*This is the **Amazon Fine Food Reviews** Dataset which including all ~500,000 reviews, Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.*

My task is to analyze this dataset and provide insights to see what is the level of customer satisfaction and their evaluations of the products.

Dataset Link : *https://2u.pw/O3oOa2O*

The visualization should answer these questions :

1- What do the Amazon Product Ratings look like across different levels?

2- What do the Amazon Product Ratings look like across different sentiment analysis?

3- What are the most 50 helpful reviews for other customers?

4- What are The Top 50 products with the most positive sentiment?

5- What are The worst 50 products ?

6- What are the most 50 positively rated reviews?

7- What are the most 50 negatively rated reviews?

Importing the libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

import plotly.io as pio
pio.templates.default = "plotly_white"

from nltk.sentiment.vader import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()

from tqdm import tqdm

import warnings
warnings.filterwarnings("ignore")
```

Data Importing & inspecting

In [2]:

```
df = pd.read_csv("F:\My Project\Classify Amazon Reviews!!\Reviews.csv")
```

In [3]:

```
data = df.copy()
data.head()
```

Out[3]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

Let's explore our dataset to examine its columns, data types, and column names.

In [4]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
 #   Column                  Non-Null Count   Dtype 
---  ------                  --------------   ----- 
 0   Id                      568454 non-null  int64 
 1   ProductId               568454 non-null  object
 2   UserId                  568454 non-null  object
 3   ProfileName             568438 non-null  object
 4   HelpfulnessNumerator    568454 non-null  int64 
 5   HelpfulnessDenominator  568454 non-null  int64 
 6   Score                   568454 non-null  int64 
 7   Time                    568454 non-null  int64 
 8   Summary                 568427 non-null  object
 9   Text                    568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

Let's take a look at the Summary of our Datasets:

It help us provide an overview of the distribution and the range in each column in our dataset.

In [5]:

```
data.describe().round(0)
```

Out[5]:

	Id	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time
count	568454.0	568454.0	568454.0	568454.0	5.684540e+05
mean	284228.0	2.0	2.0	4.0	1.296257e+09
std	164099.0	8.0	8.0	1.0	4.804331e+07
min	1.0	0.0	0.0	1.0	9.393408e+08
25%	142114.0	0.0	0.0	4.0	1.271290e+09
50%	284228.0	0.0	1.0	5.0	1.311120e+09
75%	426341.0	2.0	2.0	5.0	1.332720e+09
max	568454.0	866.0	923.0	5.0	1.351210e+09

Data Preparation & Exploration

First : Lets sum all the null values in our dataset

In [6]:

```
data.isna().sum()
```

Out[6]:

```
Id                         0
ProductId                  0
UserId                     0
ProfileName               16
HelpfulnessNumerator       0
HelpfulnessDenominator     0
Score                      0
Time                       0
Summary                   27
Text                       0
dtype: int64
```

Second: Lets drop all the null values in our dataset because we have ~500,000 record and 46 record won't affect on it.

In [7]:

```
data.dropna(inplace=True)
data.shape
# data = data.head(5000)
# data.shape
```

Out[7]:

```
(568411, 10)
```

Lets take a look at the rating scores to understand how customers rate the products.

Here, you will find the star rating ranging from 1 to 5. Understanding the relationship between the star rating and the reviews is important for our analysis journey.

In [8]:

```
ratings = data["Score"].value_counts().sort_index()
ratings
```

Out[8]:

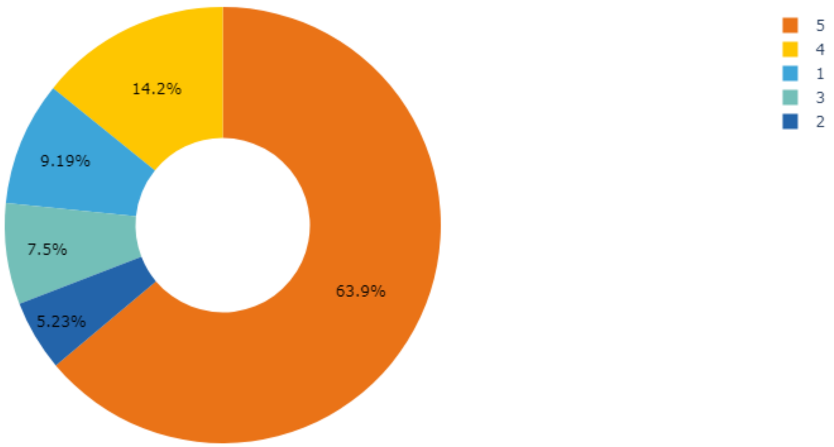
```
1     52264
2     29743
3     42638
4     80655
5    363111
Name: Score, dtype: int64
```

Lets visualize the customer ratings

In [9]:

```
# Define the color palette
color_palette = ['rgb(61, 165, 217)','rgb(35, 100, 170)','rgb(115, 191, 184)','rgb(254, 198, 1)','rgb(234, 115, 23)']

# Create the pie chart
fig = px.pie(ratings,
             values=ratings.values,
             names=ratings.index,
             hole=0.4,
             color=ratings.values,
             color_discrete_sequence=color_palette,
             title="The Amazon Product Ratings by Different Levels")

# Customize the the chart
fig.update_traces(textinfo = "percent",insidetextfont_color = "Black")
fig.update_layout(legend_itemclick= False)

# Showing the fig
fig.show()
```

The Amazon Product Ratings by Different Levels



So lets do some sentiment analysis to identify the negative, positive and neutral score of the review texts.

Notice here I used a for loop to iterate through the dataset and calculate the polarity scores using the polarity_score method.

This will allow us:

1- Analyzing the sentiment of each review or data point in the dataset.

2-Making informed decisions about the best and worst products based on customer sentiment.

In [10]:

```
#Run the polarity score on the dataset
result = {}
for i,row in tqdm(data.iterrows(),total = len(data)):
    text = row["Text"]
    my_id = row["Id"]
    result[my_id] = sia.polarity_scores(text)
```

```
100%|██████████| 568411/568411 [24:39<00:00, 384.09it/s]
```

Here I have converted the results from a dictionary to a dataframe to make it more manageable and easier to work with.

In [11]:

```
# Lets store the result into a pandas dataframe
vaders = pd.DataFrame(result).T
vaders.head()
```

Out[11]:

	neg	neu	pos	compound
1	0.000	0.695	0.305	0.9441
2	0.138	0.862	0.000	-0.5664
3	0.091	0.754	0.155	0.8265
4	0.000	1.000	0.000	0.0000
5	0.000	0.552	0.448	0.9468

Here I have organized the data into a table and renamed the index column as 'Id' for better alignment. This will enable us to seamlessly merge the two tables, 'vaders' and 'data,' and combine the relevant information for further analysis.

In [12]:

```python
#lets store the dataframe into tabel
vaders = vaders.reset_index()
# Lets rename the index column to Id column so i can merge the two tables ("vaders" and "data")
vaders = vaders.rename(columns = {"index" :"Id"})
vaders.head()
```

Out[12]:

	Id	neg	neu	pos	compound
0	1	0.000	0.695	0.305	0.9441
1	2	0.138	0.862	0.000	-0.5664
2	3	0.091	0.754	0.155	0.8265
3	4	0.000	1.000	0.000	0.0000
4	5	0.000	0.552	0.448	0.9468

Now, we can use the table to perform various analyses and gain valuable insights efficiently.

In [13]:

```python
# lets merge the the two tables ("vaders" and "data") into vaders_reviews table
vaders_reviews = data.merge(vaders , how= "left")

# Now we have sentiment score and meta data
vaders_reviews.head(2)
```

Out[13]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	neg	neu	pos	compound
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	0.000	0.695	0.305	0.9441
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	0.138	0.862	0.000	-0.5664

Let's explore the relationship between the Compound Score and the Star Rating.

By understanding this relationship, we can evaluate the accuracy and effectiveness of the sentiment analysis and its importance in reflecting customers' satisfaction levels.

In [14]:

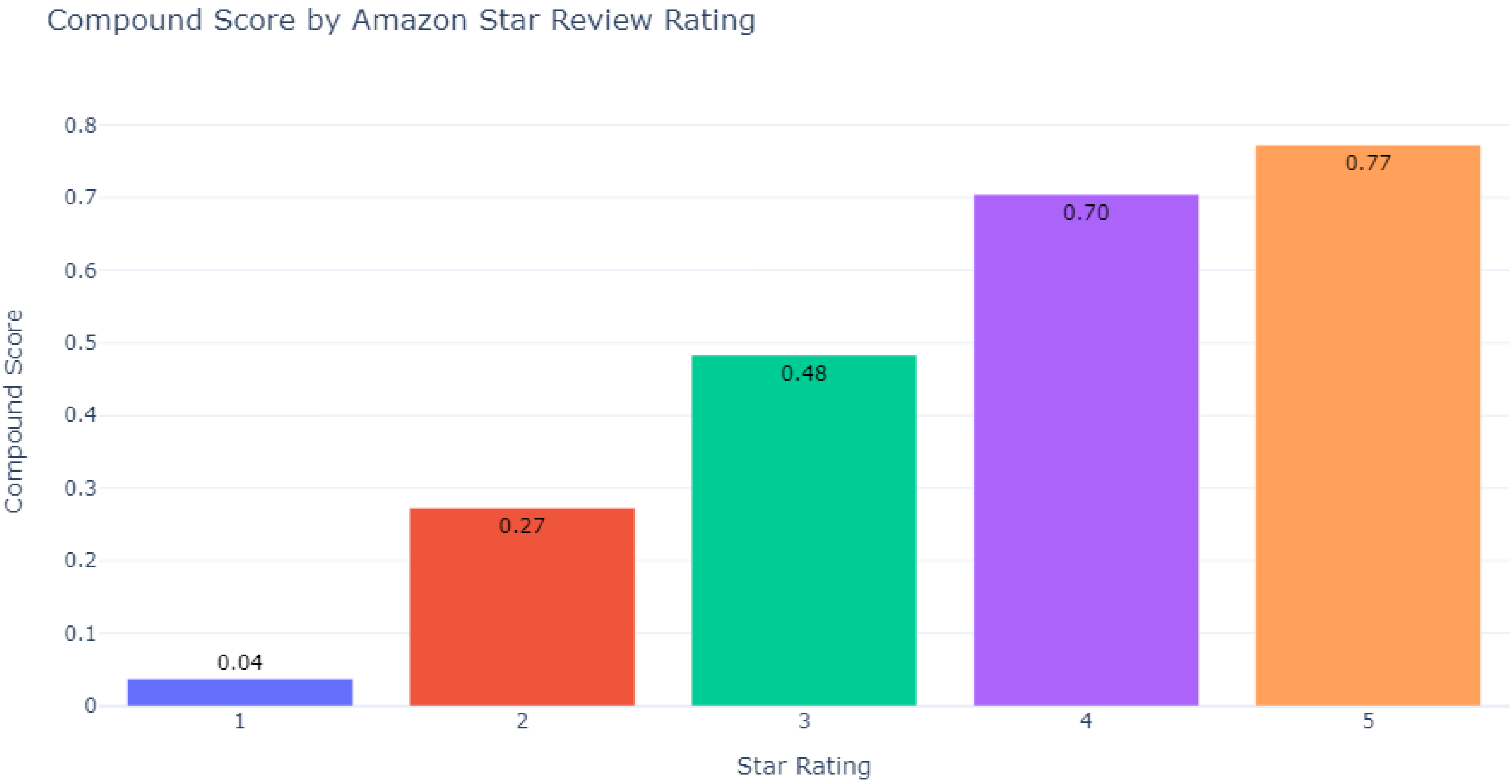
```
compoundScore_by_StarRating = vaders_reviews.groupby("Score")["compound"].mean().reset_index()
compoundScore_by_StarRating
```

Out[14]:

	Score	compound
0	1	0.037269
1	2	0.272738
2	3	0.483362
3	4	0.704605
4	5	0.772390

In [15]:

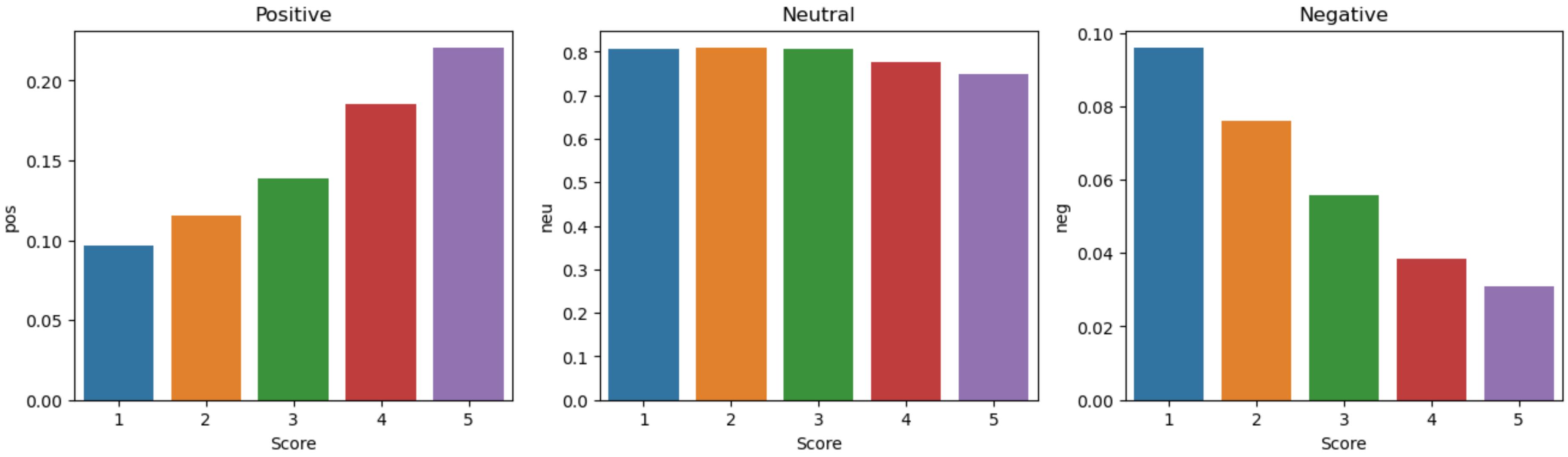
```
# Define the color palette
color_palette = ['rgb(61, 165, 217)','rgb(35, 100, 170)','rgb(115, 191, 184)','rgb(254, 198, 1)','rgb(234, 115, 23)']

# Creat the bar chart
fig = px.bar(compoundScore_by_StarRating,
              x = "Score",
              y = "compound",
              text="compound",
             color =color_palette,
             title = "Compound Score by Amazon Star Review Rating ")

# Customize the chart
fig.update_traces(texttemplate='%{text:.2f}',textfont_color = "black" )
fig.update_layout(showlegend=False)
fig.update_layout(xaxis_title = "Star Rating" )
fig.update_layout(yaxis_title = "Compound Score")


# Showing the chart
fig.show()
```


Compound Score by Amazon Star Review Rating
Compound Score
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.04
0.27
0.48
0.70
0.77
1
2
3
4
5
Star Rating

Let's examine the correlation between the Positive, Neutral, and Negative Scores and the Star Rating.

This analysis can provide valuable insights into how these sentiment scores align with the overall ratings.

In [16]:
```
# Create  a figure with 3 subplots
fig , axs = plt.subplots( 1 ,3 , figsize = ( 16 , 4 ))

# The first subplot :Positive Sentiment Scores
fig = sns.barplot(data = vaders_reviews, x = "Score", y= "pos", ax = axs[0] ,ci=None)
axs[0].set_title("Positive")

# The Second subplot :Neutral Sentiment Scores
fig = sns.barplot(data = vaders_reviews, x = "Score", y= "neu", ax = axs[1],ci=None)
axs[1].set_title("Neutral")

# The Third subplot :Negative Sentiment Scores
fig = sns.barplot(data = vaders_reviews, x = "Score", y= "neg", ax = axs[2],ci=None)
axs[2].set_title("Negative")

# Showing the plots
plt.show()
```



Let's identify and highlight the most helpful reviews based on the number of users who found them useful.

Analyzing most helpful reviews can help us:

1-Providing valuable insights to other customers.

2-Demonstrating our commitment to delivering exceptional experiences.

3-Influencing potential customers' decisions and building trust in our products or services.

In [17]:

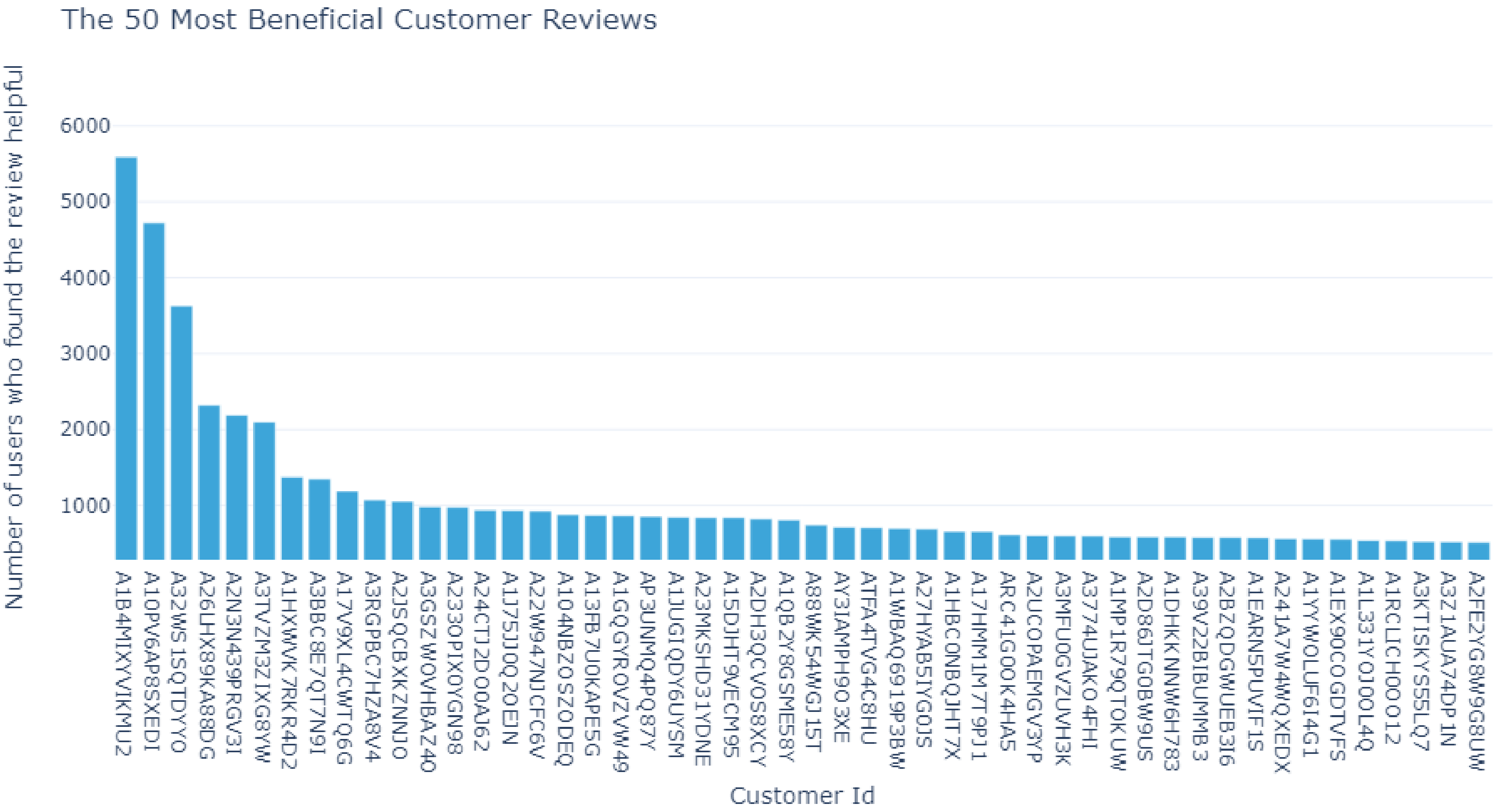
```
# Select the top 50 most helpful reviews
top_helpful_reviews = vaders_reviews.groupby(["UserId","Text"])["HelpfulnessNumerator"].sum(). \
    reset_index().sort_values("HelpfulnessNumerator",ascending = False).head(50)

top_helpful_reviews.head(5)
```

Out[17]:

	UserId	Text	HelpfulnessNumerator
31477	A1B4MIXYVIKMU2	This Ecobrew reusable Keurig K-cup is great fo...	5590
2733	A10PV6AP8SXEDI	Seriously, I love my Keurig. I love the conven...	4720
215857	A32WS1SQTDYYO	To cut to the chase, this produces a very good...	3632
121921	A26LHX89KA88DG	When first ordering a couple of ekobrew cups, ...	2320
170158	A2N3N439PRGV3I	I eat well. I read a lot of research on healt...	2190

In [18]:

```
# Define the color palette
color_palette = ['rgb(61, 165, 217)']

# Creat the bar chart
fig = px.bar(top_helpful_reviews,
              x= "UserId",
             y= "HelpfulnessNumerator",

             color="UserId",
             color_discrete_sequence=color_palette,
             title="The 50 Most Beneficial Customer Reviews ")

# Customize the chart

fig.update_layout(showlegend=False)
fig.update_layout(xaxis_title = "Customer Id")
fig.update_layout(yaxis_title = "Number of users who found the review helpful ")

# Showing the chart
fig.show()
```


The 50 Most Beneficial Customer Reviews
Number of users who found the review helpful
6000
5000
4000
3000
2000
1000
Customer Id
A1B4MIXYVIKMU2
A10PV6AP8SXEDI
A32WS1SQTDYYO
A26LHX89KA88DG
A2N3N439PRGV3I
A3TVZM3ZIXG8YW
A1HXWVK7RKR4D2
A3BBC8E7QT7N9I
A17V9XL4CWTQ6G
A3RGPBC7HZA8V4
A2JSQCBXKZNNJO
A3GSZWOVHBAZ4O
A2330PIX0YGN98
A24CTJ2DOOAJ62
A1J75JJOQ2OEIN
A22W947NJCFC6V
A104NBZOSZODEQ
A13FB7UOKAPE5G
A1GQGYROVZVW49
AP3UNMQ4PQ87Y
A1JUGIQDY6UYSM
A23MKSHD31YDNE
A15DJHT9VECM95
A2DH3QCVOS8XCY
A1QB2Y8GSME58Y
A88WK54WGJ15T
AY3IAMPH9O3XE
ATFA4TVG4C8HU
A1WBAQ6919P3BW
A27HYAB5IYGOJS
A1HBC0NBQJHT7X
A17HMM1M7T9PJ1
ARC41GOOK4HA5
A2UCOPAEMGV3YP
A3MFUOGVZUVH3K
A3774UJAKO4FHI
A1MP1R79QTOKUW
A2D86JTGOBW9US
A1DHKKNNW6H783
A39V22BIBUMMB3
A2BZQDGWUEB3I6
A1EARN5PUVIF1S
A241A7W4WQXEDX
A1YYWOLUF6I4G1
A1EX9OCOGDTVFS
A1L331YOJOOL4Q
A1RCLICHOOO12
A3KTISKYS55LQ7
A3Z1AUA74DP1N
A2FE2YG8W9G8UW

Let's highlight the Top 50 products based on the most positive sentiment.

Analyzing best products can help us:

1-Showcasing the strengths of these products and understanding what customers love about them.

2-Attracting potential customers to our offerings by using positve sentiments as powerful endorsements

In [19]:

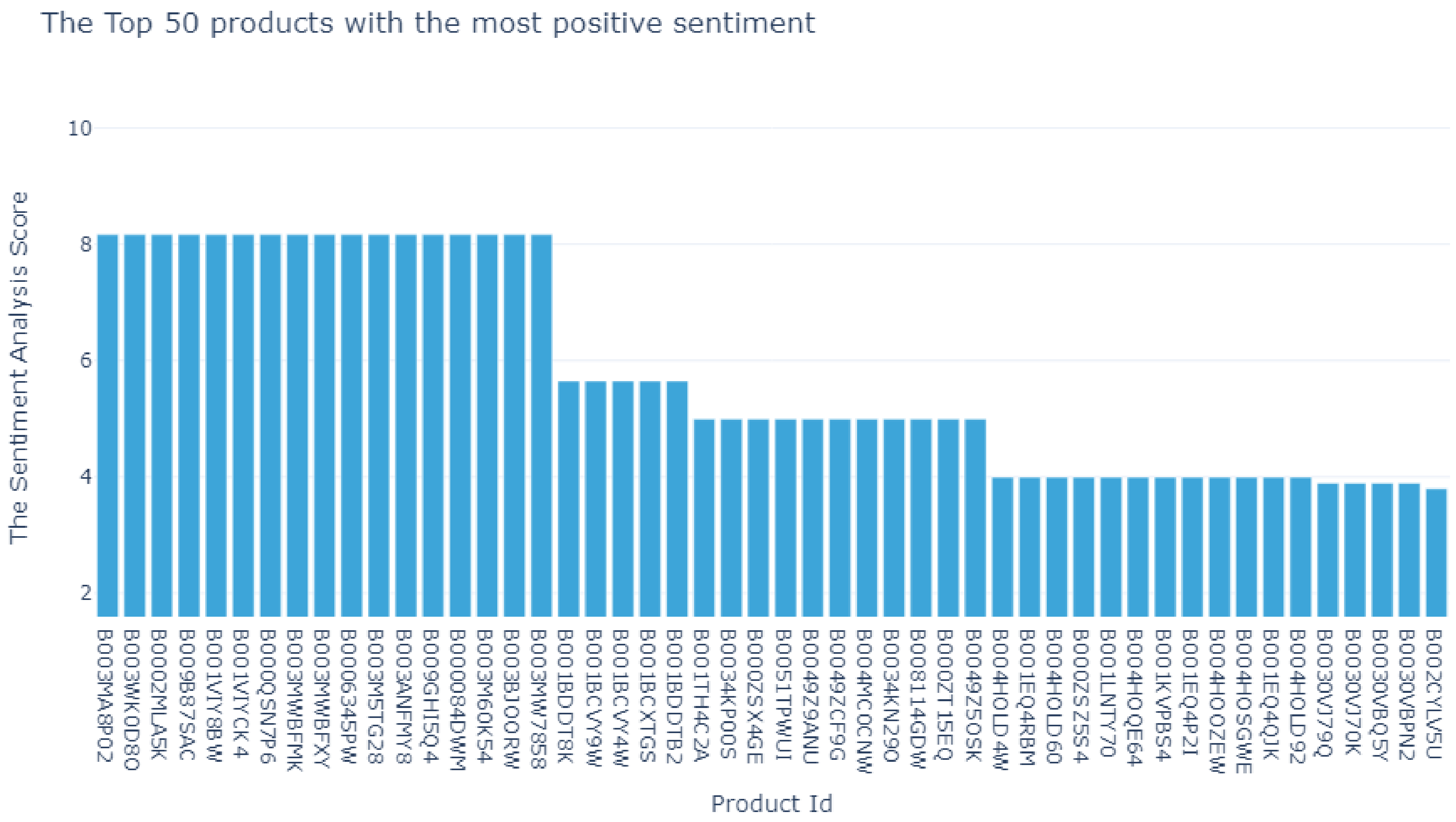
```
# Select the top 50 products with the most positive sentiment
top_best_products = vaders_reviews.groupby(["ProductId","Text"])["compound"].sum(). \
    reset_index().sort_values("compound",ascending = False).head(50)

top_best_products.head(5)
```

Out[19]:

	ProductId	Text	compound
404932	B003MA8P02	This review will make me sound really stupid, ...	8.168
421023	B003WK0D8O	This review will make me sound really stupid, ...	8.168
22436	B0002MLA5K	This review will make me sound really stupid, ...	8.168
565957	B009B87SAC	This review will make me sound really stupid, ...	8.168
311966	B001VIY8BW	This review will make me sound really stupid, ...	8.168

In [20]:

```
# Define the color palette
color_palette = ['rgb(61, 165, 217)']

# Creat the bar chart
fig = px.bar(top_best_products,
              x= "ProductId",
             y= "compound",
             color="ProductId",
             color_discrete_sequence=color_palette,
             title="The Top 50 products with the most positive sentiment")

# Customize the chart
fig.update_layout(showlegend=False)
fig.update_layout(xaxis_title = "Product Id")
fig.update_layout(yaxis_title = "The Sentiment Analysis Score")

# Showing the chart
fig.show()
```


The Top 50 products with the most positive sentiment
The Sentiment Analysis Score
10
8
6
4
2
B003MA8P02
B003WKOD8O
B0002MLA5K
B009B87SAC
B001VIY8BW
B001VIYCK4
B000QSN7P6
B003MWBFMK
B003MWBFXY
B0006345PW
B003M5TG28
B003ANFMY8
B009GHI5Q4
B00084DWM
B003M60K54
B003BJOORW
B003MW7858
B001BDDT8K
B001BCVY9W
B001BCVY4W
B001BCXTGS
B001BDDTB2
B001TH4C2A
B0034KP00S
B000ZSX4GE
B0051TPWUI
B00492Z9ANU
B00049ZCF9G
B004MCOCNW
B0034KN29O
B008114GDW
B000ZT15EQ
B00492Z5OSK
B004HOLD4W
B001EQ4RBM
B004HOLD60
B000ZSZ5S4
B001LNTY70
B004HOQE64
B001KVPBS4
B001EQ4P2I
B004HOOZEW
B004HOSGWE
B001EQ4QJK
B004HOLD92
B0030VJ79Q
B0030VJ70K
B0030VBQ5Y
B0030VBPN2
B002CYLV5U
Product Id

Let's highlight the 50 worst products based on the most negative sentiment.

Analyzing worst products can help us:

1-Improving product quality and customers satisfaction overall.

2-Identifying specific issues customers are facing and address them.

In [21]:

```python
# Select the top 50 most helpful reviews
top_best_products = vaders_reviews.groupby(["ProductId","Text"])["compound"].sum(). \
    reset_index().sort_values("compound",ascending = True).head(50)

top_best_products.head(5)
```

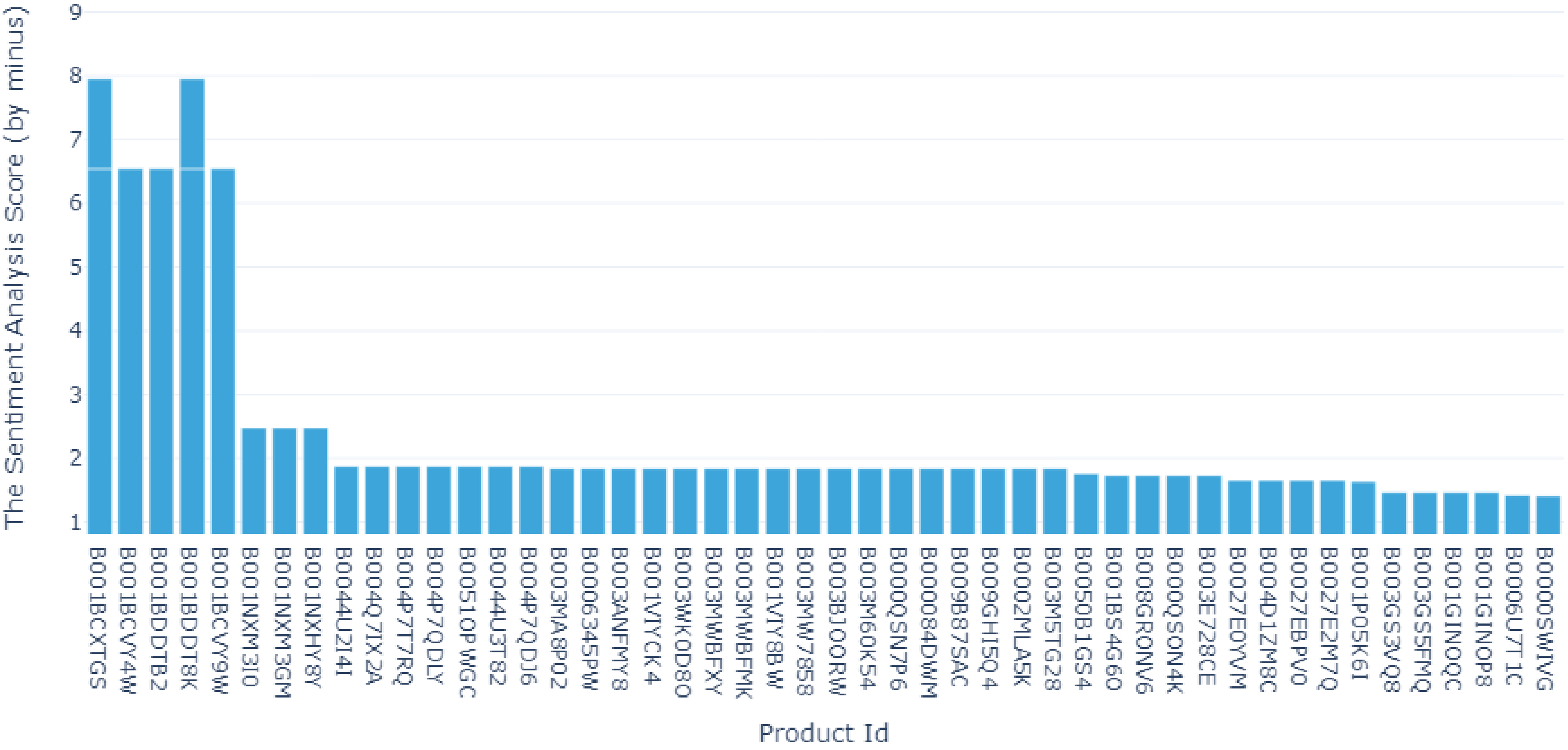
Out[21]:

	ProductId	Text	compound
223889	B001BCXTGS	According to the manufacturer's website, this ...	-6.5408
223390	B001BCVY4W	According to the manufacturer's website, this ...	-6.5408
224409	B001BDDTB2	According to the manufacturer's website, this ...	-6.5408
224180	B001BDDT8K	According to the manufacturer's website, this ...	-6.5408
223619	B001BCVY9W	According to the manufacturer's website, this ...	-6.5408

In [22]:

```python
# Define the color palette
color_palette = ['rgb(61, 165, 217)']

# Creat the bar chart
fig = px.bar(top_best_products,
              x= "ProductId",
             y= [abs(x) for x in top_best_products["compound"]],# Notic i used here abs function and for loop so,I can make the y_axis positive.
             color="ProductId",
             color_discrete_sequence=color_palette,
             title="The worst 50 products")

# Customize the chart
fig.update_layout(showlegend=False)
fig.update_layout(xaxis_title = "Product Id")
fig.update_layout(yaxis_title = "The Sentiment Analysis Score (by minus)")

# Showing the chart
fig.show()
```

The worst 50 products



Let's select the Top 50 Most Positive Reviews based on their Positive Score.

Analyzing positive reviews can help us:

1-Identifying and highlight the positive experiences customers have had with the products or services.

2-Understanding what customers appreciate about your products.

3-Helping build trust and confidence among potential customers.

In [23]:

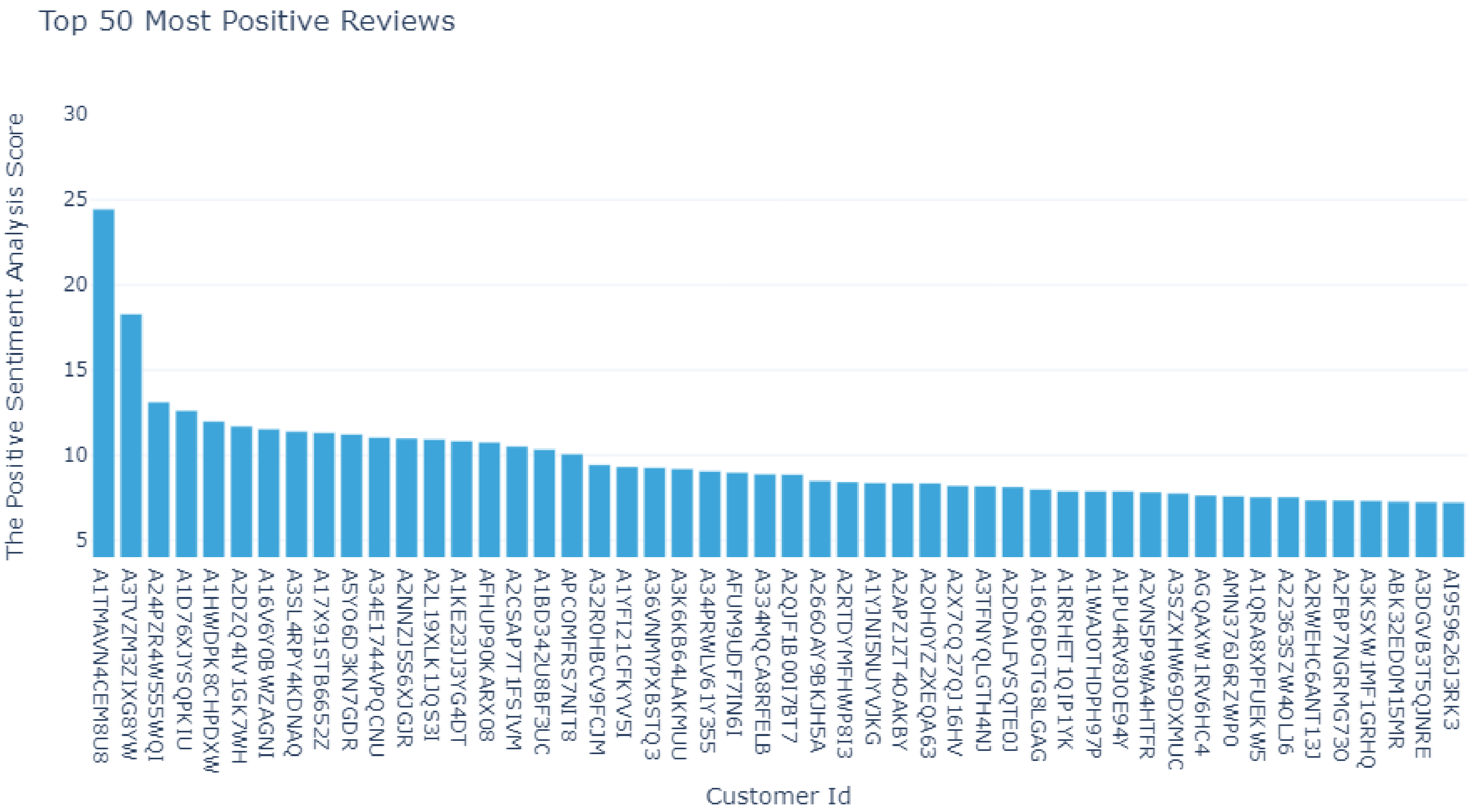
```
# Select the Top 50 Most Positive Reviews
top_positive_reviews = vaders_reviews.groupby(["UserId","Text"])["pos"].sum(). \
    reset_index().sort_values("pos",ascending = False).head(50)

top_positive_reviews.head(5)
```

Out[23]:

	UserId	Text	pos
83751	A1TMAVN4CEM8U8	Diamond Almonds<br />Almonds are a good source...	24.444
294372	A3TVZM3ZIXG8YW	This review will make me sound really stupid, ...	18.308
116239	A24PZR4W555WQI	My dogs and I love this food. They never leave...	13.132
37291	A1D76XJYSQPKIU	Received in good condition and in a timely man...	12.625
50569	A1HWDPK8CHPDXW	I love this tea. It is delicious. It came in a...	12.000

In [24]:

```
# Define the color palette
color_palette = ['rgb(61, 165, 217)']

# Creat the bar chart
fig = px.bar(top_positive_reviews,
              x= "UserId",
             y= "pos",
             color="UserId",
             color_discrete_sequence=color_palette,
             title="Top 50 Most Positive Reviews")

# Customize the chart
fig.update_yaxes(range= [0,26])
fig.update_layout(showlegend=False)
fig.update_layout(xaxis_title = "Customer Id")
fig.update_layout(yaxis_title = "The Positive Sentiment Analysis Score")

# Showing the chart
fig.show()
```


Top 50 Most Positive Reviews
The Positive Sentiment Analysis Score
30
25
20
15
10
5
A1TMAVN4CEM8U8
A3TVZM3ZIXG8YW
A24PZR4W555WQI
A1D76XJYSQPKIU
A1HWDPK8CHPDXW
A2DZQ4IV1GK7WH
A16V6Y0BWZAGNI
A3SL4RPY4KDNAQ
A17X91STB6652Z
A5YO6D3KN7GDR
A34E1744VPQCNU
A2NNZJ5S6XJGJR
A2L19XLK1JQS3I
A1KE23JJ3YG4DT
AFHUP90KARX08
A2CSAP7T1FSIVM
A1BD342U8BF3UC
APCOMFRS7NIT8
A32R0HBCV9FCJM
A1YFI21CFKYV5I
A36VNMYPXBSTQ3
A3K6KB64LAKMUU
A34PRWLV61Y355
AFUM9UDF7IN6I
A334MQCA8RFELB
A2QJF1B00I7BT7
A266OAY9BKJH5A
A2RTDYMFHWP8I3
A1YJNI5NUYVJKG
A2APZJZT4OAKBY
A2OHOYZ2XEQA63
A2X7CQ27QJ16HV
A3TFNYQLGTH4NJ
A2DDALFVSQTEOJ
A16Q6DGTG8LGAG
A1RRHET1QIP1YK
A1WAJOTHDPH97P
A1PU4RV8JOE94Y
A2VN5P9WA4HTFR
A3SZXHW69DXMUC
AGQAXW1RV6HC4
AMN376J6RZWP0
A1QRA8XPFUEKW5
A22363SZW4OLJ6
A2RWEHC6ANT13J
A2FBP7NGRMG730
A3KSXW1MF1GRHQ
ABK32EDOM15MR
A3DGVB3T5QJNRE
AI959626J3RK3
Customer Id

Let's select the Top 50 Most Negative Reviews based on their Negative Score.

Analyzing negative reviews can help us:

1- Identifying areas for improvement and address any issues customers might have encountered with the products or services.

2- Showing a proactive approach in understanding customer concerns and striving for better customer satisfaction.

In [25]:

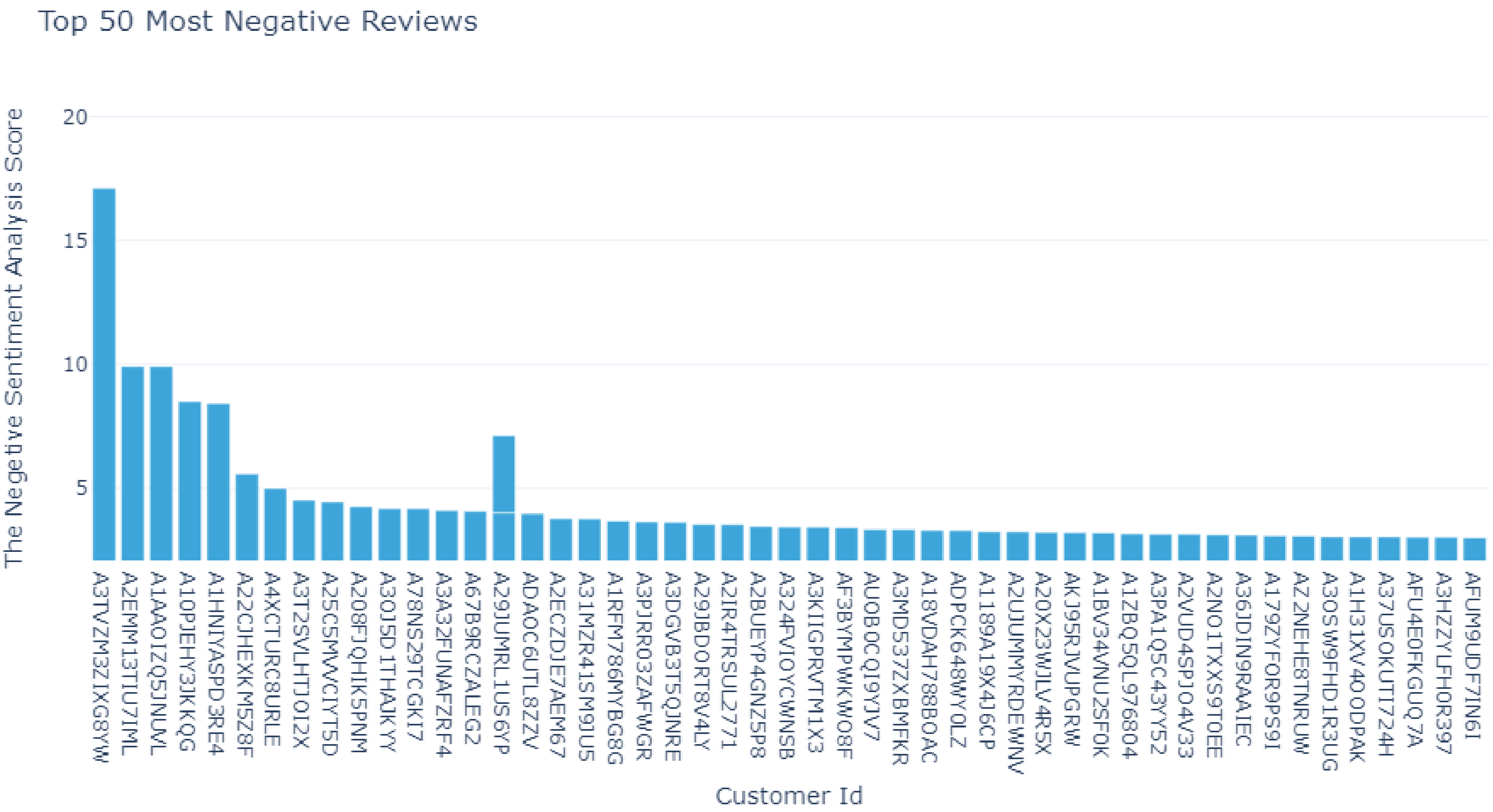
```python
# Select the Top 50 Most Negative Reviews
top_negative_reviews = vaders_reviews.groupby(["UserId","Text"])["neg"].sum(). \
    reset_index().sort_values("neg",ascending = False).head(50)

top_negative_reviews.head(5)
```

Out[25]:

	UserId	Text	neg
294372	A3TVZM3ZIXG8YW	This review will make me sound really stupid, ...	17.114
145754	A2EMM13TIU7IML	This was a waste of money, the item was late, ...	9.925
29277	A1AAOIZQ5JNUVL	This tea tastes nasty. Maybe I just dont like...	9.925
2716	A10PJEHY3JKKQG	This stuff taste the worse, no sweetner helps ...	8.500
49823	A1HNIYASPD3RE4	I purchased this tea because I was told that i...	8.425

In [26]:

```python
# Define the color palette
color_palette = ['rgb(61, 165, 217)']

# Creat the bar chart
fig = px.bar(top_negative_reviews,
              x= "UserId",
             y= "neg",
             color="UserId",
             color_discrete_sequence=color_palette,
             title="Top 50 Most Negative Reviews")

# Customize the chart
fig.update_layout(showlegend=False)
fig.update_layout(xaxis_title = "Customer Id")
fig.update_layout(yaxis_title = "The Negetive Sentiment Analysis Score")

# Showing the chart
fig.show()
```



Authority to:

Ammar Allam