

# Shakespeare\_WordTrimming

September 21, 2020

## 0.1 Trimming Player Line

The goal of this notebook is to trim the PlayerLine category of data to use to develop a model. Some of the operations, such as removing stop words, are expensive. To avoid running the operations each time, I downloaded the results of the word trimming to a new file, `PlayerLine_trimmed.csv`.

The steps in this file are taken from : <https://towardsdatascience.com/nlp-for-beginners-cleaning-preprocessing-text-data-ae8e306bef0f>

```
[1]: import numpy as np
import pandas as pd
import random
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
from sklearn import tree
```

```
[2]: import nltk
# nltk.download('stopwords')
# nltk.download('wordnet')
from bs4 import BeautifulSoup
import string
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import PorterStemmer
```

```
[3]: data = pd.read_csv('~/.Documents/EECS/EECS_731/HW/EECS731_2/data/
↳1028_2124_bundle_archive/Shakespeare_data.csv')
```

```
[4]: data.columns
```

```
[4]: Index(['Dateline', 'Play', 'PlayerLinenumber', 'ActSceneLine', 'Player',
          'PlayerLine'],
          dtype='object')
```

```
[5]: data.dtypes
```

```
[5]: Dateline          int64
     Play             object
```

```

PlayerLinenumbr    float64
ActSceneLine       object
Player             object
PlayerLine         object
dtype: object

```

What if we made the player a number? So that we could talk about what players are saying what? What if we only looked at the major players... could count how many times they speak and only use the ones that pop up the most.

Need to delete the columns where NaN is present for player.

```
[6]: data = data.dropna()
```

```
[7]: data = data.drop(columns="Dataline")
```

```
[8]: data
```

```
[8]:
```

	Play	PlayerLinenumbr	ActSceneLine	Player \
3	Henry IV	1.0	1.1.1	KING HENRY IV
4	Henry IV	1.0	1.1.2	KING HENRY IV
5	Henry IV	1.0	1.1.3	KING HENRY IV
6	Henry IV	1.0	1.1.4	KING HENRY IV
7	Henry IV	1.0	1.1.5	KING HENRY IV
...	...	...	...	...
111390	A Winters Tale	38.0	5.3.179	LEONTES
111391	A Winters Tale	38.0	5.3.180	LEONTES
111392	A Winters Tale	38.0	5.3.181	LEONTES
111393	A Winters Tale	38.0	5.3.182	LEONTES
111394	A Winters Tale	38.0	5.3.183	LEONTES

	PlayerLine
3	So shaken as we are, so wan with care,
4	Find we a time for frightened peace to pant,
5	And breathe short-winded accents of new broils
6	To be commenced in strands afar remote.
7	No more the thirsty entrance of this soil
...	...
111390	Is troth-plight to your daughter. Good Paulina,
111391	Lead us from hence, where we may leisurely
111392	Each one demand an answer to his part
111393	Perform'd in this wide gap of time since first
111394	We were dissever'd: hastily lead away.

```
[105152 rows x 5 columns]
```

**Naming Issue** In “A Comedy of Errors”, there is ANTIPHOLUS OF SYRACUSE and ANTIPHOLUS OF EPHEBUS and I think that that the ANTIPHOLUS part of the name got sepa-

rated so we can delete all of those cells with the entry for player as ANTIPHOLUS and rename the player columns that say OF EPHEBUS and OF SYRACUSE to ANTIPHOLUS OF SYRACUSE and ANTIPHOLUS OF EPHEBUS

In “Love’s Labour’s Lost” the character Don Armado is named DON in the PlayerLine column and ADRIANO DE ARMADO in the Player column. So, we can delete the cells for DON in the PlayerLine and rename in to Don Armado in the Player column.

```
[9]: data.drop(data.loc[data['PlayerLine']=="ANTIPHOLUS"].index, inplace=True)
data['Player'].replace({'OF SYRACUSE': 'ANTIPHOLUS OF SYRACUSE', 'OF EPHEBUS': '↪'
↪ 'ANTIPHOLUS OF EPHEBUS'})
```

```
[9]: 3      KING HENRY IV
4      KING HENRY IV
5      KING HENRY IV
6      KING HENRY IV
7      KING HENRY IV
...
111390      LEONTES
111391      LEONTES
111392      LEONTES
111393      LEONTES
111394      LEONTES
Name: Player, Length: 104977, dtype: object
```

```
[10]: data.drop(data.loc[data['PlayerLine']=="DON"].index, inplace=True)
data['Player'].replace({'ADRIANO DE ARMADO': 'DON ARMADO',})
```

```
[10]: 3      KING HENRY IV
4      KING HENRY IV
5      KING HENRY IV
6      KING HENRY IV
7      KING HENRY IV
...
111390      LEONTES
111391      LEONTES
111392      LEONTES
111393      LEONTES
111394      LEONTES
Name: Player, Length: 104876, dtype: object
```

## 0.2 Preprocessing text data

### 0.2.1 Goal

My goal was to preprocess the text data to use it for analysis in the ShakespeareAnalysis notebook. I wanted to reason if a line was positive or negative and see if that could help increase the accuracy when identifying the player.

First, I removed punctuation from each of the lines. We really only care about important words, not the punctuation around those words.

```
[11]: def remove_punctuation(text):
        no_punct = "".join([c for c in text if c not in string.punctuation])
        return no_punct

[12]: data['PlayerLine'] = data['PlayerLine'].apply(lambda x: remove_punctuation(x))
data.head()
```

```
[12]:      Play  PlayerLinenum  ActSceneLine  Player \
3  Henry IV              1.0         1.1.1 KING HENRY IV
4  Henry IV              1.0         1.1.2 KING HENRY IV
5  Henry IV              1.0         1.1.3 KING HENRY IV
6  Henry IV              1.0         1.1.4 KING HENRY IV
7  Henry IV              1.0         1.1.5 KING HENRY IV

      PlayerLine
3      So shaken as we are so wan with care
4      Find we a time for frightened peace to pant
5  And breathe shortwinded accents of new broils
6      To be commenced in strands afar remote
7      No more the thirsty entrance of this soil
```

Here, I broke the strings into a list of words based on spaces.

```
[13]: tokenizer = RegexpTokenizer(r'\w+')

[14]: data['PlayerLine'] = data['PlayerLine'].apply(lambda x: tokenizer.tokenize(x).
        ↪lower())
data['PlayerLine'].head(10)
```

```
[14]: 3      [so, shaken, as, we, are, so, wan, with, care]
4      [find, we, a, time, for, frightened, peace, to, ...
5      [and, breathe, shortwinded, accents, of, new, ...
6      [to, be, commenced, in, strands, afar, remote]
7      [no, more, the, thirsty, entrance, of, this, s...
8      [shall, daub, her, lips, with, her, own, child...
9      [nor, more, shall, trenching, war, channel, he...
10     [nor, bruise, her, flowerets, with, the, armed...
11     [of, hostile, paces, those, opposed, eyes]
12     [which, like, the, meteors, of, a, troubled, h...
Name: PlayerLine, dtype: object
```

Next, we must remove stop words. Stop words are words like “the” that get in the way of understanding the meaning of a line.

```
[15]: def remove_stopwords(text):
      words = [w for w in text if w not in stopwords.words('english')]
      return words
```

```
[16]: data['PlayerLine'] = data['PlayerLine'].apply(lambda x: remove_stopwords(x))
      data['PlayerLine'].head(5)
```

```
[16]: 3          [shaken, wan, care]
      4          [find, time, frightened, peace, pant]
      5  [breathe, shortwinded, accents, new, broils]
      6          [commenced, strands, afar, remote]
      7          [thirsty, entrance, soil]
      Name: PlayerLine, dtype: object
```

Now, we cut off prefixes and suffixes. For example, a conjugated word like “slept” would just become sleep. This helps us reason about the words without caring about every tense the word could be in.

```
[17]: lemmatizer = WordNetLemmatizer()

      def word_lemmatizer(text):
          lem_text = [lemmatizer.lemmatize(i) for i in text]
          return lem_text
```

```
[18]: data['PlayerLine'].apply(lambda x : word_lemmatizer(x))
      data['PlayerLine'].head(5)
```

```
[18]: 3          [shaken, wan, care]
      4          [find, time, frightened, peace, pant]
      5  [breathe, shortwinded, accents, new, broils]
      6          [commenced, strands, afar, remote]
      7          [thirsty, entrance, soil]
      Name: PlayerLine, dtype: object
```

Finally, you can add the data back together using the join function.

```
[19]: stemmer = PorterStemmer()

      def word_stemmer(text):
          stem_text = " ".join([stemmer.stem(i) for i in text])
          return stem_text
```

```
[20]: data['PlayerLine'].apply(lambda x : word_stemmer(x))
      data['PlayerLine'].head(5)
```

```
[20]: 3          [shaken, wan, care]
      4          [find, time, frightened, peace, pant]
      5  [breathe, shortwinded, accents, new, broils]
```

```

6             [commenced, strands, afar, remote]
7             [thirsty, entrance, soil]
Name: PlayerLine, dtype: object

```

### 0.3 Exporting data

Here, after we finished trimming, I exported the data to a csv. I used the first csv when performing the analysis.

```
[21]: data.to_csv('PlayerLine_trimmed_2.csv')
```

There are 104876 rows of data. For Model training, we only want to use 80% of the data for training. I will create a subset of the data for training and a subset of the data for testing. Can use `data_train = data.sample(frac=0.8)` to get 80% of the data but then you can't get the other 20% for testing.

```
[22]: data_shuf = shuffle(data)
data_train = data_shuf[:84000]
data_test = data_shuf[-21052:]
```

```
[23]: data_train
```

```
[23]:
```

	Play	PlayerLinenum	ActSceneLine	Player	\
16895	As you like it	36.0	2.7.183	AMIENS	
110371	A Winters Tale	145.0	4.4.577	FLORIZEL	
63004	Merchant of Venice	37.0	3.2.259	BASSANIO	
68185	A Midsummer nights dream	31.0	3.2.146	DEMETRIUS	
77634	Pericles	13.0	4.3.56	DIONYZA	
...	...	...	...	...	
61721	Merchant of Venice	15.0	1.3.30	SHYLOCK	
67686	A Midsummer nights dream	3.0	2.2.33	OBERON	
1139	Henry IV	52.0	2.4.138	FALSTAFF	
50668	King Lear	70.0	2.4.245	KING LEAR	
77175	Pericles	36.0	3.2.125	CERIMON	

  

	PlayerLine
16895	[thou, art, seen]
110371	[good, camillo]
63004	[ever, blotted, paper, gentle, lady]
68185	[thou, holdst, thy, hand, let, kiss]
77634	[like, one, superstitiously]
...	...
61721	[prophet, nazarete, conjured, devil]
67686	[thou, wakest, thy, dear]
1139	[call, thee, coward, ill, see, thee, damned, e...]
50668	[plaguesore, embossed, carbuncle]
77175	[lend, hands, next, chamber, bear]

[84000 rows x 5 columns]

### 0.3.1 Mindless Musings

The following are just mindless things I did when brainstorming for this project. I kept them in case I wanted to reference them in future projects.

Lets just look at the play and the player to find the most popular player's for each play. You can then say, if I am in play A, this player will speak the most.

```
[24]: play_player = data.drop(columns="PlayerLinenumber")
      play_player = play_player.drop(columns="ActSceneLine")
      play_player = play_player.drop(columns="PlayerLine")
      play_player = play_player[play_player["Play"] == "Henry IV"]
```

```
[25]: play_player
```

```
[25]:
```

	Play	Player
3	Henry IV	KING HENRY IV
4	Henry IV	KING HENRY IV
5	Henry IV	KING HENRY IV
6	Henry IV	KING HENRY IV
7	Henry IV	KING HENRY IV
...	...	...
3199	Henry IV	KING HENRY IV
3200	Henry IV	KING HENRY IV
3201	Henry IV	KING HENRY IV
3202	Henry IV	KING HENRY IV
3203	Henry IV	KING HENRY IV

[3044 rows x 2 columns]

count\_player is a list of the most common Players in the Shakespear Plays. Could say value\_counts(normalize=True) to normalize all the values.

```
[26]: count_player = data_train['Player'].value_counts()
      count_player
```

```
[26]: GLOUCESTER      1433
      HAMLET         1197
      IAGO           888
      FALSTAFF       869
      KING HENRY V   818
      ...
      HORTENSIA      1
      Outlaws        1
      HERNIA         1
```

```
Third Musician      1
All The Lords       1
Name: Player, Length: 917, dtype: int64
```

We can also put the counts into bins to better see the distrubtion of how many times a player speaks.

```
[27]: count_player.value_counts(bins=20)
```

```
[27]: (-0.433, 72.6]      614
      (72.6, 144.2]      110
      (144.2, 215.8]      70
      (215.8, 287.4]      58
      (287.4, 359.0]      16
      (359.0, 430.6]      10
      (430.6, 502.2]      11
      (502.2, 573.8]      11
      (573.8, 645.4]       5
      (645.4, 717.0]       6
      (717.0, 788.6]       1
      (788.6, 860.2]       1
      (860.2, 931.8]       2
      (931.8, 1003.4]      0
      (1003.4, 1075.0]     0
      (1075.0, 1146.6]     0
      (1146.6, 1218.2]     1
      (1218.2, 1289.8]     0
      (1289.8, 1361.4]     0
      (1361.4, 1433.0]     1
      (1433.0, 1504.6]     0
      (1504.6, 1576.2]     0
      (1576.2, 1647.8]     0
      (1647.8, 1719.4]     0
      (1719.4, 1791.0]     0
      (1791.0, 1862.6]     0
      (1862.6, 1934.2]     0
      (1934.2, 2005.8]     0
      (2005.8, 2077.4]     0
      (2077.4, 2149.0]     0
      (2149.0, 2220.6]     0
      (2220.6, 2292.2]     0
      (2292.2, 2363.8]     0
      (2363.8, 2435.4]     0
      (2435.4, 2507.0]     0
      (2507.0, 2578.6]     0
      (2578.6, 2650.2]     0
      (2650.2, 2721.8]     0
      (2721.8, 2793.4]     0
      (2793.4, 2865.0]     0
      (2865.0, 2936.6]     0
      (2936.6, 3008.2]     0
      (3008.2, 3079.8]     0
      (3079.8, 3151.4]     0
      (3151.4, 3223.0]     0
      (3223.0, 3294.6]     0
      (3294.6, 3366.2]     0
      (3366.2, 3437.8]     0
      (3437.8, 3509.4]     0
      (3509.4, 3581.0]     0
      (3581.0, 3652.6]     0
      (3652.6, 3724.2]     0
      (3724.2, 3795.8]     0
      (3795.8, 3867.4]     0
      (3867.4, 3939.0]     0
      (3939.0, 4010.6]     0
      (4010.6, 4082.2]     0
      (4082.2, 4153.8]     0
      (4153.8, 4225.4]     0
      (4225.4, 4297.0]     0
      (4297.0, 4368.6]     0
      (4368.6, 4440.2]     0
      (4440.2, 4511.8]     0
      (4511.8, 4583.4]     0
      (4583.4, 4655.0]     0
      (4655.0, 4726.6]     0
      (4726.6, 4798.2]     0
      (4798.2, 4869.8]     0
      (4869.8, 4941.4]     0
      (4941.4, 5013.0]     0
      (5013.0, 5084.6]     0
      (5084.6, 5156.2]     0
      (5156.2, 5227.8]     0
      (5227.8, 5299.4]     0
      (5299.4, 5371.0]     0
      (5371.0, 5442.6]     0
      (5442.6, 5514.2]     0
      (5514.2, 5585.8]     0
      (5585.8, 5657.4]     0
      (5657.4, 5729.0]     0
      (5729.0, 5800.6]     0
      (5800.6, 5872.2]     0
      (5872.2, 5943.8]     0
      (5943.8, 6015.4]     0
      (6015.4, 6087.0]     0
      (6087.0, 6158.6]     0
      (6158.6, 6230.2]     0
      (6230.2, 6301.8]     0
      (6301.8, 6373.4]     0
      (6373.4, 6445.0]     0
      (6445.0, 6516.6]     0
      (6516.6, 6588.2]     0
      (6588.2, 6659.8]     0
      (6659.8, 6731.4]     0
      (6731.4, 6803.0]     0
      (6803.0, 6874.6]     0
      (6874.6, 6946.2]     0
      (6946.2, 7017.8]     0
      (7017.8, 7089.4]     0
      (7089.4, 7161.0]     0
      (7161.0, 7232.6]     0
      (7232.6, 7304.2]     0
      (7304.2, 7375.8]     0
      (7375.8, 7447.4]     0
      (7447.4, 7519.0]     0
      (7519.0, 7590.6]     0
      (7590.6, 7662.2]     0
      (7662.2, 7733.8]     0
      (7733.8, 7805.4]     0
      (7805.4, 7877.0]     0
      (7877.0, 7948.6]     0
      (7948.6, 8020.2]     0
      (8020.2, 8091.8]     0
      (8091.8, 8163.4]     0
      (8163.4, 8235.0]     0
      (8235.0, 8306.6]     0
      (8306.6, 8378.2]     0
      (8378.2, 8449.8]     0
      (8449.8, 8521.4]     0
      (8521.4, 8593.0]     0
      (8593.0, 8664.6]     0
      (8664.6, 8736.2]     0
      (8736.2, 8807.8]     0
      (8807.8, 8879.4]     0
      (8879.4, 8951.0]     0
      (8951.0, 9022.6]     0
      (9022.6, 9094.2]     0
      (9094.2, 9165.8]     0
      (9165.8, 9237.4]     0
      (9237.4, 9309.0]     0
      (9309.0, 9380.6]     0
      (9380.6, 9452.2]     0
      (9452.2, 9523.8]     0
      (9523.8, 9595.4]     0
      (9595.4, 9667.0]     0
      (9667.0, 9738.6]     0
      (9738.6, 9810.2]     0
      (9810.2, 9881.8]     0
      (9881.8, 9953.4]     0
      (9953.4, 10025.0]     0
      (10025.0, 10096.6]     0
      (10096.6, 10168.2]     0
      (10168.2, 10239.8]     0
      (10239.8, 10311.4]     0
      (10311.4, 10383.0]     0
      (10383.0, 10454.6]     0
      (10454.6, 10526.2]     0
      (10526.2, 10597.8]     0
      (10597.8, 10669.4]     0
      (10669.4, 10741.0]     0
      (10741.0, 10812.6]     0
      (10812.6, 10884.2]     0
      (10884.2, 10955.8]     0
      (10955.8, 11027.4]     0
      (11027.4, 11099.0]     0
      (11099.0, 11170.6]     0
      (11170.6, 11242.2]     0
      (11242.2, 11313.8]     0
      (11313.8, 11385.4]     0
      (11385.4, 11457.0]     0
      (11457.0, 11528.6]     0
      (11528.6, 11600.2]     0
      (11600.2, 11671.8]     0
      (11671.8, 11743.4]     0
      (11743.4, 11815.0]     0
      (11815.0, 11886.6]     0
      (11886.6, 11958.2]     0
      (11958.2, 12029.8]     0
      (12029.8, 12101.4]     0
      (12101.4, 12173.0]     0
      (12173.0, 12244.6]     0
      (12244.6, 12316.2]     0
      (12316.2, 12387.8]     0
      (12387.8, 12459.4]     0
      (12459.4, 12531.0]     0
      (12531.0, 12602.6]     0
      (12602.6, 12674.2]     0
      (12674.2, 12745.8]     0
      (12745.8, 12817.4]     0
      (12817.4, 12889.0]     0
      (12889.0, 12960.6]     0
      (12960.6, 13032.2]     0
      (13032.2, 13103.8]     0
      (13103.8, 13175.4]     0
      (13175.4, 13247.0]     0
      (13247.0, 13318.6]     0
      (13318.6, 13390.2]     0
      (13390.2, 13461.8]     0
      (13461.8, 13533.4]     0
      (13533.4, 13605.0]     0
      (13605.0, 13676.6]     0
      (13676.6, 13748.2]     0
      (13748.2, 13819.8]     0
      (13819.8, 13891.4]     0
      (13891.4, 13963.0]     0
      (13963.0, 14034.6]     0
      (14034.6, 14106.2]     0
      (14106.2, 14177.8]     0
      (14177.8, 14249.4]     0
      (14249.4, 14321.0]     0
      (14321.0, 14392.6]     0
      (14392.6, 14464.2]     0
      (14464.2, 14535.8]     0
      (14535.8, 14607.4]     0
      (14607.4, 14679.0]     0
      (14679.0, 14750.6]     0
      (14750.6, 14822.2]     0
      (14822.2, 14893.8]     0
      (14893.8, 14965.4]     0
      (14965.4, 15037.0]     0
      (15037.0, 15108.6]     0
      (15108.6, 15180.2]     0
      (15180.2, 15251.8]     0
      (15251.8, 15323.4]     0
      (15323.4, 15395.0]     0
      (15395.0, 15466.6]     0
      (15466.6, 15538.2]     0
      (15538.2, 15609.8]     0
      (15609.8, 15681.4]     0
      (15681.4, 15753.0]     0
      (15753.0, 15824.6]     0
      (15824.6, 15896.2]     0
      (15896.2, 15967.8]     0
      (15967.8, 16039.4]     0
      (16039.4, 16111.0]     0
      (16111.0, 16182.6]     0
      (16182.6, 16254.2]     0
      (16254.2, 16325.8]     0
      (16325.8, 16397.4]     0
      (16397.4, 16469.0]     0
      (16469.0, 16540.6]     0
      (16540.6, 16612.2]     0
      (16612.2, 16683.8]     0
      (16683.8, 16755.4]     0
      (16755.4, 16827.0]     0
      (16827.0, 16898.6]     0
      (16898.6, 16970.2]     0
      (16970.2, 17041.8]     0
      (17041.8, 17113.4]     0
      (17113.4, 17185.0]     0
      (17185.0, 17256.6]     0
      (17256.6, 17328.2]     0
      (17328.2, 17400.0]     0
      (17400.0, 17471.8]     0
      (17471.8, 17543.6]     0
      (17543.6, 17615.4]     0
      (17615.4, 17687.0]     0
      (17687.0, 17758.6]     0
      (17758.6, 17830.2]     0
      (17830.2, 17901.8]     0
      (17901.8, 17973.4]     0
      (17973.4, 18045.0]     0
      (18045.0, 18116.6]     0
      (18116.6, 18188.2]     0
      (18188.2, 18259.8]     0
      (18259.8, 18331.4]     0
      (18331.4, 18403.0]     0
      (18403.0, 18474.6]     0
      (18474.6, 18546.2]     0
      (18546.2, 18617.8]     0
      (18617.8, 18689.4]     0
      (18689.4, 18761.0]     0
      (18761.0, 18832.6]     0
      (18832.6, 18904.2]     0
      (18904.2, 18975.8]     0
      (18975.8, 19047.4]     0
      (19047.4, 19119.0]     0
      (19119.0, 19190.6]     0
      (19190.6, 19262.2]     0
      (19262.2, 19333.8]     0
      (19333.8, 19405.4]     0
      (19405.4, 19477.0]     0
      (19477.0, 19548.6]     0
      (19548.6, 19620.2]     0
      (19620.2, 19691.8]     0
      (19691.8, 19763.4]     0
      (19763.4, 19835.0]     0
      (19835.0, 19906.6]     0
      (19906.6, 19978.2]     0
      (19978.2, 20049.8]     0
      (20049.8, 20121.4]     0
      (20121.4, 20193.0]     0
      (20193.0, 20264.6]     0
      (20264.6, 20336.2]     0
      (20336.2, 20407.8]     0
      (20407.8, 20479.4]     0
      (20479.4, 20551.0]     0
      (20551.0, 20622.6]     0
      (20622.6, 20694.2]     0
      (20694.2, 20765.8]     0
      (20765.8, 20837.4]     0
      (20837.4, 20909.0]     0
      (20909.0, 20980.6]     0
      (20980.6, 21052.2]     0
      (21052.2, 21123.8]     0
      (21123.8, 21195.4]     0
      (21195.4, 21267.0]     0
      (21267.0, 21338.6]     0
      (21338.6, 21410.2]     0
      (21410.2, 21481.8]     0
      (21481.8, 21553.4]     0
      (21553.4, 21625.0]     0
      (21625.0, 21696.6]     0
      (21696.6, 21768.2]     0
      (21768.2, 21839.8]     0
      (21839.8, 21911.4]     0
      (21911.4, 21983.0]     0
      (21983.0, 22054.6]     0
      (22054.6, 22126.2]     0
      (22126.2, 22197.8]     0
      (22197.8, 22269.4]     0
      (22269.4, 22341.0]     0
      (22341.0, 22412.6]     0
      (22412.6, 22484.2]     0
      (22484.2, 22555.8]     0
      (22555.8, 22627.4]     0
      (22627.4, 22699.0]     0
      (22699.0, 22770.6]     0
      (22770.6, 22842.2]     0
      (22842.2, 22913.8]     0
      (22913.8, 22985.4]     0
      (22985.4, 23057.0]     0
      (23057.0, 23128.6]     0
      (23128.6, 23200.2]     0
      (23200.2, 23271.8]     0
      (23271.8, 23343.4]     0
      (23343.4, 23415.0]     0
      (23415.0, 23486.6]     0
      (23486.6, 23558.2]     0
      (23558.2, 23629.8]     0
      (23629.8, 23701.4]     0
      (23701.4, 23773.0]     0
      (23773.0, 23844.6]     0
      (23844.6, 23916.2]     0
      (23916.2, 23987.8]     0
      (23987.8, 24059.4]     0
      (24059.4, 24131.0]     0
      (24131.0, 24202.6]     0
      (24202.6, 24274.2]     0
      (24274.2, 24345.8]     0
      (24345.8, 24417.4]     0
      (24417.4, 24489.0]     0
      (24489.0, 24560.6]     0
      (24560.6, 24632.2]     0
      (24632.2, 24703.8]     0
      (24703.8, 24775.4]     0
      (24775.4, 24847.0]     0
      (24847.0, 24918.6]     0
      (24918.6, 24990.2]     0
      (24990.2, 25061.8]     0
      (25061.8, 25133.4]     0
      (25133.4, 25205.0]     0
      (25205.0, 25276.6]     0
      (25276.6, 25348.2]     0
      (25348.2, 25419.8]     0
      (25419.8, 25491.4]     0
      (25491.4, 25563.0]     0
      (25563.0, 25634.6]     0
      (25634.6, 25706.2]     0
      (25706.2, 25777.8]     0
      (25777.8, 25849.4]     0
      (25849.4, 25921.0]     0
      (25921.0, 25992.6]     0
      (25992.6, 26064.2]     0
      (26064.2, 26135.8]     0
      (26135.8, 26207.4]     0
      (26207.4, 26279.0]     0
      (26279.0, 26350.6]     0
      (26350.6, 26422.2]     0
      (26422.2, 26493.8]     0
      (26493.8, 26565.4]     0
      (26565.4, 26637.0]     0
      (26637.0, 26708.6]     0
      (26708.6, 26780.2]     0
      (26780.2, 26851.8]     0
      (26851.8, 26923.4]     0
      (26923.4, 26995.0]     0
      (26995.0, 27066.6]     0
      (27066.6, 27138.2]     0
      (27138.2, 27209.8]     0
      (27209.8, 27281.4]     0
      (27281.4, 27353.0]     0
      (27353.0, 27424.6]     0
      (27424.6, 27496.2]     0
      (27496.2, 27567.8]     0
      (27567.8, 27639.4]     0
      (27639.4, 27711.0]     0
      (27711.0, 27782.6]     0
      (27782.6, 27854.2]     0
      (27854.2, 27925.8]     0
      (27925.8, 28000.0]     0
      (28000.0, 28071.8]     0
      (28071.8, 28143.6]     0
      (28143.6, 28215.4]     0
      (28215.4, 28287.0]     0
      (28287.0, 28358.6]     0
      (28358.6, 28430.2]     0
      (28430.2, 28501.8]     0
      (28501.8, 28573.4]     0
      (28573.4, 28645.0]     0
      (28645.0, 28716.6]     0
      (28716.6, 28788.2]     0
      (28788.2, 28859.8]     0
      (28859.8, 28931.4]     0
      (28931.4, 29003.0]     0
      (29003.0, 29074.6]     0
      (29074.6, 29146.2]     0
      (29146.2, 29217.8]     0
      (29217.8, 29289.4]     0
      (29289.4, 29361.0]     0
      (29361.0, 29432.6]     0
      (29432.6, 29504.2]     0
      (29504.2, 29575.8]     0
      (29575.8, 29647.4]     0
      (29647.4, 29719.0]     0
      (29719.0, 29790.6]     0
      (29790.6, 29862.2]     0
      (29862.2, 29933.8]     0
      (29933.8, 30005.4]     0
      (30005.4, 30077.0]     0
      (30077.0, 30148.6]     0
      (30148.6, 30220.2]     0
      (30220.2, 30291.8]     0
      (30291.8, 30363.4]     0
      (30363.4, 30435.0]     0
      (30435.0, 30506.6]     0
      (30506.6, 30578.2]     0
      (30578.2, 30649.8]     0
      (30649.8, 30721.4]     0
      (30721.4, 30793.0]     0
      (30793.0, 30864.6]     0
      (30864.6, 30936.2]     0
      (30936.2, 31007.8]     0
      (31007.8, 31079.4]     0
      (31079.4, 31151.0]     0
      (31151.0, 31222.6]     0
      (31222.6, 31294.2]     0
      (31294.2, 31365.8]     0
      (31365.8, 31437.4]     0
      (31437.4, 31509.0]     0
      (31509.0, 31580.6]     0
      (31580.6, 31652.2]     0
      (31652.2, 31723.8]     0
      (31723.8, 31795.4]     0
      (31795.4, 31867.0]     0
      (31867.0, 31938.6]     0
      (31938.6, 32010.2]     0
      (32010.2, 32081.8]     0
      (32081.8, 32153.4]     0
      (32153.4, 32225.0]     0
      (32225.0, 32296.6]     0
      (32296.6, 32368.2]     0
      (32368.2, 32439.8]     0
      (32439.8, 32511.4]     0
      (32511.4, 32583.0]     0
      (32583.0, 32654.6]     0
      (32654.6, 32726.2]     0
      (32726.2, 32797.8]     0
      (32797.8, 32869.4]     0
      (32869.4, 32941.0]     0
      (32941.0, 33012.6]     0
      (33012.6, 33084.2]     0
      (33084.2, 33155.8]     0
      (33155.8, 33227.4]     0
      (33227.4, 33299.0]     0
      (33299.0, 33370.6]     0
      (33370.6, 33442.2]     0
      (33442.2, 33513.8]     0
      (33513.8, 33585.4]     0
      (33585.4, 33657.0]     0
      (33657.0, 33728.6]     0
      (33728.6, 33800.2]     0
      (33800.2, 33871.8]     0
      (33871.8, 33943.4]     0
      (33943.4, 34015.0]     0
      (34015.0, 34086.6]     0
      (34086.6, 34158.2]     0
      (34158.2, 34229.8]     0
      (34229.8, 34301.4]     0
      (34301.4, 34373.0]     0
      (34373.0, 34444.6]     0
      (34444.6, 34516.2]     0
      (34516.2, 34587.8]     0
      (34587.8, 34659.4]     0
      (34659.4, 34731.0]     0
      (34731.0, 34802.6]     0
      (34802.6, 34874.2]     0
      (34874.2, 34945.8]     0
      (34945.8, 35017.4]     0
      (35017.4, 35089.0]     0
      (35089.0, 35160.6]     0
      (35160.6, 35232.2]     0
      (35232.2, 35303.8]     0
      (35303.8, 35375.4]     0
      (35375.4, 35447.0]     0
      (35447.0, 35518.6]     0
      (35518.6, 35590.2]     0
      (35590.2, 35661.8]     0
      (35661.8, 35733.4]     0
      (35733.4, 35805.0]     0
      (35805.0, 35876.6]     0
      (35876.6, 35948.2]     0
      (35948.2, 36019.8]     0
      (36019.8, 36091.4]     0
      (36091.4, 36163.0]     0
      (36163.0, 36234.6]     0
      (36234.6, 36306.2]     0
      (36306.2, 36377.8]     0
      (36377.8, 36449.4]     0
      (36449.4, 36521.0]     0
      (36521.0, 36592.6]     0
      (36592.6, 36664.2]     0
      (36664.2, 36735.8]     0
      (36735.8, 36807.4]     0
      (36807.4, 36879.0]     0
      (36879.0, 36950.6]     0
      (36950.6, 37022.2]     0
      (37022.2, 37093.8]     0
      (37093.8, 37165.4]     0
      (37165.4, 37237.0]     0
      (37237.0, 37308.6]     0
      (37308.6, 37380.2]     0
      (37380.2, 37451.8]     0
      (37451.8, 37523.4]     0
      (37523.4, 37595.0]     0
      (37595.0, 37666.6]     0
      (37666.6, 37738.2]     0
      (37738.2, 37809.8]     0
      (37809.8, 37881.4]     0
      (37881.4, 37953.0]     0
      (37953.0, 38024.6]     0
      (38024.6, 38096.2]     0
      (38096.2, 38167.8]     0
      (38167.8, 38239.4]     0
      (38239.4, 38311.0]     0
      (38311.0, 38382.6]     0
      (38382.6, 38454.2]     0
      (38454.2, 38525.8]     0
      (38525.8, 38597.4]     0
      (38597.4, 38669.0]     0
      (38669.0, 38740.6]     0
      (38740.6, 38812.2]     0
      (38812.2, 38883.8]     0
      (38
```