Amplify Analytix Recruitment Task

Powell Menezes, Great Lakes Institute, Bangalore, KA, IND

1. Introduction

With hundreds or even thousands, of travel agencies to choose from at every destination, it is difficult to know which will suit your personal preferences. Travel agency wants to provide personalized hotel recommendations to their users. This is no small task for a site with hundreds of millions of visitors every month. In this project challenges have been taken to contextualize customer data and predict the likelihood of a user who will choose to stay at different hotel groups.

The objective is to build a machine learning model which clusters and recommends the hotel for a new search event. Analysis of hotels on clicked/booked counts based on their search and other attributes associated with that user event. As part of this project following are the algorithms implemented:

1. K Means Clustering
2. Logistic Regression
3. Naive Bayes
4. Decision Trees
5. Random Forest

2. The Dataset

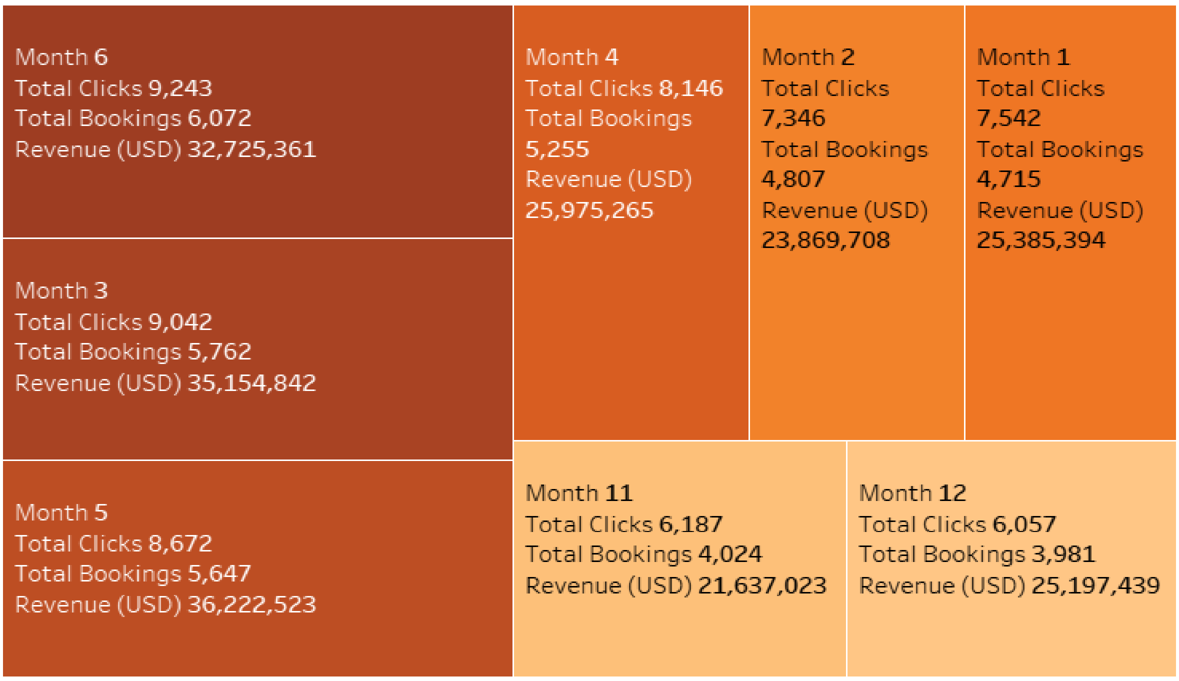
The dataset 23,80,557 records and contain 58 features.

Feature Name	Feature Type	count
'timestamp'	DataTime	1
'search_id', 'site_id', 'user_country_id', 'destination_id', 'listing_country_id', 'listing_id'	Nominal	6
'listing_stars', 'listing_review_score', 'listing_position', 'length_of_stay', 'num_adults', 'num_kids', 'num_rooms', 'booking_window',	categorical(Ordinal)	8
'user_hist_stars', 'user_hist_paid', 'location_score1', 'location_score2', 'price_usd', 'log_historical_price', 'distance_to_dest', 'log_click_proportion', 'booking_value'	Continious	9
'has_promotion', 'is_brand', 'stay_on_saturday', 'competitor1_rate', 'competitor6_rate', 'competitor1_has_availability', 'competitor1_price_percent_diff', 'competitor2_rate', 'competitor2_has_availability', 'competitor2_price_percent_diff', 'competitor3_rate', 'competitor3_has_availability', 'competitor4_rate', 'competitor7_has_availability', 'competitor3_price_percent_diff', 'booked', 'random_sort', 'competitor8_price_percent_diff', 'competitor4_has_availability', 'competitor4_price_percent_diff', 'clicked', 'competitor5_rate', 'competitor5_has_availability', 'competitor8_has_availability', 'competitor5_price_percent_diff', 'competitor6_price_percent_diff', 'competitor8_rate', 'competitor6_has_availability', 'competitor7_price_percent_diff', 'competitor7_rate'	categorical(Nomina)	29

3. <u>Data Cleaning</u>

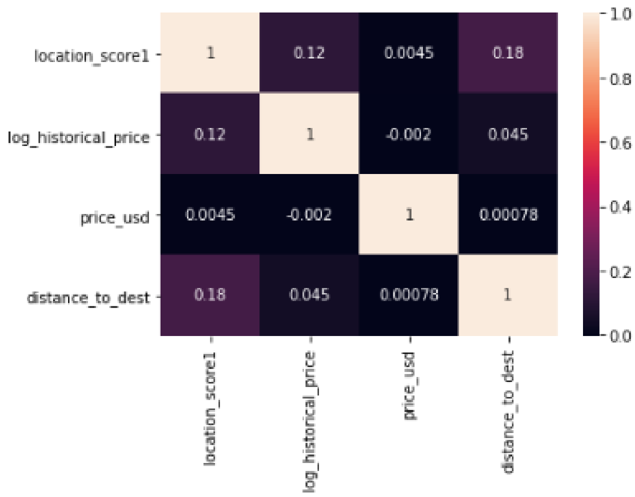
- The data set consists of null or missing values, observe that 20 features contain more than 80% null and 8 features more than 50% null values and those columns do not contribute in the prediction.
- Check for the duplicates and special symbols. Create calculated field wherever required
- Three feature contains 32, 22 and 0.14 % of null which are replaced with mean/median
- Used **Winsorize** method to cap the outliers and **log transformation** to reduce the outlier effect on the model for continuous variables
- Then the final process of data cleaning is **bucketing** that is combing the values and putting them into ranges to ease the process of further analyses.

4. <u>Exploratory data analysis and feature significance/selection</u>



- Used **Correlation Heatmap** to Check the multi collinearity for continuous features and verified the significance using **VIF**
- Used **Chi-Square** to find significance of categorical features

	VIF Factor	Features
0	3.542094	location_score1
1	3.362598	log_historical_price
3	1.657433	distance_to_dest
2	1.000178	price_usd

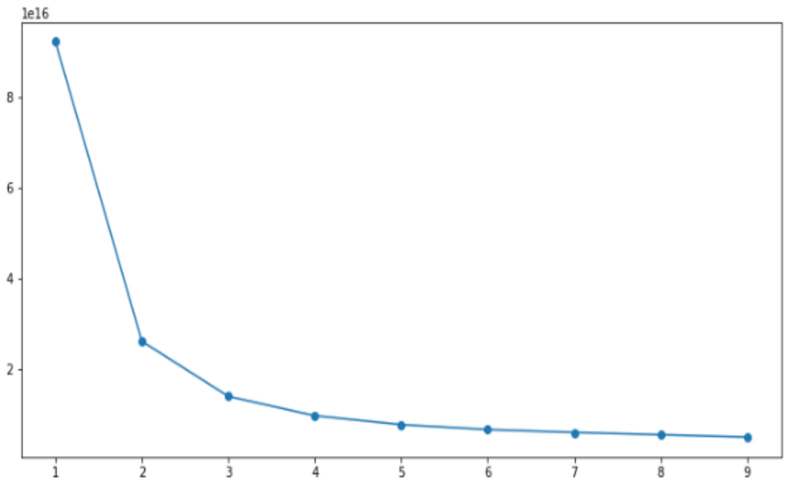


```
feature: location_score2 is significant  and the pvalue = 0.0
feature: listing_review_score is significant  and the pvalue = 0.0
feature: month is not significant  and the pvalue = 0.12335328976385629 -----------
feature: day is not significant  and the pvalue = 0.999888925331182 ---------------
feature: hour is significant  and the pvalue = 1.2950248009861857e-07
feature: Year is not significant  and the pvalue = 0.9999997458018945 -------------
feature: minute is not significant  and the pvalue = 0.9842144583650566 -----------
feature: booking_window is not significant  and the pvalue = 0.9942689699433563 ---
feature: is_brand is not significant  and the pvalue = 0.28250672633528984 --------
feature: has_promotion is significant  and the pvalue = 0.0
feature: length_of_stay is significant  and the pvalue = 0.026209496985494712
feature: num_adults is significant  and the pvalue = 1.4579773216281103e-11
feature: num_kids is significant  and the pvalue = 1.5671165652185598e-24
feature: num_rooms is significant  and the pvalue = 1.8991095774154633e-40
feature: stay_on_saturday is not significant  and the pvalue = 0.09595925956792684
feature: random_sort is significant  and the pvalue = 2.803035785052373e-13
feature: booked is significant  and the pvalue = 0.0
```

5. Clustering

K-Mean clustering was implemented to cluster the hotels based on the attributes associated with that user, whenever a user clicks the hotel the model will identify to which cluster customer belong and based on that cluster data recommendation will be done, this can reduce the cost and time.

	num_clusters	cluster_errors
0	1	9.217872e+16
1	2	2.601348e+16
2	3	1.384868e+16
3	4	9.580687e+15
4	5	7.604284e+15
5	6	6.525497e+15
6	7	5.882963e+15
7	8	5.378411e+15
8	9	4.860070e+15



- With reference to cluster errors and elbow plot, selected K value as 5 and based on K value data was clustered and cluster shapes are mentioned below

```
(df0.shape,df1.shape,df2.shape,df3.shape,df4.shape)
((479540, 28), (470534, 28), (488172, 28), (470342, 28), (471969, 28))
```

- Whenever user enters his data the customer will be clustered to one of the existing clusters. This can be done using **joblib library**
- Once the cluster is known, recommendation can be done with respect to the particular cluster data instead of using entire data.

5. Model building

- A model is built to predict if a customer will Click on a hotel or not for 5 different clusters
- Since the data is imbalance (9.5:0.5), **SMOTENC** is used for balancing the training data
- Data is Splitted into train (70%) and test (30%) using train_test_split function
- Built model using **Logistic Regression, Naïve Bayes, Decision Tree** and **Random forest** with **K-Fold** validation by taking value of K=5. and The result of various models are summarised in the below tables.
- **Accuracy, precision, recall** and **AUC** is used for performance measures.

MODEL	CLUSTER	PRECISION		RECALL		F1-SCORE		ACCURACY	AUC
		0	1	0	1	0	1		
LOGISTIC REGRESSION	0	0.98	1	1	0.63	0.99	0.77	0.98	0.810
	1	0.99	0.19	0.85	0.75	0.91	0.30	0.84	0.800
	2	0.98	1	1	0.62	0.99	0.77	0.98	0.810
	3	0.99	0.21	0.87	0.74	0.92	0.33	0.86	0.804
	4	0.99	0.19	0.85	0.76	0.92	0.31	0.85	0.800
DECISION TREE	0	0.98	0.59	0.98	0.64	0.98	0.62	0.96	0.801
	1	0.98	0.35	0.94	0.67	0.96	0.46	0.93	0.803
	2	0.98	0.34	0.94	0.66	0.96	0.45	0.93	0.798
	3	0.98	0.36	0.94	0.67	0.96	0.47	0.93	0.806
	4	0.98	0.38	0.95	0.67	0.97	0.49	0.94	0.808
NAÏVE BAYES	0	0.98	0.99	1	0.63	0.99	0.77	0.98	0.810
	1	0.98	1	1	0.62	0.99	0.77	0.98	0.811
	2	0.98	1	1	0.62	0.99	0.77	0.98	0.810
	3	0.98	1	1	0.62	0.99	0.77	0.98	0.812
	4	0.98	1	1	0.63	0.99	0.77	0.98	0.814
RANDOM FOREST	0	0.98	0.99	1	0.66	0.99	0.79	0.98	0.815
	1	0.98	0.54	0.97	0.65	0.98	0.59	0.96	0.812
	2	0.98	0.58	0.98	0.63	0.98	0.60	0.96	0.816
	3	0.98	0.63	0.98	0.64	0.98	0.64	0.97	0.812
	4	0.98	0.56	0.98	0.65	0.98	0.60	0.96	0.813

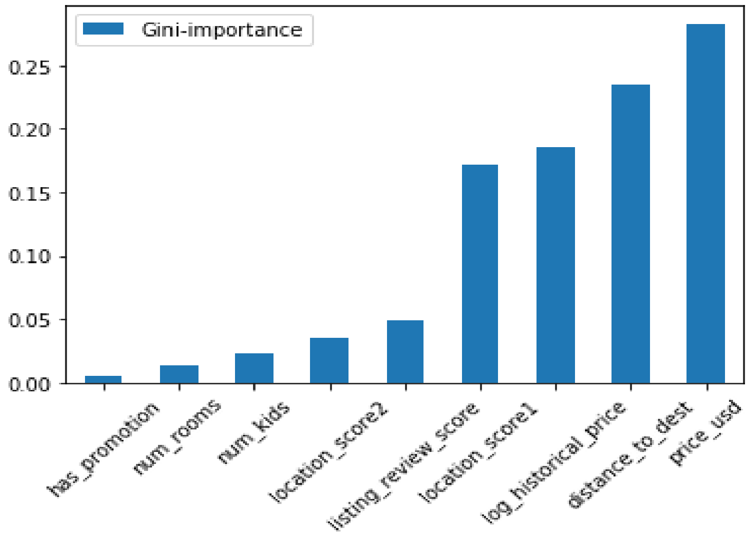
- RandomForest out performs in classifying 0's and 1's when compared to other algorithms
- Below is the Classification report of Hyper Parameter tuned RandomForest algorithm for cluster 5

```
              precision    recall  f1-score   support

           0       0.97      0.96      0.97    135356
           1       0.76      0.79      0.78      6235

    accuracy                           0.98    141591
   macro avg       0.87      0.88      0.87    141591
weighted avg       0.94      0.94      0.94    141591

auc scores:0.8170478787231762
```

- **Score card** is generated based on clicks, which gives the probability of booking and not booking. Deeper analysis can be made with the scores and its corresponding attributes which can explore the reason why the hotel was not booked/booked.

not_booking_prob	booking_prob	booked	clicked	site_id	listing_country_id	location_score1	location_score2	destination_id
1.0	0.0	0	1	5	100	0.000000	0	10455
0.0	1.0	1	1	5	219	1.474763	0	15764
0.3	0.7	1	1	29	219	1.081805	5	2463
0.1	0.9	1	1	14	138	0.741937	0	3229
0.1	0.9	1	1	5	31	1.990610	0	16361
0.0	1.0	1	1	14	219	0.524729	0	13539
0.0	1.0	1	1	5	219	1.420696	0	19585
0.0	1.0	1	1	5	219	1.311032	1	27615
0.1	0.9	1	1	5	219	1.549688	1	9137
0.1	0.9	1	1	5	219	1.163151	2	22148
0.2	0.8	1	1	24	99	1.955860	0	13292
0.2	0.8	1	1	5	219	0.959350	0	4748
0.5	0.5	0	1	5	219	1.124930	0	4748

- Observer and extracted Important Features which will be helpful in future analysis or business



6 Conclusion and Future Work

From the result table, we conclude the followings.

- The dataset was analysed by various machine learning algorithms that helped us come up with classification models for the Hotel Reservation System. The dataset has multiple classes without any significant perceived pattern that relates them to the features. This initially made it difficult to achieve reasonable accuracy.
- The dataset was clustered, if the new data fit in any of the cluster then only particular clusters data was pulled and further process or models were build. This can reduce the complexity and cost.
- For predicting clicks, Random Forest with SmoteNC gives good result. Studied the feature importance using available function and found out that price_usd, distance_to_dest, log_historical_price, location_score1 and listing_review_score have high impact in predicting if the user will click on the hotel or not.
- Hyper Parameter tuning can be done for each Forest algorithm in different clusters (requires huge system resources and time if the data is huge)
- For future work, ensemble stacking methods can be used to combine predictions from various algorithms