

**THE REPUBLIC OF TURKEY  
BAHÇEŞEHİR UNIVERSITY**

**DEVELOPING INTELLIGENT TRIP  
RECOMMENDER SYSTEM**

**Ph.D. Thesis**

**TAMER UÇAR**

**İSTANBUL, 2016**



**THE REPUBLIC OF TURKEY  
BAHÇEŞEHİR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
COMPUTER ENGINEERING PH.D. PROGRAM**

**DEVELOPING INTELLIGENT TRIP  
RECOMMENDER SYSTEM**

**Ph.D. Thesis**

**TAMER UÇAR**

**Supervisor: PROF. DR. ADEM KARAHOCA**

**İSTANBUL, 2016**

**THE REPUBLIC OF TURKEY  
BAHÇEŞEHİR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
COMPUTER ENGINEERING PH.D. PROGRAM**

Name of the thesis: Developing Intelligent Trip Recommender System

Name/Last Name of the Student: Tamer UÇAR

Date of the Defense of Thesis: 07.12.2016

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Prof. Dr. Nafiz ARICA  
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Doctor of Philosophy.

Asst. Prof. Dr. Tarkan AYDIN  
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Doctor of Philosophy.

Examining Committee Members

Signature

Thesis Supervisor  
Prof. Dr. Adem KARAHOCA

Member  
Assoc. Prof. Dr. M. Alper TUNGA

Member  
Asst. Prof. Dr. C. Okan ŞAKAR

Member  
Prof. Dr. Salih OFLUOĞLU

Member  
Asst. Prof. Dr. Oğuz KARAN

## ACKNOWLEDGEMENTS

I would like to thank all the people who have helped and inspired me during my study.

I offer my sincerest gratitude to my supervisor, Prof. Dr. Adem KARAHOCA, who has supported me with his experience and knowledge. It would be impossible to complete this study without his motivation and guidance.

I would like to thank to my wife, Elif UÇAR, for her endless love and encouragement in every part of my life.

Finally, I would like to show my gratitude to my mother, father, and brother for their support throughout my life.



## ABSTRACT

### DEVELOPING INTELLIGENT TRIP RECOMMENDER SYSTEM

Tamer UÇAR

Computer Engineering Ph.D. Program

Thesis Supervisor: Prof. Dr. Adem KARAHOCA

December 2016, 79 pages

Internet has a wide usage in almost every field. People are continuously looking and searching for information through web based platforms. Tourism domain is one of these fields. People often plan their trips by searching travel destinations and airlines for a desired date interval. It becomes more and more difficult to find relevant information within massive search results. Narrowing down this information in an accurate way is a challenging task. Recommender systems are proposed to address this problem.

A recommender system generates suggestions for providing relevant information to a user about an item or a service. Such systems analyze users' preferences and previous usage habits to generate recommendations. Recommender systems use data mining methods for extracting valuable knowledge from data sets.

This study proposes an implementation of an expert system framework which can accurately classify users and generate recommendations for travel locations. Presented implementation also suggests an airline and a travel duration for recommended location. Proposed approach evaluates several clustering and classification strategies for generating most accurate recommendations.

**Keywords:** Recommender Systems, Data Mining, Trip Planning

## ÖZET

### AKILLI SEYAHAT ÖNERİ SİSTEMİ GELİŞTİRİLMESİ

Tamer UÇAR

Bilgisayar Mühendisliği Doktora Programı

Tez Danışmanı: Prof. Dr. Adem KARAHOCA

Aralık 2016, 79 sayfa

İnternet neredeyse her alanda geniş bir kullanıma sahiptir. İnsanlar web tabanlı platformlar aracılığıyla istedikleri bilgiye erişebilmek için sıklıkla web aramaları gerçekleştirmektedirler. Turizm alanı da bu arama yapılan alanlardan biridir. İnsanlar sıklıkla gezi planlarını seyahat yerleri ve bu yerlere ulaşmakta kullanacakları havayollarını göz önünde bulundurarak yapmaktadırlar. Bu ölçütlere uygun arama sonuçlarını içeren veri kümesi büyüdükçe sonuçlar içerisinde ilgili ve anlamlı bilgileri bulmak zorlaşmaktadır. Bu bilgileri doğru ve tutarlı bir şekilde daraltabilmek önemli bir problemdir. Öneri sistemleri bu problemi çözebilmek için sunulan bir yaklaşımdır.

Bir öneri sistemi, belirli bir kullanıcıya bir öge veya hizmet hakkında alakalı bilgiler içeren tavsiyeler / öneriler üretir. Bu sistemler öneri üretmek için kullanıcıların tercihlerini ve önceki kullanım alışkanlıklarını analiz ederler. Öneri sistemleri, veri kümelerinden değerli bilgileri elde edebilmek için veri madenciliği yöntemlerini kullanmaktadır.

Bu çalışma, kullanıcıları doğru bir şekilde sınıflandırıp kullanıcılar için seyahat yerlerine ilişkin öneriler üretebilecek bir uzman sistem uygulanması içermektedir. Sunulan uygulamada kullanıcıya seyahat yerinin yanı sıra tahmini gezi süresi ve önerilen yere ulaşım için kullanılacak havayolu önerilmektedir. Sunulan bu çalışmada en doğru önerilerin üretilebilmesi için birçok kümeleme ve sınıflandırma stratejileri değerlendirilir.

**Anahtar Kelimeler:** Öneri Sistemleri, Veri Madenciliği, Seyahat Planlama

## CONTENTS

<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>2. BACKGROUND .....</b>	<b>3</b>
<b>3. MATERIAL AND METHODS.....</b>	<b>11</b>
<b>3.1 DATA GATHERING AND PROCESSING .....</b>	<b>11</b>
<b>3.2 METHODS.....</b>	<b>14</b>
<b>3.2.1 X-means (XM) Clustering .....</b>	<b>14</b>
<b>3.2.2 Fuzzy C-means (FCM) Clustering.....</b>	<b>15</b>
<b>3.2.3 Adaptive Neuro Fuzzy Inference System (ANFIS) .....</b>	<b>17</b>
<b>3.2.4 Radial Basis Function Networks (RBFN) .....</b>	<b>19</b>
<b>3.2.5 Naïve Bayes .....</b>	<b>21</b>
<b>3.2.6 C4.5 / J48.....</b>	<b>21</b>
<b>3.3 COMPARING DATA MINING TECHNIQUES.....</b>	<b>23</b>
<b>3.4 IMPLEMENTING TRIP RECOMMENDER.....</b>	<b>24</b>
<b>3.4.1 Data File Uploader Module .....</b>	<b>25</b>
<b>3.4.2 Cluster Analysis Module.....</b>	<b>26</b>
<b>3.4.3 Prediction Model Builder Module .....</b>	<b>27</b>
<b>3.4.4 Recommendation Generator Module .....</b>	<b>29</b>
<b>4. FINDINGS .....</b>	<b>31</b>
<b>4.1 BENCHMARKING CLASSIFICATION METHODS.....</b>	<b>31</b>
<b>5. DISCUSSION .....</b>	<b>41</b>
<b>6. CONCLUSION.....</b>	<b>50</b>
<b>REFERENCES .....</b>	<b>52</b>
<b>APPENDICES .....</b>	<b>59</b>
<b>APPENDIX 1: Database Schema of Travel Portal .....</b>	<b>60</b>



**APPENDIX 2: Flight-Hotel Dataset Attribute Values .....65**  
**APPENDIX 3: Confusion Matrices for Classification Algorithms.....70**



## TABLES

Table 3.1: Initial data set attributes .....	11
Table 3.2: Region codes .....	12
Table 3.3: Flight cost groups.....	13
Table 3.4: Hotel cost groups .....	13
Table 3.5: Final data set attributes .....	14
Table 4.1: Benchmarking methods .....	31
Table 4.2: Confusion matrix for ANFIS XM 8.....	33
Table 4.3: Confusion matrix for J48 XM 5.....	33
Table 4.4: Confusion matrix for RBFN FCM 8.....	33

## FIGURES

Figure 3.1: First order Sugeno fuzzy model.....	18
Figure 3.2: ANFIS architecture.....	18
Figure 3.3: Binary confusion matrix .....	23
Figure 3.4: Recommender system framework .....	25
Figure 3.5: LoadFile managed bean class .....	26
Figure 3.6: CluserAnalysis managed bean class .....	27
Figure 3.7: ClusterCollection container class .....	28
Figure 3.8: MiningModel, MiningModelCollection and BuildModel classes.....	28
Figure 3.9: FlightHotelRecord, PredictedFlightHotelRecord, PredictInput and PredictionResult classes .....	30
Figure 5.1: Trip recommender initial page .....	43
Figure 5.2: Trip recommender data loader / clusterer.....	44
Figure 5.3: Trip recommender model builder (Algorithm selection) .....	45
Figure 5.4: Trip recommender model builder (Progress) .....	46
Figure 5.5: Trip recommender model builder (Benchmark list) .....	47
Figure 5.6: Trip recommender input cluster prediction .....	48
Figure 5.7: Trip recommender output .....	49

## ABBREVIATIONS

ANFIS	:	Adaptive Neuro Fuzzy Inference System
ARFF	:	Attribute-Relation File Format
FN	:	False Negative
FP	:	False Positive
FCM	:	Fuzzy C-means
RBFN	:	Radial Basis Function Networks
RMSE	:	Root Mean Squared Error
TN	:	True Negative
TNR	:	True Negative Rate
TP	:	True Positive
TPR	:	True Positive Rate
XM	:	X-means

## 1. INTRODUCTION

Today, internet is a very common platform for getting information about various topics. Advancing technology enabled storing vast amount of data on digital platforms in an easy and simple way. Binding internet with these data sources emerged necessity of search engines. People are continuously looking and searching for information through these internet based search tools. Although it is very easy to reach a specific topic, finding the best item within a result set is not a very easy task if user is facing a very large amount of data set. For handling such problems, data mining and filtering approaches gained popularity and interest by various researchers.

Data mining is a technique for extracting knowledge from large data sets. It is the combination of statistical and mathematical methods for processing raw data to discover knowledge. Today, data mining methods are used in various topics such as filtering systems, risk analysis management, fraud detection, medicine, e-commerce and many more.

One of this popular data mining related research area is information filtering systems. Recommender systems are major applications for this area. They are used for filtering and providing relevant information about a person's search on a specific topic. To provide accurate filtering results, many different data mining methods can be used (Lü et al. 2012).

Basically, a recommender system tries to generate a rating value of an item for a target user. And according to these rating values, system tries to propose an item or many items to target user. For different users, every item gets associated with different rating scores. These rating scores can be computed in various ways. In many recommender systems target user's profile and previous behaviors are used for generating rating scores for items.

Recommender systems are being used in almost every search related area. Tourism domain is one of these sectors. In tourism, most of the recommender system applications involve proposing travel destinations, trip and activity recommendations and hotel suggestions for a destination within a given set of user defined constraints. These

constraints can be budget limits, time intervals, interests, desired locations or similar necessities. After retrieving such data, a recommender system analyzes user input and proposes a relevant output. To generate accurate predictions, many approaches are tested by different researchers. In general, most of these approaches are based on acquiring a set of parameters which can be used as constraints for the recommendation system (Borras et al. 2014).

This study proposes an implementation of an expert system which can accurately classify users and make predictions for these user classifications. Proposed approach predicts clusters for system users and according to these predicted clusters, travel locations, durations and airlines are recommended to individual users. Besides this, travel agencies can use the recommendation output of this system while planning trip campaigns for similar users.

There are many recommender systems available on tourism domain. Main purpose of this proposed approach is increasing cluster prediction correctness and developing a recommender system which can adapt itself to different data sets. To achieve this goal, proposed system tests and compares several clustering and classification strategies. Then the best fit is picked for generating recommendations based on user clusters. The following steps summarize details of the proposed approach: (1) Discretizing initial travel data set. (2) Discovering user clusters to build prediction models. (3) Training and testing data mining models using discovered clusters. (4) Finding the best clustering and data mining method pair. (5) Predicting target user's cluster using the best clustering and data mining method pair. (6) Recommending services suitable for target user's cluster.

The remainder of this document is organized as follows: Section 2 includes reviews of recent studies about recommender systems, Section 3 describes materials and methods used in the proposed approach, Section 4 presents comparison results of the proposed approach, Section 5 mentions the developed recommender system's output and Section 6 contains the conclusion and possible future plans of this study.

## 2. BACKGROUND

There are many remarkable studies conducted for generating recommender systems in many different areas. This section focuses on recent studies about such systems in a chronological way.

Mild and Reutterer (2003) presented a collaborative filtering approach for cases where only binary customer information is available. Proposed approach is applicable on binary values like choice / non-choice of items basket data which is common in basket data.

Wang et al. (2004) proposed a personalized recommender system for cosmetic business. Researchers constructed a recommender system using content-based, collaborative filtering and data mining approaches. A scoring approach for assessing customers' interest on products was also proposed by researchers.

In another study, online recommender systems were reviewed from a different view point. Researchers aimed to show that language is an important factor for developing effective recommender systems especially for online restaurant services. Proposed study provided suggestions for such systems (Xiang et al. 2007).

Diez et al. (2008) introduced an approach for discovering clusters of people who share similar preferences. To build these clusters, ranking functions are derived from individuals' preference judgement sets. Researchers aimed to use these clusters to map people to different market segments.

Castillo et al. (2008) developed a mobile software tool for planning tourist visits. System generates trip plans based on preferences from similar users. Proposed approach contains an ontology model to support this structure. Recommendations were generated to target users which can be interesting and achievable.

In other study, researchers developed a system which generates personalized recommendations of touristic attractions. Proposed system integrates heterogeneous online travel information using a tourism ontology. Travel behavior of the target user and similar users were analyzed to generate recommendations. Bayesian network technique

and the analytic hierarchy process method used for recommendation engine (Huang and Bian 2009).

An expert travel agent was developed for assisting tourists by suggesting holidays and tours. Proposed method employs a hybrid approach containing both content-based and collaborative filtering methods. Demographic data was also used in recommendation system. Authors emphasized that the choice of this hybrid approach was made to cover shortcomings of each of the individual recommendation methods (Schiaffino and Amandi 2009).

A group recommendation system was developed by researchers which aims to increase recommendation satisfaction for every individual in a group. To achieve this goal, researchers used a collaborative filtering approach to generate initial recommendations and then irrelevant items were removed according to each group member's preferences (Kim et al. 2010).

Bouhana et al. (2010) proposed of personalized itinerary search approach in transport field. Researchers presented two methods for searching personalized information. First method covers calculating item relevancy degree values in the whole set of itineraries. Second method covers similarity calculation between user profile and itinerary.

Kabassi (2010) focused on presenting guidelines for building recommendation systems for tourism area. Most common methods and approaches were analyzed and problems were discussed in detail. Researcher highlights user privacy as a great challenge which must be guaranteed in a recommender system in terms of information sharing.

Bobadilla et al. (2011a) developed a framework for collaborative filtering recommender systems. Proposed study provides measurements for evaluating recommendation novelty, equations for collaborative filtering approach and a framework which employs the mentioned measurements and equations. In the same year, researchers also presented another study for improving collaborative filtering recommender system performance using genetic algorithms by computing a similarity between users (Bobadilla et al. 2011b).



Another recommender engine study covers trip recommendations for both individuals and tourist groups. Group recommendation mechanism aggregates and intersects individual recommendations which were made for every member in a given group. Recommendation engine employs both demographic and content-based filtering methods (Garcia et al. 2011).

Montejo-Ráez et al. (2011) presented a web based planner which is built for scheduling tasks in tourism. System allows users to create a list of activities and it uses the listed activities for creating recommendations. User preferences were also considered by the recommendation engine. Researchers focused on a simple scheduling tool without transportation recommendations.

In another research, a semantic hotel recommender system was developed. To generate recommendations, hotel ontology was combined with a fuzzy logic approach. To involve customer experience, system contains a feedback mechanism which allows users to rate the generated recommendations. In order to generate more accurate recommendations, these ratings were used for updating fuzzy rules (García-Crespo et al. 2011).

Abbaspour and Samadzadegan (2011) studied on personal tour planning and scheduling problem in metropolises. Researchers used genetic algorithm approach for finding the optimum tour. Proposed method was tested by using a dataset to plan 400 tours with different attributes. According to experimental results, proposed method can discover optimum tours based on given constraints.

Noguera et al. (2012) proposed a mobile recommender system for tourism. Proposed system employs a hybrid recommendation engine containing collaborative and knowledge filtering. Users get recommendations based on their locations through a 3D GIS architecture.

Tsai and Chung (2012) presented a route recommender for theme parks. Researchers built an RFID based system for collecting visitor behaviors. Collected information is clustered based on visiting time and visiting sequence attributes. Route recommender generates output based on visitor's personal preferences and visiting behavior.

Another travel planning recommender engine was proposed by Wang and Yang (2012) which is formed on case based reasoning approach. Researchers supported the system by adding a genetic algorithm to reduce the cost in case evaluation phase.

Bobadilla et al. (2012) presented a study on collaborative filtering approach by adding item significance values. Proposed system employs calculation of k-neighbors including item and user significance ratings. Recommender generates output values based on similarities which were computed using significance ratings.

Another decision support system for tourist attractions was implemented by combining Engel–Blackwell–Miniard model and Bayesian network approaches. Data which was published by the Tourism Bureau of Taiwan was used while building the proposed recommendation system. Generated recommendations were displayed on Google Maps to provide more detailed information for tourists (Fang-Ming et al. 2012).

Hadjali et al. (2012) studied on modeling and handling route planning queries based on fuzzy set theory. Researchers proposed the outline of an SQL-like language for querying interface. A query evaluation approach is also proposed within the framework.

Garica et al. (2012) proposed a recommender system which can generate recommendations for a group of users. Based on degree of interests, system compiles a group preference model by processing preferences of individual users. Individuals' preferences are obtained by using demographic, collaborative, content-based and general-likes filtering approaches.

Another recommendation system was proposed by Batet et al. (2012). Authors implemented an agent-based recommender for cultural and leisure activities at a given location. Proposed system employs a content based and collaborative filtering approach for generating recommendations.

Parvaneh et al. (2012) studied on understanding behaviors of travelers while they make decisions about picking a route, departure time or any other preference about a travel. Researchers presented a Bayesian Belief framework to identify the connection between travel information and cognitive learning process of individuals.

A hybrid recommendation system which contains both content-based and collaborative filtering methods was built to propose better personalization for recommender systems in tourism. Proposed system was implemented using association based classification approach. Concepts from association and classification were combined to involve association rules in a prediction context (Lucas et al. 2013).

Another mobile tourism recommendation system was implemented using a location based collaborative filtering method. Proposed system generates recommendations by considering other tourists' ratings on their visited attractions. Users exchange their rating through a mobile peer to peer connection. Three data exchange methods were proposed for effectively exchanging ratings about visited attractions (Yang and Hwang 2013).

Moreno et al. (2013) developed a web based recommender for tourists using an ontology based approach. Available activities were classified according to an ontology. Proposed recommender uses demographic information, travel data, user behaviors, user interests and user similarity based opinion matching.

Moussa et al. (2013) proposed a multi-criteria decision making method for personalizing traveler's information in public transport domain. Authors focused on building a transport recommender based on users' profile data. Proposed approach calculates performance ratings and compares different solutions based on rankings.

Neves et al. (2014) presented an agent based architecture for generating event recommendations. Authors used an ontology model for defining domain knowledge. Spreading algorithm was used for discovering user patterns for building recommendations.

Borras et al. (2014) made a survey about recommender systems which were applied on tourism domain. Authors analyzed interfaces, algorithms and functionalities of such systems.

Based on a similar perspective, a review study about mobile recommender system implementations in tourism domain was proposed by Gavalas et al. (2014). Authors analyzed currently implemented recommender systems by their proposed services and they stated possible research trends for such systems.

Another study was proposed by Umanets et al. (2014). Researchers presented a mobile and web based recommender system for tourists which can integrate with social networks. Proposed approach employs collaborative filtering method. System generates recommendations for unvisited touristic locations by considering other users' ratings.

Aksenov et al. (2014) studied on a recommender system to provide smart routing capabilities for tourists. Authors presented a three level system which includes tour programming step, tour scheduling step and route planning step. Route planning step considers user preferences and interest scores for organizing possible point of interests.

Hawalah and Fasli (2014) proposed a context aware personalization method which can generate ontological user profiles based on user's preferences and interests. System uses these generated profiles for proposing contextual recommendations for providing personalization in web.

A different research group proposed a travel schedule planning algorithm which generates customized recommendations based on user requirements. With a user-adapted interface, users can make changes on recommendation results and the provided feedback mechanism improves system's accuracy for later recommendations (Chiang and Huang 2015).

García-Magariño (2015) presented an agent based tour simulator which can estimate the number of tourists who can sign up for each route which were defined for simulation task. Simulation works with different tourist types and route definitions. Proposed system aids experts for detecting overcrowded or non-popular routes.

Han and Lee (2015) implemented a recommendation system which analyzes geo-tagged social media to recommend landmarks for customized travel planning. System obtains trip's spatial and temporal properties and using these properties, it computes the significance of landmarks. Specific landmark clusters are generated for similar themes and these clusters are recommended to system users.

Vukovic and Jevtic (2015) studied on a location predictor for mobile users. Researchers focused on predicting users' locations based on their habits and past movements. Movement data was fetched from mobile devices. Proposed system intends to predict a

user's possible future location and based on this prediction, location information can be shared to allowed set of services or applications.

Tsai and Lai (2015) developed a route recommender system for theme parks based on visitors' behaviors. Researchers aimed to provide better visiting experiences for park visitors by proposing personalized route sequences. Recommendation system considers visitors' intended visiting time, favorite theme park facilities and regions.

Majid et al. (2015) proposed a tourist location recommender based on geo-tagged photos. System analyzes geo-tagged photos which can be obtained from social media sites. After discovering photos' context (date, time, weather conditions), proposed system employs a location profiling method to associate locations with system users. System uses location profiles to generate recommendations.

Another study was implemented from a similar perspective. Researchers presented methods for providing recommendations based on photo sharing and demographic information of system users. Bayesian Learning model was proposed for performing location predictions according to user preferences (Subramaniaswamy et al. 2015). Ragnathan et al. (2015) proposed an architecture for a trip planner and tourism information system. Although there is no implementation for it, proposed architecture suggests providing information about transport and available tourist locations / facilities in a city.

García-Palomares et al. (2015) presented a method for identifying tourist attractions in cities. Authors analyzed spatial distribution patterns of geo-tagged photos which were taken by residents and tourists. Based on the obtained results, photographs taken by tourists and local residents showed differences which can be used to reveal tourist attractions in cities.

TripBuilder is another framework for providing personalized recommendations for tourist locations in a given city. Proposed approach uses geo-tagged photos for discovering a user's trip behavior. System tries to match possible point of interests with a user by considering user behavior, user preferences and visiting time constraints (Brilhante et al. 2015).

Saleh et al. (2015) proposed a hybrid recommender which employs a neuro-fuzzy system, K nearest neighbor and Naïve Bayes classifiers for recommending text documents for a given domain. Presented approach tries to enhance the overall performance of recommender systems.

Sun et al. (2015) studied on recommending popular landmarks and travel routes between landmarks based on tourist preferences. Recommendation system uses geo-tagged photos from Flickr to discover popular tourist locations and possible travel destinations between them. Maximum popularity and minimum trip length is taken into account while performing recommendations.

Varfolomeyev et al. (2015) proposed a recommender system for historical tourism. Researchers proposed an approach based on smart space architecture which includes ontology for inferring information. Presented approach contains multi-agent methods.

Another study which focused on social media based recommendation includes a method for city travel recommendation system. Researchers applied principals from both content based and collaborative filtering techniques. User preferences were mined from community-contributed geo-tagged photos archive. User similarities were taken into account for improving accuracy of the proposed model (Xu et al. 2015).

Socharoentum and Karimi (2016) presented a multi-modal transportation route recommender for pedestrians. Proposed system recommends walking routes for pedestrians by considering conditions including traveler's physical capabilities, travel location and travel time.

Colomo-Palacios et al. (2016) studied on a context-aware mobile recommendation system for loyalty in tourism. Researchers presented a method for analyzing the collected data from tourist visits which is used for generating recommendations. Proposed system employs a domain ontology, opinion mining engine, recommender system and mobile interface.

### 3. MATERIAL AND METHODS

This section provides details about data gathering, data pre-processing and recommender requirements specifications. Algorithms which were used in this study are also explained.

#### 3.1 DATA GATHERING AND PROCESSING

Initial data set of customer flights and hotel bookings was obtained from an existing travel platform which is integrated with different data sources. Appendix 1 contains the database structure of this travel platform. Retrieved data was extracted from a nested XML structure. A total of 26,886 flight records and 4,367 hotel bookings were collected for processing. After obtaining tabular formatted data, all of the identity columns were removed from the data set. As a result of this data processing, 14 attributes were retrieved. Table 3.1 lists these attributes.

**Table 3.1: Initial data set attributes**

Attribute	Description
Gender	Passenger's gender.
Departure date	Starting date of trip.
Departure location	Location which the passenger is leaving from.
Arrival location	Location which the passenger is arriving to.
Departure airline	Airline company for departure flight.
Departure flight class	Ticket class for departure flight.
Returning date	Ending date of trip.
Returning location (from)	Location which the passenger is returning from.
Returning location (to)	Location which the passenger is returning to.
Returning airline	Airline company for returning flight.
Returning flight class	Ticket class for returning flight.
Flight cost	Flight's cost.
Days in hotel	Number of days stayed in hotel.
Hotel cost	Hotel's cost.

“Departure location” attribute was removed from the initial data set since “departure location” and “returning location (to)” attributes were containing the same set of values. “Arrival location” and “returning location (from/to)” values were discretized according to regions. Table 3.2 lists regions by their numeric codes.

**Table 3.2: Region codes**

Code	Description
1	Northern Europe
2	Southern Europe
3	Eastern Europe
4	Western Europe
5	Central Europe
6	Balkans
7	Middle East
8	Northern Asia
9	Southern Asia
10	Eastern Asia
11	Western Asia
12	Central Asia
13	Africa
14	America
15	Australia
16	(Turkey) Marmara Region
17	(Turkey) Black Sea Region
18	(Turkey) Central Anatolia Region
19	(Turkey) Southeastern Anatolia Region
20	(Turkey) Aegean Region
21	(Turkey) Eastern Anatolia Region
22	(Turkey) Mediterranean Region



Based on users flight and hotel expenses, cost attributes were discretized into 6 groups. Table 3.3 and Table 3.4 lists these cost attribute groups.

**Table 3.3: Flight cost groups**

Code	Description
1	< 200
2	201 – 400
3	401 – 700
4	701 – 1400
5	1401 – 3000
6	4000 +

**Table 3.4: Hotel cost groups**

Code	Description
1	< 350
2	351 – 700
3	701 – 1000
4	1001 – 1500
5	1501 – 2500
6	2500 +

Trip season and trip duration values were derived using “departure date” and “returning date” attributes. “Days in hotel” attribute was removed from data set because for each record, trip duration and “days in hotel” values were pointing to same set of values. Ticket class attributes were also removed from the data set since 97 percent of records were sharing the same ticket class type. After deriving these two new attributes, removing redundant attributes and applying discretization on initial data set, 10 attributes were obtained for data processing. Table 3.5 lists the final state of these attributes. Appendix 2 lists full discretization values of data set.

**Table 3.5: Final data set attributes**

<b>Attribute</b>	<b>Description</b>
Gender	Passenger's gender.
Trip duration	Duration of trip in days.
Season	Season the trip took place.
Arrival location	Location which the passenger is arriving to.
Departure airline	Airline company for departure flight.
Returning location (from)	Location which the passenger is returning from.
Returning location (to)	Location which the passenger is returning to.
Returning airline	Airline company for returning flight.
Flight cost	Flight's cost.
Hotel cost	Hotel's cost.

Final data set was used to discover clusters. After obtaining clusters for each record, 66 percent of data was used for training and the remaining 34 percent was used for testing the prediction models.

## **3.2 METHODS**

This section describes the methods which were used in this study.

### **3.2.1 X-means (XM) Clustering**

K-means clustering algorithm is a simple but popular approach for finding clusters in a given data set. But there are some important shortcomings for this method such as the necessity of providing the number of clusters and random located initial cluster centers. Pelleg and Moore (2000) proposed X-means clustering method to overcome these drawbacks. It works as extending K-means with efficient estimation of the number of clusters.

Original K-means algorithm groups data into given number of subsets (clusters). The number of subsets is the K number which is provided to algorithm. Initially, algorithm picks random centroids for each cluster. Then it starts an iteration for finding the best centroid locations for clusters. In each step, the following operations are performed:

- i. For each point  $x$  in data set, find and associate the centroid which is closest to it by measuring distances.
- ii. Re-calculate centroid locations for each cluster by computing center of mass.

X-means algorithm extends the original K-means implementation by estimating K value for a given data set. Estimation of K can be accomplished using a model selection criterion.

Proposed X-means algorithm performs the following operations:

- i. Run conventional K-means to convergence.
- ii. Find new centroid locations by splitting centroids into two.
- iii. If  $K > K_{\max}$  stop and record the best scoring model.  
Else go to step 1.

After running K-means algorithm which is stated in first step, X-means needs to decide how to split centroids which is required in second step. This can be performed by re-running K-means for each cluster with  $K=2$  value. After obtaining child clusters for each parent cluster, Bayesian Information Criterion can be used to score and compare newly formed clusters by their parents. By the end of this score comparison either the parent or the children will be discarded from cluster set.

### **3.2.2 Fuzzy C-means (FCM) Clustering**

Fuzzy C-means is a soft clustering algorithm where each data point in a data set has a degree of belonging to clusters. For any point  $x$ , there is a set of coefficients which gives the degree of being a member for a given cluster. As an outcome of this membership degrees, points at the edges of clusters can be shared by other clusters. Centroid of each cluster is obtained by computing the mean of all data points weighted by their cluster membership degrees. The degree of belonging is related inversely to the distance from  $x$  to the cluster center. It also depends how much weight is given to the closest center (Bezdek et al. 1981, Dunn 1973).

The steps of the algorithm can be summarized as follows:

- i. Select a number of clusters to partition data.
- ii. Randomly select cluster centroids.
- iii. Calculate the fuzzy membership for each data point using Equation 3.1.

$$\mu_{ij} = 1 / \sum_{k=1}^c (d_{ij} / d_{ik})^{(2/m-1)} \quad (3.1)$$

where  $c$  is the number of clusters,  $\mu_{ij}$  represents the membership of  $i^{\text{th}}$  data point to  $j^{\text{th}}$  cluster center  $d_{ij}$  represents the Euclidean distance between  $i^{\text{th}}$  data point and  $j^{\text{th}}$  cluster center.

- iv. Re-compute the fuzzy cluster centroids using Equation 3.2.

$$v_j = \frac{(\sum_{i=1}^n (\mu_{ij})^m x_i)}{(\sum_{i=1}^n (\mu_{ij})^m)} \quad (3.2)$$

$$\forall j = 1, 2, 3, \dots, c$$

where  $n$  is the number of tuples in data set,  $v_j$  represents the  $j^{\text{th}}$  cluster center,  $m$  is the fuzziness index,

- v. Repeat step 3 and step 4 until minimum  $J$  value is reached for Equation 3.3 or Equation 3.4 gets satisfied. .

$$J(U, V) = \sum_{i=1}^n \sum_{j=1}^c (\mu_{ij})^m \|x_i - v_j\|^2 \quad (3.3)$$

$$\|U^{(k+1)} - U^{(k)}\| < \varepsilon \quad (3.4)$$

where  $\|x_i - v_j\|$  is the Euclidean distance between  $i^{\text{th}}$  data point and  $j^{\text{th}}$  cluster center,  $k$  is the iteration step,  $\varepsilon$  is the termination criterion between  $[0,1]$ ,  $U$  is the fuzzy membership matrix and  $J$  is the objective function.

### 3.2.3 Adaptive Neuro Fuzzy Inference System (ANFIS)

ANFIS employs both neural networks and fuzzy systems for proposing a neural-fuzzy system. A fuzzy-logic system basically maps the input space to the output space in a non-linear way. To perform this type of mapping, numerical inputs of the system are converted to fuzzy domain using fuzzy sets and fuzzifiers which forms the first step of this operation. After finishing the first step, the obtained fuzzy domain gets applied with fuzzy rules and fuzzy inference engine (Jang 1993, Jang 1992). This process produces a result where defuzzifiers are used for converting it back to arithmetical domain. Gaussian functions are used for fuzzy sets and linear functions are used for rule outputs on ANFIS method. Network parameters of the system are obtained by computing coefficients of the output linear functions, mean of the membership functions and standard deviation.

The last node of the system which is the rightmost node of a network contains the calculation of summation of each output. Sugeno fuzzy model uses fuzzy if-then rules (Sugeno and Kang 1988, Takagi and Sugeno 1985). A typical fuzzy rule for a Sugeno type fuzzy system is listed below:

$$\text{If } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = f(x, y)$$

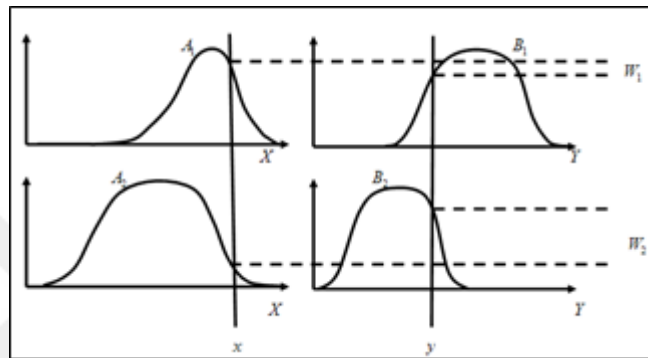
In the mentioned rule above, fuzzy sets in anterior are denoted by A and B. The result is  $z=f(x, y)$  which is crisp a function. This resulting function mostly produces a polynomial. But other than a polynomial, any other type of function can be used as long as it fits the output of the system which is within the fuzzy region characterized by the anterior of the fuzzy rule. If  $f(x, y)$  is a first-order polynomial then first-order Sugeno fuzzy model is used for such cases which is originally proposed in Sugeno and Kang (1988), Takagi and Sugeno (1985). If  $f$  is constant then zero-order Sugeno fuzzy model is adopted for these cases. This condition is a special case for Mamdani fuzzy inference system (Mamdani and Assilian 1975). In these cases, each rule's output is associated with a fuzzy singleton. It is called as a special case for Tsukamoto's fuzzy model (Tsukamoto 1979). For this condition, a given step function's membership function is defined where it is centered at each rule's result's constant. An also, a radial basis function network which involves minor constraints is functionally similar to a zero order Sugeno fuzzy model (Jang 1993). The following rules possible two rules for a first-order Sugeno fuzzy inference system:

Rule 1: If X is  $A_1$  and Y is  $B_1$ , then  $f_1 = p_1x + q_1y + r_1$

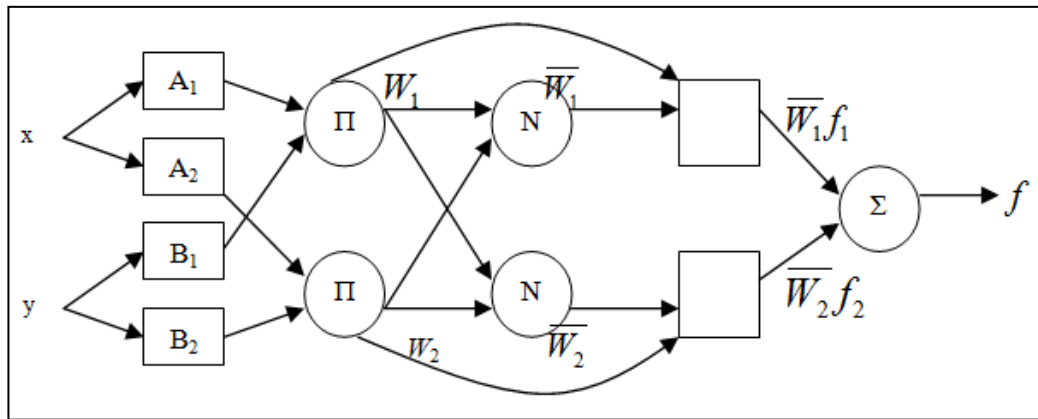
Rule 2: If X is  $A_2$  and Y is  $B_2$ , then  $f_2 = p_2x + q_2y + r_1$

In Figure 3.1 and Figure 3.2, fuzzy reasoning system is illustrated in a shortened form (Jang 1996). To increase computational efficiency, only weighted averages are used while applying defuzzification process.

**Figure 3.1: First order Sugeno fuzzy model**



**Figure 3.2: ANFIS architecture**



The output of the system is represented by  $f$  which is illustrated in Figure 3.2. An input vector such as  $[x, y]$  is supplied to system for producing this output. While producing the output, each rule's weighted average is computed. Each weight is obtained by computing the product of membership grades. Gradient vectors are computed by using adaptive networks which are bound with the fuzzy model. Computing gradient vectors is

applicable for learning phase of the Sugeno fuzzy model. The resultant network architecture is called as Adaptive Neuro-Fuzzy Inference System (ANFIS).

ANFIS employs a learning algorithm which involves both gradient descent and the least-squares estimate. Learning step continues as a series of iterations till a threshold for an acceptable error is reached. Each iteration contains two internal steps which are forward and backward steps. Forward step contains linear least-squares estimate method for getting output parameters and correcting precedent parameters. Backward step finishes correcting output parameters. For updating precedent parameters, gradient decent method is used. The output error is back-propagated through network.

While designing ANFIS, deciding the number of membership functions, training epochs, and fuzzy rules take important place to build a well-structured model. If these parameters are not adjusted properly, it may cause system to over-fit or unfit the data. To adjust these parameters correctly, a hybrid algorithm which combines the least squares method and the gradient descent method with a mean square error method is used. The smaller value for the difference between system output and actual objective means a more accurate ANFIS system. To obtain this outcome, training process tries to minimize training error as much as possible (Jang 1992, Sugeno and Kang 1988, Takagi and Sugeno 1985).

#### **3.2.4 Radial Basis Function Networks (RBFN)**

Radial basis function network (RBFN) is a popular feedforward neural network model. It contains three layers including the input layer. Unlike multilayer perceptron, hidden units in RBFN perform computations. Each point in input space is represented by a hidden unit. Output (activation) of a hidden units is based on the distance between the hidden unit's point and the instance. The activation for a given hidden unit will be stronger for closer points. A nonlinear transformation function is required for converting a distance into a similarity measure. A Gaussian activation function is a mostly used transformation function for this requirement. For each hidden unit, the bell-shaped width of Gaussian activation function can be different. The hidden units are called as Radial Basis Functions.

Radial basis function network's output layer works as similar as multilayer perceptron. Output of hidden units is received as a linear combination and it runs them through the sigmoid function.

$$f(x) = \sum_{i=1}^n w_i \varphi(\|x - c_i\|) \quad (3.5)$$

Equation 3.5 shows the output process where  $n$  is the number of neurons in hidden unit and  $c_i$  and  $w_i$  is the center vector and weight of neuron  $i$  respectively.

Network learns the following parameters:

- i. Radial basis function's centers and widths.
- ii. The weights of the hidden units which are used for producing output.

In radial basis function networks the first parameter set can be obtained without the second parameter set and the system can still produce accurate results which is an important advantage over multilayer perceptron.

To obtain first parameter set, clustering algorithms can be used by ignoring the class labels of the training data set. Applying a K-means based clustering algorithm is a possible way to obtain  $k$  basis functions for all classes independently. After obtaining the resulting radial basis functions, the second parameter set can be learned by keeping them which requires applying a learning algorithm for a linear model. This learning step can be completed very fast if the number of hidden units are less than training data set.

In Radial basis function networks, while computing distances each point is treated equally. For that reason system uses same weight for each attribute. As a result of this system behavior, such networks cannot efficiently work with irrelevant attributes which is a disadvantage (Witten and Frank 2005).



### 3.2.5 Naïve Bayes

Naïve Bayes is a simple but highly scalable probabilistic classifier which is built on Bayes' theorem. It assumes that the value of an attribute is independent of the value of any other attribute.

Let  $x$  be a data vector,  $h$  be a hypothesis for  $x$  to be member of class  $c$ .  $P(h|x)$  is the probability of  $x$  to be a member of  $c$  which is called as posterior probability.  $P(h)$  is called as the prior probability which is independent of  $x$ .

To classify an instance vector  $x = (x_1, x_2, \dots, x_n)$  having  $n$  attributes, Naïve Bayes classifier predicts that  $x$  belongs to the class which has the highest posterior probability conditioned on  $x$  among  $m$  classes. Maximum posteriori hypothesis is the class  $c_i$  for which  $P(c_i|x)$  is maximized.

$$P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)} \quad (3.6)$$

Since  $P(x)$  is constant for each class, maximizing  $P(x|c_i)P(c_i)$  is required.

$$P(x|c_i) = \prod_{k=1}^n P(x_k|c_i) \quad (3.7)$$

Probabilities of a given data being a member of a class can be easily estimated using training data set. To predict class for  $x$ ,  $P(x|c_i)P(c_i)$  is computed for each class  $c$  in data set. The classifier predicts the class if and only if:

$$P(x|c_i)P(c_i) > P(x|c_j)P(c_j) \quad \text{for } 1 \leq j \leq m, \quad j \neq i \quad (3.8)$$

Equation 3.8 indicates that the predicted class for data  $x$  is the class which has the maximum probability  $P(x|c_i)P(c_i)$  (Han and Kamber 2006).

### 3.2.6 C4.5 / J48

C4.5 (or J48) is a decision tree algorithm which uses information entropy while constructing the tree model. Algorithm uses information gain for selecting attributes which will be used for splitting nodes. The attribute with highest information gain value

suggests a shorter and a more balanced tree branch. Such a tree requires fewer tests to classify a given input.

Let node N in a tree keeps tuples form a set D. Equation 3.9 shows the formulation to calculate the expected information which is required for classifying a tuple in D where  $p_i$  is the probability for that tuple to belong to a class C and m is the number of total classes.

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (3.9)$$

$Info(D)$  returns the average amount of information required for identifying a class for a tuple in D. It is also known as the entropy of D.

Equation 3.10 is used for computing the information which is required to partition tuples in set D on attribute A containing v distinct values.

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j) \quad (3.10)$$

$Info_A(D)$  returns the information which is needed for classifying a tuple in D based on partitioning by A. Smaller information means a more desirable splitting attribute which indicates better purity of the partitions.

Formulation in Equation 3.11 defines computing information gain value for attribute A.

$$Gain(A) = Info(D) - Info_A(D) \quad (3.11)$$

The difference between the original information requirement and the new requirement which is computed after partitioning on attribute A gives the information gain value for attribute A.

While splitting nodes in the process of creating tree structure, the attribute with the highest information gain is selected as the splitting attribute for the node which is currently being processed (Han and Kamber 2006).

### 3.3 COMPARING DATA MINING TECHNIQUES

Benchmarking and comparing different data mining techniques can be done by computing confusion matrix for each method. The simplest confusion matrix can be constructed for binary classification problems where output is mapped to two clusters. For such problems the constructed confusion matrix will be a two-dimensional square matrix. For non-binary classification problems, it will be an n-dimensional square matrix.

In an n-dimensional confusion matrix, row indices represent actual values whereas column indices represent predicted values for a classification problem. Figure 3.3 shows the structure of a binary confusion matrix.

**Figure 3.3: Binary confusion matrix**

		Predicted	
		Class A	Class B
Actual	Class A	True Positive	False Negative
	Class B	False Positive	True Negative

To construct a confusion matrix, the following values are required:

- True positive (TP) value is the number of positive examples correctly predicted by the classification model.
- False negative (FN) value is the number of positive examples wrongly predicted as negative by the classification model.
- False positive (FP) value is the number of negative examples wrongly predicted as positive by the classification model.

- d. True negative (TN) value is the number of negative examples correctly predicted by the classification model.

True positive rate (TPR) which is also called as sensitivity or recall is the fraction of positive examples predicted correctly by the classification model.

$$TPR = \frac{TP}{(TP + FN)} \quad (3.12)$$

True negative rate (TNR) which is also called as specificity is the fraction of negative examples predicted correctly by the classification model.

$$TNR = \frac{TN}{(TN + FP)} \quad (3.13)$$

Precision is the ratio of true positive instances by the total number of true positive and false positive instances.

Correctness is the percentage of correctly classified instances by the classification model.

Root mean squared error (RMSE) is used for measuring the differences between actual and predicted instances. Equation 3.14 shows the RMSE formulation where n is the number of total instances, p is the predicted values and r is the actual values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (p_i - r_i)^2}{n}} \quad (3.14)$$

### 3.4 IMPLEMENTING TRIP RECOMMENDER

This study proposes a web based intelligent trip recommender system developed using Java programming language. Oracle's Mojarrá JavaServer Faces implementation was used for building web based features of the application and WEKA data mining library was included for clustering and classification functionalities.

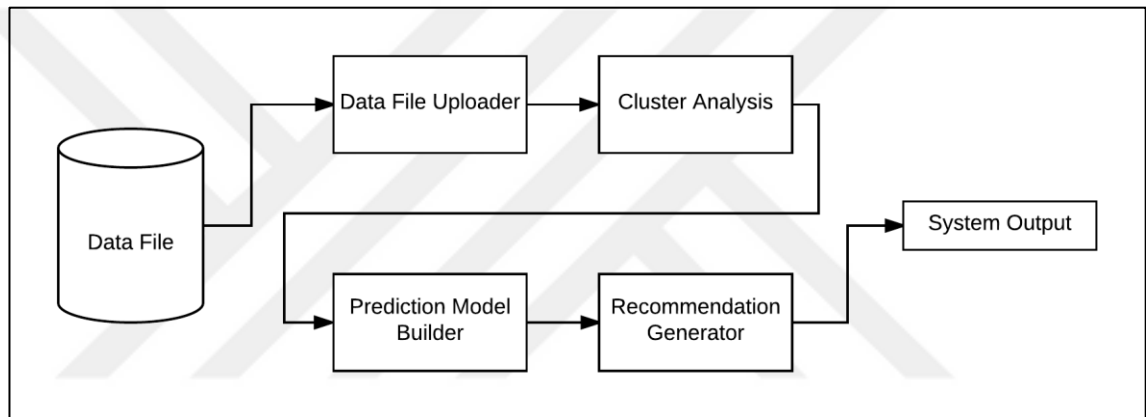
Developed application has the following four major modules:

- a. Data file uploader module.

- b. Cluster analysis module.
- c. Prediction model builder module.
- d. Recommendation generator module.

Each module performs a specific task and then it calls the next module for further processing until flow reaches recommendation generator module which is the last step. Figure 3.4 shows an overview of the connections among entire framework modules. Remaining part of this chapter describes the functionalities of each of these modules.

**Figure 3.4: Recommender system framework**



### 3.4.1 Data File Uploader Module

This module allows system user to upload data file which will be analyzed for generating predictions. Data must be provided in attribute-relation file format (ARFF). Header section of the allowed ARFF input is listed below.

@relation data

@attribute Duration numeric

@attribute Season numeric

@attribute ArrivalLocation numeric

@attribute DepartureAirline numeric

@attribute ReturningLocationFrom numeric

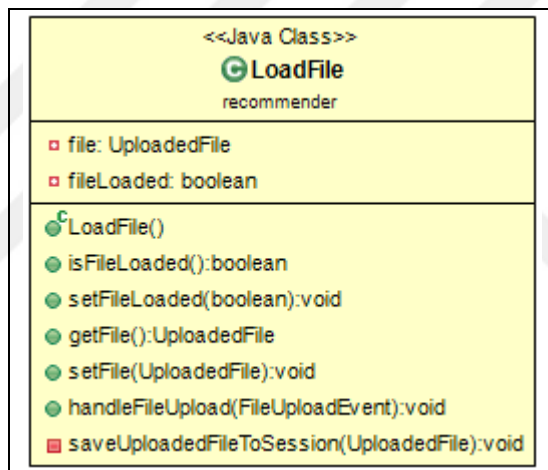
@attribute ReturningLocationTo numeric

@attribute ReturningAirline numeric  
 @attribute Gender numeric  
 @attribute FlightCost numeric  
 @attribute HotelCost numeric

Attribute definitions which are listed in the ARFF file header contains the same set of attributes which are described in data gathering and processing section.

Figure 3.5 shows the class diagram of LoadFile managed bean class. Retrieved data is saved in UploadedFile object.

**Figure 3.5: LoadFile managed bean class**



System forwards user to cluster analysis module when user successfully uploads data file.

### 3.4.2 Cluster Analysis Module

Cluster analysis module generates five different versions of the input data by clustering it into four to eight clusters. Each clustered data version is saved in ClusterCollection object which keeps clustered data in a HashMap instance. Figure 3.6 shows the class diagram of ClusterAnalysis managed bean class whereas Figure 3.7 shows the class diagram of ClusterCollection container class.

### 3.4.3 Prediction Model Builder Module

Prediction model builder allows user to select the desired algorithms which will be used for generating possible prediction models. When user selects and submits the list of desired classification algorithms, system runs the algorithms on each version of the clustered data which was generated by cluster analysis module. While building models, classification correctness of each model is computed. For each classification algorithm, evaluation object and cluster count values are saved along with the built model in a MiningModel instance. Each MiningModel instance is stored in a MiningModelCollection object which keeps these instances in a HashMap.

After building mining models, the classification-cluster combination which has the highest correctness is designated as the preferred prediction model by the system.

Recommendation generator module uses the preferred classification model for generating system output.

Figure 3.8 shows the class diagrams of MiningModel, MiningModelCollection and BuildModel classes which are used by prediction model builder module.

**Figure 3.6: CluserAnalysis managed bean class**

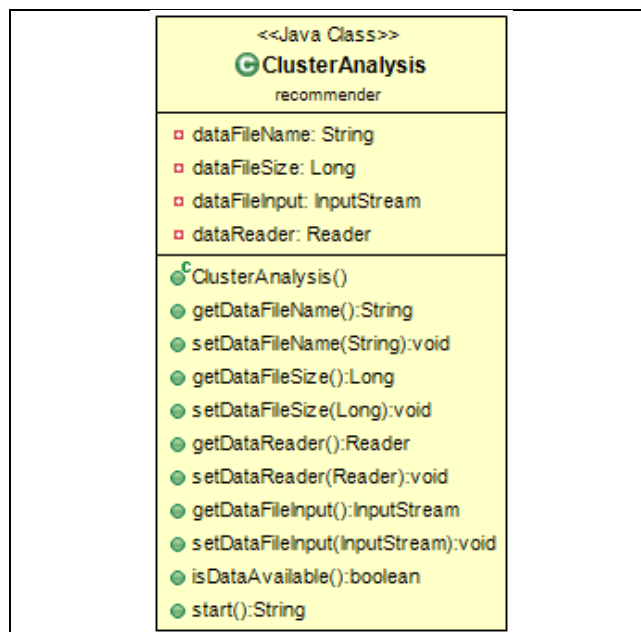


Figure 3.7: ClusterCollection container class

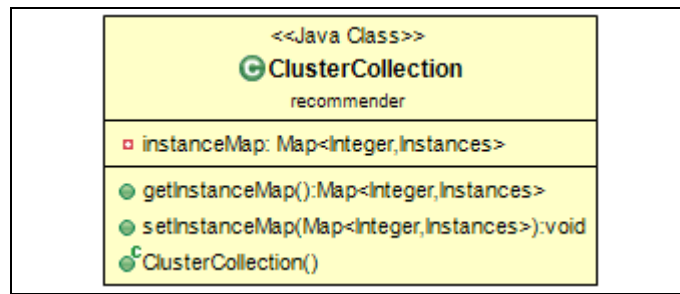
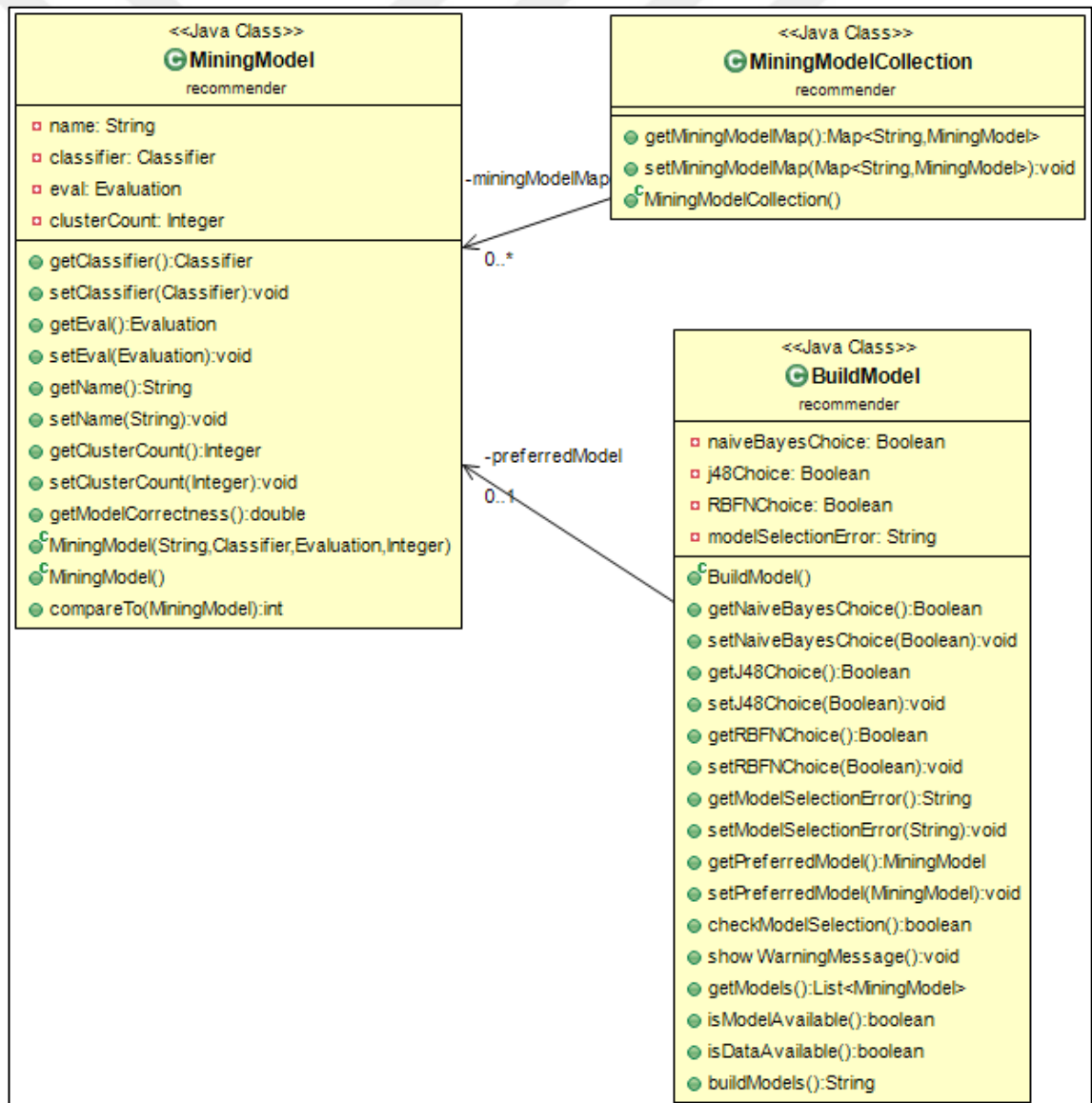


Figure 3.8: MiningModel, MiningModelCollection and BuildModel classes





### 3.4.4 Recommendation Generator Module

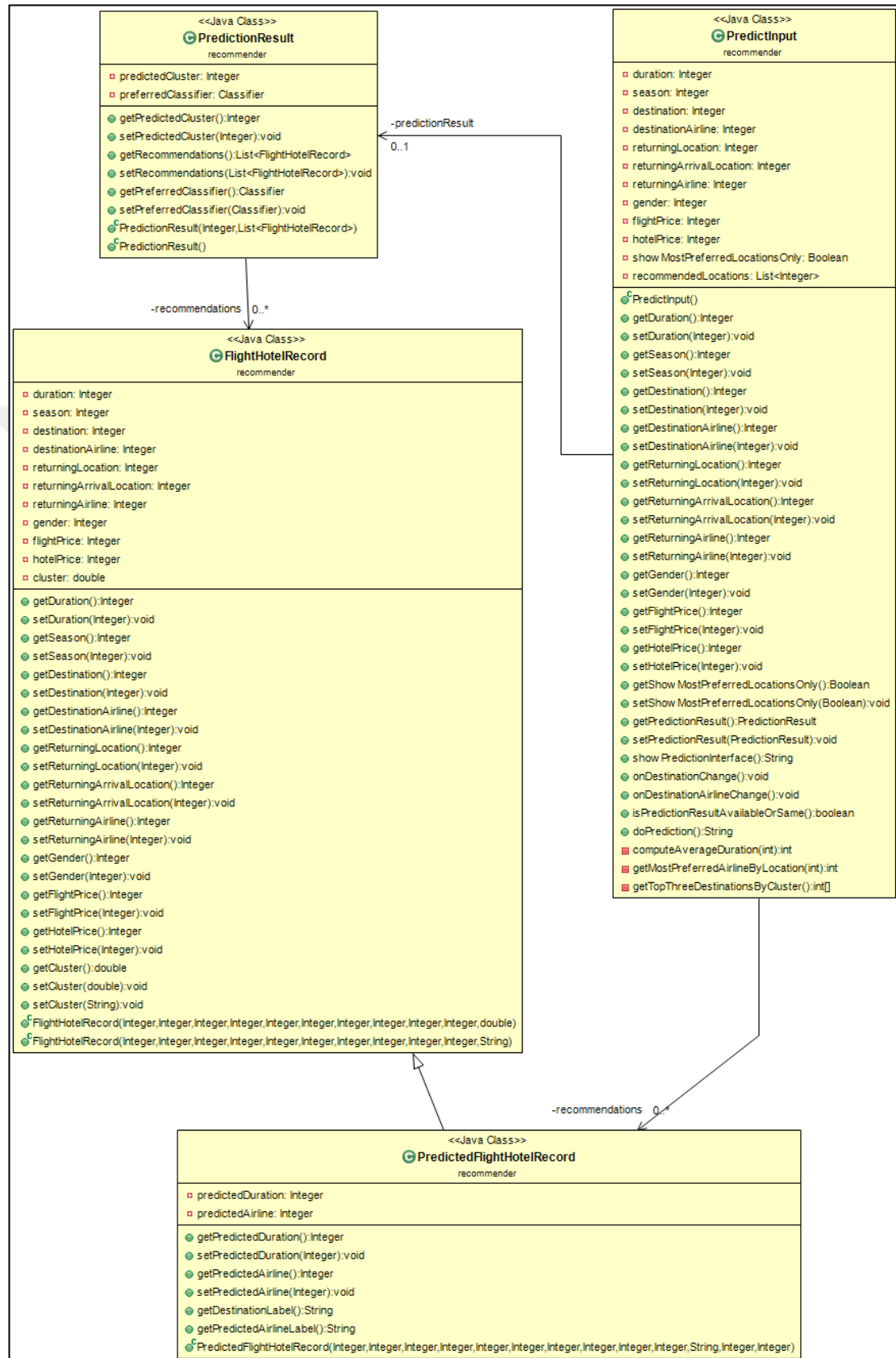
Recommendation generator module uses gender, trip duration, season, arrival city, departure airline, returning location (from), returning location (to), returning airline, flight cost and hotel cost attributes to classify a given instance. Once system finds the corresponding cluster for the given input, it uses this cluster information for finding possible locations for the obtained cluster. Based on user's choice, system can return top three locations which are preferred by the target cluster or it can return three possible locations which the other instances of the same cluster preferred before.

After finding trip locations, system computes average trip durations and it also finds the most preferred airlines for these designated locations. These tasks are performed by using preferences of similar members' past behaviors.

When system finishes the prediction process, it uses the generated values as the recommendation output.

Figure 3.9 shows the class diagrams of FlightHotelRecord, PredictedFlightHotelRecord, PredictInput and PredictionResult classes which are used by recommendation generator module.

**Figure 3.9: FlightHotelRecord, PredictedFlightHotelRecord, PredictInput and PredictionResult classes**



## 4. FINDINGS

This section states the results of this thesis study. Various classification models were built and tested using the customer hotel and flight bookings data set which was described in previous section. Other than benchmarking classification models, a web based trip recommender system was developed. Output of the developed recommender system is mentioned in Discussion section.

### 4.1 BENCHMARKING CLASSIFICATION METHODS

ANFIS, Naïve Bayes, J48 and RBFN classification methods were used for creating prediction models. ANFIS method is used for building fuzzy models, Naïve Bayes is used for building probabilistic models, J48 is used for building decision tree models and RBFN is used for building neural network models. Each method was used for building models with ten different versions of the same data set. Five of these versions were clustered using Fuzzy C-means clustering algorithm and the remaining five versions were clustered using X-means clustering algorithm. While applying clustering algorithms, each five versions of data set were generated by discovering different amount of clusters varying from four to eight. Table 4.1 shows the benchmarking results for each data set and method.

**Table 4.1: Benchmarking methods**

Classifier	Clusterer	Cluster Size	Sensitivity (Recall)	Specificity	Precision	Correctness	RMSE
ANFIS	FCM	4	0.7062	0.8863	0.6882	83.3333	0.2091
ANFIS	FCM	5	0.6348	0.9053	0.6519	85.1852	0.2257
ANFIS	FCM	6	0.5424	0.9102	0.5254	85.1852	0.2028
ANFIS	FCM	7	0.5841	0.9196	0.5133	85.9788	0.2275
ANFIS	FCM	8	0.4209	0.9233	0.4095	86.5741	0.2845
ANFIS	XM	4	0.9308	0.9746	0.9526	96.7593	0.0844
ANFIS	XM	5	0.8919	0.9802	0.9366	97.0370	0.0680
ANFIS	XM	6	0.8614	0.9745	0.9016	95.9877	0.0694
ANFIS	XM	7	0.8132	0.9782	0.8344	96.2963	0.0799
ANFIS	XM	8	0.8591	0.9841	0.8533	97.2222	0.0664
Naive Bayes	FCM	4	0.9440	0.9780	0.9470	94.4444	0.1526
Naive Bayes	FCM	5	0.9260	0.9760	0.9270	92.5926	0.1538

Naive Bayes	FCM	6	0.9260	0.9840	0.9320	92.5926	0.1575
Naive Bayes	FCM	7	0.9260	0.9880	0.9310	92.5926	0.1416
Naive Bayes	FCM	8	0.9170	0.9860	0.9210	91.6667	0.1130
Naive Bayes	XM	4	0.9540	0.9770	0.9540	95.3704	0.1322
Naive Bayes	XM	5	0.9170	0.9780	0.9200	91.6667	0.1763
Naive Bayes	XM	6	0.8520	0.9490	0.8590	85.1852	0.1822
Naive Bayes	XM	7	0.9170	0.9840	0.9230	91.6667	0.1389
Naive Bayes	XM	8	0.9350	0.9890	0.9390	93.5185	0.1145
J48	FCM	4	0.9630	0.9810	0.9630	96.2963	0.1350
J48	FCM	5	0.9260	0.9780	0.9290	92.5926	0.1601
J48	FCM	6	0.8890	0.9760	0.9020	88.8889	0.1825
J48	FCM	7	0.8980	0.9780	0.8910	89.8148	0.1624
J48	FCM	8	0.9070	0.9840	0.8970	90.7407	0.1362
J48	XM	4	0.9540	0.9830	0.9550	95.3704	0.1514
J48	XM	5	0.9810	0.9900	0.9820	98.1481	0.0936
J48	XM	6	0.9440	0.9810	0.9470	94.4444	0.1359
J48	XM	7	0.9350	0.9830	0.9380	93.5185	0.1117
J48	XM	8	0.9540	0.9870	0.9540	95.3704	0.1075
RBFN	FCM	4	0.9630	0.9870	0.9660	96.2963	0.1197
RBFN	FCM	5	0.9630	0.9870	0.9660	96.2963	0.1127
RBFN	FCM	6	0.9260	0.9830	0.9300	92.5926	0.1536
RBFN	FCM	7	0.8890	0.9780	0.9050	88.8889	0.1776
RBFN	FCM	8	0.9260	0.9910	0.9380	92.5926	0.1315
RBFN	XM	4	0.9260	0.9760	0.9360	92.5926	0.1898
RBFN	XM	5	0.8890	0.9600	0.9040	88.8889	0.2083
RBFN	XM	6	0.9260	0.9730	0.9280	92.5926	0.1490
RBFN	XM	7	0.9070	0.9800	0.9210	90.7407	0.1628
RBFN	XM	8	0.9260	0.9830	0.9320	92.5926	0.1362

According to the obtained results in Table 4.1 ANFIS has the least RMSE value when applied on data set which is clustered by X-means into eight clusters. J48 has the best sensitivity (recall), precision and correctness scores when applied on data set which is clustered by X-means into five clusters. RBFN has the best specificity score when applied on data set which is clustered by Fuzzy C-means into eight clusters.

If we compare the correctness values between these three methods we can state that correctness value of J48 method on five clustered data using X-means algorithm is 98.15. And correctness value of ANFIS method on eight clustered data using X-means algorithm

is 97.22 whereas correctness value of RBFN method on eight clustered data using Fuzzy C-means algorithm is 92.59.

Based on the stated benchmark values, J48 is the most desirable classification algorithm when applied on data set which is clustered by X-means algorithm into five clusters.

**Table 4.2: Confusion matrix for ANFIS XM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	2	1	1	0	0	0	0	0
Cluster 2	2	9	2	0	0	0	0	0
Cluster 3	0	0	23	0	0	0	0	0
Cluster 4	0	0	0	16	0	0	0	0
Cluster 5	0	0	0	2	14	1	0	0
Cluster 6	0	0	0	0	0	7	0	0
Cluster 7	0	0	0	1	0	2	18	0
Cluster 8	0	0	0	0	0	0	0	7

**Table 4.3: Confusion matrix for J48 XM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	3	0	0	0	0
Cluster 2	0	17	1	0	0
Cluster 3	0	0	50	0	1
Cluster 4	0	0	0	16	0
Cluster 5	0	0	0	0	20

**Table 4.4 Confusion matrix for RBFN FCM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	9	0	0	0	0	0	1	0
Cluster 2	0	9	0	0	0	0	0	0
Cluster 3	0	0	17	1	0	0	0	1
Cluster 4	0	0	1	14	0	0	0	2
Cluster 5	0	0	0	0	19	0	0	1
Cluster 6	0	1	0	0	0	19	0	0
Cluster 7	0	0	0	0	0	0	2	0
Cluster 8	0	0	0	0	0	0	0	11

Table 4.2, Table 4.3 and Table 4.4 shows the confusion matrices for the methods ANFIS with XM 8, J48 with XM 5 and RBFN with FCM 8 respectively. Confusion matrices of the remaining classification algorithms are listed in Appendix 3.

J48 model which has the highest correctness score generated the following decision tree paths for each cluster:

Path 1: If “Returning location (from)” > 14 and “Hotel Cost” > 3 Then output is Cluster 2

Path 2: If “Returning location (from)” > 14 and “Hotel Cost” ≤ 3 and “Gender” > 0 and “Hotel Cost” > 2 and “Season” > 2 Then output is Cluster 2

Path 3: If “Returning location (from)” > 14 and “Hotel Cost” ≤ 3 and “Gender” > 0 and “Hotel Cost” > 2 and “Season” ≤ 2 Then output is Cluster 3

Path 4: If “Returning location (from)” > 14 and “Hotel Cost” ≤ 3 and “Gender” > 0 and “Hotel Cost” ≤ 2 Then output is Cluster 2

Path 5: If “Returning location (from)” > 14 and “Hotel Cost” ≤ 3 and “Gender” ≤ 0 and “Hotel Cost” > 1 Then output is Cluster 2

Path 6: If “Returning location (from)” > 14 and “Hotel Cost” ≤ 3 and “Gender” ≤ 0 and “Hotel Cost” ≤ 1 Then output is Cluster 3

Path 7: If “Returning location (from)” ≤ 14 and “Returning Airline” > 9 Then output is Cluster 1

Path 8: If “Returning location (from)” ≤ 14 and “Returning Airline” ≤ 9 and “Season” > 2 and “Hotel Cost” > 2 Then output is Cluster 4

Path 9: If “Returning location (from)” ≤ 14 and “Returning Airline” ≤ 9 and “Season” > 2 and “Hotel Cost” ≤ 2 and “Departure Airline” > 1 Then output is Cluster 3

Path 10: If “Returning location (from)” ≤ 14 and “Returning Airline” ≤ 9 and “Season” > 2 and “Hotel Cost” ≤ 2 and “Departure Airline” ≤ 1 Then output is Cluster 5

Path 11: If “Returning location (from)”  $\leq 14$  and “Returning Airline”  $\leq 9$  and “Season”  $\leq 2$  Then output is Cluster 5

Path 1 classifies instances as Cluster 2 if returning location is between 15 and 22 and hotel cost is above 1000 TL.

Path 2 classifies instances as Cluster 2 if passenger is male and returning location is between 15 and 22 and hotel cost is between 701 TL and 1000 TL and season is summer or fall.

Path 3 classifies instances as Cluster 3 if passenger is male and returning location is between 15 and 22 and hotel cost is between 701 TL and 1000 TL and season is spring or winter.

Path 4 classifies instances as Cluster 2 if passenger is male and returning location is between 15 and 22 and hotel cost is up to 700 TL.

Path 5 classifies instances as Cluster 2 if passenger is female and returning location is between 15 and 22 and hotel cost is between 351 and 1000 TL.

Path 6 classifies instances as Cluster 3 if passenger is female and returning location is between 15 and 22 and hotel cost is up to 350 TL.

Path 7 classifies instances as Cluster 1 if returning location is between 1 and 14 and returning airline is between 10 and 77.

Path 8 classifies instances as Cluster 4 if returning location is between 1 and 14 and returning airline is between 1 and 9 and hotel cost is above 700 TL and season is summer or fall.

Path 9 classifies instances as Cluster 3 if returning location is between 1 and 14 and returning airline is between 1 and 9 and hotel cost is up to 700 TL and departure airline is not 1 and season is summer or fall.

Path 10 classifies instances as Cluster 5 if returning location is between 1 and 14 and returning airline is between 1 and 9 and hotel cost is up to 700 TL and departure airline is 1 and season is summer or fall.

Path 11 classifies instances as Cluster 5 if returning location is between 1 and 14 and returning airline is between 1 and 9 and season is winter or spring.

Based on these generated paths, characteristics of each cluster can be defined as follows:

Cluster 1 represents male or female passengers whose preferred returning location is within location codes form 1 to 14 and preferred returning airline is within company codes from 10 to 77.

Cluster 2 represents four different types of passengers:

- a. Male or female passengers whose preferred returning location is within location codes form 15 to 22 and hotel cost is above 1000 TL.
- b. Male passengers whose preferred returning location is within location codes form 15 to 22 and hotel cost is between 701 TL and 1000 TL. These passengers prefer travelling in summer season.
- c. Male passengers whose preferred returning location is within location codes form 15 to 22 and hotel cost is no more than 700 TL.
- d. Female passengers whose preferred returning location is within location codes form 15 to 22 and hotel cost is between 351 TL and 1000 TL.

Cluster 3 represents three different types of passengers:

- a. Male passengers whose preferred returning location is within location codes form 15 to 22 and hotel cost is between 701 TL and 1000 TL. These passengers prefer travelling in spring or winter seasons.
- b. Female passengers whose preferred returning location is within location codes form 15 to 22 and hotel cost is no more than 350 TL.



- c. Male or female passengers whose preferred returning location is within location codes form 1 to 14 and preferred returning airline is within company codes from 1 to 9 and preferred departure airline is any company other than the company with code 1 and hotel cost is no more than 700 TL. These passengers prefer travelling in summer or fall seasons.

Cluster 4 represents male or female passengers whose preferred returning location is within location codes form 1 to 14 and preferred airline is within company codes from 1 to 9 and hotel cost is above 700 TL. These passengers prefer travelling in summer or fall seasons.

Cluster 5 represents two different types of passengers:

- a. Male or female passengers whose preferred returning location is within location codes form 1 to 14 and preferred returning airline is within company codes from 1 to 9 and preferred departure airline is the company with code 1 and hotel cost is no more than 700 TL. These passengers prefer travelling in summer or fall seasons.
- b. Male or female passengers whose preferred returning location is within location codes form 1 to 14 and preferred returning airline is within company codes from 1 to 9. These passengers prefer travelling in winter or spring seasons.

When we observe the system output for the ANFIS model which has the highest RMSE score, we obtain the following cluster outputs for the following input vectors:

Input 1: If input is [10 3 22 1 22 16 1 1 3 3] Then output is [1]

If “Duration” = 10 and “Season” = 3 and “Arrival Location” = 22 and “Departure Airline” = 1 and “Returning Location (from)” = 22 and “Returning Location (to)” = 16 and “Returning Airline” = 1 and “Gender” = 1 and “Flight Cost” = 3 and “Hotel Cost” = 3 Then output is “Cluster 1”

Input 2: If input is [3 4 7 1 7 22 1 1 2 3] Then output is [2]

If “Duration” = 3 and “Season” = 4 and “Arrival Location” = 7 and “Departure Airline” = 1 and “Returning Location (from)” = 7 and “Returning Location (to)” = 22 and “Returning Airline” = 1 and “Gender” = 1 and “Flight Cost” = 2 and “Hotel Cost” = 3 Then output is “Cluster 2”

Input 3: If input is [4 2 22 1 22 16 1 1 1 1] Then output is [3]

If “Duration” = 4 and “Season” = 2 and “Arrival Location” = 22 and “Departure Airline” = 1 and “Returning Location (from)” = 22 and “Returning Location (to)” = 16 and “Returning Airline” = 1 and “Gender” = 1 and “Flight Cost” = 1 and “Hotel Cost” = 1 Then output is “Cluster 3”

Input 4: If input is [3 1 5 1 5 16 1 1 1 1] Then output is [4]

If “Duration” = 3 and “Season” = 1 and “Arrival Location” = 5 and “Departure Airline” = 1 and “Returning Location (from)” = 5 and “Returning Location (to)” = 16 and “Returning Airline” = 1 and “Gender” = 1 and “Flight Cost” = 1 and “Hotel Cost” = 1 Then output is “Cluster 4”

Input 5: If input is [5 2 5 1 5 16 1 1 2 4] Then output is [5]

If “Duration” = 5 and “Season” = 2 and “Arrival Location” = 5 and “Departure Airline” = 1 and “Returning Location (from)” = 5 and “Returning Location (to)” = 16 and “Returning Airline” = 1 and “Gender” = 1 and “Flight Cost” = 2 and “Hotel Cost” = 4 Then output is “Cluster 5”

Input 6: If input is [10 3 22 1 22 16 1 0 3 3] Then output is [6]

If “Duration” = 10 and “Season” = 3 and “Arrival Location” = 22 and “Departure Airline” = 1 and “Returning Location (from)” = 22 and “Returning Location (to)” = 16 and “Returning Airline” = 1 and

“Gender” = 0 and “Flight Cost” = 3 and “Hotel Cost” = 3 Then output is “Cluster 6”

Input 7: If input is [4 1 22 1 22 16 1 0 1 1] Then output is [7]

If “Duration” = 4 and “Season” = 1 and “Arrival Location” = 22 and “Departure Airline” = 1 and “Returning Location (from)” = 22 and “Returning Location (to)” = 16 and “Returning Airline” = 1 and “Gender” = 0 and “Flight Cost” = 1 and “Hotel Cost” = 1 Then output is “Cluster 7”

Input 8: If input is [3 1 10 1 10 16 1 0 2 2] Then output is [8]

If “Duration” = 3 and “Season” = 1 and “Arrival Location” = 10 and “Departure Airline” = 1 and “Returning Location (from)” = 10 and “Returning Location (to)” = 16 and “Returning Airline” = 1 and “Gender” = 0 and “Flight Cost” = 2 and “Hotel Cost” = 2 Then output is “Cluster 8”

Input 1 is classified as a member of Cluster 1 if trip duration is 10 days and season is summer and both arrival and returning (from) locations are Mediterranean Region (Turkey) and both departure and returning airline codes are 1 and returning location is Marmara Region (Turkey) and passenger is male and flight cost is between 401 TL and 700 TL and hotel cost is between 701 TL and 1000 TL.

Input 2 is classified as a member of Cluster 2 if trip duration is 3 days and season is fall and both arrival and returning (from) locations are Middle East and both departure and returning airline codes are 1 and returning location is Mediterranean Region (Turkey) and passenger is male and flight cost is between 201 TL and 400 TL and hotel cost is between 701 TL and 1000 TL.

Input 3 is classified as a member of Cluster 3 if trip duration is 4 days and season is spring and both arrival and returning (from) locations are Mediterranean Region (Turkey) and both departure and returning airline codes are 1 and returning location is Marmara Region

(Turkey) and passenger is male and flight cost is less than 200 TL and hotel cost is less than 350 TL.

Input 4 is classified as a member of Cluster 4 if trip duration is 3 days and season is winter and both arrival and returning (from) locations are Central Europe and both departure and returning airline codes are 1 and returning location is Marmara Region (Turkey) and passenger is male and flight cost is less than 200 TL and hotel cost is less than 350 TL.

Input 5 is classified as a member of Cluster 5 if trip duration is 5 days and season is spring and both arrival and returning (from) locations are Central Europe and both departure and returning airline codes are 1 and returning location is Marmara Region (Turkey) and passenger is male and flight cost is between 201 TL and 400 TL and hotel cost is between 1001 TL and 1500 TL.

Input 6 is classified as a member of Cluster 6 if trip duration is 10 days and season is summer and both arrival and returning (from) locations are Mediterranean Region (Turkey) and both departure and returning airline codes are 1 and returning location is Marmara Region (Turkey) and passenger is female and flight cost is between 401 TL and 700 TL and hotel cost is between 701 TL and 1000 TL.

Input 7 is classified as a member of Cluster 7 if trip duration is 4 days and season is winter and both arrival and returning (from) locations are Mediterranean Region (Turkey) and both departure and returning airline codes are 1 and returning location is Marmara Region (Turkey) and passenger is female and flight cost is less than 200 TL and hotel cost is less than 350 TL.

Input 8 is classified as a member of Cluster 8 if trip duration is 3 days and season is winter and both arrival and returning (from) locations are Eastern Asia and both departure and returning airline codes are 1 and returning location is Marmara Region (Turkey) and passenger is female and flight cost is between 201 TL and 400 TL and hotel cost is between 351 TL and 700 TL.

## 5. DISCUSSION

This section states the obtained findings of this study and compares the functionality of proposed recommendation framework with the recent studies which were performed on trip / travel recommendation domain.

As it is stated in the Background section, there are many studies which employ recommendation systems in tourism area. These studies can be grouped into three major titles as follows:

- a. Trip schedule recommenders.
- b. Location based travel recommenders.
- c. Social media based travel recommenders

Trip schedule recommenders get a list of desired visit locations and time constraints. Based on this input, such systems propose trip schedules to users (Chiang and Huang 2015).

Location based recommenders propose similar point of interests to users while they visit touristic attractions (Vukovic and Jevtic 2015).

Social media based recommenders use geo-tagged photos of travelers to extract information about user's preferred / interested locations and propose recommendations based on this extracted information (Xu et al. 2015).

In this study, the proposed recommender framework employs a prediction engine for finding similar users for a given user and then it proposes possible travel destinations based on user similarities. To discover similarities between users, system applies X-means clustering algorithm on data set. And this process is repeated for five times to generate five different versions of the same data set each clustered into four, five, six, seven and eight clusters respectively. These different data set versions are generated for finding the best model to represent user similarities.

After obtaining five different clustered versions of the data set, recommender builds 15 different classification models using Naïve Bayes, J48 (C4.5) and RBFN algorithms. Each classification model gets evaluated and the model which has the highest correctness value is selected as the desired model which will be used in prediction.

When system obtains the best classification algorithm, cluster of a new instance can be predicted using this selected model. After predicting instance's corresponding cluster, recommender system generates output in two different ways:

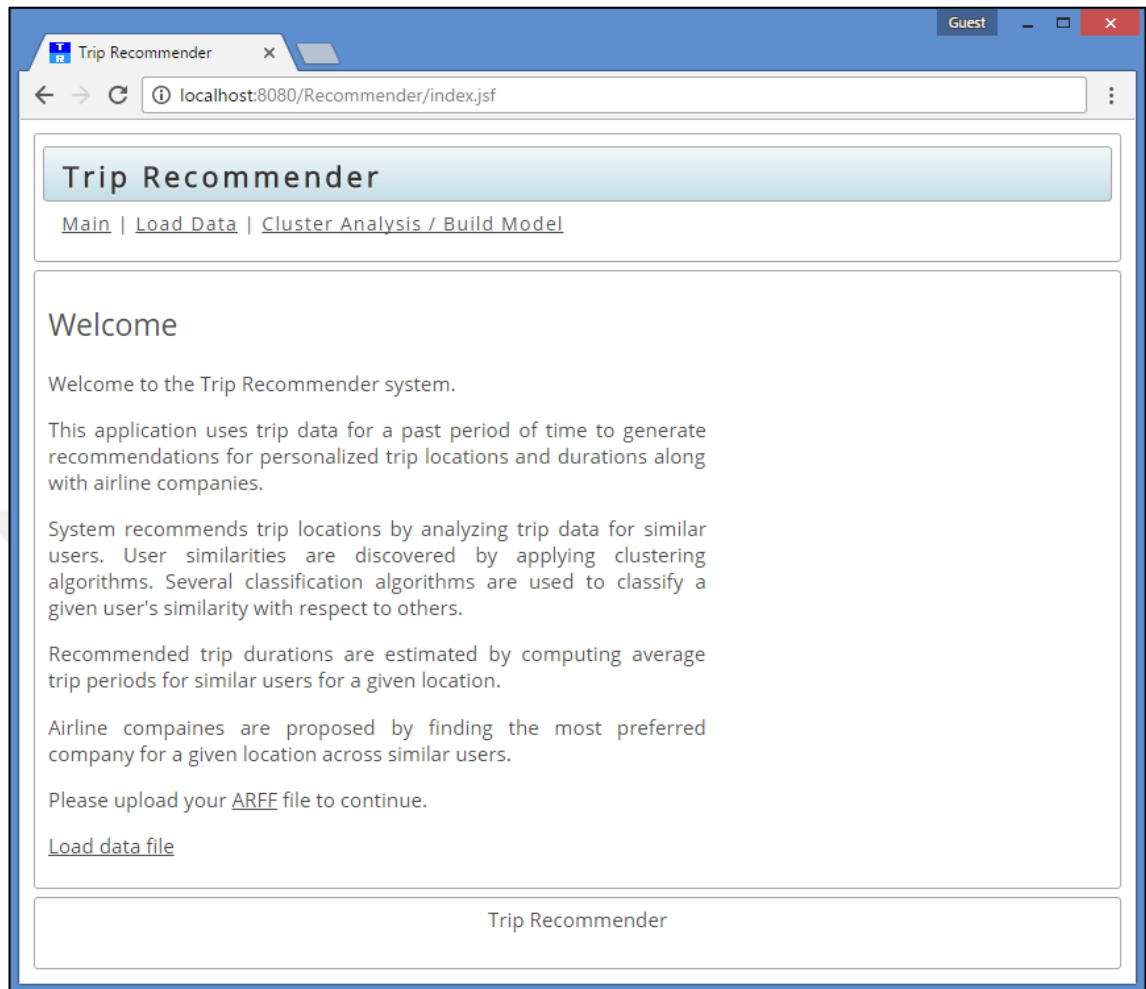
- a. System creates a list of possible destination locations by finding previously preferred destinations of users who are from the same cluster. Then it picks three random locations from this list. For each randomly selected location, system finds the most preferred airline company and average trip duration.
- b. System creates a list of possible destination locations by finding previously preferred destinations of users who are from the same cluster. Then it computes frequencies of each location and picks three locations from this list by finding top three frequency values. For each selected location, system finds the most preferred airline company and average trip duration.

In both of these approaches system finds most preferred airline company and average trip duration by using a portion of the supplied flight-hotel data set which only contains records for the predicted cluster.

The figures in this section show the output of the developed recommender system which contains the operations mentioned above.

Initial page of recommender system is displayed in Figure 5.1. System user can start the recommendation process by uploading ARFF formatted data. To access the other sections of the application, user must finish uploading data file as it is the first requirement of recommendation process. "Cluster analysis", "Build model" and "Recommendation generator" modules can only be accessible if a valid ARFF data file gets uploaded to system.

**Figure 5.1: Trip recommender initial page**



“Load data” module is listed in Figure 5.2. Only valid ARFF files are allowed to be uploaded through this module. When user successfully finishes transferring data file to server, clustering process is triggered automatically. After generating five different clustered versions of the uploaded data file, user is forwarded to the next module for building prediction models.

As it is illustrated in Figure 5.3, “Build model” module allows user to select a number of classification algorithms which will be applied on each version of data set. System requires user to pick at least one algorithm to proceed. When user selects the desired algorithms, system starts building models.

**Figure 5.2: Trip recommender data loader / clusterer**

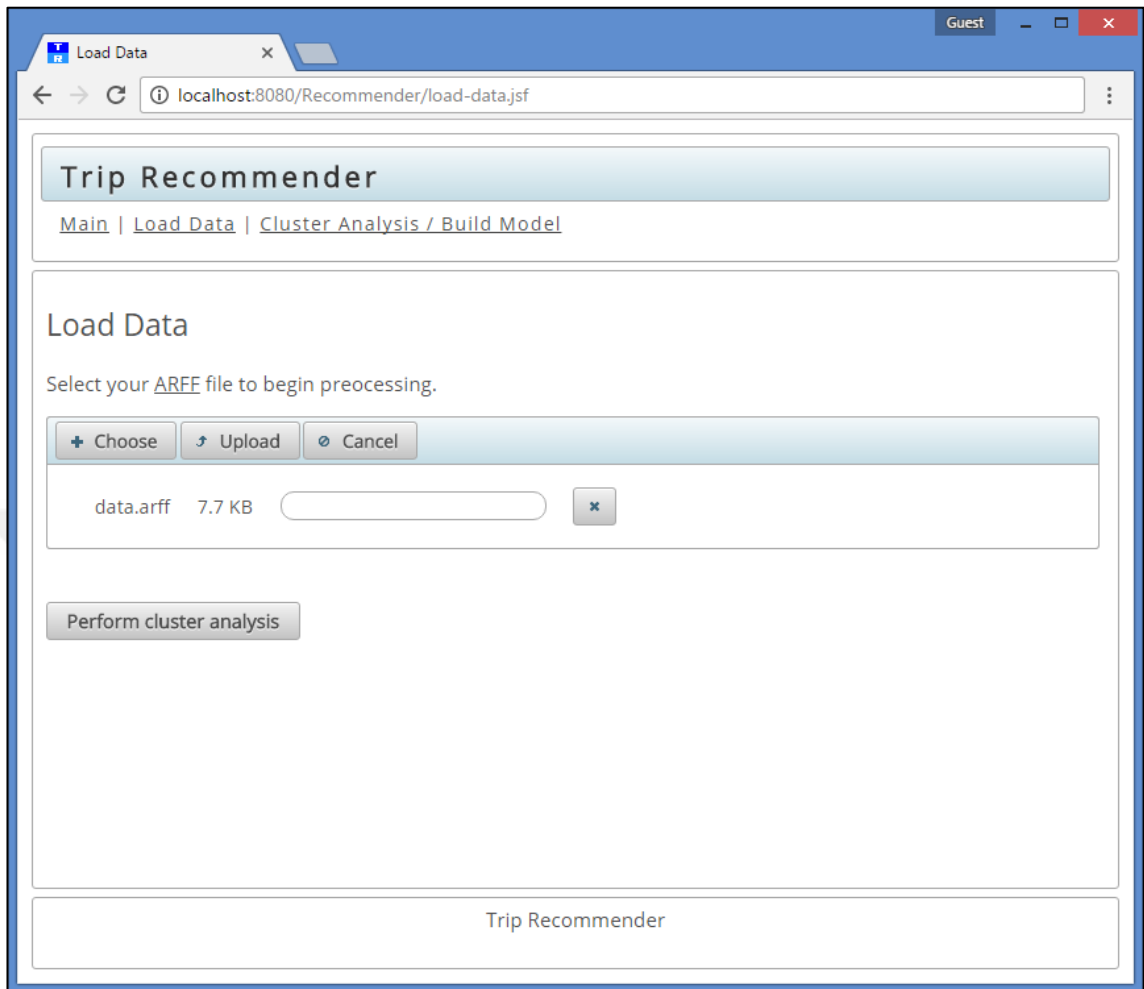


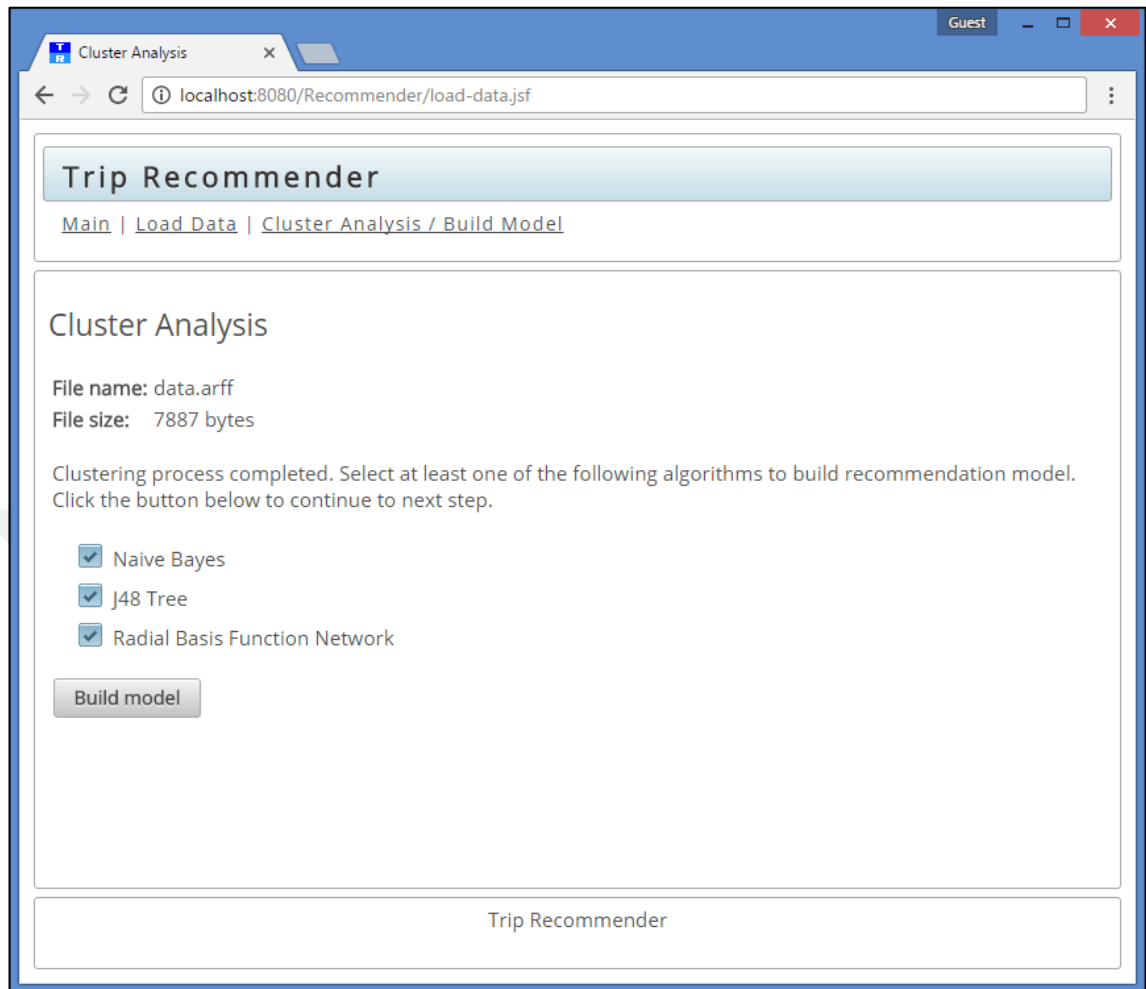
Figure 5.4 shows the progress of building models. The total amount of time which is required for building all of the prediction models depends to the number of selected algorithms and size of the uploaded data file.

On each iteration, model builder uses 66 percent of data for building and 34 percent of data for evaluating current prediction model. When model builder finishes its task, a benchmark table is displayed which contains the names of the classification algorithms, applied data set's cluster sizes and obtained correctness values. Figure 5.5 shows this benchmark table.

The prediction model which has the highest correctness score is picked as the preferred prediction model for generating recommendations.



**Figure 5.3: Trip recommender model builder (Algorithm selection)**



After obtaining the preferred prediction model, system can now generate recommendations for provided user instances. Figure 5.6 shows the first step of this process. As it is illustrated in this figure, data which will be used to represent the desired instance is collected through the provided web form. Recommender system's output type is also selected at this step. If "Recommend most preferred locations only" choice is selected then system generates its output by finding the top three locations which were preferred by users from the same cluster. But if this choice is not selected, top three condition is ignored.

System forwards user to the second step when he/she fills and submits this form. Second step is the actual output of the recommender system which is generated for the given values in this step.

**Figure 5.4: Trip recommender model builder (Progress)**

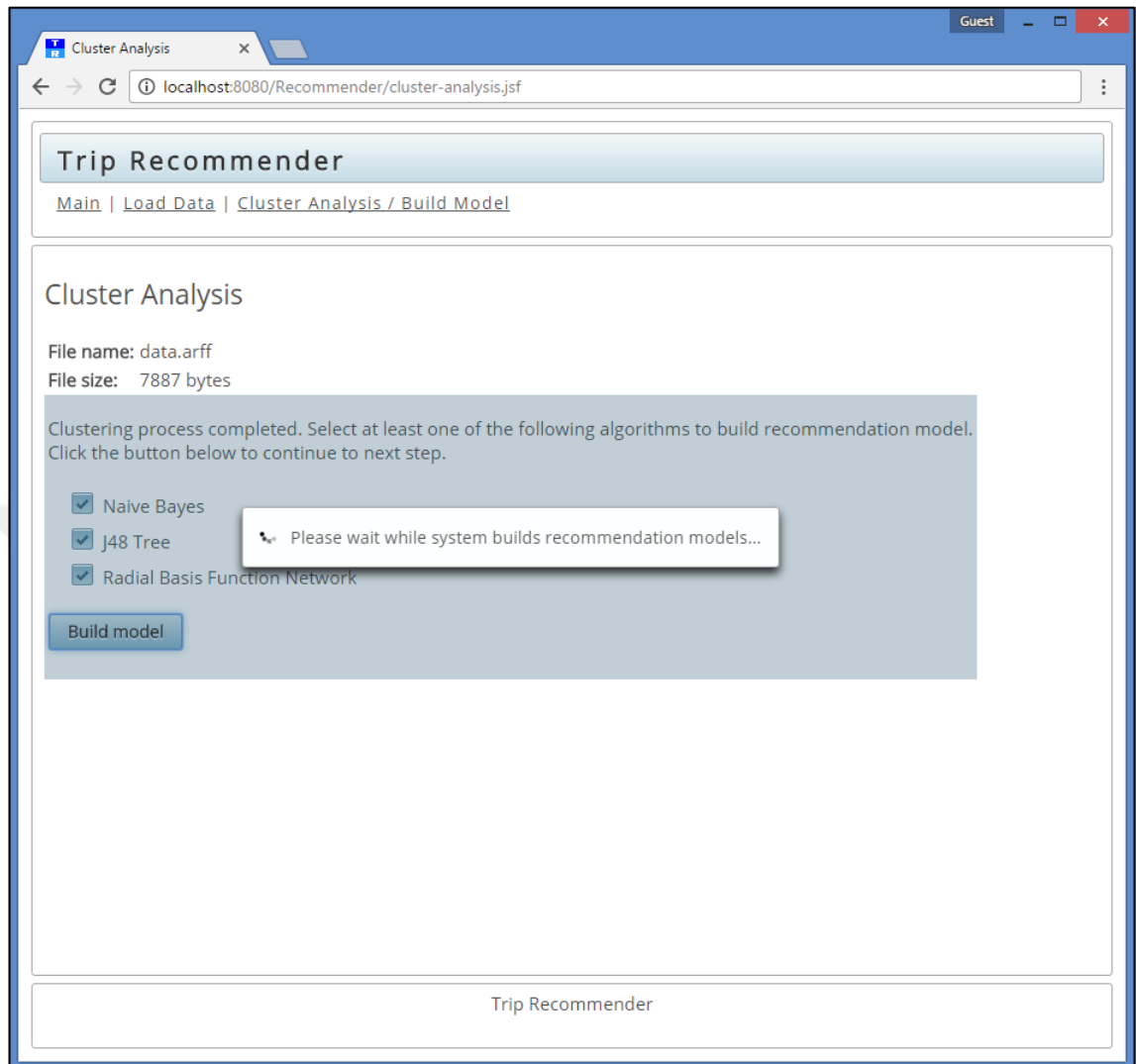


Figure 5.7 shows the output of the recommender system. For the specified input values, it generates the listed travel destinations. For each of these locations it proposes an airline company and a trip duration.

These generated locations, airline companies and trip duration values can be proposed as an individual recommendation for a user or a travel agency can use this output for planning campaigns for similar user groups.

**Figure 5.5: Trip recommender model builder (Benchmark list)**

The screenshot shows a web browser window with the address bar at localhost:8080/Recommender/cluster-analysis.jsf. The page title is 'Trip Recommender' and the breadcrumb is 'Main | Load Data | Cluster Analysis / Build Model'. The main content area is titled 'Build Model' and displays the following information:

Preferred model: J48 Tree (5 clusters)  
Model correctness: 98.15

Algorithm	Cluster Count	Correctness
J48 Tree	5	98.15
Naive Bayes	4	95.37
J48 Tree	8	95.37
J48 Tree	4	95.37
J48 Tree	6	94.44
Naive Bayes	8	93.52
J48 Tree	7	93.52
RBFN	4	92.59
RBFN	6	92.59
RBFN	8	92.59

Below the table is a pagination control showing '(1 of 2)' and buttons for navigation and a dropdown menu set to '10'. Below the table, the text reads 'Model building completed. Click the button below to start recommender.' followed by a 'Start recommender' button. At the bottom of the page, the text 'Trip Recommender' is displayed.

**Figure 5.6: Trip recommender input cluster prediction**

The screenshot shows a web browser window with the following content:

- Browser tab: Enter Data (Step 1 of 2)
- Address bar: localhost:8080/Recommender/build-model.jsf
- Page title: Trip Recommender
- Navigation links: Main | Load Data | Cluster Analysis / Build Model
- Section: Enter Data (Step 1 of 2)
- Model information: Preferred model: J48 Tree (5 clusters), Model correctness: 98.15
- Instruction: Fill the following form to generate location, trip duration and airline recommendations.
- Form fields:
  - Duration (in days): 3
  - Season: Summer
  - Destination: (Turkey) Mediterranean Region
  - Destination Airline: Turkish Airlines
  - Returning From: (Turkey) Mediterranean Region
  - Returning To: (Turkey) Marmara Region
  - Returning Airline: Turkish Airlines
  - Gender: Male
  - Flight Price: 201 - 400 TL
  - Hotel Price: 701 - 1000 TL
  - Recommend most preferred locations only: off
- Button: Generate recommendations
- Page footer: Trip Recommender

**Figure 5.7: Trip recommender output**

The screenshot shows a web browser window with the title "Recommendation Output" and the URL "localhost:8080/Recommender/prediction.jsf". The page content includes a navigation bar with "Main", "Load Data", and "Cluster Analysis / Build Model". The main heading is "Trip Recommender". Below this, the text reads "Recommendation Output (Step 2 of 2)", "Preferred model: J48 Tree (5 clusters)", and "Model correctness: 98.15". It also states "Instance classified as Cluster 2." and "The following locations and trip durations are recommended by system:". A table follows with three columns: Destination, Airline, and Recommended Trip Duration. The table lists three recommendations: (Turkey) Mediterranean Region with Atlasjet Airline for 4 days, (Turkey) Aegean Region with Pegasus Airlines for 4 days, and (Turkey) Marmara Region with Turkish Airlines for 4 days. Below the table, there are two explanatory sentences and a button labeled "Run recommender with new parameters". The footer of the page says "Trip Recommender".

**Trip Recommender**

[Main](#) | [Load Data](#) | [Cluster Analysis / Build Model](#)

### Recommendation Output (Step 2 of 2)

Preferred model: J48 Tree (5 clusters)  
Model correctness: 98.15

Instance classified as **Cluster 2.**

The following locations and trip durations are recommended by system:

Destination	Airline	Recommended Trip Duration
(Turkey) Mediterranean Region	Atlasjet Airline	4 day(s)
(Turkey) Aegean Region	Pegasus Airlines	4 day(s)
(Turkey) Marmara Region	Turkish Airlines	4 day(s)

*Airline compaines are proposed by finding the most preferred company for a given location across similar users.*  
*Recommended trip durations are estimated by computing average trip periods for similar users for a given location.*

[Run recommender with new parameters](#)

Trip Recommender

## 6. CONCLUSION

Today, searching web to obtain information about a subject is a very common task. But finding the best item within a search result is not so simple if user has many alternative data sources and is facing a large amount of data set to search. Recommender systems are proposing solutions in such cases by reducing the amount of irrelevant items and pointing relevant items to target user.

A recommender system tries to generate a rating value of an item for a target user. And according to these rating values, system tries to propose an item or many items to target user. Data mining methods are widely used for implementing such systems.

Recommender systems are being used in almost every search related area including tourism domain. Most of the implementations in this domain involve trip scheduling and location based mobile touristic attraction recommendations. But other than these two application areas, predicting possible travel destinations for users can be very advantageous especially for travel agencies. When possible travel destinations along with trip durations are combined, such information can be used for defining a package tour service which can be offered by travel agencies to a user or similar users. And proposing possible airline services for the suggested trip plan can even make the proposed trip plan more beneficial.

This study proposes an implementation of a recommender system which can generate trip suggestions to users. Implemented system processes previous flight and hotel transactions of users. Based on this analysis, proposed approach predicts clusters for system users and according to these predicted clusters, travel locations, durations and airline companies are recommended to target user or user groups.

Main purpose of this proposed approach is increasing cluster prediction correctness and providing a more flexible recommender system which can adapt itself to different data sets. To achieve this goal, proposed system tests and compares several clustering and classification strategies. Then the classification algorithm – clustering solution combination which has the highest correctness score is picked for generating

recommendations. System proposes recommendations for travel destinations, suggests possible airline companies for those proposed locations and offers trip durations. This output can be used for providing suggestions for individual users and it can be also used by travel agencies for planning and preparing travel campaigns for target user groups.

The proposed system was implemented with Java programming language. Java ServerFaces was used as the web development framework. And WEKA's data mining library was used for classification and clustering algorithms.

Data set which the implemented application used for processing is extracted from an existing travel platform's database. A total of 26,886 flight records and 4,367 hotel bookings were retrieved for processing. After removing identity columns and redundant attributes, final data set was used by the implemented application for building prediction models.

Supporting more clustering and classification algorithms can be a promising future study which can be added to this proposed system. And besides this addition, implementing a data preprocessing module can be very helpful for end users of this application. Also, allowing user to specify the set of parameters to use while building prediction models can let users to test other possible prediction models which can be derived from the same data set. In this current implementation, no hotel suggestions are available. But, proposing possible hotels for the predicted cluster's hotel price range by adding user specified constraints (like proximity to city center, being near to shore, etc.) can be another future extension for this proposed implementation.

## REFERENCES

### *Books*

Han, J. and Kamber, M., 2006. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Witten, I.H. and Frank, E., 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.





### ***Periodical Publications***

- Abbaspour, R.A. and Samadzadegan, F., 2011. Time-dependent personal tour planning and scheduling in metropolises. *Expert Systems with Applications*, 38(10), pp.12439-12452.
- Aksenov, P., Kemperman, A. and Arentze, T., 2014. Toward personalised and dynamic cultural routing: a three-level approach. *Procedia Environmental Sciences*, 22, pp.257-269.
- Batet, M., Moreno, A., Sánchez, D., Isern, D. and Valls, A., 2012. Turist@: Agent-based personalised recommendation of tourist activities. *Expert Systems with Applications*, 39(8), pp.7319-7329.
- Bezdek, J.C., Coray, C., Gunderson, R. and Watson, J., 1981. Detection and characterization of cluster substructure i. linear structure: Fuzzy c-lines. *SIAM Journal on Applied Mathematics*, 40(2), pp.339-357.
- Bobadilla, J., Hernando, A., Ortega, F. and Bernal, J., 2011. A framework for collaborative filtering recommender systems. *Expert Systems with Applications*, 38(12), pp.14609-14623.
- Bobadilla, J., Ortega, F., Hernando, A. and Alcalá, J., 2011. Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-based systems*, 24(8), pp.1310-1316.
- Bobadilla, J., Hernando, A., Ortega, F. and Gutiérrez, A., 2012. Collaborative filtering based on significances. *Information Sciences*, 185(1), pp.1-17.
- Borras, J., Moreno, A. and Valls, A., 2014. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16), pp.7370-7389.
- Bouhana, A., Soui, M. and Abed, M., 2010. A proposal of personalized itinerary search methods in the field of transport. *IFAC Proceedings Volumes*, 43(13), pp.350-355.
- Brilhante, I.R., Macedo, J.A., Nardini, F.M., Perego, R. and Renso, C., 2015. On planning sightseeing tours with TripBuilder. *Information Processing & Management*, 51(2), pp.1-15.
- Castillo, L., Armengol, E., Onaindía, E., Sebastiá, L., González-Boticario, J., Rodríguez, A., Fernández, S., Arias, J.D. and Borrajo, D., 2008. SAMAP: An user-oriented adaptive system for planning tourist visits. *Expert Systems with Applications*, 34(2), pp.1318-1332.

- Chiang, H.S. and Huang, T.C., 2015. User-adapted travel planning system for personalized schedule recommendation. *Information Fusion*, 21, pp.3-17.
- Colomo-Palacios, R., García-Peñalvo, F.J., Stantchev, V. and Misra, S., 2016. Towards a social and context-aware mobile recommendation system for tourism. *Pervasive and Mobile Computing*.
- Díez, J., Del Coz, J.J., Luaces, O. and Bahamonde, A., 2008. Clustering people according to their preference criteria. *Expert Systems with Applications*, 34(2), pp.1274-1284.
- Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3), pp.32-57.
- García, I., Sebastia, L. and Onaindia, E., 2011. On the design of individual and group recommender systems for tourism. *Expert systems with applications*, 38(6), pp.7683-7692.
- García, I., Pajares, S., Sebastia, L. and Onaindia, E., 2012. Preference elicitation techniques for group recommender systems. *Information Sciences*, 189, pp.155-175.
- García-Crespo, Á., López-Cuadrado, J.L., Colomo-Palacios, R., González-Carrasco, I. and Ruiz-Mezcua, B., 2011. Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. *Expert systems with applications*, 38(10), pp.13310-13319.
- García-Magariño, I., 2015. ABSTUR: an agent-based simulator for tourist urban routes. *Expert Systems with Applications*, 42(12), pp.5287-5302.
- García-Palomares, J.C., Gutiérrez, J. and Mínguez, C., 2015. Identification of tourist hot spots based on social networks: A comparative analysis of European metropolises using photo-sharing services and GIS. *Applied Geography*, 63, pp.408-417.
- Gavalas, D., Konstantopoulos, C., Mastakas, K. and Pantziou, G., 2014. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39, pp.319-333.
- Hadjali, A., Mokhtari, A. and Pivert, O., 2012. Expressing and processing complex preferences in route planning queries: Towards a fuzzy-set-based approach. *Fuzzy Sets and Systems*, 196, pp.82-104.
- Han, J. and Lee, H., 2015. Adaptive landmark recommendations for travel planning: Personalizing and clustering landmarks using geo-tagged social media. *Pervasive and Mobile Computing*, 18, pp.4-17.

- Hawalah, A. and Fasli, M., 2014. Utilizing contextual ontological user profiles for personalized recommendations. *Expert Systems with Applications*, 41(10), pp.4777-4797.
- Hsu, F.M., Lin, Y.T. and Ho, T.K., 2012. Design and implementation of an intelligent recommendation system for tourist attractions: The integration of EBM model, Bayesian network and Google Maps. *Expert Systems with Applications*, 39(3), pp.3257-3264.
- Huang, Y. and Bian, L., 2009. A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet. *Expert Systems with Applications*, 36(1), pp.933-943.
- Jang, J.S., 1992. Self-learning fuzzy controllers based on temporal backpropagation. *IEEE Transactions on neural networks*, 3(5), pp.714-723.
- Jang, J.S., 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), pp.665-685.
- Jang, J.S.R., 1996, September. Input selection for ANFIS learning. In *Proceedings of the fifth IEEE international conference on fuzzy systems* (Vol. 2, pp. 1493-1499).
- Kabassi, K., 2010. Personalizing recommendations for tourists. *Telematics and Informatics*, 27(1), pp.51-66.
- Kim, J.K., Kim, H.K., Oh, H.Y. and Ryu, Y.U., 2010. A group recommendation system for online communities. *International Journal of Information Management*, 30(3), pp.212-219.
- Lucas, J.P., Luz, N., Moreno, M.N., Anacleto, R., Figueiredo, A.A. and Martins, C., 2013. A hybrid recommendation approach for a tourism system. *Expert Systems with Applications*, 40(9), pp.3532-3550.
- Lü, L., Medo, M., Yeung, C.H., Zhang, Y.C., Zhang, Z.K. and Zhou, T., 2012. Recommender systems. *Physics Reports*, 519(1), pp.1-49.
- Majid, A., Chen, L., Mirza, H.T., Hussain, I. and Chen, G., 2015. A system for mining interesting tourist locations and travel sequences from public geo-tagged photos. *Data & Knowledge Engineering*, 95, pp.66-86.
- Mamdani, E.H. and Assilian, S., 1975. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1), pp.1-13.

- Mild, A. and Reutterer, T., 2003. An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services*, 10(3), pp.123-133.
- Montejo-Ráez, A., Perea-Ortega, J.M., García-Cumbreras, M.Á. and Martínez-Santiago, F., 2011. Otiüm: A web based planner for tourism and leisure. *Expert Systems with Applications*, 38(8), pp.10085-10093.
- Moreno, A., Valls, A., Isern, D., Marin, L. and Borràs, J., 2013. Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities. *Engineering Applications of Artificial Intelligence*, 26(1), pp.633-651.
- Moussa, S., Soui, M. and Abed, M., 2013. User Profile and Multi-criteria Decision Making: Personalization of Traveller's Information in Public Transportation. *Procedia Computer Science*, 22, pp.411-420.
- Neves, A.R.D.M., Carvalho, Á.M.G. and Ralha, C.G., 2014. Agent-based architecture for context-aware and personalized event recommendation. *Expert Systems with Applications*, 41(2), pp.563-573.
- Noguera, J.M., Barranco, M.J., Segura, R.J. and MartíNez, L., 2012. A mobile 3D-GIS hybrid recommender system for tourism. *Information Sciences*, 215, pp.37-52.
- Parvaneh, Z., Arentze, T. and Timmermans, H., 2012. Understanding travelers' behavior in provision of travel information: a Bayesian belief approach. *Procedia-Social and Behavioral Sciences*, 54, pp.251-260.
- Pelleg, D. and Moore, A.W., 2000, June. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *ICML (Vol. 1)*.
- Ragunathan, T., Battula, S.K., Vedika, J., Anitha, V., Tarun, T., Prasad, M.S. and Kalyani, M.U., 2015. ITTS: Intelligent Transport and Tourism System. *Procedia Computer Science*, 50, pp.191-196.
- Saleh, A.I., El Desouky, A.I. and Ali, S.H., 2015. Promoting the performance of vertical recommendation systems by applying new classification techniques. *Knowledge-Based Systems*, 75, pp.192-223.
- Schiaffino, S. and Amandi, A., 2009. Building an expert travel agent as a software agent. *Expert Systems with Applications*, 36(2), pp.1291-1299.

- Socharoentum, M. and Karimi, H.A., 2016. Multi-modal transportation with multi-criteria walking (MMT-MCW): Personalized route recommender. *Computers, Environment and Urban Systems*, 55, pp.44-54.
- Subramaniaswamy, V., Vijayakumar, V., Logesh, R. and Indragandhi, V., 2015. Intelligent Travel Recommendation System By Mining Attributes From Community Contributed Photos. *Procedia Computer Science*, 50, pp.447-455.
- Sugeno, M. and Kang, G.T., 1988. Structure identification of fuzzy model. *Fuzzy sets and systems*, 28(1), pp.15-33.
- Sun, Y., Fan, H., Bakillah, M. and Zipf, A., 2015. Road-based travel recommendation using geo-tagged images. *Computers, Environment and Urban Systems*, 53, pp.110-122.
- Takagi, T. and Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1), pp.116-132.
- Tsai, C.Y. and Chung, S.H., 2012. A personalized route recommendation service for theme parks using RFID information and tourist behavior. *Decision Support Systems*, 52(2), pp.514-527.
- Tsai, C.Y. and Lai, B.H., 2015. A location-item-time sequential pattern mining algorithm for route recommendation. *Knowledge-Based Systems*, 73, pp.97-110.
- Tsukamoto, Y., 1979. An approach to fuzzy reasoning method. *Advances in fuzzy set theory and applications*, 137, p.149.
- Umanets, A., Ferreira, A. and Leite, N., 2014. GuideMe—A tourist guide with a recommender system and social interaction. *Procedia Technology*, 17, pp.407-414.
- Varfolomeyev, A., Korzun, D., Ivanovs, A., Soms, H. and Petrina, O., 2015. Smart space based recommendation service for historical tourism. *Procedia Computer Science*, 77, pp.85-91.
- Vukovic, M. and Jevtic, D., 2015. Agent-based Movement Analysis and Location Prediction in Cellular Networks. *Procedia Computer Science*, 60, pp.517-526.
- Wang, C.S. and Yang, H.L., 2012. A recommender mechanism based on case-based reasoning. *Expert Systems with Applications*, 39(4), pp.4335-4343.

- Wang, Y.F., Chuang, Y.L., Hsu, M.H. and Keh, H.C., 2004. A personalized recommender system for the cosmetic business. *Expert Systems with Applications*, 26(3), pp.427-434.
- Xiang, Z., Kim, S.E., Hu, C. and Fesenmaier, D.R., 2007. Language representation of restaurants: Implications for developing online recommender systems. *International Journal of Hospitality Management*, 26(4), pp.1005-1018.
- Xu, Z., Chen, L. and Chen, G., 2015. Topic based context-aware travel recommendation method exploiting geotagged photos. *Neurocomputing*, 155, pp.99-107.
- Yang, W.S. and Hwang, S.Y., 2013. iTravel: A recommender system in mobile peer-to-peer environment. *Journal of Systems and Software*, 86(1), pp.12-20.



## APPENDICES



## APPENDIX 1: Database Schema of Travel Portal

Figure 1: LogData tables

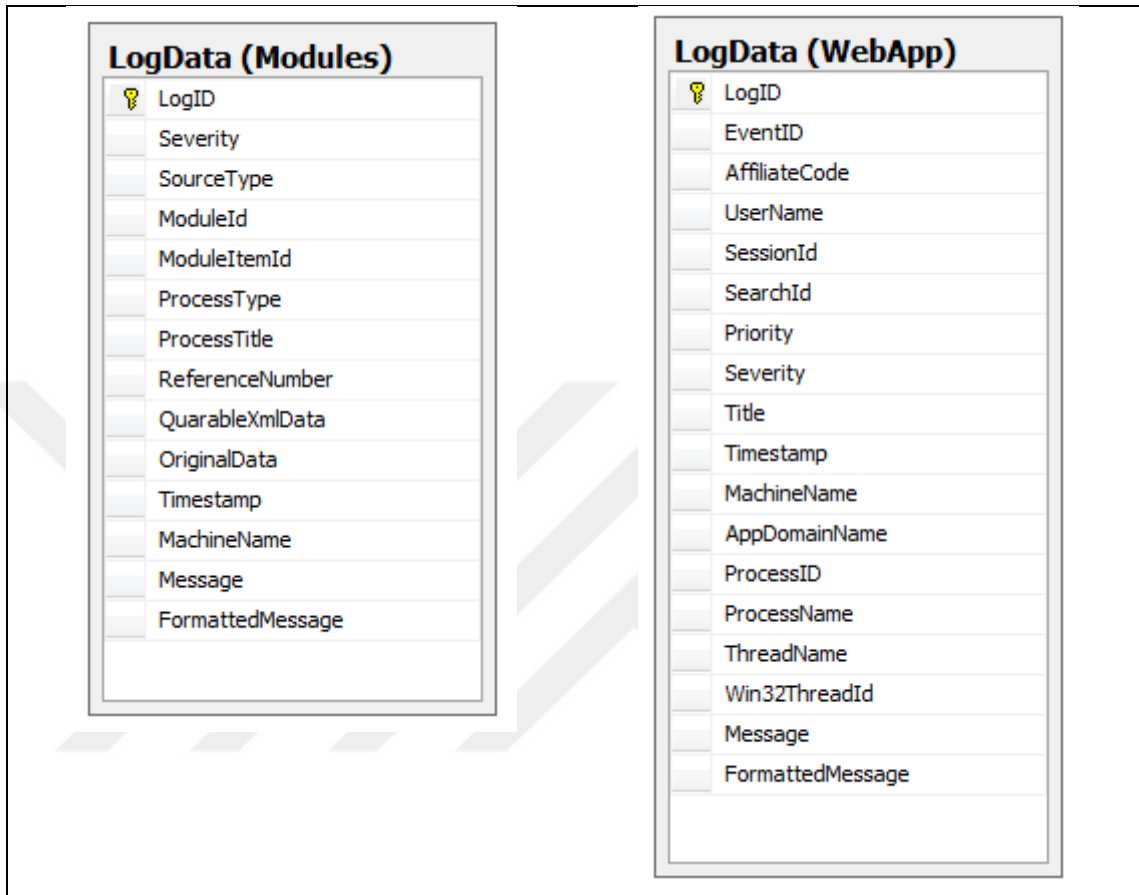




Figure 2: Baskets and BasketItems tables

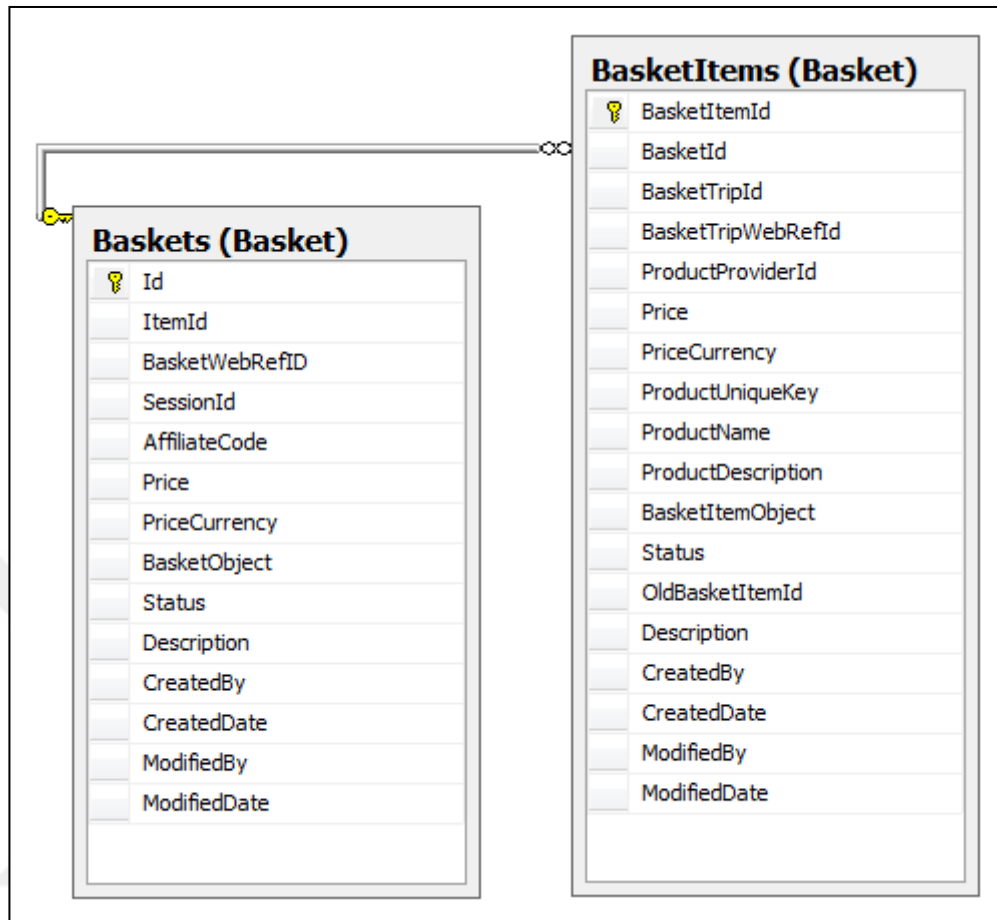
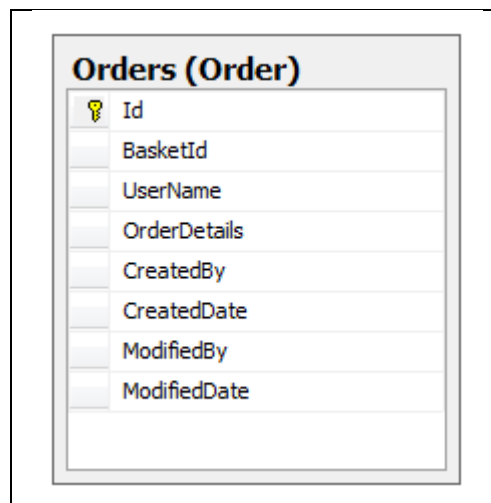


Figure 3: Orders table



**Figure 4: Reservation table**


















<b>Reservations (Order)</b>	
	Id
	UserName
	ModuleItemId
	OptionDate
	ReservationPnr
	LastName
	BasketId
	BasketItemID
	ReservationObject
	ReservationDetails
	Status
	Description
	CreatedBy
	CreatedDate
	ModifiedBy
	ModifiedDate

Figure 5: PassengerInfos and BillingInfos tables

<b>PassengerInfos (Membership)</b>	
	Id
	UserName
	AccountName
	Title
	FirstName
	LastName
	TCKNo
	Birthday
	PersonType
	ContactInfoEmail
	ContactInfoGsm
	ContactInfoPhones
	IsMasterContact
	Status
	CreatedBy
	CreatedDate
	ModifiedBy
	ModifiedDate
	Nationality
	Gender


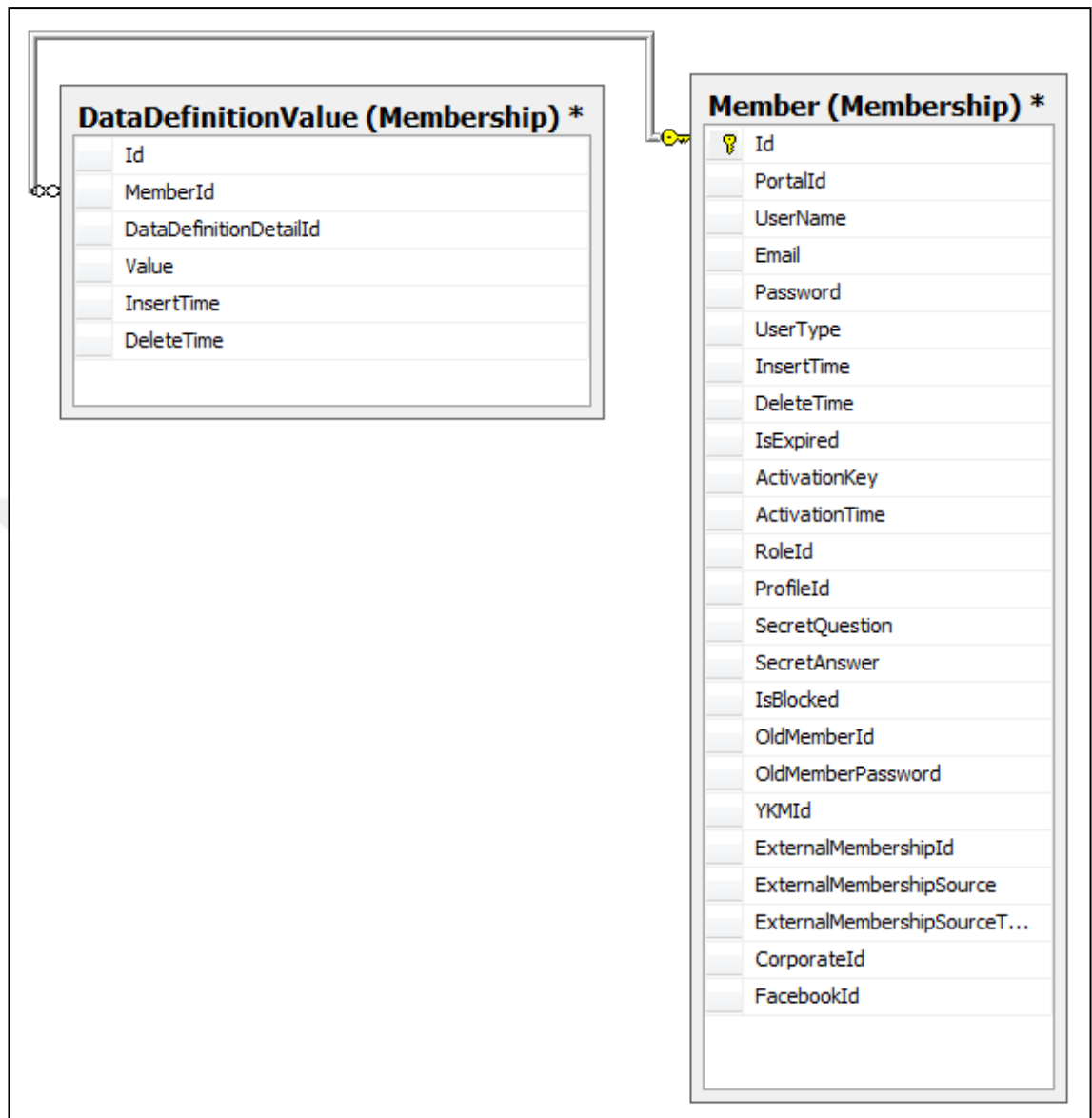
<b>BillingInfos (Order)</b>	
	Id
	UserName
	Name
	Surname
	Title
	BillingInfoName
	HomeTelephone
	MobilTelephone
	BusinessTelephone
	Fax
	IsCompany
	Address
	ZipCode
	City
	Country
	District
	TCKNo
	CompanyName
	TaxOffice
	TaxNo
	Status
	CreatedBy
	CreatedDate
	ModifiedBy
	ModifiedDate

Figure 6: DataDefinitionValue and Member tables



## APPENDIX 2: Flight-Hotel Dataset Attribute Values

**Table 1: Location codes**

<b>Code</b>	<b>Description</b>
1	Northern Europe
2	Southern Europe
3	Eastern Europe
4	Western Europe
5	Central Europe
6	Balkans
7	Middle East
8	Northern Asia
9	Southern Asia
10	Eastern Asia
11	Western Asia
12	Central Asia
13	Africa
14	America
15	Australia
16	(Turkey) Marmara Region
17	(Turkey) Black Sea Region
18	(Turkey) Central Anatolia Region
19	(Turkey) Southeastern Anatolia Region
20	(Turkey) Aegean Region
21	(Turkey) Eastern Anatolia Region
22	(Turkey) Mediterranean Region

**Table 2: Airline codes**

<b>Code</b>	<b>Airline Company</b>
1	Turkish Airlines
2	Pegasus Airlines
3	Qatar Airways
4	Anadolu Jet
5	Sun Express Airline
6	Atlasjet Airline
7	EgyptAir
8	Singapore Airlines
9	Aeroflot Russian Airlines
10	Lufthansa
11	Royal Air Maroc
12	Swissair
13	Air France
14	Asiana Airlines
15	Malaysia Airlines
16	British Airways
17	Yakutia Airlines
18	Air Canada
19	Air Moldova
20	Alitalia
21	Emirates
22	United Airlines
23	KLM Royal Dutch Airlines
24	Ukraine International Airlines
25	Air China
26	Saudi Arabian Airlines
27	Tarom
28	Delta Air Lines
29	JetBlue

30	Azerbaijan Airlines
31	American Airlines
32	Austrian Airlines
33	Thai Airways
34	Vueling Airlines
35	Olympic Air
36	Meridiana
37	Germanwings
38	Air Astana
39	Etihad Airways
40	Air Europa
41	Air Baltic
42	Aegean Airlines
43	Korean Air
44	China Southern Airlines
45	Jat Airways - Air Serbia
46	Darwin Airline
47	Jetairfly
48	Adria Airways
49	Scandinavian Airlines
50	Flydubai
51	Iberia Airlines
52	TAP Portugal
53	Belavia - Belarusian Airlines
54	US Airways
55	Middle East Airlines
56	Aerosvit Airlines
57	Aeroméxico
58	Tatarstan Airlines
59	TAM Airlines
60	Sun

61	Transaero
62	Royal Jordanian Airlines
63	Condor Flugdienst
64	Croatia Airlines
65	Air Malta
66	BH Air
67	Avianca
68	Air Transat
69	LOT Polish Airlines
70	Rossiya - Russian Airlines
71	Dniproavia - Ukrainian Airways
72	Bangkok Airways
73	South African Airways
74	Air india
75	Hong Kong Airlines
76	Air Berlin
77	Frontier Airlines

**Table 3: Gender codes**

<b>Code</b>	<b>Gender</b>
0	Female
1	Male

**Table 4: Season codes**

<b>Code</b>	<b>Season</b>
1	Winter
2	Spring
3	Summer
4	Fall



**Table 5: Flight cost codes**

<b>Code</b>	<b>Description</b>
1	< 200
2	201 – 400
3	401 – 700
4	701 – 1400
5	1401 – 3000
6	4000 +

**Table 6: Hotel cost codes**

<b>Code</b>	<b>Description</b>
1	< 350
2	351 – 700
3	701 – 1000
4	1001 – 1500
5	1501 – 2500
6	2500 +

### APPENDIX 3: Confusion Matrices for Classification Algorithms

**Table 1: Confusion matrix for ANFIS FCM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	14	0	0	0
Cluster 2	3	15	8	1
Cluster 3	0	10	17	3
Cluster 4	0	1	10	26

**Table 2: Confusion matrix for ANFIS FCM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	21	5	2	2	0
Cluster 2	6	16	2	0	0
Cluster 3	2	2	12	4	0
Cluster 4	1	4	8	7	0
Cluster 5	0	0	0	2	12

**Table 3: Confusion matrix for ANFIS FCM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	13	5	1	0	1	0
Cluster 2	9	8	4	1	0	0
Cluster 3	6	4	1	0	1	0
Cluster 4	0	1	0	13	3	3
Cluster 5	0	0	2	4	13	1
Cluster 6	0	0	0	1	1	12

**Table 4: Confusion matrix for ANFIS FCM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	10	4	2	2	1	0	1
Cluster 2	2	7	3	1	0	0	0
Cluster 3	0	2	13	2	0	0	0
Cluster 4	0	1	2	8	0	3	0
Cluster 5	1	1	3	1	8	2	5
Cluster 6	0	1	0	0	6	7	7
Cluster 7	0	0	0	0	0	0	2

**Table 5: Confusion matrix for ANFIS FCM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	11	1	2	0	0	0	0	0
Cluster 2	5	6	0	0	0	0	0	0
Cluster 3	2	2	1	2	1	3	1	4
Cluster 4	0	0	0	9	2	0	0	6
Cluster 5	0	1	0	3	10	3	0	1
Cluster 6	0	5	1	1	1	7	1	2
Cluster 7	0	0	0	0	0	0	0	2
Cluster 8	0	0	0	1	1	1	3	6

**Table 6: Confusion matrix for ANFIS XM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	12	1	0	0
Cluster 2	0	20	1	0
Cluster 3	0	0	42	1
Cluster 4	0	0	4	27

**Table 7: Confusion matrix for ANFIS XM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	3	1	0	0	0
Cluster 2	0	19	2	0	0
Cluster 3	0	0	40	2	0
Cluster 4	0	0	0	20	1
Cluster 5	0	0	0	2	18

**Table 8: Confusion matrix for ANFIS XM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	3	1	0	0	0	0
Cluster 2	0	23	6	0	0	0
Cluster 3	0	0	29	0	0	0
Cluster 4	0	0	0	15	1	0
Cluster 5	0	0	0	2	11	3
Cluster 6	0	0	0	0	0	14

**Table 9: Confusion matrix for ANFIS XM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	2	1	1	0	0	0	0
Cluster 2	2	9	2	0	0	0	0
Cluster 3	0	0	23	0	0	0	0
Cluster 4	0	0	0	19	0	0	0
Cluster 5	0	0	0	0	11	3	0
Cluster 6	0	0	0	0	0	12	2
Cluster 7	0	0	0	1	0	2	18

**Table 10: Confusion matrix for ANFIS XM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	2	1	1	0	0	0	0	0
Cluster 2	2	9	2	0	0	0	0	0
Cluster 3	0	0	23	0	0	0	0	0
Cluster 4	0	0	0	16	0	0	0	0
Cluster 5	0	0	0	2	14	1	0	0
Cluster 6	0	0	0	0	0	7	0	0
Cluster 7	0	0	0	1	0	2	18	0
Cluster 8	0	0	0	0	0	0	0	7

**Table 11: Confusion matrix for Naïve Bayes FCM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	9	0	1	0
Cluster 2	0	25	0	1
Cluster 3	0	0	29	0
Cluster 4	0	2	2	39

**Table 12: Confusion matrix for Naïve Bayes FCM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	29	0	0	0	0
Cluster 2	2	18	0	3	0
Cluster 3	0	0	20	0	0
Cluster 4	0	2	0	24	0
Cluster 5	1	0	0	0	9

**Table 13: Confusion matrix for Naïve Bayes FCM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	18	0	0	0	0	0
Cluster 2	0	20	0	0	1	0
Cluster 3	5	0	8	0	0	0
Cluster 4	0	0	0	19	0	0
Cluster 5	0	0	0	1	26	0
Cluster 6	0	0	1	0	0	9

**Table 14: Confusion matrix for Naïve Bayes FCM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	26	0	0	0	0	1	0
Cluster 2	0	8	2	0	0	0	0
Cluster 3	0	3	16	0	0	0	0
Cluster 4	0	0	0	9	0	0	1
Cluster 5	1	0	0	0	20	0	0
Cluster 6	0	0	0	0	0	19	0
Cluster 7	0	0	0	0	0	0	2

**Table 15: Confusion matrix for Naïve Bayes FCM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	9	0	0	0	0	0	1	0
Cluster 2	0	8	0	0	0	1	0	0
Cluster 3	0	0	17	1	0	0	0	1
Cluster 4	0	0	1	16	0	0	0	0
Cluster 5	0	0	0	0	20	0	0	0
Cluster 6	0	1	0	0	0	19	0	0
Cluster 7	0	0	0	0	0	0	2	0
Cluster 8	0	0	3	0	0	0	0	8

**Table 16: Confusion matrix for Naïve Bayes XM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	11	0	0	0
Cluster 2	0	16	2	0
Cluster 3	0	2	47	1
Cluster 4	0	0	0	29

**Table 17: Confusion matrix for Naïve Bayes XM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	3	0	0	0	0
Cluster 2	0	17	1	0	0
Cluster 3	0	2	47	0	2
Cluster 4	0	0	0	15	1
Cluster 5	1	0	0	2	17

**Table 18: Confusion matrix for Naïve Bayes XM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	3	0	0	0	0	0
Cluster 2	0	28	6	0	0	0
Cluster 3	0	2	28	1	1	0
Cluster 4	0	0	0	10	2	0
Cluster 5	0	0	3	1	14	0
Cluster 6	0	0	0	0	0	9

**Table 19: Confusion matrix for Naïve Bayes XM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	3	0	0	0	0	0	0
Cluster 2	0	12	0	1	0	0	0
Cluster 3	0	0	24	0	1	0	0
Cluster 4	0	0	0	13	0	0	0
Cluster 5	0	0	3	1	13	0	0
Cluster 6	0	1	0	0	0	8	0
Cluster 7	0	1	0	1	0	0	26

**Table 20: Confusion matrix for Naïve Bayes XM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	3	0	0	0	0	0	0	0
Cluster 2	0	12	0	1	0	0	0	0
Cluster 3	0	0	24	0	1	0	0	0
Cluster 4	0	0	0	12	0	0	0	0
Cluster 5	0	0	1	1	16	0	0	0
Cluster 6	0	1	0	0	0	2	0	0
Cluster 7	0	1	0	0	1	0	26	0
Cluster 8	0	0	0	0	0	0	0	6

**Table 21: Confusion matrix for J48 FCM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	9	0	1	0
Cluster 2	0	24	0	2
Cluster 3	0	0	29	0
Cluster 4	0	1	0	42

**Table 22: Confusion matrix for J48 FCM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	29	0	0	0	0
Cluster 2	0	22	0	1	0
Cluster 3	0	0	18	2	0
Cluster 4	0	4	0	22	0
Cluster 5	1	0	0	0	9

**Table 23: Confusion matrix for J48 FCM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	16	0	2	0	0	0
Cluster 2	0	21	0	0	0	0
Cluster 3	0	0	13	0	0	0
Cluster 4	0	0	0	18	1	0
Cluster 5	0	4	0	4	19	0
Cluster 6	0	0	1	0	0	9

**Table 24: Confusion matrix for J48 FCM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	22	0	0	0	0	5	0
Cluster 2	0	9	1	0	0	0	0
Cluster 3	0	1	18	0	0	0	0
Cluster 4	0	0	0	10	0	0	0
Cluster 5	1	0	0	0	20	0	0
Cluster 6	1	0	0	0	0	18	0
Cluster 7	0	1	0	0	0	1	0

**Table 25: Confusion matrix for J48 FCM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	10	0	0	0	0	0	0	0
Cluster 2	0	9	0	0	0	0	0	0
Cluster 3	0	0	15	3	0	0	0	1
Cluster 4	0	0	0	16	0	0	0	1
Cluster 5	0	0	0	0	20	0	0	0
Cluster 6	0	1	0	0	0	19	0	0
Cluster 7	0	0	0	1	0	1	0	0
Cluster 8	0	0	1	1	0	0	0	9

**Table 26: Confusion matrix for J48 XM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	10	0	0	1
Cluster 2	0	17	1	0
Cluster 3	0	0	49	1
Cluster 4	2	0	0	27

**Table 27: Confusion matrix for J48 XM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	3	0	0	0	0
Cluster 2	0	17	1	0	0
Cluster 3	0	0	50	0	1
Cluster 4	0	0	0	16	0
Cluster 5	0	0	0	0	20

**Table 28: Confusion matrix for J48 XM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	3	0	0	0	0	0
Cluster 2	0	32	2	0	0	0
Cluster 3	0	0	30	1	1	0
Cluster 4	0	0	0	12	0	0
Cluster 5	0	0	2	0	16	0
Cluster 6	0	0	0	0	0	9



**Table 29: Confusion matrix for J48 XM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	3	0	0	0	0	0	0
Cluster 2	0	12	0	0	0	0	1
Cluster 3	0	0	24	0	1	0	0
Cluster 4	0	0	0	13	0	0	0
Cluster 5	0	0	4	0	13	0	0
Cluster 6	0	0	0	0	0	9	0
Cluster 7	0	0	0	1	0	0	27

**Table 30: Confusion matrix for J48 XM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	3	0	0	0	0	0	0	0
Cluster 2	0	12	0	0	0	0	1	0
Cluster 3	0	0	24	0	1	0	0	0
Cluster 4	0	0	0	12	0	0	0	0
Cluster 5	0	0	2	0	16	0	0	0
Cluster 6	0	0	0	0	0	3	0	0
Cluster 7	0	0	0	0	1	0	27	0
Cluster 8	0	0	0	0	0	0	0	6

**Table 31: Confusion matrix for RBFN FCM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	9	0	1	0
Cluster 2	0	26	0	0
Cluster 3	0	0	29	0
Cluster 4	0	1	2	40

**Table 32: Confusion matrix for RBFN FCM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	29	0	0	0	0
Cluster 2	2	20	0	1	0
Cluster 3	0	0	20	0	0
Cluster 4	0	0	0	26	0
Cluster 5	1	0	0	0	9

**Table 33: Confusion matrix for RBFN FCM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	17	0	1	0	0	0
Cluster 2	0	18	0	1	2	0
Cluster 3	2	0	11	0	0	0
Cluster 4	0	0	0	18	1	0
Cluster 5	0	0	0	0	27	0
Cluster 6	0	0	1	0	0	9

**Table 34: Confusion matrix for RBFN FCM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	25	0	0	0	0	2	0
Cluster 2	0	10	0	0	0	0	0
Cluster 3	0	4	15	0	0	0	0
Cluster 4	0	0	0	9	0	0	1
Cluster 5	3	0	0	0	18	0	0
Cluster 6	1	0	0	0	0	18	0
Cluster 7	0	0	0	0	0	1	1

**Table 35: Confusion matrix for RBFN FCM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	9	0	0	0	0	0	1	0
Cluster 2	0	9	0	0	0	0	0	0
Cluster 3	0	0	17	1	0	0	0	1
Cluster 4	0	0	1	14	0	0	0	2
Cluster 5	0	0	0	0	19	0	0	1
Cluster 6	0	1	0	0	0	19	0	0
Cluster 7	0	0	0	0	0	0	2	0
Cluster 8	0	0	0	0	0	0	0	11

**Table 36: Confusion matrix for RBFN XM 4**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	11	0	0	0
Cluster 2	0	16	2	0
Cluster 3	0	2	48	0
Cluster 4	4	0	0	25

**Table 37: Confusion matrix for RBFN XM 5**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Cluster 1	3	0	0	0	0
Cluster 2	0	16	2	0	0
Cluster 3	0	1	48	1	1
Cluster 4	0	0	0	16	0
Cluster 5	0	0	1	6	13

**Table 38: Confusion matrix for RBFN XM 6**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Cluster 1	3	0	0	0	0	0
Cluster 2	0	31	3	0	0	0
Cluster 3	0	3	28	1	0	0
Cluster 4	0	0	0	12	0	0
Cluster 5	0	0	0	1	17	0
Cluster 6	0	0	0	0	0	9

**Table 39: Confusion matrix for RBFN XM 7**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 1	3	0	0	0	0	0	0
Cluster 2	0	10	0	1	1	0	1
Cluster 3	0	0	25	0	0	0	0
Cluster 4	0	0	0	13	0	0	0
Cluster 5	0	0	0	2	15	0	0
Cluster 6	0	0	0	0	0	9	0
Cluster 7	0	0	3	2	0	0	23

**Table 40: Confusion matrix for RBFN XM 8**

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Cluster 1	3	0	0	0	0	0	0	0
Cluster 2	0	11	0	1	0	0	1	0
Cluster 3	0	0	24	0	1	0	0	0
Cluster 4	0	0	0	12	0	0	0	0
Cluster 5	0	0	0	0	18	0	0	0
Cluster 6	0	0	0	0	0	3	0	0
Cluster 7	0	0	3	2	0	0	23	0
Cluster 8	0	0	0	0	0	0	0	6