BOOK GENRE PREDICTION USING LSTM

A PROJECT REPORT
SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By

NISHTHA VERMA

(2K20/ISY/13)

Under the guidance of

Dr Ritu Agarwal

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2022

CERTIFICATE

This is to certify that Ms. Nishtha Verma (2K20/ISY/13) has finished the major project titled "Book Genre Prediction Using Lstm Model" as a partial fulfillment for the award of Master of Technology degree in Information Systems by Delhi Technological University. Under my supervision and guidance during the academic session 2020-2022.

(Project Guide)
Dr. Ritu Agarwal
Assistant Professor
Department of Information Technology
Delhi Technological University
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I want to thank my major project guide Dr. Ritu Agarwal , Assistant Professor, lT Dept., Delhi Technological University, for her kind support and guidance that she has provided me in order to finish this major project. I would also like the to take this opportunity to thank the institute for providing us the infrastructure .At last I want to thanks my family and friends for the throughout support.

Nishtha

Nishtha Verma
Roll No. 2K20/ISY/13
MTech (Information System)
E-mail: nishthav67@gmail.com

ABSTRACT

Book genre prediction is a classification-based project where by applying various models of ML I have tried to predict the genre of book by using summary of book by using CMU book dataset that I have taken from Kaggle. I have performed text preprocessing and cleaning after that remove stop words and final on training model on dataset by taking different ratio and compare performance by F1-score and accuracy score. First part of project for machine training I have used supervised machine learning model like KNN, logistic regression and later I have moved to deep learning approach for that I have used lstm model. At last, I have made comparison based on performance matrix to judge the models and its accuracy.

Table of Contents

List of Figures

List of Tables

1 Chapter 1- INTRODUCTION

1.1 INTRODUCTION

Digitalization has become a major part of our life and also it has changed our ways to read and study about anything for example now for almost everything we want to learn we have videos on YouTube, still books have their different kind of importance there are people who prefer books over online platforms. Talking about online platforms, Books are available there also like amazon's kindle, for those who enjoy reading it's important to classify genres based on their taste. Text classification is not new in the machine learning field and is playing a vital role in our life as digital text and the number of online users are growing dramatically . Document classification by taking the text ,analyzing the Sentiment based on reviews on books and movies has been done by the researchers, various ways of vectorization of text, cleaning of data is there. Oftentimes people get misled by cover pages and advertisements about the book and as a result they buy the wrong books. This problem can be handled if we have a summary for books and we can predict genre from that. Also suppose we have huge numbers of books and we want to classify them on the basics of the genre in a particular order. So, reading a summary of each book and deciding which book belongs to which genre is a time-consuming task. What if a machine can do that for us?

Summary of books is a crucial part to detect the genre of books. But sometimes the summary doesn't help determine the correct genre, we can get hints from the summary of books. The project's main focus is to develop a model that can detect models and compare the model's performance the correct genre of a book by analyzing the summary of the books. Through my project we have tried to predict the genre of book from the dataset. I have used CMU datasets to check the accuracy of my models. In the minor project I have implemented it using supervised machine learning models and compare the performance of models by using performance metrics and in the second part of the project I have tried to extend it and apply the deep learning model in it.

1.2 Text Classification

Text in any form is crucial source of data, extracting valuable information from the given text has gained importance in the digital world with the increasing online data size by various platforms. Text classification is basically a means to classify our given set of data into the class where it belongs. Although we have a rich source of data from various sources most of the data, we get is in an unstructured form which makes it difficult to predict the class of the text. Text classification has three main types are as follow -

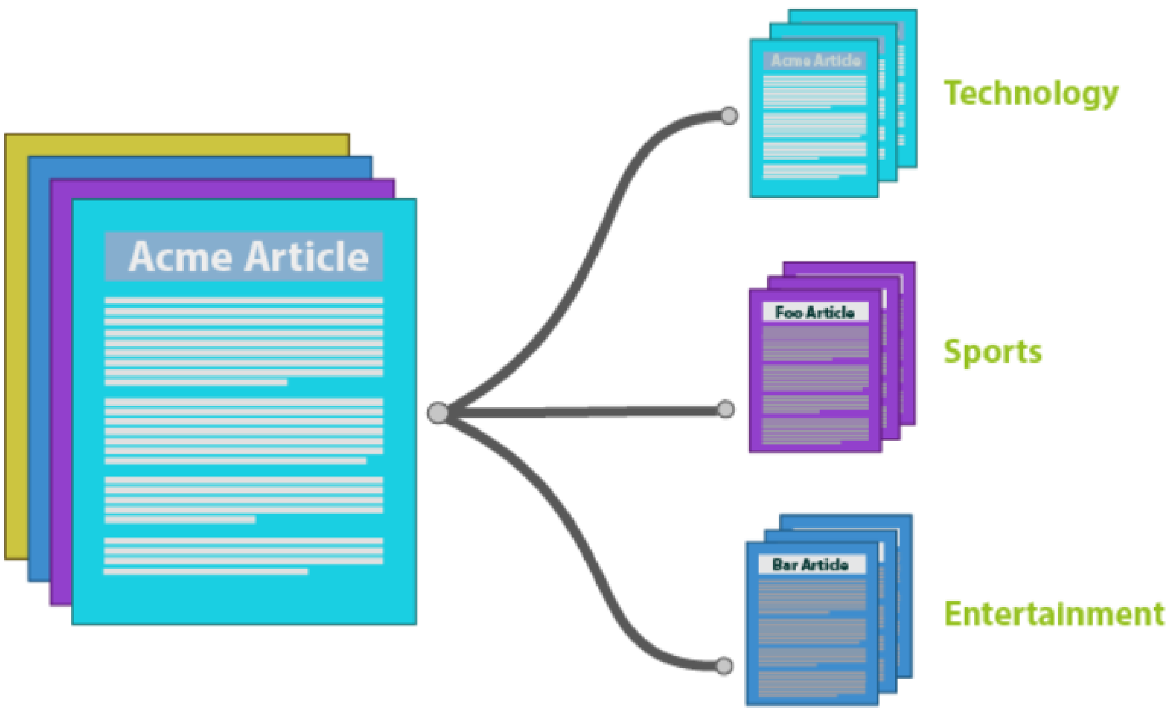

Figure 1.1 Text Classification[30]

1.2.1 Rule-based systems

This system of Rule-based categorized the text into certain groups by using those sets of predefined rules.. Let's take an example of news articles that has two groups of advertisements and covid related blogs, for this first we need to have a list of words that help us to classify to each group (examples words that are related to advertisements are offers, free, price, sale, range, etc., and words related to covid related blogs are (cases, vaccine, death, recover, state, etc.).Further, we want to organize a new range of text coming, we have to check the words count that are related to advertisement and covid blogs related and that’s how it is going to work.

1.2.2 Machine learning based systems

Classifying the text by taking the advantage of machine , machine predict the new text based on old observations. AS machine understands numerical language so first we need to change the word or text into vector form then on the bases of set of vectors machine can understand the word and can compare and classify the two words whether they are related or close enough.

.

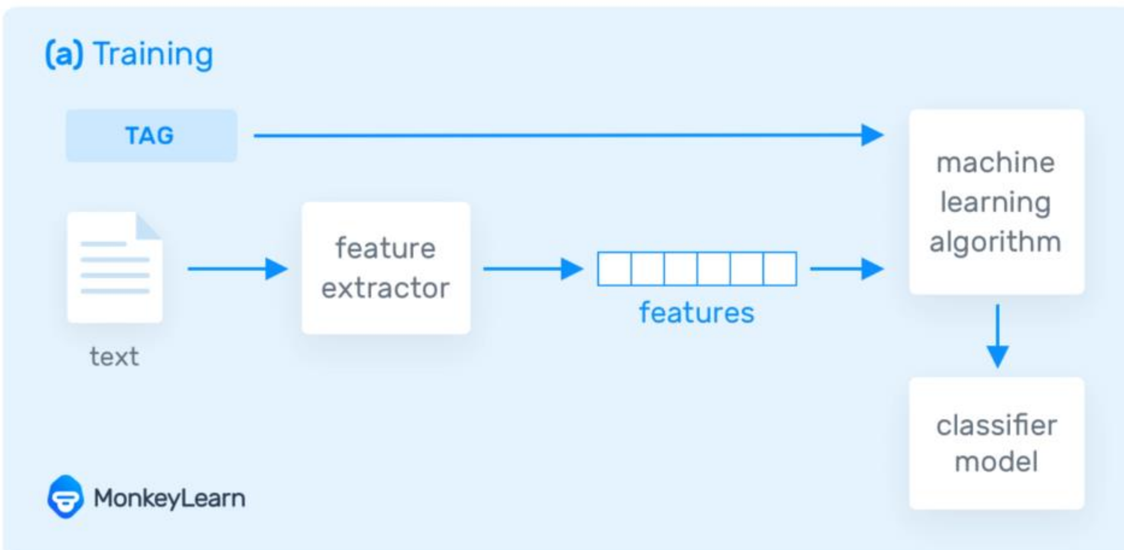

Figure1.2 Data Training[30]

.

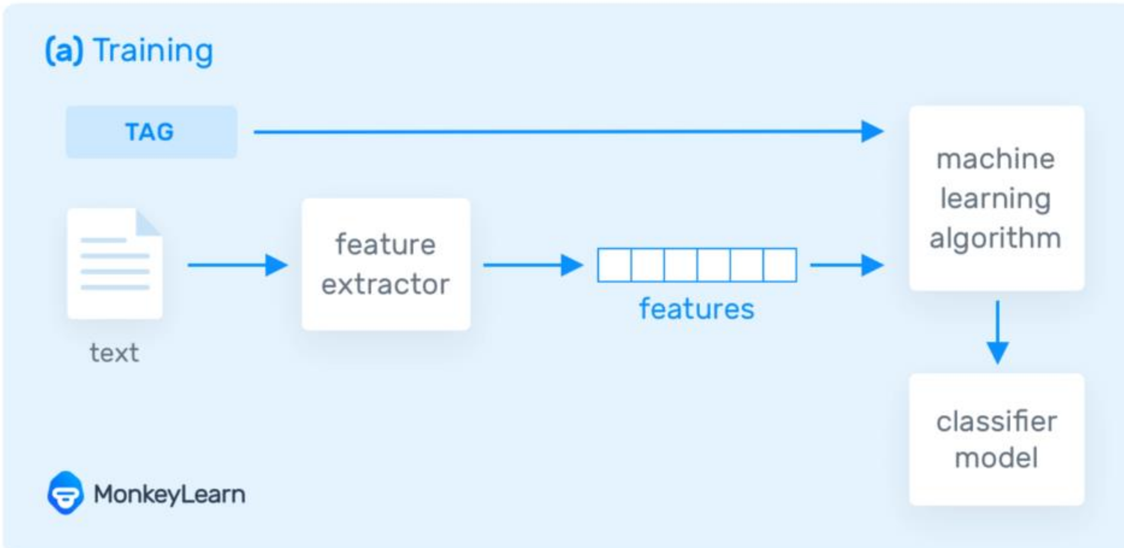

Figure1.3 Data Prediction[30]

1.2.3 HYBRID SYSTEMS

Hybrid systems involve both the handcrafted rules as well as machine learning models. It can be used by adding rules to determine accurately the text class.

.

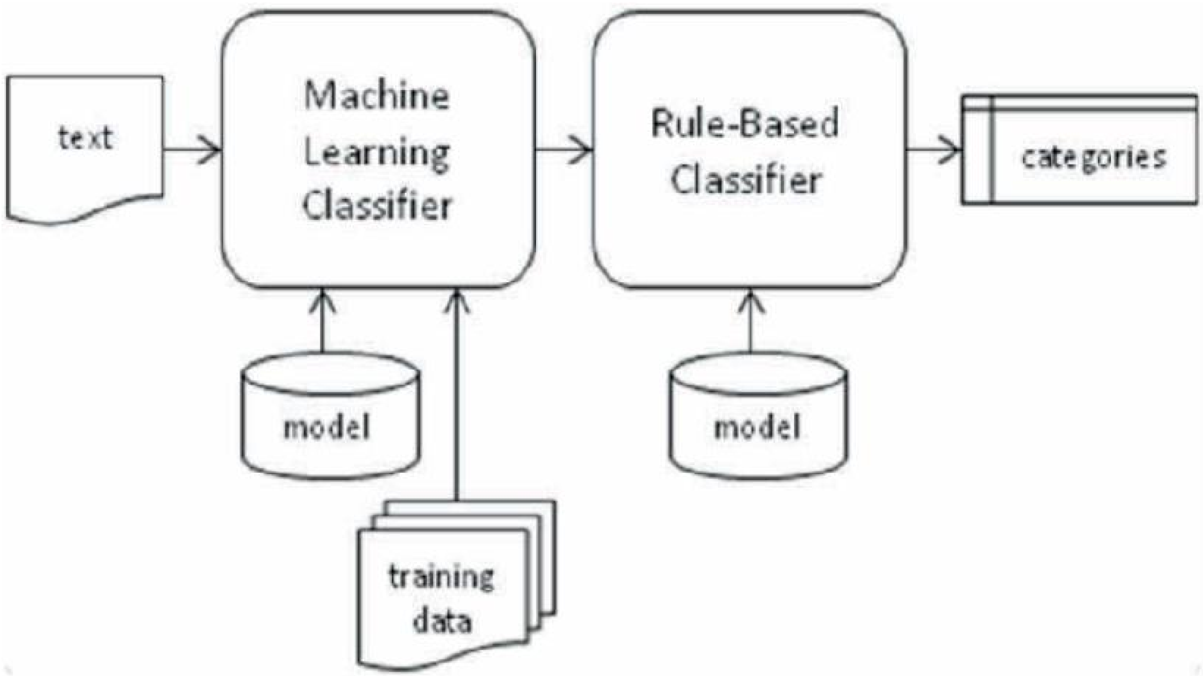

Figure 1.4 Hybrid Systems

1.3 Text Classification Techniques

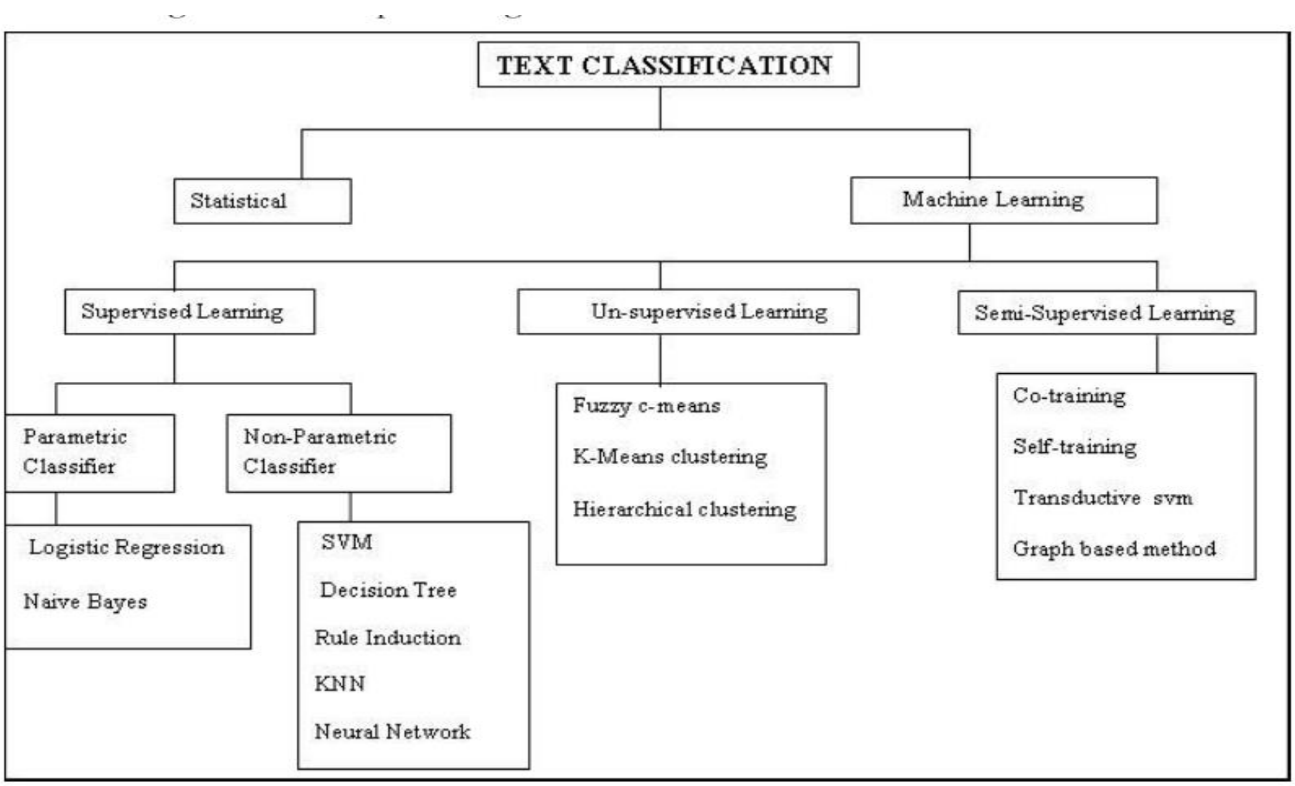

Figure 1.5 Text Classification Techniques

Text classification falls under two categories.

Statistical - The statistical approach involves a mathematics hypothesis, manually satisfies all the rules define in the hypothesis, and doesn't give good results for a large dataset. This is manual approach can be used for binary classification for example email belongs as spam or not.

Machine Learning approach - The machine learning approach is to automate the computation, first, we train the machine then all work is done by the machine when it gets the new dataset. machine learning is a fast-growing industry now as it saves manpower, makes computation faster, and gives accurate results even with a huge amount of data.

Machine Learning Text Classification Algorithms ::

1.3.1 Logistic Regression

In this technique of classification, prediction is on done on the bases of probability whether it is true or false, probability of the independent variable is calculated based on the dependent variable, for example in an image particular object is present or not.[The dependent variable is categorical, while the independent variable belongs to numerical or categorial data.[14]

$$P(Y=1|X) \text{ or } P(Y=0|X)$$

1.3.2 Naive Bayes

Naive Bayes check that what is the possibility that data belong to this particular category, which can be used in text classification ,check for the text matches the tag or phrases. Probability of one variable is calculated while assuming the other is trye, here probability of A, while B i.e., true and likewise.[6]

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

1.3.3 K-nearest Neighbors

KNN is used to detect pattern in the given dataset, for a particular point belong to this pattern or category. The algorithm presumes the similarity between the new point and old cases on which machine is already trained and accordingly keep the new point to the nearest most similar category can be used for regression as well. Depending the K value, distance will be calculated for new datapoint, for example if k=2 , then distance will be calculated from nearest two point and new point and then it will be classified into category.

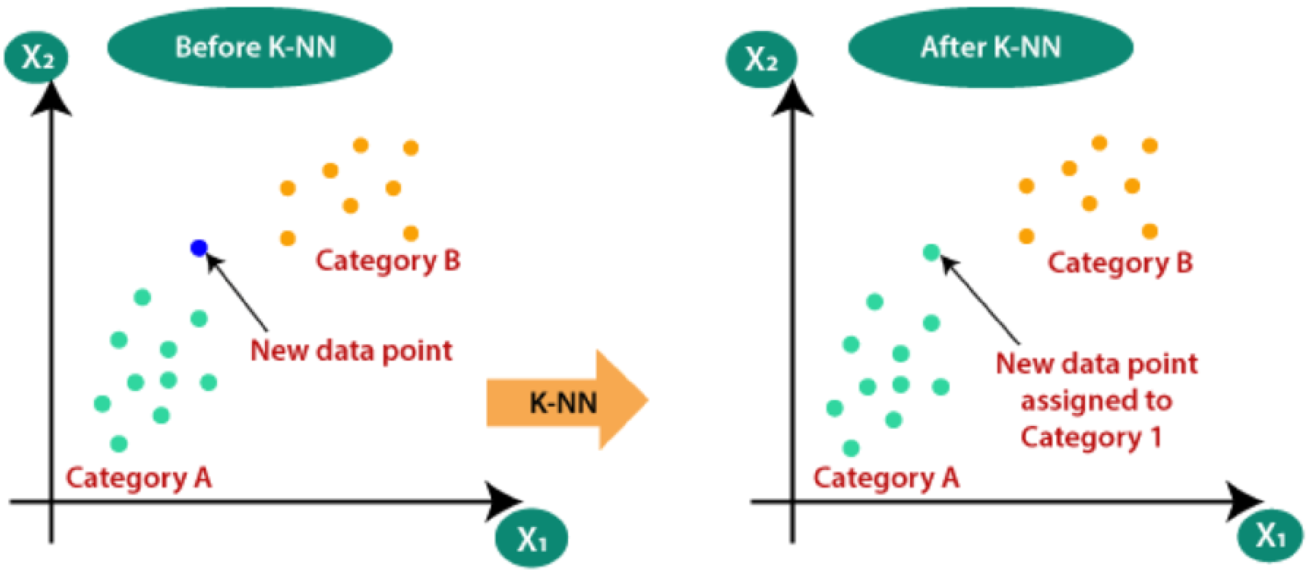

Figure 1.6 KNN Techniques

1.3.4 Decision Tree

Decision tree works like flowchart and is ideal algorithm for classification. The algorithm separates similar datapoints in same category then if sub classification is required then it goes to level by level, fine granularity.

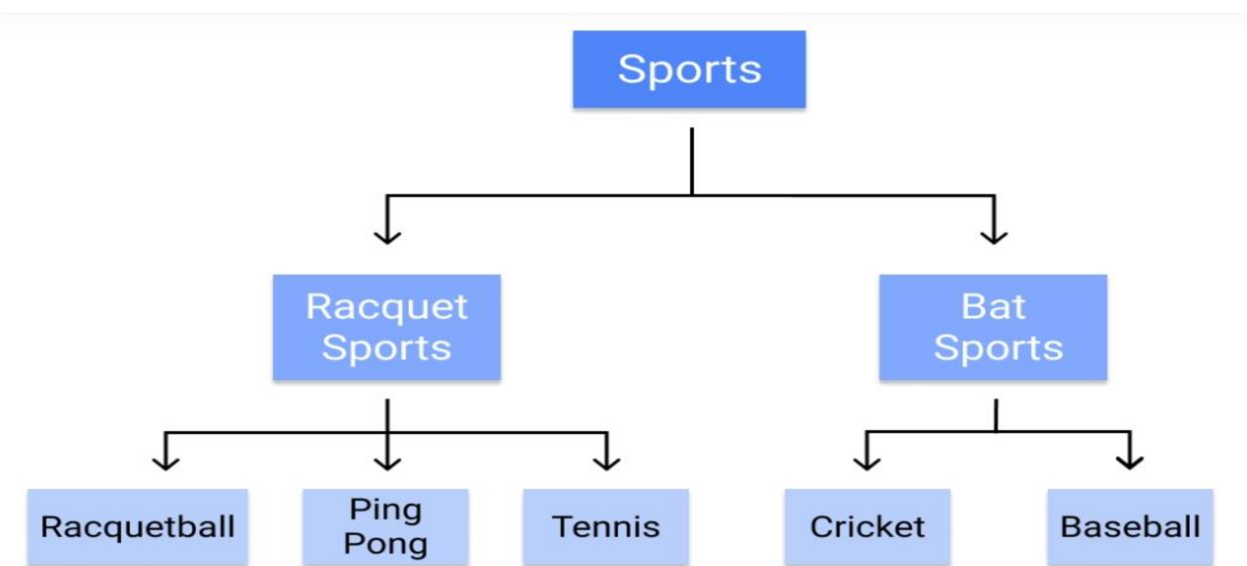

Figure 1.7 Decision Tree Techniques

1.3.5 Random Forest Techniques

Random forest is extension of decision tree, here first we select tree and then in that tree we classify our new point. All the tree here is nondependent of other which decreased the chance of error of the model.

1.3.6 Support Vector Machines Techniques

The model train and classify the data by using hyperplane, points which lies on either side of plane belongs to same class.

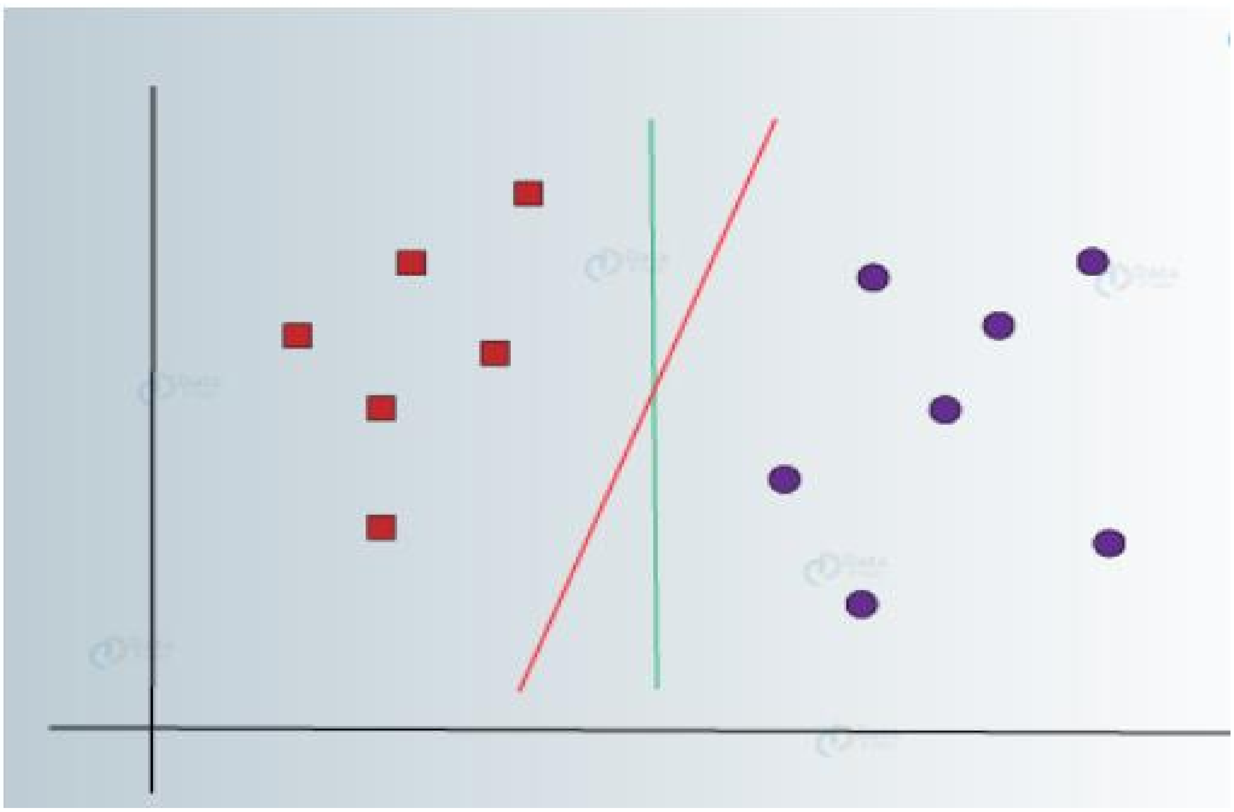

Figure 1.8 Support Vector Machine Techniques

1.4 Text Classification Application

- Sentiment Analysis

  One of the initial applications of text classification of text is identify the sentiment from text by analysis the text.Largerly used by social media platform mainly by twitter to extract the hate speech or anti content to track such people who are spreading hatred by using twitter as tool.

- Email Spam Classification

  Binary classification used for email spam classification, machine will read the text and will try to detect whether the mail is spam or not by identifying the keywords and all.

- Document Classification

  Now to classify the document based on text which category it belongs for example content is sports related or entertainment for something else, it helped us to organize the document is proper alignment and made it easy to detect= document and working on it.

- Image Classification

2 Chapter 2: Multiclass Text Classification

2.1 Data Preprocessing

Preprocessing of data is an essential step in machine learning, the clearer data we provide to machine the better it will get train and provide the output by analyzing the data. Before using machine learning or data mining methods, make sure the data is of good quality.

Figure 2.1 Data Preprocessing

2.1.1 Data Cleaning

This stage starts with the convolution process of image with the Gaussian filters at multiple scales. Interest points are the maxima or minima of the difference of Gaussian that occur at different scales. Data cleaning involving removing unwanted tags, comma, syntax, dropping duplicates values because if we give useless data to machine then it is going to return waste. Data cleaning also involves handling the outlier also.

2.1.2 Data Transformation

We can't deal with raw and unstructured data so we need to transform it as per the requirement form, so that data ingestion and management will get easy. So, transformation of data involves removing null values, make the data enriched as per the model requirement.

2.1.3 Data Reduction

It is an optimization technique to reduce the data to its simplest form possible. It means shrink the data into physical place to decrease size and increase the capacity.

2.2 Vectorization

Data vectorization means representing words as set of vectors as machine understands numeric values. In order to ingest text into machine we need to data to be in numeric form. There is various method for vectorization as follow -

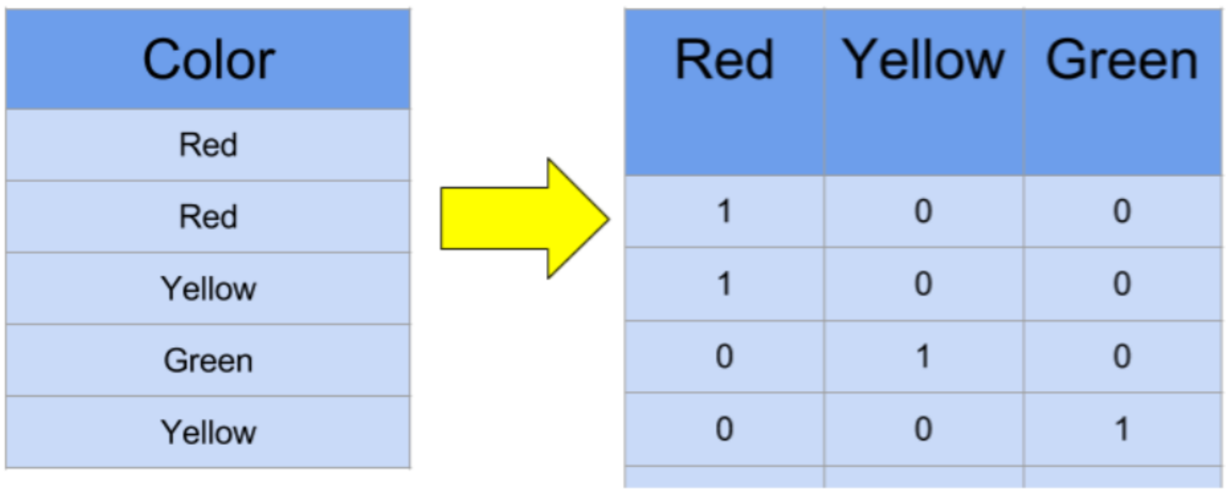

Figure 2.2 Data Vectorization

2.2.1 Glove

It is vectorization method created by Stanford Glove is an algorithm for embedding of word by gathering words across the global to generate word to word co-occurrence matrix.

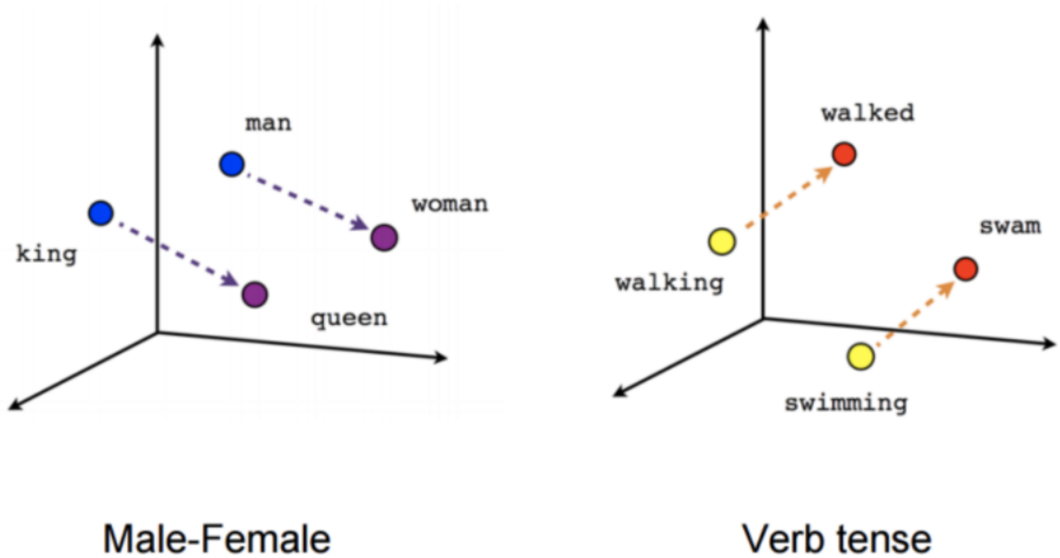

Figure 2.3 Glove

2.2.2 Tf-Idf

Term frequency-inverse document frequency(tf-idf) offers the weightage to that phrase which takes place very less time. i.e., it captures the uniqueness.

That means it will increase proportionally to the wide variety of instances withinside the textual content a phrase seems however is compensated through the phrase frequency withinside the corpus (data-set).Document Frequency: This checks the that means of the text, which may be very just like TF, withinside the complete corpus collection. The simplest distinction is that during record d, TF is the frequency counter for a time period t, at the same time as df is the quantity of occurrences withinside the record set N of the time period t. In different words, the quantity of papers wherein the phrase is gift is DF.

Inverse Document Frequency: Mainly, it checks how applicable the phrase is. The key intention of the hunt is to discover the right facts that healthy the demand. Since tf-idf considers all phrases similarly significant, it's far consequently now no longer simplest viable to apply the time period frequencies to degree the burden of the time period withinside the paper. First, locate the record frequency of a time period t through counting the quantity of files containing the time period:

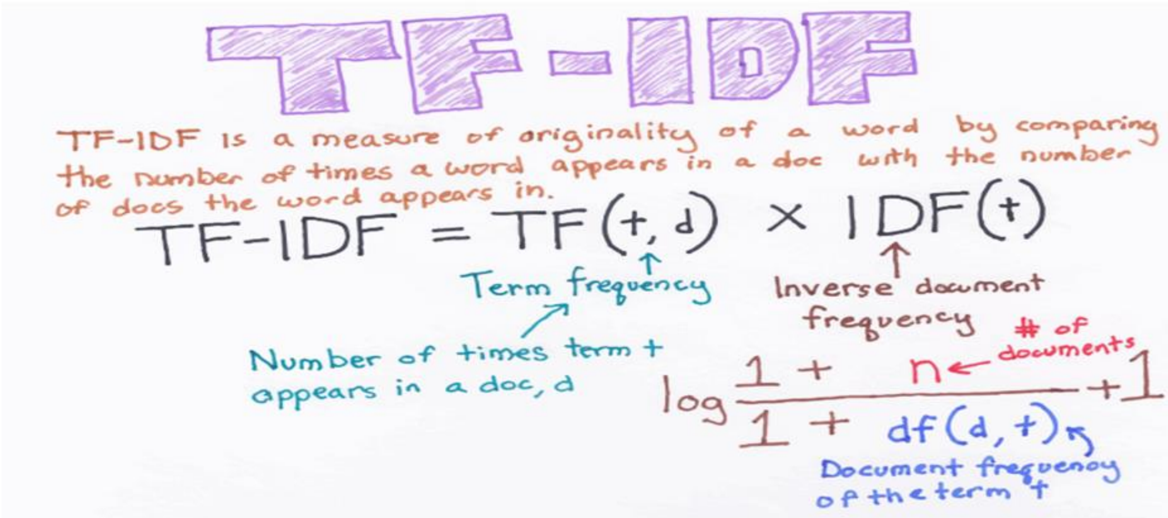

Figure 2.4 Tf-idf

2.2.3 Embedding Matrix

Word embedding is an important part as it reduces the dimension drastically, in order to understand the text by machine words needs to be converted into vector or set of arrays.

2.2.4 One hot encoding

Each letter is assigned a number, after that a matrix is developed to keep a one-dimensional array for a letter. Problem with one hot encoding sparse matrix because most of the block is zeroes only the diagonal matches each value of 1 other than that all are zeroes.

I	ate	an	apple	and	played	the	piano
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Figure 2.5 One-Hot Encoding

	1	2	3	4	5	6	7	8
I	1	0	0	0	0	0	0	0
ate	0	1	0	0	0	0	0	0
an	0	0	1	0	0	0	0	0
apple	0	0	0	1	0	0	0	0
and	0	0	0	0	1	0	0	0
played	0	0	0	0	0	1	0	0
the	0	0	0	0	0	0	1	0
piano	0	0	0	0	0	0	0	1

Figure 2.6 Representation of each word in the sentence

2.3 Feature Extraction and Selection

As we have large unstructured dataset, one of the key operations is to identify the features which can be used to extract meaning result. Suppose we have 23 columns (features) not all the features can help us to detect the final output, so we need to perform certain set of

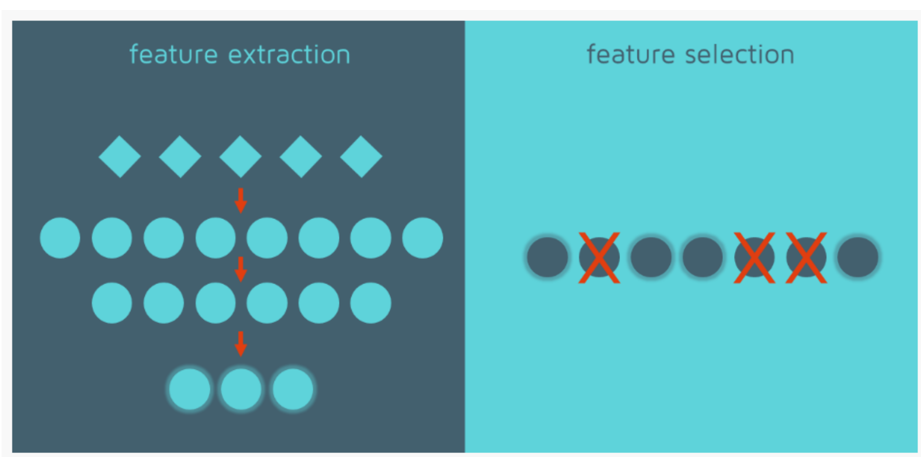

Figure 2.7 Feature Extraction and Selection

3 Chapter 3 : Dataset

There are various features in the dataset like unique identify of book ,it's description, title and anymore.Book_id is the unique id to identify the row in the dataset, a description of the book is given in the book_desc column which I have used to extract information to get the genre of the book,book_isbn code is used to identify the book in the library uniquely,book_review is a column which contains the review given by the users then the last column is a book title. Out of the all the features we have come across the four top useful columns.

	book_id	book_name	genre	summary
0	3248537	Drowned Wednesday	Fantasy	Drowned Wednesday is the first Trustee among ...
1	27796919	The Lost Hero	Fantasy	As the book opens, Jason awakens on a school ...
2	3910776	The Eyes of the Overworld	Fantasy	Cugel is easily persuaded by the merchant Fia...
3	5969644	Magic's Promise	Fantasy	The book opens with Herald-Mage Vanyel return...
4	3173445	Taran Wanderer	Fantasy	Taran and Gurgi have returned to Caer Dallben...
...	...	...	...	...
2995	10372180	White Death	Thriller	A Novel from the NUMA files, A Kurt Austin Ad...
2996	14504372	Venus with Pistol	Thriller	Gilbert Kemp is dealer specializing in antiqu...
2997	3617412	Blackwater	Thriller	"How do you know when you're in too deep? Dav...
2998	11320975	The Rainbow and the Rose	Thriller	The story concerns the life of Johnnie Pascoe...
2999	17227674	Chiefs	Thriller	The First Chief: Will Henry Lee: The novel op...

3000 rows × 4 columns

Figure 3.1 Dataset snapshot

The total number of rows is 3000 and for each row, we have 4 features that I have used in training my model, and then as some new data is introduced based on the past training, model will classify the text into categories. Total number of genres is 5 , by analyzing each summary of book we have classify book among this genre.

genre	book_id	book_name	summary
Crime Fiction	500	500	500
Fantasy	500	500	500
Historical novel	500	500	500
Horror	500	500	500
Science Fiction	500	500	500
Thriller	500	500	500

Figure 3.2 Genre in dataset

4 Chapter 4: Methodology

First we describe the data preprocessing of the data that we have performed in our dataset §4.1, Next we describe the evaluation metrics in §4.2 and lastly we will showcase the experiments we have done for this research in previous §4.3.

4.1 Data Preprocessing

The summary of the dataset is unstructured, one of the important steps involves cleaning of text while dealing with machine learning models. Preprocessing involves removing unnecessary alphabets, tags, and commas, and removing stop words, words that don't provide useful meaning to our data and repeat a number of times are known as stop words, we have cleaned the summary text column and have converted them into lowercase, also have tried to keep the length of text same.

```
0       drowned wednesday is the first trustee among t...
1       as the book opens jason awakens on a school bu...
2       cugel is easily persuaded by the merchant fian...
3       the book opens with herald mage vanyel returni...
4       taran and gurgi have returned to caer dallben ...
                               ...
2995    a novel from the numa files a kurt austin adve...
2996    gilbert kemp is dealer specializing in antique...
2997    how do you know when youre in too deep davey h...
2998    the story concerns the life of johnnie pascoe ...
2999    the first chief will henry lee the novel opens...
Name: summary, Length: 3000, dtype: object
```

Figure 4.1 Text before cleaning

```
0        Drowned Wednesday is the first Trustee among ...
1        As the book opens, Jason awakens on a school ...
2        Cugel is easily persuaded by the merchant Fia...
3        The book opens with Herald-Mage Vanyel return...
4        Taran and Gurgi have returned to Caer Dallben...
                                ...
2995     A Novel from the NUMA files, A Kurt Austin Ad...
2996     Gilbert Kemp is dealer specializing in antiqu...
2997     "How do you know when you're in too deep? Dav...
2998     The story concerns the life of Johnnie Pascoe...
2999     The First Chief: Will Henry Lee: The novel op...
Name: summary, Length: 3000, dtype: object
```

Figure 4.2 Text after cleaning

4.2 Steaming and Lemmatization

After cleaning the text, we have applied steaming into the words, in this technique we try to steam down the words to its root words (for example smelling can be written as smell because eventually here the meaning of the words is not change.

4.3 Stop Words

Frequency function, this function will give us the frequency of words which has occurred so many times and plot the graph for the same. This will help us to remove such words from our text and make it simpler to extract the meaning of text from it.

```
nltk.download('stopwords')

from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))

# function to remove stopwords
def removestopwords(text):
    no_stopword_text = [w for w in text.split() if not w in stop_words]
    return ' '.join(no_stopword_text)

books['summary'] = books['summary'].apply(lambda x: removestopwords(x))
```

Figure 4.3 Remove stop words

```
# print 25 most frequent words
freqwords(books['summary'], 25)
```

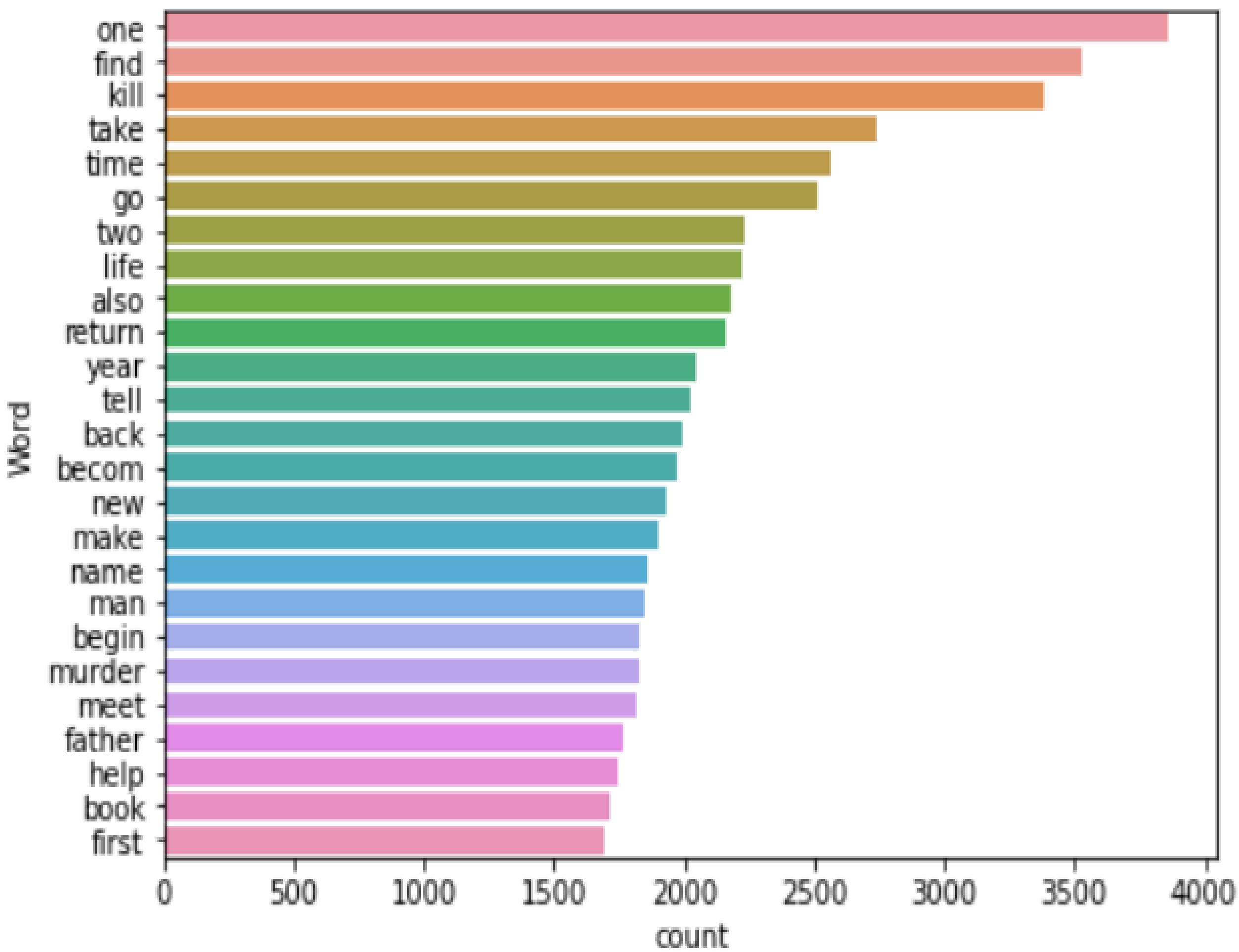

Figure 4.4 Frequency word graph

4.4 Evaluation Metrics

The evaluate and compare the performance of models we have chosen following metrics:

F1-Score

Harmonic means of Precision and Recall; confusion matrix uses a table to describe the performance of machine learning models.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- **Accuracy Score**

  I have also used Accuracy score to compare the performance of classification models.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

4.5 Experiments

4.5.1 KNN

On dividing the dataset in 40-60 ratio of test and train and apply it on KNN model we get f1 score of 0.329 and accuracy score of 0.0632 which is slightly less than Logistic regression and for other ratio that is again on dividing the dataset in 20-80 ratio of test and train and apply it on model we get f1 score of 0.489 and accuracy score of 0.0932.

4.5.2 Logistic regression

On dividing the dataset in 40-60 ratio of test and train and apply it on logistic regression model we get f1 score of 0.337 and accuracy score of 0.080 which is slightly less than Logistic regression and for other ratio that is again on dividing the dataset in 20-80 ratio of test and train and apply it on model we get f1 score of 0.57 and accuracy score of 0.0942.

5 Chapter 5: Proposed Approach

Deep learning is a subpart of machine learning, it has got it's idea by the working of neurons in humans. This architecture of deep learning is called an artificial neural network. In deep learning, we provide a huge amount of data and learning is done through the neural network without human interruptions, the neural network is trained to analyze data given, and neurons are the core where the processing of data takes place. We have three parts input layers where we are feeding the data and an output layer where we are getting the required output in between, there are hidden layers where all the processing is done by neurons. The information is transferred by each layer through the channel which is attached to some values and hence it is called a weighted channel, all neurons have unique number called biased. Biased is added to the weighted sum of input, teaching to neurons, and is applied to a function known as the activation function, which will tell as whether the neurons are active or not, as every activate neuron act as a pipeline of information to next layers this process continue to second last layer, the weight and bias are adjusted continually to get the best output.

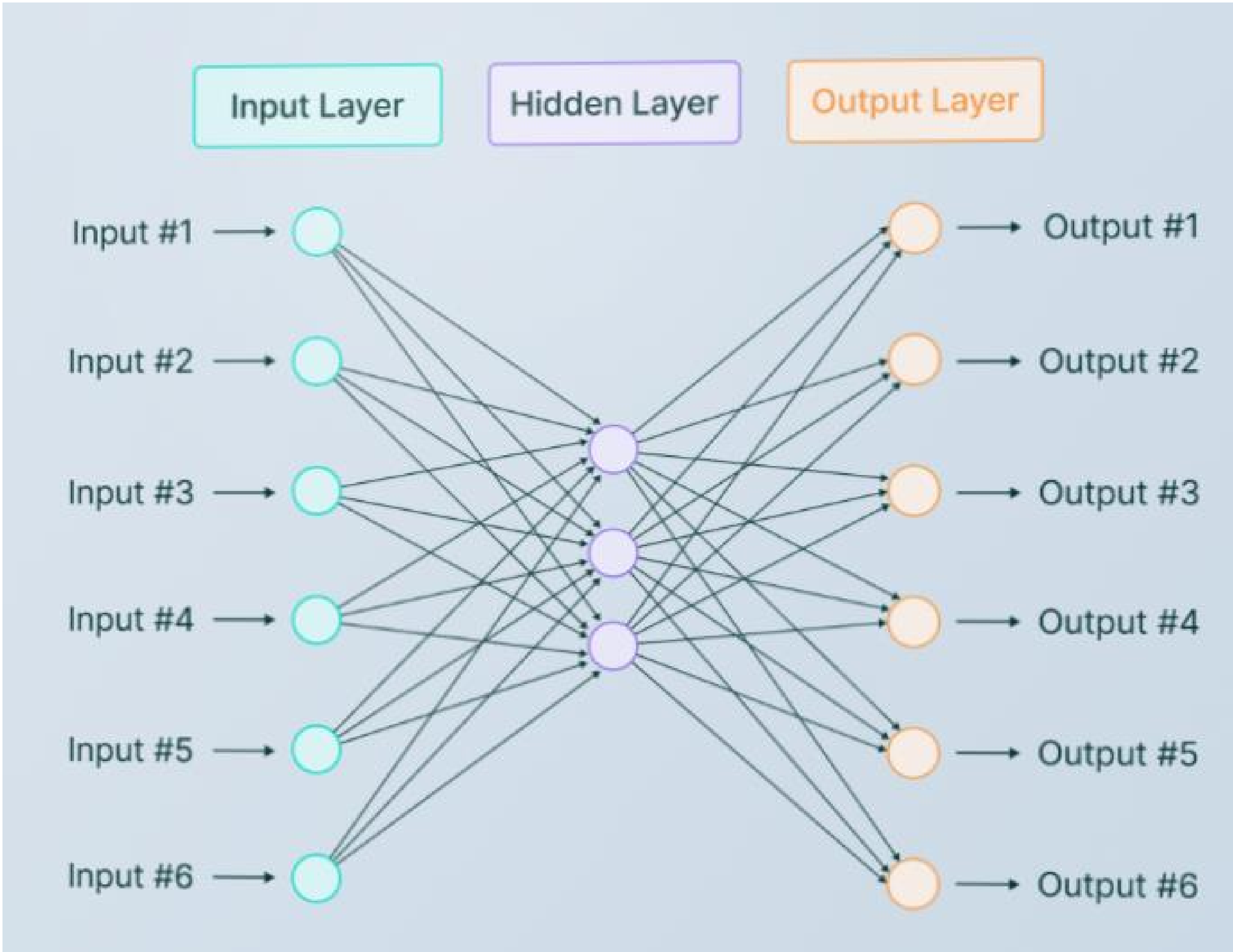

Figure 5.1 Architecture of the deep learning

5.1 Convolutional Neural Network –

CNN is expansion of DNN, it is mainly used for image classification, it permits deep analysis on huge amount of neuron training, for color image we have three primary colors again for each pixel it will build layer and then tarin the layers by layers.

5.2 Recurrent Neural Network –

Feedforward neural network focuses on present node and no memory is require only working in forward direction no loop is present. It can't handle the sequential data as it considers only current state. To overcome the drawback of feedforward neural network, RNN is introduced, in rnn we consider current as well as previous input state, for the starting timestamp we don't have previous state for that we have an extra hidden input state, that has all zeroes, which is supplied to next layer, so for the first layer we have current timestamp and first hidden state which I given as input to first layer that state is called first hidden state. Now current hidden state moves in two direction first for the output state and second as an input to the next state.

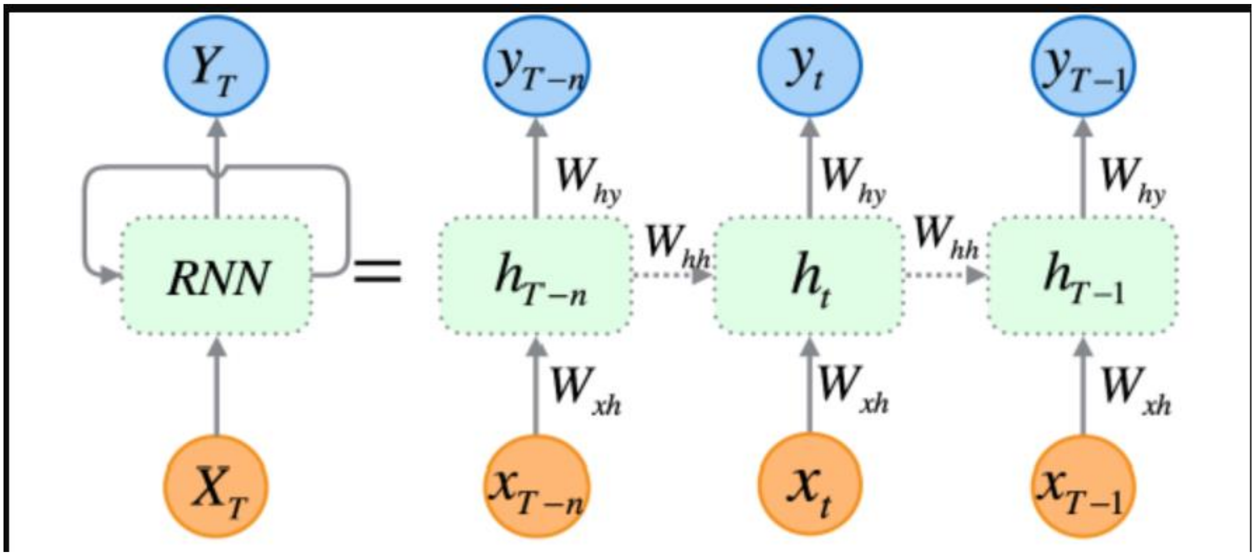

Figure 5.2 Architecture of the recurrent neural network

$$H_t = f(h_{t-1},xt)$$

5.3 LSTM:

The traditional rnn has short term memory they don't remember the data which we have passed at the initial state, they have short term memory like word nearby hence to autocomplete a sentence rnn wouldn't do a good job. Lstm stands for long short-term memory that means when it encounters the new word it is going to forget the old important word and will replace it by new. For that it uses a forget gate and new gate, forget gate we gate the old data and current data if needed to loss the old it will pass by the forget gate and then through new gate the new word will be added.

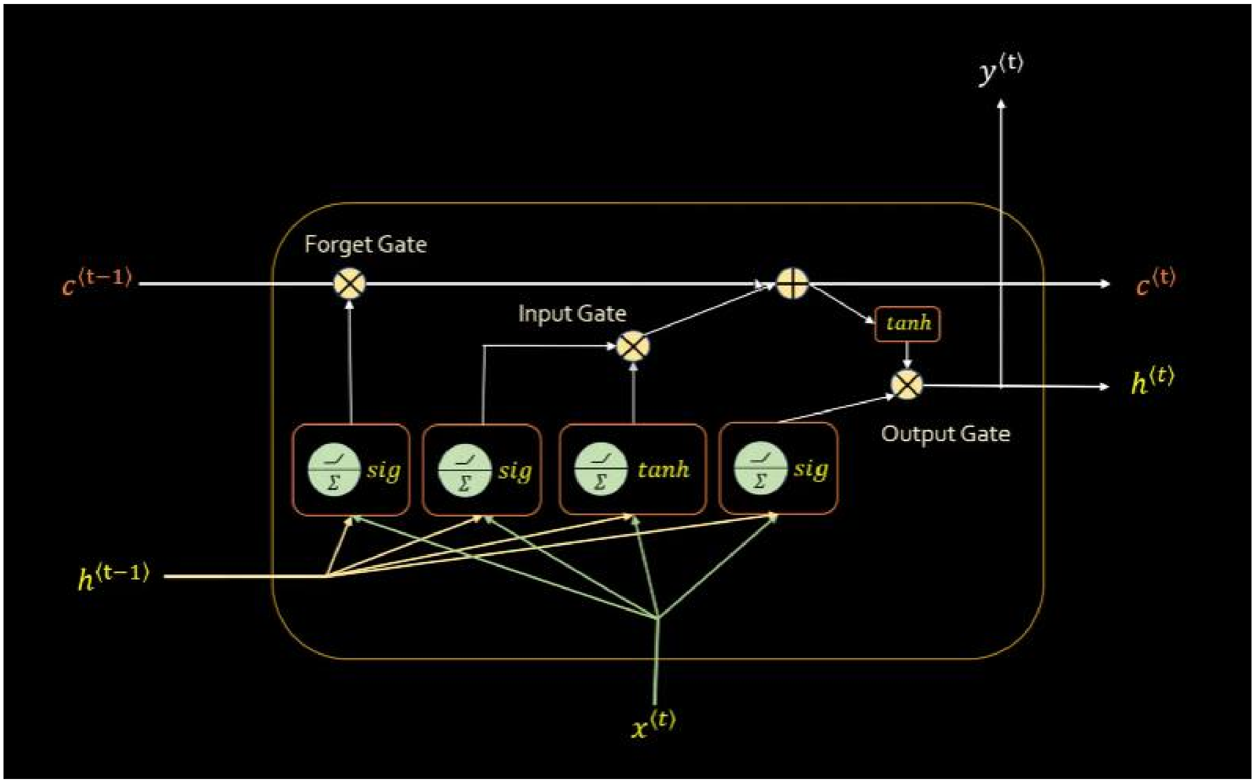

Figure 5.3 Architecture of LSTM

5.3.1 Embedding layer

The objective of embedding layer is to map the sequence of words with the vector as machine understands numbers. In the training phase the machine learns embedding of word with vector. Here I have used sequential model for word embedding. Embedding gives us a dense representation and reduces the dimensionally problem suppose if we use one hot encoding each word will be represent by a set of vectors, and then the number of words, which makes it huge in size to deal in lstm, so to reduce the size word embedding is an important step.

5.3.2 Dropout Layer

This layer dropout some neuron and nullify the effect of some neurons to the next layer. This will make the neural network more effective towards the useful data.

5.3.3 Dense Layer

The neurons in this layer receives the input from each neuron of previous layer that is why it is called dense layer.

5.3.4 Activation Layer

An activation layer is used to activate the neuron at each layer and to determine whether neuron is activated or not, in case we want to check between binary classification sigmoid function perform good. Because Sigmoid has a 0 or 1 output, it can be utilized to remember or ignore information. The data that travels via such LSTM units. An LSTM unit typically has three primary additives, which are labelled in the diagram: The architecture of the LSTM has a unique characteristic that allows it to forget irrelevant data. The LSTM architecture includes a specific feature that allows it to ignore unneeded records. The sigmoid layer uses the inputs h(t-1) and X(t) to determine which elements from the antique output should be retained. Determine and save statistics from the new input X(t) within the mobile state as the next step. A Sigmoid layer decides which of the brand-new facts should be updated and which should be ignored. A tanh layer generates a vector containing all potential values from a new entry. These have been expanded to take the role of the brand-new cellular service. This fresh recollection is then added to c(t-1) to produce c (t).Finally, we must decide what we are going to produce. A sigmoid layer determines which aspects of the cell state will be output. The mobile state was then passed through a tanh to generate all the possible values and multiply it with the result.

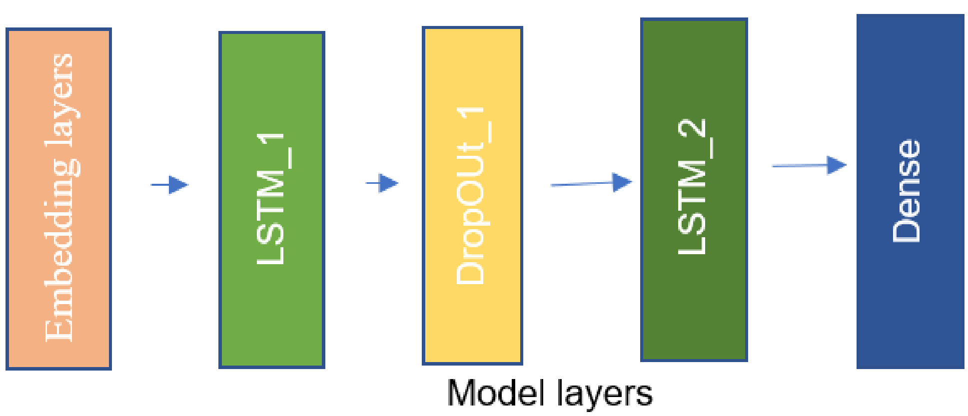

Figure 5.4 Layers of neural network

```
def bookLSTM(x_train, y_train, x_val, y_val, params):
    model = Sequential()
    model.name="Book Model"
    model.add(Embedding(len(params['vocab'])+1, output_dim=x_train.shape[1], input_length=x_train.shape[1]))
    model.add(LSTM(200, return_sequences=True))
    model.add(Dropout(params['dropout']))
    model.add(LSTM(200))
    model.add(Dense(1, activation=params['activation']))
    model.compile(loss=params['loss'],
              optimizer=params['optimizer'],
              metrics=['accuracy'])
    print(model.summary())
    model.fit(x_train,
          y_train,
          validation_data=(x_val, y_val),
          batch_size=params['batch_size'],
          epochs=params['epochs'])
    results = model.evaluate(x_test, y_test, batch_size=params['eval_batch_size'])
    return model

BookModel1 = bookLSTM(x_train, y_train, x_val, y_val, parameters)
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 200, 200)          17123400
_________________________________________________________________
lstm_1 (LSTM)                (None, 200, 200)          320800
_________________________________________________________________
dropout_1 (Dropout)          (None, 200, 200)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 200)               320800
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 201
=================================================================
Total params: 17,765,201
Trainable params: 17,765,201
Non-trainable params: 0
```

Figure 5.5 Layers of Lstm implement

6 Chapter 6: Results and Observations

I have tried two different supervised models - KNN and logistic regression. Following are the F1-score and accuracy score of the models we have observed.

Model	Train%	Test%	F1-score	Accuracy score
KNN	60	40	0.329	0.0632
	80	20	0.489	0.0932
Logistic Regression	60	40	0.337	0.080
	80	20	0.57	0.0947
LSTM	80	20	0.49	0.09290

Table 6.1 Result table

Observations

```
Epoch 1/5
23387/23387 [==============================] - 270s 12ms/step - loss: 0.3686 - accuracy: 0.8447 - val_loss: 0.2129 - val_accuracy: 0.9129
Epoch 2/5
23387/23387 [==============================] - 282s 12ms/step - loss: 0.1535 - accuracy: 0.9476 - val_loss: 0.2410 - val_accuracy: 0.9013
Epoch 3/5
23387/23387 [==============================] - 279s 12ms/step - loss: 0.0735 - accuracy: 0.9771 - val_loss: 0.2077 - val_accuracy: 0.9357
Epoch 4/5
23387/23387 [==============================] - 280s 12ms/step - loss: 0.0284 - accuracy: 0.9924 - val_loss: 0.2512 - val_accuracy: 0.9334
Epoch 5/5
23387/23387 [==============================] - 293s 13ms/step - loss: 0.0161 - accuracy: 0.9957 - val_loss: 0.2815 - val_accuracy: 0.9290
657/657 [==============================] - 3s 5ms/step
```

Figure 6.1 5 Epoch of LSTM

Here I observed that with the increase of epoch the accuracy is decreased that means for new data model is getting outlier. Also, accuracy for train data is increasing that means model is working fine for trained data.so I have decreased the epoch count and keep it 2. In that case it was giving good accuracy.

As a result, two epochs should enough in this scenario

```
Epoch 1/2
23387/23387 [==============================] - 337s 14ms/step - loss: 0.3136 - accuracy: 0.8690 - val_loss: 0.1937 - val_accuracy: 0.9305
Epoch 2/2
23387/23387 [==============================] - 332s 14ms/step - loss: 0.1099 - accuracy: 0.9636 - val_loss: 0.1774 - val_accuracy: 0.9341
657/657 [==============================] - 3s 4ms/step
```

Figure 6.2 2 Epoch of LSTM

7 Chapter 7: Conclusion And Future Work

7.1 Conclusion

As the amount of data grows exponentially in the modern day, the need for text data classification and categorization grows. Machine learning techniques may be useful in resolving this problem. Email filtering, chat message filtering, news feed filtering, and other industries can all benefit from text classification. It has also been observed in locations such as libraries, bookstores, and eBook sites where books are not classified by genre. The main goal of upgrading this point was to categories books by genre utilizing machine learning algorithms and text classification techniques, which will aid in categorizing books by genre using title and abstract. This classification can be used in libraries, bookstores, and other locations.

7.2 Future Work

Using future accessible techniques and new algorithms, text classification in more complicated and unstructured data can be made simple. Feature extraction can be made more exact by assigning appropriate weights. A multilanguage book language categorization can be given in which multiple languages are used. Sort the books by category as well. Local rural language novels that are difficult to categories can be categorized with additional research With a larger amount of complicated and unstructured data, accuracy can be improved. Correct application of new classification methods and techniques

8 References

[1] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Rehan, “A comparative review on deep learning models for text classification,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, pp. 325–335, 2020, doi: 10.11591/ijeecs.v19.i1.pp325-335.

[2] A. Wahdan, S. Hantoobi, S. A. Salloum, and K. Shaalan, “A systematic review of text classification research based on deep learning models in Arabic language,” *Int. J. Electr. Comput. Eng.*, vol. 10, no. 6, pp. 6629–6643, 2020, doi: 10.11591/IJECE.V10I6.PP6629-6643.

[3] “[PDF] Malware classification with LSTM and GRU language models and a character-level CNN | Semantic Scholar.” https://www.semanticscholar.org/paper/Malware-classification-with-LSTM-and-GRU-language-a-Athiwaratkun-Stokes/96e8126ce761700cd857c80cc70746210d32ec20 (accessed May 22, 2022).

[4] “arXiv:1702.07800v1.” https://www.yumpu.com/en/document/view/57067378/arxiv170207800v1 (accessed May 22, 2022).

[5] Mu’adzah, T. L. Ahmad, and A. N. Kusumawati, “L Iterature R Eview,” *J. Bisnis Digit. dan Sist. Inf.*, vol. 1, no. 1, pp. 1–11, 2020.

[6] K. Murphy, “CS340: Machine Learning Naive Bayes classifiers Kevin Murphy 1,” *Mach. Learn.*, pp. 1–17.

[7] M. K. Dalal and M. A. Zaveri, “Automatic Text Classification: A Technical Review,” *Int. J. Comput. Appl.*, vol. 28, no. 2, pp. 37–40, 2011, doi: 10.5120/3358-4633.

[8] R. M. Maertens, A. S. Long, and P. A. White, “Performance of the in vitro transgene mutation assay in MutaMouse FE1 cells: Evaluation of nine misleading (‘False’) positive chemicals,” *Environ. Mol. Mutagen.*, vol. 58, no. 8, pp. 582–591, 2017, doi: 10.1002/em.22125.

[9] C. Baziotis, N. Athanasiou, G. Paraskevopoulos, N. Ellinas, A. Kolovou, and A. Potamianos, “NTUA-SLP at SemEval-2018 Task 2: Predicting Emojis using RNNs with Context-aware Attention,” *NAACL HLT 2018 - Int. Work. Semant. Eval. SemEval 2018 - Proc. 12th Work.*, no. 1, pp. 438–444, 2018, doi: 10.18653/v1/s18-1069.

[10] R. Jindal, R. Malhotra, and A. Jain, “Techniques for text classification: Literature review and current trends,” *Webology*, vol. 12, no. 2, pp. 1–28, 2015.

[11] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF Models for Sequence Tagging,” 2015, [Online]. Available: http://arxiv.org/abs/1508.01991.

[12] Y. Goldberg and O. Levy, “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method,” no. 2, pp. 1–5, 2014, [Online]. Available: http://arxiv.org/abs/1402.3722.

[13] F. P. Shah and V. Patel, “A review on feature selection and feature extraction for text classification,” *Proc. 2016 IEEE Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2016*, pp. 2264–2268, Sep. 2016, doi: 10.1109/WISPNET.2016.7566545.

[14] Nidhi and V. Gupta, “Recent Trends in Text Classification Techniques,” *Int. J. Comput. Appl.*, vol. 35, no. 6, pp. 45–51, 2011.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[16] K. Knight, “Without You Telling It the Answer.,” no. September, 2009.

[17] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should i trust you?’ Explaining the predictions of any classifier,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 1135–1144, 2016, doi: 10.1145/2939672.2939778.

[18] R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning, “Dynamic pooling and unfolding recursive autoencoders for paraphrase detection Distributional Seman...,” *Proc. NIPS 2011*, pp. 1–9, 2011, [Online]. Available: http://papers.nips.cc/paper/4204-dynamic-pooling-and-unfolding-recursive-autoencoders-for-paraphrase-detection.pdf.

[19] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[20] C. Sutton and A. Mccallum, “An Introduction to Conditional Random Fields for Relational Learning,” *Introd. to Stat. Relational Learn.*, 2019, doi: 10.7551/mitpress/7432.003.0006.

[21] M. E. Peters *et al.*, “Deep contextualized word representations,” *NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 2227–2237, 2018, doi: 10.18653/v1/n18-1202.

[22] T. B. Brown *et al.*, “Language models are few-shot learners,” *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, 2020.

[23] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” *Proc. Annu. ACM Conf. Comput. Learn. Theory*, pp. 92–100, 1998, doi: 10.1145/279943.279962.

[24] J. Pennington, R. Socher, and C. Manning, *Glove: Global Vectors for Word Representation*, vol. 14. 2014.

[25] Y. Bengio, “Learning Deep Architectures for AI,” *Foundations*, vol. 2, pp. 1–55, Jan. 2009, doi: 10.1561/2200000006.

[26] Y. Bengio, R. Ducharme, and P. Vincent, “A neural probabilistic language model,” *Adv. Neural Inf. Process. Syst.*, vol. 3, pp. 1137–1155, 2001.

[27] F. Sebastiani, “Machine Learning in Automated Text Categorization,” *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002, doi: 10.1145/505282.505283.

[28] Q. Hoang, “Predicting Movie Genres Based on Plot Summaries,” pp. 1–12, 2018, [Online]. Available: http://arxiv.org/abs/1801.04813.

[29] H. Wu, Y. Liu, and J. Wang, “Review of text classification methods on deep learning,” *Comput. Mater. Contin.*, vol. 63, no. 3, pp. 1309–1321, 2020, doi: 10.32604/CMC.2020.010172.

[30] https://monkeylearn.com/text-classification/