INFO 7275 38035
Advanced Database Management System
Section 01

A Final Project Report
on
Analysis of ‘Amazon Fine Food Review’ Dataset

Tanisha Jain
Northeastern University
MS-IS
(NU Id: 001617268)

Summary:

This project is based on the Analysis of the '**Amazon Fine Food Review**' dataset. The raw dataset consists of reviews of fine foods from Amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product id and user information, ratings, and a plaintext review.

In the project, first the analysis of the raw dataset using Tableau is done. Then the cleaning of the data files is done using RScript on RStudio. The MapReduce analysis is performed using several MapReduce patterns such as Partitioning, Distinct etc. Also, in this project, Apache Hive and Apache Mahout is implemented. Amazon Elastic Map Reduce is also used to implement MapReduce algorithms on cloud provided by Amazon Web Services.

The analysis on the dataset can help Amazon products with its sales, scope of improvement and honest user reviews for many other users buying similar products.

Dataset Link: https://www.kaggle.com/snap/amazon-fine-food-reviews (originally this dataset was published on http://snap.stanford.edu/data/web-FineFoods.html)

Attributes of dataset:

Number of reviews	**568,454**
Number of users	**256,059**
Number of products	**74,258**
Users with > 50 reviews	**260**
Median no. of words per review	**56**
Timespan	**Oct 1999 - Oct 2012**

Amazon Fine Food Review Dataset:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Id	ProductId	UserId	Helpfulr	Helpfuln	Score	Time	Summary	Review											
2	1	B001E4KFG0	A3SGXH7AUHU8GW	1	1	5	1303862400	Good Quality	I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed r											
3	2	B00813GRG4	A1D87F6ZCVE5NK	0	0	1	1346976000	Not as Adver	Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to represen											
4	3	B000LQOCH0	ABXLMWJIXXAIN	1	1	4	1219017600	Delight says	This is a conf	pillowy citru	and very fla	The Witch	and The Wardrobe - this is the treat that seduces Edmund into selling out his Brother and Sisters to the Witch.							
5	4	B000UA0QIQ	A395BORC6FGVXV	3	3	2	1307923200	Cough Medic	If you are looking for the secret ingredient in Robitussin I believe I have found it. I got this in addition to the Root Beer Extract I ordered (which was good) and made											
6	5	B006K2ZZ7K	A1UQRSCLF8GW1T	0	0	5	1350777600	Great taffy	Great taffy a	this is a deal.										
7	6	B006K2ZZ7K	ADT0SRK1MGOEU	0	0	4	1342051200	Nice Taffy	I got a wild h	root beer	melon	peppermint	grape	etc. My only	my kids	and my hust	this lasted only two weeks! I would recommend this brar			
8	7	B006K2ZZ7K	A1SP2KVKFXXRU1	0	0	5	1340150400	Great! Just a	This saltwate	which did ha	Fralinger's.	Would highly recommend this candy!	I served it at a beach-themed party and everyone loved it!							
9	8	B006K2ZZ7K	A3JRGQVEQN31IQ	0	0	5	1336003200	Wonderful	tasty taffy	This taffy is so good. It is very soft and chewy. The flavors are amazing. I would definitely recommend you buying it. Very satisfying!!										
10	9	B000E7L2R4	A1MZYO9TZK0BBI	1	1	5	1322006400	Yay Barley	Right now I'm mostly just sprouting this so my cats can eat the grass. They love it. I rotate it around with Wheatgrass and Rye too											
11	10	B00171APVA	A21BT40VZCCYT4	0	0	5	1351209600	Healthy Dog	This is a very	healthy dog food. Good for their digestion. Also good for small puppies. My dog eats her required amount at every feeding.										
12	11	B0001PB9FE	A3HDKO7OW0QNK4	1	1	5	1107820800	The Best Hot	I don't know	but the flavc	because of t	we have a ca	but don't wa	grab a bottle	you will nev	incredible service!				
13	12	B0009XLVG0	A2725IB4YY9JEB	4	4	5	1282867200	My cats LOVI	One of my be	and the prot	no by-product food up higher where only my skinny boy can jump. The higher food sits going stale. They both really go for this food. Ar										
14	13	B0009XLVG0	A327PCT23YH90	1	1	1	1339545600	My Cats Are	My cats have	I now need to find a new food that my cats will eat.										
15	14	B001GVISJM	A18ECVX2RJ7HUE	2	2	4	1288915200	fresh and gre	good flavor!	these came securely packed... they were fresh and delicious! i love these Twizzlers!										
16	15	B001GVISJM	A2MUGFV2TDQ47K	4	5	5	1268352000	Strawberry T	The Strawberry Twizzlers are my guilty pleasure - yummy. Six pounds will be around for a while with my son and I.											
17	16	B001GVISJM	A1CZX3CP8IKQIJ	4	5	5	1262044800	Lots of twizzl	just what yo	My daughter loves twizzlers and this shipment of six pounds really hit the spot. It's exactly what you would expect...six packages of strawberry twizzler										
18	17	B001GVISJM	A3KLWF6WQ5BNYO	0	0	2	1348099200	poor taste	I love eating them and they are good for watching TV and looking at movies! It is not too sweet. I like to transfer them to a zip lock baggie so they stay fresh so I can											
19	18	B001GVISJM	AFKW14U97Z6QO	0	0	5	1345075200	Love it!	I am very satisfied with my Twizzler purchase. I shared these with others and we have all enjoyed them. I will definitely be ordering more.											
20	19	B001GVISJM	A2A9X58G2GTBLP	0	0	5	1324598400	GREAT SWEE	Twizzlers	Strawberry t	made in Lan	Inc. one of t	now a Subsi	the Compan	they also ma	Green Color	I like them a	the longest I	Inc. This Rec	1998. This Produ
21	20	B001GVISJM	A3IV7CL2C13K2U	0	0	5	1318032000	Home delive	Candy was delivered very fast and was purchased at a reasonable price. I was home bound and unable to get to a store so this was perfect for me.											
22	21	B001GVISJM	A1WO0KGLPR5PV6	0	0	5	1313452800	Always fresh	My husband	packed well and arrive in a timely manner.										
23	22	B001GVISJM	AZOF9E17RGZH8	0	0	5	1308960000	TWIZZLERS	I bought thes	and apparer	and this was	Strawberry	16-Ounce Bags (Pack of 6)							
24	23	B001GVISJM	ARYVQL4N737A1	0	0	5	1304899200	Delicious pro	I can remember buying this candy as a kid and the quality hasn't dropped in all these years. Still a superb product you won't be disappointed with.											
25	24	B001GVISJM	AJ613OLZZUG7V	0	0	5	1304467200	Twizzlers	I love this candy. After weight watchers I had to cut back but still have a craving for it.											
26	25	B001GVISJM	A22P2J09NJ9HKE	0	0	5	1295481600	Please sell th	I have lived c	and I so mis	I always stoc	more often than I'm able to buy them right now.								
27	26	B001GVISJM	A3FONPR03H3PJS	0	0	5	1288310400	Twizzlers - St	Product rece	Strawberry	16-Ounce Bags (Pack of 6)										
28	27	B001GVISJM	A3RXAU2N8KV45G	0	1	1	1332633600	Nasty No fla	The candy is	No flavor . Just	plan and chewy . I would never buy them again										
29	28	B001GVISJM	AAAS38B98HMIK	0	1	4	1331856000	Great Bargai	I was so glad	Amazon carried these batteries. I have a hard time finding them elsewhere because they are such a unique size. I need them for my garage door open										
30	29	B00144C10S	A2F4LZVGFLD1OB	0	0	5	1338854400	YUMMY!	I got this for	and my fath	you would never guess that they're sugar-free and it's so great that you can eat them pretty much guilt free!	i was so impressed that i've								
31	30	B0001PB9FY	A3HDKO7OW0QNK4	1	1	5	1107820800	The Best Hot	I don't know	but the flavc	because of t	we have a ca	but don't wa	grab a bottle	you will nev	incredible service!				
32	31	B003F6UO7K	AFM0O9480F04W	0	0	5	1297641600	Great machir	I have never	my mother	too (as a usu | ally non-coffee drinker!). The little Dolche Guesto Machine is super easy to use and prepares a really good Coffee/Latte										
33	32	B003F6UO7K	A31OQO709M20Y7	0	1	5	1288310400	THIS IS MY T	This offer is a	thanks Amazon for selling this product. Staral										
34	33	B001EO5QW8	AOVROBZ8BNTP7	19	19	4	1163376000	Best of the Ir	McCann's Ins	however	that even th	the McCann	all-natural b	though	is McCann's	tasty oatme	after sitting	the instant McCann's becomes too thick an		
35	34	B001EO5QW8	A3PMM0NFVEJGK9	13	13	4	1166313600	Good Instant	This is a good	so not only (	but some doctors now say that this form of sugar is better for you. Great on a cold morning when you don't have time to make McCann'										

Amazon User Profiles:

	A	B	C	D	E	F	G
1	UserId	FirstName	LastName	Age	Street	City	State
2	A3SGXH7AUHU8GW	Isabelle	Colon	38	Emazur Way	Hupobad	OK
3	A1D87F6ZCVE5NK	Katie	Hamilton	36	Pungut Plaza	Hizigu	NY
4	ABXLMWJIXXAIN	Bill	Jackson	62	Dutaf Highwa	Humcobpof	MI
5	A395BORC6FGVXV	Russell	Phelps	33	Pezok Pass	Silikomi	SC
6	A1UQRSCLF8GW1T	Sam	Rodriquez	30	Ezoid Boulev	Koofvo	NM
7	ADT0SRK1MGOEU	Duane	Ferguson	39	Vifimo Place	Jemoke	NJ
8	A1SP2KVKFXXRU1	Trevor	McCarthy	60	Usoc Parkwa	Zesevroc	IA
9	A3JRGQVEQN31IQ	Hilda	Hardy	40	Kearu Boulev	Novotukor	MT
10	A1MZYO9TZK0BBI	Jeff	Diaz	27	Duli Square	Mufojbu	MA
11	A21BT40VZCCYT4	Zachary	Johnston	46	Pada Loop	Esfokaj	AR
12	A3HDKO7OW0QNK4	Juan	Boone	31	Vigo Circle	Zijnagken	IA
13	A2725IB4YY9JEB	Terry	Daniels	18	Nese Terrace	Sehelri	WI
14	A327PCT23YH90	Josephine	Estrada	30	Dotza Key	Zuhatuca	OK
15	A18ECVX2RJ7HUE	Norman	Lamb	41	Solma Glen	Cuhdakbe	AR
16	A2MUGFV2TDQ47K	Benjamin	Harvey	64	Rophu Drive	Nejugan	UT
17	A1CZX3CP8IKQIJ	Leo	Daniels	52	Hake Ridge	Ugelehwus	UT
18	A3KLWF6WQ5BNYO	Louise	Clark	35	Ebuuj Boulev	Egizevzo	NV
19	AFKW14U97Z6QO	Rosie	Roy	35	Fusef Street	Abomaoz	TX
20	A2A9X58G2GTBLP	Luella	Kim	38	Figzo Avenue	Ursadpas	WY
21	A3IV7CL2C13K2U	Phoebe	Williamson	44	Muiz Circle	Kafjuwu	DE
22	A1WO0KGLPR5PV6	Mitchell	Silva	54	Zapok View	Riavamas	FL
23	AZOF9E17RGZH8	Polly	Watkins	48	Porap Junctio	Tuwojug	KS
24	ARYVQL4N737A1	Mae	Mendez	37	Arepu Loop	Zapodiso	FL
25	AJ613OLZZUG7V	Zachary	Howell	23	Rozi Way	Ebejipvi	UT
26	A22P2J09NJ9HKE	Bill	Butler	46	Sefin Way	Lohtaelo	SC
27	A3FONPR03H3PJS	Mayme	Carpenter	49	Erugu Extens	Semojid	ID
28	A3RXAU2N8KV45G	Logan	Long	25	Vurbel View	Vavenza	MI
29	AAAS38B98HMIK	Herbert	Maxwell	50	Kinju Boulev	Ezewiidi	MT
30	A2F4LZVGFLD1OB	Isaac	Harrington	55	Gefibi Highw	Faukte	NC
31	A3HDKO7OW0QNK4	Dennis	McGee	22	Fici Street	Mimhibam	NM

Data Cleaning using R script in RStudio:

1) The raw data file with columns - Id, ProductId, UserId, ProfileName, HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary, Text and rows-568454. File is first read into RStudio and stored in a variable.
2) The time column which showed a Unix time field is replaced with its corresponding date value. (Formula Used to calculate time from unix to date format =(A1/86400)+25569+(-5/24))
3) The data is binded in a dataframe and written to the csv file.
4) Another file with users' profile is used, binded into a dataframe and is written to csv.

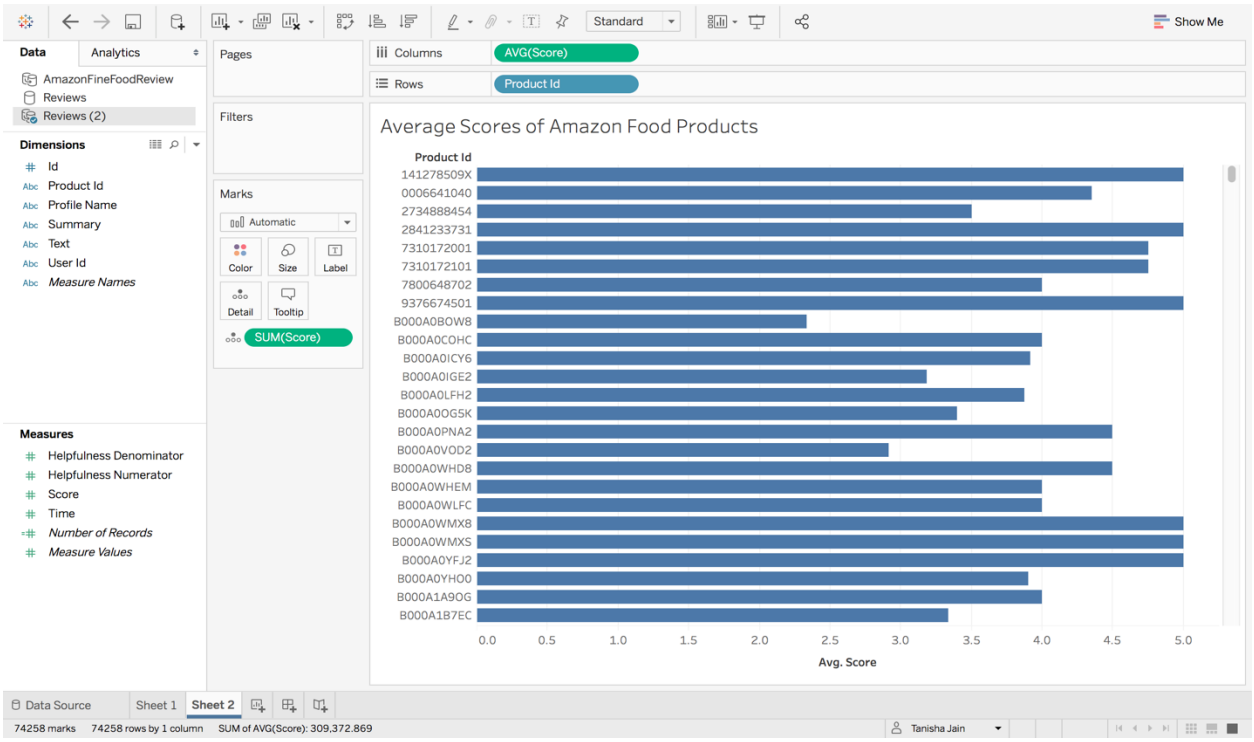
Find the RScript below-

```
#New R Script for cleaning the dataset and storing as a cleansed csv data file (Reviews.csv)
File1<-read.csv(file.choose(),header=TRUE)
View(File1)

#Data with Date to be appended
File2<-read.table(file.choose(),header=TRUE)
View(File2)

#Binding date with File1 and dropping columns 'Profile Name' and 'Time'
DataBind1<-cbind.data.frame(Id = File1$Id, ProductId = File1$ProductId,
                    UserId = File1$UserId, Date = File2$Date,
                    HelpfulnessNumerator = File1$HelpfulnessNumerator,
                    HelpfulnessDenominator = File1$HelpfulnessDenominator,
                    Rating = File1$Score, Summary = File1$Summary, Text = File1$Text)
View(DataBind1)

write.table(DataBind1, file = "AmazonFineFoodReview500k.csv", sep = ",", col.names = NA,
          qmethod = "double")
AmazonFFR<-read.table("AmazonFineFoodReview500k.csv", header = TRUE, sep = ",",
row.names = 1)
View(AmazonFFR)

#Trim the dataset
DataBind1<-head(AmazonFFR, 90000)
DataTable1<-data.frame(DataBind1)
View(DataTable1)

#Writing the first 90000 rows in a new file
write.table(DataTable1, file = "FineFoodReview90k.csv", sep = ",", col.names = NA,
          qmethod = "double")

#Editing new generated file for Users Profile
File3<-read.csv(file.choose(), header=TRUE)
View(File3)
```

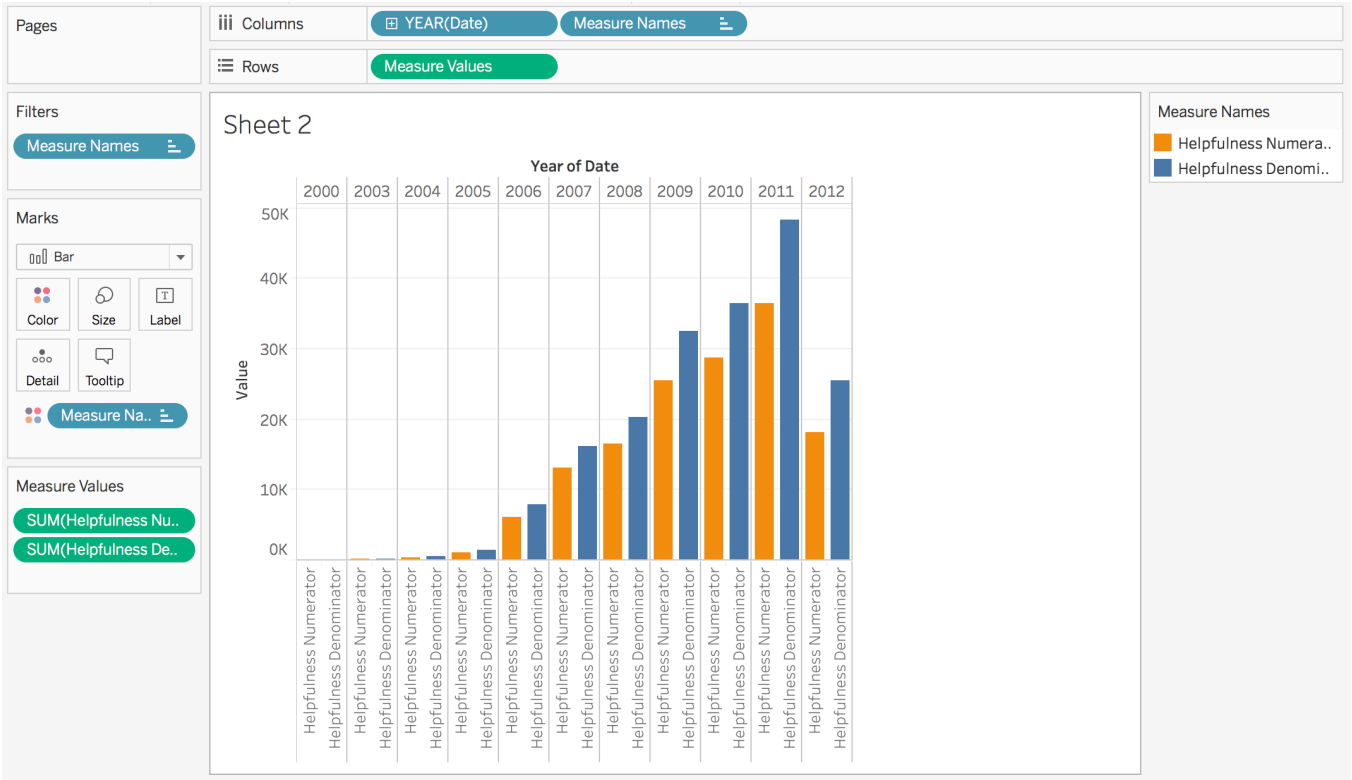
```
file3Modified<-cbind.data.frame(Seq=File3$Seq, UserId=DataTable1$UserId, First=File3$First,
Last=File3$Last, Age=File3$Age, Street=File3$Street, City=File3$City, State=File3$State)

View(file3Modified)

#Writing the user modified file
write.csv(file3Modified, "UserProfiles90k.csv", row.names = FALSE, quote = FALSE)
```

Data Analysis Using Tableau:

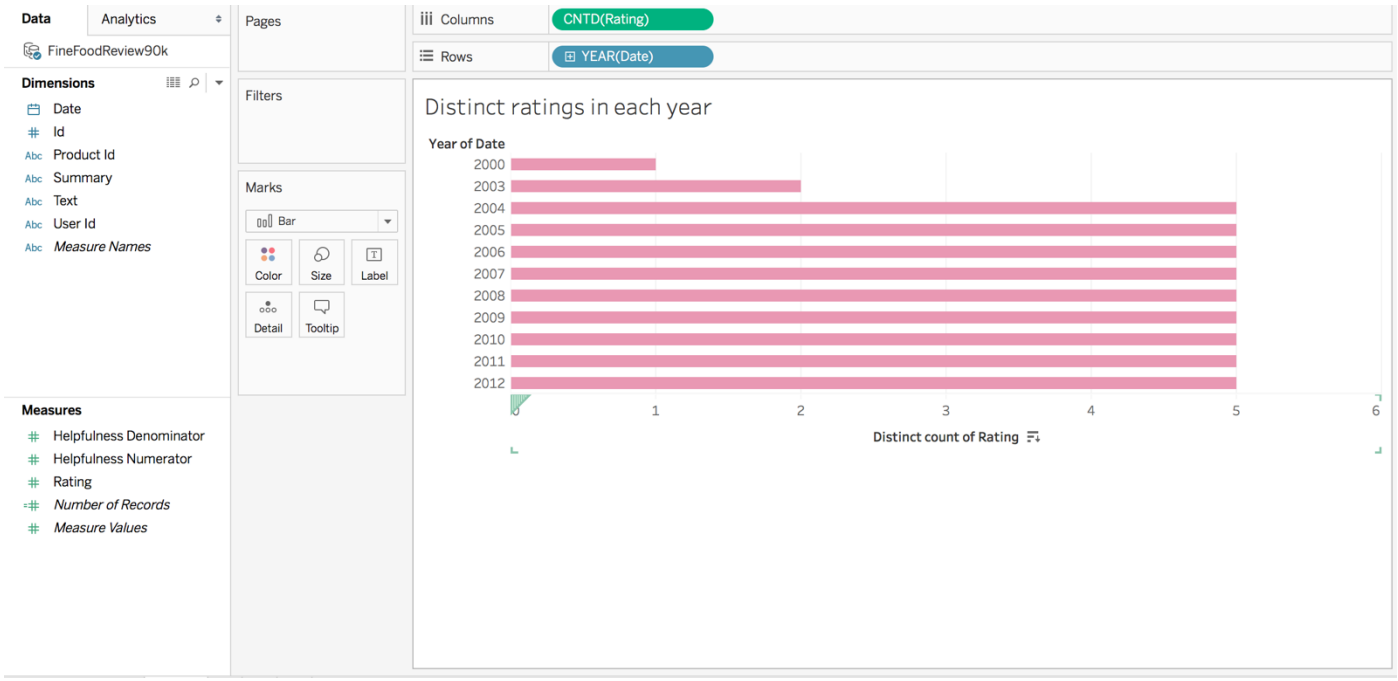
1) This is the average rating scores for the Amazon Food Products. By this analysis we can see which product has what average rating.



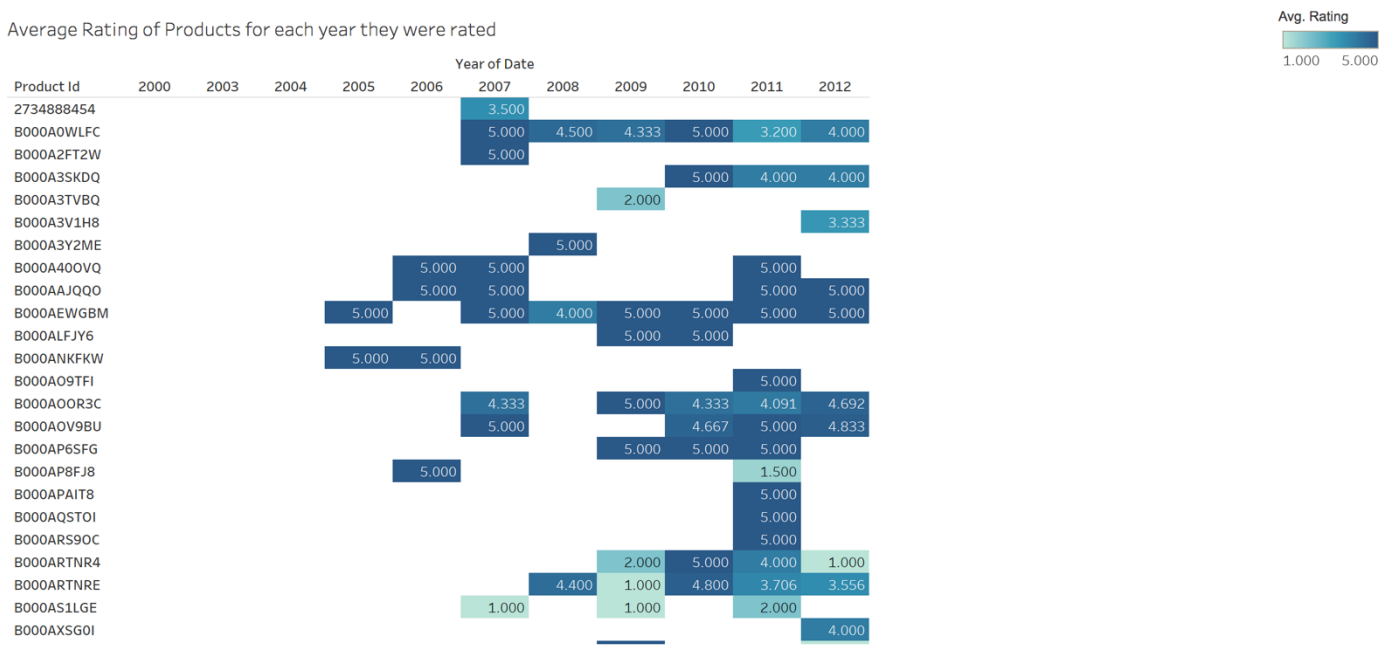
2) The columns in the dataset – Helpfulness Numerator and the Helpfulness Denominator indicates how many people liked the review out of what respectively. So here, the analysis is the overall helpfulness numerator and denominator value in a particular year.



3) The analysis graph below shows how many distinct ratings users gave for the products of that year. For example - Products received only 2 different ratings (can be either 2 from) in the year 2000 and all 5 different ratings in the year 2004 through 2012.



4) This graph below shows the average rating of a particular product in the particular year. The color strength shows the score of the rating 1 being low score and 5 being the best.

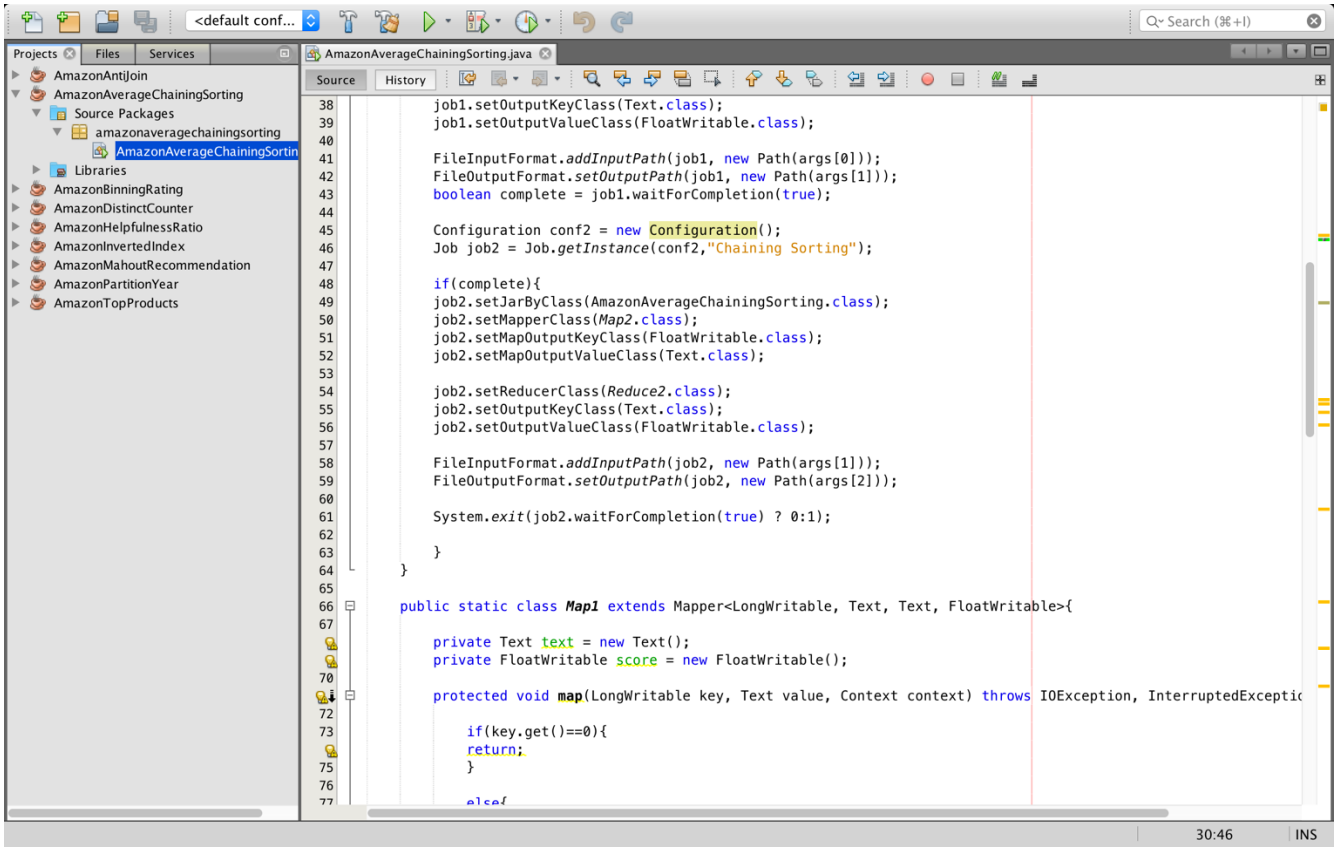


All the above analysis are created in Tableau and are also exported to Tableau Public. (URL: https://public.tableau.com/profile/tanishajain)

MapReduce Analysis on 'Amazon Fine Food Review' Dataset:

1) Amazon Fine Food Review Data: Average-Chaining-Sorting: (Numerical Summarization Pattern & Job Chaining)

This MR job uses the chaining where the output of one MapReduce job goes to the other before finally outputting it to the user. There are 2 mappers and 2 reducers.



The first MR output averages the ratings of a particular product with productId as key and average rating as value.

part-r-00000(Intermediate-Average)

```
2734888454      3.5
B00002N8SM      1.7894737
B00002NCJC      4.5
B00002Z754      5.0
B00005V3DC      5.0
B00006IDJ0      1.0
B00006IDJU      4.0
B000084DVR      5.0
B000084E1U      5.0
B000084E6V      4.1073446
B000084ED0      5.0
B000084EK4      3.0
B000084EK5      5.0
B000084EK6      5.0
B000084EK7      4.0
B000084EK8      3.0
B000084EK9      3.6666667
B000084EKA      4.0
B000084EKB      4.0
B000084EKC      5.0
B000084EKD      2.0
B000084EKG      5.0
B000084EKL      5.0
B000084EKM      4.0
B000084EKO      3.0
B000084EKQ      3.0
B000084EKW      4.3333335
B000084EKY      4.8
B000084ESM      4.3333335
B000084ETV      3.6923077
```

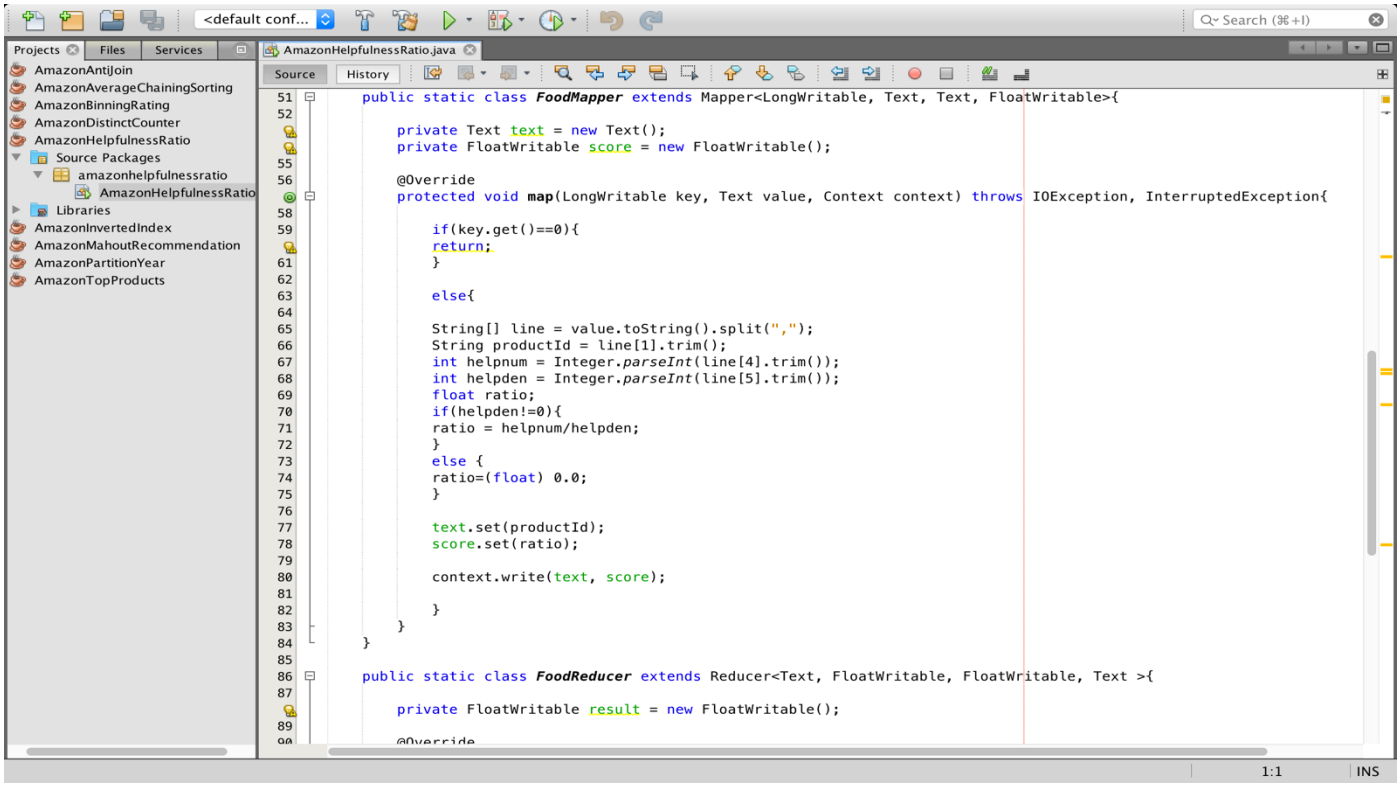
The second MR output shows the averages of the products ascendingly.

part-r-00000 (Final-sortedAverageAscending)

```
B001EQ4VXG	4.909091
B000COMPIY	4.9130435
B000F0FZCI	4.9166665
B000Q0IMOK	4.9166665
B000F4DKB2	4.9166665
B0050HZLN4	4.9166665
B001QEF4XC	4.9166665
B003KLSZGW	4.923077
B001964970	4.923077
B0002YGSA0	4.923077
B001EQ582E	4.9259257
B001EQ4G6I	4.928571
B0002BKIRW	4.928571
B002AUBJFI	4.9333334
B000YV9QMI	4.9333334
B000E682LY	4.9333334
B0030MNHEG	4.9375
B003DNL9U6	4.9411764
B008NDSNAU	4.9411764
B0056AL6G6	4.9473686
B002LMD8P4	5.0
B002LG5TWK	5.0
B002LMJ44S	5.0
B002LML5IQ	5.0
B002LDA2TS	5.0
B002LMNYG2	5.0
B002LN282M	5.0
B002LN3AOM	5.0
B002LN3DKI	5.0
B002LN4M8A	5.0
B002LN4YOC	5.0
B002L6KDQM	5.0
B002L68KDK	5.0
B002L3GKX0	5.0
B002LNZ7J8	5.0
B002KXDK48	5.0
B002LO5SFA	5.0
B002LQAZGA	5.0
B002KO8114	5.0
B002KGN4LE	5.0
B002KGEO8G	5.0
B002KE92UI	5.0
```

2) Amazon Fine Food Review Data: Helpfulness Numerator/Helpfulness Denominator (Numerical Summarization Pattern)

In this analysis, I have taken ratio of Helpfulness numerator and Helpfulness Denominator for each product and averaged the value. The ratio is then multiplied with 100 and the output only shows upto 2 decimal places for the float values.



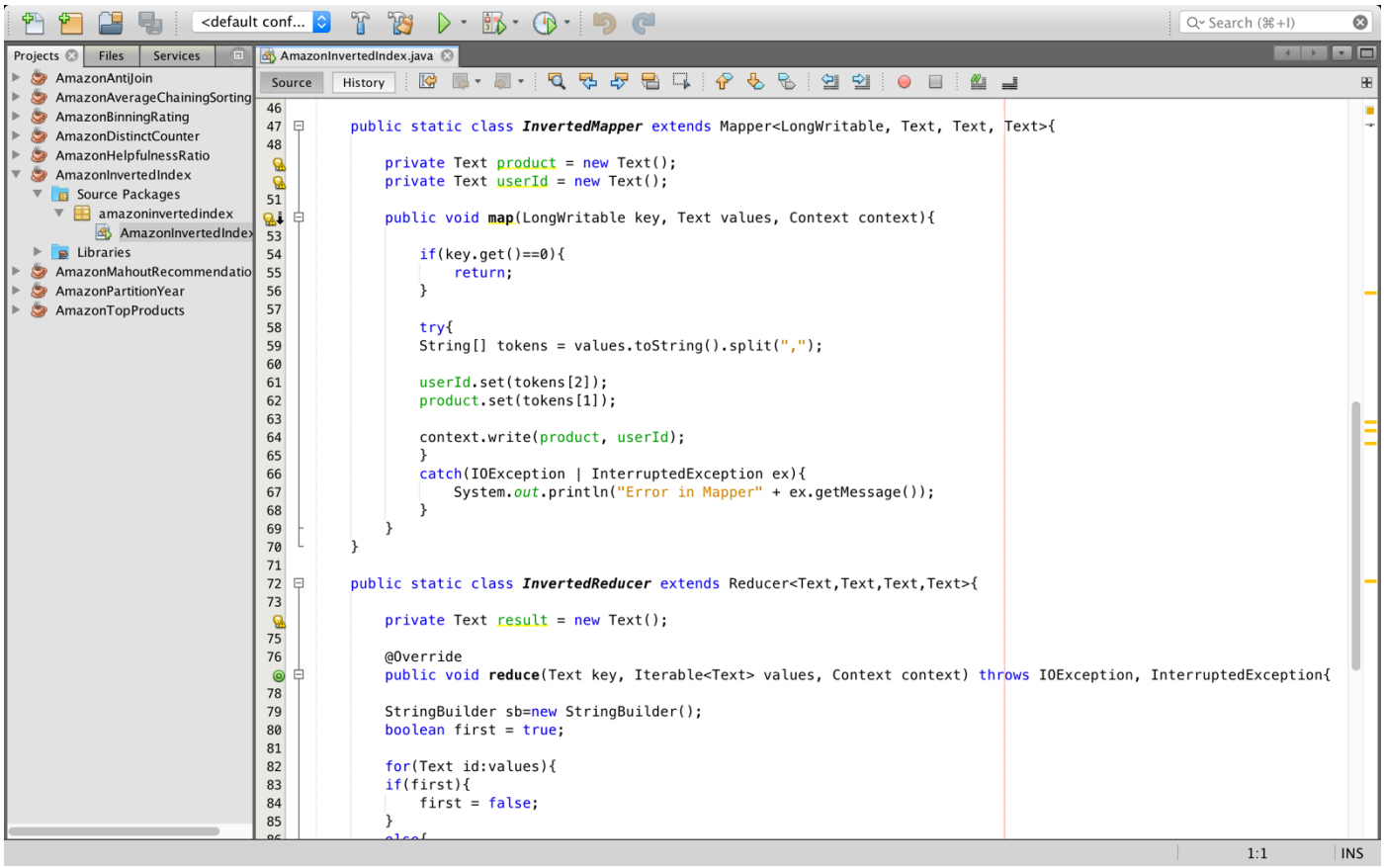
The output file is shown below:

part-r-00000(Percentage)

```
100.0    %        B00008DFR5
32.65    %        B00008JOL0
12.5     %        B00008MOJ2
0.0      %        B00008O36H
0.0      %        B00008RCMI
0.0      %        B000093HOV
0.0      %        B00009608Y
0.0      %        B00009601V
33.33    %        B00009OLE2
0.0      %        B00009OLEW
0.0      %        B00009OLFC
100.0    %        B0000A0BS8
0.0      %        B0000AH3OX
100.0    %        B0000AH3QT
28.57    %        B0000AH3QW
50.0     %        B0000AH3RR
66.67    %        B0000CDBRH
71.43    %        B0000CDBRJ
22.22    %        B0000CDBRP
28.57    %        B0000CDBRV
62.5     %        B0000CDBRY
62.5     %        B0000CDBRZ
100.0    %        B0000CE29E
100.0    %        B0000CEOKH
73.91    %        B0000CEPDP
33.33    %        B0000CERHH
100.0    %        B0000CERPU
0.0      %        B0000CERXB
25.0     %        B0000CERXU
20.0     %        B0000CERZK
0.0      %        B0000CGE3G
66.67    %        B0000CGE3R
0.0      %        B0000CGFCD
100.0    %        B0000CGFSC
0.0      %        B0000CGFV4
0.0      %        B0000CNU18
0.0      %        B0000CNU1A
```

3) Amazon Fine Food Review Data: Inverted Index (Numerical Summarization Pattern)

In this analysis, Inverted Indexing is used. It is very helpful as the output can show what Product are reviewed by which users. The key value will have multiple values.



```
    public static class InvertedMapper extends Mapper<LongWritable, Text, Text, Text>{

        private Text product = new Text();
        private Text userId = new Text();

        public void map(LongWritable key, Text values, Context context){

            if(key.get()==0){
                return;
            }

            try{
            String[] tokens = values.toString().split(",");

            userId.set(tokens[2]);
            product.set(tokens[1]);

            context.write(product, userId);
            }
            catch(IOException | InterruptedException ex){
                System.out.println("Error in Mapper" + ex.getMessage());
            }
        }
    }

    public static class InvertedReducer extends Reducer<Text,Text,Text,Text>{

        private Text result = new Text();

        @Override
        public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException{

        StringBuilder sb=new StringBuilder();
        boolean first = true;

        for(Text id:values){
        if(first){
            first = false;
        }
```

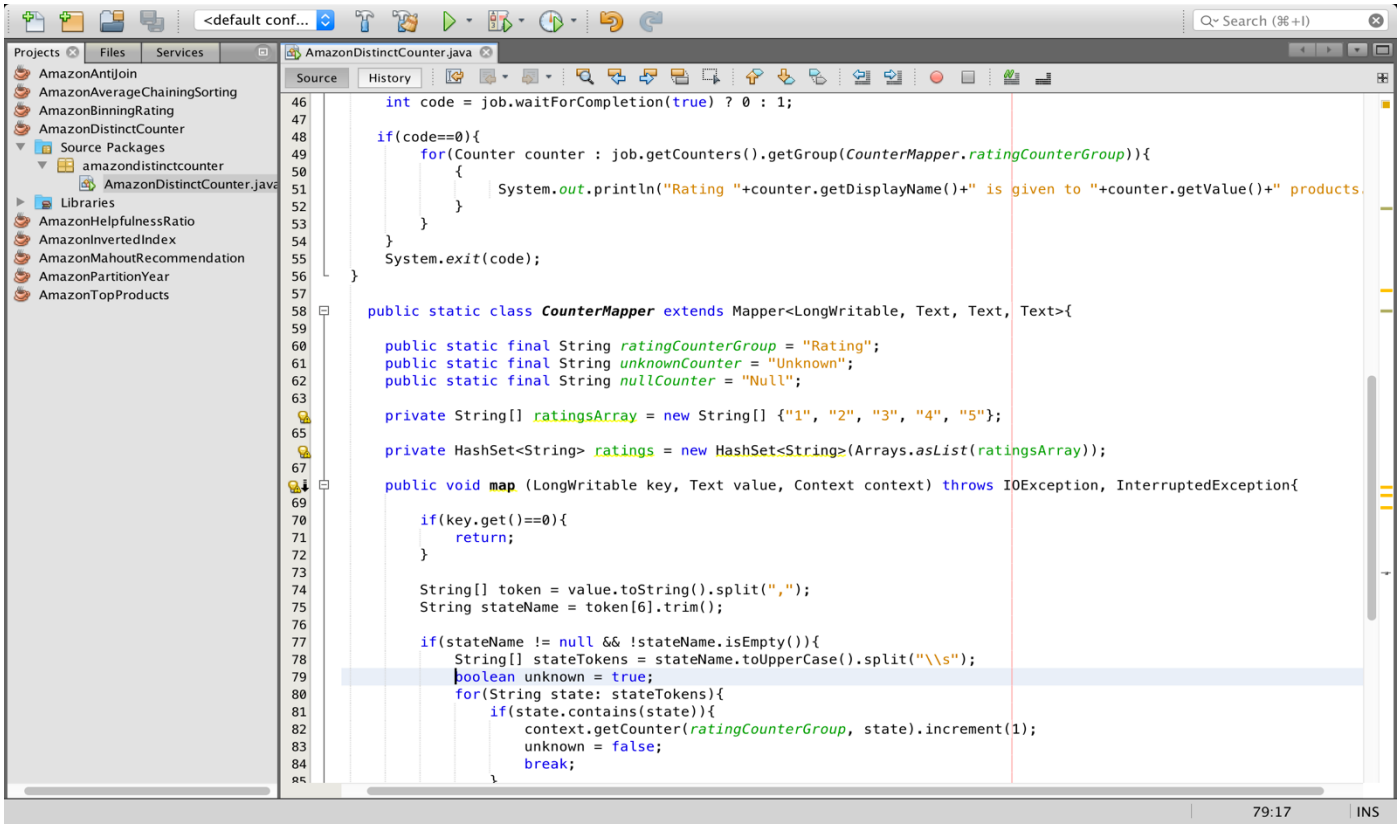
The file shows the ProductId and all the UserIds of the users who reviewed that particular product.

part-r-00000(InvertedIndex)

```
B00002N8SM      A1GQUBNRZK3RHI    A23KO0L75SEQ3K    A2WWCCK2PTY3SE    A17UIV0KM8JSV     A5W8ZREO7C1WK     A2579IQYG8Y9YF    A1S64UB4EOYFTK    A1LU93B4XI1PZZ    A32DW342WBJ6BX
A1J67DKGFR7TZ4    ASF1MKDA0N66O     A88YOFJG6IPOZ     A3EAHSVSDAYQ2K    A1G98WP2G0CMC     A29GWIJL7ZGXXZ    A2B7IZJ243KWFX    AIW48SP4GCIOO     AZKXR73X1CVIM     A3B28AM0CWI7BJ
A110QUC2ULFN7R    AXVQSCC43HE99     A8XKSGA2EA6GJ     A2281X5GBCKA6     A7E1DYOIQXM12     A2Z0O52PKZPQ4H    A1L227LG53GTLB    A392XPUTJDHSDJ    A1NH6UGER6N9EL    A19Q006CSFT011
A1FYH4S02BW7FN    AUE8TB5VHS6ZV     A1VSEBVZS81UCI    A3R3T5T1K8IN6P    A11DU8T1ZXPRT7    A71C9WY0242QU     A1CFZ6SHHRPRUY    AGF5R0NX88XPH     A1J4110YC5IFAB
B00002NCJC      A196AJHU9EASJN    A13RRPGE79XFFH
B00002Z754      A29Z5PI9BW2PU3    A3B8RCEI0FXFI6
B00005V3DC      A8KY7S48EW7LW     A2ZYCEEYBUQZND    APASCXWTM041
B00006IDJO      A1NKE5OYRM326V
B00006IDJU      A1NRU5R76B11AF    A2CA4TJR6L3J2W
B000084DVR      A1UGDJP1ZJWVPF    A3DKGXWUEP1AI2
B000084E1U      A3DH85EYHW4AQH
B000084E6V      A27VY0I52W7SP7    AS3ML2E3JY0DX     A2XO9ABXN2S6TZ    A1UW1OD4G8752Q    A1IMWSB87SZK18    AO7LLVTUYMMW7     A4PH5BJ4SGF62     AHB5QUSGWA8F2     A13O2H6WKB42OH
A1ZZ5T7VB393K6    A1HU3LZCFJXGI4    A52QHHSLLSPO8     A2NPQM672LHELI    A1STJZ70SC7HP6    A1OA0Y26U1RYLL    A2DWWQISF25UC5    A4VXM7QTDQHBC     AOQC0MVBBP17N     AYYWKPMJ2XZ2H
A12MWL2XMX7SLL    A3G5T9AHNNJ7E5    A2DF47YCA4EM7S    A1LBIZLET9CW1I    A1D3XIXEHNZRY0    A2CT6SCGRHN977    A31HD62RV56U8O    AHXFQHC9YAK9M     A2CF6NAQ8G7B58    A2HBKE72S3OUZR
A2EYIMR55IE4NG    A3OUMX34RWF098    A1GUVN6T6IZ9N7    AFG2JX5R69205     A1VL3GTRKWO3EC    A20TPVIKXZZP0E    A3H1K0HJ6BGXRL    A21IMS76KY1VPC    A2QUOTYV48HZ55    A1X2GN2X0DCPG2
AT4KWC45KUTNV     ADSG6OSKSUYY9     ATSKFCYZ9LCTN     A2K3C825MH5PDR    A3VK2ZI02M505B    AFX2WEXN91ANR     A28588OPKE5KO7    A2KI5N0HEKQTOY    A2SA0KSSQSS678    A36JDJ4WT7K578
A3ANSLZBVSRHOF    A1AGUO5PPZQL6S    AS2P0PSAT3B1F     A1RFEP3ZDU8ZJT    A1LE76IRVZNY7C    A1MFXOLD6TN0PM    A26Q4M802KH8XT    A1PYK03XUR6WER    A1V3Q7J6Z1HHI0    A6B9HJRJVEKOO
A230CK5MMCT6TW    A12CKHMMEEHBZO    A31GSJRXTBK66U    A1SBCD95KWI93R    ADIZU3LWY6III     A9B0WIHS884L3     A2JLKVQYDTEKZZ    A23MI22RUQ685D    A20H5NAANECBYF    A1KGIG7PA9UYEH
A3UHYTRDPHRM5D    ASFRQDXD2EWXG     ABGHK3QZBH37T     AZWRC1QS9JW0X     A28B5IOOQIF61H    AEBMHARILGYOF     A16II0K7CAWURW    AB8GU3MWHYZI6     ALAYW1B1PF5V8     A1LL0CERMZZDQP
A19SY2K2T41DDF    AZLMRVLITZCIY     A3A6MEPNP0VYYK    A1RN9FGWEUHDPS    A3RHPGU1HNB16G    A1VEMH5514JVZL    A2P8NUNZRBC6Y7    A3OCH7A99FLRH     A162K7K0O8G0AD    A2LN4MDWXZM7CG
A1AS4IS5LKV7PB    A1ARV0S39P37NI    ATQJBSU2GAI7S     A3UDDP74X7E1U5    A36HVLNN9HWZ4X    A7B48AJT6IC0A     A2HRGWAAMXYUS1    A395S23BFSJGAN    AY3KER0ZMFPGS     APXX698CXUSYS
AIP03JKZDR7WY     A2BSMMBY70WM1X    A2K5AFTU4YOPPW    ADEJQDAUY2YHG     AFLUPR1LM26N9     A2MV3HSRUBWA2B    AIJS0NM36XH7P     A3ZI4SL3SGY83     A1GL3V23LMM8KD    AS0DQ3WXSC97C
A3QDC3EGHCNCKB    A230Q6PBBH78AF    A2KAMN8AJOZKHI    A8891HVRDJAM6     A1C5JH384RHVRD    A1GSB2MRA6LV6B    A19X7O0NHL5ZNE    A2ND5QEVGM29KB    A357JT0S0BENA8    A2YLXPL6AVFONF
A1ITZX68GSGO8H    A3IMWFY597E2K8    AKHFT3B0I2ENA     AZ4IFJ01WKBTB     A2P1IH7383JX8Y    A19F05N7HH5U5O    AB41EZE5BVA1S     A3ETJXGI2FL00O    APS7AXG5HX2E0     AE442AGZLAGPU
A37FHDFU0H1G5J    A25BYVSHPX3YL9    AY82CFCNSYSMW     A2DN0DF3IE4K4M    A2211XUAO701D7    A2Z1BEXTMG0ZK2    A1S29177DRDWLY    A3GKXS2MS52BW1    A34AMFQ5JZO83J    A2HM80PTMV0ZGP
A3GSR3PAFBNZX4    A1V6ENRZBNFRQK    A2IVA2CMIP2UHI    A8891HVRDJAM6     A1PUM736GQ441P    A14IT3TRKVRP6O    A3ECY09BLZCQZ8    A1EQDSIRUS97GG    A2PKN7PKPJKEDS    A24PNF4Q2TK8N3
A3L2PTMSUJZF3T    AU80CU2W4059N     A2WTNYG5V4XDBP    A1XHFBJTBTPQV0    AH0ADDGUZ2J4Z     A25B1G19V71YL2    A2XRIHZ31GHAAV    A34TNH973Q9XT0    AES3U50KACKHP     AVX34RCAI3URC
A1NKMJGISMBU7O    AHI68O6B1LF1B     A3D64VZH7OE2E5    A11JWHLCLSOQ40     A2RBCF9F3MNZNO    A3GMN66FGTDLOY    A3T8XP4CHAYVQL    A1V6ENRZBNFRQK    A36470I4B4UBDO    A1K3VTQ7K565NR
APFZOD6YURK3G     A1F0MK3MMG40O2    A2URXZ0PM5BBAU    AMPUNRASRS43V     A19MVNNI6ZEV0B    A1KPDKD7VYRE3B    A38FSGH5ZVYDRK    A2F6HCTFKKJ9
B000084ED0      AC2IDFP2ICJXF
B000084EK4      A1Z54EM24Y40LL
B000084EK5      A1Z54EM24Y40LL
B000084EK6      A1Z54EM24Y40LL
B000084EK7      A1Z54EM24Y40LL
B000084EK8      A1Z54EM24Y40LL
B000084EK9      A1Z54EM24Y40LL    A2OVA909VD90P6    AUQIKXJAWMOK5
B000084EKA      A1Z54EM24Y40LL
B000084EKB      A1Z54EM24Y40LL
B000084EKC      A1Z54EM24Y40LL
B000084EKD      A1Z54EM24Y40LL
```

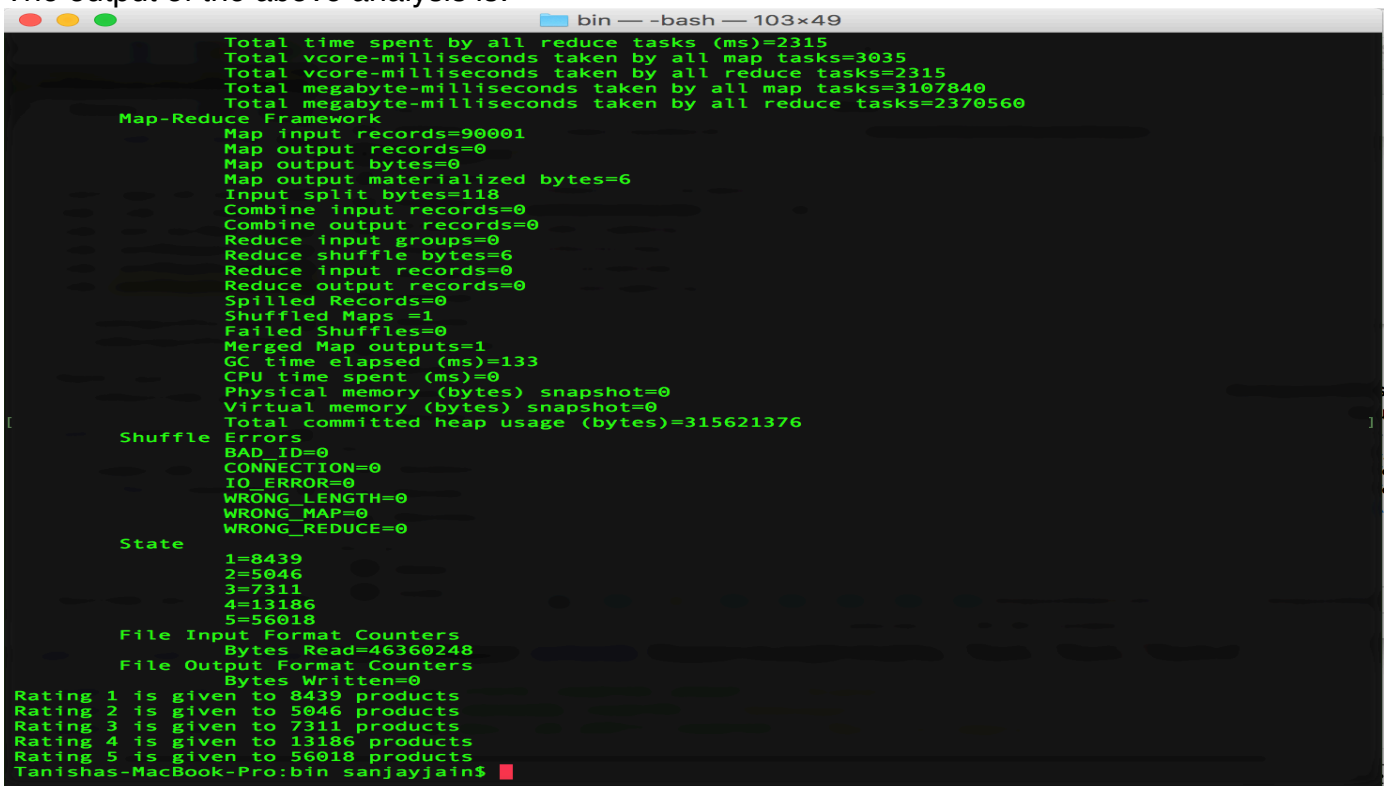
4) Amazon Fine Food Review Data: Amazon Counting with Counters (Numerical Summarization Pattern)

The counting with counters is a numerical summarization pattern where each attribute's number of occurrence can be counted. In this analysis, the counters are used to count the number of ratings appeared in the code from 1 – 5.



```
int code = job.waitForCompletion(true) ? 0 : 1;

if(code==0){
    for(Counter counter : job.getCounters().getGroup(CounterMapper.ratingCounterGroup)){
        {
            System.out.println("Rating "+counter.getDisplayName()+" is given to "+counter.getValue()+" products.
        }
    }
    System.exit(code);
}

public static class CounterMapper extends Mapper<LongWritable, Text, Text, Text>{

    public static final String ratingCounterGroup = "Rating";
    public static final String unknownCounter = "Unknown";
    public static final String nullCounter = "Null";

    private String[] ratingsArray = new String[] {"1", "2", "3", "4", "5"};

    private HashSet<String> ratings = new HashSet<String>(Arrays.asList(ratingsArray));

    public void map (LongWritable key, Text value, Context context) throws IOException, InterruptedException{

        if(key.get()==0){
            return;
        }

        String[] token = value.toString().split(",");
        String stateName = token[6].trim();

        if(stateName != null && !stateName.isEmpty()){
            String[] stateTokens = stateName.toUpperCase().split("\\s");
            boolean unknown = true;
            for(String state: stateTokens){
                if(state.contains(state)){
                    context.getCounter(ratingCounterGroup, state).increment(1);
                    unknown = false;
                    break;
```

The output of the above analysis is:

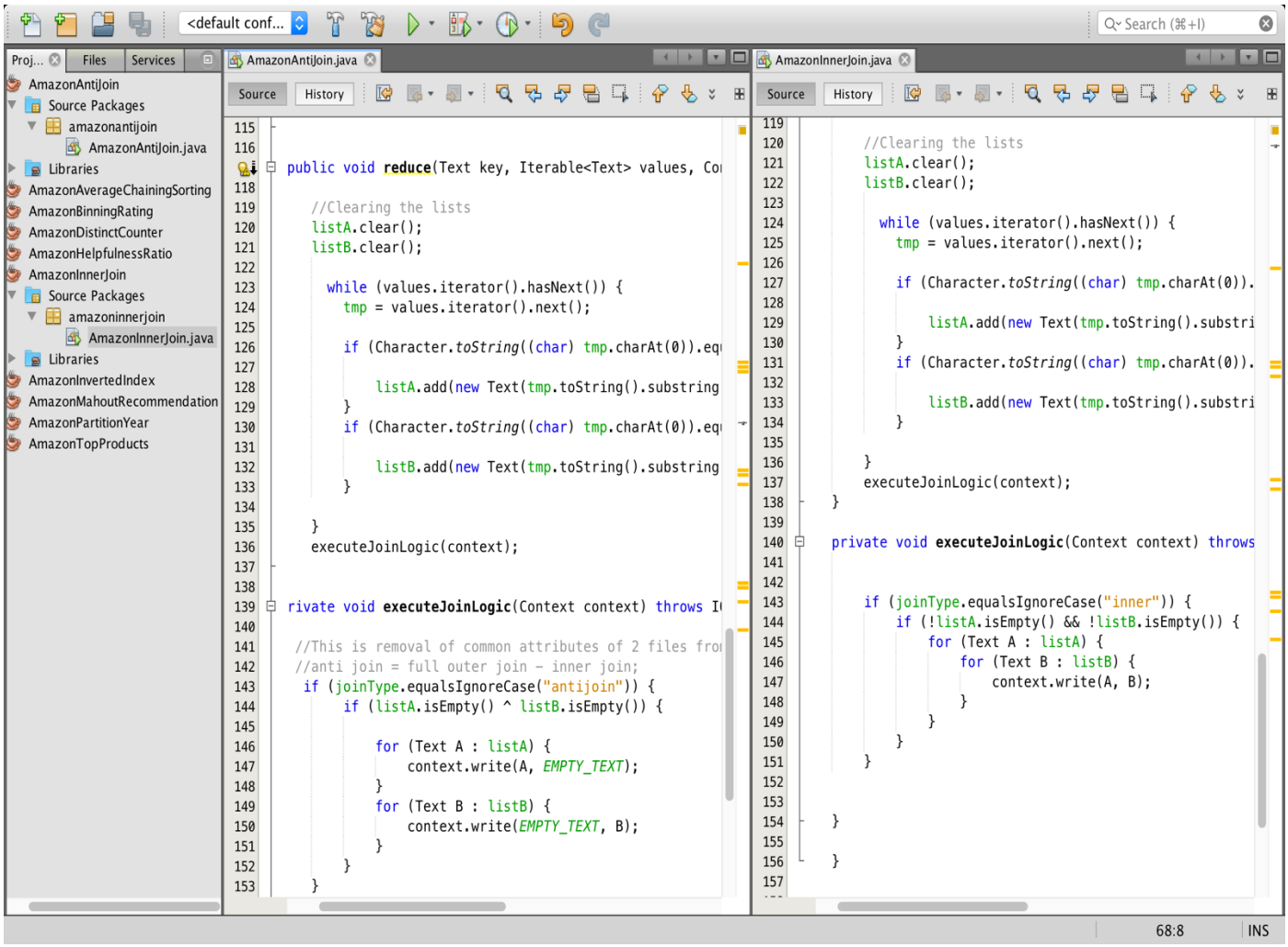


```
bin — -bash — 103×49

                Total time spent by all reduce tasks (ms)=2315
                Total vcore-milliseconds taken by all map tasks=3035
                Total vcore-milliseconds taken by all reduce tasks=2315
                Total megabyte-milliseconds taken by all map tasks=3107840
                Total megabyte-milliseconds taken by all reduce tasks=2370560
        Map-Reduce Framework
                Map input records=90001
                Map output records=0
                Map output bytes=0
                Map output materialized bytes=6
                Input split bytes=118
                Combine input records=0
                Combine output records=0
                Reduce input groups=0
                Reduce shuffle bytes=6
                Reduce input records=0
                Reduce output records=0
                Spilled Records=0
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=133
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=315621376
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        State
                1=8439
                2=5046
                3=7311
                4=13186
                5=56018
        File Input Format Counters
                Bytes Read=46360248
        File Output Format Counters
                Bytes Written=0
Rating 1 is given to 8439 products
Rating 2 is given to 5046 products
Rating 3 is given to 7311 products
Rating 4 is given to 13186 products
Rating 5 is given to 56018 products
Tanishas-MacBook-Pro:bin sanjayjain$
```

5) Amazon Fine Food Review Data: Reduce Side Join (Anti Join & Inner Join) (Join Pattern)

This pattern removes the common attributes of 2 files from a whole joining of 2 files. Anti-join is basically the full outer join minus the inner join.

Inner Join shows the output of the 2 files as one joining them on the common attributes only.



Output of Inner Join:

part-r-00000(InnerJoin).csv

```
83318,B005ZBZLT4,#oc-R115TNMSPFT9I7,3/11/12,2,3,2,"""Green"" K-cup packaging sacrifices flavor","Overall its just OK
when considering the price of other K-cups.  The SF Coffee K-cups do not look like other K-cups in that they do not
have the plastic base which contains and seals around the coffee filter.  This exposes the coffee to air and faster
oxidation.  When drinking the coffee it lacks a fresh flavor and tastes a bit stale.  The K-cups come in a sealed
plastic bag, but once its opened, the k-cups are exposed to air and they quickly go stale.<br /><br />The coffee
flavor is a bit weak as well.  The lack of the plastic housing around the coffee filter, in my opinion, causes the
water to flow thru the coffee too quickly and contributes to a weak flavor.<br /><br />I don't think I would buy these
again." 60588,#oc-R115TNMSPFT9I7,Jared,Saunders,38,Walo Mill,Ilagwu,NY
21719,B005DVVB9K,#oc-R119LM8D59ZW8Y,11/1/11,5,11,1,Not So Sweet Treat!,"I love chocolate so I was excited to see the
Chocolate Sampler Gift Basket by Back Mountain Baskets available in Vine to review. After reading a few other reviews
on the strange packaging, I was almost afraid to open it - who knows what I would find. My daughter had a strong
chocolate urge so we pulled off the wrappers.<br /><br />Oh my gosh, it is hard to know what to say. What was this
company thinking?<br /><br />Let's start with the packaging. At first glance the basket looks simple yet classy and
attractive and boasting what appears to be an assortment of delectable chocolates. That illusion is lost once you have
opened the package. There are paper wrappings around boxes with the product name. But the actual boxes are for common
grocery items like jello, fruit loops, flan etc. Can you imagine diving into a box of truffles that say Kenzi on the
surface only to find the box below is for jello? A box of jello whose expiration was several years ago. How long does
one keep old cereal boxes around, but more importantly, why would you package gift basket chocolate in old jello and
cereal boxes?<br /><br />One of the boxes has a computer generated label that has been cheaply made and taped to
it.<br /><br />So what is in the box? Well in one I pulled out a bag of peanuts that had melted and then solidified in
one chunk. In another were small brown things that I believe were supposed to be truffles. There were also some
wrapped cheap chocolate such as a chocolate graham cracker.<br /><br />The entire basket was unappetizing and
unappealing. I can't share how the chocolate tastes, because after pulling it out of jello and flan boxes and seeing
the melted condition, there was no way I was eating it despite my love for chocolate.<br /><br />With proper packaging
and shipping, there would be a much better result but as it is now, it should not be for sale.<br /><br />This is
definitely not a gift basket you will want to give."    19072,#oc-R119LM8D59ZW8Y,Minerva,Greene,24,Ulaba
Way,Fifemjov,IL
83657,B005ZBZLT4,#oc-R11DNU2NBKQ23Z,9/24/12,0,2,1,Terrible coffee,This coffee is horrible and unfortunately not
eligible for return. It tastes like burnt wood shavings. Looks like its going into the compost pile.    60877,#oc-
R11DNU2NBKQ23Z,Eliza,McKenzie,54,Hahuw Lane,Poftivjaz,MT
83478,B005ZBZLT4,#oc-R13EBF129DBX88,8/11/12,0,0,2,fog chaser coffee,"This coffee is not what I expected.  I thought it
would be a bold coffee similiar to Starbucks, but it is not, and it leaves a sediment in my cup, which other don't .
very disappointed and will not re order this brand again."      60723,#oc-R13EBF129DBX88,Austin,Pope,20,Jize
```

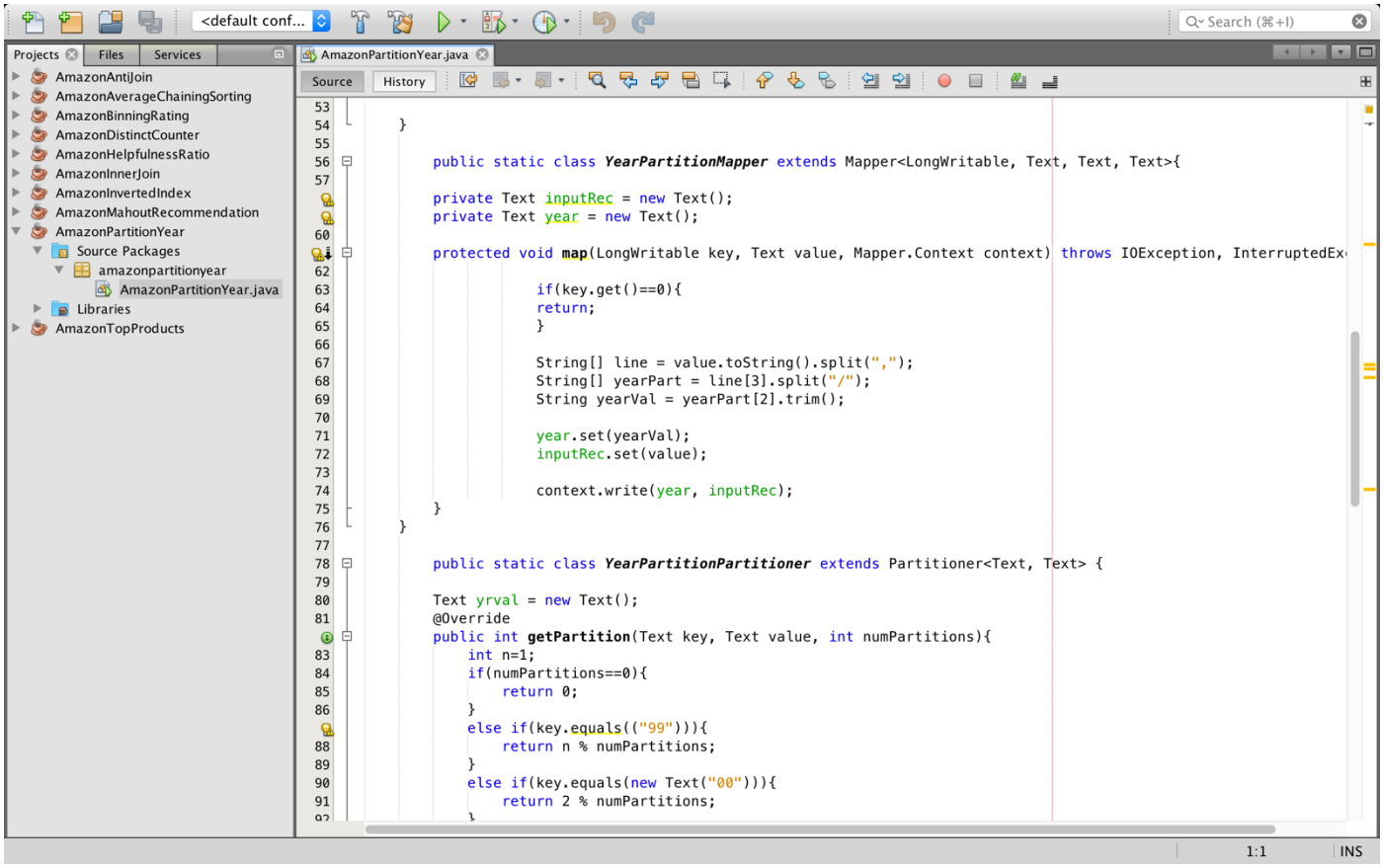
Output of Anti-Join:

part-r-00000(AntiJoin)

```
90000,,Anne,Weaver,65,Isor Square,Rumijeg,IL
89999,,John,Powers,25,Hopo Mill,Cefiora,MA
89998,,Warren,Flores,63,Wuiju Park,Rumona,WV
89997,,Bernard,Jensen,49,Wodgol Manor,Coigak,ND
89996,,Mathilda,Horton,27,Bavwe Lane,Kakgecid,VT
89995,,Dollie,Paul,33,Wisiw Street,Farzebza,WY
89994,,Charlie,Wagner,61,Nioza Drive,Latuhfa,KY
89993,,Corey,Tate,21,Ohudo Pike,Zojuklib,AK
89992,,Cordelia,Boyd,37,Azziw Key,Oleovewa,HI
89991,,Jimmy,Holt,27,Tefaw Heights,Rokunawo,CO
89990,,Michael,Byrd,19,Jagju Pike,Vefuud,LA
89989,,Maude,Wheeler,20,Bibu River,Rojahso,MO
89988,,Nellie,Edwards,48,Erair Square,Roimne,CO
89987,,Robert,Hubbard,21,Cuika Court,Vipkehtif,CT
89986,,Christine,Jefferson,54,Odikoh Loop,Noczudme,OH
89985,,Clifford,Ruiz,41,Ikimo Place,Otpopoh,CA
89984,,Francisco,Lynch,32,Karka Mill,Modusuta,SD
89983,,Ola,Reid,20,Revna River,Kajuzfi,OH
89982,,Rosa,Larson,45,Amimu Place,Lupciape,AL
89981,,Gordon,Willis,38,Cubi Plaza,Iwajamner,NY
89980,,Don,Joseph,38,Ahto Court,Gejkelrok,DE
89979,,Cecelia,Glover,33,Cakwiz Terrace,Lutubru,TN
89978,,Pauline,Pearson,30,Lungig Drive,Nusutaako,OR
89977,,Laura,Ball,59,Usafup View,Diagepav,TX
89976,,Lelia,Maxwell,27,Fomel Road,Lukeow,SD
89975,,Roy,Payne,30,Sujza Plaza,Cakjuvjuw,MN
89974,,Mitchell,Martinez,38,Papoj Junction,Weppati,DC
89973,,Mitchell,Nichols,54,Ucsic Way,Ewnovif,MD
89972,,Ricky,Strickland,25,Izgab Circle,Wonavor,CT
```

6) Amazon Fine Food Review Data: Partitioning (Organization Pattern)

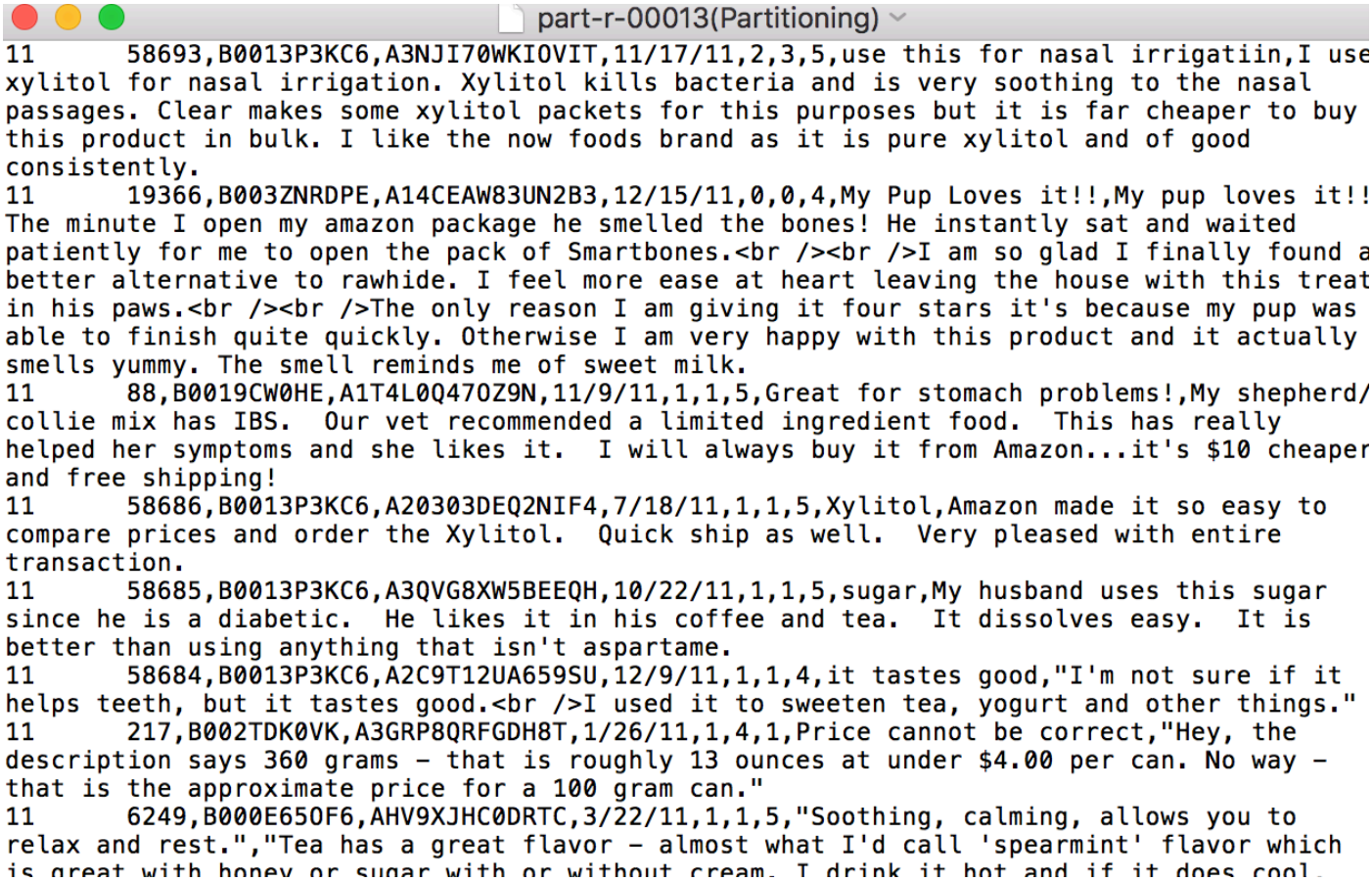
Partitioning is a pattern where data can be partitioned based on the number of partitioners mentioned in the Custom partitioner.



The number of partitioners is equal to the number of reducers and so there are 13 output files. Two of them are shown below:

part-r-00000(Partitioning)

```
12      90000,B004RBEQZI,AFOZ2DYRE9XJ6,3/1/12,0,0,4,Liked these better as a kid..,"Taste-
wise, these are pretty decent.  I found that I got tired of the orange flavor that is
added.  I remember liking them a bit more when I was younger."
12      89999,B004RBEQZI,AXIBU8SOYD3RK,4/22/12,0,0,5,Delicious,"I forgot chocolate had a
low melting point, and when I got it, the temperature was about 92 degrees Fahrenheit. Th
mailman even put it in my black mailbox. Needless to say, it melted. But, I put it in the
freezer and it was just fine! Very good! But, don't buy chocolate online."
12      89997,B002LANN56,A17UNMURMLX0ZE,9/24/12,0,0,5,The dogs rate it *****,"First the
dogs love it, this food must smell great, taste great, and have a texture that the animal
really enjoy. If you have a dog that is a picky eater then you might give this a try.<br
><br />As to the issues with a lot of the ingredients being secondary use or
byproducts.<br />I am not a processed food expert, I do see that the number one ingredien
is beef and I generally assume that animal food is made using safe secondary use food
products which seems ok to me. We take the best cuts of beef and put them on the grill,
next we grind up the next best to make our burgers, lunchmeat, and sausage, perhaps some
of that same product as well as what is left goes to the animals we care for, should we
just throw it all away instead? We pull oils and milk from soybeans to cook with and to
make tofu. What happens to the fiber? We feed it to animals. Since we as humans eat whole
soybeans I have to imagine that eating the fiber and remaining protein is not a terrible
thing.<br /><br />Certainly all animals and humans have some allergies and digestive
issues. Some people can't drink milk and some animals apparently don't do well with soy w
don't seem to have an issue in our house. If your dog has issues and needs rice or
something else then you should pick another food for them. If you have a healthy food tha
the dogs don't love, sprinkle a bit of this on top and perhaps you will have luck getting
them to eat the less flavorful and perhaps healthier food with this addition. However thi
may be all you need to feed as according to Purina ""Animal feeding tests using
Association of American Feed Control Officials (AAFCO) procedures substantiate that Chef
Michael's Grilled Sirloin Steak Flavor and potato & green bean garnishes provides complet
and balanced nutrition for maintenance of adult dogs."""
12      89996,B002LANN56,A2065HBMYDXJ1S,9/25/12,0,0,4,Seems better for your dog,"I like
```

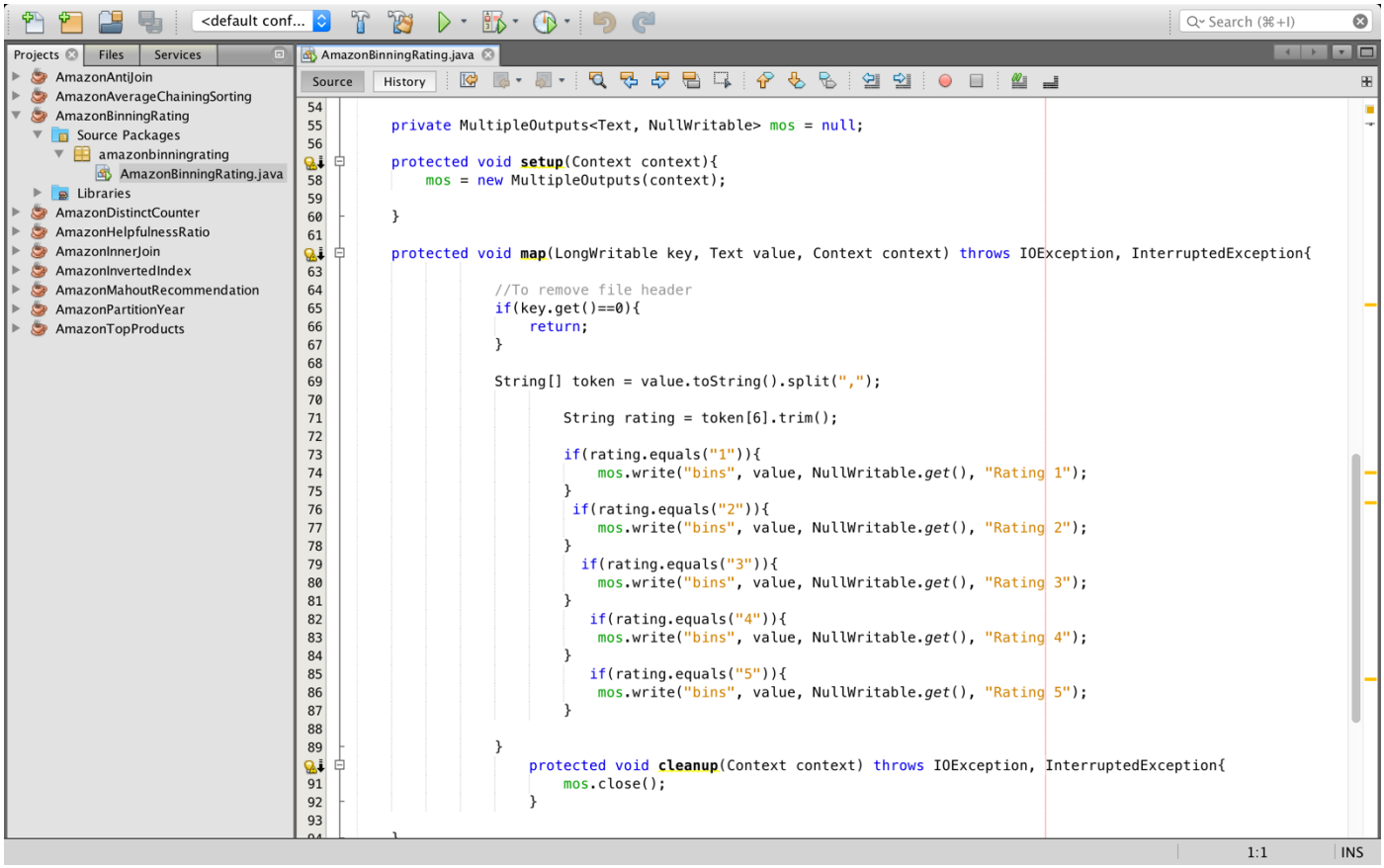

part-r-00013(Partitioning)
11 58693,B0013P3KC6,A3NJI70WKIOVIT,11/17/11,2,3,5,use this for nasal irrigatiin,I use xylitol for nasal irrigation. Xylitol kills bacteria and is very soothing to the nasal passages. Clear makes some xylitol packets for this purposes but it is far cheaper to buy this product in bulk. I like the now foods brand as it is pure xylitol and of good consistently.
11 19366,B003ZNRDPE,A14CEAW83UN2B3,12/15/11,0,0,4,My Pup Loves it!!,My pup loves it!! The minute I open my amazon package he smelled the bones! He instantly sat and waited patiently for me to open the pack of Smartbones.

I am so glad I finally found a better alternative to rawhide. I feel more ease at heart leaving the house with this treat in his paws.

The only reason I am giving it four stars it's because my pup was able to finish quite quickly. Otherwise I am very happy with this product and it actually smells yummy. The smell reminds me of sweet milk.
11 88,B0019CW0HE,A1T4L0Q47OZ9N,11/9/11,1,1,5,Great for stomach problems!,My shepherd/ collie mix has IBS. Our vet recommended a limited ingredient food. This has really helped her symptoms and she likes it. I will always buy it from Amazon...it's $10 cheaper and free shipping!
11 58686,B0013P3KC6,A20303DEQ2NIF4,7/18/11,1,1,5,Xylitol,Amazon made it so easy to compare prices and order the Xylitol. Quick ship as well. Very pleased with entire transaction.
11 58685,B0013P3KC6,A3QVG8XW5BEEQH,10/22/11,1,1,5,sugar,My husband uses this sugar since he is a diabetic. He likes it in his coffee and tea. It dissolves easy. It is better than using anything that isn't aspartame.
11 58684,B0013P3KC6,A2C9T12UA659SU,12/9/11,1,1,4,it tastes good,"I'm not sure if it helps teeth, but it tastes good.
I used it to sweeten tea, yogurt and other things."
11 217,B002TDK0VK,A3GRP8QRFGDH8T,1/26/11,1,4,1,Price cannot be correct,"Hey, the description says 360 grams - that is roughly 13 ounces at under $4.00 per can. No way - that is the approximate price for a 100 gram can."
11 6249,B000E65OF6,AHV9XJHC0DRTC,3/22/11,1,1,5,"Soothing, calming, allows you to relax and rest.","Tea has a great flavor - almost what I'd call 'spearmint' flavor which is great with honey or sugar with or without cream. I drink it hot and if it does cool

7) Amazon Fine Food Review Data: Binning (Organization Pattern)

This is similar to the partitioning pattern described above but the only difference is that this is a mapper-only job and the bins have to be specified in the mapper setup method.



```
Projects | Files | Services
AmazonAntiJoin
AmazonAverageChainingSorting
AmazonBinningRating
  Source Packages
    amazonbinningrating
      AmazonBinningRating.java
  Libraries
AmazonDistinctCounter
AmazonHelpfulnessRatio
AmazonInnerJoin
AmazonInvertedIndex
AmazonMahoutRecommendation
AmazonPartitionYear
AmazonTopProducts

AmazonBinningRating.java
Source | History
```

```
54
55      private MultipleOutputs<Text, NullWritable> mos = null;
56
57      protected void setup(Context context){
58          mos = new MultipleOutputs(context);
59
60      }
61
62      protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException{
63
64                  //To remove file header
65                  if(key.get()==0){
66                      return;
67                  }
68
69                  String[] token = value.toString().split(",");
70
71                          String rating = token[6].trim();
72
73                          if(rating.equals("1")){
74                              mos.write("bins", value, NullWritable.get(), "Rating 1");
75                          }
76                           if(rating.equals("2")){
77                              mos.write("bins", value, NullWritable.get(), "Rating 2");
78                          }
79                            if(rating.equals("3")){
80                              mos.write("bins", value, NullWritable.get(), "Rating 3");
81                          }
82                             if(rating.equals("4")){
83                              mos.write("bins", value, NullWritable.get(), "Rating 4");
84                          }
85                             if(rating.equals("5")){
86                              mos.write("bins", value, NullWritable.get(), "Rating 5");
87                          }
88
89                  }
90          protected void cleanup(Context context) throws IOException, InterruptedException{
91              mos.close();
92              }
93
94      }
```

1:1 INS

The outputs of the bins are:

Rating 1-m-00000(Binning).csv

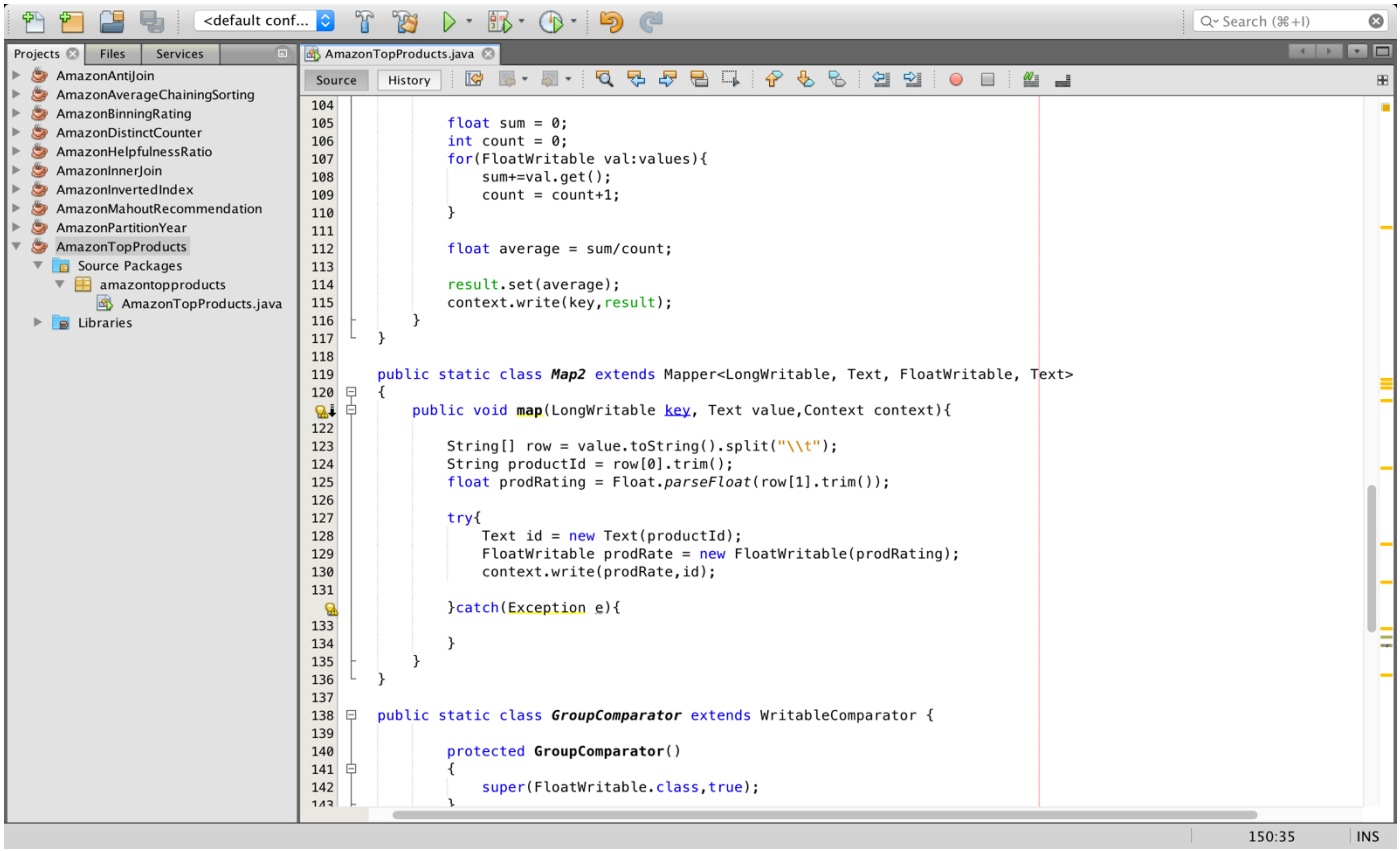
```
2,B00813GRG4,A1D87F6ZCVE5NK,9/6/12,0,0,1,Not as Advertised,"Product arrived labeled as Jumbo Salted
Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor
intended to represent the product as ""Jumbo""."
13,B0009XLVG0,A327PCT23YH90,6/12/12,1,1,1,My Cats Are Not Fans of the New Food,"My cats have been happily
eating Felidae Platinum for more than two years. I just got a new bag and the shape of the food is
different. They tried the new food when I first put it in their bowls and now the bowls sit full and the
kitties will not touch the food. I've noticed similar reviews related to formula changes in the past.
Unfortunately, I now need to find a new food that my cats will eat."
27,B001GVISJM,A3RXAU2N8KV45G,3/24/12,0,1,1,Nasty No flavor,"The candy is just red , No flavor . Just  plan
and chewy .  I would never buy them again"
51,B001EO5QW8,A108P30XVUFKXY,2/18/08,0,7,1,Don't like it,"This oatmeal is not good. Its mushy, soft, I don't
like it. Quaker Oats is the way to go."
63,B001EO5TPM,A1E09XGZUR78C6,7/5/08,2,2,1,stale product.,Arrived in 6 days and were so stale i could not eat
any of the 6 bags!!
74,B0059WXJKM,A25VFHVGI4CFTP,4/29/12,0,1,1,Warning!  WARNING!  -ALCOHOL SUGARS!,"Buyer Beware Please! This
sweetener is not for everybody.  Maltitol is an alcohol sugar and can be undigestible in the body.  You will
know a short time after consuming it if you are one of the unsuspecting many who cannot digest it by the
extreme intestinal bloating and cramping and massive amounts of gas a person can experience. Nausea,
diarrhea & headaches can also be experienced. I learned my lesson the hard way years ago when I fell in love
with the sugar-free chocolates suzanne sommers used to sell.  I thought I'd found sugar-free chocolate
nirvana at first taste but the bliss was short lived when the terrible side effects of maltitol kicked in.
The discomfort was unlike anything I had ever felt before.  I blew up like a balloon and had very painful
abdominal cramping.  As the symptoms passed, this too was very unpleasant.  Though hard for me to believe a
low calorie sweetener could be the culprit, all symptoms were gone when I stopped eating the chocolate. My
hunch it had something to do with the maltitol were unfortunately confirmed for me about a year later when I
purchased some delicious sugar-free popcorn at my local market.  The taste was amazing and while I was
looking at the label wondering what could possibly make this yummy, new sugarfree treat taste so good, my
heart sank when I followed the little asterisk next to sugarfree sweetener* down to the very bottom of the
label and read ""maltitol"" in tiny little letters! Thank goodness I'd eaten only a little. I still ended up
with all the same side-effects but for a much shorter duration. Some people can use maltitol to their
```

Rating 5-m-00000(Binning).csv

```
1,B001E4KFG0,A3SGXH7AUHU8GW,4/26/11,1,1,5,Good Quality Dog Food,I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than  most.
5,B006K2ZZ7K,A1UQRSCLF8GW1T,10/20/12,0,0,5,Great taffy,"Great taffy at a great price.  There was a wide assortment of yummy taffy.  Delivery was very quick.  If your a taffy lover, this is a deal."
7,B006K2ZZ7K,A1SP2KVKFXXRU1,6/19/12,0,0,5,Great!  Just as good as the expensive brands!,"This saltwater taffy had great flavors and was very soft and chewy.  Each candy was individually wrapped well.  None of the candies were stuck together, which did happen in the expensive version, Fralinger's.  Would highly recommend this candy!  I served it at a beach-themed party and everyone loved it!"
8,B006K2ZZ7K,A3JRGQVEQN31IQ,5/2/12,0,0,5,"Wonderful, tasty taffy",This taffy is so good.  It is very soft and chewy.  The flavors are amazing.  I would definitely recommend you buying it.  Very satisfying!!
9,B000E7L2R4,A1MZYO9TZK0BBI,11/22/11,1,1,5,Yay Barley,Right now I'm mostly just sprouting this so my cats can eat the grass. They love it. I rotate it around with Wheatgrass and Rye too
10,B00171APVA,A21BT40VZCCYT4,10/25/12,0,0,5,Healthy Dog Food,This is a very healthy dog food. Good for their digestion. Also good for small puppies. My dog eats her required amount at every feeding.
11,B0001PB9FE,A3HDKO7OW0QNK4,2/7/05,1,1,5,The Best Hot Sauce in the World,"I don't know if it's the cactus or the tequila or just the unique combination of ingredients, but the flavour of this hot sauce makes it one of a kind!  We picked up a bottle once on a trip we were on and brought it back home with us and were totally blown away!  When we realized that we simply couldn't find it anywhere in our city we were bummed.<br /><br />Now, because of the magic of the internet, we have a case of the sauce and are ecstatic because of it.<br /><br />If you love hot sauce..I mean really love hot sauce, but don't want a
```

8) Amazon Fine Food Review Data: Top 100 Products (Filtering Pattern)

This filtering pattern outputs the top products and the output file shows the ProductIds and their corresponding rating.



```
        float sum = 0;
        int count = 0;
        for(FloatWritable val:values){
            sum+=val.get();
            count = count+1;
        }

        float average = sum/count;

        result.set(average);
        context.write(key,result);
    }
}

public static class Map2 extends Mapper<LongWritable, Text, FloatWritable, Text>
{
    public void map(LongWritable key, Text value,Context context){

        String[] row = value.toString().split("\\t");
        String productId = row[0].trim();
        float prodRating = Float.parseFloat(row[1].trim());

        try{
            Text id = new Text(productId);
            FloatWritable prodRate = new FloatWritable(prodRating);
            context.write(prodRate,id);

        }catch(Exception e){

        }
    }
}

public static class GroupComparator extends WritableComparator {

        protected GroupComparator()
        {
            super(FloatWritable.class,true);
```

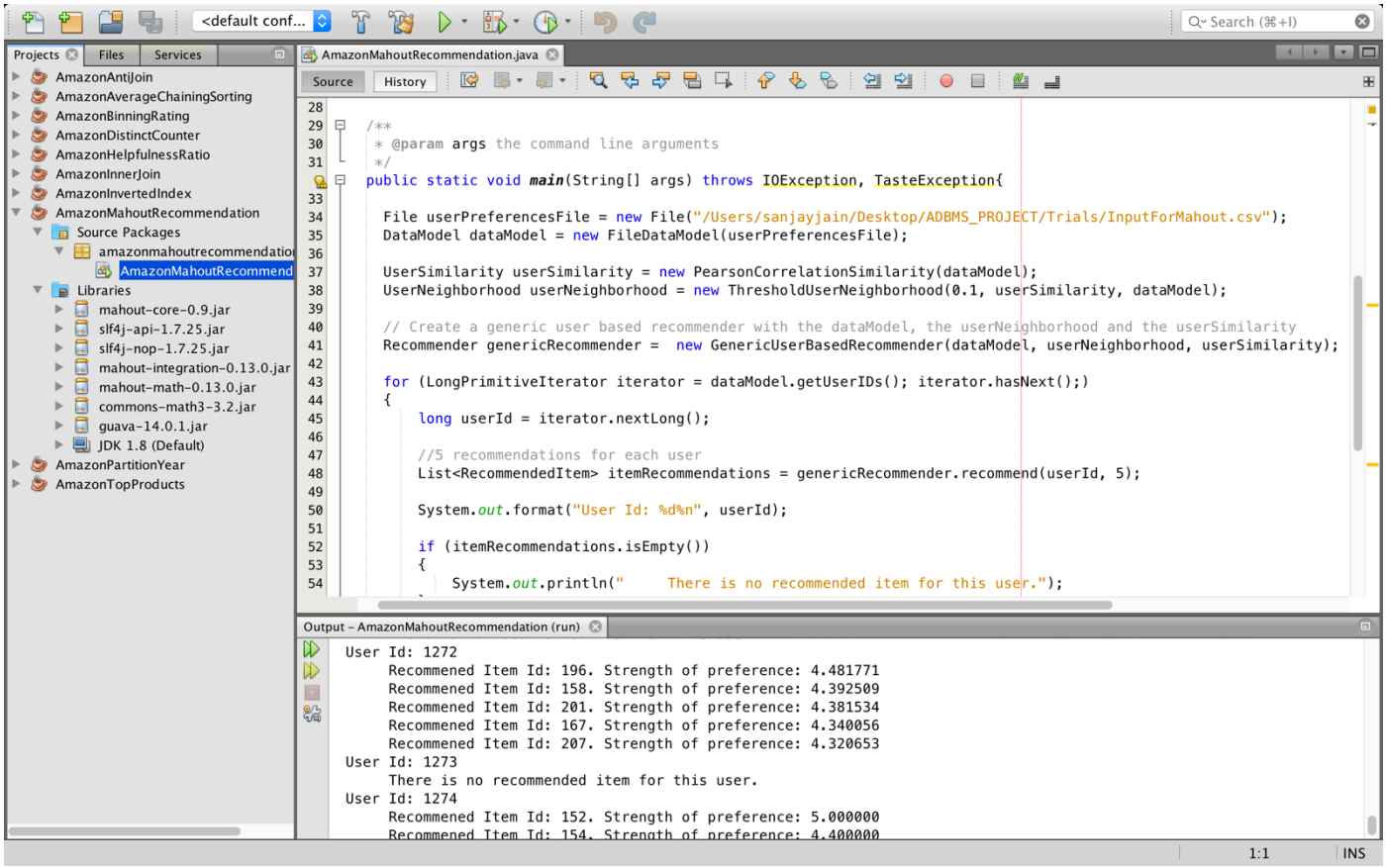
Output:

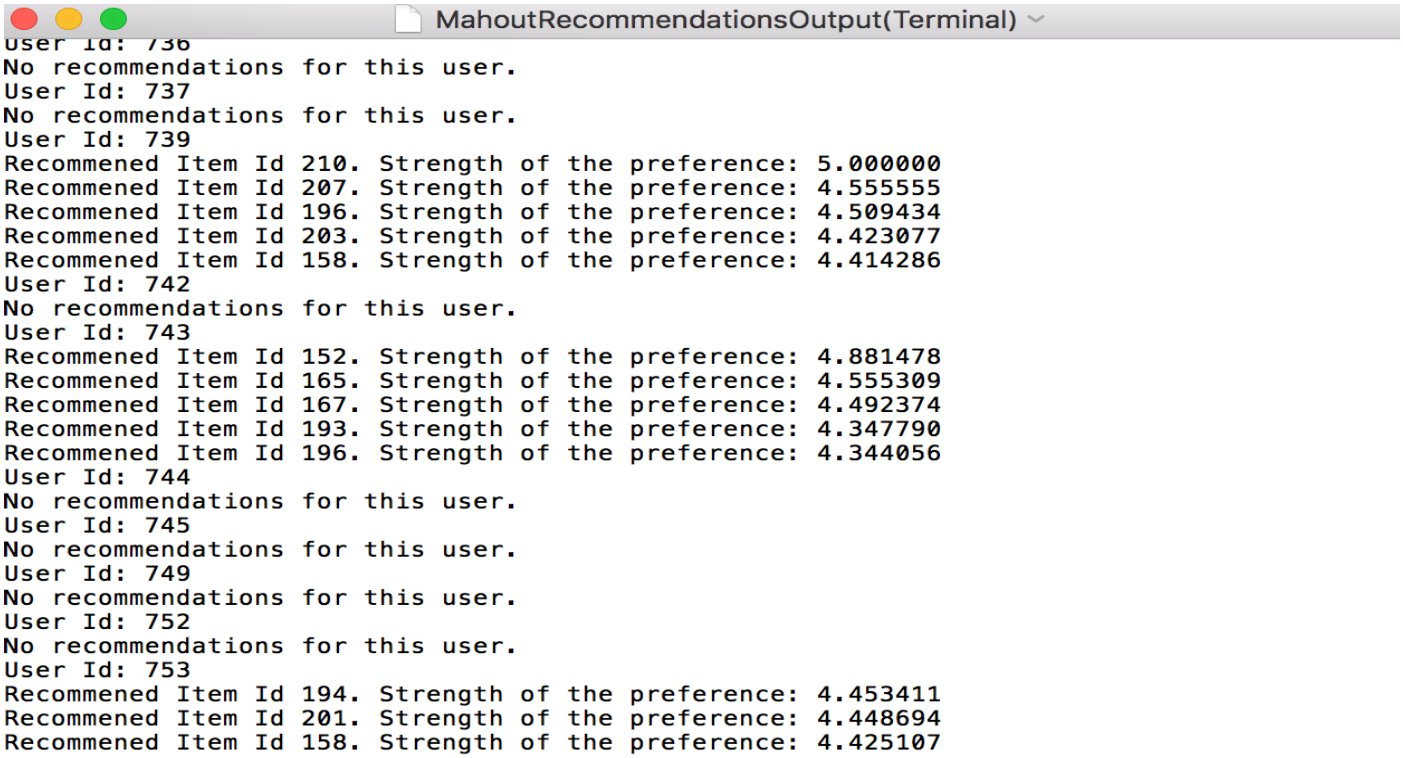
part-r-00000(Top100productswithrating)

```
5.0     B000084EKC
5.0     B009C16ZP2
5.0     B000084EKG
5.0     B000084EKL
5.0     B00995MODI
5.0     B00993CW3M
5.0     B0098MEKQQ
5.0     B0097KJ6ZY
5.0     B000084F11
5.0     B000087BJ4
5.0     B00008DF6C
5.0     B0096RP84Q
5.0     B00008DFR5
5.0     B0096E5196
5.0     B00961CUXO
5.0     B00959DMWK
5.0     B0094K4VNE
5.0     B0093NIWVO
5.0     B00009608Y
5.0     B0093A5RU6
5.0     B0093A5PDK
5.0     B00934WBNI
5.0     B0092XAMDQ
5.0     B0000960IV
5.0     B000090LEW
5.0     B0092X7B5S
5.0     B0092VQCM8
5.0     B0092SQT2E
5.0     B000090LFC
5.0     B0091J1QRC
5.0     B0091ITGW0
5.0     B0000AH3OX
5.0     B00910LUQS
5.0     B0000CDBRH
5.0     B008ZL3XS2
5.0     B0000CDBRJ
5.0     B008Z1EREG
5.0     B008YKMCNG
5.0     B008YGWIZM
5.0     B008YFFNJ6
5.0     B008YC9SMW
5.0     B0000CDBRP
5.0     B008YA1TNA
5.0     B008Y5NE10
5.0     B0000CE29E
5.0     B0000CEOKH
5.0     B0000CERHH
5.0     B008VRDF42
5.0     B0000CERXB
5.0     B008V20UV8
5.0     B008UF5JQC
5.0     B0000CGFCD
5.0     B0000CGFSC
5.0     B008U0DWV6
```

9) Apache Mahout:

Apache Mahout is mainly used for the recommendations and in this analysis, Mahout libraries and Java code helps find the recommendations for the users. The usual alphanumeric Ids cannot be used and the input is an edited file with three columns – userId (long), productId (long) and product ratings (int). The recommendation here is user-based.

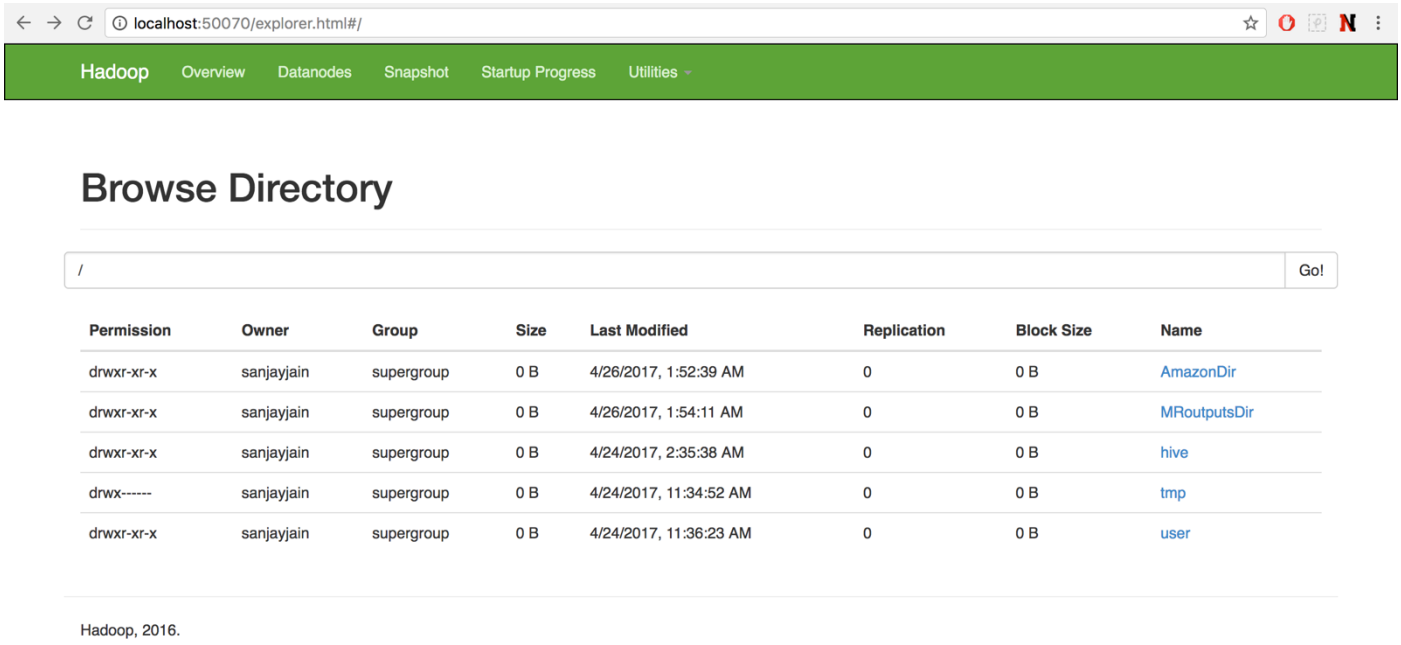




10) Apache Hive

Apache Hive provides a SQL-like interface to data stored in HDP(Hortonworks Data Platform) whose component is Hive.

Hive can run on the localhost:50070 and the data is stored inside the user folder created automatically upon installation and creation of tables.



Hive queries are used to create schema, table, load bulk data and join patterns. Here the code of hive inner join and full outer join is shown:

FullOuterJoinOutput(26secs)

```
localhost: starting nodemanager, logging to /usr/local/bin/hadoop-2.7.3/logs/yarn-sanjayjain-nodemanager-
Tanishas-MacBook-Pro.local.out
Tanishas-MacBook-Pro:sbin sanjayjain$ jps
31396 SecondaryNameNode
31509 ResourceManager
31637 Jps
31209 NameNode
31291 DataNode
31596 NodeManager
Tanishas-MacBook-Pro:sbin sanjayjain$ cd ..
Tanishas-MacBook-Pro:hadoop-2.7.3 sanjayjain$ cd ..
Tanishas-MacBook-Pro:bin sanjayjain$ cd apache-hive-2.1.0-bin/
Tanishas-MacBook-Pro:apache-hive-2.1.0-bin sanjayjain$ cd bin
Tanishas-MacBook-Pro:bin sanjayjain$ ./hive

Logging initialized using configuration in jar:file:/usr/local/bin/apache-hive-2.1.0-bin/lib/hive-
common-2.1.0.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> use review;
OK
Time taken: 0.971 seconds
hive> select * from amazonfoodreview a
    > join profiles b on a.userid = b.userid;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = sanjayjain_20170424174246_1897b6f1-5406-453e-bb9d-91014e613869
Total jobs = 1
2017-04-24 17:42:55     Starting to launch local task to process map join;      maximum memory = 477626368
2017-04-24 17:42:57     Dump the side-table for tag: 1 with group count: 64693 into file: file:/var/folders/
w9/4ccpbf0x3zl_dnpyvy9p3l3c0000gn/T/sanjayjain/88aa9c3b-d3fa-41cf-9918-24598370885e/
hive_2017-04-24_17-42-46_579_3191866684908100166-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2017-04-24 17:42:57     Uploaded 1 File to: file:/var/folders/w9/4ccpbf0x3zl_dnpyvy9p3l3c0000gn/T/sanjayjain/
88aa9c3b-d3fa-41cf-9918-24598370885e/hive_2017-04-24_17-42-46_579_3191866684908100166-1/-local-10004/HashTable-
Stage-3/MapJoin-mapfile01--.hashtable (6037966 bytes)
2017-04-24 17:42:57     End of local task; Time Taken: 2.152 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1493070008739_0001, Tracking URL = http://Tanishas-MacBook-Pro.local:8088/proxy/
application_1493070008739_0001/
Kill Command = /usr/local/bin/hadoop-2.7.3/bin/hadoop job  -kill job_1493070008739_0001
```

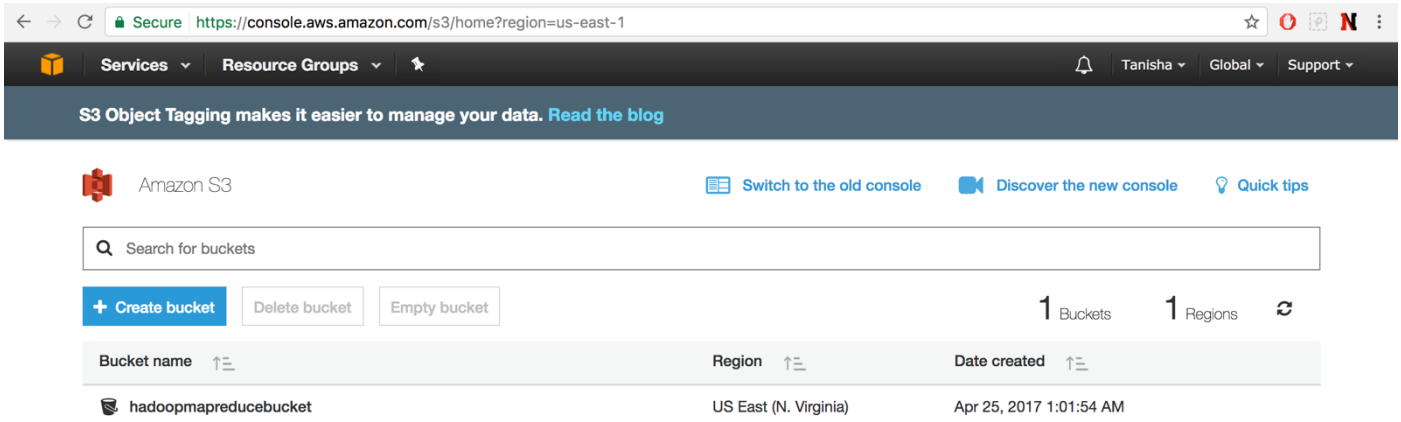
InnerJoinOutput(TimeTaken30secs)

```
Tanishas-MacBook-Pro:sbin sanjayjain$ cd ..
Tanishas-MacBook-Pro:hadoop-2.7.3 sanjayjain$ cd ..
Tanishas-MacBook-Pro:bin sanjayjain$ cd apache-hive-2.1.0-bin/
Tanishas-MacBook-Pro:apache-hive-2.1.0-bin sanjayjain$ cd bin
Tanishas-MacBook-Pro:bin sanjayjain$ ./hive

Logging initialized using configuration in jar:file:/usr/local/bin/apache-hive-2.1.0-bin/lib/hive-
common-2.1.0.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> use review;
OK
Time taken: 0.971 seconds
hive> select * from amazonfoodreview a
    > join profiles b on a.userid = b.userid;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = sanjayjain_20170424174246_1897b6f1-5406-453e-bb9d-91014e613869
Total jobs = 1
2017-04-24 17:42:55     Starting to launch local task to process map join;      maximum memory = 477626368
2017-04-24 17:42:57     Dump the side-table for tag: 1 with group count: 64693 into file: file:/var/folders/
w9/4ccpbf0x3zl_dnpyvy9p3l3c0000gn/T/sanjayjain/88aa9c3b-d3fa-41cf-9918-24598370885e/
hive_2017-04-24_17-42-46_579_3191866684908100166-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2017-04-24 17:42:57     Uploaded 1 File to: file:/var/folders/w9/4ccpbf0x3zl_dnpyvy9p3l3c0000gn/T/sanjayjain/
88aa9c3b-d3fa-41cf-9918-24598370885e/hive_2017-04-24_17-42-46_579_3191866684908100166-1/-local-10004/HashTable-
Stage-3/MapJoin-mapfile01--.hashtable (6037966 bytes)
2017-04-24 17:42:57     End of local task; Time Taken: 2.152 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1493070008739_0001, Tracking URL = http://Tanishas-MacBook-Pro.local:8088/proxy/
application_1493070008739_0001/
Kill Command = /usr/local/bin/hadoop-2.7.3/bin/hadoop job  -kill job_1493070008739_0001
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2017-04-24 17:43:08,636 Stage-3 map = 0%,  reduce = 0%
2017-04-24 17:43:17,097 Stage-3 map = 100%,  reduce = 0%
Ended Job = job_1493070008739_0001
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   HDFS Read: 46369769 HDFS Write: 25115070 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
```

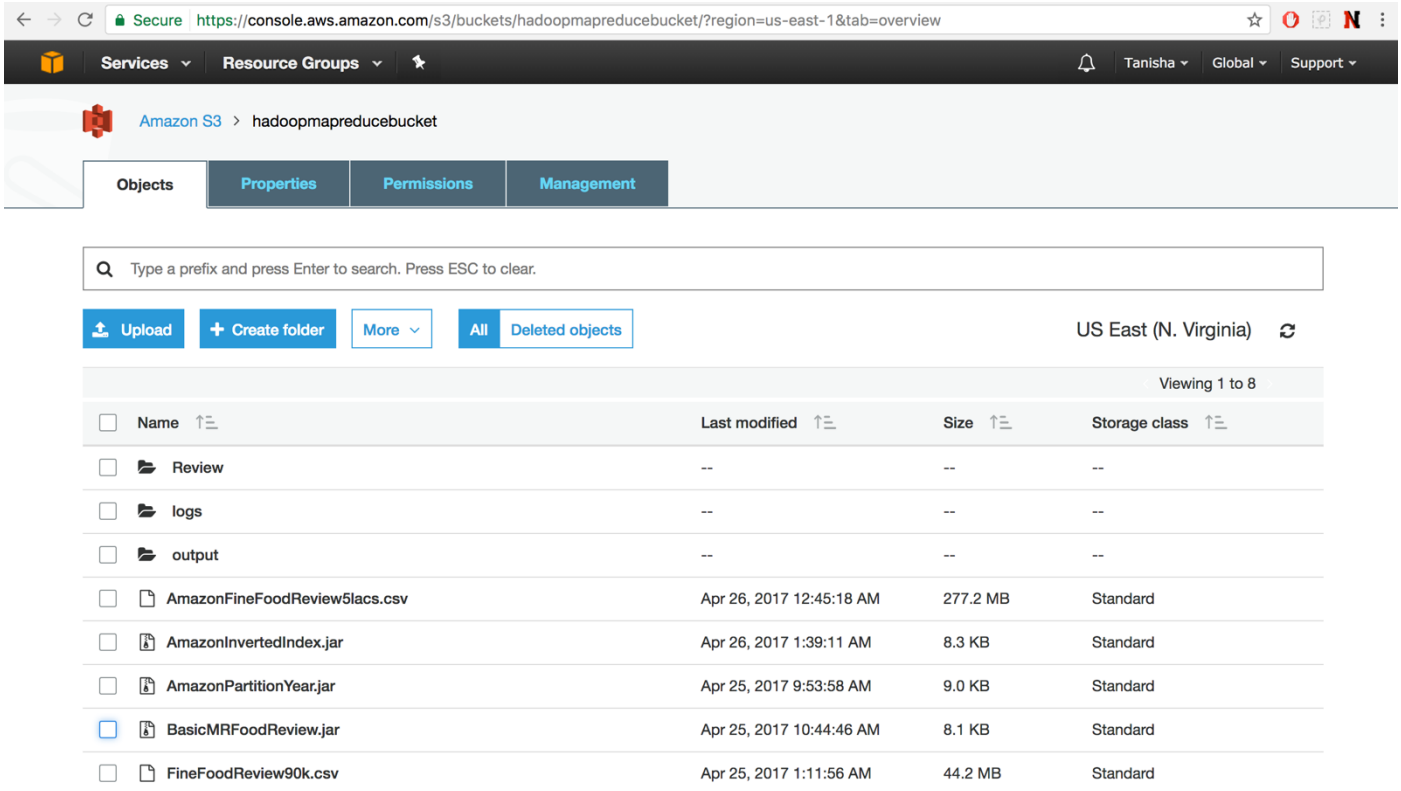
11) Amazon Elastic MapReduce:

Amazon Web Services provides a platform for executing MapReduce programs using its services. S3 () is used for storing the data into buckets and EMR platform is used for creating clusters and executing programming scripts. In this analysis, the execution of few java jar files and with input and output path specified on S3 is done. The cluster is created with EMR.

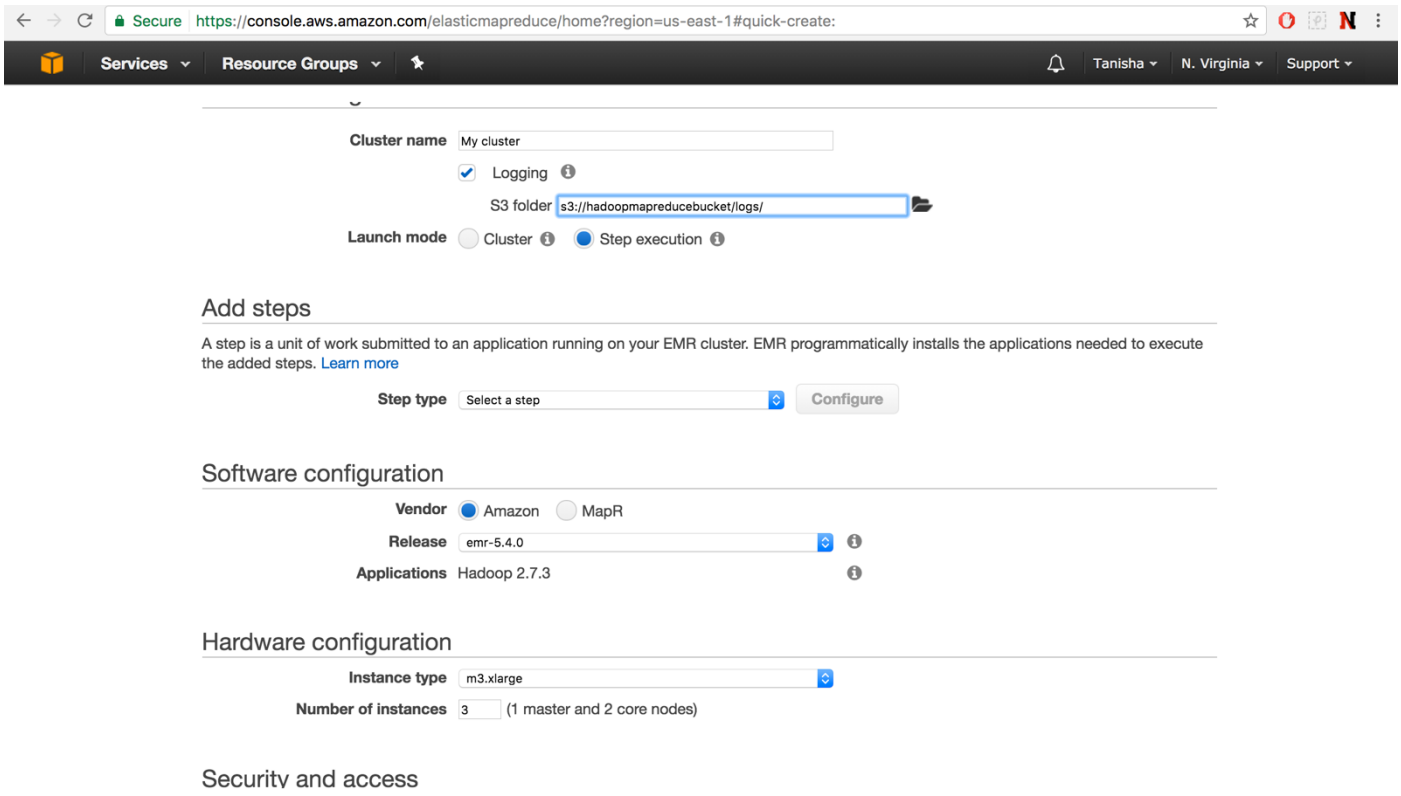
S3 bucket:



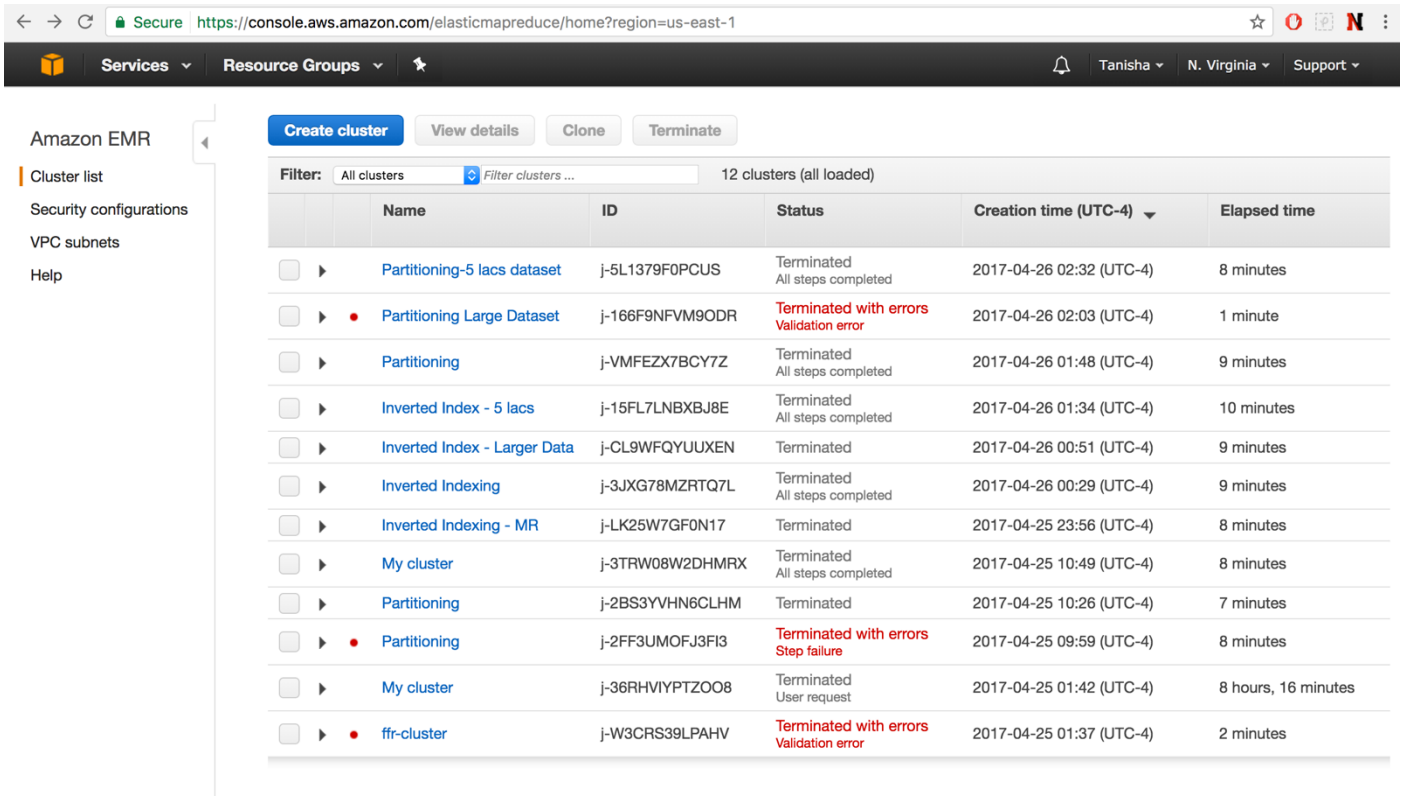
Files inside S3 bucket:



Creating Cluster with Configurations:



Running Clusters: (The output is right when it says Terminated All Steps completed)



Conclusions:

The MapReduce algorithms performed in the project are MapReduce java codes with implementation on Hadoop and filesystem HDFS. The programs are run on local machine with single node cluster and the output files are accessible on the localhost:50070.

Apache Mahout is a tool used for recommendations and is a simple java file. It can be run without Hadoop with file taken from the local machine as the input. It can display user-based recommendations.

Apache hive is a SQL-based query tool and is very fast when performing queries such as creating databases, creating tables and inserting bulk loads.

AWS EMR(Elastic Map Reduce) is very useful for storing data and creating clusters with no additional installations. This is very useful for handling large datasets.

Source Codes:

Program 1 – Average-Chaining-Sorting

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazonaveragechainingsorting;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonAverageChainingSorting {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException{
        Configuration conf1 = new Configuration();
        Job job1 = Job.getInstance(conf1,"Amazon Average");
        job1.setJarByClass(AmazonAverageChainingSorting.class);
        job1.setMapperClass(Map1.class);
        job1.setMapOutputKeyClass(Text.class);
        job1.setMapOutputValueClass(FloatWritable.class);

        job1.setReducerClass(Reduce1.class);
        job1.setOutputKeyClass(Text.class);
        job1.setOutputValueClass(FloatWritable.class);

        FileInputFormat.addInputPath(job1, new Path(args[0]));
        FileOutputFormat.setOutputPath(job1, new Path(args[1]));
        boolean complete = job1.waitForCompletion(true);
```

```
        Configuration conf2 = new Configuration();
        Job job2 = Job.getInstance(conf2,"Chaining Sorting");

        if(complete){
        job2.setJarByClass(AmazonAverageChainingSorting.class);
        job2.setMapperClass(Map2.class);
        job2.setMapOutputKeyClass(FloatWritable.class);
        job2.setMapOutputValueClass(Text.class);

        job2.setReducerClass(Reduce2.class);
        job2.setOutputKeyClass(Text.class);
        job2.setOutputValueClass(FloatWritable.class);

        FileInputFormat.addInputPath(job2, new Path(args[1]));
        FileOutputFormat.setOutputPath(job2, new Path(args[2]));

        System.exit(job2.waitForCompletion(true) ? 0:1);

        }
    }

    public static class Map1 extends Mapper<LongWritable, Text, Text, FloatWritable>{

        private Text text = new Text();
        private FloatWritable score = new FloatWritable();

        protected void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException{

            if(key.get()==0){
            return;
            }

            else{

            String[] line = value.toString().split(",");
            String productId = line[1].trim();
            float ratingVal = Float.valueOf(line[6].trim());

            text.set(productId);
            score.set(ratingVal);

            context.write(text, score);

            }
        }
    }

      public static class Reduce1 extends Reducer<Text, FloatWritable, Text, FloatWritable>{

        private FloatWritable result = new FloatWritable();
```

```
    @Override
    protected void reduce(Text key, Iterable<FloatWritable> values, Context context) throws
IOException, InterruptedException{

        float sum = 0;
        int count = 0;
        for(FloatWritable val:values){
            sum+=val.get();
            count = count+1;
        }

        float average = sum/count;

        result.set(average);
        context.write(key,result);
    }

  }

  public static class Map2 extends Mapper<LongWritable, Text, FloatWritable, Text>{

    public void map(LongWritable key, Text value, Context context){

        String[] row =value.toString().split("\\t");
        Text movieId = new Text(row[0]);
        float movieRatings = Float.valueOf(row[1].trim());

        try{
            FloatWritable count = new FloatWritable(movieRatings);
            context.write(count, movieId);
        }

        catch(Exception e){

        }
    }
  }


  public static class Reduce2 extends Reducer<FloatWritable, Text, Text, FloatWritable>{

    public void reduce(FloatWritable key, Iterable<Text> value, Context context) throws
IOException, InterruptedException{

    for(Text val : value){

    context.write(val,key);
   }
  }
 }
```

}

Program 2 - Inverted Index(Numerical Summarization Pattern)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazoninvertedindex;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonInvertedIndex {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args)throws IOException, InterruptedException,
ClassNotFoundException {
        // TODO code application logic here
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf,"Inverted Index");
        job.setJarByClass(AmazonInvertedIndex.class);
        job.setMapperClass(InvertedMapper.class);
        job.setReducerClass(InvertedReducer.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);
```

```
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }

    public static class InvertedMapper extends Mapper<LongWritable, Text, Text, Text>{

        private Text product = new Text();
        private Text userId = new Text();

        public void map(LongWritable key, Text values, Context context){

            if(key.get()==0){
                return;
            }

            try{
            String[] tokens = values.toString().split(",");

            userId.set(tokens[2]);
            product.set(tokens[1]);

            context.write(product, userId);
            }
            catch(IOException | InterruptedException ex){
                System.out.println("Error in Mapper" + ex.getMessage());
            }
        }
    }

    public static class InvertedReducer extends Reducer<Text,Text,Text,Text>{

        private Text result = new Text();

        @Override
        public void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException{

        StringBuilder sb=new StringBuilder();
        boolean first = true;

        for(Text id:values){
        if(first){
            first = false;
        }
        else{
            sb.append("    ");

        }
        sb.append(id.toString());
    }
```

```
        result.set(sb.toString());
        context.write(key, result);
    }
  }
}
```

Program 3 - Amazon Counting with Counters (Numerical Summarization Pattern)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazondistinctcounter;

import java.io.IOException;
import java.util.Arrays;
import java.util.HashSet;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonDistinctCounter {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Counting Product Ratings With Counters");
        job.setJarByClass(AmazonDistinctCounter.class);
        job.setMapperClass(CounterMapper.class);
```

```
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        Path out = new Path(args[1]);
        FileSystem.get(conf).delete(out, true);

        int code = job.waitForCompletion(true) ? 0 : 1;

       if(code==0){
           for(Counter counter : job.getCounters().getGroup(CounterMapper.ratingCounterGroup)){
              {
                  System.out.println("Rating "+counter.getDisplayName()+" is given to
"+counter.getValue()+" products.");
              }
           }
        }
        System.exit(code);
    }

      public static class CounterMapper extends Mapper<LongWritable, Text, Text, Text>{

        public static final String ratingCounterGroup = "Rating";
        public static final String unknownCounter = "Unknown";
        public static final String nullCounter = "Null";

        private String[] ratingsArray = new String[] {"1", "2", "3", "4", "5"};

        private HashSet<String> ratings = new HashSet<String>(Arrays.asList(ratingsArray));

        public void map (LongWritable key, Text value, Context context) throws IOException,
InterruptedException{

           if(key.get()==0){
              return;
           }

           String[] token = value.toString().split(",");
           String stateName = token[6].trim();

           if(stateName != null && !stateName.isEmpty()){
              String[] stateTokens = stateName.toUpperCase().split("\\s");
              boolean unknown = true;
              for(String state: stateTokens){
                 if(state.contains(state)){
                    context.getCounter(ratingCounterGroup, state).increment(1);
                    unknown = false;
                    break;
                 }
```

```
            }

        if(unknown){
            context.getCounter(ratingCounterGroup, unknownCounter).increment(1);
        }
        } else{
            context.getCounter(ratingCounterGroup, nullCounter).increment(1);
        }
      }
   }
}
```

Program 4 - Reduce Side Join (Anti Join & Inner Join) (Join Pattern)

AntiJoin:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazonantijoin;

import java.io.IOException;
import java.util.ArrayList;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.MultipleInputs;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonAntiJoin {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {
```

```
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Anti Join");
        job.setJarByClass(AmazonAntiJoin.class);

        MultipleInputs.addInputPath(job, new Path(args[0]), TextInputFormat.class,
AntiJoinMapper1.class);
        MultipleInputs.addInputPath(job, new Path(args[1]), TextInputFormat.class,
AntiJoinMapper2.class);
        job.getConfiguration().set("join.type", "antijoin");

        job.setReducerClass(UserRatingIdReducer.class);

        job.setOutputFormatClass(TextOutputFormat.class);
        TextOutputFormat.setOutputPath(job, new Path(args[2]));

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        System.exit(job.waitForCompletion(true)? 0: 2);
    }

     public static class AntiJoinMapper1 extends Mapper<LongWritable, Text, Text, Text>{

        private Text outkey = new Text();
        private Text outvalue = new Text();

        public void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException{

            if(key.get()==0){
                return;
            }

            //UserIds in Amazon Fine Food Review
            String[] separatedInput = value.toString().split(",");
            String userId = separatedInput[2].trim();

            if(userId== null) {
                return;
            }

            outkey.set(userId);
            // Flag this record for the reducer and then output
            outvalue.set("A" + value.toString());
            context.write(outkey, outvalue);
        }
    }

        public static class AntiJoinMapper2 extends Mapper<LongWritable, Text, Text, Text> {
```

```
        private Text outkey = new Text();
        private Text outvalue = new Text();

        public void map(LongWritable key, Text value, Mapper.Context context) throws
IOException, InterruptedException {

        if(key.get()==0){
            return;
        }
        //UserId in User Profiles
        String[] separatedInput = value.toString().split(",");
        String ratingId = separatedInput[1];

        if (ratingId == null) {
            return;
        }
        // The foreign join key is the user ID
        outkey.set(ratingId);
        // Flag this record for the reducer and then output
        outvalue.set("B" + value.toString());
        context.write(outkey, outvalue);
    }
  }

    public static class UserRatingIdReducer extends Reducer<Text, Text, Text, Text>{

    private static final Text EMPTY_TEXT = new Text("");
    private Text tmp = new Text();
    private ArrayList<Text> listA = new ArrayList<Text>();
    private ArrayList<Text> listB = new ArrayList<Text>();

    private String joinType = null;

    public void setup(Reducer.Context context) {

        // Get the type of join from our configuration
        joinType = context.getConfiguration().get("join.type");
    }

     public void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException {

        //Clearing the lists
        listA.clear();
        listB.clear();

          while (values.iterator().hasNext()) {
            tmp = values.iterator().next();

            if (Character.toString((char) tmp.charAt(0)).equals("A")) {
```

```
                listA.add(new Text(tmp.toString().substring(1)));
            }
            if (Character.toString((char) tmp.charAt(0)).equals("B")) {

                listB.add(new Text(tmp.toString().substring(1)));
            }

        }
        executeJoinLogic(context);
    }

    private void executeJoinLogic(Context context) throws IOException, InterruptedException{

      //This is removal of common attributes of 2 files from a whole joining of 2 files
      //anti join = full outer join - inner join;
       if (joinType.equalsIgnoreCase("antijoin")) {
            if (listA.isEmpty() ^ listB.isEmpty()) {

                for (Text A : listA) {
                    context.write(A, EMPTY_TEXT);
                }
                for (Text B : listB) {
                    context.write(EMPTY_TEXT, B);
                }
            }
        }


    }

    }

}
```

Inner Join:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazoninnerjoin;

import java.io.IOException;
import java.util.ArrayList;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.MultipleInputs;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonInnerJoin {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Inner Join");
        job.setJarByClass(AmazonInnerJoin.class);

        MultipleInputs.addInputPath(job, new Path(args[0]), TextInputFormat.class,
InnerJoinMapper1.class);
        MultipleInputs.addInputPath(job, new Path(args[1]), TextInputFormat.class,
InnerJoinMapper2.class);
        job.getConfiguration().set("join.type", "inner");
        //job.setNumReduceTasks(0);
        job.setReducerClass(UserRatingIdReducer.class);

        job.setOutputFormatClass(TextOutputFormat.class);
        TextOutputFormat.setOutputPath(job, new Path(args[2]));

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        System.exit(job.waitForCompletion(true)? 0: 2);
    }

     public static class InnerJoinMapper1 extends Mapper<LongWritable, Text, Text, Text>{

        private Text outkey = new Text();
        private Text outvalue = new Text();

        public void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException{

            if(key.get()==0){
                return;
            }
```

```
        //UserIds in Amazon Fine Food Review
        String[] separatedInput = value.toString().split(",");
        String userId = separatedInput[2].trim();

        if(userId== null) {
            return;
        }

        outkey.set(userId);
        // Flag this record for the reducer and then output
        outvalue.set("A" + value.toString());
        context.write(outkey, outvalue);
    }
}

    public static class InnerJoinMapper2 extends Mapper<LongWritable, Text, Text, Text> {

        private Text outkey = new Text();
        private Text outvalue = new Text();

        public void map(LongWritable key, Text value, Mapper.Context context) throws
IOException, InterruptedException {

        if(key.get()==0){
            return;
        }
        //UserId in User Profiles
        String[] separatedInput = value.toString().split(",");

        String ratingId = separatedInput[1];
        if (ratingId == null) {
            return;
        }
        // The foreign join key is the user ID
        outkey.set(ratingId);
        // Flag this record for the reducer and then output
        outvalue.set("B" + value.toString());
        context.write(outkey, outvalue);
    }
}

    public static class UserRatingIdReducer extends Reducer<Text, Text, Text, Text>{

    private static final Text EMPTY_TEXT = new Text("");
    private Text tmp = new Text();
    private ArrayList<Text> listA = new ArrayList<Text>();
    private ArrayList<Text> listB = new ArrayList<Text>();

    private String joinType = null;
```

```
    public void setup(Reducer.Context context) {

        // Get the type of join from our configuration
        joinType = context.getConfiguration().get("join.type");
    }

     public void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException {

        //Clearing the lists
        listA.clear();
        listB.clear();

          while (values.iterator().hasNext()) {
            tmp = values.iterator().next();

            if (Character.toString((char) tmp.charAt(0)).equals("A")) {

                listA.add(new Text(tmp.toString().substring(1)));
            }
            if (Character.toString((char) tmp.charAt(0)).equals("B")) {

                listB.add(new Text(tmp.toString().substring(1)));
            }

        }
        executeJoinLogic(context);
    }

    private void executeJoinLogic(Context context) throws IOException, InterruptedException{


        if (joinType.equalsIgnoreCase("inner")) {
            if (!listA.isEmpty() && !listB.isEmpty()) {
                for (Text A : listA) {
                    for (Text B : listB) {
                        context.write(A, B);
                    }
                }
            }
        }


    }

    }

}
```

Program 5 - Partitioning (Organization Pattern)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazonpartitionyear;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Partitioner;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonPartitionYear {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf,"Partitioning");

        job.setJarByClass(AmazonPartitionYear.class);

        job.setMapperClass(YearPartitionMapper.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.class);

        //Custom Partitioner::
        job.setPartitionerClass(YearPartitionPartitioner.class);

        job.setReducerClass(YearPartitionReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
```

```
        job.setNumReduceTasks(14);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);

    }

        public static class YearPartitionMapper extends Mapper<LongWritable, Text, Text, Text>{

        private Text inputRec = new Text();
        private Text year = new Text();

        protected void map(LongWritable key, Text value, Mapper.Context context) throws
IOException, InterruptedException{

                if(key.get()==0){
                return;
                }

                String[] line = value.toString().split(",");
                String[] yearPart = line[3].split("/");
                String yearVal = yearPart[2].trim();

                year.set(yearVal);
                inputRec.set(value);

                context.write(year, inputRec);
        }
    }

        public static class YearPartitionPartitioner extends Partitioner<Text, Text> {

        Text yrval = new Text();
        @Override
        public int getPartition(Text key, Text value, int numPartitions){
            int n=1;
            if(numPartitions==0){
                return 0;
            }
            else if(key.equals(("99"))){
                return n % numPartitions;
            }
            else if(key.equals(new Text("00"))){
                return 2 % numPartitions;
            }
             else if(key.equals(new Text("01"))){
                return 3 % numPartitions ;
            }
             else if(key.equals(new Text("02"))){
```

```
            return 4 % numPartitions;
        }
         else if(key.equals(new Text("03"))){
            return 5 % numPartitions;
        }
         else if(key.equals(new Text("04"))){
            return 6 % numPartitions;
        }
         else if(key.equals(new Text("05"))){
            return 7 % numPartitions;
        }
         else if(key.equals(new Text("06"))){
            return 8 % numPartitions;
        }
         else if(key.equals(new Text("07"))){
            return 9 % numPartitions;
        }
         else if(key.equals(new Text("08"))){
            return 10 % numPartitions;
        }
         else if (key.equals(new Text("09"))){
            return 11 % numPartitions;
        }
         else if (key.equals(new Text("10"))){
            return 12 % numPartitions;
        }
         else if (key.equals(new Text("11"))){
            return 13 % numPartitions;
        }
        else
         {
             return 14 % numPartitions;
         }
      }
   }



      public static class YearPartitionReducer extends Reducer<Text, Text, Text, NullWritable> {

      protected void reduce(Text key, Iterable<Text> values, Reducer.Context context) throws
IOException, InterruptedException{
        for(Text t: values){

            context.write(t, NullWritable.get());
        }
      }
   }
}
```

Program 6 – Amazon Binning Rating

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazonbinningrating;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonBinningRating {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args)throws IOException, InterruptedException,
ClassNotFoundException {
        // TODO code application logic here
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Binning Hour");
        job.setJarByClass(AmazonBinningRating.class);

        //Setting Mapper Class and the output key and value
        job.setMapperClass(BinningMapper.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(NullWritable.class);

        //No combiner, partitioner or reducer is used in this pattern!
        job.setNumReduceTasks(0);

        TextInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
```

```
        MultipleOutputs.addNamedOutput(job, "bins", TextOutputFormat.class, Text.class,
NullWritable.class);
        MultipleOutputs.setCountersEnabled(job, true);

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }

    public static class BinningMapper extends Mapper<LongWritable, Text, Text, NullWritable>{

    private MultipleOutputs<Text, NullWritable> mos = null;

    protected void setup(Context context){
        mos = new MultipleOutputs(context);

    }

    protected void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException{

            //To remove file header
            if(key.get()==0){
                return;
            }

            String[] token = value.toString().split(",");

                String rating = token[6].trim();

                if(rating.equals("1")){
                    mos.write("bins", value, NullWritable.get(), "Rating 1");
                }
                 if(rating.equals("2")){
                    mos.write("bins", value, NullWritable.get(), "Rating 2");
                }
                  if(rating.equals("3")){
                    mos.write("bins", value, NullWritable.get(), "Rating 3");
                }
                   if(rating.equals("4")){
                    mos.write("bins", value, NullWritable.get(), "Rating 4");
                }
                   if(rating.equals("5")){
                    mos.write("bins", value, NullWritable.get(), "Rating 5");
                }

            }
                protected void cleanup(Context context) throws IOException, InterruptedException{
                    mos.close();
                }

    }
```

}

Program 7 - Top 100 Products (Filtering Pattern)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazontopproducts;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.WritableComparable;
import org.apache.hadoop.io.WritableComparator;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonTopProducts {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {

        Configuration conf1 = new Configuration();
        Job job1 = Job.getInstance(conf1,"Top 100 Products");
        job1.setJarByClass(AmazonTopProducts.class);
        job1.setMapperClass(Map1.class);
        job1.setMapOutputKeyClass(Text.class);
        job1.setMapOutputValueClass(FloatWritable.class);

        job1.setReducerClass(Reduce1.class);
        job1.setOutputKeyClass(Text.class);
        job1.setOutputValueClass(FloatWritable.class);
```

```
        FileInputFormat.addInputPath(job1, new Path(args[0]));
        FileOutputFormat.setOutputPath(job1, new Path(args[1]));

        boolean complete = job1.waitForCompletion(true);

        Configuration conf2 = new Configuration();


        if(complete){
        Job job2 = Job.getInstance(conf2,"Top 100 Products");
        job2.setJarByClass(AmazonTopProducts.class);
        job2.setMapperClass(Map2.class);
        job2.setMapOutputKeyClass(FloatWritable.class);
        job2.setMapOutputValueClass(Text.class);

        job2.setSortComparatorClass(GroupComparator.class);

        job2.setReducerClass(Reduce2.class);
        job2.setOutputKeyClass(FloatWritable.class);
        job2.setOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job2, new Path(args[1]));
        FileOutputFormat.setOutputPath(job2, new Path(args[2]));

        System.exit(job2.waitForCompletion(true) ? 0 : 1);
        }
    }

    public static class Map1 extends Mapper<LongWritable, Text, Text, FloatWritable>{

        private Text text = new Text();
        private FloatWritable score = new FloatWritable();

        protected void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException{

            if(key.get()==0){
            return;
            }

            else{

            String[] line = value.toString().split(",");
            String productId = line[1].trim();
            float ratingVal = Float.valueOf(line[6].trim());

            text.set(productId);
            score.set(ratingVal);

            context.write(text, score);
```

```
        }
      }
   }

    public static class Reduce1 extends Reducer<Text, FloatWritable, Text, FloatWritable>{

      private FloatWritable result = new FloatWritable();

      @Override
      protected void reduce(Text key, Iterable<FloatWritable> values, Context context) throws
IOException, InterruptedException{

         float sum = 0;
         int count = 0;
         for(FloatWritable val:values){
            sum+=val.get();
            count = count+1;
         }

         float average = sum/count;

         result.set(average);
         context.write(key,result);
      }
   }

   public static class Map2 extends Mapper<LongWritable, Text, FloatWritable, Text>
   {
      public void map(LongWritable key, Text value,Context context){

         String[] row = value.toString().split("\\t");
         String productId = row[0].trim();
         float prodRating = Float.parseFloat(row[1].trim());

         try{
            Text id = new Text(productId);
            FloatWritable prodRate = new FloatWritable(prodRating);
            context.write(prodRate,id);

         }catch(Exception e){

         }
      }
   }

   public static class GroupComparator extends WritableComparator {

         protected GroupComparator()
         {
            super(FloatWritable.class,true);
```

```
        }

        public int compare(WritableComparable w1, WritableComparable w2)
           {
              FloatWritable cw1 = (FloatWritable) w1;
              FloatWritable cw2 = (FloatWritable) w2;
              int result = cw1.get() < cw2.get() ? 1 : cw1.get() == cw2.get() ? 0 : -1;
              return result;
           }
    }


    public static class Reduce2 extends Reducer<FloatWritable, Text, FloatWritable, Text>{

       int count=0;
       public void reduce(FloatWritable key, Iterable<Text> value,Context context)
             throws IOException, InterruptedException{

          for(Text val: value){

          if(count<100)
          {
             context.write(key,val);
          }
             count++;
       }
    }


    }

}
```

Program 9 - Apache Mahout

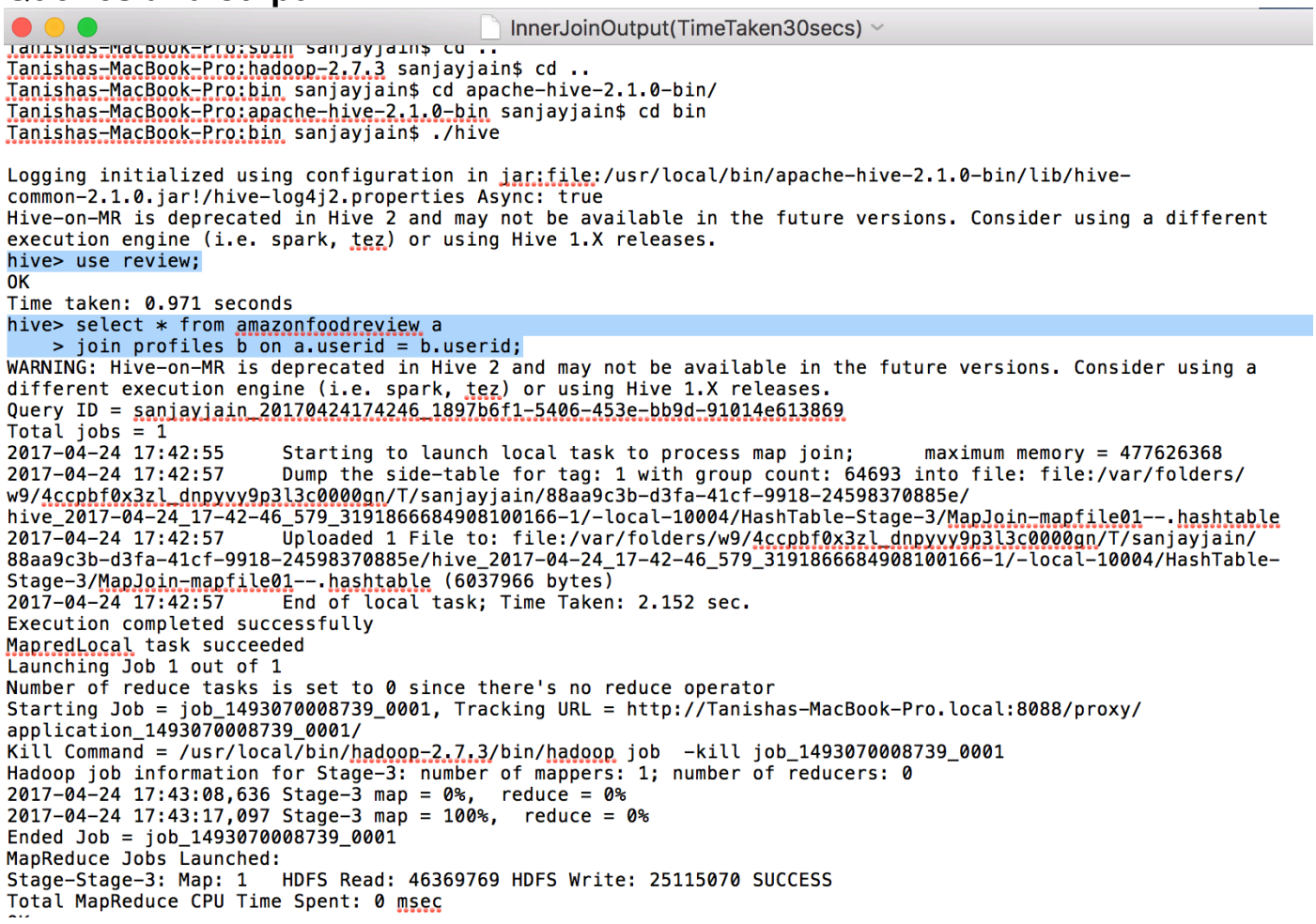
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amazonmahoutrecommendation;

import java.io.File;
import java.io.IOException;
import java.util.List;
import org.apache.mahout.cf.taste.common.TasteException;
import org.apache.mahout.cf.taste.impl.common.LongPrimitiveIterator;
```

```
import org.apache.mahout.cf.taste.impl.model.file.FileDataModel;
import org.apache.mahout.cf.taste.impl.neighborhood.ThresholdUserNeighborhood;
import org.apache.mahout.cf.taste.impl.recommender.GenericUserBasedRecommender;
import org.apache.mahout.cf.taste.impl.similarity.PearsonCorrelationSimilarity;
import org.apache.mahout.cf.taste.model.DataModel;
import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;
import org.apache.mahout.cf.taste.recommender.RecommendedItem;
import org.apache.mahout.cf.taste.recommender.Recommender;
import org.apache.mahout.cf.taste.similarity.UserSimilarity;

/**
 *
 * @author Tanisha_Jain
 */
public class AmazonMahoutRecommendation {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException, TasteException{

        File userPreferencesFile = new
File("/Users/sanjayjain/Desktop/ADBMS_PROJECT/Trials/InputForMahout.csv");
        DataModel dataModel = new FileDataModel(userPreferencesFile);

        UserSimilarity userSimilarity = new PearsonCorrelationSimilarity(dataModel);
        UserNeighborhood userNeighborhood = new ThresholdUserNeighborhood(0.1, userSimilarity,
dataModel);

        // Create a generic user based recommender with the dataModel, the userNeighborhood and
the userSimilarity
        Recommender genericRecommender =  new GenericUserBasedRecommender(dataModel,
userNeighborhood, userSimilarity);

        for (LongPrimitiveIterator iterator = dataModel.getUserIDs(); iterator.hasNext();)
        {
            long userId = iterator.nextLong();

            //5 recommendations for each user
            List<RecommendedItem> itemRecommendations =
genericRecommender.recommend(userId, 5);

            System.out.format("User Id: %d%n", userId);

            if (itemRecommendations.isEmpty())
            {
                System.out.println("     There is no recommended item for this user.");
            }
            else
            {
                // Display the list of recommendations
```

```
            for (RecommendedItem recommendedItem : itemRecommendations)
            {
                System.out.format("     Recommened Item Id: %d. Strength of preference: %f%n",
recommendedItem.getItemID(), recommendedItem.getValue());
            }
        }
    }
  }
}
```

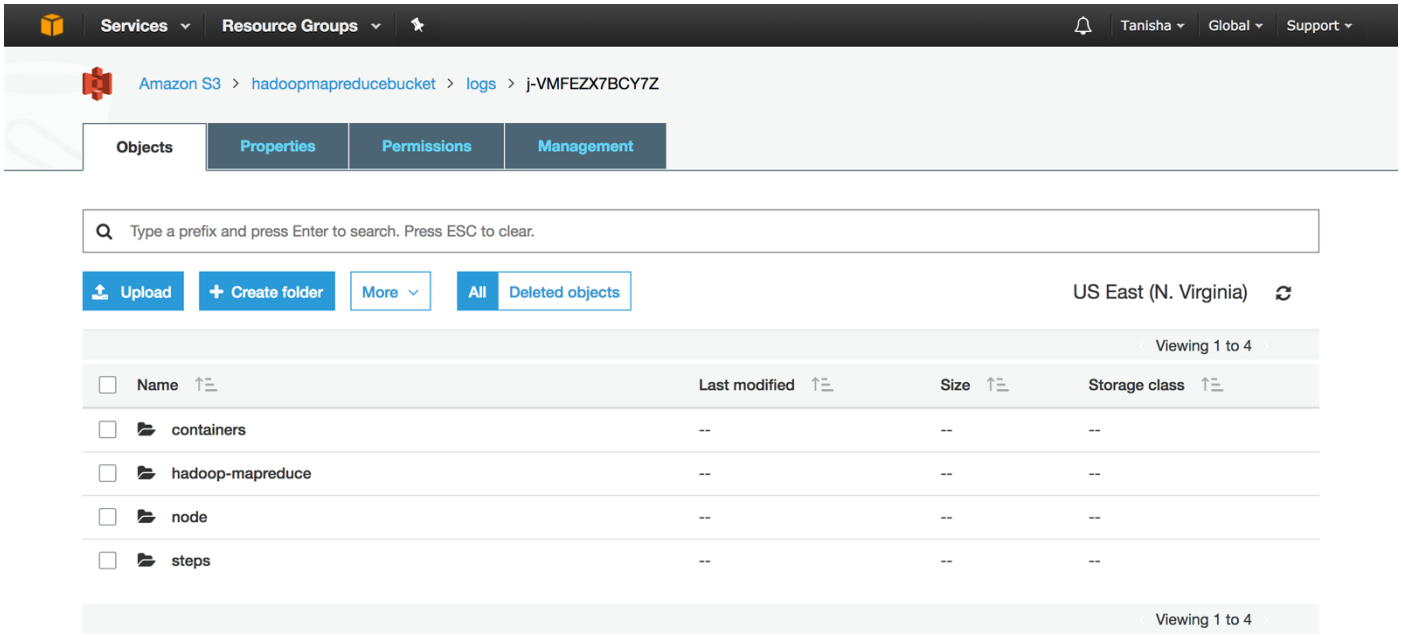
Program 10 – Apache Hive

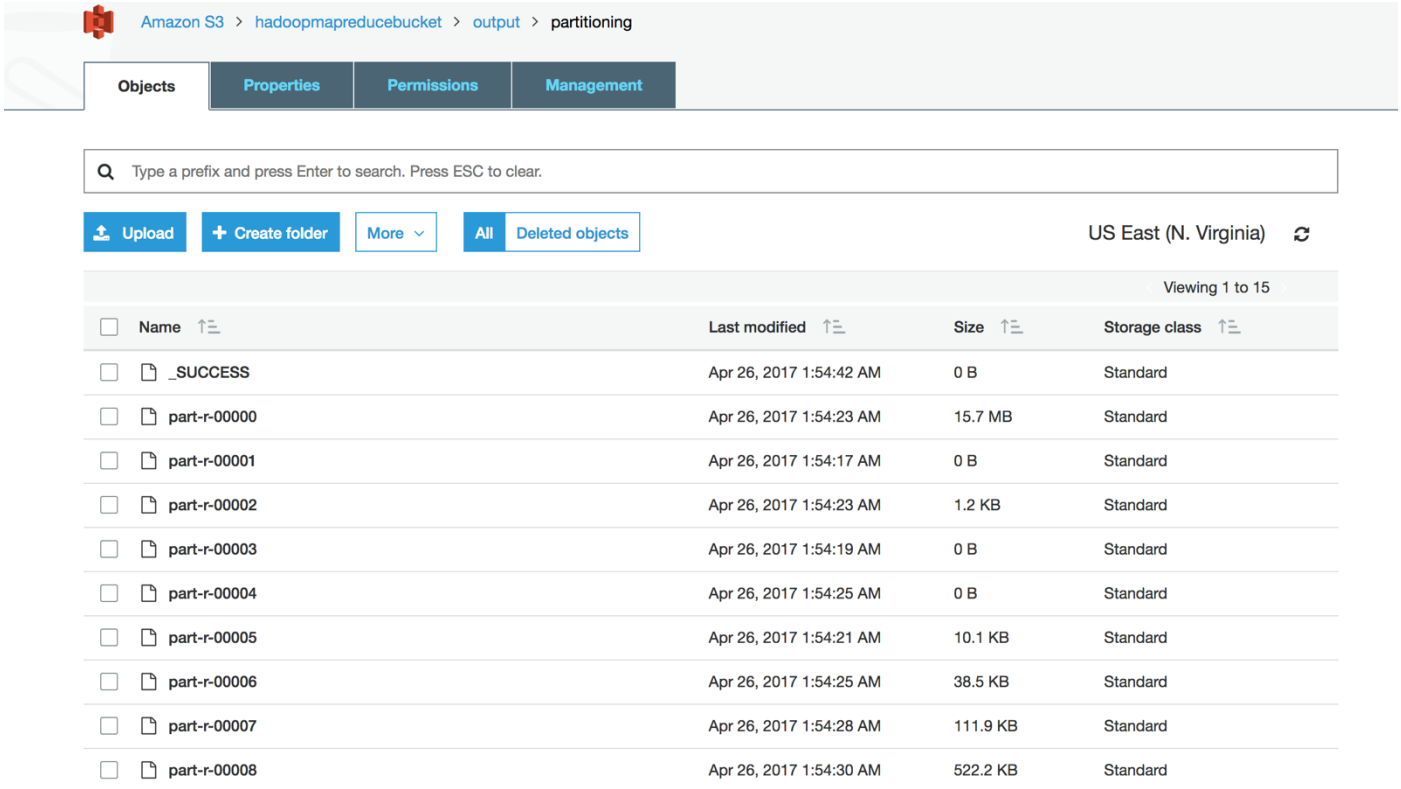
Queries and output:

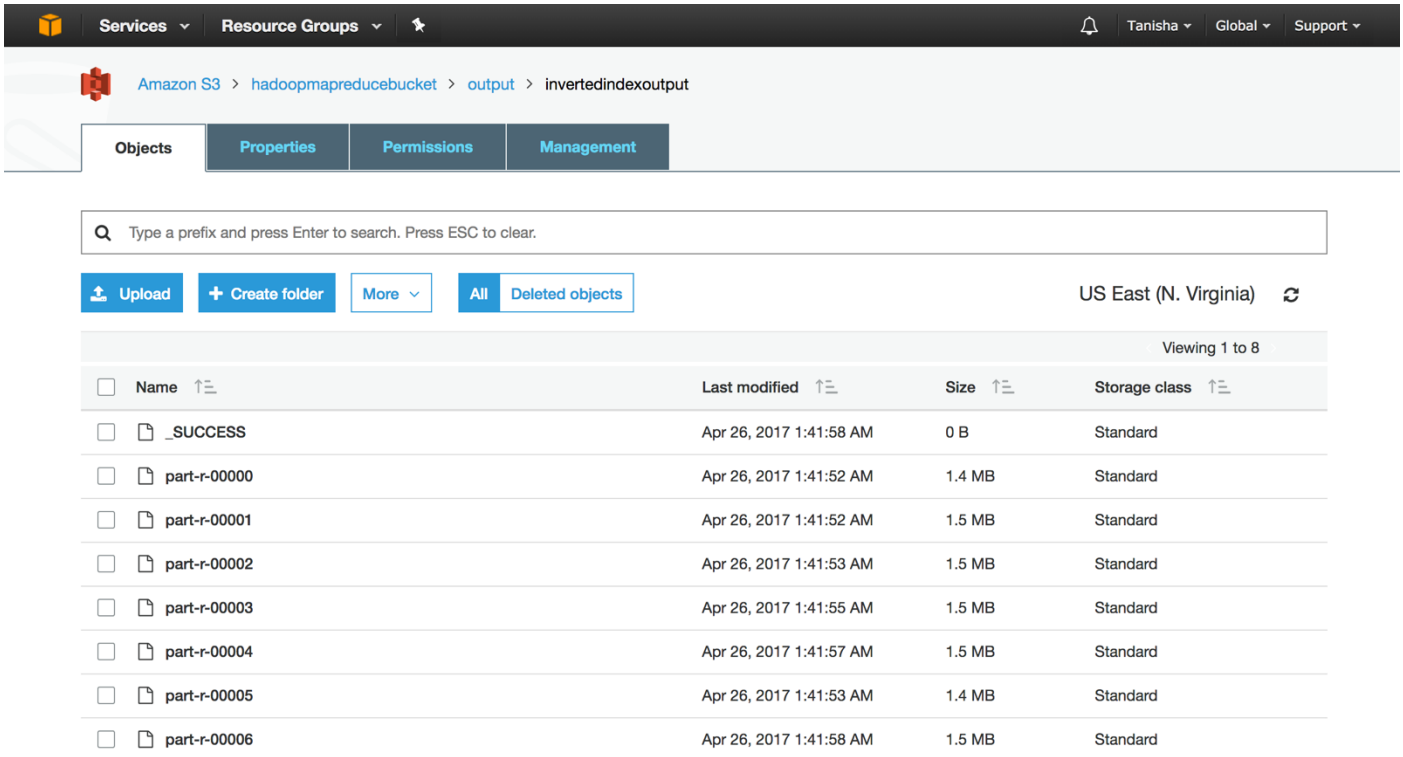


Program 11 – AWS Elastic Map Reduce

Program logs and output files screenshots:






Services
Resource Groups
Tanisha
Global
Support
Amazon S3 > hadoopmapreducebucket > output > invertedindexoutput
Objects
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
+ Create folder
More
All
Deleted objects
US East (N. Virginia)
Viewing 1 to 8
Name
Last modified
Size
Storage class
_SUCCESS Apr 26, 2017 1:41:58 AM 0 B Standard
part-r-00000 Apr 26, 2017 1:41:52 AM 1.4 MB Standard
part-r-00001 Apr 26, 2017 1:41:52 AM 1.5 MB Standard
part-r-00002 Apr 26, 2017 1:41:53 AM 1.5 MB Standard
part-r-00003 Apr 26, 2017 1:41:55 AM 1.5 MB Standard
part-r-00004 Apr 26, 2017 1:41:57 AM 1.5 MB Standard
part-r-00005 Apr 26, 2017 1:41:53 AM 1.4 MB Standard
part-r-00006 Apr 26, 2017 1:41:58 AM 1.5 MB Standard