CMPE 230 – PROJECT2

ABDULLAH YILDIZ – MELTEM ARSLAN

2018400291 - 2016400117

BEAUTIFUL SOUP

We imported Beautiful Soup4 to reach the pages of the urls.

LISTS

donem_list : Contains all terms between "98/99 -1" - "18/19-3" (We add the terms manually with append function.)

department_list : Contains tuples which has "department's complete name" as a first item and "department's code " as a second item. (We add the tuples manually with append function.)

url_list : Contains the middle part of all urls of all departments (We add the url parts manually with append function.).

FINDING GIVEN TERMS FROM ALL TERMS

starting_year : is starting year which is extracted from given statement.
finishing year : is finishing year which is extracted from given statement
starting index : is the index of the starting term. It is calculated in an if/else statement.
finishing index : is the index of the finishing term. It is calculated as the same way with the starting index.

We find the given terms from all terms by using starting index& finishing index in a for statement. In the for statement we obtain the complete url(url_composed) by adding the common part(url1), and the different parts (donem_list & url_list). Then we use the url in BeautifulSoup.

DICTIONARIES

my_dict : Contains the set of departments as keys and the tuple of course's complete names and codes as values for each departments
my_dict_instr : Contains the set of instructors for each departments.
semesterDict : Contains tuples of course list and semesterTemp set(current semester's values)

EXTRACTING CODES/NAMES/INSTRUCTORS FROM PAGE

We reach the codes/names/instructors by finding its headers('td' , 'class', 'schtitle', 'td'). We can find the headers by using find&find_all functions in soup. If the header is not found. There is no such lecture.
If a course has a lab or ps, we use "continue". Because it should not be considered. We used sets(course_list_set, course_list_instructors) to keep the extracting courses and instructors.

CALCULATING "U" "G" "I"

If the first value of the code of course is less than and equal to 4, it is regarded as an undergraduated course. Else if the first value of the code of course is grater than 4, it is regarded as a graduated course. If the course exist in the given term, we add 'x' for the course; if it is not exist, we add a whitespace character.

We obtain "I" from the length of the "my_dict_instr" of the current department.

CONSTRUCTING&FILLING TABLE

We use pandas.framework to construct the table. First we give the names('Dep./Prog.(name)', 'Course Code', 'Code Name') of the columns. And we assign a null value to them. We use 3 flags to fill the table. We assign “false” to them and then we reassign them with every given semester. If the flags are “true”,we extend the table for one column and fill the first square of the semester with its name. When all terms are completed, we add one more column to the table for the ‘Total Offerings’. When all the above operations are completed, we construct the table with empty values. Then we fill the table with the values which are already calculated. We use for to reach all departments and courses. So we fill the table with the current values of ‘U’ ,‘G’ and ’I’.