10, 12,
                                                      15],
                                        'min_child_weight': [1, 3, 5, 7]},
                   pre_dispatch='2*n_jobs', random_state=None, refit=True,
                   return_train_score=False, scoring='roc_auc', verbose=3)
```

Figure 6. Hyper Parameter Tuning

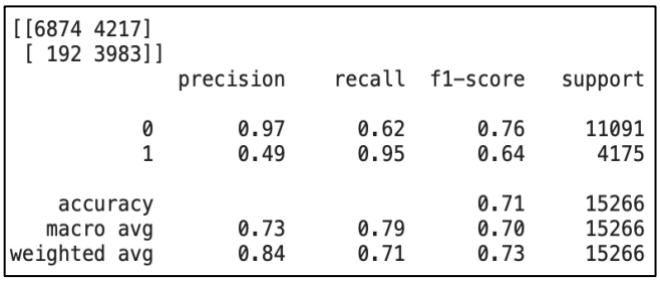
```
[24] random_search.best_estimator_

     XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                   colsample_bynode=1, colsample_bytree=0.7, gamma=0.2,
                   learning_rate=0.1, max_delta_step=0, max_depth=12,
                   min_child_weight=3, missing=None, n_estimators=100, n_jobs=1,
                   nthread=None, objective='binary:logistic', random_state=0,
                   reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                   silent=None, subsample=1, verbosity=1)

[25] random_search.best_params_

     {'colsample_bytree': 0.7,
      'gamma': 0.2,
      'learning_rate': 0.1,
      'max_depth': 12,
      'min_child_weight': 3}
```

Figure 7. Best Hyper Parameters

III. FINDINGS AND INTERPRETATION

We have performed the required data preparation for modelling. We have initially separated the independent and dependent variable, we have then converted the dependent variable into a NumPy array similarly we have converted each independent variable in to a NumPy array. Further, we have stacked the NumPy arrays of independent variable one over another to create a stack of array. We have used this stack of array of independent variables and NumPy array of dependent variables to create the train and test vectors using train test split method. Now our all parameter of extreme gradient boosting classifier are ready, so we can run our model and can check how it performed. We have got an accuracy for training and validation set almost equal. Our model training accuracy is 0.819 and validation accuracy is 0.798. It is expected that we will not get a much higher accuracy as that will denote that model is trained properly for this scenario. In our case, percentage of cases that cancelled their bookings are very less and also our main aim is to predict the cancellation of hotel bookings. We should not check the accuracy factor in this case as it will mislead us, it is obvious that due to high number of records wo did not cancelled their booking are present in our dataset will try to reduce the accuracy figure even though we have used scale_pos_weight parameter of extreme gradient boosting classifier to remove this class imbalance problem. There is another parameter in the confusion matrix which is known as recall. Confusion matrix comprises of four factors mainly which are precision, recall, f1-score and support. Since our problem is a classification problem, we will need to use the confusion matrix in order to view the classification results.

We will understand the idea of precision vs recall. They are mainly influenced by false positives and by false negatives in a particular dataset.

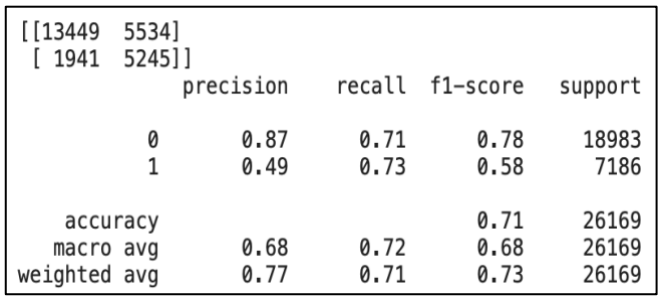


```
[[6874 4217]
 [ 192 3983]]
              precision    recall  f1-score   support

           0       0.97      0.62      0.76     11091
           1       0.49      0.95      0.64      4175

    accuracy                           0.71     15266
   macro avg       0.73      0.79      0.70     15266
weighted avg       0.84      0.71      0.73     15266
```

Figure 8. Confusion matrix of the model

Precision parameter is used when false positive cases are high. In our case as we are predicting the customers who can cancel their booking and if we produced a model that has very low precision value then many customers will be marked as customer who are suspectable to cancel their booking which should not be happen in our case. When false positives value are too high, then model will able to detect less customers who are suspectable to cancel their bookings. As we can see in the Figure 8, we have low precision value which suggest that our model is built on the principle to predict the customers who can cancel their bookings.

Recall is used when false negatives are high. In our study, customers who can cancelled their bookings comes under a negative class whereas customers who did not cancelled the bookings comes under positive class. Hotel manager can use false negative cases to detect features which are responsible for cancelation of bookings. In this case we have the liberty to detect to detect false negative case as it will help hotel manager to estimate the cancellation of bookings. As we can see in the Figure 8, our model has high recall value for customers who can cancel their bookings which suggest that it is likely to predict customers who can cancel their booking more. We will now test our model and will we use classification matrix to gauge the accuracy of the model. While splitting the data we have use 70-30 ratio for training and testing. Around 19,000 records are available for testing our model. We have followed the same steps for data preparation which we used while training the data. We have first created NumPy array for each variable that we have considered for training data. We also converted the categorical variables value in to categorical code. Further we just stacked this NumPy array one over another and feed this data to our model for predictions.



```
[[13449  5534]
 [ 1941  5245]]
              precision    recall  f1-score   support

           0       0.87      0.71      0.78     18983
           1       0.49      0.73      0.58      7186

    accuracy                           0.71     26169
   macro avg       0.68      0.72      0.68     26169
weighted avg       0.77      0.71      0.73     26169
```

Figure 9. Confusion Matrix of Test Data

Based on Figure 9, we can say that our model is performing good as it suggested that re call value of class 1 is 0.73 which is good when we compared to training confusion matrix of training data. We can conclude that with the help of our initial analysis and machine learning based model, hotel manager can manage the demands of hotel booking efficiently. As we saw in our initial analysis that our data has seasonality which can be used by manger to know the demand but as we saw there is also few cancellation for bookings in that period so in this case our model will help manager to highlight the customers who are more likely to cancel their reservation.

REFERENCES

[1] Binru Zhang, Yulian Pu , Yuanyuan Wang and Jueyou Li, "Forecasting Hotel Accommodation Demand Based on LSTM Model Incorporating Internet Search Index" *Sustainability*. China, vol. 11, pp. 1–14, August 2019.

[2] Sánchez, E. C., Sánchez-Medina, A. J., & Pellejero, M. (2020). "Identifying critical hotel cancellations using artificial intelligence. Tourism Management Perspectives".ScienceDirect, Spain, 35, 100718, pp. 1-8, 2018.

[3] Naragain Phumchusri & Phoom Ungtrakul, 2020. Hotel daily demand forecasting for high-frequency and complex seasonality data: a case study in Thailand. Journal of Revenue and Pricing Management, Palgrave Macmillan, vol. 19(1), pages 8-25, February.

[4] C.Premila Rosy, R. Ponnusamy, "Intelligent System to Support Judgmental Business Forecasting: The Case of Hotel Room Demand in Hotel Revenue Management System". IOSR Journal of Computer Engineering, India, vol. 16, pp 50-56, 2014

[5] Nuno Antonio, Ana de Almeida, and Luis Nunes, "Predicting hotel booking cancellations to decrease uncertainty and increase revenue". Tourism & Management Studies, Portugal, 13(2), pp 25-39, 2017.

[6] Lee, M. "Modeling and forecasting hotel room demand based on advance booking information". Tourism Management, 66, pp 62–71, 2018.

[7] Mavalankar, Aditi & Gupta, Ajitesh & Gandotra, Chetan & Misra, Rishabh. "Hotel Recommendation System".2019

[8] Ting Hu and Ting Song; "Research on XGboost academic forecasting and analysis modelling". Journal of Physics. 2019