# Jordan_features

July 30, 2021

[4]:
```
import pandas as pd

from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import *

from pyspark.sql import SparkSession
from pyspark.sql.functions import col, countDistinct, udf
from pyspark.sql.types import ArrayType, IntegerType,  StringType

from pyspark.ml.feature import Tokenizer
from pyspark.ml.feature import StopWordsRemover
from pyspark.ml.feature import CountVectorizer
from pyspark.ml.feature import Word2Vec
from pyspark.ml.feature import StringIndexer

spark = SparkSession.builder.getOrCreate()
```

[5]:
```
df = spark.read.csv("./jordan_subset", header=True)
```

[6]:
```
df.show(5)
```

```
+------------+-------+-----------+---+---------+----------+------------+--------
--------------+-----------------+--------------+-------+-------------+------+-
----+--------+--------------+-----+-----------+--------------------+----------
---+------+---------------+----------+
|subreddit_id|    _c0|created_utc|ups|  link_id|
name|score_hidden|author_flair_css_class|author_flair_text|      subreddit|
id|removal_reason|gilded|downs|archived|         author|score|retrieved_on|
body|distinguished|edited|controversiality| parent_id|
+------------+-------+-----------+---+---------+----------+------------+--------
--------------+-----------------+--------------+-------+-------------+------+-
----+--------+--------------+-----+-----------+--------------------+----------
---+------+---------------+----------+
|    t5_2te6p|3345542| 1430609172|  4|t3_34mhlr|t1_cqwgwou|           0|
NA|               NA|WaltDisneyWorld|cqwgwou|            NA|     0|    0|
0|    orangekid13|    4|  1432737889|Those helicopters…|           NA|     0|
0|t1_cqw6yd9|
```

```
|    t5_2qh55|3345543| 1430609172|  1|t3_34m80e|t1_cqwgwov|           0|
NA|               NA|           food|cqwgwov|            NA|     0|    0|
0|   colintheking|    1|  1432737889|     OMG.  Mouthgasm|           NA|     0|
0| t3_34m80e|
|    t5_2rfxx|3345544| 1430609172|  2|t3_34lxnr|t1_cqwgwow|           0|
katarina|             null|leagueoflegends|cqwgwow|           NA|     0|    0|
0|   aldothetroll|    2|  1432737889|Jinx is such free…|         null|  null|
null|      null|
|    t5_2qlqh|3345545| 1430609172|  2|t3_34m2d1|t1_cqwgwox|           0|
NA|               NA|        Android|cqwgwox|            NA|     0|    0|
0|Kittycat-banana|    2|  1432737889|That worked! Than…|           NA|     0|
0|t1_cqw5tch|
|    t5_2qh3p|3345547| 1430609172|  1|t3_34mzrt|t1_cqwgwoz|           0|
NA|               NA|            sex|cqwgwoz|            NA|     0|    0|
0|   thatoneguy54|    1|  1432737889|Huh, TIL. How exa…|           NA|     0|
0|t1_cqwgtv6|
+------------+-------+-----------+---+---------+----------+------------+--------
-------------+-----------------+--------------+-------+-------------+------+-
----+--------+---------------+-----+------------+--------------------+----------
---+------+---------------+----------+
only showing top 5 rows
```

[7]:
```
df.count()
```

[7]:
```
8799960
```

[8]:
```
df.columns
```

[8]:
```
['subreddit_id',
 '_c0',
 'created_utc',
 'ups',
 'link_id',
 'name',
 'score_hidden',
 'author_flair_css_class',
 'author_flair_text',
 'subreddit',
 'id',
 'removal_reason',
 'gilded',
 'downs',
 'archived',
 'author',
 'score',
 'retrieved_on',
```

```
 'body',
 'distinguished',
 'edited',
 'controversiality',
 'parent_id']
```

[9]:
```
# tokenizer
# then stop words remover
# then countvectorizer
# word2vec??
```

[10]:
```
tk = Tokenizer(inputCol="body", outputCol="words")
# tokenized = tk.transform(df)
# tokenized.select('body','words').show(5)
```

[11]:
```
sw = StopWordsRemover(inputCol="words", outputCol="filtered")
# removedData = sw.transform(tokenized)
# removedData.select('body','words','filtered').show(5)
```

[12]:
```
cv = CountVectorizer(inputCol="filtered", outputCol="counted", vocabSize=3,␣
↪minDF=2.0)
# counted = cv.fit(removedData)
# counted.select('body','words','filtered','counted').show(5)
```

[13]:
```
w2v =  Word2Vec(vectorSize=3, minCount=0, inputCol="filtered",␣
↪outputCol="word2vec")
# vectored = w2v.fit(removedData)
# vectored.select('body','words','filtered',"word2vec").show(5)
```

[14]:
```
si = StringIndexer(inputCol="subreddit_id", outputCol="sr_id_num")
```

[15]:
```
ohe = OneHotEncoder(inputCol="sr_id_num", outputCol="subr_ohe")
```

[ ]:
```
# feats = ␣
↪['ups','gilded','score_hidden','downs','score','controversiality','counted','word2vec']
feats =  ['ups','gilded','score_hidden','downs','score','controversiality']
vecAs = VectorAssembler(inputCols=feats, outputCol="features")
```

[16]:
```
# add this all to a pipeline before training the model
```

[18]:
```
# pipeline = Pipeline(stages=[tk,sw,cv,w2v,ohe,vecAs])
pipeline = Pipeline(stages=[tk,sw,si,ohe,vecAs])
df_fitted = pipeline.fit(df)
```

[ ]:
```
# Should we do an unsupervised analysis and pass in subreddits? Is there a␣
↪supervised method we can use?
```