# Assignment 1: Creating an Application with HuggingFace API and Gradio.io

## 1 Introduction

In this assignment, you will learn how to use the HuggingFace API and Gradio.io to create an interactive application for natural language processing tasks. You will utilize the power of the HuggingFace Transformers library and Gradio.io's user-friendly interface to build an LLM based application of your choice.

## 2 Tasks

1. Go over the slides and lecture material for HuggingFace API and using Gradio.io to create a web interface. There are multiple python notebook examples posted in case you missed the lecture.

2. Choose one of the Interface Pipelines not discussed in class (ie something other than sentiment analysis or NER - name entity recognition).

```
from transformers.pipelines import PIPELINE_REGISTRY

#Get the list of tasks that are supported by Huggingface pipeline
print(PIPELINE_REGISTRY.get_supported_tasks())
```

3. Create a python notebook demonstrating use of the pipeline with some sample example. You should also change the default model of the pipeline to another suitable model from the HuggingFace model list (search for keywords matching your interface name) and demonstrate examples with the new model.

4. Create a Gradio App (visualization) of your pipeline.

## 3 Submission

The following should be submitted.

- A video recording go over the code and demonstrating it works.
- A short report of the interface that you choose and what it supposed to demonstrate.
- The code used for the demo.

## 4 Marking Scheme

Your submission will be evaluated based on the following criteria:

- Working python notebook demonstrated by video (50%): Emphasis will be on the clarity of your explanations and how the code works.

  **There will be no marks for the assignment without a video recording. Video recordings where there is no audio (speaking in English) will not be accepted.** Remember to demonstrate understand of the actual code, avoid saying "well this runs and I get the answer".

- Code (25%): Properly written code clearly demonstrate that you understand the code (from your video explanations). A different model than the default model is also used for the interface and demonstrated.
- Report (15%): Importance will be given to how well-organized and understandable your report is, and how effectively it communicates your insights.
- 
- Discretionary (10%): Discretionary marks cover any components not covered by the above.

## Pitfalls

Avoid the following pitfalls.

- If something doesn't work or you can't get everything working it's not a big deal, it happens in life, just state it in your video and explain what you think is wrong. Do not try to hide details. An honest effort gets most of the marks.

- "Mimicking" where you just ask for someone's assignment and just mimic their results is considered an academic misconduct and will result in a mark of zero (for both the mimicker and supplier). Everyone must work on assignments independently. You can discuss solutions in general, but just running code that you have obtained from someone else is not considered discussion.
- Requirements for assignment must be followed as clearly stated. Changing requirements will result in a mark of zero.
- Late penalty is 10% per day, please respect course policies on late penalties and do not ask for extensions outside the scope discussed in first class.