```
!pip install transformers
!pip install pandas
```

```
Requirement already satisfied: transformers in
/usr/local/lib/python3.11/dist-packages (4.52.4)
Requirement already satisfied: filelock in
/usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.33.0)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.11/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.11/dist-packages (from transformers)
(2024.11.6)
Requirement already satisfied: requests in
/usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.21.2)
Requirement already satisfied: safetensors>=0.4.3 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (2025.3.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (4.14.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (1.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(3.4.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(2025.6.15)
Requirement already satisfied: pandas in
/usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.23.2 in
```

```
/usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2-
>pandas) (1.17.0)
```

```
import pandas as pd

tweets_table = pd.read_csv('tweets-data.csv')
tweets_table.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 3010,\n  \"fields\":
[\n    {\n      \"column\": \"Unnamed: 0\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 289,\n        \"min\": 0,\n
\"max\": 1000,\n        \"num_unique_values\": 1001,\n
\"samples\": [\n          521,\n          941,\n          741\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Date Created\",\n
\"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 2423,\n        \"samples\": [\n
\"2023-06-25 18:17:22+00:00\",\n          \"2023-06-25
16:16:10+00:00\",\n          \"2023-06-25 17:53:49+00:00\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Number of Likes\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 981,\n        \"min\": 0,\n
\"max\": 26946,\n        \"num_unique_values\": 74,\n
\"samples\": [\n          6,\n          73,\n          16\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Source of Tweet\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
null,\n        \"min\": null,\n        \"max\": null,\n
\"num_unique_values\": 0,\n        \"samples\": [],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Tweets\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\":
2616,\n        \"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"hashtag\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```python
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))

def clean_tweet(tweet_msg):
    tweet_msg = str(tweet_msg).lower()
    tweet_msg = re.sub(r"http\S+|www\S+|https\S+", '', tweet_msg)
    tweet_msg = re.sub(r"@\w+|#\w+", '', tweet_msg)
    tweet_msg = re.sub(r"[^a-z\s]", '', tweet_msg)
    words = tweet_msg.split()
    words = [word for word in words if word not in stop_words]
    return " ".join(words)

tweets_table['clean_tweet_msg'] =
tweets_table['Tweets'].apply(clean_tweet)
tweets_table[['Tweets', 'clean_tweet_msg']].head()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
{"summary":"{\n  \"name\": \"df[['Tweets', 'clean_text']]\",\n
\"rows\": 5,\n  \"fields\": [\n    {\n      \"column\": \"Tweets\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Pobrecito es discapacitado\\n#Reddetuiterosdemocraticos
#LosCorruptosSiempreFueronEllos #Russia #Wagner #EcuadorSinMiedo
#Villavicencio #Pride2023\",\n          \"Il passaggio chiave di
Machiavelli era questo (\\u2018Principe\\u2019 cap. 12). #Wagner
#Prigozhin https://t.co/aeZbvtUJJi\",\n          \"News from the EIR
Daily Alert\\n\\n\\u201c#Putin Addressed the #Russian People on the
Armed #Insurrection\\u201d\\n\\nJune 24, 2023
(EIRNS)\\u2014https://t.co/sAR7wViIVP\\n\\n#Russia, #Russian
#President #VladimirPutin, #Putin, #Wagner, #WagnerGroup, #sundayvibes
https://t.co/ufwk2xaoDZ\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"clean_text\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5,\n
\"samples\": [\n          \"pobrecito es discapacitado\",\n
\"il passaggio chiave di machiavelli era questo principe cap\",\n
\"news eir daily alert addressed people armed june eirns\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```python
tweets_table_sample = tweets_table.sample(500, random_state=42).copy()

from transformers import pipeline

sentiment_pipeline = pipeline("sentiment-analysis")
```

```
No model was supplied, defaulted to distilbert/distilbert-base-
uncased-finetuned-sst-2-english and revision 714eb0f
(https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-
sst-2-english).
Using a pipeline without specifying a model name and revision in
production is not recommended.
/usr/local/lib/python3.11/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.
  warnings.warn(
```

{"model_id":"bd0d433a2da24b10a19b7828186a35ff","version_major":2,"version_minor":0}

{"model_id":"e029e3630f82449686746040fdc5b9a3","version_major":2,"version_minor":0}

{"model_id":"835dd7100a2247fcb4531f16b3179e0e","version_major":2,"version_minor":0}

{"model_id":"337bd806902949ca9d2f08a6e177518f","version_major":2,"version_minor":0}

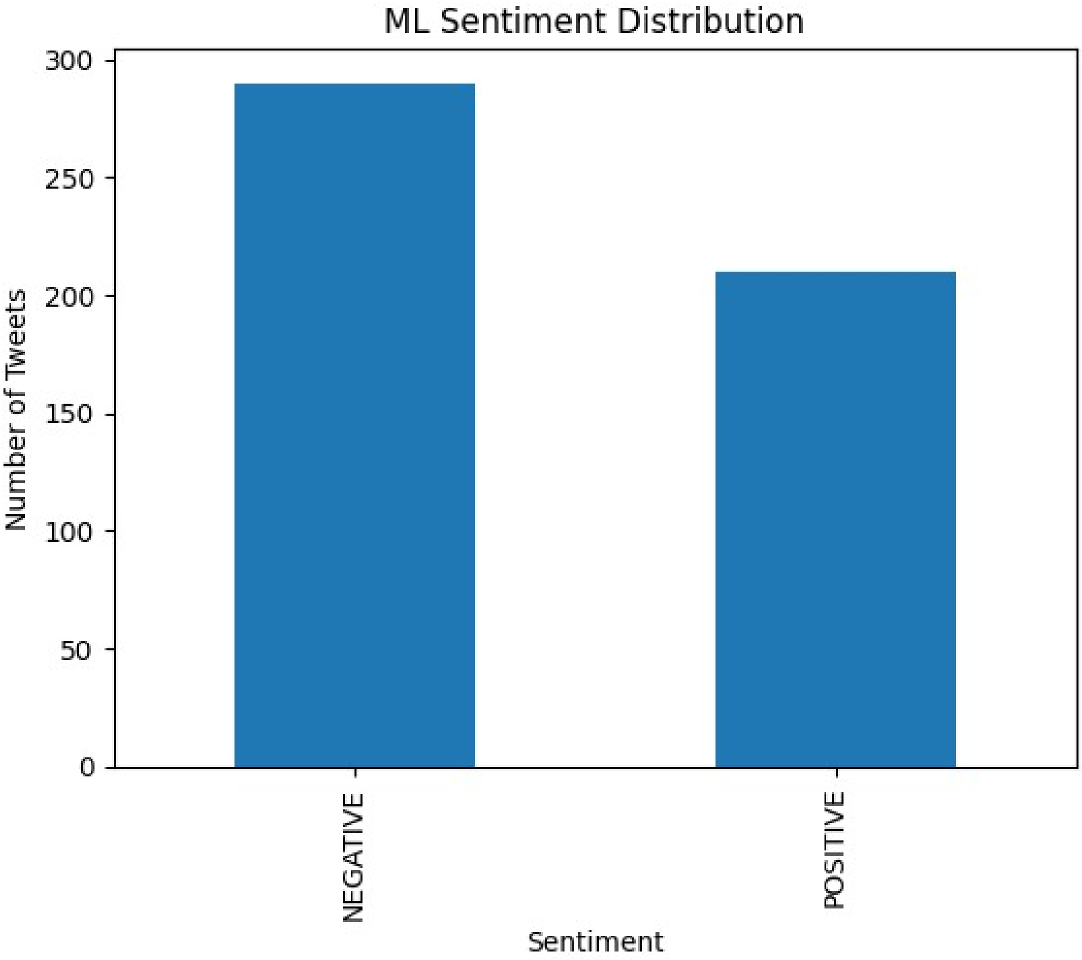
```
Device set to use cpu
```

```python
tweet_msgs = tweets_table_sample['clean_tweet_msg'].tolist()

ml_emotion_tags = []
ml_emotion_powers = []

for i in range(0, len(tweet_msgs), 50):
    batch = tweet_msgs[i:i+50]
    my_results = sentiment_pipeline(batch, truncation=True)
    ml_emotion_tags.extend([r['label'] for r in my_results])
    ml_emotion_powers.extend([r['score'] for r in my_results])

tweets_table_sample['ml_emotion_tag'] = ml_emotion_tags
tweets_table_sample['ml_emotion_power'] = ml_emotion_powers

tweets_table_sample[['Tweets', 'clean_tweet_msg', 'ml_emotion_tag',
'ml_emotion_power']].head(10)
```

{"summary":"{\n \"name\": \"df_sample[['Tweets', 'clean_text',
'ml_sentiment_label', 'ml_sentiment_score']]\",\n \"rows\": 10,\n
\"fields\": [\n   {\n   \"column\": \"Tweets\",\n

```
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n          \"#merri
le #titanic 2 le retour https://t.co/4sfvTDZNNE via @YouTube\",\n
\"#Russia #Wagner #RussiaCivilWar https://t.co/PRmMq8vnh5\",\n
\"#SUGA_AgustD_TOUR_in_Seoul #SUGA_AgustD_TOUR #glastonbury2023
#Russia #Wagner #Wagner https://t.co/aVtgad3a29\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"clean_text\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 8,\n        \"samples\": [\n          \"\",\n
\"mishap incredible force amp speed crushing water pressure floor
ocean certified huge mistake\",\n          \"le de sanaga ls sont
morts comme ils ont vcu retrouvez tous les dessins de sanaga\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"ml_sentiment_label\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n
\"POSITIVE\",\n          \"NEGATIVE\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"ml_sentiment_score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.10697040547945252,\n        \"min\": 0.7481208443641663,\n
\"max\": 0.9984667897224426,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          0.7481208443641663,\n
0.9899526238441467\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```python
import matplotlib.pyplot as plt

tweets_table_sample['ml_emotion_tag'].value_counts().plot(kind='bar',
title='ML Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Number of Tweets')
plt.show()
```


ML Sentiment Distribution
Number of Tweets
300
250
200
150
100
50
0
NEGATIVE
POSITIVE
Sentiment