Public_conversation

```
%md
# About Dataset
Originally we were planning to scrap conversation data from [Yahoo finance](https://finance.yahoo.com/quote/AA/community). The conversation data is r
time. After 20 hours running on local machine, only about 30MB of data has been generated. I also tried to run the script on a CIMS server, but Googl
considering time and data size, I obtained the Stockwits dataset(about 194.3MB) collected by an Udacity Team as an alternative option. The dataset co
those messages are similar to posts on twitter. This dataset is available in the pulic domain and contains sufficient data. More detailed description
[here](https://vkontech.com/sentiment-analysis-of-stocktwits-messages-using-lstm-in-pytorch/).
```

Exploratory data analysis & cleansing

Here, I created a schema for the dataframe, called z.show() to present some rows of the dataset. In total, there are 4 columns, 1548010 rows. Column names and types are shown in the printSchema output.

```
val filePath = "project/comments.csv"
val schema = "index STRING, message_body STRING, sentiment INT, timestamp TIMESTAMP"
val rawDF = spark.read.schema(schema)
  .option("header", "true")
  .option("multiLine", "true")
  .option("inferSchema", "true")
  .option("escape", "\"")
  .csv(filePath)
z.show(rawDF)
```

SPARK JOB (http://nyu-dataproc-w-1.c.hpc-dataproc-19b8.internal:34185/jobs/job?id=1) FINISHED

settings

index	message_body	sentiment
0	$FITB great buy at 26.00...ill wait	2
1	@StockTwits $MSFT	1
2	#STAAnalystAlert for $TDG : Jefferies Maintains with a rating of Hold setting target price at USD 350.00. Our own verdict is Buy http://www.stocktargetadvisor.com/toprating	2
3	$AMD I heard there's a guy who knows someone who thinks somebody knows something - on StockTwits.	1
4	$AMD reveal yourself!	0
5	$AAPL Why the drop? I warren Buffet taking out his	1

Output is truncated to 102400 bytes. Learn more about **ZEPPELIN_INTERPRETER_OUTPUT_LIMIT**

×

Took 0 sec. Last updated by anonymous at December 12 2022, 1:16:09 AM.

```
rawDF.printSchema
```

```
root
 |-- index: string (nullable = true)
 |-- message_body: string (nullable = true)
 |-- sentiment: integer (nullable = true)
 |-- timestamp: timestamp (nullable = true)
```

```
val filePath2 = "project/output.csv"
```

```
filePath2: String = project/output.csv
```

```
val rawDF2 = spark.read.csv(filePath2)
val filtered = rawDF2.filter(rawDF2("_c1") =!= "Symbols").cache()
z.show(filtered)
```

settings

_c0 ▲	_c1
0	$FITB
1	$MSFT
10	$SBUX
100	$BAC

1000	$TGT
1001	$KMX
1002	$GOOGL
1003	$FCX

Public_conversation

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

×

```
rawDF2: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string]
filtered: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [_c0: string, _c1: string]
```

```
filtered.printSchema
```

```
root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
```

```
val joinedDF = rawDF.join(broadcast(filtered), rawDF("index") === filtered("_c0"))
```

```
joinedDF: org.apache.spark.sql.DataFrame = [index: string, message_body: string ... 4 more fields]
```

```
z.show(joinedDF)
```

settings ▾

index	message_body	sentiment	timestamp	_c0
170	short ratio of $ZION is 14.78 at 2018-06-15 and short % to float is 13.89% http://sunshineavenue.com/stock/ZION/ via @sunshineave	-2	2018-07-01 02:33:36.0	170
14	[BREAKOUT Strategy] Current Portfolio : $ZBRA,$WEB,$TIF,$SRE,$SPPI,$OTEX,$OMF,$NVGS,$NRCIB,$MSG,$KIRK,$GHDX,$FBNK,$ESND,$DRI,$DKS,$CVTI,$CV	2	2018-07-01 00:14:19.0	14
606	short ratio of $XRAY is 2.29 at 2018-06-15 and short % to float is 2.81%	-2	2018-07-01 11:45:36.0	606

Output is truncated to 102400 bytes. Learn more about **ZEPPELIN_INTERPRETER_OUTPUT_LIMIT**

×

```
joinedDF.printSchema
```

```
root
 |-- index: string (nullable = true)
 |-- message_body: string (nullable = true)
 |-- sentiment: integer (nullable = true)
 |-- timestamp: timestamp (nullable = true)
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
```

```
val newDF = joinedDF.select($"index", $"message_body", $"sentiment", $"timestamp", split(col("_c1"),",").alias("list_of_stocks"))
```

```
newDF: org.apache.spark.sql.DataFrame = [index: string, message_body: string ... 3 more fields]
```

```
z.show(newDF)
```

settings ▾

index	message_body	sentiment	timestamp
0	$FITB great buy at 26.00...ill wait	2	2018-07-01 00:00:09.0
1	@StockTwits $MSFT	1	2018-07-01 00:00:42.0
2	#STAAnalystAlert for $TDG : Jefferies Maintains with a rating of Hold setting target	2	2018-07-01 00:01:24.0

Public_conversation

	price at USD 350.00. Our own verdict is Buy http://www.stocktargetadvisor.com/toprating		
3	$AMD I heard there’s a guy who knows someone who thinks somebody knows something - on StockTwits.	1	2018-07-01 00:01:47.0
4	$AMD reveal yourself!	0	2018-07-01 00:02:13.0

Output is truncated to 102400 bytes. Learn more about **ZEPPELIN_INTERPRETER_OUTPUT_LIMIT** ×

```
import scala.collection.mutable.WrappedArray
val convert_list = udf((values: WrappedArray[String])=> {
    values.toList})
```

```
import scala.collection.mutable.WrappedArray
convert_list: org.apache.spark.sql.expressions.UserDefinedFunction = SparkUserDefinedFunction($Lambda$4440/1186732433@7c3fdee7,ArrayType(StringType,tr
ue),List(Some(class[value[0]: array<string>])),Some(class[value[0]: array<string>]),None,true,true)
```

```
val converted = newDF.withColumn("list_of_symbols", convert_list(col("list_of_stocks")))
                    .withColumn("index", col("index"))
                    .withColumn("message_body", col("message_body"))
                    .withColumn("sentiment", col("sentiment"))
                    .withColumn("timestamp", col("timestamp"))
z.show(converted)
```

settings ▾

index	message_body	sentiment	timestamp	list_of_stoc
0	$FITB great buy at 26.00...ill wait	2	2018-07-01 00:00:09.0	WrappedArr
1	@StockTwits $MSFT	1	2018-07-01 00:00:42.0	WrappedArr
2	#STAAnalystAlert for $TDG : Jefferies Maintains with a rating of Hold setting target price at USD 350.00. Our own verdict is Buy http://www.stocktargetadvisor.com/toprating	2	2018-07-01 00:01:24.0	WrappedArr
3	$AMD I heard there’s a guy who knows someone who thinks somebody knows something - on	1	2018-07-01 00:01:47.0	WrappedArr

Output is truncated to 102400 bytes. Learn more about **ZEPPELIN_INTERPRETER_OUTPUT_LIMIT** ×

```
val flatted = converted.select($"list_of_symbols",$"index", $"message_body", $"sentiment", to_date($"timestamp").alias("timestamp"), explode($"list_o
```

```
flatted: org.apache.spark.sql.DataFrame = [list_of_symbols: array<string>, index: string ... 4 more fields]
```

```
z.show(flatted)
```

settings ▾

list_of_symbols	index	message_body	sentiment	timestamp
WrappedArray($FITB)	0	$FITB great buy at 26.00...ill wait	2	2018-07-01
WrappedArray($MSFT)	1	@StockTwits $MSFT	1	2018-07-01
WrappedArray($TDG)	2	#STAAnalystAlert for $TDG : Jefferies Maintains with a rating of Hold setting target price at USD 350.00. Our own verdict is Buy http://www.stocktargetadvisor.com/toprating	2	2018-07-01
WrappedArray($AMD)	3	$AMD I heard there’s a guy who knows someone who thinks somebody knows something - on	1	2018-07-01

Output is truncated to 102400 bytes. Learn more about **ZEPPELIN_INTERPRETER_OUTPUT_LIMIT** ×

Public_conversation

```
val groupedDF = flatted.groupBy("flatted_symbol", "timestamp").agg(avg("sentiment"))
z.show(groupedDF)
```

settings ▾

flatted_symbol	timestamp	avg(sen☰m
$FB	2018-07-01	0.37647058
$ACN	2018-07-01	-0.3333333
$JCP	2018-07-01	0.5
$WHR	2018-07-01	0.0
$BXP	2018-07-01	0.0
$IRBT	2018-07-01	0.0
$ECL	2018-07-01	0.0
$TWX	2018-07-01	-1.0

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult** ×

```
groupedDF: org.apache.spark.sql.DataFrame = [flatted_symbol: string, timestamp: date ... 1 more field]
```

```
val removeDF = groupedDF
                .withColumn("flatted_symbol", regexp_replace(col("flatted_symbol"), "\\$", ""))
z.show(removeDF)
```

settings ▾

flatted_symbol	timestamp	avg(sen☰m
FB	2018-07-01	0.37647058
ACN	2018-07-01	-0.3333333
JCP	2018-07-01	0.5
WHR	2018-07-01	0.0
BXP	2018-07-01	0.0
IRBT	2018-07-01	0.0
ECL	2018-07-01	0.0
TWX	2018-07-01	-1.0

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult** ×

```
removeDF: org.apache.spark.sql.DataFrame = [flatted_symbol: string, timestamp: date ... 1 more field]
```

```
val dfWithWeekNumber = removeDF.withColumn("dayOfWeek", date_format(col("timestamp"), "E"))
val df4 = dfWithWeekNumber.withColumn("shiftedDate", when( col("dayOfWeek") === "Sat", date_add(col("timestamp"),2))
.when(col("dayOfWeek") === "Sun", date_add(col("timestamp"),1))
.otherwise(col("timestamp")))
z.show(df4)
```

settings ▾

flatted_symbol	timestamp	avg(sentiment)	dayOfWeek
FB	2018-07-01	0.3764705882352941	Sun
ACN	2018-07-01	-0.3333333333333333	Sun
JCP	2018-07-01	0.5	Sun
WHR	2018-07-01	0.0	Sun
BXP	2018-07-01	0.0	Sun
IRBT	2018-07-01	0.0	Sun
ECL	2018-07-01	0.0	Sun
TWX	2018-07-01	-1.0	Sun

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

Public_conversation

```
dfWithWeekNumber: org.apache.spark.sql.DataFrame = [flatted_symbol: string, timestamp: date ... 2 more fields]
df4: org.apache.spark.sql.DataFrame = [flatted_symbol: string, timestamp: date ... 3 more fields]
```

```
val nflx = df4.filter(col("flatted_symbol") === "NFLX").select(col("shiftedDate"), col("avg(sentiment)")).sort(col("shiftedDate"))
```

```
nflx: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [shiftedDate: date, avg(sentiment): double]
```

```
z.show(nflx)
```

settings ▾

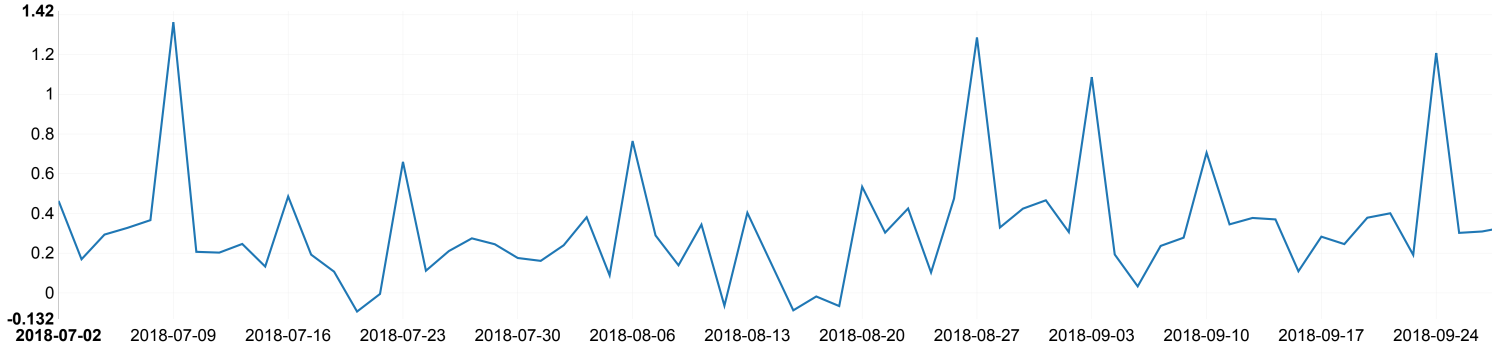
shiftedDate	avg(sentiment)
2018-07-02	0.2715827338129496
2018-07-02	0.1917808219178082
2018-07-03	0.16939890710382513
2018-07-04	0.29365079365079366
2018-07-05	0.327455919395466
2018-07-06	0.3662551440329218
2018-07-09	0.569620253164557
2018-07-09	0.37643207855973815

```
// val finalDF = nflx.withColumn("date", to_date($"shiftedDate"))
//                 .withColumn("flatted_symbol", $"flatted_symbol")
//                 .withColumn("sentiment", $"avg(sentiment)")
//                 .withColumn("dayOfWeek", $"flatted_symbol")
//                 .withColumn("flatted_symbol", $"flatted_symbol")
```

```
finalDF: org.apache.spark.sql.DataFrame = [flatted_symbol: string, shiftedDate: date ... 4 more fields]
```

```
z.show(nflx)
```

settings ▾



```
rawDF.columns.length
```

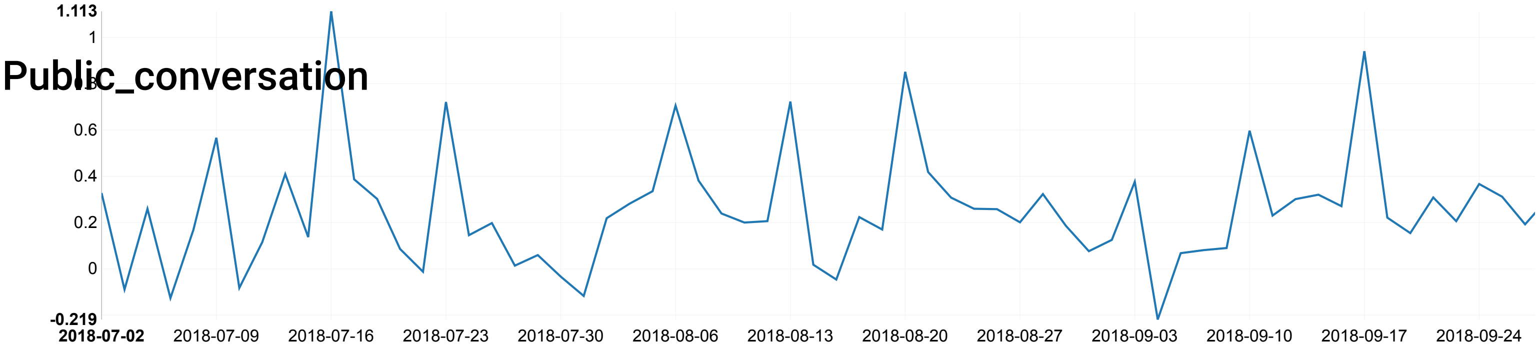
```
res62: Int = 4
```

```
val TSLA = df4.filter(col("flatted_symbol") === "TSLA").select(col("shiftedDate"), col("avg(sentiment)")).sort(col("shiftedDate"))
```

```
TSLA: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [shiftedDate: date, avg(sentiment): double]
```

```
z.show(TSLA)
```

settings ▾


Public_conversation

```
val outputPath = "project/cleanedComments.csv"
df4.write.mode("overwrite").csv(outputPath)
```

```
outputPath: String = project/cleanedComments.csv
```