Social media depression detection

Submitted By :
160121733162 V.Srinidhi
160121733163 Y.Manasvi
160121733164 A.Meghanath

Introduction

Depression (major depressive disorder) is a common and serious medical illness that negatively affects how you feel, the way you think and how you act. Fortunately, it is also treatable. Depression causes feelings of sadness and/or a loss of interest in activities you once enjoyed. It can lead to a variety of emotional and physical problems and can decrease your ability to function at work and at home.

Domain Introduction: How Social Media causes depression

Here are some ways in which social media may affect depression:

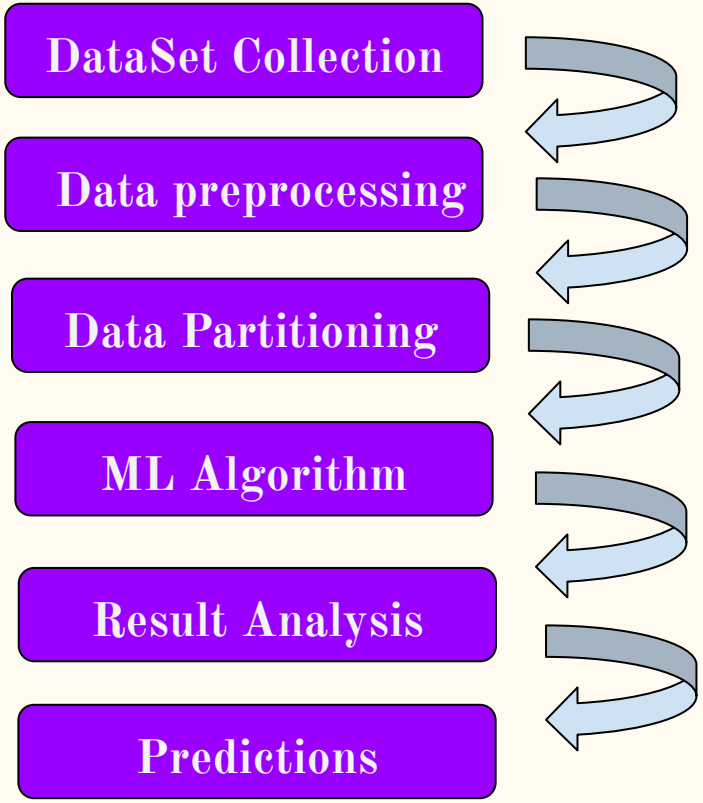
1. Social Comparison
2. Cyberbullying
3. Fear of Missing Out (FOMO)
4. Addiction and Time Wasting
5. Negative Content Exposure
6. Sleep Disruption
7. Social Isolation

Problem Introduction

Problem statement:

This project aims to develop a Natural Language Processing model for detecting indicators of depression in social media content. By analyzing text data from various social media platforms, the model will identify linguistic patterns, sentiment expressions, and contextual cues associated with depressive symptoms. The main aim is to make a system that can alert when someone might need help, so we can reach out to them.

WorkFlow Diagram



Python Libraries Used

- Pandas
- NLP
- NLTK
- Multinomial Naive Bayes
- Gaussian Naive Bayes
- Numpy

Depression detection has several applications across various fields:

1. **Healthcare**: Identifying individuals at risk for depression allows for early intervention and personalized treatment plans.
2. **Mental Health Services**: Improving access to mental health support by automating screening and monitoring processes.
3. **Research**: Studying patterns and risk factors associated with depression for better understanding and development of effective interventions.
4. **Employment**: Supporting employee well-being by detecting signs of depression and providing appropriate resources.
5. **Education**: Identifying students who may be struggling with depression and offering targeted support within educational settings.
6. **Technology**: Integrating depression detection into digital platforms, such as social media or wearable devices, for continuous monitoring and intervention.

Thank You

draft

May 16, 2024

0.1 NLP Assignment 2

Done by

160121733162

160121733163

160121733164

0.2 *Problem Statement*

This project aims to develop a Natural Language Processing model for detecting indicators of depression in social media content. By analyzing text data from various social media platforms, the model will identify linguistic patterns, sentiment expressions, and contextual cues associated with depressive symptoms. The main aim is to make a system that can alert when someone might need help, so we can reach out to them.

[ ]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
import seaborn as sns
import string
import nltk
import re
```

[ ]:
```
dataset_columns = ["target", "ids", "date", "flag", "user", "text"]
dataset_encode = "ISO-8859-1"
df=pd.read_csv("/content/training.1600000.processed.noemoticon (1).csv",␣
 ↪encoding = dataset_encode,names=dataset_columns)
```

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

[ ]:
```
   target         ids                          date      flag  \
0       0  1467810369  Mon Apr 06 22:19:45 PDT 2009  NO_QUERY
1       0  1467810672  Mon Apr 06 22:19:49 PDT 2009  NO_QUERY
```

```
2       0  1467810917  Mon Apr 06 22:19:53 PDT 2009  NO_QUERY
3       0  1467811184  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY
4       0  1467811193  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY

              user                                                text
0  _TheSpecialOne_  @switchfoot http://twitpic.com/2y1zl - Awww, t…
1    scotthamilton  is upset that he can't update his Facebook by …
2         mattycus  @Kenichan I dived many times for the ball. Man…
3          ElleCTF    my whole body feels itchy and like its on fire
4           Karoli  @nationwideclass no, it's not behaving at all…
```

[ ]:
```
df.drop(['ids','date','flag','user'],axis = 1,inplace = True)
#axis=1 means columns deleted
#inplace true means data gets modified and copied to dataframe itself.
```

[ ]:
```
df['target'].value_counts()
```

[ ]:
```
target
4    10001
0     9999
Name: count, dtype: int64
```

[ ]:
```
#remove punctuation
def remove_punctuation(text):
    no_punct=[words for words in text if words not in string.punctuation]
    words_wo_punct=''.join(no_punct)
    return words_wo_punct
df['clean_text']=df['text'].apply(lambda x: remove_punctuation(x))
df.head()
```

[ ]:
```
   target                                               text  \
0       0  @switchfoot http://twitpic.com/2y1zl - Awww, t…
1       0  is upset that he can't update his Facebook by …
2       0  @Kenichan I dived many times for the ball. Man…
3       0    my whole body feels itchy and like its on fire
4       0  @nationwideclass no, it's not behaving at all…

                                      clean_text
0  switchfoot httptwitpiccom2y1zl  Awww thats a b…
1  is upset that he cant update his Facebook by t…
2  Kenichan I dived many times for the ball Manag…
3    my whole body feels itchy and like its on fire
4  nationwideclass no its not behaving at all im …
```

[ ]:
```
#tokenization
nltk.download('punkt')
def tokenize(text):
```

```
    split=re.split("\W+",text)
    return split
df['clean_text_tokenize']=df['clean_text'].apply(lambda x: tokenize(x.lower()))
```

```
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

[ ]:
```
df['clean_text_tokenize']
```

[ ]:
```
0        [switchfoot, httptwitpiccom2y1zl, awww, thats,…
1        [is, upset, that, he, cant, update, his, faceb…
2        [kenichan, i, dived, many, times, for, the, ba…
3        [my, whole, body, feels, itchy, and, like, its…
4        [nationwideclass, no, its, not, behaving, at, …
                              …
19995    [just, woke, up, having, no, school, is, the, …
19996    [thewdbcom, very, cool, to, hear, old, walt, i…
19997    [are, you, ready, for, your, mojo, makeover, a…
19998    [happy, 38th, birthday, to, my, boo, of, alll,…
19999    [happy, charitytuesday, thenspcc, sparkscharit…
Name: clean_text_tokenize, Length: 20000, dtype: object
```

[ ]:
```
import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
```

[ ]:
```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
```

```
'yours',
'yourself',
'yourselves',
'he',
'him',
'his',
'himself',
'she',
"she's",
'her',
'hers',
'herself',
'it',
"it's",
'its',
'itself',
'they',
'them',
'their',
'theirs',
'themselves',
'what',
'which',
'who',
'whom',
'this',
'that',
"that'll",
'these',
'those',
'am',
'is',
'are',
'was',
'were',
'be',
'been',
'being',
'have',
'has',
'had',
'having',
'do',
'does',
'did',
'doing',
'a',
```

```
'an',
'the',
'and',
'but',
'if',
'or',
'because',
'as',
'until',
'while',
'of',
'at',
'by',
'for',
'with',
'about',
'against',
'between',
'into',
'through',
'during',
'before',
'after',
'above',
'below',
'to',
'from',
'up',
'down',
'in',
'out',
'on',
'off',
'over',
'under',
'again',
'further',
'then',
'once',
'here',
'there',
'when',
'where',
'why',
'how',
'all',
'any',
```

```
'both',
'each',
'few',
'more',
'most',
'other',
'some',
'such',
'no',
'nor',
'not',
'only',
'own',
'same',
'so',
'than',
'too',
'very',
's',
't',
'can',
'will',
'just',
'don',
"don't",
'should',
"should've",
'now',
'd',
'll',
'm',
'o',
're',
've',
'y',
'ain',
'aren',
"aren't",
'couldn',
"couldn't",
'didn',
"didn't",
'doesn',
"doesn't",
'hadn',
"hadn't",
'hasn',
```

```
 "hasn't",
 'haven',
 "haven't",
 'isn',
 "isn't",
 'ma',
 'mightn',
 "mightn't",
 'mustn',
 "mustn't",
 'needn',
 "needn't",
 'shan',
 "shan't",
 'shouldn',
 "shouldn't",
 'wasn',
 "wasn't",
 'weren',
 "weren't",
 'won',
 "won't",
 'wouldn',
 "wouldn't"]
```

[ ]:
```
#in this code stopwords are removed
stopword = nltk.corpus.stopwords.words('english')
def remove_stopwords(text):
    text=[word for word in text if word not in stopword]
    return text
df['clean_text_tokenize_stopwords'] = df['clean_text_tokenize'].apply(lambda x:␣
 ↪remove_stopwords(x))
df.head(10)
```

[ ]:
```
   target                                                text  \
0       0  @switchfoot http://twitpic.com/2y1zl - Awww, t…
1       0  is upset that he can't update his Facebook by …
2       0  @Kenichan I dived many times for the ball. Man…
3       0    my whole body feels itchy and like its on fire
4       0  @nationwideclass no, it's not behaving at all…
5       0                      @Kwesidei not the whole crew
6       0                                       Need a hug
7       0  @LOLTrish hey  long time no see! Yes.. Rains a…
8       0               @Tatiana_K nope they didn't have it
9       0                           @twittera que me muera ?

                                        clean_text  \
```

```
0  switchfoot httptwitpiccom2y1zl  Awww thats a b…
1  is upset that he cant update his Facebook by t…
2  Kenichan I dived many times for the ball Manag…
3    my whole body feels itchy and like its on fire
4  nationwideclass no its not behaving at all im …
5                      Kwesidei not the whole crew
6                                       Need a hug
7  LOLTrish hey  long time no see Yes Rains a bit…
8                 TatianaK nope they didnt have it
9                           twittera que me muera

                                clean_text_tokenize  \
0  [switchfoot, httptwitpiccom2y1zl, awww, thats,…
1  [is, upset, that, he, cant, update, his, faceb…
2  [kenichan, i, dived, many, times, for, the, ba…
3  [my, whole, body, feels, itchy, and, like, its…
4  [nationwideclass, no, its, not, behaving, at, …
5               [kwesidei, not, the, whole, crew, ]
6                                  [need, a, hug, ]
7  [loltrish, hey, long, time, no, see, yes, rain…
8         [tatianak, nope, they, didnt, have, it, ]
9                       [twittera, que, me, muera, ]

                       clean_text_tokenize_stopwords
0  [switchfoot, httptwitpiccom2y1zl, awww, thats,…
1  [upset, cant, update, facebook, texting, might…
2  [kenichan, dived, many, times, ball, managed, …
3         [whole, body, feels, itchy, like, fire, ]
4  [nationwideclass, behaving, im, mad, cant, see, ]
5                         [kwesidei, whole, crew, ]
6                                      [need, hug, ]
7  [loltrish, hey, long, time, see, yes, rains, b…
8                         [tatianak, nope, didnt, ]
9                          [twittera, que, muera, ]
```

[ ]:
```
new_df = pd.DataFrame()
new_df['text'] = df['clean_text']
new_df['label'] = df['target']
new_df['label'] = new_df['label'].replace(4,1)
```

[ ]:
```
print(new_df.head())
print('Label: \n', new_df['label'].value_counts())
```

```
                                                text  label
0  switchfoot httptwitpiccom2y1zl  Awww thats a b…      0
1  is upset that he cant update his Facebook by t…      0
2  Kenichan I dived many times for the ball Manag…      0
```

```
3    my whole body feels itchy and like its on fire       0
4  nationwideclass no its not behaving at all im …      0
Label:
 label
1    10001
0     9999
Name: count, dtype: int64
```

[ ]:
```
from sklearn.model_selection import train_test_split
X = new_df['text']
y = new_df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

print(X_train.shape,X_test.shape,y_train.shape,y_test.shape)
```

```
(16000,) (4000,) (16000,) (4000,)
```

[ ]:
```
y_train.value_counts()
```

[ ]:
```
label
1    8019
0    7981
Name: count, dtype: int64
```

0.3 *LogisticRegression*

[ ]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Create a TF-IDF vectorizer to convert text data into numerical features
vectorizer = TfidfVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Initialize and train the logistic regression classifier
classifier = LogisticRegression()
classifier.fit(X_train_vectorized, y_train)

# Predict labels for the test set
predictions = classifier.predict(X_test_vectorized)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.76975
```

0.4 *Bernoulli Naive Bayes*

[ ]:
```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import BernoulliNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

[ ]:
```
# Create a CountVectorizer to convert text data into binary features
vectorizer = CountVectorizer(binary=True)
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Initialize and train the Bernoulli Naive Bayes classifier
classifier = BernoulliNB()
classifier.fit(X_train_vectorized, y_train)

# Predict labels for the test set
predictions = classifier.predict(X_test_vectorized)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.774
```

0.5 *MultinomialNB*

[ ]:
```
model = make_pipeline(TfidfVectorizer(), MultinomialNB())
model.fit(X_train,y_train)
validation1 = model.predict(X_train)
accuracy_score(y_train, validation1)
# MultinomialNB: Typically used for text classification tasks where features␣
 ↪represent word counts or TF-IDF values.
```

[ ]:
```
0.913375
```

0.6 *Evalution Metrics*

[ ]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score, roc_curve, auc, confusion_matrix
import matplotlib.pyplot as plt

# Example true labels and predicted labels
true_labels = y_train
predicted_labels = validation1
```

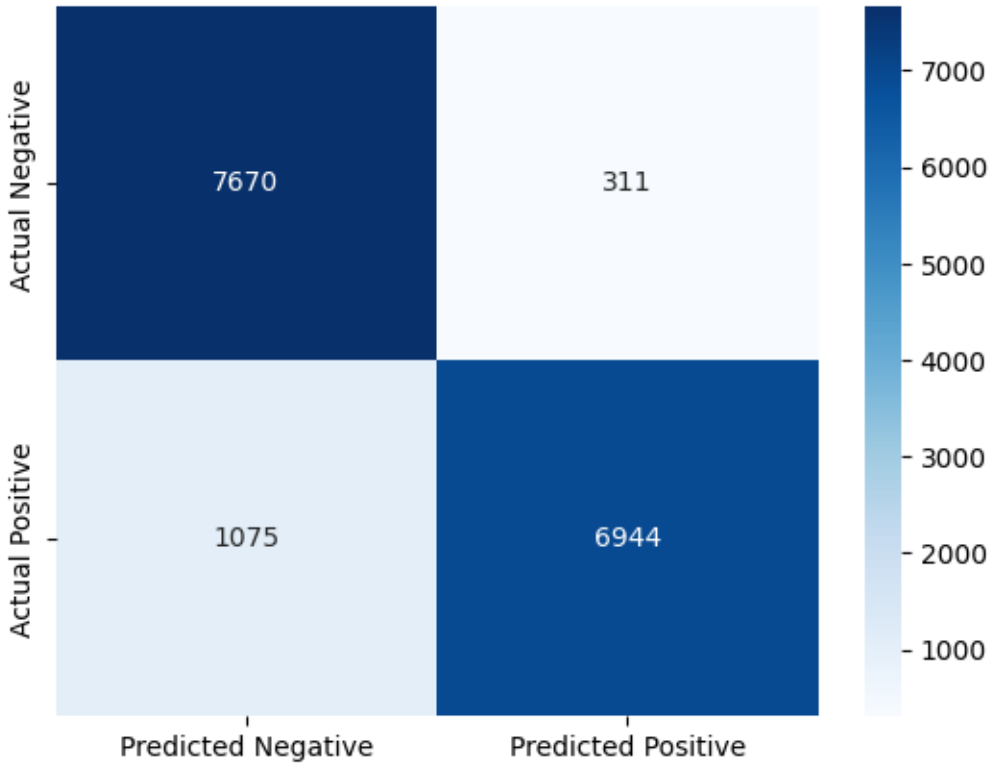
```
# Accuracy
accuracy = accuracy_score(true_labels, predicted_labels)
print("Accuracy:", accuracy)
# Precision
precision = precision_score(true_labels, predicted_labels)
print("Precision:", precision)
# Recall
recall = recall_score(true_labels, predicted_labels)
print("Recall:", recall)
# F1 Score
f1 = f1_score(true_labels, predicted_labels)
print("F1 Score:", f1)

# Confusion Matrix
conf_matrix = confusion_matrix(true_labels, predicted_labels)
print("Confusion Matrix:")
print(conf_matrix)
```

```
Accuracy: 0.913375
Precision: 0.9571330117160579
Recall: 0.865943384461903
F1 Score: 0.9092575618698442
Confusion Matrix:
[[7670  311]
 [1075 6944]]
```

[ ]:
```
import seaborn as sns
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',␣
 ↪xticklabels=['Predicted Negative', 'Predicted Positive'],␣
 ↪yticklabels=['Actual Negative', 'Actual Positive'])
```

[ ]:
```
<Axes: >
```



0.7 *Prediction*

[34]:
```
def predict_category(s, model=model):
    pred = model.predict([s])
    if(pred[0]==0):
        return "Depressed"
    else:
        return "Not Depressed"
```

[35]:
```
predict_category("i wanna shot myself")
```

[35]: 'Depressed'

[36]:
```
predict_category("i will Kill you")
```

[36]: 'Depressed'

[37]:
```
predict_category("i love my mom")
```

[37]: 'Not Depressed'

[38]:
```
predict_category("I'm cute")
```

[38]: 'Not Depressed'

[41]:
```
predict_category("I hate my life")
```

[41]: 'Depressed'

[42]:
```
predict_category("I'm excited")
```

[42]: 'Not Depressed'

[43]:
```
predict_category("sad rightnow")
```

[43]: 'Depressed'

[44]:
```
predict_category("How are you")
```

[44]: 'Not Depressed'

[45]:
```
predict_category("I am fine")
```

[45]: 'Not Depressed'

[46]:
```
predict_category("alone and helpless")
```

[46]: 'Depressed'

0.8 *Depression detection has several applications across various fields:*

1. **Healthcare:**Identifying individuals at risk for depression allows for early intervention and personalized treatment plans.
2. **Mental Health Services:**Improving access to mental health support by automating screening and monitoring processes.
3. **Research:** Studying patterns and risk factors associated with depression for better understanding and development of effective interventions.
4. **Employment:** Supporting employee well-being by detecting signs of depression and providing appropriate resources.
5. **Education:** Identifying students who may be struggling with depression and offering targeted support within educational settings.
6. **Technology:** Integrating depression detection into digital platforms, such as social media or wearable devices, for continuous monitoring and intervention.