In [2]:

*# This Python 3 environment comes with many helpful analytics libraries installed*

*# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python # For example, here's several helpful packages to load*

**import** numpy **as** np *# linear algebra*

**import** pandas **as** pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

*# Input data files are available in the read-only "../input/" directory*

*# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory*

**import** os

**for** dirname, \_, filenames **in** os**.**walk('/kaggle/input'):

**for** filename **in** filenames:

print(os**.**path**.**join(dirname, filename))

*# You can write up to 5GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version us # You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session*

In [3]:

df\_train **=** pd**.**read\_csv('train.csv') df\_test **=** pd**.**read\_csv('test.csv') print(df\_train**.**shape)

print(df\_test**.**shape)

(27481, 4)

(3534, 3)

In [4]:

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**import** plotly.figure\_factory **as** ff

**import** re

**import** string

**from** collections **import** Counter

**import** plotly.express **as** px

**from** nltk.corpus **import** stopwords

**import** spacy

In [5]:

df\_train**.**dropna(inplace**=True**) df\_train**.**info()

<class 'pandas.core.frame.DataFrame'> Int64Index: 27480 entries, 0 to 27480 Data columns (total 4 columns):

# Column Non-Null Count Dtype

1. textID 27480 non-null object
2. text 27480 non-null object
3. selected\_text 27480 non-null object
4. sentiment 27480 non-null object dtypes: object(4)

memory usage: 1.0+ MB

In [6]:

df\_test**.**info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 3534 entries, 0 to 3533

Data columns (total 3 columns):

# Column Non-Null Count Dtype

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | textID | 3534 | non-null |  | object |
| 1 |  | text | 3534 | non-null |  | object |
| 2 |  | sentiment | 3534 | non-null |  | object |

|  |  |  |
| --- | --- | --- |
|  | | dtypes: object(3)  memory usage: 83.0+ KB |
| Exploratory Data Analysis |
| In | [7]: | df\_train**.**head() |

Out[7]:

**textID text selected\_text sentiment**

**0** cb774db0d1 I`d have responded, if I were going I`d have responded, if I were going neutral

**1** 549e992a42 Sooo SAD I will miss you here in San Diego!!! Sooo SAD negative

**2** 088c60f138 my boss is bullying me... bullying me negative

**3** 9642c003ef what interview! leave me alone leave me alone negative

**4** 358bd9e861 Sons of \*\*\*\*, why couldn`t they put them on t... Sons of \*\*\*\*, negative

In [8]:

df\_train**.**tail()

Out[8]:

**textID text selected\_text sentiment**

**27476** 4eac33d1c0 wish we could come see u on Denver husband l... d lost negative

**27477** 4f4c4fc327 I`ve wondered about rake to. The client has ... , don`t force negative

**27478** f67aae2310 Yay good for both of you. Enjoy the break - y... Yay good for both of you. positive

**27479** ed167662a5 But it was worth it \*\*\*\*. But it was worth it \*\*\*\*. positive

**27480** 6f7127d9d7 All this flirting going on - The ATG smiles... All this flirting going on - The ATG smiles. Y... neutral

In [9]:

temp **=** df\_train**.**groupby('sentiment')**.**count()['text']**.**reset\_index()**.**sort\_values(by**=**'text', ascending**=False**) temp

|  |  |  |  |
| --- | --- | --- | --- |
| Out[9]: |  | **sentiment** | **text** |
|  | **1** | neutral | 11117 |
|  | **2** | positive | 8582 |
|  | **0** | negative | 7781 |

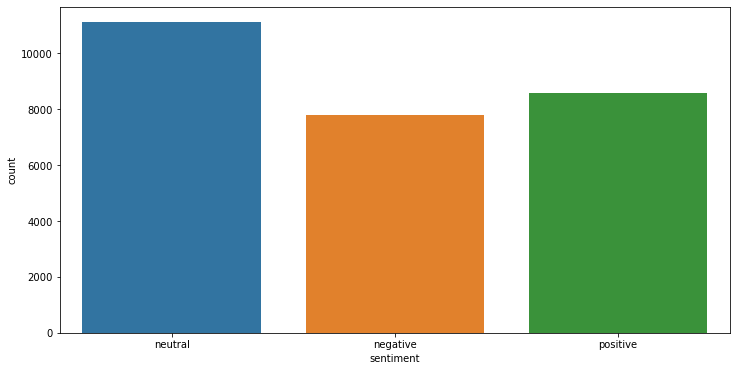
In [10]:

plt**.**figure(figsize**=**(12, 6))

sns**.**countplot(x**=**'sentiment', data**=**df\_train)

Out[10]:

<AxesSubplot:xlabel='sentiment', ylabel='count'>



In [11]:

**def** jaccard(str1, str2):

a **=** set(str1**.**lower()**.**split()) b **=** set(str2**.**lower()**.**split()) c **=** a**.**intersection(b)

**return** float(len(c))**/**len(a) **+** len(b) **-** len(c)

In [12]:

results\_jaccard **=** []

**for** ind, row **in** df\_train**.**iterrows(): sent1 **=** row**.**text

sent2 **=** row**.**selected\_text

jaccard\_score **=** jaccard(sent1, sent2)

results\_jaccard**.**append([sent1, sent2, jaccard\_score])

In [13]:

df\_jaccard **=** pd**.**DataFrame(results\_jaccard, columns**=**['text', 'selected\_text', 'jaccard\_score']) df\_train **=** df\_train**.**merge(df\_jaccard, how**=**'outer')

In [14]:

*# Number of words in selected text*

df\_train['Num\_words\_ST'] **=** df\_train['selected\_text']**.**apply(**lambda** x:len(str(x)**.**split()))

*# Number of words in text*

df\_train['Num\_words\_Text'] **=** df\_train['text']**.**apply(**lambda** x:len(str(x)**.**split()))

*# Difference in number of words in text and selected text*

df\_train['difference\_in\_words'] **=** df\_train['Num\_words\_Text'] **-** df\_train['Num\_words\_ST']

In [15]:

df\_train**.**head()

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[15]: | **textID** | **text** | **selected\_text** | **sentiment** | **jaccard\_score** | **Num\_words\_ST** | **Num\_words\_Text** | **difference\_in\_words** |

**0** cb774db0d1

I`d have responded, if I were

going

I`d have responded, if I neutral 1.000000 7 7 0

were going

**1** 549e992a42

Sooo SAD I will miss you here

in San Diego!!!

Sooo SAD negative

0.200000

2

10

8

**2** 088c60f138 my boss is bullying me... bullying me negative 1.200000 2 5 3

**3** 9642c003ef what interview! leave me alone

leave me alone negative

0.600000

3

5

2

**4** 358bd9e861 Sons of \*\*\*\*, why couldn`t they

put them on t...

Sons of \*\*\*\*, negative 0.214286 3 14 11

In [16]:

hist\_data **=** [df\_train['Num\_words\_ST'], df\_train['Num\_words\_Text']] group\_labels **=** ['Selected\_Text', 'Text']

*# Create displot with custom bin\_size*

fig **=** ff**.**create\_distplot(hist\_data, group\_labels, show\_curve**=False**) fig**.**update\_layout(title\_text**=**'Distribution of Number of words')

fig**.**update\_layout( autosize**=False**, width**=**900,

height**=**700

)

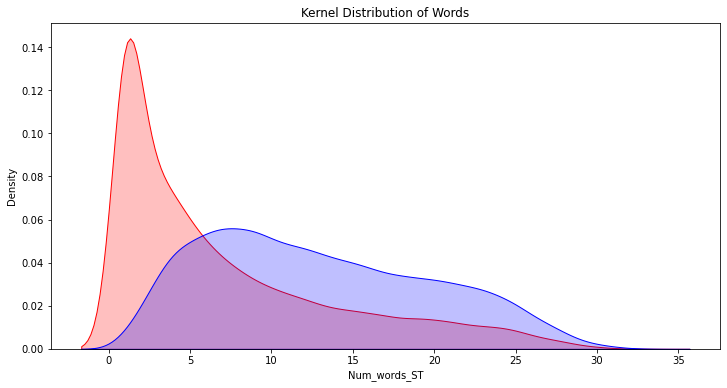
fig**.**show()

In [17]:

plt**.**figure(figsize**=**(12, 6))

p1 **=** sns**.**kdeplot(df\_train['Num\_words\_ST'], shade**=True**, color**=**'r')**.**set\_title('Kernel Distribution of Words')

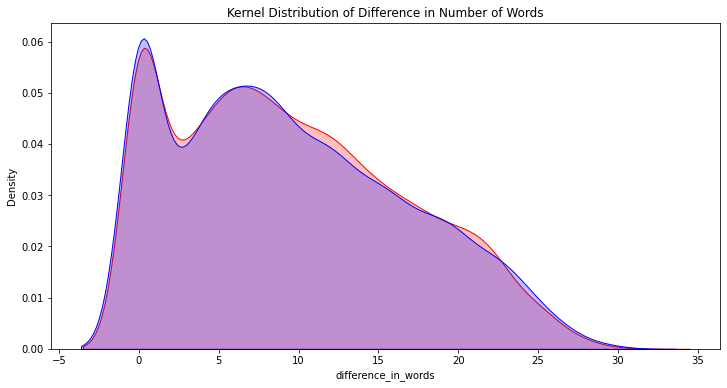
p1 **=** sns**.**kdeplot(df\_train['Num\_words\_Text'], shade**=True**, color**=**'b')



In [18]:

plt**.**figure(figsize**=**(12, 6))

p1 **=** sns**.**kdeplot(df\_train[df\_train['sentiment']**==**'positive']['difference\_in\_words'], shade**=True**, color**=**'r')**.**set\_title('Kernel Di p1 **=** sns**.**kdeplot(df\_train[df\_train['sentiment']**==**'negative']['difference\_in\_words'], shade**=True**, color**=**'b')



In [19]:

plt**.**figure(figsize**=**(12, 6))

sns**.**distplot(df\_train[df\_train['sentiment']**==**'neutral']['difference\_in\_words'], kde**=False**)

C:\Users\pratiksha\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a f igure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

Out[19]:

<AxesSubplot:xlabel='difference\_in\_words'>



In [20]:

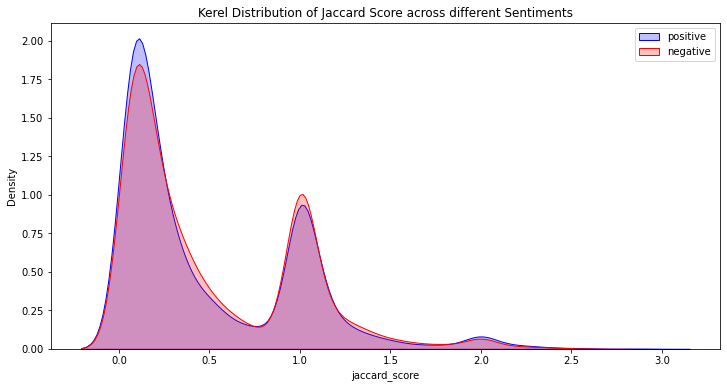
plt**.**figure(figsize**=**(12, 6))

p1 **=** sns**.**kdeplot(df\_train[df\_train['sentiment']**==**'positive']['jaccard\_score'], shade**=True**, color**=**'b')**.**set\_title('Kerel Distribut p1 **=** sns**.**kdeplot(df\_train[df\_train['sentiment']**==**'negative']['jaccard\_score'], shade**=True**, color**=**'r')

plt**.**legend(labels**=**['positive', 'negative'])

Out[20]:

<matplotlib.legend.Legend at 0x1cfa647d7c0>



In [21]:

plt**.**figure(figsize**=**(12, 6))

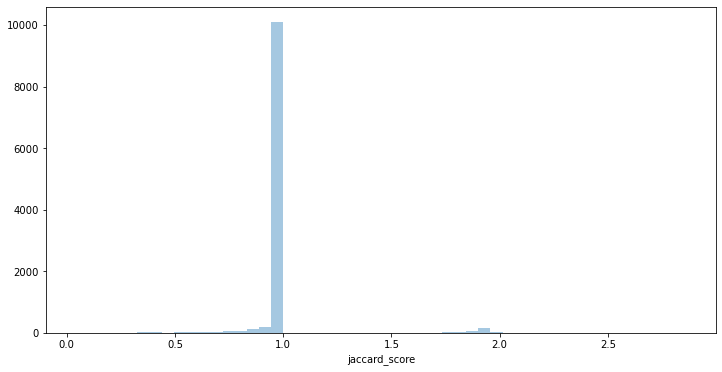
sns**.**distplot(df\_train[df\_train['sentiment']**==**'neutral']['jaccard\_score'], kde**=False**)

C:\Users\pratiksha\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a f igure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

Out[21]:

<AxesSubplot:xlabel='jaccard\_score'>



In [22]:

k **=** df\_train[df\_train['Num\_words\_Text']**<=**2]

In [23]:

Out[23]:

In [24]:

sentiment

k**.**groupby('sentiment')**.**mean()['jaccard\_score']

negative 0.856481

neutral 0.995157

positive 0.840580

Name: jaccard\_score, dtype: float64

k[k['sentiment']**==**'positive']

Out[24]:

**textID text selected\_text sentiment jaccard\_score Num\_words\_ST Num\_words\_Text dif**

**68** fa2654e730 Chilliin Chilliin positive 1.0 1 1

**80** bbbc46889b

THANK

THANK

YYYYYYYYYOOOOOOOOOOUUUUU! YYYYYYYYYOOOOOOOOOOUUUUU!

positive

1.0

2

2

**170** f3d95b57b1 good morning good morning positive 1.0 2 2

**278** 89d5b3f0b5

Thanks

Thanks

positive

1.0

1

1

**429** a78ef3e0d0 Goodmorning Goodmorning positive 1.0 1 1

**...**

...

...

...

...

...

...

...

**26689** e80c242d6a Goodnight; Goodnight; positive 1.0 1 1

**26725** aad244f37d

\*hug\*

\*hug\*

positive

1.0

1

1

**26842** a46571fe12 congrats! congrats! positive 1.0 1 1

**26959** 49a942e9b1

Happy birthday.

Happy birthday.

positive

1.0

2

2

**27292** 47c474aaf1 Good choice Good positive 0.5 1 2

207 rows × 8 columns

# Data Pre-Processing

In [25]:

**def** clean\_text(text):

'''Make text lowercase, remove text in square brackets,remove links,remove punctuation and remove words containing numbers.'''

text **=** str(text)**.**lower()

text **=** re**.**sub('\[.\*?\]', '', text)

text **=** re**.**sub('https?://\S+|www\.\S+', '', text) text **=** re**.**sub('<.\*?>+', '', text)

text **=** re**.**sub('[%s]' **%** re**.**escape(string**.**punctuation), '', text) text **=** re**.**sub('\n', '', text)

text **=** re**.**sub('\w\*\d\w\*', '', text)

**return** text

In [26]:

df\_train['text'] **=** df\_train['text']**.**apply(**lambda** x:clean\_text(x))

df\_train['selected\_text'] **=** df\_train['selected\_text']**.**apply(**lambda** x:clean\_text(x))

In [27]:

*# Most Common Word*

df\_train['temp\_list'] **=** df\_train['selected\_text']**.**apply(**lambda** x : str(x)**.**split()) temp **=** Counter(item **for** sublist **in** df\_train['temp\_list'] **for** item **in** sublist)

top **=** pd**.**DataFrame(temp**.**most\_common(20))

top**.**columns **=** ['Common\_words', 'count'] top

|  |  |  |  |
| --- | --- | --- | --- |
| Out[27]: |  | **Common\_words** | **count** |
|  | **0** | i | 7200 |
|  | **1** | to | 5305 |
|  | **2** | the | 4590 |
|  | **3** | a | 3538 |
|  | **4** | my | 2783 |
|  | **5** | you | 2624 |
|  | **6** | and | 2321 |
|  | **7** | it | 2158 |
|  | **8** | is | 2115 |
|  | **9** | in | 1986 |
|  | **10** | for | 1854 |
|  | **11** | im | 1676 |
|  | **12** | of | 1638 |
|  | **13** | me | 1540 |
|  | **14** | on | 1488 |
|  | **15** | so | 1410 |
|  | **16** | have | 1345 |
|  | **17** | that | 1297 |
|  | **18** | but | 1267 |
|  | **19** | good | 1251 |

In [28]:

fig **=** px**.**bar(top, x**=**'count', y**=**'Common\_words', title**=**'Common words in Selected Text', orientation**=**'h', width**=**700, height**=**700, co fig**.**show()

In [29]:

**def** remove\_stopwords(x):

**return** [y **for** y **in** x **if** y **not in** stopwords**.**words('english')]

df\_train['temp\_list'] **=** df\_train['temp\_list']**.**apply(**lambda** x : remove\_stopwords(x))

In [30]:

top **=** Counter(item **for** sublist **in** df\_train['temp\_list'] **for** item **in** sublist) temp **=** pd**.**DataFrame(top**.**most\_common(20))

temp**.**columns **=** ['Common\_words', 'count'] temp

|  |  |  |  |
| --- | --- | --- | --- |
| Out[30]: |  | **Common\_words** | **count** |
|  | **0** | im | 1676 |
|  | **1** | good | 1251 |
|  | **2** | day | 1058 |
|  | **3** | love | 909 |
|  | **4** | happy | 852 |
|  | **5** | like | 774 |
|  | **6** | get | 772 |
|  | **7** | dont | 765 |
|  | **8** | go | 700 |
|  | **9** | cant | 613 |
|  | **10** | work | 612 |
|  | **11** | going | 592 |
|  | **12** | today | 564 |
|  | **13** | got | 558 |
|  | **14** | one | 538 |
|  | **15** | time | 534 |
|  | **16** | thanks | 532 |
|  | **17** | lol | 528 |
|  | **18** | really | 520 |
|  | **19** | u | 519 |

In [31]:

fig **=** px**.**treemap(temp, path**=**['Common\_words'], values**=**'count', title**=**'Tree of the most Common Words') fig**.**show()

C:\Users\pratiksha\anaconda3\lib\site-packages\plotly\express\\_core.py:1637: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

In [32]:

*# Most Common Words*

df\_train['temp\_list1'] **=** df\_train['text']**.**apply(**lambda** x : str(x)**.**split())

df\_train['temp\_list1'] **=** df\_train['temp\_list1']**.**apply(**lambda** x : remove\_stopwords(x))

In [33]:

top **=** Counter([item **for** sublist **in** df\_train['temp\_list1'] **for** item **in** sublist]) temp **=** pd**.**DataFrame(top**.**most\_common(20))

temp**.**columns **=** ['Common\_words', 'Count'] temp

|  |  |  |  |
| --- | --- | --- | --- |
| Out[33]: |  | **Common\_words** | **Count** |
|  | **0** | im | 3020 |
|  | **1** | day | 2044 |
|  | **2** | good | 1549 |
|  | **3** | get | 1426 |
|  | **4** | like | 1346 |
|  | **5** | go | 1267 |
|  | **6** | dont | 1200 |
|  | **7** | love | 1122 |
|  | **8** | work | 1112 |
|  | **9** | going | 1096 |
|  | **10** | today | 1096 |
|  | **11** | got | 1072 |
|  | **12** | cant | 1020 |
|  | **13** | happy | 976 |
|  | **14** | one | 971 |
|  | **15** | lol | 948 |
|  | **16** | time | 942 |
|  | **17** | know | 930 |
|  | **18** | u | 923 |
|  | **19** | really | 908 |

In [34]:

fig **=** px**.**bar(temp, x**=**'Count', y**=**'Common\_words', title**=**'Common words in text', orientation**=**'h', width**=**700, height**=**700, color**=**'Com

fig**.**show()

In [35]:

Positive\_sent **=** df\_train[df\_train['sentiment']**==**'positive'] Negative\_sent **=** df\_train[df\_train['sentiment']**==**'negative'] Neutral\_sent **=** df\_train[df\_train['sentiment']**==**'neutral']

In [36]:

top **=** Counter([item **for** sublist **in** Positive\_sent['temp\_list'] **for** item **in** sublist]) temp **=** pd**.**DataFrame(top**.**most\_common(20))

temp**.**columns **=** ['Common\_words', 'Count']

temp

|  |  |  |  |
| --- | --- | --- | --- |
| Out[36]: |  | **Common\_words** | **Count** |
|  | **0** | good | 826 |
|  | **1** | happy | 730 |
|  | **2** | love | 697 |
|  | **3** | day | 456 |
|  | **4** | thanks | 439 |
|  | **5** | great | 364 |
|  | **6** | fun | 287 |
|  | **7** | nice | 267 |
|  | **8** | mothers | 259 |
|  | **9** | hope | 245 |
|  | **10** | awesome | 232 |
|  | **11** | im | 185 |
|  | **12** | thank | 180 |
|  | **13** | like | 167 |
|  | **14** | best | 154 |
|  | **15** | wish | 152 |
|  | **16** | amazing | 135 |
|  | **17** | really | 128 |
|  | **18** | better | 125 |
|  | **19** | cool | 119 |

In [37]:

fig **=** px**.**bar(temp, x**=**'Count', y**=**'Common\_words', title**=**'Most Common Positive Words', orientation**=**'h', width**=**700, height**=**700, colo fig**.**show()

In [38]:

top **=** Counter([item **for** sublist **in** Negative\_sent['temp\_list'] **for** item **in** sublist]) temp **=** pd**.**DataFrame(top**.**most\_common(20))

temp**.**columns **=** ['Common\_words', 'Count'] temp

|  |  |  |  |
| --- | --- | --- | --- |
| Out[38]: |  | **Common\_words** | **Count** |
|  | **0** | im | 452 |
|  | **1** | miss | 358 |
|  | **2** | sad | 343 |
|  | **3** | sorry | 300 |
|  | **4** | bad | 246 |
|  | **5** | hate | 230 |
|  | **6** | dont | 221 |
|  | **7** | cant | 201 |
|  | **8** | sick | 166 |
|  | **9** | like | 162 |
|  | **10** | sucks | 159 |
|  | **11** | feel | 158 |
|  | **12** | tired | 144 |
|  | **13** | really | 137 |
|  | **14** | good | 127 |
|  | **15** | bored | 115 |
|  | **16** | day | 110 |
|  | **17** | hurts | 108 |
|  | **18** | work | 99 |
|  | **19** | get | 97 |

In [39]:

fig **=** px**.**bar(temp, x**=**'Count', y**=**'Common\_words', title**=**'Most Common Negative Words', orientation**=**'h', width**=**700, height**=**700, colo fig**.**show()

In [40]:

top **=** Counter([item **for** sublist **in** Neutral\_sent['temp\_list'] **for** item **in** sublist]) temp **=** pd**.**DataFrame(top**.**most\_common(20))

temp**.**columns **=** ['Common\_words', 'Count'] temp

|  |  |  |  |
| --- | --- | --- | --- |
| Out[40]: |  | **Common\_words** | **Count** |
|  | **0** | im | 1039 |
|  | **1** | get | 612 |
|  | **2** | go | 569 |
|  | **3** | day | 492 |
|  | **4** | dont | 482 |
|  | **5** | going | 472 |
|  | **6** | work | 467 |
|  | **7** | like | 445 |
|  | **8** | got | 441 |
|  | **9** | today | 427 |
|  | **10** | lol | 427 |
|  | **11** | time | 413 |
|  | **12** | know | 407 |
|  | **13** | back | 402 |
|  | **14** | one | 394 |
|  | **15** | u | 376 |
|  | **16** | see | 349 |
|  | **17** | cant | 339 |
|  | **18** | home | 335 |
|  | **19** | want | 319 |

In [41]:

fig **=** px**.**bar(temp, x**=**'Count', y**=**'Common\_words', title**=**'Most Common Neutral Words', orientation**=**'h', width**=**700, height**=**700, color fig**.**show()

# Unique Words

In [42]:

raw\_text **=** [word **for** word\_list **in** df\_train['temp\_list'] **for** word **in** word\_list]

**def** words\_unique(sentiment, numwords, raw\_text):

allother **=** []

**for** sublist **in** df\_train[df\_train**.**sentiment **!=** sentiment]['temp\_list']:

**for** item **in** sublist:

allother**.**append(item) allother **=** set(allother)

specificonly **=** [x **for** x **in** raw\_text **if** x **not in** allother]

word\_count **=** Counter()

**for** sublist **in** df\_train[df\_train**.**sentiment **==** sentiment]['temp\_list']:

**for** item **in** sublist:

word\_count[item] **+=** 1

**for** word **in** list(word\_count):

**if** word **not in** specificonly:

**del** word\_count[word]

unique\_words **=** pd**.**DataFrame(word\_count**.**most\_common(numwords), columns**=**['word', 'count'])

**return** unique\_words

In [43]:

unique\_positive **=** words\_unique('positive', 20, raw\_text) print('Top 20 words in positive tweets are:')

unique\_positive

Top 20 words in positive tweets are:

|  |  |  |  |
| --- | --- | --- | --- |
| Out[43]: |  | **word** | **count** |
|  | **0** | congratulations | 26 |
|  | **1** | thnx | 8 |
|  | **2** | lov | 8 |
|  | **3** | happ | 7 |
|  | **4** | talented | 7 |
|  | **5** | brilliant | 7 |
|  | **6** | appreciated | 6 |
|  | **7** | goood | 6 |
|  | **8** | amazin | 5 |
|  | **9** | mommas | 5 |
|  | **10** | coolest | 5 |
|  | **11** | owe | 5 |
|  | **12** | blessings | 4 |
|  | **13** | mothersday | 4 |
|  | **14** | dayyyy | 4 |
|  | **15** | refreshing | 4 |
|  | **16** | goodluck | 4 |
|  | **17** | funniest | 4 |
|  | **18** | honored | 4 |
|  | **19** | amazi | 3 |

In [44]:

fig **=** px**.**treemap(unique\_positive, path**=**['word'], values**=**'count', title**=**'Tree of Unique Positive Words') fig**.**show()

C:\Users\pratiksha\anaconda3\lib\site-packages\plotly\express\\_core.py:1637: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

In [45]:

unique\_negative **=** words\_unique('negative', 20, raw\_text) print('Top 20 words in negative tweets are:')

unique\_negative

Top 20 words in negative tweets are:

|  |  |  |  |
| --- | --- | --- | --- |
| Out[45]: |  | **word** | **count** |
|  | **0** | ache | 10 |
|  | **1** | saddest | 7 |
|  | **2** | hated | 6 |
|  | **3** | weak | 6 |
|  | **4** | suffering | 6 |
|  | **5** | rly | 5 |
|  | **6** | devastated | 5 |
|  | **7** | pissing | 5 |
|  | **8** | allergic | 4 |
|  | **9** | cramps | 4 |
|  | **10** | slap | 4 |
|  | **11** | itchy | 4 |
|  | **12** | slammed | 4 |
|  | **13** | effing | 4 |
|  | **14** | regretting | 4 |
|  | **15** | fed | 4 |
|  | **16** | sob | 4 |
|  | **17** | mis | 4 |
|  | **18** | unlucky | 4 |
|  | **19** | asthma | 4 |

In [46]:

fig **=** px**.**treemap(unique\_negative, path**=**['word'], values**=**'count', title**=**'Tree of Unique Negative Words') fig**.**show()

C:\Users\pratiksha\anaconda3\lib\site-packages\plotly\express\\_core.py:1637: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

In [47]:

unique\_neutral **=** words\_unique('neutral', 20, raw\_text) print('Top 20 words in neutral tweets are:')

temp

Top 20 words in neutral tweets are:

|  |  |  |  |
| --- | --- | --- | --- |
| Out[47]: |  | **Common\_words** | **Count** |
|  | **0** | im | 1039 |
|  | **1** | get | 612 |
|  | **2** | go | 569 |
|  | **3** | day | 492 |
|  | **4** | dont | 482 |
|  | **5** | going | 472 |
|  | **6** | work | 467 |
|  | **7** | like | 445 |
|  | **8** | got | 441 |
|  | **9** | today | 427 |
|  | **10** | lol | 427 |
|  | **11** | time | 413 |
|  | **12** | know | 407 |
|  | **13** | back | 402 |
|  | **14** | one | 394 |
|  | **15** | u | 376 |
|  | **16** | see | 349 |
|  | **17** | cant | 339 |
|  | **18** | home | 335 |
|  | **19** | want | 319 |

In [48]:

fig **=** px**.**treemap(unique\_neutral, path**=**['word'], values**=**'count', title**=**'Tree of Unique Neutral Words') fig**.**show()

C:\Users\pratiksha\anaconda3\lib\site-packages\plotly\express\\_core.py:1637: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.