Advanced Machine Learning Project 1

Sonali Andani
Oct. 20th 2025
D-INFK, ETH Zürich

Task 1: Predict a person's age from brain image data

Table of Contents

Table of Contents

Magnetic Resonance Imaging (MRI)

- It uses a magnetic field and radio waves to produce 3D detailed anatomical images
- Non-invasive image technology and non-radiative investigation of sensitive organs.
- It produces high-resolution images.
- MRI machines are large, tube-shaped magnets

MRI

- Magnetic Resonance Imaging (MRI) is a key technology in medical imaging

- Non-invasive + Non-radiative investigation of sensitive organs (brain)

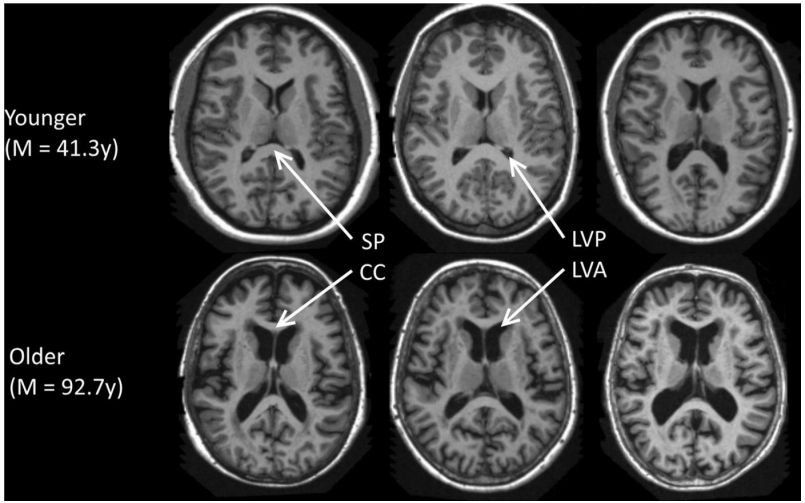
- Switzerland: 45.0 MRI units per 1.000.000 inhabitants (2016)
- West-Africa: 0.22 MRI units per 1.000.000 inhabitants (2018)

https://data.oecd.org/healtheqt/magnetic-resonance-imaging-mri-units.htm

Table of Contents

Aging Effect on Brain MRI

- The brain undergoes profound age-related neuroanatomical changes during the aging process.
- The global grey matter volume decreased with age.
- Grey matter contains most of the brain’s neuronal cell bodies.



Davis, Nick J. "Brain stimulation for cognitive enhancement in the older person: State of the art and future directions." *Journal of Cognitive Enhancement* 1.3 (2017): 337-344.

Brain Age Estimation: Why it is interesting?

- Aging does not affect people uniformly!
- Individual rates of aging are shaped by interactions between environmental, genetic, and epigenetic factors.
- Studies base on brain MRI show that there is a relation between accelerated aging and accelerated brain atrophy.
- It could improve early diagnosis and risk-assessments for age-associated neurodegenerative and neuropsychiatric diseases at a subject level:
    - Alzheimer, Parkinson, Huntington, etc.

Table of Contents

MRI Processing

Raw brain scans are difficult to handle

- 3D brain scans are ~ 200x200x200 ~10^7 voxels (3D voxel ~ $1mm^3$)
- 3D structure + individual brain shapes → difficult to recognize disease patterns
- Data is scarce:

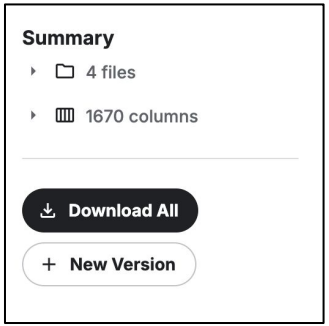
	ImageNet	MRI data set for Task 1
Image size	224 x 224 x 3	~200 x 200 x 200
Data set size	1.2 million	~2k

MRI Feature Extraction

We are using 832 anatomical features for this project

- Informative features derived from image data (with FreeSurfer)
- No need to process big images (6 GB) → csv sheet (3 MB)
- No need for image analysis
- Meaningful features are extracted using domain knowledge
  - e.g. cortex volume, left/right hemisphere surface area, white/gray matter volume etc.
- Information loss

How to download data



- Directly from kaggle.com
  - Go to competition link → Data → scroll and on right →

OR

- pip install kaggle
- kaggle.com → account → settings → API → create new API token → download as json
- Move json file

  mkdir -p ~/.kaggle

  mv ~/Downloads/kaggle.json ~/.kaggle/

  chmod 600 ~/.kaggle/kaggle.json
- In terminal: *kaggle competitions download -c eth-aml-2025-project-1*

File Description

We provide the following files:

- **X_train.csv**, **y_train.csv**: the training set, including the features and labels
- **X_test.csv**: the testing set (make predictions based on this file)
- **sample.csv**: a sample submission file in the correct format

Table of Contents

AML Task 1: Age Prediction

We have modified the derived input data in three ways:

- Irrelevant features
- Outliers
- Perturbations (e.g. missing values, etc)

	A	B	C	D	E	F	G	H	I	J
1	id	x0	x1	x2	x3	x4	x5	x6	x7	x8
2	0	14168.82	10514.38	3316.15	94230.7	102.3866	92.67713	11108.75	10866.51	10837.62
3	1	17757.04		4101.016	92959.53		99.85517	10013.96	10826.61	10076.1
4	2	14226.66	11029.64		124055.6	100.5425	92.86089		10492.34	
5	3	8766.012	7384.203	2147.308	100157.7	104.8551	101.929	10050.05	10499.52	10525.03
6	4	13801.02	13269.49	3408.317	92048.53	103.7598	95.78923	9667.354	10750.78	10618.8
7	5	11333.67	9693.98	2930.261	98892.45	103.9677	96.53505	10143.72	10889.1	10410.71
8	6	18012.99	13437.22	3928.789	105320.7	109.3164	102.4219	10218.4	10790.86	10252.3
9	7	17471.68	13219.53	3438.191		104.2403	99.32157	10678.33	10137.17	10860.43
10	8		11678.01	2782.414	93217.44	106.8652	106.655	10967.66	10236.58	10217.15
11	9	15821.05		4027.451	98841.36	103.7156	100.2716	11746.87	10549.48	10325.19
12	10	6480.594	6865.479	2074.261	99595.09	104.9383	101.421	10207.11	10520.72	10520.81
13	11	13190.9	9402.399	3224.518	84625.96	102.1383		8627.341	10362.81	10679.29
14	12	17182.58		3769.271	94769.97	108.786	113.3053	9836.233	10385.08	10041.71

Subtask 0: Filling Missing Values

Background

There are missing values in the data

- Originally they are set to NaN
- Most methods cannot handle NaNs automatically
- Different possible strategies to impute missing values: mean, median, most frequent etc.

Task requirements

We require that students

fill missing values in the training and the test set

Subtask 1: Outlier Detection

Background

In the training set, there exists outliers

- If the resulting model is not robust enough, it may be sensitive to the outliers
- Solution: outlier removal

Task requirements

We require that students

build an outlier detection model to make classification for samples in the training set i.e. whether they are outliers.

Subtask 2: Feature Selection

Background

To make the task a bit more challenging, we added some manual features to the FreeSurfer-processed dataset.

- Feature selection is needed

Advantages:

- Simplifies the models to make them interpretable
- Leads to shorter training times
- Better generatlization by reducing overfitting

Task requirements

We require that students

use feature selection methods to label the features as selected features and unselected features.

Here, unselected features includes irrelevant features and redundant features.

Main Task: Age Prediction

Background

After primary preprocessing and dimensionality reduction, now we finally arrive at the regression task.

Task requirements

We require that students

use suitable regression methods to predict the age of a person from brain data.

Table of Contents

Evaluation Metrics

Coefficient of Determination R^2

is the proportion of the variance in the dependent variable that is predictable from the independent variable.

$$R^2 := 1 - \frac{SS_{res}}{SS_{tot}}$$
$$SS_{tot} := \sum_i (y_i - \overline{y})^2$$
$$SS_{res} := \sum_i (f_i - y_i)^2$$

- Varies between 1 (best) and -∞;
  $SS_{tot} = SS_{res}$ then $R^2 = 0$
- R^2 is a scaled version of MSE
  e.g. R^2 is invariant to scaling y, unlike MSE

How to compute it in Python:

```
from sklearn.metrics import r2_score

score = r2_score(y_true, y_pred)
```

Submission to Kaggle

Kaggle link: https://www.kaggle.com/t/34eeeead34fa46d9b473eb82ae4c303f

- Team names must be alphanumeric (A-Z, a-z, 0-9).
- Must fill the **Project 1 form** in Moodle for the project.
  - Name of team in Kaggle, legis, and the description of how you solved the task.
- Ensure that your results are reproducible.
- Project period: Oct 20th 3pm – Nov 10th 2pm
- Including public/private leaderboard:
  - Public leaderboard is available
  - Private leaderboard will open from Nov 10th 3pm
- Public baseline for passing the project: 0.5
- Ensure that your results are reproducible. We may ask for your code after the deadline and ask you to reproduce your results.

Other Considerations

- Do not use AutoML packages
  - This includes anything that does automatic data clearing and automatic model selection
- Be aware of overfitting on the public test set
- Be careful about the submission time
  - Your group has a joint total **10** submissions per day.
- Describe what you did when you hand in the project
  - Keep your implementation for potential review
- Do not wait until the last day to submit something
  - Servers usually get overloaded and crash causing long waiting times

Other Considerations

To obtain points for this task, you have to individually hand in the task as follows:

- You need to select one of your group's submissions for grading. You will only be graded on this submission.
- You have to write a short description of the approach that you have used. Each student has to individually write their own description and you are not allowed to share the description with your other group members.

<u>If you do not properly hand in the task, you will receive zero points for the task.</u>

Frequently Asked Questions

Q: Which programming language and tools am I supposed to use?

A: You are free to choose any programming language and any software library.

Q: Can you give me a deadline extension?

A: We can not grant deadline extensions, except in extraordinary cases (e.g. military service). However, we will require official confirmation of your problem (e.g. certificate of illness).

Frequently Asked Questions

Q: Can I post on Moodle as soon as I have a question?

A: This is highly discouraged. Instead,

- Read the details of the task thoroughly.
- Review the frequently asked questions.
- If there is another team that solved the task, try again.
- Discuss with your teammates.

If you still consider that you should contact the TAs, you can post a private question on Moodle. Remember that collaboration with other teams beyond (general discussions) is prohibited.

Frequently Asked Questions

Q: Am I allowed to use ideas from published papers and use their code?

A: You are allowed to search for any material that presents how to train models for related tasks (e.g. articles in conferences, repositories, etc...). However, you must re-implement the code by yourself. You are NOT allowed to copy code.

Q: When will we receive private scores? And the project grades?

A: We will publish the private scores before the exam the latest.

Table of Contents

Basic Examples: Data loading

```
import numpy as np
import pandas as pd
```

[1] ✓ 0.4s

```
X_train_df = pd.read_csv('./data/X_train.csv', skiprows=1, header=None)
y_train_df = pd.read_csv('./data/y_train.csv', skiprows=1, header=None)
X_test_df = pd.read_csv('./data/X_test.csv', skiprows=1, header=None)

X_train = X_train_df.values[:, 1:]
y_train = y_train_df.values[:, 1:]
X_test = X_test_df.values[:, 1:]

print(X_train.shape, y_train.shape, X_test.shape)
```

[2] ✓ 0.3s

```
(1212, 832) (1212, 1) (776, 832)
```

Basic Examples: Splitting and Imputing

```
from sklearn.model_selection import train_test_split

# Randomly split the data into training and validation sets with 80-20 ratio
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
```

[3] ✓ 0.5s

```
# Impute missing values with mean of each column
X_mean = np.nanmean(X_train, axis=0, keepdims=True)
X_train = np.where(np.isnan(X_train), X_mean, X_train)
X_val = np.where(np.isnan(X_val), X_mean, X_val)
X_test = np.where(np.isnan(X_test), X_mean, X_test)
```

[4] ✓ 0.0s

Basic Examples: Feature Selection

```
from sklearn.feature_selection import SelectKBest, mutual_info_regression

# Select top 100 features with highest mutual information
selection = SelectKBest(mutual_info_regression, k=100).fit(X_train, y_train)
X_train = selection.transform(X_train)
X_val = selection.transform(X_val)
X_test = selection.transform(X_test)
```

[5] ✓ 3.3s

Basic Examples: Training and Validating

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

# Train a linear regression model
regressor = LinearRegression()
regressor.fit(X_train, y_train)

y_train_pred = regressor.predict(X_train)
y_val_pred = regressor.predict(X_val)

# Evaluate the model on training and validation sets
train_score = r2_score(y_train, y_train_pred)
val_score = r2_score(y_val, y_val_pred)

print(train_score, val_score)
```

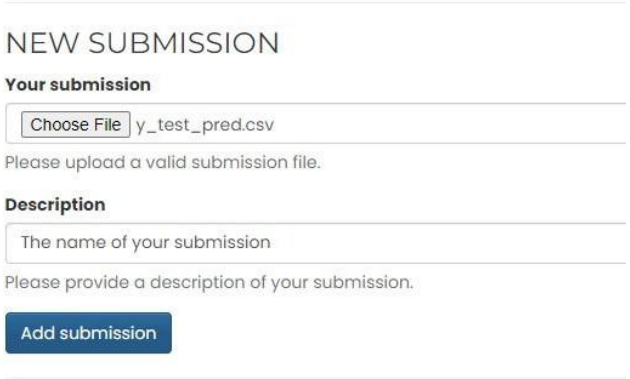
[6] ✓ 0.0s

```
0.46798100065626036 0.32067747541726366
```

Basic Examples: Export and Submit

```
# Predict on test set
y_test_pred = regressor.predict(X_test)
# Save predictions to submission file with the given format
table = pd.DataFrame({'id': np.arange(0, y_test_pred.shape[0]), 'y': y_test_pred.flatten()})
table.to_csv('./data/y_test_pred.csv', index=False)
```

[7] ✓ 0.0s



Q&A