```
pip install emoji
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting emoji
  Downloading emoji-2.0.0.tar.gz (197 kB)
     |████████████████████████████████| 197 kB 5.2 MB/s
Building wheels for collected packages: emoji
  Building wheel for emoji (setup.py) ... done
  Created wheel for emoji: filename=emoji-2.0.0-py3-none-any.whl size=193022 sha256=684a94ac1d11f24f4b2423690ee3aa75fc
  Stored in directory: /root/.cache/pip/wheels/ec/29/4d/3cfe7452ac7d8d83b1930f8a6205c3c9649b24e80f9029fc38
Successfully built emoji
Installing collected packages: emoji
Successfully installed emoji-2.0.0
```

```
pip install contractions
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting contractions
  Downloading contractions-0.1.72-py2.py3-none-any.whl (8.3 kB)
Collecting textsearch>=0.0.21
  Downloading textsearch-0.0.21-py2.py3-none-any.whl (7.5 kB)
Collecting anyascii
  Downloading anyascii-0.3.1-py3-none-any.whl (287 kB)
     |████████████████████████████████| 287 kB 5.0 MB/s
Collecting pyahocorasick
  Downloading pyahocorasick-1.4.4-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (106 kB)
     |████████████████████████████████| 106 kB 43.2 MB/s
Installing collected packages: pyahocorasick, anyascii, textsearch, contractions
Successfully installed anyascii-0.3.1 contractions-0.1.72 pyahocorasick-1.4.4 textsearch-0.0.21
```

```
import pandas as pd
import numpy as np
import emoji
import contractions
import re
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
```

```
%matplotlib inline
import nltk

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
nltk.download('wordnet')

from nltk.stem import LancasterStemmer, WordNetLemmatizer
from nltk import pos_tag
from nltk.corpus import wordnet
nltk.download('sentiwordnet')
from sklearn import svm
from sklearn.svm import SVC
from nltk.corpus import sentiwordnet as swn
import pickle

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import KFold, StratifiedKFold, cross_val_score
from sklearn.metrics import classification_report,confusion_matrix, accuracy_score
```



```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package sentiwordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/sentiwordnet.zip.
```

Import Library

```
df = pd.read_csv('ebay_reviews.csv')
df
```


	category	review title	review content	rating
			This gaming headset ticks all the boxes # look...	5
1	Headsets	Good for those with a big head, low budget	Easy setup, rated for 6 hours battery but mine...	3
2	Headsets	MezumiWireless Gaming Headset	I originally bought this wireless headset for ...	5
3	Headsets	HW- S2 great headset.	This is my 2nd Mezumi headset, It kills the fi...	5
4	Headsets	BEST HEADPHONES I'VE PURCHASED IN MY ENTIRE LIFE	This is probably the best headset I've purchas...	5
...	...	...	...	...
44751	Racks & Holders	Utensil holder	Reasonably priced but a little flimsy	3
44752	Racks & Holders	Recommended	As described	5
44753	Racks &	cheap looking	cheap looking	1

```
df.iloc[40:50]
```

	category	review title	review content	rating
40	Controllers & Attachments	Controller	Excellent condition highly recommended	5
			Verified purchase: No	5
			This is a great game for all switch owners. Th...	5
43	Video Games	the best Smash Bros. game	Incredible amount of content, very nostalgic a...	5


```
df['review content'].iloc[40:50]
```

```
40              Excellent condition highly recommended
41                                 Verified purchase: No
42    This is a great game for all switch owners. Th...
43    Incredible amount of content, very nostalgic a...
44    Won't be able to test the game until I get my ...
45    This game is the best Super Smash Bros. Crosso...
46    A great game! It has so many extras and Easter...
47    This game is off the charts! So many different...
48    The game is exactly what my children wanted, T...
49    Works great! My son loves this fighting game, ...
Name: review content, dtype: object
```

```
df['review title'].iloc[40:50]
```

```
40                                    Controller
41                                             R
42     Best fighting game on the Switch so far!
43                      the best Smash Bros. game
44    Arrived early with an included blank case
45                                Excelent Game!
46    Great Game for Experienced and New Gamers
47                          Best Smash Bros. yet!
48                              Great family game
49      Great fighting game for younger players
Name: review title, dtype: object
```

Preprocessing

1. Cleaning Data


```
    beauti = BeautifulSoup(data,'html.parser')
    return beauti.get_text()

def convert_emoji(data):
    return emoji.demojize(data)

def url_remover(data):
    return re.sub('(http|https):\/\/\S+', '',data)

def remove_round_brackets(data):
    return re.sub('\(.*?\)','',data)

def remove_punc(data):
    document = re.sub(r'[^\w\s]','', data)
    return document

def white_space(data):
    return ' '.join(data.split())

def text_lower(data):
    return data.lower()

def contraction_replace(data):
    return contractions.fix(data)

def remove_number(data):
    return re.sub(r"\d+", "", data)

def remove_singl_char(data):
    return re.sub(r"\b[a-zA-Z]\b", "", data)

def web_associated(data):
```

```
    new_data = html_remover(data)
    new_data = convert_emoji(new_data)
    new_data = url_remover(new_data)
    new_data = remove_round_brackets(new_data)
    new_data = text_lower(new_data)
    new_data = contraction_replace(new_data)
    new_data = remove_number(new_data)
    new_data = remove_singl_char(new_data)
    return new_data

df['cleaning data'] = df['review content'].apply(web_associated)
```



```
/usr/local/lib/python3.7/dist-packages/bs4/__init__.py:273: UserWarning: "b'.'" looks like a filename, not markup. You
  ' Beautiful Soup.' % markup)
/usr/local/lib/python3.7/dist-packages/bs4/__init__.py:273: UserWarning: "b'..'" looks like a filename, not markup. Yo
  ' Beautiful Soup.' % markup)
```

```
df.head()
```

	category	review title	review content	rating	cleaning data
0	Headsets	Wireless gaming headset	This gaming headset ticks all the boxes # look...	5	this gaming headset ticks all the boxes looks ...
1	Headsets	Good for those with a big head, low budget	Easy setup, rated for 6 hours battery but mine...	3	easy setup rated for hours battery but mine h...
2	Headsets	MezumiWireless Gaming Headset	I originally bought this wireless headset for ...	5	originally bought this wireless headset for m...
			This is my 2nd Mezumi headset. It		this is my nd mezumi headset it

```
df.iloc[:10]
```

	category	review title	review content	rating	cleaning data
0	Headsets	Wireless gaming headset	This gaming headset ticks all the boxes # look...	5	this gaming headset ticks all the boxes looks ...
1	Headsets	[illegible]	[illegible]d for 6 hours [illegible]y but mine...	3	easy setup rated for hours battery but mine h...
2	Headsets	MezumiWireless Gaming Headset	I originally bought this wireless headset for ...	5	originally bought this wireless headset for m...
3	Headsets	HW- S2 great headset.	This is my 2nd Mezumi headset, It kills the fi...	5	this is my nd mezumi headset it kills the firs...
4	Headsets	BEST HEADPHONES I'VE PURCHASED IN MY ENTIRE LIFE	This is probably the best headset I've purchas...	5	this is probably the best headset have purcha...
5	Headsets	[illegible]	This headsert is great for the	5	this headsert is great for the

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
df['cleaning data'].iloc[:10]
```

```
0    this gaming headset ticks all the boxes looks ...
1    easy setup rated for  hours battery but mine h...
2     originally bought this wireless headset for m...
3    this is my nd mezumi headset it kills the firs...
4    this is probably the best headset  have purcha...
5    this headsert is great for the value  recently...
6                     happy with product as described
7    verified purchase yes condition new sold by oz...
8                       all good product as described
9    no issues good distance on blutooth clear audi...
Name: cleaning data, dtype: object
```

2. Tokenizing

```
def tokenize(data):
    return nltk.word_tokenize(data)
```

```
df['tokenizing'] = df['cleaning data'].apply(tokenize)
```

3. Negation Handling


```
def Negation(sentence):
    temp = int(0)
    for i in range(len(sentence)):
        if sentence[i-1] in ['not',"n't"]:
            antonyms = []
            for syn in wordnet.synsets(sentence[i]):
                syns = wordnet.synsets(sentence[i])
                w1 = syns[0].name()
                temp = 0
                for l in syn.lemmas():
                    if l.antonyms():
                        antonyms.append(l.antonyms()[0].name())
                max_dissimilarity = 0
                for ant in antonyms:
                    syns = wordnet.synsets(ant)
                    w2 = syns[0].name()
                    syns = wordnet.synsets(sentence[i])
                    w1 = syns[0].name()
                    word1 = wordnet.synset(w1)
                    word2 = wordnet.synset(w2)
                    if isinstance(word1.wup_similarity(word2), float) or isinstance(word1.wup_similarity(word2), int):
                        temp = 1 - word1.wup_similarity(word2)
                    if temp>max_dissimilarity:
                        max_dissimilarity = temp
                        antonym_max = ant
                        sentence[i] = antonym_max
                        sentence[i-1] = ''
    while '' in sentence:
        sentence.remove('')
    return sentence
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
```

```
df['negation'] = df['tokenizing'].apply(Negation)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Stopword

```
def stopword(data):
    nltk.download('stopwords')
    clean = []
    for i in data:
        if i not in stopwords.words('english'):
            clean.append(i)
    return clean
```

```
df['stopword'] = df['negation'].apply(stopword)
```

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

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

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

Automatic saving failed. This file was updated remotely or in another tab. Show diff

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

4. Lemmatization

[ ] ↳ *3 cells hidden*

5. Pos Tagging

```
#POS tagger dictionary
post_dict = {'J':wordnet.ADJ, 'V':wordnet.VERB, 'N':wordnet.NOUN, 'R':wordnet.ADV}

def pos_tagging(tokens):
    return pos_tag(tokens)
```

```
import nltk
nltk.download('averaged_perceptron_tagger')

df['pos_tag'] = df['lemma'].apply(pos_tagging)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
```

```
df
```

	category	review title	review content	rating	cleaning data	tokenizing	negation	stopword	lemma	pos_t
			This gaming headset		this ticks all the boxes looks ...	[this, gaming, headset, ticks, all, the, boxes...	[this, gaming, headset, ticks, all, the, boxes...	[gaming, headset, ticks, boxes, looks, grate, ...	[game, headset, tick, box, look, grate, build,...	[(gan N (heads NN), (ti NN), (box,
1	Headsets	Good for those with a big head, low budget	Easy setup, rated for 6 hours battery but mine...	3	easy setup rated for hours battery but mine h...	[easy, setup, rated, for, hours, battery, but,...	[easy, setup, rated, for, hours, battery, but,...	[easy, setup, rated, hours, battery, mine, las...	[easy, setup, rat, hours, battery, mine, last,...	[(easy, J (setup, N (rat, N (hours, N
2	Headsets	MezumiWireless Gaming Headset	I originally bought this wireless headset for ...	5	originally bought this wireless headset for m...	[originally, bought, this, wireless, headset, ...	[originally, bought, this, wireless, headset, ...	[originally, bought, wireless, headset, xbox, ...	[originally, buy, wireless, headset, xbox, plu...	[(origina RB), (b V (wirele JJ),
3	Headsets	HW- S2 great headset.	This is my 2nd Mezumi headset, It kills the fi...	5	this is my nd mezumi headset it kills the firs...	[this, is, my, nd, mezumi, headset, it, kills,...	[this, is, my, nd, mezumi, headset, it, kills,...	[nd, mezumi, headset, kills, first, one, bette...	[nd, mezumi, headset, kill, first, one, better...	[(nd, J (mezu N (heads NN), (kil
		BEST	This is		this is					
4	Headsets	PURCHASED	headset	5	headset	the, best,	the, best,	headset,	headset,	JJ

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
df.to_csv('preprocessing1.csv', index=False)
```

```
df = pd.read_csv('preprocessing1.csv')
df
```

	category	review title	review content	rating	cleaning data	tokenizing	negation	stopword	lemma	p
			This gaming headset ticks all the boxes # look...		this ticks all the boxes looks ...	his', ing', headset', 'ticks', 'all', ...	['this', 'gaming', 'headset', 'ticks', 'all', ...	['gaming', 'headset', 'ticks', 'boxes', 'looks...	['game', 'headset', 'tick', 'box', 'look', 'gr...	[('h 'NN'
1	Headsets	Good for those with a big head, low budget	Easy setup, rated for 6 hours battery but mine...	3	easy setup rated for hours battery but mine h...	['easy', 'setup', 'rated', 'for', 'hours', 'ba...	['easy', 'setup', 'rated', 'for', 'hours', 'ba...	['easy', 'setup', 'rated', 'hours', 'battery',...	['easy', 'setup', 'rat', 'hours', 'battery', '...	[('eas ('setup ('rat
2	Headsets	MezumiWireless Gaming Headset	I originally bought this wireless headset for ...	5	originally bought this wireless headset for m...	['originally', 'bought', 'this', 'wireless', '...	['originally', 'bought', 'this', 'wireless', '...	['originally', 'bought', 'wireless', 'headset'...	['originally', 'buy', 'wireless', 'headset', '...	[('ori 'RB' ('
3	Headsets	HW- S2 great headset.	This is my 2nd Mezumi headset, It kills the fi...	5	this is my nd mezumi headset it kills the firs...	['this', 'is', 'my', 'nd', 'mezumi', 'headset'...	['this', 'is', 'my', 'nd', 'mezumi', 'headset'...	['nd', 'mezumi', 'headset', 'kills', 'first', ...	['nd', 'mezumi', 'headset', 'kill', 'first', '...	[('n ('m ('heac
4	Headsets	BEST HEADPHONES I'VE PURCHASED IN MY ENTIRE LIFE	This is probably the best headset I've purchas...	5	this is probably the best headset have purcha...	['this', 'is', 'probably', 'the', 'best', 'hea...	['this', 'is', 'probably', 'the', 'best', 'hea...	['probably', 'best', 'headset', 'purchased', '...	['probably', 'best', 'headset', 'purchase', 't...	[('pr 'RB') ('h
...	...	...	...	...	...	...	...	...	...	
	Racks &		Reasonably		reasonably	['reasonably',	['reasonably',	['reasonably', 'priced'	['reasonably',	[('reas
	Racks &		As		as	Las	Las			[(de

Pelabelan

TextBlob

[ ] ↳ *8 cells hidden*

Vader Sentiment

[ ] ↳ *8 cells hidden*

Split Data

```
X_train, X_test, y_train, y_test = train_test_split(data_label.values, label, test_size = 0.10, random_state = 42)
```

```
X_test.shape
```

```
(4476,)
```

EKSTRAKSI FITUR: Term presence

```
count_vect = CountVectorizer(binary=True)
X_train_counts = count_vect.fit_transform(data_label)
print(X_train_counts.shape)
count_vect.vocabulary_
```

```
'straight': 26979,
'expected': 11008,
'grandkids': 13233,
'business': 5312,
'nerdy_nero': 19117,
'fast': 11382,
'shipping': 25400,
'maybe': 17800,
'next': 19198,
```

```
'us': 29813,
'woks': 31146,
'plays': 21280,
'chance': 6018,
```



```
'due': 9730,
'cov': 7608,
'19': 432,
'situation': 25723,
'switches': 27547,
'msrp': 18738,
'shortages': 25465,
'retailers': 23820,
'facing': 11250,

'plot': 21324,
'thanks': 28134,
'literally': 16972,
'aren': 3279,
'rate': 22792,
'exaggerating': 10844,
'may': 17798,
'played': 21268,
'written': 31338,
'deliberate': 8514,
'finely': 11691,
'crafted': 7667,
'disservice': 9275,
'merely': 18006,
'call': 5488,
'something': 26209,
'playable': 21264,
'literature': 16976,
'released': 23413,
'800': 1604,
'page': 20353,
'novel': 19458,
'certainly': 5969,
'pulitzer': 22357,
'tomyamkong': 28657,
'made': 17444,
```

```
'flawlessly': 11866,
'produced': 22032,
'addictive': 2221,
'varied': 30027,
```



```
X_train_TP = count_vect.transform(X_train)
X_test_TP = count_vect.transform(X_test)
```

```
print(X_train_TP)
```

```
  (0, 2633)     1
  (0, 5762)     1
  (0, 6833)     1
  (0, 13442)    1
  (0, 20958)    1
  (0, 31098)    1
  (1, 2896)     1
  (1, 2985)     1
  (1, 3046)     1
  (1, 3060)     1
  (1, 4097)     1
  (1, 4982)     1
  (1, 5578)     1
  (1, 5794)     1
  (1, 9240)     1
  (1, 9372)     1
  (1, 9440)     1
  (1, 10990)    1
  (1, 11681)    1
  (1, 12051)    1
  (1, 12410)    1
  (1, 12838)    1
  (1, 13057)    1
  (1, 13164)    1
  (1, 13293)    1
  :     :
  (40279, 13841)        1
  (40279, 14872)        1
```

```
(40279, 15323)        1
(40279, 15613)        1
(40279, 16876)        1
(40279, 16990)        1
(40279, 20958)        1
(40279, 21929)        1
(40279, 24508)        1
(40279, 24683)        1
(40279, 24974)        1
(40279, 25223)        1
(40279, 26112)        1
(40279, 28164)        1
(40279, 28489)        1
(40279, 28587)        1
(40279, 30154)        1
(40279, 30850)        1
(40279, 31098)        1
(40279, 31224)        1
(40279, 31259)        1
(40279, 31594)        1
```


KLASIFIKASI dengan term presence

```
SVM_Clasifier = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')

# fitting data latih pada classifier
SVM_Clasifier.fit(X_train_TP,y_train)
# memprediksi label pada set data uji
predictions_SVM_TP = SVM_Clasifier.predict(X_test_TP)

# Menggunakan fungsi accuracy_score untuk mendapat nilai akurasi
print('Confusion Matrix: \n',confusion_matrix(y_test, predictions_SVM_TP))
print()
```

```
print('Accuracy: ', accuracy_score(y_test, predictions_SVM_TP))
```

```
Confusion Matrix:
```


EKSTRAKSI FITUR: TF-IDF

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(data_label)
features_train_transformed = vectorizer.transform(X_train)
features_test_transformed = vectorizer.transform(X_test)
print(features_train_transformed)
```

```
  (0, 31098)    0.16060403659989436
  (0, 20958)    0.20557964439899906
  (0, 13442)    0.5646791905873499
  (0, 6833)     0.44937783934378367
  (0, 5762)     0.6052617008202428
  (0, 2633)     0.21165678817632028
  (1, 31559)    0.12530476186850223
  (1, 31276)    0.1752593827260213
  (1, 31098)    0.05936444445905331
  (1, 30830)    0.09118542085375923
  (1, 30526)    0.11141146984859461
  (1, 29756)    0.12948409823333595
  (1, 28587)    0.09529107402231114
  (1, 28335)    0.10967092845820321
  (1, 28265)    0.09870787055457755
  (1, 28164)    0.08277150642338653
  (1, 28148)    0.0667745718404276
  (1, 26943)    0.14968731321591422
  (1, 26112)    0.07260982221489783
  (1, 24983)    0.11929864846979946
  (1, 23893)    0.05564301748033883
  (1, 23507)    0.19684493954329005
```

```
(1, 22878)    0.05807444496382093
(1, 20968)    0.1338013947774766
(1, 20958)    0.1519777664398298
:     :
```



```
(40279, 17224)        0.09185808100565548
(40279, 16990)        0.11370211696098893
(40279, 16876)        0.08837575737789898
(40279, 15613)        0.05159310599229723
(40279, 15323)        0.3080124455866271
(40279, 14872)        0.07075384099492378
(40279, 13841)        0.15627370255685713
(40279, 13798)        0.1901372538162533
(40279, 13269)        0.2516740902663807
(40279, 12661)        0.12741006011224043
(40279, 12275)        0.23853239997345266
(40279, 11702)        0.19596409458684458
(40279, 11382)        0.1157897150705048
(40279, 10773)        0.13582861583144784
(40279, 10746)        0.11450474507400692
(40279, 9440) 0.11581863009611014
(40279, 9403) 0.13380303003102784
(40279, 8504) 0.22495719979885334
(40279, 5327) 0.07297891540865092
(40279, 3521) 0.09645936741797435
(40279, 3439) 0.21338209402387953
(40279, 2896) 0.0964904377493778
```

KLASIFIKASI dengan TF-IDF

```
SVM_Clasifier = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
mymodel= SVM_Clasifier.fit(features_train_transformed,y_train)
```

```
#Evaluate the model on the training data set

predictions_SVM_Tfidf2 = SVM_Clasifier.predict(features_train_transformed)
```

```
print(classification_report(y_train, predictions_SVM_Tfidf2))
print('Confusion Matrix: \n',confusion_matrix(y_train, predictions_SVM_Tfidf2))
print()
print('Accuracy: ', accuracy_score(y_train, predictions_SVM_Tfidf2))
```



```
              precision    recall  f1-score   support

    Negative       0.97      0.66      0.79      1841
     Neutral       0.98      0.95      0.96      4517
    Positive       0.98      1.00      0.99     33922

    accuracy                           0.98     40280
   macro avg       0.98      0.87      0.91     40280
weighted avg       0.98      0.98      0.98     40280

Confusion Matrix:
 [[ 1221    51   569]
 [   19  4289   209]
 [   21    35 33866]]

Accuracy:  0.9775571002979146
```

```
#Evaluate the model on the testing data set

predictions_SVM_Tfidf = SVM_Clasifier.predict(features_test_transformed)
print(classification_report(y_test, predictions_SVM_Tfidf))
print('Confusion Matrix: \n',confusion_matrix(y_test, predictions_SVM_Tfidf))
print()
print('Accuracy: ', accuracy_score(y_test, predictions_SVM_Tfidf))
```

```
X_train.shape
```

```
(40280,)
```

```
y_train.shape
```

```
(40280,)
```

Evaluation with K-Fold and Classification Report:


```
# vectorizer = TfidfVectorizer()
# X = vectorizer.fit_transform(data_label)
kf = KFold(n_splits=11, shuffle=True, random_state=42)
scores = []
for fold, (train_index, test_index) in enumerate(kf.split(X_train,y_train), 1):
    x_train, x_test = X_train[train_index], X_train[test_index]
    # y_train1, y_test1 = y_train[train_index], y_train[test_index]

    vectorizer = TfidfVectorizer()
    vectorizer.fit_transform(X_train)
    features_train_transformed1 = vectorizer.transform(X_train)
    features_test_transformed1 = vectorizer.transform(X_test)

    SVM_Clasifier = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
    SVM_Clasifier.fit(features_train_transformed1,y_train)
    predictions_SVM_Tfidf = SVM_Clasifier.predict(features_test_transformed1)

    print(f'# Fold {fold}, Train set: {len(train_index)}, Test set:{len(test_index)}')
    print(classification_report(y_test, predictions_SVM_Tfidf), "\n")
    print('Accuracy: ', accuracy_score(y_test, predictions_SVM_Tfidf))
```

```
 Accuracy:  0.9557640750670241
 # Fold 6, Train set: 36618, Test set:3662
              precision    recall  f1-score   support

    Negative       0.88      0.36      0.52       184
     Neutral       0.93      0.90      0.91       509
    Positive       0.96      0.99      0.98      3783

    accuracy                           0.96      4476
   macro avg       0.92      0.75      0.80      4476
```

```
weighted avg       0.95      0.96      0.95      4476

Accuracy:  0.9557640750670241
# Fold 7, Train set: 36618, Test set:3662
```



```
     Neutral       0.93      0.90      0.91       509
    Positive       0.96      0.99      0.98      3783

    accuracy                           0.96      4476
   macro avg       0.92      0.75      0.80      4476
weighted avg       0.95      0.96      0.95      4476


Accuracy:  0.9557640750670241
# Fold 8, Train set: 36618, Test set:3662
              precision    recall  f1-score   support

    Negative       0.88      0.36      0.52       184

     Neutral       0.93      0.90      0.91       509
    Positive       0.96      0.99      0.98      3783

    accuracy                           0.96      4476
   macro avg       0.92      0.75      0.80      4476
weighted avg       0.95      0.96      0.95      4476


Accuracy:  0.9557640750670241
# Fold 9, Train set: 36618, Test set:3662
              precision    recall  f1-score   support

    Negative       0.88      0.36      0.52       184
     Neutral       0.93      0.90      0.91       509
    Positive       0.96      0.99      0.98      3783

    accuracy                           0.96      4476
   macro avg       0.92      0.75      0.80      4476
weighted avg       0.95      0.96      0.95      4476


Accuracy: 0.9557640750670241
```

```
Accuracy:  0.9557640750670241
# Fold 10, Train set: 36619, Test set:3661
              precision    recall  f1-score   support
```



```
import collections, numpy
print("Jumlah review: {}".format(len(data_label)))
SVM = collections.Counter(predictions_SVM_Tfidf)
print("Hasil Klasifikasi SVM : ", SVM)

results = pd.DataFrame({
    "Labeled_Data" : data_label,
    "Label" : SVM_prediction
 })
results.to_csv("Hasil_SVM2.csv", index = False)
```

```
Jumlah review: 44756
Hasil Klasifikasi SVM :  Counter({'Positive': 3908, 'Neutral': 492, 'Negative': 76})
Jumlah review: 44756
Hasil Klasifikasi SVM :  Counter({'Positive': 3908, 'Neutral': 492, 'Negative': 76})
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

2m 13s completed at 08:48