**ABSTRACT**

The consumer reviews serve as feedback for businesses in terms of performance, product quality, and consumer service. Human nature is generally structured to make decisions based on analysing and getting the benefit of other consumer experience and opinions because others often have a great influence on our beliefs, behaviours, perception of reality, and the choices we make Purchasing a product is an interaction between two entities, consumers and business owners. Consumers often use reviews to make decisions about what products to buy, while businesses, on the other hand, not only want to sell their products but also want to receive feedback in terms of consumer reviews. In this study, we examine the results of applying Term Frequency Inverse Document Frequency (TF-IDF) to determine what words in a corpus of documents might be more favorable to use in a query. As the term implies, TF-IDF calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. Words with high TF-IDF numbers imply a strong relationship with the document they appear in, suggesting that if that word were to appear in a query, the document could be of interest to the user. We study the performance of different deep learning algorithms, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Embedding matrix Models, we evaluate our model’s using accuracy. Our experiments revealed that convolutional neural network with feature extraction technique provide the best results for both the unbalanced and balanced versions of the dataset.

.

**TABLE OF CONTENTS**

**ACKNOWLEDGMENT**  **ABSTRACT** 1

1. INTRODUCTION 5

1.1 OVERVIEW 8

1.2 PROBLEM STATEMENT 10

1.3 OBJECTIVE 10

1.5 ORGANIZATION OF REPORT 10

2. LITERATURE SURVEY 11

3. REQUIREMENT SPECIFICATION 17

3.1 Functional requirements 17

3.2 System Requirements 17

3.2.1 Software Requirements 18

3.2.2 Hardware Requirements 18

4.METHODOLOGY 18

4.1 Proposed Model 18

4.2 Dataset 20

4.3 Helpfulness feature 21

4.4 Handling imbalanced Classification 25

4.5 Feature Extracting 26

4.6 Product popularity (product feature) 27

4.7 Mean score from top\_500 product 27

4.8 Histogram of mean score from top\_500\_products 28

4.9 Helpfulness with top\_500 products 28

4.10 Age to helpfulness 29

4.11 Summary and review for TF-IDF 29

4.12 distributed right skewed 30

4.13 Helpfulness Ratio 30

4.14 Reviews and summary overview 31

4.15 Summary and review for TF-IDF 21

4.16 score of binary classification 33

5.IMPLEMENTATION 34

5.1 Implementation overview 34

5.2 Implementation steps 35

5.3 Amazon review Dataset 36

5.4 Importing libraries 37

5.5 Reading the text 38

5.6 text pre-processing 39

5.8 Padding sequences 39

5.9 convolution neural networks 40

5.10 Recurrent neural networks 43

5.11 Embedding Matrix 46

6.RESULTS 49

6.1 CNN results 49

6.2 RNN results 49

6.3 Embedding Matrix 50

6.4 Result Analysis 51

7.CONCLUSION 52

**LIST OF FIGURES**

1. Fig 1.1 Overview 8
2. Fig 4.1 Proposed model 19
3. Fig 4.2 Dataset 20
4. Fig 4.3.1 Helpfulness Features 21
5. Fig 4.3.2: Review Lengths (summary feature) 22
6. Fig 4.3.3: Sentiment Score (Text, Summary Features**)** 22
7. Fig 4.3.4 Reviews Over Time (Time Feature) 23
8. Fig 4.3.6: Number of reviews over time 23
9. Fig 4.3.7 Number of scores over count 24
10. Fig 4.3.8: Sentiment Distribution 24
11. Fig 4.3.9: Summary Sentiment 25
12. Fig 4.4 Feature extraction 26
13. Fig 4.6: Product Popularity (Productid feature) 27

14.Fig 4.7 Mean score from top\_500 product 27

1. Fig 4.8 Histogram of mean score from top\_500\_products 28
2. Fig 4.9 Helpfulness with top\_500 products 28
3. Fig 4.10 Age to helpfulness 29
4. Fig 4.11 Summary and review for TF-IDF 29
5. Fig 4.12 distributed right skewed 30
6. Fig 4.13 Helpfulness Ratio 30
7. Fig 4.14 Reviews and summary overview 31
8. Fig 4.15 Summary and review for TF-IDF 32
9. Fig 4.16 score of binary classification 33
10. Fig 5.1 Implementation overview 34
11. Fig 5.2 Implementation steps 35
12. Fig 5.3 Amazon review Dataset 36
13. Fig 5.4 Importing library 37
14. Fig 5.5 Reading the text 38
15. Fig 5.6 text pre-processing 39
16. Fig 5.8 Padding sequences 39
17. Fig 5.9 convolution neural networks 40
18. Fig 5.10 Recurrent neural networks 43
19. Fig 5.11 Embedding Matrix 46
20. Fig 6.1 CNN results 49
21. Fig 6.2 RNN results 49
22. Fig 6.3 Embedding Matrix 50

## 

## CHAPTER 1

## INTRODUCTION

Purchasing a product is an interaction between two entities, consumers and business owners. Consumers often use reviews to create decisions about what products to shop for, while businesses, on the opposite hand, not only want to sell their products but also want to receive feedback in terms of consumer reviews. Consumers reviews about purchased products shared on the net have great impact. attribute is mostly structured to form decisions supported analyzing and getting the good thing about other consumer experience and opinions because others often have a good influence on our beliefs, behaviors, perception of reality. this fact applies not only to consumers but also to organizations and institutions. within the previous couple of years, consumer ways of expressing their opinions and feelings have changed per changes in social networks, virtual communities and other social media communities. Discovering large amounts of information from unstructured data on the net has become a very important challenge because of its importance in numerous areas of life. to permit better information extraction from the plethora of knowledge available, sentiment analysis has emerged to be able to predict the polarity (positive, negative, neutral) of consumer opinion. This successively would help consumers to raised analyses the textual data providing useful information. during this study, we examine the results of applying Term Frequency Inverse Document Frequency (TF-IDF) to work out what words during a corpus of documents may be more favorable to use during a query. because the term implies, TF-IDF calculates values for every word in a very document through an inverse proportion of the frequency of the word during a particular document to the proportion of documents the word appears in. Words with high TF-IDF numbers imply a robust relationship with the document they seem in, suggesting that if that word were to seem in a very query, the document might be of interest to the user. We study the performance of various deep learning algorithms, like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Embedding matrix Models. we evaluate our model’s using accuracy. Our experiments revealed that a convolutional neural network with a feature extraction technique provides the most effective results for both the unbalanced and balanced versions of the dataset.

* 1. **OVERVIEW**

Amazon review sentiment analysis we will do some text processing and then try out three fairly unoptimized deep learning models The model in general can be given as:

DATASET

TEXT PRE-PROCESSING

FEATURE EXTRATIONS

CNN

RNN

EMBEDDING MATRIX

RESULTS

Fig1,1: overview

Dataset:

The dataset contains several million reviews of Amazon products, with the reviews separated into two classes for positive and negative reviews. the 2 classes are evenly balanced here. this is often an outsized dataset, and therefore the version that i'm using here only has the text as a feature with no other metadata. This makes this a stimulating dataset for doing NLP work. its data written by users, so it's like there are various types standard spellings, and other variations that you simply might not find in curated sets published shed text.

Text Pre-processing:

For the easy interpretation of the data, the data should be transformed from its raw state. The text pre-processing steps we considered are:

1. Importing libraries

2. Normalization

3. Reading text

4. Train text

5. Test text

6. TF-IDF

7. Trian/ validation spilt

8. Padding sequence

Algorithms used:

The algorithms used in the proposed work are:

1. Convolutional Neural Network (CNN)

2. Recurrent Neural Network (RNN)

3. Embedding Matrix

Output:

The train test the model the accuracy is given by all three models so here we consider the CNN is the best model for sentiment analyses.

**1.2 PROBLEM STATEMENT:**

“Sentiment analysis and modelling using deep learning models and handling imbalanced classification problem in amazon review dataset”.

**1.3 OBJECTIVES:**

The objectives of proposed project are as follows:

1. To consider the amazon sentiment analysis.
2. To extract the features using deep learning techniques.
3. To analyse the performances of the models.
4. To handling imbalanced classification problem.

**1.4 ORGANIZATION OF REPORT**

Chapter 1 gives the brief introduction of Amazon Review Analysis Modelling using Deep Learning, problem statement objective of the system.

Chapter 2 contains literature survey that provide summary of individual paper.

Chapter 3 provides tools and technology accustomed achieve this and dataset detail.

Chapter 4 provides overview of existing work for analysis of dataset methodology high-level design.

Chapter 5 contains Implementation and its results libraries and inbuilt functionalities of every model

Chapter 6 contains model wise snap-shots of the results

Chapter 7 contains conclusion about Amazon Review Analysis Modelling using deep learning.

## CHAPTER 2

**LITERATURE SURVEY**

All Information within the world may be broadly classified into mainly two categories, facts and opinions. Facts are objective statements about entities and worldly events. On the opposite hand, opinions are subjective statements that reflect people’s sentiments or perceptions about the entities and events. The Maximum amount of existing research on text and knowledge processing is targeted on mining and getting factual information from the text or information.

[1]. A Comparison of Sentiment Analysis Methods Reviews of Mobile Phones International Journal of Advanced Computer Science and Applications, Vol. 10, No. 6, 2019, Norah Saleh Alghamdi 2 School of Computing, Dublin City University (DCU)1 Dublin, Ireland Department of Computer Science1,2 Princess Nourah bint Abdulrahman University (PNU), They apply Lime technique to produce analytical reasons for the reviews being classified as either positive, negative or neutral, during this research, they experiments revealed that convolutional neural network with word2vec as a feature extraction technique provides the simplest results for both the unbalanced and balanced versions of the dataset.

[2]. Sentiment Analysis on Reviews of Mobile Users, The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014). Lin Zhanga, a School of Computer Science and Engineering, Beihang University. Sentiment Polarity classification; Mobile Application Review; Comparative Experiment; Machine Learning. A series of comparative experiments are finished classification algorithms, feature representations review.

[3]. Using TF-IDF to Determine Word Relevance in Document Queries 2018. Juan Ramos JURAMOS@EDEN.RUTGERS.EDU Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855. Algorithms for Ad-Hoc Retrieval, Encoding TF-IDF, Mathematical Framework his paper results of applying Term Frequency Inverse Document Frequency (TF-IDF) to work out what words in an exceedingly corpus of documents might be more favorable to use during a question. TF-IDF numbers imply a robust relationship with the document they appear in, suggesting that if that word were to seem in an exceedingly very query, the document could also be of interest to the user. we provide evidence that this straightforward algorithm efficiently categorizes relevant words that will enhance query retrieval.

[4]. Learning from imbalanced data: open challenges and future directions, Prog Artif Intell (2016). Machine learning Imbalanced data, multi-class imbalance big data, data streams Imbalanced clustering Imbalanced regression. The paper addresses discussing open issues and challenges that require to be addressed to further develop the sector of imbalanced learning.

[5]. A review on classification of imbalanced data for wireless sensor networks, First Published April 14, 2020

Harshita Patel, Dharmendra Singh Rajput, School of Information Technology & Engineering Vellore,Wireless sensor networks, data processing, imbalanced data, data balancing, algorithm modification, ensemble techniques. The paper addresses a good analysis of imbalance issue for wireless sensor networks and other application domains, which can help the community to grasp WHAT, WHY, and WHEN of imbalance in data and its remedies.

[6]. Dealing with Data Imbalance in Text Classification, Procedia Computer Science 159 (2019). Cristian Padurariua, Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Imbalanced text classification, oversampling techniques, cost-sensitive methods, text vectorization, differential evolution. propose and analyses a price sensitive approach formulated as a numerical optimization problem where the prices are derived with a Differential Evolution algorithm. the analysis is concentrated on studying the interactions between classification algorithms, text vectorization choices and also the schemes to pander to data imbalance at several degrees of imbalance; b) besides state-of-the-art balancing schemes, They propose and analyze a value sensitive approach formulated as a numerical optimization problem where the prices are derived with a Differential Evolution algorithm in two steps: in an exceedingly opening costs are optimized at the category level and in an exceedingly subsequent step costs are refined at the information instance level. The results indicate that the utilization of cost-sensitive classifiers where the price matrices are optimized with a Differential Evolution algorithm brings important benefits on our real-world problem.

[7]. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Peter D. Turney Institute for Information Technology National Research Council of Canada Ottawa, Ontario, Canada, K1A 0R6. This paper presents an easy unsupervised learning algorithm for classifying reviews as recommended (thumbs up) or not recommended (thumbs down). The classification of a review is predicted by the common semantic orientation of the phrases within the review that contain adjectives or adverbs. A phrase incorporates a positive semantic orientation when it's good associations (e.g., “subtle nuances”) and a negative semantic orientation when it's bad associations (e.g., “very cavalier”), during this paper, the semantic orientation of a phrase is calculated because the mutual information between the given phrase and therefore the word “excellent” minus the mutual information between the given phrase and also the word “poor”. The algorithm achieves a median accuracy of 74% when evaluated on 410 reviews from Opinions, sampled from four different domains (reviews of automobiles, banks, movies, and travel destinations). The accuracy ranges from 84% for automobile reviews to 66% for movie reviews.

[8]. Mining and Summarizing Customer Reviews Mining Hu and Bing Liu Department of Computer Science University of Illinois at Chicago 851 South Morgan Street Chicago, IL 60607-7053. Their task is performed in three steps: (1) mining product features that are commented on by customers; (2) identifying opinion sentences in each review and deciding whether each opinion sentence is positive or negative; (3) summarizing the results. This paper proposes several novel techniques to perform these tasks. Their experimental results using reviews of variety of products sold online demonstrate the effectiveness of the techniques, used [Database Management]: Database Applications – data processing. I.2.7 [Artificial Intelligence]: tongue Processing – text analysis.

[9]. Learning Subjective Adjectives from Corpora Janyce M. Wiebe Department of Computer Science New Mexico State University, Las Cruces, NM 88003. This paper identifies strong clues of subjectivity using the results of a technique for clustering words consistent with distributional similarity (Lin 1998), seeded by alittle amount of detailed manual annotation. These features are then further refined with the addition of lexical semantic features of adjectives, specifically polarity and gradeability (Hatzivassiloglou & McKeown 1997), which might be automatically learned from corpora. In 10-fold cross validation experiments, features supported both similarity clusters and also the lexical semantic features are shown to own higher precision than features supported each alone.

[10]. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean “Distributed representations of words and phrases and their compositionality,” in Advances in neural information processing systems (2013), pp. 3111-3119. In this paper, they present several improvements that make the Skip-gram model more expressive and enable it to be told higher quality vectors faster. We show that by subsampling frequent words we obtain significant speedup, and also learn higher quality representations as measured by our tasks. We also introduce Negative Sampling, a simplified variant of Noise Contrastive Estimation (NCE) that learns more accurate vectors for frequent words compared to the hierarchical SoftMax. An inherent limitation of word representations is their indifference to ordination and their inability to represent idiomatic phrases. for instance, the meanings of Canada'' and "Air'' can not be easily combined to get "Air Canada''. Motivated by this instance, we present a straightforward and efficient method for locating phrases, and show that their vector representations are often accurately learned by the Skip-gram model.

[11]. Dealing with Data Imbalance in Text Classification. Juan Ramos juramos@eden.rutgers.EDU Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855. They perform an intensive experimental analysis using various representations of text data, several classification algorithms and balancing schemes to derive a model that achieves highest performance with reference to metrics like precision and recall. The contribution is twofold) with a comprehensive experimental design, the analysis is targeted on studying the interactions between classification algorithms, text vectorization choices and also the schemes to handle data imbalance at several degrees of imbalance. They propose and analyze a price sensitive approach formulated as a numerical optimization problem where the prices are derived with a Differential Evolution algorithm.

[12]. Review of Text Classification in Deep Learning, AUTHORS: Qi Wang, Wenling Li, Zhezhi Jin

JOURNAL NAME: Open Access Library Journal, Vol.8 No.3, March 25, 2021. Text classification is a very important research content in language processing. Compared with traditional manual processing, text classification supported deep learning improves both efficiency and accuracy. However, within the learning process, the content involved is incredibly large and sophisticated. so as to facilitate the research of more scholars, this paper summarizes the text classification of deep learning. the primary a part of this paper introduces the preprocessing of text classification. The second part introduces several feasible methods for deep learning text classification very well. The third part introduces the test method of the model. The twenty-five percent summarizes and analyzes the benefits and drawbacks of several methods to get a foundation for further research.

[13]. Sentiment Analysis of Chinese Microblog Based on Stacked Bidirectional LSTM JUNHAO ZHOU1 , YUE LU1 , HONG-NING DAI 1 , (Senior Member, IEEE), HAO WANG 2 , (Member, IEEE), AND HONG XIAO3.

Sentiment analysis on Chinese microblogs has received extensive attention recently. Most previous studies specialise in identifying sentiment orientation by encoding as many word properties as possible while they fail to think about contextual features (e.g., the long-range dependencies of words), which are however, essentially important within the sentiment analysis. during this paper, they propose a Chinese sentiment analysis method by incorporating a word2vec model and a stacked bidirectional long STM (Stacked Bi-LSTM) model. They first employ the word2vec model to capture semantic features of words and transfer words into high-dimensional word vectors. They evaluate the performance of two typical word2vec models: continuous bag-of-words (CBOW) and skip-gram. They then use the Stacked Bi-LSTM model to conduct the feature extraction of sequential word vectors. They next apply a binary SoftMax classifier to predict the sentiment orientation by using semantic and contextual features. Moreover, they also conduct extensive experiments on the important dataset collected from Weibo (i.e., one amongst the foremost popular Chinese microblogs). The experimental results show that their proposed approach achieves better performance than other machine-learning models.

[14]. Ranking Online Consumer Reviews March 2018Electronic Commerce Research and Applications 29 DOI: 10.1016/j.elerap.2018.03.008. the aim of this study is to rank the overwhelming number of reviews using their predicted helpfulness scores. The helpfulness score is predicted using features extracted from review text, product description, and customer question-answer data of a product using the random-forest classifier and gradient boosting regressor. The system classifies reviews into low or prime quality with the random-forest classifier. The helpfulness countless the high-quality reviews are only predicted using the gradient boosting regressor. The helpfulness countless the low-quality reviews don't seem to be calculated because they're never visiting be within the top k reviews. they're just added at the tip of the review list to the review-listing website. The proposed system provides fair review placement on review listing pages and makes all high-quality reviews visible to customers on the highest. The experimental results on data from two popular Indian e-commerce websites validate their claim, as 3–4 newer high-quality reviews are placed within the top ten reviews together with 5–6 older reviews supported review helpfulness. Their findings indicate that inclusion of features from product description data and customer question-answer data improves the prediction accuracy of the helpfulness score.

[15]. Predicting the “helpfulness” of online consumer reviews, Singh, Jyoti Prakash Irani, Seda Rana, Nripendra P. [Journal of Business Research](https://ideas.repec.org/s/eee/jbrese.html), Elsevier, vol. 70(C), pages 346-355. They developed models that supported machine learning which will predict the helpfulness of the buyer reviews using several textual features like polarity, subjectivity, entropy, and reading ease. The model will automatically assign helpfulness values to an initial review as soon because it is posted on the website in order that the review gets a good chance of being viewed by other buyers. The results of this study will help buyers to put in writing better reviews and thereby assist other buyers in making their purchase decisions, yet as help, businesses to boost their websites.

[16]. Sentiment Analysis Using Text Mining: A Review International Journal on Data Science and Technology Swati Redhu, Volume 4, Issue 2, June 2018, Pages: 49-53 Received: Jun. 25, 2018; Published: Jun. 26, 2018. Text mining and sentiment analysis have received huge attention recently, especially due to the supply of vast data in type of text available on social media, e-commerce websites, blogs and other similar sources. This data is sometimes unstructured and contains noise, therefore the task of gaining information is complex and expensive. there's a growing need for developing different methodologies and models for efficiently processing the texts and extracting apt information. a way to extract information is text mining and sentiment analysis, which include: data acquisition, data pre-processing and normalization, feature extraction and representation, labelling, and eventually the applying of varied linguistic communication Processing (NLP) and machine learning algorithms. This paper provides a summary of various methods employed in text mining and sentiment analysis elaborating on all subtasks.

[17]. Sentiment analysis of online product reviews using DLMNN and future prediction of online product using IANFIS Authors: P. Sasikala and L. Mary Immaculate Sheela Date: May 19, 2020 From: Journal of Big Data (Vol. 7, Issue 1) Publisher: Springer. A major task that the NLP (Natural Language Processing) needs to follow is Sentiments analysis (SA) or opinions mining (OM). for locating whether the user's attitude is positive, neutral or negative, it captures each user's opinion, belief, and feelings about the corresponding product. Most of the existent techniques on SA aimed toward these online products have extremely low accuracy and also encompassed longer amid training. By employing a Deep learning modified neural network (DLMNN), a method is proposed geared toward SA of online products review; additionally, via Improved Adaptive Neuro-Fuzzy Inferences System (IANFIS), a way is proposed aimed toward future prediction of online products to trounce the above-stated issues.

**CHAPTER 3**

**REQUIREMENT ANALYSIS**

#### 3.1 Functional Requirements:

A functional requirement is a function of a system with inputs, required for a system to function and outputs that it produces. Functional requirements for the application are as follows:

The Proposed Algorithm’s should give an accuracy.

Here we use technologies that help to build a natural language processing toolkit like Python, Jupiter notebook, Google Collab etc.

#### 3.2 Non-Functional Requirements:

Non-Functional requirements for the application are as follows:

* Compatibility: The application should work on any machine which has required configurations.
* Availability: The application should be available all the time.
* Performance: The application must provide high performance.
* Efficiency: The application must have good final test accuracy after the completion of training the model

#### 3.3 Hardware Requirements:

Below are mentioned are the hardware requirements of our project.

* + RAM 2GB and above
  + Windows OS
  + Processor: 64bit processor

#### 3.4 Software Requirements:

Below are mentioned are the software requirements of our project.

* + - Programming Language: Python
    - Python 2.7
    - Packages:
      * NumPy
      * Pandas

**CHAPTER 4**

**METHODOLOGY**

**4.1** **PROPOSED MODEL:**

In this chapter, the Proposed model will be discussed and used for analyses of the text dataset



**Pre-processing**

**Feature Extracting**

**Deep Learning algorithm's**

**CNN**

**RNN**

**EMBEDDING MATRIX**

**Result**

Fig 4.1: Proposed Model

**4.2 Dataset:**

This dataset contains several million reviews of Amazon products, with the reviews separated into two classes for positive and negative reviews. the 2 classes are evenly balanced here.

Number of reviews: 568,454

Number of users: 256,059

Number of products: 74,258

Timespan: Oct 1999 - Oct 2012

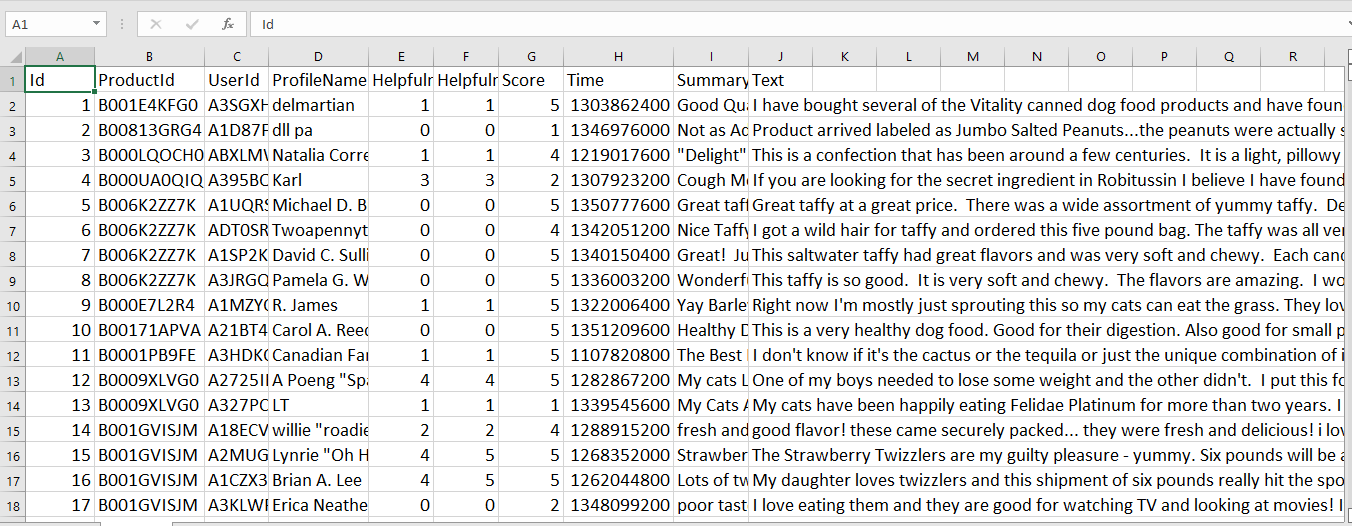
****

Fig 4.2: dataset

**4.2.1 Attribute Information:**

1. Id

2. ProductId - unique identifier for the merchandise

3. UserId - unique identifier for the user

4. Profile Name

5. Helpfulness Numerator - number of users who found the review helpful

6. Helpfulness Denominator - number of users who indicated whether or not they found the review helpful or not

7. Score - rating between 1 and 5

8. Time - timestamp for the review

9. Summary - brief summary of the review

10. Text - text of the review

**4.3 Feature Engineering**

This is an outsized dataset, and therefore the version that i'm using here only has the text as a feature with no other metadata. This makes this a noteworthy dataset for doing NLP work. it's data written by users, so it's like that there are various typos, nonstandard spellings, and other variations that you just might not find in curated sets of published text.

**4.3.1 Helpfulness Features**

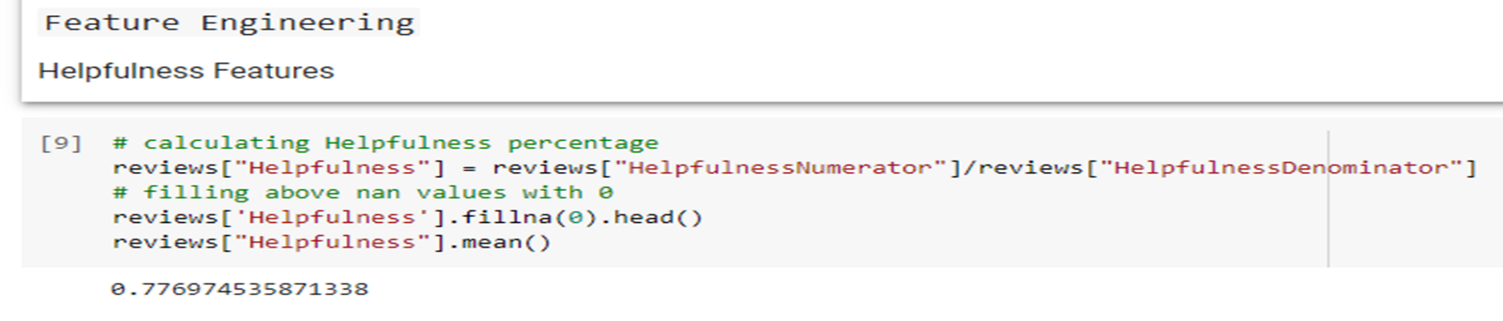
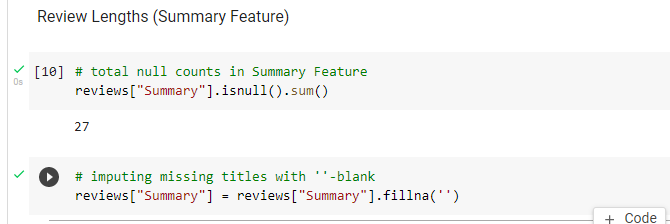


Fig 4.3.1 Helpfulness Features

**4.3.2: Review Lengths (summary feature)**

This is a straightforward metric associated with score of reviews for example, longer reviews may be more passionate so it could have a better score and contrariwise First, Imputing missing Summary titles with blanks (‘ ’)



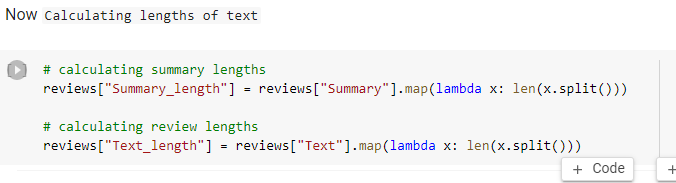


Fig4.3.2: Review Lengths (summary feature)

**4.3.3 Sentiment Score (Text, Summary Features)**



Fig 4.3.3: Sentiment Score (Text, Summary Features**)**

**4.3.4 Reviews Over Time (Time Feature)**

below raw values, dataset starts from 1999-2012.

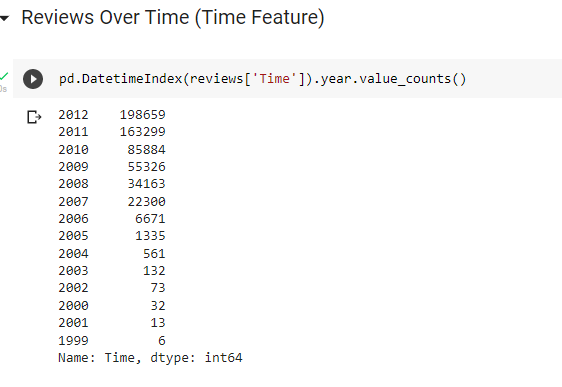


Fig 4.3.4: Reviews Over Time (Time Feature)

**4.3.5: Number of reviews over time**

below plot that number of reviews in this dataset exponentially grew over years up to 2012

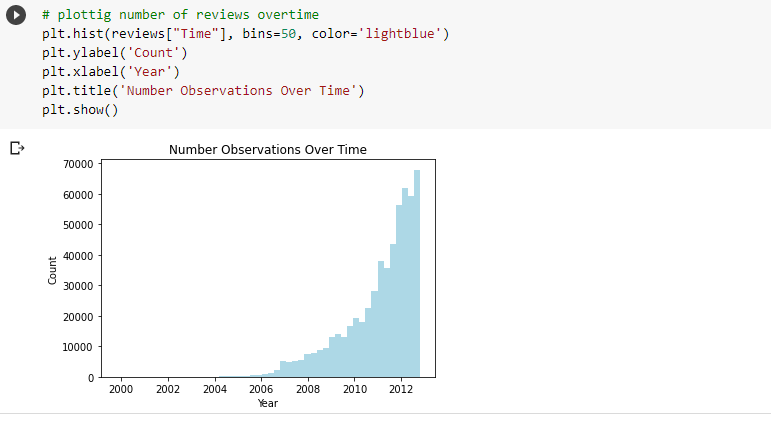


Fig 4.3.5: Number of reviews over time

**4.3.6. Number of scores over count**

below plot that number of reviews in this dataset exponentially plotting number of scores

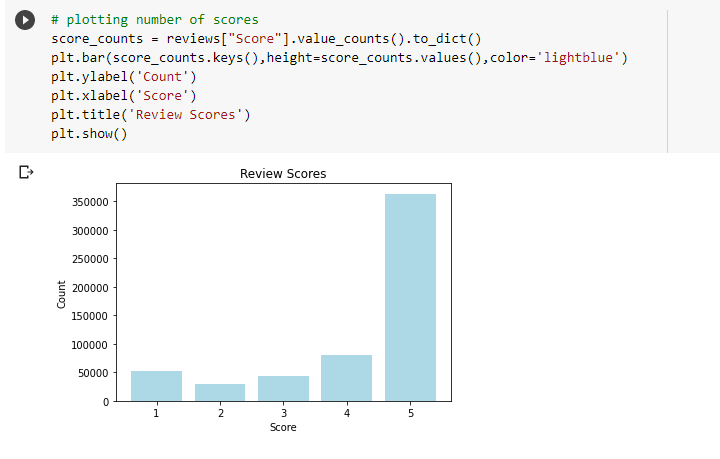
the

Fig 4.3.6. Number of scores over count

**4.3.7. Sentiment Distribution**

most of reviews are considered positive, this is often somewhat expected as most of reviews have 5 stars so reasonably would assume that it might be overwhelmingly positive in addition.

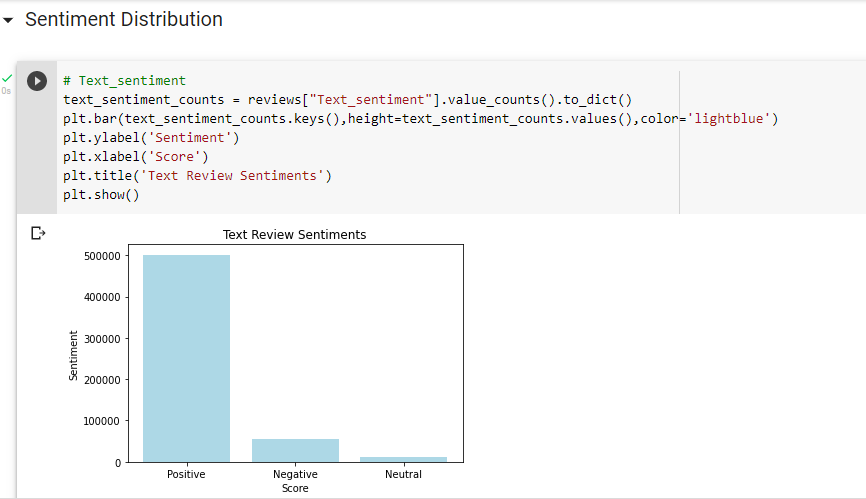


Fig 4.3.7: Sentiment Distribution

**4.3.8: Summary Sentiment**

Summary text shows a similar trend as Review text



Fig 4.3.8: Summary Sentiment

**4.4. Handling Imbalanced Classification:**

The classification is a very important technique of knowledge mining, within which unknown class samples are assigned to some class supported previous knowledge from training samples.

Imbalance appears when data are unequally distributed into classes, some classes may have profusion of knowledge called as majority classes and a few may have just few instances of knowledge called minority classes. This uneven distribution causes biased performance of traditional classifiers because they consider the error rate not the distribution of information, and thanks to having little quantity of information instances, minority classes get ignored in overall classification result.

**4.5 Feature Extracting:**

Dataset is textual, so it must be represented in numerical formats to be fed to the machine learning algorithms to create the specified classifiers. to realize this, different vectorisation techniques are performed including term frequency which involves counting all the occurrences of all the terms within the document or sentence. A term is often expressed as one word.

**Feature Engineering**

**Product Feature**

**Summary-length and Text-length feature**

**Review summary overview**

**TF-IDF Feature**

Fig 4.5: Feature extraction

**4.6: Product Popularity (Productid feature)**

in below plot there are top\_500\_reviewed products for sure number of reviews are right skewed Trying to require a summary look and see what mean score for these products are and trying to plotting it

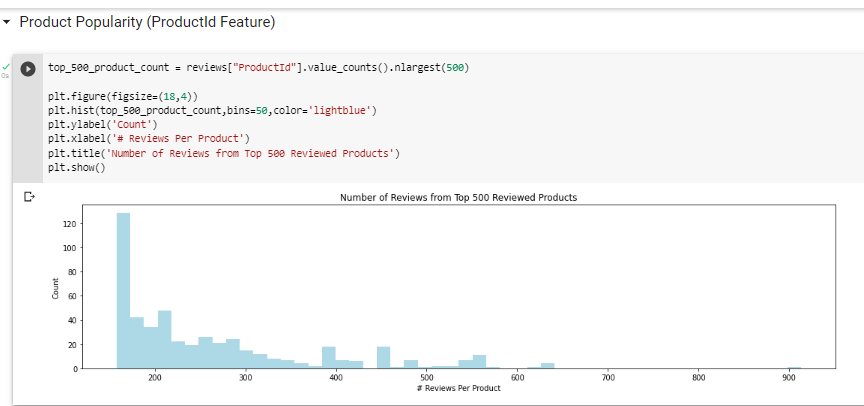


Fig 4.6: Product Popularity (Productid feature)

**4.7 Mean Score from top\_500\_products**

plot below shows histogram of Mean Score from top\_500\_products it's somewhat left skewed and features a multi-modal distribution Average stops any product from truly being 5 stars which is predicted as we'd expect to work out some variation in reviews.



Fig 4.7: Mean Score from top\_500\_products

**4.8: histogram of Mean Score from top\_500\_products**

plot below shows histogram of Mean Score from top\_500\_products.

It looks somewhat left skewed and contains a multi-modal distribution. Average stops any product from truly being 5 stars which is predicted as we'd expect to determine some variation in reviews.

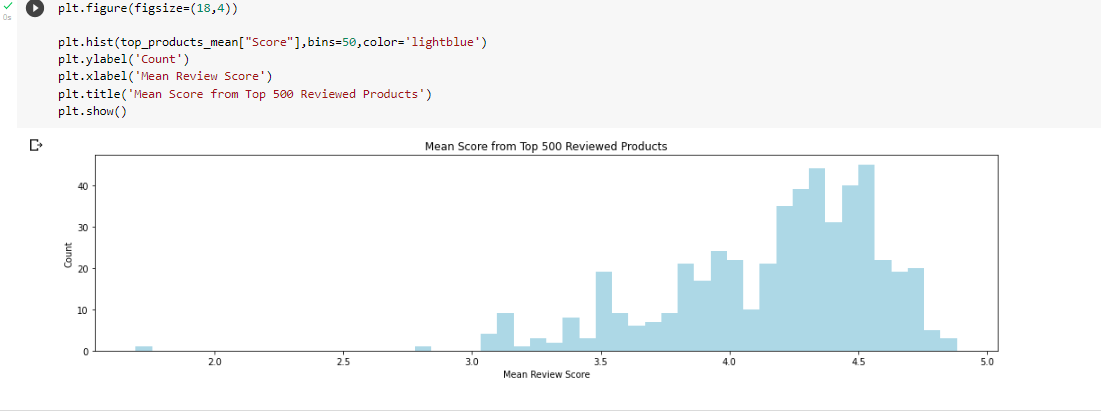


Fig 4.8: histogram of Mean Score from top\_500\_products

**4.9: Helpfulness with top\_500\_products**

n below plot similar pattern with Helpfulness with top\_500\_products None of review for these products are truly in concordance with helpfulness in either direction Getting a transparent peak at around 0.8 which looks somewhat normal

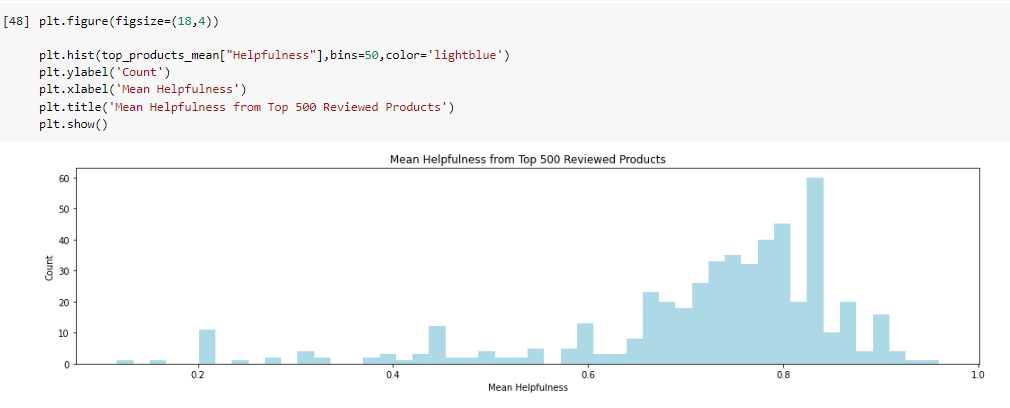


Fig 4.9: Helpfulness with top\_500\_products

**4.10: Age to helpfulness**

Trying to know if this Helpfulness has some Time-based dependency Calculating relative age of a product supported Last\_Review originates calculate Review\_Metrics and trying to visualise these trends.



Fig 4.10: Age to helpfulness

**4.11: Summary\_Length is right skewed**

plot shows that Summary\_Length is right skewed

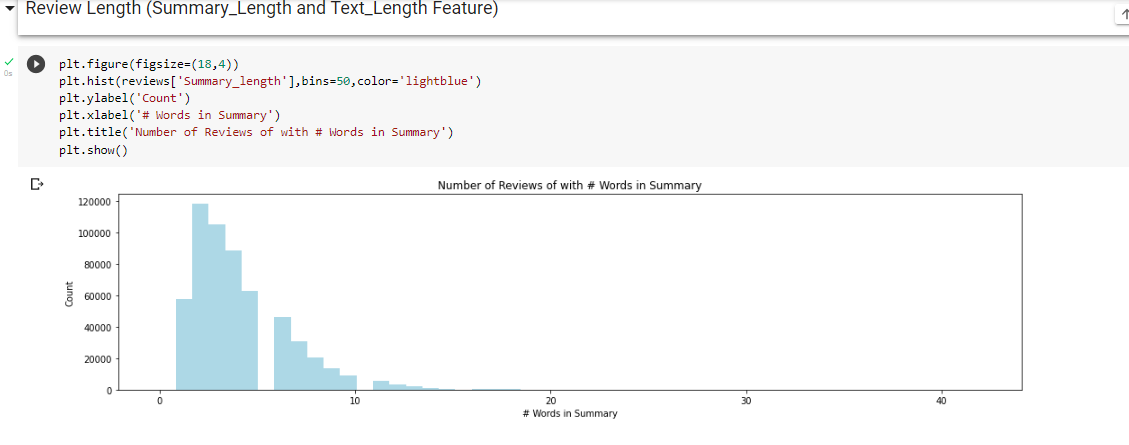


Fig4.11: Summary\_Length is right skewed

**4.12: distributed right skewed**

similar to summary, we see that distribution here is right skewed

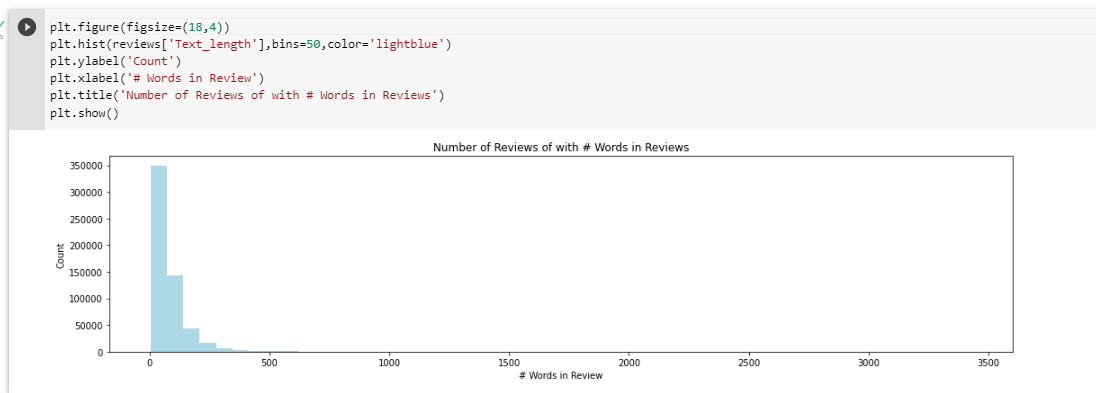


Fig4.12: distributed right skewed

**4.13: Helpfulness Ratio**

Interested in seeing if two variables have any relationship between one another can run Spearman\_Correlation as both variables are nonparametric. Spearman's correlation coefficient: 0.305, can see from above that correlation is weak but statistically significant therein there's some true relationship between two variables. in below plot most of Helpful\_Ratio is left skewed this might be expected as many reviews may need been rated helpful once which might heavily bias this score It will be checked by filtering out the only rated entries

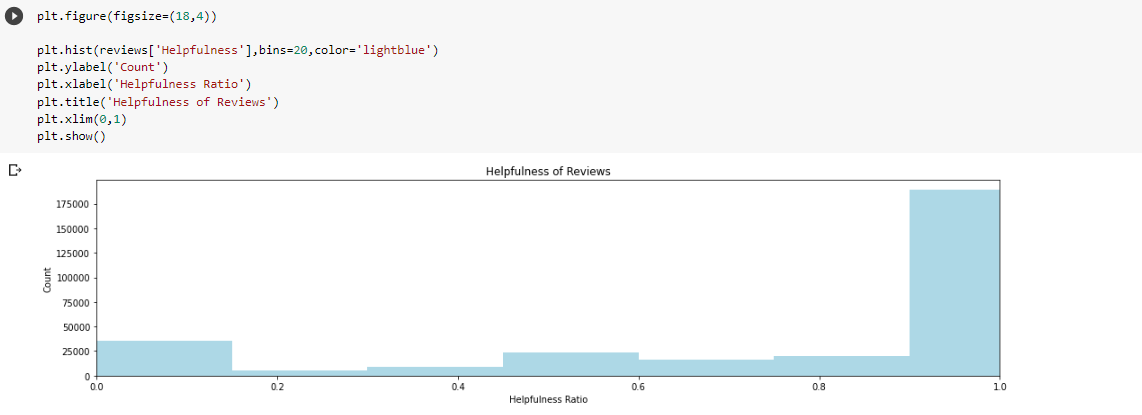


Fig4.13: Helpfulness ratio

**4.14: Reviews and Summary Overview**

Bulk of this dataset lies in Reviews and also the Summary tagline of the review.

There may be something in either of these features which could be interesting for classification.

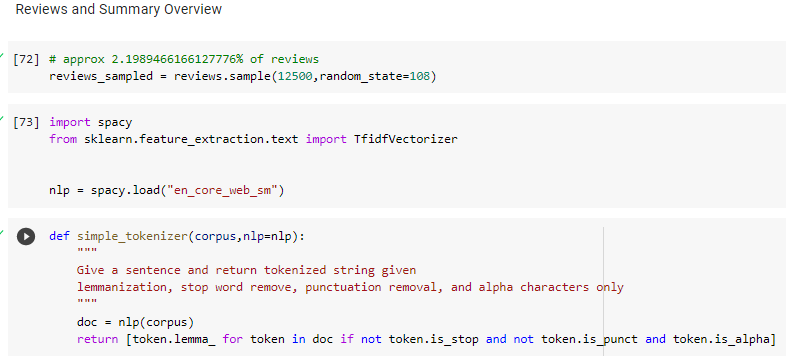


Fig4.14: Reviews and Summary Overview

**4.15: Summary and review for TF-IDF**

Additionally, it might be interesting to grasp those features for general data analysis to grasp Reviewers better. First, TF-IDF are going to be calculated high-level overview of what this corpus feels like

We will only be checking unigram for keyword checks and removing stop words, numeric, lemmatization and removing punctuation, as variety of documents are relatively large, a little of it'll be sampled before analysis.

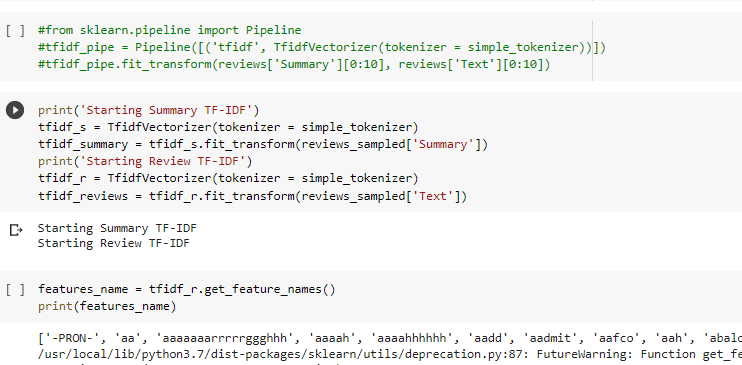


Fig 4.15: Summary and review for TF-IDF

**4.15.1: Top 40 TF-IDF features from this sample**

below Top 40 TF-IDF features from this sample

Although a number of these tokens are typos, see that the tokens are a mix of slang for positive things and food related words.



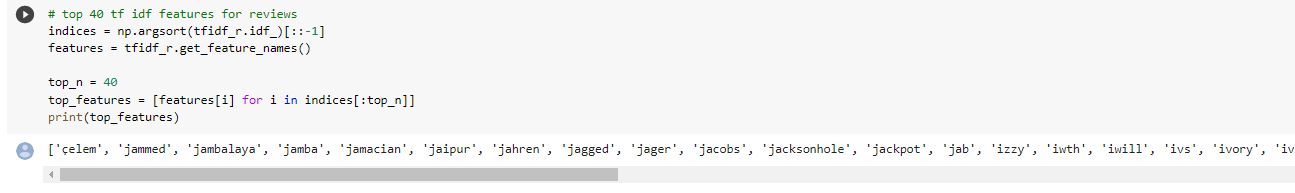
Fig 4.15.1: Top 40 TF-IDF features from this sample

**4.16: Score of Binary\_classification**

In below shows top features for actual reviews. can see some typos and a few non-English terms.

From here, can mostly see that original and a few physical ailments are becoming higher TF-IDF values.

Now can go one step further Trying to seek out top words given every year of reviews in our dataset it'd be interesting to determine if we will capture trend in foods over year using reviews.



First, Imputing Score to be Binary\_Classifcation

using scores of 4-5 as High and everything else as Low



Next, Setup Scatter text plots of every year of this corpus to get a sense of how things changed over time, to again see which years we have in dataset, following can be done:

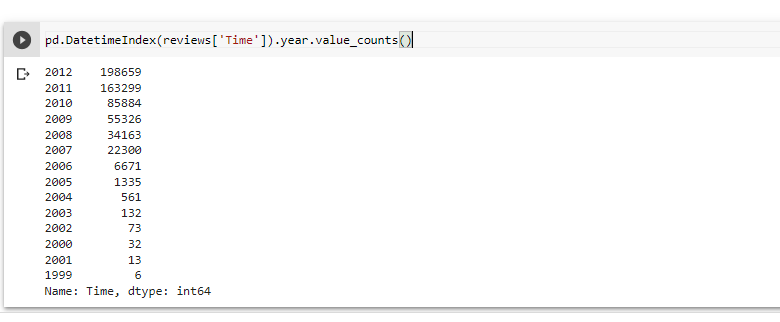


Fig4.16: Score of Binary\_classification over time count

**CHAPTER5: IMPLEMENTATION**

In this chapter, CNN RNN Embedding models will be discussed and used for analyses of the text dataset

**Dataset overview**

**Importing and installing libraries**

**Reading the text**

**Text-processing**

**Train/Validating split**

**Padding sequence**

**Convolutional Neural Net model**

**Embedding matrix**

**Recurrent Neural Net Model**

Fig5.1: Implementations steps

**5.2: Implementation steps:**

Implementation steps for Convolutions Neural network

Step1: Dataset overview

Step2: Importing and installing libraries

Step3: Reading the text

Step4: Text-processing

Step5: Train/Validating split

Step6: Padding sequence

Step7: Build CNN model

Step8: Fit CNN model

Step9: Convolutional Neural Net model

Implementation steps for Recurrent Neural Network

Step1: Dataset overview

Step2: Importing and installing libraries

Step3: Reading the text

Step4: Text-processing

Step5: Train/Validating split

Step6: Padding sequence

Step7: Build RNN model

Step8: Fit RNN model

Step9: Convolutional Neural Net model

Implementation steps for Embedding Matrix

Step1: Dataset overview

Step2: Importing and installing libraries

Step3: Reading the text

Step4: Text-processing

Step5: Train/Validating split

Step6: Build Embedding Matrix model

Step7: Fit Embedding Matrix model

Step9: Embedding Matrix model

**5.3 Amazon Reviews Dataset:**

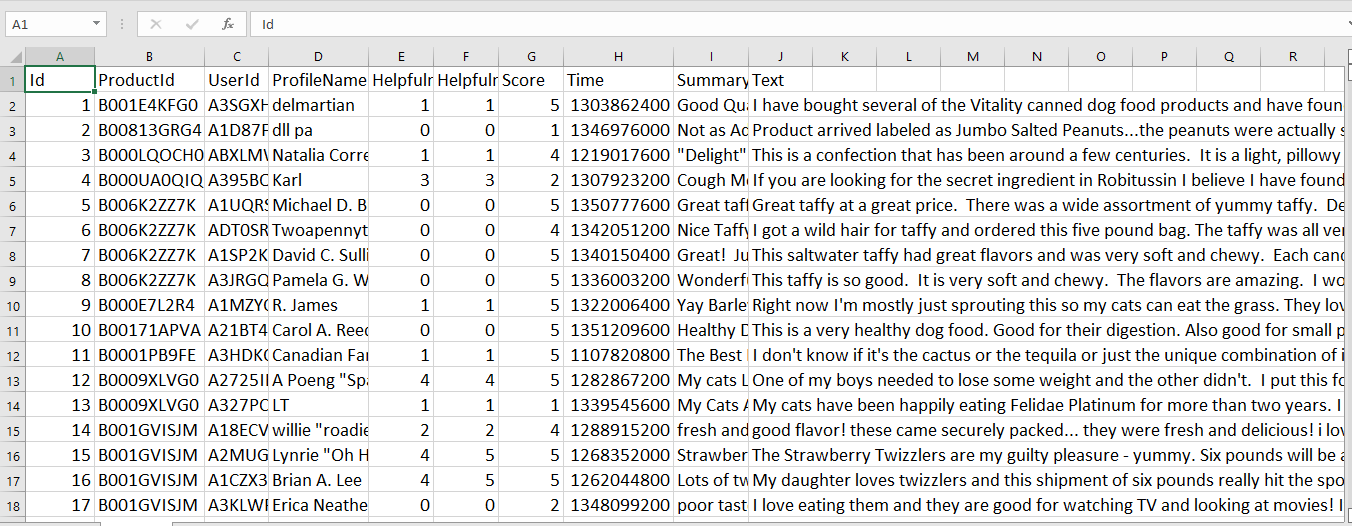
This dataset contains several million reviews of Amazon products, with the reviews separated into two classes for positive and negative reviews. the 2 classes are evenly balanced here.

This is an oversized dataset, and also the version that i'm using here only has the text as a feature with no other metadata. This makes this a remarkable dataset for doing NLP work. it's data written by users, so it's like that there are various typos, nonstandard spellings, and other variations that you just might not find in curated sets of published text. i'll do some text processing so attempt three fairly unoptimized deep learning models:

1. A convolutional neural net
2. A recurrent neural net

3. An Embedding matrix

These models should achieve results that are within a couple percent of state of the art at predicting the binary sentiment of the reviews.

****

5.3: Amazon review dataset overview

**5.3.1 Attributes information**

1. Id

2. ProductId - unique identifier for the merchandise

3. UserId - unique identifier for the user

4. Profile Name

5. Helpfulness Numerator - number of users who found the review helpful

6. Helpfulness Denominator - number of users who indicated whether or not they found the review helpful or not

7. Score - rating between 1 and 5

8. Time - timestamp for the review

9. Summary - brief summary of the review

10. Text - text of the review

**5.4 IMPORTING THE LIBERIERS:**

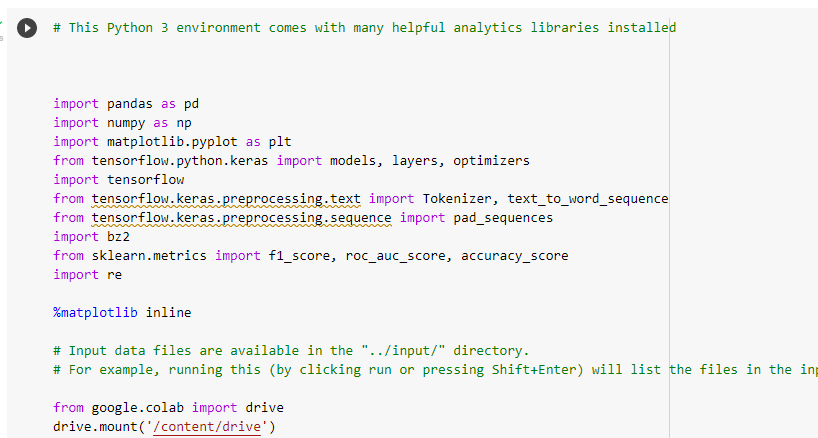


Fig 5.4: importing libraries

**5.5 READING THE TEXT:**

The text is held in a very compressed format. Luckily, we are able to still read it line by line. the primary word gives the label, so we've to convert that into variety and so take the remainder to be the comment.



Fig5.5: Reading the text

**5.6 TEXT PRE-PROCESSING:**

The first thing visiting do to process the text is to lowercase everything so remove non-word characters. Replace these with spaces since most are visiting be punctuation. Then visiting just remove the other characters (like letters with accents). It can be better to exchange a number of these with regular ascii characters. It also seems if you take a look at the counts of the various characters that there are only a few unusual characters during this corpus.



Fig5.6: Text processing

**5.7: Train/Validation Split**



Fig 5.7: Train/Validation Split

**5.8: Padding Sequences**

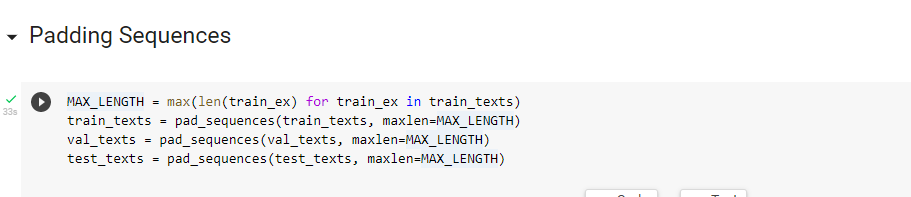
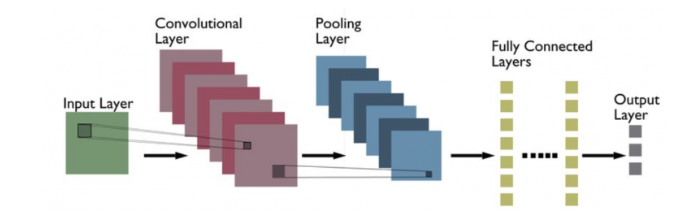


Fig: 5.8: Padding Sequences

**5.9 CONVOLUTIONAL NEURAL NETWORK:**

CNN may be a supervised algorithm, so it needs labelled data to advance the weights of its convolutional filters. it receives the information from feature extraction as input then sends it to hidden layers called convolutional layers. These layers are the premise of CNNs. the primary layer transforms the input, then the output from this layer sends it to the opposite layer. it's a sequence until the last layer. This process is termed convolutional operation.



The following are definitions of various layers shown within the above architecture:

Convolutional layer: Convolutional layers are made from a group of filters that are applied to an input image. The output of the convolutional layer may be a feature map, which may be a representation of the input image with the filters applied. Convolutional layers are often stacked to make more complex models, which may learn more intricate features from images.

Pooling layer: Pooling layers are a sort of convolutional layer employed in deep learning. Pooling layers reduce the spatial size of the input, making it easier to process and requiring less memory. Pooling also helps to scale back the number of parameters and makes training faster. There are two main styles of pooling: max pooling and average pooling. Max pooling takes the utmost value from each feature map, while average pooling takes the common value. Pooling layers are typically used after convolutional layers so as to scale back the dimensions of the input before it's fed into a totally connected layer.

Fully connected layer: Fully-connected layers are one {in alone amongst one in every of} the foremost basic styles of layers in a convolutional neural network (CNN). because the name suggests, each neuron during a fully-connected layer is Fully connected- to each other neuron within the previous layer. Fully connected layers are typically used towards the top of a CNN- when the goal is to require the features learned by the previous layers and use them to create predictions. as an example, if we were employing a CNN to classify images of animals, the ultimate Fully connected layer might take the features learned by the previous layers and use them to classify a picture as containing a dog, cat, bird, etc.

CNNs are often used for image recognition and classification tasks. as an example, CNNs are often wont to identify objects in a picture or to classify a picture as being a cat or a dog. CNNs may be used for more complex tasks, like generating descriptions of a picture or l identifying the points of interest in a picture. CNNs also can be used for time-series data, like audio data or text data. CNNs are a strong tool for deep learning, and that they are accustomed achieve state-of-the-art ends up in many alternative applications.

The CNN architectures are the foremost popular deep learning framework. CNNs are used for a range of applications, starting from computer vision to linguistic communication processing. during this blog post, we'll discuss each form of CNN architecture well and supply samples of how these models work. Even before we get to be told about the various varieties of CNN architecture, let’s briefly recall what's CNN within the first place.

CNNs are a kind of deep learning algorithm that are wont to process data with a grid-like topology. CNNs are a kind of deep learning algorithm that's want to process data that incorporates a spatial or temporal relationship. CNNs are almost like other neural networks, but they need another layer of complexity because of the very fact that they use a series of convolutional layers. Convolutional layers are an important component of Convolutional Neural Networks (CNNs). the image below represents a typical CNN architecture.

The following may be a list of various sorts of CNN architectures:

leNet: leNet is that the first CNN architecture. it absolutely was developed in l998 by Yann leCun, Corinna Cortes, and Christopher Burges for handwritten digit recognition problems. leNet was one in all the primary successful CNNs and is usually considered the “Hello World” of deep learning. it's one in every of the earliest and most widely-used CNN architectures and has been successfully applied to tasks like handwritten digit recognition. The leNet architecture consists of multiple convolutional and pooling layers, followed by a fully-connected layer. The model has five convolution layers followed by two fully connected layers. leNet was the start of CNNs in deep learning for computer vision problems. However, leNet couldn't train well thanks to the vanishing gradients problem. to unravel this issue, a shortcut connection layer referred to as max-pooling is employed between convolutional layers to scale back the spatial size of images which helps prevent overfitting and allows CNNs to coach more effectively. The diagram below represents leNet-5 architecture.

CNNs are often used for image recognition and classification tasks. as an example, CNNs are often accustomed identify objects in a picture or to classify a picture as being a cat or a dog. CNNs also can be used for more complex tasks, like generating descriptions of a picture or l identifying the points of interest in a picture. CNNs may also be used for time-series data, like audio data or text data. CNNs are a strong tool for deep learning, and that they are accustomed achieve state-of-the-art ends up in many various applications.

**5.9.1 CNN MODEL BUILDING AND FITTING**



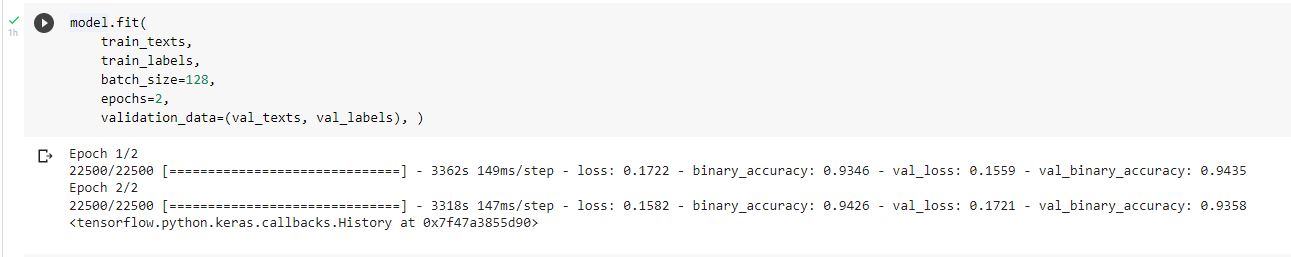


fig5.9.1 CNN model building and Model fitting

**5.10 RECURRENT NEURAL NET MODEL:**

A recurrent neural network (RNN) may be a style of artificial neural network which uses sequential data or statistic data. These deep learning algorithms are commonly used for ordinal or temporal problems, like language translation, language processing (NLP), speech recognition.

Recurrent neural networks utilize training data to be told. they're distinguished by their “memory” as they take information from prior inputs to influence the present input and output. While traditional deep neural networks assume that inputs and outputs are independent of every other, the output of recurrent neural networks rely on the prior elements within the sequence.

Training through RNN

A single time step of the input is provided to the network.

Then calculate its current state using set of current input and therefore the previous state.

The current ht becomes ht-1 for the subsequent time step.

One can go as many time steps per the matter and join the data from all the previous states.

Once all the time steps are completed the ultimate current state is employed to calculate the output.

The output is then compared to the particular output i.e the target output and therefore the error is generated.

The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained.

Advantages of Recurrent Neural Network

An RNN remembers each and each information through time. it's useful in statistic prediction only thanks to the feature to recollect previous inputs likewise. this is often called Long Short-Term Memory.

Recurrent neural network is even used with convolutional layers to increase the effective pixel neighborhood.

Disadvantages of Recurrent Neural Network.

Gradient vanishing and exploding problems.

Training an RNN may be a very difficult task.

It cannot process very long sequences if using tanh or rely as an activation function.

Recurrent Neural Networks or RNNs, are an awfully important variant of neural networks heavily employed in linguistic communication Processing. There is a category of neural networks that allow previous outputs to be used as inputs while having hidden states.

RNN features a concept of “memory” which remembers all information about what has been calculated till time step t. RNNs are called recurrent because they perform the identical task for each element of a sequence, with the output being relied on the previous computations.

Before we deep dive into the main points of what a recurrent neural network is, let’s first understand why will we use RNNs in first place.

Recurrent Neural Network (RNN):

In a general neural network, an input is fed to an input layer and is further processed through number of hidden layers and a final output is produced, with an assumption that two successive inputs are independent of every other or input at time step t has no relation with input at timestep t-1.

However, this assumption isn't true during a number of real-life scenarios. for example, if one wants to predict the value of a stock at a given time or wants to predict the subsequent word during a sequence then it's imperative that dependence on previous observations is taken into account.

**5.10.1: recurrent neural net model building**



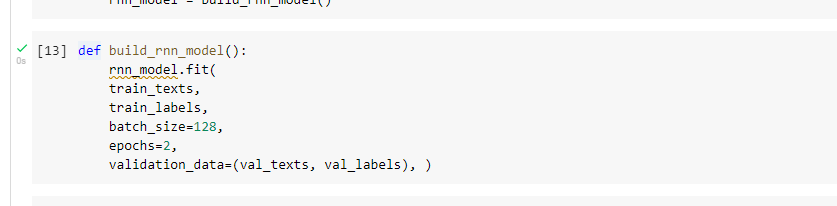


Fig 5.10.1: recurrent neural net model building

**5.11 EMBEDDING MATRIX:**

An embedding matrix may be a listing of all words and their corresponding embeddings.

The Embedding layer is initialized with random weights and may learn an embedding for all of the words within the training dataset. It is a flexible layer which is able to be utilized in a very spread of how, such as: it's used alone to be told a word embedding which is able to be saved, Embeddings make it easier to undertake to to machine learning on large inputs like sparse vectors representing words. Ideally, an embedding capture variety of the semantics of the input by placing semantically similar inputs close within the embedding space. An embedding is also learned and reused across models.

In deep learning, embedding layer looks like an enigma until you get the hold of it. Since embedding layer is a necessary a part of neural networks, it's important to grasp the working of it. during this article, i will be able to attempt to explain what's embedding layer, what's the requirement of it and the way it works, together with some coding examples. So, let’s start.

Embedding Layer

Embedding layer is one in every of the available layers in Keras. this can be mainly employed in linguistic communication Processing related applications like language modeling, but it also can be used with other tasks that involve neural networks. While coping with NLP problems, we are able to use pre-trained word embeddings like GloVe. Alternatively, we will also train our own embeddings using Keras embedding layer.

Need of Embeddings

Word embeddings are often thought of as an alternate to one-hot encoding together with dimensionality reduction.

As we all know while managing textual data, we want to convert it into numbers before feeding into any machine learning model, including neural networks. For simplicity words are often compared to categorical variables. We use one-hot encoding to convert categorical features into numbers. To do so, we create dummy features for every of the category and populate them with 0’s and 1's.

Similarly, if we use one-hot encoding on words in textual data, we'll have a dummy feature for every word, which suggests 10,000 features for a vocabulary of 10,000 words. this is often not a feasible embedding approach because it demands large space for storing for the word vectors and reduces model efficiency.

Embedding layer enables us to convert each word into a set length vector of defined size. The resultant vector may be a dense one with having real values rather than just 0’s and 1’s. The fixed length of word vectors helps us to represent words in a very better way together with reduced dimensions.

This way embedding layer works sort of a lookup table. The words are the keys during this table, while the dense word vectors are the values. to grasp it better, let’s have a look at the implementation of Keras Embedding layer

**5.11.1 Build Embedding matrix**

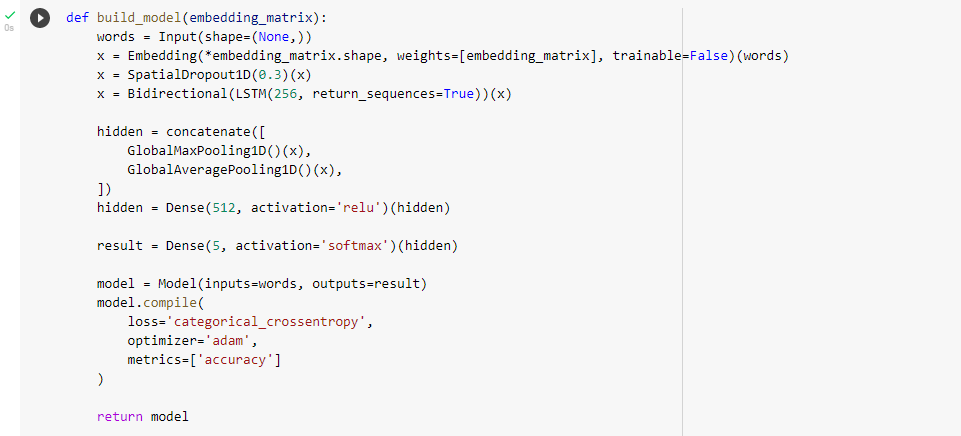
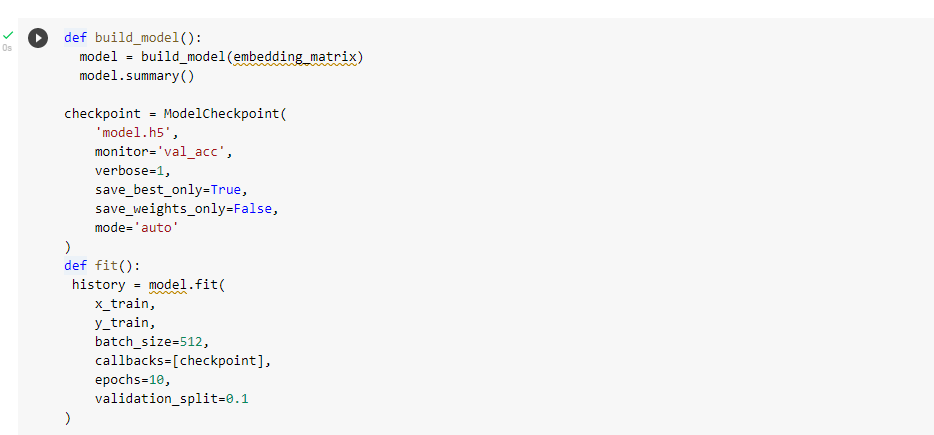


Fig5.11.1 Build Embedding matrix

**5.11.2 Fit Embedding Matrix Model**



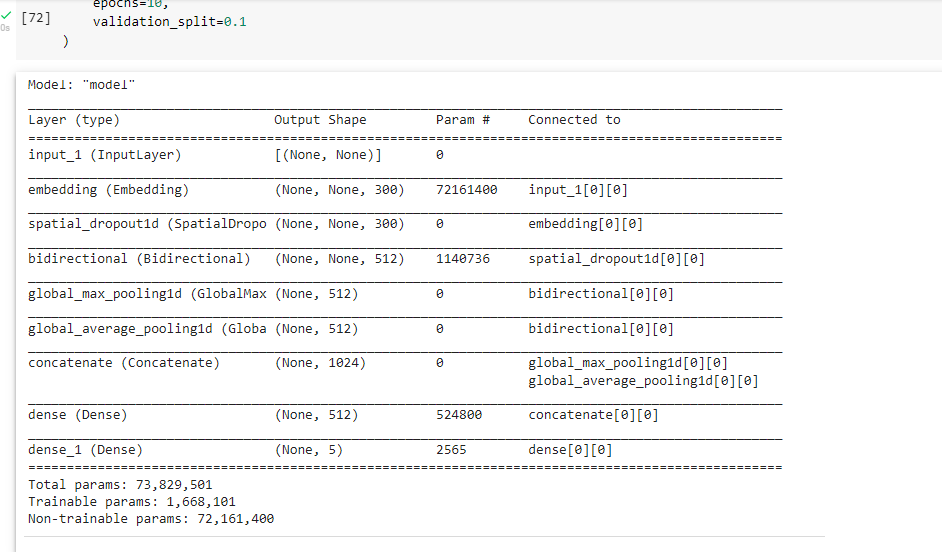


Fig5.11.2: Fit Embedding Matrix Model

**CHAPTER 6: RESULTS**

**6.1: CNN RESULT**

Got an accuracy of around 94% for this model. Roc auc score: 98%

ROC: Receiver operating characteristic's curve AUC (area under the curve) - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability.

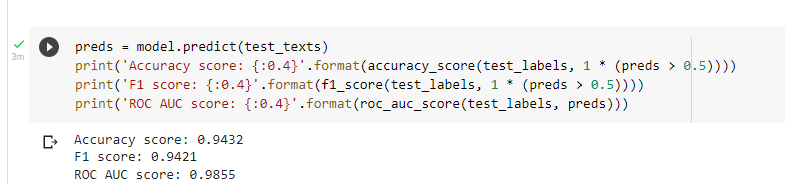


fig6.1: CNN result

**6.2: RNN Result:**

Got an accuracy of around 93% for this model.

ROC: Receiver operating characteristic's curve 98% AUC (area under the curve) ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability.

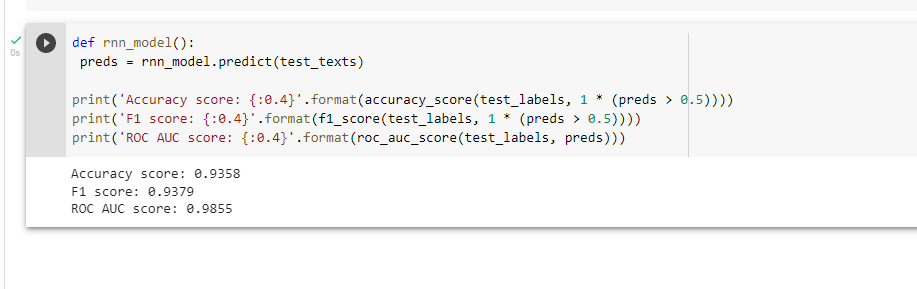
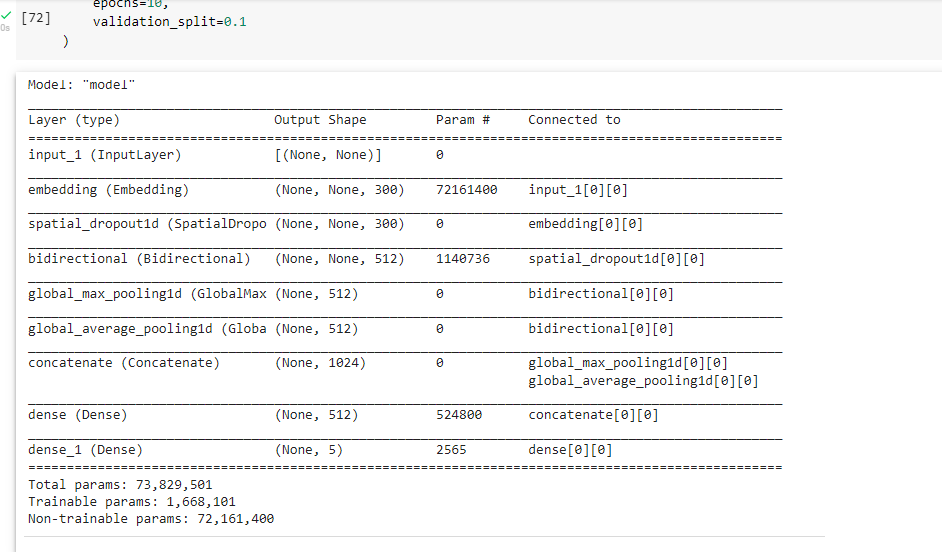
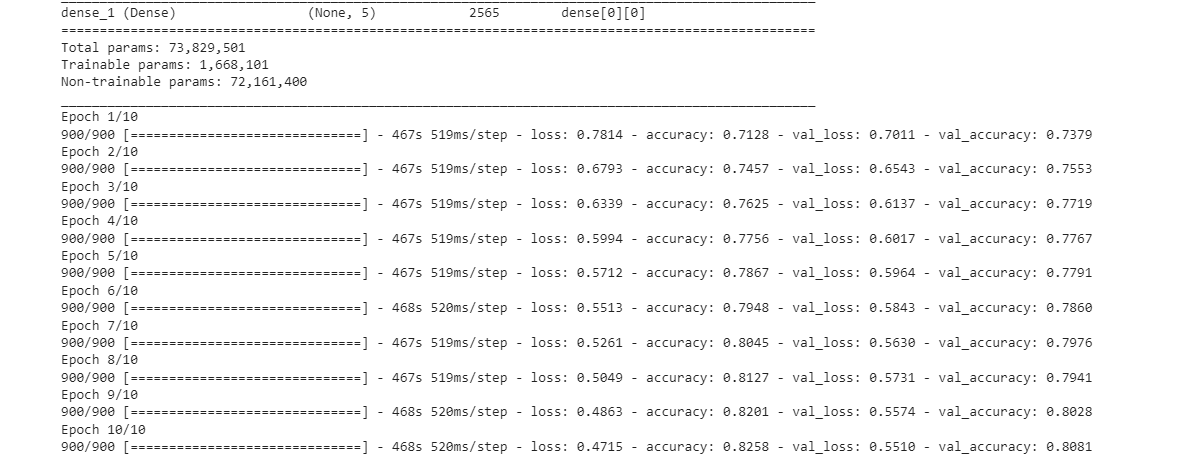


fig 6.2: RNN result

**6.3: EMBEDDING MATRIX**

Got an accuracy of around 82% for this model.





**fig6.3: Embedding Matrix**

**6.4: Result Analysis:**

Accuracy = TP+TN

TP+TN+FP+FN

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

|  |  |  |  |
| --- | --- | --- | --- |
| **RESULTS** | **CNN** | **RNN** | **EMBEDDING MATRIX** |
| ACCURACY | 94% | 93% | 82% |

Fig6.4: Result analysis

From above result analysis we got more accuracy (94%) for CNN for Amazon review sentiment analysis.so consider that Convolutional neural network model is best

**CHAPTER 7: CONCLUSION**

This chapter describes the conclusion within the process while polishing off the research together with future scope which might make more it efficient is discussed further

The proposed method was built with various deep learning algorithms like convolutional neural network Recurrent Neural Net embedding matrix algorithms. Amazon review sentiment analysis supported term frequency-inverse documents frequency is employed for and have extractions on text pre-processing for analysing review of amazon.

On implantations, the analysis results show that this can be an oversized dataset, and therefore the version used here only has the text as a feature with no other metadata. This made this a stimulating dataset for doing NLP work. its data written by users, so it's like there are various types, nonstandard spellings, and other variations that you simply might not find in curated sets of the published text. We did some text processing and so tried out three fairly unoptimized deep learning models:

A convolutional neural network got an accuracy of around 94% for this model. Roc AUC score: 98%

A recurrent neural network Got an accuracy of around 93% for this model. Roc AUC score: 98%.

ROC: Receiver operating characteristic's curve 98% AUC (area under the curve) ROC curve could be a performance measurement for the classification problems at various threshold settings. ROC may be a probability curve and AUC represents the degree or measure of separability. An Embedding matrix Got an accuracy of around 82% for this model. These models achieved results that are within a pair of percent of state-of-the-art at predicting the binary sentiment of the reviews.

**Future Scope**

There are many possible improvements that would be explored to boost accuracy and ROC AUC Score. because of time limitations, the subsequent research work must be performed within the future.

there's a desire for more deep learning algorithms for comparison of, large data set to be trained to create a system or framework for reviewing analysis of the text. we are going to investigate a comprehensive solution, especially for amazon review sentiment analysis application review classifications, supported our previous experimental results. Use more machine learning and deep learning algorithms to structure algorithms which is has more layers than the Embedding matrix which may be learned and make intelligent decisions.

**REFERENCES**

[1]. A Comparison of Sentiment Analysis Methods Reviews of Mobile Phones International Journal of Advanced Computer Science and Applications, Vol. 10, No. 6, 2019,

[2]. Sentiment Analysis on Reviews of Mobile Users, The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014). Lin Zhanga, a School of Computer Science and Engineering, Beihang University.

[3]. Using TF-IDF to Determine Word Relevance in Document Queries 2018. Juan Ramos JURAMOS@EDEN.RUTGERS.EDU Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855.

[4]. Learning from imbalanced data: open challenges and future directions, Prog Artif Intell (2016). Learning from imbalanced data: open challenges and future directions, Prog Artif Intell (2016).

[5]. A review on classification of imbalanced data for wireless sensor networks, First Published April 14, 2020 Harshita Patel, Dharmendra Singh Rajput, SIT& Engineering Vellore.

[6]. Dealing with Data Imbalance in Text Classification, Procedia Computer Science 159 (2019). Cristian Padurariua, Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi.

[7]. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Peter D. Turney Institute for Information Technology National Research Council of Canada Ottawa, Ontario, Canada, K1A 0R6.

[8]. Mining and Summarizing Customer Reviews Mining Hu and Bing Liu Department of Computer Science University of Illinois at Chicago 851 South Morgan Street Chicago, IL 60607-7053.

[9]. Learning Subjective Adjectives from Corpora Janyce M. Wiebe Department of Computer Science New Mexico State University, Las Cruces, NM 88003.

[10]. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean “Distributed representations of words and phrases and their compositionality,”

[11]. Dealing with Data Imbalance in Text Classification. Juan Ramos juramos@eden.rutgers.EDU Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855.

[12]. Review of Text Classification in Deep Learning, AUTHORS: Qi Wang, Wenling Li, Zhezhi Jin

JOURNAL NAME:

[13]. Sentiment Analysis of Chinese Microblog Based on Stacked Bidirectional LSTM JUNHAO ZHOU1 , YUE LU1 , HONG-NING DAI 1 , (Senior Member, IEEE), HAO WANG 2 , (Member, IEEE), AND HONG XIAO3.

[14]. Ranking Online Consumer Reviews March 2018Electronic Commerce Research and Applications 29 DOI:10.1016/j.elerap.2018.03.008.

[15]. Predicting the “helpfulness” of online consumer reviews, Singh, Jyoti Prakash Irani, Seda Rana, Nripendra P. [Journal of Business Research](https://ideas.repec.org/s/eee/jbrese.html), Elsevier, vol. 70(C), pages 346-355.

[16]. Sentiment Analysis Using Text Mining: A Review International Journal on Data Science and Technology Swati Redhu, Volume 4, Issue 2, June 2018, Pages: 49-53 Received: Jun. 25, 2018; Published: Jun. 26, 2018.

[17]. Sentiment analysis of online product reviews using DLMNN and future prediction of online product using IANFIS Authors: P. Sasikala and L. Mary Immaculate Sheela Date: May 19, 2020 From: Journal of Big Data (Vol. 7, Issue 1) Publisher: Springer.