CS 304

SENTIMENTAL ANALYSIS OF PRODUCT REVIEWS

Submitted By:

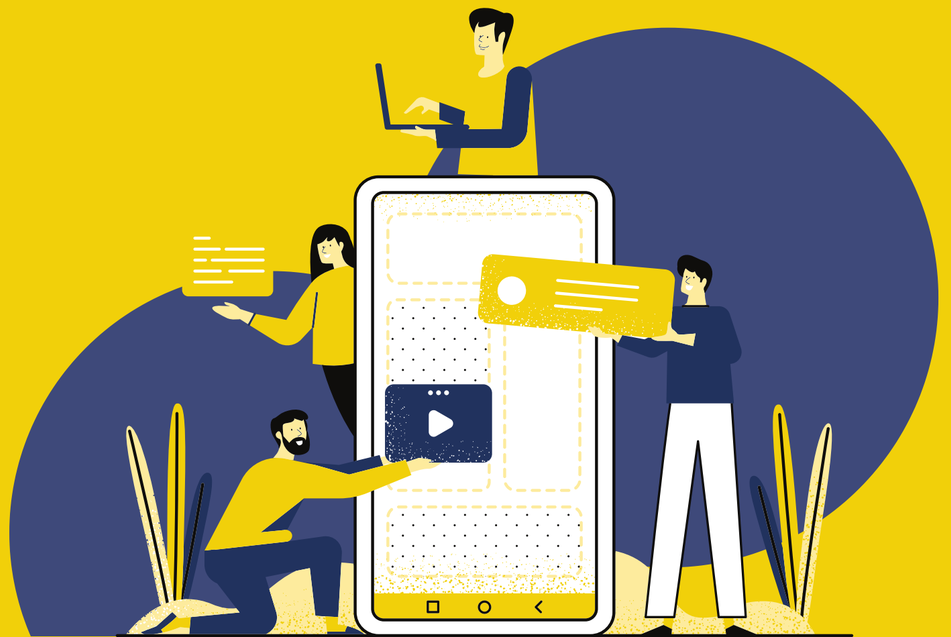
Garvit Galgat 190001016
Kuldeep Singh 190001030
Somya Mehta 190001058

Submitted To:

Dr. Aruna Tiwari
Suchitra Agrawal
Neelesh Ghanghoriya
Saurabh Saini

TABLE OF CONTENT

INTRODUCTION



PRODUCT REVIEW

The promotion of brands to connect with potential customers using the internet and other forms of digital communication. Sentiment analysis of product reviews, an application problem, has recently become very popular in text mining and computational linguistics research. It uses natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and personal information.

In this project, we want to study the correlation between the Amazon product reviews and the rating of the products given by the customers.

PROBLEM STATEMENT

We are having a review provided by a consumer, and we need to do sentiment analysis of this review so that we can conclude a relation between the review provided by the consumer and its ratings.

DATASET ?

We have chosen the following dataset:
https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products

This is a list of over 34,000 consumer reviews for Amazon products like the Kindle, Fire TV Stick, and more provided by Datafiniti's Product Database. This dataset includes:

01 **Basic product information**

02 **Ratings provided by the consumer's**

03 **Review Text**

04 **Date of Review**

05 **Brand**

06 **Category**

07 **Images linked with the review**

08 **Manufacturer Details**

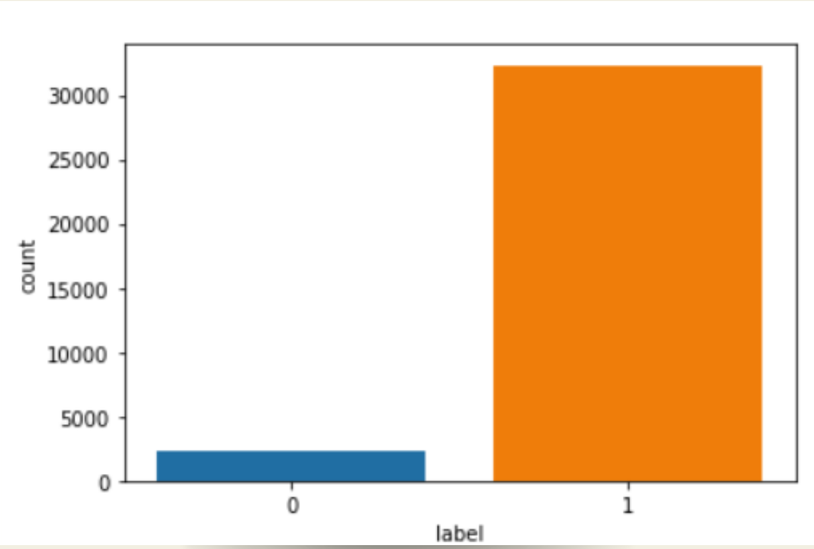
We are planning to use all of these details to create a model which can perform well with any other dataset provided.

DATA PREPROCESSING

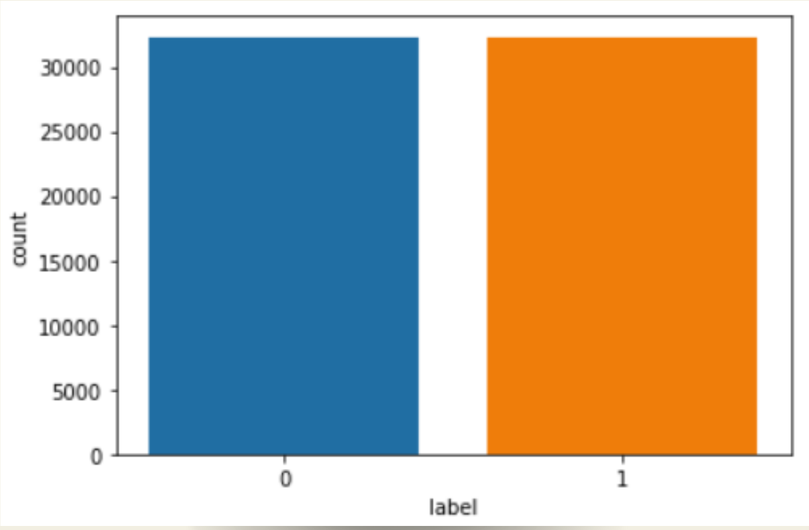
01

Data Resampling:- *We observed that our dataset was not balanced at all so we applied data resampling in order to make our data set more balanced so that our algorithms can produce effective results.*

before



after



DATA PREPROCESSING

02 **Tokenizing Words:-** Converting the words into tokens

03 ***Removing Stop Words:-*** *Stop words are basically a set of commonly used words in any language, not just English.*
The reason why stop words are critical to many applications is that, if we remove the words that are very commonly used in a given language, we can focus on the important words instead.

04 ***Stemming:-*** *Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers.*

Often when searching text for a certain keyword, it helps if the search returns variations of the word. For instance, searching for "boat" might also return "boats" and "boating". Here, "boat" would be the stem for [boat, boater, boating, boats].

Stemming is a somewhat crude method for cataloging related words; it essentially chops off letters from the end until the stem is reached. This works fairly well in most cases.

DATA PREPROCESSING

05

Converted words into bag of words:- *Bag of words is a Natural Language Processing technique of text modelling. In technical terms, we can say that it is a method of feature extraction with text data. This approach is a simple and flexible way of extracting features from documents.*

A bag of words is a representation of text that describes the occurrence of words within a document. We just keep track of word counts and disregard the grammatical details and the word order. It is called a ***"bag" of words*** *because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.*

06

TF-IDF:- *Term frequency works by looking at the frequency of a particular term you are concerned with relative to the document. There are multiple measures, or ways, of defining frequency:*

- *Number of times the word appears in a document (raw count).*
- *Term frequency adjusted for the length of the document (raw count of occurrences divided by number of words in the document).*
- *Logarithmically scaled frequency (e.g. log(1 + raw count)).*
- *Boolean frequency (e.g. 1 if the term occurs, or 0 if the term does not occur, in the document).*

HOW ?

HOW?
OUR APPROACH

FIRST

First of all we will analyze the data and preprocess it to make it suitable for our model. For ex. remove any null values present in any columns or remove any unreachable links present in the images columns.

SECOND

After preprocessing, we will be creating different models specific to algorithms.

THIRD

We will be using the following algorithms:

1. Naive Bayes Analysis
2. KNN Algorithm
3. SVM Algorithm

FORTH

By comparing the results from these different algorithms we can also create a fraud review detection system

MODELS

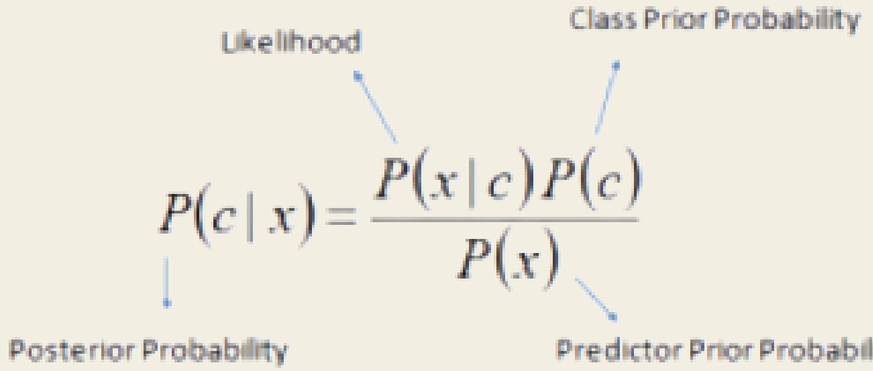
NAIVE BAYES

- *It is a **classification technique based on Bayes' Theorem** with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.*

- *For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is **known as 'Naive'**.*

- ***Naive Bayes model is easy to build and particularly useful for very large data sets.** Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.*

- *Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:*

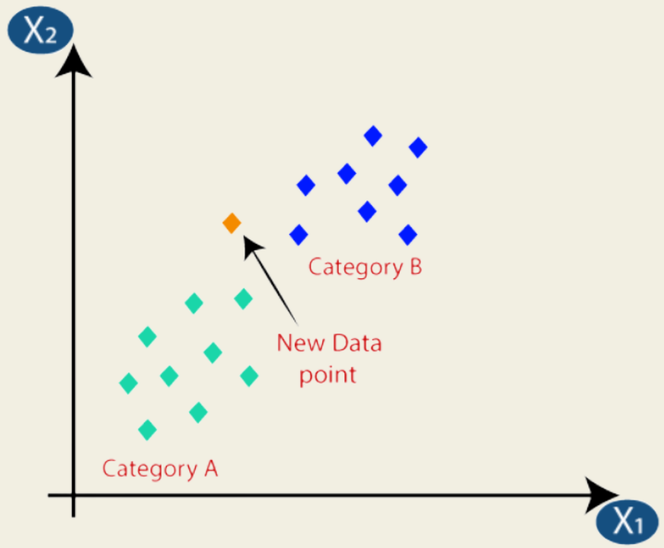


$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

KNN

The K-NN working can be explained on the basis of the below algorithm:

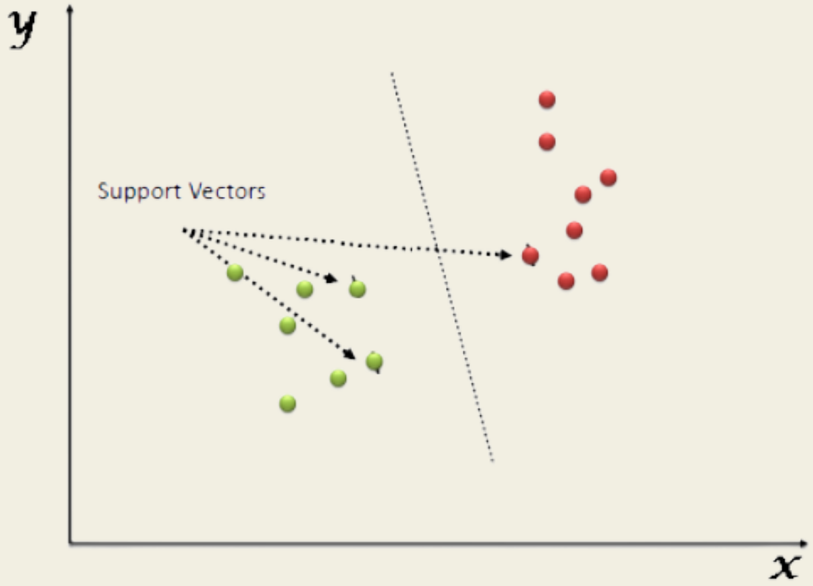
- ***Step-1:*** *Select the number K of the neighbors*
- ***Step-2:*** *Calculate the Euclidean distance of K number of neighbors*
- ***Step-3:*** *Take the K nearest neighbors as per the calculated Euclidean distance.*
- ***Step-4:*** *Among these k neighbors, count the number of the data points in each category.*
- ***Step-5:*** *Assign the new data points to that category for which the number of the neighbor is maximum.*
- ***Step-6:*** *Our model is ready.*



SVM

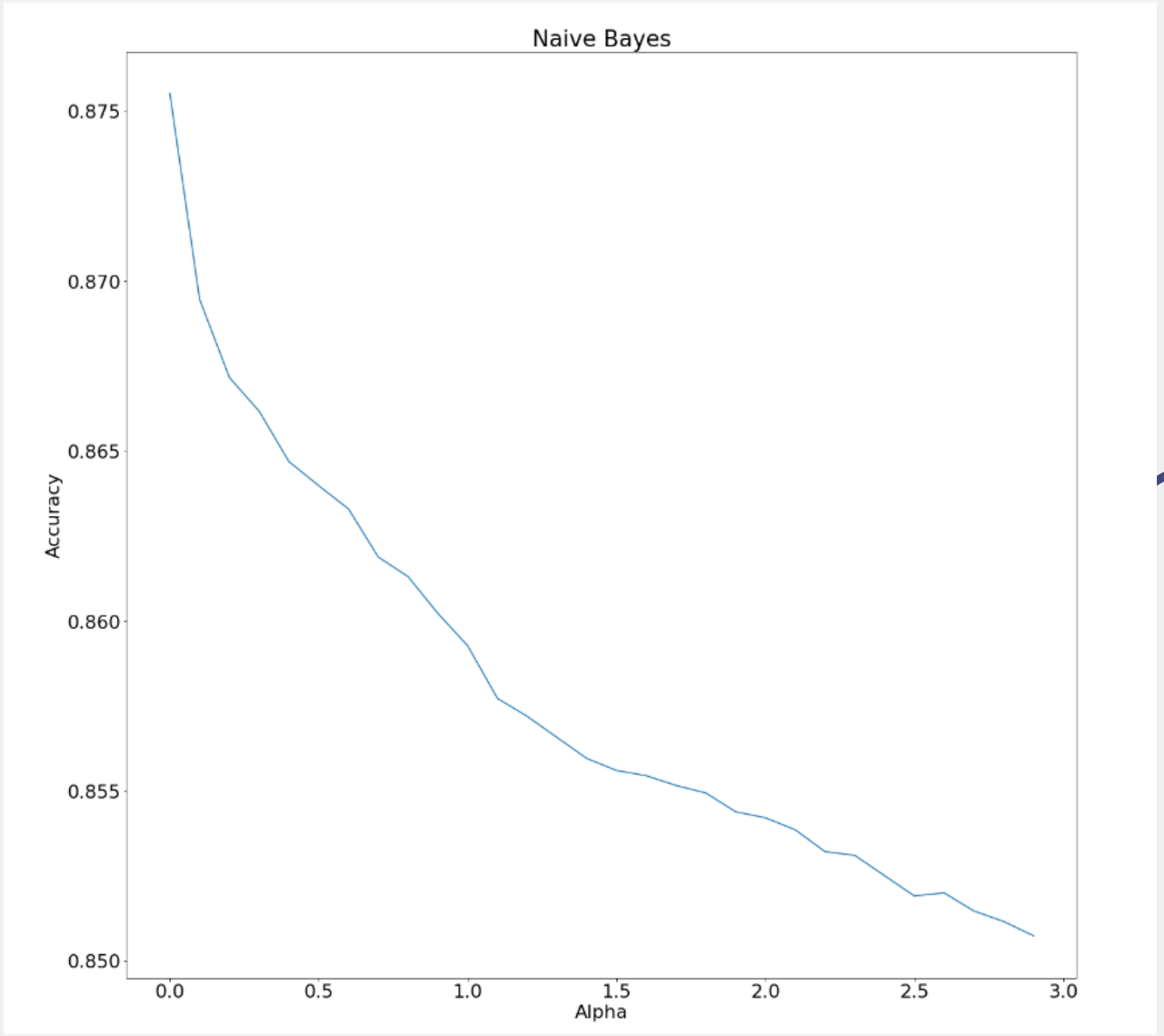
*"**Support Vector Machine**" (**SVM**) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).*

Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line).



EXPERIMENT & RESULTS

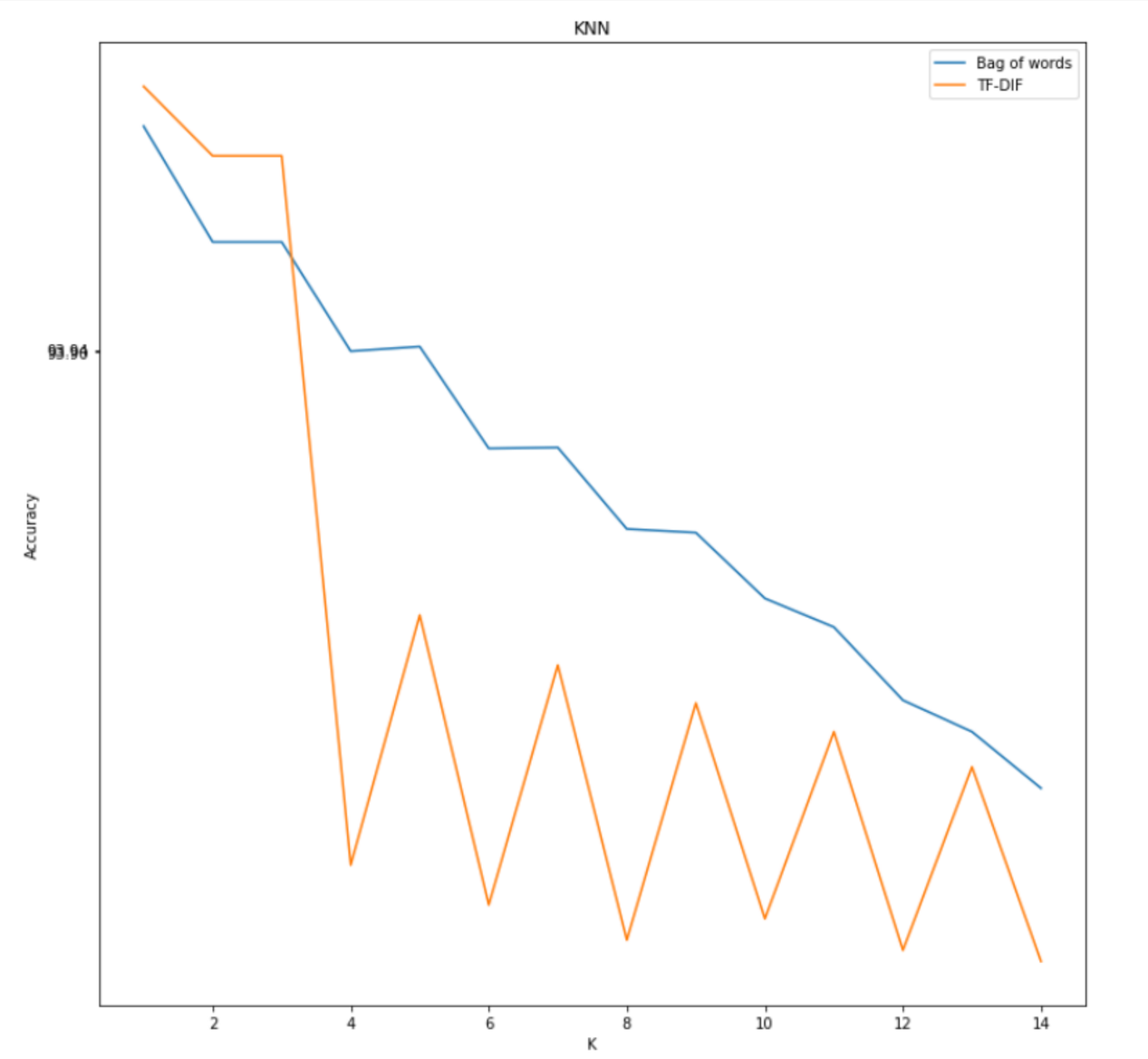
WE HAVE USED DIFFERENT HYPER PARAMETERS FOR OUR ALGORITHM



NAIVE BAYES

We have changed the value of alpha as defined above to get different accuracies of our model.

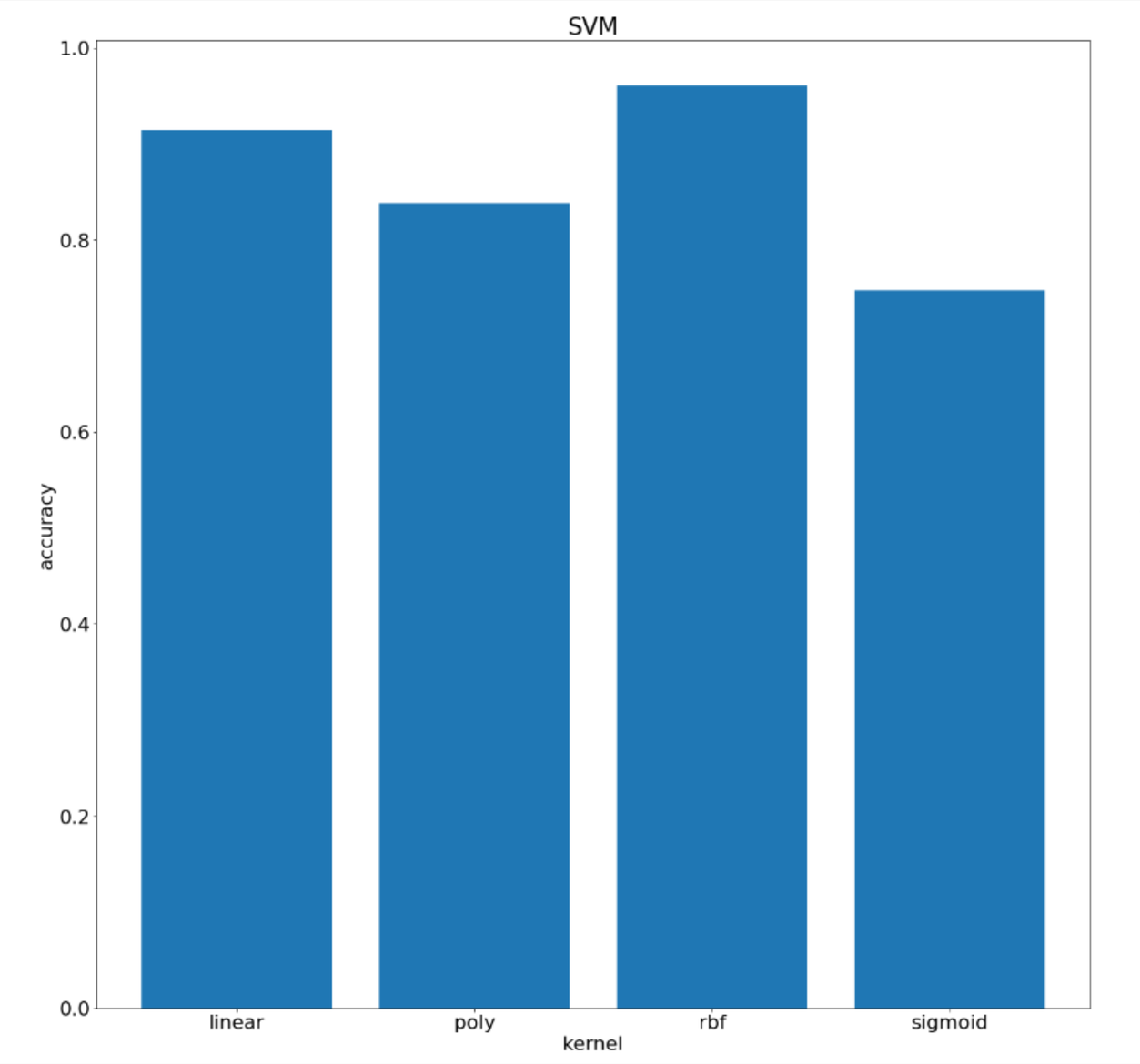
WE HAVE USED DIFFERENT HYPER PARAMETERS FOR OUR ALGORITHM



KNN ALGORITHM

For KNN we we have changed the value of k along with different methods of feature extractions like bag of words and TF-DIF.

WE HAVE USED DIFFERENT HYPER PARAMETERS FOR OUR ALGORITHM



SVM ALGORITHM

For SVM we have used different kernels and got varying accuracies.

PERFORMANCE MATRICES

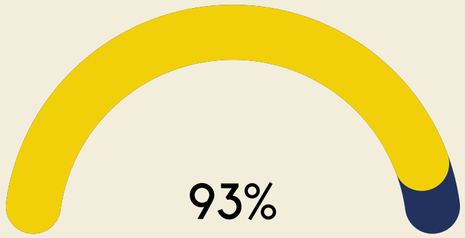
	precision	recall	f1-score	support
NEGATIVE	0.47	0.02	0.04	766
POSITIVE	0.93	1.00	0.97	10661
accuracy			0.93	11427
macro avg	0.70	0.51	0.50	11427
weighted avg	0.90	0.93	0.90	11427

```
[[   17   749]
 [   19 10642]]
0.9327907587293253
```

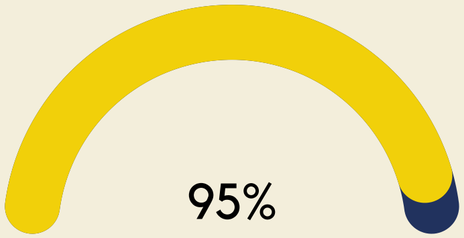
```
[[ 283  439]
 [ 139 6963]]
```

	precision	recall	f1-score	support
1.0	0.67	0.39	0.49	722
5.0	0.94	0.98	0.96	7102
accuracy			0.93	7824
macro avg	0.81	0.69	0.73	7824
weighted avg	0.92	0.93	0.92	7824

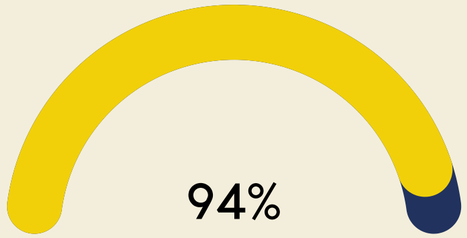
MODELS ACCURACY



Naive Bayes Analysis



KNN Algorithm



SVM Algorithm

Here, we are comparing the accuracy of the models Naive Bayes, KNN and SVM Algorithm.

CONCLUSION

- PRODUCT REVIEW -

We can observe that KNN is the best algorithm for classifying sentiments about reviews because our experiments prove it with the accuracy that we are getting with the help of KNN.

Hence we made an application which can serve people to access about the sentiments of various product's reviews.

Click Here to view

Through this project we got to learn about various applications of several machine learning algorithms such as SVM,KNN and Naive Bayes. We also learnt the importance of Data pre-processing through this project.

THANK YOU

We wish to thank **Dr. Aruna Tiwari, Professor IIT Indore** for his kind support and valuable guidance. We would also like to thank our TAs **Suchitra Agrawal Ma'am**, **Mr. Neelesh Ghanghoriya** and **Mr. Saurabh Saini** for their constant support and guidance throughout the project work.

It is their help and support, due to which we became able to complete the Project on time and also able to make project evaluatioin report.