Predicting Airbnb New User Booking

Arun Nalla Reddy, Darun Arumugham, Kedareshwara Kartikeya Rao Pagadala

***Abstract* - By providing more personalized recommendations and aiding in improved demand forecasting, the correct prediction reduces the typical booking time. In this research paper, we present the best method we have found so far for predicting the Airbnb new user's booking location, which will enable us to determine the location of the new user's first Airbnb booking. We first provide our findings about the dataset properties provided by the Airbnb on Kaggle competition, and then we discuss our feature selection methodology and various modelling approaches based on these findings. By constructing a classification report for each model that includes its precision, recall, and f1 score, we can compare these models. Since the data in our dataset is unbalanced and the accuracy score is insufficient for evaluation, these metrics are crucial.**

Keywords

Decision Tree, Random Forest, Exploratory data analysis Extra Trees, Classification, Supervised Learning, XGBoost, Data visualization.

1. INTRODUCTION

1.1 Background

Airbnb is an American company that operates an online marketplace for lodgings, primarily homestays for vacation rentals, as well as tourism-related activities. San Francisco, California-based platform is accessible by mobile app as well as website. Neither the company nor its agents own any of the real estate listings or event venues; they act as brokers and receive commissions for each booking.[9]

On Airbnb, users can book places to stay in more than 34,000 cities in more than 190 countries. As a result, Airbnb can deliver more personalized content to their community, decrease the average time to first booking, and forecast demands better by accurately anticipating where a new user will book their first travel experience.

Since 2008, Airbnb has helped travelers and hosts find new places to go and offer a more distinctive, personalized way to see the world. Today, Airbnb is a unique service that the entire globe uses and recognizes. [9] For the business, data analysis on the millions of listings offered through Airbnb is essential. There is a lot of data produced by these millions of listings that can be studied.

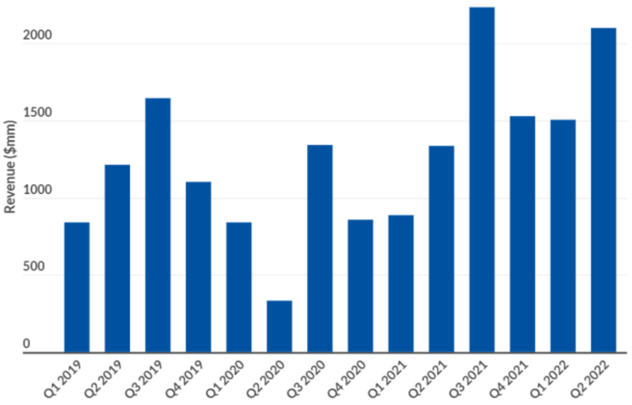


Fig 1. Airbnb quarterly revenue 2019 to 2022[9]

Airbnb has been one of the platforms that has been most badly impacted by the COVID-19. At the height of the pandemic, new bookings were down by 85%, but app usage has gradually increased as governments have loosened travel restrictions. When users together booked 103.7 million stays in the second quarter of 2022, Airbnb broke its own booking record (including Experiences).

Having a clear understanding of the behavior of its users is one of the most valuable pieces of information a company can have. It is often important to understand the customer's buying patterns and trends before making a strategic decision. Moreover, the internet has connected the world, and our data is increasingly being shared and used for marketing campaigns online. It is no secret that data has become one of the most valuable commodities of the 21st century.

Using machine learning, a system can be developed that will accurately predict where the user will book their first time and subsequently decrease the average booking time and improve the booking experience. Airbnb can effectively use this information to improve its marketing techniques and increase bookings.

1.2 Problem Definition

The problem that the case study addresses are predicting where a user will most likely book when they are booking for the first time. With accurate predictions, booking times are reduced by sharing more personalized recommendations as well as forecasting demand in a timelier manner. Using the browser session data along with the user's demographic information, we develop features that assist in solving the problem. All users in this dataset are all from the United States. There are 12 possible outcomes of the destination country: 'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL','DE', 'AU', 'NDF' (no destination found), and 'other'. In contrast to "other," which indicates that a booking was made but was for a destination not on the list, "NDF" indicates that no booking was made.[4] The method we used to approach the issue and develop a model for foretelling user intents is discussed in this paper.

1.3 Our Solution

We first perform a dataset analysis by showing several attributes. By putting missing values and outliers into the data, we do data cleaning and preprocessing. The correlation graph is then shown to identify features, and exploratory data analysis is implement to find different trends and insights using various data visualization techniques which couldn’t be found in traditional statistical methods. We use supervised learning algorithms for model selection. We resample the dataset based on the imbalance in the data. We choose five models based on various criteria, then examine and evaluate how well each of these methods performs. By adjusting these algorithms' hyperparameters and gauging performance using f1 scores, we improve them.

1.4 Related Work

They ignored the session dataset in [5] and just used the user dataset. In order to anticipate the destination countries, they used the XG boost algorithm. The NaN values were removed from the data, the timestamps were divided, and all non-numeric data, such as gender, language, device kind, browser type, etc., was categorized. Due to the lack of user sessions in their modelling, this strategy performs poorly. When the confusion matrix is plotted, it shows that there is a high rate of misclassification because the model can only predict two labels, NDF and US, in every example.

In a different study [6], they link the user dataset with the session’s dataset, selecting the number and kind of actions taken by each user from the session’s dataset. By removing a few columns and replacing any NaN values with -1, the data is cleaned. Additionally, they used the XG Boost with the following parameters: learning rate = 0.1, number = 200, and max depth = 5. Performance is enhanced over [8] as a result of this.

2. DATA EXPLORATION

2.1 Dataset Analysis

The Kaggle competition for the New User Bookings challenge includes this Airbnb dataset. To aid with the forecast of the journey destination country, it contains 5 datasets. This dataset contains data such as user web session records, demographics of particular nations, and some summary statistics.

The data contains three .csv files as mentioned below,

- train_users.csv — This is our train data set, which contains 213451 data points including id, date_account_created, timestamp_first_active, age, gender, ..., country_destination. In this case, country_destination is the target value we need to predict for a given user.
- Test_users.csv — Our test data includes 62096 data points and the same features as train data, which we have to use during the testing time.
- sessions.csv — This file contains web session

logs for users, including information such as user_id, action, action_type, and the total number of data points is 10567736. The data contains users' information from 2014.

The user's personal information, such as age, gender, signup method, language, country destination (target), etc., is contained in the first two files. Users' web session information is contained in the sessions.csv file. The user id field, which corresponds to the id field of the train datasets, uniquely identifies each record in this dataset. We discover a number of session records that include data from the several occasions the specific user has accessed the Airbnb program.

Unlike the training dataset, which contains records going back to 2010, the sessions.csv contains data on users from 2014 and after. This implies that many users' session records are not accessible. The session dataset only contains entries for 35% of the train users and 99% of the test users.

The dataset is seriously skewed. A booking has not yet been made by 58.35% of the consumers. A significant percentage of users (29.22%) booked for a destination in the "US" among those who had. France ('FR'), which is listed among the non-US nations, has a sizable stake of 2.35%. Additionally, a sizable portion of people (4.73%) went somewhere that wasn't on the list of alternatives.[3]

2.2 Data Pruning

One method for deleting useless parameters to address overfitting issues is data pruning. Six columns—user id, action, action type, action detail, device type, and secs elapsed—make up the session csv file in Figure 4. We removed extra fields like device type and action detail. The sessions.csv file contains a large number of users who repeatedly perform various tasks. We totaled up all of a user's actions and stored them by mapping them to their user id. Finally, we entered 0 in place of the missing information in session.csv.

2.3 Data Cleaning and Preprocessing

Data cleaning is the process of locating and fixing mistakes in a dataset that could harm the model. Data cleaning entails removing outliers and null values from the dataset. The dataset is cleaned using the next few stages.

1. Merging the dataset
2. Filling missing values
3. Change date format
4. Remove Outliers

We initially map the user details and sessions datasets using the user id to produce a single dataset. After that, we go on to cleaning and preparing the data.

A. *Filling missing values*

First, filling in or replacing missing values is a part of cleaning a dataset. The dataset's missing values are by default rendered as NaN. (Not a number). To look for empty fields, we utilised the functions isnull() and notnull(). We have used -1 to fill in the null values for the secs elapsed, action, and action type columns. Age null values are filled using the median age, which is 34, which is derived to fill the null values. There are two variables utilised in the gender column: other and unknown.

B. *Change date format*

The user datasets comprise three fields: date account created timestamp first active, and date first booking, each of which contains data on a distinct date. Since none of the test users will have booked a trip, this date initial booking only has value for users who have. It is therefore useless for prediction. The field will be eliminated since it contains 58% null values. The other date fields contain information on the user's first visit to the website and their decision to open an account.[2] We must switch the format to one that is simple to edit before we can clear these fields. Currently, this data is kept as strings, but we convert it to easily editable date objects. Therefore, date_account_created and timestamp_first_active are separated by year, month, and date.

C. *Removed Outliers*

Outliers make the dataset more variable, which reduces statistical power. As a result, eliminating outliers might make the findings statistically significant. The training process can be hampered and misled by outliers in the data, which leads to longer training times, fewer accurate models, and subpar results. The age field contains outliers in the training sample that range from below 15 to above 110. The boxplot of "age" is plotted in Fig. 2 to demonstrate the presence of outliers. After removing outliers, the boxplot of "age" is plotted in Fig. 3.

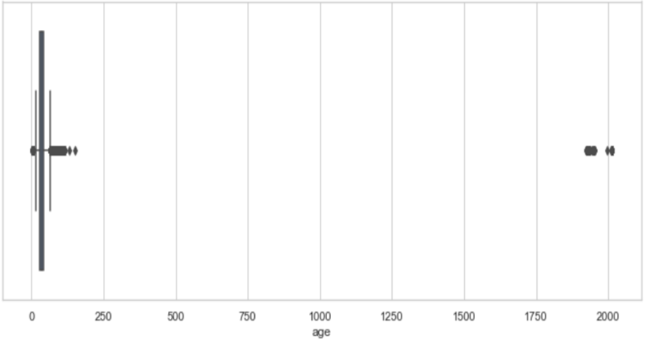


Fig 2. Age with outliers

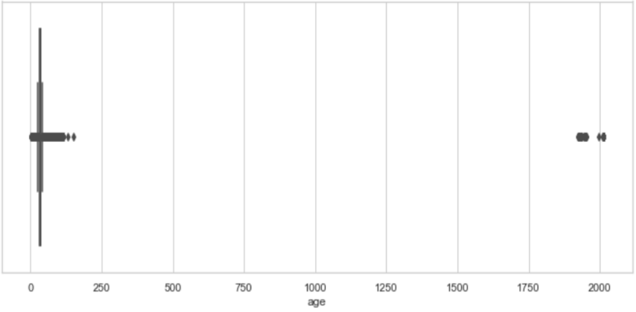


Fig 3. Airbnb without outliers

2.4 Skewness

If the skewness is positive, the data is positively skewed or skewed right. The data is negatively skewed or skewed left when skewness is negative. The data are totally symmetrical when skewness is equal to zero. When the data is too skewed, many statistical models break down. We know that outliers have an impact on the performance of the statistical model, and the tail portion of skewed data may act as an outlier for the model.

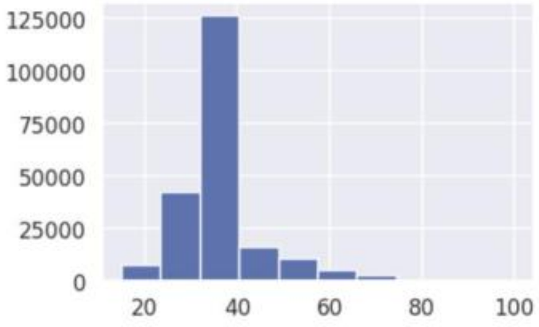


Fig 4. Right skewed data of age

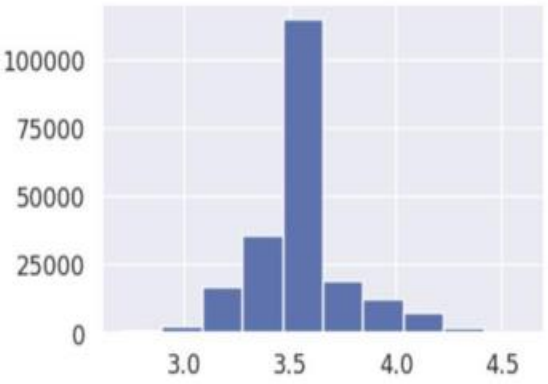


Fig 5. After removing skewness

2.5 Feature Parsing

After eliminating the NaN data, we start looking into train user df's fundamental data. Numerous features in train users use strings rather than numbers to present various quantities, as we have seen. The computer needs to be able to recognise important patterns in the data in order for the models to function properly, therefore we encode them using the one-hot technique. Data is transformed into a distribution with a mean of 0 and a standard deviation of 1 using the standard scaler.

2.6 Feature Selection

We perform feature selection for three reasons:

- To minimize overfitting - With less redundant data, there is less chance of making decisions based on noise.
- To increase Accuracy - Less deceptive data gives better results.
- Reduce training time - Lesser data means less time for analysis.

Low variance features need to be eliminated. The 18 features include id, date account created timestamp first active, date first booking, gender, age, signup method, signup flow, language, affiliate channel, affiliate provider, first affiliate tracked, signup app, first device type, first browser, country destination, action, and secs elapsed after combining features from the session dataset with training and training dataset.

The correlation matrix explains the relationship between two or more variables. These factors can be characteristics of the raw data that were utilised to forecast our target variable. In the event that two variables have a strong correlation, we can predict one from the other. Figure 6 illustrates how we develop a correlation matrix to indicate whether variables have a strong or low correlation with one another. In order to see if there is any apparent association between the features, we displayed the distribution of the independent variables. High correlation can cause model results to be incorrect.

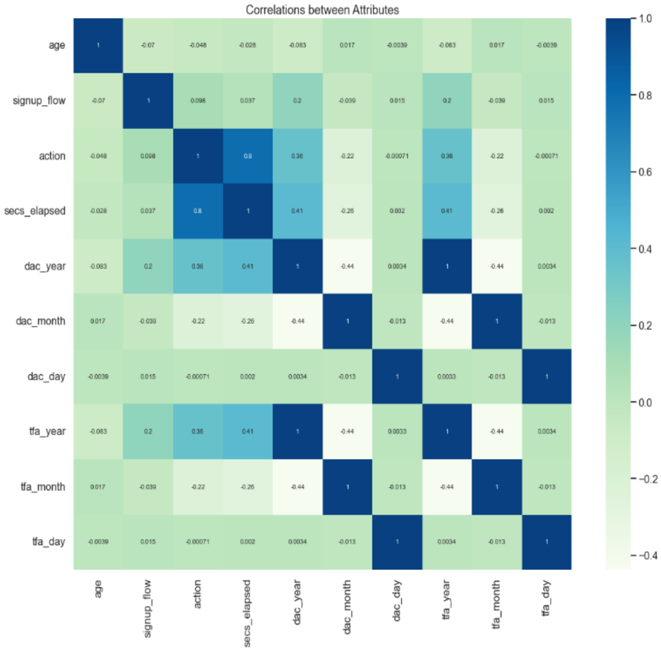


Fig 6. Correlation matrix

3. EXPLORATORY DATA ANALYSIS

An exploratory data analysis involves performing initial investigations on data with the help of summary statistics and graphical representations so that patterns can be discovered, anomalies can be spotted, hypotheses can be tested, and assumptions can be tested.

3.1 Destination country

The dataset is highly imbalanced. Over half of our customers (58.35%) have not made a reservation. There is a large share of users (29.22%) who have booked a destination in 'US' among those who have. Out of the countries given outside of the US, France ('FR') holds a high share of 2.35%. A significant percentage of users (4.73%) travelled to a destination that wasn't listed. The data indicates that US travelers are more likely to travel within the US.[7]

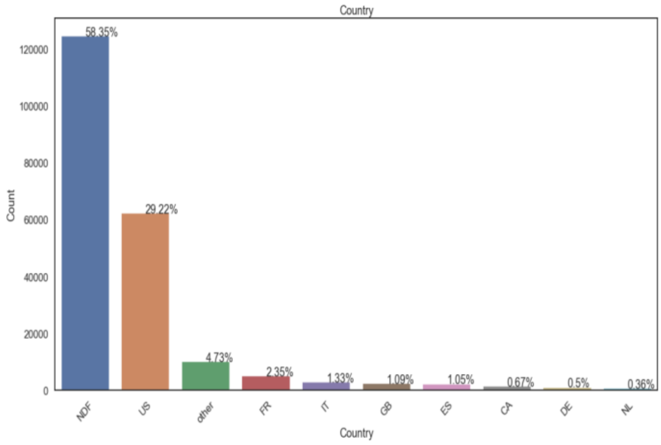


Fig 7. Percentage distribution of destination

3.2 Gender

It is reported that 44.83 percent of users have not disclosed their gender. There are marginally more female users than male users among those who have, with 29.53 % female users and 25.55 % male users. There are a small number of users (0.13%) who have selected 'other' as their gender. Consequently, they are likely to be non-binary.

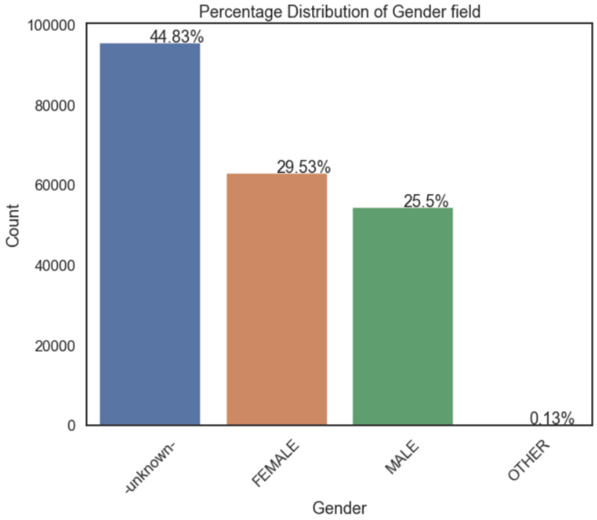


Fig 8. Percentage distribution of gender

3.3 Age Insight

A total of 87,990 users have not disclosed their age. Users who are planning trips to the locations shown in the graph do not differ significantly in age. However, individuals who plan trips to the United Kingdom appear to be somewhat older than those who book excursions to Spain and the Netherlands.

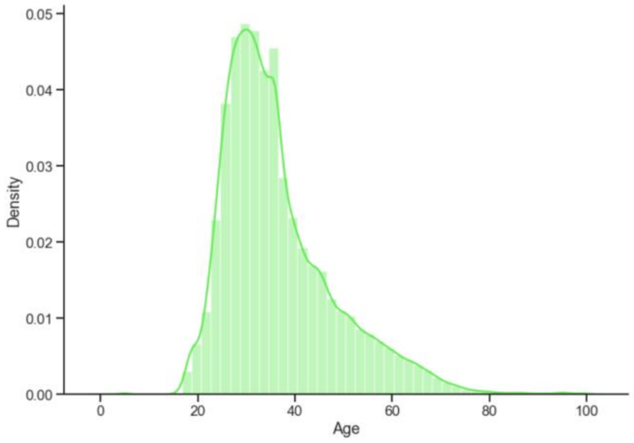


Fig 9. Age distribution

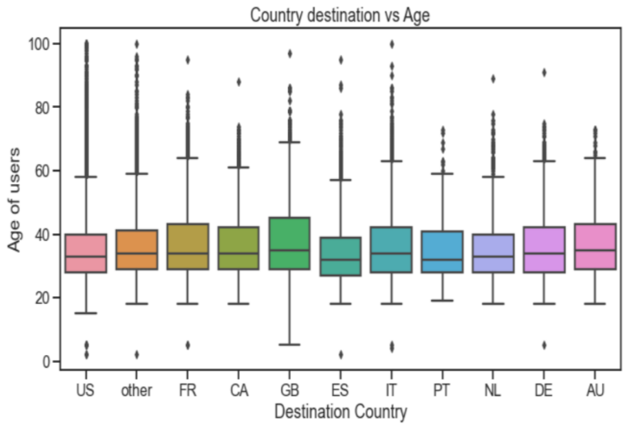


Fig 10. Country destination vs Age

3.4 Language

Most of the users' primary language is English (96.66%). Since the data comes from the US, this behaviour is very understandable given that most people there identify English as their first language. This field is giving good insight as expected. There is not much analysis could be done in language field from the dataset.[7]

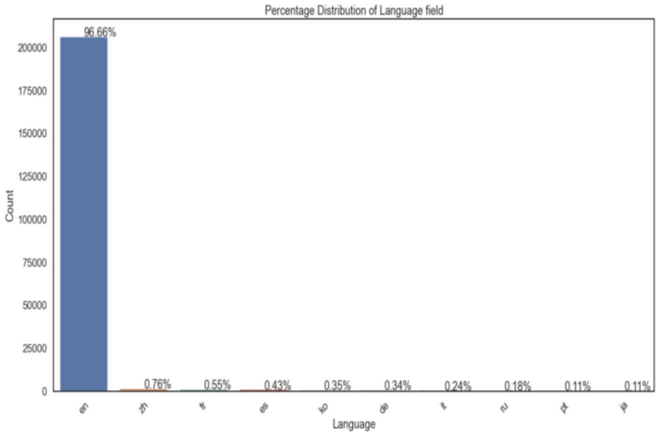


Fig 11. Percentage distribution of language

3.5 Sign-Up features

Most users (71.63%) created their Airbnb accounts via the basic process. This indicates that they registered using the standard email-password approach. Many of the remaining users (28.11%) signed up via Facebook. Google was utilized by a very small number of users (0.26%) to sign up.[7]

This suggests that using Google to join up is not very popular. Additionally, it may indicate that consumers have selected the straightforward email-password method of accessing their Airbnb account because they are less likely to give their personal information via a facebook or google account linkup. Regardless of the destination country selected, the majority of users who made at least one booking utilized the Airbnb email method to sign up.

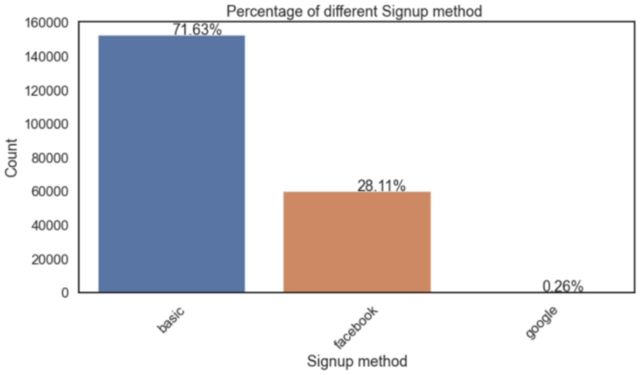


Fig 12. Percentage distribution of signup method

Most users (85.6%) created accounts using the web. Out of the others, "iOS" is just slightly more popular than "Moweb" and "Android" (around 5.98%). Moweb and Android are not very popular apps to sign up for because just 2.93% and 2.56% of users use them, respectively.

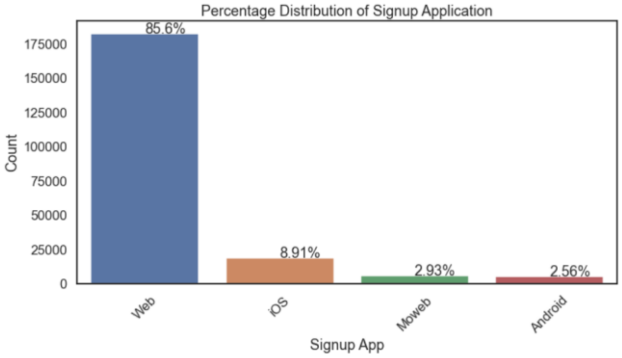


Fig 13. Percentage distribution of signup application

3.6 Device Used

The most common device for creating accounts is the "Mac Desktop," which is utilised by 41.98% of the users. With 34.07% of clients, "Windows Desktop" comes in second place. In the mobile world, the category "iPhone" is more popular than "Android Phone" of the remaining subcategories. In the tablet market, "iPad" is more well-known than "Android Tablet." Overall, this plot indicates that across all categories, Apple products are more popular than Android ones.[7]

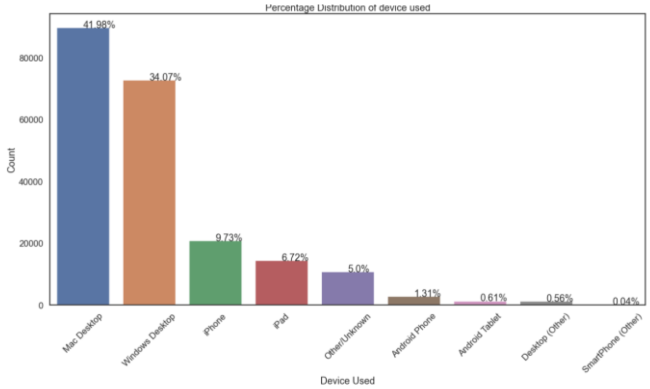


Fig 14. Percentage distribution of device used

3.7 *Browser used*

This field records which browser the user used to access the program for the first time. Chrome Browser has a sizable market share (29.91%), followed by Safari (21.16%) and Firefox (15.77%). It is clear from the previous plot that "Mac Desktop" is the most widely used device among users. This plot reveals that a sizable portion of the customer base preferred Google's Chrome web browser over Apple's own Safari web browser, even on Apple products.

Another intriguing data is that 12.77% of customers do not know who their first browser was.

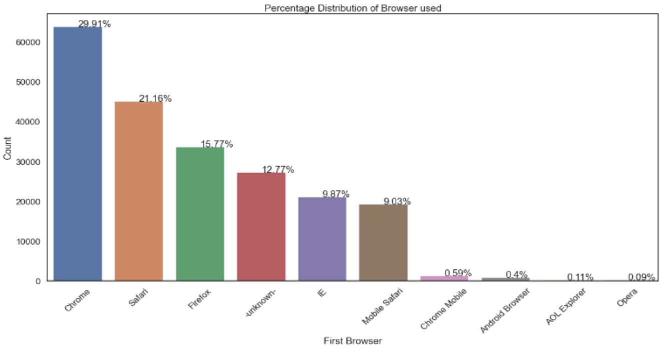


Fig 15. Percentage distribution of browser

4. METHODOLOGY

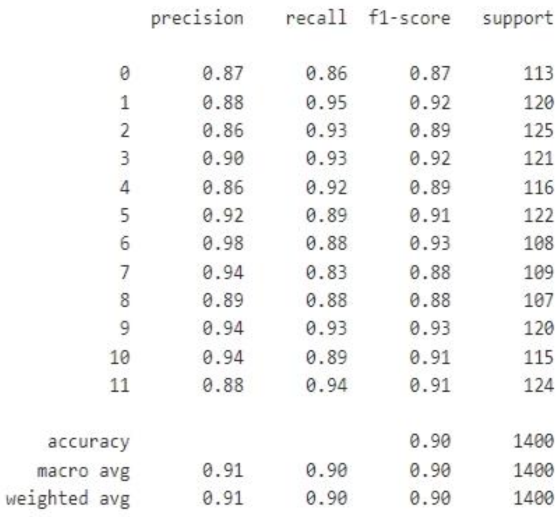
The discrete and categorical destination country variable is our target variable. The country the user would choose to book their next destination in for each entry must be predicted using user data. This is a multinomial classification problem because the destination country has more than three possible values.

To determine which categorization model fits our situation the best, we compare the performance of four different models. We adjust the hyperparameters of each of these algorithms to achieve the best performance in order to increase performance.

4.1 Support vector machine

One of the most well-liked supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression problems. However, it is largely employed in Machine Learning Classification issues. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary.

SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method.

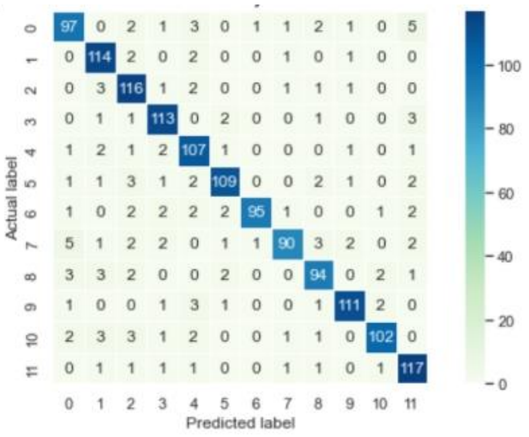


```
              precision    recall  f1-score   support

           0       0.87      0.86      0.87       113
           1       0.88      0.95      0.92       120
           2       0.86      0.93      0.89       125
           3       0.90      0.93      0.92       121
           4       0.86      0.92      0.89       116
           5       0.92      0.89      0.91       122
           6       0.98      0.88      0.93       108
           7       0.94      0.83      0.88       109
           8       0.89      0.88      0.88       107
           9       0.94      0.93      0.93       120
          10       0.94      0.89      0.91       115
          11       0.88      0.94      0.91       124

    accuracy                           0.90      1400
   macro avg       0.91      0.90      0.90      1400
weighted avg       0.91      0.90      0.90      1400
```

Fig 16. Support Vector Machine report

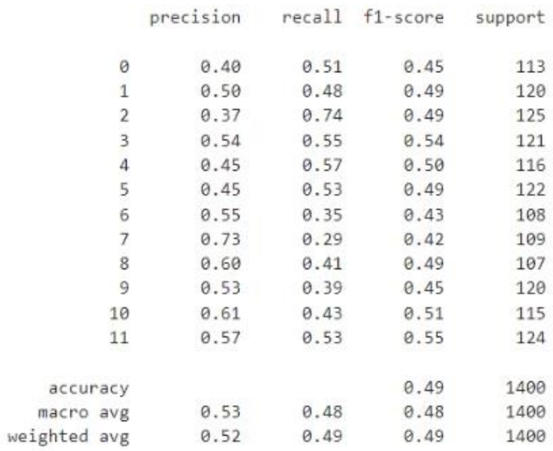


	0	1	2	3	4	5	6	7	8	9	10	11
0	97	0	2	1	3	0	1	1	2	1	0	5
1	0	114	2	0	2	0	0	1	0	1	0	0
2	0	3	116	1	2	0	0	1	1	1	0	0
3	0	1	1	113	0	2	0	0	1	0	0	3
4	1	2	1	2	107	1	0	0	0	1	0	1
5	1	1	3	1	2	109	0	0	2	1	0	2
6	1	0	2	2	2	2	95	1	0	0	1	2
7	5	1	2	2	0	1	1	90	3	2	0	2
8	3	3	2	0	0	2	0	0	94	0	2	1
9	1	0	0	1	3	1	0	0	1	111	2	0
10	2	3	3	1	2	0	0	1	1	0	102	0
11	0	1	1	1	1	0	0	1	1	0	1	117

Fig 17. Support Vector Machine matrix

4.2 K Nearest Neighbor

The supervised learning technique K-nearest neighbors (KNN) is used for both regression and classification. By calculating the distance between the test data and all of the training points, KNN tries to predict the proper class for the test data. Then choose the K spots that are closest to the test data. The KNN method determines which classes of the "K" training data the test data will belong to, and the class with the highest probability is chosen. The value in a regression situation is the average of the 'K' chosen training points.

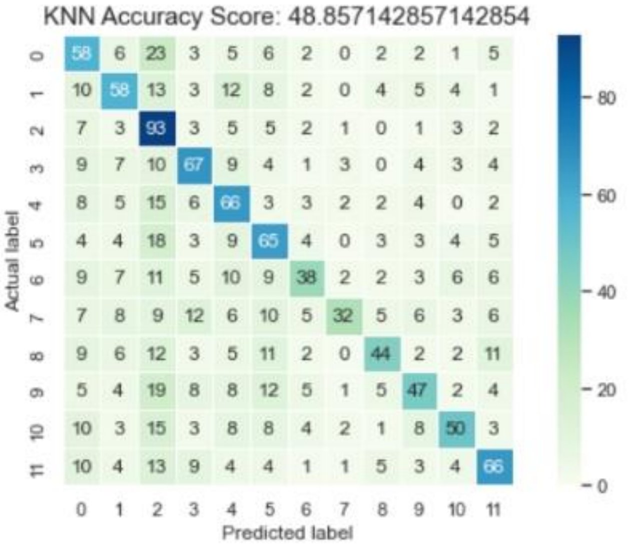


```
              precision    recall  f1-score   support

           0       0.40      0.51      0.45       113
           1       0.50      0.48      0.49       120
           2       0.37      0.74      0.49       125
           3       0.54      0.55      0.54       121
           4       0.45      0.57      0.50       116
           5       0.45      0.53      0.49       122
           6       0.55      0.35      0.43       108
           7       0.73      0.29      0.42       109
           8       0.60      0.41      0.49       107
           9       0.53      0.39      0.45       120
          10       0.61      0.43      0.51       115
          11       0.57      0.53      0.55       124

    accuracy                           0.49      1400
   macro avg       0.53      0.48      0.48      1400
weighted avg       0.52      0.49      0.49      1400
```

Fig 18. K Nearest Neighbor Algorithm Classification report



	0	1	2	3	4	5	6	7	8	9	10	11
0	58	6	23	3	5	6	2	0	2	2	1	5
1	10	58	13	3	12	8	2	0	4	5	4	1
2	7	3	93	3	5	5	2	1	0	1	3	2
3	9	7	10	67	9	4	1	3	0	4	3	4
4	8	5	15	6	66	3	3	2	2	4	0	2
5	4	4	18	3	9	65	4	0	3	3	4	5
6	9	7	11	5	10	9	38	2	2	3	6	6
7	7	8	9	12	6	10	5	32	5	6	3	6
8	9	6	12	3	5	11	2	0	44	2	2	11
9	5	4	19	8	8	12	5	1	5	47	2	4
10	10	3	15	3	8	8	4	2	1	8	50	3
11	10	4	13	9	4	4	1	1	5	3	4	66

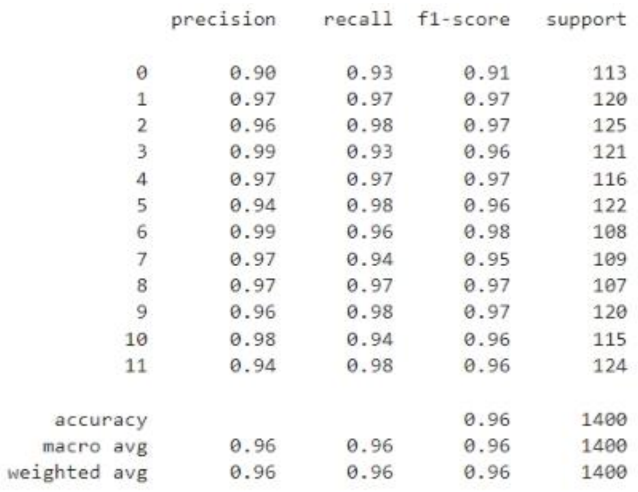
Fig 19. Confusion matrix of K Nearest Neighbor

4.3 Gradient Boosting Classifier

A machine learning method called gradient boosting is used, among other things, for classification and regression tasks. It provides a prediction model in the form of an ensemble of decision trees-like weak prediction models. The resulting technique, known as gradient-boosted trees, typically beats random forest when a decision tree is the weak learner. The construction of a gradient-boosted trees model follows the same stage-wise process as previous boosting techniques, but it generalizes other techniques by enabling the optimization of any differentiable loss function

An ensemble machine learning algorithm is gradient boosting. This kind of algorithm combines a number of weaker models to create a strong model. Any model that performs only marginally better than chance alone is considered weak. Adding additional models to existing models to fix faults is known as boosting in ensemble strategies. Up until there are no more enhancements to be made, models are uploaded in a sequential order. The fact that gradient boosting employs a gradient descent strategy to reduce loss while introducing new models is whence it gets its name.

We have created the most well-liked Xgboost algorithm out of various gradient boosting implementations. Extreme Gradient Boosting is what it stands for. Gradient, the partial derivative of the loss function, describes the steepness of the error function in each training round. The gradient is used to identify which direction to move the model parameters in order to (maximally) reduce the error in the upcoming training cycle by "descending the gradient." It is designed to be incredibly effective as well as computationally efficient (fast to perform).

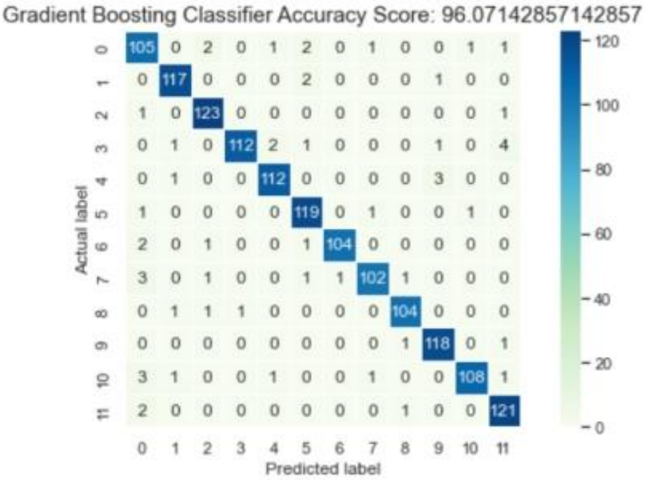


```
              precision    recall  f1-score   support

           0       0.90      0.93      0.91       113
           1       0.97      0.97      0.97       120
           2       0.96      0.98      0.97       125
           3       0.99      0.93      0.96       121
           4       0.97      0.97      0.97       116
           5       0.94      0.98      0.96       122
           6       0.99      0.96      0.98       108
           7       0.97      0.94      0.95       109
           8       0.97      0.97      0.97       107
           9       0.96      0.98      0.97       120
          10       0.98      0.94      0.96       115
          11       0.94      0.98      0.96       124

    accuracy                           0.96      1400
   macro avg       0.96      0.96      0.96      1400
weighted avg       0.96      0.96      0.96      1400
```

Fig 20. Gradient Boosting Classifier Algorithm Classification report



	0	1	2	3	4	5	6	7	8	9	10	11
0	105	0	2	0	1	2	0	1	0	0	1	1
1	0	117	0	0	0	2	0	0	0	1	0	0
2	1	0	123	0	0	0	0	0	0	0	0	1
3	0	1	0	112	2	1	0	0	0	1	0	4
4	0	1	0	0	112	0	0	0	0	3	0	0
5	1	0	0	0	0	119	0	1	0	0	1	0
6	2	0	1	0	0	1	104	0	0	0	0	0
7	3	0	1	0	0	1	1	102	1	0	0	0
8	0	1	1	1	0	0	0	0	104	0	0	0
9	0	0	0	0	0	0	0	0	1	118	0	1
10	3	1	0	0	1	0	0	1	0	0	108	1
11	2	0	0	0	0	0	0	0	1	0	0	121

Fig 21. Confusion matrix for Gradient Boosting Classifier

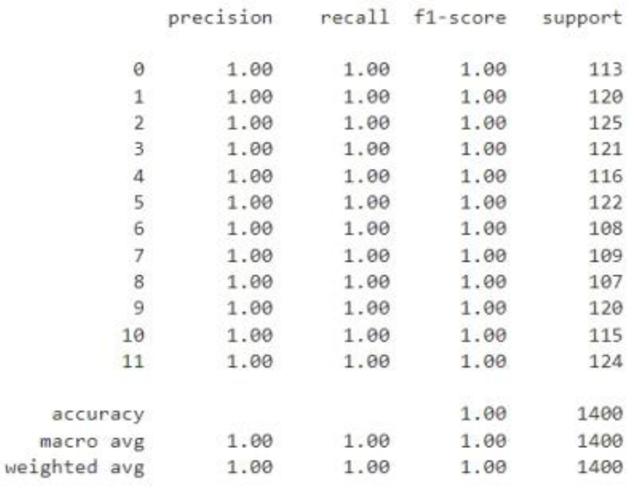
4.5 Extra Trees Classifier Algorithm

Extremely Randomized Trees is also known as Extra Trees Classification. This algorithm creates a large number of unpruned decision trees using the training dataset. An ensemble machine learning system called Extra Trees aggregates the forecasts from numerous decision trees.

It has a connection to the common Random Forest algorithm. Even though it uses a simpler technique to create the decision trees that are utilised as members of the ensemble, it can frequently produce performance that is as good as or better than the random forest algorithm.

Given that it only contains a few essential hyperparameters and good heuristics for tuning these hyperparameters, it is also simple to use.

The ultimate label for classification issues is chosen by majority vote. It is a different kind of classifier than the random forest. Both produce several trees and divide nodes according to random feature subsets. There are two key differences between the two: additional trees do not bootstrap the observations, in contrast to Random Forest. This indicates that it is appropriate for the full training set's decision. Additionally, Random Forest employs a greedy strategy while additional trees choose the split point at random in order to choose an ideal split point. In addition, compared to the random forest approach, it results in faster execution of Extra trees.

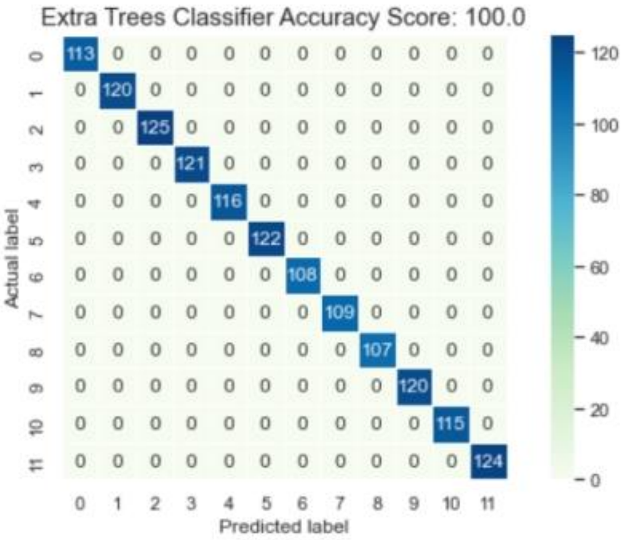


```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       113
           1       1.00      1.00      1.00       120
           2       1.00      1.00      1.00       125
           3       1.00      1.00      1.00       121
           4       1.00      1.00      1.00       116
           5       1.00      1.00      1.00       122
           6       1.00      1.00      1.00       108
           7       1.00      1.00      1.00       109
           8       1.00      1.00      1.00       107
           9       1.00      1.00      1.00       120
          10       1.00      1.00      1.00       115
          11       1.00      1.00      1.00       124

    accuracy                           1.00      1400
   macro avg       1.00      1.00      1.00      1400
weighted avg       1.00      1.00      1.00      1400
```

Fig 22. Extra Trees Classifier Algorithm Classification report



	0	1	2	3	4	5	6	7	8	9	10	11
0	113	0	0	0	0	0	0	0	0	0	0	0
1	0	120	0	0	0	0	0	0	0	0	0	0
2	0	0	125	0	0	0	0	0	0	0	0	0
3	0	0	0	121	0	0	0	0	0	0	0	0
4	0	0	0	0	116	0	0	0	0	0	0	0
5	0	0	0	0	0	122	0	0	0	0	0	0
6	0	0	0	0	0	0	108	0	0	0	0	0
7	0	0	0	0	0	0	0	109	0	0	0	0
8	0	0	0	0	0	0	0	0	107	0	0	0
9	0	0	0	0	0	0	0	0	0	120	0	0
10	0	0	0	0	0	0	0	0	0	0	115	0
11	0	0	0	0	0	0	0	0	0	0	0	124

Fig 23. Confusion matrix for Extra tree classifier

5. CONFUSION MATRIX

A Confusion matrix is an N x N matrix is used to assess the effectiveness of a classification model, where N is the total number of target classes. In the matrix, the actual goal values are contrasted with those that the machine learning model anticipated. This gives us a comprehensive understanding of the effectiveness of our classification model and the types of mistakes it is committing. Precision reveals the proportion of correctly predicted cases that actually resulted in a favourable outcome. Recall reveals the proportion of real positive cases that our model was able to properly anticipate.

In order to evaluate each model, we look at its performance on the test data in terms of precision, recall, and f1 score. In order to better comprehend each of their performances, we also develop a confusion matrix for them.

Precision and recall are intended to be balanced by the F1-score measure, which is particularly helpful when working with datasets that are not balanced. It is determined as

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

6. FUTURE WORK

Most of the accommodation results are either NDF(No destination found) or the US. More data for individual users' web activities should be collected that will surely be useful to improve the accuracy of the prediction. This kind of datasets analysis and model building could play a crucial role when an organization wants to know how a new user is going to use their services and based on the user preferences and usage of user, we can target customized deals or features for user which would increase the scope of attracting users to the services.

7. CONCLUSION

We propose to use an Extra trees classifier algorithm for such travel destination prediction tasks. This algorithm gives the best result on test data with an F1 score of 0.664. While exploring the dataset we discovered several data present in the dataset were empty and invalid data for example there are case where user age is between 1 and age>100, which is technically not possible, so in order to make the predictions and dataset more real like we have used techniques like dropping and replacing the data with median or mode. We have successfully performed data visualization and plotted correlation between the attributes of the combined dataset, as a final step we have divided the data for testing and training after which we have used various algorithms like support vector machine, k nearest neighbours, gradient boosting classifier and extra trees classifier where we got highest accuracy for gradient boosting classifier. Data cleaning, preprocessing, exploratory data analysis, feature selection greatly improves the performance of the model.

8. ACKNOWLEDGEMENT

A huge thanks to the University of Windsor, Ontario, Canada, the computer science department and to Dr. Robin Gras for the guidance.

8. REFERENCES

[1] Hugo Ulfsson “Predicting Airbnb user's desired travel destinations”, Stockholm, 2017.

[2] Srinivas Avireddy, Sathya Narayanan Ramamirtham, Sridhar Srinivasa Subramanian “Predicting Airbnb user destination using user demographic and session information”, UC San Diego

[3] https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings/overview

[4] https://infogram.com/airbnb-revenue-1hxr4zx9plzdo6y

[5] https://www.kaggle.com/manas13/airbnb-new-user-bookings

[6] https://www.kaggle.com/justk1/airbnb

[7] https://www.kaggle.com/code/nikhiljangam/exploratory-data-analysis

[8] https://towardsdatascience.com/predicting-destination-countries-for-new-users-of-airbnb-eb0d7db7579f

[9] https://ipropertymanagement.com/research/airbnb-statistics