```
Import pandas as pd

data=pd.read_csv("https://raw.githubusercontent.com/Ankit152/IMDB-sentimen
t-analysis/master/IMDB-Dataset.csv")
```

Run this command on your terminal

pip install ipykernel, pandas

```
import string
```

```
exclude=string.punctuation
```

```
def remove_punc(text):
    for char in exclude:
        text=text.replace(char,"")
    return text
```

```
text2="my name is s@nny sa##vita!!!!!!!!!"
```

```
remove_punc(text2)
```

```
def remove_punc1(text):
    return text.translate(str.maketrans("","",exclude))
```

```
data["review"]=data["review"].apply(remove_punc)
```

https://github.com/rishabhverma17/sms_slang_translator/blob/master/slang.txt

```
text="FYI this is not true"
text2="LAMO the class was so funny"
text3="i want report ASAP"
```

```
chat_words={
   " AFAIK":"As Far As I Know",
```

```
"AFK": "Away From Keyboard",
"ASAP":"As Soon As Possible",
"BTW":"By The Way",
"B4":"Before",
"LAMO":"Laugh My A.. Off",
"FYI":"For your information"
}
```

```
def chat_conversion(text):
    new_text=[]
    for w in text.split():
        if w.upper() in chat_words:
            new_text.append(chat_words[w.upper()])
        else:
            new_text.append(w)
    return " ".join(new_text)
```

Pip install textblob

Import textblob

```
txtblob=TextBlob(text)
```

```
txtblob.correct().string
```

Pip install nltk

```
from nltk.corpus import stopwords
```

```
nltk.download('stopwords')
```

```
stopwords.words("english")
```

```
def remove_stopwords(text):
    new_text=[]
```

```
    for word in text.split():
        if word in stopwords.words("english"):
            new_text.append("")
        else:
            new_text.append(word.strip())


    return " ".join(new_text).replace("    ","")
```

Pip install emoji

https://pypi.org/project/emoji/

```
def remove_emoji(text):
    clean_text=emoji.demojize(text)
    return clean_text
```

```
remove_emoji(original_text)
```

```
import re
def remove_emojis_manually(text):
    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"  # emoticons
                               u"\U0001F300-\U0001F5FF"  # symbols &
pictographs
                               u"\U0001F680-\U0001F6FF"  # transport & map
symbols
                               u"\U0001F700-\U0001F77F"  # alchemical
symbols
                               u"\U0001F780-\U0001F7FF"  # Geometric
Shapes Extended
                               u"\U0001F800-\U0001F8FF"  # Supplemental
Arrows-C
                               u"\U0001F900-\U0001F9FF"  # Supplemental
Symbols and Pictographs
                               u"\U0001FA00-\U0001FA6F"  # Chess Symbols
                               u"\U0001FA70-\U0001FAFF"  # Symbols and
Pictographs Extended-A
                               u"\U00002702-\U000027B0"  # Dingbats
                               u"\U000024C2-\U0001F251"
```

```
                                   "]+", flags=re.UNICODE)
    clean_text = emoji_pattern.sub(r'', text)
    return clean_text
```

pip install spacy
python -m spacy download en_core_web_sm

pip install ntlk

nltk.download(“alll”)

Assignment⇒https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter

```
from nltk.stem import WordNetLemmatizer

def lammatization(text):
    words=text.split()

    lemmetizer=WordNetLemmatizer()

    lemetized_word=[lemmetizer.lemmatize(word) for word in words]

    return lemetized_word
```