

Amazon Fine Food Reviews Analysis

Data Source: <https://www.kaggle.com/snap/amazon-fine-food-reviews> (<https://www.kaggle.com/snap/amazon-fine-food-reviews>)

EDA: <https://nycdatascience.com/blog/student-works/amazon-fine-foods-visualization/> (<https://nycdatascience.com/blog/student-works/amazon-fine-foods-visualization/>)

The Amazon Fine Food Reviews dataset consists of reviews of fine foods from Amazon.

Number of reviews: 568,454

Number of users: 256,059

Number of products: 74,258

Timespan: Oct 1999 - Oct 2012

Number of Attributes/Columns in data: 10

Attribute Information:

1. Id
2. ProductId - unique identifier for the product
3. UserId - unique identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or not
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - brief summary of the review
10. Text - text of the review

Objective:

Given a review, determine whether the review is positive (rating of 4 or 5) or negative (rating of 1 or 2).

[Q] How to determine if a review is positive or negative?

[Ans] We could use Score/Rating. A rating of 4 or 5 can be considered as a positive review. A rating of 1 or 2 can be considered as negative one. A review of rating 3 is considered neutral and such reviews are ignored from our analysis. This is an approximate and proxy way of determining the polarity (positivity/negativity) of a review.

[1]. Reading Data

[1.1] Loading the data

The dataset is available in two forms

1. .csv file
2. SQLite Database

In order to load the data, We have used the SQLITE dataset as it is easier to query the data and visualise the data efficiently.

Here as we only want to get the global sentiment of the recommendations (positive or negative), we will purposefully ignore all Scores equal to 3. If the score is above 3, then the recommendation will be set to "positive". Otherwise, it will be set to "negative".

```
In [78]: %matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import sqlite3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from tqdm import tqdm
from bs4 import BeautifulSoup

import re
import string
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import PorterStemmer
from nltk.stem import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer

from sklearn.model_selection import GridSearchCV
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer, TfidfTransformer
from sklearn.metrics import confusion_matrix, accuracy_score, roc_auc_score, auc, roc_curve, classification_report, precision_score, recall_score, f1_score, hamming_loss

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import TruncatedSVD
from prettytable import PrettyTable

from sklearn.svm import SVC

from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle

from tqdm import tqdm
import os
```

```
In [2]: # using SQLite Table to read data.
con = sqlite3.connect('D:\Study_materials\Applied_AI\Assignments\database.sqlite')

filtered_data = pd.read_sql_query(""" SELECT * FROM Reviews WHERE Score != 3""", con)

# Give reviews with Score>3 a positive rating(1), and reviews with a score<3 a negative rating(0).
def partition(x):
    if x < 3:
        return 0
    return 1

#changing reviews with score less than 3 to be positive and vice-versa
actualScore = filtered_data['Score']
positiveNegative = actualScore.map(partition)
filtered_data['Score'] = positiveNegative
print("Number of data points in our data", filtered_data.shape)
```

Number of data points in our data (525814, 10)

```
In [3]: sample_data = filtered_data.head(100000)
```

[2] Exploratory Data Analysis

[2.1] Data Cleaning: Deduplication

It is observed that the reviews data had many duplicate entries. Hence it was necessary to remove duplicates in order to get unbiased results for the analysis of the data. Following is an example:

```
In [4]: #Sorting data according to ProductId in ascending order
sorted_data=sample_data.sort_values('ProductId', axis=0, ascending=True, inplace=False, kind='quicksort', na_position='last')
```

```
In [5]: #Deduplication of entries
final=sorted_data.drop_duplicates(subset={"UserId","ProfileName","Time","Text"}, keep='first', inplace=False)
final.shape
```

Out[5]: (87775, 10)

```
In [6]: #Checking to see how much % of data still remains
(final['Id'].size*1.0)/(sample_data['Id'].size*1.0)*100
```

Out[6]: 87.775

```
In [7]: final=final[final.HelpfulnessNumerator<=final.HelpfulnessDenominator]
```



```
In [12]: #Code for implementing step-by-step the checks mentioned in the pre-processing phase
# this code takes a while to run as it needs to run on 500k sentences.

final_string=[]
all_positive_words=[] # store words from +ve reviews here
all_negative_words=[] # store words from -ve reviews here.
for i, sent in enumerate(tqdm(final['Text'].values)):
    filtered_sentence=[]
    sent=cleanhtml(sent) # remove HTML tags
    for w in sent.split():
        # we have used cleanpunc(w).split(), one more split function here because consider w="abc.def", cleanpunc(w) will return
        "abc def"
        # if we dont use .split() function then we will be considring "abc def" as a single word, but if you use .split() function
        we will get "abc", "def"
    for cleaned_words in cleanpunc(w).split():
        if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
            if(cleaned_words.lower() not in stop):
                s=(sno.stem(cleaned_words.lower())).encode('utf8') #snowball stemmer
                filtered_sentence.append(s)
                if (final['Score'].values[i] == 1:
                    all_positive_words.append(s) #list of all words used to describe positive reviews
                if(final['Score'].values[i] == 0:
                    all_negative_words.append(s) #list of all words used to describe negative reviews reviews
    str1 = " ".join(filtered_sentence) #final string of cleaned words
    #print("*****")
    final_string.append(str1)

#####---- storing the data into .sqlite file -----#####
final['CleanedText']=final_string #adding a column of CleanedText which displays the data after pre-processing of the review
final['CleanedText']=final['CleanedText'].str.decode("utf-8")
```

[illegible]

```
In [13]: final = final.sort_values('Time',axis = 0,ascending = True, inplace = False, kind = 'quicksort', na_position='last')
```

```
In [14]: final.columns
```

```
Out[14]: Index(['Id', 'ProductId', 'UserId', 'ProfileName', 'HelpfulnessNumerator',
               'HelpfulnessDenominator', 'Score', 'Time', 'Summary', 'Text',
               'CleanedText'],
              dtype='object')
```

```
In [15]: X = final['CleanedText'].values
          y = final['Score']
```

```
In [16]: # Creating training, test and cross validation set
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size= 0.3, random_state=0)
X_tr,X_cv,y_tr,y_cv = train_test_split(X_train,y_train, test_size = 0.3, random_state=0)
```

```
In [17]: print("Size of X_train and y_train:", X_train.shape,y_train.shape)
print("Size of X_test and y_test:", X_test.shape,y_test.shape)
print("Size of X_tr and y_tr:", X_tr.shape,y_tr.shape)
print("Size of X_cv and y_cv:", X_cv.shape,y_cv.shape)
```

```
Size of X_train and y_train: (61441,) (61441,)
Size of X_test and y_test: (26332,) (26332,)
Size of X_tr and y_tr: (43008,) (43008,)
Size of X_cv and y_cv: (18433,) (18433,)
```

SVM with SGD Classifier for Linear Kernel:

```
In [32]: from sklearn.linear_model import SGDClassifier
from sklearn.calibration import CalibratedClassifierCV

# SVM with penalty = 'L2'

def sgd_Classifier(X_train,X_cv,y_train,y_cv):
    best_alpha=0
    max_roc_auc=-1
    pred_train = []
    pred_cv = []
    alpha = [10 ** x for x in range(-4, 4)]
    for i in alpha:
        clf = SGDClassifier(alpha=i, penalty='l2', loss='hinge', random_state=1)
        clf.fit(X_train,y_train)
        sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
        sig_clf.fit(X_train, y_train)
        prob_train = sig_clf.predict_proba(X_train)[: ,1]
        prob_cv = sig_clf.predict_proba(X_cv)[: ,1]
        auc_score_train = roc_auc_score(y_train,prob_train)
        auc_score_cv = roc_auc_score(y_cv,prob_cv)

        print(i," for train data, auc score is: ",auc_score_train," and for CV data, auc score is: ",auc_score_cv)
        pred_train.append(auc_score_train)
        pred_cv.append(auc_score_cv)
        if(max_roc_auc<auc_score_cv):
            max_roc_auc=auc_score_cv
            best_alpha=i
    print(f"\n Best C Value {best_alpha} with highest roc_auc Score is {max_roc_auc}")
    sns.set_style("darkgrid")
    plt.xscale('log')
    plt.plot(alpha,pred_train,'g-', label = 'Train Data')
    plt.plot(alpha, pred_cv,'r-', label = 'CV Data')
    plt.legend(loc='upper right')
    plt.title(r'Auc Score v/s $\alpha$')
    plt.xlabel(r"alpha values",fontsize=12)
    plt.ylabel("roc_auc",fontsize=12)
    plt.show()
    # calculate roc curve for train data
    fpr, tpr, thresholds = roc_curve(y_train,prob_train)
    plt.plot([0, 1], [0, 1], linestyle='--')
    plt.plot(fpr, tpr, marker='.')
    plt.title("ROC Curve on Train Data")
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    plt.show()

    # calculate roc curve for test data
    fpr, tpr, thresholds = roc_curve(y_test,prob_test)
    plt.plot([0, 1], [0, 1], linestyle='--')
    plt.plot(fpr, tpr, marker='.')
    plt.title("ROC Curve on Test Data")
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    plt.show()

def find_best_threshold(threshold, fpr, tpr):
    t = threshold[np.argmax(tpr*(1-fpr))]
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshold):
    predictions = []
    for i in proba:
        if i>=threshold:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions

best_t = find_best_threshold(thresholds, fpr, tpr)
prediction = predict_with_best_t(prob_test, best_t)
skplt.plot_confusion_matrix(y_test,prediction)

print("macro f1 score for data :",f1_score(y_test, prediction, average = 'macro'))
print("micro f1 scoore for data:",f1_score(y_test, prediction, average = 'micro'))
print("hamming loss for data:",hamming_loss(y_test,prediction))
print("Precision recall report for data:\n",classification_report(y_test, prediction))
```

Testing model:

```
In [39]: import scikitplot.metrics as skplt

def testing(X_train,y_train,X_test,y_test,optimal_alpha):
    clf = SGDClassifier(alpha=optimal_alpha, penalty='l2', loss='hinge', random_state=1)
    clf.fit(X_train,y_train)
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(X_train, y_train)
    prob_test = sig_clf.predict_proba(X_test)[: ,1]
    print("AUC Score",roc_auc_score(y_test,prob_test))

    fpr, tpr, thresholds = roc_curve(y_test,prob_test)
    plt.plot([0, 1], [0, 1], linestyle='--')
    plt.plot(fpr, tpr, marker='.')
    plt.title("ROC Curve on Test Data")
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    plt.show()

def find_best_threshold(threshold, fpr, tpr):
    t = threshold[np.argmax(tpr*(1-fpr))]
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshold):
    predictions = []
    for i in proba:
        if i>=threshold:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions

best_t = find_best_threshold(thresholds, fpr, tpr)
prediction = predict_with_best_t(prob_test, best_t)
skplt.plot_confusion_matrix(y_test,prediction)

print("macro f1 score for data :",f1_score(y_test, prediction, average = 'macro'))
print("micro f1 scoore for data:",f1_score(y_test, prediction, average = 'micro'))
print("hamming loss for data:",hamming_loss(y_test,prediction))
print("Precision recall report for data:\n",classification_report(y_test, prediction))
```

Top 10 features:

```
def imp_feature(vectorizer, classifier, n = 10):
    feature_names = vectorizer.get_feature_names()
    coefs_with_fns = sorted(zip(classifier.coef_[0], feature_names))
    top = zip(coefs_with_fns[:n], coefs_with_fns[:-(n + 1):-1])
    print("\t\t\tNegative\t\t\t\t\tPositive")
    print("_____")
    for (coef_1, fn_1), (coef_2, fn_2) in top:
        print("\t%.4f\t%-15s\t\t\t\t\t%.4f\t%-15s" % (coef_1, fn_1, coef_2, fn_2))
```

1. Bag of Words (BoW)

```
count_vec = CountVectorizer()
BOW_X_train = count_vec.fit_transform(X_tr)
BOW_X_cv = count_vec.transform(X_cv)
BOW_X_test = count_vec.transform(X_test)
```

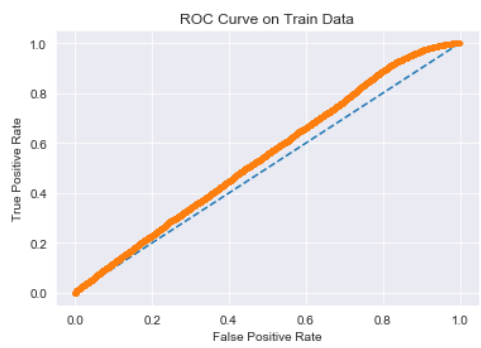
```
#Standardizing data using StandardScaler

sc = StandardScaler(with_mean=False)
BOW_X_train_sc = sc.fit_transform(BOW_X_train)
BOW_X_cv_sc = sc.transform(BOW_X_cv)
BOW_X_test_sc = sc.transform(BOW_X_test)
```

```
sgd_Classifier(BOW_X_train_sc,BOW_X_cv_sc,y_tr,y_cv)
```

```
0.0001 for train data, auc score is: 0.9936641297680245 and for CV data, auc score is: 0.8486832336756638
0.001 for train data, auc score is: 0.9945965420950227 and for CV data, auc score is: 0.8597042471520718
0.01 for train data, auc score is: 0.99436164329057121 and for CV data, auc score is: 0.89348818837383556
0.1 for train data, auc score is: 0.986726173360596 and for CV data, auc score is: 0.9030738169303547
1 for train data, auc score is: 0.9547757624882549 and for CV data, auc score is: 0.8735032187154064
10 for train data, auc score is: 0.6964633829994923 and for CV data, auc score is: 0.6447552068975213
100 for train data, auc score is: 0.5541097134984971 and for CV data, auc score is: 0.5360018094570587
1000 for train data, auc score is: 0.5489108107136415 and for CV data, auc score is: 0.5319400356302558
```

Best C Value 0.1 with highest roc_auc Score is 0.9030738169303547



```
In [42]: testing(BOW_X_train_sc,y_tr,BOW_X_test_sc,y_test,optimal_alpha = 0.1)
```

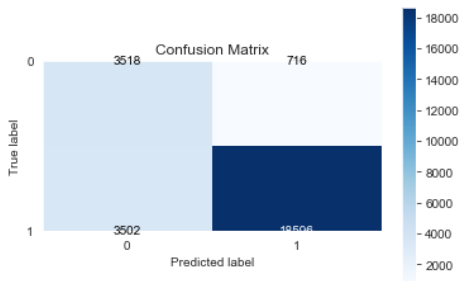
AUC Score 0.9014123135858976



the maximum value of $tpr*(1-fpr)$ 0.6992163092965065 for threshold 0.83
macro f1 score for data : 0.7616702373380286
micro f1 score for data: 0.839814674160717
hamming loss for data: 0.160185325839283

Precision recall report for data:

	precision	recall	f1-score	support
0	0.50	0.83	0.63	4234
1	0.96	0.84	0.90	22098
accuracy			0.84	26332
macro avg	0.73	0.84	0.76	26332
weighted avg	0.89	0.84	0.85	26332



Top 10 Features:

```
In [43]: clf = SGDClassifier(alpha=0.1, penalty='l2', loss='hinge', random_state=1)
clf.fit(BOW_X_train_sc,y_tr)
imp_feature(count_vec,clf)
```

Negative		Positive	
-0.1200	disappoint	0.2106	great
-0.0827	worst	0.1925	love
-0.0761	aw	0.1549	good
-0.0752	terribl	0.1403	best
-0.0746	horribl	0.1202	delici
-0.0685	return	0.0944	favorit
-0.0658	unfortun	0.0910	excel
-0.0637	bad	0.0907	perfect
-0.0587	threw	0.0856	nice
-0.0571	wast	0.0812	tasti

2. TF-IDF

```
In [44]: tfidf_vec = TfidfVectorizer(ngram_range=(1,2))
tfidf_train = tfidf_vec.fit_transform(X_tr)
tfidf_cv = tfidf_vec.transform(X_cv)
tfidf_test = tfidf_vec.transform(X_test)

print("The type of count vectorizer ",type(tfidf_train))
print("The shape of out text TFIDF vectorizer ",tfidf_train.get_shape())
print("Size of CV dataset:", tfidf_cv.shape)
print("Size of test dataset:", tfidf_test.shape)
```

The type of count vectorizer <class 'scipy.sparse.csr.csr_matrix'>
The shape of out text TFIDF vectorizer (43008, 683623)
Size of CV dataset: (18433, 683623)
Size of test dataset: (26332, 683623)

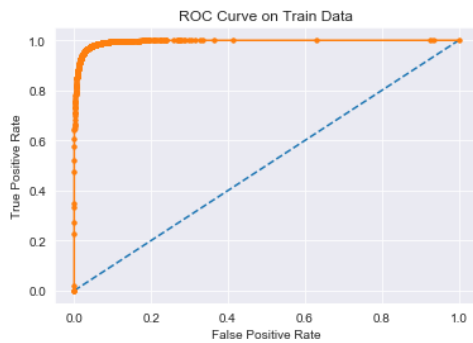
In [45]: #Standardizing data using StandardScaler

```
sc = StandardScaler(with_mean=False)
tfidf_train_sc = sc.fit_transform(tfidf_train)
tfidf_cv_sc = sc.transform(tfidf_cv)
tfidf_test_sc = sc.transform(tfidf_test)
```

In [46]: sgf_Classifier(tfidf_train_sc,tfidf_cv_sc,y_tr,y_cv)

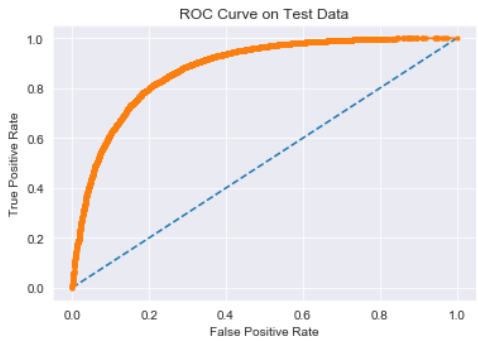
0.0001 for train data, auc score is: 0.9999512634982233 and for CV data, auc score is: 0.8056273799165158
 0.001 for train data, auc score is: 0.9999413586569227 and for CV data, auc score is: 0.8035302458158277
 0.01 for train data, auc score is: 0.9999707954008202 and for CV data, auc score is: 0.8035615845678963
 0.1 for train data, auc score is: 0.999997718235677 and for CV data, auc score is: 0.8289205479634455
 1 for train data, auc score is: 0.99999866070355 and for CV data, auc score is: 0.852079206845312
 10 for train data, auc score is: 0.9999814065969894 and for CV data, auc score is: 0.8825466496267952
 100 for train data, auc score is: 0.9964732197456504 and for CV data, auc score is: 0.7051166436673952
 1000 for train data, auc score is: 0.9949876840282609 and for CV data, auc score is: 0.6808224094458983

Best C Value 10 with highest roc_auc Score is 0.8825466496267952



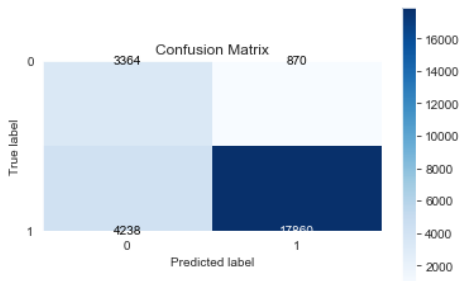

```
In [47]: testing(tfidf_train_sc,y_tr,tfidf_test_sc,y_test,optimal_alpha = 10)
```

AUC Score 0.8786545295523659



the maximum value of $tpr \cdot (1 - fpr)$ 0.6421457591773632 for threshold 0.905
macro f1 score for data : 0.7216625318562069
micro f1 score for data: 0.8060154944554154
hamming loss for data: 0.19398450554458455
Precision recall report for data:

	precision	recall	f1-score	support
0	0.44	0.79	0.57	4234
1	0.95	0.81	0.87	22098
accuracy			0.81	26332
macro avg	0.70	0.80	0.72	26332
weighted avg	0.87	0.81	0.83	26332



Top 10 Features:

```
In [48]: clf = SGDClassifier(alpha=10, penalty='l2', loss='hinge', random_state=1)
clf.fit(tfidf_train_sc,y_tr)
imp_feature(tf_idf_vec,clf)
```

Negative		Positive	
-0.0049	wast money	0.0188	love
-0.0048	disappoint	0.0179	great
-0.0047	worst	0.0149	good
-0.0044	horribl	0.0120	best
-0.0043	return	0.0098	delici
-0.0040	aw	0.0093	flavor
-0.0040	threw	0.0089	favorit
-0.0040	wast	0.0088	price
-0.0040	terribl	0.0087	like
-0.0034	disgust	0.0086	use

3. Avg-W2V

```
In [49]: i=0
list_sent_train=[]
for sent in X_tr:
    filtered_sentence=[]
    sent=cleanhtml(sent)
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if(cleaned_words.isalpha()):
                filtered_sentence.append(cleaned_words.lower())
            else:
                continue
    list_sent_train.append(filtered_sentence)
```

```
In [50]: i=0
list_sent_train1=[]
for sent in X_tr:
    filtered_sentence=[]
    sent=sent
    for w in sent.split():
        for cleaned_words in w.split():
            if(cleaned_words.isalpha()):
                filtered_sentence.append(cleaned_words.lower())
            else:
                continue
    list_sent_train1.append(filtered_sentence)
```

```
In [51]: i=0
list_sent_cv=[]
for sent in X_cv:
    filtered_sentence=[]
    sent=cleanhtml(sent)
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if(cleaned_words.isalpha()):
                filtered_sentence.append(cleaned_words.lower())
            else:
                continue
    list_sent_cv.append(filtered_sentence)
```

```
In [52]: i=0
list_sent_test=[]
for sent in X_test:
    filtered_sentence=[]
    sent=cleanhtml(sent)
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if(cleaned_words.isalpha()):
                filtered_sentence.append(cleaned_words.lower())
            else:
                continue
    list_sent_test.append(filtered_sentence)
```

```
In [53]: import gensim
w2v_model = gensim.models.Word2Vec(list_sent_train,min_count=5,size=50,workers=4)
w2v_words = list(w2v_model.wv.vocab)
```

```
In [54]: def avg_w2v(list_of_sent):
    sent_vectors = []; # the avg-w2v for each sentence/review is stored in this list
    for sent in list_of_sent: # for each review/sentence
        sent_vec = np.zeros(50) # as word vectors are of zero length
        cnt_words = 0; # num of words with a valid vector in the sentence/review
        for word in sent: # for each word in a review/sentence
            if word in w2v_words:
                vec = w2v_model.wv[word]
                sent_vec += vec
                cnt_words += 1
        if cnt_words != 0:
            sent_vec /= cnt_words
        sent_vectors.append(sent_vec)
    print(len(sent_vectors))
    print(len(sent_vectors[0]))
    return sent_vectors
```

```
In [55]: train_avgw2v = avg_w2v(list_sent_train)

43008
50
```

```
In [56]: cv_avgw2v = avg_w2v(list_sent_cv)

18433
50
```

```
In [57]: test_avgw2v = avg_w2v(list_sent_test)

26332
50
```

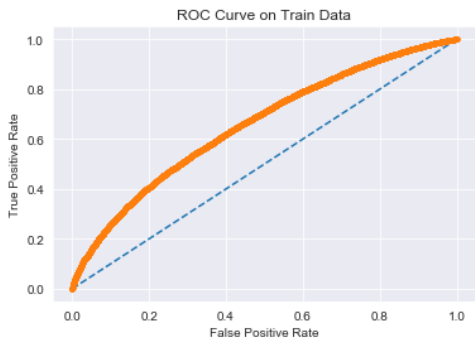
```
In [58]: #Standardizing data using StandardScaler

sc = StandardScaler(with_mean=False)
aw2v_X_train_sc = sc.fit_transform(train_avgw2v)
aw2v_X_cv_sc = sc.transform(cv_avgw2v)
aw2v_X_test_sc = sc.transform(test_avgw2v)
```

```
In [59]: sgd_Classifier(aw2v_X_train_sc,aw2v_X_cv_sc,y_tr,y_cv)
```

0.0001 for train data, auc score is: 0.8886771770948323 and for CV data, auc score is: 0.8929930098157965
0.001 for train data, auc score is: 0.8933152698287502 and for CV data, auc score is: 0.8981238708960798
0.01 for train data, auc score is: 0.8935538868208388 and for CV data, auc score is: 0.898586396712773
0.1 for train data, auc score is: 0.8914869377127547 and for CV data, auc score is: 0.8956439485023056
1 for train data, auc score is: 0.8694933923676691 and for CV data, auc score is: 0.8718704062229344
10 for train data, auc score is: 0.7229737734405649 and for CV data, auc score is: 0.7169716052072348
100 for train data, auc score is: 0.6579604189684378 and for CV data, auc score is: 0.6530919938445523
1000 for train data, auc score is: 0.6579605856364404 and for CV data, auc score is: 0.6530919938445523

Best C Value 0.01 with highest roc_auc Score is 0.898586396712773



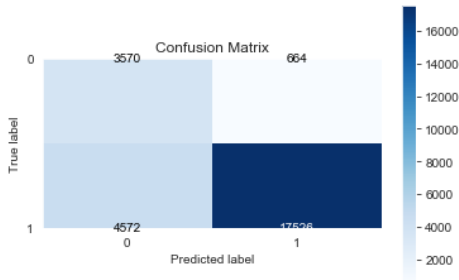
```
In [60]: testing(aw2v_X_train_sc,y_tr,aw2v_X_test_sc,y_test,optimal_alpha = 0.01)
```

AUC Score 0.8960868284888721



the maximum value of tpr*(1-fpr) 0.6687244474125715 for threshold 0.843
macro f1 score for data : 0.723479409787988
micro f1 score for data: 0.8011544888348777
hamming loss for data: 0.19884551116512228
Precision recall report for data:

	precision	recall	f1-score	support
0	0.44	0.84	0.58	4234
1	0.96	0.79	0.87	22098
accuracy			0.80	26332
macro avg	0.70	0.82	0.72	26332
weighted avg	0.88	0.80	0.82	26332



4. TF_IDF-W2V

26332
50
18.00607352955

```
In [66]: train_tfidf2v = tfidf_sent_vectors_train
cv_tfidf2v = tfidf_sent_vectors_cv
test_tfidf2v = tfidf_sent_vectors_test
```

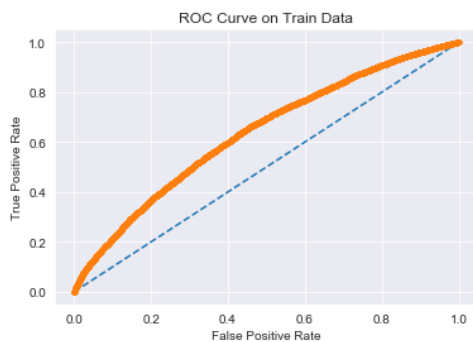
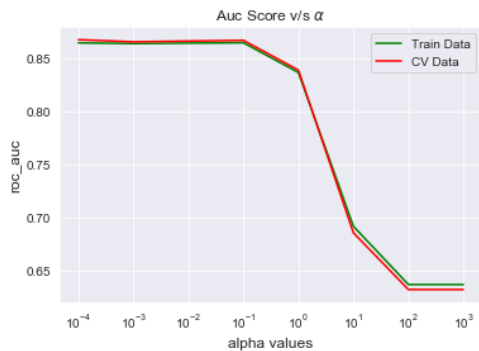
```
In [67]: #Standardizing data using StandardScaler

sc = StandardScaler(with_mean=False)
tfidf2v_X_train_sc = sc.fit_transform(train_tfidf2v)
tfidf2v_X_cv_sc = sc.transform(cv_tfidf2v)
tfidf2v_X_test_sc = sc.transform(test_tfidf2v)
```

```
In [68]: sgdc_Classifier(tfidf2v_X_train_sc,tfidf2v_X_cv_sc,y_tr,y_cv)
```

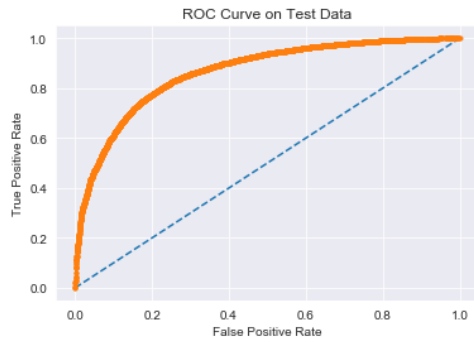
0.0001 for train data, auc score is: 0.8642888030292941 and for CV data, auc score is: 0.8672822683807803
 0.001 for train data, auc score is: 0.8635906902904539 and for CV data, auc score is: 0.8652122463521124
 0.01 for train data, auc score is: 0.8640050944059154 and for CV data, auc score is: 0.8659638070044453
 0.1 for train data, auc score is: 0.8643959705550021 and for CV data, auc score is: 0.8666425286006081
 1 for train data, auc score is: 0.8364282225595618 and for CV data, auc score is: 0.8387163279234457
 10 for train data, auc score is: 0.6917664121752087 and for CV data, auc score is: 0.6856552348330295
 100 for train data, auc score is: 0.636980675400652 and for CV data, auc score is: 0.6322705795299564
 1000 for train data, auc score is: 0.6369807785760822 and for CV data, auc score is: 0.6322706014298531

Best C Value 0.0001 with highest roc_auc Score is 0.8672822683807803



```
In [69]: testing(tfidf2v_X_train_sc,y_tr,tfidf2v_X_test_sc,y_test,optimal_alpha = 0.0001)
```

AUC Score 0.865313530362644



the maximum value of $tpr \cdot (1 - fpr)$ 0.6204479568895939 for threshold 0.825

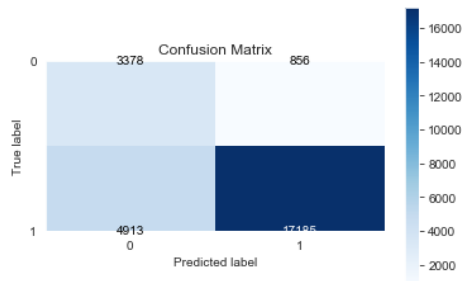
macro f1 score for data : 0.6978378219519505

micro f1 score for data: 0.7809129576181071

hamming loss for data: 0.21908704238189275

Precision recall report for data:

	precision	recall	f1-score	support
0	0.41	0.80	0.54	4234
1	0.95	0.78	0.86	22098
accuracy			0.78	26332
macro avg	0.68	0.79	0.70	26332
weighted avg	0.86	0.78	0.81	26332



```
In [70]: final_sample = final.sample(20000)
final_sample.shape
```

Out[70]: (20000, 11)

```
In [72]: X = final_sample['CleanedText']
y = final_sample['Score']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
X_tr, X_cv, y_tr, y_cv = train_test_split(X_train, y_train, test_size=0.3, random_state=0)

print("Size of X_train and y_train:", X_train.shape,y_train.shape)
print("Size of X_test and y_test:", X_test.shape,y_test.shape)
print("Size of X_tr and y_tr:", X_tr.shape,y_tr.shape)
print("Size of X_cv and y_cv:", X_cv.shape,y_cv.shape)
```

Size of X_train and y_train: (14000,) (14000,)

Size of X_test and y_test: (6000,) (6000,)

Size of X_tr and y_tr: (9800,) (9800,)

Size of X_cv and y_cv: (4200,) (4200,)

SVM for RBF Kernel:

```
In [83]: def SVM(X_train,X_cv,y_train,y_cv):
    pred_cv = []
    pred_train = []
    C = [10** x for x in range(-4, 4)]
    gamma = [10** x for x in range(-4,2)]
    for i in C:
        for j in tqdm(gamma):
            clf = SVC(kernel='rbf',probability=True,C=i,gamma=j)
            clf.fit(X_train,y_train)
            prob_train = clf.predict_proba(X_train)[:,-1]
            prob_cv = clf.predict_proba(X_cv)[:,-1]
            auc_score_train = roc_auc_score(y_train,prob_train)
            auc_score_cv = roc_auc_score(y_cv,prob_cv)
            pred_train.append(auc_score_train)
            pred_cv.append(auc_score_cv)

    cmap=sns.light_palette("green")
    # representing heat map for auc score
    print("-"*30, "AUC Score for training data", "-"*30)
    pred_train = np.array(pred_train)
    pred_train = pred_train.reshape(len(C),len(gamma))
    plt.figure(figsize=(10,5))
    sns.heatmap(pred_train,annot=True, cmap=cmap, fmt=".3f", xticklabels=gamma,yticklabels=C)
    plt.xlabel('gamma')
    plt.ylabel('C')
    plt.show()
    print("-"*30, "AUC Score for CV data", "-"*30)
    pred_cv = np.array(pred_cv)
    pred_cv = pred_cv.reshape(len(C),len(gamma))
    plt.figure(figsize=(10,5))
    sns.heatmap(pred_cv, annot=True, cmap=cmap, fmt=".3f", xticklabels=gamma, yticklabels=C)
    plt.xlabel('gamma')
    plt.ylabel('C')
    plt.show()
```

Model Testing

```
In [91]: import scikitplot.metrics as skplt
def testing_SVM(X_train,y_train,X_test,y_test,optimal_C,optimal_gamma):
    clf = SVC(C=optimal_C,kernel='rbf',probability=True,gamma=optimal_gamma)
    clf.fit(X_train,y_train)
    prob_test = clf.predict_proba(X_test)[:,-1]
    print("AUC Score",roc_auc_score(y_test,prob_test))
    # plot roc curve
    fpr, tpr, thresholds = roc_curve(y_test,prob_test)
    # plot no skill
    plt.plot([0, 1], [0, 1], linestyle='--')
    # plot the roc curve for the model
    plt.plot(fpr, tpr, marker='.')
    plt.title("Line Plot of ROC Curve on Test Data")
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    plt.show()

    def find_best_threshold(thresholds, fpr, tpr):
        t = threshold[np.argmax(tpr*(1-fpr))]
        print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshold", np.round(t,3))
        return t
    def predict_with_best_t(proba, threshold):
        predictions = []
        for i in proba:
            if i>=threshold:
                predictions.append(1)
            else:
                predictions.append(0)
        return predictions

    best_t = find_best_threshold(thresholds, fpr, tpr)
    predicted = predict_with_best_t(prob_test, best_t)
    skplt.plot_confusion_matrix(y_test,predicted)

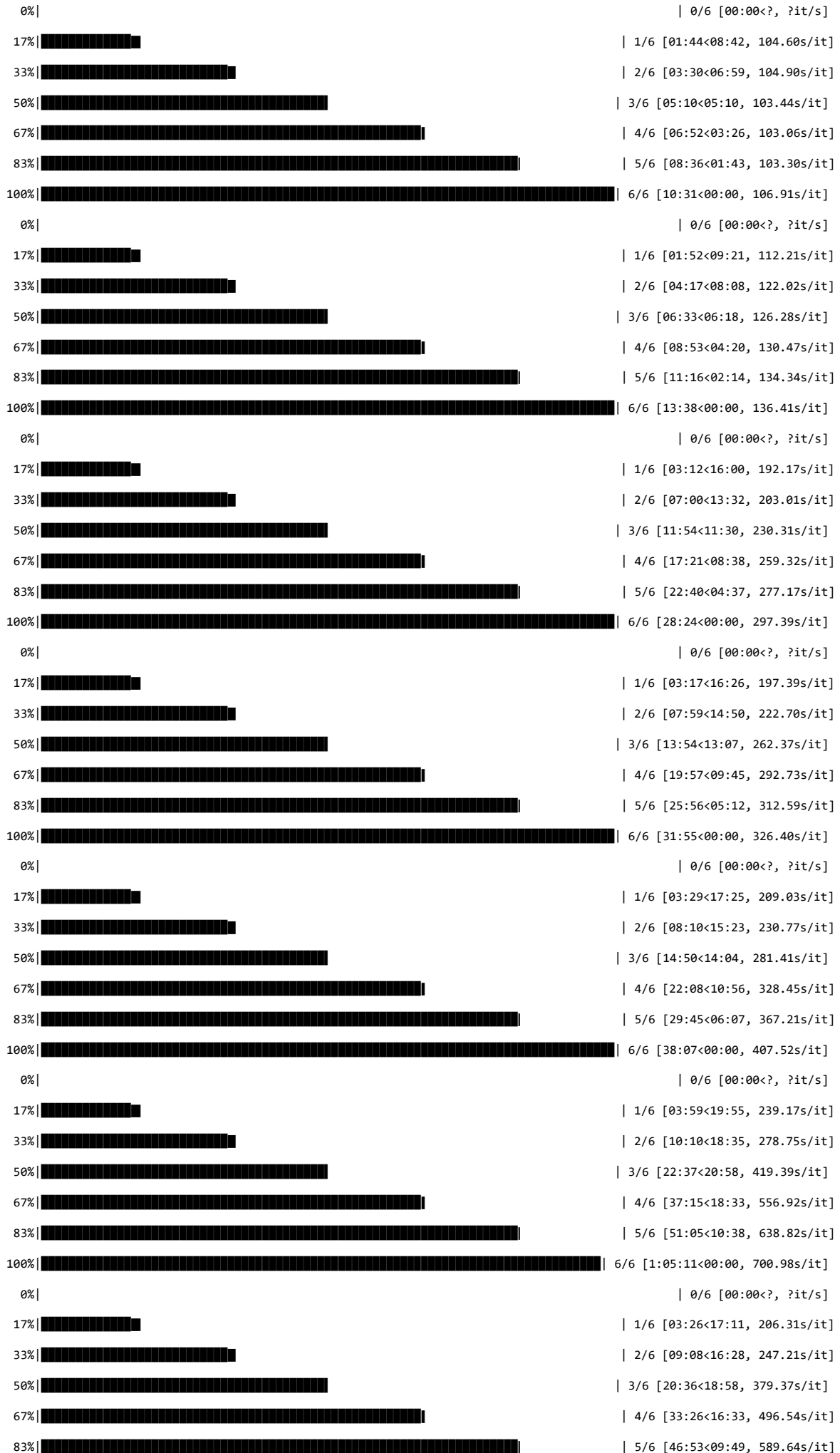
    print("macro f1 score for data :",f1_score(y_test, predicted, average = 'macro'))
    print("micro f1 scoore for data:",f1_score(y_test, predicted, average = 'micro'))
    print("hamming loss for data:",hamming_loss(y_test,predicted))
    print("Precision recall report for data:\n",classification_report(y_test, predicted))
```

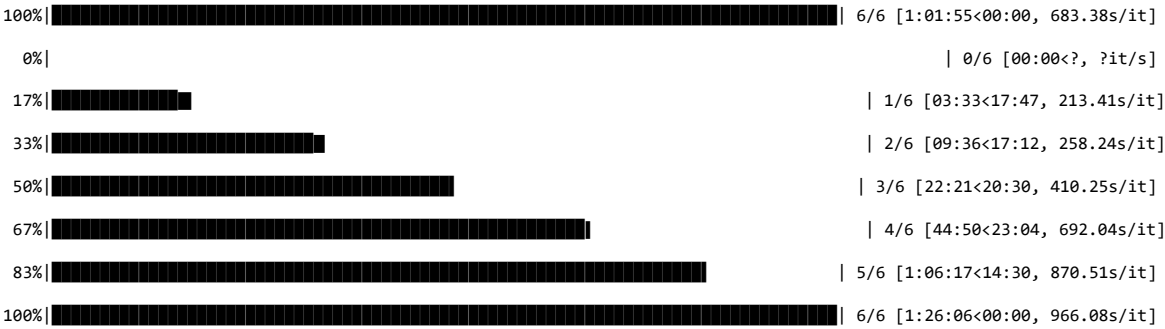
1. Bag of Words(BOW)

```
In [75]: count_vec = CountVectorizer()
BOW_X_train = count_vec.fit_transform(X_tr)
BOW_X_cv = count_vec.transform(X_cv)
BOW_X_test = count_vec.transform(X_test)
```

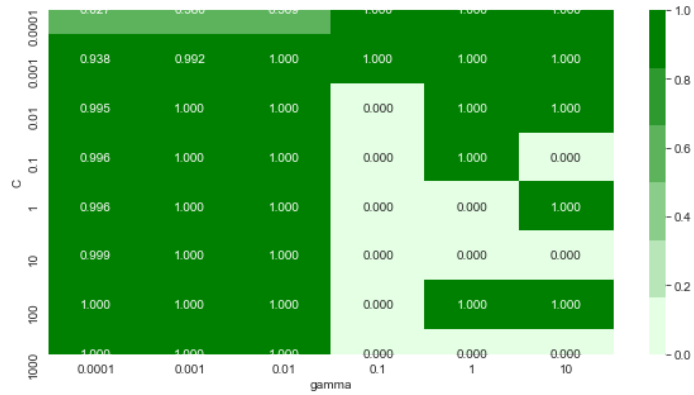
```
In [76]: sc = StandardScaler(with_mean=False)
BOW_X_train_sc = sc.fit_transform(BOW_X_train)
BOW_X_cv_sc = sc.transform(BOW_X_cv)
BOW_X_test_sc = sc.transform(BOW_X_test)
```

```
In [85]: SVM(BOW_X_train_sc,BOW_X_cv_sc,y_tr,y_cv)
```

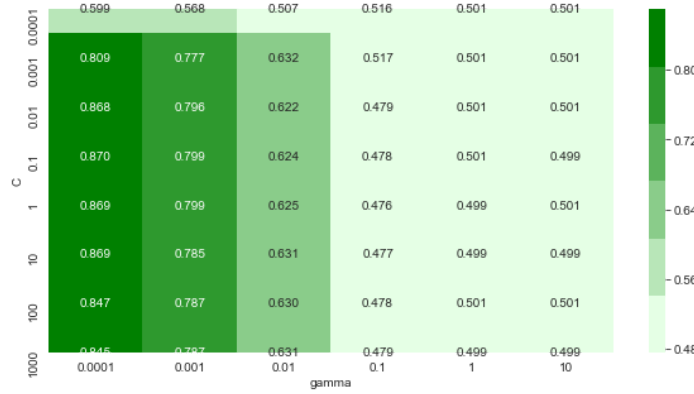





----- AUC Score for training data -----



----- AUC Score for CV data -----



```
In [92]: testing_SVM(BOW_X_train_sc,y_tr,BOW_X_test_sc,y_test,0.001,0.001)
```

AUC Score 0.7727964827687586



the maximum value of $tpr*(1-fpr)$ 0.5224915287042665 for threshold 0.791

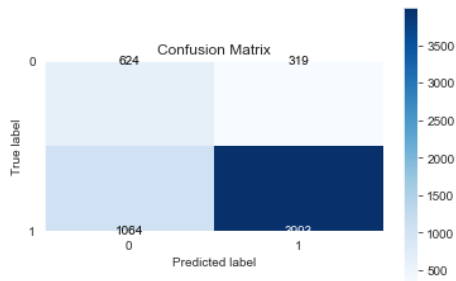
macro f1 score for data : 0.6633649412476893

micro f1 score for data: 0.7695

hamming loss for data: 0.2305

Precision recall report for data:

	precision	recall	f1-score	support
0	0.37	0.66	0.47	943
1	0.93	0.79	0.85	5057
accuracy			0.77	6000
macro avg	0.65	0.73	0.66	6000
weighted avg	0.84	0.77	0.79	6000



2. TF-IDF

```
In [94]: tfidf_vec = TfidfVectorizer(ngram_range=(1,2))
tfidf_train = tfidf_vec.fit_transform(X_tr)
tfidf_cv = tfidf_vec.transform(X_cv)
tfidf_test = tfidf_vec.transform(X_test)

print("The type of count vectorizer ",type(tfidf_train))
print("The shape of out text TFIDF vectorizer ",tfidf_train.get_shape())
print("Size of CV dataset:", tfidf_cv.shape)
print("Size of test dataset:", tfidf_test.shape)
```

The type of count vectorizer <class 'scipy.sparse.csr.csr_matrix'>

The shape of out text TFIDF vectorizer (9800, 226644)

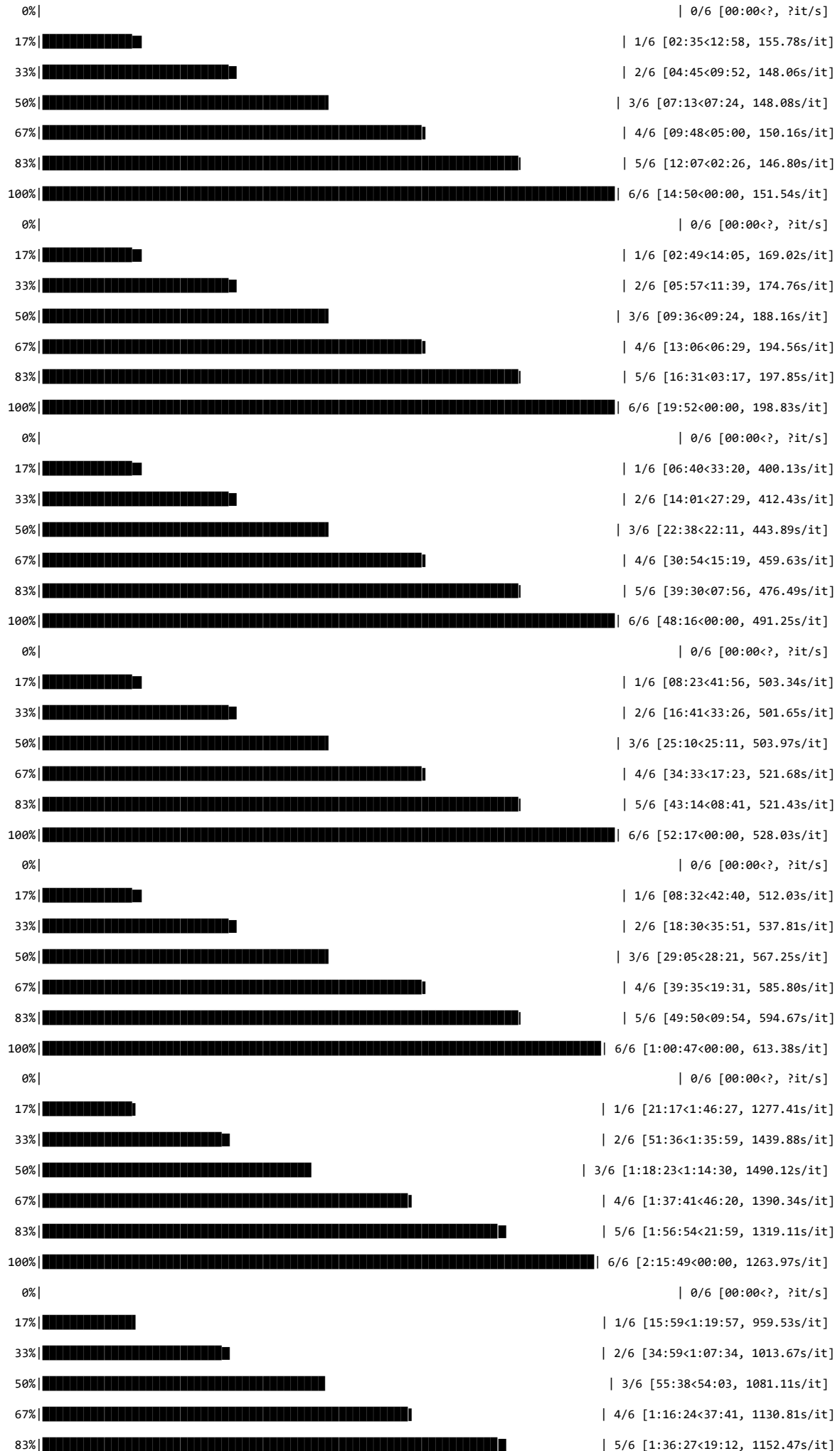
Size of CV dataset: (4200, 226644)

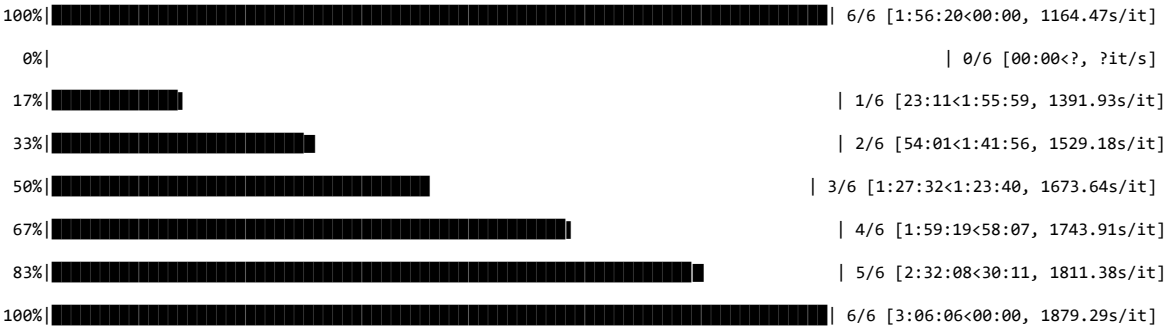
Size of test dataset: (6000, 226644)

```
In [95]: #Standardizing data using StandardScaler
```

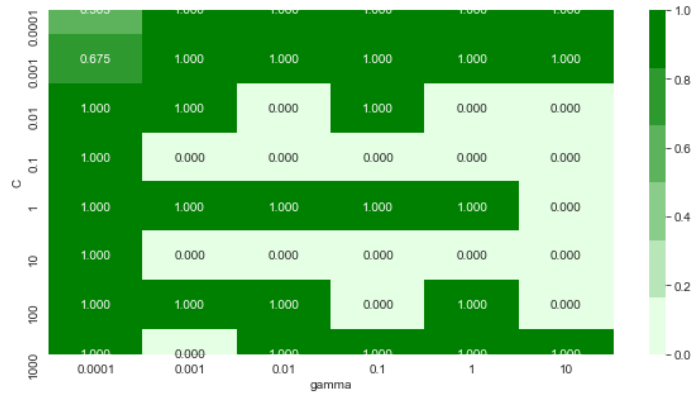
```
sc = StandardScaler(with_mean=False)
tfidf_train_sc = sc.fit_transform(tfidf_train)
tfidf_cv_sc = sc.transform(tfidf_cv)
tfidf_test_sc = sc.transform(tfidf_test)
```

```
In [96]: SVM(tfidf_train_sc,tfidf_cv_sc,y_tr,y_cv)
```

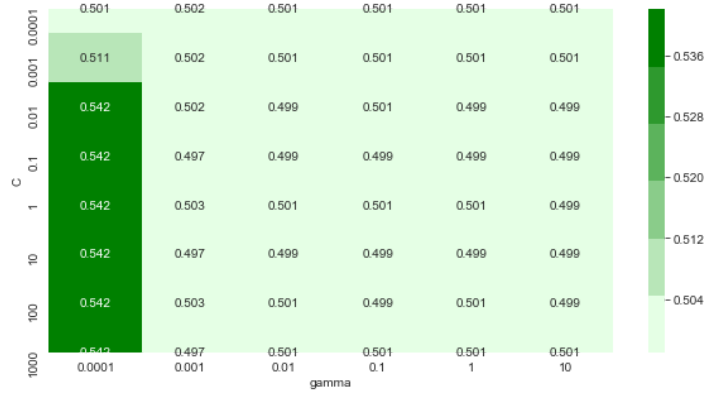




----- AUC Score for training data -----

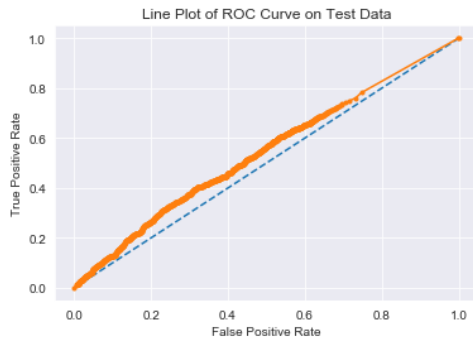


----- AUC Score for CV data -----



```
In [97]: testing_SVM(tfidf_train_sc,y_tr,tfidf_test_sc,y_test,0.001,0.0001)
```

AUC Score 0.5440874350537489



the maximum value of $tpr \cdot (1 - fpr)$ 0.2842274633336905 for threshold 0.07

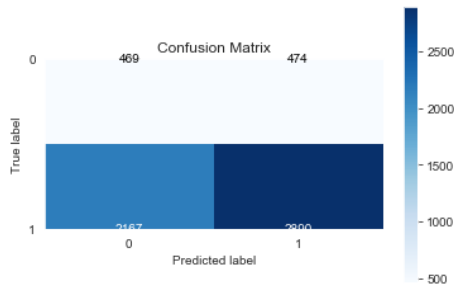
macro f1 score for data : 0.47423183549130205

micro f1 score for data: 0.5598333333333333

hamming loss for data: 0.44016666666666665

Precision recall report for data:

	precision	recall	f1-score	support
0	0.18	0.50	0.26	943
1	0.86	0.57	0.69	5057
accuracy			0.56	6000
macro avg	0.52	0.53	0.47	6000
weighted avg	0.75	0.56	0.62	6000



3. Avg W2V

```
In [98]: i=0
list_sent_train=[]
for sent in X_tr:
    filtered_sentence=[]
    sent=cleanhtml(sent)
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if(cleaned_words.isalpha()):
                filtered_sentence.append(cleaned_words.lower())
            else:
                continue
    list_sent_train.append(filtered_sentence)
```

```
In [99]: i=0
list_sent_cv=[]
for sent in X_cv:
    filtered_sentence=[]
    sent=cleanhtml(sent)
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if(cleaned_words.isalpha()):
                filtered_sentence.append(cleaned_words.lower())
            else:
                continue
    list_sent_cv.append(filtered_sentence)
```

```
In [100]: i=0
list_sent_test=[]
for sent in X_test:
    filtered_sentence=[]
    sent=cleanhtml(sent)
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if(cleaned_words.isalpha()):
                filtered_sentence.append(cleaned_words.lower())
            else:
                continue
    list_sent_test.append(filtered_sentence)
```

```
In [101]: w2v_model = gensim.models.Word2Vec(list_sent_train,min_count=5,size=50,workers=4)
          w2v_words = list(w2v_model.wv.vocab)
```

```
In [102]: train_avgw2v = avg_w2v(list_sent_train)
```

```
9800
50
```

```
In [103]: cv_avgw2v = avg_w2v(list_sent_cv)
```

```
4200
50
```

```
In [104]: test_avgw2v = avg_w2v(list_sent_test)
```

```
6000
50
```

```
In [105]: sc = StandardScaler(with_mean=False)
          aw2v_X_train_sc = sc.fit_transform(train_avgw2v)
          aw2v_X_cv_sc = sc.transform(cv_avgw2v)
          aw2v_X_test_sc = sc.transform(test_avgw2v)
```

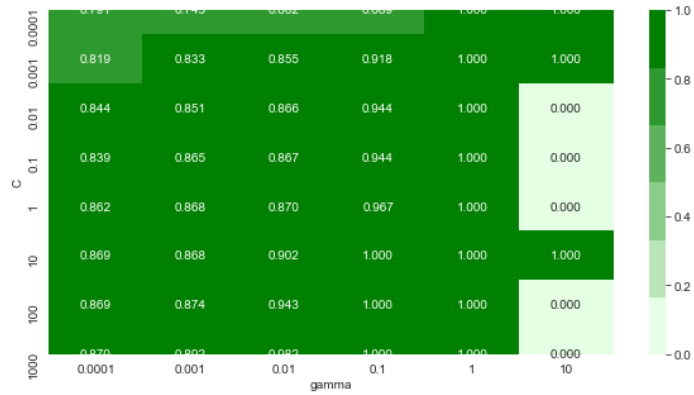


```
In [106]: SVM(aw2v_X_train_sc,aw2v_X_cv_sc,y_tr,y_cv)
```

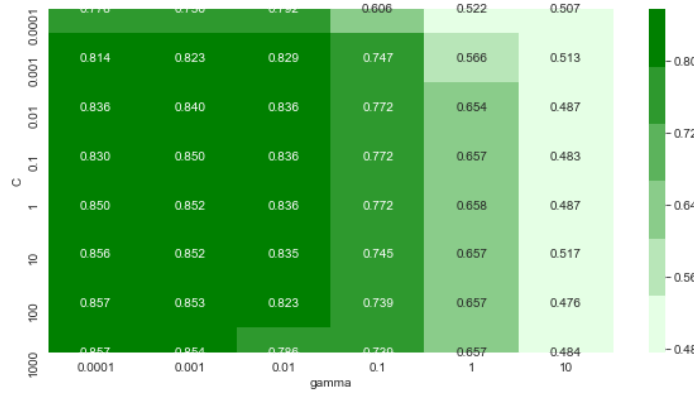




----- AUC Score for training data -----



----- AUC Score for CV data -----



```
In [107]: testing_SVM(aw2v_X_train_sc,y_tr,aw2v_X_test_sc,y_test,0.001,0.001)
```

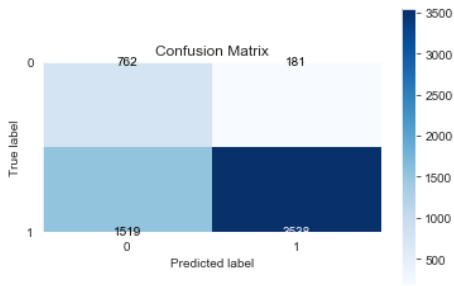
AUC Score 0.8202554505362096



the maximum value of $tpr \cdot (1 - fpr)$ 0.5653379679500985 for threshold 0.952
macro f1 score for data : 0.6394972980675924
micro f1 score for data: 0.7166666666666667
hamming loss for data: 0.2833333333333333

Precision recall report for data:

	precision	recall	f1-score	support
0	0.33	0.81	0.47	943
1	0.95	0.70	0.81	5057
accuracy			0.72	6000
macro avg	0.64	0.75	0.64	6000
weighted avg	0.85	0.72	0.75	6000

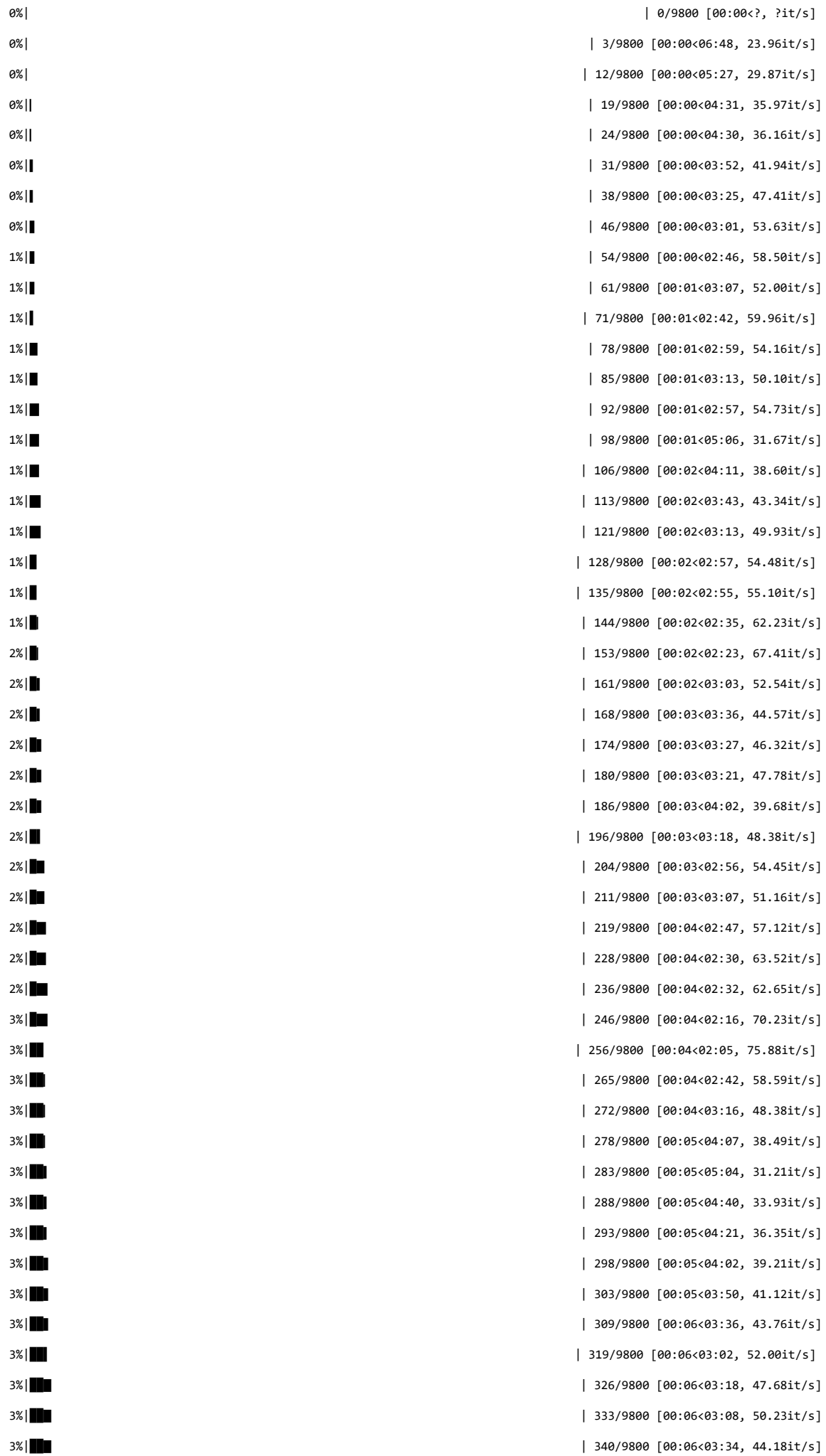


















































4. TD-IDF W2V

















































```
In [108]: tf_idf_vect = TfidfVectorizer()
tfidf_train = tf_idf_vect.fit_transform(X_tr)
print("The type of count vectorizer ",type(tfidf_train))
print("The shape of out text TFIDF vectorizer ",tfidf_train.get_shape())
tfidf_cv = tf_idf_vect.transform(X_cv)
tfidf_test = tf_idf_vect.transform(X_test)
print("CV Data Size: ",tfidf_cv.shape)
print("Test Data Size: ",tfidf_test.shape)
```

The type of count vectorizer <class 'scipy.sparse.csr.csr_matrix'>
The shape of out text TFIDF vectorizer (9800, 12498)
CV Data Size: (4200, 12498)
Test Data Size: (6000, 12498)

















































```
In [109]: t = tf_idf_vect.get_feature_names()
tfidf_sent_vectors_train = [] # the tfidf-w2v for each sentence/review is stored in this list
row=0
for sent in tqdm(list_sent_train):
    sent_vec = np.zeros(50)
    cnt_words = 0; # num of words with a valid vector in the sentence/review
    for word in sent: # for each word in a review/sentence
        if word in w2v_words:
            vec = w2v_model.wv[word]
            tfidf = tfidf_train[row,t.index(word)]
            sent_vec += (vec * tfidf)
            cnt_words += tfidf
    if cnt_words != 0:
        sent_vec /= cnt_words
    tfidf_sent_vectors_train.append(sent_vec)
    row += 1
print(len(tfidf_sent_vectors_train))
print(len(tfidf_sent_vectors_train[0]))
```
















































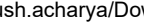



















































4%		345/9800 [00:06<03:36, 43.73it/s]
4%		351/9800 [00:06<03:22, 46.76it/s]
4%		357/9800 [00:06<03:10, 49.54it/s]
4%		363/9800 [00:07<03:17, 47.76it/s]
4%		370/9800 [00:07<03:06, 50.68it/s]
4%		377/9800 [00:07<03:04, 51.11it/s]
4%		383/9800 [00:07<03:15, 48.26it/s]
4%		388/9800 [00:07<03:22, 46.53it/s]
4%		394/9800 [00:07<03:09, 49.55it/s]
4%		400/9800 [00:07<03:23, 46.30it/s]
4%		405/9800 [00:07<03:41, 42.34it/s]
4%		410/9800 [00:08<03:41, 42.37it/s]
4%		416/9800 [00:08<03:26, 45.51it/s]
4%		422/9800 [00:08<03:41, 42.37it/s]
4%		430/9800 [00:08<03:17, 47.44it/s]
4%		436/9800 [00:08<03:14, 48.16it/s]
5%		442/9800 [00:08<03:09, 49.42it/s]
5%		449/9800 [00:08<02:53, 54.03it/s]
5%		459/9800 [00:08<02:29, 62.59it/s]
5%		466/9800 [00:09<02:33, 60.77it/s]
5%		473/9800 [00:09<03:17, 47.18it/s]
5%		480/9800 [00:09<02:59, 51.96it/s]
5%		486/9800 [00:09<02:58, 52.17it/s]
5%		492/9800 [00:09<03:19, 46.61it/s]
5%		499/9800 [00:09<03:01, 51.27it/s]
5%		506/9800 [00:10<04:36, 33.65it/s]
5%		511/9800 [00:10<04:33, 34.00it/s]
5%		516/9800 [00:10<04:07, 37.45it/s]
5%		522/9800 [00:10<03:41, 41.96it/s]
5%		527/9800 [00:10<03:31, 43.89it/s]
5%		532/9800 [00:10<03:25, 45.16it/s]
5%		537/9800 [00:10<03:45, 41.14it/s]
6%		542/9800 [00:11<04:19, 35.64it/s]
6%		546/9800 [00:11<05:07, 30.13it/s]
6%		551/9800 [00:11<04:45, 32.38it/s]
6%		555/9800 [00:11<04:44, 32.48it/s]
6%		563/9800 [00:11<03:54, 39.42it/s]
6%		569/9800 [00:11<03:43, 41.37it/s]
6%		574/9800 [00:11<03:32, 43.48it/s]
6%		580/9800 [00:11<03:15, 47.15it/s]
6%		586/9800 [00:12<03:21, 45.72it/s]
6%		592/9800 [00:12<03:30, 43.70it/s]
6%		599/9800 [00:12<03:10, 48.37it/s]
6%		606/9800 [00:12<02:58, 51.46it/s]
6%		612/9800 [00:12<02:54, 52.79it/s]
6%		621/9800 [00:12<02:33, 59.82it/s]
6%		630/9800 [00:12<02:22, 64.24it/s]
6%		637/9800 [00:12<02:38, 57.71it/s]
7%		644/9800 [00:12<02:36, 58.46it/s]

7%		651/9800 [00:13<02:32, 59.95it/s]
7%		659/9800 [00:13<02:25, 62.87it/s]
7%		666/9800 [00:13<03:02, 50.17it/s]
7%		673/9800 [00:13<02:53, 52.75it/s]
7%		680/9800 [00:13<02:41, 56.51it/s]
7%		687/9800 [00:13<03:48, 39.91it/s]
7%		692/9800 [00:14<04:30, 33.69it/s]
7%		698/9800 [00:14<04:00, 37.89it/s]
7%		705/9800 [00:14<03:32, 42.77it/s]
7%		711/9800 [00:14<03:20, 45.41it/s]
7%		717/9800 [00:14<03:10, 47.69it/s]
7%		724/9800 [00:14<02:52, 52.65it/s]
7%		734/9800 [00:14<02:31, 59.91it/s]
8%		741/9800 [00:14<02:48, 53.74it/s]
8%		747/9800 [00:15<03:27, 43.55it/s]
8%		754/9800 [00:15<03:07, 48.18it/s]
8%		760/9800 [00:15<03:59, 37.77it/s]
8%		766/9800 [00:15<03:56, 38.17it/s]
8%		771/9800 [00:15<04:31, 33.22it/s]
8%		777/9800 [00:15<03:55, 38.25it/s]
8%		782/9800 [00:16<04:18, 34.90it/s]
8%		787/9800 [00:16<04:03, 37.03it/s]
8%		794/9800 [00:16<03:33, 42.19it/s]
8%		802/9800 [00:16<03:44, 40.14it/s]
8%		807/9800 [00:16<03:48, 39.40it/s]
8%		815/9800 [00:16<03:18, 45.38it/s]
8%		821/9800 [00:16<03:09, 47.42it/s]
8%		830/9800 [00:17<02:59, 49.88it/s]
9%		836/9800 [00:17<02:59, 50.04it/s]
9%		842/9800 [00:17<03:39, 40.85it/s]
9%		847/9800 [00:17<03:31, 42.43it/s]
9%		854/9800 [00:17<03:08, 47.46it/s]
9%		862/9800 [00:17<02:46, 53.79it/s]
9%		868/9800 [00:17<02:51, 52.13it/s]
9%		876/9800 [00:17<02:37, 56.74it/s]
9%		884/9800 [00:18<02:27, 60.56it/s]
9%		891/9800 [00:18<02:22, 62.35it/s]
9%		898/9800 [00:18<02:26, 60.85it/s]
9%		905/9800 [00:18<02:44, 54.12it/s]
9%		911/9800 [00:18<02:41, 55.15it/s]
9%		920/9800 [00:18<02:28, 59.95it/s]
9%		929/9800 [00:18<02:23, 61.76it/s]
10%		936/9800 [00:18<02:38, 55.89it/s]
10%		942/9800 [00:19<02:50, 51.97it/s]
10%		948/9800 [00:19<03:14, 45.59it/s]
10%		958/9800 [00:19<02:45, 53.41it/s]
10%		965/9800 [00:19<02:43, 54.08it/s]
10%		973/9800 [00:19<02:32, 57.94it/s]











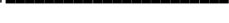





































10%	<div><div></div></div>	981/9800 [00:19<02:20, 62.61it/s]
10%	<div><div></div></div>	988/9800 [00:19<02:17, 63.86it/s]
10%	<div><div></div></div>	995/9800 [00:19<02:23, 61.46it/s]
10%	<div><div></div></div>	1005/9800 [00:20<02:19, 63.19it/s]
10%	<div><div></div></div>	1012/9800 [00:20<02:24, 60.77it/s]
10%	<div><div></div></div>	1022/9800 [00:20<02:07, 68.80it/s]
11%	<div><div></div></div>	1031/9800 [00:20<02:11, 66.88it/s]
11%	<div><div></div></div>	1041/9800 [00:20<01:58, 74.10it/s]
11%	<div><div></div></div>	1049/9800 [00:20<02:00, 72.36it/s]
11%	<div><div></div></div>	1057/9800 [00:20<02:01, 72.19it/s]
11%	<div><div></div></div>	1065/9800 [00:20<02:05, 69.35it/s]
11%	<div><div></div></div>	1073/9800 [00:21<02:10, 67.09it/s]
11%	<div><div></div></div>	1080/9800 [00:21<02:31, 57.46it/s]
11%	<div><div></div></div>	1088/9800 [00:21<02:23, 60.79it/s]
11%	<div><div></div></div>	1096/9800 [00:21<02:18, 62.90it/s]
11%	<div><div></div></div>	1103/9800 [00:21<02:39, 54.38it/s]
11%	<div><div></div></div>	1112/9800 [00:21<02:21, 61.29it/s]
11%	<div><div></div></div>	1119/9800 [00:21<02:18, 62.49it/s]
11%	<div><div></div></div>	1126/9800 [00:21<02:20, 61.52it/s]
12%	<div><div></div></div>	1134/9800 [00:22<02:11, 65.73it/s]
12%	<div><div></div></div>	1141/9800 [00:22<02:16, 63.50it/s]
12%	<div><div></div></div>	1148/9800 [00:22<02:13, 64.68it/s]
12%	<div><div></div></div>	1158/9800 [00:22<02:01, 71.28it/s]
12%	<div><div></div></div>	1166/9800 [00:22<02:15, 63.80it/s]
12%	<div><div></div></div>	1173/9800 [00:22<02:32, 56.41it/s]
12%	<div><div></div></div>	1180/9800 [00:22<02:26, 58.75it/s]
12%	<div><div></div></div>	1187/9800 [00:22<02:27, 58.54it/s]
12%	<div><div></div></div>	1194/9800 [00:23<02:31, 56.95it/s]
12%	<div><div></div></div>	1200/9800 [00:23<03:28, 41.18it/s]
12%	<div><div></div></div>	1208/9800 [00:23<03:03, 46.85it/s]
12%	<div><div></div></div>	1218/9800 [00:23<02:34, 55.68it/s]
13%	<div><div></div></div>	1226/9800 [00:23<02:27, 58.03it/s]
13%	<div><div></div></div>	1236/9800 [00:23<02:09, 66.30it/s]
13%	<div><div></div></div>	1246/9800 [00:23<01:59, 71.35it/s]
13%	<div><div></div></div>	1254/9800 [00:23<01:59, 71.28it/s]
13%	<div><div></div></div>	1262/9800 [00:24<02:26, 58.18it/s]
13%	<div><div></div></div>	1269/9800 [00:24<02:46, 51.34it/s]
13%	<div><div></div></div>	1275/9800 [00:24<02:55, 48.68it/s]
13%	<div><div></div></div>	1281/9800 [00:24<03:06, 45.74it/s]
13%	<div><div></div></div>	1290/9800 [00:24<02:41, 52.69it/s]
13%	<div><div></div></div>	1296/9800 [00:24<02:40, 53.00it/s]
13%	<div><div></div></div>	1302/9800 [00:25<02:50, 49.71it/s]
13%	<div><div></div></div>	1308/9800 [00:25<02:59, 47.35it/s]
13%	<div><div></div></div>	1315/9800 [00:25<02:43, 52.01it/s]
14%	<div><div></div></div>	1323/9800 [00:25<02:28, 57.04it/s]
14%	<div><div></div></div>	1330/9800 [00:25<03:33, 39.63it/s]
14%	<div><div></div></div>	1335/9800 [00:25<03:33, 39.72it/s]
14%	<div><div></div></div>	1340/9800 [00:25<04:05, 34.47it/s]
14%	<div><div></div></div>	1345/9800 [00:26<03:55, 35.88it/s]

14%		1350/9800 [00:26<03:38, 38.75it/s]
14%		1357/9800 [00:26<03:10, 44.27it/s]
14%		1368/9800 [00:26<02:37, 53.54it/s]
14%		1375/9800 [00:26<02:41, 52.06it/s]
14%		1382/9800 [00:26<02:42, 51.83it/s]
14%		1388/9800 [00:26<02:35, 54.03it/s]
14%		1394/9800 [00:26<02:31, 55.56it/s]
14%		1400/9800 [00:27<03:47, 36.95it/s]
14%		1405/9800 [00:27<03:40, 38.06it/s]
14%		1415/9800 [00:27<03:01, 46.24it/s]
14%		1421/9800 [00:27<02:52, 48.66it/s]
15%		1427/9800 [00:27<02:47, 49.85it/s]
15%		1433/9800 [00:27<02:52, 48.56it/s]
15%		1439/9800 [00:27<02:44, 50.90it/s]
15%		1448/9800 [00:27<02:24, 57.96it/s]
15%		1457/9800 [00:28<02:08, 64.82it/s]
15%		1465/9800 [00:28<02:28, 56.09it/s]
15%		1472/9800 [00:28<02:46, 49.96it/s]
15%		1478/9800 [00:28<02:45, 50.21it/s]
15%		1484/9800 [00:28<02:48, 49.40it/s]
15%		1491/9800 [00:28<03:10, 43.73it/s]
15%		1503/9800 [00:28<02:33, 53.96it/s]
15%		1512/9800 [00:29<02:18, 59.87it/s]
16%		1522/9800 [00:29<02:02, 67.47it/s]
16%		1530/9800 [00:29<02:12, 62.32it/s]
16%		1539/9800 [00:29<02:02, 67.42it/s]
16%		1547/9800 [00:29<02:23, 57.33it/s]
16%		1558/9800 [00:29<02:07, 64.75it/s]
16%		1566/9800 [00:29<02:07, 64.73it/s]
16%		1574/9800 [00:30<02:08, 63.78it/s]
16%		1583/9800 [00:30<01:59, 68.81it/s]
16%		1592/9800 [00:30<01:53, 72.23it/s]
16%		1600/9800 [00:30<02:21, 57.95it/s]
16%		1607/9800 [00:30<02:39, 51.49it/s]
16%		1613/9800 [00:30<02:48, 48.48it/s]
17%		1620/9800 [00:30<02:34, 53.00it/s]
17%		1626/9800 [00:31<02:48, 48.65it/s]
17%		1632/9800 [00:31<02:45, 49.47it/s]
17%		1639/9800 [00:31<02:51, 47.57it/s]
17%		1644/9800 [00:31<03:00, 45.13it/s]
17%		1649/9800 [00:31<04:08, 32.85it/s]
17%		1656/9800 [00:31<03:29, 38.83it/s]
17%		1662/9800 [00:31<03:08, 43.28it/s]
17%		1668/9800 [00:31<03:02, 44.57it/s]
17%		1674/9800 [00:32<03:03, 44.20it/s]
17%		1679/9800 [00:32<03:21, 40.25it/s]
17%		1686/9800 [00:32<03:01, 44.79it/s]
17%		1693/9800 [00:32<02:46, 48.62it/s]


17%		1700/9800 [00:32<02:38, 51.22it/s]
17%		1706/9800 [00:32<02:35, 52.13it/s]
18%		1715/9800 [00:32<02:15, 59.62it/s]
18%		1722/9800 [00:32<02:10, 62.12it/s]
18%		1729/9800 [00:33<02:23, 56.19it/s]
18%		1736/9800 [00:33<02:30, 53.47it/s]
18%		1742/9800 [00:33<02:42, 49.71it/s]
18%		1751/9800 [00:33<02:21, 56.88it/s]
18%		1758/9800 [00:33<02:37, 51.19it/s]
18%		1766/9800 [00:33<02:22, 56.19it/s]
18%		1773/9800 [00:33<02:40, 50.01it/s]
18%		1779/9800 [00:34<02:52, 46.50it/s]
18%		1785/9800 [00:34<02:53, 46.18it/s]
18%		1797/9800 [00:34<02:21, 56.63it/s]
18%		1805/9800 [00:34<02:17, 58.12it/s]
19%		1814/9800 [00:34<02:04, 64.38it/s]
19%		1823/9800 [00:34<02:00, 66.05it/s]
19%		1831/9800 [00:34<02:07, 62.49it/s]
19%		1841/9800 [00:34<01:54, 69.40it/s]
19%		1849/9800 [00:35<02:10, 60.99it/s]
19%		1858/9800 [00:35<02:04, 63.78it/s]
19%		1865/9800 [00:35<02:06, 62.63it/s]
19%		1872/9800 [00:35<02:33, 51.59it/s]
19%		1879/9800 [00:35<02:21, 55.88it/s]
19%		1887/9800 [00:35<02:11, 60.34it/s]
19%		1894/9800 [00:35<02:36, 50.63it/s]
19%		1901/9800 [00:36<02:26, 53.76it/s]
19%		1908/9800 [00:36<02:20, 56.32it/s]
20%		1914/9800 [00:36<03:00, 43.75it/s]
20%		1922/9800 [00:36<02:36, 50.26it/s]
20%		1929/9800 [00:36<02:24, 54.63it/s]
20%		1936/9800 [00:36<02:42, 48.33it/s]
20%		1946/9800 [00:36<02:18, 56.54it/s]
20%		1959/9800 [00:36<01:56, 67.32it/s]
20%		1968/9800 [00:37<02:30, 51.88it/s]
20%		1977/9800 [00:37<02:11, 59.30it/s]
20%		1985/9800 [00:37<02:11, 59.37it/s]
20%		1994/9800 [00:37<01:59, 65.45it/s]
20%		2002/9800 [00:37<02:02, 63.50it/s]
21%		2013/9800 [00:37<01:47, 72.16it/s]
21%		2023/9800 [00:37<01:40, 77.30it/s]
21%		2032/9800 [00:38<01:41, 76.69it/s]
21%		2041/9800 [00:38<01:43, 74.84it/s]
21%		2049/9800 [00:38<01:44, 73.91it/s]
21%		2057/9800 [00:38<01:50, 70.08it/s]
21%		2065/9800 [00:38<01:48, 71.58it/s]
21%		2073/9800 [00:38<01:44, 73.67it/s]
21%		2081/9800 [00:38<02:06, 61.00it/s]
21%		2089/9800 [00:38<02:00, 63.85it/s]

21%		2097/9800 [00:39<01:55, 66.97it/s]
21%		2106/9800 [00:39<01:47, 71.86it/s]
22%		2114/9800 [00:39<01:44, 73.75it/s]
22%		2122/9800 [00:39<01:50, 69.39it/s]
22%		2130/9800 [00:39<01:51, 68.76it/s]
22%		2138/9800 [00:39<01:56, 65.93it/s]
22%		2145/9800 [00:39<02:32, 50.32it/s]
22%		2154/9800 [00:39<02:13, 57.22it/s]
22%		2161/9800 [00:40<02:07, 59.69it/s]
22%		2171/9800 [00:40<01:54, 66.83it/s]
22%		2179/9800 [00:40<01:50, 69.10it/s]
22%		2187/9800 [00:40<01:49, 69.41it/s]
22%		2195/9800 [00:40<01:48, 69.89it/s]
22%		2203/9800 [00:40<02:01, 62.67it/s]
23%		2210/9800 [00:40<02:29, 50.62it/s]
23%		2218/9800 [00:40<02:23, 52.68it/s]
23%		2230/9800 [00:41<02:02, 61.84it/s]
23%		2238/9800 [00:41<02:04, 60.67it/s]
23%		2246/9800 [00:41<02:00, 62.90it/s]
23%		2253/9800 [00:41<02:18, 54.45it/s]
23%		2259/9800 [00:41<03:11, 39.39it/s]
23%		2269/9800 [00:41<02:38, 47.41it/s]
23%		2276/9800 [00:41<02:27, 50.97it/s]
23%		2283/9800 [00:42<02:21, 53.18it/s]
23%		2293/9800 [00:42<02:04, 60.13it/s]
23%		2300/9800 [00:42<02:12, 56.51it/s]
24%		2307/9800 [00:42<02:17, 54.66it/s]
24%		2313/9800 [00:42<02:19, 53.74it/s]
24%		2321/9800 [00:42<02:11, 56.92it/s]
24%		2329/9800 [00:42<02:01, 61.66it/s]
24%		2336/9800 [00:42<02:11, 56.85it/s]
24%		2343/9800 [00:43<02:04, 59.71it/s]
24%		2350/9800 [00:43<01:59, 62.45it/s]
24%		2360/9800 [00:43<01:48, 68.71it/s]
24%		2368/9800 [00:43<01:47, 69.40it/s]
24%		2376/9800 [00:43<02:10, 56.68it/s]
24%		2383/9800 [00:43<02:09, 57.08it/s]
24%		2392/9800 [00:43<01:56, 63.59it/s]
24%		2399/9800 [00:44<02:21, 52.45it/s]
25%		2407/9800 [00:44<02:08, 57.71it/s]
25%		2415/9800 [00:44<01:59, 62.02it/s]
25%		2426/9800 [00:44<01:44, 70.88it/s]
25%		2434/9800 [00:44<02:15, 54.29it/s]
25%		2441/9800 [00:44<02:07, 57.89it/s]
25%		2449/9800 [00:44<02:39, 46.02it/s]
25%		2457/9800 [00:45<02:20, 52.31it/s]
25%		2466/9800 [00:45<02:03, 59.56it/s]
25%		2473/9800 [00:45<02:13, 54.82it/s]

25%		2481/9800 [00:45<02:01, 60.19it/s]
25%		2488/9800 [00:45<02:18, 52.94it/s]
25%		2495/9800 [00:45<02:09, 56.22it/s]
26%		2502/9800 [00:45<02:05, 57.93it/s]
26%		2509/9800 [00:45<02:01, 59.87it/s]
26%		2517/9800 [00:45<01:55, 63.03it/s]
26%		2524/9800 [00:46<02:12, 54.90it/s]
26%		2530/9800 [00:46<02:11, 55.44it/s]
26%		2537/9800 [00:46<02:14, 54.20it/s]
26%		2543/9800 [00:46<02:27, 49.10it/s]
26%		2549/9800 [00:46<02:42, 44.56it/s]
26%		2559/9800 [00:46<02:16, 53.12it/s]
26%		2567/9800 [00:46<02:03, 58.52it/s]
26%		2577/9800 [00:47<01:48, 66.35it/s]
26%		2587/9800 [00:47<01:38, 73.59it/s]
26%		2596/9800 [00:47<01:44, 69.05it/s]
27%		2604/9800 [00:47<02:00, 59.78it/s]
27%		2613/9800 [00:47<01:51, 64.68it/s]
27%		2621/9800 [00:47<01:56, 61.87it/s]
27%		2628/9800 [00:47<01:52, 64.02it/s]
27%		2636/9800 [00:47<01:45, 67.89it/s]
27%		2644/9800 [00:48<01:43, 69.40it/s]
27%		2652/9800 [00:48<02:11, 54.48it/s]
27%		2660/9800 [00:48<02:02, 58.19it/s]
27%		2669/9800 [00:48<01:53, 62.74it/s]
27%		2676/9800 [00:48<01:59, 59.79it/s]
27%		2683/9800 [00:48<02:05, 56.91it/s]
27%		2689/9800 [00:48<02:07, 55.84it/s]
28%		2697/9800 [00:48<01:56, 60.80it/s]
28%		2704/9800 [00:49<02:08, 55.23it/s]
28%		2711/9800 [00:49<02:04, 56.73it/s]
28%		2718/9800 [00:49<01:59, 59.32it/s]
28%		2725/9800 [00:49<01:56, 60.58it/s]
28%		2733/9800 [00:49<01:57, 59.93it/s]
28%		2740/9800 [00:49<01:55, 60.95it/s]
28%		2749/9800 [00:49<01:45, 66.76it/s]
28%		2756/9800 [00:49<01:46, 66.10it/s]
28%		2763/9800 [00:50<01:58, 59.17it/s]
28%		2770/9800 [00:50<02:10, 53.99it/s]
28%		2777/9800 [00:50<02:02, 57.38it/s]
28%		2783/9800 [00:50<02:15, 51.76it/s]
28%		2789/9800 [00:50<02:13, 52.50it/s]
29%		2795/9800 [00:50<02:22, 49.30it/s]
29%		2803/9800 [00:50<02:08, 54.51it/s]
29%		2810/9800 [00:50<01:59, 58.38it/s]
29%		2817/9800 [00:51<02:17, 50.82it/s]
29%		2823/9800 [00:51<02:11, 52.89it/s]
29%		2829/9800 [00:51<02:29, 46.65it/s]
29%		2836/9800 [00:51<02:16, 50.91it/s]

29%		2843/9800 [00:51<02:05, 55.43it/s]
29%		2849/9800 [00:51<02:03, 56.22it/s]
29%		2857/9800 [00:51<01:53, 60.94it/s]
29%		2864/9800 [00:51<02:01, 57.31it/s]
29%		2872/9800 [00:52<01:51, 62.00it/s]
29%		2880/9800 [00:52<01:49, 63.45it/s]
29%		2887/9800 [00:52<02:09, 53.29it/s]
30%		2895/9800 [00:52<01:57, 58.57it/s]
30%		2902/9800 [00:52<02:10, 53.01it/s]
30%		2908/9800 [00:52<02:08, 53.70it/s]
30%		2916/9800 [00:52<01:56, 59.26it/s]
30%		2923/9800 [00:52<02:19, 49.42it/s]
30%		2929/9800 [00:53<03:10, 35.98it/s]
30%		2934/9800 [00:53<03:10, 36.12it/s]
30%		2944/9800 [00:53<02:36, 43.71it/s]
30%		2950/9800 [00:53<02:41, 42.31it/s]
30%		2956/9800 [00:53<02:53, 39.39it/s]
30%		2970/9800 [00:53<02:16, 49.87it/s]
30%		2980/9800 [00:54<02:04, 54.66it/s]
30%		2988/9800 [00:54<01:54, 59.73it/s]
31%		2998/9800 [00:54<01:40, 67.93it/s]
31%		3007/9800 [00:54<01:42, 66.08it/s]
31%		3015/9800 [00:54<01:41, 67.13it/s]
31%		3023/9800 [00:54<01:46, 63.82it/s]
31%		3032/9800 [00:54<01:38, 68.62it/s]
31%		3040/9800 [00:54<01:40, 67.21it/s]
31%		3048/9800 [00:55<01:45, 64.30it/s]
31%		3055/9800 [00:55<02:25, 46.24it/s]
31%		3063/9800 [00:55<02:08, 52.43it/s]
31%		3070/9800 [00:55<02:00, 55.64it/s]
31%		3077/9800 [00:55<02:03, 54.64it/s]
31%		3083/9800 [00:55<02:01, 55.08it/s]
32%		3089/9800 [00:55<01:59, 56.27it/s]
32%		3097/9800 [00:55<01:48, 61.52it/s]
32%		3104/9800 [00:56<01:50, 60.64it/s]
32%		3112/9800 [00:56<01:45, 63.69it/s]
32%		3119/9800 [00:56<01:57, 56.70it/s]
32%		3126/9800 [00:56<01:52, 59.48it/s]
32%		3138/9800 [00:56<01:35, 69.61it/s]
32%		3146/9800 [00:56<01:51, 59.65it/s]
32%		3155/9800 [00:56<01:40, 66.05it/s]
32%		3163/9800 [00:56<01:38, 67.27it/s]
32%		3171/9800 [00:57<01:39, 66.92it/s]
32%		3179/9800 [00:57<01:35, 69.05it/s]
33%		3187/9800 [00:57<01:35, 69.24it/s]
33%		3196/9800 [00:57<01:31, 72.07it/s]
33%		3205/9800 [00:57<01:45, 62.59it/s]
33%		3212/9800 [00:57<01:48, 60.89it/s]

file:///C:/Users/pratyush.acharya/Downloads/07 SVM on Amazon food reviews data.html

37%		3580/9800 [01:04<01:50, 56.48it/s]
37%		3587/9800 [01:04<01:47, 57.94it/s]
37%		3594/9800 [01:04<01:41, 61.02it/s]
37%		3605/9800 [01:04<01:31, 67.38it/s]
37%		3613/9800 [01:04<01:51, 55.55it/s]
37%		3622/9800 [01:04<01:46, 58.13it/s]
37%		3630/9800 [01:05<01:42, 60.46it/s]
37%		3638/9800 [01:05<01:35, 64.44it/s]
37%		3645/9800 [01:05<01:40, 60.95it/s]
37%		3653/9800 [01:05<01:34, 65.09it/s]
37%		3660/9800 [01:05<01:43, 59.35it/s]
37%		3667/9800 [01:05<01:44, 58.97it/s]
37%		3674/9800 [01:05<01:47, 57.19it/s]
38%		3683/9800 [01:05<01:38, 62.24it/s]
38%		3693/9800 [01:05<01:27, 69.54it/s]
38%		3705/9800 [01:06<01:17, 78.51it/s]
38%		3714/9800 [01:06<01:45, 57.43it/s]
38%		3722/9800 [01:06<01:52, 53.93it/s]
38%		3729/9800 [01:06<01:46, 57.19it/s]
38%		3736/9800 [01:06<01:42, 59.30it/s]
38%		3743/9800 [01:06<01:38, 61.34it/s]
38%		3750/9800 [01:06<01:40, 60.47it/s]
38%		3757/9800 [01:07<01:54, 52.72it/s]
38%		3763/9800 [01:07<01:57, 51.44it/s]
38%		3769/9800 [01:07<01:52, 53.63it/s]
39%		3776/9800 [01:07<01:46, 56.54it/s]
39%		3785/9800 [01:07<01:37, 61.72it/s]
39%		3792/9800 [01:07<01:37, 61.83it/s]
39%		3800/9800 [01:07<01:31, 65.36it/s]
39%		3807/9800 [01:07<01:33, 63.84it/s]
39%		3817/9800 [01:08<01:30, 65.96it/s]
39%		3824/9800 [01:08<01:57, 50.74it/s]
39%		3830/9800 [01:08<01:59, 50.16it/s]
39%		3839/9800 [01:08<01:46, 55.89it/s]
39%		3846/9800 [01:08<01:42, 58.14it/s]
39%		3853/9800 [01:08<01:47, 55.45it/s]
39%		3860/9800 [01:08<01:41, 58.56it/s]
40%		3871/9800 [01:09<01:36, 61.48it/s]
40%		3878/9800 [01:09<01:36, 61.29it/s]
40%		3888/9800 [01:09<01:27, 67.52it/s]
40%		3896/9800 [01:09<01:23, 70.35it/s]
40%		3905/9800 [01:09<01:20, 73.51it/s]
40%		3915/9800 [01:09<01:16, 77.06it/s]
40%		3923/9800 [01:09<01:17, 76.02it/s]
40%		3931/9800 [01:09<01:24, 69.79it/s]
40%		3940/9800 [01:09<01:26, 67.36it/s]
40%		3947/9800 [01:10<01:26, 67.72it/s]
40%		3954/9800 [01:10<01:41, 57.65it/s]


40%		3961/9800 [01:10<01:39, 58.64it/s]
40%		3968/9800 [01:10<01:41, 57.54it/s]
41%		3974/9800 [01:10<02:32, 38.15it/s]
41%		3979/9800 [01:10<02:27, 39.42it/s]
41%		3984/9800 [01:10<02:22, 40.80it/s]
41%		3992/9800 [01:11<02:07, 45.52it/s]
41%		3998/9800 [01:11<02:31, 38.34it/s]
41%		4008/9800 [01:11<02:04, 46.64it/s]
41%		4018/9800 [01:11<01:49, 52.99it/s]
41%		4025/9800 [01:11<01:43, 55.94it/s]
41%		4032/9800 [01:11<01:56, 49.35it/s]
41%		4038/9800 [01:11<02:01, 47.54it/s]
41%		4044/9800 [01:12<02:06, 45.40it/s]
41%		4050/9800 [01:12<02:05, 45.88it/s]
41%		4057/9800 [01:12<01:53, 50.79it/s]
41%		4064/9800 [01:12<01:46, 54.06it/s]
42%		4070/9800 [01:12<02:17, 41.56it/s]
42%		4075/9800 [01:12<02:11, 43.44it/s]
42%		4080/9800 [01:12<02:10, 43.83it/s]
42%		4085/9800 [01:13<02:08, 44.41it/s]
42%		4092/9800 [01:13<01:54, 49.68it/s]
42%		4103/9800 [01:13<01:39, 57.35it/s]
42%		4110/9800 [01:13<01:54, 49.76it/s]
42%		4120/9800 [01:13<01:37, 58.41it/s]
42%		4127/9800 [01:13<01:51, 50.82it/s]
42%		4133/9800 [01:13<02:19, 40.50it/s]
42%		4143/9800 [01:14<01:54, 49.21it/s]
42%		4151/9800 [01:14<01:41, 55.40it/s]
42%		4158/9800 [01:14<01:50, 50.89it/s]
42%		4165/9800 [01:14<01:48, 52.13it/s]
43%		4171/9800 [01:14<01:49, 51.51it/s]
43%		4179/9800 [01:14<01:39, 56.56it/s]
43%		4186/9800 [01:14<01:34, 59.51it/s]
43%		4193/9800 [01:14<01:44, 53.85it/s]
43%		4199/9800 [01:15<01:53, 49.36it/s]
43%		4206/9800 [01:15<01:47, 52.16it/s]
43%		4217/9800 [01:15<01:30, 61.43it/s]
43%		4225/9800 [01:15<01:25, 65.48it/s]
43%		4233/9800 [01:15<01:27, 63.56it/s]
43%		4240/9800 [01:15<01:37, 57.20it/s]
43%		4249/9800 [01:15<01:27, 63.58it/s]
43%		4256/9800 [01:15<01:30, 61.58it/s]
44%		4263/9800 [01:16<01:36, 57.42it/s]
44%		4271/9800 [01:16<01:28, 62.51it/s]
44%		4281/9800 [01:16<01:18, 69.98it/s]
44%		4289/9800 [01:16<01:27, 63.16it/s]
44%		4296/9800 [01:16<01:27, 63.23it/s]
44%		4303/9800 [01:16<01:28, 62.24it/s]
44%		4310/9800 [01:16<01:27, 62.56it/s]

44%		4317/9800 [01:16<01:42, 53.73it/s]
44%		4326/9800 [01:16<01:30, 60.52it/s]
44%		4333/9800 [01:17<01:33, 58.49it/s]
44%		4340/9800 [01:17<01:37, 55.81it/s]
44%		4347/9800 [01:17<01:42, 53.13it/s]
44%		4354/9800 [01:17<01:36, 56.58it/s]
44%		4361/9800 [01:17<01:38, 55.25it/s]
45%		4369/9800 [01:17<01:30, 59.72it/s]
45%		4376/9800 [01:17<01:35, 56.70it/s]
45%		4383/9800 [01:17<01:30, 59.53it/s]
45%		4390/9800 [01:18<01:40, 53.64it/s]
45%		4398/9800 [01:18<01:31, 59.21it/s]
45%		4406/9800 [01:18<01:28, 60.65it/s]
45%		4413/9800 [01:18<01:26, 62.06it/s]
45%		4420/9800 [01:18<01:29, 60.33it/s]
45%		4428/9800 [01:18<01:26, 61.88it/s]
45%		4435/9800 [01:18<01:29, 59.93it/s]
45%		4442/9800 [01:18<01:33, 57.53it/s]
45%		4453/9800 [01:19<01:20, 66.61it/s]
46%		4461/9800 [01:19<01:23, 63.91it/s]
46%		4469/9800 [01:19<01:19, 66.88it/s]
46%		4479/9800 [01:19<01:17, 68.73it/s]
46%		4487/9800 [01:19<01:18, 67.51it/s]
46%		4494/9800 [01:19<01:20, 65.58it/s]
46%		4501/9800 [01:19<01:22, 64.56it/s]
46%		4508/9800 [01:19<01:33, 56.73it/s]
46%		4514/9800 [01:20<01:32, 57.24it/s]
46%		4520/9800 [01:20<01:36, 55.00it/s]
46%		4532/9800 [01:20<01:21, 64.59it/s]
46%		4540/9800 [01:20<01:45, 50.07it/s]
46%		4548/9800 [01:20<01:38, 53.18it/s]
47%		4559/9800 [01:20<01:26, 60.66it/s]
47%		4566/9800 [01:20<01:37, 53.66it/s]
47%		4573/9800 [01:21<01:46, 48.94it/s]
47%		4580/9800 [01:21<01:37, 53.75it/s]
47%		4587/9800 [01:21<01:34, 55.00it/s]
47%		4598/9800 [01:21<01:21, 64.00it/s]
47%		4606/9800 [01:21<01:19, 65.40it/s]
47%		4614/9800 [01:21<01:41, 50.94it/s]
47%		4622/9800 [01:21<01:33, 55.28it/s]
47%		4630/9800 [01:22<01:25, 60.19it/s]
47%		4637/9800 [01:22<01:24, 60.85it/s]
47%		4644/9800 [01:22<01:23, 61.51it/s]
47%		4651/9800 [01:22<01:36, 53.25it/s]
48%		4660/9800 [01:22<01:32, 55.43it/s]
48%		4666/9800 [01:22<01:37, 52.61it/s]
48%		4675/9800 [01:22<01:25, 59.85it/s]
48%		4682/9800 [01:22<01:25, 60.21it/s]

43/92

52%		5055/9800 [01:29<01:19, 59.37it/s]
52%		5065/9800 [01:29<01:10, 67.15it/s]
52%		5073/9800 [01:29<01:14, 63.28it/s]
52%		5081/9800 [01:30<01:11, 65.56it/s]
52%		5089/9800 [01:30<01:08, 68.76it/s]
52%		5097/9800 [01:30<01:23, 56.21it/s]
52%		5104/9800 [01:30<01:22, 57.15it/s]
52%		5112/9800 [01:30<01:15, 62.04it/s]
52%		5123/9800 [01:30<01:05, 71.34it/s]
52%		5132/9800 [01:30<01:01, 75.69it/s]
52%		5141/9800 [01:30<01:03, 73.81it/s]
53%		5149/9800 [01:31<01:20, 57.83it/s]
53%		5156/9800 [01:31<01:31, 51.00it/s]
53%		5162/9800 [01:31<01:43, 44.96it/s]
53%		5169/9800 [01:31<01:32, 50.10it/s]
53%		5177/9800 [01:31<01:31, 50.31it/s]
53%		5183/9800 [01:31<01:29, 51.62it/s]
53%		5193/9800 [01:31<01:17, 59.24it/s]
53%		5201/9800 [01:32<01:11, 63.98it/s]
53%		5208/9800 [01:32<01:10, 65.25it/s]
53%		5216/9800 [01:32<01:08, 66.57it/s]
53%		5223/9800 [01:32<01:22, 55.27it/s]
53%		5230/9800 [01:32<01:24, 53.84it/s]
53%		5236/9800 [01:32<01:23, 54.71it/s]
53%		5242/9800 [01:32<01:34, 48.13it/s]
54%		5251/9800 [01:32<01:21, 55.67it/s]
54%		5258/9800 [01:33<01:18, 58.06it/s]
54%		5265/9800 [01:33<01:30, 50.36it/s]
54%		5275/9800 [01:33<01:16, 58.77it/s]
54%		5282/9800 [01:33<01:14, 60.96it/s]
54%		5289/9800 [01:33<01:14, 60.84it/s]
54%		5296/9800 [01:33<01:17, 58.11it/s]
54%		5303/9800 [01:33<01:16, 58.76it/s]
54%		5310/9800 [01:33<01:31, 49.02it/s]
54%		5316/9800 [01:34<01:26, 51.68it/s]
54%		5324/9800 [01:34<01:18, 57.21it/s]
54%		5332/9800 [01:34<01:11, 62.48it/s]
54%		5339/9800 [01:34<01:09, 64.50it/s]
55%		5346/9800 [01:34<01:10, 62.75it/s]
55%		5356/9800 [01:34<01:04, 69.18it/s]
55%		5364/9800 [01:34<01:05, 67.36it/s]
55%		5372/9800 [01:34<01:13, 60.37it/s]
55%		5380/9800 [01:35<01:09, 63.75it/s]
55%		5387/9800 [01:35<01:11, 61.39it/s]
55%		5395/9800 [01:35<01:08, 64.01it/s]
55%		5402/9800 [01:35<01:06, 65.69it/s]
55%		5409/9800 [01:35<01:09, 63.54it/s]
55%		5418/9800 [01:35<01:03, 69.20it/s]







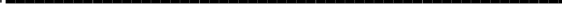








55%		5426/9800 [01:35<01:11, 61.48it/s]
55%		5438/9800 [01:35<01:02, 69.78it/s]
56%		5446/9800 [01:35<01:02, 69.12it/s]
56%		5455/9800 [01:36<00:59, 73.06it/s]
56%		5463/9800 [01:36<01:06, 65.11it/s]
56%		5471/9800 [01:36<01:03, 68.29it/s]
56%		5482/9800 [01:36<00:58, 74.42it/s]
56%		5490/9800 [01:36<00:58, 74.23it/s]
56%		5499/9800 [01:36<00:55, 77.62it/s]
56%		5507/9800 [01:36<01:02, 69.17it/s]
56%		5518/9800 [01:36<00:56, 75.79it/s]
56%		5527/9800 [01:37<00:55, 77.54it/s]
56%		5536/9800 [01:37<01:04, 66.55it/s]
57%		5544/9800 [01:37<01:03, 67.01it/s]
57%		5555/9800 [01:37<00:56, 75.36it/s]
57%		5564/9800 [01:37<01:04, 65.88it/s]
57%		5572/9800 [01:37<01:13, 57.72it/s]
57%		5579/9800 [01:37<01:09, 60.80it/s]
57%		5586/9800 [01:38<01:12, 57.87it/s]
57%		5594/9800 [01:38<01:06, 62.82it/s]
57%		5602/9800 [01:38<01:02, 67.14it/s]
57%		5610/9800 [01:38<01:11, 58.44it/s]
57%		5618/9800 [01:38<01:09, 60.33it/s]
57%		5625/9800 [01:38<01:10, 58.80it/s]
57%		5632/9800 [01:38<01:18, 52.81it/s]
58%		5639/9800 [01:38<01:13, 56.96it/s]
58%		5645/9800 [01:39<01:15, 54.82it/s]
58%		5653/9800 [01:39<01:09, 59.96it/s]
58%		5663/9800 [01:39<01:00, 68.07it/s]
58%		5671/9800 [01:39<01:07, 61.32it/s]
58%		5678/9800 [01:39<01:33, 44.04it/s]
58%		5694/9800 [01:39<01:13, 55.74it/s]
58%		5705/9800 [01:39<01:03, 64.51it/s]
58%		5716/9800 [01:39<00:55, 73.07it/s]
58%		5727/9800 [01:40<00:51, 78.96it/s]
59%		5737/9800 [01:40<00:50, 80.95it/s]
59%		5747/9800 [01:40<00:53, 76.46it/s]
59%		5756/9800 [01:40<00:59, 68.04it/s]
59%		5764/9800 [01:40<01:10, 57.00it/s]
59%		5774/9800 [01:40<01:01, 65.27it/s]
59%		5782/9800 [01:40<01:05, 60.92it/s]
59%		5789/9800 [01:41<01:07, 59.46it/s]
59%		5798/9800 [01:41<01:01, 64.60it/s]
59%		5805/9800 [01:41<01:12, 55.45it/s]
59%		5812/9800 [01:41<01:10, 56.94it/s]
59%		5820/9800 [01:41<01:05, 60.98it/s]
59%		5827/9800 [01:41<01:18, 50.80it/s]
60%		5838/9800 [01:41<01:08, 57.55it/s]
60%		5845/9800 [01:42<01:06, 59.33it/s]

60%		5852/9800 [01:42<01:04, 61.06it/s]
60%		5859/9800 [01:42<01:03, 61.77it/s]
60%		5867/9800 [01:42<01:02, 62.46it/s]
60%		5877/9800 [01:42<01:00, 64.91it/s]
60%		5888/9800 [01:42<00:53, 73.61it/s]
60%		5896/9800 [01:42<01:01, 63.25it/s]
60%		5907/9800 [01:42<00:53, 72.15it/s]
60%		5918/9800 [01:43<00:49, 78.73it/s]
60%		5927/9800 [01:43<00:52, 73.69it/s]
61%		5935/9800 [01:43<00:52, 74.20it/s]
61%		5944/9800 [01:43<00:49, 77.18it/s]
61%		5953/9800 [01:43<01:00, 63.97it/s]
61%		5961/9800 [01:43<01:01, 62.62it/s]
61%		5968/9800 [01:43<01:00, 62.91it/s]
61%		5975/9800 [01:43<00:59, 64.05it/s]
61%		5982/9800 [01:44<01:21, 46.79it/s]
61%		5988/9800 [01:44<01:20, 47.60it/s]
61%		5996/9800 [01:44<01:10, 53.70it/s]
61%		6003/9800 [01:44<01:20, 47.27it/s]
61%		6010/9800 [01:44<01:15, 50.44it/s]
61%		6016/9800 [01:44<01:13, 51.51it/s]
61%		6022/9800 [01:44<01:28, 42.69it/s]
62%		6030/9800 [01:45<01:16, 49.45it/s]
62%		6036/9800 [01:45<01:12, 51.77it/s]
62%		6044/9800 [01:45<01:05, 57.77it/s]
62%		6051/9800 [01:45<01:08, 54.78it/s]
62%		6057/9800 [01:45<01:10, 52.89it/s]
62%		6063/9800 [01:45<01:17, 48.14it/s]
62%		6069/9800 [01:45<01:22, 44.96it/s]
62%		6074/9800 [01:45<01:28, 42.33it/s]
62%		6079/9800 [01:46<01:42, 36.34it/s]
62%		6087/9800 [01:46<01:26, 42.92it/s]
62%		6094/9800 [01:46<01:20, 46.10it/s]
62%		6102/9800 [01:46<01:13, 50.64it/s]
62%		6108/9800 [01:46<01:23, 44.06it/s]
62%		6113/9800 [01:46<01:20, 45.59it/s]
62%		6118/9800 [01:46<01:23, 44.17it/s]
63%		6126/9800 [01:47<01:14, 49.17it/s]
63%		6132/9800 [01:47<01:15, 48.48it/s]
63%		6140/9800 [01:47<01:10, 52.17it/s]
63%		6146/9800 [01:47<01:09, 52.45it/s]
63%		6155/9800 [01:47<01:01, 59.32it/s]
63%		6163/9800 [01:47<00:57, 63.51it/s]
63%		6170/9800 [01:47<00:56, 64.50it/s]
63%		6180/9800 [01:47<00:50, 71.40it/s]
63%		6188/9800 [01:47<00:51, 70.04it/s]
63%		6196/9800 [01:48<00:59, 60.86it/s]
63%		6203/9800 [01:48<01:18, 45.71it/s]

63%		6212/9800 [01:48<01:17, 46.53it/s]
63%		6218/9800 [01:48<01:12, 49.15it/s]
64%		6224/9800 [01:48<01:13, 48.42it/s]
64%		6233/9800 [01:48<01:03, 55.74it/s]
64%		6242/9800 [01:49<01:01, 57.70it/s]
64%		6249/9800 [01:49<01:03, 55.88it/s]
64%		6255/9800 [01:49<01:05, 54.45it/s]
64%		6261/9800 [01:49<01:07, 52.42it/s]
64%		6271/9800 [01:49<01:00, 58.67it/s]
64%		6278/9800 [01:49<00:58, 59.86it/s]
64%		6287/9800 [01:49<00:53, 65.15it/s]
64%		6294/9800 [01:49<01:03, 54.99it/s]
64%		6300/9800 [01:50<01:06, 52.86it/s]
64%		6306/9800 [01:50<01:04, 54.14it/s]
64%		6315/9800 [01:50<00:57, 60.42it/s]
65%		6322/9800 [01:50<00:58, 59.22it/s]
65%		6329/9800 [01:50<00:56, 61.63it/s]
65%		6336/9800 [01:50<01:00, 56.97it/s]
65%		6342/9800 [01:50<01:16, 45.03it/s]
65%		6348/9800 [01:50<01:18, 43.82it/s]
65%		6353/9800 [01:51<01:23, 41.12it/s]
65%		6358/9800 [01:51<01:27, 39.48it/s]
65%		6363/9800 [01:51<01:39, 34.69it/s]
65%		6369/9800 [01:51<01:28, 38.57it/s]
65%		6374/9800 [01:51<01:30, 38.00it/s]
65%		6379/9800 [01:51<01:25, 40.01it/s]
65%		6387/9800 [01:51<01:12, 47.05it/s]
65%		6398/9800 [01:52<01:00, 55.78it/s]
65%		6405/9800 [01:52<01:02, 54.53it/s]
65%		6416/9800 [01:52<00:57, 58.84it/s]
66%		6426/9800 [01:52<00:51, 65.31it/s]
66%		6436/9800 [01:52<00:48, 69.92it/s]
66%		6444/9800 [01:52<00:52, 64.25it/s]
66%		6451/9800 [01:52<00:51, 64.59it/s]
66%		6458/9800 [01:52<00:57, 58.37it/s]
66%		6465/9800 [01:53<00:55, 59.74it/s]
66%		6472/9800 [01:53<01:01, 54.54it/s]
66%		6479/9800 [01:53<00:57, 57.98it/s]
66%		6486/9800 [01:53<01:01, 54.33it/s]
66%		6492/9800 [01:53<01:02, 53.09it/s]
66%		6498/9800 [01:53<01:19, 41.56it/s]
66%		6504/9800 [01:53<01:13, 44.94it/s]
66%		6509/9800 [01:54<01:26, 38.00it/s]
66%		6514/9800 [01:54<01:28, 37.08it/s]
67%		6520/9800 [01:54<01:18, 41.57it/s]
67%		6525/9800 [01:54<01:17, 42.03it/s]
67%		6530/9800 [01:54<01:19, 41.01it/s]
67%		6535/9800 [01:54<01:29, 36.68it/s]
67%		6540/9800 [01:54<01:22, 39.51it/s]

67%		6547/9800 [01:54<01:14, 43.81it/s]
67%		6553/9800 [01:55<01:08, 47.30it/s]
67%		6559/9800 [01:55<01:27, 37.11it/s]
67%		6564/9800 [01:55<01:21, 39.92it/s]
67%		6569/9800 [01:55<01:21, 39.64it/s]
67%		6579/9800 [01:55<01:07, 47.66it/s]
67%		6585/9800 [01:55<01:22, 39.16it/s]
67%		6590/9800 [01:56<01:26, 37.14it/s]
67%		6595/9800 [01:56<01:25, 37.57it/s]
67%		6600/9800 [01:56<01:19, 40.24it/s]
67%		6609/9800 [01:56<01:07, 47.16it/s]
68%		6615/9800 [01:56<01:06, 48.22it/s]
68%		6621/9800 [01:56<01:05, 48.59it/s]
68%		6627/9800 [01:56<01:07, 47.26it/s]
68%		6632/9800 [01:56<01:09, 45.41it/s]
68%		6637/9800 [01:57<01:22, 38.16it/s]
68%		6645/9800 [01:57<01:11, 43.89it/s]
68%		6652/9800 [01:57<01:05, 48.11it/s]
68%		6658/9800 [01:57<01:03, 49.11it/s]
68%		6664/9800 [01:57<01:04, 48.43it/s]
68%		6670/9800 [01:57<01:05, 47.85it/s]
68%		6679/9800 [01:57<00:56, 55.61it/s]
68%		6687/9800 [01:57<00:52, 58.84it/s]
68%		6694/9800 [01:58<01:02, 49.73it/s]
68%		6701/9800 [01:58<00:57, 53.88it/s]
68%		6710/9800 [01:58<00:53, 58.19it/s]
69%		6717/9800 [01:58<00:55, 55.95it/s]
69%		6723/9800 [01:58<00:54, 56.56it/s]
69%		6729/9800 [01:58<00:53, 57.02it/s]
69%		6737/9800 [01:58<00:49, 61.38it/s]
69%		6744/9800 [01:58<00:56, 53.87it/s]
69%		6755/9800 [01:59<00:47, 63.57it/s]
69%		6763/9800 [01:59<00:51, 59.51it/s]
69%		6770/9800 [01:59<00:54, 55.72it/s]
69%		6777/9800 [01:59<00:57, 52.93it/s]
69%		6783/9800 [01:59<00:56, 53.56it/s]
69%		6791/9800 [01:59<00:52, 56.99it/s]
69%		6799/9800 [01:59<00:50, 59.51it/s]
69%		6806/9800 [01:59<01:01, 48.90it/s]
70%		6815/9800 [02:00<00:55, 53.91it/s]
70%		6821/9800 [02:00<00:56, 52.40it/s]
70%		6830/9800 [02:00<00:49, 59.88it/s]
70%		6837/9800 [02:00<00:57, 51.50it/s]
70%		6845/9800 [02:00<00:51, 56.95it/s]
70%		6855/9800 [02:00<00:47, 62.55it/s]
70%		6862/9800 [02:00<00:46, 63.37it/s]
70%		6869/9800 [02:01<00:50, 57.63it/s]
70%		6876/9800 [02:01<00:52, 55.32it/s]












file:///C:/Users/pratyush.acharya/Downloads/07 SVM on Amazon food reviews data.html

73%		7184/9800 [02:07<00:54, 48.39it/s]
73%		7190/9800 [02:07<00:51, 50.25it/s]
73%		7196/9800 [02:08<00:53, 48.75it/s]
73%		7202/9800 [02:08<00:50, 50.97it/s]
74%		7208/9800 [02:08<00:52, 49.79it/s]
74%		7214/9800 [02:08<01:19, 32.63it/s]
74%		7220/9800 [02:08<01:11, 36.05it/s]
74%		7225/9800 [02:08<01:07, 38.13it/s]
74%		7230/9800 [02:08<01:04, 40.13it/s]
74%		7235/9800 [02:09<01:21, 31.42it/s]
74%		7240/9800 [02:09<01:16, 33.34it/s]
74%		7245/9800 [02:09<01:09, 36.73it/s]
74%		7250/9800 [02:09<01:12, 35.01it/s]
74%		7254/9800 [02:09<01:12, 35.15it/s]
74%		7258/9800 [02:09<01:11, 35.78it/s]
74%		7264/9800 [02:09<01:06, 38.00it/s]
74%		7270/9800 [02:10<01:03, 39.88it/s]
74%		7277/9800 [02:10<00:55, 45.63it/s]
74%		7286/9800 [02:10<00:47, 53.43it/s]
74%		7293/9800 [02:10<00:45, 54.77it/s]
75%		7303/9800 [02:10<00:39, 63.05it/s]
75%		7311/9800 [02:10<00:53, 46.68it/s]
75%		7317/9800 [02:10<01:01, 40.41it/s]
75%		7323/9800 [02:11<00:55, 44.57it/s]
75%		7329/9800 [02:11<00:52, 47.19it/s]
75%		7335/9800 [02:11<00:53, 45.75it/s]
75%		7341/9800 [02:11<00:51, 47.36it/s]
75%		7349/9800 [02:11<00:46, 52.86it/s]
75%		7356/9800 [02:11<00:43, 55.72it/s]
75%		7365/9800 [02:11<00:39, 62.06it/s]
75%		7372/9800 [02:11<00:44, 54.40it/s]
75%		7378/9800 [02:12<00:49, 48.87it/s]
75%		7384/9800 [02:12<00:48, 49.48it/s]
75%		7390/9800 [02:12<00:52, 46.15it/s]
75%		7395/9800 [02:12<00:52, 46.07it/s]
76%		7404/9800 [02:12<00:45, 52.95it/s]
76%		7412/9800 [02:12<00:42, 56.32it/s]
76%		7419/9800 [02:12<00:44, 53.55it/s]
76%		7425/9800 [02:12<00:46, 51.27it/s]
76%		7431/9800 [02:13<00:52, 45.37it/s]
76%		7440/9800 [02:13<00:44, 52.61it/s]
76%		7446/9800 [02:13<00:45, 51.47it/s]
76%		7453/9800 [02:13<00:45, 51.89it/s]
76%		7460/9800 [02:13<00:43, 53.22it/s]
76%		7470/9800 [02:13<00:37, 61.67it/s]
76%		7480/9800 [02:13<00:33, 68.91it/s]
76%		7491/9800 [02:13<00:30, 75.64it/s]
77%		7500/9800 [02:14<00:39, 58.38it/s]



















file:///C:/Users/pratyush.acharya/Downloads/07 SVM on Amazon food reviews data.html

80%		7860/9800 [02:20<00:30, 64.06it/s]
80%		7867/9800 [02:20<00:32, 59.26it/s]
80%		7875/9800 [02:20<00:29, 64.21it/s]
80%		7885/9800 [02:21<00:28, 67.16it/s]
81%		7893/9800 [02:21<00:36, 52.57it/s]
81%		7900/9800 [02:21<00:34, 55.29it/s]
81%		7908/9800 [02:21<00:32, 58.65it/s]
81%		7915/9800 [02:21<00:32, 58.42it/s]
81%		7924/9800 [02:21<00:30, 60.74it/s]
81%		7931/9800 [02:21<00:29, 62.76it/s]
81%		7938/9800 [02:22<00:33, 55.58it/s]
81%		7944/9800 [02:22<00:33, 55.72it/s]
81%		7951/9800 [02:22<00:31, 59.31it/s]
81%		7960/9800 [02:22<00:28, 64.91it/s]
81%		7967/9800 [02:22<00:28, 65.15it/s]
81%		7974/9800 [02:22<00:34, 53.12it/s]
81%		7982/9800 [02:22<00:32, 56.69it/s]
82%		7989/9800 [02:22<00:33, 54.57it/s]
82%		7996/9800 [02:23<00:33, 53.19it/s]
82%		8009/9800 [02:23<00:28, 62.71it/s]
82%		8019/9800 [02:23<00:25, 68.80it/s]
82%		8030/9800 [02:23<00:23, 75.79it/s]
82%		8039/9800 [02:23<00:24, 72.37it/s]
82%		8047/9800 [02:23<00:25, 69.04it/s]
82%		8055/9800 [02:23<00:27, 63.53it/s]
82%		8062/9800 [02:23<00:28, 60.87it/s]
82%		8072/9800 [02:24<00:25, 68.57it/s]
82%		8085/9800 [02:24<00:21, 79.27it/s]
83%		8094/9800 [02:24<00:21, 79.99it/s]
83%		8103/9800 [02:24<00:21, 79.78it/s]
83%		8112/9800 [02:24<00:21, 79.60it/s]
83%		8121/9800 [02:24<00:20, 80.49it/s]
83%		8130/9800 [02:24<00:22, 74.69it/s]
83%		8139/9800 [02:24<00:21, 75.82it/s]
83%		8147/9800 [02:24<00:23, 71.00it/s]
83%		8155/9800 [02:25<00:23, 69.63it/s]
83%		8163/9800 [02:25<00:27, 59.78it/s]
83%		8170/9800 [02:25<00:27, 59.87it/s]
83%		8177/9800 [02:25<00:26, 62.16it/s]
84%		8187/9800 [02:25<00:23, 69.28it/s]
84%		8195/9800 [02:25<00:24, 64.26it/s]
84%		8202/9800 [02:25<00:26, 59.84it/s]
84%		8209/9800 [02:26<00:29, 53.17it/s]
84%		8217/9800 [02:26<00:27, 56.67it/s]
84%		8224/9800 [02:26<00:26, 59.88it/s]
84%		8231/9800 [02:26<00:29, 52.60it/s]
84%		8238/9800 [02:26<00:28, 55.06it/s]
84%		8246/9800 [02:26<00:25, 60.40it/s]

84%		8253/9800 [02:26<00:27, 56.23it/s]
84%		8263/9800 [02:26<00:23, 64.39it/s]
84%		8271/9800 [02:27<00:28, 53.96it/s]
84%		8278/9800 [02:27<00:34, 44.39it/s]
85%		8284/9800 [02:27<00:43, 34.56it/s]
85%		8290/9800 [02:27<00:38, 38.76it/s]
85%		8295/9800 [02:27<00:36, 41.40it/s]
85%		8300/9800 [02:27<00:35, 42.26it/s]
85%		8308/9800 [02:28<00:31, 48.12it/s]
85%		8317/9800 [02:28<00:30, 48.15it/s]
85%		8325/9800 [02:28<00:28, 51.60it/s]
85%		8331/9800 [02:28<00:30, 48.59it/s]
85%		8337/9800 [02:28<00:29, 50.22it/s]
85%		8345/9800 [02:28<00:26, 54.62it/s]
85%		8353/9800 [02:28<00:24, 58.25it/s]
85%		8360/9800 [02:28<00:26, 53.45it/s]
85%		8366/9800 [02:29<00:28, 50.99it/s]
85%		8372/9800 [02:29<00:29, 49.11it/s]
85%		8378/9800 [02:29<00:29, 48.41it/s]
86%		8383/9800 [02:29<00:52, 27.06it/s]
86%		8387/9800 [02:29<00:50, 27.88it/s]
86%		8391/9800 [02:30<00:48, 29.14it/s]
86%		8395/9800 [02:30<00:45, 31.10it/s]
86%		8399/9800 [02:30<00:43, 32.23it/s]
86%		8408/9800 [02:30<00:34, 39.87it/s]
86%		8415/9800 [02:30<00:30, 45.44it/s]
86%		8424/9800 [02:30<00:26, 52.52it/s]
86%		8432/9800 [02:30<00:24, 55.79it/s]
86%		8439/9800 [02:30<00:23, 58.30it/s]
86%		8448/9800 [02:30<00:20, 64.55it/s]
86%		8460/9800 [02:30<00:18, 74.39it/s]
86%		8469/9800 [02:31<00:19, 67.10it/s]
86%		8477/9800 [02:31<00:20, 63.88it/s]
87%		8487/9800 [02:31<00:18, 71.62it/s]
87%		8495/9800 [02:31<00:19, 66.64it/s]
87%		8503/9800 [02:31<00:19, 64.93it/s]
87%		8510/9800 [02:31<00:21, 59.63it/s]
87%		8517/9800 [02:31<00:23, 55.41it/s]
87%		8523/9800 [02:32<00:23, 54.62it/s]
87%		8529/9800 [02:32<00:23, 54.68it/s]
87%		8538/9800 [02:32<00:20, 60.16it/s]
87%		8545/9800 [02:32<00:21, 57.62it/s]
87%		8552/9800 [02:32<00:21, 58.89it/s]
87%		8559/9800 [02:32<00:22, 54.81it/s]
87%		8565/9800 [02:32<00:28, 43.76it/s]
87%		8570/9800 [02:33<00:30, 40.57it/s]
88%		8575/9800 [02:33<00:39, 31.20it/s]
88%		8581/9800 [02:33<00:33, 36.36it/s]
88%		8590/9800 [02:33<00:27, 43.64it/s]

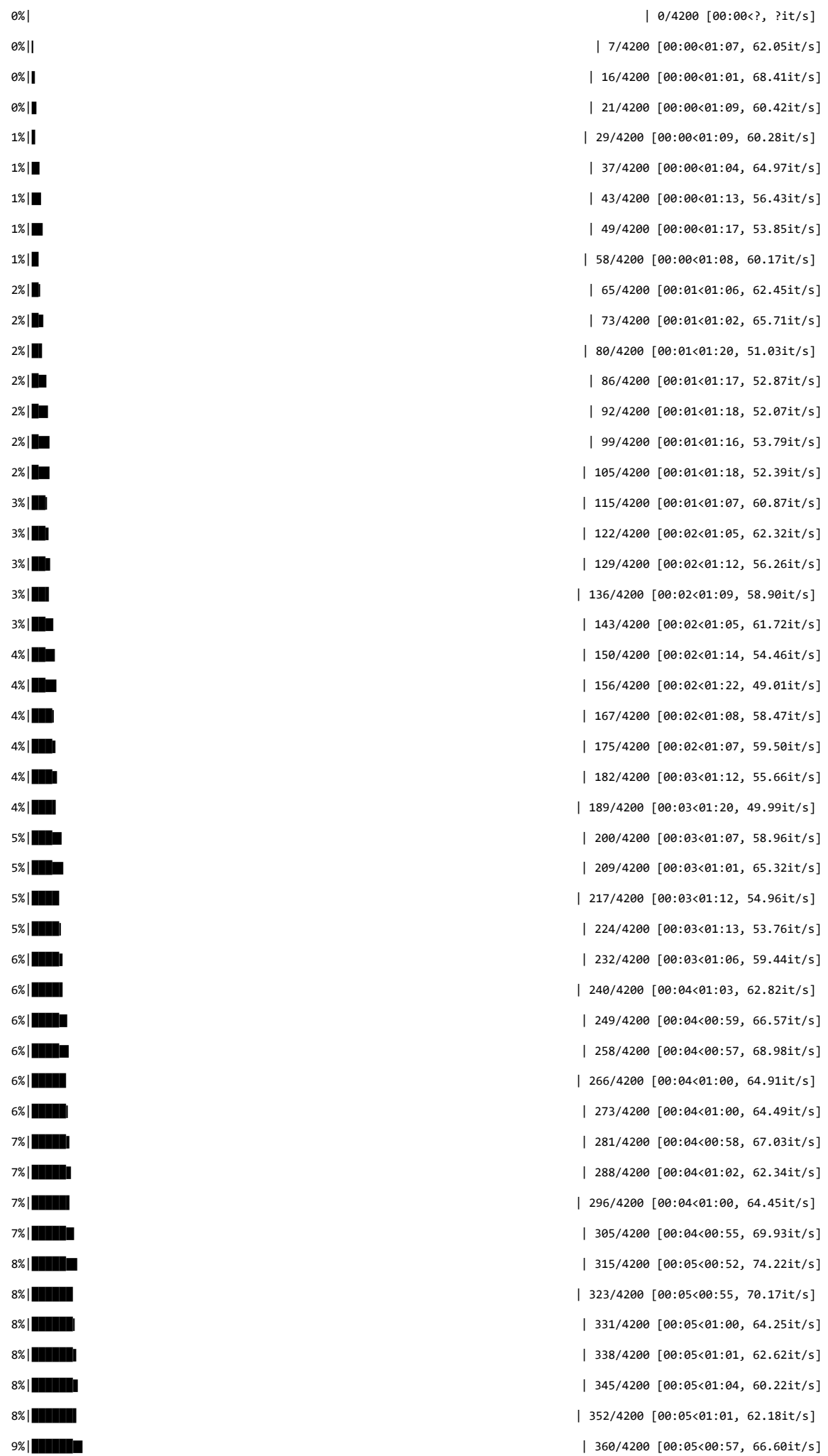
88%		8596/9800 [02:33<00:29, 40.73it/s]
88%		8601/9800 [02:33<00:31, 38.58it/s]
88%		8606/9800 [02:33<00:29, 41.04it/s]
88%		8612/9800 [02:34<00:26, 44.65it/s]
88%		8620/9800 [02:34<00:25, 46.54it/s]
88%		8630/9800 [02:34<00:21, 53.72it/s]
88%		8636/9800 [02:34<00:27, 41.99it/s]
88%		8642/9800 [02:34<00:27, 42.62it/s]
88%		8649/9800 [02:34<00:24, 47.02it/s]
88%		8655/9800 [02:34<00:23, 47.99it/s]
88%		8662/9800 [02:34<00:21, 52.47it/s]
88%		8668/9800 [02:35<00:23, 48.54it/s]
89%		8677/9800 [02:35<00:22, 50.43it/s]
89%		8685/9800 [02:35<00:19, 56.08it/s]
89%		8691/9800 [02:35<00:19, 56.33it/s]
89%		8701/9800 [02:35<00:17, 62.48it/s]
89%		8708/9800 [02:35<00:20, 53.18it/s]
89%		8716/9800 [02:35<00:19, 56.78it/s]
89%		8724/9800 [02:36<00:18, 59.19it/s]
89%		8735/9800 [02:36<00:16, 66.56it/s]
89%		8743/9800 [02:36<00:16, 64.10it/s]
89%		8752/9800 [02:36<00:14, 69.91it/s]
89%		8760/9800 [02:36<00:15, 67.62it/s]
89%		8768/9800 [02:36<00:28, 35.73it/s]
90%		8774/9800 [02:37<00:28, 35.96it/s]
90%		8782/9800 [02:37<00:23, 42.94it/s]
90%		8789/9800 [02:37<00:21, 47.69it/s]
90%		8795/9800 [02:37<00:20, 49.90it/s]
90%		8802/9800 [02:37<00:20, 48.99it/s]
90%		8808/9800 [02:37<00:19, 51.49it/s]
90%		8814/9800 [02:37<00:18, 52.95it/s]
90%		8822/9800 [02:37<00:16, 58.77it/s]
90%		8829/9800 [02:38<00:17, 55.74it/s]
90%		8839/9800 [02:38<00:15, 63.31it/s]
90%		8846/9800 [02:38<00:15, 62.61it/s]
90%		8853/9800 [02:38<00:15, 60.30it/s]
90%		8861/9800 [02:38<00:15, 62.48it/s]
91%		8871/9800 [02:38<00:13, 69.89it/s]
91%		8879/9800 [02:38<00:13, 69.67it/s]
91%		8887/9800 [02:38<00:13, 66.88it/s]
91%		8895/9800 [02:38<00:13, 66.94it/s]
91%		8904/9800 [02:39<00:12, 71.09it/s]
91%		8912/9800 [02:39<00:12, 72.71it/s]
91%		8920/9800 [02:39<00:13, 66.94it/s]
91%		8927/9800 [02:39<00:14, 59.84it/s]
91%		8934/9800 [02:39<00:15, 56.73it/s]
91%		8943/9800 [02:39<00:13, 62.77it/s]
91%		8950/9800 [02:39<00:14, 57.32it/s]

91%		8957/9800 [02:40<00:15, 54.76it/s]
91%		8965/9800 [02:40<00:14, 57.68it/s]
92%		8974/9800 [02:40<00:12, 64.60it/s]
92%		8981/9800 [02:40<00:13, 59.14it/s]
92%		8995/9800 [02:40<00:11, 67.79it/s]
92%		9003/9800 [02:40<00:11, 70.44it/s]
92%		9011/9800 [02:40<00:11, 69.68it/s]
92%		9019/9800 [02:40<00:11, 67.25it/s]
92%		9027/9800 [02:41<00:11, 66.14it/s]
92%		9038/9800 [02:41<00:10, 73.85it/s]
92%		9046/9800 [02:41<00:10, 70.66it/s]
92%		9054/9800 [02:41<00:11, 64.72it/s]
92%		9061/9800 [02:41<00:12, 58.82it/s]
93%		9068/9800 [02:41<00:11, 61.03it/s]
93%		9078/9800 [02:41<00:10, 67.78it/s]
93%		9086/9800 [02:41<00:11, 63.39it/s]
93%		9095/9800 [02:42<00:10, 65.02it/s]
93%		9104/9800 [02:42<00:09, 69.86it/s]
93%		9113/9800 [02:42<00:09, 74.19it/s]
93%		9121/9800 [02:42<00:09, 72.81it/s]
93%		9131/9800 [02:42<00:08, 78.27it/s]
93%		9140/9800 [02:42<00:09, 70.65it/s]
93%		9148/9800 [02:42<00:09, 69.66it/s]
93%		9156/9800 [02:42<00:12, 50.78it/s]
94%		9163/9800 [02:43<00:11, 55.31it/s]
94%		9170/9800 [02:43<00:11, 55.54it/s]
94%		9177/9800 [02:43<00:11, 52.12it/s]
94%		9183/9800 [02:43<00:11, 53.00it/s]
94%		9189/9800 [02:43<00:12, 47.55it/s]
94%		9195/9800 [02:43<00:14, 41.56it/s]
94%		9207/9800 [02:43<00:11, 50.30it/s]
94%		9215/9800 [02:44<00:10, 53.96it/s]
94%		9222/9800 [02:44<00:10, 56.12it/s]
94%		9229/9800 [02:44<00:11, 48.79it/s]
94%		9241/9800 [02:44<00:09, 58.63it/s]
94%		9251/9800 [02:44<00:08, 66.47it/s]
94%		9261/9800 [02:44<00:07, 70.34it/s]
95%		9269/9800 [02:44<00:07, 66.75it/s]
95%		9277/9800 [02:45<00:09, 56.19it/s]
95%		9284/9800 [02:45<00:12, 41.25it/s]
95%		9290/9800 [02:45<00:11, 44.12it/s]
95%		9296/9800 [02:45<00:11, 42.02it/s]
95%		9306/9800 [02:45<00:09, 50.42it/s]
95%		9313/9800 [02:45<00:08, 54.33it/s]
95%		9321/9800 [02:45<00:08, 55.41it/s]
95%		9328/9800 [02:46<00:10, 46.22it/s]
95%		9334/9800 [02:46<00:09, 48.46it/s]
95%		9340/9800 [02:46<00:09, 49.35it/s]
95%		9346/9800 [02:46<00:09, 46.99it/s]

















































95%		9354/9800 [02:46<00:08, 53.03it/s]
96%		9360/9800 [02:46<00:09, 46.83it/s]
96%		9370/9800 [02:46<00:07, 54.60it/s]
96%		9377/9800 [02:47<00:07, 53.44it/s]
96%		9384/9800 [02:47<00:07, 57.09it/s]
96%		9397/9800 [02:47<00:05, 68.50it/s]
96%		9406/9800 [02:47<00:05, 70.15it/s]
96%		9414/9800 [02:47<00:05, 68.92it/s]
96%		9422/9800 [02:47<00:06, 54.68it/s]
96%		9429/9800 [02:47<00:06, 55.79it/s]
96%		9438/9800 [02:47<00:05, 62.66it/s]
96%		9446/9800 [02:48<00:05, 65.36it/s]
96%		9454/9800 [02:48<00:06, 56.93it/s]
97%		9461/9800 [02:48<00:05, 59.85it/s]
97%		9468/9800 [02:48<00:05, 58.53it/s]
97%		9475/9800 [02:48<00:05, 57.99it/s]
97%		9485/9800 [02:48<00:04, 64.15it/s]
97%		9492/9800 [02:48<00:06, 49.80it/s]
97%		9498/9800 [02:49<00:06, 48.55it/s]
97%		9505/9800 [02:49<00:05, 52.50it/s]
97%		9512/9800 [02:49<00:05, 55.92it/s]
97%		9522/9800 [02:49<00:04, 64.39it/s]
97%		9530/9800 [02:49<00:04, 63.18it/s]
97%		9537/9800 [02:49<00:04, 53.33it/s]
97%		9543/9800 [02:49<00:04, 52.10it/s]
97%		9549/9800 [02:49<00:04, 52.28it/s]
98%		9555/9800 [02:50<00:04, 50.70it/s]
98%		9561/9800 [02:50<00:04, 51.60it/s]
98%		9572/9800 [02:50<00:03, 60.51it/s]
98%		9579/9800 [02:50<00:04, 53.50it/s]
98%		9586/9800 [02:50<00:04, 50.87it/s]
98%		9592/9800 [02:50<00:04, 51.63it/s]
98%		9600/9800 [02:50<00:03, 55.74it/s]
98%		9606/9800 [02:50<00:03, 48.69it/s]
98%		9614/9800 [02:51<00:03, 54.97it/s]
98%		9621/9800 [02:51<00:04, 35.81it/s]
98%		9629/9800 [02:51<00:04, 42.15it/s]
98%		9637/9800 [02:51<00:03, 48.03it/s]
98%		9644/9800 [02:51<00:03, 43.96it/s]
98%		9652/9800 [02:51<00:03, 48.82it/s]
99%		9658/9800 [02:52<00:02, 49.67it/s]
99%		9667/9800 [02:52<00:02, 55.83it/s]
99%		9674/9800 [02:52<00:02, 59.10it/s]
99%		9681/9800 [02:52<00:02, 57.40it/s]
99%		9690/9800 [02:52<00:01, 63.26it/s]
99%		9697/9800 [02:52<00:01, 55.01it/s]
99%		9704/9800 [02:52<00:01, 51.56it/s]
99%		9712/9800 [02:52<00:01, 54.75it/s]



```
In [110]: import time
start1 = time.clock()
t = tf_idf_vect.get_feature_names()
tfidf_sent_vectors_CV = []; # the tfidf-w2v for each sentence/review is stored in this list
row=0;
for sent in tqdm(list_sent_CV):
    sent_vec = np.zeros(50)
    cnt_words = 0; # num of words with a valid vector in the sentence/review
    for word in sent: # for each word in a review/sentence
        if word in w2v_words:
            vec = w2v_model.wv[word]
            tfidf = tfidf_cv[row,t.index(word)]
            sent_vec += (vec * tfidf)
            cnt_words += tfidf
    if cnt_words != 0:
        sent_vec /= cnt_words
    tfidf_sent_vectors_CV.append(sent_vec)
    row += 1
print(len(tfidf_sent_vectors_CV))
print(len(tfidf_sent_vectors_CV[0]))
print((time.clock()-start1)/60)
```



9%	<div></div>	373/4200 [00:05<00:51, 74.54it/s]
9%	<div></div>	381/4200 [00:06<01:01, 62.24it/s]
9%	<div></div>	390/4200 [00:06<00:57, 65.85it/s]
9%	<div></div>	398/4200 [00:06<01:00, 63.22it/s]
10%	<div></div>	405/4200 [00:06<01:00, 62.29it/s]
10%	<div></div>	412/4200 [00:06<01:12, 52.52it/s]
10%	<div></div>	418/4200 [00:06<01:22, 46.08it/s]
10%	<div></div>	424/4200 [00:06<01:18, 48.16it/s]
10%	<div></div>	430/4200 [00:07<01:13, 50.99it/s]
10%	<div></div>	437/4200 [00:07<01:10, 53.68it/s]
11%	<div></div>	445/4200 [00:07<01:06, 56.53it/s]
11%	<div></div>	452/4200 [00:07<01:03, 58.62it/s]
11%	<div></div>	461/4200 [00:07<01:00, 61.87it/s]
11%	<div></div>	470/4200 [00:07<00:56, 66.38it/s]
11%	<div></div>	477/4200 [00:07<01:09, 53.77it/s]
12%	<div></div>	487/4200 [00:07<01:02, 59.00it/s]
12%	<div></div>	494/4200 [00:08<01:07, 54.63it/s]
12%	<div></div>	502/4200 [00:08<01:01, 59.90it/s]
12%	<div></div>	509/4200 [00:08<01:01, 59.86it/s]
12%	<div></div>	521/4200 [00:08<00:52, 70.17it/s]
13%	<div></div>	529/4200 [00:08<00:57, 63.48it/s]
13%	<div></div>	537/4200 [00:08<00:56, 64.75it/s]
13%	<div></div>	545/4200 [00:08<01:02, 58.16it/s]
13%	<div></div>	552/4200 [00:08<01:06, 55.05it/s]
13%	<div></div>	559/4200 [00:09<01:04, 56.44it/s]
13%	<div></div>	565/4200 [00:09<01:04, 56.26it/s]
14%	<div></div>	571/4200 [00:09<01:14, 48.89it/s]
14%	<div></div>	579/4200 [00:09<01:05, 54.99it/s]
14%	<div></div>	585/4200 [00:09<01:12, 49.95it/s]
14%	<div></div>	591/4200 [00:09<01:09, 52.14it/s]
14%	<div></div>	597/4200 [00:09<01:18, 46.01it/s]
14%	<div></div>	602/4200 [00:10<01:23, 43.01it/s]
14%	<div></div>	607/4200 [00:10<01:36, 37.28it/s]
15%	<div></div>	615/4200 [00:10<01:21, 43.84it/s]
15%	<div></div>	622/4200 [00:10<01:12, 49.34it/s]
15%	<div></div>	630/4200 [00:10<01:05, 54.61it/s]
15%	<div></div>	640/4200 [00:10<00:58, 61.34it/s]
15%	<div></div>	648/4200 [00:10<00:55, 63.61it/s]
16%	<div></div>	655/4200 [00:10<00:57, 61.83it/s]
16%	<div></div>	662/4200 [00:10<00:56, 62.16it/s]
16%	<div></div>	669/4200 [00:11<00:58, 60.66it/s]
16%	<div></div>	677/4200 [00:11<00:54, 64.55it/s]
16%	<div></div>	684/4200 [00:11<00:56, 62.69it/s]
16%	<div></div>	693/4200 [00:11<00:52, 66.58it/s]
17%	<div></div>	700/4200 [00:11<00:57, 60.97it/s]
17%	<div></div>	707/4200 [00:11<01:01, 57.13it/s]
17%	<div></div>	713/4200 [00:11<01:04, 54.05it/s]
17%	<div></div>	720/4200 [00:11<01:03, 55.18it/s]
17%	<div></div>	726/4200 [00:12<01:01, 56.11it/s]

17%		732/4200 [00:12<01:03, 55.02it/s]
18%		738/4200 [00:12<01:02, 55.64it/s]
18%		744/4200 [00:12<01:02, 55.60it/s]
18%		750/4200 [00:12<01:03, 54.14it/s]
18%		756/4200 [00:12<01:04, 53.59it/s]
18%		763/4200 [00:12<01:00, 56.91it/s]
18%		771/4200 [00:12<00:56, 60.53it/s]
19%		778/4200 [00:12<00:55, 61.31it/s]
19%		786/4200 [00:13<00:53, 64.30it/s]
19%		793/4200 [00:13<00:59, 57.15it/s]
19%		800/4200 [00:13<01:00, 56.36it/s]
19%		806/4200 [00:13<01:46, 31.86it/s]
19%		811/4200 [00:13<01:48, 31.10it/s]
19%		817/4200 [00:14<01:36, 35.23it/s]
20%		823/4200 [00:14<01:25, 39.38it/s]
20%		828/4200 [00:14<01:21, 41.31it/s]
20%		833/4200 [00:14<01:36, 34.86it/s]
20%		841/4200 [00:14<01:21, 41.30it/s]
20%		852/4200 [00:14<01:06, 50.43it/s]
21%		863/4200 [00:14<00:55, 59.80it/s]
21%		871/4200 [00:14<00:54, 60.63it/s]
21%		879/4200 [00:15<00:55, 60.22it/s]
21%		886/4200 [00:15<00:52, 62.64it/s]
21%		894/4200 [00:15<00:50, 64.85it/s]
21%		901/4200 [00:15<00:50, 65.30it/s]
22%		908/4200 [00:15<00:51, 64.04it/s]
22%		915/4200 [00:15<00:56, 58.21it/s]
22%		923/4200 [00:15<00:52, 62.10it/s]
22%		930/4200 [00:15<00:54, 60.14it/s]
22%		938/4200 [00:15<00:50, 64.17it/s]
23%		950/4200 [00:16<00:44, 73.56it/s]
23%		958/4200 [00:16<00:54, 59.18it/s]
23%		967/4200 [00:16<00:49, 65.44it/s]
23%		977/4200 [00:16<00:45, 71.28it/s]
23%		985/4200 [00:16<00:52, 61.79it/s]
24%		992/4200 [00:16<00:53, 59.94it/s]
24%		999/4200 [00:16<00:53, 59.54it/s]
24%		1006/4200 [00:16<00:53, 59.88it/s]
24%		1015/4200 [00:17<00:48, 66.30it/s]
24%		1023/4200 [00:17<00:50, 63.05it/s]
25%		1032/4200 [00:17<00:49, 64.54it/s]
25%		1039/4200 [00:17<00:50, 62.35it/s]
25%		1047/4200 [00:17<00:48, 64.40it/s]
25%		1054/4200 [00:17<00:50, 62.36it/s]
25%		1062/4200 [00:17<00:47, 66.58it/s]
26%		1071/4200 [00:17<00:43, 72.20it/s]
26%		1080/4200 [00:18<00:41, 75.01it/s]
26%		1089/4200 [00:18<00:41, 74.23it/s]









































26%		1097/4200 [00:18<00:43, 71.57it/s]
26%		1105/4200 [00:18<00:43, 71.72it/s]
27%		1116/4200 [00:18<00:39, 77.73it/s]
27%		1124/4200 [00:18<00:42, 72.90it/s]
27%		1132/4200 [00:18<00:44, 69.48it/s]
27%		1140/4200 [00:18<00:42, 71.95it/s]
27%		1148/4200 [00:19<00:51, 59.10it/s]
28%		1155/4200 [00:19<00:52, 58.29it/s]
28%		1166/4200 [00:19<00:48, 62.87it/s]
28%		1175/4200 [00:19<00:45, 66.48it/s]
28%		1185/4200 [00:19<00:41, 72.34it/s]
28%		1193/4200 [00:19<00:41, 73.12it/s]
29%		1201/4200 [00:19<00:41, 71.68it/s]
29%		1211/4200 [00:19<00:39, 76.08it/s]
29%		1219/4200 [00:19<00:39, 75.40it/s]
29%		1227/4200 [00:20<00:40, 72.57it/s]
30%		1240/4200 [00:20<00:36, 82.11it/s]
30%		1249/4200 [00:20<00:42, 69.83it/s]
30%		1257/4200 [00:20<00:44, 66.79it/s]
30%		1265/4200 [00:20<00:42, 68.52it/s]
30%		1273/4200 [00:20<00:41, 70.34it/s]
31%		1282/4200 [00:20<00:43, 67.31it/s]
31%		1289/4200 [00:21<00:48, 60.45it/s]
31%		1296/4200 [00:21<00:53, 54.46it/s]
31%		1302/4200 [00:21<00:55, 52.22it/s]
31%		1308/4200 [00:21<00:55, 52.04it/s]
31%		1316/4200 [00:21<00:50, 56.93it/s]
32%		1328/4200 [00:21<00:44, 64.52it/s]
32%		1336/4200 [00:21<00:42, 66.95it/s]
32%		1344/4200 [00:21<00:47, 59.81it/s]
32%		1353/4200 [00:22<00:43, 65.31it/s]
32%		1360/4200 [00:22<00:43, 65.22it/s]
33%		1370/4200 [00:22<00:41, 68.32it/s]
33%		1378/4200 [00:22<00:50, 55.71it/s]
33%		1389/4200 [00:22<00:46, 60.72it/s]
33%		1396/4200 [00:22<00:50, 55.50it/s]
33%		1403/4200 [00:22<00:53, 52.71it/s]
34%		1411/4200 [00:23<00:48, 57.93it/s]
34%		1419/4200 [00:23<00:46, 59.20it/s]
34%		1426/4200 [00:23<00:46, 60.08it/s]
34%		1433/4200 [00:23<00:46, 59.92it/s]
34%		1440/4200 [00:23<00:45, 60.15it/s]
34%		1447/4200 [00:23<00:45, 60.15it/s]
35%		1455/4200 [00:23<00:48, 57.14it/s]
35%		1461/4200 [00:23<00:49, 55.09it/s]
35%		1471/4200 [00:24<00:43, 63.19it/s]
35%		1478/4200 [00:24<00:42, 64.61it/s]
35%		1485/4200 [00:24<00:44, 61.65it/s]
36%		1492/4200 [00:24<00:46, 58.61it/s]



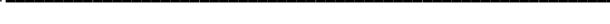








36%		1499/4200 [00:24<01:01, 43.91it/s]
36%		1506/4200 [00:24<00:55, 48.58it/s]
36%		1515/4200 [00:24<00:48, 54.95it/s]
36%		1523/4200 [00:24<00:45, 59.21it/s]
36%		1530/4200 [00:25<00:44, 59.41it/s]
37%		1538/4200 [00:25<00:42, 62.65it/s]
37%		1545/4200 [00:25<00:48, 54.77it/s]
37%		1553/4200 [00:25<00:43, 60.48it/s]
37%		1560/4200 [00:25<00:42, 62.29it/s]
37%		1567/4200 [00:25<00:45, 57.93it/s]
38%		1575/4200 [00:25<00:41, 62.61it/s]
38%		1583/4200 [00:25<00:39, 66.41it/s]
38%		1593/4200 [00:26<00:36, 71.37it/s]
38%		1602/4200 [00:26<00:35, 74.16it/s]
38%		1610/4200 [00:26<00:51, 49.85it/s]
39%		1619/4200 [00:26<00:45, 56.33it/s]
39%		1630/4200 [00:26<00:40, 63.89it/s]
39%		1640/4200 [00:26<00:36, 69.85it/s]
39%		1648/4200 [00:27<00:52, 48.24it/s]
39%		1655/4200 [00:27<00:54, 46.69it/s]
40%		1663/4200 [00:27<00:49, 51.56it/s]
40%		1674/4200 [00:27<00:41, 60.86it/s]
40%		1682/4200 [00:27<00:43, 57.41it/s]
40%		1694/4200 [00:27<00:37, 67.45it/s]
41%		1704/4200 [00:27<00:33, 74.02it/s]
41%		1713/4200 [00:27<00:33, 73.49it/s]
41%		1722/4200 [00:28<00:34, 72.37it/s]
41%		1730/4200 [00:28<00:45, 54.64it/s]
41%		1738/4200 [00:28<00:41, 59.99it/s]
42%		1745/4200 [00:28<00:39, 61.48it/s]
42%		1752/4200 [00:28<00:44, 54.70it/s]
42%		1759/4200 [00:28<00:45, 53.10it/s]
42%		1767/4200 [00:28<00:41, 58.80it/s]
42%		1774/4200 [00:29<00:47, 51.37it/s]
42%		1780/4200 [00:29<00:46, 52.12it/s]
43%		1786/4200 [00:29<00:52, 46.15it/s]
43%		1797/4200 [00:29<00:43, 54.80it/s]
43%		1804/4200 [00:29<00:42, 56.11it/s]
43%		1812/4200 [00:29<00:40, 58.83it/s]
43%		1819/4200 [00:29<00:41, 57.92it/s]
43%		1826/4200 [00:29<00:40, 58.90it/s]
44%		1835/4200 [00:30<00:36, 64.68it/s]
44%		1842/4200 [00:30<00:37, 62.77it/s]
44%		1853/4200 [00:30<00:32, 71.36it/s]
44%		1861/4200 [00:30<00:32, 73.06it/s]
44%		1869/4200 [00:30<00:38, 61.01it/s]
45%		1876/4200 [00:30<00:39, 58.75it/s]
45%		1884/4200 [00:30<00:36, 62.97it/s]











45%		1891/4200 [00:30<00:36, 63.48it/s]
45%		1900/4200 [00:30<00:33, 68.32it/s]
45%		1908/4200 [00:31<00:32, 69.79it/s]
46%		1916/4200 [00:31<00:32, 69.35it/s]
46%		1924/4200 [00:31<00:36, 61.99it/s]
46%		1932/4200 [00:31<00:35, 64.09it/s]
46%		1939/4200 [00:31<00:38, 59.12it/s]
46%		1946/4200 [00:31<00:37, 59.47it/s]
46%		1953/4200 [00:31<00:36, 62.09it/s]
47%		1962/4200 [00:31<00:33, 67.60it/s]
47%		1969/4200 [00:32<00:39, 56.74it/s]
47%		1978/4200 [00:32<00:35, 62.74it/s]
47%		1987/4200 [00:32<00:32, 67.63it/s]
48%		1995/4200 [00:32<00:38, 57.88it/s]
48%		2002/4200 [00:32<00:37, 57.89it/s]
48%		2009/4200 [00:32<00:42, 52.08it/s]
48%		2015/4200 [00:32<00:46, 46.53it/s]
48%		2024/4200 [00:33<00:41, 52.33it/s]
48%		2032/4200 [00:33<00:38, 56.60it/s]
49%		2039/4200 [00:33<00:44, 48.98it/s]
49%		2045/4200 [00:33<00:54, 39.55it/s]
49%		2050/4200 [00:33<00:57, 37.37it/s]
49%		2055/4200 [00:33<01:05, 32.90it/s]
49%		2059/4200 [00:34<01:13, 29.01it/s]
49%		2067/4200 [00:34<01:00, 35.37it/s]
49%		2075/4200 [00:34<00:51, 41.39it/s]
50%		2081/4200 [00:34<00:49, 42.48it/s]
50%		2086/4200 [00:34<00:49, 42.63it/s]
50%		2091/4200 [00:34<00:49, 42.70it/s]
50%		2096/4200 [00:34<00:49, 42.91it/s]
50%		2103/4200 [00:34<00:44, 46.63it/s]
50%		2111/4200 [00:35<00:39, 53.28it/s]
50%		2117/4200 [00:35<00:39, 52.27it/s]
51%		2124/4200 [00:35<00:36, 56.20it/s]
51%		2130/4200 [00:35<00:43, 47.25it/s]
51%		2136/4200 [00:35<00:46, 44.20it/s]
51%		2141/4200 [00:35<00:45, 45.40it/s]
51%		2149/4200 [00:35<00:40, 50.83it/s]
51%		2157/4200 [00:35<00:36, 56.41it/s]
52%		2164/4200 [00:36<00:36, 56.17it/s]
52%		2171/4200 [00:36<00:34, 59.59it/s]
52%		2178/4200 [00:36<00:32, 61.64it/s]
52%		2186/4200 [00:36<00:31, 64.17it/s]
52%		2193/4200 [00:36<00:31, 64.28it/s]
52%		2200/4200 [00:36<00:31, 62.95it/s]
53%		2207/4200 [00:36<00:34, 57.31it/s]
53%		2215/4200 [00:36<00:31, 62.46it/s]
53%		2222/4200 [00:37<00:34, 57.18it/s]
53%		2228/4200 [00:37<00:42, 46.86it/s]

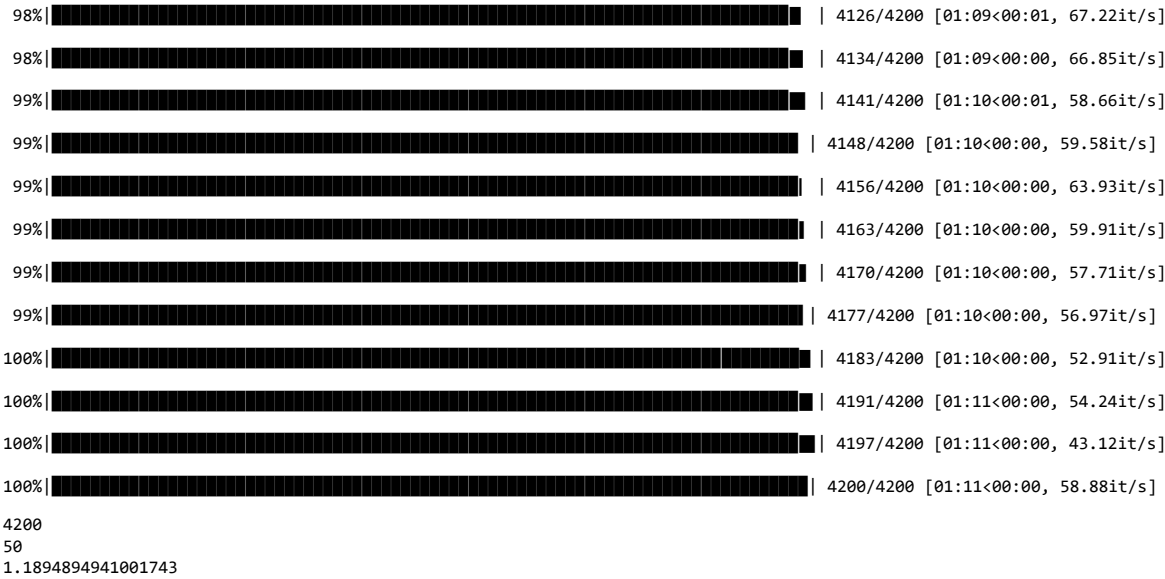
53%		2234/4200 [00:37<00:45, 43.09it/s]
53%		2239/4200 [00:37<00:44, 43.66it/s]
54%		2250/4200 [00:37<00:36, 53.11it/s]
54%		2257/4200 [00:37<00:34, 56.16it/s]
54%		2264/4200 [00:37<00:39, 48.69it/s]
54%		2274/4200 [00:37<00:33, 57.41it/s]
54%		2281/4200 [00:38<00:32, 58.66it/s]
55%		2289/4200 [00:38<00:32, 59.14it/s]
55%		2296/4200 [00:38<00:31, 61.11it/s]
55%		2303/4200 [00:38<00:33, 57.08it/s]
55%		2310/4200 [00:38<00:37, 50.95it/s]
55%		2316/4200 [00:39<01:06, 28.28it/s]
55%		2324/4200 [00:39<00:53, 34.96it/s]
56%		2331/4200 [00:39<00:45, 40.77it/s]
56%		2338/4200 [00:39<00:41, 45.32it/s]
56%		2347/4200 [00:39<00:35, 52.18it/s]
56%		2357/4200 [00:39<00:30, 60.48it/s]
56%		2365/4200 [00:39<00:31, 58.66it/s]
56%		2372/4200 [00:39<00:32, 56.83it/s]
57%		2380/4200 [00:39<00:29, 62.19it/s]
57%		2387/4200 [00:40<00:28, 63.43it/s]
57%		2394/4200 [00:40<00:28, 64.00it/s]
57%		2401/4200 [00:40<00:33, 54.38it/s]
57%		2407/4200 [00:40<00:32, 54.73it/s]
57%		2413/4200 [00:40<00:38, 46.22it/s]
58%		2422/4200 [00:40<00:35, 50.68it/s]
58%		2430/4200 [00:40<00:31, 55.97it/s]
58%		2437/4200 [00:40<00:29, 59.49it/s]
58%		2444/4200 [00:41<00:29, 59.00it/s]
58%		2451/4200 [00:41<00:36, 47.98it/s]
58%		2457/4200 [00:41<00:36, 47.23it/s]
59%		2463/4200 [00:41<00:35, 48.52it/s]
59%		2471/4200 [00:41<00:31, 54.10it/s]
59%		2480/4200 [00:41<00:28, 60.29it/s]
59%		2487/4200 [00:41<00:29, 57.10it/s]
59%		2494/4200 [00:42<00:33, 51.34it/s]
60%		2500/4200 [00:42<00:34, 48.92it/s]
60%		2511/4200 [00:42<00:29, 56.51it/s]
60%		2518/4200 [00:42<00:33, 50.25it/s]
60%		2525/4200 [00:42<00:30, 54.27it/s]
60%		2531/4200 [00:42<00:39, 42.78it/s]
60%		2537/4200 [00:42<00:36, 45.08it/s]
61%		2544/4200 [00:43<00:32, 50.44it/s]
61%		2550/4200 [00:43<00:31, 52.85it/s]
61%		2556/4200 [00:43<00:30, 53.96it/s]
61%		2564/4200 [00:43<00:27, 58.92it/s]
61%		2572/4200 [00:43<00:27, 60.02it/s]
61%		2579/4200 [00:43<00:28, 57.09it/s]

62%		2585/4200 [00:43<00:28, 56.14it/s]
62%		2594/4200 [00:44<00:36, 43.52it/s]
62%		2599/4200 [00:44<00:42, 37.79it/s]
62%		2610/4200 [00:44<00:34, 45.73it/s]
62%		2616/4200 [00:44<00:36, 43.03it/s]
62%		2623/4200 [00:44<00:34, 46.36it/s]
63%		2629/4200 [00:44<00:32, 47.98it/s]
63%		2635/4200 [00:44<00:34, 45.45it/s]
63%		2640/4200 [00:45<00:34, 45.57it/s]
63%		2645/4200 [00:45<00:34, 45.67it/s]
63%		2656/4200 [00:45<00:28, 54.68it/s]
63%		2664/4200 [00:45<00:25, 59.34it/s]
64%		2671/4200 [00:45<00:25, 61.15it/s]
64%		2678/4200 [00:45<00:24, 62.15it/s]
64%		2685/4200 [00:45<00:28, 53.22it/s]
64%		2691/4200 [00:45<00:28, 53.52it/s]
64%		2702/4200 [00:45<00:23, 62.90it/s]
65%		2710/4200 [00:46<00:22, 65.48it/s]
65%		2718/4200 [00:46<00:22, 65.56it/s]
65%		2726/4200 [00:46<00:22, 64.42it/s]
65%		2733/4200 [00:46<00:34, 41.99it/s]
65%		2742/4200 [00:46<00:29, 49.03it/s]
66%		2751/4200 [00:46<00:26, 54.61it/s]
66%		2760/4200 [00:46<00:23, 61.53it/s]
66%		2768/4200 [00:47<00:22, 64.93it/s]
66%		2776/4200 [00:47<00:22, 64.02it/s]
66%		2784/4200 [00:47<00:21, 66.07it/s]
66%		2792/4200 [00:47<00:24, 57.46it/s]
67%		2800/4200 [00:47<00:24, 56.39it/s]
67%		2809/4200 [00:47<00:22, 62.02it/s]
67%		2819/4200 [00:47<00:20, 66.63it/s]
67%		2827/4200 [00:48<00:23, 57.53it/s]
67%		2834/4200 [00:48<00:26, 51.16it/s]
68%		2843/4200 [00:48<00:23, 56.89it/s]
68%		2851/4200 [00:48<00:22, 60.97it/s]
68%		2862/4200 [00:48<00:19, 67.49it/s]
68%		2870/4200 [00:48<00:19, 69.04it/s]
69%		2878/4200 [00:48<00:20, 63.37it/s]
69%		2885/4200 [00:49<00:39, 33.24it/s]
69%		2895/4200 [00:49<00:32, 40.73it/s]
69%		2902/4200 [00:49<00:28, 44.96it/s]
69%		2911/4200 [00:49<00:24, 52.61it/s]
70%		2921/4200 [00:49<00:20, 61.30it/s]
70%		2929/4200 [00:49<00:19, 65.54it/s]
70%		2941/4200 [00:49<00:17, 73.63it/s]
70%		2950/4200 [00:50<00:16, 75.16it/s]
70%		2959/4200 [00:50<00:18, 66.12it/s]
71%		2967/4200 [00:50<00:24, 50.19it/s]
71%		2974/4200 [00:50<00:23, 51.37it/s]

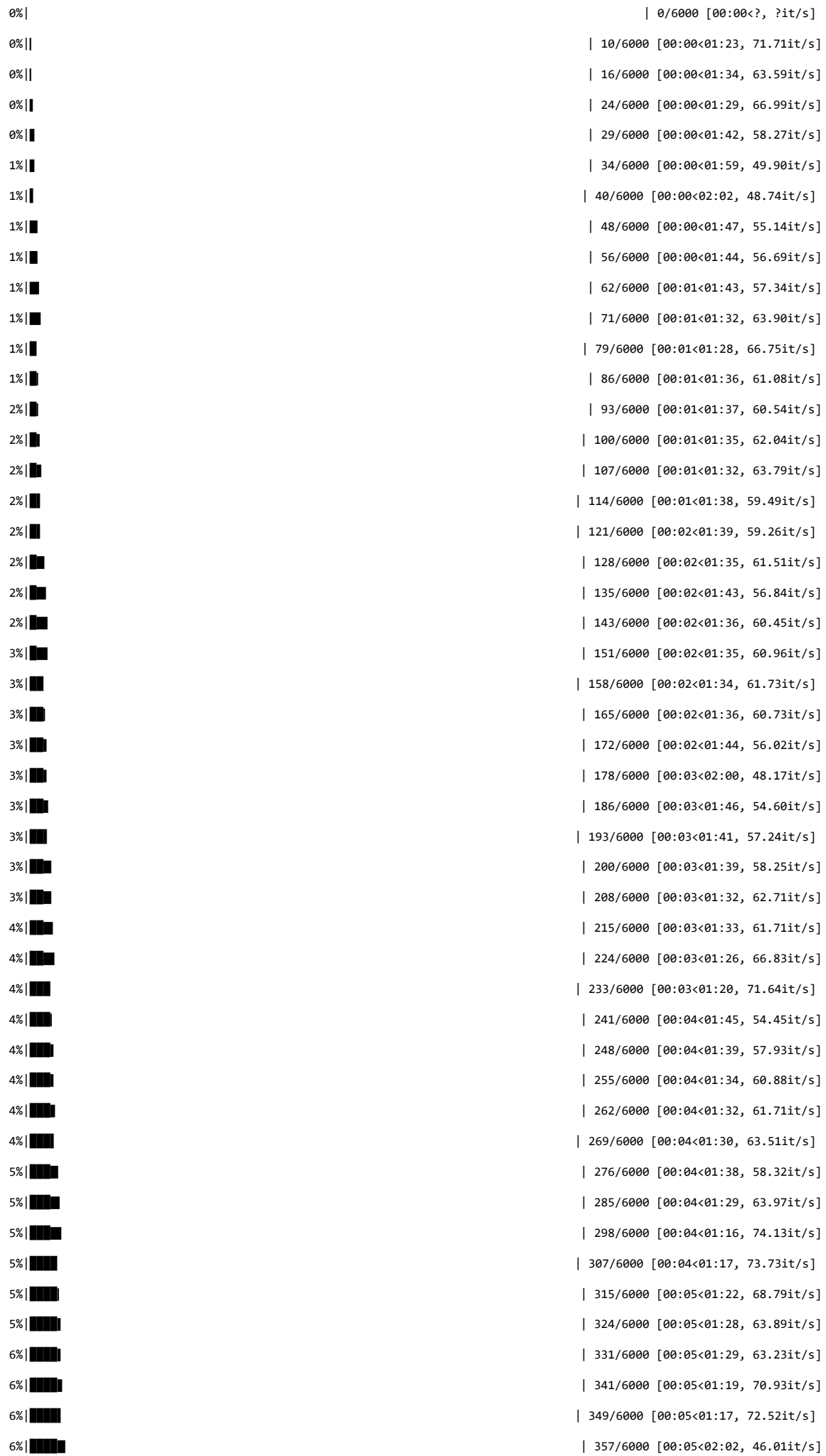
71%		2980/4200 [00:50<00:23, 53.01it/s]
71%		2986/4200 [00:50<00:22, 53.83it/s]
71%		2995/4200 [00:50<00:20, 60.25it/s]
72%		3003/4200 [00:51<00:18, 63.50it/s]
72%		3012/4200 [00:51<00:17, 68.71it/s]
72%		3023/4200 [00:51<00:15, 74.51it/s]
72%		3031/4200 [00:51<00:17, 66.50it/s]
72%		3039/4200 [00:51<00:19, 60.31it/s]
73%		3046/4200 [00:51<00:20, 56.17it/s]
73%		3052/4200 [00:51<00:24, 46.37it/s]
73%		3060/4200 [00:51<00:21, 52.74it/s]
73%		3069/4200 [00:52<00:19, 58.19it/s]
73%		3079/4200 [00:52<00:17, 65.58it/s]
74%		3087/4200 [00:52<00:16, 67.66it/s]
74%		3095/4200 [00:52<00:16, 65.35it/s]
74%		3105/4200 [00:52<00:17, 63.38it/s]
74%		3112/4200 [00:52<00:19, 55.41it/s]
74%		3119/4200 [00:52<00:19, 54.58it/s]
74%		3125/4200 [00:53<00:20, 51.60it/s]
75%		3134/4200 [00:53<00:18, 58.41it/s]
75%		3141/4200 [00:53<00:19, 55.66it/s]
75%		3149/4200 [00:53<00:17, 60.88it/s]
75%		3156/4200 [00:53<00:16, 62.54it/s]
75%		3164/4200 [00:53<00:15, 66.29it/s]
76%		3171/4200 [00:53<00:15, 64.81it/s]
76%		3178/4200 [00:53<00:16, 63.10it/s]
76%		3187/4200 [00:53<00:15, 66.14it/s]
76%		3194/4200 [00:54<00:15, 65.94it/s]
76%		3201/4200 [00:54<00:16, 59.51it/s]
77%		3214/4200 [00:54<00:14, 70.23it/s]
77%		3222/4200 [00:54<00:13, 71.36it/s]
77%		3233/4200 [00:54<00:12, 78.98it/s]
77%		3242/4200 [00:54<00:14, 68.40it/s]
77%		3254/4200 [00:54<00:12, 78.14it/s]
78%		3263/4200 [00:54<00:14, 66.00it/s]
78%		3271/4200 [00:55<00:15, 59.44it/s]
78%		3278/4200 [00:55<00:16, 55.26it/s]
78%		3285/4200 [00:55<00:16, 55.04it/s]
78%		3294/4200 [00:55<00:16, 55.81it/s]
79%		3300/4200 [00:55<00:16, 53.31it/s]
79%		3306/4200 [00:55<00:19, 44.93it/s]
79%		3311/4200 [00:56<00:19, 44.87it/s]
79%		3320/4200 [00:56<00:16, 52.17it/s]
79%		3330/4200 [00:56<00:14, 60.19it/s]
79%		3337/4200 [00:56<00:14, 59.62it/s]
80%		3344/4200 [00:56<00:15, 56.83it/s]
80%		3352/4200 [00:56<00:13, 61.59it/s]
80%		3359/4200 [00:56<00:15, 54.20it/s]

80%		3366/4200 [00:56<00:15, 53.06it/s]
80%		3372/4200 [00:57<00:17, 46.10it/s]
80%		3378/4200 [00:57<00:16, 48.42it/s]
81%		3384/4200 [00:57<00:18, 45.02it/s]
81%		3393/4200 [00:57<00:16, 49.16it/s]
81%		3401/4200 [00:57<00:14, 55.14it/s]
81%		3408/4200 [00:57<00:16, 48.34it/s]
81%		3418/4200 [00:57<00:14, 54.58it/s]
82%		3428/4200 [00:57<00:12, 62.94it/s]
82%		3436/4200 [00:58<00:16, 47.35it/s]
82%		3445/4200 [00:58<00:13, 54.84it/s]
82%		3452/4200 [00:58<00:14, 53.35it/s]
82%		3462/4200 [00:58<00:12, 60.18it/s]
83%		3470/4200 [00:58<00:11, 64.80it/s]
83%		3478/4200 [00:59<00:16, 44.45it/s]
83%		3484/4200 [00:59<00:18, 39.66it/s]
83%		3490/4200 [00:59<00:18, 37.98it/s]
83%		3498/4200 [00:59<00:15, 44.23it/s]
84%		3507/4200 [00:59<00:13, 50.72it/s]
84%		3519/4200 [00:59<00:11, 61.10it/s]
84%		3527/4200 [00:59<00:11, 61.09it/s]
84%		3535/4200 [00:59<00:11, 59.29it/s]
84%		3545/4200 [01:00<00:09, 66.06it/s]
85%		3553/4200 [01:00<00:11, 57.12it/s]
85%		3561/4200 [01:00<00:10, 61.01it/s]
85%		3568/4200 [01:00<00:11, 57.26it/s]
85%		3577/4200 [01:00<00:09, 62.78it/s]
85%		3586/4200 [01:00<00:08, 68.44it/s]
86%		3597/4200 [01:00<00:07, 76.71it/s]
86%		3606/4200 [01:01<00:09, 64.52it/s]
86%		3614/4200 [01:01<00:08, 68.27it/s]
86%		3622/4200 [01:01<00:08, 69.89it/s]
86%		3630/4200 [01:01<00:10, 55.54it/s]
87%		3638/4200 [01:01<00:09, 61.14it/s]
87%		3645/4200 [01:01<00:08, 62.82it/s]
87%		3652/4200 [01:01<00:09, 56.32it/s]
87%		3659/4200 [01:02<00:11, 46.05it/s]
87%		3668/4200 [01:02<00:10, 52.78it/s]
88%		3675/4200 [01:02<00:09, 53.41it/s]
88%		3686/4200 [01:02<00:08, 61.91it/s]
88%		3695/4200 [01:02<00:07, 63.91it/s]
88%		3703/4200 [01:02<00:07, 65.69it/s]
88%		3711/4200 [01:02<00:07, 62.03it/s]
89%		3718/4200 [01:02<00:08, 57.23it/s]
89%		3725/4200 [01:03<00:09, 52.60it/s]
89%		3735/4200 [01:03<00:07, 59.97it/s]
89%		3742/4200 [01:03<00:09, 48.41it/s]
89%		3749/4200 [01:03<00:08, 52.18it/s]
89%		3755/4200 [01:03<00:08, 53.49it/s]

90%		3761/4200 [01:03<00:08, 50.31it/s]
90%		3770/4200 [01:03<00:07, 56.17it/s]
90%		3777/4200 [01:04<00:07, 56.40it/s]
90%		3786/4200 [01:04<00:06, 62.42it/s]
90%		3793/4200 [01:04<00:06, 60.69it/s]
90%		3800/4200 [01:04<00:06, 59.14it/s]
91%		3807/4200 [01:04<00:07, 55.81it/s]
91%		3813/4200 [01:04<00:07, 50.46it/s]
91%		3825/4200 [01:04<00:06, 60.27it/s]
91%		3834/4200 [01:04<00:05, 66.30it/s]
91%		3842/4200 [01:05<00:06, 54.05it/s]
92%		3851/4200 [01:05<00:05, 60.34it/s]
92%		3860/4200 [01:05<00:05, 65.08it/s]
92%		3868/4200 [01:05<00:05, 59.32it/s]
92%		3875/4200 [01:05<00:05, 55.18it/s]
92%		3882/4200 [01:05<00:05, 53.13it/s]
93%		3891/4200 [01:05<00:05, 59.26it/s]
93%		3900/4200 [01:05<00:04, 61.50it/s]
93%		3909/4200 [01:06<00:04, 66.64it/s]
93%		3917/4200 [01:06<00:04, 62.05it/s]
93%		3926/4200 [01:06<00:04, 68.30it/s]
94%		3934/4200 [01:06<00:04, 58.00it/s]
94%		3941/4200 [01:06<00:04, 53.16it/s]
94%		3947/4200 [01:06<00:04, 54.02it/s]
94%		3953/4200 [01:06<00:04, 55.42it/s]
94%		3961/4200 [01:07<00:03, 60.72it/s]
94%		3968/4200 [01:07<00:03, 61.49it/s]
95%		3976/4200 [01:07<00:03, 65.21it/s]
95%		3983/4200 [01:07<00:03, 61.66it/s]
95%		3990/4200 [01:07<00:04, 49.36it/s]
95%		3996/4200 [01:07<00:04, 44.86it/s]
95%		4001/4200 [01:07<00:05, 39.48it/s]
95%		4010/4200 [01:08<00:04, 45.79it/s]
96%		4016/4200 [01:08<00:03, 47.65it/s]
96%		4023/4200 [01:08<00:03, 52.24it/s]
96%		4029/4200 [01:08<00:03, 52.96it/s]
96%		4035/4200 [01:08<00:03, 54.88it/s]
96%		4041/4200 [01:08<00:03, 49.89it/s]
96%		4050/4200 [01:08<00:02, 55.52it/s]
97%		4057/4200 [01:08<00:02, 55.49it/s]
97%		4064/4200 [01:08<00:02, 56.39it/s]
97%		4070/4200 [01:09<00:02, 52.71it/s]
97%		4079/4200 [01:09<00:02, 59.82it/s]
97%		4086/4200 [01:09<00:01, 59.25it/s]
98%		4096/4200 [01:09<00:01, 66.37it/s]
98%		4104/4200 [01:09<00:01, 65.96it/s]
98%		4111/4200 [01:09<00:01, 62.55it/s]
98%		4119/4200 [01:09<00:01, 66.88it/s]
















































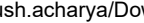















































```
In [111]: start2 = time.clock()
t = tf_idf_vect.get_feature_names()
tfidf_sent_vectors_test = []; # the tfidf-w2v for each sentence/review is stored in this list
row=0;
for sent in tqdm(list_sent_test):
    sent_vec = np.zeros(50)
    cnt_words = 0; # num of words with a valid vector in the sentence/review
    for word in sent: # for each word in a review/sentence
        if word in w2v_words:
            vec = w2v_model.wv[word]
            tfidf = tfidf_test[row,t.index(word)]
            sent_vec += (vec * tfidf)
            cnt_words += tfidf
    if cnt_words != 0:
        sent_vec /= cnt_words
    tfidf_sent_vectors_test.append(sent_vec)
    row += 1
print(len(tfidf_sent_vectors_test))
print(len(tfidf_sent_vectors_test[0]))
print((time.clock()-start1)/60)
```



6%	<div><div></div></div>	364/6000 [00:05<01:56, 48.55it/s]
6%	<div><div></div></div>	372/6000 [00:06<01:42, 54.75it/s]
6%	<div><div></div></div>	379/6000 [00:06<01:40, 55.84it/s]
7%	<div><div></div></div>	391/6000 [00:06<01:24, 66.34it/s]
7%	<div><div></div></div>	399/6000 [00:06<01:22, 67.51it/s]
7%	<div><div></div></div>	407/6000 [00:06<01:32, 60.43it/s]
7%	<div><div></div></div>	414/6000 [00:06<01:32, 60.14it/s]
7%	<div><div></div></div>	421/6000 [00:06<01:32, 60.57it/s]
7%	<div><div></div></div>	433/6000 [00:06<01:18, 70.55it/s]
7%	<div><div></div></div>	441/6000 [00:07<01:25, 64.73it/s]
8%	<div><div></div></div>	452/6000 [00:07<01:16, 72.92it/s]
8%	<div><div></div></div>	461/6000 [00:07<01:18, 70.16it/s]
8%	<div><div></div></div>	469/6000 [00:07<01:20, 69.07it/s]
8%	<div><div></div></div>	479/6000 [00:07<01:13, 75.34it/s]
8%	<div><div></div></div>	488/6000 [00:07<01:18, 69.91it/s]
8%	<div><div></div></div>	496/6000 [00:07<01:23, 65.58it/s]
8%	<div><div></div></div>	503/6000 [00:07<01:27, 62.50it/s]
8%	<div><div></div></div>	510/6000 [00:08<01:32, 59.16it/s]
9%	<div><div></div></div>	518/6000 [00:08<01:26, 63.47it/s]
9%	<div><div></div></div>	529/6000 [00:08<01:16, 71.95it/s]
9%	<div><div></div></div>	537/6000 [00:08<01:23, 65.81it/s]
9%	<div><div></div></div>	545/6000 [00:08<01:33, 58.06it/s]
9%	<div><div></div></div>	552/6000 [00:08<01:33, 58.54it/s]
9%	<div><div></div></div>	560/6000 [00:08<01:33, 57.96it/s]
9%	<div><div></div></div>	567/6000 [00:09<01:41, 53.78it/s]
10%	<div><div></div></div>	575/6000 [00:09<01:40, 54.07it/s]
10%	<div><div></div></div>	583/6000 [00:09<01:31, 58.97it/s]
10%	<div><div></div></div>	593/6000 [00:09<01:20, 67.02it/s]
10%	<div><div></div></div>	601/6000 [00:09<01:45, 50.97it/s]
10%	<div><div></div></div>	609/6000 [00:09<01:35, 56.18it/s]
10%	<div><div></div></div>	620/6000 [00:09<01:23, 64.40it/s]
10%	<div><div></div></div>	628/6000 [00:10<01:48, 49.29it/s]
11%	<div><div></div></div>	637/6000 [00:10<01:36, 55.54it/s]
11%	<div><div></div></div>	644/6000 [00:10<01:39, 54.10it/s]
11%	<div><div></div></div>	653/6000 [00:10<01:27, 60.90it/s]
11%	<div><div></div></div>	661/6000 [00:10<01:22, 64.35it/s]
11%	<div><div></div></div>	669/6000 [00:10<01:21, 65.48it/s]
11%	<div><div></div></div>	677/6000 [00:10<01:35, 55.51it/s]
11%	<div><div></div></div>	684/6000 [00:10<01:31, 57.90it/s]
12%	<div><div></div></div>	692/6000 [00:11<01:33, 57.04it/s]
12%	<div><div></div></div>	699/6000 [00:11<01:37, 54.58it/s]
12%	<div><div></div></div>	706/6000 [00:11<01:33, 56.36it/s]
12%	<div><div></div></div>	712/6000 [00:11<02:09, 40.73it/s]
12%	<div><div></div></div>	721/6000 [00:11<01:51, 47.27it/s]
12%	<div><div></div></div>	727/6000 [00:11<02:06, 41.81it/s]
12%	<div><div></div></div>	732/6000 [00:12<02:27, 35.65it/s]
12%	<div><div></div></div>	737/6000 [00:12<02:32, 34.47it/s]
12%	<div><div></div></div>	743/6000 [00:12<02:13, 39.51it/s]
12%	<div><div></div></div>	749/6000 [00:12<02:14, 39.16it/s]

13%	<div></div>	754/6000 [00:12<02:27, 35.58it/s]
13%	<div></div>	759/6000 [00:12<02:14, 38.86it/s]
13%	<div></div>	766/6000 [00:12<02:05, 41.86it/s]
13%	<div></div>	771/6000 [00:13<02:03, 42.24it/s]
13%	<div></div>	778/6000 [00:13<01:49, 47.67it/s]
13%	<div></div>	787/6000 [00:13<01:40, 51.94it/s]
13%	<div></div>	796/6000 [00:13<01:29, 58.14it/s]
13%	<div></div>	805/6000 [00:13<01:21, 63.96it/s]
14%	<div></div>	812/6000 [00:13<01:31, 56.49it/s]
14%	<div></div>	819/6000 [00:13<01:33, 55.40it/s]
14%	<div></div>	825/6000 [00:13<01:36, 53.47it/s]
14%	<div></div>	831/6000 [00:14<01:36, 53.50it/s]
14%	<div></div>	837/6000 [00:14<01:33, 55.02it/s]
14%	<div></div>	843/6000 [00:14<01:35, 53.83it/s]
14%	<div></div>	849/6000 [00:14<01:41, 50.79it/s]
14%	<div></div>	855/6000 [00:14<01:40, 51.24it/s]
14%	<div></div>	861/6000 [00:14<01:56, 44.23it/s]
14%	<div></div>	867/6000 [00:14<01:47, 47.81it/s]
15%	<div></div>	873/6000 [00:14<01:43, 49.77it/s]
15%	<div></div>	879/6000 [00:15<01:39, 51.59it/s]
15%	<div></div>	885/6000 [00:15<01:35, 53.33it/s]
15%	<div></div>	891/6000 [00:15<01:54, 44.71it/s]
15%	<div></div>	896/6000 [00:15<02:03, 41.22it/s]
15%	<div></div>	901/6000 [00:15<01:58, 42.99it/s]
15%	<div></div>	906/6000 [00:15<01:55, 44.06it/s]
15%	<div></div>	914/6000 [00:15<01:41, 50.35it/s]
15%	<div></div>	920/6000 [00:15<01:46, 47.76it/s]
15%	<div></div>	926/6000 [00:16<02:05, 40.54it/s]
16%	<div></div>	932/6000 [00:16<01:53, 44.64it/s]
16%	<div></div>	939/6000 [00:16<01:43, 48.93it/s]
16%	<div></div>	945/6000 [00:16<01:57, 43.07it/s]
16%	<div></div>	955/6000 [00:16<01:37, 51.56it/s]
16%	<div></div>	963/6000 [00:16<01:27, 57.63it/s]
16%	<div></div>	970/6000 [00:16<01:28, 57.16it/s]
16%	<div></div>	978/6000 [00:16<01:20, 62.13it/s]
16%	<div></div>	985/6000 [00:17<01:32, 54.33it/s]
17%	<div></div>	993/6000 [00:17<01:24, 59.54it/s]
17%	<div></div>	1002/6000 [00:17<01:16, 65.29it/s]
17%	<div></div>	1013/6000 [00:17<01:07, 73.38it/s]
17%	<div></div>	1022/6000 [00:17<01:08, 72.81it/s]
17%	<div></div>	1030/6000 [00:17<01:10, 70.09it/s]
17%	<div></div>	1038/6000 [00:17<01:09, 71.53it/s]
17%	<div></div>	1046/6000 [00:17<01:31, 54.04it/s]
18%	<div></div>	1055/6000 [00:18<01:23, 59.20it/s]
18%	<div></div>	1062/6000 [00:18<01:22, 59.90it/s]
18%	<div></div>	1073/6000 [00:18<01:13, 67.43it/s]
18%	<div></div>	1081/6000 [00:18<01:17, 63.17it/s]
18%	<div></div>	1088/6000 [00:18<01:22, 59.75it/s]

18%		1096/6000 [00:18<01:15, 64.57it/s]
18%		1103/6000 [00:18<01:20, 60.56it/s]
19%		1113/6000 [00:18<01:11, 68.58it/s]
19%		1121/6000 [00:19<01:17, 62.90it/s]
19%		1128/6000 [00:19<01:53, 42.93it/s]
19%		1137/6000 [00:19<01:38, 49.31it/s]
19%		1147/6000 [00:19<01:25, 56.46it/s]
19%		1154/6000 [00:19<01:41, 47.91it/s]
19%		1160/6000 [00:19<01:40, 48.37it/s]
19%		1166/6000 [00:20<01:40, 48.09it/s]
20%		1173/6000 [00:20<01:31, 52.56it/s]
20%		1179/6000 [00:20<01:28, 54.53it/s]
20%		1188/6000 [00:20<01:19, 60.70it/s]
20%		1197/6000 [00:20<01:16, 62.92it/s]
20%		1204/6000 [00:20<01:16, 63.04it/s]
20%		1211/6000 [00:20<01:16, 62.45it/s]
20%		1218/6000 [00:21<01:45, 45.15it/s]
20%		1227/6000 [00:21<01:30, 52.57it/s]
21%		1236/6000 [00:21<01:19, 59.83it/s]
21%		1244/6000 [00:21<01:21, 58.61it/s]
21%		1251/6000 [00:21<01:22, 57.39it/s]
21%		1263/6000 [00:21<01:10, 67.25it/s]
21%		1271/6000 [00:21<01:08, 69.06it/s]
21%		1279/6000 [00:21<01:12, 65.10it/s]
21%		1287/6000 [00:22<01:23, 56.33it/s]
22%		1294/6000 [00:22<01:22, 56.98it/s]
22%		1301/6000 [00:22<01:34, 49.95it/s]
22%		1312/6000 [00:22<01:20, 58.40it/s]
22%		1321/6000 [00:22<01:12, 64.50it/s]
22%		1329/6000 [00:22<01:11, 65.45it/s]
22%		1337/6000 [00:22<01:10, 65.87it/s]
22%		1344/6000 [00:23<01:46, 43.66it/s]
22%		1350/6000 [00:23<01:47, 43.33it/s]
23%		1356/6000 [00:23<03:17, 23.57it/s]
23%		1360/6000 [00:23<02:55, 26.44it/s]
23%		1364/6000 [00:23<02:39, 29.10it/s]
23%		1370/6000 [00:24<02:15, 34.18it/s]
23%		1376/6000 [00:24<01:58, 39.08it/s]
23%		1386/6000 [00:24<01:37, 47.15it/s]
23%		1394/6000 [00:24<01:25, 53.72it/s]
23%		1401/6000 [00:24<01:20, 57.26it/s]
24%		1411/6000 [00:24<01:13, 62.02it/s]
24%		1419/6000 [00:24<01:14, 61.50it/s]
24%		1426/6000 [00:24<01:12, 63.19it/s]
24%		1433/6000 [00:24<01:18, 57.90it/s]
24%		1440/6000 [00:25<01:24, 53.90it/s]
24%		1449/6000 [00:25<01:16, 59.74it/s]
24%		1456/6000 [00:25<01:32, 49.10it/s]
24%		1463/6000 [00:25<01:24, 53.91it/s]

24%		1469/6000 [00:25<01:25, 52.70it/s]
25%		1479/6000 [00:25<01:13, 61.40it/s]
25%		1487/6000 [00:25<01:09, 65.30it/s]
25%		1495/6000 [00:26<01:22, 54.47it/s]
25%		1506/6000 [00:26<01:10, 63.54it/s]
25%		1514/6000 [00:26<01:24, 52.92it/s]
25%		1521/6000 [00:26<01:21, 55.04it/s]
26%		1530/6000 [00:26<01:11, 62.11it/s]
26%		1542/6000 [00:26<01:01, 72.57it/s]
26%		1551/6000 [00:26<01:06, 66.98it/s]
26%		1560/6000 [00:27<01:11, 61.97it/s]
26%		1567/6000 [00:27<01:12, 60.90it/s]
26%		1574/6000 [00:27<01:25, 52.06it/s]
26%		1580/6000 [00:27<01:22, 53.59it/s]
26%		1589/6000 [00:27<01:13, 60.27it/s]
27%		1598/6000 [00:27<01:07, 65.38it/s]
27%		1606/6000 [00:27<01:06, 66.57it/s]
27%		1614/6000 [00:27<01:07, 64.58it/s]
27%		1621/6000 [00:28<01:11, 61.60it/s]
27%		1628/6000 [00:28<01:14, 58.76it/s]
27%		1638/6000 [00:28<01:05, 66.52it/s]
27%		1647/6000 [00:28<01:00, 72.09it/s]
28%		1655/6000 [00:28<01:06, 65.58it/s]
28%		1662/6000 [00:28<01:14, 58.22it/s]
28%		1669/6000 [00:28<01:19, 54.17it/s]
28%		1675/6000 [00:28<01:27, 49.71it/s]
28%		1681/6000 [00:29<01:23, 51.83it/s]
28%		1687/6000 [00:29<01:30, 47.42it/s]
28%		1694/6000 [00:29<01:22, 52.47it/s]
28%		1704/6000 [00:29<01:10, 60.76it/s]
29%		1712/6000 [00:29<01:06, 64.82it/s]
29%		1721/6000 [00:29<01:02, 68.79it/s]
29%		1732/6000 [00:29<00:56, 76.16it/s]
29%		1741/6000 [00:29<00:54, 77.46it/s]
29%		1750/6000 [00:30<01:03, 66.47it/s]
29%		1758/6000 [00:30<01:13, 57.37it/s]
29%		1766/6000 [00:30<01:08, 61.86it/s]
30%		1773/6000 [00:30<01:22, 51.25it/s]
30%		1779/6000 [00:30<01:24, 49.96it/s]
30%		1787/6000 [00:30<01:18, 53.44it/s]
30%		1796/6000 [00:30<01:09, 60.47it/s]
30%		1803/6000 [00:30<01:07, 62.51it/s]
30%		1815/6000 [00:31<00:58, 71.73it/s]
30%		1823/6000 [00:31<00:57, 72.21it/s]
31%		1835/6000 [00:31<00:54, 76.33it/s]
31%		1844/6000 [00:31<01:09, 59.87it/s]
31%		1851/6000 [00:31<01:11, 58.08it/s]
31%		1858/6000 [00:31<01:07, 61.08it/s]















31%		1865/6000 [00:31<01:11, 58.24it/s]
31%		1872/6000 [00:32<01:20, 51.11it/s]
31%		1880/6000 [00:32<01:12, 57.17it/s]
32%		1890/6000 [00:32<01:04, 63.30it/s]
32%		1898/6000 [00:32<01:10, 58.05it/s]
32%		1909/6000 [00:32<01:01, 66.28it/s]
32%		1918/6000 [00:32<00:58, 69.34it/s]
32%		1926/6000 [00:32<01:03, 64.64it/s]
32%		1933/6000 [00:32<01:04, 63.13it/s]
32%		1940/6000 [00:33<01:08, 59.57it/s]
32%		1947/6000 [00:33<01:08, 59.21it/s]
33%		1954/6000 [00:33<01:16, 52.58it/s]
33%		1963/6000 [00:33<01:08, 58.55it/s]
33%		1970/6000 [00:33<01:13, 54.97it/s]
33%		1979/6000 [00:33<01:06, 60.39it/s]
33%		1988/6000 [00:33<01:00, 66.48it/s]
33%		1997/6000 [00:33<00:56, 71.04it/s]
33%		2005/6000 [00:34<00:56, 71.01it/s]
34%		2013/6000 [00:34<01:03, 62.52it/s]
34%		2020/6000 [00:34<01:07, 58.95it/s]
34%		2027/6000 [00:34<01:32, 42.89it/s]
34%		2036/6000 [00:34<01:18, 50.61it/s]
34%		2045/6000 [00:34<01:09, 56.95it/s]
34%		2052/6000 [00:34<01:09, 56.94it/s]
34%		2061/6000 [00:35<01:03, 62.49it/s]
34%		2068/6000 [00:35<01:14, 52.95it/s]
35%		2077/6000 [00:35<01:05, 59.54it/s]
35%		2087/6000 [00:35<01:00, 65.04it/s]
35%		2095/6000 [00:35<01:01, 63.55it/s]
35%		2102/6000 [00:35<01:08, 57.07it/s]
35%		2112/6000 [00:35<01:00, 64.42it/s]
35%		2120/6000 [00:36<01:09, 55.74it/s]
35%		2127/6000 [00:36<01:13, 52.94it/s]
36%		2135/6000 [00:36<01:06, 58.39it/s]
36%		2142/6000 [00:36<01:18, 49.43it/s]
36%		2150/6000 [00:36<01:11, 54.20it/s]
36%		2156/6000 [00:36<01:10, 54.27it/s]
36%		2162/6000 [00:36<01:13, 52.51it/s]
36%		2172/6000 [00:36<01:03, 60.29it/s]
36%		2180/6000 [00:37<00:59, 64.73it/s]
36%		2187/6000 [00:37<01:01, 61.66it/s]
37%		2196/6000 [00:37<00:56, 67.68it/s]
37%		2204/6000 [00:37<01:09, 55.00it/s]
37%		2211/6000 [00:37<01:08, 55.62it/s]
37%		2218/6000 [00:37<01:10, 53.41it/s]
37%		2225/6000 [00:37<01:07, 56.14it/s]
37%		2235/6000 [00:38<00:58, 63.85it/s]
37%		2242/6000 [00:38<00:58, 64.49it/s]
38%		2252/6000 [00:38<00:52, 71.87it/s]

38%		2260/6000 [00:38<00:57, 64.74it/s]
38%		2269/6000 [00:38<00:52, 70.52it/s]
38%		2277/6000 [00:38<00:59, 62.08it/s]
38%		2286/6000 [00:38<00:54, 68.27it/s]
38%		2295/6000 [00:38<00:52, 70.92it/s]
38%		2303/6000 [00:39<00:59, 62.46it/s]
39%		2313/6000 [00:39<00:52, 70.31it/s]
39%		2321/6000 [00:39<00:51, 71.53it/s]
39%		2329/6000 [00:39<00:53, 68.39it/s]
39%		2338/6000 [00:39<00:49, 73.68it/s]
39%		2346/6000 [00:39<00:52, 69.78it/s]
39%		2355/6000 [00:39<00:51, 70.54it/s]
39%		2363/6000 [00:39<01:00, 60.56it/s]
40%		2371/6000 [00:39<00:58, 62.48it/s]
40%		2380/6000 [00:40<00:56, 64.45it/s]
40%		2391/6000 [00:40<00:50, 71.77it/s]
40%		2399/6000 [00:40<00:51, 70.56it/s]
40%		2407/6000 [00:40<01:06, 53.96it/s]
40%		2417/6000 [00:40<00:57, 62.27it/s]
40%		2425/6000 [00:40<01:04, 55.68it/s]
41%		2432/6000 [00:41<01:08, 51.85it/s]
41%		2440/6000 [00:41<01:01, 57.86it/s]
41%		2447/6000 [00:41<00:58, 60.78it/s]
41%		2454/6000 [00:41<00:59, 60.04it/s]
41%		2461/6000 [00:41<01:02, 56.55it/s]
41%		2468/6000 [00:41<00:59, 59.22it/s]
41%		2479/6000 [00:41<00:52, 67.35it/s]
41%		2487/6000 [00:41<00:49, 70.66it/s]
42%		2496/6000 [00:41<00:46, 74.79it/s]
42%		2507/6000 [00:42<00:42, 81.89it/s]
42%		2516/6000 [00:42<00:46, 75.23it/s]
42%		2525/6000 [00:42<00:44, 77.43it/s]
42%		2537/6000 [00:42<00:40, 85.32it/s]
42%		2546/6000 [00:42<00:56, 60.96it/s]
43%		2555/6000 [00:42<00:51, 66.33it/s]
43%		2563/6000 [00:42<01:05, 52.82it/s]
43%		2570/6000 [00:43<01:09, 49.34it/s]
43%		2581/6000 [00:43<00:59, 57.89it/s]
43%		2588/6000 [00:43<01:09, 49.15it/s]
43%		2594/6000 [00:43<01:13, 46.27it/s]
43%		2603/6000 [00:43<01:03, 53.61it/s]
44%		2610/6000 [00:43<01:12, 46.91it/s]
44%		2616/6000 [00:43<01:09, 49.00it/s]
44%		2624/6000 [00:44<01:05, 51.31it/s]
44%		2630/6000 [00:44<01:13, 45.87it/s]
44%		2639/6000 [00:44<01:05, 51.62it/s]
44%		2646/6000 [00:44<01:00, 55.41it/s]
44%		2652/6000 [00:44<01:04, 52.28it/s]












44%		2659/6000 [00:44<01:02, 53.70it/s]
44%		2665/6000 [00:44<01:05, 51.15it/s]
45%		2673/6000 [00:44<00:58, 56.93it/s]
45%		2680/6000 [00:45<00:59, 55.51it/s]
45%		2688/6000 [00:45<01:00, 54.52it/s]
45%		2700/6000 [00:45<00:50, 64.96it/s]
45%		2708/6000 [00:45<00:47, 68.64it/s]
45%		2716/6000 [00:45<01:11, 46.11it/s]
45%		2723/6000 [00:45<01:14, 43.72it/s]
45%		2729/6000 [00:46<01:11, 45.86it/s]
46%		2735/6000 [00:46<01:08, 47.99it/s]
46%		2742/6000 [00:46<01:03, 51.01it/s]
46%		2748/6000 [00:46<01:02, 51.89it/s]
46%		2756/6000 [00:46<00:56, 57.50it/s]
46%		2763/6000 [00:46<00:54, 59.04it/s]
46%		2770/6000 [00:46<00:58, 55.59it/s]
46%		2779/6000 [00:46<00:51, 62.15it/s]
46%		2786/6000 [00:47<00:52, 60.72it/s]
47%		2796/6000 [00:47<00:47, 67.30it/s]
47%		2804/6000 [00:47<00:45, 69.58it/s]
47%		2813/6000 [00:47<00:43, 72.74it/s]
47%		2821/6000 [00:47<00:47, 66.78it/s]
47%		2828/6000 [00:47<00:51, 61.91it/s]
47%		2836/6000 [00:47<00:49, 63.62it/s]
47%		2843/6000 [00:47<00:51, 61.19it/s]
48%		2850/6000 [00:47<00:52, 59.52it/s]
48%		2859/6000 [00:48<00:52, 60.08it/s]
48%		2866/6000 [00:48<00:59, 52.52it/s]
48%		2874/6000 [00:48<00:54, 57.32it/s]
48%		2881/6000 [00:48<00:57, 54.34it/s]
48%		2887/6000 [00:48<01:00, 51.38it/s]
48%		2893/6000 [00:48<00:58, 53.40it/s]
48%		2900/6000 [00:48<00:57, 54.36it/s]
48%		2909/6000 [00:49<00:50, 61.18it/s]
49%		2916/6000 [00:49<00:56, 54.51it/s]
49%		2924/6000 [00:49<00:51, 60.21it/s]
49%		2934/6000 [00:49<00:45, 66.93it/s]
49%		2943/6000 [00:49<00:43, 70.44it/s]
49%		2952/6000 [00:49<00:40, 74.55it/s]
49%		2960/6000 [00:49<00:46, 64.69it/s]
49%		2968/6000 [00:49<00:44, 67.86it/s]
50%		2978/6000 [00:49<00:40, 73.76it/s]
50%		2986/6000 [00:50<00:40, 73.74it/s]
50%		2994/6000 [00:50<00:41, 73.19it/s]
50%		3002/6000 [00:50<00:46, 64.28it/s]
50%		3010/6000 [00:50<00:44, 67.57it/s]
50%		3018/6000 [00:50<00:42, 70.15it/s]
50%		3026/6000 [00:50<00:56, 52.76it/s]
51%		3033/6000 [00:51<01:06, 44.83it/s]

80/92








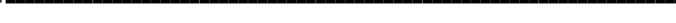
































57%	<div></div>	3426/6000 [00:57<00:51, 49.94it/s]
57%	<div></div>	3432/6000 [00:57<01:11, 35.67it/s]
57%	<div></div>	3439/6000 [00:57<01:09, 36.72it/s]
57%	<div></div>	3447/6000 [00:58<00:58, 43.73it/s]
58%	<div></div>	3454/6000 [00:58<00:51, 49.21it/s]
58%	<div></div>	3460/6000 [00:58<00:49, 51.69it/s]
58%	<div></div>	3470/6000 [00:58<00:42, 59.48it/s]
58%	<div></div>	3477/6000 [00:58<00:42, 59.49it/s]
58%	<div></div>	3484/6000 [00:58<00:41, 60.97it/s]
58%	<div></div>	3491/6000 [00:58<00:45, 55.53it/s]
58%	<div></div>	3498/6000 [00:58<00:42, 58.33it/s]
58%	<div></div>	3506/6000 [00:58<00:39, 63.41it/s]
59%	<div></div>	3513/6000 [00:59<00:46, 53.15it/s]
59%	<div></div>	3519/6000 [00:59<00:55, 45.07it/s]
59%	<div></div>	3525/6000 [00:59<00:55, 44.33it/s]
59%	<div></div>	3532/6000 [00:59<00:49, 49.64it/s]
59%	<div></div>	3542/6000 [00:59<00:42, 58.44it/s]
59%	<div></div>	3549/6000 [00:59<00:48, 50.82it/s]
59%	<div></div>	3558/6000 [00:59<00:44, 54.96it/s]
59%	<div></div>	3565/6000 [01:00<00:43, 56.38it/s]
60%	<div></div>	3572/6000 [01:00<00:58, 41.34it/s]
60%	<div></div>	3582/6000 [01:00<00:48, 49.42it/s]
60%	<div></div>	3591/6000 [01:00<00:42, 56.25it/s]
60%	<div></div>	3598/6000 [01:00<00:47, 50.13it/s]
60%	<div></div>	3607/6000 [01:00<00:42, 55.69it/s]
60%	<div></div>	3615/6000 [01:00<00:39, 59.94it/s]
60%	<div></div>	3622/6000 [01:01<00:40, 59.39it/s]
60%	<div></div>	3630/6000 [01:01<00:38, 62.30it/s]
61%	<div></div>	3639/6000 [01:01<00:34, 67.51it/s]
61%	<div></div>	3647/6000 [01:01<00:34, 68.52it/s]
61%	<div></div>	3655/6000 [01:01<00:39, 59.19it/s]
61%	<div></div>	3662/6000 [01:01<00:37, 62.03it/s]
61%	<div></div>	3669/6000 [01:01<00:38, 60.48it/s]
61%	<div></div>	3676/6000 [01:02<00:45, 50.99it/s]
61%	<div></div>	3685/6000 [01:02<00:43, 53.13it/s]
62%	<div></div>	3691/6000 [01:02<00:42, 54.68it/s]
62%	<div></div>	3697/6000 [01:02<00:43, 52.53it/s]
62%	<div></div>	3703/6000 [01:02<00:53, 43.25it/s]
62%	<div></div>	3709/6000 [01:02<00:49, 46.54it/s]
62%	<div></div>	3715/6000 [01:02<00:50, 45.51it/s]
62%	<div></div>	3725/6000 [01:02<00:43, 51.79it/s]
62%	<div></div>	3732/6000 [01:03<00:44, 50.95it/s]
62%	<div></div>	3738/6000 [01:03<00:51, 43.87it/s]
62%	<div></div>	3743/6000 [01:03<01:02, 36.38it/s]
62%	<div></div>	3750/6000 [01:03<00:54, 41.31it/s]
63%	<div></div>	3759/6000 [01:03<00:45, 48.75it/s]
63%	<div></div>	3773/6000 [01:03<00:37, 60.04it/s]
63%	<div></div>	3781/6000 [01:03<00:37, 58.47it/s]
63%	<div></div>	3789/6000 [01:04<00:41, 53.66it/s]

63%		3797/6000 [01:04<00:37, 59.43it/s]
63%		3806/6000 [01:04<00:34, 64.46it/s]
64%		3814/6000 [01:04<00:33, 66.16it/s]
64%		3823/6000 [01:04<00:30, 71.53it/s]
64%		3831/6000 [01:04<00:30, 70.28it/s]
64%		3839/6000 [01:04<00:30, 70.50it/s]
64%		3847/6000 [01:04<00:30, 69.94it/s]
64%		3855/6000 [01:05<00:33, 64.29it/s]
64%		3866/6000 [01:05<00:29, 72.10it/s]
65%		3874/6000 [01:05<00:29, 72.70it/s]
65%		3882/6000 [01:05<00:33, 62.30it/s]
65%		3889/6000 [01:05<00:38, 55.29it/s]
65%		3896/6000 [01:05<00:42, 49.82it/s]
65%		3902/6000 [01:05<00:41, 50.41it/s]
65%		3908/6000 [01:06<00:40, 51.60it/s]
65%		3914/6000 [01:06<00:51, 40.45it/s]
65%		3924/6000 [01:06<00:42, 48.90it/s]
66%		3934/6000 [01:06<00:35, 57.44it/s]
66%		3942/6000 [01:06<00:33, 62.00it/s]
66%		3950/6000 [01:06<00:32, 63.24it/s]
66%		3958/6000 [01:06<00:31, 64.23it/s]
66%		3967/6000 [01:06<00:29, 68.72it/s]
66%		3975/6000 [01:07<00:35, 56.67it/s]
66%		3983/6000 [01:07<00:33, 60.64it/s]
66%		3990/6000 [01:07<00:33, 59.49it/s]
67%		3998/6000 [01:07<00:31, 63.33it/s]
67%		4005/6000 [01:07<00:37, 53.04it/s]
67%		4015/6000 [01:07<00:32, 61.51it/s]
67%		4023/6000 [01:07<00:38, 51.73it/s]
67%		4031/6000 [01:08<00:35, 55.10it/s]
67%		4038/6000 [01:08<00:42, 45.84it/s]
68%		4051/6000 [01:08<00:34, 56.77it/s]
68%		4059/6000 [01:08<00:34, 55.97it/s]
68%		4067/6000 [01:08<00:33, 57.42it/s]
68%		4074/6000 [01:08<00:34, 55.71it/s]
68%		4081/6000 [01:08<00:33, 57.54it/s]
68%		4088/6000 [01:09<00:39, 48.94it/s]
68%		4095/6000 [01:09<00:39, 48.11it/s]
68%		4102/6000 [01:09<00:37, 50.67it/s]
69%		4111/6000 [01:09<00:34, 55.06it/s]
69%		4119/6000 [01:09<00:31, 59.67it/s]
69%		4126/6000 [01:09<00:30, 61.03it/s]
69%		4133/6000 [01:09<00:34, 54.16it/s]
69%		4140/6000 [01:09<00:32, 57.57it/s]
69%		4147/6000 [01:10<00:30, 60.10it/s]
69%		4155/6000 [01:10<00:29, 62.77it/s]
69%		4162/6000 [01:10<00:32, 56.57it/s]
70%		4171/6000 [01:10<00:29, 62.84it/s]

70%		4178/6000 [01:10<00:28, 63.49it/s]
70%		4185/6000 [01:10<00:31, 57.54it/s]
70%		4192/6000 [01:10<00:38, 46.59it/s]
70%		4201/6000 [01:11<00:33, 54.16it/s]
70%		4208/6000 [01:11<00:37, 47.72it/s]
70%		4214/6000 [01:11<00:54, 32.83it/s]
70%		4219/6000 [01:11<00:57, 31.08it/s]
71%		4231/6000 [01:11<00:44, 39.89it/s]
71%		4238/6000 [01:11<00:43, 40.59it/s]
71%		4245/6000 [01:12<00:41, 42.42it/s]
71%		4253/6000 [01:12<00:35, 48.99it/s]
71%		4260/6000 [01:12<00:32, 53.08it/s]
71%		4267/6000 [01:12<00:31, 55.29it/s]
71%		4274/6000 [01:12<00:31, 55.53it/s]
71%		4281/6000 [01:12<00:29, 58.72it/s]
71%		4288/6000 [01:12<00:29, 58.54it/s]
72%		4298/6000 [01:12<00:25, 65.56it/s]
72%		4306/6000 [01:13<00:25, 66.20it/s]
72%		4314/6000 [01:13<00:24, 69.00it/s]
72%		4322/6000 [01:13<00:25, 65.32it/s]
72%		4332/6000 [01:13<00:23, 71.55it/s]
72%		4340/6000 [01:13<00:22, 72.57it/s]
72%		4348/6000 [01:13<00:27, 61.09it/s]
73%		4355/6000 [01:13<00:25, 63.43it/s]
73%		4362/6000 [01:13<00:28, 56.92it/s]
73%		4369/6000 [01:14<00:28, 56.94it/s]
73%		4375/6000 [01:14<00:33, 48.33it/s]
73%		4382/6000 [01:14<00:34, 47.07it/s]
73%		4388/6000 [01:14<00:32, 49.98it/s]
73%		4394/6000 [01:14<00:30, 51.91it/s]
73%		4400/6000 [01:14<00:36, 44.30it/s]
73%		4408/6000 [01:14<00:31, 49.97it/s]
74%		4414/6000 [01:14<00:30, 52.14it/s]
74%		4422/6000 [01:15<00:27, 57.43it/s]
74%		4429/6000 [01:15<00:28, 55.01it/s]
74%		4436/6000 [01:15<00:28, 54.67it/s]
74%		4444/6000 [01:15<00:29, 53.56it/s]
74%		4450/6000 [01:15<00:31, 49.82it/s]
74%		4458/6000 [01:15<00:27, 55.85it/s]
74%		4466/6000 [01:15<00:25, 60.34it/s]
75%		4473/6000 [01:15<00:25, 58.88it/s]
75%		4480/6000 [01:16<00:26, 57.83it/s]
75%		4486/6000 [01:16<00:27, 54.24it/s]
75%		4492/6000 [01:16<00:27, 54.47it/s]
75%		4500/6000 [01:16<00:25, 57.99it/s]
75%		4508/6000 [01:16<00:24, 60.43it/s]
75%		4515/6000 [01:16<00:24, 61.36it/s]
75%		4525/6000 [01:16<00:22, 65.52it/s]
76%		4536/6000 [01:16<00:19, 73.73it/s]

76%		4545/6000 [01:17<00:18, 77.90it/s]
76%		4554/6000 [01:17<00:19, 76.04it/s]
76%		4562/6000 [01:17<00:19, 72.07it/s]
76%		4570/6000 [01:17<00:26, 53.01it/s]
76%		4577/6000 [01:17<00:25, 56.69it/s]
76%		4585/6000 [01:17<00:23, 60.95it/s]
77%		4592/6000 [01:17<00:22, 62.46it/s]
77%		4601/6000 [01:17<00:20, 67.10it/s]
77%		4609/6000 [01:18<00:24, 57.16it/s]
77%		4618/6000 [01:18<00:23, 58.19it/s]
77%		4627/6000 [01:18<00:21, 64.41it/s]
77%		4634/6000 [01:18<00:22, 59.50it/s]
77%		4641/6000 [01:18<00:22, 59.33it/s]
78%		4650/6000 [01:18<00:20, 65.43it/s]
78%		4659/6000 [01:18<00:19, 69.20it/s]
78%		4668/6000 [01:18<00:17, 74.04it/s]
78%		4676/6000 [01:19<00:20, 64.56it/s]
78%		4685/6000 [01:19<00:20, 63.70it/s]
78%		4693/6000 [01:19<00:19, 66.61it/s]
78%		4700/6000 [01:19<00:29, 44.04it/s]
78%		4707/6000 [01:19<00:26, 48.48it/s]
79%		4713/6000 [01:19<00:26, 49.29it/s]
79%		4719/6000 [01:20<00:26, 48.21it/s]
79%		4725/6000 [01:20<00:27, 46.73it/s]
79%		4735/6000 [01:20<00:23, 53.50it/s]
79%		4741/6000 [01:20<00:28, 44.25it/s]
79%		4747/6000 [01:20<00:29, 42.30it/s]
79%		4754/6000 [01:20<00:26, 46.61it/s]
79%		4760/6000 [01:20<00:28, 44.16it/s]
79%		4769/6000 [01:21<00:23, 51.62it/s]
80%		4779/6000 [01:21<00:20, 59.47it/s]
80%		4786/6000 [01:21<00:21, 57.50it/s]
80%		4793/6000 [01:21<00:20, 59.17it/s]
80%		4802/6000 [01:21<00:18, 63.49it/s]
80%		4809/6000 [01:21<00:18, 63.08it/s]
80%		4819/6000 [01:21<00:16, 70.78it/s]
80%		4827/6000 [01:21<00:20, 57.15it/s]
81%		4837/6000 [01:22<00:19, 60.33it/s]
81%		4844/6000 [01:22<00:23, 48.59it/s]
81%		4855/6000 [01:22<00:19, 57.36it/s]
81%		4862/6000 [01:22<00:19, 59.17it/s]
81%		4869/6000 [01:22<00:26, 42.83it/s]
81%		4875/6000 [01:22<00:25, 43.91it/s]
81%		4881/6000 [01:22<00:23, 46.64it/s]
82%		4890/6000 [01:23<00:20, 53.27it/s]
82%		4897/6000 [01:23<00:25, 42.69it/s]
82%		4903/6000 [01:23<00:23, 46.27it/s]
82%		4909/6000 [01:23<00:22, 48.11it/s]

82%		4915/6000 [01:23<00:24, 45.07it/s]
82%		4920/6000 [01:23<00:28, 38.08it/s]
82%		4926/6000 [01:24<00:27, 38.44it/s]
82%		4931/6000 [01:24<00:31, 33.53it/s]
82%		4935/6000 [01:24<00:32, 33.07it/s]
82%		4946/6000 [01:24<00:25, 41.58it/s]
83%		4954/6000 [01:24<00:22, 46.28it/s]
83%		4961/6000 [01:24<00:20, 49.88it/s]
83%		4968/6000 [01:24<00:19, 53.19it/s]
83%		4975/6000 [01:24<00:20, 51.21it/s]
83%		4984/6000 [01:25<00:17, 56.98it/s]
83%		4991/6000 [01:25<00:17, 58.98it/s]
83%		4998/6000 [01:25<00:22, 44.25it/s]
83%		5006/6000 [01:25<00:20, 47.60it/s]
84%		5014/6000 [01:25<00:19, 51.59it/s]
84%		5020/6000 [01:26<00:35, 27.23it/s]
84%		5027/6000 [01:26<00:29, 32.85it/s]
84%		5032/6000 [01:26<00:27, 35.13it/s]
84%		5040/6000 [01:26<00:23, 41.70it/s]
84%		5053/6000 [01:26<00:18, 52.10it/s]
84%		5064/6000 [01:26<00:15, 60.19it/s]
85%		5073/6000 [01:26<00:14, 63.77it/s]
85%		5081/6000 [01:26<00:14, 64.06it/s]
85%		5089/6000 [01:27<00:16, 54.11it/s]
85%		5096/6000 [01:27<00:17, 52.29it/s]
85%		5103/6000 [01:27<00:15, 56.13it/s]
85%		5110/6000 [01:27<00:17, 50.54it/s]
85%		5117/6000 [01:27<00:16, 53.15it/s]
85%		5128/6000 [01:27<00:14, 58.27it/s]
86%		5135/6000 [01:28<00:16, 51.76it/s]
86%		5142/6000 [01:28<00:15, 55.58it/s]
86%		5150/6000 [01:28<00:15, 56.62it/s]
86%		5159/6000 [01:28<00:13, 63.23it/s]
86%		5166/6000 [01:28<00:17, 48.65it/s]
86%		5172/6000 [01:28<00:18, 45.03it/s]
86%		5178/6000 [01:28<00:19, 41.80it/s]
86%		5184/6000 [01:29<00:18, 44.33it/s]
86%		5190/6000 [01:29<00:16, 48.07it/s]
87%		5198/6000 [01:29<00:14, 54.40it/s]
87%		5205/6000 [01:29<00:13, 57.54it/s]
87%		5216/6000 [01:29<00:12, 63.90it/s]
87%		5225/6000 [01:29<00:11, 65.24it/s]
87%		5232/6000 [01:29<00:12, 59.98it/s]
87%		5239/6000 [01:29<00:15, 50.10it/s]
87%		5247/6000 [01:30<00:13, 56.38it/s]
88%		5254/6000 [01:30<00:13, 55.95it/s]
88%		5261/6000 [01:30<00:16, 45.48it/s]
88%		5269/6000 [01:30<00:14, 49.58it/s]
88%		5276/6000 [01:30<00:13, 53.48it/s]

88%		5282/6000 [01:30<00:13, 55.15it/s]
88%		5288/6000 [01:30<00:15, 46.75it/s]
88%		5294/6000 [01:30<00:14, 48.01it/s]
88%		5302/6000 [01:31<00:13, 52.74it/s]
88%		5309/6000 [01:31<00:12, 56.90it/s]
89%		5316/6000 [01:31<00:14, 48.47it/s]
89%		5325/6000 [01:31<00:12, 55.83it/s]
89%		5332/6000 [01:31<00:11, 58.67it/s]
89%		5339/6000 [01:31<00:11, 59.02it/s]
89%		5347/6000 [01:31<00:10, 61.18it/s]
89%		5355/6000 [01:31<00:09, 65.65it/s]
89%		5362/6000 [01:32<00:09, 65.61it/s]
89%		5369/6000 [01:32<00:09, 66.13it/s]
90%		5377/6000 [01:32<00:09, 68.84it/s]
90%		5385/6000 [01:32<00:10, 59.98it/s]
90%		5392/6000 [01:32<00:11, 52.82it/s]
90%		5398/6000 [01:32<00:11, 52.64it/s]
90%		5406/6000 [01:32<00:10, 55.17it/s]
90%		5417/6000 [01:32<00:09, 62.16it/s]
90%		5424/6000 [01:33<00:09, 62.02it/s]
91%		5431/6000 [01:33<00:09, 60.15it/s]
91%		5439/6000 [01:33<00:09, 56.84it/s]
91%		5449/6000 [01:33<00:08, 64.70it/s]
91%		5459/6000 [01:33<00:07, 71.35it/s]
91%		5467/6000 [01:33<00:08, 64.42it/s]
91%		5474/6000 [01:33<00:09, 55.76it/s]
91%		5485/6000 [01:34<00:07, 64.53it/s]
92%		5493/6000 [01:34<00:08, 62.60it/s]
92%		5503/6000 [01:34<00:07, 68.77it/s]
92%		5511/6000 [01:34<00:06, 70.02it/s]
92%		5520/6000 [01:34<00:06, 74.49it/s]
92%		5529/6000 [01:34<00:06, 77.55it/s]
92%		5538/6000 [01:34<00:06, 71.63it/s]
92%		5546/6000 [01:34<00:07, 62.62it/s]
93%		5554/6000 [01:34<00:06, 66.59it/s]
93%		5562/6000 [01:35<00:06, 64.88it/s]
93%		5569/6000 [01:35<00:07, 58.34it/s]
93%		5578/6000 [01:35<00:06, 64.10it/s]
93%		5585/6000 [01:35<00:07, 58.90it/s]
93%		5593/6000 [01:35<00:06, 61.85it/s]
93%		5600/6000 [01:35<00:07, 53.31it/s]
93%		5606/6000 [01:35<00:08, 48.11it/s]
94%		5613/6000 [01:36<00:07, 50.76it/s]
94%		5621/6000 [01:36<00:06, 56.99it/s]
94%		5628/6000 [01:36<00:07, 48.78it/s]
94%		5639/6000 [01:36<00:06, 54.98it/s]
94%		5648/6000 [01:36<00:05, 61.70it/s]
94%		5655/6000 [01:36<00:06, 50.13it/s]

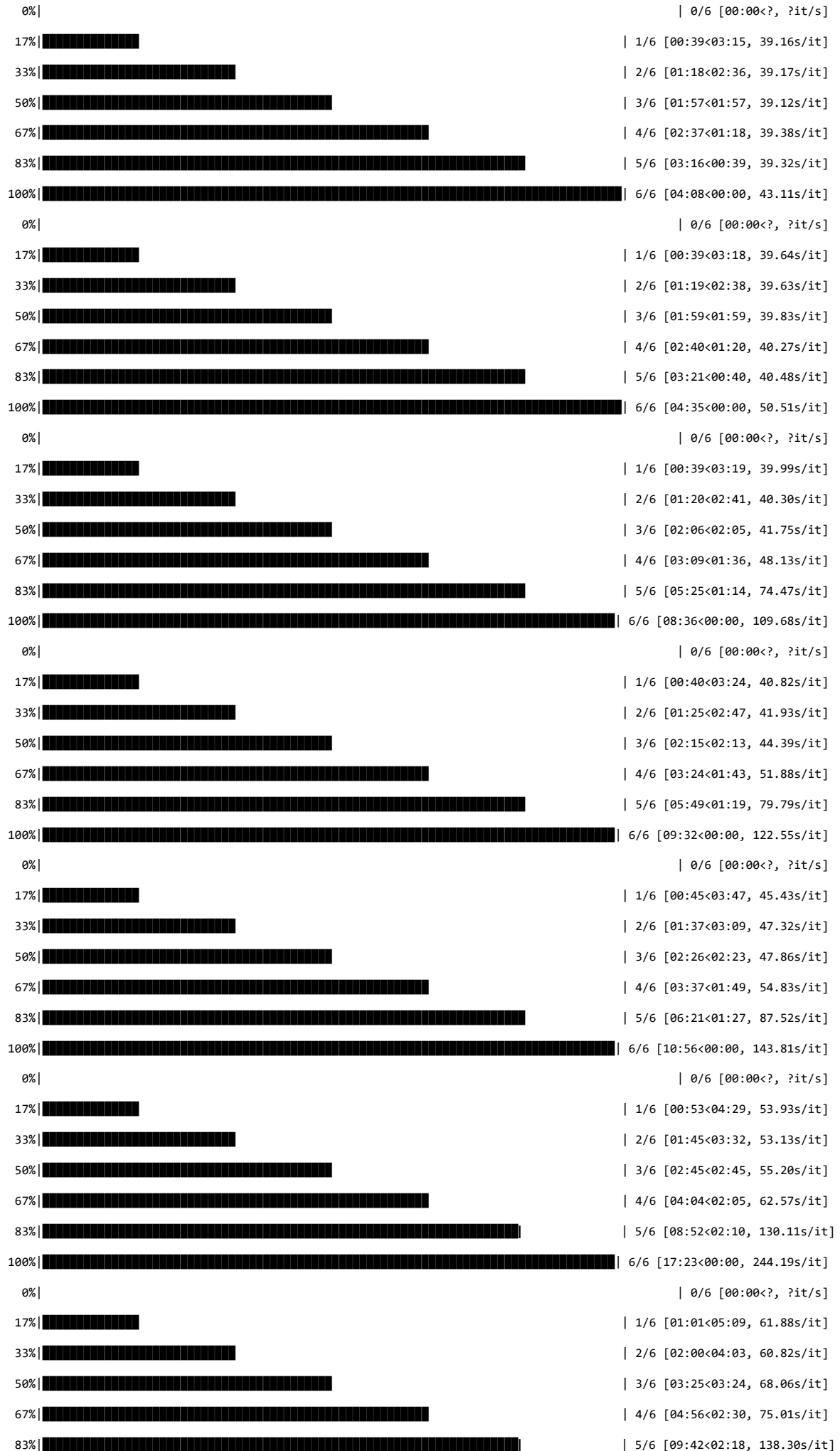


```
In [112]: train_tfidf2v = tfidf_sent_vectors_train  
cv_tfidf2v = tfidf_sent_vectors_CV  
test_tfidf2v = tfidf_sent_vectors_test
```

```
In [113]: sc = StandardScaler(with_mean=False)  
tfidf2v_X_train_sc = sc.fit_transform(train_tfidf2v)  
tfidf2v_X_cv_sc = sc.transform(cv_tfidf2v)  
tfidf2v_X_test_sc = sc.transform(test_tfidf2v)
```

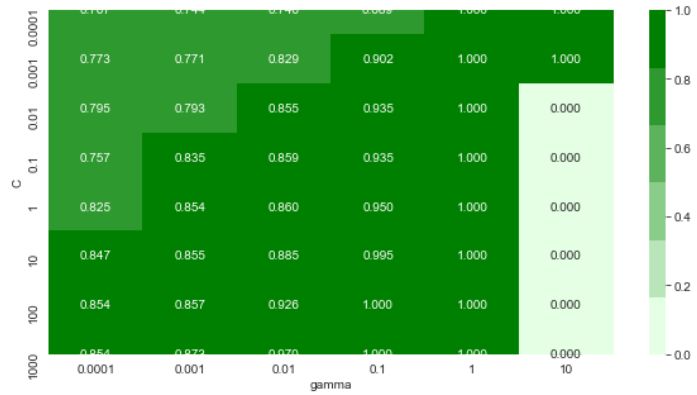


```
In [114]: SVM(tfidf2v_X_train_sc,tfidf2v_X_cv_sc,y_tr,y_cv)
```

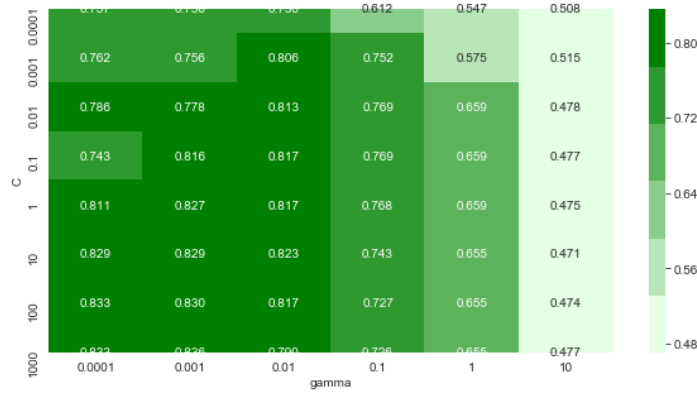




----- AUC Score for training data -----



----- AUC Score for CV data -----



```
In [116]: testing_SVM(tfidf2v_X_train_sc,y_tr,tfidf2v_X_test_sc,y_test,0.001,0.01)
```

AUC Score 0.8038964500348205



the maximum value of $tpr \cdot (1 - fpr)$ 0.5356987605349912 for threshold 0.896

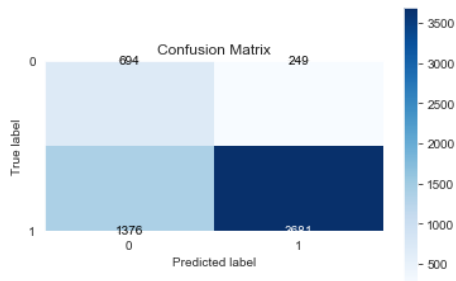
macro f1 score for data : 0.6399268464302035

micro f1 score for data: 0.7291666666666665

hamming loss for data: 0.2708333333333333

Precision recall report for data:

	precision	recall	f1-score	support
0	0.34	0.74	0.46	943
1	0.94	0.73	0.82	5057
accuracy			0.73	6000
macro avg	0.64	0.73	0.64	6000
weighted avg	0.84	0.73	0.76	6000



```
In [117]: from prettytable import PrettyTable
x = PrettyTable()
x.field_names = ["Vectorizer", "Model", "Best Hyper Parameter(C)", "Test AUC Score"]
x.add_row(["Bow", "Liner kernel", 0.1, 90.14])
x.add_row(["Tf-Idf", "Liner kernel", 10, 87.86])
x.add_row(["Avg-W2V", "Liner kernel", 0.01, 89.60])
x.add_row(["TfIdf-W2V", "Liner kernel", 0.0001, 86.53])

y = PrettyTable()
y.field_names = ["Vectorizer", "Model", "Best Hyper Parameter(C & gamma)", "Test AUC Score"]
y.add_row(["Bow", "RBF kernel", (0.001, 0.001), 72.27])
y.add_row(["Tf-Idf", "RBF kernel", (0.001, 0.0001), 54.40])
y.add_row(["Avg-W2V", "RBF kernel", (0.001, 0.001), 82.02])
y.add_row(["TfIdf-W2V", "RBF kernel", (0.001, 0.01), 80.28])
from IPython.display import Markdown, display
def printmd(string):
    display(Markdown(string))
printmd('****Final Conclusion:****')
print(x)
print(y)
```

Final Conclusion:

Vectorizer	Model	Best Hyper Parameter(C)	Test AUC Score
Bow	Liner kernel	0.1	90.14
Tf-Idf	Liner kernel	10	87.86
Avg-W2V	Liner kernel	0.01	89.6
TfIdf-W2V	Liner kernel	0.0001	86.53

Vectorizer	Model	Best Hyper Parameter(C & gamma)	Test AUC Score
Bow	RBF kernel	(0.001, 0.001)	72.27
Tf-Idf	RBF kernel	(0.001, 0.0001)	54.4
Avg-W2V	RBF kernel	(0.001, 0.001)	82.02
TfIdf-W2V	RBF kernel	(0.001, 0.01)	80.28