Data Analysis for douban reading data

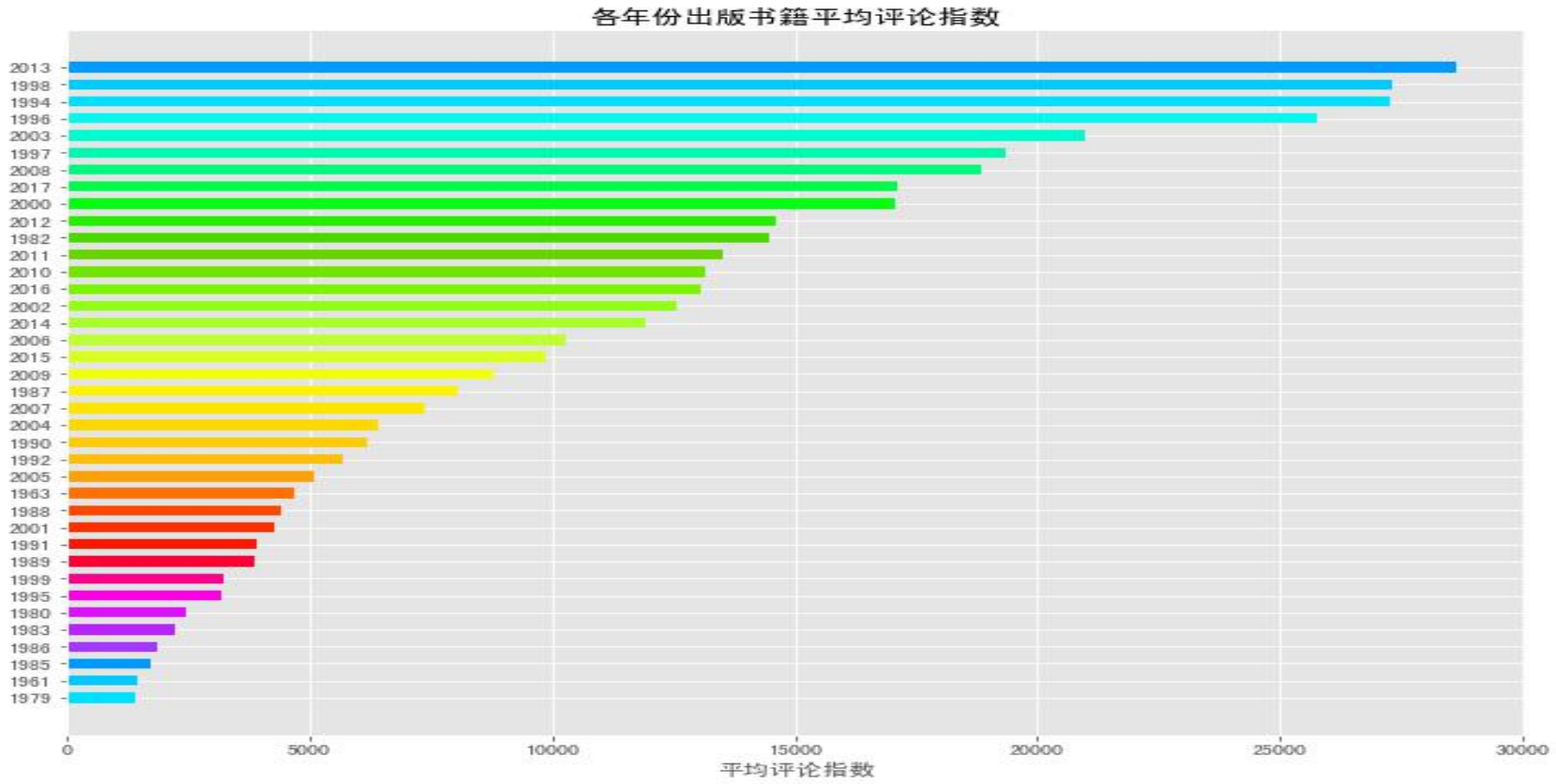
JUNYAN ZHONG

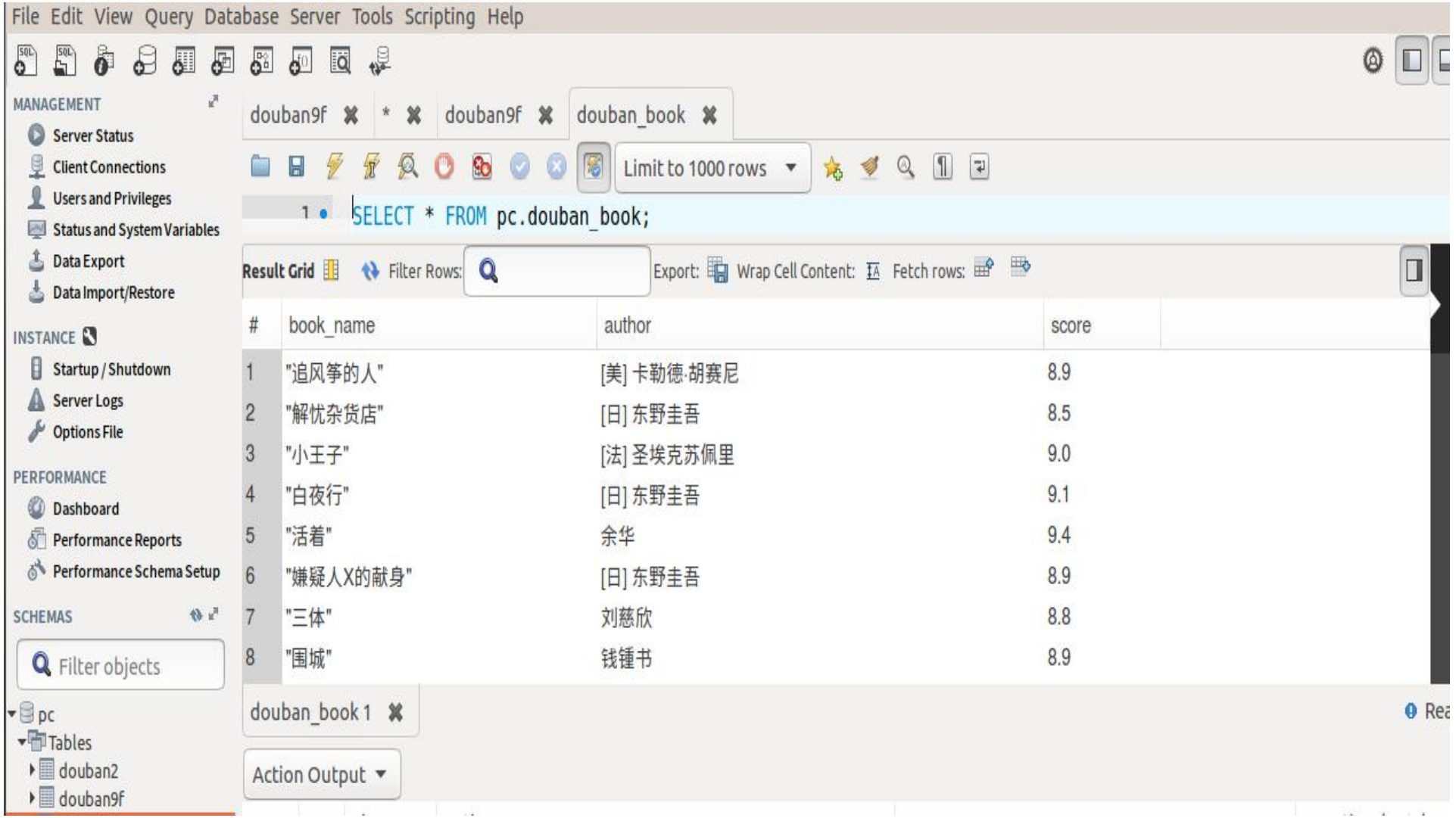
My code url

- https://github.com/W55699/douban_book-web-crawler

Briefly review of last presentation

- In last presentation I have crawled some data from the book of douban and make some briefly analysis for the data.

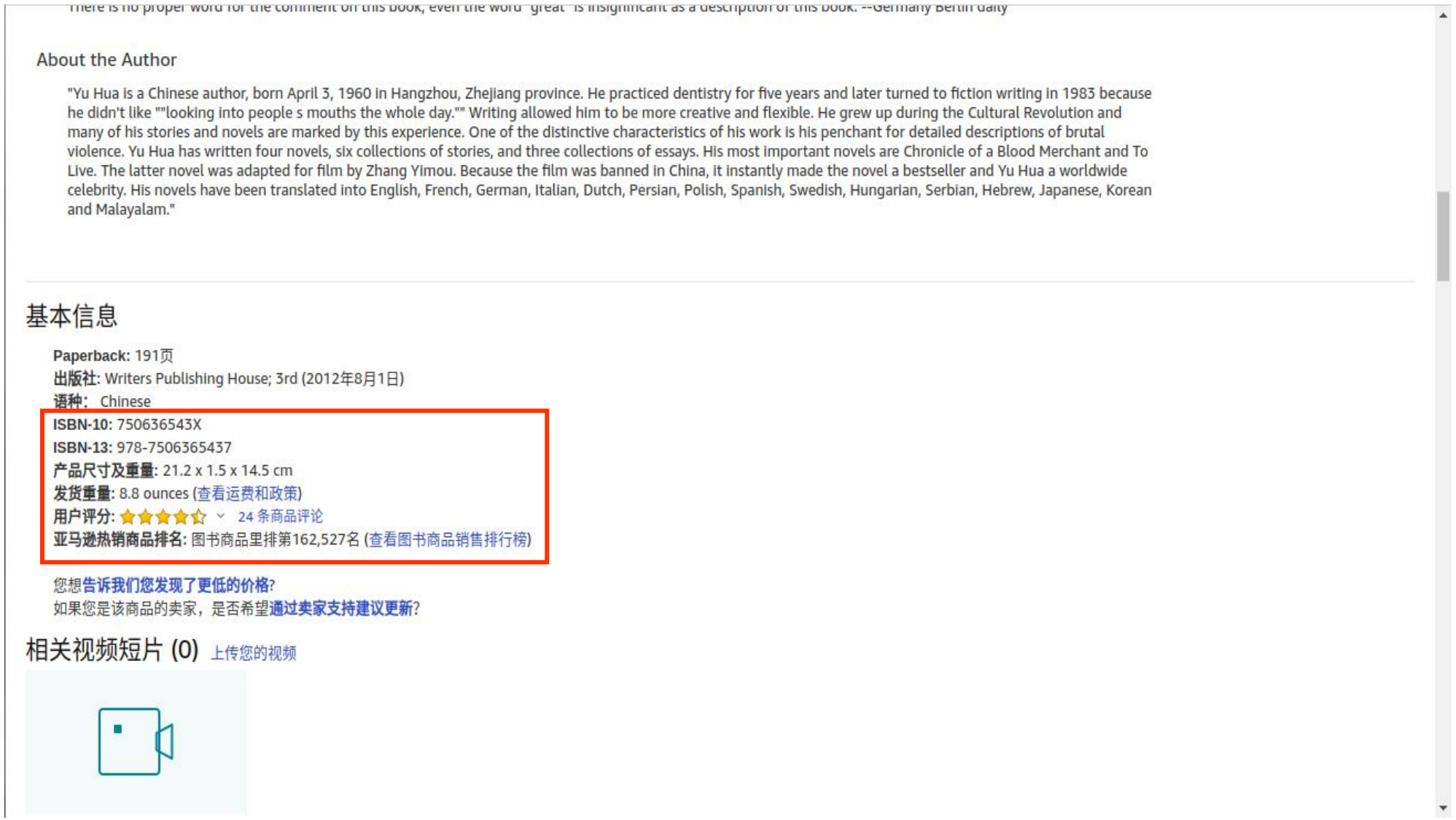



File Edit View Query Database Server Tools Scripting Help
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
pc
Tables
douban2
douban9f
douban9f
*
douban9f
douban_book
Limit to 1000 rows
1 • SELECT * FROM pc.douban_book;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
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
douban_book 1
Action Output

Some preparation work

- In order to find some more valueable information for my data I decide to add some new data for my dataset.I deciede to add the kindle books sell rank to my dataset.

I use the ISBN code to locate the book and then get the rank of the book.



There is no proper word for the comment on this book, even the word "great" is insignificant as a description of this book. --Germany Berlin daily

About the Author

"Yu Hua is a Chinese author, born April 3, 1960 in Hangzhou, Zhejiang province. He practiced dentistry for five years and later turned to fiction writing in 1983 because he didn't like ""looking into people s mouths the whole day."" Writing allowed him to be more creative and flexible. He grew up during the Cultural Revolution and many of his stories and novels are marked by this experience. One of the distinctive characteristics of his work is his penchant for detailed descriptions of brutal violence. Yu Hua has written four novels, six collections of stories, and three collections of essays. His most important novels are Chronicle of a Blood Merchant and To Live. The latter novel was adapted for film by Zhang Yimou. Because the film was banned in China, it instantly made the novel a bestseller and Yu Hua a worldwide celebrity. His novels have been translated into English, French, German, Italian, Dutch, Persian, Polish, Spanish, Swedish, Hungarian, Serbian, Hebrew, Japanese, Korean and Malayalam."

基本信息

Paperback: 191页
出版社: Writers Publishing House; 3rd (2012年8月1日)
语种: Chinese
ISBN-10: 750636543X
ISBN-13: 978-7506365437
产品尺寸及重量: 21.2 x 1.5 x 14.5 cm
发货重量: 8.8 ounces (查看运费和政策)
用户评分: ★★★★☆ 24 条商品评论
亚马逊热销商品排名: 图书商品里排第162,527名 (查看图书商品销售排行榜)

您想**告诉我们您发现了更低的价格**?
如果您是该商品的卖家，是否希望**通过卖家支持建议更新**?

相关视频短片 (0) 上传您的视频

My data set

```
In [33]: data.head()
```

Out[33]:

	book_name	score	num	picturelink	publisher	publishyear	ISBN	rank
0	呼兰河传	9.0	2069	https://img3.doubanio.com/view/subject/l/publi...	中国青年出版社	2003	9.787500e+12	206141
1	孽子	9.1	10902	https://img3.doubanio.com/view/subject/l/publi...	广西师范大学出版社	2010	9.787560e+12	205948
2	罗马人的故事2	9.2	2936	https://img1.doubanio.com/view/subject/l/publi...	中信出版社	2011	9.787510e+12	132734
3	木心作品八种	9.3	2121	https://img1.doubanio.com/view/subject/l/publi...	广西师范大学出版社	2009	9.787560e+12	118278
4	万物既聪慧又奇妙	9.2	3059	https://img3.doubanio.com/view/subject/l/publi...	中国城市出版社	2010	9.787510e+12	116627

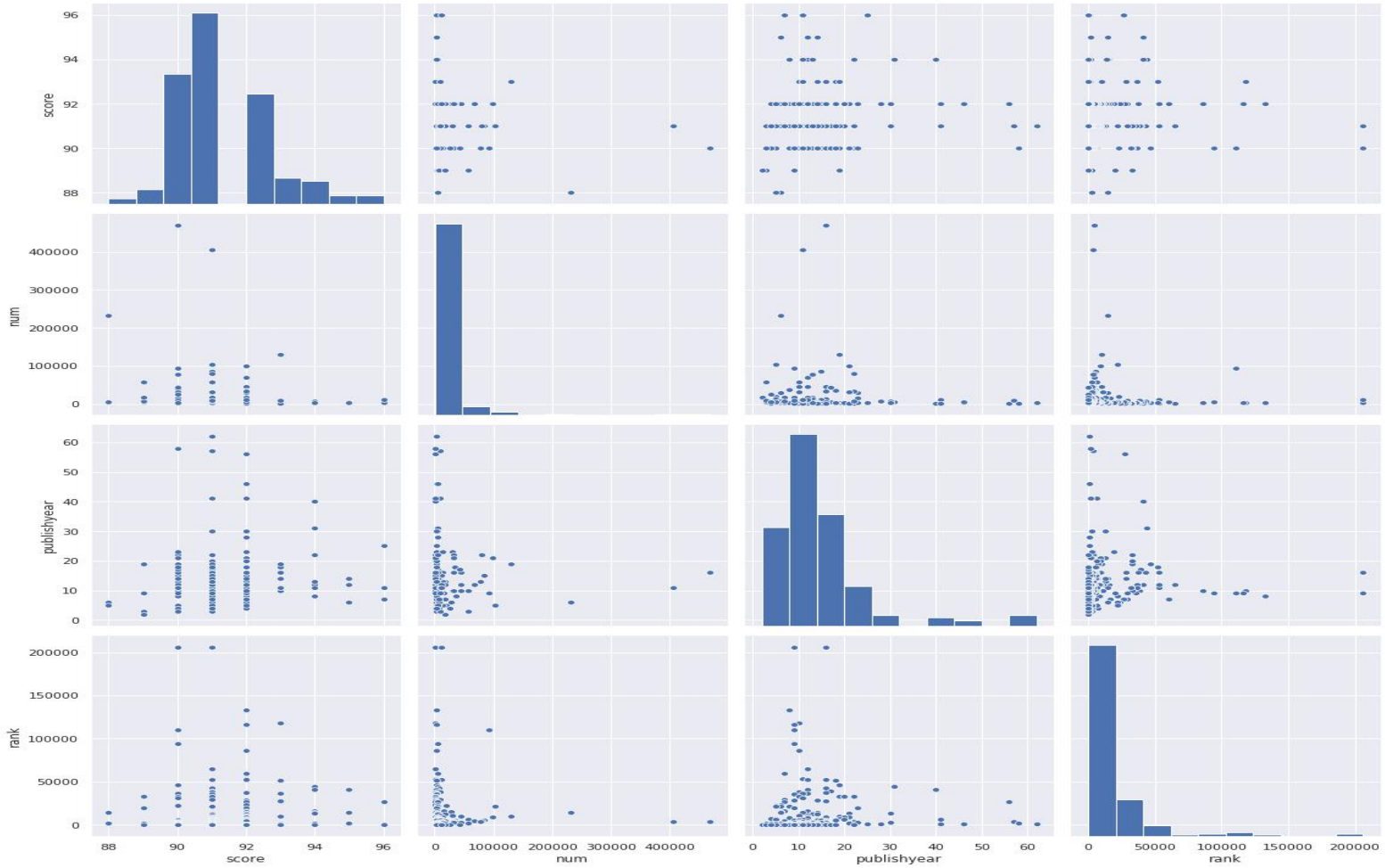
My goal

- 1.I want to use the linear regression to find the relation between rank and another three colnumns(the socre,the number of comments and the publish year of the book).

- 2 I want to find some books which are unpopular in douban but it's popular in amazon.

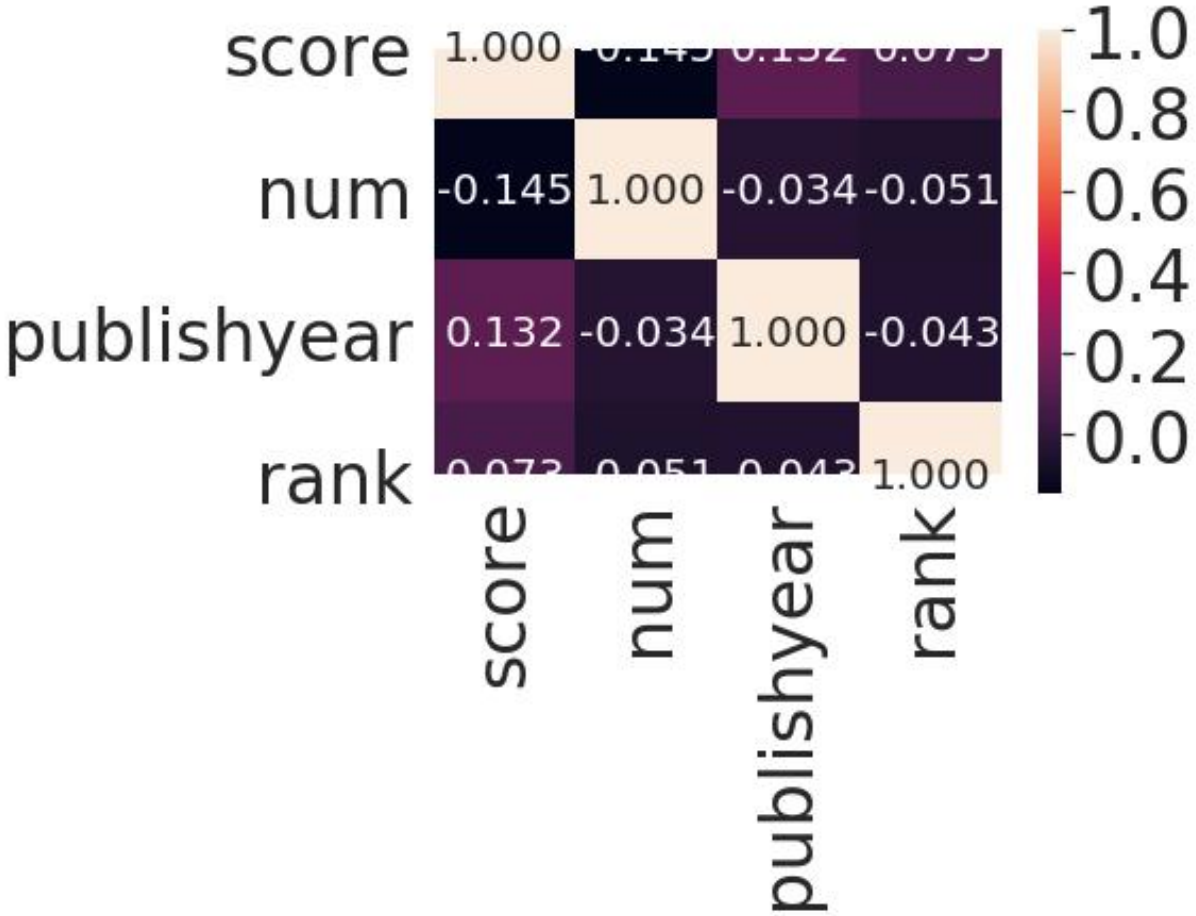
Linear regression

Draw the scatter plot of diffierent variables.



Pearson correlation coefficient

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$



We can find the correlation coefficient between all the variables is less than 0.2,which means that it's suitable for us to make linear regression for our data.

Maybe we can use the neural network fit the data.

Second goal

- 1.Brifiely analyze the factors which can reflect the popularity of book.
- 2.Use the PCA to reduce the data dimensions(from 4 to 2) to find the main factors that affect the book.
- 3.Use unsupervised learning methods to cluster the data.
- 4.Make labels for the data base on our goal.
- 5.Split the data into train set and test set.
- 6.Use supervised learning methods to classify the data.
- 7.Make predict on the test data.

Brifiely analyze the influence factors

```
In [33]: data.head()
```

Out[33]:

	book_name	score	num	picturelink	publisher	publishyear	ISBN	rank
0	呼兰河传	9.0	2069	https://img3.doubanio.com/view/subject/l/publi...	中国青年出版社	2003	9.787500e+12	206141
1	孽子	9.1	10902	https://img3.doubanio.com/view/subject/l/publi...	广西师范大学出版社	2010	9.787560e+12	205948
2	罗马人的故事2	9.2	2936	https://img1.doubanio.com/view/subject/l/publi...	中信出版社	2011	9.787510e+12	132734
3	木心作品八种	9.3	2121	https://img1.doubanio.com/view/subject/l/publi...	广西师范大学出版社	2009	9.787560e+12	118278
4	万物既聪慧又奇妙	9.2	3059	https://img3.doubanio.com/view/subject/l/publi...	中国城市出版社	2010	9.787510e+12	116627

We can see all of the three columns have some influence to popularity of douban.

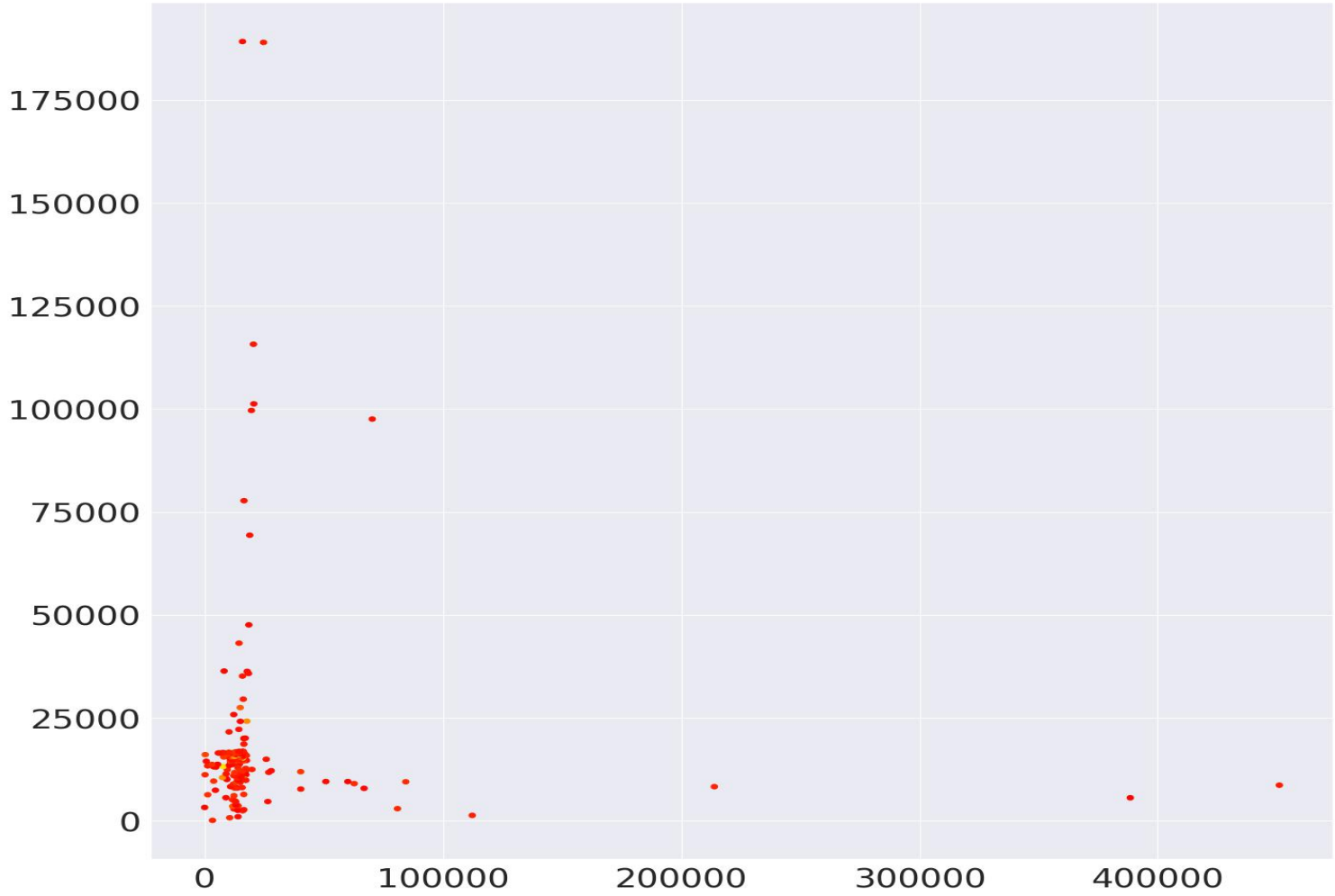
Use the PCA to reduce the data dimensions

```
cols = ['score','num', 'publishyear', 'rank']

pca=PCA(n_components=3)

X=data[cols]

NX=pca.fit_transform(X)
```

We can find some negative value in the pca transform matrix.I turn it to postive value.

```
NX=abs(NX)
```

```
plt.figure(figsize=(20, 20), dpi=80)
plt.scatter(NX[:, 0], NX[:, 1],c=NX[:, 2],cmap=plt.cm.autumn);
```

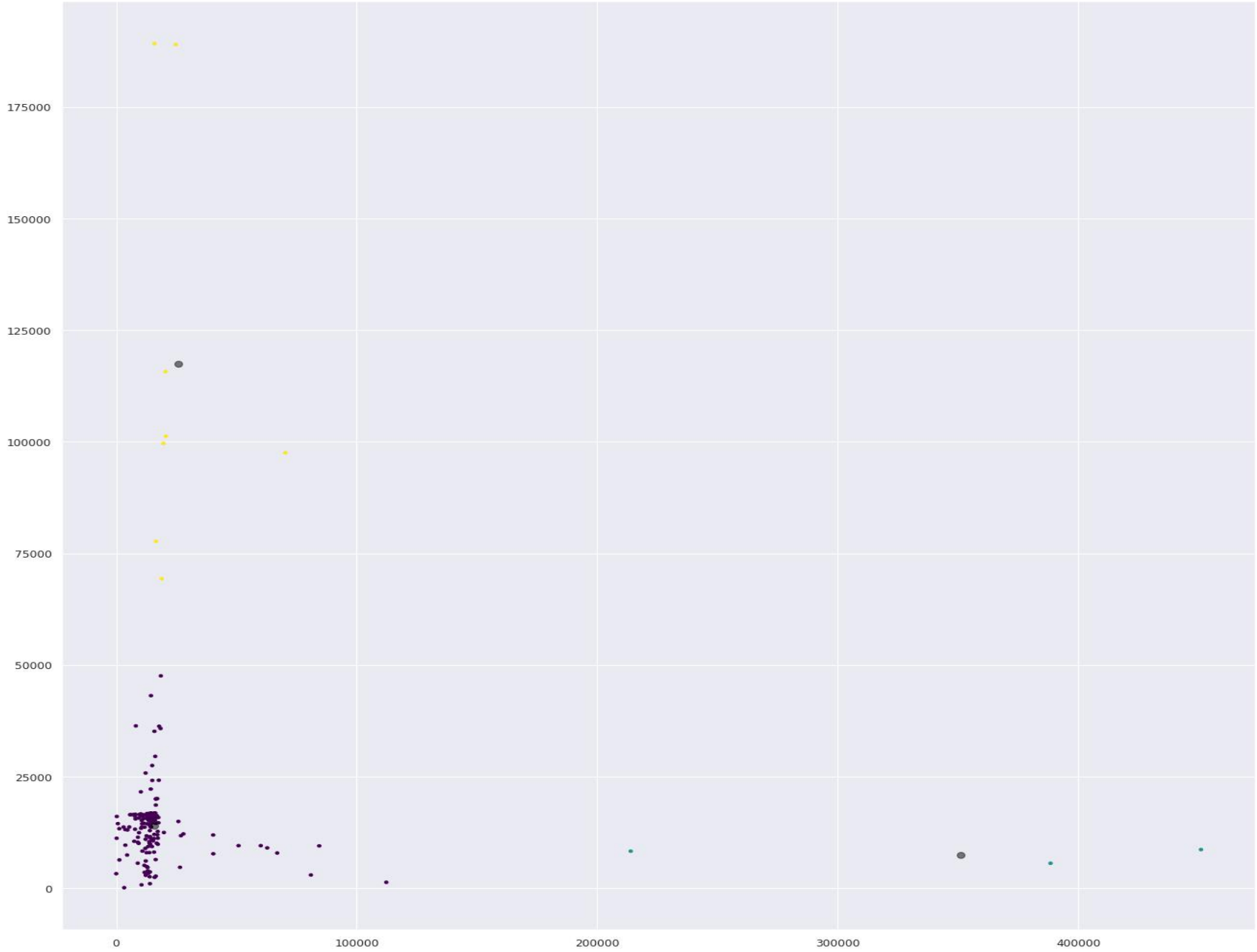


```
pca=PCA(n_components=2)
NX=pca.fit_transform(NX)
NX=abs(NX)
```

Finally we get two columns data,which is the singular value of the number of comments in douban and the rank in amazon.

Use unsupervised learning methods to cluster the data

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3)
kmeans.fit(NX)
y_kmeans = kmeans.predict(NX)
plt.figure(figsize=(20, 20), dpi=80)
plt.scatter(NX[:, 0], NX[:, 1], c=y_kmeans, s=10, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=50, alpha=0.5);
```

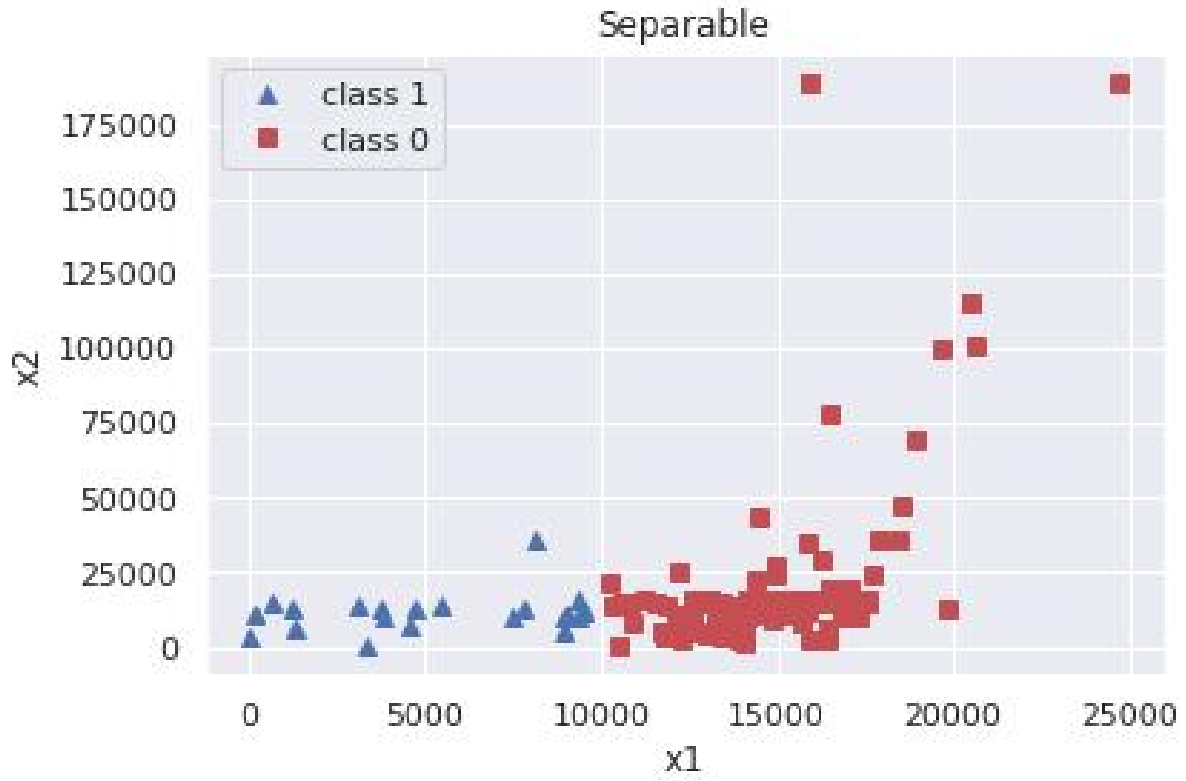

175000
150000
125000
100000
75000
50000
25000
0
0
100000
200000
300000
400000

We can see that we have 4 clusters of the data and each clusters have its center.

Then we focus on the cluster which include most number of data.

Make labels for the data base on our goal

```
mydict={'x1':rk, 'x2':cm}
df = pd.DataFrame(mydict)
df['target'] = 0

for i,_ in df.iterrows():
    if df.loc[i, 'x1'] <= 10000:
        df.loc[i, 'target'] = 1
```

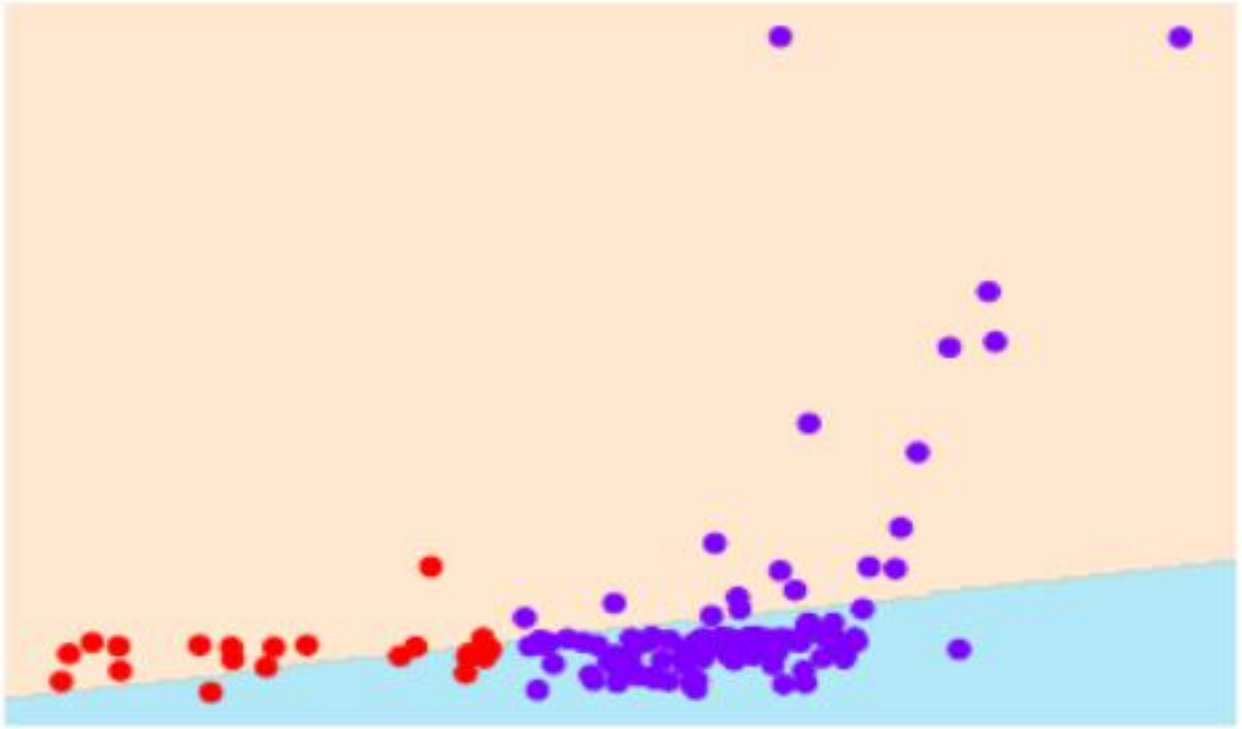


Split the data into train set and test set.

```
df2=df[0:130]
X_train = df2[['x1', 'x2']]
y_train = df2.target
df4=df[135:]
X_test = df4[['x1', 'x2']]
y_test = df4.target
```

Use Lsvc to classify the data

```
lsvc = LinearSVC(penalty='l2', loss='hinge', random_state=42,C=2)

lsvc.fit(X_train, y_train)

visualize_classifier(lsvc, X_train, y_train)
```



```
#print(y_test)
lsvc.predict(X_test)
print('score:
{}'.format(lsvc.score(X_test,
y_test)))
```

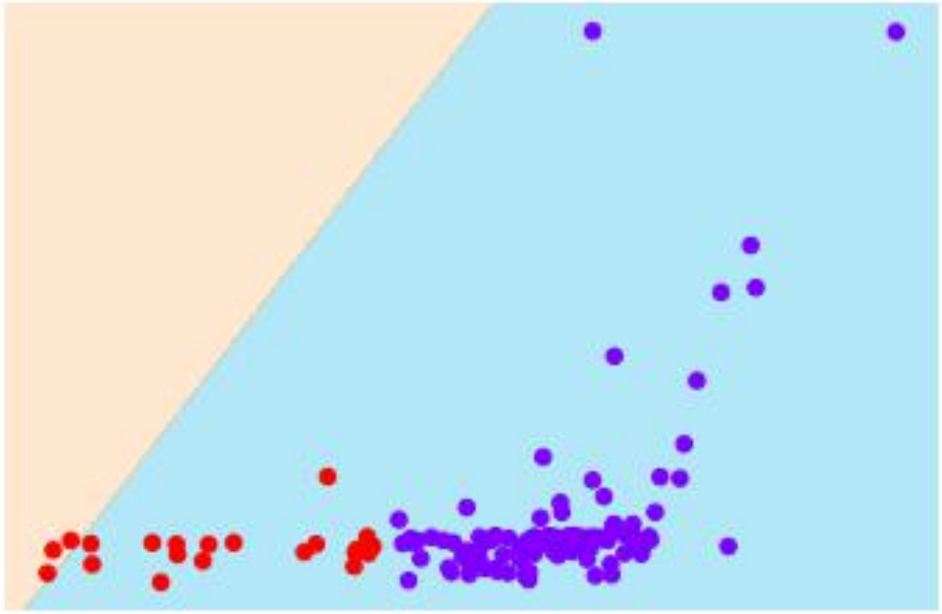
score: 0.875

Use logistic regression to classify the data

```
LR = linear_model.LogisticRegression(solver='lbfgs',
max_iter = 1000,multi_class='auto')

LR.fit(X_train, y_train)

LR.predict(X_test)

#print(y_test)

print('score: {}'.format(LR.score(X_test, y_test)))
```

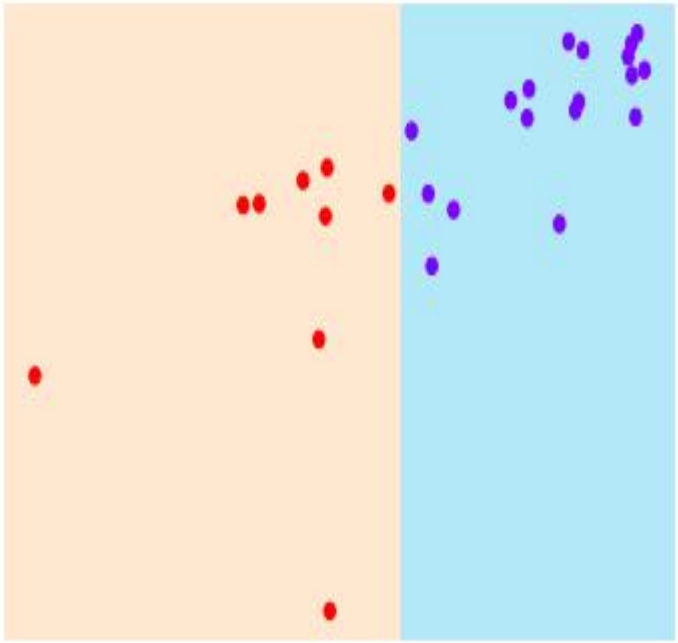
score: 0.75

```
visualize_classifier(lsvc, X_train,
y_train)
```



Use decision tree to classify the data

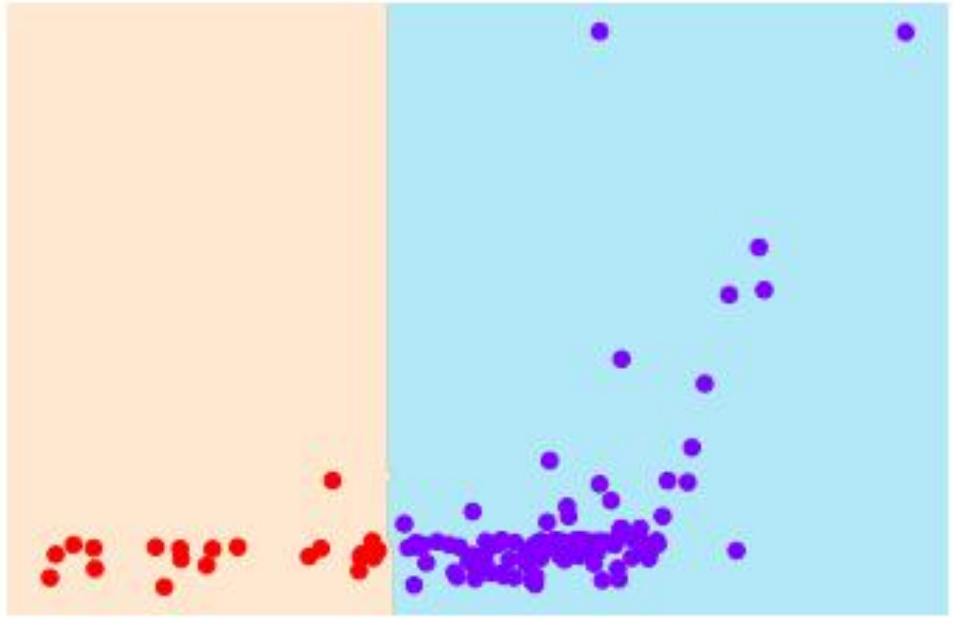
```
from sklearn.tree import DecisionTreeClassifier
# Instantiate the classifier class
tree_clf = DecisionTreeClassifier(max_depth=3, random_state=42)
# Grow a Decision Tree
tree_clf.fit(X_test, y_test)
from sklearn.tree import export_graphviz
export_graphviz(tree_clf, out_file='tree.dot',
                feature_names=['x1', 'x2'],
                class_names=['Yellow', 'Blue', 'Red'],
                rounded=True, filled=True)
visualize_classifier(DecisionTreeClassifier(), X_test, y_test)
```



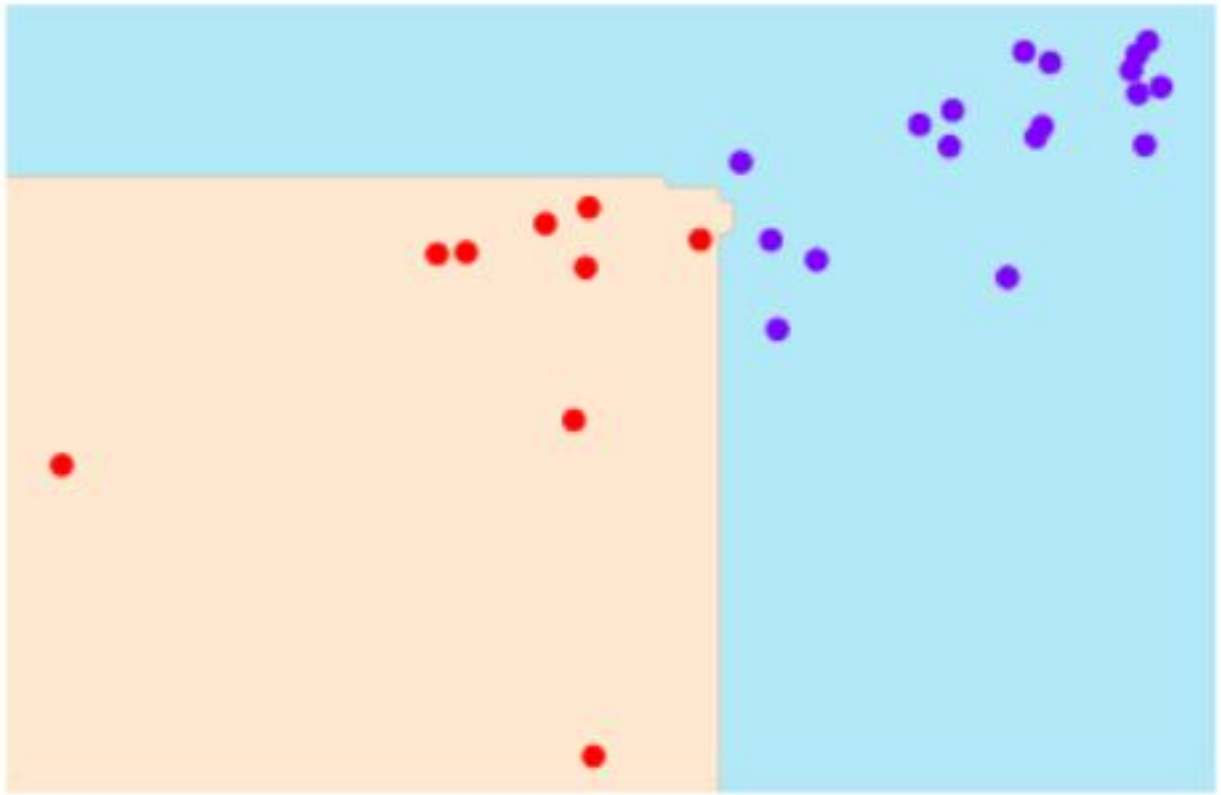
```
tree_clf.predict(X_test)

print('score: {}'.format(lsvc.score(X_test, y_test)))
```

```
score: 0.875
```

```
rf=RandomForestClassifier(n_estimators=500,criterion = 'entropy',
random_state = 0)

rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

print(metrics.accuracy_score(y_test, y_pred))

visualize_classifier(LR, X_train1, y_train1)
```



```
X_test1=np.array(X_test)
y_test1=np.array(y_test)
visualize_classifier(rf,
X_test1, y_test1)
```



Conclusion

- We can draw a conclusion that the random forest make greatest performance in the classification test.
- We can find the target book like this:

香初上舞·终上

金庸全集

雕刻时光

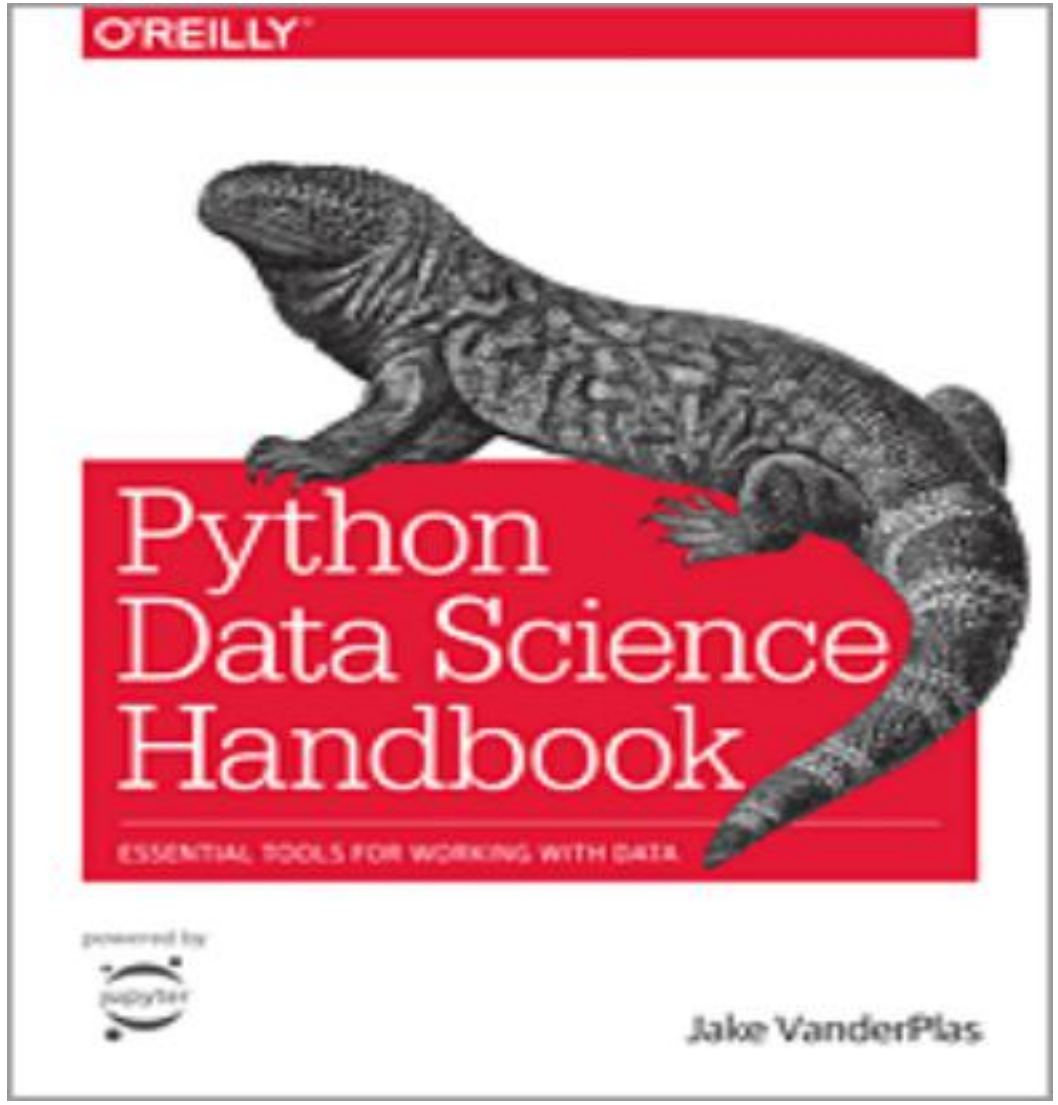
死屋手记

沙与沫

Further thinking

- The reason why douban's unpopular book become popular in Amazon, I think it's the cultural gap between China and America.

Reference


O'REILLY
Python
Data Science
Handbook
ESSENTIAL TOOLS FOR WORKING WITH DATA
Jake VanderPlas

Thanks!