Machine Learning Algorithm Predicts The Popularity Of A Meme

Robin Leman

MAIS 202 - Winter 2021 - Deliverable 3

Abstract

In this experiment, a machine learning algorithm tries to predict ”how funny” a meme is. By training the algorithm on a dataset of memes from Reddit, the machine learns the popularity of a meme based on its number of upvotes. It is then able to predict how funny an original meme is through a Web user interface. We obtained a model with a 95% training accuracy and 30% validation accuracy, showing an overfitting.

1 Model modifications and Interim results

A few modifications were made to the model in order to increase its performance. First, we shuffled the data to prevent the ordered dataset that we had in our preliminary results. The model was also deepened by adding more convutional layers, allowing better training on more characteristics. We increased the split of a the train and test dataset to respectively 90% and 10%. Finally, we added 2 additional categories, for a total of four, depending on the 1st, 2nd and 3rd quartiles of our upvote data represented by respectively 0, 1, 2 and 3. This diminished the effect of a random guess in the model, allowing a better interpretation of our results.

With this modifications, we reached a 95% training accuracy and 30% validation accuracy in 38 epochs.

2 Discussion

Our results are significantly better than the last iteration, a 30% accurcay being better than a random guess. However, the model seems to be overfitting. Indeed, the accuracy on our testing data is still too low. Moreover, the model’s predictions seem to be out of place. For example, it will find a Physics assignment ”merely funny” (3/4), or worse, some of Raph’s memes ”very funny” (4/4). Those predictions are completely wrong and suggest a big improvement margin on our model.

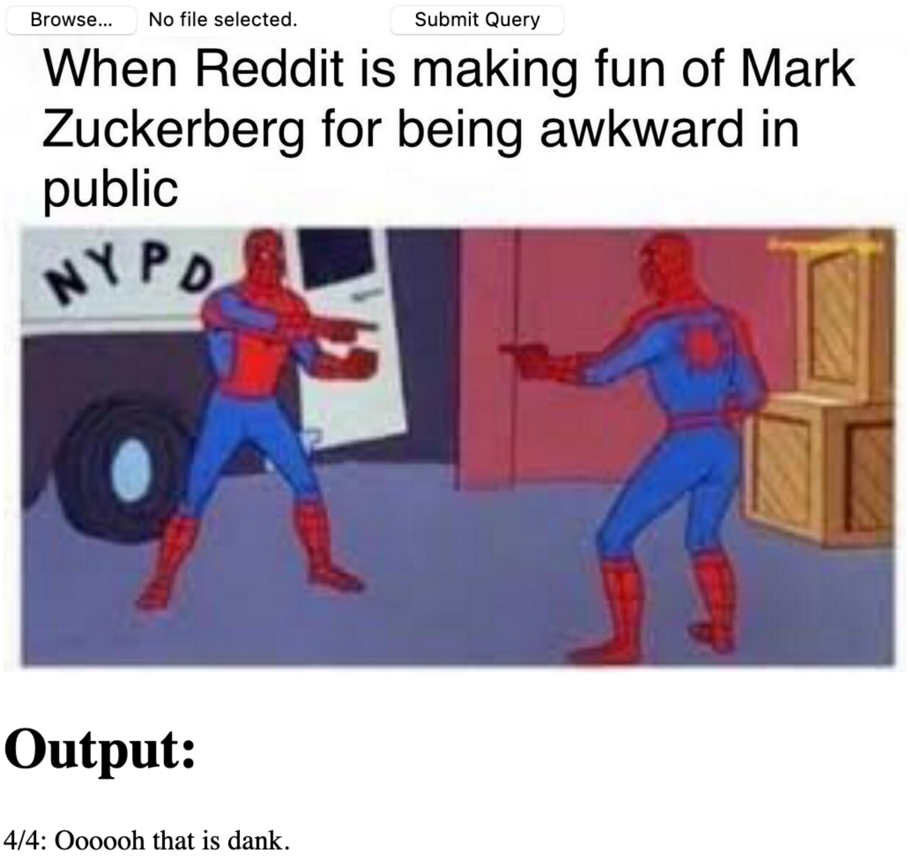
3 User interface

To make the model accessible, we developed a user interface through a web page. Users are able to upload a picture, and this picture is processed and analyzed by the model. The output integer of the machine is translated to a readable string like ”boriiiiing”, ”okayish”, ”pretty funny” or ”ooooh that’s dank”.

The frontend is pretty basic but you know, I’m not an designer soooo... Because the website is not yet deployed, a picture of it is shown in Figure 1.

4 Next Steps

First, an improvement of data is necessary. We can use built in libraries like Keras’s dataset library to have a better optimized dataset. It will perform optimized split, shuffling and caching. We can also increase our data size with tensorflow’s ImageDataGenerator. Finally, we can also improve the model by adding text caption analysis.



When Reddit is making fun of Mark Zuckerberg for being awkward in public

Output:

```
4/4: Oooooh that is dank.
```

Figure 1: Web user interface with poor frontend.