Degree Deviation Detection Software

User Manual and Documentation

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, K.K. BIRLA GOA CAMPUS

Python Libraries to be Installed

- xlrd
- xlwt
- collections
- json
- os
- sys
- re
- pandas

Input Files

All input files are present in data folder.
Input file names (and sheet names in the case of excel files) should not be changed, except for files in reg_data folder. If file names are changed (apart from those in reg_data folder), then they should be changed in the python files and the shell scripts.

Input File Descriptions:

- Time Table.xlsx – timetable of current semester in excel format. Must have a sheet named “erp time table”.
- CURRENT_SEM_REGISTRATION_DATA.xls – registration data excel sheet of current semester.
- coursetype.xlsx – excel file with the elective tag for each course
- noofcourse.xls – excel file with the branch-wise total number of courses to be completed for graduation
- semester_identifier.xlsx – excel file giving the mapping between year of admission to the current semester number using values like ‘1142’
- Semester_wise_number_of_courses.xlsx – excel file with branch-wise semester wise count of different type of courses to be done so that the student does not lag behind the “plan” for his branch. Example – the data for “3-1” is the number of courses the student should have completed from 1-1 up to 3-1 (both including).
- students.xlsx – list of ID numbers of the students whose degree deviation and lag are to be checked.
- sorted.xlsx – Excel file with all courses (and course details) already completed by the students. It is automatically generated from the input files mentioned above but is also used as further input.
- reg_data folder – contains excel files of the results (i.e., grades) of previous semesters – i.e., it should contain the data of all semesters the students in students.xlsx have completed.

Steps to run the software*

- Open terminal in the folder containing all the files of this software.
- Type: **bash run.sh**, then press enter.
- Results will be obtained in the result folder.
- Commands which are run are:

```
• python3 coursetype.py data\coursetype.xlsx
• python3 check_absence_in_coursetype_excel.py
• python3 noofcourse.py data\noofcourse.xls
• python3 makesort.py
• python3 studentdata.py data\sorted.xlsx
• python3 jsontoxls_pre.py
• python3 logic.py
• python3 jsontoxls_final.py
• python3 find_lagging_students.py
• python3 suggest_courses.py
```

***Please see the section "Key Points and Running the Software in a New Semester" before running the code**

Output Files

Output files are present in result folder.

Output File Descriptions

- courses_absent_in_coursetype_excel.xlsx – Lists the courses which are present in Time Table.xlsx but not present in input file coursetype.xlsx. Please ensure all these courses are listed appropriately in coursetype.xlsx, else they will be counted as OPEN elective (which may give wrong results).

- final_pending_courses.xls – for each student, gives the following details: CDC Pending, DEL1 Pending, DEL2 Pending, HUM Pending, OPEN Pending, PROJ Flag and ELEC Flag.
  Note: These details **do not include courses registered for in the current semester.**

- final_tag.xls – for each course **completed (not including courses registered for in the current semester)** by each student, gives the TAG = [CDC, HUM, DEL1, DEL2, OPEN] = course type.

- simplified_lag_output.xlsx – lists the students who are lagging behind the "suggested plan", in which course type they are lagging and by how much.
  Includes courses registered for in the current semester by checking CURRENT_SEM_REGISTRATION_DATA.xls

- detailed_lag_output.xlsx – for each lagging student, by checking Time Table.xlsx, gives the list of courses the student can register in the current semester to compensate the lag. Clash checking and pre-requisite checking is not done. OPEN courses are not listed due to the large number of options.
  Includes courses registered for in the current semester by checking CURRENT_SEM_REGISTRATION_DATA.xls

Key Points and Running the Software in a New Semester

1. **Correctness of coursetype.xlsx, Semester_wise_number_of_courses.xlsx and semester_identifier.xlsx is extremely important. Data should be such that all lagging students can be detected.**
2. The paths for the files have / in linux systems (ex. data/coursetype.xlsx) and \ in windows systems (ex. data\coursetype.xlsx)
3. Ensure all the input files are updated with latest student and course data. Please refer Input Files Section. Ensure all file names and sheet names in the excel files are the same as before.
4. Time Table.xlsx Must have a sheet named “erp time table”.
5. Type the following command in terminal opened in current folder and press enter: python3 coursetype.py data\coursetype.xlsx
6. Type the following command in terminal opened in current folder and press enter: python3 check_absence_in_coursetype_excel.py
7. courses_absent_in_coursetype_excel.xlsx – Lists the courses which are present in Time Table.xlsx but not present in input file coursetype.xlsx. Please ensure all these courses are listed appropriately in coursetype.xlsx, else they will be counted as OPEN elective (which may give wrong results).
8. Students IDs can be copy-pasted batch-wise into students.xlsx so that output is also obtained batch-wise.
9. The code for finding lag and suggesting courses will not run appropriately if CURRENT_SEM_REGISTRATION_DATA.xls is registration data of a summer term.
10. **The code for finding lag and suggesting courses caters to most students. It skips students such as only MSc, MSc+MSc, reverse dual, BE dual, backlog.**
11. studentdatalog.json does not indicate backlog. This software does not detect backlog.
12. Software has not been written for higher degree students.
13. Please refer to Output Files section to check which output files include current semester’s registered courses.
14. Install required libraries

Code File Descriptions

Comments for particular functions are not present in every file since they are used repeatedly across multiple files. If looking for comments, please search for the function in every python file.

- run.sh – contains the commands needed to run the individual python files

- python3 coursetype.py data/coursetype.xlsx – takes input as coursetype.xlsx and prepares a json file coursedesc.json for the "tag" for each course.

- python3 check_absence_in_coursetype_excel.py – gives the output as courses_absent_in_coursetype_excel.xlsx – Lists the courses which are present in Time Table.xlsx but not present in input file coursetype.xlsx. Please ensure all these courses are listed appropriately in coursetype.xlsx, else they will be counted as OPEN elective (which may give wrong results).

- python3 noofcourse.py data/noofcourse.xls – takes noofcourse.xls as input and prepares a json file noofcourse.json.

- python3 makesort.py – takes all excel files in reg_data folder and prepares sorted.xlsx

- python3 studentdata.py data/sorted.xlsx – takes input as sorted.xlsx and prepares two json files studentdatarf.json (details of successfully **completed** courses by each student, semester-wise) and studentdatalog.json (details of courses in which the student **obtained a grade** other than ['', 'A', 'A-','B','B-','C','C-','D','E','Good','Poor'], such as NC, W)

- python3 jsontoxls_pre.py – uses data in studentdatarf.json and prepares final_tag.xls

- python3 logic.py – processes studentdatarf.json and prepares finaldata.json, which consists of course ids of all **completed** courses and the number of pending courses of each type, for each student.

- python3 jsontoxls_final.py - finaldata.json is converted to final_pending_courses.xls

- python3 find_lagging_students.py – prepares two json files student_course_count.json (for each student, contains the count of courses of each course type **completed or registered in current semester**, and a list of all courses **completed or registered in current semester)** and student_lag_list.json (same details as in student_course_count.json, but only of students lagging in any course type).

- python3 suggest_courses.py – using student_lag_list.json, for each student lagging in any course type, prepares simplified_lag_output.xlsx and detailed_lag_output.xlsx