Project report
"Detecting Dementia Using Natural Language Processing"
DAT305

University of Stavanger

Daniil Krichevskiy

2024

1 Introduction

This project aims to develop several predictive models for detecting dementia using Natural Language Processing (NLP). Both machine learning (ML) and deep learning techniques are used.

NLP is a branch of artificial intelligence that focuses on the interaction between computers and human language. By enabling machines to understand, interpret, and generate human language, NLP has proven instrumental in fields like healthcare, where analyzing textual or spoken data can reveal patterns indicative of specific conditions.

In the fight against dementia, NLP offers unique advantages. Speech and language are often affected in the early stages of cognitive decline, making linguistic patterns a valuable diagnostic tool. By processing and analyzing patient transcripts, NLP models can identify subtle changes in language use, providing a non-invasive, cost-effective, and scalable method for early detection and monitoring of dementia.

All the files related to this project are stored in the GitHub repository.

The whole project is done by Daniil Krichevskiy.

2 Dataset overview

We were provided with three datasets from the DementiaBank:

- Control_db.csv: Contains speech and demographic data for participants without dementia (control group).
- Dementia_db.csv: Contains speech and demographic data for participants diagnosed with dementia.
- Testing_db.csv: Contains speech and demographic data for participants, used exclusively for testing the models.

The first two datasets, Control_db.csv and Dementia_db.csv, are combined to create the training dataset, which consists of 108 samples. The third dataset, Testing_db.csv, is used for evaluating the models and contains 48 samples.

All datasets share the same structure with the following columns:

- Language: The language of the participants transcript (e.g., eng).
- Data: The source of data.
- Participant: A unique identifier for the participant.
- Age: The age of the participant in years.
- Gender: The gender of the participant (male or female).
- Diagnosis: The diagnostic category (e.g., Control or ProbableAD for probable Alzheimers disease).
- Category: Binary encoding of the diagnosis (0 for Control, 1 for Dementia).

- MMSE: The Mini-Mental State Examination score, used to measure cognitive impairment.
- Filename: The identifier for the corresponding transcript file.
- Transcript: The transcript of the participants speech.

The train dataset, created by merging the Control and Dementia datasets, provides the data necessary for training models. The testing dataset remains separate to evaluate the performance of the developed models.

3 Data preprocessing

In the data preprocessing stage, several steps were undertaken to ensure the datasets were clean, consistent, and ready for model training and evaluation. First, missing values in the training data were addressed. Specifically, a missing value in the MMSE column of the training dataset was replaced with the mean MMSE value calculated for participants in the same category and gender group.

Next, categorical data was transformed for compatibility with machine learning algorithms. The Gender column, for instance, was converted into binary values, with female represented as 1 and male as 0.

The text data in the Transcript column underwent extensive cleaning. Punctuation marks and special characters were removed, underscores were replaced with spaces, and all text was converted to lowercase. Numbers and single isolated characters were also stripped from the text. To standardize the linguistic content further, lemmatization was applied using SpaCy's pretrained English model, ensuring that words were reduced to their base forms [1]. Finally, stop words (common words with little contextual importance) were removed to focus on meaningful linguistic patterns using the NLTK library [2].

By the end of this preprocessing phase, the text data was cleaned and transformed into a format suitable for feature extraction and model training.

4 Exploratory Analysis

In the exploratory analysis phase, we examined the dataset to ensure its balance and suitability for building robust predictive models. The gender distribution was found to be fairly balanced, with 55.6% females and 44.4% males. Similarly, the diagnostic categories were evenly distributed, with 50% of the samples belonging to the control group (Category = 0) and the remaining 50% to the dementia group (Category = 1) (see Figure 1). Within each gender, participants were evenly split between the control and dementia groups. This balanced distribution across both gender and diagnostic categories eliminates potential biases and ensures fairness in model development.

Age distribution was also analyzed across genders. Males were observed to have a slightly higher median age and a wider age range compared to females, but overall, the age distribution was well-balanced and exhibited minimal outliers (see Figure 2).

To investigate the relationships between numerical features, a correlation matrix was calculated (see Figure 3). This analysis revealed a strong negative correlation between MMSE scores and the dementia category (-0.84), indicating that lower MMSE scores are strongly associated with dementia. Conversely, age and gender showed very weak or negligible correlations with the dementia category, suggesting they are not significant predictors in this dataset.

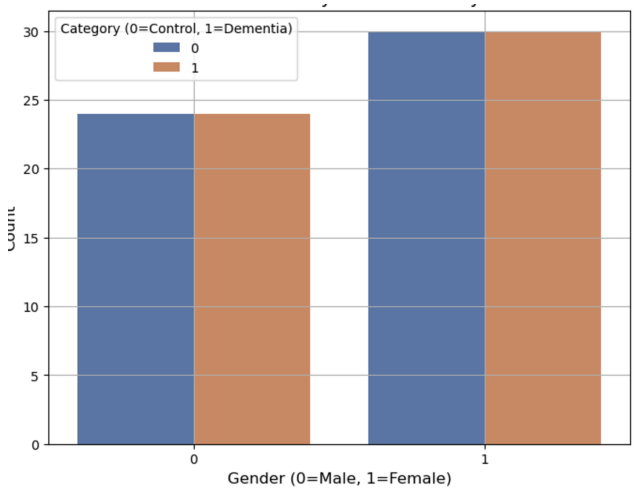

Figure 1: Distribution of patients with and without dementia in males and females.

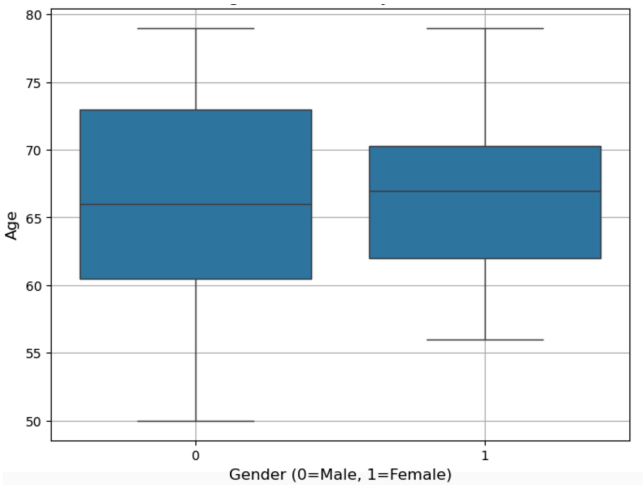

Figure 2: Age distribution by gender for males and females

5 Feature Extraction

In the feature extraction phase, we transformed the textual data in the Transcript column into numerical representations that could be used as input features for machine learning and deep learning models. Two different approaches were employed for this purpose: TF-IDF and BERT embeddings.

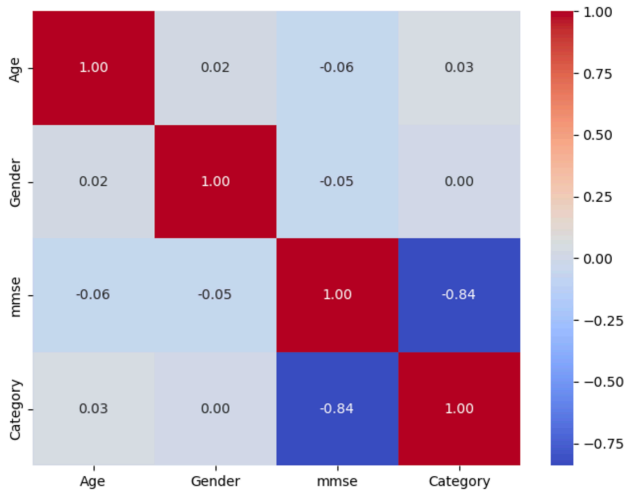

Figure 3: Correlation matrix for numeric features in the training dataset

The first method, TF-IDF (Term Frequency-Inverse Document Frequency), was used to convert the textual data into numerical with the help of TfidfVectorizer from the Scikit-learn library [3]. Firstly, the transcripts were tokenized and then TF-IDF score was calculated. TF-IDF reflects the importance of words within each document relative to the entire corpus. This approach captures the frequency of terms in a transcript while penalizing overly common terms that occur across many transcripts, ensuring that the resulting features are informative for distinguishing between dementia and control groups. The TF-IDF score of a term t in document d can be calculated as follows

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \text{IDF}(t), \tag{5.1}$$

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \tag{5.2}$$

$$\text{IDF}(t) = \log\left(\frac{N}{1 + |\{d \in D : t \in d\}|}\right), \tag{5.3}$$

where $\text{TF}(t, d)$ is the term t frequency in d divided by the total number of terms in the document and $\text{IDF}(t)$ is the inverse document frequency of t, calculated as the logarithm of the total number of documents N divided by the number of documents containing t (plus 1 to avoid division by zero) (D is the total number of documents, i.e. size of a corpus).

The second method utilized BERT embeddings, derived from the pre-trained BERT model (bert-base-uncased) [4]. In this approach, each transcript was tokenized and passed through the BERT model to extract contextualized word embeddings. Specifically, we used the CLS token embedding from the final hidden layer as a summary representation for each transcript. This embedding captures semantic and syntactic nuances of the language.

Both feature extraction methods produced numerical datasets: one based on TF-IDF vectors and another based on BERT embeddings. These feature sets were then used to train ML and DL models.

6 ML approach

In this part, we applied various ML models to predict dementia status using the features extracted in the previous step. The focus was on building and comparing models with both TF-IDF and BERT embeddings to evaluate their effectiveness. The ML techniques used included logistic regression and Support Vector Machines (SVM). They were implemented using Scikit-learn library [3].

6.1 Baseline model

We started with a baseline logistic regression model using a single numerical feature (see Figure 4), the MMSE score, as it was shown during exploratory analysis to be a strong predictor of dementia. This baseline model achieved high accuracy ($\approx 90\%$) and served as a benchmark to evaluate the performance of models trained on textual features.

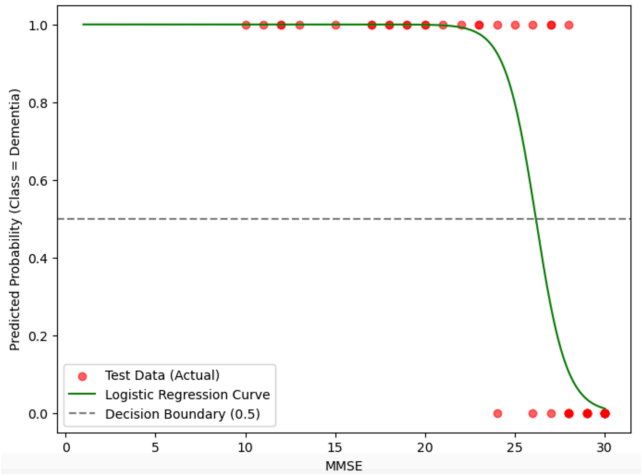

Figure 4: Logistic regression model based on the MMSE score

6.2 Logistic regression

Then we extended the logistic regression approach to work with the TF-IDF features extracted from the transcripts This model achieved an accuracy of 87.5%, with strong precision, recall, and F1 scores. While the accuracy was slightly lower than the MMSE-only model, the results demonstrated that textual features could also serve as a reliable basis for prediction.

Next, the logistic regression model was trained using BERT embeddings. However, the results with BERT features were significantly worse, with an accuracy of only 75%. This likely reflects the relatively small dataset size, as BERT embeddings often require fine-tuning on larger datasets to achieve optimal performance.

6.3 Support Vector Machines

At the next step the SVM approach was used. SVM is a powerful Ml algorithm that performs well with high-dimensional data. Using the TF-IDF features, the SVM model achieved the best performance, matching the accuracy of the MMSE baseline at 90% (see Figure 5). The SVM classifier demonstrated robustness in distinguishing between the control and dementia groups. When applied to BERT embeddings, the SVM model performed similarly to logistic regression with BERT, achieving an accuracy of 75%.

The results of our ML classifiers are summarized in Figure 6.

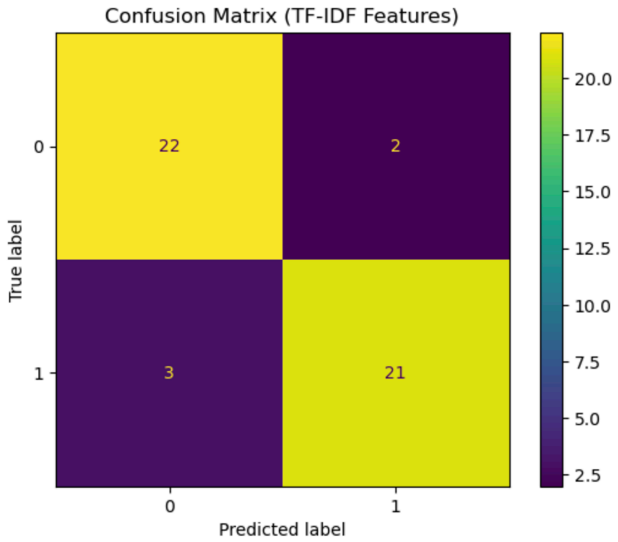

Figure 5: Confusion matrix for the SVM based on the TF-IDF embedding

Performance on the testing dataset	Logistic regression with MMSE	Logistic regression with transcripts		SVM with transcripts	
		TF-IDF embeddings	BERT embeddings	TF-IDF embeddings	BERT embeddings
Precision	0.91	0.88	0.77	0.91	0.75
Recall	0.88	0.88	0.71	0.88	0.75
F1 - score	0.89	0.88	0.74	0.89	0.75
Accuracy	0.90	0.88	0.75	0.90	0.75

Figure 6: Results for ML techniques

7 Deep learning approach

In this section, we explored two deep learning approaches to predict dementia: a feedforward neural network and a pre-trained transformer-based model (BERT) implemented using the Hugging Face Transformers library.

7.1 Feedforward neural network

For this part the library PyTorch was used [5]. The feedforward neural network worked with the TF-IDF features extracted from the transcripts. The optimal performance was reached by a neural network which consisted of two fully connected hidden layers, each with 64 neurons and ReLU activation functions, followed by an output layer with a sigmoid activation for binary classification. We trained the network using Binary Cross-Entropy Loss and the Adam optimizer, leveraging 4-fold cross-validation to evaluate its generalization. On the test set, the model achieved an accuracy of 81.25%, with better performance on control samples compared to dementia samples. However, the training and validation loss trends indicated overfitting, likely due to the small dataset size (see Figure 7).

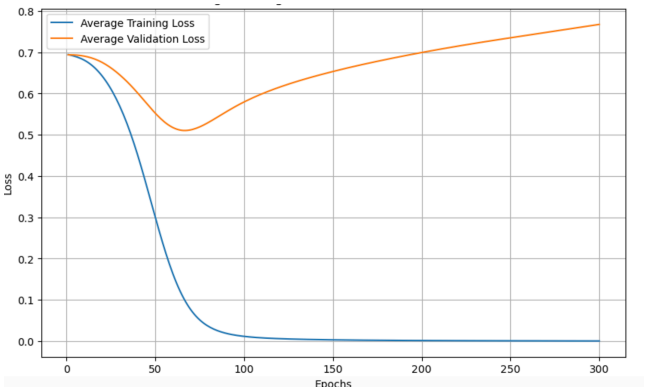

Figure 7: Training and validation loss for the feedforward neural network

7.2 Transformer-based model

For the transformer-based model, we utilized BERT (Bidirectional Encoder Representations from Transformers) via the Hugging Face Transformers library [4]. The transcripts were tokenized using BertTokenizer with padding and truncation to prepare the input for the model. We employed a pre-trained BERT base model (bert-base-uncased) with a classification heada fully connected layeradded on top to perform binary classification. The model was fine-tuned using Binary Cross-Entropy Loss and optimized with the Hugging Face implementation of AdamW.

The BERT-based model achieved an accuracy of 85% on the test dataset, outperforming the feedforward network but still falling short of the SVM model with TF-IDF features (see Figure 8 for confusion matrix). The Hugging Face library significantly streamlined the implementation of the transformer model, from tokenization to fine-tuning, making it straightforward to apply a pre-trained model for this binary classification task. Despite its stronger performance, the BERT model exhibited signs of overfitting, as the validation loss began to increase after several epochs of training (see Figure 9). This highlights the limitations posed by the small dataset size, which restricted the ability to fully exploit the power of BERT.

The results of our DL classifiers are summarized in Figure 10.

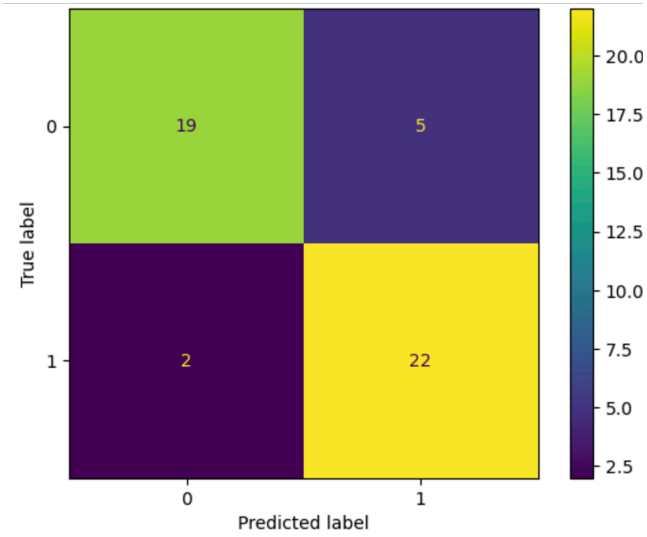

Figure 8: Confusion matrix for the BERT-based model

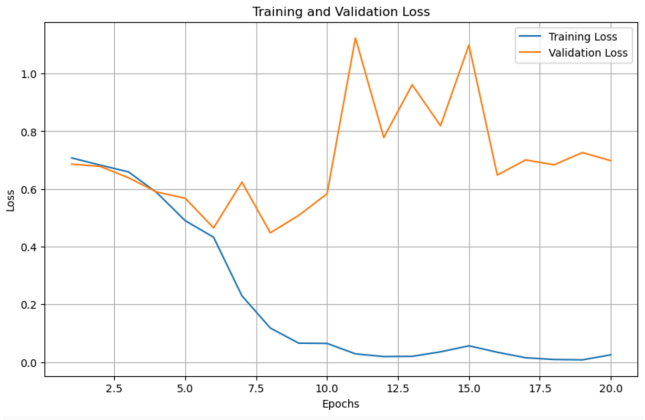

Figure 9: Training and validation loss for the BERT-based model

Performance on the testing dataset	Feedforward neural network	BERT-based model
Precision	0.89	0.81
Recall	0.71	0.92
F1 - score	0.79	0.86
Accuracy	0.81	0.85

Figure 10: Results for DL techniques

8 Conclusion

In this project, we developed and evaluated multiple models for detecting dementia using both ML and DL techniques. Among all the models, the SVM with TF-IDF features achieved the best performance, matching the baseline MMSE-based logistic regression with an accuracy of 90%.

On the other hand, DL, including the BERT-based transformer and the feedforward neural network, showed moderate performance, with the BERT model achieving an accuracy of 85%. However, these models were limited by the small dataset size, which caused overfitting and restricted their ability to reach their full potential.

To improve the performance of DL models in future work, it is essential to expand the dataset. This could involve collecting more real-world data or generating synthetic data (e.g.text augmentation) to increase the training size and variability.

References

[1] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing." 2017.

[2] E. Loper and S. Bird, *Nltk: The natural language toolkit*, 2002.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., *Scikit-learn: Machine learning in python*, 2018.

[4] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi et al., *Huggingface's transformers: State-of-the-art natural language processing*, 2020.

[5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., *Pytorch: An imperative style, high-performance deep learning library*, 2019.