

**Sentiment Analysis on Amazon Reviews**

Vinit Shah

ITCS-5156

11/28/2021

## **Introduction**

In today's online world where customer service has become the highest priority, companies are becoming more and more interested in analyzing and gathering insights on customer feedback to improve or enhance their products or services. As the unstructured customer feedback data collected by these companies are often in very high volumes, they can't analyze it manually without introducing any bias. To enhance the decision-making process, the sentiment analysis on data can help uncover valuable insights about the customer's opinions, emotions, and attitudes towards the specific product or service.

Sentiment analysis is a subcategory in Natural Language Processing (NLP) that tries to identify and extract the customer attitudes from different sources like social media, blogs, product reviews, online forums, etc. Machine learning models accompanied with NLP techniques makes it possible to examine the unstructured data and develop a decision-making process while still preserving its sequential order, and complex logic. In this paper, several different machine learning algorithms and NLP-based methods are used utilized to examine customer sentiments. Machine learning models like Naïve Bayes, Support Vector Machine (SVM), Decision Tree, Logistic Regression, and deep learning models like Long Short-Term Memory (LSTM) have been experimented with.

## **Problem Statement**

To build a model that can perform sentiment analysis on the Amazon customer review dataset to categorize whether the product review is positive or negative using machine learning models accompanied by Natural Language Processing (NLP).

## **Motivation**

Customer feedbacks are the most predominant aspect for the e-commerce companies like Amazon to better serve their customer. Hence, it is fundamental for the suppliers and resellers to understand the opinions and emotions which customers share. In the big data world, where there is a high volume of unstructured data is generated every second, Natural Language Processing and Machine Learning could help provide critical insights for the companies directly by their customers. Using the knowledge gathered from the sentiment analysis, the companies can expand their business by identifying the gaps and enhancing them to better match the customer demands.

The main goal of this project is to discover insights from the customer reviews and identify the patterns within the text to analyze whether a customer is expressing a negative attitude towards the product or is trying to positively recommend the product based on their experience.

## Challenges

The major challenge which I faced in this project was training the big dataset for the LSTM model. To preprocess and train the models on the dataset, my local system was not sufficient due to the CPU and GPU constraints so, after testing the code on a few different environments, I found the Kaggle to be a better alternative to train and validate different parameters for the LSTM model.

Another challenge that I faced while training the LSTM model was the issue of underfitting. To reduce the overfitting, I tried several different approaches like increasing the size of the dataset, I tried increasing the size of the dataset, reducing the size of the features extracted from the tokenization, trying regularization of the text data through different NLP techniques, etc. but I was still not able to achieve the optimal results from the LSTM model as I was expecting from the research paper which was able to achieve 93% of accuracy.

## Concise Summary of the Approach

The key goal of this paper is to develop the sentiment analysis model which would allow the company to classify whether a particular customer review is positive or negative. To develop the model, the raw unstructured data is preprocessed through a series of NLP techniques like removing stop words, applying lemmatization, applying tokenization, etc. Once the data is cleaned, several different Machine Learning models are tested:

- Gaussian Naïve Bayes
- Multinomial Naïve Bayes
- Support Vector Machine (SVM)
- Decision Tree Classifier
- Logistic Regression
- Long Short-Term Memory (LSTM) model

After applying the models, several performance metrics like accuracy, precision, recall, F1-measure, and AUI score will be considered and analyzed to determine the model which best fits the dataset.

## Backgrounds/Related Work

In the paper by Kamal Hassan, Mohan, et al, the Naïve Bayes algorithm, and semantic decision tree is used to classify the polarity of comments on the e-commerce websites (2017). The researchers developed a web crawler to fetch the consumer comments from the e-commerce web pages. After extracting the comments, they performed the spelling correction on the comments using the WordNet dictionary to understand the polarity of the words. They also performed stemming and stop words removal on the dataset as a part of preprocessing step. Afterward, they applied the Naïve Bayes algorithm and calculated the overall polarity of the positive and negative words using the decision tree. Since their problem was an unsupervised learning model, they were able to attain 73% accuracy using the Naïve Bayes model. According to the authors, it can be difficult to perform for the unsupervised models as the correlation for the same clusters and actual features is not the same.

In the paper by Rathor, Abhilasha Singh, et al., several different machine learning models like Naive Bayes (NB) Classifier, Maximum Entropy (ME) Classifier, and Support Vector Machines (SVM) classifier was tested (2018). SVM model attained maximum accuracy on the Amazon products review dataset. They normalized the data by removing the motivations from the data, removing usernames, removing stop-words. They computed unigrams and weighted unigrams for their feature vectors. After developing the feature vectors, they tested their model on a keyword-based model where the positive/good and negative/bad keyword list and for each review, they counted the number of positive, negative, and neutral reviews. The polarity of the highest count is returned by the classifier. They then applied the data to Naive Bayes (NB) Classifier, Maximum Entropy (ME) Classifier, and Support Vector Machines (SVM) classifiers. From their experiments, they found that the weighted unigrams performed better than the unigrams, and the SVM model performed the best compared to the other algorithms.

In the paper by Shrestha, Nishit, and Fatma Nasoz, the recurrent neural networks (RNN) model with the gated recurrent unit (GRU) was developed (2019). The authors first converted the product reviews to fixed-length feature vectors using paragraph vectors. These feature vectors are grouped by product and sorted in chronological order. Each group is then trained on RNN with GRU. The vectors generated in the RNN capture important information like product qualities and temporal relations among reviews. The product embeddings are concatenated with the fixed-length vectors generated by the paragraph vectors and it is then trained on a support vector machine (SVM) model to classify positive and negative samples.

The authors in this paper preprocessed the data by removing hyperlinks, removing extra spaces, formalizing the words, and treating punctuations as separate tokens to try to improve the accuracy of the classifier. Using the paragraph vectors, they converted 3.5 million product reviews to 300-dimensional fixed-length vectors. Then they grouped the reviews by product id and ordered them by review time to compute the product embeddings. The generated temporally sorted vectors are the input sequence and their corresponding ratings are the targeted output which is used to train a gated recurrent unit (GRU) to learn embeddings for that particular product.

The product embeddings for all products are saved in a file and then the embedding for the given product id is retrieved from the file and concatenated with review embedding to form the final 428-dimensional vector. This vector is then applied to the SVM model to predict a sentiment class.

In the paper by K. K. Mohbey, Long Short-Term Model (LSTM) model is developed to predict the customer review. This approach has received higher accuracy when compared to several other techniques like Naïve Bayes, Support Vector Machine (SVM), Decision Tree, and Logistic Regression.

Comparing the above paper with the other 3 papers, K. K. Mohbey's model has attained the highest accuracy of 93.66%. To evaluate the performance, he has used precision, recall, accuracy, AUC, and f-measure score. Shrestha, Nishit, and Fatma Nasoz's model trained on review embedding and product embedding gained an average precision score of 59%, recall score of 42%, and classification accuracy of 81%. The model trained by Rathor, Abhilasha Singh, et al. received an accuracy of 73%.

From the above papers, it can be seen that the Deep Neural network models like Long Short-Term Memory (LSTM) for the sentiment analysis have been proved to be more effective and robust (with higher precision) than other Machine learning algorithms. Due to the issue of the vanishing gradient problem which is solved by the LSTM model, it can achieve the highest accuracy compared to the other traditional classification algorithms.

In the paper by K. K. Mohbey, he used LSTM deep learning model on the product review dataset to gain perception about product quality and performance (2021). He tested several different classification algorithms like Naïve Bayes, Support Vector Machine (SVM), Decision Tree, and Logistic Regression model. From this experimentation, his LSTM model attained an accuracy of 93.6%, F1-score of 95%, and AUC of 97.4% which significantly surpassed all the existing models. To preprocess the data, he normalized the data by removing unwanted characters and symbols. He eliminated the stop words and then developed the tokenization to split the character sequences into tokens. Afterward, the vectorization is performed to convert text into numeric form for easy and fast processing.

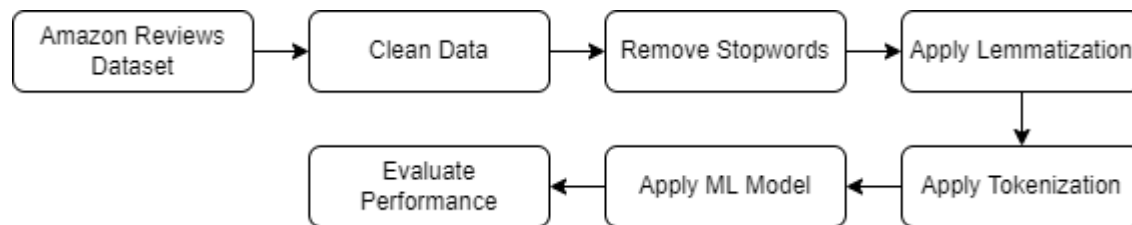
The negative part of the above papers was that only in one research paper, the researchers provided the details regarding the hyperparameter which were used for their LSTM model. All three papers have provided very limited information on the hyperparameters which were used to train their models.

The papers discussed above have implemented various approaches on customer review datasets. Though each of these approaches have their advantages and disadvantages, the literature survey provided a good summary and evaluation of these approaches about the sentiment analysis. Most of the papers have utilized almost similar preprocessing steps on the text data to ensure that the data is normalized and ready for application on the machine learning models.

## Method

### Model Flow:

Below is the flow diagram of the project which is utilized for testing various machine learning algorithms on the amazon reviews dataset.



### Data Introduction

This original dataset consisted of the Amazon customer reviews text and star ratings in the format which is suitable for training on the fastText model. So, to make the data format readable by the panda's library, the data were first preprocessed to convert it into the below format:

	text	label
0	stuning even for the non-gamer: this sound track was beautiful! it paints the senery in your mind so well i would recomed it even to people who hate vid. game music! i have played the game chrono cross but out of all of the games i have ever played it has the best music! it backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras. it would impress anyone who cares to listen! ^_ ^	0
1	the best soundtrack ever to anything.: i'm reading a lot of reviews saying that this is the best 'game soundtrack' and i figured that i'd write a review to disagree a bit. this in my opinino is yasanori mitsuda's ultimate masterpiece. the music is timeless and i'm been listening to it for years now and its beauty simply refuses to fade.the price tag on this is pretty staggering i must say, but if you are going to buy any cd for this much money, this is the only one that i feel would be worth every penny.	0
2	amazing!: this soundtrack is my favorite music of all time, hands down. the intense sadness of "prisoners of fate" (which means all the more if you've played the game) and the hope in "a distant promise" and "girl who stole the star" have been an important inspiration to me personally throughout my teen years. the higher energy tracks like "chrono cross ~ time's scar~", "time of the dreamwatch", and "chronomantique" (indefinably remeniscent of chrono trigger) are all absolutely superb as well.this soundtrack is amazing music, probably the best of this composer's work (i haven't heard the xenogears soundtrack, so i can't say for sure), and even if you've never played the game, it would be worth twice the price to buy it.i wish i could give it 6 stars.	0
3	excellent soundtrack: i truly like this soundtrack and i enjoy video game music. i have played this game and most of the music on here i enjoy and it's truly relaxing and peaceful.on disk one. my favorites are scars of time, between life and death, forest of illusion, fortress of ancient dragons, lost fragment, and drowned valley.disk two: the draggons, galdorb - home, chronomantique, prisoners of fate, gale, and my girlfriend likes zelbessdisk three: the best of the three. garden of god, chronopolis, fates, jellyfish sea, burning orphange, dragon's prayer, tower of stars, dragon god, and radical dreamers - unstealable jewel.overall, this is a excellent soundtrack and should be brought by those that like video game music.xander cross	0
4	remember, pull your jaw off the floor after hearing it: if you've played the game, you know how divine the music is! every single song tells a story of the game, it's that good! the greatest songs are without a doubt, chrono cross: time's scar, magical dreamers: the wind, the stars, and the sea and radical dreamers: unstolen jewel. (translation varies) this music is perfect if you ask me, the best it can be. yasanori mitsuda just poured his heart on and wrote it down on paper.	0

### Data Preprocessing

In this process, the amazon reviews dataset is first fetched from the data source and converted into the standard format. Then the data is further cleaned by converting the text to lowercase, removing punctuation, URLs, HTML tags, and line breaks, removing emojis, etc.

In the below function, there are several preprocessing setups implemented like converting the data to lowercase, removing URLs, removing HTML elements, removing numbers, line breaks, and punctuations.

```
def clean_text(data):
    """ Normalize words and remove unwanted symbols and punctuation marks """
    # Convert data to lowercase
    data = str(data).lower()
    # Remove punctuation
    data = re.sub(r"[^\w\s]", "", data)
    # Remove URL
    data = re.sub(r"^https?:\/\/.*[\r\n]*", "", data, flags=re.MULTILINE)
    # Remove HTML
    data = re.sub("<[^\>+?>", "", data)
    # Remove numbers
    data = re.sub("^d+|s|s\d+|s|s\d+$", " ", data)
    # Remove line breaks
    data = re.sub("\n", "", data)
    # Remove punctuation
    translator = str.maketrans("", "", string.punctuation)
    return data.translate(translator)
```

Since several two-letter words were there in the model, the words with lengths lesser than 2 were also eliminated. In the below code, the stop words like “a”, “the”, “is”, etc. are filtered out from the sentences since it is not useful in the context and provide no significant value in the sentiment analysis. Once the stop words are removed, the lemmatization is applied to the text to normalize the words by extracting the word to its base form.

```
def remove_stopwords(text):
    """ Eliminates stop words """
    text = [word.lower() for word in text.split() if word.lower() not in stop]
    return " ".join(text)
```

```
from nltk.corpus import wordnet

def get_wordnet_pos(word):
    """Maps POS tag to lemmatizer """
    tag = nltk.pos_tag([word])[0][1][0].upper()
    tag_dict = {
        "J": wordnet.ADJ,
        "N": wordnet.NOUN,
        "V": wordnet.VERB,
        "R": wordnet.ADV
    }
    return tag_dict.get(tag, wordnet.NOUN)

def apply_lemmatization_with_pos_tag(sentence):
    """ Apply lemmatization with POS tag """
    lemmatizer = WordNetLemmatizer()
    lemma_sent = [lemmatizer.lemmatize(word, get_wordnet_pos(word)) for word in word_tokenize(sentence)]
    return " ".join(lemma_sent)
```

After applying the lemmatization, the tokenizer is used to tokenize the words and convert the text to sequences.

```

MAX_FEATURES = 4500
INPUT_LENGTH = 256
DIMENSIONS = 64

tokenizer = Tokenizer(num_words=MAX_FEATURES)

tokenizer.fit_on_texts(train_df['clean_text'])
tokenizer.fit_on_texts(test_df['clean_text'])

train_token = tokenizer.texts_to_sequences(train_df["clean_text"])
test_token = tokenizer.texts_to_sequences(test_df["clean_text"])

X_train = pad_sequences(train_token, maxlen=INPUT_LENGTH)
X_test = pad_sequences(test_token, maxlen=INPUT_LENGTH)

```

### **Models Creation:**

After the preprocessing and tokenization step, the data is ready for applying the Machine learning models. To determine the best performing model, several different models were developed and tested:

#### **Naïve Bayes Model:**

The naïve Bayes model computes the posterior probability for each class and assigns the class to the sample with maximum posterior probability.

#### **Support Vector Machine (SVM) Model:**

The main goal of the support vector machine algorithm is to find a hyperplane in an n-dimensional space that can distinctly classify the positive and negative reviews. SVM can classify both linear and nonlinear data. For the given data, SVM tries to find a hyperplane as the decision surface such that the separation between positive and negative reviews is maximized. If the data is non-linear, then the SVM model transforms the data into a higher dimension and then solves the problem by finding a linear hyper-plane.

#### **Decision Tree Classifier:**

A decision tree classifier is a nonparametric approach that does not rely on probability distribution assumptions but rather produces a sequence of rules to classify the class of the reviews.

#### **Logistic Regression:**

Logistic regression interprets the relationship between more or more independent variables. The main goal of this algorithm is to find the best fitting model that can describe the relationship between the dependent and the independent variables.

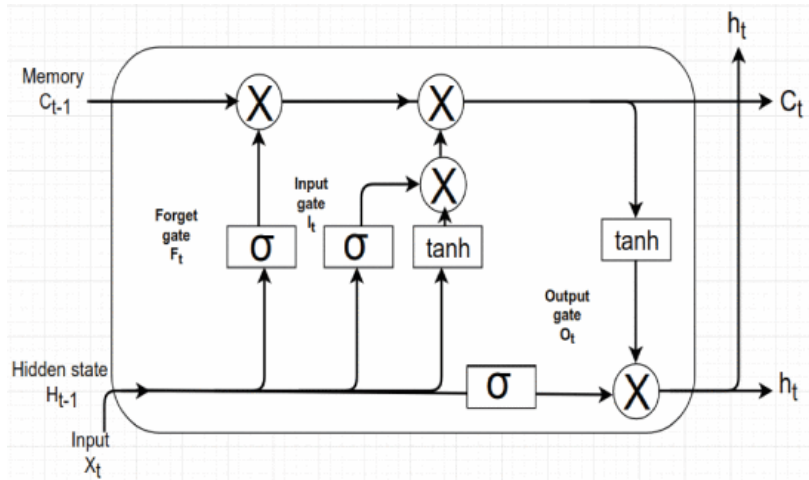
#### **Long Short-Term Memory (LSTM):**

Long Short-Term Memory (LSTM) networks are a sub-category of recurrent neural networks. It is designed to avoid the vanishing gradient problem which recurrent neural networks face. There are three important parts of the LSTM cell which are called gates. The cell state transfers relative data throughout the processing of the sequence chain. The forgot gate determines whether the previous information needs to be remembered or can be forgotten. The input gate learns new information from the input. The output gate passes the updated information as output from the current cell state.

The gates use a sigmoid activation function. It is similar to the tanh activation but instead of squeezing the values between -1 and 1, it squeezes the values between 0 and 1. This allows to update or forget



data because the values which are multiplied by 0 are forgotten and the values multiplied by 1 are the same value so it is kept in memory.



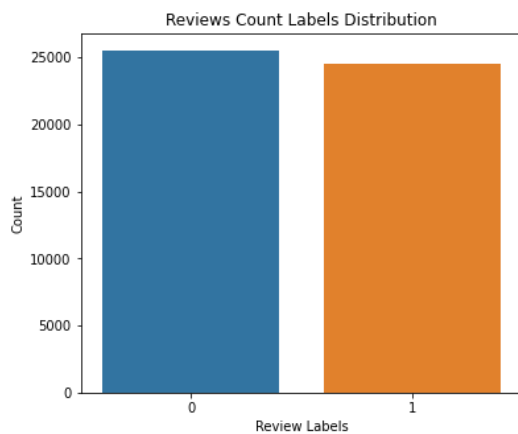
### **Bidirectional LSTM:**

In this project, the bidirectional is LSTM also experiments. In the bidirectional LSTM, the model training is performed on two LSTM models. The first model learns the sequence of the provided input, while the second model learns the reversed copy of the input sequence. This provides additional context to the network and results in better accuracy in learning the problem.

## Experiments

The raw customer review data from amazon required to be preprocessed first before applying the machine learning algorithms so, to normalize the data, several preprocessing steps were applied. The data was first converted to lowercase and then the URLs, HTML elements, numbers, line breaks, and punctuations were eliminated to reduce the noise from the data. Later, the stop words were eliminated and the lemmatization technique was applied to bring the word to its base form. The text paragraphs were converted into sequences and the padding was applied to extract the 4,500 features from the data.

To train the dataset, an almost equal number of positive reviews and negative reviews are utilized for the model to be trained properly.

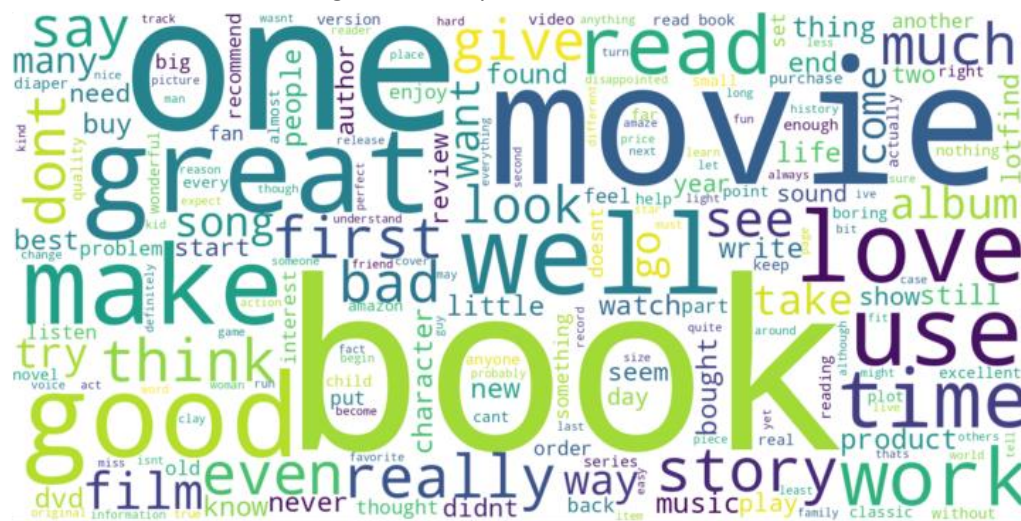


To gain a better understanding of the words which are frequently appearing in the review data, the word cloud model was developed to get gain some interesting insights.

Below is the word cloud diagram developed from the training dataset:



Below is the word cloud diagram developed from the test dataset:



From the above word cloud, it can be seen that there are several reviews regarding the books, music, and movies. People seem to have used well, good, love, great more often to describe a positive review. They have used doesn't, never, bad, nothing, more often to describe a negative review.

After preprocessing using various NLP methods and gathering a few insights on the review dataset, the classification models were developed to test the accuracy of the test data. For each model, several evaluation metrics are collected like accuracy, precision, recall, f1-measure, and AUC score.

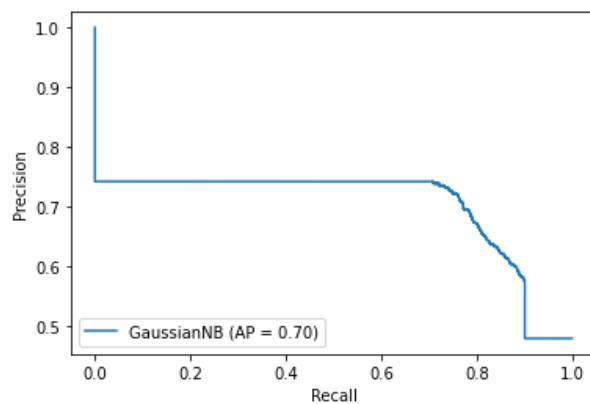
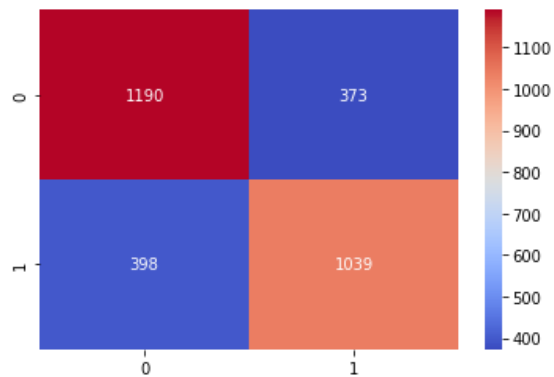
Below is the cleaned data example after applying the preprocessing step on training and test data:

	text	label	clean_text
0	stuning even for the non-gamer: this sound track was beautifull it paints the senery in your mind so well i would recomed it even to people who hate vid. game music! i have played the game chrono cross but out of all of the games i have ever played it has the best music! it backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras. it would impress anyone who cares to listen! ^_^	0	stun even nongamer sound track beautiful paint senery mind well would recomend even people hate vid game music played game chrono cross game ever played best music back away crude keyboarding take fresher step grate guitar soulful orchestra would impress anyone care listen
1	the best soundtrack ever to anything.: i'm reading a lot of reviews saying that this is the best 'game soundtrack' and i figured that i'd write a review to disagree a bit. this in my opinino is yasanori mitsuda's ultimate masterpiece. the music is timeless and i'm been listening to it for years now and its beauty simply refuses to fade. the price tag on this is pretty staggering i must say, but if you are going to buy any cd for this much money, this is the only one that i feel would be worth every penny.	0	best soundtrack ever anything reading lot review say best game soundtrack figure write review disagree bit opinino yasanori mitsuda's ultimate masterpiece music timeless listen year beauty simply refuse fade the price tag pretty stagger must say go buy much money one feel would worth every penny
	text	label	clean_text
0	great cd: my lovely pat has one of the great voices of her generation. i have listened to this cd for years and i still love it. when i'm in a good mood it makes me feel better. a bad mood just evaporates like sugar in the rain. this cd just oozes life. vocals are just stunning and lyrics just kill. one of life's hidden gems. this is a desert isle cd in my book. why she never made it big is just beyond me. everytime i play this, no matter black, white, young, old, male, female everybody says one thing "who was that singing ?"	0	great lovely pat one great voice generation listen year still love good mood make feel well bad mood evaporates like sugar rain ooze life vocal justat stunning lyric kill one life hidden gem desert isle book never make big beyond everytime play matter black white young old male female every/body say one thing sing
1	one of the best game music soundtracks - for a game i didn't really play: despite the fact that i have only played a small portion of the game, the music i heard (plus the connection to chrono trigger which was great as well) led me to purchase the soundtrack, and it remains one of my favorite albums. there is an incredible mix of fun, epic, and emotional songs. those sad and beautiful tracks i especially like, as there's not too many of those kinds of songs in my other video game soundtracks. i must admit that one of the songs (life-a distant promise) has brought tears to my eyes on many occasions. my one complaint about this soundtrack is that they use guitar fretting effects in many of the songs, which i find distracting. but even if those weren't included i would still consider the collection worth it.	0	one best game music soundtrack game didnt really play despite fact played small portion game music heard plus connection chrono trigger great well lead purchase soundtrack remains one favorite album incredible mix fun epic emotional song sad beautiful track especially like there many kind song video game soundtrack must admit one song life distant promise brought tret eye many occasionsmy one complaint soundtrack use guitar fret effect many song find distract even werent include would still consider collection worth it

### Gaussian Naïve Bayes:

In the first experiment, the Gaussian Naïve Bayes model was developed and experimented with. Below are the evaluation metrics which were generated from this model:

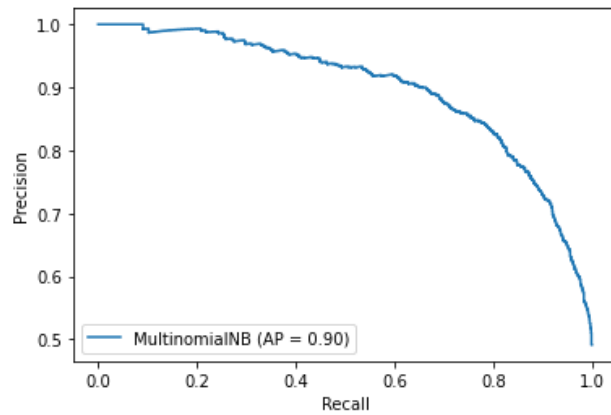
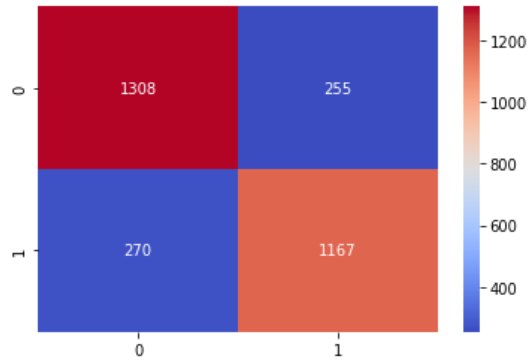
	Accuracy	Precision	Recall	F1-Measure	AUC
Gaussian Naïve Bayes	0.743	0.735836	0.723034	0.729379	0.742195



### Multinomial Naïve Bayes:

Below is the evaluation of the multinomial naïve Bayes model. From the results, it can be seen that the multinomial naïve Bayes model has performed relatively well when compared to the Gaussian naïve Bayes model.

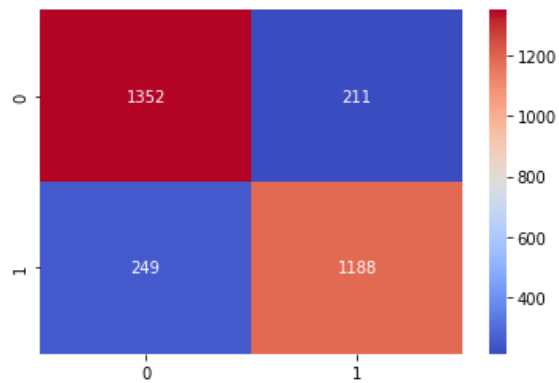
	Accuracy	Precision	Recall	F1-Measure	AUC
Multinomial Naïve Bayes	0.825	0.820675	0.812109	0.816369	0.82448

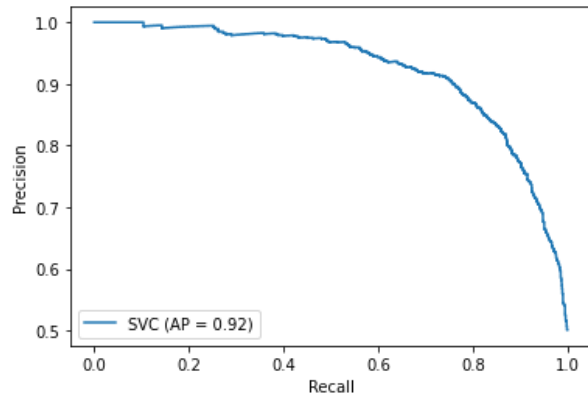


### Support Vector Machine (SVM):

The support vector machine model was developed with the 0.1 regularization parameter value and linear kernel. Below are the results from the support vector machine model.

	Accuracy	Precision	Recall	F1-Measure	AUC
<b>SVM</b>	0.846667	0.849178	0.826722	0.8378	0.845863

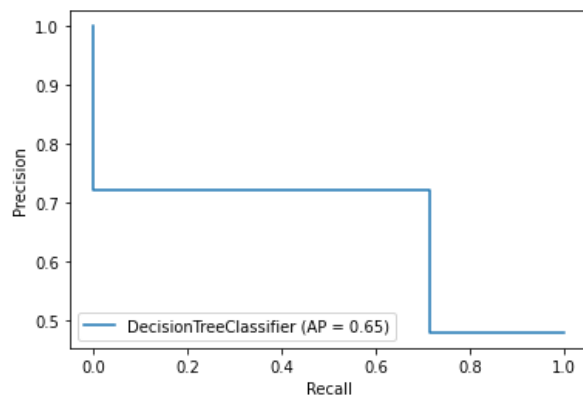
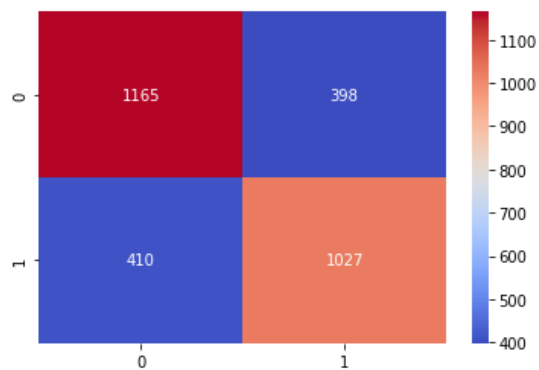




### Decision Tree Classifier

Since the decision tree classifier is a nonparametric approach that does not rely on probability distribution assumptions, a decision tree classifier was developed to understand the sequence of rules to classify the class of the reviews. Below are evaluation metrics from the model:

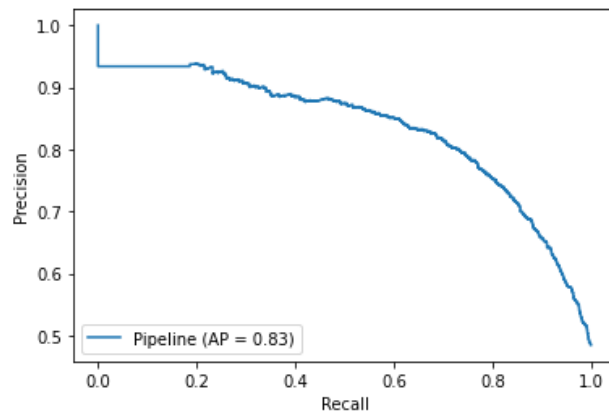
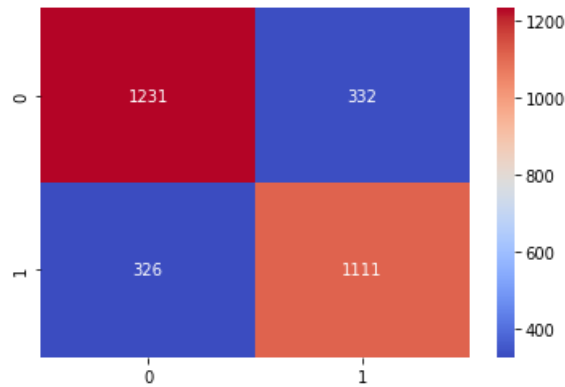
	Accuracy	Precision	Recall	F1-Measure	AUC
Decision Tree	0.730667	0.720702	0.714683	0.71768	0.730022



### Logistic Regression:

To test the logistic regression, the model pipeline is developed which contains the StandardScalar and Logistic regression model. Since Logistic regression requires the features to be standardized, the StandardScaler function is added to the pipeline to standardize a feature by subtracting the mean and then scaling to unit variance. After standardizing, the logistic regression model is applied.

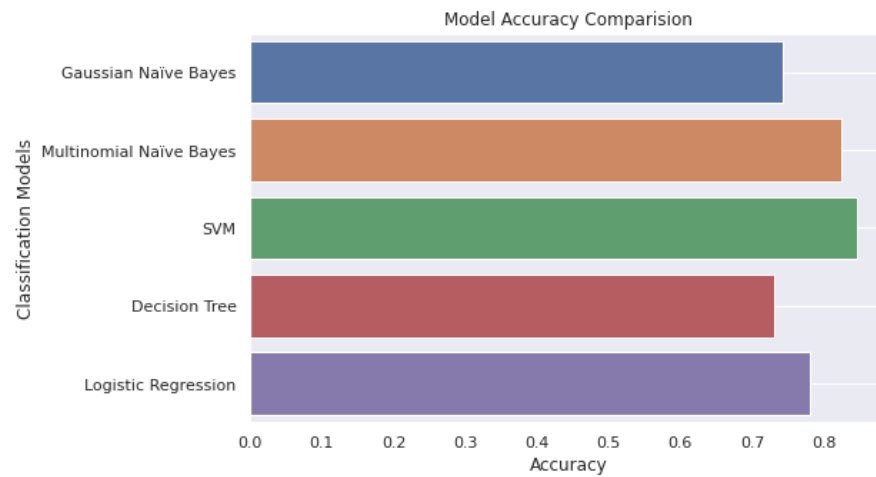
	Accuracy	Precision	Recall	F1-Measure	AUC
Logistic Regression	0.780667	0.769924	0.773138	0.771528	0.780363



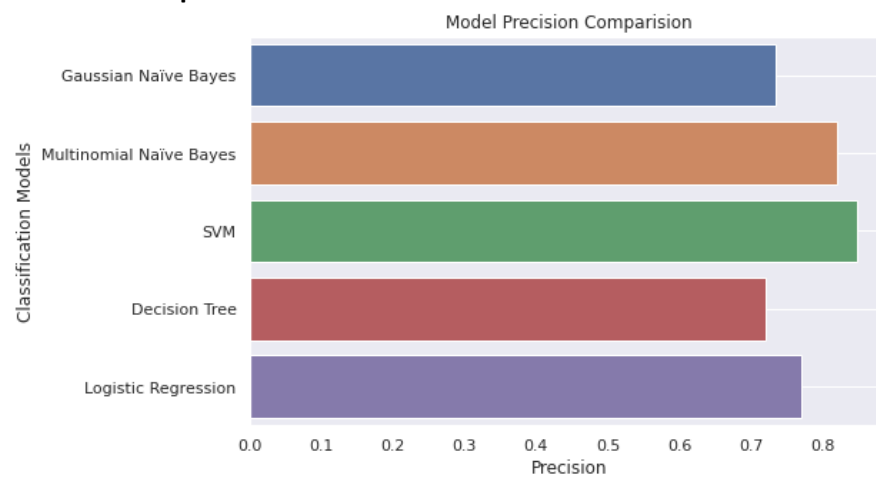
Summary of all the above classifier models:

	Accuracy	Precision	Recall	F1-Measure	AUC
Gaussian Naïve Bayes	0.743000	0.735836	0.723034	0.729379	0.742195
Multinomial Naïve Bayes	0.825000	0.820675	0.812109	0.816369	0.824480
SVM	0.846667	0.849178	0.826722	0.837800	0.845863
Decision Tree	0.730667	0.720702	0.714683	0.717680	0.730022
Logistic Regression	0.780667	0.769924	0.773138	0.771528	0.780363

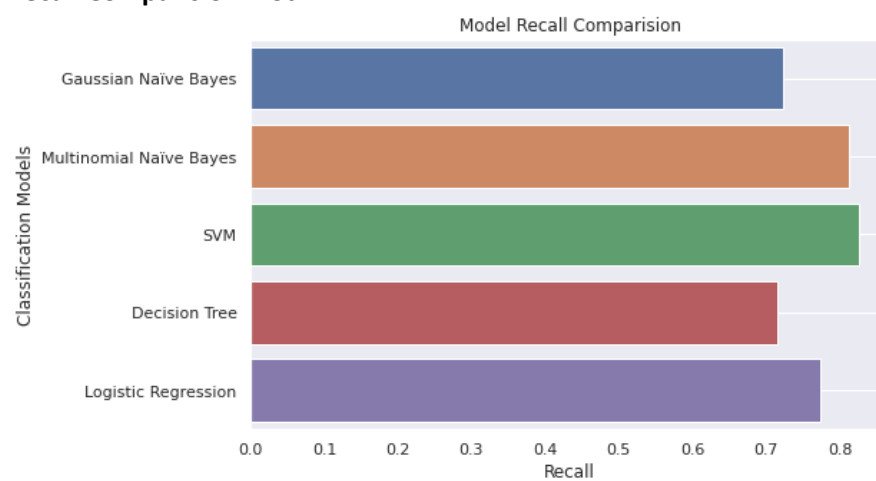
Accuracy Comparison Plot:



### Precision Comparison Plot:

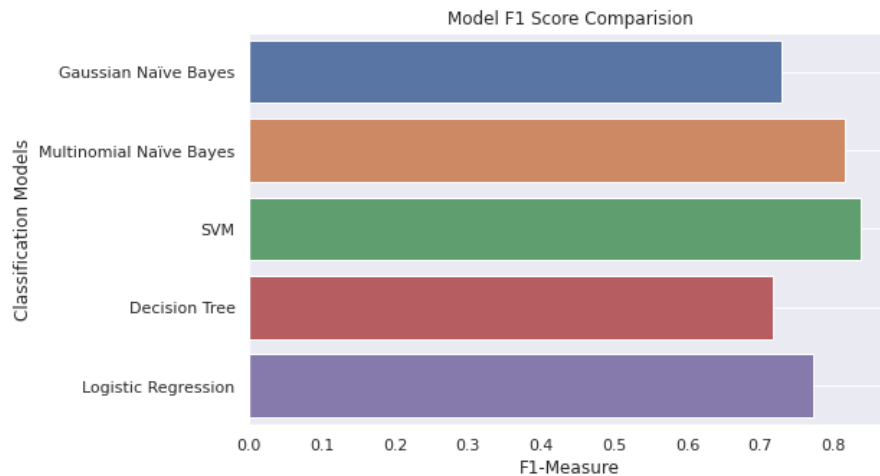


### Recall Comparison Plot:

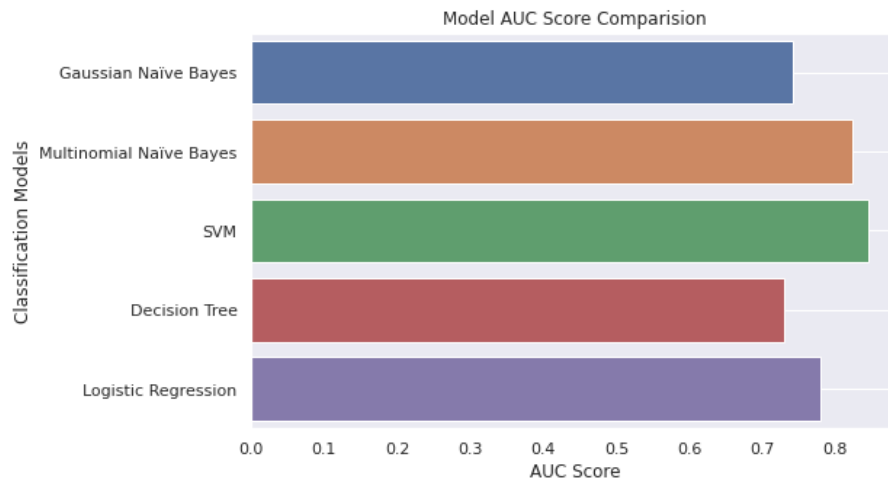




### F1 Score Comparison Plot:



### AUC Score Comparison Plot:



From the above plots, it can be seen that the SVM model has performed relatively well when compared to the other models. It has attained 84% accuracy while the Multinomial Naïve Bayes model is the second-best performing model after the SVM with an accuracy of 82%.

After performing the analysis on the above classification algorithms, I have developed an LSTM model to test the model performance.

### Long Short-Term Memory (LSTM):

There are several different variations of hyperparameters that I have used to test the LSTM models. While experimenting, I noted down hyperparameters that worked the best for the model and used it for the subsequent runs. So below are the results from the experiments which I have performed.

### LSTM Model 1:

Below are the hyper-parameters used for the first LSTM model:

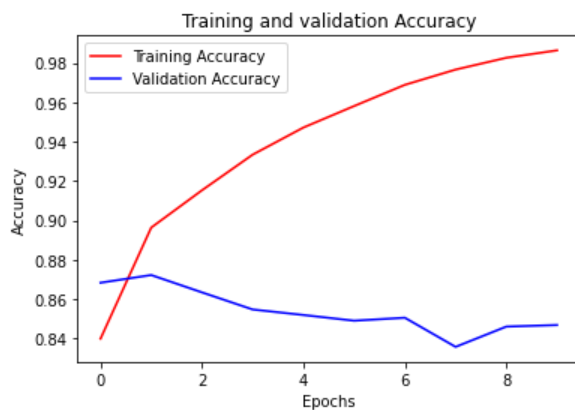
Input Dimension	4500
Output Dimension	300
Input length	256
LSTM layer	256
Activation Function	relu
Learning Rate	0.001
Dropout	.20
Optimizer	adam
Batch Size	64
Epochs	10

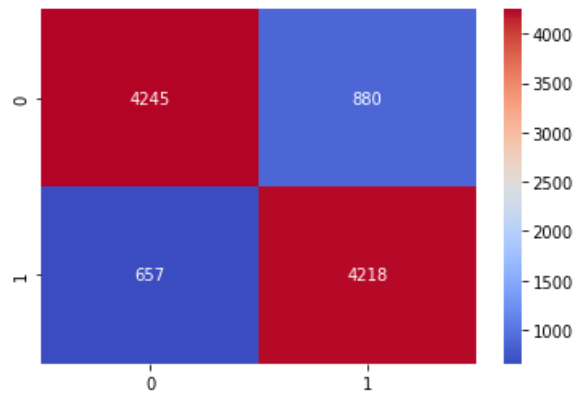
```
===== LSTM Model =====  
Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 256, 300)	1350000
lstm (LSTM)	(None, 256, 256)	570368
dropout (Dropout)	(None, 256, 256)	0
lstm_1 (LSTM)	(None, 256)	525312
dropout_1 (Dropout)	(None, 256)	0
dense (Dense)	(None, 1)	257

```
=====
```

Total params: 2,445,937  
Trainable params: 2,445,937  
Non-trainable params: 0





	Accuracy	Precision	Recall	F1-Measure	AUC
<b>LSTM</b>	0.8463	0.827383	0.865231	0.845884	0.846762

Looking at the performance metrics, it can be seen that the model did a good job in classifying the customer review samples but the epoch plots for accuracy and loss shows that the model is significantly underperforming.

### LSTM Model 2:

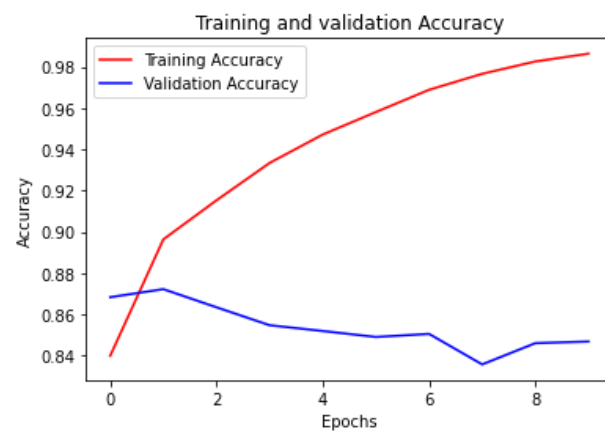
Below are the hyper-parameters used for the second LSTM model:

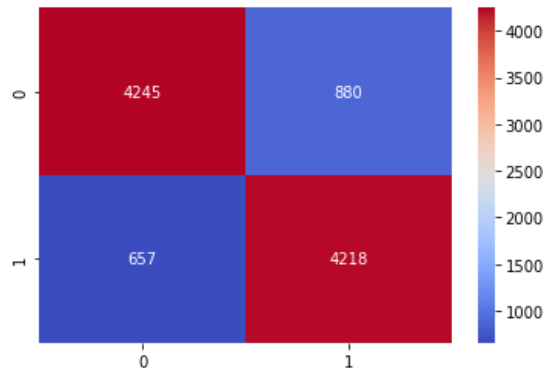
Input Dimension	4500
Output Dimension	136
Input length	256
LSTM layer	256
Activation Function	relu
Learning Rate	0.001
Dropout	.20
Optimizer	adam
Batch Size	128
Epochs	10

===== LSTM Model =====  
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 256, 300)	1350000
lstm (LSTM)	(None, 256, 256)	570368
dropout (Dropout)	(None, 256, 256)	0
lstm_1 (LSTM)	(None, 256)	525312
dropout_1 (Dropout)	(None, 256)	0
dense (Dense)	(None, 1)	257

Total params: 2,445,937  
Trainable params: 2,445,937  
Non-trainable params: 0





	Accuracy	Precision	Recall	F1-Measure	AUC
LSTM	0.8463	0.827383	0.865231	0.845884	0.846762

Again, in this model, it seems like the model is significantly underperforming as the training set is gaining higher accuracy but the validation set is not able to learn and perform as expected.

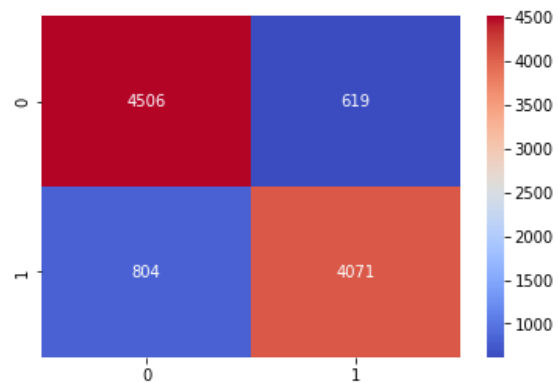
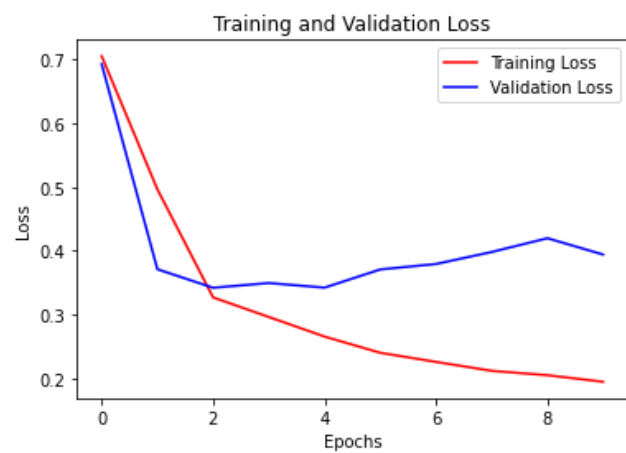
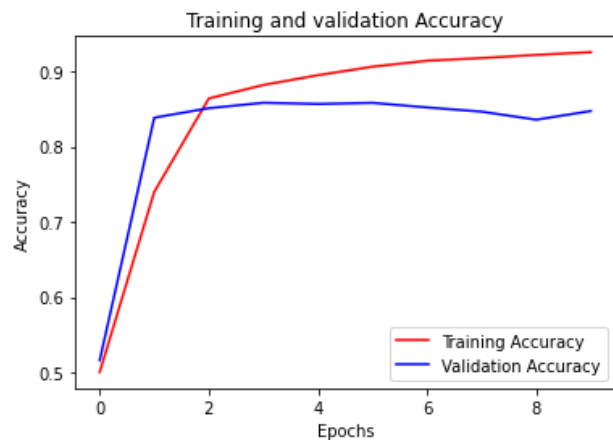
### LSTM Model 3:

Below are the hyper-parameters used for the third LSTM model:

Input Dimension	4500
Output Dimension	128
Input length	256
LSTM layer	300
Activation Function	relu
Learning Rate	0.01
Dropout	.20
Optimizer	adam
Batch Size	128
Epochs	10

===== LSTM Model =====  
Model: "sequential\_2"

Layer (type)	Output Shape	Param #
=====		
embedding_2 (Embedding)	(None, 256, 128)	576000
lstm_4 (LSTM)	(None, 256, 300)	514800
dropout_4 (Dropout)	(None, 256, 300)	0
lstm_5 (LSTM)	(None, 300)	721200
dropout_5 (Dropout)	(None, 300)	0
dense_2 (Dense)	(None, 1)	301
=====		
Total params: 1,812,301		
Trainable params: 1,812,301		
Non-trainable params: 0		



	Accuracy	Precision	Recall	F1-Measure	AUC
<b>LSTM</b>	0.8577	0.868017	0.835077	0.851228	0.857148

With the above hyperparameters, it can be seen that the model has performed significantly well with 85% accuracy. Even the loss and accuracy plots for the epoch show that the model is not underperforming like it did in the other two models.

## Bidirectional Long Short-Term Memory (LSTM):

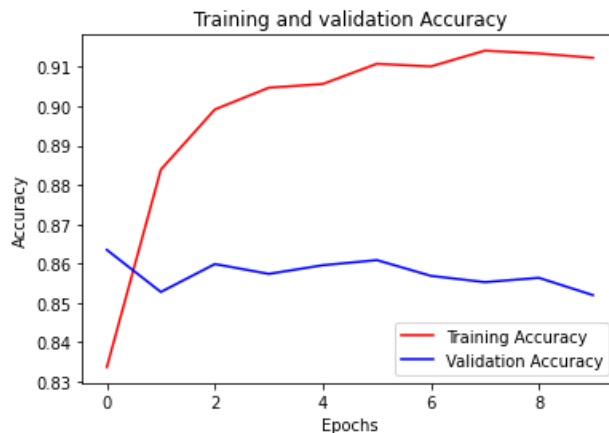
For the bidirectional LSTM model, I have also tested several different hyperparameters. Below are a few of the experiments which I have performed:

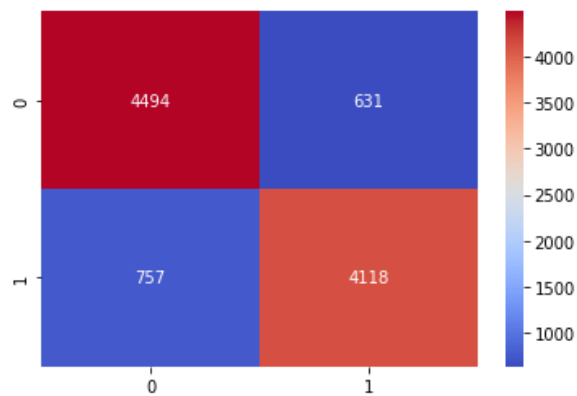
### Bidirectional-LSTM Model 1:

Input Dimension	4500
Output Dimension	64
Input length	256
LSTM layer	25
Dense Layer	34
Activation Layer	relu
Learning rate	.01
Optimizer	adam
Batch Size	32
Epochs	10

```
===== Bidirectional LSTM Model =====  
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, 256, 64)	288000
spatial_dropout1d (SpatialDropout1D)	(None, 256, 64)	0
bidirectional (Bidirectional LSTM)	(None, 50)	18000
dropout_6 (Dropout)	(None, 50)	0
dense_3 (Dense)	(None, 34)	1734
dense_4 (Dense)	(None, 1)	35
Total params: 307,769		
Trainable params: 307,769		
Non-trainable params: 0		





	Accuracy	Precision	Recall	F1-Measure	AUC
LSTM	0.8612	0.86713	0.844718	0.855777	0.860798

Looking at the performance metrics, it seems that the model has performed relatively well. However, the model is underfitting.

### Bidirectional-LSTM Model 2:

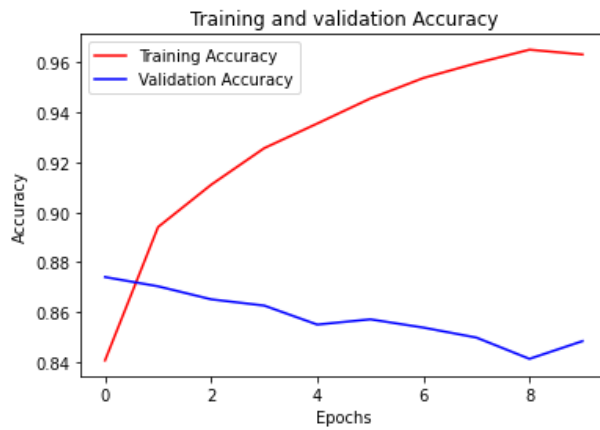
Input Dimension	4500
Output Dimension	64
Input length	256
LSTM layer	25
Dense Layer	32
Activation Layer	relu
Learning rate	.001
Optimizer	adam
Batch Size	32
Epochs	10

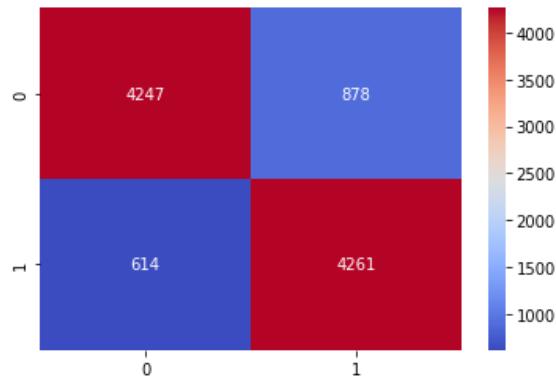


===== Bidirectional LSTM Model =====  
Model: "sequential\_4"

Layer (type)	Output Shape	Param #
embedding_4 (Embedding)	(None, 256, 64)	288000
spatial_dropout1d_1 (Spatial Dropout)	(None, 256, 64)	0
bidirectional_1 (Bidirectional LSTM)	(None, 50)	18000
dropout_7 (Dropout)	(None, 50)	0
dense_5 (Dense)	(None, 32)	1632
dense_6 (Dense)	(None, 1)	33

Total params: 307,665  
Trainable params: 307,665  
Non-trainable params: 0





	Accuracy	Precision	Recall	F1-Measure	AUC
LSTM	0.8508	0.82915	0.874051	0.851009	0.851367

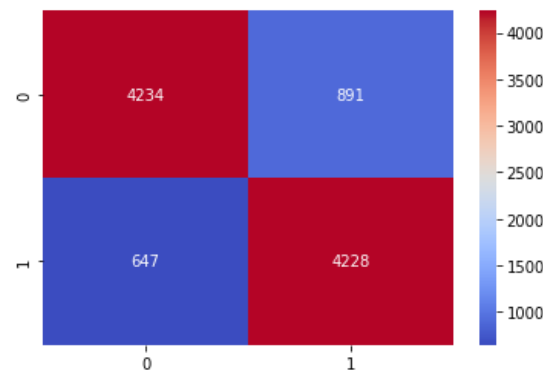
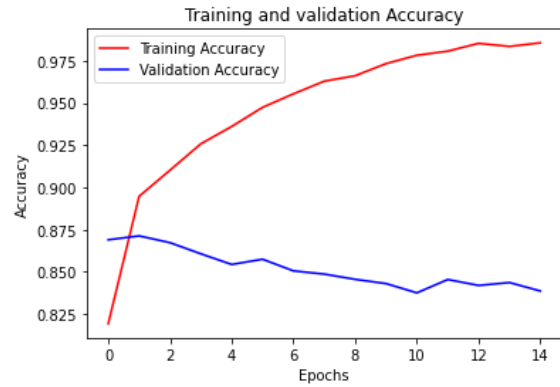
Looking at the performance metrics, the accuracy of the model has reduced from the previous model and it is still underfitting even after tweaking the hyperparameters.

### Bidirectional-LSTM Model 3:

Input Dimension	4500
Output Dimension	200
Input length	256
LSTM layer	256
Dense Layer	80
Activation Layer	relu
Learning rate	.001
Optimizer	adam
Batch Size	128
Epochs	15

```
===== Bidirectional LSTM Model =====
Model: "sequential_12"
```

Layer (type)	Output Shape	Param #
embedding_12 (Embedding)	(None, 256, 200)	900000
spatial_dropout1d_9 (Spatial Dropout)	(None, 256, 200)	0
bidirectional_9 (Bidirectional LSTM)	(None, 512)	935936
dropout_15 (Dropout)	(None, 512)	0
dense_21 (Dense)	(None, 80)	41040
dense_22 (Dense)	(None, 1)	81
Total params: 1,877,057		
Trainable params: 1,877,057		
Non-trainable params: 0		



	Accuracy	Precision	Recall	F1-Measure	AUC
LSTM	0.8462	0.825943	0.867282	0.846108	0.846714

From the above results, it seems that the model has gained good accuracy but the model is not able to learn from the tests data for applying the validation data.

### Final Results:

After experimentation, it can be seen that the bidirectional-LSTM model attained the maximum accuracy of 86% followed by the LSTM model with 85%, and the SVM model the third place with an accuracy of 84%.

## Conclusion

In this project, I have used the proposed deep learning approach along with the other classical machine learning methods to solve the problem which is often faced by e-commerce websites to determine the polarity of the customer reviews on their products and services. The experiments from the different models showed that the bidirectional LSTM model attained the highest accuracy of 86%, the precision of 86%, recall of 84%, F1-score of 85%, and AUC score of 86% when compared to the other models.

The overall experience of this project was great. I got to learn different NLP techniques and implemented them in this project. I have not worked on the application of the Deep Learning models like LSTM, and bi-directional models before so this project allowed me to learn the application of these models on the real data.

There are several challenges that I faced during the development and experimentation of the classification models. Since the deep learning models require lots of training data to be trained on, my local environment was not sufficient to handle the computation load so I had to test various environments like Kaggle, google Collab, etc. to train the models. Another major problem that I have faced while learning LSTM models is underfitting. I tried to tweak hyperparameters, increased the size of training and testing data, and ran several different experiments with different epochs but it did not provide me the expected results as the research paper.

For future work, I would like to test a few more NLP approaches to preprocess the dataset and would like to test the hybrid approaches with the combination of CNN and LSTM models. Also due to the time crunch, I was not able to utilize the entire dataset to train the model so, in the future, I would try to use the entire dataset for training and testing the data to attain maximum accuracy.

## References

- Kamal Hassan, Mohan, et al. "Sentimental Analysis of Amazon Reviews Using Naïve Bayes on Laptop Products with MongoDB and R." IOP Conference Series: Materials Science and Engineering, vol. 263, 2017, p. 042090., <https://doi.org/10.1088/1757-899x/263/4/042090>.
- K. K. Mohbey, "Sentiment analysis for product rating using a deep learning approach," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 121-126, DOI: 10.1109/ICAIS50930.2021.9395802.
- Rather, Abhilasha Singh, et al. "Comparative Study of Machine Learning Approaches for Amazon Reviews." Procedia Computer Science, vol. 132, 2018, pp. 1552–1561., <https://doi.org/10.1016/j.procs.2018.05.119>.
- Shrestha, Nishit, and Fatma Nasoz. "Deep Learning Sentiment Analysis of Amazon.com Reviews and Ratings." International Journal on Soft Computing, Artificial Intelligence, and Applications, vol. 8, no. 1, 2019, pp. 01–15., <https://doi.org/10.5121/ijscai.2019.8101>.