UNIVERSITY OF BUCHAREST

FACULTY OF
MATHEMATICS AND
COMPUTER SCIENCE

COMPUTER SCIENCE DEPARTMENT

Dissertation Thesis

DECODING DEPRESSION: NLP-DRIVEN DETECTION IN SOCIAL MEDIA CONVERSATIONS

Graduate
Raluca-Maria Hanciu

Scientific coordinator
Lect.dr. Ana Sabina Uban

Bucharest, September 2024

Abstract

In a world where social media gives a peek into people's lives and minds, these platforms are increasingly essential for understanding and recognizing mental health disorders. As more people turn to social networks to express their thoughts and difficulties, there is a greater possibility to uncover connections that may indicate mental health issues such as depression. Understanding these issues early, particularly in virtual environments, can be a vital step toward prompt intervention and support. In this context, we tackle the first task of the eRisk Lab for 2024: analyzing Reddit comments to identify symptoms of depression. Our objective is to use information collected from Reddit's various user communities to identify sentiments associated with symptoms of depression listed in the BDI-II questionnaire. As a place where people may openly discuss ideas and experiences, Reddit provides a unique perspective on how people express themselves, including their difficulties with mental health. Through utilizing this content created by users, we hope to bring to light how depression presents itself in digital environments.

Rezumat

Intr-o lume in care social media actioneaza ca o ferestra catre vietile si mintile oamenilor, aceste platforme sunt din ce in ce mai importante pentru a intelege si identifica tulburarile de sanatate mintala. Pe masura ce tot mai multi oameni se indreapta spre social media pentru a isi expune gandurile si dificultatile, crește posibilitatea de a crea legaturi intre postarile utilizatorilor si problemele de sănătate mintală, cum ar fi depresia. Identificarea timpurie a acestor probleme, mai ales în mediul virtual, poate fi esențială pentru o intervenție și un sprijin rapid. În acest context, ne concentrăm pe primul obiectiv din cadrul eRisk Lab pentru 2024: analiza postarilor de pe Reddit pentru a identifica simptomele depresiei. Obiectivul nostru este să folosim informațiile colectate din diversele comunități de utilizatori de pe Reddit pentru a identifica sentimentele asociate simptomelor depresiei enumerate în chestionarul BDI-II. Reddit, ca spațiu unde oamenii discută deschis si anonim idei și experiențe, oferă o perspectivă unică asupra modului în care oamenii își exprimă trăirile, inclusiv dificultățile legate de sănătatea mintală. Prin analiza conținutului acestora sperăm să evidențiem modul în care depresia se manifestă în mediile digitale.

Contents

Chapter 1

Introduction

1.1 Problem statement

Millions of people worldwide suffer from depression, a widespread mental health illness that raises serious public health concerns [1]. The amount of cases of depression has increased in the fast-paced, globally connected society of today, where social demands and stressors are prevalent. Numerous causes, such as higher expectations at work, unstable economy state, loneliness, and the widespread effect of social media, are contributing to this trend [2, 3].

Early detection of depression is critical since it allows for prompt support and care for those who are affected. But conventional diagnosis techniques frequently depend on patients' self-reported symptoms during professional evaluations, which can be arbitrary and vulnerable to underreporting because of stigma or ignorance [4]. Because of this, a large number of depressed patients remain undetected or untreated, resulting in severe outcomes and ongoing suffering for the individuals.

The development of natural language processing (NLP) and artificial intelligence (AI) technology in recent years has opened up new possibilities for diagnosing depression and comprehending mental health problems [5]. Artificial intelligence (AI) algorithms are able to recognize patterns, attitudes, and language cues that are suggestive of sadness by analyzing the massive amounts of data collected on social media sites [5]. This method has many benefits, such as scalability, privacy, and real-time monitoring, which makes it especially useful for connecting with those who might not seek assistance through conventional channels.

Social media platforms function as digital forums where people can freely share their ideas, feelings, and life experiences [6]. Users frequently discuss their own setbacks, victories, and vulnerable moments—including their issues with depression. These online disclosures allow researchers to have access to plenty of data for researching depression and related illnesses. They are provided with insightful descriptions of the lived experi-

ences of people coping with mental health issues.

Through the use of artificial intelligence (AI) algorithms to examine social media information, researchers can find hidden patterns and possibly even diagnose depression. These realizations can guide the creation of support networks, preventative initiatives, and treatments that are suited to the requirements of those who are at risk [5]. Moreover, by providing scalable and affordable alternatives to conventional diagnostic techniques, AI-driven technologies have the potential to widen access to mental health care[5].

1.2 Contribution of the thesis

The CLEF (Conference and Labs of the Evaluation Forum) organization comprises an international community of researchers, practitioners, and organizations dedicated to advancing the field of information retrieval and related disciplines [7]. With a lengthy history covering several decades, CLEF has built a name for itself in the research community by planning conferences, workshops, and evaluation campaigns. Through the provision of a global collaborative platform, CLEF enables academics to share ideas, methodology, and datasets, hence promoting innovation and propelling advancements in the area [7].

eRisk is a vital component of the larger CLEF (Conference and Labs of the Evaluation Forum) group. It is a major project concentrating on risk assessment in virtual environments [8] [9]. ERisk, or Early Risk Prediction on the Internet, is a collaborative research project aimed at using digital data for mental health issues early identification and treatment [8]. It was founded in 2017. Using data from social media platforms, eRisk's annual shared tasks bring together researchers from different fields to create novel methods for spotting early indicators of mental health problems like depression, eating disorders, compulsive gambling, and self-harm. Through fostering collaboration, encouraging the implementation of automated detection methods, and ultimately working toward improving outcomes for those affected by these diseases, these tasks act as an encouragement for the advancement of the area of mental health research.

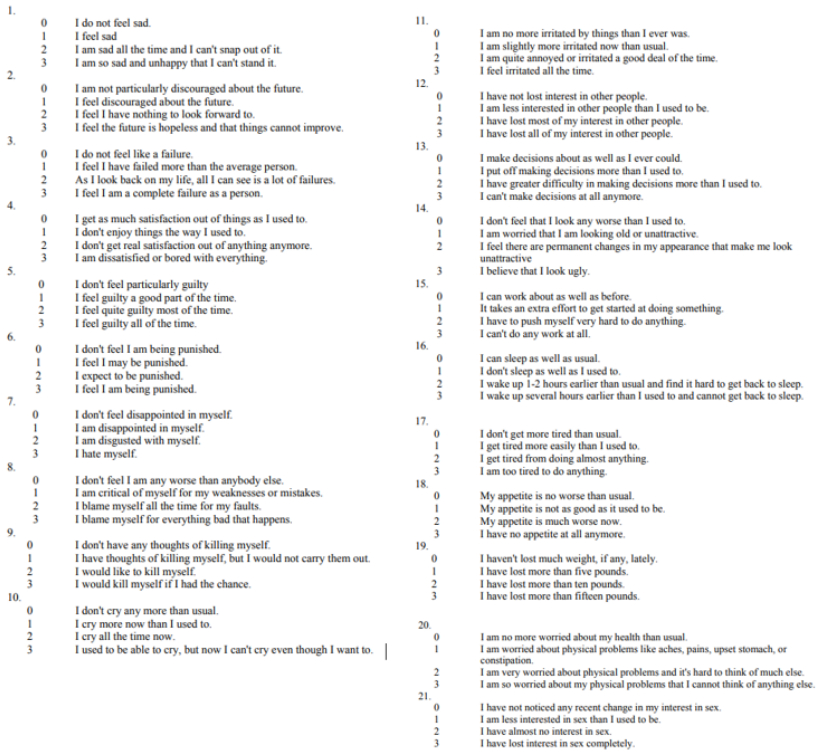
eRisk seeks to close the gap among research and practice by arranging shared tasks and provide a place for practitioners and researchers to work together to create efficient early risk detection. eRisk is dedicated to utilizing digital data in order to facilitate the creation of solutions that are both scalable and easily accessible, with the ultimate goal of improving the lives of those who are dealing with mental health concerns[8].

Under the eRisk system, we focus on the first task for 2024, which aims to identify depressive symptoms from user-generated content. This involves ranking sentences from a sample of Reddit user postings based on their relevance to depressive symptoms.The ones participating in this task will have to provide rankings for the symptoms comprised by Beck Depression Inventory (BDI) [8]. In total there are 21 symptoms in BDI.

The Beck Depression Inventory (BDI) is a widely used self-report questionnaire care-

fully constructed to assess the intensity of depression symptoms. Each one of the 21 items depicts a different depression symptom, and participants rate their level of experience with each symptom during a certain time period [10]. In the given setting of this task, a sentence is regarded as relevant to a BDI symptom when it offers insights about how the user is feeling concerning that symptom. Therefore, a sentence can still be considered significant despite the fact that it can suggest that the user does not have the symptom [8].

We can visualize the BDI Questionnaire in the next figure:



1.
0 I do not feel sad.
1 I feel sad
2 I am sad all the time and I can't snap out of it.
3 I am so sad and unhappy that I can't stand it.

2.
0 I am not particularly discouraged about the future.
1 I feel discouraged about the future.
2 I feel I have nothing to look forward to.
3 I feel the future is hopeless and that things cannot improve.

3.
0 I do not feel like a failure.
1 I feel I have failed more than the average person.
2 As I look back on my life, all I can see is a lot of failures.
3 I feel I am a complete failure as a person.

4.
0 I get as much satisfaction out of things as I used to.
1 I don't enjoy things the way I used to.
2 I don't get real satisfaction out of anything anymore.
3 I am dissatisfied or bored with everything.

5.
0 I don't feel particularly guilty
1 I feel guilty a good part of the time.
2 I feel quite guilty most of the time.
3 I feel guilty all of the time.

6.
0 I don't feel I am being punished.
1 I feel I may be punished.
2 I expect to be punished.
3 I feel I am being punished.

7.
0 I don't feel disappointed in myself.
1 I am disappointed in myself.
2 I am disgusted with myself.
3 I hate myself.

8.
0 I don't feel I am any worse than anybody else.
1 I am critical of myself for my weaknesses or mistakes.
2 I blame myself all the time for my faults.
3 I blame myself for everything bad that happens.

9.
0 I don't have any thoughts of killing myself.
1 I have thoughts of killing myself, but I would not carry them out.
2 I would like to kill myself.
3 I would kill myself if I had the chance.

10.
0 I don't cry any more than usual.
1 I cry more now than I used to.
2 I cry all the time now.
3 I used to be able to cry, but now I can't cry even though I want to.

11.
0 I am no more irritated by things than I ever was.
1 I am slightly more irritated now than usual.
2 I am quite annoyed or irritated a good deal of the time.
3 I feel irritated all the time.

12.
0 I have not lost interest in other people.
1 I am less interested in other people than I used to be.
2 I have lost most of my interest in other people.
3 I have lost all of my interest in other people.

13.
0 I make decisions about as well as I ever could.
1 I put off making decisions more than I used to.
2 I have greater difficulty in making decisions more than I used to.
3 I can't make decisions at all anymore.

14.
0 I don't feel that I look any worse than I used to.
1 I am worried that I am looking old or unattractive.
2 I feel there are permanent changes in my appearance that make me look unattractive
3 I believe that I look ugly.

15.
0 I can work about as well as before.
1 It takes an extra effort to get started at doing something.
2 I have to push myself very hard to do anything.
3 I can't do any work at all.

16.
0 I can sleep as well as usual.
1 I don't sleep as well as I used to.
2 I wake up 1-2 hours earlier than usual and find it hard to get back to sleep.
3 I wake up several hours earlier than I used to and cannot get back to sleep.

17.
0 I don't get more tired than usual.
1 I get tired more easily than I used to.
2 I get tired from doing almost anything.
3 I am too tired to do anything.

18.
0 My appetite is no worse than usual.
1 My appetite is not as good as it used to be.
2 My appetite is much worse now.
3 I have no appetite at all anymore.

19.
0 I haven't lost much weight, if any, lately.
1 I have lost more than five pounds.
2 I have lost more than ten pounds.
3 I have lost more than fifteen pounds.

20.
0 I am no more worried about my health than usual.
1 I am worried about physical problems like aches, pains, upset stomach, or constipation.
2 I am very worried about physical problems and it's hard to think of much else.
3 I am so worried about my physical problems that I cannot think of anything else.

21.
0 I have not noticed any recent change in my interest in sex.
1 I am less interested in sex than I used to be.
2 I have almost no interest in sex.
3 I have lost interest in sex completely.

Figure 1.1: BDI Questionnaire

Source: BDI Questionnaire

The dataset used for this task was given by the eRisk team, making use of historical data of eRisk and structured into TREC formatted phrases assigned to each user. This work required around 4 million sentences collected from 3,107 individuals.

At the evaluation step, upon the submission of runs for every single participant, the eRisk team will use human assessors to derive relevant conclusions. By completing this exercise, users help to create useful insights regarding the manifestation of depressive symptoms in user-generated material [8]. This research not only improves our knowledge of mental health difficulties, but it also has practical benefits for enhancing information retrieval systems and assisting those in need. In this study, we show our approach to recognizing depressive symptoms using the eRisk Lab framework. Our methodology is built on several key elements, such as feature engineering, classification methods, and

semantic similarity tools.

In this research, we used the paraphrase-MiniLM-L12-v2 model to encode textual input into numerical representations, which served as the cornerstone for our prediction system. By integrating Logistic Regression (LogReg) and MLPClassifier (MLP) algorithms, we aimed to predict the relevance of textual entries to depression symptoms through fine-tuning and optimization.

We also added novel variables like cosine similarity as well as first-person indications to improve our predictive models. Cosine similarity assessed semantic similarity between textual entries [11] and reference responses, whereas first-person indicators captured the subjective part of user tales, hence increasing contextual relevance.

We aimed to provide a complete system for symptom identification by combining advanced feature engineering along with classification approaches. The objective of our work was to increase the accuracy and quality of our predictive models, helping to progress natural language processing in mental health research.

Chapter 2

Related work

The first task in eRisk 2024 is an extension of the one in 2023. In 2023, the goal has been similar to this year: score sentences from a selection of user writings based on their relevance to a depressive symptom [12]. The participants will have to rank the 21 indicators of depression from the BDI Questionnaire.

FormulaML's approach to eRisk 2023 included preprocessing and encoding the dataset with Sentence Transformers, namely the MiniLM-L3-v2 model. They used the BDI-II questionnaire as a query, computed cosine similarity scores, and applied weighted scoring to assess sentence relevance. This strategy allowed for a complete investigation of symptoms recognition and similarity assessment [13].

OBSER-MENH compared texts and symptoms using Sentence Transformers (ST), which convert them into fixed-sized vectors. They used pre-trained models such as BERT to build these vectors, which can represent phrases in a dense vector space. For example, the all-mpnet-base-v2 model converts phrases into 768-dimensional vectors, whereas the all-distilroberta-v1 model is trained on a large dataset using the distilroberta-base model. [14].

BLUE team used an approach inspired by current research to enhance data using LLMs, creating synthetic Reddit posts associated with every BDI-II symptom to increase dataset diversity. While they expected that adding more diverse data would improve results, their findings revealed that the model using original BDI-II responses outperformed the model using produced data. They emphasized the problem of the specificity of ChatGPT data and proposed improving prompts for semantically comparable but heterogeneous material. Despite this, they acknowledged the useful character of the created text and pointed out the potential of LLMs to generate mental health data for future research [15].

In 2023, Formula-ML received the top scores across all parameters, followed by OBSER-MENH and then the BLUE team. In total there were 10 participating teams [12].

As of August 2024, with the publication of working notes from all teams, I was able to compare my outputs against those of the other participants.

The APB-UC3M team investigated three approaches for detecting depressive symptoms in Reddit posts: semantic similarity models, RoBERTa Classifier Model, and an Ensemble Method [16].

For the Semantic Similarity Models all-MPNet-base-v2, all-MiniLM-L12-v2, and all-MiniLM-L6-v2, each sentence was numerically represented. Cosine similarity was used to determine how similar these representations were to annotated phrases in the training set. For example, a sentence that is quite similar to those expressing "Sadness" would be rated as such, with a relevancy score ranging from 1 to 10. However, these models can only categorize phrases as one symptom at a time, which limits their ability to handle multi-label scenarios in which sentences may refer to numerous symptoms. Overall, the Semantic Similarity Models proved to be the most effective approach, with the all-MiniLM-L12-v2 model achieving the highest metrics, particularly in precision at 10 (P@10). RoBERTa did not generalize as well, and the Ensemble Method underperformed [16].

The team named REBECCA used the bge-small-en-v1.54 Transformer model in order to perform sentence rankings which calculated embeddings and applied cosine similarity to match sentences with symptom-specific answers. Sentences were ranked based on the highest similarity score for each symptom [17]. Result refinement was done using GPT-4. Symptom-specific prompts were used to assess and filter out non-relevant sentences, resulting in 14,815 retained sentences out of the original set [17].

Chapter 3

Dataset structure

Regarding the dataset used in this study, it comprised two main folders: one containing the training data and the other the test data. Within the training data folder, a subfolder contained a total of 3107 TREC files, each file representing a distinct user. For instance, a snippet from user s_405 exemplifies the structure, wherein each text is uniquely associated with a 'docno'.

```
        <DOC>
                <DOCNO>s_405_1279_15</DOCNO>
                <TEXT>I feel like everything I thought about and cared about is
destroyed.</TEXT>
        </DOC>
        <DOC>
                <DOCNO>s_405_1279_16</DOCNO>
                <TEXT>I'm just lucky I have my Fiancee and have her love.</TEXT>
        </DOC>
```

Figure 3.1: Dataset structure

Furthermore, the training folder encompassed two CSV files containing subsets of 'docnos' and corresponding labels categorized by query. One CSV file documented labels assigned by the majority of annotators for each 'docno', while the other recorded consensus labels. In the former, a '1' label signified agreement among at least two out of three annotators regarding the text's relevance, whereas in the latter, a '1' indicated unanimous agreement among all three annotators. To facilitate data visualization, both sets of labels were consolidated into a single CSV file, which can be seen in the following figure.

	A	B	C	D	E
1	query	q0	docid	rel_majority	rel_consensus
2	1	0	s_405_1279_15	1	1
3	1	0	s_2519_356_0	0	0
4	1	0	s_2038_51_7	1	1
5	1	0	s_975_61_2	1	0
6	1	0	s_577_923_1	1	1
7	1	0	s_2146_559_0	0	0
8	1	0	s_2100_926_0	0	0

Figure 3.2: Combined CSV structure

The dataset features the following attributes:

- `query`: representing the query number associated with the BDI questionnaire;
- `Q0`: which retained a static value of '0' throughout and is irrelevant to the task;
- `Docid`: denoting the unique identifier associated with a text from a user's TREC files;
- `Rel_majority`: denoting the label provided by the majority of annotators (2/3);
- `Rel_consensus`: indicating a '1' label if all three annotators concurred on the text's relevance for the specified query

The CSV comprised a total of 21,581 rows, with approximately 1000 examples per query; however, the distribution of '1' and '0' labels varied across the 21 queries. In this csv there are in total 21,581 rows, with approximately 1000 examples per each query, however the proportion of labels 1 and 0 is not the same across all 21 queries. Notably, the aggregated texts from all TREC files in the training set totaled approximately 4.2 million, illustrating that the annotated data represented only a fraction of the entire corpus. This disparity is expected, given the substantial volume of texts that are unrelated to any of the queries such as the below example of user s_405:

```
        <DOC>
                <DOCNO>s_405_108_0</DOCNO>
                <TEXT> For me America has always represented a land that is sposed to be a
mix of people and cultures.</TEXT>
        </DOC>
```

Figure 3.3: Illustration of a non-contributing data point

Subsequently, a matching process was conducted between 'docnos' from the TREC files and 'docids' from the CSV to associate corresponding texts and generate a new CSV.

Additionally, a ‘difference’ column was introduced, wherein a ’1’ label denoted discrepancies between ’rel_majority’ and ’rel_consensus’ labels, facilitating an investigation into texts that were not considered relevant by all annotators.

query	q0	docid	rel_majority	rel_consensus	difference	TEXT
1	0	s_405_1279_15	1	1	0	I feel like everything I thought about and cared about is destroyed.
1	0	s_2519_356_0	0	0	0	I'm sad that you believe that dissent is pointless.
1	0	s_2038_51_7	1	1	0	Since I feel rejected i have been feeling sad.
1	0	s_975_61_2	1	0	1	I feel this emotional pain straight to my soul.
1	0	s_577_923_1	1	1	0	I am sad waiting.

Figure 3.4: Extract from the CSV file containing text entries corresponding to each document number

Annotations were highlighted in red when the ’rel_majority’ and ’rel_consensus’ labels were set to ’1,’. The red highlight was done independently for each of the two columns. Annotations in the ’difference’ column were colored in yellow when a ’1’ label was present, suggesting differences between the ’rel_majority’ and ’rel_consensus’ labels. This color-coded approach allowed for a thorough study of inputs that were collectively regarded relevant by all annotators, alongside with the cases where annotators’ views differed.

It is also important to note the inclusion of the test dataset folder, which has the same structure as the training dataset. This folder contains 553 TREC files, each with numerous sentences similar to the training dataset. These files are critical for evaluating the performance of models trained on the training dataset, since they allow us to analyze generalization capabilities and model efficacy on previously encountered data. This thorough assessment procedure guarantees the model’s predictive potential is strong and reliable, hence improving the general quality and validity of the study’s results.

Chapter 4

Data preprocessing

During the data preprocessing phase, many procedures were performed to verify that the textual data was clean and homogeneous. These approaches aim to quicken subsequent analysis while mitigating potential noise or abnormalities in the dataset. The following preprocessing approaches were used:

1. **Lowercase Text Data**: All text in the dataset has been converted to lowercase. This standardization process helps to remove differences caused by irregular capitalization, maintaining consistency throughout the dataset.

2. **Non-alphanumeric characters were eliminated from the dataset** to focus on meaningful textual material. This process helps to simplify text representations and focus on the important semantic information included inside the text.

3. **Long words (above 20 characters) were omitted from the dataset**. This choice was motivated by a desire to remove irrelevant information, such as excessively long URLs or character strings, that may not add meaningfulness to the analysis.

Furthermore, several columns deemed unnecessary for the research were removed from the dataset. These included ”q0”, ”rel_consensus”, ”highlight” and ”docid”. The decision to keep only the labels from the ”rel_majority” column was chosen to maintain consistency and reliability, especially in cases when the number of texts classified as ’1’ fluctuated dramatically across searches. A structured overview of the distribution of labels ’1’ and ’0’ per question, as defined by the ”rel_majority” column, was created to provide insight into the proportionality and distribution of labels across different searches. This analysis indicated disparities in the quantity of positive and negative labels across queries, pointing to potential issues and imbalances in the data. In the below table we can see the counts per query of each of the 2 labels.

Query	Label 1 Count	Label 0 Count
Query 1.0	319	791
Query 2.0	334	816
Query 3.0	304	669
Query 4.0	207	806
Query 5.0	143	686
Query 6.0	50	1029
Query 7.0	288	717
Query 8.0	174	898
Query 9.0	349	604
Query 10.0	320	663
Query 11.0	155	925
Query 12.0	168	909
Query 13.0	141	969
Query 14.0	144	923
Query 15.0	204	878
Query 16.0	351	587
Query 17.0	155	892
Query 18.0	224	760
Query 19.0	141	883
Query 20.0	222	811
Query 21.0	159	812

Figure 4.1: Per query count of each of the two labels

Following the completion of preprocessing steps, the cleaned dataset was saved in a new CSV file, and a new DataFrame was created to facilitate further analyses. This preprocessing method established a basis for reliable and accurate analysis, ensuring the integrity and validity of the study's results. In the below figure we can see a snippet of our datframe post preprocessing:

	query	rel_majority	TEXT
0	1.0	1.0	i feel like everything i thought about and cared about is destroyed
1	1.0	0.0	im sad that you believe that dissent is pointless
2	1.0	1.0	since i feel rejected i have been feeling sad
3	1.0	1.0	i feel this emotional pain straight to my soul
4	1.0	1.0	i am sad waiting

Figure 4.2: Snippet of the cleaned dataset

Chapter 5

Feature engineering

5.1 Cosine similarity

Capturing semantic nuances in natural language processing (NLP) presents a major challenge due to language's innate complexity and ambiguity [18]. To solve this difficulty, several techniques have been developed, each with its own set of strengths and limitations. One typical technique is to use pre-trained language models, that were developed and fine-tuned on large amounts of text data, to learn complicated syntax and semantic correlations [19].

Hugging Face offers such a pre-trained language model called paraphrase-MiniLM-L12-v2 [20]. This model distinguishes itself by focusing on paraphrase detection, which entails detecting pairs of phrases that contain the same idea but are expressed differently. By training on a variety of paraphrase data sets, the paraphrase-MiniLM-L12-v2 model has gained a thorough understanding of semantic equivalency, allowing it to build embeddings that successfully represent semantic similarities between text sections [20][21].

Contrasting to other selections, such as general-purpose language models like BERT, the paraphrase-MiniLM-L12-v2 model has numerous notable advantages. First of all, it's distinctive focus on paraphrase identification gives it a higher sensitivity to small semantic nuances, making it especially good at detecting differences in language use [21]. Furthermore, the model's lightweight architecture (MiniLM) allows for efficient computation and deployment, which renders it ideal for tasks demanding scalability and real-time processing.

Furthermore, the paraphrase-MiniLM-L12-v2 model performs well at developing concise but informative embeddings, which are required for subsequent tasks such as similarity computations [21]. By reducing complicated linguistic patterns to dense representations, the approach makes it easier to extract nuanced semantic elements from textual input, improving subsequent studies.

In the context of this dissertation study, the paraphrase-MiniLM-L12-v2 model was

chosen due to its unique capacity to capture semantic nuances efficiently. Using the model's experience in paraphrase identification, we want to produce embeddings that capture the semantic links inherent in textual data about mental health problems. This, in turn, allows us to extract important elements that provide deeper insights into the expressions of mental health illnesses in language, ultimately contributing to our comprehension of mental health in the age of technology.

The major goal of this newly designed feature is to compute cosine similarity scores for Beck Depression Inventory (BDI) symptoms and dataset sentences. Cosine similarity is a frequently employed measure in NLP that assesses the similarity of two vectors by calculating the cosine of their angle [11]. By computing cosine similarity scores, we hope to obtain the semantic similarity between BDI symptoms and dataset sentences, effectively reflecting the degree of resemblance between them.

The four alternative responses to each of the 21 BDI symptoms are encoded into dense embeddings using the paraphrase-MiniLM-L12-v2 model. These embeddings capture the semantic information contained in each response, converting textual data into high-dimensional vector representations [20]. The sentences are also encoded with paraphrase-MiniLM-L12-v2, and the cosine similarity of each text to its associated query responses is determined. In other words, for each paragraph, there would be four cosine similarities, one for each possible response to the query symptom. I have got the maximum out of these 4 similarities and used it as a feature, because a text would be deemed as relevant no matter how severe they experience the specific symptom.

The obtained cosine similarity scores are used as features to encode the semantic similarity among BDI symptoms and dataset sentences. Higher similarity scores imply a stronger semantic relationship between a dataset text and a BDI symptom response, whereas lower scores indicate a less semantic overlap. By encompassing these cosine similarity scores as features in later analyses, the algorithm allows for the extraction of useful insights regarding the appearance of mental health problems in textual data.

5.2 First person check

Apart from the fact that a sentence is relevant even if the person affirms they don't have that symptom, a text is only relevant if it is talking about the user only, so:

- A text such as "I feel sad lately" would be labeled as 1 for the first BDI query which is 'sadness', but
- A text such as "My sister is very sad"/ "that is sad", would be label 0 as it is not written from a subjective point of view

As this is an important aspect of our texts, I have created an additional feature called "first_person" which can get values of 0 or 1. If in that specific text there are pronouns

such as : ['i', 'me', 'my', 'mine', 'myself', 'im'], then it would be relevant for this new feature.

However, there are many sentences written in first person that actually are irrelevant for a specific query, such as this one: "im sure that symbolizes loneliness" labeled 0 for query 1, which denotes sadness. To balance this impediment, I have added a condition to this "first_person" feature to only be labeled as 1 if there any of those pronouns are present in the text AND also the cosine similarity is greater than 0.4 for that query. Here we can see below a snippet of how the dataset looks like now:

	query	rel_majority	TEXT	cosine_similarity	first_person
0	1.0	1.0	i feel like everything i thought about and car...	0.465106	1
1	1.0	0.0	im sad that you believe that dissent is pointless	0.219973	0
2	1.0	1.0	since i feel rejected i have been feeling sad	0.677007	1
3	1.0	1.0	i feel this emotional pain straight to my soul	0.666604	1
4	1.0	1.0	i am sad waiting	0.784951	1

Figure 5.1: Snippet of the dataset with the new added features

In summary, the feature engineering process used in this dissertation project makes use of powerful natural language processing (NLP) techniques, specifically the paraphrase-MiniLM-L12-v2 model from Hugging Face. By producing embeddings that capture semantic nuances, our model makes it easier to compute cosine similarity scores across Beck Depression Inventory (BDI) symptoms and dataset sentences. These scores function as informative features, providing insights into the appearance of mental health conditions in textual data. Furthermore, an additional feature titled "first_person" was developed to distinguish between first-person narratives about mental health problems and those that are not. By incorporating both cosine similarity and first-person indicators, this approach enhances our understanding of mental health issues symptoms in natural language.

Chapter 6

Approaches

6.1 Preliminary phase

In the beginning phase of my research, I undertook a thorough review of numerous machine learning models to establish a performance baseline. I specifically explored with models known for their computing efficiency, such as Logistic Regression, MLPClassifier, DecisionTreeClassifier, and ensemble models such as GradientBoostingClassifier, XGBClassifier, and LGBMClassifier. My motivation for choosing those models was for a couple of reasons: computational efficiency and the possibility to provide first glimpses into the dataset's features[22].

After assessing the performance of these models, I noticed considerable differences in performance metrics across different queries. This observation motivated further inquiry into the underlying causes of this variability. One potential explanation for this phenomena is the unequal distribution of positive labels (label 1) among queries. It is possible that particular queries have a disproportionately low number of samples categorized as 1 relative to others, affecting the models' predicted accuracy.

Furthermore, I reasoned that the semantic complexity of the textual data connected with each query may play an important role in determining the models' performance. Queries with unclear or complex semantics may represent a severe challenge to machine learning algorithms, limiting their capacity to recognize and extract relevant characteristics effectively. At this initial stage I have not yet created and used my two additional features: cosine_similarity and first_person.

For assessing the performance of the models I have implemented a function that provides the code's main functionality. This function receives a list of machine learning models as input and then groups the data by query, allowing for query-level analysis. Using train and test splitting, each group of data is divided into features (X) and target variables (y). Stratify was also used inside train and test splitting [23] to guarantee that the class distribution is consistent across both the training and testing sets, which is

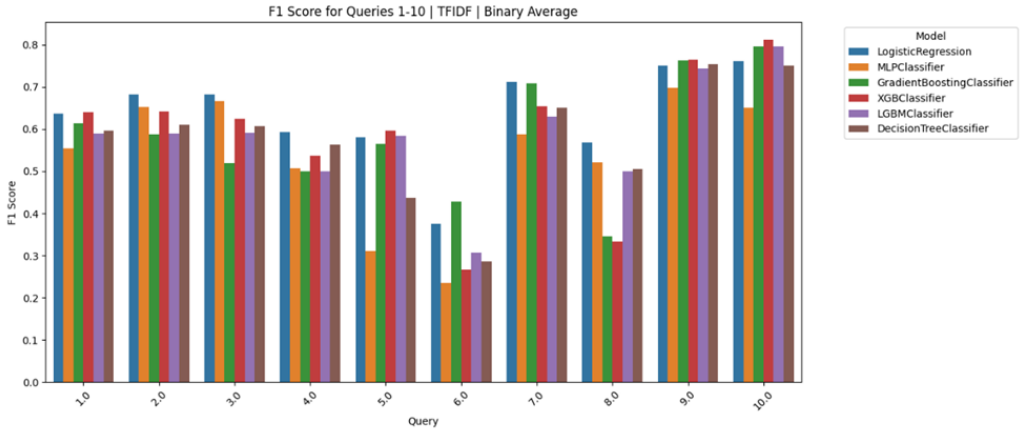
critical for proper evaluation.

Following splitting the data, the text is vectorized using TF-IDF [24]. This method converts textual data into numerical features that machine learning algorithms can understand [24]. The TF-IDF technique is widely incorporated in natural language processing tasks in order to effectively represent text data.

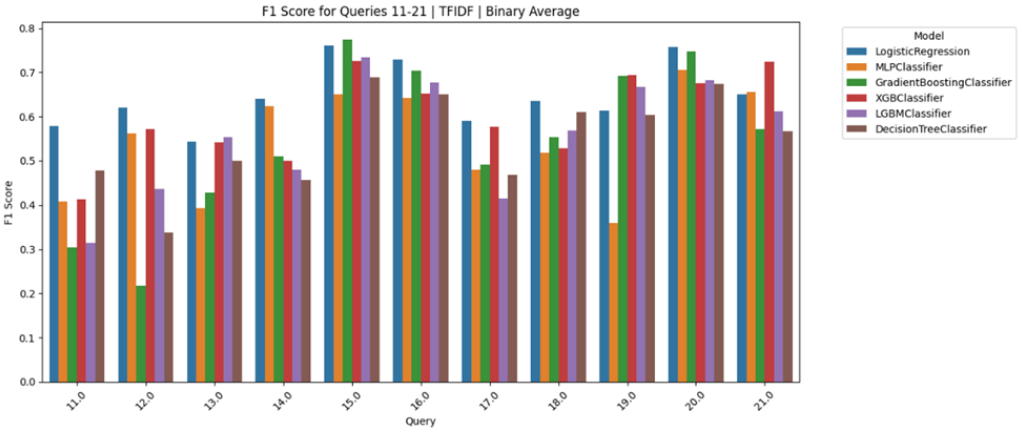
Next, the code iterates through each model in a predefined list. A classifier for each model is created and trained using the training data. Predictions are created based on the test data, and numerous evaluation metrics are calculated. These measurements include accuracy, precision, recall, and the F1 score. The code collects these information for each model and query, allowing for a full comparison of performance.

The evaluation metrics are added to a list, along with the query and model names. This list is then used to generate a DataFrame, which arranges the evaluation findings in an organized manner. This DataFrame is a useful resource for further study and comparison of model performance.

In figure 6.1 we can see a comparison of the models' performances:



(a) Performance Comparison of TF-IDF Across Multiple Models for Queries 1-10



(b) Performance Comparison of TF-IDF Across Multiple Models for Queries 11-21

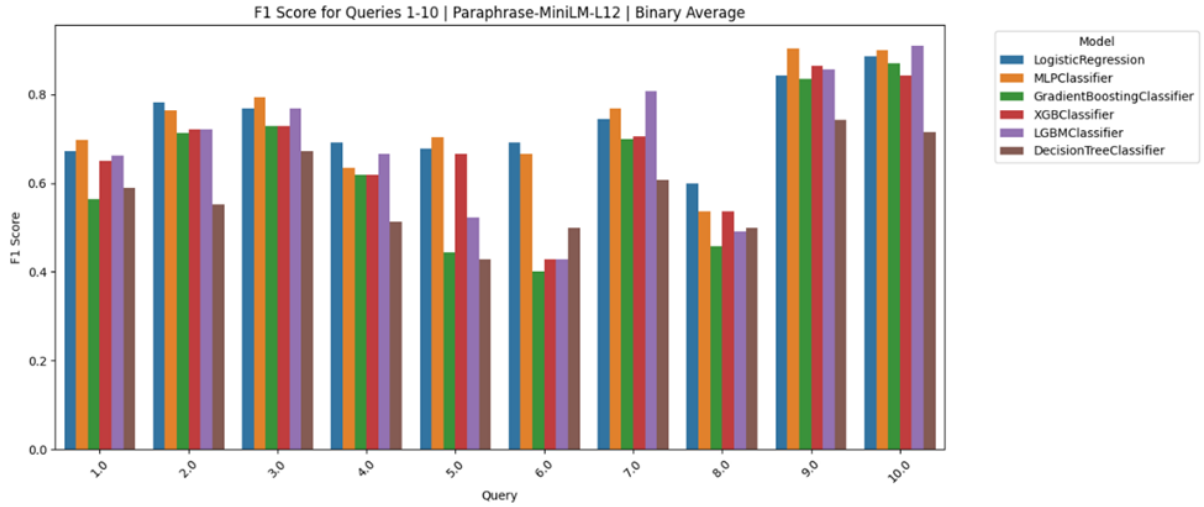
Figure 6.1: Performance Comparison of TF-IDF Across Multiple Models

As shown in the visual representations above, the f1_score varies significantly between different queries. The f1_score representations were carried out using only our positive label (label 1). Notably, Logistic Regression model ranks as the best performer, with reasonably consistent performance across multiple queries. In contrast, models such as GradientBoost and XGB exhibit significant performance variations, with instances of notably high scores in some queries (e.g., queries 15, 16) and significantly lower scores compared to other models in queries such as 8 and 12.

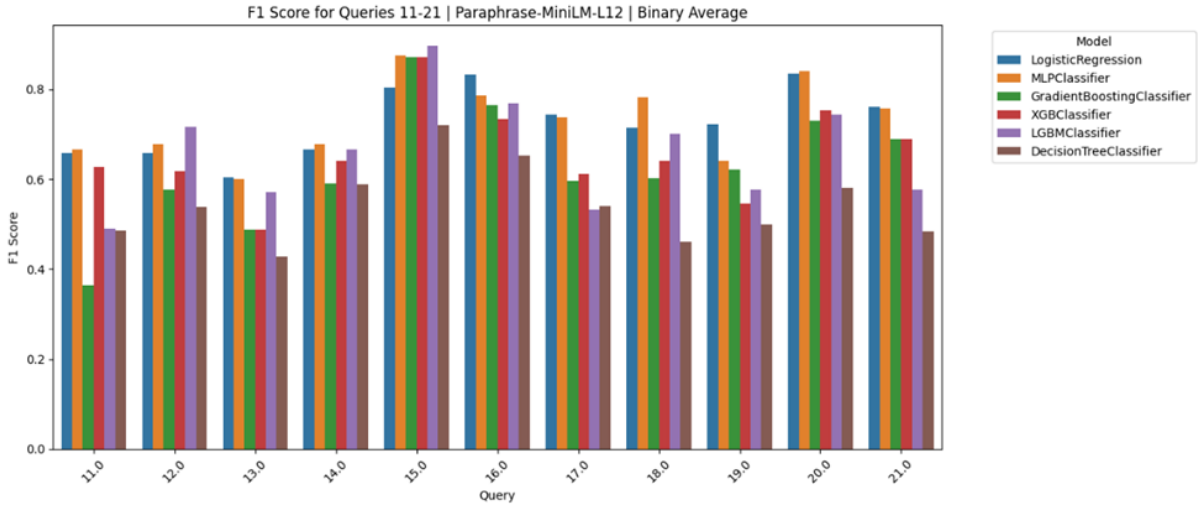
The initial baseline approach used typical TF-IDF vectorization to represent textual data. This method is well-established and commonly utilized, however it may be limited in detecting nuanced semantic similarities and contextual information in the text. The classification algorithms were trained directly on these TF-IDF representations, and while they are useful in many circumstances, they might have trouble with more complex patterns and relationships in the data.

To overcome these limitations and hopefully enhance results, I used a more advanced strategy. This method required using forefront transformer-based models, notably the Paraphrase-MiniLM-L12 model from the Sentence Transformers library, to create dense vector representations (embeddings) of the textual input. Using the transformer model, I hoped to capture richer semantic interpretations and contextual details in the text, which could lead to better performance in the classification tasks. This model was also used to create the cosine_similarity feature, which we discussed in the previous chapter.

In addition, as in the baseline approach, I used a variety of classifiers, including Logistic Regression, MLPClassifier, DecisionTreeClassifier, and ensemble approaches such as GradientBoostingClassifier, XGBClassifier, and LGBMClassifier, on these dense embeddings. This enabled the models to function on high-dimensional data, possibly capturing more complicated patterns and correlations than the TF-IDF based approach. The updated outcomes are shown below in Figure 6.2:



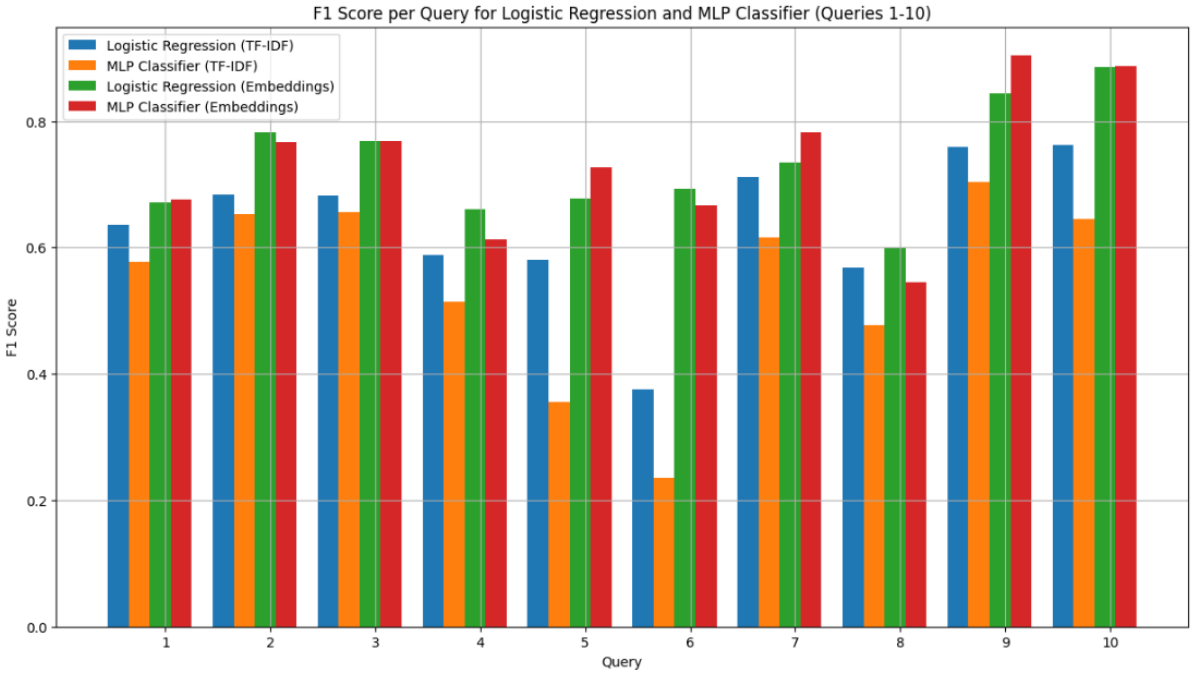
(a) Performance Comparison of encoding with paraphrase-multilingual-MiniLM-L12 Across Multiple Models for Queries 1-10



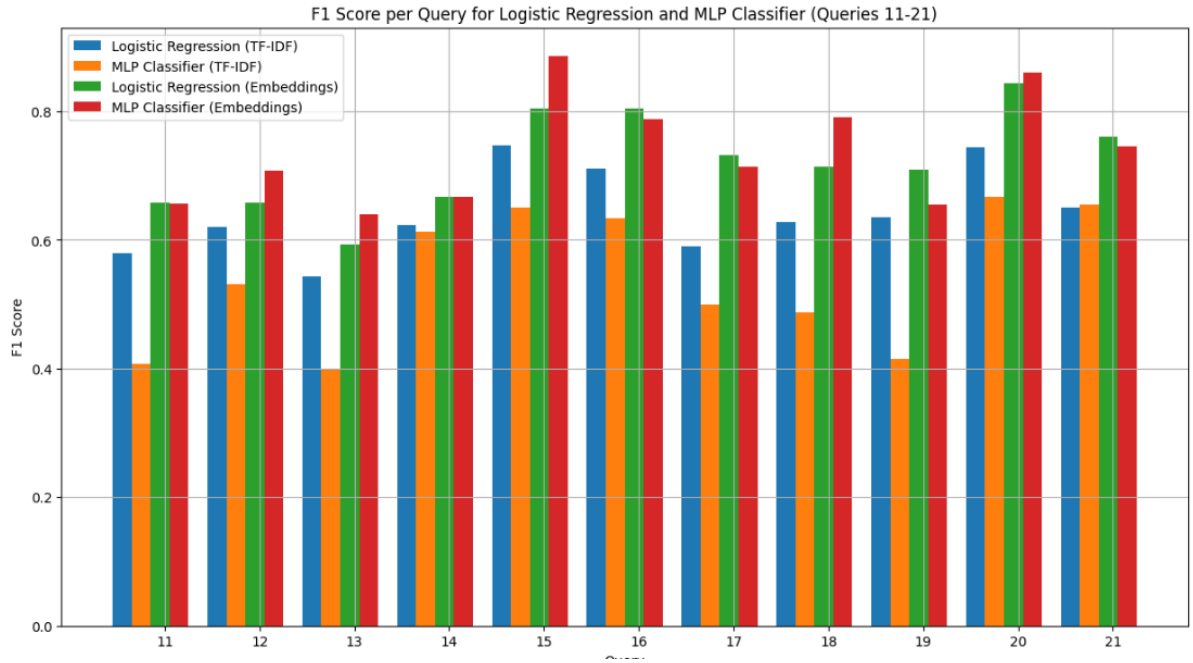
(b) Performance Comparison of encoding with paraphrase-multilingual-MiniLM-L12 Across Multiple Models for Queries 11-21

Figure 6.2: Performance Comparison of encoding with paraphrase-multilingual-MiniLM-L12 Across Multiple Models

The updated visualizations show an important shift in the top performer, with the MLPClassifier surfacing as the new leader. Both MLPClassifier and Logistic Regression consistently outperform other models in the majority of queries. The following plots show the improvements in Logistic Regression and MLPClassifier with Paraphrase-MiniLM-L12 encodings over TF-IDF.



(a) Performance Comparison of encoding with paraphrase-multilingual-MiniLM-12 vs TF-IDF | Logistic Regression and MLPClassifier | Queries 1-10



(b) Performance Comparison of encoding with paraphrase-multilingual-MiniLM-12 vs TF-IDF | Logistic Regression and MLPClassifier | Queries 11-21

Figure 6.3: Performance Comparison of encoding with paraphrase-multilingual-MiniLM-12 vs TF-IDF | Logistic Regression and MLPClassifie

The use of embeddings created using Paraphrase MiniLM 12 results in a noticeable improvement in model performance. With the incorporation of embeddings, both Logistic Regression and MLPClassifier make significant improvements, demonstrating the efficacy

of this embedding strategy in capturing semantic nuances in the data. MLPClassifier, in particular, experiences a significant uplift in performance as a result of this transition to embeddings. Given the fact that they are the top performers at this point in my research, I have decided to continue working with these two models in my future undertakings.

In the following stage, I included the two newly developed features (cosine_similarity, first_person) into my models to evaluate potential improvements in their performance. The differences in performance can be observed in figure 6.4 and figure 6.5:

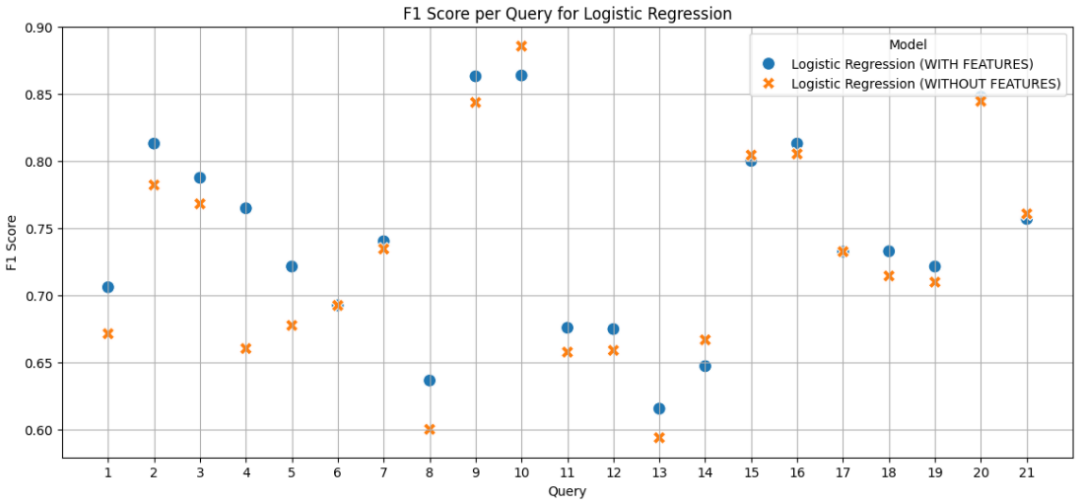


Figure 6.4: Logistic Regression Performance with/ without added features

The plotted graph aimed to evaluate the performance of Logistic Regression. In the blue points, we observe the F1 scores per query with the incorporation of two additional features, while the orange crosses represent the F1 scores attained without these features. Across most queries, there is an enhancement in performance with the inclusion of the new features. However, in queries 10, 14, 16, and 17, a marginal decline in F1 score is evident when compared to the performance without these features.

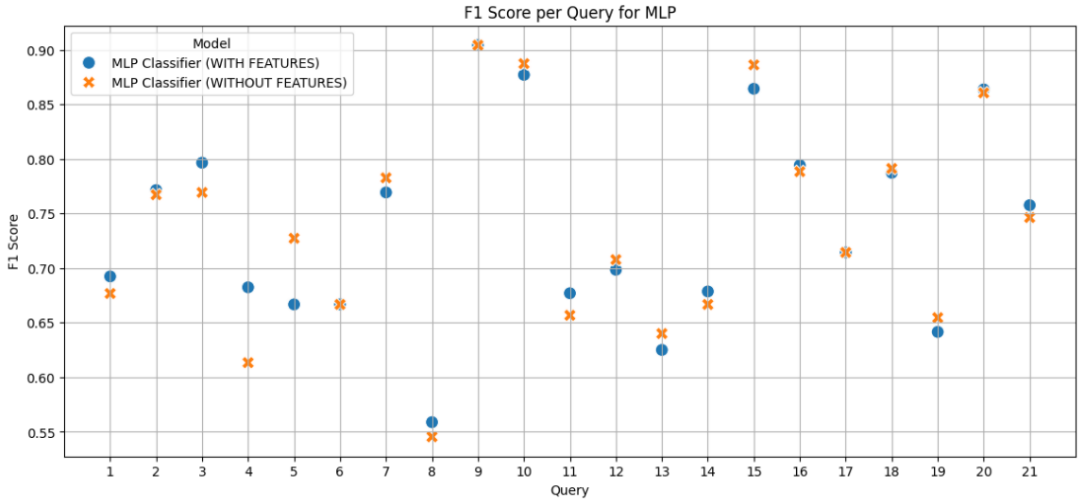


Figure 6.5: MLPClassifier Performance with/without added features

The plotted graph aimed to assess the performance of the MLPClassifier. In the blue points, we observe the F1 scores per query with the incorporation of two additional features, while the orange crosses represent the F1 scores attained without these features. The inclusion of additional features resulted in performance improvements across 11 queries. For 5 queries, the F1 score remained unchanged, while for the remaining 5 queries, there was a slight decrease in the F1 score.

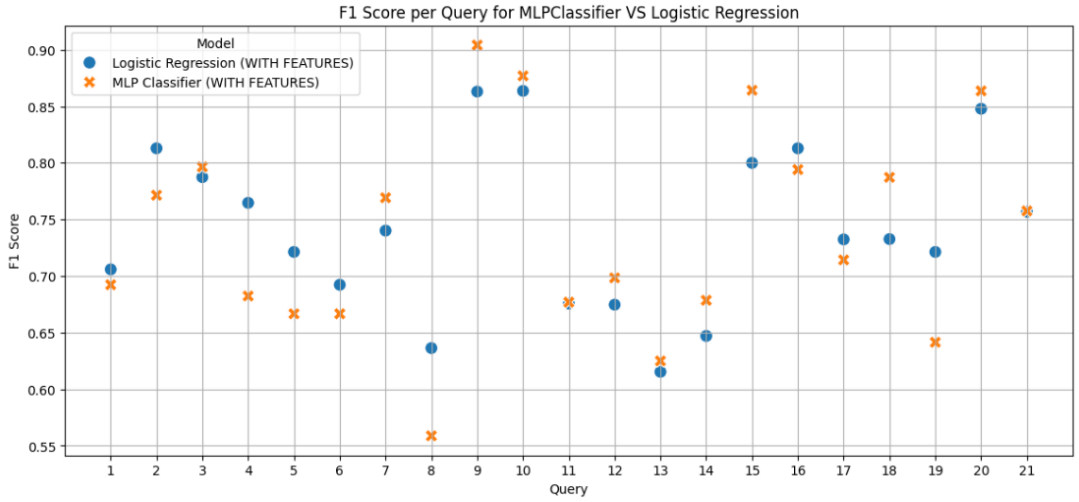


Figure 6.6: MLPCLassifier vs Logistic Regression Performance with added features

This above plot (Figure 6.6) presents a comparative analysis of the performance between Logistic Regression and MLPClassifier, incorporating the two additional features: cosine_similarity and first_person. MLPClassifier outperformed Logistic Regression on 11 queries, while Logistic Regression presented superior performance on 8 queries. Interestingly, the performance was comparable for both algorithms on 2 queries (16 and 17). Guided by these findings, I opted to proceed with the integration of the two features alongside the algorithms for further analysis.

6.2 Hyperparameters tuning

In the context of our research, the section on hyperparameters tuning is a crucial phase in which the prediction process takes priority.

A new prediction function was developed as a key component of our research process, with the goal of optimizing model parameters and assessing F1 scores across various queries in our dataset.

The function operates through each query, organizing and preprocessing the data to ensure it is ready for the future model training and evaluation phases.

When the function receives a query, it initiates a SentenceTransformer model and encodes the textual features into embeddings using paraphrase-multilingual-MiniLM-L12-v2. These embeddings, which include additional features such as cosine_similarity and first_person, improve the model's knowledge and allow for more detailed analysis in following phases.

The dataset has been split into training and testing subsets using cross-validation, which preserves class distributions. This splitting approach preserves the consistency of the evaluation process by reducing biases and increasing the reliability of the outcomes[25].

In each iteration, the model's hyperparameters are fine-tuned using a GridSearchCV pipeline. This pipeline systematically investigates a variety of configurations contained within the parameter grid [26], including for example for MLPClassifier parameters such as: hidden_layer_sizes, alpha, and learning_rate_init [27], and for LogisticRegression: C, penalty [28]. Its major goal is to determine the ideal parameters that result in the highest F1 results.

A key component of the pipeline is the RandomOverSampler, a mechanism used to solve class imbalance. This approach works by oversampling minority class instances within each fold, which corrects the dataset's disproportionate distribution of class labels [29]. By boosting the proportion of minority class samples, the RandomOverSampler helps ensure the model is trained on a more balanced dataset, lowering the risk of biased predictions and improving the classification model's overall performance.

During optimization, the pipeline checks metrics to determine the optimal model configuration for each query. It saves these configurations, including parameters and F1 scores, to provide insights into how different combinations perform across multiple searches.

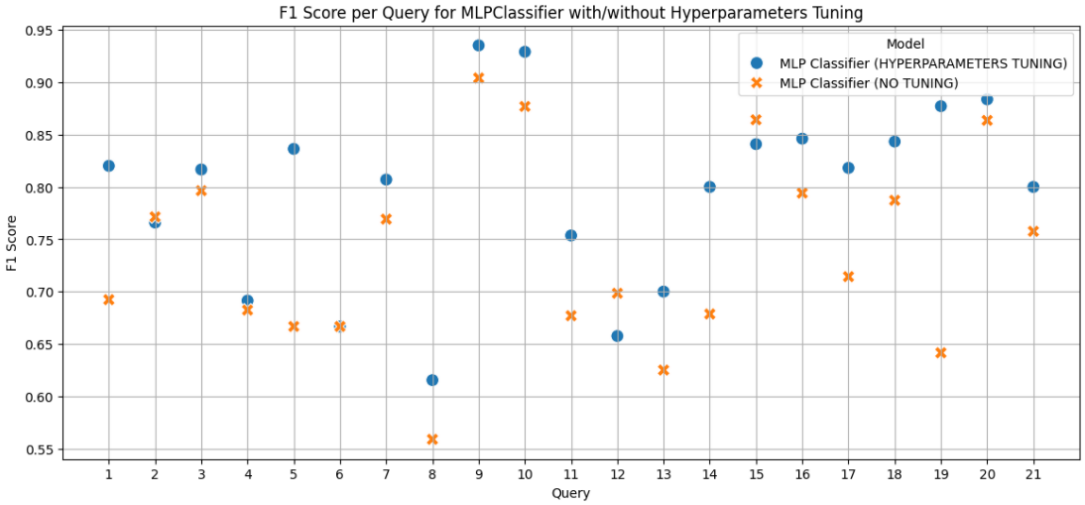


Figure 6.7: MLPCLassifier performance per query with/without hyperparameters tuning

The plotted graph (Figure 6.7) displays F1 scores for the MLPClassifier model with hyperparameter tuning, utilizing grid search, stratified k-fold cross-validation and Over-Sampling. The blue dots represent F1 scores achieved through this approach, while the orange dots depict F1 scores at the last checkpoint. Overall, this method notably enhanced results across most queries, except for queries 12 and 15. Particularly noteworthy is the substantial improvement observed in query 19, where the F1 score increased from 0.58 to 0.88. A comparable function to the one used for MLPClassifier was developed for Logistic Regression, and the outcomes are presented below:

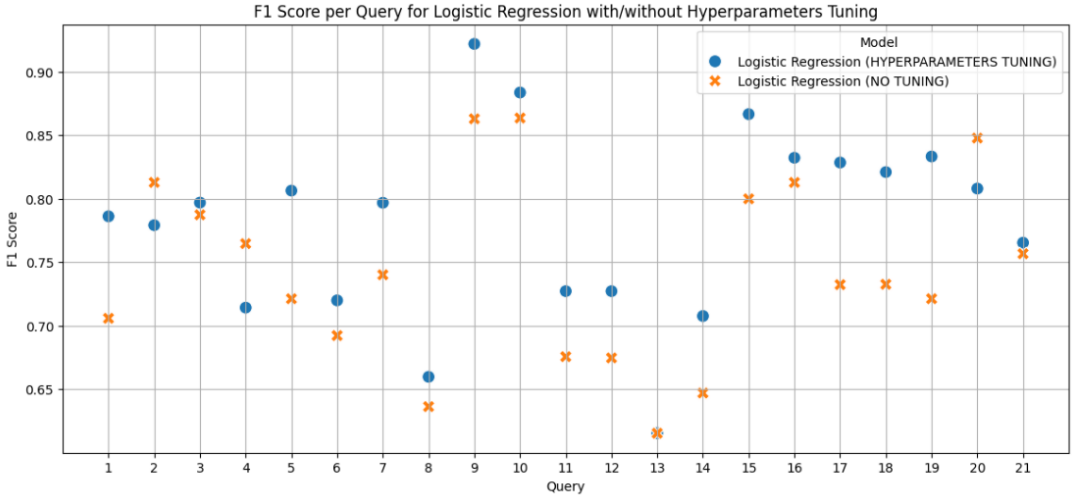


Figure 6.8: Logistic Regression performance per query with/without hyperparameters tuning

In the depicted plot, we can visually inspect the F1 scores per query conducted by Logistic Regression utilizing the latest updates, illustrated by the blue dots, in comparison with the performance of Logistic Regression at the last checkpoint, depicted in orange. Notably, an enhanced performance trend is evident across the majority of the queries, indicative of the efficacy of the implemented updates.

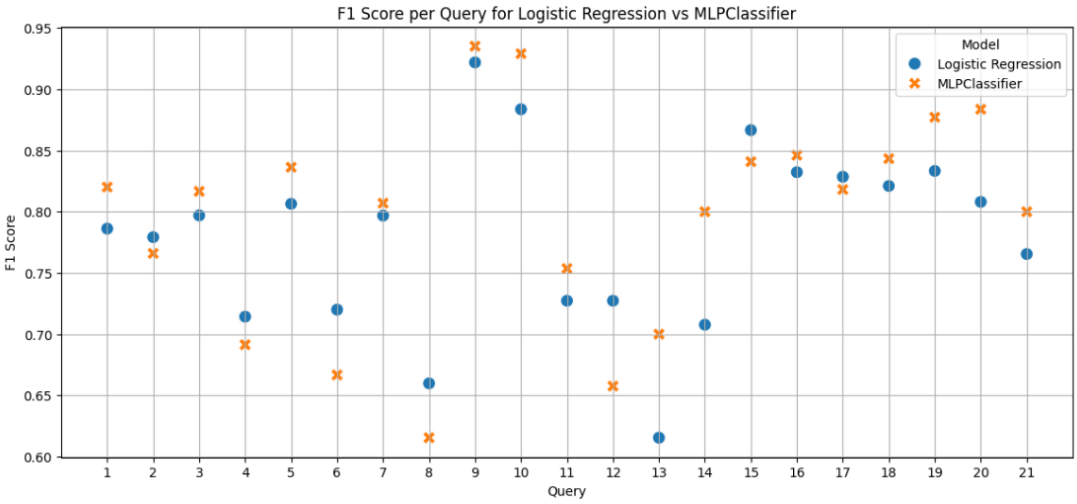


Figure 6.9: Performance comparison between Logistic Regression and MLPClassifier after Hyperparameters Tuning

A detailed understanding of how MLPClassifier and Logistic Regression perform differently for various queries may be obtained by comparing them (Figure 6.9). It seems that MLPClassifier performed better in 14 queries than Logistic Regression, demonstrating its higher effectiveness in identifying the underlying patterns in the data. On the other hand, it's important to notice that Logistic Regression performed better on questions where MLPClassifier had lower F1 values, such as questions 6, 8, and 12. Of particular interest are the highest and lowest performances achieved by both classifiers. For instance, MLPClassifier attained its highest F1 scores in queries 9 and 10, highlighting its proficiency in discerning subtle nuances within the data pertaining to suicidal ideation. Conversely, query 8 presented the lowest F1 scores for both classifiers, with Logistic Regression achieving a score of 0.66 and MLPClassifier attaining a marginally lower score of 0.62. The notable disparities in F1 scores can be attributed to the inherent ambiguity present in certain queries compared to others, compounded by the subjective interpretations of the annotators. For instance, query 8 addresses the theme of self-criticalness, where a positive label manifests as introspective but ambiguous statements such as:

- "I used to love myself and be proud of things I did."

In contrast, query 9, which delves into the realm of suicidal thoughts, exhibits more explicit language, such as

- "I feel extremely suicidal and I need to hurt myself."

The disparity in the specificity and clarity of language across queries contributes to variations in F1 scores, underscoring the multifaceted nature of text classification tasks.

Finally, I've preserved the models that demonstrated the highest F1 scores across all five folds for each individual query. Given the performance variations between Logistic Regression and MLPClassifier across different query contexts, I've made the decision to retain both models for future use in making predictions on the test dataset. This approach ensures that we can leverage the strengths of each model depending on the specific characteristics and complexities of the queries encountered during testing, thus optimizing our predictive capabilities and ensuring robustness in our analysis.

6.3 Predictions on test dataset

6.3.1 Preparing test data

The dataset for testing, as described in the 'dataset structure' segment of the thesis, comprises 553 TREC file folders, each corresponding to a unique user profile. Within these folders, a multitude of sentences is stored, with an aggregate count totaling approximately 15 million sentences across all files.

To obtain the test data utilized in this study, I first structured a function to facilitate natural sorting of filenames, ensuring an orderly processing sequence. Following this, I employed a function designed to rectify any invalid tokens present within the XML content, aiming to ensure data integrity. The data acquisition process involved iterating through all files within the designated folder, sorted in a natural order. Each XML file was parsed, and its contents read and processed to construct a well-formed XML document. Subsequently, I navigated through each document within the XML file, extracting relevant information such as document numbers, text content. This information was then organized into dictionaries and appended to a list for further processing. Upon completion, the accumulated data was structured into a DataFrame for subsequent analysis and examination.

A series of preprocessing techniques were used in the test dataset preparation phase to guarantee the uniformity and suitability of the data for additional analysis.

In order to begin this process, a basic step was to convert all textual content to lowercase, which would normalize the dataset's case sensitivity. After that, a refining process was carried out to ensure that the text contained just alphanumeric characters. This removed any special characters or unnecessary symbols that could have impeded the analysis. Additionally, a function was added to eliminate words longer than 20 characters in order to further improve the dataset's quality and relevance. This is because words with such elongated expressions usually don't include important information and can even add noise to the dataset.

Furthermore, a language detection mechanism was incorporated by utilizing the "langdetect" library's capabilities to identify and subsequently eliminate any data instances that were not marked in the English language [30]. This ensured the linguistic coherence of the dataset and allowed for a more targeted analytical attempt. Following these methods, all data points lacking textual information were eliminated. After undergoing these carefully designed preparation stages, the test dataset was thoroughly reformed, making it more effective and relevant for further analysis and interpretation.

	DOCNO	TEXT
0	0_0_0	i guess it depends on what cheating entails
1	0_0_2	our friendship thus far is completely platonic but i know if i had the opportunity guiltfree i would pounce on dat ass
2	0_0_3	i tried to break up with my sweet so saying that i just hadnt been feeling the same way i had when we met not a technical lie but he begged to make it work and i figured i would give it a shot because although the other guy is wide open i know he still has feelings for his ex
3	0_0_4	its been a year but its noticeable hes still not stable from it
4	0_0_5	although most wouldnt consider it cheating im harboring feelings for another guy
...	...	...
15113624	552315_0_27	and heres where im stuck guys
15113625	552315_0_28	i dont know what to do to in order to become financially independent
15113626	552315_0_29	im scared about talking this over with my boyfriend because any mention of my wanting to leave him makes him shut down
15113627	552315_0_30	i dont want to move back in with my parents because they have zero boundaries and im an adult with boundaries that need to be respected
15113628	552315_0_31	tldr my boyfriend is a jerk when he is sober i dont want to live with him anymore but i depend on him financially and i dont know what to do to become financially independent

15113629 rows × 2 columns

Figure 6.10: Snippet of preprocessed test dataset

In the image displayed above, we're provided with a snapshot depicting the current state of the test data post undergoing the series of preprocessing steps. Notably, the 'DOCNO' attribute stands out as it serves as a unique identifier assigned to each individual text entry within the dataset. This identifier aids in distinguishing and organizing the textual content effectively.

The text data used in this study was collected from Reddit, a social media platform. Although there is a lot of written content retrieved from this platform, a good portion of it may consist of informal conversations, irrelevant debates, or off-topic material that has little to do with the queries we are interested in. Such unrelated information might affect our results and make them harder to interpret.

Moreover, encoding and prediction would need a significant amount of time due to the large number of sentences. So, it would very much help us to reduce the number of irrelevant texts.

Here, we aim to address this problem by presenting a preprocessing approach that employs semantic similarity approaches in order to filter out irrelevant text input and retain only the entries that are relevant to our queries. In essence, we aim to ensure the quality of our dataset by eliminating text entries that don't truly correspond with the subjects we are studying.

In order to accomplish this, we will first use powerful Sentence Transformers models, such as the paraphrase-MiniLM-L3-v2, to encode the text data into numerical represen-

tations [31]. In this stage, the fundamental semantic meaning of each text fragment is captured by converting our raw text into high-dimensional numerical embeddings. Despite the fact that paraphrase-MiniLM-L12-v2 is the enhanced and more efficient model, the encoding process would require at least 40 hours. Considering the time limits, I have chosen to utilize paraphrase-MiniLM-L3-v2 for this phase. Next, for each text item, we'll compute the cosine similarity using a set of reference replies taken from the Beck Depression Inventory (BDI). Each query has four alternative answers, thus for every one of those, four cosine similarity are computed.

For the filtering part, we'll set a threshold for cosine similarity scores of around 0.39, and any text entry that falls below this threshold will be considered non-relevant. In the filtering process, text entries that are deemed irrelevant will be eliminated from our dataset. In this method, we ensure that we retain just the text elements that closely correspond to our query topics while striking a balance between inclusivity and specificity. This threshold is compared to the max cosine similarity per each query. Finally, we'll reconstruct our dataset using only the retained text entries, along with their corresponding similarity scores. This filtered dataset will be clean of any irrelevant text content, making it easier for our query-specific analyses.

After this process, the dataset would comprise about 2.6 Million texts, so we managed to filter out almost 13 million irrelevant texts.

6.3.2 Feature engineering

During the feature engineering process, I chose to use the same two features as in the training phase. While the paraphrase-MiniLM-L3-v2 model was sufficient for filtering out unnecessary text data, I chose to improve efficiency by using the paraphrase-MiniLM-L12-v2 model to compute cosine similarity as a feature. I intended to guarantee the model's performance was robust and reliable by preserving consistency in feature selection across both the training and testing phases.

	DOCNO	TEXT	query_1_cosine_similarity	query_2_cosine_similarity	...	query_20_cosine_similarity	query_21_cosine_similarity
0	0_0_7	im trying to work on breaking it off but my so...	0.291331	0.171185	...	0.307091	0.257844
1	1_0_1	i am completely heartbroken	0.624389	0.312791	...	0.189872	0.255011
2	1_0_4	i cant stop crying and ive just been playing b...	0.500549	0.314372	...	0.145921	0.100074
3	1_0_12	honestly it only came back to me after i found...	0.126384	0.034410	...	0.127855	0.203662
4	1_0_13	it said something like i miss you	0.314383	0.156961	...	0.003459	0.027799
...	...	...	...	...	...	...	...
2687192	552315_0_20	not everything he says is hurtful obviously bu...	0.295398	0.269662	...	0.208648	0.153561
2687193	552315_0_21	i try very hard not to say hurtful things to h...	0.201422	0.190251	...	0.201475	0.243794
2687194	552315_0_22	it hurts me when i say hurtful things	0.476589	0.239376	...	0.255252	0.087250
2687195	552315_0_24	i want to leave but im scared	0.443399	0.308547	...	0.201323	0.222708
2687196	552315_0_26	ive never been completely independent and i do...	0.349270	0.342689	...	0.293511	0.305569

2687197 rows × 7 columns

Figure 6.11: Test dataset with cosine similarity added feature for each of the 21 queries

In the above snippet we can observe how the dataset looks like now with the added cosine similarities per query, computed for each individual text.

Furthermore, I created 21 individual datasets, each with its own cosine similarity score according to the corresponding query. These datasets are designed to represent the 21 different queries, and the cosine similarity score provides a quantifiable measure of semantic concordance between the textual data and the query.

Following that, I calculated the first_person feature using the identical approach employed during the training phase. Specifically, the feature is assigned a positive label only when the cosine similarity exceeds 0.4, and it encompasses personal pronouns or expressions in the first person. This consistency ensures alignment with the methodology utilized during model training, thereby maintaining coherence across the feature engineering process.

Displayed below is the finalized form of the DataFrame corresponding to query 1, prepared and optimized for predictive analysis.

Dataframe for Query 1

	DOCNO	TEXT	cosine_similarity	first_person
0	0_0_7	im trying to work on breaking it off but my so...	0.291331	0
1	1_0_1	i am completely heartbroken	0.624389	1
2	1_0_4	i cant stop crying and ive just been playing b...	0.500549	1
3	1_0_12	honestly it only came back to me after i found...	0.126384	0
4	1_0_13	it said something like i miss you	0.314383	0

Figure 6.12: Test dataset with both cosine_similarity and first_person features

6.3.3 Predictions

In preparation for predictive analyses, the text data undergoes encoding using the paraphrase-MiniLM-L12-v2 model. To manage the computational complexity associated with large datasets, the dataset is segmented into smaller parts. This sequential processing approach facilitates efficient encoding, even when dealing with extensive datasets.

Following the encoding process, the embeddings are saved in separate files for each dataset section. This storage structure allows for easy access and administration of embeddings in following phases of research. The iterative structure of the encoding procedure guarantees that every part of the dataset is encoded sequentially. This methodical technique to processing vast amounts of data maintains the integrity and thoroughness of the encoding process.

Throughout the encoding process, progress updates are sent using tqdm to follow the status of the encoding [32]. These updates provide information on the currently processed segment as well as the overall progress toward completion. Such insights allow for efficient tracking and oversight of the encoding workflow, which facilitates the smooth operation of predictive analyses [32].

I've chosen to prioritize saving the encodings initially, as they will serve as the basis for generating predictions with both Logistic Regression and MLPClassifier.

First, I process the stored text embeddings files to prepare them for prediction. I loop through each file, load its embeddings, and append them to a list. After loading all embeddings, I concatenate them into a single array called X_train_features_text. This consolidated array contains all the necessary embeddings for prediction.

Next, I defined a function to load pre-trained model weights and make predictions. The load_model(query) function retrieves the pre-trained model weights corresponding to a specific query. For prediction I used another function that utilizes the pre-trained model to generate predictions and probabilities based on input features.

Further, the function called final_preds(df, query_number) concatenates the text embeddings with the other 2 features created and utilizes the pre-trained model to generate predictions and probabilities. The resulting predictions and probabilities are then appended to the dataframe.

Finally, I iterate over each query's dataframe, apply the prediction process using the defined functions, and save the updated dataframes with predictions. This systematic approach ensures that predictions are efficiently generated for each query's dataset, facilitating subsequent analysis and evaluation.

This process was done for both MLPClassifier and for Logistic Regression. Now I have 2 sets of 21 dataframes of the following format for both algorithms :

	DOCNO	TEXT	cosine_similarity	first_person	predictions	predictions_proba
0	0_0_7	im trying to work on breaking it off but my so...	0.291331	0	0.0	0.039000
1	1_0_1	i am completely heartbroken	0.624389	1	1.0	0.963118
2	1_0_4	i cant stop crying and ive just been playing b...	0.500549	1	1.0	0.531733
3	1_0_12	honestly it only came back to me after i found...	0.126384	0	0.0	0.200340
4	1_0_13	it said something like i miss you	0.314383	0	0.0	0.006050
...	...	...	...	...	...	...
2687192	552315_0_20	not everything he says is hurtful obviously bu...	0.295398	0	0.0	0.001008
2687193	552315_0_21	i try very hard not to say hurtful things to h...	0.201422	0	0.0	0.200213
2687194	552315_0_22	it hurts me when i say hurtful things	0.476589	1	0.0	0.136512
2687195	552315_0_24	i want to leave but im scared	0.443399	1	1.0	0.576853
2687196	552315_0_26	ive never been completely independent and i do...	0.349270	0	0.0	0.073140

Figure 6.13: Example of 1 of the 21 datasets(for each query) with predictions

6.3.4 eRisk submission results

Each participant for this task at erisk had the possibility to send up to 5 submissions of the results[8]. Participants are required to submit up to 1000 results sorted by estimated relevance for each of the 21 symptoms and they must follow the below format in their submissions:

```
1       Q0      sentence-id-121 0001    10      myGroupNameMyMethodName
1       Q0      sentence-id-234 0002    9.5     myGroupNameMyMethodName
1       Q0      sentence-id-345 0003    9       myGroupNameMyMethodName
...
21      Q0      sentence-id-456 0998    1.25    myGroupNameMyMethodName
21      Q0      sentence-id-242 0999    1       myGroupNameMyMethodName
21      Q0      sentence-id-347 1000    0.9     myGroupNameMyMethodName
```

Figure 6.14: Submission format for eRisk participants
Source: https://erisk.irlab.org/

In this format, we have the first column representing the number of the symptom (query), Q0 (unchanged variable across submissions), sentence_id meaning the document number in our dataframe, the position in the ranking, the relevance score corresponding to each sentence, and finally the name of the method/system used in that submission [8]. After the submission phase is over, the submitted runs will be subjected to relevance assessments made using traditional pooling techniques with human assessors.

These judgments will then serve as the basis for evaluating systems using conventional ranking metrics. The metrics used were the following:

1. **Average Precision (AP):** A typical statistic in information retrieval to assess the accuracy of ranked retrieval results is Average Precision (AP). It considers both the relevance and the rank of each retrieved document to determine the average precision for each relevant document in the ranked list [33].

2. **Precision at K (P@K):** P@K, or precision at K, assesses the percentage of pertinent documents among the top k documents that were retrieved [34]. It sheds light on how well a retrieval system performs when returning results inside the first K items of the ranked list [34].

3. **R-Precision (R-PREC):** R-Precision (R-PREC) is a precision version that only takes into account the top R retrieved documents, where R is the total number of documents that are relevant to a given query [35]. Compared to conventional precision, retrieval accuracy is measured more strictly.

4. **Normalized Discounted Cumulative Gain (NDCG):** The Normalized Discounted Cumulative Gain (NDCG) ranking metric assesses the quality of a ranked list of documents by taking into account the order in which relevant documents appear in the list as well as their relevance [36]. With a range of 0 to 1, it is a normalized version of the Discounted Cumulative Gain (DCG) statistic that indicates the success of retrieval.

I submitted three entries structured as follows:

1. The first submission served as a benchmark to gauge the performance of my models. For this submission, I exclusively relied on the cosine similarity scores generated by the paraphrase-MiniLM-L12-v2 model. These scores were arranged in descending order for each query, and only the top 1000 scores per query were considered.

2. In the second submission, I utilized the probability scores outputted by my algorithms for each query. Depending on the query, I selected either MLPClassifier or Logistic Regression, based on their respective performance during the training phase. Subsequently, I computed the mean of the cosine similarity score and the algorithm's probability score per query.

3. The third and final submission employed both MLP and Logistic Regression models per query, determined by their superior performance during training. However, a distinctive feature of this submission was the utilization of different thresholds to generate scores:

    - For queries with an F1 score ranging between 0.60 and 0.75, the new relevance score was computed as:

    $$0.3 \times \text{prediction\_probability} + 0.7 \times \text{cosine\_similarity\_score}$$

    - Queries with an F1 score between 0.75 and 0.85 utilized a relevance score

calculated as:

$$0.5 \times \text{prediction\_probability} + 0.5 \times \text{cosine\_similarity\_score}$$

- For queries achieving an F1 score between 0.85 and 0.95, the relevance score was determined as:

$$0.7 \times \text{prediction\_probability} + 0.3 \times \text{cosine\_similarity\_score}$$

In the below figures we can observe the results received from eRisk team:

Table 6.1: Ranking-based evaluation for Task 1 (majority voting)

Team	**Run**	**AP**	**R-PREC**	**P@10**	**NDCG@1000**
MindwaveML	MindwaveMLMiniLML12MLP_weighted	0.159	0.240	0.567	0.396
MindwaveML	MindwaveMLMiniLML12MLP_0.5	0.149	0.231	0.538	0.378
MindwaveML	MindwaveMLMiniLML12	0.133	0.212	0.490	0.330
Official Best results					
NUS-IDS	Config_5	**0.375**	**0.434**	0.924	**0.631**
APB-UC3M	APB-UC3M_sentsim-all-MiniLM-L6-v2	0.354	0.391	**0.986**	0.591

Table 6.2: Ranking-based evaluation for Task 1 (unanimity voting)

Team	**Run**	**AP**	**R-PREC**	**P@10**	**NDCG@1000**
MindwaveML	MindwaveMLMiniLML12MLP_weighted	0.158	0.238	0.471	0.427
MindwaveML	MindwaveMLMiniLML12MLP_0.5	0.147	0.227	0.457	0.408
MindwaveML	MindwaveMLMiniLML12	0.128	0.203	0.410	0.360
Official Best results					
NUS-IDS	Config_5	**0.392**	**0.436**	0.795	**0.692**
MeVer-REBECCA	TransformerEmbeddings_CosineSimilarity_gp	0.305	0.357	**0.833**	0.551

In the presented tables we can see the outcomes of our team compared with the results of the team/teams that performed best, determined through voting by the annotators. Table 6.1 delineates the system performance rankings derived from a majority voting perspective, while Table 6.2 provides rankings based on unanimity among all three annotators. Our team, identified as 'MindwaveML', is included in these assessments.

Upon comparing the proposed methodologies, notable performance enhancements were observed with the approach integrating varied weights for cosine similarity and the probability score generated by the classification algorithm. Notably, the system named "MindwaveMLMiniLML12MLP_weighted" exhibited the highest performance, closely followed by "MindwaveMLMiniLML12MLP_0.5," which computed the mean between the probability score and cosine similarity. In contrast, "MindwaveMLMiniLML12" served as our comparative baseline. These findings affirm that while the improvements were not groundbreaking, the approach demonstrated efficacy by yielding higher scores in comparison to the baseline method.

Chapter 7

Conclusions

This thesis provides a thorough investigation of techniques for assigning a sentence's relevance to depressive symptoms, as defined by the eRisk task. The main objective was to create and assess machine learning models that could efficiently identify meaningful text from user-written content on symptoms of depression, enabling more effective information retrieval procedures.

The eRisk task required ranking phrases by assessing their relevance to 21 depressive symptoms obtained from the Beck Depression Inventory (BDI) questionnaire. Participants were provided with a TREC formatted sentence-tagged dataset as well as the BDI questionnaire, and they were given the task of developing techniques to derive queries from the questionnaire to rank the sentences accordingly..

Numerous classification models were created and assessed in order to tackle this problem. These models integrated a variety of features, obtained through preprocessing and feature engineering procedures, such as cosine similarity scores and indications for first-person pronouns. In order to maximize performance, the logistic regression and MLP-Classifier algorithms were employed and fine-tuned to optimize performance.

In the evaluation stage people uploaded TREC formatted runs to evaluate sentences according to estimated relevance to each symptom. The systems were assessed using conventional ranking measures such Average Precision (AP), R-Precision (R-PREC), Precision at K (P@K), and Normalized Discounted Cumulative Gain (NDCG). Relevance assessments were derived utilizing standard pooling procedures with human assessors.

Results from the evaluation process revealed insights into the performance of the developed computational models. Notably, models incorporating a weighted combination of cosine similarity scores and probability outputs from classification algorithms demonstrated superior performance compared to baseline approaches.

Future study could concentrate on a number of areas to improve information retrieval systems' functionality and performance for tasks involving the ranking of depression symptoms. Feature engineering is one area that might benefit from some improvements. Further research into more complex methods for obtaining significant characteristics from

textual data may result in more accurate depictions of the underlying semantics.

Observing the performance of other teams, it's clear that my approach might have improved with an alternative strategy. The higher performance of models that use extensive semantic similarity techniques, such as all-MiniLM-L12-v2, demonstrates how important the use of transformer models is in achieving high scores. These models excel at extracting semantic information, which is vital for tasks such as recognizing depressive symptoms in textual data. While my approach of encoding with paraphrase-MiniLM-L12-v2 and using ML models for predictions provided a good basis, it may have been less effective than the direct application of transformer models and semantic similarities.

Furthermore, combining the above-mentioned methods with result refining modern language models like GPT-4 might have considerably increased the performance.

Finally, this thesis adds to the continuing issue of textual data analysis and information retrieval. It offers extensive approaches, methods, and computational models. This paper contributes to the creation of resilient and efficient information retrieval systems by understanding the complex nature of the preprocessing, feature engineering, and model evaluation procedures.

Bibliography

[1] World Health Organization. *Depressive disorder (depression)*. 2023. URL: https://www.who.int/news-room/fact-sheets/detail/depression.

[2] J. F. Greden. "The burden of recurrent depression: causes, consequences, and future prospects." In: *Journal of Clinical Psychiatry* 62.Suppl 22 (2001), pp. 5–9.

[3] Simone Cunningham, Chloe C. Hudson, and Kate Harkness. "Social Media and Depression Symptoms: a Meta-Analysis." In: *Research on Child and Adolescent Psychopathology* 49.2 (Feb. 2021), pp. 241–253. ISSN: 2730-7174. DOI: 10.1007/s10802-020-00715-7. URL: https://doi.org/10.1007/s10802-020-00715-7.

[4] A. Handy, R. Mangal, T. S. Stead, R. L. Jr Coffee, and L. Ganti. "Prevalence and Impact of Diagnosed and Undiagnosed Depression in the United States." In: *Cureus* 14(8) (Aug. 2022). DOI: 10.7759/cureus.28013.

[5] F. Zafar, L. Fakhare Alam, R. R. Vivas, J. Wang, S. J. Whei, S. Mehmood, A. Sadeghzadegan, M. Lakkimsetti, and Z. Nazir. "The Role of Artificial Intelligence in Identifying Depression and Anxiety: A Comprehensive Literature Review." In: *Cureus* 16(3) (Mar. 2024). DOI: 10.7759/cureus.564723.

[6] Andreas Munzel and Werner Kunz. "Sharing Experiences via Social Media as Integral Part of the Service Experience." In: *SSRN Electronic Journal* (Jan. 2013). DOI: 10.2139/ssrn.2307120.

[7] *CLEF 2024 Conference and Labs of the Evaluation Forum*. 2024. URL: https://clef2024.imag.fr/.

[8] J. Parapar, P. Martín Rodilla, D. E. Losada, and F. Crestani. "Overview of eRisk 2024: Early Risk Prediction on the Internet." In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 15th International Conference of the CLEF Association, CLEF 2024*. Grenoble, France: Springer International, 2024.

[9] Javier Parapar, Patricia Martín-Rodilla, David E. Losada, and Fabio Crestani. "Overview of eRisk 2024: Early Risk Prediction on the Internet (Extended Overview)." In: *Working Notes of the Conference and Labs of the Evaluation Forum CLEF 2024, Grenoble, France, September 9th to 12th, 2024*. CEUR Workshop Proceedings. 2024.

[10] Jane Upton. "Beck Depression Inventory (BDI)." In: *Encyclopedia of Behavioral Medicine*. Ed. by Marc D. Gellman and J. Rick Turner. New York, NY: Springer New York, 2013, pp. 178–179. ISBN: 978-1-4419-1005-9. DOI: 10.1007/978-1-4419-1005-9_441. URL: https://doi.org/10.1007/978-1-4419-1005-9_441.

[11] Baoli Li and Liping Han. "Distance Weighted Cosine Similarity Measure for Text Classification." In: *Intelligent Data Engineering and Automated Learning – IDEAL 2013*. Ed. by Hujun Yin, Ke Tang, Yang Gao, Frank Klawonn, Minho Lee, Thomas Weise, Bin Li, and Xin Yao. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 611–618. ISBN: 978-3-642-41278-3.

[12] Javier Parapar, Patricia Martín-Rodilla, David E. Losada, and Fabio Crestani. "Overview of eRisk 2023: Early Risk Prediction on the Internet." In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 14th International Conference of the CLEF Association, CLEF 2023*. Thessaloniki, Greece: Springer International Publishing, 2023.

[13] N. Recharla, P. Bolimera, Y. Gupta, and A. K. Madasamy. "Exploring Depression Symptoms through Similarity Methods in Social Media Posts." In: *CLEF (Working Notes)*. 2023.

[14] Juan Martinez-Romo, Lourdes Araujo, Xabier Larrayoz, Maite Oronoz, and Alicia Pérez. "OBSER-MENH at eRisk 2023: Deep Learning-Based Approaches for Symptom Detection in Depression and Early Identification of Pathological Gambling Indicators." In: *CLEF (Working Notes)*. 2023.

[15] Ana-Maria Bucur. "Utilizing ChatGPT Generated Data to Retrieve Depression Symptoms from Social Media." In: *CLEF (Working Notes)*. 2023.

[16] Alejandro Pardo Bascuñana and Isabel Segura-Bedmar. "APB-UC3M at eRisk 2024: Natural Language Processing and Deep Learning for the Early Detection of Mental Disorders." In: *Conference and Labs of the Evaluation Forum*. 2024. URL: https://api.semanticscholar.org/CorpusID:271779588.

[17] Anna Barachanou, Filareti Tsalakanidou, and Symeon Papadopoulos. "REBECCA at eRisk 2024: Search for Symptoms of Depression Using Sentence Embeddings and Prompt-Based Filtering." In: *Conference and Labs of the Evaluation Forum*. 2024. URL: https://api.semanticscholar.org/CorpusID:271772750.

[18] E. Gabrilovich and S. Markovitch. "Wikipedia-based Semantic Interpretation for Natural Language Processing." In: *Journal of Artificial Intelligence Research* 34 (2009), pp. 443–498. DOI: 10.1613/jair.2669. URL: https://doi.org/10.1613/jair.2669.

[19] Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun. "Pre-Trained Language Models and Their Applications." In: *Engineering* 25 (2023), pp. 51–65. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2022.04.024. URL: https://www.sciencedirect.com/science/article/pii/S2095809922006324.

[20] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[21] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[22] Lamir Shkurti, Faton Kabashi, Vehebi Sofiu, and Arsim Susuri. "Performance Comparison of Machine Learning Algorithms for Albanian News articles." In: *IFAC-PapersOnLine* 55.39 (2022). 21st IFAC Conference on Technology, Culture and International Stability TECIS 2022, pp. 292–295. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2022.12.037. URL: https://www.sciencedirect.com/science/article/pii/S2405896322030774.

[23] Jimin Tan, Jianan Yang, Sai Wu, Gang Chen, and Jake Zhao. *A critical look at the current train/test split in machine learning*. 2021. arXiv: 2106.04525 [cs.LG]. URL: https://arxiv.org/abs/2106.04525.

[24] Akiko Aizawa. "An information-theoretic perspective of tf–idf measures." In: *Information Processing Management* 39.1 (2003), pp. 45–65. ISSN: 0306-4573. DOI: https://doi.org/10.1016/S0306-4573(02)00021-3. URL: https://www.sciencedirect.com/science/article/pii/S0306457302000213.

[25] Xinchuan Zeng and Tony R. Martinez. "Distribution-balanced stratified cross-validation for accuracy estimation." In: *Journal of Experimental & Theoretical Artificial Intelligence* 12.1 (2000), pp. 1–12. DOI: 10.1080/095281300146272.

[26] Petro Liashchynskyi and Pavlo Liashchynskyi. "Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS." In: (Dec. 2019).

[27] Terry Windeatt. "Ensemble MLP Classifier Design." In: *Computational Intelligence Paradigms: Innovative Applications*. Ed. by Lakhmi C. Jain, Mika Sato-Ilic, Maria Virvou, George A. Tsihrintzis, Valentina Emilia Balas, and Canicious Abeynayake. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 133–147. ISBN: 978-3-540-79474-5. DOI: 10.1007/978-3-540-79474-5_6. URL: https://doi.org/10.1007/978-3-540-79474-5_6.

[28] Ekaba Bisong. "Logistic Regression." In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA: Apress, 2019, pp. 243–250. DOI: 10.1007/978-1-4842-4470-8_20. URL: https://doi.org/10.1007/978-1-4842-4470-8_20.

[29] Ramin Ghorbani and Rouzbeh Ghousi. "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques." In: *IEEE Access* 8 (2020), pp. 67899–67911. DOI: 10.1109/ACCESS.2020.2986809.

[30] Google. *langdetect*. 2021. URL: https://pypi.org/project/langdetect/.

[31] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[32] Casper da Costa-Luis. "tqdm: A fast, Extensible Progress Bar for Python and CLIe." In: *Zenodo* (2024). DOI: 10.5281/zenodo.11107065.

[33] Ethan Zhang and Yi Zhang. "Average Precision." In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 192–193. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_482. URL: https://doi.org/10.1007/978-0-387-39940-9_482.

[34] Sujatha Pothula and P Dhavachelvan. "Precision at K in Multilingual Information Retrieval." In: *International Journal of Computer Applications* 24 (June 2011). DOI: 10.5120/2990-3929.

[35] Javed Aslam and Emine Yilmaz. "A geometric interpretation and analysis of R-precision." In: Oct. 2005, pp. 664–671. DOI: 10.1145/1099554.1099721.

[36] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. "Learning to Rank by Optimizing NDCG Measure." In: Jan. 2009, pp. 1883–1891.