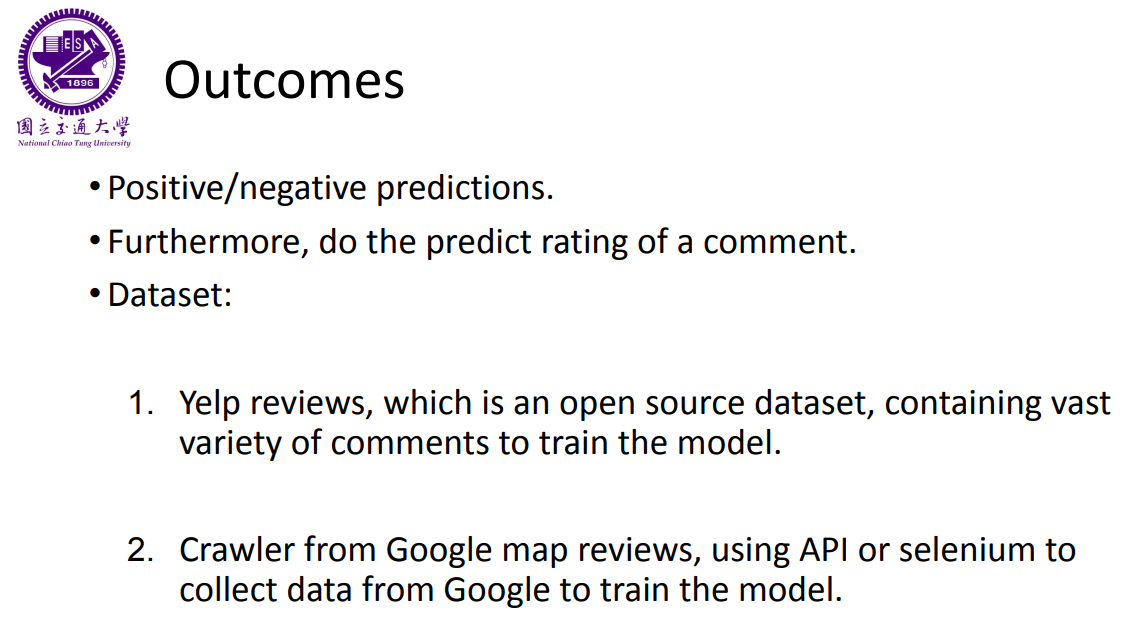
**Undergraduate AI Capstone NYCU Spr2022 Final Group Project**

0816175 高靖 0816006郭冠德 0816081 張以廉

Our project is about NLP. We try to collect reviews on google map & try to do some prediction of the reviews rating.

Our original slides of our short present:



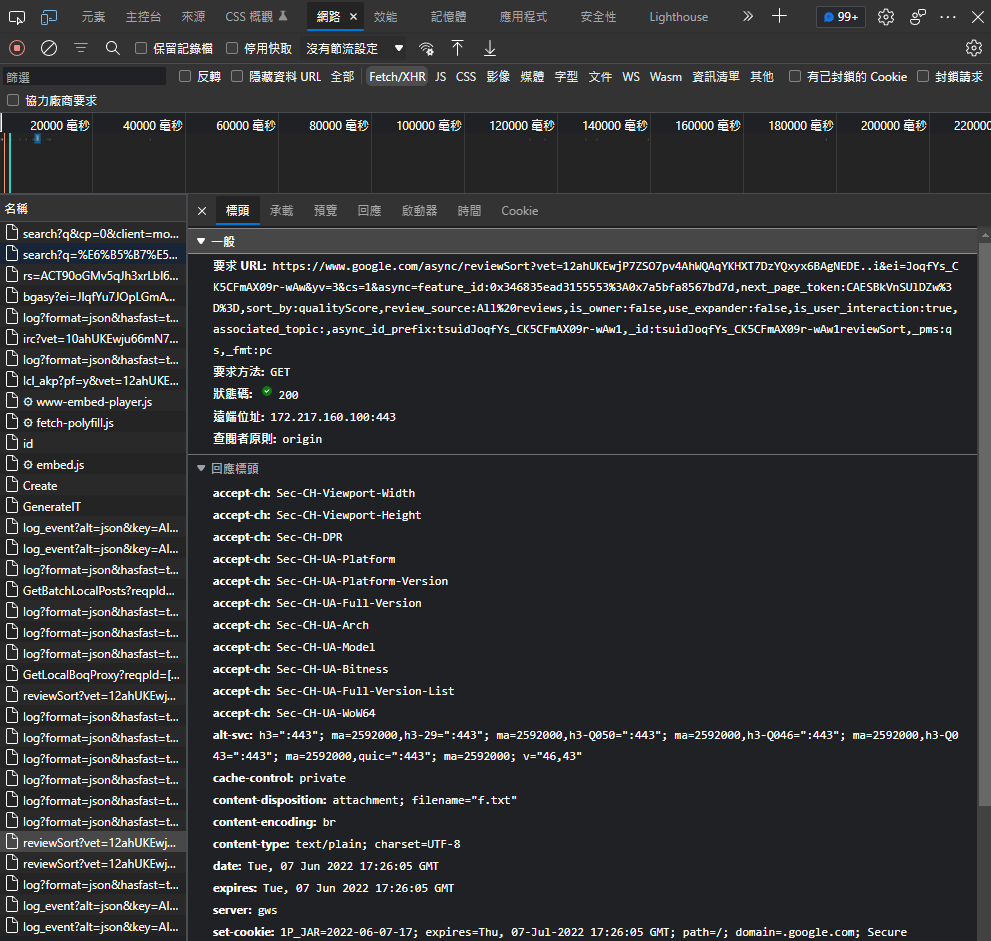
**Data collection:**

So, originally, we try to use some online open-source dataset. But in the end, we try to make it our own way. So instead to use some existing data on the Internet, we decide to do crawler on our own.

We were trying to use Yelp dataset([Yelp Dataset](https://www.yelp.com/dataset)) which is a famous Open dataset which includes personal ,educational, and some academic purpose and reviews & user data dataset. It has JSON & SQL form. However, we think do the Chinese part is more interesting. So we collect the google reviews of some famous store in Taiwan. So, how wee collect our data?

By developer tool, we can check the console of Internet/Fetch/XHR/Headers, to see how the google map backend call the backend api. So as we scroll down the reviews, we can check there’s an request URL which is called repeatedly.

So we can cut in this point, just to get the reviews by calling the same URL.(next page)

We can compare the 2 URL of the request, to do the loop crawler. However, I found that

URL1:https://www.google.com/async/reviewSort?vet=12ahUKEwjP7ZSO7pv4AhWQAqYKHXT7DzYQxyx6BAgNEDE..i&ei=JoqfYs\_CK5CFmAX09r-wAw&yv=3&cs=1&async=feature\_id:0x346835ead3155553%3A0x7a5bfa8567bd7d,next\_page\_token:CAESBkVnSUlDZw%3D%3D,sort\_by:qualityScore,review\_source:All%20reviews,is\_owner:false,use\_expander:false,is\_user\_interaction:true,associated\_topic:,async\_id\_prefix:tsuidJoqfYs\_CK5CFmAX09r-wAw1,\_id:tsuidJoqfYs\_CK5CFmAX09r-wAw1reviewSort,\_pms:qs,\_fmt:pc

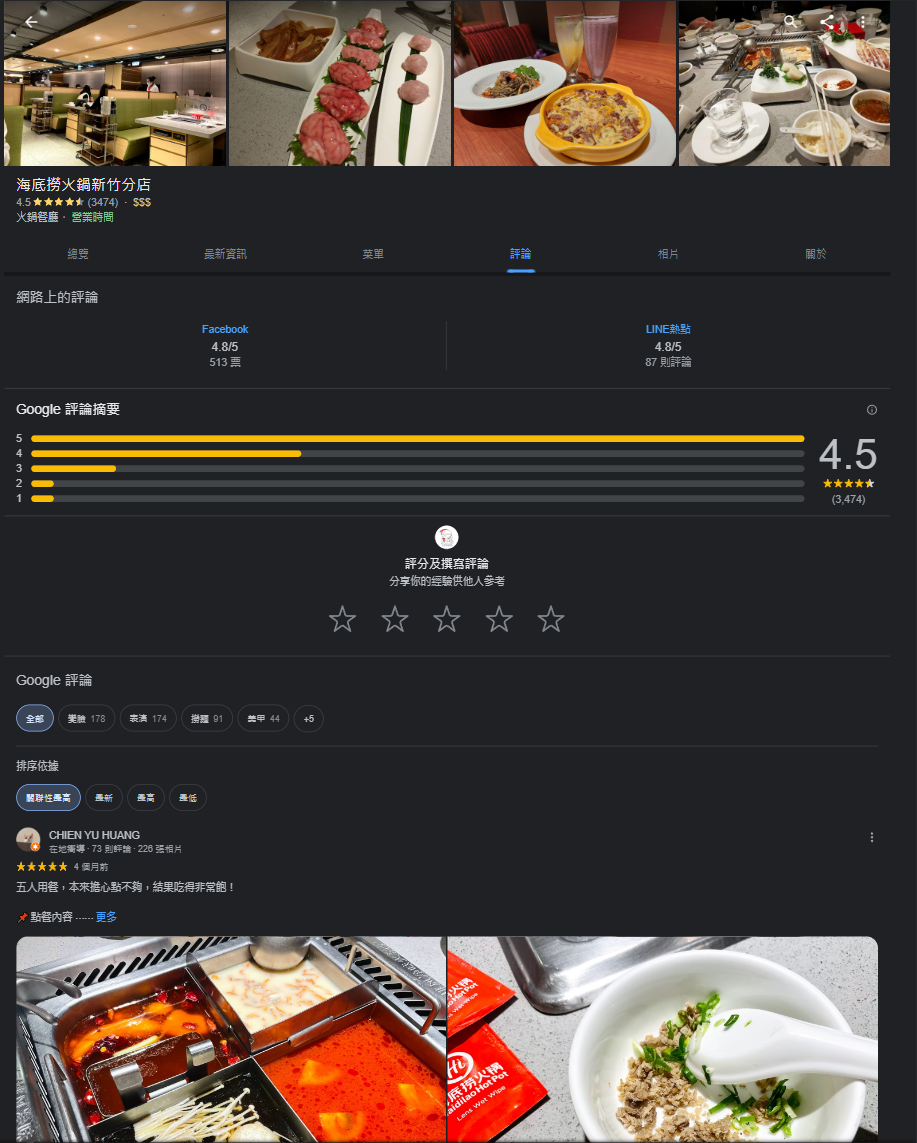
URL2:https://www.google.com/async/reviewSort?vet=12ahUKEwjP7ZSO7pv4AhWQAqYKHXT7DzYQxyx6BAgNEDE..i&ei=JoqfYs\_CK5CFmAX09r-wAw&yv=3&cs=1&async=feature\_id:0x346835ead3155553%3A0x7a5bfa8567bd7d,next\_page\_token:CAESBkVnSUlGQQ%3D%3D,sort\_by:qualityScore,review\_source:All%20reviews,is\_owner:false,use\_expander:false,is\_user\_interaction:true,associated\_topic:,async\_id\_prefix:tsuidJoqfYs\_CK5CFmAX09r-wAw1,\_id:tsuidJoqfYs\_CK5CFmAX09r-wAw1reviewSort,\_pms:qs,\_fmt:pc

The crucial part is about the param: **next\_page\_token.**

Rest part are the same. So here comes the question, how can we decide find the pattern of the URL? Because the token seems to be random. After some research, I found that it do have regular pattern is that, **each store start with DZw, GQQ,IZw,LQQ**……. So the token seems no rules, but can we get the next page token before do the crawler? The answer is negative, we still need to find the token manually and, then, apply to every store.

So, it may be not a ‘smart’ way to collect way to collect data but a ‘not bad’ way, if you do have time to do so. For us, we want to find more ‘intelligent’ way that not to do any manual work. So here’s work around we find:

Instead of using the reviews on web page of the google store

We change to using the map application interface, because the api called here is different from web interface.

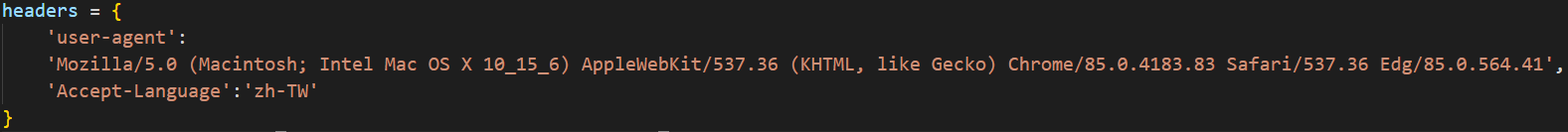
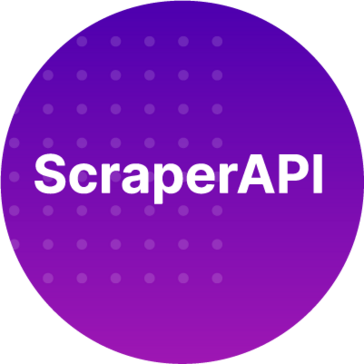
The left hand side is

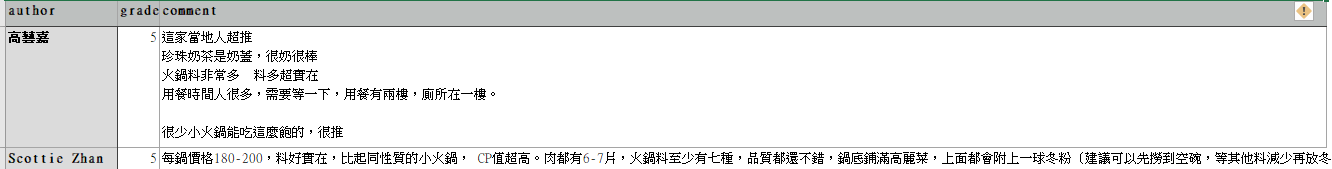
While the right hand side is 

So the map version has rules to follow to do the crawler, by iterating one of the params i, which loop as we scroll the page.

So next part is about how to do the crawler.

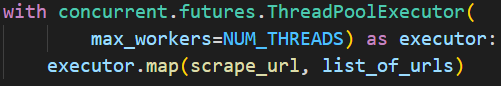
Initially, we try to use selenium/webdriver, but we finally decide to call the api of backend. And, google’s anti-crawler still block my continuous request in about 1 min. So, to avoid such circumstance, I add the header(language for not to get reviews other than Chinese, or we would receive English review)

Hope not be blocked again, however, it’s of no use. So I use VPN, try not to request from same ip for too many times. But it still of no use, because our demand is so big that these method can’t prevent we from being blocked. So, we look up for another third party tool. Scraper API.

It can automatically change from millions of proxy server, and do the retry also. So I use the tool and do implementation in python. (all codes will be on github also).

So I use the tool to get data, and here comes the last question,

It’s too slow. Only 2500 reviews takes few hours to craw. So I change the way we craw to multi-thread. Use the python module, ,we create all URL as a list, and view them as a pool, then the workers will execute the crawler from pool once they’re free.

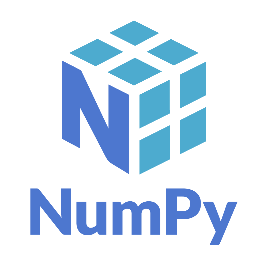
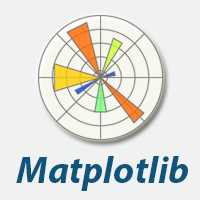
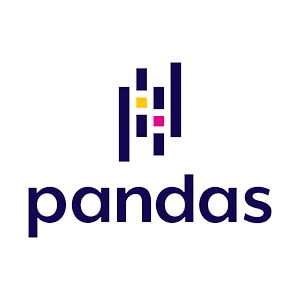


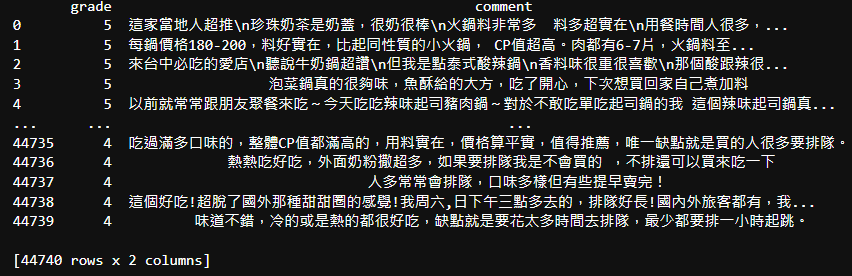
After crawling, export the data as excel.

So, that’s about the whole process of the data collection.

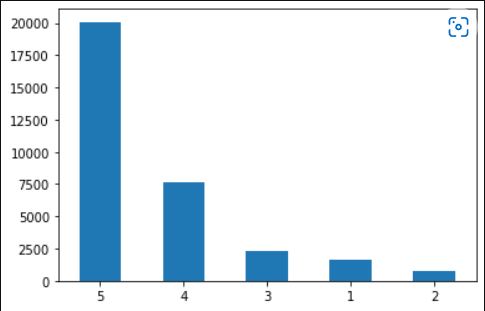
**NLP part:**

For data analysis, preprocessing, I use 3 famous tool in Machine Learning & in NLP field. The Pandas & matplotlib & numpy.

So, first, let’s import our data using pandas.read\_excel()

So the data is import as dataFrame like this: 

Let’s using matplotlib to see how our data distribution:



There’s huge difference between amounts of 5 starts & others, Actually it’s normal, because real trend on Google maps is that Taiwanese seem love to give 5 stars on revies.

Let’s hand over it in next version, let’s move on first.

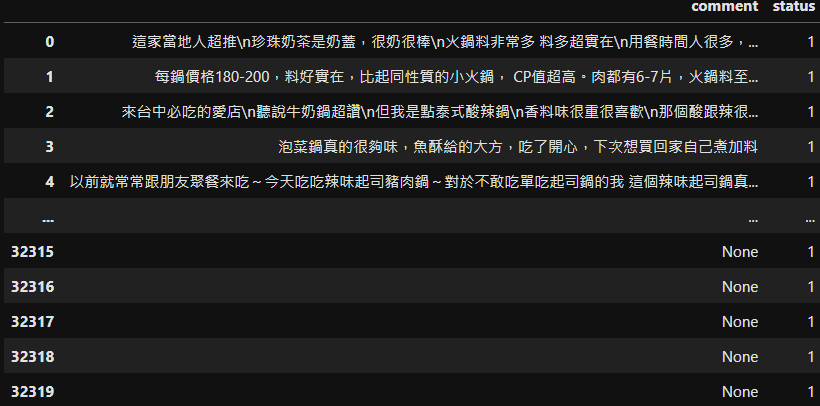
Next step, we’re going to deal with the row without comment. Because some people have no comments but only stars like Sasa Chen:

(this step is address reviews below)



So our crawler will only get None value in excel. So we replace ‘None’ to Nan, then use pandas.dropna() to get rid of those empty value. And after dealing with the some basic operation. We can start to train our model.

Let’s see positive/negative analysis first, So we change the reviews into 2 categories, rating >3 are positive, others are negative. And we choose to use bert as our model, which is a bidirectional encoder from transformers. So using pyTorch & CUDA, I build the model on my own environment. Finishing the fine tune part of bert is to set the label as int, and use f1score, confusion matrix to present the outcome. I use sklearn’s package.

So the original data would look like this after doing 2-label.

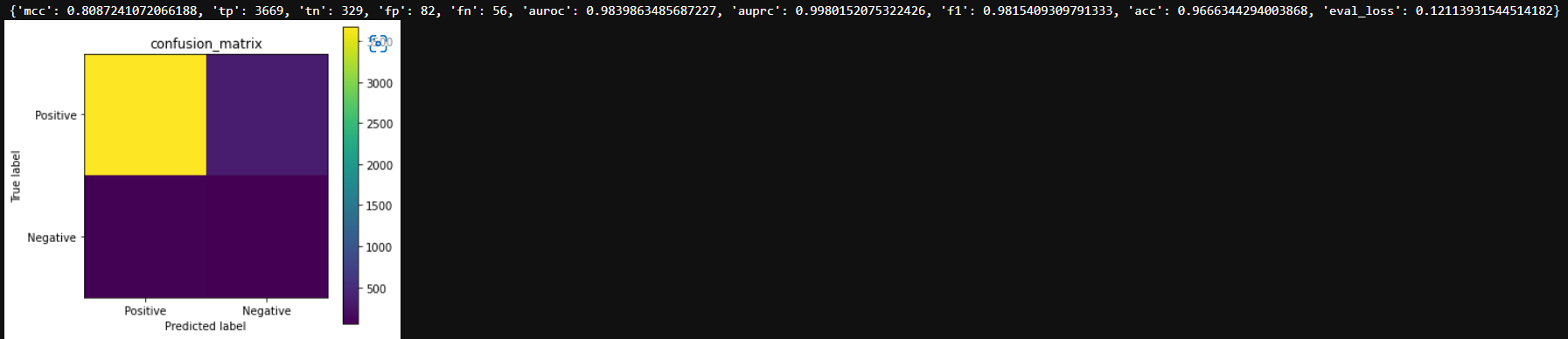
Using train\_test\_split in sklearn.model\_selection, I use test data

Size of 4200. And most of data are positive, which is quite

normal and fulfill original trend. So let’s do 5-label version.

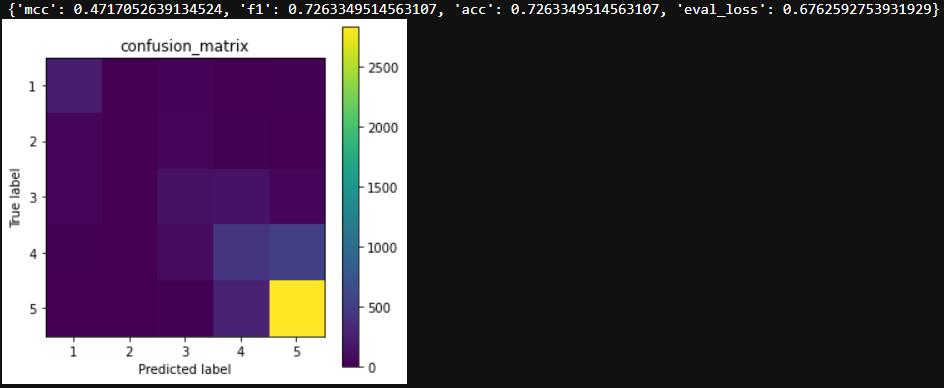
With the same processing before, we change the None to Nan

then remove them from data. And minus 1 from original

score, Cause the bert’s label need to start from 0, instead of 1.

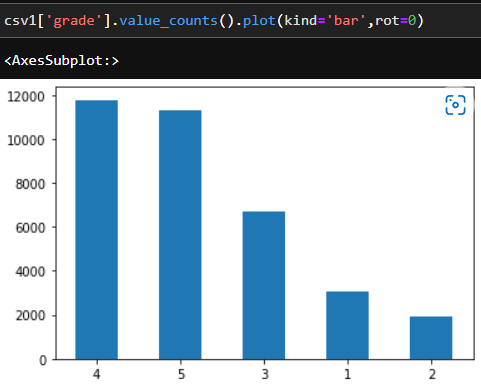


2-label version

Here’s the outcome of 5-label:

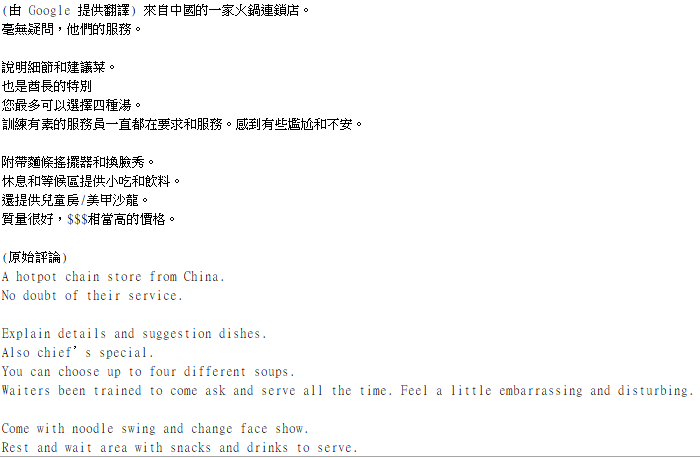
5-label version

We can see that most of the data still appears in label 5-stars. So to get a more accurate outcome, we try to make the numbers of 5-stars not exceed others too much. So we remove 10000 reviews of 5-stars. And the data distribution now:



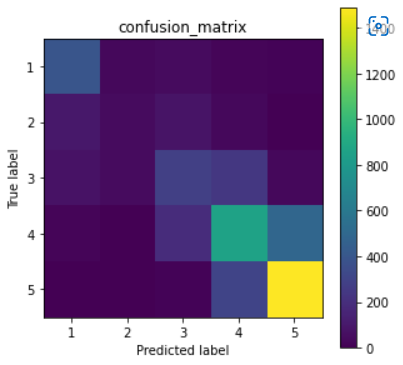
Also, we remove the reviews that contains translation.

Because some foreigner might comment in language other than Chinese. Like the example of a comment of 海底撈:

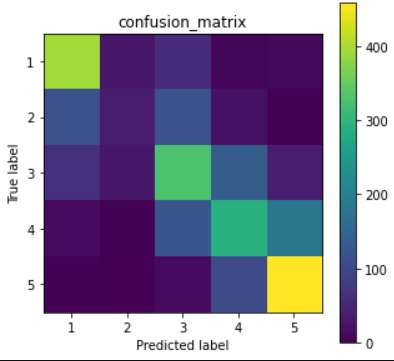


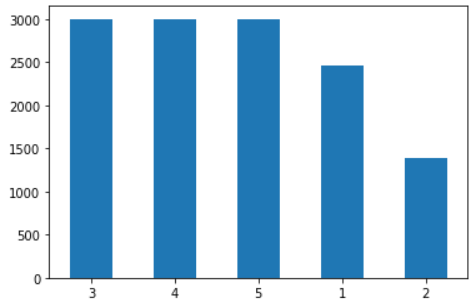
So if comment contains ‘Google提供翻譯’, we remove it.

And here’s the consequence:



We can see the diagonal is more clear, which means our model is getting better after the label become more balanced.

So we change the data distribution again, make data more balanced. And here’s our final outcome:



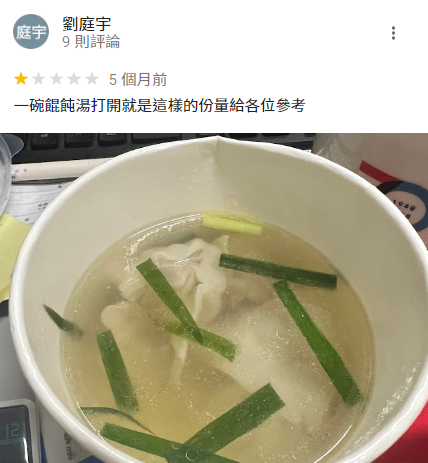
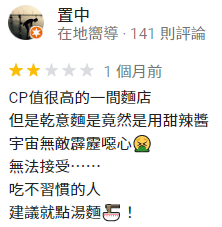
We can see the diagonal is clear but a little bit divergent.

And start-2 seems the most bad, so let’s check the original data:





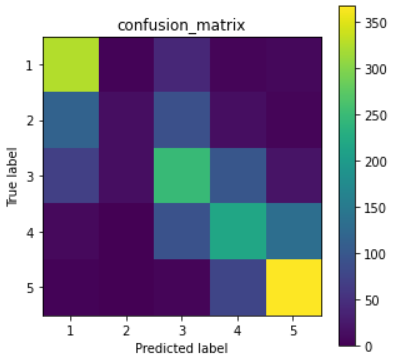
All of these comments are not ‘that’ bad I think. But they’re only worth of 2 stars. Basically, the comment is too subjective & someone will comment in strange viewpoint to affect our prediction.

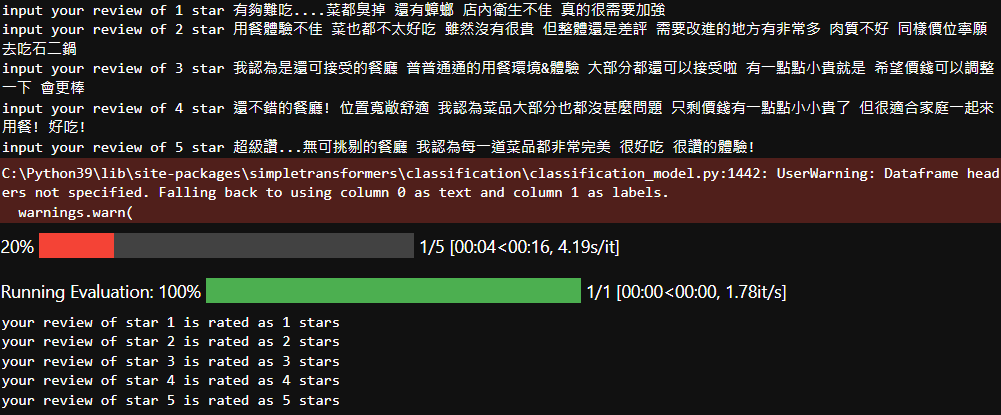
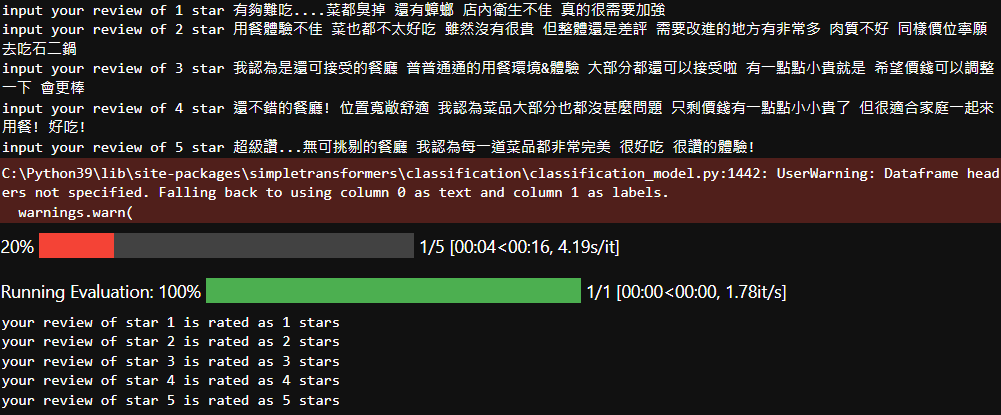
So comments seem to be neutral can be low stars also, which affect our model.

To do it more precise, I filter the comment that’s too short like:

‘難吃’,’還不錯’,’流浪漢不會跳舞’, only leave the comment whose length is more than 15.

And the outcome looks much the same as the previous one.

So I think the most of the reason is that reviews are subjective, and we can analysis in such way to get most of the result correct(70% of the majority people).

Last part, I do a manually input reviews, to check the outcome by your own.

The reviews are all from my own opinion when I share comment in real condition, and the outcome is good!

So that’s all about our project. In the process, most of time spending on the model, tons of error and fix the crawler bug.

It’s a long journey & We learn a lot in the project.

Hope you enjoy our final project!

Group 23

Cute Bert 😊 (easy to learn, hard to utilize & debug, my environment was a chaos after the project)

