Human Stress Detection

April 14, 2024

1 Enhancing Mental Health: Stress Level Prediction through a Machine Learning and NLP Approach

1.1 Importing Libraries

[139]:
```
import sys
import keras
import pandas as pd
import sklearn as sk
import scipy as sp
import tensorflow as tf
import numpy as np
import platform

print (f"Python Platform: {platform.platform ()}")
print (f"Tensor Flow Version: {tf.__version__}")
print(f"Keras Version: {keras.__version__}")
print ()

print (f"Python {sys.version}")
print (f"Pandas {pd.__version__}")
print (f"Scikit-Learn {sk.__version__}")
print (f"SciPy {sp.__version__}")
gpu = len (tf.config.list_physical_devices ('GPU'))>0
print ("GPU is", "available" if gpu else "NOT AVAILABLE")
```

```
Python Platform: macOS-14.2-arm64-arm-64bit
Tensor Flow Version: 2.16.1
Keras Version: 3.2.1

Python 3.11.7 (main, Dec 15 2023, 12:09:56) [Clang 14.0.6 ]
Pandas 2.2.2
Scikit-Learn 1.2.2
SciPy 1.13.0
GPU is available
```

1.1.1 Loading Dataset

[2]:
```
# Load CSV file into a DataFrame

df = pd.read_csv('mental_health.csv')

# Display the DataFrame
df.head(20)
```

[2]:

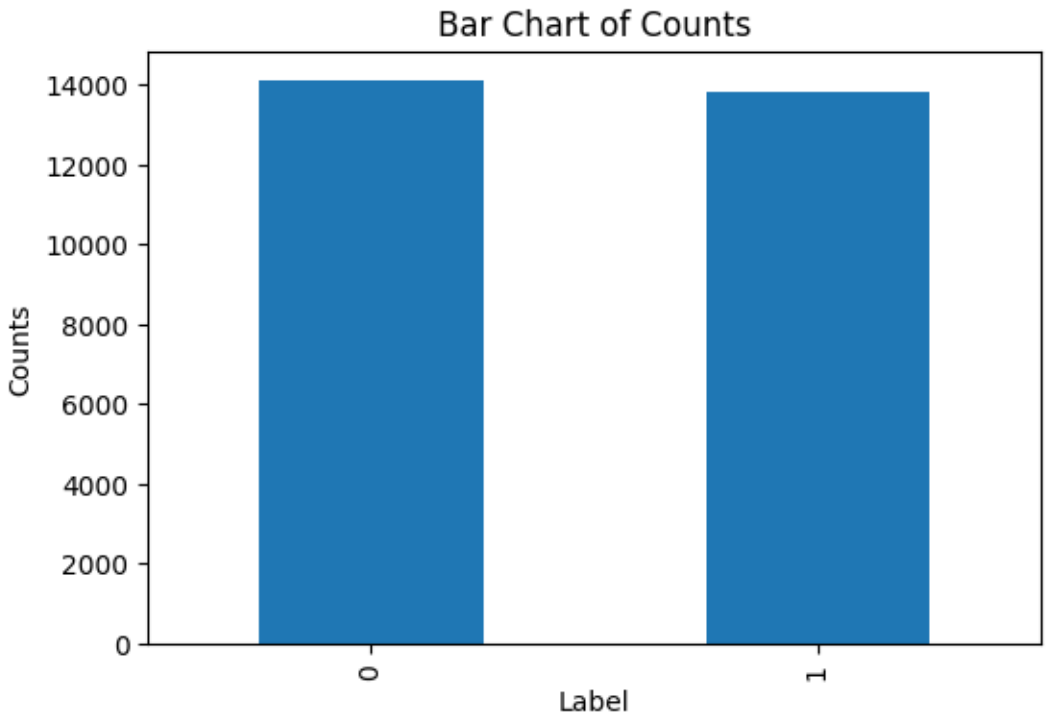
	text	label
0	dear american teens question dutch person hear…	0
1	nothing look forward lifei dont many reasons k…	1
2	music recommendations im looking expand playli…	0
3	im done trying feel betterthe reason im still …	1
4	worried year old girl subject domestic physic…	1
5	hey rredflag sure right place post this goes …	1
6	feel like someone needs hear tonight feeling r…	0
7	deserve liveif died right noone would carei re…	1
8	feels good ive set dateim killing friday nice …	1
9	live guiltok made stupid random choice its ge…	1
10	excercise motivated ngl cant wait get shape kn…	0
11	know youd rather laid big booty body hella pos…	0
12	even time fuck supposed mean	0
13	usual hollywood stereotyped everyone movie but…	0
14	think it nearly unbelievable film could made d…	0
15	trying rd time k krma special	0
16	guy coming sure wear f hey guy friend coming t…	0
17	one best episodes entire xfiles series creepy …	0
18	good byehey you know sure hell know me goodbye…	1
19	tried put sugar coffee back spoon happy monday…	1

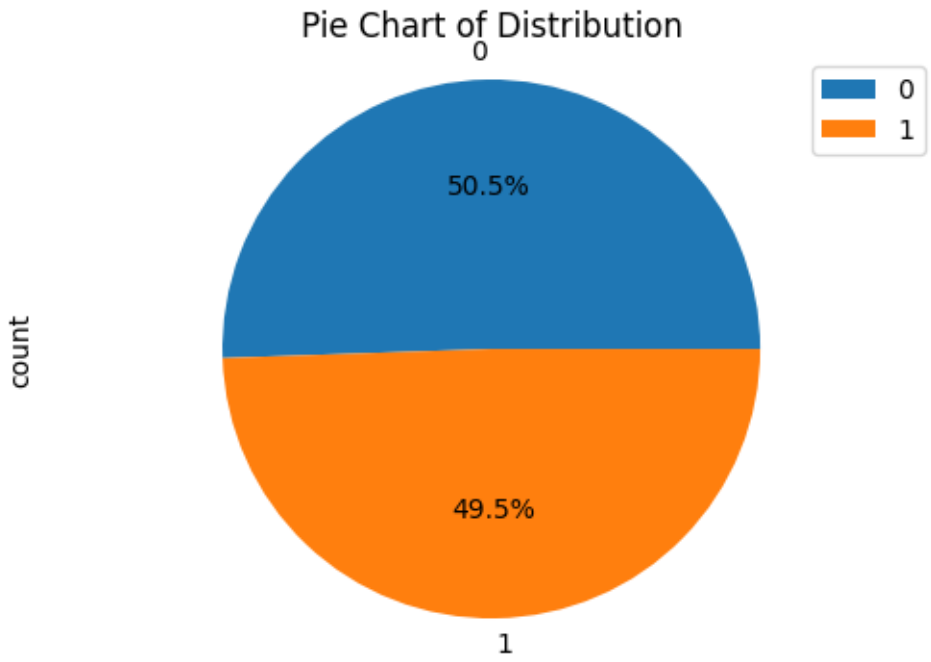
[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27977 entries, 0 to 27976
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   text    27977 non-null  object
 1   label   27977 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 437.3+ KB
```

[4]:
```
import matplotlib.pyplot as plt

# Calculate the value counts of the 'category' column
category_counts = df['label'].value_counts()
```

```
# Bar chart
plt.figure(figsize=(6, 4))
category_counts.plot(kind='bar')
plt.xlabel('Label')
plt.ylabel('Counts')
plt.title('Bar Chart of Counts')
plt.show()
print()

# Pie chart
plt.figure(figsize=(6, 4))
category_counts.plot(kind='pie', autopct='%1.1f%%')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
plt.title('Pie Chart of Distribution')

# Add legend
plt.legend()

plt.show()
```





1.2 Duplicates and Missing Values

Next, we'll evaluate the dataset's size and search for any duplicate rows within the DataFrame. We'll achieve this by counting the duplicates and displaying the results. Handling duplicate rows is crucial as they can distort our analysis. Additionally, we'll check for missing values by calculating and printing the sum of missing values for each column. Properly addressing missing values is essential to ensure the accuracy and reliability of our analysis.

[5]:
```
# How many reviews do we have?
print('There are', df.shape[0], 'data in this dataset')

#Dupicate Check?
print('Number of Duplicates:', len(df[df.duplicated()]))

# Missing Values Check
missing_values = df.isnull().sum()
print('Number of Missing Values by column:\n',missing_values)

print('Number of Missing Values:', df.isnull().sum().sum())
```

```
There are 27977 data in this dataset
Number of Duplicates: 5
Number of Missing Values by column:
 text     0
label    0
dtype: int64
```

```
Number of Missing Values: 0
```

[6]:
```
df.replace("", np.nan, inplace=True)
missing_values = df.isnull().sum()
print('No. of Missing Values and Empty Spaces by column:\n',missing_values)
```

```
No. of Missing Values and Empty Spaces by column:
 text     0
label    0
dtype: int64
```

[7]:
```
# all duplicate rows (keep=False ensures all duplicates are kept)
duplicate_rows = df[df.duplicated(keep=False)]

# Then sort the dataframe on all columns to ensure duplicates are adjacent
sorted_duplicates = duplicate_rows.sort_values(by=list(duplicate_rows.columns))

# Now, if we want to see 5 pairs of duplicates (10 rows), we can simply:
top_5_duplicate_pairs = sorted_duplicates.head(20)

top_5_duplicate_pairs
```

[7]:

	text	label
15524	happy birthday everyone birthday st october ha…	0
24502	happy birthday everyone birthday st october ha…	0
16742	need help anyone good pythagriam tribometry h…	0
24970	need help anyone good pythagriam tribometry h…	0
1646	posting ara ara forget day ara ara	0
22603	posting ara ara forget day ara ara	0
11570	real suppleroot hours up day far	0
12573	real suppleroot hours up day far	0
22389	real suppleroot hours up day far	0

1.2.1 Drop Duplicates

[8]:
```
df = df.drop_duplicates()
print('Number of Duplicates:', len(df[df.duplicated()]))
```

```
Number of Duplicates: 0
```

1.2.2 Drop Missing Values

[9]:
```
df = df.dropna()
print('Number of Missing Values:', df.isnull().sum().sum())
```

```
Number of Missing Values: 0
```

[10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 27972 entries, 0 to 27976
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   text    27972 non-null  object
 1   label   27972 non-null  int64
dtypes: int64(1), object(1)
memory usage: 655.6+ KB
```

View random samples for each category

Here, a function called "random_sample_reviews" is defined to randomly sample text from the DataFrame based on the specified number of samples. This function groups the text by label and selects a specified number of samples from each label. The sampled reviews are then returned as a DataFrame. This function helps in obtaining a representative subset of reviews for analysis.

[11]:
```
def random_sample_reviews(df, num_samples):
    # Use groupby on 'Rating' and then apply the sample function to␣
 ↪'Review_Text' of each group
    samples = df.groupby('label')['text'].apply(lambda x: x.sample(num_samples))

    # Convert series to dataframe and reset index
    samples_df = samples.reset_index().drop(columns='level_1')
    return samples_df

pd.set_option('display.max_colwidth', 200) # This will display up to 100␣
 ↪characters
samples = random_sample_reviews(df, num_samples=3)
samples.head(20)
```

[11]:
```
   label  \
0      0
1      0
2      0
3      1
4      1
5      1

                                                                                                                                                                                                        text
0  lesser known film starring roy thinnes from tvs invaders actually consider
lost gem made time story important special effects though effect fairly good
time scientist theorizes another world earth…
1  cant seem make friends honestly cant make friends feels like matter do happen
well id say one friend who love much appreciate every day couple people
occasionally talk like aside nobody strangers …
2  mess genres mainly based stephen chows genre mashups inspiration theres magic
kungfu college romance sports gangster action weepy melodrama topping production
```

```
excellent pacing fast easy get past m…
3  im ready take forever napi reasons continue hold onto bullshit life cant
myself family fucking garbage nobody would believe tried escape them fucking
mess mention fact ive fucking havent able onli…
4  record suicidal never been going tell something true story might make think
another wayfor background moms friend call susan speaks russian english works
translation hospitals youssusan client cal…
5  know whats funnyhow people fucking tell shit care truth care dead people
supposed make feel safe fucking make feel worse fucking hope feel guilty lost
fucking horrible damage inflicted onto never …
```

1.3 Data Cleaning

[12]:
```
#libraries
from sklearn.model_selection import train_test_split
from sklearn import metrics

import re
import string

from tensorflow import keras
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding
from tensorflow.keras.layers import SimpleRNN, LSTM
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

[13]:
```
##CUSTOM FUNCTIONS TO CLEAN THE TEXT
def emoji_strip(text):
    emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
        u"\U00002500-\U00002BEF"  # chinese characters
        u"\U00002702-\U000027B0"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
```

```
        u"\ufe0f"  # dingbats
        u"\u3030"
                          "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', text)

#Remove punctuations, links, mentions and \r\n new line characters
def remove_entities(text):
    text = text.replace('\r', '').replace('\n', ' ').replace('\n', ' ').lower()␣
 ↪#remove \n and \r and lowercase
    text = re.sub(r"(?:\@|https?\://)\S+", "", text) #remove links and mentions
    text = re.sub(r'[^\x00-\x7f]',r'', text) #remove non utf8/ascii characters␣
 ↪such as '\x9a\x91\x97\x9a\x97'
    banned_list= string.punctuation + 'Ã'+'±'+'ã'+'¼'+'â'+'»'+'§'
    table = str.maketrans('', '', banned_list)
    text = text.translate(table)
    return text

#clean hashtags at the end of the sentence, and keep those in the middle of the␣
 ↪sentence by removing just the # symbol
def hastag_cleaning(text):
    new_text = " ".join(word.strip() for word in re.split('#(?!(?:
 ↪hashtag)\b)[\w-]+(?=(?:\s+#[\w-]+)*\s*$)', text)) #remove last hashtags
    new_text2 = " ".join(word.strip() for word in re.split('#|_', new_text))␣
 ↪#remove hashtags symbol from words in the middle of the sentence
    return new_text2

#Filter special characters such as & and $ present in some words
def filter_chars(a):
    sent = []
    for word in a.split(' '):
        if ('$' in word) | ('&' in word):
            sent.append('')
        else:
            sent.append(word)
    return ' '.join(sent)

def remove_spaces(text): # remove multiple spaces
    return re.sub("\s\s+" , " ", text)
```

[14]:
```
df['text1'] = (df['text']
                    .apply(emoji_strip)
                    .apply(remove_entities)
                    .apply(hastag_cleaning)
                    .apply(filter_chars)
                    .apply(remove_spaces))
```

[15]:
```
df.head()
```

[15]:

```
                                               text  \
0                                                                    dear american
teens question dutch person heard guys get way easier things learn age us sooooo
thth graders like  right guys learn math
1
nothing look forward lifei dont many reasons keep going feel like nothing keeps
going next day makes want hang myself
2  music recommendations im looking expand playlist usual genres alt pop
minnesota hip hop steampunk various indie genres artists people like cavetown
aliceband bug hunter penelope scott various rhym…
3  im done trying feel betterthe reason im still alive know mum devastated ever
killed myself ever passes im still state im going hesitate ending life shortly
after im almost  take meds go therapy no…
4  worried  year old girl subject domestic physicalmental housewithout going lot
know girl know girl etc let give brief background known girl  years lives uk
live different country kept touch electro…

   label  \
0      0
1      1
2      0
3      1
4      1

                                              text1
0                                                                     dear
american teens question dutch person heard guys get way easier things learn age
us sooooo thth graders like right guys learn math
1
nothing look forward lifei dont many reasons keep going feel like nothing keeps
going next day makes want hang myself
2  music recommendations im looking expand playlist usual genres alt pop
minnesota hip hop steampunk various indie genres artists people like cavetown
aliceband bug hunter penelope scott various rhym…
3  im done trying feel betterthe reason im still alive know mum devastated ever
killed myself ever passes im still state im going hesitate ending life shortly
after im almost take meds go therapy not…
4  worried year old girl subject domestic physicalmental housewithout going lot
know girl know girl etc let give brief background known girl years lives uk live
different country kept touch electroni…
```

Let's compare the original and cleaned text data and analyze the impact of text cleaning on the text length.

[17]:

```
df_compare = pd.DataFrame()

# Original text and its length
df_compare['pre-clean text'] = df['text']
```

```
df_compare['pre-clean len'] = df['text'].apply(lambda x: len(str(x).split()))

# Cleaned text and its length
df_compare['post-clean text'] = df['text1']
df_compare['post-clean len'] = df['text1'].apply(lambda x: len(str(x).split()))

df_compare.head(10)
```

[17]:

```
                                pre-clean text  \
0                                                                  dear american
teens question dutch person heard guys get way easier things learn age us sooooo
thth graders like  right guys learn math
1
nothing look forward lifei dont many reasons keep going feel like nothing keeps
going next day makes want hang myself
2  music recommendations im looking expand playlist usual genres alt pop
minnesota hip hop steampunk various indie genres artists people like cavetown
aliceband bug hunter penelope scott various rhym…
3  im done trying feel betterthe reason im still alive know mum devastated ever
killed myself ever passes im still state im going hesitate ending life shortly
after im almost  take meds go therapy no…
4  worried  year old girl subject domestic physicalmental housewithout going lot
know girl know girl etc let give brief background known girl  years lives uk
live different country kept touch electro…
5  hey rredflag sure right place post this goes  im currently student intern
sandia national labs working survey help improve marketing outreach efforts many
schools recruit around country were looki…
6  feel like someone needs hear tonight feeling right think cant anything people
keep puting listen this its your life everyone else living it someone tells
unable something work get done say wrong s…
7  deserve liveif died right noone would carei real friendsi always start
conversations get dry responses i feel comfortable around females emotional
abuse mom put left usi never find love i keep get…
8
feels good ive set dateim killing friday nice finally know im gonna it bye
9  live guiltok made stupid random choice  its getting me basically molested
relative super erratic thing haunting right now random walk home randomly
assaulted classmate screamed name loud pretty mu…

   pre-clean len  \
0             23
1             20
2             64
3            100
4            311
5             61
6             79
```

```
7             51
8             14
9             66

                          post-clean text  \
0                                                                    dear
american teens question dutch person heard guys get way easier things learn age
us sooooo thth graders like right guys learn math
1
nothing look forward lifei dont many reasons keep going feel like nothing keeps
going next day makes want hang myself
2  music recommendations im looking expand playlist usual genres alt pop
minnesota hip hop steampunk various indie genres artists people like cavetown
aliceband bug hunter penelope scott various rhym…
3  im done trying feel betterthe reason im still alive know mum devastated ever
killed myself ever passes im still state im going hesitate ending life shortly
after im almost take meds go therapy not…
4  worried year old girl subject domestic physicalmental housewithout going lot
know girl know girl etc let give brief background known girl years lives uk live
different country kept touch electroni…
5  hey rredflag sure right place post this goes im currently student intern
sandia national labs working survey help improve marketing outreach efforts many
schools recruit around country were lookin…
6  feel like someone needs hear tonight feeling right think cant anything people
keep puting listen this its your life everyone else living it someone tells
unable something work get done say wrong s…
7  deserve liveif died right noone would carei real friendsi always start
conversations get dry responses i feel comfortable around females emotional
abuse mom put left usi never find love i keep get…
8
feels good ive set dateim killing friday nice finally know im gonna it bye
9  live guiltok made stupid random choice its getting me basically molested
relative super erratic thing haunting right now random walk home randomly
assaulted classmate screamed name loud pretty muc…

   post-clean len
0              23
1              20
2              64
3             100
4             311
5              61
6              79
7              51
8              14
9              66
```

1.4 Remove Stopwords

[18]:
```
def rm_stopwords(sentence):

    # List of stopwords
    stopwords = ["a", "about", "above", "after", "again", "against", "all",␣
 ↪"am", "an", "and", "any", "are", "as", "at", "be", "because", "been",␣
 ↪"before", "being", "below", "between", "both", "but", "by", "could", "did",␣
 ↪"do", "does", "doing", "down", "during", "each", "few", "for", "from",␣
 ↪"further", "had", "has", "have", "having", "he", "he'd", "he'll", "he's",␣
 ↪"her", "here", "here's", "hers", "herself", "him", "himself", "his", "how",␣
 ↪"how's", "i", "i'd", "i'll", "i'm", "i've", "if", "in", "into", "is", "it",␣
 ↪"it's", "its", "itself", "let's", "me", "more", "most", "my", "myself",␣
 ↪"nor", "of", "on", "once", "only", "or", "other", "ought", "our", "ours",␣
 ↪"ourselves", "out", "over", "own", "same", "she", "she'd", "she'll",␣
 ↪"she's", "should", "so", "some", "such", "than", "that", "that's", "the",␣
 ↪"their", "theirs", "them", "themselves", "then", "there", "there's",␣
 ↪"these", "they", "they'd", "they'll", "they're", "they've", "this", "those",␣
 ↪"through", "to", "too", "under", "until", "up", "very", "was", "we", "we'd",␣
 ↪"we'll", "we're", "we've", "were", "what", "what's", "when", "when's",␣
 ↪"where", "where's", "which", "while", "who", "who's", "whom", "why",␣
 ↪"why's", "with", "would", "you", "you'd", "you'll", "you're", "you've",␣
 ↪"your", "yours", "yourself", "yourselves" ]

    # Sentence converted to lowercase-only
    sentence = sentence.lower()

    words = sentence.split()
    no_words = [w for w in words if w not in stopwords]
    sentence = " ".join(no_words)

    return sentence
```

[19]:
```
df['text2'] = (df['text1'].apply(rm_stopwords))
```

compare the original and stopwords-removed text data and analyze the impact of removing stopwords on the text length.

[20]:
```
df_comp = pd.DataFrame()

# Original text and its length
df_comp['pre-clean text'] = df['text1']
df_comp['pre-clean len'] = df['text1'].apply(lambda x: len(str(x).split()))

# Cleaned text and its length
df_comp['post-clean text'] = df['text2']
df_comp['post-clean len'] = df['text2'].apply(lambda x: len(str(x).split()))
```

```
df_comp.head(5)
```

[20]:

```
                                pre-clean text  \
0                                                                    dear
american teens question dutch person heard guys get way easier things learn age
us sooooo thth graders like right guys learn math
1
nothing look forward lifei dont many reasons keep going feel like nothing keeps
going next day makes want hang myself
2  music recommendations im looking expand playlist usual genres alt pop
minnesota hip hop steampunk various indie genres artists people like cavetown
aliceband bug hunter penelope scott various rhym…
3  im done trying feel betterthe reason im still alive know mum devastated ever
killed myself ever passes im still state im going hesitate ending life shortly
after im almost take meds go therapy not…
4  worried year old girl subject domestic physicalmental housewithout going lot
know girl know girl etc let give brief background known girl years lives uk live
different country kept touch electroni…

   pre-clean len  \
0             23
1             20
2             64
3            100
4            311

                               post-clean text  \
0                                                                    dear
american teens question dutch person heard guys get way easier things learn age
us sooooo thth graders like right guys learn math
1
nothing look forward lifei dont many reasons keep going feel like nothing keeps
going next day makes want hang
2  music recommendations im looking expand playlist usual genres alt pop
minnesota hip hop steampunk various indie genres artists people like cavetown
aliceband bug hunter penelope scott various rhym…
3  im done trying feel betterthe reason im still alive know mum devastated ever
killed ever passes im still state im going hesitate ending life shortly im
almost take meds go therapy nothing seems he…
4  worried year old girl subject domestic physicalmental housewithout going lot
know girl know girl etc let give brief background known girl years lives uk live
different country kept touch electroni…

   post-clean len
0              23
1              19
2              61
```

```
3              96
4             296
```

1.5 Lemmatization

[21]:
```
import nltk
nltk.download('wordnet')
nltk.download('punkt')

from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

lemmatizer = WordNetLemmatizer()

def lemmatize_text(text):
    # Tokenize the sentence
    word_list = nltk.word_tokenize(text)

    # Lemmatize list of words and join
    lemmatized_output = ' '.join([lemmatizer.lemmatize(w) for w in word_list])

    return lemmatized_output
```

```
[nltk_data] Downloading package wordnet to /Users/raja/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to /Users/raja/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

The lemmatization function is then applied to the 'text2' column of the DataFrame 'df' using the apply method, and the lemmatized output is assigned to the 'text3' column.

[22]:
```
df['text3'] = df['text2'].apply(lemmatize_text)
```

A new DataFrame, 'df_lemma', is created to store the original and lemmatized text data along with their respective lengths.

[23]:
```
df_lemma = pd.DataFrame()

# Original text and its length
df_lemma['pre-clean text'] = df['text2']
df_lemma['pre-clean len'] = df['text2'].apply(lambda x: len(str(x).split()))

# Cleaned text and its length
df_lemma['post-clean text'] = df['text3']
df_lemma['post-clean len'] = df['text3'].apply(lambda x: len(str(x).split()))

df_lemma.head(20)
```

```
[23]:                              pre-clean text  \
0                                                                      dear
american teens question dutch person heard guys get way easier things learn age
us sooooo thth graders like right guys learn math
1
nothing look forward lifei dont many reasons keep going feel like nothing keeps
going next day makes want hang
2   music recommendations im looking expand playlist usual genres alt pop
minnesota hip hop steampunk various indie genres artists people like cavetown
aliceband bug hunter penelope scott various rhym…
3   im done trying feel betterthe reason im still alive know mum devastated ever
killed ever passes im still state im going hesitate ending life shortly im
almost take meds go therapy nothing seems he…
4   worried year old girl subject domestic physicalmental housewithout going lot
know girl know girl etc let give brief background known girl years lives uk live
different country kept touch electroni…
5   hey rredflag sure right place post goes im currently student intern sandia
national labs working survey help improve marketing outreach efforts many
schools recruit around country looking current …
6   feel like someone needs hear tonight feeling right think cant anything
people keep puting listen life everyone else living someone tells unable
something work get done say wrong someone says youl …
7   deserve liveif died right noone carei real friendsi always start
conversations get dry responses feel comfortable around females emotional abuse
mom put left usi never find love keep getting remin…
8
feels good ive set dateim killing friday nice finally know im gonna bye
9   live guiltok made stupid random choice getting basically molested relative
super erratic thing haunting right now random walk home randomly assaulted
classmate screamed name loud pretty much annoy…
10
excercise motivated ngl cant wait get shape know gonna overnight im happy right
now
11
know youd rather laid big booty body hella positive cuz got big booty
12
even time fuck supposed mean
13  usual hollywood stereotyped everyone movie one classic uptight white collar
banker russian woman well done even facial expressions great language perfect
even russian language nicole splendid job …
14  think nearly unbelievable film made death penalty one worlds controversial
topics offends neither testament tim robbins extraordinary intelligence
sensitivity traits seen acting roles well shawsha…
15
trying rd time k krma special
16  guy coming sure wear f hey guy friend coming tomorrow im excited im sure
wear ive known since middle school weve talking couple months honest know really
```

```
care wear will want wear dress something t…
17  one best episodes entire xfiles series creepy beyond words tension suspense
episode well executed entire minutes managed almost scary entire movie episode
joins ranks best episodes greats home hum…
18  good byehey know sure hell know goodbye probably mean anything plus bother
read rules sub may may taken hard getting harder weakened much ever since small
innocent child things bad almost every da…
19
tried put sugar coffee back spoon happy monday everyonestay safe sunflowers one
days

    pre-clean len  \
0              23
1              19
2              61
3              96
4             296
5              57
6              69
7              47
8              13
9              62
10             14
11             13
12              5
13             31
14             55
15              6
16            122
17             58
18            179
19             13

                            post-clean text  \
0                                                                          dear
american teen question dutch person heard guy get way easier thing learn age u
sooooo thth grader like right guy learn math
1
nothing look forward lifei dont many reason keep going feel like nothing keep
going next day make want hang
2   music recommendation im looking expand playlist usual genre alt pop
minnesota hip hop steampunk various indie genre artist people like cavetown
aliceband bug hunter penelope scott various rhymesay…
3   im done trying feel betterthe reason im still alive know mum devastated ever
killed ever pass im still state im going hesitate ending life shortly im almost
take med go therapy nothing seems help …
4   worried year old girl subject domestic physicalmental housewithout going lot
```

know girl know girl etc let give brief background known girl year life uk live different country kept touch electronic …
5 hey rredflag sure right place post go im currently student intern sandia national lab working survey help improve marketing outreach effort many school recruit around country looking current under…
6 feel like someone need hear tonight feeling right think cant anything people keep puting listen life everyone else living someone tell unable something work get done say wrong someone say youl nev…
7 deserve liveif died right noone carei real friendsi always start conversation get dry response feel comfortable around female emotional abuse mom put left usi never find love keep getting reminded…
8
feel good ive set dateim killing friday nice finally know im gon na bye
9 live guiltok made stupid random choice getting basically molested relative super erratic thing haunting right now random walk home randomly assaulted classmate screamed name loud pretty much annoy…
10
excercise motivated ngl cant wait get shape know gon na overnight im happy right now
11
know youd rather laid big booty body hella positive cuz got big booty
12
even time fuck supposed mean
13 usual hollywood stereotyped everyone movie one classic uptight white collar banker russian woman well done even facial expression great language perfect even russian language nicole splendid job h…
14 think nearly unbelievable film made death penalty one world controversial topic offends neither testament tim robbins extraordinary intelligence sensitivity trait seen acting role well shawshank r…
15
trying rd time k krma special
16 guy coming sure wear f hey guy friend coming tomorrow im excited im sure wear ive known since middle school weve talking couple month honest know really care wear will want wear dress something th…
17 one best episode entire xfiles series creepy beyond word tension suspense episode well executed entire minute managed almost scary entire movie episode join rank best episode great home humbug bad…
18 good byehey know sure hell know goodbye probably mean anything plus bother read rule sub may may taken hard getting harder weakened much ever since small innocent child thing bad almost every day …
19
tried put sugar coffee back spoon happy monday everyonestay safe sunflower one day

	post-clean len
0	23
1	19

```
2                61
3                97
4               296
5                57
6                69
7                47
8                14
9                62
10               15
11               13
12                5
13               31
14               55
15                6
16              124
17               58
18              179
19               13
```

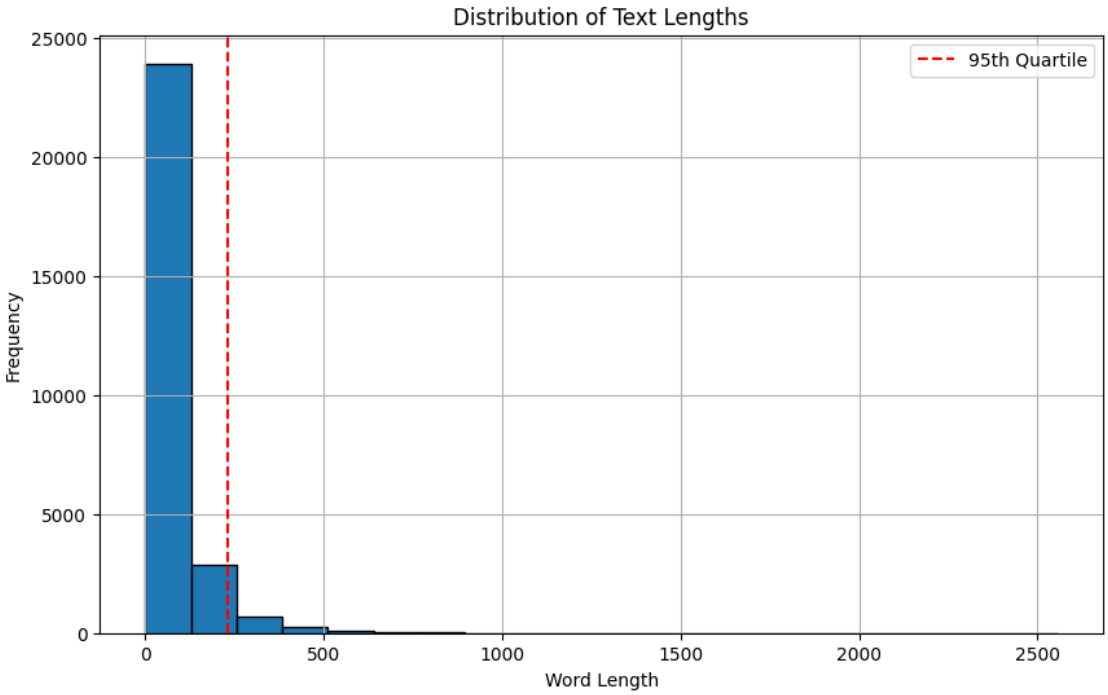
1.5.1 Text Length

[24]:
```
df['text_length'] = df['text3'].apply(lambda x: len(str(x).split()))
```

[25]:
```
# Calculate the length of each text in X_train
text_lengths = [len(text.split()) for text in df["text3"]]

# Find the 95th quartile
quartile_95 = np.percentile(text_lengths, 95)

print(f"95th Quartile of Text Lengths: {quartile_95}")
```

```
95th Quartile of Text Lengths: 231.0
```

[26]:
```
# Plotting the histogram
plt.figure(figsize=(10, 6))
plt.hist(text_lengths, bins=20, edgecolor='black')
plt.xlabel('Word Length')
plt.ylabel('Frequency')
plt.title('Distribution of Text Lengths')

# Adding a vertical line for the 95th quartile
quartile_95 = np.percentile(text_lengths, 95)
plt.axvline(x=quartile_95, color='red', linestyle='--', label='95th Quartile')
plt.legend()

plt.grid(True)
plt.show()
```



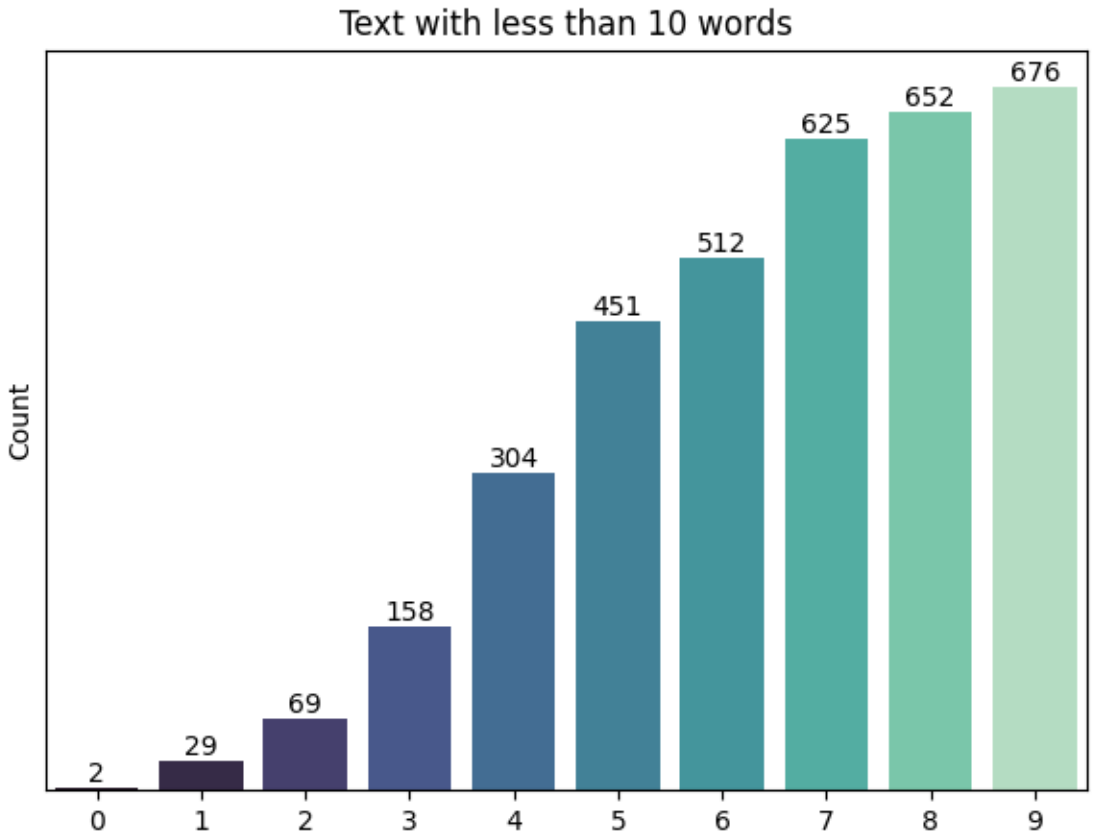
[27]:
```
df.text_length.describe()
```

[27]:
```
count    27972.000000
mean        68.121014
std         97.784015
min          0.000000
25%         15.000000
50%         36.000000
75%         82.000000
max       2556.000000
Name: text_length, dtype: float64
```

1.5.2 Visualize text with low frequency words

[28]:
```
import seaborn as sns

plt.figure(figsize=(7,5))
ax = sns.countplot(x='text_length', data=df[df['text_length']<10],␣
 ↪palette='mako')
plt.title('Text with less than 10 words')
plt.yticks([])
ax.bar_label(ax.containers[0])
plt.ylabel('Count')
plt.xlabel('')
```

```
plt.show()
```



A subset of the DataFrame 'df' is created

[29]:
```
data_head=df[df['text_length']<2] # rows where the text length is less than 2
data_head.head(30)
```

[29]:

	text
654	
sleep	
1811	
karent	
2781	
male	
3354	
cthgisnialpnidenodnabamocefilretfalarutcetihcrasptth	
3626	
enoughhttpimgurcomhqermql	
4094	whiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelya…

4578
name
4743
hello
4837
something
6579
8616
this hopeless
8746
medicationmarijuana
8971
hopehttpsyoutubecyhejlya
10556
tetetethrsrhthtfhtfht
10560
moment
10696
heyhi
11755
enemies
11920
sorry
12794
what life
13183
13268
im
15583
fap i
17841
i nothing
18242
godammit
21309
deleted
21535
vsauce
22078
i tried
22095
yes
25772
feel
25843
cout

```
       label  \
654        0
1811       0
2781       0
3354       1
3626       1
4094       1
4578       1
4743       1
4837       0
6579       0
8616       1
8746       1
8971       1
10556      1
10560      0
10696      1
11755      0
11920      1
12794      1
13183      1
13268      0
15583      0
17841      1
18242      0
21309      0
21535      0
22078      1
22095      0
25772      0
25843      0

                                              text1  \
654
sleep
1811
karent
2781
male
3354
cthgisnialpnidenodnabamocefilretfalarutcetihcrasptth
3626
enoughhttpimgurcomhqermql
4094   whiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemik
eechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikee
chowhiskeyhotelyankeemikeechowhiskeyhotelya…
4578
```

```
name
4743
hello
4837
something
6579
8616
this hopeless
8746
medicationmarijuana
8971
hopehttpsyoutubecyhejlya
10556
tetetethrsrhthtfhtfht
10560
moment
10696
heyhi
11755
enemies
11920
sorry
12794
what life
13183
13268
im
15583
fap i
17841
i nothing
18242
godammit
21309
deleted
21535
vsauce
22078
i tried
22095
yes
25772
feel
25843
cout
```

```
                                        text2  \
```

654
sleep
1811
karent
2781
male
3354
cthgisnialpnidenodnabamocefilretfalarutcetihcrasptth
3626
enoughhttpimgurcomhqermql
4094 whiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemik
eechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikee
chowhiskeyhotelyankeemikeechowhiskeyhotelya…
4578
name
4743
hello
4837
something
6579
8616
hopeless
8746
medicationmarijuana
8971
hopehttpsyoutubecyhejlya
10556
tetetethrsrhthtfhtfht
10560
moment
10696
heyhi
11755
enemies
11920
sorry
12794
life
13183
13268
im
15583
fap
17841
nothing
18242
godammit

```
21309
deleted
21535
vsauce
22078
tried
22095
yes
25772
feel
25843
cout

                                          text3  \
654
sleep
1811
karent
2781
male
3354
cthgisnialpnidenodnabamocefilretfalarutcetihcrasptth
3626
enoughhttpimgurcomhqermql
4094   whiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemik
eechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikeechowhiskeyhotelyankeemikee
chowhiskeyhotelyankeemikeechowhiskeyhotelya…
4578
name
4743
hello
4837
something
6579
8616
hopeless
8746
medicationmarijuana
8971
hopehttpsyoutubecyhejlya
10556
tetetethrsrhthtfhtfht
10560
moment
10696
heyhi
11755
```

```
enemy
11920
sorry
12794
life
13183
13268
im
15583
fap
17841
nothing
18242
godammit
21309
deleted
21535
vsauce
22078
tried
22095
yes
25772
feel
25843
cout
```

	text_length
654	1
1811	1
2781	1
3354	1
3626	1
4094	1
4578	1
4743	1
4837	1
6579	0
8616	1
8746	1
8971	1
10556	1
10560	1
10696	1
11755	1
11920	1
12794	1

```
13183            0
13268            1
15583            1
17841            1
18242            1
21309            1
21535            1
22078            1
22095            1
25772            1
25843            1
```

[30]:
```
len(df)
```

[30]:
```
27972
```

[31]:
```
df = df[df['text_length'] >= 3]
```

[32]:
```
len(df)
```

[32]:
```
27872
```

Drop the columns and shuffle

[33]:
```
df = df.drop(['text', 'text1', 'text2'], axis=1)
```

[34]:
```
# Shuffle training dataframe
df = df.sample(frac=1, random_state=42) # shuffle with random_state=42 for␣
 ↪reproducibility
df.head(30)
```

[34]:
```
       label  \
10110      0
16118      0
4336       1
7496       1
7954       1
25377      0
17739      0
1762       1
21839      0
26355      1
12034      1
3933       1
14574      1
6021       0
20925      1
314        0
```

```
17378      1
15376      0
5554       1
11866      0
24358      0
26465      0
13681      1
17242      0
8707       1
10649      0
10951      1
11810      1
4161       1
18247      1

                                          text3  \
10110                                                          video picture
death video photo always death youll eventually die youll already taken video
solved clickbait video video taken death thank later
16118
help help commenting post want orange mail pls
4336
anyone talk toi need someone talk situation
7496   tonightno shitty life bad circumstance no im lazy incompetent
unsuccessful nothing happened beyond normal im done living want anymore ive
downward spiral since im now shit together im well way ish…
7954   anyone feel like life force throati feel like everything life force upon
im forced act way way im forced go school im forced work job passion im forced
smile im forced cry im forced get married ki…
25377  anyone else absolutely hate situation mean getting scolded parent
remaining calm trying explain everything time finally getting annoyed situation
exact second turn theyre calm nowhere trying act l…
17739
anyone want silver idk whoever comment first ig
1762   destined failure tragedy mom dy im year old im beside dy pneumonia breast
cancer chemo fail law school year debt age got pregnant gave birth month
daughter dy hour alive cousin age gave birth heal…
21839
karma hit today cake make
26355  last year sold beautiful condominium foolish use mind time cat got sick
one passed away one week later due move new vet killed got sick needed help
given taken mental hospital police vet told poli…
12034  suicidal feel life worthlesshey reddit thought id clarify thing bat im
suicidal though contemplated concept action several time ive never cut severely
depressed least think exactly plea help advic…
3933   take med harm others harm deep feel like nasty person narcissist
compassionmonday tried hang bear situation anymore wanted easy way avoid harming
```

people also avoid harmed peoplewhen got chair hang…
14574
scocity revoled around deathwhy redflag much problem today past
6021
accept fact literally funny secy struggle cant
20925
little brother wasnt born kill myselfbut need wait fucking year adult
314
talk favorite theory
17378 im fucking lonely driving insanei cant anything anymore without period completely space start cry cant find soul want around life matter okay happy normal act
15376 im mad sad dont know friend moved since kicked told stay till enough money move hasnt even full week since lived brought gf sleep queen bed ready go bed room sitting outside currently havent slept…
5554 worry much life worry always hating can not keep people want think want matter many time time person tell worry care can not listen like parent mentioned price therapy care pain l cared burden exi…
11866
sister teaching dad tiktok dance guy help edit step mom help getting hand
24358 start first job tomorrow work subway coworker probably definitely user reddit put hand together called woman culture met atla shirt im excited cool coworker amd cool bos good paying job excited ho…
26465 many american peabrains worship support political halftruths huckster like michael moore well sit movie see hypnotic manipulator scare intimidate lie underinformed public get people fear loathe ki…
13681
painless poisonsdrugs redflagi need know quickly please let know
17242
im tired seeing chadwick boseman post every second ok sad guy stop flooding page please mourn peace
8707
push offwatch drown slave moon waiting drown
10649 ok remember july th tik tok supposed banned august th tik tok supposed banned apparently tik tok banned sunday mean spying facebook banned well
10951 one question im suicidal feel insulted everytime someone dropping line like hey let talk get better promise come people get people like done talking thing got worse year curious reading line bring…
11810
wake fall asleep fall asleep wake uphelp answer
4161 venting also storybeen feeling like since felt useless bullied people take anger funny god im fuckup worst decision life ever bully someone come back bite turned friend lost everyone found trans c…
18247 happeninghi first greeting every human im going die tonight im couple minute away local train station im going walk track im halfway station im going kneel track see next train coming im posting s…

```
       text_length
10110           22
16118            8
4336             7
7496           147
7954            84
25377           40
17739            8
1762           132
21839            5
26355           69
12034          238
3933            86
14574            9
6021             7
20925           11
314              3
17378           25
15376           41
5554            39
11866           13
24358           53
26465           95
13681            9
17242           17
8707             7
10649           24
10951           36
11810            8
4161            52
18247           72
```

[35]:
```
df.label.value_counts()
```

[35]:
```
label
0    14074
1    13798
Name: count, dtype: int64
```

[36]:
```
# Define data
data = {
    'Label': ["Non-mental-health", "Mental-health"],
    'Label Encoded': [0,1]
}

# Create DataFrame
```

```
dr = pd.DataFrame(data)

# Print DataFrame
dr
```

[36]:

	Label	Label Encoded
0	Non-mental-health	0
1	Mental-health	1

```
class_names=dr.Label.to_list()
class_names
```

[37]: ['Non-mental-health', 'Mental-health']

1.6 Define Features, X & Labels, y

```
X = df['text3'].to_numpy()
y = df['label'].to_numpy()
```

1.7 Split the Data

```
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2,␣
↪stratify=y, random_state=42)
```

```
X_train.shape, X_valid.shape, y_train.shape, y_valid.shape
```

[40]: ((22297,), (5575,), (22297,), (5575,))

```
# Check the lengths
len(X_train), len(X_valid), len(y_train), len(y_valid)
```

[41]: (22297, 5575, 22297, 5575)

```
X_train
```

[45]: array(['im scared myselfi live life filled depression go school ignored come home ignored want someone hear im almost done everything im close breaking point able turn back im tired ignored want hear',
'ive felt dead longer ive felt alivei never happy kid painfully vivid memory raped young kid memory head slammed wall floor even distinct memory concussion one time slammed hard ground head ended bouncing hitting metal support bed day im scared men man imy first attempt kill around time failed inexperienced kid second time tried kill around time caught complete mental breakdown mom stay local mental hospital day try get help opened nice lady help rationalize fear hospital feeling broken tool ready thrown away butthats also gave trying thing general stopped trying school beyond minimum effort required pas stopped really associating friend school everything painfully mediocre year around time parent finally let liberty complaining year never hung friend like

ever let first place helped one school friend become one best friend able hang outside school talk demon dare share school ground made u feel sense solidarity one another also met girl ex one friend kept trying push u together put like black people like crushed heart bit sort gave pursuing focused friend instead always use somehow ended falling anyways one friend tell went asked birthday party two u standing outside moonlight together sound happy right thought well month passed found cheating another dude blamed entire thing deserve loved ended attempt number three point best friend rely pull back life went blissfully uneventful around time id grown entirely discontent monotony life hated every second planned kill halloween day banking hope people assume im decoration sort really care late instead ended hitting really great girl distracted demon otherwise attempt number four two year later clusterfuck emotion turmoil cant help wonder survived really close family started die godmother loved people died really young cancer ruined thought one role model ended taking life noticeable struggle mental health better part year physical health started take dive deep end ive got crohn day struggle able get bed pain sometimes much struggle even eat know stomach throw entire fit combine constant stream migraine headache ended spending nearly every day excruciating pain people told get help time wanted afford insurance aging mom working minimum wage job ever schedule part time premium always around month got denied reduction assistance forced choose eating place sleep able fix body theni ended getting rape accusation levied mind person claim time allegedly went placeand someone always hated physical contact full trauma one believed aside crazy aunt made remember memory happened memory long tried lock away broke made hateful distrustful already ended affecting performance work oftentimes depressive instead usual fake cheery self physically energy keep act anymore people company conspired get fired worked jobless insuranceless painfilled girl hit halloween decided right got fired want deal depression anymore dumped shattered whole world served light hope yearsmade feel sense happiness belonging otherwise felt dead unwanted reached redflag prevention hotline three separate occasion make feel better moment long run point best friend grown apart lot found another group friend vibe theyre recreational drug use stay sober idea might affect already fractured psyche december st attempt number spent day darkness tear wishing release physical pain mental anguish left house run away somewhere kill cousin caught wind plan waiting outside stop forced talk parent bothering year old black man broke pile tear floor explained want die even happen day end happening point day laterand sentiment gotten stronger know exactly know make year hope ill finally free lifetime pain another option redflag welcome point feel ive exhausted every option gain thank hearing guy',

'one last post night strange stupid questionsigh firstly want thank everyone helping still guy cant guarantee tomorrow still alive also sick dog freaking terrible head cold can say shitty immune system thanks anorexia stupid question hear term mentally ill thrown quite bit day hear term think schizophrenia psychosisthe bad thing know yall doctor looking definitive medical diagnosis anything get next week go appointment surefinally concrete name name mentally ill fuck wrong blah already dx officially ago anorexia nervosa know disability otherwise idea wrong meh wondering opinion thats uu hate feeling need

```
sleep happening though damn insomnia lg',
       …,
       'broke w boyfriend im sad fuck feeling fuck stupid robot deleting post co
doesnt enough word enough word u rusty little shit',
       'first wet dream really really bad timing st need include background info
girlfriend year going college musical theater order get college there ton
audition go super stressful also job taking relatively difficult class school
knowing backed settled hanging weekend sometimes every weekend also taking home
learning person relatively little contact u aside nightly call well nowhere
around christmas asked take break super stressed said ok said week ton work
didnt mind want happy well week went got thinking happy felt great reason feel
stressed relationship felt guilty cheating approached asked admitted maybe
started feeling something one coworkers fuck ok im upset see spending hour
someone time week might make feel something towards especially never see current
partner explained wasnt mad didnt tell issue know start spending time plus im
person school ill start seeing talking daily accepted point wanted decide wanted
continue start something coworker fuck x well day every single day ive wet dream
wtf never get now body bring right im hurting tldr dumb stupid horny teenage
brain kick im note think good chance getting back weve never big argument treat
wonderfully though im hard time competing someone see frequently also im hoping
none friend see hi thomas',
       'find struggling find help deal can not may one reaching sub allowed deal
struggle listening trying help struggle thank keeping want thank'],
      dtype=object)
```

[46]:
```
y_train, y_valid
```

[46]:
```
(array([1, 1, 1, …, 0, 0, 1]), array([0, 0, 1, …, 1, 0, 1]))
```

To determine the typical length of training texts, we calculate the average number of tokens (words) per text. This measurement aids in selecting an appropriate input size for neural network models, ensuring they are tailored to handle the data effectively.

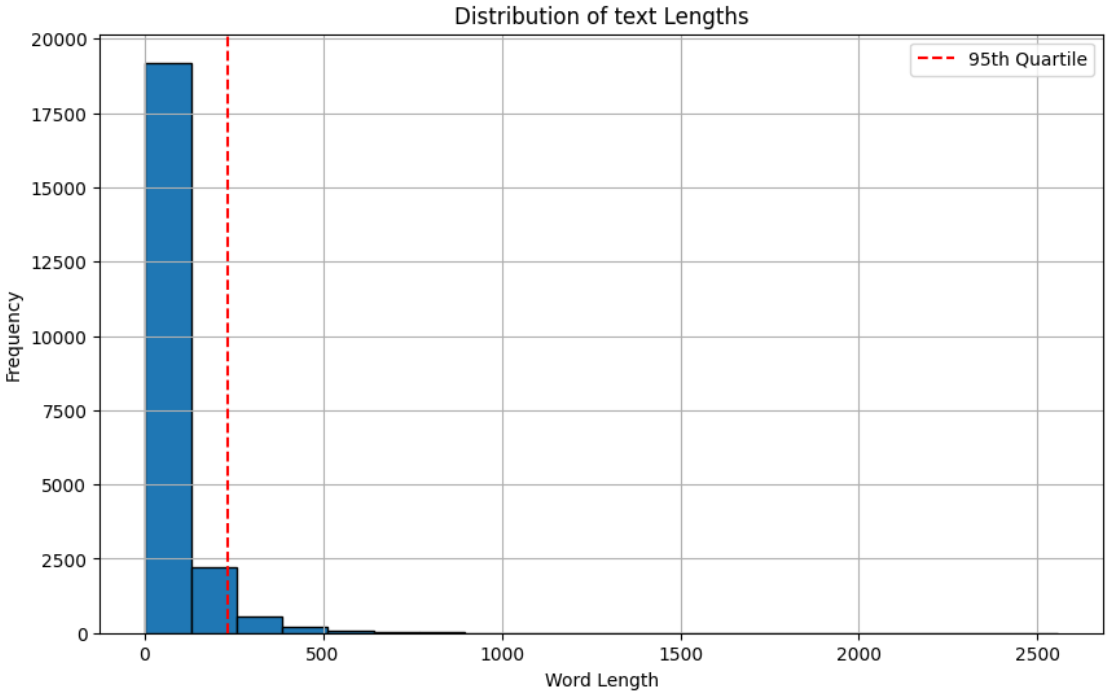
[47]:
```
# Find average number of tokens (words) in training texts
round(sum([len(i.split()) for i in X_train])/len(X_train))
```

[47]:
```
68
```

[48]:
```
# Calculate the length of each text in X_train
text_lengths = [len(text.split()) for text in X_train]

# Find the 98th percentile
percentile_95 = np.percentile(text_lengths, 95)

print(f"95th Percentile of Text Lengths: {percentile_95}")
```

```
95th Percentile of Text Lengths: 232.0
```

[49]:
```
# Plotting the histogram
plt.figure(figsize=(10, 6))
plt.hist(text_lengths, bins=20, edgecolor='black')
plt.xlabel('Word Length')
plt.ylabel('Frequency')
plt.title('Distribution of text Lengths')

# Adding a vertical line for the 95th quartile
quartile_95 = np.percentile(text_lengths, 95)
plt.axvline(x=quartile_95, color='red', linestyle='--', label='95th Quartile')
plt.legend()

plt.grid(True)
plt.show()
```



We calculate the maximum text length to identify the longest sequence that the model can process. This is crucial for deciding how to pad or truncate sequences during the preprocessing stage, ensuring all inputs are uniform in length.

[50]:
```
max_text_length = max(text_lengths)
print(f"Maximum Text Length: {max_text_length}")
```

```
Maximum Text Length: 2556
```

1.8 Text Vectorization

[51]:
```
import tensorflow as tf
from tensorflow.keras.layers import TextVectorization

# Setup text vectorization with custom variables
max_vocab_length =None # max number of words to have in our vocabulary
max_length = int(percentile_95) # max length our sequences will be (e.g. how many words from a text does our model see?)

text_vectorizer = TextVectorization(max_tokens=max_vocab_length,
                                    output_mode="int",
                                    output_sequence_length=max_length)
```

```
2024-04-13 15:55:18.783273: I metal_plugin/src/device/metal_device.cc:1154]
Metal device set to: Apple M2 Max
2024-04-13 15:55:18.783318: I metal_plugin/src/device/metal_device.cc:296]
systemMemory: 32.00 GB
2024-04-13 15:55:18.783325: I metal_plugin/src/device/metal_device.cc:313]
maxCacheSize: 10.67 GB
2024-04-13 15:55:18.783364: I
tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:305]
Could not identify NUMA node of platform GPU ID 0, defaulting to 0. Your kernel
may not have been built with NUMA support.
2024-04-13 15:55:18.783394: I
tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:271]
Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 0
MB memory) -> physical PluggableDevice (device: 0, name: METAL, pci bus id:
<undefined>)
```

The text vectorizer is fitted to the training text to build the vocabulary based on the training data. This allows the vectorizer to learn the mapping between words and their integer representations.

[52]:
```
# Fit the text vectorizer to the training text
text_vectorizer.adapt(X_train)
```

[53]:
```
# Get the unique words in the vocabulary
words_in_vocab = text_vectorizer.get_vocabulary()
top_5_words = words_in_vocab[:5] # most common tokens (notice the [UNK] token for "unknown" words)
bottom_5_words = words_in_vocab[-5:] # least common tokens
print(f"Number of words in vocab: {len(words_in_vocab)}")
print(f"Top 5 most common words: {top_5_words}")
print(f"Bottom 5 least common words: {bottom_5_words}")
```

```
Number of words in vocab: 58460
Top 5 most common words: ['', '[UNK]', 'im', 'like', 'want']
Bottom 5 least common words: ['aaaaaaaaaaaaaaaaaaaa', 'aaaaaaaaaaaaaaaaa',
'aaaaaaaaaaaaaa', 'aaaaaaaa', 'aaaaa']
```

The maximum vocabulary length is updated with the actual length of the vocabulary obtained from the text vectorizer.

[54]:
```
max_vocab_length=len(words_in_vocab)
```

2 Model: Baseline

[55]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline

# Create tokenization and modelling pipeline
baseline_model = Pipeline([
                    ("tfidf", TfidfVectorizer()), # convert words to numbers␣
 ↪using tfidf
                    ("clf", MultinomialNB()) # model the text
])

# Now fit the model
baseline_model.fit(X_train, y_train)
```

[55]:
```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', MultinomialNB())])
```

[56]:
```
baseline_score = baseline_model.score(X_valid, y_valid)
print(f" Baseline model achieves an accuracy of: {baseline_score*100:.2f}%")
```

```
 Baseline model achieves an accuracy of: 85.43%
```

[57]:
```
# Make predictions
baseline_preds = baseline_model.predict(X_valid)
baseline_preds[:20]
```

[57]:
```
array([0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0])
```

Then a function calculate_results is defined to calculate the accuracy, precision, recall, and F1 score of the model's predictions.

[58]:
```
# Function to evaluate: accuracy, precision, recall, f1-score
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

def calculate_results(y_true, y_pred):

  # Calculate model accuracy
  model_accuracy = accuracy_score(y_true, y_pred) * 100
  # Calculate model precision, recall and f1 score using "weighted" average
  model_precision, model_recall, model_f1, _ =␣
 ↪precision_recall_fscore_support(y_true, y_pred, average="weighted")
  model_results = {"accuracy": model_accuracy,
```

```
                  "precision": model_precision,
                  "recall": model_recall,
                  "f1": model_f1}
  return model_results
```

The function is used to calculate the results of the baseline model and these are printed out.

[59]:
```
# baselineModel results
baseline_results = calculate_results(y_true=y_valid,
                                     y_pred=baseline_preds)
baseline_results
```

[59]:
```
{'accuracy': 85.43497757847534,
 'precision': 0.881010158721504,
 'recall': 0.8543497757847534,
 'f1': 0.8519790697475407}
```

[60]:
```
# Create a helper function to compare our baseline results to new model results
def compare_baseline_to_new_results(baseline_results, new_model_results):
  for key, value in baseline_results.items():
    print(f"Baseline {key}: {value:.2f}, New {key}: {new_model_results[key]:.
 ↪2f}, Difference: {new_model_results[key]-value:.2f}")
```

Callbacks

[61]:
```
from tensorflow.keras.callbacks import ModelCheckpoint

def create_checkpoint_callback(checkpoint_path):
    checkpoint_callback = ModelCheckpoint(filepath=checkpoint_path,
                                          monitor='val_accuracy',
                                          mode='max',
                                          save_best_only=True,
                                          verbose=1)
    return checkpoint_callback
```

Embedding layer

[62]:
```
import tensorflow_hub as hub
from tensorflow.keras import layers
```

[65]:
```
# from tensorflow.keras import layers

tf.random.set_seed(42)

embedding = layers.Embedding(input_dim=max_vocab_length, # set input shape
                             output_dim=300, # set size of embedding vector
                             embeddings_initializer="uniform", # default,␣
 ↪intialize randomly
                             input_length=max_length, # how long is each input
```

```
                                   name="embedding_1")

embedding
```

```
/Users/raja/anaconda3/lib/python3.11/site-
packages/keras/src/layers/core/embedding.py:86: UserWarning: Argument
`input_length` is deprecated. Just remove it.
  warnings.warn(
```

[65]:
```
<Embedding name=embedding_1, built=False>
```

3 Model: LSTM

[67]:
```
# Set random seed and create embedding layer (new embedding layer for each␣
 ↪model)
tf.random.set_seed(42)
from tensorflow.keras import layers
lstm_model_embedding = layers.Embedding(input_dim=max_vocab_length,
                                        output_dim=300,
                                        embeddings_initializer="uniform",
                                        input_length=max_length,
                                        name="embedding_2")


# Create LSTM model
inputs = layers.Input(shape=(1,), dtype="string")
x = text_vectorizer(inputs)
x = lstm_model_embedding(x)
print(x.shape)
x = layers.LSTM(64, return_sequences=True)(x) # return vector for each word in␣
 ↪the text (you can stack RNN cells as long as return_sequences=True)
x = layers.LSTM(64)(x) # return vector for whole sequence
print(x.shape)
x = layers.Dense(64, activation="relu")(x) # optional dense layer on top of␣
 ↪output of LSTM cell
outputs = layers.Dense(1, activation="sigmoid")(x)
lstm_model = tf.keras.Model(inputs, outputs, name="lstm_model")
```

```
(None, 232, 300)
(None, 64)
```

[68]:
```
# Compile model
lstm_model.compile(loss="binary_crossentropy",
                   optimizer=tf.keras.optimizers.Adam(),
                   metrics=["accuracy"])
```

[69]:
```
lstm_model.summary()
```

```
Model: "lstm_model"
```

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 1)	0
text_vectorization (TextVectorization)	(None, 232)	0
embedding_2 (Embedding)	(None, 232, 300)	17,538,000
lstm_2 (LSTM)	(None, 232, 64)	93,440
lstm_3 (LSTM)	(None, 64)	33,024
dense_2 (Dense)	(None, 64)	4,160
dense_3 (Dense)	(None, 1)	65

```
Total params: 17,668,689 (67.40 MB)

Trainable params: 17,668,689 (67.40 MB)

Non-trainable params: 0 (0.00 B)
```

We now create a checkpoint callback for model LSTM.

[70]:
```
# Define the checkpoint path
checkpoint_path = "best_model_Bi-LSTM.keras"

cc = create_checkpoint_callback(checkpoint_path)
```

[71]:
```
# Fit model
lstm_model_history = lstm_model.fit(X_train, y_train,
                                    epochs=10,
                                    validation_data=(X_valid, y_valid),
                                    callbacks=[cc])
```

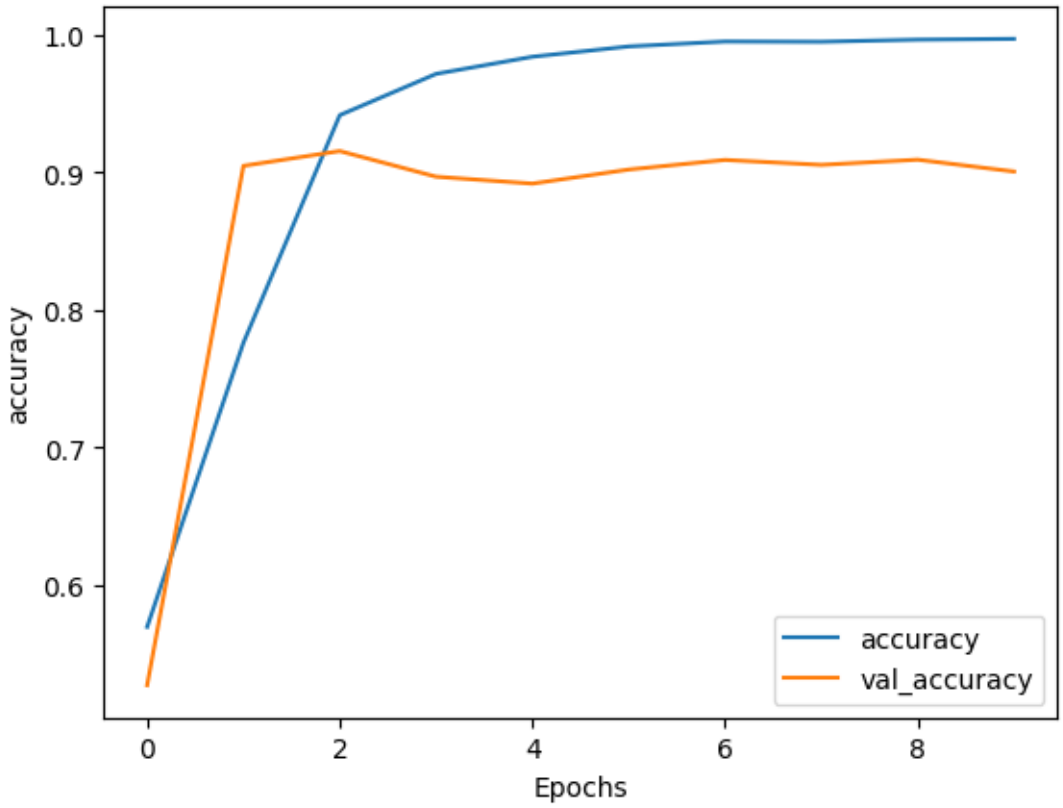
```
Epoch 1/10

2024-04-13 16:11:06.790205: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:117]
Plugin optimizer for device_type GPU is enabled.
```

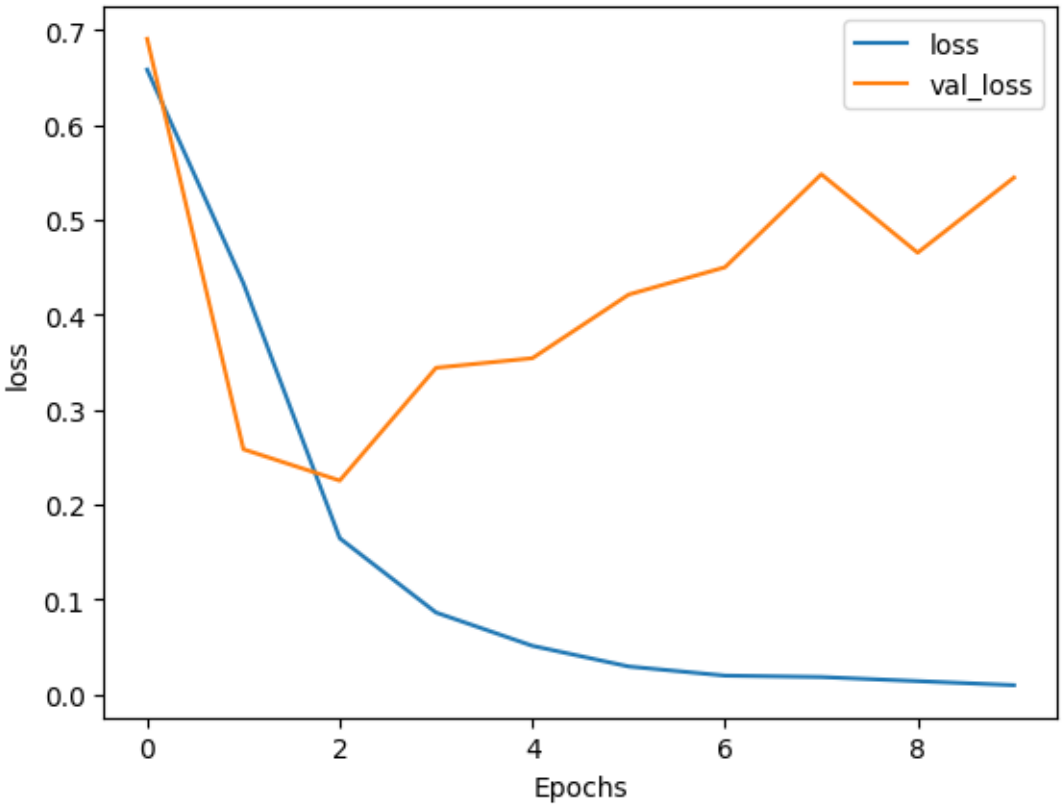
```
697/697            0s 52ms/step -
accuracy: 0.5370 - loss: 0.6790
Epoch 1: val_accuracy improved from -inf to 0.52700, saving model to
best_model_Bi-LSTM.keras
697/697            43s 60ms/step -
accuracy: 0.5370 - loss: 0.6790 - val_accuracy: 0.5270 - val_loss: 0.6909
Epoch 2/10
697/697            0s 57ms/step -
accuracy: 0.6397 - loss: 0.5831
Epoch 2: val_accuracy improved from 0.52700 to 0.90475, saving model to
best_model_Bi-LSTM.keras
697/697            45s 64ms/step -
accuracy: 0.6398 - loss: 0.5829 - val_accuracy: 0.9048 - val_loss: 0.2584
Epoch 3/10
697/697            0s 57ms/step -
accuracy: 0.9253 - loss: 0.1997
Epoch 3: val_accuracy improved from 0.90475 to 0.91552, saving model to
best_model_Bi-LSTM.keras
697/697            45s 65ms/step -
accuracy: 0.9253 - loss: 0.1997 - val_accuracy: 0.9155 - val_loss: 0.2254
Epoch 4/10
697/697            0s 58ms/step -
accuracy: 0.9667 - loss: 0.0981
Epoch 4: val_accuracy did not improve from 0.91552
697/697            42s 61ms/step -
accuracy: 0.9668 - loss: 0.0981 - val_accuracy: 0.8969 - val_loss: 0.3442
Epoch 5/10
697/697            0s 57ms/step -
accuracy: 0.9824 - loss: 0.0576
Epoch 5: val_accuracy did not improve from 0.91552
697/697            42s 60ms/step -
accuracy: 0.9824 - loss: 0.0575 - val_accuracy: 0.8918 - val_loss: 0.3543
Epoch 6/10
697/697            0s 56ms/step -
accuracy: 0.9906 - loss: 0.0329
Epoch 6: val_accuracy did not improve from 0.91552
697/697            41s 59ms/step -
accuracy: 0.9906 - loss: 0.0329 - val_accuracy: 0.9021 - val_loss: 0.4214
Epoch 7/10
697/697            0s 56ms/step -
accuracy: 0.9947 - loss: 0.0219
Epoch 7: val_accuracy did not improve from 0.91552
697/697            41s 59ms/step -
accuracy: 0.9947 - loss: 0.0219 - val_accuracy: 0.9091 - val_loss: 0.4502
Epoch 8/10
697/697            0s 56ms/step -
accuracy: 0.9958 - loss: 0.0168
Epoch 8: val_accuracy did not improve from 0.91552
```

```
697/697             41s 59ms/step -
accuracy: 0.9958 - loss: 0.0168 - val_accuracy: 0.9055 - val_loss: 0.5482
Epoch 9/10
697/697             0s 56ms/step -
accuracy: 0.9970 - loss: 0.0131
Epoch 9: val_accuracy did not improve from 0.91552
697/697             41s 59ms/step -
accuracy: 0.9970 - loss: 0.0131 - val_accuracy: 0.9092 - val_loss: 0.4656
Epoch 10/10
697/697             0s 56ms/step -
accuracy: 0.9958 - loss: 0.0142
Epoch 10: val_accuracy did not improve from 0.91552
697/697             41s 59ms/step -
accuracy: 0.9958 - loss: 0.0142 - val_accuracy: 0.9006 - val_loss: 0.5447
```

Following training, the history of LSTM model's accuracy and loss over the epochs is plotted.

[72]:
```python
# Plot Utility
def plot_graphs(history, string):
  plt.plot(history.history[string])
  plt.plot(history.history['val_'+string])
  plt.xlabel("Epochs")
  plt.ylabel(string)
  plt.legend([string, 'val_'+string])
  plt.show()

# Plot the accuracy and loss history
plot_graphs(lstm_model_history, 'accuracy')
plot_graphs(lstm_model_history, 'loss')
```


1.0
0.9
0.8
0.7
0.6
accuracy
0
2
4
6
8
Epochs
accuracy
val_accuracy



[74]:
```
from tensorflow.keras.models import load_model

# Load the entire model
lstm_model = load_model(checkpoint_path)
```

The LSTM model is evaluated on the validation set to understand its performance on unseen data.

[75]:
```
lstm_model.evaluate(X_valid, y_valid)
```

```
175/175            3s 13ms/step -
accuracy: 0.9104 - loss: 0.2309
```

[75]:
```
[0.22536934912204742, 0.9155157208442688]
```

[76]:
```
# Make predictions on the validation dataset
lstm_model_pred_probs = lstm_model.predict(X_valid)
lstm_model_pred_probs.shape, lstm_model_pred_probs[:10] # view the first 10
```

```
175/175            2s 8ms/step
```

[76]:
```
((5575, 1),
 array([[0.00646534],
        [0.00192491],
```

```
       [0.9809474 ],
       [0.03597093],
       [0.67051554],
       [0.97818726],
       [0.98397255],
       [0.98496115],
       [0.7622405 ],
       [0.22307119]], dtype=float32))
```

[77]:
```
# Convert prediction probabilities to labels
lstm_model_preds = tf.squeeze(tf.round(lstm_model_pred_probs))
lstm_model_preds[:10]
```

[77]:
```
<tf.Tensor: shape=(10,), dtype=float32, numpy=array([0., 0., 1., 0., 1., 1., 1.,
1., 1., 0.], dtype=float32)>
```

Metrics such as accuracy, precision, recall, and F1-score are calculated to evaluate the performance of the LSTM model.

[78]:
```
# Calculate LSTM model results
lstm_model_results = calculate_results(y_true=y_valid,
                                       y_pred=lstm_model_preds)
lstm_model_results
```

[78]:
```
{'accuracy': 91.55156950672647,
 'precision': 0.9155406811826331,
 'recall': 0.9155156950672646,
 'f1': 0.9155175544145038}
```

The function compares the performance metrics of the baseline model with the LSTM model. The comparison include various metrics such as accuracy, precision, recall, and F1-score.

[79]:
```
# Compare model ltsm to baseline
compare_baseline_to_new_results(baseline_results, lstm_model_results)
```

```
Baseline accuracy: 85.43, New accuracy: 91.55, Difference: 6.12
Baseline precision: 0.88, New precision: 0.92, Difference: 0.03
Baseline recall: 0.85, New recall: 0.92, Difference: 0.06
Baseline f1: 0.85, New f1: 0.92, Difference: 0.06
```

[80]:
```
y_true = y_valid.tolist()  # Convert labels to a list
preds = lstm_model.predict(X_valid)
y_probs = preds.squeeze().tolist()  # Store the prediction probabilities as a␣
↪list
y_preds = tf.round(y_probs).numpy().tolist()  # Convert probabilities to class␣
↪predictions and convert to a list
```

```
175/175             1s 7ms/step
```

A confusion matrix is generated to visualize the classification performance of the LSTM model. A custom function is used to make the matrix more readable.

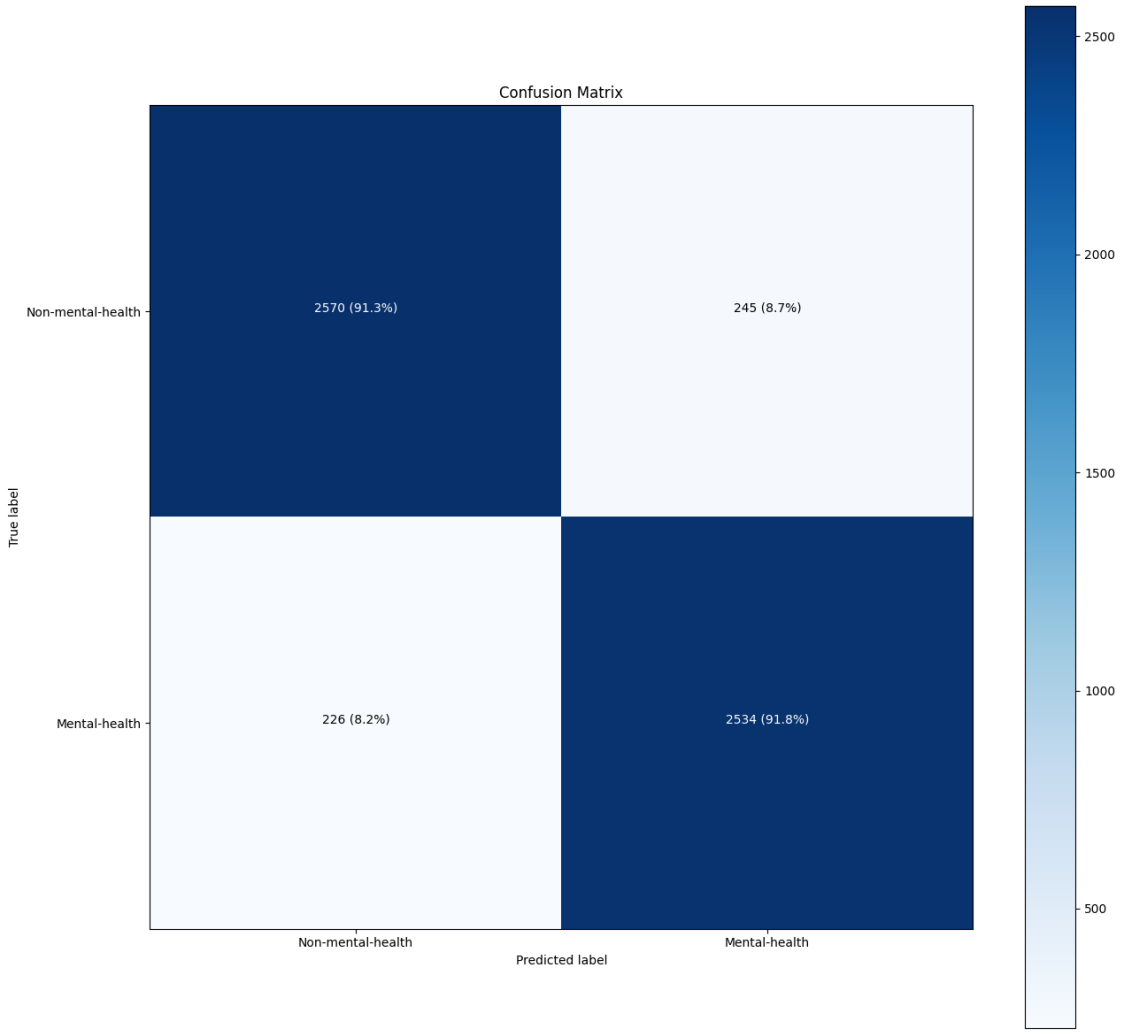
[81]:
```
# Check out the non-prettified confusion matrix
from sklearn.metrics import confusion_matrix
confusion_matrix(y_true=y_true,
                 y_pred=y_preds)
```

[81]:
```
array([[2570,  245],
       [ 226, 2534]])
```

[83]:
```
import itertools
from sklearn.metrics import confusion_matrix

# Our function needs a different name to sklearn's plot_confusion_matrix
def make_confusion_matrix(y_true, y_pred, classes=None, figsize=(10, 10),␣
 ↪text_size=15):

  # Create the confustion matrix
  cm = confusion_matrix(y_true, y_pred)
  cm_norm = cm.astype("float") / cm.sum(axis=1)[:, np.newaxis] # normalize it
  n_classes = cm.shape[0] # find the number of classes we're dealing with

  # Plot the figure and make it pretty
  fig, ax = plt.subplots(figsize=figsize)
  cax = ax.matshow(cm, cmap=plt.cm.Blues) # colors will represent how 'correct'␣
 ↪a class is, darker == better
  fig.colorbar(cax)

  # Are there a list of classes?
  if classes:
    labels = classes
  else:
    labels = np.arange(cm.shape[0])

  # Label the axes
  ax.set(title="Confusion Matrix",
         xlabel="Predicted label",
         ylabel="True label",
         xticks=np.arange(n_classes), # create enough axis slots for each class
         yticks=np.arange(n_classes),
         xticklabels=labels, # axes will labeled with class names (if they␣
 ↪exist) or ints
         yticklabels=labels)

  # Make x-axis labels appear on bottom
  ax.xaxis.set_label_position("bottom")
```

```
  ax.xaxis.tick_bottom()

  # Set the threshold for different colors
  threshold = (cm.max() + cm.min()) / 2.

  # Plot the text on each cell
  for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, f"{cm[i, j]} ({cm_norm[i, j]*100:.1f}%)",
             horizontalalignment="center",
             color="white" if cm[i, j] > threshold else "black",
             size=text_size)
```

[84]:
```
# Make a prettier confusion matrix
make_confusion_matrix(y_true=y_true,
                      y_pred=y_preds,
                      classes=class_names,
                      figsize=(15, 15),
                      text_size=10)
```



[86]:
```
!pip install colorama
from colorama import Fore, Style
import numpy as np

def random_predictions(model, X_valid, y_valid, num_samples=5,␣
 ↪class_names=None):
    # Check if it's binary or multi-class classification
    is_binary_classification = len(np.unique(y_valid)) == 2

    # Getting indices of the random samples
    random_indices = np.random.choice(np.arange(len(X_valid)),␣
 ↪size=num_samples, replace=False)

    # Selecting the random samples
    random_X_samples = X_valid[random_indices]
    random_y_samples = y_valid[random_indices]

    # Making predictions on the random samples
    y_pred_probs = model.predict(random_X_samples)

    if is_binary_classification:
        y_pred = np.squeeze(np.round(y_pred_probs).astype(int))
    else:
        y_pred = np.argmax(y_pred_probs, axis=1)

    # Print the actual and predicted labels
    for i in range(num_samples):
        text = random_X_samples[i]
        true_label = random_y_samples[i] if is_binary_classification else np.
 ↪argmax(random_y_samples[i])
        predicted_label = y_pred[i]

        # If class names are provided, use them for printing
        if class_names is not None:
            true_label_name = class_names[true_label]
            predicted_label_name = class_names[predicted_label]
        else:
            true_label_name = true_label
            predicted_label_name = predicted_label

        # Determine the color of the text (green for correct, red for incorrect)
        text_color = Fore.GREEN if true_label == predicted_label else Fore.RED

        print(f"\nSample {i + 1}:")
```

```
        print(f"Text: {text}")
        print(text_color + f"True: {true_label_name} \n Predicted:␣
↪{predicted_label_name}" + Style.RESET_ALL)
```

```
Requirement already satisfied: colorama in
/Users/raja/anaconda3/lib/python3.11/site-packages (0.4.6)
```

[87]:
```
random_predictions(lstm_model,
                   X_valid,
                   y_valid,
                   num_samples=20,
                   class_names=class_names)
```

```
1/1             0s 71ms/step

Sample 1:
Text: divorcewell look like im ruled doesnt matter ill dead make broke take kid
True: Mental-health

 Predicted: Non-mental-health

Sample 2:
Text: slide done english doesnt suck much took hour slide hey progress finding
image kinda fun
True: Non-mental-health

 Predicted: Non-mental-health

Sample 3:
Text: asian pussy tight really wan na pipe oh god big as tiddys five
True: Non-mental-health

 Predicted: Non-mental-health

Sample 4:
Text: count suicidal know fucking scared pain actually go anythingi want die
want final minuteshours filled excruciating pain thing pill knife id never able
idk guess needed say something want go sleep wake sorry waste everyones time
True: Mental-health

 Predicted: Mental-health

Sample 5:
Text: im tiredim tired life im tired alone im tired horrible world want end
True: Mental-health

 Predicted: Mental-health

Sample 6:
Text: losing continue husband last violent outburstlike title say already brink
```

husband destroyed trust really know
True: Mental-health

Predicted: Mental-health

Sample 7:
Text: really want go back hospitali difficulty expressing text person ive pretty much shut anything everyone since january med therapy year even treatment think redflag daily get frustrated see change today rent due money landlord accepts money order seems think secure way make payment cant seem motivate go bank get printed im going common argument even worth effort know simple thing hate always somehow convince simplest thing part know need cant seem find maybe posting help see stupid im pay rent try deal stuff
True: Mental-health

Predicted: Mental-health

Sample 8:
Text: every try type idk accidently say dik heehee funni
True: Non-mental-health

Predicted: Non-mental-health

Sample 9:
Text: feel sickto get bit understanding read httpwwwredditcomraskredditcommentsplketgayteeninneedhelp fast forward month life back normal except feel sick physically sort way ever since felt pretty intense self loathing want turn feel sick hey reddit cooky whoever make happy
True: Mental-health

Predicted: Mental-health

Sample 10:
Text: please talk someoneforget
True: Mental-health

Predicted: Non-mental-health

Sample 11:
Text: someone take blue cheese smell bad want
True: Non-mental-health

Predicted: Non-mental-health

Sample 12:
Text: spilt koolaid carpet making koolaid spilled stain im fucked
True: Non-mental-health

Predicted: Non-mental-health

Sample 13:

Text: know feeling one day pain much end life really painful everything hurt die suicide
True: Mental-health

Predicted: Mental-health

Sample 14:
Text: fantasy never realyou ever see movie think yourselfdamn wish person life full meaning adventure look real world see nothing empty boring life within
True: Mental-health

Predicted: Mental-health

Sample 15:
Text: father making suffer entirety middle school school ive made many good friend relationship came gone year emotionally destroyed dad retiring want go back home town forcing whole family go finished grade entering grade school nearly year ripped apart everybody old man fantasy im seeking emotional support whole family telling man stop baby
True: Non-mental-health

Predicted: Non-mental-health

Sample 16:
Text: noticed scar hip fake smile lip forced laugh adopted way care thing used love dare stand grave cryhow cry someone even know suicide note
True: Mental-health

Predicted: Mental-health

Sample 17:
Text: janam janam janam sath chalna younhi kasam tumhein kasam akay milna younhi el el
True: Non-mental-health

Predicted: Non-mental-health

Sample 18:
Text: really need someone talk toif aim pm ill give sni really appreciate advice comfortanything really joke
True: Mental-health

Predicted: Non-mental-health

Sample 19:
Text: baby dancin shes dancin another man
True: Non-mental-health

Predicted: Non-mental-health

Sample 20:

```
Text: even care anymoreim fat cant get fit im sad cant get happy nothing matter
year whats even point anything get feeling last minute wish reset button life
different person different family different country probably live see next solar
eclipse world turning trash
True: Mental-health

 Predicted: Mental-health
```

The model_lstm is fit to the training data (X_train and y_train) for 10 epochs, with validation data (X_valid and y_valid) used for evaluation. The training progress is recorded in history_lstm, and the defined callbacks (cc) are utilized during training.

Post-training, the model's accuracy and loss evolution across epochs is visualized.

[88]:
```
lstm_model.evaluate(X_valid, y_valid)
```

```
175/175             2s 12ms/step -
accuracy: 0.9104 - loss: 0.2309
```

[88]:
```
[0.22536934912204742, 0.9155157208442688]
```

Class predictions are generated by transforming predicted probabilities on the validation dataset.

4 Model: GRU

[89]:
```
# Set random seed and create embedding layer (new embedding layer for each␣
 ↪model)
tf.random.set_seed(42)

from tensorflow.keras import layers
model_GRU_embedding = layers.Embedding(input_dim=max_vocab_length,
                                       output_dim=128,
                                       embeddings_initializer="uniform",
                                       input_length=max_length,
                                       name="embedding_GRU")

# Build an RNN using the GRU cell
inputs = layers.Input(shape=(1,), dtype="string")
x = text_vectorizer(inputs)
x = model_GRU_embedding(x)
x = layers.GRU(64, return_sequences=True)(x)  # Add parentheses here
x = layers.GRU(64)(x)
x = layers.Dense(64, activation="relu")(x) # optional dense layer after GRU cell

outputs = layers.Dense(1, activation="sigmoid")(x)

model_GRU = tf.keras.Model(inputs, outputs, name="model_GRU")
```

```
/Users/raja/anaconda3/lib/python3.11/site-
packages/keras/src/layers/core/embedding.py:86: UserWarning: Argument
```

```
`input_length` is deprecated. Just remove it.
  warnings.warn(
```

The 'model_GRU' is compiled using the Adam optimizer and binary cross-entropy as the loss function, suitable for binary classification tasks.

[90]:
```
# Compile GRU model
model_GRU.compile(loss="binary_crossentropy",
                  optimizer=tf.keras.optimizers.Adam(),
                  metrics=["accuracy"])
```

[91]:
```
# Get a summary of the GRU model
model_GRU.summary()
```

```
Model: "model_GRU"
```

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 1)	0
text_vectorization (TextVectorization)	(None, 232)	0
embedding_GRU (Embedding)	(None, 232, 128)	7,482,880
gru (GRU)	(None, 232, 64)	37,248
gru_1 (GRU)	(None, 64)	24,960
dense_4 (Dense)	(None, 64)	4,160
dense_5 (Dense)	(None, 1)	65

```
Total params: 7,549,313 (28.80 MB)

Trainable params: 7,549,313 (28.80 MB)

Non-trainable params: 0 (0.00 B)
```

A checkpoint callback is created for the GRU model.

[92]:
```
# Define the checkpoint path
checkpoint_path = "best_model_GRU.keras"
```

```
cc = create_checkpoint_callback(checkpoint_path)
```

The improved is fit to the training data (X_train and y_train) for 10 epochs, with validation data (X_valid and y_valid) used for evaluation. The training progress is recorded in model_GRU_history, and the defined callbacks (cc) are utilized during training.

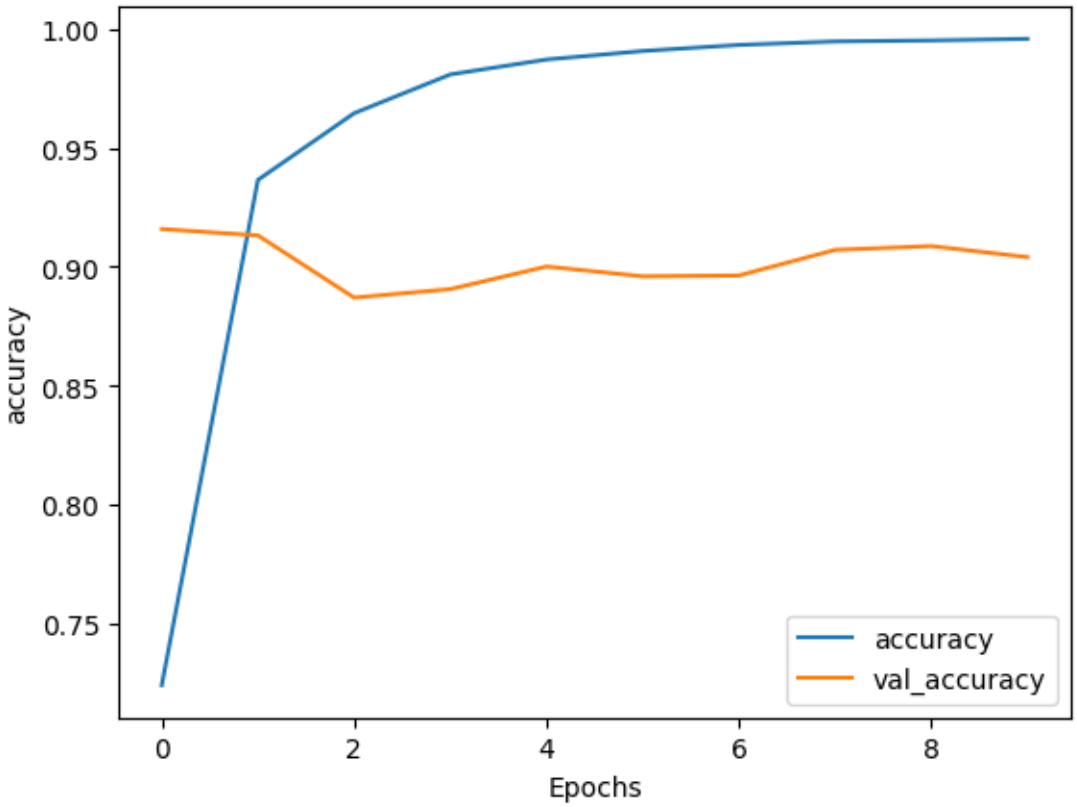
[93]:
```
# Fit model
model_GRU_history = model_GRU.fit(X_train, y_train,
                                  epochs=10,
                                  validation_data=(X_valid, y_valid),
                                  callbacks=[cc])
```

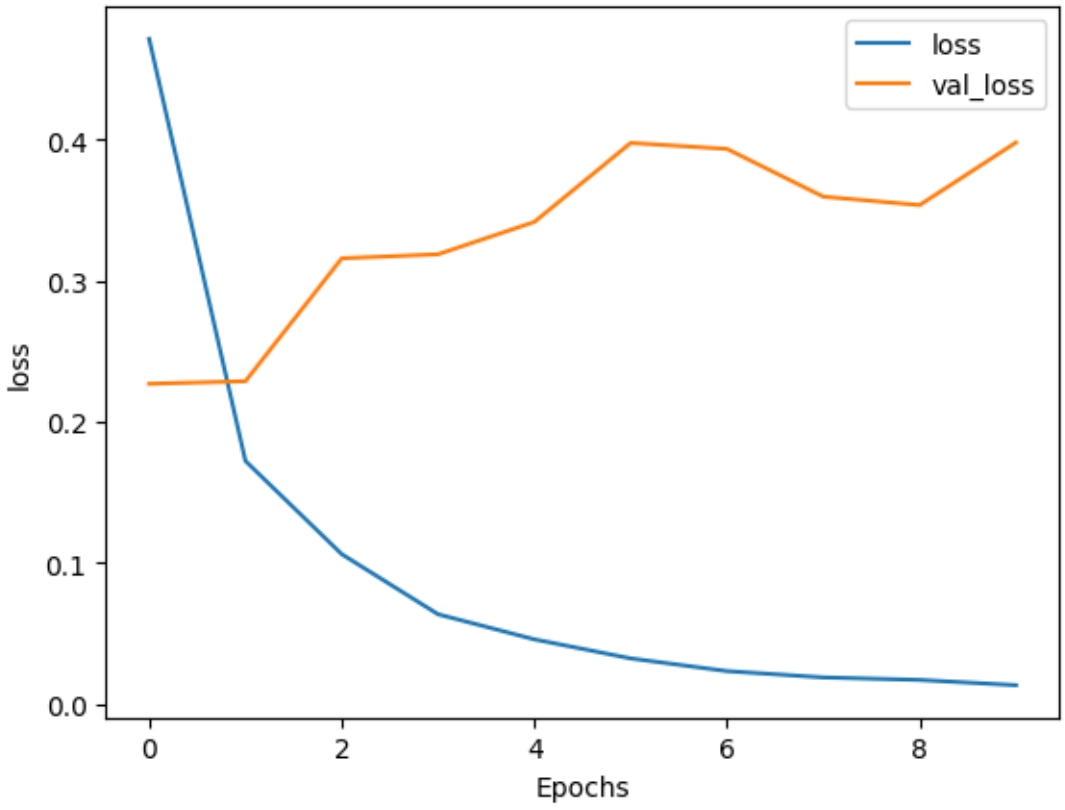
```
Epoch 1/10
697/697             0s 46ms/step -
accuracy: 0.5894 - loss: 0.6219
Epoch 1: val_accuracy improved from -inf to 0.91587, saving model to
best_model_GRU.keras
697/697             38s 53ms/step -
accuracy: 0.5896 - loss: 0.6217 - val_accuracy: 0.9159 - val_loss: 0.2267
Epoch 2/10
697/697             0s 49ms/step -
accuracy: 0.9235 - loss: 0.1994
Epoch 2: val_accuracy did not improve from 0.91587
697/697             36s 52ms/step -
accuracy: 0.9236 - loss: 0.1994 - val_accuracy: 0.9132 - val_loss: 0.2286
Epoch 3/10
696/697             0s 48ms/step -
accuracy: 0.9593 - loss: 0.1154
Epoch 3: val_accuracy did not improve from 0.91587
697/697             36s 51ms/step -
accuracy: 0.9594 - loss: 0.1154 - val_accuracy: 0.8870 - val_loss: 0.3157
Epoch 4/10
697/697             0s 49ms/step -
accuracy: 0.9786 - loss: 0.0696
Epoch 4: val_accuracy did not improve from 0.91587
697/697             36s 52ms/step -
accuracy: 0.9786 - loss: 0.0696 - val_accuracy: 0.8906 - val_loss: 0.3185
Epoch 5/10
696/697             0s 48ms/step -
accuracy: 0.9875 - loss: 0.0466
Epoch 5: val_accuracy did not improve from 0.91587
697/697             35s 51ms/step -
accuracy: 0.9875 - loss: 0.0466 - val_accuracy: 0.9001 - val_loss: 0.3415
Epoch 6/10
696/697             0s 48ms/step -
accuracy: 0.9913 - loss: 0.0321
Epoch 6: val_accuracy did not improve from 0.91587
697/697             35s 51ms/step -
```

```
accuracy: 0.9913 - loss: 0.0321 - val_accuracy: 0.8960 - val_loss: 0.3975
Epoch 7/10
696/697             0s 48ms/step -
accuracy: 0.9933 - loss: 0.0228
Epoch 7: val_accuracy did not improve from 0.91587
697/697             36s 51ms/step -
accuracy: 0.9933 - loss: 0.0228 - val_accuracy: 0.8963 - val_loss: 0.3932
Epoch 8/10
696/697             0s 48ms/step -
accuracy: 0.9951 - loss: 0.0166
Epoch 8: val_accuracy did not improve from 0.91587
697/697             36s 51ms/step -
accuracy: 0.9951 - loss: 0.0167 - val_accuracy: 0.9071 - val_loss: 0.3594
Epoch 9/10
697/697             0s 48ms/step -
accuracy: 0.9961 - loss: 0.0141
Epoch 9: val_accuracy did not improve from 0.91587
697/697             36s 51ms/step -
accuracy: 0.9961 - loss: 0.0141 - val_accuracy: 0.9087 - val_loss: 0.3534
Epoch 10/10
696/697             0s 49ms/step -
accuracy: 0.9970 - loss: 0.0093
Epoch 10: val_accuracy did not improve from 0.91587
697/697             36s 51ms/step -
accuracy: 0.9970 - loss: 0.0093 - val_accuracy: 0.9040 - val_loss: 0.3978
```

The model's accuracy and loss history is visualized post-training.

[94]:
```python
# Plot Utility
def plot_graphs(history, string):
  plt.plot(history.history[string])
  plt.plot(history.history['val_'+string])
  plt.xlabel("Epochs")
  plt.ylabel(string)
  plt.legend([string, 'val_'+string])
  plt.show()

# Plot the accuracy and loss history
plot_graphs(model_GRU_history, 'accuracy')
plot_graphs(model_GRU_history, 'loss')
```


1.00
0.95
0.90
0.85
0.80
0.75
accuracy
0
2
4
6
8
Epochs
accuracy
val_accuracy



[95]:
```
# Load the entire model
model_GRU = load_model(checkpoint_path)
```

Model evaluation occurs on the validation set.

[96]:
```
model_GRU.evaluate(X_valid, y_valid)
```

```
175/175             2s 12ms/step -
accuracy: 0.9120 - loss: 0.2298
```

[96]:
```
[0.22674039006233215, 0.9158744215965271]
```

The model predicts probabilities on the validation set, converting these into class predictions.

[97]:
```
# Make predictions on the validation data
model_GRU_pred_probs = model_GRU.predict(X_valid)
model_GRU_pred_probs.shape, model_GRU_pred_probs[:10]
```

```
175/175             2s 9ms/step
```

[97]:
```
((5575, 1),
 array([[0.32981187],
        [0.02121094],
```

```
[0.9546341 ],
[0.07093085],
[0.7601235 ],
[0.90935177],
[0.91980696],
[0.93996817],
[0.9445129 ],
[0.2996596 ]], dtype=float32))
```

[98]:
```
# Convert prediction probabilities to labels
model_GRU_preds = tf.squeeze(tf.round(model_GRU_pred_probs))
model_GRU_preds[:10]
```

[98]:
```
<tf.Tensor: shape=(10,), dtype=float32, numpy=array([0., 0., 1., 0., 1., 1., 1.,
1., 1., 0.], dtype=float32)>
```

Performance metrics, including accuracy, precision, recall, and F1-score, are computed for model evaluation.

[99]:
```
# Calcuate model_GRU results
model_GRU_results = calculate_results(y_true=y_valid,
                                      y_pred=model_GRU_preds)
model_GRU_results
```

[99]:
```
{'accuracy': 91.58744394618834,
 'precision': 0.9165497934850814,
 'recall': 0.9158744394618834,
 'f1': 0.9158214744588596}
```

The baseline model's performance is compared with the GRU model.

[100]:
```
# Compare to baseline
compare_baseline_to_new_results(baseline_results, model_GRU_results)
```

```
Baseline accuracy: 85.43, New accuracy: 91.59, Difference: 6.15
Baseline precision: 0.88, New precision: 0.92, Difference: 0.04
Baseline recall: 0.85, New recall: 0.92, Difference: 0.06
Baseline f1: 0.85, New f1: 0.92, Difference: 0.06
```

[101]:
```
y_true = y_valid.tolist()  # Convert labels to a list
preds = model_GRU.predict(X_valid)
y_probs = preds.squeeze().tolist()  # Store the prediction probabilities as a␣
↪list
y_preds = tf.round(y_probs).numpy().tolist()  # Convert probabilities to class␣
↪predictions and convert to a list
```

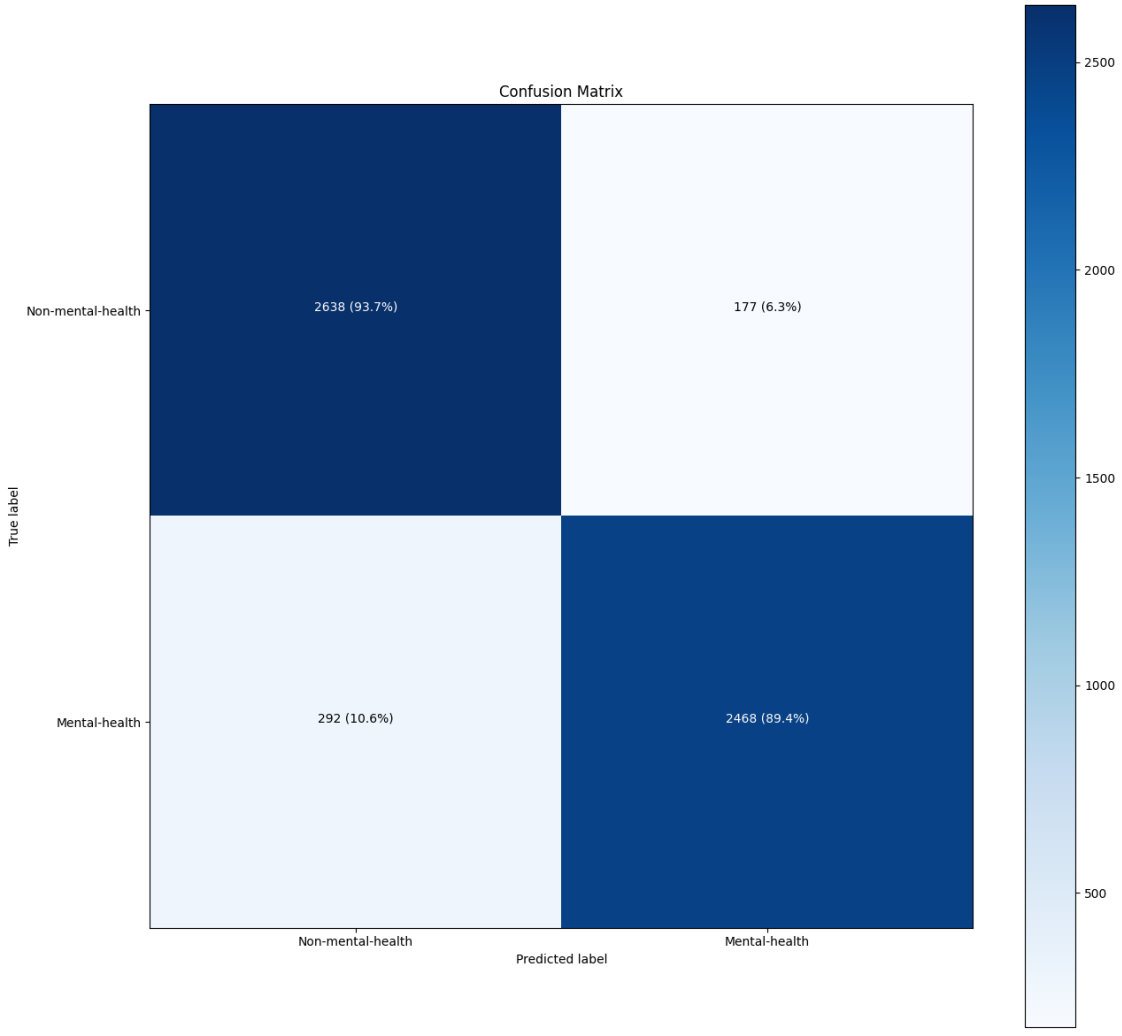
```
175/175             2s 9ms/step
```

A confusion matrix is created for visualization of the model's classification performance. The matrix readability is enhanced via a custom function.

[102]:
```
# Check out the non-prettified confusion matrix
from sklearn.metrics import confusion_matrix
confusion_matrix(y_true=y_true,
                 y_pred=y_preds)
```

[102]:
```
array([[2638,  177],
       [ 292, 2468]])
```

[103]:
```
# Make a prettier confusion matrix
make_confusion_matrix(y_true=y_true,
                      y_pred=y_preds,
                      classes=class_names,
                      figsize=(15, 15),
                      text_size=10)
```



Lastly, the 'random_predictions' function generates and displays predictions on random samples.

[104]:
```
random_predictions(model_GRU,
                   X_valid,
                   y_valid,
                   num_samples=20,
                   class_names=class_names)
```

```
1/1             0s 35ms/step

Sample 1:
Text: financial crisis evicted house lost lighti feel like everythings gone shit
fam financial problem year roll one day evicted cant shake feeling restlessness
since need take care mum time back lost girl bcs got college like reason think
shot spent le time going back almost always weekend whenever free time serious
since family knew abt visited others place holiday everything went shit now miss
lot year still feel yesterday tried go feel burnt seems happy friend lost world
friend mine mind keep telling long happy cant foolmyself longer peace home study
always feel like somethings going cant sleep want thing like go home see smile
True: Mental-health

 Predicted: Mental-health

Sample 2:
Text: anybody first generation iphone se black reddit icon please hit youre
going update please let know first really want black icon back
True: Non-mental-health

 Predicted: Non-mental-health

Sample 3:
Text: biting bulletim year old drop working software developer since ive spend
time learning trying improve everything around seems falling apart ive always
bullied downer seems life cycle unhappiness time checkout know hurt family shock
people around feel like ive wasted everyones timeenergy enough already yes
therapy made feel like shit guess rant ive mindset many time feel much getting
better point seems impossible
True: Mental-health

 Predicted: Mental-health

Sample 4:
Text: im updating eso taking everrrrrrrr
True: Non-mental-health

 Predicted: Non-mental-health

Sample 5:
Text: thinking killing last daysi thought recently getting argument girlfriend
parent secret relationship told beginning good idea talk parent disagreed dumb
kept secret relationship secret part love girlfriend mom took phone read message
```

sending last day girlfriend said deleting everything found lie parent said stay away contact im home think hanging ive tried hard last year feel like slap face im tried life crapping matter hard try seem good enough world idk im lose feel like option end point
True: Mental-health

Predicted: Mental-health

Sample 6:
Text: feeling worthless take year figure im addictim doctor prescribed benzos benzos since im guess im depth withdrawal keep reliving whole life new pyschiatrist cussed friday month ago messed rx reduced w refill set telehealth appt march th informed said get rx fix march th till hell friday spoke w nurse told change called said im making look crazy wanted go script change person sorry im going suffer long guess thought pleasantly suffer silence asked cold heartedly want prescribe valium liking scolded calling cv make sure dont pick refill told wanted prescribed year want billed two telehealth appts hung phone screaming bye moved hour away leaf office pick anyways guess im addict keep hating im anything right addicted benzos never anything hard dealt w ignorant doc past pull shit take business street isnt prescribed benzos doesnt know somebody know somebody get benzos im broke getting older dont want illegal shit whole shit questioning relationship failure like damn prescribed wrong drug chose heroin younger least help benzo addict get fucked system making u look pyscho doc make mistake arent feeling good even mention risk getting pegged addict losing rx sanity im really hating hour ride tomorrow make thought dissipate feeling hopeless anxiety anxiety thanks
True: Mental-health

Predicted: Mental-health

Sample 7:
Text: feel love someone online fuck dont qant shit really dont stress please help
True: Non-mental-health

Predicted: Non-mental-health

Sample 8:
Text: even knowive made set condition mind going kill believe fine may snap break condition comfort worry one know want anybody know exist society like possible
True: Mental-health

Predicted: Mental-health

Sample 9:
Text: think going kill year old male know father left year old negligent disconnected alcoholic kid mom used abuse relentlessly beat belt extension chord punch face tell im stupid burned cigarette time taken year pay student loan debt

degree completely useless live poverty hurt knee month ago cant even manual labour job used get everything seems completely hopeless clue totally atomised family support live pay check pay check cause living high even though work hour week barely scraping whenever start get ahead something bad happens month ago shell dollar get car fixed needed new control arm plus new tire plus alignment tow need new muffler pas emission test can not legally drive work knee get better grinding like mad man shelling physio therapy get knee fixed costing ton ill laid bed month able work able pay rent bank account need last rest month life exhausting work hard can never pull pit im sad anything hopeless depressed stuck escape
True: Mental-health

Predicted: Mental-health

Sample 10:
Text: stupid state taking long finish covid app month since latest news app tested new info since since live red prefer call blue state people wont take covid seriously fuck kemp
True: Non-mental-health

Predicted: Non-mental-health

Sample 11:
Text: attracted celebrity crige attracted anime girl scientifically attracted celebrity peson attracted simps bother never reach somehow possible one trillion chance crush na celebrity seeing canvas art flawed person might completely fake anime girl exist simps bother never reach everyone know thats truth crush character literally see whole character show literally get know literally flawless
True: Non-mental-health

Predicted: Non-mental-health

Sample 12:
Text: sure might jump bridge onto highway might jump work want fucking live anymore wanted die fucking long feel like thing ever really wanted probably going actually bitch hopefully find courage finally kill really reason anything life basically perfect always wanted kill always will might well get ruin life life around hopefully killing today
True: Mental-health

Predicted: Mental-health

Sample 13:
Text: need brace went dentist earlier teeth cleaned checked didnt concern although chatting amongst dentist assistant apprentice think made wonder going need retainer overbite underbite no know crooked teeth didnt really care turn teeth blocking teeth coming need teeth pulled brace brace literally last thing wanted ive heard hurt lot need avoid certain food dont want dont know cant eat

drink able satiate orange juice addiction able eat dad butter chicken beef broccoli even survive pain
True: Non-mental-health

Predicted: Non-mental-health

Sample 14:
Text: wont post thr black chapter wont post im thinking whats gon na happen next chapter idea ill post
True: Non-mental-health

Predicted: Non-mental-health

Sample 15:
Text: wtf kissing booth coming next year like bruhhhhh second one came
True: Non-mental-health

Predicted: Non-mental-health

Sample 16:
Text: im year old ambition want get far life thought killing haunting long time want end itthis going long one type read long piece text think move elsewhere im year old male live uk im aspiring filmmaker im currently university filmmaking course young age wideeyed kid saw world playground imagination run wild place become almost anything time life discovered many thing created many goal aspire towards everyday look important relic something thrive towards time childhood fun full optimism carried forward early year teenager friend school moved first sense loss next grandmother grandfather passing away within year carried forward time high school optimism still alive far remember high school fun time dispute overly shy able gain good group friend dream ambition seemed closer ever began watch film range arthouse cinema built love writing able come story im still working even day shy think remembered everyone latter half final year got stage sang song everyone loved remember boy came shell sang heart attended college soon finally wanted held back little poor math grade level one art design medium course first moving towards level film course year went began lose confidence disregarding script calling awful even deleting computer attending class began lose hope able get highest grade class got university made happy spark gone know ive enjoyed university far ive working likeminded people reason feel like lesser person latter half first year considering redflag stopped shortly attended music festival ive attending festival since last year helped forgive see thousand people time life made think lose kill want got back work thought killing stuck ive degrading ive person parent worried brother worried really affected filmmaking well think future hold haunting long time wish end life right now write im edge hope itbut urge overwhelming actually read want thank taking time wonderful life hope everything work certainly ive never love sense success really see point continuing life take life post want reflect life make wonderful want see flaw hope stronger point overcome tough time live see happiness waiting thank reading wish happy life know mine worth life worth everything

```
True: Mental-health

 Predicted: Mental-health

Sample 17:
Text: uninteresting title text post not optional
True: Non-mental-health

 Predicted: Non-mental-health

Sample 18:
Text: yever feel like cant anything lost motivation let vent sec school suck
hate good part friend like two fun teacher online fcked doggy style im missing
like thirty assignment motivation went im failing like three class matter many
time turn computer go canvas work brain shuts laziness least thats like tell
last ive felt super empty bunch stuff used love playing guitar try ill get one
two practice month loved making art everytime open program lose spark feel tired
time motivation anything lying bed listening music hour hard take care hard eat
hard clean room hard get bed im basically walking corporal husk yeah im thriving
please help whats wrong lol
True: Non-mental-health

 Predicted: Non-mental-health

Sample 19:
Text: someone someone school replaced bathroom qr code qr code leading never gon
na give rick astley
True: Non-mental-health

 Predicted: Non-mental-health

Sample 20:
Text: dude sure personality anymore like always liked male thing think seems
like always wanted girl like little kid yo always dreamt girl turning girl even
know trans people know dude sure gender right know looked like girl outfit
spesific item look cool wan na cant sometimes maybe sound strange sometimes
sometimes iwhen home alone pick mother item underwear stuff wear use mi weird
know
True: Non-mental-health

 Predicted: Non-mental-health
```

4.1 Comparing all models

[105]:
```
# Combine model results into a DataFrame
all_model_results = pd.DataFrame({"baseline": baseline_results,
                                  "LSTM": lstm_model_results,
                                  "GRU": model_GRU_results,
                                  })
```

```
all_model_results = all_model_results.transpose()
all_model_results["accuracy"] = all_model_results["accuracy"]/100
all_model_results
```

[105]:

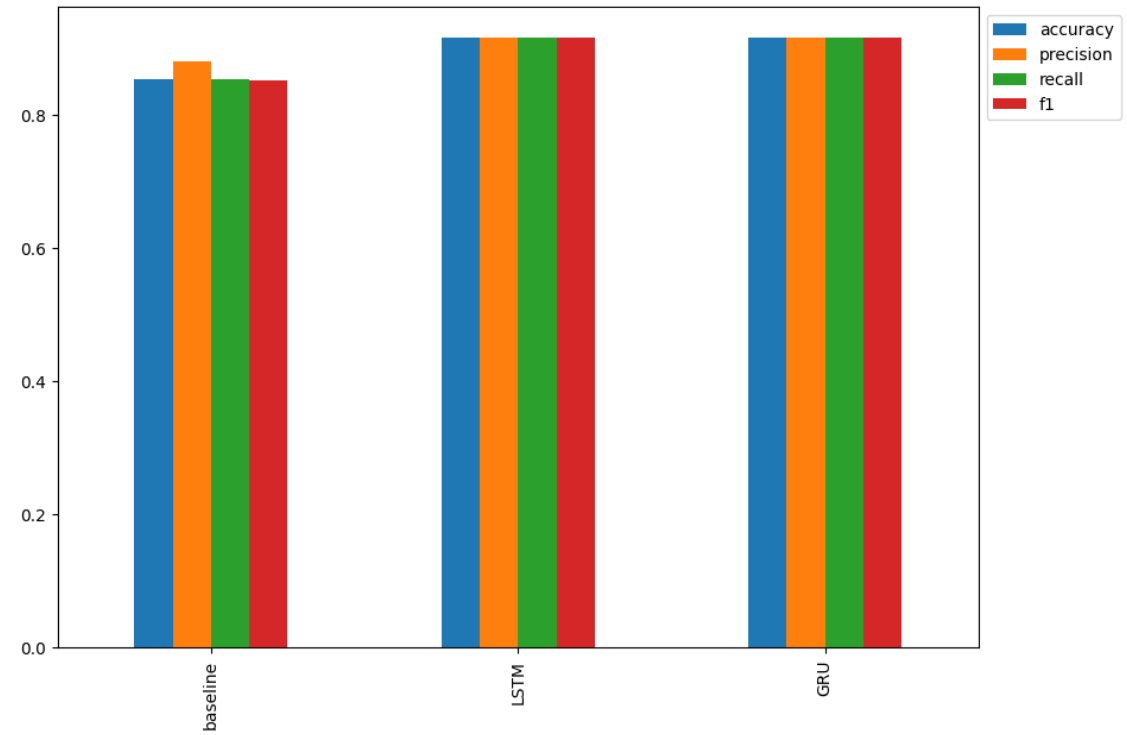
	accuracy	precision	recall	f1
baseline	0.854350	0.881010	0.854350	0.851979
LSTM	0.915516	0.915541	0.915516	0.915518
GRU	0.915874	0.916550	0.915874	0.915821

[106]:
```
all_model_results_sorted = all_model_results.sort_values("f1", ascending=False)
all_model_results_sorted
```

[106]:

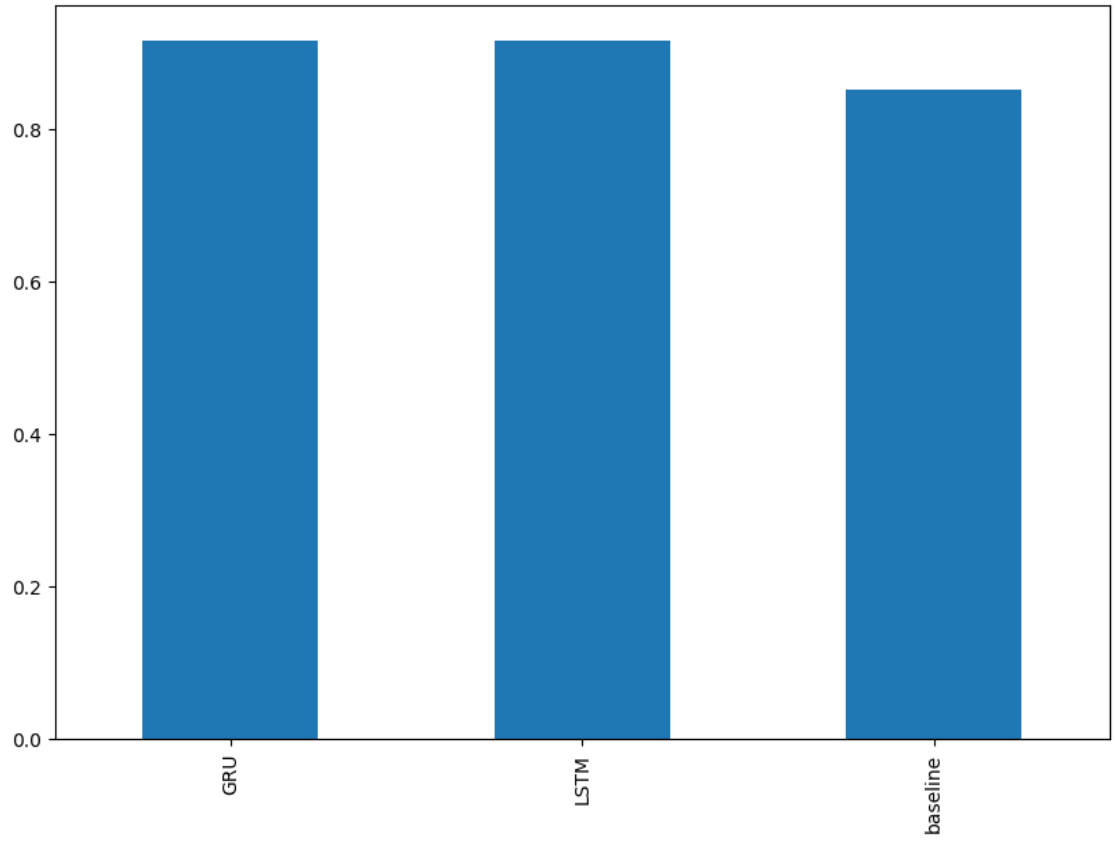
	accuracy	precision	recall	f1
GRU	0.915874	0.916550	0.915874	0.915821
LSTM	0.915516	0.915541	0.915516	0.915518
baseline	0.854350	0.881010	0.854350	0.851979

[107]:
```
# Plot and compare all of the model results
all_model_results.plot(kind="bar", figsize=(10, 7)).legend(bbox_to_anchor=(1.0,␣
↪1.0));
```



[108]:
```
# Sort model results by f1-score
```

```
all_model_results.sort_values("f1", ascending=False)["f1"].plot(kind="bar",␣
↪figsize=(10, 7));
```



Evaluation Metrics

Moving forward, we will deploy the Model GRU for further analysis.

[109]:
```
y_true = y_valid.tolist()  # Convert labels to a list
preds = model_GRU.predict(X_valid)
y_probs = preds.squeeze().tolist()  # Store the prediction probabilities as a␣
↪list
y_preds = tf.round(y_probs).numpy().tolist()  # Convert probabilities to class␣
↪predictions and convert to a list
```

```
175/175             2s 9ms/step
```

[110]:
```
from sklearn.metrics import classification_report, accuracy_score, f1_score,␣
↪recall_score, precision_score

report = classification_report(y_true, y_preds)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.90      0.94      0.92      2815
           1       0.93      0.89      0.91      2760

    accuracy                           0.92      5575
   macro avg       0.92      0.92      0.92      5575
weighted avg       0.92      0.92      0.92      5575
```

5 Ensemble Models

[138]:
```
import numpy as np
import tensorflow as tf

# Get prediction probabilities
# Baseline model probabilities for the positive class
baseline_pred_probs = baseline_model.predict_proba(X_valid)[:, 1]

# LSTM and GRU model probabilities
lstm_pred_probs = lstm_model.predict(X_valid).flatten()
gru_pred_probs = model_GRU.predict(X_valid).flatten()

# Average the prediction probabilities
combined_pred_probs = (baseline_pred_probs + lstm_pred_probs + gru_pred_probs) /
 ↪ 3

# Convert averaged probabilities to binary predictions
combined_preds = tf.round(combined_pred_probs)

# Calculate ensemble results
ensemble_results = calculate_results(y_valid, combined_preds.numpy())
print("Ensemble Results:", ensemble_results)

# Add ensemble results to DataFrame for comparison
all_model_results.loc['Ensemble'] = {
    'accuracy': ensemble_results['accuracy'] / 100,  # Convert percentage if␣
 ↪necessary
    'precision': ensemble_results['precision'],
    'recall': ensemble_results['recall'],
    'f1': ensemble_results['f1']
}
all_model_results
```

```
175/175             2s 9ms/step
175/175             2s 9ms/step
Ensemble Results: {'accuracy': 92.46636771300449, 'precision':
```

```
0.9250171000796918, 'recall': 0.9246636771300448, 'f1': 0.9246593285796662}
```

[138]:

	accuracy	precision	recall	f1
baseline	0.854350	0.881010	0.854350	0.851979
LSTM	0.915516	0.915541	0.915516	0.915518
GRU	0.915874	0.916550	0.915874	0.915821
Ensemble	0.924664	0.925017	0.924664	0.924659

[ ]: