Hotel Dataset

Problem Statement:

The data is from one of the world's largest online travel agencies (OTA) which powers search results for millions of travel shoppers every day.

In this competitive market matching users to hotel inventory is very important since users easily jump from website to website. As such, having the best ranking of hotels ("sort") for specific users with the best integration of price competitiveness gives an OTA the best chance of winning the sale.

The dataset includes shopping and purchase data as well as information on price competitiveness. The data are organized around a set of "search result impressions", or the ordered list of hotels that the user sees after they search for a hotel on the agency's website. The user response is provided as a click on a hotel or/and a purchase of a hotel room.

To predict if the user chooses a hotel based on the users' browsing history from search to bookings and payments.

Exploratory Data Analysis and Data Pre-processing:

Types of features in the dataset

search_id	float64
timestamp	object
site_id	int64
user_country_id	int64
user_hist_stars	float64
user_hist_paid	float64
listing_country_id	int64
listing_id	int64
listing stars	int64
listing_review_score	float64
is_brand	int64
location_score1	float64
location_score2	float64
log_historical_price	float64
listing_position	int64
price_usd	float64
has_promotion	int64
destination_id	int64
length_of_stay	int64
booking_window	int64
num_adults	int64
num_kids	int64
num_rooms	int64

stay_on_saturday	int64
log_click_proportion	float64
distance_to_dest	float64
random_sort	int64
competitor1_rate	float64
competitor1_has_availability	float64
competitor1_price_percent_diff	float64
competitor2_rate	float64
competitor2_has_availability	float64
competitor2_price_percent_diff	float64
competitor3_rate	float64
competitor3_has_availability	float64
competitor3_price_percent_diff	float64
competitor4_rate	float64
competitor4_has_availability	float64
competitor4_price_percent_diff	float64
competitor5_rate	float64
competitor5_has_availability	float64
competitor5_price_percent_diff	float64
competitor6_rate	float64
competitor6_has_availability	float64
competitor6_price_percent_diff	float64
competitor7_rate	float64
competitor7_has_availability	float64
competitor7_price_percent_diff	float64
competitor8_rate	float64
competitor8_has_availability	float64
competitor8_price_percent_diff	float64
clicked	int64
booking_value	float64
booked	int64

Summary statistics

	search_id	site_id	user_country_id	user_hist_stars	user_hist_paid	listing_country_id	listing_id	listing_stars
count	2.380557e+06	2.380557e+06	2.380557e+06	122780.000000	123494.000000	2.380557e+06	2.380557e+06	2.380557e+06
mean	3.337016e+05	9.970224e+00	1.754588e+02	3.382814	178.094940	1.739159e+02	7.008190e+04	3.180607e+00
std	1.923719e+05	7.667827e+00	6.585934e+01	0.694562	108.568025	6.832483e+01	4.060398e+04	1.052086e+00
min	4.000000e+00	1.000000e+00	1.000000e+00	1.500000	0.000000	1.000000e+00	1.000000e+00	0.000000e+00
25%	1.674260e+05	5.000000e+00	1.000000e+02	2.950000	111.090000	1.000000e+02	3.502800e+04	3.000000e+00
50%	3.332720e+05	5.000000e+00	2.190000e+02	3.450000	152.620000	2.190000e+02	6.961500e+04	3.000000e+00
75%	5.007050e+05	1.400000e+01	2.190000e+02	3.950000	215.950000	2.190000e+02	1.051420e+05	4.000000e+00
max	6.655730e+05	3.400000e+01	2.310000e+02	5.000000	1507.120000	2.300000e+02	1.408210e+05	5.000000e+00

Handling missing values:

List of features with missing values

‘user_hist_stars'	'competitor3_price_percent_diff'
'user_hist_paid'	'competitor4_rate'
'listing_review_score'	'competitor4_has_availability'
'location_score2'	'competitor4_price_percent_diff'
'log_click_proportion'	'competitor5_rate'
'distance_to_dest'	'competitor5_has_availability'
'competitor1_rate'	'competitor5_price_percent_diff'
'competitor1_has_availability'	'competitor6_rate'
'competitor1_price_percent_diff'	'competitor6_has_availability'
'competitor2_rate'	'competitor6_price_percent_diff'
'competitor2_has_availability'	'competitor7_rate'
'competitor2_price_percent_diff'	'competitor7_has_availability'
'competitor3_rate'	'competitor7_price_percent_diff'
'competitor3_has_availability'	'competitor8_rate'
'booking_value'	'competitor8_has_availability'
'competitor8_price_percent_diff'	

All the missing values have been handled by replacing with:

- median if the feature is skewed
- mode if it contains categorical data
- mean if it follows normal distribution

Data Sampling and Normalization:

- Used Stratified split to select 1% of the data from the original dataset and to ensure the class balance is kept intact.
- Normalized the features which had higher range to bring it to a uniform scale.

Model details

I have choosen logistic regression model as the class variable is categorical in nature. Initial model was ran by considering all the parameters. This resulted in a low f1 score. Hence feature selection and hyper parameter tuning was applied to increase the score.

Feature selection

1. Used corr() for detecting multicollinearity among features.

- In case of multicollinearity, we might lose reliability in determining the effects of individual features in the model and hence, we remove these features.

- To resolve this, we use corr() method which returns a correlation score for each feature.
- This score determines the strength of correlation between the independent variables. All the features with a score greater than 0.50 were removed from the dataset suggesting high multicollinearity.

The following features: search_id, listing_id and time_stamp are irrelevant and hence have been removed.

Model Building and Hyperparameter tuning

1. Choosing the right predictor variable:

Out of the 3 possible outcome variables: clicked, bookings_value and booked. I chose "clicked" as the outcome variable for my model considering the following reasons:

1. Since, we are trying to predict whether the user is going to pick the hotel given multiple choices of hotels and competitors' websites, the ideal choice would be to predict the maximum probability of a click ("clicked") which indicates a visitor's interest and potentially a decision to book.
2. "Bookings_value" does not apply here since it does not address the problem statement in hand, our goal is not price prediction rather if the user is going to pick a hotel given multiple choices of hotels and competitor's websites.
3. The outcome variable "Booked" would come into picture when we are trying to predict if the user is going to book a particular hotel or not given a single website/source. In our case, we are also trying to compare between different websites, hence the outcome variable "Booked" is not applicable here.

2. Machine Learning algorithm used:

Applied "Cost Sensitive/Weighted Logistic Regression" taking into consideration the following reasons:

- Predicted variable is categorical
- Class has imbalanced data: The dataset has imbalanced class with approximately:

  94% - negative class and 6% - positive class because of which the accuracy metrics get inflated. This has been handled by considering the metric "F1 score" while tuning hyper parameters and measuring the success of the model.

3.Hyperparameter Tuning:

- Applied GridSearchCV to fetch optimum values for weights and alpha.
- Used Kfold cross validation to ensure we are not overfitting the model.

4.Selection of significant features:

- Used RFE algorithm to find out significant features for the model and removed insignificant features.

4. Evaluation metrics:

I chose “average f1 score” to evaluate the accuracy of the model and also to tune the hyper parameters of the model.

Since we are predicting class labels and both false negatives and false positives are equally costly, we use the F1 measure as the evaluation metric for our model.

5. Click through rate and conversion rate for property stars:

From the below prediction, we can see that CTR for property stars is highest for 4-star hotel and conversion rate is highest for 3-star hotel

	listing_stars	Click Through Rate for Property Stars
0	0.0	0.034335
1	1.0	0.041667
2	2.0	0.039497
3	3.0	0.043162
4	4.0	0.046075
5	5.0	0.043478

	listing_stars	Conversion Rate for Property Stars
0	0.0	0.025751
1	2.0	0.026930
2	3.0	0.031624
3	4.0	0.031286
4	5.0	0.028986

6. Click through rate and conversion rate for property review score:

From the below prediction, we can see that CTR for property review score is highest for score 5 and conversion rate is highest for 2.5 score.

	listing_review_score	Click Through Rate for Property Review Score
0	0.0	0.020202
1	2.0	0.036145
2	2.5	0.051136
3	3.0	0.033126
4	3.5	0.045802
5	4.0	0.050538
6	4.5	0.039387
7	5.0	0.052045

	listing_review_score	Conversion Rate for Property Review Score
0	0.0	0.010101
1	2.0	0.024096
2	2.5	0.039773
3	3.0	0.022774
4	3.5	0.034896
5	4.0	0.033333
6	4.5	0.028993
7	5.0	0.033457

Conclusions:

I achieved a model f1-score of 89%. Click through rate and conversion rate have been calculated for different levels of property stars and property review score for the test data.