ChatGPT Sentiment Analysis Using ML and NLP

In [1]:
```
#importing the Dependinces
from nltk.util import pr
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import re
import nltk
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))
```

In [2]:
```
nltk.download('stopwords')
```

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

Out[2]:
```
True
```

Data collection

In [3]:
```
#data loading in pandas
data=pd.read_csv('file.csv')
```

In [4]:
```
#check first five rows
data.head()
```

Out[4]:

	Unnamed: 0	tweets	labels
0	0	ChatGPT: Optimizing Language Models for Dialog...	neutral
1	1	Try talking with ChatGPT, our new AI system wh...	good
2	2	ChatGPT: Optimizing Language Models for Dialog...	neutral
3	3	THRILLED to share that ChatGPT, our new model ...	good
4	4	As of 2 minutes ago, @OpenAI released their ne...	bad

In [5]:
```
#check last five rows
data.tail()
```

Out[5]:

	Unnamed: 0	tweets	labels
219289	219289	Other Software Projects Are Now Trying to Repl...	bad
219290	219290	I asked #ChatGPT to write a #NYE Joke for SEOs...	good
219291	219291	chatgpt is being disassembled until it can onl...	bad
219292	219292	2023 predictions by #chatGPT. Nothing really s...	bad
219293	219293	From ChatGPT, neat stuff https://t.co/qjjUF2Z2m0	neutral

In [6]:
```
#check shape
data.shape
```

Out[6]:
```
(219294, 3)
```

In [7]:
```
#check more infomation
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 219294 entries, 0 to 219293
Data columns (total 3 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   Unnamed: 0  219294 non-null  int64 
 1   tweets      219294 non-null  object
 2   labels      219294 non-null  object
dtypes: int64(1), object(2)
memory usage: 5.0+ MB
```

In [8]:
```
#check missing value
data.isnull().sum()
```

Out[8]:
```
Unnamed: 0    0
tweets        0
labels        0
dtype: int64
```

In [9]:
```
#check duplicated value in data set
data.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
#check unused columns
del data['Unnamed: 0']
```

In [11]:
```
#clean data set
def clean(text):
    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)
    text = [word for word in text.split(' ') if word not in stopwor
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["tweets"] = data["tweets"].apply(clean)
```

In [12]:
```
#after check clean dataset
data.head()
```

Out[12]:

	tweets	labels
0	chatgpt optim languag model dialogu openai	neutral
1	tri talk chatgpt new ai system optim dialogu f...	good
2	chatgpt optim languag model dialogu ai machin...	neutral
3	thrill share chatgpt new model optim dialog pu...	good
4	minut ago openai releas new chatgpt nnand use...	bad

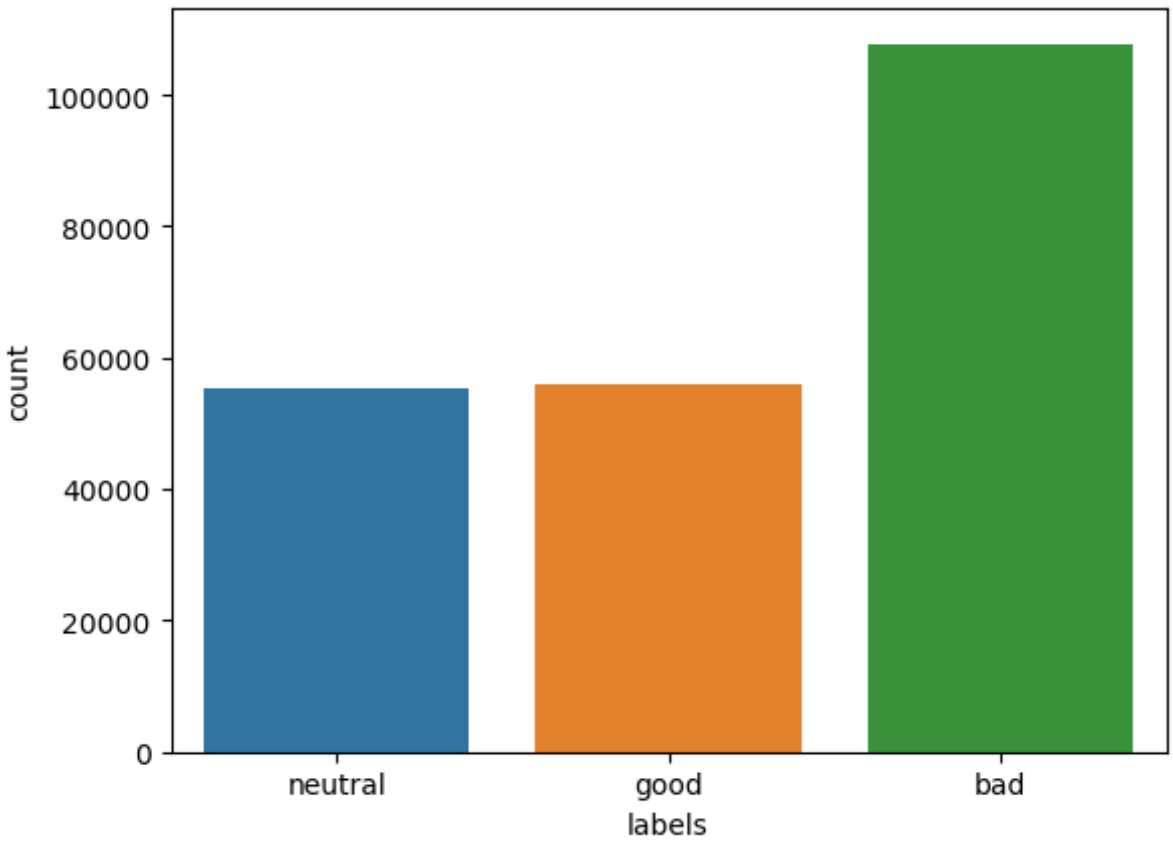
In [13]:
```
#check value of labels
data['labels'].value_counts()
```

Out[13]:
```
bad        107796
good        56011
neutral     55487
Name: labels, dtype: int64
```

In [14]:
```
import seaborn as sns
sns.countplot(data['labels'])
```

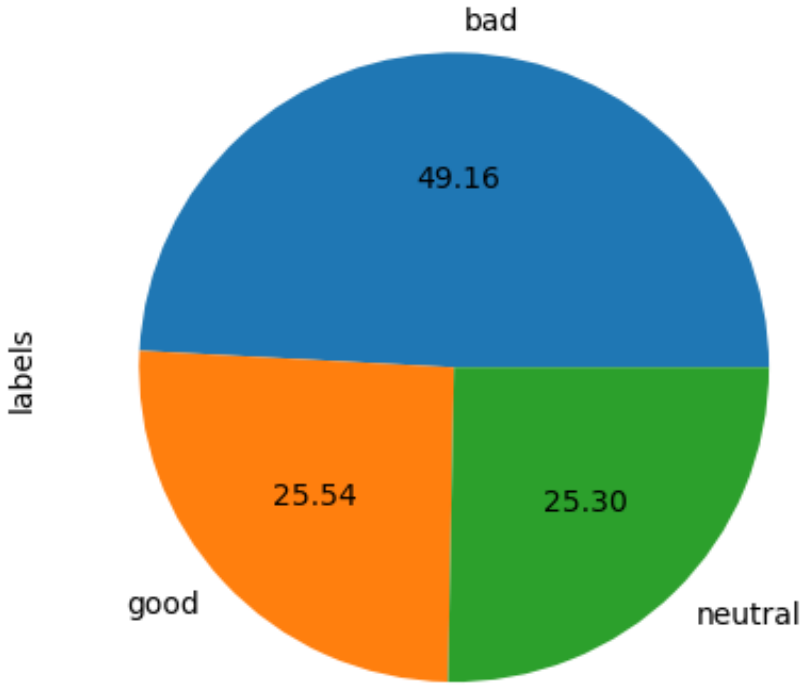
```
/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/seaborn/_de
corators.py:36: FutureWarning: Pass the following variable as a ke
yword arg: x. From version 0.12, the only valid positional argumen
t will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[14]: <AxesSubplot:xlabel='labels', ylabel='count'>

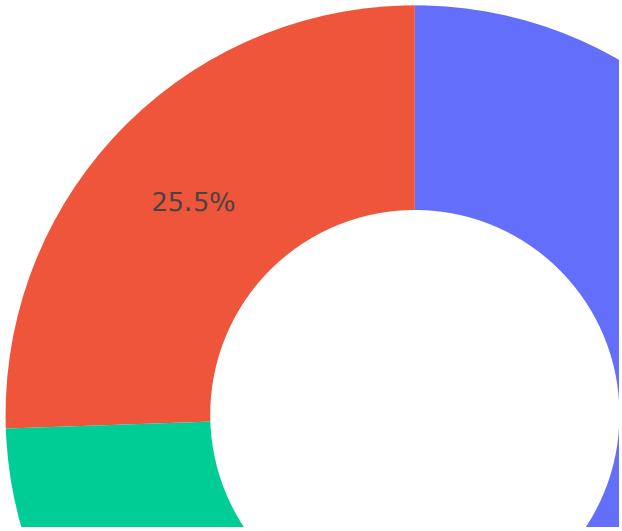


In [15]:
```
data['labels'].value_counts().plot(kind='pie',autopct='%.2f')
```

Out[15]: <AxesSubplot:ylabel='labels'>



In [16]:
```
labels = data["labels"].value_counts()
numbers = labels.index
quantity = labels.values

import plotly.express as px
figure = px.pie(data,
             values=quantity,
             names=numbers,hole = 0.5)
figure.show()
```



In [17]:
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

In [18]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /Users/vikky/opt/anaco
nda3/lib/python3.9/site-packages (1.8.2.2)
Requirement already satisfied: pillow in /Users/vikky/opt/anaconda
3/lib/python3.9/site-packages (from wordcloud) (9.2.0)
Requirement already satisfied: matplotlib in /Users/vikky/opt/anac
onda3/lib/python3.9/site-packages (from wordcloud) (3.5.2)
Requirement already satisfied: numpy>=1.6.1 in /Users/vikky/opt/an
aconda3/lib/python3.9/site-packages (from wordcloud) (1.21.5)
Requirement already satisfied: python-dateutil>=2.7 in /Users/vikk
y/opt/anaconda3/lib/python3.9/site-packages (from matplotlib->word
cloud) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/vikky/o
pt/anaconda3/lib/python3.9/site-packages (from matplotlib->wordclo
ud) (1.4.2)
Requirement already satisfied: pyparsing>=2.2.1 in /Users/vikky/op
t/anaconda3/lib/python3.9/site-packages (from matplotlib->wordclou
d) (3.0.9)
Requirement already satisfied: packaging>=20.0 in /Users/vikky/opt
/anaconda3/lib/python3.9/site-packages (from matplotlib->wordcloud
) (21.3)
Requirement already satisfied: fonttools>=4.22.0 in /Users/vikky/o
pt/anaconda3/lib/python3.9/site-packages (from matplotlib->wordclo
ud) (4.25.0)
Requirement already satisfied: cycler>=0.10 in /Users/vikky/opt/an
aconda3/lib/python3.9/site-packages (from matplotlib->wordcloud) (
0.11.0)
Requirement already satisfied: six>=1.5 in /Users/vikky/opt/anacon
da3/lib/python3.9/site-packages (from python-dateutil>=2.7->matplo
tlib->wordcloud) (1.16.0)

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: pip install --upgrade pip
```

In [19]:
```
#The labels column of the data contains the labels given by every re
text = " ".join(i for i in data.labels)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
                      background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

In [20]:
```
#The tweets column of the data contains the tweets given by every re
text = " ".join(i for i in data.tweets)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
                      background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

In [21]:
```
#spliting the dataset
X = data['tweets']
Y = data['labels']
```

In [22]:
```
#loading CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(X)
```

In [23]:
```
print(X)
```

```
  (0, 18069)    1
  (0, 87811)    1
  (0, 63492)    1
  (0, 73132)    1
  (0, 31060)    1
  (0, 87253)    1
  (1, 18069)    1
  (1, 87811)    1
  (1, 31060)    1
  (1, 122661)   1
  (1, 115953)   1
  (1, 77580)    1
  (1, 2194)     1
  (1, 115594)   1
```

```
(1, 40734)    1
(1, 51059)    1
(1, 125978)   1
(1, 55151)    1
(2, 18069)    1
(2, 87811)    1
(2, 63492)    1
(2, 73132)    1
(2, 31060)    1
(2, 2194)     1
(2, 67874)    1
:     :
(219290, 79731)       1
(219290, 82867)       1
(219290, 85574)       1
(219290, 49448)       1
(219290, 50721)       1
(219290, 77887)       1
(219291, 18069)       1
(219291, 31835)       1
(219291, 32196)       1
(219292, 18069)       1
(219292, 98853)       1
(219292, 106117)      1
(219292, 90006)       1
(219292, 133676)      1
(219292, 6922)        1
(219292, 28648)       1
(219292, 122588)      1
(219292, 121700)      1
(219292, 94110)       1
(219292, 83814)       1
(219292, 111482)      1
(219292, 84729)       1
(219293, 18069)       1
(219293, 113876)      1
(219293, 76907)       1
```

In [24]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Y = le.fit_transform(Y)
```

In [25]:
```
print(Y)
```

```
[2 1 2 ... 0 0 2]
```

In [26]:
```
#spliting the dataset
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=
```

In [27]:
```
#Check shape of X_train and X_test or Y_train
print(X_train.shape, X_test.shape, Y_train.shape)
```

```
(146926, 135976) (72368, 135976) (146926,)
```

In [28]:
```
from sklearn.linear_model import LogisticRegression
lg = LogisticRegression()
lg.fit(X_train, Y_train)
```

```
/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/lin
ear_model/_logistic.py:458: ConvergenceWarning:

lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
(https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver opti
ons:
    https://scikit-learn.org/stable/modules/linear_model.html#logi
stic-regression (https://scikit-learn.org/stable/modules/linear_mo
del.html#logistic-regression)
```

Out[28]:
```
▾ LogisticRegression

LogisticRegression()
```

In [29]:
```
from sklearn.metrics import accuracy_score
#traing dataset accuracy score
train_test = lg.predict(X_train)
accuracy_score(train_test,Y_train)
```

Out[29]:
```
0.9091651579706791
```

In [30]:
```
#test dataset accuracy score
test_data = lg.predict(X_test)
accuracy_score(test_data, Y_test)
```

Out[30]:
```
0.8460092858722087
```

In [31]:
```
from sklearn.metrics import jaccard_score, accuracy_score, f1_score,
preds = lg.predict(X_test)
print(classification_report(Y_test, preds))
```

```
              precision    recall  f1-score   support

           0       0.89      0.93      0.91     35518
           1       0.86      0.83      0.84     18508
           2       0.74      0.70      0.72     18342

    accuracy                           0.85     72368
   macro avg       0.83      0.82      0.82     72368
weighted avg       0.84      0.85      0.84     72368
```

In [33]:
```
#using from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
DT = DecisionTreeClassifier()
```

In [34]:
```
#model fit
DT.fit(X_train,Y_train)
```

Out[34]:
```
▾ DecisionTreeClassifier

DecisionTreeClassifier()
```

In [35]:
```
#traing dataset accuracy score
train_test = DT.predict(X_train)
accuracy_score(train_test,Y_train)
```

Out[35]: 0.9992921606795258

In [36]:
```
#test dataset accuracy score
test_data = DT.predict(X_test)
accuracy_score(test_data, Y_test)
```

Out[36]: 0.787862038470042

In [37]:
```
#classification report
preds = DT.predict(X_test)
print(classification_report(Y_test, preds))
```

```
              precision    recall  f1-score   support

           0       0.89      0.87      0.88     35518
           1       0.75      0.73      0.74     18508
           2       0.64      0.68      0.66     18342

    accuracy                           0.79     72368
   macro avg       0.76      0.76      0.76     72368
weighted avg       0.79      0.79      0.79     72368
```

In [43]:
```
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
nb.fit(X_train, Y_train)
```

Out[43]:
```
▾ MultinomialNB

MultinomialNB()
```

In [44]:
```
preds = nb.predict(X_test)
print(classification_report(Y_test, preds))
```

```
              precision    recall  f1-score   support

           0       0.80      0.89      0.84     35518
           1       0.63      0.81      0.71     18508
           2       0.58      0.30      0.39     18342

    accuracy                           0.72     72368
   macro avg       0.67      0.66      0.65     72368
weighted avg       0.70      0.72      0.70     72368
```

In [45]:
```
#Hyperparameter tuning
from sklearn.model_selection import GridSearchCV, RepeatedStratifiec
```

In [46]:
```
# Hyperparameter tuning for Multinomial Naive Bayes model

param_grid = {"alpha": [0.1,0,1.0, 10, 100]}

grid_search = GridSearchCV(MultinomialNB(), param_grid, verbose=2)

grid_search.fit(X_train, Y_train)
```

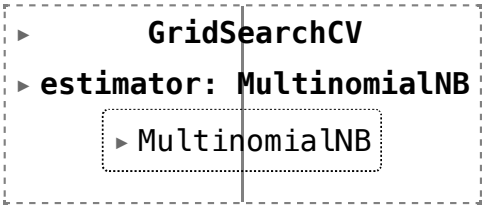
```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV] END .....................................alpha=0.1; tota
```

```
l time=   0.1s
[CV] END ..........................................alpha=0.1; tota
l time=   0.1s
[CV] END ..........................................alpha=0.1; tota
l time=   0.1s
[CV] END ..........................................alpha=0.1; tota
l time=   0.1s
[CV] END ..........................................alpha=0.1; tota
l time=   0.1s
[CV] END ............................................alpha=0; tota
l time=   0.1s
[CV] END ............................................alpha=0; tota
l time=   0.1s

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:627: FutureWarning:

The default value for `force_alpha` will change to `True` in 1.4.
To suppress this warning, manually set the value of `force_alpha`.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:633: UserWarning:

alpha too small will result in numeric errors, setting alpha = 1.0
e-10. Use `force_alpha=True` to keep alpha unchanged.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:627: FutureWarning:

The default value for `force_alpha` will change to `True` in 1.4.
To suppress this warning, manually set the value of `force_alpha`.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:633: UserWarning:

alpha too small will result in numeric errors, setting alpha = 1.0
e-10. Use `force_alpha=True` to keep alpha unchanged.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:627: FutureWarning:

The default value for `force_alpha` will change to `True` in 1.4.
To suppress this warning, manually set the value of `force_alpha`.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:633: UserWarning:

alpha too small will result in numeric errors, setting alpha = 1.0
e-10. Use `force_alpha=True` to keep alpha unchanged.


[CV] END ............................................alpha=0; tota
l time=   0.1s
[CV] END ............................................alpha=0; tota
```

```
l time=   0.1s

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:627: FutureWarning:

The default value for `force_alpha` will change to `True` in 1.4.
To suppress this warning, manually set the value of `force_alpha`.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:633: UserWarning:

alpha too small will result in numeric errors, setting alpha = 1.0
e-10. Use `force_alpha=True` to keep alpha unchanged.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:627: FutureWarning:

The default value for `force_alpha` will change to `True` in 1.4.
To suppress this warning, manually set the value of `force_alpha`.

/Users/vikky/opt/anaconda3/lib/python3.9/site-packages/sklearn/nai
ve_bayes.py:633: UserWarning:

alpha too small will result in numeric errors, setting alpha = 1.0
e-10. Use `force_alpha=True` to keep alpha unchanged.


[CV] END .....................................alpha=0; tota
l time=   0.2s
[CV] END .....................................alpha=1.0; tota
l time=   0.2s
[CV] END .....................................alpha=1.0; tota
l time=   0.1s
[CV] END .....................................alpha=1.0; tota
l time=   0.1s
[CV] END .....................................alpha=1.0; tota
l time=   0.3s
[CV] END .....................................alpha=1.0; tota
l time=   0.1s
[CV] END .....................................alpha=10; tota
l time=   0.1s
[CV] END .....................................alpha=10; tota
l time=   0.2s
[CV] END .....................................alpha=10; tota
l time=   0.2s
[CV] END .....................................alpha=10; tota
l time=   0.2s
[CV] END .....................................alpha=10; tota
l time=   0.2s
[CV] END .....................................alpha=100; tota
l time=   0.3s
[CV] END .....................................alpha=100; tota
l time=   0.2s
[CV] END .....................................alpha=100; tota
```

```
l time=   0.1s
[CV] END .....................................alpha=100; tota
l time=   0.1s
[CV] END .....................................alpha=100; tota
l time=   0.1s
```

Out[46]:



In [47]:
```
grid_search.best_params_
```

Out[47]:
```
{'alpha': 1.0}
```

In [ ]:
```

```