TOXIC COMMENT CLASSIFICATION: KAGGLE NLP COMPETITION

Hector Cadeaux

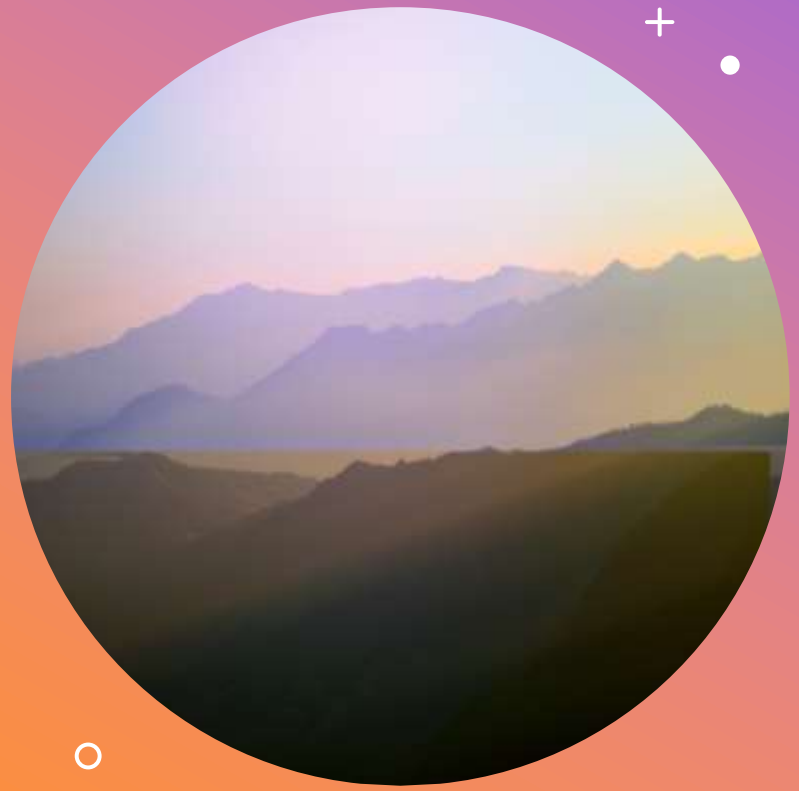
Capstone 3

Divergence Academy

AGENDA

Elevator Pitch

- Introduction
- Algorithm Review
- Embedding
- Bake-off Summary
- Conclusion



"THERE'S A STATISTICAL THEORY THAT IF YOU GAVE A MILLION MONKEYS TYPEWRITERS AND SET THEM TO WORK, THEY'D EVENTUALLY COME UP WITH THE COMPLETE WORKS OF SHAKESPEARE. THANKS TO THE INTERNET, WE NOW KNOW THIS ISN'T TRUE."

~ Ian Hart

Introduction

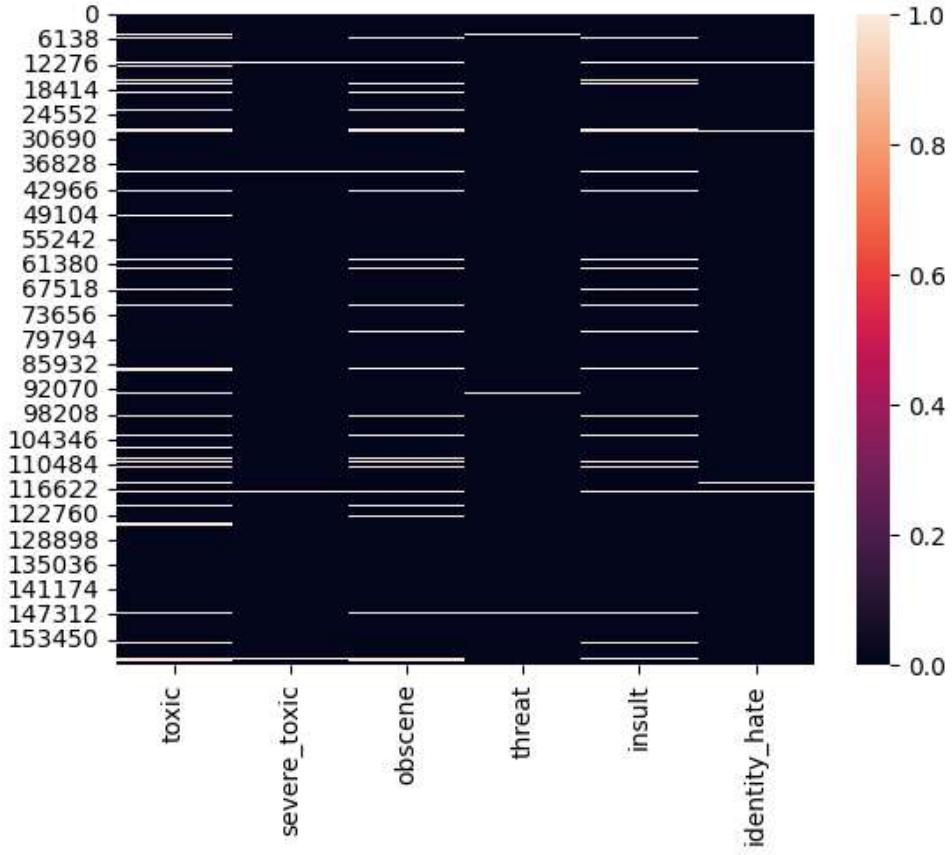
- Beauty of the Internet
- Lots of information
  - Class
  - Wikipedia
  - Help on complex subjects
- Bad actors known to exist
- Catch that misbehavior

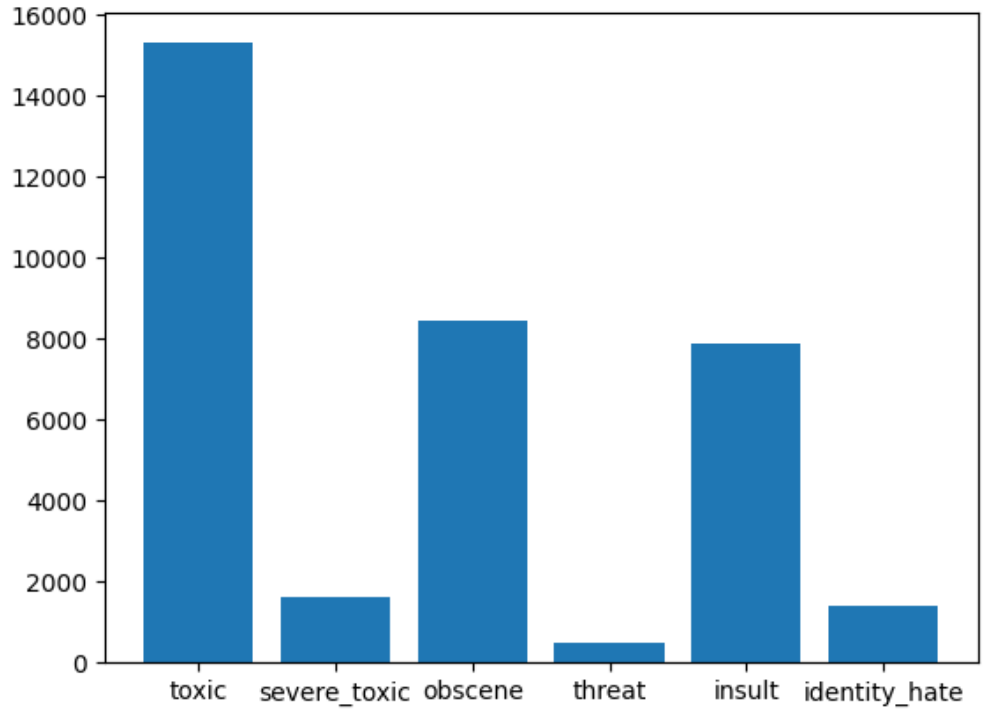
DATA EXPLORATION

What does bad behavior look like?

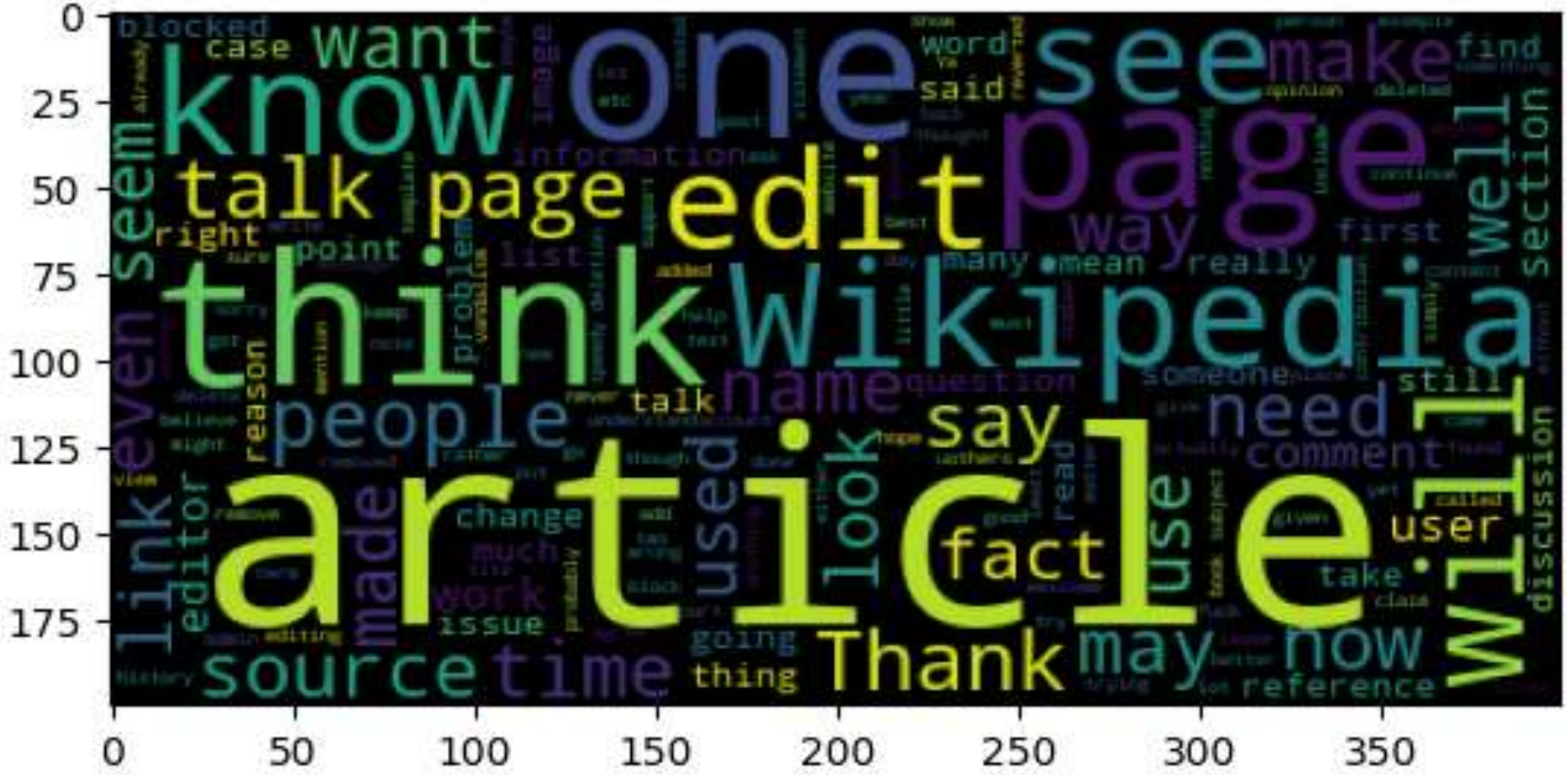
Warning: Triggering Language

Distribution





Wordclouds - Benign



Wordclouds

Toxic

Obscence

Severe Toxic

Wordclouds

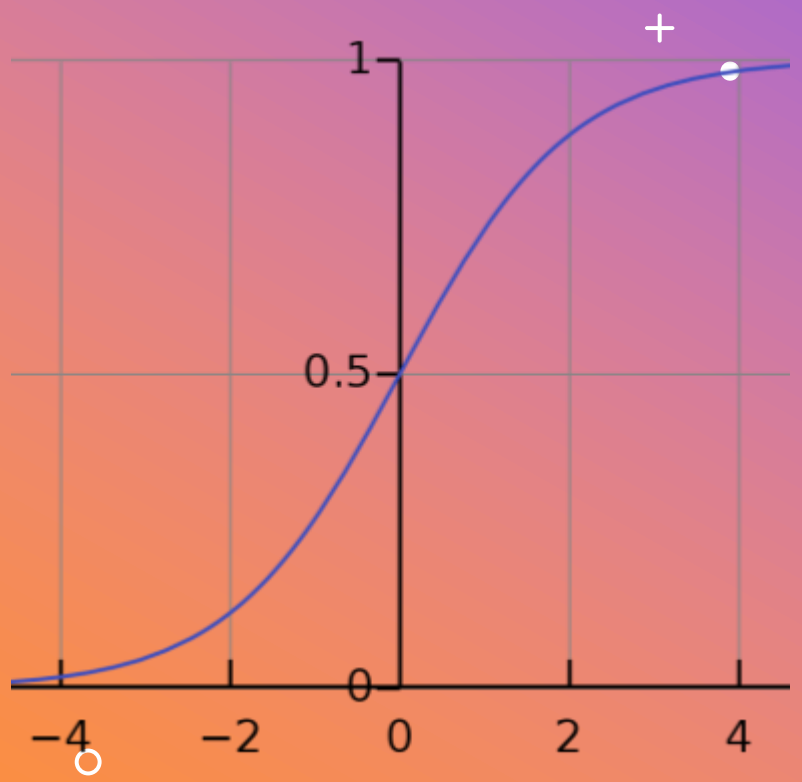
Threat

Identity Hate

Insult

TOXIC COMMENT CLASSIFICATION

ALGORITHMS



Scikit-learn: Random Forest

Convolutional Neural Networks (CNN)

Recurrent Neural Networks (RNN)

Sentence Embeddings with Neural Nets

Scikit-Learn

- Many algorithms
- Supervised, Unsupervised and Deep Learning
- Wide range of uses in NLP
  - Text classification
  - Sentiment Analysis
  - Text summarization

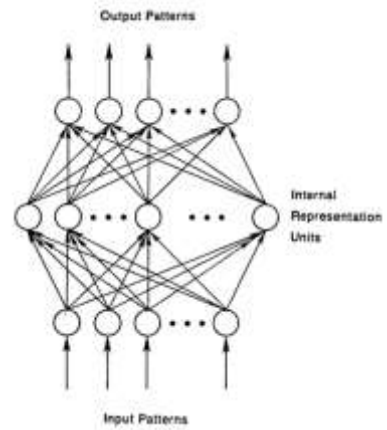
Random forest can be programmed to fully use resources

CNNS

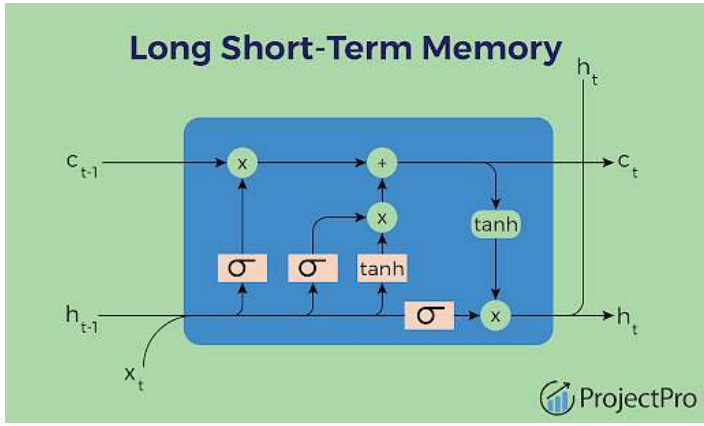
- Used primarily for picture data
- Use a sliding window to decrease dimensions and extract meaning from the window
- “Convolutions” are used to decrease the dimensions
- Multilayered
- Can be fast and efficient
- Difficult to train

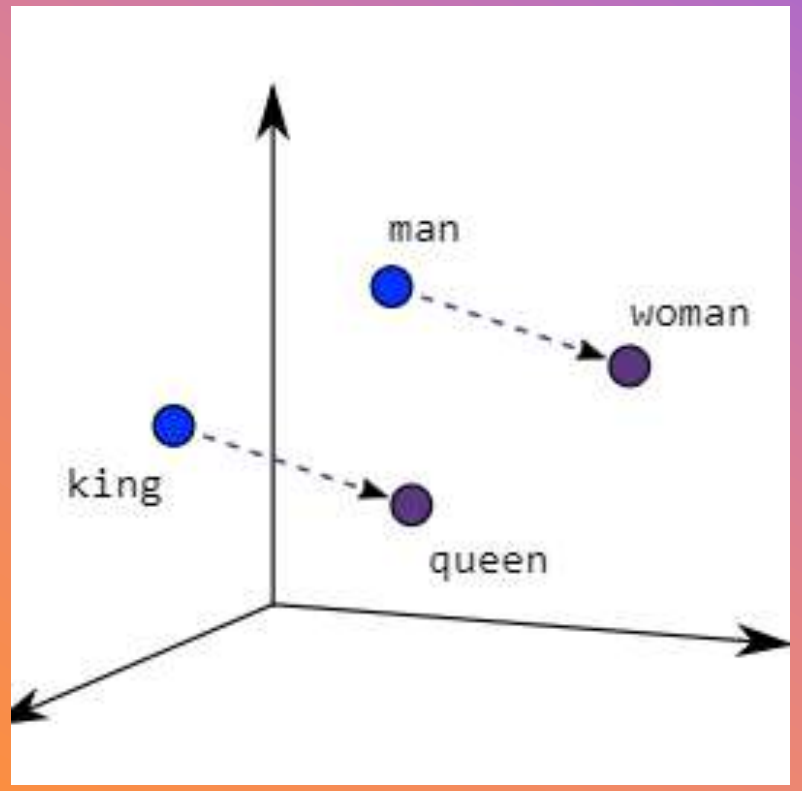
RNNs and LSTM

- RNN models relationships between words
  - Handle sequential data well
- Uses a representation of historical data
  - Predicts the next word in the sequence



- LSTM (Long Short-Term Memory)
  - Considered a strong variation of RNN
  - 3 gates (input, output, and forget)
  - Gates controlled by learned weights
  - Keep track of the history of inputs





SENTENCE EMBEDDINGS

Takes vector representations of sentences

Outputs a vector for each word

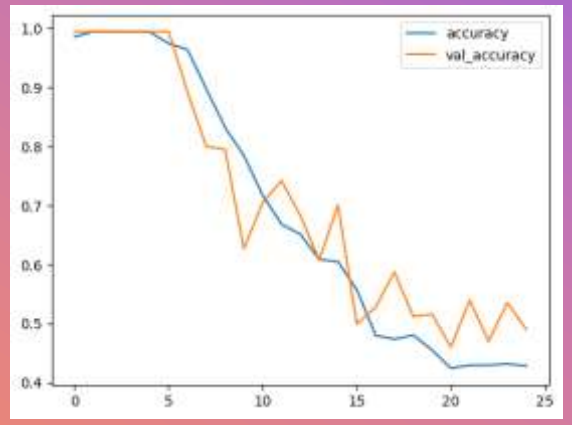
Averages word vectors for each sentence and creates a new vector

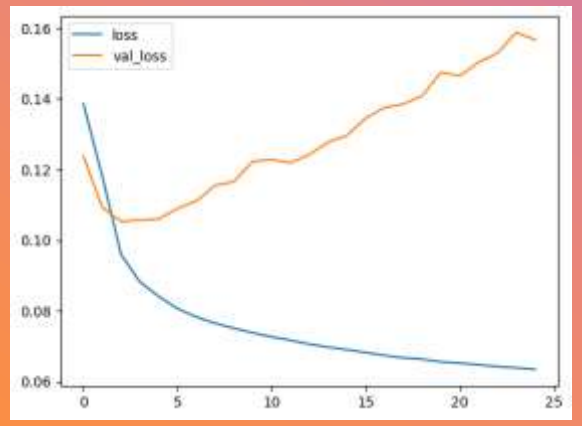
I used “Bidirectional Encoder Representations from Transformers” (BERT)

Transformers are its own kind of neural network

*A sample of the dataset was run

ACCURACY VS LOSS





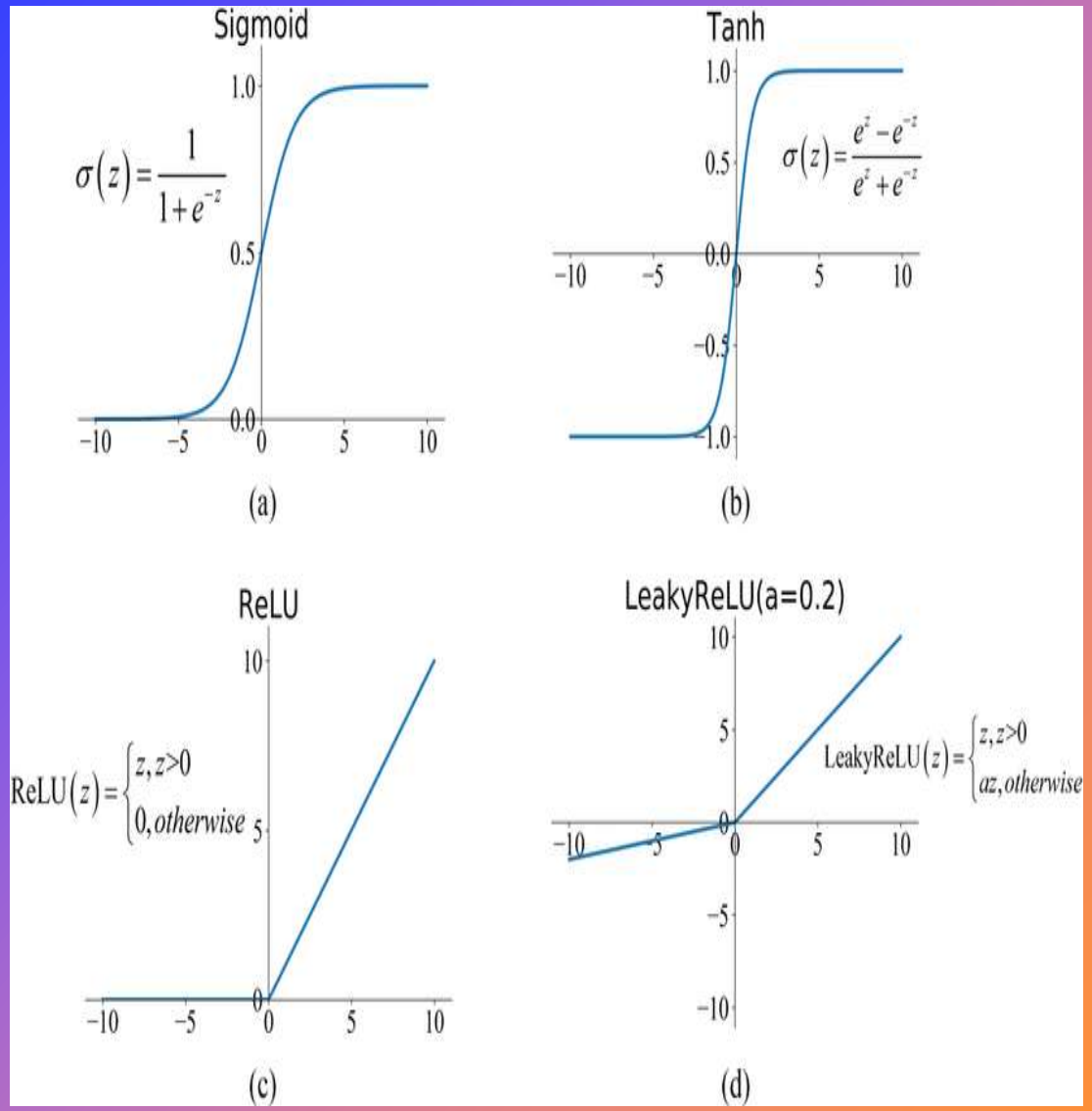
Loss is a measure of how close to the truth

Loss should be minimized

Accuracy is how exact the predictions are

Accuracy should be maximized

Decreases in loss may cause a decrease in accuracy



Activation Functions

Tanh is the default

Relu was the most used

Sigmoid is just Logistic Regression

BAKE-OFF

What you came to see!

SCIKIT-LEARN

	Tfidf V		Count V	
	Train	Valid	Train	Valid
Accuracy	0.9987	0.9015	0.9987	0.9022
Precision	0.9992	0.8218	0.9990	0.8307
Recall	0.9900	0.2141	0.9902	0.2114
F1	0.9947	0.3382	0.9946	0.3356
Loss	0.2152	0.9951	0.2139	0.9161

CNN Descriptions

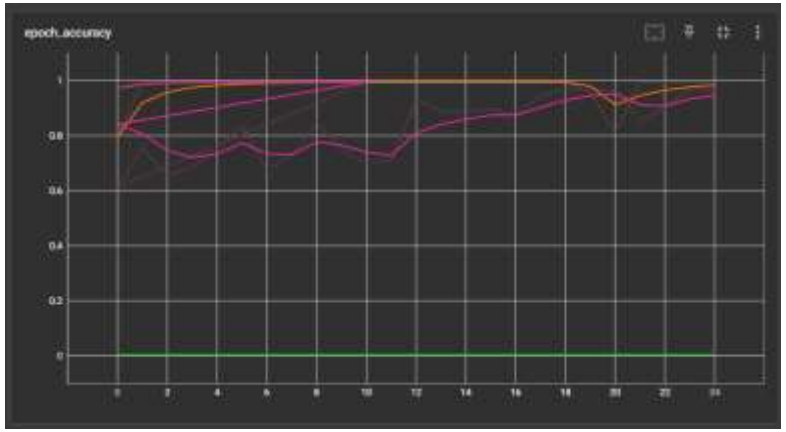
Model Name	Depth	Neuron Multiplier	Adam	Cross Entropy	Layers	Activation
CNN_CCE_02_q	60	32	0.001	Categorical	3	Relu
CNN_CCE_03_03_s	60	64	0.001	Categorical	4	Relu
CNN_CCE_00_f	60	32	0.001	Categorical	4	Sigmoid
CNN_CCE_09_g	60	32	0.0001	Categorical	4	Sigmoid
CNN_CCE_08_h	60	32	0.0005	Categorical	4	Relu-3
CNN_CCE_07_e	60	32	0.0005	Categorical	4	Relu-3
CNN_CCE_06_k	60	32	0.00005	Categorical	4	Relu-2
CNN_CCE_SIG_a1_e	60	32	SDG= 0.00005	Categorical	4	Relu-3

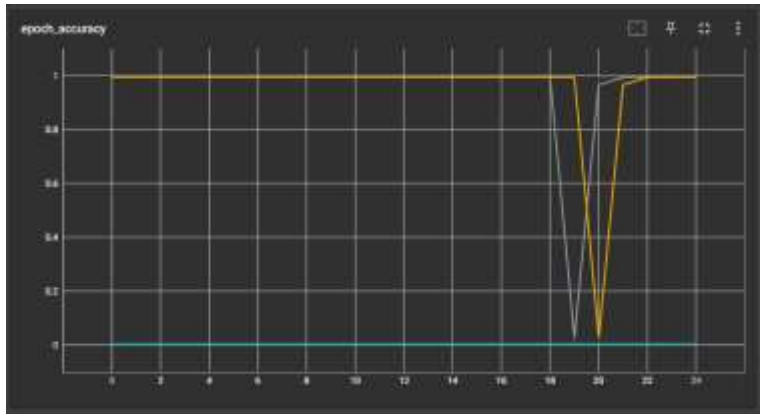
CNNs

Train

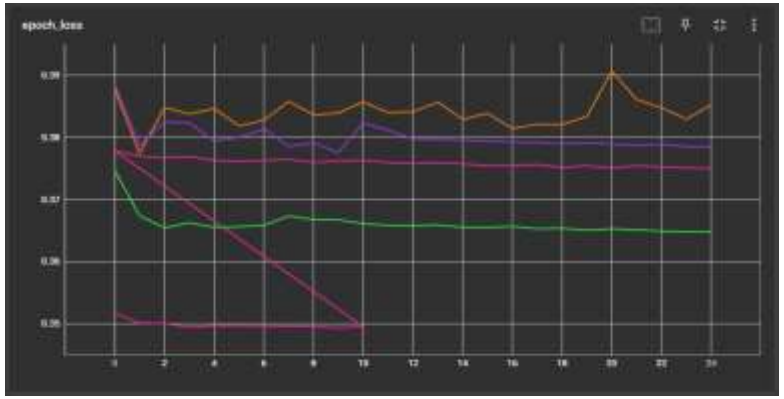
Validation

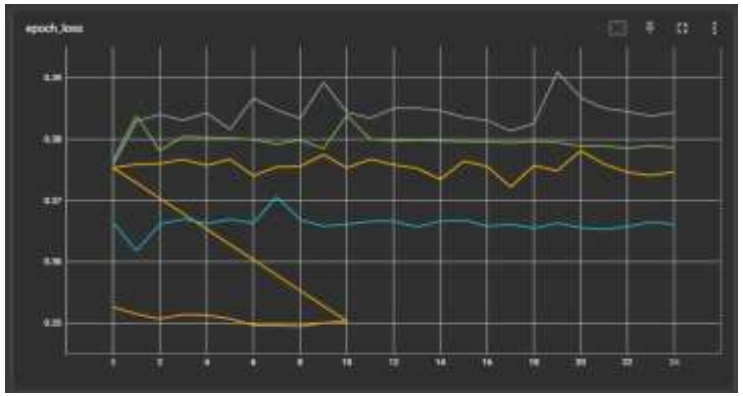
Accuracy





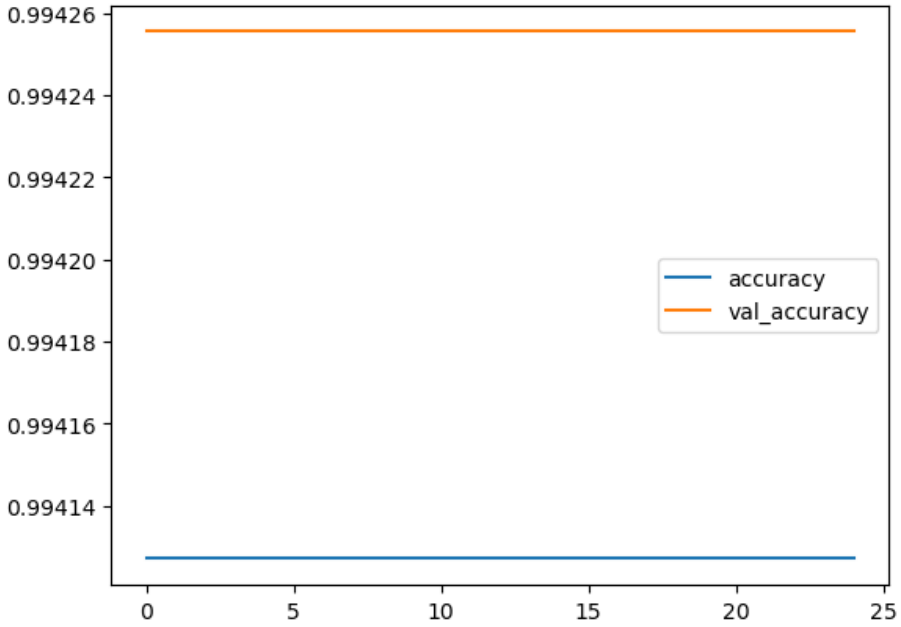
Loss

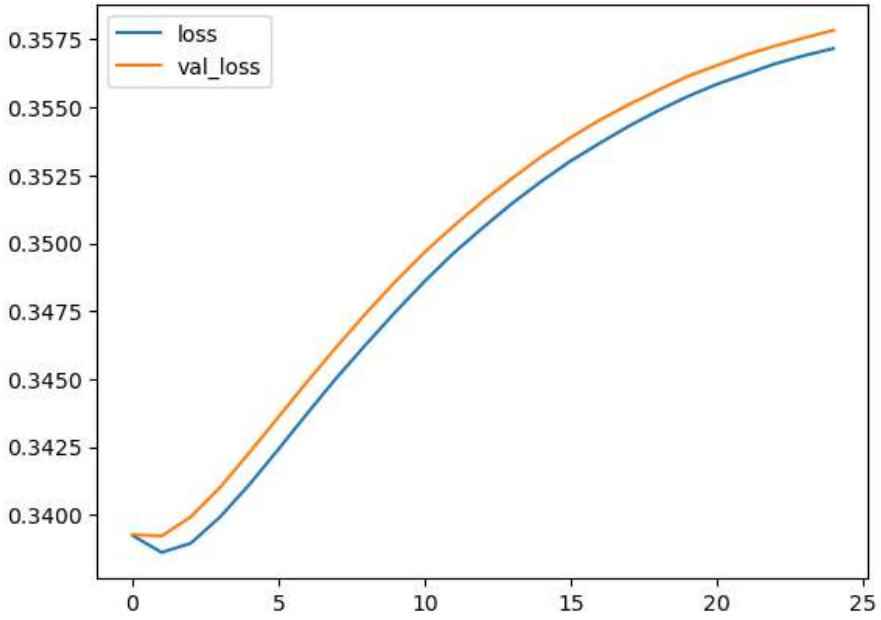




Best CNN

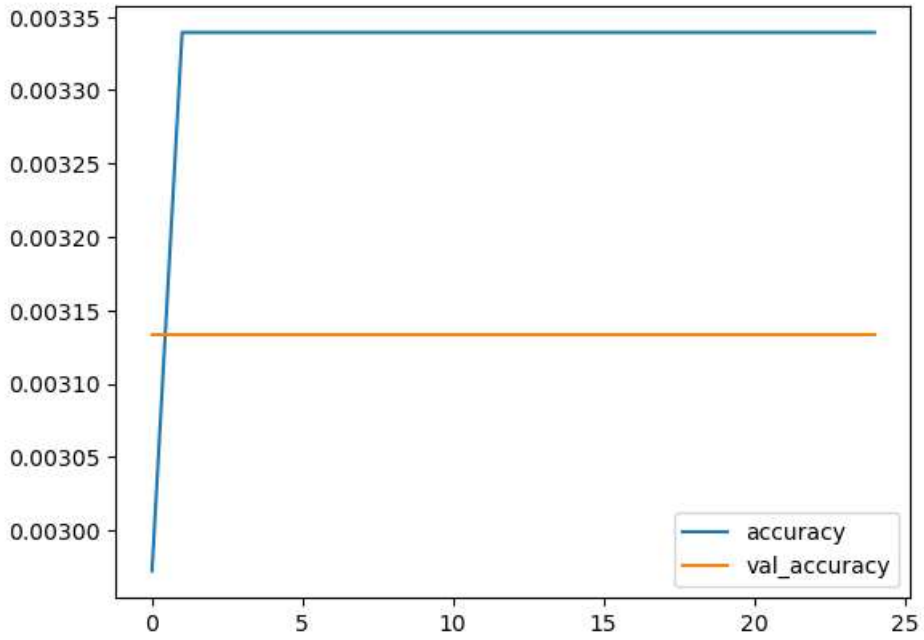
- CNN_CCE_SIG_a1_e: 60-depth, 32 neuron multiplier, SDG= 0.00005, Categorical, 4 Layers, Relu-3

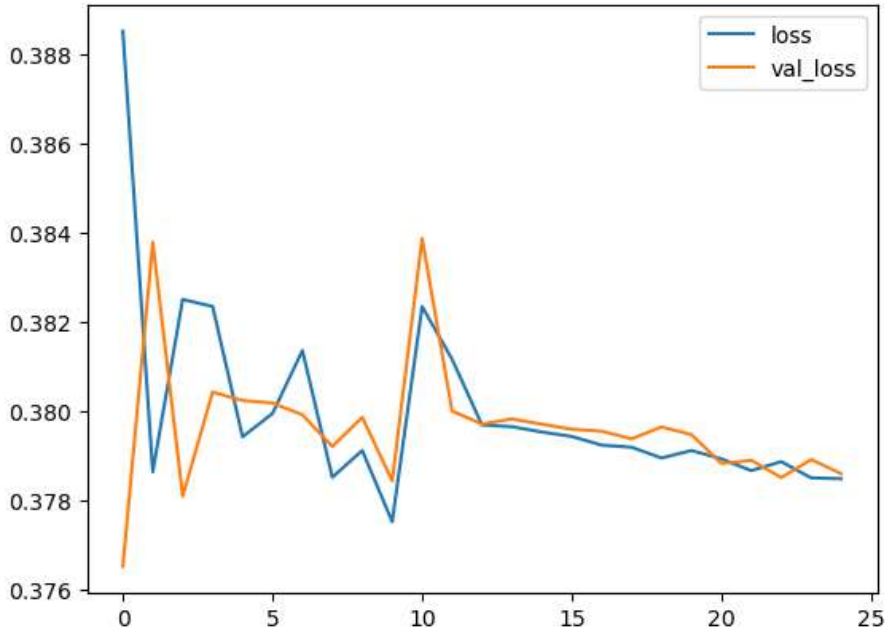




Worst CNN

- CNN_CCE_07_e: 60 depth, 32 neuron multiplier, Adam=0.00005, Categorical, 4 layers, Relu-3





RNN- Descriptions

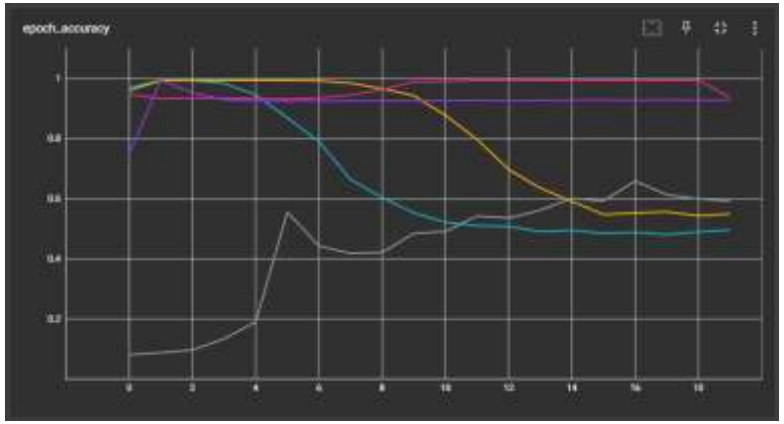
Model Name	Depth	Neurons	Adam	Cross Entropy
RNN_00_r	60	35	0.001	Binary
RNN_02_p	60	32	0.001	Binary
RNN_03_q	30	32	0.001	Binary
RNN_05_s	60	35	0.001	Categorical
RNN_06_t	30	16	0.001	Categorical
RNN_07_w	30	32	0.0001	Categorical

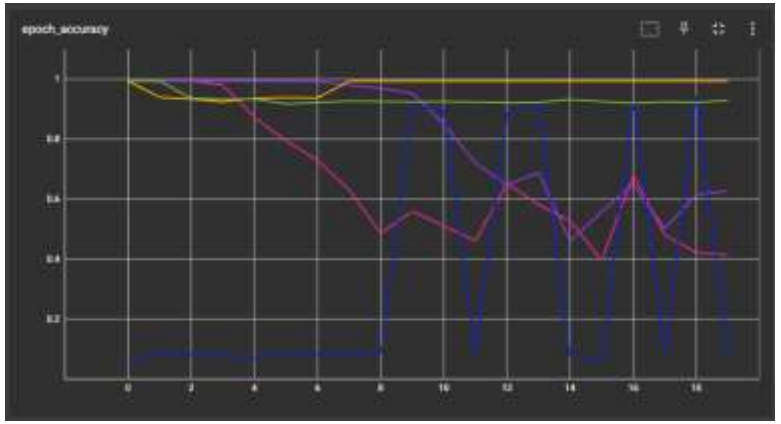
RNNs

Train

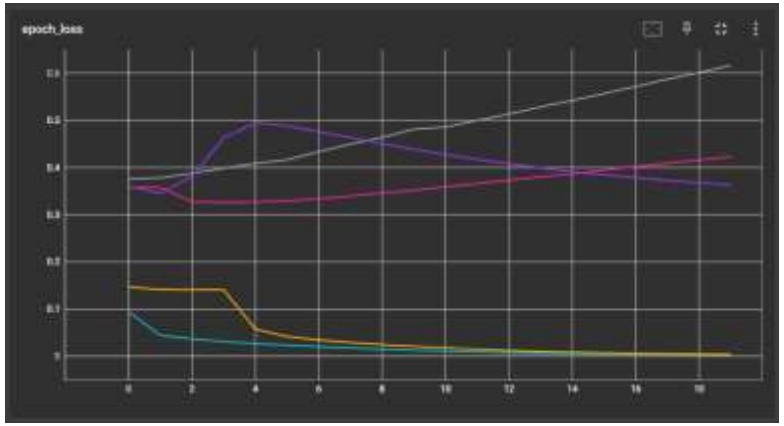
Validation

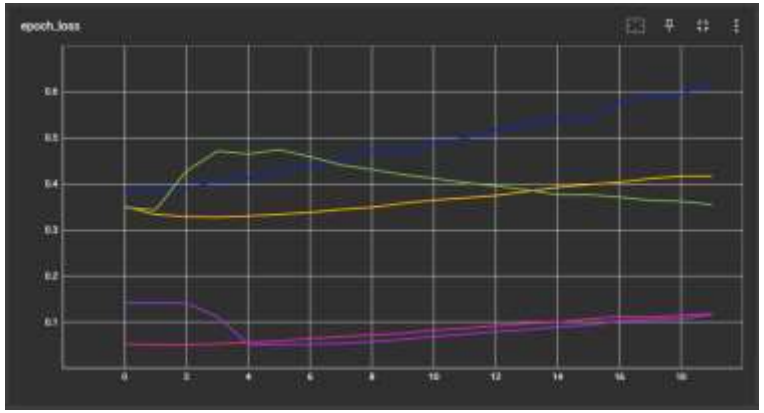
Accuracy





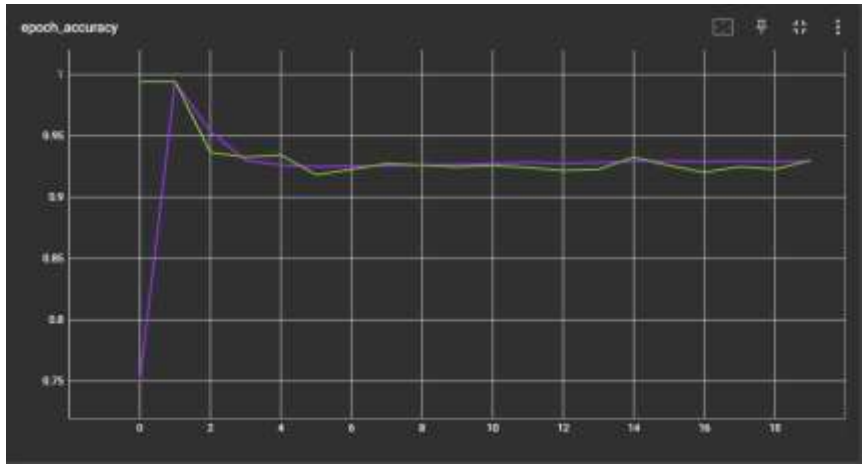
Loss

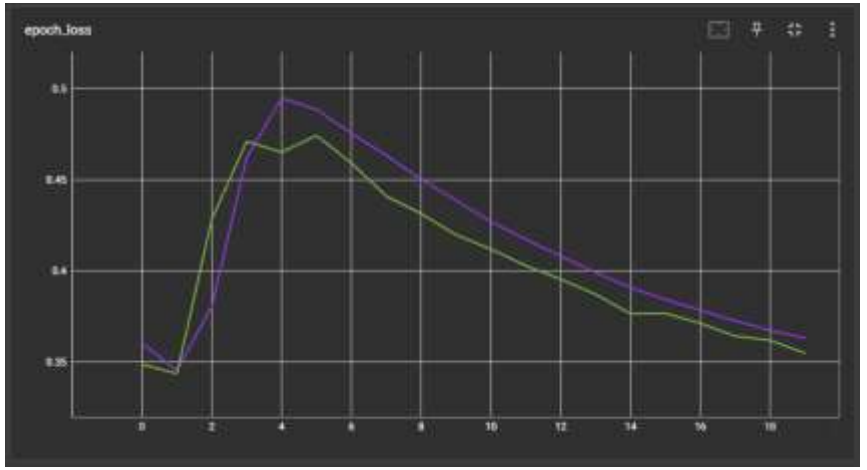


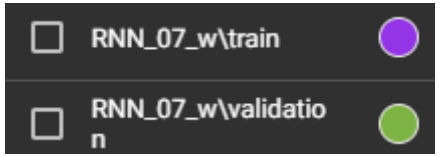


Best RNN

- RNN_07_w: 30 depth, 32 neurons, Adam = 0.0001, Categorical

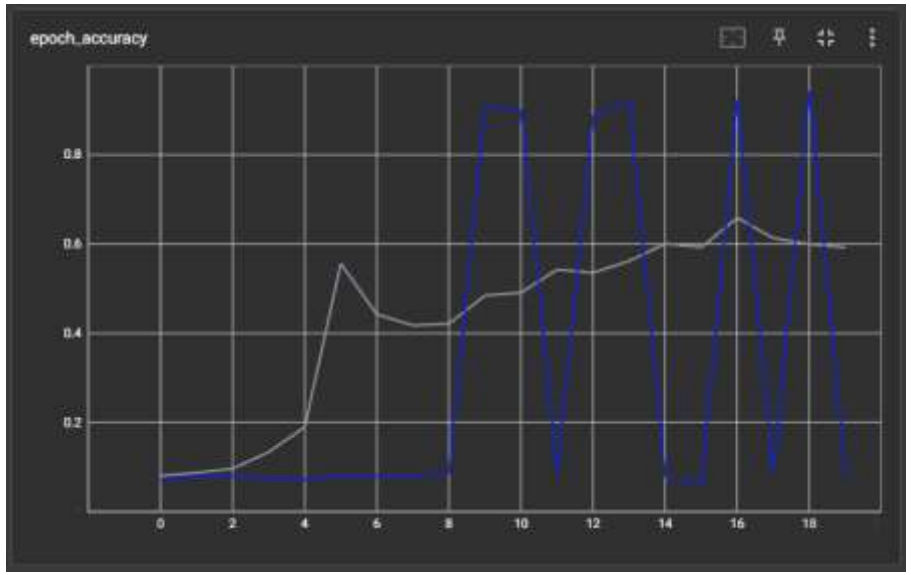


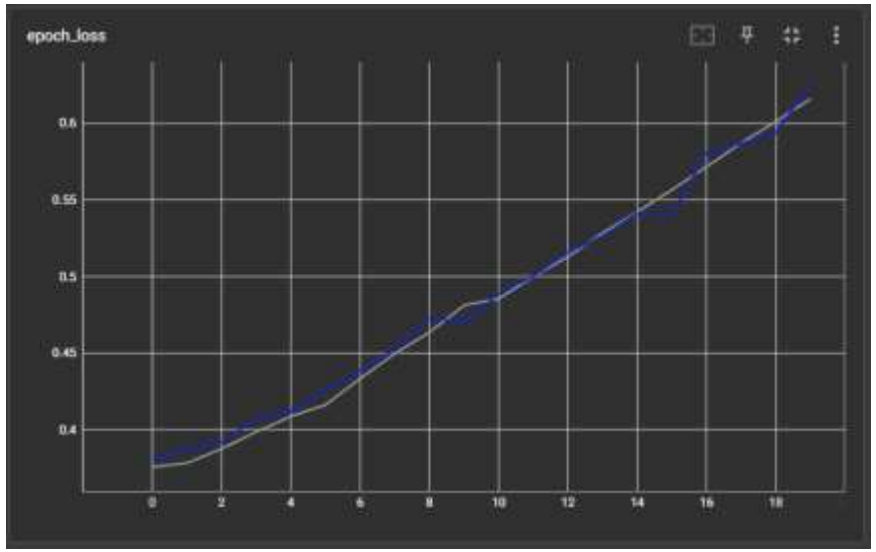


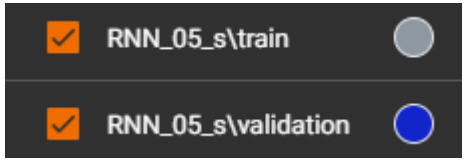


Worst RNN

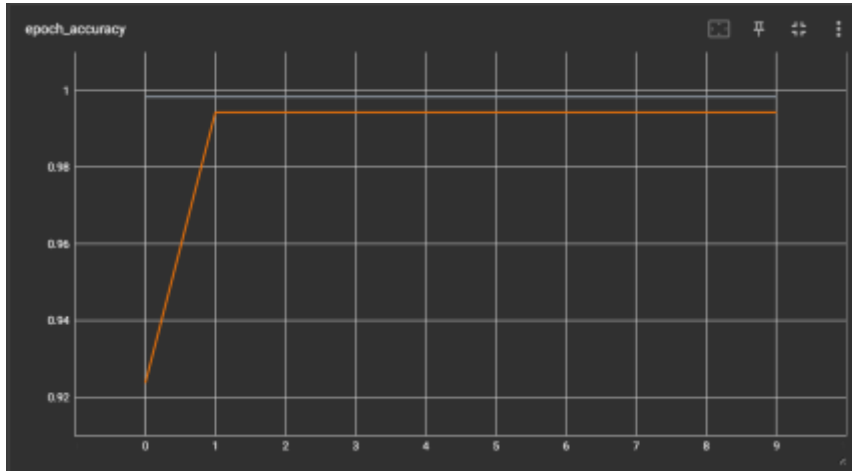
- RNN_05_s: 60 depth, 35 neurons, Adam = 0.001, Categorical

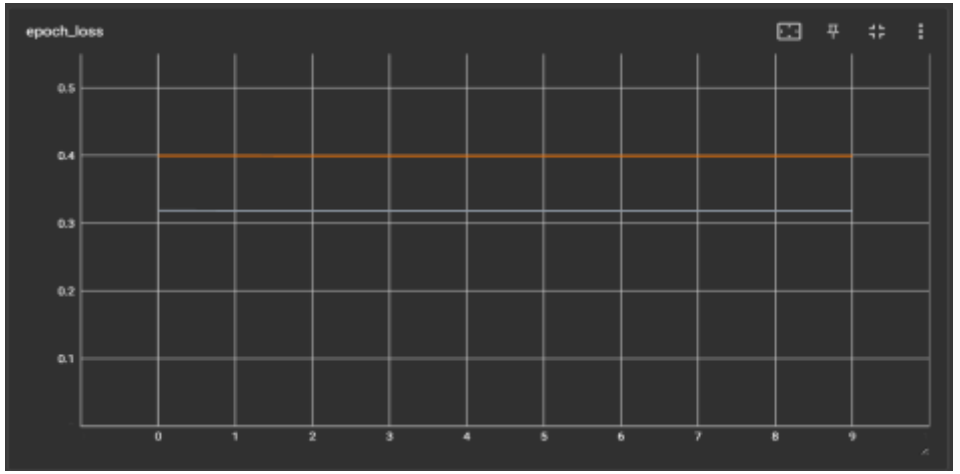




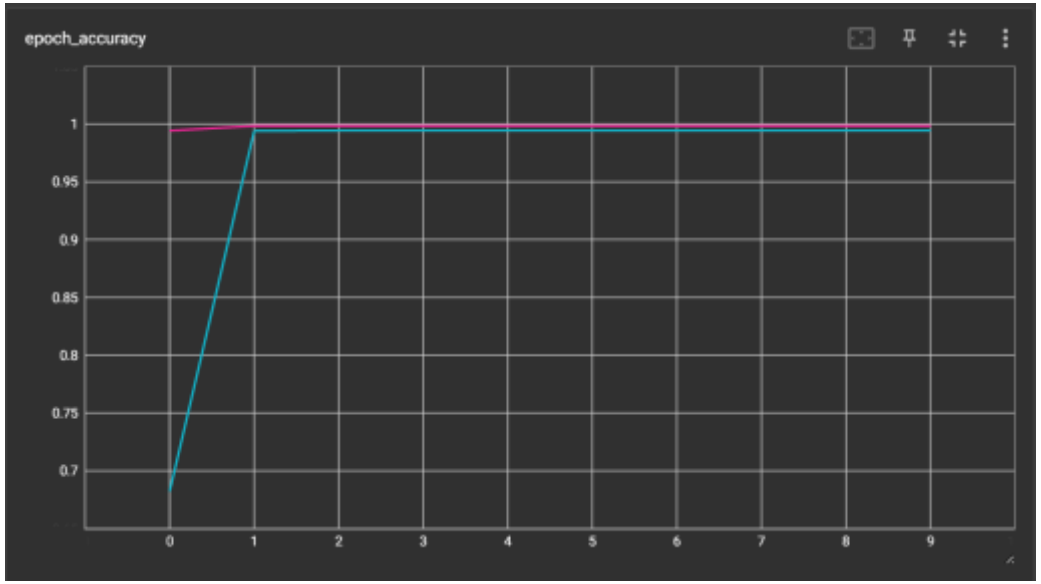


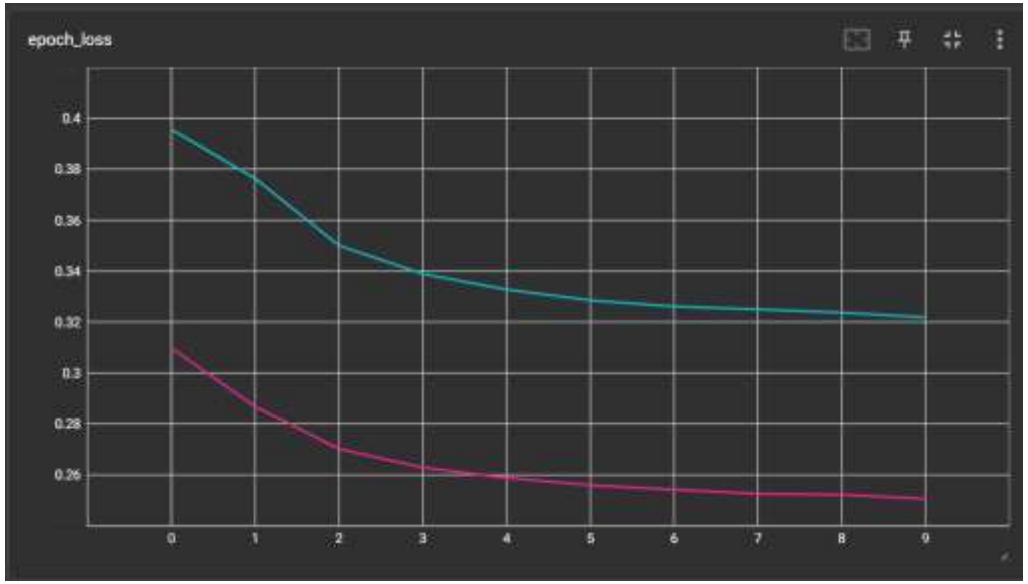
Sentence Embeddings: LSTM





Sentence Embeddings: CNN





Sentence Embeddings

- Would have liked to do the entire dataset
- It worked well with the sample of 4000
- Lower loss with CNN
- Faster convergence with LSTM
- Might be overfit

Lessons Learned

- A lower learning rate and lower epochs can produce better results
- All methods were valid, with good results from each
- The implementations were very time-consuming
- It’s best to understand the methodologies you are trying to implement

Further Considerations

Dataset Imbalance

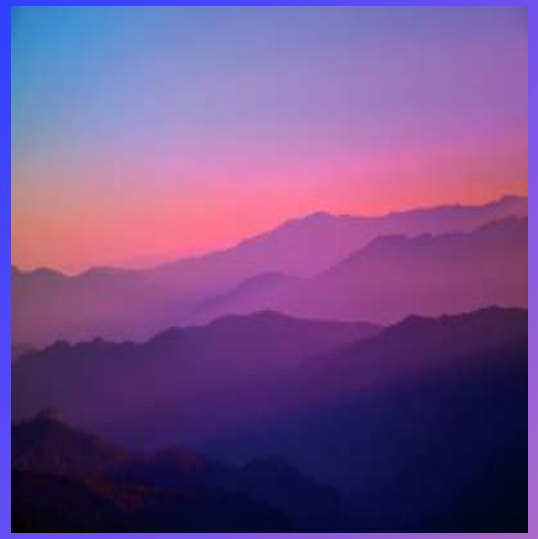
- I would have liked to have run SMOTE
- This would have made it easier for the dataset to speak to the algorithms

Algorithms

- I would have liked to do more algorithms
- More variety
- A more traditional approach
- Faster data cleaning
- More consistent

Libraries

- Knowledge and preparation of a system of Libraries
- Greater compatibility with more packages
- More Speed
- Integration with Cloud-based systems



Summary

It was fun to play with different algorithms. Learning the power of RNNs and CNNs while getting acquainted with new skills was eye-opening. The experience of installing and running the new software and getting all the packages and libraries to work together inspired me to push harder. The experience was harrowing but worth every second of strife and processor time.

SPECIAL THANKS

Drew Minkin

ChatGPT

Bard

Divergence Academy and all the instructors

Kaggle

Udemy

Doc

Spotify

THANK YOU

QUESTIONS/COMMENTS

Hector Cadeaux

hcadeaux@outlook.com

https://www.linkedin.com/in/hector-cadeaux/