Predicting the Travel Destination of New User Bookings

CS 6375.002 - Machine Learning - S17

Project Report

04.30.2017

Ashwath Santhanam axs161730
Haripriyaa U Manian hum160030
Preethi Sekar pxs163930

TABLE OF CONTENTS

**click on a heading to go directly to that section*

INTRODUCTION

Using the Online Reservation System in Airbnb, users book the accommodation for their desired travel location. In these systems, reservations of users of different age group, gender would be done in various countries. New users can book a place to stay in 34000+ cities across 190+ countries. Using the data of the booking methodology of various in the past, We present a system using the Predictive Modelling techniques of Machine Learning, where we will accurately predict which country a new user's first booking destination will be. We first did some comprehensive analysis on the dataset, explored most features and collected all features we thought were useful. We then described and interpreted the prediction task and the evaluation method.

MOTIVATION

Data scientists at Airbnb collect and use data to optimize products, identify problem areas, and inform business decisions. For most guests, however, the defining moments of the Airbnb experience happen in the real world - exploring the destination. These are the moments that make or break the Airbnb experience, no matter how great the website is. The purpose of this project is to show how we can use Airbnb's data to understand the process of booking experience, and in particular how the Predictive Modelling study adds a huge value in increasing the user base. By accurately predicting where a new user will book their first travel trip, we can share a personalized content with their community, decrease the average time for booking, and better forecast demand, thereby giving the new user a better and smooth booking experience overall.

PROBLEM STATEMENT

Large amount of reservation data in Airbnb can be interpreted to acquire knowledge about tasks that will occur in the environment. Patterns in these data can be used to predict the future events. Knowledge about these tasks facilitates the automation of task components to improve the inhabitant's experience.

We collect the data from the airbnb data set that contains detailed information about the list of users and the different factors that led them to book their first travel destination, which we describe in detail in the next section. We apply some of the Machine Learning algorithms like Decision Trees, Neural Networks, Naive Bayes Classifier, SVM etc. to this dataset to predict the travel destination of the new user.

DATASET

In the Airbnb data set, we are given a list of users along with their demographics, web session records, and some summary statistics. We will predict which country a new user's first booking destination will be. All the users in this dataset are from the USA.

Details of the actual dataset:

train_users.csv - the training set of users
test_users.csv - the test set of users

id	user id
date_account_created	the date of account creation
timestamp_first_active	date_first_booking because a user can search before signing up
date_first_booking	date of first booking
gender	User's gender
age	User's age
signup_method	User's method of signup
signup_flow	the page a user came to signup up from
language	international language preference
affiliate_channel	what kind of paid marketing
affiliate_provider	where the marketing is e.g. google, craigslist, other
first_affiliate_tracked	what's the first marketing the user interacted with before the signing up
signup_app	The app they used to sign up

first_device_type	The type of device
browser_type	Type of the browser
country_destination	**this is the target variable to be predicted**

Country_destination - target variable:

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'. Please note that 'NDF' is different from 'other' because 'other' means there was a booking, but is to a country not included in the list, while 'NDF' means there wasn't a booking.

sessions.csv - web sessions log for users
user_id: to be joined with the column 'id' in users table
action
action_type
action_detail
device_type
secs_elapsed

countries.csv - summary statistics of destination countries in this dataset and their locations

age_gender_bkts.csv - summary statistics of users' age group, gender, country of destination

Based on the combination of the files a consolidated file **(airbnb_dataset.csv)** was generated from the list of files present. In that file around 13000 instances and 9 attributes were chosen.

ALGORITHMS USED

Following are some of the Machine Learning algorithms and their definitions. We have implemented these algorithms which help in computing the target variable and its prediction:

Decision Trees

DTs use training instances to build a sequence of evaluations which can be used to permit the correct category (prediction). This algorithm hence can be used to identify the countries from which prominent number of users would be booking tickets upfront. Best attribute that can be used to split the attribute set is done based on information gain, which can be calculated based on Shannon's entropy.

Neural Networks

Artificial neural networks are relatively crude electronic networks of neurons based on the neural structure of the brain. They process records one at a time, and learn by comparing their classification of the record with the known actual classification of the record. The errors from the initial classification of the first record is fed back into the network, and used to modify the networks algorithm for further iterations.

Naïve Bayes Classifier

We use Bayes probabilities to determine the most likely next event for the given instance for all the training data. Conditional probabilities are determined from the training data. Based on those values, classification would be done.

Support Vector Machines

Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Bagging

It is a method which generates multiple versions of predictor by bootstrap samples and using them to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class.

Boosting

The weight of all training samples would assigned equally. Then training on the model is done. Based on the error calculated in the iteration, we would increase the weights of incorrectly classified data. This process would be repeated until accurate prediction of weights is done for the model. Boosting are of two types: Ada Boosting and Gradient Boosting. Both algorithms are implemented in our system.

k-NN algorithm was not used for implementation as the dataset had lot of non-numerical values. Hence finding a nearest neighbor is complicated.

EXPERIMENTAL METHODOLOGY

Predictive Modelling:

It encompasses a variety of techniques from machine learning that analyze current and historical facts to make predictions about future or otherwise unknown events.

We use Implicit data collection procedure where we observe the various factors that led the user to decide upon the destination and then make a prediction for future users based on their selection preferences.

In this project, we have two implementation files.

i. Classifiers_by_Cross_Validation.R

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. We are performing a 3-fold cross validation (k=3).

We first take the Training Data from airbnb and pre-process it. We are making sure that the attribute(country_destination) to be predicted is a factor type.

We then find the Accuracy and Kappa Statistic for all the major Machine Learning Algorithms

Accuracy:

The Accuracy factor is defined as, Overall, how often is the classifier correct.

Kappa statistic:

This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.

ANALYSIS OF RESULTS

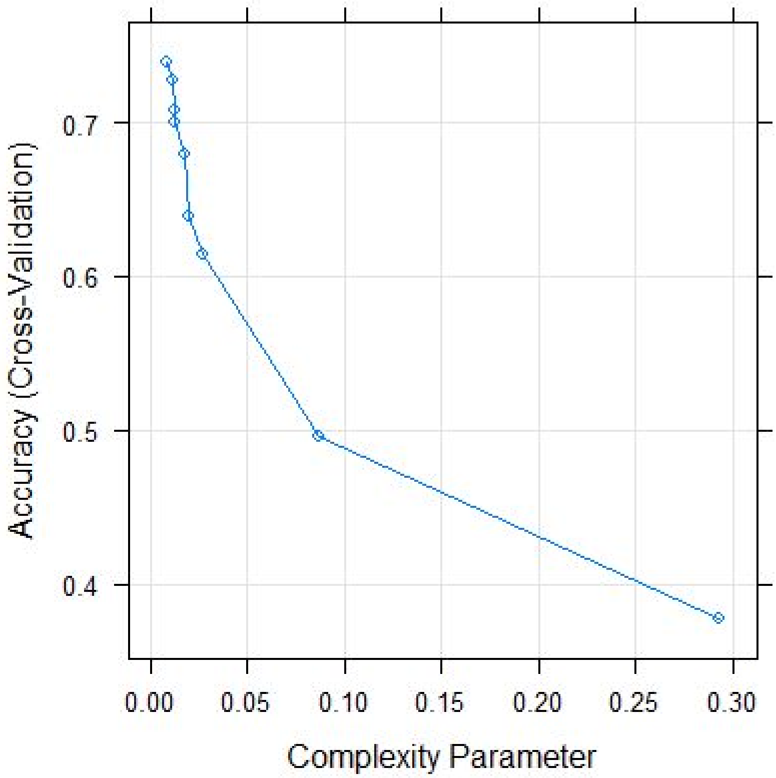
Decision Tree

```
> train_dtree
```

Output

```
CART

12960 samples
    8 predictor
   10 classes: 'AU', 'CA', 'DE', 'ES', 'FR', 'GB', 'IT', 'NL', 'PT', 'US'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 8638, 8641, 8641
Resampling results across tuning parameters:

  cp           Accuracy   Kappa
  0.009177973  0.7394289  0.6811587
  0.012270551  0.7270032  0.6665700
  0.012619713  0.7071751  0.6387743
  0.012869114  0.7004652  0.6266645
  0.017857143  0.6788553  0.5964350
  0.020550678  0.6395078  0.5430828
  0.027633679  0.6148108  0.5071912
  0.086492418  0.4959205  0.3485743
  0.292298484  0.3772202  0.1948880

Accuracy was used to select the optimal model using  the largest value.
```

The final value used for the model was cp = 0.009177973.



Neural Networks

> train_nnet

Output

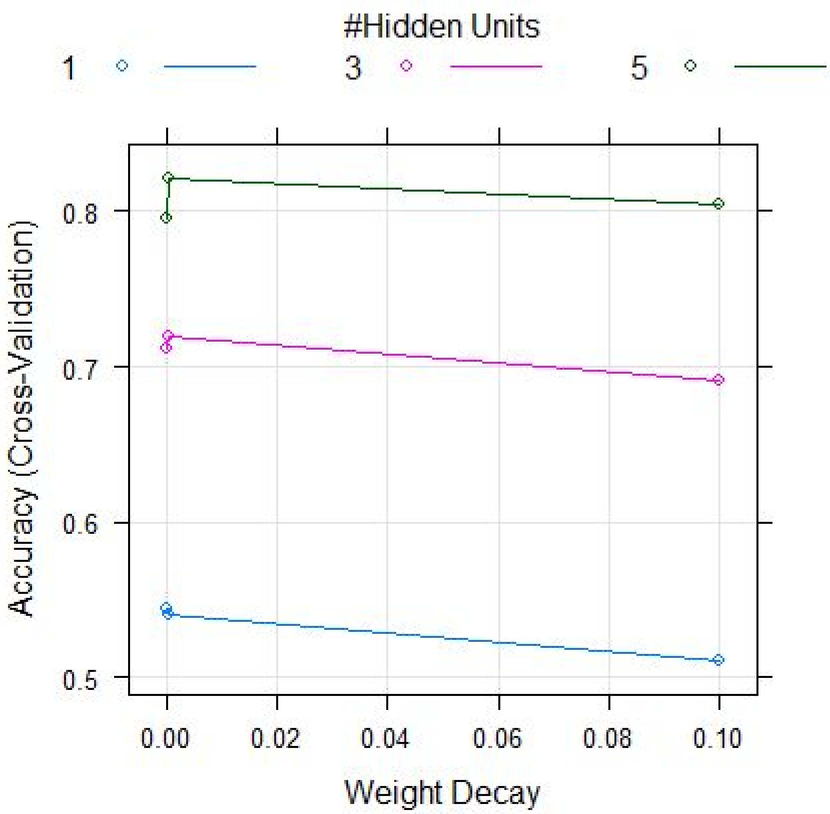
Neural Network

12960 samples
 8 predictor
 10 classes: 'AU', 'CA', 'DE', 'ES', 'FR', 'GB', 'IT', 'NL', 'PT', 'US'

Pre-processing: centered (19), scaled (19)
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 8642, 8640, 8638
Resampling results across tuning parameters:

size	decay	Accuracy	Kappa
1	0e+00	0.5442955	0.4141787
1	1e-04	0.5404539	0.4102498
1	1e-01	0.5108841	0.3744005
3	0e+00	0.7121937	0.6419082
3	1e-04	0.7193039	0.6521025
3	1e-01	0.6905799	0.6129390
5	0e+00	0.7954373	0.7490833
5	1e-04	0.8213770	0.7822056
5	1e-01	0.8047003	0.7631422

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were size = 5 and decay = 1e-04.

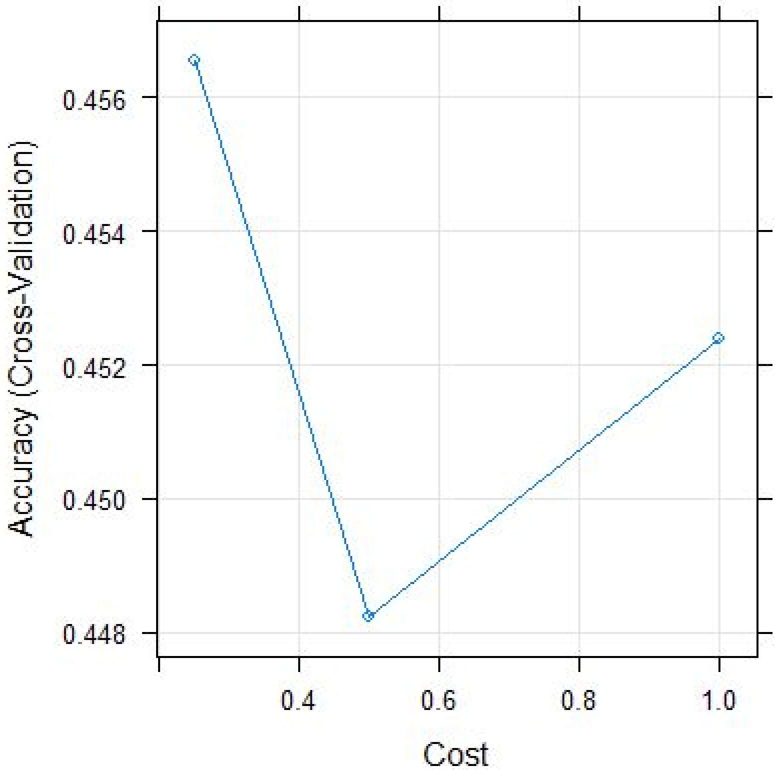


Support Vector Machines(svm)

```
> train_svm
```

Output

```
Support Vector Machines with Radial Basis Function Kernel

12960 samples
    8 predictor
   10 classes: 'AU', 'CA', 'DE', 'ES', 'FR', 'GB', 'IT', 'NL', 'PT', 'US'

Pre-processing: centered (19), scaled (19)
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 8640, 8641, 8639
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.4565594  0.3843945
  0.50  0.4482257  0.3738164
  1.00  0.4523930  0.3752967

Tuning parameter 'sigma' was held constant at a value of 0.05
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.05 and C = 0.25.
```



Bagging

> train_bag

Output

Bagged CART

12960 samples
 8 predictor
 10 classes: 'AU', 'CA', 'DE', 'ES', 'FR', 'GB', 'IT', 'NL', 'PT', 'US'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 8642, 8639, 8639
Resampling results:

```
  Accuracy   Kappa
  0.9760039  0.9711222
```

Boosting

```
> train_gboost
```

Output

```
Stochastic Gradient Boosting

12960 samples
    8 predictor
   10 classes: 'AU', 'CA', 'DE', 'ES', 'FR', 'GB', 'IT', 'NL', 'PT', 'US'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 8639, 8641, 8640
Resampling results across tuning parameters:

  interaction.depth  n.trees  Accuracy   Kappa
  1                   50      0.6679008  0.5831176
  1                  100      0.7099532  0.6395745
  1                  150      0.7266202  0.6613974
  2                   50      0.7516957  0.6930501
  2                  100      0.8262340  0.7871181
  2                  150      0.8689817  0.8404986
  3                   50      0.8203698  0.7800268
  3                  100      0.9108025  0.8921545
  3                  150      0.9378086  0.9250478

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter 'n.minobsinnode' was held constant at a value
 of 10
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were n.trees = 150, interaction.depth = 3, shrinkage = 0.1
and n.minobsinnode = 10.
```

> train_aboost

Output

AdaBoost.M1

12960 samples
8 predictor
10 classes: 'AU', 'CA', 'DE', 'ES', 'FR', 'GB', 'IT', 'NL', 'PT', 'US'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 8642, 8639, 8639
Resampling results:
Accuracy Kappa
0.982485 0.978935
Tuning parameter 'mfinal' was held constant at a value of 10
Tuning parameter 'maxdepth' was held constant at a value of 25

Tuning parameter 'coeflearn' was held constant at a value of Breiman

Accuracy and Kappa Statistic - Comparison

Algorithm	**Accuracy**	**Kappa Statistic**
Decision Tree	73.94289	68.11587
Neural Networks	82.1377	78.22056
SVM	45.65594	38.43945
Bagging	97.60039	97.11222
Boosting_StocGradientBoost	93.78086	92.50478
Boosting_AdaBoost	98.22531	97.86546

Observation

From the above table we see that the accuracy and Kappa values are very high for ensemble method techniques namely Bagging and Boosting.

ii. Classifiers_by_Split.R

Here we split the data - 80% for training and 20% for testing to find the accuracy using various algorithms and also plot the results accordingly. Along with the accuracies we also determine the confusion matrices for each of the algorithms. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Below are the confusion matrices, for various algorithms where:

Rows = Actual Values
Columns = Predicted Values

Decision Trees

```
> confusionMatrix_dtree

prediction_dtree  AU  CA  DE  ES  FR  GB  IT  NL  PT  US
              AU  69   0   0   0   0   0   0   0   0  33
              CA   0  30   0   0   0   0   0   0   0   0
              DE   0   0  49  10   0   0   0   0   0   0
              ES   0   0   0   0   0   0   0   0   0   0
              FR   0   0   0  12  88   0   0  13   6   3
              GB  69  26   0   0   0 589   6   0   0 132
              IT   0   0   0  35   0   0 128   0   0  18
              NL   0  54   0   0  17   0   0 586   8   0
              PT   0   0   0   0   0   0   0   0 108   0
              US  24  26  38  24   4  17   0   0   0 370
```

Neural Networks-Perceptron

```
> confusionMatrix_nn

prediction_perceptron  AU  CA  DE  ES  FR  GB  IT  NL  PT  US
                   AU  81   0   0   0   0   0   0   0   0  12
                   CA   0  11   1   0   6   6   0   6   0   8
                   DE   0   0  54   2   9   0   0   0   0  26
                   ES   0   0   0  59   0   0   3   0   0  16
                   FR   0  13   3   1  44   1   0  18   0   6
                   GB  56  35   0   0   0 545   1   0   0  43
                   IT   0   0   0  11   0   0 125   0   0   8
                   NL   0  69   0   0  11   0   0 554  13   0
                   PT   0   0   0   0  22   0   0  21 109   0
                   US  25   8  29   8  17  54   5   0   0 437
```

Neural Network

```
> confusionMatrix_ann

prediction_nn  AU  CA  DE  ES  FR  GB  IT  NL  PT  US
           AU 160   0   0   0   0   1   0   0   0   1
           CA   0 135   0   0   0   1   0   0   0   1
           DE   0   0  87   0   0   0   0   0   0   3
           ES   0   0   0  81   0   0   0   0   0   4
           FR   0   0   0   0 107   0   0   0   0   1
           GB   0   0   0   0   0 603   0   0   0   1
           IT   0   0   0   0   0   0 134   0   0   0
           NL   0   1   0   0   2   0   0 599   1   0
           PT   0   0   0   0   0   0   0   0 121   0
           US   2   0   0   0   0   1   0   0   0 545
```

SVM

```
> confusionMatrix_svm

prediction_svm  AU  CA  DE  ES  FR  GB  IT  NL  PT  US
            AU   0   0   0   0   0   0   0   0   0   0
            CA   0   0   0   0   0   0   0   0   0   0
            DE   0   0   0   0   0   0   0   0   0   0
            ES   0   0   0   0   0   0   0   0   0   0
            FR   0   0   0   0   0   0   0   0   0   0
            GB  72  28   2   0   9 542   3   0   0  48
            IT   0   0   0  14   0   0  87   0   0   0
            NL   0  84   0   0  21   0   0 588  36   0
            PT   0   0   0   0  37   0   0  11  84   0
            US  90  24  85  67  42  64  44   0   2 508
```

Naive Bayes

```
> confusionMatrix_nb

prediction_nb  AU  CA  DE  ES  FR  GB  IT  NL  PT  US
           AU  60   0   2   0   0   0   0   0   0  13
           CA   0   0   0   0   0   0   0   0   0   0
           DE   0   0  52   0  12   0   0   0   0  27
           ES   0   0   0  19   0   0   1   0   0   1
           FR   0   2   0   0  30   0   0   1   0   2
           GB  63  48   0   0   0 572   2   0   0  87
           IT   0   0   0   2   0   0 113   0   0   9
           NL   0  82   0   0  18   0   0 597  38   0
           PT   0   0   0   0  23   0   0   1  84   0
           US  39   4  33  60  26  34  18   0   0 417
```

Bagging

```
> confusionMatrix_bag

prediction_bag  AU  CA  DE  ES  FR  GB  IT  NL  PT  US
            AU 162   0   0   0   0   0   0   0   0   0
            CA   0 134   0   0   0   2   0   0   0   0
            DE   0   0  87   2   0   0   0   0   0   2
            ES   0   0   0  75   0   0   0   0   0   0
            FR   0   0   0   1 109   0   0   0   0   0
            GB   0   0   0   0   0 604   0   0   0   2
            IT   0   0   0   2   0   0 133   0   0   0
            NL   0   1   0   0   0   0   0 599   1   0
            PT   0   0   0   0   0   0   0   0 121   0
            US   0   1   0   1   0   0   1   0   0 552
```

AdaBoost

```
$confusion_adaboost
               Observed Class
Predicted Class  AU  CA  DE  ES  FR  GB  IT  NL  PT  US
             AU 123   0   0   0   0   4   0   0   0   8
             CA   0  62   0   0   0   1   0   0   0   3
             DE   0   0  51  11   0   0   0   0   0   1
             FR   0   0   0  12  88   0   0  13   6   3
             GB  24   4   0   0   0 599   2   0   0  77
             IT   0   0   0  35   0   0 128   0   0  18
             NL   0  54   0   0  17   0   0 586   8   0
             PT   0   0   0   0   0   0   0   0 108   0
             US  15  16  36  23   4   2   4   0   0 446

$error
[1] 0.1547068
```

Interpretation of Confusion Matrix:

For instance, using the AdaBoost algorithm,

- 162 destinations were predicted as AU. (total sum of the first column AU)

But in reality, out of the total 162 AU predictions:

- 123 destinations were correctly predicted as AU (refer to row AU, first column AU)
- 24 destinations were actually GB but incorrectly predicted as AU (refer to row GB, first column AU)
- 15 destinations were actually US but incorrectly predicted as AU (refer to row US, first column AU)

Hence, the above confusion matrix is a result of the predictions that were encountered upon the execution of adaBoost algorithm. The prediction had an error rate of 0.154.

Accuracy - Comparison

Algorithm	Prediction Accuracy
Decision Trees	77.81636
NN-Perceptron	77.89352
Neural Network	99.2284
SVM	69.79167
Naive Bayes	75
AdaBoost	84.52932

Observation

From the above table we see that Neural Networks produce the maximum prediction accuracy when compared to other algorithms, when we use classifiers by split.

FUTURE WORK

We can try collecting efficient information for providing further accuracy because the Airbnb data contains lot of non-numerical values. Hence it is difficult to implement k-NN classifier for this dataset. In addition to that, most of the data have faulty records. Lot of destinations are not found for the present dataset. Using that data for training the classifier might degrade the performance. Hence, if we could extract more accurate information from the data set, it would greatly aid in improvising the prediction results to a better extent. We would definitely consider expanding this project in future as there is a wider scope.

ACKNOWLEDGEMENT

We would like to thank Dr. Anjum Chida - Senior Lecturer, Department of Computer Science at The University of Texas at Dallas, for her effective teaching and making us understand various Machine Learning concepts and algorithms that were highly useful in developing this project.

CONCLUSION

Upon execution of various Machine Learning algorithms to the Airbnb dataset, we can see that the classifier selection depends largely on the data. In addition to that, data has multiple classification attributes for the predicting the right destination. Overall, the usage of ensemble methods - Bagging and Boosting, would be the ideal choice for this particular data. However, the time taken for execution of the ensemble methods, is large. The order of total time taken for training the datasets were - Neural Networks, Support Vector machines, Decision Tree, Perceptron and Ensemble methods. But it was in the reverse order for testing the dataset.

REFERENCES

1. *Simple guide to confusion matrix terminology*
2. *Comparison of Boosting Algorithms*
3. *Understanding the Kappa Statistic*
4. *Bagging in R*
5. *Using Naive Bayes to predict values*