

1 Subjective/Objective Classification Problem Definition

The first lectures of this course, and the first assignment showed how textual words can be turned into word vectors that represent the meaning of a word. These form the basis of all forms of modern deep-learning-based NLP. In this assignment, we will build models that classify a sentence as objective (a statement of fact) or subjective (a statement based on opinion). To begin, make sure you understand the distinction between these two words - look up a few definitions of these words, when used as adjectives.

Subjective sentences express personal opinions, feelings, or beliefs, and are influenced by an individual's perspective. They often include emotions and judgments, such as "I think this movie is amazing."

Objective sentences present facts, data, or information without personal bias. They are based on observable evidence, like "The movie has a runtime of two hours."

2 Software Environment and Dataset

2.1 Software Environment

Software environment loaded.

2.2 Dataset

Dataset examined.

3 Preparing the data

3.1 Human Data Review

The data for this assignment was provided in the file data.tsv that you downloaded from Quercus. This is a tab-separated-value (TSV) file that contains two columns, text and label. The text column contains a text string (including punctuation) for each sentence (or fragment or multiple sentences) that is a data sample. The label column contains a binary value {0,1}, where 0 represents the objective class and 1 represents the subjective class. Do/answer each of the following questions:

1. Review the data to see how it is organized in the file. How many examples are in the file data.tsv?

```
Header: ['text', 'label']
Example 1: ['smart and alert , thirteen conversations about one thing is a small gem . ', '1']
Example 2: ['color , musical bounce and warm seas lapping on island shores . and just enough science to send you home thinking . ', '1']
Example 3: ['it is not a mass-market entertainment but an uncompromising attempt by one artist to think about another . ', '1']

Total number of examples in data.tsv: 10000
```

Here we can see some examples, there are 10000 of these in total.

2. Select two random examples each from the positive set (subjective) and two from the negative set. For all four examples, explain, in English, why it has the given label. [1 point]

Examples with label 1:

```
['the story is far-flung , illogical , and plain stupid . ', '1']
```

```
['the result is good gossip , entertainingly delivered , yet with a distinctly
musty odour , its expiry date long gone . ', '1']
```

"the story is far-flung, illogical, and plain stupid." – This sentence conveys a judgment about the story, using terms like "illogical" and "stupid," which are subjective assessments based on the speaker's personal perception.

"the result is good gossip, entertainingly delivered, yet with a distinctly musty odour, its expiry date long gone." – The phrases "good gossip," "entertainingly delivered," and "distinctly musty odour" reflect the speaker's opinion about the quality and appeal, showing subjective interpretation rather than objective fact.

Examples with label 0:

```
['at the age of 34 , with no producing credits to his name , he landed a
job as chief of production at paramount pictures . ', '0']
```

```
['lead by billy clemens , the gang includes maggie and virginia caulder ,
two beautiful sisters who have a penchant for disguise as well as gun
handling . ', '0']
```

"at the age of 34, with no producing credits to his name, he landed a job as chief of production at paramount pictures." – This sentence states specific, verifiable details about a person's age, experience, and job title, without any evaluative language.

"lead by billy clemens, the gang includes maggie and virginia caulder, two beautiful sisters who have a penchant for disguise as well as gun handling." – This sentence describes the characters and their actions in a factual manner.

While the word "beautiful" could suggest subjectivity, it is used here as a descriptive trait rather than an evaluative opinion about the story's quality. The rest of the sentence remains focused on concrete details.

3. Find one example from each of the positive and negative sets that you think has the incorrect label, and explain why each is wrong [2 points].

Example with label 0:

```
['given their jaded and precocious sophistication , is there anything the girls can look forward to ? ', '0']
```

This sentence is mislabeled because it is indeed subjective. The phrase "*jaded and precocious sophistication*" reflects the speaker's personal interpretation of the girls' characteristics, suggesting an opinion rather than stating an objective fact. Additionally, the rhetorical question "*is there anything the girls can look forward to?*" conveys a sense of judgment and implies the speaker's negative outlook, which further indicates subjectivity.

Examples with label 1:

```
['this tells us nothing about el gallo other than what emerges through his music . ', '1']
```

The statement "*this tells us nothing about el gallo other than what emerges through his music*" presents a factual observation about the information provided by the subject. It does not include any personal feelings, opinions, or emotional language. It simply states what is and is not conveyed about "el gallo" through the music, making it an objective description.

3.2 Create train/validation/test splits and Overfit Dataset

You should (programmatically) verify that there are equal number of examples in the two classes in each of the datasets, and that you did not accidentally put the same sample in more than one of the training, validation and test sets. Your code should report both of these checks. Submit your code to perform these functions in the file A2P3_2.py. [2 points]

Train/validation/test splits created, python file provided in upload.

```
Overlap between train and validation: 0
Overlap between train and test: 0
Overlap between validation and test: 0
Number of label 0 examples in overfit.tsv: 25
Number of label 1 examples in overfit.tsv: 25
```

3.3 Processing the Training Data

Preprocessing completed using starter code.

4 Baseline Model and Training

4.1 Embedding Layer

Embedding layer created inside the model

```
self.embedding = nn.Embedding.from_pretrained(vocab.vectors)
self.fc = nn.Linear(embedding_dim, 1)
```

4.2 Baseline Model

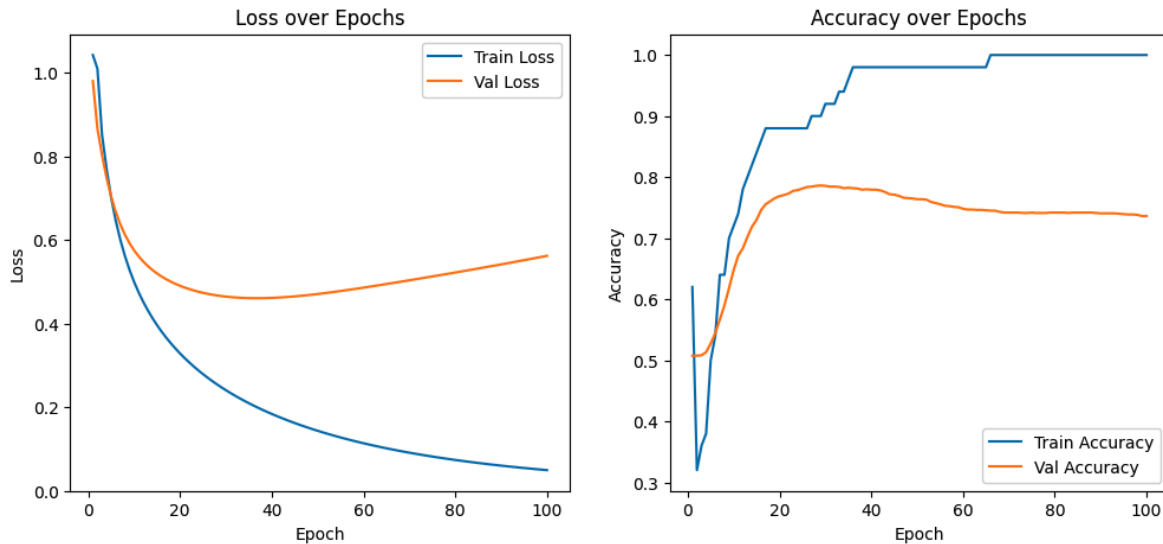
A simple baseline model was created to average the word embeddings into an FC linear layer.

4.3 Training the Baseline Model

A simple training loop was created that allows defined parameters, it doubles as a way to print and plot the results of the validation and training data sets.

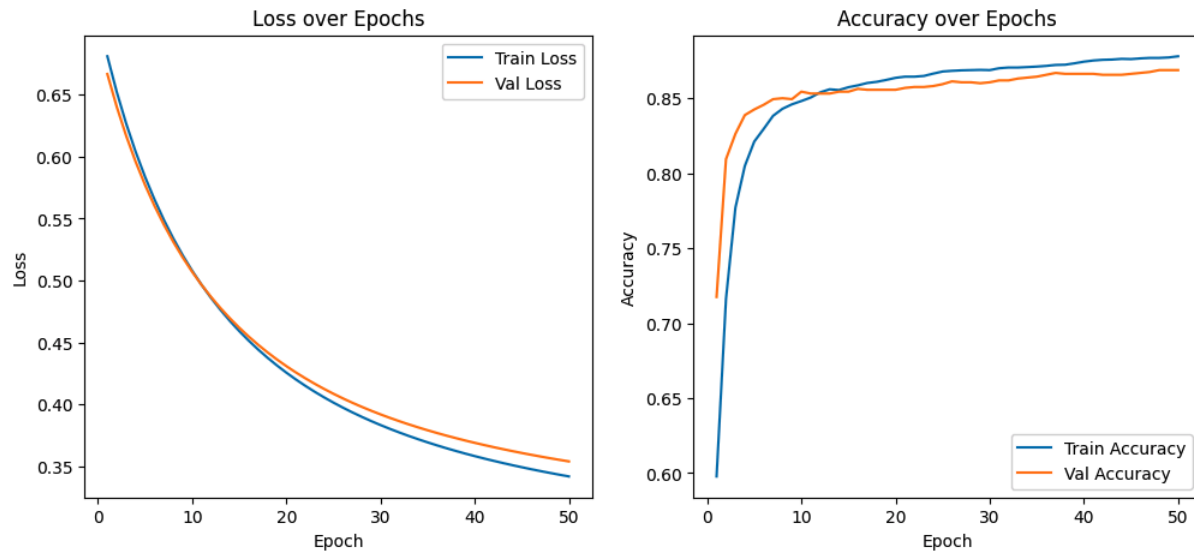
4.4 Overfitting to debug

At a higher learning rate of 0.01 to account for the lower sample size, we can observe clear overfitting over about 100 epochs. The loss graph is representative of memorization as accuracy reaches 1 on the smaller overfit set while the losses grow worse overtime for validation data as it is no longer generalizable.



4.5 Full Training Data

At a learning rate of 0.001 and over 50 epochs, we can see the model quickly achieves high accuracy over about 10 epochs and then stagnates. A batch size of 32 was chosen over trial and error for the overall best validation result at the end of 50 epochs.



Answer this questions: In the baseline model, what information contained in the original sentence is being ignored? [1 points]

The placement of each word in relation to other words is completely ignored; it takes the average of all words as a jumbled whole instead of considering the relational aspects that word meanings can take on. A simple example is that an adjective placed between different nouns can completely change the meaning of both the noun and the adjective. However, by averaging all embeddings across a single layer, this important information is lost.

4.6 Extracting Meaning from the Trained Parameters

The dimension of the parameters in the linear neuron is the same size as the word embedding, which suggests that there is a meaning attributable to the learned parameters. You can explore that meaning using the function `print_closest_cosine_words` from Assignment 1. Use that function to determine the 20 closest words to those trained parameters of the neuron. You should see some words that make it clear what the classifier is doing. Do some of the words that you generated make sense? Explain. [4 points]

```
print_closest_cosine_words(torch.tensor(neuron_parameters), n=20)
```

```
visuals      0.57
realistic    0.55
disquieting  0.53
pleasing     0.53
insipid      0.53
cheesy       0.52
flattering   0.52
laughable    0.51
cartoonish   0.51
watchable    0.51
cartoony     0.50
ludicrous    0.50
refreshingly 0.50
encapsulates 0.50
escapist     0.50
portentous   0.50
refreshing   0.50
bracingly    0.50
gimmicky     0.50
campy        0.49
```

It is quite clear from the neuron outputs that the data was trained on a movie review set. There is an emphasis on movie review terms such as "visuals" and "pleasing," as well as others like "pretentious," "refreshing," "watchable," and "cartoony," which signify a visual medium related to movie reviews. Additionally, there is an overall goal of encapsulation indicated by the term "encapsulates," and a judgment of realism, depending on the objective and subjective statements, seen in words like "realistic". Overall, these words outline how the model can filter these reviews using these parameters to identify subjectivity. It is also notable how most of these terms are based on opinions, the model is likely filtering for these identifications to find which statements are more likely to be subjective.

4.6 Saving and loading your model

See jupyter notebook for the procedure for saving the model.

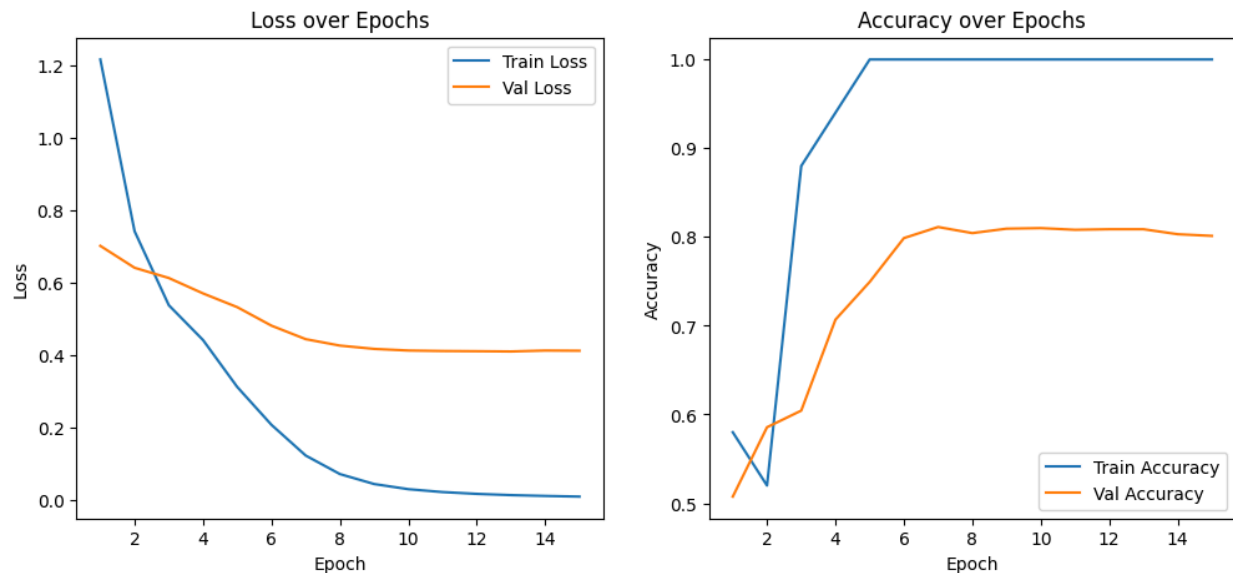
4.7 Submit Baseline Code

The notebook was submitted.

5 Convolutional Neural Network (CNN) Classifier

5.1 Overfit

Once you've finished coding the model, use the overfit dataset, and the parameters $k1 = 2$, $n1 = 50$, $k2 = 4$, $n2 = 50$ to make sure that you can overfit the model, as discussed in Section 4.4. Report the training accuracy that you were able to achieve with the overfit dataset. [1 point]



As expected, the more powerful model overfitted much more quickly, and full memorization of the small sample set was possible at 5 epochs.

5.2 Training and Parameter Exploration

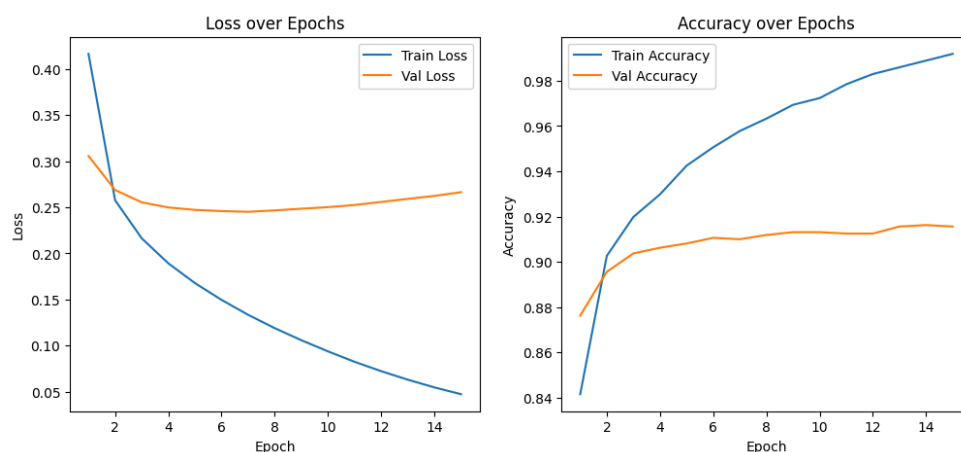
Explore the parameter space of the CNN in the following steps, using the full dataset:

1. Here you should explore the normal hyper parameters for neural networks along with the specific ones in this CNN - $k1$, $n1$, $k2$ and $n2$. As a suggestion, start with $k1 = 2$, $n1 = 10$, $k2 = 4$, $n2 = 10$ and select the other hyperparameters. After that, explore different values of $k1$, $n1$, $k2$, $n2$ to achieve the best accuracy that you can. Report the accuracy and the full hyperparameter settings. Give the training and validation curves for that best model, and describe your overall hyperparameter tuning approach. [4 points]

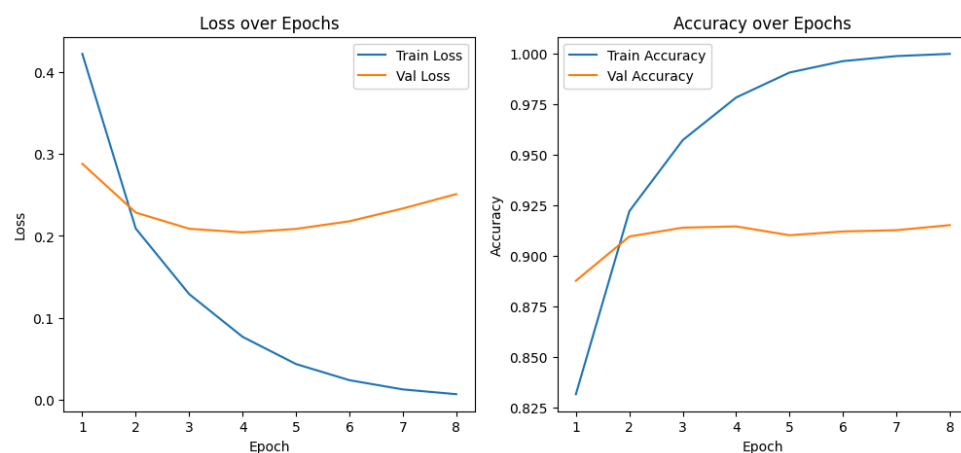
The hyperparameter optimization process involved a grid search in combination with a manual seed approach to explore parameters systematically. Initially, I started with the midpoint values for kernel sizes (' k ') and the number of filters (' n '), as these provided a balanced starting point. From there, I incrementally adjusted each value to observe their effects, using intervals of 1 to 3 depending on their impact on the model's performance. This allowed for fine-tuning and broad adjustments based on model sensitivity to each parameter change. To evaluate these adjustments efficiently, I ran the model for 10 epochs at each setting, which provided an early indication of how well the selected parameters performed without incurring a high computational cost. This early evaluation strategy helped me quickly determine the parameters' general impact before committing to a longer training cycle.

Throughout the tuning process, I observed that larger values for 'n', particularly beyond 90, led to overfitting, suggesting that the model was becoming too complex and capturing noise rather than meaningful patterns. Upon finding the most promising values for 'k' and 'n', I focused on adjusting the learning rate and the number of epochs to optimize the final model performance. I monitored the training and validation accuracy to avoid overfitting, reducing the learning rate when I observed significant deviations between them. This helped to stabilize the training, resulting in smoother training and validation loss curves, indicating a more consistent learning process.

The final hyperparameters chosen for the CNN architecture were 'k1 = 2', 'n1 = 20', 'k2 = 3', and 'n2 = 25', with a learning rate of '0.001' over '15' epochs. These settings provided the best validation accuracy while balancing sufficient model complexity and the ability to generalize effectively to unseen data. The final validation accuracy was 0.9156.



2. Re-run your best model, but allow the embeddings to be fine-tuned during the training, by setting the freeze parameter to False on the nn.Embedding.from_pretrained class. Report the accuracy of the result, and comment on the result. Save this model in a .pt file as you did in Section 4.7, for use below in Section 6. [2 points]



It was difficult to determine whether this approach produced better results; however, the peak validation accuracy was slightly higher, albeit at the cost of a longer training time. Overall, the epochs were capped at around 8, as overfitting effects became too severe beyond that point. This approach should produce better results, but the model, optimized for the validation data during hyperparameter tuning, was likely overfitted to that scenario and did not

generalize well in this context. Overall the final accuracy for validation data was 0.9150.

Based on this assumption the following parts were done assuming this “should” have been the best model, even though it is not.

5.3 Extracting Meaning From Kernels

Observe that one of the kernel’s dimension is the same size as the word embedding. Since the kernels are trained to learn values that should be present (or not present) in the input, they have an interpretable meaning, as was the case in Section 4.6. You can explore that meaning using the function `print_closest_cosine_words` from Assignment 1.

Use that function to determine the five closest words to each of the words in the the kernels trained in your best classifier. Do those words make sense? Do the set of words in each given kernel give a broader insight into what the model is looking for? Explain. [4 points]

```
Kernel 1:
inadvisable debatable advantageous preferable pourable
Kernel 2:
pursuits unsuccessful housemates in-laws housemate
Kernel 3:
roommates tormentors labelmates co-worker dorm
Kernel 4:
17-inch 556-1927 tohr problematic inbio
Kernel 5:
sweeter troublingly trifle icier impressively
Kernel 6:
l&br nsdap forty-eighth trinamul lyoness
Kernel 7:
manchevski avy brisseau padmarajan tresnjak
Kernel 8:
via ; vols. travels originating
Kernel 9:
te part-time apprentices privateer merchantman
Kernel 10:
recruited bramshill pre-med pre-law raided
Kernel 11:
decent puny pleasing classier enjoyable
Kernel 12:
life-affirming whiny pretentious inelegant nauseating
Kernel 13:
tormentors buhera hideout loughton lairs
Kernel 14:
kwin ahth pales tehl chehr
Kernel 15:
underwhelming unshowy rollicking chintzy captivating
Kernel 16:
brothel poker casino millionaire bookie
Kernel 17:
pyrams in-laws adoptive her half-sisters
Kernel 18:
art-house writer-director huppert auteur 14-screen
Kernel 19:
oscar-nominated leghorn asinine palatable prescient
Kernel 20:
impales antoin wvon kabc-tv phill
```

Using the first convolution as an example, the kernels learned by the CNN model appear to capture a range of semantic and thematic features in movie reviews. Many of the kernels focus on different aspects such as sentiment, interpersonal relationships, thematic content, and specific genres.

For instance, kernels like Kernel 1 ("inadvisable," "debatable," "advantageous," "preferable") and Kernel 11 ("decent," "puny," "pleasing," "classier," "enjoyable") seem to be focused on evaluative language, which is crucial in movie reviews where expressing opinions and judgments is common. These kernels capture the sentiment and descriptive quality of the text, which is important for distinguishing positive and negative reviews.

Other kernels, such as Kernel 3 ("roommates," "tormentors," "co-worker," "dorm") and Kernel 16 ("brothel," "poker," "casino," "millionaire"), capture the context and thematic content of the reviews. These terms often relate to specific settings or character relationships, suggesting that the model is learning to identify elements related to the storyline or the environment in which the movie takes place. This is valuable for understanding the overall themes or topics being discussed in the reviews.

Kernel 18 ("art-house," "writer-director," "auteur") highlights specific movie-related terminology, indicating that this kernel is likely identifying references to the style or production of films, such as genre or director-driven narratives. This is particularly relevant in reviews discussing the artistic aspects of movies.

On the other hand, some kernels, like Kernel 6 ("nsdap," "trinamul," "lyoness") and Kernel 14 ("kwin," "ahth," "chehr"), contain terms that appear unrelated or obscure, which suggests that these kernels may have learned patterns from less informative or noisy parts of the vocabulary. This could indicate the presence of rare words or less relevant features that could be more useful for the classification task.

The kernels highlight the model's ability to capture diverse features, ranging from sentiment and subjective evaluation to specific thematic content and terminology relevant to movie reviews. The best-performing kernels—such as Kernel 1, Kernel 11, Kernel 3, and Kernel 18—demonstrate the most obvious relationships by focusing on common themes, sentiment, and contextual details that are likely important for the model's task of understanding and classifying movie reviews effectively.

5.4 Submit CNN Code

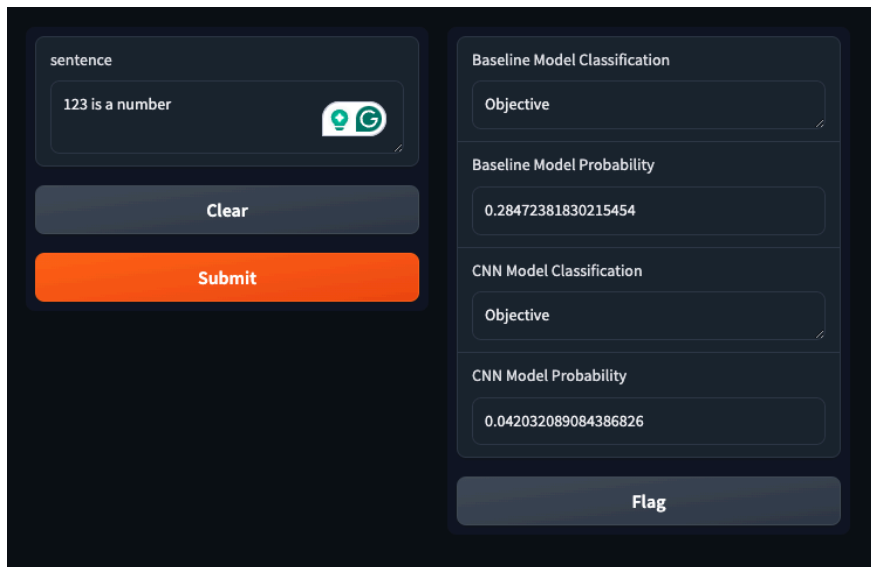
The code was submitted.

6 Web-based User Interface to Classify Input Sentences Using Gradio

6.1 Run and Compare

Run your two best stored models on 4 sentences that you come up with yourself, where two of the sentences are definitely objective/subjective, and the other two are borderline subjective/objective, according to your opinion. Include the input and output in your write up. Comment on how the two models performed and whether they are behaving as you expected. Do they agree with each other? Which model seems to be performing the best? [1 point]

2 objective sentences classified correctly:



sentence

123 is a number

Clear

Submit

Baseline Model Classification

Objective

Baseline Model Probability

0.28472381830215454

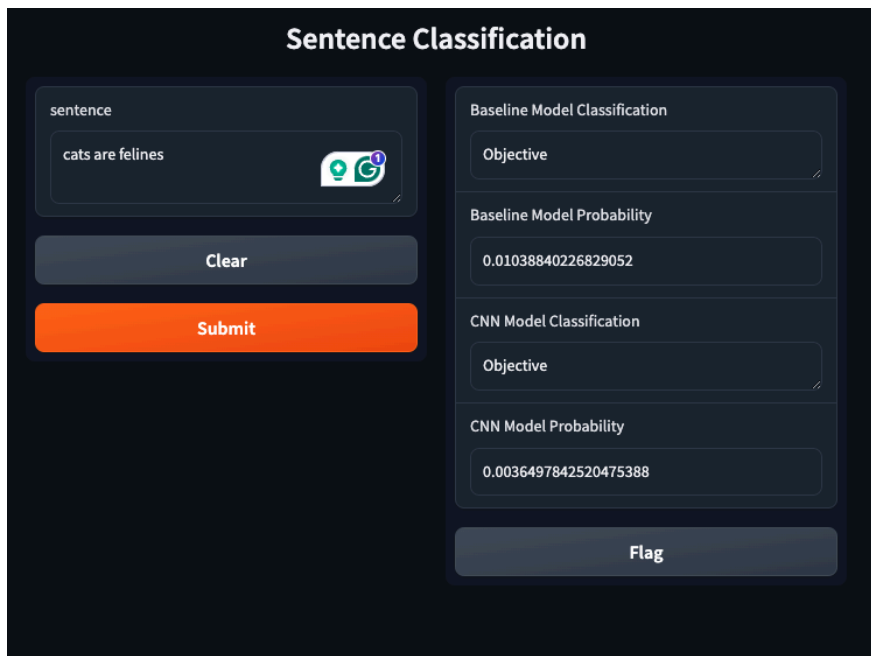
CNN Model Classification

Objective

CNN Model Probability

0.042032089084386826

Flag



Sentence Classification

sentence

cats are felines

Clear

Submit

Baseline Model Classification

Objective

Baseline Model Probability

0.01038840226829052

CNN Model Classification

Objective

CNN Model Probability


0.0036497842520475388

Flag

2 subjective sentences classified correctly:

Sentence Classification

sentence

I don't like math it's too hard 

Clear

Submit

Baseline Model Classification

Subjective

Baseline Model Probability

0.9999232292175293

CNN Model Classification

Subjective


CNN Model Probability

0.9998339414596558

Flag

Sentence Classification

sentence

Some parrots can be annoying pets 

Clear

Submit

Baseline Model Classification

Subjective

Baseline Model Probability

0.9928374886512756

CNN Model Classification

Subjective

CNN Model Probability

0.9996732473373413


Flag

The two models mostly behaved similarly and agreed on most statements. Overall, they performed as expected; both were able to correctly label the very obvious test cases provided. Notably, the CNN model appeared to be more "certain" about its answers, producing more extreme output probabilities after the sigmoid function was applied. Specifically, the baseline model tended to trend towards the middle and was less confident in identifying these obvious cases.

In some instances, however, they disagreed, with the CNN model usually being correct while the baseline model was significantly off the mark. This difference can be attributed to the baseline model's lower complexity and accuracy, as we know from previous observations. The slight 1% difference in accuracy seemed to translate into a considerable difference in performance. Below is an example where they disagreed.

Sentence Classification

sentence

dogs can be pets for some people 

Clear

Submit

Baseline Model Classification

Subjective

Baseline Model Probability

0.8157961368560791

CNN Model Classification

Objective

CNN Model Probability

0.1438072919845581

Flag

6.2 Submit Gradio Code

The code was submitted.