Jingfei Chen, Yutong Shen, Yiran Wang, Wendy Weng

Creative EDM Assignment: Course Selection

HUDK 4050

2021.12.13

Problem Statement

In college, many of us have experienced the situation where the academic advisor threw a list of required and elective courses at us and made us choose on our own. The online registration system only provides brief introductions and inadequate knowledge about the courses. Some students may get confused about the courses, and thus have a hard time making decisions. Many of us might have spent a lot of time reaching out to previous students for advice on course selection and going to “rate my professor” to check the rating of the courses and professors. How can the schools take actions to help their students with course selection? We believe that by using social network analysis and clustering, we will be able to provide the students with more information that could help them with class selection.

Literature Review

Course selection is an interesting educational topic that deserves deep research. With the rapid development of the network, we think there exists some algorithms that can reflect some relationships between students’ course selection and their social network. Past studies raised some meaningful approaches on discovering the course selection mechanism in higher education. Baker (2018) talked about the complex choices that students faced when they were deciding majors, especially meta majors. Using the clustering and social network analysis, the study found

that “meta majors rely on clustering” (Baker, 2018). Moreover, students who were majoring in A usually considered B as their second majors, thus constructing the groups of networks (Baker, 2018). It is inferential that our study may reflect the same phenomenon that students with same majors will make similar choices on course selection.

In addition to social network analysis, prior studies also mentioned intelligent recommender systems for course selection in today’s educational administration (Su, Tang, & Zhang, 2021; Tang, Chi, & Tang, 2020), which coincides with our research topic. Our aim is to offer effective recommendations for students who are disoriented during course registration: not only on mandatory major courses, but also on elective courses. Su, Tang, and Zhang (2021) concluded that the recommendation model can quickly help students target the courses and compare similarities using big data and cloud computing, thus helping them to avoid random selection mode. Tang, Chi, and Tang (2020) further discussed multiple approaches on course selection algorithms based on different learning outcomes and students’ expectations.

Our study will focus on investigating how to give suggestions on course selection for students with the same majors. During our analysis, we seek to use social network analysis and clustering on senior students to figure out if students with the same majors share a similar course selection and reveal some clusters.

Data Source

As mentioned in the Problem Statement, we are hoping to provide the students with more information that could help them with class selection. We will use social network analysis to create lists of classes that are often chosen together and use clustering to give the students some ideas about what the classes are like.

To provide students with relevant pieces of advice, we need access to the school registration system and the course evaluation database.

From the **registration system**, we will be able to collect the following data of a specific student: courses taken(session, instructor), the requirement for graduation, elective course choices, GPA(specialization and cumulative), final course grade, major, degree, and graduation year (expected or actual).

From the **course evaluation database**, the following data will be helpful for the analysis: course type (required, optional, or voluntarily), student self-reflections, course design, organization, syllabus, instruction review, student learning and satisfaction, assessments, average number of hours put into class per week, and demographics.

Analysis Plan

Social Network Analysis

We will first use the courses taken (course number & instructor) data to conduct a social network analysis. We will take the information from graduates and see what courses they have taken while in college/the program. The analysis will be based on data from each major/program. We will only choose the students who have declared a single major/program in the database and ignore the required courses that everyone needs to take. Thus, the network we choose will be the elective courses that are commonly chosen together by the students in major A at school A. The vertex will be courses (course number & instructor) and the edge will be students taking the courses together. The graph we got will be an undirected network. The number of vertices will tell us the number of courses students have taken. We would be expected to have a high diameter for that means the students have more options to choose from. Degree Centrality gives us an idea

of how 'important' each node is in the network. This would tell us the most popular courses chosen by the students in the major.

What would be most helpful to the students will be the communities we detected. Previous students might have spent a lot of time on course selection, so the communities might reflect their common interests. If the current students have taken several classes or plan to register for some courses, he or she may look at the community that shares these classes and then explore more options in this community. We are suggesting that the courses in the same community will have some similarities either in course content, delivery method, professor rating, or course difficulty.

However, only having the communities of courses commonly chosen together might not be enough for the students to make decisions. The previous students might take the courses but not value some of them. The network didn't include detailed information but only provided the students' overview. Therefore, we are hoping to do further analysis to give more detailed information.

Course Clustering

We will use the data from course evaluations and create a course ranking/clustering for the students. The first step is to create a dataset with courses' rating scores and professors' ratings from the course evaluation database. To create the dataset, we will calculate the average scores for the course and the corresponding professor according to students' responses to the course evaluation form (see Appendix A). We will assign the scores of 2, 1, 0, -1, -2 respectively to responses of "Strongly Agree," "Agree," "Neutral," "Disagree," "Strongly Disagree." For instance, for the question: "The professor shows interest and enthusiasm for teaching the

course," a response of "Strongly Agree" will result in a score of 2, and "Strongly Disagree" will result in a score of -2. We will calculate the average score for each question inside the course rating section and the professor rating section. Then we will find the average scores for the two sections. Therefore, the resulting dataset will have a course rating score and a professor rating score for each course (the rows will be courses, and the columns will be course rating score and professor rating score).

We will adopt the same approach in preparing the data from other kinds of courses evaluation (see Appendix B). However, we need to be more careful when dealing with this type of course evaluation. For instance, in the "demographic" section, we will only take the students who have master degrees and enrolled at Teachers College. The responses to questions other than the demographics are required, thus we should not have missing data. However, in case of missing data, we will replace it with the average.

We will use Scikit-Learn's PCA estimator for dimension reduction and then use Kmeans to get the different clusters based on the course evaluation results. We have attached the code using some random sample data (thus the result might not be representative) in the Appendix C.

Discussion

Social Network Analysis

In the social network analysis, we expect to identify communities composed of different courses that could give students a broad view on what other courses he or she might be interested in. Since the data are based on students with the same major, the communities might be on the basis of their orientations and personal interests. The good side about social network analysis is that it doesn't require lots of data to support the analysis; all we need are the courses taken by the

graduates. The limitation is also obvious because it is not supported with detailed information and can only serve as a general guidance. Thus we decide to explore further with course evaluation and clustering.

Clustering

In clustering, we need more data from both the course evaluation and the registration system. We will get a better idea of how the course is structured: whether the course is engaging or not, is the professor available outside of the classroom, are the courses well-organized, if you can bring what you learn and apply in real life… We are adopting clustering so that different schools can have their own versions of course evaluation. For instance, Appendix A put courses evaluation into mainly two categories: ratings for professors and ratings for courses; while Appendix B have more detailed classifications including student self-reflections, course design, organization, syllabus, instruction review, student learning and satisfaction, assessments, facilities and technology support, courses requirement, and average number of hours put into this class per week. We are hoping that after analysis, we will be able to put the courses into clusters.

In the **Appendix A** situation, the clusters might result in four categories which are: *a) courses high instructors high, b) courses high instructors low, c) courses low instructors high, d) courses low instructors low*. Students getting the list might look at the commonly taken together courses (social network analysis results) and try hard to enroll in the ones listed on the *a) courses high instructors high* category. Their next options might be category *b)* or *c)* depending on whether they want to learn the course material or want to find an easy-pass course. We expect that they will try to avoid the courses listed in the *d)* category, which have low scores both for the course and the instructor.

In **Appendix B**, the clusters will provide the students with an idea of how the course will be framed in a more specific way. Again, the students will be able to find the cluster that they fitted in most. The most desirable cluster might include the following aspects: encouraging self-exploration, having clear course structure, teaching in an engaging way, and receiving high satisfaction. By contrast, the most undesirable one might state the opposite in the above sides.

Based on the communities generated by the social network analysis and the clusters formed by more detailed information, we are hoping this model could better serve our students with course selection.

Implication & Future Study

However, the implication of the analysis should not stop here, the school should also pay close attention to the clusters that are considered the most undesirable to see what's happening in those courses and make changes for enhancement. Additionally, we suggest that the school should look at the courses with lowest degree centrality (courses that students with the same major/program are less interested in) in social network analysis to see if there ways to encourage the students to participate more.

The future study could include more variables such as student's personality traits, personal interests, or career goals in the clustering stage to see if we could get more "personalized" clusters. Social network analysis can be implemented at different scope (different department / major or program / concentration) to see if the communities have some overlap that could facilitate further research.

Reference

Baker, R. (2018). Understanding college students' major choices using social network analysis. *Research in higher education*, *59*(2), 198-225.

Su, F., Tang, J., & Zhang, Z. (2021). Research on College Students' Course Selection Recommendation Model Based on Big Data and Cloud Computing. *Journal of Physics: Conference Series* (Vol. 1982, No. 1, p. 012203). IOP Publishing.

Tang, C., Chi, X., & Tang, J. (2020). Intelligent Selection Recommendation Algorithm in the "Double" Mechanism. *IOP Conference Series: Materials Science and Engineering* (Vol. 750, No. 1, p. 012090). IOP Publishing.

Appendices

Appendix A - Courses Evaluation Form Pepperdine University

Professor Rating

The professor shows interest and enthusiasm for teaching the course.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The professor is available outside of class for consultation if needed.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The professor is prepared for class and makes good use of class time.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The professor presents course material in a clear and engaging manner.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The professor is an excellent teacher.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Course Rating

The course is well-organized.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The course textbook and other reading assignments are appropriate in content.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The course tests and other evaluations are appropriate in content and difficulty.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The course assignments are reasonable and appropriate in content and difficulty.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The course is demanding in comparison to other courses.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The course has increased my knowledge or understanding of the subject.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The course is excellent.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The overall class experience has enhanced my ability to think clearly, logically, independently, and critically.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

The overall class experience has contributed to the development of my sense of personal values and moral integrity.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Appendix B - Courses Evaluation Form Teachers College

Student Self-Reflection

1) When I didn't understand something, I asked questions in class

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

2) I completed assignments on time

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

3) I completed assignments thoughtfully and did my best work

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

4) I made an effort to communicate with the course instructor during office hours

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

5) I shared my opinions, answered questions, and generally participated in class sessions

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

6) I made a conscious effort to link the class content to my own interests

- Strongly Agree
- Agree
- Neutral
- Disagree

- Strongly Disagree

Course Design, Organization, and Syllabus

7) Course objectives were clearly stated and aligned with course content

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

8) Course requirements were clearly defined

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

9) Course materials included multiple viewpoints and perspectives

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

10) Class size was appropriate for this course

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Instruction

11) Class sessions were well organized

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

12) Subject matter was presented effectively

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

13) Reading assignments contribute to my understanding of the subject

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

14) Instructor is responsive to students' questions and/or comments

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

15) Instructor treats all students with respect

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

16) Instructor is accessible to students outside of class

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

17) Instructor adhered to, and was consistent with, class meeting times

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

18) The instructor used technology effectively to promote student learning (online or in-person)

- Strongly Agree
- Agree

- Neutral
- Disagree
- Strongly Disagree

19) The activities in this course encouraged student engagement and participation (online or in-person)

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

20) I was able to access appropriate learning tools and resources for this course (online or in-person)

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Student Learning and Satisfaction

21) Course assignments were valuable learning experiences

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

22) I would recommend the course to other students

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

23) I would recommend this instructor to other students

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

24) I learned a lot in this course

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Assessments/Evaluations

25) Evaluations reflected course objectives

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

26) Evaluation/grading criteria were clearly defined

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

27) Sufficient number of opportunities to evaluate my learning

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

28) Instructor provides helpful feedback on assignments

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Facilities and Technology Support

29) Classroom space / facilities were adequate for the needs of the class

- Strongly Agree

- Agree
- Neutral
- Disagree
- Strongly Disagree

30) Equipment and technology support by the College were adequate

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

Demographics

Items in this section are optional: feel free to not respond to any items below if you are concerned with issues of anonymity

31) My degree program is:

- Non-degree
- Master's
- Doctoral

32) I am enrolled at:

- Teachers College
- Other school of Columbia University
- Other college/university

33) This course is:

- required for my degree
- one option in a list of courses that satisfy my requirements
- selected and taken voluntarily

34) The average number of hours I put into this class per week is:

- Less than 2 hours
- 2-4 hours
- 4-6 hours
- 6-8 hours
- More than 8 hours

Appendix C - Clustering Example Using Random Data

Course: HUDK 4050

Authors: Shirley Chen, Yutong Shen, Yiran Wang, Wendy Weng

Background: We will use the data from course evaluations and create a course ranking/clustering for the students. The first step is to create a dataset with courses' rating scores and professors' ratings from the course evaluation database. In the data preparation process, we will calculate the average scores for the course and the corresponding professor according to students' responses to the course evaluation form (see Appendix A). We will assign the scores of 2, 1, 0, -1, -2 respectively to responses of "Strongly Agree," "Agree," "Neutral," "Disagree," "Strongly Disagree." For instance, for the question: "The professor shows interest and enthusiasm for teaching the course," a response of "Strongly Agree" will result in a score of 2, and "Strongly Disagree" will result in a score of -2.

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [2]:
```
# Import dataset
df = pd.read_csv("Sample.csv")
df.head()
```

Out[2]:

	Course Number	Professor	Course	Difficulty
0	4000	-1.204699	-1.142584	0.606862
1	4001	1.521202	-0.186447	-1.153300
2	4002	-1.125400	-0.838220	0.202952
3	4003	-0.095452	-0.938903	-0.614851
4	4004	1.584738	1.208184	-1.331121

In [3]:
```
Selected = df[['Course Number','Professor','Course','Difficulty']]
Selected
```

Out[3]:

	Course Number	Professor	Course	Difficulty
0	4000	-1.204699	-1.142584	0.606862
1	4001	1.521202	-0.186447	-1.153300
2	4002	-1.125400	-0.838220	0.202952
3	4003	-0.095452	-0.938903	-0.614851
4	4004	1.584738	1.208184	-1.331121
...	...	...	...	...
96	4096	-1.238082	0.330791	-0.025611
97	4097	1.627054	0.342121	0.110961
98	4098	-1.205515	0.566869	-0.709801
99	4099	-1.621594	-0.256754	1.246266
100	4100	1.887829	1.307731	0.091984

101 rows × 4 columns

In [4]:
```
#Drop the Course Number for PCA
SelectedClean = Selected.drop(columns =['Course Number'])
```

In [5]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(SelectedClean)
scaled_data = scaler.transform(SelectedClean)
scaled_data
```

Out[5]:
```
array([[-1.04187886, -1.08447663,  0.71304267],
       [ 1.18316739, -0.21460281, -0.90638351],
       [-0.97715022, -0.80757219,  0.34142827],
       [-0.1364445 , -0.89917123, -0.41098657],
       [ 1.23502985,  1.05420419, -1.06998597],
       [-0.8945188 , -0.6312976 ,  0.26800056],
       [ 1.1287489 ,  0.13445635, -0.84850411],
       [ 0.90910396, -0.86270135, -1.39942355],
       [ 0.62988973,  1.29284048,  0.88235775],
       [-0.93516826, -0.46392588, -0.03564079],
       [-0.4186468 , -1.35571405, -1.54141233],
       [ 0.83225168, -0.51676824,  0.36241514],
       [ 1.12436476, -0.99562355, -1.48458789],
       [-1.14082807, -1.22787669,  1.05515174],
       [ 0.60792418, -1.04297311, -1.64061526],
       [ 0.4050523 ,  0.75652182, -1.20963382],
       [ 0.21710746,  1.43934522, -0.04414158],
       [-1.12694058, -0.21011164,  1.19344943],
       [ 0.46889423,  1.07383187, -1.5291859 ],
       [ 0.66830418, -0.90571345,  0.18203562],
       [ 0.40165893,  0.43398187, -0.86294658],
       [ 0.05753718, -1.10454655, -0.10815033],
       [ 0.598153  , -0.97823091,  0.26139269],
       [-0.09935624, -1.23591939,  0.74391794],
       [-0.3567514 ,  0.68376748, -0.96526399],
       [ 0.23356358,  1.22558502, -0.94563115],
       [ 1.49606063, -0.74537826, -0.19054232],
       [-0.60105024, -0.60498618, -1.15895355],
       [ 0.0912709 , -0.12670318, -0.59219987],
       [ 1.04392789, -1.14510241,  0.41315038],
       [ 1.15213348, -0.09693354, -0.48869217],
       [ 0.47781586, -1.14023073,  0.08124241],
       [ 1.3464263 ,  1.0647202 ,  0.8679326 ],
       [ 0.08738693, -0.42860941,  1.04695586],
       [-0.11849821,  1.65984915,  0.8584026 ],
       [ 0.90388057, -0.54230157,  1.80073399],
       [-1.42954602,  0.24701421, -0.51440881],
       [ 0.91523802, -1.02090852,  1.21800837],
       [ 1.26985453,  0.41587581,  0.14012868],
       [-0.62602446, -0.62370988,  0.18391281],
       [ 1.40288423,  1.08052145, -1.55039185],
       [-0.24842329, -1.67158492,  1.36417057],
       [ 0.30672288, -1.71009534,  1.6898972 ],
       [ 0.44067323,  0.4089069 ,  1.09451005],
       [ 1.25036102, -1.31919179,  1.86913555],
       [-1.18789372, -0.84951405,  0.09073989],
       [-0.6889074 , -1.06094297,  1.50375565],
       [-0.29382715,  0.14550569, -1.11772106],
       [-1.40459726,  1.54288787,  1.40209837],
       [ 0.61252469,  0.53348065, -0.32953474],
       [-0.80092194,  0.03439304, -0.20862978],
       [ 0.64815889,  0.11712008,  1.31180084],
       [-1.59441669, -1.0701843 ,  0.91942949],
       [ 0.06384979,  1.72036478,  0.73891985],
       [ 1.21889489, -0.7470347 , -1.49533752],
       [ 0.89455226,  1.16776208, -0.13618229],
       [-0.94199852,  0.57593711, -0.0741447 ],
       [-1.39799053,  0.59076153, -1.20874493],
       [-1.52521501, -0.34268163,  1.1325334 ],
       [ 0.60666907, -1.32436696,  0.08199469],
       [ 0.80046472,  0.58651418,  0.42844277],
       [-1.29620764, -0.25002248,  0.09730351],
       [-0.52275242, -1.6072742 ,  0.94502107],
       [-0.89820065,  0.78027912, -0.162889  ],
       [ 1.12050019,  0.4749247 ,  1.42394666],
       [-0.49538118, -0.89635663,  0.13684547],
       [ 0.3864307 , -1.61515424, -1.08918174],
       [-0.33456437,  0.33969418, -1.59924391],
       [-0.45104746,  1.6674313 , -1.09691908],
       [-0.50815784,  1.7673254 , -1.00184215],
       [-1.65405827,  1.70757874,  0.48320382],
       [ 0.9182798 ,  0.13169275, -0.39478238],
       [ 1.05538016, -0.13962336,  1.70259913],
       [ 1.55037276,  0.92842565, -0.59202675],
       [-0.64022313, -1.73804874,  1.87843192],
       [-1.1860881 , -0.03697811,  1.19751378],
       [-1.23262066,  1.52222163, -1.09753934],
       [-1.55144138,  1.58791243, -1.24452299],
       [-1.23052473,  1.4041929 ,  1.96784103],
       [-0.99675395, -0.99968276, -1.39857274],
       [-1.67167413,  0.70268582, -0.57407193],
       [ 0.5224098 , -0.00656187, -1.47528993],
       [ 0.65317398,  0.41266126, -1.65219653],
       [ 1.41468019,  1.46652941,  0.01827586],
       [ 1.27253993,  1.68848644,  0.18709085],
       [ 0.44668425, -0.8220625 ,  1.07366239],
       [-0.73359849,  0.82853677,  1.09986285],
       [ 1.43900407,  0.77423015, -0.72760562],
       [-1.51709169, -0.6882845 , -1.24838523],
       [-1.44787542,  0.91993419, -1.15370465],
       [-1.5814817 , -1.38045437,  0.22671984],
       [ 1.44330623, -1.80957081,  0.10837523],
       [ 0.9909861 ,  1.425966  , -1.13268452],
       [ 1.22036723, -0.25630208,  0.33990026],
       [-1.56590333,  0.68677834,  1.21909347],
       [ 0.07986985,  0.29122412, -0.06758261],
       [-1.06912854,  0.25597023,  0.13113971],
       [ 1.26957041,  0.26627828,  0.25679191],
       [-1.0425452 ,  0.47074922, -0.49834438],
       [-1.38217411, -0.27856627,  1.30132283],
       [ 1.48243083,  1.14477048,  0.239333  ]])
```

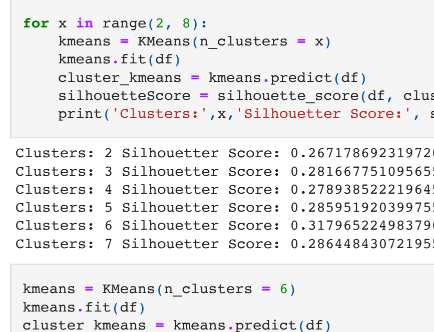
In [6]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components = 3)
pca.fit(scaled_data)

print("The principal components are:")
print(pca.components_)
print("The explained variances are:")
print(pca.explained_variance_)
```

```
The principal components are:
[[-0.34661949 -0.62400095  0.70034116]
 [-0.88580976  0.46334319 -0.02557658]
 [-0.3085385  -0.62923438 -0.71334991]]
The explained variances are:
[1.23825565 1.00174029 0.79000407]
```

In [7]:
```
df = pca.transform(scaled_data)
plt.scatter(df[:,0], df[:,1])
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7fb0d39a2ca0>
```



In [8]:
```
# K-Means Clustering
#testing for clusters with the highest silhouetteScore
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans

for x in range(2, 8):
    kmeans = KMeans(n_clusters = x)
    kmeans.fit(df)
    cluster_kmeans = kmeans.predict(df)
    silhouetteScore = silhouette_score(df, cluster_kmeans, metric='euclidean')
    print('Clusters:',x,'Silhouetter Score:', silhouetteScore)
```

```
Clusters: 2 Silhouetter Score: 0.2671786923197202
Clusters: 3 Silhouetter Score: 0.2816677510956554
Clusters: 4 Silhouetter Score: 0.2789385222196457
Clusters: 5 Silhouetter Score: 0.2859519203997556
Clusters: 6 Silhouetter Score: 0.31796522498379015
Clusters: 7 Silhouetter Score: 0.286448430721955
```

In [9]:
```
kmeans = KMeans(n_clusters = 6)
kmeans.fit(df)
cluster_kmeans = kmeans.predict(df)
```

In [10]:
```
clusterListing = Selected.copy(deep=True)
clusterListing['Cluster'] = cluster_kmeans
clusterListing
```

Out[10]:

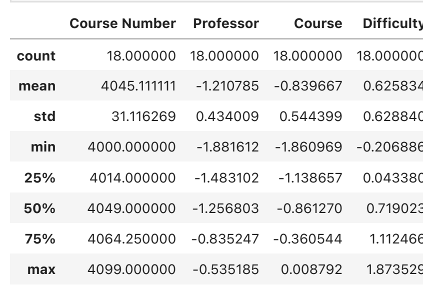
	Course Number	Professor	Course	Difficulty	Cluster
0	4000	-1.204699	-1.142584	0.606862	0
1	4001	1.521202	-0.186447	-1.153300	1
2	4002	-1.125400	-0.838220	0.202952	0
3	4003	-0.095452	-0.938903	-0.614851	1
4	4004	1.584738	1.208184	-1.331121	2
...	...	...	...	...	...
96	4096	-1.238082	0.330791	-0.025611	3
97	4097	1.627054	0.342121	0.110961	2
98	4098	-1.205515	0.566869	-0.709801	3
99	4099	-1.621594	-0.256754	1.246266	0
100	4100	1.887829	1.307731	0.091984	2

101 rows × 5 columns

In [11]:
```
# Use cluster_means as the indicator for clusters
plt.scatter(df[:, 0], df[:, 1], c=cluster_kmeans, cmap='viridis')

centers = kmeans.cluster_centers_

# plotting the centers onto scatter plot
# c is for color, s is for dot size, and alpha is for transparency
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=100, alpha=0.5);
```



In [12]:
```
#Get a summary of Cluster0
cluster0 = clusterListing[clusterListing['Cluster']==0]
cluster0.describe()
```

Out[12]:

	Course Number	Professor	Course	Difficulty	Cluster
count	18.000000	18.000000	18.000000	18.000000	18.0
mean	4045.111111	-1.210785	-0.839667	0.625834	0.0
std	31.116269	0.434009	0.544399	0.628840	0.0
min	4000.000000	-1.881612	-1.860969	-0.206886	0.0
25%	4014.000000	-1.483102	-1.138657	0.043380	0.0
50%	4049.000000	-1.256803	-0.861270	0.719023	0.0
75%	4064.250000	-0.835247	-0.360544	1.112466	0.0
max	4099.000000	-0.535185	0.008792	1.873529	0.0

In [13]:
```
#Get a summary of Cluster 1
cluster1 = clusterListing[clusterListing['Cluster']==1]
cluster1.describe()
```

Out[13]:

	Course Number	Professor	Course	Difficulty	Cluster
count	20.000000	20.000000	20.000000	20.000000	20.0
mean	4033.350000	0.529399	-0.295217	-1.467057	1.0
std	28.236361	0.809971	0.811090	0.428116	0.0
min	4001.000000	-1.149416	-1.725888	-1.963928	1.0
25%	4011.500000	-0.143655	-0.965407	-1.802635	1.0
50%	4023.500000	0.607041	-0.138139	-1.585584	1.0
75%	4057.000000	1.251376	0.262734	-1.141497	1.0
max	4082.000000	1.564971	1.229758	-0.614851	1.0

In [14]:
```
#Get a summary of Cluster 2
cluster2 = clusterListing[clusterListing['Cluster']==2]
cluster2.describe()
```

Out[14]:

	Course Number	Professor	Course	Difficulty	Cluster
count	21.000000	21.000000	21.000000	21.000000	21.0
mean	4057.904762	1.274705	1.036714	-0.243286	2.0
std	30.306608	0.559402	0.506903	0.838892	0.0
min	4004.000000	0.169554	0.194189	-1.853276	2.0
25%	4038.000000	0.843382	0.571459	-0.811624	2.0
50%	4060.000000	1.444428	1.208184	-0.216125	2.0
75%	4084.000000	1.721210	1.396560	0.110961	2.0
max	4100.000000	1.971064	1.905366	1.379547	2.0

In [15]:
```
#Get a summary of Cluster 3
cluster3 = clusterListing[clusterListing['Cluster']==3]
cluster3.describe()
```

Out[15]:

	Course Number	Professor	Course	Difficulty	Cluster
count	15.000000	15.000000	15.000000	15.000000	15.0
mean	4068.466667	-1.260956	0.864147	-0.959289	3.0
std	21.357055	0.519927	0.744435	0.522125	0.0
min	4024.000000	-1.976260	-0.707103	-1.525024	3.0
25%	4056.500000	-1.690857	0.448830	-1.391592	3.0
50%	4069.000000	-1.238082	0.801012	-1.217298	3.0
75%	4084.000000	-0.969091	1.391606	-0.552355	3.0
max	4098.000000	-0.365350	1.992023	-0.025611	3.0

In [16]:
```
#Get a summary of Cluster 4
cluster4 = clusterListing[clusterListing['Cluster']==4]
cluster4.describe()
```

Out[16]:

	Course Number	Professor	Course	Difficulty	Cluster
count	20.00000	20.000000	20.000000	20.000000	20.0
mean	4043.25000	0.911536	-0.978251	0.666198	4.0
std	24.58899	0.610817	0.578202	0.754900	0.0
min	4011.00000	-0.232638	-1.939584	-0.375249	4.0
25%	4025.25000	0.576071	-1.331928	0.009692	4.0
50%	4036.00000	0.878104	-1.049257	0.460664	4.0
75%	4053.00000	1.354127	-0.539627	1.271885	4.0
max	4093.00000	1.904527	0.178172	1.863425	4.0

In [17]:
```
#Get a summary of Cluster 5
cluster5 = clusterListing[clusterListing['Cluster']==5]
cluster5.describe()
```

Out[17]:

	Course Number	Professor	Course	Difficulty	Cluster
count	7.000000	7.000000	7.000000	7.000000	7.0
mean	4066.142857	-1.090971	1.549045	1.038227	5.0
std	21.835969	0.855597	0.473096	0.530081	0.0
min	4034.000000	-1.954678	0.804321	0.357049	5.0
25%	4050.500000	-1.747872	1.276509	0.699921	5.0
50%	4070.000000	-1.435808	1.745329	1.027298	5.0
75%	4082.000000	-0.450246	1.900121	1.256345	5.0
max	4094.000000	0.149928	1.940406	1.970708	5.0

In [18]:
```
#Cluster 0  professor-good  Course-Good  Difficulty - Relative Difficult
#Cluster 1  professor-ok    Course-ok    Difficulty - Relative easy
#Cluster 2  professor-good  Course-ok    Difficulty - Relative Difficult
#Cluster 3  professor-ok    Course-Great Difficulty - Difficult
#Cluster 4  professor-ok    Course-Good  Difficulty - Relative easy
#Cluster 5  professor-great Course-Ok    Difficulty - Easy

# Students could make the decisicion based on the factor they consider as
# the most important to them.

#Instance, if a student would like to select a easy course with a
#caring and kind professor, he might want to take a deep look at cluster5

clusterListing = clusterListing[clusterListing['Cluster'] == 5]
clusterListing
```

Out[18]:

	Course Number	Professor	Course	Difficulty	Cluster
34	4034	-0.073467	1.873889	0.764854	5
48	4048	-1.649064	1.745329	1.355800	5
53	4053	0.149928	1.940406	0.634988	5
70	4070	-1.954678	1.926352	0.357049	5
78	4078	-1.435808	1.592880	1.970708	5
86	4086	-0.827025	0.960138	1.027298	5
94	4094	-1.846680	0.804321	1.156891	5

In [19]:
```
#If the student cares more about the difficulty level than professor rating,
#we could rank the list from the easiest to most challenging
clusterListing.sort_values(by = ['Difficulty'])
```

Out[19]:

	Course Number	Professor	Course	Difficulty	Cluster
70	4070	-1.954678	1.926352	0.357049	5
53	4053	0.149928	1.940406	0.634988	5
34	4034	-0.073467	1.873889	0.764854	5
86	4086	-0.827025	0.960138	1.027298	5
94	4094	-1.846680	0.804321	1.156891	5
48	4048	-1.649064	1.745329	1.355800	5
78	4078	-1.435808	1.592880	1.970708	5

In [ ]: