Master's in Information Technology and Analytics

Rutgers – The State University of New Jersey, Newark

Expedia Hotel Recommendation System

Algorithmic Machine Learning

Team Members:

Ashita Shetty
(ashita.shetty@rutgers.edu)

Manan Desai
(md1810@scarletmail.rutgers.edu)

Rishita Chebrolu
(rishita.chebrolu@rutgers.edu)

Ramya Sri Gautham
(ramya.srigautham@rutgers.edu)

Sanskritee Rajpal
(sr1926@scarletmail.rutgers. edu)

***Abstract*— The Expedia Hotel Recommendation System is a ground-breaking innovation in the field of online travel agencies because it uses cutting-edge machine learning algorithms to improve the user experience significantly. It runs on a huge dataset that was carefully curated from Expedia's platform. It includes a lot of user interactions, details about hotels, and a complete log of booking events. This system delves deeply into comprehending the intricate web of user behaviors, preferences, and nuanced search patterns through an intricate process of exploratory data analysis and preprocessing. The system effectively distills this vast dataset into actionable insights by leveraging the strength of supervised learning techniques, particularly classification. This makes it possible to generate highly tailored and meticulously optimized hotel recommendations. The system's commitment to providing users with personalized travel experiences that resonate with their unique preferences and expectations is demonstrated by this tailored approach, which marks a paradigm shift in the landscape of online travel platforms. We aim to develop a recommendation model that predicts the hotel cluster based on user's search attributes.**

I. INTRODUCTION

Software systems that use user data and content characteristics to provide users with customized recommendations are known as recommendation engines. By considering previous interactions, preferences, and behaviors, they assist individuals in discovering relevant content, products, or services. These come in different sorts, including cooperative separating, content-based sifting, half breed frameworks, lattice factorization, profound learning models, and setting mindful suggestions, and are broadly utilized in online business, web-based features, virtual entertainment, and more to upgrade client encounters and drive commitment.

Expedia is a well-known brand in the online travel industry. It is a comprehensive platform that provides a wide range of travel services, including hotel reservations, flight reservations, vacation packages, and a variety of travel-related activities that can be accessed through its user-friendly website and mobile app. Our primary goal is to accurately predict booking outcomes, or hotel clusters, for individual user events in a market that is fiercely competitive and where precision in matching customers with the best hotel inventory is crucial. Utilizing our extensive Training Data, a comprehensive collection of user interactions, clicks, and bookings made on Expedia's platform, is fundamental to this endeavor.

The Expedia Hotel Recommendation system can offer travelers customized hotel recommendations, making it simpler for them to find accommodations that meet their requirements and preferences. Recommendation engines can improve the user experience by customizing hotel recommendations to each user, increasing the likelihood that they will quickly locate suitable options.Expedia's revenue can rise because of users booking hotels through the platform thanks to accurate recommendations.

II. THE EXPEDIA DATASET

Expedia provided datasets that captured the logs of user behavior. These include details about what the customers searched for, how they interacted with the search results - i.e. whether they booked the hotel, or simply clicked to view details, whether the search result was a travel package, and so on. The goal is to predict which "hotel cluster" the user is likely to book, given his search details. These "clusters" have been created by Expedia based on some undisclosed in-house algorithms. But the intuition is that hotels belonging to a cluster are similar for a particular search - based on historical price, customer star ratings, geographical locations relative to city center, etc. These hotel clusters serve as good identifiers to which types of hotels people are going to book, while avoiding outliers such as new hotels that don't have historical data.

The training and testing datasets are split based on time: training data from 2013 and 2014, while test data are from 2015. Training data includes all the users in the logs, including both click events and booking events. Test data only includes booking events. Data fields in train/ test datasets are as below-

COLUMN	DESCRIPTION	DATA TYPE
date_time	Timestamp	object
site_name	ID of the Expedia point of sale (i.e., Expedia.com, Expedia.co.uk, Expedia.co.jp,...)	int64
posa_continent	ID of continent associated with site_name	int64
user_location_country	The ID of the country the customer is located	int64
user_location_region	The ID of the region the customer is located	int64
user_location_city	The ID of the city the customer is located	int64
orig_destination_distance	Physical distance between a hotel and a customer at the time ot search. As null means the distance could not be calculated	float64
user_id	ID of user	int64
is_mobile	1 when a user connect from a mobile device, 0 otherwise	int64
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	int64
channel	ID of a marketing channel	int64
srch_ci	Checkin date	object
srch_co	Checkout date	object
srch_adults_cnt	The number of adults specified in the hotel room	int64
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room	int64
Srch_rm_cnt	The number of hotel rooms specified in the search	int64
Srch_destination_id	ID of the destination where the hotel search was performed	int64
Srch_destination_type_id	Type of destination	int64
is_booking	1 if a booking, 0 if a click	int64
cnt	Number of similar events in the context of the same user session	int64
hotel_continent	Hotel continent	int64
hotel_country	Hotel country	int64
hotel_market	Hotel market	int64
hotel_cluster	ID of a hotel cluster	int64

We also have a Destinations dataset which gives us unlabeled latent information which is extracted from the hotel reviews. This has 149 columns for each *srch_destination_id.* Despite its

difficulty in interpretation, this dataset provides us a valuable opportunity to decipher important insights for improving our recommendation system.

Data fields in Destinations dataset are as below-

COLUMN	DESCRIPTION	DATA TYPE
srch_destination_id	ID of the destination where the hotel search was performed	int64
d1-d149	latent description of search regions	int64

III. PRE-PROCESSING

A. DATA CLEANING

Due to the huge size of the dataset, we realized the need for data pre-processing in order to understand what columns can be altered or have values not considered. Data cleaning helps to increase the quality of the dataset to contribute to efficient decision-making.
Major steps of data cleaning followed, are as mentioned in the table below:

ACTION	COMMENTS
Load Original Data	~ 37 million rows
Filter train data with only *is_booking* flag with 1 value	Data set rows reduced to ~3 million
Filter train data removing *agents* (bookings>20)	Data set rows reduced to ~2.5 million
Convert column *orig_destination_distance* with category column to remove null values	Physical distance between a hotel and customer at the time of search is what is indicated here. A 'null' value could be due to un-recorded/unavailable data.
Convert Date columns to Date format - *srch_ci, srch_co, date_time*	3 columns' data was updated.

Since our test dataset only includes flag 1 in the *is_booking* column, we ensured to carry out the same with our training dataset. This ensures that users with no productive interaction may not affect or create any bias in the model's ability.

In the next step, we focus on removing agent bookings from our training data since they involve bulk bookings that may not directly factor in our recommendations. The only column with null values was *orig_destination_distance* which was then converted to remove those values.

B. FEATURE ENGINEERING

Timestamp columns are indicative of different temporal patterns or trends in the stay period among users in correlation with their trips. Furthermore, data related to the day user made the booking specifies how long it took to convert the user from a 'lurker' to an actual 'user'.

NEW FEATURES
Stay_dur (Duration of stay (checkout - Check in))
No_of_days_before_booking
Current_mon
Current_year
Srch_ci_day
Srch_ci_mon
Srch_ci_year
Srch_co_mon
Srch_co_year

C. DIMENTIONALITY REDUCTION (PCA)

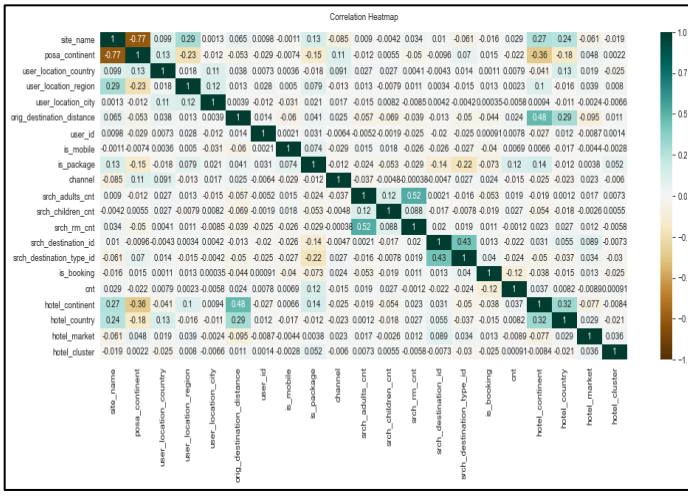
The "curse of dimensionality" refers to the difficulties that come with working with high-dimensional data like the Destination set's 149 unlabeled columns. As a result, there may be an increased chance of models being overfit, more computational complexity, and longer training times. Principal Component Analysis (PCA) emerges as a powerful method for resolving these issues by reducing dimensionality without compromising significant dataset information.

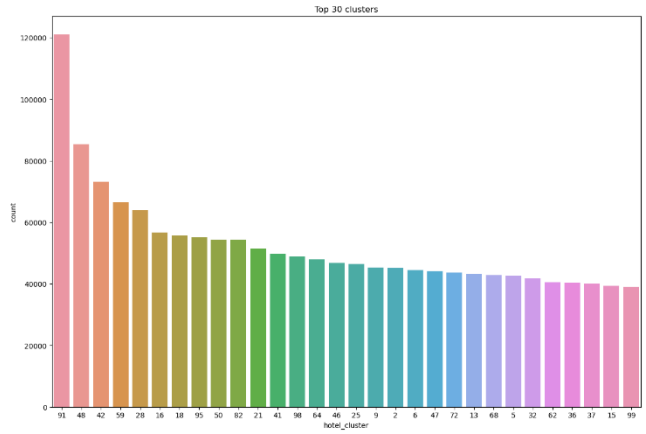
PCA effectively summarizes essential information by using a smaller set of dimensions known as principal components to identify the most impactful basis for reorganizing the dataset. PCA significantly reduces the total number of dimensions while maintaining the fundamental characteristics of the original data by carefully selecting these components. We use PCA to reduce the number of columns to 10.

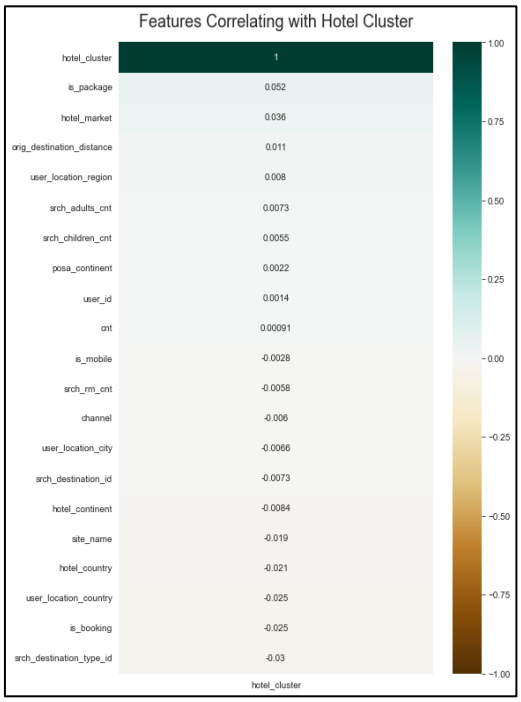
Through a common identifier, the *srch_destination_id* column, these transformed columns, now reduced to 10 dimensions, are merged with the initial training dataset. The enriched data from both datasets are combined during this merging process, resulting in a final dataset with reduced dimensionality while keeping important information about user events and hotel interactions. The resulting dataset has 2.5 million rows and 39 columns, is structured and concise, and incorporates essential features from both datasets. This change gives us access to a dataset that is easier to use and more streamlined, which makes it easier to run analyses, train models, and test.

D. CORRELATION

The below (L) is indicative of how well or how poorly columns are correlated to each other. The figure (R) shows the correlation of all the columns with the target variable *hotel_cluster*. Since none of the columns are strongly correlated, it indicates that there is a weak relationship between the features and the target variable. Individual features do not exhibit any pattern of relationship with the target variable. Feature *is_package* relatively has the best correlation with *hotel_cluster*.







IV. EXPERIMENTAL EVALUATION

A. METHODOLOGY

The training dataset with user events from 2013-2014 is a huge set containing 37 million records with a size of 4GB. The process of training with such a large dataset is excessively time-consuming and impractical given the limitations of our system's memory. Additionally, it poses a challenge to our ability to experiment with different models and parameter variations. Therefore, it is imperative to reduce the size of our dataset, ensuring that we can iterate through diverse models while preserving the representativeness of the entire data.

To overcome this issue, we experiment with three different approaches for developing our models.

1. Dataset of 50,000 unique users- Randomly sampling 50,000 unique users from the clean dataset will let us preserve all the events associated with that user. This reduced the dataset size to 156452 records.

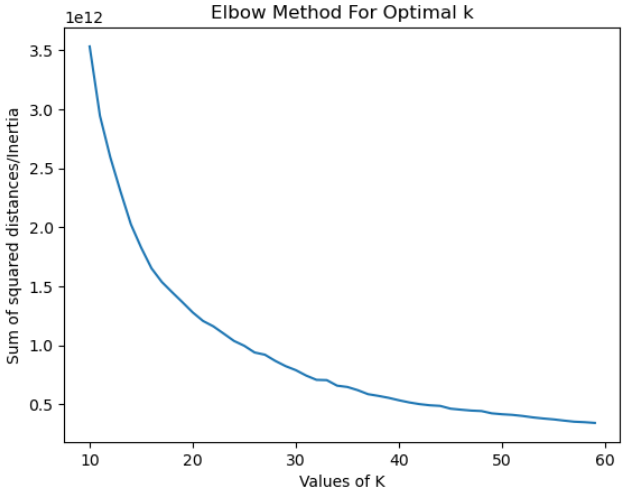
2. Dataset of top 10 clusters- Considering only the top 10 frequent clusters from the clean dataset. This reduced the dataset size to 568972 records.

3. Clustering as a pre-processing step-

The adoption of clustering techniques as a pre-processing step is a wise choice for improving data preparation methods for improved model performance. This strategy aims to improve the model's performance by addressing several inherent challenges in the dataset and provides a balanced dataset. It plays a crucial role in the pursuit of optimizing model development in technical contexts because of its numerous benefits, which include handling imbalanced distributions, encouraging feature engineering, reducing dimensionality, and ultimately improving model performance.

The first step was to reduce the original dataset's 100 clusters to the optimal number of clusters, k. K-Means clustering was used to complete this task, focusing solely on the dataset's destination attributes- *srch_destination_id*, *srch_destination_type_id*, *cnt*, *hotel_continent*, *hotel_country*, *hotel_market*. The Elbow Method was used to identify the optimum number of clusters.

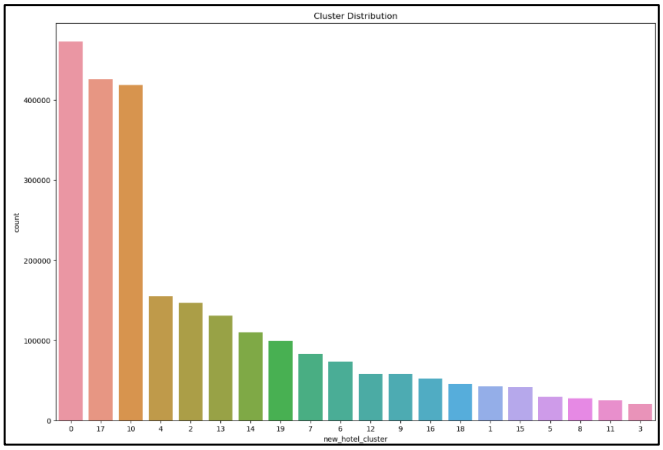
K-means clustering defines clusters such that the total intra-cluster variation (or total within-cluster sum of square-WSS) is



minimized. The total WSS measures the compactness of the clustering, and we want it to be as small as possible. The elbow method runs k-means clustering algorithm on the dataset for a range of values of k (10-60). For the elbow method, for each k, plot the total WSS and look for the elbow point where the rate of decrease shifts. This elbow point can be used to determine k.

After figuring out the optimum number of clusters, the dataset went through a transformation phase in which each data point got a specific cluster value from the K-Means clustering results. The dataset was effectively enriched with cluster-based information as a result of the integration of these newly attributed cluster values as an additional feature-

new_hotel_cluster. The *new_hotel_cluster* distribution is as shown below-



B. MODELLING

The goal is to predict a hotel cluster based on user's search attributes. Since, hotel cluster is already a known information, the problem we have at hand is essentially a classification problem. We can use various supervised machine learning methods to predict the hotel cluster.

1. Random Forest Classifier- Random Forest works in giving higher accuracy and reduce overfitting due to averaging multiple decision trees. It works well with large datasets with more attributes because of independent trees which makes the process parallel. It also allows us to fit the non-linear data models. The modelling is done by varying *RandomForestClassifier* parameters- *n_estimators* and *min_weight_fraction_leaf* and performing cross validation in 3 folds.

2. Keras- This is a neural network model built using Keras API for classification. This provides us with high level and user-friendly framework to build a neural network model. We use the default *Adam* optimiser for our dataset. Adam optimiser is a popular algorithm which works by reducing the loss function. The modelling is done by defining different dense layers and using the *softmax* activation for our multi-class classification problem. The model accuracy is improved by varying *epochs* and *batch_size.*

3. Support Vector Machines- SVMs can handle nonlinear relationships and perform well with large datasets. We use the default Radial Basis Function (RBF) kernel to model for our dataset. But due to the high time consumption and memory limitations, the accuracy for SVM on all the three approaches isn't feasible.

C. EVALUATION METRICS

Accuracy- Accuracy is the percentage of correct classifications that a trained machine learning model achieves, i.e., the number of correct predictions divided by the total number of predictions across all classes.

Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)

Mean Average Precision at K- *map@K*, is a fundamental metric for evaluating the quality and relevance of a model's predictions in situations where top K recommendations are crucial. In the context of evaluation metrics for rank-based classifications and recommendation systems. The accuracy and relevance of a model's top K predictions within the specified K items are measured by the average precision (AP) of those predictions. It basically measures how good the top recommendations from the model are. A higher *map@K* score, for instance, indicates that the model's top K recommendations are more in line with users' preferences or requirements across the entire dataset. It is a useful benchmark for evaluating and comparing the performance of various models in terms of providing accurate and useful predictions from a set of options. As a result, *map@K* is a crucial metric, especially in recommendation systems, as it provides insight into the precision and relevance of the model's top K predictions and helps evaluate and enhance recommendation algorithms.

$$MAP@5 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{min(5,n)} P(k)$$

D. RESULTS

Model	Accuracy/ map@K (%)				
	50k users	Top 10 Clusters	Clustering		
			k=10	k=20	k=30
Random Forest	22.47	42.4	49.3	63.6	41
Keras	-	24.1	87.1	93	73.9
SVM	3.9	-	-	-	-

From the results, it is evident that models perform better when clustering is included as a pre-processing step. k=20 is the optimum number of clusters, k=30 performs poorly comparitively and k=10 outperforms because it might not be explaining the complexity in the dataset.

V. FUTURE WORK

The Expedia Hotel Recommendation System is poised to change the way personalized travel experiences are planned and delivered soon. Looking ahead, the following key areas appear to be the focus of further innovation and improvement:

Usage of Cutting-edge ML Strategies (e.g., XGBoost, Ensembling): Advanced methods like XGBoost, Gradient Boosting, and ensemble techniques can be used in addition to conventional machine learning algorithms.
Beyond ML, Direct Approaches to Research: These strategies could incorporate measurable investigation, rule-based frameworks, or direct calculations customized explicitly to the area issue. Distributed computing frameworks like Apache Spark or Hadoop could be used to efficiently process and analyze large datasets simultaneously when these methods are run in parallel on clusters.
Utilizing Information from Kaggle Forums About Data Leaks: Exploitation of insights from discussions and forums, such as Kaggle discussions regarding the Expedia challenge, can be done. These insights may include community-identified data leaks or patterns, making it possible to improve models or strategies to take advantage of these potential benefits for better predictive accuracy or feature engineering.

User, Destination, and Cluster Similarity Advanced Analysis: The creation of similarity metrics between users, destinations, and clusters was the subject of a comprehensive analysis. To find hidden relationships or patterns in the dataset, methods like cosine similarity, collaborative filtering, or graph-based approaches can be used. By identifying latent similarities between users, destinations, and clusters model enhancement can be done.

VI. CONCLUSION

In conclusion, this report reveals a path for ongoing improvement, despite the significant progress that has been made in developing the Expedia hotel recommendation system. The in-depth examination of user interactions and data relationships, as well as the investigation of cutting-edge methodologies and insights from community forums, present opportunities for future enhancements. Expedia's quest for an ever-evolving, accurate, and personalized hotel recommendation system is still ongoing, driven by innovation and a commitment to providing users worldwide with exceptional travel experiences.

VII. REFERENCES

[1] E. Rencberoglu, "Fundamental Techniques of Feature Engineering for Machine Learning," towardsdatascience.com, April 1, 2019.
Available: https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114.

[2] Kaggle, "Expedia Hotel Recommendations," kaggle.com, 2016.
Available: https://www.kaggle.com/c/expedia-hotel-recommendations/data.

[3] P. Khandelwal, "Which algorithm takes the crown: Light GBM vs XGBOOST?", analyticsvidhya.com, June 12, 2017.
Available:
https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

[4] Scikit Learn, "Multiclass and multilabel algorithms," scikit-learn.org.
Available: https://scikit-learn.org/stable/modules/multiclass.html.

[5] Adam, Wendy Kan. (2016). Expedia Hotel Recommendations. Kaggle.
Available: https://www.dataquest.io/blog/kaggle-tutorial/

[6] Gourav G. Shenoy, Mangirish A. Wagle, Anwar Shaikh "Expedia Hotel Recommendations,"
Available: https://arxiv.org/pdf/1703.02915.pdf