Analyzing online travel search history

→ Dheeraj Perumandla

Given data:

ID		Data Type	Description
1	search_id	Integer	The ID of the search
2	timestamp	Date/time	Date and time of the search
3	site_id	Integer	ID of the website point of sale (i.e..com, .co.uk, .co.jp, ..)
4	user_country_id	Integer	The ID of the country the customer is located
5	user_hist_stars	Float	The mean star rating of hotels the customer has previously purchased; null signifies there is no purchase history on the customer
6	user_hist_paid	Float	The mean price per night (in US$) of the hotels the customer has previously purchased; null signifies there is no purchase history on the customer
7	listing_country_id	Integer	The ID of the country the hotel is located in
8	listing_id	Integer	The ID of the hotel
9	listing_stars	Integer	The star rating of the hotel, from 1 to 5, in increments of 1. A 0 indicates the property has no stars, the star rating is not known or cannot be publicized.
10	listing_review_score	Float	The mean customer review score for the hotel on a scale out of 5, rounded to 0.5 increments. A 0 means there have been no reviews, null that the information is not available.
11	is_brand	Integer	+1 if the hotel is part of a major hotel chain; 0 if it is an independent hotel
12	location_score1	Float	A (first) score outlining the desirability of a hotel's location
13	location_score2	Float	A (second) score outlining the desirability of the hotel's location
14	log_historical_price	Float	The logarithm of the mean price of the hotel over the last trading period. A 0 will occur if the hotel was not sold in that period.
15	listing_position	Integer	Hotel position on the search results page. This is only provided for the training data, but not the test data.
16	price_usd	Float	Displayed price of the hotel for the given search. Note that different countries have different conventions regarding displaying taxes and fees and the value may be per night or for the whole stay
17	has_promotion	Integer	+1 if the hotel had a sale price promotion specifically displayed
18	booking_value	Float	Total value of the transaction. This can differ from the price_usd due to taxes, fees, conventions on multiple day bookings and purchase of a room type other than the one shown in the search
19	destination_id	Integer	ID of the destination where the hotel search was performed
20	length_of_stay	Integer	Number of nights stay that was searched
21	booking_window	Integer	Number of days in the future the hotel stay started from the search date
22	num_adults	Integer	The number of adults specified in the hotel room
23	num_kids	Integer	The number of (extra occupancy) children specified in the hotel room
24	num_rooms	Integer	Number of hotel rooms specified in the search
25	stay_on_saturday	Boolean	+1 if the stay includes a Saturday night, starts from Thursday with a length of stay is less than or equal to 4 nights (i.e. weekend); otherwise 0
26	log_click_proportion	Float	The log of the probability a hotel will be clicked on in Internet searches (hence the values are negative) A null signifies there are no data (i.e. hotel did not register in any searches)
27	distance_to_dest	Float	Physical distance between the hotel and the customer at the time of search. A null means the distance could not be calculated.
28	random_sort	Boolean	+1 when the displayed sort was random, 0 when the normal sort order was displayed
29	competitor1_rate	Integer	+1 if agency has a lower price than competitor 1 for the hotel; 0 if the same; -1 if the agency's price is higher than competitor 1; null signifies there is no competitive data
30	competitor1_has_availability	Integer	+1 if competitor 1 does not have availability in the hotel; 0 if both agency and competitor 1 have availability; null signifies there is no competitive data
31	competitor1_price_percent_diff	Float	The absolute percentage difference (if one exists) between the agency and competitor 1's price (agency's price the denominator); null signifies there is no competitive data
32	competitor2_rate	Integer	(same, for competitor 2 through 8)
33	competitor2_has_availability	Integer	
34	competitor2_price_percent_diff	Float	
35	competitor3_rate	Integer	
36	competitor3_has_availability	Integer	
37	competitor3_price_percent_diff	Float	
38	competitor4_rate	Integer	
39	competitor4_has_availability	Integer	
40	competitor4_price_percent_diff	Float	
41	competitor5_rate	Integer	
42	competitor5_has_availability	Integer	
43	competitor5_price_percent_diff	Float	
44	competitor6_rate	Integer	
45	competitor6_has_availability	Integer	
46	competitor6_price_percent_diff	Float	
47	competitor7_rate	Integer	
48	competitor7_has_availability	Integer	
49	competitor7_price_percent_diff	Float	
50	competitor8_rate	Integer	
51	competitor8_has_availability	Integer	
52	competitor8_price_percent_diff	Float	
53	clicked	Boolean	if the listing is clicked 1, else 0
54	booked	Boolean	if the listing is booked 1, else 0

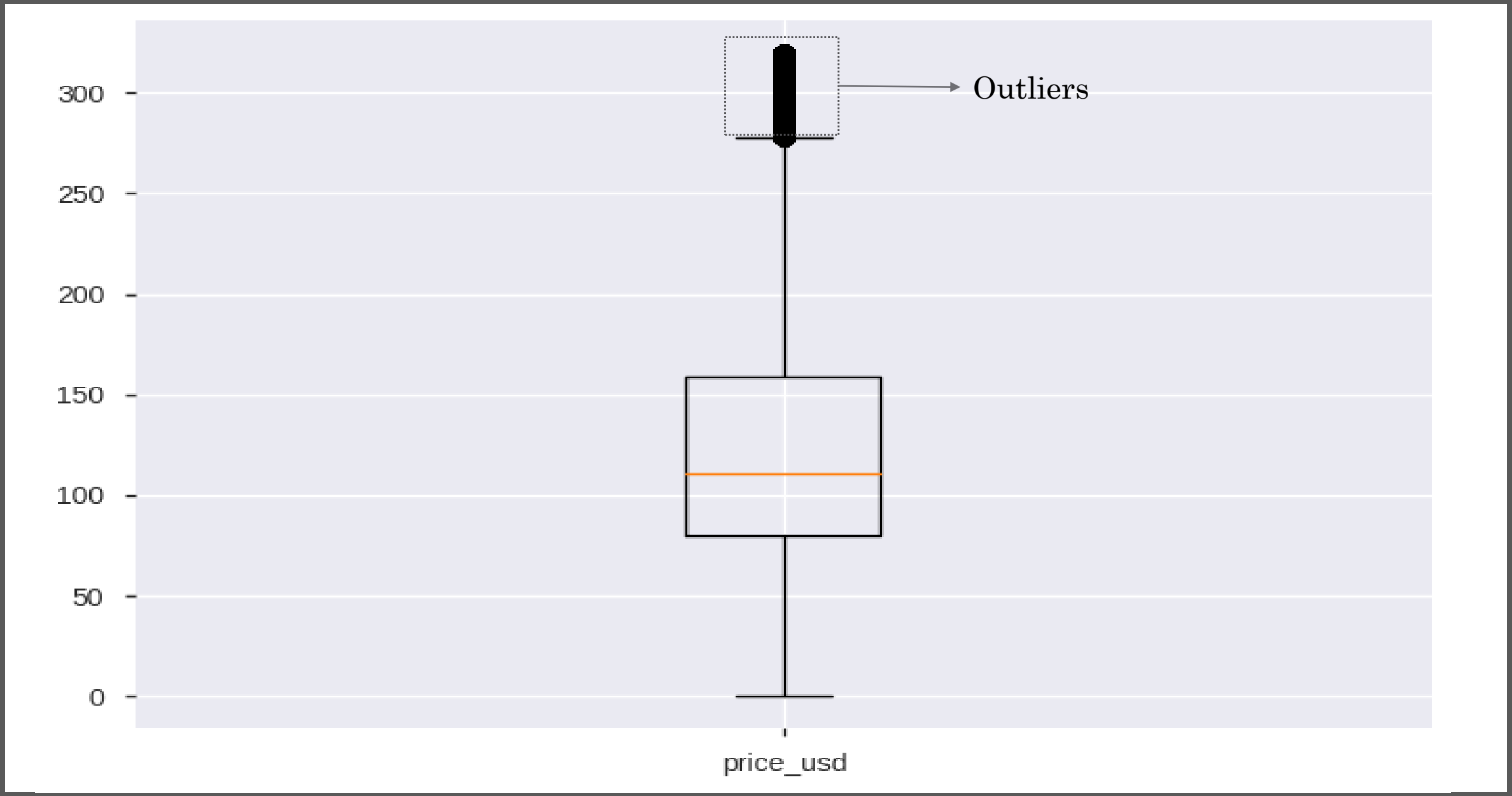
	df_index	search_id	timestamp	site_id	user_country_id	listing_country_id	listing_id	listing_stars	listing_review_score	is_brand	location
0	0	4	2012-12-31 08:59:22	5	219	219	3625	4	4.0	0	3.22
1	1	4	2012-12-31 08:59:22	5	219	219	11622	4	4.0	0	2.71
2	11	4	2012-12-31 08:59:22	5	219	219	64344	4	3.0	1	1.79
3	12	4	2012-12-31 08:59:22	5	219	219	65984	2	3.5	0	3.09
4	19	4	2012-12-31 08:59:22	5	219	219	85567	2	4.0	1	0.69
5	20	4	2012-12-31 08:59:22	5	219	219	85742	3	4.0	0	1.10
6	21	4	2012-12-31 08:59:22	5	219	219	89119	4	4.5	1	3.18
7	24	4	2012-12-31 08:59:22	5	219	219	110813	2	4.0	0	2.40
8	25	4	2012-12-31 08:59:22	5	219	219	116696	2	4.0	0	2.40
9	26	4	2012-12-31 08:59:22	5	219	219	125069	3	4.0	1	1.61

Handling Null values:

- Features with large number of null values are dropped in order to not corrupt the data
- Features with minimal number of null values are imputed with mean or mode or with zeros depending on the type of feature it is
- Null values in booking_value are filled with 0.
- Null values in listing_review_score are filled with corresponding values in listing_stars as they both are similar and reflect similar information.
- Null values in dinstance_to_dest are filled with mean.

Outlier handling:

- Box plot use the IQR method to display data and outliers(shape of the data) but in order to be g et a list of identified outlier, we will need to use the mathematical formula and retrieve the out lier data.
- The interquartile range (IQR), also called the midspread or middle 50%, or technically H-spread, is a measure of statistical dispersion, being equal to the difference between 75th and 2 5th percentiles, or between upper and lower quartiles, IQR = Q3 − Q1.
- In other words, the IQR is the first quartile subtracted from the third quartile; these quartiles can be clearly seen on a box plot on the data.
- It is a measure of the dispersion similar to standard deviation or variance, but is much more ro bust against outliers.
- IQR is somewhat similar to Z-score in terms of finding the distribution of data and then keeping some threshold to identify t he outlier.


Outliers
300
250
200
150
100
50
0
price_usd

Handling class Imbalance:

- The distribution of diagnoses is important because it speaks to class imbalance within machine learning and data mining applications. Class imbalance is a term used to describe when a target class within a data set is outnumbered by another target class (or classes). This can create misleading accuracy metrics, known as an accuracy paradox. To make sure our target classes aren't imbalanced, create a function that will output the distribution of the target classes.
- A problem with imbalanced classification is that there are too few examples of the minority class for a model to effectively learn the decision boundary.
- One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.
- An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective.
- Perhaps the most widely used approach to synthesizing new examples is called the **Synthetic Minority Oversampling TEchnique**, or SMOTE for short

booked

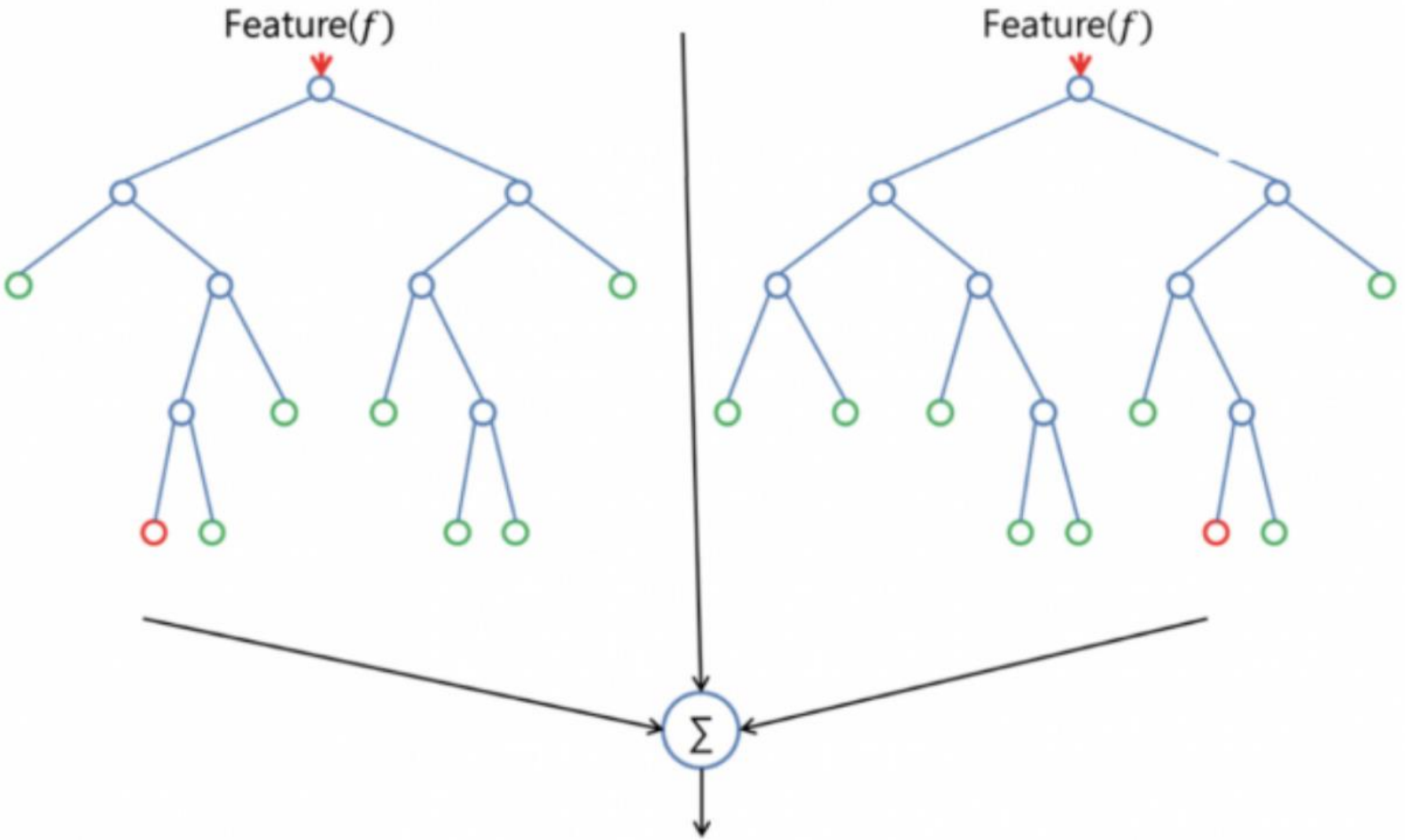
Boolean

Distinct count	2
Unique (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	9.9 MiB

Toggle details

Algorithm used: Random Forest

- Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.
- **Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.**
- One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:


Feature(f)
Feature(f)
Σ

Cross Validation

Cross validation is a powerful tool that is used for estimating the predictive power of your model, and it performs better than the conventional training and test set. Using cross validation, we can create multiple training and test sets and average the scores to give us a less biased metric.
In this case, we will create 10 sets within our data set that calculate the estimations we have done already, then average the prediction error to give us a more accurate representation of our model's prediction power. The model's performance can vary significantly when utilizing different training and test sets.

K-Fold Cross Validation

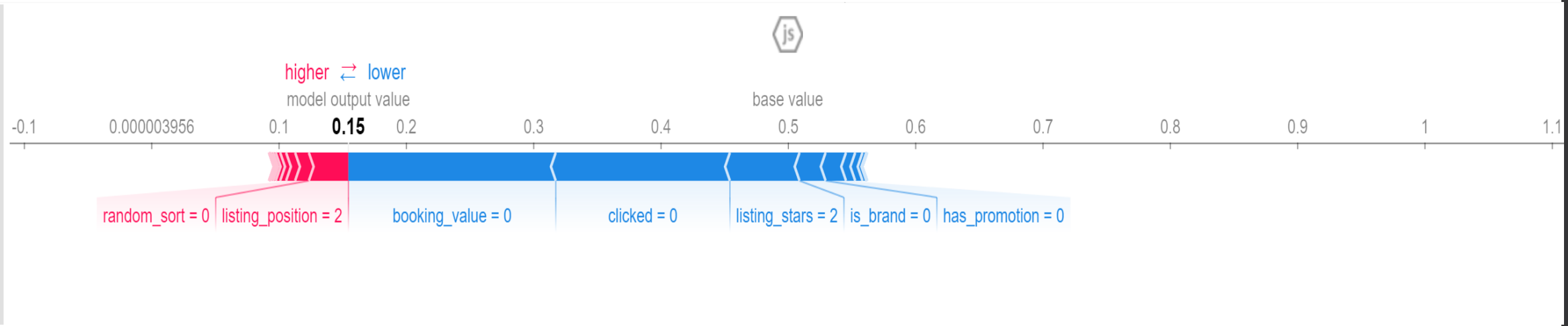
Here we are employing K-fold cross validation; more specifically, 10 folds. We are creating 10 subsets of our data on which to employ the training and test set methodology; then we will average the accuracy for all folds to give us our estimation.
Within a random forest context, if your data set is significantly large, you can choose to not do cross validation and instead use the OOB error rate as an unbiased metric for computational costs. But for the purposes of this tutorial, I included it to show the different accuracy metrics available.

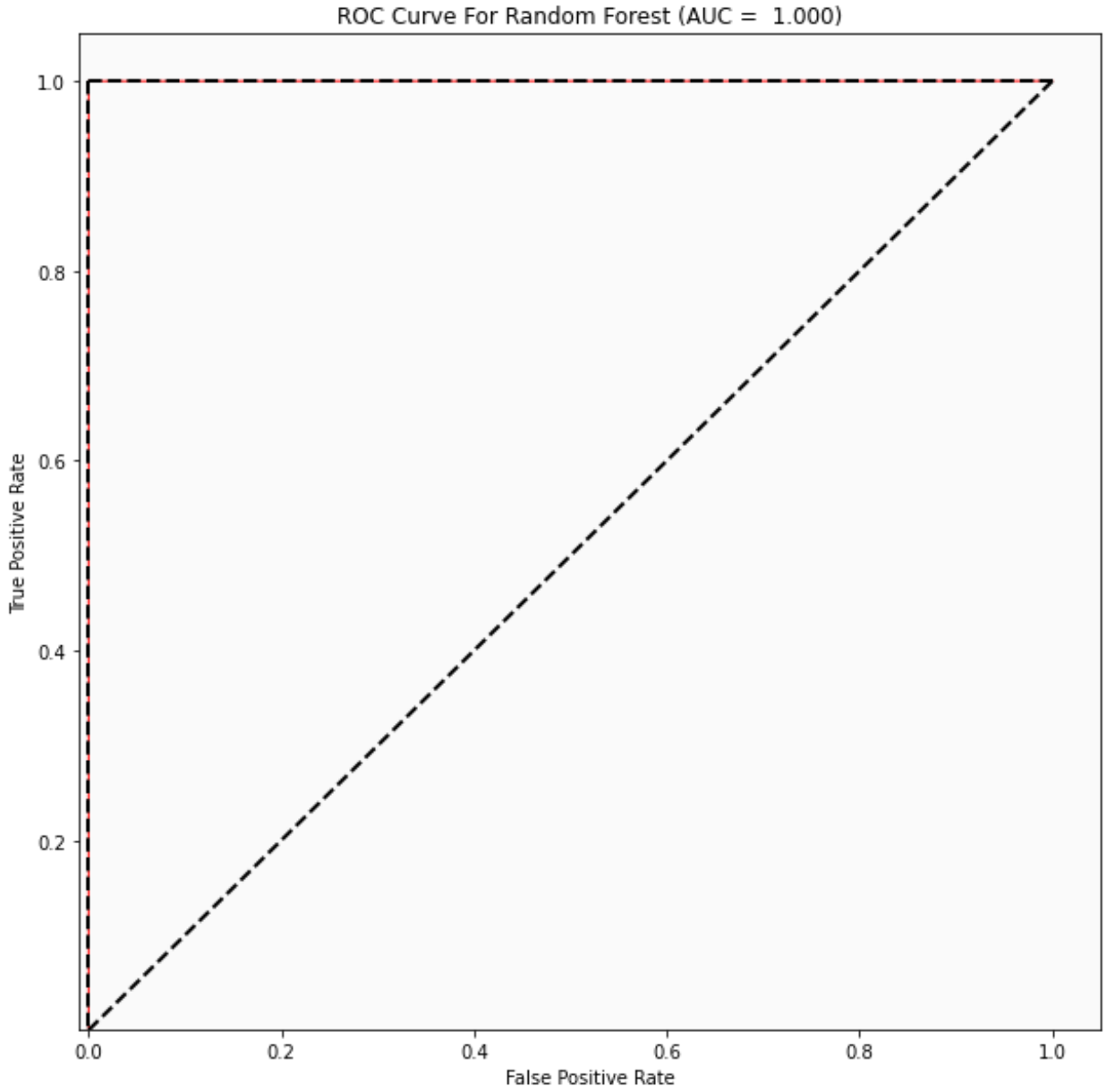
ROC Curve Metrics

- A receiver operating characteristic (ROC) curve calculates the false positive rates and true positive rates across different thresholds. Let's graph these calculations.
- If our curve is located in the top left corner of the plot, that indicates an ideal model; i.e., a false positive rate of 0 and true positive rate of 1. On the other hand, a ROC curve that is at 45 degrees is indicative of a model that is essentially randomly guessing.
- We will also calculate the area under the curve (AUC). The AUC is used as a metric to differentiate the prediction power of the model for patients with cancer and those without it. Typically, a value closer to 1 means that our model was able to differentiate correctly from a random sample of the two target classes of two customers who book and who doesn't book the hostel

Model Interpretation using SHAP:

- SHAP Values (an acronym from SHapley Additive exPlanations) break down a prediction to show the impact of each feature.





Our model did great
AUC = 1

Classification Report

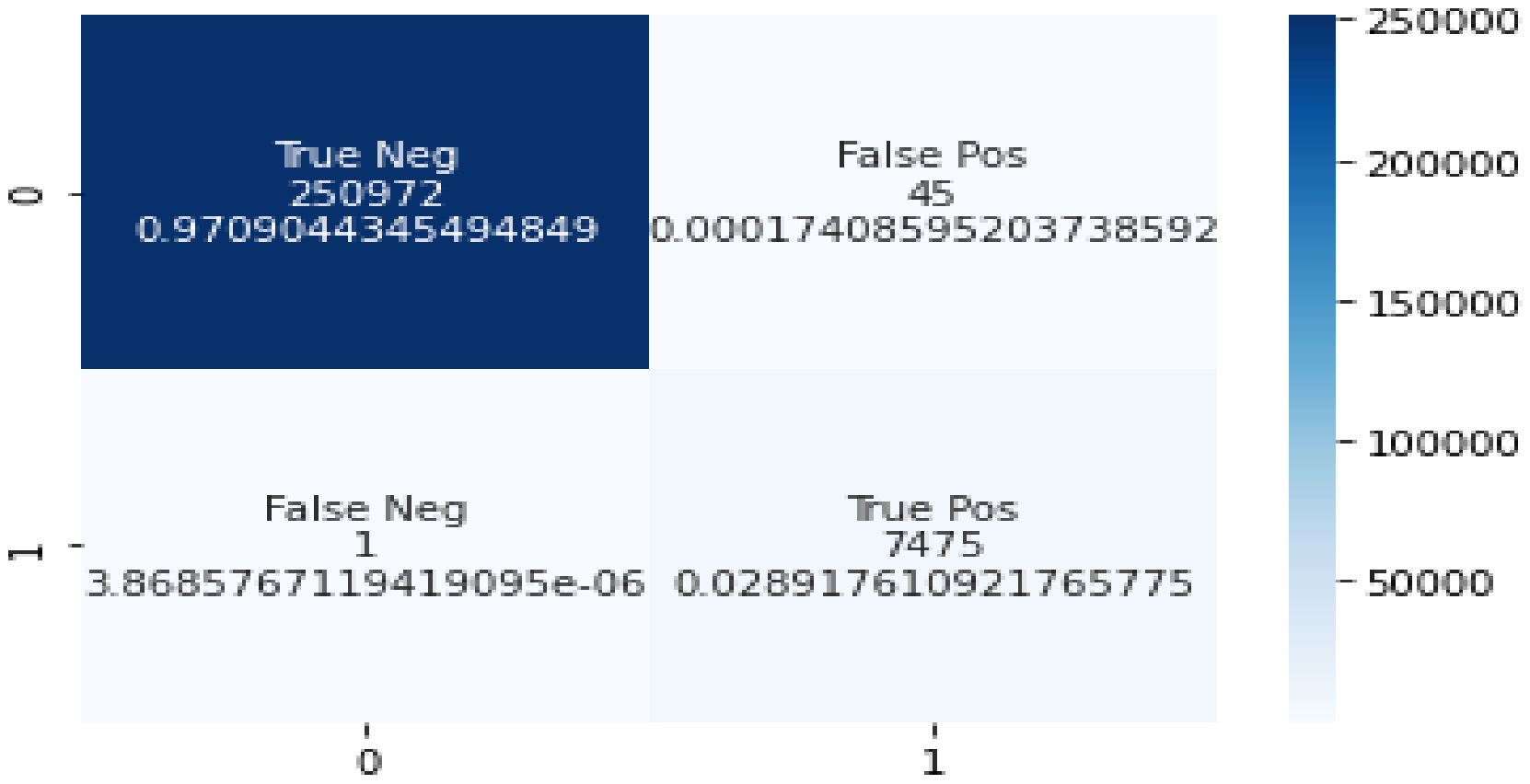
- The classification report is available through sklearn.metrics, this report gives many important classification metrics including:
- Precision: also the positive predictive value, is the number of correct predictions divided by the number of correct predictions plus false positives, so tp/(tp+fp)
- Recall: also known as the sensitivity, is the number of correct predictions divided by the total number of instances so tp/(tp+fn) where fn is the number of false negatives
- f1-score: this is defined as the weighted harmonic mean of both the precision and recall, where the f1-score at 1 is the best value and worst value at 0, as defined by the documentation support: number of instances that are the correct target values

```
Classification Report for Random Forest:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    250973
           1       1.00      0.99      1.00      7520

    accuracy                           1.00    258493
   macro avg       1.00      1.00      1.00    258493
weighted avg       1.00      1.00      1.00    258493
```

Confusion Matrix:



Conclusion:

- We've now gone through a number of metrics to assess the capabilities of our random forest, but there's still much that can be done using background information from the data set. Feature engineering would be a powerful tool for extracting information and moving forward into the research phase, and would help define key metrics to utilize when optimizing model parameters.