IMPACT OF MACHINE LEARNING ALGORITHMS ON BRAIN AGE ESTIMATION ACCURACY

AUTHORS:

1.Mishika Agarwal ,.2 Boyapally Meghana Reddy, 3. Yata Yashmitha Reddy.

B -Tech, 4th year students Department of Data Science (CSE) Sphoorthy Engineering College.

mishikaagarwal2002@gmail.com. , meghanareddyb93@gmail.com , yashvireddy123@gmail.com .

Under the Supervision of Dr. Karanam Ramesh Rao. (Professor).

rameshrao@sphoorthyengg.ac.in.

ABSTRACT:

This study explores the accuracy of brain age estimation evaluates multiple regression algorithms such as linear regression, support vector regression, Knn, decision tree. The research, crucial for understanding neurological development and aging, examines how these algorithms perform in predicting brain age and their and clinical factors. It finds varying accuracy in brain age estimation among different regression algorithms, influenced by factors like data-set characteristics. The research explores the interpretability and generalizability of each algorithm. Insights aid in refining brain age estimation models, guiding researchers and clinicians to choose suitable algorithms for specific objectives and datasets. This contributes to better understanding brain development, aging, and neurodegenerative disorders. This study examines how different regression algorithms affect the accuracy of predicting brain age using neuroimaging data. We assess algorithms like linear regression, support vector regression, Knn ,decision tree and datasets. Findings reveal variations inaccuracy. Insights gained contribute to refining brain age estimation models for better understanding brain development . Our findings can serve as a starting point and quantitative reference for future efforts at improving brain age prediction using machine learning models applied to brain data.

Keywords:

Brain Age Estimation,Regression Algorithms, Accuracy Assessment,Age Prediction,Brain Development.

INTRODUCTION:

Recent times have witnessed an increased interest in the brain age-delta as a heritable metric for monitoring cognitively healthy (CH) aging and diagnosing various neurological disorders and co-morbidities [1]. The brain age-delta is defined as the difference between the chronological age and the age predicted from machine learning models trained on brain imaging data. The brain shrinks with increasing age, and there are changes at all levels, from molecules to morphology. A brain age-delta equal to zero indicates a 'healthy aging trajectory', whereas a large brain age-delta is indicative of an accelerative cognitive aging', pointing to a higher risk of age related neurological diseases or abnormal brain changes for a given age [2]. To date, brain age metric has been successfully used in the context of different neurological disorders such as

Alzheimer's disease (AD) [3] - [4], Parkinson's disease [5], Epilepsy [6], and Schizophrenia [7]. A summary of brain age estimation studies in the context of clinical application is presented in [1]. The prediction accuracy level in the brain age estimation frameworks is associated with different items such as feature extraction methods, data reduction strategies, bias correction methods, and regression algorithms. In the context of feature extraction, various neuroimaging modalities such as anatomical MRI [1], [8], [9], functional MRI [10], fluoride oxy glucose positron emission tomography imaging[3], and diffusion tensor imaging [10] can be used to extract the brain imaging features after respective preprocessing stage. Among different neuroimaging modalities, anatomical MRI is the most frequently used in brain age studies because of its widespread availability, excellent spatial resolution, and good tissue contrast. When the number of extracted features is larger than the number of samples, a data reduction technique, such as principal component analysis (PCA), can be used for 7 avoiding the curse of dimensionality [5]. In the prediction stage, a supervised learning technique (i.e., regression algorithm) is used to predict the brain age values for the given input data. The prediction model in a brain age estimation framework is vital to accurately predict the brain age values for clinical applications. The most widely used regression algorithms include Gaussian process regression [11] - [12], and support vector regression [4], [6], [8]. While considering the regression algorithm for brain age estimation, the following points should be considered: _ The algorithm should be accurate and sensitive to various data points in the training data. Generally, the performance of such models is measured on the basis of Mean Absolute Error (MAE) between the predicted age and the chronological age. _ The chosen algorithm should be able to draw a relation between naturally occurring variation, such as that caused by genetic factors. Many aspects of brain aging and susceptibility to age-related brain disease are thought to be under genetic influence. Hence, the model should be capable to "learn" these variations. _ The algorithm should be able to produce reliable results across different data-sets and patient groups. To date, few brain age studies have addressed the effects of regression algorithms on prediction accuracy in the brain age estimation frameworks [4], [13]. For instance, Valizadeh and peers [13] investigated six statistical regression algorithms (random forest, multiple linear regression, neural network, ridge regression, k-nearest neighborhood, and support vector machine) on brain age prediction results based on brain anatomical measurements (e.g. thicknesses, volumes, and cortical surfaces) among CH individuals. They reported the best results based on Neural Network and Support Vector Machine (SVM) based algorithms (R2 = 0:84) over the entire data-set. The most significant issue raised in [13] was that the effects of different regression algorithms should be assessed at the clinical level (i.e., testing on clinical populations). In order to address this issue, we conducted this study to comprehensively assess the brain age prediction results followed by various salient regression techniques (22 different algorithms in total) not only on Ch individuals but also in the clinical population (i.e., the context of neurodegeneration, such as that due 8 to AD). We also adjudge the best performing regression technique for this task, and discuss future works needed in this direction.

LITERATURE SURVEY:

The present study will conducted a systematic and rigorous evaluation of six machine learning (ML) algorithms applied to the UK Bio bank data-set for predicting brain age. Results indicate that all six models examined are suitable for this task, withT1- weighted MRI and DWI emerging as the most informative image modalities. Interestingly, our analysis suggests that image modality plays a more significant role than ML algorithm selection in determining prediction accuracy. While interaction effects between image features and ML algorithms were observed, they accounted for only a small variance. Notably, the multi-modality model outperformed uni modal ones, with the Lasso model delivering the best outcomes for mean absolute error (MAE) in multi-modality prediction. However, ensemble learning surpassed Lasso, especially when computational efficiency was not a critical factor. These insights offer valuable guidance for improving brain age prediction accuracy and understanding age-related brain changes, particularly in middle-aged and older adults.

EXISTING SYSTEM:

Recent publications have shown that training supervised regression methods on MRI brain imaging can be used to predict the brain age of an individual with high precision. We can use these predictions to detect diseases associated with abnormal brain ageing where the predicted age does not match the chronological age. In an existing system, the system develops convolutional neural network to predict brain age accurately. The architecture of the model is a simplified adaptation of the VGG architecture. Then network is trained on healthy grey matter segmented images and applied to clinical T1-weighted MRIs. The model is trained on a publicly available healthy data-set and applied to a clinical data-set consisting of Schizophrenia, Parkinson's Disease, and Post-Traumatic Stress Disorders patients. We demonstrated bias in brain age prediction, and we corrected it to improve the reliability of the results. Our Brain Age model obtained a mean absolute error (MAE) of 4.03 years and 0.96 R2 on the healthy data-set after correcting the bias. We used transfer learning to apply the Brain Age model to the clinical data and compared the brain age delta(predicted age – chronological age) for each condition. The results were not statistically significant $p<=0.5$ meaning that the brain age delta does not indicate abnormal brain ageing in this instance.

PROPOSED SYSTEM:

The prediction model in a brain age estimation framework is vital to accurately predict the brain age values for clinical applications. The most widely used regression algorithms include Gaussian process regression, and support vector regression. While considering the regression algorithm for brain age estimation, the following points should be considered:

_ The algorithm should be accurate and sensitive to various data points in the training data. Generally, the performance of such models is measured on the basis of Mean Absolute Error (MAE) between the predicted age and the chronological age.

_ The chosen algorithm should be able to draw a relation between naturally occurring variation, such as that caused by genetic factors. Many aspects of brain aging and

susceptibility to age-related brain disease are thought to be under genetic influence. Hence, the model should be capable to "learn" these variations.

_ The algorithm should be able to produce reliable results across different datasets and patient groups.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY :

This study is carried out to check the economic impact that the system will have another organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY :

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY :

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEMDESIGN:

SYSTEM ARCHITECTURE:

The system architecture encompasses a healthcare data management and analysis platform, with the service provider managing user access and functionalities like login and data retrieval. A web server hosts the platform, facilitating communication and interaction between users and the system components. Datasets are stored for browsing and analysis, with results stored for retrieval and display, including trained and tested accuracy show cased through bar charts. Users can access and process healthcare data, analyze brain age types, and view ratios thereof. The system also supports predictive modeling, enabling users to download financial type predicted data-sets. A web database stores user, data-set, and result information, ensuring efficient data management. Overall, the architecture provides acomprehensive solution for healthcare data analysis and prediction, catering to both local and remote users.

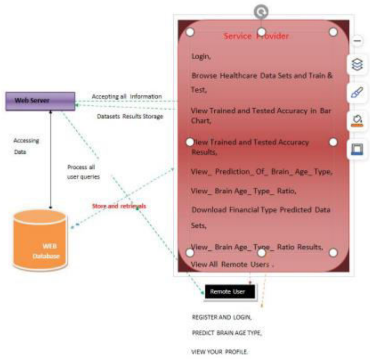


Fig1: SYSTEM ARCHITECTURE

Class Diagram :

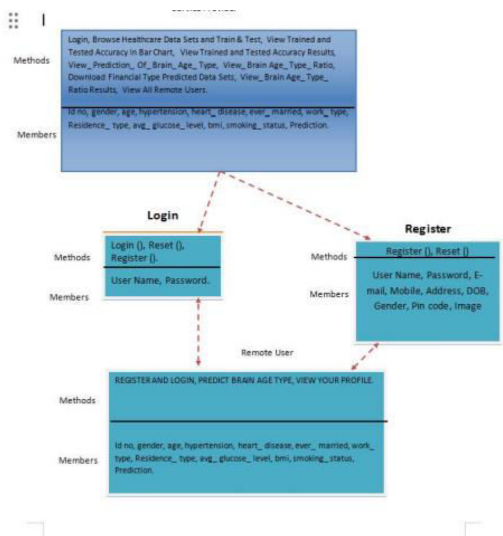


Fig2: class diagram

Data Flow Diagram :

In this data flow is like your digital ID within the system, offering a snapshot of your personal information and preferences. It typically includes details such as your name, contact information, and notification preferences. You can also find any customizations you've made to your account settings, like display options or language preferences. It serves as a convenient way to manage and update your information within the platform. Accessing your profile allows you to review and potentially modify these details, ensuring they're up-to-date and tailored to your preferences. It's a centralized hub where you can control how you interact with the system and how it interacts with you, making your user experience smoother and more personalized.

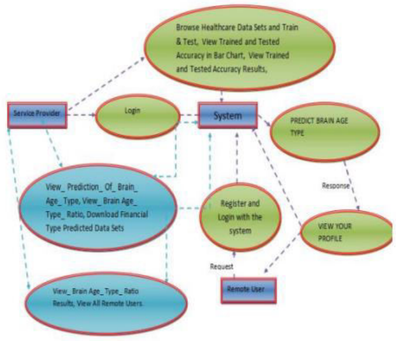


Fig4:data flow diagram

SYSTEM FLOW :

USE CASE:

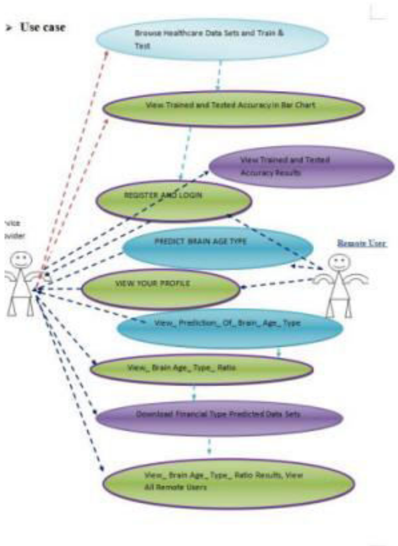


Fig5:use case

- Data Handling and ML Model Training:
- Gather healthcare data sets.
- Preprocess and split data.
- Train ML models for brain age type prediction. Visualization:

- Display accuracy using a bar chart.
- Show accuracy results.
- User Management:
- Implement registration and login. Functionality:
- Allow brain age type prediction.
- Provide profile viewing.
- Enable downloading predicted data sets.
- Documentation:
- Provide information about the use case diagram.
- Each step involves implementing specific functionalities using appropriate tools and frameworks.
- Sequence Diagram:
- Web Serve is a platform offering predictive analytics for brain age types and related healthcare data analysis. Users can register and log in to access features like predicting brain age types, browsing healthcare data-sets for training and testing models, and viewing their profiles. The platform provides visualizations such as bar charts to display the accuracy of trained and tested models, as well as detailed results for further analysis. Users can also download predicted financial type data-sets and view the distribution of brain age types within the data. Additionally, Web Serve facilitates user interaction by allowing them to view a list of all remote users on the platform.

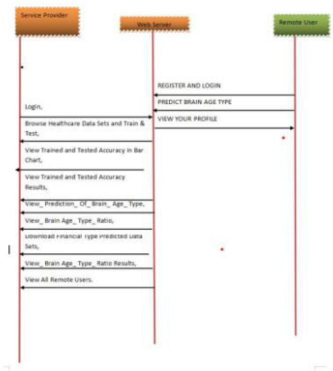
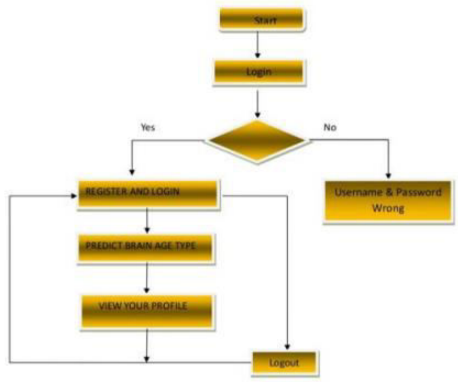
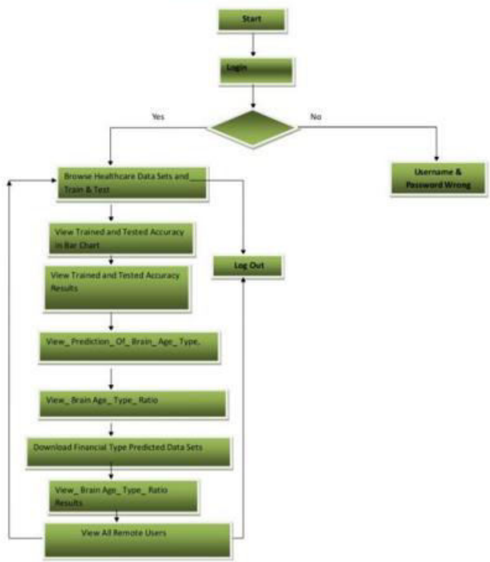


Fig6: Sequence Diagram

- **Flow Chart : Remote User**



- **Flow Chart** : Service Provider



Modules:

Service Provider :

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Browse Healthcare Data Sets and Train &Test, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View_ Prediction_ Of_ Brain _Age _Type, View_ Brain Age_ Type_ Ratio, Download Financial Type Predicted Data Sets, View_ Brain Age_Type_ Ratio Results, View All Remote Users.

View and Authorize:

Users In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User :

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT BRAINAGETYPE, VIEW YOUR PROFILE.

IMPLEMENTATION :

PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

• Python is Interpreted:

Python is processed at run time by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

• Python is Interactive:

You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

• Python is Object-Oriented:

Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

• Python is a Beginner's Language:

Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

SYSTEM TESTING :

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in a nun acceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS :

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions testedare available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items: Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures:

interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined. 62 System Test System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a blackbox .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works. Unit Testing: Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. Test strategy and approach Field testing will be performed manually and functional tests will be written in detail. Test objectives

• All field entries must work properly.

• Pages must be activated from the identified link.

• The entry screen, messages and responses must not be delayed. Features to be tested

• Verify that the entries are of the correct format

• No duplicate entries should be allowed

•All links should take the user to the correct page.

Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error. Test Results: All the test cases mentioned above passed successfully. No defects encountered. Acceptance Testing User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

SYSTEM TESTING TESTING METHODOLOGIES The following are the Testing Methodologies:

o Unit Testing.

o Integration Testing.

o User Acceptance Testing.

o Output Testing.

o Validation Testing.

Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1.**Top Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner. In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. **Bottom-up Integration :**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps: ♣ The low-level modules are combined into clusters into clusters that perform a specific Software sub-function. ♣ A driver (i.e.) the control program for testing is written to coordinate test case input and output. ♣ The cluster is tested. ♣ Drivers are removed and clusters are combined moving upward in the program structure The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

CONCLUSION:

This study aimed to comprehensively evaluate various regression models for estimating Brain Age not only on CH individuals but also in clinical population. We assessed 22 different regression models on a dataset comprising CH individuals as a training set. We then quantified each regression model on independent test sets composed of CH individuals, MCI subjects, and AD patients. Our comprehensive evaluation suggests that the type of regression algorithm affects down stream comparisons between groups, and caution should be taken to select the regression model in clinical settings.

REFERENCES:

[1] K. Franke and C. Gaser, “Ten years of brain age as a neuro imaging biomarker of brain aging: what insights have we gained?” Frontiers in Neurology, vol. 10, p. 789, 2019.

[2] J. H. Cole, S. J. Ritchie, M. E. Bastin, M. V. Hern´andez, S. M. Mani ega, N. Royle, J. Corley, A. Pattie, S. E. Harris, Q. Zhang, et al., “Brain age predicts mortality,” Molecular Psychiatry, vol. 23, no. 5, pp. 1385–1392, 2018.

[3] I. Beheshti, S. Nugent, O. Potvin, and S. Duchesne, “Disappearingmetabolic youthfulness in the cognitively impaired female brain,”Neurobiology of Aging, 2021.

[4] K. Franke, G. Ziegler, S. Kl¨oppel, C. Gaser, and A. D. N. Initiative, “Estimating the age of healthy subjects from T1-weighted MRI scans using kernel methods: exploring the influence of various parameters,”Neuroimage, vol. 50, no. 3, pp. 883–892, 2010.

[5] I. Beheshti, S. Mishra, D. Sone, P. Khanna, and H. Matsuda, “T1- weighted MRI-driven brain age estimation in alzheimer’s disease andparkinson’s disease,” Aging and Disease, vol. 11, no. 3, p. 618, 2020.

[6] D. Sone, I. Beheshti, N. Maikusa, M. Ota, Y. Kimura, M. Koepp, andH. Matsuda, “Neuroimaging-based brain-age prediction in diverse formsof epilepsy: a

signature of psychosis and beyond," Molecular Psychiatry, 06 2019.

[7] I. Nenadi´c, M. Dietzek, K. Langbein, H. Sauer, and C. Gaser, "Brainage score indicates accelerated brain aging in schizophrenia, but not bipolar 73 disorder," Psychiatry Research: Neuroimaging, vol. 266, pp. 86–89, 2017.

[8] I. Beheshti, P. Gravel, O. Potvin, L. Dieumegarde, and S. Duchesne, "A novel patch-based procedure for estimating brain age across adulthood,"NeuroImage, vol. 197, pp. 618–624, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811919304173

[9] A. Cherubini, M. E. Caligiuri, P. P´eran, U. Sabatini, C. Cosentino, and F. Amato, "Importance of multimodal MRI in characterizing braintissueand its potential application for individual age prediction," IEEEJournal of Biomedical and Health Informatics, vol. 20, no. 5, pp. 1232–1239, 2016.

[10] J. H. Cole, "Multimodality neuroimaging brain-age in UKbiobank: relationship to biomedical, lifestyle, and cognitive factors," Neurobiologyof Aging, vol. 92, pp. 34–42, 2020.

[11] J. H. Cole, R. Leech, D. J. Sharp, and A. D. N. Initiative, "Predictionof brain age suggests accelerated atrophy after traumatic brain injury,"Annals of Neurology, vol. 77, no. 4, pp. 571–581, 2015.

[12] J. H. Cole, J. Underwood, M. W. Caan, D. De Francesco, R. A. vanZoest, R. Leech, F. W. Wit, P. Portegies, G. J. Geurtsen, B. A. Schmand, et al., "Increased brain-predicted aging in treated HIVdisease,"Neurology, vol. 88, no. 14, pp. 1349–1357, 2017.

[13] S. Valizadeh, J. H¨anggi, S. M´erillat, and L. J¨ancke, "Ageprediction on the basis of brain anatomical measures," Human Brain Mapping, vol. 38, no. 2, pp. 997–1008, 2017. 74

[14] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," Advances in Neural InformationProcessing Systems, vol. 9, pp. 155–161, 1997.

[15] C. Cortes and V. Vapnik, "Support-vector networks," MachineLearning, vol. 20, no. 3, pp. 273–297, 1995.

[16] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and Regression Trees. CRC press, 1984.

[17] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classificationand regression-recent developments, applications and future directions,"IEEE Computational intelligence magazine, vol. 11, no. 1, pp. 41–53, 2016.

[18] M. A. Ganaie, M. Hu, M. Tanveer, and P. N. Suganthan, "Ensembledeep learning: A review," arXiv preprint arXiv:2104.02395, 2021.

[19] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.

[20] Y.-H. Shao, C.-H. Zhang, Z.-M. Yang, L. Jing, and N.-Y. Deng, "An"-twin support vector machine for regression," Neural Computing and Applications, vol. 23, no. 1, pp. 175–185, 2013.

[21] M. Tanveer, T. Rajani, R. Rastogi, and Y. H. Shao, "Comprehensivereview on twin support vector machines," arXiv preprint arXiv:2105.00336, 2021.

[22] R. Tibshirani, "Regression shrinkage and selection via the lasso,"Journal of the Royal Statistical Society: Series B

(Methodological), vol. 58, no. 1, pp. 267–288, 1996. 75

[23] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biasedestimation for nonorthogonal problems,” Technometrics, vol. 12, no. 1, pp. 55–67, 1970.

[24] C. E. Rasmussen, “Gaussian processes in machine learning,”inSummer School on Machine Learning. Springer, 2003, pp. 63–71.

[25] C. Saunders, A. Gammerman, and V. Vovk, “Ridge regressionlearning algorithm.