# Project 3: A Multi-Modal Deep Learning Model for Fake News Detection

CSC 215-01 Artificial Intelligence (Spring 2023)

Raul Arambula (SID:302741322) and Bashar Allwza (SID:301780232)

Due March 27, 2023

## Problem Statement

The circulation of fake news on Web multimedia has grown into a prevalent issue in recent years. The circulation of this fake news leads some who come upon it to believe in its veracity. It has become more necessary now that effective tools are implemented to check the accuracy of posts to prevent the spread of misinformation. The goal of these models is to detect whether a Twitter post is fake or real based on the text of the tweet and its included image. These models were built using Word2Vec, VGG16, and BERT transfer learning.

## Methodology

To have better results on our models we started preparing the data with the following steps.

1. Dropping unimportant columns such as: ‘tweetId’, ‘userId’, ‘username’, and ‘timestamp’
2. Remove tweets with videos.
3. Removing tweets with the ‘humor’ label.
4. Label encode ‘real’ and ‘fake’ as 1 and 0 respectively.
5. Check for tweets with no images.
6. Load tweet images into an ‘images’ column.
7. Keep first image if a tweet has more than one image.
8. Resize all images to (224,224,3) shape.
9. Convert images to types ‘float32’.
10. Normalize images.

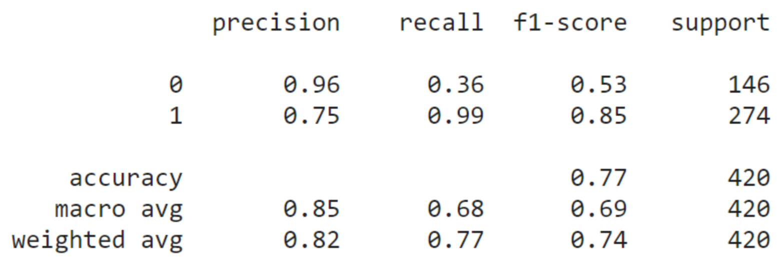
An 70/30 split was used for training. Images were resized to the same shape for consistency and for their use in the VGG16 model. Tweets were tokenized for the Word2Vec model.

## Experimental Results and Analysis

### Transfer Learning Multi Input Model With Word2Vec

The version of Word2Vec used was the one trained on Goggle News data. The tweet input model had 1 LSTM layer, 5 dense layers, and 1 dropout layer. The LSTM layer had a neuron count of 300. The dense layers had neuron counts of 300, 150, 64, 32, and 16. The activation function for all these layers was relu. The LSTM layer had a dropout of .5 and recurrent dropout of .2. The dropout layer used in between dense layers was set to .5. The image input CNN model had 2 layers. Both layers had kernel size set to 4 and pool size set to (2,2). The filter count for the Conv2D layers were 32 and 16. The activation function was relu. A dropout layer of .5 was used before the flattening layer. Then both layers were merged. After the merge was one dense layer with sigmoid activation and 10 neurons. Next was a dropout layer of .5. The final layer was the softmax output.

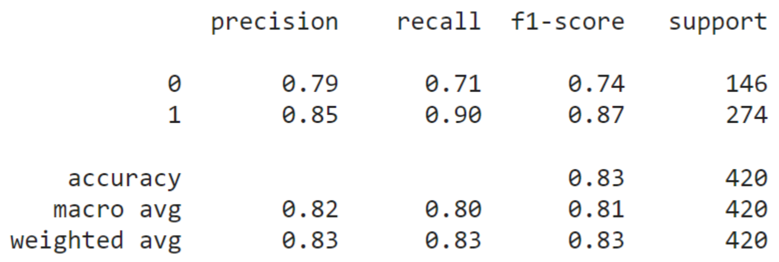
Best Model:



```
              precision    recall  f1-score   support

           0       0.96      0.36      0.53       146
           1       0.75      0.99      0.85       274

    accuracy                           0.77       420
   macro avg       0.85      0.68      0.69       420
weighted avg       0.82      0.77      0.74       420
```

## Transfer Learning Multi Input Model With Word2Vec and VGG16

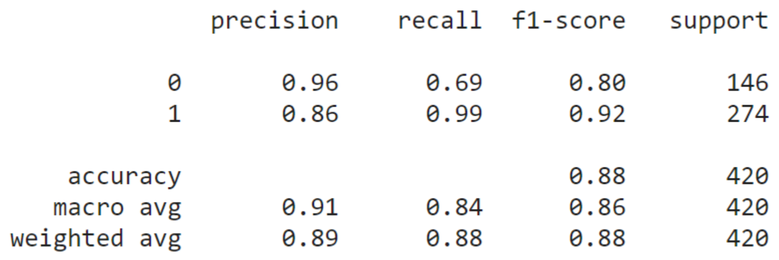
The same version of Word2Vec as before was used and the version of VGG16 used was the one trained on 'imagenet' data. The tweet input model had 1 LSTM layer, 3 dense layers, and 1 dropout layer. The LSTM layer had a neuron count of 300. The dense layers had neuron counts of 150, 32, and 16. The activation function for all these layers was relu. The LSTM layer had a dropout of .5. The dropout layer used in between dense layers was set to .5. The embedding of the VGG16 model was flattened and then a dropout layer of .5 was added. Then both models were merged. After the merge was one dense layer with sigmoid activation and 10 neurons. Next was a dropout layer of .5. The final layer was the softmax output.
Best Model:



```
              precision    recall  f1-score   support

           0       0.79      0.71      0.74       146
           1       0.85      0.90      0.87       274

    accuracy                           0.83       420
   macro avg       0.82      0.80      0.81       420
weighted avg       0.83      0.83      0.83       420
```

## Transfer Learning Multi Input Model With BERT

The version of BERT used was 'all-MiniLM-L6-v2'. BERT was implemented at a high-level. The tweet input model had 1 LSTM layer, 5 dense layers, and 1 dropout layer. The LSTM layer had a neuron count of 300. The dense layers had neuron counts of 300, 150, 64, 32, and 16. The activation function for all these layers was relu. The LSTM layer had a dropout of .5 and recurrent dropout of .2. The dropout layer used in between dense layers was set to .5. The image input CNN model had 3 layers. All layers had kernel size set to 4 and pool size set to (2,2). The filter count for the Conv2D layers were 64, 32, and 16. The activation function was relu. A dropout layer of .5 was used before the flattening layer. Then both layers were merged. After the merge was one dense layer with sigmoid activation and 10 neurons. Next was a dropout layer of .5. The final layer was the softmax output.
Best Model:

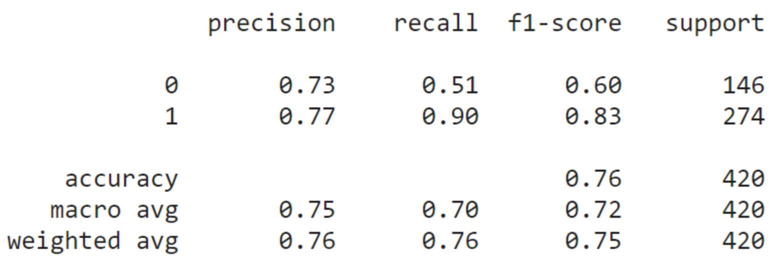


```
              precision    recall  f1-score   support

           0       0.96      0.69      0.80       146
           1       0.86      0.99      0.92       274

    accuracy                           0.88       420
   macro avg       0.91      0.84      0.86       420
weighted avg       0.89      0.88      0.88       420
```

## Transfer Learning Multi Input Model With BERT and VGG16

The same version of BERT and VGG16 were used as before. The tweet input model had 1 LSTM layer, 3 dense layers, and 1 dropout layer. The LSTM layer had a neuron count of 300. The dense layers had neuron counts of 150, 32, and 16. The activation function for all these layers was relu. The LSTM layer had a dropout of .5 and recurrent dropout of .2. The dropout layer used in between dense layers was set to .5. The embedding of the VGG16 model was flattened and then a dropout layer of .5 was added. Then both models were merged. After the merge was one dense layer with sigmoid activation and 10 neurons. Next was a dropout layer of .5. The final layer was the softmax output.

Best Model:



```
              precision    recall  f1-score   support

           0       0.73      0.51      0.60       146
           1       0.77      0.90      0.83       274

    accuracy                           0.76       420
   macro avg       0.75      0.70      0.72       420
weighted avg       0.76      0.76      0.75       420
```

## Task Division and Project Reflection

Work was done together as a team with online meetings. All code was written together in a pair programming style. The report was written together simultaneously. The first challenge was correctly preparing the data for the Word2Vec model. This difficulty was overcome by searching for guides and tutorials. Another challenge was correctly connecting the Word2Vec model with the rest of the model. This was solved by following guides and through trial and error. The final challenge was correctly implementing the VGG16 model with the rest of the model. This was overcome by looking at tutorials and trial and error. As a team we have learned cooperation and effective communication regarding pair programming. We also learned that transfer learning can be very difficult to implement when trying to correctly connect multiple models into one larger more complex model. There are many layers that must be intricately balanced, modified, and connected for the whole model to work correctly.

## Additional Features

The additional features that were implemented were VGG16 and BERT transfer learning. The Word2Vec combined with VGG16 model did outperform the Word2Vec model without VGG16 as we hypothesized. The BERT model with VGG16 did not outperform the BERT model without VGG16. The BERT model with VGG16 was the worst performing model while the BERT model without VGG16 was the best performing model. These results were puzzling. The BERT model with VGG16 does use different layer counts than the BERT model without VGG16. However, using the same layer counts still did not make the BERT model with VGG16 perform better. A lower-level implementation of BERT or possibly a different combination of layers that was not discovered in our testing may cause the BERT model with VGG16 to perform better.