EECS E6893 Homework 2

Zhuxi Cai zc2270

1. Recommendation:

(1) Choose any two datasets from Yahoo Labs Ratings and Classification Data

We choose the following two datasets from Yahoo Labs Ratings and Classification Data:

- ydata-ymusic-rating-study-v1_0-train.txt
- ydata-delicious-popular-urls-and-tags-v1_0.txt

(2) Try various recommendation algorithms provided by Mahout

First we try Mahout working with Eclipse and Maven using dataset ydata-ymusic-rating-study-v1_0-train.txt. We apply three recommendation algorithms: user-based recommendation, item-based recommendation and SVD recommendation. Then we evaluate the three models we built separately.

User-based Recommender Java Code:

```
public class User_based_recommender
{
    public static void main( String[] args ) throws IOException, TasteException
    {
        DataModel model = new FileDataModel(new File("data/ydata-ymusic-rating-study-v1_0-train.txt"));
        UserSimilarity similarity = new PearsonCorrelationSimilarity(model);
        UserNeighborhood neighborhood = new ThresholdUserNeighborhood(0.1, similarity, model);
        UserBasedRecommender recommender = new GenericUserBasedRecommender(model, neighborhood, similarity);
        int userID=2;
        int itemNum=3;
        List<RecommendedItem> recommendations = recommender.recommend(userID, itemNum);
        System.out.println("User-based Recommender");
        System.out.println("For user "+userID+", recommend "+itemNum+" items:");
        for (RecommendedItem recommendation : recommendations) {
          System.out.println(recommendation);
        }
    }
}
```

User-based Recommender Result:

```
User-based Recommender
For user 2, recommend 3 items:
RecommendedItem[item:514, value:4.619235]
RecommendedItem[item:266, value:4.6109624]
RecommendedItem[item:69, value:4.4430842]
```

User-based Recommender Evaluation:

```java
public class User_based_recommender_eva {

    public static void main(String[] args) throws IOException, TasteException
    {
        DataModel model = new FileDataModel(new File("data/ydata-ymusic-rating-study-v1_0-train.txt"));
        RecommenderEvaluator evaluator = new AverageAbsoluteDifferenceRecommenderEvaluator();
        RecommenderBuilder builder = new UserRecommenderBuilder();
        double result = evaluator.evaluate(builder, null, model, 0.9, 1.0);
        System.out.println("User-based Recommender Evaluation");
        System.out.println(result);
    }

}

class UserRecommenderBuilder implements RecommenderBuilder{

    public Recommender buildRecommender(DataModel dataModel) throws TasteException {
        UserSimilarity similarity = new PearsonCorrelationSimilarity(dataModel);
        UserNeighborhood neighborhood = new ThresholdUserNeighborhood(0.1, similarity, dataModel);
        return new GenericUserBasedRecommender(dataModel, neighborhood, similarity);
    }
}
```

```
User-based Recommender Evaluation
1.267003051378381
```

Item-based Recommender Java Code:

```java
public class Item_based_recommender
{
    public static void main( String[] args ) throws IOException, TasteException
    {
        DataModel model = new FileDataModel(new File("data/ydata-ymusic-rating-study-v1_0-train.txt"));
        ItemSimilarity similarity = new PearsonCorrelationSimilarity(model);
        ItemBasedRecommender recommender = new GenericItemBasedRecommender(model, similarity);
        int userID=2;
        int itemNum=3;
        List<RecommendedItem> recommendations = recommender.recommend(userID, itemNum);
        System.out.println("Item-based Recommender");
        System.out.println("For user "+userID+", recommend "+itemNum+" items:");
        for (RecommendedItem recommendation : recommendations) {
          System.out.println(recommendation);
        }
    }
}
```

Item-based Recommender Result:

```
Item-based Recommender
For user 2, recommend 3 items:
RecommendedItem[item:26, value:5.0]
RecommendedItem[item:13, value:5.0]
RecommendedItem[item:5, value:5.0]
```

Item-based Recommender Evaluation:

```java
public class Item_based_recommender_eva {

    public static void main(String[] args) throws IOException, TasteException
    {
        DataModel model = new FileDataModel(new File("data/ydata-ymusic-rating-study-v1_0-train.txt"));
        RecommenderEvaluator evaluator = new AverageAbsoluteDifferenceRecommenderEvaluator();
        RecommenderBuilder builder = new ItemRecommenderBuilder();
        double result = evaluator.evaluate(builder, null, model, 0.9, 1.0);
        System.out.println("Item-based Recommender Evaluation");
        System.out.println(result);
    }

}

class ItemRecommenderBuilder implements RecommenderBuilder{

    public Recommender buildRecommender(DataModel dataModel) throws TasteException {
        ItemSimilarity similarity = new PearsonCorrelationSimilarity(dataModel);
        return new GenericItemBasedRecommender(dataModel, similarity);
    }
}
```

```
Item-based Recommender Evaluation
0.9814732361074119
```

SVD Recommender Java Code:

```java
public class SVD_recommender
{
    public static void main( String[] args ) throws IOException, TasteException
    {
        DataModel model = new FileDataModel(new File("data/ydata-ymusic-rating-study-v1_0-train.txt"));
        SVDRecommender recommender = new SVDRecommender(model, new ALSWRFactorizer(model, 10, 0.05, 10));
        int userID=2;
        int itemNum=3;
        List<RecommendedItem> recommendations = recommender.recommend(userID, itemNum);
        System.out.println("SVD Recommender");
        System.out.println("For user "+userID+", recommend "+itemNum+" items:");
        for (RecommendedItem recommendation : recommendations) {
          System.out.println(recommendation);
        }
    }
}
```

SVD Recommender Result:

```
SVD Recommender
For user 2, recommend 3 items:
RecommendedItem[item:773, value:6.2266994]
RecommendedItem[item:763, value:5.7881827]
RecommendedItem[item:629, value:5.784256]
```

SVD Recommender Evaluation:

```java
public class SVD_recommender_eva {

    public static void main(String[] args) throws IOException, TasteException
    {
        DataModel model = new FileDataModel(new File("data/ydata-ymusic-rating-study-v1_0-train.txt"));
        RecommenderEvaluator evaluator = new AverageAbsoluteDifferenceRecommenderEvaluator();
        RecommenderBuilder builder = new SVDRecommenderBuilder();
        double result = evaluator.evaluate(builder, null, model, 0.9, 1.0);
        System.out.println("SVD Recommender Evaluation");
        System.out.println(result);
    }

}

class SVDRecommenderBuilder implements RecommenderBuilder{

    public Recommender buildRecommender(DataModel dataModel) throws TasteException {
        return new SVDRecommender(dataModel, new ALSWRFactorizer(dataModel, 10, 0.05, 10));
    }
}
```

```
SVD Recommender Evaluation
0.9719408046658731
```

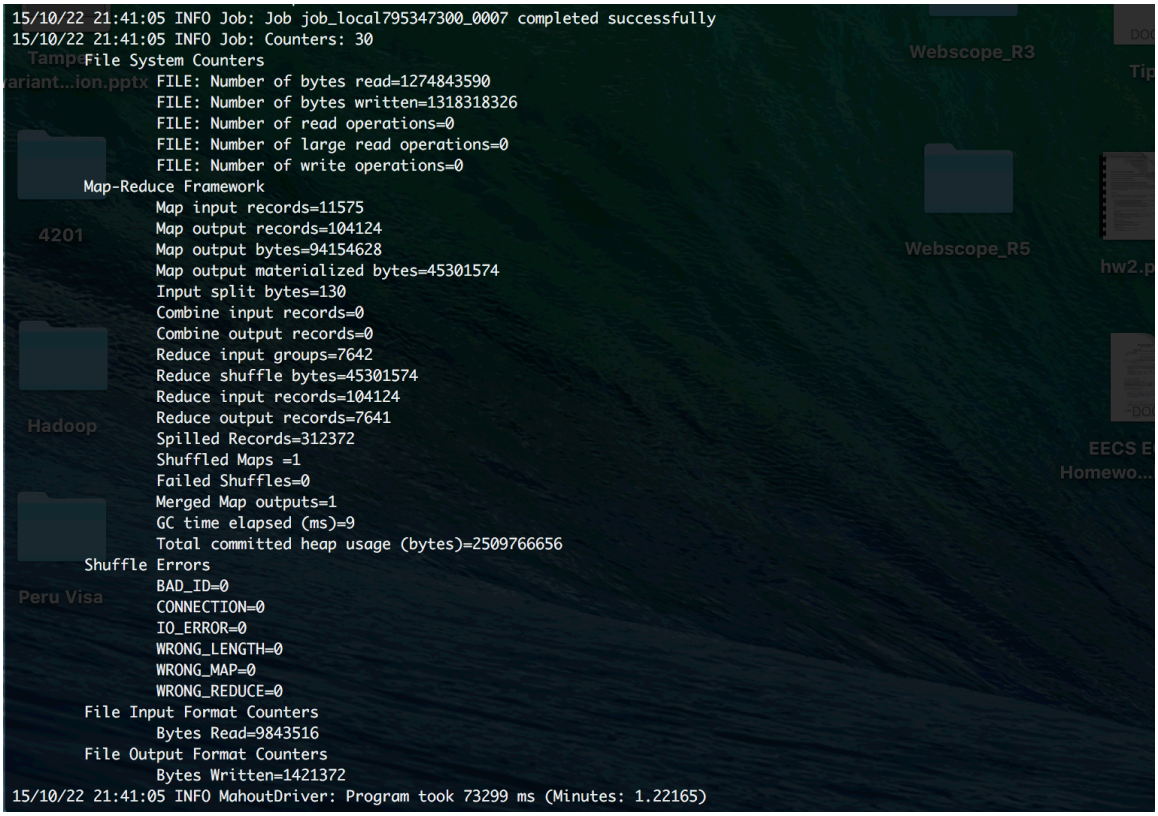
To compare these three models, we separate the whole dataset into training and testing dataset, each of which is 90% and 10%. Then we calculate the average absolute difference for three recommenders and find that SVD recommender performs best here.

Second we install Mahout in bash and try Mahout locally using dataset ydata-delicious-popular-urls-and-tags-v1_0.txt. We choose item-base recommendation algorithm and get the following results:

Item-based Recommender Code:

```
Zhuxis-MacBook-Pro:mahout-trunk Jovial$ bin/mahout recommenditembased -s SIMILARITY_LOGLIKELIHOOD -i /Users/Jovial/Desktop/Web
scope_R5/ydata-delicious-popular-urls-and-tags-v1_0.txt -o /Users/Jovial/Desktop/HW2/output --numRecommendations 10
```

Item-based Result:



```
15/10/22 21:41:05 INFO Job: Job job_local795347300_0007 completed successfully
15/10/22 21:41:05 INFO Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=1274843590
                FILE: Number of bytes written=1318318326
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=11575
                Map output records=104124
                Map output bytes=94154628
                Map output materialized bytes=45301574
                Input split bytes=130
                Combine input records=0
                Combine output records=0
                Reduce input groups=7642
                Reduce shuffle bytes=45301574
                Reduce input records=104124
                Reduce output records=7641
                Spilled Records=312372
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=9
                Total committed heap usage (bytes)=2509766656
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=9843516
        File Output Format Counters
                Bytes Written=1421372
15/10/22 21:41:05 INFO MahoutDriver: Program took 73299 ms (Minutes: 1.22165)
```

```
1
[1800159233:13.0,1800353817:13.0,1800202853:13.0,1800024141:13.0,1800122300:13.0,1807858489:13.0,18040
90611:13.0,1800019431:13.0,1800353749:13.0,1800026398:13.0]
2
[1800112450:11.001758,1800071079:10.370626,1804738128:10.332838,1800212379:10.018739,1808405650:10.009
023,1800242003:9.877988,1803453994:9.799826,1800128578:9.77842,1802771203:9.54708,1800258211:9.54532]
3
[1802953401:13.0,1800185256:13.0,1800090318:13.0,1800444945:13.0,1800021580:13.0,1804090611:13.0,18000
82735:13.0,1804857429:13.0,1804383575:13.0,1800019316:13.0]
4
[1800019304:10.5210285,1800379216:10.506128,1800022996:10.503612,1800174173:10.501894,1800019665:10.49
9486,1800185256:10.498645,1800060404:10.497791,1807733433:10.496107,1800026332:10.490099,1800128578:10
.0433445]
5
[1807993019:13.0,1800168311:13.0,1800058561:13.0,1804090611:13.0,1808404659:13.0,1808416901:12.758992,
1808465610:12.746555,1800223031:12.67576,1804861086:12.6658535,1808403030:12.661338]
6
[1808412549:13.0,1808481189:13.0,1800112611:13.0,1800361193:12.65157,1808405708:12.530771,1808437167:1
2.523912,1808443070:12.514433,1808405886:12.511748,1807993019:12.505257,1800018659:12.503544]
7
[1800444945:13.0,1802953401:13.0,1800022145:13.0,1800026205:13.0,1808486011:13.0,1800021539:13.0,18002
01616:13.0,1800155161:13.0,1800183197:13.0,1800020102:13.0]
8
[1804476879:13.0,1800019191:13.0,1808458391:13.0,1808431618:13.0,1808403836:13.0,1800112450:13.0,18084
35651:13.0,1808457961:13.0,1800062915:13.0,1808406133:13.0]
9
[1808467432:13.0,1808404170:12.510724,1808403030:12.503729,1808429384:12.503609,1800018810:12.50152,18
07634180:12.501279,1808404010:12.500442,1808421673:12.499154,1808420230:12.496009,1808410707:12.490825
]
10
[1808505124:13.0,1807592183:13.0,1807537463:12.67474,1808488292:12.667972,1808471149:12.665307,1808405
256:12.505206,1800265585:12.490035,1808442969:12.395251,1808415480:12.321217,1807859434:12.250883]
```

2. Clustering:

(1) Using datasets from Online news and Wikipedia articles

We choose the following two datasets to do clustering:

- reuters21578
- enwiki-20150901-pages-articles1.xml-p000000010p000010000

(2) Do clustering to find related documents

First we apply K-means and Fuzzy K-means on Reuters news dataset:

K-means clustering:

```
15/10/25 12:29:39 INFO Job: Job job_local1473567275_0004 completed successfully
15/10/25 12:29:39 INFO Job: Counters: 15
        File System Counters
                FILE: Number of bytes read=384817170
                FILE: Number of bytes written=307065530
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=21578
                Map output records=21578
                Input split bytes=164
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=12
                Total committed heap usage (bytes)=492830720
        File Input Format Counters
                Bytes Read=17038887
        File Output Format Counters
                Bytes Written=18065095
15/10/25 12:29:39 INFO MahoutDriver: Program took 18344 ms (Minutes: 0.305733333333
33336)
```

```
Key: 5646: Value: wt: 1.0 distance: 0.8102479274555142  vec: [{"1697":5.609},{"2689":3.135},{"3730":3.051},{"5997":1.1
Key: 1531: Value: wt: 1.0 distance: 0.7224820861423382  vec: [{"2689":3.135},{"3730":3.051},{"5997":1.119},{"6236":5.6
Key: 371: Value: wt: 1.0 distance: 0.9216730299859334  vec: [{"2962":2.807},{"3730":3.051},{"5636":9.593},{"5703":3.9
Key: 1531: Value: wt: 1.0 distance: 0.7031398855630863  vec: [{"1697":5.609},{"2689":3.135},{"2962":2.807},{"3730":3.0
Key: 18541: Value: wt: 1.0 distance: 0.5817853698775622  vec: [{"797":8.9},{"2689":3.135},{"2962":2.807},{"3730":3.051}
Key: 1531: Value: wt: 1.0 distance: 0.7343683798371794  vec: [{"2689":3.135},{"3170":2.829},{"3310":9.881},{"3730":3.0
Key: 5646: Value: wt: 1.0 distance: 0.8016823132333883  vec: [{"2540":9.188},{"2689":3.135},{"3730":3.051},{"4397":2.9
Key: 18734: Value: wt: 1.0 distance: 0.8459251196403172  vec: [{"2689":3.135},{"3730":3.051},{"5997":1.119},{"7424":4.5
Key: 15960: Value: wt: 1.0 distance: 0.8350294307372016  vec: [{"933":9.188},{"2689":3.135},{"3730":3.051},{"5997":1.11
Key: 659: Value: wt: 1.0 distance: 0.6897259934760255  vec: [{"2689":3.135},{"3000":7.514},{"3730":3.051},{"5433":7.7
Key: 1531: Value: wt: 1.0 distance: 0.7279302559069704  vec: [{"2689":3.135},{"3730":3.051},{"4397":5.024},{"5997":1.1
Key: 10601: Value: wt: 1.0 distance: 0.7403892433923107  vec: [{"689":9.593},{"2689":3.135},{"3170":2.829},{"3730":3.05
Key: 6533: Value: wt: 1.0 distance: 0.8124119927554354  vec: [{"2689":3.135},{"2983":9.188},{"3730":3.051},{"5848":3.8
Key: 10601: Value: wt: 1.0 distance: 0.6675468748327895  vec: [{"2962":2.807},{"3730":3.051},{"3932":7.514},{"4918":3.6
Key: 659: Value: wt: 1.0 distance: 0.5518133521258943  vec: [{"649":9.37},{"1509":7.545},{"1711":7.453},{"2090":7.088
Key: 10601: Value: wt: 1.0 distance: 0.8691348832521122  vec: [{"2689":3.135},{"3730":3.051},{"7027":4.332},{"30839":3.
Key: 1531: Value: wt: 1.0 distance: 0.7326471852442071  vec: [{"2689":3.135},{"2962":2.807},{"3730":3.051},{"4471":9.1
Key: 659: Value: wt: 1.0 distance: 0.720175488989391  vec: [{"1696":2.643},{"2689":3.135},{"2962":2.807},{"3358":4.31
Key: 659: Value: wt: 1.0 distance: 0.7337654372961773  vec: [{"1982":3.839},{"2689":3.135},{"3170":2.829},{"3730":3.0
Key: 659: Value: wt: 1.0 distance: 0.7574858324075376  vec: [{"1043":4.128},{"2689":3.135},{"2962":2.807},{"3730":3.0
Key: 659: Value: wt: 1.0 distance: 0.7363096348627212  vec: [{"1982":2.714},{"2273":9.37},{"2689":3.135},{"2962":2.80
Key: 659: Value: wt: 1.0 distance: 0.7287894296950976  vec: [{"955":9.37},{"2689":3.135},{"3170":2.829},{"3730":3.051
Key: 1531: Value: wt: 1.0 distance: 0.6777502276330576  vec: [{"616":9.034},{"2689":3.135},{"3730":3.051},{"5997":1.11
Key: 5646: Value: wt: 1.0 distance: 0.7965285548503003  vec: [{"2689":3.135},{"3730":3.051},{"4500":4.899},{"5997":1.1
Key: 3362: Value: wt: 1.0 distance: 0.86023320857273  vec: [{"1696":2.643},{"2962":2.807},{"3730":3.051},{"5081":3.633
Key: 8462: Value: wt: 1.0 distance: 0.7869521142778453  vec: [{"2689":3.135},{"3730":3.051},{"5359":9.188},{"6316":3.1
Key: 659: Value: wt: 1.0 distance: 0.7751149363273682  vec: [{"2689":3.135},{"3730":3.051},{"5081":3.633},{"5241":3.4
Key: 1531: Value: wt: 1.0 distance: 0.8307084824069053  vec: [{"2689":3.135},{"3730":3.051},{"5081":3.633},{"5997":1.1
Key: 659: Value: wt: 1.0 distance: 0.6494662008824592  vec: [{"2460":2.742},{"2689":3.135},{"3730":3.051},{"4918":3.6
Key: 659: Value: wt: 1.0 distance: 0.6428341618646571  vec: [{"1512":7.545},{"1528":9.602},{"1890":9.37},{"2215":9.18
Key: 5798: Value: wt: 1.0 distance: 0.7720094460152325  vec: [{"2689":3.135},{"3730":3.051},{"3948":3.954},{"4747":4.0
Key: 1531: Value: wt: 1.0 distance: 0.850513526595389  vec: [{"2689":3.135},{"3730":3.051},{"4481":7.647},{"4747":4.02
Key: 371: Value: wt: 1.0 distance: 0.8835230467926887  vec: [{"2490":7.129},{"2689":3.135},{"3730":3.051},{"3861":8.4
Key: 864: Value: wt: 1.0 distance: 0.860525819961259  vec: [{"2563":8.677},{"2689":3.135},{"3555":3.035},{"3730":3.05
Key: 659: Value: wt: 1.0 distance: 0.5775528611191603  vec: [{"834":8.677},{"1555":7.453},{"2689":3.135},{"3730":3.05
Count: 21578
15/10/25 12:31:00 INFO MahoutDriver: Program took 2508 ms (Minutes: 0.0418)
```

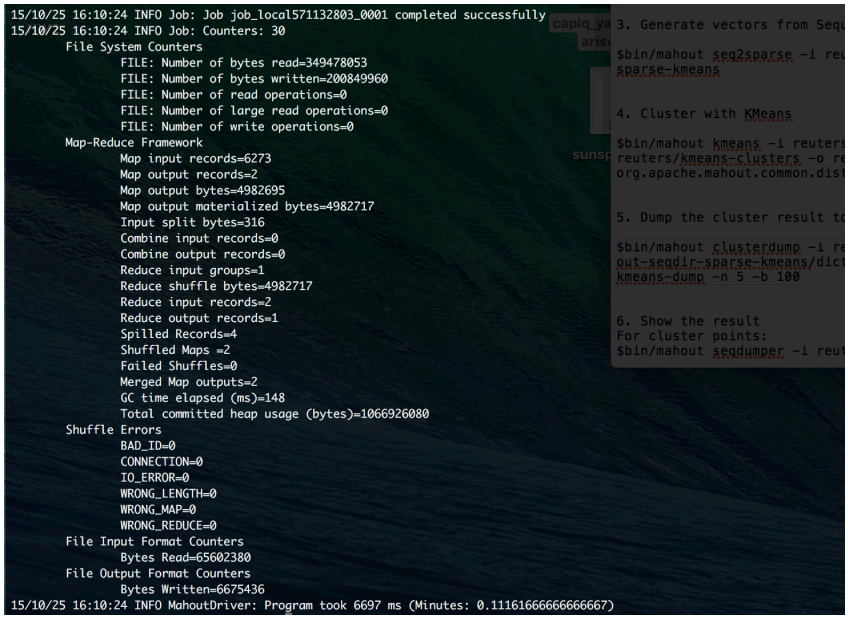
Fuzzy K-means clustering:

```
15/10/25 12:44:48 INFO Job: Job job_local656430242_0003 completed successfully
15/10/25 12:44:48 INFO Job: Counters: 15
        File System Counters
                FILE: Number of bytes read=421570140
                FILE: Number of bytes written=336995791
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=21578
                Map output records=21578
                Input split bytes=164
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=62
                Total committed heap usage (bytes)=374341632
        File Input Format Counters
                Bytes Read=17038887
        File Output Format Counters
                Bytes Written=18064863
15/10/25 12:44:48 INFO MahoutDriver: Program took 37710 ms (Minutes: 0.6285)
```

```
Key: 19713: Value: wt: 1.0 distance: 0.8944488474491454  vec: [{"1697":5.609},{"2689":3.135},{"3730":3.051},{"5997":1.1
Key: 10770: Value: wt: 1.0 distance: 0.8215352424766806  vec: [{"2689":3.135},{"3730":3.051},{"5997":1.119},{"6236":5.6
Key: 3632: Value: wt: 1.0 distance: 0.9433907515767265  vec: [{"2962":2.807},{"3730":3.051},{"5636":9.593},{"5703":3.9
Key: 10770: Value: wt: 1.0 distance: 0.823656911829862  vec: [{"1697":5.609},{"2689":3.135},{"2962":2.807},{"3730":3.05
Key: 10770: Value: wt: 1.0 distance: 0.821178598849267  vec: [{"797":8.9},{"2689":3.135},{"2962":2.807},{"3730":3.051},
Key: 10770: Value: wt: 1.0 distance: 0.8391374242563706  vec: [{"2689":3.135},{"3170":2.829},{"3310":9.881},{"3730":3.0
Key: 10770: Value: wt: 1.0 distance: 0.8774963852944634  vec: [{"2540":9.188},{"2689":3.135},{"3730":3.051},{"4397":2.9
Key: 10770: Value: wt: 1.0 distance: 0.9033489384910172  vec: [{"2689":3.135},{"3730":3.051},{"5997":1.119},{"7424":4.5
Key: 18309: Value: wt: 1.0 distance: 0.8904144999396243  vec: [{"933":9.188},{"2689":3.135},{"3730":3.051},{"5997":1.11
Key: 3632: Value: wt: 1.0 distance: 0.8440708899234151  vec: [{"2689":3.135},{"3000":7.514},{"3730":3.051},{"5433":7.7
Key: 10770: Value: wt: 1.0 distance: 0.8252576101160689  vec: [{"2689":3.135},{"3730":3.051},{"4397":5.024},{"5997":1.1
Key: 18309: Value: wt: 1.0 distance: 0.8309546087624643  vec: [{"689":9.593},{"2689":3.135},{"3170":2.829},{"3730":3.05
Key: 18309: Value: wt: 1.0 distance: 0.8434792643576414  vec: [{"2689":3.135},{"2983":9.188},{"3730":3.051},{"5848":3.8
Key: 19713: Value: wt: 1.0 distance: 0.8726591351570927  vec: [{"2962":2.807},{"3730":3.051},{"3932":7.514},{"4918":3.6
Key: 3632: Value: wt: 1.0 distance: 0.8189114506286984  vec: [{"649":9.37},{"1509":7.545},{"1711":7.453},{"2090":7.088
Key: 3632: Value: wt: 1.0 distance: 0.9259366882680182  vec: [{"2689":3.135},{"3730":3.051},{"7027":4.332},{"30839":3.
Key: 10770: Value: wt: 1.0 distance: 0.8270938788222709  vec: [{"2689":3.135},{"2962":2.807},{"3730":3.051},{"4471":9.1
Key: 3632: Value: wt: 1.0 distance: 0.8565607439727175  vec: [{"1696":2.643},{"2689":3.135},{"2962":2.807},{"3358":4.3
Key: 3632: Value: wt: 1.0 distance: 0.8667990170391024  vec: [{"1982":3.839},{"2689":3.135},{"3170":2.829},{"3730":3.0
Key: 3632: Value: wt: 1.0 distance: 0.869694044406639  vec: [{"1043":4.128},{"2689":3.135},{"2962":2.807},{"3730":3.05
Key: 3632: Value: wt: 1.0 distance: 0.8685169304792749  vec: [{"1982":2.714},{"2273":9.37},{"2689":3.135},{"2962":2.80
Key: 3632: Value: wt: 1.0 distance: 0.8653340016551617  vec: [{"955":9.37},{"2689":3.135},{"3170":2.829},{"3730":3.051
Key: 10770: Value: wt: 1.0 distance: 0.7832506280366477  vec: [{"616":9.034},{"2689":3.135},{"3730":3.051},{"5997":1.11
Key: 3632: Value: wt: 1.0 distance: 0.8973751845574479  vec: [{"2689":3.135},{"3730":3.051},{"4500":4.899},{"5997":1.1
Key: 10770: Value: wt: 1.0 distance: 0.9170032145028807  vec: [{"1696":2.643},{"2962":2.807},{"3730":3.051},{"5081":3.6
Key: 3632: Value: wt: 1.0 distance: 0.9420723425297225  vec: [{"2689":3.135},{"3730":3.051},{"5359":9.188},{"6316":3.1
Key: 3632: Value: wt: 1.0 distance: 0.8613135254723545  vec: [{"2689":3.135},{"3730":3.051},{"5081":3.633},{"5241":3.4
Key: 10770: Value: wt: 1.0 distance: 0.8641642374205118  vec: [{"2689":3.135},{"3730":3.051},{"5081":3.633},{"5997":1.1
Key: 3632: Value: wt: 1.0 distance: 0.828936147581975  vec: [{"2460":2.742},{"2689":3.135},{"3730":3.051},{"4918":3.67
Key: 3632: Value: wt: 1.0 distance: 0.8456475378733378  vec: [{"1512":7.545},{"1528":9.602},{"1890":9.37},{"2215":9.18
Key: 18309: Value: wt: 1.0 distance: 0.8543240999782055  vec: [{"2689":3.135},{"3730":3.051},{"3948":3.954},{"4747":4.0
Key: 18309: Value: wt: 1.0 distance: 0.8763700768290311  vec: [{"2689":3.135},{"3730":3.051},{"4481":7.647},{"4747":4.0
Key: 3025: Value: wt: 1.0 distance: 0.9258367793715158  vec: [{"2490":7.129},{"2689":3.135},{"3730":3.051},{"3861":8.4
Key: 18309: Value: wt: 1.0 distance: 0.8654578762723307  vec: [{"2563":8.677},{"2689":3.135},{"3555":3.035},{"3730":3.0
Key: 3632: Value: wt: 1.0 distance: 0.8398590781420272  vec: [{"834":8.677},{"1555":7.453},{"2689":3.135},{"3730":3.05
Count: 21578
15/10/25 12:47:31 INFO MahoutDriver: Program took 3038 ms (Minutes: 0.050633333333333336)
```

By comparing the two clustering results, we find that the result we get using K-means is different from what we get using Fuzzy K-means. Besides, Fuzzy K-means will take more time to process the data.

Second, we apply Canopy clustering on Wikipedia articles dataset:



By observing the running time, we find that Canopy clustering takes more time even thought we use a smaller dataset. According to the algorithm, we know that Canopy clustering needs to find the number of clusters that have approximately that size first in order to prevent all points close to an already existing canopy from being the center of a new canopy. This can explain the longer running time of Canopy clustering.

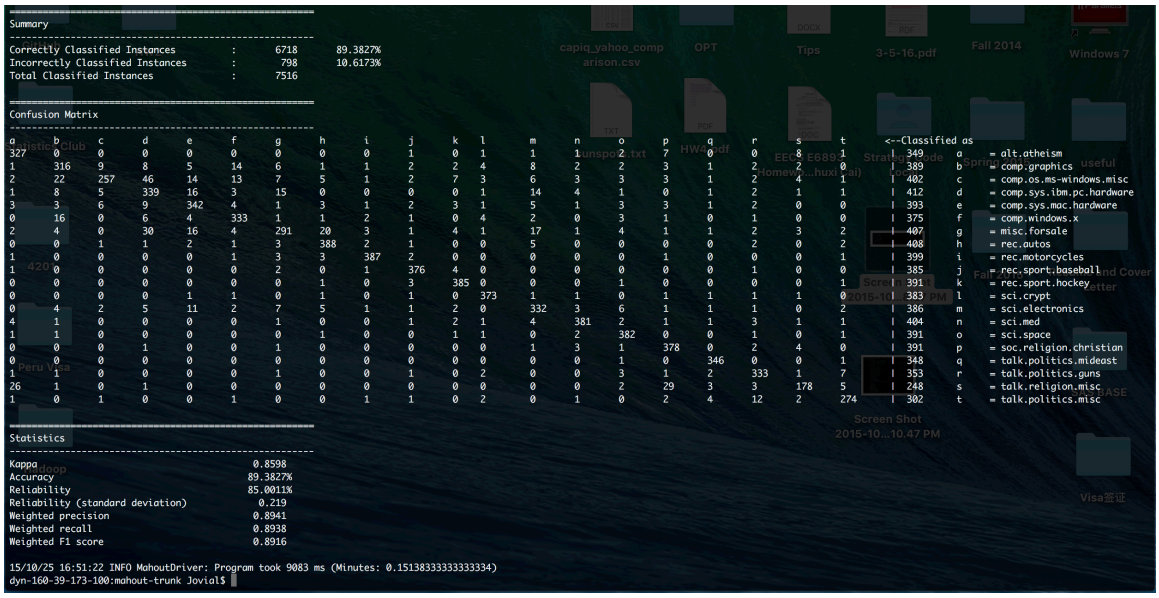
3. Classification:

(1) Using the 20 newspaper data, try various classification algorithms provided by Mahout, and discuss their performance

First we apply the classification algorithms provided by Mahout:

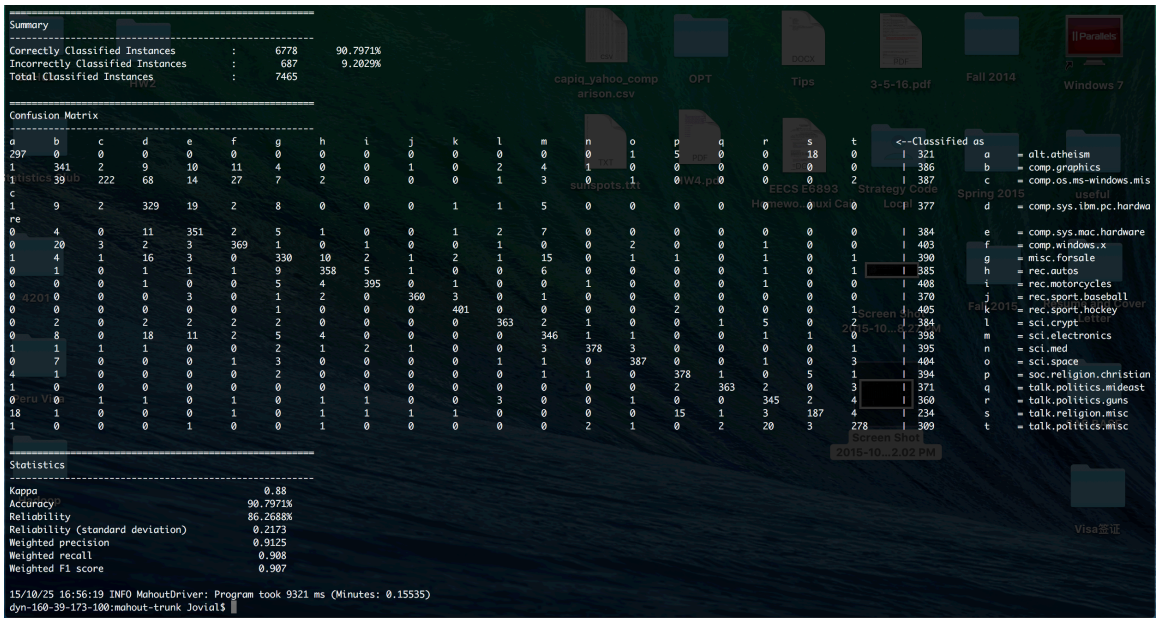
We apply Complement Naïve Bayes, Naïve Bayes and Stochastic Gradient Descent on our 20 newsgroups data:

Complement Naïve Bayes:



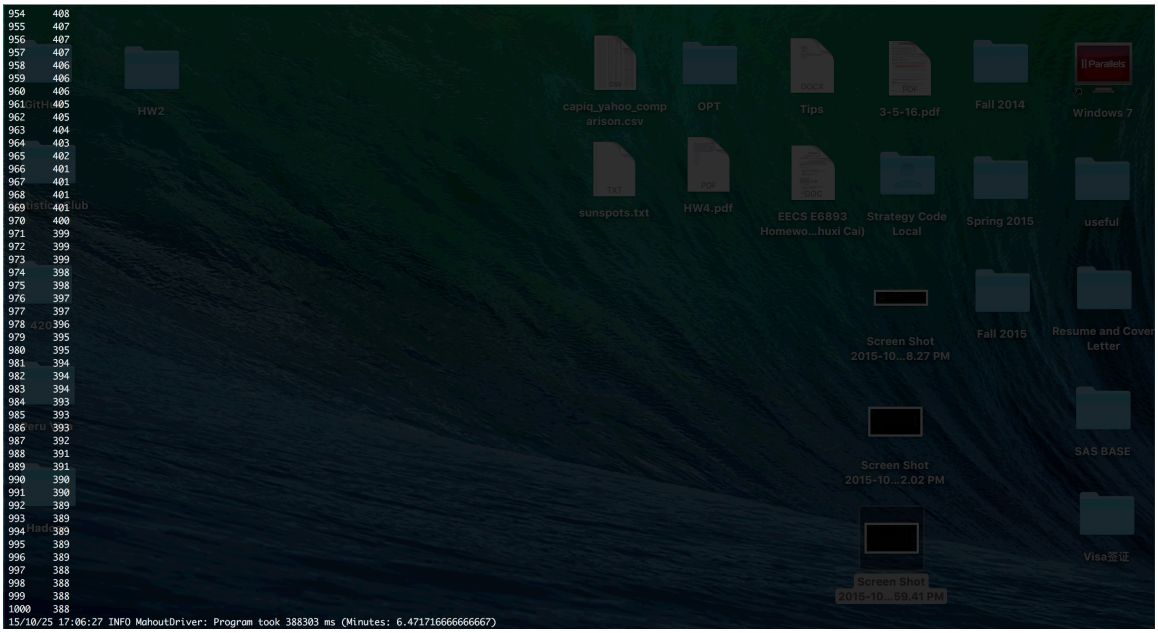
```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       6718       89.3827%
Incorrectly Classified Instances        :        798       10.6173%
Total Classified Instances              :       7516

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       c       d       e       f       g       h       i       j       k       l       m       n       o       p       q       r       s       t       <--Classified as
327     0       0       0       0       0       0       0       0       1       0       1       1       1       2       7       0       0       8       1     |  349       a     = alt.atheism
1       316     9       8       5       14      6       1       1       2       2       4       8       2       2       3       1       2       2       0     |  389       b     = comp.graphics
2       22      257     46      14      13      7       5       1       2       7       3       6       3       3       3       1       2       4       1     |  402       c     = comp.os.ms-windows.misc
1       8       5       339     16      3       15      0       0       0       0       1       14      4       1       0       1       2       1       1     |  412       d     = comp.sys.ibm.pc.hardware
3       3       6       9       342     4       1       3       1       2       3       1       5       1       3       3       1       2       0       0     |  393       e     = comp.sys.mac.hardware
0       16      0       6       4       333     1       1       2       1       0       4       2       0       3       1       0       1       0       0     |  375       f     = comp.windows.x
2       4       0       30      16      4       291     20      3       1       4       1       17      1       4       1       1       2       3       2     |  407       g     = misc.forsale
0       0       1       1       2       1       3       388     2       1       0       0       5       0       0       0       0       2       0       2     |  408       h     = rec.autos
1       0       0       0       0       1       3       3       387     2       0       0       0       0       0       1       0       0       0       1     |  399       i     = rec.motorcycles
1       0       0       0       0       0       2       0       1       376     4       0       0       0       0       0       0       1       0       0     |  385       j     = rec.sport.baseball
0       0       0       0       0       0       0       1       0       3       385     0       0       0       1       0       0       0       0       1     |  391       k     = rec.sport.hockey
0       0       0       0       1       1       0       1       0       1       0       373     1       1       0       1       1       1       1       0     |  383       l     = sci.crypt
0       4       2       5       11      2       7       5       1       1       2       0       332     3       6       1       1       1       0       2     |  386       m     = sci.electronics
4       1       0       0       0       0       1       0       0       1       2       1       4       381     2       1       1       3       1       1     |  404       n     = sci.med
1       1       0       0       0       0       0       1       0       0       1       1       0       2       382     0       0       1       0       1     |  391       o     = sci.space
0       0       0       1       0       0       1       0       0       0       0       0       1       3       1       378     0       2       4       0     |  391       p     = soc.religion.christian
0       0       0       0       0       0       0       0       0       0       0       0       0       0       1       0       346     0       0       1     |  348       q     = talk.politics.mideast
1       1       0       0       0       0       1       0       0       1       0       2       0       0       3       1       2       333     1       7     |  353       r     = talk.politics.guns
26      1       0       1       0       0       0       0       0       0       0       0       0       0       2       29      3       3       178     5     |  248       s     = talk.religion.misc
1       0       1       0       0       1       0       0       1       1       0       2       0       1       0       2       4       12      2       274   |  302       t     = talk.politics.misc

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.8598
Accuracy                                   89.3827%
Reliability                                85.0011%
Reliability (standard deviation)            0.219
Weighted precision                          0.8941
Weighted recall                             0.8938
Weighted F1 score                           0.8916

15/10/25 16:51:22 INFO MahoutDriver: Program took 9083 ms (Minutes: 0.15138333333333334)
dyn-160-39-173-100:mahout-trunk Jovial$
```

Naïve Bayes:



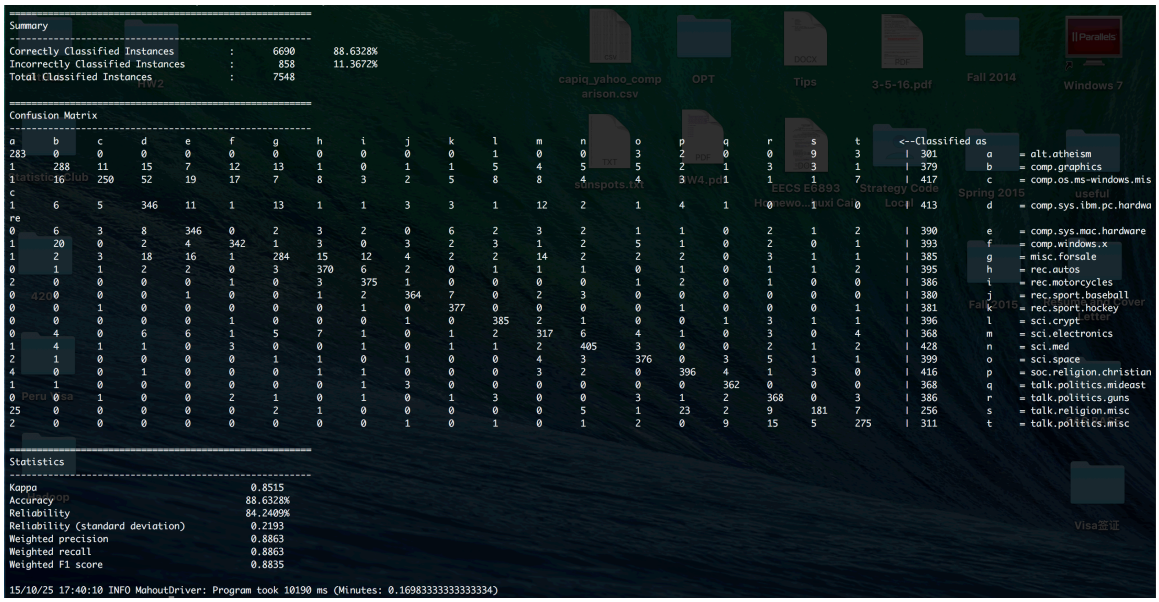
```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       6778       90.7971%
Incorrectly Classified Instances        :        687        9.2029%
Total Classified Instances              :       7465

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       c       d       e       f       g       h       i       j       k       l       m       n       o       p       q       r       s       t       <--Classified as
297     0       0       0       0       0       0       0       0       0       0       0       0       0       1       5       0       0       18      0     |  321       a     = alt.atheism
1       341     2       9       10      11      4       0       0       1       0       2       4       1       0       0       0       0       0       0     |  386       b     = comp.graphics
1       39      222     68      14      27      7       2       0       0       0       1       3       0       1       0       0       0       2       1     |  387       c     = comp.os.ms-windows.mis
c
1       9       2       329     19      2       8       0       0       0       1       1       5       0       0       0       0       0       0       0     |  377       d     = comp.sys.ibm.pc.hardwa
re
0       4       0       11      351     2       5       1       0       0       1       2       7       0       0       0       0       0       0       0     |  384       e     = comp.sys.mac.hardware
0       20      3       2       3       369     1       0       1       0       0       1       0       0       2       0       0       1       0       0     |  403       f     = comp.windows.x
1       4       1       16      3       0       330     10      2       1       2       1       15      0       1       1       0       1       0       1     |  390       g     = misc.forsale
0       1       0       1       1       1       9       358     5       1       0       0       6       0       0       0       0       1       0       1     |  385       h     = rec.autos
0       0       0       1       0       0       5       4       395     0       1       0       0       1       0       0       0       1       0       0     |  408       i     = rec.motorcycles
0       0       0       0       3       0       1       2       0       360     3       0       1       0       0       0       0       0       0       0     |  370       j     = rec.sport.baseball
0       0       0       0       0       0       0       1       0       0       0       401     0       0       0       0       0       2       0       0     |  405       k     = rec.sport.hockey
0       2       0       2       2       2       2       0       0       0       0       0       363     2       1       0       0       1       5       0     |  384       l     = sci.crypt
0       8       0       18      11      2       5       4       0       0       0       0       0       346     1       1       0       0       1       1     |  398       m     = sci.electronics
1       1       1       1       0       0       0       2       1       2       1       0       0       3       378     3       0       0       0       0     |  395       n     = sci.med
0       7       0       0       0       1       3       0       0       0       0       0       1       1       0       387     0       0       1       0     |  404       o     = sci.space
4       1       0       0       0       0       0       2       0       0       0       0       0       1       1       0       378     1       0       5     |  394       p     = soc.religion.christian
1       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       2       363     2       0     |  371       q     = talk.politics.mideast
0       0       1       1       0       1       0       1       1       0       0       0       3       0       0       1       0       0       345     2     |  360       r     = talk.politics.guns
18      1       0       0       0       1       0       1       1       1       1       0       0       0       0       15      1       3       187     4     |  234       s     = talk.religion.misc
1       0       0       0       1       0       0       1       0       0       0       0       0       0       2       1       0       2       20      3     |  309       t     = talk.politics.misc

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.88
Accuracy                                   90.7971%
Reliability                                86.2688%
Reliability (standard deviation)            0.2173
Weighted precision                          0.9125
Weighted recall                             0.908
Weighted F1 score                           0.907

15/10/25 16:56:19 INFO MahoutDriver: Program took 9321 ms (Minutes: 0.15535)
dyn-160-39-173-100:mahout-trunk Jovial$
```

Stochastic Gradient Descent



By comparing three classification results, we find that SGD takes much more running time than Complement Naïve Bayes and Naïve Bayes. The reason is that SGD does the same simple operation for each training example and the data size here is quite large. For Complement Naïve Bayes and Naïve Bayes, they have the similar running time while Naïve Bayes has higher accuracy. So Naïve Bayes is the best algorithm here to apply on our 20 newsgroups data.

Second we train our own model and apply our model on the 20 newsgroups data:



By observing the running time and accuracy, we can conclude that our model

performs well.

(2) Do similar experiments on the Wikipedia data that you downloaded

Here we apply CBayes and Binary CBayes classification algorithms on the Wikipedia data:

```
Please select a number to choose the corresponding task to run
1. CBayes (may require increased heap space on yarn)
2. BinaryCBayes
3. clean -- cleans up the work area in /tmp/mahout-work-wiki
Enter your choice :
```

CBayes:

```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       1399       83.4228%
Incorrectly Classified Instances        :        278       16.5772%
Total Classified Instances              :       1677

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       c       d       e       f       g       h       i       j       <--Classified as
391     9       13      0       3       4       2       21      9       3       |  455       a     = australia
3       114     1       4       0       1       0       2       1       2       |  128       b     = austria
1       0       4       0       0       0       0       0       0       0       |  5         c     = bahamas
1       4       7       342     9       2       1       6       7       2       |  381       d     = canada
0       1       1       0       28      0       0       0       0       2       |  32        e     = colombia
0       0       1       0       1       28      1       1       0       1       |  33        f     = cuba
0       0       1       0       2       1       58      2       0       0       |  64        g     = pakistan
0       0       0       1       2       0       0       3       0       1       |  7         h     = panama
2       19      41      6       15      8       11      23      381     7       |  513       i     = united kingdom
0       0       0       1       1       1       5       1       0       50      |  59        j     = vietnam

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.7598
Accuracy                                   83.4228%
Reliability                                73.6005%
Reliability (standard deviation)            0.2791
Weighted precision                          0.9137
Weighted recall                             0.8342
Weighted F1 score                           0.8645

15/10/25 19:57:58 INFO MahoutDriver: Program took 10316 ms (Minutes: 0.17193333333333333)
dyn-160-39-173-100:mahout-trunk Jovial$
```

Binary CBayes:

```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       2083       86.3599%
Incorrectly Classified Instances        :        329       13.6401%
Total Classified Instances              :       2412

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       <--Classified as
497     7       |  504       a     = united kingdom
322     1586    |  1908      b     = united states

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.6626
Accuracy                                   86.3599%
Reliability                                60.5783%
Reliability (standard deviation)            0.5303
Weighted precision                          0.9144
Weighted recall                             0.8636
Weighted F1 score                           0.8737

15/10/25 20:09:33 INFO MahoutDriver: Program took 3957 ms (Minutes: 0.06595)
dyn-160-39-173-100:mahout-trunk Jovial$
```

By comparing the results of two classification algorithms, we find that Binary CBayes here takes less time and has higher accuracy. So we conclude that Binary CBayes performs better here.