Sentiment Analysis
Jayaprakash (prakashjz121@gmail.com)

Problem statement : Customer Reviews – Sentiment Analysis Business Use case

What is sentiment analysis?

- *Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative our neutral.*

- *Understanding people's emotions is essential for businesses since customers are able to express their thoughts and feelings more openly than ever before.*

- *It is quite hard for a human to go through each single line and identify the emotion being the user experience. Now with technology, we can automatically analyzing customer feedback, from survey responses to social media conversations, brands are able to listen attentively to their customers, and tailor products and services to meet their needs.*

- *"Your most unhappy customers are your greatest source of learning." — Bill Gates*

Problem Statement

- *You are working in an e-commerce company, and your company has put forward a task to analyze the customer reviews for various products. You are supposed to create a report that classifies the products based on the customer reviews.*

Project Objective

- Find various trends and patterns in the reviews data, create useful insights that best describe the product quality, and
- Classify each review based on the sentiment associated with the same.

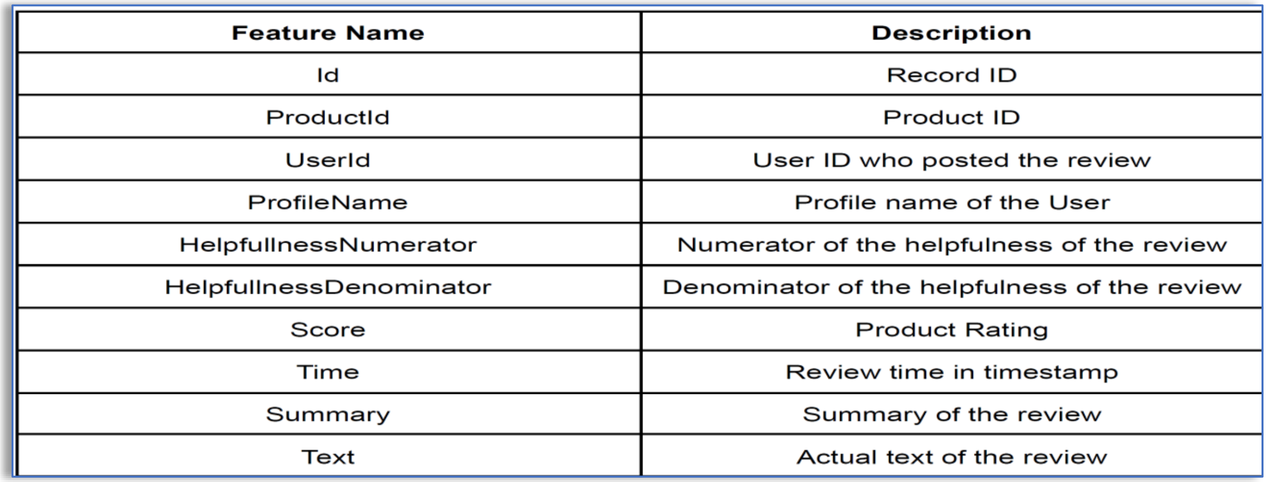
Data Description

- Given dataset contains 568423rows and 10 columns,
- Primary key is UserId , and
- Data volume is very HIGH , processing time took more than expected.

Data Pre-processing Steps and Inspiration

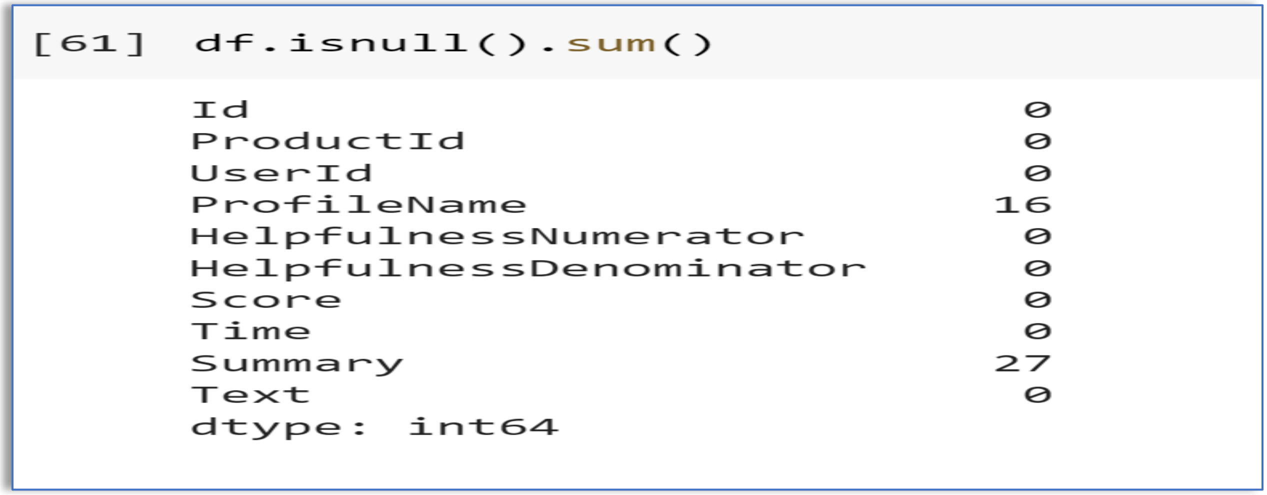
Data Cleanup

- During data analysis, few rows/columns had null values which were cleaned up to improve the data quality,
- Columns details in the given dataset,



Feature Name	Description
Id	Record ID
ProductId	Product ID
UserId	User ID who posted the review
ProfileName	Profile name of the User
HelpfullnessNumerator	Numerator of the helpfulness of the review
HelpfullnessDenominator	Denominator of the helpfulness of the review
Score	Product Rating
Time	Review time in timestamp
Summary	Summary of the review
Text	Actual text of the review

- Validate the null values and drop them if exists, and

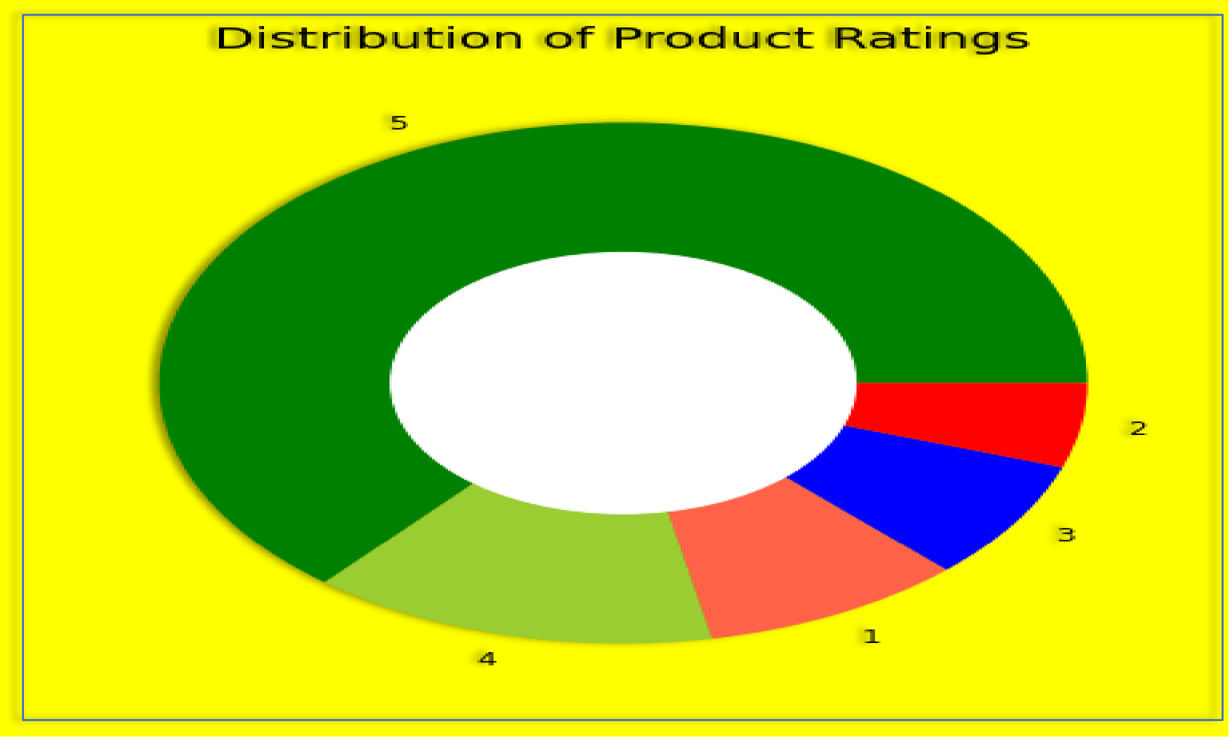


```
[61]  df.isnull().sum()

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

- Text column cleanup: Have cleaned up text data like - make text lowercase, remove text in square brackets, remove links, remove punctuation and remove words containing numbers
- Transform timestamp columns.
  - Time Data in given dataset was in unusual format (in milliseconds) so split the data into month and year for better data analysis,
  - Extract time attributes/features (year, month) from time data columns,
  - Evaluate the e-commerce scenario using these features, and
- New feature called ‘Customer review’ (summary and Text columns) have been combined for better assessment.

Exploratory Data Analysis (EDA)

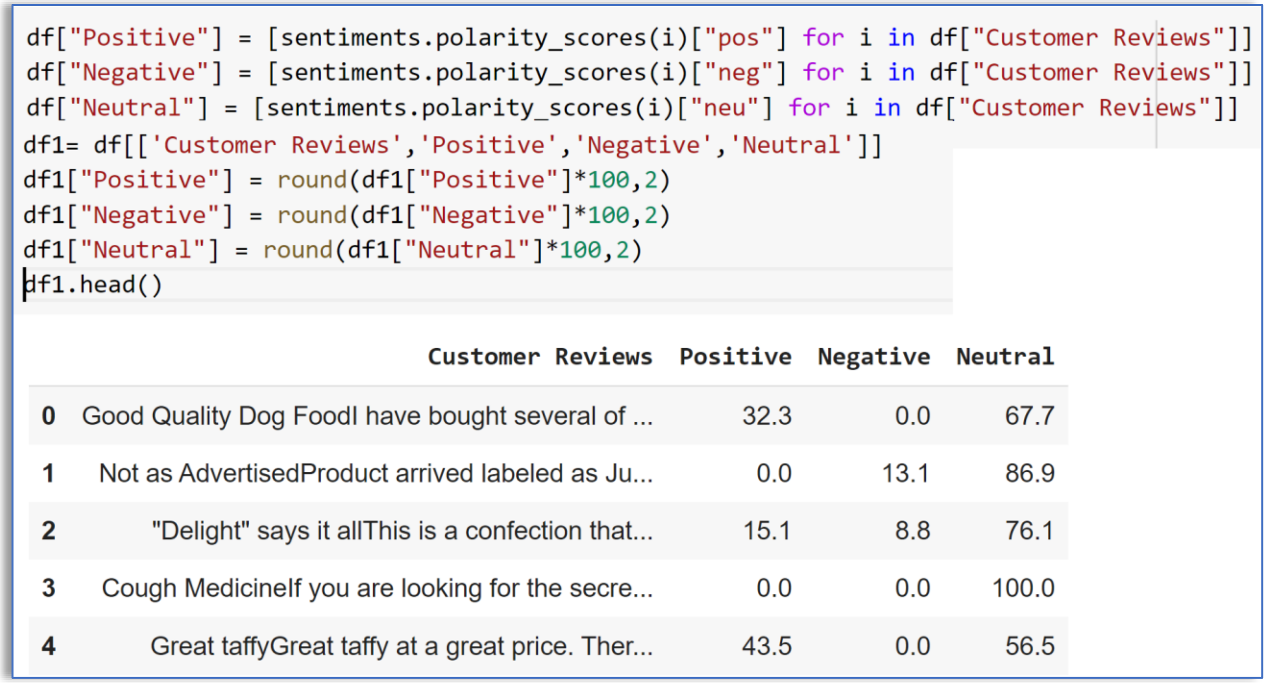
- Distribution on product ratings ,



- According to the pie chart plotted on score column, more than half of people rated products they bought from e- commerce site, with 5 stars, which is good , and
- Let's add three more features (columns) to this dataset as *Positive, Negative, and Neutral* by calculating the sentiment scores of the customer reviews mentioned in the Text column of the dataset:

Sentiment Score review

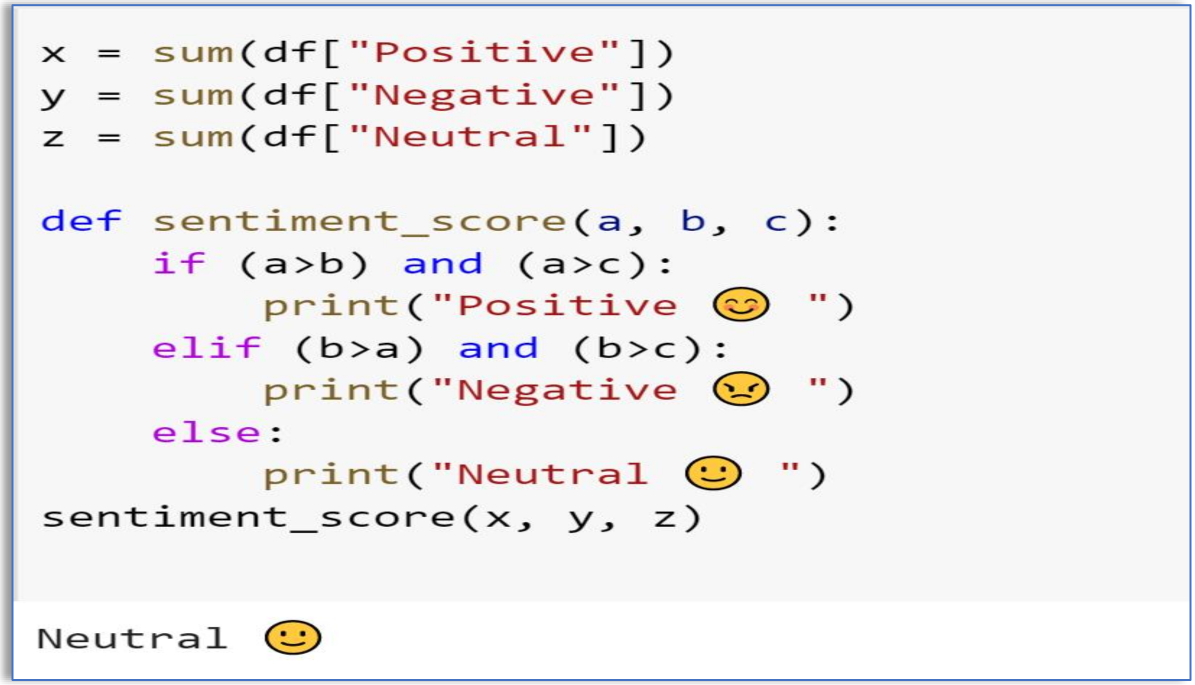
- This is an important preprocessing phase, the outcome column (sentiment of review) based on the overall score using *SentimentIntensityAnalyzer* & VADER technique in Python,
- Have used new feature 'Customer review' to calculate the polarity score ,



```
df["Positive"] = [sentiments.polarity_scores(i)["pos"] for i in df["Customer Reviews"]]
df["Negative"] = [sentiments.polarity_scores(i)["neg"] for i in df["Customer Reviews"]]
df["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i in df["Customer Reviews"]]
df1= df[['Customer Reviews','Positive','Negative','Neutral']]
df1["Positive"] = round(df1["Positive"]*100,2)
df1["Negative"] = round(df1["Negative"]*100,2)
df1["Neutral"] = round(df1["Neutral"]*100,2)
df1.head()
```

	Customer Reviews	Positive	Negative	Neutral
0	Good Quality Dog FoodI have bought several of ...	32.3	0.0	67.7
1	Not as AdvertisedProduct arrived labeled as Ju...	0.0	13.1	86.9
2	"Delight" says it allThis is a confection that...	15.1	8.8	76.1
3	Cough MedicineIf you are looking for the secre...	0.0	0.0	100.0
4	Great taffyGreat taffy at a great price. Ther...	43.5	0.0	56.5

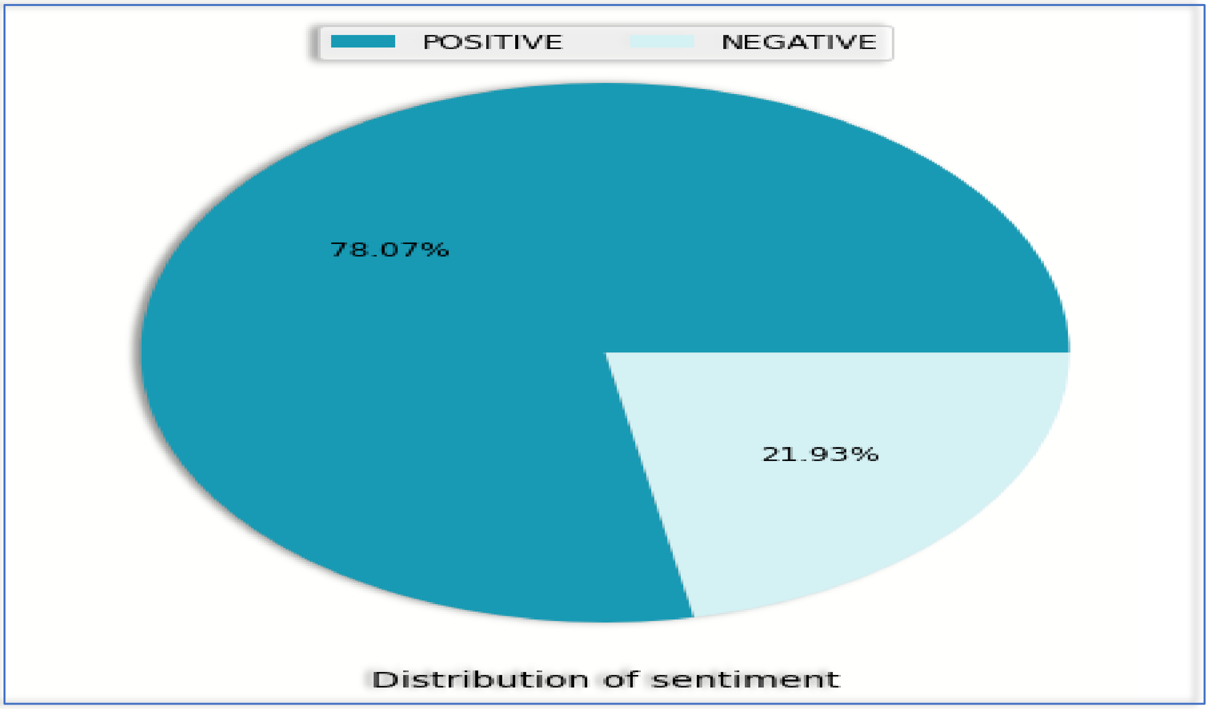
- Above snapshot shows the sentiment score % for the given dataset / text columns,



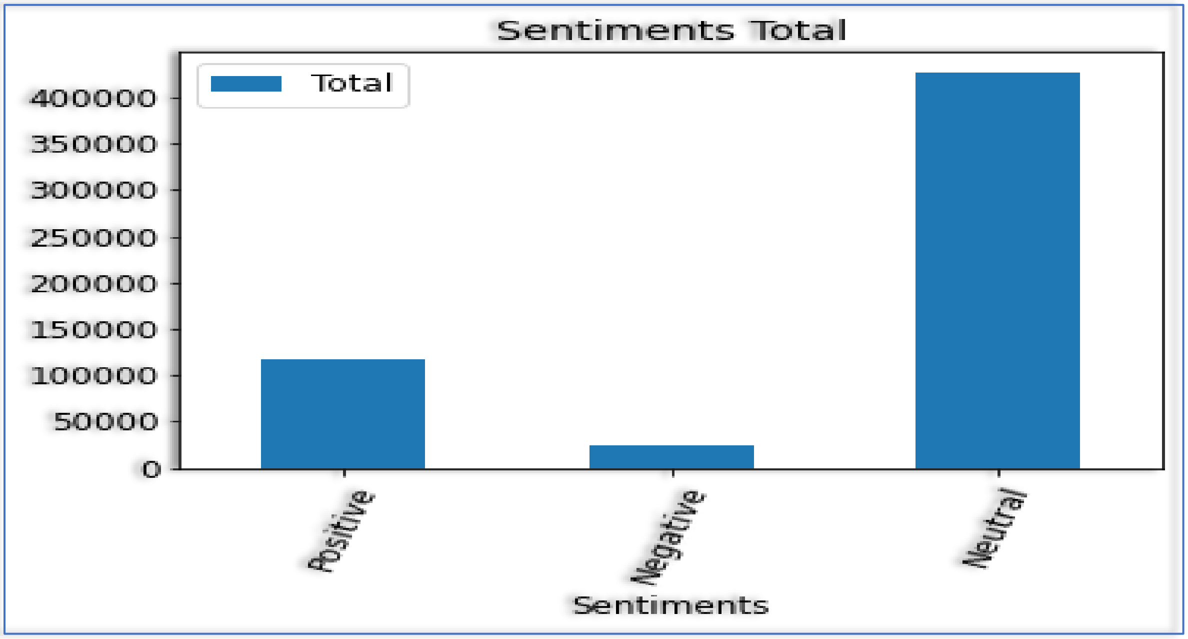
```
x = sum(df["Positive"])
y = sum(df["Negative"])
z = sum(df["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

- So, most people are neutral when submitting their experiences with the products they have purchased from shopping ecommerce site. Now let's see the total of all sentiment scores , and



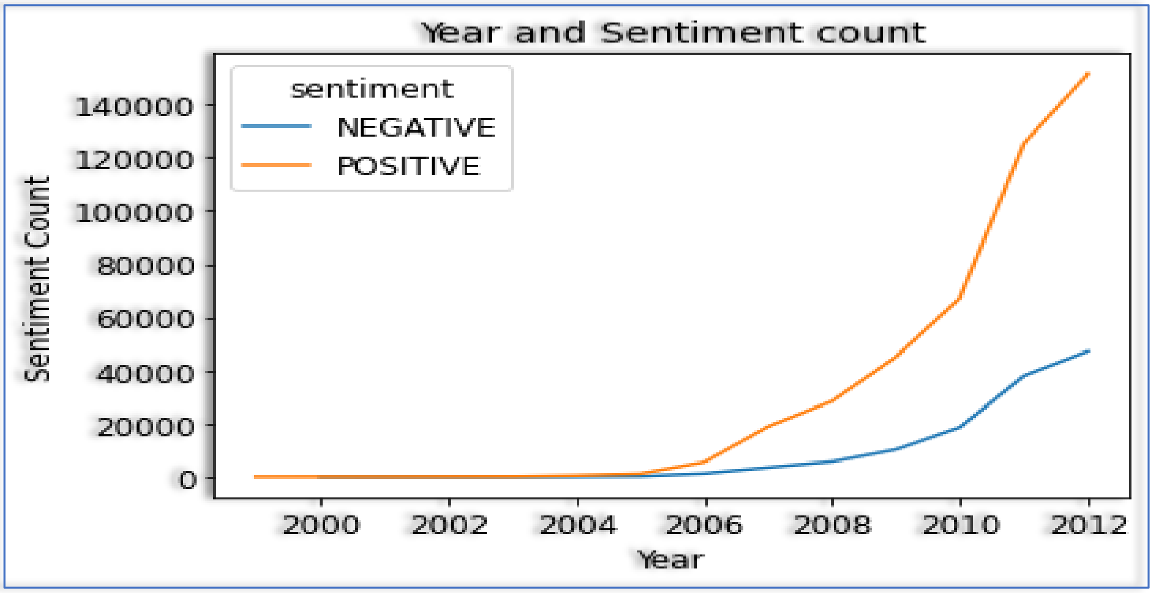
- The positive reviews are greater than negative reviews , and
- From the above pie chart for the same, the positive sentiments are more than negative which can build understanding as people are happy with service.



- From the above chart , seems like most of the reviews of the products available on ecommerce site are positive, as the total sentiment scores of Positive and Neural are much higher than Negative scores , so it looks good for further analysis.

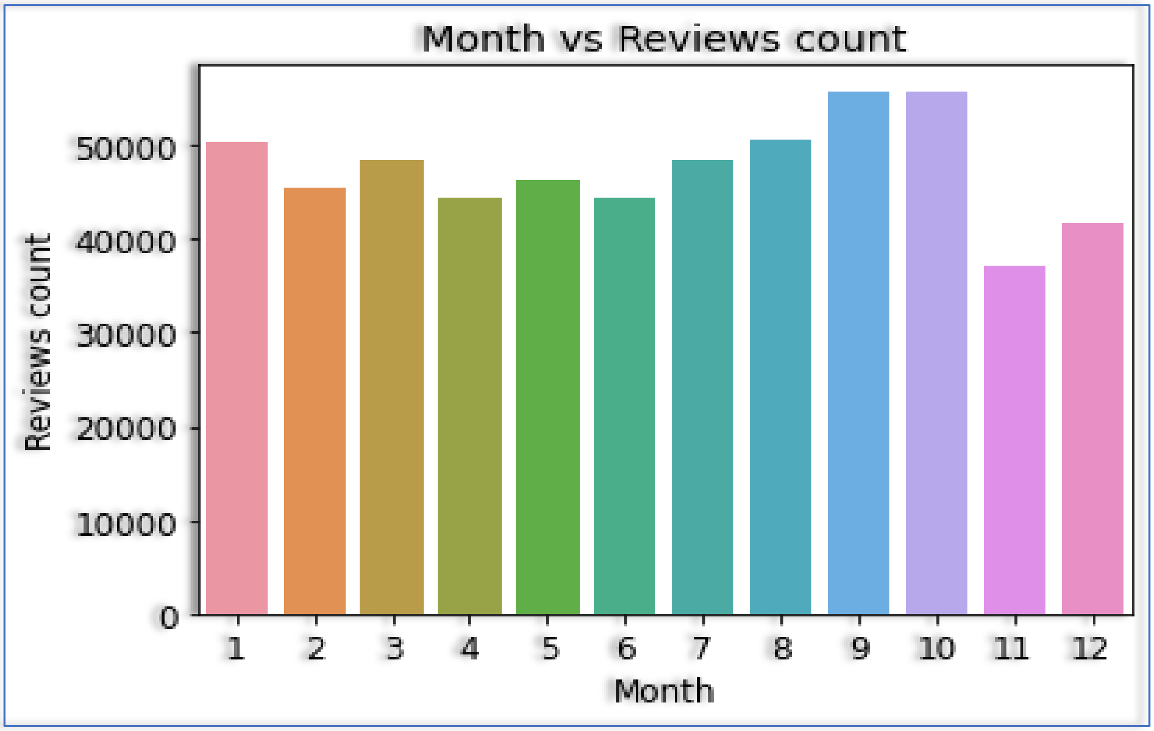
Year VS Sentiment insights

- From the below plot, we can clearly see the rise in positive reviews from 2010. Reaching its peak around 2012, and
- Negative reviews are very low as compared to the positive reviews.



Month Vs Sentiment insights

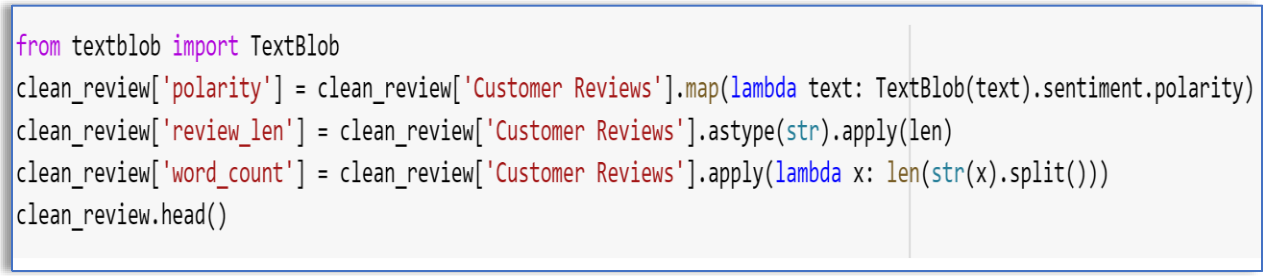
- The customer review counts are more or less uniformly distributed. There isn't much variance between the days. But there is a drop at the end of year.



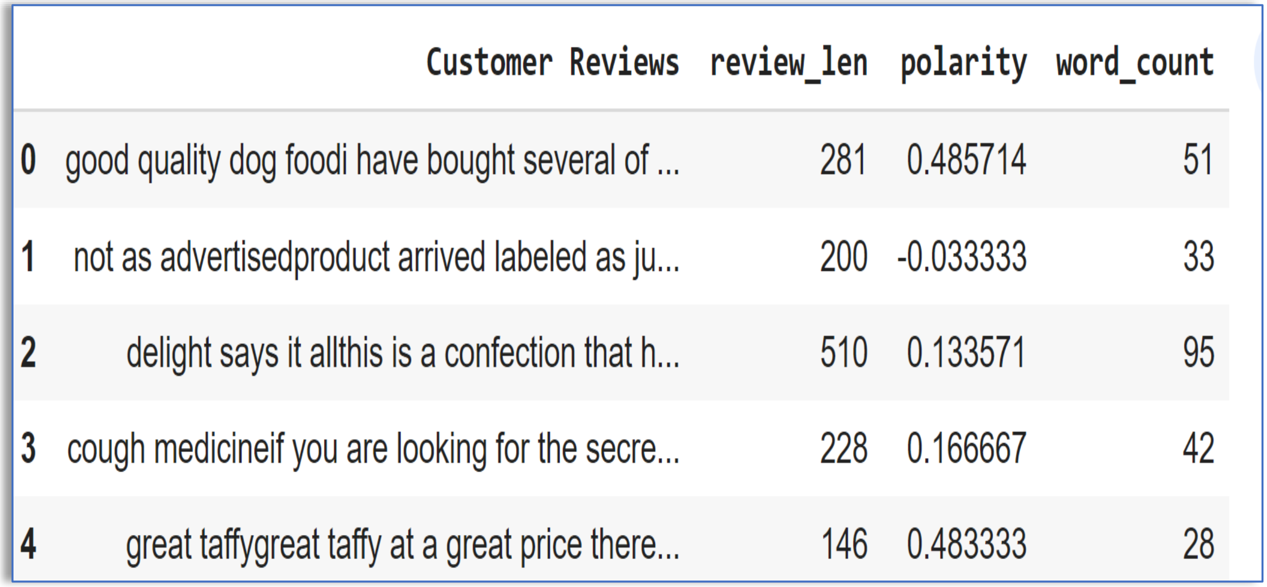
Derive Other Sentiment features

Sentiment polarity distribution

- As a measurement of human opinions and affective states, a sentiment score can often be decomposed into two aspects: the polarity and intensity of the sentiment. I am going to look at the polarity score for further investigation,
- Polarity refers to the strength of an opinion. It could be positive or negative. If something has a strong positive feeling or emotion associated with it, such as admiration, trust, love; this will indeed have a certain orientation towards all other aspects of that object's existence. The same goes for negative polarities.
    - A good example would be the following: 'I don't think I'll buy this item because my previous experience with a similar item wasn't so good.' That will have a negative polarity.
- The key aspect of sentiment analysis is to analyze a body of text for understanding the context and opinion indicated by it. As computers only understand numerical data, we can quantify the sentiment towards the positive and negative sentiment by using Polarity, and
- Polarity Distribution [It is between [-1,1] where -1 is negative and 1 is positive polarity.



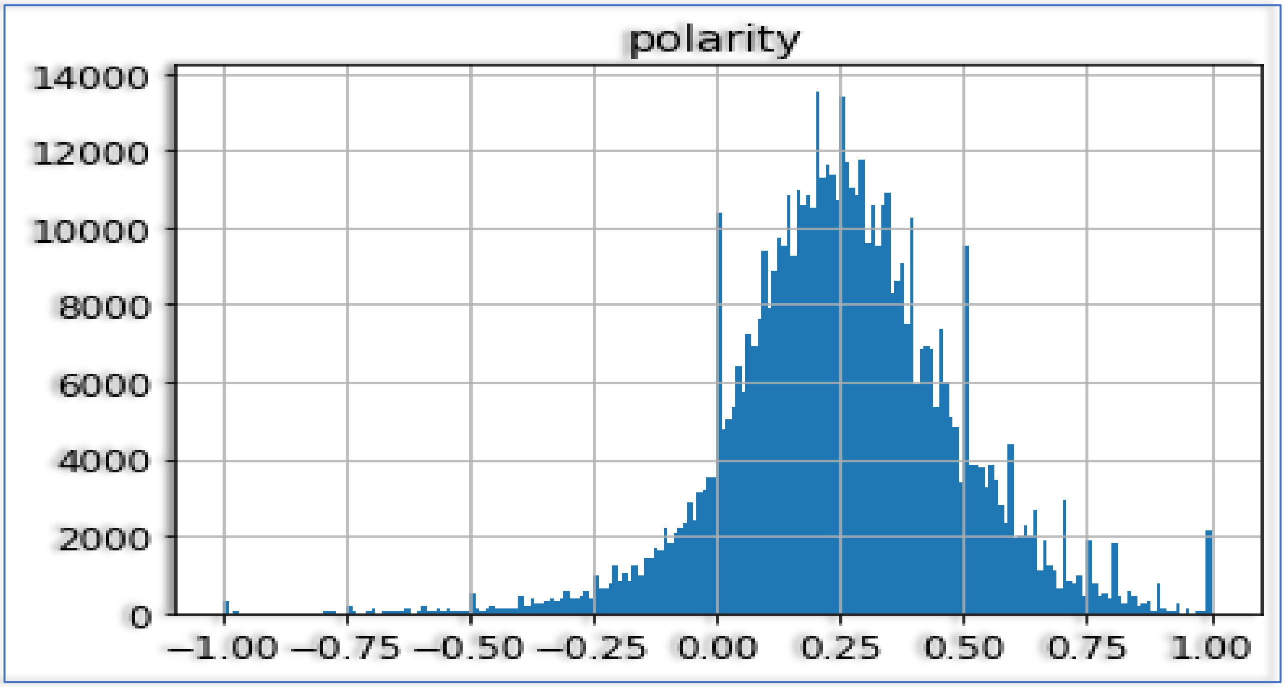
```
from textblob import TextBlob
clean_review['polarity'] = clean_review['Customer Reviews'].map(lambda text: TextBlob(text).sentiment.polarity)
clean_review['review_len'] = clean_review['Customer Reviews'].astype(str).apply(len)
clean_review['word_count'] = clean_review['Customer Reviews'].apply(lambda x: len(str(x).split()))
clean_review.head()
```



	Customer Reviews	review_len	polarity	word_count
0	good quality dog foodi have bought several of ...	281	0.485714	51
1	not as advertisedproduct arrived labeled as ju...	200	-0.033333	33
2	delight says it allthis is a confection that h...	510	0.133571	95
3	cough medicineif you are looking for the secre...	228	0.166667	42
4	great taffygreat taffy at a great price there...	146	0.483333	28

Sentiment polarity distribution

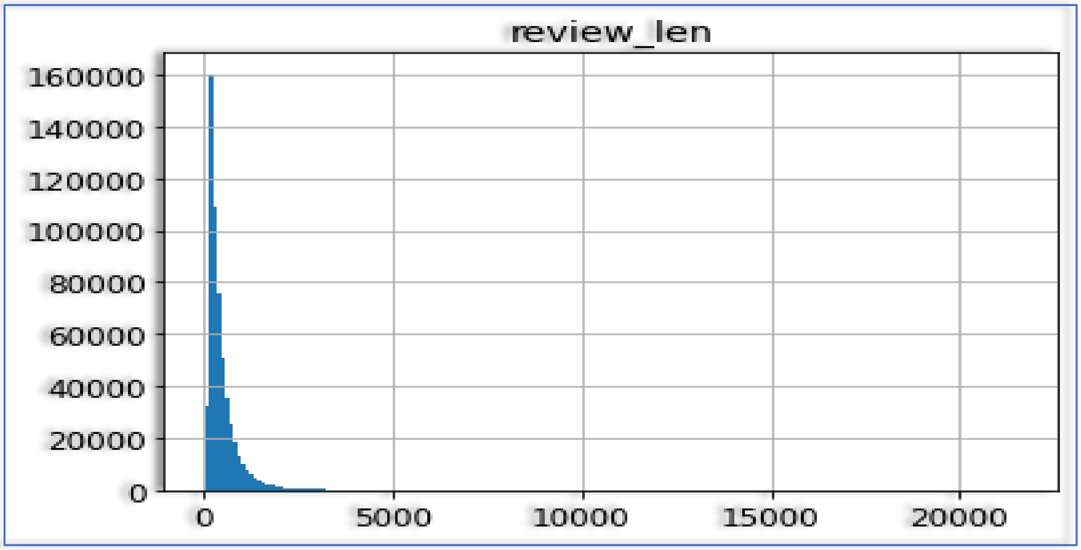
- Polarity Distribution for the new features [It is between [-1,1] where -1 is negative and 1 is positive polarity] ,



- Have a lot of positive polarities compared to the negative polarities ,
- This polarity distributions assures the number of positive reviews in the given dataset , and
- I can say that this polarity is a normally distributed but not standard normal.

Review Text Length Distribution:

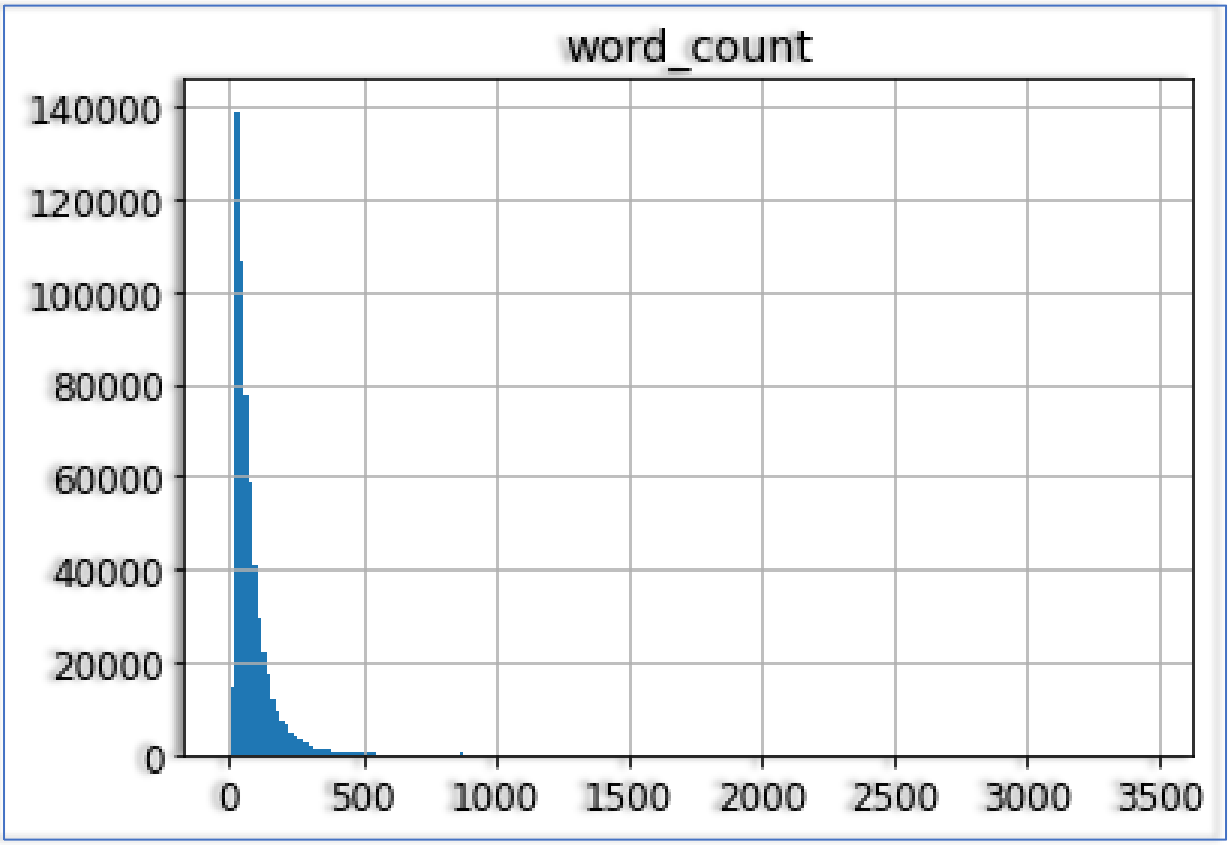
- This is distribution where length of the review which includes each letters and spaces.



- From the above plot , sound like it has right skewed distribution where most of the lengths falls between 0-2000

Review Text Word Count Distribution

- Let's check out the word count of review text, and

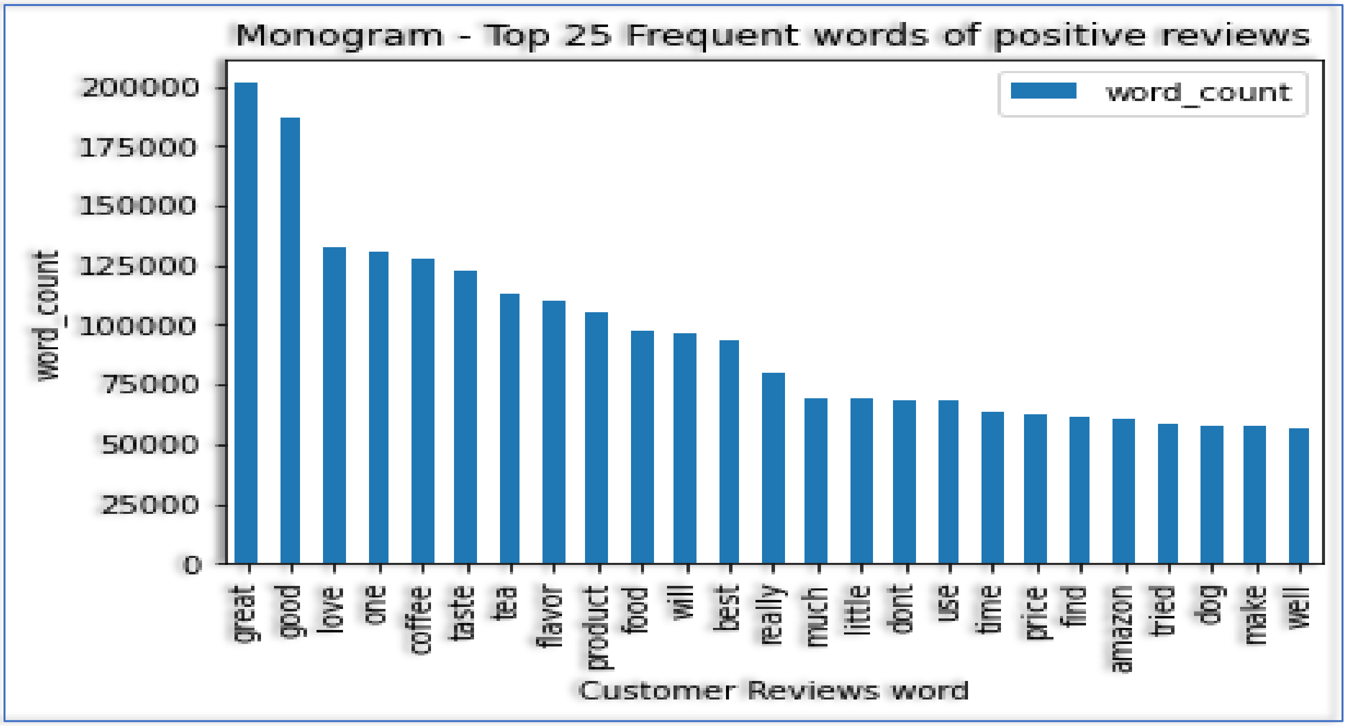


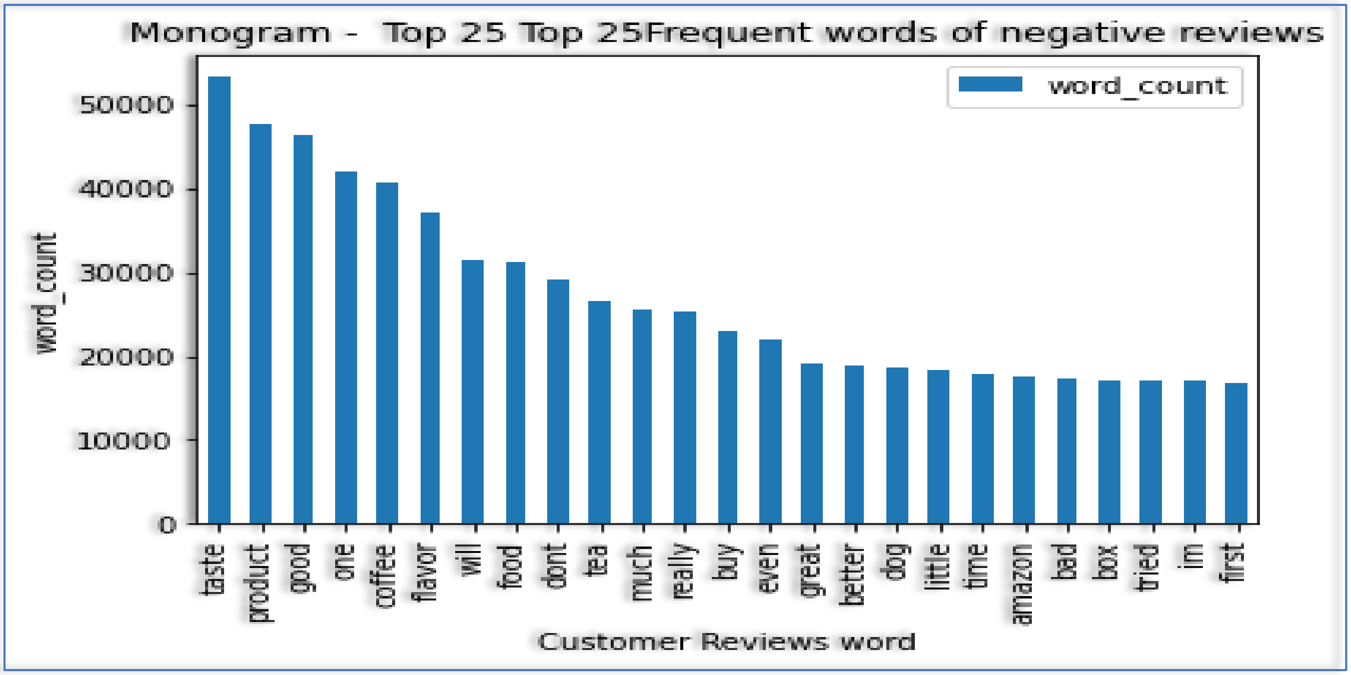
- From the above plot , sound like right skewed distribution with most of the words falling between 0-300 in a review.

N-gram analysis

Monogram analysis

- To plot most frequent of one word in reviews based on sentiments ,

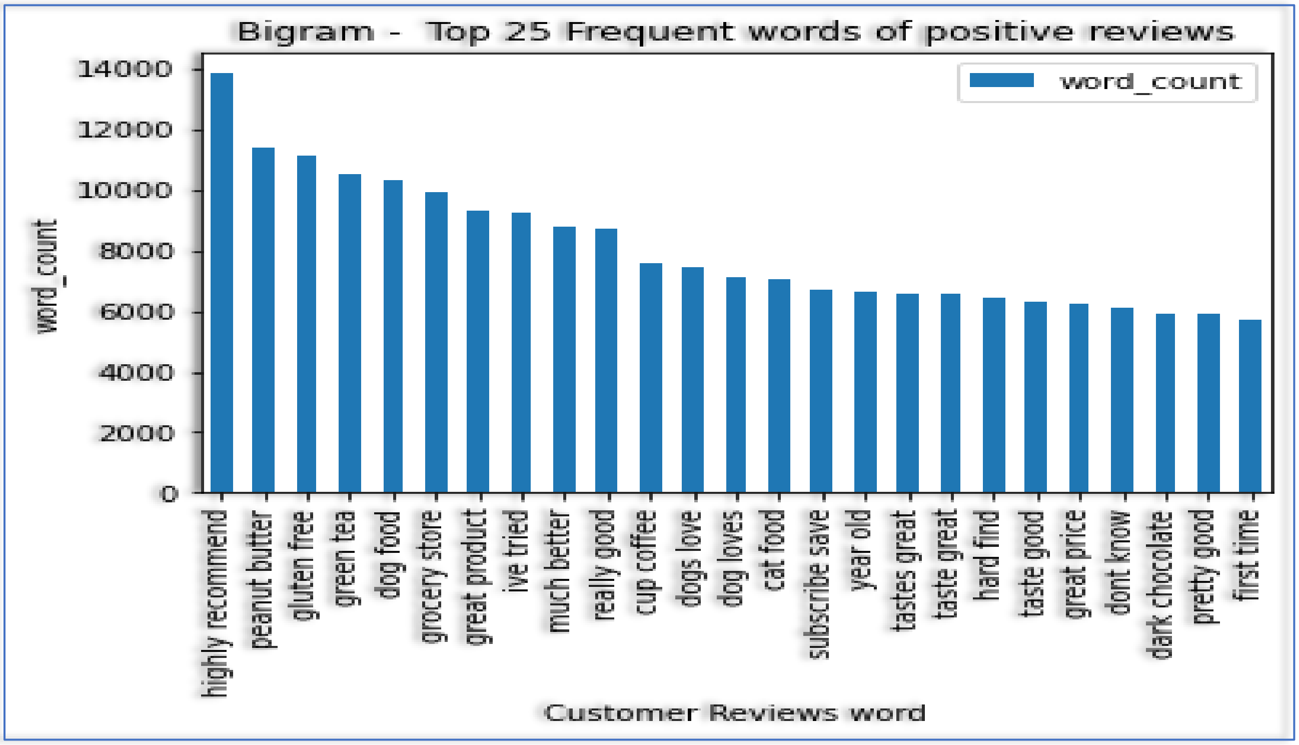


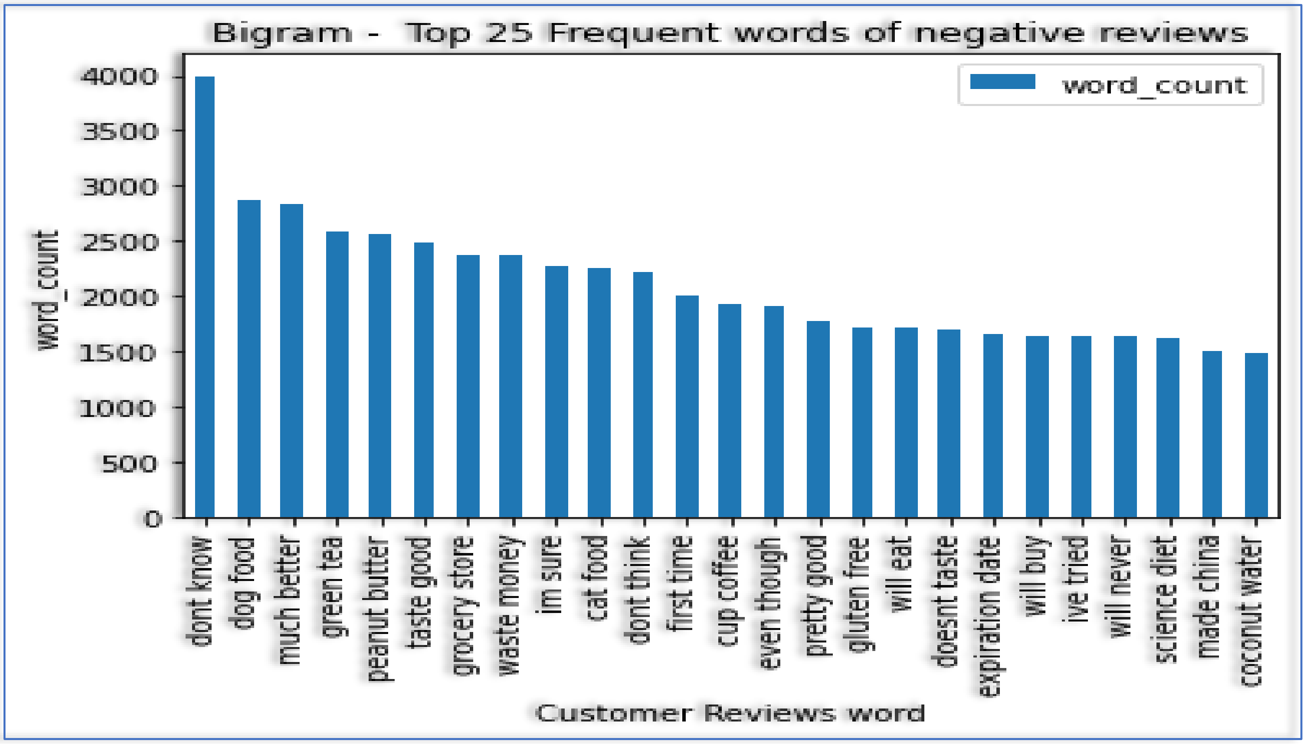


- As we see, the words don’t match with the sentiment except few. Through monogram we can't judge a sentiment based on one word, and
- So let's try with frequent two words

Bigram analysis

- To plot most frequent two words in reviews based on sentiments,

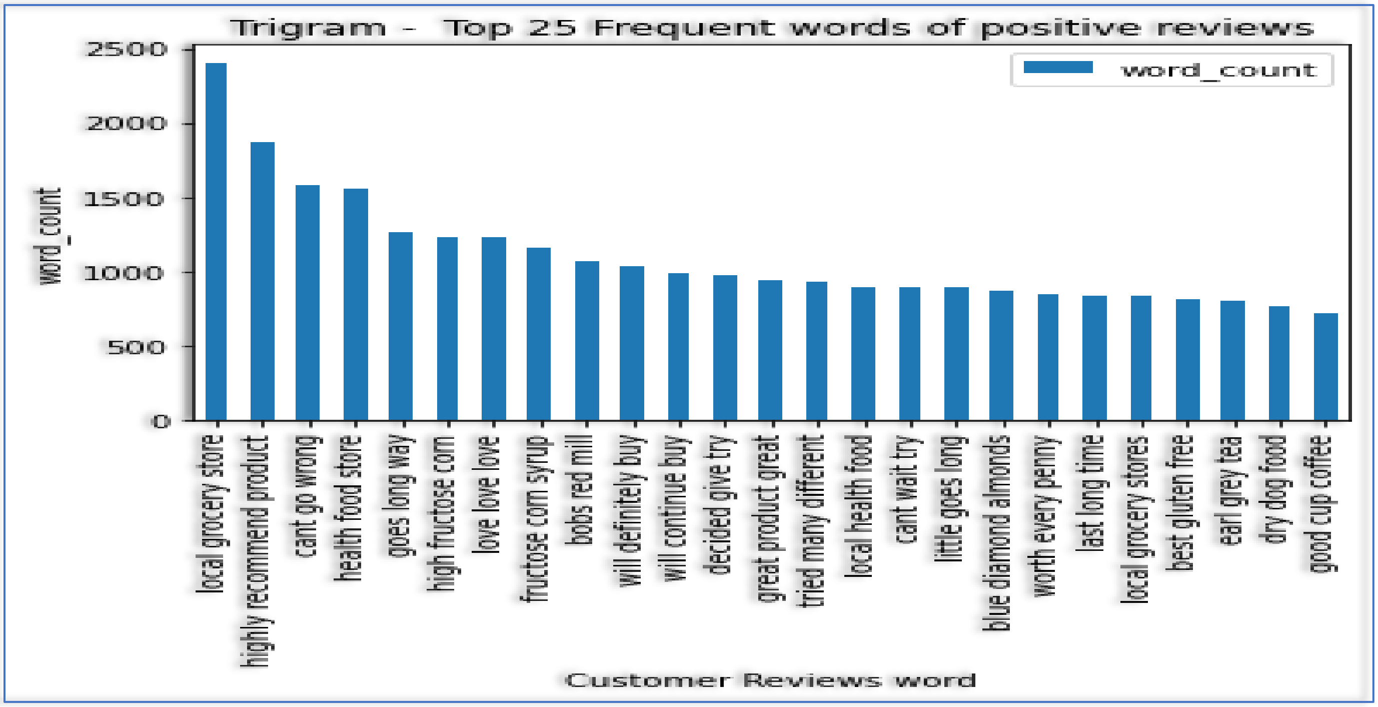


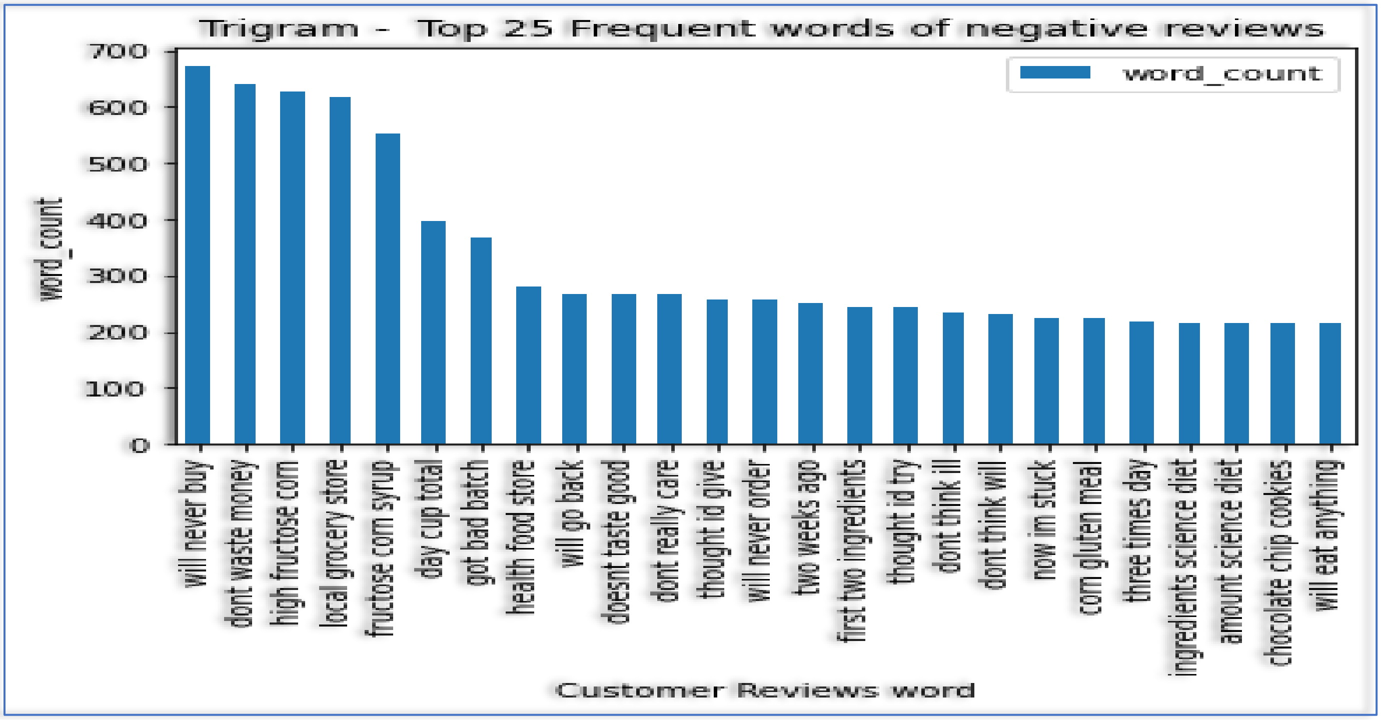


- Above plot gives clear idea about the sentiments from the bi-words, and
- Some of the positive words are Higher side like ‘highly recommend,gluten free’ and negative words are ‘much better, waste money’.

Trigram analysis

- To plot most frequent two words in reviews based on sentiments , and



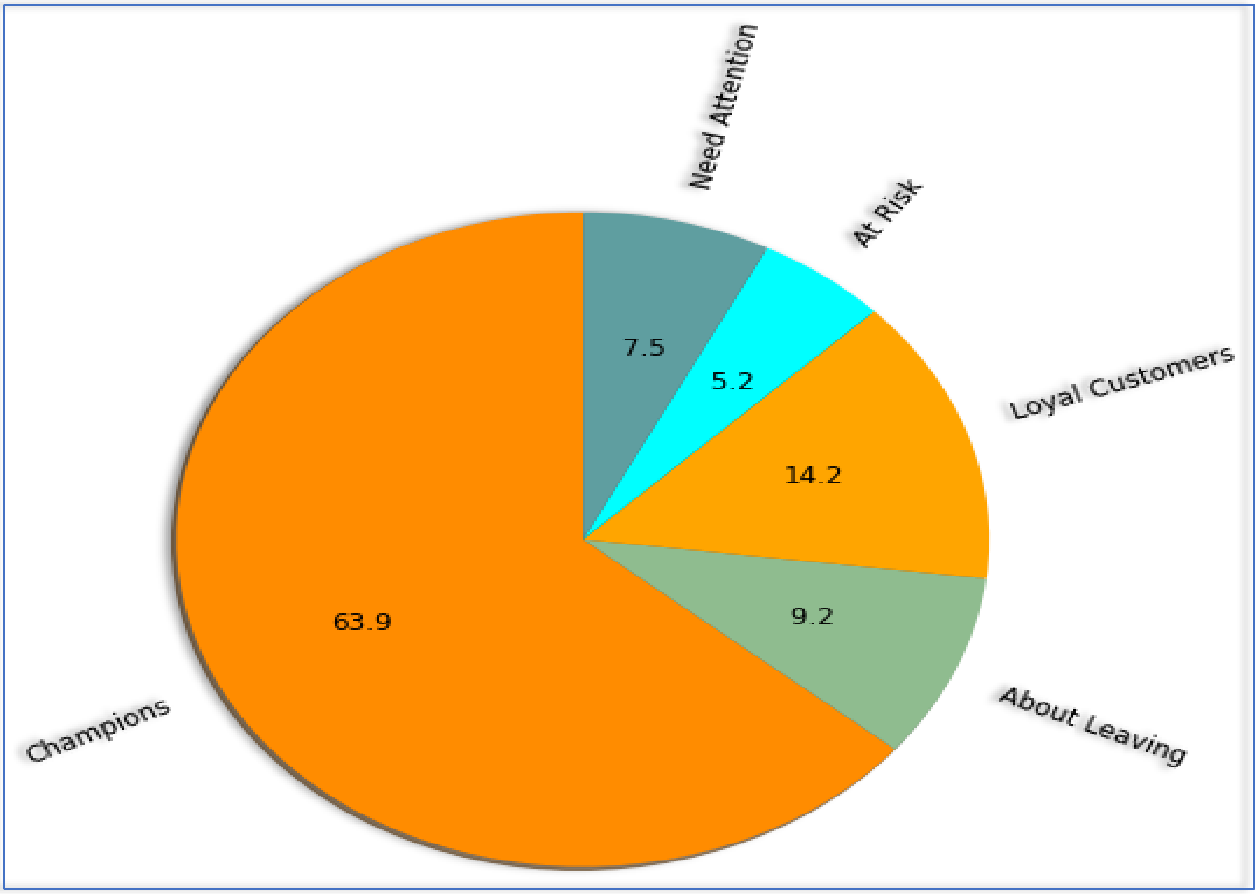


- Above plot gives clear idea about the sentiments from the tri-words, and
- Some of the positive words are Higher side like ‘highly recommend, gluten free’ and negative words are ‘much better, waste money’.

Customer Segments

- **Customer Segmentation Using RFM Analysis :**
  - Based on the score , have mapped the below customer sentiment types and plotted a pie chart for better visualization,

Score	Sentiment Type
5	Champions
4	Loyal Customers
3	Need Attention
2	At Risk
1	About Leaving



- From the plot we can clearly see the rise in positive reviews from 2010. Reaching its peak around 2012 ,Negative and neutral reviews are very low as compared to the positive reviews.

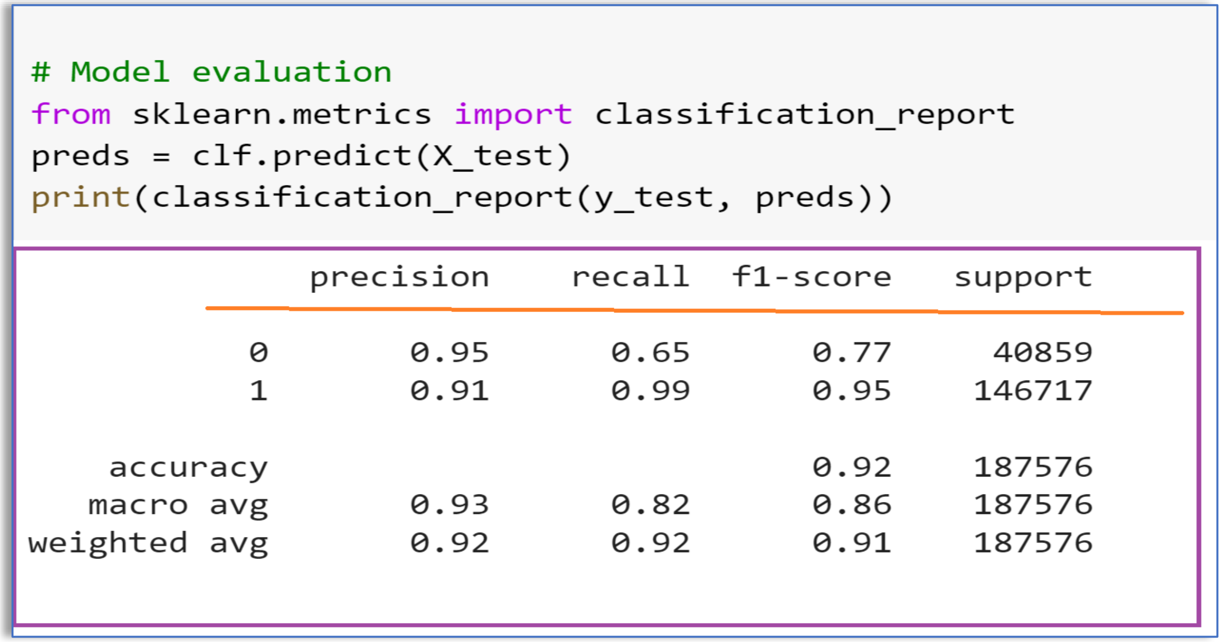
Model Building: Sentiment Analysis

- Well, I went through a lot of steps together and this is the final one! After all the text preparation I have done, it's now time to put it together into a classification model to train an algorithm that understands wherever a text string has a positive or a negative feeling based on the features we extracted from the given dataset.

Model Fit & Selection

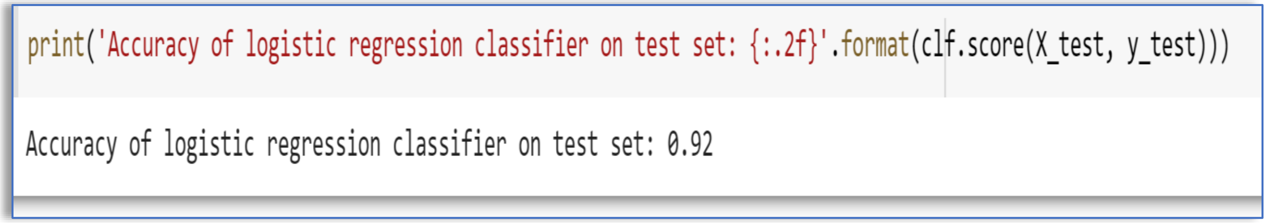
- Have selected RandomForestClassifier model for the given problem statemen and dataset , and
- Due to high volume of data model fit took more time (***46 minutes***) to process.

Model Evaluation



```
# Model evaluation
from sklearn.metrics import classification_report
preds = clf.predict(X_test)
print(classification_report(y_test, preds))
```

```
              precision    recall  f1-score   support

           0       0.95      0.65      0.77     40859
           1       0.91      0.99      0.95    146717

    accuracy                           0.92    187576
   macro avg       0.93      0.82      0.86    187576
weighted avg       0.92      0.92      0.91    187576
```



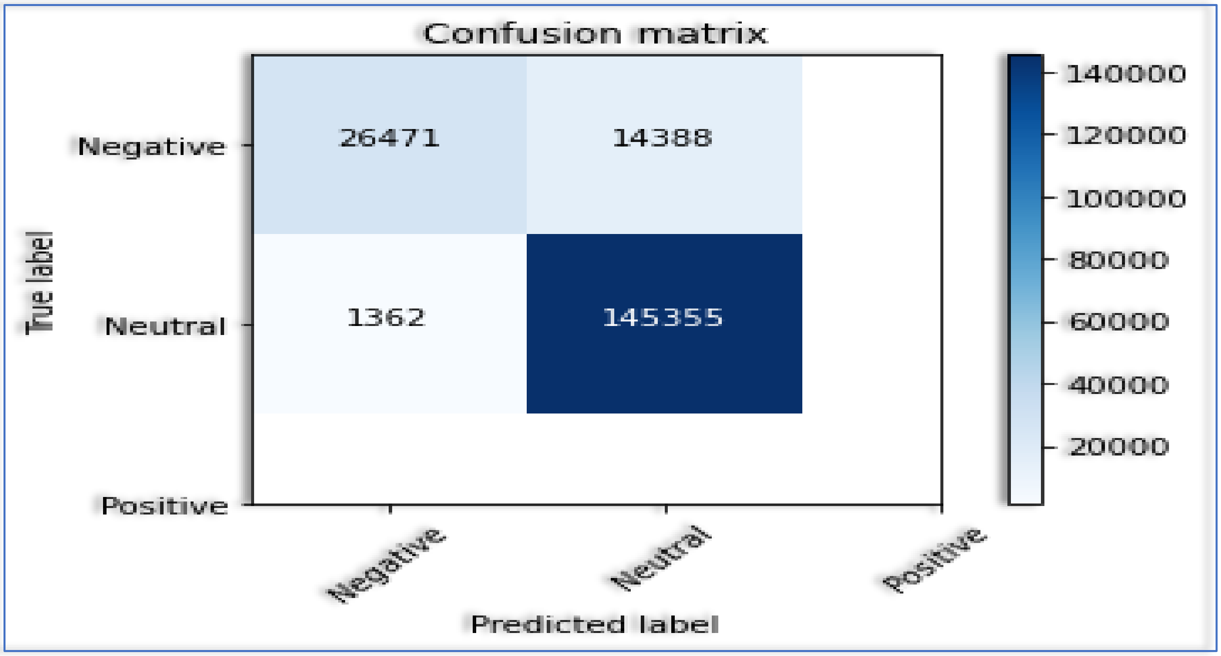
```
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(clf.score(X_test, y_test)))
```

```
Accuracy of logistic regression classifier on test set: 0.92
```

- Accuracy is 92 % accuracy. That aren't bad. But for classification problems we need to get confusion matrix and check f1 score rather than accuracy

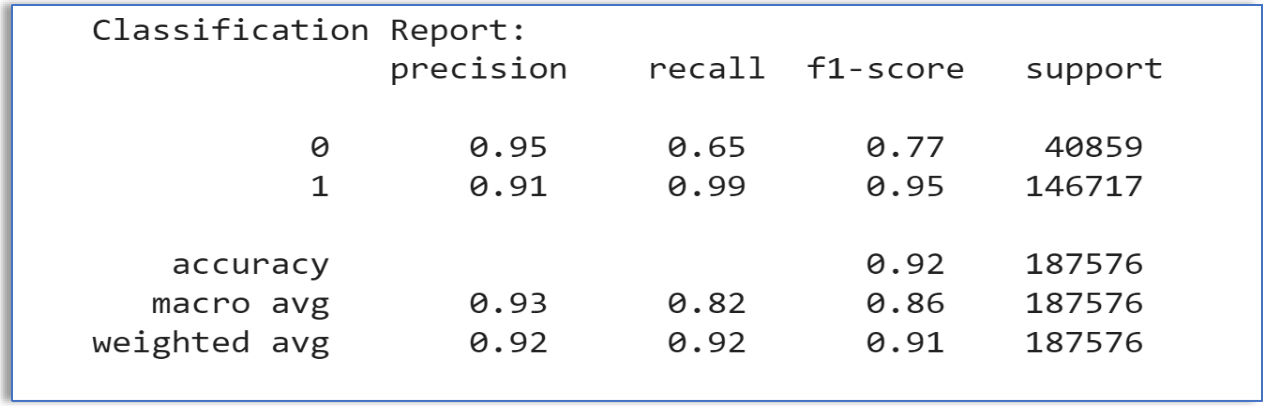
Classification matrix & F1-Score

- Let's plot the confusion matrix with ROC and check our f1 score , and



- From the above chart, the diagonal elements (26471+14388+145355), they are correctly predicted records and rest are incorrectly classified by the algorithm

F1-score



```
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.65      0.77     40859
           1       0.91      0.99      0.95    146717

    accuracy                           0.92    187576
   macro avg       0.93      0.82      0.86    187576
weighted avg       0.92      0.92      0.91    187576
```

- Since predicting both positive, negative and neutral reviews are important , from given dataset got a pretty good f1 score.

Conclusion

- Have done a pretty neat job on classifying all the classes starting from splitting the sentiments based on overall score, text cleaning, customize the stop words list based on requirement and finally built a model w/ 92% accuracy. Here are few insights from the project analysis,
    - Consider welcoming ngram in sentiment analysis as one word can't give is proper results and stop words got to be manually checked as they have negative words. It is advised to avoid using stop words in sentiment analysis,
    - Most of our neutral reviews were actual critic of product from the buyers, so retail store can consider these as feedback and give them to the seller to help them improve their products,
    - In sentiment analysis, we should concentrate on our f1 score where we got an average of 95 % so it was a pretty good job.
    - Well, it was a long journey and I hope you all had experienced a really explained and useful notebook for a Sentimental Analysis task.