Individual report

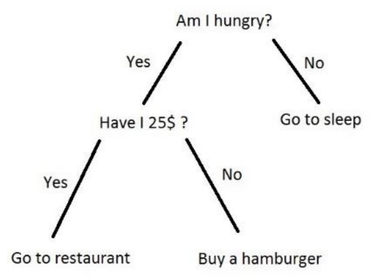
Huihan Gao

1.Introduction

We use the dataset from Kaggle, which is about Airbnb new user bookings. We only use the dataset called train_users_2 as training set and test_users as test set. And the label of this dataset is about the destination for the Airbnb users. In order to training the features in the dataset we use four models: navie bayes, decision tree, random forest and support vector machine. And the result shows that the most models among these four models is decision tree.

2.description algorithm

I focus on data preprocessing to deal with missing value and create features and fit decision tree model and the introduction.
Decision trees can be applied to both regression and classification problems. As the name suggests, a decision tree is a tree-like structure where internal nodes represent a test on a feature, each branch represents outcome of a feature, and each leaf node represents class label, and the decision is made after computing all features. A path from root to leaf represents classification rules. A simple decision tree example is just like the figure below.



Decision tree model's output is easy to interpret. Another advantage of a decision tree is that it is a non-parametric model so it requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Moreover, it can handle data of different types, including continuous, categorical, ordinal, and binary. It is also useful for detecting important variables, interactions, and identifying outliers

How the tree splits and grows?

The base algorithm is known as a greedy algorithm, in which the tree is constructed in a top-down recursive divide-and-conquer manner.

- At start, all the training examples are at the root.
- Input data is partitioned recursively based on selected attributes.
- Test attributes at each node are selected on the basis of a heuristic or statistical impurity measure example, gini, or information gain(entropy).
- Gini=$1-\sum_i (p_i)^2$,where $p_i$is the probability of each label.
- Entropy = $-\mathrm{plog2(p)}-\mathrm{qlog2(q)}$,where p and q represent the probability of success/failure respectively in a given model.

Conditions for stopping partitioning

- All samples for a given node belong to the same class.
- There are no remaining attributes for further partitioning. Majority voting is employed for classifying the leaf.
- There are no samples left.

Decision trees are simple to use, easy to understand, and offer many advantages compared to other decision-making tools, but they still don't find wide acceptance, due to its instability, overfitting and other limitations.

Introduction

In this report, we use the dataset from Airbnb new user bookings in a Kaggle competition to deal with the classification problem. The label for this dataset is first destination the users went to. We use four models to train the data and check their performance by using test data and using Kaggle to score our training result. The four models we use are Naïve Bayes, decision tree, random forest and support vector machine (SVM). We first doing the data exploration for this Airbnb dataset and then preprocessing for training model. Next, we start fit the model and explain the results we get. At last, we reach a conclusion for the work we have done.

Data pre-processing

This is section focusing on how to extract and build attributes from the training dataset. First, I merge the two datasets, train and test, in order to dealing with data more efficiently. Then I drop out the feature date_first_booking, which does not be contained in the test dataset.

NaN processing

When confronted a part of the data set that had a small amount of missing values, deleting whole rows or columns was impractical. Usually the missing data comes in the way of NaN, but in the data above we can see that the gender column, language column and first_browser column containing some values being -unknown- and first_affiliate_tracked column containing some values being untracked. I transformed those values into NaN first and then summarize the percentage of unknowns in each field. The missing values for all attributes in the train and test dataset are shown below:

Attribute name	Percentage of missing value
Age	42.412365
Country_destination	22.535538
First_affiliate_tracked	2.208335
First_browser	16.111226
Gender	46.990169

The table above shows that there are 42% in age attribute being missing values. And There are 46% in gender attribute being missing value. And There are 46% in gender attribute being missing value. It means that the missing value play an important role in the gender attribute, therefore we decided to set these missing vales as -1. We treat the -unknown- as another category. Then I use the same method to deal with other attributes expect for age attribute. I focus on age attribute independently. From the result of exploration part containing density distribution for age attribute. It is shown that there are some age values being greater than 1000, which is obvious impossible. This might be caused by using the born year as age. So we transfer these 'born-year' data as 'age' data. What's more, the Airbnb require the users should be older than 18 years old. Then we also set these people whose age is younger than 18 as Nan, whose values are -1. And for some people whose ages are bigger than 100, we also set its age as -1.

Feature engineering

- Date-related attributes

The raw data just put a series of numbers in the date-related attributes, date_account_created and timestamp_first_active. And then we split the date_account_created into day, week, month, year, which form four new features. At the same time, splitting the timestamp_first_active into day, weekday, week, month and year, which form five new features. And based on these two attributes we create a new feature called time_lag, which is the difference value between these two.

- Age attribute

After dealing with the NaN in the age column, we create age_group feature to set 14 categories, which divide the age into different goup. It will help us to know different age groups have different preference when using Airbnb.

- Other dummy attributes

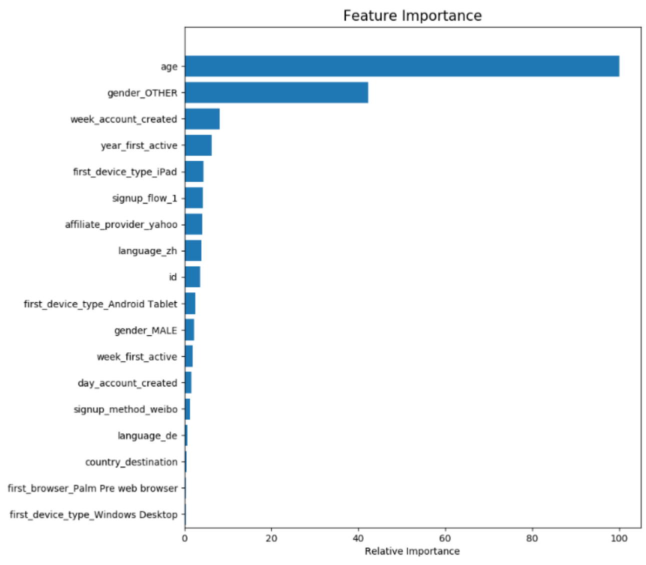
There are some remaining dummy attributes containing gender, signup_method, signup_flow, language, affiliate_channel, affiliate_provider, first_affiliate_tracked, signup_app, first_device_type and first_brower. We encode these category attributes.

At last, we have 169 features in our dataset.

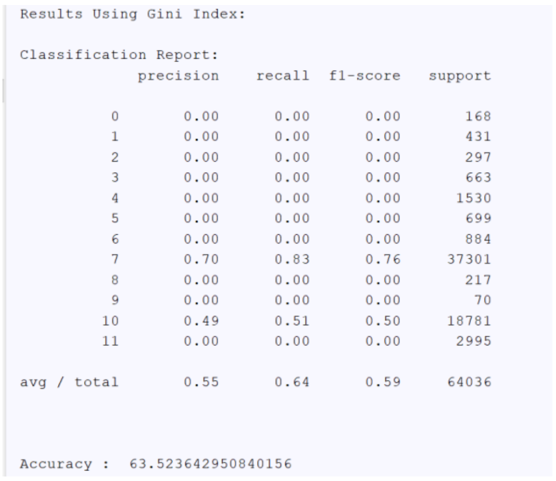
Decision Tree

I use decision tree for classification, the splits are chosen so as to minimize entropy or Gini impurity in the resulting subsets. The resulting classifier separates the feature space into distinct subsets. Prediction of an observation is made based on which subset the observation falls into. The decision tree is suitable for this kind of problem, because this is a multiclass classification problem.

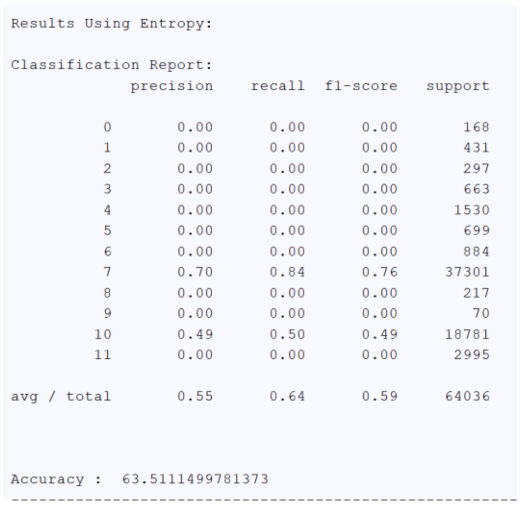
The result of feature importance of the attributes is shown below:



It shows that by using decision tree the most important feature is age attribute and second important attribute is gender_OTHER. And the relatively useless attribute is first browser used by users. From the result of decision tree, we notice that the root node for this training method is age_group attribute. And during the fitting process, we use both entropy and gini method to build the tree. The accuracy of these two methods are shown below:



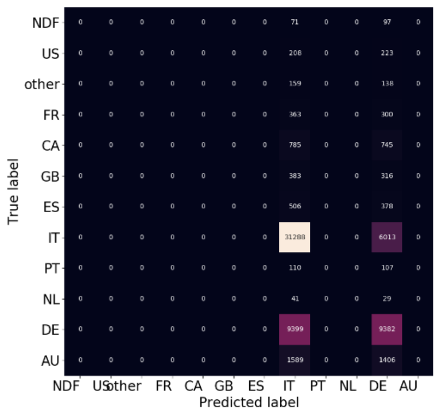
```
Results Using Gini Index:

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       168
           1       0.00      0.00      0.00       431
           2       0.00      0.00      0.00       297
           3       0.00      0.00      0.00       663
           4       0.00      0.00      0.00      1530
           5       0.00      0.00      0.00       699
           6       0.00      0.00      0.00       884
           7       0.70      0.83      0.76     37301
           8       0.00      0.00      0.00       217
           9       0.00      0.00      0.00        70
          10       0.49      0.51      0.50     18781
          11       0.00      0.00      0.00      2995

avg / total       0.55      0.64      0.59     64036


Accuracy :  63.523642950840156
```

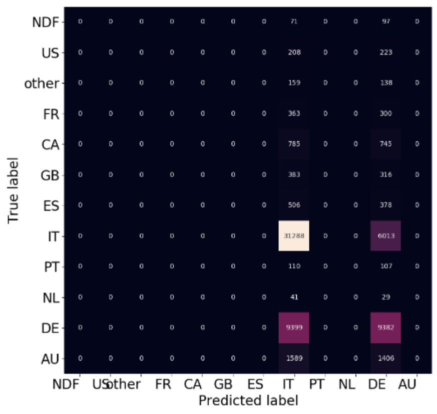


```
Results Using Entropy:

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       168
           1       0.00      0.00      0.00       431
           2       0.00      0.00      0.00       297
           3       0.00      0.00      0.00       663
           4       0.00      0.00      0.00      1530
           5       0.00      0.00      0.00       699
           6       0.00      0.00      0.00       884
           7       0.70      0.84      0.76     37301
           8       0.00      0.00      0.00       217
           9       0.00      0.00      0.00        70
          10       0.49      0.50      0.49     18781
          11       0.00      0.00      0.00      2995

avg / total       0.55      0.64      0.59     64036


Accuracy :  63.5111499781373
-----------------------------------------------------------------
```

The results tell us that using these two methods make no big difference for accuracy of decision tree. In fact, using entropy is computational expensive. But I still fit two models by using both methods.

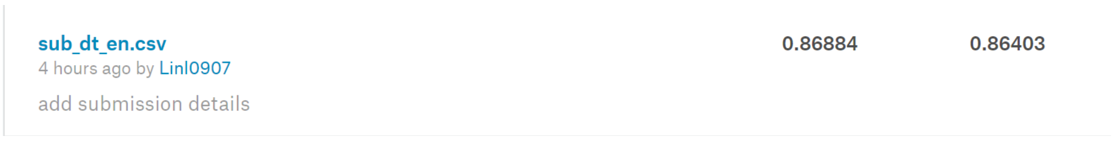
And about the results of confusion matrix are shown below:





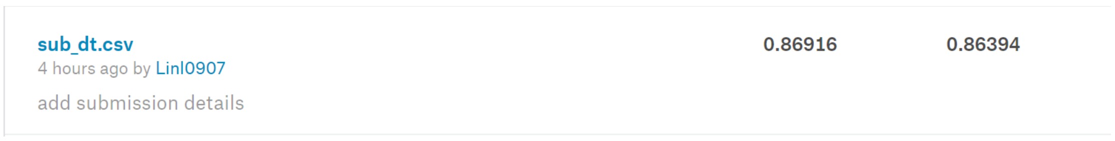
The confusion matrices are derived from entropy and gini index from left to right. The results of both seem similar. They both show that the decision tree misclassified IT and DE. There are 23.100% of IT has been classified to DE. And there are 39.058% of DE has been misclassified to IT. It shows that many predicted labels has been classified to IT and DE incorrectly.

The results in Kaggle are as following:



sub_dt_en.csv
4 hours ago by Linl0907
add submission details

0.86884 0.86403


sub_dt.csv
4 hours ago by Linl0907
add submission details

0.86916 0.86394

The results show that the decision tree performed well on this destination classification problem.

5. Summary and conclusion

From my part, I found that the attribute age is the most important feature among all 169 attributes when using decision tree algorithm. And using entropy or gini makes no big difference to the accuracy. And comparing with other algorithm the decision tree algorithm seems more efficient and more accurate than other three models. If there is other thing to be done, I think we should try other algorithms like XGboost and moreover try to use framework like tensorflow and keras.

6. Coding percentage

The percentage of code is 64%.

7.Reference

Python, P. D. A. U., & Swamynathan, M. Mastering Machine Learning with Python in Six Steps.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: springer.