**Analyzing Motor vehicle crashes in the state of New York using Big Data Solutions**

Santosh Muniswamygowda Nagaraja, UBID- 1015433  
Department of computer science, University of Bridgeport

***Abstract –* Today we live in a modern world where time is money, everyone is trying to get to their work in rush. Which is one of the major contributors for road accidents. Even though the Governments has a strict traffic rules and regulations there are still thousands of casualties each year. In the year of 2017 there was a total of 40,000 deaths in USA due to road accidents. The government comes up with various solutions and successfully reduces the road accidents. But the problem is that the reduction in total accidents is very low each year about 1% to 3% compared to previous years. One solution to decrease the total number of accidents is to go for big data solutions. Since, the various factors for a road accident are heterogeneous, analyzing every accident becomes challenging. Also, another major challenge analyzing the road accidents is that most of the road accidents goes unregistered. Even though there is a record there will be a number of missing information about the accident.**

# 1. INTRODUCTION

The safety of a road can be measured by analyzing the total number of accidents that have occurred in that road. The Road is not only the major contributor of the accident in a location, there is the electronic traffic signals and road signals that also contribute in the road accidents. A faulty road traffic signal for just a second can cause a road accident. There are several parameters that affect the road accidents, some of the major parameters are faulty traffic signals, faulty road signs, the state of the driver, speed limit in the road, changing lanes and neighboring drivers. One of a good solution for the future is autonomous driving. The chances of a machine causing an error is typically less compared to a human being. So, for the future autonomous driving sounds good, but as of for now the technology is far behind for autonomous driving. Although companies like Tesla are coming forward with autonomous vehicles, even these vehicles have a long way to go. The tesla models are still not able to read the road signs properly, they can just change lanes and go on autonomous driving in highways with proper road signs and road lane paints. The others solution for decreasing the accidents is to come up with various solutions by analyzing the previous accident database. But the problem with this is that the data base is not 100% reliable, and most of the road accidents are not usually reported. And the reported accidents usually have number of the information missing. In our solution to this problem is using big data we have chosen the New York database of road accidents over the span of 2 years. In our analytics to analyze the database we are implementing MapReduce programming, Hive, Spark and Big Query to determine various factors for a road accident in a given location.

# METHODOLOGY

First step is to obtain a database of road accidents, there are many data bases available but for this project we have chosen the data base provided by the state of New York which contain the accident data base of over the span two years. The data base is obtained from the website data.gov. Once the data is obtained the next step is to preprocess the data. Check the data base for any null values or duplicates in the database. If there is a record of accident with no location of the accidents, then record might not be much of use for this project, similarly if there is a accident recorded without the factors for accident, it will also not be much of useful for our project. So, to filter all these unwanted records, we decided to use the power of Big Query, with the help of SQL queries. Once we have the pre-processed data the next step is to perform analytics of the data. In the process of analytics first we need to determine the list of various locations of the accidents, Next is to list out the various factors affecting the accidents. Once we have these two, we need to check each factor and how many accidents it has caused in each location, this is done using Spark. The final step is to come on with solutions for the accidents for various factors caused the accidents. Using Spark, we plot graphs using the outputs obtained by Map reduce programming

# IMPLEMENTATION

The data base selected for this project consists of about 1.26 million records of previous accidents in the state of NYC. The data base is about 500MB obtained from data.gov website. As said earlier the next step is to pre-process the data. We have selected the Big Query to pre-process the data. To pre-process the data first we need to upload the data into google cloud and select the big query tool to perform various analytics on the database. First step in pre-processing involves removing all the null values in the database. We have found out that out of 1.26 million records in the database about 40000 records had to be neglected because of invalid data or no proper information about the accident. Once we have the data base the next step is to setup Hadoop ecosystem for big data analysis. For this project we have installed Hadoop in Ubuntu 16.04 version.

# HADOOP SETUP

The first step is to download apache Hadoop for ubuntu. The latest version of Hadoop available in apache is Hadoop 3.1.0. For this project we have selected the version 2.7.1 which is a stable version released in 2016. We can directly download the latest version of Hadoop from the website apache Hadoop. Once downloaded open terminal and execute the first command $sudo apt-get update, this command updates the system to the latest version. Next step is to install java since the Hadoop ecosystem is written in java we need java to implement Hadoop file system HDFS. We can install java with the following command $sudo apt-get install openjdk-7-jdk. Next step is to create a open ssh server, we can install the ssh server with the following command $sudo apt-get install openssh-server. Next step is to create a ssh key this can be created with the following commands $ssh0keygen-t rsa -P “”, $cat $HOME/.ssh/id\_rsa.pub>> $HOME/.ssh/suthorized\_keys. Next step extract the apache Hadoop file and place it in the home folder under the name Bigdata. Once we have confirmed that the data is completed its pre-processing stage and all the database can be used for further processing we upload the database into HDFS using the HDFS “-hdfs fs -copyFromLocal” command. Next update the “.bashrc” file to setup environmental variable for Hadoop home. So first we need to go to home directory. Type cd home in the terminal to change the directory to home. To edit the bash file type $sudo gedit. bashrc, now in the text file. bashrc file type the following,

#set JAVA\_HOME

Export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64.

Export PATH=”$PATH:$JAVA\_HOME/bin”

The first export is the path for the java jdk file which is available in the location usr/lib/jvm/ folder. Now we must set environmental variable for the Hadoop system

# Set Hadoop-related environment variables

export HADOOP\_HOME=$HOME/Bigdata/hadoop-2.7.1

export HADOOP\_CONF\_DIR=$HOME/Bigdata/hadoop-2.7.1/etc/hadoop

export HADOOP\_MAPRED\_HOME=$HOME/Bigdata/hadoop-2.7.1

export HADOOP\_COMMON\_HOME=$HOME/Bigdata/hadoop-2.7.1

export HADOOP\_HDFS\_HOME=$HOME/Bigdata/hadoop-2.7.1

export YARN\_HOME=$HOME/Bigdata/hadoop-2.7.1

# Add Hadoop bin/ directory to PATH

export PATH="$PATH:$HOME/Bigdata/hadoop-2.7.1/bin"

Once all the data is entered we need to save the bash file, close the. bashrc file and execute the command $source .bashrc. Next step is to configure the Hadoop files to setup namenode, datanode, yarn resource manager and other Hadoop configuration files. First Go to Bigdata/hadoop-2.7.1/etc/hadoop/ directory. Open hadoop-env.sh in gedit and update

JAVA\_HOME to:

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-i386

Create DataNode and NameNode directories to store HDFS data.

$mkdir -p $HADOOP\_HOME/hadoop2\_data/hdfs/namenode

$mkdir -p $HADOOP\_HOME/hadoop2\_data/hdfs/datanode

Go to Bigdata/hadoop-2.7.1/etc/hadoop/ directory. Open hdfs-site.xml in gedit and update

Configuration to:

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>/home/honey/Bigdata/hadoop-

2.7.1/hadoop2\_data/hdfs/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>/home/honey/Bigdata/hadoop-

2.7.1/hadoop2\_data/hdfs/datanode</value>

</property></configuration>

Go to Bigdata/hadoop-2.7.1/etc/hadoop/ directory. Open yarn-site.xml in gedit and update

Configuration to:

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux- services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

Next Open Terminal and go to hadoop directory using following command:

$ cd $HADOOP\_HOME/etc/Hadoop

Next create a new mapred-site.xml file with the following command $cp mapred-site.xml.template mapred-site.xml. Go to Bigdata/hadoop-2.7.1/etc/hadoop/ directory. Open mapred -site.xml in gedit and

update Configuration to:

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

Next to format the file-system, run the command:

$cd

$hadoop namenode –format.

Next to the start the Hadoop system go the Hadoop sbin directory with the following command.

$cd HADOOP\_HOME/sbin

Not type $./start-all.sh to start all the Hadoop services. Once executed to check if the Hadoop service are up and running type jps in the terminal. We must see the the datanode, namenode, jps, nodemanager, resourcemangarer, and jobhistoryserver. Now all the Hadoop evnvironment is setup successfully. Now we need to upload the data to the Hadoop file system. To do that type $hadoop fs -copyFromLocal path\_of \_database path\_in\_HDFS to upload the data. Once the data is uploaded we can verify that the data is uploaded using the local host HTML viewer of the Hadoop data base file system HDFS using the local host <https://localhost:50070>. Once we log into the localhost we can select the “dfs health” check-up to check if the database is properly uploaded into the database. Once we have successfully uploaded the data we need to write MapReduce programming to perform the further analytics.

# MAP-REDUCE

For our project we have selected to use python as the programming language to code the MapReduce programs. To use python programming, we must first install python in the Linux system. Python can be installed in Linux by downloading the latest version 3.6 from the python official website python.org. Since, we are using Ubuntu are as our Linux system to setup the HDFS, there is no need to install python. Latest version of python is installed by default in ubuntu systems. We have used python version 3.6 to code our MapReduce programs. To check the active version of python running in Ubuntu we can run the command “python -V” or “python –version” or “apt-cache policy python”. For every MapReduce we must select a Key and value, the keys are mapped using the mapper.py file which contains the mapper file and the values are aggregated using the reducer.py file.

For our project’s first map reduce program we have used the first key value as the location of the accident. The location of the accident is present in the second column of the database. Hence in the mapper program we have to select record [2] as our key value which the location of the accident. Once we the list of all the keys, we need to send it the reduce where shuffle, sort and aggregation is performed on the results obtained from the mapper.py file.

MapReduce program are by default written in Java programming. In order to execute the MapReduce program in python we have two options either install a tool for big data called as the “anaconda” or use a inbuilt tool in the HDFS system called as the Hadoop streaming jar file. The command needed to execute a python program in the HDFS file system is as follows:

“hadoop jar /user/lib/Hadoop-0.2.7-mapreduce/contrib/streaming/Hadoop-streaming-2.1.7-mrl-cdhr-4.4.0.jar” followed by -file /location of the mapper file -mapper “python mapper.py” -reduce /location of the reducer file “python reduce.py” followed by the input file -input /location of the input file that is in the HDFS file system. Once the execute the following command with appropriate file locations with the record column number as 2 for the location of the accident for found of the list of all the location with the total number of accidents in that particular location. The output is usually stored in the HDFS file system as file name /part-0000. The file can be downloaded with the help of the local host file viewer or we can use the terminal command to download the file. The output file contains

LOCATION COUNT

Bronx 355086  
Brooklyn 272651  
Manhattan 222055  
Queens 231991  
Staten Island 40566

From the output we can clearly see that the total number of accidents in Bronx is more than compared to all others. The next step is to determine the various factors and number of times the accidents caused due to this factor. The factors for the accidents are present as in the database in the column number 19 as accident factor 1, number 20 as accident factor 2 and column 21 as accident factor 3. So, the next MapReduce program contain the key value as the record [19] and the next the key value will be record [20] and the next key value will be record [21]. Some of the combined output obtained form the above MapReduce programs is as follows.

Accelerator Defective 472  
Aggressive Driving/Road Rage 3622  
Alcohol Involvement 9076  
Animals Action 615  
Backing Unsafely 34924

# HIVE

Once we have the outputs from the MapReduce we must next process the outputs using HIVE. In hive we mix and map the location and various factors affecting the accidents. From the first output of the MapReduce program we have all the location with the count of the accidents, in the next output we have all the factors for the accidents with the count of the accidents. Now using HIVE, we can determine all the accidents in a location due to a factor for the accident. For example, we selected Staten Island for our first HIVE program. The total accident caused in Staten Island is about 40566 over the span of two years. Next, we found out various factors that affect an accident in Staten Island we found out that

Accelerator Defective 19  
Aggressive Driving/Road Rage 567  
Alcohol Involvement 206  
Animals Action 17  
Backing Unsafely 4012

This is just the first five records of the output from the output we can easily come to know that there are corporately very less accidents in Staten Island due to Road rage and Aggressive driving. But the Backing Unsafely has over more than 4000 records for the accidents.

Another major attribute in the database was the longitude and the latitude values in each record. In the project I implemented a MapReduce program to find out the total number of accidents in each location using longitude and latitude values. The results were very useful. We found out about 140 different points where the number of accidents were very high. For example, at a location in Bronx New York with the longitude and latitude values of 40.7606005, -73.9643142 a street near crunch 59th street Bronx NY has a total of 781 recorded accidents in the past 2 years. Most of the accidents involves driver unconscious, drink and drive and driver under influence of drugs. By analyzing this we can reduce the accidents at this location by adding more traffic cop at this location. Like wise there are many locations found for each longitude and latitude values.

# Conclusion

We consider our project to be very important because, our project can save lives, we can prevent accidents before then even happen by taking proper actions, for example from our project we have found out that drink and drive accident are very high in Manhattan and Bronx so if the government strengths the security by traffic cops in Manhattan and Bronx location, we can prevent many accidents. And, there are many accidents happen in Bronx in the past year due to faulty electronic signal and road signs. We can prevent his by taking proper action and fixing the road signs and fixing the road signals. Like wise in the project we have found more than 140 locations with specific longitude and latitude values where the number of accidents is very high. Now we know the reasons and location for the accidents, now we must come up with solution for preventing these in the future. We must strengthen traffic cops where there are more number of accidents caused due to driver under influence and under age driving. Fix some of the electronic traffic signals where accidents are occurring due to faulty traffic signals.

REFERENCES

[1] Tessa K. Anderson, “Kernel density estimation and K-means clustering to profile road accident hotspots”, Accident Analysis and Prevention 41 (2009) 359–364.  
[2] Sachin Kumar, Durga Toshniwal, “A data mining framework to analyze road accident data”,  
[3] Miao Chong, Ajith Abraham2 and Marcin Paprzycki1, “Traffic Accident Analysis Using Machine Learning Paradigms”