Madhav Agarwal et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Special Issue: Conscientious and Unimpeachable Technologies 2016, pp. 1-4

Machine Learning in Big Data Analytics

Madhav Agarwal, Raj Kumar Maurya

Department of Information Technology, Ajay Kumar Garg Engineering College Ghaziabad, India

***Abstract*- In this paper, we are going to explore the possibilities of integrating Machine Learning in Big Data. We can do so by using Apache Hadoop and Apache Mahout. Apache Mahout runs on the top of HDFS, a component of Hadoop. The integration of two technologies will increase the data processing speed many times. It will enable us to do use the speed of Hadoop for data processing and Machine Learning algorithms for analyzing the data. We are also going to explore the possible scopes of the integrated system of Big Data and Machine Learning in redefining the future. We will also discuss how this framework will make it possible to analyze larger amount of data.**

Keywords- Apache Hadoop, Machine Learning, Big Data, Apache Mahout

I. INTRODUCTION

Today's world is the world of data. With the abundance of data being generated every day, the rise of new automated technologies is inevitable. We have entered the quickly revolutionizing world of Big Data, Machine Learning and Artificial Intelligence, and today Big Data Specialists and Data Scientists are leading this revolution. The total amount of data stored by industry is getting doubled every one and half year. At present 4.4 Zettabytes of data is all over the internet, which is expected to grow to 44 Zettabytes by year 2020 [1]. Among all of that data only 0.5% data is ever analyzed or used. So it's a great opportunity to analyze more and more data so that we can predict things more efficiently than ever. Just think of using the whole world's useful data to draw insights, predictions and many more.

With so much of data being generated every day, efficient processing of such huge amount of data is very necessary. To store, process and analyze this data we use Apache Hadoop technology. Hadoop enables us to draw insights from various types of data that may no longer be useful for the companies at blazing fast processing speed and many advanced tools to process the data.

But only processing data at a very high speed to draw insights is not going to suffice. We need to build systems that are capable of thinking and are artificially intelligent. So to fulfill this requirement, the concept of machine learning comes into picture. Machine Learning is used for predictive analysis, pattern recognition, product recommendation system, thinking machines etc. Today machine learning is possible using different native languages like R, Python, Scala and SAS but processing big data using native languages is not efficient. So we are going to discuss how to use Machine Learning in Big Data analytics using Apache Hadoop and Apache Mahout.

A. Machine Learning

Machine Learning is a method that enables computer to learn and automates the model building without being explicitly programmed. It enables computer to teach them to grow and learn according to the new data that they encounter. The algorithms that can iteratively learn from data, machine learning allows computer to look for insights without being programmed every time by a human.[2]

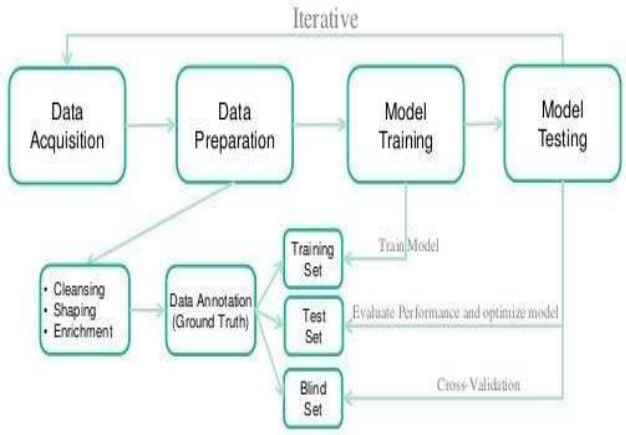

Fig. 1: Flowchart of Machine Learning analysis

The working of machine learning is similar to that of data mining as both looks from a set of data to search for patterns. However apart from it, machine learning uses that data to detect patterns and adjust actions accordingly. Machine Learning is broadly categorized into supervised and unsupervised learning. Supervised algorithms grows from the data that they learn in past. Unsupervised algorithms can predict and learn by drawing inferences from the given datasets.

"Humans can typically create one or two good models a week; machine learning can create thousands of models a week." [3]

B. Apache Hadoop

The traditional technologies which were used to analyze the big data are now not efficient in terms of money, time and effort. So now we use Apache Hadoop's Map-Reduce framework, which overcomes the disadvantages of the past technologies by reducing manpower, cost

and time. Due to the distributed architecture of Hadoop Framework the data is distributed among various nodes so processing speed is quite high.

There are two main components in Hadoop Framework. The first component is Hadoop Distributed File System (HDFS) which act as a data store and the other component is Map-Reduce which is used for the analysis of data. Both HDFS and Map- Reduce follow the master and slave architecture in which many slaves are monitored by a single master. There are 5 daemons in Hadoop, namely Namenode, Datanode, Jobtracker, Tasktracker and Secondary Namenode among them Namenode, Secondary Namenode and Jobtracker daemons act as a masters while Datanode and Tasktracker daemons act as slaves.

- Namenode: In Hadoop cluster there is a single namenode which controls all the datanodes present in the cluster. Namenode contains all the metadata that tells where the data is stored on datanodes. When a client node wants to store the data in the cluster, client asks the namenode for metadata about available datanodes.
- Datanode: In cluster there are number of nodes which act as a datanode. These datanodes are the actual place of storage of data. The data is replicated at 3 different datanodes to ensure the safety of data, satisfying the rule that not more than 2 replica can exist in the same rack.
- Jobtracker**:** In Hadoop cluster there is a single jobtracker which monitors and coordinates with all the tasktrackers present in the cluster. The jobtracker assigns the job to the task tracker. It also perform Job scheduling and Resource allocation.
- Tasktracker**:** Tasktracker is a slave daemon that accepts the task which is assigned to it by Jobtracker. Tasktracker sends a heartbeat to the jobtracker in every 3 seconds which implies that particular datanode is alive.
- Secondary Namenode**:** It helps to overcome the namenode failure issues due to crash, large edit logs or long starting time of namenode, by taking over responsibility of merging edit logs with FSImage from the namenode.[4]

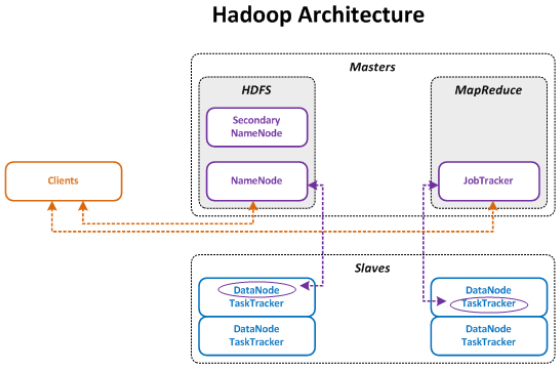

Fig. 2: Hadoop Architecture

C. Apache Mahout

Apache Mahout has the primary goal of creating scalable machine-learning algorithms on the top of HDFS. It is an open source project developed by Apache Software Foundation (ASF)

The Mahout project was started with a prime objective to build a well-documented, scalable and robust implementation of machine learning algorithms for distributed computation. The idea to develop the system was initiated by a paper "Map- Reduce for Machine Learning on Multicore" by ng et al.[5]

II. FEATURES OF APACHE MAHOUT

In spite of being young, Mahout already has a large amount of functionality, especially in relation to clustering and Collaborative Filtering (CF). Mahout's main features are:

- Several Map-Reduce enabled clustering implementations including k-Means, fuzzy k-Means, Canopy, Dirichlet, and Mean- Shift.
- Distributed Naive Bayes and Complementary Naive Bayes classification implementations.
- Distributed fitness function capabilities for evolutionary programming.
- Matrix and vector libraries.

III. BUILDING A RECOMMENDATION ENGINE

Mahout provides various tools for building a recommendation engine through the Taste library - a fast and flexible engine for Collaborative Filtering. Taste library supports both user-based and item- based recommendations and comes with various choices for making recommendations. It also allows defining your own recommendation systems. Taste has of five primary components that work with Users, Items and Preferences:

- DataModel: For storing Users, Items, and Preferences
- UserSimilarity: Provides interface to define the similarity between two users
- ItemSimilarity: Provides interface to define the similarity between two items
- Recommender: Provides interface for providing recommendations
- UserNeighborhood: Provides interface for computing a neighborhood of similar users that can then be used by the Recommenders

The above components and their implementations make it possible to build out complex recommendation systems for either real-time-based recommendations or offline recommendations. Real-time-based recommendations often can handle only a few thousand users, whereas offline recommendations can be much higher. Taste also has the tools for leveraging Hadoop to calculate recommendations offline.

To demonstrate how to create recommendations for a user given the set of ratings in recommendation_data.txt:
Load the data containing the recommendations and store it in a FileDataModel. It expects input form of each line to be of the form: user ID, item ID, preference
Compare users by declaring a UserSimilairty implementation.
Use setPrefrenceInferrerto infer preferences in the absence of an explicit setting for the user.
Use the PearsonCollerationSimilarity for UserSimilairty, which measured the correlation between two variables.[5]

Script-1: Creating the model and defining user similarity

```
//create the data model
FileDataModeldataModel = new FileDataModel(new
File(recsFile));
UserSimilarityuserSimilarity = new
PearsonCorrelationSimilarity(dataModel);
// Optional: userSimilarity.setPreferenceInferrer(new
AveragingPreferenceInferrer(dataModel));
```

Now, construct a UserNeighborhood and a Recommender.
The UserNeighborhood identifies users similar to my user and hands it off to the Recommender Recommender does the work of creating a ranked list of recommended items

```
//Get a neighborhood of users
UserNeighborhood neighborhood =
newNearestNUserNeighborhood(neighborhoodSize,
userSimilarity, dataModel);
//Create the recommender
Recommender recommender =
newGenericUserBasedRecommender(dataModel,
neighborhood, userSimilarity);
User user = dataModel.getUser(userId);
System.out.println("-----");
System.out.println("User: " + user);
//Print out the users own preferences first
TasteUtils.printPreferences(user, handler.map);
//Get the top 5 recommendations List<RecommendedItem>
recommendations = recommender.recommend(userId, 5);
TasteUtils.printRecs(recommendations, handler.map);
```

Script-2: Generating recommendations

```
    [java] Title: August 21 Rating:
3.930000066757202
    [java] Title: April Rating: 2.203000068664551
    [java] Title: April 11 Rating: 4.230000019073486
    [java] Title: Battle of Gettysburg Rating: 5.0 [java]
    Title: Abraham Lincoln Rating:
4.739999771118164
    [java] Title: History of The Church of Jesus Christ of
Latter-day Saints
            Rating: 3.430000066757202
    [java] Title: Boston Corbett Rating:
2.009999990463257
    [java] Title: Atlanta, Georgia Rating:
4.429999828338623
    [java] Recommendations:
    [java] Doc Id: 50575 Title: April 10 Score: 4.98
    [java] Doc Id: 134101348 Title: April 26 Score:
4.860541
    [java] Doc Id: 133445748 Title: Folklore of the
United States Score: 4.4308662
    [java] Doc Id: 1193764 Title: Brigham Young
Score: 4.404066
[java] Doc Id: 2417937 Title: Andrew Johnson Score:
4.24178
```

After running above scripts on our data, we get the output as follows
Output: Output from user recommendation

```
 [echo] Getting similar items for user: 995 with a
neighborhood of 5
    [java] 09/08/20 08:13:51 INFO
file.FileDataModel: Creating FileDataModel for
filesrc/main/resources/recommendations.txt
    [java] 09/08/20 08:13:51 INFO
file.FileDataModel: Reading file info...
    [java] 09/08/20 08:13:51 INFO
file.FileDataModel: Processed 100000 lines
    [java] 09/08/20 08:13:51 INFO
file.FileDataModel: Read lines: 111901
    [java] Data Model: Users: 1000 Items: 2284
    [java] -----
[java] User: 995
```

The user-based approach often does not scale. So it is better to use an item-item based approach. Taste has straightforward item-item approach. The basic code to get up and running with item-item similarity is as follows

IV. WORKING OF MAHOUT WITH HADOOP

The Apache Mahout works on the top of HDFS. The data that is needed to be processed is stored on HDFS. The Mahout framework captures data from this distributed system and applies the models and algorithm of Machine Learning. The data that is being sent to the models is sent through Hadoop in form of blocks rather than a collective unit. It enables the model to run independently and learn from each block of data on distributed system. The simultaneous processing has enabled learning process to be much faster rather than when it is done on a standalone system.

V. REAL TIME APPLICATIONS

The integrated environment of Machine Learning and Big Data has a very promising future and they can improve almost every aspect of human life. Big Data is going to play a major role in developing Smart Cities.

Also it will enable us to create homes which will be smart enough to take decisions on its own. Smart Homes will be fully automated. With the data gathered from these smart houses, it will be help in predicting the future movements of a person. It will help hotels to predict the time when number of customers will increase and hence help them in serving better. Also it will help Hospitals to predict when a person is likely to be infected by a disease and hence enable them to prepare in advance and take necessary actions to give best possible treatment to the person. The scope of integrated system is not limited to these.

A. Financial services

The large amount of data will help the banks and other businesses operating in financial industry to build models for their possible customers. It will enable them to identify investment opportunities with lower risk factor. It will also provide the best possible conditions for doing trade or invest money in the market. Also the analysis of past records of a client will enable banks to identify those customers with high-risk profile. The surveillance on the entire transactions will enable them to pinpoint any possible fraud.

B. Government

Government and other agencies are going to get the best output of the system as they possess the largest amount of data. This data can be mined and analyzed to formulate best policies. The government can use this data to figure out the impact of any decision on the public, both financial and sentimental. It will also help government to minimize theft by setting proper identification of every individual. The sensors at every corner will help in keep track of the possible criminals and will greatly improve the safety of people from any possible terrorist attack.

C. Health care

Machine Learning has already started enhancing the health care facilities. By gathering more data about human health, it will be possible to predict possible diseases that may going to affect an individual in near future. If doctors and hospitals know in advance that a medical casualty may occur in near future, then they can prepare for it with more efficacy. Today's medical system is based on studying the previous cases where certain common symptoms are matched with patient's current situation and based on that a medicine is recommended. But it is limited to the memory of a doctor. If Machine Learning will be implemented and the entire data of medical history recorded so far is analyzed before recommending a medicine, then the result would be much more effective.

D. Marketing and sales

The current recommending system of the items that a person may like based on his previous records is a great example of the integrated model. It has already shown great increase in sales. The ability to grab data, analyze it and use it to give the best possible personalized shopping experience is the future of retail industry. This model can be implemented on purchase of every small commodity by gathering and analyzing data of every individual. This will also enable industries like hotels to predict which customer will going to order which food and hence provide the best service.

E. Renewable and Non-Renewable Energy

By gathering data about the planet will help in predicting the possibilities of energy sources in an area. It will also help in exploiting renewable energy sources to their full extent. The wastage of energy can also be controlled by optimizing the power supply for every locality as per their needs. This will also affect current energy producing plants by helping in predicting possible failures and suggesting backup options for it.

F. Transportation

The exponential increase in personalized cars in past decade has increase the problem of traffic in every country. By analyzing the patter and trends in transportation, it will help in building more effective routes and help in dealing with the traffic more effectively. The transportation authorities will formulate a better plan if they know the possible traffic problems in near future. Also self-driven cars are no more a fantasy, they have already been in testing phase on roads. But the challenge that they are facing is accurate prediction of the next move they are going to make on road. This prediction model can be further enhanced by using the integrated environment.

VI. CONCLUSION

The integrated system developed by Big Data and Machine Learning through Apache Mahout has removed many barriers. By implementing it, Machine Learning algorithms are not confined to single system. They can use the distributive computing capabilities of Hadoop. The computing capabilities can be increased manyfolds by processing data on as many computers as per the requirements. The data processing speed will increase manyfolds. At the same time, user can now analyze and done prediction of such a large amount of data, which was beyond imagination.

VII. REFERENCES

[1]. Bernard Marr. Big Data: 20 Mind-Boggling Facts Everyone Must Read. Web link: http://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must- read/#1550ab626c1d

[2]. Machine Learning: What it is and why it matters. Web link: http://www.sas.com/en_us/insights/analytics/machine-learning.html

[3]. Thomas H. Davenport. The Wall Street Journal

[4]. Tom White. Hadoop: The Definitive Guide, 4th Edition. (2015)

