Esercitazione
Big Data Analytics

Esercitazione Hadoop & Spark

MapReduce

L'esercitazione prevede l’analisi delle reviews. In particolare:

1. **Determinare** per ogni mese dell’anno (01,02,03…) il **sentiment score** medio di ogni recensione.

2. **Produrre** come output la lista dei soli *review_id* relative a frasi con **sentiment score positivo.**

Road Map

1. Analisi SentiNet e Dataset CSV.
2. Classi:
    a. **Preprocessing Sentiment:**
        i. ChallengeSentiMapper
        ii. ChallengeSentiReducer
    b. **Analisi reviews dataset:**
        i. ChallengeMapper
        ii. ChallengeReducer
        iii. ChallengePartitioner
        iv. ChallengeDriver

*Il driver è univoco, si farà lui carico di eseguire entrambi gli algoritmi.

ChallengeDriver

Abbiamo previsto che il **Driver** si occuperà di entrambi i job legati agli algoritmi di **MapReducer**:

1. **Preprocessing** del dataset del sentiment.
2. **Analisi** del dataset delle reviews.

```
//configurazione alto livello
Configuration config = new Configuration();
Job jobSenti = Job.getInstance(config, "Job Name: Challenge SENTI");
//la classe principale al interno jar è WordCountDriver.class)
jobSenti.setJarByClass(ChallengeDriver.class); // Indico la classe che costituirà l'entry point del job
//il mapper della classe / progetto
jobSenti.setMapperClass(ChallengeSentiMapper.class);
//il reducer della classe / progetto
jobSenti.setReducerClass(ChallengeSentiReducer.class);

jobSenti.setOutputKeyClass(Text.class);
jobSenti.setOutputValueClass(FloatWritable.class);

//visto prima...
FileInputFormat.addInputPath(jobSenti, new Path(inputDir));
FileOutputFormat.setOutputPath(jobSenti, new Path(outputDir));

boolean success = jobSenti.waitForCompletion(true); // success vale true se il job termina correttamente, false altrimenti
if (!success) {
        throw new IllegalStateException("Job Senti Challenge failed!");
}
```

```
config = new Configuration();
Job jobReview = Job.getInstance(config, "Job Name Review : Challenge");
jobReview.setJarByClass(ChallengeDriver.class); // Indico la classe che costituirà l'entry point del job
//il mapper della classe / progetto
jobReview.setMapperClass(ChallengeMapper.class);
//il reducer della classe / progetto
jobReview.setReducerClass(ChallengeReducer.class);

//nel caso utilizzo dei partitioners
jobReview.setPartitionerClass(ChallengePartitioner.class);
//relazione tra reducer e task .... (quanti reducer, quanti task ...)
jobReview.setNumReduceTasks(12); //1 per ogni mese

jobReview.setOutputKeyClass(Text.class);
jobReview.setOutputValueClass(FloatWritable.class);

//visto prima...
FileInputFormat.addInputPath(jobReview, new Path(inputDir));
FileOutputFormat.setOutputPath(jobReview, new Path(outputDir));
boolean successReview = jobReview.waitForCompletion(true); // success vale true se il job termina correttamente, false
if (!successReview) {
        throw new IllegalStateException("Job Review Challenge failed!");
}
```

ChallengeSentiMapper

```
//1 Step - converto il nostro value in text, perché dobbiamo lavorare con String Tokenizer
String text=value.toString(); //converto in String perchè non lavoriamo con i Text
if(text.charAt(0) != 35) {
        //Tokenizzo - Text che ho convertito da value e la punteggiatura che su cui voglio tokenizzare Salvo su works
        StringTokenizer sentiWords = new StringTokenizer(text, "\t");
        //finché ci sono parole (TOKEN) ....
        String votazione = "";
        String test = "";
        while (sentiWords.hasMoreTokens()) {
                votazione += sentiWords.nextToken() + ";";
        }
        String [] campi = votazione.split(";");
        if(campi.length > 5){
                Float somma_voti = Float.parseFloat(campi[2]) - Float.parseFloat(campi[3]);
                if(somma_voti != 0 ){
                        String [] parole_riga = campi[4].split(" ");
                        for(int z = 0; z < parole_riga.length ; z++) {
                                test = parole_riga[z].split("#")[0].toLowerCase();
                                context.write(new Text(test),new FloatWritable(somma_voti));
                        }
                }
        }
}
```

1. Separare contestualmente parola e #
2. Scrivere la parola come chiave e il suo valore
3. Il valore corrisponde alla differenza tra sentiment positivo e quello negativo

ChallengeSentiReducer

```
//i primi 2 parametri sono la coppia key-(list of values) )
protected void reduce(Text key, Iterable<FloatWritable> votazioni, Context context) throws IOException, InterruptedException {
        float avg=0;
        int i=0;
        for (FloatWritable votazione: votazioni) {
                avg += votazione.get();
                i++;
        }
        avg = avg/i;
        context.write(key, new FloatWritable(avg));
}
```

Il **reducer** relativo al **preprocessing**, si occupa di calcolare la media relativa alle singole parole contenute.

ChallengeMapper

Il challenge mapper, si occupa di **controllare** il valore di ogni singola word, attraverso una HashMap ove è contenuto il risultato del preprocessing del dataset SentiNet

```
if(reviews[0].equals("marketplace")){
        FileSystem fs = FileSystem.get(new Configuration());
        Path file = new Path("/output6SENTI/part-r-00000");
        BufferedReader reader = new BufferedReader(new InputStreamReader(fs.open(file)));
        String line = "";
        while((line = reader.readLine()) != null){
                String [] extractedLineParts = line.split("\t");
                this.sentiWords.put(extractedLineParts[0], Float.parseFloat(extractedLineParts[1]));
        }
        reader.close();
}else if (!reviews[12].equals("") && !reviews[13].equals("")) {
        //ignoro le stringhe vuote
        String line = reviews[12].toLowerCase()+" "+reviews[13].toLowerCase();
        StringTokenizer parole = new StringTokenizer(line, " .,?!;:()[]{}'");
        while(parole.hasMoreTokens()) {
                String parola = parole.nextToken().toLowerCase().trim();
                if(!parola.equals(" ") && this.sentiWords.get(parola)!=null){
                        float votazione = this.sentiWords.get(parola);
                        context.write(new Text(reviews[2]+";"+reviews[14]),new FloatWritable(votazione));
```

ChallengePartitioner

Il **challenge partitioner**, contiene appunto lo snippet di codice per partizionare l'esecuzione in 12 "*sotto-task*" uno per ciascun mese, il tutto viene fatto sulla base della data.

```
public int getPartition(Text key, FloatWritable review, int numReduceTasks) {

	try{
		LocalDate date = LocalDate.parse(key.toString().split(";")[1]);

		switch(date.getMonthValue()) {
			case 1:
				return 0;
			case 2:
				return 1;
			case 3:
				return 2;
			case 4:
				return 3;
			case 5:
				return 4;
			case 6:
				return 5;
			case 7:
				return 6;
			case 8:
				return 7;
			case 9:
				return 8;
			case 10:
				return 9;
			case 11:
				return 10;
			case 12:
				return 11;
			default:
				return 0;
		}
	}catch(DateTimeParseException e){
		System.out.println("ERRORE PARTIONER");
		return 0;
	}
}
```

ChallengeReducer

```
protected void reduce(Text key, Iterable<FloatWritable> words, Context context) throws IOException, InterruptedException {
        float sum=0;
        int i=0;
        for (FloatWritable word: words) {
                sum += word.get();
                i++;
        }
        float avg = (float) sum/i;
        float rounded = (float) (Math.round(avg *100.0)/100.0);
        if(rounded > 0){
                context.write(new Text(key.toString().split(";")[0]), new FloatWritable(rounded));
        }
}
```

Snippet di codice nel Reducer per arrotondare alla seconda cifra decimale, come da specifiche.

Spark

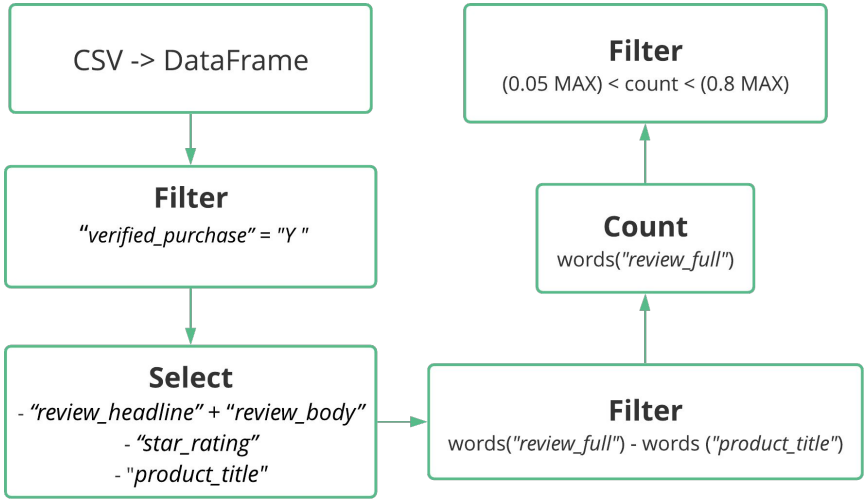
Utilizzando **pyspark** e **pysparkSQL** scrivere su file distinti in base allo “*star_rating*”:

- Numero di occorrenze sia in "*review_headline*" che in "*review_body*".
- **Eliminando** le sottostringhe presenti anche nel "*product_title*".
- **Solo** i record con “*verified_purchase*” = "*Y* ".
- **Solo** le parole tali che:

  0.05*MAX < occorrenza < 0.8*MAX.

Algoritmo

1. **Lettura** file CSV in DataFrame.
2. **Filtering** tenendo solo i record con “*verified_purchase*” = "*Y* "
3. **Select** di:
   a. *“review_headline”* e *“review_body”*
   b. *“star_rating”*
   c. *“product_title”*
4. **Rimozione** delle parole comprese nel “product_title”.
5. **Count** parole & calcolo dell’occorrenza **Massima**.
6. **Filtering** per il conteggio delle parole.
7. **Scrittura** risultati per ogni Star Rating.



Lettura file

1. **Schema** personalizzato per una buona formattazione del CSV
2. **Lettura** file e creazione **DataFrame**

```
customSchema = StructType([
    StructField("marketplace", StringType(), True),
    StructField("customer_id", IntegerType(), True),
    StructField("review_id", StringType(), True),
    StructField("product_id", StringType(), True),
    StructField("product_parent", IntegerType(), True),
    StructField("product_title", StringType(), True),
    StructField("product_category", StringType(), True),
    StructField("star_rating", IntegerType(), True),
    StructField("helpful_votes", IntegerType(), True),
    StructField("total_votes", IntegerType(), True),
    StructField("vine", StringType(), True),
    StructField("verified_purchase", StringType(), True),
    StructField("review_headline", StringType(), True),
    StructField("review_body", StringType(), True),
    StructField("review_date", DateType(), True)
])

#data_sample = "/content/gdrive/MyDrive/Colab Notebooks/BDA&ML/sample_us.tsv"
data_complete = "/home/andrea/Desktop/esercitazione_spark/amazon_reviews_us_Video_Games_v1_00.tsv"
data_sample = "/home/andrea/Desktop/esercitazione_spark/sample_us.tsv"


reviews = spark.read.option("sep", "\t").csv(data_complete, header=True, schema = customSchema)
```

review_id	product_id	product_parent	product_title	product_category	star_rating	helpful_votes	total_votes	vine	verified_purchase	review_headline	review_body	review_date
RTIS3L2M1F5SM	B001CXYMFS	737716809	Thrustmaster T-Fl...	Video Games	5	0	0	N	Y	an amazing joysti...	Used this for Eli...	2015-08-31
R1ZV7R40OLHKD	B00M920ND6	569686175	Tonsee 6 buttons ...	Video Games	5	0	0	N	Y	Definitely a sile...	Loved it, I didn...	2015-08-31
R3BH071QLH8QMC	B0029CSOD2	98937668	Hidden Mysteries:...	Video Games	1	0	1	N	Y	One Star	poor quality work...	2015-08-31
R127K9NTSXA2YH	B00GOOSV98	23143350	GelTabz Performan...	Video Games	3	0	0	N	Y	good, but could b...	nice, but tend to...	2015-08-31
R32ZWUXDJPW27Q	B00Y074JOM	821342511	Zero Suit Samus a...	Video Games	4	0	0	N	Y	Great but flawed.	Great amiibo, gre...	2015-08-31
R3AQQ4YUKJWBA6	B002UBI6W6	328764615	Psyclone Recharge...	Video Games	1	0	0	N	Y	One Star	The remote consta...	2015-08-31
R2F0POU5K6F73F	B008XHCLFO	24234603	Protection for yo...	Video Games	5	0	0	N	Y	A Must	I have a 2012-201...	2015-08-31
R3VNR804HYSMR6	B00BRA9R6A	682267517	Nerf 3DS XL Armor	Video Games	5	0	0	N	Y	Five Stars	Perfect, kids lov...	2015-08-31
R3GZTM72WA2QH	B009EPWJLA	435241890	One Piece: Pirate...	Video Games	5	0	0	N	Y	Five Stars	Excelent	2015-08-31
RNQOY62705W1K	B0000AV7GB	256572651	Playstation 2 Dan...	Video Games	4	0	0	N	Y	Four Stars	Slippery but expe...	2015-08-31
R1VTIA3JTYBY02	B00008KTNN	384411423	Metal Arms: Glitc...	Video Games	5	0	0	N	N	Five Stars	Love the game. Se...	2015-08-31
R29DOU8791QZL8	B000A3IA0Y	472622859	72 Pin Connector ...	Video Games	1	0	0	N	Y	Game will get stuck	Does not fit prop...	2015-08-31
R15DUT1VIJ9RJZ	B0053BQN34	577628462	uDraw Gametablet ...	Video Games	2	0	0	N	Y	We have tried it ...	This was way too ...	2015-08-31
R3IMF2MQ3OU9ZM	B002I0HIMI	988218515	NBA 2K12(Covers M...	Video Games	4	0	0	N	Y	Four Stars	Works great good ...	2015-08-31
R23H79DHOZTYAU	B0081EH12M	770100932	New Trigger Grips...	Video Games	1	1	1	N	Y	Now i have to buy...	It did not fit th...	2015-08-31
RIV24EQAIXA4O	B005FMLZQQ	24647669	Xbox 360 Media Re...	Video Games	5	0	0	N	Y	Five Stars	perfect lightweig...	2015-08-31
R3UCNGYDVN24YB	B002BSA388	33706205	Super Mario Galaxy 2	Video Games	5	0	0	N	Y	Five Stars	great	2015-08-31
RUL4H4XTTN2DY	B00BUSLSAC	829667834	Nintendo 3DS XL -...	Video Games	5	0	0	N	Y	Five Stars	Works beautifully...	2015-08-31
R20JF7Z4DHTNX5	B00KWF38AW	110680188	Captain Toad: Tr...	Video Games	5	0	0	N	Y	Five Stars	Kids loved the ga...	2015-08-31
R2T1AJ5MFI2260	B00BRQJYA8	616463426	Lego Batman 2: DC...	Video Games	4	0	0	N	Y	Four Stars	Goodngame	2015-08-31

Preprocessing

Estrazione record con *verified_purchase”* = ‘*Y*’

Estrazione colonne:

- *“review_headline”*
- *“review_body”*
- *“star_rating”*
- *“product_title”*

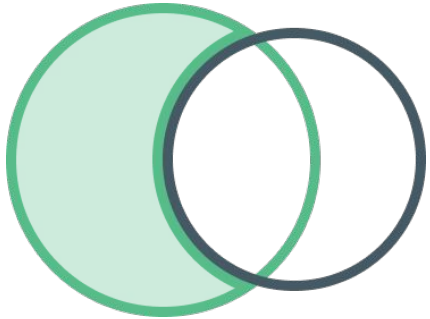
Trasformando in **lowercase** e **togliendo** caratteri speciali.

review_full	product_title	star_rating
an amazing joysti...	thustmaste t-flig...	5
definitely a sile...	tonsee 6 uttons w...	5
one sta poo quali...	hidden mysteies: ...	1
good, ut could e ...	geltaz pefomance ...	3
geat ut flawed ge...	zeo suit samus am...	4
one sta the emote...	psyclone echage s...	1
a must i have a 2...	potection fo you ...	5
five stas pefect,...	nef 3ds xl amo	5
five stas excelent	one piece: piate ...	5
fou stas slippey ...	playstation 2 dan...	4
game will get stu...	72 pin connecto f...	1
we have tied it w...	udaw gametalet wi...	2
fou stas woks gea...	na 2k12coves may vay	4
now i have to uy ...	new tigge gips la...	1
five stas pefect ...	xox 360 media emote	5
five stas geat	supe maio galaxy 2	5
five stas woks ea...	nintendo 3ds xl -...	5
five stas kids lo...	captain toad: te...	5
fou stas goodngame	lego atman 2: dc ...	4
not woth it ette ...	odycount	1

Filtering & conteggio parole

1. **Estrazione** parole all'interno di "*product_title"* in un nuovo DataFrame.
2. **Rimozione** delle parole appena estratte dal DF principale con un *'LeftAnti Join'*.
3. **Count delle** occorrenze.

LeftAnti Join Filtering



"review_full" ***"product_title"***

Filtraggio in base alle occorrenze

1. Calcolo *MAX* = occorrenza massima.
2. Filtraggio usando *MAX*.

```
MAX = reviews.agg(max('count')).collect()[0][0]
reviews = reviews.filter(f"count > {0.05*MAX} AND count < {0.8*MAX}")
```

Scrittura alternativa partizionata su file

- coalesce(1) per ridurre il numero di partizioni a 1 senza il shuffle.
- write.partitionBy(*“star_rating”*) per il partizionamento e scrittura su disco.
- 5 file diversi.


```
reviews.coalesce(1).write.partitionBy("star_rating").csv("/home/andrea/Desktop/esercitazione_spark/reviews_words", sep=";")
```

Team

La soluzione qui proposta è stata sviluppata, ideata e creata dal team :

- Andrian Melnic
- Lorenzo Federici
- Giacomo Licci
- Denis Bernovschi

Grazie per l’attenzione