In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('ggplot')

import nltk
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A NumP
y version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected ver
sion 1.23.5
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
data = pd.read_csv('../input/amazon-fine-food-reviews/Reviews.csv')
```

In [3]:
```
data = data.head(500)
print(data.shape)
data.info()
```

```
(500, 10)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Id                      500 non-null    int64
 1   ProductId               500 non-null    object
 2   UserId                  500 non-null    object
 3   ProfileName             500 non-null    object
 4   HelpfulnessNumerator    500 non-null    int64
 5   HelpfulnessDenominator  500 non-null    int64
 6   Score                   500 non-null    int64
 7   Time                    500 non-null    int64
 8   Summary                 500 non-null    object
 9   Text                    500 non-null    object
dtypes: int64(5), object(5)
memory usage: 39.2+ KB
```

In [4]:
```
print(data.columns)
data.head()
```

```
Index(['Id', 'ProductId', 'UserId', 'ProfileName', 'HelpfulnessNumerator',
       'HelpfulnessDenominator', 'Score', 'Time', 'Summary', 'Text'],
      dtype='object')
```

Out[4]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenomina
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	

EDA

In [5]:

```
ax = data['Score'].value_counts().sort_index() \
        .plot(kind='bar', title='Reiviews by star', figsize = (6, 3))
ax.set_xlabel('Review Stars')
plt.show()
```

VADER Sentiment Scoring

In [6]:
```
from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm

sia = SentimentIntensityAnalyzer()
```

```
/opt/conda/lib/python3.10/site-packages/nltk/twitter/__init__.py:20: UserWarning:
The twython library has not been installed. Some functionality from the twitter pa
ckage will not be available.
  warnings.warn("The twython library has not been installed. "
```

In [7]:
```
sia.polarity_scores("I am not in a good mood today!")
```

Out[7]:
```
{'neg': 0.35, 'neu': 0.65, 'pos': 0.0, 'compound': -0.4015}
```

In [8]:
```
# running the polarity score on the entire dataset
result = dict()
for i, row in tqdm(data.iterrows(), total=len(data)):
    text = row['Text']
    myid = row['Id']
    result[myid] = sia.polarity_scores(text)
```

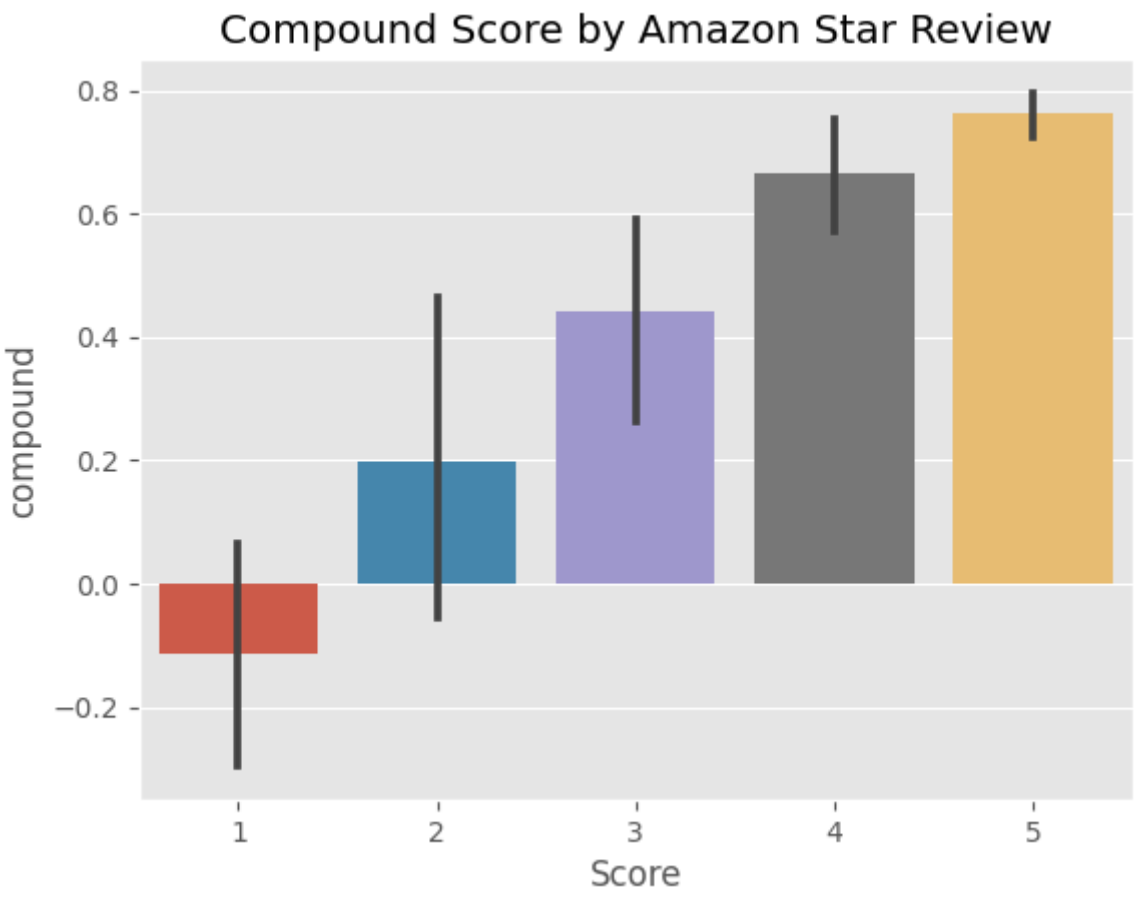
In [9]:
```
vaders = pd.DataFrame(result).T
vaders = vaders.reset_index().rename(columns={'index': 'Id'})
vaders = vaders.merge(data, how='left')
vaders.head(2)
```

Out[9]:

	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessI
0	1	0.000	0.695	0.305	0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	
1	2	0.079	0.853	0.068	-0.1027	B00813GRG4	A1D87F6ZCVE5NK	dll pa	

Plot VADERS Result

In [10]:
```
ax = sns.barplot(data=vaders, x='Score', y='compound')
ax.set_title('Compound Score by Amazon Star Review')
plt.show()
```

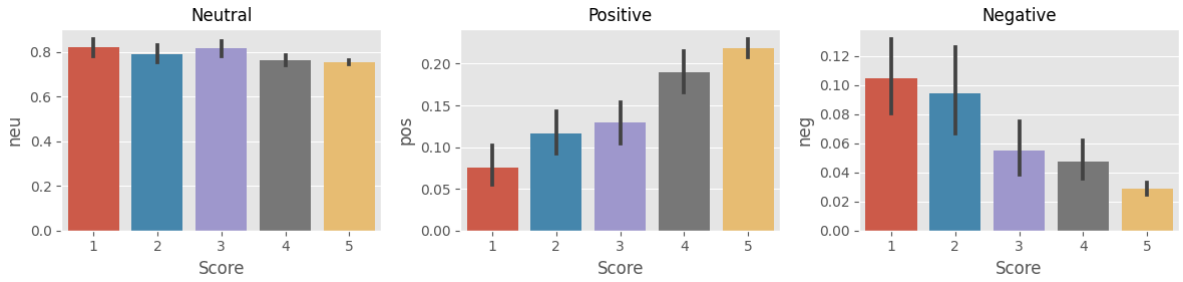


Representing Neutral, Positive & Negative scores on charts

In [11]:
```
fig, axes = plt.subplots(1, 3, figsize=(12, 3))
sns.barplot(data=vaders, x='Score', y='neu', ax=axes[0])
sns.barplot(data=vaders, x='Score', y='pos', ax=axes[1])
sns.barplot(data=vaders, x='Score', y='neg', ax=axes[2])

axes[0].set_title('Neutral', font='Jetbrains Mono', fontsize=12)
axes[1].set_title('Positive', font='Jetbrains Mono', fontsize=12)
axes[2].set_title('Negative', font='Jetbrains Mono', fontsize=12)

plt.tight_layout()
plt.show()
```



Step 2: Roberta Pretrained Model

- Use a model trained on a large corups of data
- Transformer model accounts for the words but also the content related to other words.

In [12]:
```
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from scipy.special import softmax
```

In [13]:
```
MODEL = f'cardiffnlp/twitter-roberta-base-sentiment'
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

In [14]:
```
example = 'Seems like I would get to eat some tasty food today, but I don\'t actua

# running for roberta model
encoded_text = tokenizer(example, return_tensors='pt')
output = model(**encoded_text)
scores = output[0][0].detach().numpy()
scores = softmax(scores)

scores_dict = {
    'roberta_neg': scores[0],
    'roberta_neu': scores[1],
    'roberta_pos': scores[2]
}

scores_dict
```

Out[14]:
```
{'roberta_neg': 0.803285,
 'roberta_neu': 0.16965581,
 'roberta_pos': 0.027059125}
```

In [15]:
```
def polarity_scores_roberta(example):
    encoded_text = tokenizer(example, return_tensors='pt')
    output = model(**encoded_text)
    scores = output[0][0].detach().numpy()
    scores = softmax(scores)

    scores_dict = {
        'roberta_neg': scores[0],
        'roberta_neu': scores[1],
        'roberta_pos': scores[2]
    }

    return scores_dict
```

In [16]:
```
result = dict()
for i, row in tqdm(data.iterrows(), total=len(data)):
    try:
        text = row['Text']
        myid = row['Id']

        vader_result = sia.polarity_scores(text)
        vader_result_rename = dict()
        for key, value in vader_result.items():
            vader_result_rename[f'vader_{key}'] = value

        roberta_result = polarity_scores_roberta(text)
        both = {**vader_result_rename, **roberta_result}

        result[myid] = both
    except Exception as error:
        print(f'{error} occured at {myid} id')
```

```
The expanded size of the tensor (571) must match the existing size (514) at non-si
ngleton dimension 1.  Target sizes: [1, 571].  Tensor sizes: [1, 514] occured at 8
3 id
The expanded size of the tensor (546) must match the existing size (514) at non-si
ngleton dimension 1.  Target sizes: [1, 546].  Tensor sizes: [1, 514] occured at 1
87 id
```

In [17]:
```
results_df = pd.DataFrame(result).T
results_df = results_df.reset_index().rename(columns={'index': 'Id'})
results_df = results_df.merge(data, how='left')
```

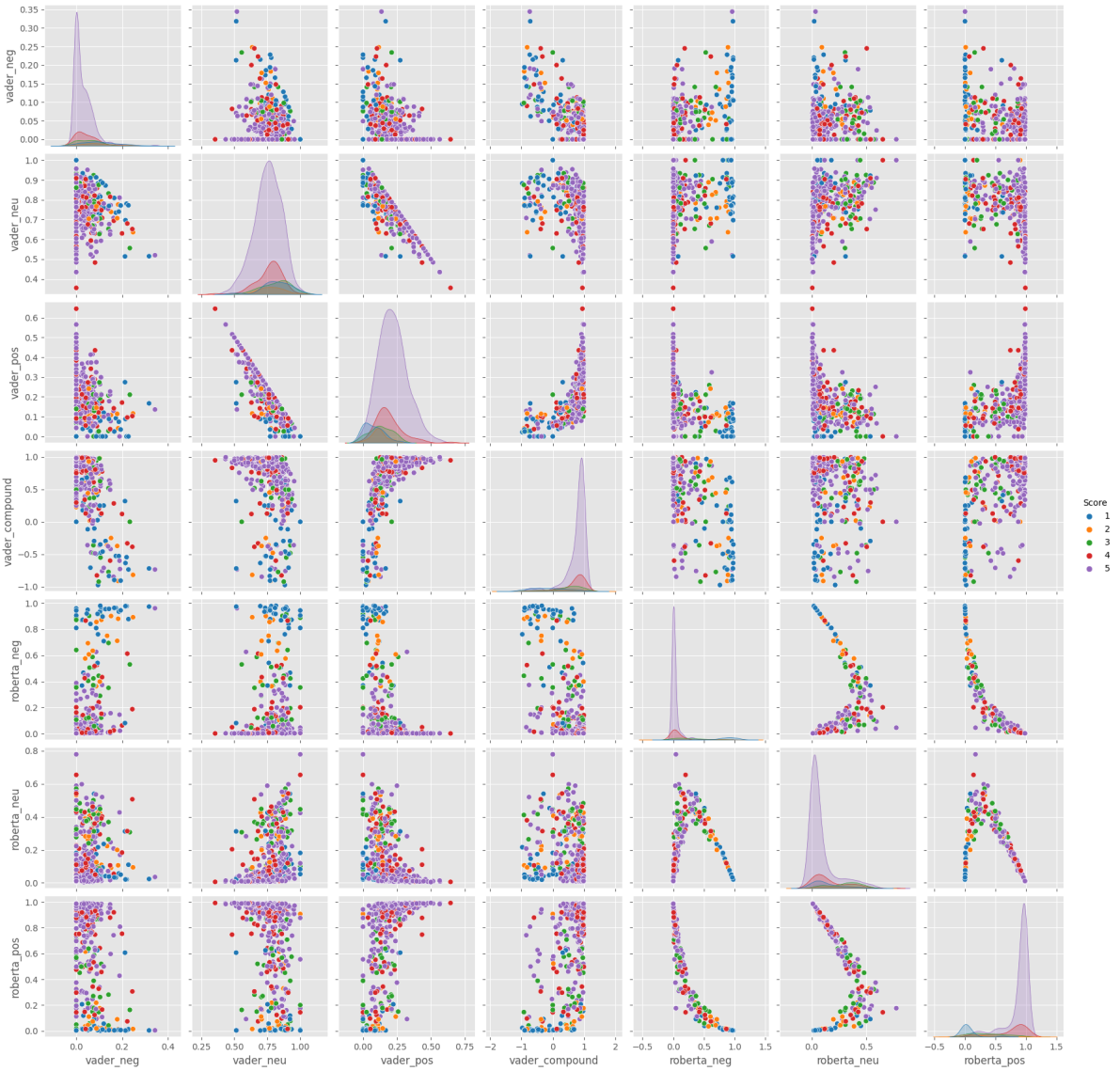
Compare scores between both models

In [18]:
```
results_df.columns
```

Out[18]:
```
Index(['Id', 'vader_neg', 'vader_neu', 'vader_pos', 'vader_compound',
       'roberta_neg', 'roberta_neu', 'roberta_pos', 'ProductId', 'UserId',
       'ProfileName', 'HelpfulnessNumerator', 'HelpfulnessDenominator',
       'Score', 'Time', 'Summary', 'Text'],
      dtype='object')
```

In [19]:
```
sns.pairplot(data=results_df,
             vars=['vader_neg', 'vader_neu', 'vader_pos', 'vader_compound',
                   'roberta_neg', 'roberta_neu', 'roberta_pos'],
             hue='Score', palette='tab10')

plt.show()
```



Step 3: Review Examples

- Positive 1-star and negative 5-star reviews

Lets look at some examples where the model scoring and review score differ the most

In [20]:
```
# checking for roberta_model
results_df.query('Score == 1').sort_values('roberta_pos', ascending=False)[['Text'
```

Out[20]:
```
array(['I felt energized within five minutes, but it lasted for about 45 minutes.
I paid $3.99 for this drink. I could have just drunk a cup of coffee and saved my
money.',
       0.625636637210846], dtype=object)
```

In [21]:
```
# checking for vader model
results_df.query('Score == 1').sort_values('vader_pos', ascending=False)[['Text',
```

Out[21]:
```
array(['So we cancelled the order.  It was cancelled without any problem.  That is
a positive note...',
       0.6071790456771851], dtype=object)
```

In [22]:
```
# checking for roberta_model
results_df.query('Score == 5').sort_values('roberta_neg', ascending=False)[['Text'
```

Out[22]:
```
array(['this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault',
       0.004412965849041939], dtype=object)
```

In [23]:
```
# checking for vader model
results_df.query('Score == 5').sort_values('vader_neg', ascending=False)[['Text',
```

Out[23]:
```
array(['this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault',
       0.004412965849041939], dtype=object)
```

The transformers Pipeline

In [24]:
```
from transformers import pipeline

sent_pipeline = pipeline('sentiment-analysis')
```

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not recommended.
```

In [25]:
```
text = "This is good for my health, right?"
sent_pipeline(text)
```

Out[25]:
```
[{'label': 'POSITIVE', 'score': 0.9995703101158142}]
```