Suicide Analysis and Prevention Application using Machine Learning Classifiers

Vasudha Phaltankar[1], Abhishek Chavan[2], Mayuresh Bhangale[3], Ayan Memon[4], Apurva Dikondawar[5]

[1]Assistant Professor, Dept. of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pune, Maharashtra, India
[2-5]Student, Dept. of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pune, Maharashtra, India

---***---

Abstract - *Suicide is the 2nd leading cause of death causing more than 8 lakh deaths per year and even more suicide attempts. The phenomenon of an individual having suicidal thoughts is called suicidal ideation and to seek out and help these individuals is suicidal ideation detection. Although much research has been conducted on the assessment and management of individuals on the risk of committing suicide, what these research papers lack is the analysis of real time data and therefore fails to provide medical or psychological help. We set to tackle both these problems, firstly by increasing the incoming data and collecting data from an environment which is more effective than any other surveys. We coined this as seeking out the virtual suicide notes. This led us to a platform like reddit, which creates a same and anonymous place for the user to express without any animosity. In this project we will be using machine learning based classifiers for suicide detection like TF-IDF and CNN. This paper uses a TFIDF-Count Vectorizer-Multinomial Bayes model to analyze reddit data with the highest AUC score of 92% and F1- Score of 96%.*

Key Words*: *Suicide, depression, analysis, natural-language, reddit, subreddits, prevention.

1.INTRODUCTION

In the modern society mental health issues like depression and anxiety are increasing day by day and appear to be more severe in developed countries. A WHO report says that almost 800,000 people die of suicide and suicidal attempts every year making it the 2nd emerging cause of death and occur mostly in the young age group of 15 to 29 years.

Many factors lead to suicide such as work pressure, loneliness, hopelessness, schizophrenia, social isolation, negative events and many more. These mental disorders if left untreated can lead to suicidal thoughts and ultimately suicide.

Suicidal ideation is also called suicidal thoughts which are thoughts experienced b people about committing suicide. It is an indicator that the individual is susceptible to committing suicide. Large number of people mostly teenagers are observed to have suicidal ideation. These teenagers are likely to share these thoughts on online platform and social media.

A study conducted on the suicidality found that young people report their suicidal ideations not verbally, but through online modes of communication. This includes posts on social networking sites, blogs, text messages, statuses, emails, etc.

An extensive study of this report shows that in the future this method of expressing distress online and anonymously is only going to increase. It further gets slippery as to what extent these online expressions produce an actual suicidal risk. Some studies have tried to analyze this risk and assess the correlation between the two but they have been conducted on a general spectrum and do not capture the broad classification of data of suicidal ideation.

Suicidal Ideation Detection studies and evaluates whether an individual is having suicidal thoughts based on a particular data be it may a text message, an online post, a blog or anything else. The anonymous nature of these platform helps individuals to freely express themselves that they wouldn't otherwise in the real world.

This online user-generated data gives an idea for detecting suicidal intensions early and thus providing treatment. Ultimately, these online platforms have started to act as surveillance for suicidal ideation, and mining social content to improve suicide prevention. But these platforms have brought along a conundrum of social phenomenon which include online communities performing self-harm and copycat suicide.

For example, a social network phenomenon called the "Blue Whale Game" in 2016 uses many tasks (such as self-harming) and leads game members to commit suicide in the end. Suicide is a critical social issue and takes thousands of lives every year. Thus, it is necessary to detect suicidality and to prevent suicide before victims end their life. Detecting these indications at an early stage and taking proper steps accordingly can prevent many potential suicide attempts.

In this project we develop a machine learning approach based on reddit data. This project aims to use machine learning classifiers in correspondence with Natural

Language Processing to accurately identify the severity of the text and the risk of suicide.

2. LITERATURE SURVEY

In recent years, much research has been done to successfully detect suicidal ideation among individuals especially those influenced by social media. highlight the influencing potential of social media on suicide ideation. One of such research was conducted by Amanda Merchant which explored the relationship between self-harm and the use of internet. This paper conducted a review of articles publish between 4 years. These articles were based on various studies which exploited individuals engaging in self-harm or suicidal ideations due to the use of internet. Although this study was successful in establishing a correlation between suicidal ideation and internet it failed to specify down to which platforms were causing these ideations.

The factors lacking the first paper were somewhat tackled in this another paper by Thomas D. Ruder, Gary M. Hatch in which they specifically researched the influence of a social media platform-Facebook on suicidal behaviour. This paper explored the effects of suicide notes on Facebook, what were causing them, copycat suicides and most importantly how to prevent them. The results were such that found various reports of suicide notes on Facebook, however there was no evidence of copycat suicide and therefore no clear distinction between which cases needed immediate intervention and which did not.

Alison L. Calear, Philip J. Batterham introduced a study that aims how often individuals experience suicidal ideations. They conducted a survey spanning for 12 months and found out that 39% of the survey participants did not disclose suicidal ideations to anyone, 47% disclosed on social media platforms and 42% sought medical professional help. This survey however did not provide a solution for those undisclosed 39% participants.

Another research by Jason B. Luoma, Catherine E. Martin, Jane L. Pearson studied how many primary care and medical care professionals come in contact with individuals before they die with suicide. The study resulted that 45% of suicide victims had contact with a medical professional within 1 month of committing suicide. But it failed to provide to what degree these medical and primary health care professionals can prevent suicide.

In research conducted by Glen Xiong, MD and Debra Kahn they explored the number of factors that hinder suicide risk assessments and thereby prevent suicidal ideation detection. The study discussed that despite having medical care professionals, adolescent suicide rates are still rising due to lack of screening for suicidality and lack of training of the first responders.

Another survey conducted by Manuel A Franco-Martin, Juan Luis Munoz-Sanchez, Beatriz Sainz-de-Abajo, they explored the role of different technologies for suicide ideation detection and their prevention. The results pointed towards various platforms like web, mobile, primary care and social networks but also suggested that collaboration of these platforms would be more beneficial towards the advancements on suicidal ideation detection.

In another research conducted by M J Marttunen, M M Henriksson, H M Aro, J K Lönnqvist, they presented a study which showed among a survey how many people communicate their suicidal intent.

Our final survey was one of a research conducted by Shini Renjith , Annie Abraham, Surya B. Jyothi, Lekshmi Chandran, Jincy Thomson. This research uses a real time database to build a suicidal ideation model using deep learning techniques and settle upon a LSTM-CNN model to detect emotions from the databse.

3. DATA

3.1 Data Collection

For this project, we will be using Reddit's API to collect posts from two subreddits: "r/depression" and "r/SuicideWatch". This API gives us the data of 1000 unique posts each day. We aim to automate as much of this process as possible into neat functions to enable repeatability on the data collection front. When collecting data from servers, we will create a randomized delay between requests as a consideration to Reddit's servers and security staff. The user's privacy will be respected by assigning a unique ID to each post. The data for this project was collected between August 2021 to December 2022. Do note that if you collect the data between different time period, it will result in a new set of posts being scraped. These posts were mapped into six different progressive stages from "falling short of expectations"(stage one) to "high self-awareness"(stage three) to the final stage of "disinhibition".

- Stage 1: Falling short of expectations
- Stage 2: Attributions to self
- Stage 3: High self-awareness
- Stage 4: Negative effect
- Stage 5: Cognitive Deconstruction
- Stage 6: Disinhibition

The following table (Table 1) gives some examples of all six stages.

Table -1: Stage Classification of Reddit data

Label	Reddit Post
Stage 1	I never really understood the value of honesty until recently, it can both heal and destroy you.
Stage 2	I am the definition of a hypocrite. An angst-ridden, over-sensitive delinquent . . . I'm a stupid fucking drama queen. It's not other people. It's just me.
Stage 3	I'm still marvelling at the fact you can thrive in a world of inner happiness; she observes. And then let peripheral reminders crash through and rewire your brain to make it think that pursuing contentment is useless.
Stage 4	I know realistically that this anxiety thing is not going to go away. I will live my life completely alone.
Stage 5	Right now, the future seems like a dark, obscure place that I can't see myself in. There's nothing ahead. Nothing.
Stage 6	I guess I'm nothing more than another suicidal white girl, just another first-world brat succumbing to society's perfect illusions.

A) Exploring the architecture of the subreddit pages

We start by scraping the r/depression and r/SuicideWatch subreddits. This data is then stored in two separate csv files with the same name. Then we explore the HTML innards of both the subreddits in json format. We define a user agent and make sure the status is good to go. The reddit data is then organized as a dictionary with 100 attributes. Some of them are: kind, data, dist, subreddit, selftext, author_fullname, hide_score, quarantine, gilded, clicked, subreddit_type, is_meta, category, approved_by, is_media, etc. We now get the keys from our dictionary data; *['after', 'before', 'children', 'dist', 'modhash']*. The *after* key is the query string that will indicate in our URL that we want to see in the next 25 posts. We then data frame the csv files by using *data* and *children* keys. This is the final data we will use for processing.

B) Creating functions to automate the Data Collection process

Now, we can define a function to scrape a reddit page. The scraped posts will be contained in output list which should be empty. This is useful for the first scrape from the virgin subreddit.

```
after = None
for _ in range(number_of_scrapes):
    if _ == 0:
        print("SCRAPING {}".format(url_string))
        print("<<<SCRAPING COMMENCED>>>")
        print("Downloading Batch {} of {}...".format(1, number_of_scrapes))
    elif (_+1) % 5 ==0:
        print("Downloading Batch {} of {}...".format((_ + 1), number_of_scrapes))

    if after == None:
        params = {}
    else:
```

Fig -1: Code snippet for scraping data

This will tell the scraper to get next set of data from reddit API

```
        params = {"after": after}
    res = requests.get(url_string, params=params, headers=headers)
    if res.status_code == 200:
        the_json = res.json()
        output_list.extend(the_json["data"]["children"])
        after = the_json["data"]["after"]
    else:
        print(res.status_code)
        break
    time.sleep(randint(1,6))

print("<<<SCRAPING COMPLETED>>>")
print("Number of posts downloaded: {}".format(len(output_list)))
print("Number of unique posts: {}".format(len(set([p["data"]["name"] for p in output_list]))))
```

Fig -2: Code snippet for automating collection

After scraping we perform the following operations:

- Calling the function on depression subreddit
- Defining an empty list to contain our scraped data
- Creating a function to output a list of unique posts
- Checking if the new list is of same length as of unique posts
- Calling the function on our scraped data
- Putting depression data into data frame and saving to csv
- Checking if there are100 columns and last column is *is_suicide*.

C) Running our functions on the r/SuicideWatch subreddit

We perform the same operations performed on the depression subreddit to the SuicideWatch subreddit and get the corresponding classification of data for suicide parameters.

Summary:
The API allows us to collect data on approximately 1000 data per day. We collection 980 r/SuicideWatch posts and 917 r/depression posts on our first round of collection. In the automation process for further data collection rounds we aimed to even out the posts with a target set to 950 posts for both the subreddits.

Following tables show some examples of the final form of the data from both the subreddits.

Table -2: r/depression data collection and classification

Sr. No.	Approved at UTC	Subreddit	Selftext	Author full name	Saved	Mod reason title	Gilded	Clicked	Title	Permalink	Parent Whitelist status	Stickied	URL	Num cross posts	Media	Is video	Is suicide
1	None	depression	We understand that most people who reply immed...	t2_1t70	FALSE	None	0	FALSE	Our most-broken and least-understood rules is ...	/r/depression/comments/doqwow/our_mostbroken_a...	no_ads	TRUE	https://www.reddit.com/r/depression/comments/d...	0	None	FALSE	0
2	None	depression	Welcome to /r/depression's check-in post - a p...	t2_64qjj	FALSE	None	0	FALSE	Regular Check-In Post	/r/depression/comments/exo6f1/regular_checkin_...	no_ads	TRUE	https://www.reddit.com/r/depression/comments/e...	0	None	FALSE	0
3	None	depression	I've been feeling really depressed and lonely ...	t2_17aooz	FALSE	None	0	FALSE	I hate it so much when you try and express you...	/r/depression/comments/fedwbi/i_hate_it_so_muc...	no_ads	FALSE	https://www.reddit.com/r/depression/comments/f...	0	None	FALSE	0
4	None	depression	I literally broke down crying and asked to go ...	t2_5v2j4itq	FALSE	None	0	FALSE	I went to the hospital because I was having re...	/r/depression/comments/feel0k/i_went_to_the_ho...	no_ads	FALSE	https://www.reddit.com/r/depression/comments/f...	0	None	FALSE	0
5	None	depression	Any kind soul want to give a depressed person ...	t2_15xfmv	FALSE	None	0	FALSE	Cake day for me	/r/depression/comments/fe6ua3/cake_day_for_me/	no_ads	FALSE	https://www.reddit.com/r/depression/comments/f...	0	None	FALSE	0

Table -3: r/SuicideWatch data collection and classification

Sr. No.	Approved at UTC	Subreddit	Selftext	Author full name	Saved	Mod reason title	Gilded	Clicked	Title	Author cake_day	Parent Whitelist status	Stickied	URL	Num cross posts	Media	Is video	Is suicide
1	None	SuicideWatch	We've been seeing a worrying increase in pro-s...	t2_1t70	FALSE	None	1	FALSE	New wiki on how to avoid accidentally encourag...	NaN	no_ads	TRUE	https://www.reddit.com/r/SuicideWatch/comments...	0	None	FALSE	1
2	None	SuicideWatch	If you want to recognise an occasion, please d...	t2_1t70	FALSE	None	0	FALSE	Reminder: Absolutely no activism of any kind i...	NaN	no_ads	TRUE	https://www.reddit.com/r/SuicideWatch/comments...	0	None	FALSE	1
3	None	SuicideWatch	I really fucking feel you	t2_111wkq	FALSE	None	0	FALSE	To every single poster here i wanne say one thing	NaN	no_ads	FALSE	https://www.reddit.com/r/SuicideWatch/comments...	0	None	FALSE	1
4	None	SuicideWatch	Everyone ends up hating me eventually. \nMy	t2_4de9uxb2	FALSE	None	0	FALSE	I just want it all to stop	NaN	no_ads	FALSE	https://www.reddit.com/r/SuicideWatch/comments...	0	None	FALSE	1

3.2. Data Cleaning, Preprocessing and EDA

A) Data Cleaning

Cutting down the dataset - As both sets have 100 columns, it was to choose a few columns that will be useful as our predictors.

Concatenation - As we have already created a "*is_suicide*" column indicating which subreddit the posts are from; we concatenated both datasets together.

Imputation - If there are missing values, we imputed the data.

Choosing relevant columns

a) Title and Post - We felt that the text data in the both the title and the post itself can potentially serve our classifier well.

b) Author's handle and number of comments - The author's name and the number of comments is curve ball choices. There just might be some connection between a user's handle and his/her psyche. There also might be a connection between the number of comments made.

c) URL - We left the URL in for reference. In case we'd want to look deeper into a particular post.

We picked out five columns that could help in our EDA: *Title, Selftext, Author, Num_comments, is_suicide, URL*. Further, we concatenated the selected parts of both the databases before cleaning it. Then we stored the data into a combined csv file. We then checked for the missing data and then data

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 02 | Feb 2022 www.irjet.net p-ISSN: 2395-0072

imputation decision is made which is replacing NaNs with "emptypost".

B) Data Preprocessing

As the posts are written by different individuals, they come in different forms. In order to prepare the data for our classifier, we took steps to pre-process the posts.

Build processing functions - We built a processing function that will help change the text to lowercase, remove punctuations, reduce related words down to a common base word. With these functions, we created a separate column for our clean data.

The operations performed to clean the data were:

- Tokenised long string in a list
- Created a list with no punctuations
- Lemmatise the items in above mentioned list
- Removed stop words from the lemmatized list
- Returned separated words into a long string

```
def processing_text(series_to_process):
    new_list = []
    tokenizer = RegexpTokenizer(r'(\w+)')
    lemmatizer = WordNetLemmatizer()
    for i in range(len(series_to_process)):
        dirty_string = (series_to_process)[i].lower()
        words_only = tokenizer.tokenize(dirty_string)
        words_only_lem = [lemmatizer.lemmatize(i) for i in words_only]
        words_without_stop = [i for i in words_only_lem if i not in stopwords.words("english")]
        long_string_clean = " ".join(word for word in words_without_stop)
        new_list.append(long_string_clean)
    return new_list
```

Fig -3: Code snippet for cleaning data

Table -4: Data after cleaning operations

Title	**After Cleaning**
Our most-broken and least-understood rules is "helpers may not invite private contact as a first...	broken least understood rule helper may invite private contact first resort made new wiki explain
Regular Check-In Post	regular check post
I hate it so much when you try and express your feelings to your parents, but they turn it aroun...	hate much try express feeling parent turn around compare suffering
I went to the hospital because I was having really bad panic attacks, and theycontinued in there...	went hospital wa really bad panic attack theycontinued ended collapsing nurse wa trying help sai...
Cake day for me	cake day

C) Exploratory Data Analysis

a) Top words - This was our obvious first step in our EDA. To peek and see what are the most used words in the title, posts and usernames.

We visualised the most-used words in the respective subreddit posts in a word cloud and a bar plot. We first defined a function that helped us with that. The function was re-used for our titles and usernames later.

Word Cloud

- Checked out the common words from both data files using Count-Vectorizer
- Created a data frame of extracted words
- Created a long string for word cloud module consisting of 40 words

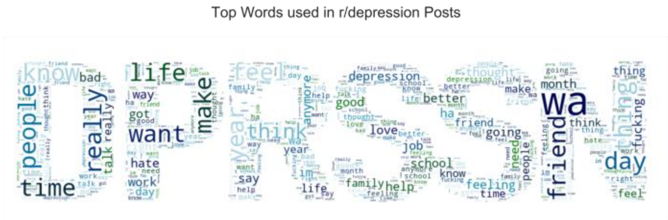

Fig -4: Depression Word Cloud

Fig -5: SucideWatch Word Cloud

Bar Plot

- Created a final data frame of top 20 words
- Plotted the count of those 20 words

b) Significant Authors - This might not really affect the classifier that we are using, but it is worth it to check out users who post often and users who has posted on both subreddits.

- Used Masks and Slices to fish out authors who have posted in both subreddits
- Created a data frame with mean values of *is_suicide* column
- A mean value that is not 1 or 0 would indicate that they post in both subreddits.
- Created a mask to isolate redditors who have posted in both the forums
- Created a list of double posters

c) Average length of posts - We have already noticed a significant number of posts with no words in r/SuicideWatch posts. So, we decided to in to check out how many numbers of words are in an average post in each subreddit.

Average length of a r/depression title: 39.319520174
Average length of a r/SuicideWatch title: 39.7224489
Average length of a r/depression post: 964.32170119
Average length of a r/SuicideWatch post: 836.360204

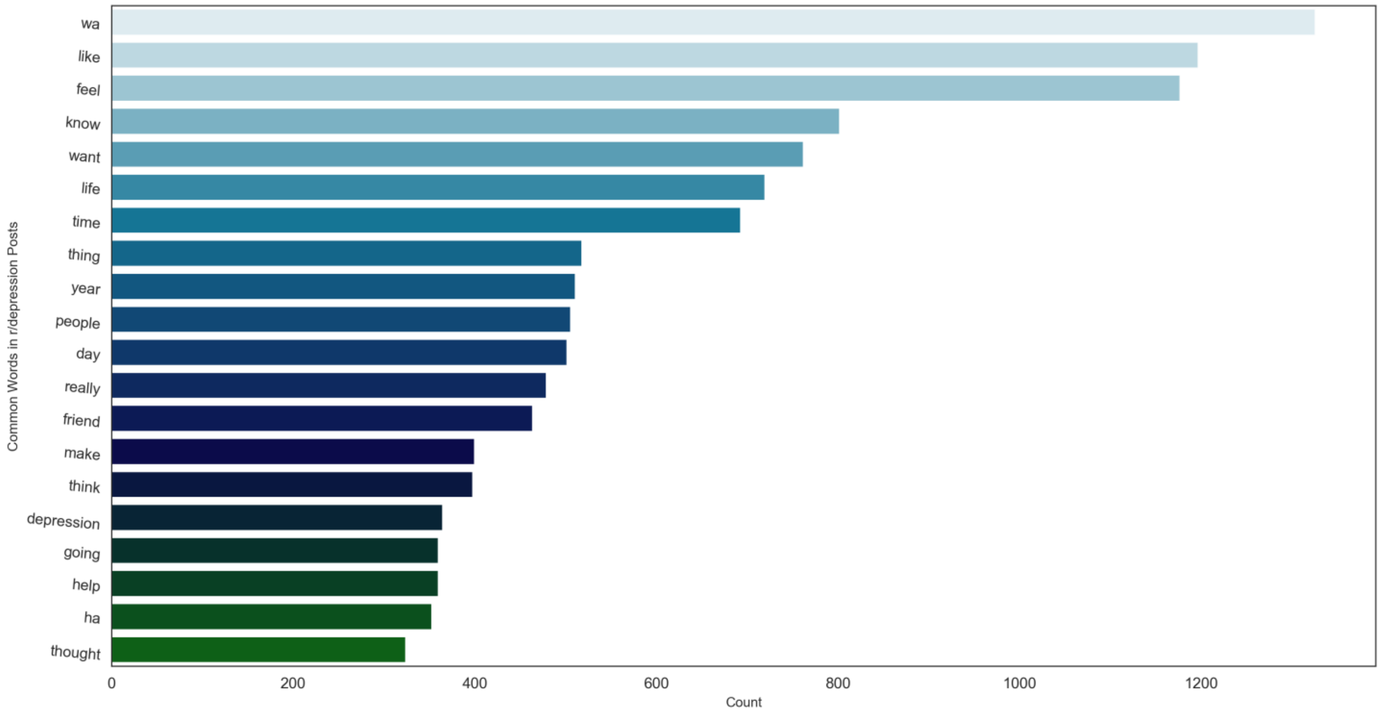

Fig -6: Depression Bar Plot

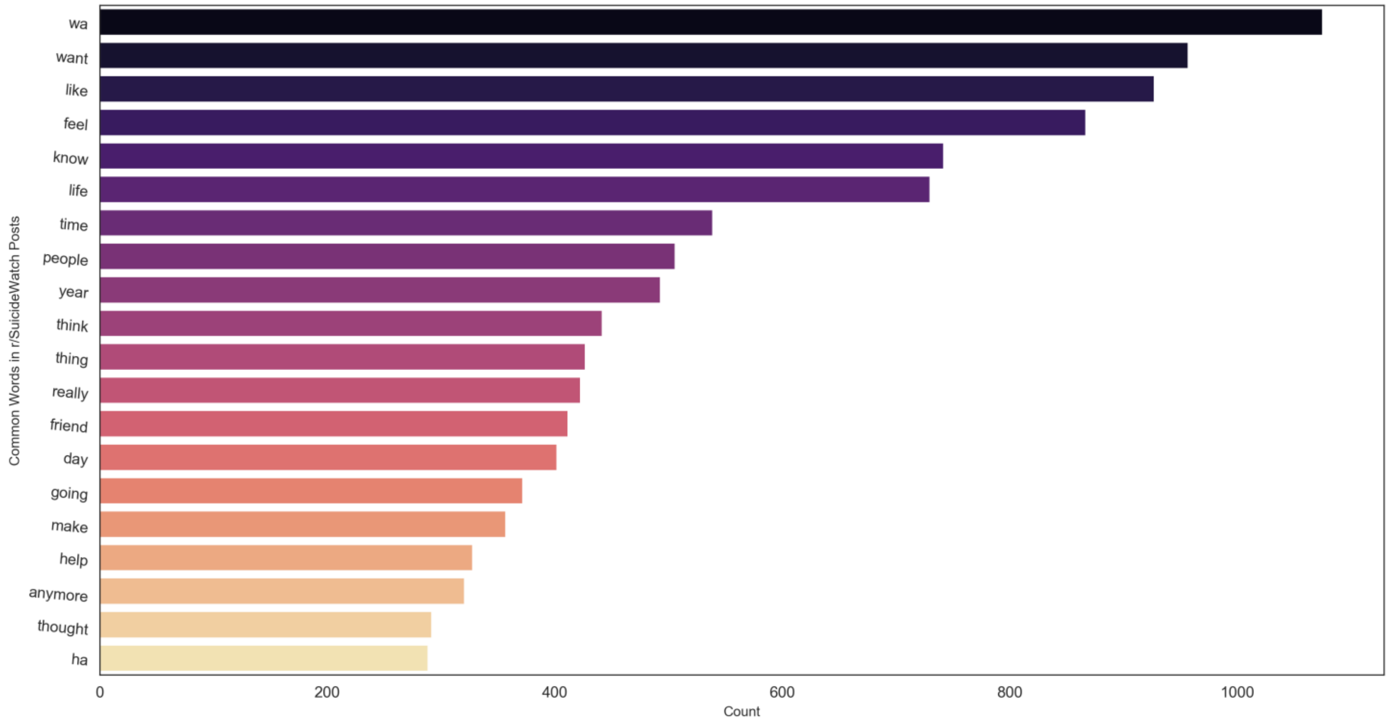

Fig -7: SuicideWatch Bar Plot

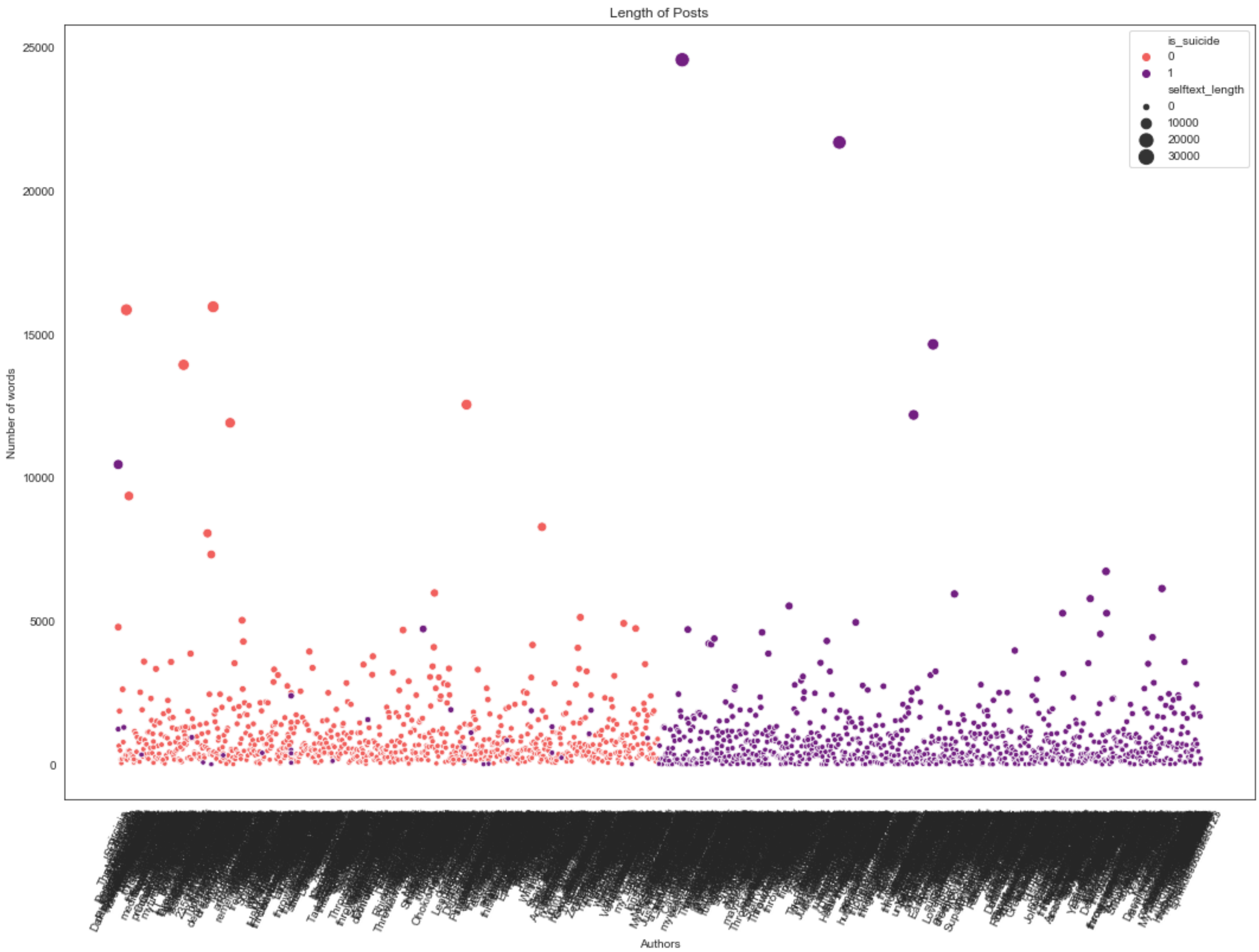

Fig -8: Length of Posts Scatter Plot

4. MODELLING

4.1 Establishing a Baseline

We will first calculate the baseline score for our models to "out-perform". A baseline score in the context of our project be the percentage of us getting it right if we predict that all our reddit posts are from the r/SuicideWatch subreddit. The baseline models considered include some of the machine learning classifiers like:

a. Count Vectorizer – Count Vectorizer is used in Natural Language Processing related problems where a large amount of data needs to be vectorized i.e., removing stop words, lemmatizing, stemming, tokenization.
b. LSTM – LSTM is a type of recurrent neural network which is better, in terms of memory, than the traditional recurrent neural networks thus helping to maintain relevant data and discard irrelevant data.
c. Multinomial Naive Bayes – It is a supervised learning classifier that is used for analyzing categorical text data.
d. K-Nearest Neighbours – K-nearest neighbors (KNN) algorithm is a supervised machine learning algorithm that can be used to solve both classification and regression problems
e. TF-IDF Vectorizer – TF-IDF Vectorizer is widely used in extracting information and text mining. It measures the importance of a word in a document.
f. CNN – Convolutional layer applies convolutional operation on the data to produce feature maps which contains information about the patterns in the data.
g. Combined TF-IDF-CNN model – This model uses the TF-IDF layer to assign a score to each word and their corresponding categorization along with the convolutional layer for generating feature maps.

Our Baseline Accuracy is 71.66%. Baseline accuracy is basically our score if we assume everything is 1.

4.2 Selecting Columns for Feature Engineering

Before moving forward to creating a production model, we will run a Count Vectorizer + Naive Bayes model on different columns and score them. This will help us pick which one that we will use to build more models on.

Understanding the confusion matrix: In the context of our project, these are what the parameters in our confusion matrix represent:

True Positives (TP) - We predict that an entry is from the r/SuicideWatch subreddit and we got that right. As we are seeking to identify suicide cases, our priority is to get as many of these!

True Negatives (TN) - We predict that an entry is from the r/depression subreddit and we get it right. This also means that we did well.

False Positives (FP) - We predict that an entry is from the r/SuicideWatch subreddit and we get it wrong. Needless to say, this is undesirable.

False Negatives (FN) - We predict that an entry is from the r/depression subreddit and BUT the entry is actually from r/SuicideWatch. This is the worst outcome. That means we might be missing out on helping someone who might be thinking about ending their life.

Table -5: Initial Score

Series	AUC Score	Precision	Recall	Confusion Matrix	Train Accuracy	Test Accuracy	Baseline Accuracy	Specificity	F1-score
selftext	0.74	0.71	0.73	{'TP': 161, 'FP': 78, 'TN': 152, 'FN': 84}	0.92	0.76	0.62	0.76	0.76
author	0.71	0.77	0.77	{'TP': 235, 'FP': 204, 'TN': 26, 'FN': 10}	0.99	0.65	0.62	0.71	0.65
title	0.82	0.8	0.71	{'TP': 167, 'FP': 104, 'TN': 126, 'FN': 78}	0.85	0.72	0.62	0.65	0.72
selftext_clean	0.75	0.77	0.71	{'TP': 165, 'FP': 78, 'TN': 152, 'FN': 80}	0.91	0.77	0.62	0.76	0.77
author_clean	0.69	0.66	0.73	{'TP': 169, 'FP': 155, 'TN': 75, 'FN': 76}	0.95	0.61	0.62	0.63	0.65
title_clean	0.82	0.78	0.78	{'TP': 178, 'FP': 112, 'TN': 118, 'FN': 67}	0.84	0.72	0.62	0.61	0.72
megatext_clean	0.91	0.88	0.82	{'TP': 160, 'FP': 71, 'TN': 159, 'FN': 85}	0.95	0.82	0.62	0.79	0.82

Final choice made: ***megatext_clean*** **as our "Production Column"**
Based on a combination of scores from our modelling exercise above, we proceeded with **megatext_clean** -- a combination of our cleaned titles, usernames and posts -- as the column we will use to draw features from.
Some reasons why:
> Generalising Well - The model using *megatext_clean*'s test set scored a 0.82 (the joint highest) while its training set score a 0.95.
> High ROC Area Under Curve score - As our classes are largely balanced, it is suitable to use AUC Scores as a metric to measure the quality of our model's predictions. Our top choice performs best there.
> Best recall/sensitivity score - This score measures the ratio of the correctly positive-labelled (is in r/SuicideWatch) by our program to all who are truly in r/SuicideWatch. As that is the target of our project, that the model performed well for this metric is important and perhaps, most important to us.
>False Negatives (FN) - We predicted that an entry is from the r/depression subreddit and BUT the entry is actually from r/SuicideWatch. This is the worst outcome. That means we might be missing out on helping someone who might be thinking about ending their life.

4.3 Searching Production model

Inspired by our earlier function, we created a similar function that will run multiple permutations of models with Count, Multinomial Naïve Bayes and TF-IDF Vectorizers. The resulting metrics will be held neatly in a data frame.

We started by splitting the train test function and instantiated the pipeline. After getting predictions from the model, we defined the confusion matrix elements thereby creating a dictionary from the classification report which we will use for drawing the metrics from.

After calculating the area under the curve, we define data frame columns and append our results in a neatly arranged data frame.

```
def gridsearch_multi(steps_titles, steps_list, pipe_params):
    X = model_data["megatext_clean"]
    y = model_data['is_suicide']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42, stratify=y)
    gs_results = pd.DataFrame(columns=['model','AUC Score', 'precision', 'recall (sensitivity)',
                                       'best_params', 'best score', 'confusion matrix',
                                       'train_accuracy','test_accuracy','baseline_accuracy',
                                       'specificity', 'f1-score'])
    for i in range(len(steps_list)):
        pipe = Pipeline(steps=steps_list[i])
        gs = GridSearchCV(pipe, pipe_params[i], cv=3)
        gs.fit(X_train, y_train)
        pred = gs.predict(X_test)
        tn, fp, fn, tp = confusion_matrix(y_test, gs.predict(X_test)).ravel()
        classi_dict = (classification_report(y_test,pred, output_dict=True))
        gs.predict_proba(X_test)
        pred_proba = [i[1] for i in gs.predict_proba(X_test)]
        auc = roc_auc_score(y_test, pred_proba)
        model_results = {}
        model_results['model'] = steps_titles[i]
        model_results['AUC Score'] = auc
        model_results['precision']= classi_dict['weighted avg']['precision']
        model_results['recall (sensitivity)']= classi_dict['weighted avg']['recall']
        model_results['best params'] = gs.best_params_
        model_results['best score'] = gs.best_score_
        model_results['confusion matrix']={"TP": tp,"FP":fp, "TN": tn, "FN": fn}
        model_results['train accuracy'] = gs.score(X_train, y_train)
        model_results['test accuracy'] = gs.score(X_test, y_test)
        model_results['baseline accuracy'] = 0.5166
        model_results['specificity']= tn/(tn+fp)
        model_results['f1-score']= classi_dict['weighted avg']['f1-score']
        df_list.append(model_results)
        pd.set_option("display.max_colwidth", 200)
    return (pd.DataFrame(df_list)).round(2)
```

Fig -9: Code snippet of production model function

We used this derived function with Count Vectorizer, TF-IDF Vectorizer and on Multinomial Naive Bayes algorithm.

```
df_list=[]
steps_titles = ['cvec+ multi_nb','cvec + ss + knn','cvec + ss + logreg']
steps_list = [
    [('cv', CountVectorizer()),('multi_nb', MultinomialNB())],
    [('cv', CountVectorizer()),('scaler', StandardScaler(with_mean=False)),('knn', KNeighborsClassifier())],
    [('cv', CountVectorizer()),('scaler', StandardScaler(with_mean=False)),('logreg', LogisticRegression())]
]
pipe_params = [
    {'cv__stop_words':['english'], 'cv__ngram_range':[(1,1),(1,2)],'cv__max_features': [20, 30, 50],'cv__min_df'
    {'cv__stop_words':['english'], 'cv__ngram_range':[(1,1),(1,2)],'cv__max_features': [20, 30, 50],'cv__min_df'
    {'cv__stop_words':['english'], 'cv__ngram_range':[(1,1),(1,2)],'cv__max_features': [20, 30, 50],'cv__min_df'
]
gridsearch_multi(steps_titles, steps_list, pipe_params)
```

Fig -10: Code snippet of Count Vectorizer function

```
steps_titles = ['tvec + multi_nb','tvec + ss + knn','tvec + ss + logreg']

steps_list = [
    [('tv', TfidfVectorizer()),('multi_nb', MultinomialNB())],
    [('tv', TfidfVectorizer()),('scaler', StandardScaler(with_mean=False)),('knn', KNeighborsClassifier())],
    [('tv', TfidfVectorizer()),('scaler', StandardScaler(with_mean=False)),('logreg', LogisticRegression())]
]

pipe_params = [
    {'tv__stop_words':['english'], 'tv__ngram_range':[(1,1),(1,2)],'tv__max_features': [20, 30, 50],'tv__min_
    {'tv__stop_words':['english'], 'tv__ngram_range':[(1,1),(1,2)],'tv__max_features': [20, 30, 50],'tv__min_
    {'tv__stop_words':['english'], 'tv__ngram_range':[(1,1),(1,2)],'tv__max_features': [20, 30, 50],'tv__min_
]

gridsearch_multi(steps_titles, steps_list, pipe_params)
```

Fig -11: Code snippet of TF-IDF Vectorizer function

```
steps_titles = ['hvec + multi_nb','hvec + ss + knn','hvec + ss + logreg']

# CODE FOR PIPELINE TO INSTATIATE MODELS
steps_list = [
    [('hv', MultinomialNB(alternate_sign=False)),('multi_nb', MultinomialNB())],
    [('hv', MultinomialNB(alternate_sign=False)),('scaler', StandardScaler(with_mean=False))
    [('hv', MultinomialNB(alternate_sign=False)),('scaler', StandardScaler(with_mean=False))
]
pipe_params = [
    {'hv__stop_words':['english'], 'hv__ngram_range':[(1,1),(1,2)]},
    {'hv__stop_words':['english'], 'hv__ngram_range':[(1,1),(1,2)]},
    {'hv__stop_words':['english'], 'hv__ngram_range':[(1,1),(1,2)]}
]

gridsearch_multi(steps_titles, steps_list, pipe_params)
```

Fig -12: Code snippet of Multinomial Naive Bayes function

Table -6: Final Production model

Model	AUC Score	Precision	Recall	Best Params	Best Score	Confusion Matrix	Train Accuracy	Test Accuracy	Baseline Accuracy	Specificity	F1-Score
cvec+ multi_nb	0.82	0.77	0.77	{'cv__max_df': 0.3, 'cv__max_features': 50, 'cv__min_df': 2, 'cv__ngram_range': (1, 1), 'cv__stop_words': 'english'}	0.85	{'TP': 160, 'FP': 71, 'TN': 159, 'FN': 85}	0.68	0.84	0.82	0.85	0.82
cvec + ss + logreg	0.83	0.79	0.79	{'cv__max_df': 0.3, 'cv__max_features': 50, 'cv__min_df': 2, 'cv__ngram_range': (1, 1), 'cv__stop_words': 'english'}	0.85	{'TP': 173, 'FP': 75, 'TN': 155, 'FN': 72}	0.69	0.78	0.82	0.83	0.85
tvec + multi_nb	0.85	0.84	0.78	{'tv__max_df': 0.3, 'tv__max_features': 50, 'tv__min_df': 2, 'tv__ngram_range': (1, 2), 'tv__stop_words': 'english'}	0.88	{'TP': 169, 'FP': 77, 'TN': 153, 'FN': 76}	0.68	0.85	0.82	0.82	0.91
tvec + ss + logreg	0.83	0.77	0.82	{'tv__max_df': 0.3, 'tv__max_features': 50, 'tv__min_df': 2, 'tv__ngram_range': (1, 2), 'tv__stop_words': 'english'}	0.82	{'TP': 160, 'FP': 72, 'TN': 158, 'FN': 85}	0.68	0.87	0.82	0.88	0.89
cvec + tvec + multi_nb	0.92	0.91	0.9	{'hv__ngram_range': (1, 1), 'hv__stop_words': 'english'}	0.93	{'TP': 148, 'FP': 54, 'TN': 176, 'FN': 97}	0.89	0.92	0.82	0.93	0.96

Narrowing down to three models

The *Count Vectorizer + TF-IDF Vectorizer + Multinomial Naive Bayes* model out-performed other models on all metrics. Especially our much-prized AUC score (0.92) and the recall score (which measures our model's ability to predict True Positives well). Another notable performer is the *TFID Vectorizer + Multinomial Naive Bayes* combination. Apart from the joint-second-highest AUC score of 0.84, its consistent performance on both the test and training sets showed that the model generalises well.

4.3 Running the Optimized Production model

Our production model is a combination of three models: Count Vectorizer, TF-IDF and Multinomial Naive Bayes.

The first one, Count Vectorizer, counts the most used words and displays the most used words from both the threads

Multinomial Naive Bayes classifier, assigns scores to the words (or in our case, the top 70 words) in our selected feature. Multinomial Naive Bayes will penalise a word that appears too often in the document.

A matrix of "word scores" is then transferred into a TF-IDF (or “Term Frequency – Inverse Document” Frequency) Vectorizer, which makes predictions based on the calculation of the probability of a given word falling into a certain category.

5. PROPOSED MODEL

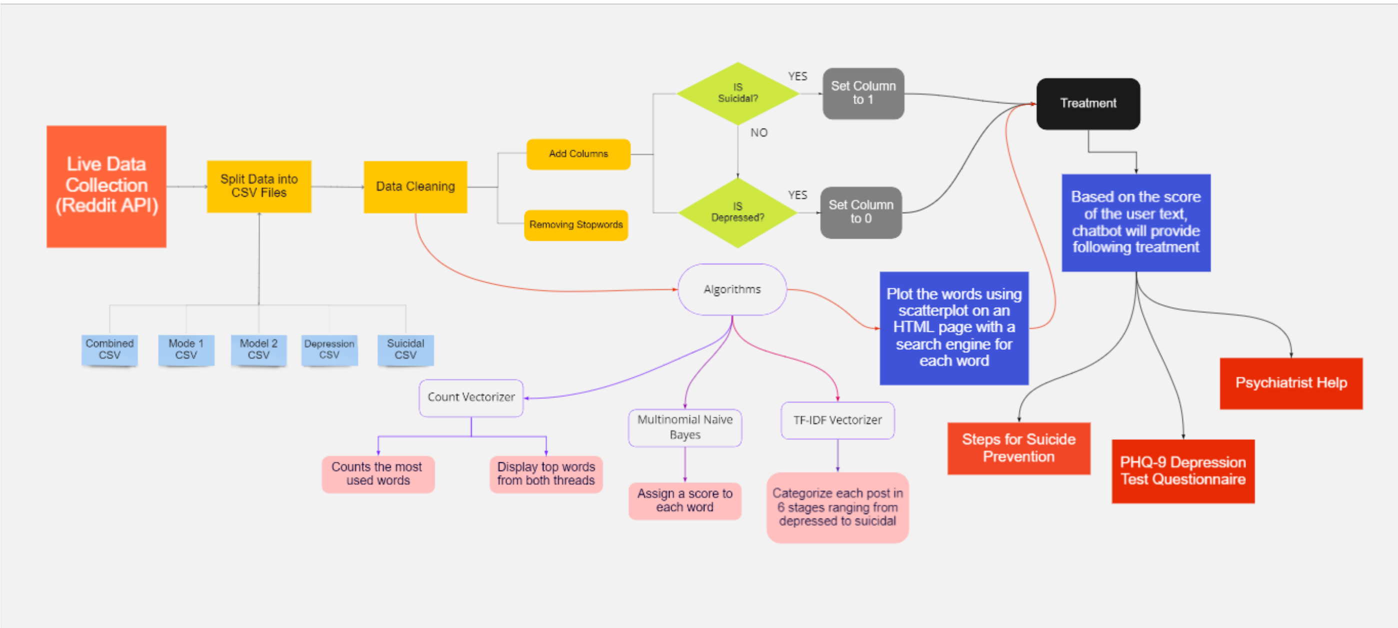

Fig -13: Final Production Model for Suicide Ideation Detection

To overcome the traditional static data set trained model we have used a dynamic data set that gets updated every time you run the program.

On reddit, we found two support communities for depression and suicide. A quick read through the posts reveals the subreddits to be genuine online spaces to seek help. And thus, good forums for us to yield honest text data about people's mental state.

These are our two subreddits and their tag-lines (which hint at their mission statements):

1. r/depression: because nobody should be alone in a dark place
2. r/SuicideWatch: Peer support for anyone struggling with suicidal thoughts.

After collecting the data from reddit we clean the data by removing the duplicate data entries and null columns. Then we combine the data from both threads into a csv file and then add a column named is_suicide having binary values 0 and 1 depending on the thread.

Then we remove the unnecessary stop words from the data. After that, using a count vectorizer we count the number of instances of a word and using a model we will assign a score to each word.

After that using scatterplot we will plot the words rendered on a html page. Then using an application, we will analyze the text taken from the user and based on the model we trained the score will be assigned and checked if the user is suicidal or not. Then we will suggest them suicide prevention helpline number or a nearest psychiatrist number or psychometric test.

6. RESULTS

6.1 Data Analysis Result

As a final step in our EDA, we used Scatter text to produce a user-friendly way of visualising our corpus in HTML.

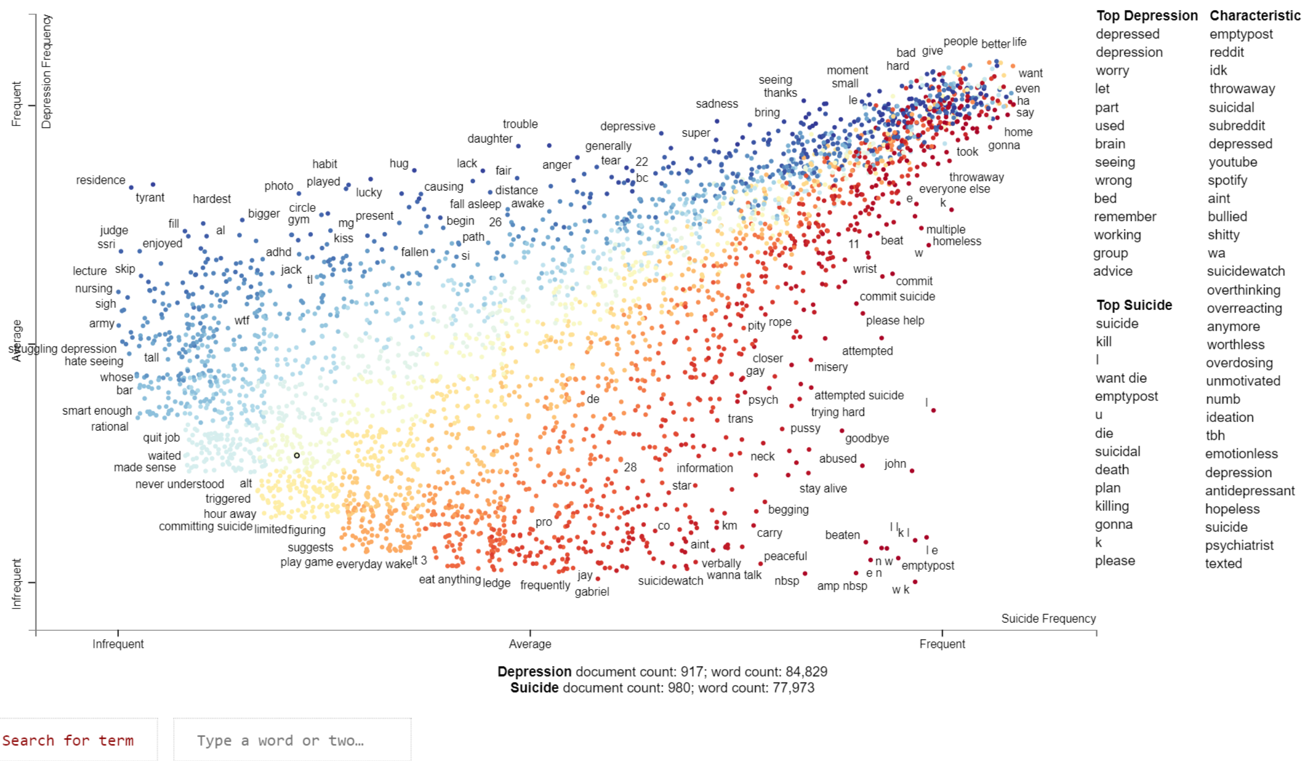

Fig -14: Data Analysis Corpus Scatter Plot

Fig -15: Search Engine of Production Model

7. CONCLUSION

Noise in the data - In analyzing the results of the model's application, we must first address the presence of some "noise" in the data. Our model was trained on posts on online support communities.

Six Stages – From the pure data entries, the model classified 78.8 in the suicide category. This level of accuracy means that the model might have the potential to be generalized for usage in institutions like schools, where students might be asked to fill in survey forms or meet the school counsellor. Textual data gathered from sessions like these might yield some revealing results about a student's suicidal tendencies

8. FUTURE WORK

Emerging Learning Techniques: Deep learning techniques have advanced to an extent to help in the research of suicidal ideation detection. Furthermore, advanced learning technologies like transfer learning, reinforcement learning, attention mechanisms, transfer learning, graph neural networks and many more can boost the research.

Suicidal Intention Understanding and Interpretability: There are many factors that contribute to suicide such as mental health, economic crisis, gun violence, alcoholism, heartbreak etc. A better understanding of these factors and how much big of a part they play individually can provide better understanding of suicidal intention and possible intervention.

Temporal Suicidal Ideation Detection: In future we can use temporal information as a method for detecting suicidal ideation. This will work in stages like stress, depression, suicidal thoughts and suicidal ideation and the model will provide a temporal trajectory and monitor the change of the individual's mental health through these stages which will eventually lead to early detection of suicidal ideation.

9. ACKNOWLEDGEMENT

I would like to take this opportunity to thank our internal guide Prof. Vasudha Phaltankar for giving us all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful. We are also grateful to Dr. Santosh Chobe, Head of Computer Engineering Department, Dr. D. Y. Patil Institute of Technology, Pimpri for his indispensable support and suggestions.

10. REFERENCES

[1] Marchant, A. et al. A systematic review of the relationship between internet use, self-harm and suicidal behavior in young people: the good, the bad and the unknown. PLoS ONE 12, e0181722 (2017).

[2] Ruder, Thomas & Hatch, Gary & Ampanozi, Garyfalia & Thali, Michael & Fischer, Nadja. (2011). Suicide Announcement on Facebook. Crisis. 32. 280-2. 10.1027/0227-5910/a000086.

[3] F. L. Lévesque, J. M. Fernandez and A. Somayaji, "Risk prediction of malware victimization based on user behavior," 2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE), 2014, pp. 128-134, doi: 10.1109/MALWARE.2014.6999412.

[4] Calear AL, Batterham PJ. Suicidal ideation disclosure: Patterns, correlates and outcome. Psychiatry Res. 2019 Aug; 278:1-6. doi: 10.1016/j.psychres.2019.05.024. Epub 2019 May 16. PMID: 31128420.

[5] Luoma JB, Martin CE, Pearson JL. Contact with mental health and primary care providers before suicide: a review of the evidence. Am J Psychiatry. 2002 Jun;159(6):909-16. doi: 10.1176/appi.ajp.159.6.909. PMID: 12042175; PMCID: PMC5072576

[6] Ahmedani BK, Simon GE, Stewart C, et al. Health care contacts in the year before suicide death. J Gen Intern Med. 2014;29(6):870-877. doi:10.1007/s11606-014-2767-3

[7] Frankenfield DL, Keyl PM, Gielen A, Wissow LS, Werthamer L, Baker SP. Adolescent patients--healthy or hurting? Missed opportunities to screen for suicide risk in the primary care setting. Arch Pediatr Adolesc Med. 2000 Feb;154(2):162-8. doi: 10.1001/archpedi.154.2.162. PMID: 10665603.

[8] Franco-Martín MA, Muñoz-Sánchez JL, Sainz-de-Abajo B, Castillo-Sánchez G, Hamrioui S, de la Torre-Díez I. A Systematic Literature Review of Technologies for Suicidal Behavior Prevention. J Med Syst. 2018 Mar 5;42(4):71. doi: 10.1007/s10916-018-0926-5. PMID: 29508152.

[9] Van Orden KA, Witte TK, Cukrowicz KC, Braithwaite SR, Selby EA, Joiner TE Jr. The interpersonal theory of suicide. Psychol Rev. 2010;117(2):575-600. doi:10.1037/a0018697

[10] Isometsä ET, Heikkinen ME, Marttunen MJ, Henriksson MM, Aro HM, Lönnqvist JK. The last appointment before suicide: is suicide intent communicated? Am J Psychiatry. 1995 Jun;152(6):919-22. doi: 10.1176/ajp.152.6.919. PMID: 7755124