Enhancing Customer Retention Using Big Data Analysis

Vikramaditya Battina
Computer Science and
Electrical Engineering,
University of Maryland,
Baltimore County
vikramb1@umbc.edu

Harika Konagala
Computer Science and
Electrical Engineering,
University of Maryland,
Baltimore County
hk12@umbc.edu

Nikhil Kumar Mengani
Computer Science and
Electrical Engineering
University of Maryland,
Baltimore County
mnikhil1@umbc.edu

***Abstract*— Customer retention is increasingly being seen as an important managerial issue, especially in the context of saturated market or lower growth of the number of new customers. It has also been acknowledged as a key objective of relationship marketing, primarily because of its potential in delivering superior relationship economics, i.e. it costs less to retain than to acquire new customers. Consumer brands often offer discounts to attract new shoppers to buy their products. The most valuable customers are those who return after this initial incented purchase. With enough purchase history, it is possible to predict which shoppers, when presented an offer, will buy a new item. However, identifying the shopper who will become a loyal buyer prior to the initial purchase is a more challenging task. This paper presents a web service that uses the purchase history in predicting the top N repeat customers using logistic regression model. We provide users with both an API and a front end interface for predicting a loyal buyer and we utilize various technologies including Hadoop, Python, and Nodejs to form the architecture of our fast, scalable, and efficient prediction web service.**

Keywords - Customer retention, repeat customer, offer, category, product, brand, company.

I. INTRODUCTION

Customer retention is important to many businesses as it is cheaper to build loyal relationships with a customer than to source for new customers [1]. A study by Bain and Company stated 25 percent to 95 percent increase in profits can be made just by increasing 5 percent of customer retention rates and a 30percent rise in company value with an increase of 10 percent of customer retention [1]. From marketing to offering discounts to loyalty programs, companies have been continually innovating in order to increase customer retention, albeit at an initial cost to themselves. A good marketing strategy to look into would be product offers. Product offers aim to attract new and old customers alike with attractive product deals as an incentive to continue buying from them. However, this comes at the expense of businesses as these deals equates to lower revenue. Hence, it is important that these costs translate to loyal customers that repeat product purchase from them within and outside of product offer periods. Whether a customer decides to repeat a purchase is dependent on a myriad of factors. These can range from loyalty and trust to a particular company or brand, or maybe the product is a necessity, such as toothpaste. As Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities. This model perfectly fits our project since we are interested in finding out the top N loyal customers with the use of product offers.

II. RELATED WORK

Our motivation to use Logistic Regression to predict the top N repeat customers comes from trying to find an algorithm that can handle both continuous and discrete variables as customer transactions have properties that are continuous (e.g. amount spent) and discrete (e.g. ID of company, category, brand). Also Suppose there are 10 customers who are loyal, in this case algorithms like decision trees and others cannot decide on the top 2 loyal customers, since logistic regression estimates the probabilities in the range of [0,1], it becomes easy for us to find the top N customers.

A similar work to our project would be Predicting Customer Shopping Lists from Point of-Sale Purchase Data by Cumby, Fano, Ghani and Krema[3]. Instead of predicting whether a customer would repeat a purchase with an offer, this research predicts what a customer would want to purchase from their past transactions using decision trees (specifically C4.5[4]), linear methods (such as Perceptron, Winnow and Naive Bayes) as well

as hybrids of different algorithms. They accounted their research results by the accuracy, precision and coverage of how well the algorithms do in predicting a customers potential shopping list, of which C4.5 have the highest precision (the number of true positive predictions)[3] at 42percent and second highest accuracy (the total number of correct predictions over the total number of examples) at 73percent. Another similar work would be Using Decision Tree to predict repeat customers by Jia En Nicholette Li, Jing Rong Lim[8]. They have also tried to predict whether a customer will repeat a purchase using decision trees.Although their results were not very accurate and good, the model still attained a test error of less than 5 percent upon cross validation and is therefore a relatively good model to predict customer repeats. Since our problem statement was slightly different from the previous approaches we have decided to go with the use of logistic regression to predict the top N repeat customers.

III. CUSTOMER AND SENARIO

For any company, the most important thing is to retain the customers. Analysis show that the customers those who repeat the purchases will contribute more to the revenue of the company than the new customers. So, it is very much important that we focus on the customer who are already associated with the company. Many companies follow various strategies to retain the customers. Some of the well-known ways are giving additional discounts, providing offer based on quantity purchased. The companies provide incentives to its customers, so that they will remain loyal to the company and repeat purchases. But the companies cannot provide incentives to all its customers. It should choose the customers who will be loyal to the company in future. Our web service will help the companies to find their top N customers who will repeat purchases after incentives has been given to them. This service just takes the inputs as the number of customers N, the company wants to give the offers to, the product ID on which offer should be given and the offer value. Then our service will return the top N customers, whom the company wants to target.

IV. DATASET

The dataset we have used for the implementation of our service comes from kaggle. It is a humongous dataset which consists of 350 million transactions of 3 million customers. The total size of the dataset is about 22 GB. We have three tables Transactions, History, offers in which data is anonymized.

Transactions table has columns customerId , chain, department, category, company, brand, Date, product size , product measure, purchase quantity and purchase measure as shown in Table 1.

CustId	Chain	Dept	Category	Company	Brand	Date	Productsize	Productmeasure	Purchase quantity	purchase amount

Table 1: Transactions Table

History table has columns customerId, chain, offer, market, repeattrips, repeater, offerDate as shown in Table 2.

CustId	Chain	Offer	Market	Repeattrips	Repeater	Offerdate

Table 2: History Table

Offers table has columns Offer, category, quantity, company, offervalue, brand as shown in Table 3.

Offer	Category	Quantity	Company	Offervalue	Brand

Table 3: Offers Table

A. Data preprocessing

From the three relational tables, we have created the training data. History and offers table have offer attribute in common. So we merged those two tables. The resulting table and transactions table have common attribute CustomerId. Joining them, we get another huge table. Joining process of the tables is shown in below Fig 1.

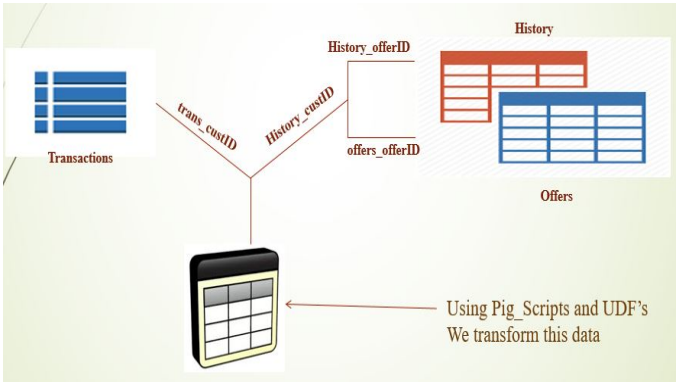

Fig. 1: Resultant Table after Joining

Now we calculated the number of times particular customer repeating purchasing in Company, Brand, Category and their combinations and mapped that to

the repeater column. To do this, we have written user defined functions in Java and executed it by using pig scripts. Pig scripts are used for map reduce on Hadoop cluster. Advantage of pig script is that, it has SQL query like statements which helps in map reducing and we can leverage the power of user defined function. After pre-processing the data it looks as follows.

COMPANY	CATEGORY	BRAND	COMP_CAT	COMP_BRAND	CAT_BRAND	REPEAT
3	2	3	2	1	1	1
0	0	0	0	0	0	0
2	2	6	2	1	1	1
6	6	16	3	2	1	0
1	1	1	1	1	1	1
2	2	0	1	0	0	0
6	0	4	0	2	0	0

Table 4: Training data

In the dataset, there is no information about the product. It has been given as combination of company, category and brand. Since we need the product Id to identify on which product offer has to be given, for the different combinations of company, category and brand, we created a unique number.

V. SERVICE DESCRIPTION

Our service incorporates Service Oriented Architecture for implementation and Representational State Transfer for communication between client and server. Our service is vendor specific. The user of our service inputs number of top N customers needed, Product Id and offer value. Then, it returns the top N customers, the company is looking for.

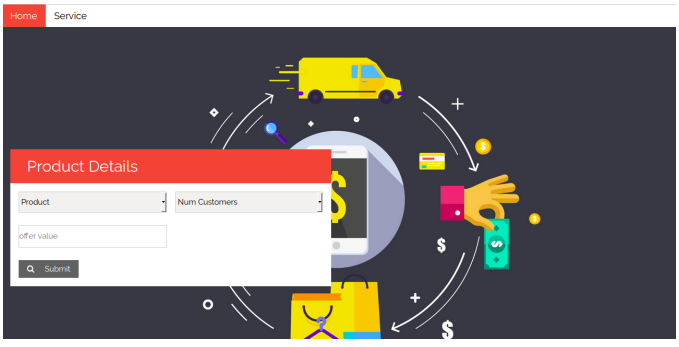

Fig. 2: Input to Service

Our service makes use of the transactions of customers which include all the history of purchases before the offer is given to them. The raw data of this purchases is stored in server and it is formatted to obtain preprocessed data. This preprocessed data is stored in MongoDB server and used for training the machine learning model. Whenever the request comes from the client, machine learning model predicts who are the top N customers for the input product and returns it to the Client as shown in Table 6.

VI. WEB SERVICE ARCHITECTURE AND SERVICE IMPLEMENTATION

Design architecture of this project shown in Fig.3 is based on assumption that streaming live data of customer transactions is on Hadoop. Data is processed and transformed by using pig scripts. Instead of processing entire data every time, which is an over exploitation of system resources like CPU and Memory, we processed only recent data on Hadoop by comparing with last processed timestamp. This task is scheduled using Cron job which runs midnight of every day. Once data is transformed into machine learning feedable i.e. attributes and its corresponding value for an individual customer, MongoDB client which knows how to interact with MongoDB middleware data server, pushes transformed data to MongoDB middleware server which contains processed data. Requests are sent in the form of JSON, key as customerId and value as the key-value pair of attribute and its value, attributes are features like same_company_count, same_category_count as discussed in this paper. MongoDB middleware data server aggregates attribute values for each customer. Usually aggregation is summation because most of our attribute values to be summed, so there is exactly a record for each customer with attribute and its values. After successful updations of data in MongoDB middleware data server, it makes a notification to Data Analysis Engine(Machine learning Model) so that it creates a new machine learning with new data from middleware data server. Most essential part is how Data Analysis Engine continues to serve requests while updating the machine learning model.

When client makes an HTTP request to client interaction server built in nodeJs, this request is routed to Data Analysis Engine(machine learning model). Requests are sent in form of JSON, key-value pairs as offer value, the number of customers and productID to which we need to predict top N repeat customers. In Data Analysis Engine, Min-Heap data structure is used to maintain the top N repeat customers. When machine learning model predicts a repeat probability

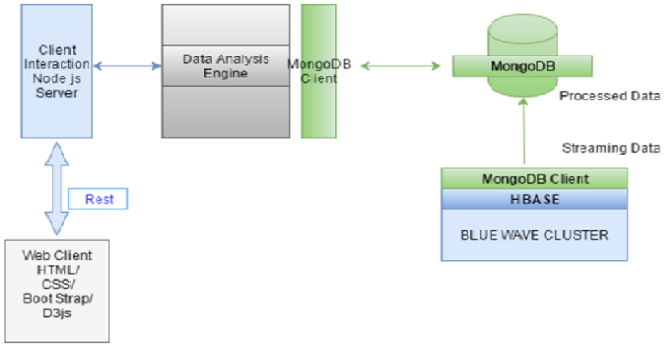

Fig. 3: web service Architecture

of a customer for a product, if the size of the heap is less than N, we will insert that element into Min-Heap otherwise, we compare minimum element in the heap. If it is greater than minimum element we will remove the minimum element and insert the new element into Min-Heap. By using this data structure we can maintain top N repeat customers as space efficient, and time efficient approach. All these top N customers are returned as a response to client interaction server in JSON list. This response is formatted and sent to the client.

A. TECHNOLOGIES USED

- **JQuery**: To dynamically update the DOM object, to register all event handlers in portal and to make an HTTP calls to Restful web services.
- **NodeJs**: Client Interaction Server in built in nodeJS. NodeJS is very good at handling concurrent requests in single process.
- **Python**: Data Analysis Engine is built using python web.py restify frame work.
- **Apache Pig**: Instead of writing complex chained map reduce jobs for processing and transformation of data, we used pig scripts which is an efficient way to handle this use case.
- **Hadoop**: Transactions, offers and train history tables, which are of 22 GB size is stored in HDFS.
- **MongoDB**: All our processed data is stored in MongoDB which acts like a cache between Data Analysis Engine and HDFS. We specially opted MongoDB as it is schema less.It helps us to add more attributes in future.
- **Java**: Apache UDFs(User Defined Functions) are written in java, UDFs are helpful when creating attributes values for transformed data.

B. APPROACH

Transaction table has 350 million transactions. So, after joining three relational tables Transactions, History and Offers, the size of the resulting relational table becomes very huge. To process such huge data is difficult. So we divided the resulting table after joining into 10 chunks. So now we have 10 tables with each having 35 million rows. On each table, we performed map reduce by using pig script. Then, we merged two tables and performed map reduce again. We continued this process until we will remain with single table. This final table obtained is the training data.

To predict whether customer repeats the purchase or not after giving offer, we just need a classification algorithm. So, we initially thought of using Support Vector Machine(SVM) . But, what we require is to pick the top N customers who will remain loyal to the company. SVM can just classify whether customers repeat or not. It cannot tell us if we need the top N customers from all the customers repeating the purchases.

Suppose if we have 10 customers who are all repeat buyers. How would we know top 2 customers?. We figured out Logistic regression algorithm is the solution to tackle this sort of problem.Logistic regression classifies the data based on the probabilities. Table 5 shows who SVM and logistic regression classifies the data.

	SVM	LOGISTIC REGRESSION
INPUT:	YES/NO 0/1	PROBABILITIES: 0.6,0.5,0.7/0.2,0.3

Table 5: Advantage of Logistic regression

We can choose the threshold probability to decide the class. As we need the top N customers from those who are all repeating purchases, we can just sort the results based on probabilities.

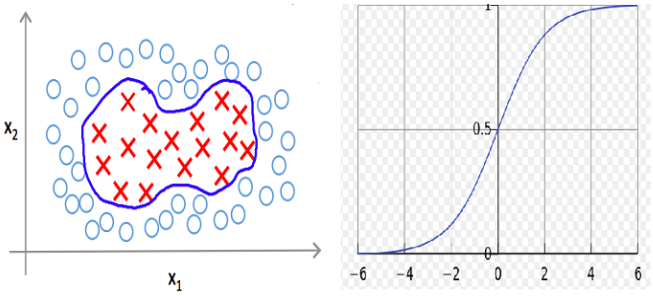

Fig. 4: SVM vs Logistic regression

We have created a rest based web service through which client can interact with the server. When the user

inputs customer details, the control goes to backend and the top N customers are retrieved as shown in Table 6. This service is implemented using Nodejs.

VII. RESULTS

We input productid, number of customers the company wants to provide offer, offer value.

CUSTOMERS NAMES
112323
23145
44532
2341
82456
110234
231452
34521

Table :6 customers Id's returned from server

Upon clicking submit, the request goes from client to server. In the server, the data is processed and based on the machine learning model prediction, the JSON response is sent to the client. The customers names are returned in the form of customer ID.

VIII. FUTURE WORK

Looking at this project as a set of sequential phases, the initial phase represents a three month concept study followed by execution of a fully functional web service which we have documented in this paper. Future work would address new capabilities to work on.

- Demonstrate on a commercial public cloud (Amazon).
- Mobile demonstration.
- Performance improvement.
- Combination of different machine learning models like Decision Trees and SVM .
- Better feature selection.
- Implement other services like targeting less repeat customers.

IX. CONCLUSIONS

In conclusion, we have shown that using the technologies of MapReduce through Hadoop, logistic regression and a framework supporting REST, WADL, and front end requests through Nodejs; we have been able to successfully demonstrate an architecture and implementation of a web service framework for enhancing customer retention by predicting the top n loyal customers.

ACKNOWLEDGMENT

We would like to thank Dr. Milton Halem for his direction and encouragement over the course of this project and his teaching assistant Yin Huang for his support in the installation and configuration of various software services on the UMBC BlueWave computing platform.

REFERENCES

[1] Reichheld, F. (2001). Prescription for cutting costs. Bain & Company. Boston: Harvard Business School Publishing.

[2] Cumby C., Fano A., Ghani R., & Krema M. (2004). Predicting customer shopping from point-of-sale purchase data. KDD '04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, (pp. 402-409). New York.

[3] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1992. Classification: Basic Concepts. Decision Trees and Model Evaluation. In Introduction to Data Mining

[4] Ivar Jrstad, Schahram Dustdar, Do Van Thanh, A Service Oriented Architecture Framework for Collaborative Services

[5] Vijayalakshmi Sampath, Andrew Flagel, Carolina Figueroa. A Logistic Regression Model To Predict Freshmen Enrollments

[6] Ying So. A Tutorial on Logistic Regression. SAS Institute Inc., Cary, NC

[7] Marti A. Hearst.Support vector machines,University of California, Berkeley

[8] Jia En Nicholette, Li Jing Rong Lim. Using Decision Tree to predict repeat customers

[9] Durgesh K. Srivastava, lekha Bhambhu .Data Classification Using Support Vector Machine

[10] Inamullah khan,Impact of Customers Satisfaction And Customers Retention on Customer Loyalty

APPENDIX A

(WADL)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application xmlns="http://localhost:3000">
   <grammars/>
   <resources base="http://localhost:3000/">
      <resource path="{continentid}">
         <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:int"
style="template" />
          <request>
          <param name="productid" type="xsd:string"
           style="query" required="true"/>
         <param name="NumberOfCustomers" type="xsd:string"
           style="query" required="true"/>
         <param name="Offervalue" type="xsd:string style="query" required="true" />
          </request>
          <response status="200">
        <representation mediaType="application/json"
          element="yn:ResultSet"/>
       </response>
      </resource>
   </resources>
</application>
```

APPENDIX B

(PIGSCRIPT)

```
export PIG_OPTS="-Djava.io.tmpdir=/data/s1/mnikhil1/output_pig"

offers = LOAD 'hdfs://n117.bluewave.umbc.edu:8020/user/mnikhil1/myinput/offers.csv' USING
PigStorage(',') AS (offer:
chararray,offers_category:chararray,quantity:int,offers_company:chararray,offervalue:double,off
ers_brand:chararray);

transactions = LOAD
'hdfs://n117.bluewave.umbc.edu:8020/user/mnikhil1/myinput/transactions.csv' USING
PigStorage(',') AS
(id:chararray,trans_chain:chararray,tirans_dept:chararray,trans_category:chararray,trans_compan
y:chararray,trans_brand:chararray,date:chararray,productsize:double,productmeasure:chararray,p
urchasequantity:double,purchaseamount:double);

trainHistory = LOAD
'hdfs://n117.bluewave.umbc.edu:8020/user/mnikhil1/myinput/trainHistory.csv' USING
PigStorage(',') AS
```

```
(id:chararray,train_chain:chararray,offer:chararray,market:chararray,repeattrips:int,repeat:chararray,offerdate:chararray);

transactions_trainHistory = JOIN transactions BY id, trainHistory BY id;
transactions_trainHistory_offers= JOIN transactions_trainHistory BY offer, offers BY offer;

REGISTER /data/s1/mnikhil1/cust_retention.jar;

x= FOREACH transactions_trainHistory_offers GENERATE
cust_retention.FeatureCalculation(trans_company,trans_brand,trans_category,offers_company,offers_brand,offers_category,repeattrips,repeat,market,train_chain,transactions_trainHistory::transactions::id) as t;

y = FOREACH x GENERATE t.$0 as (same_company:int),t.$1 as (same_brand: int),t.$2 as(same_category: int),t.$3 as (market: chararray), t.$4 as (chain: chararray), t.$5 as (repeat_trips:int),t.$6 as (repeat: int),t.$7 as (id: chararray) ;

grouped_data = GROUP y by (id,market,chain,repeat_trips,repeat);

aggregated_data = FOREACH grouped_data GENERATE
group.market,group.repeat,group.repeat_trips,group.chain,SUM(y.same_company) as num_company,SUM(y.same_brand) as num_brand,SUM(y.same_category) as num_category;
```