Classification Model Development Based on Cluster-to-Class Distance Mapping for Tourism Form Prediction of Inbound Tourism Market in Thailand

Unnadathorn Moonpen[1], Surasak Mungsing[1] and Thepparit Banditwattanawong[2]*

[1]Faculty of Information Technology, Sripatum University, Bangkok, Thailand
[2]Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand

Received: 16 July 2020, Revised: 9 November 2020, Accepted: 27 December 2020

Abstract

This paper describes the classification model development of inbound tourism form in Thailand. The models utilized both labeled and originally unlabeled data sets. The latter data set, which was obtained from the Ministry of Tourism and Sports of Thailand that regularly collects unlabeled data, mandated the synthesis of tourism form labels to be usable for classification. To achieve such a label synthesis, we proposed a cluster-to-class mapping algorithm that consisted of three steps. First, searching the best tourist clustering model among the unlabeled tourist data set by comparing the results of K-means, hierarchical cluster analysis, random clustering, and DBSCAN techniques. Second, mapping the clusters to the classes of the labeled data set based on Euclidean similarity to reveal the tourism form labels for the clusters. Finally, searching the best tourism-form classification model based on the data sets with real and synthesized labels by engaging Naïve Bayes, support vector machine, linear regression, and decision tree techniques. Experimental results show that our algorithm effectively generated the tourism form labels since, when using them, we obtained a neutral network model that was capable of predicting the inbound tourism forms of an unseen tourist data set with an F-measure value as high as 98.99%.

Keywords: tourism form; classification algorithm; clustering algorithm; cluster-to-class mapping
DOI.................

1. Introduction

Tourism industry plays a leading role in Thailand's economy as domestic and international tourism contributes 2.01 trillion baht per annum by average to the Thai national Gross Domestic Product (GDP) or 15% of total GDP. Moreover, the tourism industry has in recent years employed 4,129,382 people per year or 10.82% of national employment by average. Specifically, income from inbound tourism averaged as high as 10% of the national tourism direct gross domestic products (TDGDP) [1]. It was expected that a tourism situation in Thailand in 2020 would gain a contribution by 37 million foreign tourists or 1.73 trillion baht, which has declined from the 2019 report on the number of 39.8 million foreign tourists who contributed 1.88 trillion baht.

*Corresponding author: Tel: +66(0)2942 8200-45
E-mail: thepparit.b@ku.th

To stimulate the tourism industry's resiliency in the second half of the year after the end of COVID-19 pandemic is deemed important [2]. One strategy that can be used to boost Thailand tourism is to exploit technology and digital platform to meet the personal needs and lifestyles of tourists in terms of tourism forms [3]. Therefore, relevant marketing analysts and entrepreneurs must understand and adapt to such needs [4] although tourism planning and developing is large-scaled and requires highly accurate forecasting to reduce potential risk in a decision making process [5]. The forecasting can be conducted by utilizing recorded tourist data that is collected regularly by Office of the Permanent Secretary of Ministry of Tourism and Sports. Unfortunately, such data is totally unlabeled and inbound tourist data has no label of tourism forms because the data collection has been done without the awareness of the tourism forms's benefits. The unlabeled data cannot be used to enable both tourism-relevant public and private sectors to discover and respond to the lifestyle of inbound tourists. To solve such a problem, the unlabeled data must be processed to synthesize tourism form labels that are useful for the forecasting of the needs of inbound tourists to advocate the inbound tourism market in Thailand.

In Panawong *et al*. [6], classification model of tourism forms, synthesized by latent semantic analysis and machine learning, was created based on 10,250 places of data and 11 types of tourism places. The results were that support vector machine (SVM)- and back propagation neural network (BPNN)-based models using such latent semantic analysis were effectively accurate by 77.82% and 75.96%, respectively. In Chatcharaporn *et al*. [7], Naïve Bayes was used to classify Thailand tourism websites based on a lightweight tourism ontology into 6 types from totally 475 websites. It showed that classification model was effectively accurate to 97.39%. Market segmentation of inbound tourism for foreign tourists in Thailand revealed that the most accurate classification model of tourists was Naïve Bayes [8]. Liu *et al*. [9] created a cluster of new entry tourists and analyzed the main features of tourism packages and seasonal tourism development that defined the identity of tourism packages and gave effective recommendation to individual tourist tourism packages. Types of travelling residents at destinations were analyzed from tourist behaviors and the tagging period of photos taken by the tourists along their roaming places [10]. Empirical analysis system based on neural network for tourism resources appraisal showed that ecosystem and culture values were effectively able to build up the values of recreational resources [11]. The unsupervised machine learning of the market segmentation of leading international tourism businesses in Thailand [8] reported that the best method was the K-means technique that clustered into 5 segments. The segmentation of local tourists in Thailand based on a correlation-based weighting algorithm, self-organizing map (SOM), K-means, and Fuzzy C-Mean revealed that the weighting algorithm performed best [12]. Cufoglu [13] found that hierarchical clustering both forward and backward of tourists yielded a precision of 85.51% for recommending tourism services to individuals. Rodríguez *et al*. [14] developed a hierarchical clustering approach for smartphone geo-localized data to detect meaningful tourism-related market segments. Clustering results were divided into two main clusters and four sub-clusters that could be interpreted according to tourists' temporary spatial patterns and the repetition of visiting patterns. Based on these related works, there is no method for transforming unlabeled tourism data into labeled data. Therefore, a process for the generation of labels for tourism forms facilitating effective classification is needed.

For the above reason, we propose in this paper a novel algorithm for tourism-form label synthesis based on Euclidian distance to create effective forecasting models of tourism forms for Thailand's inbound tourism market.

2. Methodology

Figure 1 portrays the conceptual framework of our research and is described as follows. There are two input data sets: data set 1 contains tourism form labels whereas data set 2 has no tourism form labels. Data set 2 is initially clustered by using the most efficient clustering model that emerges from the experiments of K-mean, hierarchical clustering, random clustering, and density-based spatial clustering of applications with noise (DBSCAN) algorithms. The best model is measured based on a metric called Davies-Bouldin index. Then, the resulting clusters together with data set 1 is exploited by our proposed cluster-to-class mapping algorithm to generate tourism form labels for all records in data set 2. Subsequently, data set 2 with labels and data set 1 are used to train and test classification models to seek the most effective one based on Naïve Bayes, neural network, support vector machine, linear regression, and decision tree algorithms. The performance of the classification models are measured in terms of accuracy, precision, and recall as well as F-measure. By comparing such models' performance values, the effectiveness of our algorithm is finally revealed.

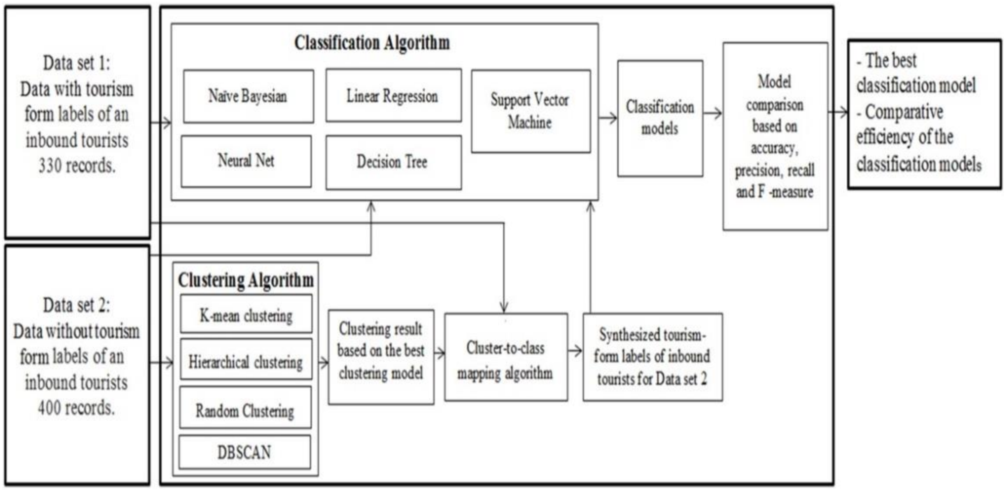

Data set 1:
Data with tourism form labels of an inbound tourists 330 records.
Classification Algorithm
Naïve Bayesian
Linear Regression
Support Vector Machine
Neural Net
Decision Tree
Classification models
Model comparison based on accuracy, precision, recall and F -measure
- The best classification model
- Comparative efficiency of the classification models
Data set 2:
Data without tourism form labels of an inbound tourists 400 records.
Clustering Algorithm
K-mean clustering
Hierarchical clustering
Random Clustering
DBSCAN
Clustering result based on the best clustering model
Cluster-to-class mapping algorithm
Synthesized tourism-form labels of inbound tourists for Data set 2

Figure 1. Conceptual framework

2.1 Definition of tourism forms in Thailand

Since we utilized Thailand's tourism forms as classification labels throughout this paper, it is important to understand what they mean. Tourism authority of Thailand (TAT) has classified the inbound tourism forms into 10 forms as follows [15].

1) Eco-tourism is a tourism form that is exclusively concerned with local natural feature having environment management and local cooperative tourism focusing on building conscious mind toward sustainable ecosystem preservation.

2) Arts and science educational attraction standard is defined as a tourism form related to the visiting places of special interest, for example, exclusive museum, educational tourism places including science, industry, technology, and meeting, and seminar.

3) Historical attraction is a tourism form that focused on places with historical values such as archaeological sites, religion, and ancient places, or those with historical parks, fortifications, museums, temples, and places of worship.

4) Natural attraction is a tourism form with the sightseeing of nature such as geographical landscapes, waterfalls, and the mighty symbols of local areas.

5) Recreational attraction is a tourism form focuses on recreation, amusement, entertainment, and education such as entertainment, zoo, amusement park, and the places of Thai culinary experience.

6) Cultural attraction is a tourism form involved with the values of art and culture that are legacies from generation to generation, for example, festivals, lifestyle, art exhibition, culture, local products, dress code, language, and tribal groups.

7) Health tourism is a tourism form with the purposes of medical treatment, hot spring, spa, and health massage.

8) Sea and beach attractions consist of sea and beach as the natural locations that offer activities for visitors, for example, swimming, sun-bathing, diving, water sports, and beachside recreation.

9) Sport tourism focuses on sport activities for tourists or sport competition.

10) Adventure tourism is a form with extreme activities such as hiking and mountain biking.

2.2 Data gathering and engineering

We collected data set 1 at Suvarnabhumi airport in 2019 from the questionnaire surveys of 330 inbound tourists traveling back to their countries. The data set included the tourists' tourism forms according to Section 2.1. Data set 2 was obtained via an official data acquisition process from the Ministry of Tourism and Sports of Thailand in 2017. The data set was collected from 400 tourists. This data set was totally unlabeled in terms of tourism forms unlike data set 1. However, both data sets contained the same 26 basic attributes: tourist ID, gender, age, marital status, region, nationality, occupation, income, tour purpose (e.g., business, conference, studying, sightseeing), sea and beach, eco-tourism, adventure, historical city tour, learn local resident, medical treatment, spa and wellness, Thai food, night life, theme park entertainment, nation special event festival, diving snorkeling, golfing, shopping, Thai boxing, Thai cooking class, and other activities. The values of the 10th to the last attributes were binary (i.e., yes or no) while the other attributes had nonnegative integer values. Tourist ID represented the seasonal behavior of tourists. Tourist ID was actually the ID of questionnaire taken by each tourist. The questionnaires were collected in every quarter throughout the year. Therefore, tourists with nearby tourist ID values might favor similar tourism forms. As for data set 1, tourism type was additionally the 27th attribute whose possible values were of the 10 tourism forms and served as the labels. All attribute values in both data sets were also converted to integers, and fortunately both data sets had no missing values.

2.3 Algorithm and model development

This section describes the formulation of our tourism-form-label synthesis algorithm, which was used to unlock the benefits of data set 2. The steps are explained below while their corresponding results will be reported in Section 3. As for experimental tools, we employed RStudio version 3.6.1 [16] and RapidMiner version 9.5 [17].

2.3.1 Clustering unlabeled data

We experimented four well-known clustering algorithms to construct the best clustering results based on data set 2 to be used in the next step. Clustering is unsupervised learning for grouping data [18] based on partitioning, hierarchy, density, or grids [19]. We shortly describe such algorithms below, which are the basis of our proposed algorithm.

K-mean clustering [18] performs data partitioning based on the predefined number of clusters that is called k. Let a data set $T = \left\{x_1, x_2, \dots, x_{|T|}\right\}$ and an attribute set $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ where n is the number of dimensions. Let $C = \{c_1, c_2, \dots, c_k\}$ be k clusters, each of which has a member set $c_j = \left\{x'_{j1}, \dots, x'_{j|c_j|}\right\}$ where c_j is a member of C, and x_i is a member of T. The formal description of K-means can be described in equation (1)

$$d(\mathrm{x_i}, \mathrm{c'_j}) = \left|\mathrm{x_i} - \mathrm{c'_j}\right|, \mathrm{c'_j} = \frac{1}{|\mathrm{c_j}|}\sum_{\mathrm{k=1}}^{|\mathrm{c_j}|} \mathrm{x'_{jk}} \qquad (1)$$

where d represents the distance from x_i to c'_j. x_i is represented by an n-dimensional attribute vector. c'_j represents the centroid of each cluster. x'_{jk} dictates the member object of c_j dimensions. x_i is assigned to c'_j if their $d(\mathrm{x_i}, \mathrm{c'_j})$ is minimal. To calculate d, Euclidean distance is engaged in this paper.

Hierarchical clustering [18] relies on a tree structure called a dendrogram. This approach groups data into a tree of clusters without a predefined number of clusters by merging the similar objects or object groups and splitting dissimilar objects or object groups. There are two types of hierarchical clustering methods, agglomerative and divisive, depending on whether the hierarchical structure (tree) is formed in either bottom-up (merging) or top-down (splitting) style. The agglomerative hierarchical clustering starts with having each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are included in a single cluster or until some certain termination conditions are satisfied. On the other hand, the divisive hierarchical clustering performs the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster and subdivides it into smaller and smaller pieces, until each object forms a cluster on its own or until satisfying certain termination conditions, such as a desired number of clusters is obtained or the diameter of each cluster is within a certain threshold. Four widely used measures of distance between two clusters are single linkage, complete linkage, centroid comparison, and element comparison. In this research, agglomerative hierarchical clustering and the criteria for the single linkage (minimum distance) defined in equation (2) were utilized. Single linkage (minimum distance) is

$$d_{min}\left(C_iC_j\right) = min_{p\in C_i, p'\in C_j}|p - p'| \qquad (2)$$

where C_i is cluster i and C_j is cluster j, p is distance between objects in C_i, p' is distance between objects in C_j and $|p - p'|$ is distance from C_i to C_j [20].

Random clustering was generated by shuffling the elements between the fixed clusters. A prevailing assumption for the random clustering ensemble is the permutation model in which the number and sizes of clusters are fixed [21]. If ball 1, ..., ball k are thrown to form a partition $A \in A_k$, then ball $k + 1$ is put into an empty urn with probability p_k and into an urn with j balls with probability $(1 - p_k)j/k$. This ball throwing is continued for $k = 1, 2\ 3, \dots$. Mathematically, the clusters at the nth step form a partition of the finite set $u_n = \{1, 2, \dots, n\}$, i.e., the set of labeling numbers on balls up to ball n. The family of all partitions of u_n is denoted by A_n. If a partition $A \in A_n$ has s_j subsets of cardinality j (i.e., s_j clusters of size j or s_j urns with j balls), $j = 1, \dots, n$. At the nth step of the random clustering process mentioned above, the probability that ball 1, ..., ball n form a partition $A \in A_n$, as shown in equation (3) [22].

$$P(A; A_n) = f_n(s) = f_n(s;p) = \frac{\propto^u}{\propto^{[n]}}\prod_{j=1}^{n}((j-1)!)^{s_j}, \ 0 < p < 1, \ 0 < \alpha < \infty, \qquad (3)$$

Define $s = S(A) \in s_n, u = \sum_{j=1}^{n} s_j$, and $\propto^{[n]} = \propto (\propto +1) \cdots (\propto +n-1), \qquad \propto = p/(1-p)$.

where $P(A; A_n)$ is independent of the order of n balls thrown in, and is invariant with respect to the permutation of the indices of the balls. $S(A)$is the size index of A. $f_n(s)$ is the invariance with respect to the indexing of balls in $S(A)$, $f_n(s;p)$ is the invariance with respect to the indexing of the probability of $S(A)$, u is the number of cycles of $A \in A_n$, $\propto$ is the Poisson approximation, p is the probability of an event in sample space A_n. n is thrown to form a partition $A \in A_n$.

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering algorithm that finds the regions of objects with sufficiently high density into cluster. By this nature, it discovers clusters of arbitrary shape in spatial databases with noise. In this method, a cluster is defined as a maximal set of density-connected points. Neighborhood objects within a radius ε of a given object is called the $\varepsilon - neighborhood$ of the object. Each core object has a *minimum number* called *MinObjs*. Given a set of objects D, an object p is directly density-reachable from the object q if p is within the $\varepsilon - neghborhood$ of q, and q is core object. An object p is density-reachable from object q within respect to ε and *MinObjs* in a set of objects, D, if there is a chain of objects $p_1, \ldots, p_n$, where $p_1 = q$ and $p_n = p$ such that p_{i+1} is directly density-reachable form p_i with respect to ε and *MinObjs*, for $1 \leq i \leq n,\ p_i \in D$. An object p is density-connected to object q with respect to ε and *MinObjs* in a set of object, D, if there is an object $o \in D$ such that both p and q are density-reachable from o with respect to ε and *MinObjs*. Finally, a density-based cluster is a set of density-connected objects that is maximal with respect to density-reachability. Every object not contained in any cluster is considered to be noise [18, 23].

Besides the clustering algorithms, a clustering performance metric enlisted in our experiments was Davies–Bouldin index (DBI). DBI value is high when data within the same cluster has high similarity and data between clusters has low similarity. DBI can be calculated by the following formula [24]:

$$DBI = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \max_{\mathrm{i} \neq \mathrm{j}} \left(\frac{\sigma_{\mathrm{i}} + \sigma_{\mathrm{j}}}{d(c_i, c_j)} \right) \tag{4}$$

where n is the number of clusters, c_x is the centroid of cluster x, σ_x is the average distance of all elements in cluster x to centroid c_x, and $d(c_i, c_j)$ is the distance between centroids c_i and c_j. Since algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low DBI, a clustering model that yields the smallest DBI is considered the best. Table 1 lists the parameter values of experimented clustering algorithms.

Table 1. Optimal parameter for each classification model

Algorithm	**Parameter configuration**
K-mean clustering	k=10, max runs=30, Euclidean Distance, Numerical Measure, max optimization steps =100
Hierarchical clustering	k=10, max runs=30, Euclidean Distance, max optimization steps =100, Single Link
Random clustering	k=10, max runs=30, Euclidean Distance, Numerical Measure, max optimization steps =100
DBSCAN	k=10, max runs=30, Euclidean Distance, Numerical Measure, epsilon=0.25, Min point=0.5

As the result of this step, 10 optimal clusters produced by the lowest DBI model will be engaged in the next Section.

2.3.2 Cluster to class mapping

This section describes our cluster-to-class mapping approach that consists of 4 steps as follows.

1) The 10 optimal clusters had their centroids calculated. Similarly, 10 data classes (according to data labels) of data set 1 had their centroids figured out as if each class was a cluster. The calculation was done by averaging the 26 attribute vectors of all records belonging to the same cluster or class as in equation (5) [18, 25].

$$\mathrm{c'}_{\mathrm{i}} = \frac{1}{|\mathrm{c_i}|}\sum_{\mathrm{k}=1}^{|\mathrm{c_i}|} \mathrm{x'}_{\mathrm{ik}} \tag{5}$$

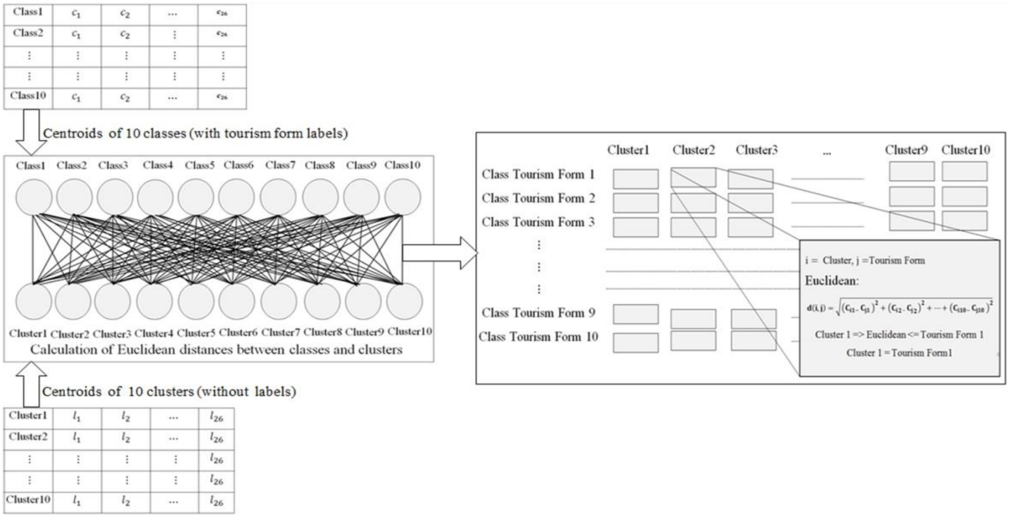
where c'_i is the centroid vector of cluster or class $\mathrm{c_i}$, x'_{ik} represents each element of c_i and has 26 dimensions.

2) In this step, we calculated the Euclidean distance between the centroids of each possible pair of cluster and class. Since there were 10 clusters and 10 classes, there were totally 100 possible distances in total to compute. The formula of Euclidean distance simply follows equation (6):

$$d(i,j) = \sqrt{\left(\mathrm{C}_{\mathrm{i1}-}\,\mathrm{C}_{\mathrm{j1}}\right)^2 + \left(\mathrm{C}_{\mathrm{i2}-}\,\mathrm{C}_{\mathrm{j2}}\right)^2 + \cdots + \left(\mathrm{C}_{\mathrm{i}p-}\,\mathrm{C}_{\mathrm{j}p}\right)^2} \tag{6}$$

where $d(i,j) \geq 0$ is a distance from object i to object j. The strength of this algorithm was a non-variable to the interpretation and rotation of featured area [26] that is suitable for our mapping purpose.

The ideas of the first two steps are clearly illustrated in Figure 2. The centroid vectors of 10 classes and the centroid vectors of 10 clusters are paired to compute 100 possible Euclidian distances. The distances of these pairs are represented as a 10 x 10 matrix with its rows representing 10 tourism forms (i.e., 10 classes) and its columns representing 10 clusters. Each element in the matrix notates each of the calculated Euclidian distance.



Class1	c_1	c_2	...	c_{26}
Class2	c_1	c_2	⋮	c_{26}
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
Class10	c_1	c_2	...	c_{26}

Cluster1	l_1	l_2	...	l_{26}
Cluster2	l_1	l_2	...	l_{26}
⋮	⋮	⋮	⋮	l_{26}
⋮	⋮	⋮	⋮	l_{26}
Cluster10	l_1	l_2	...	l_{26}

Centroids of 10 classes (with tourism form labels)
Class1 Class2 Class3 Class4 Class5 Class6 Class7 Class8 Class9 Class10
Cluster1 Cluster2 Cluster3 Cluster4 Cluster5 Cluster6 Cluster7 Cluster8 Cluster9 Cluster10
Calculation of Euclidean distances between classes and clusters
Centroids of 10 clusters (without labels)
Cluster1 Cluster2 Cluster3 ... Cluster9 Cluster10
Class Tourism Form 1
Class Tourism Form 2
Class Tourism Form 3
Class Tourism Form 9
Class Tourism Form 10
i = Cluster, j =Tourism Form
Euclidean:
d(i,j) = √((Ci1- Cj1)² + (Ci2- Cj2)² + ··· + (Ci10- Cj10)²)
Cluster 1 => Euclidean <= Tourism Form 1
Cluster 1 = Tourism Form1

Figure 2. Euclidean distance-based cluster-to-class mapping

3) The 100 Euclidean distance values are sorted to find out 10 class-cluster pairs whose distances were shortest and classes were unique. The 10 classes served as tourism forms associated with the 10 clusters. In this way, tourism form labels were synthesized for each record of data set 2.

All these three steps are described as our proposed algorithm in Table 2. Input variables are C1 (the vector of 10 tourism form classes x_1 to x_{10}) and C2 (the vector of 10 tourist clusters y_1 to y_{10}). Output variable L is the vector of 10 tourism forms, each of which is associated with each member of C2 (i.e., l_1 is tourism form of y_1, l_2 is tourism form of y_2, and so on). Line (1) to line (6) calculate Euclidean distances between 10 tourist clusters and 10 tourism form classes. The result is stored in 10 x 10 sized distance matrix, EDmatrix, (i.e., the 10 x 10 matrix in Figure 2) in line (3). Line (7) transforms EDmatrix to a vector of 100 Euclidean distance values. The 100 values are then sorted on line (8). Vector L in line (9) is initialized to nulls. The loop on line (10) repeats until unique labels are associated with all 10 clusters in C2 and saved into L. Line (11) to line (13) search for the shortest Euclidean distance with respect to each class.

2.3.3 Classification performance comparison

At this point, both data set 1 and data set 2 have been labeled. We deployed them as training and test sets in combination to construct four classification models: the first model was trained and tested by using data set 1 based on a hold-out technique, the second model was trained and tested by using data set 2 based on a hold-out technique, the third model was trained by data set 1 and tested by using data set 2, and the last model was trained by using data set 2 and tested by using data set 1. The performances of these models are compared to conclude research findings.

Table 2. Thailand-inbound-tourism-form label synthesis algorithm by mapping 10 clusters to 10 classes based on Euclidean distances

Algorithm: Tourism form label synthesis

```
Input     C1 is vector of 10 tourism form classes, ($x_{1,}$ $x_{2,}$ …,$x_{10}$) where $x_i$ has centroid vector ($x_{i1,}$ $x_{i2,}$ …,$x_{i26}$)
          C2 is vector of 10 tourist clusters, ($y_{1,}$ $y_{2,}$ …,$y_{10}$) where $y_i$ has centroid vector ($y_{i1,}$ $y_{i2,}$ …,$y_{i26}$)
Output  L = ($l_{1,}$ $l_{2,}$ …,$l_{10}$) is vector of tourism forms associated with members of C2
Begin
(1)      For (i ← 1; i=i+1; i<=10) do
(2)         For (j ← 1; j=j+1; i<=10) do
(3)            EDmatrix ← $\mathrm{sqrt}(\sum_{k=1}^{26}(x_{ik} - y_{ijk})^2)$ // Euclidean distance vector between C1& C2 is stored in 10 x 10
(4)    // distance matrix. EDmatrix’s rows represent 10 tourism type classes. Matrix1’s columns represent 10 clusters.
(5)         End for
(6)       End for
(7)       M ← convertMatrixToRowVector(EDmatrix) // convert EDmatrix to row vector
(8)       S ← sort (M)
(9)       L ← (null, …, null)
(10)      While (countNotNullMembersOf(L) < 10) // while not all 10 tourism types are assigned as clusters’ labels
(11)         u ← min(S) // assign minimum element of S to u
(12)         S ← S-u // remove u from S
(13)         z ← getClassFromRowIndexInEDmatrixOf(u) // assign class name associated with u in EDmatrix to z
(14)         If (z ∉ L) // if z has not been assigned as label
(15)                L[getColumnIndexOf(u)] ← z //Assign z as label by storing it in L at the same column as that of u in
(16)                                            // EDmatrix
(17)     End while
(18)     Return(L)
End.
```

To create the models, we employed five well-known classification algorithms as follows. Classification refers to a supervised machine learning technique used to find objects of relevant class to help with human decision making such as forecasting unexpected events based on presented input data [18, 27].

Naïve Bayes is possible to presume that all attributes are independent of each other. The values of the attributes are conditionally independent of one another, given the class label of the object, the formula as shown in equation (7):

$$c(x) = {}^{argmax}_{c_i \in C} P(c_i)P(a_1(x)|c_i)P(a_2(x)|c_i) \dots P(a_n(x)|c_i) \tag{7}$$

In this formula, each term, except the first term, is the probability to obtain the attribute value, given only the class value. Assume that $a_3(x)$ is dependent on $a_2(x), a_4(x)$ is dependent on $a_1(x)$ and $a_3(x)$, and others are independent of each other. $P(c_i)$ is the prior probability of $c_i . P(a_1(x)|c_i)$ is the first number of objects classes c_i in the training data set. $P(a_2(x)|c_i)$ $P(a_n(x)|c_i)$ is the second number of the number of objects classes c_i in the training data set.

Support vector machine (SVMs) determines a decision boundary to be as far away from the data of two classes as possible. Given the training data $\{x_i, y_i\},\ i = 1, \dots n,\ \mathrm{x_i} \in \mathrm{R}^d, y_i \in \{-1,1\}$ as shown in equation (8):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}||\mathrm{w}||^2 \\ &\text{subject to} \quad y_i(w^T x_i + b) - 1 \geq 0 \ \text{ for all } \ i \end{aligned} \tag{8}$$

where w is the Euclidean distance, x_i is datum, represented as a vector of d dimension, y is the binary class of -1 or +1, the support vector machine finds the best hyperplane which separates the positive from the negative examples. The point x_i on the hyperplane satisfies the formula $w^T x_i + b = 0$, where w is a normal vector that is perpendicular to the hyperplane. $|b|/||\mathrm{w}||$ is the perpendicular distance from the hyperplane to the origin.

A neural network that was employed in this paper was specifically a multilayer feed-forward neural network consisting of an input layer, one or more hidden layers, and an output layer. Neural networks operate by requiring cooperation through different nodes in the layers to forecast a result as shown in equation (9):

$$\mathrm{y_i} = \int\left(\textstyle\sum_{\mathrm{i=0}}^{\mathrm{N}} \mathrm{w_{ij}x_i} + \theta_\mathrm{j}\right) \tag{9}$$

where y_i is the output layer, x_i is the input layer, N is the total number hidden layers, w_{ij} is a weight for node i to node j, and θ_j is the bias associated with j.

Linear regression uses a straight line to approximate correlation between predictor variables and responsive variables as shown in equation (10):

$$w_0 = \bar{y} - w_1\bar{x} \tag{10}$$

where w_0 and w_1 are regression coefficients or weights, $\bar{x}$ is the mean of $x^{(1)}, x^{(2)}, \dots, x^{(|s|)}$, and $\bar{y}$ is the mean of $y^{(1)}, y^{(2)}, \dots, y^{(|s|)}$, x is predictor variable of presented data, y is responsive variable for coefficient. |s| is a rule of sample of test data set s from $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(|s|)}, y^{(|s|)})$.

A decision tree is a tree-like graph consisting of three components: (1) leaf nodes (rectangular), (2) decision criterion nodes (ovals), and (3) decision branches (lines). The leaf nodes represent classification (decision) outcomes, the root and the intermediate nodes express decision criteria. Given a trained set of object and their associated class label, denoted by $T = \{x_1, x_2, \dots, x_{|\mathrm{T}|}\}$, each object x_i is represented by an n–dimensional attribute vector, $x_i = (x_{t1}, x_{t2}, \dots, x_{\mathrm{tn},}, c_t)$, depicting the measure values of n attributes, $A_1, \dots, A_n$, of the object with its class $c_i \in C$, one form m possible class, $C = \{c_1, c_2, \ldots, c_\mathrm{m}\}$. Here, suppose that A_i has n_i possible values $\{a_{i1}, a_{i2}, \ldots, a_{\mathrm{i}(n_i)}\}$. That is, $x_{it} \in \{a_{i1}, a_{i2}, \ldots, a_{\mathrm{i}(n_i)}\}$, a binary partition p divides the values of the attribute A_i. The decision tree selects the best attribute as the first node in order to split the training set into a number of subsets. To select such an attribute, we used the gain ratio defined in equation (11).

Information gain:
$$\mathrm{InfoGain}(\mathrm{T}, \mathrm{A_i}) = \mathrm{Info}(\mathrm{T}) - \sum_{\mathrm{k}=1}^{n_i} \frac{|\mathrm{T_{ij}}|}{|\mathrm{T}|} Info\left(\mathrm{T_{ij}}\right) \tag{11}$$
$$\mathrm{Info}(\mathrm{T}) = \sum_{k=1}^{m} (-1) \cdot p(c_k, T) \cdot \log_2\big(p(c_k, T)\big)$$
$$\mathrm{Info}\big(\mathrm{T_{ij}}\big) = \sum_{k=1}^{m} (-1) \cdot p(c_k, \mathrm{T_{ij}}) \cdot \log_2\left(p\big(c_k, \mathrm{T_{ij}}\big)\right)$$
$$\text{Where } p(c_k, T) = \frac{|\mathrm{T_k}|}{|\mathrm{T}|} \text{ and } p(c_k, \mathrm{T_{ij}}) = \frac{|\mathrm{T_{ijk}}|}{|\mathrm{T_{ij}}|}$$

Gain ratio:
$$\mathrm{GainRatio}(\mathrm{T}, \mathrm{A_i}) = \frac{\mathrm{InfoGain}(\mathrm{T}, \mathrm{A_i})}{\mathrm{SplitInfo}(\mathrm{T}, \mathrm{A_i})}$$
$$\mathrm{SplitInfo}(\mathrm{T}, \mathrm{A_i}) = \sum_{\mathrm{k}=1}^{\mathrm{m}} (-1) \cdot \mathrm{p}\big(\mathrm{T_{ij}}\big) \cdot \log_2\left(\mathrm{p}\big(\mathrm{T_{ij}}\big)\right)$$
$$\text{where } \mathrm{p}\big(\mathrm{T_{ij}}\big) = \frac{|\mathrm{T_{ij}}|}{|\mathrm{T}|}$$

where T is a training set before splitting, node A_i is selected for splitting, c_k is the *k*-th class, T_k is the set of the instances with the class c_k in the set T, T_{ij} is a subset of the training set after splitting, contains the objects which have the value of a_{ij} for the attribute A_i, that is $A_i = a_{ij}$ and T_{ijk} is the set of the instances with the class $\mathrm{c_k}$ in the subset T_{ij}.With this notation, $|T|$ is the total number of instances with the training set before splitting, $|T_k|$ is the number of class-*k* instances in the set T, $|T_{ij}|$ is the number of instances in the subset, and $|T_{ijk}|$ is the number of class-*k* instances in the subset T_{ij}.

To measure the performance of classification models, we employed accuracy, precision, recall, and f-measure defined in equations (12), (13), (14), and (15), respectively.

$$\text{accuracy} = \frac{TP}{\mathrm{TP+FP}} \tag{12}$$
$$\text{precision} = \frac{TP}{\mathrm{TP+FN}} \tag{13}$$
$$\text{recall} = \frac{TP+TN}{\mathrm{TP+TN+FP+FN}} \tag{14}$$
$$\text{f-measure} = \frac{\text{2 x recall x precision}}{\text{recall+precision}} \tag{15}$$

where TP is true positive results expressing the number of objects which are classified to be true correctly, FP is false positive errors representing the number of objects classified wrongly to be true, FN is false negative errors indicating the number of objects to be classified as false but it should have been true, and TN is true negative results indicating the number of objects to be classified as false correctly, the higher all thee four metrics, the better classification model [17].

To conduct classification a training set and a test set are needed. We generated both sets by means of a holdout method [19], in which given data was randomly partitioned into two mutually exclusive sets, a training set and a test set. Typically, two-thirds of the data are used as the training set, and remaining one-third is used as the test set. After that, the training set is used to derive a model, whose performance is estimated with the test set. In our classification experiments, each data set (i.e., data set 1 and data set 2) is split into a training set (70% of the whole data set) and test set (30% of the remaining data). Table 3 shows our parameter value configuration of each classification algorithm.

Table 3. Optimal parameter configuration for each classification model

Algorithm	**Parameter configuration**
Naïve Bayes	laplace correction =True
Neural Network	hidden layers =2, training cycles =500, learning rate =0.01, momentum =0.9,decay =False, shuffle =True, normalize =True, use local random seed =False, local random seed =2000
Support Vector Machine	kernel type = polynomial, kernel degree =3.0,kernel cache =200,C =0.0, convergence epsilon =0.01, max iterations=100000, scale =True, L pos =1.0, L neg =1.0
Linear Regression	feature selection =*M5 prime*, eliminate colinear features =True, min tolerance =0.05, use bias =True, ridge =1.0E-8
Decision Tree	criterion = accuracy, maximal depth =20,apply pruning =True, confidence =0.1, apply prepruning =True, minimal gain =0.01, minimal leaf size=2, minimal size for split =4, number of prepruning alternatives =3

3. Results and Discussion

3.1 Clustering result

As described in Section 2.3.1, data set 2 was exclusively processed by different clustering algorithms to generate 4 models with 10 clusters each. The algorithm configuration follows Table 1. The DBI values of the models are shown in Table 4 indicating that the best clustering model was derived from DBSCAN.

Table 4. Effectiveness of clustering models for data set 2

Model	**DBI**
K-mean clustering	1.833
Hierarchical clustering	1.102
Random clustering	0.982
DBSCAN	0.963

3.2 Label synthesis result

Subsequently, as mentioned in Section 2.3.2, the distance matrix in Figure 2 (i.e., EDmatrix in Table 2) is depicted in Table 5. By enlisting our proposed algorithm in Table 2, the shortest distances with respect to each unique class were identified and mapped each unique class to each unique cluster as portrayed in the X-axis label of Figure 3. The class labels became the labels of the tourist data of the clusters. In this way, we could determine which tourist clusters (i.e., data set 2) had tourism forms conforming to which tourist classes (i.e., data set 1). Cluster 1 is the sea and beach tourism form. Cluster 2 is the sport tourism form. Cluster 3 is the spa & wellness tourism form. Cluster 4 is the cultural tourism form. Cluster 5 is the ecotourism form. Cluster 6 is the natural tourism form. Cluster 7 is the MICE tourism form. Cluster 8 is the historical tourism form. Cluster 9 is the adventure tourism form. And, cluster 10 is the recreational tourism form.

Table 5. Euclidean distances between clusters and classes

Tourism-form	Cluster									
class label	1	2	3	4	5	6	7	8	9	10
Sea_Beach	2.14	25.70	23.61	19.58	12.22	11.18	13.68	17.60	20.43	13.88
Eco_tourism	3.30	24.95	22.72	19.57	10.98	11.10	14.65	15.96	21.28	12.95
Cultural	3.34	29.57	26.94	16.39	13.68	15.10	9.83	20.73	16.52	17.53
Natural	5.57	21.40	20.39	23.18	11.82	7.75	18.00	13.75	24.69	9.38
Recreational	3.30	23.89	22.40	21.12	11.99	9.71	15.49	15.98	22.21	11.97
Historical	3.34	24.34	22.39	20.88	12.12	9.68	15.08	16.75	21.87	12.74
Adventure	2.67	28.47	26.09	17.26	13.10	14.08	11.02	19.72	17.69	16.46
MICE_tourism	8.11	33.92	30.30	12.15	15.42	19.92	6.90	23.66	12.79	21.66
Sport_tourism	2.77	25.52	24.23	20.20	13.24	11.83	14.14	17.24	20.79	13.22
Health_tourism	7.20	24.14	21.57	22.62	13.14	9.54	17.02	17.77	23.56	13.79

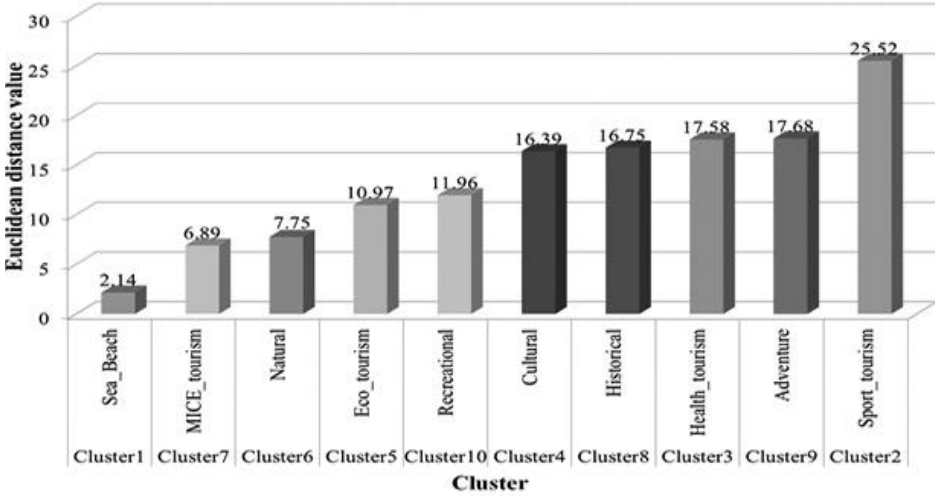


Figure 3. The value of Euclidean distance mapping of clustering and tourism form

3.3 Classification model construction

Based on Section 2.3.3, Table 6 shows the performance of the four classification models trained and tested with the combination of data set 1 and labeled data set 2. When training with data set 1 and testing with labeled data set 2, the neural network model outperformed the other models with F-measure of 94.85%. This means that the synthesized labels reasonably conformed to the labels of data set 1. On the other hand, models that were trained and tested with the same data set 1 and labeled data set 2 using the holdout technique had the best F-measure of 96.08% and 98.99%, respectively. Nevertheless, all of the models trained with labeled data set 2 poorly classified data set 1. In particular, the best of such models was SVM having F-measure of only 27.72%. The reason seems to be that the newer data set (i.e., data set 2) did not cover most patterns existing in the older data set (i.e., data set 1) whereas data set 1 covered most patterns in data set 2 (i.e., tourism behavior changed over the studied time period).

Table 6. Performance comparison of tourism-form classification models

	Data set 1 (train 70%, test 30%)				**Data set 1 (train) Data set 2 (test)**				**Data set 2 (train 70%, test 30%)**				**Data set 2 (train) Data set 1 (test)**			
Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F-measure (%)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F-measure (%)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F-measure (%)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Naïve Bayes	22.32	46.98	40.13	43.29	26.43	43.12	61.76	50.78	41.13	64.02	66.67	65.32	16.00	20.52	21.80	21.14
Neural Network	87.98	81.40	85.46	83.38	96.43	95.13	94.57	*94.85*	99.29	98.21	99.78	*98.99*	9.25	14.06	15.98	14.96
Support Vector Machine	93.56	97.53	94.68	*96.08*	61.43	94.11	32.53	48.35	89.72	93.16	92.62	92.88	18.00	34.26	23.27	*27.72*
Linear Regression	58.37	69.52	47.20	56.23	71.07	75.68	45.70	56.99	68.09	73.77	68.67	71.13	10.00	37.84	11.31	17.41
Decision Tree	75.11	75.93	64.51	69.76	65.36	74.38	39.68	51.75	95.39	95.32	97.27	96.29	5.50	25.58	11.27	15.65

Figure 4 comparatively visualizes the models that were optimal (i.e., italized F-measures in Table 6) based on each of four combinations of the data sets. A significant finding is that when classifying unseen and unlabeled tourist data (i.e., data set 2 as a test set), which is regularly collected by the Ministry of Tourism and Sports of Thailand, the synthesized labels of such data set can produce an efficient tourism-form prediction model with F-measure of 98.99%, which surpasses a model trained with data set 1 (F-measure = 94.85%). Therefore, our proposed algorithm serves as a reasonable solution to the research problem stated in Section 1.

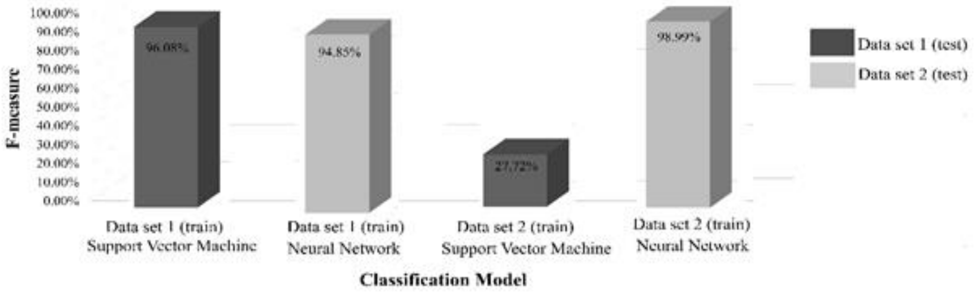


Figure 4. Comparison of effectiveness of the best classification models of each pair of the training data set and the test data set

4. Conclusions

This research proposed an algorithm for generating the tourism form labels of unlabeled Thailand-inbound-tourist data based on Euclidean distance mapping. First, we clustered the unlabeled data set with DBSCAN into 10 clusters. Then the algorithm began with computing the centroids of the clusters and the other labeled data set, containing 10 tourism form classes. Subsequently, the algorithm calculated Euclidean distances between possible pairs of the classes and the clusters to determine the clusters' labels. The data set with the synthesized labels was used to train a classification model to classify unseen and unlabeled tourist data in an effective manner with F-measure of 98.99%. A possible reason that our cluster-to-class mapping algorithm performed relatively well is the technique of exhaustive search in Figure 2 to evaluate similarity between all possible pairs of classes and clusters. Applying a neural network technique in conjunction with our algorithm yielded the best classification model for forecasting the tourism forms of inbound tourists in Thailand. Tourism entrepreneurs and organization are encouraged to apply our approach to unlabeled tourist data that is available at the Office of the Permanent Secretary of Ministry of Tourism and Sports in order to promote the growth of Thailand's tourism economy.

5. Acknowledgments

We would like to express our sincere gratitude to Office of the Permanent Secretary, Ministry of Tourism and Sports and Suvarnabhumi Airport for kind support in data acquisition.

References

[1] Economic Army of Tourism and Sport, 2020. Thailand tourism situation. *Tourism Economic Review*, 2(1), 11-15.

[2] Office of the National Economic and Social Development Council (NESDC), 2020. *NESDC ECONOMC REPORT*. [Online] Available at: https://www.nesdc.go.th/ewt_dl_link.php?nid=9895&filename=QGDP_report

[3] Department of Tourism, 2018. *The Tourism Development Strategic Plan (2018-2021)*. Bangkok: Department of Tourism.

[4] Tourism Authority of Thailand, 2020. The travel trends to know in 2020. *TAT Review*, 6(1), 17-26.

[5] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T., 2006. YALE: Rapid prototyping for complex data mining tasks. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA., 2006, 935-940.

[6] Panawong, N., Namahoot, C.S. and Brückner, M., 2014. Classification of tourism web with modified Naïve Bayes algorithm. *Advanced Materials Research*, 931-932, 1360-1364.

[7] Chatcharaporn, K., Angskun, J. and Angskun, T., 2014. Tourist attraction categorization using a latent semantic analysis and machine learning techniques. *Information*, 17, 2683-2698.

[8] Srivihok, A. and Yotsawat, W., 2014. Market segmentation of inbound business tourists to Thailand by binding of unsupervised and supervised learning techniques. *Journal of Software,* 9(5), 1334-1341.

[9] Liu, Q., Ge, Y., Li, Z., Chen, E. and Xiong, H., 2011. Personalized travel package recommendation. *IEEE 11th International Conference on Data Mining*, December, 2011, 407-416.

[10] Oender, I., 2017. Classifying multi-destination trips in Austria with big data. *Tourism Management Perspectives*, 21, 54-58.

[11] Zhu, L.N., 2017. Empirical analysis of tourism resources evaluation and promotion based on data mining neural network. *Revista de la Facultad de Ingeniería UCV,* 32(2), 385-389.

[12] Hayamin, P. and Srivihok, A., 2018. Segmentation of domestic tourist in Thailand by combining attribute weight with clustering algorithm. *Journal of Advances in Information Technology*, 9(2), 39-44.

[13] Cufoglu, A., 2014. User profiling- a short review. *International Journal of Computer Applications*, 108(3), 1-9.

[14] Rodríguez, J., Semanjski, I., Gautama, S., de Weghe, N.V. and Ochoa, D., 2018. Unsupervised hierarchical clustering approach for tourism market segmentation based on crowd sourced mobile phone data. *Sensors*, 18(9), https://doi.org./10.3390/s18092972

[15] Department of Tourism in Thailand, 2018. *Tourism Forms*. [online] Available at: https://www.tourismthailand.org

[16] RStudio Team, 2020. *RStudio: Integrated Development for R.* [online] Available at: https://www.rstudio.com

[17] RapidMiner, 2014. *RapidMiner Studio Manual*. [online] Available at: https://docs.rapid miner.com/download/RapidMiner-v6-user-manual.pdf

[18] Theeramunkong, T., 2017. *Introduction to Concepts and Techniques in Data Mining and Application to Text Mining*. 2rd ed. Bangkok: Thammasat University Press.

[19] Jane, E.M. and Raj, E.G.D.P., 2018. Comparative study on partition based clustering algorithms. *International Journal of Research in Advent Technology*, 6(9), 2398-2403.

[20] Han, J. and Kamber, M., 2006. *Data Mining: Concepts and Techniques.* 2nd ed., Illinois: University of Illinois at Urbana-Champaign.

[21] Gates, A.J. and Ahn, Y.-Y., 2017. The impact of random models on clustering similarity. *Journal of Machine Learning Research*, 18, 1-28.

[22] Sibuya, M., 1993. A random clustering process. *Annals of the Institute of Statistical Mathematics*, 45(3), 459-465.

[23] Tran, T.N., Drab, K. and Daszykowski, M., 2013. Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems,* 120, 92-96.

[24] Mary, A.V.A. and Jebarajan, T., 2014. Performance metrics of clustering algorithm. *Indian Journal of Applied Research*, 4(8), 165-167.

[25] Witten, I.H., Frank, E. and Hall, M.A., 2005. *Data Mining. Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufmann.

[26] Kumar, V., Chhabra, J.K. and Kumar, D., 2014. Performance evaluation of distance metrics in the clustering algorithms. *INFOCOMP Journal of Computer Science,* 13(1), 38-52.

[27] Mitchell, T.M, 1997. *Machine Learning*. New York: McGraw-Hill.