**Minor Project Report**

**Submitted in partial fulfilment of the degree of**

B.**Tech**

**Computer Science & Engineering**

**By**

**Dhiraj Agarwal (11900121081),**

**Aryan kumar Giri(11900121077),**

**Abhishek Roy(11900121081),**

**Pushan Karmakar(11900121079),**

**Arkapriya Chanda(11900121076).**

**Third year student of**

**Siliguri Institute of Technology**

A blue circle with white letters and a red dot in it

Description automatically generated

**Under the supervision of**

Mr. Kumarjeet **( Sikharthy Infotech Pvt. Ltd.)**

**Department of Computer Science & Engineering**

Date:

I hereby forward the documentation prepared by me **Dhiraj Argarwal, Aryan Kumar Giri, Abhishek Roy, Pushan Karmakar, Arkapriya Chanda** under the supervision of Mr. Kumarjeet Sir entitled “hate speech Dectection using Deep learning “accepted as fulfilment of the requirement for the Degree of Bachelor of Technology in **Computer Science & Technology** (B.Tech) from **Siliguri Institute of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**).

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    **Mr. Kumarjeet**  **(Senior Software Engineer & Project Manager)**    **Project Guide**  **Sikharthy Infotech Pvt. Ltd.** |

**Auto-Correct Feature Using NLP**

By

**Dhiraj Agarwal (11900121081),**

**Aryan kumar Giri(11900121077),**

**Abhishek Roy(11900121081),**

**Pushan Karmakar(11900121079),**

**Arkapriya Chanda(11900121076).**

UNDER THE GUIDANCE OF

**Mr. Kumarjeet**

**Project Guide**

**Sikharthy Infotech Pvt. Ltd.**

THEIS SUBMITTED IN FULFILLMENT OFTHE REQUIREMENTS FOR THE DEGREE OF

**B.C.A**

IN

COMPUTER APPLICATION

**SILIGURI INSTITUTE OF TECHNOLOGY**

**AFFILIATED TO**

**Maulana Abul Kalam Azad University of Technology**

[**Address**](https://www.google.co.in/search?rlz=1C1GTPM_enIN734IN734&biw=1366&bih=613&q=future+institute+of+technology+address&stick=H4sIAAAAAAAAAOPgE-LVT9c3NEyrMC-JTzbO0JLNTrbSz8lPTizJzM-DM6wSU1KKUouLAYQ6jCEwAAAA&sa=X&ved=0ahUKEwi3htjS0_rbAhVLQY8KHTJnC-kQ6BMI1wEwEQ)**:** Hill Cart Road, Salbari, Sukna, West Bengal 734009

[**Phone**](https://www.google.co.in/search?rlz=1C1GTPM_enIN734IN734&biw=1366&bih=613&q=future+institute+of+technology+phone&sa=X&ved=0ahUKEwi3htjS0_rbAhVLQY8KHTJnC-kQ6BMI6gEwEw)**:**  094345 27272

**Website**: <https://sittechno.org/>

**Certificate of Approval**

The foregoing project is hereby approved as a creditable study for the B. Tech in Computer Science & Engineering and presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approved any statement made, opinion express or conclusion therein but approve this project only for the purpose for which it is submitted.

Final Examination for

Evaluation of the Project ---------------------------------

--------------------------------

---------------------------------- Signatures of Examiners

**ABSTRACT**

The purpose of the project entitled as “Auto-Correct Feature Using NLP” is to develop a system which is user friendly simple, fast, and cost effective. It deals with the detection of typing errors made by users and suggests corrections to them. The main function of the system is to correct typos (Typing Errors). This makes use of various features of NLP (Natural Language Processing) and a data-set and finds out the most suitable suggestions to correct a typing error.

**ACKNOWLEDGEMENT**

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals to this attempt. Our project report has been structured under the valued suggestion, support and guidance of **Mr. Kumarjeet**. Under his guidance we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us.

**TABLE OF CONTENTS**

Chapter 1: Introduction

1: Introduction

Chapter 2: What We Used

2.1: Python

2.2: Jupyter Notebook

2.3: IDE

Chapter 3: Purpose

Chapter 4: Functionality

4.1: User

Chapter 5: System Analysis

5.1: Identification of the need

5.2: Preliminary Investigation

5.3: Feasibility Study

Chapter 6: Project Planning and Scheduling

Chapter 7: Functional Requirement of the System

**INTRODUCTION**

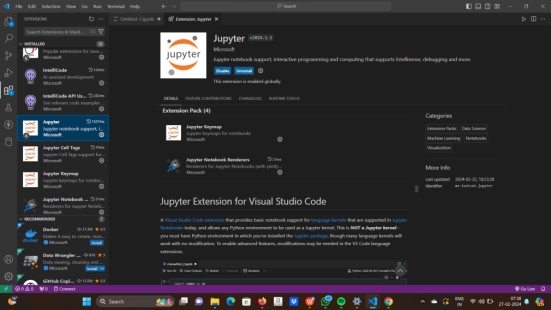
Hate speech detection is a crucial task in natural language processing (NLP) aimed at identifying and mitigating harmful language that targets individuals or groups based on attributes such as race, ethnicity, religion, gender, sexual orientation, disability, etc. Deep learning, a subfield of machine learning, has shown promising results in hate speech detection due to its ability to automatically learn complex patterns and representations from raw data.

**Python**

We used Python in this project we learned python datatypes, methods, class, sorting, OOPs concept, loops and many elements in our python program.

**2Jupyter Notebook**

We also used Jupyter Notebook in this project.



**PURPOSE**

The purpose of hate speech detection is multifaceted and extends to various domains, including social, ethical, legal, and technological aspects. Here are some key purposes:

1. Promoting Online Safety and Well-being.
2. Preventing Harm and Discrimination.
3. Ensuring Compliance with Regulations.
4. Maintaining Platform Integrity and Reputation.

**FUNCTIONALITY**

Hate speech detection systems analyze text input, preprocess it by cleaning and standardizing, extract relevant features, apply machine learning models to make predictions, and output classifications indicating the presence or absence of hate speech. These systems often incorporate thresholding mechanisms and post-processing steps to refine predictions and continuously improve performance through feedback loops, ensuring a safer and more inclusive online environment

MODULES

1. Import joblib.
2. Import pandas as pd.
3. Import tkinter as tk.
4. Import pyttsx.
5. Import spacy.
6. Import re.
7. Import numpy as np.
8. From nitk.corpus import stopwords.
9. Import matplotlib.pyplot as pit.
10. From PIL import image, ImageTk.
11. Import matplotlib.pyplot as plt
12. From sklearn.svm import svc
13. Import joblib: ’joblib’is a popular Python library used primarily for efficiently saving and loading Python objects, particularly machine learning models, to disk. It provides functionalities similar to the built-in pickle module but is optimized for storing large NumPy arrays efficiently. One common use case of joblib is in machine learning pipelines where trained models need to be saved for later use without the overhead of retraining. For example, after training a machine learning model, you can use joblib.dump() to save the model to a file and joblib.load() to reload it later. This enables seamless integration of trained models into production systems and facilitates sharing models across different environments. Here's a short example:

from sklearn.datasets import load\_iris

from sklearn.ensemble import RandomForestClassifier

import joblib

# Load dataset

iris = load\_iris()

X, y = iris.data, iris.target

# Train a model

model = RandomForestClassifier()

model.fit(X, y)

# Save the model to disk

joblib.dump(model, 'random\_forest\_model.joblib')

# Load the model from disk

loaded\_model = joblib.load('random\_forest\_model.joblib')

# Use the loaded model for prediction

prediction = loaded\_model.predict(X[[0]])

print("Predicted class:", prediction)

2) Import pandas as pd: pandas is a powerful Python library commonly used for data manipulation and analysis. It provides data structures like DataFrame and Series that allow users to work with structured and tabular data effectively. One of the main purposes of pandas is to handle data in a way that facilitates cleaning, transformation, and exploration. Users can easily load data from various file formats such as CSV, Excel, SQL databases, and more, into pandas DataFrames for analysis. pandas also offers a wide range of functions for data manipulation, including indexing, selection, grouping, merging, and reshaping operations. Here's a short example:

import pandas as pd

# Create a DataFrame from a dictionary

data = {

'Name': ['Alice', 'Bob', 'Charlie'],

'Age': [25, 30, 35],

'City': ['New York', 'Los Angeles', 'Chicago']

}

df = pd.DataFrame(data)

# Display the DataFrame

print("DataFrame:")

print(df)

# Select rows based on a condition

print("\nFiltered DataFrame:")

filtered\_df = df[df['Age'] > 28]

print(filtered\_df)

# Calculate summary statistics

print("\nSummary Statistics:")

summary\_stats = df.describe()

print(summary\_stats)

1. Import tkinter as tk: tkinter is a standard GUI (Graphical User Interface) library in Python used to create desktop applications with graphical interfaces. It provides a set of tools for building windows, buttons, labels, textboxes, and other GUI components. By importing tkinter as tk, developers can access its functionality more conveniently. An example of its usage is creating a simple window with a button:

import tkinter as tk

# Create the main application window

root = tk.Tk()

root.title("Example Window")

# Define a function to be called when the button is clicked

def button\_click():

label.config(text="Button Clicked!")

# Create a button widget

button = tk.Button(root, text="Click Me", command=button\_click)

button.pack()

# Create a label widget

label = tk.Label(root, text="")

label.pack()

# Start the main event loop

root.mainloop()

1. Import pyttsx: pyttsx is a Python library used for text-to-speech (TTS) conversion, allowing developers to generate speech output from text strings. By importing pyttsx, users can easily integrate text-to-speech functionality into their Python applications. An example demonstrates how to use pyttsx to convert a text string into speech:

import pyttsx

# Initialize the pyttsx engine

engine = pyttsx.init()

# Set properties (optional)

engine.setProperty('rate', 150) # Speed of speech (words per minute)

# Convert text to speech

text = "Hello, how are you?"

engine.say(text)

# Wait for speech to finish

engine.runAndWait()

5)Import spacy: import spacy: Spacy is a popular natural language processing (NLP) library in Python. It provides tools for various NLP tasks such as tokenization, named entity recognition, part-of-speech tagging, and more. Example:

import spacy

# Load the English language model

nlp = spacy.load("en\_core\_web\_sm")

# Process a text string

doc = nlp("This is an example sentence.")

# Print tokenized words

for token in doc:

print(token.text)

6)import re: The re module in Python provides support for regular expressions, which are powerful tools for pattern matching and text manipulation. Example:

import re

# Define a pattern to match words

pattern = r'\b\w+\b'

# Define a text string

text = "This is an example sentence."

# Find all words in the text

words = re.findall(pattern, text)

# Print the matched words

print(words)

7) import numpy as np: NumPy is a fundamental package for scientific computing in Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. Example:

import numpy as np

# Create a NumPy array

arr = np.array([1, 2, 3, 4, 5])

# Print the array

print(arr)

8) from nltk.corpus import stopwords: NLTK (Natural Language Toolkit) is a library for building Python programs to work with human language data. The stopwords module provides a list of common stopwords (words that are considered to have little value in text analysis) for various languages. Example

from nltk.corpus import stopwords

# Get English stopwords

stop\_words = stopwords.words('english')

# Print the stopwords

print(stop\_words)

9) import matplotlib.pyplot as plt: Matplotlib is a plotting library for Python that provides a MATLAB-like interface for creating a variety of plots and visualizations. Example:

import matplotlib.pyplot as plt

# Create a simple plot

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

plt.plot(x, y)

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Example Plot')

plt.show()

10) from PIL import Image, ImageTk: PIL (Python Imaging Library) is a library for opening, manipulating, and saving many different image file formats. Example:

from PIL import Image, ImageTk

# Open and display an image

image = Image.open("example\_image.jpg")

image.show()

12)from sklearn.svm import SVC: Scikit-learn is a library in Python for machine learning. SVC stands for Support Vector Classifier, which is an implementation of the Support Vector Machine (SVM) algorithm for classification tasks. Example

from sklearn.svm import SVC

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

# Load the Iris dataset

iris = load\_iris()

X, y = iris.data, iris.target

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize the SVM classifier

svm\_classifier = SVC()

# Train the classifier

svm\_classifier.fit(X\_train, y\_train)

# Evaluate the classifier

accuracy = svm\_classifier.score(X\_test, y\_test)

print("Accuracy:", accuracy)