Expedia: Hotel Recommendation

Group 9:

Neha Gilson

Aastha Grover

Shuxian Wu

INFO 7390 ADS

Website: team9ans.pgigtupmik.us-west-2.elasticbeanstalk.com

Table of Contents

Expedia: Hotel Recommendation

Problem Statement:

- Expedia has provided logs of customer behavior.
- This data includes hotel clusters based on historical price, customer star ratings, geographical locations relative to city center, etc.
- They are interested in predicting which hotel group a user is going to book.
- Our goal is to predict the booking outcome for a search.

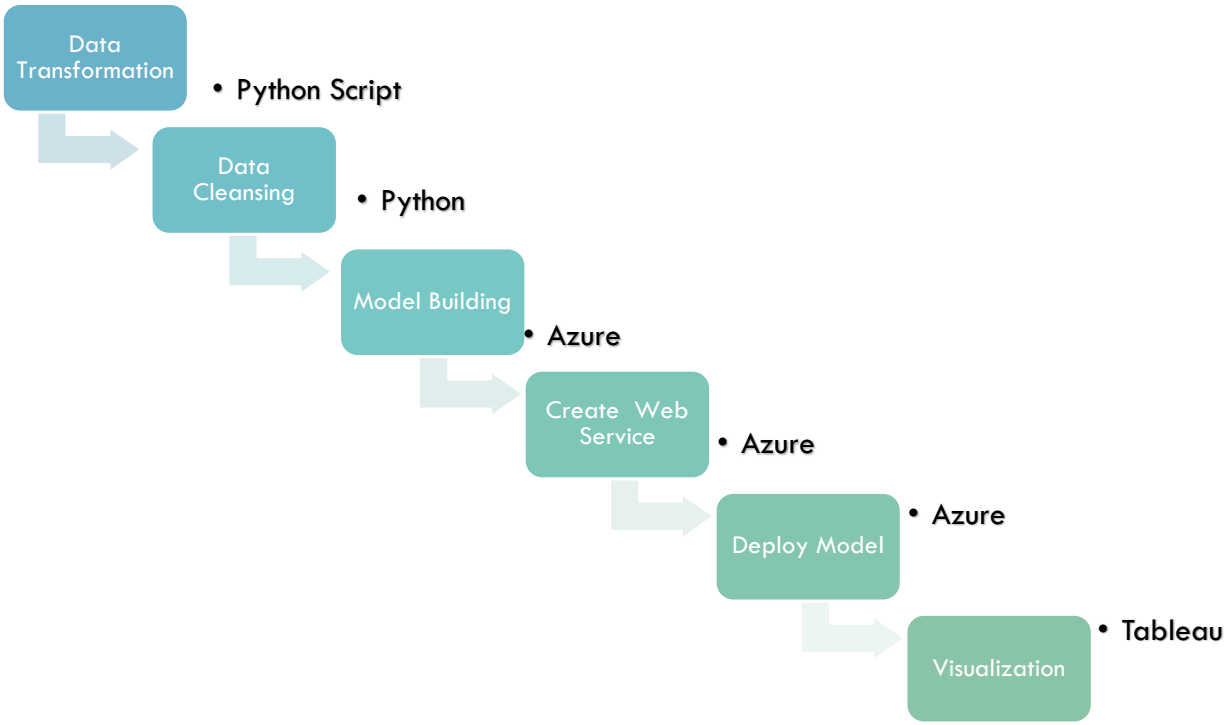
End User:

- Our model can be used by any travel company like Expedia, Trip Advisor to understand travel patterns of their customers.
- They can use it to expand their customer base and market their products and packages based on the results of pattern analysis.

Application use case

- Predict if the search is going to result in a booking. And if so, which cluster will the booked hotel belong to.
- Hotel recommendation for a particular search criteria for the US.
- Study the travel patterns of users of the website.
- Analyze the locations preferred by tourists based on various factors like season, time, vacation etc.

Work Flow:


Data Transformation
• Python Script
Data Cleansing
• Python
Model Building
• Azure
Create Web Service
• Azure
Deploy Model
• Azure
Visualization
• Tableau

Date Cleansing

The data that we had was very big, about 4GB. Below are the column descriptions for the data set.

train/test.csv

Column name	Description	Data type
date_time	Timestamp	string
site_name	ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp, ...)	int
posa_continent	ID of continent associated with site_name	int
user_location_country	The ID of the country the customer is located	int
user_location_region	The ID of the region the customer is located	int
user_location_city	The ID of the city the customer is located	int
orig_destination_distance	Physical distance between a hotel and a customer at the time of search. A null means the distance could not be calculated	double
user_id	ID of user	int
is_mobile	1 when a user connected from a mobile device, 0 otherwise	tinyint
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	int
channel	ID of a marketing channel	int
srch_ci	Checkin date	string
srch_co	Checkout date	string
srch_adults_cnt	The number of adults specified in the hotel room	int

srch_adults_cnt	The number of adults specified in the hotel room	int
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room	int
srch_rm_cnt	The number of hotel rooms specified in the search	int
srch_destination_id	ID of the destination where the hotel search was performed	int
srch_destination_type_id	Type of destination	int
hotel_continent	Hotel continent	int
hotel_country	Hotel country	int
hotel_market	Hotel market	int
is_booking	1 if a booking, 0 if a click	tinyint
cnt	Numer of similar events in the context of the same user session	bigint
hotel_cluster	ID of a hotel cluster	int

It takes very long time to run any code for the original train dataset, and our python notebook, R studio or Azure Machine Learning could not handle the heavy data when we tried to run our analysis. We analyzed the data distribution and found that even when we randomly split the data to get a small subset, the demographics remain roughly the same. So we took 2% of the dataset available for our further analysis and model building. The analysis is present in the python code file: Data Exploration based on raw dataset.

When thinking of preprocessing the dataset, we found only three columns contain null value. Before dealing with missing values, we first dropped columns that are considered having no meanings, such as unnamed column, “user_id” . Additionally, we calculated “adv_times” which is how much time people will book their trips in advance, by subtract searching date from check in time, and “hotel_night” which is how long people needed for each trip, by taking the difference between check in time and check out time. For the columns “srch_ci” and “srch_co”,

the missing value only takes a comparatively low percentage of the total records, we decided to simply drop them. Also, we checked the validation of timestamps, like “srch_ci” should be earlier than “srch_co”, but later than “date_time”. We dropped all the records which are considered outliers or abnormal. The other column with missing value is named “orig_destination_distance”, in which the missing value is as many as about 33% of the total records. In order to avoid overlapping problems, we decided to calculate imputation and fill the missing value with Expectation Maximization Algorithm by using Python module Sklearn. Mixture. GMM.

In the imputation process, we set n_components equal to 20 which means we want our dataset clustering into 20 groups. We split our sample data based on where the data contains missing value or not. Basically we took all the variables except destination distance from complete sample data to fit in our model and labeled each record from 0 to 19. Then we calculated means for destination distance for each group. We labeled and fit the model on our test data which has null value on “destination_distance”. We filled the missing value by merging means for each group based on predicted labels from 0 to 19. Finally, we got our complete sample dataset for modeling.

Based on different purpose, we generated different datasets. We used the data which has “is_booking” only equals to 1 for hotel cluster recommendation purposes, and used the whole dataset to train out binary classification model which is whether a user will book or not based on other features.

Final Dataset after cleaning:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	adv_times	channel	cnt	date_time	hotel_cluste	hotel_contin	hotel_count	hotel_marke	hotel_nights	is_booking	is_mobile	is_package	orig_destina	site_name	srch_adults_	srch_childre	srch_ci	srch_destina	srch_rm_cnt
2	79.3008102	9	1	12/2/14 16:46	71	2	50	682	1	0	0	0	956.5488	2	2	0	##########	1	1
3	20.1605671	9	1	6/17/14 20:08	79	2	50	628	2	1	0	0	208.5084	2	2	1	##########	1	1
4	54.1684838	2	2	11/3/14 19:57	28	2	50	675	6	0	0	0	412.8715	2	2	0	##########	1	1
5	1.93842593	9	1	7/12/14 01:28	10	2	50	700	1	0	1	0	112.9187	2	2	0	##########	1	1
6	20.047037	9	1	3/18/14 22:52	99	2	50	529	5	0	1	0	2416.4157	2	2	1	##########	1	1
7	12.1408449	9	1	10/1/14 20:37	20	4	8	126	4	0	1	1	1053.98	2	2	0	##########	1	1
8	69.0334606	9	1	10/11/14 23:11	91	2	50	682	6	0	0	1	9344.1919	8	2	2	##########	1	1
9	200.590856	9	1	12/30/14 09:49	54	2	50	628	4	0	0	1	1735.2296	2	2	0	##########	1	1
10	1.31188657	9	3	8/30/14 16:30	77	2	50	679	1	0	1	0	390.1352	2	2	2	##########	1	1
11	0.03971065	2	1	11/30/13 23:02	6	2	50	358	1	1	0	0	334.3823	2	2	0	##########	5	1
12	7.39938657	9	1	12/15/14 14:24	95	2	50	637	1	0	0	0	1757.3654	2	2	2	##########	1	1
13	40.2270139	1	1	8/4/14 18:33	28	2	50	697	1	0	0	0	38.82	2	2	0	##########	1	1
14	89.3830671	9	2	5/11/13 14:48	87	4	163	1503	7	0	0	1	899.5435	2	3	0	##########	1	1
15	1.20177083	9	2	10/10/14 19:09	73	2	50	743	1	0	0	0	185.0857	2	2	1	##########	6	1
16	50.8103241	0	1	7/17/14 04:33	18	2	50	637	2	0	1	0	385.2045	2	2	0	##########	4	1
17	9.06	0	1	7/5/14 22:33	82	4	196	1987	1	1	0	0	1351.3148	2	1	0	##########	1	1
18	30.8857639	9	1	10/6/13 02:44	23	2	50	663	5	0	0	0	6780.8342	2	1	0	##########	6	1
19	3.74315972	3	1	12/4/14 06:09	2	2	50	656	1	1	1	0	43.7367	2	2	0	##########	1	1
20	74.6939815	9	1	4/12/13 07:20	48	2	50	425	1	1	0	0	766.0087	2	2	0	##########	1	1
21	62.0653241	9	1	8/19/14 22:25	83	2	50	1457	4	0	0	0	911.8797	2	2	0	##########	6	1
22	280.34316	9	1	12/14/14 15:45	26	2	50	213	7	0	1	1	2374.7023	2	2	0	##########	5	1
23	10.58125	9	1	7/9/14 10:03	32	2	50	440	4	0	0	0	246.2289	2	1	0	##########	4	1
24	30.2457755	9	1	11/29/14 18:06	76	2	50	366	4	0	0	0	2579.3209	2	2	1	##########	6	1
25	3.58266204	9	1	9/18/13 10:00	11	6	144	4	2	0	1	0	5313.266	2	1	0	##########	1	1
26	188.448449	9	3	6/24/13 13:14	10	2	50	1457	1	0	0	0	4986.3952	2	3	0	##########	6	1
27	1.58813657	9	5	10/17/13 09:53	21	2	198	750	1	0	1	0	307.6166	2	2	1	##########	6	1
28	29.3752778	9	1	6/10/14 14:59	11	6	7	8	1	0	0	0	3597.227	11	1	0	##########	1	1
29	59.6446528	9	1	9/24/14 08:31	48	2	50	718	3	0	0	0	52.6419	2	2	0	##########	6	1
30	23.0792361	9	2	5/27/14 22:05	83	2	50	355	1	0	0	0	109.0067	2	2	0	##########	6	1
31	28.3687731	1	1	4/13/14 15:08	55	2	50	1457	1	0	0	0	1329.3693	2	2	0	##########	6	1
32	39.5645139	1	1	9/29/13 10:27	89	2	50	623	1	1	0	0	50.7574	2	2	0	##########	3	1

Keywords: Python 3, Data Cleansing, Pandas, Numpy.

Data Visualization

We used Tableau to perform data visualization and Tableau Public to share this visualization. Below are our observations.

Link to tableau Public:

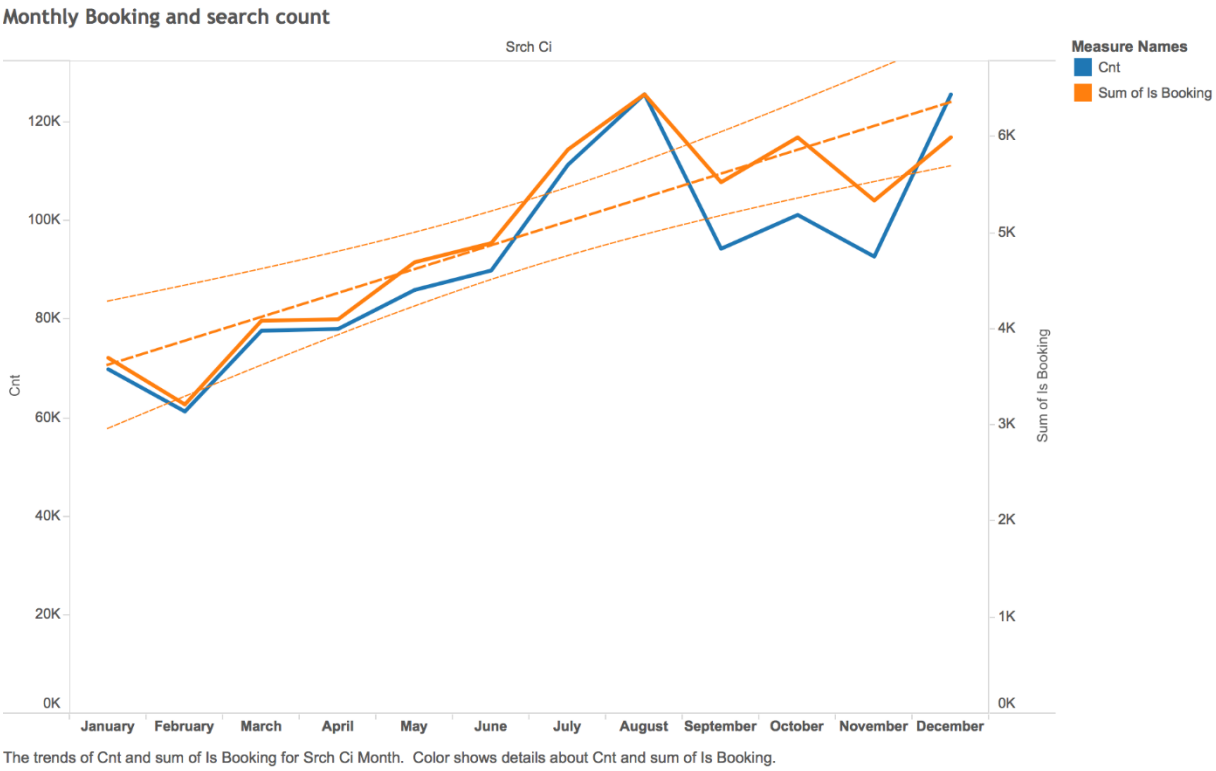
1. Link to view the pattern of locations preferred by users based on multiple criteria. This link shows a video that takes us through monthly travel pattern of people with different search criteria like months, number of children, destination location etc. Since the dataset remains anonymous about the actual physical location of the destination it is impossible to show a geographical representation.

https://public.tableau.com/views/hist_loc_pref/LocationPreference?:embed=y&:display_count=yes&:showTabs=y

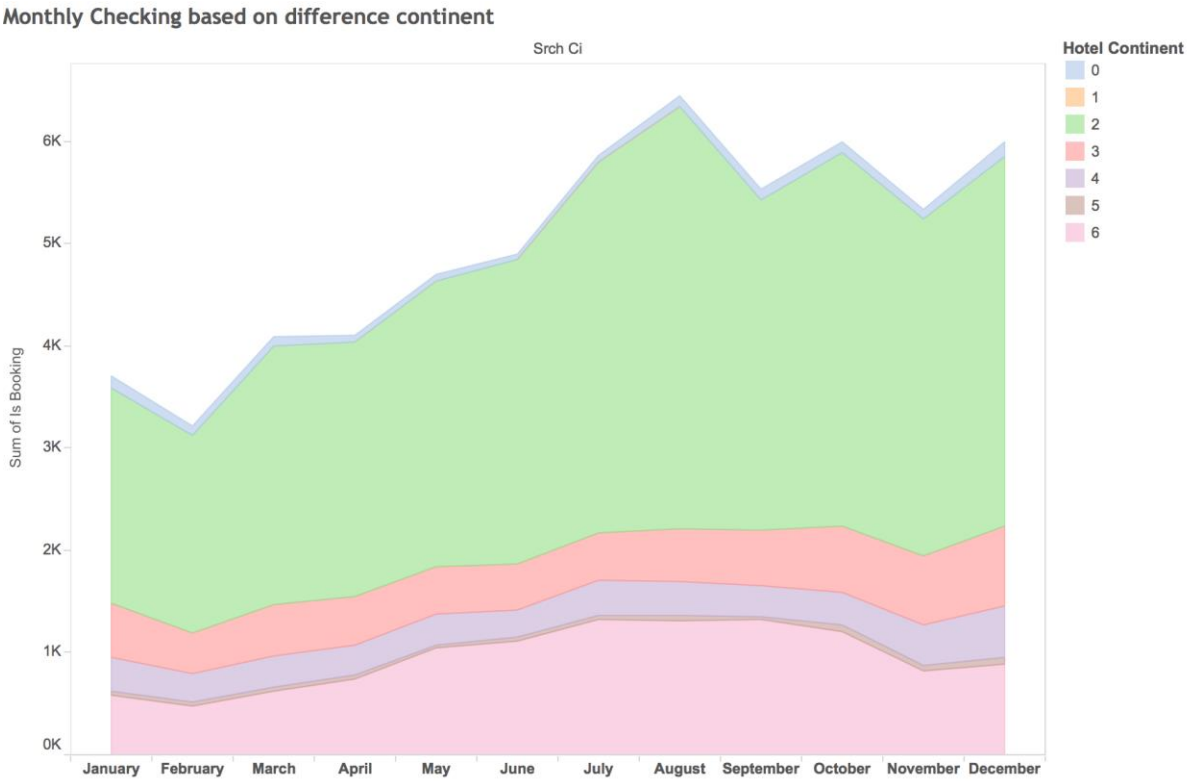
2. The below link shows a dashboard of out exploratory data analysis using Tableau.

   https://public.tableau.com/shared/336R25MZZ?:display_count=yes

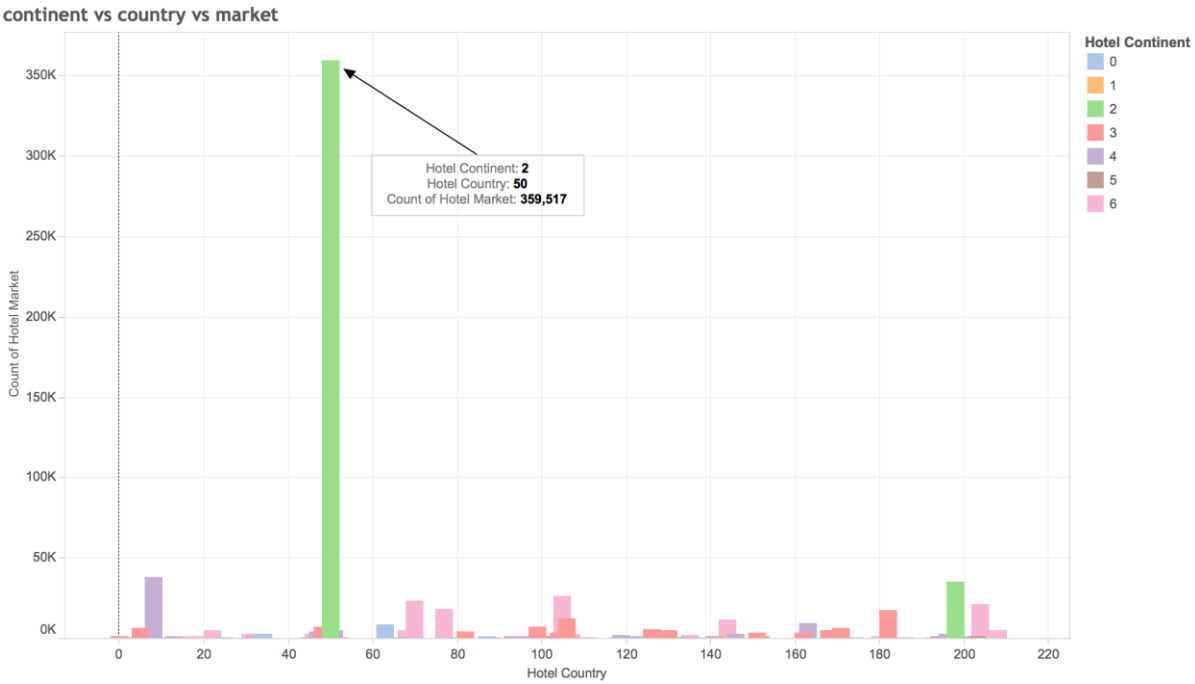
   Below are the brief descriptions of our findings:



Tracking monthly bookings and how many similar events in the context of the same user session for each month. For summer time, both of the numbers increased a lot. We think that it because of good weather conditions and also because many students finished school and go for travels. In overall, monthly sum of bookings are higher than the sum of Cnt.



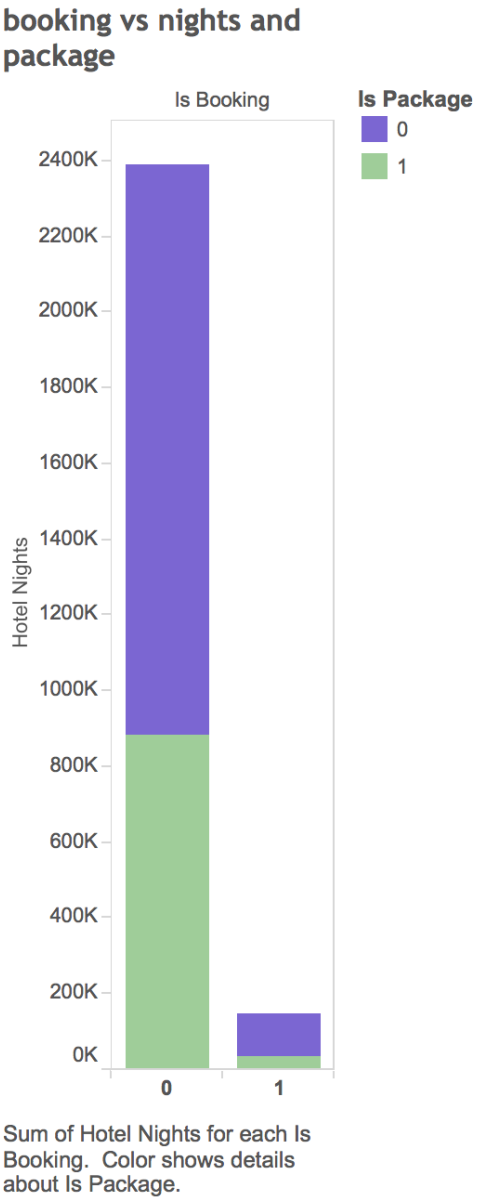
Continent 2 has the most bookings regardless seasons and months, which we draw a conclusion that countries or destinations in continent 2 are the biggest market for Expedia when compared to other continents. Continent 0 does not have any booking record at all, which we think might be Antarctica, since it's not a common destination for travel.



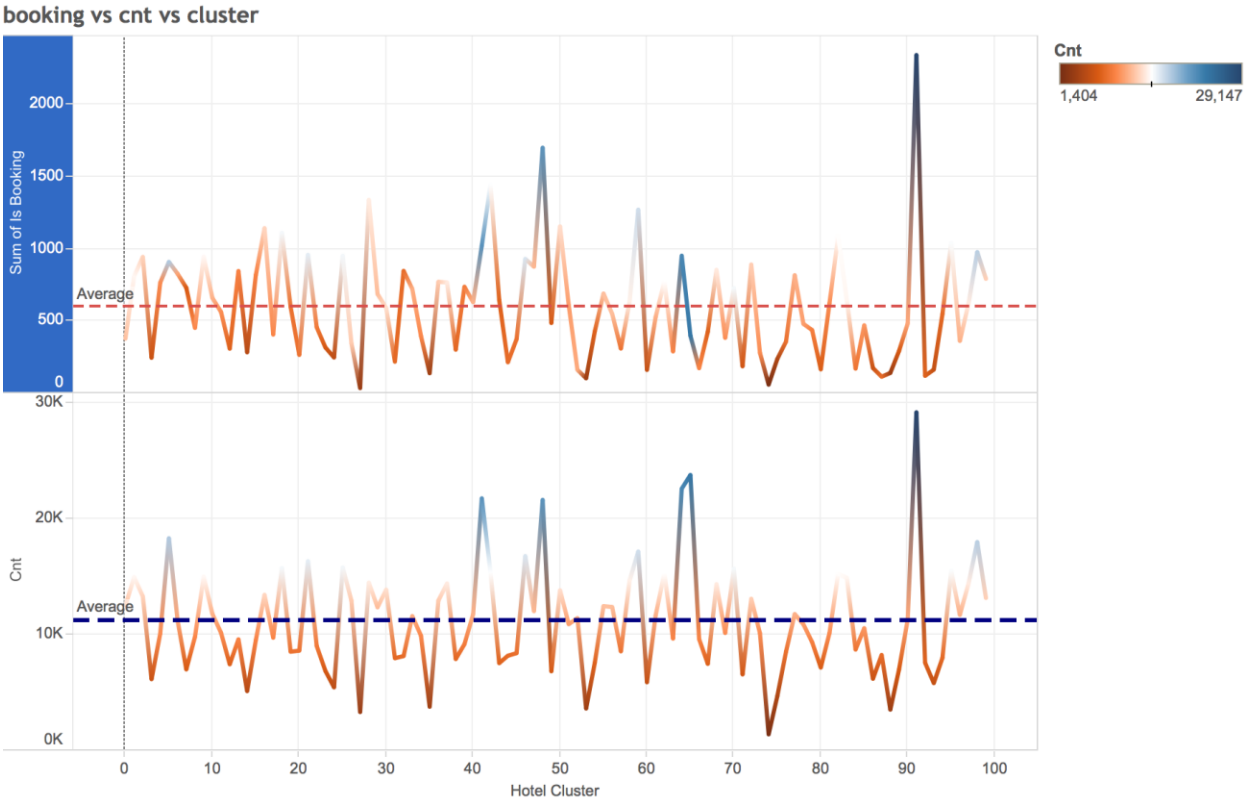
The plot of count of Hotel Market for Hotel Country. Color shows details about Hotel Continent.

Besides previous picture, this gives more details for that the country 50 is the most attractive destination for the customers of Expedia. And so we know that country 50 in continent 2 has the highest business value .

We compared

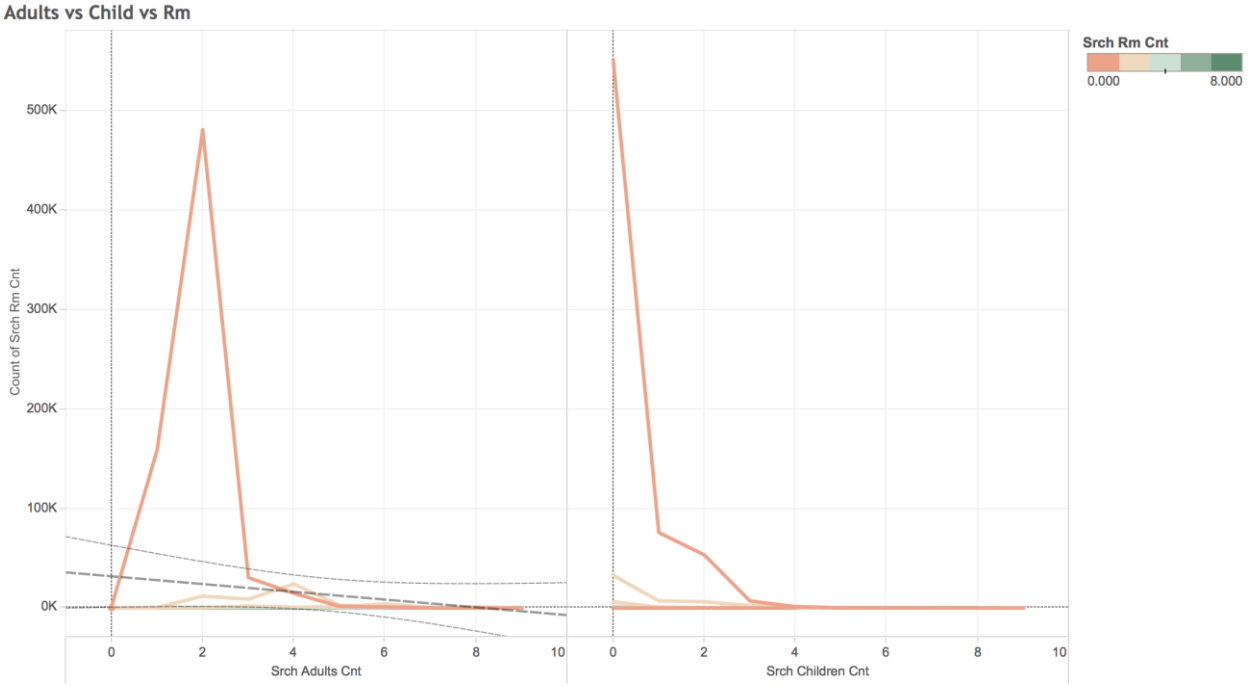


Here we see that most searches on Expedia don't result in a booking. So most of the traffic is just browsing. We also see that even when the booking is successful, only about 30% take up the offered packages. This directs our attention to various business questions about investing in the travel packages offered and target segment of travelers and vacationers.

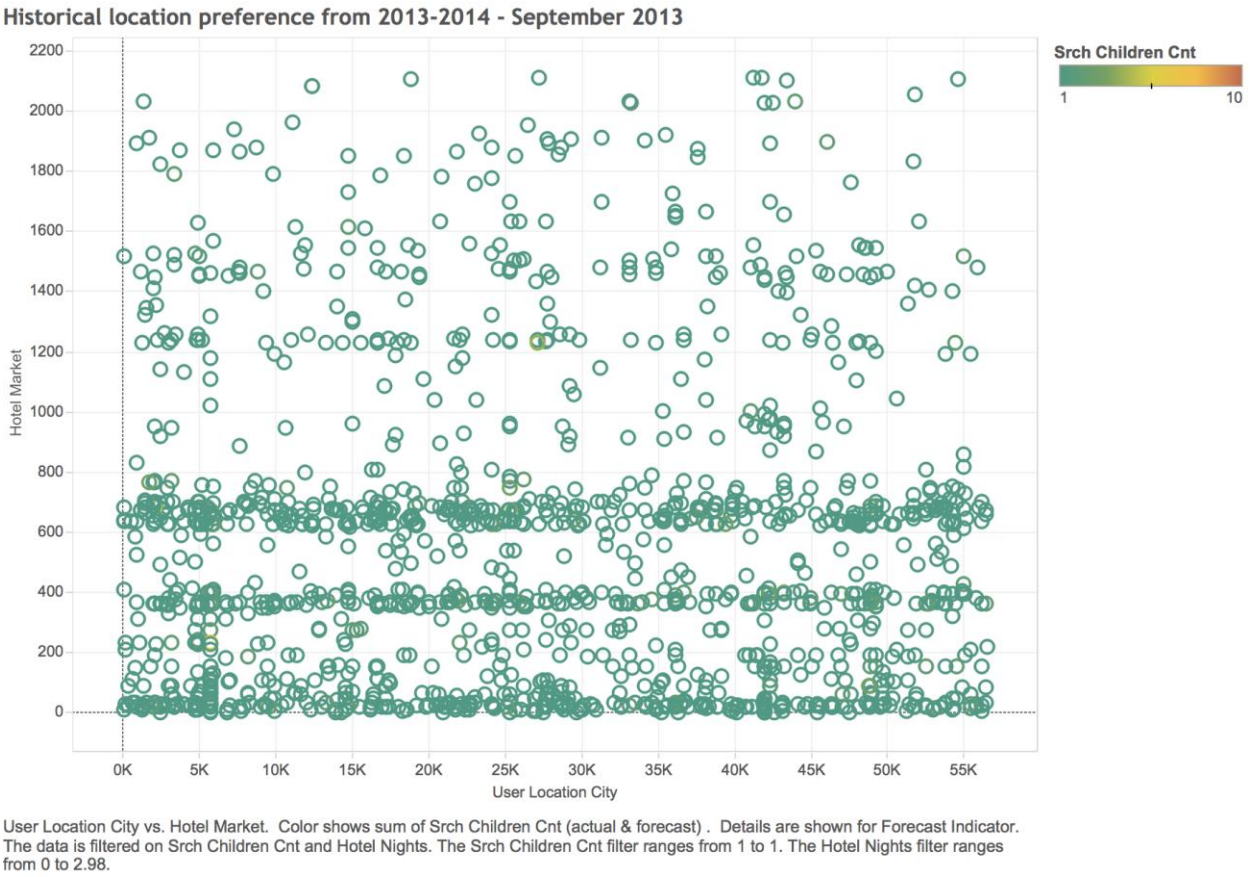


The trends of sum of Is Booking and sum of Cnt for Hotel Cluster. Color shows sum of Cnt. The data is filtered on Is Booking, which keeps 0 and 1.

This graph shows the relationship between cnt and booking variables. Since they tend to change in a very similar pattern we used one of the 2 variables in our model.



The first graph here shows the relation between the number of adults travelling and the number of rooms booked. We see that most of the searches are for 1 or 2 adults. So they could either be singles or families with kids. Similarly in the case of number of children most of them are for 0 to 3 children. These visualizations helped us to determine whether the adult count, children count are relevant to our model. We know that Expedia has their own clustering algorithm for hotels which include these factors.



This graph shows a destination location preference by users of expedia based on different factors. The tableau public link is more interactive to make more observations.

Model

1.1.Hotel Cluster Recommendation:

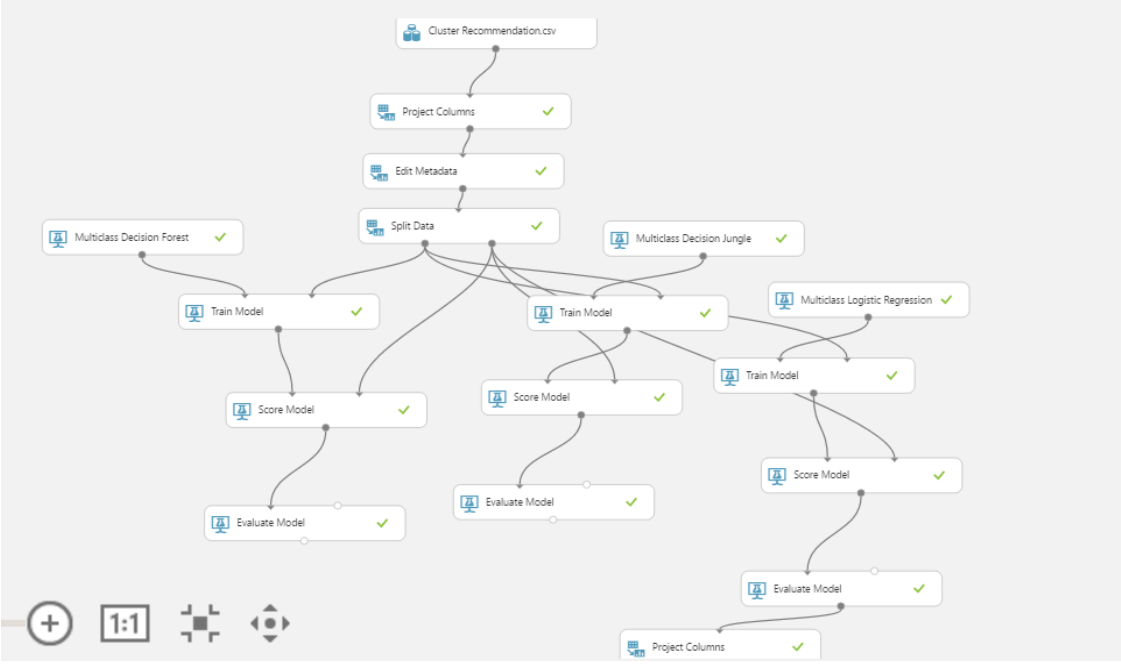
This model can recommend the hotel clusters based on various input variables as per the feature section table below.

Feature Selection

Column Name	Hotel Cluster Classification
Date-Time	We used this to calculate the difference between Search Date & Search Check in date i.e. (Advance Times)
Site-name	We ignored this variable as the site ID does not add value to our model
User-City	User City is included in the model as it encompasses the value for User's city, country, region and site.
User-Country	User City variable will help to specify which country the user belongs to
User-Region	Ignored this variable as it is correlated to user city
Channel	This variable is not relevant to our model as it does not help to predict the hotel cluster.
Count	We used this since it gives the trend about what other users are searching for in the same user session's context
Is_booking	Since the hotel cluster recommendation is required only for booked hotels, we considered only those where booking is 1
Is_Mobile	It does not matter if the user used a mobile device or not for us to recommend the cluster, so we ignored this column
Is_Package	Yes
Hotel-Market	This gives the least granularity for the hotel searched for so we included this in out model
Hotel-Continent	No
Hotel-Country	We ignored this since the Hotel Market gives us the necessary information regarding hotel.
Hotel-Night	Yes, the number of nights of stay in the hotel is calculated to recommend the cluster
Advance-Times	This gives the time between the search and actual check-in date.
Search-destination-id	Since it's an ID column which will not add any value to the model, we dropped it.
Search-child-count	Since this affects the search and the type of hotel a user might chose, we included this in the model

Search-check-in	This date affects the availability, type, location etc. of the hotel a user might chose, so we included it in the model
Search-room-count	Yes
Search-Adult-Count	Number of rooms and the count of adults are highly correlated since as the number of adults increase so does the room count so we ignored this column
Hotel-Cluster	This is the Y variable or the recommendation value.

We used multiple algorithms like Decision tree forest, Decision Jungle, Neural Network Regression and Multiclass logistic regression algorithms. Upon comparing the results of the models, we found that **Multiclass Logistic Regression** gave us the best results, with 12.8% accuracy. It is good since there are 100 columns and each one would have a probability of 0.01%. But with the model the accuracy is raised to 12.8%.



The error metrics are as shown below:

Decision Forest:

Metrics

Overall accuracy	0.091724
Average accuracy	0.981834
Micro-averaged precision	0.091724
Macro-averaged precision	NaN
Micro-averaged recall	0.091724
Macro-averaged recall	0.07063

Decision Jungle:

Metrics

Overall accuracy	0.081872
Average accuracy	0.981637
Micro-averaged precision	0.081872
Macro-averaged precision	NaN
Micro-averaged recall	0.081872
Macro-averaged recall	0.060669

Logistic Regression:

Metrics

Overall accuracy	0.128068
Average accuracy	0.982561
Micro-averaged precision	0.128068
Macro-averaged precision	NaN
Micro-averaged recall	0.128068
Macro-averaged recall	0.110975

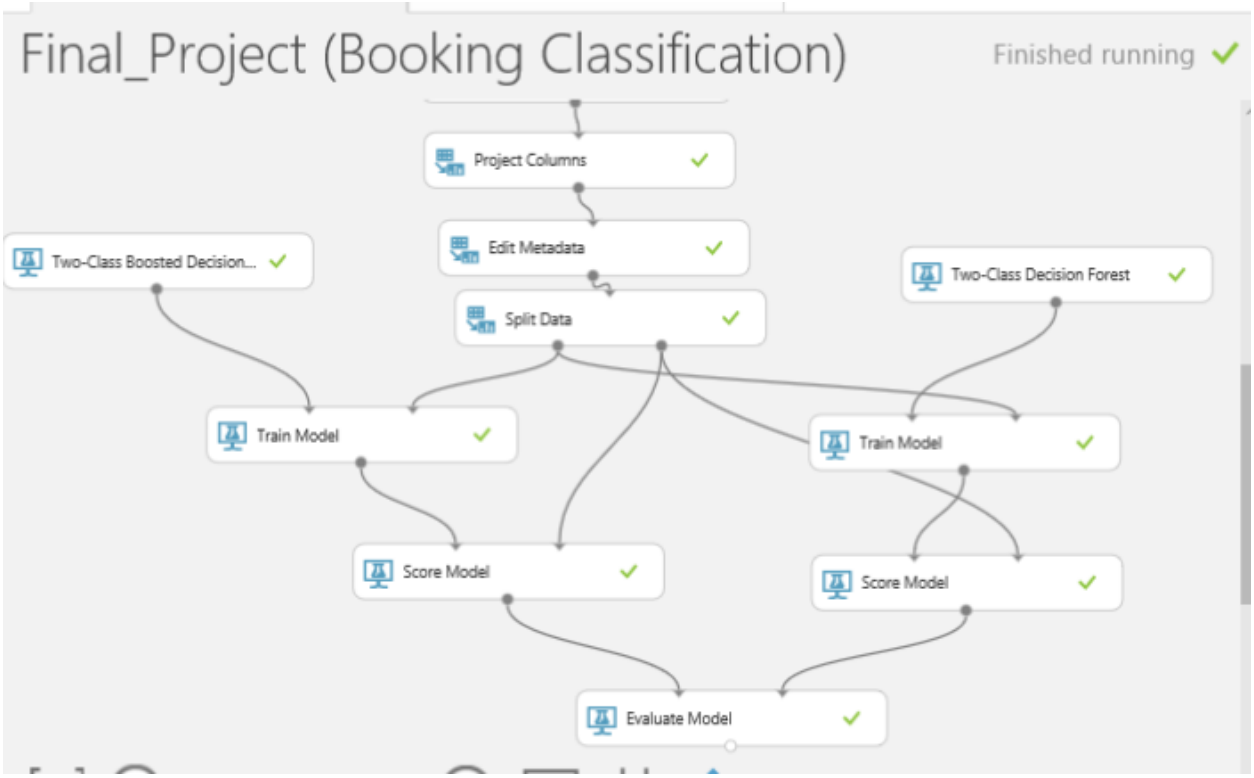
1.2. Booking prediction

Feature Selection

Column Name	Hotel Cluster Classification
Date-Time	We used this to calculate the difference between Search Date & Search Check in date i.e. (Advance Times)
Site-name	We ignored this variable as the site ID does not add value to our model
User-City	User City is included in the model as it encompasses the value for User's city, country, region and site.
User-Country	User City variable will help to specify which country the user belongs to
User-Region	Ignored this variable as it is correlated to user city
Channel	This variable is not relevant to our model as it does not help to predict the hotel cluster.
Count	We used this since it gives the trend about what other users are searching for in the same user session's context
Is_booking	Output Y variable
Is_Mobile	It does not matter if the user used a mobile device or not for us to recommend the cluster, so we ignored this column
Is_Package	Yes
Hotel-Market	This gives the least granularity for the hotel searched for so we included this in out model
Hotel-Continent	No
Hotel-Country	We ignored this since the Hotel Market gives us the necessary information regarding hotel.
Hotel-Night	Yes, the number of nights of stay in the hotel is calculated to recommend the cluster
Advance-Times	This gives the time between the search and actual check-in date.
Search-destination-id	Since it's an ID column which will not add any value to the model, we dropped it.
Search-child-count	Since this affects the search and the type of hotel a user might chose, we included this in the model
Search-check-in	This will not affect if the booking is done or not. Also we have the
Search-room-count	Yes
Search-Adult-Count	Number of rooms and the count of adults are highly correlated since as the number of adults increase so does the room count so we ignored this column
Hotel-Cluster	This is the Y variable or the recommendation value.

Azure ML Model:

The booking model predicts whether a particular search will result in a booking or not. The data set that we got had most of the data as is_booking = 0, which meant that the model would be very good to predict if the search would result in not booking. Using the above features we ran the Two-Class Boosted Decision tree algorithm on Azure ML as shown below.



As expected the confusion matrix showed a better accuracy for True Negative values than for True Positives. Azure ML did not provide the facility to change the cut off probability of the success class which is why it was difficult to achieve a better accuracy than this. The Area Under The Curve for ROC came up to be 0.862 and we concluded that this was the most optimum classification model for classification of Booking.

Confusion Matrix and Accuracy

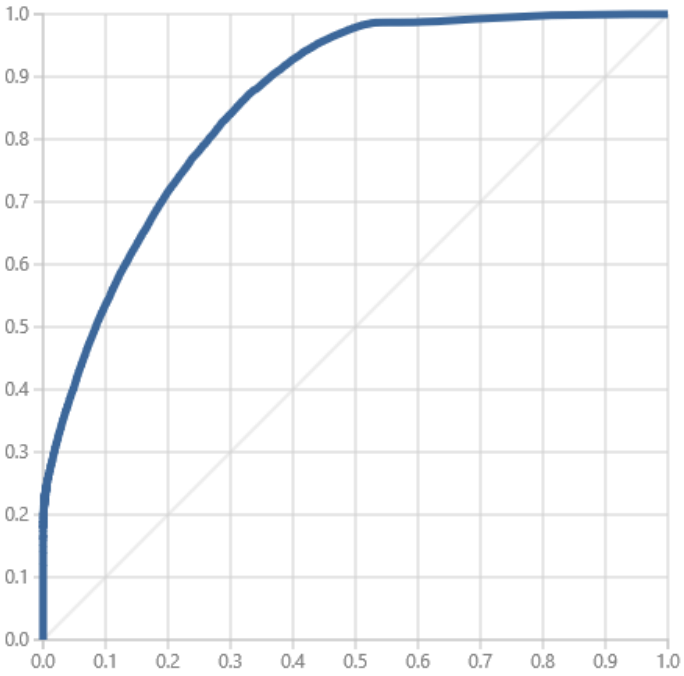
Final_Project (Booking Classification) › Evaluate Model › Evaluation results

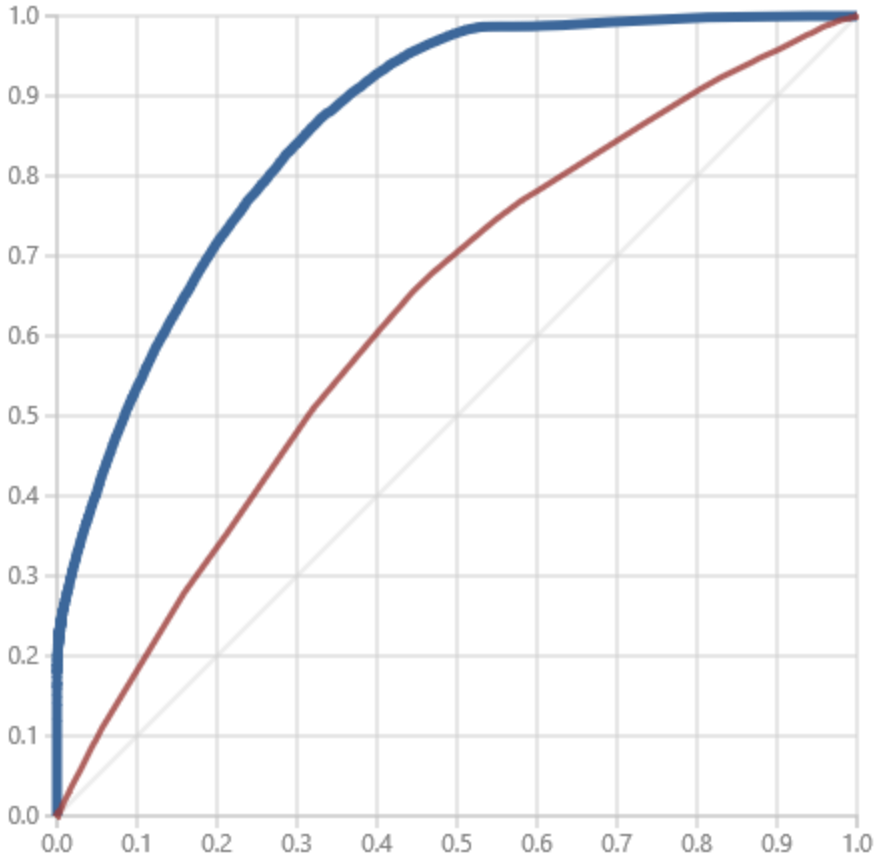
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
4117	13659	0.937	0.861	0.5	0.862

False Positive	True Negative	Recall	F1 Score
667	207293	0.232	0.365

Positive Label	Negative Label
1	0

ROC Curve



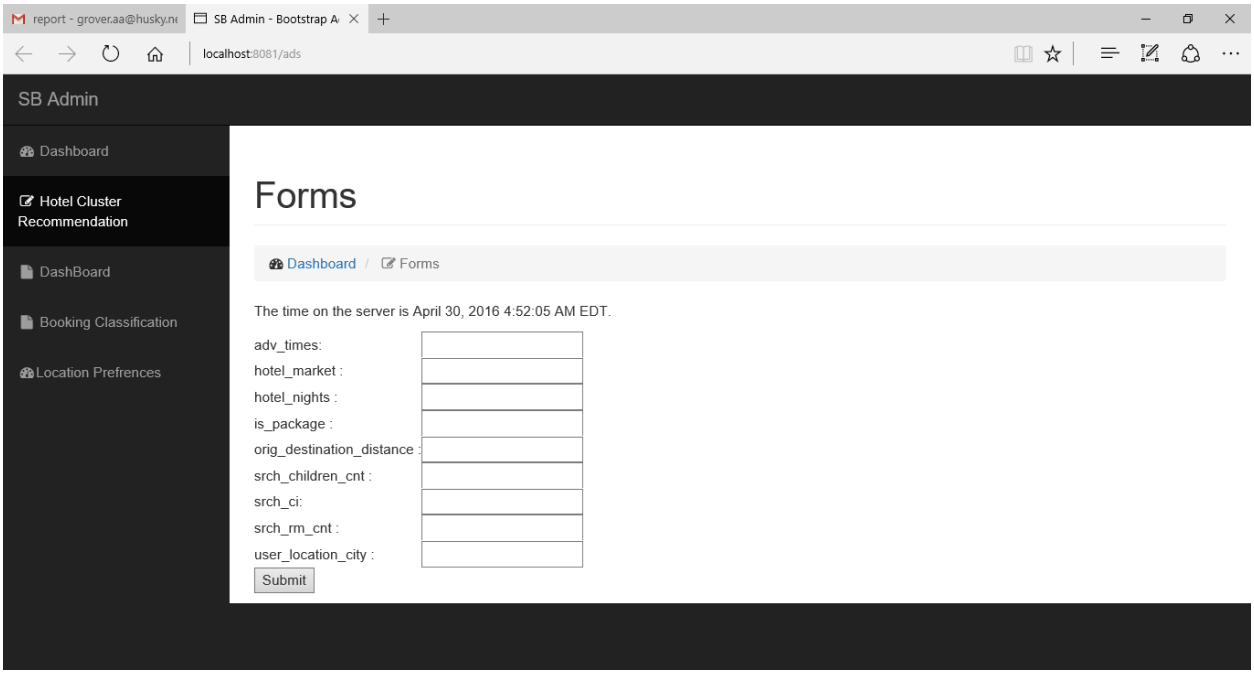


ROC Curve comparison between Boosted Decision Tree (Blue) and Two-Class Decision Forest (Red). The above graph clearly shows that the Boosted Decision tree gives a better AUC value.

We then deployed our model and generated the URL and API key for integration.

The Application Link:

team9ans.pgigtupmik.us-west-2.elasticbeanstalk.com



Link	Redirects
Dashboard	Data Visualization on Tableau Public
Hotel Cluster Recommendation	Classification Model for recommending Hotel Cluster
Location Preference	Tableau Visualization for seasonal trends of users
Booking Classification	Classification Model for predicting the result of a search

References

Kaggle Competition: https://www.kaggle.com/competitions

Microsoft Azure Machine Learning Studio