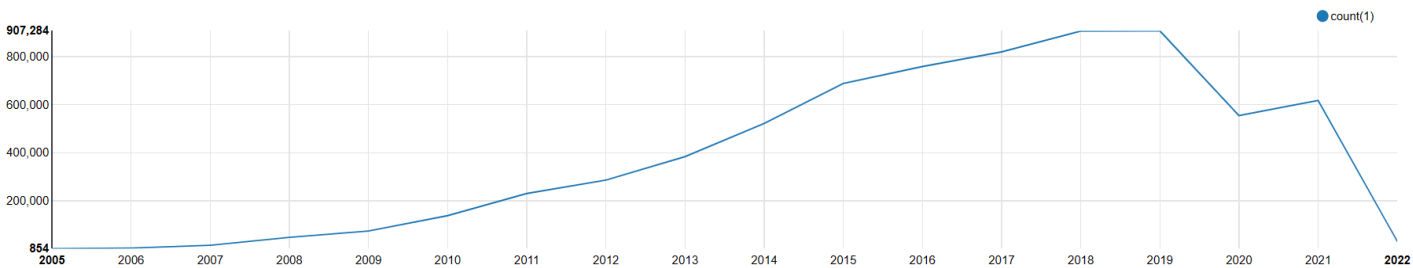
评价统计及分析模块Spark

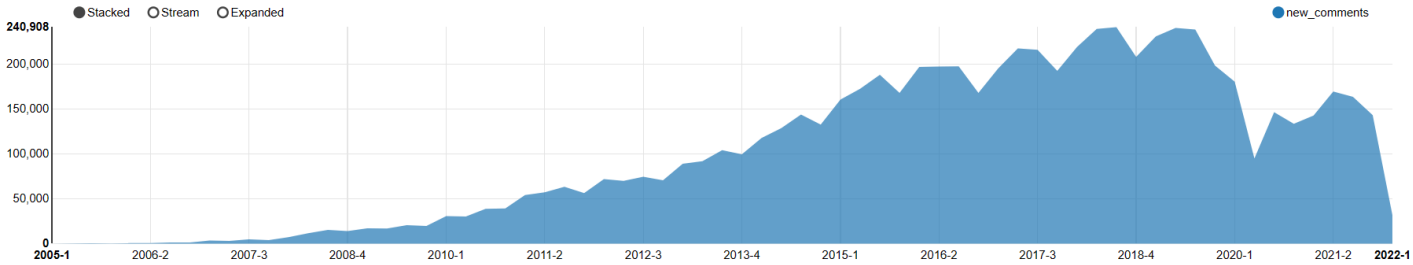
```
%pyspark

#编写者：张子杰
#功能：评论词云图
#时间：2024.1.3


from pyspark import HiveContext
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import pandas as pd


hc=HiveContext(sc)
df=hc.sql("SELECT rev_text from review limit 100000")


wordcloud = WordCloud(width=1600, height=800,
background_color="white").generate(df.toPandas().to_string())

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()

```


```
%pyspark

# 编写者:凌晨
# 功能:分析每年评论的数量
# 时间:2024.1.3
from pyspark.sql import HiveContext

hc = HiveContext(sc)
result = hc.sql("SELECT year(to_date(rev_date)) as year, COUNT(*) AS cot FROM review GROUP BY year(to_date(rev_date)) ORDER BY year DESC")
z.show(result)
```

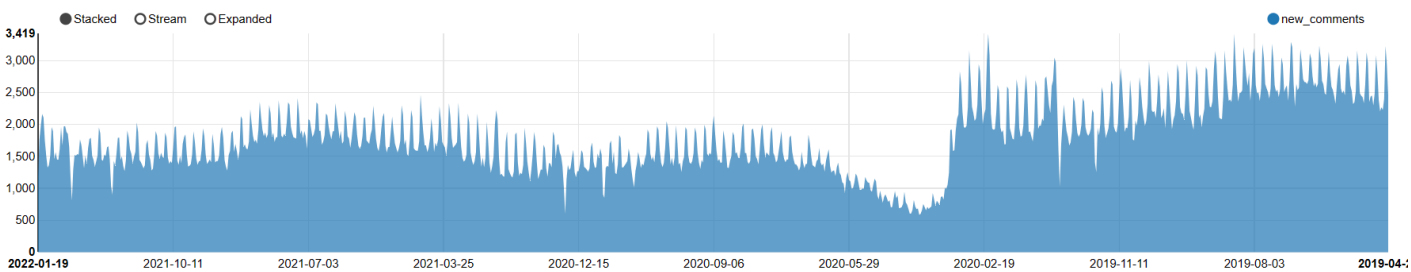



```
%pyspark

# 编写者:凌晨
# 功能:分析每季度评论的数量,每三个月视为一个季度。如2004-4代表2004的第四个季度
```

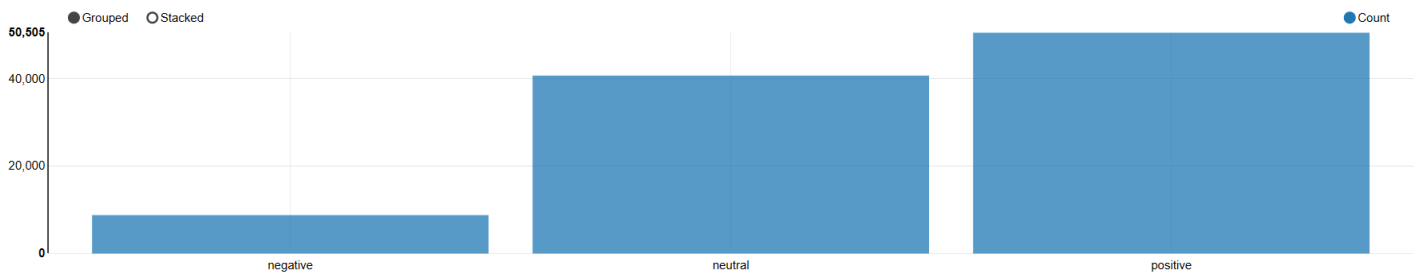
```
# 时间:2024.1.3

hc = HiveContext(sc)
result = hc.sql('''
SELECT
  CONCAT(YEAR(rev_date), '-', CEIL(MONTH(rev_date) / 3)) AS quarter,
  COUNT(*) AS new_comments
FROM review
GROUP BY CONCAT(YEAR(rev_date), '-', CEIL(MONTH(rev_date) / 3))
ORDER BY quarter
''')
z.show(result)
```



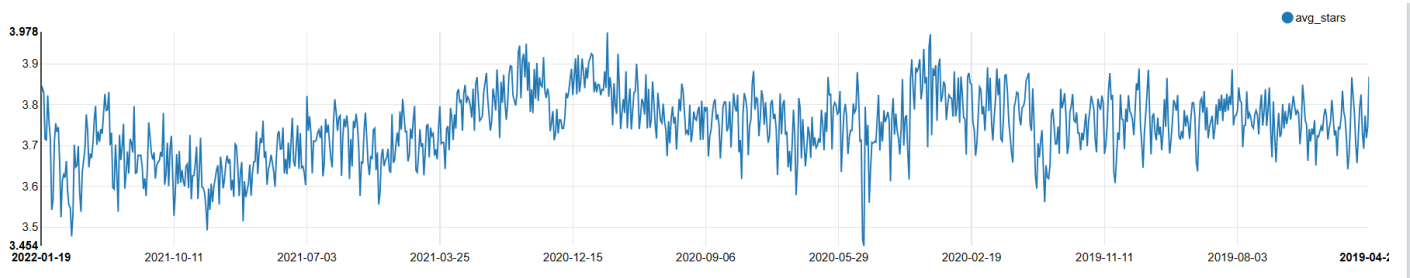

```
%pyspark

# 编写者:凌晨
# 功能:统计按天的评论数目
# 时间:2024.1.3

from pyspark.sql import HiveContext

hc = HiveContext(sc)
result = hc.sql("SELECT DATE(rev_date) AS date, COUNT(*) AS new_comments FROM review GROUP BY DATE(rev_date) ORDER BY date DESC LIMIT 1000")
z.show(result)
```



```pyspark
%pyspark

# 编写者:张子杰
# 功能: 评论情感分析统计
# 时间:2024.1.4

from pyspark import HiveContext
from pyspark.sql.functions import col, udf
from pyspark.sql.types import StringType
from textblob import TextBlob

hc = HiveContext(sc)

df = hc.sql("SELECT rev_text FROM review LIMIT 100000")

def analyze_sentiment(text):
    analysis = TextBlob(str(text))
    # 判断情感极性
    if analysis.sentiment.polarity > 0.15:
        return 'positive'
    elif analysis.sentiment.polarity < -0.15:
        return 'negative'
    else:
        return 'neutral'

df = df.toPandas()

df['sentiment'] = df.apply(analyze_sentiment, axis=1)


result = df.groupby('sentiment').size().reset_index(name='Count')


z.show(result)
```

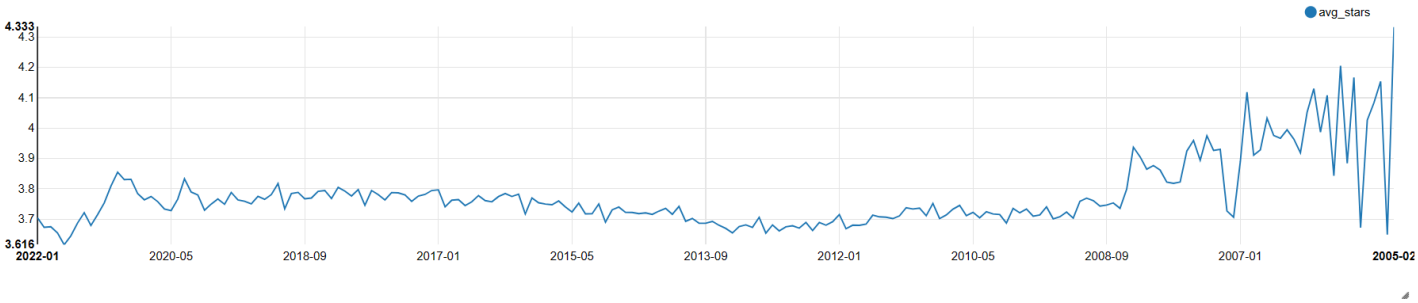


```pyspark
%pyspark

# 编写者：钟鹏
# 功能：统计每天评论分数
# 时间：2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df_review = hcx.table('review')

# 将日期列转换为日期格式，并按日期分组计算平均stars
result = df_review.select('rev_date', 'rev_stars') \
    .withColumn('date', to_date(col('rev_date'))) \
    .groupBy('date') \
    .agg(avg('rev_stars').alias('avg_stars')) \
    .orderBy('date',ascending=False)

z.show(result)
```




```pyspark
%pyspark
# 编写者：钟鹏
# 功能：统计每月评论分数
# 时间：2024.1.4
from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df_review = hcx.table('review')
result_monthly = df_review.select('rev_date', 'rev_stars') \
    .withColumn('year_month', date_format(to_date(col('rev_date')), 'yyyy-MM')) \
    .groupBy('year_month') \
```

```
    .agg(avg('rev_stars').alias('avg_stars')) \
    .orderBy('year_month',ascending=False)

z.show(result_monthly)
```




```
%pyspark
# 编写者：钟鹏
# 功能：统计每年评论分数
# 时间：2024.1.4
from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df_review = hcx.table('review')
result_yearly = df_review.select('rev_date', 'rev_stars') \
    .withColumn('year', year(to_date(col('rev_date')))) \
    .groupBy('year') \
    .agg(avg('rev_stars').alias('avg_stars')) \
    .orderBy('year',ascending=False)

z.show(result_yearly)
```