Sentiment Analysis Assessment

Task #1: Perform vector arithmetic on your own words

Write code that evaluates vector arithmetic on your own set of related words. The goal is to come as close to an expected word as possible. Please feel free to share success stories in the Q&A Forum for this section!

In [ ]:

```
# Import spaCy and load the language library. Remember to use a larger model!
```

In [ ]:

```
# Choose the words you wish to compare, and obtain their vectors
```

In [ ]:

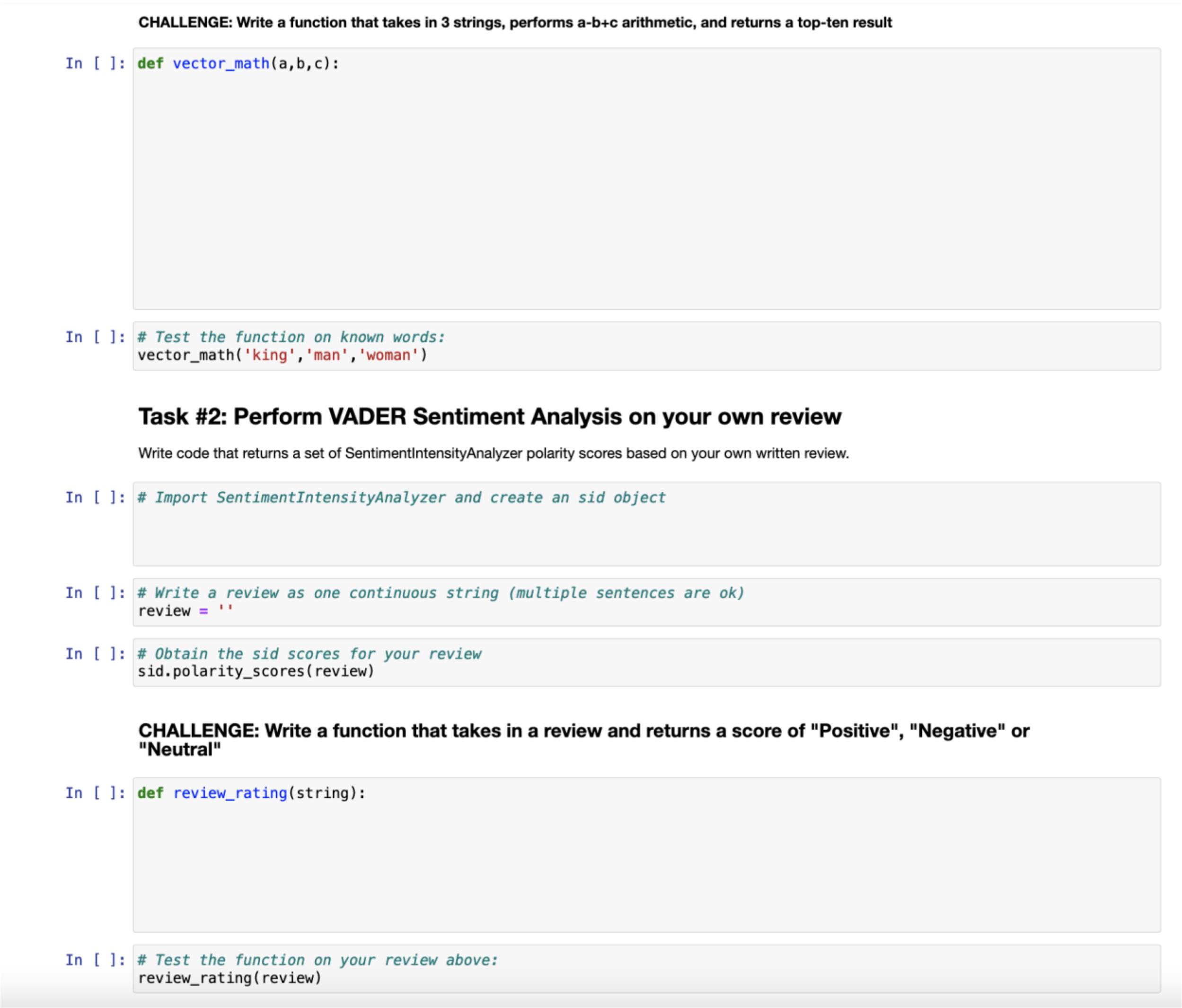
```
# Import spatial and define a cosine_similarity function
```

In [ ]:

```
# Write an expression for vector arithmetic
# For example: new_vector = word1 - word2 + word3
```

In [ ]:

```
# List the top ten closest vectors in the vocabulary to the result of the expression above
```



CHALLENGE: Write a function that takes in 3 strings, performs a-b+c arithmetic, and returns a top-ten result

In [ ]:
```
def vector_math(a,b,c):
```

In [ ]:
```
# Test the function on known words:
vector_math('king','man','woman')
```

Task #2: Perform VADER Sentiment Analysis on your own review

Write code that returns a set of SentimentIntensityAnalyzer polarity scores based on your own written review.

In [ ]:
```
# Import SentimentIntensityAnalyzer and create an sid object
```

In [ ]:
```
# Write a review as one continuous string (multiple sentences are ok)
review = ''
```

In [ ]:
```
# Obtain the sid scores for your review
sid.polarity_scores(review)
```

CHALLENGE: Write a function that takes in a review and returns a score of "Positive", "Negative" or "Neutral"

In [ ]:
```
def review_rating(string):
```

In [ ]:
```
# Test the function on your review above:
review_rating(review)
```