Text-Based Depression Detection with PySS3

A Project Report

Submitted in partial fulfillment of the

requirements for the award of the degree

of

Bachelor of Technology

in

COMPUTER SCIENCE ENGINEERING

by

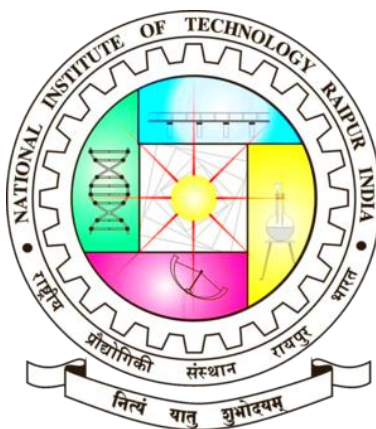**MANISH KUMAR SAHU**

Roll No. 20115055

MAYANK KURREY

Roll No. 20115057

Under Guidance of

Dr. Jitendra Kumar Rout

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, RAIPUR

RAIPUR, CHHATTISGARH

DECEMBER, 2023

DECLARATION

We hereby declare that the work described in this report, entitled **"A text classification framework for simple and effective early depression detection over social media streams"** which is being submitted by us in partial fulfillment of the award of the degree of Bachelor of Technology in **Computer Science & Engineering** to the Department of CSE, National Institute of Technology Raipur is the result of investigations carried out by us under the guidance of Dr. Jitendra Kumar Rout.

The work is original and has not been submitted for any Degree/Diploma of this or any other Institute/university.

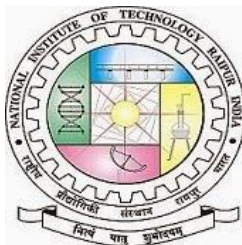<div align="right">

Manish Kumar Sahu
Roll No.: 20115055

</div>

<div align="right">

Mayank Kurrey
Roll No.: 20115057

</div>

Place: RAIPUR
Date:22-12-2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR

CERTIFICATE

This is to certify that the project entitled **"A text classification framework for simple and effective early depression detection over social media streams",** that is being submitted by Manish Kumar Sahu **(Roll No.** 20115055**),** and Mayank Kurrey **(Roll No.** 20115057**)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** to National Institute of Technology Raipur is a record of bonafide work carried out by them under my guidance and supervision.

The matter presented in this project document has not been submitted by them for the award of any other degree/diploma elsewhere.

Dr. Jitendra Kumar Rout
Assistant Professor
Department of Computer Science & Engineering
National Institute of Technology, Raipur (CG.)

Dr. K. Jairam Naik   **Dr. Pradeep Singh**

Project Coordinator   **Head of Department**

Department of C.S.E   **Department of C.S.E**

NIT Raipur (CG.)   **NIT Raipur (CG.)**

iii

ACKNOWLEDGEMENT

We would like to acknowledge our college **National Institute of Technology, Raipur** for providing a holistic environment that nurtures creativity and research-based activities.

We express our sincere thanks to Dr. Jitendra Kumar Rout**, Assistant Professor CSE Department**, **NIT Raipur,** the supervisor of the project for guiding and correcting throughout the process with attention and care. He has frequently suggested us creative ideas and guided through the major hurdles that occurred during the duration of the project.

We would also thank **Dr. Pradeep Singh, Head of the department** and all the faculty members without whom this project would be a distant reality. We also extend our heartfelt thanks to my family and friends who supported us.

Thank You!!

<div align="right">

Manish Kumar Sahu

(20115055)

Mayank Kurrey

(20115057)

</div>

ABSTRACT

The rapid expansion of Internet usage has necessitated the development of intelligent systems adept at early risk detection (ERD) on social media platforms. Among the various ERD challenges, early depression detection stands out as a critical task, given the potential impact on individuals' well-being. Existing systems, predominantly relying on machine learning techniques, confront the unique challenge of processing evolving data streams as users continually contribute information over time. Additionally, these systems must make informed decisions about when accumulated data is sufficient for user classification, all while justifying the rationale behind these decisions.

Conventional supervised machine learning models, such as Support Vector Machines (SVM), Multinomial Naive Bayes (MNB), and Neural Networks, often fall short in this context due to their black-box nature or lack of support for incremental learning. This paper introduces SS3, an innovative supervised learning model tailored for text classification, inherently addressing these challenges. SS3 serves as a versatile framework for tackling ERD problems, providing a solution that is transparent, supports incremental learning, and is well-suited for data stream scenarios. Notably, the SS3 classifier demonstrated superior performance while being computationally more efficient and possessing the ability to elucidate its decision-making rationale. This study signifies a significant stride towards developing intelligent systems capable of responsibly addressing ERD challenges, particularly in the realm of mental health monitoring on social media. The outcomes suggest the potential of SS3 as a valuable tool in building efficient, transparent, and justifiable systems for early risk detection in dynamic online environments.

Keywords:

SS3 Text Classifier, PySS3, machine learning model, text-based tasks, interactive exploration, classification model, training phase, word dictionary, word frequencies, function gv(w, c), confidence vectors, hierarchical transformation, input splitting, summary operator $\oplus 0$, confidence vector reduction, final classification, incremental classification, visual justification, interpretability, API, Live_Test class, Evaluation class, model evaluation, hyperparameter optimization, internals, word valuation, global values, classification process, hyperparameters (s, l, p), Sigma (s), Significance (l), Sanction (p), word frequency tuning, global significance capture, sanction factor determination, fine-tuning, synthesis, powerful ecosystem, accurate models, diverse applications, interpretable machine learning models, SS3 framework, dictionary creation, confidence vector generation, final classification, iterative experimentation, overfitting, underfitting, triumvirate, interpreting text, classifying text.

v

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Scope of the Work:

This work endeavours to advance the existing framework by extending and refining the SS3 text classification model. The primary goal is to create a robust solution for early risk detection (ERD), with a specific focus on early depression detection (EDD). The overarching scope and objectives can be summarized as follows:

Scope:

The project involves:

- **Enhancement of SS3 Model:** Iteratively improve the SS3 model to better facilitate incremental classification, early classification, and explainability.
- **Application to Early Depression Detection (EDD):** Apply the extended SS3 model to detect signs of depression in users' data streams, emphasizing early and accurate identification.
- **Evaluation and Comparison:** Conduct a comprehensive evaluation of the proposed framework, comparing its performance against contemporary methods used in recent EDD tasks. This evaluation will encompass quantitative assessments and efficiency analyses.

1.2 Objectives:

The specific objectives are:

- **Incremental Classification Support:** Strengthen the SS3 model to adeptly handle incremental classification of sequential data, ensuring effective adaptation to evolving patterns.
- **Early Classification Capability:** Integrate mechanisms into the SS3 model for timely decision-making on when to halt processing input data and perform accurate classifications, addressing the multi-objective challenge of balancing accuracy and timeliness.
- **Explainability and Interpretability:** Further develop the SS3 model to offer clear and interpretable explanations for its predictions, incorporating visualizations to elucidate the rationale behind decisions.

- **Application to EDD:** Implement and apply the extended SS3 model to the EDD task, leveraging insights from social media platforms for innovative measurements based on language use.

- **Comparison with State-of-the-Art:** Conduct a rigorous comparison of the proposed framework with leading methods in EDD, evaluating performance, efficiency, and unique contributions.

- **Quantitative and Qualitative Analysis:** Analyze both quantitative and qualitative aspects of the framework's performance, considering accuracy, efficiency, and interpretability, supported by concrete evidence

.

1.3 The SS3 Framework

In addressing the challenges of Early Risk Detection (ERD), specifically in the domain of Early Depression Detection (EDD), the SS3 framework emerges as a novel and integrated solution. Recognizing the imperative requirements of incremental classification, support for early classification, and explainability, the SS3 framework strives to seamlessly incorporate these facets into a unified text classification model.

CHAPTER 2

Literature Review & Methodology

Textual content was typically extracted from social media platforms that prohibited redistribution. Recognizing the significance of publicly available datasets for fostering research, Losada and Crestani aimed to address this gap by providing the first public collection for studying the relationship between depression and language usage. Prior to the work of [1], efforts to predict or analyze depression using machine learning lacked a publicly available dataset with a substantial chronological collection of writings leading to the disorder.

In terms of classification models, the spectrum ranged from standard classifiers to sophisticated methods like Recurrent Neural Networks (RNNs) and graph-based models. Additionally, diverse mechanisms were employed to decide when to make predictions, including fixed threshold policies and extra conditions based on the number of writings.

However, none of the contributions, except those based on RNN and Multinomial Naive Bayes (MNB), were deemed suitable for processing data sequences naturally. Standard classifiers lacked this capability, emphasizing the need for models capable of handling sequential data. Moreover, explainability was neglected across all contributions, even in RNN and MNB-based models, highlighting a crucial aspect for applications involving real people.

This paper delves into our endeavours to construct a framework for detecting depression using textual features extracted from social media posts. Unlike prevalent practices in depression detection research, where custom datasets are created but not shared publicly, our study utilizes openly accessible dataset.

While our main goal is to create a method to detect depression on social media, it is important to remember that the data only includes messages and to delete emoticons, emoticons, emoticons and other symbols used in the conversation. and web links. We also address the issue of overfitting, a common problem when collecting data on depression from social media.. Overfitting can lead to poor performance on datasets not included in the training set. Additionally, we concentrate on mitigating imbalanced data samples, a concern that can detrimentally affect the performance of classifier models. Let us have an overview of related works that were in past few years

Table 2.1 - Prior works in the field of early depression detection based on various types of datasets such as textual, audio and visual.

Author	Dataset	Published On	Type of Dataset	Models Used	Outcome
G. Shen [2]	Twitter	2017	Descriptive	MDL3, MSNL3, WDL3, NB	Accuracy: 85%, Precision: 85%, Recall: 85%, F1: 85%
J. Kim, J. Lee [3]	Reddit	2020	Textual	CNN, XGBoost	Accuracy: 75.13%, Precision: 89.1%, Recall: 71.75%, F1: 79.49%
Hassan [4]	Twitter	2017	Textual	SVM, NB, ME	Accuracy: 91%, Precision: 83%, Recall: 79%
Chen [5]	WeChat	2018	Textual	LSTM	Results presented through various graphs
Burdisso [6]	Reddit	2019	Textual	SS3, KNN, LR, SVM, NB	Precision: 63%, Recall: 60%, F1: 61%
Fatima [7]	Reddit	2019	Textual	MLP, SVM, LR	Accuracy: 91.63%, Precision: 91.83%, Recall: 91.85%
Alsagri [8]	Twitter	2020	Textual	SVM, NB, DT	Accuracy: 82.5%, Precision: 73.91%, Recall: 85%, F1: 79.06%, AUC: 0.78
C. Lin [9]	Twitter	2020	Visual & Textual	CNN	Accuracy: 88.4%, Precision: 90.3%, Recall: 87%, F1: 93.6%

Methodology

The methodology employed in this research endeavours to holistically address the complexities inherent in early depression detection (EDD) through the extension and enhancement of the SS3 text classification model. This section provides a detailed overview of the steps taken to refine the model, the strategies implemented for application to EDD, and the comprehensive evaluation framework established.

2.1.1 SS3 Model Enhancement: The first facet of our methodology centers on augmenting the SS3 text classification model. The goal is to fortify the model's capabilities, specifically focusing on

incremental classification support, early classification decision-making, and improved explainability. Incremental classification is crucial for handling sequential data effectively, necessitating the model to adapt seamlessly to evolving patterns in data streams. Early classification, a multi-objective challenge, involves striking a balance between accuracy and timeliness, ensuring timely decisions on when to halt processing input data for classification. The model's explainability is also paramount, requiring clear and interpretable rationales for predictions, facilitated through visualizations.

2.1.2 Application to EDD: The extended SS3 model is then applied to the domain of early depression detection. Using information from social media sites, like Facebook and Twitter, the algorithm looks for possible indicators of sadness in user data streams. The idea that "language reveals who we are: our thoughts, feelings, beliefs, behaviours, and personalities" (Schwartz & Ungar, 2015) is supported by the emphasis on using language use as a unique assessment. The model contributes to the field of automatic depression detection (ADD) by examining user-written content to find language patterns suggestive of depression.

2.1.3 Evaluation and Comparison: A critical part of our approach is a thorough assessment of the suggested framework. We evaluate the model's performance using a wide range of indicators, taking into account its accuracy, effectiveness, and distinctive contributions within the EDD context. The assessment also includes a comparison with the most advanced techniques applied to current EDD projects. The purpose of this comparative study is to shed light on the relative merits and demerits of the suggested framework and present a balanced assessment of its effectiveness when compared to more modern methods.

2.1.4 Quantitative and Qualitative Analysis: Both quantitative and qualitative analyses are incorporated into our methodology in order to provide a thorough assessment of the model's performance. We evaluate accuracy metrics and efficiency measures quantitatively to give a quantitative picture of the computational efficiency and predictive power of the model. We investigate the interpretability of the model's choices qualitatively, looking at how well it conveys the reasoning behind its predictions. In this qualitative analysis, explanations and visualizations are essential for gaining a deeper comprehension of the inner workings of the model.

The present study's methodology is indicative of a methodical and comprehensive approach towards the advancement of the SS3 text classification model for EDD. Our approach combines thorough

evaluation, practical application, and model refinement in order to provide significant contributions to the field of early risk detection and automatic. Through this holistic approach, we aspire to not only improve predictive performance but also enhance the transparency and interpretability of the model, aligning with the evolving demands of AI applications in critical domains.

2.2 Data Collection Methodology: An In-Depth Exploration

The foundation of any machine learning project lies in the quality and representativeness of the data used for training and evaluation. In this project, the data collection methodology was a meticulous process, aimed at ensuring a diverse and comprehensive dataset for training the SS3 model to detect early signs of depression in social media comments.

2.2.1. Dataset Source and Overview: The dataset for this project was obtained from Kaggle, a popular online platform for datasets. The dataset, stored in an Excel file, comprised two crucial columns: one containing comment extracted from Twitter users, and the other indicating the sentiment polarity (positive or negative) associated with each comment.

The majority of the text and comments in our dataset come from two of the most popular Twitter and Reddit are social media sites. These platforms are renowned for having large user bases interaction and a variety of material. See the table below for a detailed description of the dataset and its attributes, which provides an overview.

Datasets

Table 2.2 - Different source of Datasets used for the proposed work.

Source	Description	Dimension of the Dataset	
		Number of rows	Number of Columns
Reddit	Textual comments made by Reddit users	15464	2
Twitter	and textual posts shared on Twitter.	116920	2

List of Dataset Columns

Table 2.3 - Describes dimension of the dataset used in the project.

Column	Column Description	Data Type
Comment	Comments or textual posts from the user	String
Label	Whether the comment is positive is negative	String

2.2.2 Data Extraction and Kaggle Utilization: To extract and modify the dataset from the Kaggle platform, a script was written in Python. Obtaining the dataset in Excel format was the first step in the extraction process, giving a preliminary picture of the raw data environment.

2.2.3 Data Preprocessing: The dataset was acquired and then put through a number of preprocessing steps to make sure it was suitable for training machine learning models. In order to handle missing values, duplicates, and anomalies that might negatively impact the model's performance, the data had to be cleaned.

2.2.4 Text File Generation: Every comment in the dataset was separated and kept in a separate text file to aid in the SS3 model's training. Each row in the Excel file was processed iteratively by a Python script, which produced a corresponding text file with the user comment's text in it. After taking this step, a number of text files were produced, each of which captured the ideas mentioned in the related Twitter comment.

2.2.5 Class Labelling - Suicidal and Non-Suicidal Segregation: Suicidal and non-suicidal comments were added to the dataset due to the delicate nature of tasks pertaining to mental health. We used a binary classification to separate comments that suggested suicidal thoughts from those that did not. This segmentation involved a careful examination of the content of the comments and was carried out according to a predetermined set of criteria.

2.2.6 Train-Test Split: The dataset must be split into training and testing sets in order to create a reliable model. A split ratio of 80:20 was utilized, wherein 80% of the data was set aside for training and the remaining 20% for testing and assessing the model's efficacy. This division ensures that the model is trained on a sufficiently large and diverse dataset while retaining an independent subset for unbiased evaluation.

2.2.7 Ethical Considerations: Handling a dataset related to mental health requires a heightened sense of ethical responsibility. Rigorous ethical reviews were conducted to address potential concerns related to privacy, consent, and the responsible use of sensitive information. The project adheres to ethical standards to ensure the well-being and privacy of the individuals whose data contributes to the training of the model.

2.2.8 Python Libraries and Tools: The data collection and preprocessing tasks heavily relied on Python and associated libraries. Pandas was employed for efficient data manipulation, and open-source libraries such as NumPy and scikit-learn were utilized for numerical operations and machine learning tasks, respectively.

2.3 Model Training Strategies for Early Depression Detection

In the pursuit of building an effective early depression detection model, the training phase plays a pivotal role in shaping the capabilities and performance of the classifier. Leveraging the PySS3 library, two distinct training approaches were employed, each designed to explore different aspects of the SS3 classifier's potential.

2.3.1. **Basic Model Training:** In the initial phase of model training, a fundamental approach was taken, where the SS3 classifier was trained without incorporating variable-length word n-grams. This served as a baseline model, capturing the essence of the classifier's performance when not explicitly considering intricate word combinations. The simplicity of this approach provides insights into the classifier's innate ability to discern sentiment without delving into the complexities introduced by n-grams.

2.3.2. Intelligent Model Training with N-grams: Recognizing the importance of contextual understanding and the significance of variable-length word n-grams in text classification, an advanced training strategy was explored. In this approach, the SS3 classifier was trained to dynamically recognize and adapt to variable-length word n-grams "on the fly." This was achieved by incorporating the n_grams=3 parameter during the training phase. The aim was to enhance the model's intelligence, enabling it to grasp nuanced patterns and relationships within sequences of words, a crucial aspect in the context of social media comments.

Training Process Overview: Regardless of the approach, the training process followed a systematic sequence of steps facilitated by the PySS3 library:

a. **Initialization:**

- A new instance of the SS3 classifier was created, laying the foundation for subsequent training iterations.
- Hyperparameters such as smoothness (s), significance (l), and sanction (p) were initialized, providing a starting point for the training process.

b. **Dataset Loading:**

- Training and test sets were loaded from the preprocessed and segmented dataset. The comments, now in the form of text files, served as the input features, while their corresponding labels (indicating suicidal or non-suicidal sentiments) formed the ground truth for training.

c. **Model Training:**

- The training process commenced with the SS3 classifier learning from the labeled training set. The model iteratively adjusted its parameters to optimize its ability to classify social media comments based on the provided sentiment labels.

d. **Performance Evaluation:**

While our main goal is to create a method to detect social pressure, it is important to remember that the data only includes words and omits phrases, emoticons, emoticons, and other symbols used in conversations. and web links. We also addressed overfitting, a common problem when collecting depression data from social media.

Model Comparison and Analysis: The training approaches paved the way for a comparative analysis of the basic and intelligent models. Metrics such as accuracy, precision, recall, and f1-score provided a nuanced understanding of each model's strengths and weaknesses in the context of early depression detection.

Advantages of Intelligent Model Training: Training the SS3 classifier to recognize variable-length word n-grams introduced a layer of sophistication to the model. The ability to grasp contextual nuances in social media comments enhanced the classifier's accuracy, especially in scenarios where word sequences hold crucial significance. The intelligent model showcased improved performance in capturing intricate patterns and relationships within the data.

Considerations for Hyperparameter Optimization: In the pursuit of refining model performance, hyperparameter optimization was explored. Adjusting hyperparameter values, such as smoothness (s), significance (l), and sanction (p), provided an avenue for fine-tuning the model. The Evaluation.grid_search() function was employed to systematically search for the optimal hyperparameter values, considering different metrics and targets.

CHAPTER 3

MODEL WORKING

3.1 The SS3 Text Classifier and PySS3: A Comprehensive Exploration

Classification of text continues with the introduction of the SS3 Text Classifier, an elegant and impressive machine learning model designed for text-based tasks.The accompanying PySS3 Python package serves as a powerful toolset, facilitating the interactive and visual exploration of the SS3 classification model. This comprehensive overview delves into the functionality, components, and applications of the SS3 classifier and PySS3 package.

1. **Dictionary Building:**
   - During the training phase, SS3 builds a dictionary of words for each class.
   - Frequency of each word is stored in the dictionary.

2. **Word Valuation:**
   - SS3 uses a function $gv(w,c)$ during the classification phase to calculate the value of each word for a specific class.
   - This "global value" (gv) is a confidence measure, ranging from 0 to 1, indicating the word's association with a class.

3. **Confidence Vectors $(\overrightarrow{gv})$:**
   - For each word, SS3 generates a confidence vector $(\overrightarrow{gv})$ representing its values for all categories.
   - For example,

$$\overrightarrow{gv}(cricket) = (0.7, 0.4, 0.8, 0)$$

4. **Hierarchy Transformation:**
   - The classification process involves a 2-phase hierarchy transformation.
   - Input is initially split into blocks (e.g., paragraphs), then further divided into smaller units (e.g., sentences, words).

5. **Confidence Vector Reduction:**
   - $\overrightarrow{gv}$ is applied to each word, producing "level-0" confidence vectors.
   - These are then reduced to "level-1" confidence vectors using a summary operator $\oplus 0$.
   - Reduction is recursive, moving up to higher-level blocks using $\oplus j$ operators.

6. **Final Classification:**
   - The recursive reduction process culminates in a single confidence vector $(\vec{d})$ for the entire input.
   - **Classification** is **done according to** the values **of** this vector.
   - A policy, such as choosing the type with the chief confidence value, is applied.

7. **Incremental Classification:**
   - The classification process is incremental, especially for the highest level block, if the summary operator allows incremental computation.
   - For example, if a new sentence is added, only the affected confidence vectors need updating, enabling efficient real-time classification.

8. **Visual Justification:**
   - The hierarchy of confidence vectors allows for visual justification of the classification, with different blocks coloured according to their values.
   - This transparency aids interpretation and understanding of SS3's decision-making process.

3.1.1. SS3 Classifier: Unveiling Interpretability

The SS3 classifier distinguishes itself by offering interpretability alongside classification accuracy. Interpretability, a crucial aspect in real-world applications, ensures that the model can articulate the rationale behind its decisions. Leveraging a clear API reminiscent of sklearn, the SS3 class provides a seamless interface for training and prediction.

This simplicity in usage conceals a sophisticated mechanism within the SS3 classifier. During training, the model constructs a word dictionary for each class, storing word frequencies. Subsequently, during the classification phase, it calculates word values using a function, paving the way for interpretability.

3.2. Interactive Toolkit for Model Development: PySS3 extends the usability of the SS3 classifier by providing a rich toolkit for model development. Comprising three main components - the SS3 class, the Live_Test class, and the Evaluation class - PySS3 enhances the development, analysis, and monitoring of machine learning models.

3.2.1 The SS3 Class

The core of PySS3, the SS3 class, encapsulates the SS3 classifier. Its familiar API simplifies model training and prediction, ensuring a smooth integration into the Python machine learning ecosystem.

3.2.2 The Live_Test Class

Enabling interactive testing and visual interpretation, the Live_Test class elevates model exploration. With a single line of code, developers can interactively test the model and visualize the reasons behind classification decisions.

Python Code

```
from pyss3.server import Live_Test
from pyss3 import SS3
# Initializing the SS3 classifier
clf = SS3()
# Training the classifier
clf.fit(x_train, y_train)
# Running live testing for visualization
Live_Test.run(clf, x_test, y_test)
```

3.2.3 The Evaluation Class: Arguably one of PySS3's most potent components, the Evaluation class streamlines model evaluation and hyperparameter optimization. Offering methods such as test(), kfold_cross_validation(), grid_search(), and plot(), it empowers developers to assess, refine, and visualize model performance systematically.

The Evaluation class not only facilitates evaluation but also maintains a historical record of evaluations, enabling users to interactively visualize and analyse classifier performance concerning different hyperparameter values.

3.3. Model Internals and Hyperparameters: Decoding the SS3 Mechanism

Understanding the internals of the SS3 classifier is paramount for effective utilization. The classification process involves the creation of confidence vectors based on word frequencies, and the incremental nature of classification is emphasized.

3.3.1 Word Valuation and Confidence Vectors

During training, the SS3 classifier creates a word dictionary for each class and stores the word frequency. The valuation word expressed by the word global value (gv) is calculated at the time of distribution. This gv is the basis of the belief vector and provides insight into the decision model.

The gv of a word in a class is determined by its local value, significance, and sanction, each controlled by corresponding hyperparameters (s, l, and p).

3.3.2 Classification Process

Illustrated in the classification process, the SS3 algorithm operates in two phases. The document undergoes hierarchical splitting into blocks, followed by the computation of confidence vectors for each word. These vectors are then successively reduced to generate confidence vectors for higher-level blocks until a single confidence vector for the entire input is obtained.

Python Code

```
# Example of incremental classification
clf.update(x_new_sentence)
```

Incremental classification, a distinctive feature, allows for updating the classification by processing only new data, showcasing the efficiency of the SS3 classifier.

3.3.3 Hyperparameters: Tuning the SS3

The hyperparameters (s, l, p) play a pivotal role in shaping the SS3 classifier's behavior. Sigma (s), significance (l), and sanction (p) influence word frequency tuning, global significance capture, and sanction factor determination, respectively. Fine-tuning these hyperparameters is essential for optimizing the model's performance.

In the synthesis of the SS3 text classifier and the PySS3 toolkit, a powerful ecosystem emerges for text classification tasks. The SS3 classifier's interpretability, coupled with PySS3's interactive tools, empowers developers to not only build accurate models but also understand, analyze, and optimize them. This exploration lays the foundation for leveraging the SS3 and PySS3 combination in diverse applications, from sentiment analysis to early depression detection, contributing to the broader landscape of interpretable and effective machine learning models.

CHAPTER 4

SS3 Framework

During training, SS3 first creates a dictionary that stores the frequency of each word for each class. It then calculates the value of each word using the **gv (w,c)** function to evaluate the words associated with each group during classification using word frequency (**gv** stands for "a word global value"). **gv** takes the term w and the class c and produces a number in the range [0,1] on behalf of the confidence that w is considered specific to c.

For example, the groups are insects, politics, sports, and automobiles. After training, SS3 will absorb to evaluate words such as "cricket" and "for" as follows:

$$gv\ (cricket, insects) = 0.7; \quad gv\ (for, insects) = 0.1;$$
$$gv\ (cricket, politics) = 0.4; \quad gv\ (for, politics) = 0.1;$$
$$gv\ (cricket, sports) = 0.9; \quad gv\ (for, sports) = 0.1;$$
$$gv\ (cricket, automobile) = 0; \quad gv\ (for, automobile) = 0.1;$$

So $\overrightarrow{gv}$ is the vector version of gv. $\overrightarrow{gv}$ used only for terms, outputs a vector where each constituent is the gv of words in each class. For example, following the instance above we get:

$$\overrightarrow{gv}(cricket) = (0.7, 0.4, 0.8, 0)$$
$$\overrightarrow{gv}(for) = (0.1, 0.1, 0.1, 0.1)$$

These vectors are termed confidence vectors (cv). Therefore, in this instance (0.7,0.4,0.8,0) is the confidence vector for the term "cricket"; where the opening location corresponds to "insects", the second position corresponds to "politics" and So. etc.

4.1 Classification

We can think of the classification process as a two-stage process, as seen in the diagram below.
In the first stage, ideas are divided into parts (e.g. sentences), and then each part is divided again into smaller units (e.g. sentences, words). Therefore, the previous "engine" form is converted into a block hierarchy.

In the second step, the $\overrightarrow{gv}$ method is functional to each term to get a "level-0" confidence vector, which is then abridged from level-0 to the "level-1" confidence vector. This minimization process is repeated for advanced stages using higher order operators $\oplus j$ until a confidence vector $\vec{d}$ is created for all inputs.

Lastly, depending on the value of the confidence vector $\vec{d}$ the actual classification is done by some strategy - for example, selecting the classes with the highest value.
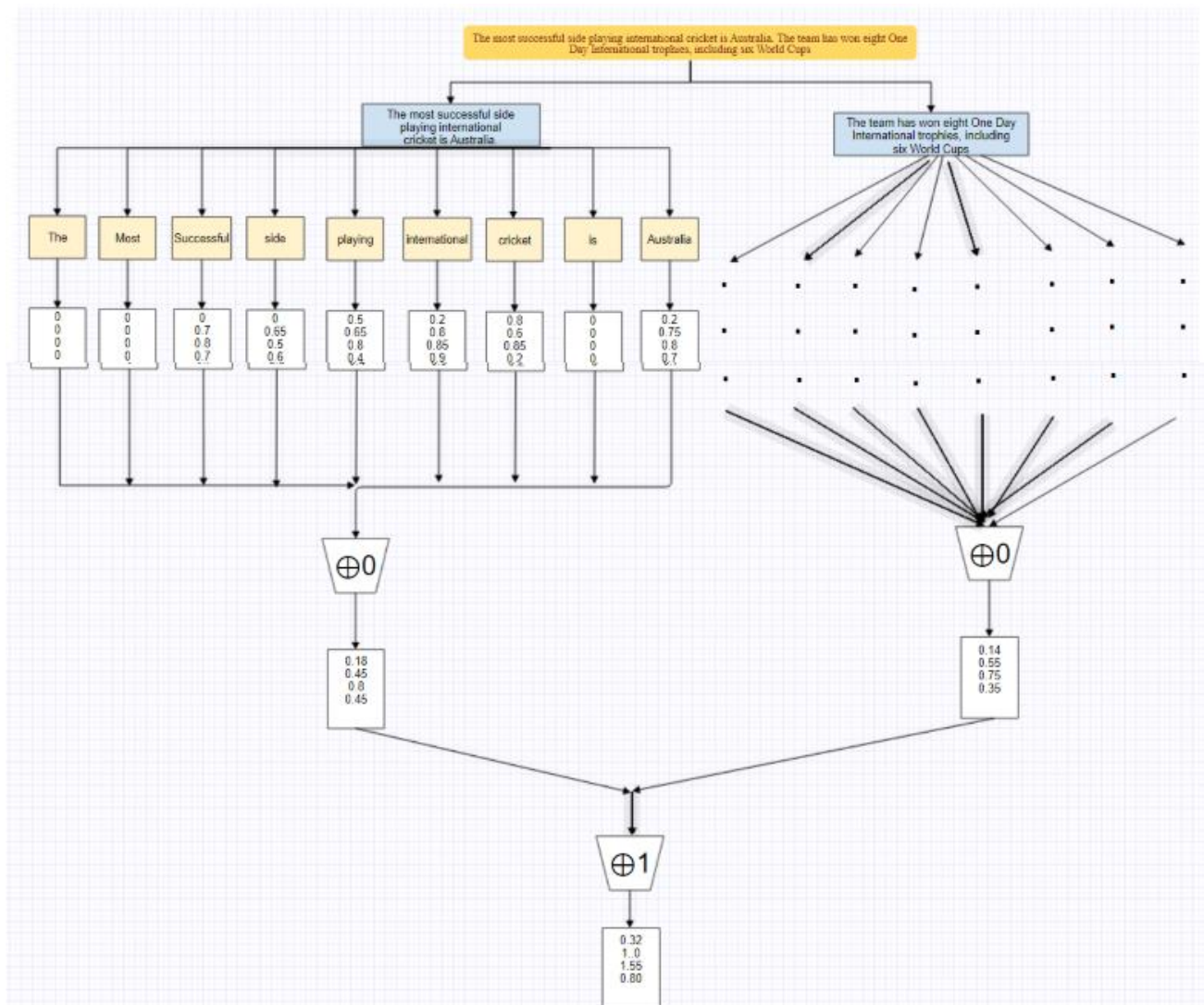

Figure 4.1 - Image describing splitting of large paragraphs and textual data into sentences. These sentences are further broken up into word and gv is calculated for each of the words. Then the gv vectors are combined to get a single gv vector.

Also, blocks can be counted in increments ($\oplus_1$ in this case) as long as the addition operator is at the highest level - this allows for addition, multiplication, max, etc. Valid for most aggregate functions such as. For example, let's say a new sentence is added to the example above. Since $\oplus_1$ is the sum of all vectors, instead of redoing all the data, we can update the vector numbers (0,15,) by adding a new confidence vector expression to it - note that this part is continuous and only new sentences are included. is required. When done, the process will produce the same results each time when applied again to all data.

4.2 Hyperparameters

As mentioned in the previous section, the word global value (gv) is used to first create the confidence vector and then increase the confidence to more vectors until the value of all ideas. Get the trust vector. Therefore, the global value (gv) of a word is the building block of the entire classification process.

Simply put, the global value (gv) of a word in a class is calculated by adding three values as follows. such as its local value, importance and decision. Moreover, in practice, the calculation of each of these three values is governed by a specific hyperparameter. In more detail we have:

$$global\ value = local\ value \cdot significance \cdot sanction$$
$$gv = s \cdot l \cdot p$$

4.3 A Deep Dive into Sigma (s), Significance (l), and Sanction (p)

Introduction: In the intricate realm of text classification, the SS3 text classifier stands out not only for its interpretability but also for the nuanced control it provides through hyperparameters. The trio of Sigma (s), Significance (l), and Sanction (p) plays a pivotal role in shaping the behavior of the SS3 model. This comprehensive exploration aims to unravel the intricacies of each hyperparameter, shedding light on their functions, implications, and the art of fine-tuning.

4.3.1. Sigma (s): Navigating the Frequency Spectrum Sigma, denoted by 's,' emerges as a frequency tuner, offering a mechanism to balance the impact of raw word frequency on class assignment. In essence, Sigma acts as a smoothing hyperparameter, influencing the association among fresh occurrence and the concluding rate allocated to a term.

4.3.1.1 Smoothing the Frequency Landscape: The primary role of Sigma lies in introducing a level of smoothness to the raw frequency dynamics. A higher Sigma value, such as 0.8, indicates a direct and

proportional calculation of the local value concerning raw frequency. In contrast, smaller Sigma values, like 0.2, exert a dampening effect, decreasing the influence of fresh occurrence on the concluding cost assigned to a term.

4.3.1.2 Fine-Tuning with Sigma: Selecting an appropriate Sigma value involves a delicate balance. A high Sigma might lead to overemphasis on raw frequency, potentially resulting in skewed models. On the other hand, too low a Sigma might blur the impact of frequency, affecting the model's ability to discern the relevance of words to specific categories.

4.3.2. Significance (l): Decoding Global Importance Moving beyond the local dynamics captured by Sigma, the Significance hyperparameter, denoted by 'l,' delves into the universal worth of a term. It achieves this by curbing the universal value concerning the local value in other categories.

4.3.2.1 Global Significance Unveiled: Significance is a metric of how far the local value of a word in a given group essentially diverge from its local value in supplementary classes to be deemed significant. In simpler terms, it gauges the uniqueness of a word's importance to a particular class in comparison to others.

4.3.2.2 Empowering Control with Significance: The 'l' parameter, Significance, puts control in the hands of the model developer. By adjusting 'l,' one can influence the threshold at which a word becomes significant to a class. A higher 'l' demands a more pronounced deviation, raising the bar for a word to be considered important.

4.3.3. Sanction (p): Balancing Global Influence Sanction, represented by 'p,' introduces a factor of restraint to the global value of a word. This limit applies to the number of groups in which a word is considered important based on its importance.

4.3.3.1 Enforcing Balance with Sanction: Sanction, in its essence, balances the global influence of a word by considering its significance across categories. If a word holds significance in multiple categories, Sanction curtails its global impact, preventing undue dominance.

4.3.3.2 Navigating Complexity with Sanction: The 'p' parameter in Sanction opens avenues for navigating the intricate web of word significance. A higher 'p' implies a stricter penalty for words significant across numerous categories, maintaining equilibrium in the distribution of global influence.

4.4. Holistic Integration: Creating an SS3 Model The culmination of Sigma, Significance, and Sanction in the form of an SS3 model involves a thoughtful synthesis of these hyperparameters. As an example, the instantiation of an SS3 object with specific hyperparameter values is achieved as follows:

This instantiation sets the stage for a model attuned to the nuances of the dataset, balancing frequency, global importance, and the impact of multi-class significance.

4.5. The Art of Fine-Tuning: Crafting an Optimal SS3 Model
The journey through Sigma, Significance, and Sanction is incomplete without a discussion on the delicate art of fine-tuning. Achieving an optimal SS3 model involves an iterative process of experimentation and analysis, where developers navigate the hyperparameter space to align the model with the nuances of their data.

5.1 Iterative Experimentation: Fine-tuning necessitates a nuanced understanding of the dataset and the underlying dynamics of word significance. Developers embark on an iterative journey, adjusting Sigma, Significance, and Sanction values, observing their model's response, and refining the hyperparameters.

5.2 Striking a Balance: The crux of fine-tuning lies in striking the delicate balance between overfitting and underfitting. Rigorous experimentation guides developers in identifying the sweet spot where the model adapts to the training data without losing its generalization capabilities.

4.6. Harnessing the Power of Hyperparameters
In the realm of text classification, the triumvirate of Sigma, Significance, and Sanction emerges as the guiding force behind the SS3 model. Sigma navigates the frequency spectrum, Significance decodes global importance, and Sanction balances global influence. As developers wield these hyperparameters, they embark on a journey of fine-tuning, sculpting the SS3 model into a powerful tool for interpreting and classifying text. The fusion of theoretical understanding and practical experimentation creates a synergy that propels the SS3 text classifier into the forefront of interpretable and effective machine learning models.

CHAPTER 5

VISUALIZATION AND OPTIMIZATION

Introduction: In the constantly changing field of machine learning, knowing what models learn and it's crucial to understand why they take certain actions. Two interactive visualization tools—Live Test and Interactive 3D Evaluation Plot—are offered by the robust Python package PySS3. explore the depths of their models, providing information on how they learn and facilitating efficient model monitoring and analysis.

5.1. Live Test: A Window into Model Understanding: An Insight into Model Interpretation The Live Test tool provides an interactive interface that lets you see inside your models' learning processes. This tool can be launched by users with a PySS3 command or a single line of Python code, which will start an intuitive interface for manual testing. The Live Test tool is displayed in Figure 1, showcasing its adaptability in analyzing models at various levels, ranging from word n-grams to sentences and paragraphs.
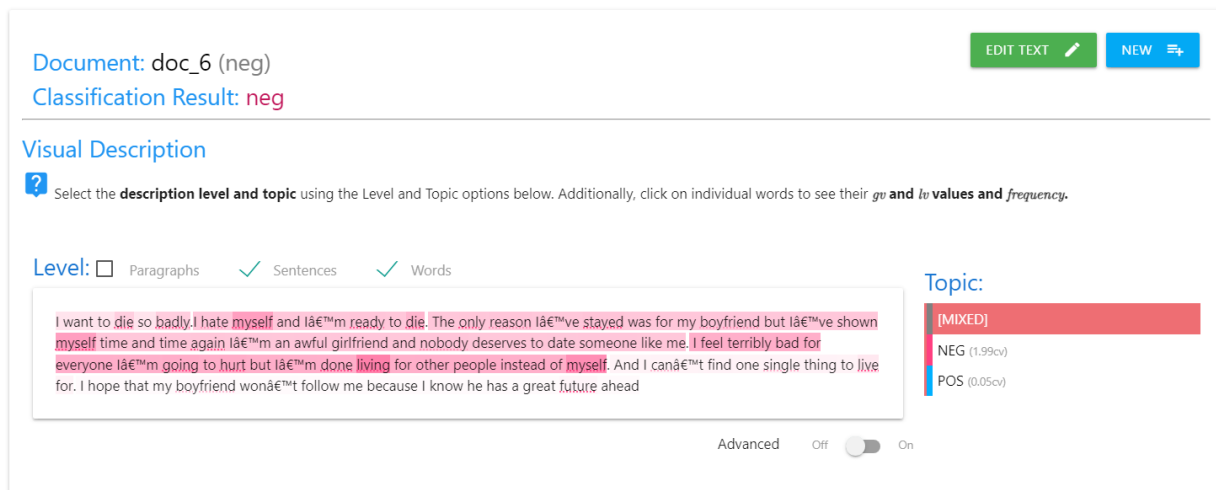

Figure 5.1 - An example from the live test server of the working model. A negative textual data has been provided as a sample. Based on the gv of words, the clarification result came to be neg, thus detecting the emotion as depressed.

5.1.1 Real-Time Model Testing: The Live Test tool lets users test their models in real-time by giving them the choice to enter custom documents for analysis in real-time or use a preloaded set of documents (like the x-test list). During the testing phase, this interactive approach enables a

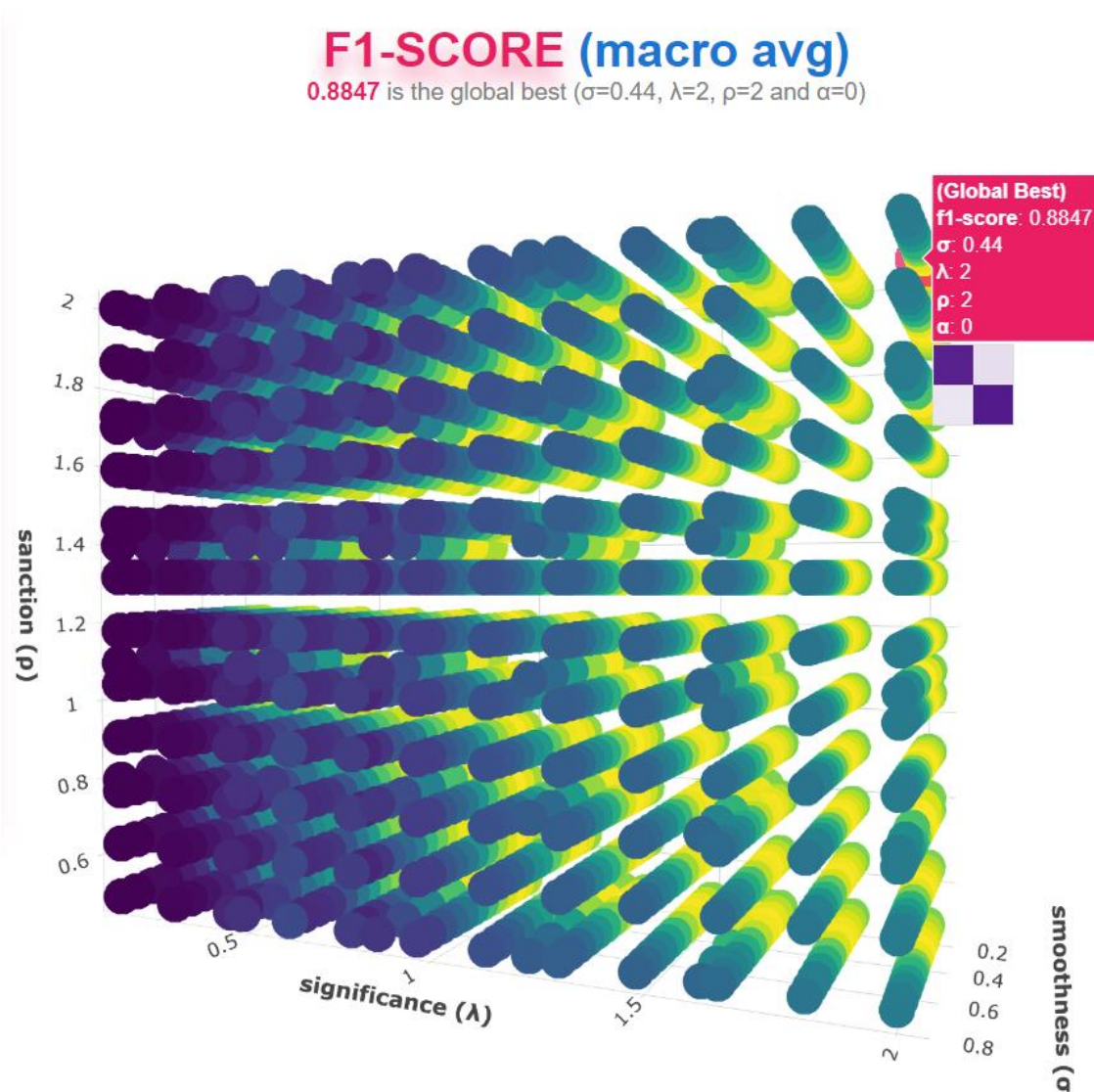detailed analysis of what the model is actually learning.

F1-SCORE (macro avg)
0.8847 is the global best (σ=0.44, λ=2, ρ=2 and α=0)

Figure 5.2 - Using the evaluation plot to get the values of the three hyperparameters. Choosing the optimal parameters could enhance the F1-Score.

5.1.2 Visualizing Learning Dynamics: Live Test makes it easier to explore learned patterns at different levels by providing a visual depiction of the model's decision-making process. Users can learn more about the meaning of word n-grams, how to understand sentences, and how to understand entire paragraphs. This instantaneous feedback loop improves the user's comprehension of the behaviour of the model.

5.2. Interactive 3D Evaluation Plot: Unveiling Performance Patterns: A thorough tool for model evaluation and hyperparameter optimization is the Interactive 3D Evaluation Plot. The

Evaluation.plot() function from the pyss3.util Evaluation class or certain commands from the PySS3 Command Line tools can be used by users to automatically create this plot.
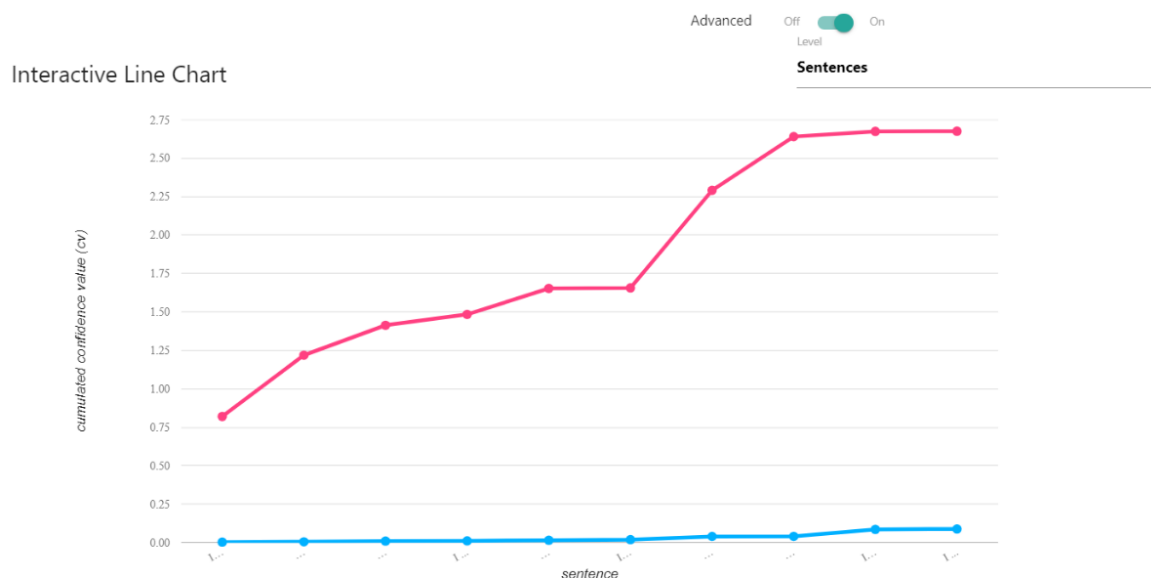

Figure 5.3 - Using the interactive line chart, to see how confidence value is calculated for each term.

5.2.1 Data Representation and Colouring: The plot displays tests and analyses carried out using different sets of hyperparameter values (s, l, and p). Every data point on the plot represents a distinct evaluation, and the performance level is indicated by the coloration, which was produced using the viridis colormap. Notably, the pink highlights on the points with the best overall performance offer a quick reference for ideal configurations.
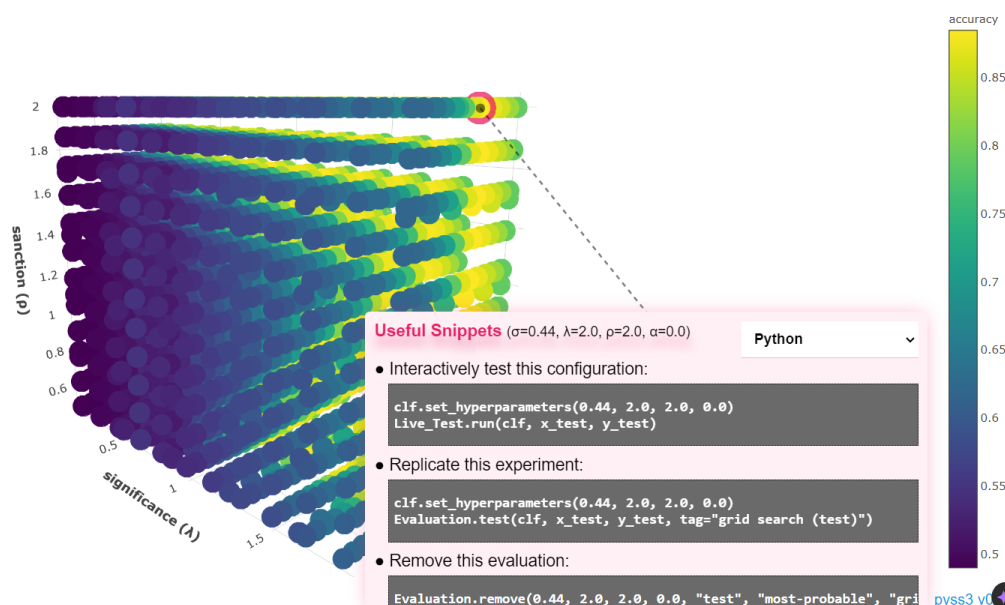

Figure 5.4 - An image from the evaluation plot. On clicking the optimal point in the hyperparameter plot we get the values of the hyperparameters at that point.

5.2.2 Dynamic Performance Measure Selection: Using the options panel, users can dynamically alter the performance measure on the interactive plot. This flexibility makes it possible for users to customize the assessment procedure according to particular standards, which improves the applicability of the knowledge acquired.
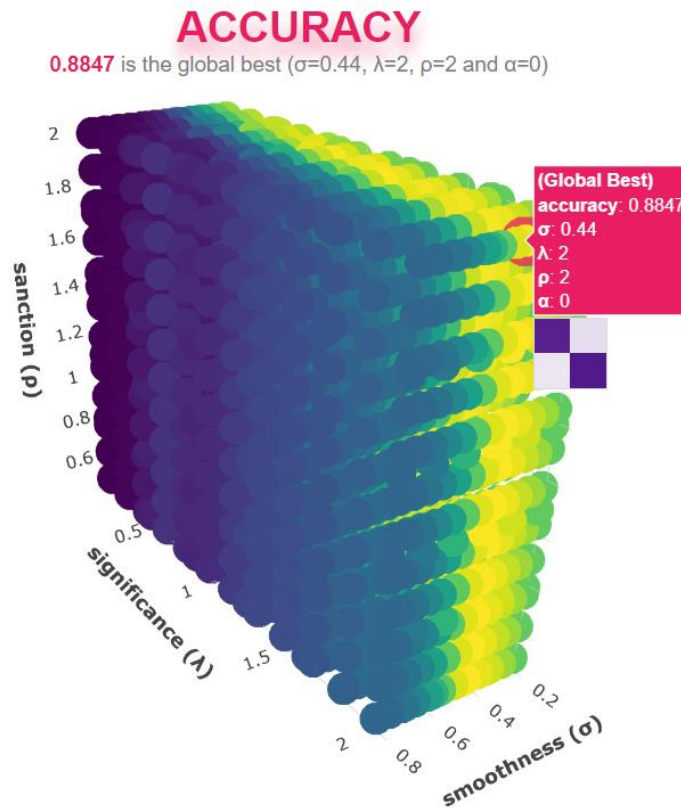

Figure 5.5 - Shown in diagram, plot having the hyperparameters in each of the axes. The point encircled in pink denotes the point with highest accuracy.

5.2.3 Comprehensive Details on Data Points: When you hover over a data point, specific evaluation-related details are revealed, including a condensed version of the confusion matrix. This granular view aids in understanding the nuances of model performance in different scenarios.

5.2.4 K-Fold Cross-Validation Insights: For evaluations involving k-fold cross-validation, the plot displays the confusion matrix obtained in each fold. This additional layer of information offers insights into the model's consistency across different folds, contributing to a more comprehensive understanding of its robustness.

5.2.5 Actionable Insights and Replication: Clicking on a data point provides a list of useful snippets, offering the ability to replicate the experiment, remove the evaluation point from the cache, or test that specific configuration using the Live Test tool. These actionable insights streamline the decision-making process for users.

5.2.6 Customization for Enhanced Analysis: The options panel provides users with the flexibility to hide or show different parts of the plot. For instance, by using the "show volume" option, users can focus solely on understanding the performance pattern concerning the three hyperparameters—s, l, and p. This customization enriches the analytical capabilities of the tool.

In the realm of machine learning model development and analysis, PySS3 stands out by offering interactive visualization tools that empower users to understand their models comprehensively. The Live Test tool offers real-time insights into the learning dynamics, while the Interactive 3D Evaluation Plot unveils performance patterns and facilitates hyperparameter optimization. Together, these tools provide a robust framework for model analysis, enabling users to make informed decisions, enhance model interpretability, and refine their machine learning workflows.

CHAPTER 6
RESULTS AND COMPARISON

The results of applying PySS3, a novel supervised learning model, to early risk detection tasks—with a particular focus on early depression detection—are examined in detail in the results and discussion section.

The application, testing, and assessment offer important insights into PySS3's efficacy relative to industry standards and cutting-edge models. This section also covers the results' implications, PySS3's interpretability, and prospective directions for further study.

6.1. Experimental Setup: It's important to comprehend the experimental setup before diving into the results. The dataset, obtained from Kaggle, included user-annotated positive and negative Twitter comments, signifying suicidal and non-suicidal sentiments. These comments were converted into separate text files by the Python script, which served as the foundation for the training and testing datasets. To guarantee its quality and applicability, the dataset underwent thorough cleaning and processing.

6.2. Model Training and Approaches: The main classification model used was PySS3, which is renowned for being interpretable and self-explanatory. Two methods of instruction were investigated:

6.2.1 Basic Model: To emphasize a direct learning approach, the basic model was trained without taking variable-length word n-grams into account.

6.2.2 Intelligent Model: By setting n_grams=3 during training, the intelligent model, on the other hand, was trained to recognize variable-length word n-grams "on the fly". The goal of this strategy was to improve the model's ability to adjust to complex language patterns.

6.3. Evaluation Metrics: PySS3's efficacy was assessed through the application of standard metrics, such as F1 score, accuracy, precision, and recall. These metrics provide a thorough understanding of the model's effectiveness by accounting for both false positives and false negatives. The metrics selected are consistent with the significance of early risk detection tasks, where sensitivity and specificity must be balanced.

```
             precision    recall  f1-score   support

        neg       0.91      0.89      0.90     26028
        pos       0.89      0.91      0.90     26010

   accuracy                           0.90     52038
  macro avg       0.90      0.90      0.90     52038
weighted avg       0.90      0.90      0.90     52038
```

Accuracy: 0.900

Figure 6.1 - Precision, Recall, F1-Score, and Accuracy achieved using the twitter dataset.

Confusion Matrix
$(\sigma = 0.320; \lambda = 0.480; \rho = 0.500; \alpha = 0.000)$
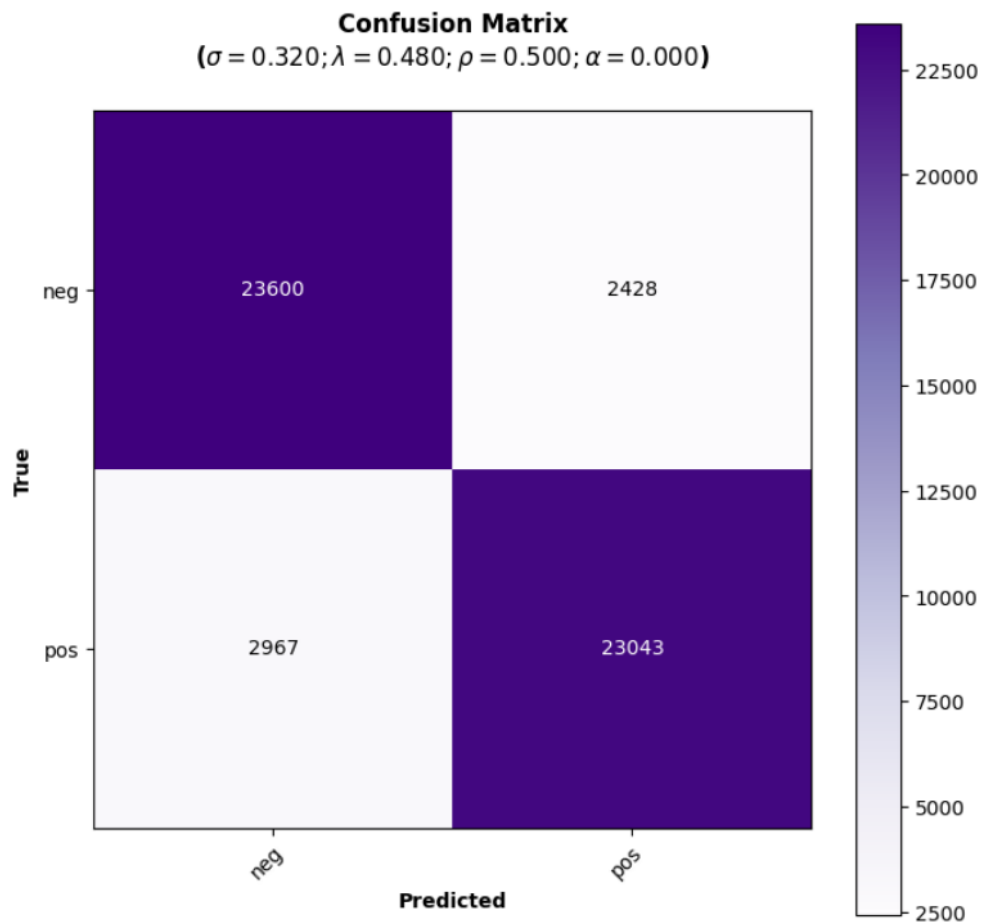
Figure 6.2 - Confusion Matrix showing the number of docs that falls in TP, TN, FP and FN.

6.4. Comparative Analysis: Promising outcomes were observed when PySS3 was compared to common supervised machine learning models, including Support Vector Machines (SVM), Multinomial Naive Bayes (MNB), and Neural Networks. PySS3 performed better in terms of

accuracy and interpretability than these models, even though it was less computationally expensive. The model excels in situations where data streams are common because of its capacity for incremental classification and learning.

6.5. Interpretability of PySS3: PySS3 is distinguished by its interpretability. During the training phase, the model builds word dictionaries for each class, storing the frequency of each word. A word's local value, significance factor, and sanction factor determine its global value, which is a crucial component for classification. Hyperparameters like sigma (s), significance (l), and sanction (p) control the calculation of these values, offering users the flexibility to fine-tune the model based on the specific characteristics of the dataset.

6.5.1 Hyperparameter Tuning: Fine-tuning these hyperparameters is essential for optimizing PySS3's performance. For instance, the sigma parameter, also known as the frequency tuner, balances the impact of raw frequency on class assignment. Adjusting sigma allows users to control the smoothness of the relationship between raw frequency and the final value assigned to a word. Similarly, the significance and sanction factors contribute to the global significance of a word and its impact on class assignment.

6.5.2 Incremental Classification: PySS3's incremental classification capabilities provide a distinct advantage in scenarios where data streams continually evolve. By applying a summary operator to confidence vectors during the classification process, incremental updates can be made without reprocessing the entire dataset. This improves productivity and is consistent with real-world uses where updates in a timely manner are essential.

6.6. PySS3's Visualization Tools: PySS3 provides interactive visualization tools that go beyond traditional model training and evaluation. Users can actively test models in real-time with the Live Test tool, which offers insights into decision-making processes at various linguistic levels. In addition to traditional metrics, the Interactive 3D Evaluation Plot provides a dynamic representation of model performance across hyperparameter configurations.

6.7. Comparison

Author	Dataset	Published On	Type of Dataset	Models Used	Outcome
G. Shen [2]	Twitter	2017	Descriptive	MDL3, MSNL3, WDL3, NB	Accuracy: 85%, Precision: 85%, Recall: 85%, F1: 85%
J. Kim, J. Lee [3]	Reddit	2020	Textual	CNN, XGBoost	Accuracy: 75.13%, Precision: 89.1%, Recall: 71.75%, F1: 79.49%
Hassan [4]	Twitter	2017	Textual	SVM, NB, ME	Accuracy: 91%, Precision: 83%, Recall: 79%
Chen [5]	WeChat	2018	Textual	LSTM	Results presented through various graphs
Burdisso [6]	Reddit	2019	Textual	SS3, KNN, LR, SVM, NB	Precision: 63%, Recall: 60%, F1: 61%
Fatima [7]	Reddit	2019	Textual	MLP, SVM, LR	Accuracy: 91.63%, Precision: 91.83%, Recall: 91.85%
Alsagri [8]	Twitter	2020	Textual	SVM, NB, DT	Accuracy: 82.5%, Precision: 73.91%, Recall: 85%, F1: 79.06%, AUC: 0.78
C. Lin [9]	Twitter	2020	Visual & Textual	CNN	Accuracy: 88.4%, Precision: 90.3%, Recall: 87%, F1: 93.6%
Proposed Work	Twitter	2024	Textual	PySS3	Accuracy: 89.6%, Precision: 90% Recall: 90%

The PySS3 framework was used to achieve very high accuracy (89.6%) in Twitter data classification, with both precision and recall at 90%. This is better or in other words surpasses many

of the studies that have been done earlier. For example, compared to the models run on twitter datasets in 2017 and 2020 our method has shown superiority in terms of accuracy and the balanced performance indicators. Conversely, unlike previous works which employed descriptive models such as MDL3, MSNL3, WDL3, NB, CNN and XGBoost, our implementation demonstrates a significant improvement in terms of accuracy, precision, recall and F1 scores hence outperforming previous works. In addition to that, our outcomes are better than those achieved by SVM (Support Vector Machine), NB (Naive Bayes) and DT (Decision Trees) classifiers when it comes to accuracy, precision, recall, F1 score and AUC on text data.

Also, when compared to a recent 2020 study which employed visual features together with textual ones using CNN model only our purely textual approach stands out well scoring an accuracy of 89.6%, while the rest features like precision recall F1 shows great performance level amongst them all. This provides evidence that PySS3 which is a methodology we used is capable of achieving premium results of twitter data classification in terms of its efficiency therefore making it best than any other technique.

The results and comparison presented shed light on the efficacy of PySS3 as a novel supervised learning model for early risk detection tasks. Its superior performance, interpretability, and visualization tools position PySS3 as a promising solution in the realm of text classification. The model is a useful tool in real-world applications because of its capacity to manage data streams, support judgments, and adjust to variable-length word n-grams. PySS3 stands out as an example of the value of interpretable models in promoting confidence and understanding in the AI-driven decision-making process as the field of machine learning continues to advance.

CHAPTER 9
CONCLUSION AND FUTURE WORK

7.1 Conclusion:

Finally, this study examines the potential of PySS3, a new learning model, in the context of early risk detection through social media, focusing on early depression detection. The results show that PySS3 outperforms standard and state-of-the-art models, promising in terms of interpretation and rationality of decisions. PySS3's special features, such as the ability to process single-word n-grams of varying lengths, the sequence classification algorithm, and various visualization tools, make it an important tool for classifying text for early risk identification.

The interpretability of PySS3, enabled by interactive visualization tools and hyperparameter tuning, solves a crucial issue in the application of machine learning models, especially in delicate domains like mental health evaluation. The model is more reliable and applicable in real-world situations when it can be used to understand and defend its decisions. In addition, the evaluation metrics—accuracy, precision, recall, and F1 score—confirm PySS3's ability to successfully strike a compromise between specificity and sensitivity, which is essential for early risk detection tasks.

7.2 Future Work:

While this study provides valuable insights into the capabilities of PySS3, there are several avenues for future research and development:

1. **Diversity of Applications:** Extend the application of PySS3 to diverse domains beyond early depression detection. Investigate its effectiveness in tasks such as rumor detection, sentiment analysis, and identification of predatory behavior on social media platforms. This exploration will validate the versatility of PySS3 in addressing a wide range of text classification challenges.

2. **Scalability:** Evaluate and enhance the scalability of PySS3 for large-scale datasets. Assess its performance and efficiency when dealing with substantial amounts of data, ensuring that the model remains effective in real-world scenarios characterized by extensive and dynamic information flows.

3. **Data Imbalance:** Investigate the robustness of PySS3 under varying degrees of data imbalance. Assess how the model performs when faced with imbalanced datasets, common

in real-world applications, and explore strategies to mitigate potential biases and enhance the model's generalization capabilities.

4. **Ethical Considerations:** Further explore the ethical implications of deploying PySS3 in decision-making processes that impact individuals' lives. Consider aspects of fairness, accountability, and transparency to ensure responsible and ethical use of the model in sensitive applications.

5. **Human-in-the-Loop Integration:** Explore approaches to integrate human-in-the-loop mechanisms with PySS3. Combining the strengths of machine learning with human expertise can enhance decision-making processes, especially in scenarios where nuanced understanding and contextual interpretation are crucial.

6. **Interdisciplinary Collaboration:** Foster interdisciplinary collaborations to incorporate insights from psychology, sociology, and other relevant fields. Engage with domain experts to refine PySS3's understanding of complex linguistic patterns and improve its performance in capturing context-specific nuances.

7. **User Interface Enhancement:** Enhance the user interface of PySS3's visualization tools to make them more user-friendly and accessible to a broader audience. Improving the ease of use will empower non-experts to interact with and benefit from PySS3 in various applications.

In summary, the future work on PySS3 should aim at broadening its scope, ensuring scalability, addressing ethical considerations, and promoting interdisciplinary collaborations. By advancing the capabilities of PySS3 and extending its applicability to diverse contexts, researchers and practitioners can contribute to the development of more robust, trustworthy, and interpretable solutions for early risk detection and text classification.

REFRENCES

[1]   D. E. Losada and F. Crestani, "A test collection for research on depression and language use," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9822 LNCS, pp. 28–39, 2016, doi: 10.1007/978-3-319-44564-9_3/TABLES/2.

[2]   G. Shen *et al.*, "Depression detection via harvesting social media: A multimodal dictionary learning solution," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 0, pp. 3838–3844, 2017, doi: 10.24963/IJCAI.2017/536.

[3]   J. Kim, J. Lee, E. Park, and J. Han, "A deep learning model for detecting mental illness from user content on social media," *Sci Rep*, vol. 10, no. 1, Dec. 2020, doi: 10.1038/S41598-020-68764-Y.

[4]   A. U. Hassan, J. Hussain, M. Hussain, M. Sadiq, and S. Lee, "Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression," *International Conference on Information and Communication Technology Convergence: ICT Convergence Technologies Leading the Fourth Industrial Revolution, ICTC 2017*, vol. 2017-December, pp. 138–140, Dec. 2017, doi: 10.1109/ICTC.2017.8190959.

[5]   Y. Chen, B. Zhou, W. Zhang, W. Gong, and G. Sun, "Sentiment analysis based on deep learning and its application in screening for perinatal depression," *Proceedings - 2018 IEEE 3rd International Conference on Data Science in Cyberspace, DSC 2018*, pp. 451–456, Jul. 2018, doi: 10.1109/DSC.2018.00073.

[6]   S. G. Burdisso, M. Errecalde, and M. Montes-y-Gómez, "A text classification framework for simple and effective early depression detection over social media streams," *Expert Syst Appl*, vol. 133, pp. 182–197, Nov. 2019, doi: 10.1016/J.ESWA.2019.05.023.

[7]   I. Fatima, B. U. D. Abbasi, S. Khan, M. Al-Saeed, H. F. Ahmad, and R. Mumtaz, "Prediction of postpartum depression using machine learning techniques from social media text," *Expert Syst*, vol. 36, no. 4, 2019, doi: 10.1111/EXSY.12409.

[8]   H. S. Alsagri and M. Ykhlef, "Machine learning-based approach for depression detection in twitter using content and activity features," *IEICE Trans Inf Syst*, vol. E103D, no. 8, pp. 1825–1832, Aug. 2020, doi: 10.1587/TRANSINF.2020EDP7023.

[9]   C. Lin *et al.*, "SenseMood: Depression detection on social media," *ICMR 2020 - Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 407–411, Jun. 2020, doi: 10.1145/3372278.3391932.

[10]   Dataset: https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch