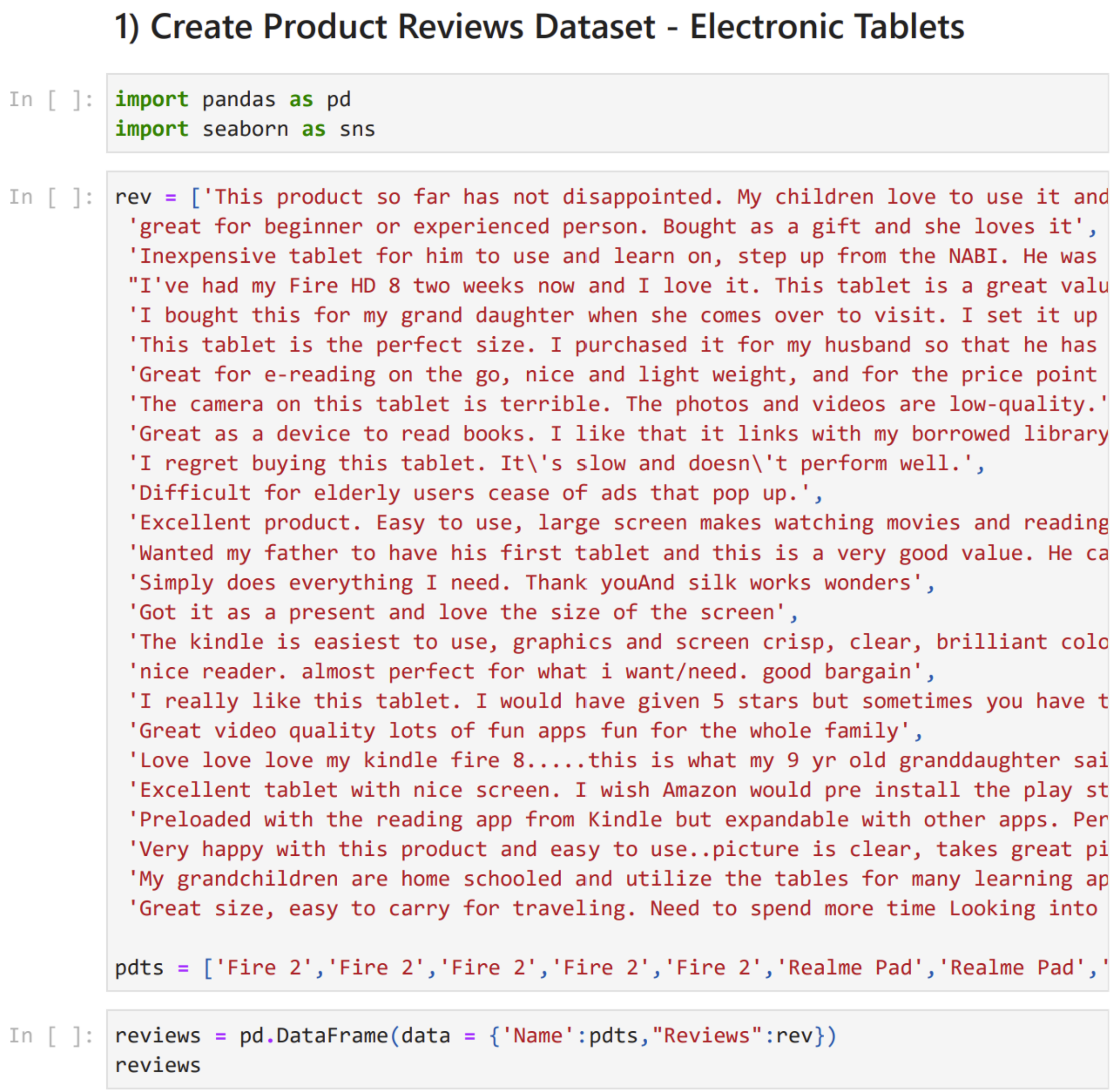
Machine Leaning Lab

Experiment – 2

Name: Ankith Motha

Reg No.: 21BAI1284

Create and Analyze Product Reviews



1) Create Product Reviews Dataset - Electronic Tablets

```
In [ ]: import pandas as pd
        import seaborn as sns
```

```
In [ ]: rev = ['This product so far has not disappointed. My children love to use it and
         'great for beginner or experienced person. Bought as a gift and she loves it',
         'Inexpensive tablet for him to use and learn on, step up from the NABI. He was
         "I've had my Fire HD 8 two weeks now and I love it. This tablet is a great valu
         'I bought this for my grand daughter when she comes over to visit. I set it up
         'This tablet is the perfect size. I purchased it for my husband so that he has
         'Great for e-reading on the go, nice and light weight, and for the price point
         'The camera on this tablet is terrible. The photos and videos are low-quality.'
         'Great as a device to read books. I like that it links with my borrowed library
         'I regret buying this tablet. It\'s slow and doesn\'t perform well.',
         'Difficult for elderly users cease of ads that pop up.',
         'Excellent product. Easy to use, large screen makes watching movies and reading
         'Wanted my father to have his first tablet and this is a very good value. He ca
         'Simply does everything I need. Thank youAnd silk works wonders',
         'Got it as a present and love the size of the screen',
         'The kindle is easiest to use, graphics and screen crisp, clear, brilliant colo
         'nice reader. almost perfect for what i want/need. good bargain',
         'I really like this tablet. I would have given 5 stars but sometimes you have t
         'Great video quality lots of fun apps fun for the whole family',
         'Love love love my kindle fire 8.....this is what my 9 yr old granddaughter sai
         'Excellent tablet with nice screen. I wish Amazon would pre install the play st
         'Preloaded with the reading app from Kindle but expandable with other apps. Per
         'Very happy with this product and easy to use..picture is clear, takes great pi
         'My grandchildren are home schooled and utilize the tables for many learning ap
         'Great size, easy to carry for traveling. Need to spend more time Looking into

        pdts = ['Fire 2','Fire 2','Fire 2','Fire 2','Fire 2','Realme Pad','Realme Pad','
```

```
In [ ]: reviews = pd.DataFrame(data = {'Name':pdts,"Reviews":rev})
        reviews
```

Out[ ]:

	Name	Reviews
0	Fire 2	This product so far has not disappointed. My c...
1	Fire 2	great for beginner or experienced person. Boug...
2	Fire 2	Inexpensive tablet for him to use and learn on...
3	Fire 2	I've had my Fire HD 8 two weeks now and I love...
4	Fire 2	I bought this for my grand daughter when she c...
5	Realme Pad	This tablet is the perfect size. I purchased i...
6	Realme Pad	Great for e-reading on the go, nice and light ...
7	Realme Pad	The camera on this tablet is terrible. The pho...
8	Realme Pad	Great as a device to read books. I like that i...
9	Realme Pad	I regret buying this tablet. It's slow and doe...
10	Lenovo Tab	Difficult for elderly users cease of ads that ...
11	Lenovo Tab	Excellent product. Easy to use, large screen m...
12	Lenovo Tab	Wanted my father to have his first tablet and ...
13	Lenovo Tab	Simply does everything I need. Thank youAnd si...
14	Lenovo Tab	Got it as a present and love the size of the s...
15	Kindle	The kindle is easiest to use, graphics and scr...
16	Kindle	nice reader. almost perfect for what i want/ne...
17	Kindle	I really like this tablet. I would have given ...
18	Kindle	Great video quality lots of fun apps fun for t...
19	Kindle	Love love love my kindle fire 8.....this is wh...
20	Samsung	Excellent tablet with nice screen. I wish Amaz...
21	Samsung	Preloaded with the reading app from Kindle but...
22	Samsung	Very happy with this product and easy to use.....
23	Samsung	My grandchildren are home schooled and utilize...
24	Samsung	Great size, easy to carry for traveling. Need ...

In [ ]:

```
reviews.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Name     25 non-null     object
 1   Reviews  25 non-null     object
dtypes: object(2)
memory usage: 528.0+ bytes
```

In [ ]:

```
# from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
```

In [ ]:

```
# stopWords = set(stopwords.words('english'))

with open(r"./stopwords.txt") as f:
    stopWords = f.readlines()
    stopWords = [i[:-1] for i in stopWords]
    print(stopWords)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'you
rs', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'her
s', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themsel
ves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'i
s', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having',
'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'becaus
e', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against',
'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 't
o', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'fu
rther', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'a
ny', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor',
'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'wil
l', 'just', 'don', 'should', 'no']
```

In [ ]:

```
comments = reviews['Reviews']
comments = list(comments)
```

In [ ]:

```
# Split the sentences to individual words(tokenize). Then remove the stopwords a
filtered=[]
for i in comments:
    t = []
    for w in i.split():
        w="".join(ch for ch in w if ch.isalnum())
        if w.lower() not in stopWords:
            t.append(w)
    filtered.append(t)
```

In [ ]:

```
with open("./good_words.txt") as f:
    gw = f.readlines()
    gw = [i[:-1] for i in gw]

with open("./bad_words.txt") as f:
    bd = f.readlines()
    bd = [i[:-1] for i in bd]
    print(bd)
```

```
['2-faced', '2-faces', 'abnormal', 'abolish', 'abominable', 'abominably', 'abomin
ate', 'abomination', 'abort', 'aborted', 'aborts', 'abrade', 'abrasive', 'abrup
t', 'abruptly', 'abscond', 'absence', 'absent-minded', 'absentee', 'absurd', 'abs
urdity', 'absurdly', 'absurdness', 'abuse', 'abused', 'abuses', 'abusive', 'abysm
al', 'abysmally', 'abyss', 'accidental', 'accost', 'accursed', 'accusation', 'acc
usations', 'accuse', 'accuses', 'accusing', 'accusingly', 'acerbate', 'acerbic',
'acerbically', 'ache', 'ached', 'aches', 'achey', 'aching', 'acrid', 'acridly',
'acridness', 'acrimonious', 'acrimoniously', 'acrimony', 'adamant', 'adamantly',
'addict', 'addicted', 'addicting', 'addicts', 'admonish', 'admonisher', 'admonish
ingly', 'admonishment', 'admonition', 'adulterate', 'adulterated', 'adulteratio
n', 'adulterier', 'adversarial', 'adversary', 'adverse', 'adversity', 'afflict',
'affliction', 'afflictive', 'affront', 'afraid', 'aggravate', 'aggravating', 'agg
ravation', 'aggression', 'aggressive', 'aggressiveness', 'aggressor', 'aggrieve',
'aggrieved', 'aggrivation', 'aghast', 'agonies', 'agonize', 'agonizing', 'agonizi
ngly', 'agony', 'aground', 'ail', 'ailing', 'ailment', 'aimless', 'alarm', 'alarm
ed', 'alarming', 'alarmingly', 'alienate', 'alienated', 'alienation', 'allegatio
n', 'allegations', 'allege', 'allergic', 'allergies', 'allergy', 'aloof', 'alterc
ation', 'ambiguity', 'ambiguous', 'ambivalence', 'ambivalent', 'ambush', 'amiss',
'amputate', 'anarchism', 'anarchist', 'anarchistic', 'anarchy', 'anemic', 'ange
r', 'angrily', 'angriness', 'angry', 'anguish', 'animosity', 'annihilate', 'annih
ilation', 'annoy', 'annoyance', 'annoyances', 'annoyed', 'annoying', 'annoyingl
y', 'annoys', 'anomalous', 'anomaly', 'antagonism', 'antagonist', 'antagonistic',
'antagonize', 'anti-', 'anti-american', 'anti-israeli', 'anti-occupation', 'anti-
proliferation', 'anti-semites', 'anti-social', 'anti-us', 'anti-white', 'antipath
y', 'antiquated', 'antithetical', 'anxieties', 'anxiety', 'anxious', 'anxiously',
'anxiousness', 'apathetic', 'apathetically', 'apathy', 'apocalypse', 'apocalypti
c', 'apologist', 'apologists', 'appal', 'appall', 'appalled', 'appalling', 'appal
lingly', 'apprehension', 'apprehensions', 'apprehensive', 'apprehensively', 'arbi
trary', 'arcane', 'archaic', 'arduous', 'arduously', 'argumentative', 'arroganc
e', 'arrogant', 'arrogantly', 'ashamed', 'asinine', 'asininely', 'asinininity',
'askance', 'asperse', 'aspersion', 'aspersions', 'assail', 'assassin', 'assassina
te', 'assault', 'assult', 'astray', 'asunder', 'atrocious', 'atrocities', 'atroci
ty', 'atrophy', 'attack', 'attacks', 'audacious', 'audaciously', 'audaciousness',
'audacity', 'audiciously', 'austere', 'authoritarian', 'autocrat', 'autocratic',
'avalanche', 'avarice', 'avaricious', 'avariciously', 'avenge', 'averse', 'aversi
on', 'aweful', 'awful', 'awfully', 'awfulness', 'awkward', 'awkwardness', 'ax',
'babble', 'back-logged', 'back-wood', 'back-woods', 'backache', 'backaches', 'bac
kaching', 'backbite', 'backbiting', 'backward', 'backwardness', 'backwood', 'back
woods', 'bad', 'badly', 'baffle', 'baffled', 'bafflement', 'baffling', 'bait', 'b
```

In [ ]:

```
# Check the number of positive and negative words in the sentences
pos_neg = []
for i in filtered:
    t = [0,0]
    for j in i:
        if j.lower() in gw:
            t[0]+=1
    for k in i:
        if k.lower() in bd:
            t[1]-=1
    pos_neg.append(t)

senti_score = pd.Series([i[0]+i[1] for i in pos_neg ])

sentiment = []
for i in senti_score:
    if i>0:
        sentiment.append("positive")
    elif i<0:
        sentiment.append("negative")
    else:
        sentiment.append("Neutral")
```

2) Reviews with Positive, Negative and Neutral Sentiments

In [ ]:

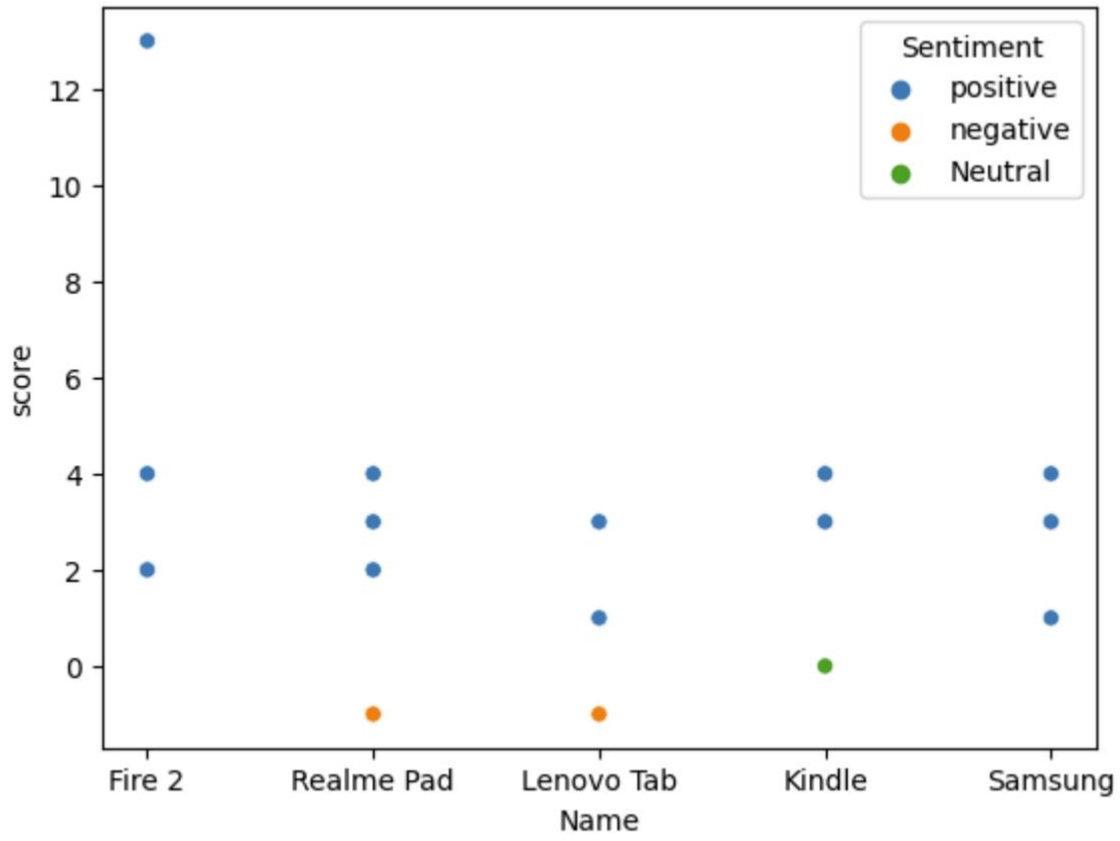
```
df = pd.DataFrame({'Name':pdts,"reviews":rev,"Sentiment":sentiment, "score":sent
df
```

Out[ ]:

	Name	reviews	Sentiment	score
0	Fire 2	This product so far has not disappointed. My c...	positive	2
1	Fire 2	great for beginner or experienced person. Boug...	positive	2
2	Fire 2	Inexpensive tablet for him to use and learn on...	positive	2
3	Fire 2	I've had my Fire HD 8 two weeks now and I love...	positive	13
4	Fire 2	I bought this for my grand daughter when she c...	positive	4
5	Realme Pad	This tablet is the perfect size. I purchased i...	positive	2
6	Realme Pad	Great for e-reading on the go, nice and light ...	positive	3
7	Realme Pad	The camera on this tablet is terrible. The pho...	negative	-1
8	Realme Pad	Great as a device to read books. I like that i...	positive	4
9	Realme Pad	I regret buying this tablet. It's slow and doe...	negative	-1
10	Lenovo Tab	Difficult for elderly users cease of ads that ...	negative	-1
11	Lenovo Tab	Excellent product. Easy to use, large screen m...	positive	3
12	Lenovo Tab	Wanted my father to have his first tablet and ...	positive	3
13	Lenovo Tab	Simply does everything I need. Thank youAnd si...	positive	3
14	Lenovo Tab	Got it as a present and love the size of the s...	positive	1
15	Kindle	The kindle is easiest to use, graphics and scr...	positive	4
16	Kindle	nice reader. almost perfect for what i want/ne...	positive	4
17	Kindle	I really like this tablet. I would have given ...	Neutral	0
18	Kindle	Great video quality lots of fun apps fun for t...	positive	3
19	Kindle	Love love love my kindle fire 8.....this is wh...	positive	3
20	Samsung	Excellent tablet with nice screen. I wish Amaz...	positive	3
21	Samsung	Preloaded with the reading app from Kindle but...	positive	1
22	Samsung	Very happy with this product and easy to use.....	positive	4
23	Samsung	My grandchildren are home schooled and utilize...	positive	3
24	Samsung	Great size, easy to carry for traveling. Need ...	positive	3

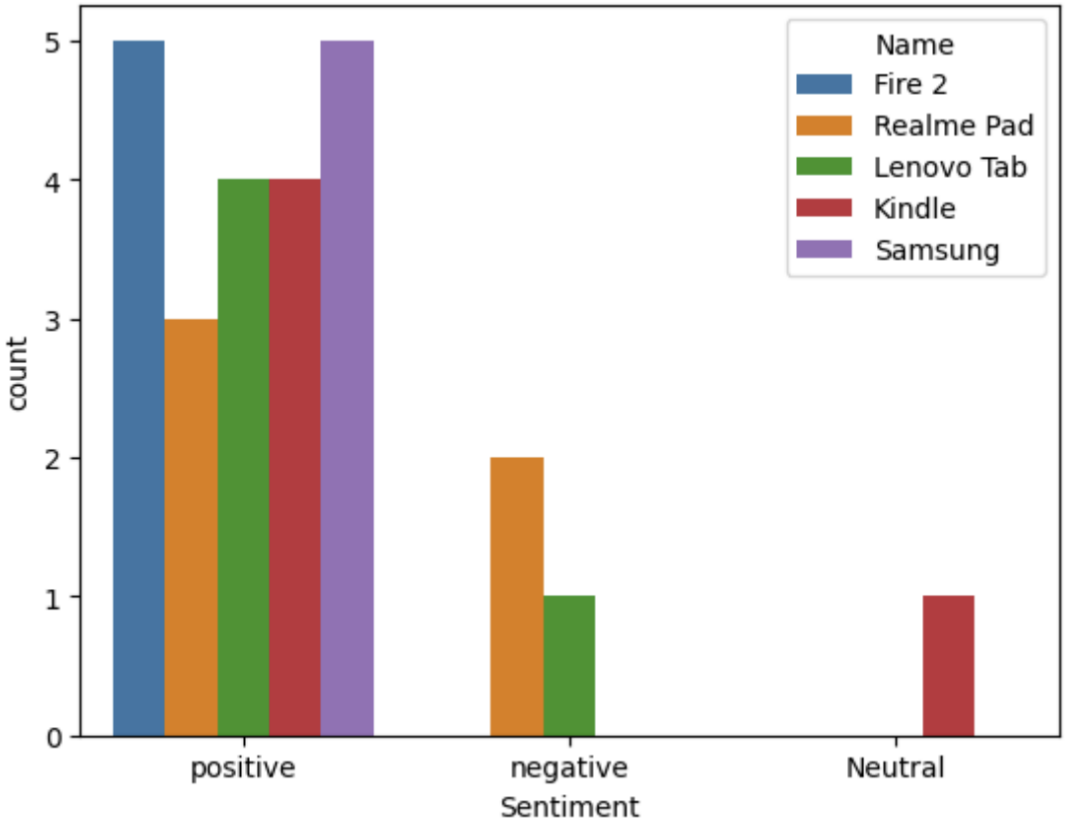
```
In [ ]: sns.scatterplot(x =df["Name"],y = df["score"],hue=df["Sentiment"])
```

```
Out[ ]: <Axes: xlabel='Name', ylabel='score'>
```



```
In [ ]: sns.countplot(x = df["Sentiment"], data=df, hue=df["Name"])
```

```
Out[ ]: <Axes: xlabel='Sentiment', ylabel='count'>
```



In [ ]:
```
rank = []
for i in range(5):
    rank.append(sum(senti_score[i*5:(i+1)*5] > 0))
rank
```

Out[ ]:
```
[5, 3, 4, 4, 5]
```

In [ ]:
```
Rank_df = pd.DataFrame({"Name":df["Name"].unique(), "Score":rank})
Rank_df.sort_values(by=["Score"],axis=0,ascending=False)
```

Out[ ]:

	Name	Score
0	Fire 2	5
4	Samsung	5
2	Lenovo Tab	4
3	Kindle	4
1	Realme Pad	3

4) Recommended Products - Based on if the review score for the product is positive or not

In [ ]:
```
Recommended = Rank_df[Rank_df["Score"]>0].sort_values(by=["Score"],axis=0,ascend
Recommended["Name"]
```

Out[ ]:
```
0         Fire 2
4        Samsung
2     Lenovo Tab
3         Kindle
1     Realme Pad
Name: Name, dtype: object
```

In [ ]:
```
# Input: Kindle
customer_requested_pdt = input("Enter the product you require")
ls = list(Recommended["Name"].values)
if customer_requested_pdt in ls:
    ind = ls.index(customer_requested_pdt)
    print(f"Yes, we recommend this product. It is our {ind+1}th best product")
else:
    print("No, we don't recommend this product")
```

```
Yes, we recommend this product. It is our 4th best product
```