

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import re
```

```
!pip install requests
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (2.32.4)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests) (2025.10.5)
```

```
!pip install spacy
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.8)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.13)
Requirement already satisfied: cymem<1.0.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.11)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.10)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.8)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.2)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.20.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.1)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
Requirement already satisfied: pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.11.10)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (2.33.2)
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (4.15.0)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (0.4.2)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2025.10.5)
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (1.3.0)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (0.1.5)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spacy) (8.3.0)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy) (0.23.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy) (7.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.4.2->spacy) (2.0.1)
```

```
!python - m spacy download en_core_web_sm
```

```
Python 3.12.12 (main, Oct 10 2025, 08:52:57) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
KeyboardInterrupt
>>> ^C
```

```
import nltk
import spacy
```

```
def clean_text(text):
```

```
    text = text.lower() # Coverts all the characters in the text to lowercase
    text = re.sub(r'http\S+|www\S+|https\S+', '', text) # Removes URL
    text = re.sub(r'<.*?>', '', text) # Removes HTML tags (ex: <div>, <p>, etc)
    text = re.sub(r'[^\a-z\s]', '', text) # Removes numbers, punctuations and special characters
    text = re.sub(r'\s+', ' ', text).strip() # Removes leading/trailing and multiple spaces
    text = re.sub(r'\S+@\S+\.\S+', '', text) # Removes email addresses
    text = re.sub(r'[^\w\s]', '', text) # Removes punctuations (non word characters)
    text = re.sub(r'(\.|\{2,}|r\'\\1', text) # Removes repeated characters to normalise enlogated words
    text = re.sub(r'#', '', text) # Removes hashtag symbols
    text = re.sub(r'@\w+', '', text) # Removes user mentions
    text = re.sub(r'[^\x00-\x7F]+', '', text) # Removes non ASCII characters (ex: emojis and foreign characters)
```

```
    return text
```

```
# Word Lemmatization using Spacy
```

```
nlp = spacy.load('en_core_web_sm')
```

```
def clean_text_spacy(text):
```

```
    doc = nlp(text)
    tokens = []

    for i in doc:
        if not i.is_stop and not i.is_punct:
            tokens.append(i.lemma_)
```

```
    return tokens
```

```
text = """"
OMG!!! 🤔🤔🤔 I can't believe it's already 2025... time flies!! 🕒
btw did u see the new AI thingy by openai?? sooo coool 🤖🔥🔥🔥
like... idk if it's gna take over the world or what lol 🤖
but tbh ppl r freaking out for no reason (smh).
also, here's a link: https://bit.ly/3fakeURL and my email is test_email1123@domain.com
#AI #Future #omg
""""
```

```
cleaned_text = clean_text(text)
print(cleaned_text)

omg i cant believe its already time flies btw did u see the new ai thingy by openai so col like idk if its gna take over the world or what lol but tbh ppl r freaking out for no reason smh also heres a link and my email is testemai domaincom ai future omg

clean_text_spacy(cleaned_text)

['omg',
 'not',
 'believe',
 'time',
 'fly',
 'btw',
 'u',
 'new',
 'ai',
 'thingy',
 'openai',
 'col',
 'like',
 'idk',
 'gna',
 'world',
 'lol',
 'tbh',
 'ppl',
 'r',
 'freak',
 'reason',
 'smh',
 'here',
 'link',
 'email',
 'testemai domaincom',
 'ai',
 'future',
 'omg']
```

Word Cloud

```
from wordcloud import WordCloud

review = """
ok sooo i've had the iphone 17 pro for like 2 weeks now and ngl idk how to feel abt it 🥰. the camera is insane fr, like the pics are so crisp it makes my old 13 look like a potato 🥔😂.

BUT bruh... the price 🤔🤔🤔 like why tf is this thing so expensive??? i sold my kidney (jk... kinda) just to upgrade. battery's okish, but not the '2-day beast' apple promised smh. if u game a bit or scroll tiktok for too long, it drops faster than my grades lol.

the new titanium body feels premium, yeah, but also mad slippery?? i already dropped it once (rip screen protector). the phone's hell a heavy too – like u could do bicep curls with it 🤏.

also idk wtf they did with iOS 19 but it keeps glitching out whenever i switch between apps. sometimes the brightness randomly jumps up like it's tryna burn my retinas 🥵.

ngl the cameras slap tho – that zoom is wild, i took a pic of the moon and it looked legit 🌌🌕. but still, half the features feel the same as the 16 pro max. apple really just added 'Pro-Motion-Ultra-Mega' and called it innovation 🤔🤔

oh and the AI "Smart Assist" thing?? kinda creepy tbh. it started suggesting msgs before i even finished typing... like chill siri i got it 😏.

so yeah overall: love the camera + design, hate the price, battery meh, and it def ain't worth the hype unless ur a hardcore apple stan 🍏. 7.5/10 maybe? 🤖
"""

clean_text(review)

'ok so ive had the iphone pro for like weeks now and ngl idk how to feel abt it the camera is insane fr like the pics are so crisp it makes my old look like a potato but bruh the price like why tf is this thing so expensive i sold my kidney jk kinda just to upgrade batterys okish but not the day beast apple promised smh if u game a b
it or scroll tiktok for too long it drops faster than my grades lol the new titanium body feels premium yeah but also mad slippery i already dropped it once rip screen protector the phones hell a heavy too like u could do bicep curls with it also idk wtf they did with ios but it keeps glitching out whenever i switch between apps somet
imes the brightness randomly jumps up like its tryna burn my retinas ngl the cameras slap tho that zoom is wild i took a pic of the moon and it looked legit but still half the features feel the same as the pro max apple really just added promotionultramega and called it innovation oh and the ai smart assist thing kinda creepy tbh...'

cleaned_review = clean_text_spacy(clean_text(review))

cleaned_review
```

```
year',
'overall',
'love',
'camera',
'design',
'hate',
'price',
'battery',
'meh',
'def',
'be',
'not',
'worth',
'hype',
'ur',
'hardcore',
'apple',
'stan',
'maybe']
```

```
len(cleaned_review)
```

```
139
```

```
def wordcloud(text):
    try:
        wordcloud = WordCloud(background_color='white', width=800, height=400).generate(text)
        plt.figure(figsize=(10, 5))
        plt.imshow(wordcloud, interpolation='bilinear')
        plt.axis('off')
        plt.show()
    except Exception as e:
        print(f'Error: {e}')
```

```
wordcloud(review)
```



▼ N-GRAM Analysis

```
from nltk.util import ngrams
from collections import Counter
import plotly.graph_objects as go
```

```
# Creating chart for n-gram analysis

def ngram_chart(tokens, n, top_n):
    try:
        ngram = list(ngrams(tokens, n))
        ngram_count = Counter(ngram).most_common(top_n)

        if not ngram_count:
            raise ValueError(f'No {n}-grams found')

        labels = []
        counts = []

        for i, j in ngram_count:
            labels.append(" ".join(i))
            counts.append(j)

        fig = go.Figure(data=[go.Bar(x=labels, y=counts, text=counts, textposition='outside')])

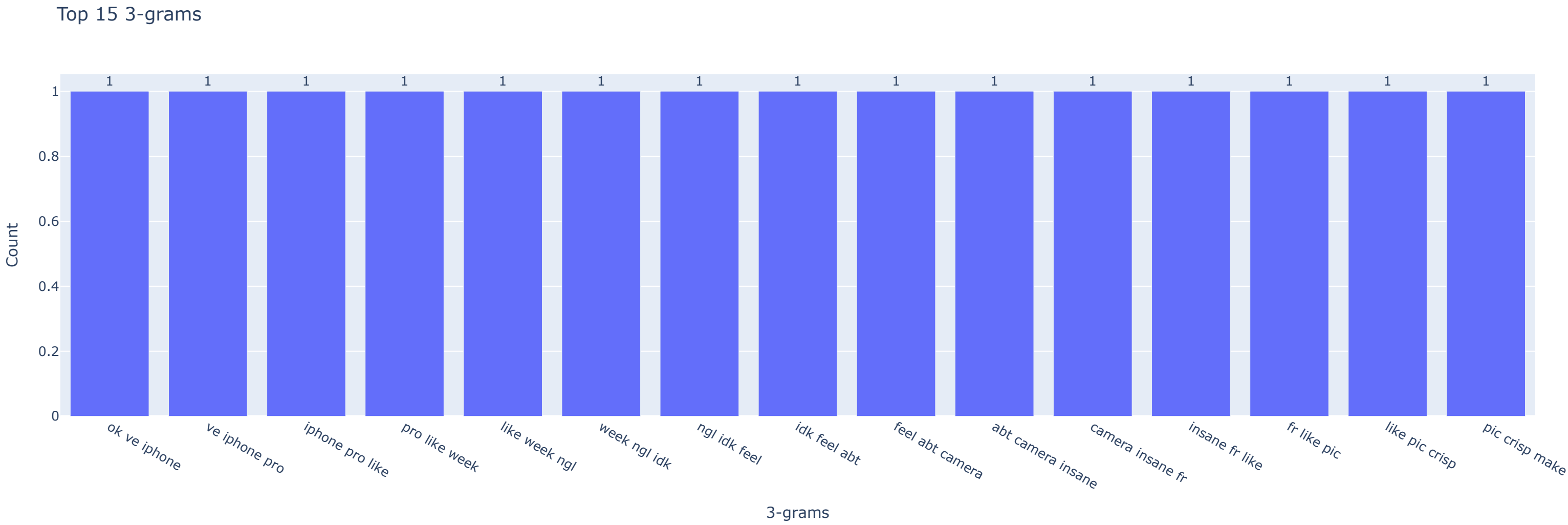
        fig.update_layout(
            title = f'Top {top_n} {n}-grams',
            xaxis_title = f'{n}-grams',
            yaxis_title = 'Count',
        )

        fig.show()

    except Exception as e:
        print(f'Error: {e}')
```

```
ngram_chart(cleaned_review, 3, 15)
```





Emotion Detection

```
!pip install transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.36.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.22.1)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.6.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.34.0->transformers) (2025.3.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.34.0->transformers) (4.15.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.34.0->transformers) (1.2.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (2025.10.5)
```

```
from transformers import pipeline
```

```
model_name = "nateraw/bert-base-uncased-emotion"
```

```
emotion_classifier = pipeline("text-classification", model=model_name, tokenizer=model_name, top_k = 5)
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
```

The secret 'HF\_TOKEN' does not exist in your Colab secrets.  
To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as secret in your Google Colab and restart your session.  
You will be able to reuse this secret in all of your notebooks.  
Please note that authentication is recommended but still optional to access public models or datasets.

```
config.json: 100% 768/768 [00:00<00:00, 19.4kB/s]
pytorch_model.bin: 100% 438M/438M [00:04<00:00, 239MB/s]
model.safetensors: 100% 438M/438M [00:06<00:00, 59.0MB/s]
tokenizer_config.json: 100% 252/252 [00:00<00:00, 7.19kB/s]
vocab.txt: 232k/? [00:00<00:00, 655kB/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 1.01kB/s]
Device set to use cpu
```

```
text = "I can't believe how amazing this phone is! I love it so much!"
results = emotion_classifier(text)
```

```
results[0]
```

```
[{'label': 'joy', 'score': 0.8282867074012756},
 {'label': 'surprise', 'score': 0.14573289453983307},
 {'label': 'love', 'score': 0.010380861349403858},
 {'label': 'fear', 'score': 0.008571169339120388},
 {'label': 'anger', 'score': 0.004593316465616226}]
```

```
# Creating Emotion Detection Chart
```

```
def detect_emotion(text):
```

```
    try:
        results = emotion_classifier(text)
        emotion_data = [{'Emotion':i['label'], 'Confidence':i['score']} for i in results[0]]
        df = pd.DataFrame(emotion_data)
```

```
        # Highest Scoring Emotion
        highest_emotion = df.loc[df['Confidence'].idxmax()]
```

```
        fig = px.bar(df, x='Emotion', y='Confidence', color='Emotion', title=f'Top 5 Emotion')
        fig.update_layout(showlegend=False)
```

```
    return{
```

```
'Text':text,
'Emotion':highest_emotion['Emotion'],
'Confidence':highest_emotion['Confidence'],
'Emotion Table':df,
'Plot':fig
}

except Exception as e:
    print(f'Error: {e}')
```

```
emotion_result = detect_emotion(review)
```

```
if 'error' not in emotion_result:
    print(f'''Text: {emotion_result['Text']}''')
    print('Emotion Table')
    print(emotion_result['Emotion Table'])
    print(f'Detected Emotion: {emotion_result['Emotion']} with Confidence: {emotion_result['Confidence']}')
    emotion_result['Plot'].show()
```

Text:  
ok sooo i've had the iphone 17 pro for like 2 weeks now and ngl idk how to feel abt it 🥰. the camera is insane fr, like the pics are so crisp it makes my old 13 look like a potato 🥔😓.

BUT bruh... the price 😭😭😭 like why tf is this thing so expensive??? i sold my kidney (jk... kinda) just to upgrade. battery's okish, but not the '2-day beast' apple promised smh. if u game a bit or scroll tiktok for too long, it drops faster than my grades lol.

the new titanium body feels premium, yeah, but also mad slippery?? i already dropped it once (rip screen protector). the phone's hellla heavy too – like u could do bicep curls with it 🤏.

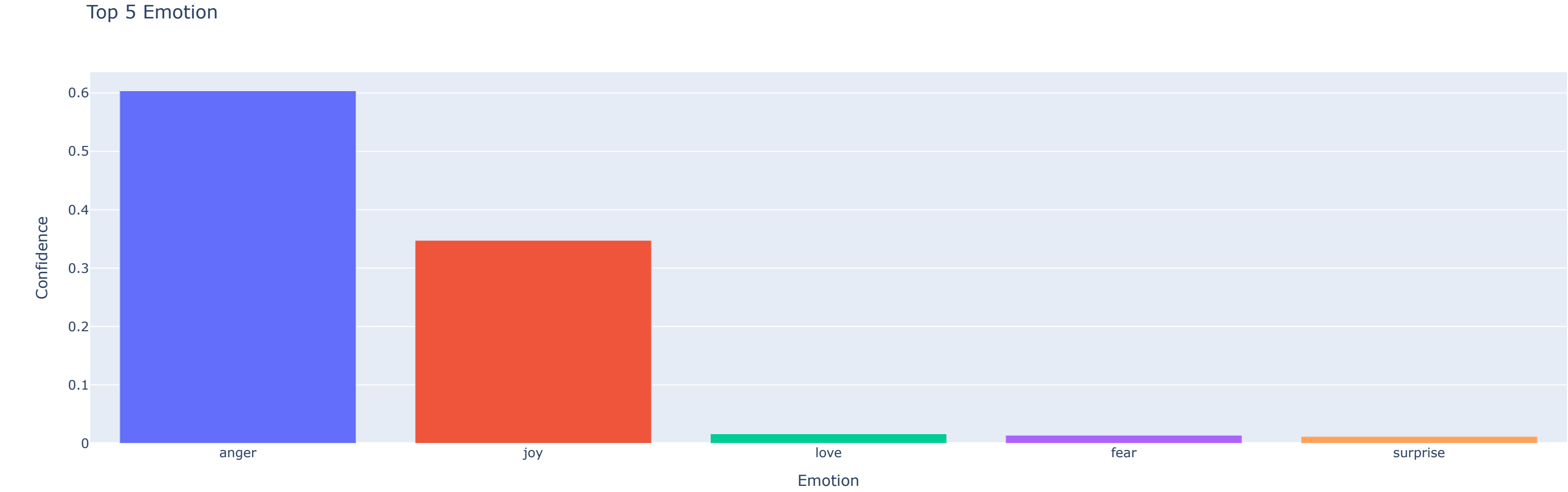
also idk wtf they did with iOS 19 but it keeps glitching out whenever i switch between apps. sometimes the brightness randomly jumps up like it's tryna burn my retinas 🤯.

ngl the cameras slap tho – that zoom is wild, i took a pic of the moon and it looked legit 🌌🌕. but still, half the features feel the same as the 16 pro max. apple really just added 'Pro-Motion-Ultra-Mega' and called it innovation 🤔🤔

oh and the AI "Smart Assist" thing?? kinda creepy tbh. it started suggesting msgs before i even finished typing... like chill siri i got it 😏.

so yeah overall: love the camera + design, hate the price, battery meh, and it def ain't worth the hype unless ur a hardcore apple stan 🍏. 7.5/10 maybe? 🙄

```
Emotion Table
Emotion  Confidence
0      anger    0.603640
1       joy    0.347409
2       love    0.016051
3       fear    0.013613
4  surprise    0.011466
Detected Emotion: anger with Confidence: 0.6036401391029358
```



## Creating Chunks

```
# Function to create chunks

def create_chunks(text, max_length = 500):

    nlp = spacy.load('en_core_web_sm')
    doc = nlp(text)
    chunks = []
    current_chunk = ''

    for i in doc.sents:
        sentence = i.text.strip()

        if len(current_chunk) + len(sentence) <= 500:
            current_chunk += ' '+sentence
        else:
            chunks.append(current_chunk.strip())
            current_chunk = sentence

    if current_chunk:
        chunks.append(current_chunk.strip())

    return chunks
```

```
# Creating function for emotion detection

model_name = "nateraw/bert-base-uncased-emotion"
emotion_classifier = pipeline("text-classification", model=model_name, tokenizer=model_name, top_k = None)

from collections import defaultdict

def detect_chunk_emotion(text):
```

```
chunks = create_chunks(text)
emotions_total = {}
emotions_count = {}
emotions_count = defaultdict(int)

for i in chunks:
    result = emotion_classifier(i)[0]

    for i in result:
        label = i['label']
        score = i['score']
        emotions_total[label] = emotions_total.get(label, 0) + score
        emotions_count[label] += 1

    emotion_count = dict(emotions_count)

emotions_average = {i: emotions_total[i]/emotions_count[i] for i in emotions_total}
sorted_emotions = sorted(emotions_average.items(), key=lambda x: x[1], reverse = True)
top_emotions = sorted_emotions[:5]
df = pd.DataFrame(top_emotions, columns=['Emotion', 'Confidence'])

return df
```

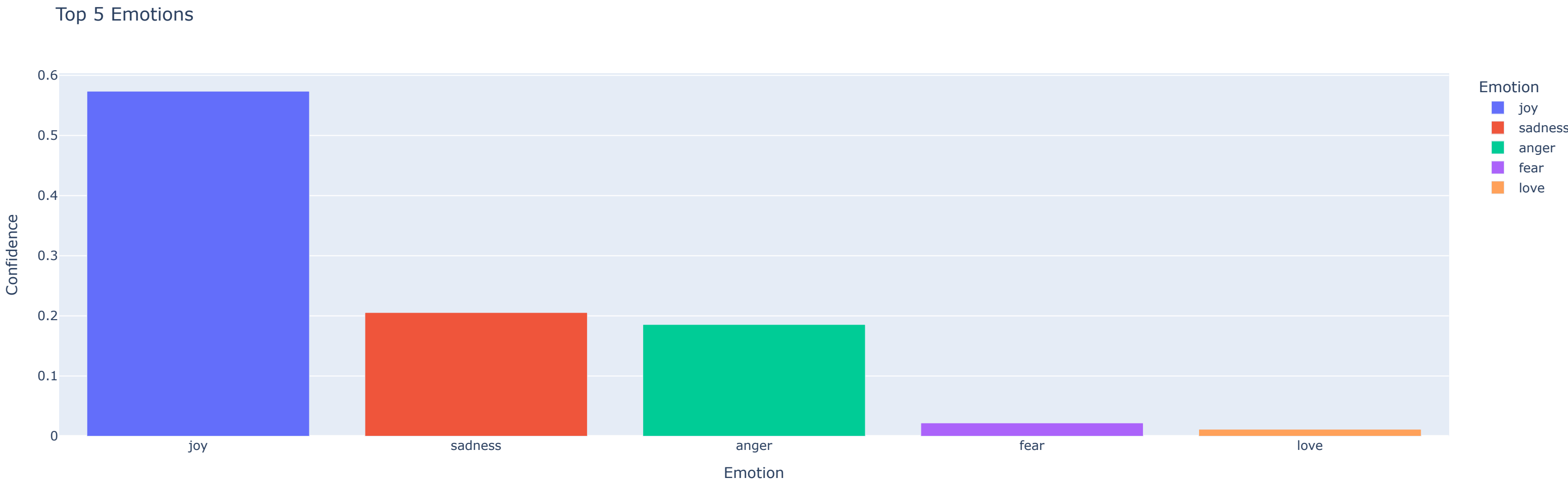
Device set to use cpu

news = """"  
While addressing the Knesset, the Israeli Parliament, U.S. President Donald Trump declared that the ceasefire agreement between Hamas and Israel, which he had helped broker, marked “the historical dawn of the new Middle East”. In Sharm el-Sheikh, Egypt, Arab and Muslim leaders joined him in praising his 20-point peace plan. While all  
  
Mr. Trump later claimed that Hamas had promised to disarm, warning that “If they don’t disarm, we will disarm them..., perhaps violently”. But Israel, despite its two years in Gaza, has still failed to disarm Hamas. How then does Mr. Trump plan to disarm them? He also claimed that the Iran threat had been neutralised, and urged more  
""""

```
result = detect_chunk_emotion(news)
```

```
max_index = result['Confidence'].idxmax()
Emotion = result.loc[max_index, 'Emotion']
Score = result.loc[max_index, 'Confidence']
print(f"Predicted Emotion: {Emotion} with {round(Score*100)}% Confidence")
fig = px.bar(result, x='Emotion', y='Confidence', color='Emotion', title='Top 5 Emotions')
fig.show()
```

Predicted Emotion: joy with 57% Confidence



SENTIMENTAL ANALYSIS

```
model_name= "cardiffnlp/twitter-roberta-base-sentiment"
sentiment_classifier= pipeline("sentiment-analysis", model= model_name, tokenizer= model_name , return_all_scores=True)
```

```
config.json: 100% 747/747 [00:00<00:00, 42.9kB/s]
pytorch_model.bin: 100% 499M/499M [00:03<00:00, 260MB/s]
model.safetensors: 100% 499M/499M [00:06<00:00, 43.5MB/s]
vocab.json: 899k/? [00:00<00:00, 18.6MB/s]
merges.txt: 456k/? [00:00<00:00, 5.85MB/s]
special_tokens_map.json: 100% 150/150 [00:00<00:00, 4.00kB/s]
Device set to use cpu
/usr/local/lib/python3.12/dist-packages/transformers/pipelines/text_classification.py:111: UserWarning:
`return_all_scores` is now deprecated, if want a similar functionality use `top_k=None` instead of `return_all_scores=True` or `top_k=1` instead of `return_all_scores=False`.
```

```
# Creating Function for Sentiment Analysis
def sentiment_analysis(text):
    try:

        sentiment_labels = {'LABEL_0': 'Negative', 'LABEL_1': 'Neutral', 'LABEL_2': 'Positive'}

        chunks = create_chunks(text)
        total_score = {'Negative': 0, 'Neutral':0, 'Positive':0}
        chunk_count = len(chunks)

        for i in chunks:
```

```

    result = sentiment_classifier(i)[0]

    for res in result:
        label = sentiment_labels[res['label']]
        total_score[label] += res['score']

average_score = {}

for j in total_score:
    average_score[j] = total_score[j]/chunk_count

overall_sentiment = max(average_score, key=average_score.get)

return {
    'Overall Sentiment':overall_sentiment,
    'Average Score':average_score
}

except Exception as e:
    print(f'Error: {e}')
```

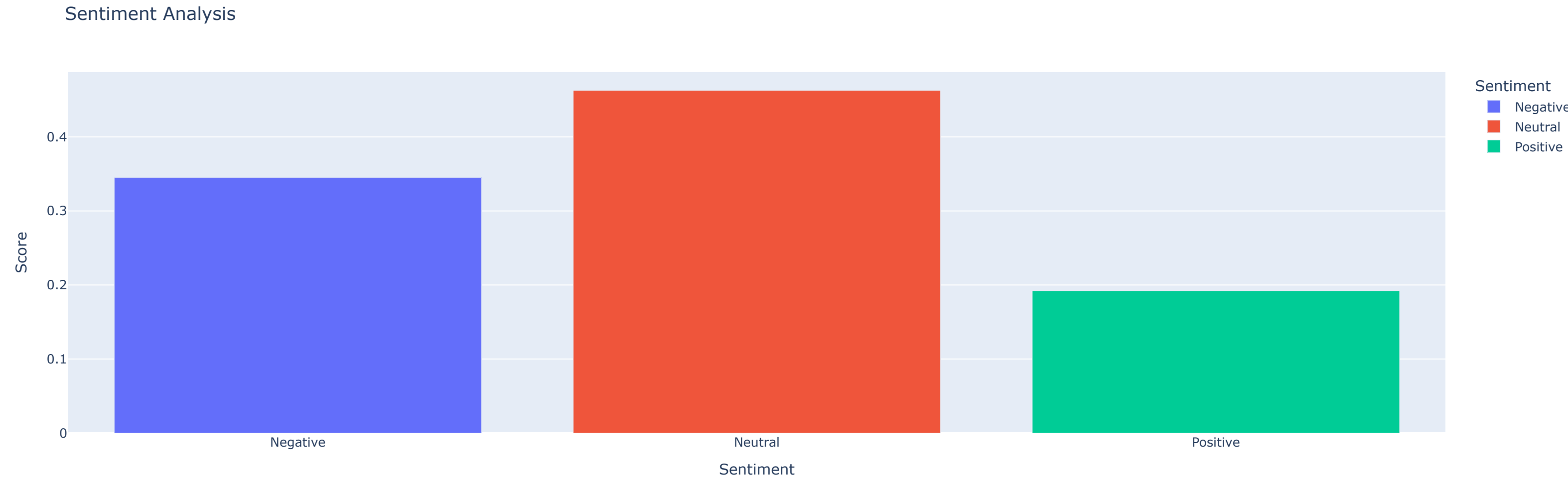
```
result = sentiment_analysis(news)
```

```

if 'error' not in result:
    print(f"Overall Sentiment: {result['Overall Sentiment']} with {max(result['Average Score'].values())} Score")
    df = pd.DataFrame(result['Average Score'].items(), columns=['Sentiment', 'Score'])
    print(df)
    fig = px.bar(df, x='Sentiment', y='Score', color='Sentiment', title='Sentiment Analysis')
    fig.show()
```

```

Overall Sentiment: Neutral with 0.462871858051845 Score
Sentiment    Score
0  Negative    0.345045
1   Neutral    0.462872
2   Positive    0.192083
```



## ▼ TONE OF SPEECH DETECTION

```
classifier= pipeline("zero-shot-classification",model="facebook/bart-large-mnli")
```

```

config.json:      1.15k/? [00:00<00:00, 88.7kB/s]
model.safetensors: 100%                               1.63G/1.63G [03:43<00:00, 449kB/s]
tokenizer_config.json: 100%                             26.0/26.0 [00:00<00:00, 2.69kB/s]
vocab.json:       899k/? [00:00<00:00, 46.8MB/s]
merges.txt:       456k/? [00:00<00:00, 24.5MB/s]
tokenizer.json:   1.36M/? [00:00<00:00, 46.3MB/s]
Device set to use cpu
```

```

tone= """"
Climate Change and Technology: An Informative Report
```

Climate change represents one of the most pressing global challenges of the 21st century. Driven largely by the emission of greenhouse gases through industrial activities, deforestation, and fossil fuel consumption, the Earth’s temperature is rising at an unprecedented rate. This warming is leading to erratic weather patterns, rising sea levels, and threats to biodiversity. Historically, industrial revolutions and technological advancements have been linked to increased carbon emissions and environmental degradation. The combustion engine, coal-based power generation, and mass production all significantly contributed to the current levels of atmospheric CO2. Technological development, when pursued without regard for environmental consequences, has accelerated these trends. One of the most prominent ways technology is addressing climate change is through the development of renewable energy. Solar panels, wind turbines, and hydroelectric systems now provide cleaner alternatives to coal and oil. Technological advances have made these sources more efficient, cost-effective, and scalable. Smart grids powered by AI optimize energy distribution, while electric vehicles (EVs) reduce reliance on fossil fuels. Agriculture is a significant contributor to greenhouse gas emissions, particularly methane and nitrous oxide. Technology is helping transform this sector through smart agriculture, which uses sensors, satellite data, and AI to monitor crop health, optimize irrigation, and reduce fertilizer use. Vertical farming and lab-grown meat are innovative approaches to reduce land use and emissions. Another technological response to climate change involves carbon capture and storage (CCS). These systems extract CO2 from the air or directly from industrial emissions and store it underground or repurpose it for industrial use. While CCS is still in its early stages, it holds promise as a means to directly remove carbon from the atmosphere. The transportation sector is undergoing a technological revolution in response to climate concerns. Electric vehicles (EVs), powered by renewable electricity, are replacing traditional gasoline and diesel vehicles. Advancements in battery technology have significantly extended the range and reduced the cost of EVs. Additionally, public transit systems are being optimized with AI to improve efficiency and reduce congestion. From renewable energy to smart agriculture, green transportation to carbon capture, technology is playing a pivotal role in shaping a more sustainable and resilient future. However, innovation must be guided by ethical considerations, inclusivity, and sustainability to ensure that technological progress does not come at the cost of environmental degradation or social inequality.

```

LABELS = ["Factual", "Opinion", "Question", "Command", "Angry", "Sadness", "Surprise", "Persuasive", "Humorous", "Sarcastic",
          "Suggestion", "Prediction", "Warning", "Narrative", "Argument", "Confident", "Doubtful", "Inappropriate", "Information"]
```

```
nlp = spacy.load("en_core_web_sm")
```



```
classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli")

def classify_text(text,
                  labels= LABELS,
                  chunk_size=5,
                  overlap=2):

    if not isinstance(text, str):
        raise ValueError("text must be a string")

    doc = nlp(text)

    # Split into sentences
    sentences = [sent.text.strip() for sent in doc.sents]
    if not sentences:
        return {"Text": text, "Predicted_Category": None, "Score": 0.0, "All_Categories": []}

    # If short enough, classify whole text at once
    if len(sentences) <= chunk_size:
        result = classifier(text, candidate_labels=labels)
        all_categories = list(zip(result["labels"], result["scores"]))
        return {
            "Text": text,
            "Predicted_Category": result["labels"][0],
            "Score": float(result["scores"][0]),
            "All_Categories": all_categories
        }

    # Build overlapping chunks (each chunk is a list of sentences)
    chunks = []
    start = 0
    while start < len(sentences):
        end = start + chunk_size
        chunk = sentences[start:end]
        chunks.append(chunk)
        if end >= len(sentences):
            break
        start += (chunk_size - overlap)

    # Aggregate scores across chunks
    total_scores = {label: 0.0 for label in labels}
    for chunk in chunks:
        chunk_text = " ".join(chunk)
        result = classifier(chunk_text, candidate_labels=labels)

        for label, score in zip(result["labels"], result["scores"]):
            total_scores[label] += float(score)

    # Average and sorting
    num_chunks = len(chunks)
    avg_scores = {label: (total_scores[label] / num_chunks) for label in labels}
    sorted_categories = sorted(avg_scores.items(), key=lambda x: x[1], reverse=True)
    top_label, top_score = sorted_categories[0]

    return {
        "Text": text,
        "Predicted_Category": top_label,
        "Score": float(top_score),
        "All_Categories": sorted_categories
    }
```

Device set to use cpu

classify\_text(tone)

```
{'Text': '\nClimate Change and Technology: An Informative Report\n\nClimate change represents one of the most pressing global challenges of the 21st century. Driven largely by the emission of greenhouse gases through industrial activities, deforestation, and fossil fuel consumption, the Earth’s temperature is rising at an unprecedented rate. This warming is leading to erratic weather patterns, rising sea levels, and a surge in climate-related natural disasters. In the face of such challenges, technology emerges as both a contributor to the problem and a powerful tool for the solution. This report explores how technology is being harnessed to combat climate change, reduce emissions, adapt to new environmental realities, and build a sustainable future.\n\nHistorically, industrial revolutions and technological advancements have been linked to increased carbon emissions and environmental degradation. The combustion engine, coal-based power generation, and mass production all significantly contributed to the current levels of atmospheric CO2. Technological development, when pursued without consideration for ecological balance, has caused deforestation, pollution, and over-reliance on non-renewable resources. However, recognizing this history is essential to better redirect innovation toward green and sustainable alternatives.\n\nOne of the most prominent ways technology is addressing climate change is through the development of renewable energy. Solar panels, wind turbines, and hydroelectric systems now provide cleaner alternatives to coal and oil. Technological advances have made these sources more efficient, cost-effective, and scalable. Smart grids powered by machine learning optimize electricity distribution, reduce waste, and integrate renewable energy into national grids more seamlessly. Battery storage technology has also seen rapid innovation, allowing excess energy to be stored and used when renewable sources are not generating power.\n\nAgriculture is a significant contributor to greenhouse gas emissions, particularly methane and nitrous oxide. Technology is helping transform this sector through smart agriculture, which uses sensors, satellite data, and AI to monitor crop health, optimize irrigation, and reduce fertilizer use. Vertical farming and lab-grown meat are examples of how technology is creating sustainable food systems that require less land and emit fewer greenhouse gases. These innovations aim to increase food production efficiency while minimizing environmental impact.\n\nAnother technological response to climate change involves carbon capture and storage (CCS). These systems extract CO2 from the air or directly from industrial emissions and store it underground or repurpose it for industrial use. While CCS is still in its early stages, it holds promise as a means to directly remove carbon from the atmosphere. Geoengineering, including solar radiation management and ocean fertilization, is more controversial but continues to be researched as a potential last resort. Ethical considerations and long-term risks must be thoroughly evaluated before large-scale deployment.\n\nThe transportation sector is undergoing a technological revolution in response to climate concerns. Electric vehicles (EVs), powered by renewable electricity, are replacing traditional gasoline and diesel vehicles. Advancements in battery technology have significantly extended the range and reduced the cost of EVs. Additionally, public transportation systems are integrating clean energy buses and trains. Innovations in mobility-as-a-service (MaaS), carpooling platforms, and smart traffic systems also contribute to reduced emissions and improved urban planning.\n\nFrom renewable energy to smart agriculture, green transportation to carbon capture, technology is playing a pivotal role in shaping a more sustainable and resilient future. However, innovation must be guided by ethical considerations, inclusivity, and sustainability to ensure that technological progress does not come at the cost of environmental degradation. With global collaboration, policy support, and continued investment, technology can be harnessed to not only address climate change but to build a healthier planet for future generations.',
'Predicted_Category': 'Question',
'Score': 0.1442592039704323,
'All_Categories': [('Question', 0.1442592039704323),
('Persuasive', 0.11529204249382019),
('Suggestion', 0.09527212381362915),
('Information', 0.07762827910482883),
('Narrative', 0.07645975910127163),
('Warning', 0.05764712020754814),
('Opinion', 0.054705016687512396),
('Confident', 0.052377103827893734),
('Argument', 0.051533482223749164),
('Factual', 0.042704323679208754),
('Prediction', 0.04267776422202587),
('Sarcastic', 0.036016089282929896),
('Doubtful', 0.03431627433747053),
('Command', 0.030942018516361712),
('Surprise', 0.025612798891961573),
('Inappropriate', 0.02180404020473361),
('Angry', 0.014910202473402023),
('Humorous', 0.014071317669004203),
('Sadness', 0.01177105917595327)]})
```

▼ SUMMARY GENERATION

summarizer= pipeline("summarization",model="facebook/bart-large-cnn")



config.json: 1.58k/? [00:00<00:00, 94.3kB/s]

model.safetensors: 100% 1.63G/1.63G [00:29<00:00, 67.1MB/s]

generation\_config.json: 100% 363/363 [00:00<00:00, 37.8kB/s]

vocab.json: 899k/? [00:00<00:00, 11.2MB/s]

merges.txt: 456k/? [00:00<00:00, 27.2MB/s]

tokenizer.json: 1.36M/? [00:00<00:00, 42.7MB/s]

Device set to use cpu

#Creating Function for Text Summarization

```
def summarize(text):
    summarizer = pipeline('summarization', model='facebook/bart-large-cnn')
    chunks = create_chunks(text, max_length=500)

    # Summarize each chunk
    chunk_summaries = []
    for i in chunks:
        input_length = len(i.split())
        max_summary_length = int(input_length * 0.8)
        min_summary_length = int(input_length * 0.3)
        summary = summarizer(i, max_length=max_summary_length, min_length=min_summary_length, do_sample=False)[0]["summary_text"]
        chunk_summaries.append(summary)

    # Combining summaries of all the chunks
    combined_summary = " ".join(chunk_summaries)

    # Final Summary of combined summary

    input_length = len(combined_summary.split())
    max_summary_length = int(input_length * 0.8)
    min_summary_length = int(input_length * 0.3)
    final_summary = summarizer(combined_summary, max_length=max_summary_length, min_length=min_summary_length, do_sample=False)[0]["summary_text"]

    return final_summary
```

long\_text = """"

Artificial intelligence (AI) has become one of the most transformative technologies of the 21st century. It is now impacting virtually every industry, from healthcare to transportation, finance, education, manufacturing, entertainment, and more. By automating tasks that previously required human intelligence, AI is improving operatio

In healthcare, AI-powered tools assist doctors and medical professionals with diagnosis, personalized treatment recommendations, patient monitoring, and even robotic surgeries. These innovations have the potential to significantly improve patient outcomes while simultaneously reducing costs. AI can analyze vast amounts of medical dat

Financial institutions are leveraging AI for fraud detection, credit scoring, personalized banking, and algorithmic trading, allowing for more accurate risk assessment and better investment decisions. AI-driven automation streamlines customer service with chatbots handling routine inquiries 24/7, freeing up human agents for more comp

The transportation industry is undergoing a significant transformation thanks to AI. The rise of autonomous vehicles promises to reduce accidents caused by human error and enhance traffic management. AI-enabled smart traffic lights dynamically adjust their timing based on real-time traffic data to reduce congestion, pollution, and tr

Education is also being reshaped through AI-driven personalized learning platforms that adapt content and pace based on individual student needs and learning styles. AI tutors and virtual assistants provide additional help outside classrooms, enabling students to learn at their own pace and receive instant feedback. Data analytics po

The entertainment industry leverages AI to create personalized content recommendations on streaming platforms, improving user engagement by tailoring suggestions to individual preferences. AI-generated content, including music, art, and even movie scripts, is increasingly sophisticated, sparking new debates about creativity and intel

Despite these advances, the rapid development of AI raises important ethical and societal questions that require careful consideration. Concerns include job displacement due to automation, data privacy issues, and algorithmic bias that can perpetuate or amplify social inequalities. As AI systems make more decisions affecting people’s

Ensuring responsible AI development requires transparency, fairness, and accountability from developers, companies, and policymakers alike. International cooperation is necessary to set standards and norms that promote ethical AI while fostering innovation and competition. Public awareness and education about AI’s capabilities and li

As AI continues to evolve rapidly, collaboration between technology experts, industry leaders, governments, academia, and civil society will be essential to harness its full potential. By investing in research, infrastructure, and education, society can leverage AI to solve complex problems such as climate change, healthcare accessib

In summary, AI stands as a powerful tool with immense possibilities, poised to reshape our world in profound ways. However, it must be guided by ethical principles and responsible governance to ensure it creates a better future for all. The balance between innovation and responsibility will define the path AI takes in the coming deca

""""

len(long\_text)

7115

summarize(long\_text)

Device set to use cpu

'Artificial intelligence (AI) has become one of the most transformative technologies of the 21st century. It is now impacting virtually every industry, from healthcare to transportation, finance, education, manufacturing, entertainment, and more. Despite these advances, the rapid development of AI raises important ethical and societal questions that require careful consideration. The balance between innovation and responsibility will define the path AI takes in the coming decades. By embracing AI thoughtfully and inclusively, humanity can unlock unprecedented opportunities for progress and well-being, writes Andrew Keen, author of The Art of AI: How to Use AI to Save the World. The book, published by Oxford University Press, is out now. For more information, visit the book's website or order a copy by calling 08457 90 90 90 or visit [www.thebookshop.com](http://www.thebookshop.com).'