# Sentiment Analysis Assignment Report

## 1. Executive Summary

*This assignment revolves around Sentiment Analysis using the Cornell Sentence Polarity Dataset. The goal is to predict whether a given movie review snippet carries a positive or negative sentiment. Two models are constructed and compared: a traditional supervised learning model and a deep learning-based neural network.*

*The initial steps involve text preprocessing, such as tokenization and vectorization, to prepare the data for modeling. The conventional model utilizes engineered features from the preprocessed text, while the deep learning model leverages a fully connected feedforward neural network to automatically learn relevant features. Hyperparameter tuning is emphasized for optimal performance in both models. The assignment serves as a comprehensive exploration of sentiment analysis techniques, offering insights into the efficacy of conventional and deep learning approaches in text classification tasks.*

## 2. Data Understanding and Preparation

### 2.1 - Data Description

The starter notebook dataset contains 5331 positive and 5331 negative snippets of movie reviews. Authors assumed snippets (from Rotten Tomatoes webpages) for reviews marked with ‘fresh’ are positive, and those for reviews marked with ‘rotten’ are negative.

### 2.2 - Data Preparation using NLP techniques

- *Converted the sentences to words using gensim utilities*
- *Defined a function for removing stop words using the stop words from nltk english library*
- *Tried both stemming and lemmatization on sample reviews using nltk library*
- *Went ahead with lemmatized results*
- *Split the data in training and validation dataset*
- *Created n-gram model using TfIdfVectorizer*
- *At the end of the preprocessing, the shapes of datasets were:*
    - *Training: (8529, 500)*
    - *Validation: (2133, 500)*

# 3 – Conventional ML Model

## 3.1 - Model Building

*Built several ML models to figure out which fits the best to the dataset and gets highest accuracy. The models included:*

- *Logistic Regression*
- *Decision Tree*
- *kNN*
- *Random Forest*
- *LGBM*

## 3.2 – Model Evaluation

*Light GBM model got the highest RMSE values of 52% for training dataset and 71% for validation dataset.*
*ROC AUC for training dataset using LGBM model is 0.813 and 0.499 for validation dataset*
*Plotted the Precision-Recall curve for visualization.*

# 4 – Deep Learning Model (FCFNN)

## 4.1 - Data Preparation

*Defined a function to convert datasets into sparse matrices for Keras model using tensorflow library. Then applied the function on both training and validation dataset.*
*Unfortunately, for the model that was built ultimately it turns out that converting to sparse matrix is not required. So, decided to opt out of it.*

## 4.2 - Basic Neural Network Model

*Defined a neural network function with activation function and number of neurons in the hidden layer as arguments. The function was basic and it has 2 hidden layers, binary crossentropy as loss and rmsprop as optimizer.*
*The basic model had 144,769 total parameters and trainable parameters. It had the validation accuracy of 50%.*

### 4.3 - Hyperparameter Tuning

*Chose ‘sigmoid’ & ‘relu’ for activation functions and 128 & 256 as number of hidden neurons in hidden layer.*

*Used gridsearch function from sklearn library to run through the hyperparameters for finding the optimal combination with highest accuracy.*

## 5 - Final Model

*Created an FCFFN model using the best hyperparameters and 200 epochs. Although the training accuracy was above 90% but validation accuracy was still at 50%.*

## 6 - Way Forward

*Better accuracies can be achieved using more complex models, using more options & more parameters for grid searching in different models.*