...db-emotion-analysis

Load Data

FINISHED

Took 0 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:06:48 PM.

```
val basePath = "/user/yc7093_nyu_edu/imdb-reviews-w-emotion/part"
val fileSuffixes = List("-01-all") //, "-02-all", "-03-all", "-04-all")

val initialPath = s"$basePath${fileSuffixes.head}"
var rawDF = spark.read.parquet(initialPath)

for (suffix <- fileSuffixes.tail) {
  val fullPath = s"$basePath$suffix"
  val part_df = spark.read.parquet(fullPath)
  rawDF = rawDF.union(part_df)
}

rawDF.show(5)
```

SPARK JOB FINISHED

```
potter_tag|       review_detail| helpful|   predicted_emotion|             sadness|
joy|               love|              anger|               fear|           surprise|emo
tion|
+---------+-----------+--------------------+------+--------------------+----------------+-
----------+--------------------+--------+--------------------+--------------------+--------
-----------+--------------------+--------------------+--------------------+---------------
----+-------+
|rw5552176|  FeastMode|It Chapter Two (2...|   2.0|      bad and BORING|    15 March 2020|
0|I was enjoying it...|  [0, 2]|[{score -> 0.1004...| 0.10041969269514084| 0.67338609695434
57|0.017813226208090782|  0.1923339068889618|0.011774818412959576|0.004272174555808306|
joy|
|rw6455111|rapadgettra|Perry Mason: The ...|   4.0|        Not feasible|   8 January 2021|
0|It's hard when gu...|  [0, 0]|[{score -> 0.0441...| 0.04410260170698166| 0.68059456348419
19|0.013725311495363712| 0.24673616886138916|0.010360205546021461|0.004481049720197916|
joy|
|rw5178379| MehnaJain2|     Fixerr (2019– )|  10.0|              Fixerr|  11 October 2019|
0|Everyone plays th...|  [0, 0]|[{score -> 0.0035...|0.003539941739290...| 0.98156654834747
31|0.007583207450807095|0.002556430874392...|0.0024530023802071 81|0.002300753956660...|
```

Took 1 sec. Last updated by anonymous at December 08 2024, 6:17:03 PM. (outdated)

Rating Distribution

FINISHED

Took 0 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:06:25 PM.

```
val ratingDistribution = rawDF.groupBy("rating")
  .count()
  .orderBy("rating")

ratingDistribution.show()
```

SPARK JOB FINISHED

```
+------+-----+
|rating|count|
+------+-----+
|   1.0|35253|
|   2.0|13427|
|   3.0|13669|
|   4.0|14233|
|   5.0|18553|
|   6.0|23809|
|   7.0|31730|
|   8.0|38719|
|   9.0|35389|
|  10.0|79125|
+------+-----+

ratingDistribution: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [rating: doubl
e, count: bigint]
```

Took 3 sec. Last updated by anonymous at December 08 2024, 6:17:09 PM. (outdated)

```
val outputPath = "/user/yc7093_nyu_edu/imdb-emotion-analysis/rating-distribution"

ratingDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/rating-distribution
```

Took 2 sec. Last updated by anonymous at December 08 2024, 6:59:05 PM. (outdated)

Emotion Distribution

FINISHED

Took 0 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:06:14 PM.

```
val emotionDistribution = rawDF.groupBy("emotion")
  .count()
  .orderBy("emotion")

emotionDistribution.show()
```

SPARK JOB FINISHED

```
+--------+------+
| emotion| count|
+--------+------+
|   anger| 81480|
|    fear| 11217|
|     joy|158506|
|    love| 22475|
| sadness| 17854|
|surprise| 12375|
+--------+------+

emotionDistribution: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [emotion: str
ing, count: bigint]
```

Took 3 sec. Last updated by anonymous at December 08 2024, 6:17:12 PM. (outdated)

```
val outputPath = "/user/yc7093_nyu_edu/imdb-emotion-analysis/emotion-distribution"

emotionDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/emotion-distribution
```

Took 2 sec. Last updated by anonymous at December 08 2024, 6:58:48 PM. (outdated)

Ratings Distribution Within Emotion

FINISHED

Took 0 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:07:40 PM.

```
import org.apache.spark.sql.expressions.Window
import org.apache.spark.sql.functions._

val emotionWindow = Window.partitionBy("emotion")

val ratingDistributionWithinEmotion = rawDF.groupBy("emotion", "rating")
  .count()
  .withColumn("total_count", sum("count").over(emotionWindow))
  .withColumn("percentage", (col("count") / col("total_count")) * 100)
  .orderBy("emotion", "rating")


val rowCount = ratingDistributionWithinEmotion.count()
ratingDistributionWithinEmotion.show(rowCount.toInt, truncate = false)
```

SPARK JOB FINISHED

```
|sadness |7.0   |1389 |17854      |7.779769239386132 |
|sadness |8.0   |1521 |17854      |8.519099361487621 |
```

```
|sadness |9.0   |1292 |17854      |7.2364736193570O6 |
|sadness |10.0  |3253 |17854      |18.220006721182926|
|surprise|1.0   |746  |12375      |6.028282828282829 |

|surprise|2.0   |333  |12375      |2.690909090909091 |
|surprise|3.0   |377  |12375      |3.0464646464646465|
|surprise|4.0   |402  |12375      |3.2484848484848485|
|surprise|5.0   |503  |12375      |4.064646464646464 |
|surprise|6.0   |848  |12375      |6.852525252525253 |
|surprise|7.0   |1308 |12375      |10.569696969696968|
|surprise|8.0   |1866 |12375      |15.078787878787878|
|surprise|9.0   |1904 |12375      |15.385858585858585|
|surprise|10.0  |4088 |12375      |33.034343434343434|
+--------+------+-----+-----------+------------------+

import org.apache.spark.sql.expressions.Window
```

Took 5 sec. Last updated by anonymous at December 08 2024, 6:25:52 PM. (outdated)

```
val outputPath = "/user/yc7093_nyu_edu/imdb-emotion-analysis/rating-distribution-w-emotion"


ratingDistributionWithinEmotion.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/rating-distribution-w-emoti
on
```

Took 3 sec. Last updated by anonymous at December 08 2024, 6:58:13 PM. (outdated)

Emotion Distribution Within Ratings

FINISHED

Took 0 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:07:35 PM.

```
import org.apache.spark.sql.expressions.Window
import org.apache.spark.sql.functions._

val ratingWindow = Window.partitionBy("rating")

val emotionDistributionWithinRating = rawDF.groupBy("rating", "emotion")
  .count()
  .withColumn("total_count", sum("count").over(ratingWindow))
  .withColumn("percentage", (col("count") / col("total_count")) * 100)
  .orderBy("rating", "emotion")

val rowCount = emotionDistributionWithinRating.count()
```

SPARK JOB FINISHED

```
emotionDistributionWithinRating.show(rowCount.toInt, truncate = false)
```

```
|7.0   |love    |2420 |31730      |7.626851560037819 |
|7.0   |sadness |1389 |31730      |4.377560668137409 |
|7.0   |surprise|1308 |31730      |4.122281752284904 |
|8.0   |anger   |8391 |38719      |21.6715307730055  |
|8.0   |fear    |1617 |38719      |4.176244221183398 |
|8.0   |joy     |22050|38719      |56.94878483431906 |
|8.0   |love    |3274 |38719      |8.455796895580981 |

|8.0   |sadness |1521 |38719      |3.928303933469356 |
|8.0   |surprise|1866 |38719      |4.819339342441696 |
|9.0   |anger   |6157 |35389      |17.39806154454774 |
|9.0   |fear    |1185 |35389      |3.3484981208850204|
|9.0   |joy     |21489|35389      |60.72225832885925 |
|9.0   |love    |3362 |35389      |9.50012715815649  |
|9.0   |sadness |1292 |35389      |3.6508519596484783|
|9.0   |surprise|1904 |35389      |5.38020288790302  |
|10.0  |anger   |11075|79125      |13.996840442338074|
|10.0  |fear    |1848 |79125      |2.3355450236966826|
```

Took 5 sec. Last updated by anonymous at December 08 2024, 6:25:24 PM. (outdated)

```
val outputPath = "/user/yc7093_nyu_edu/imdb-emotion-analysis/emotion-distribution-w-rating"

emotionDistributionWithinRating.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/emotion-distribution-w-rati
ng
```

Took 2 sec. Last updated by anonymous at December 08 2024, 6:59:47 PM. (outdated)

Keyword Distribution by Emotions

FINISHED

Took 0 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:08:27 PM.

```
import org.apache.spark.sql.functions._

val basePath = "/user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word"
val specificEmotion = "sadness"

val emotionDf = spark.read.parquet(s"$basePath/emotion=$specificEmotion")
```

SPARK JOB FINISHED

```
val keywordDistribution = emotionDf.groupBy("word")
  .count()
  .orderBy(desc("count"))



val rowCount = keywordDistribution.count()
keywordDistribution.show(rowCount.toInt, truncate = false)
```

```
+--------------------+-----+
|word                |count|
+--------------------+-----+
|good                |4971 |
|great               |3333 |
|bad                 |3303 |
|love                |3143 |
|boring              |2861 |
|best                |2286 |
|fun                 |1753 |
|interesting         |1165 |
|worst               |1156 |
|family              |1149 |
|original            |990  |
|amazing             |973  |
|terrible            |965  |
|perfect             |902  |
|beautiful           |897  |
```

Took 43 sec. Last updated by anonymous at December 08 2024, 7:03:52 PM. (outdated)

```
val outputPath = s"/user/yc7093_nyu_edu/imdb-emotion-analysis/$specificEmotion-keyword-distribution"

keywordDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/sadness-keyword-distributio
n
```

Took 19 sec. Last updated by anonymous at December 08 2024, 7:04:11 PM. (outdated)

```
import org.apache.spark.sql.functions._

val basePath = "/user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word"
val specificEmotion = "love"

val emotionDf = spark.read.parquet(s"$basePath/emotion=$specificEmotion")

val keywordDistribution = emotionDf.groupBy("word")
```

SPARK JOB FINISHED

```
  .count()
  .orderBy(desc("count"))


val rowCount = keywordDistribution.count()
keywordDistribution.show(rowCount.toInt, truncate = false)
```

```
+--------------------+-----+
|word                |count|
+--------------------+-----+
|love                |12167|
|good                |7470 |
|great               |6234 |
|best                |4045 |
|fun                 |3664 |
|bad                 |3074 |
|amazing             |2278 |
|beautiful           |1974 |
|perfect             |1952 |
|funny               |1886 |
|family              |1723 |
|interesting         |1646 |
|original            |1596 |
|comedy              |1495 |
|excellent           |1201 |
```

Took 51 sec. Last updated by anonymous at December 08 2024, 7:05:42 PM. (outdated)

```
val outputPath = s"/user/yc7093_nyu_edu/imdb-emotion-analysis/$specificEmotion-keyword-distribution"

keywordDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/love-keyword-distribution
```

Took 21 sec. Last updated by anonymous at December 08 2024, 7:06:03 PM. (outdated)

```
import org.apache.spark.sql.functions._

val basePath = "/user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word"
val specificEmotion = "joy"

val emotionDf = spark.read.parquet(s"$basePath/emotion=$specificEmotion")

val keywordDistribution = emotionDf.groupBy("word")
  .count()
  .orderBy(desc("count"))


val rowCount = keywordDistribution.count()
```

SPARK JOB FINISHED

```
|visually_stunning   |392  |
|action_flick        |388  |
|die_hard            |382  |
|absolutely_amazing  |358  |
|good_fun            |315  |
|cult_classic        |303  |
|horror_flick        |295  |
|rom_com             |260  |
|best_action         |250  |
|hidden_gem          |226  |
|best_horror         |198  |
|rip_off             |177  |
|great_family        |158  |
|wow_wow             |45   |
|waste_money         |21   |
|hear_warming        |1    |
+--------------------+-----+
```

Took 1 min 5 sec. Last updated by anonymous at December 08 2024, 7:07:08 PM. (outdated)

```
val outputPath = s"/user/yc7093_nyu_edu/imdb-emotion-analysis/$specificEmotion-keyword-distribution"

keywordDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/joy-keyword-distribution
```

Took 29 sec. Last updated by anonymous at December 08 2024, 7:07:37 PM. (outdated)

```
import org.apache.spark.sql.functions._

val basePath = "/user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word"
val specificEmotion = "surprise"

val emotionDf = spark.read.parquet(s"$basePath/emotion=$specificEmotion")

val keywordDistribution = emotionDf.groupBy("word")
  .count()
  .orderBy(desc("count"))


val rowCount = keywordDistribution.count()
keywordDistribution.show(rowCount.toInt, truncate = false)
```

SPARK JOB FINISHED

```
+--------------------+-----+
|word                |count|
+--------------------+-----+
|fun                 |4469 |
```

```
|funny                |3913 |
|good                 |3836 |
|great                |3282 |
|love                 |2655 |
|best                 |2139 |
|interesting          |1911 |
|amazing              |1751 |
|bad                  |1480 |
|comedy               |1000 |
|perfect              |923  |
|original             |763  |
|beautiful            |732  |
|brilliant            |683  |
```

Took 39 sec. Last updated by anonymous at December 08 2024, 7:08:16 PM. (outdated)

```
val outputPath = s"/user/yc7093_nyu_edu/imdb-emotion-analysis/$specificEmotion-keyword-distribution"

keywordDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/surprise-keyword-distribution
```

Took 17 sec. Last updated by anonymous at December 08 2024, 7:08:33 PM. (outdated)

```
import org.apache.spark.sql.functions._

val basePath = "/user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word"
val specificEmotion = "anger"

val emotionDf = spark.read.parquet(s"$basePath/emotion=$specificEmotion")

val keywordDistribution = emotionDf.groupBy("word")
  .count()
  .orderBy(desc("count"))



val rowCount = keywordDistribution.count()
keywordDistribution.show(rowCount.toInt, truncate = false)
```

SPARK JOB FINISHED

```
+--------------------+-----+
|word                |count|
+--------------------+-----+
|good                |23585|
|bad                 |17137|
|great               |14953|
|love                |12776|
```

```
|best                |11210|
|fun                 |10915|
|worst               |6221 |
|interesting         |5674 |
|family              |5450 |
|funny               |5449 |
|original            |5293 |
|boring              |4993 |
|horror              |4899 |
|comedy              |4709 |
```

Took 35 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:28:13 PM.

```
val outputPath = s"/user/yc7093_nyu_edu/imdb-emotion-analysis/$specificEmotion-keyword-distribution"

keywordDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

SPARK JOB FINISHED

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/anger-keyword-distribution
```

Took 15 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:28:28 PM.

```
import org.apache.spark.sql.functions._

val basePath = "/user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word"
val specificEmotion = "fear"

val emotionDf = spark.read.parquet(s"$basePath/emotion=$specificEmotion")

val keywordDistribution = emotionDf.groupBy("word")
  .count() // Count occurrences of each word within the emotion
  .orderBy(desc("count"))


val rowCount = keywordDistribution.count()
keywordDistribution.show(rowCount.toInt, truncate = false)
```

SPARK JOB FINISHED

```
+--------------------+-----+
|word                |count|
+--------------------+-----+
|good                |3629 |
|great               |2370 |
|love                |1947 |
|bad                 |1868 |
|fun                 |1831 |
|best                |1657 |
|horror              |1618 |
|interesting         |1191 |
|funny               |1096 |
|family              |846  |
```

```
|original            |813  |
|perfect             |766  |
|comedy              |677  |
|amazing             |668  |
```

Took 36 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:29:04 PM.

SPARK JOB (http://nyu-dataproc-w-0.c.hpc-dataproc-19b8.internal:45771/jobs/job?id=82) SPARK JOB FINISHED

99%

```
keywordDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)
```

```
outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/fear-keyword-distribution
```

Took 9 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:29:13 PM.
Started a few seconds ago.

Rating Distribution Within Keywords

FINISHED

Took 0 sec. Last updated by yc7093_nyu_edu at December 10 2024, 2:10:04 PM.

SPARK JOB FINISHED

```
import org.apache.spark.sql.functions._

val basePath = "/user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word"

val partitionedDf = spark.read.parquet(basePath)

val ratingDistributionByKeyword = partitionedDf.groupBy("word", "rating")
  .count()
  .orderBy("word", "rating")

val pivotedDistribution = ratingDistributionByKeyword.groupBy("word")
  .pivot("rating")
  .sum("count")

pivotedDistribution.show(truncate = false)
```

```
|awesome             |238 |137 |151 |163 |277 |408  |752  |1374 |1864 |5086 |
|good                |6946|3921|4578|5372|7583|10765|15179|15780|11516|17553|
|wonderful           |205 |131 |190 |205 |336 |573  |1256 |2010 |2155 |4388 |
|funny               |2148|1078|1141|1228|1469|2038 |2840 |3546 |2786 |4629 |
|disappointing       |740 |507 |659 |764 |824 |659  |466  |318  |192  |172  |
|absolutely_amazing  |8   |4   |2   |2   |9   |10   |25   |51   |76   |396  |
|comedy              |1387|735 |842 |922 |1281|1959 |2568 |2832 |2178 |3734 |
|romantic_comedy     |33  |27  |40  |55  |55  |111  |149  |167  |91   |126  |
|well_worth          |7   |3   |3   |10  |32  |148  |380  |610  |422  |371  |
|pleasantly_surprised|3   |7   |4   |13  |27  |106  |219  |323  |206  |215  |
```

```
|pleasantly_surprised|3   |7   |4   |15  |27  |106  |219  |323  |206  |213  |
|horror_flick        |33  |31  |51  |39  |62  |89   |109  |114  |53   |48   |
+--------------------+----+----+----+----+----+-----+-----+-----+-----+-----+
only showing top 20 rows

import org.apache.spark.sql.functions._
basePath: String = /user/yc7093_nyu_edu/imdb_partitioned_by_emotion_rating_word
partitionedDf: org.apache.spark.sql.DataFrame = [review_id: string, movie: string ... 6 mor
 fields]
```

Took 4 min 34 sec. Last updated by anonymous at December 08 2024, 6:48:21 PM. (outdated)

```
val outputPath = "/user/yc7093_nyu_edu/imdb-emotion-analysis/rating-distribution-w-word"

pivotedDistribution.coalesce(1)
  .write
  .mode("overwrite")
  .option("header", "true")
  .csv(outputPath)


outputPath: String = /user/yc7093_nyu_edu/imdb-emotion-analysis/rating-distribution-w-word
```

SPARK JOB FINISHED

Took 2 min 8 sec. Last updated by anonymous at December 08 2024, 7:03:09 PM. (outdated)