BodyFat Project

Zhengyong Chen, Xiangsen Dong, Xupeng Tang, Zhaoqing Wu

Overview

- Data Cleaning
- Model Selection
- Trade-off
- Final Model

Data Cleaning

Data Cleaning

- We converted the unit of Height to **cm**.
- We imputed **individual 42's Height** due to an unusually low value.

Individual	*Original Value*	*Imputed Value*	*Imputation Method*
42	74.93 cm	181.47 cm	Using ADIPOSITY and WEIGHT

- We removed three individuals (**IDNO: 39, 41, 216**) due to outliers detected across multiple variables using the IQR method.

Individual	*Outlier_Variables*
39	WEIGHT, ADIPOSITY, NECK, CHEST, ABDOMEN, HIP, THIGH, KNEE, ANKLE, BICEPS, WRIST
41	WEIGHT, ADIPOSITY, CHEST, ABDOMEN, HIP, THIGH, WRIST
216	BODYFAT, DENSITY, ADIPOSITY, ABDOMEN

Data Cleaning

- We deleted column DENSITY and IDNO.
- We scaled the data before modeling.

Final Data: 249 rows with 14 predictors

Predictors: AGE, WEIGHT, HEIGHT, ADIPOSITY, NECK, CHEST, ABDOMEN, HIP, THIGH, KNEE, ANKLE, BICEPS, FOREARM, WRIST

Model Selection

Model Construction Principles

Select variables:
Stepwise regression

Reason:
Stepwise regression iteratively adds or removes predictors, effectively identifying the most valuable variables for predicting body fat and excluding those with lesser contributions.

Consideration of Candidate Models

Single Variable Model → Two Variable Model → Interaction Term Model

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \epsilon_i$$

$$Y_i = \beta_0 + \beta_1 X_{1i} X_{2i} + \epsilon_i$$

The variable "Abdomen" is present in all the best models:

$$Y_i = \beta_0 + \beta_1 \mathrm{Abdomen}_i^2 + \beta_2 X_i + \epsilon_i$$

where X_i is one of other factors

Next step: We will select among these models based on performance metrics such as R^2 and MSE.

Trade-off

Trade-Offs

Accuracy: we measured accuracy using the following criteria

RMSE
- Pros: RMSE measures the average prediction error, indicating model accuracy
- Cons: Sensitive to outliers. 😱 But we have removed them 🤗

adjust_R2
- Pros: Accounts for model complexity, prevents overfitting, useful for multiple predictors.
- Cons: Does not provide a direct measure of prediction error. 😱 But we could combine 2 metric 🤗

Simplicity: Measure by linear models with different number of predictors:

BODYFAT ~ AGE + WEIGHT + NECK + ABDOMEN + THIGH + FOREARM + WRIST

BODYFAT ~ ABDOMEN 🤗

BODYFAT ~ WEIGHT + ABDOMEN

BODYFAT ~ ABDOMEN * WEIGHT 🤗

BODYFAT ~ ABDOMEN_squared

BODYFAT ~ ABDOMEN_squared + WEIGHT

BODYFAT ~ ABDOMEN + ABDOMEN_squared + WEIGHT

Trade-Offs

Robustness-method select:

- **Definition**: The definition of robustness is how model could hold its performance on noisy data.

Hence by definition, we evaluate robustness by add noise on training/test dataset. **But the question is, where should we add noise?** 🤔

- **Method 1**: Train 1 model on training set, then test on both the original test set and a noisy test set with noise added to both $(X_{test}\ Y_{test})$.
  - Model trained only on clean data; weak against noisy test data. 😟
  - Does not consider training noise, limiting comprehensive robustness evaluation. 😟

- **Method 2**: Train 1 model on training set, then test on both the original test set and a noisy test set with noise added to only X_{test}
  - Ignores target noise, failing to assess output uncertainty. 😟
  - Trained only on original data, lacks robustness against noisy training data. 😟

- **Method 3**: Train two models: one on the original dataset, and another with noise added to only X_{train}, then test both on the same test set.
  - limiting the model's adaptability to target noise, making output noise robustness unclear. 😟

- **Method 4**: Train two models: one on the original dataset, and another with noise added to only Y_{train}, then test both on the same test set.
  - leaving input noise effects untested. 😟

Trade-Offs

Robustness – Final Proposal method:

- **Final method**: Train two models: one on the original dataset, and another with noise added to both $(Y_{train}\ X_{train})$, then test both on the same test set.
  - Noise added to both input features XX and targets YY in training enhances adaptability to real-world uncertainties.
  - Training on noisy datasets improves the model's performance under noisy conditions.

- **Process:**

For each fold $k = 1, 2, \ldots, K$:

1. Split the dataset into training set $D_{\text{train},k}$ and testing set $D_{\text{test},k}$.
2. For each model $m \in \text{models}$:
   - Train the baseline model m on $D_{\text{train},k}$.
   - Add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to $D_{\text{train},k}$ and train the noise-perturbed model m_{noise} on this noisy dataset.
3. Evaluate both m and m_{noise} on $D_{\text{test},k}$ using the metric:

$$\text{MSE}(m, D_{\text{test},k}), \quad \text{MSE}(m_{\text{noise}}, D_{\text{test},k})$$

After completing the cross-validation, compare the robustness of the models using the **retention rate** r, defined as:

1, Accuracy Retention in Noisy Data

$$RetentionRate = 1 - \frac{|metric_{noisy} - metric_{original}|}{metric_{original}}$$

metric = MSE

Why Use Cross-Validation in Robustness Testing?

- Training and evaluating on a single train-test split can lead to **overfitting** or **overestimating the model's performance**.
- If evaluated on a single data split, the model's performance may be dominated by the characteristics of the particular split, such as anomalies or certain data points.

Results

model	adjusted_R	MSE	retention_MSE_rate	num_predictors
BODYFAT ~ AGE + WEIGHT + NECK + ABDOMEN + THIGH + FOREARM + WRIST	0.727	0.272	0.978	7
BODYFAT ~ ABDOMEN	0.666	0.33	0.993	1
BODYFAT ~ WEIGHT + ABDOMEN	0.708	0.292	0.987	2
BODYFAT ~ ABDOMEN * WEIGHT	0.023	0.978	0.982	1
BODYFAT ~ ABDOMEN_squared	0.013	0.985	0.999	1
BODYFAT ~ ABDOMEN_squared + WEIGHT	0.539	0.461	0.997	2
BODYFAT ~ ABDOMEN + ABDOMEN_squared + WEIGHT	0.709	0.286	0.981	3

Combining all of above results, our final model is BODYFAT ~ WEIGHT + ABDOMEN, since it has relatively large R2 value and small MSE, few predictors and high retention rate

Final Model

Bodyfat_scaled = 1.18*Abdomen_scaled - 0.42Weight_scaled.

sample mean of Bodyfat = 18.94%, sample sd of Bodyfat = 7.75%,

sample mean of Abdomen = 92.08 cm, sample sd of Abdomen = 9.85 cm,

sample mean of Weight = 177.69 lbs, sample sd of weight = 26.48 lbs.

Bodyfat (%)= -44.71 (%)+ 0.93*Abdomen (cm) - 0.12*Weight (lbs)

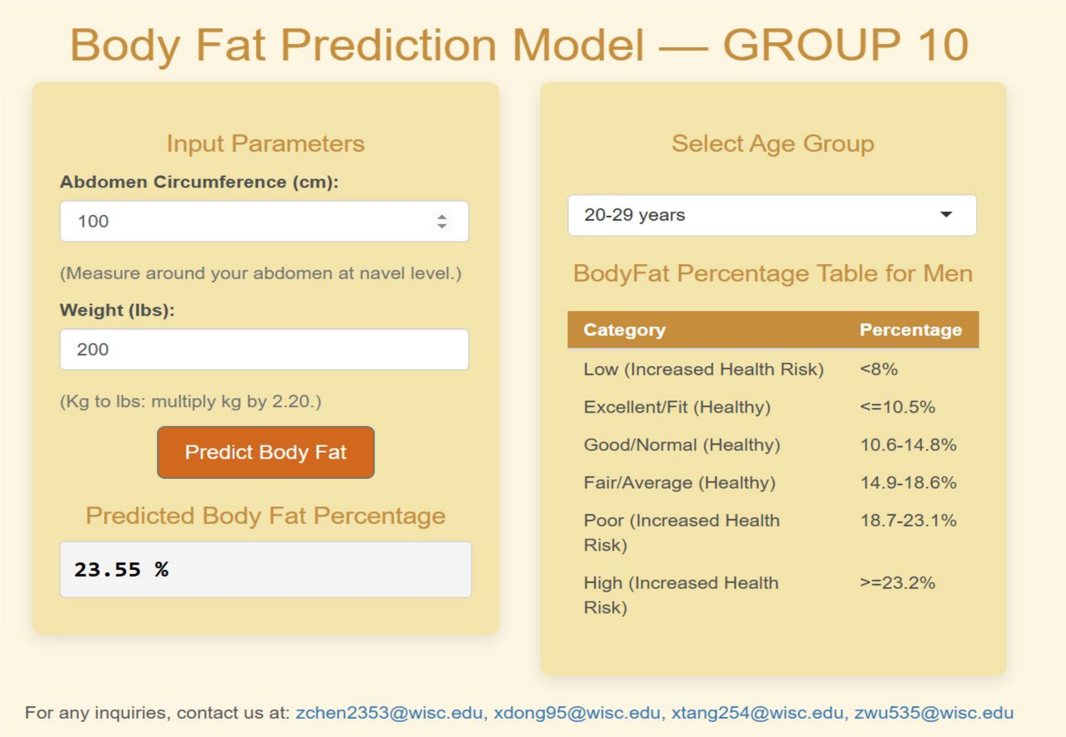
Example

A man of 200 lbs weight and 100 cm abdomen circumference, his body fat is predicted to be:

$$(1.18 \times \frac{100 - 92.08}{9.85} - 0.42 \times \frac{200 - 177.69}{26.48}) \times 7.75\% + 18.94\% = 23.55\%.$$

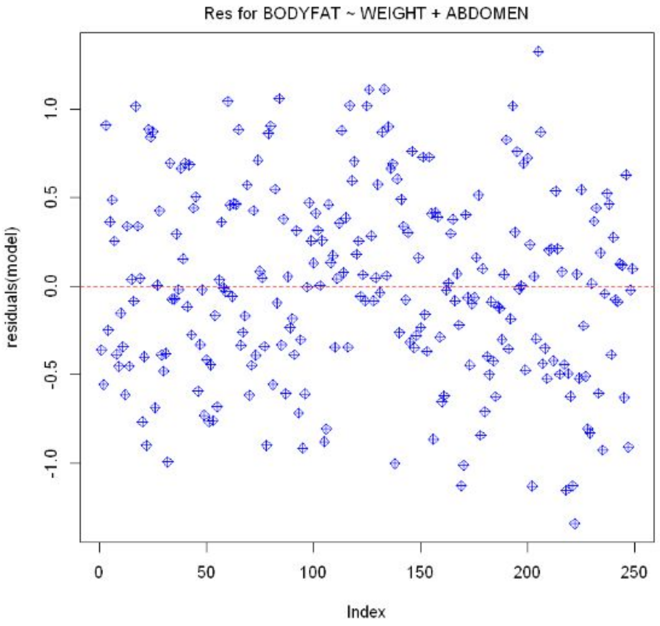
Example

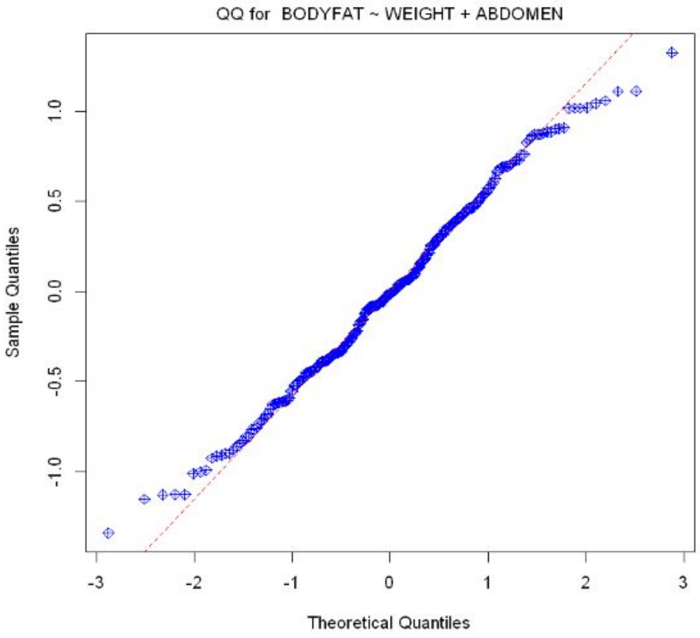
We have developed this model into a Shiny app:
https://andrewchanshiny.shinyapps.io/Bodyfat-Group10-P2-628/



P-value of predictors is greatly less than 0.01, which indicates statistical significance.

Analysis on residuals





Strength:

Easy to use.

A certain level of accuracy.

Weakness

Since we only use simplest model and combination of predictors, the result is far from optimal.

Thanks for listening