**Anime score prediction**

Team33

0516025 張智閔

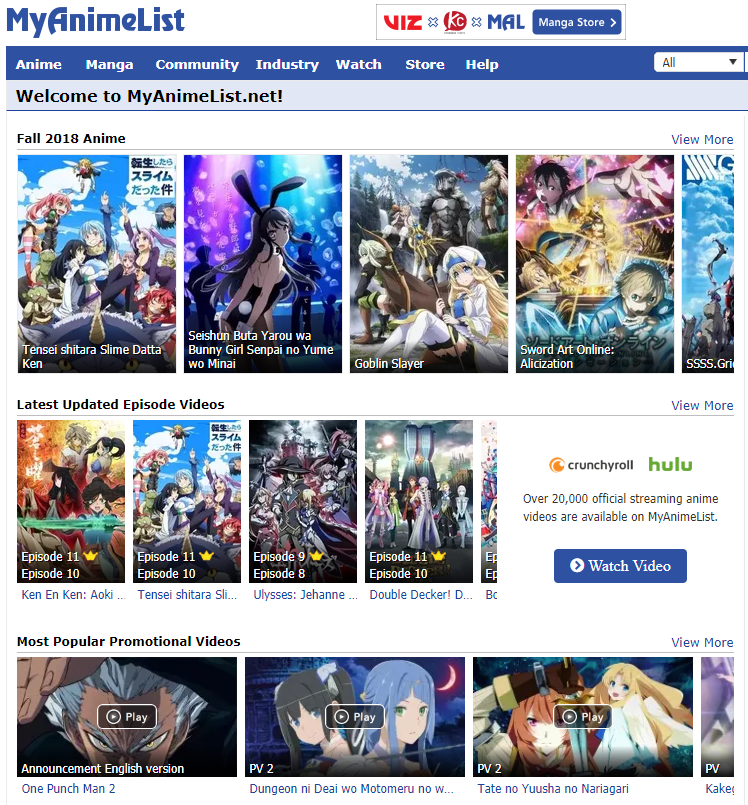
0516032 邱繼聖

0516049 吳柏劭

0516215 林亮穎

0516220 李元毓

Introduction



MyAnimelist - [myanimelist.net](https://myanimelist.net/)

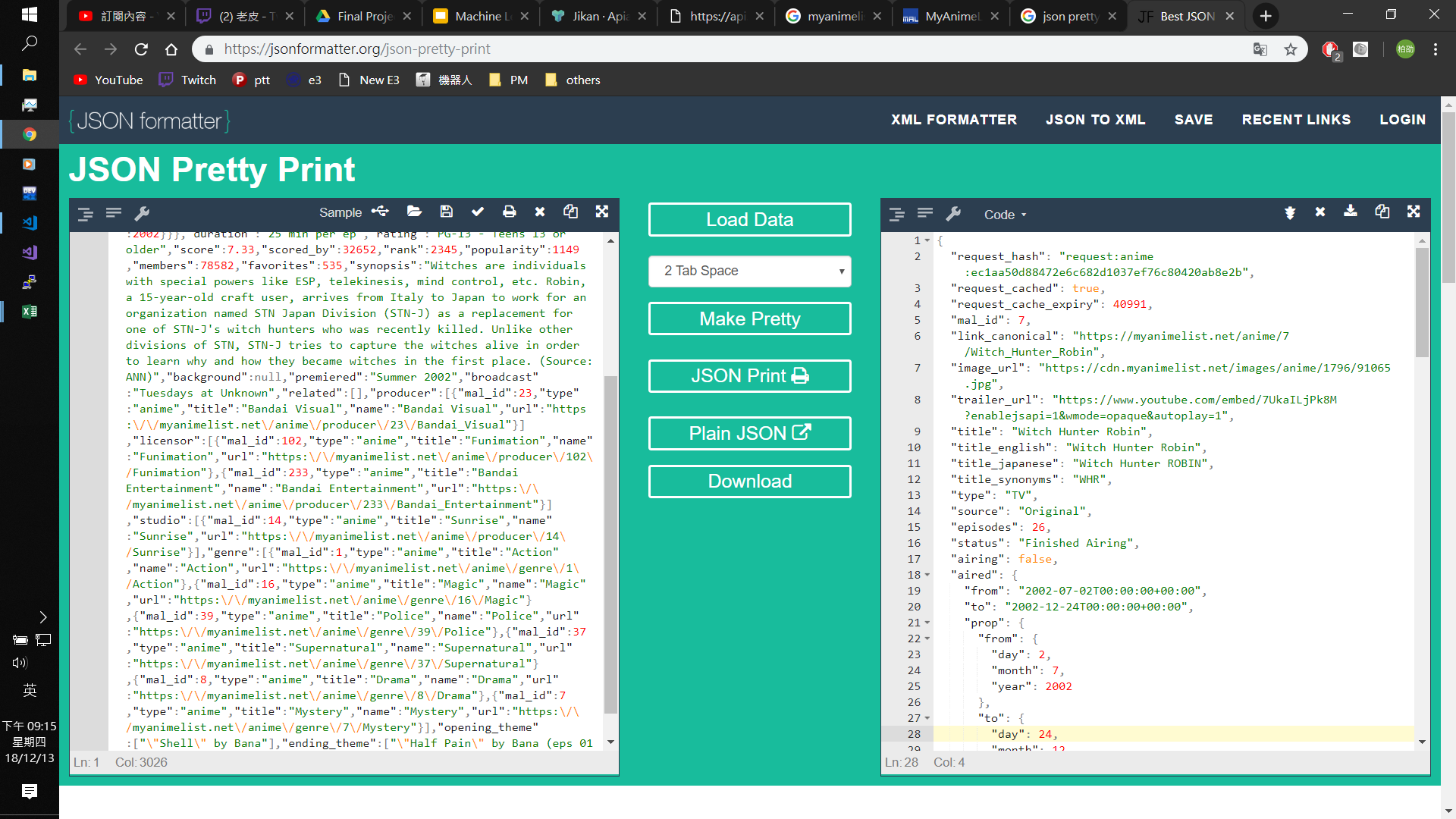
The world’s largest anime database and community

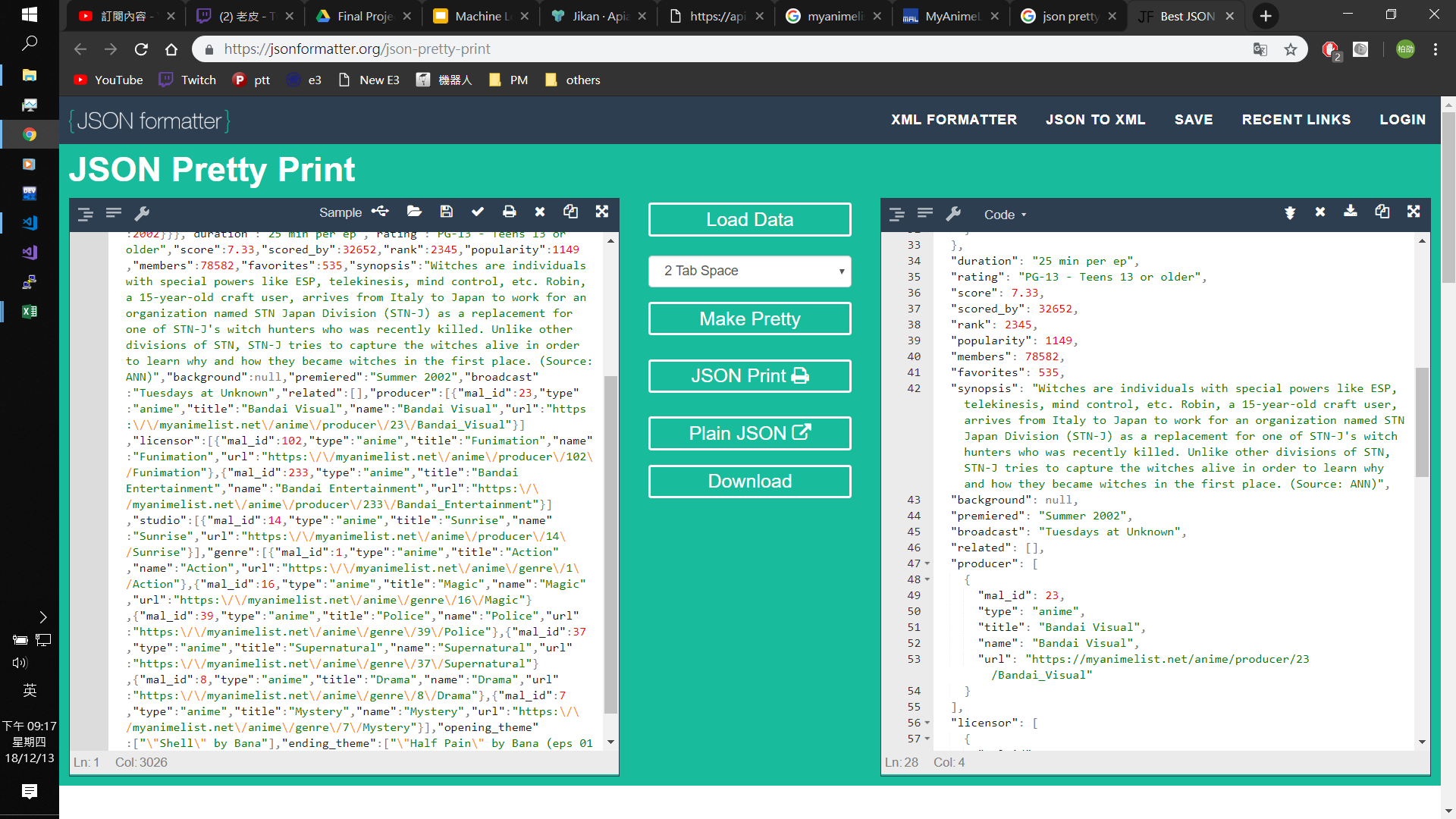
Crawler & Data Preprocessing

使用第三方API取得網站資料

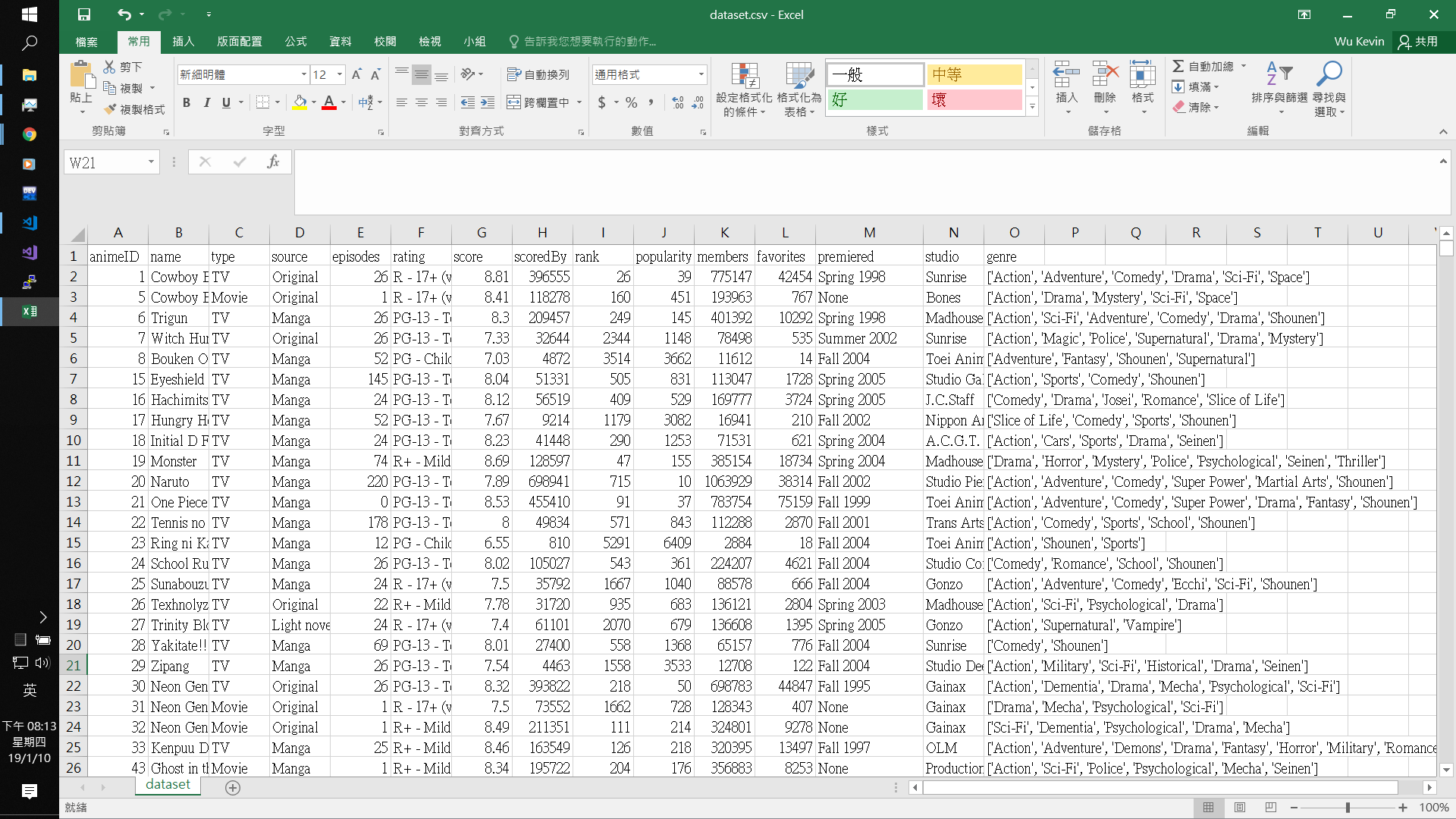
url = 'http://api.jikan.moe/anime/' + str (id)

res = requests.get (url, headers = headers, timeout = None)





以 id 抓取各個動畫的數據，再將所有資料整理成表格

animeID : 動畫在網站上的ID

type : 類型 (ex: TV, Movie, OVA...)

source : 改編自何種作品 (ex: Original, Manga, Light novel...)

episodes : 播放集數

rating : 分級 (ex: R-17+, PG-13...)

score : 網站評分

scoredBy : 評分人數

rank : 網站排名

popularity : 人氣

members : 關注人數

favorites : 加為最愛的人數

premiered : 播放時間 (播放季節 + 播放年份)

studio : 工作室

genre : 動畫分類 (ex: Action, Adventure, Comedy...)

為了方便之後的分析，對其中幾個欄位做 Preprocessing

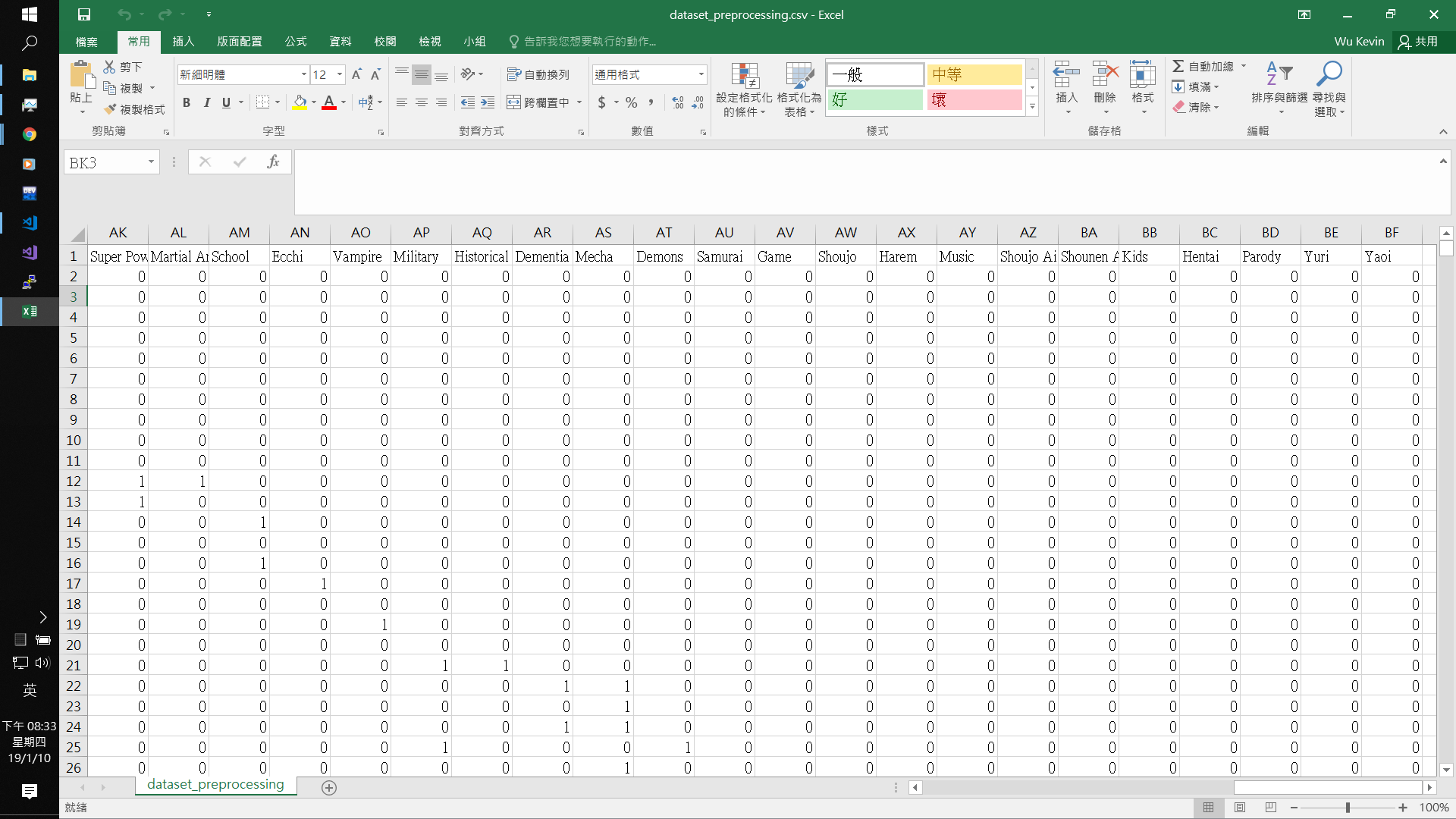
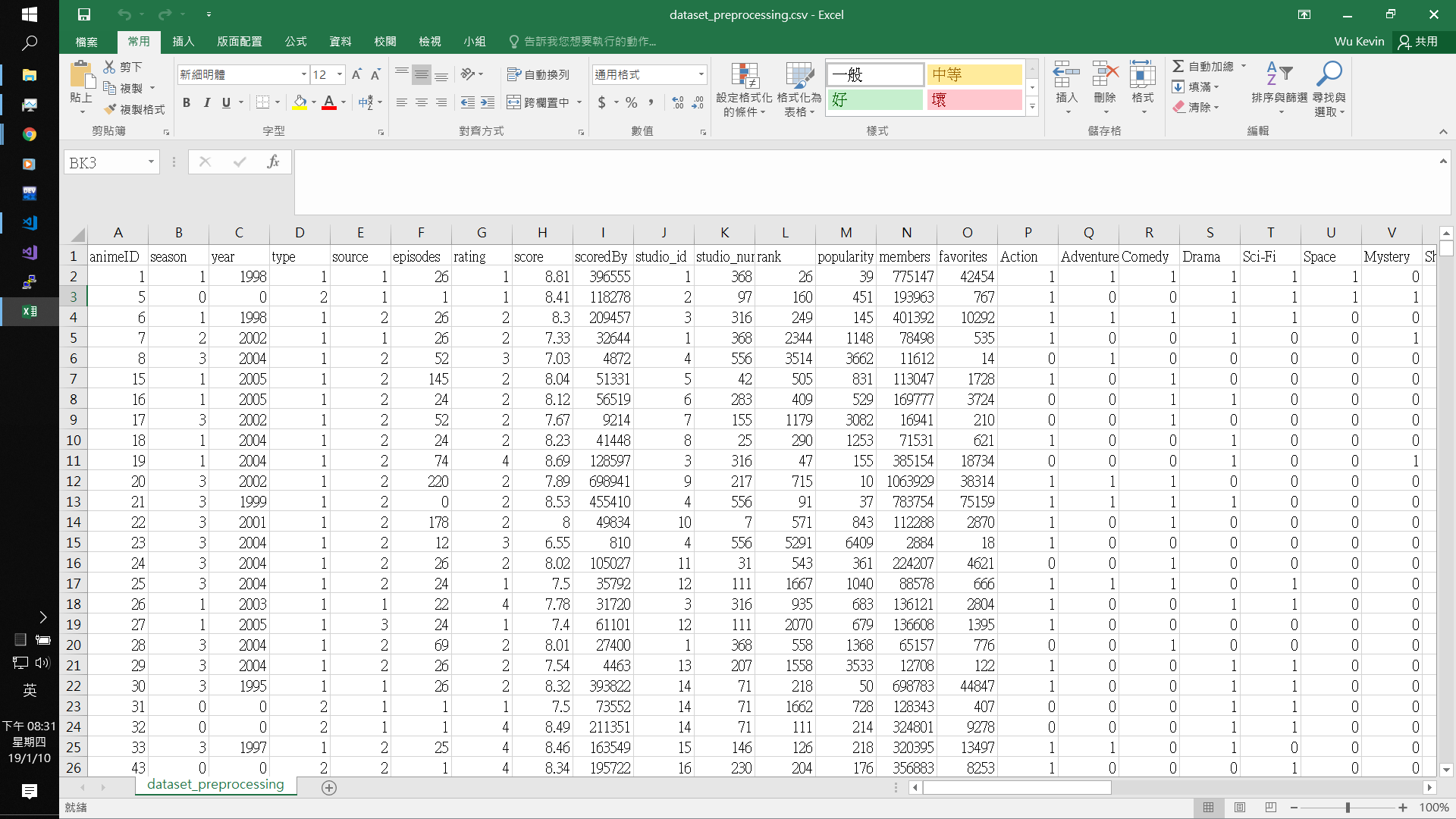
type, source, rating, studio : 為所有項目編號，以數字取代文字

premiered : 分開成兩個欄位 season, year，其中 season 同樣以數字取代

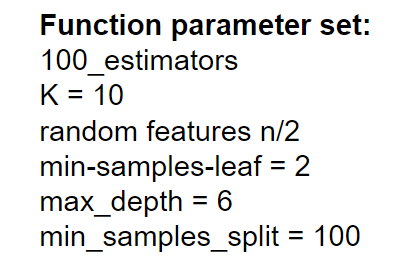
增加欄位

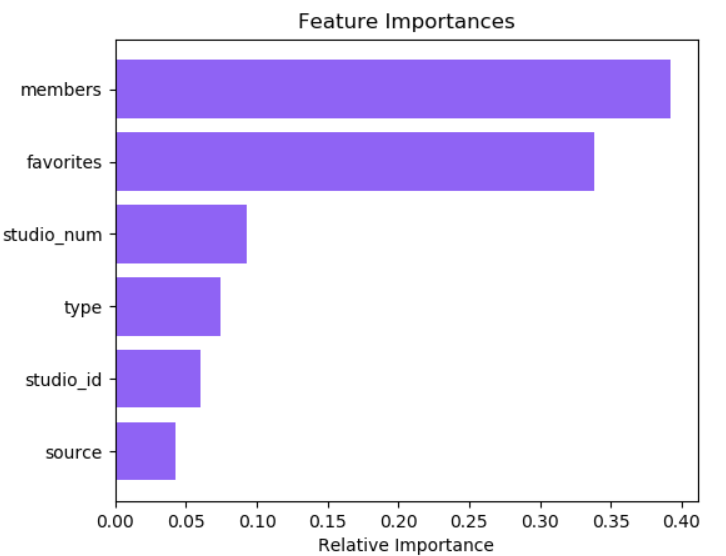
studio\_num : 各個工作室所生產的作品數量

Action~Yaoi : 以 one-hot encoding 的方式將 genre 內的所有項目獨立成一個欄 位，若為1則代表動畫屬於這項分類，反之則為0

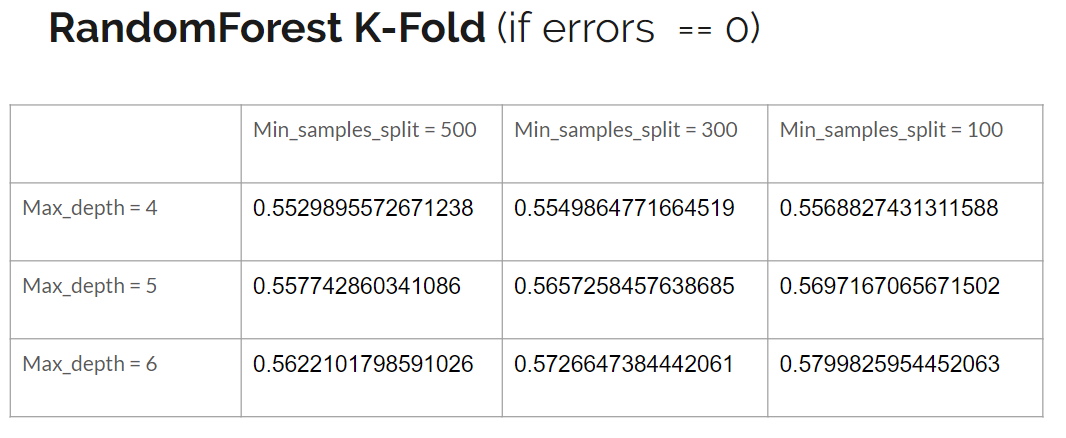


…

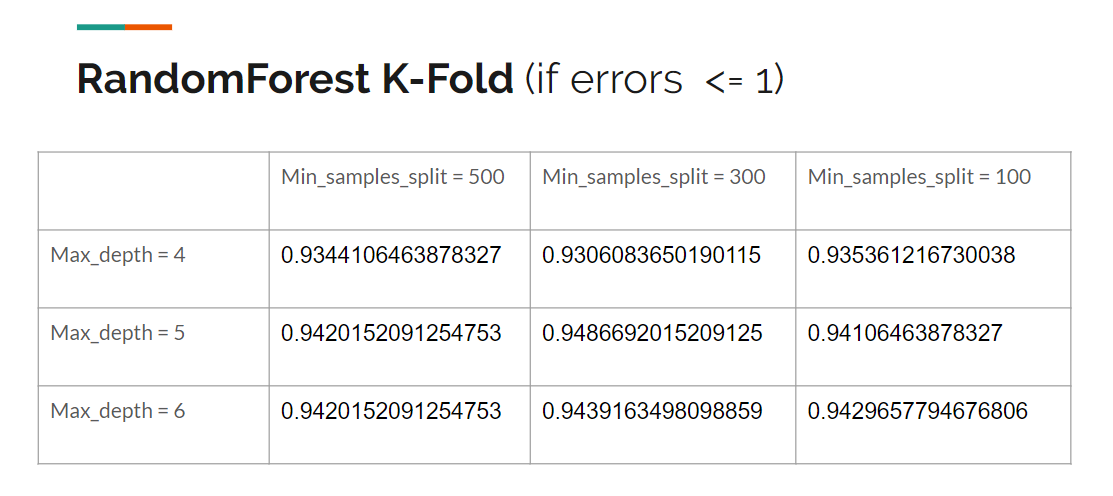
**RandomForest K-Fold**

* Features:
  + type
  + source
  + studio\_id
  + studo\_num
  + members
  + favorites
* Target:
  + score

After testing, the most important feature is members and favorites. The “genres” ‘s importance is very low.



When errors equals to Zero, the accuracy in different Max\_depth or different Min\_samples\_split conditions.



When errors no more than One, the accuracy in different Max\_depth or different Min\_samples\_split conditions.

By RandomForest K-fold, we could get a high accuracy.

KNN (Drop scoreBy < 500, K = 25)

* First try:
  + Attribute:
    - genres(Action, …, Yaoi)(total: 43)
  + Target:
    - score
* Second try:
  + Attribute:
    - (studio\_num)
    - members
  + Target:
    - score
* Third try:
  + Attribute:
    - popularity
    - favorites
  + Target:
    - score

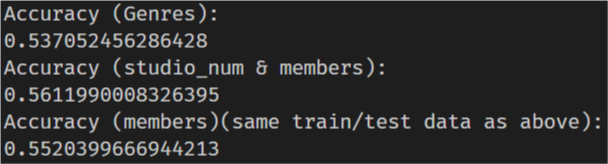
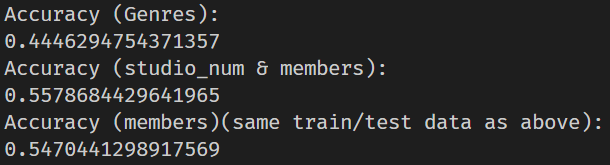
Review:

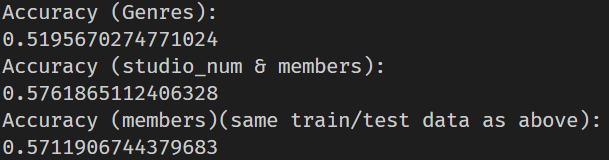
At first, I thought that using “genres” as attributes should have higher accuracy.

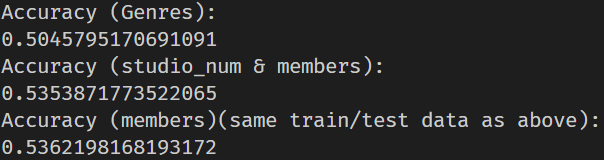
However, the accuracy is around 0.44 to 0.53.

In contrast, using “studio\_num” & “members” as attributes instead gets a higher score.

The accuracy is around 0.53 to 0.57 as the screenshot below:





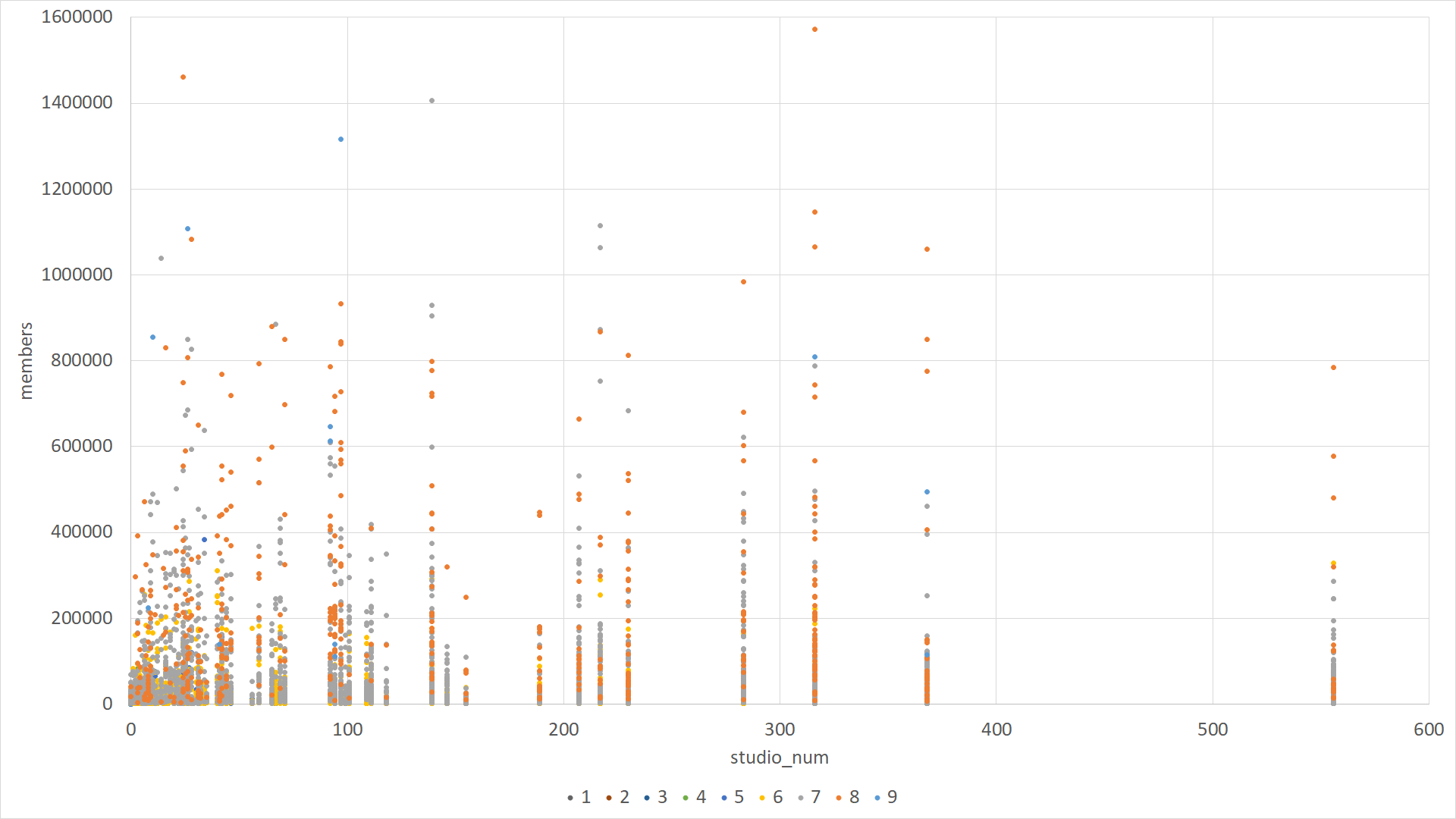


Yet, using “popularity” & “favorites” as attributes gets highest score.  
The accuracy is around 0.55 to 0.59 as the screenshot below:

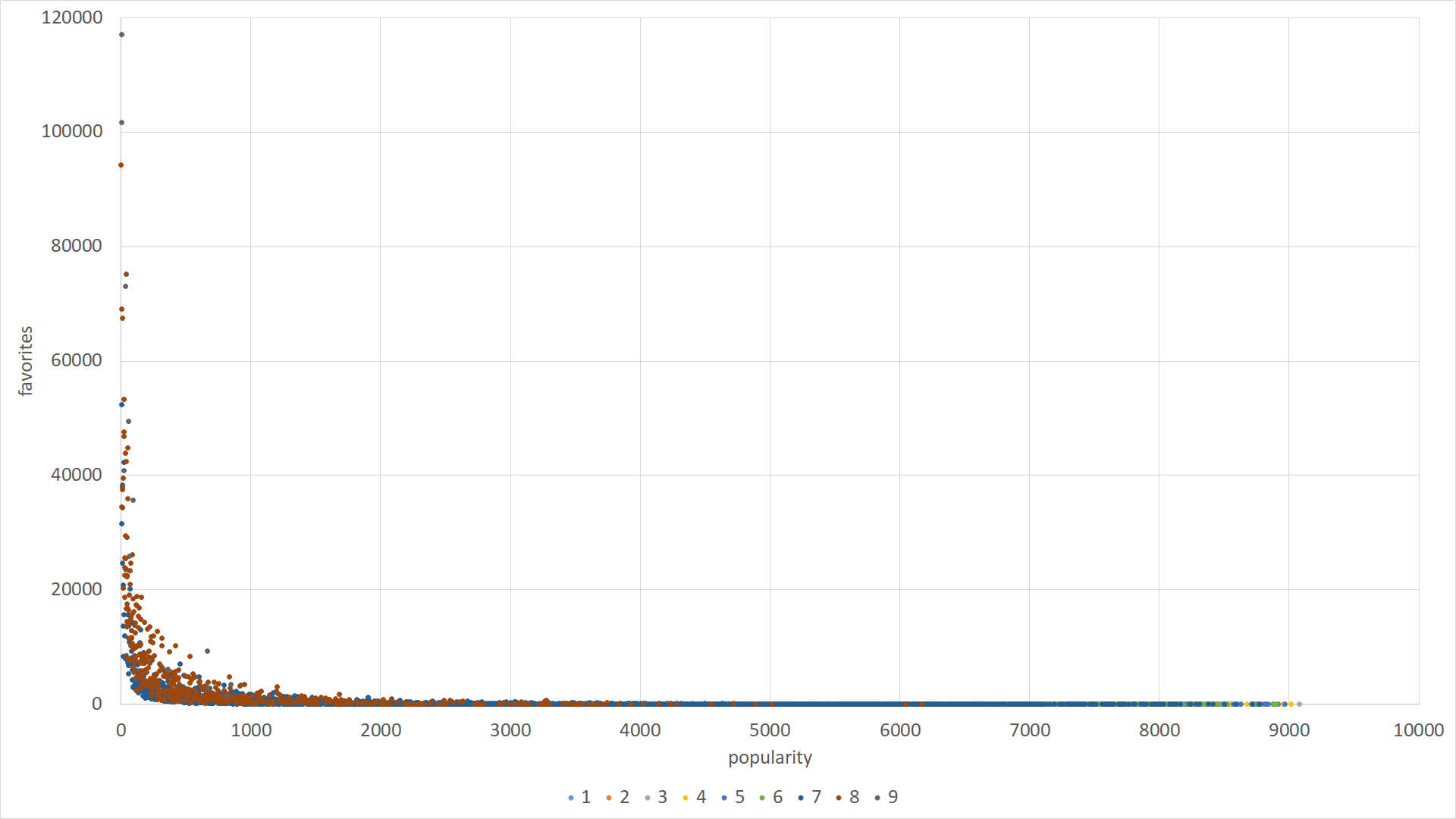


Notice that if I use same training data and same testing data, the accuracy is third > second > first.

Generally, the more people watch, the higher the score.  
However, studios that have done more work are also more likely to get higher score even if “members” is less.



▲This chart shows that if studio makes more animes before(studio\_num), it’s easier to get higher score even if members is less

  
▲And this chart shows that the higher the popularity, the more people collect, the higher the score.

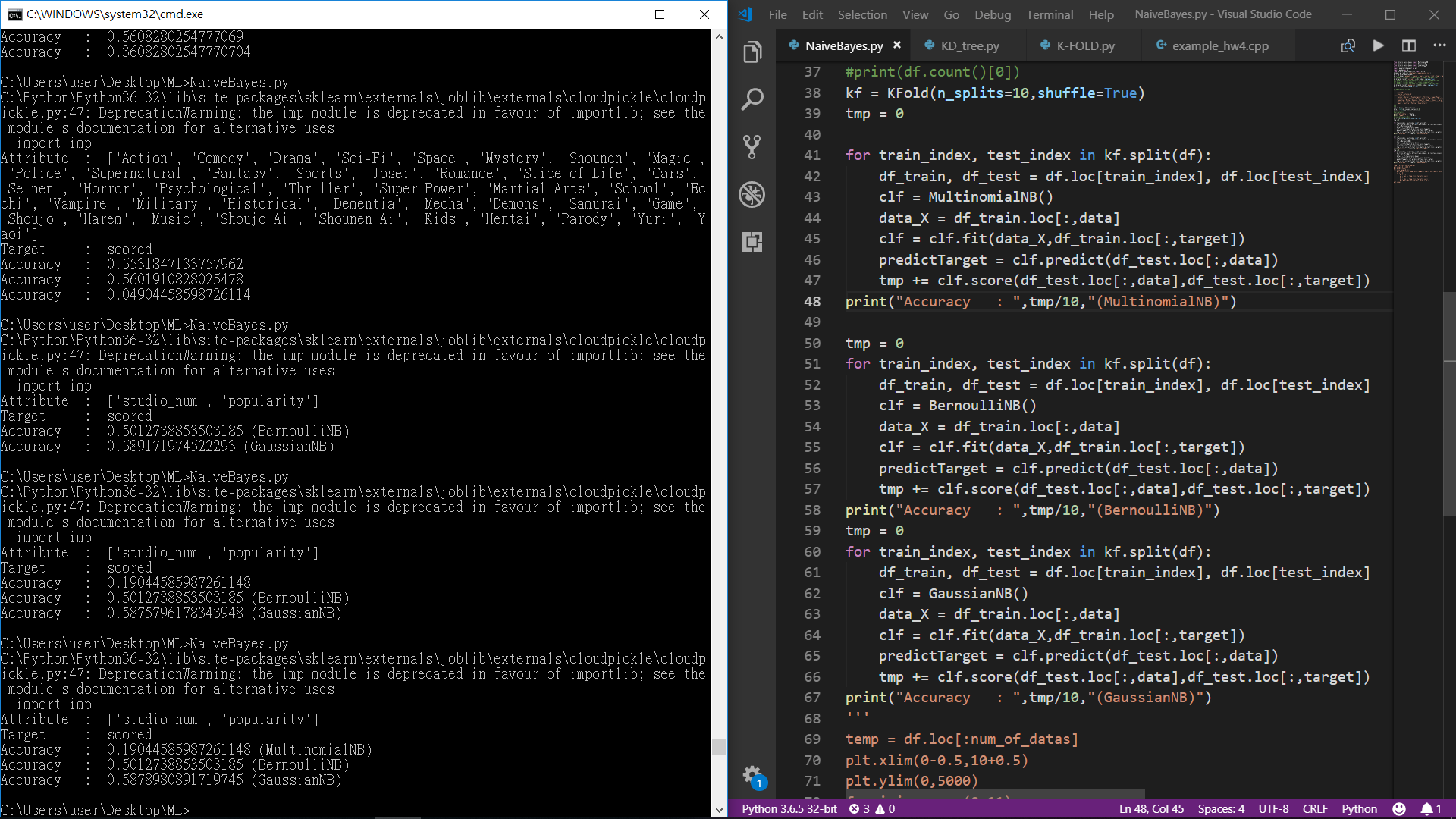
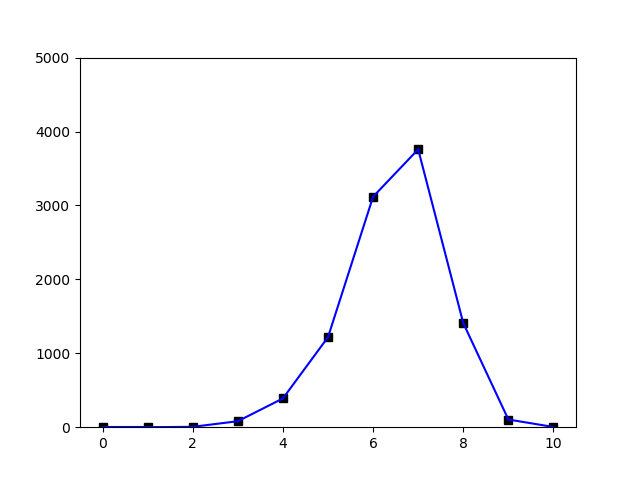
Another thing that I want to mention about is that although I decided to choose K=25 in our briefing, but after the briefing, I went back and thought about K again, I think K=25 might not be a good choice, because the final predicted result has a high probability that it will be 5 to 7 since they are more than 50% of the total. Maybe choosing K=5~7 is better, though the predicted result is not higher than K=25.

***Naive Bayes***

我們選用Naive Bayes中的Gaussian、Multinomial、Bernoulli來做這次的預測。Gaussian通常用於連續型的資料但在離散行也依舊可以使用，Bernoulli所要的資料型態為0&1，Multinomial在一些狀況下類似於Bernoulli。

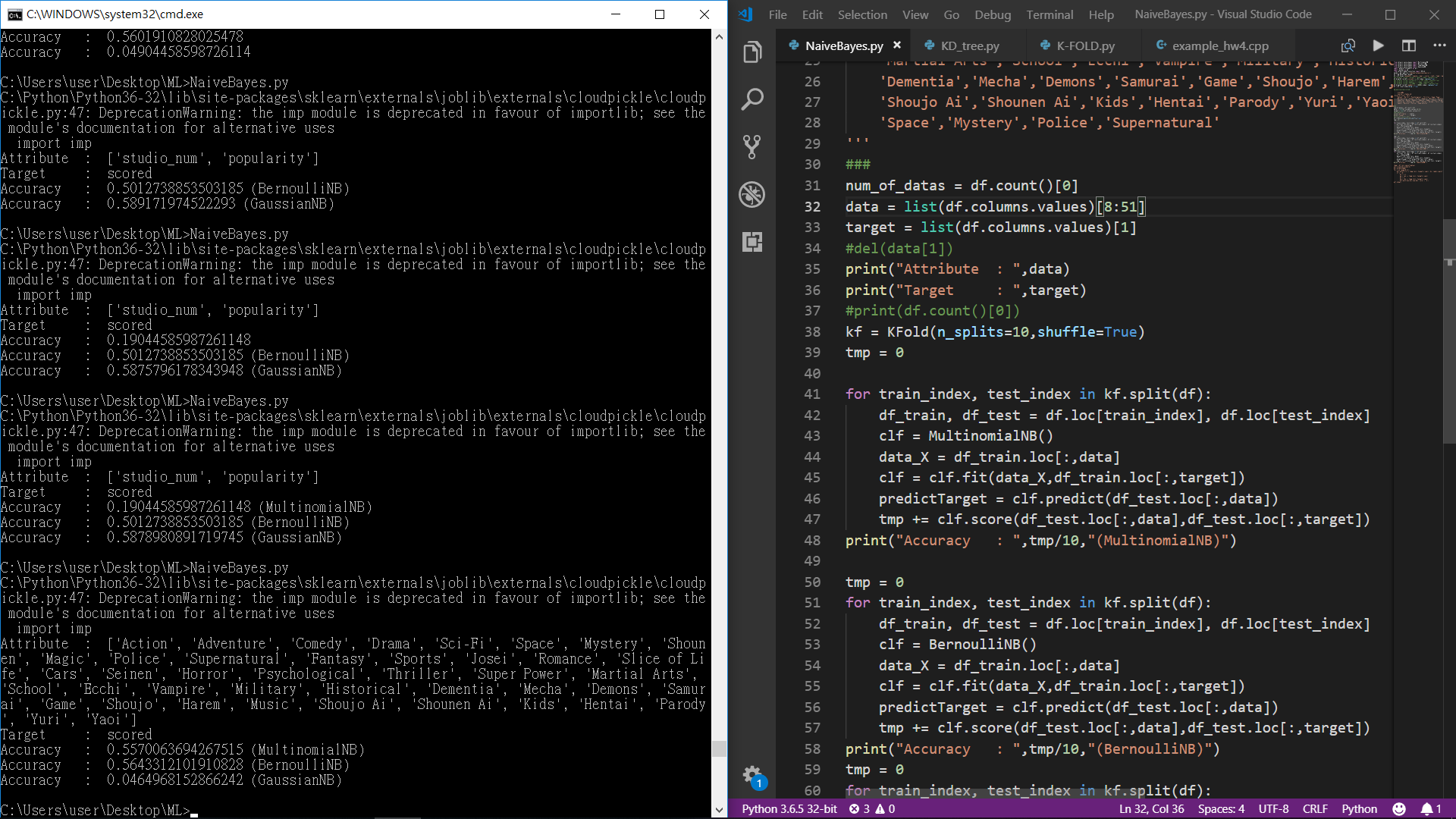
**GaussianNB(k-fold)**

用studio\_num及popularity作為attribute時，由於我們的資料類似於常態分布(左偏)，所以在用Bernoulli時雖然accuracy有五成，但仔細去看它得出的結果會發現它只是全部猜測7分，才能夠得到五成左右的準確率。Gaussian在使用這個資料型態時，最多有接近六成的準確率。



**MultinomialNB & BernoulliNB(k-fold)**

這邊attribute的資料型態是0&1，所以相當適合Bernoulli。在得到的Accuracy可以看出Multinomial及Bernoulli大約五成五，而Gaussian則不太適合這個資料型態，準確率不到5%。

****

從結果來看Naive Bayes不太適合這次的預測，不論是Gaussian、Multinomial還是Bernoulli準確率都難以突破六成。

**Linear regression**

(一) 單變數迴歸

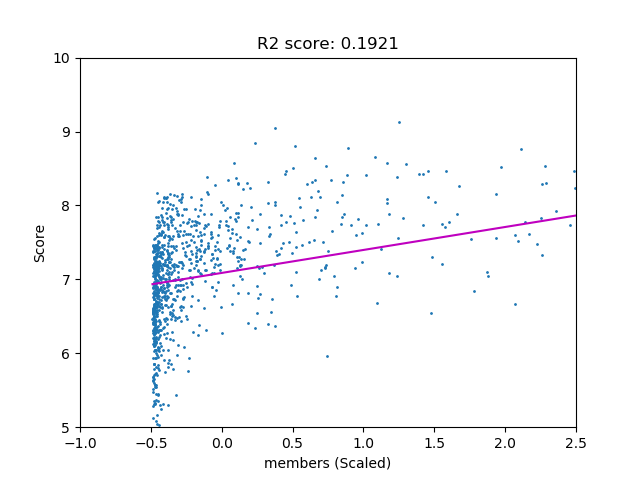
針對members、popularity、favorites、studio\_num、episodes，這些動畫開播之前就可以取得的5個離散型變數，分別以他們五個作為單變數線性迴歸的自變數，來預測動畫的評分。

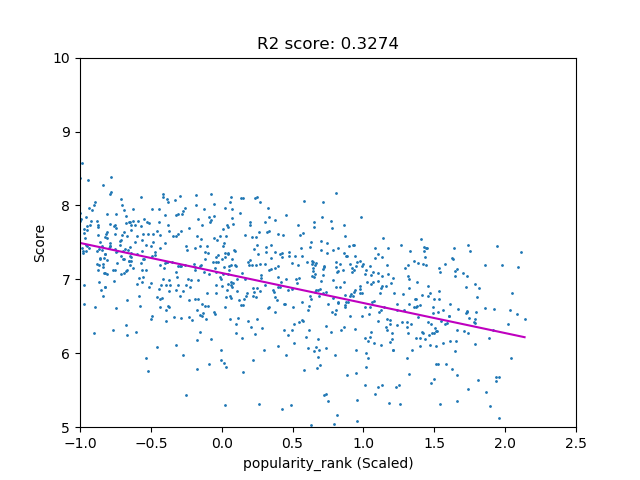
在做迴歸分析之前，先以下述的標準篩掉資料中的離群值：

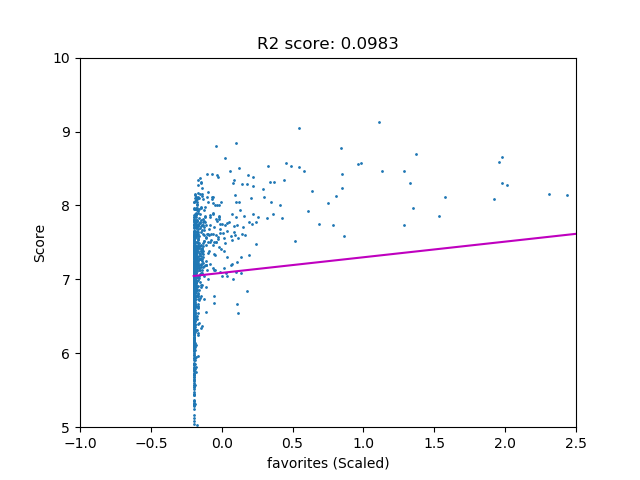
* Drop scoreBy <= 1000
* n\_data = 5007 (i.e. 共有5007筆資料)
* 80% of training set, 20% of testing set

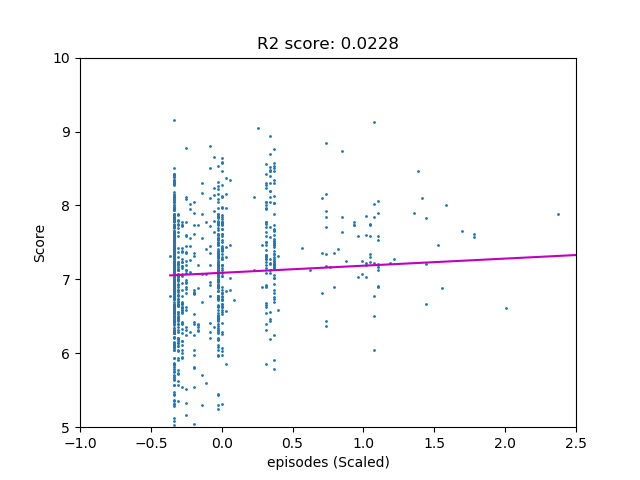
以下是單變數線性迴歸分析的結果：

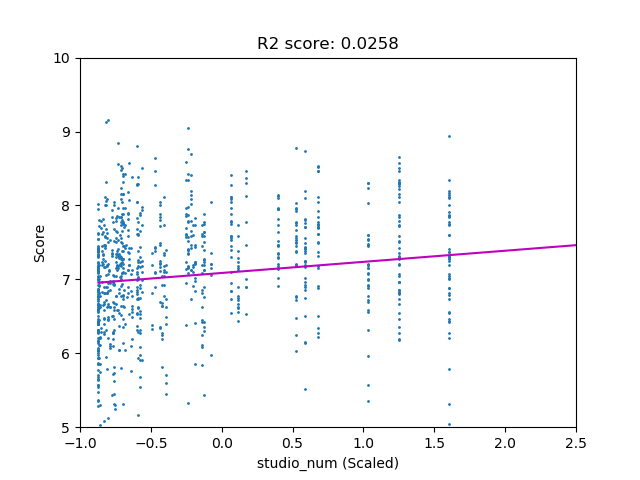
(使用檔案：SingleVariable\_Regression.py)











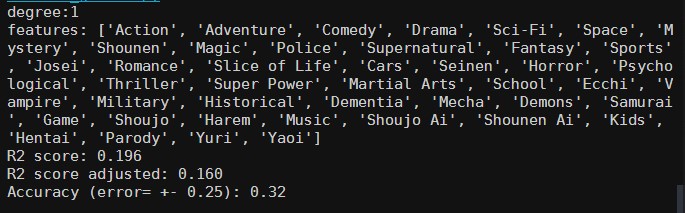
可以看的出來其中members和popularity，對於預測評分來說，是較為重要的自變數。

(二) 多變數迴歸

使用檔案：SingleVariable\_Regression.py

這裡我們針對genres ，也就是 [Action, …, Yuri]這43個binary variable，以他們為自變數，作多變數線性迴歸來預測動畫評分。

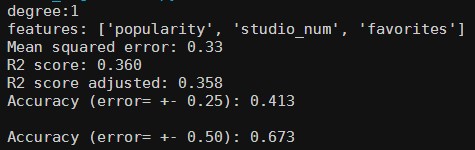
以下是預測結果：(使用檔案：MultiVariable\_Regression\_genres.py)



大約可獲得三成的準確度。

另外，針對第一部分單變數線性迴歸，經過窮舉各種自變數的組合，我找出以[studio\_num, popularity, favorites]這三個變數做為自變數的話，可以獲得最高的準確度，以下是預測結果：

(使用檔案：MultiVariable\_Regression.py)

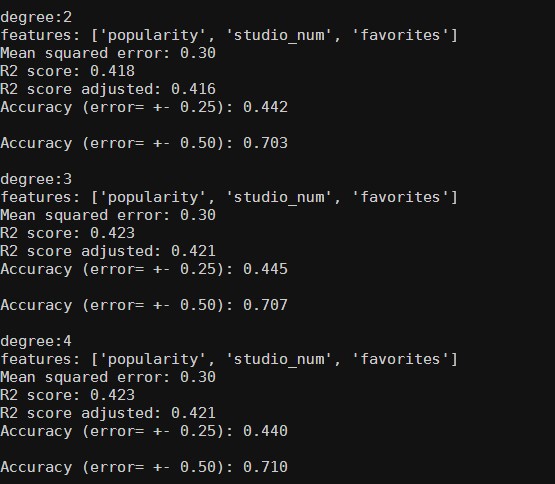


大約有41%的準確度，若是放寬誤差到正負0.5分，則可來到67%的準確度

**Polynomial regression**

針對前面做過多變數線性迴歸的 [studio\_num, popularity, favorites]，以他們自變數，分別做二、三、四次方的多變數迴歸來預測動畫評分。

以下是預測結果：(使用檔案：MultiVariable\_Regression.py)



|  |  |  |  |
| --- | --- | --- | --- |
| Polynomial degree | 2 | 3 | 4 |
| **R2\_score** | **0.418** | **0.423** | **0.423** |
| R2\_score\_adjusted | 0.416 | 0.421 | 0.421 |
| Accuracy (±0.25) | 0.442 | 0.445 | 0.440 |
| Accuracy (±0.5) | 0.703 | 0.707 | 0.710 |

可以看到約有44%的準確度，在誤差為正負0.5分時，準確度可高達70%

另外也可以觀察到，除了從一次方提高至二次方時，R^2\_Score和accuracy有略微提高之外，在二次方以上的狀況時，提高迴歸式的次方，對預測的準確度以及R^2\_Score並沒有太大的幫助。

Conclusion

* Cannot predict the score precisely.
* Can still roughly predict the score of an unreleased anime, with enough certain attributes
* Improvement:
  + collect more attribute : voice actor/actress, director, etc.
  + source
  + model