A comparison between NLP and traditional statistical analyses for dementia prediction

Chanpreet Singh

Academic Supervisor: Dr. Frank Rudzicz

A report presented for the degree of

Master of Applied Computer Science

Department of Computer Science
Dalhousie University
Canada
April 2023

Abstract

Dementia is a cognitive disorder characterized by a decline in memory, thinking, behavior, and the ability to perform daily activities. It can significantly affect a person's quality of life and is often progressive, meaning that the symptoms get worse with time. While dementia does not yet have a cure [1], early detection and treatment can help to control symptoms and enhance the quality of life for people who are affected. With the help of modern techniques in the fields of statistical machine learning and natural language processing, dementia can be assessed by using data from electronic health records. Thus, it might act as a detective as well as preventive measure from getting dementia. In this project, we used data from the survey conducted as a part of the Island [2] project and applied both approaches to the dataset. During our experiments, we employed both techniques to analyze the data and further work is required to explore these findings in greater detail and to obtain more conclusive results.

Acknowledgements

This work was carried out as an experimental research-based project and completed as a part of the curriculum of the program Master of Applied Computer Science at Dalhousie University. The project was funded by my supervisor Dr. Frank Rudzicz via Dalhousie University. The project was conducted in collaboration with the Vector Institute, Toronto and University of Tasmania, Australia.

I want to thank my academic supervisor for his great mentorship, guidance, and support throughout the project. This work would not have been possible without his expertise and insightful ideas. I would also like to thank Eddy Roccati, from the University of Tasmania, for helping me gain insights about the raw data.

1 Background Information

This project involved a survey of the elderly population on King Island and Flinders Island, with a focus on identifying risk factors for dementia and other age-related conditions.

The dataset consists of two types of data:

- Background & Health Data – this data comprises covariates including age, gender, education, background health, and cognition (CANTAB).
- Interview Q/A - this data was collected via the Talk2Me platform[7] wherein participants are asked to complete eight different types of tasks. In every session, users carry out one or multiple iterations of each task, where every iteration corresponds to a distinct stimulus such as a distinct word to define or a picture to describe.

Table 1: Details of the tasks completed by the participants

Task	Mode	# of stimuli
Image naming	Text	6
Picture description	Audio	1
Fluency	Text	1
Story recall	Audio	1
Vocabulary	Text	6
Winograd schemas	Multiple-choice	5
Word-color Stroop	Audio	18
General disposition	Multiple-choice, Likert scale	5

Questions were posed to the participants, and their responses were recorded. The answers ranged from a single word to multiple paragraphs and were stored as digital audio as well as textual data. This audio data was further converted to transcripts using ASR models like Wav2Vec[3].

1823 participants completed the updated Background survey, with 1447 completing the updated survey multiple times between October 2019 and October 2020. Any data after October 2020 was not considered. Each participant was allotted a CANTAB ID, and 1432 participants gave the interview via Talk2Me.

2 Expected Outcome

We intend to build a predictive model (or ensemble of models) to predict dementia or probable dementia in a way that may be deployable in practice.

3 Methodology

We will apply classification algorithms to both types of data. For both types of data, we take *memory_change* as our target variable, which is 1 if the candidate noticed a substantial change in their memory and mental function in recent years, and 0 otherwise. We apply regression algorithms on both types of data using the *PALFAMS28Percentile* variable, which is the percentile value of the PAL First Attempt Memory Score. This score is the frequency of correct identification of pattern locations related to the CANTAB memory task [11].

4 Classification and Regression on background and health data

4.1 Data preprocessing

Before proceeding to classification, we need to clean and preprocess the data. The main issues in the data were:

- Missing values – missing values were treated by dropping the feature. Using this technique, by keeping 45% (of the total count of null values) as the threshold value, we dropped 21 features. These features included *reside_with_other, usual_work_pattern,* and *occupation*.
- Irregular cardinality - we identified categorical features with cardinality greater than 20, (e.g., *num_grandchildren, reside_with_other)* and those with cardinality of 1e.g., *Rater, sid*). We removed those identified features.

After all necessary preprocessing, 50 features out of 183 features remained.

4.2 Feature selection

We calculated p-values and correlation values with respect to the *memory_change* variable. Finally, 50 features were used for the modelling, shown in Table 2

Table 2: Sorted 50 features with their correlation values and p-values

Feature	corr_value	p_value
memory_change_discussed	0.9383	0
pysch_diagnosis	0.1268	0.034
stroke_tia_attack	0.12	0
medications_new_since_last_surveys	0.0822	0.024
head_injury	0.0779	0.736
head_injury_severity	0.0691	0
marital_status	0.0649	0
hearing_impairment	0.0594	0.688
medications	0.0586	0.936
event_movie_cinema_normal	0.0586	0.619
visual_legally_blind	0.0549	0.55
hearing_impairment_correction	0.0519	0.267
kidney_disease_diagnosis	0.0475	0.013
dementia_family_history	0.0474	0.203
host_visitors_how_often	0.0449	0.401
heart_disease_diagnosis	0.0445	0.221
event_restaurant_normal	0.0415	0.083
dementia_diagnosis	0.0414	0.137
visual_corrective_glasses	0.0386	0.182
event_visiting_friends_normal	0.0381	0.801

gender	0.0368	0.129
heart_disease_type	0.0334	0.833
employed	0.0316	0.611
cns_diagnosis	0.0293	0.542
event_pub_rsl_normal	0.0291	0.906
articles_about_dementia_risk	0.0287	0.292
epilepsy_diagnosis	0.0267	0.035
language_english_only	0.0256	0.935
event_play_drama_normal	0.0241	0.387
memory_impairment_diagnosis	0.0217	0.01
event_sporting_event_normal	0.0211	0.359
other_dementia_risk_programs	0.0211	0.92
meeting_club_group_how_often_month	0.0209	0.903
cancer_diagnosis	0.0207	0.987
b12_deficiency_diagnosis	0.02	0.729
active_member_club_group_num	0.0198	0.237
event_special_performance_normal	0.0185	0.193
cns_diagnosis_disorder	0.0184	0.053
event_dancing_normal	0.0183	0.498
event_music_recital_normal	0.0153	0.497
retired	0.0119	0.446
visual_colour_blind	0.0105	0.673
active_member_club_group	0.0075	0.156
volunteer	0.0071	0.72
outing_family_friend_how_often	0.0065	0.166
delerium_diagnosis	0.0061	0.297
age_in_years	0.0055	0.01
liver_disease_diagnosis	0.002	0.978
event_none_normal	0.0017	0.856
cancer_type	0.0007	0.188

4.3 Classification models

We have selected Logistic Regression, Decision Tree Classifier, and Random Forest Classifier as our classification models.

4.3.1 Logistic Regression

Logistic regression is a statistical method used to analyze the relationship between a binary dependent variable and one or more independent variables by estimating the probability of the dependent variable taking a certain value.

Table 3: Parameters used in the model training using logistic regression [8]

Parameter	Value	Use
penalty	l2	Regularization
tol	1e-4	Tolerance for stopping criteria.
C	1	Inverse of regularization strength – to strengthen regularization
max_iter	100	Maximum number of iterations taken for the solvers to converge.

4.3.2 Decision Tree Classifier

A decision tree classifier is a machine learning approach that makes predictions about the categorization of new data using a tree-like model of decisions. Each leaf node of the tree is given a label based on the majority class of the training samples that reach that node after the training data has been recursively divided into subsets according to the values of the input characteristics.

Table 4: Parameters used in the model training using decision tree classifier [9]

Parameter	Value	Use
criterion	entropy	The function to measure the quality of a split.
random_state	0	Controls the randomness of the estimator.
splitter	best	The strategy used to choose the split at each node.
max_depth	None	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
min_samples_split	2	The minimum number of samples required to split an internal node
min_samples_leafint	1	The minimum number of samples required to be at a leaf node.

4.3.3 Random Forest Classifier

A random forest classifier is a collective machine learning technique that constructs several decision trees and combines the results. To create each tree, random subsets of the training data and features are chosen, and the results are aggregated to increase accuracy overall and decrease overfitting.

Table 5: Parameters used in the model training using random forest classifier

Parameter	Value	Use
n_estimators	100	The number of trees in the forest.
criterion	entropy	The function to measure the quality of a split.
max_depth	None	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
min_samples_split	2	The minimum number of samples required to split an internal node
random_state	0	Controls the randomness of the estimator.

Thus, for each of our classification models, we use top-5, top-10, top-15, top-20, top-25, top-30, top-35, top-40, top-45, and op-50 and find their performance.

Table 6: Performance of the classifiers

	Logistic Regression			**Decision Tree Classifier**			**Random Forest Classifier**		
# of features	**Precision**	**Recall**	**f1-score**	**Precision**	**Recall**	**f1-score**	**Precision**	**Recall**	**f1-score**
5	0.99451	0.99448	0.99443	0.99451	0.99448	0.99443	0.99451	0.99448	0.99443
10	0.99451	0.99448	0.99443	0.99168	0.99171	0.99168	0.99451	0.99448	0.99443
15	0.99451	0.99448	0.99443	0.99168	0.99171	0.99168	0.99451	0.99448	0.99443
20	0.99451	0.99448	0.99443	0.99168	0.99171	0.99168	0.99451	0.99448	0.99443
25	0.99179	0.99171	0.99162	0.99168	0.99171	0.99168	0.99451	0.99448	0.99443
30	0.99179	0.99171	0.99162	0.99168	0.99171	0.99168	0.99451	0.99448	0.99443
35	0.99179	0.99171	0.99162	0.99168	0.99171	0.99168	0.99451	0.99448	0.99443
40	0.99451	0.99448	0.99443	0.99451	0.99448	0.99443	0.99451	0.99448	0.99443
45	0.99451	0.99448	0.99443	0.99451	0.99448	0.99443	0.99451	0.99448	0.99443
50	0.99451	0.99448	0.99443	0.9918	0.99171	0.99174	0.99451	0.99448	0.99443

4.4 Regression

For regression, we use both a linear regressor and a random forest regressor and consider PALFAMS*28Percentile* as our dependent variable.

4.4.1 Linear Regressor

A relationship between a dependent variable and one or more independent variables can be found statistically using linear regression. The data are fitted to a linear equation, and the least squares approach is used to estimate the coefficients of the equation.

4.4.2 Random Forest Regressor

The random forest regressor is a machine learning technique that forecasts continuous numerical values using a group of decision trees. Due to its ability to handle complicated and high-dimensional data, it is a widely used technique for solving regression problems.

Table 8: Parameters used in the model training using random forest regressor

Parameter	**Value**	**Use**
n_estimators	100	The number of trees in the forest.
criterion	entropy	The function to measure the quality of a split.
max_depth	None	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
min_samples_split	2	The minimum number of samples required to split an internal node
random_state	0	Controls the randomness of the estimator.

Thus, for each of our regression models, we use top-5, top-10, top-15, top-20, top-25, top-30, top-35, top-40 and all columns and find their performance (mean squared error between the actual and predicted values).

Table 9: Performance(mean squared error) of regression models

# of features	**Linear Regression**	**Random Forest Regression**
5	706.2462	712.6081
10	704.7776	814.5931
15	709.5347	934.6112
20	710.7635	913.279
25	697.0351	846.5854
30	701.7834	794.2321
35	697.0804	761.6439
40	709.0861	733.2701
45	699.4043	735.0267
50	694.8297	720.4585

5 Classification and Regression on Talk2Me transcripts using NLP

5.1 Classification

We mapped the transcripts with the *memory_change* variable from the health and background data based on user ids. We used BERT [5] for the classification, specifically bert-base-uncased [6]. We performed several experiments with the data, each using the stratified K-fold technique (k=10)

There can be more than 1 transcript for each user, because of different tasks. Thus, while training, a user's textual data exclusively falls in either the training, validation, or test data.

5.1.1 Model training without preprocessing of data

In this experiment, we did not perform any kind of preprocessing on the text data.

Table 10: Training Arguments while training without any preprocessing

Metric name	value	use
evaluation_strategy	epoch	The evaluation strategy to adopt during training. 'epoch' means evaluation is done at the end of each epoch.
save_strategy	epoch	The checkpoint save strategy to adopt during training. 'epoch' means save is done at the end of each epoch.
learning_rate	2e-5	The initial learning rate.
per_device_train_batch_size	8	The batch size per GPU/TPU core/CPU for training.
per_device_eval_batch_size	8	The batch size per GPU/TPU core/CPU for evaluation.
num_train_epochs	15	Total number of training epochs to perform
weight_decay	0.01	The weight decay to apply (if not zero) to all layers except all bias and LayerNorm weights in the optimizer. It is one of the regularization techniques to prevent overfitting.
load_best_model_at_end	TRUE	Whether or not to load the best model found during training at the end of training.
metric_for_best_model	f1	Model selection strategy, to compare models in the end and loading the best one

Table 11: Performance of the model (raw data)

K(fold)	Precision	Recall	f1-score	Accuracy
1	0.757	0.87	0.81	0.87
2	0.728	0.853	0.786	0.853
3	0.758	0.871	0.811	0.871
4	0.725	0.851	0.783	0.851
5	0.718	0.847	0.777	0.847
6	0.67	0.819	0.737	0.819
7	0.681	0.825	0.746	0.825
8	0.747	0.864	0.801	0.864
9	0.693	0.833	0.757	0.833
10	0.748	0.865	0.802	0.865

The performance of the model (as per Table 11) might look satisfactory, but this is because of majority class classification. Also, it was observed that the training loss in every fold after all epochs tended to minimize to 0.004 approximately whereas the validation loss tended to reach 0.85 approximately.

5.1.2 Model training after undersampling

Since the data in the previous experiment was highly imbalanced (Class 1 – 86%, Class 0 – 14%), we tried undersampling and retried with the same parameters as in Table 10.

Table 12: Performance of the model (after undersampling)

K(fold)	Precision	Recall	f1-score	Accuracy
1	0.413	0.448	0.378	0.448
2	0.536	0.5	0.439	0.5
3	0.541	0.543	0.542	0.543
4	0.44	0.441	0.44	0.441
5	0.465	0.492	0.452	0.492
6	0.455	0.446	0.435	0.446
7	0.515	0.522	0.493	0.522
8	0.552	0.545	0.54	0.545
9	0.547	0.544	0.543	0.544
10	0.481	0.481	0.478	0.481

As per table 12, the model's performance was still unsatisfactory. The average f1-score being 0.474 is non-considerable. Thus, applied regularization by increasing dropouts. The batch size was increased to 32 and number of epochs decreased to 2.

5.1.4 Model training after undersampling and regularization

Regularization was applied by increasing dropouts. The model was trained with the dropout values of range [0, 0.6] with an equal interval of 0.025. For every dropout value, the performance metrics were recorded.

Table 10: Training Arguments while training without any preprocessing

Metric name	value
evaluation_strategy	epoch
save_strategy	epoch
learning_rate	2e-5
per_device_train_batch_size	32
per_device_eval_batch_size	32
num_train_epochs	5
weight_decay	0.01
load_best_model_at_end	TRUE
metric_for_best_model	f1

Table 14: Performance of the model at different dropout values (after undersampling and regularization)

dropout	avg_10-fold_Precision	avg_10-fold_Recall	avg_10-fold_f1-score	avg_10-fold_Accuracy
0	0.477	0.475	0.404	0.475
0.025	0.466	0.507	0.429	0.507
0.05	0.502	0.495	0.43	0.495
0.075	0.434	0.49	0.391	0.49
0.1	0.451	0.505	0.426	0.505
0.125	0.407	0.481	0.408	0.481
0.15	0.516	0.509	0.469	0.509
0.175	0.447	0.501	0.438	0.501
0.2	0.421	0.475	0.412	0.475
0.225	0.422	0.496	0.399	0.496
0.25	0.467	0.5	0.42	0.5
0.275	0.427	0.485	0.416	0.485
0.3	0.434	0.487	0.42	0.487
0.325	0.481	0.485	0.42	0.485
0.35	0.394	0.464	0.398	0.464
0.375	0.403	0.481	0.417	0.481
0.4	0.403	0.479	0.41	0.479
0.425	0.402	0.493	0.403	0.493
0.45	0.408	0.483	0.394	0.483
0.475	0.457	0.497	0.434	0.497

0.5	0.46	0.492	0.425	0.492
0.525	0.58	0.519	0.446	0.519
0.55	0.555	0.503	0.448	0.503
0.575	0.46	0.493	0.451	0.493
0.6	0.498	0.503	0.453	0.503

5.2 Regression

We mapped the transcripts with the *PALFAMS28Percentile* variable from the health and background data based on user ids. We used BERT[5] for the regression, specifically bert-base-uncased[6].

5.2.1 Regression without regularization

In the initial trial, we did not apply any dropouts.

Table 15: Training Arguments while training without any preprocessing

Metric name	value	use
evaluation_strategy	epoch	The evaluation strategy to adopt during training. 'epoch' means evaluation is done at the end of each epoch.
save_strategy	epoch	The checkpoint save strategy to adopt during training. 'epoch' means save is done at the end of each epoch.
learning_rate	2e-5	The initial learning rate.
per_device_train_batch_size	32	The batch size per GPU/TPU core/CPU for training.
per_device_eval_batch_size	32	The batch size per GPU/TPU core/CPU for evaluation.
num_train_epochs	20	Total number of training epochs to perform
weight_decay	0.01	The weight decay to apply (if not zero) to all layers except all bias and LayerNorm weights in the optimizer. It is one of the regularization techniques to prevent overfitting.
load_best_model_at_end	TRUE	Whether or not to load the best model found during training at the end of training.

metric_for_best_model	accuracy	Model selection strategy, to compare models in the end and loading the best one

Table 16: Performance of the model(without dropouts)

Mean squared error	2140.131
Mean absolute error	38.510
r2-score	-1.787

5.2.2 Regression with regularization

Regularization was applied by increasing dropouts. The model was trained with the dropout values of range [0, 0.4] with an equal interval of 0.025. For every dropout value, the performance metrics were recorded.

Training arguments were kept the same as Table 15 (the previous experiment)

Table 17: Performance of the model (without dropouts)

Dropout value	**Mean squared error**	**Mean absolute error**	**r2-score**
0	2037.064	38.424	-1.872
0.025	2062.505	37.649	-1.686
0.05	2138.906	38.496	-1.786
0.075	2139.959	38.508	-1.787
0.1	2140.131	38.51	-1.788
0.125	2139.076	38.498	-1.786
0.15	2139.075	38.498	-1.786
0.175	2139.298	38.501	-1.786
0.2	2139.493	38.503	-1.787
0.225	2139.521	38.503	-1.787
0.25	2139.67	38.505	-1.787
0.275	2139.609	38.504	-1.787
0.3	2044.184	38.752	-1.82
0.325	2141.244	38.522	-1.789
0.35	2141.171	38.521	-1.789
0.375	2140.176	38.51	-1.788
0.4	2140.955	38.519	-1.789

6 Limitations and further work – write as paragraph, not too long

There are some more BERT models like XLNET [12], RoBERTa [13], and many more which can be tried. Moreover, we can opt for more regularization techniques like applying class-specific weights to the loss function without performing undersampling. Also, if the data were available, it would be interesting to see if these results are consistent across languages, since there are a few BERT language models which work on languages other than English. It would also be appropriate to check for covariates in the data, and any relationships between aspects of the transcripts and structured data. This is left as future work.

References

[1] Dementia (2023) World Health Organization. World Health Organization. Available at: https://www.who.int/news-room/fact-sheets/detail/dementia (Accessed: April 1, 2023).

[2] Island project (2022) Wicking Dementia Research and Education Centre - University of Tasmania, Australia. Available at: https://www.utas.edu.au/wicking/research/distinct-projects/island-project (Accessed: April 1, 2023).

[3] Facebook/WAV2VEC2-large-960h · hugging face (no date) facebook/wav2vec2-large-960h · Hugging Face. Available at: https://huggingface.co/facebook/wav2vec2-large-960h (Accessed: April 9, 2023).

[4] Bach, M., Werner, A. and Palt, M. (2019) "The proposal of undersampling method for learning from imbalanced datasets," Procedia Computer Science, 159, pp. 125–134. Available at: https://doi.org/10.1016/j.procs.2019.09.167.

[5] Classify text with BERT: text : tensorflow (2023) TensorFlow. Available at: https://www.tensorflow.org/text/tutorials/classify_text_with_bert (Accessed: April 11, 2023).

[6] Devlin, J. et al. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2. Available at: https://doi.org/10.48550/arXiv.1810.04805.

[7] M. Komeili, C. Pou-Prom, D. Liaqat, K. C. Fraser, M. Yancheva, and F. Rudzicz, "Talk2Me: Automated linguistic data collection for personal assessment," PLOS ONE, vol. 14, no. 3. Public Library of Science (PLoS), p. e0212342, Mar. 27, 2019. doi: 10.1371/journal.pone.0212342.

[8] Sklearn.linear_model.logisticregression (2023) scikit. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (Accessed: April 13, 2023).

[9] Sklearn.tree.decisiontreeclassifier (2023) scikit. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (Accessed: April 13, 2023).

[10] 1.10. decision trees (2023) scikit. Available at: https://scikit-learn.org/stable/modules/tree.html#tree-mathematical-formulation (Accessed: April 13, 2023).

[11] Savannah K H Siew, Madeline F Y Han, Rathi Mahendran, Junhong Yu, Regression-Based Norms and Validation of the Cambridge Neuropsychological Test Automated Battery among Community-Living Older Adults in Singapore, Archives of Clinical Neuropsychology, Volume 37, Issue 2, March 2022, Pages 457–472, https://doi.org/10.1093/arclin/acab073

[12] XLNet - hugging face (2023). Available at: https://huggingface.co/docs/transformers/model_doc/xlnet (Accessed: April 19, 2023).

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized Bert pretraining approach,” arXiv.org, 26-Jul-2019. [Online]. Available: https://arxiv.org/abs/1907.11692. [Accessed: 19-Apr-2023].