Predicting Chronological Age from Health and Lifestyle Factors Using Machine Learning

Nikhil Gokhale
Ubit no.- 50560271

Prathamesh Ravindra Varhadpande
Ubit no.- 50559586

Nachiket Dabhade
Ubit no.- 50540189

A. Selecting Best Model for Chronological Age Prediction

The task of predicting chronological age was addressed by testing and optimizing various machine learning models, with XGBoost emerging as the top performer due to its exceptional accuracy, scalability, and capacity to handle high-dimensional datasets. Performance was rigorously evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R^2). Extensive hyperparameter tuning through GridSearchCV further refined the model, ensuring an optimal configuration for precise and reliable predictions. Thus, Gradient Boosting Model was selected to implement our Webapp on Streamlit.

B. Streamlit Model Building

a. Storing Gradient Boosting Algorithm parameters in pickle file

Initially, we implemented an end-to-end pipeline for training and saving an XGBoost regression model to predict a target variable, `Age (years)`, from a dataset loaded from `predict.csv`. It begins by splitting the data into training and testing sets and uses `RandomizedSearchCV` for hyperparameter tuning, optimizing parameters such as `n_estimators`, `max_depth`, `learning_rate`, `subsample`, and `colsample_bytree` through cross-validation. After identifying the best model configuration, it evaluates the model's performance on the test set using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2). Finally, the best-performing XGBoost model is saved to a file named `best_xgb_model.pkl` using the `joblib.dump` function, ensuring the trained model's parameters and state are stored for later use without the need to retrain it.

Reference code file: pvarhadp_gokhale7_dabhade_phase2.py

b. Streamlit model building

The application collects user inputs for predicting age by prompting users to provide values for a range of features. These features include both continuous and categorical types, and specific processing steps are applied to ensure the inputs align with the format expected by the machine learning model.

For **continuous features** such as cholesterol level, BMI, blood glucose level, bone density, and cognitive function, the system displays normalization formulas derived from the minimum and maximum values of the training dataset. This guidance helps users preprocess their data manually to ensure the values are scaled appropriately. The normalized inputs are then entered into the system via numerical input fields.

For **categorical features** such as gender, smoking status, physical activity level, diet, and chronic diseases, the system uses one-hot encoding to transform these variables into binary representations. For example, a selection of "female" for gender results in Gender_female = 1 and Gender_male = 0. Features with multiple categories (e.g., smoking status with options like "current," "former," and "never") are encoded as mutually exclusive binary variables to avoid multicollinearity and maintain consistency with the training data format.

The application dynamically generates a form using Streamlit to collect the inputs. Continuous features are captured through number_input fields, while categorical features are collected via dropdown menus (selectbox) with predefined options. This approach prevents invalid entries and simplifies the input process for users.

Data Preparation and Model Compatibility

Once all inputs are collected, they are structured into a DataFrame where each column corresponds to a feature expected by the machine learning model. Continuous feature values are directly placed in their respective columns, and categorical features are encoded as binary variables (1 for selected categories, 0 otherwise). This DataFrame mirrors the schema of the dataset used for training, ensuring seamless compatibility with the model during inference.

Model Parameters and Age Prediction

The application leverages pre-trained model parameters stored in a pickled file (best_xgb_model.pkl) to make predictions. The pickled file contains the following key elements:

The list of selected features obtained during the feature selection process.
Optimal hyperparameters, including the number of hidden units and the regularization coefficient (λ).
Weight matrices w1 and w2 for the model layers.

During the prediction process, the input DataFrame is passed through the model, utilizing the stored parameters to compute the output. The age prediction is generated as the final output, which is then displayed to the user in a user-friendly format. By saving the model parameters in a pickled file, the application ensures efficiency, as the model does not need to be re-trained or re-initialized each time it is used for predictions.

Benefits of the Approach

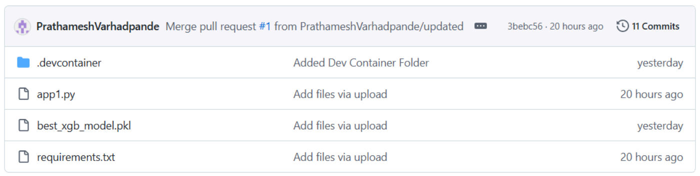
This system ensures a robust, consistent, and user-friendly experience. By aligning the input handling and feature processing with the model's training data, the application minimizes errors and maintains high accuracy. Furthermore, the use of pickled model parameters reduces computational overhead, making the prediction process both efficient and reliable.

Reference code file: app1.py

C. Model Hosting on Streamlit

To streamline deployment and version control, we first committed our `app1.py` file, `best_xgb_model.pkl` file and requirements.txt file to our GitHub repository at https://github.com/PrathameshVarhadpande/chronological-age . We then connected the repository to Streamlit Community Cloud, enabling seamless integration and deployment of our application. The app, hosted on https://chronoapp.streamlit.app/, provides an interactive platform for predicting chronological age using our pre-trained XGBoost model. This approach ensures easy access, maintainability, and scalability for future updates.

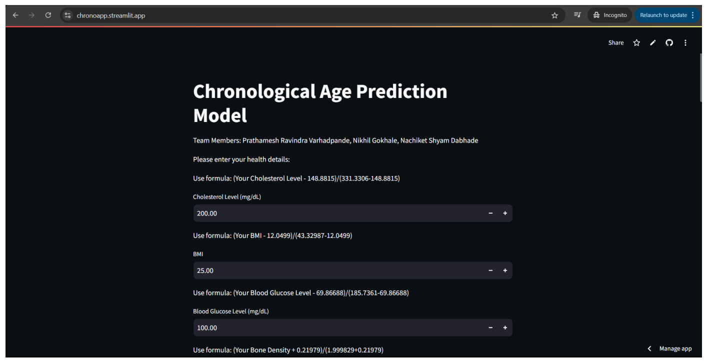
Github repository:



PrathameshVarhadpande Merge pull request #1 from PrathameshVarhadpande/updated 3bebc56 · 20 hours ago 11 Commits

.devcontainer	Added Dev Container Folder	yesterday
app1.py	Add files via upload	20 hours ago
best_xgb_model.pkl	Add files via upload	yesterday
requirements.txt	Add files via upload	20 hours ago

Streamlit Webapp:



Chronological Age Prediction Model

Team Members: Prathamesh Ravindra Varhadpande, Nikhil Gokhale, Nachiket Shyam Dabhade

Please enter your health details:

Use formula: (Your Cholesterol Level - 148.8815)/(331.3306-148.8815)

Cholesterol Level (mg/dL): 200.00

Use formula: (Your BMI - 12.0499)/(43.32987-12.0499)

BMI: 25.00

Use formula: (Your Blood Glucose Level - 69.86688)/(185.7361-69.86688)

Blood Glucose Level (mg/dL): 100.00

Use formula: (Your Bone Density + 0.21979)/(1.999829+0.21979)

b. Demonstration and Use Case Analysis:

To demonstrate the functionality of the model, we entered a set of demo values from our dataset to validate the proximity of the predicted age to the actual age. The demo input values were filled for all features, with the following values selected:

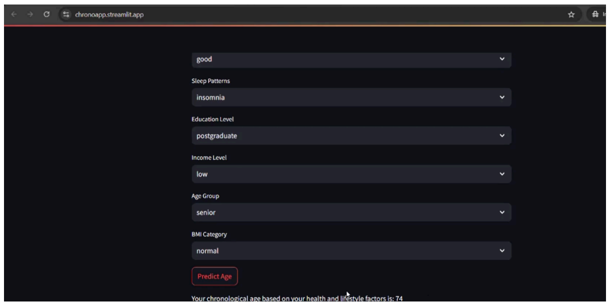
Numerical Features:

Cholesterol Level: 0.60
BMI: 0.14
Blood Glucose Level: 0.40
Bone Density: 0.36
Vision Sharpness: 0.08
Hearing Ability: 0.60
Cognitive Function: 0.43
Stress Levels: 0.10

Pollution Exposure: 0.98
Sun Exposure: 0.44
Systolic BP: 0.55
Diastolic BP: 0.66

Categorical Features:

Gender: Female
Physical Activity Level: High
Smoking Status: Current
Alcohol Consumption: Occasional
Diet: Balanced
Chronic Diseases: Diabetes
Medication Use: Occasional
Family History: Diabetes
Mental Health Status: Good
Sleep Patterns: Insomnia
Education Level: Postgraduate
Income Level: Low
Age Group: Senior
BMI Category: Normal



good
Sleep Patterns: insomnia
Education Level: postgraduate
Income Level: low
Age Group: senior
BMI Category: normal

Your chronological age based on your health and lifestyle factors is: 74

Prediction and Analysis

Upon entering the demo input values, the model predicted the chronological age to be 74 years. However, the actual age of the individual was 69 years. This indicates that the predicted chronological age is greater than the actual age, suggesting that the individual may be aging faster than expected based on their health and lifestyle factors.

The analysis derived from this output can be valuable for healthcare professionals and fitness coaches. The model's prediction can be used as an indicator for targeted interventions, encouraging the individual to make lifestyle adjustments to slow down their biological aging. Such insights can also be used to guide the person in making informed decisions about their health, ensuring better overall well-being and quality of life.

Conclusion

In conclusion, the Chronological Age Prediction model provides an innovative way to assess a person's chronological age based on various health and lifestyle factors. By comparing the predicted chronological age to the actual age, the model can help identify individuals who may benefit from preventive healthcare strategies, personalized fitness plans, and early interventions. The insights derived from this project have the potential to drive more proactive health management strategies, benefiting both individuals and healthcare providers alike.