DETECTION OF NEAR DUPLICATE THREADS IN QUESTION AND ANSWER WEBSITE

Project thesis submitted in partial fulfillment of major project for VIII semester

B. Tech in Computer Science and Engineering

BY

B Sai Pranay Kumar(17131A0525)

B Sai Aiswarya(17131A0526)

B Mano Rachna(17131A0527)

B Sruthi(17131A0529)

Under the Esteemed Guidance of

Mrs Deepika Sonal

Assistant Professor

Department of Computer Science and Engineering

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING (Autonomous)

Approved by AICTE, New Delhi and Affiliated to JNTU-Kakinada

Re-accredited by NAAC with "A" Grade with a CGPA of 3.47/4.00

Madhurawada, Visakapathanam-530 048

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING (Autonomous)
Approved by AICTE, New Delhi and Affiliated to JNTU-Kakinada
Re-accredited by NAAC with "A" Grade with a CGPA of 3.47/4.00
Madhurawada, Visakapathanam-530 048

CERTIFICATE

This is to certify that the project thesis entitled "Detection Of Near Duplicate Threads In Question and Answer Website" being submitted by

B Sai Pranay Kumar(17131A0525)
B Sai Aiswarya(17131A0526)
B Mano Rachna(17131A0527)
B Sruthi(17131A0529)

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University, Kakinada is a record of bonafied work carried out under my guidance and supervision.

The results embodied in this project thesis have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Guide
Mrs. Deepika Sonal
Assistant Professor
Department of CSE
GVPCOE (A)

Head of the Department
Dr. P. Krishna Subba Rao
Professor & Head of Department
Department of CSE
GVPCOE (A)

ACKNOWLEDGEMENT

We thank **Prof. A. B. Koteswara Rao**, principal, **Gayatri Vidya Parishad College of Engineering (Autonomous)** for extending his utmost support and cooperation in providing all the provisions for the successful completion of the project.

We consider it our privilege to express our deepest gratitude to **Dr. P. Krishna Subba Rao**, Professor, Head of Department, Computer Science and Engineering for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

We would like to thank our guide **Mrs. Deepika Sonal** and our project coordinator **Mr. Om Prakash** for helping us throughout the development of project. We are thankful for their support, guidance and valuable suggestions.

We also thank all the members of the staff in Computer Science Engineering for their sustained help in our pursuits.

We thank all those who contributed directly or indirectly in successfully carrying out this work.

Project Members

B Sai Pranay Kumar(17131A0525)

B Sai Aiswarya(17131A0526)

B Mano Rachna(17131A0527)

B Sruthi(17131A0529)

ABSTRACT

The number of questions asked on question and answer (Q&A) forums like Stack Overflow, Quora, and Twitter, is increasing rapidly. Millions of users visit these sites each month and post their questions. It is no surprise that many of these questions are duplicates. Users may have to wait for a long time to get answers to their questions even though related questions have already been answered. So, it is important to have an automatic way of identifying duplicate threads. Quora, on the other hand, uses a Xgboost model to identify duplicate questions. In this research, we have built a ML model using word2vec from Gensim, trained on Google's 3 million word news dataset. For dimensionality reduction in preprocessing we are choosing TSNE rather than PCA. For detection of word similarity TF-IDF,WORD2VEC would be used. For test and Train data classification Logistic regression, Support Vector Machine and Xgboost would be used.

INDEX

1. INTRODUCTION

1.1 Introduction to the Project

Quora is a very popular website among internet users and a substantial amount of people from around the world visit Quora. Quora is a question answering website where users ask questions and other users respond. The best answers are up-voted and these answers are a valuable learning resource for many topics. Duplicate questions on this site are not uncommon, particularly as the number of questions asked grows. This poses an issue because, if treated independently, duplicate questions may prevent a user from seeing a high quality response that already exists and responders are unlikely to answer the same question twice. Identifying duplicate questions addresses these issues. It reduces the answering burden for responders and makes it possible to direct users to the best responses, improving the overall user experience.

1.2 Existing system

Quora serves as a platform for individuals to interact with other individuals and ask and answer questions. Since the user base is quite high there are instances when there are multiple questions which are based on the same topic and are redundant in nature. These redundant questions decreases the efficiency and creates data that is repeating in the data servers. Since these questions have similar answers and users have to write similar content for each of these question which is waste of time. User of the site have to view answers to the same redundant type of questions which results in wastage of time and reduces the efficiency.

1.3 Proposed System

1.3.1 Routing up the Similarity

User of the site have to view answers to the same redundant type of questions which results in wastage of time and reduces the efficiency. It would be beneficial if the redundant question can be reduced. This will help the users of the site in getting knowledge pertaining to a topic at one place rather than searching to same material over different questions which results in wastage of time and resources. Machine

Learning can be used for tackling this problem. We aim on devising some techniques that would help to judge the similarity between two questions in a more meaningful sense. Also we then aim to decide the similarity between a pair of question using various machine learning algorithms and compare the efficiency of different algorithms in tackling the problem.

This task requires the model to be successful at Natural Language Understanding (NLU). The task of NLU is to build good representations of human language. It's a challenging and important problem, and success at NLU is necessary to be able to succeed at a host of other Natural Language Processing (NLP) tasks like translation, summarization, and reading comprehension. The problem of determining if two sentences have the same meaning or not requires a model to capture the lexical and syntactic meanings of the sentences presented. Therefore the model must be able to grapple with linguistic phenomena like quantification, tense, modality, and syntactic ambiguity. Due to the difficulty of this task, we think the Quora dataset poses an interesting problem. In this paper, we aimed to present a comprehensive set of machine learning models, and to study their performance on the dataset. Duplicate question detection is a binary classification problem on various length strings. The challenging part of the problem is to represent sentences as numerical inputs such that the learning algorithms can work on it.

1.3.1.1 EDA(Exploratory Data Analysis)

Exploratory data analysis (**EDA**) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily **EDA** is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

1.3.1.2 Tokenization

Tokenization is nothing but partitioning of sentences. In this step, we will tokenize or segment text with the help of partitioning text by spaces and punctuation marks to form container of words.

1.3.1.3 Preprocessing of data

Before tokenization all the sentences are preprocessed i.e removal of html tags, stemming of words, expanding contractions etc takes place in this phase so that the sentences are filtered precisely.

1.3.1.4 Applying TFIDFW2V Vectorization

This gives the result of whether two sentences are similar in terms of words similarity as well as similarity in terms of semantics. After we find TF-IDF scores, we convert each question to a weighted average of word2vec vectors by these scores.

1.3.1.5 Classification

The pre-processed question pairs are now classified using Logistic Regression, Support Vector Machine and XGBoost (Ensemble Machine learning Algorithm).

1.3.1.6 Output

Finally, the output obtained is a machine learning model which finds the similarity between two sentences and result whether any two sentences given in any data are similar or not.

2. LITERATURE SURVEY

2.1 Study of related papers

In the first paper their first approach is to first vectorize all words in their question set based on Google's Word2Vec model. To build a vector for the question from its words' vectors, they simply take the mean of all words vectors. Note that in this method, they did not consider the effect of word sequences in the questions. Also, it is very likely that two questions with different words appearances could happen to have very similar vector representations. Further, to make up this loss of information, they first multiply each word vector by its TF-IDF weight, then they still take the mean of each word vectors to construct the vector representations for the question. They keep the question vectors in both methods to achieve a better result. Based on these two kinds of question vectors, after normalization, they computed the similarity of two questions by performing an inner product on two vectors. The two similarity scores along with the common words percentage (number of common words in two questions divided by total length of questions), length difference percentage (difference in length of two questions divided by total length of questions), forms our 4 features for training the classifier.

Further, they tried 5 different classification methods, which are Decision Tree, Random Forest, K Nearest Neighbors, SVM and Adaboost, to train the dataset they acquired from the first approach. Since the KNN method gives us the least Mean Square Errors and Training Error, they finally choose to use the KNN model for prediction on the test set. As for the second approach, they simply ran the built-in evaluation function to see the accuracy, no classification method performed since they regarded that as just an experiment. They finally build a search engine based on their labeled test dataset, thus, it can show the results of not only the related question pairs, but also whether the pairs are duplicate. [1]. In the second paper, they performed SVM classification on the dataset. The dataset they have considered is hosted by Kaggle. It consists of about 404351 training examples having the following format:

• id - the id question pair
• qid1, qid2 - unique ids of each question
• question1, question2 - the full text of each question
• is duplicate - the target variable, set to 1 if question1 and question2 have essentially the same meaning, and 0 otherwise.

The evaluation metric used is the score, which is nothing but the mean accuracy on the given testset and labels. The predictions made by the trained model on the test split are compared with the corresponding labels and the mean accuracy is computed.

The system proposed in the paper got an accuracy score of 85% when trained on a subset of 35000 question pairs sampled from the training data. It achieved an accuracy higher than the baseline.[2]

In this paper, they found the similarity checking using TF-IDF, word2vec model and later on to develop a classification, they have opted many classification models and performed them.

K nearest Neighbors with k =3 and Decision trees give considerably good scores of 0.8021 and 0.8047 respectively while ensemble methods outperform all other models with scores of Random Forests, Gradient Boosting Classifier and Decision trees being 0.8097, 0.83 and 0.8626 respectively

Thus they said that Decision Trees performed best for this dataset and they said they can apply deep learning concepts to achieve greater scores.

2.2 Technology overview

2.2.1 Quora

Quora is an American question-and-answer website where questions are asked, answered, and edited by Internet users, either factually or in the form of opinions. Users can collaborate by editing questions and suggesting edits to answers that have been submitted by other users. Quora is an information exchange site that discourages low-quality answers and requires users to use their real name to sign up. There are

many professionals on Quora, and as such, you can expect to get great answers and have intelligent discussion on this site.

Quora is a continually growing user generated collection of questions and answers. All the questions and answers are created, edited, and organized by the people who use it. While many people use Quora as a resource for research, information, and general interest, some use Quora to build their social network.

2.2.2 Introduction to Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.It is used for:

- Web development(server-side)
- Software development
- Mathematics
- System scripting

Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development.

2.2.3. Python Packages

2.2.3.1 Pandas

Pandas is an open-source Python package that provides high-performance, easy-to-use data structures and data analysis tools for the labelled data in Python programming language. Pandas stand for Python Data AnalysisLibrary. Pandas is a perfect tool for data wrangling or mangling. It is designed for quick and easy data manipulation; reading, aggregation, and visualization. Pandas take data in a CSV or TSV file or a SQL database and create a Python object with rows and columns called a data frame. The data frame is very similar to a table in statistical software, say Excel or SPSS.

- Indexing, manipulating, renaming, sorting, merging data frame

- Update, Add, Delete columns from a data frame
- Impute missing files, handle missing data or NANs
- Plot data with histogram or box plot.

2.2.3.2 NumPy

One of the most fundamental packages in Python, NumPy is a general-purpose array-processing package. It provides high-performance multidimensional array objects and tools to work with the arrays. NumPy is an efficient container of generic multi-dimensional data. NumPy's main object is the homogeneous multidimensional array. It is a table of elements or numbers of the same data type, indexed by a tuple of positive integers. In NumPy, dimensions are called axes and the number of axes is called *rank*. NumPy's array class is called ndarray aka array. NumPy is used to process arrays that store values of

the same data type. NumPy facilitates math operations on arrays and their vectorization. This significantly enhances performance and speeds up the execution time correspondingly.

- Basic array operations: add, multiply, slice, flatten, reshape, indexarrays
- Advanced array operations: stack arrays, split into sections, broadcast arrays
- Work with Date Time or Linear Algebra
- Basic Slicing and Advanced Indexing in NumPy Python.

2.2.3.3 Matplotlib

Matplotlib is the plotting library for Python that provides an object-oriented API for embedding plots into applications. It is a close resemblance to MATLAB embedded in Python programming language. Matplotlib also facilitates labels, grids, legends, and some more formatting entities with Matplotlib.

- Line plots, Scatter plots, Area plots, Bar charts and Histograms, Pie charts, Stem plots.

2.2.3.4 Seaborn

It is defined as the data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. Putting it simply, Seaborn is an extension of Matplotlib with

advanced features. Seaborn provides a variety of visualization patterns with fewer complexes and less syntax.

- Determine relationships between multiple variables (correlation).
- Observe categorical variables for aggregate statistics.
- Analyse uni-variant or bi-variant distributions and compare them between different data subsets.
- Plot linear regression models for dependent variables.
- Provide high-level abstractions, multi-plot grids. Seaborn is a great second-hand for R visualization libraries like corrplot and ggplot.

2.2.3.5 NLTK

NLTK (Natural Language Toolkit) mainly works with human language more than computer language to apply natural language processing (NLP). It contains text processing libraries with which you can perform tokenization, parsing, classification, stemming, tagging and semantic reasoning of data. It may sound repetitive of what this library can do but every lib in Python was written to address some efficiency.

2.2.4. Importing Modules

To make use of the functions in a module, you'll need to **import** the module with an **import** statement. An **import** statement is made up of the **import** keyword along with the name of the module. In a **Python** file, this will be declared at the top of the code. To refer to items from a module within your program's namespace, you can use the **from … import** statement. When you import modules this way, you can refer to the functions by name rather than through dot notation.

3. REQUIREMENT SPECIFICATION

3.1 Functional requirements

- Selection of the particular Online Forum to work on.
- Selection of DATASET.
- Extracting Features from the sentences in the dataset.
- Finding Similarity based on words similarity.
- Finding Similarity based on underlying meaning of words.
- Result whether the provided sentences are semantically similar or not.

3.2 Non – Functional requirements

- Reduces difficulty to find necessary information.
- Help the users of the site in getting knowledge pertaining to a topic at one place rather than searching to same material over different questions which results in wastage of time and resources.
- Performance
- Fault Tolerance

3.3 System requirements

- **Software requirements**

  Python , Google Colab, Jupyter notebook

- **Hardware requirements**

  Windows OS
  Processor: Intel i5
  RAM: 4GB and
  above.

4. DETAILED SYSTEM STUDY

4.1 MODULES

4.1.1 Import Dataset through Kaggle

In this module, firstly we downloaded the question pairs from Kaggle which is "quora question pairs" dataset, It consists

- Size of Train.csv - 60MB
- Number of rows in Train.csv = 404,290

4.1.2 Pre-processing

Once we downloaded the dataset, we need to perform data pre-processing i.e, the data is to be in a form that is feasible to accomplish our task. Each project requires data pre-processing in a unique manner.

But the basic pre-processing steps are :

- Filling Null Values
- Performing Dimensionality Reduction

Besides these, the advance pre-processing techniques involved in our project are:

- Appending extra meaning attributes to our dataset
- Tokenization
- Basic Feature Extraction

Under EDA we perform certain analysis on features present in dataset and Basic Feature Extraction:

- freq_qid1 = Frequency of qid1's
- freq_qid2 = Frequency of qid2's
- q1len = Length of q1
- q2len = Length of q2
- q1_n_words = Number of words in Question 1
- q2_n_words = Number of words in Question 2
- word_Common = (Number of common unique words in Question 1 and Question 2)

- word_Total =(Total num of words in Question 1 + Total num of words in Question 2)
- word_share = (word_common)/(word_Total)
- freq_q1+freq_q2 = sum total of frequency of qid1 and qid2
- freq_q1-freq_q2 = absolute difference of frequency of qid1 and qid2

4.1.3 Applying Word2Vec Algorithm

Word embedding is one of the most popular representation of document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc.

Word2Vec is a method to construct such an embedding. It can be obtained using two methods (both involving Neural Networks): Skip Gram and Common Bag Of Words (CBOW)

4.1.4 Classification and View Results

4.1.4.1.Classification of the data

The main working principle of Detection of Near Duplicate Threads is to find the similarity between two sentences and classify whether they are similar or not.

- Questions or Sentences, if similar are said to be one class.
- Questions or Sentences, if dissimilar are said to be one class.

So to be able to achieve this task by machine itself, we need to develop a Machine learning model that classifies whether two sentences given as input or similar or not. Machine learning Models we opted to perform classification are three: Logistic Regression, SVM (Support Vector Machine), XgBoost (Xtreme Gradient Boosting).

4142. Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm. Logistic regression algorithm uses a linear equation with independent predictors to predict a value. The

predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be class variable, i.e 0-no, 1- yes.

4.1.4.3. SVM(Support Vector Machine)

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems.

Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line). A hyperplane in an n-dimensional Euclidean space is a flat, n-1 dimensional subset of that space that divides the space into two disconnected parts. For example let's assume a line to be our one dimensional Euclidean space(i.e. let's say our datasets lie on a line).

4.1.4.4. XGBoost:

Xgboost is an ensemble technique for machine learning. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. Xgboost uses gradient boosting and it converts weak learners into strong learners.

Of the 3 classification models we applied in this module, we obtained more accuracy for Xgboost when compared to other two.

5. SAMPLE CODE

XGBOOST CLASSIFICATION ALGORITHM

```
import xgboost as xgb
 n_estimators = [50,100,150,200,300,400,500]
test_scores = []
train_scores = []
for i in n_estimators:
    clf = xgb.XGBClassifier(learning_rate=0.1,n_estimators=i,n_jobs=-1)
    clf.fit(X_train,y_train)
    y_pred = clf.predict_proba(X_train)
    log_loss_train = log_loss(y_train, y_pred, eps=1e-15)
    train_scores.append(log_loss_train)
    y_pred = clf.predict_proba(X_test)
    log_loss_test = log_loss(y_test, y_pred, eps=1e-15)
    test_scores.append(log_loss_test)


clf=xgb.XGBClassifier(learning_rate=0.1,n_estimators=500,n_jobs=-1)
clf.fit(X_train,y_train)
y_pred=clf.predict_proba(X_test)
print("The test log loss is:",log_loss(y_test, y_pred, eps=1e-15))
predicted_y =np.argmax(y_pred,axis=1)
plot_confusion_matrix(y_test, predicted_y)
```

In all the machine learning models used XGBOOST algorithm gives less log loss i.e 0.36 So with this model better accuracy is obtained compared to other two models SVM , Logistic Regression.Here the confusion matrix is used for performance metric.

6. TESTING

Testing is the process of detecting errors. Testing performs a very critical role for Quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

6.1 Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

- Testing is a process if executing a program with the intent of finding an error.
- A successful test is one that uncovers an as yet undiscovered error.

6.2 Levels of Testing

In order to uncover the errors, present in different phases we have the concept of levels of testing. The basic levels if testing is as shown below…

6.2.1 System Testing

The philosophy behind testing is to find errors. Test cases are devised with this in mind. A strategy employed for system testing is coding testing.

6.2.2 Code Testing

This strategy examines the logic of the program. To follow this method, we developed some test data that resulted in executing every instruction in the program and module i.e., every path is tested.

6.2.3 White Box Testing

This is unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors.

This application is tested step wise on every piece of code, taking care that every statement in the code is executed.

6.2.4 Black Box Testing

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a black box and ensures that the application is executed correctly.

Test Results: All the test cases mentioned above passed successfully.

7. OUTPUTS SCREENS

Output Screen 1: View of Dataset with Attributes in it

```
df.head()
```

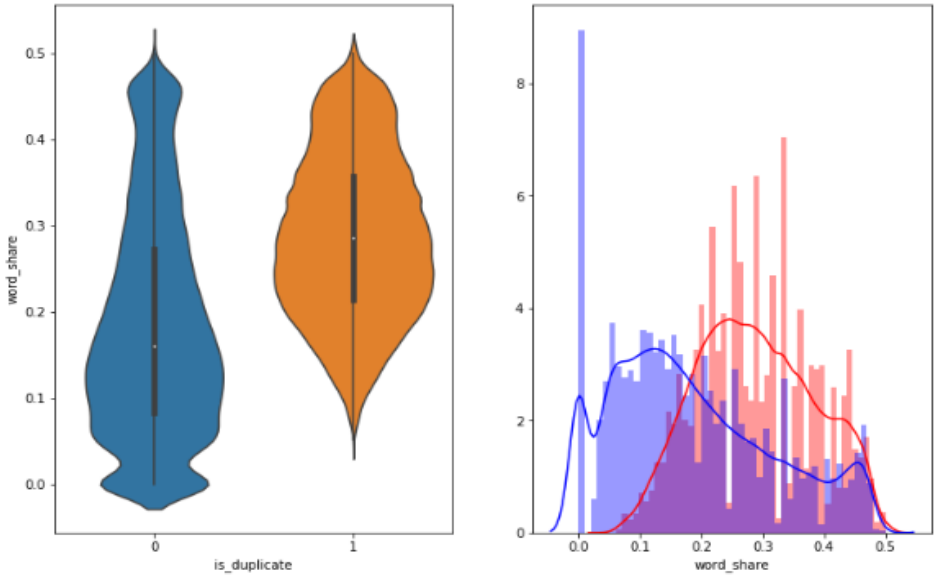
	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

Output Screen 2: Plot showing the word_share among two corresponding questions in the dataset

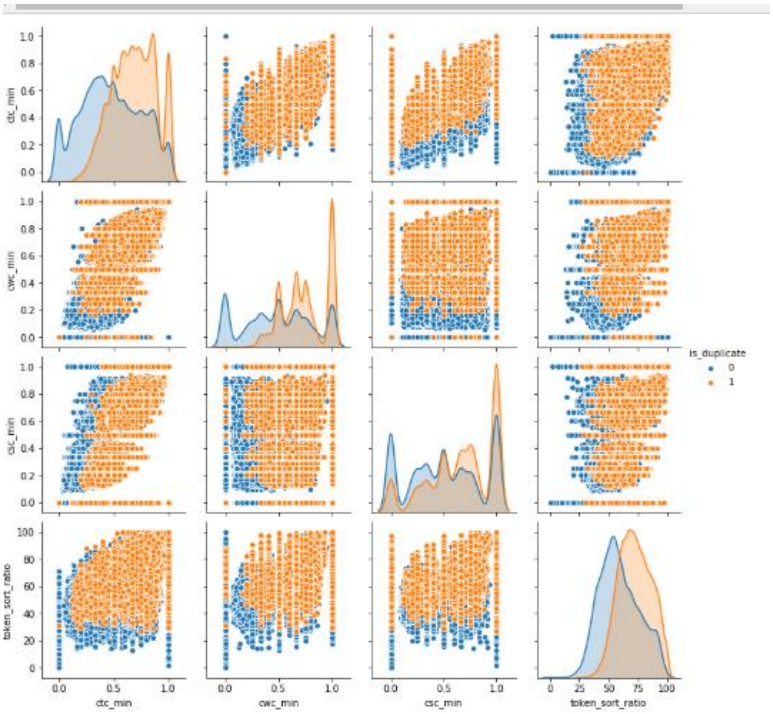
```
In [17]: plt.figure(figsize=(12, 8))

         plt.subplot(1,2,1)
         sns.violinplot(x = 'is_duplicate', y = 'word_share', data = df[0:])

         plt.subplot(1,2,2)
         sns.distplot(df[df['is_duplicate'] == 1.0]['word_share'][0:] , label = "1", color = 'red')
         sns.distplot(df[df['is_duplicate'] == 0.0]['word_share'][0:] , label = "0" , color = 'blue' )
         plt.show()
```

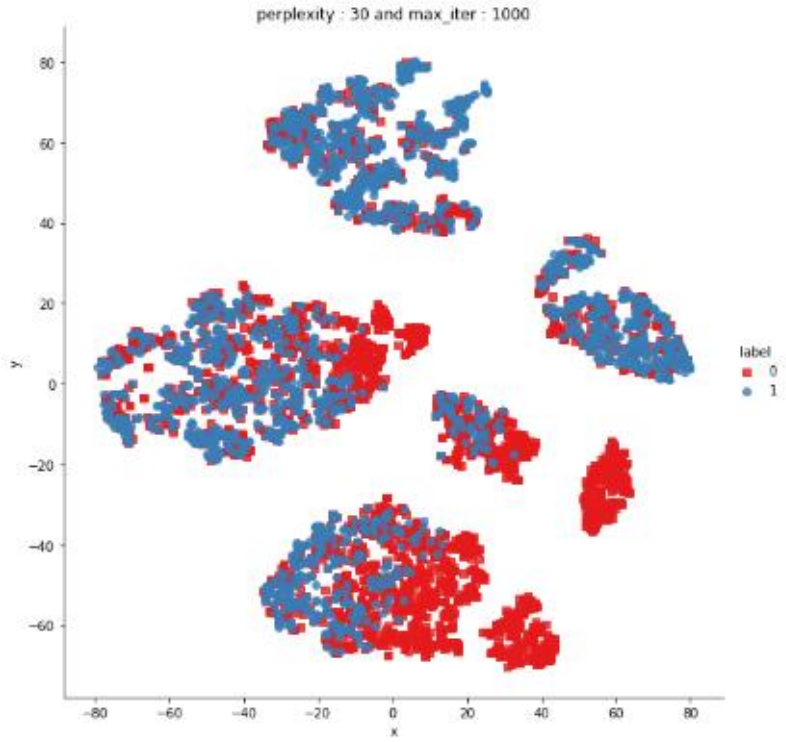


Output Screen 3: Pair plot of features ['ctc_min', 'cwc_min', 'csc_min', 'token_sort_ratio']



Output Screen 4: TSNE for Dimentionality reduction for 15 Features

```
sns.lmplot(data=df, x='x', y='y', hue='label', fit_reg=False, size=8,palette="Set1",markers=['s','o'])
plt.title("perplexity : {} and max_iter : {}".format(30, 1000))
plt.show()
```



Output Screen 5: Storing final features and their targets with respective splitting

id	qid1	qid2	question1	question2	is_duplicate	cwc_min	cwc_max	csc_min	...	ctc_max	last_word_eq	first_word_eq
237030	33086	348102	how can i stop playing video games	should i stop playing video games with my child	0	0.999975	0.799984	0.333322	...	0.555549	0.0	0.0
247341	73272	8624	who is better donald trump or	why is hillary clinton a better	1	0.999980	0.833319	0.333322	...	0.599994	0.0	0.0

abs_len_diff	mean_len	token_set_ratio	token_sort_ratio	fuzz_ratio	fuzz_partial_ratio	longest_substr_ratio
2.0	8.0	87	69	72	86	0.777778
2.0	9.0	92	83	50	50	0.361702

Output Screen 6: Showing about train and test data

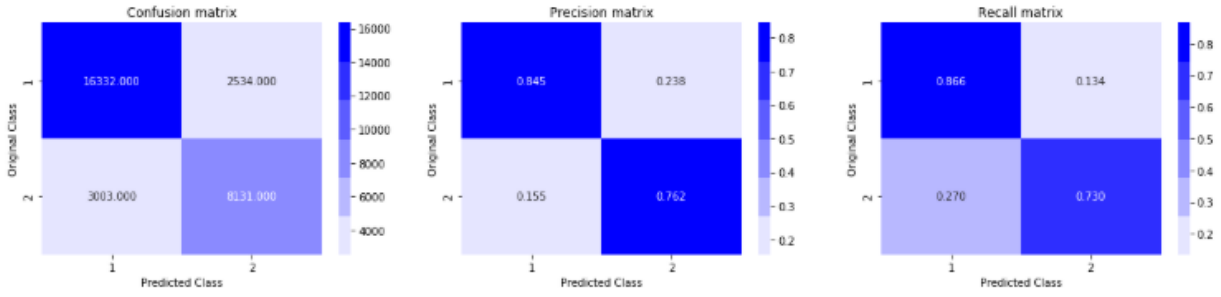
```
#splitting data into train and test
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,random_state=3,test_size=0.3)
```

```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(70000, 27)
(70000,)
(30000, 27)
(30000,)
```

Output Screen 7: Confusion, precision, recall matrices for XGBoost Model

```
The test log loss is: 0.3651387075353528
```



8. CONCLUSION

In this paper we have made a comparison among logistic regression, SVM and Xgboost classification machine learning models regarding routing up the similarity between any two sentences given.

In this paper, we created a classification model using logistic regression, SVM and Xgboost(ensemble machine learning model). The results we obtained in terms of log loss are : For Logistic regression model , it is 0.44. For Lineae SVM model,it is 0.47. For XGBoost model, it is 0.36. XGBoost outperformed the results of Logistic Regression and Linear SVM.

XGBoost model is best classifiying whether the given two questions are similar or not.

FUTURE SCOPE

The future work can be extended to find similarity from sentences to paragraphs.In this project we have considered only two sentences as an example for proving what is actually happening behind the popular search engines like Google etc. In the similar way we can extend this project on applying the task of finding the similarities and detecting plagiarism for huge texts like paragraphs also and the future scope for this project is that the similar tasks and methods are applied in many other search engines which are used daily by many users.

REFERENCES

1. “Proceedings of the 2018 Conference on Supervised Clustering of Questions into Intents for Dialog System Applications”, pages 2310–2321 Brussels, Belgium, October 31 - November 4, 2018. c 2018.

2. Torsten Zesch et al. UKP: “Identifying Quora question pairs having the same intent” IEEE Transactions on Software Engineering, Vol. 37, No. 1, January/February 2018, pp. 132-139.

3. Jiang Zhao, Tian Tian Zhu, and Man Lan. Ecnu: One stone two birds: “Ensemble of heterogenous measures for semantic relatedness and textual entailment”. In SemEval, 2014.