Churn Prediction Using MapReduce and HBase

Gauri D. Limaye[1], Jyoti P Chaudhary[2], Prof. Sunil K Punjabi[3]
[1]*gaurilimaye21@gmail.com*, [2]*jyotichaudhary18@gmail.com*, [3] *skpunjabi@hotmail.com*
Department of Computer Engineering, Mumbai University
SIES Graduate School of Technology
Nerul, Navi Mumbai, India

***Abstract*—**The mobile telecommunication market is rapidly increasing with numerous service providers stepping in the market. This makes the customer think to leave the service provided by one service provider and move to another service provider for some better offers. This project is an attempt to design and implement an application that an take Customer Records as input and give Customer churn prediction details as output. It will enable service provider to know in advance about the valuable customer who are about to churn. By merely giving customer data records as input, user can get the desired customer behaviour pattern, which is the churn output. The output obtained will basically distinguish the churners and the non-churners. The system is built using Apache Hadoop, Apache HBase and a Data Mining Algorithm under MapReducecode. The use of Hadoop framework makes it easy to process the large datasets containing the information of customers.

***Keywords*—** *Churn Prediction, Hadoop, MapReduce, HBase, C4.5*

*****

I. INTRODUCTION

Indian Telecom has emerged as one of the greatest economic success stories, registering a consistent overall growth rate of more than 35 percent over the past decade in terms of subscribers.[13] Therefore, the aim of these telecommunication companies mainly is to retain the existing customersrather than increasing the number of customers. And hence, it is very important for these companies to know which of their customers might leave their services and switch to the competitor. If a customer leaves one service provider and joins the services given by another service provider then that customer is known as churn customer. This can be determined by analysing the customer behaviour based on various attributes such as his number of calls per day or by the usage of services provided to him.Churn prediction is currently a relevant subject in data mining and has been applied in the field ofbanking, mobile telecommunication, life insurances, and others [13]. This type of prediction allows the companies to focus their resources on the customers who are about to churn. Also, it willavoid the loss to the company. Telecommunications companies create enormous amounts of data every day. [3] The data these companies have includes call history of the customer, details about various plans enabled by the customers, etc. This data is used to determine whether the customer is about to churn or not. As the data is large and unorganized, efficient system which can handle a large amount of unorganized data is required. For this reason the open source framework, Apache Hadoop [5] along with MapReduce [5] and HBase [6] are used. The components of Hadoop provide efficient processing and mining of data. The Hadoop Distributed File System can store a large amount of data. There are different approaches for predicting churn under data mining technologies, such as Neural Networks, Clustering, Decision Tree, Regression, and Support Vector Machine. Here in this project, we are going to use decision tree approach using C4.5 Data Mining Algorithm which provides great accuracy in giving results.

II. RELATED TECHNOLOGIES

A. Apache Hadoop

Apache Hadoop is an open source software framework [5]. Hadoop consists of two main components: a distributed processing framework named MapReduce and a distributed file system known as the Hadoop distributed file system, or HDFS. [2] One of the most important reasons for using this framework in this project is to process a large amount of data and do its analysis which is not possible with other system. The storage is provided by HDFS and the analysis is done by MapReduce. Although Hadoop is best known for MapReduce and its distributed file system, the other subprojects provide complementary services, or build on the core to provide high-level abstractions. [1]

B. Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is the storage component. In short, HDFS provides a distributed architecture for extremely large scale storage, which can easily be extended by scaling out. When a file is stored in HDFS, the file is divided into evenly sized blocks. The size of block can be customized or the predefined one can be used. In this project, the customer dataset is stored in HDFS. The dataset contains a lot of customer records which are the main constraint of this project. Also, the output of is written into HDFS.

C. MapReduce

MapReduce is a programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster [10]MapReduce works by breaking the processing into two phases: the map phase and theReduce phase. Each phase has key-value pairs as input and output, the types of whichmay be chosen by the programmer. The programmer also specifies two functions: theMap function and the Reduce function. The input to our map phase is the raw data of customers. We choose a text input format that gives us each line in the dataset as a text value. The key is the offset of the beginningof the line from the beginning of the file. The output from the map function is processed by the MapReduce framework before being sent to the reduce function. This processing sorts and groups the key-value pairs by key. [1]

 ISSN: 2321-8169
1699 - 1703

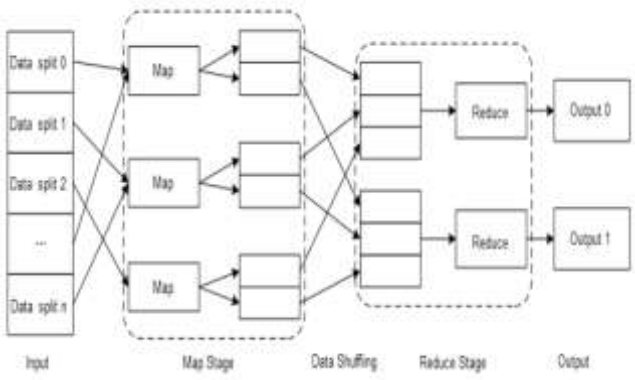

Fig. 1MapReduce Process [4]

Java code for the map function and the reduce function for this project is written for overriding the default map and reduce function provided by Hadoop framework. The logic for the respective is based on C4.5 algorithm.

D. HBase

HBase is a distributed column-oriented database built on top of HDFS. HBase is the Hadoop application to use when you require real-time read/write random access to very large datasets. [1] It provides full consistency of data, which means the database is quickly updated. As HBase has been built on top of Hadoop, it supports parallel processing. HBase can be used as data source as well as data sink. It can used be used to retrieve a particular customer's detail by writing a query.

E. C4.5 Decision Tree Algorithm

C4.5 algorithm is used to generate decision tree and is an extension of ID3. C4.5 is a standard algorithm for inducing classification rules in the form of decision tree. As an extension of ID3, the default criteria of choosing splitting attributes in C4.5 is information gain ratio. Instead of using information gain as that in ID3, information gain ratio avoids the bias of selecting attributes with many values.
The methodology of C4.5 is it first generates the rules from training data and later applies those rules on testing data to determine which customer is going to churn. The generation of rule process involves calculation of entropy of every attribute of each record along with the information gain.

F. Hue

Hue is an open-source Web interface that supports Apache Hadoop and its ecosystem.Hue aggregates the most common Apache Hadoop components into a single interface and targets the user experience. Its main goal is to have the users "just use" Hadoop without worrying about the underlying complexity or using a command line [12] .In this project Hue is used for accessing HBase file browser and HDFS file system components.

G. Swing Interface

Swing [7] Interface is used for designing Graphical User Interface in Java. This project uses Swing for designing GUI which triggers the execution of algorithm on training and test data.

III. PROPOSED CHURN PREDICTION MODEL

A. Components of the Model

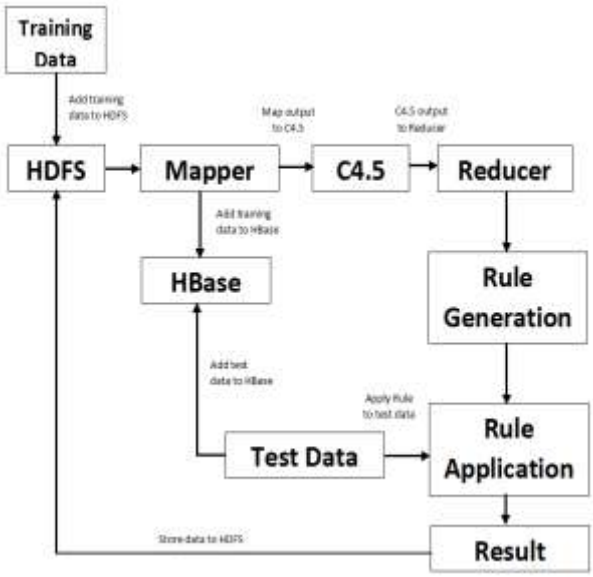

Fig. 2 Diagrammatic Representation of Model

Following are the components of the above diagram.

1) Input data files:

a) Training Data: It consists of numerous data records giving details of customers. It is initially given as an input and is stored in HDFS.

b) Testing Data: It also consists of data records giving the details of customer. It is taken as an input in churn prediction model after rule generation phase.

2) HDFS: The input data files and the output data files are written to HDFS.

3) Map Process: The map process takes input as training data from HDFS to use it for mapping.

4) HBase: The customer records from training data and testdata are written to HBase for easy retrieval of information of customer through querying.

5) C4.5: The map process invokes C4.5 algorithm toperform all the calculations.

6) Reduce Process: The C4.5 algorithm gives its output to reduce phase which aggregates the result.

7) Rule Generation: From processing and evaluation of data, basic rules are generated based on training data which will be used in rule application phase.

8) Rule Application: Testing data is given as an input to this phase where the rules generated are applied to the records of testing data.

9) Result: The churners and non-churners are predicted after all the rule application process and the result is written to HDFS.

B. Methodology

The best part of using Hadoop and MapReduce framework is that it can handle large datasets very efficiently. The dataset generated for this project has various attributes which are shown below as:

TABLE I
ATTRIBUTES FOR CALL RECORDS OF CUSTOMERS

Attributes	Description
Name of Account Holder	Gives the identity of the customer.
Account number	It is the account number which is unique for each customer.
International plan	It shows whether the customer has his international plan enabled or disabled.
Voice mail plan	It shows whether the customer has his voice message plan enabled or disabled.
Voice mail messages received	Gives the total number of voice mail message received by the customer.
Total day minutes used	It gives the total minutes the service was used by the customer during day time.
Total day calls	It gives total number of calls made by the customer during day.
Total day charges	The total charge of the calls made by the user during daytime is given.
Total evening minutes used	It gives the total minutes the service was used by the customer during evening time.
Total evening calls	It gives total number of calls made by the customer during evening.
Total night minutes used	It gives the total minutes the service was used by the customer during night time.
Total night calls	It gives total number of calls made by the customer during night.
Total international minutes used	It gives the total minutes the service was used by the customer.
Total international calls	It gives total number of international calls made by the customer.
Total international charges	The total charge of the international calls made by the user.

Based on these attributes all the calculations are done. Initially, the training data is loaded into HDFS so that it can be used by the Map().[10] The path for the input file is specified in the java code written using appropriate file input format required by Hadoop framework. The Map() takes the file and splits it record wise line by line. The Map() takes input as <key, value> pair. The Input format of key is LongWritable and the value is of Input format Text. The Map() process also writes these records to HBase. The table in HBase in which these records needs to be written is created first. Also, the row key and the column family name need to be decided before writing the data into the table

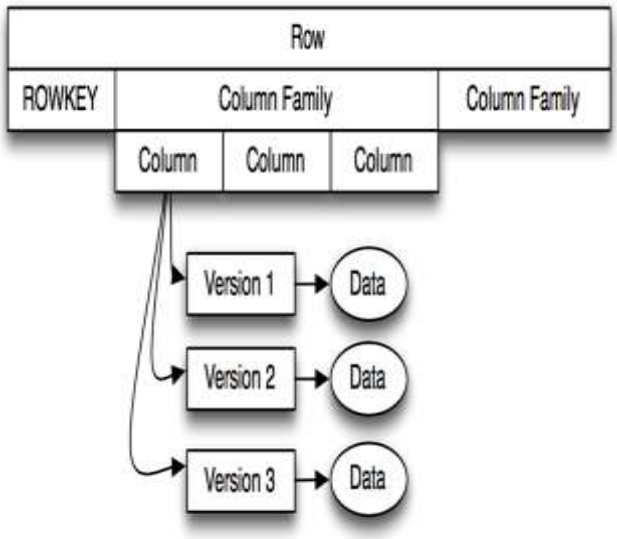

Fig.3. Diagram of row key and column family of HBase [13]

So, in this case HBase row key is taken as the 'Name' of the account holder. There are two column families taken here as 'Plan' and 'Total'. The attributes such as international plan, voice mail message plan and voice mail received are taken under the column family Plan. Remaining all the attributes are taken under the column family name Total. This helps in fast and easy retrieval of information of customers from HBase.After the Map() is done with writing the data to HBase table it invokes C4.5 algorithm.

C4.5 Algorithms does the following:

Let C denote the number of classes. In this case, there are two classes in which the records will be classified into. The classes are yes and no. The p(S, j) is the proportion of instances in S that are assigned to jth class. Therefore, the entropy of attribute S is calculated as:

$$\text{Entropy(S)} = -\sum_{j=1}^{c} p(S,j) * \log p(S,j)$$

Entropy is calculated of each record of a particular attribute. Accordingly, the information gain by a training dataset T is defined as:

$$\text{Gain(S, T)= Entropy(S) } -\sum_{v \in Values\ (T_S)} \frac{|T_{S,v}|}{|T_S|} \times \log \frac{|T_{S,v}|}{|T_S|}$$

Where, Values(T_S) is the set of values of S in T , $T_s$is the subset of T induced by S, and $T_{S,v}$ is the subset of T in which attribute S has a value of v.[4]

After calculating entropy and information gain from the above formulas GainRatio and SplitInfo are calculated as per the formulas given in the algorithm.
The output of Map() is also in <key, value> format. The output format of key is Text and that of value is IntWritable. The output of map phase is all shuffled and is sent to the reducer for aggregation. The output format of the mapper and

Algorithm 1 C4.5(T)

Input: training dataset T; attributes S.
Output: decision tree $Tree$.

1: **if** T is NULL **then**
2: **return** failure
3: **end if**
4: **if** S is NULL **then**
5: **return** $Tree$ as a single node with most frequent class label in T
6: **end if**
7: **if** $|S| = 1$ **then**
8: **return** $Tree$ as a single node S
9: **end if**
10: set $Tree = \{\}$
11: **for** $a \in S$ **do**
12: set $Info(a,T) = 0$, and $SplitInfo(a,T) = 0$
13: compute $Entropy(a)$
14: **for** $v \in values(a,T)$ **do**
15: set $T_{a,v}$ as the subset of T with attribute $a = v$
16: $Info(a,T) += \frac{|T_{a,v}|}{|T_a|} Entropy(a_v)$
17: $SplitInfo(a,T) += -\frac{|T_{a,v}|}{|T_a|} log\frac{|T_{a,v}|}{|T_a|}$
18: **end for**
19: $Gain(a,T) = Entropy(a) - Info(a,T)$
20: $GainRatio(a,T) = \frac{Gain(a,T)}{SplitInfo(a,T)}$
21: **end for**
22: set $a_{best} = \underset{a}{argmax}\{GainRatio(a,T)\}$
23: attach a_{best} into $Tree$
24: **for** $v \in values(a_{best},T)$ **do**
25: call $C4.5(T_{a,v})$
26: **end for**
27: **return** $Tree$

Fig. 4. C4.5 Algorithm Description[4]

the input format of the reducer must match. Therefore, the Reduce()[10] has input format for key as Text and that for value as IntWritable. The Reduce() also includes an output collector for collecting the output from Map() and a Reporter to check the status of Mapper. The reducer phase collects all the <key, value> pairs from the map phase and sorts it on the basis of key.

Through the MapReduce phase the behaviour of all the dataset is analysed. Data mining is basically known for deducing unseen patterns which can be further used to analyse the customer behaviour.

The rules are generated at the end of the MapReduce phase. These rules need to be applied on the testing data. The testing data is taken as input during rule application which contains the details of the new customers. So, the main goal now is to determine whether these new customers are going to churn or not depending on the behaviour analysed from the training data. The rules generated are applied on the testing data. It calculates the average of all the attributes and compares it with the threshold value defined in the code. All those customers who lie below that defined threshold value are predicted as churners while the remaining ones are predicted as non-churners. The result is finally written to HDFS.
The testing data may also contain the details of the customers which are already present in training data. The records which are there in testing data and are not present in training data are written to HBase.

IV. RESULT

The results after applying the Data Mining Algorithm using MapReduce gives us a list of churners and non-churners. This information can be seen in the output folder of HDFS. The results with corresponding customer records are added to a text file located in HDFS.

The main reason to write the customer records into HBase is to easily access the information required at times other than processing. For example, if the company wants to know the list of customers who have their international plan enabled then this can be done by writing a simple query to the HBase table to get the information. Also, searching a record by its unique id or by name of the customer is very easy through HBase. In other words, data about customers can be easily viewed from HBase. This is one of the benefit of using HBase over any other database systems.

V. CONCLUSION

This paper helps to predict the churn in telecommunication domain using MapReduce and HBase. The implementation of data mining algorithm C4.5 is done using MapReduce to make it work for Hadoop framework. The prediction is done by analyzing the customer behavior. As there are numerous service providers having numerous customers under them, it is not feasible to concentrate on each and every customer to make sure that the customer won't leave. So this prediction can help them focus more on the customers who might churn in the near future and go to some other service provider. It will help the company in utilizing their valuable efforts and giving their best deals to customers who might leave.
In future, some other algorithm which is more efficient than C4.5 can be used for doing the churn prediction. C5.0 is considered as an algorithm which can be used for prediction and is an extension of C4.5 with some improvements.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to the teaching faculty at SIES GST whose timely inputsand suggestions, helped in the completion of the project. We would like to thank our project guide Prof. Sunil K Punjabi for helping us to complete this project. Finally, we are thankful for having been given this opportunity to learn something new about the world of technology.

REFERENCES

[1] Tom White, *Hadoop: The Definitive Guide*, 3rd ed., O'Reilly Media, Inc.
[2] Dirk deRoos, Paul C. Zikopoulos, Roman B. Melnyk, Bruce Brown, and Rafael Coss,Hadoopfor Dummies, 3rd ed.,John Wiley & Sons, Inc.
[3] S. Ezhilmathi Sonia, Prof. S. BrinthaRajakumar, Prof. Dr. C. Nalini, "Churn Prediction using

MAPREDUCE,"International Journal of Scientific Engineering and Technology Volume No.3 Issue No.5, pp : 597-600, May 2014.
[4] Wei Dai and Wei Ji,"A MapReduce Implementation of C4.5 Decision Tree Algorithm," International Journal of Database Theory and Application Vol.7, No.1 (2014), pp.49-60 http://dx.doi.org/10.14257/ijdta.2014.7.1.05.
[5] (2008) The Apache Hadoop website. [Online]. Available: https://hadoop.apache.org/
[6] (2010) The Apache HBase website. [Online]. Available:http://hbase.apache.org/
[7] (1997) Swing website. [Online]. Available: http://www.oracle.com/technetwork/java/architecture-142923.html
[8] (2010) The Apache HBase website. [Online]. Available:http://gethue.com/
[9] Apache Hadoop Wikipedia page: http://en.wikipedia.org/wiki/Apache_Hadoop
[10] MapReduce: http://en.wikipedia.org/wiki/MapReduce
[11] C4.5:http://en.wikipedia.org/wiki/C4.5_algorithm
[12] Hue: http://en.wikipedia.org/wiki/Hue_%28Hadoop%29
[13] http://a4academics.com/final-year-be-project/11-be-it-csecomputer-science-project/560-data-mining-by-evolutionary-learning-dmel-using-hbase
[14] http://www.informit.com/articles/article.aspx?p=2253412