**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：

图片包含 游戏机

描述已自动生成

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.