```
%pyspark
rdd = sc.textFile('s3://megadados-alunos/dados/all_reviews_clean_tsv/')
```

FINISHED

Took 0 sec. Last updated by anonymous at December 12 2021, 6:47:33 PM.

```
%pyspark

column_names = ["marketplace","customer_id", "review_id","product_id","product_parent","pr
    ,"verified_purchase","review_headline","review_body","review_date"]

df = spark.read.option("header", "false").option("delimiter", "\t").csv("s3://megadados-alu
df = df \
    .withColumnRenamed("_c0", column_names[0])\
    .withColumnRenamed("_c1", column_names[1])\
    .withColumnRenamed("_c2", column_names[2])\
    .withColumnRenamed("_c3", column_names[3])\
    .withColumnRenamed("_c4", column_names[4])\
    .withColumnRenamed("_c5", column_names[5])\
    .withColumnRenamed("_c6", column_names[6])\
    .withColumnRenamed("_c7", column_names[7])\
    .withColumnRenamed("_c8", column_names[8])\
    .withColumnRenamed("_c9", column_names[9])\
    .withColumnRenamed("_c10", column_names[10])\
    .withColumnRenamed("_c11", column_names[11])\
    .withColumnRenamed("_c12", column_names[12])\
    .withColumnRenamed("_c13", column_names[13])\
    .withColumnRenamed("_c14", column_names[14])
```

SPARK JOB FINISHED

```
['marketplace', 'customer_id', 'review_id', 'product_id', 'product_parent', 'product_titl
e', 'product_category', 'star_rating', 'helpful_votes', 'total_votes', 'vine', 'verified_pu
rchase', 'review_headline', 'review_body', 'review_date']
```

Took 13 sec. Last updated by anonymous at December 12 2021, 5:57:42 PM. (outdated)

FINISHED

Tarefa 1:

Took 0 sec. Last updated by anonymous at December 12 2021, 7:31:43 PM.

```
%pyspark
count = rdd.count()
print("Tarefa 1: Quantos reviews existem? ---> {0} reviews".format(count))
```

SPARK JOB (http://ip-172-31-61-237.ec2.internal:4040/jobs/job?id=21) FINISHED

```
Tarefa 1: Quantos reviews existem? ---> 150962278 reviews
```

Took 1 min 13 sec. Last updated by anonymous at December 12 2021, 5:59:04 PM. (outdated)

Projeto

```
%pyspark
clientes_existentes = df[["customer_id"]].distinct().count()
print("Tarefa 1: Quantos clientes existem? ---> {0} clientes".format(clientes_existentes))
```

SPARK JOB FINISHED

```
Tarefa 1: Quantos clientes existem? ---> 33497620 reviews
```

Took 1 min 15 sec. Last updated by anonymous at December 12 2021, 6:02:06 PM. (outdated)

SPARK JOB FINISHED

```
%pyspark
produtos = df[["product_id"]].distinct().count()
print("Tarefa 1: Quantos produtos existem? ---> {0} produtos".format(produtos))
```

```
Tarefa 1: Quantos produtos existem? ---> 21390118 produtos
```

Took 1 min 14 sec. Last updated by anonymous at December 12 2021, 6:03:43 PM.

SPARK JOB FINISHED

```
%pyspark
rating = df["star_rating"]
print("Tarefa 1: Quantos reviews existem para cada star rating?:")
df.where((rating == '1') | (rating == '2') | (rating == '3') | (rating == '4') | (rating ==
```

```
Tarefa 1: Quantos reviews existem para cada star rating?:
+-----------+--------+
|star_rating|   count|
+-----------+--------+
|          3|12133772|
|          1|12099424|
|          5|93199322|
|          4|26223155|
|          2| 7304329|
+-----------+--------+
```

Took 1 min 24 sec. Last updated by anonymous at December 12 2021, 6:11:25 PM.

FINISHED

```
%md
# Tarefa 2:
######## Além do conteúdo das aulas, utilizamos a seguinte referência para aprofundar os qu
    -reviews-spot-175430368.html, e vimos que fazer várias reviews no mesmo dia é algo car
```

Tarefa 2:

Além do conteúdo das aulas, utilizamos a seguinte referência para aprofundar os quesitos relevantes na caracterização de bots: https://finance.yahoo.com/news/rise-fake-amazon-reviews-spot-175430368.html, e vimos que fazer várias reviews no mesmo dia é algo característico de bots.

Took 0 sec. Last updated by anonymous at December 12 2021, 7:32:34 PM.

SPARK JOB FINISHED

```
%pyspark

repeat_date_reviews = df.groupBy("customer_id", "product_title", "product_category", "star_
rdr_ordered= repeat_date_reviews.orderBy(["count"], ascending=False)
rdr_filtered= rdr_ordered.filter(((rdr_ordered["count"]) >= 2) )
rdr_filtered_ordered= rdr_filtered.orderBy(["count"], ascending=False)
```

```
216333
```

Took 1 min 53 sec. Last updated by anonymous at December 12 2021, 7:16:55 PM. (outdated)

Projeto

SPARK JOB FINISHED

```
%pyspark
count=rdr_filtered_ordered[["customer_id"]].distinct().count()
print("Número de bots: {}".format(count))
print("Porcentagem de bots: {}%".format((count/clientes_existentes)*100))
```

```
Número de bots: 188747
Porcentagem de bots: 0.5634639117644776%
```

Took 1 min 36 sec. Last updated by anonymous at December 12 2021, 7:29:28 PM.

```
%pyspark
rating = rdr_filtered_ordered["star_rating"]
rdr_filtered_ordered.where((rating == '1') | (rating == '2') | (rating == '3') | (rating ==
```

SPARK JOB FINISHED

```
+-----------+------+
|star_rating| count|
+-----------+------+
|          3| 11478|
|          1| 14841|
|          5|153617|
|          4| 29637|
|          2|  6745|
+-----------+------+
```

Took 1 min 35 sec. Last updated by anonymous at December 12 2021, 7:31:49 PM.

```
%pyspark

rdr_filtered_ordered.groupBy("product_category").count().orderBy(["count"], ascending=Fals
```

SPARK JOB FINISHED

```
|        Pet Products| 7907|
|         Video Games| 7751|
|             Grocery| 7702|
|           Video DVD| 6474|
|              Beauty| 6254|
|            Wireless| 5823|
|          Automotive| 5753|
|                Toys| 5360|
|Health & Personal...| 5114|
|     Office Products| 4957|
|              Sports| 4409|
|                  PC| 4068|
|               Video| 3774|
|Digital_Music_Pur...| 3753|
|         Electronics| 3735|
+--------------------+-----+
only showing top 20 rows
```

Took 1 min 35 sec. Last updated by anonymous at December 12 2021, 7:39:18 PM.

```
%md
# Tarefa 3:
######## Baseado no exemplo disponível em: https://ai.plainenglish.io/build-naive-bayes-spa
```

FINISHED

Tarefa 3:

Projeto

Took 0 sec. Last updated by anonymous at December 12 2021, 7:40:47 PM. (outdated)

```
%pyspark
from pyspark.ml.feature import CountVectorizer
from pyspark.ml.feature import Tokenizer, RegexTokenizer
from pyspark.ml.feature import StringIndexer
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import NaiveBayes
from pyspark.ml import Pipeline
from pyspark.sql.functions import when
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
```

FINISHED

Took 10 min 26 sec. Last updated by anonymous at December 12 2021, 9:00:32 PM.

SPARK JOB (http://ip-172-31-61-237.ec2.internal:4040/jobs/job?id=105) FINISHED

```
%pyspark
naive_bayes = df.select("star_rating","review_body")
df_class =  naive_bayes.withColumn("nb", when(naive_bayes["star_rating"] == "1", "negativo
    "negativo").when(naive_bayes["star_rating"] == "4", "neutro").when(naive_bayes["star_r
df_class.show()
```

```
+-----------+--------------------+--------+
|star_rating|         review_body|      nb|
+-----------+--------------------+--------+
|          4|Dyan Cannon, the ...|  neutro|
|          5|The book was in e...|positivo|
|          3|This book deals w...|negativo|
|          5|I'm still new to ...|positivo|
|          5|Absolutely the mo...|positivo|
|          1|Take this GD book...|negativo|
|          5|This book is FANT...|positivo|
|          1|In his own words:...|negativo|
|          5|Light, very plesa...|positivo|
|          3|It was a nice lit...|negativo|
|          5|Love the scriptur...|positivo|
|          5|Sweet Land of Lib...|positivo|
|          4|In the book Lit b...|  neutro|
|          4|Today's pick is S...|  neutro|
|          5|This is an excell...|positivo|
```

Took 9 sec. Last updated by anonymous at December 12 2021, 8:06:54 PM. (outdated)

SPARK JOB (http://ip-172-31-61-237.ec2.internal:4040/jobs/job?id=113) FINISHED

```
%pyspark
naive_bayes_final = df_class.select("review_body","nb")
naive_bayes_final=naive_bayes_final.na.drop()
naive_bayes_final.show()
```

```
|Take this GD book...|negativo|
|This book is FANT...|positivo|
|In his own words:...|negativo|
|Light, very plesa...|positivo|
|It was a nice lit...|negativo|
|Love the scriptur...|positivo|
|Sweet Land of Lib...|positivo|
|In the book Lit b...|  neutro|
|Today's pick is S...|  neutro|
|This is an excell...|positivo|
|This collection c...|positivo|
|I have used Swork...|negativo|
|This book was a l...|negativo|
|Paolo Bacigalupi ...|positivo|
|I was expecting t...|positivo|
+--------------------+--------+
```

Projeto

```
only showing top 20 rows
```

Took 7 sec. Last updated by anonymous at December 12 2021, 8:24:47 PM.

FINISHED

```
%pyspark
#auxilio do exemplo

stages = []

regexTokenizer = RegexTokenizer(inputCol="review_body", outputCol="tokens", pattern="\\W+"
stages += [regexTokenizer]


cv = CountVectorizer(inputCol="tokens", outputCol="token_features", minDF=2.0)#, vocabSize
stages += [cv]


indexer = StringIndexer(inputCol="nb", outputCol="label")
stages += [indexer]


vecAssembler = VectorAssembler(inputCols=['token_features'], outputCol="features")
stages += [vecAssembler]

[print('\n', stage) for stage in stages]
```

```
 RegexTokenizer_ac4a1bfd5292

 CountVectorizer_00e41641133a

 StringIndexer_b08da211e078

 VectorAssembler_b3353d7742bb
[None, None, None, None]
```

Took 0 sec. Last updated by anonymous at December 12 2021, 8:24:58 PM.

SPARK JOB FINISHED

```
%pyspark

pipeline = Pipeline(stages=stages)
data = pipeline.fit(naive_bayes_final).transform(naive_bayes_final)
```

Took 8 min 4 sec. Last updated by anonymous at December 12 2021, 8:33:05 PM.

FINISHED

```
%pyspark

train, test = data.randomSplit([0.7, 0.3], 2018)
```

Took 0 sec. Last updated by anonymous at December 12 2021, 8:34:46 PM.

Projeto

SPARK JOB FINISHED

```
%pyspark

# Initialise the model
nb = NaiveBayes(smoothing=1.0, modelType="multinomial")
# Fit the model
```

```
model = nb.fit(train)
# Make predictions on test data
predictions = model.transform(test)
predictions.select("label", "prediction", "probability").show()
```

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|  0.0|       2.0|[0.21193880303236...|
|  0.0|       2.0|[6.17455199392373...|
|  2.0|       2.0|[0.16239651487742...|
|  0.0|       0.0|[0.99994862214379...|
|  2.0|       2.0|[2.15400653176401...|
|  0.0|       2.0|[0.46364489092212...|
|  1.0|       2.0|[3.99046376990837...|
|  0.0|       0.0|[0.98964947428947...|
|  0.0|       0.0|[0.93587463903845...|
|  1.0|       2.0|[0.01299592274411...|
|  2.0|       2.0|[3.58384487178370...|
|  0.0|       2.0|[0.17086892143348...|
|  2.0|       2.0|[0.07438118253936...|
|  2.0|       0.0|[0.99242802820056...|
|  0.0|       2.0|[0.00883781343555...|
```

Took 12 min 59 sec. Last updated by anonymous at December 12 2021, 8:49:03 PM.

```
%pyspark

evaluator = BinaryClassificationEvaluator(rawPredictionCol="prediction")
accuracy = evaluator.evaluate(predictions)
print ("Model Accuracy: ", accuracy)
```

SPARK JOB FINISHED

```
Model Accuracy:  0.7417602489074283
```

Took 11 min 16 sec. Last updated by anonymous at December 12 2021, 9:00:32 PM.

```
%pyspark


paramGrid = ParamGridBuilder().addGrid(nb.smoothing, [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2
cvEvaluator = BinaryClassificationEvaluator(rawPredictionCol="prediction")


cv = CrossValidator(estimator=nb, estimatorParamMaps=paramGrid, evaluator=cvEvaluator)
cvModel = cv.fit(train)


cvPredictions = cvModel.transform(test)


evaluator.evaluate(cvPredictions)
```

SPARK JOB FINISHED

```
0.7417920627261595
```

Took 3 hrs 32 min 4 sec. Last updated by anonymous at December 13 2021, 12:34:20 AM.

```
%pyspark
```

READY