

# Douban book crawler and analysis

Zhong Junyan\*

Faculty of Information Technology , M.U.S.T

---

## Abstract

*With the development of the society, the demand of finding a good book is increasing. In order to find a good book, I firstly use python to crawl book details from the Douban book. Secondly, I made data cleaning and then store the data into the MySQL database. Thirdly I make briefly data analysis for the data. Last but not least, I add the sell ratings information for my data from Amazon and then use  $k$ -means cluster methods to cluster the data and then I divide my data into train set and test set and use some classification methods to classify my data. Finally I find the Random forest classification make the best performance in the classification of test set.*

**Keywords:** Data crawl, Data storage, MySQL database ,  $K$ -means cluster, Random forest classification

---

## 1 Introduction

If we want to find a good book we usually refer to the Douban book website. And if we want to know the popularity of a book we would refer to the rank

---

\*Student ID: 1909853GIM20007

on the Amazon. Therefore I got a book list and its details on Douban book and then get the rank information from the Amazon. Then I make some analysis based on the data. What's more, in order to find the book which has less comments on Douban book and high rank on the Amazon I make cluster analysis and classification analysis for my data. I would also show the details and the optimization of the algorithms.

## 2 Data Crawl and analysis

The website of the Douban book is: <https://www.douban.com/doulist/1264675>.

In order to crawl the data from the website we firstly analyze the structure of the website:

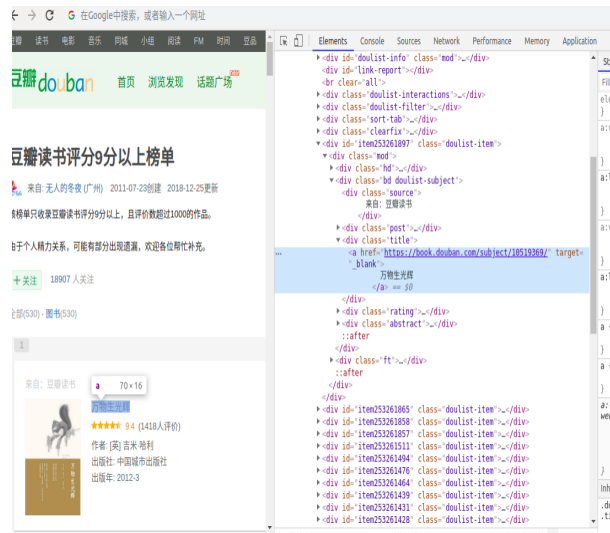


Figure 1: Douban book website.

## 2.1 Data crawl

We can see that the list is controlled by the start= so we can use a loop to crawl the information for the book. Then we can see that the details of the book is shown in the graph above. Therefore we can use the regular expression to crawl the data[1]:

```
1 def getDetail(html):
2     detailList=re.findall(r'<ahref="(https.*?)".*?target="_blank">.*?</a>',html,re.S)
3     newDetailList = []
4     for index,item in enumerate(detailList):
5         if item.find("subject") != -1 and index % 2!=0:
6             newDetailList.append(item);
7 def getPublishYear(html):
8     publishYearList = re.findall(r'<span class="pl
9     ">.*?</span>(.*?)<br/>',html,re.S)
10    return publishYearList
```

Listing 1: Data crawl

## 2.2 Data storage and cleaning

After I crawl the data I use the MySQL database to store the data. Firstly I create a table in the database:

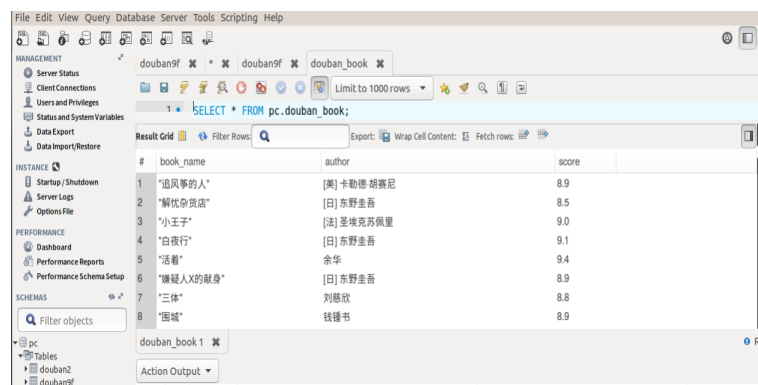
```

1 import pymysql
2 conn = pymysql.connect(host = '127.0.0.1',user = 'root',
    passwd = '123456',db = 'sys',port = 3306,charset = 'utf8
    ')
3 cursor = conn.cursor()
4 sql = "create table douban9f(book_name text,score text,num
    text,picturelink text,publisher text,pulishyear text,
    ISBN text)engine innodb default charset=utf8"
5 try:
6     cursor.execute(sql)
7     conn.commit()
8 except:
9     conn.rollback()
10 conn.close()

```

Listing 2: Creat table

Then I storage the csv file into the database as follows:



The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query: `SELECT * FROM pc.douban_book;`. The 'Result Grid' shows the following data:

#	book_name	author	score
1	"追风筝的人"	[美] 卡勒德·胡赛尼	8.9
2	"解忧杂货店"	[日] 东野圭吾	8.5
3	"小王子"	[法] 圣埃克苏佩里	9.0
4	"白夜行"	[日] 东野圭吾	9.1
5	"活着"	余华	9.4
6	"嫌疑人X的献身"	[日] 东野圭吾	8.9
7	"三体"	刘慈欣	8.8
8	"围城"	钱锺书	8.9

The interface also shows a sidebar with 'MANAGEMENT' and 'SCHEMAS' tabs, and a 'Result Grid' tab selected. The 'Action Output' tab is also visible at the bottom.

Figure 2: Data storage.

Then I remove the null value in my data:

```
1 db=pd.read_csv("9p.csv",encoding='utf-8')
2 db=db.dropna()
3 db.publishyear=pd.to_numeric(db.publishyear,downcast='
  integer')
```

Listing 3: Remove the null value

## 2.3 Data analysis

Then we can draw the distribution of the score of the book as follows:

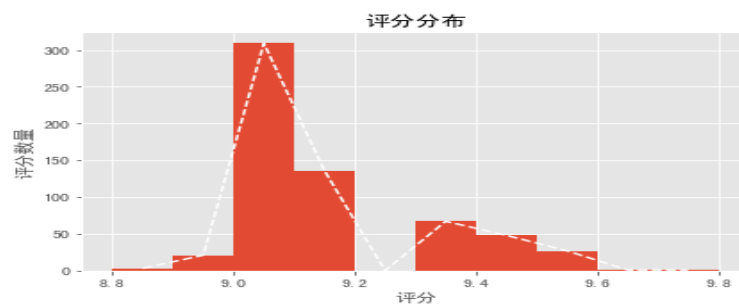


Figure 3: Score distribution.

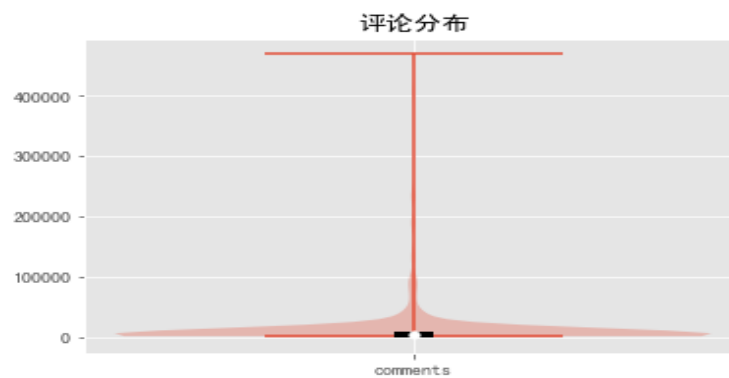


Figure 4: Comments distribution.

We can see that the score of the book densely distributed on the interval of 9.0 to 9.2 and we can see that many books comments is less than 10000.

Then I plot the following graph base on my data and get the top10 books which have the most comments and the average comments of each year:

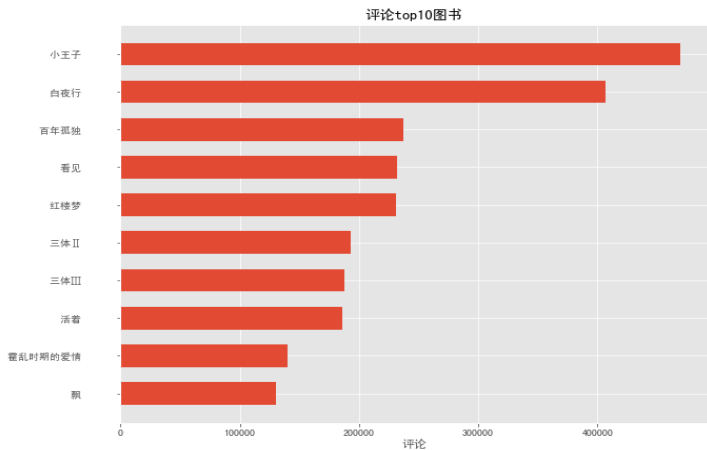


Figure 5: Top10 comments books.

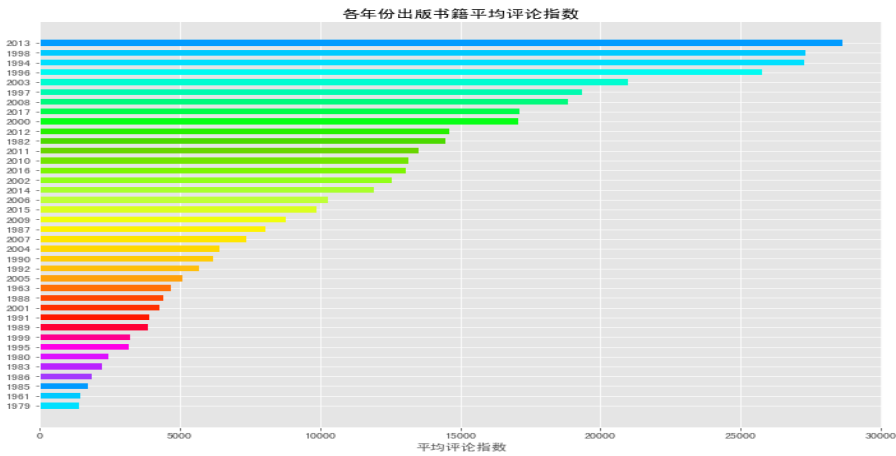


Figure 6: Average comments of each year.

## **2.4 Conclusion**

We can draw a conclusion that the book with high score may not have high popularity. Therefore, we shall create a new way to recommend books to others in a new way which both refer to the popularity of the book and the score of the book.

## 3 Clustering and classification

### 3.1 PCA

PCA is a statistical process[2] that uses an orthogonal transformation to transform the observations set of possibly related variables into a set of linearly uncorrelated variables. In order to reduce the dimensionality of features I use the PCA to extract the two most influential features for analysis.

```
1 pca=PCA(n_components=2)
2 X=data[cols]
3 NX=pca.fit_transform(X)
4 NX=abs(NX)
```

Listing 4: PCA

In order to simplify the following work, I turn eigenvectors positive value.

### 3.2 Clustering

$k$ -means clustering[3] is a well-known phenomenon in geometric data, and has application in machine learning. To simplify the problem, we define the distance function in standardized euclidean metric:

$$\begin{aligned} d(p, q) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \end{aligned}$$

We use the knn function to select the best value of K as follows:



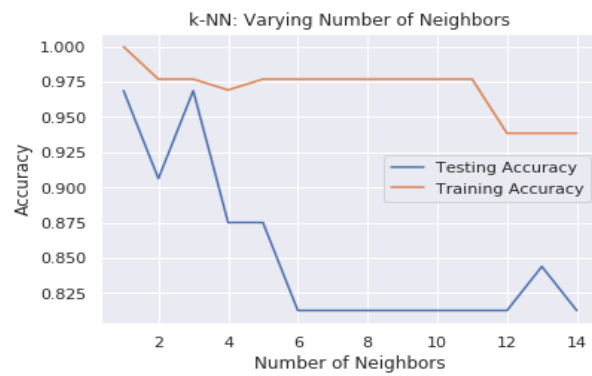


Figure 7: Select best K value.

We can that  $K=3$  is the best value for the  $k$ -means clustering so we use  $K=3$  to cluster the data.

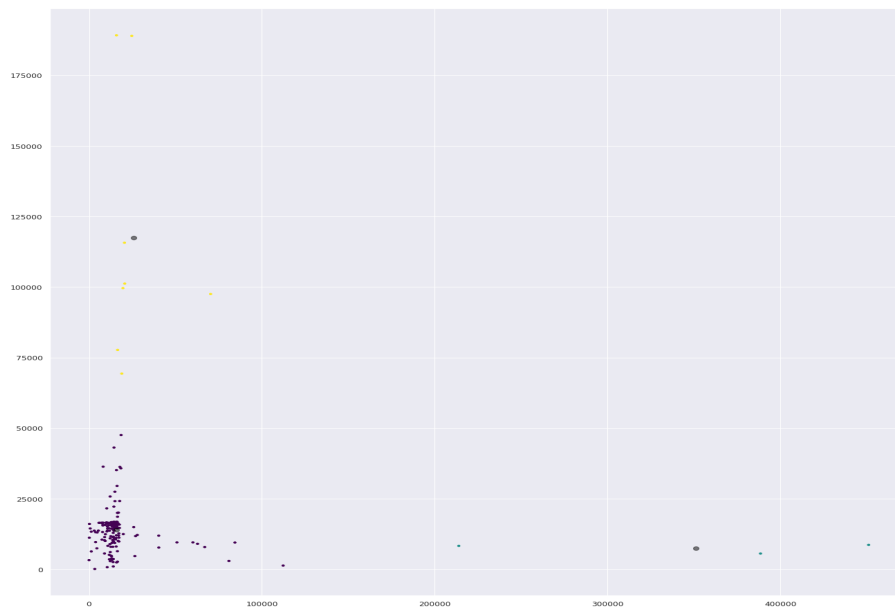


Figure 8: Cluster plot.

### 3.3 Classification

In order to find some books which are unpopular in douban but it's popular in amazon I select the most dense clusters for analysis. Then I select the first 130 samples as the train data set and rest of the data as the test data set. Then I use four classification methods[4] to classify the data and see which classification methods make the best performance in the test data set.

#### 3.3.1 Linear-SVM classification method

Firstly I use Linear-SVM classification method[5] to classify the data.

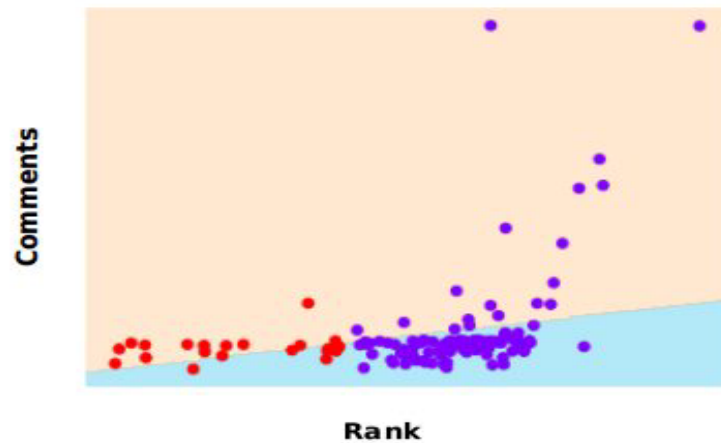


Figure 9: LSVM classification.

### 3.4 logistic regression classification method

The second method I used is the logistic regression classification.

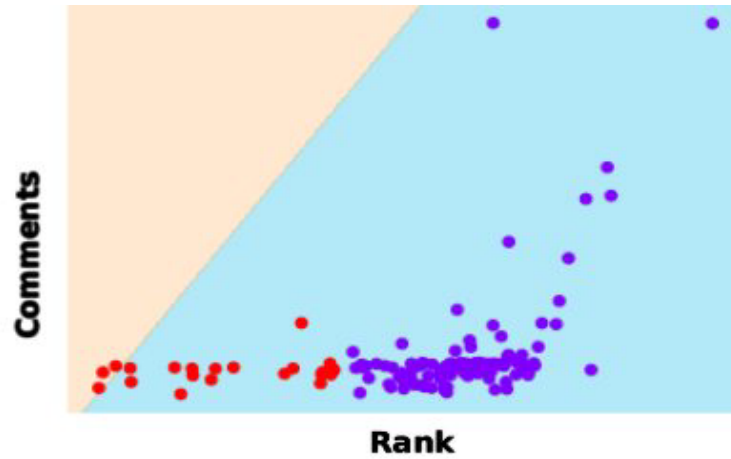


Figure 10: Logistic regression classification.

### 3.5 Decision tree classification method

The third method I used is the Decision tree method. And we can see that the effect of this method has significantly improved than the above two methods.

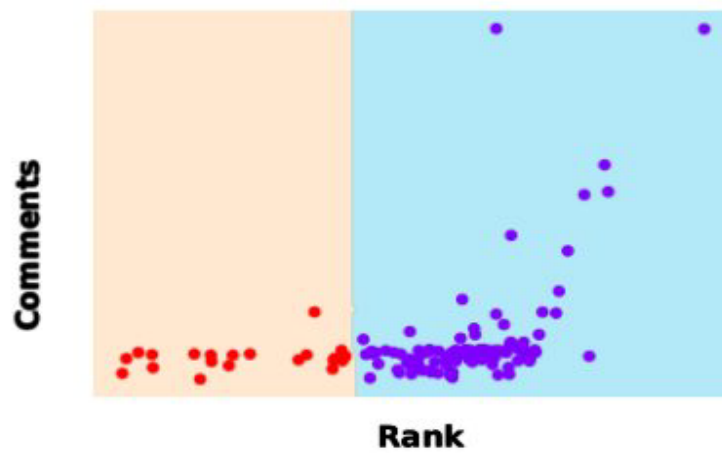


Figure 11: Decision tree classification train set.

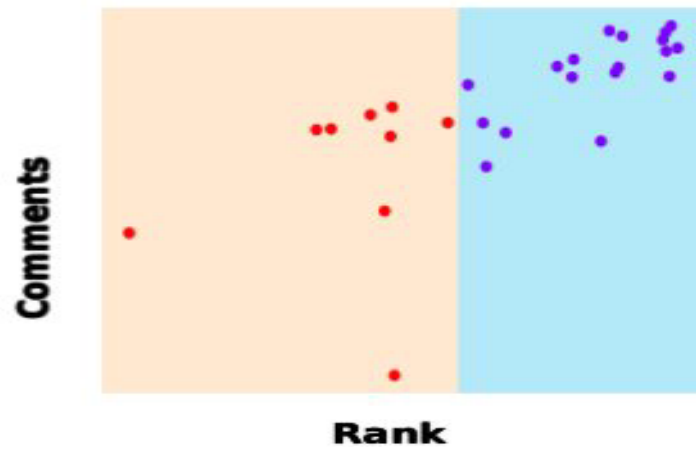


Figure 12: Decision tree classification test set.

### 3.6 Random forest classification method

The last method I used is the Random forest classification method[6]. This algorithm is an improved algorithm based on the decision tree classification algorithm. It aggregates the votes from different decision trees to decide the final class of the test object.

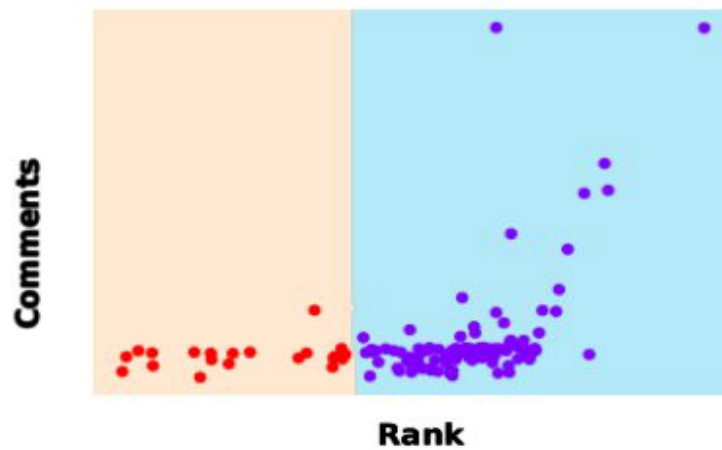


Figure 13: Random forest classification train set.

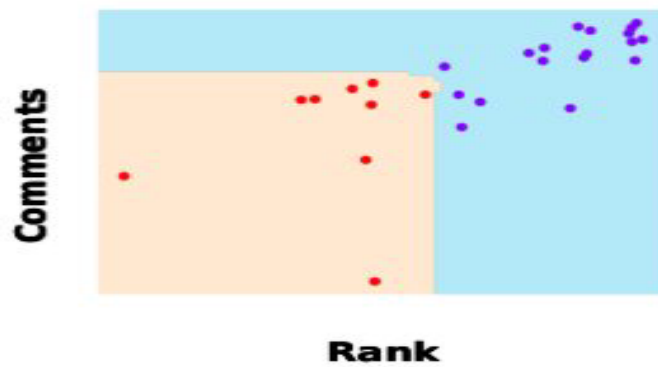


Figure 14: Random forest classification test set.

Then we can see the Random forest as follows:

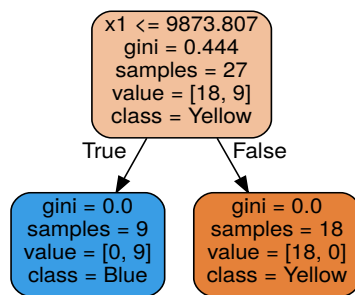


Figure 15: Random forest.

### 3.7 Conclusion

We can draw a conclusion that the random forest classification make the greatest performance in the classification test. And the score for each method performance are as follows:

methods	score
Lsvm	0.875
Logistic regression	0.75
Decision tree	0.875
Random forest	1

Table 1: Classification methods comparsion

## 4 Discussion

Our web crawler can be improved in the following way:

1.Firstly,we should set a reasonal sleep function which can aviod the block of the website.

2.Secondly,we can turn our Single thread web crawler to multi thread web crawler, which can improve the speed of our web crawler.

3.Lastly,we may try using distribution web crawler(like redis) when we crawl Large amounts of data .

And in the  $k$ -means clustering we can use the k-d tree methods to improve the speed of the algorithm.

## 5 Conclusion

My project is including data crawling and data storage as well as data analysis.Then I do use clustering and classification for my data.Although the work is hard I try my best to finishing the work.And I learn a lot in this project and I would keep making progress in coding and learning other tools in reasearch.

## References

- [1] Ryan Mitchell. *Web Scraping with Python: Collecting More Data from the Modern Web*. " O'Reilly Media, Inc.", 2018.
- [2] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [3] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.
- [4] Jake VanderPlas. *Python data science handbook: essential tools for working with data*. " O'Reilly Media, Inc.", 2016.
- [5] Yin-Wen Chang and Chih-Jen Lin. Feature ranking using linear svm. In *Causation and Prediction Challenge*, pages 53–64, 2008.
- [6] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.