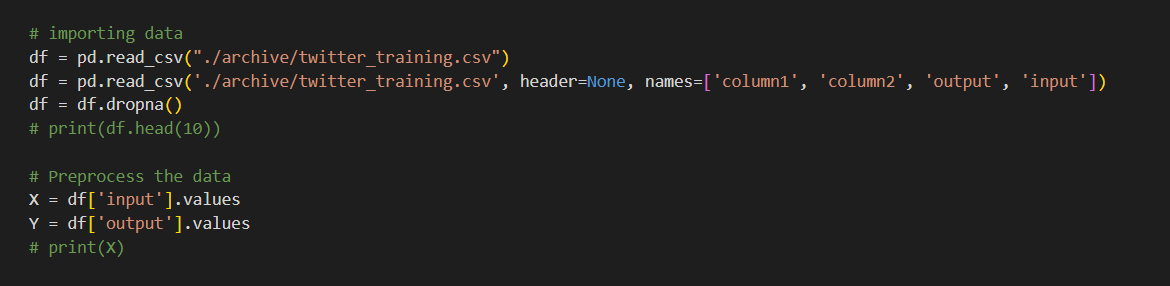
Sentiment Analysis

1. Data Collection

<https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis>

1. Data importing and preprocessing

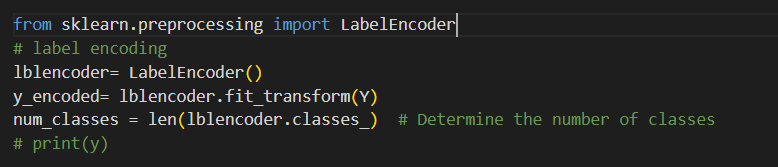


1. Label encoding

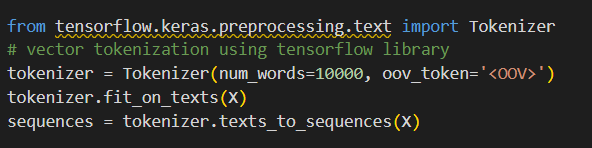
Label encoding assigns a unique integer to each category. Here’s an example:

Categories: ['cat', 'dog', 'fish']

Encoded labels: {'cat': 0, 'dog': 1, 'fish': 2}



1. Tokenization of data



* The fit\_on\_texts method updates the internal vocabulary based on the words in the text data X.
* It creates a word index dictionary where each unique word is assigned a unique integer value based on its frequency in the dataset.
* The most frequent word gets the index 1, the second most frequent word gets the index 2, and so on up to num\_words.
* The texts\_to\_sequences method transforms each text in X into a list of integers.
* Words not found in the word index (out-of-vocabulary words) are replaced with the index of the oov\_token.

Eg.

[

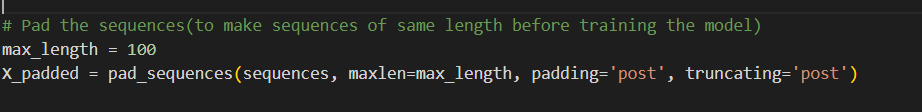
[4, 5, 6, 3], # "I love machine learning"

[7, 3, 2, 8], # "Deep learning is amazing"

[9, 10, 11, 2, 12] # "Natural language processing is fun"

]

1. Sequence padding



Sequence padding is a preprocessing technique used in natural language processing (NLP) to ensure that all sequences in a dataset have the same length.

Eg.

array([

[2, 3, 4, 5, 0], # "I love machine learning" (padded with 0 at the end)

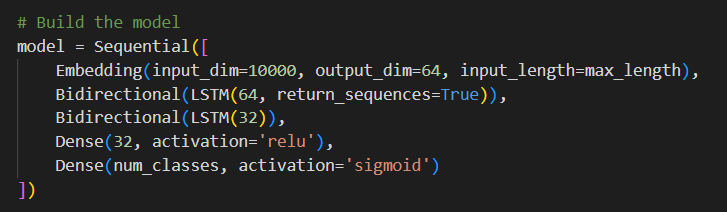
[6, 5, 7, 8, 0], # "Deep learning is amazing" (padded with 0 at the end)

[9, 10, 11, 0, 0] # "Natural language processing" (padded with 0 at the end)

])

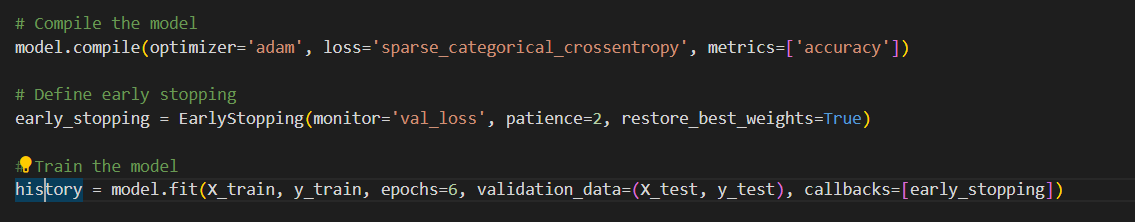
1. Building the model

The Sequential class is used to create a linear stack of layers, meaning the output of one layer is the input of the next.

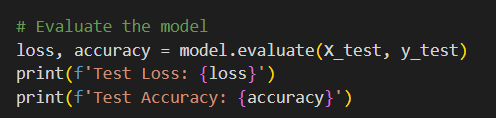


1. Training the model

An Adam optimizer trains the model, changing the learning rate as needed. “sparse categorical cross entropy” is used as a loss function. Because cross-entropy is used for categorical data and specifically sparse, cross-entropy is used to work with natural integer categorical vectors like ytest=[0 0 1 0 2 3]. If we use categorical cross entropy, one-hot encoding should be done, which is more memory-consuming.



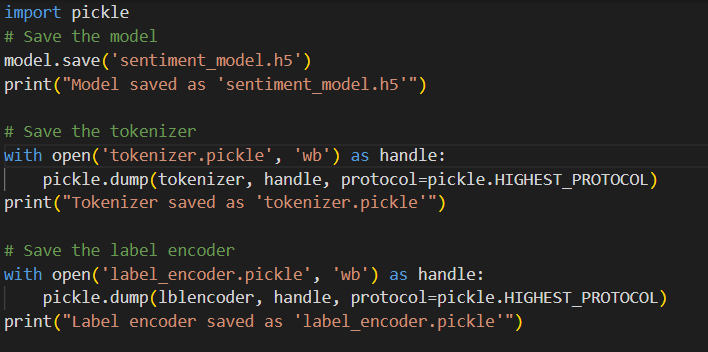
1. Validating



This code trains the model, uses early stopping to prevent overfitting, and evaluates the model's performance on the test set, providing useful metrics to assess the model's generalization ability.

1. Using pickle to save model, label encoder and tokenizer

This code saves a trained model, tokenizer, and label encoder to disk, which allows you to reuse them later without needing to retrain the model or recreate the tokenizer and label encoder.



1. with open('tokenizer.pickle', 'wb') as handle:: Opens a file named tokenizer.pickle in write-binary mode.
2. pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST\_PROTOCOL): Serializes the tokenizer object and writes it to the opened file using the highest available protocol for pickling.
3. Tokenizers are often used in text processing to convert text into sequences of tokens (e.g., words or subwords), and saving it ensures that you can later preprocess text in the same way as during training.
4. Similarly, pickle.dump(lblencoder, handle, protocol=pickle.HIGHEST\_PROTOCOL): Serializes the label encoder object and writes it to the opened file using the highest available protocol for pickling.
5. Label encoders are used to convert categorical labels into numerical form, and saving it ensures that you can later convert labels in the same way as during training.