PROJECT REPORT ON

## Solve Quora Question Pair Similarity Problem By using Machine Learning Algorithms

#### SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS OF THE AWARDS OF THE

**PG-DIPLOMA IN ADVANCED SECURE SOFTWARE DEVELOPMENT**

**OFFERED BY**

**C-DAC HYDERABAD**

#### BY

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |



**ADVANCED COMPUTING TRAINING SCHOOL C-DAC**

**HYDERABAD-500005 SEP 2022**



#### CERTIFICATE

This is to certify that this is a Bonafide record of a project entitled **“Solve Quora Question Pair Similarity Problem By using Machine Learning Algorithms''** Pushpa Lokre 220950329011, Sachin Shah 220950329019, Siddhesh Sethi 220950329022, Yashwi Das 220950329026, has completed project work as part of **Diploma in Secure Software Development (Sep 2022 Batch)**, a PG course offered by C-DAC Hyderabad. They have completed project work under the supervision of Mr. Yuvraj Takey. Their Performance found to be Good.

Name of Project guide

(Mr. Yuvraj Takey)

**DATE** :

**PLACE** : C-DAC, HARDWARE PARK, HYDERABAD.

## ACKNOWLEDGEMENT

#### Solve Quora question pair similarity Problem using MachineLearning Algorithm Aim of this project is to Identify which questions asked on Quora are duplicates of questions that have already been asked. This could be useful to instantly provide answers to questions that have already been answered. We are tasked with predicting whether a pair of questions are duplicates or not.

However, we could not have made it without the support and guidance from the Following, Firstly I want to take this opportunity to have special thanks to our guide **Mr. Yuvraj Takey** who helped us throughout this project by providing valuable guidance,advice as well as acquiring all components needed for this project to become a success.

|  |  |
| --- | --- |
|  | (PG-DASSD SEP 2022) |
| **PUSHPA LOKRE** | **220950329011** |
| **SACHIN SHAH** | **220950329019** |
| **SIDDHESH SETHI** | **220950329022** |
| **YASHVI DAS** | **220950329026** |

TABLE OF CONTENTS

Solve Quora Question Pair Similarity Problem By using Machine Learning Algorithms 1

ACKNOWLEDGEMENT 3

Abstract : Error! Bookmark not defined.

SOLVE QUORA QUESTION PAIR SIMILARITY Error! Bookmark not defined.

Chapter-1 INTRODUCTION 6

1.1 Project Requirements: 7

1.2 Problem Statement: 7

1.3 Software & hardware Requirements : 7

1.4 Coding and Testing : 8

1.5 Basic Terminology 8

1.5.2 DataFrame: 8

1.5.3 Supervised Learning: 8

2.1 Importing Libraries 10

Chapter-3 IMPLEMENTATION 14

CHAPTER-4 MACHINE LEARNING MODELS 23

Chapter - 5 RESEARCH 27

Chapter - 6 CONCLUSION 35

References 37

**SOLVE QUORA QUESTION PAIR SIMILARITY**

**ABSTRACT**

Quora is a famous platform for Questions and their Answers (Q&A), similar to answers.com, stackoverflow.com and askbot.com. Among all of them Quora is most frequently used among various categories of users. It has a huge database of (Q&A), a big problem with this database is "similar question pair". For Example, someone asked "How to score good marks in Maths?" and another question could be "What can I do to be good in Maths?". As both the questions are having similar context, we can have the same answer for the second question too. This gives Quora a big advantage as, It can quickly deliver answer to the similar question and the answerer does not have to repeat himself. This proves out to be a time and effort efficient solution. Our target will be to find duplicate "question pairs" from Quora and feed them with same answers. To achieve this, we will be implementing robust Machine Learning models.Our goal of this project is to find the duplicate question pairs on Quora and ignore them by using the Machine Learning Algorithms especially by Classification Models(Logistic Regression, Random Forest, SVM etc..), first we need to collect the data from Kaggle for detection of duplicate questions, and we will do preprocessing and apply the classification models on that large datasets and we will evaluate the model accuracy and at last we will propose the best model based on that accuracy.

### Chapter-1 INTRODUCTION

Quora is a platform where people can ask their questions and get answers from different people, sometimes the questions asked may be asked in different format but the intent was the same . let's discuss it with an example

We have a question like *“How can I be a good geologist?”* and there are some answers to that question. Later someone asks another question like *“What should I do to be a great geologist?”.*

We can see that both the questions are asking the same thing. Even though the wordings in the question are different, the intention of both questions are the same. So the answers will be the same for both questions. That means we can just show the answers to the first question so that the person who is asking the question will get the answers immediately and people who have already answered the first question don’t have to repeat themselves.

If duplicates are allowed, it would corrupt the quality of answers thereby negatively affecting the experience of the person asking the questions, person answering the questions and the person searching the web for an answer (imagine searching Google for a question and finding 3 results from Quora instead of 1). This problem however is not unique to Quora and many organizations have similar issues.

Ideally, what would happen is that once a question is asked, Quora would use some “technique” to find a subset of its existing question database such that this subset contains questions which are “similar” to or about the same topic as the new question being asked. Once this subset has been identified, Quora would employ a machine learning technique to then determine if a duplicate question exists in this selected subset. If yes, it would notify the questioner and point them to it, else it creates the question.

Our job in this project is to develop the machine learning part. We are not concerned with how Quora generates the subset, assume that it is there and works properly.

## Project Requirements:

**Dataset:** Kaggle Dataset

**Software :** Jupiter notebook.

**Problem Statement:**

Identify which questions asked on Quora are duplicates of questions that have already been asked. This could be useful to instantly provide answers to questions that have already been answered. We are tasked with predicting whether a pair of questions are duplicates or not.

## Software & hardware Requirements :

###### Hardware Specification:

* + - * Processor – i5 or more

###### Hard Disk – 50 GB

* + - * Memory – Min 8GB RAM

###### Software Specification:

* + - * Operating System : Windows 7 or more

###### Language: Python

* + - * IDE : Jupyter Notebook

# Coding and Testing :

In this we develops a code using following software languages:

* PYTHON

### Basic Terminology

* + 1. **Dataset:**

A **data set** (or **dataset**) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every [column](https://en.wikipedia.org/wiki/Column_(database)) of a table represents a particular [variable](https://en.wikipedia.org/wiki/Variable_(computer_science)), and each [row](https://en.wikipedia.org/wiki/Row_(database)) corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Data sets can also consist of a collection of documents or files.

### DataFrame:

A Data frame is a two-dimensional data structure.

All the ndarrays must be of the same length. If index is passed, then the length of the index should equal to the length of the arrays.

A pandas DataFrame can be created using various inputs like −

1. Lists
2. dictionary
3. Series
4. Numpy ndarrays

All the ndarrays must be of the same length. If index is passed, then the length of the index should equal to the length of the arrays.

### Supervised Learning:

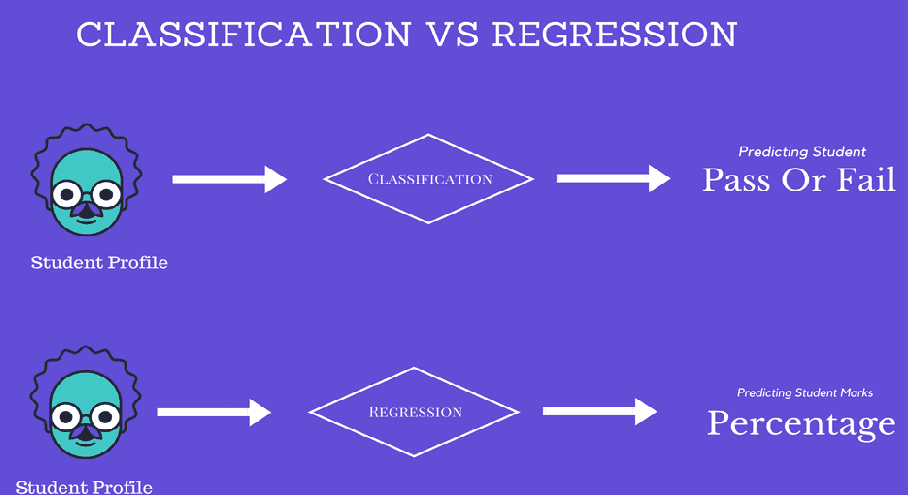
Supervised Learning is defined as the category of data analysis where the target outcome is known or labeled.

Classification and Regression both belong to Supervised Learning, but the former is applied where the outcome is finite while the latter is for infinite possible values of outcome

A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. Outcomes are labels that can be applied to a dataset.

There are a number of classification models. Classification models include

1. logistic regression,
2. decision tree
3. random forest
4. gradient-boosted tree
5. multilayer perceptron
6. one-vs-rest
7. Naive Bayes.



**Chapter-2**

**LIBRARIES**

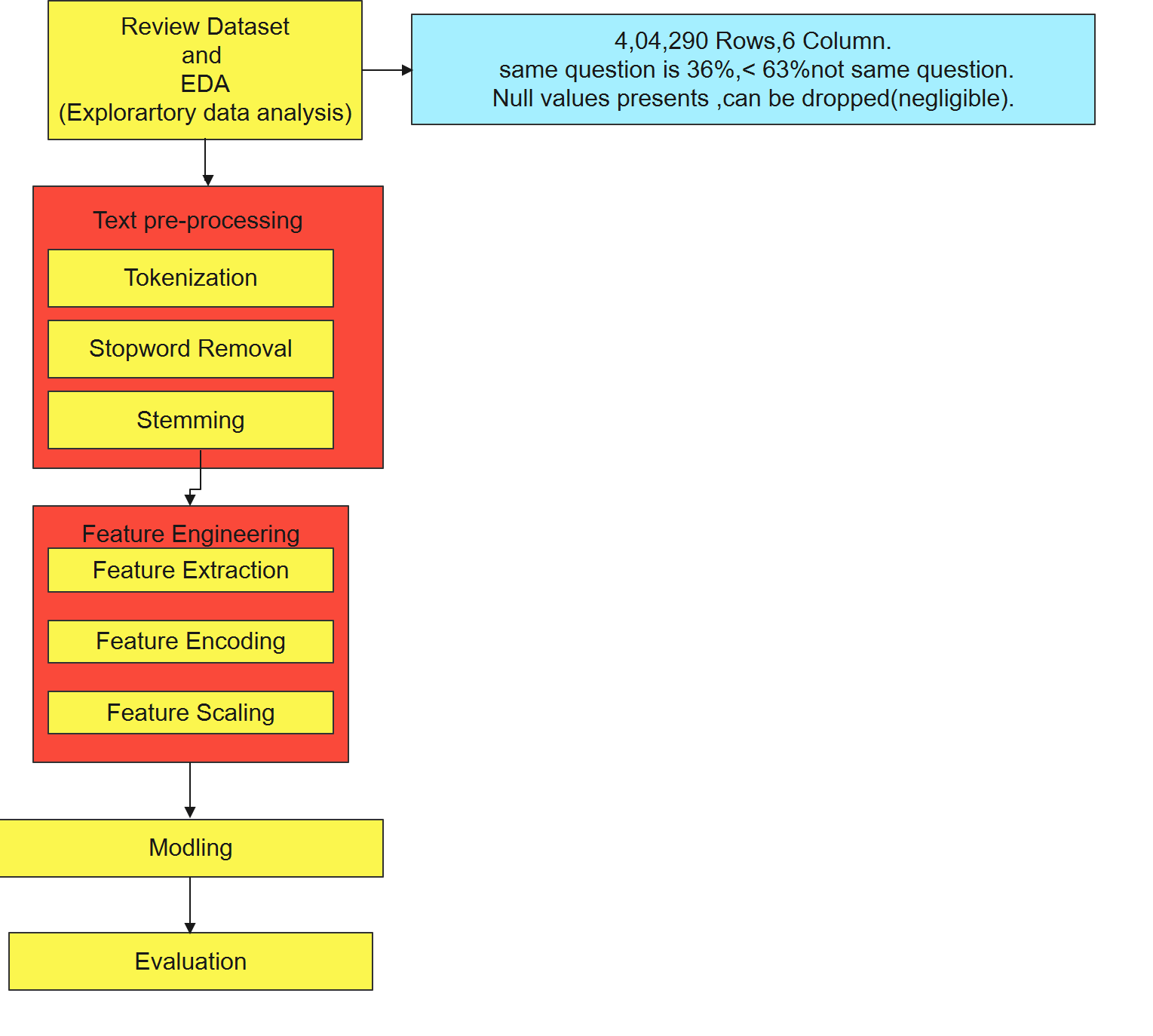
# 2.1 Importing Libraries

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Description** | **Why do we use it?** |
| Numpy  **Command:**  import numpy as np | 1. Numpy is the core library for scientific computing in python. 2. This Library works with multidimensional arrays very efficiently. 3. This is also useful in Linear Algebra, Fourier Transform, and Random Number Capabilities. | 1. Numpy is very efficient when working with   multidimensional arrays.   1. It occupies very less memory, so we can save space and we can use it easily 2. it is very Fast as compared to Lists. |
| Matplotlib  **Command:**  import matplotlib.pyplot as plt | 1. Matplotlib is a cross-platform, data visualization and graphical plotting library for Python. 2. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits. 3. It is a robust, free and easy   library for data visualization. | 1. This Library is Used to plot the 2D Graphs to visualization to understand easily. 2. Matplotlib is effective when we integrate it with other GUI toolkits. |
| Pandas  **Command:**  import pandas as pd | 1. It is a manipulation tool, built on top of Python. 2. It has functions for analyzing, cleaning, exploring, and manipulating data. 3. pandas is a fast, powerful, flexible and easy to use open   source data analysis. | 1. We used it for reading and analyzing the dataset. 2. it has high performance for data analysis 3. It has the ability to handle the missing data in large datasets. |
|  | 3. It is built on the top of Matplotlib Library. |  |
| NLTK  **Command:**  import nltk | 1. The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for application in statistical natural language processing (NLP). 2. It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. | 1. It is used while performing various NLP tasks during the text processing step. |
| RegEx  **Command:**  import re | 1. It has a built-in package in   Python.   1. It can work with Regular Expressions. | 1. It can be used to search   patterns to find a string.   1. It can be used to detect the absence of a text by matching with a particular pattern |
| Fuzzywuzzy  **Command:**  from fuzzywuzzy import fuzz | 1. Fuzzy string matching is the process of finding strings that match a given pattern. | 1. It is used for string matching. |
| Warnings  **Command:**  import warnings | 1. The warning module is actually a subclass of Exception which is a built-in class in Python. 2. Base category for warnings triggered during the process of importing a module (ignored by default). | 1. This module is used to show warning messages. 2. In order to disable all warnings in the current script/notebook just use filterwarnings(‘ignore’) |
| Distance  **Command:**  import distance | 1. This helps for computing similarities between arbitrary sequences 2. All distance computations are implemented in pure Python, and most of them are also   implemented in C. | 1. In clustering, distance measures are used to group similar data points together based on their distance in a feature space. |
| Word2vec  **Command**  from genism.model  import word2vec | 1. library is used for training and working with word embeddings. 2. To use word2vec for training word embeddings, you need a corpus of text data | 1. we are using this for feature encoding 2. word2vec is giving us minimum dimensions |
| TSNE  **Command:**  from  sklearn.manifold import TSNE | 1.T-distributed Stochastic Neighbor Embedding.  2.It is a class from sklearn.manifold | 1.t-SNE is a dimensionality reduction technique used to represent high dimensional dataset to low dimensional. so that we can visualize it. |
| Confusion Matrix  **Command:**  from  sklearn.metrics import  confusion\_matrix | 1. It is a class from sklearn.metrics. 2. Confusion matrix metrics are performance measures which helps us to find the accuracy of our classifier. | 1.Confusion matrix helps us to identify the correct predictions of a model for different individual classes as well as errors. |
| Accuracy Score **Command:** from  sklearn.metrics import  accuracy\_score | 1. It is a class from sklearn.metrics. 2. The sklearn.metrics has an accuracy\_score() function that used to calculate the accuracy. | 1.The accuracy of the model is calculated as a ratio between the number of correct predictions to the total number of predictions. |
| OS  **Command:**  import os | 1.The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory etc., | 1.The os and os.path modules include many functions to interact with the file system. |
| Spacy  **Command:**  import spacy | 1.spaCy is a free, open-source library for NLP in Python. | 1 It.is designed to build information extraction or natural language understanding systems. |
| Snowball Stemmer  **Command:**  from nltk.stem import SnowballStemmer | 1.It is a stemming algorithm which is also known as Porter2 stemming algorithm. | 1.It is the process of reducing the word to its word stem. |
| TIIDF  **Command:**  from sklearn.feature\_extraction.text import TfidfVectorizer | 1.TF-IDF (Term Frequency-Inverse Document Frequency) is a commonly used technique in machine learning and natural language processing for text classification and information retrieval. | 1. is a commonly used technique in machine learning and natural language processing for text classification and information retrieval. |

### Chapter-3 IMPLEMENTATION

**NLP CYCLE (NATURAL LANGUAGE PROCESSING)**

The NLP cycle refers to the general steps involved in developing a natural language processing (NLP) application. The steps typically involve:



##### Steps involved in implementing the problem were:

* + 1. Problem background and data source.
    2. Exploring the data set and its format.
    3. Data cleaning and preprocessing.
    4. Featurization.
    5. Visualizing features and removing redundant ones.
    6. Feature Encoding
    7. Vectorizing textual features.
    8. Applying machine learning for classification.

##### Problem background and data source:

If duplicates are allowed, it would corrupt the quality of answers asked by questioner thereby it may negatively affect the experience of the person asking the questions, person answering the questions and the person searching the web for an answer.Ideally, what would happen is that once a question is asked, Quora would use some “Technique” to find a subset of existing question in the database such that this subset contains questions which are “similar” to or about the same topic as the new question that is being asked. Once a subset has been identified, Quora would employ a machine learning technique to determine if a duplicate question exists in this selected subset or not. If yes, it would notify the questioner and point them to it, else it creates the question.

##### Exploring the data set and its format:

The train set consists of 1,99,999 question pairs (rows) and has 6 variables (columns). The columns are the row ID, question 1, question 2, question 1 ID, question 2 ID and the class label which is 0 for non-duplicate pairs and 1 for duplicate pairs. The question IDs (qids) uniquely identify each question.

The test set consists of over a million question pairs. The test set consists of 3 columns which are the row ID, question 1 and question 2.

The train set is what we train our models on and the test set is how we evaluate the performance of our models.

##### Data cleaning and preprocessing:

In this DataCleaning Stage We try to remove bad observations, if any, from the data. These could be duplicate rows, it cannot be negative. We would remove that row. This is a domain specific problem.

In the data preprocessing stage, we clean up each row’s data. This is a major step in the data science and machine learning pipeline. The preprocessing steps performed were:

* + - * Convert text to lower case.
      * Remove punctuation.
      * Replace some numerical values with strings (Eg: 1,000,000 with 1m and 1000 with 1k).
      * Replace some characters with their string equivalents (Eg: $, % @ etc.).
      * Stemming the word
      * Stop word removal
    1. Featurization

The following features were extracted:

* + - * Token features

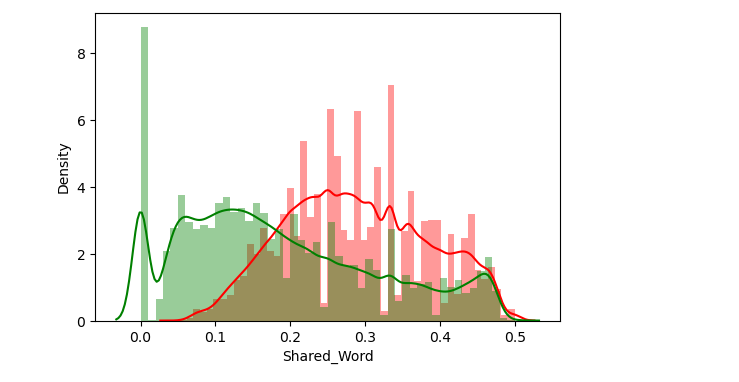
1. **q1\_len**: Number of characters in question 1
2. **q2\_len**: Number of characters in question 2
3. **q1\_words**: Number of words in question 1
4. **q2\_words**: Number of words in question 2
5. **words\_total**: Sum of **q1\_words** and **q2\_words**
6. **words\_common**: Number of words which occur in question 1 and two, repeated occurrences are not counted
7. **words\_shared**: Fraction of **words\_common** to **words\_total**
8. **cwc\_min**: This is the ratio of the number of common words to the length of the smaller question
9. **cwc\_max**: This is the ratio of the number of common words to the length of the larger question
10. **csc\_min**: This is the ratio of the number of common stop words to the smaller stop word count among the two questions
11. **csc\_max**: This is the ratio of the number of common stop words to the larger stop word count among the two questions.
12. **ctc\_max**: This is the ratio of the number of common tokens to the larger token count among the two questions
13. **last\_word\_eq**: 1 if the last word in the two questions is same, 0 otherwise
14. **first\_word\_eq**: 1 if the first word in the two questions is same, 0 otherwise
15. Fuzzy features
    * **fuzz\_ratio**: fuzz\_ratio score from fuzzywuzzy
    * **fuzz\_partial\_ratio**: fuzz\_partial\_ratio from fuzzywuzzy
    * **token\_sort\_ratio**: token\_sort\_ratio from fuzzywuzzy
    * **token\_set\_ratio**: token\_set\_ratio from fuzzywuzzy
      + - Length features
16. **mean\_len**: Mean of the length of the two questions (number of words)
17. **abs\_len\_diff**: Absolute difference between the length of the two questions (number of words)
18. **longest\_substr\_ratio**: Ratio of the length of the longest substring among the two questions to the length of the smaller question

##### Visualizing features and removing redundant ones

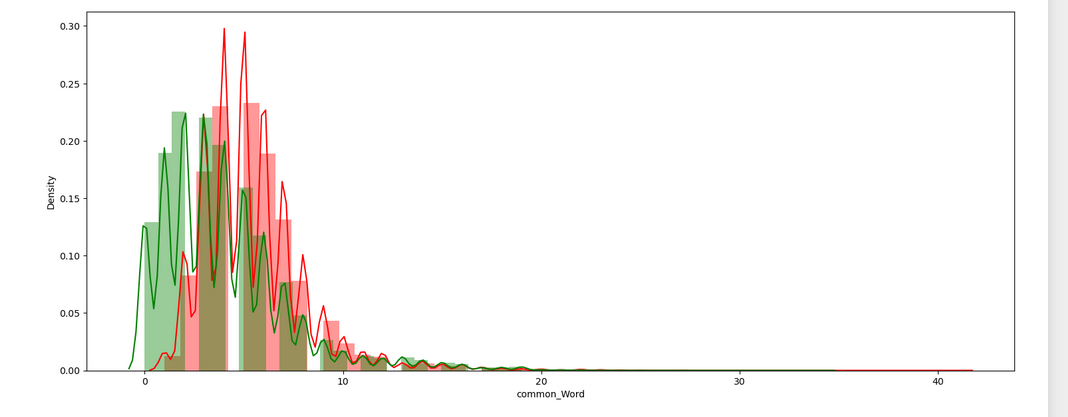
word vector is a numerical representation of textual data. There are many ways to go about transforming (vectorizing) textual data to its numerical representation (vectors)

We have used CountVectorizer While Counter is used for counting all sorts of things, the CountVectorizer is specifically used for counting words. The vectorizer part of CountVectorizer is the process of converting text into numbers.

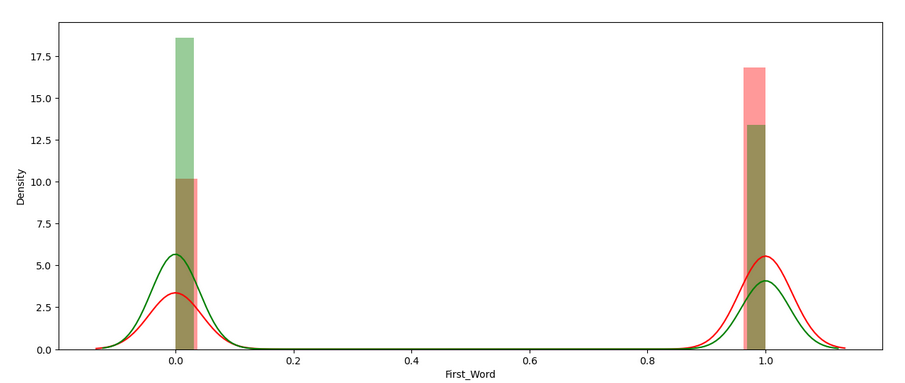
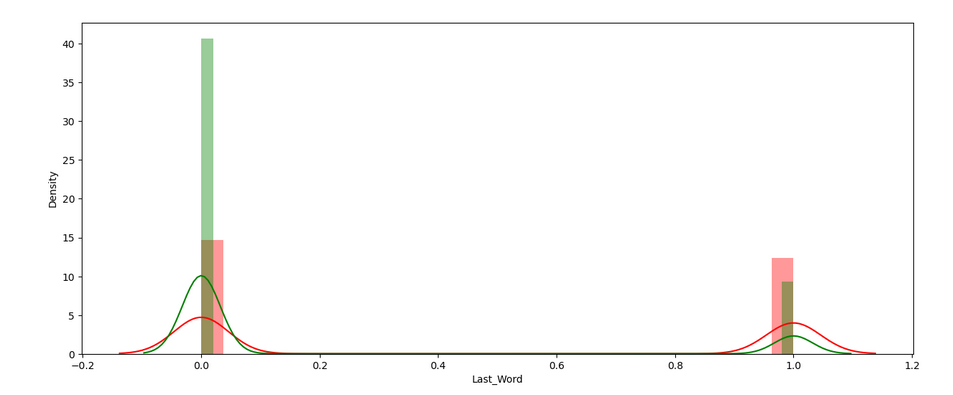
* + - * **Word Share:** We can check from below that it is overlapping a bit, but it is giving some classifiable score for dissimilar questions**.**

****

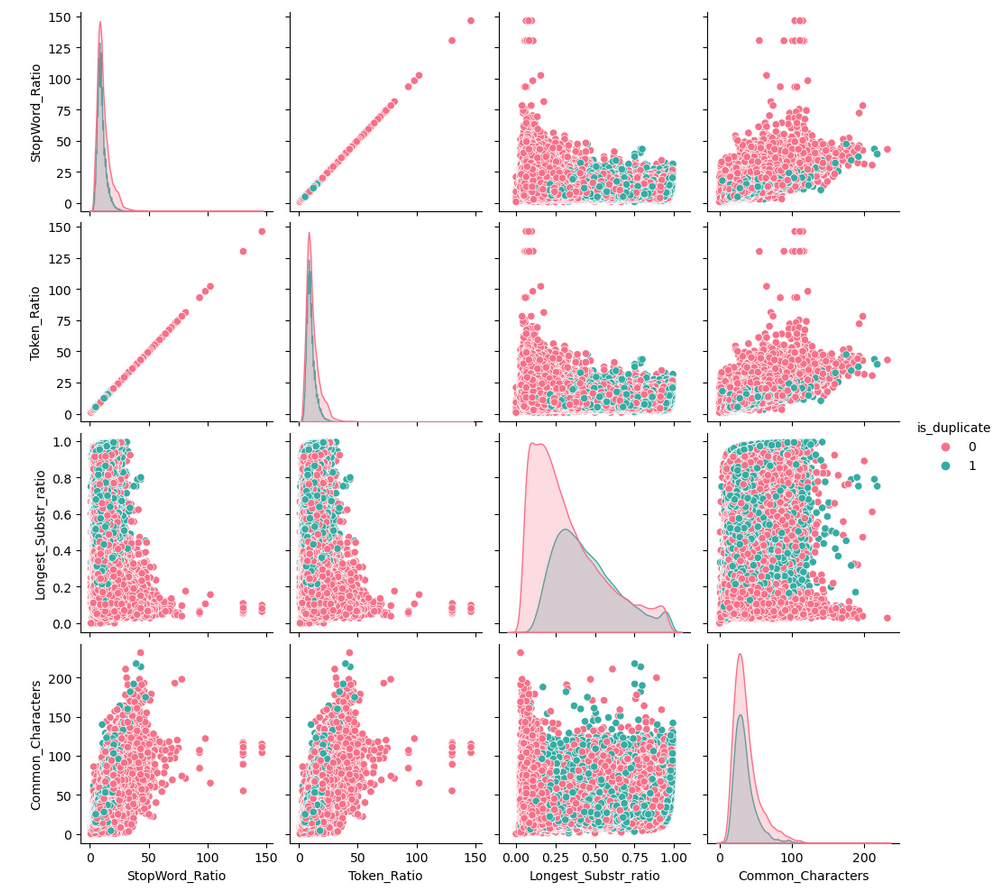
* + - * **Word Common:** it is almost overlapping



* + - * Bivariate analysis of features ‘last\_word\_eq’, ‘first\_word\_eq’. We can observe that we can divide duplicate and non duplicate with some of these features with some patterns.



* + - * Bivariate analysis of features ‘common\_charecters’,’longest\_substr\_ratio’,’token\_ratio’,’stopword\_ratio’. We can observe that we can divide duplicate and non duplicate with some of these features with some patterns.



**3.1.6. Feature Encoding:**

Till now we did text cleaning and we get our clean data after that we have to change our string into integer because our model cannot understand string. TF-IDF (Term Frequency-Inverse Document Frequency) is a commonly used technique in machine learning and natural language processing for text classification and information retrieval.

TF-IDF is used to evaluate how important a word is to a document in a collection or corpus. The importance of a word increases proportionally to the number of times it appears in the document but is offset by the frequency of the word in the corpus. This helps to identify words that are unique and important to a particular document, and down-weight words that are common across many documents.

##### 3.1.7 Applying machine learning for classification:

* + - * A model with a higher bias would not match the data set closely.
      * A low bias model will closely match the training data set.
      * Models with high bias will have low variance.
      * Models with high variance will have a low bias.
      * Characteristics of a high variance model include:
      * Noise in the data set
      * Potential towards overfitting
      * Complex models
      * Trying to put all data points as close as possible

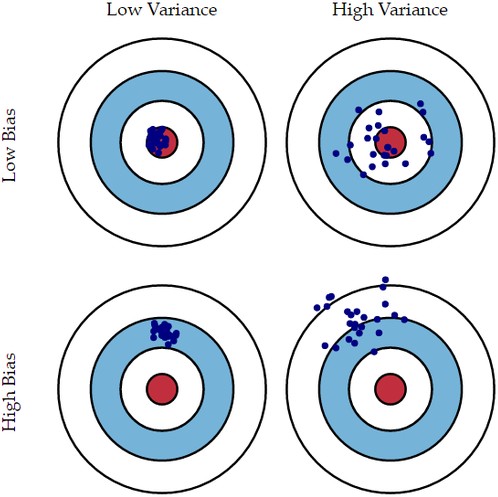
Examples of **low-variance** machine learning algorithms include:

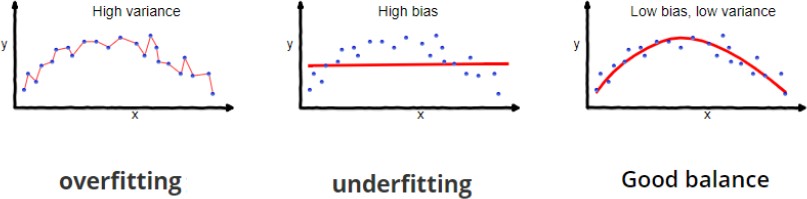
* + - * Linear Regression,
      * Linear Discriminant Analysis and
      * Logistic Regression.

Examples of **high-variance** machine learning algorithms include:

* + - * Decision Trees
      * k-Nearest Neighbors and
      * Support Vector Machines.

we can choose the model by using accuracy score of model, and we have





### CHAPTER-4 MACHINE LEARNING MODELS

##### RandomForest:

Random forest is a versatile machine learning method capable of performing both regression and classification tasks.Like bagging and boosting, random forest works by combining a set of other tree models. Random forest builds a tree from a random sample of the columns in the test data.

Here’s are the steps how a random forest creates the trees:

* + - Take a sample size from the training data.
    - Begin with a single node.
    - Run the following algorithm, from the start node:
      * If the number of observations is less than node size then stop.
      * Select random variables.
      * Find the variable that does the “best” job of splitting the observations.
      * Split the observations into two nodes.
      * Call step `a` on each of these nodes.
    1. Advantages:
       - Random Forest can be used to solve both classification as well as regression problems.
       - Random Forest works well with both categorical and continuous variables.
       - Random Forest can automatically handle missing values.
       - Random Forest can automatically handle missing values.
       - Random Forest is usually robust to outliers and can handle them automatically.
       - Random Forest is comparatively less impacted by noise.
    2. Limitations:
       - **Complexity:** Random Forest creates a lot of trees (unlike only one tree in case of decision tree) and combines their outputs.
       - **Longer Training Period:** Random Forest requires much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decisions on the majority of votes.

##### Decision Tree:

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

* + 1. Advantages:
       - An advantage of the decision tree algorithm is that it does not require any transformation of the features if we are dealing with non-linear data because decision trees do not take multiple weighted combinations into account simultaneously.
       - They are very fast and efficient compared to KNN and other classification algorithms.
       - Easy to understand, interpret, and visualize.
       - Normalization is not required in the Decision Tree.
       - The decision tree is one of the machine learning algorithms where we don’t worry about its feature scaling. Another one is random forests. Those algorithms are scale-invariant.
    2. Limitations:

1. Method of overfitting: If we discuss overfitting, it is one of the most difficult methods for decision tree models. The overfitting problem can be solved by setting constraints on the parameters model and pruning method.
   * + 1. Reusability in decision trees: In a decision tree there are small variations in the data that might output in a complex different tree is generated. This is known as variance in the decision tree, which can be decreased by some methods like bagging and boosting.
       2. It can’t be used in big data: If the size of data is too big, then one single tree may grow a lot of nodes which might result in complexity and lead to overfitting.
       3. There is no guarantee to return the 100% efficient decision tree.

##### XGBOOST:

**XGBoost** is an optimized distributed gradient boosting library designed to be highly ***efficient****,* ***flexible*** and ***portable***. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting that solves many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of example

XGBoost uses decision trees as base learners; combining many weak learners to make a strong learner. As a result it is referred to as an ensemble learning method since it uses the output of many models in the final prediction. XGBoost or Extreme Gradient Boosting! It can be put into various use cases such as ranking, classification, regression and user-defined prediction problems. It can be referred to as an "ALL in One" algorithm. It is an ideal blend of software and hardware optimization techniques to yield prevalent outcomes by using fewer computing resources in the shortest amount of time.

4.3.1Advantages:

* + - * It is Highly Flexible
      * It uses the power of parallel processing
      * It is faster than Gradient Boosting
      * It supports regularization
      * It is designed to handle missing data with its in-build features.
      * The user can run a cross-validation after each iteration.
      * It Works well in small to medium dataset
  1. Advantages of quora website:

**4.4.1 Advantages :**

* + - * You could get much information at one place instead of going through multiple Google searches and links.
      * You could get answers from experts from across the world without even contacting them.
      * You can get and give information anonymously.
      * Quora is knowledge oriented, not person oriented. Persons with 100K upvote and 0 upvote could get banned for the same reason.
      * You could edit questions and answers anytime. You could also edit any question on quora and suggest edits for others answers.
      * You can thank people and get thanks.
      * Quora is one of the best platforms for cultural exchange. Global village? Something like that.
    - Limitations :
      * Quora's got a bunch of great content creators, but no good way to find them.
      * It's becoming too easy for Quora to be about the answers rather than the questions
      * You have to select proper topics else you find yourself in the wrong sector of quora.

### Chapter - 5 Research

1. Lemmatization
2. Stemming
3. NER (Named Entity Recognition)
4. Bag of Word
5. Count Vectorizer
6. TFIDF (term frequency-inverse document frequency)
7. N-grams
8. Word2vec
9. BERT (Bidirectional Encoder Representations from Transformers)
10. **Lemmatization**

Lemmatization is a natural language processing technique that is used to convert words to their base or dictionary form, called a "lemma". The purpose of lemmatization is to reduce words to a common base form, which can simplify text analysis, improve accuracy, and reduce noise in the data.

Lemmatization is different from stemming, which is another text preprocessing technique that also reduces words to their base form, but typically uses a more crude approach by simply removing the suffix of a word to obtain a root stem. In contrast, lemmatization uses a knowledge-based approach that takes into account the part of speech and context of the word in the sentence.

For example, consider the following sentence: "The dogs are barking loudly". With stemming, the word "barking" would be reduced to its root "bark", whereas with lemmatization, the word would be reduced to its base form "bark", but the part of speech would be preserved. Similarly, the word "dogs" would be reduced to "dog" with lemmatization, whereas with stemming, it would be reduced to "dog".

In our project meaning is not matter for questions. If one word converted to its root form then similar word will also convert into its root form so we will get similar word that’s why we are using stemming instead of lemmatization. It is not putting more weight to our questions

1. **Stemming**

Stemming is the process of reducing a word to its base or root form, known as the stem. This is done by removing suffixes and prefixes from the word, so that different inflections of the same root word can be treated as equivalent. For example, the stem of the words "jumped", "jumping", and "jumps" is "jump".

Stemming is a commonly used technique in natural language processing and information retrieval, where it is used to normalize words and reduce the size of the vocabulary. By reducing words to their stems, different variations of the same word can be treated as the same token, which can improve the accuracy of text analysis.

There are several algorithms used for stemming, including Porter stemming, Snowball stemming, and Lancaster stemming. Each algorithm has its own set of rules for removing suffixes and prefixes from words.

For example, the Porter stemming algorithm applies a series of rules to remove common English suffixes such as "-ed", "-ing", and "-s", while the Snowball stemming algorithm is an extension of the Porter algorithm that also handles other languages.

However, stemming can sometimes produce incorrect stems or reduce words to meaningless forms, known as "stemming errors". For example, the Porter stemming algorithm can produce the incorrect stem "jump" for the word "jumper", or reduce the word "argue" to the meaningless stem "argu". As a result, stemming is often used in conjunction with other text processing techniques such as lemmatization or part-of-speech tagging to improve accuracy.

1. **NER (Named Entity Recognition)**

Named Entity Recognition (NER) is a natural language processing technique that is used to identify and extract named entities from text. Named entities refer to specific types of objects, such as people, places, organizations, and products, which are often referred to by proper names or specific terms.

The goal of NER is to automatically identify and classify these named entities in a text, and assign them to pre-defined categories. For example, the sentence "Steve Jobs was the founder of Apple Inc." would be identified as containing two named entities, "Steve Jobs" and "Apple Inc.", and would be classified as a person and an organization, respectively.

NER is commonly used in a variety of applications, such as information extraction, text classification, and information retrieval. It can be used to extract structured information from unstructured text data, and can help to improve the accuracy of natural language processing models by reducing noise in the data.

We are not using NER in our project because if question came

(i). Who is president of India?

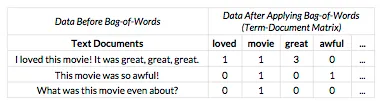
(ii). Who is president of Pakistan?

If we are using NER so it will label India and Pakistan as a country and our question will be country and country. As we can see both questions are different but when we are using NER it’s giving label to India and Pakistan as country so they will become same question although they are totally different. If this types of questions occurs again it may be possibility it will consider it as a same question and it will affect our model for this reason we are not using NER.

**3.Bag of word**

It first constructs a dictionary of the set of all the words in the text. It consists of all unique words in the text. It represents word as a sparse matrix.

For each document(row), find unique words where each word is a different dimension. Each cell consists of the number of times the word occurs in the respective row. will be very large where most of the cells have zero value. This is the reason a sparse matrix will be formed.



1. **Count vectorizer**

Machines cannot understand characters and words. So when dealing with text data we need to represent it in numbers to be understood by the machine. Countvectorizer is a method to convert text to numerical data. To show you how it works let’s take an example:

text = [‘Hello my name is james, this is my python notebook’]

The text is transformed to a sparse matrix as shown below.

We have 8 unique words in the text and hence 8 different columns each representing a unique word in the matrix. The row represents the word count. Since the words ‘is’ and ‘my’ were repeated twice we have the count for those particular words as 2 and 1 for the rest.

Count vectorizer makes it easy for text data to be used directly in machine learning and deep learning models such as text classification.

1. **TFID** **(term frequency-inverse document frequency)**

TF-IDF (Term Frequency-Inverse Document Frequency) is a commonly used technique in machine learning and natural language processing for text classification and information retrieval.

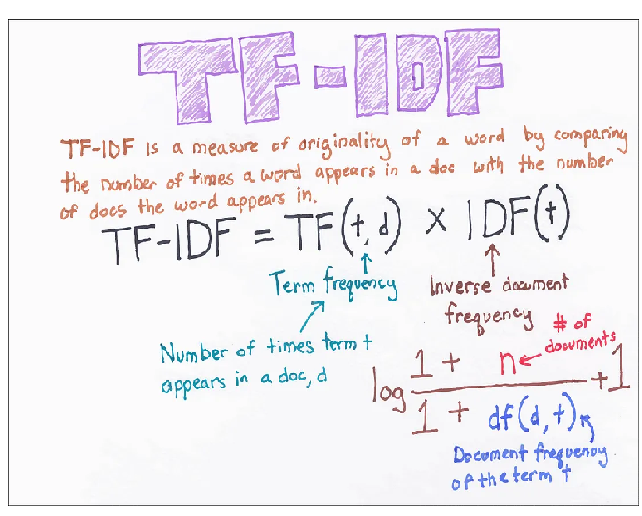
TF-IDF is used to evaluate how important a word is to a document in a collection or corpus. The importance of a word increases proportionally to the number of times it appears in the document but is offset by the frequency of the word in the corpus. This helps to identify words that are unique and important to a particular document, and down-weight words that are common across many documents.

Here's how TF-IDF works:

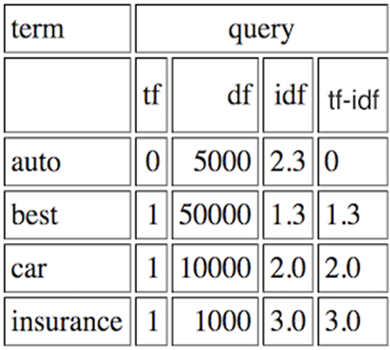
Term Frequency (TF): This measures how frequently a term occurs in a document. It is calculated by dividing the number of times a term appears in a document by the total number of terms in the document.

Inverse Document Frequency (IDF): This measures how important a term is by calculating the logarithm of the total number of documents in the corpus divided by the number of documents containing the term.

TF-IDF: This is calculated by multiplying the TF and IDF values for each term in a document. The resulting value reflects the importance of the term in the context of the document and the corpus as a whole.

TF-IDF is often used as a feature extraction technique for text classification algorithms. It can also be used for information retrieval, such as search engines, to rank documents based on their relevance to a search query.

Example of tf-ids distribution :-



If a word occurs in more documents then IDF decreases. The cell value is a multiplication of TF \* IDF.

1. **N-grams**

In natural language processing, n-grams are contiguous sequences of n words from a given text. N-grams are used to represent text as a numerical feature vector for use in machine learning models.

For example, consider the sentence: "The quick brown fox jumps over the lazy dog." Here are some examples of different n-grams:

Unigrams (n=1): ["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog"]

Bigrams (n=2): ["The quick", "quick brown", "brown fox", "fox jumps", "jumps over", "over the", "the lazy", "lazy dog"]

Trigrams (n=3): ["The quick brown", "quick brown fox", "brown fox jumps", "fox jumps over", "jumps over the", "over the lazy", "the lazy dog"]

In general, higher values of n result in more detailed representations of the text, but also require more data and computational resources.

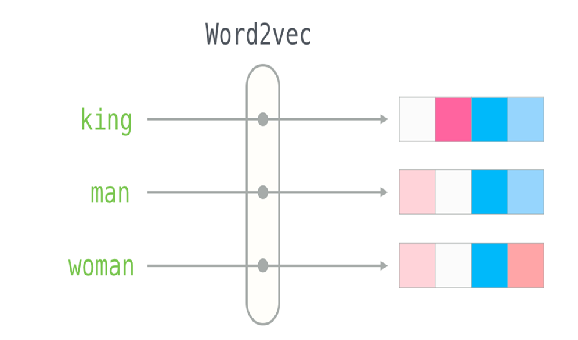
N-grams are often used in conjunction with other text feature extraction techniques such as Count Vectorizer or TF-IDF (term frequency-inverse document frequency). In these cases, the n-grams are extracted from the text and then represented as a numerical feature vector using the selected technique.

N-grams can be useful for tasks such as text classification, sentiment analysis, and language modeling. However, they can also be sensitive to noise in the data and may require careful tuning and regularization to avoid overfitting.

1. **Word2vec**

Till now we did text cleaning and we get our clean data after that we have to change our string into integer because our model cannot understand string. So we have to convert our whole string into integer with unique identity. We have tried count vectorizer but it is giving a very large no. of dimensions which is making difference in our model accuracy, due to more dimensions model is taking more time and space to execute.

Word2vec basically place the word in the feature space is such a way that their location is determined by their meaning i.e. words having similar meaning are clustered together and the distance between two words also have the same meaning.



1. **BERT (Bidirectional Encoder Representations from Transformers)**

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained deep learning model for natural language processing tasks developed by Google. BERT is based on the transformer architecture and is trained using a large amount of unannotated text data, as well as annotated data from specific tasks such as question answering and sentiment analysis.

The key innovation of BERT is its ability to capture the context of a word in a sentence by using a technique called masked language modeling. During training, a certain percentage of words in the input text are masked, and the model is trained to predict the masked words based on the context of the surrounding words. This forces the model to learn a rich representation of the context of each word, which can be used for a variety of downstream tasks.

BERT has achieved state-of-the-art results on a wide range of natural language processing tasks, including text classification, named entity recognition, sentiment analysis, and question answering. It has also been used for language generation tasks such as text summarization and machine translation.

In practice, BERT is typically fine-tuned on a specific task by adding a few task-specific layers on top of the pre-trained BERT model and training the entire model end-to-end on a smaller dataset. This approach has been shown to be effective for a wide range of natural language processing tasks.

However, one of the main drawbacks of BERT is its large size and computational requirements, which make it difficult to deploy in certain settings. There have been efforts to develop smaller and more efficient versions of BERT, such as DistilBERT and MobileBERT, which trade off some performance for reduced size and computational requirements.

### 

### Chapter 6

### Conclusion

In this Project, The Quora Question Pair dataset is used to predict the similarity between the two questions for duplicity using Machine Learning Algorithms. Before applying Machine Learning Models we done the Preprocessed the dataset and applied some models and we got the results as below mentioned.

By seeing the above table we can conclude that Random Forest classifier is the efficient among all other Classification Models with an accuracy of 80.35%, followed by XGBoost classifier with 79.7%, Decision Tree with 80.35% and Logistic Regression Classifier with 71%

### References

* Rishickesh Ramesh, R.P Ram Kumar, A. Shahina, Dr. A. Nayeemulla Khan “**Identification of Duplication in Questions Posed on Knowledge Sharing**

Platform Quora using Machine Learning Techniques” IJITEE Journal Paper

* AINAB IMTIAZ, MUHAMMAD UMER, MUHAMMAD AHMAD, SALEEM ULLAH, GYU SANG CHOI, ARIF MEHMOOD “**Duplicate Questions Pair Detection Using Siamese MaLSTM**”, **IEEE Journal Paper**
* Logistic Regression classifier

**“**[**https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regres**](https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/)[**sion/**](https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/)**”**

* Random Forest

“[**https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest**](https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest)”

* Decision Tree “[**https://www.slideshare.net/DakshGoyal3/decision-tree-69802132**](https://www.slideshare.net/DakshGoyal3/decision-tree-69802132)”
* XG Boost Classifier

“https://analyticsindiamag.com/xgboost-internal-working-to-make-decision-trees

**-and-deduce-predictions**”

* [**https://github.com/UdiBhaskar/Quora-Question-pair-similarity**](https://github.com/UdiBhaskar/Quora-Question-pair-similarity)
* [**https://www.kaggle.com/code/veerpalsingh/quora-question-pair-final**](https://www.kaggle.com/code/veerpalsingh/quora-question-pair-final)
* [**https://github.com/campusx-official/quora-question-pairs/blob/main/bow-with-p**](https://github.com/campusx-official/quora-question-pairs/blob/main/bow-with-preprocessing-and-advanced-features.ipynb)[**reprocessing-and-advanced-features.ipynb**](https://github.com/campusx-official/quora-question-pairs/blob/main/bow-with-preprocessing-and-advanced-features.ipynb)
* [**https://web.stanford.edu/~jurafsky/slp3/5.pdf**](https://web.stanford.edu/~jurafsky/slp3/5.pdf)
* **https://towardsdatascience.com/**
* **Chatgpt**