Unstructured Data Analysis and Processing Using Big Data Tool-Hive and Machine Learning Algorithm-Linear Regression

Neha Mangla
Atria Institute of Technology
17 PUBLICATIONS **31** CITATIONS

Priya Rathod
HKBK College of Engineering
2 PUBLICATIONS **8** CITATIONS

International Journal of Computer Engineering & Technology (IJCET)
Volume 9, Issue 2, March-April 2018, pp. 61–73, Article ID: IJCET_09_02_006
Available online at
http://www.iaeme.com/ijcet/issues.asp?JType=IJCET&VType=9&IType=2
Journal Impact Factor (2016): 9.3590(Calculated by GISI) www.jifactor.com
ISSN Print: 0976-6367 and ISSN Online: 0976–6375
© **IAEME** Publication

UNSTRUCTURED DATA ANALYSIS AND PROCESSING USING BIG DATA TOOL - HIVE AND MACHINE LEARNING ALGORITHM - LINEAR REGRESSION

Neha Mangla
Department of ISE, Atria Institute of Technology, India

Priya Rathod
Department of CSE, HKBK College of Engineering, India

ABSTRACT

Big data represents the information assets characterized by a high volume, velocity and variety to require specific technology and analytical methods for its transformation into value. The storage of large chunks of data is difficult as even terabytes and petabytes of traditional data warehousing solutions is insufficient and exorbitant [1][2].

It is viable to store and process these ransom amounts of data [13][14]15][16]17]18][19][20][21] on Hadoop; which is a low cost, reliable, scalable and fault tolerant Java-based programming framework that supports the processing of large data sets in a distributed computing environment. Hadoop implements MapReduce programing model for storing and processing large data sets with a parallel, distributed algorithm on commodity hardware. Nevertheless, the programming model expects the developers to write bespoke programs that are less flexible, time consuming, hard to code; maintain and reuse. This challenging task of writing complex MapReduce codes was rationalized by making use of HiveQL.

Hive is the platform required to run HiveQL. Hive is built on top of Hadoop to query Big Data. Internally the Hive queries are converted into the corresponding MapReduce task [3][4].

In this paper, by making use of machine learning algorithm a movie rating prediction system is built based on MovieLens dataset.

Key words: Big Data, HDFS, Hadoop, Hive, MapReduce, linear regression.

Cite this Article: Neha Mangla and Priya Rathod, Unstructured Data Analysis and Processing Using Big Data Tool - Hive and Machine Learning Algorithm - Linear Regression. *International Journal of Computer Engineering and Technology,* 9(2), 2018, pp. 61-73.
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=9&IType=1

1. INTRODUCTION

The prediction system is built using machine learning algorithm. This system employs sentiment analysis that identifies and extracts subjective information based on selected training sets from MovieLens dataset.

At present, cinema has the greatest potential to be the most effective and entertaining mass media instrument. Various software applications and websites such as Bookmyshow, Moviefone, etc.., which are the biggest online movie brand don’t just assist for ticket booking but also aim at reaching users' satisfaction by providing them with the facility to rate the movies that they have watched and also access feedback on yet to watch movies based on others' ratings. Hence prediction on movie ratings is remarkable.

The system predicts and provides the users with suggestions based on their previous ratings recorded and other users’ ratings. These predictions provide an opportunity for movie makers to have better understanding about the viewer’s expectations which in turn is beneficial for marketing. This is done by determining the relationship between viewers’ and their ratings. Further by making use of effective BigData analysis tools as in this paper Hadoop and Hive are made use of, larger datasets can be analyzed which provides statistically accurate outcomes.[5] These findings provide better understanding about viewers’ expectations and hence movie choice.

In this paper, we use MovieLens dataset which is an open dataset collected by GroupLens research; University of Minnesota. This dataset is made available on the website for the users to rate movies. MovieLens Dataset comprises of 100K, 1M, 10M datasets having 100 thousand ratings on 1,700 movies from 1,000 users, 1 million ratings on 4,000 movies from 6,000 users and 100 thousand ratings on 10,000 movies from 72,000 users respectively.[6][7][8]

As traditional approaches are not appropriate solution for the analysis of big data many research communities have recommended various solutions for managing various Big Data challenges. Amongst various solutions Hadoop, MapReduce Programming codes, HIVE, PIG, Hbase, Sqoop, NoSQL are the leading ones. In this paper, HiveQL is used to analyze the dataset which is elaborated in section 2.

2. DATASET PREPROCESSING USING HIVE

Hive was started at Facebook in the year 2006 because of the difficulty in controlling of a large amount of data which was increasing like, from few gigabytes to terabytes. Hive acts as data warehouse system built inside the hadoop file system. It is used to analyse large data sets which cannot be handled by tradition RDBMS. It provides user with a platform where they can easily use queries similar to SQL but is named differently called HiveQL. As people are now days more prone to SQL.

HiveQL help in managing structured data. It hides various complexity of Hadoop like now there is no need to learn map reduces which is very important in Hadoop. Apart from this, no need to learn JAVA and Hadoop APIs. All in all it is very useful but with just one constraint that it can be just used for structured data, it cannot handle unstructured and semistructured data. Following steps are followed for preprocessing MovieLens dataset using HIVE:

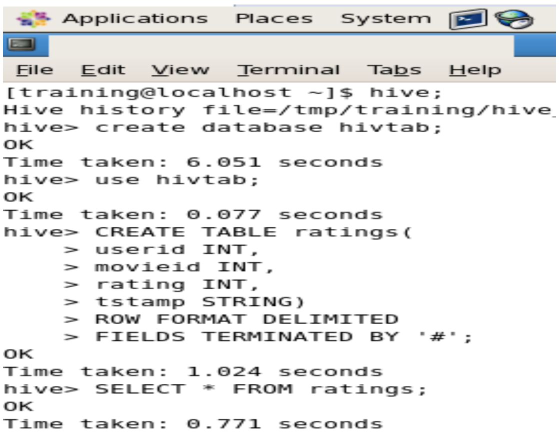
2.1. MovieLens Dataset schema

MovieLens Dataset is collected and stored into HDFS (Hadoop Distributed File System) from the websitehttp://grouplens.org/datasets/movielens. For the ease of analysis 100K data set has been chosen. The movies.dat, ratings.dat and the users.dat files have [movieID, tile, gednre], [userID, movieID, rating, timestamp] and [userID, gender, age, occupation, zipCode] fields respectively;[9] with each field delimited from the other by # symbol.

2.2. Creating Tables and Loading Data

Tables with same schema as that of the data is created for each of the three files. Hive query to create ratings table and result for the same is as shown in Fig 1.0

Similarly, tables have been created for movies and users files based on their attributes respectively.



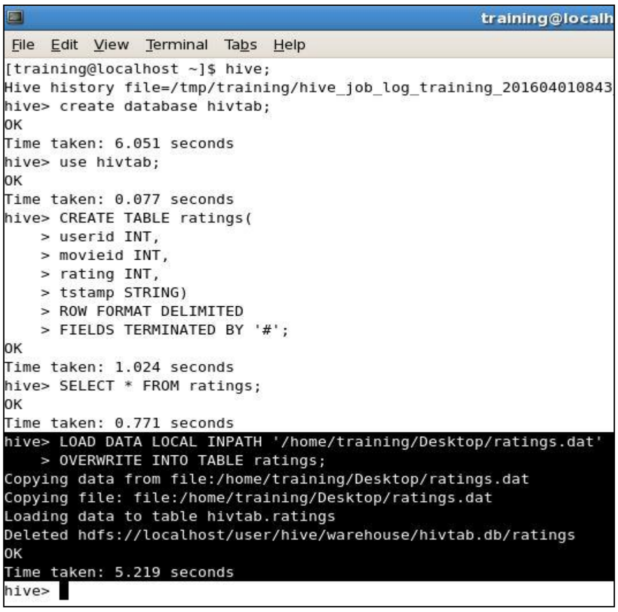
```
[training@localhost ~]$ hive;
Hive history file=/tmp/training/hive_
hive> create database hivtab;
OK
Time taken: 6.051 seconds
hive> use hivtab;
OK
Time taken: 0.077 seconds
hive> CREATE TABLE ratings(
    > userid INT,
    > movieid INT,
    > rating INT,
    > tstamp STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '#';
OK
Time taken: 1.024 seconds
hive> SELECT * FROM ratings;
OK
Time taken: 0.771 seconds
```

Figure 1 Creating Ratings

The next step is to load the data into the tables after creating all the three tables. Hive provides us with the utilities to load datasets from flat files stored on HDFS using the LOAD DATA command. The following is the command signature:

LOAD DATALOCAL INPATH <’path_to_flat_file’> OVERWRITE INTO TABLE <table_name>;

The result is as shown below:

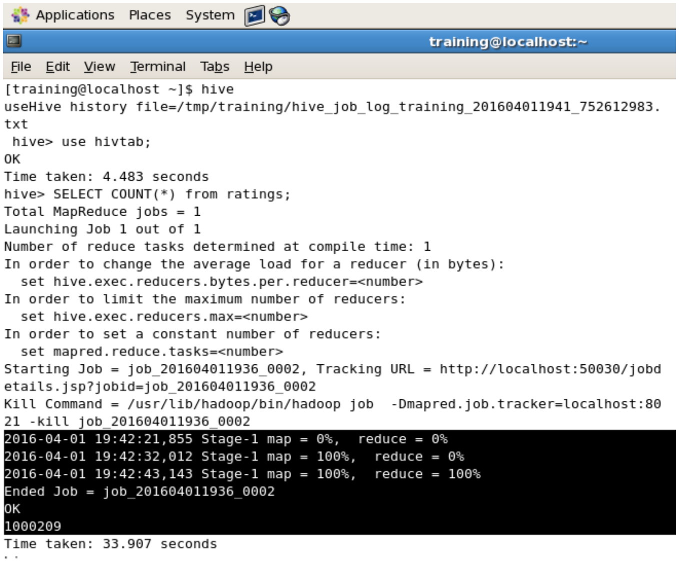


```
[training@localhost ~]$ hive;
Hive history file=/tmp/training/hive_job_log_training_201604010843
hive> create database hivtab;
OK
Time taken: 6.051 seconds
hive> use hivtab;
OK
Time taken: 0.077 seconds
hive> CREATE TABLE ratings(
    > userid INT,
    > movieid INT,
    > rating INT,
    > tstamp STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '#';
OK
Time taken: 1.024 seconds
hive> SELECT * FROM ratings;
OK
Time taken: 0.771 seconds
hive> LOAD DATA LOCAL INPATH '/home/training/Desktop/ratings.dat'
    > OVERWRITE INTO TABLE ratings;
Copying data from file:/home/training/Desktop/ratings.dat
Copying file: file:/home/training/Desktop/ratings.dat
Loading data to table hivtab.ratings
Deleted hdfs://localhost/user/hive/warehouse/hivtab.db/ratings
OK
Time taken: 5.219 seconds
hive>
```

Figure 1.1 Loading Data Into Ratings Table

The same process of loading the data is carried out for all the three tables.

The verification of data getting loaded into the table can be carried out by displaying the table contents or in the following way making use of SELECT COUNT.



```
[training@localhost ~]$ hive
useHive history file=/tmp/training/hive_job_log_training_201604011941_752612983.
txt
 hive> use hivtab;
OK
Time taken: 4.483 seconds
hive> SELECT COUNT(*) from ratings;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201604011936_0002, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201604011936_0002
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201604011936_0002
2016-04-01 19:42:21,855 Stage-1 map = 0%,  reduce = 0%
2016-04-01 19:42:32,012 Stage-1 map = 100%,  reduce = 0%
2016-04-01 19:42:43,143 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201604011936_0002
OK
1000209
Time taken: 33.907 seconds
```

Figure 1.1 Verification

The highlighted region conveys 2 important points. Firstly, the number of rows present in the table which is as expected approximately equal to 10K. secondly, how hive is internally converted into map-reduce tasks and only after the completion of map phase the reduce phase begins.

3. APPLYING HIVE QUERIES ON DATASETS

Now that the tables have been created and loaded with the respective datasets successfully, they can be queried using hiveQL which is depicted in the following sections.

3.1. Differential Rating based on Gender

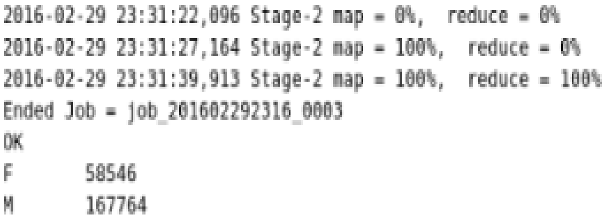
The subsequent hive query determines the number of people who have rated 5 for the movies based on gender.

hive> select users.gender, count(*)

from ratings join users on(users.userid=ratings.userid)

where rating=5 group by users.gender;

The result is as shown below:



```
2016-02-29 23:31:22,096 Stage-2 map = 0%,  reduce = 0%
2016-02-29 23:31:27,164 Stage-2 map = 100%,  reduce = 0%
2016-02-29 23:31:39,913 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_201602292316_0003
OK
F       58546
M       167764
```

Figure 1.2 Gender based ratings

From the result obtained we can infer that more number of males rate a movie 5 than female.

3.2. Differential Rating based on Occupation

The subsequent hive query determines the number of people who have rated 5 for the movies based on occupation:

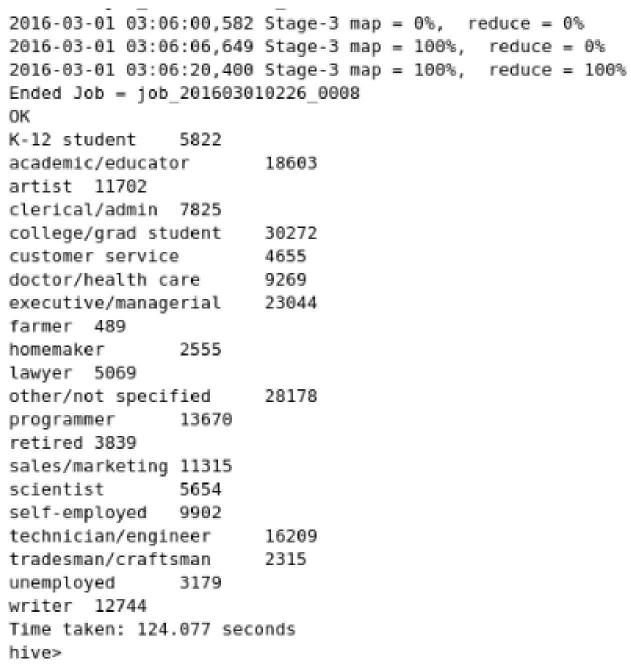
hive> select occupations.occupation,count(*)

from users join occupations on(occupation.id=users.occupation)

join ratings on(ratings.userid=users.userid)

where ratings=5

group by occupation.occupation;



```
2016-03-01 03:06:00,582 Stage-3 map = 0%,  reduce = 0%
2016-03-01 03:06:06,649 Stage-3 map = 100%,  reduce = 0%
2016-03-01 03:06:20,400 Stage-3 map = 100%,  reduce = 100%
Ended Job = job_201603010226_0008
OK
K-12 student    5822
academic/educator       18603
artist  11702
clerical/admin  7825
college/grad student    30272
customer service        4655
doctor/health care      9269
executive/managerial    23044
farmer  489
homemaker       2555
lawyer  5069
other/not specified     28178
programmer      13670
retired 3839
sales/marketing 11315
scientist       5654
self-employed   9902
technician/engineer     16209
tradesman/craftsman     2315
unemployed      3179
writer  12744
Time taken: 124.077 seconds
hive>
```

Figure 1.3 Occupation based ratings

3.3. Differential Rating based on Age

The following hive query determines the number of people who have rated 5 for the movies based on age:

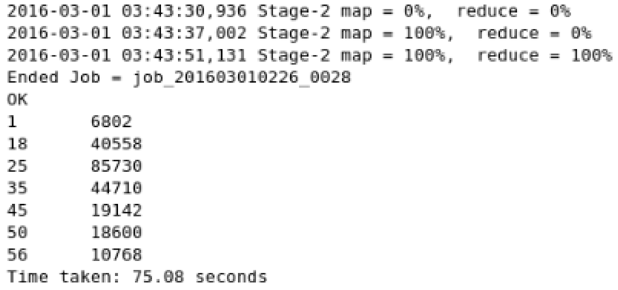
hive> select users.age, count(*)

from ratings join users on(ratings.userid=users.userid)

where rating=5

GROUP BY users.age;

The outcome of the query is as shown below:



```
2016-03-01 03:43:30,936 Stage-2 map = 0%,  reduce = 0%
2016-03-01 03:43:37,002 Stage-2 map = 100%,  reduce = 0%
2016-03-01 03:43:51,131 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_201603010226_0028
OK
1       6802
18      40558
25      85730
35      44710
45      19142
50      18600
56      10768
Time taken: 75.08 seconds
```

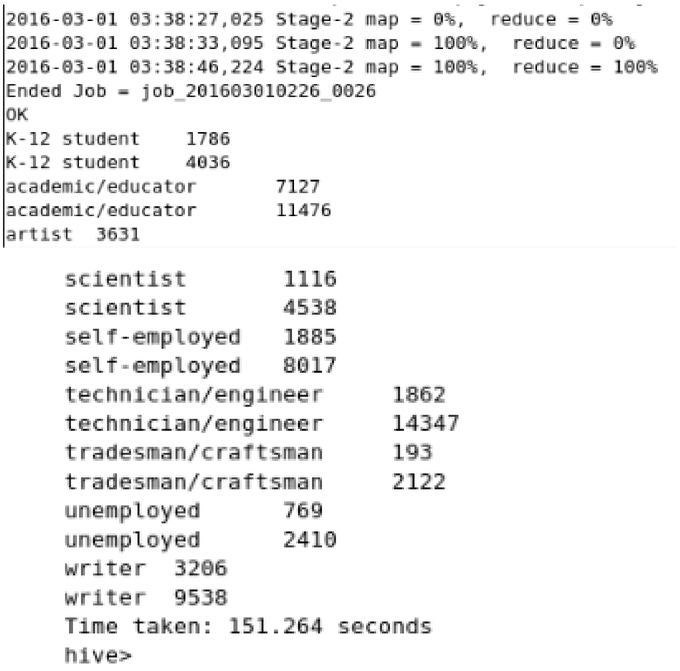
Figure 1.4 Age based ratings

From the result obtained we can conclude that viewers around the age group 25years rate movies the highest (rate movies 5).

3.4. Differential Rating based on Occupation and Gender

The following query determines the number of people who have rated 5 for the movies based on occupation and gender.

```
hive> SELECT occupations.occupation, count(*)
from users join ratings on(ratings.userid=users.userid)
join occupations on(users.occupation=occupations.id)
where ratings=5
GROUP BY occupations.occupation, gender
```



```
2016-03-01 03:38:27,025 Stage-2 map = 0%,  reduce = 0%
2016-03-01 03:38:33,095 Stage-2 map = 100%,  reduce = 0%
2016-03-01 03:38:46,224 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_201603010226_0026
OK
K-12 student    1786
K-12 student    4036
academic/educator       7127
academic/educator       11476
artist  3631
```

```
scientist       1116
scientist       4538
self-employed   1885
self-employed   8017
technician/engineer     1862
technician/engineer     14347
tradesman/craftsman     193
tradesman/craftsman     2122
unemployed      769
unemployed      2410
writer  3206
writer  9538
Time taken: 151.264 seconds
hive>
```

Figure 1.5 Occupation and gender based ratings

From the above shown outcome we can conclude that each occupation’s rating is mentioned twice with respect to the gender females rating followed by the males rating.

4. ALGORITHMS

4.1. Introduction

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. It grew out of Artificial Intelligence.

Machine Learning (ML) generally refers to the development of methods that optimize their performance iteratively by ”learning from the data”. Machine Learning (ML) are broadly understood as methods that analyze data, and make useful discoveries and inferences from the data. It relies heavily on techniques and theory from statistics, optimization, algorithms and biological inspired system.

In literature we have many learning algorithms which comes under either supervised or unsupervised learning.

- Supervised Learning: is a type of machine learning algorithm that uses a known training dataset to make predictions.
- Unsupervised learning: is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses. [10][11]

Through supervised learning we can learn what makes the rating a certain value from the selected training dataset. In our paper we will focus on linear regression which is a type of supervised learning.

4.1.1. Linear Regression

Linear Regression is an approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables denoted by x. In subsequent section we have described the mathematical description of linear regression for our problem statement.

4.1.2. Variables Description

Let,

m=number of training examples

x=input variables

y=output/target variables

i=an index to training set

(x^i, y^i) implies i^{th} training example

In the following equation,

$h_\theta(x)=\theta_0+\theta_1 x$

where,

$h_\theta(x)$ is the numerically calculated values based on chosen parameters also termed as hypothesis function

θ_i are parameters

θ_o is zero condition

θ_1 is gradient

4.1.3. Cost Function

Cost function lets us figure out how to fit the best straight line to our data by choosing values for θ_i .

Based on the training set values for the parameters have to be generated so as to fit in the best possible straight line.

Values for the parameters are chosen such that $h_\theta(x)$ is close to y for the training example. Basically, uses xs in training set with $h_\theta(x)$ to give output which is as close as possibleto the actual y value. $h_\theta(x)$ can be considered as a "y imitator" - it tries to convert the x into y,and considering we already have y we can evaluatehowwell $h_\theta(x)$ converges with y.

The cost function is given by:

$J(\theta_0,\theta_1)=1/2m\ \Sigma(h_\theta(x^{(1)})-y^{(i)})^2$

where,

$J(\theta_0,\theta_1)$ is the cost function

y is the linear function of x

Σ varies from i=1 to m

$(h_\theta(x^{(1)})-y^{(i)})^2$ impliestrying to minimize squared difference between predicted ratings and actual ratings called in general as minimization problem.

- 1/2m
    - 1/m – average determination
    - 1/2m the 2 doesn't change the constant value negligibly.
- Minimizing θ_0,θ_1 means finding the values of θ_0 and θ_1 , which find on average the minimal deviation of x from y when the parameters are used in hypothesis function.
  Above θ_0,θ_1 is taken only for one input and one output. But our problem statement is having multiple attributes so we have more theta values for experiment.

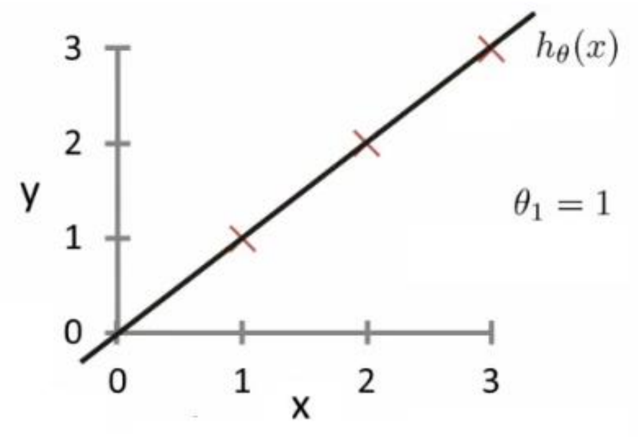
4.2. Gradient Descent

Gradient descent is an optimization method for minimizing an objective function that is written as a sum of differentiable functions. Used in machine learning for minimization of cost function.

Gradient Descent is all about:

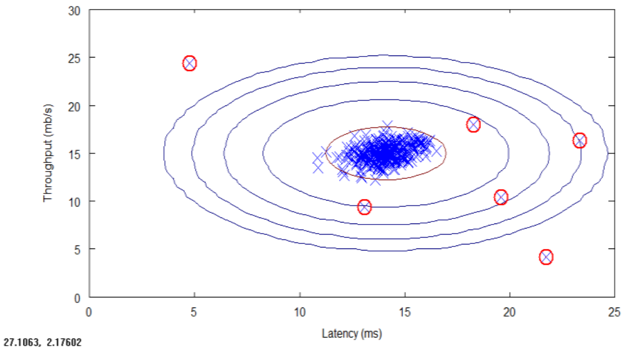
We have $J(\theta_0, \theta_1)$

We want to get min $J(\theta_0, \theta_1)$



Graph 1 uni-variant linear regression

As shown in the above graph y directly depends on the value of x. Repeated computation of the hypothesis function $h_\theta(x)$ and applying minimization to the hence obtained cost function, most accurate graph for the prediction system canbe determined.



Graph 2 Multi-variant

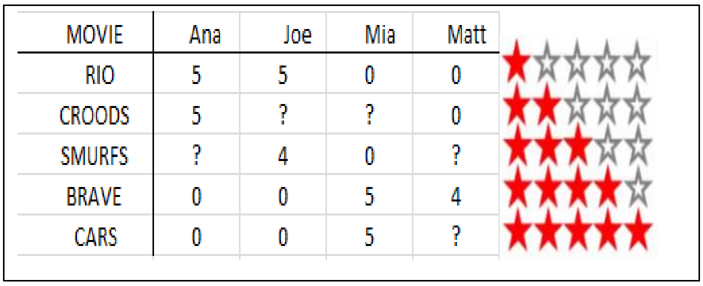
The above graph is with respect to multiple variables as implemented in this paper.

4.3. Implementation

In this paper, we make use of GNU Octave which is a high-level interpreted language intended for numerical computations. It provides capabilities for the numerical solution of linear and non-linear problems and for performing other numerical experiments. It also provides extensive graphic capabilities for data visualization and manipulation.

The Octave language is quite similar to Matlab so that programs are easily portable. [12]

Consider the following rating table for the movie dataset:



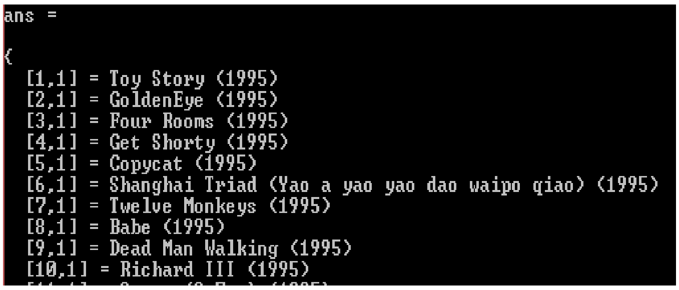
MOVIE	Ana	Joe	Mia	Matt	
RIO	5	5	0	0	★☆☆☆☆
CROODS	5	?	?	0	★★☆☆☆
SMURFS	?	4	0	?	★★★☆☆
BRAVE	0	0	5	4	★★★★☆
CARS	0	0	5	?	★★★★★

Figure 2.1 Ratings table

Applying linear regression on the previously obtained and stored ratings we can predict the possible unknown ratings.

4.3.1. Implementation Steps

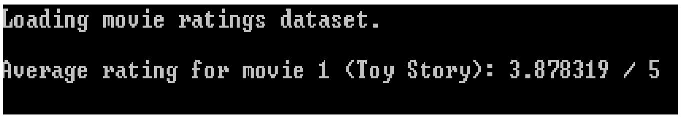
- Loading movie dataset: We will start by loading the movie ratings dataset to understand the structure of the data using load(‘movies.m')



```
ans =

{
  [1,1] = Toy Story (1995)
  [2,1] = GoldenEye (1995)
  [3,1] = Four Rooms (1995)
  [4,1] = Get Shorty (1995)
  [5,1] = Copycat (1995)
  [6,1] = Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
  [7,1] = Twelve Monkeys (1995)
  [8,1] = Babe (1995)
  [9,1] = Dead Man Walking (1995)
  [10,1] = Richard III (1995)
```

Figure 2.2 A part of loading process result

here Y is considered a matrix,containing rating (1-5) and R is a matrix,where R(i,j)=1 if and only if user j gives rating to movie i.From these matrices we can calculate statics like average rating using mean(Y(1,R(1,:)))



```
Loading movie ratings dataset.

Average rating for movie 1 (Toy Story): 3.878319 / 5
```

Figure 2.3 Computation of mean

We can visualize the ratings matrix by plotting it with imagesc function as:

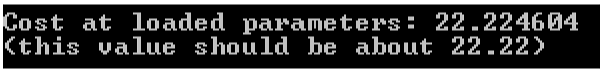
imagesc(Y);

ylabel('Movies')

xlabel('Users')

- Collaborative Filtering: now we implement the collaborative filtering for cost function. The cost function is evaluated using:

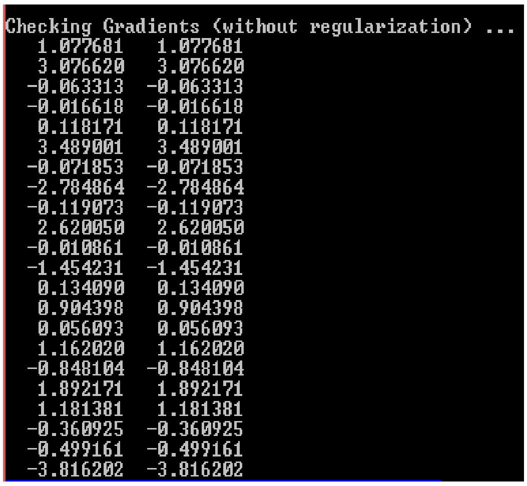
J = cofiCostFunc([X(:) ; Theta(:)], Y, R, num_users, num_movies,num_features, 0);



```
Cost at loaded parameters: 22.224604
(this value should be about 22.22)
```

Figure 2.4 J value

- Collaborative Filtering Gradient: Once our cost function matches with expected value as shown in Fig 2.4,Collaborative Filtering Gradient function should be implemented where in we check Gradients by running checkNNGradients and checkCostFunction. (without using regularization)

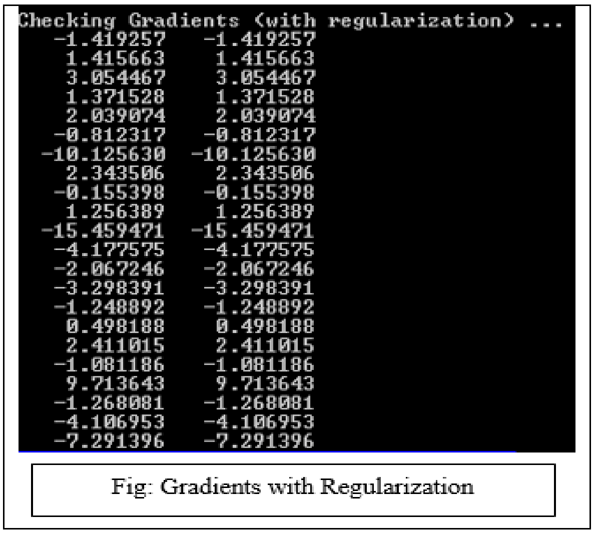


```
Checking Gradients (without regularization) ...
   1.077681   1.077681
   3.076620   3.076620
  -0.063313  -0.063313
  -0.016618  -0.016618
   0.118171   0.118171
   3.489001   3.489001
  -0.071853  -0.071853
  -2.784864  -2.784864
  -0.119073  -0.119073
   2.620050   2.620050
  -0.010861  -0.010861
  -1.454231  -1.454231
   0.134090   0.134090
   0.904398   0.904398
   0.056093   0.056093
   1.162020   1.162020
  -0.848104  -0.848104
   1.892171   1.892171
   1.181381   1.181381
  -0.360925  -0.360925
  -0.499161  -0.499161
  -3.816202  -3.816202
```

Figure 2.5 Gradients without Regularization

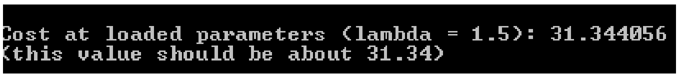
- Collaborative Filtering Gradient with Regularization: now we implement regularization for cost function for collaborative filtering this is done by adding the cost of regularization to the original cost computation. It is evaluated as follows:

J = cofiCostFunc([X(:) ; Theta(:)], Y, R, num_users, num_movies, num_features, 1.5);



```
Checking Gradients (with regularization) ...
    -1.419257    -1.419257
     1.415663     1.415663
     3.054467     3.054467
     1.371528     1.371528
     2.039074     2.039074
    -0.812317    -0.812317
   -10.125630   -10.125630
     2.343506     2.343506
    -0.155398    -0.155398
     1.256389     1.256389
   -15.459471   -15.459471
    -4.177575    -4.177575
    -2.067246    -2.067246
    -3.298391    -3.298391
    -1.248892    -1.248892
     0.498188     0.498188
     2.411015     2.411015
    -1.081186    -1.081186
     9.713643     9.713643
    -1.268081    -1.268081
    -4.106953    -4.106953
    -7.291396    -7.291396
```

Fig: Gradients with Regularization



```
Cost at loaded parameters (lambda = 1.5): 31.344056
(this value should be about 31.34)
```

Figure 2.7 CFG cost

- Collaborative Filtering Gradient Regularization:

As the cost matches as shown in Fig 2.7 we procced to implement regularization for the gradient.

We check the gradient by running:

checkNNGradients checkCostFunction(1.5);

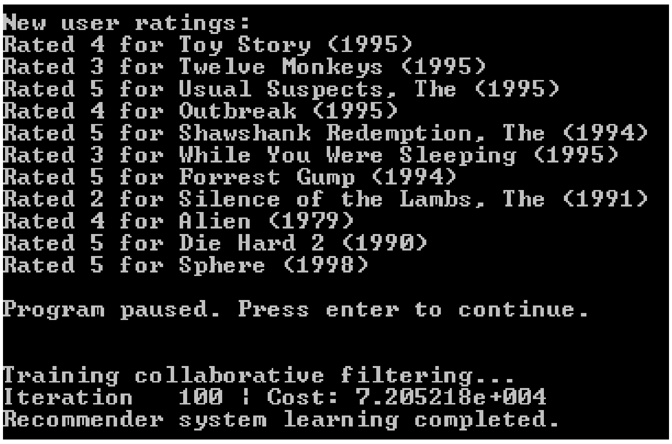
- Enter ratings for a new user: We would train the collaborative filtering model first by adding ratings that correspond to new users, by using:

movieList=loadMovieList();

and we initialize the ratings for the new movies:

my_ratings(u)=v;

where u represents the movie ID and v represents rating(1-5).



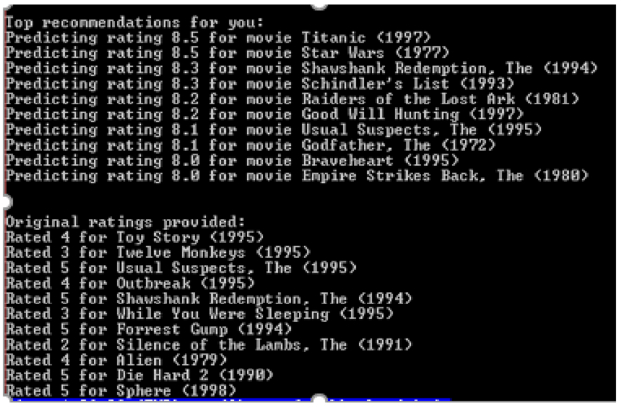
```
New user ratings:
Rated 4 for Toy Story (1995)
Rated 3 for Twelve Monkeys (1995)
Rated 5 for Usual Suspects, The (1995)
Rated 4 for Outbreak (1995)
Rated 5 for Shawshank Redemption, The (1994)
Rated 3 for While You Were Sleeping (1995)
Rated 5 for Forrest Gump (1994)
Rated 2 for Silence of the Lambs, The (1991)
Rated 4 for Alien (1979)
Rated 5 for Die Hard 2 (1990)
Rated 5 for Sphere (1998)

Program paused. Press enter to continue.


Training collaborative filtering...
Iteration   100 | Cost: 7.205218e+004
Recommender system learning completed.
```

Figure 2.8 training collaborative filtering

- Learning movie ratings and recommendations:

Now the collaborative filtering model similarly and complete the recommender system leaning as shown in Fig 2.8.

After obtaining the trained model, now recommendations can be computed using prediction matrix. And hence the following result for recommendation based on original is obtained.



```
Top recommendations for you:
Predicting rating 8.5 for movie Titanic (1997)
Predicting rating 8.5 for movie Star Wars (1977)
Predicting rating 8.3 for movie Shawshank Redemption, The (1994)
Predicting rating 8.3 for movie Schindler's List (1993)
Predicting rating 8.2 for movie Raiders of the Lost Ark (1981)
Predicting rating 8.2 for movie Good Will Hunting (1997)
Predicting rating 8.1 for movie Usual Suspects, The (1995)
Predicting rating 8.1 for movie Godfather, The (1972)
Predicting rating 8.0 for movie Braveheart (1995)
Predicting rating 8.0 for movie Empire Strikes Back, The (1980)


Original ratings provided:
Rated 4 for Toy Story (1995)
Rated 3 for Twelve Monkeys (1995)
Rated 5 for Usual Suspects, The (1995)
Rated 4 for Outbreak (1995)
Rated 5 for Shawshank Redemption, The (1994)
Rated 3 for While You Were Sleeping (1995)
Rated 5 for Forrest Gump (1994)
Rated 2 for Silence of the Lambs, The (1991)
Rated 4 for Alien (1979)
Rated 5 for Die Hard 2 (1990)
Rated 5 for Sphere (1998)
```

Figure 2.9 Recommendation

5. CONCLUSIONS

Machine learning is a method of data analysis that automates analytical model building. Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look. Machine learning solves problem that cannot be solved by other numerical means. We have seen, in this paper by applying linear regression, we can predict the ratings for future movies. This way machine learning helps in improving performance for any such applications. Here we have made use of huge dataset on Hadoop platform which help us process these datasets at a faster rate using MapReduce processing which is not possible by other traditional processing system. Hence Hadoop and Machine Learning together can be used to solve a variety of learning problems more efficiently.

REFERENCES

[1] DunrenChe, MejdlSafran, and Zhiyong Peng, “From Big Data to Big Data Mining: Challenges, Issues, and Opportunities”, DASFAA Workshops 2013, LNCS 7827, pp. 1–15, 2013

[2] Surajit Mohanty, Kedar NathRout, Shekharesh Barik, Sameer Kumar Das, ”A Study on Evolution of Data in Traditional RDBMS to Big Data Analytics”, International Journal of Advanced Research in Computer and Communication Engineering ISSN 2278-1021; Vol 4, pp.230-232, October 2015

[3] Barkha Jain, Manish Km. Kakhani, “Query Optimization in Hive for Large Datasets” ISSN: 2393-9915; Volume 2, Number 4; April-June, 2015 pp. 321-325

[4] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[5] https://cwiki.apache.org/confluence/display/Hive/Home;jsessionid=986ED81CB5EB4E18AB21A6BDDE4610EA

[6] Surekha Sharad Muzumdar, Jharna Majumdar, “Big Data Analytics Framework using Machine Learning on Multiple Datasets” IJSR ISSN:2319-7064 Volume 4 Issue 8, August 2015 pp.414-418

[7] http://grouplens.org/datasets/movielens/

[8] http://www.recsyswiki.com/wiki/MovieLens

[9] http://files.grouplens.org/datasets/movielens/ml-latest-README.html

[10] http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

[11] https://en.wikipedia.org/wiki/Machine_learning

[12] https://www.gnu.org/software/octave/

[13] Neha Mangla, K. Sushma and Lithin Kumble," IPB - Implementation of Parallel Mining for Big Data", Indian Journal of Science and Technology, Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106497, December 2016

[14] Neha Mangla," Machine Learning Approach for Unstructured Data Using Hive", International Journal of Engineering Research, Volume No.5 Issue: Special 4, pp: 790-991

[15] Neha Mangla “A Comprehensive Review: Internet of Things (IOT)” , IOSR Journal of Computer Engineering (IOSR-JCE) ,e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 4, Ver. III (Jul.-Aug. 2017), PP 62-72

[16] Neha Mangla,” A Gps-Gsm Predicated Vehicle Tracking System, Monitored In A Mobile App Based On Google Maps” International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) 2017, IEEE, 7July,2017.

[17] Neha mangla,” A Practical Approach For parallel Image Processing”, IOSR-JCE,Volume 18,Issue 2,Pages 82-92, March 2016

[18] Neha Mangla, Sushma K,” EPH-Enhancement of Parallel Mining using Hadoop”, International Journal of Engineering Research ISSN:2319-6890)(online),2347-5013(print) Volume No.5 Issue: Special 4, pp: 790-991 20 May 2016.

[19] Tripti Mehta, Neha Mangla,” A Survey Paper on Big Data Analytics using Map Reduce and Hive on Hadoop Framework”, International Journal of Recent Advances in Engineering & Technology (IJRAET),Volume 4,Issue2,Pages 112-118

[20] Neha Mangla,” Analysis of Diabetic Dataset and Developing Prediction Model by using Hive and R*”,* Indian Journal of Science and Technology, Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106497, December 2016.

[21] Mangla N and Khola R.K ” Optimization of IP Routing with Content Delivery Network” published in “Networking and Information Technology (ICNIT), 2010 InternationalConference “of IEEEExplore,June11-122010, E-ISBN : 978-1-4244-