CS143 Project 2 Part B

Yao Xie 804946717 allenxie@cs.ucla.edu
Kaiyuan Xu 505033984 kyxu@g.ucla.edu

June 6, 2018

1.

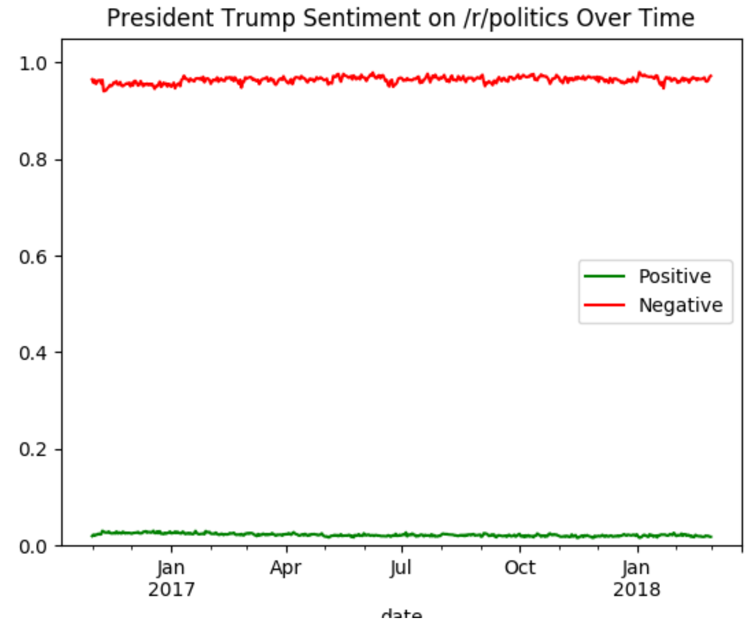


Figure 1: Sentiment over time

2. Parameters *vmin* and *vmax* has been adjusted.

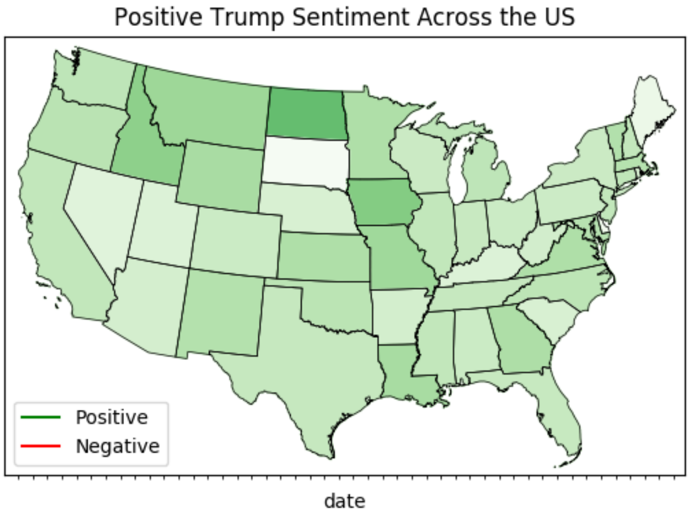


Figure 2: Positive across the US

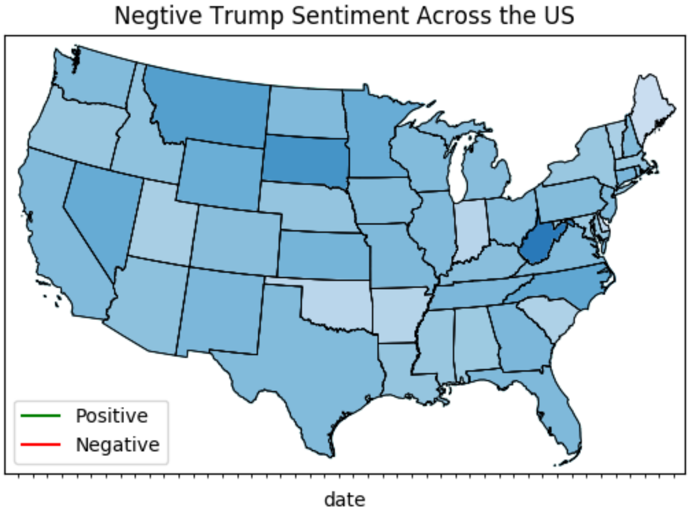


Figure 3: Negative across the US

3.

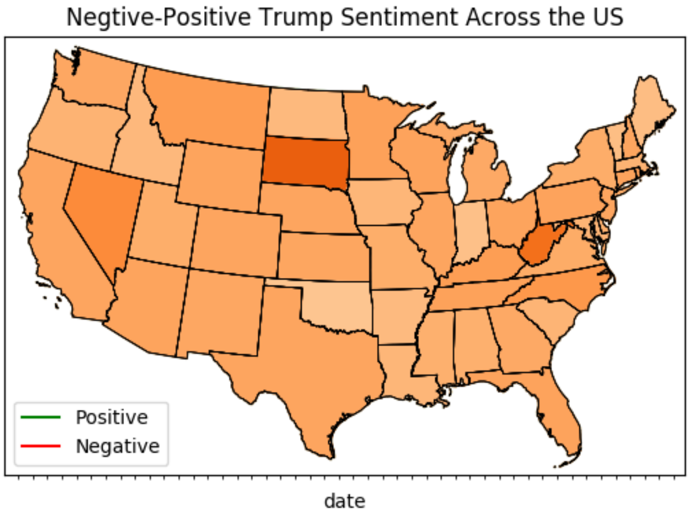
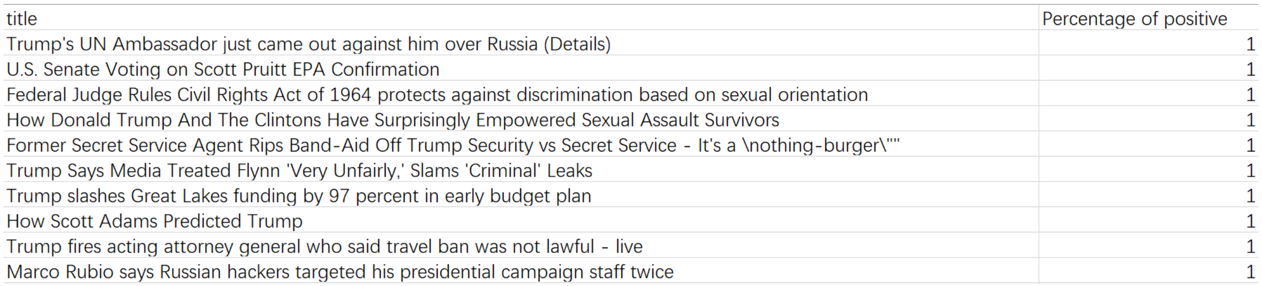


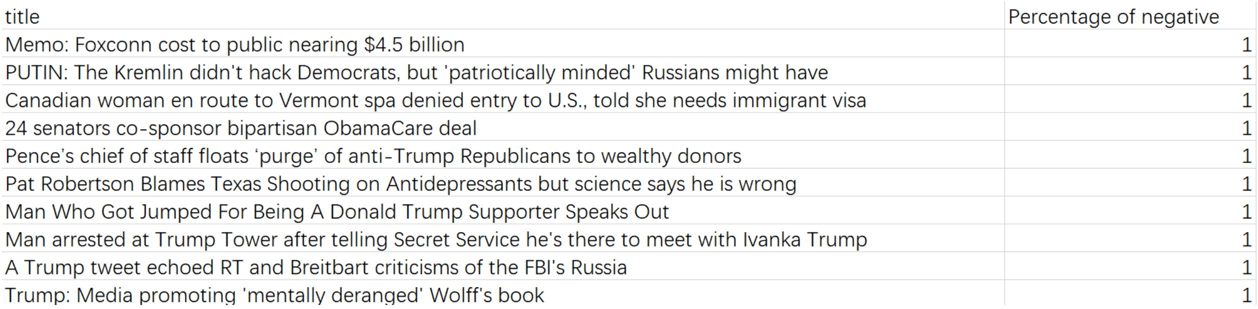
Figure 4: Negative-Positive across the US

4.



title	Percentage of positive
Trump's UN Ambassador just came out against him over Russia (Details)	1
U.S. Senate Voting on Scott Pruitt EPA Confirmation	1
Federal Judge Rules Civil Rights Act of 1964 protects against discrimination based on sexual orientation	1
How Donald Trump And The Clintons Have Surprisingly Empowered Sexual Assault Survivors	1
Former Secret Service Agent Rips Band-Aid Off Trump Security vs Secret Service - It's a \nothing-burger\""	1
Trump Says Media Treated Flynn 'Very Unfairly,' Slams 'Criminal' Leaks	1
Trump slashes Great Lakes funding by 97 percent in early budget plan	1
How Scott Adams Predicted Trump	1
Trump fires acting attorney general who said travel ban was not lawful - live	1
Marco Rubio says Russian hackers targeted his presidential campaign staff twice	1

Figure 5: Positive story



title	Percentage of negative
Memo: Foxconn cost to public nearing $4.5 billion	1
PUTIN: The Kremlin didn't hack Democrats, but 'patriotically minded' Russians might have	1
Canadian woman en route to Vermont spa denied entry to U.S., told she needs immigrant visa	1
24 senators co-sponsor bipartisan ObamaCare deal	1
Pence's chief of staff floats ‘purge' of anti-Trump Republicans to wealthy donors	1
Pat Robertson Blames Texas Shooting on Antidepressants but science says he is wrong	1
Man Who Got Jumped For Being A Donald Trump Supporter Speaks Out	1
Man arrested at Trump Tower after telling Secret Service he's there to meet with Ivanka Trump	1
A Trump tweet echoed RT and Breitbart criticisms of the FBI's Russia	1
Trump: Media promoting 'mentally deranged' Wolff's book	1

Figure 6: Negative story

5.

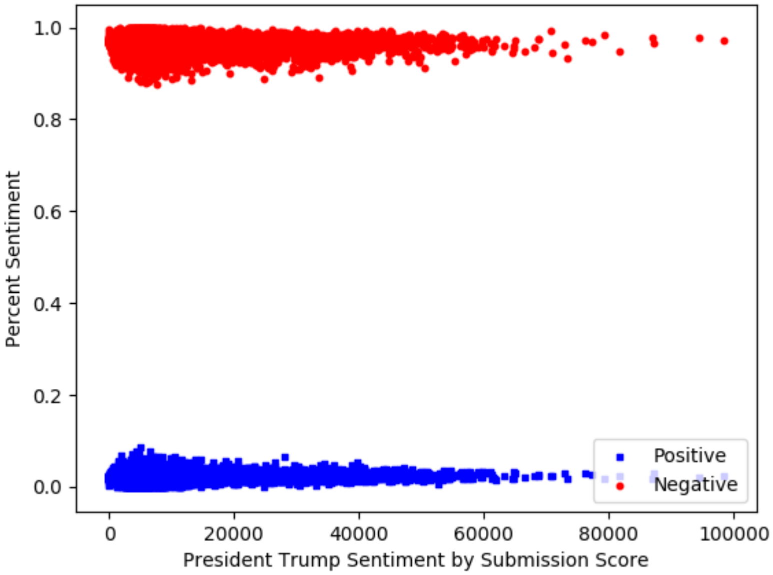


Figure 7: Sentiment by submission score

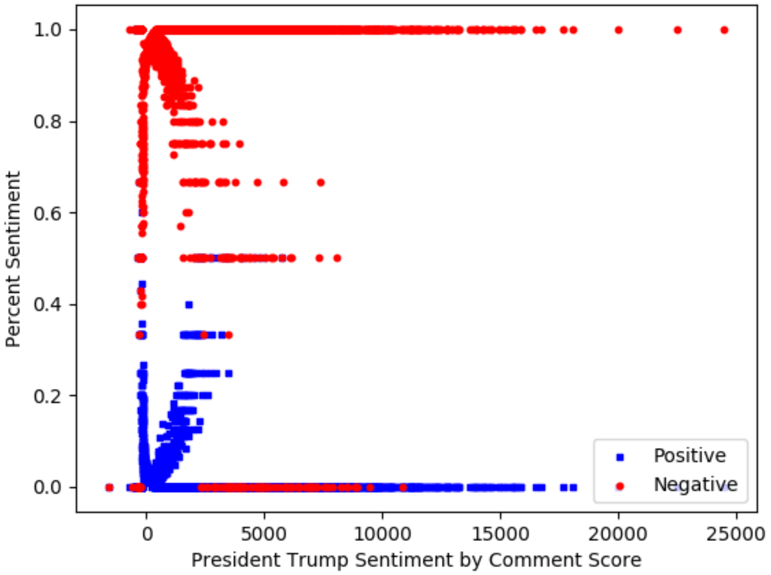


Figure 8: Sentiment by comment score

7. The AUC for negative results is 0.94 and the AUC for positive results is 0.93. From the curve, we can see that the trained model is fairly good.

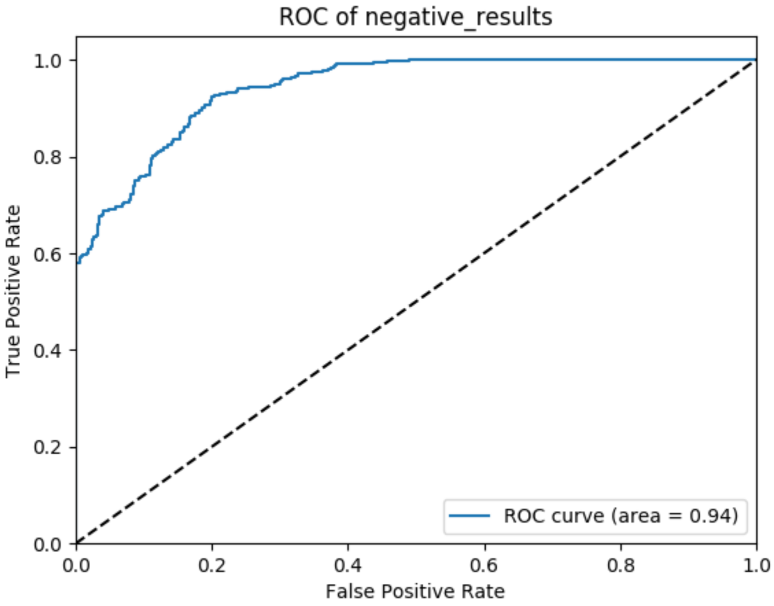


Figure 9: Negative ROC

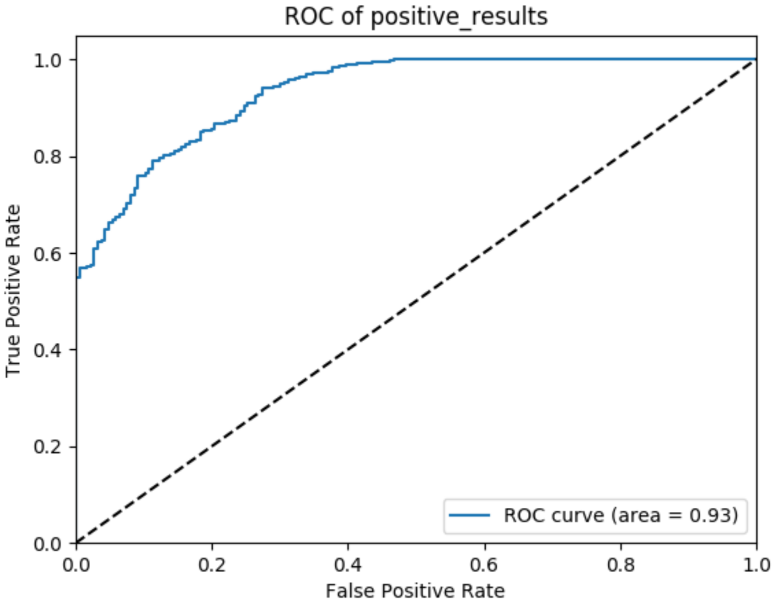


Figure 10: Positive ROC

8.

We can see from Figure 1 that negative comments are significantly more than positive comments. The possible reasons may be that this dataset is biased and /r/politics thinks about President Trump negatively. We can see from Figure 2 and 3 that it varies by state, and from Figure 4 that, the deeper the color is, the more negatively /r/politics thinks about President Trump. From Figure 1 again, we can see that it does not really vary over time, the sentiment is roughly stable. And from Figure 5 and 6, we can see it varies by story, different stories have different sentiments. We can see from Figure 7 and 8 that submission score might be a good feature for classificaition, and comment score could probably not be a good feature for classification.

QUESTION 1.

The functional dependencies in label dataset could be:

$$input\_id \rightarrow \{labeldem, labelgop, labeldjt\}.$$

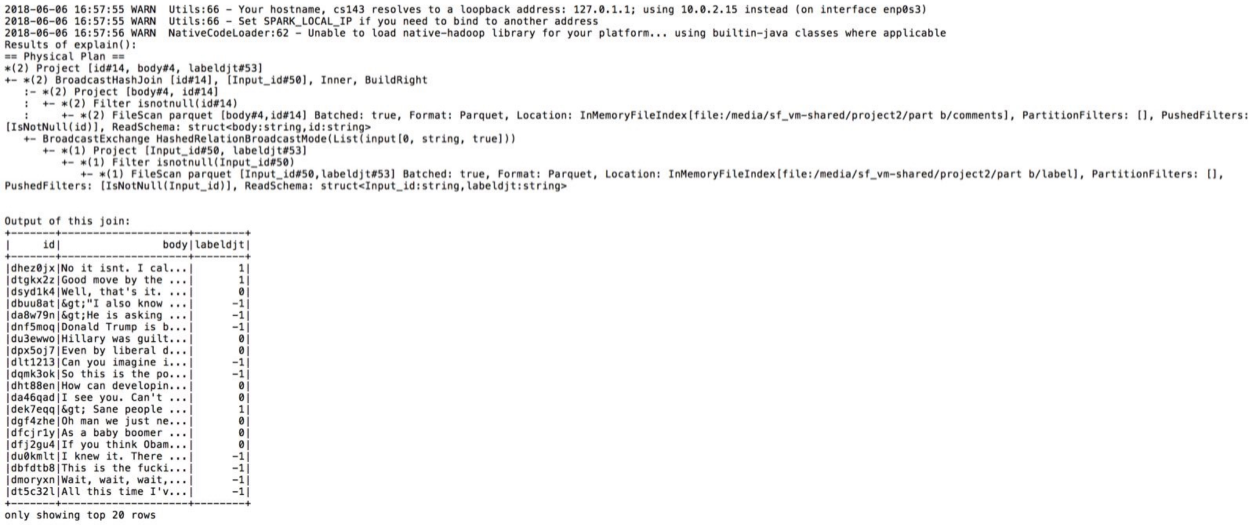
QUESTION 2.

It doesn't look normalized. The existence of columns $\{subreddit, subreddit\_id, subreddit\_type\}$ is redundant since we need to ensure the dependency between $subreddit\_id$ and $subreddit, subreddit\_type$ when inserting/updating. We can decompose the whole table as $R_1 = \{\underline{subreddit\_id}, subreddit, subreddit\_type\}$ with $R_2 = \{\text{all columns without } 'subreddit' \text{ and } 'subreddit\_type'\}$. But by doing so, there may be a foreign key set on 'subreddit_id' and this may cause insert/update integrity. And this can avoid a big join when querying. So maybe that's why the collector stored the data in a whole table.

QUESTION 3.

Figure 12 is part of the codes to generate the join result. And Figure 11 shows the results of join between label and comments in task 3 with explain() and the corresponding output.

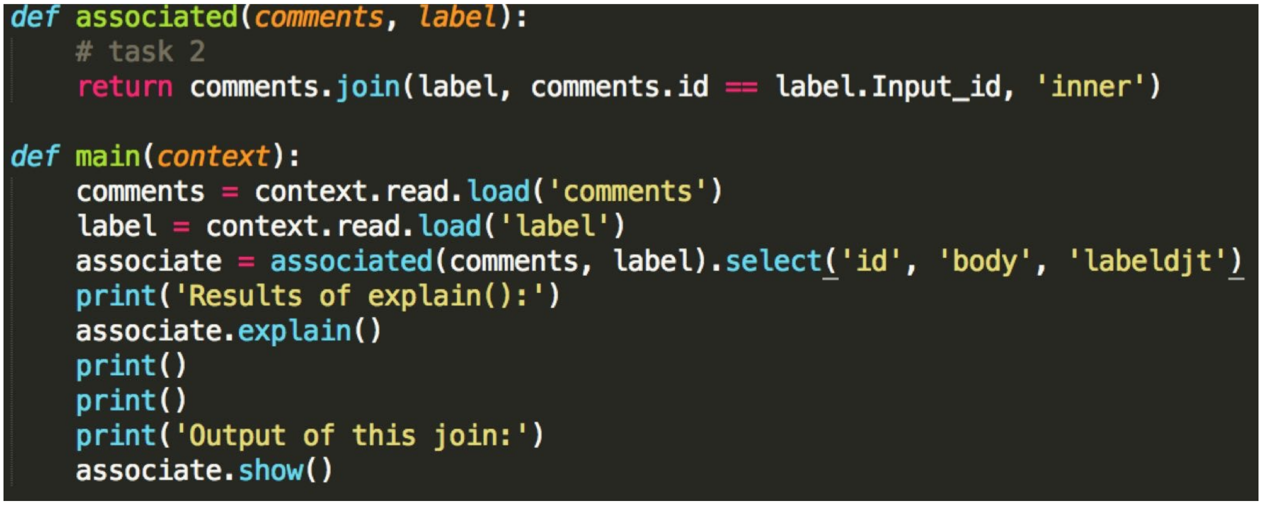
And we noticed that the procedure of a inner join here can be separated into three parts: first of all, Spark load the tables to be joined with filtering the invalid rows with null value of the attributes as the join key; then it will select the corresponding attributes from each table according to the select schema, and this part is similar to do select operation before the join operation which may save space and time when joining; finally Spark did a hash-join on the join keys as well as a selection again just to match the final result.



```
2018-06-06 16:57:55 WARN  Utils:66 - Your hostname, cs143 resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
2018-06-06 16:57:55 WARN  Utils:66 - Set SPARK_LOCAL_IP if you need to bind to another address
2018-06-06 16:57:56 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Results of explain():
== Physical Plan ==
*(2) Project [id#14, body#4, labeldjt#53]
+- *(2) BroadcastHashJoin [id#14], [Input_id#50], Inner, BuildRight
   :- *(2) Project [body#4, id#14]
   :  +- *(2) Filter isnotnull(id#14)
   :     +- *(2) FileScan parquet [body#4,id#14] Batched: true, Format: Parquet, Location: InMemoryFileIndex[file:/media/sf_vm-shared/project2/part b/comments], PartitionFilters: [], PushedFilters:
[IsNotNull(id)], ReadSchema: struct<body:string,id:string>
   +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
      +- *(1) Project [Input_id#50, labeldjt#53]
         +- *(1) Filter isnotnull(Input_id#50)
            +- *(1) FileScan parquet [Input_id#50,labeldjt#53] Batched: true, Format: Parquet, Location: InMemoryFileIndex[file:/media/sf_vm-shared/project2/part b/label], PartitionFilters: [],
PushedFilters: [IsNotNull(Input_id)], ReadSchema: struct<Input_id:string,labeldjt:string>


Output of this join:
+-------+--------------------+--------+
|     id|                body|labeldjt|
+-------+--------------------+--------+
|dhez0jx|No it isnt. I cal...|       1|
|dtgkx2z|Good move by the ...|       1|
|dsyd1k4|Well, that's it. ...|       0|
|dbuu8at|&gt;"I also know ...|      -1|
|da8w79n|&gt;He is asking ...|      -1|
|dnf5moq|Donald Trump is b...|      -1|
|du3ewwo|Hillary was guilt...|       0|
|dpx5oj7|Even by liberal d...|       0|
|dlt1213|Can you imagine i...|      -1|
|dqmk3ok|So this is the po...|      -1|
|dht88en|How can developin...|       0|
|da46qad|I see you. Can't ...|       0|
|dek7eqq|&gt; Sane people ...|       1|
|dgf4zhe|Oh man we just ne...|       0|
|dfcjr1y|As a baby boomer ...|       0|
|dfj2gu4|If you think Obam...|       0|
|du0kmlt|I knew it. There ...|      -1|
|dbfdtb8|This is the fucki...|      -1|
|dmoryxn|Wait, wait, wait,...|      -1|
|dt5c32l|All this time I'v...|      -1|
+-------+--------------------+--------+
only showing top 20 rows
```

Figure 11: Result of explain join and the output



```python
def associated(comments, label):
    # task 2
    return comments.join(label, comments.id == label.Input_id, 'inner')

def main(context):
    comments = context.read.load('comments')
    label = context.read.load('label')
    associate = associated(comments, label).select('id', 'body', 'labeldjt')
    print('Results of explain():')
    associate.explain()
    print()
    print()
    print('Output of this join:')
    associate.show()
```

Figure 12: Part of the codes to do explain