PROJECT REPORT

The dataset given contained the following - student_id, level, course, grade and major. In the training set, has 10000 unique entries for each student id. The testing set has 2000 of these students.
To extract features from both the training and test set, I first started to process training data and separately. But in this approach, I encountered that the courses for training and testing were exclusive for some cases. Thus, during the encoding of the categorical data, it generated some un-labeled courses in testing. Therefore, I concatenated the training and testing data, separating them with 'train' and 'eval' labels as additional columns.
This way same methodology for feature extraction could be applied. Later the concatenated data were split again into training and testing data after preprocessing both.

PREPROCESSING:

During preprocessing, the grades were converted from strings to weighted values with A+ being the highest. In this process, it is assumed that the students who get better grades in a course are interested in that course and thus, are likely to pursue that as a major down the years. During this process, I found additional grade categories apart from the 20 given, such as – I, R, WX. These were treated as noise and their entries removed, along with other grades like - S, AUS, AUU, F, IP, N, U, P. The rest were kept. This was done because these grades were very less in counts – AUS = 184, AUU = 7, I = 43, N = 3, P = 87, F = 0, U = 56 compared to count of other grades in thousands.

Now to group the data, I took the parameters as grade, frequency and difficulty aggregated over the student ids. The mean of the 12 grades were taken. The frequency here represents the number of times a student has taken a particular course. That is decided on the basis of the string part of the course. For example, a student taking the course ASIA:260 was split into ASIA and 260 where ASIA is the course and 260 is called the course number in my dataframe. So, for each student, the frequency of that course was computed. Now, for the difficulty part, the remaining part of the course data, i.e. the course number has useful content. It can represent the level of the course taken. Generally, if the MSB of a course number is higher, it is a higher level course. So course numbers with 700 and 100 has 7>1, thus 700 course is a more difficult one. Later this difficulty will be one-hot encoded.

After grouping the data, it is sorted on the basis of course frequency. This gives the total data size to be 68221. Since it is a very large number, I decided to take the top three courses for each student. But on checking the data, I found that it contained cases where only one data entry is given for student. So, in that case, to consider each student, I only took the top course frequency values for each student. Thus, the dataset reduced to 11963 for training and eval.

Now, the data is one-hot encoded for categorical data – course and difficulty. In the end, our concatenated data is split into training and testing on the basis of the labels 'train' and 'eval'.

TRAINING:

After pre-processing, my training data has now become 9969 with 114 categories. For training I have split it into 70%-30% for training and testing respectively. I used random forest model for training my dataset. This is because Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Grid-Search was done to find best parameters in number of trees, max - features and maximum depth of tree. Values for these were chosen in the following ranges –
Estimators(trees) - [200, 300, 400]
Max depth – [7,8 , 9]
Max features = [sqrt, log2]

The best params obtained after 10-fold cross validation were were estimators = 400, max depth = 9, max features = log2. The mean values for accuracy 10 folds were stored.

The mean accuracy for train was – 84% and test was – 82%.

PREDICTION:

The prediction values for testing data is stored. However, since the problem asks to predict the top three majors for each student, we need to calculate probability of student having a particular major. The top three majors were found by sorting the probability values and mapping them with classes. In the end, the data is aligned with the original evals data provided by left merging evals with this resultant dataset.

The output file is stored as eval.csv.