Enhancing Customer Relationship Management System using Data Classification and Data Mining

Sanket Sandip Ranaware
Department of Computer Engineering
Pune Institute of Computer Technology Pune, India

Dr. Girish P. Potdar
Department of Computer Engineering
Pune Institute of Computer Technology Pune, India

Abstract

To be successful in any business, it is necessary to meet the current market conditions and requirements. The competition in any field is growing day by day, hence one has to adopt to the changes to keep growing with the business. It is not only required to attract good number of customers but also retain them over the life cycle of the business. Many organizations, institutions or industries always keep focus on how they can improve their customer support. So there must be any methodology that help such organizations to have their customer for long time. Customer relationship management is term which helps organizations to acquire customers and maintain them. In this paper, educational institutes are considered as a case study. Emphasis is given on building a customer support system, which can be effectively put in use for attracting good number of students. Proposed customer relationship management system assist organizations to establish an effective communication with their prospect student and guide them through the whole admission process. To implement the effective customer relationship management system mobile based recommendation is proposed here. In this dissertation prospect student's records are basically classified and based on classified records recommendations are provided to user, which will help user to communicate with prospect students.

Keywords: Customer Relationship management, Data Classification, Decision Tree Classification, Recommendation systems

I. INTRODUCTION

Customer Relationship management (CRM) is basically used to denote the activities which are used for managing and analyzing customer interactions and respective data throughout the customer lifecycle for improving business relationships with customers and retaining customers for long duration. In our concern, Educational organizations are collecting the student's data from various sources like social media, offline sources etc. Using such data, organizations need to establish the communication with the prospect students or we can say leads. Effective CRM will help educational organizations to acquire more students. Sales management and Digital marketing are two important parts of CRM. These two helps organization to advertise themselves to the students and maintain effective communication with them, so that more number of students get admitted.

For establishing effective CRM with leads, organizations must need to effectively deal with the all collected lead data. Counselor who is the user of platform must be able to contact with all leads in the system and ensure most of them get enrolled into the organization. Amount of data present in the database is large and collected from various sources. So data must be arranged in such manner that counselor will get the right information at the right time.

Basically, lead's data is maintained by using different status values i.e. leads are managed with specific state in admission process, some checkpoints are set for each step in the admission process. There are multiple counselors present there which will handle different leads in different state for example, new leads might have entertained by counselor A, leads which are in the state of taking admission and paying fees will be entertained by counselor B. So to distribute such data among counselor there must be a mechanism in the system which will classify the data and then provide it to the respective counselor to entertain that student and let the lifecycle of lead continues till it get admitted in the organization.

Similarly, here we are looking for mobile application for this platform, we also need to consider amount of resource power and limitations we have with mobile devices. So by considering the capacity of mobile devices we proposed a model for this system that will effectively implement CRM and help counselors to always get record of relevant leads and establish effective communication with leads. Lead data classification will be most crucial part in this system.

The process of sorting and categorizing of the data into various categories is known as Data Classification. For various business needs, data classification allows separation of the data. It is basically data management process.

A. Decision Tree Classification

Decision trees are also called as Classification trees. Following are the several reasons, why decision trees are used in data mining environment,

- Decision trees help to understand the resulting classification model easily.
- To construct a decision tree there is no need to provide any input from the analyst.
- The Decision tree classification model has more accuracy than the other models.

– It can be constructed using fast, scalable algorithms from very large training databases.

There are basically three types of nodes present in decision tree, Root node, internal node, and Leaf node. Intermediate node is labeled with splitting attribute, which defines the splitting condition. Leaf node is labeled with class label. Decision trees are built in many applications using Hunts Algorithm. These are built by Greedy method.

II. LITERATURE SURVEY

In data mining data classification has an important place and role. There are several number of Data classification algorithms are present. These algorithms use different techniques to classify data according to their application. Different algorithms have advantages over another in different conditions i.e. each classification algorithm may perform differently depending on where it is going to be applied and in which form data is present. Classifiers are built across multiple databases, tables by using Data classification which are driven by applications from various domains. Traditionally all the databases are integrated into single and them the respective algorithm were applied while performing data mining on multiple databases. But the problem occurs in case of huge dataset [1]. So to tackle such problem Tahar Mehenni, Abdelouahab Moussaoui proposed classification approach in [1] which is based on the decision tree. In proposed approach after integrating multiple relations, it builds multi-relational decision tree. To build the multi-relational tree author used the method which predicts the most useful links in tree. For that he used regression model in his method.

There are many classification models proposed [3]: neural networks, genetic algorithms, Bayesian methods, log-linear and other statistical methods, decision tables, and tree-structured models so called classification trees. Classification trees, also called decision trees.

Decision trees are used in various applications, like Machine learning, Decision Support Systems for classifying and categorizing data. Bahareh Bina et.al in [2], used decision tree classifier as base learner for dependencies within a single table. Author opted decision tree because; Decision tree learners mostly perform better than other methods. Other well researched methods are also there for learning trees which provide class probability estimates in their leaves [4, 5, 6]. In [2], proposed classification model target table contains the class attribute. The set of tables here are basically joined to the target table via a chain of foreign key's links. Author defines a cross table Naïve-bayes assumptions, according to which different join tables are independent given the class label. To allow different join tables to contribute the classification Log linear classification model is extended here [2].

Vapnik introduced Support Vector Machine (SVM) as a kernel based machine learning [3]. Its properties like great generalization capability and discriminative power have attracted the attention of data mining, pattern recognition, and machine learning communities in the last years [3]. Due to its properties SVM is used as a powerful tool for solving practical binary classification problems and regression. But its main disadvantage is that it demands to solve quadratic programming problem, which is computationally expensive. Solving quadratic programming problems while dealing with huge datasets becomes impractical, because amount of memory and time invested in it is between $O(n^2)$ and $O(n^3)$[3]. In [3], author proposed a novel method to reduce the size of the data sets based on decision tree. Each disjoint region discovered by a decision tree is used to train a SVM.

Basically, while classifying the data in information systems, some types of uncertainty may occur, if the class boundaries of objects are not defined. To overcome this problem the most common solution is fuzzy sets. Yauheni Veryha in [7], proposed a framework for implementing a fuzzy classification using conventional SQL querying. Like the conventional non-Fuzzy classification, use of SQL queries and fuzzy classification provides easy-to-use functionality for data extraction. Advantage of proposed approach is that it provides high flexibility for data analysis. Extra layer of data security features because of an additional view-based data layers that hide numerical values from users [7].

While classifying multiple records not only improving the efficiency of system but insuring the accuracy of classification is important. Yun Li et.al. Proposed approach in which, they sort the tables in multi-relational database according to their contribution. And then they delete some tables which have a little effect on the classification. Then to improve the efficiency and accuracy of the classification and they traverse the remaining tables [8].

Weiling Cai et.al. Proposed two step nonlinear classifier which is based on Fuzzy Relational Classifier. In proposed approach, initially the unsupervised fuzzy C-means (FCM) is performed to explore the underlying groups of the given dataset. Then a fuzzy relation matrix indicating the relationship between the formed groups and the given classes are constructed for subsequent classification [11]. Fuzzy relational classifier has some disadvantages. Robustness is not present there, which is important for a classifier. It does not perform well with non-spherical distributions. The Fuzzy Relational Classifier is sensitive to improper class labels of the training sample [11]. Weiling Cai et.al. proposed Robust Fuzzy Relational Classifier to overcome and mitigate all of the above disadvantages with maintaining original advantages of Fuzzy relational classifier.

III. PROPOSED SYSTEM

As counselor will be working from mobile devices or we can say from smartphones, the computing power will be very low to perform direct operations on the main database with large records. While designing the system it is very much important to put very less load on the mobile devices with maintaining performance and user convenience. So with keeping this constraint in mind we proposed Three-tier Client Server model. Following figure 1 shows the architecture of proposed system. Upper tier will

contain main global database. The data collected from various sources will be stored in it. This data is unclassified. Second tier contains the temporary database which will contain the classified data. This model will provide the classified data to the counselor's mobile devices. Similarly we use here Decision tree classification, because it implemented using greedy method, and in our concern with application, we need to consider the current state of the lead and based on that classify the lead data. Proposed system architecture is shown in figure 1.

A. Lead Lifecycle:

During lifetime of lead in the system, Lead goes through various states. This states basically helps counselor to identify where the lead is in the admission process. And according to state of leads they are distributed to different counselors or we can say system recommends it to counselor.

In this system, lead's data maintains its state like- made an enquiry, form issued, form submitted, Contacted, Came Online, Personal visit, take Follow up, fees pending, fees paid, etc. So each lead has an attribute named lead_Status that indicates this values and help counselor to manage each lead. Each lead lifecycle starts with New-Enquired state where he enters in to system to closed state where lead exits the system. So here we've to classify this all leads based on the state of the lead.

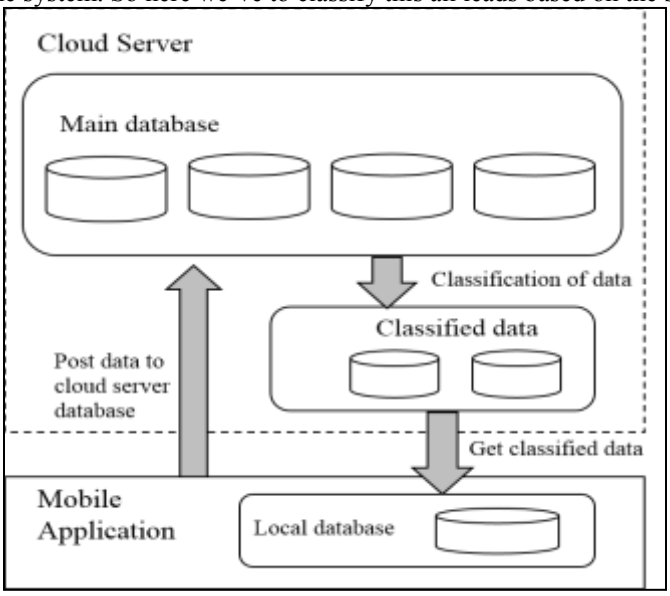

Fig. 1: Proposed Architecture

B. Data Classification:

As mentioned previously data coming from different sources is not managed or classified, due to which system has to search entire database for specific leads to assign to specific counselor. So unnecessarily it will traverse the whole database and takes more amount of time. So to reduce the unnecessary traversal of whole database records, we can classify the lead records into four categories "New leads, Working leads, Enrolled leads, Closed leads".

For classifying the records, we maintain status id to each lead. For different state of lead will have the different status id.

So what we need, is to classify the data into categories. As we have lead data with their status in the system. We will store this classified data into four temporary databases i.e. New lead's database, working leads database, Enrolled leads database, Closed leads database.

So we will deal with status id of lead, we will use this values to classify the lead data. We will use Decision tree classification for classifying this data. We have status of leads like Enquiry made, Form issued, Form submitted, Contacted-via mail, Contacted-via phone call, Came online, Enrolled-Fees pending, Document pending, etc. Basically we will classify the data into four categories with respective labels as below:

- New leads will contains leads with status id Enquired, Form issued, Form submitted and all other leads that enter into system.
- Working leads will contain leads with status Contacted, Came online, Personal visit, Follow up and those who are in process of admission.
- Enrolled leads, leads those have reached at the state of taking admission and have status like Fees pending, Document pending.
- Closed leads, leads having status closed and who have advanced to this level by simply clearing all the status.

Above conditions for classification are given, now we need to check the lead status and separate the data at the initial stage. Now this classified data we will store in the temporary databases on server itself. We will store records for each i.e. New leads, working leads, Enrolled leads, Closed leads in separate relations. So Counselors which are appointed to handle New leads will

only access to relation containing New leads, rather than traversing the whole records. Similarly other counselors will access respective classified records from temporary relations. This will definitely improve the system performance as it is very much efficient to access small data source than accessing whole database. So we can see Classification of records plays important role as classified records will help system to assign or recommend different leads to different counselors automatically. For classification, we'll use Decision Tree approach and will enhance it by recommendation system which will recommend leads to the counselors according to their state in lead lifecycle.

C. *Mathematical Model:*

Let, 'S' be a system such that,
$S = \{s, e, X, Y, F \mid \phi\}$
Where,
s = initial state
e = End state
X = Input of the system
Y = Output of the system
F= Main approach resulting into outcome 'Y'
ϕ = Constraint
Now,
s = Collection of whole records of prospect student; let's say leads
Initial state of system contains the unclassified records of leads.

$$R = \{r_1, r_2, \ldots, r_n\}$$

$$e = \{List_{classified}, List_{recommend}\}$$

End state contains the classified records and recommended lead list.

$$X = \{R\}$$

Here input is all lead records "R" given to the system.

$$Y = \{List_{classified}, List_{recommend}\}$$

Output is classified lists and recommended lead list provided to user.
Now, we define functions which are useful for our main approach.

$$F = \{F_{classify}, F_{recommend}\}$$

Where F is set of functions

1) For Classification of Lead Records

- $List_{classified} = F_{classify}(R)$

For each in record 'r_i' in Records 'R'
If (s_i = new OR s_i=new_re-enquired)
$C_{new} = C_{new} + r_i$
Else if (s_i = walkin OR s_i = personal_visit)
$C_{walkin} = C_{walkin} + r_i$
Else if (s_i = callback)
$C_{callback} = C_{callback} + r_i$
Else if (s_i = enroll)
$C_{enroll} = C_{enroll} + r_i$
Where, $List_{classified} = \{C_{new}, C_{walkin}, C_{callback}, C_{enroll}\}$
s_i = status in record r_i
C_{new} = Set of leads with new status
C_{walkin} = Set of leads with status walkin
$C_{callback}$ = Set of leads with status callback
C_{enroll} = Set of leads with status enrolled
For recommendation of lead records according to counselor activity:
$List_{recommend} = F_{recommended}$(Classified_list, Date_parameter, priority)
Input to $F_{recommended}$ function is classified list.
$R_{todays} = F_{recommended}(C_{new}$, current_date, any_priority)+$F_{recommended}(C_{walkin}$, current_date,any_priority)+$F_{recommended}(C_{callback}$,current_date,any_priority)+ $F_{recommended}(C_{enroll}$, current_date, any_priority);
$R_{missed} = F_{recommended}(C_{new}$, previous_date, any_priority)+$F_{recommended}(C_{walkin}$, previous_date,any_priority)+$F_{recommended}(C_{callback}$,previous_date,any_priority)+ $F_{recommended}(C_{enroll}$, previous_date, any_priority);
$R_{upcoming} = F_{recommended}(C_{new}$,upcoming_dates,any_priority)+$F_{recommended}(C_{walkin}$,upcoming_date,any_priority)+$F_{recommended}(C_{callback}$,upcoming_date,any_priority)+$F_{recommended}(C_{enroll}$,upcoming_date, any_priority);
$R_{highpriority}=F_{recommended}(C_{new}$,any_dates,high_priority)+$F_{recommended}(C_{callback}$,any_dates,high_priority)+$F_{recommended}(C_{walkin}$,any_dates,high_priority)+$F_{recommended}(C_{enrolled}$,any_dates,high_priority)

Where, $List_{recommend}$ = { R_{todays} , $R_{,missed}$, $R_{,upcoming}$, $R_{highpriority}$ }
current_date= it is the todays date at the runtime.
previous_date= the date other than present and future date.
upcoming_date=the date other than todays and past date.
any_priority= task with any priority.

Figure 2 represents mapping of mathematical model into system. Here as a Decision tree algorithm for classification, R will be the root node, which is basically set of all records from main database. We are classifying R into mainly four categories. Then the functions fnew, fworking, fenrolled, fclosed are used for classifying the R.

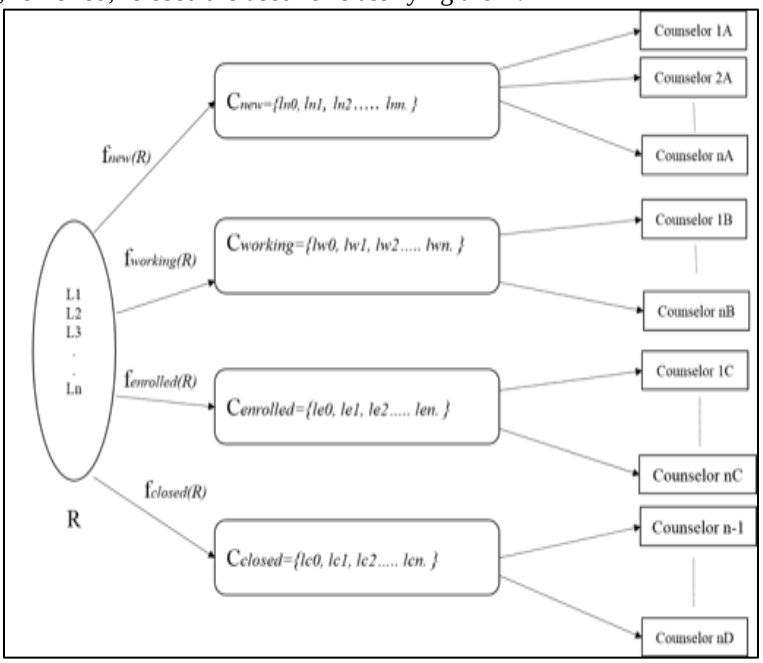

Fig. 2: Mapping of Mathematical model

`They contain conditions to compare status of the records. So basically these functions will act as the intermediate nodes. At the end we get classified records as Cnew, Cworking, Cenrolled, Cclosed. . So these records will be the leaf node in the decision tree. These classified records will now help us to assign leads to respective counselors. All counselors that are entertaining specific leads, let the counselor with New leads will access only classified records of new leads instead of accessing all records. So this will reduce access time and will improve performance and response time.

IV. ALGORITHM SOLUTION

A. Decision Tree Algorithm:

This algorithm is a widely used for classification technique due to its simplicity.

- Structure of this tree is divided into three nodes: -
  1) A root node which does not have any incoming edge but more outgoing edges.
  2) Internal nodes consist of exactly one incoming edge and two or more outgoing edges.
  3) Leaf nodes, have exactly one incoming edge but no outgoing edges.

1) Algorithm 1 Decision Tree Classification
Input:
Dataset of Prospect student (lead) records.
Output:
Classified data into classes as New, Callback, Walk-in, Enrolled .
Procedure:

1) Step 1: Creating Root node c = lead as a starting point of classification.
2) Step 2: Check if attributes as

If lead_status= new or new-re-enquired.
Then add lead record to 'New' class
If lead_status = ent-callback or post-callback or deferred-callback.
Then add lead record to 'Callback' class

If lead_status = walk-in or personal_visit
Then add lead record to 'Walk-in' class
If lead_status = enrolled
Then add lead record to 'Enrolled' class

3) Step 3: Repeat step 2, 3 till the last record.
4) Step 4: Now cache this lists or classes into class objects as Array lists.
5) Step 5: Terminate tree.

B. Data Parsing and Generating Recommendation List:

- In the class object classified records are stored as objects. These objects are stored int the form Key-value pair. So while accessing this records class object name and key name needed to access the cached records.
- Once data is fetched for accessing from objects, each class of classified records is traversed and action date and time regarding each activity of each lead is checked. Priority field is also checked of each lead record.
- Based on action date and time, each record is stored in to recommendation lists as Today's, Missed, High priority and Online.

1) Algorithm 2 Data Parsing and Generating Key-Value pair

Input:
Cached classified lead records
Output:
Generate recommendation lead lists according activity action date and time.
Procedure:

1) Step 1: Fetch classified lead records from objects.
2) Step 2: Check for activity action date-time and priority of each record.
3) Step 3: Generate recommendation lists as

If activity_action_datetime = not todays or future and priority=any
Then add record into Missed list
If activity_action_datetime = todays and priority=any
Then add record into Todays
If activity_action_datetime = any and priority=high
Then add record into High priority list

4) Step 4: Store this records into object as cache

V. RESULT

A. Space Required:

System application on mobile requires 7.5 of space. Mobile database is not used in system. For caching purpose programming language object and classes are used.

B. Compilation Time:

Initially it takes 20 to 30 seconds to compile and run the application on mobile phone.

C. Running Time:

Time required to launch and run the application on mobile is mostly dependent on internet connection. On the cellular 3G connection application requires 2-3 seconds to launch and run. While on slower connections like 2G, it takes 4-5 seconds to launch and run the application.

D. Comparison with Existing Web Application:

Table – 1
Shows the comparison between proposed system and existing web application.

	Proposed Mobile Application	*Existing Web Application*
Space required	*7.5 MB*	*550 MB*
Time to load Lead List	*6-10 seconds*	*15-20 seconds*
Time to load Counselor Dashboard	*6-10 seconds*	*15-20 seconds*
Time to Add new lead	*2-5 seconds*	*5-7 seconds*
Time to Communicate with student through SMS	*4-5 seconds*	*5-6 seconds*
Time to Communicate with student through Email	*4-5 seconds*	*5-6 seconds*
Time to Communicate with student through Voice calls	*3-4 seconds*	*10-15 seconds*

VI. CONCLUSION

While dealing with the large dataset, relational records it is very important that system should access it not only in efficient manner but also with accuracy and good performance. Data Classification plays important role as it helps in sorting and managing the datasets and using that classified records we can make system to access only required records instead of whole records. Here in our case, if application tries to access all the lead records each time, then it will lower the performance of the system. So, it is better to classify the datasets into required classes and then allow access to particular counselor to the respective classified classes. So by this way, instead of traversing all the records each time, application will access only relevant class of records. This will improve efficiency of the system and improve the performance of the mobile application.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Principal Dr. P.T. Kulkarni and my dissertation guide and Head of Computer Department Dr. G. P. Potdar for their precious guidance, continuous support, valuable suggestions in my work and most valuable time lent as and when required. I also like to

Table 1 Comparison with existing Web application

thank to my family for their continuous support and encouragement at every stage of my life.

REFERENCES

[1] Tahar Mehenni, Abdelouahab Moussaoui, "Data mining from multiple heterogeneous relational databases using decision tree classification," in ELSVIER Journal May 2012, pp.1768-1775.
[2] Bahareh Bina, Oliver Schulte, Branden Crawford, Zhensong Qian, Yi Xiong, "Simple decision forests for multi-relational classification" in ELSEVIER Journal, December 2012, pp. 1269-1279.
[3] Jair Cervantes, Farid García Lamont, Asdrúbal López-Chau, Lisbeth Rodríguez Mazahua, J. Sergio Ruíz "Data selection based on decision tree for SVM classification on large data sets," in ELSEVIER Applied Soft Computing Journal, September 2015, pp. 787-798.
[4] A. Van Assche, C. Vens, H. Blockeel, S. Dvzeroski, First order random forests: Learning relational classifiers with complex aggregates, Machine Learning , January 2006, pp. 149-182.
[5] S. Kramer, N. Lavrac, P. Flach, Propositionalization approaches to relational data mining, in: Relational Data Mining, Springer, August 2000, pp.262–286.
[6] A.Y. Ng, M.I. Jordan,"Discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in: NIPS, 14, MIT Press, 2001, pp. 841–848.
[7] Yauheni Veryha,"Implementation of fuzzy classification in relational databases using conventional SQL querying", in ELSEVIER Infromation and Software technology Journal, November 2005, pp. 357-364.
[8] Yun Li, Luan Luan, Yan Sheng, Yunhao Yuan, "Multi-relational Classification Based on the Contribution of Tables", in IEEE AICI 2009, 370-374.
[9] G. Suresh kumar, G. Zayaraz, "Concept relation extraction using Naïve Bayes classifier for ontology-based question answering systems", in ELSEVIER Journal of King Saud University – Computer and Information Sciences May 2014, 13-24 .
[10] Weiling Cai, Songcan Chen, Daoqiang Zhang, "Robust fuzzy relational classifier incorporating the soft class labels", in ELSEVIER Journal of Pattern Recognition Letters, August 2007, 2250-2263
[11] Hisao Ishibuchi, Tomoharu Nakashima, "Effect of Rule weights in Fuzzy Rule-Based Classification System" in IEEE Transactions of Fuzy Systems VOL.9, August 2001, 506-515.
[12] Qinghua Hu, Xunjian Che, Lei Zhang, David Zhang, Maozu Guo, Daren Yu,"Rank Entropy-Based Decision Trees for Monotonic Classification", in IEEE transactions on Knowledge and Data Engineering, VOL.24, November 2012, 2052-2053.
[13] Bin Yao, Feifei Li, Piyush Kumar, "K Nearest Neighbor Queries and KNN-Joins in Large Relational Databases (Almost) for Free", in IEEE ICDE Conference, November 2010, 4-1