Identifying trends - Suggesting template changes
&
SCAR: Student Courses Automated Recommender

Abhishek Gunda 14807257
Gopichand Kotana 14807249
Gaurav Chawla 14237
Kaushal Agrawal 14313
Siddharth Mahapatra 14817693

CS685A Project: Group 29

1 Introduction

The data containing course details of IITK and courses done by a particular student of IITK is available to the residents of IITK at [1]. We use this rich data resource to make a data set which includes the department of each student from batches Y11 to Y16 and the courses each one has done during their stay at IITK. We use this data to identify trends (batch-wise and year-wise) existing in the courses being offered and suggest changes in current official template by identifying disparity in the courses being taken by students and the ones suggested by the template. We also built a course recommendation system, ***SCAR*** (Student Courses Automated Recommender) which uses *Epsilon ball Nearest Neighbor algorithm* for recommendation. The choice of epsilon, which we call tolerance, is given to the user (optional); Small tolerance suggests that the recommendations are given using the data of those students who had done courses very similar to the user. In the following sections, we present the implementation details, results, and possible future work that could extend the work done by us.

2 Implementation Details

2.1 Scraping

The first step is scraping data from [1]. The code used is in the python notebook, *Scraping.ipynb*. For every batch from Y11 to Y16, we collect two kinds of data - courses done by each student of a particular batch, which is stored in *y[batch].csv*, and name and department of each roll number in the batch, which is stored in *y[batch]_details.csv*.

Domain expertise comes in handy at every facet of this project. For example, during scraping, knowing the roll number range for batch isn't trivial from [1]. However, we know that ESC101A is compulsory for all freshers, and looking at the people who have done it in their first year lets us know the roll number range for that particular batch.

2.2 Data Processing

The list of departments we considered for the project are:

Aerospace Engineering(AE)	Biological Sciences & Bio-Engineering(BSBE)
Chemical Engineering(CHE)	Civil Engineering(CE)
Computer Science & Engineering(CSE)	Electrical Engineering(EE)
Mechanical Engineering(ME)	Material Sciences & Engineering(MSE)
Economics(ECO)	Mathematics & Statistics(MTH)

The data is in two files for each batch with filenames *y[batch]_details.csv* and *y[batch].csv*
Files *y[batch]_details.csv* contain one-to-one relation between academic roll numbers and corresponding department. Files *y[batch].csv* contain details of the courses done by a student in a semester-wise manner.

From the above listed departments, roll numbers are filtered out and three json objects are created. All these json objects segregate the data into department and for each department, results are compiled in a semester-wise manner for underlying objects.

- Batch: Stores details of the courses done by a student in a semester-wise fashion.
- Batch_courses: For all courses, stores the count of students from the same department as well as from other departments who availed that course. For all the courses not coming under above listed departments, total count of students is stored in the same fashion under the section **others**
- Batch_aggregated: Stores the count of students in the same department taking a particular course.

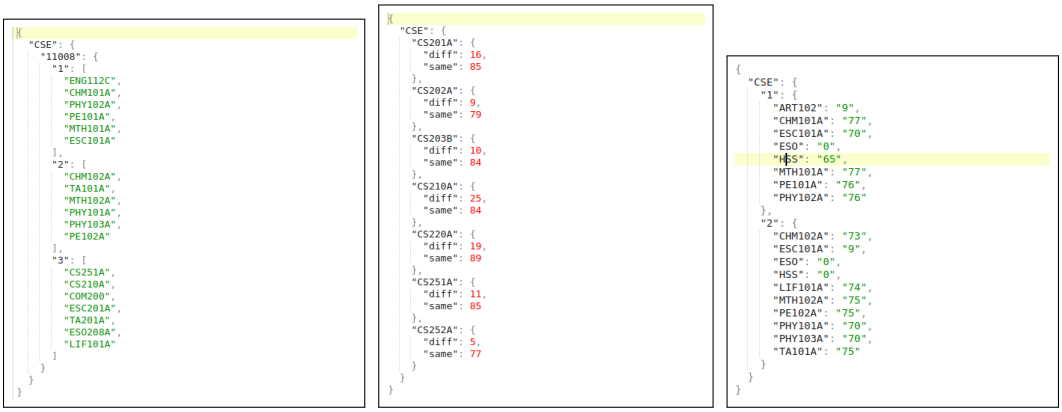

Figure 1: JSON Structures

Data processing for template suggestion is done in the python notebook *Template Suggestions.ipynb*
Data processing for aggregated trends batchwise is done in the python notebook *Batchwise.ipynb*
Data processing for aggregated trends batchwise is done in the python notebook *Yearwise.ipynb*

2.3 SCAR

2.3.1 Introduction to SCAR

Using the data-files *y[batch]_details.csv* and *y[batch].csv* created during data processing (see 2.2), we create three consolidated data-dumps viz. *courses.csv*, *departments.csv*, and *dept_rollnos.csv* storing informations as follows:

- courses: Corresponding to each roll number, stores a semester-wise list of courses
- departments: Stores a many-to-one mapping of roll number to department for the students

- dept_rollnos: For each department, stores a mapping of roll number to the number of usual semesters completed by the student. Here, usual semester refers to academic registrations except for summer semesters

Once we have the above files, we can run the method *semesterRecommender* from *recommender.py* which takes roll_no and optionally takes few other parameters (see 2.3.2) to recommend a list of courses for the given roll number. Since we already have all relevant data for batches Y11 to Y16, the method first tries to use that data in order to recommend the courses. In case data is missing for the queried roll number, we first scrape the list of courses done by that student and then use it against the stored information of students from batches Y11 to Y16 to get an epsilon ball matching with provided tolerance and finally make a majority voting to get the list of recommended courses.

2.3.2 Course Recommendation Platform

Using the above methods and data-sets we created a webapp to recommend courses to students using Django framework in python. We request the following information from the user

- User Roll Number
- Semester for which the user needs recommendation
- Number of courses required
- Tolerance level

Except roll number, all other fields are optional. Return value is a list of course recommendations based on the trend of courses being taken by the IITK students from batches Y11 to Y16.

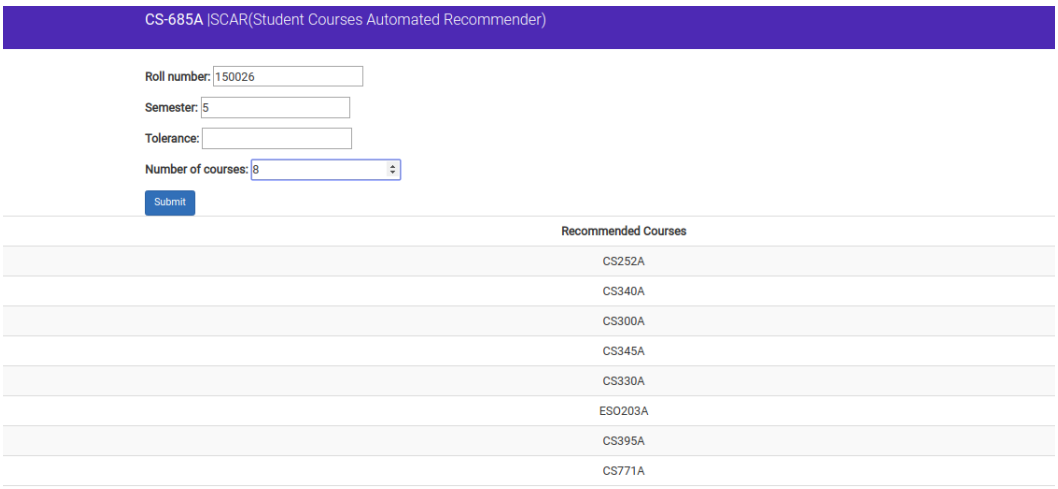

Figure 2: **SCAR** sample screenshot

2.3.3 Diving deeper into the recommender

Given the query roll number, first step is to fetch the student's department. If we have the student's details in our saved databases, we use it directly, otherwise make a GET request to [1]. If successfully retrieved from the saved data, we also fetch the list of courses done by the student, or else proceed to make another GET request for the same using *getCourseList* method. If the user doesn't provide a semester number (call it *semno*) for the recommendation list, we assume that the user needs recommendation for his/her next semester.

The next step is to go through course-lists of all the students from the same department as the user and considering only the students who've done more semesters than *semno*, we check if the list of

courses done by the user is a subset of the list of courses done by this student (with a tolerance level as provided). If it is a subset, we add the list of courses done by this student in the semester *semno* to a voting booth of courses. Doing this for all the relevant students, we get a majority voting of courses across students from Y11 to Y16 who are from same department as the user, have done almost all the courses as the user, and are senior to the user. We use this to recommend top few courses as requested by the user. Note that this method is particularly useful for students of Y16, Y17, Y18, and hopefully Y19 as we have a rich dataset of seniors and the algorithm works very well for these students.

3 Results

3.1 Suggestions for Template making

Among the json dumps created in 2.2, the file *Batch_aggregated* contains aggregated count of students taking a particular course for each department. These counts can be used to suggest template changes, if any, in a semester-wise fashion.

The code under the *Template Suggestions* in the **Template Suggestions.ipynb** takes department and semester as variables and generates a list of courses in a sorted order for each of the batches Y11, Y12, Y13, Y14 and Y15.
These five lists are checked manually and are used to make suggestions to the template and subsequently draw inferences.

AE

The Aerospace department has a very rigid template with a lot of compulsory courses until their eighth semester. There is very little room to branch out or explore. AE641A, AE673A, AE 618A are the most famous DEs.

- AE641E is being done by almost all the students by the end of their stay. It can be made into a compulsory course, and some other courses can be offered as a DE.

BSBE

- Semester 3: Majority of students are opting for ESO208A and ESO201A as the elective ESO.
- Semester 5: Students are opting to take courses from their department as an Open Elective. This can be attributed to the leverage in grading policy in these courses.
- Semester 8: Students after Y12 batch seem to be uninterested in the UGP course BSE499A. Lately, people are opting to take courses like CHM242A and NT602A as an elective.

CHE

- Semester 3: A very relaxed semester. Some students are taking COM200, which is a fourth semester course. In Y13, 35 out of 68 have taken it in 3rd semester, in Y15 18 have taken it in 3rd semester, and 52 in 4th semester. It can be included in the 3rd semester.

CE

- Semester 3: Electives chosen by students are different from batch to batch. Also CE412A (Waste Water Management) is taken by far more students than CE462A (Applied Hydrology). This can be attributed to the easy content and grading policy of CE412A.
- Semester 7: Number of people taking HSS courses decreased in Y13 and Y14 batch. This is because students do an extra HSS course in the summer. Also number of students taking CE481A and CE451A are decreasing continuously over the years. This can be attributed to the inclusion of new courses like CE666A.

CSE

- Semester 3: For the ESO most people are opting for ESO204A and ESO201A. These two ESOs are being taken by majority of the students from Y12 to Y15 batches. The reason is - even though there are multiple choices, CSE students are accepted majorly in these courses. A variety of ESOs have to be offered to mitigate this.
- Semester 5: Template suggests to take an ESO but lately from Y14 onward, people are preferring to opt for ML (CS771A) with 33/100 from Y14 and 56/110 from Y15 taking it. ML could be made into a compulsory course, or ESO can be shifted to another semester, shifting a DE into this slot.
- Semester 6: Template suggests to take 2 DEs this semester and from Y12 onward, majority of people are taking CS315A - Databases (55%) and CS628A - Systems Security (40%)
- Semester 7: Template suggests to take 1 DE and 2 OEs for the DE part. Students from Y11 onward are taking CS425A - Computer Networks with around 50% and lately from Y13 onward students are opting towards taking departmental courses as OE with people preferring ML (CS771A) or Data Mining (CS685A). One OE can be changed into a DE as too many students from CSE are opting for DE than an OE.

EE

- Semester 5: Significant number of students are taking Data Structures and Algorithms. Y12 has 42 students, Y13 has 65 students, Y14 students are also seen taking Machine Learning in this semester. Data Structures and Algorithms can be made compulsory for them - it is a course in great demand and is useful for them.

MSE

- Semester 5: There are 8 compulsory courses in this semester which make the schedule hectic for students. So students generally choose to do an extra course - HSS or ESO in the summer term.
- Semester 8: MSE639A and MSE638A are the most chosen electives. Lately from Y14 batch, students are choosing ECO413A. This shows growing inclination of students towards Economics.

ME

- Semester 6: Students start taking specialized and non-traditional courses like ME623A - Finite Element Method and ME766A - Robot Motion Planning. The common thing among these courses are that they do not have an equivalent UG course. These courses are something new which students face and are more engaging in terms of coding assignments.

ECO

- Semester 5: Instead of an OE there should be a DE since an OE is been taken as a course from the department itself. ECO523A has been taken by almost the entire batch of Economics students since Y13 onward, 31 out of 36 students in Y13, 33 out of 36 students in Y14 and 25 out of 34 students in Y15
- Semester 6: The number of students taking an HSS in this semester has gone down by almost 50% from Y12 and Y13 to Y14 and Y15. So HSS can be avoided in this semester. Game Theory and Monetary Economics are been taken as DE/OE by almost all students whenever offered.
- Semester 7: Y13 started taking IME courses as OE. 18 out of 36 took IME courses in this semester. Placement Effect can be observed in this semester as students start taking courses like Machine Learning and Data Structure along IME and MBA courses in this semester 12 out of 34 students in Y14 took Data Structures and IME courses.

MTH

- Semester 3: People before Y15 are been observed to take ESO207 - Data Structures(60%) as their ESO. Data Structures and Algorithms can be made compulsory to them which has been implemented by the Department from Y15 onward.
- Semester 8: Template suggests to take 1 DE and 2 OEs and among these, from y13 onward majority of the people are opting for MTH628A (around 50%) which can be made into a compulsory course.

3.2 Aggregated Trends: Batchwise & Yearwise

We wanted to see how the demand for every course has changed over the years, both batchwise and yearwise.

By batchwise demand of a course we mean the number of people from a particular batch that have done a particular course in their four years stay here. We take batches Y11, Y12, Y13, and Y14 for our analysis. We omit Y15 and Y16 as they have not graduated and hence might not have had the chance to certain courses yet.
This trend is helpful in realizing what kind of courses a particular batch was inclined to do.

Yearwise demand of a course is the number of people doing a particular course in a year (could be from any batch). We consider year 2014-2017 for this analysis. We ignore the years before 2014 as the statistics of those years are affected by batches before Y11. Since the batches before Y11 would have graduated by 2014, we consider the statistics only from 2014.

Trends Observed: Here, we present some examples of courses which had a stark increase or decrease in demand. Note that we can choose our own statistic to identify these particular courses. In the code, the statistic is the difference between maximum and minimum over years/batches. We say the course has an interesting trend if the difference is more than x% of the maximum. The x was chosen empirically, and is different for each trend. The hyperparameters used can be found in the code. By replacing the code of this statistic with another, you can observe other trends that you want to.

3.2.1 Batch-wise

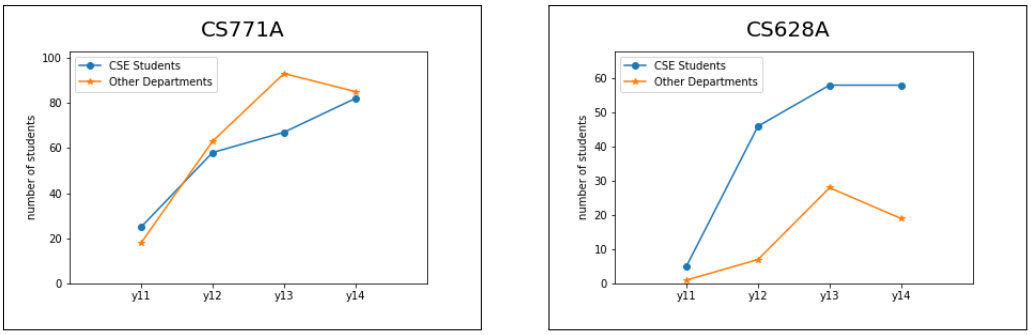

Figure 3: Courses with increasing demand

We know that the demand for Machine Learning and Security courses is increasing every year, and we can see that in the plots.

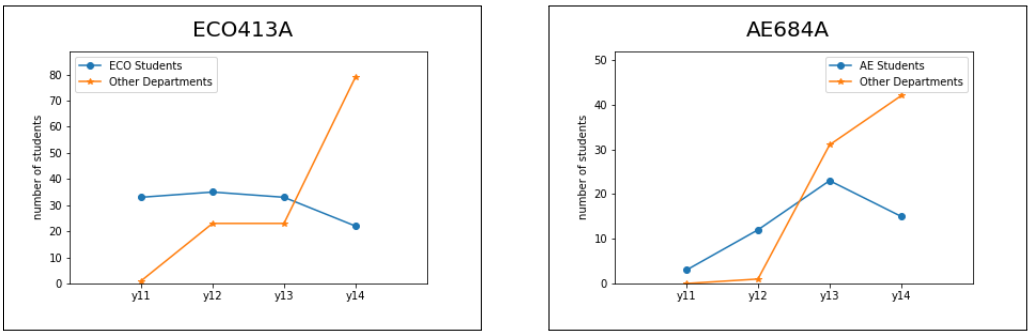

Figure 4: Courses with an increasing demand from other departments

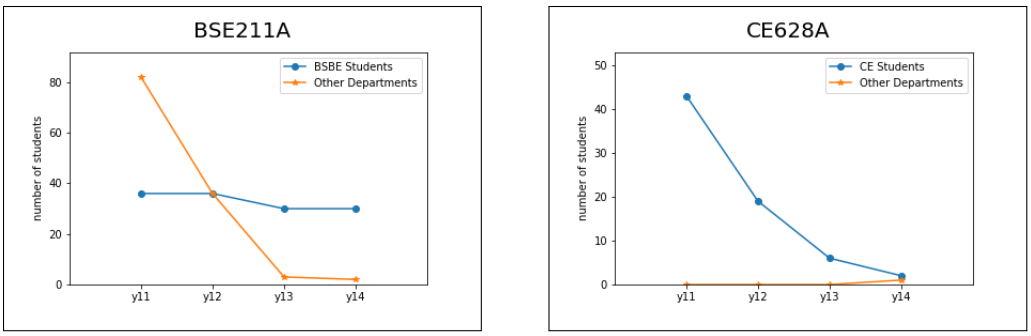

Figure 5: Courses with an decreasing demand. BSE211A has a stark decrease from other departments and CE628A has a stark decrease in demand among it's parent department.

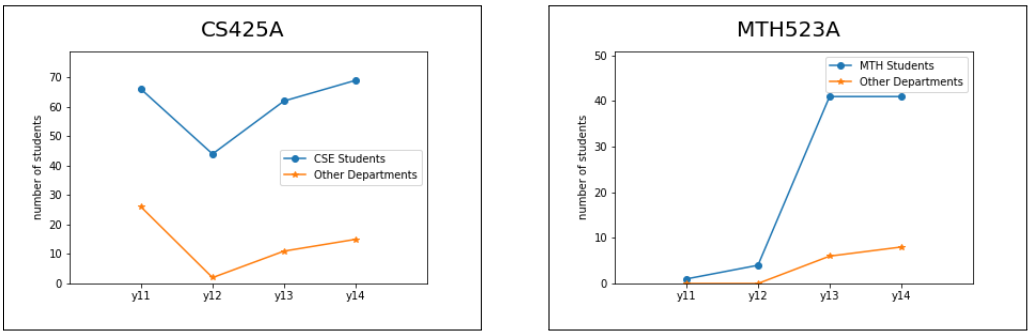

Figure 6: CS425A has an interesting trend, the number of people from CS and other departments vary proportionally over the batches. MTH 523A shows the importance of domain expertise. There seems to be a stark increase in the demand, but in reality it's only because it's been made into a compulsory course.

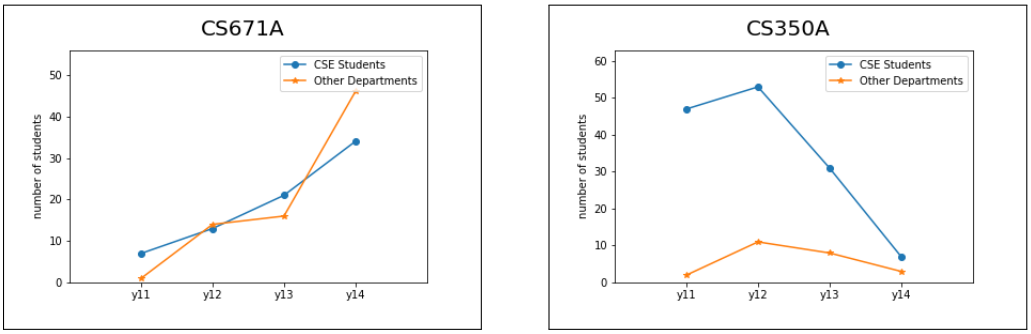

Figure 7: This shows an interesting trend in the CS department. Principles of programming language which was once a very famous course is taken by a very few people now. NLP has quickly become popular.

3.2.2 Year-wise

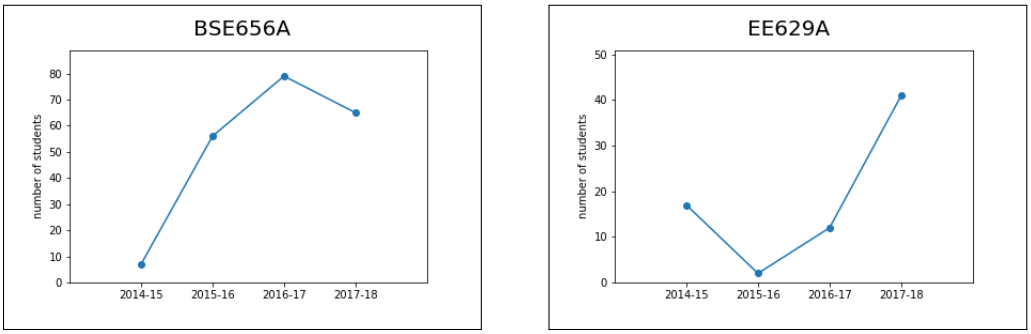

Figure 8: Courses with increasing demand

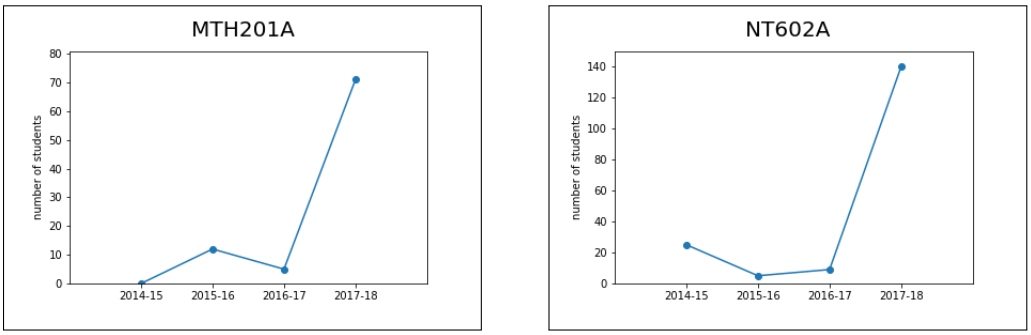

Figure 9: Courses with increasing demand. NT602A has become popular among the final years as an OE as it has lenient grading policy.

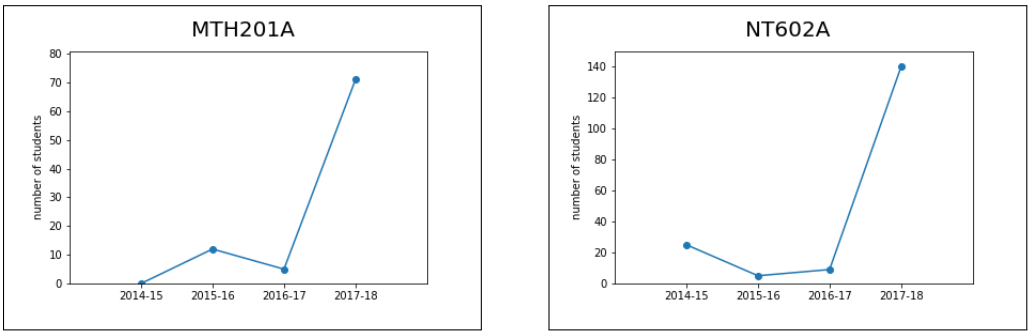

Figure 10: Courses with increasing demand. NT602A has become popular among the final years as an OE as it has lenient grading policy.

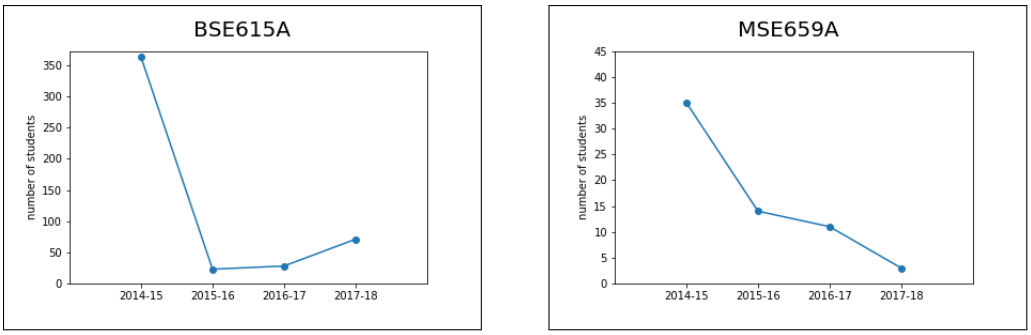

Figure 11: A lot of people used to take Bioelectricity and Bioelectronic Devices(BSE615A) but the number has significantly decreased. And similarly for MSE 659A

These are just a few examples from the courses with the trends that we were trying to observe. More can be seen in the trends/batchwise/plots and trends/yearwise/plots. It's important to note that domain expertise is essential to infer and make sense of these plots.

4 Conclusion

We suggested some changes to templates of different departments. We used the help of students of each department to understand the data and to find the reasons behind some trends that we have discovered. A lot of manual work was needed as the final decision to recommend a suggestion rested on us. We also built a recommendation system to help the current Y16/17 and upcoming batches in choosing courses. Domain expertise is necessary for this project. These results can help in policy making and understanding student needs. This helps departments in offering courses that are interesting and inclined with student interests.

References

[1] *Dataset* http://172.26.142.68/dccourse/