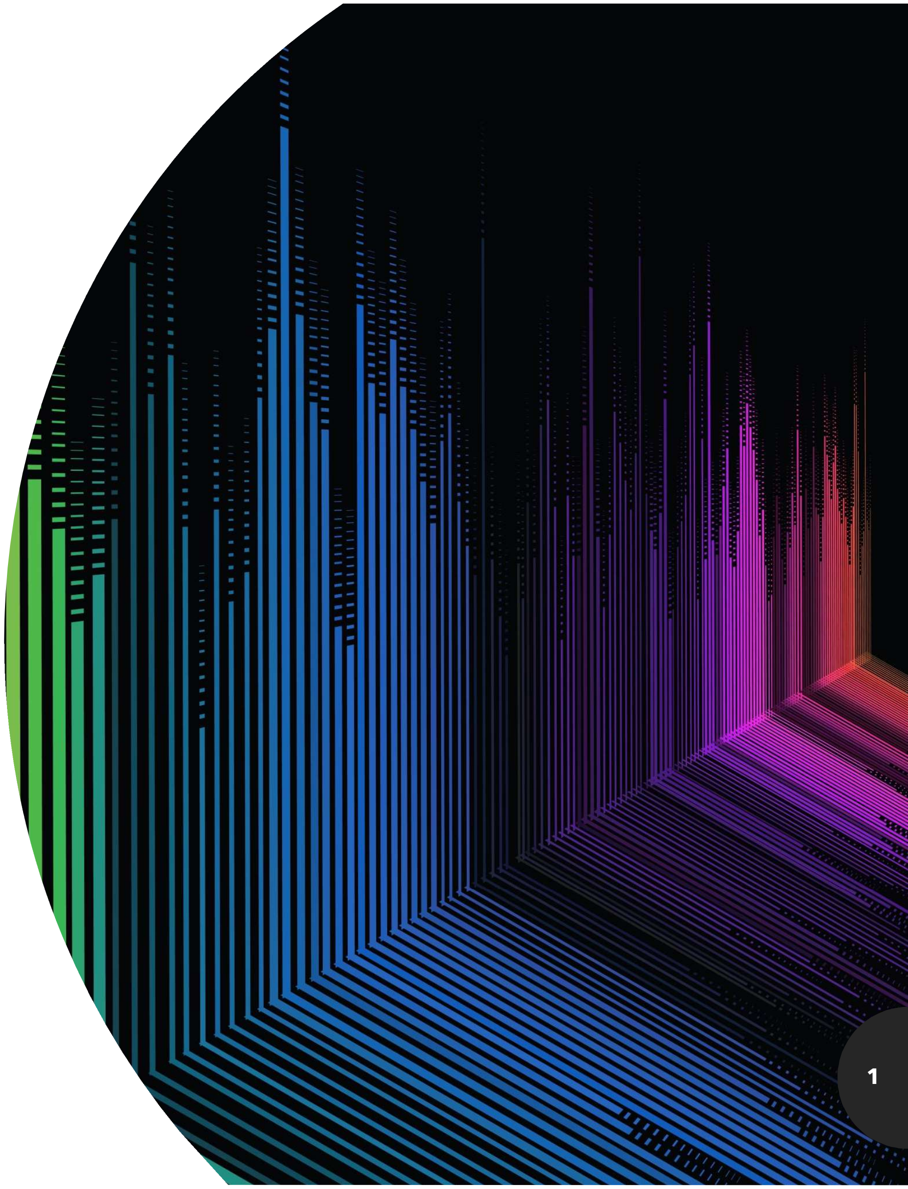
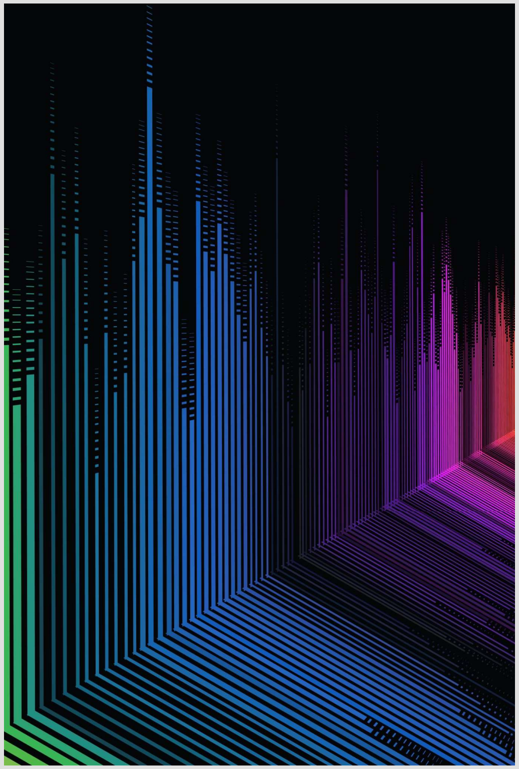
Sentiment Analysis using Deep Learning

DKM Project

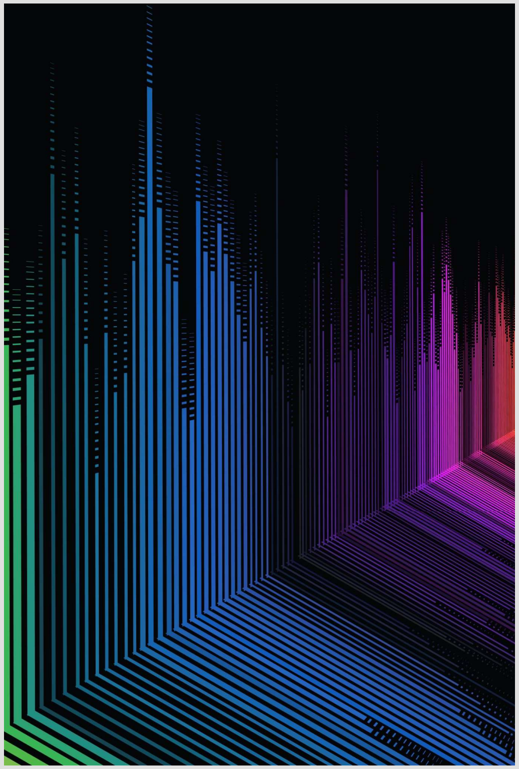
David Gonzalez
Matthew Langschwager
Kevin Lee



OBJECTIVE: Identify the Sentiment behind a provided statement ("tweet"), based on previously learned sentiment of happy / sad.



Labeling and Data Preparation



Labeling the Data: the Challenge

- *Provided with a dataset of 1864 tweets*

  – Goal: label each tweet as either "Happy" or "Sad" by hand

  –Many of the statements did not easily fall into either category, due to:

  ** Irreverence*
  ** Pop Culture Reliance*
  ** Incoherence*
  ** Advertising*

Labeling the Data: the Strategy

Goal: label each tweet as either “Happy” or “Sad” by hand

– Division of labor: each group member responsible for 2/3 of the set

– Any disagreements between two labelers settled by the 3rd member

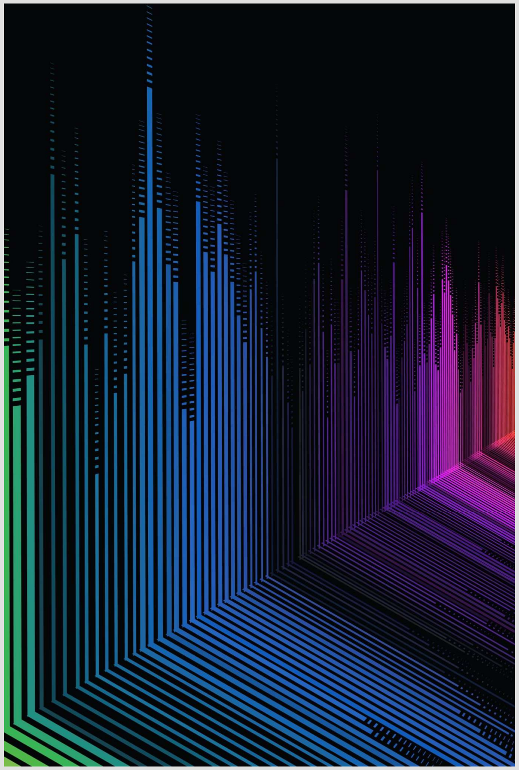
– Overlap in labor division ensures robustness:

No single group member sways the sentiment labeling

– End result: 753 “Happy” tweets, 1111 “Sad" tweets

Data Preprocessing

- Text convert to lower case
- Remove punctuation
- Remove non alphabetic
- Filter out Stopwords
- Filter out short tokens
- Stemming and Lemmatization



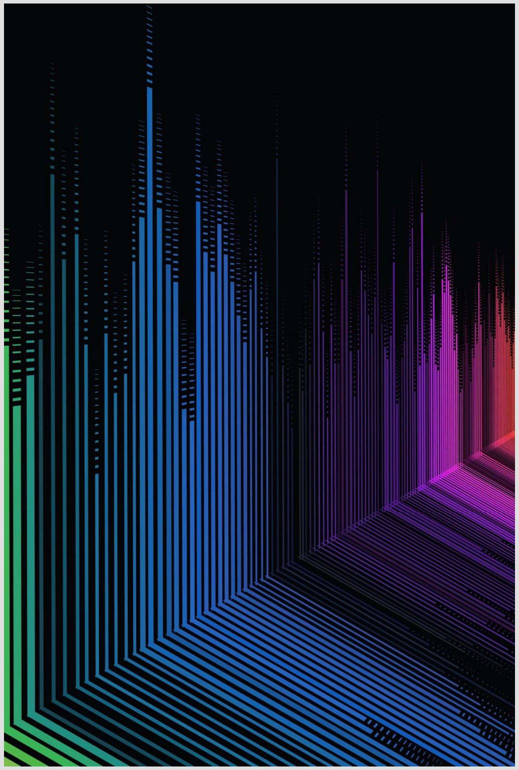
Data Preprocessing

screams in 25 different languages

↓

scream different language

Punctuation: !"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~

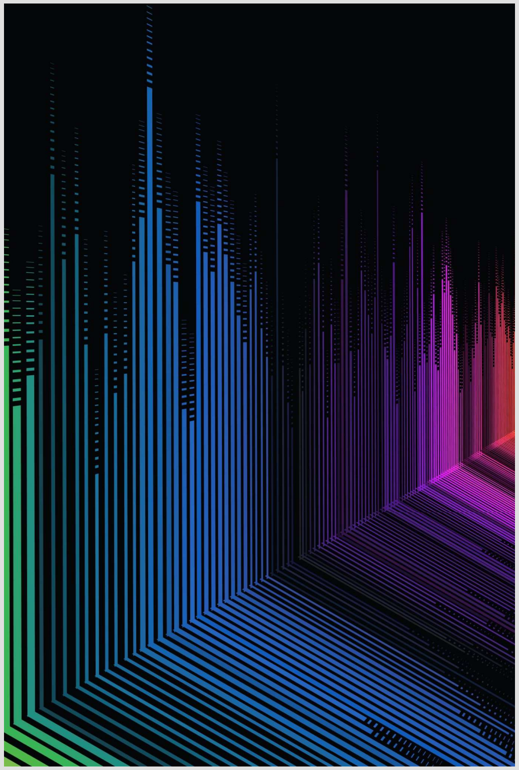


Lemmatization

Studies → Study

Studying → Study

Lemma

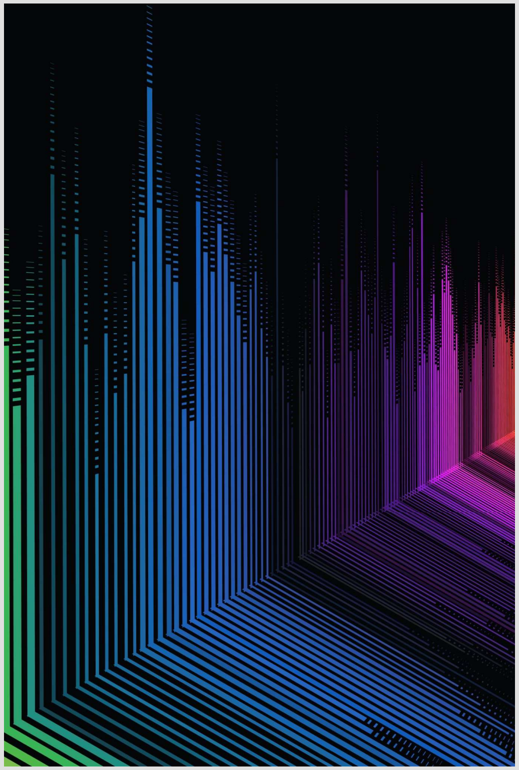


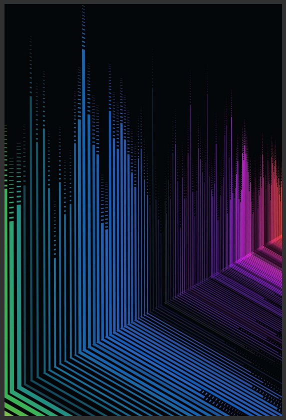
Stemming

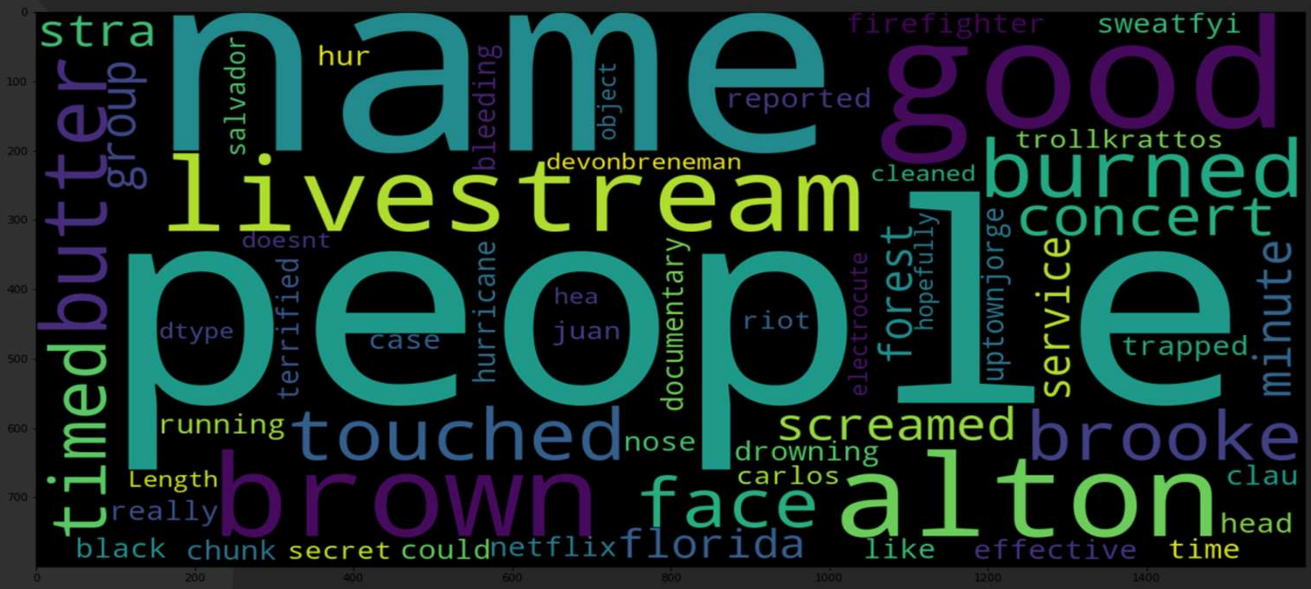
Studies → Studi

Studying → Study

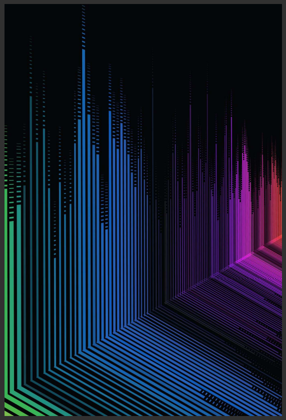
Stem

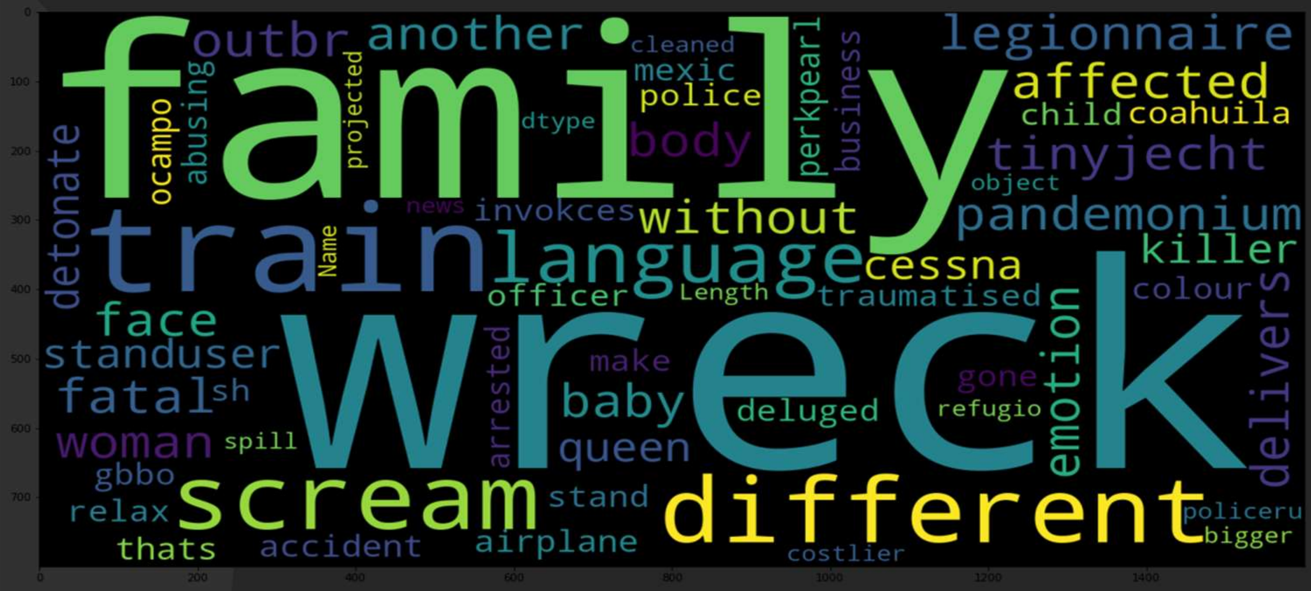






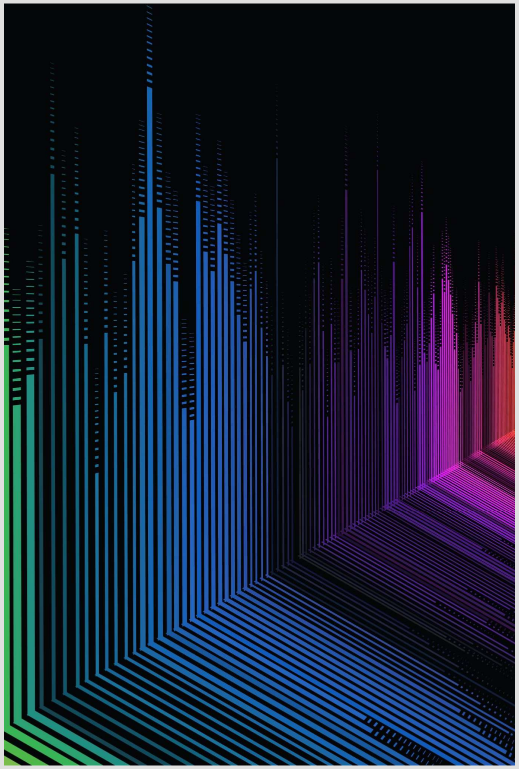
Word Cloud: Happy





Word Cloud: Sad

Modeling:
Sentiment Analysis Using TensorFlow



Sentiment Analysis using TensorFlow

Split the data into 3 groups: Training, Validation, Testing

- Two features: cleaned tweet, category encoding number

– Convert the arrays to TensorShapes

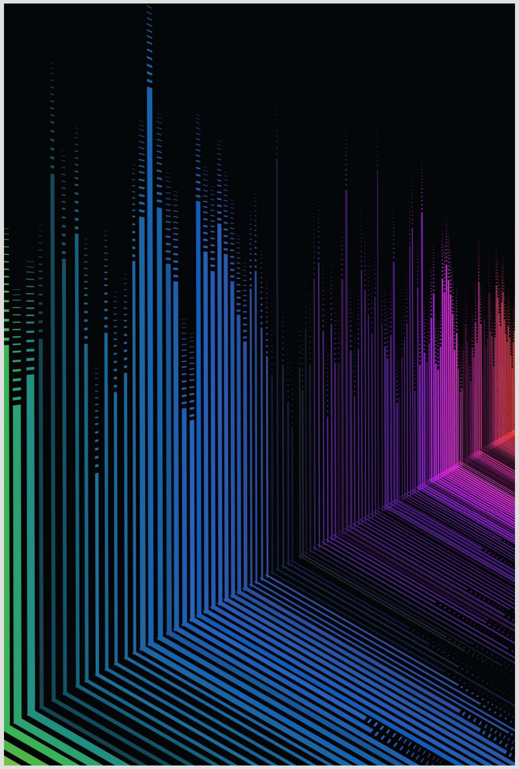
– Utilize available Embedding to determine the vectors of the training data

(https://tfhub.dev/google/tf2-preview/gnews-swivel-20dim/1)

– Separate the features into individual variables, then re-combine them

Modeling: Keras

Neural Network Architecture

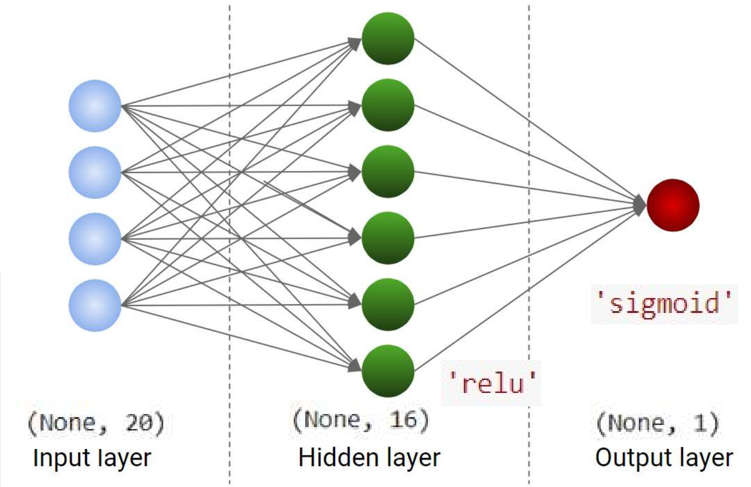


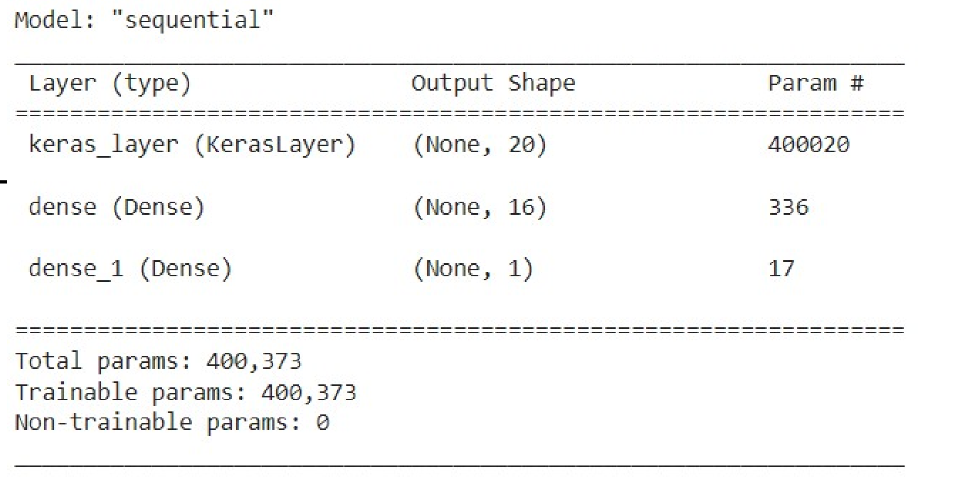
Neural Network Architecture

Determine the weights of the model through iterations

- *16 nodes in hidden layer*

- *Activations: ReLu and sigmoid*

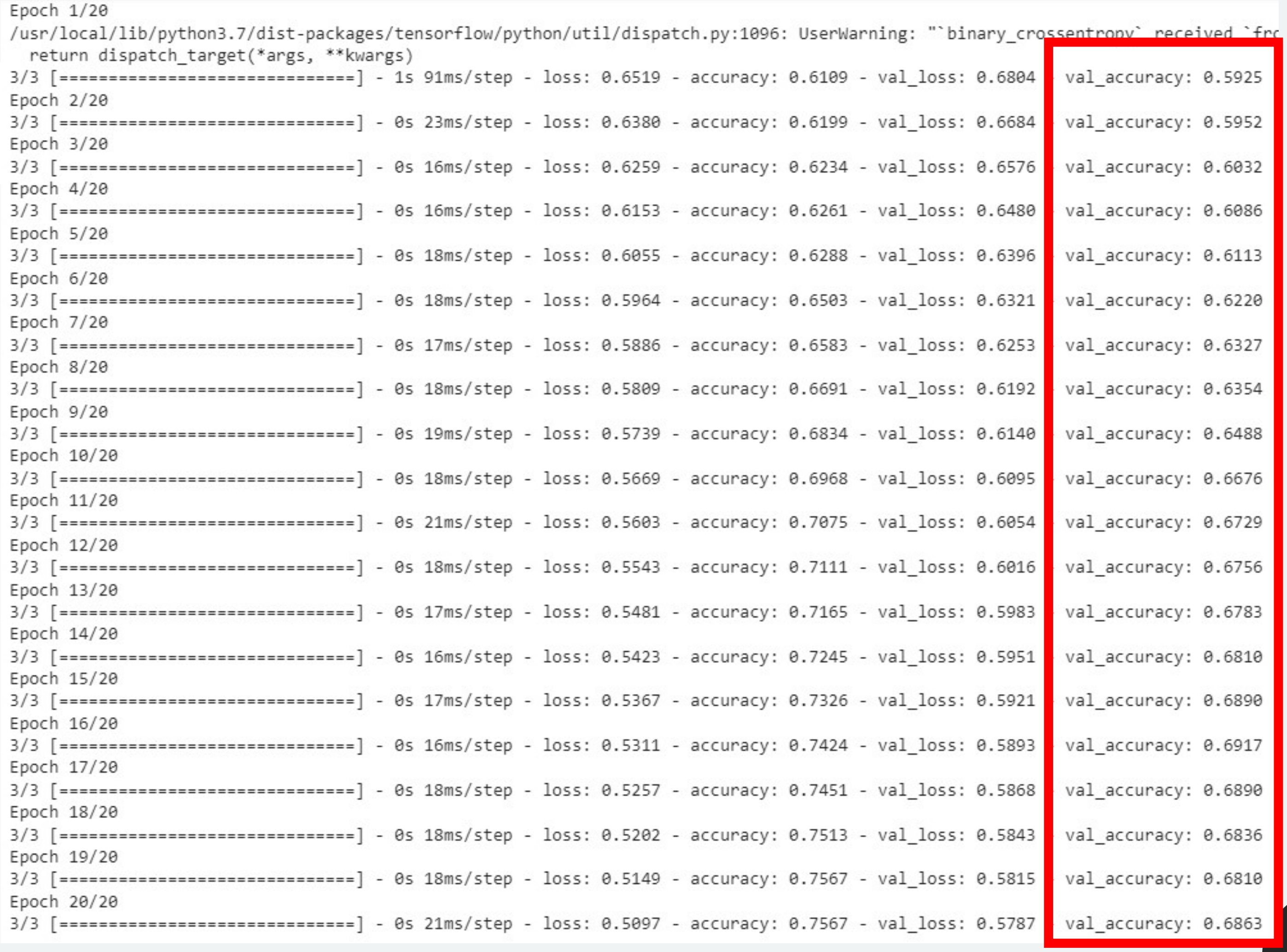




```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 20)                400020

 dense (Dense)               (None, 16)                336

 dense_1 (Dense)             (None, 1)                 17

=================================================================
Total params: 400,373
Trainable params: 400,373
Non-trainable params: 0
_________________________________________________________________
```

20 Epochs:

- The Accuracy has an initial state of 0.59.
- The iterations increase 0.004 in each step.
- Accuracy achieved: 0.68



```
Epoch 1/20
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1096: UserWarning: "`binary_crossentropy` received `frc
  return dispatch_target(*args, **kwargs)
3/3 [==============================] - 1s 91ms/step - loss: 0.6519 - accuracy: 0.6109 - val_loss: 0.6804 - val_accuracy: 0.5925
Epoch 2/20
3/3 [==============================] - 0s 23ms/step - loss: 0.6380 - accuracy: 0.6199 - val_loss: 0.6684 - val_accuracy: 0.5952
Epoch 3/20
3/3 [==============================] - 0s 16ms/step - loss: 0.6259 - accuracy: 0.6234 - val_loss: 0.6576 - val_accuracy: 0.6032
Epoch 4/20
3/3 [==============================] - 0s 16ms/step - loss: 0.6153 - accuracy: 0.6261 - val_loss: 0.6480 - val_accuracy: 0.6086
Epoch 5/20
3/3 [==============================] - 0s 18ms/step - loss: 0.6055 - accuracy: 0.6288 - val_loss: 0.6396 - val_accuracy: 0.6113
Epoch 6/20
3/3 [==============================] - 0s 18ms/step - loss: 0.5964 - accuracy: 0.6503 - val_loss: 0.6321 - val_accuracy: 0.6220
Epoch 7/20
3/3 [==============================] - 0s 17ms/step - loss: 0.5886 - accuracy: 0.6583 - val_loss: 0.6253 - val_accuracy: 0.6327
Epoch 8/20
3/3 [==============================] - 0s 18ms/step - loss: 0.5809 - accuracy: 0.6691 - val_loss: 0.6192 - val_accuracy: 0.6354
Epoch 9/20
3/3 [==============================] - 0s 19ms/step - loss: 0.5739 - accuracy: 0.6834 - val_loss: 0.6140 - val_accuracy: 0.6488
Epoch 10/20
3/3 [==============================] - 0s 18ms/step - loss: 0.5669 - accuracy: 0.6968 - val_loss: 0.6095 - val_accuracy: 0.6676
Epoch 11/20
3/3 [==============================] - 0s 21ms/step - loss: 0.5603 - accuracy: 0.7075 - val_loss: 0.6054 - val_accuracy: 0.6729
Epoch 12/20
3/3 [==============================] - 0s 18ms/step - loss: 0.5543 - accuracy: 0.7111 - val_loss: 0.6016 - val_accuracy: 0.6756
Epoch 13/20
3/3 [==============================] - 0s 17ms/step - loss: 0.5481 - accuracy: 0.7165 - val_loss: 0.5983 - val_accuracy: 0.6783
Epoch 14/20
3/3 [==============================] - 0s 16ms/step - loss: 0.5423 - accuracy: 0.7245 - val_loss: 0.5951 - val_accuracy: 0.6810
Epoch 15/20
3/3 [==============================] - 0s 17ms/step - loss: 0.5367 - accuracy: 0.7326 - val_loss: 0.5921 - val_accuracy: 0.6890
Epoch 16/20
3/3 [==============================] - 0s 16ms/step - loss: 0.5311 - accuracy: 0.7424 - val_loss: 0.5893 - val_accuracy: 0.6917
Epoch 17/20
3/3 [==============================] - 0s 18ms/step - loss: 0.5257 - accuracy: 0.7451 - val_loss: 0.5868 - val_accuracy: 0.6890
Epoch 18/20
3/3 [==============================] - 0s 18ms/step - loss: 0.5202 - accuracy: 0.7513 - val_loss: 0.5843 - val_accuracy: 0.6836
Epoch 19/20
3/3 [==============================] - 0s 18ms/step - loss: 0.5149 - accuracy: 0.7567 - val_loss: 0.5815 - val_accuracy: 0.6810
Epoch 20/20
3/3 [==============================] - 0s 21ms/step - loss: 0.5097 - accuracy: 0.7567 - val_loss: 0.5787 - val_accuracy: 0.6863
```

```
Write a sentence: Free icecreamtoday
array = 1 (Happy!), confidence:  55 %
```

```
Write a sentence: Good taste!, like it!
array = 1 (Happy!), confidence:  63 %
```

```
Write a sentence: Today is football day!
array = 1 (Happy!), confidence:  58 %
```

```
Write a sentence: I'm having good time with my family
array = 1 (Happy!), confidence:  69 %
```

```
Write a sentence: I don't like brocolli
array = 0 (Sad!), confidence:  90 %
```

```
Write a sentence: I hate you
array = 0 (Sad!), confidence:  99 %
```

```
Write a sentence: I feel like my head is going to explode
array = 0 (Sad!), confidence:  99 %
```

Conclusions

- The size and quality of the dataset affected the modeling

*Too many “middle ground” statements

- Model still has a good level of success

- More epochs increase validation accuracy

- Future work: finding larger datasets with definitive “Happy” and “Sad” sentiments for better training of the model

Thank you for your attention!

Any questions?