

Global Journal of Artificial Intelligence (DOI:10.28933/GJAI)



A Model of Intelligent Recommender System With Explicit Feedback Mechanism for Performance Improvement

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 work focused on using users bank statements that explicitly shows inflow and outflow of funds. The dataset used is real and reliable because the use of non-reliable data in a recommender system causes users lack of trust in the system. However, the data collected were anonymized for privacy reasons. The recommender system was developed as a web application using Java programming language. Unlike other recommender systems, the graph-oriented database management system was used. 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. This work provides detailed descriptions of methods employed to proffer solutions to intelligent recommender system with explicit feedback mechanism. The methodology of this research work refers to the research approach adopted by the researcher to tackle the research problem as stated in earlier chapter. Since the efficiency and maintainability of any application is solely dependent on how the designs are prepared, this chapter describes the various processes, methods and procedures used to achieve set objectives and the conceptual structure within which the research was conducted

*Correspondence to Author:

Ebiesuwa Seun

Department of Computer Science,
Babcock University, Nigeria

How to cite this article:

Olakunle Temitope., Awodele, O., Adekunle, Y.A., Eze, M.O., Ebiesuwa Seun. A Model of Intelligent Recommender System With Explicit Feedback Mechanism for Performance Improvement. Global Journal of Artificial Intelligence, 2020; 2:6.



eSciPub LLC, Houston, TX USA.

Website: <https://escipub.com/>

Introduction

Machine learning has proven to be of great value in data mining problems especially where large databases contain valuable implicit regularities that can only be discovered automatically or in poorly understood domains. It involves searching a very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner (Michie, *et al.*, 1994). It draws on ideas from diverse set of disciplines including artificial intelligence, probability and statistics, computational complexity, information theory, psychology and neurobiology, control theory, evolutionary models and philosophy. The eBusiness world has also advanced due to this new way of personalization which in turn has improved the financial sector (Vico, *et al.*, 2012). The wealth of information available increases nowadays due to a continuous increase in digital information, resources and online contents (Palomares & Kovalchuk, 2017).

Recommender System has improved over the years in terms of sophistication and connectivity to other systems due to the improved quality and quantity of available data to the Recommender System (Guo, *et al.*, 2016; Palomares & Kovalchuk, 2017). The major goal of any recommender system as described by Swearingen and Sinha (2002), Tintarev and Masthoff (2011) and Vico, Huecas and Rodriguez (2012) is to maximise the "trust" property to increase the user's confidence in the systems generated recommendations. Users might not trust recommendations from a system because of the data origin. For example, restaurant recommendation from Google Places is based on subjective evaluation/rating of users. The evaluation might not depict the reality.

This work focused on using users bank statements that explicitly shows inflow and outflow of funds. The dataset used is real and reliable because the use of non-reliable data in a recommender system causes users lack of trust in the system. However, the data collected were anonymized for privacy reasons.

The importance of this research can be viewed from the following domains:

Financial: Since the goal of this research is to recommend financial products and services to customers, the product selection component chooses the best product for a customer based on customer's inflow and outflow of funds. This improves the wealth value of the products and increases Return on Investment and can ultimately reduce customer churning.

Scientific: This study applied and compared artificial intelligence techniques for pattern creation; thus, presents a framework of intelligent recommender system that recommends financial products and services using customer's financial records.

End Users/customers: This research supports financial advisors in the task of providing clients/customers with personalized investment strategies that specifies products and services.

Body of Knowledge: This research creates a model that can be used for recommendation even with transaction history and very little demographic information. In addition, the graph technology proposed helps to solve the latency incurred when using the relational databases. 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.

2.0 Review of related works

Recommender system assist users in decision making by providing suggestion of items to users (Aguilar, *et al.*, 2016; Ricci, *et al.*, 2011). A recommender system overlaps domains such as Artificial intelligence, data and semantics mining, information retrieval. Formally, a recommendation problem can be defined by a utility function "*rec*", which predicts the utility of an item *i* of a set of items *I*, for a specific user *u* of a set of users *U*.

rec is a function $R \leftarrow U \times I$.

where *R* is in the interval [0,1], and

it is the utility score of the recommended item.

Rec denotes the item's capability to satisfy the needs of the users

Items are objects to recommend which is defined by its complexity and utility

Thus, the prediction task of a recommender system is to define this utility score for a given user and item (Ricci et al, 2017). "RSs attempt to filter items to users, by predicting a rating

value for unseen items by such users so as to filter and rank the "best" unrated items in terms of their prediction value." Users are diverse and RS personalizes recommendation based on users' information. RS filters items to users by predicting a rating value for unseen items, filter and rank the items based on their prediction value (Palomares & Kovalchuk, 2017).

Taxonomy of Recommender System.

Table 2.1 presents existing RS techniques (Aguilar, et al., 2017; Jalloulia, et al., 2017; Palomares & Kovalchuk, 2017).

Type	Description
Content-based recommendation approach	RS recommend items, which are similar to the ones that the user chosen in the past. The similarity is computed according to the characteristics associated with the compared items
Collaborative recommendation approach	uses the opinions of users' community, or the information about the past behaviour, to predict the items the user will be interested. It is the mostly used approach
Demographic recommendation approach	It recommends items according to the demographic profile of the user. The idea is that each demographic niche has different recommendation needs
Utility-based or knowledge-based recommendation approach	It exploits the knowledge domain by recommending items on the basis of the Knowledge about how the item characteristics meet the needs and preferences of the users. There are two cases: case-based where the system uses the knowledge, about both the user and the items, to carry out recommendations based on similarity metrics. Constraint-based recommenders use knowledge bases with set of recommendation rules about how to map user requirements with item characteristics
Community-based recommendation approach	This approach is based on the idea that people have more confidence in recommendations from their friends than from anonymous individuals. The popularity of social networks has generated interest in this RS
Hybrid approach	is the combination of the previous techniques, in order to use the advantages of each one? The combination of the different techniques generates better or more precise recommendations, and exploits better the current information

From the above discussed approaches, this study will be adopting the hybrid approach. This is owing to the type of recommender that is to be built which will function in the financial domain.

Gigli et al., (2017) developed a recommender system for banking services that accepts implicit feedback from users. Three different recommender system algorithms were compared and the result shows that these

algorithms perform optimally with diverse combinations. The dataset used were those of investment assets acquired from the European bank. The first algorithm tested was the Bayesian Personalized Ranking algorithm. Matrix factorization method was used so as to maximize the posterior probability of user preference structure, and tune model's parameters via a 5-fold cross-validation. The

second algorithm is the Alternating Least Squares algorithm. The third algorithm is an adaptation of the Word2Vec algorithm. The result shows that the recommender system performs well with no popular item removed and poorly when some popular items are filtered out. Adebayo *et al.*, (2015) developed a web-based recommender system for financial products and services. 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 Mark-up Language, Cascading Style Sheets, and Hypertext Pre-processor were used to develop the system. The need for bank staff to manually recommend products and services to current and prospective customers. Hence, they developed an expert system that will recommend these products based on customer information. The result shows that customers can be led through the complex financial consultation process without losing their bearings in the rather complicated domain.

Farajian and Mohammadi (2019) developed a model for mining the ranking customer behaviour using clustering and association rules methods. The study presents a new two-stage frame-work of customer behaviour analysis that integrated a K-means algorithm and Apriori association rule inducer. The K-means algorithm was used to identify groups of customers based on recency, frequency, monetary behavioural scoring predictors. It also divides customers into three major profitable groups of customers. Apriori association rule inducer was used to characterize the groups of customers by creating customer profiles. Identifying customers by a customer behaviour analysis model is helpful characteristics of customer and facilitates marketing strategy development. The customers were divided into three profitable groups of customers according to their shared behaviour

and characteristics. Marketers then can infer the profiles of customers in each group and propose management strategies appropriate to each group.

Methodology

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.

The dataset on financial products and services was compiled from the Nigerian stock market. The dataset was anonymized using the tokenization technique by replacing personally identifiable information in the dataset with a token such as a numerical representation of that information and the token can be used as a reference to the original data.

Assumption patterns were created using the available balance attribute of the datasets. These patterns were employed on the anonymized dataset and the dataset is cross validated using the k-Fold cross validation technique which entailed splitting the dataset into k groups and each unique group was taken as a hold out or test dataset while the remaining groups were taken as the training dataset. The model was fitted on the training set and subsequently evaluated on the test set. The model and rules were generated using a hybrid approach (i.e. the content based and the rule-based classification algorithms).

To prevent misleading results, the dataset was cleaned. Before running the analysis, the first step taken is to ensure the proper representation and quality of data. Data processing is simply the conversion of raw data to meaningful information through a process. Data is manipulated to produce results that lead to a resolution of a problem or improvement of an existing situation.

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.

Data Collection and Processing

Financial statements of customers were used as the source of data. However, basic identifiers such as the account name and account number were removed. Random set of account numbers (dummy account numbers) were used just for identification purposes (i.e. a way to identify group of transactions performed by an individual). This was done in in order not to violate ethical standards and protect privacy of bank customers. Data gathered were not limited to just one financial institution. Data collection

was a bit challenging as financial institutions were reluctant to provide this information. In overcoming this challenge, many friends were contacted and requested to provide financial statements for some of the banks they use. Confidentiality was assured to the people that responded. Some responded positively while others refused to provide this information. In total, about twenty-four (24) different account statements were gathered; 5 from Guarantee Trust (GT) bank, 3 from Eco bank, 7 from Zenith bank, 4 from Sterling bank, 5 from United Bank of Africa (UBA). The acquired were not enough and this called for the need to collect more data. Additional data were collected over the internet and preprocessed to align with the format of the data earlier acquired from the respondents. The total available data sum up to over 600,000 instances. By using the KNN algorithm was used to train the model, customers (using account number as identifiers) are grouped into clusters with some characteristics (characteristics are; transaction frequency which was obtained from transaction date, regular inflow and average outflow of funds which was obtained from the Dr and Cr flags). The cluster was also designed to be based on products they have previously consumed. Dimensionality reduction was done to reduce the data set by identifying outliers. Using the cluster, a customer belongs to, the system looks up the business rule and makes recommendation to such

customer. It is important to state that the system also uses the products consumed by customer to improve decision to be taken while providing recommendations.

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, as shown in Figure 4.1, 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.

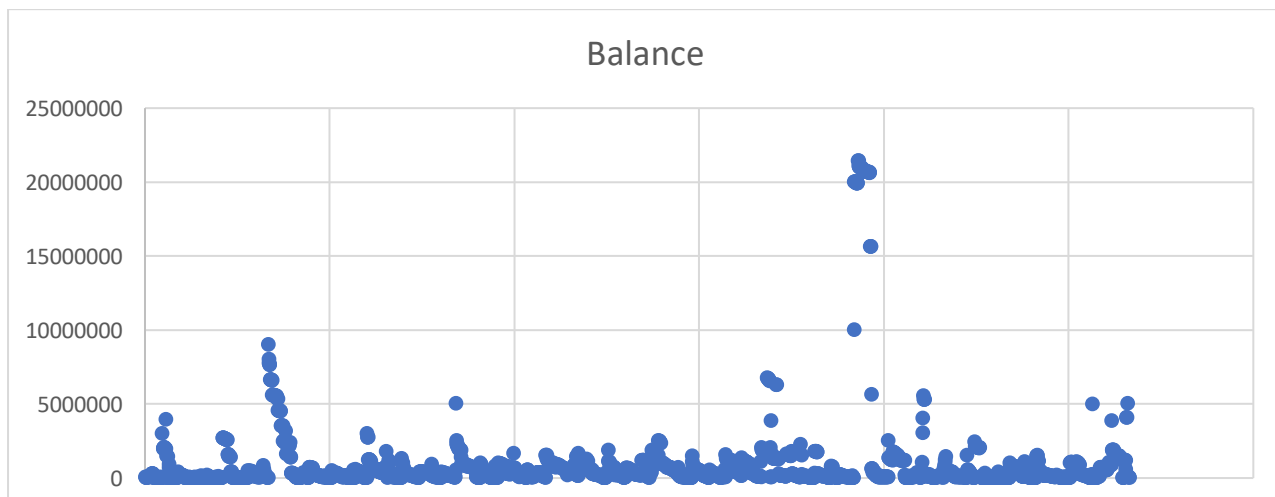


Figure 3.2: graph showing the range of average balance

RS Taxonomy Algorithms

RS has various taxonomy but four were selected for this research. The taxonomy and algorithms are –

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

The processed data is loaded into the algorithms for classification based on inputs.

Algorithm Result Comparison

The results generated from each RS Taxonomy algorithm were compared against each other and the optimal algorithm was used for recommendation

Recommender System

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.

Requirements Analysis and Design Definition

This section covers iterative and incremental activities ranging from the need concept and exploration of the need through the transformation of the need into a solution.

Need

The purpose of this research is to design an Intelligent Recommender System that provides high quality recommendations in different contexts of the financial knowledge domain.

Requirement Modelling

A model is a descriptive and visual way to convey information to a specific audience in order to support analysis, communication and understanding. Models can either be in Matrices or Diagram formats. The following modelling categories were utilized in this research.

Data and Information

This model category represents the characteristics and the exchange of information within the system. The Techniques in this category are:

Data Dictionary

Data Dictionary was used to document standard definitions of data elements. Their meanings and allowable values. Data dictionary can also be called metadata repositories and is used to manage data within the context of the solution. The data dictionary, shown in table 3.1, holds the meaning of the data columns and the allowable values in the solution dataset (Section 3.1.1).

Table 3.1: Data Dictionary

Primitive Data Elements	Name	Alias		Values/Meanings	Description
Data Element 1	Posting Date	Posting date		Date Value, Not Null	Date of transaction posting
Data Element 2	Value Date	Value Date		Date Value, Not Null	Value Date
Data Element 3	Description	Desc		Minimum characters 2	Transaction description
Data Element 4	Outflow	DR		Decimal (18,2)	Outflow
Data Element 5	Inflow	CR		Decimal (18,2)	Inflow
Data Element 6	Balance	Bal		Decimal (18,2)	Balance
Composite		Balance = Balance + Inflow - Outflow			

Database Structure

The database approach that was used is the graph-oriented database management system. Unlike other databases, relationships take first priority in graph databases. This means that applications do not have to infer data connections using foreign keys as done in relational databases. The data model for a graph database is also significantly simpler and more expressive than those of relational or other NoSQL databases (Reilly, 2019).

Graph databases are built for use with transactional systems and are engineered with transactional integrity and operational availability. Graph databases use native graph storage that is specifically designed to store and manage graphs, while others use relational or object-oriented databases instead. Non-native storage is often much more latent and takes more time. Native graph processing (a.k.a. "index-free adjacency") is the most efficient means of processing graph data since connected nodes physically "point" to each other in the database. Non-native graph processing uses other means to process operations. Relational database management systems (RDBMS) are poor at handling data relationships. Their rigid schemas make it difficult to add different connections or adapt to new requirements.

Not only do graph databases effectively store data relationships; they are also flexible when expanding a data model or conforming to changing needs. For intensive data relationship handling, graph databases improve performance by several orders of magnitude. With traditional databases, relationship queries will come to a grinding halt as the number and depth of relationships increase. In contrast, graph database performance stays constant even as the data grows year over year. With graph databases, IT and data architect teams move at the speed of business because the structure and schema of a graph model flexes as applications and industries change. Rather than exhaustively modelling a domain ahead of time,

data teams can add to the existing graph structure without endangering current functionality. Using the graph approach, the Neo4J database will be used for this research.

The data after being preprocessed and mined using the clustering algorithms were loaded into the Neo4j database. This is a NoSQL database and one of the most commonly used databases in developing recommender systems. This database approach was selected because of the time it takes to response to queries for a real time recommendation system as compared to relational databases. Neo4j had been proven to be literally thousands of times faster than MySQL solution, with queries that require 10-100 times less code. Recommendation system works hand in hand with customer relationship management system. In recent past, most customer relationship management system are developed using relational database. In order to be able to forecast customers behaviors as well as customers treatment offering, the adaptive heterogenous environment presented today must adopt more dynamic technology that help to prevent churn customers output which can be reduced with the used of recommendation systems. Neo4J database is built from the ground on a native graph database. The architecture is designed for optimizing fast management, storage, and traversal of nodes and relationships. The Neo4jrelationships are taken as referential integrity relationship and this enables it to connect to entities. A single node can connect to many entities. Entity is considered to be nodes in neo4j. An operation known in the relational database world as a join, whose performance degrades exponentially with the number of relationships, is performed byNeo4j as navigation from one node to another, whose performance is linear. nodes and relationships are simply low-level building blocks. The real strength of the property graph lies in its ability to encode patterns of connected nodes and relationships

For Neo4j to be query, a neo4j tool called cypher query language is adopted. Neo4j supports rapid

development of graph powered systems that take advantage of the rich connectedness of data. Features of Neo4j Graph Technology include; Whiteboard friendliness which allows all business stakeholders to participate throughout the development cycle. Cypher Programming Language for querying graph database, the declarative graph query language, is designed to visually represent graph. Nodes and relationships creation, Supports rapid development. Provides true data safety through ACID transactions. Designed for business-critical and high-performance operations. Neo4j can be deployed as a scalable, fault tolerant cluster of machines. The recommendation systems work on data generated by the customers and that is being used to predict their needs. These data are highly connected databases and a lot of entities needs to be designed, modeled and programmed. Since customers database entities queries at each point in time for recommendation can be a lot slow and brings issues to system produce, for this project, the architecture adopted will involve the use of NoSQL database.

The key technology in enabling real-time recommendations is the graph database architecture. This is a technology that is quickly leaving traditional relational database (RDBMS) behind. Graph technology easily outperforms relational and other NoSQL data stores when it comes to connecting masses of buyer and product data (and connected data in general) to gain insight into customer needs and product trends. Significantly, graph databases are a core technology platform of internet giants like Google, Facebook and LinkedIn. But while those pioneers had to build their own in-house datastores from scratch, off-the-shelf graph databases – notably Neo4j – are now available to any business wanting to make the most of real-time recommendation engines. In this project, graph database was used. The architectural model in publishing the graph database will be discussed below. The architectural design model used to publish the content of the graph database will be shown in Figure 3.3.

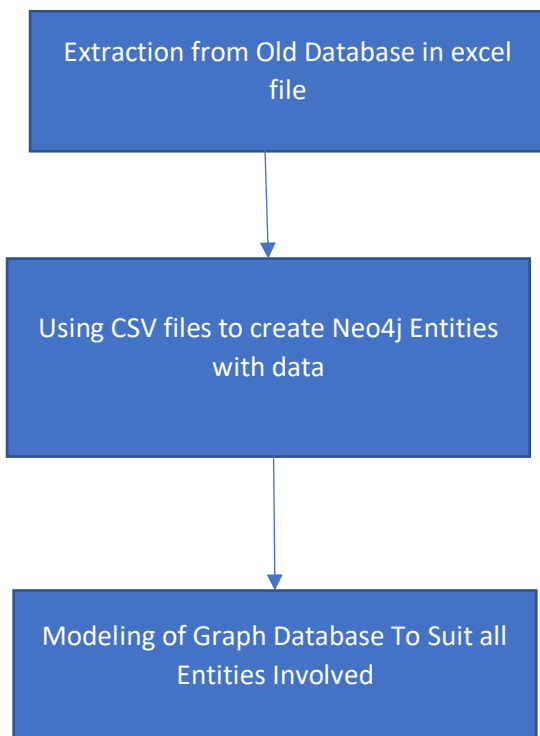


Figure 3.3: Loading the Neo4j database.

Figure 3.4 shows the interaction between the database and the application code. This facilitates communication between the Neo4j and the web interface of the developed system.

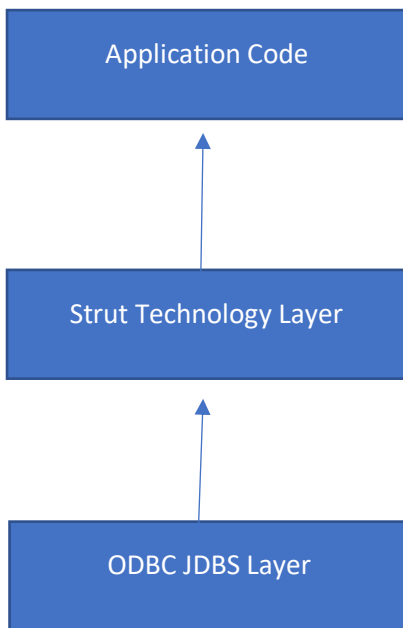


Figure 3.4. Model of interaction between the open database connectivity the application code.

Model to Analyze the Spending behavior of Customers

Spending behaviors of customers were design to be grouped in clusters. This is to identify customers with similar spending behavior and financial capability as the financial ability of a customer because the spending behavior of a customer is very dependent on his financial ability. Customer clustering is the use of past transaction data to divide customer to the similar groups. The results produced are based on the assumptions that the customer behavior follows patterns similar to past pattern and repeats in the future. This clustering approach aids decision making for the financial institutions. The decisions to be made include which target

groups of customers will be encouraged to use more products, what will be the estimated probability of acceptance of new products, promotion of new products to target groups of customers, how to manage groups of customers to reach the customer satisfaction and targeted marketing. However, quality and volume of data affects the analysis of customer behavior. For this research, the account statements of the customers were used to analyze their spending behavior. From available datasets, the customers were divided into four clusters using rules shown in listing 3.1. the rules were generated based on a model developed to model the consumer behavior as shown in equation 1.

$$Financial_{power} = \frac{\text{sum of customer score}}{\text{number of month}} \quad \text{--- Equation 1}$$

Customer score was assigned based on the transaction amount. Score table is shown in table 3.x. the score was assigned following the

model adopted by Farajian and Mohammadi, (2019)

Table 3.x: customer score table

Transaction amount (TA)	Score	User type
$0 < TA < 750,000$	1	Pamper user
$750,000 < TA < 2,500,000$	2	Moderate user
$2,500,000 < TA < 5,000,000$	3	Transactor user
$5,000,000 \leq TA$	4	Heavy user

For example, if a customer has a TA of 150000, he is assigned the score of 1. In another transaction, if same customer has a TA of 2,000,000, he is assigned the score of 2. All these scores are summed up and divided by month duration. The Financial power obtained will determine the category the customer falls into. The rules were defined by the requirements of the available products and services. The details of the products are available in the appendix.

Modelling of the database with the preprocessed data.

The preprocessed data were loaded into the graph database and modeled to have five nodes. Unlike other types of databases, the database approach is a graph-oriented database. A graph database is essentially a collection of nodes and edges. Each node represents an entity (such as a person or business) and each edge represents a connection or relationship between two nodes. Every node in a graph database is defined by a unique identifier, a set of outgoing edges and/or incoming edges and a set of properties expressed as key/value pairs. Each node relates with the other nodes as specified in Figure 3.5. The nodes; transaction node houses data and information about transactions carried out on the account as stated in the account statement. Debit and credit transactions are identified and marked. The bank node and account nodes related directly with the transaction node. The bank node houses list of available banks. This approach was adopted for easy identification of bank and transactions carried out and products to be recommended. The bank node relates

directly with the products -which are offered by the banks to customers- and the account node which is the channel of transaction between the customer and the bank. The bank node also interacts with the transaction node. The Product node holds the available bank products to be recommended to customers. 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.

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,
accountTotalWithdrawals:
toFloat(accounts.TotalWithdrawals),
```

```

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),
account: tofloat(btransactions.Amount), dr_cr:
toFloat(btransactions.Dr_Cr),          currency:
btransactions.Currency,                balance:

```

```

toFloat(btransactions.Balance), accountNumber:
btransactions.AccountNumber})
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)

```

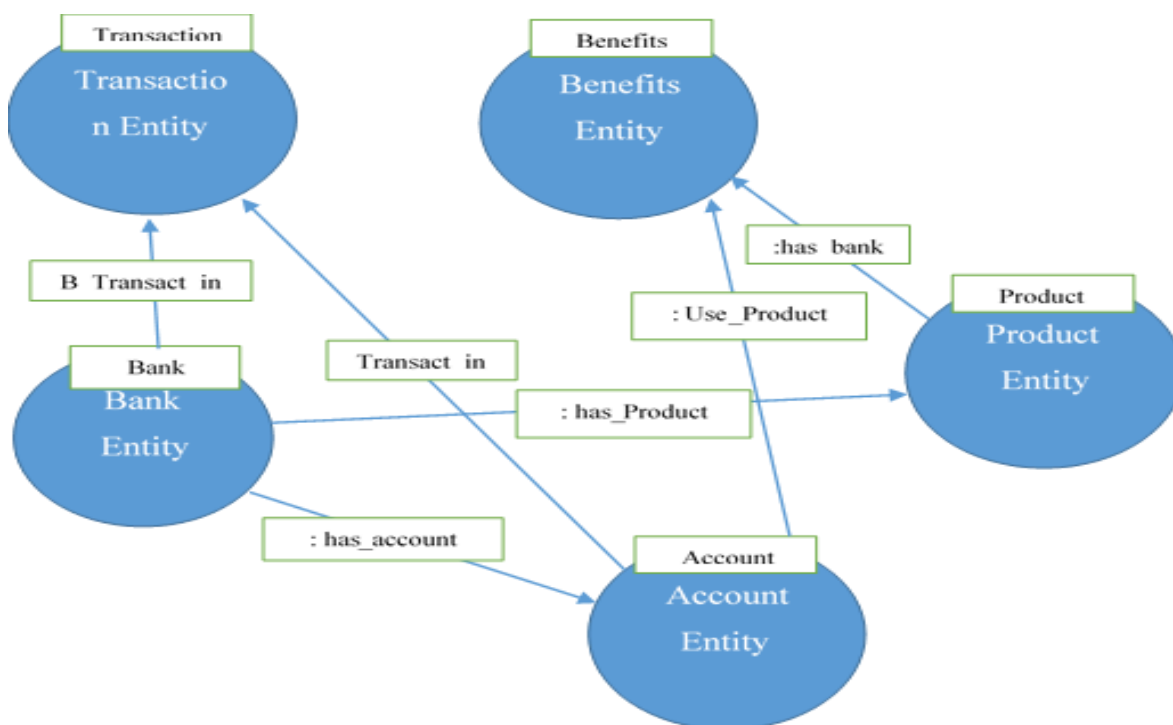


Figure 3.5 Model design of the system nodes.

Conclusion

Recommender system gives a piece of advice about the products, information or services that the user wants to know. It is an intelligent application to assist the user in a decision-

making process where they want to choose one item amongst the potentially overwhelming set of alternative products or services. 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.

REFERENCES

1. Abbas, A., Bilal, K., Zhang, L., & Khan, S. (2015). A cloud-based health insurance plan recommendation system: A user centred approach. *Future Generation Computer Systems Journal*, 43(44), 99–109
2. Adebayo A, Agbola I, Ayangbade A, & Obajimi O. (2015). Bank Products Recommender. *The International Journal Of Engineering And Science (IJES)*, 4(7), 2319–1805. Retrieved from www.theijes.com
3. Aguilar, J., Valdiviezo-Diaz, P., & Riofrio, G. (2016). A general framework for intelligent recommender systems. *Applied Computing and Informatics*. 13, 147–160
4. Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*. 46 (3), 175–185. doi:10.1080/00031305.1992.10475879.
5. Anyanwu, M., & Shiva, A., (2009). Comparative Analysis of Serial Decision Tree Classification Algorithms. *International Journal of Computer Science and Security*. 3, 230-237
6. Apte, C., Hong, S., Natarajan, R., Pednault, E., Tipu, F., & Weiss, S. (2002). Data Intensive Analytics for Predictive Modelling. Retrieved December, 11, 2018, from http://www.research.ibm.com/dar/papers/pdf/dar_overview_ibmjrd.pdf
7. Asosheh, A, Bagherpour, S., Yahyapou, A., & Asosheha. A. (2008). Extended Acceptance Models for Recommender System Adaption.
8. Bhaskar, T., & Subramanian, G. (2011). Loan recommender system for microfinance loans: Increasing efficiency to assist growth. *Journal of Financial Services Marketing*, 15(4), 334–345.
9. Brownlee, J., (2016). K-Nearest Neighbours for Machine Learning. Understand Machine Learning Algorithms.
10. Case of Retail and Banking Service in Iran (2008). World Scientific and Engineering Academy and Society (WSEAS). Transactions on Business and Economics, 5(5), 189–200.
11. Chih-Min M., Wei-Shui Y. & Bor-Wen C. (2014). How the parameters of K-nearest neighbor algorithm impact on the best classification accuracy: In case of parkinson dataset. *Journal of Applied Sciences*, 14: 171-176.
12. Fano, A., & Kurth, S. (2003). Personal Choice Point: Helping users visualize what it means to buy a BMW. *Control*, 46–52.
13. Farajian, M. A., & Mohammadi, S. (2019). Mining the banking customer behavior using clustering and association methods. *International Journal of Industrial Engineering & Production Research*, 21(4).
14. Felfernig, A., & Kiener, A. (2005). Knowledge-based interactive selling of financial services with FSAdvisor. *Proceedings of the National ...*, 100, 1475–1482.
15. Felfernig, A., & Stettinger, M. (2015). Conflict Management in Interactive Financial Service Selection. *Organizational Support*, 3.
16. Felfernig, A., Isak, K., Szabo, K., & Zachar, U. (2007). The VITA financial services sales support environment. *Proceedings of the National Conference on Artificial Intelligence*, 2(1), 1692–1699.
17. Felfernig, A., Jeran, M., Stettinger, M., Absenger, T., Gruber, T., & Haas, S. (2015). Human computation based acquisition of financial service advisory practices. *Proceedings of the 1st International Workshop on Personalization & Recommender Systems in Financial Services*
18. Felfernig, A., & Burke, R. (2008). Constraint-based recommender systems: Technologies and research issues. *10th International Conference on Electronic Commerce*, 3. ACM.

19. Gallego, D., & Huecas, G. (2012). An empirical case of a context aware mobile recommender system in a banking environment. Proceedings. 3rd FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, MUSIC 2012, 13–20.
20. Gigli, A., Lillo, F., & Regoli, D. (2017). Recommender Systems for Banking and Financial Services. In Proceedings of RecSys Posters, Como, Italy.
21. Ginevicius, T., Alchimovien, J., Kazokaitis, P., & Kaklauskas, A. (2011). Recommender System for Real Estate Management. Verslas: teorija ir praktika, (3), 258–267.
22. Gulzar, Z., Leema, A., & Deepak, G. (2018). PCRS: Personalized Course Recommender System Based on Hybrid Approach. Procedia Computer Science, 6th International Conference on Smart Computing and Communications, ICSCC 2017, 125(20) 518–524. 7-8 December 2017, Kurukshetra, India.
23. Guo, G., Zhang, J., & Yorke-Smith, N. (2015). Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. Knowledge-based Systems 74, 14–27.
24. Gupta, A., & Jain, A., (2013). Life Insurance Recommender System Based on Association Rule Mining and Dual Clustering Method for Solving Cold-Start Problem. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7), 1356 – 1360
25. Jallouli, M., Lajmi, S., & Amous, I. (2017). Designing Recommender System: Conceptual Framework and Practical Implementation. International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France Procedia Computer Science 112 (17), 1701–1710
26. Janikow, C. (1998). Fuzzy Decision Trees: Issues and Methods. IEEE Transactions on Systems, Man and Cybernetics, 28(1): 1–14., New Jersey, USA.
27. Musto, C., Lops, P., Gemmis, M., Lekkas, G., & Semeraro, G. (2015). Personalized finance advisory through case-based recommender systems and diversification strategies. Decision Support Systems, 77, 100-111
28. Palomares, I., & Kovalchuk, S., (2017). Multi-View Data approaches in Recommender Systems: an overview. International Young Scientists Conference in HPC and Simulation, YSC 2017, Kotka, Finland.
29. Pazzani, M., & Billsus, D. (2007). Content-based recommendation systems, in the adaptive web, 325–341, Springer
30. Peng, W., Chen, J., & Zhou, H. (2006). An Implementation of ID3 – Decision Tree Learning Algorithm. University of New South Wales. Retrieved October 11, 2018, from <http://web.arch.usyd.edu.au/~wpeng/DecisionTree2.pdf>
31. Press-William, H., Saul, A., William, T., Flannery, O., & Brian, P. (2007). Support Vector Machines. *Numerical Recipes: The Art of Scientific Computing (3rd ed.)*. New York: Cambridge University Press.

