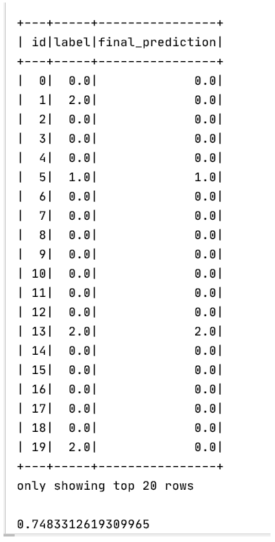
COMP9313 Proj2 Report

Name: ZANNING WANG
zID: z5224151

Q1: Evaluation of your stacking model on the test data.

Answer1:
In the function: base_features_gen_pipeline(), we use built-in function: tokenizer to translate the comment into separate words, use built-in function CountVectorizer() to convert words into spare vector, use built-in function StringIndexer() to translate category into number from 0 to 4.

In order to improve efficiency, I write two more functions to decrease the space complexity:
Function convert_joint(): to convert the category into label which is an integer between 0 and 2.
Function generate_joint(): to add column joint_pred_0, joint_pred_1, joint_pred_2 to the output.
In task 1.3, I print the result according to the test_data：



```
+---+-----+----------------+
| id|label|final_prediction|
+---+-----+----------------+
|  0|  0.0|             0.0|
|  1|  2.0|             0.0|
|  2|  0.0|             0.0|
|  3|  0.0|             0.0|
|  4|  0.0|             0.0|
|  5|  1.0|             1.0|
|  6|  0.0|             0.0|
|  7|  0.0|             0.0|
|  8|  0.0|             0.0|
|  9|  0.0|             0.0|
| 10|  0.0|             0.0|
| 11|  0.0|             0.0|
| 12|  0.0|             0.0|
| 13|  2.0|             2.0|
| 14|  0.0|             0.0|
| 15|  0.0|             0.0|
| 16|  0.0|             0.0|
| 17|  0.0|             0.0|
| 18|  0.0|             0.0|
| 19|  2.0|             0.0|
+---+-----+----------------+
only showing top 20 rows

0.7483312619309965
```

In the first 20 sets of data, most of the predictions of the data are consistent with the label, and the overall accuracy rate can reach 74.83%

Q2: How would you improve the performance (e.g., F1) of the stacking model.

Answer2:
I have observed that when processing raw data in the code, punctuation is not processed. This means that a word with and without punctuation will be treated as two different words for training, thereby reducing the accuracy of prediction. Therefore, in the base_features_gen_pipeline() function, the previous step of Tokenizer can be performed to process the characters and convert them into words without punctuation, all in lowercase, to narrow the scope of items.