

HW1-CSCI544

September 8, 2021

```
[1]: import pandas as pd
import numpy as np
import nltk
nltk.download('wordnet')
import re
from bs4 import BeautifulSoup
```

[nltk_data] Downloading package wordnet to /Users/chenlin/nltk_data...

[nltk_data] Package wordnet is already up-to-date!

```
[2]: #!/ pip install bs4 # in case you don't have it installed

# Dataset: https://s3.amazonaws.com/amazon-reviews-pds/tsv/
→amazon_reviews_us_Kitchen_v1_00.tsv.gz
```

0.1 Read Data

```
[3]: import pandas as pd
test = pd.read_csv("amazon_reviews_us_Kitchen_v1_00.tsv", sep = '\t', error_bad_lines = False)
test['label'] = -1
```

b'Skipping line 16148: expected 15 fields, saw 22\nSkipping line 20100: expected 15 fields, saw 22\nSkipping line 45178: expected 15 fields, saw 22\nSkipping line 48700: expected 15 fields, saw 22\nSkipping line 63331: expected 15 fields, saw 22\n'

b'Skipping line 86053: expected 15 fields, saw 22\nSkipping line 88858: expected 15 fields, saw 22\nSkipping line 115017: expected 15 fields, saw 22\n'

b'Skipping line 137366: expected 15 fields, saw 22\nSkipping line 139110: expected 15 fields, saw 22\nSkipping line 165540: expected 15 fields, saw 22\nSkipping line 171813: expected 15 fields, saw 22\n'

b'Skipping line 203723: expected 15 fields, saw 22\nSkipping line 209366: expected 15 fields, saw 22\nSkipping line 211310: expected 15 fields, saw 22\nSkipping line 246351: expected 15 fields, saw 22\nSkipping line 252364: expected 15 fields, saw 22\n'

b'Skipping line 267003: expected 15 fields, saw 22\nSkipping line 268957:
expected 15 fields, saw 22\nSkipping line 303336: expected 15 fields, saw
22\nSkipping line 306021: expected 15 fields, saw 22\nSkipping line 311569:
expected 15 fields, saw 22\nSkipping line 316767: expected 15 fields, saw
22\nSkipping line 324009: expected 15 fields, saw 22\n'
b'Skipping line 359107: expected 15 fields, saw 22\nSkipping line 368367:
expected 15 fields, saw 22\nSkipping line 381180: expected 15 fields, saw
22\nSkipping line 390453: expected 15 fields, saw 22\n'
b'Skipping line 412243: expected 15 fields, saw 22\nSkipping line 419342:
expected 15 fields, saw 22\nSkipping line 457388: expected 15 fields, saw 22\n'
b'Skipping line 459935: expected 15 fields, saw 22\nSkipping line 460167:
expected 15 fields, saw 22\nSkipping line 466460: expected 15 fields, saw
22\nSkipping line 500314: expected 15 fields, saw 22\nSkipping line 500339:
expected 15 fields, saw 22\nSkipping line 505396: expected 15 fields, saw
22\nSkipping line 507760: expected 15 fields, saw 22\nSkipping line 513626:
expected 15 fields, saw 22\n'
b'Skipping line 527638: expected 15 fields, saw 22\nSkipping line 534209:
expected 15 fields, saw 22\nSkipping line 535687: expected 15 fields, saw
22\nSkipping line 547671: expected 15 fields, saw 22\nSkipping line 549054:
expected 15 fields, saw 22\n'
b'Skipping line 599929: expected 15 fields, saw 22\nSkipping line 604776:
expected 15 fields, saw 22\nSkipping line 609937: expected 15 fields, saw
22\nSkipping line 632059: expected 15 fields, saw 22\nSkipping line 638546:
expected 15 fields, saw 22\n'
b'Skipping line 665017: expected 15 fields, saw 22\nSkipping line 677680:
expected 15 fields, saw 22\nSkipping line 684370: expected 15 fields, saw
22\nSkipping line 720217: expected 15 fields, saw 29\n'
b'Skipping line 723240: expected 15 fields, saw 22\nSkipping line 723433:
expected 15 fields, saw 22\nSkipping line 763891: expected 15 fields, saw 22\n'
b'Skipping line 800288: expected 15 fields, saw 22\nSkipping line 802942:
expected 15 fields, saw 22\nSkipping line 803379: expected 15 fields, saw
22\nSkipping line 805122: expected 15 fields, saw 22\nSkipping line 821899:
expected 15 fields, saw 22\nSkipping line 831707: expected 15 fields, saw
22\nSkipping line 842829: expected 15 fields, saw 22\nSkipping line 843604:
expected 15 fields, saw 22\n'
b'Skipping line 863904: expected 15 fields, saw 22\nSkipping line 875655:
expected 15 fields, saw 22\nSkipping line 886796: expected 15 fields, saw
22\nSkipping line 892299: expected 15 fields, saw 22\nSkipping line 902518:
expected 15 fields, saw 22\nSkipping line 903079: expected 15 fields, saw
22\nSkipping line 912678: expected 15 fields, saw 22\n'
b'Skipping line 932953: expected 15 fields, saw 22\nSkipping line 936838:
expected 15 fields, saw 22\nSkipping line 937177: expected 15 fields, saw
22\nSkipping line 947695: expected 15 fields, saw 22\nSkipping line 960713:
expected 15 fields, saw 22\nSkipping line 965225: expected 15 fields, saw
22\nSkipping line 980776: expected 15 fields, saw 22\n'
b'Skipping line 999318: expected 15 fields, saw 22\nSkipping line 1007247:
expected 15 fields, saw 22\nSkipping line 1015987: expected 15 fields, saw
22\nSkipping line 1018984: expected 15 fields, saw 22\nSkipping line 1028671:

expected 15 fields, saw 22\n'
b'Skipping line 1063360: expected 15 fields, saw 22\nSkipping line 1066195:
expected 15 fields, saw 22\nSkipping line 1066578: expected 15 fields, saw
22\nSkipping line 1066869: expected 15 fields, saw 22\nSkipping line 1068809:
expected 15 fields, saw 22\nSkipping line 1069505: expected 15 fields, saw
22\nSkipping line 1087983: expected 15 fields, saw 22\nSkipping line 1108184:
expected 15 fields, saw 22\n'
b'Skipping line 1118137: expected 15 fields, saw 22\nSkipping line 1142723:
expected 15 fields, saw 22\nSkipping line 1152492: expected 15 fields, saw
22\nSkipping line 1156947: expected 15 fields, saw 22\nSkipping line 1172563:
expected 15 fields, saw 22\n'
b'Skipping line 1209254: expected 15 fields, saw 22\nSkipping line 1212966:
expected 15 fields, saw 22\nSkipping line 1236533: expected 15 fields, saw
22\nSkipping line 1237598: expected 15 fields, saw 22\n'
b'Skipping line 1273825: expected 15 fields, saw 22\nSkipping line 1277898:
expected 15 fields, saw 22\nSkipping line 1283654: expected 15 fields, saw
22\nSkipping line 1286023: expected 15 fields, saw 22\nSkipping line 1302038:
expected 15 fields, saw 22\nSkipping line 1305179: expected 15 fields, saw 22\n'
b'Skipping line 1326022: expected 15 fields, saw 22\nSkipping line 1338120:
expected 15 fields, saw 22\nSkipping line 1338503: expected 15 fields, saw
22\nSkipping line 1338849: expected 15 fields, saw 22\nSkipping line 1341513:
expected 15 fields, saw 22\nSkipping line 1346493: expected 15 fields, saw
22\nSkipping line 1373127: expected 15 fields, saw 22\n'
b'Skipping line 1389508: expected 15 fields, saw 22\nSkipping line 1413951:
expected 15 fields, saw 22\nSkipping line 1433626: expected 15 fields, saw 22\n'
b'Skipping line 1442698: expected 15 fields, saw 22\nSkipping line 1472982:
expected 15 fields, saw 22\nSkipping line 1482282: expected 15 fields, saw
22\nSkipping line 1487808: expected 15 fields, saw 22\nSkipping line 1500636:
expected 15 fields, saw 22\n'
b'Skipping line 1511479: expected 15 fields, saw 22\nSkipping line 1532302:
expected 15 fields, saw 22\nSkipping line 1537952: expected 15 fields, saw
22\nSkipping line 1539951: expected 15 fields, saw 22\nSkipping line 1541020:
expected 15 fields, saw 22\n'
b'Skipping line 1594217: expected 15 fields, saw 22\nSkipping line 1612264:
expected 15 fields, saw 22\nSkipping line 1615907: expected 15 fields, saw
22\nSkipping line 1621859: expected 15 fields, saw 22\n'
b'Skipping line 1653542: expected 15 fields, saw 22\nSkipping line 1671537:
expected 15 fields, saw 22\nSkipping line 1672879: expected 15 fields, saw
22\nSkipping line 1674523: expected 15 fields, saw 22\nSkipping line 1677355:
expected 15 fields, saw 22\nSkipping line 1703907: expected 15 fields, saw 22\n'
b'Skipping line 1713046: expected 15 fields, saw 22\nSkipping line 1722982:
expected 15 fields, saw 22\nSkipping line 1727290: expected 15 fields, saw
22\nSkipping line 1744482: expected 15 fields, saw 22\n'
b'Skipping line 1803858: expected 15 fields, saw 22\nSkipping line 1810069:
expected 15 fields, saw 22\nSkipping line 1829751: expected 15 fields, saw
22\nSkipping line 1831699: expected 15 fields, saw 22\n'
b'Skipping line 1863131: expected 15 fields, saw 22\nSkipping line 1867917:
expected 15 fields, saw 22\nSkipping line 1874790: expected 15 fields, saw

22\nSkipping line 1879952: expected 15 fields, saw 22\nSkipping line 1880501: expected 15 fields, saw 22\nSkipping line 1886655: expected 15 fields, saw 22\nSkipping line 1887888: expected 15 fields, saw 22\nSkipping line 1894286: expected 15 fields, saw 22\nSkipping line 1895400: expected 15 fields, saw 22\n' b'Skipping line 1904040: expected 15 fields, saw 22\nSkipping line 1907604: expected 15 fields, saw 22\nSkipping line 1915739: expected 15 fields, saw 22\nSkipping line 1921514: expected 15 fields, saw 22\nSkipping line 1939428: expected 15 fields, saw 22\nSkipping line 1944342: expected 15 fields, saw 22\nSkipping line 1949699: expected 15 fields, saw 22\nSkipping line 1961872: expected 15 fields, saw 22\n' b'Skipping line 1968846: expected 15 fields, saw 22\nSkipping line 1999941: expected 15 fields, saw 22\nSkipping line 2001492: expected 15 fields, saw 22\nSkipping line 2011204: expected 15 fields, saw 22\nSkipping line 2025295: expected 15 fields, saw 22\n' b'Skipping line 2041266: expected 15 fields, saw 22\nSkipping line 2073314: expected 15 fields, saw 22\nSkipping line 2080133: expected 15 fields, saw 22\nSkipping line 2088521: expected 15 fields, saw 22\n' b'Skipping line 2103490: expected 15 fields, saw 22\nSkipping line 2115278: expected 15 fields, saw 22\nSkipping line 2153174: expected 15 fields, saw 22\nSkipping line 2161731: expected 15 fields, saw 22\n' b'Skipping line 2165250: expected 15 fields, saw 22\nSkipping line 2175132: expected 15 fields, saw 22\nSkipping line 2206817: expected 15 fields, saw 22\nSkipping line 2215848: expected 15 fields, saw 22\nSkipping line 2223811: expected 15 fields, saw 22\n' b'Skipping line 2257265: expected 15 fields, saw 22\nSkipping line 2259163: expected 15 fields, saw 22\nSkipping line 2263291: expected 15 fields, saw 22\n' b'Skipping line 2301943: expected 15 fields, saw 22\nSkipping line 2304371: expected 15 fields, saw 22\nSkipping line 2306015: expected 15 fields, saw 22\nSkipping line 2312186: expected 15 fields, saw 22\nSkipping line 2314740: expected 15 fields, saw 22\nSkipping line 2317754: expected 15 fields, saw 22\n' b'Skipping line 2383514: expected 15 fields, saw 22\n' b'Skipping line 2449763: expected 15 fields, saw 22\n' b'Skipping line 2589323: expected 15 fields, saw 22\n' b'Skipping line 2775036: expected 15 fields, saw 22\n' b'Skipping line 2935174: expected 15 fields, saw 22\n' b'Skipping line 3078830: expected 15 fields, saw 22\n' b'Skipping line 3123091: expected 15 fields, saw 22\n' b'Skipping line 3185533: expected 15 fields, saw 22\n' b'Skipping line 4150395: expected 15 fields, saw 22\n' b'Skipping line 4748401: expected 15 fields, saw 22\n'

0.2 Keep Reviews and Ratings

```
[4]: test.label[test.star_rating>3] = 1
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1:
SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

"""Entry point for launching an IPython kernel.

1 Labelling Reviews:

1.1 The reviews with rating 4,5 are labelled to be 1 and 1,2 are labelled as 0. Discard the reviews with rating 3'

```
[5]: test.label[test.star_rating>3] = 1
test.label[test.star_rating<3] = 0
test = test[['label','review_body']]
```

/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

"""Entry point for launching an IPython kernel.

/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
[6]: show = list(test.groupby('label').count().review_body)
```

```
[7]: print('The numbers for 3 classes are: ' +
      +str(show[0])+' , '+str(show[1])+' , '+str(show[2]))
new_test = test.loc[test['label']!= -1]
```

The numbers for 3 classes are:349539,668809,3856296

We select 200000 reviews randomly with 100,000 positive and 100,000 negative reviews.

```
[8]: postive = new_test.loc[test['label']==1].sample(100000)
negative = new_test.loc[test['label']== 0].sample(100000)
new_p_n = pd.concat([postive,negative])
```

```
[9]: length_before_cleaning = new_p_n['review_body'].apply(lambda x:len(str(x))).
      +mean()
```

2 Data Cleaning

2.1 Convert the all reviews into the lower case.

```
[10]: new_p_n['review_body'] = new_p_n['review_body'].str.lower()
```

2.2 remove the HTML and URLs from the reviews

```
[11]: def tag(x):  
        return re.sub('<.*?>', '', str(x))  
new_p_n['review_body'] = new_p_n['review_body'].apply(lambda x:tag(x))  
  
def url(x):  
    return re.sub('(https?|ftp|file):\/\/[-A-Za-z0-9+&@#/%?=_|!:,.;  
→]+[-A-Za-z0-9+&@#/%?=_|]', '', str(x))  
  
new_p_n['review_body'] = new_p_n['review_body'].apply(lambda x:url(x))
```

2.3 perform contractions on the reviews.

```
[12]: import contractions  
new_p_n['review_body'] = new_p_n['review_body'].apply(lambda x:contractions.  
→fix(x))
```

2.4 remove non-alphabetical characters

```
[13]: def non_alphabetical(x):  
        return re.sub('[^a-zA-Z\s]', '', str(x))  
  
new_p_n['review_body'] = new_p_n['review_body'].apply(lambda x:  
→non_alphabetical(x))
```

2.5 Remove the extra spaces between the words

```
[14]: def extra_space(x):  
        return re.sub(' +', ' ', str(x))  
new_p_n['review_body'] = new_p_n['review_body'].apply(lambda x:extra_space(x))
```

Average length of reviews before and after data cleaning (with comma between them)

```
[15]: length_after_cleaning = new_p_n['review_body'].apply(lambda x:len(str(x))).  
→mean()
```

```
[16]: print("Average length of reviews before and after data cleaning :  
→"+str(length_before_cleaning)+' '+str(length_after_cleaning))
```

Average length of reviews before and after data cleaning :323.2885,308.44504

3 Pre-processing

3.1 remove the stop words

```
[17]: import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words_set = set(stopwords.words('english'))
from nltk import word_tokenize, pos_tag

def stop_words(x):
    word_tokens = word_tokenize(x)
    temp = []
    for i in word_tokens:
        if i not in stop_words_set:
            temp.append(i)
    return temp
new_p_n['review_body'] = new_p_n['review_body'].apply(lambda x:stop_words(x))
```

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

3.2 perform lemmatization

```
[18]: from nltk.corpus import wordnet
from nltk.stem import WordNetLemmatizer
nltk.download('averaged_perceptron_tagger')

def get_wordnet_pos(tag):
    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('V'):
        return wordnet.VERB
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    else:
        return None

def lemmatization(x:list):
    tagged_sent = pos_tag(x)
    lemmas_sent = []
    for tag in tagged_sent:
        wnl = WordNetLemmatizer()
```

```

        pos = get_wordnet_pos(tag[1]) or wordnet.NOUN
        lemmas_sent.append(wnl.lemmatize(tag[0], pos))
    return lemmas_sent

new_p_n['review_body'] = new_p_n['review_body'].apply(lambda x:lemmatization(x))

```

```

[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] /Users/chenlin/nltk_data...
[nltk_data] Package averaged_perceptron_tagger is already up-to-
[nltk_data] date!

```

```

[19]: length_after_process = new_p_n['review_body'].apply(lambda x:len(str(x))).mean()
[20]: print("Average length of reviews before and after data cleaning :
      ->"+str(length_after_cleaning)+' '+str(length_after_process))

```

Average length of reviews before and after data cleaning :308.44504,272.204315

4 TF-IDF Feature Extraction

```

[21]: new_p_n['review_str'] = new_p_n['review_body'].apply(lambda x:' '.join(x))
      from sklearn.feature_extraction.text import TfidfVectorizer
      X = new_p_n['review_str']
      Y = new_p_n['label']
      v = TfidfVectorizer()
      x_tfidf = v.fit_transform(X)

      from sklearn.model_selection import train_test_split

      x_train, x_test, y_train, y_test = train_test_split(x_tfidf, Y, random_state = 19,
      ->19, test_size = 0.2)

```

5 Perceptron

```

[22]: from sklearn.linear_model import Perceptron
      perceptron = Perceptron()
      perceptron.fit(x_train, y_train)

      from sklearn.metrics import precision_recall_fscore_support as score
      from sklearn.metrics import classification_report

      y_train_predict = perceptron.predict(x_train)

      y_test_predict = perceptron.predict(x_test)

```



```
[23]: report_train = classification_report(y_train,y_train_predict,output_dict=True)
precision_train = report_train['macro avg']['precision']
recall_train = report_train['macro avg']['recall']
f1_train = report_train['macro avg']['f1-score']
accuracy_train = report_train['accuracy']

[24]: report_test = classification_report(y_test,y_test_predict,output_dict=True)
precision_test = report_train['macro avg']['precision']
recall_test = report_train['macro avg']['recall']
f1_test = report_train['macro avg']['f1-score']
accuracy_test = report_train['accuracy']

[38]: print('Accuracy, Precision, Recall, and f1-score for training and testing split
→\
(in the mentioned order)for Perceptron (with comma between them):\n'+ \
      str(accuracy_train)+','+str(precision_train)+','+str(recall_train)+','+ \
      \
      →str(f1_train)+','+str(accuracy_test)+','+str(precision_test)+','+str(recall_test)+','+str(f
```

Accuracy, Precision, Recall, and f1-score for training and testing split (in the mentioned order)for Perceptron (with comma between them):
0.91366875,0.9137763034536276,0.9136685485397928,0.9136631116312313,0.91366875,0.9137763034536276,0.9136685485397928,0.9136631116312313

6 SVM

```
[39]: from sklearn.svm import LinearSVC
svc = LinearSVC()
svc.fit(x_train, y_train)
y_train_predict = svc.predict(x_train)
y_test_predict = svc.predict(x_test)

[40]: report_train = classification_report(y_train,y_train_predict,output_dict=True)
precision_train = report_train['macro avg']['precision']
recall_train = report_train['macro avg']['recall']
f1_train = report_train['macro avg']['f1-score']
accuracy_train = report_train['accuracy']

[41]: report_test = classification_report(y_test,y_test_predict,output_dict=True)
precision_test = report_train['macro avg']['precision']
recall_test = report_train['macro avg']['recall']
f1_test = report_train['macro avg']['f1-score']
accuracy_test = report_train['accuracy']

[42]: print('Accuracy, Precision, Recall, and f1-score for training and testing split
→(in the mentioned order) for SVM (with comma between them):
→\n'+str(accuracy_train)+','+ \
+str(precision_train)+','+str(recall_train)+','+str(f1_train)+','+str(accuracy_test)+','+str(p
```

Accuracy, Precision, Recall, and f1-score for training and testing split (in the mentioned order) for SVM (with comma between them):

0.937625,0.9376284660071734,0.9376250352735156,0.9376248799181477,0.937625,0.9376284660071734,0.9376250352735156,0.9376248799181477

7 Logistic Regression

```
[43]: from sklearn.linear_model import LogisticRegression
logistic = LogisticRegression(random_state=0,max_iter = 300).fit(x_train,
    →y_train)
y_train_predict = logistic.predict(x_train)
y_test_predict = logistic.predict(x_test)

[44]: report_train = classification_report(y_train,y_train_predict,output_dict=True)
precision_train = report_train['macro avg']['precision']
recall_train = report_train['macro avg']['recall']
f1_train = report_train['macro avg']['f1-score']
accuracy_train = report_train['accuracy']

[45]: report_test = classification_report(y_test,y_test_predict,output_dict=True)
precision_test = report_train['macro avg']['precision']
recall_test = report_train['macro avg']['recall']
f1_test = report_train['macro avg']['f1-score']
accuracy_test = report_train['accuracy']

[46]: print('Accuracy, Precision, Recall, and f1-score for training and testing split,
    →(in the mentioned order) for Logistic Regression (with comma between them):
    →\n'+str(accuracy_train)+'\n'
    +str(precision_train)+'\n'+str(recall_train)+'\n'+str(f1_train)+'\n'+str(accuracy_test)+'\n'+str(precision_test)+'\n'+str(recall_test)+'\n'+str(f1_test))
```

Accuracy, Precision, Recall, and f1-score for training and testing split (in the mentioned order) for Logistic Regression (with comma between them):

0.91050625,0.9105340411888039,0.9105063529128165,0.9105047501679859,0.91050625,0.9105340411888039,0.9105063529128165,0.9105047501679859

8 Naive Bayes

```
[47]: from sklearn.naive_bayes import MultinomialNB
NB= MultinomialNB()
NB.fit(x_train, y_train)
y_train_predict = NB.predict(x_train)
y_test_predict = NB.predict(x_test)

[48]: report_train = classification_report(y_train,y_train_predict,output_dict=True)
precision_train = report_train['macro avg']['precision']
recall_train = report_train['macro avg']['recall']
f1_train = report_train['macro avg']['f1-score']
```

```
accuracy_train = report_train['accuracy']
```

```
[49]: report_test = classification_report(y_test,y_test_predict,output_dict=True)
precision_test = report_train['macro avg']['precision']
recall_test = report_train['macro avg']['recall']
f1_test = report_train['macro avg']['f1-score']
accuracy_test = report_train['accuracy']
```

```
[50]: print('Accuracy, Precision, Recall, and f1-score for training and testing split
→(in the mentioned order) for Naive Bayes (with comma between them):
→\n'+str(accuracy_train)+', '\
+str(precision_train)+', '+str(recall_train)+', '+str(f1_train)+', '+str(accuracy_test)+', '+str(p
```

Accuracy, Precision, Recall, and f1-score for training and testing split (in the mentioned order) for Naive Bayes (with comma between them):

0.88859375,0.8891044528958303,0.8885942028991214,0.8885572654553013,0.88859375,0.8891044528958303,0.8885942028991214,0.8885572654553013

```
[ ]:
```