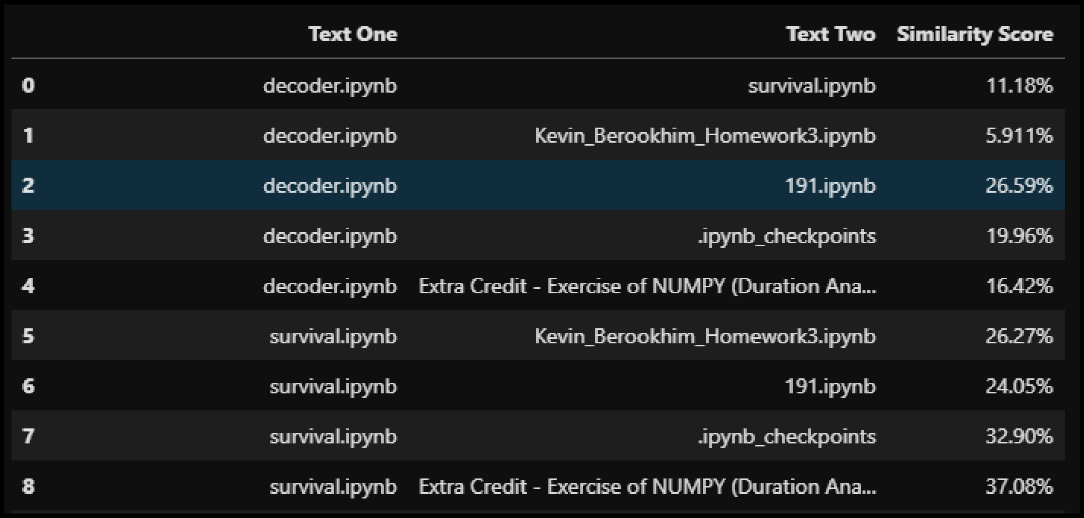
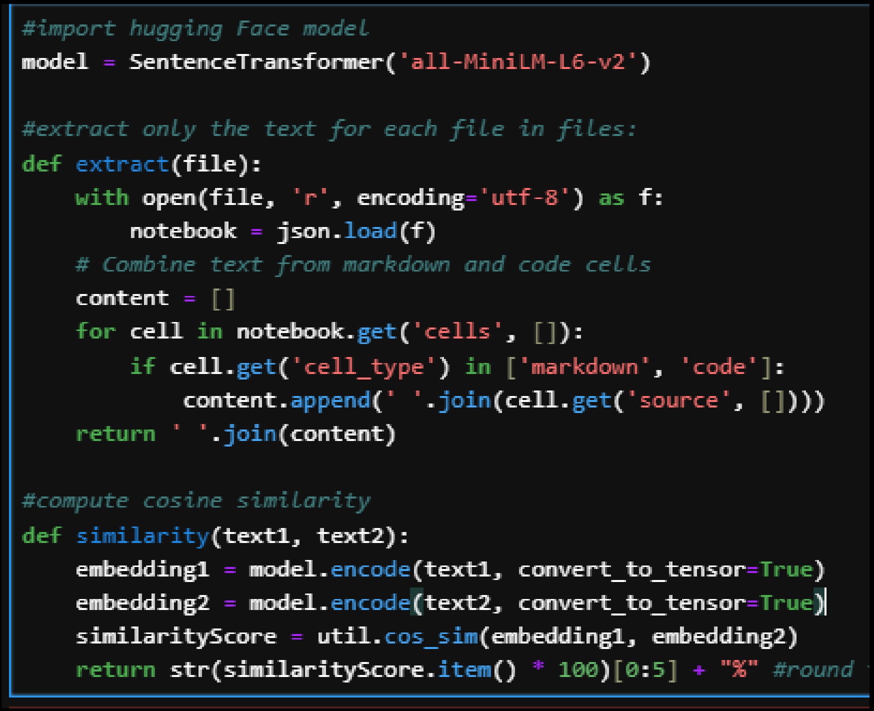


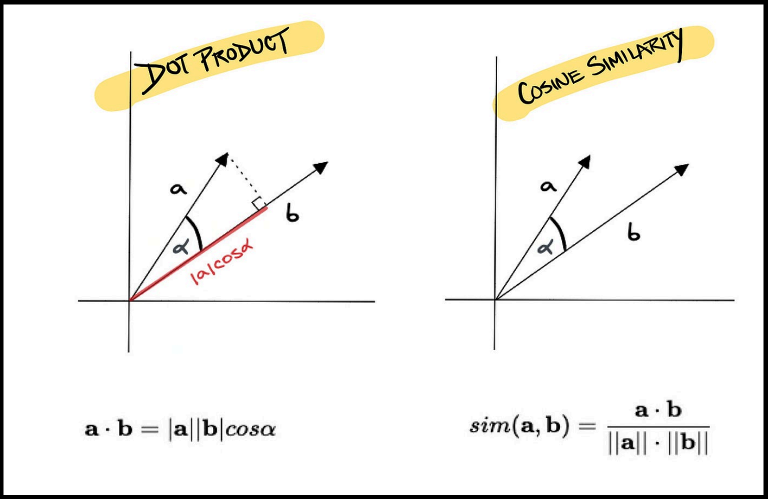
	Text One	Text Two	Similarity Score
0	decoder.ipynb	survival.ipynb	11.18%
1	decoder.ipynb	Kevin_Berookhim_Homework3.ipynb	5.911%
2	decoder.ipynb	191.ipynb	26.59%
3	decoder.ipynb	.ipynb_checkpoints	19.96%
4	decoder.ipynb	Extra Credit - Exercise of NUMPY (Duration Ana...	16.42%
5	survival.ipynb	Kevin_Berookhim_Homework3.ipynb	26.27%
6	survival.ipynb	191.ipynb	24.05%
7	survival.ipynb	.ipynb_checkpoints	32.90%
8	survival.ipynb	Extra Credit - Exercise of NUMPY (Duration Ana...	37.08%

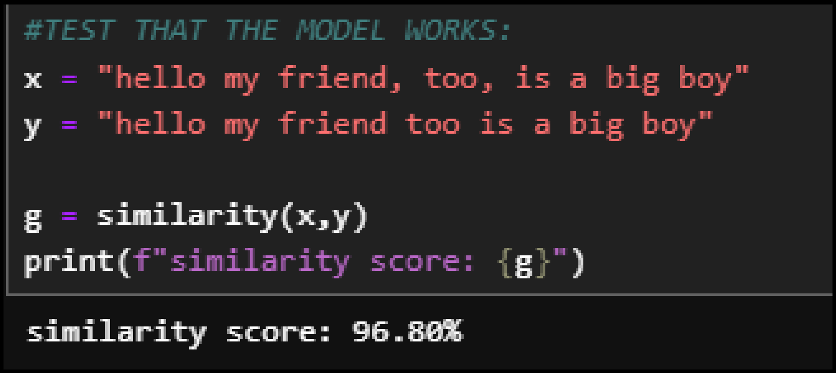


```python
#import hugging Face model
model = SentenceTransformer('all-MiniLM-L6-v2')

#extract only the text for each file in files:
def extract(file):
    with open(file, 'r', encoding='utf-8') as f:
        notebook = json.load(f)
    # Combine text from markdown and code cells
    content = []
    for cell in notebook.get('cells', []):
        if cell.get('cell_type') in ['markdown', 'code']:
            content.append(' '.join(cell.get('source', [])))
    return ' '.join(content)

#compute cosine similarity
def similarity(text1, text2):
    embedding1 = model.encode(text1, convert_to_tensor=True)
    embedding2 = model.encode(text2, convert_to_tensor=True)
    similarityScore = util.cos_sim(embedding1, embedding2)
    return str(similarityScore.item() * 100)[0:5] + "%" #round
```


Dot Product
a
b
α
|a|cosα
$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}|cos\alpha$
Cosine Similarity
a
b
α
$sim(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||}$



```
#TEST THAT THE MODEL WORKS:
x = "hello my friend, too, is a big boy"
y = "hello my friend too is a big boy"

g = similarity(x,y)
print(f"similarity score: {g}")
```

```
similarity score: 96.80%
```