# **Plagiarism Project, Machine Learning Deployment**

Purpose: to be able to detect whether a paper was originally written in English or it has been translated from a foreign language in order to catch cross-lingual plagiarism.

STEP BY STEP METHODOLOGY:

1. Collect originally French data (CS conference papers) and their manual translations
2. Use Google Translate to translate each chunk of French data into English
3. Compare the machine translations and manual translations to the original French in order to find features that indicate machine translation (red flags, things that don’t quite make sense, etc.)
4. Use these features to inform how we build our final model.
5. Build a model with the intention of being able to differentiate between genuine English and MT English. The features from the previous step will be included in the learning as things that point toward MT.
6. Train this model on a dataset that’s half genuine English and half MT English.
7. Test it on a portion of that dataset.
8. Wahoo.

This repository contains code and associated files for deploying a plagiarism detector

## **Project Overview**

In this project, we were tasked with building a plagiarism detector that examines a text file and performs binary classification; labeling that file as either *plagiarized* or *not*, depending on how similar that text file is to a provided source text. This project will be broken down into three main notebooks:

**Notebook 1: Data Exploration**

* Load in the corpus of plagiarism text data.
* Explore the existing data features and the data distribution.

**Notebook 2: Feature Engineering**

* Clean and pre-process the text data.
* Define features for comparing the similarity of an answer text and a source text, and extract similarity features.
* Select "good" features, by analyzing the correlations between different features.
* Create train/test .csv files that hold the relevant features and class labels for train/test data points.

**Notebook 3: Train and Deploy Your Model in SageMaker**

* Upload your train/test feature data.
* Define a binary classification model and a training script.
* Train your model and deploy it.
* Evaluate your deployed classifier.