HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 1 (2023)

Analyzing the Performance of TabTransformer in Brain Stroke Prediction

Hao Ming Xia[1] and Ramin Ramezani[2#]

[1]University Hill Secondary School, Canada
[2]University of California, Los Angeles
[#]Advisor

ABSTRACT

The adoption of electronic patient health records has paved the way for machine learning and deep learning in disease diagnostics and prediction. Though traditionally tree-based algorithms have performed well on structural data, neural networks are known to perform well on unstructured data and data with a large number of input features. Furthermore, transformer-based models such as TabTransformer have been shown to perform competitively with tree-based algorithms (Huang et al. 2020). In this paper, we compare TabTransformer's performance with other state-of-art machine learning algorithms such as XGBoost, RandomForest, DecisionTree, and feed-forward Multilayer Perceptron. We discovered that TabTransformer shows no significant improvement over MLP and performs worse in certain metrics. Neither TabTransformer nor MLP performed better than XGBoost, the best-performing algorithm for brain stroke prediction in Kaggle competitions.

Introduction

The widespread adoption of electronic health records in recent decades has dramatically improved the quality of patient care. EHRs improve communication between patient and provider by increasing transparency and accessibility and reduce medical errors by enhancing legibility and availability of patient data. Furthermore, they assist in developing novel treatments and algorithms as securely anonymized data sources for machine learning and other data mining techniques.

Tree-based algorithms like XGBoost, LightGBM and random forest perform well on structural/tabular data such as a dataset of electronic health records. However, TabTransformer, a deep tabular data modeling architecture introduced by Amazon in 2020 (Huang et al. 2020), claims to outperform (or perform competitively with) tree-based algorithms in problems with structural data. Therefore, we are interested in evaluating these algorithms' performance on our medical record dataset.

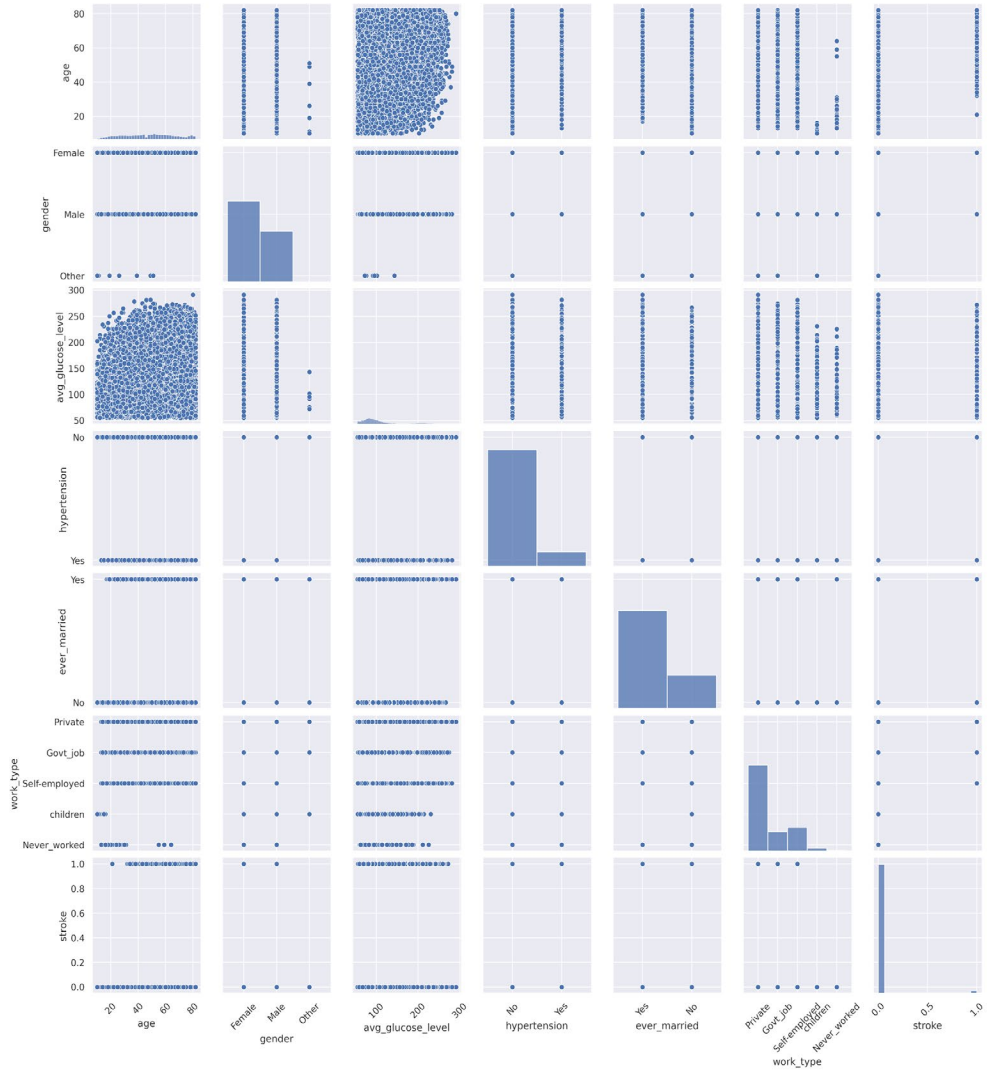
In this paper, we analyze the performance of TabTransformer in predicting the occurrence of brain stroke. We then benchmark other state-of-the-art methods: the Kaggle competition winner for the stroke prediction dataset[1], XGBoost (Chen and Guestrin 2016), and the model with the most performance in a 2019 (Nwosu et al. 2019) and 2022 (Dev et al. 2022) study, Multilayer Perceptron.

Methodology

Dataset

[1] https://www.kaggle.com/code/ahmtcnbs/stroke-prediction-xgboost-97

We selected a dataset of patient records first released for the McKinsey Analytics Hackathon[2] by McKinsey Analytics. The dataset contains information from 43,401 patients and 11 clinical features for predicting stroke events: patient gender, age, hypertension, marital status, work type, residence type, average blood glucose level, hypertension, body mass index, and current smoking status. The 12th attribute, the target variable, indicates if the patient has had a stroke or not. The dataset is published on Kaggle[3], a public data science competition platform and dataset repository.



Brain stroke prediction dataset pairwise relationships. The most important features for stroke prediction are age, marital status, hypertension, gender, work type, and average glucose level. (Dev et al. 2022).

Recognizing and removing redundant features that can be safely ignored without sacrificing prediction model performance would both assist clinicians in diagnosing stroke and reduce the computational cost of training. However, prior research on the impact of risk factors on stroke prediction using Principal Component Analysis (Nwosu et al. 2019), Learning Vector Quantization (Dev et al. 2022), and experimentation with different feature combinations (Dev et al.

[2] https://datahack.analyticsvidhya.com/contest/mckinsey-analytics-online-hackathon/

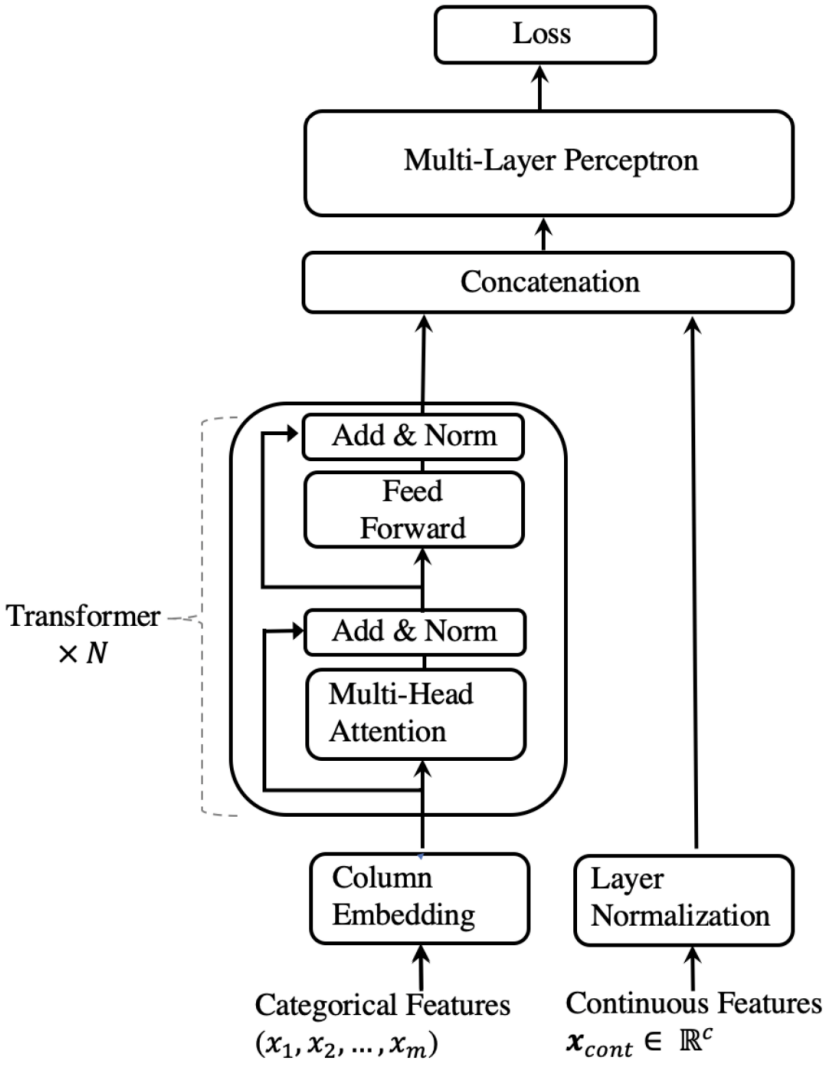
[3] https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

2022) show that the features are not highly correlated. Therefore, all patient attributes should be used to train our stroke prediction model.

Hyperparameters

We compare the performance of five classification algorithms — DecisionTree, RandomForest, XGBoost, Multilayer Perceptron, and TabTransformer.

A typical TabTransformer consists of a column embedding layer, a stack of Transformer layers, and a Multilayer Perceptron. However, in our TabTransformer model, we split categorical and numeric features before feeding both into a concatenation layer. The continuous features are input into a normalization layer. The categorical features are input into a StringLookup layer, a column embedding layer, and a stack of transformer layers. We initially tested TabTransformer with 32 embedding dimensions, 6 transformer blocks, and 8 attention heads as described in the paper (Huang et al. 2020) but discovered that the model performed better with 16 embedding dimensions, 3 transformer layers, and 4 transformer blocks.

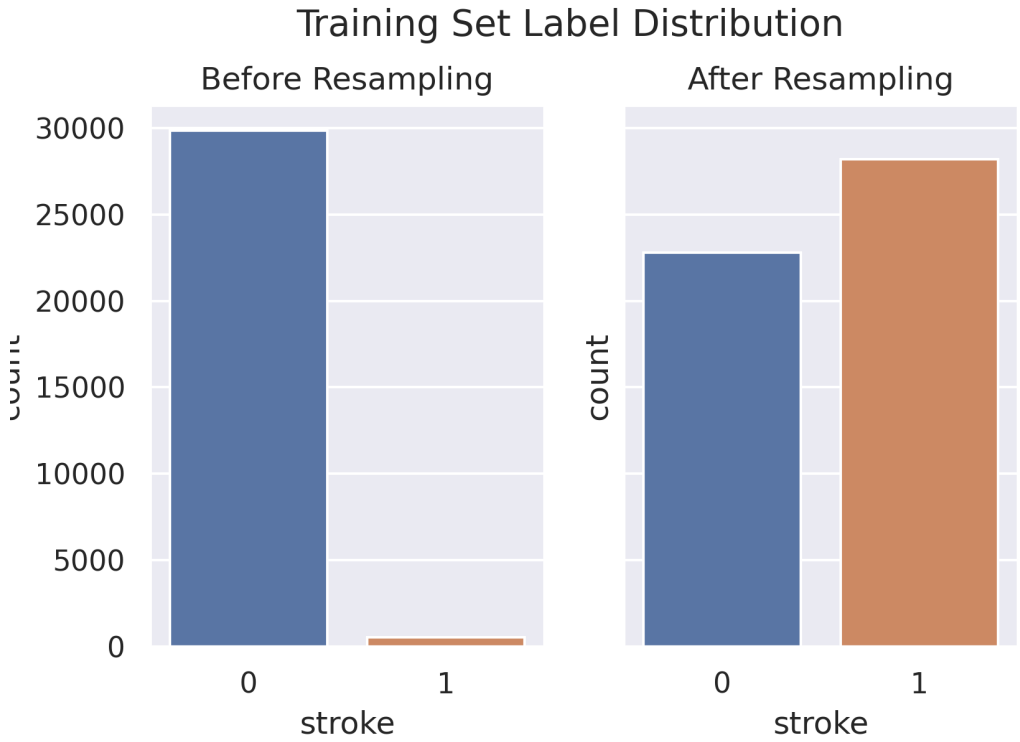


TabTransformer model architecture (Huang et al. 2020).

When we benchmark the baseline MLP model, we remove the transformer layers. We do not change RandomForest's or XGBoost's default parameters. We use the "gbtree" booster, a learning rate of 0.3, a tree maximum depth of 6, and regression with squared loss as the learning task and objective. For RandomForest, we use the default value of 100 trees in the forest, no maximum tree depth, and Gini impurity.

Preprocessing

Our first problem is that 30% of the records are from patients with unknown smoking status. Our dataset contains categorical and continuous values, so we impute these unknowns with MissForest (Stekhoven and Buhlmann 2011), a performant and computationally efficient missing value imputation algorithm that handles mixed-type data.
Moreover, the dataset of electronic health records is highly unbalanced at 98.2% negative and 1.8% positive. Out of 43,401 records in the dataset, only 748 are from patients with stroke. So, we divide our dataset into 70%/15%/15% cross-validation splits. We shuffle and split the records into sets of 3 training records, 6510 validation records, and 6510 test records and balance the training set. Treating the positive cases as the minority case and the negative cases as the majority case, we upsample the minority case with the SMOTE-NC algorithm and clean the resulting values with Edited Nearest Neighbors, resulting in a training set with 51013 rows. Finally, for XGBoost, RandomForest, and DecisionTree, we one-hot-encode the categorical features of each row.



Training Set Label Distribution

We repeat this training-validation-test split and resampling experiment 10 times to minimize sampling bias. Then, we take the mean results of all 10 experiments as the evaluation metrics of each classification algorithm.

Results

XGBoost has the best average performance over all experiments at 87.6% AUROC. Next are MLP, TabTransformer, RandomForest, and DecisionTree.

Mean precision, recall, F-score, accuracy, AUC, and miss rate with original training set over 10 experiments.

	Mean Precision	Mean Recall	Mean F-score	Mean Accuracy	Mean AUC - ROC	Mean AUC - PRC	Mean Miss Rate
XGBoost	0.5	0.013	0.025	0.983	0.906	0.236	0.017
RandomForest	0.99	0.27	0.414	0.988	0.935	0.688	0.012
DecisionTree	0.397	0.458	0.424	0.979	0.723	0.432	0.009
TabTransformer	0.338	0.009	0.018	0.982	0.773	0.08	0.018
MLP	0.778	0.147	0.238	0.984	0.901	0.357	0.015

Mean precision, recall, F-score, accuracy, AUC, and miss rate with balanced training set (after resampling) over 10 experiments.

	Mean Precision	Mean Recall	Mean F-score	Mean Accuracy	Mean AUC - ROC	Mean AUC - PRC	Mean Miss Rate
XGBoost	0.433	0.009	0.018	0.983	0.876	0.14	0.017
RandomForest	0.2	0.007	0.013	0.983	0.79	0.073	0.017
DecisionTree	0.166	0.292	0.211	0.964	0.634	0.235	0.012
TabTransformer	0.133	0.004	0.007	0.981	0.751	0.073	0.018
MLP	0.182	0.023	0.039	0.981	0.833	0.112	0.018

Discussion

TabTransformer shows no significant improvement over MLP. While both models have comparable accuracy and miss-rate at 98.1% and 1.8%, TabTransformer has worse precision, recall, f-score, and AUROC.

Neither TabTransformer nor MLP performed better than XGBoost, the best-performing algorithm for brain stroke prediction in Kaggle competitions. TabTransformer's performance is similar to RandomForest, while MLP comes closest in performance to XGBoost. Future work can go in two directions. First, we can perform an interpretability analysis to understand why TabTransformer does not perform as well as MLP. Second, we can experiment with other Transformer-based models such as GatedTabTransformer and TabNet.

References

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. https://doi.org/10.1145/2939672.2939785.

Dev, Soumyabrata, Hewei Wang, Chidozie Shamrock Nwosu, Nishtha Jain, Bharadwaj Veeravalli, and Deepu John. 2022. "A Predictive Analytics Approach for Stroke Prediction Using Machine Learning and Neural Networks." arXiv. https://doi.org/10.48550/ARXIV.2203.00497.

Huang, Xin, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. 2020. "TabTransformer: Tabular Data Modeling Using Contextual Embeddings." arXiv. https://doi.org/10.48550/ARXIV.2012.06678.

Nwosu, Chidozie Shamrock, Soumyabrata Dev, Peru Bhardwaj, Bharadwaj Veeravalli, and Deepu John. 2019. "Predicting Stroke from Electronic Health Records." arXiv. https://doi.org/10.48550/ARXIV.1904.11280.

Stekhoven, D. J., and P. Buhlmann. 2011. "MissForest–Non-Parametric Missing Value Imputation for Mixed-Type Data." *Bioinformatics* 28 (1): 112–18. https://doi.org/10.1093/bioinformatics/btr597.