Data Analytics Project - Airbnb New User Bookings

Report

Anamika Sen
Virginia Tech
Blacksburg, Virginia
anamikas@vt.edu

Aidin Ferdowsi
Virginia Tech
Blacskburg, Virginia
aidin@vt.edu

ABSTRACT

New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. By accurately predicting where a new user will book their first travel experience, Airbnb can share more personalized content with their community, improve the customer service, and better forecast demand. In this project, a prediction model is proposed to predict that in which country a new user will make his or her first booking. We have applied four classifiers: naive Bayes, decision trees, k nearest neighbor, and Ensemble model to the provided dataset. The evaluation criteria in this project, is the NDCG score which is based on the relevance of top 5 results to a query. Our results show that decision trees and ensemble models provide a higher performance by having an NDCG score of higher than 0.82.

CCS CONCEPTS

• **Information systems** → *Data management systems*;

KEYWORDS

Data Analytics, Classification, Prediction

1 PROBLEM STATEMENT

In this project we aim to predict the Airbnb new users' first booking destination at the granularity of country[3]. There are 12 destinations to choose from, including 'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL', 'DE', 'AU', 'NDF' (no destination found), and 'other'. Users who haven't made a booking are categorized with the "NDF" label in the destination field.

2 DATA DESCRIPTION

We are using the dataset from the **Airbnb Recruiting: New User Bookings completion** of Kaggle for this experiment. In the dataset, Airbnb provides a labeled training set of 171239 entries, and a testing set of 43673 entries to predict with. There are 2 other statistical data sheets regarding some geographical information about each of the listed countries and their population age and gender distribution. A session set documented the time elapsed for each client side and server side actions, with no references provided to explain how the naming matches with operations. The training and testing sets are split by dates. Even though a testing dataset was provided in the competition, the dataset did not contain the target variable to be predicted, and thus we chose to divide the training dataset for training and testing.

To sum up, all information is listed as follows:

- train_users.csv:

(1) **id**: user id.
(2) **date_account_created**: the date of account creation.
(3) **timestamp_first_active**: timestamp of the first activity, note that it can be earlier than date_account_created or date_first_booking because a user can search before signing up.
(4) **date_first_booking**: date of first booking.
(5) **gender**: This attribute had many unknown values.
(6) **age**
(7) **signup_method**
(8) **signup_flow**: the page a user came to signup up from language: international language preference.
(9) **affiliate_channel**: what kind of paid marketing.
(10) **affiliate_provider**: where the marketing is e.g. google, craigslist, other.
(11) **first_affiliate_tracked**: whats the first marketing the user interacted with before the signing up .
(12) **signup_app**: which app did the user use to signup for Airbnb.
(13) **first_device_type**
(14) **first_browser**
(15) **country_destination**: this is the target variable you are to predict .

- sessions.csv: web sessions log for users.

(1) userid: to be joined with the column 'id' in users table action.
(2) action_type
(3) action_detail
(4) device_type
(5) secs_elapsed

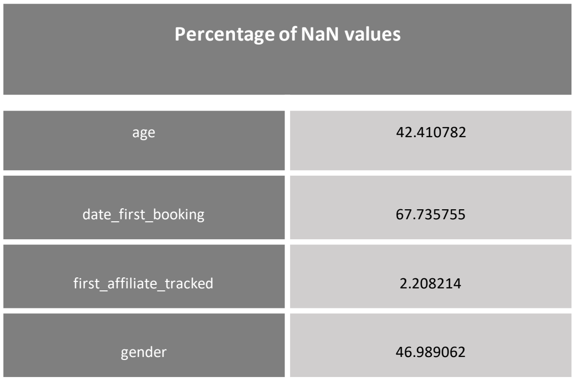
- **countries.csv**: summary statistics of destination countries.
- **age_gender_bkts.csv**: summary statistics of users' age group, gender, country of destination.
- **sample_submission.csv**: correct format for submitting our model to the competition.

3 DATA PRE-PROCESSING

This project was done on MATLAB. Some of the data pre-processing steps that was implemented in this project can be summarized as follows:

3.1 Handling Missing Data

There were a lot of NaN values in the following attributes as shown in Fig. 1. Since the test dataset provided did not have date_first_booking, we decided to discard it. The -unknown- values in gender were replaced by Male, Female and Other in the same distribution as presented in the dataset. Moreover, the age attribute has some unacceptable values such as ages higher than 95 and lower than 16.



Percentage of NaN values	
age	42.410782
date_first_booking	67.735755
first_affiliate_tracked	2.208214
gender	46.989062

Fig. 1: Percentage of NaN values in the dataset

Thus, we replace such values with NaN values. Furthermore, since 67% of date_first_booking attribute is missing from the dataset, we decide to remove this field from our model building. Finally, we replace all the NaN values in first_affiliate_tracked with 'untracked' value, since it is one of the categories of this attribute.

3.2 Time Conversion

Some of the time features has different formats in each table. We convert all of them to a unique format. For instance year is not a very useful feature since it is limited to a few years. While month can give us more information.

3.3 Loading Sessions Data

Since the sessions data for some of the users was provided by Airbnb in a separate file, we add this data as a new feature to our data set. However, since the session data for some users are not available, we leave the session value of those user as NaN. Sessions data was very useful in the classifier as it shows the number of pages one user visits before booking.

3.4 Applying One Hot Encoding

Since we have a combination of nominal and numerical attributes, we apply one-hot encoding on the nominal attributes. One hot encoding is a process by which categorical variables are converted into a form that can be fed to data mining algorithms to do a better job in prediction. The basic idea is to perform a "binarization" on the categorical data. This increases the size of columns as for instance if the number of possibilities for a categorical data is *n*, then we will have *n* columns, with 1 column having value 1 and others being 0. Applying one hot encoding results in having 154 columns, which is compared to our original data set, is a large number.

3.5 Principle Component Analysis

Next, we apply Principle Component Analysis (PCA) to our one hot encoded data. Fig. 2 shows the scree plot of first 10 highest eigenvalues. From this plot we can see that the first eigenvalue has the highest power, thus, we will choose only this eigenvalue while training.

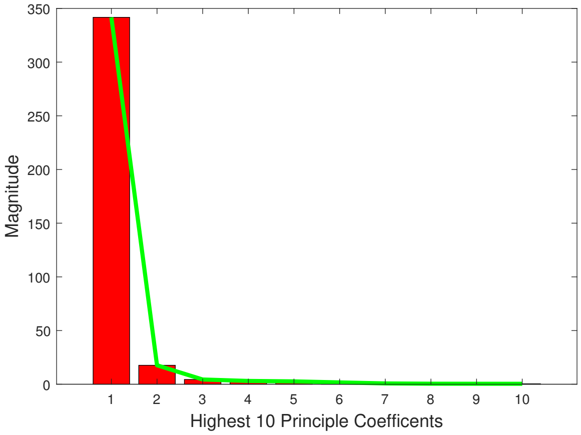


Fig. 2: Principle component analysis on the dataset.

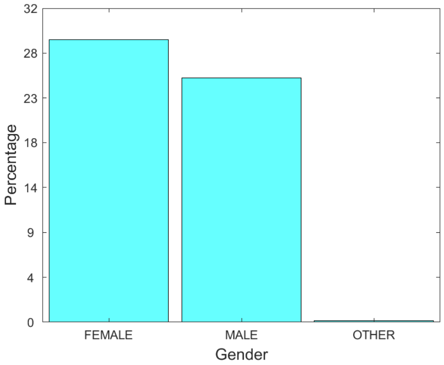


Fig. 3: Gender visualization for the known values.

4 DATA EXPLORATION

Even though we have 15 attributes for each row, however, some of these attributes are missing in some rows. Fig. 1 shows the percentage of missing values for each attribute. We can see from Fig.1 that, based on these percentage values, 62 % of date_first_booking, 42% of age, 47% of gender and 2% of first_affiliate_tracked is missing. Thus using these features will yield a lower performance in built classifier. Moreover from Fig. 3 we can see that while almost 45% of the users did not insert their gender, percentage of female users is higher than the male users.

Fig. 4 shows the age histogram plot. At the first glance, we can observe that not only we have almost half of data missing, but also there are some inputs which are not logical, for instance they are either less than 18 or very high. Thus, we limit our age entries between 16 and 95. We can see that the users between the age of 25-40 tend to use Airbnb more than the other users. Also to better understand the age distribution of the users, we use a box plot as shown in Fig. 5. We can see from this plot that users booking for countries Spain, Portugal and Netherlands tend to be younger while users who book for Great Britain and France tend to be older.

Figure 6 shows the number of accounts created by time. It is obvious that during a particular time, Airbnb's reputation increased which resulted in higher number of account creations. Moreover, we extract the weekdays of account creations from date_account_created

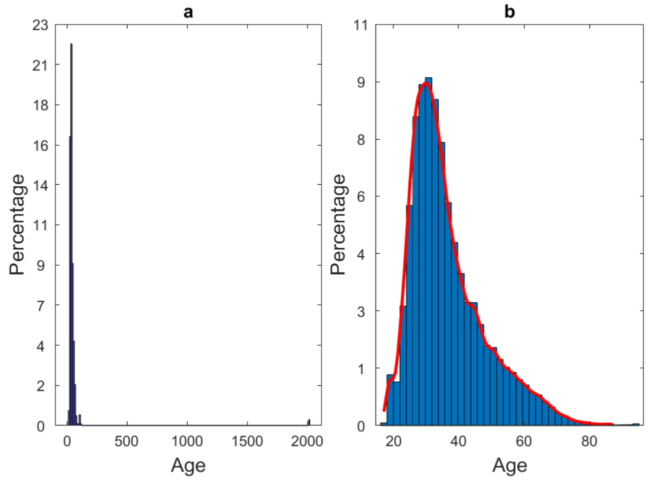


Fig. 4: Age attribute histogram plot: a) for all entries, b) after removing the non valid values.

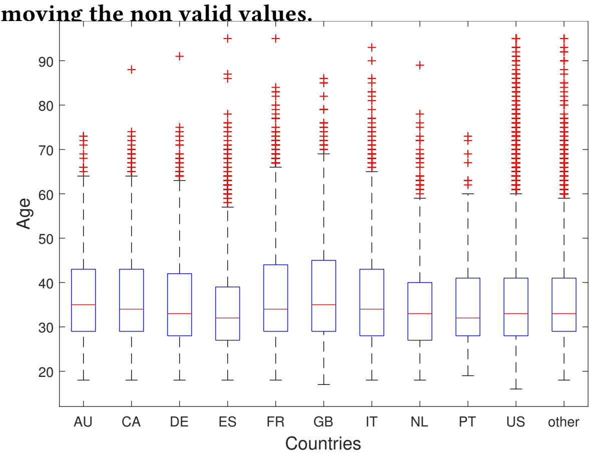


Fig. 5: Box plot of user age distribution based on their country destination.

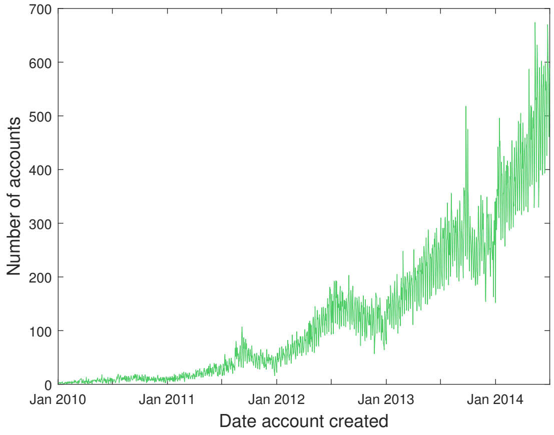


Fig. 6: Account creation date visualization.

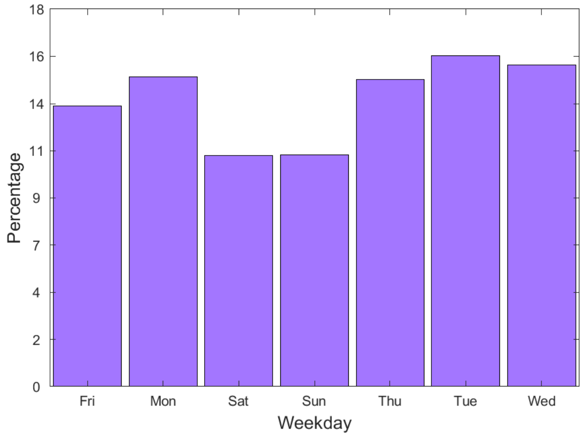


Fig. 7: Weekday of account creation.

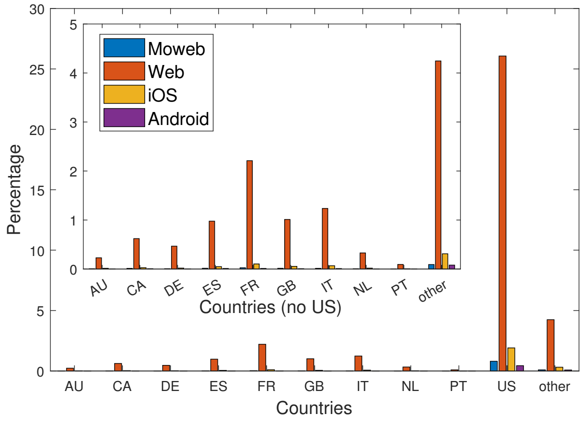


Fig. 8: Destination country VS. user's sign-up app.

in Fig. 7. We can see from this figure that the users are less active on weekends.

Next, we find the destination country distribution and users' signup app in Fig. 8. We can see that most of the users' signup using web, while iOS users have the second most popularity between the users. In addition, we analyze the destination country distribution versus users' signup method in Fig. 9. It can be seen that most users signup directly through the Airbnb application, while the second most applied signup method is using Facebook. In addition, Fig. 9 shows that Google has the least popularity between the users while signing up for Airbnb, while, basic signup method is almost 2.5 times Facebook signup count. However, Fig. 10 shows that even though percentage of users signing up with Google are 0.03%, most users are coming from Google after direct method.

5 MODEL BUILDING

Predicting which destination a new user will go to, is basically a classification problem.[13] We have implemented several classifiers, the details of which are explained below. For each of the classifiers, we followed the following methodology:

(1) Run the classifier on dataset split for training and testing.
(2) Tune the parameters for each classifier, and find which combination gives the best result.

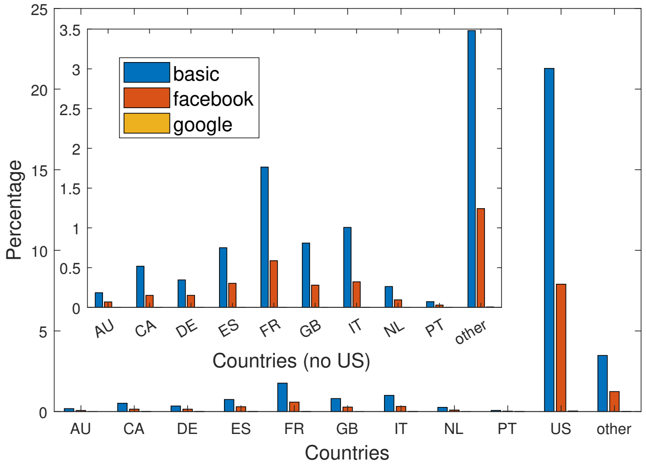


Fig. 9: Destination Country distribution VS. user's sign-up method.

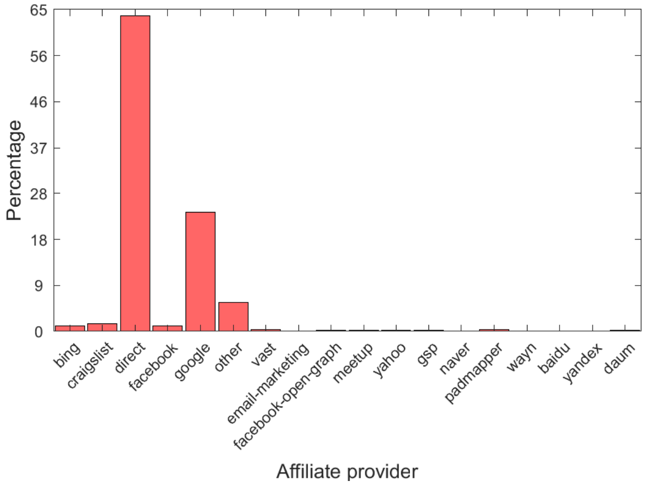
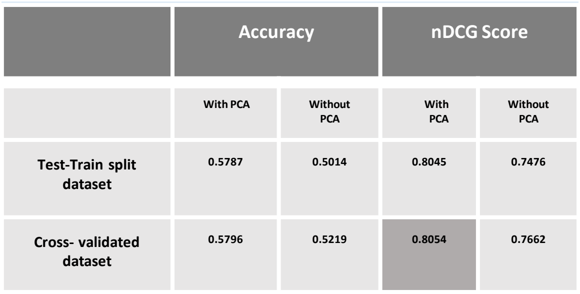


Fig. 10: Destination Country distribution VS. user's sign-up method.

(3) Use cross-validated dataset to evaluate classifier's performance.
(4) Evaluate the performance using the parameters found in the above steps on PCA-applied dataset.
(5) We evaluate accuracy and NDCG score for each of the cases.

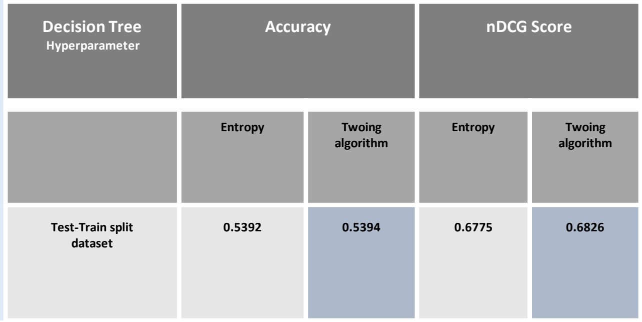
5.1 Naive Bayes Classifier

Our first classifier is Naive Bayes[9] which provided the worst results (in terms of accuracy) out of all the classifiers used. The results are summarized in the Fig. 11: From the results, we can see that PCA-applied cross-validated dataset gives the best NDCG score which is our evaluation metric. This combination provides the best accuracy as well. We repeat the same experiment for different classifiers,[11] each time finding the best tuned hyperparameter as shown in Fig.11.



	Accuracy		nDCG Score	
	With PCA	Without PCA	With PCA	Without PCA
Test-Train split dataset	0.5787	0.5014	0.8045	0.7476
Cross- validated dataset	0.5796	0.5219	0.8054	0.7662

Fig. 11: Results of Naive Bayes classifier summarized



Decision Tree Hyperparameter	Accuracy		nDCG Score	
	Entropy	Twoing algorithm	Entropy	Twoing algorithm
Test-Train split dataset	0.5392	0.5394	0.6775	0.6826

Fig. 12: Evaluation of split parameter

5.2 Decision Trees

[5] The test-dataset that was provided in the competition had the attribute **date_account_created** from 7/1/2014 which is unrelated to the date in the training dataset. Thus, in decision tree, we remove this attribute and compare the result with the initial dataset. Fig.17 summarizes the result of using the modified dataset vs the initial dataset. We select the combination which provides the highest accuracy and NDCG score for our model hyperparameters.
The hyperparameters evaluated in the Decision Tree model is listed below.

5.2.1 *Split Parameter.* : The default split parameter[8] in MATLAB for Decision Trees is Gini index. We try 2 different criterion for splitting in this section ie. entropy and twoing algorithm. As seen from the Fig.17 and Fig.12, the best result is obtained using twoing algorithm as the split criterion. We use this parameter in further evaluating the rest of the hyperparameters.

5.2.2 *Maximum number of splits.* : Our next hyperparameter is the maximum number of decision splits.[6] We evaluated our result varying the split upto 30. We see from Fig.13 and Fig.14 that the best result is obtained when the split is 22. We use this result to evaluate the next hyperparameter.

5.2.3 *Minimum number of leaf nodes.* : The final hyperparameter used for modeling the Decision Tree is the minimum number of leaf nodes. As seen from Fig.15 and Fig.16, varying the minimum number of leaf nodes does not change the results of our model much. Hence, we do not include it in our final model building.

As we see from the results in 17, our Decision Tree model performs slightly better after tuning the hyperparameters. Next, we construct a final Decision Tree model here employing the combination of the hyperparameters which gave the best result in the

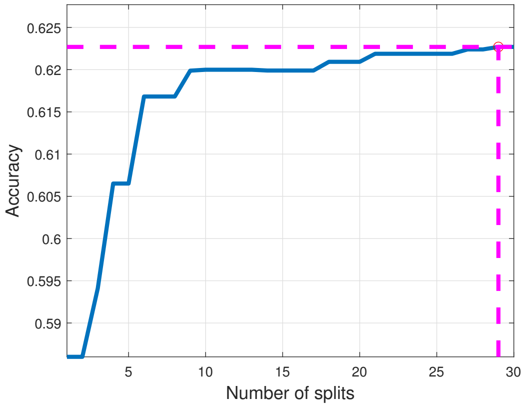


Fig. 13: Accuracy varying the maximum number of splits

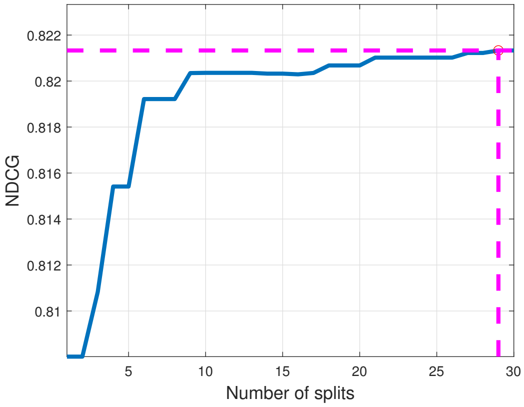


Fig. 14: NDCG score varying the maximum number of splits

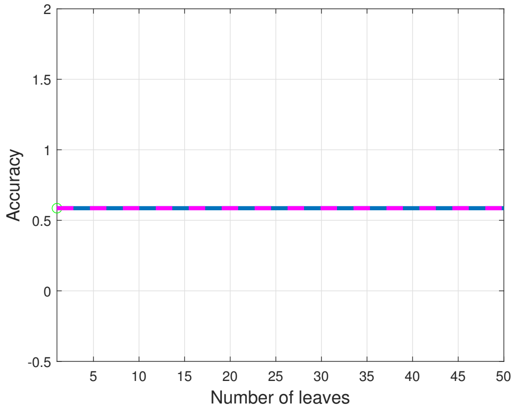


Fig. 15: Accuracy varying the minimum leaf nodes number.

above steps. Our final model gives the results summarized in Table. 1.

Final Decision Tree Model	
PCA applied	Yes
Cross-validated dataset	No
Split Criterion	Twoing algorithm
Maximum number of decision splits	22
Accuracy	**0.6220**
nDCG Score	**0.8208**

Table 1: Final Decision Tree Hyper-parameters and Result

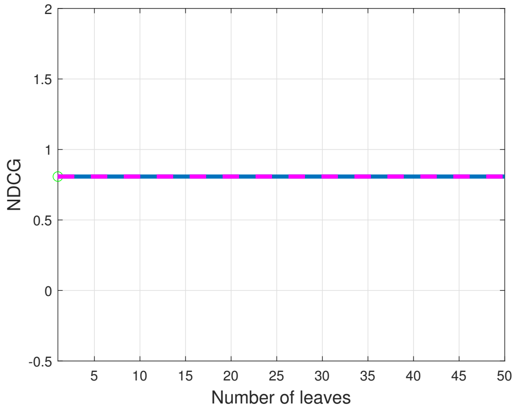
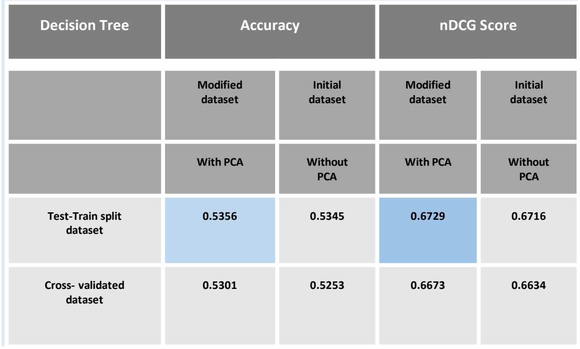


Fig. 16: nDCG score varying the minimum leaf nodes number.



Decision Tree	Accuracy		nDCG Score	
	Modified dataset	Initial dataset	Modified dataset	Initial dataset
	With PCA	Without PCA	With PCA	Without PCA
Test-Train split dataset	0.5356	0.5345	0.6729	0.6716
Cross- validated dataset	0.5301	0.5253	0.6673	0.6634

Fig. 17: Results of Decision Tree summarized

5.3 K Nearest Neighbor algorithm

kNN classification algorithm[1] is our next choice to model our data. The hyperparameters evaluated in the kNN algorithm are listed below.

5.3.1 Value of k. To pick an optimal k, we initially experimented with k ranging 1 to 51 as shown in Fig. 18.

k	Time	Accuracy	NDCG
100	162.8706	0.5849	0.8021
100000	498.9192	0.5844	0.8071

Table 2: kNN model k tuning.

Distance formula	Time	Accuracy	NDCG
Euclidean	0.877569	0.5830	0.8010
Cosine	29.883017	0.5829	0.7918

Table 3: kNN model distance criteria selection

Final kNN Model	
PCA applied	Yes
Cross-validated dataset	Yes
No. of k	100
Distance formula	Euclidean
Accuracy	**0.5839**
nDCG Score	**0.8022**

Table 4: kNN final model hyperparameters and result

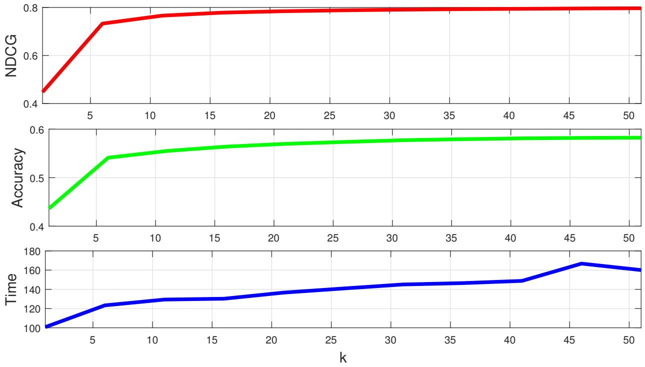
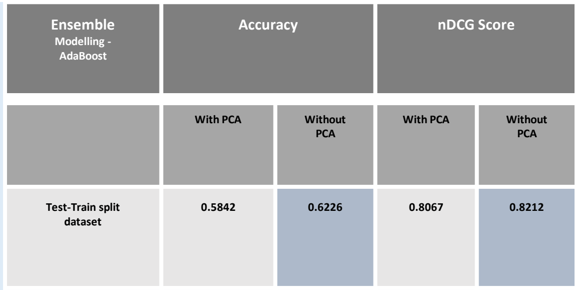


Fig. 18: Evaluating k=1 to 51

It can be seen from the graph above that although the time complexity increases, it tends to decrease after k=46. It can be a good trade-off considering that both accuracy and nDCG score increases with k.[14] We now try to see our result on a larger k ie. 100 and 10000 which is shown in Table. 2. From this table we see that k=100 will be an optimum choice with a good balance for time complexity as well as NDCG and accuracy.

5.3.2 Distance Formula. MATLAB's default distance formula is Euclidean. We try **cosine** distance formula here. We test our model with k=100 and with both Euclidean and Cosine as the distance metric on PCA-applied data which is summarized in Table. 3.We can see that cosine has a higher time complexity while having the same accuracy.

Considering the time complexity above, Euclidean distance metric comes out as the optimal choice. We construct a final kNN model[7] here employing the combination of the hyperparameters which gave the best result in the above steps. Our final model is shown in Table.4.



Ensemble Modelling - AdaBoost	Accuracy		nDCG Score	
	With PCA	Without PCA	With PCA	Without PCA
Test-Train split dataset	0.5842	0.6226	0.8067	0.8212

Fig. 19: Effect of PCA on the Ensemble Model

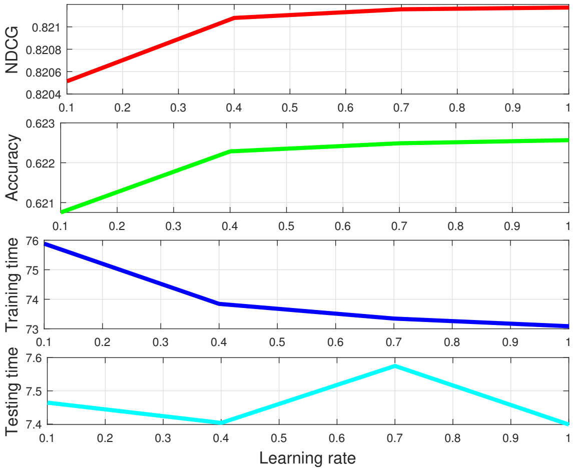


Fig. 20: Effect of learning rate on the ensemble model

Final Ensemble Model - AdaBoost	
PCA applied	No
Cross-validated dataset	Yes
Learning Rate	1
Accuracy	**0.6224**
nDCG Score	**0.8211**

Table 5: Ensemble final model hyperparameters and result

5.4 Ensemble Modeling

Our final model is an ensemble modeling[2] using **AdaBoost**. We train our model based on PCA-applied test-train split dataset as shown in Fig. 19. From the Fig.19, we see that not applying PCA on the dataset gives a better result. The hyperparameter in this case is the **Learning Rate**. We train our model for values from 0.1 to 1 as shown in Fig. 20. From Fig.20, we can see that the best result is obtained by the Learning Rate = 1. We construct a final Ensemble model here employing the combination of the hyperparameters which gave the best result in the above steps. Our final model is summarized in Table. 5

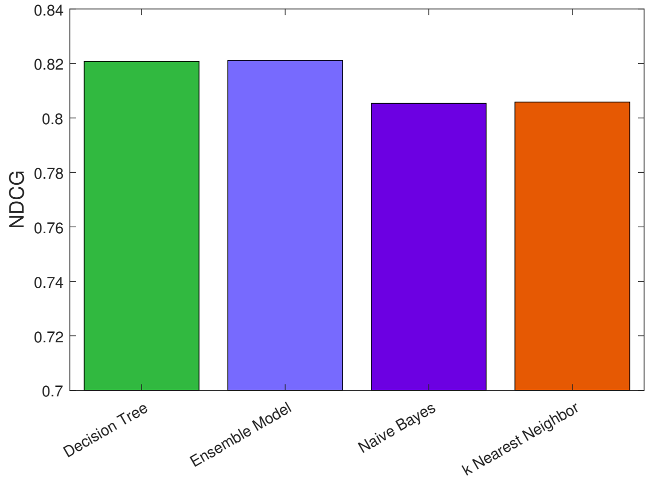


Fig. 21: NDCG score comparison between the used models

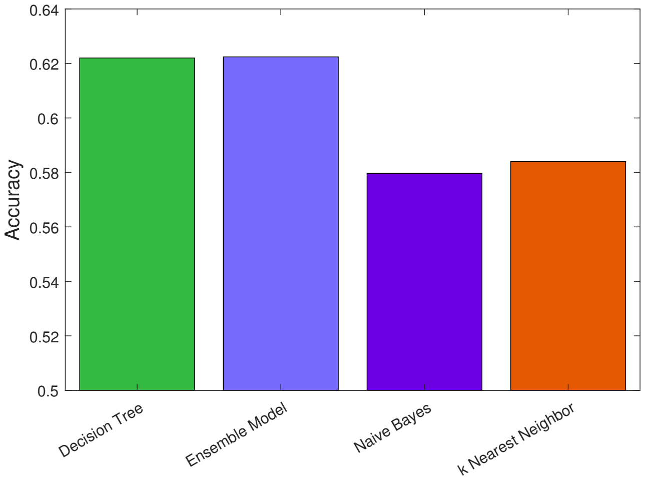


Fig. 22: Accuracy comparison between the used models

6 MODEL EVALUATION

The evaluation metric is proposed by Airbnb which is *NDCG* (Normalized Discounted Cumulative Gain) @k where, k=5.

$$\mathbf{DCG_k} = \sum_{i=1}^{k} \frac{2^{\mathbf{rel_i}} - 1}{\log_2(i+1)}$$

$$\mathbf{nDCG_k} = \frac{\mathbf{DCG_k}}{\mathbf{IDCG_k}}$$

where $\mathbf{rel_i}$ is the relevance of the result at position $\mathbf{i}$[10].
$\mathbf{IDCG_k}$ is the maximum possible (ideal) **DCG** for a given set of queries. All NDCG calculations are relative values on the interval 0.0 to 1.0.

For each new user, we are needed to make a maximum of 5 predictions on the country of the first booking. The ground truth country is marked with relevance = 1, while the rest have relevance = 0.
For example, if for a particular user the destination is FR, then the predictions become:

[FR] gives a **NDCG** $= \frac{2^1 - 1}{\log_2(1+1)} = 1.0$

[US, FR] gives a **DCG** $= \frac{2^0 - 1}{\log_2(1+1)} + \frac{2^1 - 1}{\log_2(2+1)} = \frac{1}{1.58496} = 1.0$

NDCG scores of our analyzed models are shown in Fig. 21. We can see from this figure that, the decision tree and ensemble models have very close NDCG scores while the NDCG score of kNN and Naive Bayes models are comparably low. Moreover, in Fig. 22 we compare the accuracy of the chosen models with each other. The accuracy values also follow the same order as NDCG. However, in general they are not very high. This is why in the competition focus was on NDCG score which gives a better model evaluation.

Fig. 23 shows ROC curves of the models. Since we have a multi-class classification problem, thus, we consider a one-VS-all approach[4] to find the true positive and false positive values by applying the ROC procedure on the 'US' class. We can see from Fig. 23 that while Naive Bayes and kNN classifiers have very bad performances, decision tree and ensemble method[12] have a close and acceptable performance.

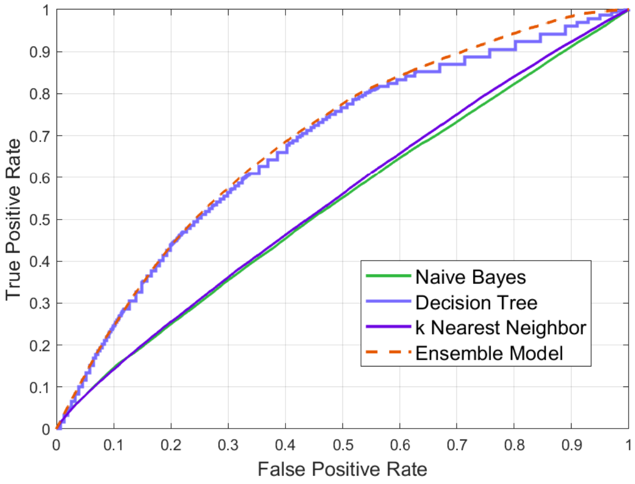


Fig. 23: ROC curve comparison between the used models

Finally, to analyze the complexity of the models, in Fig. 24, we show the training and testing time duration for all the chose models. The ensemble model takes comparably longer time to converge. Even the testing delay is very longer than the other models. Thus, since the performance of the ensemble method and the decision tree are very close, to obtain a better generalization accuracy which takes into account also the complexity of the model, one should choose the decision trees for this problem.

7 REAL-WORLD INSIGHTS

This project served as our first hands-on experience working with large real datasets. The dataset contained many attributes and figuring out which attributes should be used in building our model

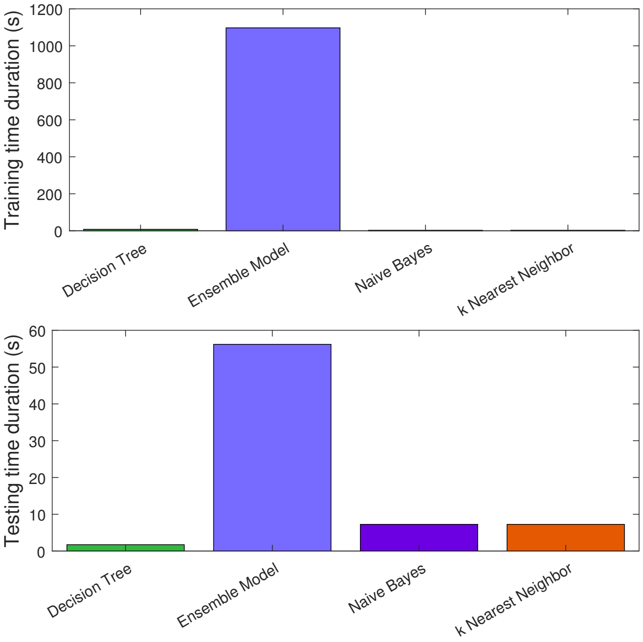


Fig. 24: Complexity comparison between the used models

was be a challenging task. Exploring the data helped us understand the correlation between different attributes and their distribution. Out of this project, we have gathered the direction and motivation to start working on similar projects. Specifically, we chose this subject since the idea and solution can fit in many other applications such as flight destination prediction and choosing hotel for the new users of online hotel finder websites such as Trivago or online shopping websites in which providing location specific options for the users might encourage them to buy the products. In this respect, we find this project very suitable for the goals of this course since, first, we have learned how to apply the course material on a real-world problem and, second, the solution of this project will help us in dealing with many of the future data analytics projects.

In summary, we have applied four classification models to the provided dataset from Airbnb. We have done a number of preprocessing techniques such as PCA, one hot encoding and distribution matching. We have removed some unnecessary or missing attributes from our model to achieve higher accuracy. Then, using bar plot, box plot, histogram and line plots we have visualized the data. The goal of this project is to achieve a higher NDCG score which basically shows the relevance of a models output to a query. Thus, by tuning the hyperparamters of our models we maximized the NDCG score. Our final results show that, the decision tree and ensemble models have better performance compared to Naive Bayes and kNN models. Moreover, due to lower complexity of the decision tree model, we choose the decision tree as our final model in which we achieve an NDCG score of 0.8208.

[The project can be found at: www.github.com/AidinFerdowsi/Airbnb-new-user-booking]

REFERENCES

[1] 2010. A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm. (2010). https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/

[2] 2010//. *Ensemble Methods in Data Mining Improving Accuracy Through Combining Predictions.* San Rafael, CA, USA. 108 – pages. data mining;influential development;machine learning;investment timing;drug discovery;fraud detection;recommendation systems;predictive accuracy;model interpretability;decision trees;ensembling algorithms;importance sampling;rule ensemble methods;decision tree;linear rule models;fault diagnosis;industry experts;leading academics;.

[3] 2016. Airbnb New User Bookings. (2016). https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings

[4] 2016. Airbnb New User Bookings. (2016). https://github.com/nikhiljangam/Airbnb-new-user-bookings

[5] W.M. Al-Hoqani. 2017//. Difficulties of marking decision tree diagrams. *2017 Computing Conference*, 1190 – 4. http://dx.doi.org/10.1109/SAI.2017.8252241 decision tree diagrams;marking process;decision tree assessment;data mining methods;r inductive inference;educator assessors;.

[6] Ying Cao and Chunhai Zhang. 2012. Algorithm with weighted attributes for unresolved exception in decision tree induction algorithm. *Proceedings of the 2012 2nd International Conference on Business Computing and Global Informatization, BCGIN 2012*, 515 – 517. http://dx.doi.org/10.1109/BCGIN.2012.140 accuracy;Classification methods;Data sets;Decision tree classification;Decision tree induction;Decision-tree algorithm;exception;Information gain;Majority voting algorithm;Weighted attributes;.

[7] Gongde Guo, Hui Wang, D. Bell, Yaxin Bi, and K. Greer. 2003//. KNN model-based approach in classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2003. Proceedings. (Lecture Notes in Comput. Sci. Vol. 2888)*, 986 – 96. KNN;classification;k-nearest-neighbours;lazy learning;dynamic Web mining;model construction;public dataset;UCI machine learning repository;.

[8] Bi Jiandong and Yang Guifang. 1997/04/. Algorithm for merging of branches in decision tree induction. *Journal of the Harbin Institute of Technology* 29, 2 (1997/04/), 44 – 6. decision tree induction;merging of branches;learning by example;splitting algorithm;agglomerative algorithm;ID<sub>3</sub>;decision tree;EMID;.

[9] Liangxiao Jiang and H. Zhang. 2006//. Learning naive Bayes for probability estimation by feature selection. *Advances in Artificial Intelligence. 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006. Proceedings (Lecture Notes in Artificial Intelligence Vol. 4013)*, 503 – 14. http://dx.doi.org/10.1007/11766247_43 learning naive Bayes;probability estimation;feature selection;classification algorithm;conditional log likelihood;search process;classification accuracy;selective Bayesian classifier;attribute selection;.

[10] Ping Li, Christopher J. C. Burges, and Qiang Wu. 2007. McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07).* Curran Associates Inc., USA, 897–904. http://dl.acm.org/citation.cfm?id=2981562.2981675

[11] Bai Li-yuan, Huang Hui, Liu Su-hua, and Yan Qiu-ling. 2007/08/. Naive Bayes classifier based on bootstrap average. *Computer Engineering* 33, 15 (2007/08/), 190 – 2. naive Bayes classifier;bootstrap average;classification accuracy;distribution estimation;naive Bayes text classifier;word clusters;bootstrap method;parameter estimation;text dataset;.

[12] K.-M. Osei-Bryson. 2010/02/01. Towards supporting expert evaluation of clustering results using a data mining process model. *Information Sciences* 180, 3 (2010/02/01), 414 – 31. http://dx.doi.org/10.1016/j.ins.2009.09.019 supporting expert evaluation;clustering results;data mining process model;.

[13] Zhenxing Qin, Chengqi Zhang, Tao Wang, and Shichao Zhang. 2010//. Cost sensitive classification in data mining. *Advanced Data Mining and Applications. Proceedings 6th International Conference (ADMA 2010)* pt.1, 1 – 11. http://dx.doi.org/10.1007/978-3-642-17316-5_1 semilearning strategy;lazy-learning;test cost;misclassification cost;cost-sensitive learning;diagnosis data analysis;unbalance class distribution;machine learning;data mining;cost sensitive classification;.

[14] H. Yigit. 2013//. A weighting approach for KNN classifier. *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, 228 – 31. http://dx.doi.org/10.1109/ICECCO.2013.6718270 KNN classifier;weighting approach;k nearest neighbors algorithm;artificial bee colony algorithm;ABC based distance-weighted kNN;dW-ABC kNN;UCI data sets;iris data set;Haberman data set;breast cancer data set;performance degradation;zoo data set;.

2nd May, 2018

CS 5525 - Data Analytics

Final Project - **Airbnb New User Bookings**

By - Aidin Ferdowsi & Anamika Sen

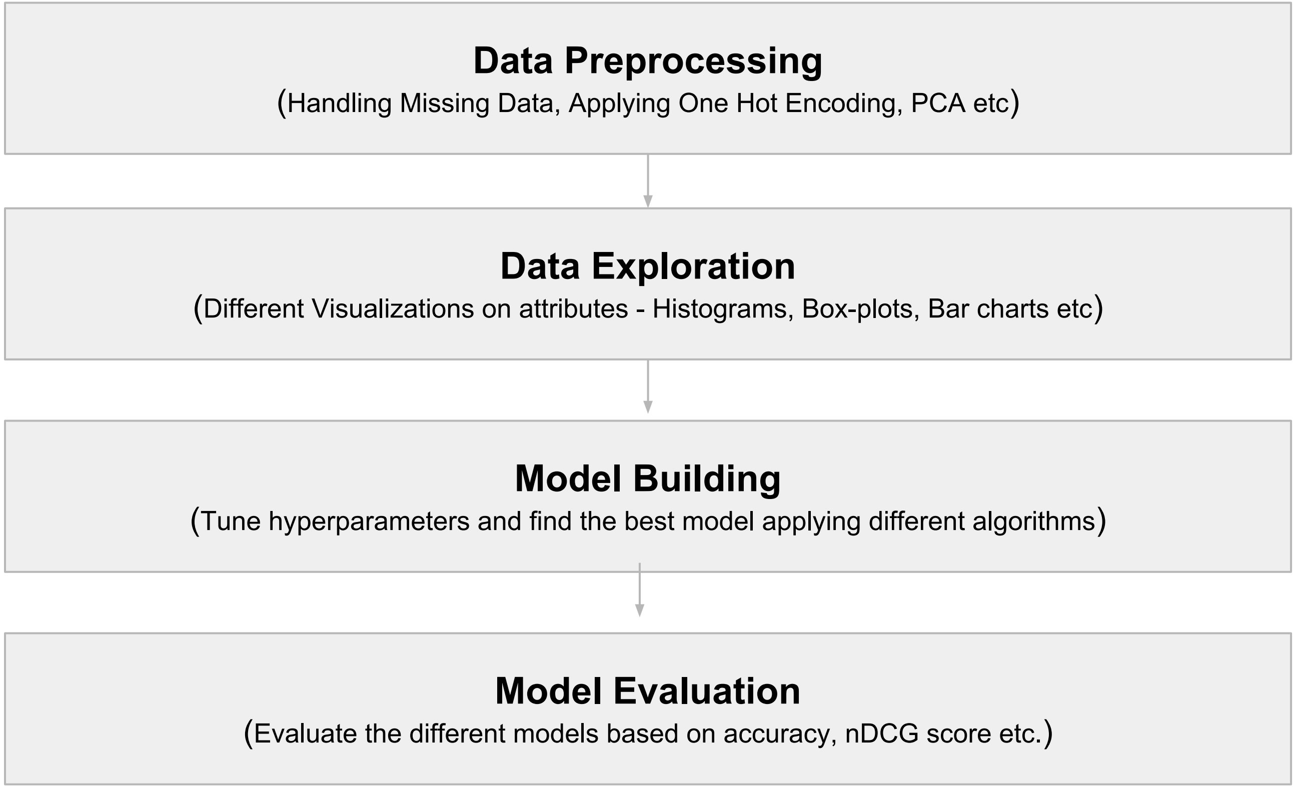
Problem Statement

- Predict the country where a new user make his/ her booking.
- Dataset consists of list of users along with their demographics, web session records, and some summary statistics.
- All the users are from US.
- Predictor variables are a total of 12 destination including NDF (No Destination Found)

	Size
Training Dataset	171239
Test Dataset	43673
Attributes	16

2nd May, 2018

Problem Approach



Data Preprocessing
(Handling Missing Data, Applying One Hot Encoding, PCA etc)

Data Exploration
(Different Visualizations on attributes - Histograms, Box-plots, Bar charts etc)

Model Building
(Tune hyperparameters and find the best model applying different algorithms)

Model Evaluation
(Evaluate the different models based on accuracy, nDCG score etc.)

2nd May, 2018

Data Preprocessing

- Handling Missing Data

  gender : Replaced -unknown- values in the same distribution for Male, Female and others as given in the dataset.
  date_first_booking: Discard this attribute from the training dataset since it is not present in the test dataset.
  age: Unacceptable ages such as higher than 95 and lower than 16 are replaced with NaN values.
  first_affiliate_tracked: Replace NaN values with untracked

- One Hot Encoding

  Applied one-hot encoding on the nominal features

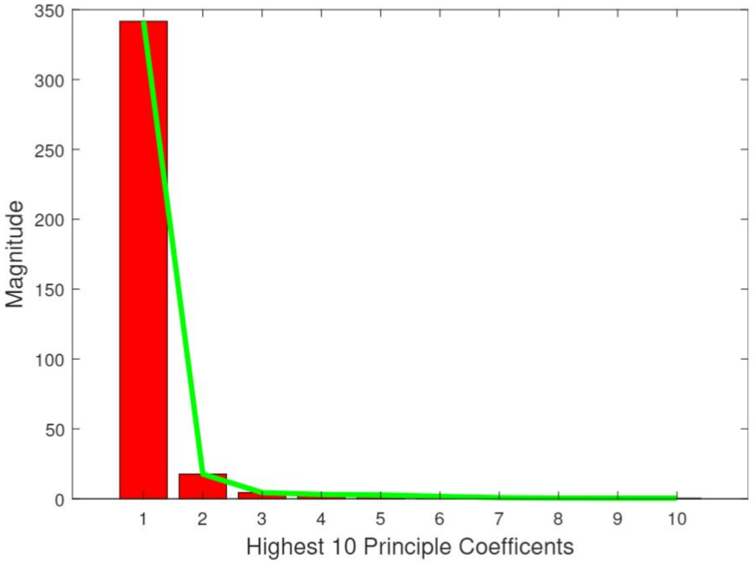
- Time Conversion

  Separate time attribute into day, month and year. Since month can give us more information than year.

2nd May, 2018

Data Preprocessing

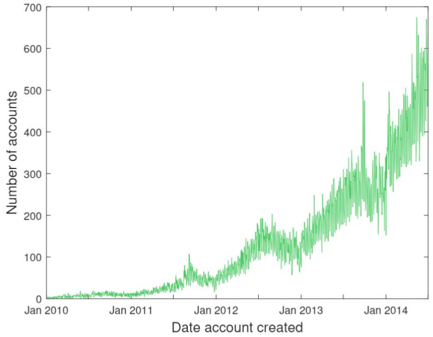
- Principal Component Analysis (PCA)
  - Applied PCA to one-hot encoded data.
  - The figure shows the scree plot of the first 10 highest Eigenvalues.
  - Choose the first Eigenvalue for the training data because it has the highest power.



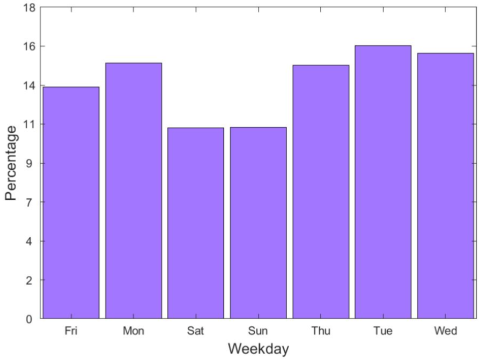
Data Exploration

Some of the interesting visualizations are listed here:

Date of account creation:



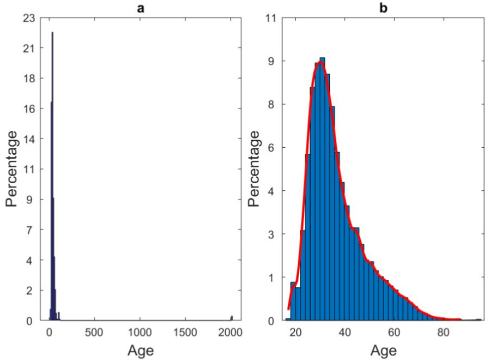
- The adjoining figure of **date_account_created** shows that Airbnb's reputation increased during a particular time, which led to a higher number of accounts created.



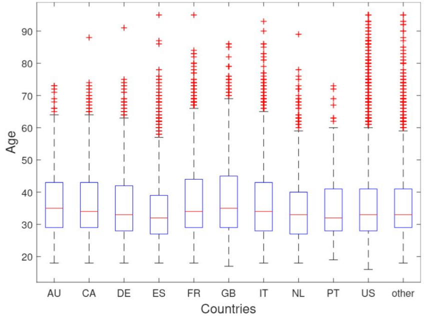
- After extracting the weekday from date_account_created, we see that users are **less active on weekends**, while most of the users created their accounts on Tuesday.

Data Exploration

Age attribute:



- The first histogram plot shows that half the data are missing with the inclusion of some illogical inputs. Like age less than 18 or very high.
- Limiting the age of users between 16 and 95 shows that users between the **age of 25-40 tend to use Airbnb more** (from the second histogram)



- The boxplot in the adjoining figure can be used to understand the age distribution of users.
- It can be seen that users booking for countries like **Spain, Portugal and Netherlands tend to be younger population** while the older population book countries like Great Britain and France.

Data Exploration

Destination country v/s user sign-up method and app:

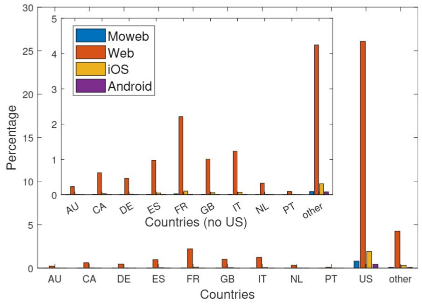


Fig. 9: Destination Country distribution VS. user's sign-up app.

- Graph shows the country distribution and the user's signup app.
- Most users signup using Web.
- **iOS is the second most popular** signup app.

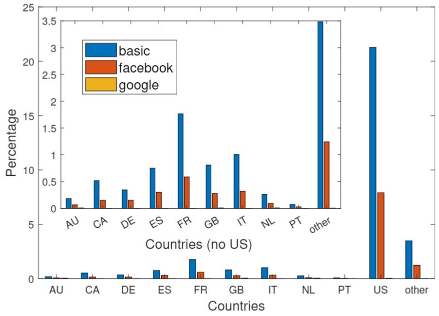


Fig. 10: Destination Country distribution VS. user's sign-up method.

- Graph shows the country distribution and the user's signup method.
- Most users signup directly using Airbnb application.
- **Second most** applied signup method is **Facebook.**

2nd May, 2018

Model Building

Predicting which country a user will go to, is basically a classification problem. The classifiers used in this project are as follows:

Naive Bayes Classifier:

- Classifier is used on both test-train split and cross-validated data.

	Accuracy		nDCG Score	
	With PCA	Without PCA	With PCA	Without PCA
Test-train split dataset	0.5787	0.5014	0.8045	0.7476
Cross-validated dataset	**0.5796**	0.5219	**0.8054**	0.7662

Decision Tree:

- Found best combination of hyperparameters for the best working Decision Tree.
- Hyperparameters tuned:
  Split criterion: Twoing Algorithm
  Maximum number of decision splits: 22
  Minimum number of leaf nodes: 1

Accuracy	0.6220
nDCG Score	0.8208

Model Building

k Nearest Neighbour:

- Evaluated time complexity, accuracy and nDCG for different values of k.
- Optimal k was found to be **k=100**. Distance formula used is **Euclidean.**

Accuracy	0.5839
nDCG Score	0.8022

Ensemble Modeling - AdaBoost:

- Not applying PCA gave better results.
- Varied and evaluated Learning Rate from 0.1 to 1.
- Learning Rate=1 gave the optimum result considering accuracy, nDCG score and Time Complexity.

Accuracy	0.6224
nDCG Score	0.8211

Model Evaluation

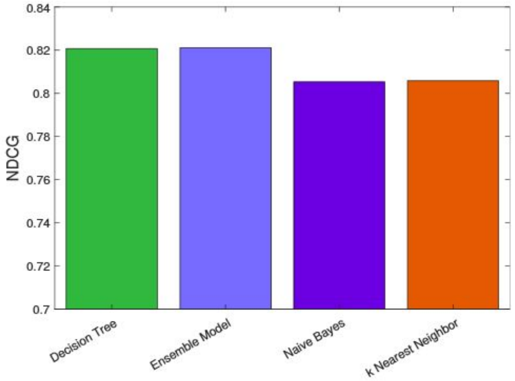


Fig. 21: NDCG score comparison between the used models

- The **NDCG** graph shows that **Ensemble Modeling** gives the best result.
- Decision Trees perform the second-best, by a slight difference.

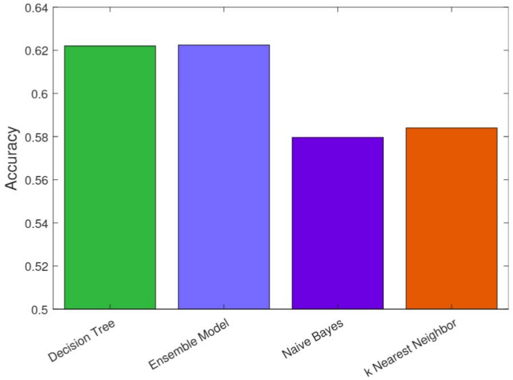


Fig. 23: Accuracy comparison between the used models

- Also, the adjoining **Accuracy** graph shows that **Ensemble Modeling** gives the best result.
- Decision Trees perform the second-best, by a slight difference.

2nd May, 2018

Model Evaluation

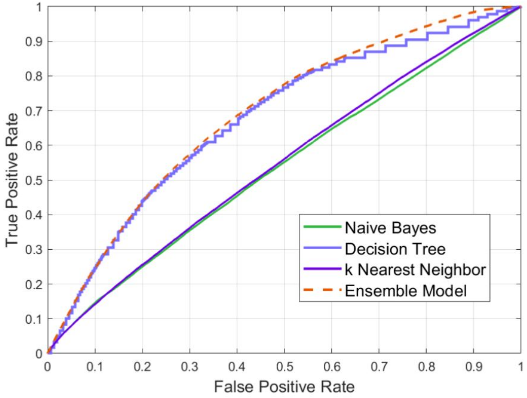


Fig. 24: ROC curve comparison between the used models

ROC curve

- Used **One v/s All approach** for the multiclass classification problem to construct ROC.
- Naive Bayes and kNN have very bad performance.
- Decision Trees and Ensemble Modeling have a close and acceptable performance.

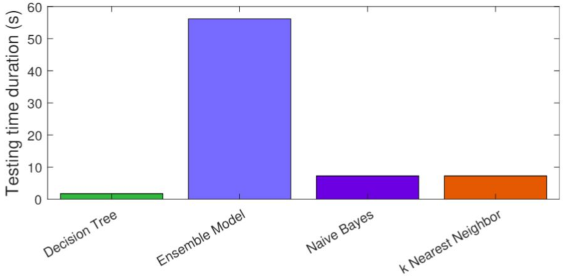


Fig. 25: Complexity comparison between the used models

- The adjoining **Time complexity** graph shows that Ensemble Modeling take a lot of time while **Decision Trees** perform the best.
- Considering the trade-off between model complexity and accuracy, **Decision Tree would is the best model in this case**.