In [150…
```
import os
import shutil
import tarfile

import tensorflow as tf
import torch
from transformers import BertTokenizer, TFBertForSequenceClassification, BertForSeq
import pandas as pd
from bs4 import BeautifulSoup
import re
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.offline as pyo
import plotly.graph_objects as go
from wordcloud import WordCloud, STOPWORDS
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
```

In [151…
```
df = pd.read_csv('test.csv',encoding='latin1')
```

In [152…
```
df = df[['text', 'sentiment']]
```

In [153…
```
df = df.dropna()
```

In [154…
```
df['sentiment'] = df['sentiment'].replace({'positive': 2, 'negative': 0, 'neutral':
```

```
/tmp/ipython-input-1850783318.py:1: FutureWarning:

Downcasting behavior in `replace` is deprecated and will be removed in a future vers
ion. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`.
To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting',
True)`
```

In [155…
```
from sklearn.utils import resample

# Separate classes
df_0 = df[df['sentiment'] == 0]
df_1 = df[df['sentiment'] == 1]
df_2 = df[df['sentiment'] == 2]

# Find the size of the smallest class
min_size = min(len(df_0), len(df_1), len(df_2))

# Downsample all classes to the smallest class size
df_0_balanced = resample(df_0, replace=False, n_samples=min_size, random_state=42)
df_1_balanced = resample(df_1, replace=False, n_samples=min_size, random_state=42)
df_2_balanced = resample(df_2, replace=False, n_samples=min_size, random_state=42)

# Combine balanced classes
df_balanced = pd.concat([df_0_balanced, df_1_balanced, df_2_balanced])
```

```
# Shuffle the dataset
df_balanced = df_balanced.sample(frac=1, random_state=42).reset_index(drop=True)
```

In [156…

```
sentiment_counts = df_balanced['sentiment'].value_counts()

fig =px.bar(x= {-1:'negative',1:'positive', 0:"neutral"},
            y= sentiment_counts.values,
            color=sentiment_counts.index,
            color_discrete_sequence =  px.colors.qualitative.Dark24,
            title='<b>Sentiments Counts')

fig.update_layout(title='Sentiments Counts',
                  xaxis_title='Sentiment',
                  yaxis_title='Counts',
                  template='plotly_dark')

# Show the bar chart
fig.show()
pyo.plot(fig, filename = 'Sentiments Counts.html', auto_open = True)
```