# CSE-6363-004
# Sentiment Analysis on Movie Review

**Team 26:**

- Varun Perumandla - 1002082466
- Hemanth Sukumar Vangala- 1002118951
- Sathwik Reddy Avula – 1002120527

**Data Preprocessing:**

In this project, data is preprocessed before being used for testing and training. Data was first removed from the folder, and the texts were divided based on labels.

```
import os
import pandas as pd

def read_text_files(directory, sentiment):
    data = []
    for file in os.listdir(directory):
        if file.endswith('.txt'):
            with open(os.path.join(directory, file), 'r', encoding='utf-8') as f:
                data.append((f.read(), sentiment))
    return data

def create_dataframe(data_directory):
    combined_data = []
    for sentiment in ['pos', 'neg']:
        sentiment_dir = os.path.join(data_directory, sentiment)
        combined_data.extend(read_text_files(sentiment_dir, sentiment))
    return pd.DataFrame(combined_data, columns=['text', 'label'])

# Load and create DataFrame for training and testing data
train_data_directory = 'aclImdb/train'  # Path to the training data
test_data_directory = 'aclImdb/test'  # Path to the testing data

train_df = create_dataframe(train_data_directory)
test_df = create_dataframe(test_data_directory)
```

Subsequently, stopwords, punctuation, and HTML elements were eliminated from the labels and transformed to lowercase. Following filtering, the data appears as follows:

| | text | label | label_encoded |
|---|---|---|---|
| **0** | For a movie that gets no respect there sure ar... | pos | 1 |
| **1** | Bizarre horror movie filled with famous faces ... | pos | 1 |
| **2** | A solid, if unremarkable film. Matthau, as Ein... | pos | 1 |
| **3** | It's a strange feeling to sit alone in a theat... | pos | 1 |
| **4** | You probably all already know this by now, but... | pos | 1 |

Lemmatization is a technique used to interpret a sentence's tone. It is mostly used in sentiment analysis to identify positive and negative sentences. After lemmatization, the data is displayed in the excerpt below.

```
0         im watching star world network overseas buys a...
1         undoubtedly best heavy metal horror item made ...
2         hey arnold slowpaced slightly boring movie plo...
3         movie like bad train wreck horrible still cont...
4         first lets agree lorenzo lamas could never con...
                              ...
24995     cheapest film made 21st century way low qualit...
24996     movie contains personalities deliciously playi...
24997     read great interest available comment made min...
24998     usually dont write reviews cant understand rat...
24999     show incredibly hilarious couldnt stop watchin...
Name: text, Length: 25000, dtype: object
```

Tokenizer has then been used. Tokenizer is used because it makes it possible for machine learning to comprehend the vast quantity of text data.

```
[[   0    0    0 ...   30 1019  198]
 [   0    0    0 ... 7181  204 2664]
 [   0    0    0 ...  647   44  651]
 ...
 [   0    0    0 ...  819   27  410]
 [   0    0    0 ...  277  212   37]
 [   0    0    0 ... 1670  239 5763]]
[[   0    0    0 ... 5313  586 4192]
 [   0    0    0 ... 5894  140  301]
 [   0    0    0 ...  367  339  730]
 ...
 [   0    0    0 ...  333 3457 5033]
 [   0    0    0 ... 1974 2514  224]
 [   0    0    0 ... 1526  230  284]]
```

**Splitting the datasets:**

Since a dataset provides both training and test accuracy, splitting it into smaller parts is crucial. We are splitting the dataset in half for this project, with 80% of the data being used for training data and 20% being used for test data.

```
In [26]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
         X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
```

**Models used in this project:**

**Convolutional neural networks**: Using this model, the provided dataset has been processed to yield the highest accuracy. To achieve the best accuracy, we implemented CNN using a series of layers and passed both train and test data. The model's implementation is seen in the excerpt below.

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Embedding(input_dim=max_features,
                                    output_dim=output_dim, input_length=max_input_lenght))
model.add(Conv1D(128, 5, activation='relu'))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.GlobalAveragePooling1D())
model.add(tf.keras.layers.Dense(16, activation='relu'))
model.add(tf.keras.layers.Dense(num_classes, activation='softmax'))
```

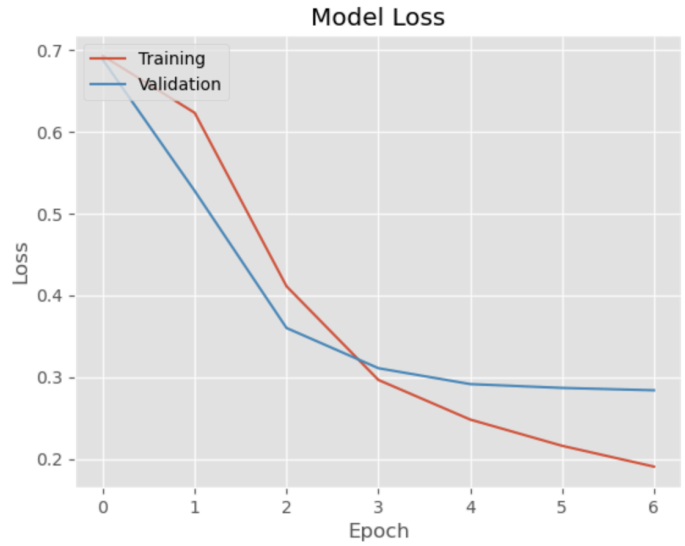
Epochs for train data from CNN:

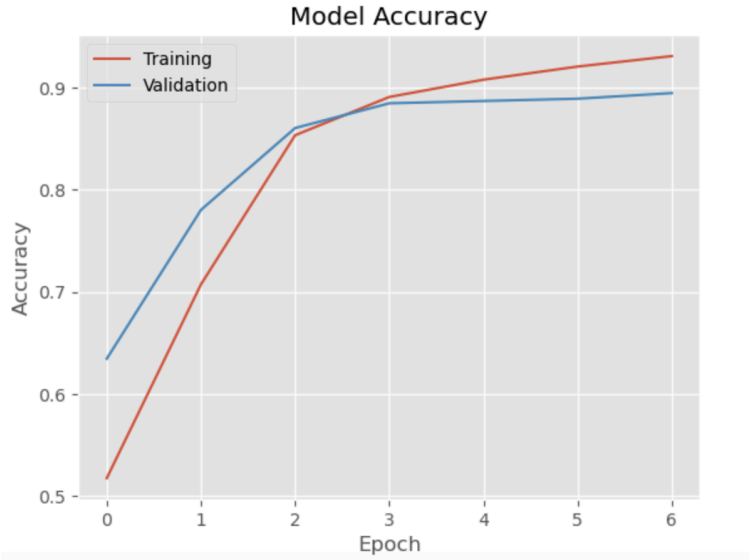
```
Epoch 1/7
2000/2000 ━━━━━━━━━━━━━━━━━━━━ 13s 6ms/step - accuracy: 0.5014 - loss: 0.6932 - val_accuracy: 0.6345 - val_loss: 0.
6878
Epoch 2/7
2000/2000 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - accuracy: 0.6741 - loss: 0.6616 - val_accuracy: 0.7803 - val_loss: 0.
5275
Epoch 3/7
2000/2000 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - accuracy: 0.8328 - loss: 0.4623 - val_accuracy: 0.8605 - val_loss: 0.
3599
Epoch 4/7
2000/2000 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - accuracy: 0.8909 - loss: 0.3060 - val_accuracy: 0.8848 - val_loss: 0.
3108
Epoch 5/7
2000/2000 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - accuracy: 0.9064 - loss: 0.2489 - val_accuracy: 0.8870 - val_loss: 0.
2913
Epoch 6/7
2000/2000 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - accuracy: 0.9204 - loss: 0.2222 - val_accuracy: 0.8892 - val_loss: 0.
2867
Epoch 7/7
2000/2000 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - accuracy: 0.9313 - loss: 0.1923 - val_accuracy: 0.8947 - val_loss: 0.
2838
```

Accuracy from validation data:

```
157/157 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step

Results
Accuracy: 0.90

              precision    recall  f1-score   support

           0       0.89      0.87      0.88      2463
           1       0.88      0.89      0.89      2537

    accuracy                           0.88      5000
   macro avg       0.88      0.88      0.88      5000
weighted avg       0.88      0.88      0.88      5000
```

Graphical Representation of Model Loss and Model Accuracy from CNN



*Bidirectional Encoder Representations from Transformers(BERT):* This approach predicts sentiment from ambiguous messages and is used to handle and interpret them. The BERT model is shown below.

```
# Define model
model = text.text_classifier(name='bert',
                             train_data=(X_Train, Y_Train),
                             preproc=preproc)
```

```
# Define data
train_data = (X_Train, Y_Train)
val_data = (X_Test, Y_Test)
batch_size = 6

# Create learner
learner = ktrain.get_learner(model=model,
                             train_data=train_data,
                             val_data=val_data,
                             batch_size=batch_size)
```

Epochs for train data from BERT:

```
begin training using onecycle policy with max lr of 2e-05...
4167/4167 [==============================] - 33475s 8s/step - loss: 0.2493 - accuracy: 0.8986 - val_loss: 0.1594 -
val_accuracy: 0.9388
```

Accuracy from validation data in the BERT model:

```
782/782 [==============================] - 11102s 14s/step
              precision    recall  f1-score   support

         pos       0.94      0.93      0.94     12500
         neg       0.93      0.94      0.94     12500

    accuracy                           0.94     25000
   macro avg       0.94      0.94      0.94     25000
weighted avg       0.94      0.94      0.94     25000
```

Comparisons of accuracies from both the models:

| Accuracy of CNN | Accuracy of BERT |
|---|---|
| 90% | 94% |

When compared to CNN, the BERT model performs better because to its good accuracy, as evidenced by the observation of both accuracies. In terms of usability, both models are the best.

**References:**


- A Hybrid BiLSTM-ATT Model for Improved Accuracy Sentiment Analysis - Langxue Dang, Chunyu Wang, Hongyu Han, Yan-E Hou – 2022

https://ieeexplore-ieee-org.ezproxy.uta.edu/document/10074634

- Comparative Study of Algorithms for Sentiment Analysis on IMDB Movie Reviews – Neelisetty Sri Lakshmi Sai Charitha, Vellanki Rakesh, Mandadapu Varun – 2023

https://ieeexplore-ieee-org.ezproxy.uta.edu/document/10113113

- Sentiment Prediction of IMDb Movie Reviews Using CNN-LSTM Approach – Mahesh Mishra, Amol Patil – 2023

https://ieeexplore-ieee-org.ezproxy.uta.edu/document/10165155

- ktrain: A Lightweight Wrapper for Keras to Help Train Neural Network

https://towardsdatascience.com/ktrain-a-lightweight-wrapper-for-keras-to-help-train-neural-networks-82851ba889c%E2%80%8B

- BERT Text Classification in 3 Lines of Code Using Keras

https://towardsdatascience.com/bert-text-classification-in-3-lines-of-code-using-keras-264db7e7a358