Acquire Valued Shoppers

Ayushi Singhal
Computer Science Department
Columbia University
as4521@columbia.edu

Dharmen Mehta
Computer Science Department
Columbia University
dmm2215@columbia.edu

Nimai Buch
Computer Science Department
Columbia University
ncb2127@columbia.edu

Abstract — **Recent technological advancements have led to a deluge of data from distinctive domains (e.g., health care and scientific sensors, user-generated data, Internet and financial companies, retail, supply chain systems and more) over the past two decades. This paper focuses on application of Big Data Analytics algorithms on a retail dataset, to predict the binary likelihood of a customer's behavior in response to a discount offer presented to him by the retail firm. We have analyzed performance of algorithms such as Adaptive Logistic Regression, Online Logistic Regression and Cross-fold Learner run using mahout on Hadoop clusters. Additionally, the paper shows analysis of the data using Hive, where we find interesting patterns within the data.**

Keywords-component; Adaptive Logistic Regression; Hive; Mahout; Cross-fold Learner; Online Logistic Regression(key words)

I. INTRODUCTION

Our project is inspired from the Acquire Valued Shoppers Challenge on kaggle.com[1]. The motivation behind this project is to identify shoppers who will return to buy if they are presented with an offer, using their previous incented purchases. The shoppers who become repeat buyers are the most valuable ones. If we have sufficient purchase history it is possible to determine which of the existing customers will purchase when provided with a relevant offer. Identifying loyal buyers without order history is a more challenging task, but since we have a lot of order history of the customers who are provided with an offer, we can use that history to predict the outcome, that is, whether or not the customer will accept the offer provided.

Apart from predicting customer behavior, another major motivation behind this project is to customize the offers based on each customer's previous transactional details. The offers would be schemed in a way such that not only would that customer be willing to take the offer but also the retail firm involved would make considerable profits. Therefore, the ideal scenario would be to create a win-win situation for both the customers as well as the retail firms.

This project can be used to maintain the existing customer base as well as attracting new customer base. If we provide the customers with exactly the deals that they would be willing to take, they will (in all probability) return. The retail firm can also get an edge over other competing retail firms with the new "customized deals" approach in order to attract new customers.

II. RELATED WORKS

Big Data Analytics has found its applications in various domains and businesses.

One of the areas in retail where big data has been applied is Micro-segmentation of customer. Customers that were traditionally segmented on the basis of macro-variables such as age, gender, life stage, store visits, basket spend etc are now segmented on varied multi-dimensional attributes based on basket analysis, social interactions among others. Big data has enabled better and granular customer segmentation by collating customer information from various sources and feedback from marketing campaigns. [2]

Another area is Generating Customer Loyalty. Amazon is able to generate over 25% of its sales using its recommendations engine that suggests popular products. Big Data allows retailers to make such instantaneous recommendations and advertisements to customers, thereby improving merchandise sales and associated services. Big data analytics can also leverage unused data assets such as surveillance videos to improve shop floor design, thus improving customer satisfaction. [3]

III. SYSTEM OVERVIEW

We have three relational files – one containing the previous transactions of the customers, one with the history of offers extended to the customers and one containing the information about the offers. There are almost 350 million rows of previous transactions from over 300,000 customers. [1] This data is in the file transactions.csv, which is almost 22GB in size. We reduce this data significantly which we will discuss shortly. The various fields in these three files are listed below –

transactions.csv [1]

id - A unique id representing a customer
chain - An integer representing a store chain
dept - An aggregate grouping of the Category (e.g. water)

category - The product category (e.g. sparkling water)
company - An id of the company that sells the item
brand - An id of the brand to which the item belongs
date - The date of purchase
productsize - The amount of the product purchase (e.g. 16 oz of water)
productmeasure - The units of the product purchase (e.g. ounces)
purchasequantity - The number of units purchased
purchaseamount - The dollar amount of the purchase

***history (trainHistory.csv & testHistory.csv)* [1]**

id – Same as above
chain – Same as above
offer - An id representing a certain offer
market - An id representing a geographical region
repeattrips - The number of times the customer made a repeat purchase
repeater - A boolean, equal to repeattrips > 0
offerdate - The date a customer received the offer

offers.csv [1]

offer – same as above
category – same as above
quantity - The number of units one must purchase to get the discount
company – same as above
offervalue - The dollar value of the offer
brand – same as above

We perform data cleaning as not all of the customers are offered incentives and we do not require the history of all such customers as they are irrelevant. Also, a number of attributes like the ‘company’ or ‘category’ which do not appear in any of the offers contained in the offers.csv file are irrelevant. As a result all such rows in the transactions file are removed. We do so using a python script called clean_data. We call these new file for transactions as reduced.csv.

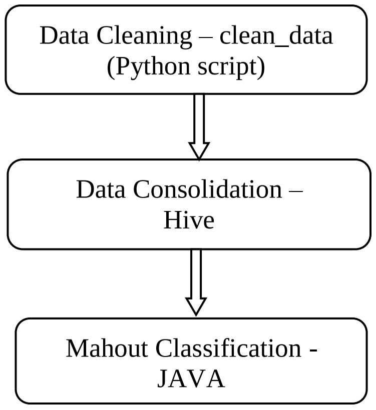
Data Cleaning – clean_data (Python script)
Data Consolidation – Hive
Mahout Classification - JAVA

Figure 1: System Design

We used Hive to create the tables and then perform join operations to create the training and testing data files. We create four different tables (‘testhistory’, ‘trainhistory’, ‘reduced’ and ‘offers’) and load the data from the csv files into these tables.

```
create table testhistory (id INT, chain INT, offer INT,
market INT, offerdate TIMESTAMP) ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',';
```

```
LOAD DATA LOCAL INPATH
'/home/bigdata/Downloads/data/testHistory.csv' INTO
TABLE testhistory;
```

After creating the tables, we first join ‘history’ with offers. The outcome of this join is then joined with ‘reduced’. Here is a sample join query that we run –

```
select * from (select * from testhistory t JOIN offers o ON
(t.offer = o.offer)) j  RIGHT OUTER JOIN  reduced r ON
(r.id = j.id AND r.chain = j.chain AND r.category =
j.category AND r.brand = j.brand AND r.company =
j.company);
```

We create the training the testing data from these joins and export them to two corresponding csv files that we use as input for our JAVA based classification module. We discuss the details of the algorithms that we use to classify in the next section.
.

IV. ALGORITHM

In all our experiments we used logistic regression and varied the different parameters involved in it. This algorithm creates a model to predict the probability of the occurrence of an event. It can also be used to predict binary values. The variables used for making the prediction can either be categorical or numerical. In particular, we used the stochastic gradient descent algorithm.

Stochastic Gradient Descent

The stochastic gradient descent algorithm is an online learning algorithm whereby it learns incrementally and performance testing can be done as the system is running. [4] In this algorithm we consider the problem of minimizing an objective function of the form:

$$Q(w) = \sum_{i=1}^{n} Q_i(w),$$

where w is the parameter to be estimated and typically each function
Q_i is associated with the ith observation in the training set.

The feature vectors used have to be fixed in length and hence the hashed feature vector encoding system is used. The idea behind this implementation is to create a vector and then use different feature encoders to iteratively add features to the vectors. The size of the vector is large enough to avoid feature collisions as features are hashed. In Mahout, there are specialized encoders available for different data types.

Online Logistic Regression

This algorithm used the stochastic gradient descent mentioned earlier. In addition, it varies the learning rates progressively by maintaining a set of learning schedules, which apply annealing. [5]

Adaptive Logistic Regression

Adaptive logistic regression uses a pool of online logistic regression learners. It varies the learning schedule of each learner and hence each learner has a different learning rate. Each time, the learner whose log-likelihood is less than the average is replaced with the survivor learners that have a higher likelihood. This derives its own annealing schedule that optimizes the speed. Instead of using the simple online logistic regression learners, we can also maintain a pool of cross fold learners which contain online logistic regression learners inside them. [6] This increases the training cost, because if we are using a k-fold cross validation, each training vector is used 4 times for training and once for classification. The adaptive logistic regression learner in Mahout maintains a pool of 100 learners.

Cross Fold Learner

In this implementation we achieve cross-validation on the data by using log-likelihood and AUC on several online logistic regression learners. [7] Each data point is passed to all but one model for training and to the remaining model for testing. In order to maintain proper segregation between the different folds across training data iterations, data should either be passed to this learner in the same order each time the training data is traversed or a tracking key such as the file offset of the training record should be passed with each training example.

V. Software Package Description

Based on our system design we first perform data cleaning using a python script. The script is stored as “clean_data.py”. We can run the script using the command –

```
python clean_data.py <path/to/transactions.py>
```

This stores the output data in the file “reduced.csv”. Next, we perform join operations using these 4 files – trainHistory.csv, testHistory.csv, reduced.csv and offers.csv. We now create the tables, load the data in the tables and then perform the join operations on these tables. We have a Hive script – create-and-load.sql for creating the tables and loading data. We run the script as follows -

```
$HIVE_HOME/bin/hive -f /home/bigdata/scripts/create-
and-load.sql
```

Now, the tables are created and data is also loaded from the csv files. Next, we perform join operation using the script – join.sql. We run the script as follows –

```
$HIVE_HOME/bin/hive -f /home/bigdata/scripts/join.sql
```

Once we get the final training the testing data files we run the classification algorithms to get the output which is stored in the file “submissions.csv”. To run the classification algorithm we run the following commands –

```
mvn package
```

```
mvn exec:java -
Dexec.mainClass=com.bigdata.project.acquireValuedShop
pers.App -DargumentA=</path/to/training.csv> -
DargumentB=</path/to/testing.csv>
```

The output – “submissions.csv” will be generated and stored in the current working directory.

VI. Experiment Results

We used the different algorithms mentioned before and varied the algorithms with some parameters. Some results obtained in these experiments will be discussed here.

The accuracies obtained by running the different algorithms are shown under:

Algorithm	***Accuracy***
Online Logistic Regression	54.85%
Adaptive Logistic Regression	56.17%
Crossfold Learner	58.25%

Table 1: Algorithm performance

For purposes of tuning, we varied the prior function to estimate which prior function resembled our dataset more closely. We varied the prior function with each algorithm but the Crossfold Learner’s performance was the best among all. The following shows the readings for Crossfold Learner Algorithm with various Prior values:

Prior function	***Accuracy***
ElasticBand Prior	54.11%
L1 Prior	55.46%
L2 Prior	58.25%
Uniform Prior	56.34%

Table 2: Crossfold Learner for Prior function values

In addition to running Adaptive Logistic Regression, Online Logistic Regression and Crossfold Learner, we used Apache Hive, we found interesting patterns from the data that could help retails stores improve business by analyzing customer behavior. This would help them find the most successful offers, customers, and type of products. Using the "brand", "id" and "repeater" attribute, we deduced which brands tend to influence the customer most to return to shop from the store. Using HiveQL we found the following:

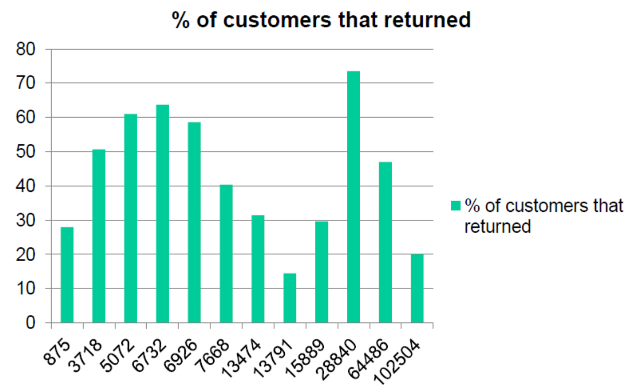

Figure 2: % of customers that returned for each brand

This graph shows trends in customer behavior based on purchases (by brands). For e.g., buyers of brand 875, return 28% of the time. This can help the retail store take crucial decisions such as which brands to sell, stock of brands to maintain in the inventory etc.

We performed similar analysis on customer behavior based on purchases (by departments).

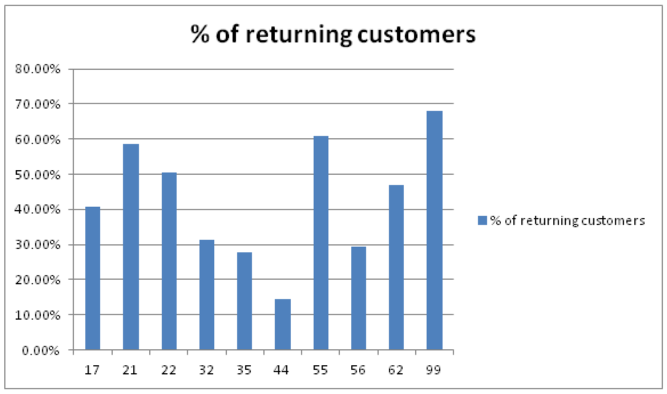

Figure 3: % of customers that returned for each department

This graph shows performance of department in the retail store. For e.g. one interpretation of this would be that brand 99 and 55 are performing well, since they have a 60% - 70% customer return rate. However, brand 44 only has a 13% customer return rate. Using this, the retail stores can further inspect product quality and other metrics that could possibly be causing low customer return rate.

In order to analyze customer behavior to certain the offers they were offered, we performed analysis on the offers and customer return rate as well.

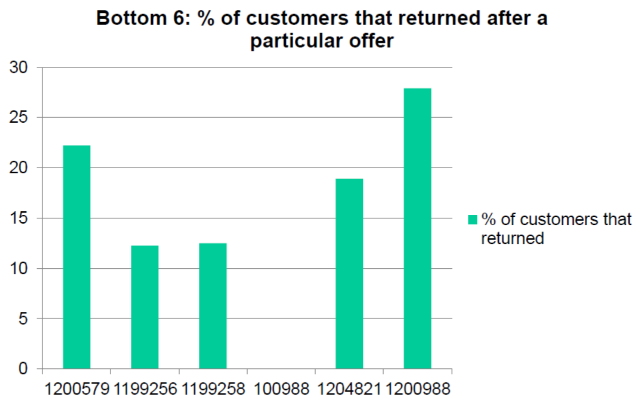

Figure 4: Bottom 6 conversion rate for offers

The above graph shows the worst performing offers in the store. For e.g. offer #100988 has a return rate of 0%, which means that of all the customers who accepted this offer, none of the customers decided to return to shop from the retails store. This analysis could help the retail store analyze the customer perception of offers.

VII. Conclusion

The system we have built can be used to predict customer behavior when presented a certain offer. To build a model, we used historical data of customer's past transactions in the retail store, the offers he/she was offered and his/her behavior. We generated three different models using Online Logistic Regression, Adaptive Logistic Regression and Crossfold Learner respectively.

We achieved best results with the Crossfold Learner. Given this model, and a set of testing data, we can categorize each customer as a probable repeater with a 58.25% confidence.

This system can help retail companies make informed decisions about creating customer-specific offers, and recommending the right products to each customer which can help the business perform better.

Ayushi Singhal, Dharmen Mehta, and Nimai Buch contributed equally towards the implementation of the project for the idea, dataset acquiring, dataset cleaning and reduction, classification and data analysis on Hive.

ACKNOWLEDGMENT

We would like to thank Prof Ching-Yung Lin for his support and advice on the project throughout the course at Columbia University. We would also like to extend our thanks to Mr. Bhavdeep Sethi for all his insightful thoughts and comments throughout the semester.

REFERENCES

[1] The Acquire Valued Shoppers Challenge on Kaggle's website - https://www.kaggle.com/c/acquire-valued-shoppers-challenge

[2] P. Hinssen, Analyzing Customer Behaviour Predicting What Happens Next, http://datascienceseries.com/assets/blog/GREENPLUM_Analyzing_customer_behavior-web.pdf

[3] Srinivasan N, Rajeev Nayar – Harnessing the power of Big Data Big opporutnity for retailers to win customers - http://www.infosys.com/industries/retail/white-papers/Documents/big-data-big-opportunity.pdf

[4] Mahout's Stochastic Gradient Descent - https://mahout.apache.org/users/classification/logistic-regression.html

[5] Mahout's Online Logistic Regression Documentation - http://archive.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.1.1/mahout-core/org/apache/mahout/classifier/sgd/OnlineLogisticRegression.html

[6] Mahout's Adaptive Logistic Regression Documentation - http://archive.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.1.1/mahout-core/org/apache/mahout/classifier/sgd/AdaptiveLogisticRegression.html

[7] Mahout's Cross Fold Learner Documentation - http://archive.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.1.1/mahout-core/org/apache/mahout/classifier/sgd/CrossFoldLearner.html