

GEBZE TECHNICAL UNIVERSITY
Computer Engineering Department



CSE 611-CSE 458
Big Data Analytics
Homework 2

ONUR
SEZER
121044074

April – 2017
Gebze – KOCAELİ

a)

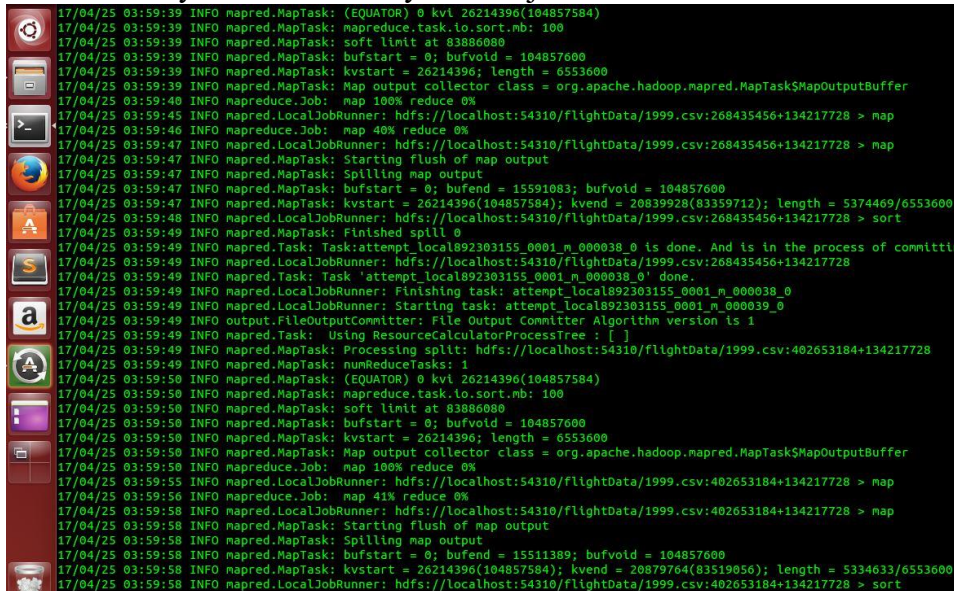
```
export HADOOP_CLASSPATH=/usr/lib/jvm/java-7-openjdk-amd64/lib/tools.jar
```

```
hadoop com.sun.tools.javac.Main CalculateDelay.java
```

```
jar cf cd.jar CalculateDelay*.class
```

```
hadoop jar cd.jar CalculateDelay /flightData /cikti
```

Terminalden yukardaki kodlar yazılarak java kodu derlenir



```
public class CalculateDelay extends Configured implements Tool {
    public static class MapClass extends MapReduceBase implements Mapper<LongWritable, Text, Text,
Text> {
        private Text loc = new Text();
        private Text rating = new Text();

        @Override
        public void map(LongWritable key, Text value, OutputCollector<Text, Text> output, Reporter
reporter)
            throws IOException {
            String[] rows = value.toString().split(",");
            if (rows.length == 29) {
                String arrDelay = rows[14]; // actual departure time
                String depDelay = rows[15]; // scheduled departure time

                loc.set("flight");

                rating.set(arrDelay + "\t" + depDelay);
                output.collect(loc, rating);
            }
        }
    }

    public static class Reduce extends MapReduceBase implements Reducer<Text, Text, Text, Text> {
        @Override
        public void reduce(Text key, Iterator<Text> values, OutputCollector<Text, Text> output,
Reporter reporter)
            throws IOException {
            int arrDelayNum = 0;
            int depDelayNum = 0;

            while (values.hasNext()) {
                String tokens[] = (values.next().toString()).split("\t");
```

```

        if(!(tokens[0].equals("NA") || tokens[1].equals("NA")))
        {

            int arrDelay = Integer.parseInt(tokens[0]);
            int depDelay = Integer.parseInt(tokens[1]);

            if(arrDelay > depDelay)
                arrDelayNum++;
            else
                depDelayNum++;

        }
    }
    if(depDelayNum > arrDelayNum){
        output.collect(key, new Text( " depDelay > arrDelay  ->> result = 0"));
    }
    else{
        output.collect(key, new Text( " arrDelay > depDelay  ->> result = 1"));
    }
}

}

static int printUsage() {
    System.out.println("FlightRatings [-m <maps>] [-r <reduces>] <input> <output>");
    return 0;
}

@Override
public int run(String[] args) throws IOException {
    return 0;
}

public static void main(String[] args) throws IOException {
    JobConf conf = new JobConf(CalculateDelay.class);
    conf.setJobName("CalculateDelay");

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

    conf.setMapperClass(MapClass.class);
    conf.setReducerClass(Reduce.class);

    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);

    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));

    JobClient.runJob(conf);
}
}

```

Sonuc :

```
1 fligt depDelay > arrDelay ->> result = 0
```

b)

SVM: Sınıflandırma (Classification) konusunda kullanılan oldukça etkili ve basit yöntemlerden birisidir. Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır. İşte SVM bu sınırın nasıl çizileceğini belirler.

Bu işlemin yapılması için iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizilir ve bu sınır çizgileri birbirine yaklaştırılarak ortak sınır çizgisi üretilir.

Decision Tree: decision tree learning yöntemi, makine öğrenmesi konularından birisidir. Literatürde karar ağacı öğrenmesinin alt yöntemleri olarak kabul edilebilecek sınıflandırma ağacı (classification tree) veya ilkelleştirme ağacı (regression tree ,tahmin ağacı) gibi uygulamaları vardır. Karar ağacı öğrenmesinde, bir ağaç yapısı oluşturularak ağacın yaprakları seviyesinde sınıf etiketleri ve bu yapraklara giden ve başlangıçtan çıkan kollar ile de özellikler üzerindeki işlemler ifade edilmektedir.

c)

PCA: Bilgisayar bilimlerinde boyut indirmeye yarayan bir yöntemdir. Kısaca iki bilgi arasında bir bağlantı varsa bu bağlantı sayesinde iki veriden birisini tutmak ve bağlantıyı tutmak iki bilginin de geri bulunabilmesini sağlar. Kısaca PCA olarak da ifade edilen bu terime göre bir veri kümesinin (veri matrisinin , data matrix) kovaryans matrisinin (covariance matrix) veya tekil değer çıkarımının (singular value decomposition) yöntemi ile elde edilen basitleştirilmiş halidir.

d)

Rastgele Orman (Random Forest) : Sınıflandırma işlemi sırasında birden fazla karar ağacı kullanılarak sınıflandırma değerinin yükseltilmesi hedeflenir.

e)

Linear Regression, en basit supervised learning algoritmalarından biridir. Tahmin etmeye çalıştığımız Y değişkeni ile tahminleyici değişkenlerimiz X_1, X_2, \dots, X_n arasında doğrusal bir ilişki olduğunu var sayar. Ancak bir önceki yazıda da belirttiğimiz üzere gerçek regresyon fonksiyonu azaltılamaz hatalar yüzünden hiçbir zaman doğrusal bir metotla tam olarak modellenemez.