A Graph Oriented Based Recommender System for Financial Products and Services
Olakunle Temitope, Awodele, O., Adekunle, Y.A., Eze, M.O., *Ebiesuwa Seun
Department of Computer Science, Babcock University, Nigeria

ABSTRACT

Recommender Systems are intelligent applications designed to assist the user in a decision-making process whereby user wants to choose one item amongst the potentially overwhelming set of alternative products or services. This research is aimed at developing an intelligent recommender system that provides high quality recommendations in the financial domain. Hashed and anonymized datasets (which are account statements) were acquired from online sources and bank customers. The acquired data was pre-processed using the Microsoft Excel 2016 and WEKA 3.8.3 data mining software. The K-nearest neighbor (KNN) algorithm was used to classify the dataset and train the model. The trained model was used to develop a recommender system using the Java 2 platform Enterprise Edition (J2EE). For effective management of the data and consideration of rapid increase in data growth, a graph-oriented database approach was proposed and utilized. The database management system used was the Neo4j. From the evaluation of the algorithms implemented in the recommender system taxonomy, the KNN algorithm recorded the best performance building the model in 0.3 seconds with an accuracy of 89.8%. The fuzzy decision tree algorithm performed second best building the model within 0.48 seconds with an accuracy of 62.8%. The decision table algorithm performed poorly building the model in 3.9 seconds with an accuracy of 53%. However, the baseline accuracy of the dataset used was evaluated to be 62.75% of accuracy in 0.4 seconds. It is therefore recommended, as proposed in this study that the graph technology be used in developing recommender systems especially for institutions with massively growing data like the financial institutions. In addition, bank products should be classified and targeted towards customers in order to bolster their level of involvements and improve financial inclusion. With a targeted product, customers will be more willing to opt-in if products are suitable and within financial reach. This will help financial institutions earn more and the customer’s financial power will also be strengthened.
Introduction

In Google news, 38% of the total views are the result of recommendations; similarly, 60% of the rented movies from Netflix come from recommendations and more than that Amazon sales percentage due to recommendations are 35%. Successful integration of recommendation system by online companies like Amazon, eBay, Flipkart amongst others impelled the research community to avail similar benefits in financial domain to recommend product and services (Lim, 2015). Therefore, recommendation systems are considered an expedient factor in business nowadays. The aim of all recommender systems is to provide recommendation that will be favourably evaluated and accepted by its users. Recommender systems made filtering of information easy and simple for its users because recommender systems use different information retrieving techniques to find and recommend items of interest to its users. Therefore, if a recommender system is able to recognize the intent and requirements that a user expresses in the form of queries, it can generate more valid recommendations (Gulzar, et al., 2018).

Review of related work

Several recommendation frameworks have been proposed over the years and a comparison across their experimental results is necessary to evaluate the best algorithm. Aguilar et al (2016) described a general framework for a recommender system that extended the concept of the traditional knowledge-based approach. As shown in figure 2.1, the framework Intelligent Recommender System (IRS) is defined by learning algorithms, knowledge representation mechanisms, and reasoning motors using five knowledge models: users, items, domain, context and criticisms. The framework was implemented using Fuzzy Cognitive Maps (FCMs) and tested with specialized criteria linked to the use of the knowledge. FCMs are based on Cognitive Maps theory which infers modelling of systems based on concepts that describe the core characteristics of the modelled system and the relationship between them. The IRS has four sections: Knowledge Modelling, Knowledge Acquisition, Reasoning Mechanism and Criticality System.

Figure 1 Intelligent Recommender System Architecture (Aguilar et al., 2016)
Thus, the IRS uses the knowledge available without a degrade in performance at any instance to carry out recommendations.

Jallouli, Lajmi and Amous (2017) in their research proposed a framework that clarified principles of a RS and detailed each step. The framework bridges the gap of representation of data and recommendation process due to various types of input structure by allowing an input of various types into the framework. They proposed a conceptual framework (shown in figure 2.2) that has three phases - Input, recommendation algorithm and output. The input section involves dataset collection, data transformer and social calculator. The algorithm classifies based on the type of input data into baselines, social based, contextual or Socio Contextual approaches. Recommendations based on history is the output of the framework. Summary of algorithms in the algorithm section follows: General baseline recommender, Baseline based Average recommender, Baseline based collaborative filtering, Baseline based context recommender, Baseline based top-N recommender, Social Based recommender systems, Contextual based recommender systems and Social-contextual recommender systems.

Thus, the presented conceptual framework provides efficient data transformation irrespective of the input type, use of existing algorithms and easy comparison amongst algorithms.

Recommender System was used in the Education domain to suggest and guide a learner in select appropriate courses per their requirement. The objective of Gulzar, Leema and Deepak (2018) study work was to design and develop a hybrid Recommender system that can be integrated to enhance the effectiveness of any E-learning system, to ease information access and to provide personalization to learners. The Hybrid methodology was used along with ontology to retrieve useful information and make accurate recommendations. The framework is shown in figure 2.3. The techniques used are N-gram query classification and expansion-based information retrieval for course recommendation along with ontology support. It was discovered that the approach was helpful to learners to increase their performance and improve their satisfaction level.
By the rapid growth of information technology, the financial industry changed significantly in the last decade. With the spreading of online payment solutions in various devices, a massive online data flow appeared in bank systems centralizing data from multiple domains. Banks are forced to change technologies that is capable to handle big data and exploit business value from the massive information flow.

Table 2.1: Summary table of closely related works

<table>
<thead>
<tr>
<th>Author</th>
<th>Problem Identified</th>
<th>Method</th>
<th>Result</th>
<th>Strength</th>
<th>Gaps Identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aguilar et al., (2016)</td>
<td>Recommender systems tend to use a traditional knowledge-based approach.</td>
<td>Fuzzy Cognitive Maps was used to develop a knowledge representation mechanism in the developed Intelligent recommender system.</td>
<td>Inferences and the reasoning mechanisms were used for induction and deduction while giving recommendation.</td>
<td>An intelligent approach of mining available information to deduce items to be recommended.</td>
<td>All available knowledge needs to be discovered for effective recommendation. This approach seems unsuitable for systems with some kind of incomplete dataset. Feedback mechanism to evaluate the satisfaction of user as regards provided recommendations.</td>
</tr>
<tr>
<td>Jallouli et al., (2017)</td>
<td>The authors identified the challenge of data representation in recommender systems and thus developed a framework that accepts diverse input structure.</td>
<td>Baseline based collaborative filtering approach was used in the development of the recommendation framework.</td>
<td>The framework bridges the gap of representation of data and recommendation process due to various types of input structure by allowing an input of various types.</td>
<td>The framework provides efficient data transformation irrespective of the input type.</td>
<td>Details about input data and recommendation algorithm were not well defined.</td>
</tr>
<tr>
<td>Gulzar et al., (2018)</td>
<td>The need for learners to select appropriate courses as per their requirements. Thus, the researchers proposed a recommender system that can aid e-learning system and provide personalized recommendation to users.</td>
<td>The Hybrid methodology was used along with ontology to retrieve useful information and make accurate recommendation. The specific techniques used are the N-gram query classification and expansion-based information retrieval approach.</td>
<td>It was discovered that the approach was helpful to learners to increase their performance and improve their satisfaction level.</td>
<td>Information were gathered from domain experts to equip the ontology tool component of the system.</td>
<td>Despite the hybrid approach utilized, the recommendation is somewhat dependent on the explicit feedback from users.</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Year</td>
<td>Description</td>
<td>System/Methodologies</td>
<td>Findings/Impact</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Felfering et al.</td>
<td>2015</td>
<td>general-purpose knowledge-based recommender systems with intelligent user interface, which can be flexibly applied on various financial products.</td>
<td>knowledge-based algorithms over the conventional collaborative- and content-based filtering. The system was able to recommend multi-criteria-based financial decisions to customers. The interactive user interface offered to user.</td>
<td>Highly dependent on explicit feedback from users.</td>
<td></td>
</tr>
<tr>
<td>Musto et al.</td>
<td>2015</td>
<td>complex task of recommending financial investment strategies.</td>
<td>The framework proposed combines case-based reasoning with a novel diversification strategy to support financial advisors. They used a combination of Basic Ranking, Greedy Diversification and Financial Confidence Value (FCV) techniques. The result showed the yield obtained by recommended portfolios overcame that of portfolios proposed by human advisors in most experimental settings while meeting the preferred risk profile.</td>
<td>Using the case-based approach together with a novel diversification strategy to support financial advisors. Evaluation of recommended products was not catered for.</td>
<td></td>
</tr>
<tr>
<td>San Miguel et al.</td>
<td>2015</td>
<td>The challenges of personal data collection, integration, retrieval, and identity and privacy management. Thus, the authors proposed a framework that addresses this.</td>
<td>They design a data framework architecture, which is capable to integrate both public and private data dealing with privacy issues. Comprehensive Personal Data Framework (PeDF) for loan recommendation via social network that allows service providers to share and exchange personal data and knowledge about users, while facilitating users to decide who can access which data and why.</td>
<td>Dependent on information that are not financial based.</td>
<td></td>
</tr>
<tr>
<td>Guo et al.</td>
<td>2016</td>
<td>Effective allocation of money across different loans by accurately assessing the credit risk of each loan.</td>
<td>an instance-based credit risk assessment model, which has the ability of evaluating the return and risk of each individual loan. Experimental results revealed that the proposed model can effectively improve investment performances compared with existing methods in P2P lending.</td>
<td>Ability to make recommendations based on level of risk associated with loan application. The accuracy is dependent on the rating-based model.</td>
<td></td>
</tr>
<tr>
<td>Pazzani &amp; Bilus,</td>
<td>2007</td>
<td>Recommender system to be used in various domains so curb information overload.</td>
<td>Using the content-based filtering approach, the researchers recommends items based on the metadata of items in user history and other available items. The system was able to create profiles of users and their shared behavior and characteristics.</td>
<td>CBF algorithms can cope with the cold start problem and their recommendations are easy to explain by meta words. this method requires metadata and creates problems with individual interactions. The models strongly rely on the quality of metadata and they are usually less accurate.</td>
<td></td>
</tr>
<tr>
<td>Gigli et al.</td>
<td>2017</td>
<td>developed a recommender system for banking services that accepts implicit feedback from users</td>
<td>Bayesian Personalized Ranking algorithm, Matrix factorization method, Alternating Least Squares algorithm and Word2Vec algorithm were used to evaluate the recommender system. The result shows that the recommender system performs well with no popular item removed and poorly when some popular items are filtered out.</td>
<td>The ability for the system to capture feedback from the user implicitly. As popular items were filtered out, the reliability and accuracy of the recommender system reduces.</td>
<td></td>
</tr>
<tr>
<td>Adebayo et al.</td>
<td>2015</td>
<td>The need for bank staff to recommend products and services to current and prospective customers. Hence, they developed an expert system that will recommend these products based on customer information.</td>
<td>A number of customer service personnel were interviewed for expert knowledge, and related systems were examined in order to incorporating learnt ideas. The Prototype, horizontal and throwaway versions, software development life cycle model was adopted. Software development environment include WampServer, Adobe Dreamweaver, Hypertext Markup Language, Cascading Style Sheets, and Hypertext Preprocessor were used to develop the system.</td>
<td>Recommendations were provided based on the customers wants. Information and dataset were acquired explicitly from customers.</td>
<td></td>
</tr>
<tr>
<td>Farajan &amp; Mohammadi</td>
<td>2019</td>
<td>The need to analyse customer behavioural pattern and apply acquired information to develop new business strategies.</td>
<td>This study presents a new two-stage framework of customer behavior analysis that integrated a K-means algorithm and Apriori association rule inducer. The customers were divided into three profitable groups of customers according to their shared behavior and characteristics. Marketers then can infer the profiles of customers in each group and propose management strategies appropriate to each group.</td>
<td>This study provides a new method of analyzing bank databases. The k-means algorithm is a non-hierarchical classifier.</td>
<td></td>
</tr>
</tbody>
</table>
Methodology
An overview of recommender system conceptual model. Although, various works has been proposed for recommending a financial product and service using different RS taxonomy, this research focuses on recommending financial product and service by comparing various RS taxonomy algorithms.

Data pre-processing was done because some machine learning algorithms require the data to be in a specific form, other algorithm can perform better if the data is prepared in a specific way; some of the raw data were not be in the best format to best expose the underlying structure and relationships to the predicted variables. The representation of the data was done by the bank since the dataset is in spreadsheet format – excel. The tasks involved in the pre-processing were:

i. Cleaning – this involved:
   a. Fill in missing values (attribute or class value) by Ignoring the tuple if the class label was missing, Using the attribute mean (or majority nominal value) to fill in the missing value, Using the attribute mean (or majority nominal value) for all samples belonging to the same class and predicting the missing value by using a learning algorithm.
   b. Identifying outliers and smooth out noisy data through Binning - The attribute values were sorted and partition into bins which were later smoothened by bin means, bin median, or bin boundaries, Clustering by grouping values in clusters was also done and the outliers (automatic or manual) were detected and removed.
   c. inconsistent data were corrected using domain knowledge/expert decision.

ii. Data transformation: this involved;
   a. The normalization process which was about scaling attribute values to fall within a specified range and the scaling was done by using mean and standard deviation
   b. Aggregation was carried out by moving up the numeric attributes in the concept hierarchy.
   c. Generalization was done by moving up the nominal attributes in the concept hierarchy.
   d. Attribute construction was done by replacing or adding new attributes inferred by existing attributes.

iii. Data reduction
   a. Number of attributes were reduced through Data cube aggregation: roll-up, slice and dice operations were applied; irrelevant attributes were removed, valid attributes were selected, the attribute space was searched and principle component was analysed (numeric attributes only).
   b. Reducing the number of attribute values was done by binning (histograms): attributes were group into intervals (bins); and values were grouped into clusters and Aggregation or generalization
   c. Sampling was carried out to reduce the number of tuples

iv. Discretization and concept hierarchies were generated using the following approach
   a. Unsupervised discretization – the class variable was not used through Equal-interval (equiwidth) binning: splitting the whole range of numbers in intervals with equal size and Equal-frequency (equidepth) binning which use intervals containing equal number of values to split numbers.
b. Supervised discretization – This used the values of the class variable.
   i. Using class boundaries. values were sorted, breakpoints were placed between values belonging to different classes. Intervals that were too many were merge with equal or similar class distributions.

c. Concept hierarchies were generated by recursively applying partitioning and discretization methods.

The tools used for Data pre-processing are Microsoft excel 2016 and WEKA 3.8.3.

The data acquired from respondents were in pdf format while those from the internet were in comma separated value (CSV) format. Using an online .pdf to .xls converter, the pdf documents were converted and the identifiers were replaced with dummy data. The .xls file were further converted into csv and these enabled data merging and a single file for all the acquired data. Missing values were searched for manually and filled using expert judgement.

Using the balance filed, outliers in the data set were identified. Outlier data are data with extremely low or extremely high values. This was gotten by calculating the first quarter (Q1), third quartile(Q3), and interquartile range (IQR) in excel. Outliers = Q1 - 1.5(IQR) or Q3+1.5(IQR).

From the data acquired, Q1 has a value of 57,956.00, Q3 had a value of 708,552.81. the IQR gave the value of 650,596.81. Using the above formulas, the outliers were values below -917,939.22 and above 1,684,448.03.

Values greater or lower than these were seen as outliers as shown in Figure 3.2. outliers were identified so as to smooth out noisy data and to ensure data consistency.

![Figure 4: graph showing the range of average balance](https://escipub.comernational-journal-of-service-science-management/7)
RS has various taxonomy but four were selected for this research. The taxonomy and algorithms are:

a. Collaborative filtering: K-nearest Neighbour Algorithm
b. Knowledge Based: Fuzzy decision tree
c. Case based: Singular Value Decomposition

The processed data is loaded into the algorithms for classification based on inputs. The results generated from each RS Taxonomy algorithm were compared against each other and the optimal algorithm was used for recommendation. The recommender system consists of the rules generated from the optimal algorithm and the recommended products and service based on the input. The recommender system is a web-based system.

This node interacts directly with the benefit node which houses possible benefits bank and customers can obtain when recommended products are used. The higher the acceptance rate of products recommended, the higher the benefit. This approach was introduced as a means of evaluating the effectiveness and accuracy of recommendations made by the system.

**Figure 5 Model design of the system nodes.**

Data after loaded the Neo4J were used to develop the system. The system was developed using Java Enterprise Edition (Java EE). Limited efforts were put in the user interface as more efforts were put into the effective functionality of the system.

In creating the databases, some queries were executed.

**Flow of creating Account and Transactions nodes:** The queries below help to create the transaction nodes and load the data from the CSV file into the database.

LOAD CSV WITH HEADERS FROM 'file:///accountowner.csv' as accounts
Create (a:Account {
accountNumber: toInteger(accounts.AccountNumber),
accountName: accounts.AccountName,
accountTotalWithdraws: tofloat(accounts.TotalWithdraws),
accountTotalLodgements: tofloat(accounts.TotalLodgements),
closingBalance: tofloat(accounts.ClosingBalance),
closingBalance: tofloat(accounts.ClearedBalance),
unclearedBalance: tofloat(accounts.UnclearedBalance))
return a
LOAD CSV WITH HEADERS FROM 'file:///transactionQ1.csv' as btransactions
Create (t:Transactions {
date: btransactions.Date, description: btransactions.Description, referenceNo: btransactions.ReferenceNo, currencyType: tofloat(btransactions.Currency),
return t

Flow to generate relationship between nodes: The set of queries below help to establish the relationship and dependencies between the nodes.

match (a:Account), (t:Transaction)
where a.account_id = t.account_id
Create (a)-[:ACCOUNT_IN]->(t)
match (a:Bank), (t:Account)
where a.Id = t.bankid
Create (a)-[:HAS_ACCOUNT]->(t)
match (a:Bank), (t:Product)
where a.Id = t.bankid
Create (a)-[:HAS_PRODUCT]->(t)

Results and Conclusion.
In preprocessing the data, some of the attributes were adjusted to the appropriate value type. Using the filter option in the explorer window of the preprocessed tab in WEKA, account number and bank id attributes were converted from a numeric value to a normal value because they were only to be used as a means of easy identification of transactions. The balance attribute which was initially in a nominal state was also converted to numeric values. As shown in Figure 6, the arff viewed used to view the data converted from csv to arff before the preprocessing in weka. This allows the researcher identify the value type of attributes and help plan how to perform the needed attribute value conversion. Some conversions were done in excel while some were done using the filter feature as stated earlier and as shown in Figure 7.

Figure 6: Dataset viewed in the arff viewer.
Figure 7: Applying filters to the dataset attributes.

On the preprocessed view of the attributes. The instances were identified with color codes in relation to other attributes. Figure 7 shows the account numbers and the volume of instances.

The class attribute of the account numbers were used on the bank and amount attributes as shown in Figure 8.

Figure 8: preprocessed view showing the instances for the account number attribute.

Figure 9: preprocessed view of the bank using the accounted as the class attribute.
The debit and credit attributes were also applied on the account number as class attribute so as to view the rate of credit and debit on an account. This is shown in Figure 9. The debit transactions are in blue while the credit transactions are in red.

![Figure 9: preprocessed view of account numbers using the DRCi as class attributes.](image)

**Model evaluation for generating recommendation**

The recommender system adopted a hybrid approach as depicted in researcher’s model. This allows pertinent recommendations with a continuous improvement over time by gathering and using more users’ information. The algorithms selected were dependent on the performance of the evaluated algorithms from the RS taxonomy. For evaluating the result of the algorithms on the dataset, the baseline accuracy was determined. The baseline accuracy for the dataset is 62.75% as shown in figure 10.

The result obtained when the KNN algorithm was used on the dataset is shown in figure 4.8 below. The result shows that the KNN algorithm gives an accuracy of 89.8%. The value if K was set to ten because of the possible noise associated with the dataset as well as the volume of dataset. Following the theory where $k \to \infty$ and $n \to \infty$; $k/n \to 0$. i.e. the larger the number of instances(n), the larger the value of k, the more the error approaches minimum for the dataset (Chih-Min et al., 2014).

![Figure 10: result of the baseline accuracy for the dataset.](image)
The performance of the K-NN algorithm was evaluated an accuracy of 89.8% was recorded as shown in figure 11

For the Knowledge based component, the Fuzzy decision tree algorithm (J48) was evaluated. The result shows a 62.8% accuracy

Class P are Pamper users, class M are moderate users, class T are Transactor users, class H are heavy users. The results are shown in figure 13

Figure 12: Evaluation result when the KNN algorithm is used.

Figure 13: tree generated when evaluated using the fuzzy random tree (J48) algorithm

Figure 14: result of the J48 algorithm on the dataset.
The decision table algorithm had an accuracy of 53% as shown in figure 14.

Data feeding into the graph model

After loading the data - using the set of queries listed in section 3.6 - into the database model, the result is what is depicted in figures 16 and 17 below.

As shown in figure 4.14 below, the relationship between the various nodes in the database were also defined.

Figure 15 result of the evaluation using the decision table algorithm.

Figure 16 View of the Neo4J with the loaded data and object definitions.

Figure 17 View of the neo4j database after data had been loaded.
Figure 18: view of relationship between database objects.

**Business Rules and Mapping Recommendations**

Algorithm used to map interest rate and charges to the products recommended and accepted by user is depicted below.

Algorithm for the computing charges associated with products.

**Axiom 1:**
A fixed deposit Account product, this is short tenor. Tenor of between 30-90 days.

**Iteration 1:**
The average will be obtained. Let the Average transaction money be Avg.

If avg between 100000 and 499000 and min_avl_bal $< 50000
Then percentage is 9% && w_tax is 10%
Compute interest_rate

**Axiom 2:**
A fixed deposit Account product, this is short tenor. Tenor of between 60 days.

**Iteration 2:**
average will be obtained. Let the Average transaction money be Avg.

If avg $\geq 500000$ and min_avl_bal $< 100000$
Then percentage is 4% && w_tax is 10%
compute interest_rate

**Axiom 3:**
A fixed deposit Account product, this is short tenor. Tenor of between 30-90 days.

**Iteration 3:**
The average will be obtained. Let the Average money be Avg.

If avg $\geq 2000000$ and min_avl_bal $< 500000$
Then percentage is 13% && w_tax is 0%
compute interest_rate

**Axiom 4:**
A Treasury Bills and FGN Bonds Account product, this is long tenor. Tenor of 30 days.

**Iteration 4:**
The average will be obtained. Let the Average money be Avg.

If avg $\geq 5000000$ and min_avl_bal $< 50000$
Then percentage is 15% && w_tax is 0%
compute interest_rate

The feedback mechanism of the system was designed not to be explicit but rather implicit. When products are recommended to user and user takes/accepts the products, the system marks that particular product to the user. After the product tenure i.e. after user had enjoyed the benefit of the product, a rating score is ascribed to the product enjoyed by the user. For example. If a user gets the recommendation to subscribe for the fixed deposit product, once the system accepts this recommendation, the system flags the product for the user. Once the tenor of the
product is expired, the product is rated for the user based on the time duration. The shorter the time duration, the higher the rating. Rating ranges from 5 to 1 where 5 is rated highest.

**Conclusion**

This research had been conducted to develop an efficacious approach that will be suitable for recommending financial products and services to customers. Contrasting to the popularly used K means and fuzzy based algorithms, this research has investigated the suitability of the KNN algorithm as a result of the randomness observed in financial inflow and outflow of funds. Unlike other financial recommendation system that tends to associate demographic information of customer so as to aid the grouping of customers, the system developed intelligently recommends products and services to customer after analyzing the financial history of the customer. An implicit feedback mechanism is also included to further improve performance of the system as the data grows. The growth of financial data had also been a disadvantage for recommender systems since they tend to possess some form of latency as the data grows significantly. This is due to the dependency of the dependencies needed to use the database approach. With the graph technology proposed in this research, the dependencies would be eliminated and the significant growth in size would have a miniscule effect on the recommendation other than to improve the performance as a result of data availability.

**Contribution to Knowledge**

Unlike other previous studies on recommendation systems, this research used the graph-oriented database to develop the recommendation system. In addition, using this dataset (account statement), the K nearest neighbor clustering algorithm tend to have a better performance as against the commonly used k-means algorithm. By comparing artificial intelligence techniques for pattern creation, this research had presented a framework for intelligent recommendation for financial products and services. In addition, target based products could be offered by financial institutions to customers using this model. This will in turn improve customer satisfaction and involvement of customers in opting for financial based products. Generally, this research offers great benefits to the financial sector in maximizing available information, profiling clients for key products, recommend products to clients, improve products efficiency and wealth management.

**REFERENCES**


