人工智能与教育 --- MOOC 学习者数据分析

笔者选用 Python 语言分析数据，原因是其中的 Scikit-Learn、Matplotlib 等套件工具可以让做图更多变性，数据调试点也较为精细。

首先将原始数据载入 notebook，个人数据分析倾向使用集成平台 Anaconda 中的 Jupyter-notebook，此网页式的 cell 能从运行诚实到结果都一页一视窗呈现，无论是可视化还是一般的数据结构。如果有需要，亦可以直接在网页上书写解释文本。

	session_user_id	locale	timezone	access_group_id	registration_time	last_access_time	email_announcement
1	001029e4c5ab324ebcabb9e46f5f4f9b8701d6b9	en_US	America/Los_Angeles	4	1381828688	1382865234	1
2	00146603111f71eb6fa5bd4cfc3fa83aa37dd3f0	en_US	America/Los_Angeles	4	1385541132	1385541186	1
3	00236ce2c9977a076dfa6f8e8d51d6330b27bf4b	en_US	America/Los_Angeles	4	1387181822	1387181997	1
4	002bf11b1a03dc7d236b5cd7c836ddd4ead53c35	en_US	America/Los_Angeles	4	1388458001	1388458047	1
5	00354a3e7c017fe329b3fe7c00ec7b4388666846	en_US	America/Los_Angeles	4	1381246622	1382237531	1
6	0036e2d13b8ca9b9383252bfe0ce28ac540ee131	en_US	America/New_York	4	1384745174	1389589985	1
7	003cecd156c1204774e551d40aea7d62064e121f	en_US	America/Los_Angeles	4	1382772409	1386557729	1
8	004150b54bc9c10ac2fda7f9c2c543e511d07aaa	en_US	America/Los_Angeles	4	1382366722	1383656114	1
9	00487951baf2a403c063d49f4b14756e49865225	en_US	America/Los_Angeles	4	1384700591	1384700599	1

相关性分析

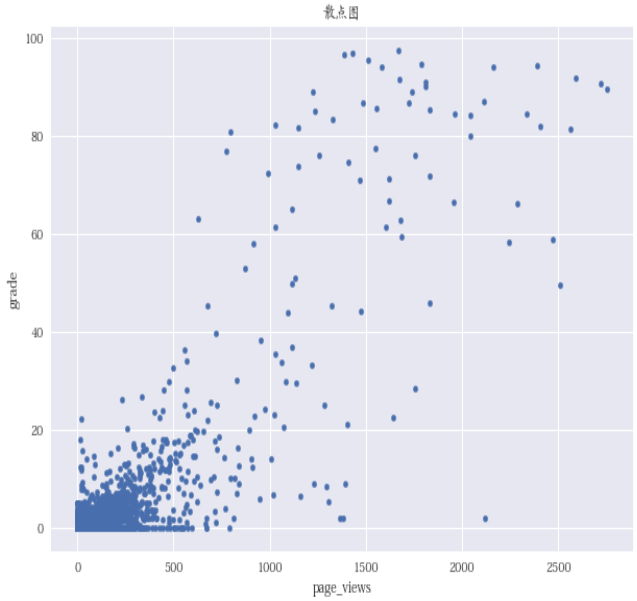
为了便于选择需要更深入分析的数据变项（学习者特征），相关系数分析得初步将数据展现他的重要性，但需要事前先清晰数据，清理过的数据如下：

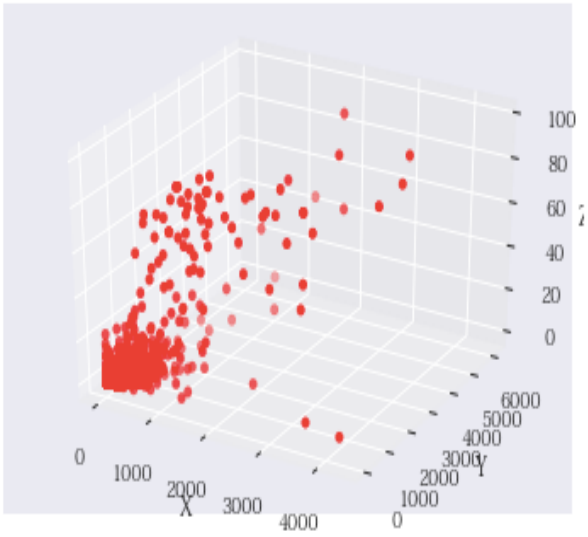
1. 学习者的基本资料，有如 ID、IP 等无助信息删除
2. 论坛参与行为的数值过多缺失，不具备高参考性
3. 剩余主要学习行为的缺失值以零代替（后续的建模不可有缺失值）

	duration	begin_time	video_views	page_views	assignment_score_sum	lecture_item_views	quiz_score_sum	grade
1	1036546	3431888	0.0	16.0	0.0	0.0	0.00	0.000000
2	54	7144332	0.0	16.0	0.0	5.0	0.00	0.000000
3	175	8785022	28.0	16.0	0.0	2.0	2.00	0.000000
4	46	10061201	4.0	12.0	0.0	1.0	0.50	0.000000
5	990909	2849822	0.0	56.0	0.0	6.0	3.25	0.919811
6	4844811	6348374	0.0	0.0	0.0	0.0	0.00	0.000000
7	3785320	4375609	68.0	170.0	10.0	11.0	11.25	3.239850
8	1289392	3969922	14.0	64.0	0.0	21.0	0.00	0.000000
9	8	6303791	2.0	10.0	0.0	1.0	0.00	0.000000

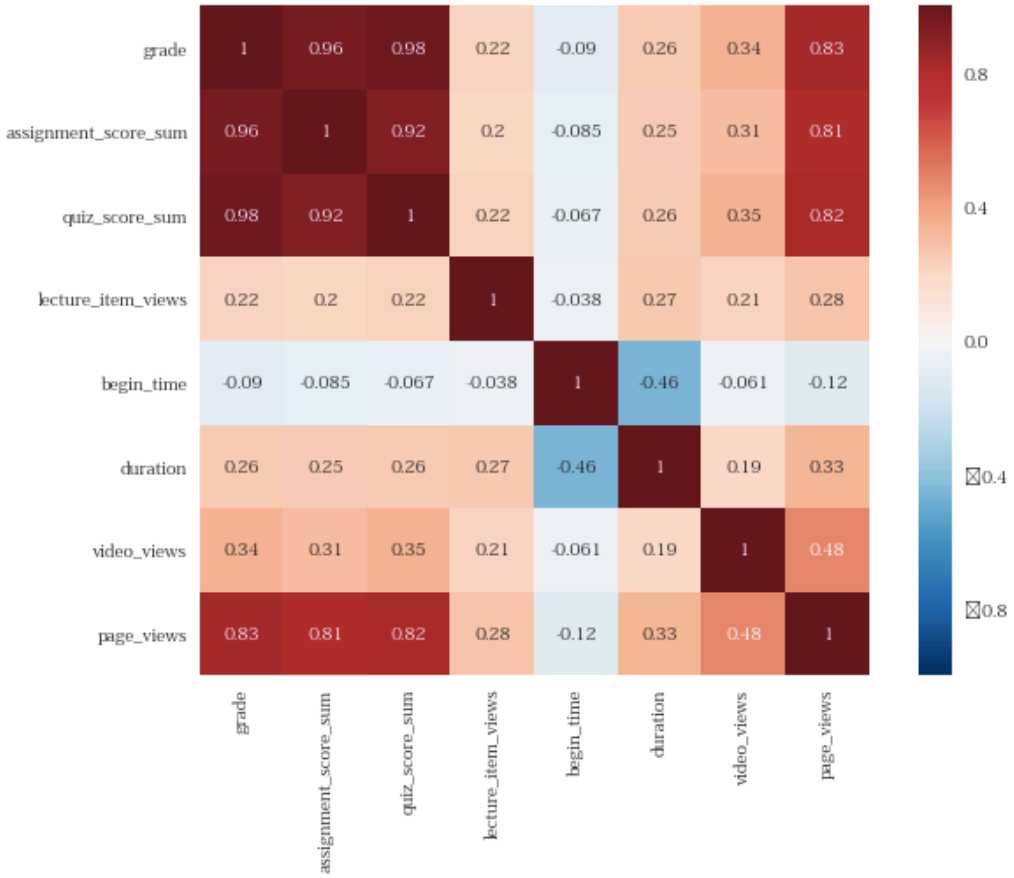
	grade	assignment_score_sum	quiz_score_sum	lecture_item_views	begin_time	duration	video_views	page_views
grade	1.000000	0.957034	0.982349	0.217803	-0.090318	0.257004	0.338275	0.832502
assignment_score_sum	0.957034	1.000000	0.922980	0.203592	-0.084779	0.247117	0.311573	0.809662
quiz_score_sum	0.982349	0.922980	1.000000	0.219991	-0.066822	0.257139	0.351436	0.822830
lecture_item_views	0.217803	0.203592	0.219991	1.000000	-0.037827	0.265412	0.211049	0.278836
begin_time	-0.090318	-0.084779	-0.066822	-0.037827	1.000000	-0.459531	-0.061103	-0.117111
duration	0.257004	0.247117	0.257139	0.265412	-0.459531	1.000000	0.189003	0.332942
video_views	0.338275	0.311573	0.351436	0.211049	-0.061103	0.189003	1.000000	0.481450
page_views	0.832502	0.809662	0.822830	0.278836	-0.117111	0.332942	0.481450	1.000000

上表则是呈现了学习行为的相关系数，为了更容易观看数值的关系，通常会把数据进一步做图，像是常见的散点图，有鉴于散点图笔者这里以其中几项来做图，但是散点图碍于视觉化的三维限制，无法过多的展现所有特征之间的关系。





左上图即是选择学习者特征中的 grade、page_views 进行做图，右上图则是多加上 vedio_views 做图，3 维图若非特别完美的表现的话，并不容易看清楚，因此笔者更建议以下列的热层次图来表现特征之间的相关性回来的更直观。



回归分析

在题目要求的分类算法中，本文选用回归分析，因为使用回归的结果已经算是漂亮的，让我们利用相关性分析前所清洗的数据。至于数十种的回归分析方式中，为了确认最佳的学习曲线与整体拟合程度，笔者尝试过了一般线性回归、随机梯度下降回归、正规化后的多元非线性回归等，尝试后以正规化回归中的弹性网（Elastic Net）最佳，结果步骤如下：

第一步

笔者将数据切分成训练集与测试集，上图是切分后的学习者特征，切分比例为 7:3。

```
x = df[['page_views','video_views','lecture_item_views','assignment_score_sum',\
        'quiz_score_sum','duration','begin_time']]
y = df[['grade']]
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,test_size = 0.3)
x_train.head()
```

	page_views	video_views	lecture_item_views	assignment_score_sum	quiz_score_sum	duration	begin_time
7609	92.0	132.0	27.0	0.0	0.4	1519	11820161
4789	76.0	14.0	1.0	0.0	0.0	4935804	4373823
10011	10.0	2.0	1.0	0.0	0.0	9	3968071
3595	22.0	26.0	2.0	0.0	1.5	1476	10535033
3107	32.0	26.0	3.0	0.0	4.0	372617	3536865

第二步

```
In [110]: from sklearn.preprocessing import StandardScaler
          sc = StandardScaler()
          sc.fit(x_train)
          x_train_std = sc.transform(x_train)
          x_test_std = sc.transform(x_test)
          x_std = sc.transform(x)
```

```
In [176]: from sklearn.preprocessing import PolynomialFeatures
          quadratic = PolynomialFeatures(degree=1)

          x_train_poly = quadratic.fit_transform(x_train_std)
          x_test_poly = quadratic.fit_transform(x_test_std)
          x_poly = quadratic.fit_transform(x_std)
```

切分后的数据在建模前标准化数据范围，避免像是学习者特征中的时间与成绩的变项之间数值差异过大，同时建立高次方项的对象（quadratic），以便于调试特征的权重值。

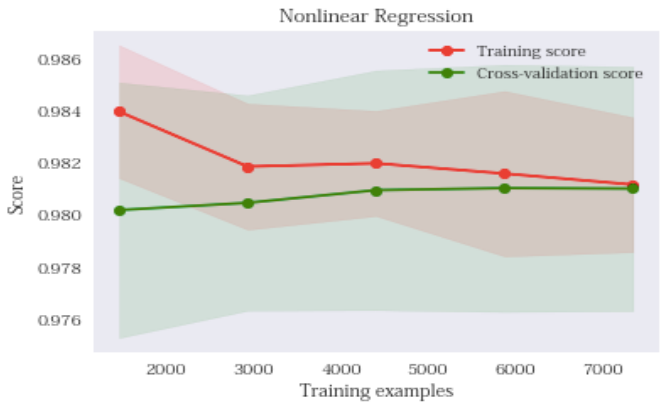
本文在数据建模的验证法所选的是 K-Fold 交叉验证，能从新从原始数据集中均等的抽取样本测试，比起单单从切分数据时切分出一个测试集来得更能计算出平均的效能，而为了能拿原始数据验证，这里需要再另外将无切分的原始数据一同标准化。

第三步

针对选择回归法的过程，笔者是先采取简单线性回归来获得假性建模成果，此目的是为了确认是否需要使用其他类型回归，或者选择其他更适合的分类型建模。建模分析过程的程式段落太长，这里直接以数据显示如下：

1. 一般线性回归

```
各變項參數: [[ 0.          0.13382489 -0.06900108  0.01401138  2.16172377  3.82498988
  -0.05610694 -0.13400765]]
MSE: 0.95
R Square: 0.978393401379
```

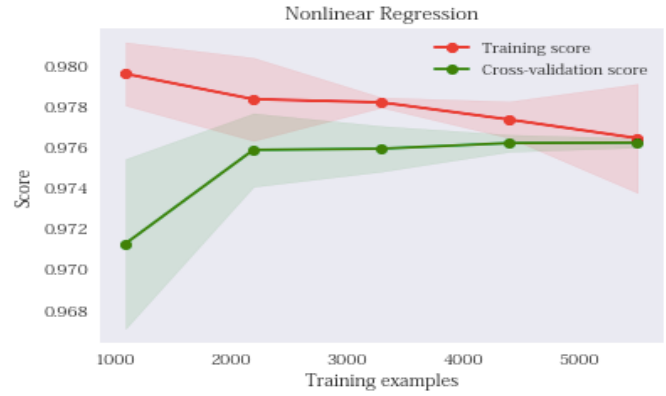


上列的数据与图标分别是依照切分前的数据顺序排列的变项参数，从均方误差(MSE)与正确系数(R-Square)来看，是一个极高准确率的回归建模，学习曲线图的训练线路偏差也小极（红线终点离 1 的距离近），学习曲线的变异程度接近零（红绿两线的终点靠的近）。

但是若我们要解释变项与模型的关系，正常而言除了「begin_time」本来就是与「grade」呈现负相关的特征值之外，其他的特征值应该是呈现正的参数，但是在一般线性回归的结果中并非如此，可判定可能有变项相关性过大，导致建模时计算偏重了某些项目。

2. 弹性网正规回归（Elastic Net）

```
各變項參數: [ 0.          0.54448122 -0.          0.          2.22931661  3.10800133
  0.         -0.0282207 ]
MSE: 1.23
R Square: 0.9721498806
```



在一般线性回归的测试建模中可以观测到，用回归的方式分析本文所筛选的学习者特征是适合的，剩下的就是为了避免权某些变项的重值过大，算法上可以在 Cost Function 中增加惩罚项来抑制，调整 alpha 的系数即可。

上列图标中，虽然均方误差与正确系数比起原先降低些许，学习曲线的分数也降低 0.05 左右，但依然属于低偏差，并且几乎无变异区间值，因此并没有牵拟合或是过拟合的状况。回到本来有问题的变项参数，各参数回复到了正常值，「video_views」虽然归零，但是跟其他参数相比后，能确定这是此变项较为正确的位置，也就是说此模型是可以使用的。

聚类分析 K - Means

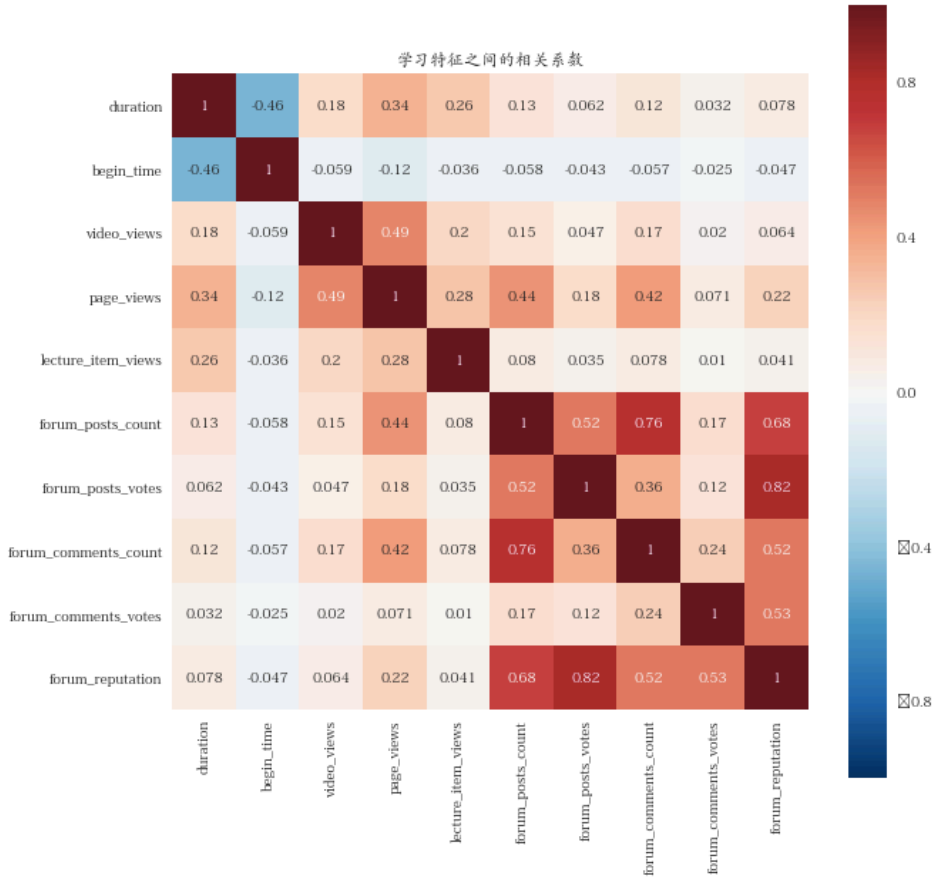
在聚类分析中，笔者重新整理了数据，将成绩的部分删除，保留了论坛参与的部分，虽然此部分缺失值多，但因为本文期望能尝试在学习行为中找到学习者的主要区别加以分群，让分析结果成为能够调整课程的依据，因此纳入论坛参与。

第一步

首先将挑选的数据进行标准化确保一定尺度。

```
x = df.values
from sklearn.preprocessing import StandardScaler
x_std = StandardScaler().fit_transform(x)
x_std
```

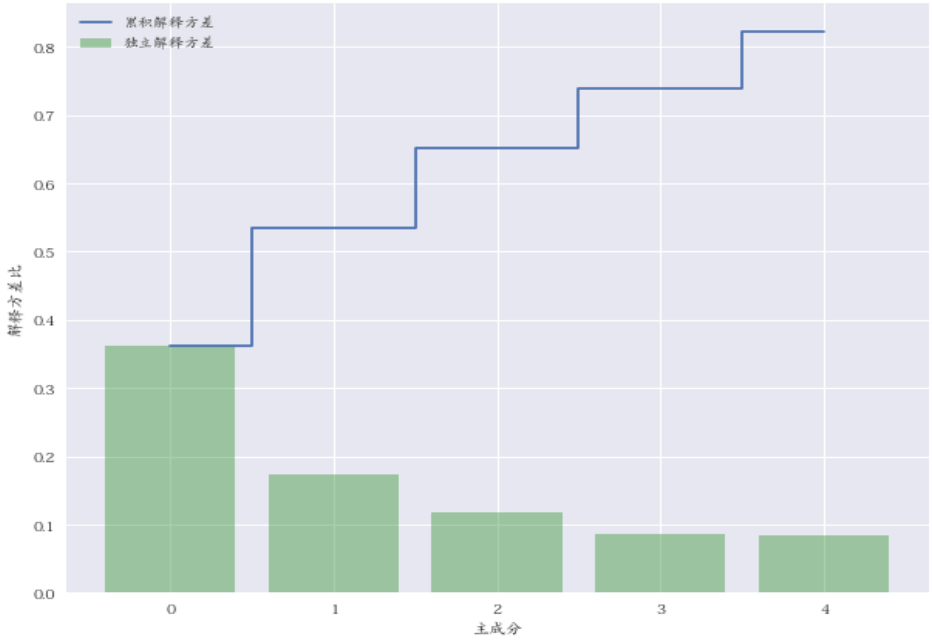
再者一样将数据相关性分析后制作成热分层图。



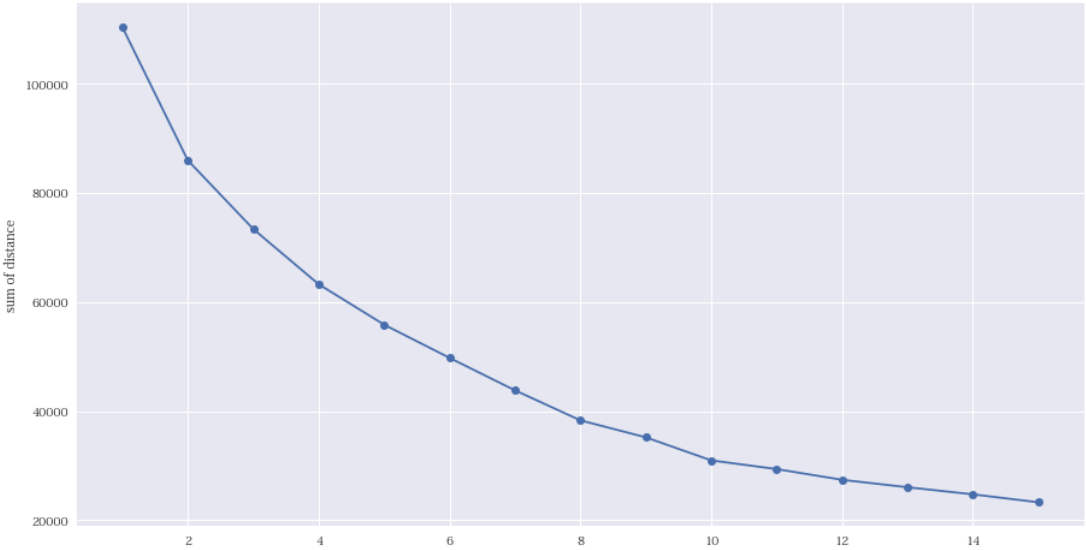
此处的相关性分析主要是了解新的数据内彼此的关系，以便于用数据值增加对聚类后的群体的解释性，尽量减少对新分群特征值的主观解释。

第二步

接下来的步骤本来为了使用主成分分析(PCA)的方式来事先处理特征值，主要是减少不同让新的主成分之间相关性降低，减少建模后过度拟合的状况发生，但是会因为同时进行了降维，导致不容易定义聚类后的学习者群体，因此这里仅放上一张尝试 PCA 后所建构的累积长条图来展示，若能有更好的领域知识理论框架能解释数据的结果，或许有经过 PCA 预处理的聚类结果会更精准。

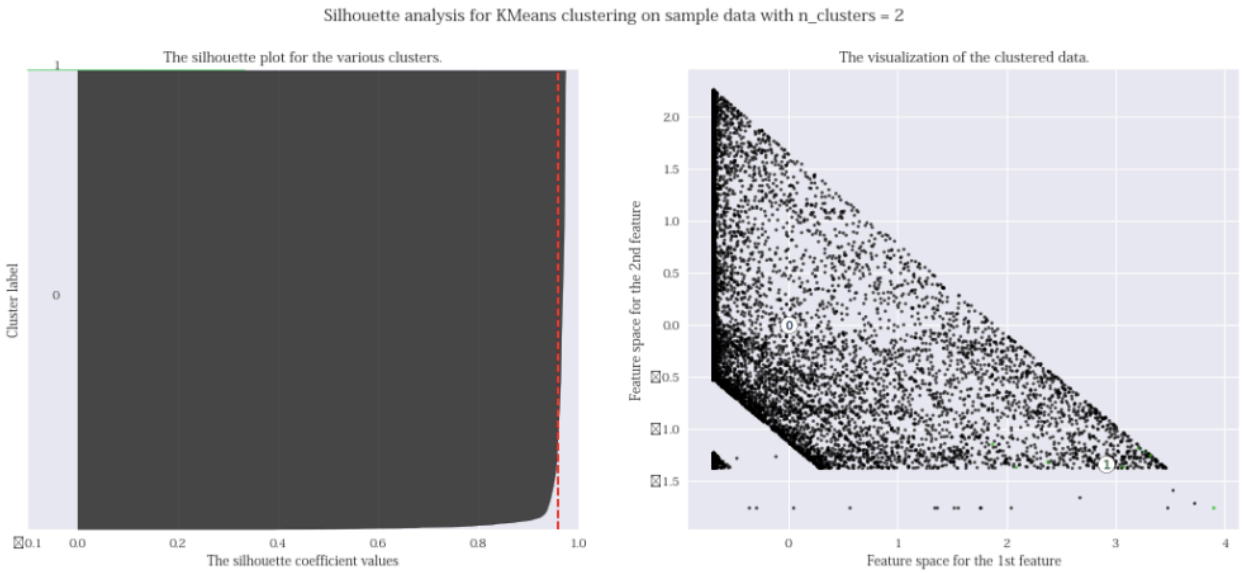


跳过 PCA，本文直接选用标准化后的数据放入 K-means 聚类法进行分析，分析之前计算不同 K 值的各集群总距离和，可以了解设定几个 K 值分群能最快速的将群体分开，也更准确，而本次的数据的**最长距离分群点**是在第二 K 值。



```
kmeans = KMeans(n_clusters=2, init = 'k-means++')
x_clust = kmeans.fit_predict(x_std)

plt.figure(figsize = (7,7))
plt.scatter(x_std[:,0],x_std[:,1], c= x_clust ,alpha=0.5)
plt.show()
```



分析过后的结果并不好看，在 kmeans 的做图中使用肉眼来辨识有一定难度，主要是大多的学习者都呈现了相同的学习表现，而起初笔者将成绩的特征值拿开，亦是减少成绩对学习者分群的影响，但是从纯粹的分析来看，仅是将 K 值设定为 2 的分群结果是非常不平均的，我们这里先查看聚类结果的轮廓系数：

```
from sklearn.metrics import silhouette_samples, silhouette_score
silhouette_avg = silhouette_score(x_std, kmeans.labels_)
print('平均輪廓係數:',silhouette_avg)
```

```
平均輪廓係數: 0.960694312611
```

分析过后的轮廓系数是非常接近于一，代表组间的距离远大与组内的距离，这系数表现下的聚类成果应该是准确的，因此我们可以采纳这个聚类的结果。

至于只有两群的聚类下，要如何定义呢？这是后续可以继续研究的范畴，譬如找到适合的概念框架结合，本文在此只表示出聚类成果。

总结

在回归分析中，也是可以将论坛参与加入模型训练的，因为从相关性分析可以看到的是，学习者的论坛参与其实与分数的关系更高，但是在回归分析中如果转换成零则可能会产生太多的零，进而影响建模准确性，所以最后不采纳。

根据原有的数据，其中数据属性分为文本信息、时间信息、成绩信息，像是针对时间信息，若是以时间序列法分析，可能也会得到意外的收获，能找的关于学习者整个课程的需求，以及课程环节对学习者的不同重要程度。