Expedia Hotel Recommendation

Sravya Sudarsanan
Dept. of Software Engineering
Lakehead University
Thunder Bay, Canada
1161998

Eunwha Park
Dept. of Software Engineering
Lakehead University
Thunder Bay, Canada
1202000

Jude Jolly Mampilly
Dept. of Software Engineering
Lakehead University
Thunder Bay, Canada
1114353

***Abstract*—Expedia Group is an American online travel agency through which one can reserve hotel rooms, rental cars, cruises, and holiday packages via the website and/or mobile application. This project focuses on predicting which hotel a customer booking through Expedia would prefer by getting 5 different hotels. For the purpose of this project, logs of customer behavior has been retrieved through the platform Kaggle. The logs were uploaded by the Expedia Group itself in Kaggle to conduct a competition based on an evaluation matrix. The logs/datasets include customer search terms, click-through rates, booking behavior, and whether or not the search result was a travel package. Our goal is to use k-NN, XGBoost and Random Forest classifiers to generate this recommendation system.**

***Index Terms*—Recommendation Systems, k-NN, Random Forest, Machine Learning, XGBoost**

I. Introduction

Today, most internet products are driven by recommender systems. Famous platforms like YouTube, Netflix, Amazon, etc. rely on recommender systems to filter out unnecessary content to get personalized recommendations for users. Recommender systems are proven to have provided a huge advantage to both internet businesses and their consumers. For Example, in 2009 Netflix hosted a competition in which they awarded a $1 million prize to a developer team that produced an algorithm that increased the accuracy of the company's recommendation system by 10% [3].Recently the tourism industry gained a steep rise in their market. With the evolution of e-commerce, selecting overseas destinations has become much easier through the web than before. It is important for people to select the best hotel to stay in to have a good impact on their trip. Therefore, it is essential to determine and suggest hotels based on a customer's previous interactions [4]. A recommendation System could play a vital role to increase a business's profit margin.

Using the internal algorithms of Expedia, we will provide 5 different hotel groups a user will most likely purchase. Based on past prices, client star ratings, physical distances from the city center, etc., we attempt to create a hotel cluster that would catch users' attention. This initiative may improve the likelihood that consumers will stick around and be considerably happier on any platform like this. The accommodations shown would be more individually designed for each guest.

The project will be approached through the data analytical life cycle phases. We are using the datasets uploaded in Kaggle by Expedia. These data were randomly selected from Expedia rather than from their overall statistics. The datasets include train, test, and destination files. The training datasets range from 2013 to 2014, while the test data is selected from 2015. There are 22 features that are analyzed in the dataset.

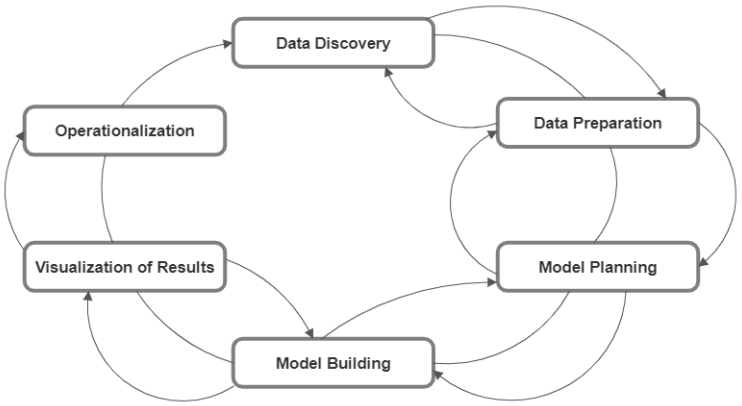

Fig. 1. Data Analytical Life Cycle.

II. Literature Review

The evolution of big data has made online markets produce an efficient algorithm for recommendation systems. Many researchers have put their efforts into finding efficient recommendation systems using Machine Learning (ML).

Mavalankar et al. [10] proposed using XGBoost and Random Forest models that provided much better results than other models like Naïve Bayes and SGD Classifier. Using Random Forest will provide more accurate results and handle over-fitting. Shenoy et al. [11] statement contradicts the study mentioned above as they state that algorithms using decision trees will not work well with this dataset, they work better with discrete datasets. The study states the difficulty in achieving accuracy due to the dataset having multiple classes without any patterns. To receive an increase in accuracy they proposed using Clustering and Ensemble techniques. The most observable results were obtained using Multinomial Logistic Regression because of its ability to handle numeric values. Aruzza et al. [12] proposed work agrees with Shenoy et al. that using clustering and boosting algorithms work well with

the provided dataset by Kaggle. They also mention that the user information lacks linearity which makes it difficult to recommend a hotel of their preference. Although the number of datasets varies within each provided study, Goa et al. [13] provided accurate results by filtering and boosting algorithms while omitting scalability situations. Narinder Kaur et al. [14] focused on improving hotel recommendation systems using machine-learning techniques based on the user's location. They implemented algorithms like Random Forest, Naïve Bayes, Link Prediction, and J48, out of which Random Forest attained a better rate of prediction accuracy.

Anindita et al. [15] worked on developing an automatic storage space recommendation system for an object-based cloud storage system. A smart storage system was produced to handle big data by recommending storage space. Through machine learning, they were able to make the system automatic. K-NN classifier helped them provide better performance than the other ML models.

Tan et al. [16] proposed building an E-learning recommendation system that helps new learners without any good prior knowledge to make choices. In their study, the collaborative filtering method was chosen to be the primary recommendation algorithm which proved to be effective. Zhou et al. [17] researched the impact of the YouTube recommendation system on video views. In this paper, they were able to reveal that a strong correlation existed between the view count of a video and the average view count of its top referrer videos. This suggests that a video's likelihood of becoming popular increases when it appears on popular video's related video suggestion lists.Han et al. [18] worked on a feature-based recognition system. He mentions that user-based and model-based collaborative filtering approaches impacted effectively on building recommender systems to data. This method is extensively used in most commercial recommender systems. The first two months of the online retail website's implementation of the recommended feature-based recommendation algorithms demonstrate a 75% increase in recommendation income. Hsu et al. [19] developed a personalized TV Recommendation System called AIMED based on user properties such as activities, interests, moods, experiences, and demographic information. The data of these properties were fed into a neural network model to predict TV viewers' program preferences. The results of this work indicated a significant increase in recommendation accuracy and decreased prediction errors when compared to the conventional models.

III. Proposed Model

Expedia consists of in-house clusters, which gather together a hotel based on customer preference. Hence, Expedia predicts which hotel group the customer will choose. We will be using classification models like K-Nearest Neighbor, Random Forest, and XGBoost to approach the solution to this problem since our data is related to supervised learning. By using both algorithms, we will compare each result by using ROC and AUC analysis to determine the best-generalized model.Our final result will show the 5 best clusters.

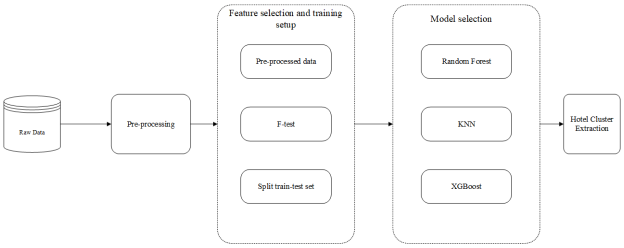

Fig. 2. Concept Diagram for the Hotel Recommendation System.

The proposed solution would be done through various phases. Initially we have analyzed the dataset provided by Kaggle. Table V-B is a list of the features and their respective datatype. The training and testing data are also differentiated based on booking and clicking events, where a customer may just click the hotel ad, or they will book the hotel. In the training set, the booking and clicking events are considered while the testing data only includes the booking events. There is also a destination file that shows reviews of hotels [1]. The attributes selected are based on customer preference, this includes date and time, location, marketing channel, number of rooms, number of adults, etc.

We then have done some pre-processing analysis on the data to identify feature relations. We performed correlation maps, identified some important features and grabbed statistics on these information. This included understanding the number of users who have booked using a package, number users who click or book. The peak times to travelling, the most visited country and the distance from the origin to the destination.

After the pre-processing stage, we further analyzed features using f-test. To understand what is best fit for the models selected and coming with the best result for the top hotel clusters.

Finally, once the train test sets were split we decided to use these on the chose models. The models selected for the purpose of achieving our goals are KNN, RandomForest and XGBoost. The final result will include the top 5 best clusters.The proposed solution for the problem follows the phases as shown in Figure 2.

A. K-Nearest Neighbor Algorithm

The K-nearest neighbors algorithm (KNN) is a supervised learning classifier to make classifications or predictions about the grouping of an individual data point using the assumption that similar points can be found near one another [5].

1) Determine distance metrics*:* We need to decide the distance between each point in order to see which points are close to each other. There are several distance measures we can choose from.Here, n represents the dimension of data sets.

(a) Euclidean distance: This is the most common measurement to calculate distance between data points. Below is the formula for this method.

$$d(x,y) = \sum_{i=1}^{n} \sqrt{(y-x)^2} \tag{1}$$

TABLE I
TABLE TO TEST CAPTIONS AND LABELS.

Field Name	Description	dataType
date_time	Timestamp	string
site_name	ID of the expedia POS	int
posa_continent	ID of continent w.r.t site_name	int
user_location_country	ID of the country	int
orig_destination_distance	distance b/w hotel & customer	double
user_id	id of user	int
is_mobile	1 connection with mobile device	int
is_package	1 as a part of a package	int
channel	ID of a marketing channel	int
srch_ci	Checkin date	string
srch_co	Checkout date	string
srch_adults_cnt	# of adults specified in the hotel room	int
srch_children_cnt	# of children specified in the hotel room	int
srch_rm_cnt	# of hotel rooms specified in the search	int
srch_destination_id	ID of the destination of search	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	int
cnt	# similar events w.r.t same user session	int
hotel_cluster	ID of a hotel cluster	int

(b) Manhattan distance: This method calculates the absolute value between data points. Given is the formula for this method

$$d(x, y) = \sum_{i=1}^{n} |y - x| \tag{2}$$

(c) Minkowski distance: This measurement is the generalized form of Euclidean and Manhattan distance metrics. The parameter p determines whether you will use Euclidean distance or Manhattan distance. Below is the formula for this method.

$$d(x, y) = \sum_{i=1}^{n} (|y - x|)^{(1/p)} \tag{3}$$

(d) Hamming distance: This is used to find out the points or vectors are matching each other. This method is commonly used in text classification. Below is the formula for this method.

$$d(x, y) = \sum_{i=1}^{n} |y - x| \tag{4}$$

2) Data preprocessing*:* We need to explore datasets in order to find out if missing values exist and whether data types are applicable to KNN algorithms. If there are NaN values or Null, then we need to fill this values with some reasonable numerical values. If there are non-numerical data types such as object or string, we need to transform those into numerical data types.

3) Prevent overfitting*:* KNN is susceptible to high-dimensional data inputs. Therefore, we can use some methods in order to prevent overfitting by taking some samples as validation sets out of train sets or process feature selection and dimensionality reduction techniques. Moreover, the value of K can affect the generalizability of the model. In this sense, we need to figure out which K value can adequately help the model perform efficiently. If K is too small, then the model will memorize input data and therefore, the model will be overfitted. On the other hand, if K is too high, then the model will not be able to accurately make a prediction. We will choose the optimal factor K by using and segregating training error rate and validation error rate with a varying value of K.

4) Sort the distances and choose the top K rows*:* In order to identify new data point's class, we will calculate the distances between new data point and all the existing data points that have already been assigned to specific classes. Then, we will sort these distance values in ascending order to choose the top K rows that have minimum distance. Finally, we can assign a class to the new data point based on the most frequent class of these rows. This can be formulated by the below formula where x0 is the new point.N0 is the set of k-nearest observations, and I(yi = j) is an indicator variable that is 1 if a given point yi , which was used to calculate to get a distance with x0, is a member of N0 and 0 if otherwise.

B. Random Forest

We will use random forest algorithm, which is an ensemble method used both for classification and regression by constructing decision trees while training the model [7]. After random forest creates parallel decision trees on the subset of sample data, it will provide the output of the class selected among the predictions from each tree by means of voting [7].

In our project, we will use RandomForestClassifier from sklearn library and focus on tuning the parameters of this function to improve the power of prediction of the model.

1) Hyper parameters *:* There are several parameters we need to tune before using random forest classification function.

(a) 'max_features': This represents the maximum number of features that an individual tree can allow [8]. We can simply allow trees to handle all the features of samples

or we can reduce the number of features by using sqrt. Generally, 'max_features' improves the performance of the model. However, if we keep increasing the 'max_features', then it will take a long time for the model to get updated. Therefore, we need to find the optimal amount of features for trees.

(b) 'n_estimators': This shows the number of trees we want to build in a forest before making decisions [8]. When a forest has more trees, the more robust the forest is and the model will perform better. However, it also has issues with velocity similar to 'max_features'. Therefore, we need to find out as much value as possible within the optimal time.

(c) 'max_depth': This decides the maximum number of splits that a decision tree can have [8]. The deeper the tree, the more splits it has and the more information it can collect about the data. That is, the model will underfit if there are too few splits, and will overfit if there are too many splits.

(d) 'min_samples_split': This is a parameter that specifies the minimum number of observations for a node to be split [8]. If we increase the value of this parameter, we are more likely to prevent the model from overfitting. However, we cannot indefinitely increase this number since it could cause the model to underfit.

(e) 'min_samples_leaf': Leaf is the decision tree's terminal node. The model is more likely to detect noise in data samples if the number of leaf is smaller [8]. In order to choose the best option, we will test out various leaf sizes.

2) Gini Index*:* We will use Gini Index as a criterion which is used to decide which feature will be the root node. Gini Index is important since it helps us know how much impurity the particular node has and give a hint to the random forest when it decides the appropriate root node [9]. The lowest Gini Index means low impurity and therefore, a Gini Imputify of 0 is the best possible impurity for any data set. Below is the formula of Gini Index where m is the number of splits and n is the number of classification.

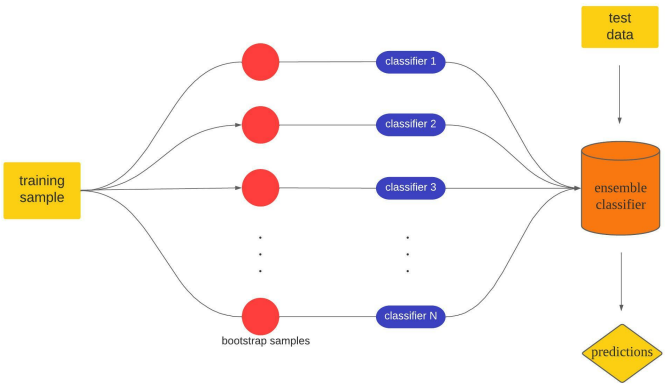

Fig. 3. Random Forest Diagram.

C. XGBoost

XGBoost is a distributed, scalable gradient-boosted decision tree library (GBDT) [20]. As with random forest, XGBoost use

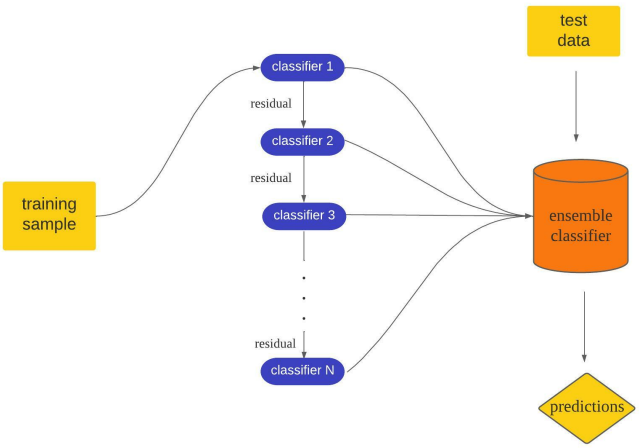

Fig. 4. XGBoost Diagram.

a decision tree ensemble learning algorithm where multiple machine learning algorithms are combined to create a better model for classification [20]. Unlike random forest, XGBoost is a sequential ensemble model where a next model is made on the loss errors of previous models and tries to minimize the errors[21].

1) Boosting*:* The boosting ensemble technique consists of three simple steps.

(a) An initial model H_0 is defined to predict the target variable y. H_0 should be a function that minimizes the loss function.

$$H_0(x) = \arg\min_{\beta} \sum_{i=1}^{n} Ł(y_i, \beta), \quad n = number of samples \tag{5}$$

(b) A new model f_1 is fit to the residual, which is (y-H_0), from the previous model.

(c) H_0 and f_1 are combined to give H_1, which is the boosted version of H_0.

$$H_1(x) \leftarrow H_0(x) + f_1(x) \tag{6}$$

The whole steps can be done for m times until residuals become minimized as much as possible.

$$H_m(x) \leftarrow H_{m-1}(x) + f_m(x) \tag{7}$$

To generalize the formula, we can rewrite the equations like the below.

$$\beta_{im} = y_i - H_{m-1}(x_i) \tag{8}$$

By using equation (8), fit $f_m(x)$ to β_{im}. Training set will be $\{(x_i, \beta_{im})\}$, where $1 \leq i \leq n$. Then, update the model by using $f_m(x)$.

$$H_m(x) = H_{m-1}(x) + learning\_rate \times f_m(x) \tag{9}$$

TABLE II
ACCURACIES FOR EACH MODEL WITH OUTLIERS.

Model Name	Map@5	Accuracy
KNN	0.28813333333333335	0.1911666666666668
RandomForest	0.3735944444444445	0.2808333333333333
XGBoost	0.28805694444444446	0.197833333333333333

TABLE III
ACCURACIES FOR EACH MODEL WITHOUT OUTLIERS.

Model Name	Map@5	Accuracy
KNN	0.29557860082304527	0.19476543209876543
RandomForest	0.3795736625514403	0.2860246913580247
XGBoost	0.29879670721893003	0.2077037037037037

IV. ANALYSIS

The models were run with and without outliers. As shown in Table 2 and 3, we can see that the accuracy have changed for each model once the removal of outliers. The data collected for this study is in random order, the purpose of this study is to predict the booking outcome for a user event. The target variable is hotel_cluster. The train and test datasets are split based on time ranging from 2013 to 2014. The test data only focuses on details from 2015. To understand user booking outcome it was necessary to understand the following attributes:

- Most travel country
- Booking/Clicking
- Booking through packages
- Mobile device search or not

These attributes were independently analyzed as well as in comparison to hotel_cluster. The count plot displayed the most traveled country as ID50. Majority of the users who visit seem to book the trip. As shown in the figure, booking is far higher compared to clicks, where representations are as follows: 1 if a booking and 0 if a click. However, the majority of the users did not book as part of a package. For is_package, the representations were as follows: 1 if booking/clicking was part of a deal and 0 if otherwise. It was also evident that most users did not connect using their mobile device. However, with is_booking compared to hotel_cluster, cluster ID 91 had the highest rate of booking/clicking.

In the travel industry, the peak season is approximately around June-August. So, one of the most important features to focus on is peak times. From the given dataset, 2014 June and 2014 December seemed to be the highest travel times.

We investigate if there is any 'null' or 'nan' value in data samples and we fill those values with the most frequent value for each attribute. Also, we add extra columns, 'cin_day', 'cin_month', 'cin_year', and 'stay_dur', to further analyze data. Each column implicates 'check-in day', 'check-in month', 'check-in year', and 'stay duration' for each booking. After adding these columns, we conduct our in-depth data analysis.

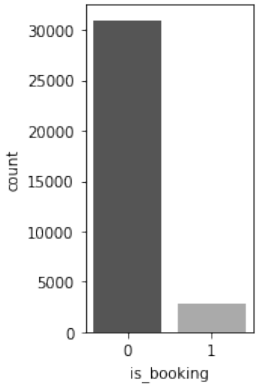

Fig. 5. A representation of how many users have booked or just clicked(1).

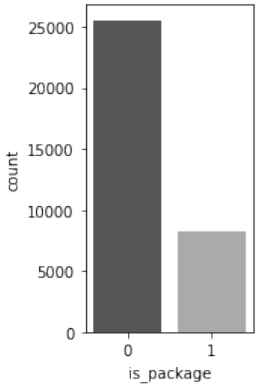

Fig. 6. Representation of how many users have booked as a package deal(1).

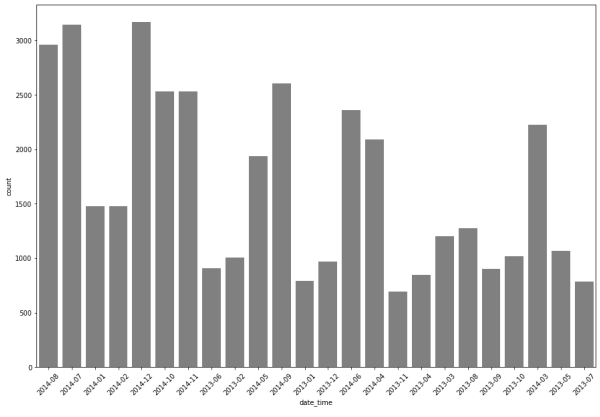

Fig. 7. Peak time of travel in 2013 and 2014.

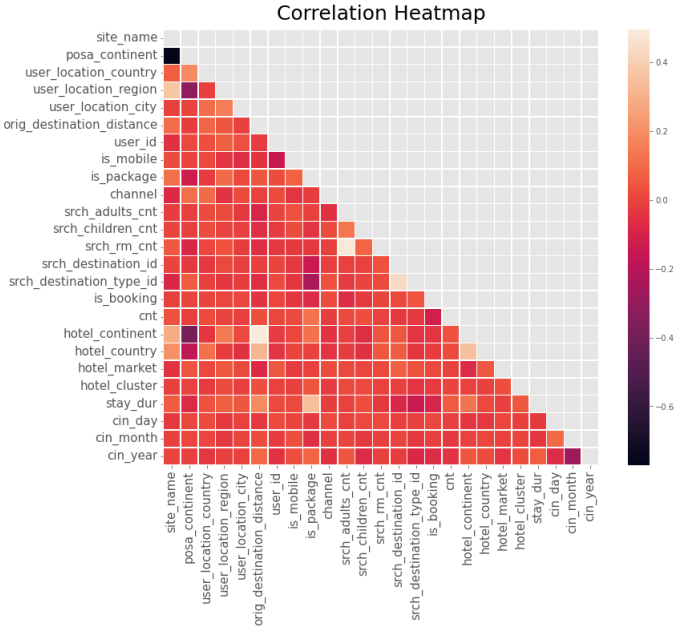

Fig. 8. Correlation map.

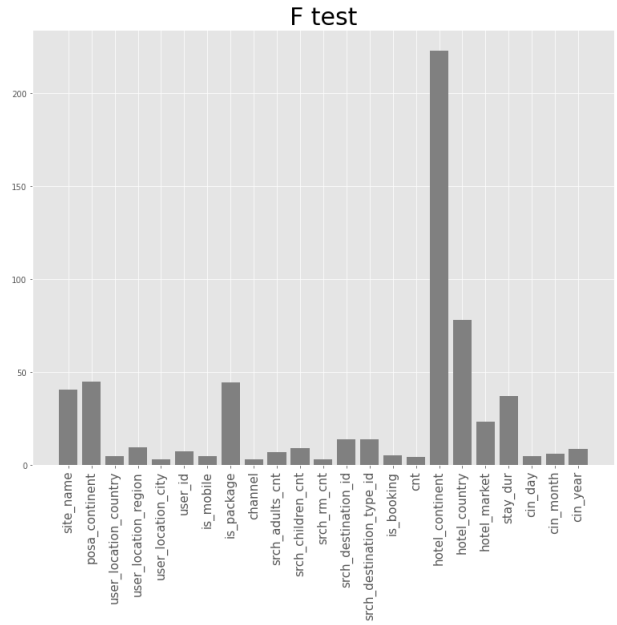

Fig. 9. F test analysis.

The correlation map provides relationships between the features of the given dataset. As given in Figure 7, the strongest negative relationship occurs between posa_continent and site_name, hotel_continent and posa_continent, and user_location_region and posa_continent. There exists a positive relationship between the number of adults and children so it can be considered families book rooms based on the number of people in a family. Orig_destionation_distance and hotel_continent have a positive relationship which means that people prefer to travel further away from their origin. Further analysis will be done through our models, KNN and Random Forest.

We also conduct 'f-test' analysis to figure out which attributes have strong relationship with 'hotel_cluster' target attribute. 'Figure 8' shows us that hotel_continent and hotel_country have the most correlation with the target variable. We can use this result for the feature selection if our model does not have a good performance later.

V. Compare Models

A. Evaluation metric

We will use the 'Mean Average Precision@k' (MAP@k) metric to compare the performance of three models. Since our project purpose is to recommend 5 different hotel clusters based on a user's character, focusing on single numeric accuracy fails to capture the order in which items are recommended. We want to find first 5 hotel clusters which have the highest scores that the model predicted. To calculate MAP@k, we first need the AP@K value which stands for Average Precision [22].

$$AP@k = (1/N(k)) \sum_{i=1}^{k} TP_{seen}(i)/i \tag{10}$$

Here, TP stands for True Positives, and N(k) is the minimum value between k and the total number of TP. In our project, since we need five hotel clusters, k value is set to 5.

$$N(k) = min(k, TP_{total}) \tag{11}$$

$$TP_{seen}(i) = \begin{cases} 0, & \text{i}^{\text{th}} \text{ is False} \\ \text{TP seen till i}, & \text{i}^{\text{th}} \text{ is True} \end{cases}$$

Using AP@k, we can find MAP@k by calculating the average of overall AP@k.

$$MAP@k = 1/N \sum_{i=1}^{N} AP@k_i \tag{13}$$

For example, if a model predicts [1,2,3,4,5] where the real value is 1, then AP@k value is 1. When the prediction is [2,1,3,4,5], then AP@k value is 0.5. For these two cases, MAP@k is 0.75.

B. Result of each model

As we mentioned, we used 3 different models - KNN, Random Forest, XGBoost - to make a comparison and choose the best model. Before comparing each model, we trained each model many times with different hyper-parameters and selected the best hyper-parameters for each of the 3 models that give better predictions. Using the best performance models for each of 3 models, we compared the accuracy and Map@5 scores. We can see the performance score has increased with optimal hyperparameters for each model. Also, Random Forest and XGBoost have relatively better scores than KNN meaning that KNN is inappropriate for this problem.

TABLE IV
ACCURACIES FOR EACH MODEL USING HYPERPARAMETER TUNING.

Model Name	Map@5	Accuracy
KNN	0.3007374485596707	0.2123456790123457
RandomForest	0.40961975308641974	0.3160493827160494
XGBoost	0.40524609053497945	0.30607407407407405

Finally, we also applied f-test results to reduce model dimensionality with feature selection from 1 feature to 22 number of features. However, both 3 models do not improve in evaluation scores and the best scores for each of them are accuracies without feature selection.

VI. CONCLUSION

This project is meaningful as many people are planning to go on a trip since the curve of Covid-19 has been flattened [3]. We can think of some positive effects of the Expedia project. First, it is possible to achieve the main goal of this project which is letting customers find their optimal hotel easier. Second, not only can this project impact customers but hotel service quality can also be improved. Once we apply this recommendation model to other platforms, it would be easier to compare many different hotels simultaneously which will motivate hotels to enhance their amenities. Last but not least, this project can be expanded to other areas that require selection or decision-making, not just confined to the hotel industry. There are tons of areas that many people want to get recommendations such as food, items, or even tourist attractions.

From the models selected, Random Forest and XGBoost showed the best accuracy in selecting the top hotel_cluster. The accuracy levels were analyzed in a few different ways to ensure the best model was chosen. This included running the models with and without outliers. With the use of hyperparameter tuning, the accuracy levels were increased by small amounts when compared to the initial values. However, when we use f-test feature selection, 3 models did not work well. Especially, KNN produced the worst accuracy. We conclude that KNN is not suitable for this problem since KNN uses a lazy algorithm which is just memorizing the data without learning the pattern of the data set. When it comes to RandomForest and XGBoost, we analyzed why they do not work well with f-test feature selection. First, even though feature selection can generally improve classification accuracies, it depends on the method adopted [23]. That is, the f-test might not be adequate for this problem and these models. Second, it might be because we already reduced enough features during the data analysis phase. It is important to take the ratio between the number of samples and features into consideration while training models to increase accuracies. Models with fewer features have high bias, which can result in too many predictions near the same value, reducing their accuracy [24].

VII. FUTRE WORK

Even though we successfully increase the evaluation score by removing outliers and tuning hyperparameters, we still need to figure out if other feature selection methods can improve accuracies. We would further expand our study by using other different feature selection methods to predict the best values based on the dataset. In addition, future work could include more models and studies done like the use of the Support Vector Machine (SVM) or Naive Bayes. This would give more results and better analysis.

VIII. ACKNOWLEDGEMENT

We thank Dr. Akilan (Lakehead University) for his patience and valuable feedback throughout the duration of this project. The team members of this group for the constant effort put into this project and ensuring the best results. Finally, a special thanks to Kaggle for putting forth a challenge to such topics. This can be further used in other business areas as well.

REFERENCES

[1] "Expedia Hotel Recommendations," Kaggle. [Online]. Available: https://www.kaggle.com/competitions/expedia-hotel-recommendations. [Accessed: 16-Sep-2022

[2] D. Zaidi, "Dramatic increase in Canadian residents travelling overseas since last June: Statcan," CTVNews, 23-Aug-2022. [Online]. Available: https://www.ctvnews.ca/canada/dramatic-increase-in-canadian-residents-travelling-overseas-since-last-june-statcan-1.6038829. [Accessed: 16-Sep-2022

[3] K. Liao, "Prototyping a recommender system step by Step Part 1: Knn Item-based collaborative filtering," Medium, 19-Nov-2018. [Online]. Available: https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea. [Accessed: 11-Oct-2022]

[4] E. Abdul Hafiz and N. Kaur, "Improved Hotel Recommendation System Using Machine Learning Technique," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), 2022, pp. 769-773, doi: 10.1109/AIC55036.2022.9848942.

[5] "What is the K-nearest neighbors algorithm?," IBM. [Online]. Available: https://www.ibm.com/topics/knn. [Accessed: 12-Oct-2022]

[6] T. Srivastava, "K nearest neighbor: Knn algorithm: KNN in Python R," Analytics Vidhya, 18-Oct-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/. [Accessed: 12-Oct-2022]

[7] A. Navlani, "Sklearn Random Forest classifiers in python tutorial," DataCamp, 16-May-2018. [Online]. Available: https://www.datacamp.com/tutorial/random-forests-classifier-python. [Accessed: 12-Oct-2022]

[8] S. Saxena, "Random Forest hyperparameter tuning in Python: Machine learning," Analytics Vidhya, 20-Apr-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/. [Accessed: 12-Oct-2022]

[9] A. Saini, "Random Forest Algorithm for absolute beginners in Data Science," Analytics Vidhya, 26-Aug-2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/[Accessed: 12-Oct-2022]

[10] A. A. Mavalankar, R. Misra, C. Gandotra, and A. Gupta, "Hotel Recommendation System." [Online]. Available: https://arxiv.org/pdf/1908.07498.pdf. [Accessed: 09-Oct-2022].

[11] G. G. Shenoy, M. A. Wagle, and A. Shaikh, "Kaggle Competition: Expedia Hotel Recommendations." [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1703/1703.02915.pdf. [Accessed: 09-Oct-2022].

[12] M. Arruza, J. Pericich, and M. Straka, "The Automated Travel Agent: Hotel recommendations ... - stanford university." [Online]. Available: http://cs229.stanford.edu/proj2016spr/report/017.pdf. [Accessed: 12-Oct-2022]

[13] G. Huming and L. Weili, "A Hotel Recommendation System Based on Collaborative Filtering and Rankboost Algorithm," 2010 Second International Conference on Multimedia and Information Technology, 2010, pp. 317-320, doi: 10.1109/MMIT.2010.14.

[14] E. Abdul Hafiz and N. Kaur, "Improved Hotel Recommendation System Using Machine Learning Technique," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), 2022, pp. 769-773, doi: 10.1109/AIC55036.2022.9848942.

[15] A. S. Mondal, M. Anirban and C. Samiran, "Machine learning-driven automatic storage space recommendation for object-based cloud storage system," Complex Intelligent Systems, vol. 8, (1), pp. 489-505, 2022. Available: http://ezproxy.lakeheadu.ca/login?url=https://www.proquest.com/scholarly-journals/machine-learning-driven-automatic-storage-space/docview/2635338428/se-2. https://doi.org/10.1007/s40747-021-00517-4.

[16] H. Tan, J. Guo and Y. Li, "E-learning Recommendation System," 2008 International Conference on Computer Science and Software Engineering, 2008, pp. 430-433, doi: 10.1109/CSSE.2008.305.

[17] R. Z. H. E. University, R. Zhou, H. E. University, S. K. U. of Massachusetts, S. Khemmarat, U. of Massachusetts, L. G. U. of Massachusetts, L. Gao, I. C. S. Institute, and O. M. V. A. Metrics, "The impact of YouTube recommendation system on Video Views: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement," ACM Conferences, 01-Nov-2010. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1879141.1879193?Accessed: 12-Oct-2022].

[18] E.-H. (S. H. iX. Inc., E.-H. (S. Han, iX. Inc., G. K. iX. Inc., G. Karypis, U. of Bremen, U. for H. Sciences, U. of Duisburg-Essen, A. Online, I. B. M. T. J. W. R. Center, and O. M. V. A. Metrics, "Feature-based recommendation system: Proceedings of the 14th ACM international conference on information and knowledge management," ACM Conferences, 01-Oct-2005. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1099554.1099683?Accessed: 13-Oct-2022].

[19] S. H. Hsu, M.-H. Wen, H.-C. Lin, C.-C. Lee, and C.-H. Lee, "Aimed- A personalized TV recommendation system," SpringerLink, 01-Jan-1970. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-72559-6_18citeas.[Accessed: 13-Oct-2022].

[20] "What is XGBoost?," NVIDIA Data Science Glossary. [Online]. Available: https://www.nvidia.com/en-us/glossary/data-science/xgboost/. [Accessed: 06-Nov-2022].

[21] N. Kapoor, "XG boost -the base and mathematics behind it," Numpy Ninja, 23-Aug-2021. [Online]. Available: https://www.numpyninja.com/post/xg-boost-the-base-and-mathematics-behind-it. [Accessed: 06-Nov-2022].

[22] A. U., "How mean average precision at K (map@k) can be more useful than other evaluation metrics," Medium, 07-Jul-2020. [Online]. Available: https://medium.com/@misty.mok/how-mean-average-precision-at-k-map-k-can-be-more-useful-than-other-evaluation-metrics-6881e0ee21a9. [Accessed: 14-Nov-2022].

[23] Chu C;Hsu AL;Chou KH;Bandettini P;Lin C; ; "Does feature selection improve classification accuracy? impact of sample size and feature selection on classification using anatomical magnetic resonance images," NeuroImage. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/22166797/. [Accessed: 25-Nov-2022].

[24] T. Yiu, "The curse of dimensionality," Medium, 29-Sep-2021. [Online]. Available: https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e. [Accessed: 25-Nov-2022].