Conservation of AI-Based Artworks
Assignment #5 Experiments with Predictive Models
Hee-Eun Kim
Report on distilbert/distilbert-base-uncased-finetuned-sst-2-english

Sentiment Analysis: Prediction & Text Classification

- A brief description of the chosen model and its original purpose:

This is a text classification model designed to analyze the sentiment of text inputs. It is a predictive model that determines whether a given sentence expresses a positive or negative sentiment. The model can be particularly useful for classifying text sentiments and evaluating reviews of products, services, or food, enabling companies to better understand customers' feedback and make improvements.

- Testing methodology and metrics of success:

To evaluate the model, I have tested it with a variety of text inputs. These include general statements that convey clear sentiment, ambiguous and sarcastic expressions, descriptions of artwork or a product, and Amazon reviews (positive, negative, and neutral) that vary in writing style and length. Additionally, since the model is primarily designed for English-speaking users, I have evaluated its performance on non-English inputs such as Korean and French. I also experimented with inputs that include or exclude punctuation, paying particular attention to the effect of an exclamation point.

- Analysis of when the model works best and when it doesn't.

    - Accuracy for clearly positive or negative statements

"Good" answers to this model were straightforward sentences that clearly express emotions, especially when they include positive words like "good, love, delicious, wonderful," along with exclamation marks. However, when I tested sentences with mixed emotions, combining positive and negative elements, vague or neutral sentiments, and sarcastic expressions, the model did not work properly. For example, when I input sarcastic sentences about cold weather, the model failed to detect the intended sarcasm and returned a "positive" sentiment.

```
"I loved the taco! It was so good.", #english text
"I absolutely loved the taco! It was soooooo good.", #english text
"I absolutely loved the taco! It was really delicious.", #english text
"J'ai vraiment adoré le taco ! C'était incroyable.", #french text
"The dinner was average, neither good nor bad.", #neutral sentiment, english
"Le dîner était moyen, ni bon ni mauvais.", #netrual, french
"여기 밥 정말 맛있다.", #positive sentiment with a period, Korean
"여기 밥 정말 맛있다!", #positive sentiment with an exclamation mark, Korean
"여기 밥 정말 맛없다!", #negative sentiment with an exclamation mark, Korean
"Food was so-so.", #neutral sentiment, english
"Food was okay.", #neutral sentiment, english
"la nourriture était ni bien ni mal.", #neutral sentiment, French
```

    - Slight modifications in sentences phrasing affect the prediction

Conservation of AI-Based Artworks
Assignment #5 Experiments with Predictive Models
Hee-Eun Kim

I wanted to evaluate whether the model could detect varying intensities of emotion by using intensifiers such as “so,” “soooooo,” or “really” to emphasize strong feelings. However, the model does not seem to recognize these intensifiers effectively. For instance, a sentence without any emphasizing adverb received a more positive sentiment score than one that used an intensifier, suggesting that the model may not capture the increased emotional intensity.

```
Device set to use cpu
Input: I loved the taco! It was so good.
Output: [{'label': 'POSITIVE', 'score': 0.9998798370361328}]

Input: I absolutely loved the taco! It was soooooo good.
Output: [{'label': 'POSITIVE', 'score': 0.9998747110366821}]

Input: I absolutely loved the taco! It was really delicious.
Output: [{'label': 'POSITIVE', 'score': 0.9998830556869507}]
```

- Accuracy for neutral statements or non-English texts

The model struggles with neutral sentences and non-English texts. For example, when I provided a positive statement in French, the output was strongly negative. This suggests that the model is not effective at detecting sentiments in foreign languages such as French or Korean, and it appears to be optimized solely for English. Additionally, when given a neutral statement that does not convey a clear positive or negative sentiment, the model fails to classify it accurately.

```
Device set to use cpu
Input: I loved the taco! It was so good.
Output: [{'label': 'POSITIVE', 'score': 0.9998798370361328}]

Input: I absolutely loved the taco! It was soooooo good.
Output: [{'label': 'POSITIVE', 'score': 0.9998747110366821}]

Input: I absolutely loved the taco! It was really delicious.
Output: [{'label': 'POSITIVE', 'score': 0.9998830556869507}]

Input: J'ai vraiment adoré le taco ! C'était incroyable.
Output: [{'label': 'NEGATIVE', 'score': 0.884181022644043}]

Input: The dinner was average, neither good nor bad.
Output: [{'label': 'NEGATIVE', 'score': 0.9314054250717163}]

Input: Le dîner était moyen, ni bon ni mauvais.
Output: [{'label': 'POSITIVE', 'score': 0.6998422145843506}]

Input: 여기 밥 정말 맛있다.
Output: [{'label': 'POSITIVE', 'score': 0.7750498652458191}]

Input: 여기 밥 정말 맛있다!
Output: [{'label': 'POSITIVE', 'score': 0.9777301549911499}]

Input: 여기 밥 정말 맛없다!
Output: [{'label': 'POSITIVE', 'score': 0.9777301549911499}]

Input: Food was so-so.
Output: [{'label': 'POSITIVE', 'score': 0.9465882182121277}]

Input: Food was okay.
Output: [{'label': 'POSITIVE', 'score': 0.9997972846031189}]

Input: la nourriture était ni bien ni mal.
Output: [{'label': 'NEGATIVE', 'score': 0.9593074917793274}]
```

- Accuracy for ambiguous or sarcastic statements

The model struggles with detecting sarcastic sentiments. When provided with sarcastic inputs, it tends to latch onto positive words such as “wonderful” and mistakenly predicts a positive sentiment. Additionally, even when a sentence includes an exclamation mark despite conveying a negative comment, the model still classifies it as positive.

```
Input: Oh, wonderful, another freezing day. Just what I needed.
Output: [{'label': 'POSITIVE', 'score': 0.9998562335968018}]

Input: It was a great experience, though the wait was sooo long!
Output: [{'label': 'POSITIVE', 'score': 0.9946223497390747}]
```

- Long sentences including a product review, artwork description, and warning quotes

I experimented further with Amazon reviews, including positive (5 stars), negative (1 star), and neutral (3 stars) examples. The model accurately detects long reviews, particularly when the sentiment is clearly positive or negative. However, it assigns a relatively lower positive score to neutral reviews. The output is very deterministic when the sentiment is clear. In tests with artwork descriptions, even when the content involved themes of destruction, the model still classified the text as “positive.” Conversely, for a product’s suggested use, despite the instruction conveying neutral sentiment, the output was “negative.” So, I searched for a warning quote by James Russell Lowell, and the model classified it as “positive” as well.

```
Input: This is my first time using this brand. I tend to use much more expensive well known brands but my favorite was sold out, so I did extensive research and decided to try
Output: [{'label': 'POSITIVE', 'score': 0.9896043539047241}]

Input: New packaging states still 4.2 oz. Look at the difference in the jar sizes. There is just no way you’re getting the same amount of product. I feel ripped off.
Output: [{'label': 'NEGATIVE', 'score': 0.9997307658195496}]

Input: It absorbs well, does not leave a greasy feel. Found the product just ok, nothing special.
Output: [{'label': 'POSITIVE', 'score': 0.9201300740242004}]

Input: This is a piece of Tinguely's “self-constructing and self-destroying work of art,” an enormous kinetic sculpture composed of bicycle wheels, motors, a player piano, a g
Output: [{'label': 'POSITIVE', 'score': 0.8743187785148621}]

Input: Ultra-hydrating facial cream with 4.5% Squalane and Glacial Glycoprotein for uniquely boosted and deeply hydrated skin. Works to reduce redness and infuse a calming ser
Output: [{'label': 'POSITIVE', 'score': 0.993124783039093}]

Input: Use after cleansing and toning. Apply dime-sized amount to skin and massage until absorbed.
Output: [{'label': 'NEGATIVE', 'score': 0.9518316984176636}]

Input: Long-lasting 72-hour hydration, leaving skin feeling soft, supple, and nourished. Balances skin's driest areas.
Output: [{'label': 'POSITIVE', 'score': 0.9610685706138611}]

Input: Leah Dickerman: From 1959 to about 1962, Rauschenberg made a series of works that he called trophies. All of these works were dedicated to key people in his life, peop
Output: [{'label': 'POSITIVE', 'score': 0.999376118183136}]

Input: Robert Rauschenberg: Every now and then, you wanna thank somebody back who has given you so much, then there’s a new trophy. It’s just a special kind of thanks that has
Output: [{'label': 'POSITIVE', 'score': 0.9995813965797424}]
```

- Visuals (screenshots or charts) that illustrate some of the results.

```python
import torch
from transformers import DistilBertTokenizer, DistilBertForSequenceClassification

tokenizer = DistilBertTokenizer.from_pretrained("distilbert-base-uncased-finetuned-sst-2-english")
model = DistilBertForSequenceClassification.from_pretrained("distilbert-base-uncased-finetuned-sst-2-english")

inputs = tokenizer("Hello, my dog is cute", return_tensors="pt")
inputs = tokenizer("The salmon bread was amazing. I loved it so much.", return_tensors="pt")
inputs = tokenizer("I don't understand why the music has to be so loud in this cafe.", return_tensors="pt")
inputs = tokenizer("wow, today's whether is so nice. It's like I am going to get cold in a second.", return_tensors="pt")
inputs = tokenizer("Oh, what a magnificent day—I've always dreamed of becoming a human popsicle!", return_tensors="pt")

with torch.no_grad():
    logits = model(**inputs).logits

predicted_class_id = logits.argmax().item()
model.config.id2label[predicted_class_id]
```

✓ 0.6s

```
'POSITIVE'
```

```
Device set to use cpu
Input: I absolutely loved the taco! It was amazing.
Output: [{'label': 'POSITIVE', 'score': 0.9998847246170044}]

Input: J'ai vraiment adoré le taco ! C'était incroyable.
Output: [{'label': 'NEGATIVE', 'score': 0.884181022644043}]

Input: The dinner was average, neither good nor bad.
Output: [{'label': 'NEGATIVE', 'score': 0.9314054250717163}]

Input: Le dîner était moyen, ni bon ni mauvais.
Output: [{'label': 'POSITIVE', 'score': 0.6998422145843506}]

Input: Oh, wonderful, another freezing day. Just what I needed.
Output: [{'label': 'POSITIVE', 'score': 0.9998562335968018}]

Input: It was a great experience, though the wait was sooo long!
Output: [{'label': 'POSITIVE', 'score': 0.9946223497390747}]
```

- Information about the model's provenance (who trained it, where, the training data used, etc.)

This model was developed by the Hugging Face, and the parent model is a transformer model called DistilBERT. It is smaller and faster than BERT, which was trained on the same corpus in a self-supervised fashion, using the BERT base model as a teacher. This means it was pre-trained on the raw texts only, with no humans labeling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts using the BERT base model. DistilBERT pre-trained on the same data as BERT, which is BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (excluding lists, tables, and headers).