Age prediction using blood laboratory data.

By

Phu Nguyen (015970994) , Kashyap Tamakuwala (015953353)

Introduction

Motivation

Aging has a tremendous effect on the human anatomy, this is especially shown through blood laboratory data such as testosterone and plasma fibrinogen. As the world population experiences aging every day, it is important to study how age relates to blood data.

Predictors for one age group may fail to generalize to other groups and investigate non-linearity in biomarkers near adulthood. As populations worldwide undergo major demographic changes, it is increasingly important to catalogue biomarker variation across age groups and discover new biomarkers to distinguish chronological and biological aging.

Objective

Our aim is to systematically study the predictive capacity of individual and large collections of blood laboratory biomarkers for predicting chronological age across the lifespan.In this we create regression models that make age predictions based on blood laboratory biomarkers.

System Design & Implementation details

Data Collection

We collected blood laboratory analyte measurements and demographic data from nine waves of the Centers for Disease Control and Prevention (CDC) National Health and Nutrition Examination Survey (NHANES) including the following cohorts: 1999-2000, 2001-2002, 2003-2004, 2005-2006, 2007-2008, 2009-2010, 2011-2012, 2013-2014, 2015-2016 , 2017- 2018,2019-2020.

For collecting data from the site we used python3 and beautifulsoup4.

Data Preparation

After Scraping data from the National Health and Nutrition Examination Survey (NHANES) website our main job was to merge data of different lab results based on the sequence number (representing a individual) for each year and then to create a consistent dataset by combining the data generated for each year into a single file.

After creating the dataset we observed individuals with very few lab tests and also noticed certain laboratory tests were less ordered so most of the values for those columns were null.Therefore during preprocessing Laboratory test with greater than 95% missingness were removed, and individuals with fewer than 20 measured labs were also removed.

We imputed all missing values using mean imputation, where each missing value was replaced by the mean value for that analyte over all individuals.

For processing and cleaning the data we took help from the following libraries

- Pandas
- Numpy
- Sickit-learn
- python3.

For Visualizing data we used seaborn, matplotlib and yellow-brick

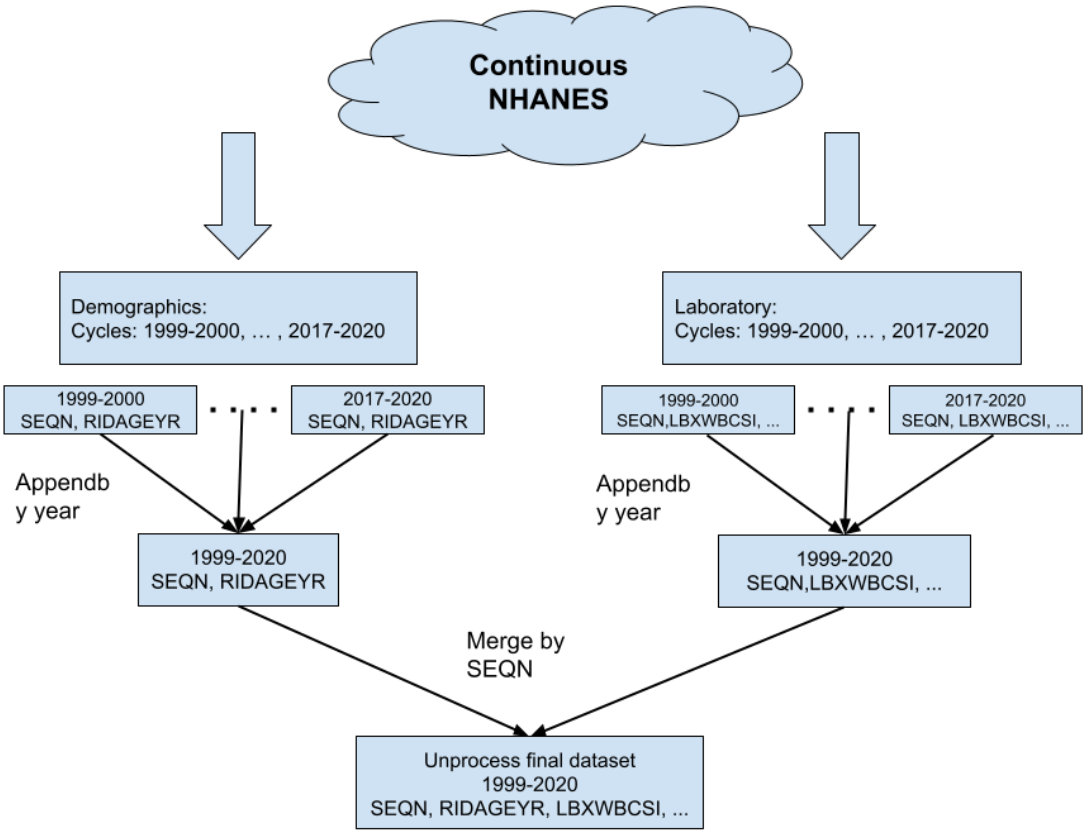

Figure 1 → Data Collection and Preparation Diagram

Algorithms

For this regression problem we considered using 3 different algorithms-

1. **Random Forest Regressor**

   A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

It is considered because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models.

2. **Epsilon-Support Vector Regression (SVR)**

   SVR gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data.

   The objective function of SVR is to minimize the coefficients — more specifically, the *l*2-norm of the coefficient vector — not the squared error.

3. **Xgboost.**

   Gradient boosting is one of the most powerful techniques for building predictive models, and it is called a Generalization of AdaBoost. The main objective of Gradient Boost is to minimize the loss function by adding weak learners using a gradient descent optimization algorithm. The generalization allowed arbitrary differentiable loss functions to be used, expanding the technique beyond binary classification problems to support regression, multi-class classification and more.

Experiments / Proof of concept evaluation

Dataset

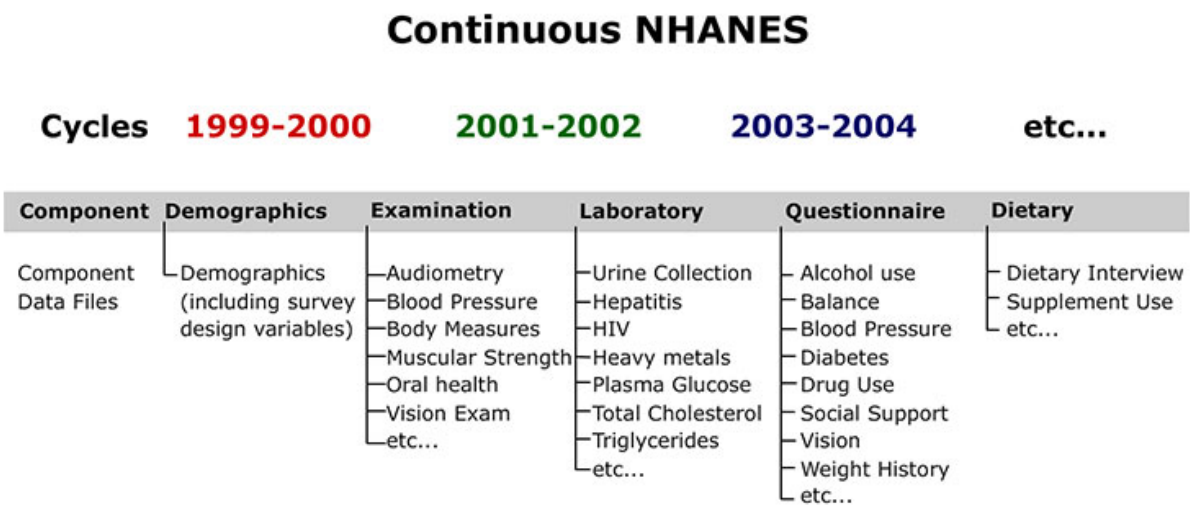

Figure 2 → (https://wwwn.cdc.gov/nchs/nhanes/tutorials/module1.aspx)

We use the National Health and Nutrition Examination Survey NHANES dataset, under the Continuous NHANES (Figure 2.). We collected cycles from 1999 to 2020. From these cycles, we only download the Demographics dataset which includes the age column and the Laboratory dataset which consists of the necessary blood test data.
The final dataset's attributes include:

- SEQN: from Demographics and Laboratory.
- RIDAGEYR (Age) from Demographics.
- 356 descriptors from Laboratory. For this, we consult some insights provided by the Prediction of chronological and biological age from laboratory data paper (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7244024/)

The final dataset after the data collection step had 106203 rows and 402 columns.After cleaning the dataset contained 92492 rows and 326 columns.

Visualization and Statistic of response variable (RIDAGEYR)

Statistics for " RIDAGEYR "

Mean	STD	Max	Min	25%	50%	75%
35.146	24.055	85	1	14	31	55

Distribution for RIDAGEYR is Shown below

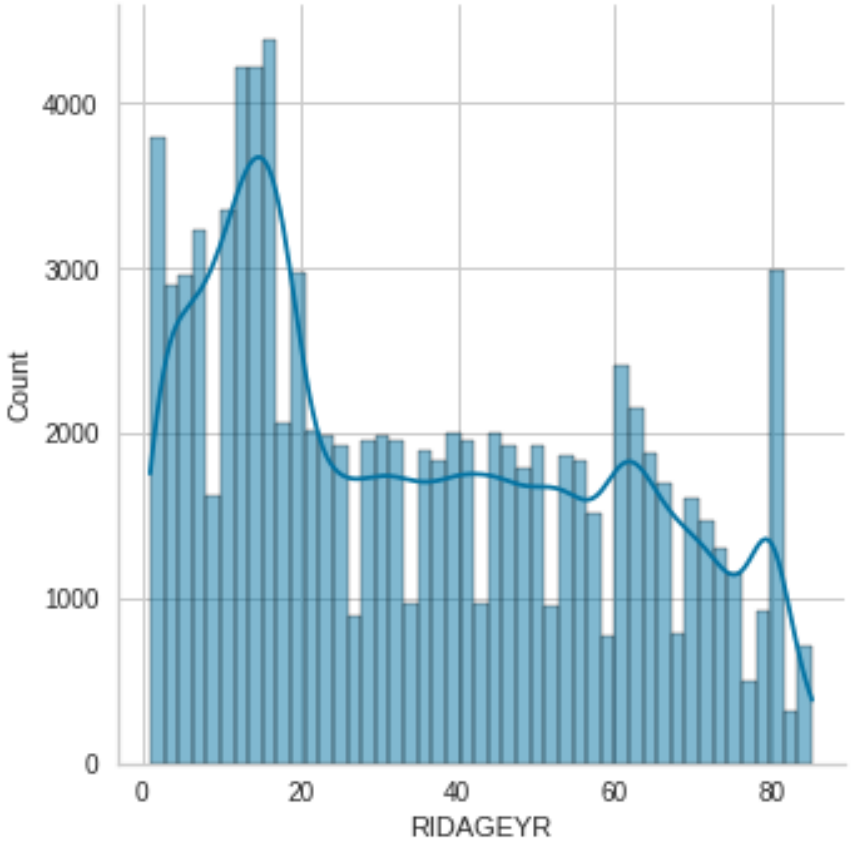

Figure 3 → Distribution of RIDAGEYR .

Methodology followed

For this project, we chose to split our training data and testing data as 80% and 20%.

Regression model

Random Forest Regressor

We first take a look at the parameters for this model: n_estimators, max_features, bootstrap. We use RandomizedSearchCV to try different combinations. In the end, we find that Random Forest Regressor returns best results with n_estimators= 100, max_features= 100, bootstrap= False. The R2 score for this configuration is 0.923. The plots below show the prediction error of the Random Forest Regressor model and the feature importance rankings.

For cross validation, we use the cross_val_score method from sklearn. We choose negative mean square error as our metric and the cross validation will be carried out over 10 folds. At the end, we got the mean of -negative mean square error as 4.6.

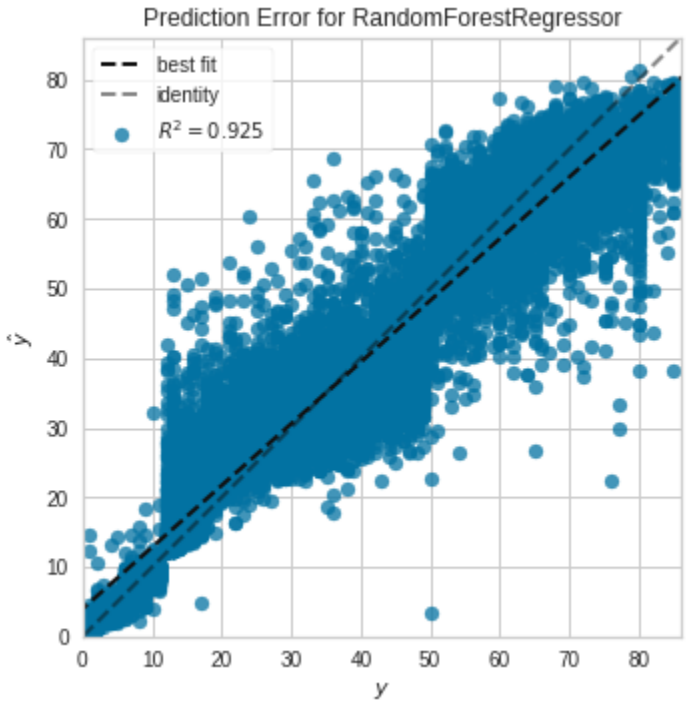

Figure 4 → Prediction Error for Random Forest Regressor

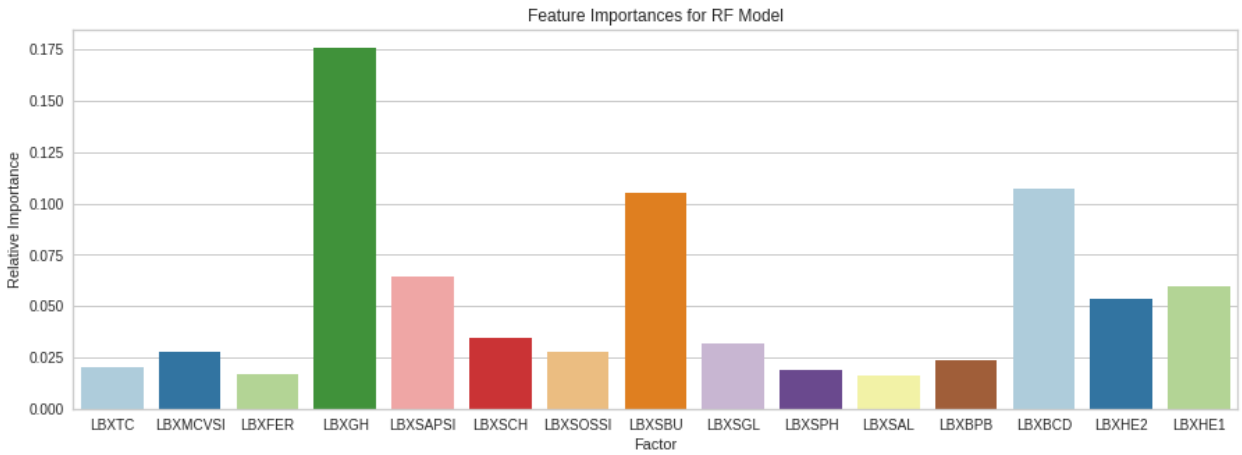

Figure 5 → Feature Importance for Random Forest Regressor

Gradient Boosting Regressor

Very similar to the Random Forest Regressor model, The parameter configurations we use for this model are max_depth= 14.0, max_features= 25, n_estimators= 1000, random_state= 1, subsample= 1. This configuration gives us a R2 score of 0.918. The plots below show the prediction error of the Gradient Boosting Regressor model and the feature importance rankings.

For cross validation, similar to the random forest model, we choose negative mean square error as our metric and the cross validation will be carried out over 10 folds. At the end, we got the mean of -negative mean square error as 4.9.

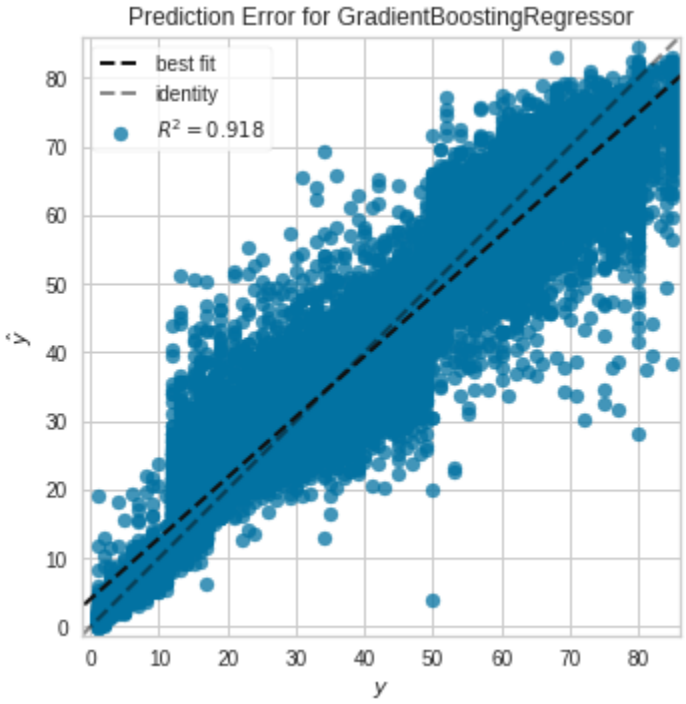

Figure 6 → Prediction Error for XGBOOST

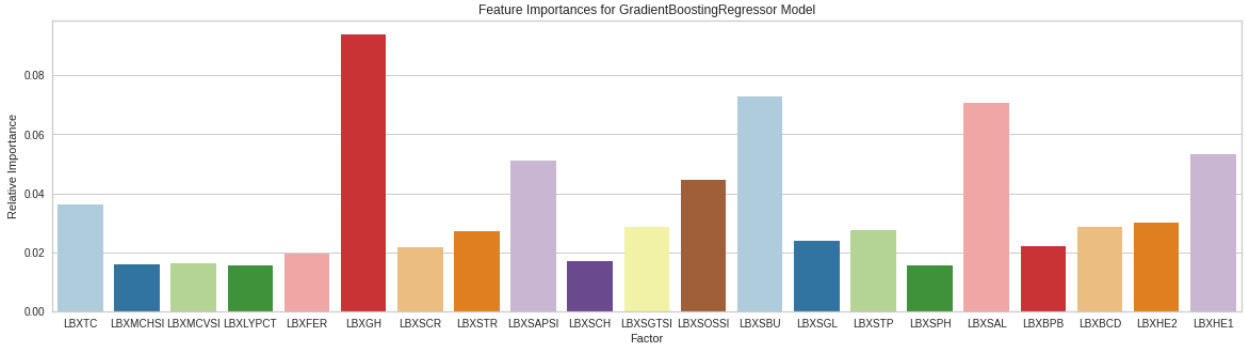

Figure 7 → Feature Importance for Random Forest Regressor

Discussion & Conclusions

Decisions made

For the final project, we initially wanted to look into the topic of health longevity and population aging clocks, diseases related to aging. However, after some research, we decided that this topic is broader than the scope of the class and is surely over our time constraints. We agreed to reduce the scope of the project to using blood laboratory data to predict the age of an individual.

Difficulties faced

We encountered some difficulties with collecting data from NHANES website. We had to take some time to familiarize ourselves with NHANES data scheme.

Things that worked

At the beginning, we planned to use some NHANES api available out there for collecting data. However, we encountered difficulties with the interfaces of these APIs. We eventually decided to write our own web scraper script. This turned out to be much easier.

Conclusion

In this project, we built a machine learning regression model to predict the age of an individual through blood laboratory data. The data is collected from the National Health and Nutrition Examination Survey (NHANES) website under the continuous section. To clean and preprocess the raw data, besides, dropping missing data columns and rows, we applied mean imputation. We then standardized the dataset to prepare for the model training phase. For this study, we used the Random Forest Regressor model which achieved 0.925 R2 score and Gradient Boosting Regressor which achieved 0.918 R2 score.

Project Plan / Task Distribution

We started the project early on because we only have 2 members in our group. The task is distributed equally among members. We got together once or twice every week to work on the project together. The full implementation can be found on https://github.com/phunguyen1195/Age-prediction-NHANES