In [1]:

```
import pandas as pd

df = pd.read_csv("wine reviews.csv")
df.head()
```

Out[1]:

	ID	brand	manufacturerNumber	name	reviews_text	reviews_title	weight
0	W1	Gallo	12446.0	Ecco Domani174 Pinot Grigio - 750ml Bottle	This a fantastic white wine for any occasion!	My Favorite White Wine	2.0
1	W2	Fresh Craft Co.	17173.0	Fresh Craft174 Mango Citrus - 4pk / 250ml Bottle	Tart, not sweet...very refreshing and delicious!	Yum!!	1.5
2	W3	1000 Stories	18864.0	1000 Stories174 Zinfandel - 750ml Bottle	I was given this wine so it was a delightful s...	A New Favorite!	0.9
3	W4	1000 Stories	13420.0	1000 Stories174 Zinfandel - 750ml Bottle	This is a phenomenal wine and my new favorite ...	Bold, Flavorful, Aromatic, Delicious	0.2
4	W5	Wine Cube153	15247.0	Pink Moscato - 3l Bottle - Wine Cube153	4 750ml bottles for the price of two With way ...	Yum! Plus, Environmentally Friendly!	1.4

In [2]:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import re

lemma = WordNetLemmatizer()
stop_words = stopwords.words('english')

def text_prep(x):
     corp = str(x).lower()
     corp = re.sub('[^a-zA-Z]+',' ', corp).strip()
     tokens = word_tokenize(corp)
     words = [t for t in tokens if t not in stop_words]
     lemmatize = [lemma.lemmatize(w) for w in words]

     return lemmatize
```

In [3]:

```
preprocess_tag = [text_prep(i) for i in df['reviews_text']]
df["preprocess_txt"] = preprocess_tag

df['word_count'] = df['preprocess_txt'].map(lambda x: len(x))

f1 = open('negative-words.txt', 'r')
neg_words = f1.read().split()
f2 = open('positive-words1.txt', 'r')
pos_words = f2.read().split()

num_pos = df['preprocess_txt'].map(lambda x: len([i for i in x if i in pos_words]))
df['pos_score'] = num_pos
num_neg = df['preprocess_txt'].map(lambda x: len([i for i in x if i in neg_words]))
df['neg_score'] = num_neg

df['polarity_score'] = (df['pos_score'] - df['neg_score']) / ((df['pos_score'] + df['neg_sc
df['subjectivity'] = (df['pos_score'] + df['neg_score']) / (df['word_count'] + 0.000001)

df['sentiment'] = round((df['pos_score'] - df['neg_score']) / (df['word_count']+ 0.000001),

df.head()
```

Out[3]:

	ID	brand	manufacturerNumber	name	reviews_text	reviews_title	weight	prepro
0	W1	Gallo	12446.0	Ecco Domani174 Pinot Grigio - 750ml Bottle	This a fantastic white wine for any occasion!	My Favorite White Wine	2.0	[fantast wine, c
1	W2	Fresh Craft Co.	17173.0	Fresh Craft174 Mango Citrus - 4pk / 250ml Bottle	Tart, not sweet...very refreshing and delicious!	Yum!!	1.5	[ta re c
2	W3	1000 Stories	18864.0	1000 Stories174 Zinfandel - 750ml Bottle	I was given this wine so it was a delightful s...	A New Favorite!	0.9	[giv c surp
3	W4	1000 Stories	13420.0	1000 Stories174 Zinfandel - 750ml Bottle	This is a phenomenal wine and my new favorite ...	Bold, Flavorful, Aromatic, Delicious	0.2	[phe w favc
4	W5	Wine Cube153	15247.0	Pink Moscato - 3l Bottle - Wine Cube153	4 750ml bottles for the price of two With way ...	Yum! Plus, Environmentally Friendly!	1.4	[r price, t le, pa

In [4]:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer

analyzer = SentimentIntensityAnalyzer()
#df['compound'] = [analyzer.polarity_scores(x)['compound'] for x in df['reviews.text']]
```

In [5]:

```
df.dtypes
```

Out[5]:

```
ID                    object
brand                 object
manufacturerNumber    float64
name                  object
reviews_text          object
reviews_title         object
weight                float64
preprocess_txt        object
word_count             int64
pos_score              int64
neg_score              int64
polarity_score        float64
subjectivity          float64
sentiment             float64
dtype: object
```

In [6]:

```
reviews = df[["ID", "name", "reviews_text", "reviews_title"]].copy()
reviews.head()
```

Out[6]:

	ID	name	reviews_text	reviews_title
0	W1	Ecco Domani174 Pinot Grigio - 750ml Bottle	This a fantastic white wine for any occasion!	My Favorite White Wine
1	W2	Fresh Craft174 Mango Citrus - 4pk / 250ml Bottle	Tart, not sweet...very refreshing and delicious!	Yum!!
2	W3	1000 Stories174 Zinfandel - 750ml Bottle	I was given this wine so it was a delightful s...	A New Favorite!
3	W4	1000 Stories174 Zinfandel - 750ml Bottle	This is a phenomenal wine and my new favorite ...	Bold, Flavorful, Aromatic, Delicious
4	W5	Pink Moscato - 3l Bottle - Wine Cube153	4 750ml bottles for the price of two With way ...	Yum! Plus, Environmentally Friendly!

In [7]:

```
products = pd.read_csv("products.csv")
products.head()
```

Out[7]:

	name	brand
0	Ecco Domani174 Pinot Grigio - 750ml Bottle	Gallo
1	Fresh Craft174 Mango Citrus - 4pk / 250ml Bottle	Fresh Craft Co.
2	1000 Stories174 Zinfandel - 750ml Bottle	1000 Stories
3	Pink Moscato - 3l Bottle - Wine Cube153	Wine Cube153
4	Beck's174 Non Alcoholic Beer - 6pk / 12oz Bottles	Beck's Na

In [8]:

```
analysis = df[["ID", "word_count", "pos_score", "neg_score", "polarity_score", "subjectivit
analysis.head()
```

Out[8]:

	ID	word_count	pos_score	neg_score	polarity_score	subjectivity	sentiment
0	W1	4	1	0	0.999999	0.250000	0.25
1	W2	4	3	0	1.000000	0.750000	0.75
2	W3	10	3	0	1.000000	0.300000	0.30
3	W4	5	2	0	1.000000	0.400000	0.40
4	W5	29	6	3	0.333333	0.310345	0.10

In [9]:

```
import mysql.connector as mysql
from mysql.connector import Error
try:
    conn = mysql.connect(host='localhost', database='wine_reviews', user='root', password='
    if conn.is_connected():
        cursor = conn.cursor(buffered = True)
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)

        cursor.execute('DROP TABLE IF EXISTS analysis;')
        cursor.execute('DROP TABLE IF EXISTS reviews;')
        cursor.execute('DROP TABLE IF EXISTS products;')

        #TABLE 1

        print('Creating table REVIEWS')

        cursor.execute("CREATE TABLE reviews(ID VARCHAR(6) PRIMARY KEY, NAME VARCHAR(100),
        print("REVIEWS TABLE CREATED")

        for i,row in reviews.iterrows():

            sql = "INSERT INTO wine_reviews.reviews VALUES (%s,%s,%s,%s)"
            cursor.execute(sql, tuple(row))
            conn.commit()

        #TABLE 2

        print('Creating table PRODUCTS')

        cursor.execute("CREATE TABLE products(NAME VARCHAR(100) PRIMARY KEY, BRAND VARCHAR(
        print('PRODUCTS TABLE CREATED')

        for i,row in products.iterrows():

            sql = "INSERT INTO wine_reviews.products VALUES (%s,%s)"
            cursor.execute(sql, tuple(row))
            conn.commit()

        #TABLE 3

        print('Creating table ANALYSIS')

        cursor.execute("CREATE TABLE analysis(ID VARCHAR(6), word_count INT(5), pos_score F
        print('ANALYSIS TABLE CREATED')

        for i,row in analysis.iterrows():

            sql = "INSERT INTO wine_reviews.analysis VALUES (%s,%s,%s,%s,%s,%s,%s)"
            cursor.execute(sql, tuple(row))
            conn.commit()

except Error as e:
            print("Error while connecting to MySQL", e)

print("Successfully created all tables and inserted all records")
```

```
You're connected to database:  ('wine_reviews',)
Creating table REVIEWS
REVIEWS TABLE CREATED
Creating table PRODUCTS
PRODUCTS TABLE CREATED
Creating table ANALYSIS
ANALYSIS TABLE CREATED
Successfully created all tables and inserted all records
```

In [13]:

```
print("Select Action:")
print("1. Display all reviews")
print("2. Display all products")
print("3. Display all analysis")
print("4. Insert record in reviews")
print("5. Insert record in products")
print("6. Display reviews grouped by Product Name")
print("7. Display Average Word Count")
print("8. Exit")


#c = input("Do you want to continue? (Y/N)")
#def menu():
ch = int(input("Enter choice:"))

if ch == 1:
    cursor.execute("SELECT * FROM REVIEWS;")
    print(cursor.fetchall())

elif ch == 2:
    cursor.execute("SELECT * FROM PRODUCTS;")
    print(cursor.fetchall())

elif ch == 3:
    cursor.execute("SELECT * FROM PRODUCTS;")
    print(cursor.fetchall())

elif ch == 4:
    id1 = input("Enter ID:")
    name = input("Enter name:")
    review_text = input("Enter review text:")
    review_title = input("Enter review title:")
    t1 = (id1, name, review_text, review_title)
    df0 = pd.DataFrame(t1)
    reviews.append(df0)
    q1 = "INSERT INTO REVIEWS VALUES(%s, %s, %s, %s);"
    print(cursor.execute(q1, tuple(t1)))
    if name not in products["name"]:
        brand = "na"
        t2 = (name, brand)
        df2 = pd.DataFrame(t2)
        products.append(df2)
        q2 = "INSERT INTO PRODUCTS VALUES(%s, %s);"
        cursor.execute(q2, tuple(t2))
        print("n p u")
        preprocess_text1 = text_prep(review_text)
        id1 = len(reviews["ID"])+1
        wordcount = len(preprocess_text1)
        pos_count1 = 0
        neg_count1 = 0
        for i in preprocess_text1:
            if i in pos_words:
                pos_count1 += 1
            if i in neg_words:
                neg_count1 += 1
            polarity = (pos_count1 - neg_count1)/((pos_count1 + neg_count1) + 0.00001)
            subj = (pos_count1 + neg_count1) / (wordcount + 0.000001)
            sentiment1 = round((pos_count1 + neg_count1) / (wordcount + 0.000001), 2)
        t3 = (id1, wordcount, pos_count1, neg_count1, polarity, subj, sentiment1)
```

```
        df3 =pd.DataFrame(t3)
        analysis.append(df3)
        q3 = "INSERT INTO ANALYSIS VALUES(%s, %s, %s, %s, %s, %s, %s);"
        cursor.execute(q3, tuple(t3))


elif ch == 5:
    name = input("Enter name:")
    name = input("Enter brand:")
    t4 = (id1, name)
    df4 = pd.DataFrame(t4)
    products.append(df4)
    q4 = "INSERT INTO PRODUCTS VALUES(%s, %s);"
    cursor.execute(q4, tuple(t4))

elif ch == 6:
    print(cursor.execute("SELECT PRODUCTS.NAME, REVIEW_TEXT, REVIEW_TITLE FROM REVIEWS, PRO
    myresult0 = cursor.fetchall()
    for x in myresult0:
        print(x)

elif ch == 7:
    cursor.execute("SELECT AVG(WORD_COUNT) FROM ANALYSIS;")
    myresult = cursor.fetchall()
    for x in myresult:
        print(x)

elif ch == 8:
    print("Thank you!")

else:
    print("Error")

    ##if c == "Y" or "y":
#return menu()

#menu()
```

```
Select Action:
1. Display all reviews
2. Display all products
3. Display all analysis
4. Insert record in reviews
5. Insert record in products
6. Display reviews grouped by Product Name
7. Display Average Word Count
8. Exit
Enter choice:6
None
('Ecco Domani174 Pinot Grigio - 750ml Bottle', 'This a fantastic white win
e for any occasion!', 'My Favorite White Wine')
('Charles Charles174 Red Blend - 750ml Bottle', 'This is a very smooth red
with Aromas of cocoa, coffee, tobacco sweet black cherry. Bold but soft. I
highly recommend this one.', 'Charles & Charles Red Blend')
("Basil Hayden's174 Bourbon - 750ml Bottle", 'Outstanding. Simply an outst
anding beverage. Spicy and light with a comforting finish. A staple in my
home bar rotation - I never let a bottle get near empty without backfillin
```