**Current Situation:**

Microsoft has developed a set of API’s, known as Cognitive Services, that undertake a variety of functionality in areas of computer vision, speech, language, knowledge, and search. Of particular interest is the Emotion API, which can distinguish various emotions on multiple faces on a picture or video. For pictures, the API returns a JSON object containing the position of the face and a list of emotions and their corresponding probabilities of applying to that face. For videos, the API does the same but also tracks these measurements over time.

Current web and app content is designed to engage users, i.e. evoke emotional responses. Currently used metrics of such emotional engagement are centered around clicks, time spent on a piece of content, questionnaires, etc. Although useful, such metrics can often lack insight on their own.

**Problem Statement:**

In this project, we aim to add an element of emotion recognition to develop a highly accurate recommendation and feedback engine. The business case comes from a need to assess users’ engagement levels. For instance, in an educational platform, there is a need for the content-provider to gauge the engagement of the students in order to improve the course material. Similarly, in an online shopping context, adding an element of emotion can enhance a platform’s recommendation engine, thus providing more relevant items to the user and creating a deeper emotional experience.

Essentially, the project aims to integrate the Emotions API inside a platform to provide accurate emotional engagement metrics to the content-provider in response to different material available on the web/app, as well as provide real-time recommendations to the user based on a combination of emotional responses and the traditional metrics of engagement.

**Proposed Solution:**

We propose the creation of an educational platform, containing online lectures, which will be able to provide feedback to the content-provider on the different levels of engagement at different portions of the lecture, and aggregate levels over the entire course. The levels of engagement will be calculated based on results from the Emotion API for pictures taken of the user at specified intervals, as well as more traditional metrics of engagement, such as clicks and time spent on a page. The feedback will be provided in the form of a graph of engagement level (similar to Bing Pulse). Furthermore, the same application can be used in other contexts such as broadcast calls.

We have split the project’s ambition into two stages, the MVP stage and post-MVP stage. For the MVP stage, we aim to detect the users’ engagement levels from facial expressions and other simple methods, and ultimately display this to both the user and the content-provider. At the post-MVP stage, we would like to add further functionality, such as course recommendation based on students’ levels of engagement in certain courses, career recommendations, and the use of further inputs to get a more accurate description of emotion.

**Technology:**

The platform will make use of the Emotions API to obtain preliminary emotional responses. Furthermore, we will apply a second level of machine learning to deduce the levels of engagement from the emotional responses and other metrics, using open-source libraries. In terms of the recommendation engine that will be implemented in the post-MVP stage, we will either employ the Microsoft Recommendations API, which is also part of Cognitive Services, or we will create a recommendation engine from scratch, again using open-source libraries, to have greater flexibility in using the input features that we require.

The basic break-up of the project can be seen overleaf. The webapp will consist of a front-end, serving as the data collector and displayer, and a back-end which will handle all the data processing. Specifically, the front-end will display the products available for purchase, along with the action buttons (such as “View Course”, “Enroll”, etc.) and will display the recommendations generated by the engine. The back-end will consist of a server and the recommendation engine, which will be hosted on Azure. The server will be in charge of communicating with the Emotions API and feeding the results to the recommendation engine, either to train it or to make an actual recommendation. Interfacing between the different modules will be through JSON or CSV files (for the recommendation engine).

