PROJECT REPORT

ON

“Travel Data Analysis”

Prepared by

PRAYAG SINGH (3331)

MANDEEP SINGH (3323)

PAWAN KUMAR (3329)

Supervised by

Mr. P. Sonawane

Onward to Glory

DEPARTMENT OF COMPUTER ENGINEERING

(2016-2017)

ARMY INSTITUTE OF TECHNOLOGY

Pune, Maharashtra, India

TABLE OF CONTENTS

ACKNOWLEDGEMENT

To acknowledge and thank every individual who directly or indirectly contributed to this venture personally, it would require an inordinate amount of time. We are deeply indebted to many individual, whose cooperation made this job easier.

We avail this opportunity to express our gratitude to our friends and our parents for their support and encouragement throughout project. We feel it is as a great pleasure to express our deep sense of profound thank to Mr. P. Sonawane, who guided us at every step and also encouraged us to carry out the project.

PRAYAG SINGH (3331)

MANDEEP SINGH (3323)

PAWAN KUMAR (3329)

ABSTRACT

The travel dataset is publicly available and the contents are detailed under the heading, ‘Travel Sector Dataset Description’.
Based on the data, we will find the top 20 destination people travel the most, top 20 locations from where people travel the most, top 20 cities that generate high airline revenues for travel, based on booked trip count.

Top 20 destination people travel the most: Based on the given data, we can find the most popular destination that people travel frequently. There are many destinations out of which we will find only first 20, based on trips booked for particular destinations.

Top 20 locations from where people travel the most: We can find the places from where most of the trips are undertaken, based on the booked trip count. Top 20 cities that generate high airline revenues for travel, so that the site can concentrate on offering discount on booking, to those cities to attract more bookings.

In this paper, we have proposed a Hadoop framework to analyze travel and tourism data. This framework segments the time series data into different clusters. A time series merging algorithm is proposed to find the representative time series (RTS) for each cluster. This RTS is further used for trend analysis of different clusters. The results reveal that road travel and tourism trend is going to increase in certain cluster and those cities which are less visited should be concerned and we can increase the tours of people at the particular place by providing certain picnic and family tour packages.

TRAVEL AND TOURISM DATA ANALYSIS

SYNOPSIS

These are the certain recent reports.

Sojern is collecting and aggregating information across airlines, hotels, rental car agencies and credit card companies and is using machine learning and advanced analytics to develop rich profiles of segments of travelers to determine "when people go, where they go, how many people are travelling, and preferred brands, travel times and class of service," said Rick Farnell, president and co-founder of Think Big Analytics, the Mountain View, Calif.-based consulting firm that helped Sojern with its analytics. Sojern's insights are used by the very same airlines, hotel chains and car rental agencies to hone their pricing and selection of services. "Customers like Delta or Starwood can find all the business travelers that flew between New York and San Francisco over the last month, and it helps them make the right cross-sell and up-sell offers and shape their inventories.

Today travel providers have vast amount (petabytes) of data on every step that their customers take during their travel booking cycle. But the main challenge they face is converting this data into value for customers. The right solution can open up immense opportunities for all – the customers, the travel providers and the solution providers.

Data mining techniques have certain advantages over traditional statistical techniques. Data mining techniques do not require certain assumptions between dependent and independent variables which are required in traditional statistical techniques. Also, data mining techniques is capable of handling large dimensional data whereas statistical techniques have some limitations.

Time series constitute a series of data points collected or sampled at fixed interval. Monthly travelling by people counts of travel packages for a certain period of time also constitute a time series data. The time series data of tour and travel is very important to study as it can reveal the future trend of travel and tourism. This future trend can help in identifying the least visited place and measures can be taken to increase the place's popularity by providing discount and various other offers.

Those techniques can certainly reveal the hidden factors behind travel and tourism, but they cannot reveal the trend of the travel and tourism in different locations. In this manuscript, we are trying to elaborate travel and tourism counts data to identify different places where the trend of visiting is increasing throughout the years. So that less focus would be on these places to overcome the less visiting trend. In order to do this, we have proposed a framework to analyze travel and tourism time series data that uses both data mining and traditional statistical techniques. This framework inputs the tour counts for different time series and then normalizes the time series. Further a time series merging algorithm is proposed to find the representative time series (RTS) for each cluster. Finally, trend analysis is performed on every RTS of different clusters.

HOW CUSTOMER GENERATES DATA:

- Browsing through travel service provider‘s website, traversing through OTA sites, clicking on ads on social media sites like facebook, pinterest, using marketing promotions by travel service providers, using travel search sites, online searching and travel blogs and publications
- Calling up the travel providers‘ call centers, browsing OTA sites, surfing websites of travel service providers, reading other travelers‘ experiences on social media sites and blogs, surfing through travel review sites such as Tripadvisor, seeing pictures and videos
- Travel service provider‘s website/call center/social media page/on property, travel agency, OTA
- Traveler‘s on property feedback, online reviews/complaints registered via call centers, site, his movements, time spent on the service

WHAT VALUE CAN BE GENERATED THROUGH THIS DATA?

The good news is that the data about customer interaction sits with you. But the big question is – what to do with this big data that can deliver value to the end customer and the travel service provider as well. The following represents a most impactful of the many benefits that can be derived by analyzing and making this big data work for you:

1.Personalization – Gone are the days of profiling, when the customers were segregated into different categories and offers were made for a particular category. Present day customers are becoming more demanding and the competition is growing each day. To retain the loyalty of its customers a service provider needs to stitch an offer which meets their individual requirements - a truly personalized offer. This personalization has to be based on:

a)Past behaviors: *"We see that you like an aisle seat with extra leg space. Book a seat on our economy plus class which has extra leg space and also get 30% off on selection of aisle seat."*

b)Social media relationships: *"50 of your friends have flown with us this month. Know what they have to say about us and give us a chance to serve you!"*

c)Location: *"Want to try the best local cuisine? Visit "The Alpines", just two blocks away from your hotel."*

d)Itinerary: *"We see that you are flying from Peru to Costa Rica next week. Would you like to try our Limo service for the airport transits?"*

e)Ancillary Sales: *"Did shopping from the beautiful streets of Paris. Don't worry book for an extra bag now and get 20% off on the extra baggage allowance!"1* Today travel firms need to embrace —**Next Best Offer**‖ analytics (Next Best Offer refers to the use of predictive analytics to identify the product or services that the customers, based on their past purchase behavior, are most likely to buy in their next purchase) to reach their customers at the right time, at the right place, through the right channel and most importantly with the right offer.

2.Enhanced Customer experience: A key factor in the success of any service organization is the ‗customer happiness index' and travel service providers are no exception to this rule. The key differentiator which can help a service provider stand out is the quality of customer experience that it can offer. By combining customer interaction footprints through all the channels, a firm can get a 360-degree view of its customers. This comprehensive knowledge can help the firm make better suited offers for its customers, provide additional services which can increase customers' delight or simply make a kind gesture which can touch its customers. All these actions go a long way in ensuring that your customers remain loyal to you. An example of the customer delight that can be created is the —KLM Surprise‖ campaign. In November 2010, the international airline KLM surprised its customers: As passengers arrived at security checkpoints and gates, flight attendants were there to greet them by name and give them a personalized gift — something that the passenger could use on his or her trip, or enjoy when they returned home. Flight attendants browsed Twitter and Foursquare, looking for people who mentioned that they were taking a KLM flight. Then, using the information the customer provided about him or herself on social media platforms, the flight attendants purchased a suitable gift and presented it to the passenger upon his/her arrival at the airport.

As a result of this campaign, news of the KLM surprises spread like wildfire through social media - mentions, tweets, retweets, and word-of-mouth. That November, the KLM Twitter feed was viewed more than million times. What made this particular campaign stand apart from other run-of-the-mill marketing campaigns is personalization - which enabled the airline to offer customers something that held real, tangible value.

3.Identifying most valuable customers: The major chunk of value from harnessing big data analytics can be derived by identifying the most profitable customers. Marketing efforts can be directed to capture their attention, the offers can be better suited to their needs, and efforts can be made to drive loyalty. This is more important keeping in mind the fact that cost of new acquisition is higher than the cost of retention.

4.Revenue generation through Cross-sell and Up-sell: Inventory is only a part of the complete suite of services that a travel firm provides. There is a complete range of ancillary services which often get neglected. This is primarily due to incomplete customer behavior profile, because of which the seller, be it through OTA or any other channel, is not in a position to anticipate the right ancillary service that the customer is most likely to buy and hence misses out on a great revenue generation opportunity. By analyzing the past behavior, social media activities and online behavior of the customers, the travel service firms can target the right cross-sell and up-sell opportunities and the right channel as well. *For e.g. – If a customer always books an aisle seat in the airplane, the next time he books an air ticket, the airline or the OTA can offer him pre-booking option for the aisle seat upfront.*

5.Targeted marketing: Marketing ROI is a figure which often raises eyebrows in the boardrooms. A high ROI tops every marketer's wish list. Key techniques for a high ROI demonstrate a campaign intelligently targeted to a specific set of customers, executed at the right time and delivered through the right channel. Consider a scenario where Julie, searches Tripadvisor for customer reviews on Spain and accommodation. Being a heavy Facebook user, she checks out the Facebook page for Spain travel & tourism. Based on Julie's online activity, an airline posts an offer for a flight to Barcelona on Julie's facebook profile page. It's obvious that the chances for Julie to click on that advertisement would be much higher than clicking a promotional offer for flights to Prague.

TECHNICAL DETAILS:

Our project will use latest upcoming techniques in the big data field for example we have hive, map reduce, pig etc. The main steps required for the transformation of data will be as follows:

1. **Acquisition:** Collecting the required data from various travel and tourism industries. This data is the data that has been made available to the public and is available online. This data does not a defined structure and is in semi structured format.

2. **Transforming and Preprocessing:** Cleansing and filtering the data

3. **Processing:** Running analysis on the raw data, for this purpose we have used pig.

4. **Frequencies and Analysis:** Calculating frequencies and occurrences this can be done with the help of excel. This is to give a manageable and easily interpretable data to the respective parties.

5. **Mining:** Extracting the insights and findings, this is done by analytics experts and these interpretations are then used when making the companies policies.

While this is by no means a linear or complete order, it will help you frame your own projects.

For the purpose of the project we shall be using a virtual machine running the required image due to the restrictions on the availability of multiple high end devices.

Main topics that one should be aware of the Hadoop Distributed File System(HDFS), basic Unix commands, map reduce, Apache Pig etc.

Hadoop Distributed File System: The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks.

Map reduce: MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster.

Apache Pig: It is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

CURRENT STATUS AND DEVELOPMENT:

- Development of prototype complete.
- Testing of various variables using prototype is going on.

MARKET POTENTIAL:

- **Customer engagement.** Big data can deliver insight into not just who your customers are, but where they are, what they want, how they want to be contacted and when.

- **Customer retention and loyalty.** Big data can help you discover what influences customer loyalty and what keeps them coming back again and again.

- **Marketing optimization/performance.** With big data, you can determine the optimal marketing spend across multiple channels, as well as continuously optimize marketing programs through testing, measurement and analysis.

- **Competitor analysis.** With this we can achieve an advantage over our competitors by offering the content that the consumer wants to watch.

Hadoop: Setting up a Single Node Cluster.

Purpose

This document describes how to set up and configure a single-node Hadoop installation so that you can quickly perform simple operations using Hadoop MapReduce and the Hadoop Distributed File System (HDFS).

Prerequisites

Supported Platform

- GNU/Linux is supported as a development and production platform. Hadoop has been demonstrated on GNU/Linux clusters with 2000 nodes.
- Windows is also a supported platform but the followings steps are for Linux only.
  Required Software

Required software for Linux include:

1. Java™ must be installed. Recommended Java versions are described at HadoopJavaVersions.
2. ssh must be installed and sshd must be running to use the Hadoop scripts that manage remote Hadoop daemons.

Installing software

If your cluster doesn't have the requisite software you will need to install it. For example on Ubuntu Linux:

```
$ sudo apt-get install ssh
$ sudo apt-get install rsync
```

Download: To get a Hadoop distribution, download a recent stable release from one of the Apache Download Mirrors.

Prepare to Start the Hadoop Cluster: Unpack the downloaded Hadoop distribution. In the distribution, edit the file etc/hadoop/hadoop-env.sh to define some parameters as follows:

```
# set to the root of your Java installation
export JAVA_HOME=/usr/java/latest
```

Try the following command:

```
$ bin/hadoop
```

This will display the usage documentation for the hadoop script.Now you are ready to start your Hadoop cluster in one of the three supported modes:

- Local (Standalone) Mode
- Pseudo-Distributed Mode
- Fully-Distributed Mode

Standalone Operation: By default, Hadoop is configured to run in a non-distributed mode, as a single Java process. This is useful for debugging.

The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.

```
  $ mkdir input
  $ cp etc/hadoop/*.xml input
  $ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar grep input
output 'dfs[a-z.]+'
  $ cat output/*
```

Pseudo-Distributed Operation

Hadoop can also be run on a single-node in a pseudo-distributed mode where each Hadoop daemon runs in a separate Java process.

Configuration

etc/hadoop/core-site.xml:

```
<configuration>

<property>

<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

etc/hadoop/hdfs-site.xml:

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

Setup Passphraseless ssh

Now check that you can ssh to the localhost without a passphrase:

```
$ ssh localhost
```

If you cannot ssh to localhost without a passphrase, execute the following commands:

```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

Execution

The following instructions are to run a MapReduce job locally. If you want to execute a job on YARN, see YARN on Single Node.

1. Format the filesystem:
2. `$ bin/hdfs namenode -format`
3. Start NameNode daemon and DataNode daemon:
4. `$ sbin/start-dfs.sh`

   The hadoop daemon log output is written to the $HADOOP_LOG_DIR directory (defaults to $HADOOP_HOME/logs).

5. Browse the web interface for the NameNode; by default it is available at:
   - NameNode - http://localhost:50070/
6. Make the HDFS directories required to execute MapReduce jobs:
7. `$ bin/hdfs dfs -mkdir /user`
8. `$ bin/hdfs dfs -mkdir /user/<username>`
9. Copy the input files into the distributed filesystem:
10. `$ bin/hdfs dfs -put etc/hadoop input`
11. Run some of the examples provided:
12. `$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar grep input output 'dfs[a-z.]+'`
13. Examine the output files: Copy the output files from the distributed filesystem to the local filesystem and examine them:
14. `$ bin/hdfs dfs -get output output`
15. `$ cat output/*`
16. When you're done, stop the daemons with:
17. `$ sbin/stop-dfs.sh`

YARN on a Single Node

You can run a MapReduce job on YARN in a pseudo-distributed mode by setting a few parameters and running ResourceManager daemon and NodeManager daemon in addition.The following instructions assume that 1. ~ 4. steps of the above instructions are already executed.

1. Configure parameters as follows:etc/hadoop/mapred-site.xml:
2. <configuration>
3. <property>
4. <name>mapreduce.framework.name</name>
5. <value>yarn</value>
6. </property>
7. </configuration>

   etc/hadoop/yarn-site.xml:

   ```
   <configuration>
   <property>
   <name>yarn.nodemanager.aux-services</name>
   <value>mapreduce_shuffle</value>
   </property>
   </configuration>
   ```

8. Start ResourceManager daemon and NodeManager daemon:
9. $ sbin/start-yarn.sh
10. Browse the web interface for the ResourceManager; by default it is available at:
    - ResourceManager - http://localhost:8088/
11. Run a MapReduce job.
12. When you're done, stop the daemons with:
13. $ sbin/stop-yarn.sh

.

STEPS IN THE PROJECT:

1) Data file preparation and preview: -

Pig being a query based system, a basic understanding of the data file is important. Before moving forward let us go through a sample of the data used for the trend analysis.

```
ZIH-ZIH      ZIH   ZIH   4     2     0     0     0     0     2014-10-23
00:00:00.000   2014-10-25 00:00:00.000   0     0     2003.19995117188
        Viceroy Zihuatanejo

ZIH-ZIH      ZIH   ZIH   4     2     0     0     0     0     2014-10-23
00:00:00.000   2014-10-25 00:00:00.000   0     0     1556.76
        Capella Ixtapa Resort & Spa

YYZ-YYZ      YYZ   YYZ   4     1     2     0     0     0     2014-11-24
00:00:00.000   2014-11-26 00:00:00.000   0     0     268.02                  Hilton
Hotel and Suites Niagara Falls/Fallsview

YYZ-YYZ      YYZ   YYZ   4     1     2     0     0     0     2014-11-24
00:00:00.000   2014-11-26 00:00:00.000   0     0     674.36                  Hilton
Hotel and Suites Niagara Falls/Fallsview

YYZ-YYZ      YYZ   YYZ   2     0     1     0     0     0     2014-11-11
08:00:00.000   2014-11-14 08:00:00.000   0     254.19 0               ZL

YYZ-YYZ      YYZ   YYZ   4     1     0     0     0     0     2015-02-16
00:00:00.000   2015-02-22 00:00:00.000   0     0     916.46                  Hilton
Suites Conference Centre and Spa

YYC-YYC      YYC   YYC   4     1     0     0     0     0     2014-02-03
00:00:00.000   2014-02-08 00:00:00.000   0     0     903.36                  Delta
Bow Valley

YYC-YYC      YYC   YYC   4     2     0     0     0     2     2014-06-21
00:00:00.000   2014-06-25 00:00:00.000   0     0     1107.14
        The Rimrock Resort Hotel
```

```
YVR-YYZ      YVR   YYZ   7       1       1       0       0       0       2015-01-12
00:00:00.000  2015-01-15 00:00:00.000      4351.4 176.85 238.14 AC      Air Canada      ZT
          Comfort Hotel Airport North

YVR-YYZ      YVR   YYZ   7       1       1       0       0       0       2015-01-12
00:00:00.000  2015-01-15 00:00:00.000      4351.4 176.85 281.4354       AC      Air      Canada
          ZT      Novotel Toronto Mississauga Centre

YVR-YVR      YVR   YVR   2       1       0       0       0       0       2014-12-09
08:00:00.000  2014-12-11 08:00:00.000      0       74.97  0                       AL
```

The basic scheme in the above sample and the data file used in this project is:

(City pair (Combination of *from* and *to*): String), (From location: String), (To Location: String), (Product type: Integer (1=Air, 2=Car, 3 =Air+Car, 4 =Hotel, 5=Air+Hotel, 6=Hotel +Car, 7 =Air+Hotel+Car)), (Adults traveling: Integer), (Seniors traveling: Integer), (Children traveling: Integer), (Youth traveling: Integer), (Infant traveling: Integer), (Date of travel: String), (Time of travel: String), (Date of Return: String), (Time of Return: String), (Price of booking: Float), (Hotel name: String)

Note: -

Pig can work on semi structured, structured and non-structured files in the above case we have a semi structured file. Before loading a semi-structured file into pig {using PigStorage ()} it is important to prepare the file so that it has a rough structure for example in the above case each field is separated by a space. Thus when we are loading the file we can specify that the fields are separated by spaces, in some other files the separators may be "&", "&&", "," etc. these tell pig to separate the values of one field from the next when it encounters them. These symbols are called delimiting symbols.

Sometimes the file may need to be prepared before loading for example the file may have a combination of two delimiting characters then one of the characters may be needed to be replaced by the other in order for the semi structured file to be correctly loaded.

For example: Data before any changes (it has two delimiters, and &)

-Field1, field2 & field3

: After changes

-Field1 & field2 & field3

Now a single delimiter is present in the file. This can be achieved by using the **'sed'** utility in the Linux system.

2) LOADING THE FILE INTO THE HDFS: -

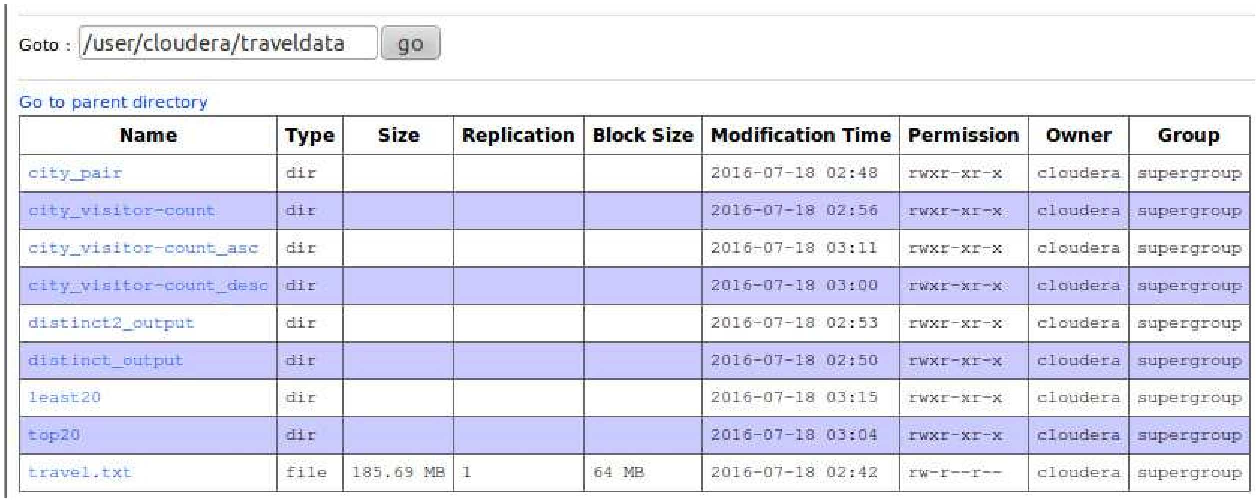
Since pig in its map reduce mode works in the Hadoop file system we need to move the data file into the HDFS partition. This can be achieved by running the put command. For the project we have put the file into a directory named training so first we shall make the desired directory.

hadoop fs -mkdir /user/cloudera/traveldata

next we shall load the data file named "travel.txt" into the directory.

hadoop fs -put '/home/cloudera/Desktop/travel.txt' '/user/cloudera/traveldata'

This command has two parts first is the path name (home/cloudera/Desktop/travel.txt) of the file to be put in the Hadoop file system second is the path (/user/cloudera/traveldata) of the location where the file is to be loaded.



Goto : /user/cloudera/traveldata go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
city_pair	dir				2016-07-18 02:48	rwxr-xr-x	cloudera	supergroup
city_visitor-count	dir				2016-07-18 02:56	rwxr-xr-x	cloudera	supergroup
city_visitor-count_asc	dir				2016-07-18 03:11	rwxr-xr-x	cloudera	supergroup
city_visitor-count_desc	dir				2016-07-18 03:00	rwxr-xr-x	cloudera	supergroup
distinct2_output	dir				2016-07-18 02:53	rwxr-xr-x	cloudera	supergroup
distinct_output	dir				2016-07-18 02:50	rwxr-xr-x	cloudera	supergroup
least20	dir				2016-07-18 03:15	rwxr-xr-x	cloudera	supergroup
top20	dir				2016-07-18 03:04	rwxr-xr-x	cloudera	supergroup
travel.txt	file	185.69 MB	1	64 MB	2016-07-18 02:42	rw-r--r--	cloudera	supergroup

SNAPSHOT OF HADOOP FILE SYSTEM SHOWING traveldata DIRECTORY

3) LOADING THE FILE INTO PIG:

The below mentioned command will load the file into the pig thus enabling us to do further work on the data. The **load** statement will simply load the data into the specified relation in Apache Pig. In the given e.g. the relation is PigStorage().The **PigStorage()** function loads and stores data as structured text files. It takes a delimiter using which each entity of a tuple is separated as a parameter. By default, it takes **'\t'** as a parameter.

a=LOAD '/user/training/TravelData.txt' Using PigStorage('\t') as

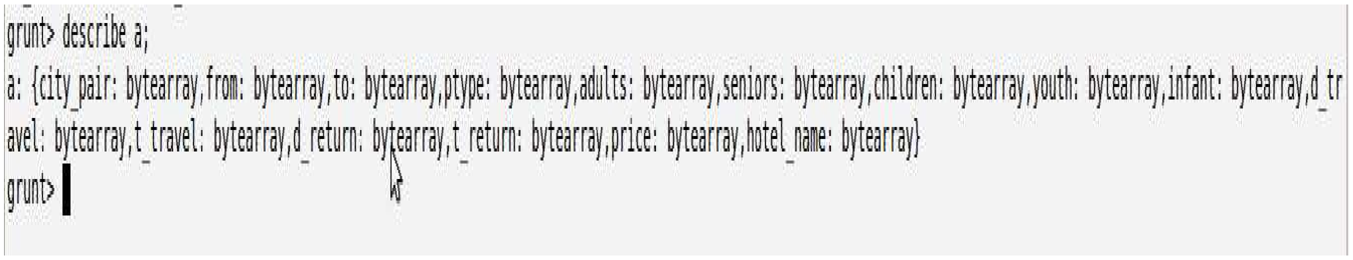
(city_pair,from,to,ptype,adults,seniors,children,youth,infant,d_travel,t_travel,d_return,t_return,price,hotel_name)

This line loads the information from HDFS into a **Pig collection named Data**. Pig expects data to be tab-delimited by default and as such our file is by default tab (space) delimited. Let's say our file was semi colon ";" delimited we could have told the system that semicolons are field separators by providing colon as the PigStorage() argument.

The **AS** clause defines how the fields in the file are mapped into Pig data types. As can be seen in the command a field is followed by its data type for example

City_pair : chararray,

(here the part left to the colon tells the name of the column and the right part tells the data type)



```
grunt> describe a;
a: {city_pair: bytearray,from: bytearray,to: bytearray,ptype: bytearray,adults: bytearray,seniors: bytearray,children: bytearray,youth: bytearray,infant: bytearray,d_tr
avel: bytearray,t_travel: bytearray,d_return: bytearray,t_return: bytearray,price: bytearray,hotel_name: bytearray}
grunt>
```

SCHEMATIC OF HOW DATA IS STORED IN PIG

NOTE:

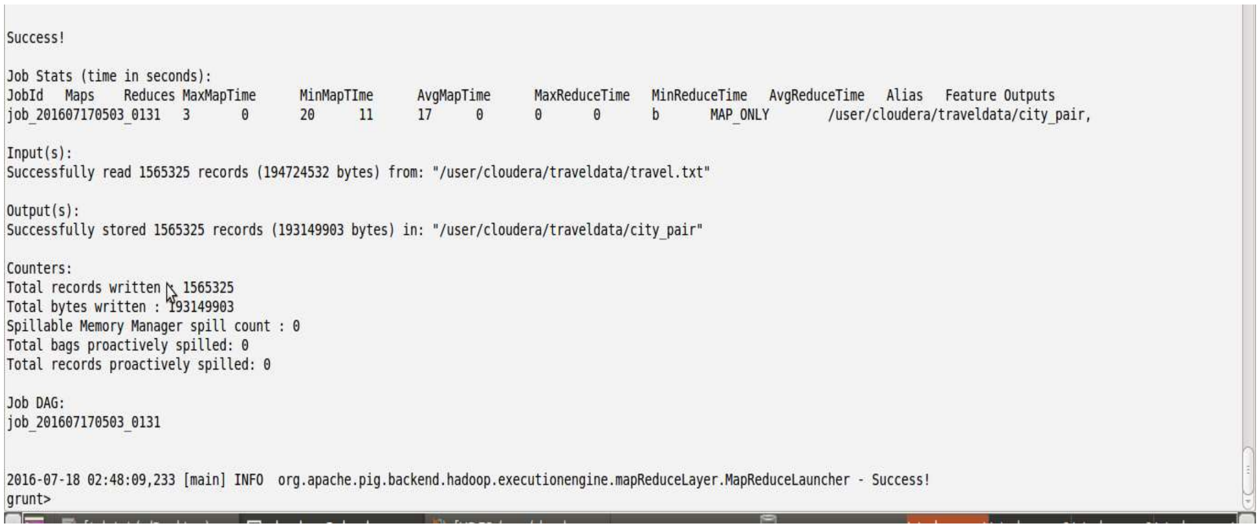
Pig Latin commands are terminated with semicolons. If you press Return on a line without terminating it, you'll get the >> characters, indicating that the command has been continued.

The **Load** and **Store** functions in Apache Pig are used to determine how the data goes ad comes out of Pig. These functions are used with the load and store operators. Given below is the list of load and store functions available in Pig.

S.N.	Function & Description
1	**PigStorage()** To load and store structured files.
2	**TextLoader()** To load unstructured data into Pig.
3	**BinStorage()** To load and store data into Pig using machine readable format.
4	**Handling Compression** We can load and store compressed data in Apache Pig using the functions BinStorage() and TextLoader()

b= load '/user/cloudera/traveldata/travel.txt' using PigStorage('\t') as (city_pair)

Above command when executed it stores whole row into a single column named city_pair



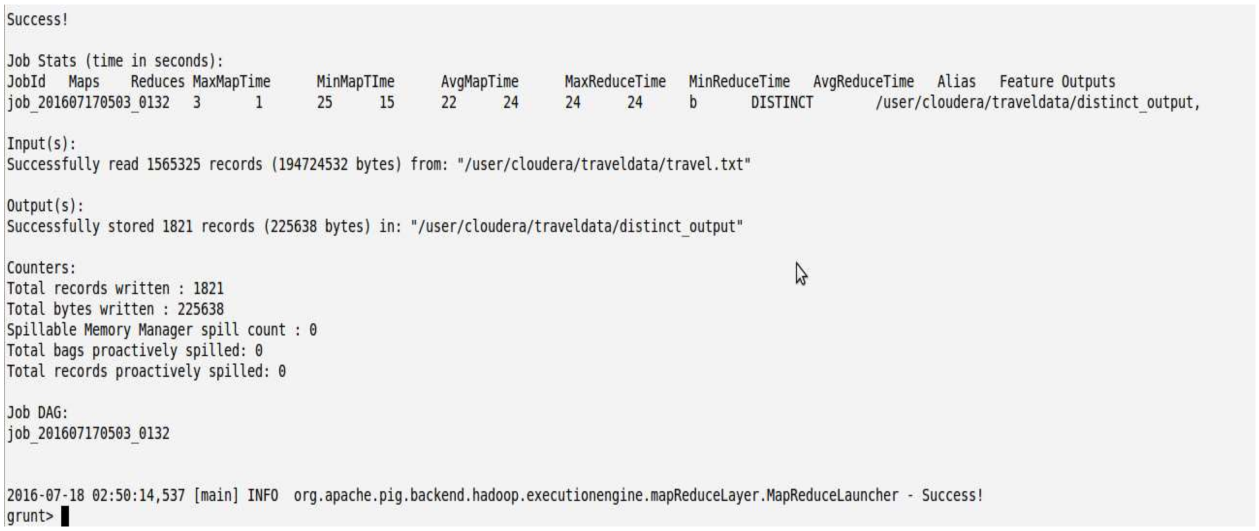
```
Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MaxReduceTime   MinReduceTime   AvgReduceTime   Alias   Feature Outputs
job_201607170503_0131   3       0       20      11      17      0       0       0       b       MAP_ONLY        /user/cloudera/traveldata/city_pair,

Input(s):
Successfully read 1565325 records (194724532 bytes) from: "/user/cloudera/traveldata/travel.txt"

Output(s):
Successfully stored 1565325 records (193149903 bytes) in: "/user/cloudera/traveldata/city_pair"

Counters:
Total records written : 1565325
Total bytes written : 193149903
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_201607170503_0131


2016-07-18 02:48:09,233 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

<u>SNAPSHOT SHOWING THE BACKEND WORKING OF MAP REDUCE WHILE LOADING DATA INTO PIG</u>

c=distinct b

This command will separate out all the distinct data from travel.txt file.

Store c into '/user/cloudera/traveldata/distinct_output' using PigStorage('\t')



```
Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MaxReduceTime   MinReduceTime   AvgReduceTime   Alias   Feature Outputs
job_201607170503_0132   3       1       25      15      22      24      24      24      b       DISTINCT        /user/cloudera/traveldata/distinct_output,

Input(s):
Successfully read 1565325 records (194724532 bytes) from: "/user/cloudera/traveldata/travel.txt"

Output(s):
Successfully stored 1821 records (225638 bytes) in: "/user/cloudera/traveldata/distinct_output"

Counters:
Total records written : 1821
Total bytes written : 225638
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_201607170503_0132


2016-07-18 02:50:14,537 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

d=group c by city_pair

Store d into '/user/cloudera/traveldata/distinct2_output' using PigStorage('\t');

e=foreach d generate flatten ($0), COUNT ($1) as count;

purpose of this command is:

for e.g.

d:{group: bytearray,c:{city_pair:bytearray}}

e:{group:bytearray,count:long}

store e into '/user/cloudera/traveldata/city_visitor-count' using PigStorage('\t');

FINDING TOP 20 DESTINATIONS

f=order e by count desc;

it will set all data from e to f in descending order

store f into '/user/cloudera/traveldata/city_visitor-count_desc' using PigStorage('\t')

ans =LIMIT f 20

store ans into '/user/cloudera/traveldata/top20' using PigStorage('\t')

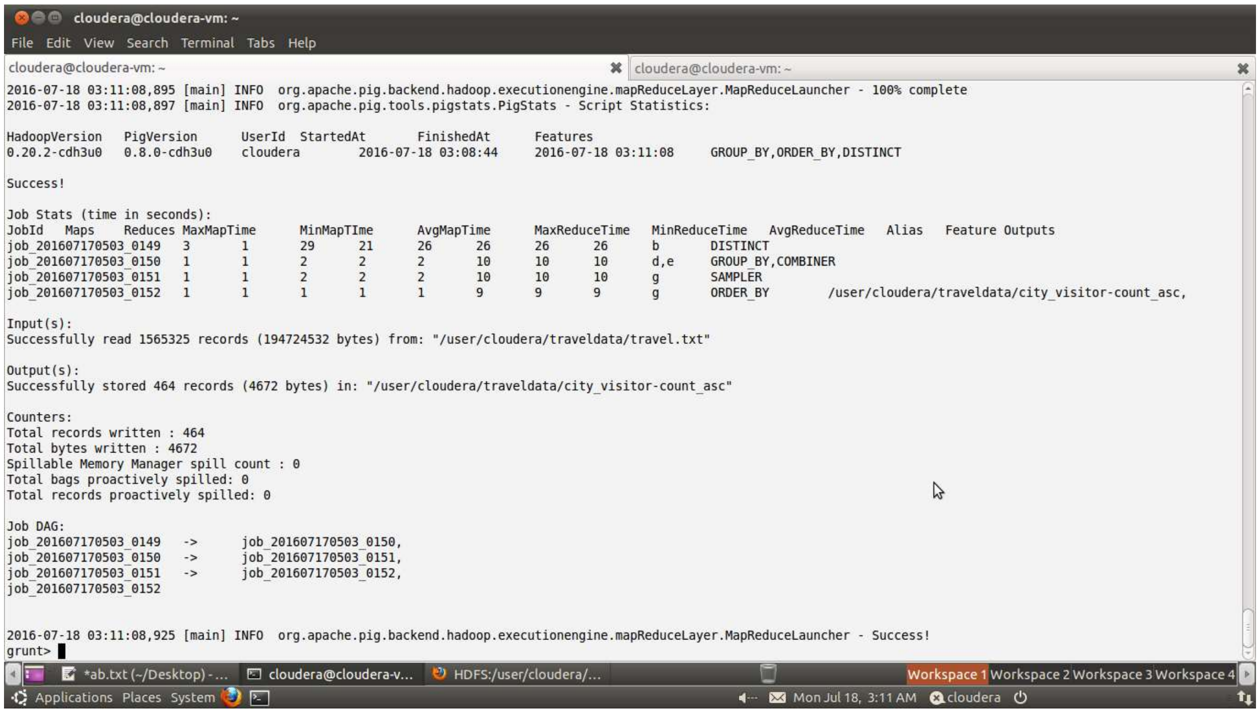
dump ans;

it will show contents of ans on terminal

g=order e by count asc;

it will set all data from e to g in descending order

store g into '/user/cloudera/traveldata/city_visitor-count_asc' using PigStorage('\t')



```
2016-07-18 03:11:08,895 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2016-07-18 03:11:08,897 [main] INFO  org.apache.pig.tools.pigstats.PigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
0.20.2-cdh3u0   0.8.0-cdh3u0    cloudera        2016-07-18 03:08:44     2016-07-18 03:11:08     GROUP_BY,ORDER_BY,DISTINCT

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MaxReduceTime   MinReduceTime   AvgReduceTime   Alias   Feature Outputs
job_201607170503_0149   3       1       29      21      26      26      26      26      b       DISTINCT
job_201607170503_0150   1       1       2       2       2       10      10      10      d,e     GROUP_BY,COMBINER
job_201607170503_0151   1       1       2       2       2       10      10      10      g       SAMPLER
job_201607170503_0152   1       1       1       1       1       9       9       9       g       ORDER_BY        /user/cloudera/traveldata/city_visitor-count_asc,

Input(s):
Successfully read 1565325 records (194724532 bytes) from: "/user/cloudera/traveldata/travel.txt"

Output(s):
Successfully stored 464 records (4672 bytes) in: "/user/cloudera/traveldata/city_visitor-count_asc"

Counters:
Total records written : 464
Total bytes written : 4672
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_201607170503_0149   ->      job_201607170503_0150,
job_201607170503_0150   ->      job_201607170503_0151,
job_201607170503_0151   ->      job_201607170503_0152,
job_201607170503_0152


2016-07-18 03:11:08,925 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt> 
```

ans2 =LIMIT f 20

store ans2 into '/user/cloudera/traveldata/least20' using PigStorage('\t')

dump ans2;

4) DATA PROCESSING

Now that the data has been loaded we can perform operations on the data to get the required data from the data i.e. the data is ready for analysis so that useful conclusions can be drawn out of the data.

Following are the various conclusions that were drawn from the data along with the way in which the steps were carried out.

A) CATEGORY COUNT: -

The following script is used to generate a count of all the various categories and the number of videos uploaded in each category over the time period of data in the file.

Note: “--“represents a comment;

Script: -

--extract the categories from the data

group1 = foreach data generate flatten(TOKENIZE(category)) as t:chararray

--group similar types of category tokens together into groups

g2 =GROUP group1 by t;

--count the number of entries in each group

cnt= FOREACH g2 generate group, COUNT(group1);

--now save the output file to the HDFS

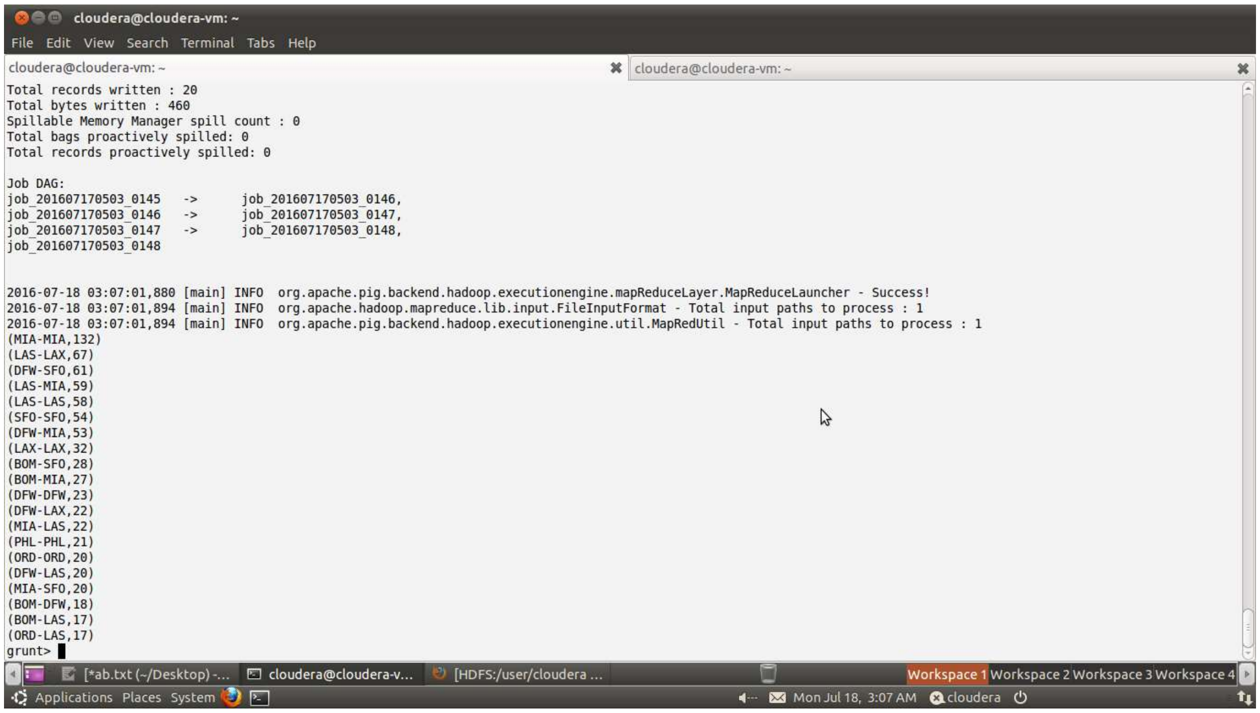
STORE cnt into 'hdfs://localhost:8020/user/cloudera/category_count';

Command 1:

The **TOKENIZE()** function of Pig is used to split a string (which contains a group of words) in a single tuple and returns a bag which contains the output of the split operation. In this case it will return bags containing the type of category.

The **FLATTEN** is an operator that changes the structure of tuples and bags. Flatten un-nests tuples as well as bags. The working of flatten is complex and varies from case to case but in this case it converts the bag into tuple. Thus **group1** has tuples of category type corresponding to each entry in the data relation.

SNAPSHOT OF SIMILAR TOKENS INTO GROUPS



```
Total records written : 20
Total bytes written : 460
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_201607170503_0145   ->      job_201607170503_0146,
job_201607170503_0146   ->      job_201607170503_0147,
job_201607170503_0147   ->      job_201607170503_0148,
job_201607170503_0148


2016-07-18 03:07:01,880 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2016-07-18 03:07:01,894 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2016-07-18 03:07:01,894 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(MIA-MIA,132)
(LAS-LAX,67)
(DFW-SFO,61)
(LAS-MIA,59)
(LAS-LAS,58)
(SFO-SFO,54)
(DFW-MIA,53)
(LAX-LAX,32)
(BOM-SFO,28)
(BOM-MIA,27)
(DFW-DFW,23)
(DFW-LAX,22)
(MIA-LAS,22)
(PHL-PHL,21)
(ORD-ORD,20)
(DFW-LAS,20)
(MIA-SFO,20)
(BOM-DFW,18)
(BOM-LAS,17)
(ORD-LAS,17)
grunt>
```

SNAPSHOT OF TUPLES CONTAINING city_pair AND count

Group command: The GROUP operator groups together tuples that have the same group key (key field). The key field will be a tuple if the group key has more than one field, otherwise it will be the same type as that of the group key. The result of a GROUP operation is a relation that includes one tuple per group.

In this case each tuple will have a collection of one kind of category. For example, one tuple will have all the entries of the category "Entertainment".

COUNT(): This function counts all values ignoring the null values. In the above case it basically counts the number of occurrences of each category in their respective tuple.

Group all: Use ALL if you want all tuples to go to a single group; for example, when doing aggregates across entire relations.

Foreach: The FOREACH operator is used to generate specified data transformations based on the column data.

Order: The ORDER BY operator is used to display the contents of a relation in a sorted order based on one or more fields.

Limit: The LIMIT operator is used to get a limited number of tuples from a relation. In this case we specify that we need only the top entry.

Similarly, we can run similar steps of commands multiple times to find the max and min of various fields like the most comments on a video. This will give us an insight to how many people watch a video and how many actually comment on a video which is the most watched category also these outputs help us to understand the output of other processes.

Group command: The GROUP operator groups together tuples that have the same group key (key field). The key field will be a tuple if the group key has more than one field, otherwise it will be the same type as that of the group key. The result of a GROUP operation is a relation that includes one tuple per group.

In this case we group the data according to the interval field in the data relation.

**Position notifier ($1): -**Fields are referred to by positional notation or by name (alias). Positional notation is generated by the system. Positional notation is indicated with the dollar sign ($) and begins with zero (0); for example, $0, $1, $2.

For example, in a relation we can refer to the first field by using the notation $0, the second field by $1 and so on.

Describe: The describe operator is used to view the schema of a relation.

5)STORING THE FILE FROM PIG TO HDFS:

Once the data has been processed it must be moved to the hadoop file system this is achieved by using the STORE() command. As seen above after each and every time after the data has been processed and a suitable result has been reached the output relation in stored. In each case the script ends with the STORE command/query given below is a brief summary of how the data store utility works.

Stores or saves results to the file system.

Syntax: -

STORE alias INTO 'directory' [USING function];

Terms

1. **Alias**: The name of a relation to be stored.
2. **INTO**: Required keyword.
3. **'directory'**: The name of the storage directory, in quotes. If the directory already exists, the STORE operation will fail.The output data files, named part-nnnnn, are written to this directory.

4. **USING:** Keyword. Use this clause to name the store function.If the USING clause is omitted, the default store function PigStorage is used.

5. **Function:** The store function. You can use a built in function.

   - **HBaseStorage:**Loads and stores data from an HBase table.
   - **PigStorage:** Loads and stores data as structured text files**.**
   - **PigDump:** Stores data in UTF-8 format.
   - **JsonLoader, JsonStorage:** Load or store JSON data.
   - **BinStorage:** Loads and stores data in machine-readable format.

PigStorage is the default store function and does not need to be specified (simply omit the USING clause). You can write your own store function if your data is in a format that cannot be processed by the built in functions (see User Defined Functions).

6. PERFORMING DATA ANALYSIS

A) Bar Graph showing top 20 travelled destinations

One of the main features of pig is that it has the capability to be integrated with many other tools and services

for example, through the first script we get the different categories of videos and the count of videos in watch category. This is in a simple table format and is txt file. this enables this file to be exported to many other tools for further analysis for example here we wanted to show this data in the form of a pie chart and a bar graph, so we imported the data into excel and for this we moved the file from the hadoop file system to Linux file system using

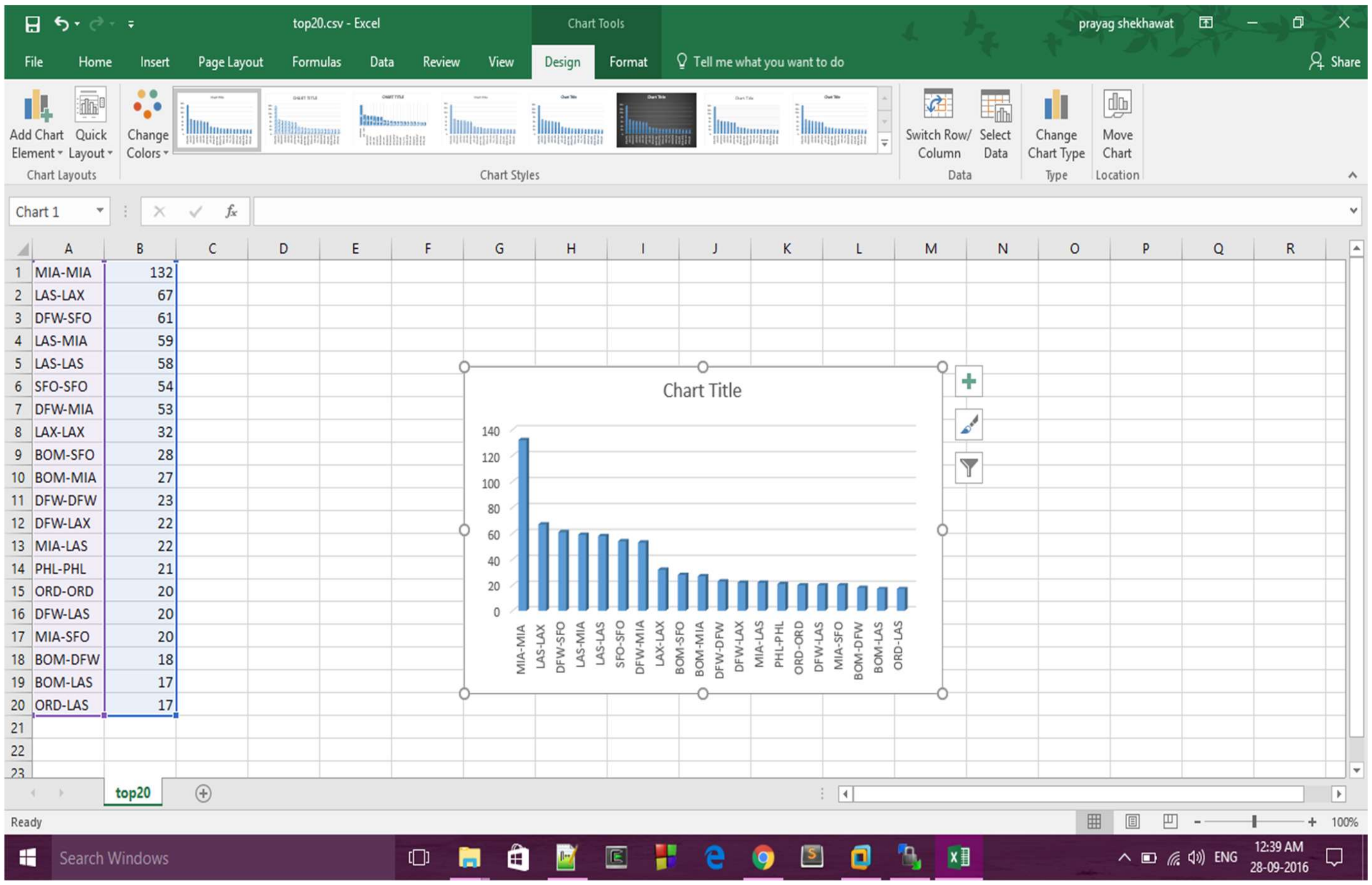
the following command.

hadoop fs -copyToLocal user/cloudera/top20.txt /home/Desktop

this was then exported to Microsoft excel.

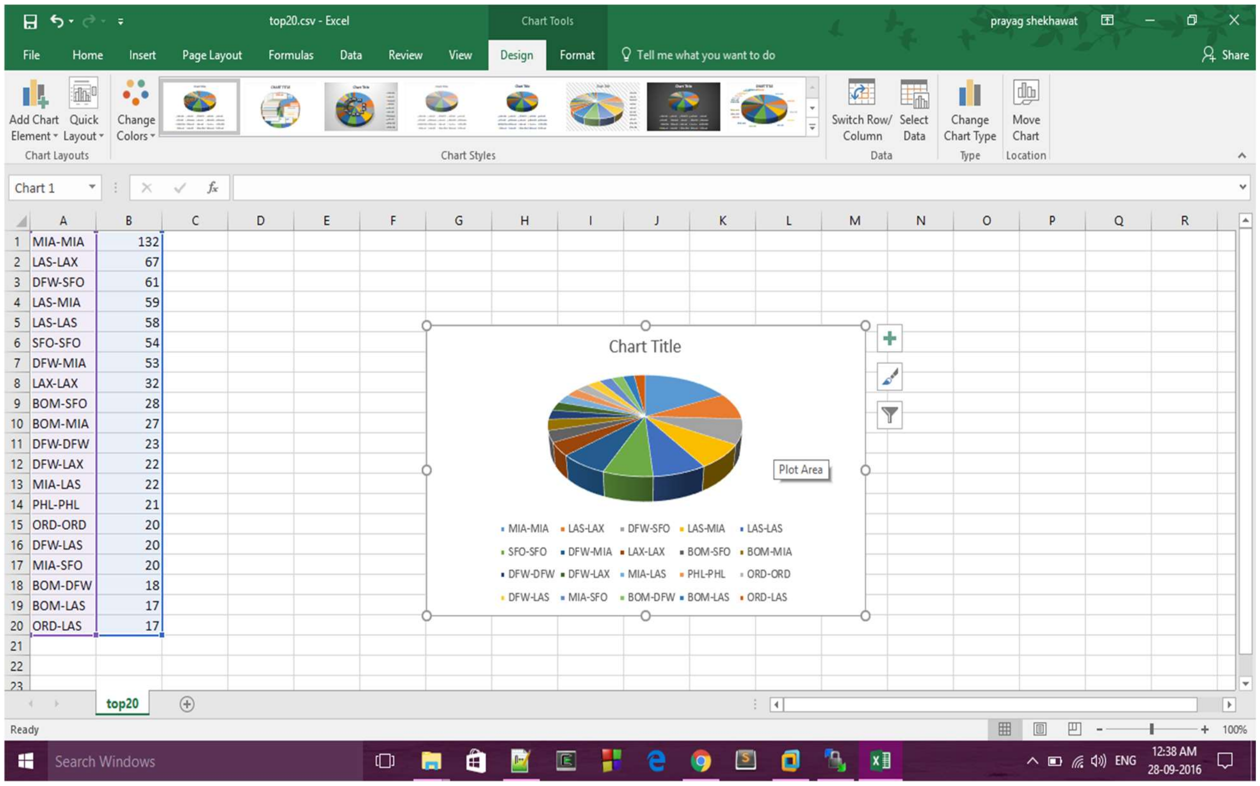
Analysis and conclusion drawn:

A) graph showing the top 20 destinations travelled.



	A	B
1	MIA-MIA	132
2	LAS-LAX	67
3	DFW-SFO	61
4	LAS-MIA	59
5	LAS-LAS	58
6	SFO-SFO	54
7	DFW-MIA	53
8	LAX-LAX	32
9	BOM-SFO	28
10	BOM-MIA	27
11	DFW-DFW	23
12	DFW-LAX	22
13	MIA-LAS	22
14	PHL-PHL	21
15	ORD-ORD	20
16	DFW-LAS	20
17	MIA-SFO	20
18	BOM-DFW	18
19	BOM-LAS	17
20	ORD-LAS	17

Pie chart for the above representation



	A	B
1	MIA-MIA	132
2	LAS-LAX	67
3	DFW-SFO	61
4	LAS-MIA	59
5	LAS-LAS	58
6	SFO-SFO	54
7	DFW-MIA	53
8	LAX-LAX	32
9	BOM-SFO	28
10	BOM-MIA	27
11	DFW-DFW	23
12	DFW-LAX	22
13	MIA-LAS	22
14	PHL-PHL	21
15	ORD-ORD	20
16	DFW-LAS	20
17	MIA-SFO	20
18	BOM-DFW	18
19	BOM-LAS	17
20	ORD-LAS	17

The above graph was also made using the same steps following the steps involved in making the previous graphs the file was imported to Linux from the hdfs using the CopytoLocal command and the file was then exported to Excel. Point to be noted is that we can use any other graph generating or any alternative analysis tool.

Analysis and conclusion from above graphs:

From above graph we can see that there are some destinations which are travelled most while other least , above graph can be useful to derive the conclusion that attention must be paid to the destinations which had least visitors and improve the number by increasing the offers while reducing the travel cost by tourism industries.

<u>ANALYSIS AND CONCLUSION: -</u>

By analyzing the data and analyzing the graph we analyze that there are destination which are frequently travelled and while others are very less travelled.

To increase the popularity of less visited place, travel and tourism industries should provide the travelling at lesser cost which some amazing offers like free travel for children etc.

Also this data can greatly help the industries to overcome their losses, and how they can increase their profit and the same time with the benefit of people coming to their travelling agencies

Bibligraphy

References

[1] Big Data Analysis: http://www.informationweek.com/big-data/big-data-analytics/big-data-analysis-drives-revolution-in-travel/d/d-id/1110636?

[2] Spark Use Case: https://acadgild.com/blog/spark-use-case-travel-data-analysis/

[3]https://www.igt.in/resources/download/44/Big-Data-The-360-degree-overview-for-the-Travel-industry

[4] Big Data and infinite responsibility: https://www.tnooz.com/article/big-data-and-the-infinite-possibilities-for-the-travel-industry/