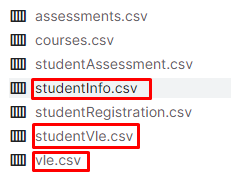
* This project’s ideas are summarised in “project summary.pdf”; More details of this project’s design, implementation and results can be found in “thesis-yutongLiu-568983-submit-v2.pdf”
* This project used open-source data from Kaggle (link below), 3 out of 7 datasets are used - studentInfo.csv, studentVle.csv and vle.csv

<https://www.kaggle.com/datasets/anlgrbz/student-demographics-online-education-dataoulad>



* This project generated a set of features based on the raw data (link above). The file “data structure of all strategies.xlsx” shows the details of the created features (i.e. Strategy 1, 2, 3 data structures)
* This project demonstrated a practical use case in data manipulation, feature engineering, and ML models.
  + **Data manipulation**: This project solves some commonly seen practical data issues through manipulation, such as concatenating, mean transformation of missing data, re-binning data categories to create balanced dataset
  + **Feature engineering:** Both the time frequency and students’ activity features are considered in feature engineering, to transform datasets with various dimensional combinations. Such a method can be confidently applied in the practical case on panel data with similar time-series and behavior patterns.
  + **ML models:** The project firstly applies widely used tree-based machine learning algorithms to train GBT and Random Forest (RF) models. To compare with other commonly used techniques, this project also trains logistic regression and K-NN type classification models. With pros and cons of those four model types in mind, the project further trains deep-learning models - Convolutional Neural Network (CNN) with enhanced feature extraction to improve predictability. Given the input data might have long-term dependencies over time, the project takes one step further to train long short-term memory (LSTM) models to prevent neural network failures due to particular input decaying or exploding. This method leads to a highly performed prediction model with over 90% accuracy.

Methodology wise, this project demonstrates the deep-learning models perform better than traditional machine-learning ones for similar types of datasets. In particular, the LSTM model with high accuracy can be an optimal model design option for education sectors to track and predict the students’ performance. This could be meaningful not only for educators to guide their curriculum improvement, but also informative for policy-makers to design or fine-tuning education supporting programs with accurately measurable or even predictable students performance. It therefore provides a timely and advanced analytical method to meet the emerging technical needs for the digitalization transformation of education sectors. This can also be empirically applied to any industry with similar data type and technical demands, for example, to understand, track and predict target groups’ behavior such as clients’ consumption behavior.

To sum up, the feature engineering design under this project is able to compare and suggest which behavior aspects are the key to influence the performance, for example, students’ click behaviours on the homepage, subpage, forum and resources are important influencer of their course performance; and during the entire time period, weeks 22 to 35 are vital. Same techniques can be used in other industry cases, to analyse the influencing aspects on clients consumption or default behaviors in marketing, risk analysis fields or even supporting policy-makers to understand the implementation outcomes of existing programs.

* Python codes and datasets

| file | details |
| --- | --- |
|  | Step1: three raw datasets from Kaggle website |
|  | Step2: process raw data to create two temporary datasets |
|  | Step3: process temp data to produce the 6 datasets.   * S1 was produced using Python, * S2 and S3 were produced using RapidMiner (not included in the shared folder) |
|  | Step4: the 6 datasets were produced |
|  | Step5: using 4 traditional machine learning to train models, implemented in RapidMiner (not included in the shared folder)  (In the left screenshot of RapidMiner, planA = S1, planB = S2, planD = S3) |
|  | Step6: using 2 deep learning to train models, implemented using Python |