515 HOMEWORK 5:
ONLINE REVIEW CLASSIFICATION

Your assignment is to create a tool that trains several machine learning models to perform the task of classifying online reviews. Some of these online reviews refer to hazardous products, so these machine learning models will help to identify the most serious product complaints.

The dataset is available at https://dgoldberg.sdsu.edu/515/appliance_reviews.json and contains approximately 1,000 reviews, approximately half of which refer to safety hazards. The data is formatted as a JSON array. An example of the formatting is below:


```
[
    {"Review": "I was really surprised how quickly it was
    shipped. Ordered it on Sunday and it was delivered the
    following Friday. I couldn't be happier with my steamer.
    Works perfectly for any type of vegetable or rice. Easy to
    use and clean up after.", "Stars": 5, "Safety hazard": 0},

    ...

    {"Review": "On the 2nd time that I used it, one of the
    PLASTIC clips holding the upper plate broke and the VERY
    HOT plate came loose falling.  Fortunately, I was standing
    there at the time it fell.  This was a very dangerous
    situation and could have ruined my kitchen counter and
    caused a fire.  Heating element behind the plate was
    exposed.  Using a cheap plastic clip to hold a heavy
    cooking plate in place is just plain stupid engineering.  I
    expected better from a Cuisinart product.", "Stars": 1,
    "Safety hazard": 1}
                                                            ]
```


The purpose of the machine learning models is to predict the "Safety hazard" field, which is already formatted as a 0 or 1. A value of 1 indicates that the review refers to a safety hazard; a value of 0 indicates that the review does not refer to a safety hazard. However, to transform the reviews into a format usable by the machine learning models, perform the following steps:

- Throughout the problem, ensure that you handle case-sensitivity (for example, by converting all reviews to lowercase).

- Next, create a list of all the *unique* words in the dataset. For example, the word "plastic" occurs multiple times throughout the dataset. However, this is only one unique word, so only append it to your list one time.

- The dataset consists of many words, so the next step is to narrow down which words are relevant to the classification problem (otherwise, the machine learning models may have too many variables to consider and run very slowly). To do so, generate a "relevance score" for each word by first computing totals of A, B, C, and D:

	Review refers to a safety hazard	Review does not refer to a safety hazard
Review contains word	A	B
Review does not contain word	C	D

Then, create a consolidated relevance score using:

$$Relevance\ score = \frac{\sqrt{(A+B+C+D)} \times (AD - CB)}{\sqrt{(A+B) \times (C+D)}}$$

It is possible, rarely, for this formula to cause errors because of a denominator of zero (you can handle this with try-except). In these situations, use a relevance score of zero for that word.

Words with higher relevance scores are more likely to make good predictors. To filter down to just the most important words, we can set a cutoff for relevance scores. For the purposes of this assignment, create a list of words whose relevance scores are at least 4,000. Based on this cutoff, you should have *approximately* 29 words. You may have slightly more or fewer words depending on how you handled some of the steps above. More than one solution is acceptable here, so please do not worry if you have a slightly different number of words.

For example, the word "dangerous" should be one of the 29 words that meet the cutoff. For the word "dangerous," you should see the following values:

	Review refers to a safety hazard	Review does not refer to a safety hazard
Review contains word	A = 101	B = 0
Review does not contain word	C = 397	D = 502

$$Relevance\ score = \frac{\sqrt{(A+B+C+D)} \times (AD - CB)}{\sqrt{(A+B) \times (C+D)}} = 5320.893$$

- Next, create a 2D list to train the machine learning models based on the relevant words from the previous step. If a review contains a given word, then use a value of 1, and if not, then use a value of 0. For example, suppose that the relevant words are [“dangerous”, “hazard”, “broken”] and that you are considering the review “the product was dangerous and scary.” This review should be treated as [1, 0, 0] because it contains the word “dangerous” but does not contain the words “hazard” or “broken.”

- Finally, train decision tree, k-nearest neighbors, and neural network machine learning models. You may choose an appropriate training/test split. Report the accuracy values from all three machine learning models and save a joblib file from the most accurate model.

The printout of your code may be brief. *Optionally*, you can give the user some indication of the code’s current progress by showing some progress messages along the way. For example:

```
Loading data...
Completed in 0.2932143211364746 seconds.

Identifying unique words...
Completed in 3.895111322402954 seconds.

Generating relevance scores...
Completed in 21.75375986099243 seconds.

Formatting 2D list...
Completed in 0.022971391677856445 seconds.

Training machine learning models...
Completed in 1.0733835697174072 seconds.

Decision tree accuracy: 0.81
k-nearest neighbors accuracy: 0.815
Neural network accuracy: 0.875
Neural network model performed best; saved to model.joblib.
```

Optionally, you can time each step of the process using Python's pre-installed `time` module. You can use this as follows:

```
import time
start_time = time.time() # Timestamp for when process started

# Insert code to perform some process here

end_time = time.time() # Timestamp for when process ended
time_elapsed = end_time - start_time # Difference between times

print("Completed in", time_elapsed, "seconds.")
```

Some considerations as you write your program:

- Consider the possibility that, when loading the dataset, some connection issue occurs (that is, a status code other than 200). Ensure that your code handles this case and provides the user with a helpful printout if it does occur.

- It is possible that your accuracy numbers may differ slightly from the values above due to differences in your training/test split; if your values are generally close, though, then this is not a problem.

- When training your neural network model, you may see a warning message "Maximum iterations reached and the optimization hasn't converged yet." This message means that the neural network model would have preferred to work with a larger dataset, but it does not actually cause an error. However, *optionally*, if you would like to turn this message off, then you can do so using the following:

  ```
  import warnings
  warnings.filterwarnings("ignore")
  ```

- Ensure that your output is crisp, professional, and well-formatted. For example, ensure that you have used spaces appropriately and checked your spelling.

- Adding comments in your code is encouraged. At minimum, please use a comment at the start of your code to describe its basic functionality. In addition, for any functions or classes in your code, write appropriate docstrings. Ensure that your code would be as understandable as possible for a programmer working with your code for the first time.