SENTIMENT ANALYSIS FOR PRODUCT RATING

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my professor, Dr. Bhaskar Karn, for giving me this wonderful opportunity to research on this interesting topic.

I would also like to thank my family and friends, who have been a huge support and encouragement throughout the research. I am also grateful to all the developers and AI researchers and the abundance of online resources that has helped me conduct my research thoroughly

ABSTRACT

Recent developments in research in Artificial Intelligence and Machine Learning have had massive impact on major developments around the world. These have helped create technologies that have helped companies and service providers evaluate the performance of their products and services, and one major way of doing that is by analysing what customers have to say through reviews. A lot can be understood about how satisfied customers are with a certain type of product/service by proper evaluation of their reviews, and this is where Sentiment Analysis comes into play. Sentiment analysis is the process where a piece of text is processed to determine the kind of sentiment it propagates – more specifically, classifying it into three emotions: positive, neutral or negative. Sentiment Analysis is a method to automate the process that a person would have to go through to understand the general response of the customers about a certain product or service.

KEYWORDS

Sentiment Analysis, Product Rating, TextBlob, VADER, Logistic Regression, SVM

INTRODUCTION

With the widespread access to internet and a plethora of websites, it has become increasingly easier for customers to voice their opinions about a products and services, giving rise to large reserves of data that can be processed to understand the underlying emotion behind each text piece. Many social media platforms such as Facebook, Instagram and Twitter have proven to be important mediums for customers to review products. Apart from that, every E-commerce website and service provider have a review section where millions of comments are put up by customers. For the purpose of this research, textual reviews will be processed using machine learning algorithms to classify them as positive, negative or neutral, and this polarity will be plotted to give a sense of how semantic analysis can help businesses in today's world. The overall work flow has been depicted below:

1. Get text data to train a machine learning algo.
2. Train and predict the data using different techniques
3. Compare the performance of different models

There are numerous ways to perform sentiment analysis, and the best approach will be determined through this research.

The following techniques will be used for sentiment analysis:

1. Rules based Approach (TextBlob and VADER)
2. Feature Extraction (Logistic regression, Support vector Machine)

This research compares all these methods based on accuracy and the macro F1 score, in order to determine the best approach to perform Sentiment Analysis.

RELATED WORKS

In this 21st century, people are more social in social media, internet, online shopping etc. Thus directly or indirectly online judgments, opinions are eventually gaining great attention. But

the real deal is analysis or mining of opinions. Below is the review of some existing solutions available for SA.

OPINE, an unsupervised, web-based information extraction system proposed by Propescu extracted product feature and opinions from reviews. It identifies product feature, opinion regarding product feature, determines polarity of opinions and then ranks product accordingly. In feature identification, nouns from dataset or reviews are extracted. Frequencies higher than the threshold frequency are kept else discarded. OPINE's feature assessor is used to extract explicit features (occurrence of frequent features). Researchers have used manual extraction rule to extract data. Advancement of OPINE is its domain independency. But fails to find its real life uses as OPINE system is not easily available.

Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone, is a linguistic approach of sentiment analysis at document level, proposed by Benamara et al. [8] in the year 2006. This research work began with measuring the intensity of degree of adverbs (using Linguistic Classifiers) and adverb-adjective combinations (using Scoring Methods). Variable Priority Scoring, Adjective Priority Scoring and Adverb First Scoring are the said Scoring methods used herein. The goal of all these methods are nothing but to add a relative weight (in a variable, on a scale of 0 to 1) of score of adverb relative to the score of adjective. This paper aim to determine which weight most losely matches human assignments of opinions. Experimenting on about 200 documents of news resources it shows that analysis that best matches the human sentiments must comprise of 35% of adverbs along with adjectives. Produces Pearson correlation (correlation between human sentiment and Sentiment Analysis Algorithms) and of about 0.47 (ranging in between -1 and 1) [8]. Though this approach shows higher Pearson correlation but considered very few dataset. One of the solutions to Sentiment Analysis namely Opinion Digger was introduced by Moghaddam and Ester [1]. This unsupervised Machine Learning methodology works at Sentence level. Correlates and compares product aspect and standard rating guidelines (used in Amazon, Snapdeal, flipkart1 etc). This proposed work is divided into two sub methods. At first, input information is fragmented into sentences. Repeated nouns in the sentences are coined as aspects. Aspect (repeated nouns) if forms any pattern, are stored. Secondly, aspects are compared to the rating guideline (like 4 means "Good", 3 means "Average", etc) and accordingly labelled as "Good" ,"Average" and "Bad" [1]. Major advantage is its high performance in product rating at aspect level with a loss of 0.49 only. Demanding guidelines and known data to rate are its major drawbacks and it was compared with very few methodologies. Therefore lacks more number of performance comparisons.

PRINCIPLE OF WORKING

Acquiring the Dataset:

The first step was acquiring the dataset to feed into the training models. This data set was downloaded from https://nijianmo.github.io/amazon/index.html. The dataset has 150,0000 samples. Each sample has two categories: the textual review that will be input for training the model, and ratings that will be the output to which the model will map the input. They are stored in pandas dataframe as shown below:

asin	review	rating
B0050SW93S	I don't usually comment about other reviews bu...	2.0
B005WWZUQ0	Firstly I would like to mention that giving th...	2.0
B017L187YG	The game is great itself and would have 5 star...	2.0
B000MMLNNY	Typically, I'm not a huge fan of movie-themed ...	3.0
B00J5WK5R2	I'm not completely sure how I feel about this ...	4.0
...	...	...
B0011Z72TK	This crappy song isn't even worthy of getting ...	1.0
B000BYQJCI	I found a number of problems in this first unp...	2.0
B006OITIWS	i bought this song after hearing it on the rad...	3.0
B0000657SP	the games controls are the worst and this game...	2.0
B00DBRM3G8	I was excited to purchase the game, based upon...	1.0

The workflow employed for model training and evaluation is:

Model Training → Prediction → Evaluation → Explanation

Model Training:

Each classifier(except for the rule-based ones) is trained on the 150,000 samples of Amazon review data using a supervised learning algorithm. After prediction, the accuracy and the F1 scores will be calculated, and the model will be evaluated using a confusion matrix. The confusion matrix will be plotted using scikit-learn and matplotlib. It tabulates the number of correct predictions versus the number of incorrect predictions for each class, so it becomes easier to see which classes were the least accurately predicted for each class. Ideally, the classifier would get 100% of its predictions correct, which means that all elements outside the diagonal would be as close to zero as possible.

I. RULES-BASED APPROACH

1. TextBlob:

TextBlob is a Python library widely used for processing textual data to evaluate the underlying sentiment. It is built on top of NLTK – another Natural Language Processing tootlbox. TextBlob uses a sentiment lexicon (consisting of predefined words) to assign scores for each word, which are then averaged out using a weighted average to give an overall sentence sentiment score. Three scores: “polarity”, “subjectivity” and “intensity” are calculated for each word. The polarity scores were calculated for each statement in the dataset to determine the sentiment, and these scores were then binned to give a rating out of 5.

```
10  from textblob import TextBlob
11  pol = lambda x:TextBlob(x).sentiment.polarity
12  data['polarity'] = data['review'].apply(pol)
13  data['pred'] = pd.cut(data['polarity'], bins=5, labels=[1,2,3,4,5])
```

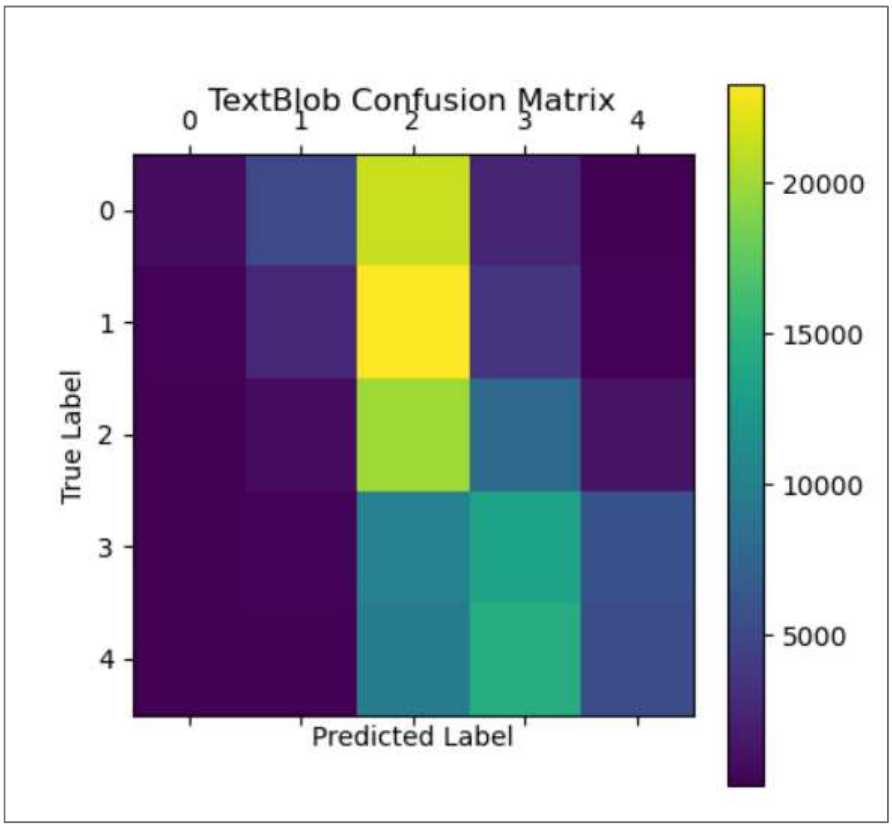
The following snippet shows the dataframe containing polarity of reviews, and the next one shows the dataframe after the polarities were binned to give a rating out of 5.

asin	review	rating	polarity
B0050SW93S	I don't usually comment about other reviews bu...	2.0	0.018931
B005WWZUQ0	Firstly I would like to mention that giving th...	2.0	-0.038783
B017L187YG	The game is great itself and would have 5 star...	2.0	0.171429
B000MMLNNY	Typically, I'm not a huge fan of movie-themed ...	3.0	0.102576
B00J5WK5R2	I'm not completely sure how I feel about this ...	4.0	0.356299
B005UWEKZE	I loved this song while watching the movie. Th...	4.0	0.157143
B00020LZAW	This game I thought would be good but it is ve...	1.0	-0.210000
B000W16ZEM	Jefferey Osbourne made a name for himself sing...	5.0	0.334921
B005AGO4LU	Best shoes ever!!!	5.0	1.000000
B006ONY954	I have a phonograph record of this album in my...	4.0	-0.032143

asin	review	rating	pred
B0050SW93S	I don't usually comment about other reviews bu...	2.0	3
B005WWZUQ0	Firstly I would like to mention that giving th...	2.0	3
B017L187YG	The game is great itself and would have 5 star...	2.0	3
B000MMLNNY	Typically, I'm not a huge fan of movie-themed ...	3.0	3
B00J5WK5R2	I'm not completely sure how I feel about this ...	4.0	4
B005UWEKZE	I loved this song while watching the movie. Th...	4.0	3
B00020LZAW	This game I thought would be good but it is ve...	1.0	2
B000W16ZEM	Jefferey Osbourne made a name for himself sing...	5.0	4
B005AGO4LU	Best shoes ever!!!	5.0	5
B006ONY954	I have a phonograph record of this album in my...	4.0	3

Accuracy = 28.26%

F1 Score = 23.29



Cell[0,0] shows the number of samples that were rated 1 and predicted 1 by the classifier. Similarly, Cell[1,1] shows the number of samples that were rated 2 and predicted 2 by the classifier. In Cell[1,2] of the confusion matrix, we can see that over 20,000 samples that were rated 2 were incorrectly predicted as 3 by TextBlob.

It is clear that our TextBlob classifier predicts most samples as neutral or mildly positive, i.e. of class 3 or 4, which explains why the model accuracy is so low. Very few predictions are strongly negative or positive — this makes sense because TextBlob uses a weighted average sentiment score over all the words in each sample. This can very easily diffuse out the effect of sentences with widely varying polarities between words.

2. VADER(Valence Aware Dictionary sEntiment Reasoner):

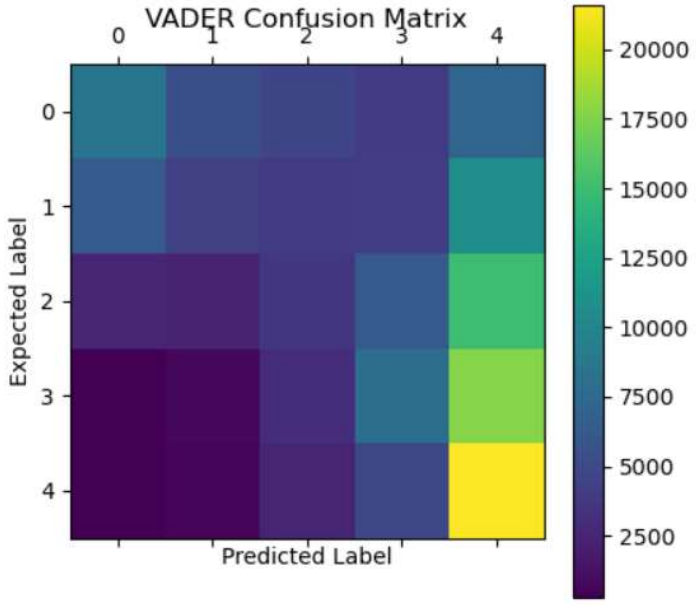
VADER is another widely used rule-based library for sentiment analysis. Similar to TextBlob, it uses a sentiment lexicon that contains intensity measures for each word based on human-annotated labels. However, what makes it different is that VADER **was designed with a focus on social media texts**. Thus, it puts a lot of emphasis on rules that capture the essence of text typically seen on social media — for example, short sentences with emojis, repetitive vocabulary and copious use of punctuation (such as exclamation marks). VADER calculates a compound score between -1 and 1, which a result of normalization of the positive, negative and neutral scores it gives to the text. These compound scores are then binned to give a rating out of 5.

```
10   vader_obj = SentimentIntensityAnalyzer()
11   compound = lambda x:vader_obj.polarity_scores(x)['compound']
12   data['pred'] = pd.cut(data['compound'], bins=5, labels=[1,2,3,4,5])
13   data['compound'] = data['review'].apply(compound)
```

The following snippet shows the dataframe containing polarity of reviews, and the next one shows the dataframe after the polarities were binned to give a rating out of 5.

asin	review	rating	pred
B0050SW93S	I don't usually comment about other reviews bu...	2.0	5
B005WWZUQ0	Firstly I would like to mention that giving th...	2.0	1
B017L187YG	The game is great itself and would have 5 star...	2.0	4
B000MMLNNY	Typically, I'm not a huge fan of movie-themed ...	3.0	5
B00J5WK5R2	I'm not completely sure how I feel about this ...	4.0	5
B005UWEKZE	I loved this song while watching the movie. Th...	4.0	5
B00020LZAW	This game I thought would be good but it is ve...	1.0	1
B000W16ZEM	Jefferey Osbourne made a name for himself sing...	5.0	5
B005AGO4LU	Best shoes ever!!!	5.0	5
B006ONY954	I have a phonograph record of this album in my...	4.0	5

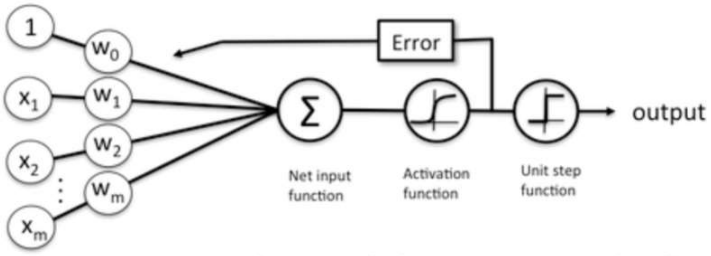
Accuracy = 30.84% F1 Score = 28.19



II. FEATURE EXTRACTION

1. Logistic Regression:

Logistic regression is the most commonly used supervised learning algorithms for classification. This linear model is trained on labelled data, using only linear combinations of inputs and parameters to produce a class prediction. The input features and their weights are fed into an activation function (a sigmoid for binary classification, or a softmax for multi-class). The output of the classifier is just the index of the sigmoid/softmax vector with the highest value as the class label.



Schematic of a logistic regression classifier.

The dataset was divided into training set and test set of

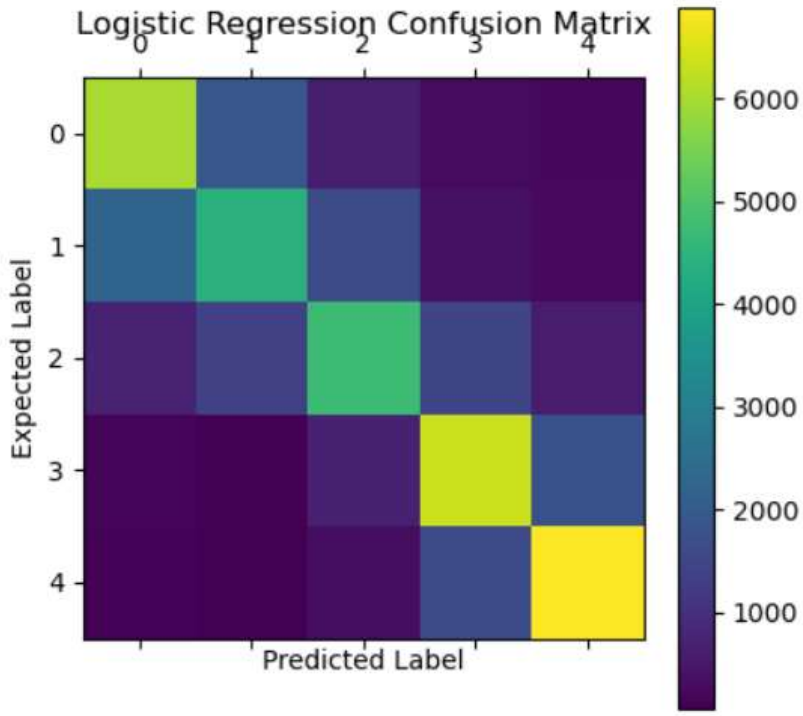
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
lr_object = Pipeline(
    [
        ('vect', CountVectorizer()),
        ('tfidf', TfidfTransformer()),
        ('clf', LogisticRegression(solver='liblinear', multi_class='auto')),
    ]
)
#Spliiting dataset into training and test sets in 70:30 ratio
data_train = data[:105000]
data_test = data[105000:]

#Train the dataset
learner = lr_object.fit(data_train['review'], data_train['rating'])
data_test['pred'] = learner.predict(data_test['review'])
```

The text is transformed into features using scikit-learn's CountVectorizer, which coverts the entire corpus(the reviews) of the training data into a matrix of token counts. Tokens (words, punctuation symbols, etc.) are created using NLTK's tokenizer and commonly-used stop words like "a", "an", "the" are removed because they do not add much value to the sentiment scoring. Next, the count matrix is converted to a TF-IDF (Term-frequency Inverse document frequency) representation.

Once we obtain the TF-IDF representation of the training corpus, the classifier is trained by fitting it to the existing features. A sentiment label is returned for each test sample (using scikit-learn's learner.predict method) as the index of the maximum class probability in the softmax output vector.

asin	review	rating	pred
B001BWRBA8	Didn't quite play it, I'm sure it could turn o...	2.0	2.0
B0092UF54A	Perfect fit, very comfortable, and a great col...	5.0	5.0
B0051S9912	I really liked that song when it came out. Nic...	4.0	4.0
B00T0I6J4O	I've used a variety of video editing software ...	4.0	4.0
B0002RQ3H0	I only hope this review is written quickly eno...	2.0	2.0
...	...	...	...
B0011Z72TK	This crappy song isn't even worthy of getting ...	1.0	4.0
B000BYQJCI	I found a number of problems in this first unp...	2.0	2.0
B006OITIWS	i bought this song after hearing it on the rad...	3.0	4.0
B0000657SP	the games controls are the worst and this game...	2.0	1.0
B00DBRM3G8	I was excited to purchase the game, based upon...	1.0	1.0

Accuracy = 63.28% F1 Score = 62.94



The feature-based method shows significant improvement in the classification accuracy and F1 scores, and mostly predicts correctly (along the diagonal). Logistic Regression works very well on our dataset compared to the Rule-based methods.

2. Support Vector Machine(SVM):

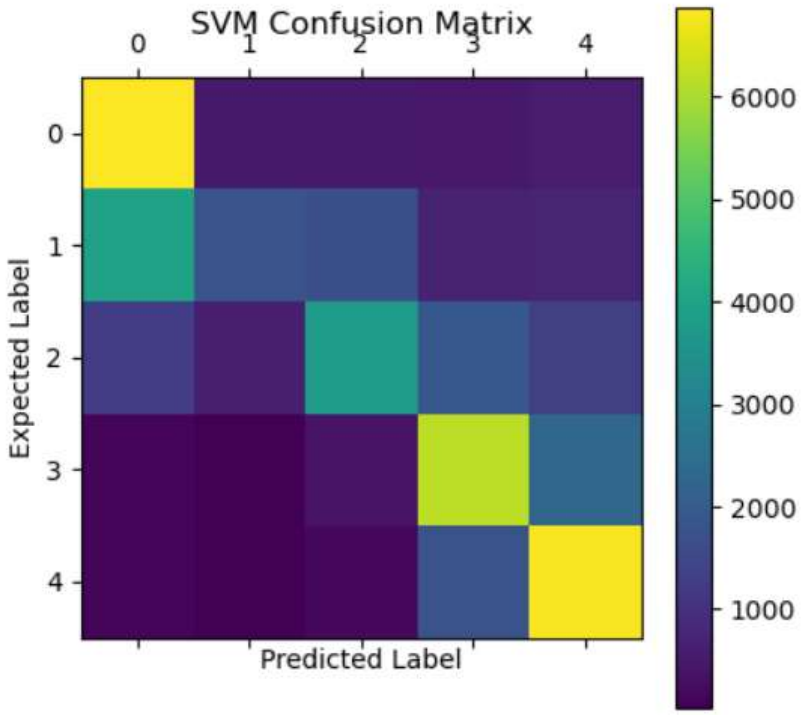
Support Vector Machines are similar to logistic regression with respect to how the loss function is optimized to generate a decision boundary between data points. However, SVMs use "kernel fuctions" – functions that transform a complex, nonlinear decision space to one that has higher dimensionality, so that an appropriate hyperplane separating the data points can be found. SVM classifier looks to maximize the distance of each data point from this hyperplane using "support vectors" that characterize each distance as a vector. A key feature of SVMs is the fact that it uses a *hinge* loss rather than a logistic loss. This makes it more robust to outliers in the data, since the hinge loss does not diverge as quickly as a logistic loss.

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.linear_model import SGDClassifier
from sklearn.pipeline import Pipeline
svm_object = Pipeline(
    [
        ('vect', CountVectorizer()),
        ('tfidf', TfidfTransformer()),
        ('clf', SGDClassifier(
            loss='hinge',
            penalty='l2',
            alpha=1e-3,
            random_state=42,
            max_iter=100,
            learning_rate='optimal',
            tol=None,
        )),
    ]
)

#Splitting data into training and test sets
data_train = data[:105000]
data_test = data[105000:]

#Train the model
learner = svm_object.fit(data_train['review'], data_train['rating'])

# Predict class labels using the learner and output DataFrame
data_test['pred'] = learner.predict(data_test['review'])
```

The linear SVM in scikit-learn is set up using a similar pipeline as done for the logistic regression described in earlier. Once we obtain the TF-IDF representation of the training corpus, we train the SVM model by fitting it to the training data features. A hinge loss function with a stochastic gradient descent (SGD) optimizer is used, and L2 regularization is applied during training. The sentiment label is returned (using scikit-learn's learner.predict method) as the index of the maximum class probability in the softmax output vector.

asin	review	rating	pred
B001BWRBA8	Didn't quite play it, I'm sure it could turn o...	2.0	3.0
B0092UF54A	Perfect fit, very comfortable, and a great col...	5.0	5.0
B0051S9912	I really liked that song when it came out. Nic...	4.0	4.0
B00T0I6J4O	I've used a variety of video editing software ...	4.0	5.0
B0002RQ3H0	I only hope this review is written quickly eno...	2.0	1.0
...	...	...	...
B0011Z72TK	This crappy song isn't even worthy of getting ...	1.0	4.0
B000BYQJCI	I found a number of problems in this first unp...	2.0	1.0
B006OITIWS	i bought this song after hearing it on the rad...	3.0	4.0
B0000657SP	the games controls are the worst and this game...	2.0	1.0
B00DBRM3G8	I was excited to purchase the game, based upon...	1.0	1.0

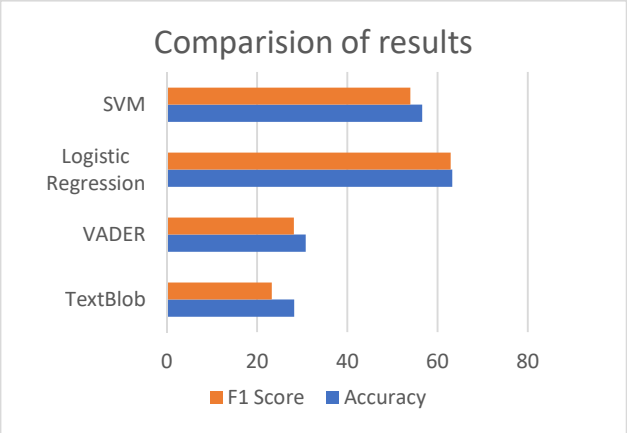
Accuracy = 56.62% F1 Score = 54.01



Looking at the Confusion Matrix, it can be seen that the SVM model predicts the strongly negative/positive classes (1 and 5) more accurately than logistic regression.

RESULT AND DISCUSSION

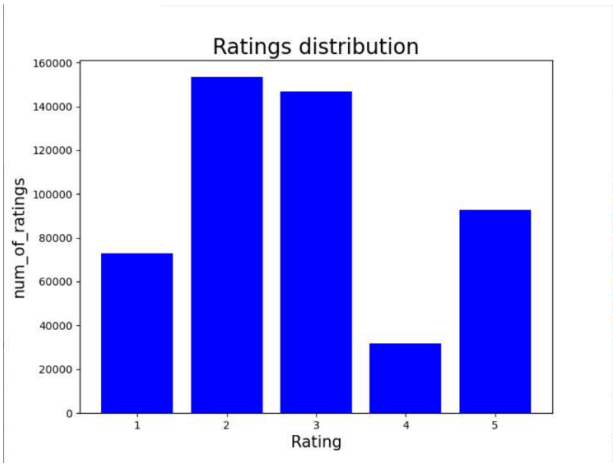
Classifier	Accuracy	F1 Score
TextBlob	28.26	23.29
VADER	30.84	28.19
Logistic Regression	63.28	62.94
SVM	56.62	54.01



150,000 data samples of Amazon review texts were classified into five ratings based on the sentiment in the review. It can be clearly evaluated that compared to Rule – based methods, feature extraction approach works noticeably better on our dataset. Accuracy and F1 scores provide a solid enough basis for comparing the performance of different NLP classifiers in Python. The confusion matrices provided further insight into understanding how well the classifier makes true predictions.

An example of how sentiment analysis can be used:

The reviews of customers purchasing only video games was input into the logistic regression classifier to predict ratings of each review as a measure of how happy they were with the product or Amazon's services regarding the video games category.



From this graph, it can be seen that most of the customers are either neutral or slightly dissatisfied about the video games sold on Amazon. This allows Amazon to give direction and try to understand

why customers are not happy with the video games being sold on its platform. Sentiment Analysis methods can be modified to give ratings based on customer satisfaction on each quality of the product – such as price, durabilty, asthetics, etc. These would give companies an edge in improving products to meet with customer expectations.

CONCLUSION

Sentiment analysis for product review can be utilized by service providers and product manufacturers to improve the quality of their products and services, and meet the expectations of majority of the market. Analyzing text in reviews, social media posts, and online blogs allows companies to have a better understanding of where their product por service is lacking, giving them insights into what changes can be made to satisfy customers.

REFERENCES

[1]Python (programming language) - https://en.wikipedia.org/wiki/Python_(programming_language)

[2]"Stanford Core NLP Toolkit" Available: http://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf

[3]Weishu Hu, Zhiguo Gong, Jingzhi Guo, "Mining Product Features from Online Reviews", IEEE International Conference on E-Business Engineering

[4]Isa Maks, Piek Vossen, A lexicon model for deep sentiment analysis and opinion mining applications

[5]Yulan He, Deyu Zhou, Self-training from labeled features for sentiment analysis