# ECE 364 Project Option #2: IMDB movie reviews
## Spring 2025

## 1 Project Background

This project is based on the IMDB Dataset of 50K Movie Reviews. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 45,000 highly polar movie reviews for training and 5,000 for testing, you can separate partial data from the training set as validation set if you wish to monitor accuracy in training. So, predict the number of positive and negative reviews using either classification or deep learning algorithms.
The dataset comes in the form of `.csv` files. Each file contains three fields:

1. `Id`: ID assigned to a review.
2. `Review`: The content of the movie review.
3. `Label`: Either `Positive` or `Negative`.

This is a binary classification problem. We want to classify each test example in one of two classes. For evaluation, please generate and submit **prediction.csv** file to kaggle in the following format. Accuracy score will be calculated by kaggle as the evaluation metric.
**Note:** To ensure effective evaluation, head and content of prediction.csv file must exactly match the format, case of text matters.
You can use the ” submit prediction” button at the upper right corner in kaggle page to submit, remember to write names of all group members in the description. After submitting, you need to go to the ” Submissions” tag and press the ” select” box after your best submission, so your submission will be shown on the leaderboard.
**Note: use the original id provided by the test.csv, the ids in the table are only for reference**

| Id | Label |
|---|---|
| 1 | Positive |
| 2 | Positive |
| 3 | Negative |
| … | … |

**Table 1.** Format of **prediction.csv**

## 2 Deliverables

The following are the deliverables of this project:

- A binary classifier to classify each review in the test set as `Positive` or `Negative`.
- The number of parameters in the model must not exceed 10 million.
- You are free to use any publicly available model (pre-trained or otherwise) with or without augmentation, but it is not a requirement. You can also augment the data as you see fit.

As a starting point, check HuggingFace[1] for readily available pre-trained language models. Try models like Tiny-BERT. The idea is to get some hands-on experience with training models, so don't worry too much about getting a very high accuracy score. Any reasonable model is fine.

## 3 Submission

1. Submit all your code, including training and evaluation, as a `.zip` file.
2. Submit **prediction.csv** file to Kaggle for scoring and evaluation
3. Submit a 2-page report (1-inch margin, 12-point font) and include
   - Your approach, model, and any other design choice.
   - Hyperparameters that you used for training.
   - Training and test results.
   - Any other interesting details about the approach or model.

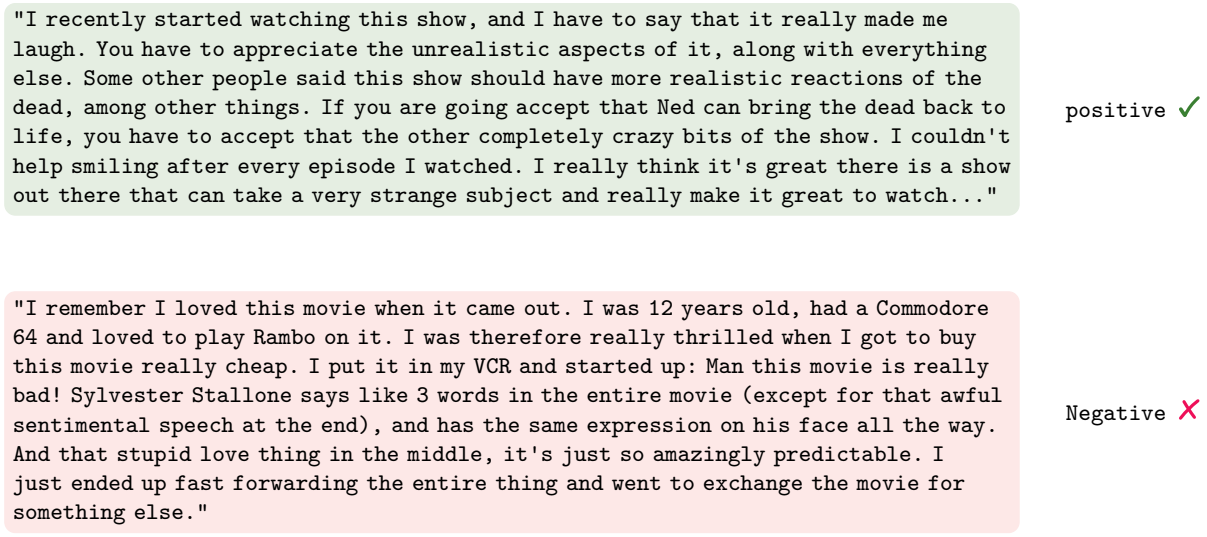


**Fig. 1.** An positive and negative example from the training set.

[1] `https://huggingface.co`