1. INTRODUCTION

Age estimation is an important problem that has witnessed an increased attention, given its role in various daily activities, from health assessment, to social interaction, to forensic science, to security and identity profiling. Although age estimation has been practiced for centuries, accurate age estimation is known to be a very difficult problem. Doing this automatically by a machine is an even much more onerous task. The major challenge is that most measures used to characterize age, for instance, visual appearance, and biological/physiological markers vary significantly from person to person, even for people with the same chronological age. The reason is that many unknowns (e.g., genetics, nutrition, body shape, health condition, social conditions, life style, weather, and even cultural considerations), all contribute to influence the perceived age of an individual. There are various scenarios or applications when the chronological age is required, but the true birth date is unknown, and a genuine birth certificate may not be available.

Age has a deep connection with health and mortality. Aging is a gradual process that results in increased health risk, and mortality over time. In general, a younger person is expected to have a better health condition and his/her mortality hazard should be low in comparison with a relatively older person. But two different people of the same age may have very different health conditions and mortality hazards. This brings up the debate on “chronological” versus “biological” age.

Chronological age is typically what we know and is based on the date of birth. However, biological age is based on the interesting, yet confounded, idea that a person’s true age can be different from his/her chronological age. Biological age lacks a precise definition, but it is often viewed as the true age of an individual . The common idea is that biological age provides a better estimator of the true life expectancy of the individual than chronological age.

2. LITERATURE SURVEY

2.1 Existing problem

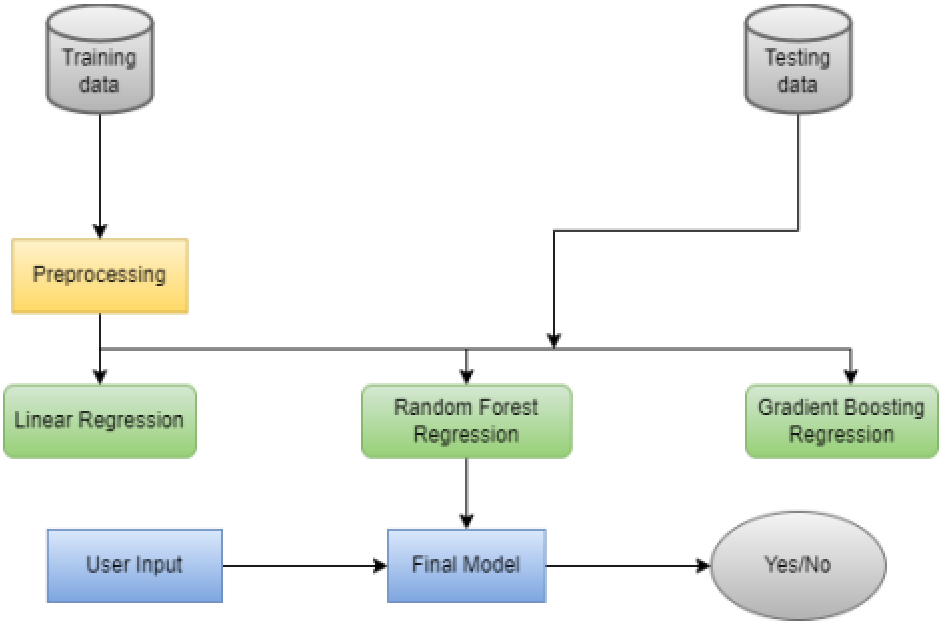
Age-related diseases are killing 150,000 people per day. Age estimation is an important medical and public health challenge. Age has a deep connection with health and mortality. Aging is a gradual process that results in increased health risk, and mortality over time. In general, a younger person is expected to have a better health condition and his/her mortality hazard should be low in comparison with a relatively older person. But two different people of the same age may have very different health conditions and mortality hazards. This brings up the debate on “chronological” versus “biological” age.

2.2 Proposed solution

Depending on your genetics and your lifestyle actions, your biological age will be higher or lower than your chronological one. Building a ML model which will provide a biological age taking methylation levels into consideration.

3. THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

Software Requirements:

Front-end: HTML, CSS, JavaScript

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) Web browsers receive HTML documents from a web server or from local storage CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content.

Back-end:

Jupyter Notebook

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. program used to mix code, comments, and visualizations in an interactive document called notebook that can be shared, reused, and reworked in a web browser. Jupyter Notebook is a web-based interactive computational environment for creating notebook documents. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text plots.

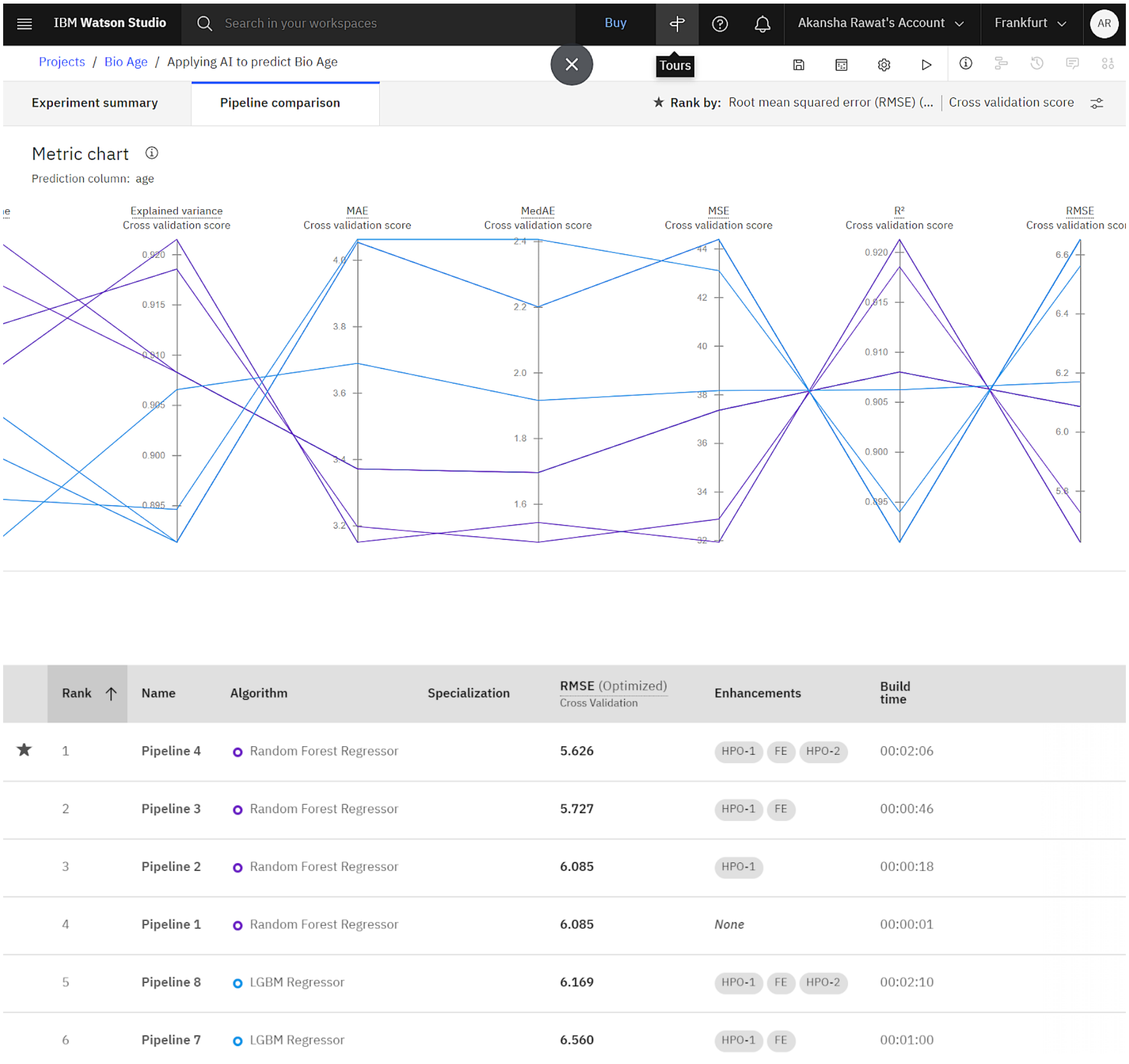
Flask

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. Flask is based on the WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine.

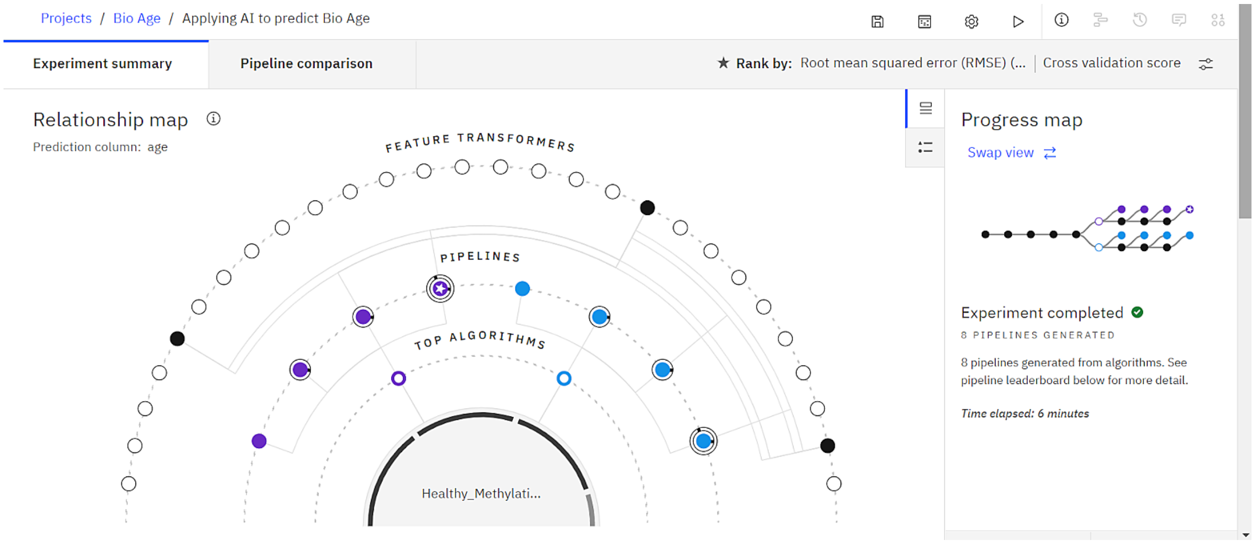
4. EXPERIMENTAL INVESTIGATIONS

Sr.No	Research Paper	Overview
1.	Biological Age Predictors	This paper aims on predicting the biological age for human based on DNAmAge and Aging Phenotypes
2.	Deep learning for biological age estimation	In this paper,the age is predicated based BA estimation methods using human biomarkers, human anthropometry and locomotor activity
3.	Prediction of biological age and all-cause mortality by 12-lead electrocardiogram in patients without structural heart disease.	This model showed a similar predictive capability to CA for all cause death and cardiovascular death among total patients, but partially showed a significant increase in the predictive capability among patients aged 60–74 years old.

5. Results



	Rank ↑	Name	Algorithm	Specialization	RMSE (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 4	○ Random Forest Regressor		5.626	HPO-1 FE HPO-2	00:02:06
	2	Pipeline 3	○ Random Forest Regressor		5.727	HPO-1 FE	00:00:46
	3	Pipeline 2	○ Random Forest Regressor		6.085	HPO-1	00:00:18
	4	Pipeline 1	○ Random Forest Regressor		6.085	*None*	00:00:01
	5	Pipeline 8	○ LGBM Regressor		6.169	HPO-1 FE HPO-2	00:02:10
	6	Pipeline 7	○ LGBM Regressor		6.560	HPO-1 FE	00:01:00



Projects / Bio Age / Applying AI to predict Bio Age

Experiment summary
Pipeline comparison
★ Rank by: Root mean squared error (RMSE) (... | Cross validation score

Relationship map

Prediction column: age

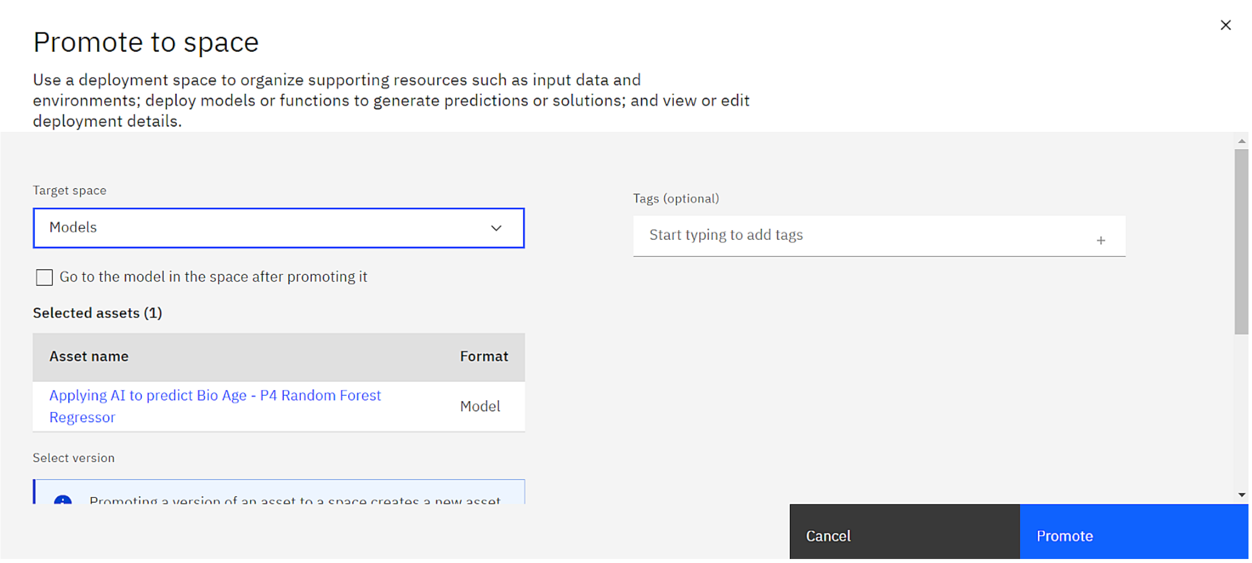
Progress map

Experiment completed ✓

8 PIPELINES GENERATED

8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.

Time elapsed: 6 minutes



Promote to space

Use a deployment space to organize supporting resources such as input data and environments; deploy models or functions to generate predictions or solutions; and view or edit deployment details.

Target space: Models

Tags (optional): Start typing to add tags

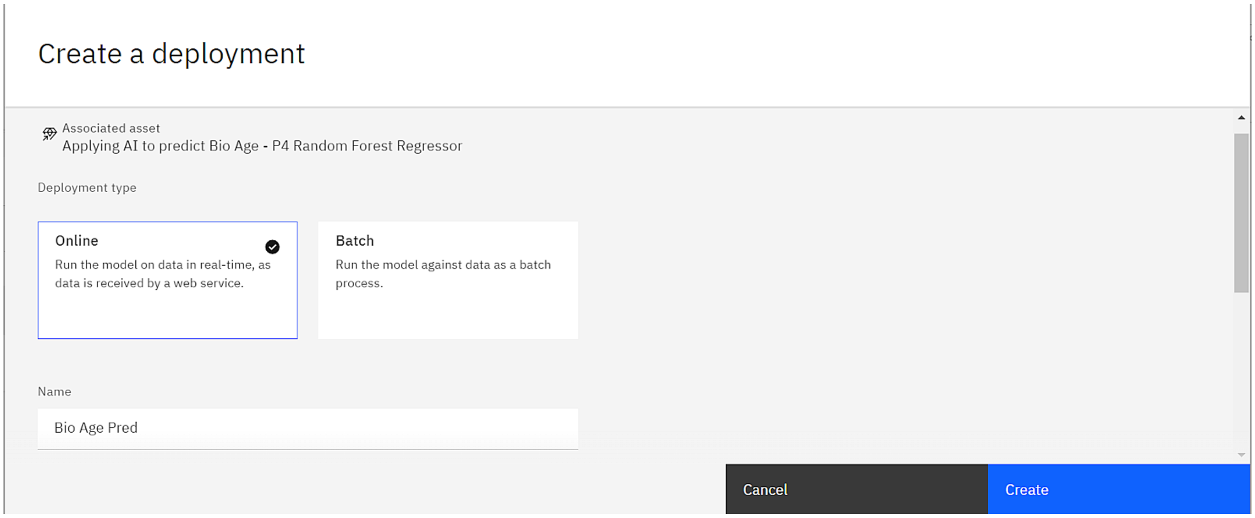
☐ Go to the model in the space after promoting it

Selected assets (1)

Asset name	Format
Applying AI to predict Bio Age - P4 Random Forest Regressor	Model

Select version

Promoting a version of an asset to a space creates a new asset



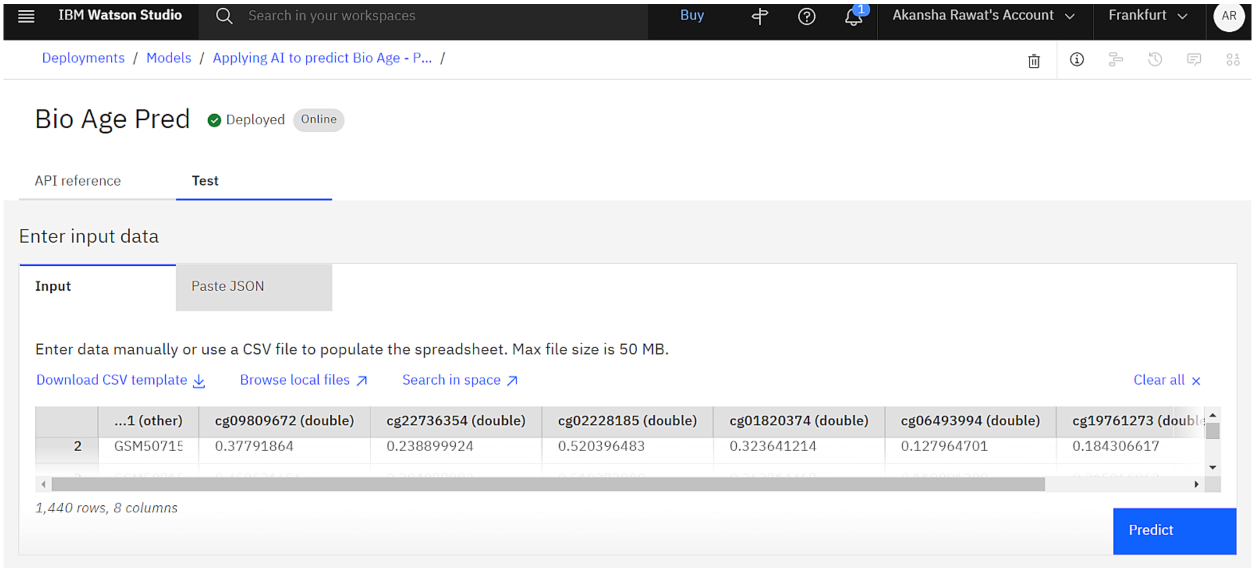
Create a deployment

Associated asset
Applying AI to predict Bio Age - P4 Random Forest Regressor

Deployment type

- ● **Online**: Run the model on data in real-time, as data is received by a web service.
- **Batch**: Run the model against data as a batch process.

Name: Bio Age Pred



IBM Watson Studio
Search in your workspaces
Buy
Akansha Rawat's Account
Frankfurt
AR
Deployments / Models / Applying AI to predict Bio Age - P... /

Bio Age Pred

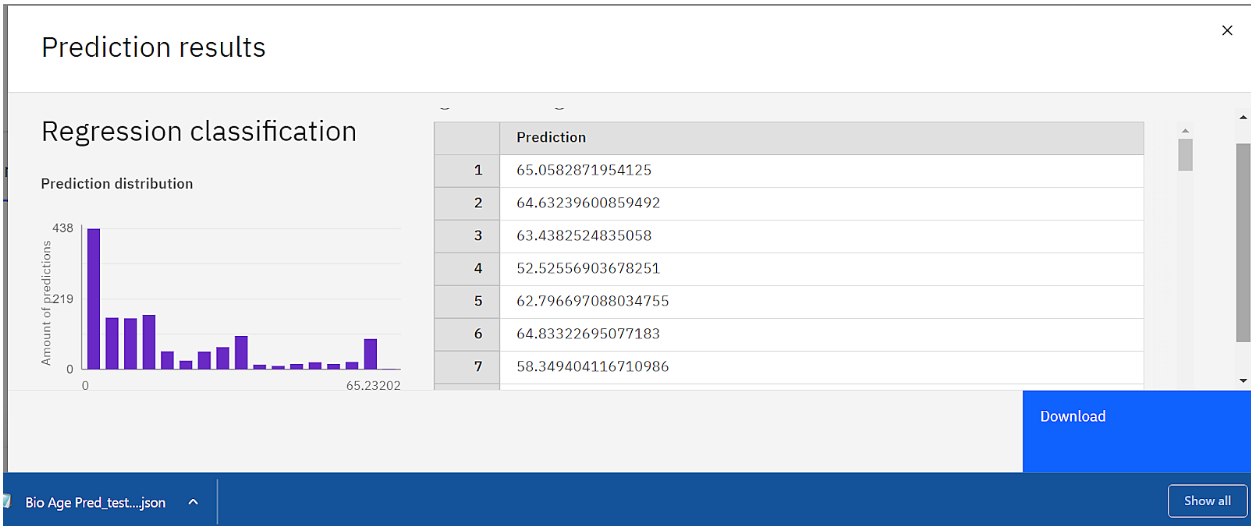
Deployed · Online

Enter input data

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

	...1 (other)	cg09809672 (double)	cg22736354 (double)	cg02228185 (double)	cg01820374 (double)	cg06493994 (double)	cg19761273 (doubl
2	GSM5071Ŀ	0.37791864	0.238899924	0.520396483	0.323641214	0.127964701	0.184306617

1,440 rows, 8 columns

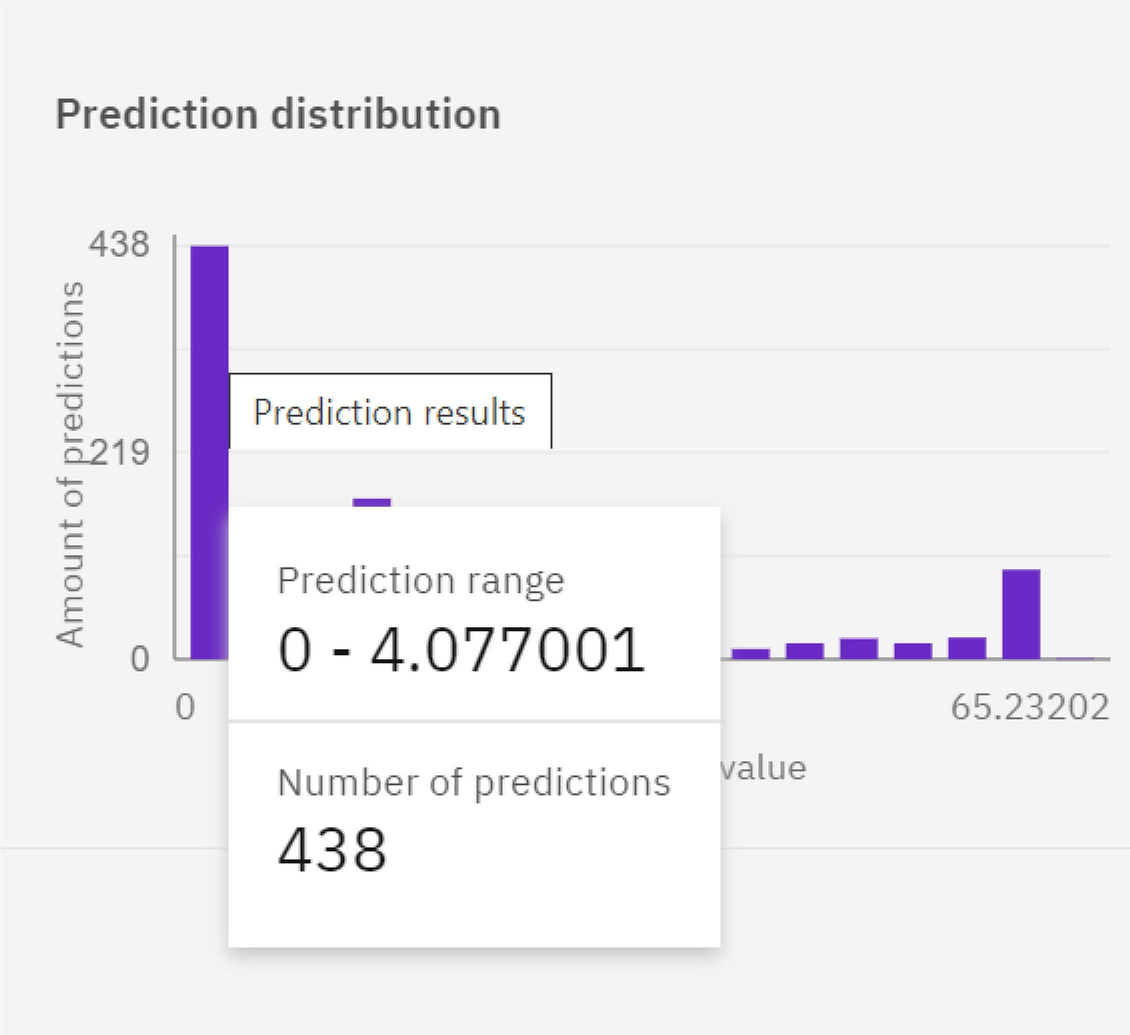


Prediction results

Regression classification

Prediction distribution

	Prediction
1	65.0582871954125
2	64.63239600859492
3	63.4382524835058
4	52.52556903678251
5	62.796697088034755
6	64.83322695077183
7	58.349404116710986



Prediction distribution

Prediction range
0 - 4.077001

Number of predictions
438

7. ADVANTAGES & DISADVANTAGES

Advantages

1. The model can accurately predict the biological age of person.
2. The model would help people understand their lifestyle
3. The project can help people to step toward a healthy lifestyle.

Disadvantages

1. Model do not use aging biomarkers so developing model became tedious.
2. Relationship between Biological age and Cardiorespiratory Fitness cannot be inferred from the model

8. APPLICATIONS

- Can be applied in researches, medicine development.
- Can be used as a useful index to predict a person's risk of death in the future.
- Can be helpful for transformation step towards a high tech-world.

9. CONCLUSION

In summary, this project provides a valid ML-based measure of biological aging for different age grouped people. We further demonstrated that this model gives best accuracy which was associated with physical disability incidence and mortality. These associations were comparable to that of a valid physiological biomarkers-based aging measure that were previously developed. The findings support the application of ML in geroscience research and promote further understanding of the aging process. Together with the model and dashboard, these aging measures could serve as a proxy of life span and help with the risk stratification in the general older adults.

10. FUTURE SCOPE

The future scope is limitless for the purpose of improving lifestyle, as humans always want to live longer, stay healthy and happy. So that project will be more useful for human beings. In this we have just developed a model. By using this system we were able to accurately predict users' biological age. Currently, we have used a random forest algorithm and flask as front-end to do so. Considering all above points, followings are our future works set to improve the system:

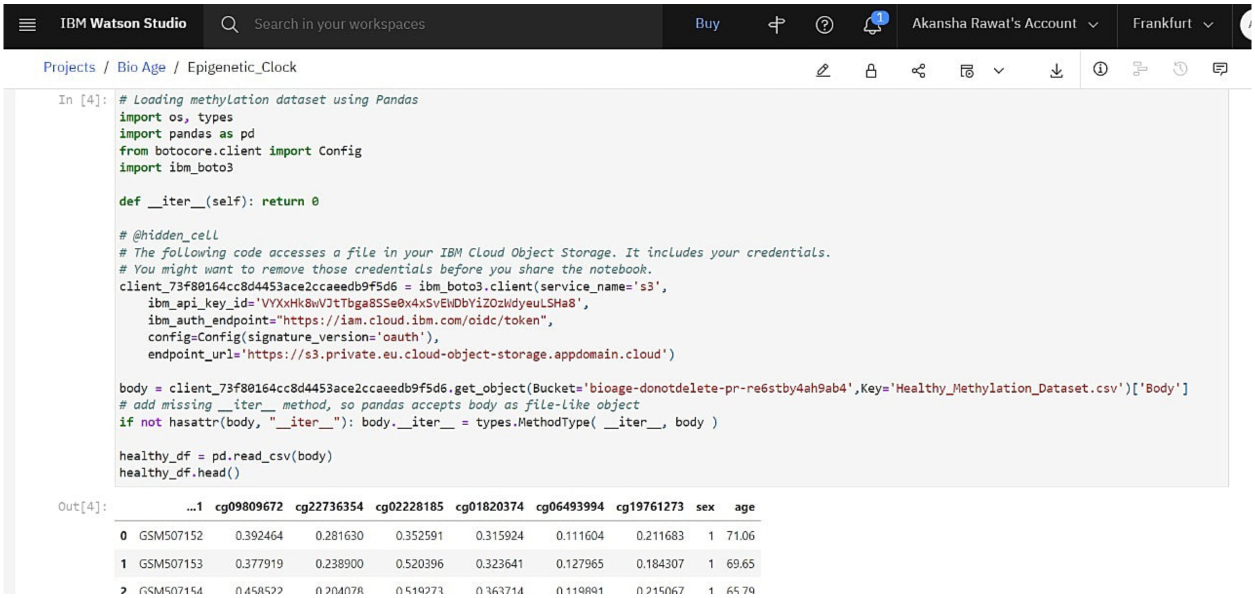
- Use Geometric deep learning to build a 3D model.
- Use of aging biomarkers development provided by AI instead of tedious multi-stage process which requires proof of concept and experimental validation.
- To study the relationship between Biological Age(BA) and Cardiorespiratory Fitness(CRF), using a deep learning framework.

11. BIBILOGRAPHY

1. “Biological Age Predictors”, Juulia Jylhävä, Nancy L. Pedersen and Sara Hägg, Research Gate, April 2017.
2. “Deep learning for biological age estimation”, Syed Ashiqur Rahman, Peter Giacobbi, Lee Pyles, Charles Mullett, Gianfranco Doretto and Donald A Adjeroh, Oxford Academic, May 2020.
3. “Prediction of biological age and all-cause mortality by 12-lead electrocardiogram in patients without structural heart disease”, Naomi Hirota, Shinya Suzuki, Naoharu Yagi and Takuto Arita, Research Gate, August 2021.

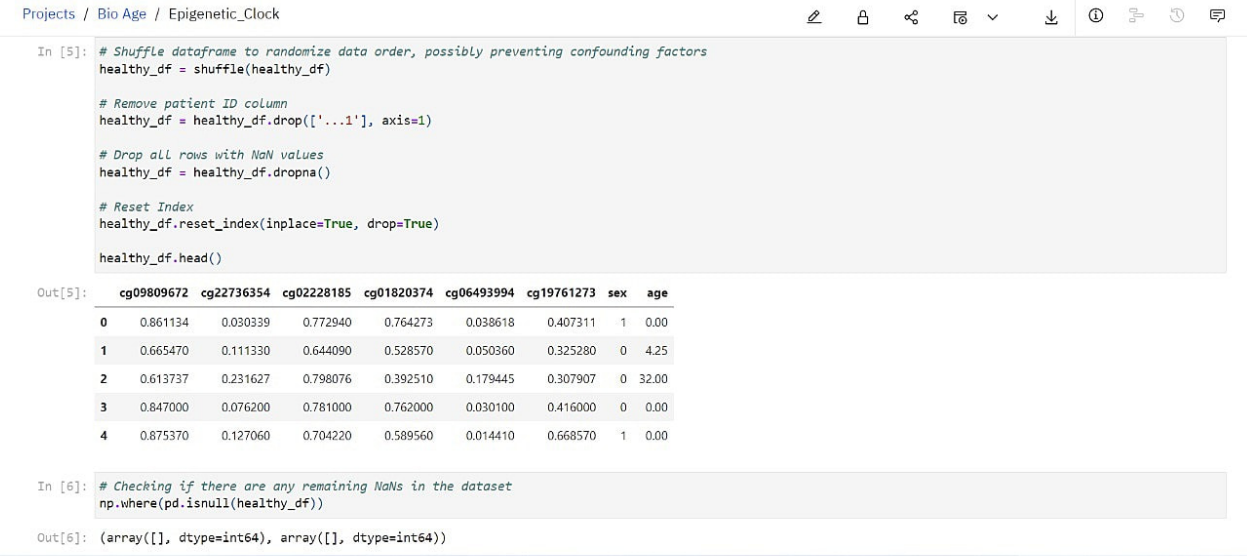
APPENDIX

A. Source Code


IBM Watson Studio
Search in your workspaces
Buy
1
Akansha Rawat's Account
Frankfurt
Projects / Bio Age / Epigenetic_Clock

```
In [4]: # Loading methylation dataset using Pandas
        import os, types
        import pandas as pd
        from botocore.client import Config
        import ibm_boto3

        def __iter__(self): return 0

        # @hidden_cell
        # The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
        # You might want to remove those credentials before you share the notebook.
        client_73f80164cc8d4453ace2ccaeedb9f5d6 = ibm_boto3.client(service_name='s3',
            ibm_api_key_id='VYXxHk8wVJtTbga8SSe0x4xSvEWDbYiZOzWdyeuLSHa8',
            ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
            config=Config(signature_version='oauth'),
            endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

        body = client_73f80164cc8d4453ace2ccaeedb9f5d6.get_object(Bucket='bioage-donotdelete-pr-re6stby4ah9ab4',Key='Healthy_Methylation_Dataset.csv')['Body']
        # add missing __iter__ method, so pandas accepts body as file-like object
        if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

        healthy_df = pd.read_csv(body)
        healthy_df.head()
```

```
Out[4]:
          ...1  cg09809672  cg22736354  cg02228185  cg01820374  cg06493994  cg19761273  sex   age
0   GSM507152     0.392464    0.281630    0.352591    0.315924    0.111604    0.211683    1  71.06
1   GSM507153     0.377919    0.238900    0.520396    0.323641    0.127965    0.184307    1  69.65
2   GSM507154     0.458522    0.204078    0.519273    0.363714    0.119891    0.215067    1  65.79
```



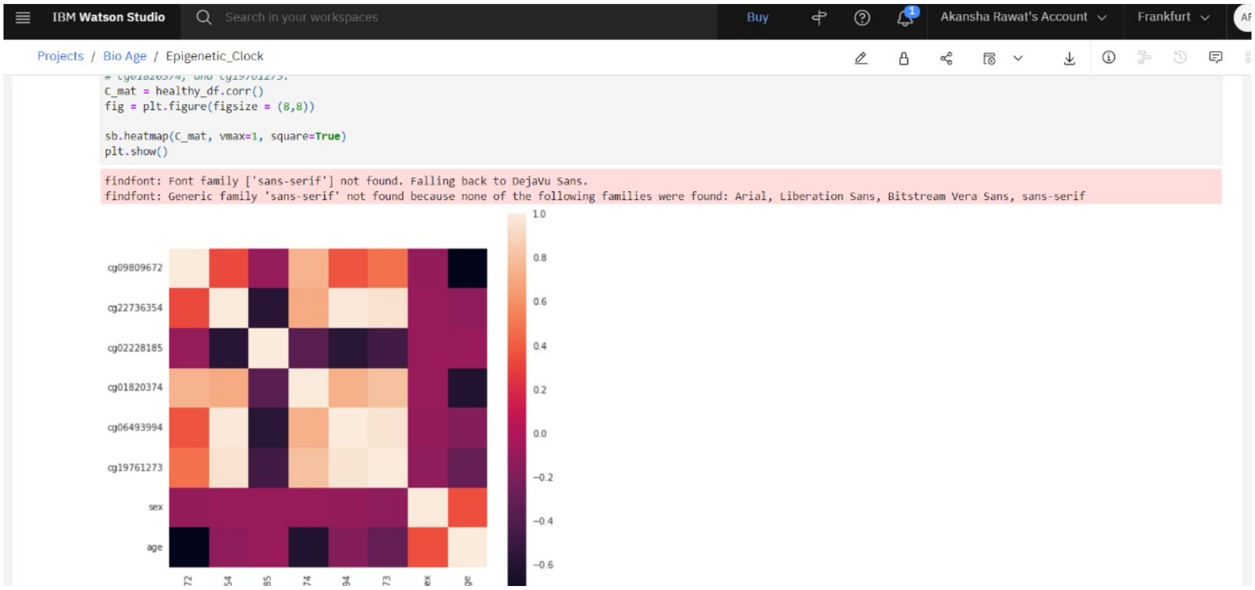
Projects / Bio Age / Epigenetic_Clock

```
In [5]: # Shuffle dataframe to randomize data order, possibly preventing confounding factors
        healthy_df = shuffle(healthy_df)

        # Remove patient ID column
        healthy_df = healthy_df.drop(['...1'], axis=1)

        # Drop all rows with NaN values
        healthy_df = healthy_df.dropna()

        # Reset Index
        healthy_df.reset_index(inplace=True, drop=True)

        healthy_df.head()
```

Out[5]:

	cg09809672	cg22736354	cg02228185	cg01820374	cg06493994	cg19761273	sex	age
0	0.861134	0.030339	0.772940	0.764273	0.038618	0.407311	1	0.00
1	0.665470	0.111330	0.644090	0.528570	0.050360	0.325280	0	4.25
2	0.613737	0.231627	0.798076	0.392510	0.179445	0.307907	0	32.00
3	0.847000	0.076200	0.781000	0.762000	0.030100	0.416000	0	0.00
4	0.875370	0.127060	0.704220	0.589560	0.014410	0.668570	1	0.00

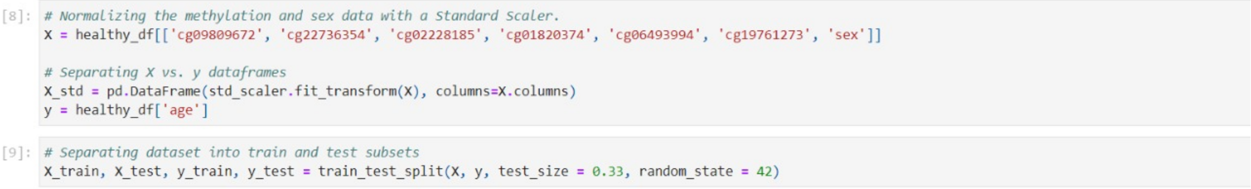
```
In [6]: # Checking if there are any remaining NaNs in the dataset
        np.where(pd.isnull(healthy_df))

Out[6]: (array([], dtype=int64), array([], dtype=int64))
```



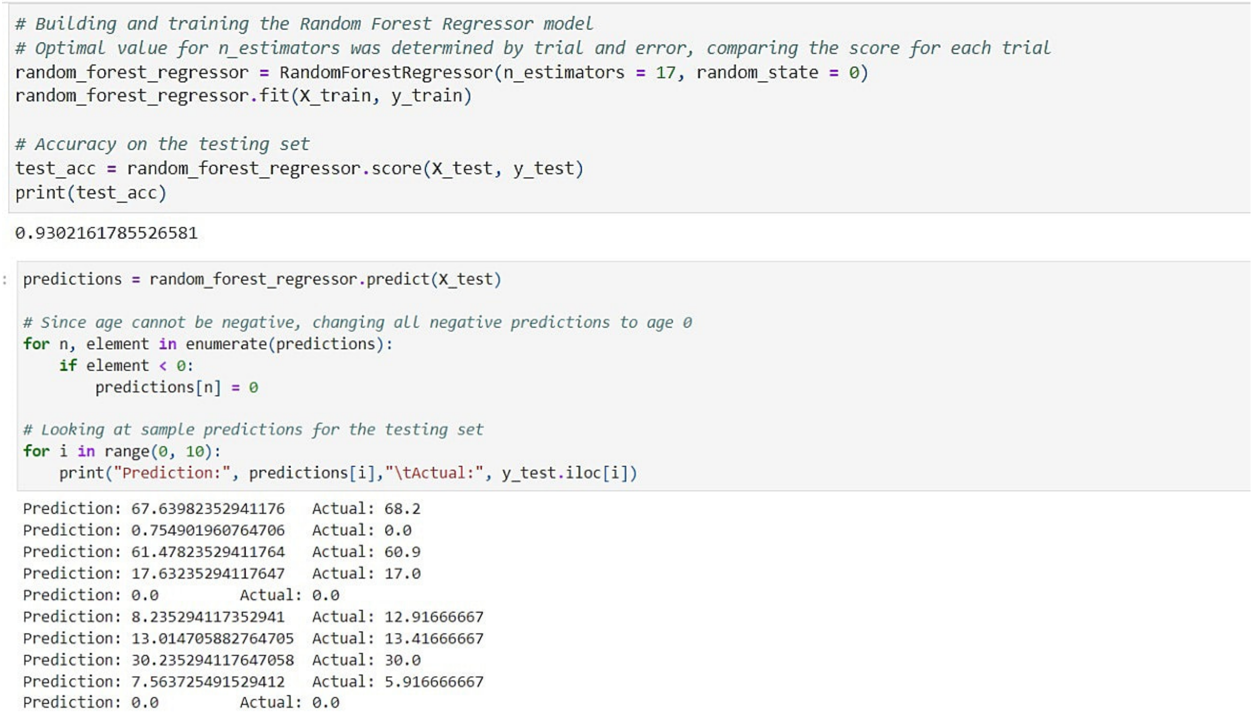
Projects / Bio Age / Epigenetic_Clock

```
# cg01820374, and cg19761273.
C_mat = healthy_df.corr()
fig = plt.figure(figsize = (8,8))

sb.heatmap(C_mat, vmax=1, square=True)
plt.show()
```

```
findfont: Font family ['sans-serif'] not found. Falling back to DejaVu Sans.
findfont: Generic family 'sans-serif' not found because none of the following families were found: Arial, Liberation Sans, Bitstream Vera Sans, sans-serif
```



```
[8]: # Normalizing the methylation and sex data with a Standard Scaler.
     X = healthy_df[['cg09809672', 'cg22736354', 'cg02228185', 'cg01820374', 'cg06493994', 'cg19761273', 'sex']]

     # Separating X vs. y dataframes
     X_std = pd.DataFrame(std_scaler.fit_transform(X), columns=X.columns)
     y = healthy_df['age']

[9]: # Separating dataset into train and test subsets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)
```

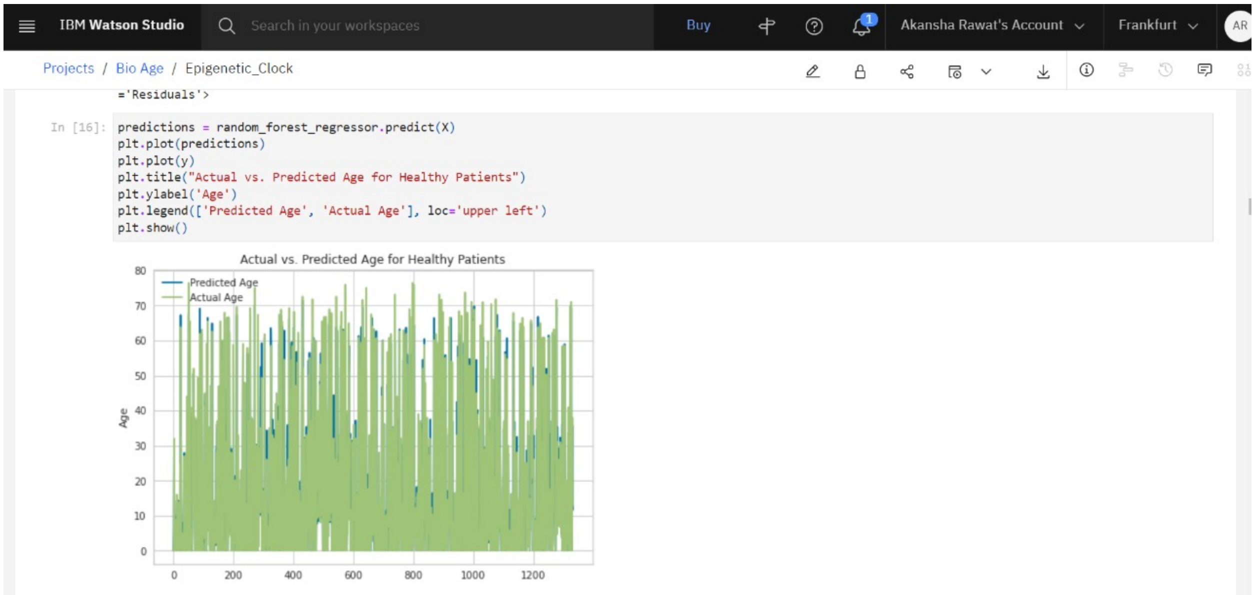


```python
# Building and training the Random Forest Regressor model
# Optimal value for n_estimators was determined by trial and error, comparing the score for each trial
random_forest_regressor = RandomForestRegressor(n_estimators = 17, random_state = 0)
random_forest_regressor.fit(X_train, y_train)

# Accuracy on the testing set
test_acc = random_forest_regressor.score(X_test, y_test)
print(test_acc)
```

```
0.9302161785526581
```

```python
predictions = random_forest_regressor.predict(X_test)

# Since age cannot be negative, changing all negative predictions to age 0
for n, element in enumerate(predictions):
    if element < 0:
        predictions[n] = 0

# Looking at sample predictions for the testing set
for i in range(0, 10):
    print("Prediction:", predictions[i],"\tActual:", y_test.iloc[i])
```

```
Prediction: 67.63982352941176   Actual: 68.2
Prediction: 0.754901960764706   Actual: 0.0
Prediction: 61.47823529411764   Actual: 60.9
Prediction: 17.63235294117647   Actual: 17.0
Prediction: 0.0         Actual: 0.0
Prediction: 8.235294117352941   Actual: 12.91666667
Prediction: 13.014705882764705  Actual: 13.41666667
Prediction: 30.235294117647058  Actual: 30.0
Prediction: 7.563725491529412   Actual: 5.916666667
Prediction: 0.0         Actual: 0.0
```


IBM Watson Studio
Search in your workspaces
Buy
Akansha Rawat's Account
Frankfurt
AR
Projects / Bio Age / Epigenetic_Clock

```
='Residuals'>
```

```python
In [16]: predictions = random_forest_regressor.predict(X)
plt.plot(predictions)
plt.plot(y)
plt.title("Actual vs. Predicted Age for Healthy Patients")
plt.ylabel('Age')
plt.legend(['Predicted Age', 'Actual Age'], loc='upper left')
plt.show()
```