Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 10, Issue. 3, March 2021, pg.14 – 25

Book Genre Categorization Using Machine Learning Algorithms (K-Nearest Neighbor, Support Vector Machine and Logistic Regression) using Customized Dataset

Parilkumar Shiroya[1]; Darshan Vaghasiya[2]; Meet Soni[3]; Vrajkumar Patel[4]; Brijeshkumar Y. Panchal[5]

[1]Student, CSE Department, PIT, Parul University, Vadodara, India
[2]Student, ICT Department, PIT, Parul University, Vadodara, India
[3]Student, ICT Department, PIT, Parul University, Vadodara, India
[4]Student, ICT Department, PIT, Parul University, Vadodara, India
[5]Assistant Professor, CSE Department, PIT, Parul University, Vadodara, India
parilshiroya84@gmail.com, vaghasiadarshan95@gmail.com, meetsoni784@gmail.com, vasupatel1930@gmail.com, panchalbrijesh02@gmail.com

DOI: 10.47760/ijcsmc.2021.v10i03.002

Abstract— Text classification is playing a vital role in current era. Its requirement is increasing day by day because of increase of text data as number of digital users are increasing rapidly. As a result, machine learning algorithms are used to classify certain text data, resulting in better predictions and accuracy. By constructing a data set with proper structure and data, the genre is predicted by the title and abstract of the book. The dataset will consist books which are translated to English from Guajarati or Hindi originate books. In this paper, some weaknesses in text classification techniques are analysed and worked on to improve the accuracy of structured data. The main focus here was to classify a book by genre using machine learning algorithms.

Keywords— Text Classification, Book Categorization, K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Logistic Regression (LR), Text Mining, Machine Learning, Genre Prediction.

I. INTRODUCTION

Machine learning is used to teach machines how to handle the data more efficiently. The text mining studies are gaining more importance recently because of the availability of the increasing number of the electronic documents from a variety of sources [3]. Most text classification can be unravelled into the following phases: Data pre-processing, Text cleaning, Feature Selection, training model, assigning classifiers and evaluating the output. Nowadays, most problem faced in libraries are classification of genre the book lies on. There are many books which are not classified by its genre which makes librarians and reader difficult to classify the book. To classify genre of this books, prediction of genres is made based on the book title and summaries. The goal is to create a model that can determine how representative a title is of its genre. And by the way, it is very difficult

for even a human to distinguish between books of different categories. Furthermore, a dataset will be used which contains title, writer name, book dialects, its sort and dynamic. This dataset will be utilized to group and foresee the class of the book. The dataset will incorporate books which are been meant English from Gujarati and Hindi. it will utilize three distinctive ML algorithms and discover contrast of one another's exactness and expectation yield to improve results. The motivation was to getting a proper genre categorized books collections which will make user easy to get classified books as per its genre requirement. And also, can be used in big books stores and libraries to organize books according to their requirement.

II. LITERATURE REVIEW

In natural language processing, text classification has always been an interesting topic. Traditional machine learning-based text classification approaches have a number of drawbacks, including dimension explosion, data sparsity, and limited generalisation ability [2]. Text classification can be divided in two stages. Stages are training and testing. During the training phase, the documents are pre-processed and are conditioned by a learning algorithm to create a classifier. Validation of the classifier is done in the testing stage. Support Vector Machines (SVM), K-Nearest Neighbor (K-NN), Logistic Regression (LR), and other conventional learning algorithms can all be used to train the data [7].

Generally, ML algorithms are categorised by supervised, unsupervised and semi-supervised. In first, the network is presented with the right response for each input pattern in the supervised learning technique. Furthermore, unsupervised learning does not involve a correct response to each input pattern in the training data set. Additionally, semi-supervised learning is a hybrid of labelled and unlabelled data [1].

K-nearest Neighbor algorithm (KNN) is the simplest method for determining the class of unlabelled documents and is a common non-parametric method. However, due to the high dimensions, the computational time increases as a result, this approach is not ideal for such documents [10]. K-NN algorithm performed better as more local text characteristics are considered, however, classification time is long and it is difficult to find an optimum value of k [9]. LR and SVM both offer an acceptable and easy result in four different datasets compared to eight other ML algorithms and different extraction techniques [7]. Several algorithms or combinations of algorithms as hybrid methods have been suggested for automated text classification. Among these algorithms, SVM, NB, KNN and their hybrid scheme are seen to be most suitable in the current literature, with the combination of various other algorithms and feature selection techniques [8]. When the data collection is big, the error of classification tends to be less. It was also recognised that the collection of appropriate algorithms for a given dataset plays a key role in the classification of text [5]. The accuracy of the classification algorithm is significantly influenced by the consistency of the data source. Irrelevant and redundant data features not only increase the cost of the mining operation, but also reduce the quality of the outcome in certain cases [4]. It is observed that for the given classification system, the classification efficiency of the classifiers on the basis of various data sets, the corpuses are different. Various algorithms behave differently depending on the data collection [6]. In certain cases, using knowledge engineering techniques and expert opinions to define a set of logical rules to classify documents will help to simplify the classification task [5].

This, above all else, is a review of text classification. It also contains information on various methods of machine learning algorithms, as well as a few of their characteristics. The above data also demonstrates issues that can arise by using text classification algorithms, such as high dimensional explosion data parity, and so on. If the data is not properly relevant, it may not be properly classified. Different feature extraction techniques can have an impact on data classification in certain cases.

Advantages:

- Results of short text classification were good in K-NN and SVM. And KNN showed the best accuracy.
- In the supervised techniques, support vector machines achieve the highest performance.
- K-Nearest Neighbor is Effective for text data sets and Non-parametric.
- More local characteristics of text or document are considered in K-Nearest Neighbor.
- SVM can model non-linear decision boundaries.
- SVM Performs similarly to logistic regression when linear separation and Robust against overfitting problems.
- Logistic regression is easier to implement, interpret, and very efficient to train.
- It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions.

Disadvantages:

- In Support Vector Machine Lack of transparency in results caused by a high number of dimensions (especially for text data).
- Computational of K-Nearest Neighbor model is very expensive and Difficult to find optimal value of k.

- Finding a meaningful distance function is difficult for text data sets.
- LR constructs linear boundaries.
- Logistic Regression requires average or no multicollinearity between independent variables.

Limitations:

- The logistic regression will not be able to handle a large number of categorical features.
- With a new sample, you have to specify K.
- KNN doesn't learn any mode.
- Choosing a "good" kernel function is not easy.

III.RESEARCH METHODOLOGY

A. Dataset

For experimental purpose, two different datasets were used. First dataset used was CMU book summary dataset. This dataset contains plot summaries for 16,559 books extracted from Wikipedia, along with Freebase aligned metadata, including author, title, and genre. Second dataset was created from data extracted from various sites. This dataset includes book title, Language, Author name, Genre and Abstract of books. Dataset includes about books which were been translated from Gujarati and Hindi to English. There were about books included in the dataset.

	Book Name	Language	Author	Genre	Abstract
0	The desk	English, Gujarati	Ketki Shah	Culture, Personal Development	This book is based solely on the thoughts of a...
1	Jalebi Curls	English, Gujarati	Niveditha Subramaniam	Children's Literature, Fiction	The raja loves jalebis. He even dreams of them...
2	One Life Is Not Enough	English, Gujarati	K. Natwar Singh	Literature, Personal Development	Natwar Singh joined the Indian Foreign Service...
3	Dosa	English, Gujarati	Sandhya Rao	Children's Literature, Fiction	Who ate the dosas? Amma makes dosas but they k...
4	Deepak's Diwali	English, Gujarati	Divya Karwal	Children's Literature, Fiction	It's not shaping up to be a good Diwali so far...
5	Ali Baba and the Forty Thieves	English, Gujarati	Enebor Attard	Children's Literature, Fiction	In a village in the Arabian Desert, a band of ...
6	Dragon's Tears	English, Gujarati	Manju Gregory	Fiction	When Chun Li releases a golden fish, he is rew...
7	Ashoka	English, Hindi	Meena Talim	History, Children's Literature	A power-hungry warrior and a peace-loving wife...
8	Chanakya Mantra	English, Hindi	Ashwin Sanghi	History, Personal Development	The year is 340 BC. A hunted, haunted Brahmin ...
9	Bharat Ki Murtikala Ki Kahani	English, Hindi	Bhagwatsharan Upadhyay	Arts, Culture, Literature	Swadesh-Paricha Mala - India is a great countr...

Fig. 1 [Dataset]

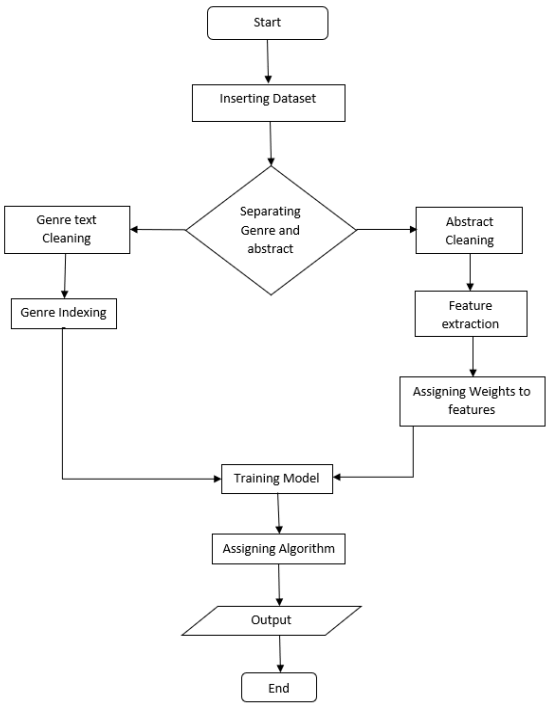

Fig. 2 [Flow Diagram]

B. Data Pre-Processing

Here libraries like tqdm, pandas, matplotlib, etc. are used for pre-processing of data. The pre-processing of data stages like data loading, reading, splitting, counting, labializing. After all this processing the processed data is been cleaned by removing unwanted characters and words which will be not used to classify data. For example: the words and characters like 'the', 'is', '5', '&' etc cannot be used to classify any data. Other characters are can be used for classification.

```
data = []
with open("F:/aa Project/Dataset/Book1.txt", 'r') as f:
    reader = csv.reader(f, dialect='excel-tab')
    for row in tqdm(reader):
        data.append(row)
```

```
51it [00:00, 12777.58it/s]
```

```
book_name = []
Language = []
author = []
genre = []
abstract = []

for i in tqdm(data):
    book_name.append(i[0])
    genre.append(i[3])
    abstract.append(i[4])


books = pd.DataFrame({'book_name': book_name, 'genre': genre, 'abstract': abstract})
books.head(5)

books.shape
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 51/51 [00:00<?, ?it/s]
(51, 3)
```

Fig. 3 [Data Pre-processing]

C. Genre Graph-plotting

Here counting and plotting number of genres allocated in the dataset is done and then shown the counting in graph form. By using sns function from seaborn library to labialize the genre. And also used matplotlib library to plot graph of genre counts in dataset.

```
g = genres_df.nlargest(columns="Count", n = 50)
plt.figure(figsize=(12,15))
ax = sns.barplot(data=g, x= "Count", y = "genre")
ax.set(ylabel = 'Count')
plt.show()
```

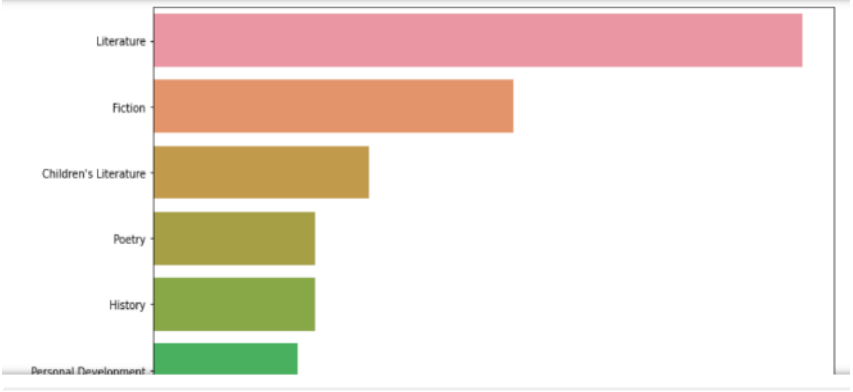

Fig. 4 [Genre Graph Plotting]

D. Abstract Cleaning

Here re library is used to remove unwanted spacing and remove all unwanted characters also used inbuilt stop word function from nltk corpus library to remove stop words from abstract. i.e., if sentence is" Books are important to read" will become "Books important read".


```
def clean_text(text):
    text = re.sub("\'", "", text)
    text = re.sub("[^a-zA-Z]"," ",text)
    text = ' '.join(text.split())
    text = text.lower()

    return text
```

```
books['clean_abstract'] = books['abstract'].apply(lambda x: clean_text(x))

books[['abstract', 'clean_abstract']].sample(3)
```

	abstract	clean_abstract
35	Book Summary of Soordas :Mystic Saints Of Indi...	book summary of soordas mystic saints of india...
17	The ballads of Rajput prowess, the aphorisms o...	the ballads of rajput prowess the aphorisms of...
34	India entered the 21st century with multiple c...	india entered the st century with multiple cha...

Fig. 5 [Abstract Cleaning]

E. Feature Extraction

TfidVectorizer function from sklearn library is used to extract features from abstract and assigning weights to the feature values. for example, if there are 5 word" play cricket football basketball" then this function will assign weight of play as half of the other weights because the term pay is repeated twice. Not only TfidfVectorizer, any feature extraction techniques can be used for extraction of feature values from data like bag of words, etc.

```
def freq_words(x, terms = 30):
  all_words = ' '.join([text for text in x])
  all_words = all_words.split()

  fdist = nltk.FreqDist(all_words)
  words_df = pd.DataFrame({'word':list(fdist.keys()), 'count':list(fdist.values())})

  d = words_df.nlargest(columns="count", n = terms)
  plt.figure(figsize=(12,15))
  ax = sns.barplot(data=d, x= "count", y = "word")
  ax.set(ylabel = 'Word')
  plt.show()
```

```
freq_words(books['clean_abstract'], 100)
```

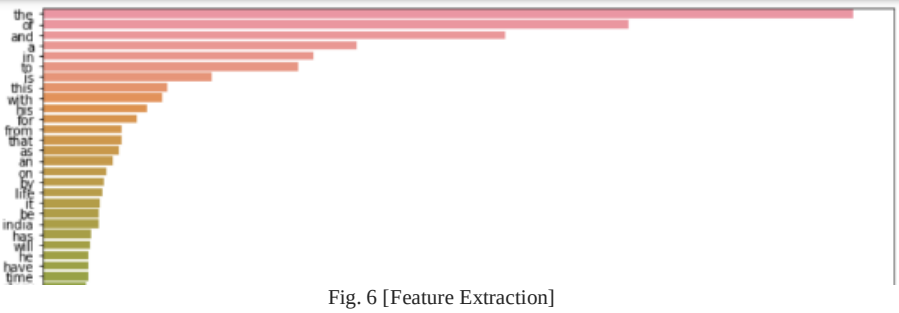

Fig. 6 [Feature Extraction]

F. Training model with Algorithms

Here imported train_test_spllit function from sklearn.model_selection function to train model. this model includes xtrain value as the features extracted from clean abstract and xval as genres of book to be classified. This trained model uses Tfidfvectorizer to assign weight of features to genre. After that assigned machine learning algorithm's classifier from their respective libraries. With use of that algorithm tried to predict the genre of the book.


```
In [16]: from sklearn.feature_extraction.text import TfidfVectorizer
         tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=10000)

In [17]: from sklearn.model_selection import train_test_split
         xtrain, xval, ytrain, yval = train_test_split(books['clean_abstract'], books['genre'], test_size=0.2)

In [18]: xtrain_tfidf = tfidf_vectorizer.fit_transform(xtrain)
         xval_tfidf = tfidf_vectorizer.transform(xval)

In [19]: from sklearn.neighbors import KNeighborsClassifier

In [20]: knn = KNeighborsClassifier(n_neighbors=7)

In [21]: knn.fit(xtrain_tfidf, ytrain)
Out[21]: KNeighborsClassifier(n_neighbors=7)

In [22]: y_pred = knn.predict(xval_tfidf)
```

Fig. 7 [Training model and Assigning Algorithm]

G. Output & Details of ML Algorithms

There is a function which used inverse functions of TfidfVectorizer and try to convert the predicted output into a proper genre. After all this the predicted genre and the actual genre will be printed with book names.

K-Nearest Neighbor (K-NN):

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on the Supervised Learning methodology. The K-NN algorithm assumes the similarities between the new case/data and the available cases and places the new case in the category that is more similar to the available categories. K-NN algorithm stores all of the available data and classifies a new data point based on similarities. This means that new data can be rapidly grouped into a well-defined group using the K-NN algorithm. It uses distance formulas such as the Euclidean Distance Formula and the Manhattan Formula to determine similarity. The distance between data points is determined using the K-NN algorithm. We use the basic Euclidean Distance theorem for this.

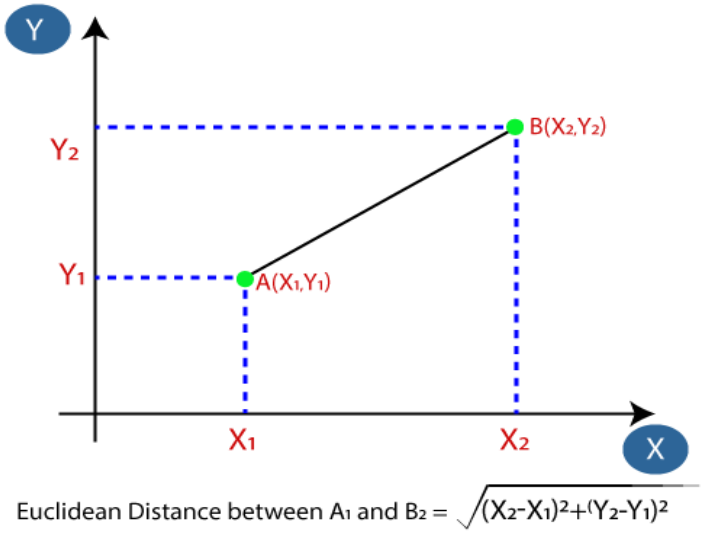

Fig. 8 [Euclidean Distance Formula][13]

The K-NN method functions as follows: Calculate the Euclidean distance of K number of neighbors after choosing the number K of neighbors. Then, based on the measured Euclidean distance, choose the K nearest neighbor. Count the number of data points in each group among these k neighbors. Finally, add the latest data points to the segment in which the neighbor's number is the highest. No predefined methods are there to find optimal value of K in K-NN algorithm. As it can only say that low value of k can be sometime unproper to find proper precision output. As it gets high optimal k value the prediction will be sorted properly. And also, k value should mostly be odd because if k value is in even terms then there will be issue in classifying if probability percentage are same in similar classes.


```
def infer_tags(q):
    q = clean_text(q)
    q = remove_stopwords(q)
    q_vec = tfidf_vectorizer.transform([q])
    q_pred = knn.predict(q_vec)
    return q_pred
```

```
for i in range(5):
    k = xval.sample(1).index[0]
    print("Books: ", books['book_name'][k], "\nPredicted genre: ",
```

```
Books:  Chanakya Mantra
Predicted genre:  ['Arts, Culture, Literature']
Actual genre: History, Personal Development

Books:  Jhansi Ki Rani
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Biography, Literature

Books:  Tantrasangraha of Nilakantha Somayaji
Predicted genre:  ['Biography, Literature']
Actual genre: History, Culture, Literature

Books:  Jhansi Ki Rani
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Biography, Literature

Books:  Jhansi Ki Rani
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Biography, Literature
```

Fig. 9 [K-NN Output]

Support Vector Machine (SVM):

SVM stands for Support Vector Machine which can be used for both regression and classification. However, it is commonly used in classification goals. Support vectors are data points that are relatively close to the hyperplane and have an impact on the hyperplane's direction and alignment. SVM is useful because it can control both continuous and classified variables.

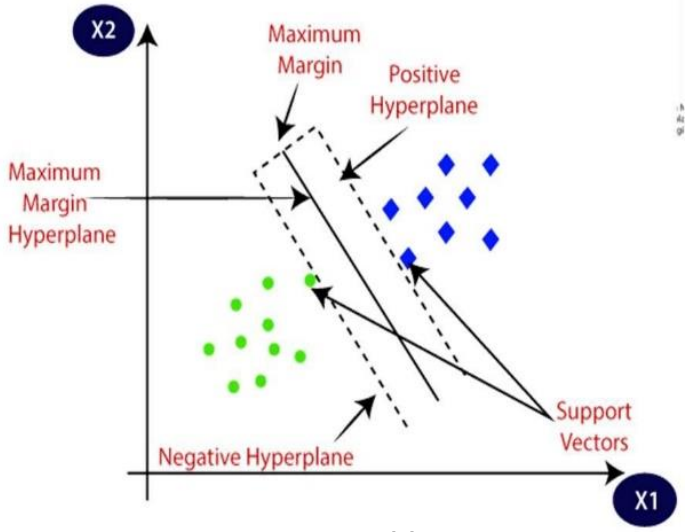

Fig. 10 [SVM][11]

In multidimensional space, an SVM model is simply a representation of different classes in a hyperplane. SVM can create the hyperplane in an iterative way in order to reduce the error. SVM splits datasets into classes in order to find the optimal marginal hyperplane. With respect to the support vectors, the SVM is used to construct a Hyperplane.

The following equation can be used to describe the Hyperplane:

$$y = w\,x + b$$

The data points of the classes are used to construct the support vectors. The generated hyperplane would be used as a classifier, dividing data points into classes based on which planes they lie on.


```
def infer_tags(q):
    q = clean_text(q)
    q = remove_stopwords(q)
    q_vec = tfidf_vectorizer.transform([q])
    q_pred = clf1.predict(q_vec)
    return q_pred
```

```
for i in range(5):
    k = xval.sample(1).index[0]
    print("Books: ", books['book_name'][k], "\nPredicted genre: ",
```

```
Books:  A History of Hindi LiteratureÊ
Predicted genre:  ['Poetry, Literature']
Actual genre: History, Literature

Books:  The Unshaken Mind
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Literature, Personal Development, Fiction

Books:  Lullabies of the World
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Children's Literature, Literature

Books:  Lullabies of the World
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Children's Literature, Literature

Books:  The Bloodstained Throne
Predicted genre:  ['Poetry, Literature']
```

Fig. 11 [SVM Output]

The data points of the classes are used to construct the support vectors. The generated hyperplane would be used as a classifier, dividing data points into classes based on which planes they lie on.

Logistic Regression:

Logistic regression is a supervised classification algorithm and a binary classifier. This regression is generally used to separate data into two classes. Multinomial logistic regression can be used to classify data into three or more classes. Logistic Regression is a model that is been formed with the use of Logistic Function. The Sigmoid Function is another name for this Logistic function. This function is used to squish data that is in the range of 0 to 1, or [0,1].
The Logistic Function is as follows:

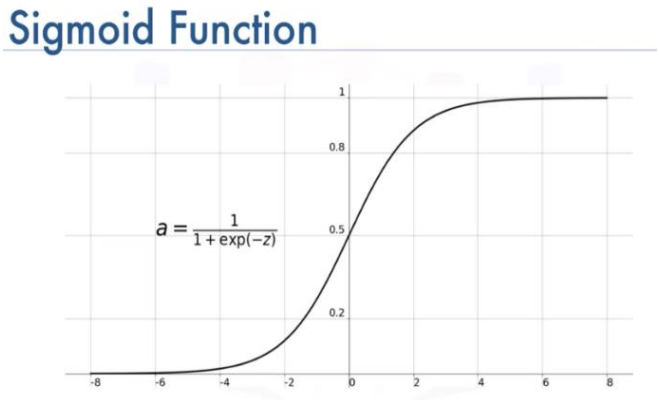

Fig. 12 [Sigmoid Function][12]

This function will classify data from 0 to 1. If the probability lies between 0 to 0.5 then it will classify the data into negative class and if the probability lies between 0.5 to 1, then it will classify the data into positive class.


```
def infer_tags(q):
    q = clean_text(q)
    q = remove_stopwords(q)
    q_vec = tfidf_vectorizer.transform([q])
    q_pred = clf.predict(q_vec)
    return q_pred
```

```
for i in range(5):
    k = xval.sample(1).index[0]
    print("Books: ", books['book_name'][k], "\nPredicted genre: ", infer_
```

```
Books:  ChandrakantaÉ
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: History, Literature, Fiction

Books:  ChandrakantaÉ
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: History, Literature, Fiction

Books:  ChandrakantaÉ
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: History, Literature, Fiction

Books:  Soordas
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Poetry, Literature

Books:  Says Kabir: A Collection of the Hundred and Ten Poems of Kabir
Predicted genre:  ["Children's Literature, Literature, Fiction"]
Actual genre: Poetry, Literature
```

Fig. 13 [LR Output]

H. Result Evolution

The result shows that there is a large difference in accuracies from both datasets. Accuracy result of first dataset were 2.68%, 9.53%, 7.27% in KNN, LR and SVM respectively. And results in second dataset was 45.45% accurate in both KNN and LR while SVM accuracy was 54.54%.

SVM Algorithm:

```
from sklearn import svm
from sklearn.svm import LinearSVC
```

```
clf1= LinearSVC(random_state=0)
```

```
clf1.fit(xtrain_tfidf, ytrain)
```

```
LinearSVC(random_state=0)
```

```
y_pred = clf1.predict(xval_tfidf)
```

```
y_pred[5]
```

```
"History, Children's Literature, Literature"
```

```
from sklearn import metrics
print (metrics.accuracy_score(yval, y_pred))
```

```
0.0727272727272727
```

Fig. 14 [CMU Dataset SVM Output]

```
from sklearn import svm
from sklearn.svm import LinearSVC
```

```
clf1= LinearSVC(random_state=0)
```

```
clf1.fit(xtrain_tfidf, ytrain)
```

```
LinearSVC(random_state=0)
```

```
y_pred = clf1.predict(xval_tfidf)
```

```
y_pred[5]
```

```
'Poetry, Literature'
```

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print(accuracy_score(yval, y_pred))
```

```
0.5454545454545454
```

Fig. 15 [2nd Dataset SVM Output]

LR Algorithm:

```
OneVsRestClassifier(estimator=LogisticRegression())

y_pred = clf.predict(xval_tfidf)

y_pred[5]

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0])

multilabel_binarizer.inverse_transform(y_pred)[1]

()

from sklearn import metrics
print (metrics.accuracy_score(yval, y_pred))

0.09536784741144415
```

Fig. 16 [CMU Dataset LR Output]

```
lr = LogisticRegression()
clf = OneVsRestClassifier(lr)

clf.fit(xtrain_tfidf, ytrain)

OneVsRestClassifier(estimator=LogisticRegression())

y_pred = clf.predict(xval_tfidf)

y_pred[9]

"Children's Literature, Literature, Fiction"

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print(accuracy_score(yval, y_pred))

0.45454545454545453
```

Fig. 17 [2nd Dataset SVM Output]

K-NN Algorithms:

```
knn.fit(xtrain_tfidf, ytrain)

KNeighborsClassifier(n_neighbors=7)

y_pred = knn.predict(xval_tfidf)

y_pred[5]

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0])

multilabel_binarizer.inverse_transform(y_pred)[1]

('Fantasy', 'Science Fiction', 'Speculative fiction')

from sklearn import metrics
print (metrics.accuracy_score(yval, y_pred))

0.026858699883223043
```

Fig. 18 [CMU Dataset K-NN Output]

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=7)

knn.fit(xtrain_tfidf, ytrain)
KNeighborsClassifier(n_neighbors=7)

y_pred = knn.predict(xval_tfidf)

y_pred[8]
'Biography, Literature'

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print(accuracy_score(yval, y_pred))
0.45454545454545453
```

Fig. 19 [2nd Dataset K-NN Output]

TABLE I

Model	**Train %**	**Test %**	**CMU Dataset Accuracy**	**2nd Dataset Accuracy**
KNN (N=7)	80	20	2.68 %	45.45 %
LR	80	20	9.53 %	45.45 %
SVM	80	20	7.27 %	54.54 %

Result shows that the SVM have the highest accuracy compared to KNN and LR in the second dataset and LR have the highest accuracy in first dataset. The large difference of the accuracy in both data set can have many reasons. This reason can be explained as followed. Assigning weight to the features can sometime makes issues in classifications. If the number of features is more than the selected features then classification of the genre will not be predicted properly. For example: if the number of features is about 15000 and the selection of feature limit is 10000 then the remaining 5000 feature wont ne used in classification. The spelling mistakes in data inserted can also make issues in classification. The wrong spelling cannot be removed in cleaning which will also be counted as a feature in classification. For example: if there is a word name 'thhhe' which original idea is 'the' which should be removed in stop word would not be removed while cleaning and would be counted as a feature in genre classification. As CMU dataset has more data with more genre classes, the classification of genre will get more complicated. The more data will lead to create more feature values which will not be considered as the vectorizers have feature value limits. More number of genres will make prediction probability low which tends to make class selection difficult.

IV. CONCLUSION

In current era, Classification and categorization of text data requirement is increasing from time to time as the dramatic increase in the data. To solve this issue, machine learning algorithms can play a vital role in it. Text classification can be used in fields like email filtering, chat message filtering, news feed, etc. It has also been seen places like libraries, book stores and eBook sites where books are not been categorized by its genre. By revising this point, the main aim here was to classify the books by its genre using machine learning algorithm and text classification techniques which will help to categorize books by its genre using title and abstract. This classification can be used in places like libraries, book stores, etc. to organize and categorize books as per there requirement. Three algorithms were selected for classification of genre. These algorithms were Logistic Regression (LR), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). In start, libraries and a dataset were added to the model. Following that, data was cleaned and then from it feature values were extracted from it. Then using ML classifiers, the gerne and feature values where inserted in training model. With all of this, books were classified into different genres. The result of first dataset was 2.96%, 9.09%, 27.27% accurate in first dataset and 8.18%, 27.27%, 36.36% accuracy in KNN, LR, SVM respectively in second dataset. This difference in result is because of complex and unstructured data. And with increase in feature values and genres the accuracy of prediction decreased. But as per the results it has shown that classification with SVM was most accurate and fast in processing and predicting output.

FUTURE WORK

Classification of text can be easy in more complex and unstructured data by using future available techniques and new algorithms. Feature extraction can be made more precise and with proper weights assigned to it. Multilanguage book language classification can be proposed in which with use of more than one language can also classify the books by its genre. Local rural language books which are hard to classify can be classified with further researches. Accuracy can be increased with a greater number of complex and unstructured data with proper use of new classification techniques and algorithms.

REFERENCES

[1] P.V. Arivoli, T. Chakravarthy "Document Classification Using Machine Learning Algorithms", IJSER, 23473878, 2015.

[2] Hongping Wu, Yuling Liu, and Jingwen Wang "Review of Text Classification Methods on Deep Learning", Computers, Materials & Continua CMC, vol.63, no.3, pp.1309-1321, 2020.

[3] Khushbu Khamar "Short Text Classification Using kNN Based on Distance Function", IJARCCE, Vol.3, Issue 4, 2013.

[4] Gamal, D. Alfonse, M. El-Horbaty, E.S.Salem, A.B. "A comparative study on opinion mining algorithms of social media statuses ". In Proceedings of the Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017; pp. 385–390

[5] Bafna P, Pramod D, Vaidya A," Document clustering: TF IDF approach ", IEEE int. conf. on electrical, electronics, and optimization techniques (ICEEOT). pp 61–66

[6] S. Wang and H. Wang, "A Knowledge Management Approach to Data Mining", Industrial Management and Data Systems, vol. Vol. 108, No. 5, pp. 622634, 2008

[7] Amey K. Shet Tilve, Surabhi N. Jain "A Survey on Machine Learning Techniques for Text Classification", International Journal Engineering Science and Research Technology, 2017.

[8] Mrs. B. Meena Preethi, Dr.P. Radha," A Survey Paper on Text Mining - Techniques, Applications and Issues", IOSR Journal of Computer Engineering (IOSR-JCE), 62.86 | 3.791,2019

[9] R. Manikandan, Dr. R Sivakumar "Machine learning algorithms for text-documents classification", International Journal of Academic Research and Development ISSN: 2455-4197, 2018.

[10] Nidhi, Vishal Gupta "Recent Trends in Text Classification Techniques", International Journal of Computer Applications (0975 – 8887), 2011.

[11] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[12] https://medium.com/@toprak.mhmt/activation-functions-for-deep-learning-13d8b9b20e

[13] https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[14] http://www.cs.cmu.edu/~dbamman/booksummaries.html

[15] Prateek Joshi, Predicting Movie Genres using NLP-An Awesome Introduction to Multi-label Classification, available:- https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/

[16] Akshay Bhatia, Book-Genre-Classification, available at: https://github.com/akshaybhatia10/Book-Genre-Classification

