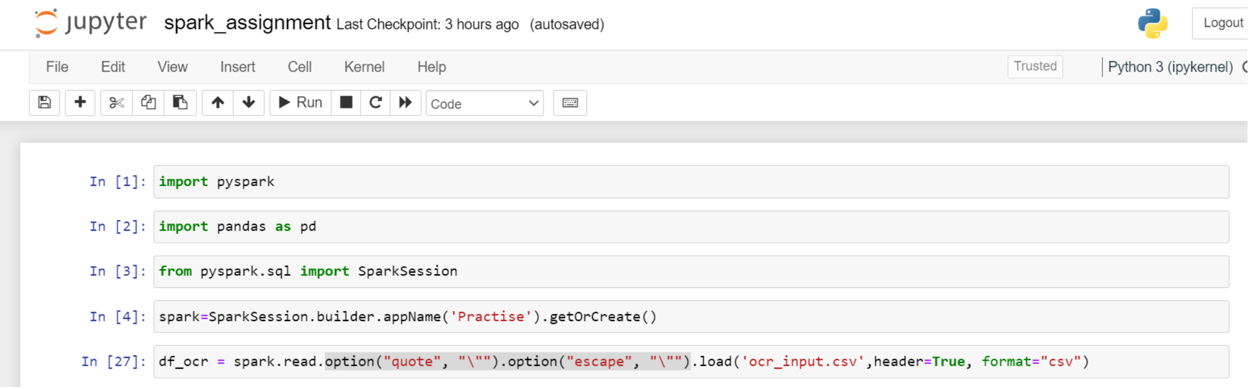
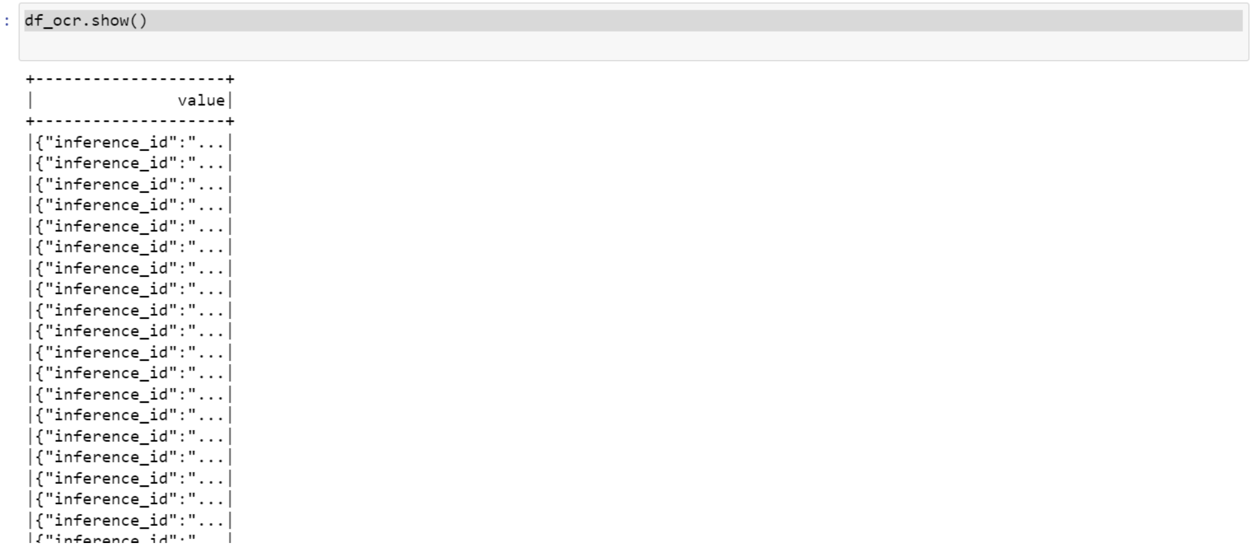
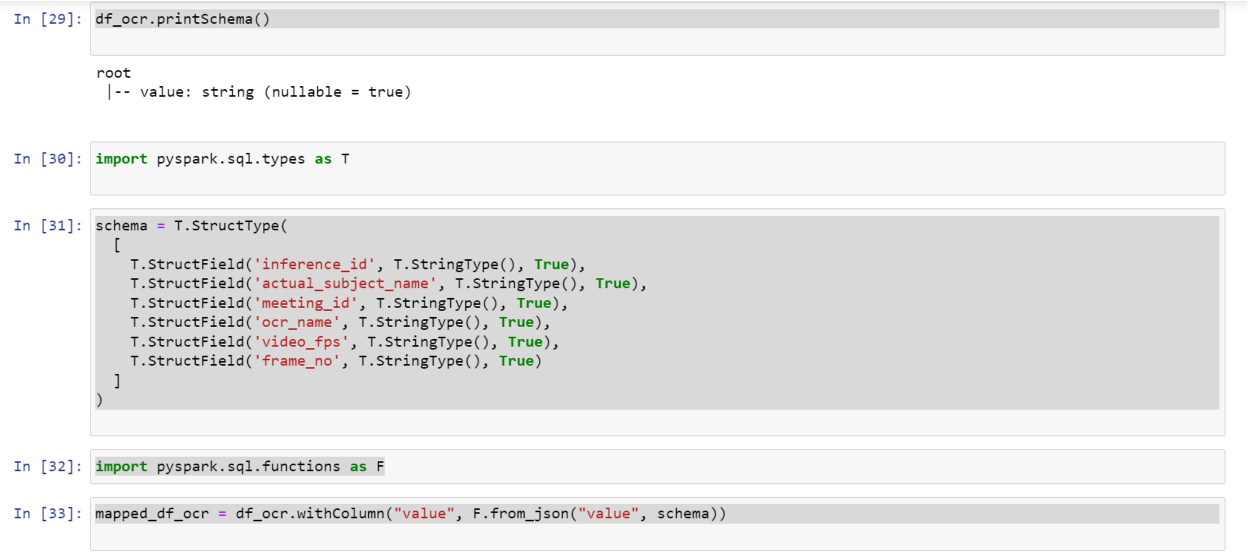


```
In [1]: import pyspark

In [2]: import pandas as pd

In [3]: from pyspark.sql import SparkSession

In [4]: spark=SparkSession.builder.appName('Practise').getOrCreate()

In [27]: df_ocr = spark.read.option("quote", "\"").option("escape", "\"").load('ocr_input.csv',header=True, format="csv")
```



```
: df_ocr.show()

+--------------------+
|               value|
+--------------------+
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"   |
```



```
In [29]: df_ocr.printSchema()

root
 |-- value: string (nullable = true)

In [30]: import pyspark.sql.types as T

In [31]: schema = T.StructType(
           [
             T.StructField('inference_id', T.StringType(), True),
             T.StructField('actual_subject_name', T.StringType(), True),
             T.StructField('meeting_id', T.StringType(), True),
             T.StructField('ocr_name', T.StringType(), True),
             T.StructField('video_fps', T.StringType(), True),
             T.StructField('frame_no', T.StringType(), True)
           ]
         )

In [32]: import pyspark.sql.functions as F

In [33]: mapped_df_ocr = df_ocr.withColumn("value", F.from_json("value", schema))
```

In [34]:
```
mapped_df_ocr.show(truncate=False)
```

```
+--------------------------------------------------------------------------------+
|value                                                                           |
+--------------------------------------------------------------------------------+
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Holden, 25, 1}           |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Holden, 25, 2}           |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Holden, 25, 3}           |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Holden, 25, 4}           |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 5}   |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 6}   |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 7}   |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 8}   |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 9}   |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 10}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 11}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 12}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 13}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 14}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 15}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 16}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 17}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 18}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 19}  |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, Holden, 10131, Senior Partner, 25, 20}  |
```

In [36]:
```
import pyspark.sql.functions as F
```

In [39]:
```
parsed_df_ocr = mapped_df_ocr.select(F.col("value.*"))
parsed_df_ocr.show(truncate=False)
parsed_df_ocr.printSchema()
```

```
+------------------------------------+-------------------+----------+--------------+---------+--------+
|inference_id                        |actual_subject_name|meeting_id|ocr_name      |video_fps|frame_no|
+------------------------------------+-------------------+----------+--------------+---------+--------+
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Holden        |25       |1       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Holden        |25       |2       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Holden        |25       |3       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Holden        |25       |4       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |5       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |6       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |7       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |8       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |9       |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |10      |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |11      |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |12      |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |13      |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |14      |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |15      |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |16      |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|Holden             |10131     |Senior Partner|25       |17      |
```

```
root
 |-- inference_id: string (nullable = true)
 |-- actual_subject_name: string (nullable = true)
 |-- meeting_id: string (nullable = true)
 |-- ocr_name: string (nullable = true)
 |-- video_fps: string (nullable = true)
 |-- frame_no: string (nullable = true)
```

In [40]:
```
df_emotions = spark.read.option("quote", "\"").option("escape", "\"").load('emotions_input.csv',header=True, format="csv")
```

In [42]:
```
df_emotions.show()
```

```
+--------------------+
|               value|
+--------------------+
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
|{"inference_id":"...|
```

In [43]:
```
df_emotions.printSchema()
```

```
root
 |-- value: string (nullable = true)
```

In [44]:
```
schema_emotions = T.StructType(
  [
    T.StructField('inference_id', T.StringType(), True),
    T.StructField('meeting_id', T.StringType(), True),
    T.StructField('frame_no', T.StringType(), True),
    T.StructField('emotion', T.StringType(), True)
]
)
```

In [45]:
```
mapped_df_emotions = df_emotions.withColumn("value", F.from_json("value", schema_emotions))
```

In [47]:
```
mapped_df_emotions.show(truncate=False)
```

```
+-----------------------------------------------------------------------------------------------------------------------+
|value                                                                                                                  |
+-----------------------------------------------------------------------------------------------------------------------+
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 1, {"sad":0,"angry":1,"happy":0,"disgust":1,"neutral":100,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 2, {"sad":0,"angry":1,"happy":0,"disgust":0,"neutral":100,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 3, {"sad":1,"angry":1,"happy":1,"disgust":0,"neutral":100,"surprise":0}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 4, {"sad":0,"angry":0,"happy":0,"disgust":1,"neutral":100,"surprise":0}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 5, {"sad":1,"angry":1,"happy":0,"disgust":0,"neutral":100,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 6, {"sad":0,"angry":0,"happy":0,"disgust":0,"neutral":100,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 7, {"sad":1,"angry":0,"happy":1,"disgust":1,"neutral":100,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 8, {"sad":0,"angry":0,"happy":1,"disgust":1,"neutral":100,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 9, {"sad":0,"angry":0,"happy":1,"disgust":1,"neutral":100,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 10, {"sad":1,"angry":0,"happy":0,"disgust":1,"neutral":100,"surprise":0}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 11, {"sad":1,"angry":1,"happy":0,"disgust":0,"neutral":100,"surprise":1}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 12, {"sad":0,"angry":1,"happy":1,"disgust":1,"neutral":100,"surprise":0}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 13, {"sad":0,"angry":1,"happy":1,"disgust":0,"neutral":100,"surprise":1}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 14, {"sad":1,"angry":0,"happy":1,"disgust":46,"neutral":54,"surprise":0}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 15, {"sad":0,"angry":1,"happy":1,"disgust":84,"neutral":16,"surprise":1}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 16, {"sad":0,"angry":0,"happy":1,"disgust":84,"neutral":15,"surprise":1}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 17, {"sad":1,"angry":1,"happy":1,"disgust":84,"neutral":16,"surprise":0}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 18, {"sad":1,"angry":1,"happy":0,"disgust":96,"neutral":0,"surprise":1}} |
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 19, {"sad":1,"angry":1,"happy":1,"disgust":100,"neutral":1,"surprise":1}}|
|{f78dd9f4-beb6-4ab7-bc5f-e43674565c14, 10131, 20, {"sad":0,"angry":0,"happy":1,"disgust":100,"neutral":0,"surprise":0}}|
+-----------------------------------------------------------------------------------------------------------------------+
```

In [48]:
```
parsed_df_emotions = mapped_df_emotions.select(F.col("value.*"))
parsed_df_emotions.show(truncate=False)
parsed_df_emotions.printSchema()
```

```
+------------------------------------+----------+--------+--------------------------------------------------------------------+
|inference_id                        |meeting_id|frame_no|emotion                                                             |
+------------------------------------+----------+--------+--------------------------------------------------------------------+
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |1       |{"sad":0,"angry":1,"happy":0,"disgust":1,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |2       |{"sad":0,"angry":1,"happy":0,"disgust":0,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |3       |{"sad":1,"angry":1,"happy":1,"disgust":0,"neutral":100,"surprise":0}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |4       |{"sad":0,"angry":0,"happy":0,"disgust":1,"neutral":100,"surprise":0}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |5       |{"sad":1,"angry":1,"happy":0,"disgust":0,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |6       |{"sad":0,"angry":0,"happy":0,"disgust":0,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |7       |{"sad":1,"angry":0,"happy":1,"disgust":1,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |8       |{"sad":0,"angry":0,"happy":1,"disgust":1,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |9       |{"sad":0,"angry":0,"happy":1,"disgust":1,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |10      |{"sad":1,"angry":0,"happy":0,"disgust":1,"neutral":100,"surprise":0}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |11      |{"sad":1,"angry":1,"happy":0,"disgust":0,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |12      |{"sad":0,"angry":1,"happy":1,"disgust":1,"neutral":100,"surprise":0}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |13      |{"sad":0,"angry":1,"happy":1,"disgust":0,"neutral":100,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |14      |{"sad":1,"angry":0,"happy":1,"disgust":46,"neutral":54,"surprise":0}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |15      |{"sad":0,"angry":1,"happy":1,"disgust":84,"neutral":16,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |16      |{"sad":0,"angry":0,"happy":1,"disgust":84,"neutral":15,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |17      |{"sad":1,"angry":1,"happy":1,"disgust":84,"neutral":16,"surprise":0}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |18      |{"sad":1,"angry":1,"happy":0,"disgust":96,"neutral":0,"surprise":1} |
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |19      |{"sad":1,"angry":1,"happy":1,"disgust":100,"neutral":1,"surprise":1}|
|f78dd9f4-beb6-4ab7-bc5f-e43674565c14|10131     |20      |{"sad":0,"angry":0,"happy":1,"disgust":100,"neutral":0,"surprise":0}|
root
 |-- inference_id: string (nullable = true)
 |-- meeting_id: string (nullable = true)
 |-- frame_no: string (nullable = true)
 |-- emotion: string (nullable = true)
```

In [49]:
```
# First identify the subject frames from ocr file. If the actual_subject_name and ocr_name
# column values are same then that frame number is the subject's frame number.
```

In [51]:
```
mapped_df_ocr.head(10)
```

Out[51]:
```
[Row(value=Row(inference_id='f78dd9f4-beb6-4ab7-bc5f-e43674565c14', actual_subject_name='Holden', meeting_id='10131', ocr_name
='Holden', video_fps='25', frame_no='1')),
 Row(value=Row(inference_id='f78dd9f4-beb6-4ab7-bc5f-e43674565c14', actual_subject_name='Holden', meeting_id='10131', ocr_name
='Holden', video_fps='25', frame_no='2')),
 Row(value=Row(inference_id='f78dd9f4-beb6-4ab7-bc5f-e43674565c14', actual_subject_name='Holden', meeting_id='10131', ocr_name
='Holden', video_fps='25', frame_no='3')),
 Row(value=Row(inference_id='f78dd9f4-beb6-4ab7-bc5f-e43674565c14', actual_subject_name='Holden', meeting_id='10131', ocr_name
='Holden', video_fps='25', frame_no='4')),
 Row(value=Row(inference_id='f78dd9f4-beb6-4ab7-bc5f-e43674565c14', actual_subject_name='Holden', meeting_id='10131', ocr_name
='Senior Partner', video_fps='25', frame_no='5')),
 Row(value=Row(inference_id='f78dd9f4-beb6-4ab7-bc5f-e43674565c14', actual_subject_name='Holden', meeting_id='10131', ocr_name
='Senior Partner', video_fps='25', frame_no='6')),
 Row(value=Row(inference_id='f78dd9f4-beb6-4ab7-bc5f-e43674565c14', actual_subject_name='Holden', meeting_id='10131', ocr_name

    Senior Partner', video_fps='25', frame_no='10'))]
```

In [ ]:
```
df_ocr['subject_frame_no']=mapped_df_ocr.apply(lambda x:x['frame_no'] if x['actual_subject_name']=x['ocr_name'] else null,axis=1)
```

In [58]:
```
from pyspark.sql.functions import when, col, lit
parsed_df_ocr = parsed_df_ocr.withColumn(
    'subject_frame_no',
    when(col('actual_subject_name') == col('ocr_name'), col('frame_no'))
    .otherwise(lit(None))
)
```

In [59]:
```
parsed_df_ocr.show()
```

```
+--------------------+-------------------+----------+--------------+---------+--------+----------------+
|        inference_id|actual_subject_name|meeting_id|      ocr_name|video_fps|frame_no|subject_frame_no|
+--------------------+-------------------+----------+--------------+---------+--------+----------------+
|f78dd9f4-beb6-4ab...|             Holden|     10131|        Holden|       25|       1|               1|
|f78dd9f4-beb6-4ab...|             Holden|     10131|        Holden|       25|       2|               2|
|f78dd9f4-beb6-4ab...|             Holden|     10131|        Holden|       25|       3|               3|
|f78dd9f4-beb6-4ab...|             Holden|     10131|        Holden|       25|       4|               4|
|f78dd9f4-beb6-4ab...|             Holden|     10131|Senior Partner|       25|       5|            null|
|f78dd9f4-beb6-4ab...|             Holden|     10131|Senior Partner|       25|       6|            null|
|f78dd9f4-beb6-4ab...|             Holden|     10131|Senior Partner|       25|       7|            null|
|f78dd9f4-beb6-4ab...|             Holden|     10131|Senior Partner|       25|       8|            null|
```

In [ ]:
```
# left join emotions data on ocr data
```

In [ ]:
```
# subject_emotion_df=emotions_df.merge(how='left',left_on=['inference_id','meeting_id','frame_no'],right_on=['inference_id','meet
```

In [62]:
```
emotions_df_alias = parsed_df_emotions.alias('emotions')
ocr_df_alias = parsed_df_ocr.alias('ocr')



subject_emotion_df = emotions_df_alias.join(
    other=ocr_df_alias,
    on=[
        col('emotions.inference_id') == col('ocr.inference_id'),
        col('emotions.meeting_id') == col('ocr.meeting_id'),
        col('emotions.frame_no') == col('ocr.subject_frame_no')
    ],
    how='left'
)
```

In [89]:
```
# parsed_df_emotions.columns
# result_df = subject_emotion_df.select(emotions_df_alias.columns + [col('ocr.subject_frame_no')])
result_df = subject_emotion_df.select(['emotions.inference_id','emotions.meeting_id','emotions.frame_no','emotions.emotion','ocr.
# emotions_df_alias.columns
```

In [90]:
```
result_df.show()
```

```
+--------------------+----------+--------+--------------------+----------------+---------+
|        inference_id|meeting_id|frame_no|             emotion|subject_frame_no|video_fps|
+--------------------+----------+--------+--------------------+----------------+---------+
|f78dd9f4-beb6-4ab...|     10131|       1|{"sad":0,"angry":...|               1|       25|
|f78dd9f4-beb6-4ab...|     10131|      10|{"sad":1,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      11|{"sad":1,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      12|{"sad":0,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      13|{"sad":0,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      14|{"sad":1,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      15|{"sad":0,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      16|{"sad":0,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      17|{"sad":1,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      18|{"sad":1,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|      19|{"sad":1,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|       2|{"sad":0,"angry":...|               2|       25|
|f78dd9f4-beb6-4ab...|     10131|      20|{"sad":0,"angry":...|            null|     null|
|f78dd9f4-beb6-4ab...|     10131|   20132|{"sad":1,"angry":...|            null|     null|
```

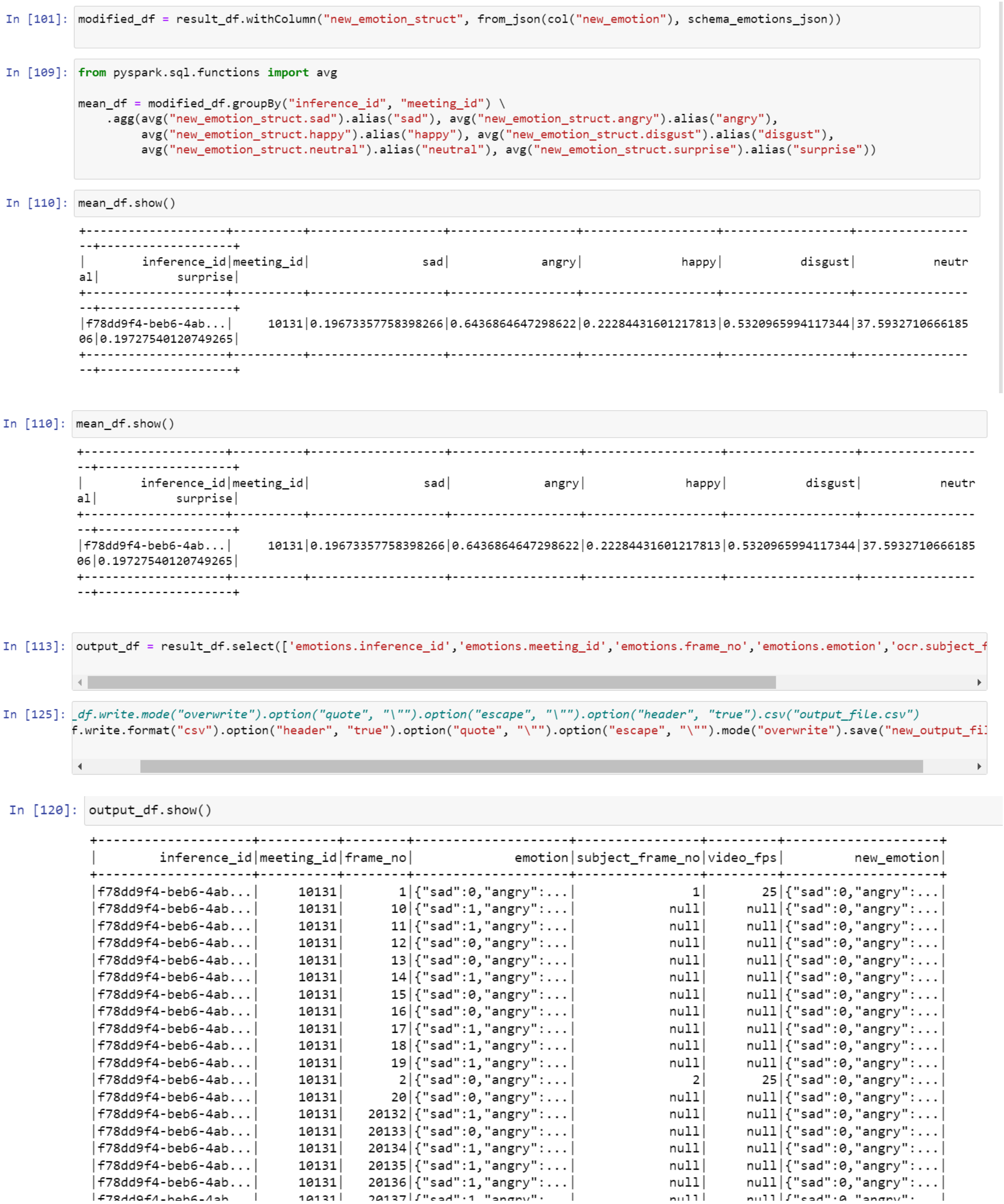
In [ ]:
```
# result_df['new_emotion']=result_df.apply(lambda x:{"sad":0,"angry": 0,"happy": 0,"disgust": 0,"neutral": 0,"surprise": 0} if x[
```

In [93]:
```
from pyspark.sql.functions import when, col, lit

result_df = result_df.withColumn(
    "new_emotion",
    when(col("subject_frame_no").isNull(), lit('{"sad":0,"angry": 0,"happy": 0,"disgust": 0,"neutral": 0,"surprise": 0}'))
    .otherwise(col("emotion"))
)
```

In [92]:
```
result_df.show()
```

```
+--------------------+----------+--------+--------------------+----------------+---------+--------------------+
|        inference_id|meeting_id|frame_no|             emotion|subject_frame_no|video_fps|         new_emotion|
+--------------------+----------+--------+--------------------+----------------+---------+--------------------+
|f78dd9f4-beb6-4ab...|     10131|       1|{"sad":0,"angry":...|               1|       25|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      10|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      11|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      12|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      13|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      14|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      15|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      16|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      17|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
```

In [96]:
```
# df = result_df.withColumn("sad_emotion", col("new_emotion").getItem("sad"))

from pyspark.sql.functions import from_json, col
from pyspark.sql.types import StructType, StructField, IntegerType

# Define the schema for the new_emotion column
schema_emotions_json = StructType([
    StructField("sad", IntegerType()),
    StructField("angry", IntegerType()),
    StructField("happy", IntegerType()),
    StructField("disgust", IntegerType()),
    StructField("neutral", IntegerType()),
    StructField("surprise", IntegerType())
])
```



```
In [101]: modified_df = result_df.withColumn("new_emotion_struct", from_json(col("new_emotion"), schema_emotions_json))
```

```
In [109]: from pyspark.sql.functions import avg

          mean_df = modified_df.groupBy("inference_id", "meeting_id") \
              .agg(avg("new_emotion_struct.sad").alias("sad"), avg("new_emotion_struct.angry").alias("angry"),
                   avg("new_emotion_struct.happy").alias("happy"), avg("new_emotion_struct.disgust").alias("disgust"),
                   avg("new_emotion_struct.neutral").alias("neutral"), avg("new_emotion_struct.surprise").alias("surprise"))
```

```
In [110]: mean_df.show()

+--------------------+----------+-------------------+------------------+-------------------+------------------+----------------
--+-------------------+
|        inference_id|meeting_id|                sad|             angry|              happy|           disgust|           neutr
al|           surprise|
+--------------------+----------+-------------------+------------------+-------------------+------------------+----------------
--+-------------------+
|f78dd9f4-beb6-4ab...|     10131|0.19673357758398266|0.6436864647298622|0.22284431601217813|0.5320965994117344|37.5932710666185
06|0.19727540120749265|
+--------------------+----------+-------------------+------------------+-------------------+------------------+----------------
--+-------------------+
```

```
In [110]: mean_df.show()

+--------------------+----------+-------------------+------------------+-------------------+------------------+----------------
--+-------------------+
|        inference_id|meeting_id|                sad|             angry|              happy|           disgust|           neutr
al|           surprise|
+--------------------+----------+-------------------+------------------+-------------------+------------------+----------------
--+-------------------+
|f78dd9f4-beb6-4ab...|     10131|0.19673357758398266|0.6436864647298622|0.22284431601217813|0.5320965994117344|37.5932710666185
06|0.19727540120749265|
+--------------------+----------+-------------------+------------------+-------------------+------------------+----------------
--+-------------------+
```

```
In [113]: output_df = result_df.select(['emotions.inference_id','emotions.meeting_id','emotions.frame_no','emotions.emotion','ocr.subject_f
```

```
In [125]: _df.write.mode("overwrite").option("quote", "\"").option("escape", "\"").option("header", "true").csv("output_file.csv")
          f.write.format("csv").option("header", "true").option("quote", "\"").option("escape", "\"").mode("overwrite").save("new_output_fi
```

```
In [120]: output_df.show()

+--------------------+----------+--------+--------------------+----------------+---------+--------------------+
|        inference_id|meeting_id|frame_no|             emotion|subject_frame_no|video_fps|         new_emotion|
+--------------------+----------+--------+--------------------+----------------+---------+--------------------+
|f78dd9f4-beb6-4ab...|     10131|       1|{"sad":0,"angry":...|               1|       25|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      10|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      11|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      12|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      13|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      14|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      15|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      16|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      17|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      18|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      19|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|       2|{"sad":0,"angry":...|               2|       25|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|      20|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|   20132|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|   20133|{"sad":0,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|   20134|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|   20135|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|   20136|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
|f78dd9f4-beb6-4ab...|     10131|   20137|{"sad":1,"angry":...|            null|     null|{"sad":0,"angry":...|
```