EDA IA2 -Plagiarism Checker Application of Levenshtein Distance

Technical Report · April 2021

DOI: 10.13140/RG.2.2.11011.78881

Dev Vora
Somaiya Vidyavihar

EDA IA2 - Plagiarism Checker Application of Levenshtein Distance

Dev Vora - 1814120
Jainam Zobaliya - 1814123
Vaibhavi Kundle - 1814127
Shreya Kulkarni- 1814129

April 2021

1 Introduction

Information technology is a constantly changing field. Accessing and exchanging data have now become very common in today's world. The data shared includes documents, personal files or certificates and good varieties of information and data. Thus, this in-depth usage of knowledge and technologies such as search engines have raised many issues related to data rights. On the internet, it is very simple to plagiarise previously published work with no repercussions. Through this research we are trying to say that Levenshtein distance can be described as a method for detecting plagiarism.

Plagiarism has become more common as a result of convenient access to other people's work. Plagiarism is whether you steal someone else's job or idea without their approval or in an unjustified case. This act can be carried out on any kind of internet data, but it is most generally carried out on text-based data such as documents, posts, and papers. As a result, there is a pressing need for a reliable method of detecting correlations between two documents. This re- port explains a way of checking plagiarism using Levenshtein Distance by comparing two documents and finding the similarity measure between them and finding the percent of plagiarism. Similarity measure is the measure of how similar 2 strings can be and how frequently it lies between 0 and 1 (0 - no similarity, 1 - total similarity).

2 Levenshtein Distance

The Levenshtein distance (LD) is a measure of how close two strings are. In our application, we'll be calling the text data which are being used as source string and the target string.

- The number of insert, substitute or delete actions needed is the distance between s and t. LD(s,t) = 0 when s and t are both "test", for example, so no transformations are needed. Already, the strings are similar.
- If s is "test" and t is "tent", then LD(s,t) = 2, since transforming s into t requires only one substitution (changing "s" to "n").

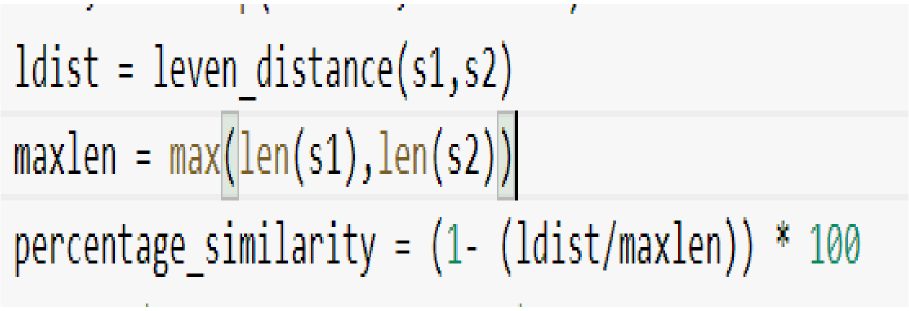
The more distinct the strings are, the greater the Levenshtein distance.

2.1 Applications of Levenshtein Distance

- DNA analysis
- Spell checking
- Speech recognition
- Plagiarism detection
- Auto suggestion of words

3 Methodology

This report is intended to demonstrate the implementation of Plagiarism checker using Levenshtein Distance algorithm.Levenshtein Distance algorithm is used to measure the distance difference between two sequences.Levenshtein distance between two string is determined based on minimum number of changes needed to make the transformation of one string to another.The application gives two options to the user, first to check plagiarism in two Strings and second to check plagiarism between two files. The application takes two inputs from the user i.e. the sample text files/paragraph and the text file/paragraphs to be tested.Levenshtein distance between the two files or sentences is calculated after cleaning the texts and the similarity percentage or the plagiarism percentage is presented to the user as the result. The Levenshtein distance is calculated sentence wise.Initially we split the given text into sentences based on full stop and a space ['. '] as the splitting criteria. It is followed by text cleaning where all the stop-words from the text is removed and the text is also converted in lower-case to meet the function condition. The Levenshtein distance gives us the dissimilarity between the texts in such a way that more the Levenshtein Distance more the dissimilarity, so to calculate the similarity percentage we use the formula: Here, we are calculating the similarity by subtracting the ratio of



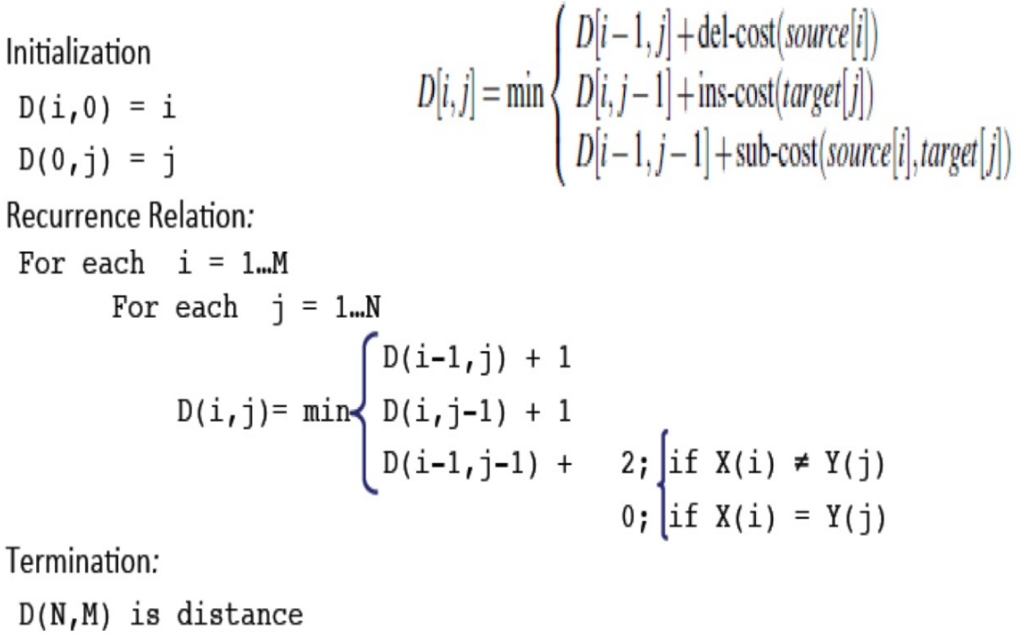
```
ldist = leven_distance(s1,s2)
maxlen = max(len(s1),len(s2))
percentage_similarity = (1- (ldist/maxlen)) * 100
```

Figure 1: Formula used for Percentage Similarity

Levenshtein distance and maximum length from the two strings from 1.We then multiply this value by 100 to get the similarity percentage. If this calculated similarity percentage is greater than 50% we consider the line as plagiarised and we increment the plag_counter by 1. Plagiarism percentage is calculated by taking the ratio of plag_counter and total lines of the source sentence or source files. After calculating plagiarism percentage, we have given an option to download the plagiarism report which displays the plagiarism percentage as well as the string which is detected as plagiarised.

3.1 Levenshtein Pseudo-code

A simple recursive implementation of the Levenshtein distance calculator using dynamic programming is seen in the following pseudo-code :



$$D[i,j] = \min \begin{cases} D[i-1,j] + \text{del-cost}(\mathit{source}[i]) \\ D[i,j-1] + \text{ins-cost}(\mathit{target}[j]) \\ D[i-1,j-1] + \text{sub-cost}(\mathit{source}[i],\mathit{target}[j]) \end{cases}$$

```
Initialization
 D(i,0) = i
 D(0,j) = j
Recurrence Relation:
 For each  i = 1…M
       For each  j = 1…N
                          ⎧ D(i-1,j) + 1
           D(i,j)= min ⎨ D(i,j-1) + 1
                          ⎩ D(i-1,j-1) +   2; ⎧if X(i) ≠ Y(j)
                                           0; ⎩if X(i) = Y(j)
Termination:
 D(N,M) is distance
```

Figure 2: Levenshtein Pseudo-code

3.2 Workflow Diagram:

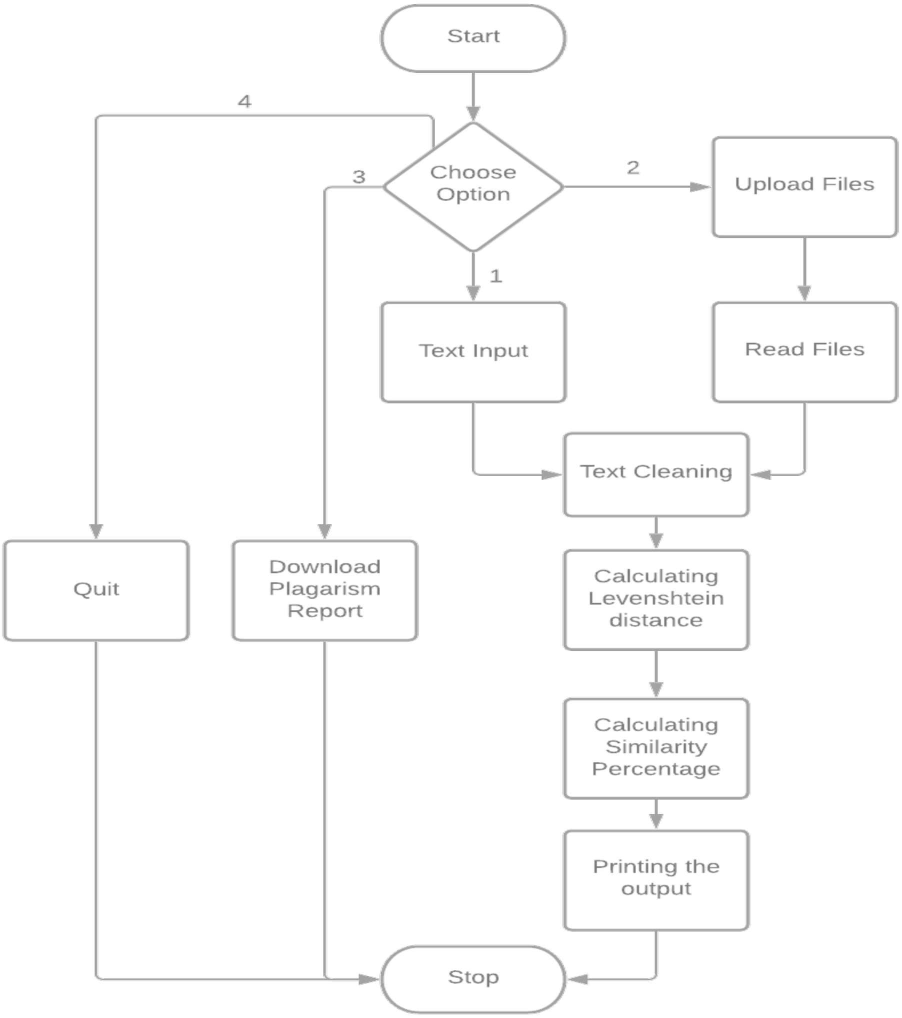
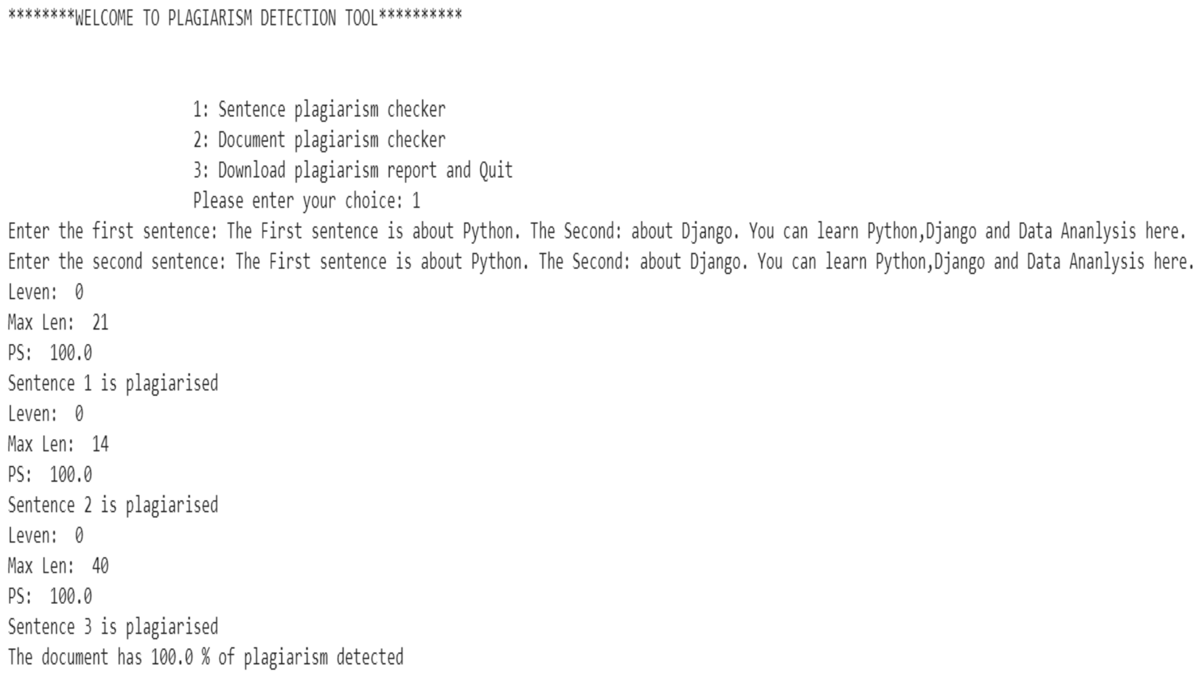


Figure 3: Workflow Diagram

4 Implementation

1. Implemation in detail - https://drive.google.com/file/d/1tXkqv7g7lmGIH6r5plAU_NqyX0w0gW77/view?usp=sharing

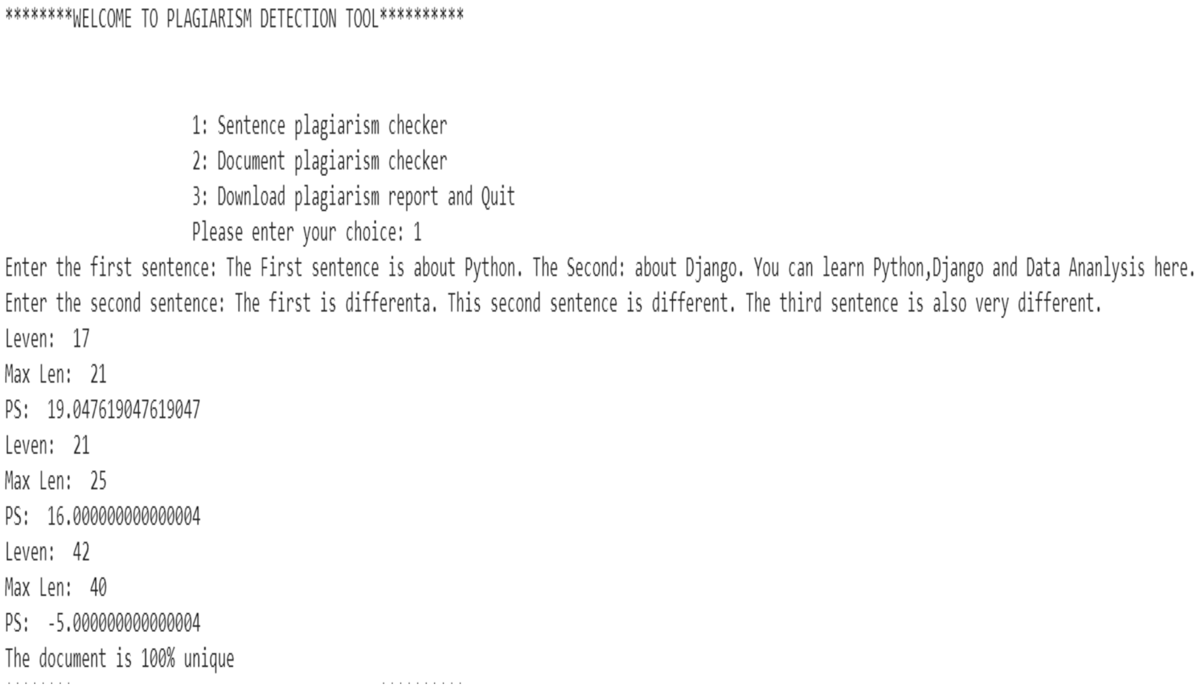
5 Results and Discussion



```
********WELCOME TO PLAGIARISM DETECTION TOOL**********


                    1: Sentence plagiarism checker
                    2: Document plagiarism checker
                    3: Download plagiarism report and Quit
                    Please enter your choice: 1
Enter the first sentence: The First sentence is about Python. The Second: about Django. You can learn Python,Django and Data Ananlysis here.
Enter the second sentence: The First sentence is about Python. The Second: about Django. You can learn Python,Django and Data Ananlysis here.
Leven:  0
Max Len:  21
PS:  100.0
Sentence 1 is plagiarised
Leven:  0
Max Len:  14
PS:  100.0
Sentence 2 is plagiarised
Leven:  0
Max Len:  40
PS:  100.0
Sentence 3 is plagiarised
The document has 100.0 % of plagiarism detected
```

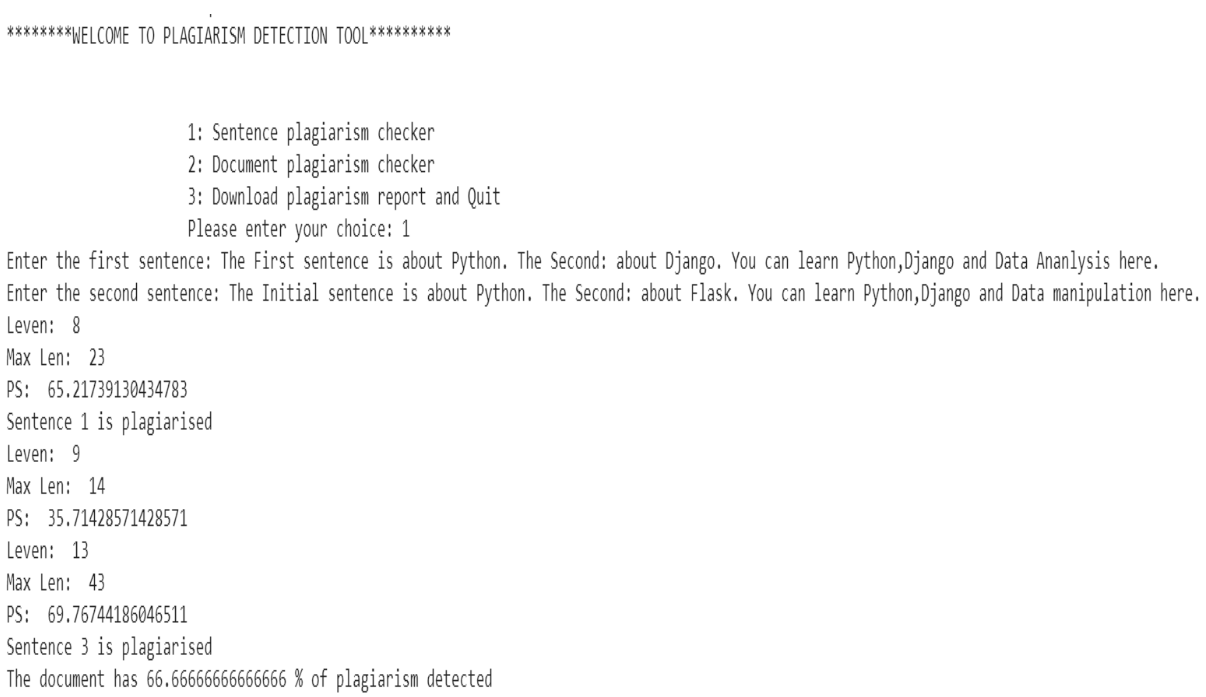
Figure 4: Case1



```
********WELCOME TO PLAGIARISM DETECTION TOOL**********


                      1: Sentence plagiarism checker
                      2: Document plagiarism checker
                      3: Download plagiarism report and Quit
                      Please enter your choice: 1
Enter the first sentence: The First sentence is about Python. The Second: about Django. You can learn Python,Django and Data Ananlysis here.
Enter the second sentence: The first is differenta. This second sentence is different. The third sentence is also very different.
Leven:  17
Max Len:  21
PS:  19.047619047619047
Leven:  21
Max Len:  25
PS:  16.000000000000004
Leven:  42
Max Len:  40
PS:  -5.000000000000004
The document is 100% unique
```

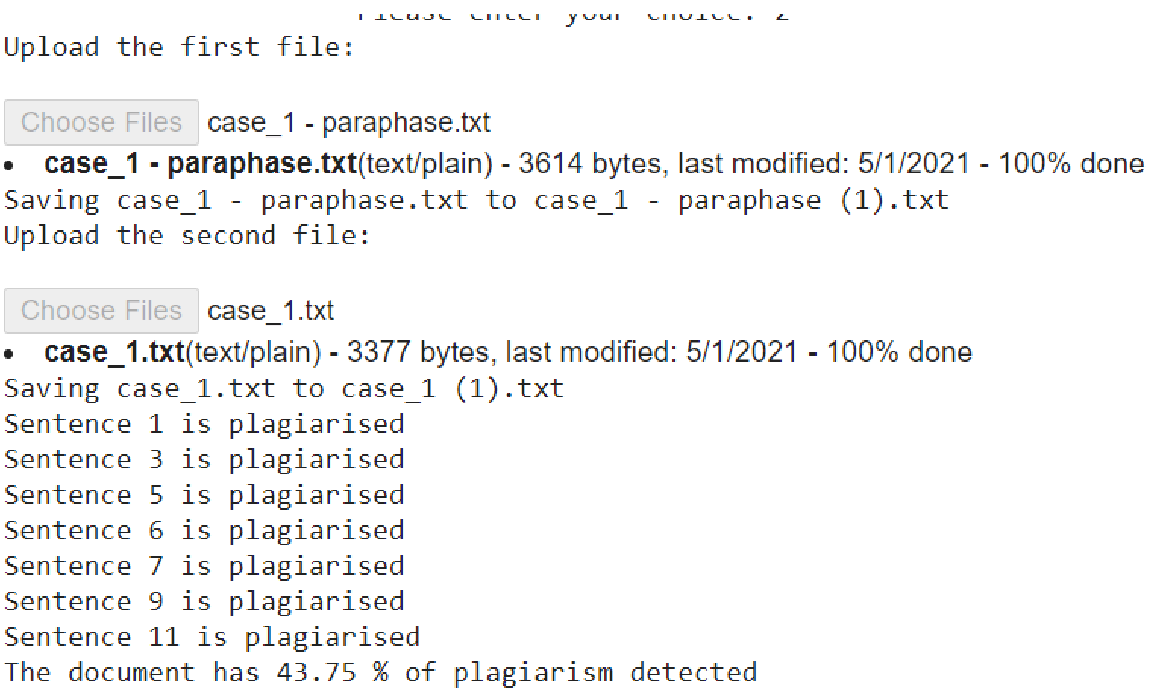
Figure 5: Case 2



```
********WELCOME TO PLAGIARISM DETECTION TOOL**********


                      1: Sentence plagiarism checker
                      2: Document plagiarism checker
                      3: Download plagiarism report and Quit
                      Please enter your choice: 1
Enter the first sentence: The First sentence is about Python. The Second: about Django. You can learn Python,Django and Data Ananlysis here.
Enter the second sentence: The Initial sentence is about Python. The Second: about Flask. You can learn Python,Django and Data manipulation here.
Leven:  8
Max Len:  23
PS:  65.21739130434783
Sentence 1 is plagiarised
Leven:  9
Max Len:  14
PS:  35.71428571428571
Leven:  13
Max Len:  43
PS:  69.76744186046511
Sentence 3 is plagiarised
The document has 66.66666666666666 % of plagiarism detected
```

Figure 6: Case 3

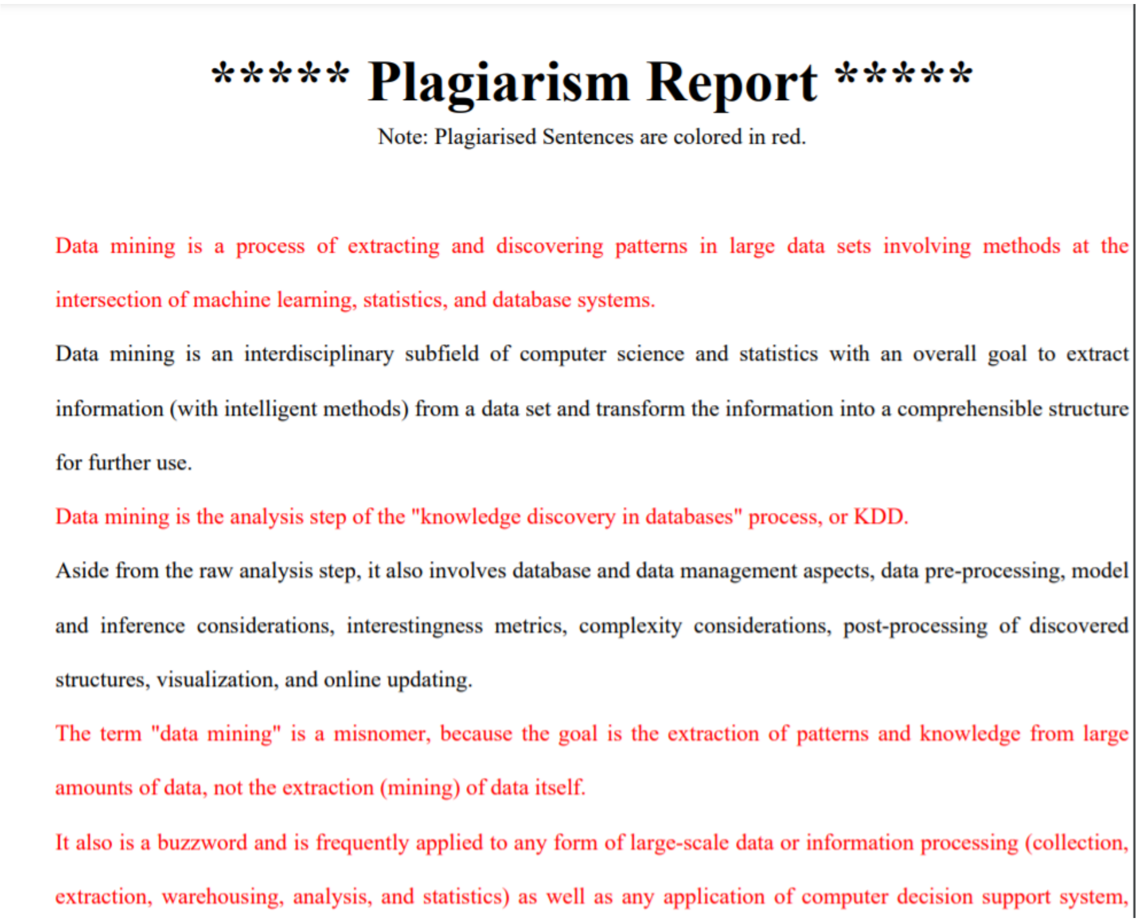


```
                 Please enter your choice: 2
Upload the first file:

Choose Files case_1 - paraphase.txt
• case_1 - paraphase.txt(text/plain) - 3614 bytes, last modified: 5/1/2021 - 100% done
Saving case_1 - paraphase.txt to case_1 - paraphase (1).txt
Upload the second file:

Choose Files case_1.txt
• case_1.txt(text/plain) - 3377 bytes, last modified: 5/1/2021 - 100% done
Saving case_1.txt to case_1 (1).txt
Sentence 1 is plagiarised
Sentence 3 is plagiarised
Sentence 5 is plagiarised
Sentence 6 is plagiarised
Sentence 7 is plagiarised
Sentence 9 is plagiarised
Sentence 11 is plagiarised
The document has 43.75 % of plagiarism detected
```

Figure 7: Checking Plagiarism for files



***** Plagiarism Report *****

Note: Plagiarised Sentences are colored in red.

Data mining is a process of extracting and discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.

Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use.

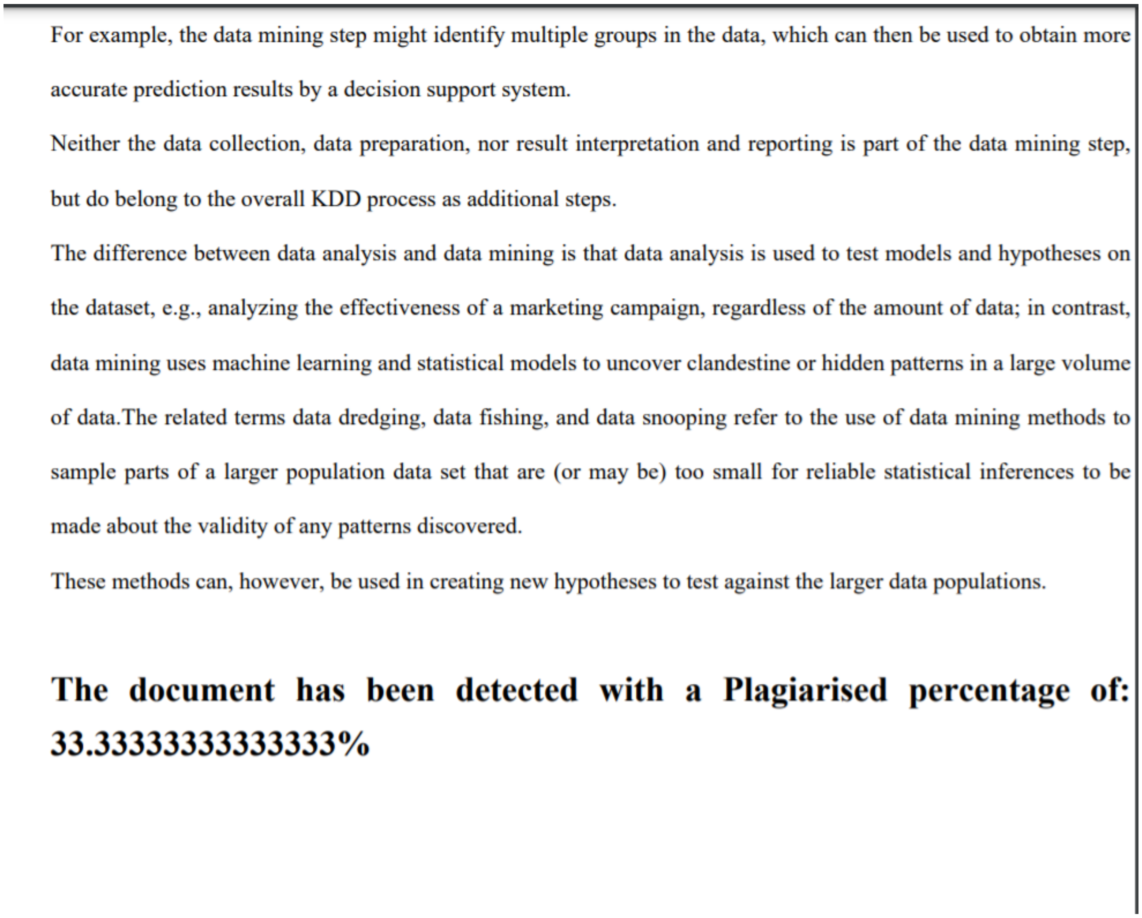
Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD.

Aside from the raw analysis step, it also involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.

The term "data mining" is a misnomer, because the goal is the extraction of patterns and knowledge from large amounts of data, not the extraction (mining) of data itself.

It also is a buzzword and is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system,

Figure 8: Plagiarism Report - I



For example, the data mining step might identify multiple groups in the data, which can then be used to obtain more accurate prediction results by a decision support system.

Neither the data collection, data preparation, nor result interpretation and reporting is part of the data mining step, but do belong to the overall KDD process as additional steps.

The difference between data analysis and data mining is that data analysis is used to test models and hypotheses on the dataset, e.g., analyzing the effectiveness of a marketing campaign, regardless of the amount of data; in contrast, data mining uses machine learning and statistical models to uncover clandestine or hidden patterns in a large volume of data.The related terms data dredging, data fishing, and data snooping refer to the use of data mining methods to sample parts of a larger population data set that are (or may be) too small for reliable statistical inferences to be made about the validity of any patterns discovered.

These methods can, however, be used in creating new hypotheses to test against the larger data populations.

The document has been detected with a Plagiarised percentage of: 33.333333333333333%

Figure 9: Plagiarism Report - II

5.1 Drawbacks of the implementation:

1. This type of plagiarism checker is not guaranteed to give the optimal plagiarism percentage.Since we are calculating the sentence by sentence plagiarism percentage.

2. For cases where the text is having similar words or negatives, there can be chances of false positive case as the algorithm only considers the amount of percentage similarity between them.

3. The threshold is only based on the analysis from the research papers and is open to many possible optimization.

6 Conclusion

Plagiarism checker using Levenshtein distance algorithm was implemented. The threshold condition for the application was that, if the similarity percentage is greater than 50% then we consider the sentence to be plagiarised. Since it can reflect the amount of editing required to transform one sentence into the other, the Levenshtein algorithm is an important method for detecting similarities between two sentences. It may be used as a supplement to current plagiarism detection software that mostly employs string pattern matching algorithms. However, further analysis is needed to decide the most appropriate threshold to use for determining whether two sentences are same or not.

7 References

1. https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Makalah/Makalah-Stima-2019-071.pdf

2. https://ieeexplore.ieee.org/document/8864339

3. https://www.ieee.org/publications/rights/plagiarism/id-plagiarism.html

4. https://online.stat.psu.edu/stat508/lesson/1b/1b.2/1b.2.1

5. https://people.cs.pitt.edu/~kirk/cs1501/assignments/editdistance/Levenshtein%20Distance.htm

6. https://copyleaks.com/plagiarism-checker/what-is-plagiarism