影评情感分析

1、数据介绍

数据集 DMSC.csv 收集了来自豆瓣网针对28部电影超过两百万条中文影评，包含影片名、影评文件、打分等信息。

2、实验过程

2.1 准备阶段

2.1.1 环境配置

jupyter notebook + python + spark

准备 spark 环境

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

2.1.2 读取数据并查看

```
df = spark.read.csv('DMSC.csv',header=True, inferSchema=True, escape="\"",
multiLine=True)
```

```
df.show()
```

```
+---+--------------------+------------+----------+------+----------------+------
------+----+--------------------------------------+----+
| ID|       Movie_Name_EN|Movie_Name_CN|Crawl_Date|Number|        Username|
Date|Star|                               Comment|Like|
+---+--------------------+------------+----------+------+----------------+------
------+----+--------------------------------------+----+
|  0|Avengers Age of U...|  复仇者联盟2|2017-01-22|     1|             然潘|2015-05-
13|   3|           连奥创都知道整容要去韩国。|2404|
|  1|Avengers Age of U...|  复仇者联盟2|2017-01-22|     2|       更深的白色|2015-04-
24|   2| 非常失望，剧本完全敷衍了事，主线...|1231|
|  2|Avengers Age of U...|  复仇者联盟2|2017-01-22|     3|    有意识的贱民|2015-04-
26|   2|     2015年度最失望作品。以为面面...|1052|
|  3|Avengers Age of U...|  复仇者联盟2|2017-01-22|     4|  不老的李大爷耶|2015-04-
23|   4|    《铁人2》中勾引钢铁侠，《妇联1...|1045|
|  4|Avengers Age of U...|  复仇者联盟2|2017-01-22|     5|         ZephyrO|2015-
04-22|   2|  虽然从头打到尾，但是真的很无聊啊。| 723|
```

```
|  5|Avengers Age of U...|  复仇者联盟2|2017-01-22|     6|同志亦凡人中文站|2015-04-
22|   3| 剧情不如第一集好玩了，全靠密集笑...| 671|
|  6|Avengers Age of U...|  复仇者联盟2|2017-01-22|     7|           Danny|2015-
04-23|   2| 只有一颗彩蛋必须降一星。外加漫威...| 641|
|  7|Avengers Age of U...|  复仇者联盟2|2017-01-22|     8|           gYroS|2015-
04-28|   2|            看腻了这些打来打去的烂片| 576|
|  8|Avengers Age of U...|  复仇者联盟2|2017-01-22|     9|          tidd熊|2015-04-
23|   3| 漫威粉勿喷，真感觉比第一部差了些...| 481|
|  9|Avengers Age of U...|  复仇者联盟2|2017-01-22|    10|      桃桃淘电影|2015-05-
12|   3| 属于超级英雄的春晚，角色如走马灯...| 443|
| 10|Avengers Age of U...|  复仇者联盟2|2017-01-22|    11|            影志|2015-04-
30|   4| “一个没有黑暗面的人不值得信任。...| 381|
| 11|Avengers Age of U...|  复仇者联盟2|2017-01-22|    12|            玖萬|2015-05-
12|   2| 请漫威华丽地滚出电影界！每年都炮...| 275|
| 12|Avengers Age of U...|  复仇者联盟2|2017-01-22|    13|        亵渎电影|2015-05-
12|   2| 承认这货很烂很难吗？混乱的节奏，...| 231|
| 13|Avengers Age of U...|  复仇者联盟2|2017-01-22|    14|      陀螺凡达可|2015-04-
22|   3| 跟第一部很不一样，叙事加强了不少...| 228|
| 14|Avengers Age of U...|  复仇者联盟2|2017-01-22|    15|      别惹小白兔|2015-04-
27|   3| 漫威第二阶最中庸的一集。承上启下...| 270|
| 15|Avengers Age of U...|  复仇者联盟2|2017-01-22|    16|          高压电|2015-05-
08|   1| 什么破烂反派，毫无戏剧冲突能消耗...| 158|
| 16|Avengers Age of U...|  复仇者联盟2|2017-01-22|    17|      牛腩羊耳朵|2015-04-
22|   4| 总体来说没有达到第一部想让人立马...| 165|
| 17|Avengers Age of U...|  复仇者联盟2|2017-01-22|    18|          文文周|2015-04-
24|   5| 机甲之战超超好看，比变形金刚强；...| 182|
| 18|Avengers Age of U...|  复仇者联盟2|2017-01-22|    19|          抽先桑|2015-04-
29|   2|  结局就差寡姐握着绿巨人的手说：”...| 153|
| 19|Avengers Age of U...|  复仇者联盟2|2017-01-22|    20|      时间的玫瑰|2015-04-
23|   4|       全程挥之不去美队的胸和banne...| 144|
+---+--------------------+-------------+----------+------+----------------+------
----+----+--------------------------------------+----+
only showing top 20 rows
```

2.1.3 提取字段

```
data = df.select('Comment', 'star')
data.head()
```

```
Row(Comment=' 连奥创都知道整容要去韩国。', star=3)
```

2.2 向量化

安装 jieba 分词

```
pip install jieba
```

2.2.1 jieba分词

```
from pyspark.sql.functions import udf
from pyspark.sql.types import *
import jieba

word_udf = udf(lambda x: list("/".join(jieba.cut_for_search(x)).split("/")),
ArrayType(StringType()))
data = data.withColumn('words', word_udf('Comment'))
```

```
data.head()
```

```
Row(Comment=' 连奥创都知道整容要去韩国。', star=3, words=[' ', '连', '奥创', '都', '知
道', '整容', '要', '去', '韩国', '。'])
```

2.2.2 计算词频

```
from pyspark.ml.feature import HashingTF,IDF
hashingTF = HashingTF(inputCol="words", outputCol="tfFeatures")
tf_df = hashingTF.transform(data)

tf_df.head()
```

```
Row(Comment=' 连奥创都知道整容要去韩国。', star=3, words=[' ', '连', '奥创', '都', '知
道', '整容', '要', '去', '韩国', '。'], tfFeatures=SparseVector(262144, {42071:
1.0, 59328: 1.0, 61385: 1.0, 74331: 1.0, 146416: 1.0, 167159: 1.0, 186636: 1.0,
208750: 1.0, 211921: 1.0, 239248: 1.0}))
```

2.2.3 计算IDF值

```
idf = IDF(inputCol="tfFeatures", outputCol="features")
idfModel = idf.fit(tf_df)
idf_df = idfModel.transform(tf_df)

idf_df.head()
```

```
Row(Comment=' 连奥创都知道整容要去韩国。', star=3, words=[' ', '连', '奥创', '都', '知
道', '整容', '要', '去', '韩国', '。'], tfFeatures=SparseVector(262144, {42071:
1.0, 59328: 1.0, 61385: 1.0, 74331: 1.0, 146416: 1.0, 167159: 1.0, 186636: 1.0,
208750: 1.0, 211921: 1.0, 239248: 1.0}), features=SparseVector(262144, {42071:
2.9817, 59328: 4.9502, 61385: 3.3849, 74331: 1.9807, 146416: 0.0, 167159: 3.5969,
186636: 5.249, 208750: 7.9998, 211921: 0.9291, 239248: 6.6791}))
```

2.3 模型训练与评估

2.3.1 构建好样本特征，训练模型

```
trainSet, testSet = idf_df.randomSplit([0.9, 0.1])
```

利用 pyspark.ml.classification 朴素贝叶斯分类器训练模型

```
from pyspark.ml.classification import NaiveBayes
nb = NaiveBayes(featuresCol="features", labelCol='star', smoothing=1.0)
model = nb.fit(trainSet)
```

2.3.2 将模型存到文件系统

```
model.save('PATH')

NaiveBayesModel.load('PATH')
```

```
NaiveBayesModel: uid=NaiveBayes_83dc77734292, modelType=multinomial,
numClasses=5, numFeatures=262144
```

2.3.3 对训练好的模型进行评估

```
result = model.transform(testSet)
result
```

```
DataFrame[Comment: string, star: int, words: array<string>, tfFeatures: vector,
features: vector, rawPrediction: vector, probability: vector, prediction:
double]
```

2.3.4 将预测结果放到文件系统

文本中包含标点符号，需要选择特殊分隔符保存影评信息

```
result.select("Comment", "star", "prediction").write.csv(path='Model', sep="[@")
```

```
df = spark.read.csv('Model/part-00000-e6737d7e-0a5f-445c-8965-cf634f96c92f-
c000.csv',inferSchema=True, escape="\"", multiLine=True, sep="[@")
```

```
df.head()
```

```
Row(_c0='温水煮青蛙的桥段那年看《我想和这个世界谈谈》时印象非常深刻，如今搬进了电影里，播出时整个片场的人都在笑，但我知道他们笑过后，会思考更多。\n　　没有太多的矫饰，用喜剧表达，是这部电影最大的亮点没有之一。', _c1=5, _c2=2.0)
```

2.3.5 将真实评分数据 int 类型转换为和预测值相同 double

```
int2Float_udf = udf(lambda x: float(x))
result_df = df.withColumn("star", int2Float_udf("_c1"))

result_df.head()
```

```
Row(_c0='温水煮青蛙的桥段那年看《我想和这个世界谈谈》时印象非常深刻，如今搬进了电影里，播出时整个片场的人都在笑，但我知道他们笑过后，会思考更多。\n　　没有太多的矫饰，用喜剧表达，是这部电影最大的亮点没有之一。', _c1=5, _c2=2.0, star='5.0')
```

3 结果

查看测试集样本总数

```
result_df.count()
```

```
213160
```

查看准确预测评分

```
result_df.filter("star=_c2").count()
```

```
36162
```

精准预测评分比例不高，进一步还可以进行调参或分词更加细分操作