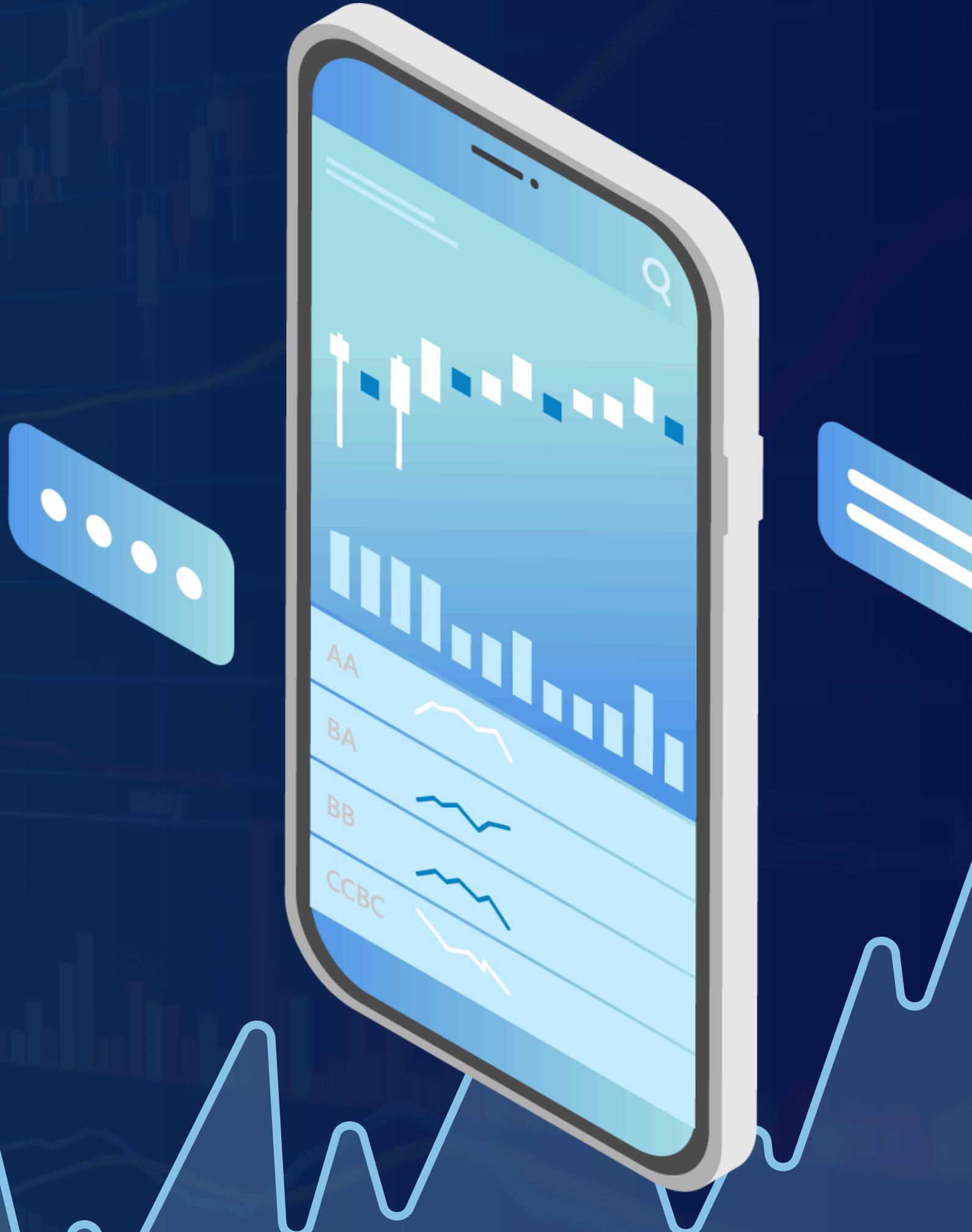


Mini Project

Text_Analytics



Problem 3

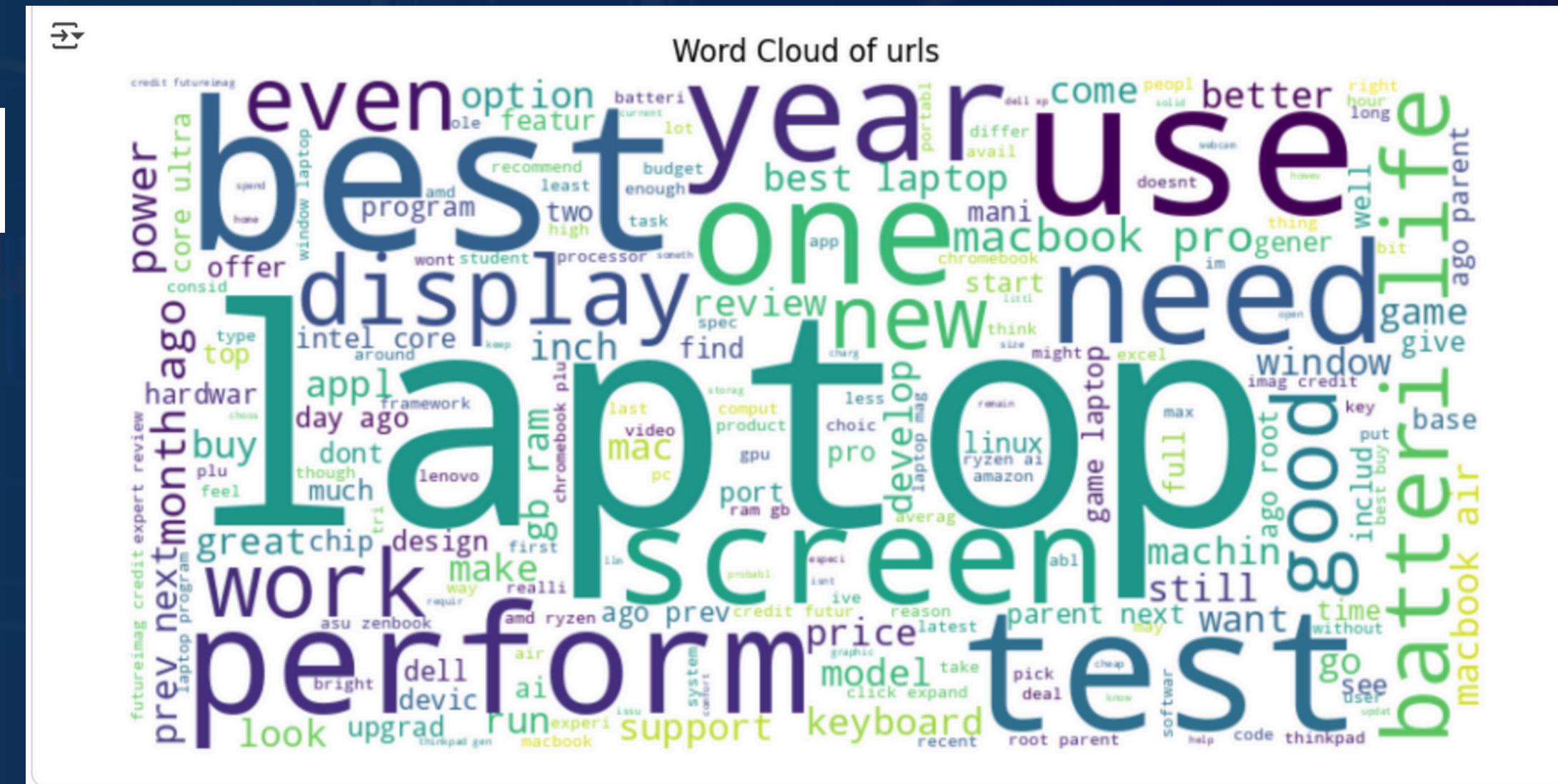
Review Laptop for programming 2025

ใช้ counter เพื่อนับจำนวนความถี่ค่า

```
5 word_freq = Counter(cleaned_text.split())
6 top_words = word_freq.most_common(10)
```

→ Top 10 most frequent words:

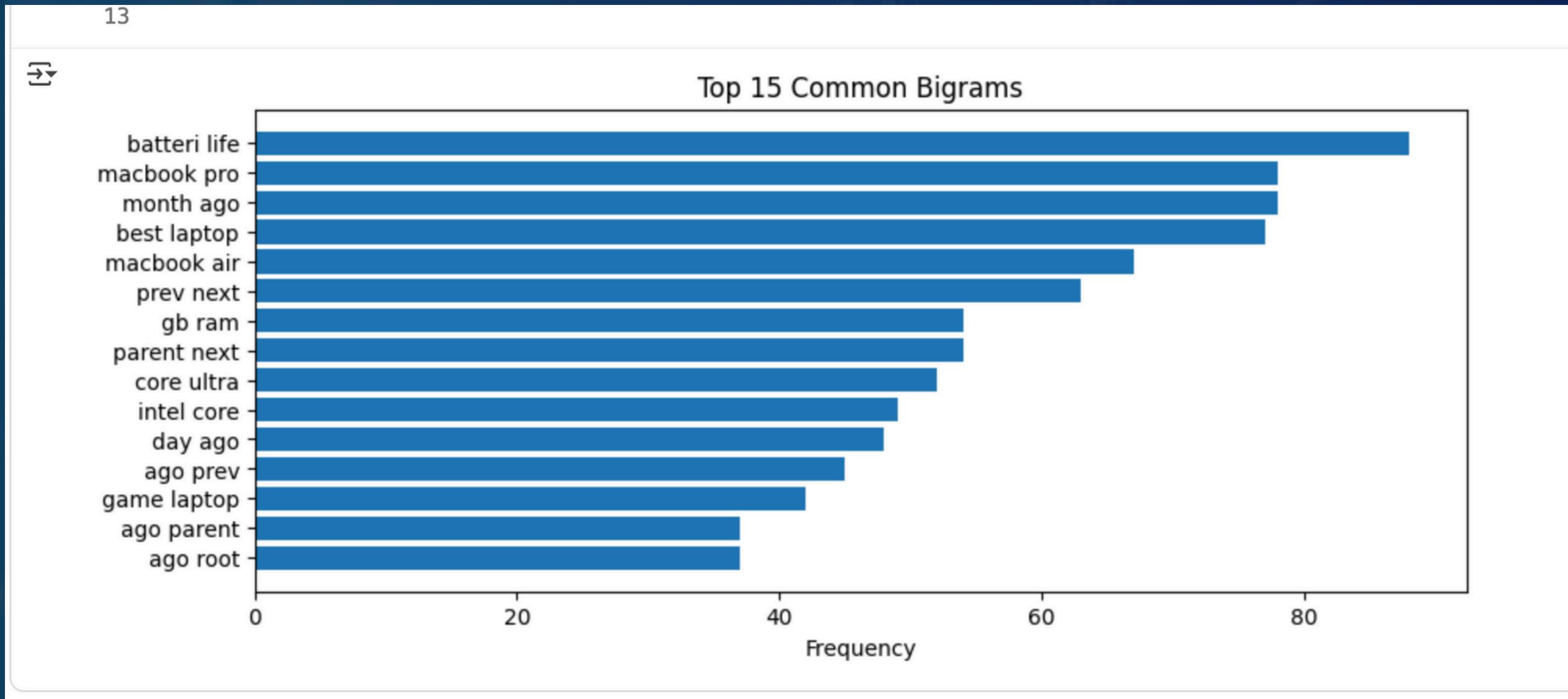
- laptop: 713
- best: 384
- macbook: 213
- pro: 199
- gb: 179
- inch: 178
- batteri: 156
- use: 146
- review: 143
- like: 140



Problem 3

คำที่มักปรากฏร่วมกัน บ่งบอกประเด็นสำคัญอะไร

ใช้ bigram



บ่งบอกว่าการรีวิว laptop ส่วนใหญ่พูดถึง batteri life คืออายุการใช้งานของแบตเตอรี่

Problem 3

คำสำคัญของแต่ละ url

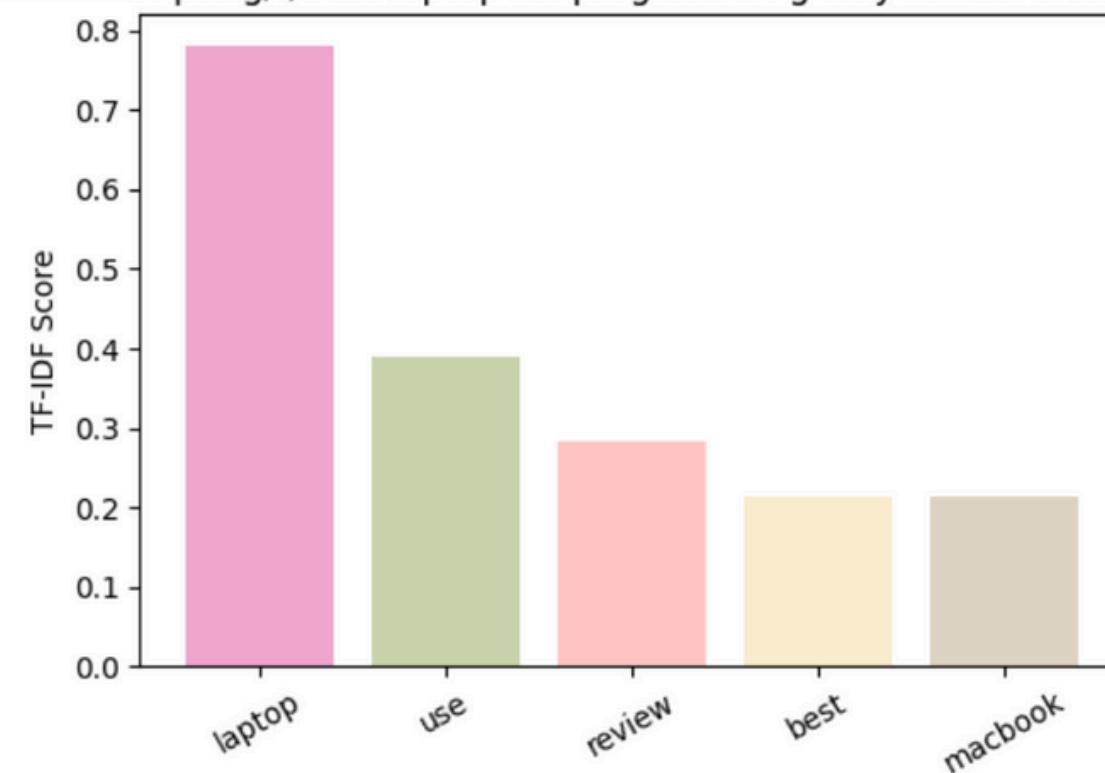
7 # สร้าง TF-IDF matrix

```
8 vectorizer = TfidfVectorizer(max_features=20)  
9 tfidf_matrix = vectorizer.fit_transform(docs)
```

เข้าใกล้ 1 แปลว่าคำนี้สำคัญมาก ในเอกสาร
นั้น และไม่ค่อยปรากฏในเอกสารอื่น
ใกล้ 0 → คำนี้ปรากฏบ่อยในเอกสารอื่น ๆ
ด้วย → ไม่เฉพาะเจาะจง

◆ Top 5 words for https://forum.freecodecamp.org/t/best-laptop-for-programming-why-review-sites-got-it-wrong/762870?utm_source=chatgpt.com
laptop: 0.7806
use: 0.3886
review: 0.2838
best: 0.2129
macbook: 0.2129

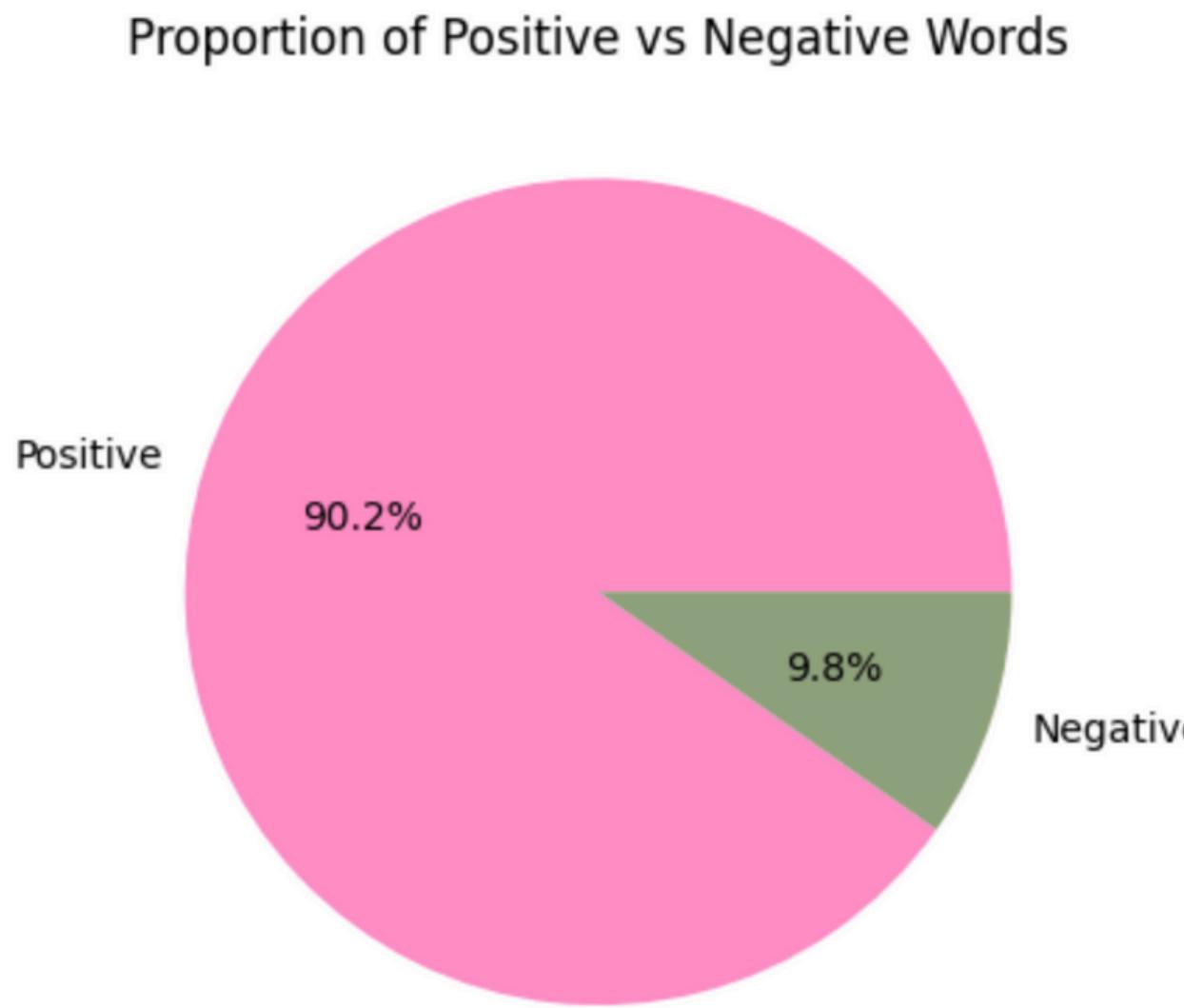
Top 5 TF-IDF words for https://forum.freecodecamp.org/t/best-laptop-for-programming-why-review-sites-got-it-wrong/762870?utm_source=chatgpt.com



Problem 3

สัดส่วนของคำประเทบทวิคบอร์ด (positive/negative words)

```
1 sia = SentimentIntensityAnalyzer()  
2 pos_words = []  
3 neg_words = []  
4 for w in tokens:  
5     sc = sia.polarity_scores(w) ['compound']  
6     if sc > 0.3: pos_words.append(w)  
7     elif sc < -0.3: neg_words.append(w)
```



- ถ้าค่าคะแนนรวม $> 0.3 \rightarrow$ ถือว่า เป็นคำบวก → เก็บใน pos_words
- ถ้า $< -0.3 \rightarrow$ ถือว่า เป็นคำลบ → เก็บใน neg_words

word could ของ positive words และ negative words



Problem 3

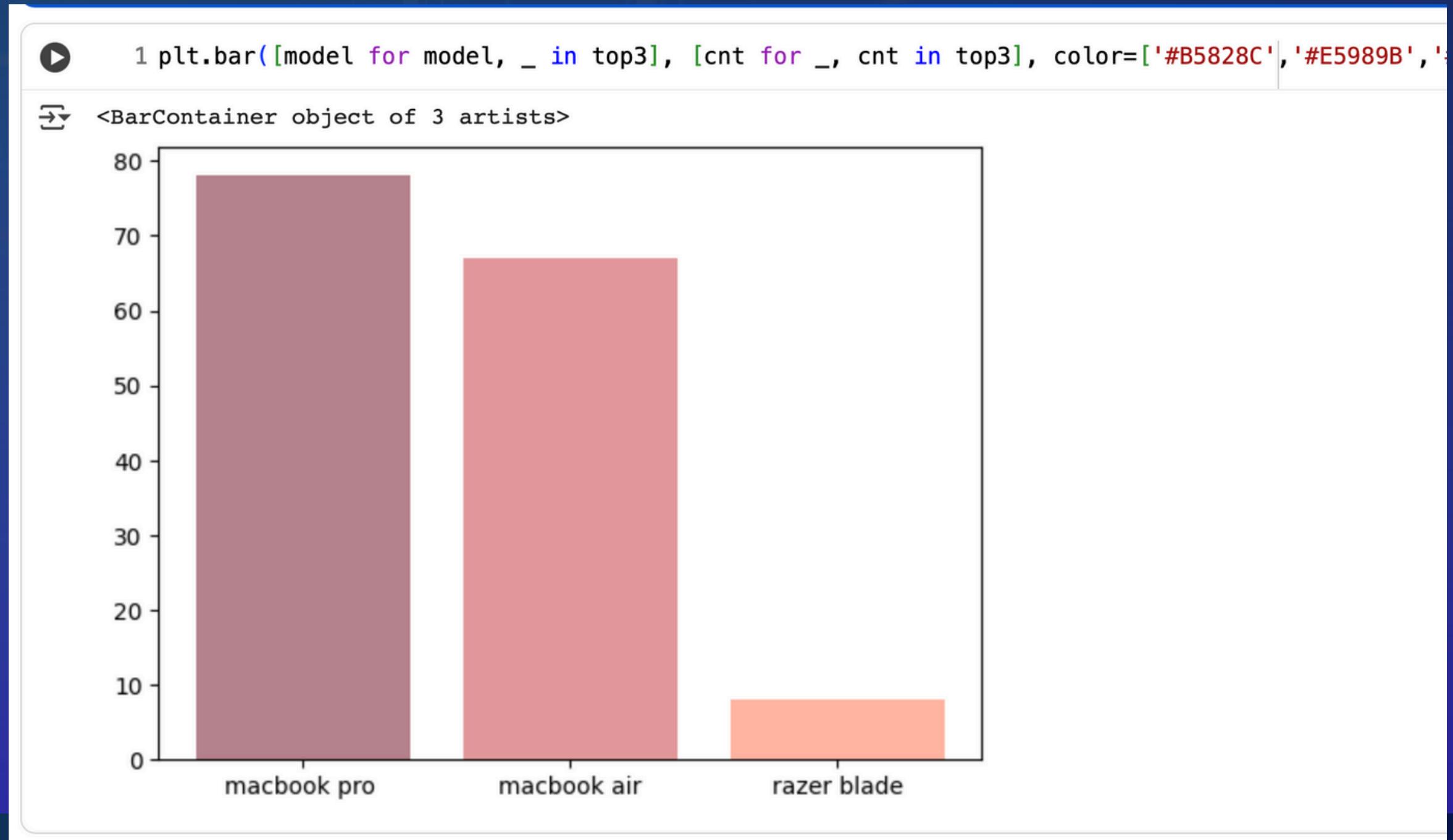
Top 3 laptops ที่ถูกพูดถึงมากที่สุด

```
models = [  
    "macbook air",  
    "macbook pro ",  
    "dell xps 13",  
    "asus rog zephyrus g14",  
    "lenovo thinkpad t14",  
    "hp spectre x360",  
    "acer nitro",  
    "razer blade",  
    "surface laptop",  
]
```

```
0  
1 for model in models:  
2     # Escape model for regex, ensure word boundaries  
3     import re  
4     pattern = r'\b' + re.escape(model) + r'\b'  
5     count = len(re.findall(pattern, text))  
6     model_counts[model] = count  
7
```

วนลูปตรวจสอบแต่ละรุ่น

- ใช้ regular expression (regex) เพื่อค้นหาคำแบบ “ตรงเป๊ะ”
- \b = word boundary ป้องกันการนับคำที่เป็นส่วนของคำอื่น
- เช่น คำว่า macbook จะไม่ไปแมตช์ใน macbookairplane
- re.escape(model) → ป้องกัน error ถ้ามีอักษรพิเศษในชื่อรุ่น
- re.findall(pattern, text) → คืน list ของการ match ทั้งหมด
- len(...) → นับจำนวนครั้งที่เจอ และบันทึกลงใน model_counts



สมาชิกกลุ่ม

1. นายวัฒนศักดิ์ คลังแสง 663020039-3
2. นางสาวศศิวิมล ภาณุโชค 663020040-8
3. นายภัทรswa วรรณยศ 663020287-4
4. นายวชากร สุขเกษณ 663020293-9



Thank You!