

# WEB APPLICATION USING TENSORFLOW.JS

Project report submitted in partial fulfillment of the requirement for the degree of  
Bachelor of Technology

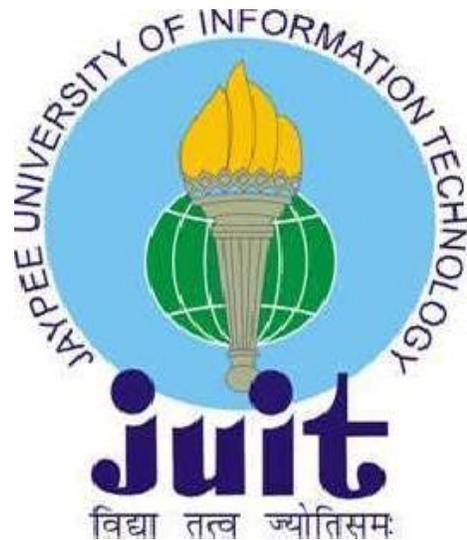
In

**Computer Science and Engineering/Information Technology**

By

Abhishek Agarwal (171212)

Under the supervision  
of Dr. Aman Sharma



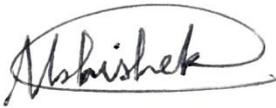
Department of Computer Science & Engineering and  
Information Technology

**Jaypee University of Information Technology**  
**Waknaghat, Solan-173234, Himachal Pradesh**

## Candidate's Declaration

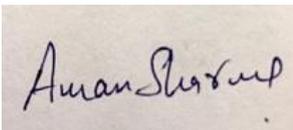
I hereby declare that the work presented in this report entitled “**WEB APPLICATION USING TENSORFLOW.JS**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2021 to May 2021 under the supervision of **Dr. Aman Sharma**, Computer Science and Engineering and Information Technology.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Abhishek Agarwal,171212

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



Dr. Aman Sharma

Computer Science and Engineering and Information

## **Acknowledgement**

I would like to express my special thanks of gratitude to my teacher and mentor **Dr. Aman Sharma** who gave me the golden opportunity to do this project on the topic “**WEB APPLICATION USING TENSORFLOW.JS**”, which also helped me in doing a lot of research and I came to know about so many new things. I am really thankful to him.

Secondly, I would also like to thank Lab assistants who helped me a lot in finalizing this project within the limited time frame.

Abhishek Agarwal, 171212

## Table of Content

<b>1)Abstract.....</b>	<b>(1)</b>
<b>2)Aims and Objectives.....</b>	<b>(2)</b>
<b>3)Chapter1-Introduction.....</b>	<b>(3)</b>
Introduction to the Project.....	(3)
Problem Statement.....	(4)
Traditional Approaches.....	(5)
Organization of Report.....	(7)
<b>4)Chapter 2-Literature Review.....</b>	<b>(8)</b>
Terminologies.....	(8)
Related Work.....	(20)
<b>5)Chapter 3- System Design and Development.....</b>	<b>(22)</b>
System Requirements	
.(22)	
Model Development.....	(26)
<b>6)Chapter 4-Performance Analysis</b>	
..(32)	
<b>7)Chapter5-Conclusion.....</b>	<b>(38)</b>
Future Scope.....	(39)
<b>8)References.....</b>	<b>(40)</b>
)	

## **Abstract**

Humans are social creatures. We like to conform to a community to ensure that we are not left out. But the hectic lifestyle of the modern age demands a tremendous amount of time from an average working human, leaving too little time for leisure. At the same time being a part of society requires them to partake in some sort of common experience like watching movies and discussing them later with their friends. Hence we try to invest what little time we have in watching good movies, not only for entertainment but also for culturally and anthropologically enriching ourselves.

Watching movies as per the liking is always a great experience but finding such movies is a tedious task and we mostly end up watching some different kinds of movies and shows whereas missing out on the right kind of movies as per our taste. By this work of mine, I aim to use sentiment analysis as a tool to correctly predict the right kind of movies for everyone.

This project also aims to be a customer feedback system by simply analysing the sentiments of a customer and help in providing better services next time. It will understand by itself, whether the customer was satisfied with the service and there will be no need of asking for feedback explicitly.

## **Aims and Objectives**

**Aim:-**To train a sentiment analysis algorithm so as to help various platform in improving their recommendation and customer feedback system.

### **Objectives:-**

- The objective of this project was to design a sentiment analysis algorithm for movie lovers to give their detailed opinion about each movie they watched and help in making other's experience better.
- The project covered tasks such as designing a front-end using HTML,CSS,JS and using the power of TensorFlow js to integrate the machine learning algorithm directly into the front-end using a javascript file.
- The project's objective was to serve as a learning experience for us beginners to get familiar with the fascinating field of Natural Language Processing and front-end frameworks.

## **Chapter-1**

### **INTRODUCTION**

#### **(1.1) Introduction to the Project**

This project inspires to become a solution for many platforms which try recommending it's products through reviews and especially OTT platforms like Netflix, Hotstar and Amazon Prime. Because nowadays, movies are not only a form of art but also pivotal to conform to social groups around us. That is why, one always tries to watch good movies in the short leisure time they have. But finding a good review on recent movies is pivotal for that and that is where the Indian movie lovers suffer because most of us find writing a well structured reviews as tedious. As with all things we need something quick and easy. Also, because most of the review platforms are in english, the vernacular movie lovers do not get to express an opinion on a good regional movie as they are mostly ignored and do not get the light they deserve. Furthermore, most of the reviewers on IMDB and rotten tomatoes are high end movie critics whose reviews for a casual movie watcher, may be too harsh to hear and as a result they may be discouraged from watching a good movie.

## **(1.2) Problem Statement**

A huge amount of content to watch is being uploaded on the internet each day on netflix, amazon prime, hotstar and the list is endless and with that is increasing the amount of the consumers. It has become a billion dollar market with ever increasing consumers and thus it has become an important part of everyone's life.

Hence, in this project, I am committed to create a solution that would allow a user to be part of a community that can share their thoughts on the movies they watch via an online platform.

The motivation behind this project comes from various platforms already available such as IMDB and rotten tomato. We felt that there were three major problems with existing solutions and have tried to counter them in our project. They are as follows:

1. Existing solutions like IMDB are server based with limited client operation meaning the movies that can be reviewed by the user are not decided by the user but rather the decision lies with the organization owning these platforms at the server side, meaning that some movies may get ignored by the users even if they are good watch, but with complete client operability, there is lesser chance of missing a good movie.
2. Sites like rotten tomatoes do not take casual movie watcher's review into account rather they rely on journal critics whose judgement may sometimes feel obscure to a less pragmatic audience who look for an entertainment factor rather than an artistic angle.
3. Existing solutions especially IMDB is crammed with information making the review section harder to find and less appealing to technophobic people like old people or small town folk. Also a large percentage of movie buffs in India do not like to sit down to write a complex and structured review but alternatives provided by existing solutions to get feedback from this segment of movie buffs are few and do not convey all the details that can be used to properly score a movie.

### **(1.3) Traditional Approach**

PHP is the most widely used programming and server-side scripting language for the development of web-based applications. In some area, PHP is the best like for developing small applications and for the developers having financial issues. They can easily use this language, which is readily available and develop the application based on their requirements.

But PHP is becoming obsolete day by day because –

- It is not that secure because of its open-source, as the source code can be easily available.
- It is not suitable for large content-based web applications.
- It has a weak type, which may lead to incorrect data and information to the user.
- PHP frameworks need to learn to use the PHP built-in functionalities to avoid writing additional code.
- Using more features of PHP framework and tools lead to poor performance of the web applications.
- PHP do not allow the change or modification in core behavior of the web applications.
- The PHP frameworks are not the same in behavior so does their performance and features.

JavaScript is a lightweight language often mentioned as a client-side scripting language that generally executes in the browsers to make interactive and dynamic web pages. It is widely used for building user-friendly creative web pages as well as non-web projects like game development, mobile apps, PDF and much more. It doesn't require re-compiling, thereby increases the time efficiency. There are also advanced server-side versions of JavaScript like Node.js, and Angular.js, which enables designing websites with more functionalities than just downloading files.

#### **1. Light Weight Scripting Language**

JavaScript is a lightweight scripting language because it is made for data handling at the browser only. Since it is not a general-purpose language so it has a limited set of libraries.

Also as it is only meant for client-side execution and that too for web applications, hence the lightweight nature of JavaScript is a great feature.

## **2. Dynamic Typing**

JavaScript supports dynamic typing which means types of the variable are defined based on the stored value. For example, if you declare a variable x then you can store either a string or a Number type value or an array or an object. This is known as dynamic typing.

To understand this, in languages like Java, we explicitly mention that a particular variable will store a certain type of data, whereas in JavaScript we do not have to provide the data type while declaring a variable. In JavaScript, we just have to use var or let keyword before the variable name to declare a variable without worrying about its type.

## **3. Object-Oriented Programming Support**

Starting from ES6, the concept of class and OOPs has been more refined. Also, in JavaScript, two important principles with OOP in JavaScript are Object Creation patterns (Encapsulation) and Code Reuse patterns (Inheritance). Although JavaScript developers rarely use this feature but its there for everyone to explore.

## **3. Functional Style**

This implies that JavaScript uses a functional approach, even objects are created from the constructor functions and each constructor function represents a unique object-type. Also, functions in JavaScript can be used as objects and can be passed to other functions too.

## **4. Platform Independent**

This implies that JavaScript is platform-independent or we can say it is portable; which simply means that you can simply write the script once and run it anywhere and anytime.

In general, you can write your JavaScript applications and run them on any platform or any browser without affecting the output of the Script. [4]

## **(1.4) Organization of Report**

In Chapter 2, we have discussed the literature review, which consists of all the terminologies, importance and the working of the algorithms and its types. We have described various algorithms and its advantages and disadvantages to identify the best-suited algorithm for our project.

In Chapter 3, we have developed a system design and listed all the system requirements to run these algorithms and the environment on which these algorithms are tested to clearly analyze the data to get better results.

In Chapter 4, we have discussed about all the algorithms used and the mathematics or the formulation behind these algorithms for better understanding and discussed about how all these algorithms are used.

In Chapter 5, we have concluded the report and listed the conclusion learning of the project. In addition, we have discussed about the future scope and the advancements that could be carried out in the project.

## **Chapter-2**

### **LITERATURE REVIEW**

#### **(2.1) Terminologies**

##### **Convolutional Neural Networks**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.[1]

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

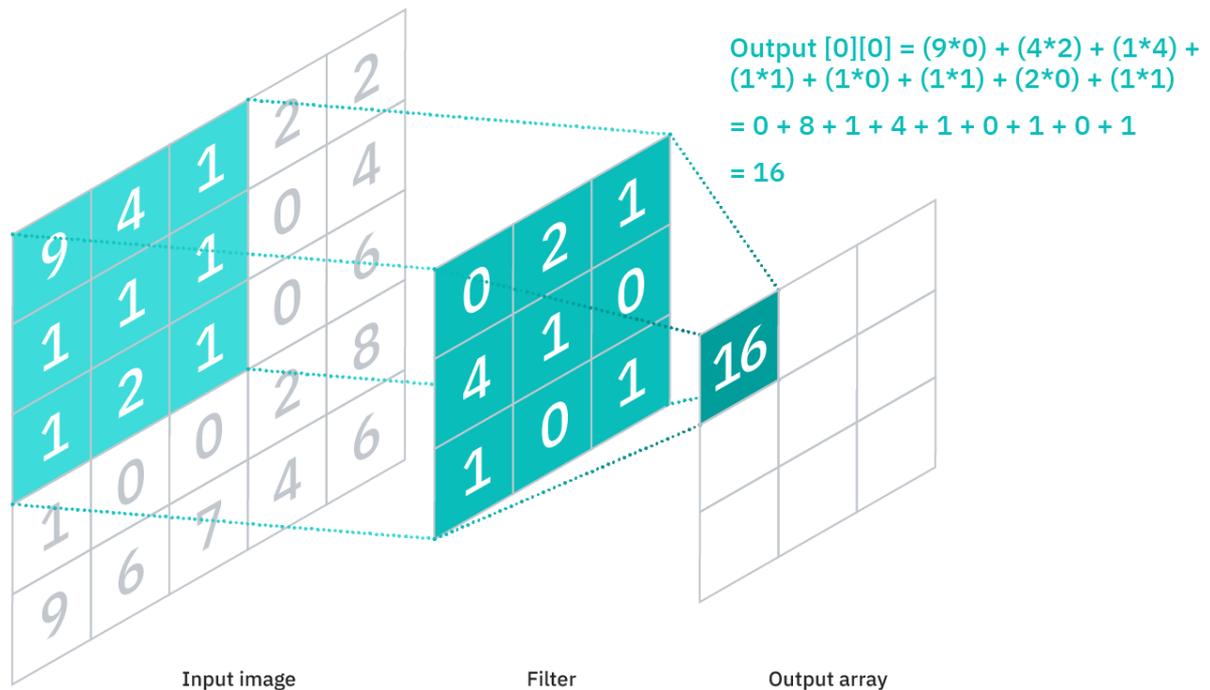
- Convolutional layer
- Pooling layer
- Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

### **Convolutional Layer**

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.



As you can see in the image above, each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as “partially connected” layers. However, this characteristic can also be described as local connectivity.

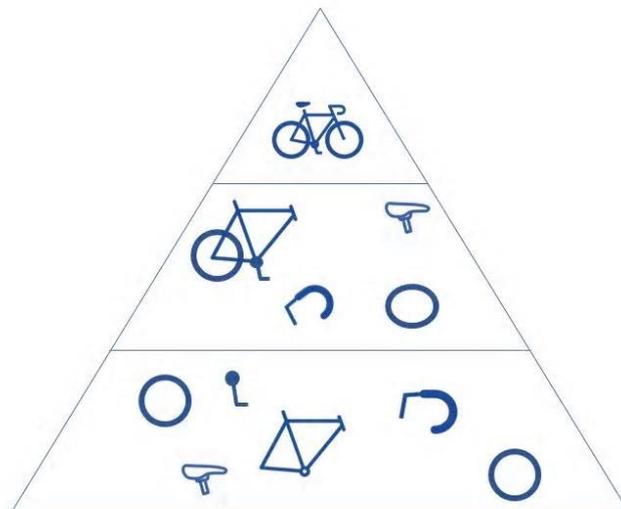
Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.
2. Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.
3. Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:
  - Valid padding: This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.
  - Same padding: This padding ensures that the output layer has the same size as the input layer

- Full padding: This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

As we mentioned earlier, another convolution layer can follow the initial convolution layer. When this happens, the structure of the CNN can become hierarchical as the later layers can see the pixels within the receptive fields of prior layers. As an example, let's assume that we're trying to determine if an image contains a bicycle. You can think of the bicycle as a sum of parts. It is comprised of a frame, handlebars, wheels, pedals, et cetera. Each individual part of the bicycle makes up a lower-level pattern in the neural net, and the combination of its parts represents a higher-level pattern, creating a feature hierarchy within the CNN.



Ultimately, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract relevant patterns.

### **Pooling Layer**

Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

- Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

### **Fully-Connected Layer**

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1. [2]

### **LSTM (Long Short Term Memory)**

When we have a small RNN, we would be able to effectively use the RNN because there is no problem of vanishing gradients. But, when we consider using long RNN's there is not much we could do with the traditional RNN's and hence it wasn't widely used. That is the reason that led to the finding of LSTM's which basically uses a slightly different neuron structure. This was created with one basic thing in mind- the gradients shouldn't vanish even if the sequence is very large.

1. In LSTM, we will be referring to a neuron as a cell. In a traditional RNN, the only way the model can remember something is through updating the hidden states and their respective weights. But, in LSTM this problem is solved by using an explicit

memory unit for learning and remembering tasks. It stores information that is relevant for learning.

2. It also using something called “Gating Mechanism”, which regulates information that the network stores-if it has to pass the information to the next layer or forget the information it has.

3. Constant Error Carousel is another very important characteristic of LSTM. It allows LSTM to have a smooth and uninterrupted flow of gradients while propagation.

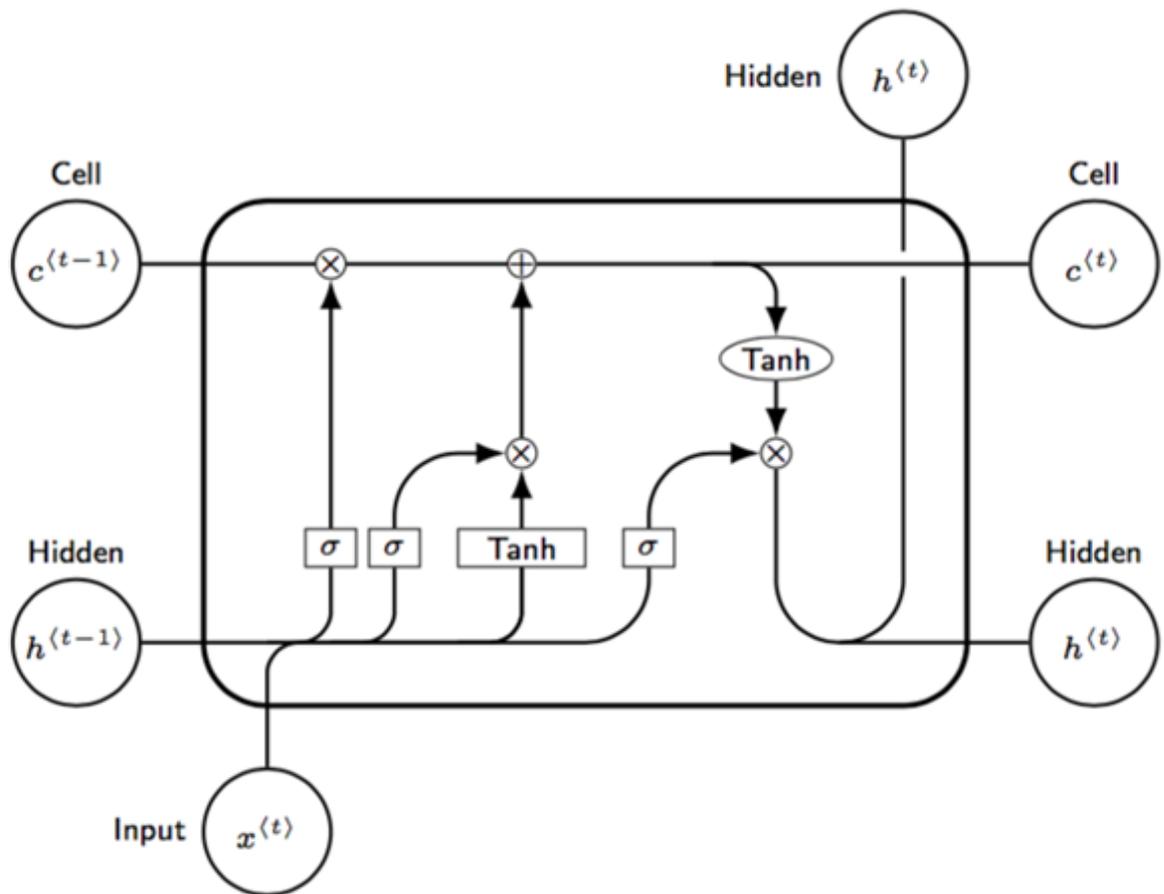


Fig :LSTM (Long Short Term Memory)

1. The big rectangular box is called ‘cell’ which takes an input  $x(t)$  at time  $t$ , a previous hidden layer  $h(t-1)$  and a previous cell state  $c(t-1)$ . The cell state is nothing but the explicit memory unit.

2. The cell gives 2 outputs- 1 is the output of the hidden state  $h(t)$  and the other is the output of the cell state  $c(t)$  at any given time  $t$ .

3. The constant error carousel is responsible for transferring the gradients smoothly from  $c(t-1)$  to  $c(t)$

4. A sigmoid function will output values between 0 and 1, while tanh function will output values between -1 and 1. These are the 2 main activation functions that we will use in LSTM.

5. We combine the inputs from  $x(t)$  and  $h(t-1)$  into a sigmoid activation function and we do a multiplication operation of it with the previous cell state  $c(t-1)$ . This multiplication operation is called 'gate'. If the value from the sigmoid function is close to 1, then the multiplication will lead to a value close to  $c(t-1)$ , that means erase only little of the previous memory but retain most of it. In the contrary, if the sigmoid function is close to 0, then the multiplication will lead to a value that would be close to 0. This means erase almost everything from the previous cell state (memory). This whole part is called as 'Forget Gate'

6. The next gate is called 'Update Gate' which uses a sigmoid and a tanh function, which will both have a multiplication gate followed by an addition gate with output from 'Forget Gate'. The 'tanh' function controls how much to increase or decrease the value of the next cell state. The sigmoid function decides how much information should it write to the new cell state  $c(t)$ .

7. The next and last gate is called 'Output Gate'. This will have a sigmoid function followed by a multiplication gate with a tanh activation function, thus releasing values to hidden state for both feed forward and recurrent sides. Here, the higher the value of the sigmoid function and tanh function, the higher will be the value transmitted to the next hidden state  $h(t)$ .

8. In LSTM, you can see that all the 3 sigmoid and 1 tanh activation functions for which the input would be a concatenation of  $h(t-1)$  and  $x(t)$ , has different weights associated with them, say  $w(f), w(i), w(c)$  and  $w(o)$ . Then the total parameters required for

training an LSTM model is 4 times larger than a normal RNN. So, the computational cost is extremely higher. To solve this problem, they invented something called GRU.[3]

## **Word embedding**

Word embedding plays a critical role in the realization of transfer learning in NLP. The intuition behind word embeddings is that words are represented as low-dimensional vectors that capture both the syntax and semantics of the text corpus. Words with similar meanings tend to occur in similar context.

The word representations are learned by exploiting vast amounts of text corpora. A popular implementation of word embeddings is the Word2Vec model which has two training options—Continuous Bag of Words and the Skip-gram model. Word embeddings are often used as the first data processing layer in a deep learning model.

One limitation of standard word embedding techniques such as Word2Vec, fasttext, and Glove is that they aren't able to better disambiguate between the correct sense of a given word. In other words, each instance of a given word ends up having the same representation regardless of the context in which it appears.

Recently, contextual word embeddings such as Embeddings from Language Models (ELMo) and Bidirectional Encoder Representations from Transformers (BERT) have emerged. These techniques generate embeddings for a word based on the context in which the word appears, thus generating slightly different embeddings for each of word's occurrence.

ELMo uses a combination of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. On the other hand, BERT representations are jointly conditioned on both the left and right context and use the Transformer, a neural network architecture based on a self-attention mechanism. The Transformer has been shown to have superior performance in modeling long-term dependencies in the text, compared to recurrent neural network architecture.

The integration of the contextual word embeddings into neural architectures has led to consistent improvements in important NLP tasks such as sentiment analysis, question answering, reading comprehension, textual entailment, semantic role labeling, coreference resolution, or dependency parsing.

Language model embeddings can be used as features in a target model or a language model can be fine-tuned on target task data. Training a model on a large-scale dataset and then fine-tuning the Pre-trained model for a target task (transfer learning, if you'll recall), can particularly be beneficial to low-resource languages where labeled data is limited.

## **Debiasing**

Debiasing is a process through which the influence of cognitive biases is reduced, generally with the goal of helping people think in a more rational and optimal manner.

Debiasing is usually accomplished through the use of various debiasing techniques, that can work on any number and type of cognitive biases. For example, when it comes to reducing cognitive biases that revolve our tendency to underestimate how different other people's views are from our own, one useful debiasing technique is to visualize how a certain situation looks from someone else's perspective.

Research shows that cognitive debiasing works in some cases, meaning that the use of appropriate training, interventions, and debiasing techniques can reduce some cognitive biases, to some degree, in some situations. For example, one study on the topic found that even a single training session, in the form of playing an instructional computer game or watching an educational video, improved people's ability to reduce various cognitive biases in the long term, months after the training.[5]

Effective cognitive debiasing generally occurs through a process, rather than a single event. In general, this debiasing process involves the following stages:

**Awareness.** Become aware that a certain bias might be triggered or has been triggered already.

**Decision.** Decide to take action to reduce the bias.

**Analysis.** Figure out when, where, why, and how the bias in question is likely to occur.

**Planning.** Based on the information from the analysis stage, identify the optimal debiasing approach to use, and figure out how exactly it will be implemented.

**Action.** Implement the planned debiasing approach.

## **TensorFlow.js**

TensorFlow.js, an open-source library you can use to define, train, and run machine learning models entirely in the browser, using Javascript and a high-level layers API. If you're a Javascript developer who's new to ML, TensorFlow.js is a great way to begin learning.

### **In-Browser ML**

Running machine learning programs entirely client-side in the browser unlocks new opportunities, like interactive ML. ML running in the browser means that from a user's perspective, there's no need to install any libraries or drivers. Just open a webpage, and your program is ready to run. In addition, it's ready to run with GPU acceleration. TensorFlow.js automatically supports WebGL, and will accelerate your code behind the scenes when a GPU is available. Users may also open your webpage from a mobile device, in which case your model can take advantage of sensor data, say from a gyroscope or accelerometer. Finally, all data stays on the client, making TensorFlow.js useful for low-latency inference, as well as for privacy preserving applications.

If you're developing with TensorFlow.js, here are three workflows you can consider.

1. You can import an existing, pre-trained model for inference. If you have an existing TensorFlow or Keras model you've previously trained offline, you can convert into TensorFlow.js format, and load it into the browser for inference.

2. You can re-train an imported model. You can use transfer learning to augment an existing model trained offline using a small amount of data collected in the browser using a technique called Image Retraining. This is one way to train an accurate model quickly, using only a small amount of data.
3. Author models directly in browser. You can also use TensorFlow.js to define, train, and run models entirely in the browser using Javascript and a high-level layers API. If you're familiar with Keras, the high-level layers API should feel familiar.

Thus, using JavaScript and frameworks like Tensorflow.js is a great way to get started and learn more about machine learning. Even though a production-ready application should probably be built in a language like Python, JavaScript makes it really accessible for developers to play around with the different features and get a better understanding of the fundamental concepts before eventually moving on and investing time into learning another language. [7]

## **HTML, CSS, JavaScript**

HTML and CSS are actually not technically programming languages; they're just page structure and style information as they are on the front end of every web page and application.

### **HTML**

HyperText Markup Language (HTML) can be broken down into HyperText, which is what grants access to other texts through links, and Markup which outlines the basic structure and appearance of raw text. What this means is that HTML describes and defines the content and basic structure of the website. It does this through a means of special tags or codes which tell the browser what to do. HTML is the bare basics of a website.

An HTML only website can be compared to a functioning human body. An HTML only website has all of its body parts, although it doesn't offer much to look at because it doesn't have any accessories or personal style. At this stage, it's also a body which is not capable of moving or speaking.

This is where CSS comes in.

## **CSS**

A Cascading Style Sheet is the website's accessories. It's responsible for outlining the colours, font and positioning of the content on a website. It adds some style and structure to the content. In order to make use of the CSS capabilities it needs to be linked within the HTML content so that style can be added to the website. CSS will tell the browser how to display the existing HTML.

CSS can be compared to adding personal style to the body. When you link CSS to HTML, it's like dressing up the body. For example, you can choose a specific colour shirt and match it with a specific colour pair of trousers. On a website, you can choose the colour of the background or the font size of a heading and much more. It's important to note that CSS cannot live without HTML as there would be nothing to style. Just like clothes or shoes would be pointless without someone to wear them.

However, you can't help but notice that something is missing. The web page is lacking certain functions like a search box or options to comment. Right now the body, with all its accessories, looks more like a mannequin in a store window than a real human being. That's where JavaScript comes in.

## **JavaScript**

JavaScript controls the behaviour of the website. It's important to note that JavaScript and Java are two different things. JavaScript was designed to manipulate web pages and it is used to create interactive functionality. Without JavaScript a website will still be functional, but in a limited way. JavaScript is what animates HTML and CSS, and it's what brings your website to life.

JavaScript can be compared to the body's ability to perform actions such as walking or talking. So when you add JavaScript to HTML and CSS, it transforms the body from being a beautifully dressed mannequin into a real-life walking talking human being. It animates the body, giving it lifelike qualities. JavaScript can also be compared to a fully functional body that has the ability to interact. As we all know, having an interactive website is critical, otherwise its just a boring page filled with information. Here we see a website which consists of HTML, CSS and JavaScript:

If you look at this example of twitter, JavaScript allows you to expand the tweet to see re-tweets, to set a tweet as a favourite and more. A popular JavaScript App is Google Maps. [6]

## **(2.2) Related Work**

Information about millions of films and television programs as well as their cast and crew. The name is an acronym for Internet Movie Database. As a wholly owned subsidiary of Amazon.com, IMDb is based in Seattle, but the office of Col Needham, the founder and CEO, remains in Bristol, England, where the Web site was founded. Needham, an English software engineer and film buff, began what became IMDb with a list of all the films he had seen since 1980. The stated founding date of IMDb is October 17, 1990, when Needham posted his movie-listing software to a USENET film discussion group. The site was cooperatively expanded and became an early migrant to the World Wide Web. IMDb.com was incorporated in January 1996 and was purchased by Amazon.com two years later. Amazon, an online bookseller then in the process of expanding its product line, intended to use IMDb as a platform for sales of videotapes and DVDs. (In the 21st century many films became electronically accessible directly from the IMDb site through Amazon Video.) But even after becoming an Amazon subsidiary, IMDb retained its identity and considerable autonomy. Needham stayed on as chief executive.

The IMDb site grew beyond its original purpose of indexing credits in film and TV productions. An entry for a film may now cover studios and other companies associated with the film, release dates in various countries, censor classifications, box-office grosses, awards won, and other information. The entries for performers, directors, writers, and others involved in film and TV productions are also comprehensive. Some entries include photos, videos, and coming-attraction trailers.

Rotten Tomatoes was launched on August 12, 1998, as a spare-time project by Senh Duong. His objective in creating Rotten Tomatoes was "to create a site where people can get access to reviews from a variety of critics in the U.S." As a fan of Jackie Chan, Duong was inspired to create the website after collecting all the reviews of Chan's Hong Kong action movies as they were being released in the United States. The catalyst for the creation of the website was *Rush Hour* (1998), Chan's first major Hollywood crossover, which was originally planned to release in August 1998. Duong coded the website in two weeks and the site went live the same month, but the release of *Rush Hour* was delayed until September 1998. Besides Jackie Chan films, he began including other films on Rotten Tomatoes, extending it beyond Chan's fandom. The first non-Chan Hollywood movie whose reviews were featured on Rotten Tomatoes was *Your Friends & Neighbors* (1998). The website was an immediate success, receiving mentions by Netscape, Yahoo!, and USA Today within the first week of its launch; it attracted "600–1000 daily unique visitors" as a result.[9]

## **Chapter-3**

### **SYSTEM DESIGN AND DEVELOPMENT**

#### **(3.1) System Requirements**

The algorithms that are being implemented in this project requires some generic system as it requires the processing of algorithms.

1. Windows 10 (64-bit)
2. Visual Studio Code
3. JavaScript

4. 8 GB RAM
5. Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz

### Why Python

Python is a programming language with a large audience and it is very easy to understand and can be easily readable. Furthermore, python offers the collection of packages that makes the most intimidating algorithms or projects simpler. Python has libraries for almost every usable file i.e. - with working with images, working with text, or working with audio files. Even when working with a new OS, python is very malleable. Python has a large community which makes it easier to seek help and tips and tricks.

### Why ANACONDA

ANACONDA is widely popular as it provides all the libraries pre-installed and makes the user free from the hassle of otherwise installing all libraries. It has around 100 packages that can be used for data science, machine learning, or statistical analysis.

### Why Tensorflow

TensorFlow provides excellent functionalities and services when compared to other popular deep learning frameworks. These high-level operations are essential for carrying out complex parallel computations and for building advanced neural network models. TensorFlow is a low-level library that provides more flexibility.

### Why Keras

Keras, is a high-level neural networks library which is running on the top of TensorFlow, CNTK, and Theano. Using Keras in deep learning allows for easy and fast prototyping as well as running seamlessly on CPU and GPU. This framework is written in Python code which is easy to debug and allows ease for extensibility. The main advantages of Keras are described below:

- **User-Friendly:**Keras has a simple, consistent interface optimized for common use cases which provides clear and actionable feedback for user errors.
- **Modular and Composable:**Keras models are made by connecting configurable building blocks together, with few restrictions.
- **Easy To Extend:** With the help of Keras, you can easily write custom building blocks for new ideas and researches.
- **Easy To Use:**Keras offers consistent & simple APIs which helps in minimizing the number of user actions required for common use cases, also it provides clear and actionable feedback upon user error.

## Why HTML, CSS, JAVASCRIPT

**HTML** provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.HTML is at the core of every web page, regardless the complexity of a site or number of technologies involved. It's an essential skill for any web professional. It's the starting point for anyone learning how to create content for the web.

**CSS** is used to control presentation, formatting, and layout.CSS stands for Cascading Style Sheets. This programming language dictates how the HTML elements of a website should actually appear on the frontend of the page.This language affects the entire mood and tone of a web page, making it an incredibly powerful tool -- and an important skill for web developers to learn.

**JavaScript** is used to control the behavior of different elements.JavaScript is supported by all modern web browsers and is used on almost every site on the web for more powerful and complex functionality.JavaScript is a programming language that lets web

developers design interactive sites. Most of the dynamic behavior you'll see on a web page is thanks to JavaScript, which augments a browser's default controls and behaviors.

## Why Bootstrap

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project. Some of its features -

- Faster and Easier Web-Development.
- It creates Platform-independent web-pages.
- It creates Responsive Web-pages.
- It is designed to be responsive to mobile devices too.
- It is Free! Available on [www.getbootstrap.com](http://www.getbootstrap.com)

## Why Visual Studio Code

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs. [8]

## **(3.2) Model Development**

### **Why Experimental model development?**

Analysis by experiment was the most enlightening way to understand the project requirements and fulfilling them. Most of the work in this project is focussed on a better User Experience and User Interaction. Through this section, we would like to explain our experimentation with the Framework and the product it yielded.

### **Setting up the project and the backend**

The prerequisites are three major softwares. They are:

1. Node.js
2. TensorFlow.js
3. Visual Studio Code

Node.js is responsible for containing all the background libraries, for example, Bootstrap, JQuery, FontAwesome, etc. so they need not be included in our code again and again and are a part of the project instead of being included through a URL in the index.html file. We can also install or upgrade additional tools or components using node.js

TensorFlow.js is a JavaScript library developed by Google for training and using machine learning (ML) models in the browser. It's a companion library to TensorFlow, a popular ML library for Python.

Visual Studio Code is the editor we went with for this project. It is an industry-standard editor that offers easy version control with git. It also has IntelliSense for HTML which allows writing a large amount of code with relative ease.

## Dataset

The IMDb Movie Reviews dataset is a binary sentiment analysis dataset consisting of 25,000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. Only highly polarizing reviews are considered. A negative review has a score  $\leq 4$  out of 10, and a positive review has a score  $\geq 7$  out of 10. No more than 30 reviews are included per movie. The dataset contains additional unlabeled data.

Review	Sentiment
no such thing is sort of a minimalist beauty and the beast , but in this case th '. .	negative
not for everyone , but for those with whom it will connect , it's a nice departu '. .	positive
it virtually defines a comedy that's strongly mediocre , with funny bits surfaci '. .	negative
meandering and confusing .	negative
if you can keep your eyes open amid all the blood and gore , you'll see del toro '. .	positive

## Tokenization

Language in its original form cannot be accurately processed by a machine, so you need to process the language to make it easier for the machine to understand. The first part of making sense of the data is through a process called tokenization, or splitting strings into smaller parts called tokens.

A token is a sequence of characters in text that serves as a unit. Based on how you create the tokens, they may consist of words, emoticons, hashtags, links, or even individual characters. A basic way of breaking language into tokens is by splitting the text based on whitespace and punctuation.

## **Normalization**

Words have different forms—for instance, “ran”, “runs”, and “running” are various forms of the same verb, “run”. Depending on the requirement of your analysis, all of these versions may need to be converted to the same form, “run”. Normalization in NLP is the process of converting a word to its canonical form.

Normalization helps group together words with the same meaning but different forms. Without normalization, “ran”, “runs”, and “running” would be treated as different words, even though you may want them to be treated as the same word. In this section, you explore stemming and lemmatization, which are two popular techniques of normalization.

Stemming is a process of removing affixes from a word. Stemming, working with only simple verb forms, is a heuristic process that removes the ends of words.

I will use the process of lemmatization, which normalizes a word with the context of vocabulary and morphological analysis of words in text. The lemmatization algorithm analyzes the structure of the word and its context to convert it to a normalized form. Therefore, it comes at a cost of speed. A comparison of stemming and lemmatization ultimately comes down to a trade off between speed and accuracy.

## **Removal of noise from data**

In this step, I will remove noise from the dataset. Noise is any part of the text that does not add meaning or information to data.

Noise is specific to each project, so what constitutes noise in one project may not be in a different project. For instance, the most common words in a language are called stop words. Some examples of stop words are “is”, “the”, and “a”. They are generally irrelevant when processing language, unless a specific use case warrants their inclusion.

I will use ‘regular expressions’ in Python to search for and remove these items:

**Hyperlinks** - All hyperlinks in Twitter are converted to the URL shortener t.co. Therefore, keeping them in the text processing would not add any value to the analysis.

**Twitter handles in replies** - These Twitter usernames are preceded by a @ symbol, which does not convey any meaning.

**Punctuation and special characters** - While these often provide context to textual data, this context is often difficult to process. For simplicity, you will remove all punctuation and special characters from tweets.

To remove hyperlinks, you need to first search for a substring that matches a URL starting with http:// or https://, followed by letters, numbers, or special characters. Once a pattern is matched, the .sub() method replaces it with an empty string.

Since we will normalize word forms within the remove\_noise() function, you can comment out the lemmatize\_sentence() function from the script.

Similarly, to remove @ mentions, we substitute the relevant part of text using regular expressions. The code uses the ‘re’ library to search @ symbols, followed by numbers, letters, or \_, and replaces them with an empty string.

Finally, I can remove punctuation using the library ‘string’.

In addition to this, I will also remove stop words using a built-in set of stop words in NLTK, which needs to be downloaded separately.

There are certain issues that might arise during the preprocessing of text. For instance, words without spaces (“iLoveYou”) will be treated as one and it can be difficult to separate such words. Furthermore, “Hi”, “Hii”, and “Hiiiiii” will be treated differently by the script unless you write something specific to tackle the issue. It’s common to fine tune the noise removal process for your specific data.

## **Determining Word Density**

The most basic form of analysis on textual data is to take out the word frequency. A single review is too small of an entity to find out the distribution of words, hence, the analysis of the frequency of words would be done on all positive reviews.

## **Preparing Data for the Model**

Sentiment analysis is a process of identifying an attitude of the author on a topic that is being written about. You will create a training data set to train a model. It is a supervised learning machine learning process, which requires you to associate each dataset with a “sentiment” for training. In this tutorial, your model will use the “positive” and “negative” sentiments.

Sentiment analysis can be used to categorize text into a variety of sentiments. For simplicity and availability of the training dataset, this tutorial helps you train your model in only two categories, positive and negative.

A model is a description of a system using rules and equations. It may be as simple as an equation which predicts the weight of a person, given their height. A sentiment analysis model that you will build would associate tweets with a positive or a negative sentiment. You will need to split your dataset into two parts. The purpose of the first part is to build the model, whereas the next part tests the performance of the model.

In the data preparation step, you will prepare the data for sentiment analysis by converting tokens to the dictionary form and then split the data for training and testing purposes.

### **Converting Tokens to a Dictionary**

First, you will prepare the data to be fed into the model. You will use the Naive Bayes classifier in NLTK to perform the modeling exercise. Notice that the model requires not just a list of words in a tweet, but a Python dictionary with words as keys and True as values. The following function makes a generator function to change the format of the cleaned data.

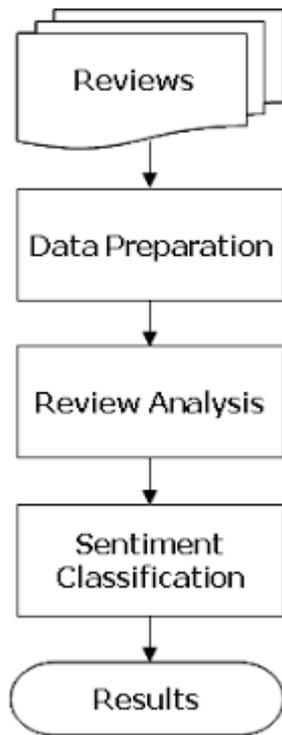
### **Splitting the Dataset for Training and Testing the Model**

Next, you need to prepare the data for training the NaiveBayesClassifier class. The code splits the shuffled data into a ratio of 70:30 for training and testing, respectively.

### **Building and Testing the Model**

Finally, I can use the NaiveBayesClassifier class to build the model. Use the `.train()` method to train the model and the `.accuracy()` method to test the model on the testing data.

In this step I built and tested the model. I also explored some of its limitations, such as not detecting sarcasm in particular examples.



Steps in a summarized way

## **Chapter-4**

### **PERFORMANCE ANALYSIS**

**Why JavaScript performs better than PHP**

- Being a client-side framework, making the application faster
- Easier programming as it has its own forms, variables, loops etc.
- Continuous improvements.
- Powerful, reliable, capacity to handle complex coding with ease
- Cheaper as doesn't involve much manual coding
- Ease of managing lengthy codes with proper directories
- Directly involves HTML through directives
- Better for Single Page Applications
- Clearly separates back end from the front end, hence easier to maintain
- Facilitates better design patterns, less coding, fewer things to test, fewer bugs

### **The Responsiveness of my project:**

Responsive web design (RWD) is a web development approach that creates dynamic changes to the appearance of a website, depending on the screen size and orientation of the device being used to view it. RWD is one approach to the problem of designing for the multitude of devices available to customers, ranging from tiny phones to huge desktop monitors.

RWD uses so-called breakpoints to determine how the layout of a site will appear: one design is used above a breakpoint and another design is applied below that breakpoint. The breakpoints are commonly based on the width of the browser.

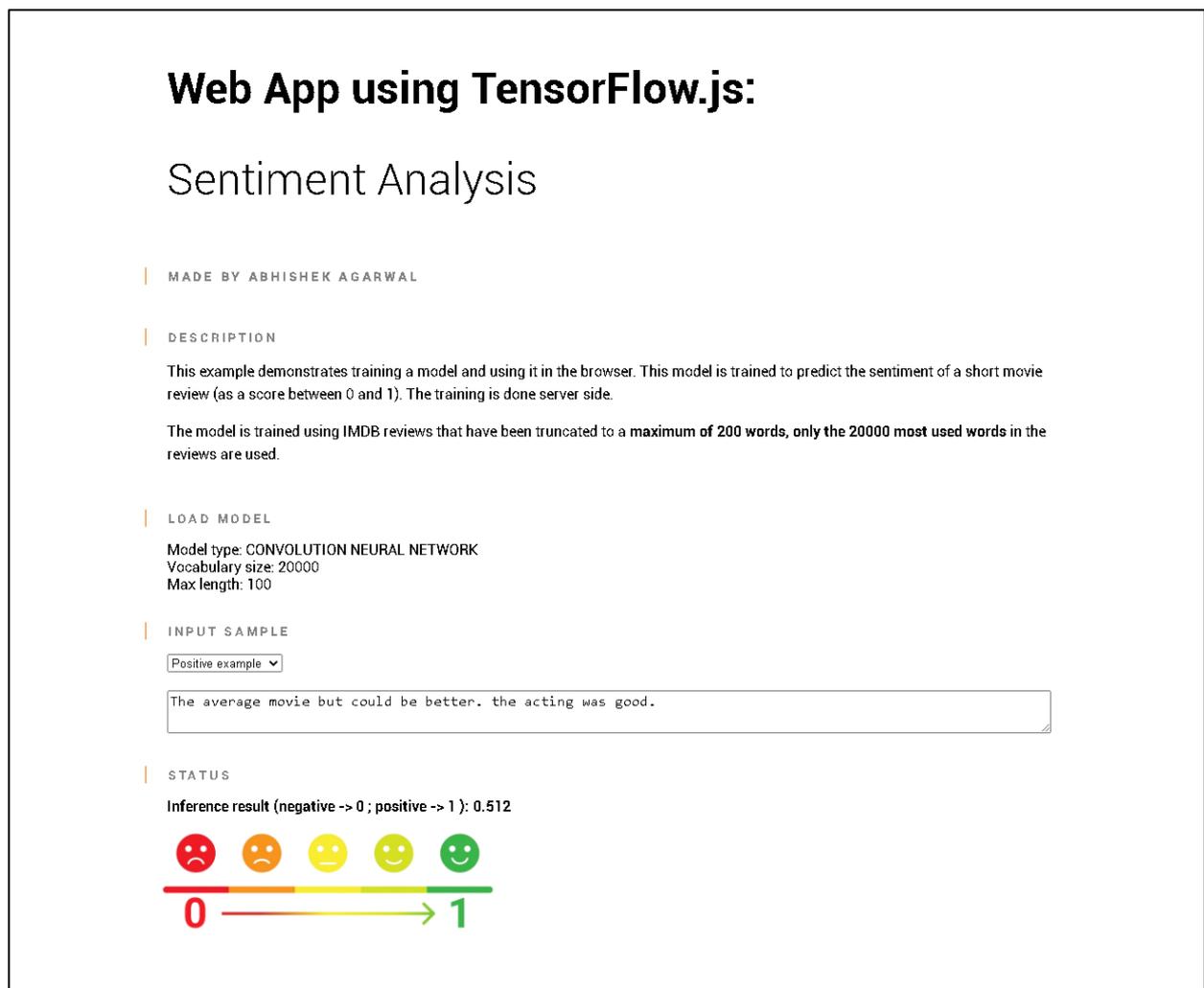
The same HTML is served to all devices, using CSS (which determines the layout of webpage) to change the appearance of the page. Rather than creating a separate site and corresponding codebase for wide-screen monitors, desktops, laptops, tablets and phones of all sizes, a single codebase can support users with differently sized viewports.

In responsive design, page elements reshuffle as the viewport grows or shrinks. A three-column desktop design may reshuffle to two columns for a tablet and a single column for a smartphone. Responsive design relies on proportion-based grids to rearrange content and design elements.

While responsive design emerged as a way to provide equal access to information regardless of device, it is also possible to hide certain items. Decisions about hiding content and functionality

or altering appearance for different device types should be based on knowledge about your users and their needs.

I have gone to great lengths to ensure a smooth user experience for the user in our project. I have made my application responsive for Desktop, mobile and Tablet views. Down Below I present a few screenshots to demonstrate what I mean:



This is the standard desktop view

# Web App using TensorFlow.js:

## Sentiment Analysis

MADE BY ABHISHEK AGARWAL

### DESCRIPTION

This example demonstrates training a model and using it in the browser. This model is trained to predict the sentiment of a short movie review (as a score between 0 and 1). The training is done server side.

The model is trained using IMDB reviews that have been truncated to a **maximum of 200 words**, only the **20000 most used words** in the reviews are used.

### LOAD MODEL

Model type: CONVOLUTION NEURAL NETWORK  
Vocabulary size: 20000  
Max length: 100

### INPUT SAMPLE

Positive example ▾

The average movie but could be better.  
the acting was good.



### STATUS

Inference result (negative -> 0 ; positive -> 1 ): 0.512



This is the standard mobile view

## Results at various stages

After completely training the model and testing the project with various inputs to check if sentiments were being analysed correctly, I took the screenshot of the results at various stages to demonstrate how my project is performing. And I am very happy to attach those screenshots below to see for yourself.

### Stage 1

**DESCRIPTION**

This example demonstrates training a model and using it in the browser. This model is trained to predict the sentiment review (as a score between 0 and 1). The training is done server side.

The model is trained using IMDB reviews that have been truncated to a **maximum of 200 words, only the 20000 most reviews are used.**

**LOAD MODEL**

Model type: CONVOLUTION NEURAL NETWORK  
Vocabulary size: 20000  
Max length: 100

**INPUT SAMPLE**

Positive example ▾

One of the best movie of all times...

**STATUS**

Inference result (negative -> 0 ; positive -> 1): 1.000



Analyzed as a positive review (1.000) for a positive feedback

## Stage 2

**DESCRIPTION**

This example demonstrates training a model and using it in the browser. This model is trained to predict the sentiment review (as a score between 0 and 1). The training is done server side.

The model is trained using IMDB reviews that have been truncated to a **maximum of 200 words, only the 20000 most reviews** are used.

**LOAD MODEL**

Model type: CONVOLUTION NEURAL NETWORK  
Vocabulary size: 20000  
Max length: 100

**INPUT SAMPLE**

Positive example ▾

average movie. Could be better. Acting was amazing.

**STATUS**

Inference result (negative -> 0 ; positive -> 1 ): 0.836



0 → 1

Analyzed as an average review (0.836) for an average feedback

## Stage 3

**DESCRIPTION**

This example demonstrates training a model and using it in the browser. This model is trained to predict the sentiment review (as a score between 0 and 1). The training is done server side.

The model is trained using IMDB reviews that have been truncated to a **maximum of 200 words**, only the **20000 most reviews** are used.

**LOAD MODEL**

Model type: CONVOLUTION NEURAL NETWORK  
Vocabulary size: 20000  
Max length: 100

**INPUT SAMPLE**

Positive example ▾

Bad movie. Lost the interest in first hour.

**STATUS**

Inference result (negative -> 0 ; positive -> 1 ): 0.000



Analyzed as a negative review (0.000) for a negative feedback

Thus it can be concluded that the algorithm was able to correctly analyze the sentiments and was giving the values as expected, ranging from 0 as being extremely negative to 1 being extremely positive. The result is capped to making predictions upto 3 decimal places.

I also added an illustrative image at the bottom of the project for easy understanding for the users as images are easy for humans to understand.

## **Chapter-5**

# **CONCLUSION**

### **(5.1) Conclusion**

Throughout this project I have experimented with various types of components to figure out which would be effective in a certain user scenario. I have learned to use the functionality provided by TensorFlow.js to make an organized project which is easy to upgrade in the future.

I explored a lot about this powerful tool and evenly powerful community which is very active and really helpful. TensorFlow.js makes integrating Machine Learning with the front-end, really easy and opens up many possibilities where Machine Learning needs the sensors available on the smartphones like Gyroscope as using TensorFlow.js which is a JavaScript file allows a developer to access those sensors.

Furthermore, I also got to understand about the front-end part, and how does it interact with the back-end using JavaScript. I understood the importance for a responsive design for better user experience as our users could be a desktop user, or a phone user.

Lastly, I experimented with various forms of visual feedback and how they play a role in effective user experience. Armed with these insights I am now confident to tackle similar problems and is excited to learn about more advanced ones.

## **(5.2) Future Scope**

- This project can now be integrated with any platform which takes user's reviews and wants to know how is the users experiencing their platform overall.
- It can now also help in recommendation system for those platforms where they have a vast category of content and wants their user to consume the content that's ment for them.
- Sentiment Analysis can also help in product analysis and thus a company can get an insight of how their product will perform in the market once it's launched. And this project is very much appropriate for that purpose.
- This project can be useful in Social media monitoring as well. Now-a-days we see that there are many negative words being spread through social media which has a bad impact on society. Using this project we can track such accounts or pages and report them.
- Using the 'Speech to Text' algorithm through Machine Learning, this project can have whole new applications which can help an organisation to analyze the sentiments of their users through speech which will be converted to text and then analyzed for sentiments.
- Last but not the least, it can also work as a tool to suggest the content writer that how the sentiments of his overall content feels. As a matter of fact, 'Grammarly' is a good application of this tool.

## **REFERENCES**

[1]. Convolutional Neural Network. Online available at:

<https://www.ibm.com/cloud/learn/convolutional-neural-networks>

[2]. Convolutional Neural Network. Online available at:

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks>

[3]. Long Short-Term Memory. Online available at:

<https://medium.com/@madhuramiah/bi-directional-rnn-basics-of-lstm-and-gru-e114aa4779bb>

[4]. JavaScript features. Online available at:

<https://www.studytonight.com/javascript/javascript-features>

[5]. Debiasing. Online available at:

<https://effectiviology.com/cognitive-debiasing-how-to-debias>

[6]. HTML,CSS,JavaScript. Online available at:

<https://www.itonlinelearning.com/blog/how-do-html-css-and-javascript-work-together>

[7]. TensorFlow.js. Online available at:

<https://medium.com/tensorflow/introducing-tensorflow-js-machine-learning-in-javascript-bf3eab376db>

[8]. Visual Studio Code. Online available at:

<https://code.visualstudio.com/docs/editor/whyvscode>

[9]. Rotten Tomatoes History. Online available at:

[https://en.wikipedia.org/wiki/Rotten\\_Tomatoes](https://en.wikipedia.org/wiki/Rotten_Tomatoes)

jhkjhkhkj

ORIGINALITY REPORT

---

---

**1** %  
SIMILARITY INDEX

**1**%  
INTERNET SOURCES

**1**%  
PUBLICATIONS

**0**%  
STUDENT PAPERS

---

PRIMARY SOURCES

---

**1** [www.mwnews.am](http://www.mwnews.am) 1%  
Internet Source

---

**2** [hdl.handle.net](http://hdl.handle.net) <1%  
Internet Source

---

---

Excludequotes      On Excludebibliography      On  
Excludematches      Off

