


```
len(reviews)
```

```
13623233
```

```
soup = BeautifulSoup(reviews, 'html.parser')
regex = re.compile('.*comment.*')
results = soup.find_all('p', {'class': regex})
customer_reviews = [result.text for result in results]
```

```
customer_reviews[0]
```

```
'It was ok. The coffee wasn't the best but it was fine. The relish on the breakfast roll was yum which did make it sing. So perhaps I ju
my visit.'
```

```
store_review = customer_reviews[1]
store_review_score = sentiment_analysis(reviews_data['reviews'].iloc[1])
print(f"{store_review} --> the sentiment score is --> {store_review_score}")
```

```
if store_review_score >= 4:
    print('POSITIVE REVIEW')
elif store_review_score == 3:
    print("NEUTRAL REVIEW")
else :
    print("NEGATIVE REVIEW")
```

```
This place is a gem. The ambiance is to die for. The service is really nice. The coffee is a must. Ah I can still remember how that Ice
POSITIVE REVIEW
```

```
reviews_data = pd.DataFrame(np.array(customer_reviews), columns=['reviews'])
```

```
reviews_data.sample(5)
```

	reviews	
1	This place is a gem. The ambiance is to die fo...	
6	I'm so glad I found Social Brew! It was so won...	
71	Great Ambiance, friendly service. Overall a wi...	
18	What probably would have been a 4 star rating ...	
13	Sunday brunch was very busy! However the staff...	

```
reviews_data.shape
```

```
(133, 1)
```

```
def sentiment_analysis(review):
    tokens = import_bert_tokenizer.encode(review, return_tensors='pt')
    result = bert_model(tokens)
    return int(torch.argmax(result.logits)) + 1
```

```
reviews_data['sentimental_score'] = reviews_data['reviews'].apply(lambda x : sentiment_analysis(x[:512]))
```

```
reviews_data.head()
```

	reviews	sentimental_score	
0	It was ok. The coffee wasn't the best but it w...	3	
1	This place is a gem. The ambiance is to die fo...	3	
2	I went here a little while ago- a beautiful mo...	2	
3	Ron & Jo are on the go down under and Wow! We...	5	
4	5 stars all around for the staff and delicious...	5	

```
def sentimental_string_score(reviews):
    if reviews >= 4:
        return 'POSITIVE REVIEW'
    elif reviews == 3:
        return 'NEUTRAL REVIEW'
    else :
        return 'NEGATIVE REVIEW'
reviews_data['sentimental_string_score'] = reviews_data['sentimental_score'].apply(sentimental_string_score)

reviews_data.sample(5)
```

	reviews	sentimental_score	sentimental_string_score	
53	Wonderful settings, nice service. Came 3 times...	5	POSITIVE REVIEW	
117	Charming place with fantastic service, and ver...	5	POSITIVE REVIEW	
74	We're only visiting for a week on business, an...	1	NEGATIVE REVIEW	
47	Great atmosphere. Food perfect to rejuvenate a...	5	POSITIVE REVIEW	
116	Wonderful food & super friendly gals working t...	5	POSITIVE REVIEW	

```
reviews_data['sentimental_score'].value_counts()
```

```
5    84
4    30
3     9
2     5
1     5
Name: sentimental_score, dtype: int64
```

```
reviews_data['sentimental_string_score'].value_counts()
```

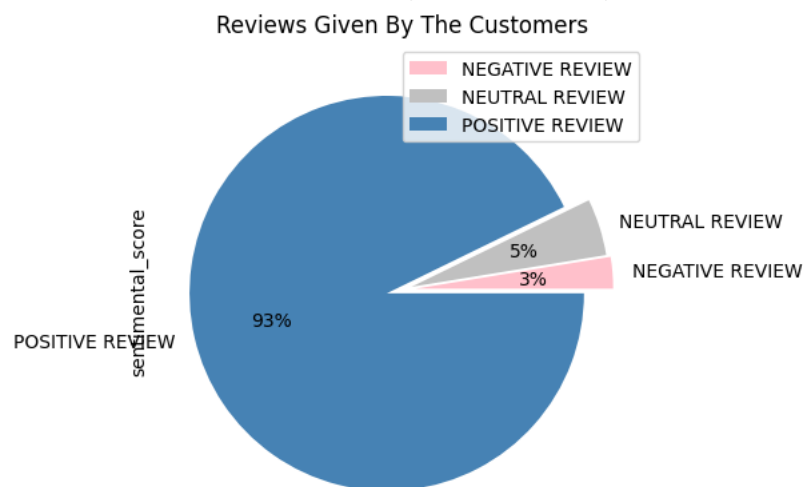
```
POSITIVE REVIEW    114
NEGATIVE REVIEW     10
NEUTRAL REVIEW      9
Name: sentimental_string_score, dtype: int64
```

```
colors = ['pink', 'silver', 'steelblue']
```

```
explode = (0.07, 0.05, 0.08)
```

```
reviews_data.groupby(['sentimental_string_score']).sum().plot(kind='pie', y='sentimental_score', autopct='%1.0f%%',
                    colors = colors, explode=explode,
                    title='Reviews Given By The Customers')
```

```
<ipython-input-167-05ce9b3a510e>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future
reviews_data.groupby(['sentimental_string_score']).sum().plot(kind='pie', y='sentimental_score', autopct='%1.0f%%',
<Axes: title={'center': 'Reviews Given By The Customers'}, ylabel='sentimental_score'>
```



Conclusion : The data is extracted from yelp website using BeautifulSoup, the sentimental score of the extracted reviews are done using pretrained bert model the majority i.e 93% reviews given by the customers are positive and only 3% of the customers gave bad reviews and 5% of the customers are just satisfied with the food and restaurant

