Prophetia: Artificial Intelligence for TravelBox® Technology

B.B. Akila Geethal[1], R. C. Gunasekara[1], M. Hassan[1], C.D. Tharindu Mathew[1], A.S. Perera[1], H. Subasinghe[2], Y. Welikala[2], L. Wimalasena[2]

[1]Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka

[2]Codegen International (Pvt.) Ltd, Sri Lanka

Abstract - **In today's fiercely competitive and dynamic market scenario, business enterprises are facing many problems due to the increasing complexity and latency of the decision making process. In addition, the amount of data to be analyzed has increased substantially. This has resulted in AI stepping in to assist decision makers to make better business decisions, reduce latency and enhance revenue opportunities. Prophetia, is a research project carried out to integrate Artificial Intelligence capabilities into TravelBox® technology – a range of solutions developed by CodeGen. IT for tour operators. This research paper discusses three main areas that were researched on for the above purpose. These are Probability Prediction – the use of Feed Forward Neural Networks for calculating the probability of selling a particular vacation package, Package Recognition – the use of Self Organizing Maps for recognizing patterns in past vacation package records, and Customer Interest Prediction – the use of association rule mining for determining the influence of customer characteristics on the vacation destination.**

Keywords — **Neural Networks; Self Organizing Maps; Data Mining**

I. Introduction

Prophetia is a research project carried out to integrate Artificial Intelligence capabilities into TravelBox® technology – a range of solutions developed by CodeGen.IT for tour operators. The ultimate goal of this research is to add, prediction and pattern recognition capabilities to the armament of features that TravelBox® provides to its customers.

The volume of data that is available for today's travel industry to base their decisions upon is overwhelming. In addition to this, the variability of the data available (for example, unknown patterns/relationships in sales data, customer buying habits, and so on) makes the analysis of this data an even more complex task.

Artificial Intelligence has seen a revival in its adaptation for commercial use in many industries such as the finance industry as a result of the advent of web-enabled infrastructure and giant strides made by the AI development community. AI has been widely adopted in such areas of risk management, compliance, and securities trading and monitoring, with an extension into customer relationship management (CRM). The use of AI technologies brings more complex data-analysis features to existing applications [1]. This has resulted in CodeGen.IT also exploring the use of AI technologies to improve the functionalities it can provide to its customers.

Prophetia concentrates on using AI technologies to solve the following three problems related with TravelBox®:

A. Probability Prediction

The problem to be addressed was that a probability that indicates the selling potential of a certain vacation package should be predicted once its parameters are known. A vacation package consists of fifteen parameters. These parameters are; booking date, departure date, Pax Count, Lead Age, Total Cost, Total Price, Nights, Product Group, Flight Departure city, Flight Destination, City of Accommodation, Flight Cabin Class, Airline, Hotel Code and Room Type. An example of the available data is shown below:

TABLE I
SAMPLE OF THE AVAILABLE DATA FOR PROBABILITY PREDICTION

TOTAL_COST	BOOKING_DATE	HOTEL_NAME
2045.81	6/12/2008 0:00 A8P8	Hilton Hotel
565.71	8/12/2008 0:00 A8P8	Queens Hotel
1225.9	9/12/2008 0:00 A8P8	Plaza Hotel

This research was carried out for 10,000 records, keeping room for future developments targeting a much larger data collection.

The expectation of the project was to come up with a mechanism that calculates the predicted probability values in an efficient and accurate way. There was a possibility of coming up with a non linear function that calculates the selling probability of vacation packages when the input parameters are given. But coming up with a regression function for the purpose is really a time consuming task due to the higher number of input parameters and variance of input data [2]. Neural Networks are general and data driven. This means that once the data is fed, the network learns from the data itself without human intervention and the variability and volume of the data does not affect the ability of the neural network to solve the problem [3]. Based on this, Neural Network approach was selected as the suitable method to solve this problem.

B. Package Recognition

The second requirement of Prophetia is to recognize "hot spots" or patterns in transaction records of sold vacation packages and recommend vacation package combinations from these identified patterns.

The pattern identification problem was broken down into a typical problem of data mining, specifically into data clustering. For the purpose of data mining, the many techniques that exist were explored through available literature. There was the choice of using classical methods for this purpose versus modern AI related methods.

The data set was the same as in the data used in *Section I - A* of this paper (15 variables, 10000 records). This variability of the data set makes the use of classical methods useless in this process [4]. The AI related techniques were considered, and the Self-Organzing Map was selected, which has the following advantages when clustering data [5] [6]:

Data mining typically deals with high-dimensional data. A record in database typically consists of a large number of items. The data do not have regular multivariate distribution and thus the traditional statistical methods have their limitations and they are not effective. SOMs work with high-dimensional data efficiently.

Kohonen's self-organizing maps provide means for visualization of multivariate data, because two clusters of similar members activate output neurons with small distance in the output layer. In other words, neurons that share a topological resemblance will be sensitive to inputs that are similar. This property has no other algorithm of cluster analysis.

C. Customer Interest prediction

The 3rd and final requirement of Prophetia deals with predicting customer's future interests according to their past data. This is done by producing a value for the influence factor of a customer's characteristics on the holiday destination. Anything above the threshold influence will be recommended to the customer as a holiday destination in his holiday destination search.

With prediction coming into play, first the research team was very much inclined to use a neural network oriented approach, either supervised or unsupervised, but given the simple nature of the requirement, associative rule mining seemed to be the most suited AI technique considering its intended application [7]. The aim of using association rule mining in this project is to find interesting and useful patterns in the data given, and come up with the association rules processing those data.

The data consists of 10,000 records with fifteen parameters. These are, Passenger Age, Marital Status, Sex, Booking Status, Option Status, Booking Price, Booking Departure Gap, Departure Month, Booking Month, Travel Together Packs, Holiday Destination, Holiday Type, Holiday Duration, Holiday_Has_Flight, Holiday_Has_Transfer, Holiday_Has_Excursion, Holiday _Has_Tour and 7_or_Less_Nights. Shown below is an extract of the data:

TABLE II
SAMPLE OF THE AVAILABLE DATA FOR CUSTOMER INTEREST PREDICTION

PASS_AGE	MARITAL_NAME	SEX
85	Married	M
76	Unknown	F
23	Single	M

For the purpose of rule mining, algorithms such as apriori, predictive apriori and Tertius were considered [8]. The decision of going through Predictive Priori was taken after going through relevant literature and discovering its advantage in using for large item sets. [9]

This paper will discuss about the research carried out by the research team and the results obtained by the research. Section II of this paper will explain the underlying theory behind the AI techniques used in this project. Section III will look at the implementation details followed by the research team in order to come up with the expected results. The results obtained after the implementation will be discussed in the Section IV. Finally Section V will give the conclusion to this paper.

II. THEORY

The main AI techniques that are used in this project are Neural Networks, Self Organizing maps and Association Rule Mining. The basic theoretical concepts of these three techniques will be explained in this section.

A. Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. ANNs also have layers of neurons with synapses connecting them. Fig. 1 illustrates the different layers of a ANN and how the synaptic connections are used to interconnect those layers.

Generally there are two methods of learning used in neural networks. These are called supervised learning and unsupervised learning. Those two learning methods are made use in different scenarios where neural networks are used to solve problems. Both learning methods have their own specific application vicinities.

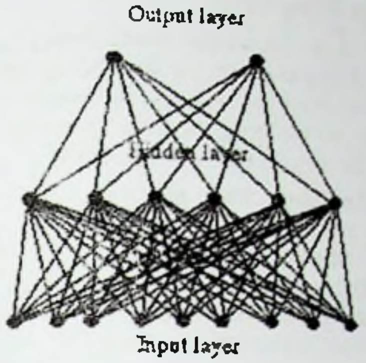

Fig 1 – interconnections between different Layers of an ANN

- Supervised learning - which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be an important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values.
- Unsupervised learning - uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. [10],[11]

B. *Self Organizing Maps*

A self-organizing map (SOM) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two dimensional) representation of the input space of the training samples. This makes SOM useful for visualizing low-dimensional views of high-dimensional data. Fig. 2 illustrates neuron structure and connections in a Self Organizing Map.

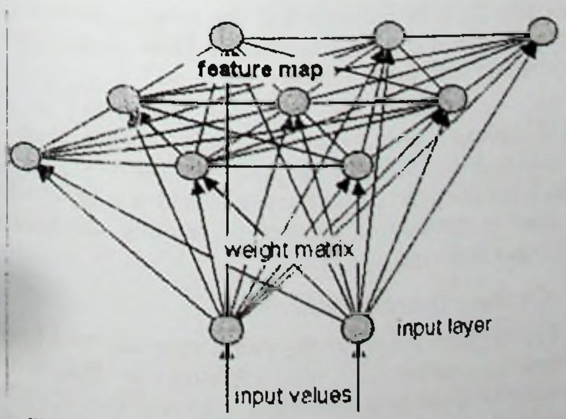

Fig 2 - Neurons and connections in a Self Organizing Map

The model was first described as an artificial neural network by the Finnish professor Teuvo Kohonen, and is sometimes called a Kohonen Feature map [12].

C. *Association Rule Mining*

Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database.

The problem is usually decomposed into two sub problems. One is to find those item sets whose occurrences exceed a predefined threshold in the database; those item sets are called frequent or large item sets. The second problem is to generate association rules from those large items sets with the constraints of minimal confidence [7].

III. Implementation

The following sections discuss the algorithms and implementation details of each of the three problems:

A. *Probability Prediction*

To train the supervised neural network of the Probability Calculation module of the system, the selling probability value of each and every data item had to be calculated first. A simple approach was taken to calculate a probability. The following formula was used:

$$Probability = \frac{no.\ of\ occurences\ of\ a\ certain\ package}{total\ no.\ of\ records} \qquad (1)$$

Therefore, if a specific parameter combination occurred six times, the probability would be (Total No. of records = 10000): 6/ 10000 = 0.0006. With this approach, every record would have an associated probability.

A feed forward artificial neural network was trained using the input parameter values as the input values and the derived probability value as the output value. Once the network was trained, when a particular parameter combination is provided, the neural network provided the probability value as the output.

Joone (Java Object Oriented Neural Engine) was used to implement the Neural Networks. It is a framework where a neural network can be easily constructed using basic components associated with ANNs, such as synapses and layers [13].

The input parameters of our data set were comprised of several data types like 'Date' values, 'String' values and Numerical values. So, before inputting the data in to the supervised neural network, we had to convert all the data types into numerical values [14]. Different algorithms were used for different data types in order to convert them into numerical values. In dealing with 'Date' data type, first, the day of the year of each and every 'Date' value was calculated and then number of occurrences of that day of the year value in the given data set was taken as the numerical value for the particular 'Date' value. In that manner the highest occurred day of the year in a certain input parameter was given the highest weight compared to other days of the year. For 'String' data type, the number of occurrences of a certain 'String' value was taken as the respective numerical value of the 'String' value.

Then, the data set which was made numerical, had to be normalized before inputting to the neural network since the data of almost all input parameters, had a huge range and deviated a lot from the respective mean values. So, all the numerical values were normalized into an acceptable range before inputting into the neural network. Equation (2), which is given below was used to normalize the input value *D* in to normalized input value *I*. In (2), I_{max} is the highest value to which the input values were normalized into and I_{min} is the lowest value to which input values were normalized into. In our implementation I_{max} was taken as +1 and I_{min} was taken as -1. So, all the normalized input values were in the range of -1 to +1. Then, highest input value and the lowest input value of each input parameter were calculated. In (2), D_{max} was taken as the highest input value of a particular input parameter and D_{min} was taken as the lowest input value of the same input parameter.

$$I = I_{min} + (I_{max}-I_{min})*((D-D_{min})/(D_{max}-D_{min})) \quad (2)$$

This method of normalization will scale input data into the appropriate range but will not increase its uniformity [15]. The normalized data for the example shown in Table I is given in the Table III below.

TABLE III
SAMPLE OF THE NORMALIZED DATA FOR CUSTOMER INTEREST PREDICTION

TOTAL_COST	BOOKING_DATE	HOTEL_NAME
-0.6920073	-0.66025641	-0.35135
-0.9240703	-0.34562982	-0.86486
-0.8205600	0 25985201	-0.67568

After coming up with the input necessary for the Neural Network, the research was extended to find out the most appropriate configuration for the Neural Network to be used in the application. There were some guidelines followed in deciding the number of hidden layers and the number of neurons in the hidden layers [16]:

- The number of hidden neurons should be in the range between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be around 2/3 of the input layer size, plus the size of the output layer.
- The number of hidden neurons should be less than twice the input layer size.

While following the above guidelines, the neural network was trained changing the number of hidden layers and number of neurons in the hidden layers. The Root Mean Squared Error (RMSE) value for each configuration of the neural network was recorded and the neural network which produced the lowest RMSE value was saved. The RMSE value for each configuration was calculated by (3), which is given below. In (3), $f(x_i)$ is the value predicted by the neural network, y_i is the actual value which is in the data set and n is the number of records in the data set.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(\left(f(x_i)-y_i\right)\right)^2}{n}} \quad (3)$$

When input parameters of a certain vacation package was given to the system to find the selling probability, the neural network which was generated earlier is retrieved and the selling probability is calculated using that neural network.

B. Package Recognition

The SOM is an unsupervised neural network that would identify and group the data being fed into clusters without the need of human intervention. Therefore, appropriate normalizing of data is paramount for the success of the results that will be obtained.

Since the same data set was used here as the probability prediction module, the same algorithm which was described in *Section III – A,* is used in normalizing the data. But since we are concentrating on clustering here, assigning numeric values for other data types was done without concerning about their number of occurrences.

Joone was used to implement the SOM with the use of its Kohonen Synapse. A 10 x 10 map was used to cluster the data. This unsupervised network is trained and then saved to preserve consistency. Then, whenever there was a need to find out new vacation packages, that saved neural network is loaded and clustering of the data is done based on that. When clustering is done, the results are displayed graphically using Java 2D. This Java 2D map displays the number of vacation packages found in the data set and the number of records in each of those vacation packages. The size of the neurons is determined by the number of records on each.

After getting the dataset clustered the next task was to analyze the data and give away probable new package types. As mentioned above, some of the clusters contain records with same product group, which implies that cluster is an existing vacation package type. So those clusters can be neglected for analyzing. So to analyze the other clusters, first, the largest clusters with different product groups are chosen. Although the clustering is done based on normalized data, for analyzing purposes actual records of those normalized data had to be fetched. This was done by extracting the record number of the particular normalized record and retrieving the actual data values corresponding to that record number from the data set. Then in each cluster, the most occurred value for each parameter is determined. Then output objects are created with these values.

So, to predict the vacation packages all the combinations from these highest values can be considered. To filter out the predicted vacation packages which are not practical in the real world, a rule set which is used in the current process of creating the vacation packages was used.

C. Customer Interest Prediction

The Predictive Apriori algorithm was followed here to determine association rules. Apriori is designed to operate on databases containing transactions. It uses a breadth-first search strategy to counting the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. [17]

The dataset consisted of sequence of transaction: T= (t_1, t_2, t_3... t_n). A transaction is a set of items ($t_i \in I$). The (absolute) support or the occurrence of x (denoted by supp(x)) is the number of transactions that are supersets of x (i.e. that contain x). The relative support is the absolute support divided by the number of transactions (i.e. n). An itemset is frequent if its support is greater or equal than a threshold value.

Association Rules were generated using the Destination of a travel package as the target parameter (ex: Colombo, Beijing) from the Predictive Apriori algorithm. The strength or the reliability of an association rule is called the confidence value. The confidence of an association rule is the percentage value that shows how frequently the rule head (right hand side of the association rule) occurs among all the groups containing the rule body (left hand side of the association rule). The confidence value indicates how reliable this rule is. Rules that exceed a threshold confidence value of 0.5 are selected and this value is presented as the influence value of the holiday destination. The WEKA java library is used to carry out the implementation.

Some rules generated in this manner are shown below:

i. Booking_Month = Jan & Hol_has_exec = No & Hol_has_tour = No ==> Holiday_destination =Beijing (Accuracy: 0.95135)

ii. Passenger_age = 60-80 & Sex = Female & Departure_month = Sep ==> Holiday_destination =Tokyo (Accuracy: 0.93782)

iii. Departure_month =Sep & Booking_Month =Jan & Hol_Has_tour = no ==> Holiday_destination = London (Accuracy: 0.89752)

Using these rules, a set of preferred destinations for each customer can be presented in the following manner:

1. When a customer requests for a package search, booking details are entered and the respective customer details are gotten from the database.
2. Using these details, the rules are searched for matching parameter values.
3. Respective destinations are gotten from the rules that match, sorted according to the confidence value and presented.

D. RESULTS

The following sections discuss the results and analysis of each of the three problems:

A. Probability Calculation

The first test was carried out to find out the suitable configuration for the network. From this test following parameters and combinations had to be determined.

- The number of Layers for the network
- Number of neurons in each layer
- Number of training cycles (epochs)

To identify the number of layers and number of neurons, the supervised network was trained with several configurations as given in fig. 3 and fig. 4 below, while taking RMSE value as the deciding factor.

Fig 3 illustrates the RMSE values obtained when the network was tested using one hidden layer and changing the number of neurons from 1 to 17. Fig. 4 illustrates the RMSE values obtained using two hidden layers changing the number of neurons of layer 2 from 1 to 11 while keeping only one neuron in layer 1. The research team recorded RMSE values changing the number of neurons in layer 1 from 1 to 17, while for each combination in layer 1 the changing number of neurons in layer 2 from 1 to 11.

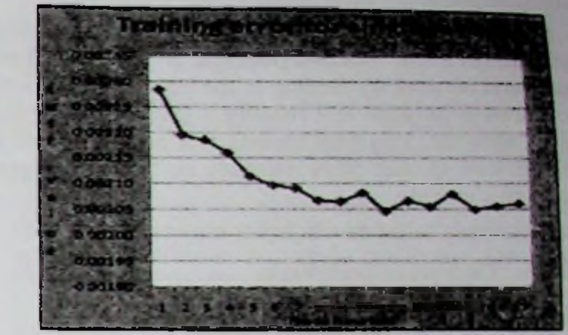

Fig 3 - RMSE values against the number of neurons used in Hidden Layer 1

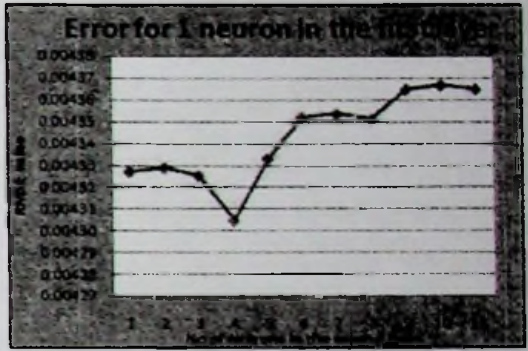

Fig 4 - RMSE values against number of neurons in Hidden Layer 2 keeping 1 neuron in Hidden Layer 1

The combinations that returned the lowest values of the training error were then subjected to validation. This was done by using 80% of the data for training and using the left 20% for validating results. This is done to determine the number of effective training cycles to be used. Fig. 5 illustrates the training error obtained for different number of cycles that the network was trained for. The number of cycles which produces the minimum validation error was taken as the number of training cycles for the network.

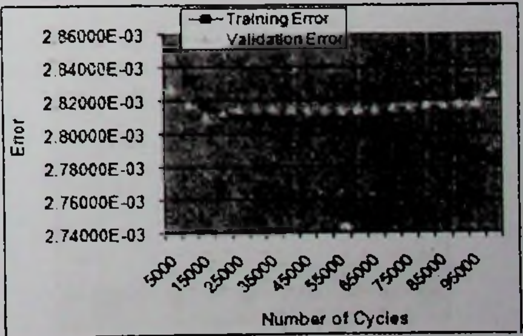

Fig 5 - Validation graph for the supervised Neural Network

Table IV illustrates the optimum results obtained by testing:

TABLE IV
OPTIMUM RESULTS FOR THE FEED-FORWARD NEURAL NETWORK

FACT	RESULT
No. of hidden layers	1
No. of neurons in the hidden layer	11
No. of training cycles	15000

The output of the application, which is the selling probability of the vacation package, is generated from the neural network by feeding in the input parameters of the vacation package and running a single cycle of the neural network. This final method was developed into a web service so that other modules in TravelBox® can access this module.

An example of the functionality is given below:

INPUT - Package Combination:

BOOKING_DATE	-	08/12/2008 0:00 A8P8
DEPARTURE_DATE	-	11/23/2008 0:00 A11P11
PAX_COUNT	-	2
LEAD_AGE	-	72
TOTAL_COST	-	2045.81
TOTAL_PRICE	-	2742
NIGHTS	-	11
PRODUCT_GROUP	-	09
FLIGHT_DEP	-	LON
FLIGHT_DES	-	TOK
ACCOM_CITY	-	TOK
FLIGHT_CABIN_CLASS	-	ECONOMY
AIRLINE	-	SA
HOTEL_CODE	-	PC8
ROOM_TYPE	-	ECONOMY

OUTPUT – Selling Probability

PROBABILITY	-	0.013983

The RMSE value which corresponds to the optimum result is 0.00205. The accuracy of the result improves as the RMSE value approaches 0 [18]. Since the obtained RMSE value is really close to 0, it can be assumed that the predicted probability values are really accurate.

B. Package Recognition

Fig.6 illustrates the clusters generated by the SOM used in Package Recognition module implemented using Joone.

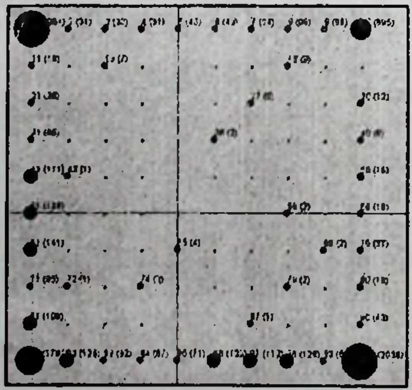

Fig 6 - Clusters generated by Joone based Clustering

The number of clusters that were formed was 47 clusters. These were close to the expected number of results as the package types that exist are 45. The other observation was that the number of clusters that were formed after training the network each time was a constant value.

A second option was also tried which is still in its research phase known as GSOM (Growing SOM) which is a dynamic SOM which grows its neurons according to the data that is being clustered. The tool was modified according to the requirement of Prophetia and used for data clustering. [19]

Fig.7 illustrates the clusters generated by GSOM based clustering.

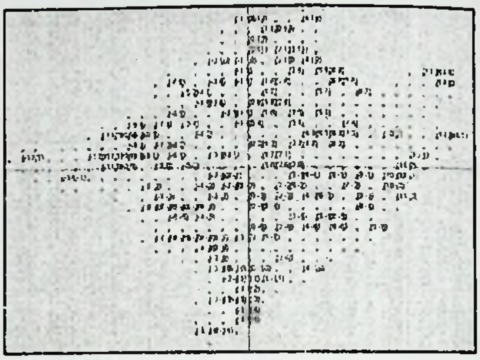

Fig 7 - Clusters generated by GSOM based clustering

There were over 100 clusters formed that made it a difficult task to analyze and get a meaningful result. The large number of clusters that were formed caused a problem in analyzing and presenting these as feasible results. The project team experimented in trying to leverage the GSOM to reduce the number of clusters by adjusting the values of GSOM parameters such as Spread Factor, Learning Rate, and Weight Update Neighborhood. Still the number of clusters that were formed was above 100 clusters. Another problem that occurred with the GSOM was that every time it was trained different numbers of clusters were formed with different compositions. These results lead to abandoning the GSOM approach for this requirement.

Therefore, the results generated through the SOM were analyzed and the data extracted by the algorithm mentioned in *Section III - B*. Finally those newly identified product types are displayed in a tabular format where it can be used for decision making. Fig. 8 shows the newly identified vacation packages presented in a tabular format.

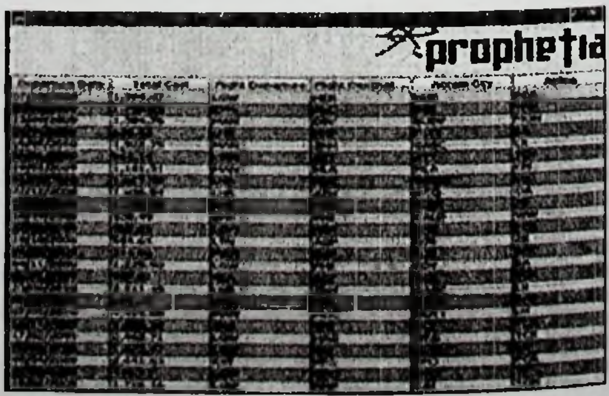

Figure 8 – Newly Identified Vacation Packages

C. Customer Interest Prediction

When customers enter their details to find out the holiday destinations which are most appropriate to them, the system uses the association rules which are generated using the Predictive Apriori algorithm to present them with the most appropriate holiday destinations. Without the loss of generality, the list of preferred destinations shown below for a customer that has the following parameters:

SEX	-	M
MARITAL_STATUS	-	MARRIED
AGE	-	34
BOOKING_DATE	-	20/04/2009
DEPARTURE_DATE	-	15/09/2009
FLIGHT	-	NO
TOUR	-	NO
EXCURTION	-	NO
7_OR_LESS_DATES	-	YES

List of Destinations

Destination	Confidence Value
Beijing	0.8751
London	0.7400
Tokyo	0.7397
Sydney	0.7390
Colombo	0.7382

E. CONCLUSION

The effort of integrating AI techniques into TravelBox® Technology has been a success with promising results. With the use of supervised neural networks, Self organized maps and association rule mining, Prophetia has achieved results that would have been hardly achieved with classical statistical methods. Further improvements should be made to these methods to further improve results before adopting it in commercial use but the steps that have been taken in this project in integrating AI in TravelBox® Technology will pave the way for future innovations in the range of products.

F. CONFIDENTIALITY

The Contents of this paper represent intellectual effort from Codegen International (Pvt) Ltd, and are confidential and bound by copyright laws. No part or ideas in this paper shall be reproduced or used without the written consent from Codegen International (Pvt) Ltd.

REFERENCES

[1] Xindong Wu, "Data Mining: Artificial Intelligence in Data Analysis", *Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference*, Sept. 2004

[2] H. White: "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns," *Proceedings of the Second Annual IEEE Conference on Neural Networks*, II: pp. 451-458, 1988.

[3] F. J. Delgado, D. Santín and A. Valiño, "The measurement of technical efficiency: a neural network approach", *Applied Economics*", 2004, 36(6),pp. 627-635

[4] V. R. Kumar and N. Mani, "The application of artificial intelligence techniques for intelligent control of dynamical physical systems", *International Journal of Adaptive Control and Signal Processing Volume 8, Issue 4*, pp. 379 – 392, Mar 2007.

[5] X. Zhang and Y. Li, "Self-organizing map as a new method for clustering and data analysis", *Neural Networks, 1993. IJCNN '93-Nagoya. Proceedings of 1993 International Joint Conference on*, Vol. 3 1993, pp. 2448-2451 vol.3.

[6] M. Craven and J. Shavlik, "Using Neural Networks for Data Mining", *Future Generation Computer Systems*, 13, pp. 211-229, 1997

[7] R. Agrawal, T. Imielinski, and A. Swami. *Mining assosiation rules between sets of items in large databases.* In Proceedings of the 1993 International Conference on Management of Data (SIGMOD 93), pp. 207--216, May 1993

[8] B. Tunc and H. Dag, "Generating classification association rules with modified Apriori algorithm", *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Databases*, pp 384-387, in 2006

[9] J. Hipp, U. Güntzer and G. Nakhaeizadeh, "Algorithms for association rule mining — a general survey and comparison", *ACM SIGKDD Explorations Newsletter, Volume 2 , Issue 1*, pp. 58-64, June 2000

[10]T. D Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network", *Neural Networks*, 2:pp 459–473, 1989

[11]S. Haykin, *Neural Networks: A Comprehensive Foundation*, Second Edition, Prentice Hall, 1999

[12]J. Fröhlich, "Kohenan Feature Map", *Neural Networks with Java 2004 Edition*, [Online]. Available: http://www.nnwj.de/. [Accessed: Feb. 12, 2009]

[13] P. Marrone, "The Complete Guide: All you need to know about Joone", *Joone: Java Object oriented Neural Engine, 17 January 2007*.[Online].Available:http://biznetnetworks.dl. sourceforge.net/ sourceforge/joone/JooneCompleteGuide.pdf.[Accessed:13.06.2008]

[14]I. Klevecka and J. Lelis, "Pre-Processing of Input Data of Neural Networks: The Case of Forecasting Telecommunication Network Traffic", *Telektronikk 3/4.2008*, pp 168-178, 2008

[15]L.Mendelsohn,"Preprocessing Data for Neural Networks", *Technical Analysis of Stocks and Commodities*,1993 [Online], Available: http://www.tradertech.com/preprocessing_data.asp [Accessed: 21.06.2008]

[16] J.Heaton, "*Introduction to Neural Networks for Java*, 2nd Edition", Heaton Research, October 2008

[17]S. Kotsiantis and D. Kanellopoulos, "*Introduction, Association Rules Mining: A Recent Overview*", Educational Software Development Laboratory, Department of Mathematics, University of Patras, Greece, 2006.

[18]S. Mandal, "*Prediction of Tides using back-Propergation Neural Networks*", International Conference in Ocean Engineering, December 2001

[19] D.Alahakoon, S. Halgamuge and B. Sirinivasan, "*A Self Growing Cluster Development Approach to Data Mining*", Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San Diego, USA, pp 2901-2906, 1998