# PLAGIARISM DETECTOR USING PYTHON

**A Mini Project Report**

**Submitted to the**

**Faculty of Engineering**

**in fulfillment of the requirements for the award of the degree of**

**Bachelors in Technology**

**in**

**School of Electronics & Communication Engineering**


Submitted by :

1. KUMAR RAUSHAN (19BEC040)
2. SALLA CHAITANYA KRISHNA (19BEC077)
3. SIDDHANT VARDHAN SINGH (19BEC088)


UNDER THE SUPERVISION OF



**SCHOOL OF ELECTRONICS & COMMUNICATION ENGINEERING**

**SHRI MATA VAISHNO DEVI UNIVERSITY**

**KATRA-182320, JAMMU AND KASHMIR (INDIA)**


**February, 2022**


**SHRI MATA VAISHNO DEVI UNIVERSITY**

**School of Electronics and Communication Engineering**

# CANDIDATE'S DECLARATION

---

I hereby certify that the work which is being presented in the report, entitled "PLAGIARISM CHECKER USING PYTHON" for the award of the degree of "Bachelors of Technology" submitted in the School of Electronics & Communication, Faculty of Engineering & Technology, Shri Mata Vaishno Devi University, Katra is an authentic record of my own work carried out under the supervision of Dr. Vikram Singh, Associate Professor, School of Electronics & Communication Engineering, Shri Mata Vaishno Devi University, Katra.

The matter presented in this report has not been submitted by me for the award of any degree/diploma of this or any other University/Institute.

Student's Name and Signature

1. Kumar Raushan Parag (19BEC040)
2. Salla Chaitanya Krishna (19BEC077)
3. Siddhant Vardhan Singh (19BEC088

**SHRI MATA VAISHNO DEVI UNIVERSITY**

**School of Electronics and Communication Engineering**

# CERTIFICATE

This is to certify that the project entitled "PLAGIARISM CHECKER USING PYTHON" being submitted by Kumar Raushan (19BEC040), Salla Chaitanya Krishna (19BEC077) to School of Electronics & Communication of "Shri Mata Vaishno Devi University, Katra" for the award of the degree of **"Bachelors of Technology"** in "Electronics & Communication Engineering", is a bonafide research work carried out by them under my supervision and guidance. Their project report has reached the standard of fulfilling the requirements of regulations relating to degree. The report is an original piece of research work and embodies the findings made by the research scholar himself.

The results presented have not been submitted in part or in full to any other University/Institute for the award of any degree or diploma.

Dr. Vikram Singh

Supervisor

Associate Professor, School of Electronics & Communication Engineering

Shri Mata Vaishno Devi University, Katra -182320, Jammu and Kashmir (India)


**SHRI MATA VAISHNO DEVI UNIVERSITY**

**School of Electronics and Communication Engineering**

# ACKNOWLEDGEMENTS

I would like to thank Dr.Vikram singh who gave me
the opportunity to do this internship . It has been a great learning experience with some hands-on project within the stipulated time period.

I thank my team members who helped me and collaborated with me in this internship. Those guys have been amazing at solving certain problems . I really appreciate their efforts and the moments we enjoyed together.

Moreover, I want to thank my family members without whom this internship would never have been possible especially in the gloomy days of the lockdown. Thanks for supporting me always and making it even easier for me .

Students' Name

Kumar Raushan Parag(19bec040)

Salla Chaitanya Krishna(19bec077)

Siddhant Vardhan Singh(19bec088)


**SHRI MATA VAISHNO DEVI UNIVERSITY**

**School of Electronics and Communication Engineering**

# ABSTRACT

---

My Objective is to create a Plagiarism Detector using Python – High Level Language. In this project, we're going to make a Plagiarism Detector in Python using machine learning techniques such as word2vec and cosine similarity in just a few lines of code.

Once finished our plagiarism detector will be capable of loading a student's assignment from files and then compute the similarity to determine if students copied from each other. The percentage of similarity is given in the form of percentage per match . In case multiple files are uploaded, the program detects them and makes all the possible matches.

For this Project, we need to install the scikit library using Python Installation Package. The same can be achieved using the version control of the libraries by msys.

Kumar Raushan (19BEC040)

Salla Chaitanya Krishna (19BEC077)

Siddhant Vardhan Singh (19BEC088)

# Contents

# Introduction

In this Project, We are making Plagiarism checker Using Python - High level Language. The best plagiarism checker should be able to detect plagiarism most accurately even if the original phrasing has been altered. The tool should also provide a clear, comprehensive plagiarism report.

Plagiarism is any unauthorized use of parts or the whole of any article without giving proper credit to the original writer. Any unethical copying of any writing is basically considered theft, and therefore it takes away the originality and trustworthiness of the content.

Consequently, the content creators, including the bloggers, researchers, and students, must know the importance of plagiarism. It affects the writer who treasures his writing and can also change the content creator's career adversely. Therefore, they must be careful about plagiarism.

## Consequences of Plagiarism:

- Most academic institutions include a dissertation paper as a part of their curriculum to ensure that they learn and think at a deeper level. Therefore, if a student submits an article with plagiarized content, the academic institution may take action against the student.

- Why is avoiding plagiarism important? It can have consequences in the form of probation and even fines in some cases. If plagiarism is being done by bloggers, then as a consequence, they may lose their readership, and if the writer wants, he can also take legal steps against the content creator. All cases of plagiarism can sabotage the career of the content creator. To avoid plagiarism, the content creator must use plagiarism checker tools.

Our plagiarism detector will be capable of loading a student’s assignment from files and then compute the similarity to determine if students copied from each other. The percentage of similarity is given in the form of percentage per match . In case multiple files are uploaded, the program detects them and makes all the possible matches.

# About the project

The Project “Plagiarism Checker” is very relevant in this pandemic situation, as all the major works have gone online. All the projects, assignments, and thesis which were to be submitted in different curriculum across the world have gone online. So, students are made to submit their work in PDF, DOCX format. This plagiarism checker will help to check the authenticity of the work presented.

This project will help to retain the authenticity of the work. Any unethical copying of any writing is basically considered theft, and therefore it takes away the originality and trustworthiness of the content.

This project can be used in the implementation of familiar technologies such Face Recognition, Handwriting to Text Convertor, and many more AI Based Projects which seems to be magic in today’s world.

This project takes the help of PYTHON as the programming Language (most significantly it’s scikit learn library).

## SCIKIT LEARN LIBRARY IN PYTHON:

It is used in regression, clustering, preprocessing, model selection, classification, and dimensionality reduction. It provides dozens of machine learning algorithms and models called estimators. Each estimator can be fitted to some data using its fit method.

Some points about SCIKIT LEARN LIBRARY

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

To use this library, we need to install it first on our PC. For installing it we need to open command terminal in our PC and run command “pip install -U scikit-learn”

Here is a simple example where we fit a RandomForestClassifier to some very basic data:

```
>>> from sklearn.ensemble import RandomForestClassifier
>>> clf = RandomForestClassifier(random_state=0)
>>> X = [[ 1,  2,  3],  # 2 samples, 3 features
...      [11, 12, 13]]
>>> y = [0, 1]  # classes of each sample
>>> clf.fit(X, y)
RandomForestClassifier(random_state=0)
```

The fit method generally accepts 2 inputs:

The sample matrix (or design matrix) X. The size of X is typically (n_samples, n_features), which means that samples are represented as rows and features are represented as columns.

The target values y which are real numbers for regression tasks, or integers for classification (or any other discrete set of values). For unsupervised learning tasks, y does not need to be specified. y is usually a 1d array where the i th entry corresponds to the target of the i th sample (row) of X.

Both X and y are usually expected to be numpy arrays or equivalent array-like data types, though some estimators work with other formats such as sparse matrices.

Once the estimator is fitted, it can be used for predicting target values of new data. You don’t need to re-train the estimator:

```
>>>
>>> clf.predict(X)  # predict classes of the training data
array([0, 1])
>>> clf.predict([[4, 5, 6], [14, 15, 16]])  # predict classes of new data
array([0, 1])
```

# Working Methodology

The project that we have made uses of Python as a programming language and In python we have specifically used the “ scikit-learn ” library to implement our project.

This library is used in the conversion of the raw data into the encrypted numeric vectors which can be compared since the humanlike process of identifying each symbol and comparing it takes a lots of time by the machine and it is easy to compare the binary file or the encrypted vector symbol file generated.

For the program to be running it requires two or more basic files in the text format to be called for the comparison. The text files are then converted into their vector form and compared bit-by-bit.

The library has a basic methodology working underneath the hood . It simply makes the use of lists to compare the data by keeping the original data vector in an array and making multiple lists for the matching symbols . The matched symbol list is then compared in the number of characters with the original list and the percentage calculation is displayed . This way of comparison is very efficient in applying AI or ML algorithms where we need to identify the object based on the training data such as the technologies used in the face recognition and object identifier.

The implementation process also involves the checking of the garbage files since those files are most likely to have a very small portion of matching characters with the general files.

# LITERATURE REVIEW

We propose novel technically oriented typologies for plagiarism prevention and detection efforts, the forms of academic plagiarism, and computational plagiarism detection methods. We show that academic plagiarism detection is a highly active research field. Over the period we review, the field has seen major advances regarding the automated detection of strongly obfuscated and thus hard-to-identify forms of academic plagiarism. These improvements mainly originate from better semantic text analysis methods, the investigation of non-textual content features, and the application of machine learning. We identify a research gap in the lack of methodologically thorough performance evaluations of plagiarism detection systems. Concluding from our analysis, we see the integration of heterogeneous analysis methods for textual and non-textual content features using machine learning as the most promising area for future research contributions to improve the detection of academic plagiarism further.

# KEY FEATURES FOR PLGIARISM CHECKER

## 1. Billions of web pages

This tool has the ability to check plagiarism ,after extension, by matching your content against billions of webpages on the Internet . Once you upload your content, it will automatically run it against every existing content on the web within seconds, making it the most sophisticated yet fastest plagiarism scanner you'll ever come across till date.

## 2. Automatic rewriting feature

It has an option for automatically rewriting the content you run on it in just one click. If your content contains plagiarized work, all you have to do is click on the rewrite option and you'll be taken to our auto-paraphrasing tool, where your content will be updated immediately. This is a built-in feature available right inside the tool.

## 3. Multiple document formats

Our similarity checker allows you to upload different formats of documents including .doc, .docx, .txt, .tex, .rtf, .odt, and .pdf. This means it does not matter what format your content takes, as long as it is digital, our tool will do the rest of the work.

## 4. Reporting option

Our anti-plagiarism engine comes with a reporting option which allows you to download a report of the plagiarism search you run. This means you now have some sort of evidence to send across to the relevant parties and a record to keep. Awesome!

## 5. Multiple languages

This feature allows you to check plagiarism on documents in other languages other than English. So whether your content is written in русский, italiano, français, Português, Español, Deutsche, our tool speaks your language.

## 6. Cloud compatibility

Live in the cloud? Our originality checker is able to check content from the cloud, including Google Drive and Dropbox. Though this cloud space is limited for now since we are using our google drive space. Simply choose where your content lives in the cloud and pick the particular piece of document you want to run, and our copyright checker will do the rest.

## 7. Local storage

If your content is rather local, living in a file within your computer, then you can upload it directly from the local storage.

## 8. Percentage gauges

Once you've uploaded your content and clicked to check for plagiarism, our duplication checker will show you, in percentages, the levels of both plagiarized and unique content in the document. For example, it'll let you know that 82% of the content is unique while 18% is plagiarized.

## 9. A list-based, sentence-wise result

The tool does not stop at showing you the percentage levels of plagiarized and unique content. It also shows you, in a list format for easy detection, both plagiarized (if any) and unique areas of the content piece, sentence-by-sentence. Plagiarized sentences are shown in red while the unique ones are shown in green for your convenience.

## 12. Highlighted document view

With just one click, you can also see the result in a document view, where the whole content is displayed in one document and the plagiarized materials are highlighted in red. This feature is achieved with the integration of web based tools like js frameworks.

## 13. Ability to view matched results

Right within the tool, you can view the external content that matches the red sentences in your document. Plus, the URL of the external webpage is added for a quick and easy examination of the

content.

## 14. One-click comparison feature

After the results are in, you can click on the "Compare" button on any red (plagiarized) line to go to Google and compare that particular content with similar ones already published on the web. Great for finding where the plagiarized content is coming from

## 15. Exclude Specific URL

If you don't want to detect plagiarism for a specific URL? Simply Insert that URL in the Exclude URL box and that'll be done for you automatically, Copied (plagiarized) content from that URL won't be countered as plagiarism.

## 16. Plagiarism Checker API

If you want to develop a real-time multitasking plagiarism detection system, incorporated into your website, then we have your back. The Plagiarism Checker API offers you a great API integration solution. This completely eliminates the need to check each and every article for every student individually and saves you hours upon hours of work and headache. You can check plagiarism for multiple essays, thesis or assignments of your students in just one click. This also works great for big websites who accept dozens of articles from contributors frequently.

# FEATURES OF PLAGIARISM DETECTOR

## Upload File and URL

Our plagiarism detector allows you to upload content of around 1500 words from your computer or from the cloud or you can directly paste the URL of a webpage for a quick and free plagiarism check. It supports various file types such as doc, Docx, pdf, txt, etc.

## Increased Word Count

There are a plethora of free plagiarism detection tools available online. However, we brag about it to be the best due to many reasons. Unlikely other free tools available online are offering a maximum limit of 800 to 1000 words but we offer 1500 words. Our plagiarism checker 1500 words free helps you to check content in one go without cutting content into pieces. Furthermore, our tool displays instant results with percentages, which outweighs all other available plagiarism software.

## Compare Duplicate Content

In the case of duplication, you will see links to the URL of the websites that contain a similar passage anchored by "Compare". You can check plagiarism to determine the cause of similarity in the detailed plagiarism test report.

## Download and Share Reports

Our free online plagiarism checker can give you the option to download a detailed plagiarism test report for your content by clicking "Download Report". You can also share this report. Furthermore, click on "Start New Search" to perform a plagiarism check free for new content.Though we have not made it online yet but the local terminal is able to perform all such

tasks.

## Quick and efficient plagiarism Detection

Our  duplicate checker is specially designed to detect even the minutest of replication. It also provides you with a list of similar content pieces so you can take the appropriate action instantly. Our plagiarism detector is super easy to use and has featured far better than you hardly find in paid similar tools.

## Highlighted Results

Once the plagiarism detection is completed, the tool will display your text by highlighting the unique and plagiarized portions. The text in green color represents uniqueness, while the red color demonstrates plagiarized chunks.

## Comprehensive Plagiarism Score

Besides highlighting your text, our plagiarism tool also provides you with an accurate plagiarism score. What proportion of your text is unique and plagiarized? The plagiarism detector will give you the answer to this query by displaying the percentages of unique and plagiarized content.

## Sentence and Document View

This plagiarism checker online allows its users to display results in two modes: sentence and document. In the sentence-wise mode, each sentence of your text is separated, allowing you to easily navigate the plagiarized phrases. Whereas the document view displays the whole content while highlighting the unique and plagiarized paragraphs or sentences.

# WHO CAN BE BENEFITED FROM OUR PLAGIARISM DETECTOR

## Web Entrepreneurs/ Web Owners

The ultimate aim of web owners is to publish unique content on their websites. This task can be achieved if they check duplication by using this website plagiarism checker. Our tool will deeply analyze the text and match it with its directory to let you know if duplication exists or not.

## Bloggers

A blogger has to come up with original ideas and text that help him/her flourish in the web world. If you are 100% sure that your content doesn't contain any sort of duplication and grammar errors, then you must check for plagiarism free to identify if any other source has used your text without permission. Also, ensure that your content is free of grammatical errors by utilizing a professional grammar check.

## Freelance Content Writers

A freelance content writer cannot flourish without providing unique content. Among freelancers, the competition is quite severe for content writing jobs. Save your reputation and credibility in the freelance market by using our copyright checker. As it helps you assure content originality and plagiarism.

## Students

Duplication is considered a crime in academics; therefore, students should check for plagiarism in their assignments with a free plagiarism checker for students before submitting them. This will help students in properly citing the sources from where they have taken assistance while creating an assignment.

## Authors

Becoming a well-recognized author isn't a piece of cake, and a little mistake of duplication can drown an author's career. The copyright checker will help you in avoiding duplication from your work, as it will inform you even about unintentional plagiarism. Before publishing any of your work, make sure to conduct a plagiarism test.

## Reporters and Journalists

The reporters and journalists should come up with original content to maintain their fame and respect among the audience. If your work is being published under someone else's name, it can hurt your career badly. Therefore, the similarity checker is the best way out of this nuisance. You can keep a regular check on your work's authenticity with the help of this online tool.

## Professors and Teachers

As a professor or teacher, it's quite hectic to manage workload and deeply investigate students' work to detect plagiarism. The manual process can take hours to check a single copy; hence, you can take advantage of a plagiarism checker for teachers that provides you with comprehensive results within a matter of seconds.

# WORKING CODE

```
import os
from numpy import vectorize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
sample_files = [doc for doc in os.listdir() if doc.endswith('.txt')]
sample_contents = [open(File).read() for File in sample_files]
vectorize = lambda Text: TfidfVectorizer().fit_transform(Text).toarray()
similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2])
vectors = vectorize(sample_contents)
s_vectors = list(zip(sample_files, vectors))
def check_plagiarism():
    results = set()
    global s_vectors
    for sample_a, text_vector_a in s_vectors:
        new_vectors = s_vectors.copy()
        current_index = new_vectors.index((sample_a, text_vector_a))
        del new_vectors[current_index]
        for sample_b, text_vector_b in new_vectors:
            sim_score = similarity(text_vector_a, text_vector_b)[0][1]
       sample_pair = sorted((sample_a, sample_b))
            score = sample_pair[0], sample_pair[1], sim_score
            results.add(score)
    return results
for data in check_plagiarism():
    print(data)
```

# Code Explained

**Installing scikit :**

In order to use scikit-learn library, we need to install it first:

- We can install it using command prompt
- Run command "pip install –U scikit-learn"

```
pip install scikit-learn
```

**Import OS:**

- We are importing os module in our program.
- The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

```
# importing os module
import os
```

**Using numPy :**

- Numpy is a python library which works with multidimensional array, and it also has in-built functions for fourier transforms and all.
- 
- numpy.vectorize takes a function f:a->b and turns it into g:a[]->b[].

```
from numpy import vectorize
```

## From sklearn.feature_extraction.text import TfidfVectorizer :

Tf means term-frequency while tf-idf means term-frequency times inverse document-frequency. This is a common term weighting scheme in information retrieval, that has also found good use in document classification.

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

## from sklearn.metrics.pairwise import cosine_similarity :

- Compute cosine similarity between samples in X and Y.
- Cosine similarity, or the cosine kernel, computes similarity as the normalized dot product of X and Y:
- K(X, Y) = <X, Y> / (||X||*||Y||)

```
from sklearn.metrics.pairwise import cosine_similarity
```

## Source_files and source_contents:

- Source_files identify the nature of the file to be in text format.
- Source_contents keeps the contents of the file , it is accessed after the verification of the file.

```
sample_files = [doc for doc in os.listdir() if doc.endswith('.txt')]
sample_contents = [open(File).read() for File in sample_files]
```

## Vectorize and similarity :

- It is used to transform the file text using vector array fit.
- Similarity parameter combines the document pairwise using its cosine similarity.

```
vectorize = lambda Text: TfidfVectorizer().fit_transform(Text).toarray()
similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2])
```

## Vectors and s_vectors:

- Vectors keeps the vectorized sample_contents.
- S_vectors pairs each sample_file with its vectors.

```
vectors = vectorize(sample_contents)
s_vectors = list(zip(sample_files, vectors))
```

## Check_plagiarism function:

- This function performs the required task using the comparision in the array of vectors.
- It finally assignes the simulation score to the variable sim_score.

```
def check_plagiarism():
    results = set()
    global s_vectors
    for sample_a, text_vector_a in s_vectors:
        new_vectors = s_vectors.copy()
        current_index = new_vectors.index((sample_a, text_vector_a))
        del new_vectors[current_index]
        for sample_b, text_vector_b in new_vectors:
            sim_score = similarity(text_vector_a, text_vector_b)[0][1]
            sample_pair = sorted((sample_a, sample_b))
            score = sample_pair[0], sample_pair[1], sim_score
            results.add(score)
    return results
```

## function calling :

- The function is called using this code snippet for each segment in the datafile.

```
for data in check_plagiarism():
    print(data)
```

## Final output:

- The percentage of similarity between the files is clearly visible pairwise.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP\Documents\plagiarism_checker> python .\plagiarism_checker
('chaitanya.txt', 'raushan.txt', 0.14806887549598566)
('chaitanya.txt', 'siddhant.txt', 0.5008213456319839)
('raushan.txt', 'siddhant.txt', 0.17082115673289683)
PS C:\Users\HP\Documents\plagiarism_checker>
```

# IMPORTANCE OF PLAGIARISM DETECTOR

Whether you know it or not, plagiarism does affect you in some way. It doesn't matter whether you're the content creator or the one who consumes the content, plagiarism affects us all.

As a content creator (writer, author, researcher, student, freelancer, blogger, social media manager, etc. It's no news that you should be regularly checking to be sure that nobody is copying your content without your permission or giving you credit.

But much more than that, you also have to always run your content on a plagiarism detection tool before publishing or submitting it to be sure that your work does not contain plagiarism.

Why? Because plagiarism is an act of academic dishonesty, a breach of journalistic ethics, and above all, a publishing crime.

As such, you don't want to fall victim. If your work contains plagiarized content, whether intentionally or by ignorance, you stand to face serious penalties including:

- Legal actions
- Monetary restitution and fines
- Damaged reputation

This is not to mention other consequences like SEO content duplication penalties and lowered rankings, lost trust, possible academic sanctions, and more.

Now, on the other end of the spectrum, as a content consumer or user (reader, professor or teacher who vets students' work, client of freelance writers, etc.), it is equally important to check for plagiarism before accepting or taking action on any content you come across or submitted to you.And that is why we created the Plagiarism Checker by Small SEO Tools.

This tool is carefully designed to help you easily and quickly detect plagiarism in any digital text-based content.

It is used and trusted by millions of people all around the world and can easily boast of being the single most sophisticated, feature-rich, user-friendly content checker online.

# Project :

https://siddhant-vardhansingh.github.io/mini-project/

https://github.com/Siddhant-vardhansingh/mini-project

# BIBLIOGRAPHY:


- **Kalebu Jordan (software developer in Tanzania)**
- **Mikael codes (Youtube tutor )**
- **Google for references**
- **www.python.org**
- **www.scikit-learn.org**
- **www.numpy.org**