Koneru Lakshmaiah Education Foundation

(Deemed to be University)

ENGINEERING DEPARTMENT

A Machine Learning Project Based Report

ON

Sentiment Analysis

SUBMITTED BY:

ID NUMBER	NAME
180030646	Puppala Siva
180030887	Neerukonda Sai Sruthi
180030956	Chekuri Chekitha

UNDER THE GUIDANCE OF

Dr. Sagar Imambi

ASSOCIATE PROFESSOR

KL UNIVERSITY

Green fields, Vaddeswaram – 522 502

Guntur Dt., AP, India.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project-based report entitled "**Sentiment Analysis**" submitted by **Mr. Puppala Siva, Ms. Neerukonda Sai Sruthi, Ms. Chekuri Chekitha** bearing Regd. No. 180030646, 180030887, 180030956 to the **Department of Computer Science and Engineering, K L University** in partial fulfillment of the requirements for the completion "**Machine Learning**" course in B. Tech V Semester, is a bonafied record of the work carried out by her under my supervision during the academic year 2018 – 2019.

Dr. Sagar Imambi
FACULTY INCHARGE
DEPARTMENT OF CSE
K L (Deemed to be University)

Mr. V. HARI KIRAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF CSE
K L (Deemed to be University)

ACKNOWLEDGEMENTS

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project-based report.

I express sincere gratitude to HOD – **Mr. V. Hari Kiran** for his leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **Dr. Sagar Imambi** for his/her novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

ID NUMBER	NAME
180030646	Puppala Siva
180030887	Neerukonda Sai Sruthi
180030956	Chekuri Chekitha

INDEX

ABSTRACT

Sentiment analysis is one of the fastest growing research areas in Computer Science and is very popular among the NLP community, making it challenging. It is a series of methods, techniques and tools about detecting and extracting subjective information, such as opinions and attitudes, from language; a type of classification where the data is classified into different classes. These classes can be binary (positive or negative) or can have multiple classes (happy, sad, angry, and many).

Sentiment analysis is a way to analyze the subjective information in a text and mine the opinion, hence giving it another name – opinion mining. It can be best described as a process by which information is extracted from a writing that has different polarities for the opinions of people in regards to entities, events and their attributes. By polarities, we mean the positive, negative or neutral feelings that are bound to the text. Many topics beyond product reviews like stock markets, elections, disasters, medicine, cyberbullying, etc… extend the utilization of sentiment analysis

INTRODUCTION

Sentiment analysis is the process of using natural language processing, text analysis, and statistics to analyze customer sentiment.

It is often said that the best businesses understand the sentiment of their customers – what people are saying, how they're saying it, and what they mean. Customer sentiment can be found in tweets, comments, reviews, or other places where people mention your brand. The opinions of others have a significant influence in our daily decision – making process. These decisions range from buying a product such as a smart phone to making investments to choosing a school – all decisions that affect various aspects of our daily life.

Before the Internet, people would seek opinions on products and services from sources such as friends, relatives, or consumer reports. However, in the Internet era, it is much easier to collect diverse opinions from different people around the world. People look to review sites (e.g., CNET, Epinions.com), e-commerce sites (e.g., Amazon, eBay), online opinion sites (e.g., TripAdvisor, Rotten Tomatoes, Yelp) and social media (e.g., Facebook, Twitter) to get feedback on how a particular product or service may be perceived in the market.

Similarly, organizations use surveys, opinion polls, and social media as a mechanism to obtain feedback on their products and services. sentiment analysis or opinion mining is the computational study of opinions, sentiments, and emotions expressed in text. The use of sentiment analysis is becoming more widely leveraged because the information it yields can result in the monetization of products and services. For example, by obtaining consumer feedback on a marketing campaign, an organization can measure the campaign's success or learn how to adjust it for greater success. Product feedback is also helpful in building better products, which can have a direct impact on revenue, as well as comparing competitor offerings.

This project aims to perform sentiment classification of online product reviews using various Machine Learning classifiers. This project analyzes sentiment on dataset from document level (review level). Reviews include product and user information, ratings, and

a plaintext review. This project involves the performance of Machine Learning classifier models - Multinomial Naïve Bayes, Logistic Regression, Linear SVC, Random Forest. The best classifier was chosen to standardize the model to classify any product reviews in the future with promising outcomes.

Problem context

Since a long time, opinions have been central to almost all human activities and are key influencers of our behavior and in recent years, humans have grown accustomed to online shopping and purchasing anything that is just a click away leading to a high competition in the industry to urge and analyze the feedback so as to evolve over time and retain their customers with millions of customers, it is near impossible to manually review the customer sentiment and that is where the problem is. Our main objective is to build and identify a suitable ML model for the sentiment analysis of customer reviews using supervised machine learning techniques.

METHODOLOGY

In this project, we worked with the data (i.e., the customer reviews that were collected from amazon); applied a few models to learn the patterns and evaluated them for the optimal model. We experimented with multiple classifiers such as Logistic Regression, SVM (Support Vector Machines), etc… and semantic embeddings such as char n – grams, TF – IDF vectors, Bag of words, etc…

Data description

Data is a precious resource for every organization. But, if we don't analyze that statement further, it can negate itself. Businesses use data for various purposes. On a broad level, it is used to make informed business decisions, execute successful sales and marketing campaigns, etc. But these cannot be implemented with just raw data.

Data becomes a precious resource only if it is cleansed, well-labelled, annotated, and prepared. Once the data goes through various stages of fitness tests it then finally becomes qualified for further processing. This Process contains many steps like Data Extraction, Data Profiling, Data Cleaning, Data Transformation, Data Augmentation, Feature Engineering, etc…

The data we used is set of reviews in CSV file format. Each review contains ProductID, UserID, Name, helpful votes and Total Votes given, Review headline and the actual review. We will be considering the review text and the ratings to predict the sentiment form the text.

Data Analysis

Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

We are visualizing the text using word clouds though as text data is quite different from numerical data and many of the reviews talk about food related things - coffee, flavor, taste,

drink etc… including some positive words such as 'positive' and 'good'. Next, we segregated the ratings into different classes (So, ratings of 1 or 2 will be classified to negative, ratings of 4 or 5 to positive and ratings of 3 to neutral)

Visually representing the content of a text document is one of the most important tasks in the field of text mining. As a data scientist or NLP specialist, not only we explore the content of documents from different aspects and at different levels of details, but also we summarize a single document, show the words and topics, detect events, and create storylines.

However, there are some gaps between visualizing unstructured (text) data and structured data. For example, many text visualizations do not represent the text directly, they represent an output of a language model (word count, character length, word sequences, etc.).

Pre – processing

It is a known fact that text pre - processing and normalization is crucial before building a model and the reviews that were taken have unnecessary characters and common words which won't make the model's performance any better. Such factors can be eliminated by converting all the words into a consistent case format, removing special characters and stop words

Features Extraction (Validation and Optimization)

Coming to feature extraction - as our computers can't work on strings, the review text is converted into a numerical representation of vector form and this is referred to as featurization. There are various featurization techniques available such as Bag-of-words, TFIDF vectorizer and word2vec.

We've used bag - of - words and TFIDF vectorizer in our project to see which performs better.

- Bag-of-words

It is one of the most fundamental methods to transform tokens into a set of features. In this, each word is used as a feature for training the classifier.

So, what we do over here is create a vocabulary of all the unique words from the corpus. Then, we create a matrix of the features by assigning a separate column for each word while each row corresponds to a review. This process is nothing but text vectorization. Now, each entry in the matrix signifies the presence or absence of the word in the review (1 if it is present and 0 if it is not present)

The drawback of this model is that the order of occurrence of words is lost as we create a vector of tokens in the randomized order. This issue is solved using n-grams (as in we consider bigrams instead of unigrams [i.e., individual words]) to preserve the local ordering of words. Another drawback of Bag-of-words strategy is that it does not take the low frequency n grams into account as the higher average count values on words overshadow the shorter ones in many cases (key word being many cases). To reduce this redundancy, we use TFIDF vectorizer which is nothing but term frequency - inverse document frequency.

To implement the bag-of-words strategy, we used Scikit Learn's Count Vectorizer

- TFIDF vectorizer

What tf-idf vectorizer does is it highlights a specific issue that might not be too frequent but is of great importance. It is of two sub - parts:

Term frequency (TF)

Term frequency specifies how frequently a term appears in the review w.r.t the total number of words in the review

Inverse document frequency (IDF)

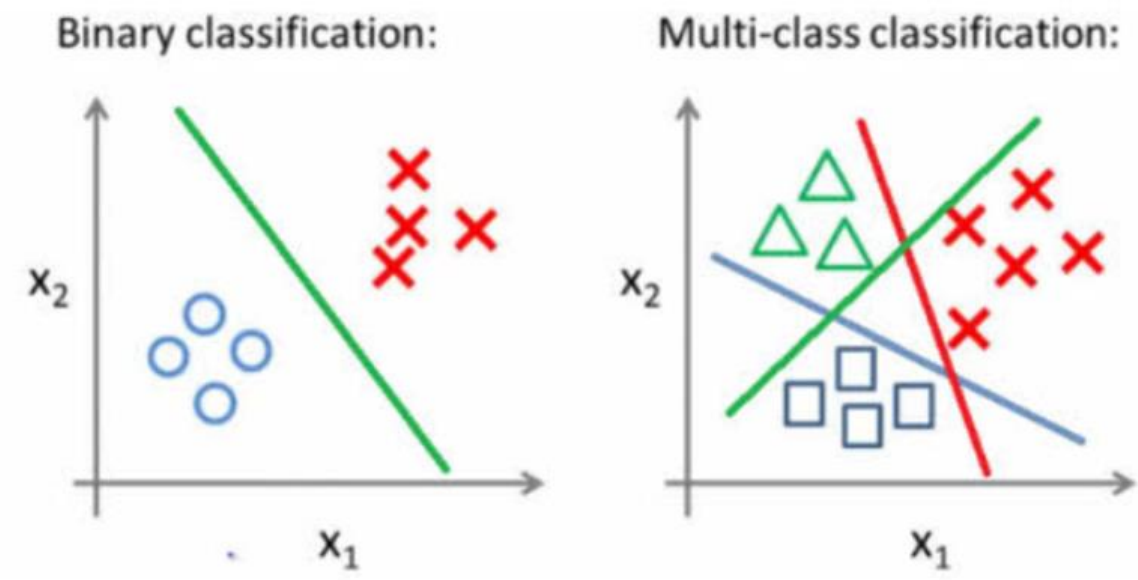
It is a measure of whether a term is rare or frequent across the reviews. A high Tf-idf score is obtained means a term has high frequency in a review but low frequency across all the reviews. For a word that appears in almost all reviews, the tf-idf score is closer to 0.

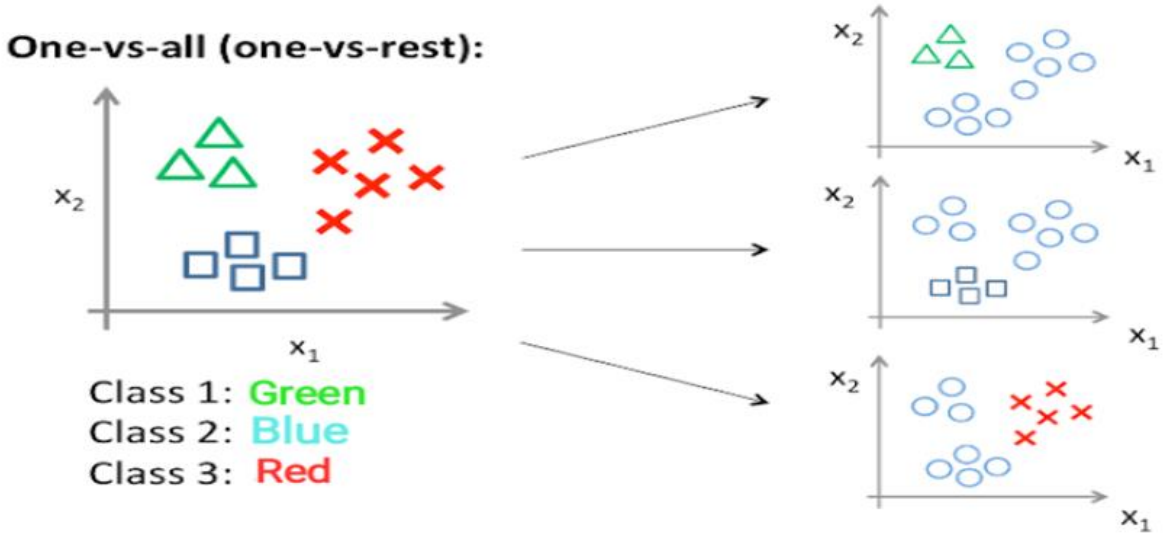
This value increases proportionally to the number of times a word appears in a review and decreases with the number of reviews that contain the word.

Proposed Models

Logistic Regression

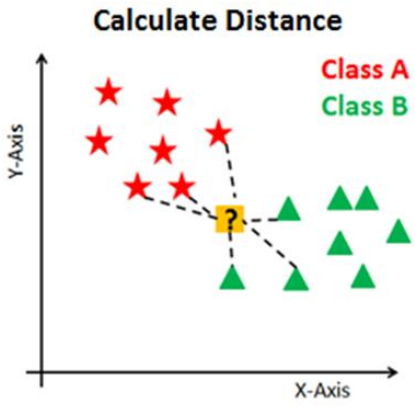
Coming to Logistic Regression, it's often used when the dependent variable is categorical (in our case, the variables are positive, negative and neutral) There are two ways in which a model can be classified. It can be classified using binary classification (where only two class instances - 1 or 0 are present in the dataset) or we can use multi class classification. So, multi class classification allows us to categorize the test data into multiple class labels and can be implemented using two techniques. 1) one vs all 2) One vs one In this problem, we used the one vs all method in which n-binary classifier models are generated for the n-class instances (i.e., the number of class labels present in the dataset and the number of generated binary classifiers must be the same. So, the class that we are working with is denoted by 1 and the rest of the classes become 0

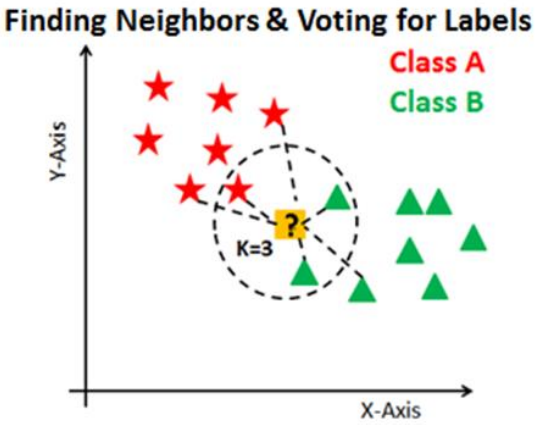




KNN

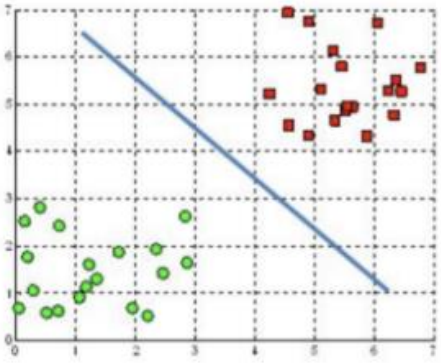
Next, we have K nearest neighbors classifier which assumes that similar things exist in close proximity. The output is calculated as the class with the highest frequency form k – most similar instances. Each instance votes for their class and the class with the most votes is taken as the prediction. The class probabilities can be calculated as the normalized frequency of samples that belong to each class in the set of K most similar instances for a new data instance. It doesn't require training before making predictions, and the new data can be added seamlessly which will not impact the accuracy. Generally, if K is used and there even number of classes, it is advised to choose K value with an odd number to avid a tie and vice-versa

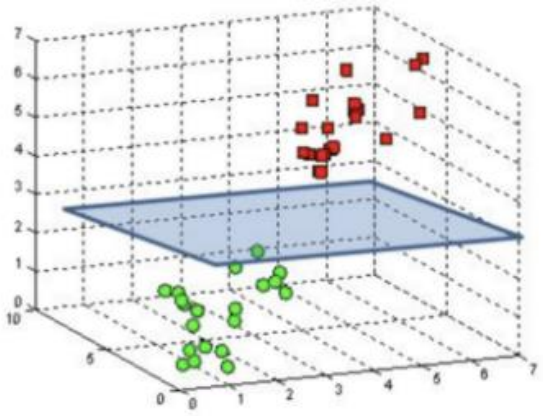




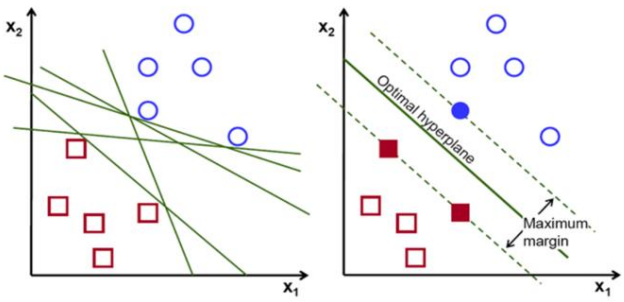
Support Vector Machine

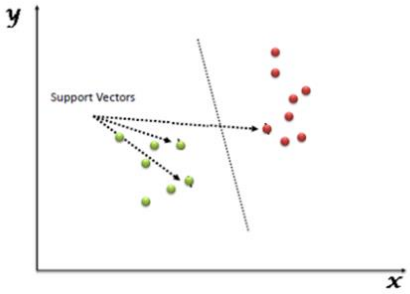
Finally, we used Support vector classifier. The objective of this ML algorithm is to find a hyperplane (which is nothing but a decision boundary – the SVM classifier) in an N-dimensional space that distinctly classifies the data points and has the maximum margin (i.e., the maximum distance between data points of both classes) Here, N is the number of features on which the dimension of the hyperplane depends on.





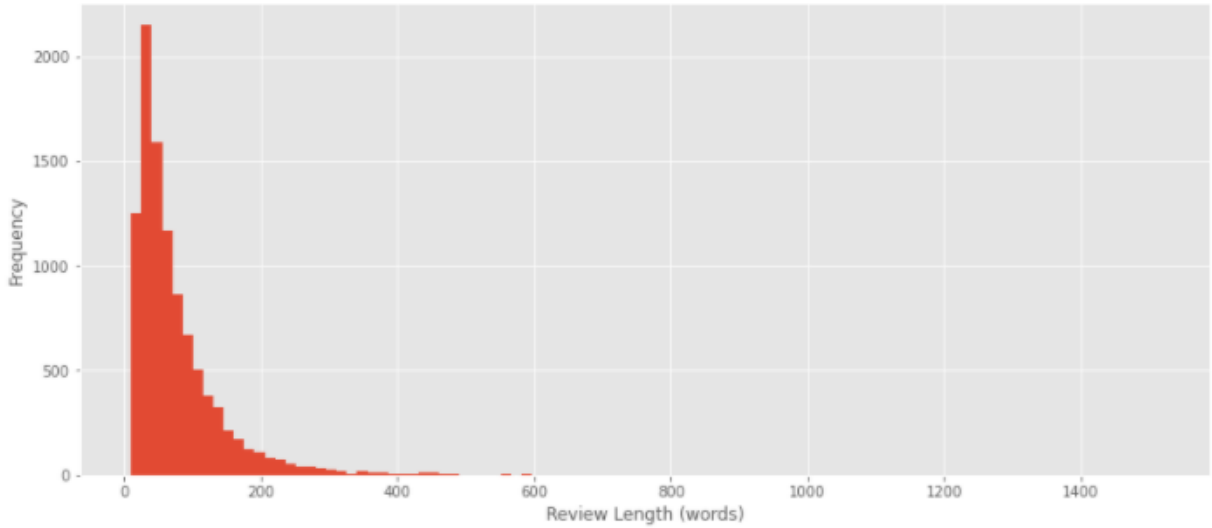
Hence, if the number of input features is 2, the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane Support vectors are data points that help us build our SVM. They are closer to the hyperplane and influence the position and orientation of the hyperplane

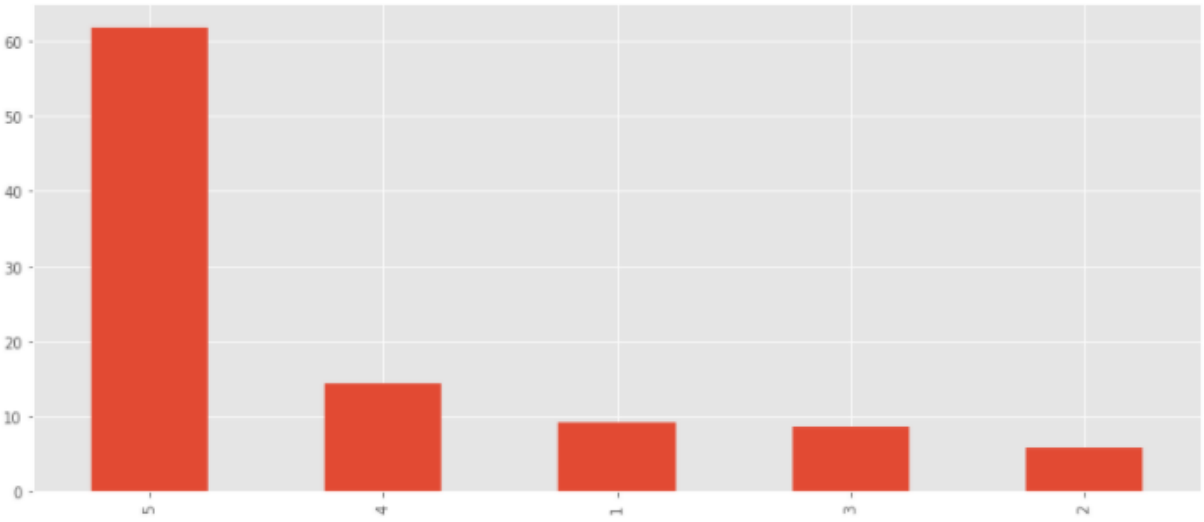




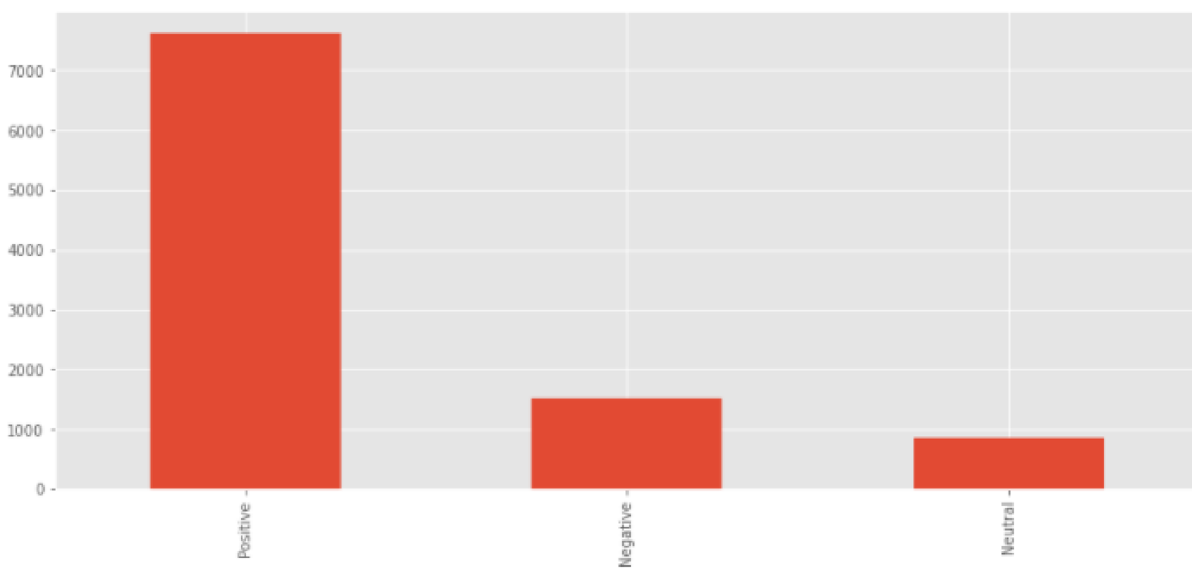
RESULTS AND DISCUSSION

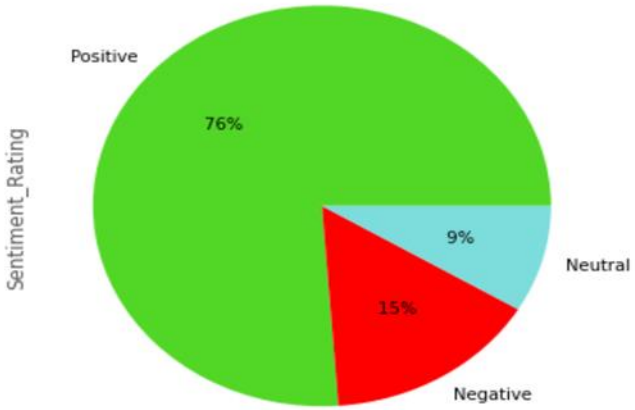
Here is a graphical representation of our data. We are looking at the distribution of ratings and the distribution of number of words per review length which is done by applying lambda function that splits each review by spaces






taste
good
br
coffee
product
flavor
great
make
one
Amazon
love
will
food
find
really
tea
much
little
cup
tried
use
drink
try
buy
best
time
well
now
price
bag
better
chip
give
eat
found
delicious
without
even
made
want
still
sugar
treat
box
add
sweet
first
nice
water
used
know
say
bought
favorite
brand
think
dog
cat
lot
bit
way
two
work
day
mix
order
review
fresh
stuff
ordered
snack
chocolate
ingredient
recommend
calorie
organic
store
need
never
strong
something
come
pack
go
hard
keep
enjoy
healthy
people
anything
purchase
right
many
seem
thing
got
tasty
almost
going
regular
put
feel
may
instead
easy
happy
enough
item
small
milk
look
look
looking
perfect
package
bad
far
texture
available
actually
take
trying
though
purchased
added
less
cookie
family
quite
problem
buying
natural
chicken
bitter
contain
gluten free
quality
case
loved
rather
bean
type
hot
fine
recipe
sure
light
eating
kind
diet
half
using
shipping
others
excellent
expensive
smell
company
another
bottle
year
blend
open
received
different
making
morning
fact
prefer
high
every
around
usually
definitely
especially
friend
ordering
salt
sauce
href
candy
juice
soda
tasted
star
least
store
alway
thought
able
probably


7000
6000
5000
4000
3000
2000
1000
0
Positive
Negative
Neutral




```
- Old Review -
this saltwater taffy had great flavors and was very soft and chewy.  each c
andy was individually wrapped well.  none of the candies were stuck togethe
r, which did happen in the expensive version, fralinger's.  would highly re
commend this candy!  i served it at a beach-themed party and everyone loved
it!

- New Review -
this saltwater taffy had great flavors and was very soft and chewy   each c
andy was individually wrapped well   none of the candies were stuck togethe
r  which did happen in the expensive version  fralinger s   would highly re
commend this candy   i served it at a beach themed party and everyone loved
it
```

	reviews_text_new	reviews_text_nonstop
0	i have bought several of the vitality canned d...	[bought, several, vitality, canned, dog, food,...
1	product arrived labeled as jumbo salted peanut...	[product, arrived, labeled, jumbo, salted, pea...
2	this is a confection that has been around a fe...	[confection, around, centuries, light, pillowy...
3	if you are looking for the secret ingredient i...	[looking, secret, ingredient, robitussin, beli...
4	great taffy at a great price there was a wid...	[great, taffy, great, price, wide, assortment,...
...	...	...
9995	we switched from the advance similac to the or...	[switched, advance, similac, organic, product,...
9996	like the bad reviews say the organic formula ...	[like, bad, reviews, say, organic, formula, co...
9997	i wanted to solely breastfeed but was unable t...	[wanted, solely, breastfeed, unable, keep, sup...
9998	i love the fact that i can get this delieved t...	[love, fact, get, delieved, house, delievy, ch...
9999	we have a 7 week old he had gas and constipat...	[7, week, old, gas, constipation, problems, fi...

10000 rows × 2 columns

```
# Fitting a logistic regression model
lr_model_all = LogisticRegression()
lr_model_all.fit(X_train_bow, y_train_bow)

# Class prediction
test_pred_lr_all = lr_model_all.predict(X_test_bow)

# Calculating key performance metrics
print(classification_report(test_pred_lr_all, y_test_bow))
print('Accuracy: {}'. format(accuracy_score(test_pred_lr_all, y_test_bow)))
```

```
              precision    recall  f1-score   support

    Negative       0.59      0.68      0.63       258
     Neutral       0.36      0.50      0.42       129
    Positive       0.94      0.89      0.91      1613

    accuracy                           0.84      2000
   macro avg       0.63      0.69      0.65      2000
weighted avg       0.86      0.84      0.84      2000

Accuracy: 0.836
```

```
from sklearn.neighbors import KNeighborsClassifier
k=KNeighborsClassifier(n_neighbors=18)
k.fit(X_train_bow, y_train_bow)

# Predicting the results
knn_pred= k.predict(X_test_bow)

# Evaluating the model
print(classification_report(knn_pred, y_test_bow))
print('Accuracy: {}'. format(accuracy_score(knn_pred, y_test_bow)))
```

```
              precision    recall  f1-score   support

    Negative       0.02      0.56      0.03         9
     Neutral       0.01      0.67      0.02         3
    Positive       1.00      0.77      0.87      1988

    accuracy                           0.76      2000
   macro avg       0.34      0.66      0.31      2000
weighted avg       0.99      0.76      0.86      2000

Accuracy: 0.764
```

```
from sklearn.svm import SVC
svm_classifier =SVC()
svm_classifier.fit(X_train_bow,y_train_bow)
svm_pred=svm_classifier.predict(X_test_bow)
print(classification_report(svm_pred, y_test_bow))
print('Accuracy: {}'. format(accuracy_score(svm_pred, y_test_bow)))
```

```
              precision    recall  f1-score   support

    Negative       0.20      0.90      0.33        67
     Neutral       0.02      1.00      0.03         3
    Positive       1.00      0.79      0.88      1930

    accuracy                           0.79      2000
   macro avg       0.41      0.89      0.41      2000
weighted avg       0.97      0.79      0.86      2000

Accuracy: 0.793
```

Model	Accuracy (Bag-of-words)	Accuracy (TFIDF Vectorizer)
Logistic Regression	0.836	0.831
KNN	0.764	0.824
SVC	0.793	0.775

CONCLUSION

In this project, we evaluated the models for the task of reviews detection. Our experiment on Customer Reviews' dataset shows that **Logistic regression model when combined with Bag – of – words** featurization technique significantly outperformed the other models, hence making it the most suitable model for sentiment analysis for the given dataset. However, the accuracy of the models may change w.r.t datasets and featurization techniques (this model may be outperformed by the others)

Future scope

A lot of research is present in literature for detecting sentiment from the text. Still, there is a huge scope of improvement of these existing sentiment analysis models. Existing sentiment analysis models can be improved further with more semantic and commonsense knowledge.

Social media is exploding and the amount of information available is unprecedented. In order to correctly understand the collective sentiment, we need to change the analysis paradigm (I personally believe is wrong). Instead of calculating and attribute a numerical value to sentiments, we will be better off if we assigned a spectrum of values to sentiments. The overall flavor of the documents will change from; Positive, Neutral or Negative, to a more comprehensive and colorful (human like) sentiment output. The tools and methods for this typology of challenges already exist and can be found in the shape of fuzzy logic

REFERENCES

1. Akkaya, C., J. Wiebe, and R. Mihalcea. Subjectivity word sense disambiguation. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-2009), 2009.
2. Bickerstaffe, A. and I. Zukerman. A hierarchical classifier applied to multi-way sentiment detection. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), 2010
3. Alm, C.O. Subjective natural language problems: motivations, applications, characterizations, and implications. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers (ACL-2011), 2011.
4. Andreevskaia, A. and S. Bergler. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), 2006.
5. Archak, N., A. Ghose, and P.G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-2007), 2007.
6. Aue, A. and M. Gamon. Customizing sentiment classifiers to new domains: a case study. In Proceedings of Recent Advances in Natural Language Processing (RANLP-2005), 2005.