Neuroimaging Multiscanner Normative Age and Assessments Prediction Using Stacked Ensemble Learning

Donovan Quimby
Team ID: 182
GTid: 903547532
Email: dquimby3@gatech.edu

Srinath Jagarlapudi
Team ID: 182
GTid: 902248445
Email: srinath@gatech.edu

Tawheed Altowaitee
Team ID: 182
GTid: 903543217
Email: taltowaitee3@gatech.edu

Human brain research using neuroimaging techniques such as MRI scans is an essential area of study as scientists uncover how age and other factors can affect the brain's function and structure. More recently, machine learning models have been used to identify how different brain regions contribute to healthy or disorder symptomatic effects. This paper describes a stacked ensemble learning model used to predict four independent assessments and age from a multimodal MRI Features dataset. A diverse ensemble of various machine learning models is developed and tuned independently using k-fold cross-validation in the level 1 *analysis. The predicted results (meta-features) from each model in the level* 1 *analysis are used to construct and compare four different linear regression models in the level* 2 *analysis. We demonstrate that the stacked ensemble models perform better than the predictions from the level* 1 *analysis except for the equally weighted linear model. Possible extensions of this work and ways to further improve performance are discussed.*

1 Problem Statement

This paper addresses the neuroimaging Kaggle challenge presented by the Center for Translational Research in Neuroimaging and Data Science (TReNDS). The challenge's goal is to build prediction models for a number of assessments of subjects such as age, based on multimodal brain features. The multimodal brain features are derived from MRI scans of unaffected subjects. The images from which the features were derived were generated by different types of scanners adding to the complexity of the problem.

2 Introduction

2.0.1 Background Information

The human brain is a very complex domain of study. The complexity stems from the low resolution of current non-invasive techniques common for studying the human brain. Low-resolution methods include fMRI, Electroencephalography (EEG), Positron Emission Tomography (PET), and many others. The non-invasive nature of these methods greatly impacts the information content and noise levels in the data collected. Machine learning based pattern recognition and matching methods present a promising avenue to expand the utility of currently available brain imaging techniques. A key challenge in utilizing machine learning techniques for brain research is the need for solutions that generalize to different imaging methods.

2.0.2 Data Set and Feature Selection

Features supplied for the competition are derived from brain MRI images using an unbiased strategy. To create the provided features, a separate, unrelated, and much larger imaging data set was utilized to learn feature templates. The templates were "projected" onto the original imaging data of each subject used in this Kaggle challenge using spatially constrained independent component analysis (scICA) via group information guided ICA (GIG-ICA). The competition organizers provide two separate sets of features created from different imaging sources.

The first set of features are source-based morphometry (SBM) loadings. These are subject-level weights from a group-level ICA decomposition of gray matter concentration maps from structural MRI (sMRI) scans. An example sMRI scan is provided in Figure 1. There were 27 features associated with this source.

The second set of features is derived from static func-

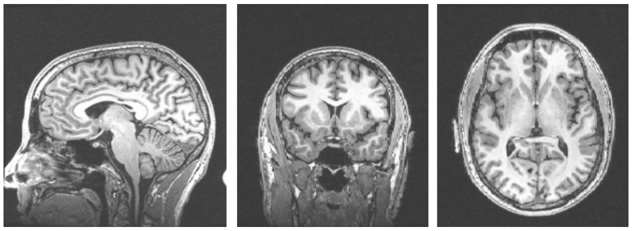

Fig. 1: Example sMRI scan image [1]

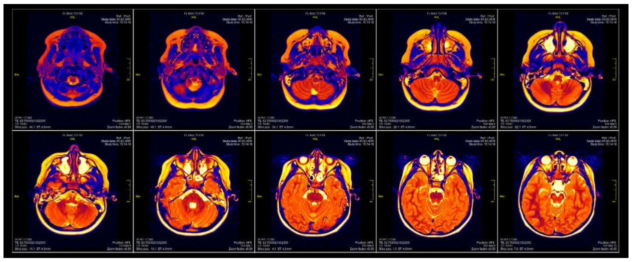

Fig. 2: Example fMRI scan image [2]

tional network connectivity (FNC) matrices. These are the subject-level cross-correlation values among 53 component-time courses estimated from GIG-ICA of resting-state functional MRI (fMRI). An example fMRI scan can be seen in Figure 2. 1379 FNC features are provided.

The data set considered in this study combines both provided feature sets for a total of 1406 features with 4703 observations. The provided features are all continuous and normalized.

Five response variables of interest were considered for this study and are listed below. For the analysis, each of the five responses is fitted with there own models.

Response Variables of Interest:

1. *age*
2. *domain1_var1*
3. *domain1_var2*
4. *domain2_var1*
5. *domain2_var2*

Approximately 15% of the 4703 observations contain at least one missing response variable. The missing values were imputed using an iterative imputation technique that fits a ridge regression model on the 50 most highly correlated features.

Dimensionality reduction was attempted using Principal Component Analysis (PCA). However, both low explained variance ratios and poor predictions results were observed. All the features were therefore used for model generation.

3 Modeling Approach

In this section, we introduce the machine learning methodology employed in this analysis. A brief discussion of each model used in the initial (Level 1) analysis and the various regression stacking methods that were studied for the level 2 analysis is discussed.

3.1 Stacked Ensemble Learning Overview

Ensemble Learning methods are a well-established modeling technique often used to improve the accuracy of a model. Stacked ensemble methods (also referred to as blended models in some literature) combine the predictions (meta-features) of a diverse set of models and use them as features to create a secondary learning model as first described by Wolpert in 1992 [3]. The secondary algorithm is trained to combine the meta-features in an optimal manner to make a final set of predictions. Combining a diverse variety of models helps reduce overfitting and generally results with a final accuracy higher than any of the individual models alone [4]. The stacking idea can be extended to multiple stages employing a mix of both linear and non-linear stacking models as desired. An example of a Kaggle winning ensemble stacking approach is shown in Figure 3 below.

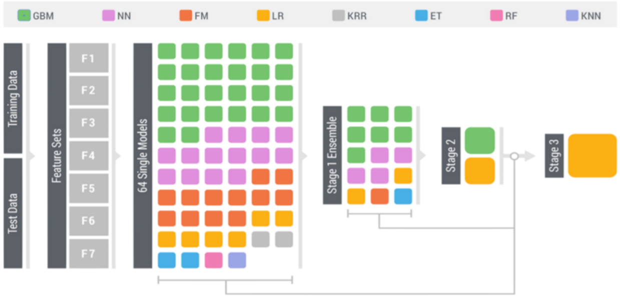

Fig. 3: Multi-level Ensemble Stacking Example [5]

The ensemble stacking modeling technique can be divided into various levels. The level 1 analysis is conducted on the original data features. The resulting predictions for each model in the level 1 analysis are used as meta-features for the level 2 analysis. Additional model levels may be added as needed to improve accuracy with each extra layer built upon the previous layer's predictions.

A possible problem that needs to be considered when created stacked ensemble models is data leakage. Data leakage can be defined as any time information from outside the original training data set is used to create the model. The additional information allows the model to learn on information it would otherwise not know, which can lead to an overestimation of the model's performance [6]. In ensemble stacking methods used in this study, the meta-features derived from the level 1 analysis may cause data leakage in the level 2 model.

Numerous methods have been suggested to minimize data leakage effects, each with their benefits and drawbacks [7]. We employ the strategy described in [8] and shown below to reduce the impact of data leakage.

Leakage Prevention Strategy:

1. Create a holdout of 25% of the train set
2. Fit the first stage models on the train set using 5-fold CV
3. Predict the untouched holdout with the trained first stage models
4. Split the predicted holdout into 5 folds
5. Fit a second stage model on $k-1$ folds and predict the k^{th} fold
6. Repeat 5) to predict each fold
7. The CV error of the second stage is calculated on each predicted fold and averaged

This analysis uses a 2-level approach to the stacking model. The initial level 1 analysis and subsequent predictions are conducted using the following four models.

Level 1 Models:

- Support Vector Machine (SVM)
- Multi-Adaptive Regression Spline (MARS)
- CatBoost Gradient Boosting (CAT)
- XGBoost Gradient Boosting (XGBoost)

Four types of Linear level 2 models are constructed and evaluated using the meta-features from level 1. Details of each of the Level 2 stacking models, shown below, are explained in the subsection 3.2. The performance of each model is discussed in their results section 4.

Level 2 Stacking Models:

- Simple Equally Weighted Linear Regression Model
- Optimally Weighted Linear Regression Model
- Optimally Weighted Linear regression Model With LASSO Regularization
- Optimally Weighted Linear Regression Model With Non-Negative Constraint

The overall structure of the model used in this study can be seen in Figure 4 below.

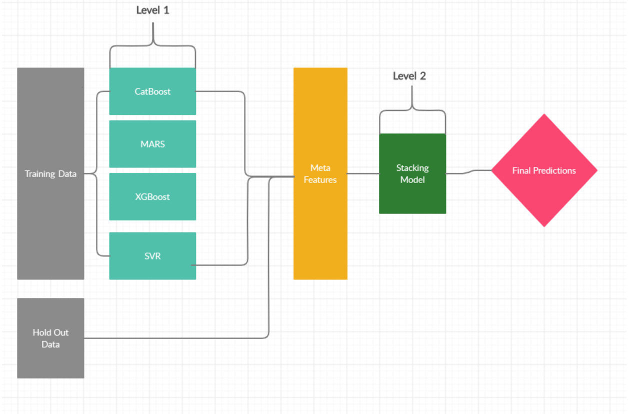

Fig. 4: Multi-level Ensemble Stacking Model used in Analysis

3.2 Level 2 Model Description

In this section, the various regression stacking models are discussed in more detail. The intent is to give a brief description of each model for the reader's general knowledge. More detailed descriptions can be found in the following references [3,4,9,10].

For this paper, let X represent the original feature matrix, y represent the original responses, $\hat{g}$ represent predicted responses using k-fold validation from the level 1 analysis, n represent the number of observations, and m represent the number of level 1 models.

3.2.1 Simple Equally Weighted Linear Regression Model

Let the simple equally weighted linear regression prediction model be defined as

$$b(g) = \sum_i w_i g_i(X), \forall(x) \in X \tag{1}$$

where w_i are the model weights defined by

$$w_i = \frac{1}{m} \tag{2}$$

such that all of the models contribute equally to the ensemble model. This methodology essentially takes the average of the predicted values from the level-1 analysis. This is a very simplistic and easily implemented method. However, this method's simplistic approach may lead to predictions that are worse than some of the individual modals in the level 1 analysis. For example, if one level 1 model is performing exceptionally poorly, assigning it an equal weight may result in poor performance of the level 2 stacked model. As discussed in the following sections, natural extensions of this method are likely to result in better predictive performance.

3.2.2 Optimally Weighted Linear Regression Model

The most obvious extension of the method discussed in 3.2.1 is to optimize the value of the weights w_i by solving the least square problem

$$\min_{w_i} \sum_{i=1}^{n} \left(y_i - \sum_i w_i g_i(X) \right)^2 \tag{3}$$

This method will assign unconstrained weights to the model resulting in a higher contribution of better performing models and smaller contribution of poorly performing models from the level 1 analysis.

3.2.3 Optimally Weighted Linear Regression Model With LASSO Regularization

Another possible methodology to use for the level 2 ensemble model is the use of Lasso Regularization. Lasso regularization seeks to minimize the least-squares problem while adding a penalty equal to the value of the coefficients' $L1$ norm. The problem can be formulated as

$$\min_{w_i} \sum_{i=1}^{n} \left(y_i - \sum_{i} w_i g_i(X) \right)^2 \tag{4}$$

subject to

$$\sum_{i=1}^{m} |\beta| \leq s \tag{5}$$

where s is a user defined tuning parameter.

3.2.4 Optimally Weighted Linear Regression Model With Non-Negative Constraint

The final method used in this study is to solve the linear regression problem described in 3.2.2 with the added constraint; the regression coefficients are non-negative. It has been shown that placing the non-negative on the regression model results in better predictions than any of the singular level 1 models in various studies [9, 10]. The linear regression model with the non-negative coefficient constraint takes the form

$$\min_{w_i} \sum_{i=1}^{n} \left(y_i - \sum_{i} w_i g_i(X) \right)^2 \tag{6}$$

subject to

$$w_i > 0, \ \ for\, i = 1, \ldots, m \tag{7}$$

4 Results

In this section, details of the level 1 and level 2 analysis are presented and discussed.

4.1 Level 1 Results

All level 1 models were tuned and trained on the training data set. Estimations of each model's individual out of sample performance on the holdout data set are determined using 5-fold cross-validation and presented for comparison to the final stacked models.

4.1.1 MARS

A Multi-Adaptive Regression Spline (MARS) model was trained on the training set. The MARS model performs fitting in a forward and pruning pass. The forward pass assesses each data point as a knot and creates a linear regression model with the features. This process is repeated until many knots are placed, often resulting in a highly non-linear prediction model. A subset of knots that produce a localized minimum cross-validation score is selected on the pruning pass. Additional information about the details of the MARS algorithm can be found in [11].

The mean 5-fold CV root mean squared error (RMSE) of the trained MARS models on the holdout data set for the five responses were as follows:

	age	domain1_var1	domain1_var2	domain2_var1	domain2_var2
MARS	9.7370	9.4696	11.2077	10.9704	11.8171

Table 1: 5-Fold CV RMSE of MARS model on holdout data set

4.1.2 CatBoost Gradient Boosting

CatBoost is an open-source Gradient Boosting decision tree algorithm. Details about the Algorithm can be found in [12]. For this study, hyperparameters were chosen using a grid-search technique in conjunction with a 5-fold CV on the training data set. For each response variable of interest, combinations of the following hyperparameter values were considered in the grid search.

- Tree number: 200,300,400,500,600,700
- Depth: 3,4,5,6
- Learning Rate: 0.03, 0.1
- l2 Leaf Regularization: 4,5,6,7
- Random Strength: 0.01, 1, 10
- Bagging Temperature: 0.1, 0.5, 1
- Border Count: 255, 1000

Based on the tuned hyper-parameters setup discussed above, The mean 5-fold CV RMSE of each model on the hold out set was as follows.

	age	domain1_var1	domain1_var2	domain2_var1	domain2_var2
CatBoost	3.7631	8.0191	8.6019	8.3693	9.6984

Table 2: 5-Fold CV RMSE of CatBoost model on holdout data set

4.1.3 XGBoost Gradient Boosting

XGBoost is another open source gradient boosting model. Specifically, a decision tree-based model combined with a k-fold cross-validation approach, was used. The gradient boosted model comprised 500 estimators with two variations of depth, 3 and 4. Each of the 500 estimators was trained on 70% of the samples and the features. Lastly, five models were trained for each of the target variables.

Based on the hyperparameters setup discussed above, the root mean squared error of the model on the holdout data set were as follows

	age	domain1_var1	domain1_var2	domain2_var1	domain2_var2
XGBoost	9.7770	9.3250	11.26	11.0620	11.8517

Table 3: 5-Fold CV RMSE of XGBoost model on holdout data set

4.1.4 Support Vector Regression

A support vector regression (SVR) model was used as the final level 1 model. An SVR model uses the same principals as an SVM for classification but is used for real value prediction. A grid-search based optimization using K-fold cross-validation to compare efficiencies was performed on the training data set to optimize parameters for the SVR model. The cost of miscalculation C was optimized to be 100 for the 'Age' parameter and 10 for the other four parameters. Radial Basis Function (RBF) was observed to give better predictions when compared to different functions.

Based on the hyperparameters optimized on the training data set, the RMSE values on the holdout data set are

	age	domain1_var1	domain1_var2	domain2_var1	domain2_var2
SVR	11.4399	9.5169	11.4306	11.2294	11.9346

Table 4: 5-Fold CV RMSE of SVR model on holdout data set

4.2 Level 2 Results

The four models constructed in the level 1 analysis were used to predict the hold-out data set's responses. The predictions are used as meta-features for training the level 2 stacking models. Four separate stacking models, as discussed in 3 were trained using the meta-features for each response value using 5-fold cross-validation and the mean CV RMSE of each level 2 model was determined. The model performance results for all five features of interest and all level 1 and level 2 models is shown in Table 5.

	age	domain1_var1	domain1_var2	domain2_var1	domain2_var2
MARS	9.7370	9.4696	11.2077	10.9704	11.8171
CatBoost	3.7631	8.0191	8.6019	8.3693	9.6984
XGBoost	9.7770	9.3250	11.26	11.0620	11.8517
SVR	11.4399	9.5169	11.4306	11.2294	11.9346
Equal Weight Stacking	7.5624	8.8792	10.3825	10.1426	11.0888
Optimized Weight Stacking	2.7939	7.0826	7.1111	6.3683	8.4263
LASSO Stacking	2.7939	7.0826	7.1026	6.3683	8.4252
Non-Negative Constraint Stacking	3.4310	7.4522	7.0995	6.7144	8.5092

Table 5: 5-Fold CV RMSE of various models with top performing models highlighted in green for each feature

The Equally Weighted Linear Regression model does not improve upon the best level 1 model's performance for any of the five features. This model's poor performance is to be expected, considering the performance discrepancies of the level 1 models. The CatBoost model outperforms the other three level 1 models by a significant margin. By applying equal weights to all of the level 1 models, the performance becomes worse. The possibility of this phenomenon is briefly discussed in 3.2.1.

Both the Optimally Weighted Linear Regression Model and the Optimally Weighted Linear Regression Model With LASSO Regularization show noticeably better performance over the best level 1 model on all responses. The LASSO model performs slightly better on $domain1\_var2$ and $domain2\_var2$ compared to the Optimally Weighted model, but the differences are minimal.

The Optimally Weighted Linear Regression Model With Non-Negative Constraint also performs better than all level 1 models on every response. However, performance gains are not as significant as the two models discussed above.

5 Conclusion

This study presented a stacked ensemble modeling approach to address the research problem of building prediction models for several assessments based on multimodal brain features. A two-level stacking approach is implemented with measures introduced to minimize data leakage. Two gradient boosting models, a MARS model, and an SVR model were used in the level 1 analysis to construct meta-features for the level 2 stacking model. Four linear regression model variations were investigated for the level 2 analysis. All but one stacking model demonstrated noticeable performance improvements over the best performing level 1 model. Additional extensions of this study could involve creating a more diverse set of level 1 models, performing more detailed hyperparameter tuning on the existing level 1 models to improve their baseline performance, exploring non-linear level 2 stacking models, and adding additional stacking levels.

References

[1] "Structural mri imaging." [Online]. Available: https://cfmriweb.ucsd.edu/Howto/3T/structure.html

[2] B. Crew, "A bug in fmri software could invalidate 15 years of brain research." [Online]. Available: https://www.sciencealert.com/

[3] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[4] J. Sill, G. Takács, L. Mackey, and D. Lin, "Feature-weighted linear stacking," *arXiv preprint arXiv:0911.0460*, 2009.

[5] "Winning Data Science Competitions: Jeong-Yoon Lee - YouTube." [Online]. Available: https://www.youtube.com/watch?v=ClAZQI_B4t8

[6] J. Brownlee, "Data leakage in machine learning," Jun 2020. [Online].

Available: https://machinelearningmastery.com/data-leakage-machine-learning/

[7] "Cross validation strategy when blending/stacking." [Online]. Available: https://www.kaggle.com/general/18793

[8] Jun 2015. [Online]. Available: https://mlwave.com/kaggle-ensembling-guide/

[9] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49–64, 1996.

[10] F. Güneş, R. Wolfinger, and P.-Y. Tan, "Stacked ensemble models for improved prediction accuracy," in *Proc. Static Anal. Symp.*, 2017, pp. 1–19.

[11] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, pp. 1–67, 1991.

[12] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Advances in neural information processing systems*, 2018, pp. 6638–6648.