Project Plan

# Description

The goal of this project is to utilize the current transcript data and historical end-of-course survey to improve the online learning experience and learning efficiency of students.

We intend to implement a regression model fitting features extracted from both datasets to accomplish the task of predicting NPS score as well as provide visualization on outliers for course designers to take a closer look at.

# Datasets

Based on the goal of the project, two datasets are provided by the capstone partner in a shared google drive where we can access directly:

## EndOfCourseSurvey

Domain: The data is collected from students who finished the course

Format: csv

size: 21 columns and 619122 rows

columns discription: **(useful columns highlighted)**

* 'FactId': int64, this is the unique identifier of each survey.
* 'DateCompleted': str, this is the date of the survey completed by the user.
* **'Question\_1': “How likely is it that you would recommend this course to a friend or colleague?”**
* **'Response\_Q1': int64, This is the answer to “Question\_1”, on a scale of 1 to 10. This will be the most valuable information in the dataset because our final goal is to improve the course quality and customer satisfaction.**
* **'Question\_2': “This course covered the topics and content that you were expecting.”**
* **'Response\_Q2': str, “Strongly Agree”, “Agree”, “Neutral”, “Disagree”, ”Strongly Disagree”, “Did Not Respond”. This column can be transformed into integers to represent the satisfaction of customers.**
* **'Question\_3': “The course materials were engaging, clear, and informative.”**
* **'Response\_Q3': str, “Strongly Agree”, “Agree”, “Neutral”, “Disagree”, ”Strongly Disagree”, “Did Not Respond”. This column can be transformed into integers to represent the satisfaction of customers.**
* **'Question\_4': “The instructor was effective in delivering the course content.”**
* **'Response\_Q4': str, “Strongly Agree”, “Agree”, “Neutral”, “Disagree”, ”Strongly Disagree”, “Did Not Respond”. We would not use this column because we don’t evaluate the instructors in the project.**
* **'Question\_5': “Do you have any other feedback to help us improve our course?”**
* **'Response\_Q5': These are free responses so we are not gonna use the data.**
* 'CustomerID': This is the ID of the customer who finished the survey
* **'CourseName': This is the name of the course. There are 132 course categories in this column. This will be used to match the course transcripts in the SRT dataset.**
* 'Instructor': This is the instructor of the course.
* 'Program': This is the abbreviation for the program that the course belongs to. There are intersections between some programs so this data can be tricky.
* 'Requirement': This is the type of the course.
* 'Level': This indicates the level of the course, but mixed with int and str type. The data needs to be transformed into a universal type.
* 'VideoLenght': str, This is the length range of the course.
* 'Price': This is the price of the course.
* 'SurveyType': all values are “NewSurvey” in this column.

## SRT files

Domain: The data is the transcripts of each courses with timestamp

Format: zipped srt files

size: 132 courses (folders) with the chapter srt files in it

There are 6533 available files in total and one unencodable file (we just ignore it)

* the shortest srt file has 74 characters (13 tokens) and the longest one has 10198 characters (2180 tokens)

# Expected deliverables

The external partner is interested in Net Promoter Score (NPS) of the courses, it is done in the end-of-course survey on a scale of 0-10, 0 being least satisfied and least likely to recommend and 10 being the most satisfied and most likely to recommend. Given the transcript of each course in a chapter-by-chapter format with time stamps, the external partner expected us to come up with some sort of labels that indicates and reflects on this metric. We also proposed to dig into this NPS from the customers’ perspective incorporating additional data if time permitting. We will organize the code implementation to make the results that we have reproducible and conduct presentation slides weekly that is easy for any stakeholder to understand.

# Methods

## Data handling:

* For the EndOfCourseSurvey.csv file, we have used the Python Pandas library to load the .csv file and ~~calculated the average Response\_Q1 score for each value in CourseName, and saved the results in a Pandas DataFrame.~~  calculated the NPS for each course as the final label, converting columns into binary features.
* For the SRT files.zip file, we have used the Python zipfile library to extract and read all files in the .zip file. We have then decoded the UTF-8-encoded strings, removed time stamp lines, and used the defaultdict() from collections to store the course name as keys and the list of the transcripts as values. Finally, we converted the dictionary to Pandas DataFrame and merged the two Pandas DataFrame to obtain our final training data.
* Feature Extraction: We are going to extract linguistic features such as the summary statistics, reading ease, the lexical density, coherence and the type to token ratio. We plan to use sent\_tokenize() and word\_tokenize() from the Python NLTK toolkit to tokenize the sentences and words. We will also use the spaCy library to parse and tag raw text strings.

## Implementation:

We plan to implement several baseline models.

* ~~For neural models:~~
  + ~~We used the transformers library to download and train the state-of-the-art pretrained transformer models. We have attempted to use~~ [~~Longformer~~](https://arxiv.org/pdf/2004.05150.pdf)) ~~with an attention mechanism to process long documents that contain more than 512 tokens (~~[~~tutorial~~](https://www.kaggle.com/code/sumantindurkhya/bert-for-regression/notebook)~~), and we plan to try other models such as~~ [~~BERT~~](https://huggingface.co/docs/transformers/model_doc/bert) ~~and~~ [~~RoBERTa~~](https://huggingface.co/docs/transformers/model_doc/roberta)~~.~~
  + ~~We may also attempt to use RNN models with PyTorch, such as LSTM and GRU.~~
* For non-neural models:

We plan to use linear models such as Ridge and tree-based regression models such as CatBoostRegressor with scikit-learn.

## 

## Evaluation of the deliverables:

Since we are working on a regression problem, we are going to use the MSELOSS() from PyTorch to evaluate the model performance.

## 

# Schedule

In the first two weeks, we are going to explore the data thoroughly and get an idea of what the project is trying to deliver, and we will have brainstorming meeting sessions to communicate ideas and findings from each group member.

In the third and fourth week, we are going to do related research on the topic and try to apply robust models that have already been widely used as our baseline and explore more about the task.

In the fifth and sixth week, we are going to add any additional features to the current implementation to finalize the deliverable and do fine-tuning on existing work to maximize the performance.

In the last two weeks, we will use this period of time to wrap up the whole project and conduct the final presentation and report. The detail of work distribution will be determined when time comes closer.

On a weekly basis, there are two meetings happening regularly and any additional meetings by appointment with mentor or external partner, one on Monday 10 am PST with mentor to discuss what has been done in the previous week, and what we should be planning on doing this week, to get an insight of the direction of project and make sure everyone is on the right track and on the same page, the other one on Wednesday 10 am PST(subject to change) with both mentor and external partner to present the weekly findings in a clear, straightforward and informative way.