# Assignment 3: Logistic Regression

## Yu-Chen Su

1. **Download** the Bone Mass Density (BMD) patient dataset, BMD-2.csv

   Import pandas to read file and show data head (by Jupyter Notebook)

```
import pandas as pd
df = pd.read_csv('BMD-2.csv')
df.head()
```

| | Age | Weight_kg | Height_cm | BMD | Fracture |
|---|---|---|---|---|---|
| **0** | 57.052768 | 64.0 | 155.5 | 0.8793 | no fracture |
| **1** | 75.741225 | 78.0 | 162.0 | 0.7946 | no fracture |
| **2** | 70.778900 | 73.0 | 170.5 | 0.9067 | no fracture |
| **3** | 78.247175 | 60.0 | 148.0 | 0.7112 | no fracture |
| **4** | 54.191877 | 55.0 | 161.0 | 0.7909 | no fracture |

2. Determine the **data dimensionality** by finding the following (5pts):

   A. Total number of patients.

```
# Display the count
num_rows = len(df)
print(f"Total number of customers: {num_rows}")
df.count()
```

```
Total number of customers: 169

Age          169
Weight_kg    169
Height_cm    169
BMD          169
Fracture     169
dtype: int64
```

   B. Number of attributes (categories).

```
# Display the column (categories)
num_column = df.shape[1]
print(f"Total number of categories: {num_column}")
```

```
Total number of categories: 5
```

   C. Data types.

```
# Check data types of each column
print(df.dtypes)
```

```
Age          float64
Weight_kg    float64
Height_cm    float64
BMD          float64
Fracture      object
dtype: object
```

D. Missing values.

```
missing_values = df.isnull().sum()
print(missing_values)
```

```
Age          0
Weight_kg    0
Height_cm    0
BMD          0
Fracture     0
dtype: int64
```

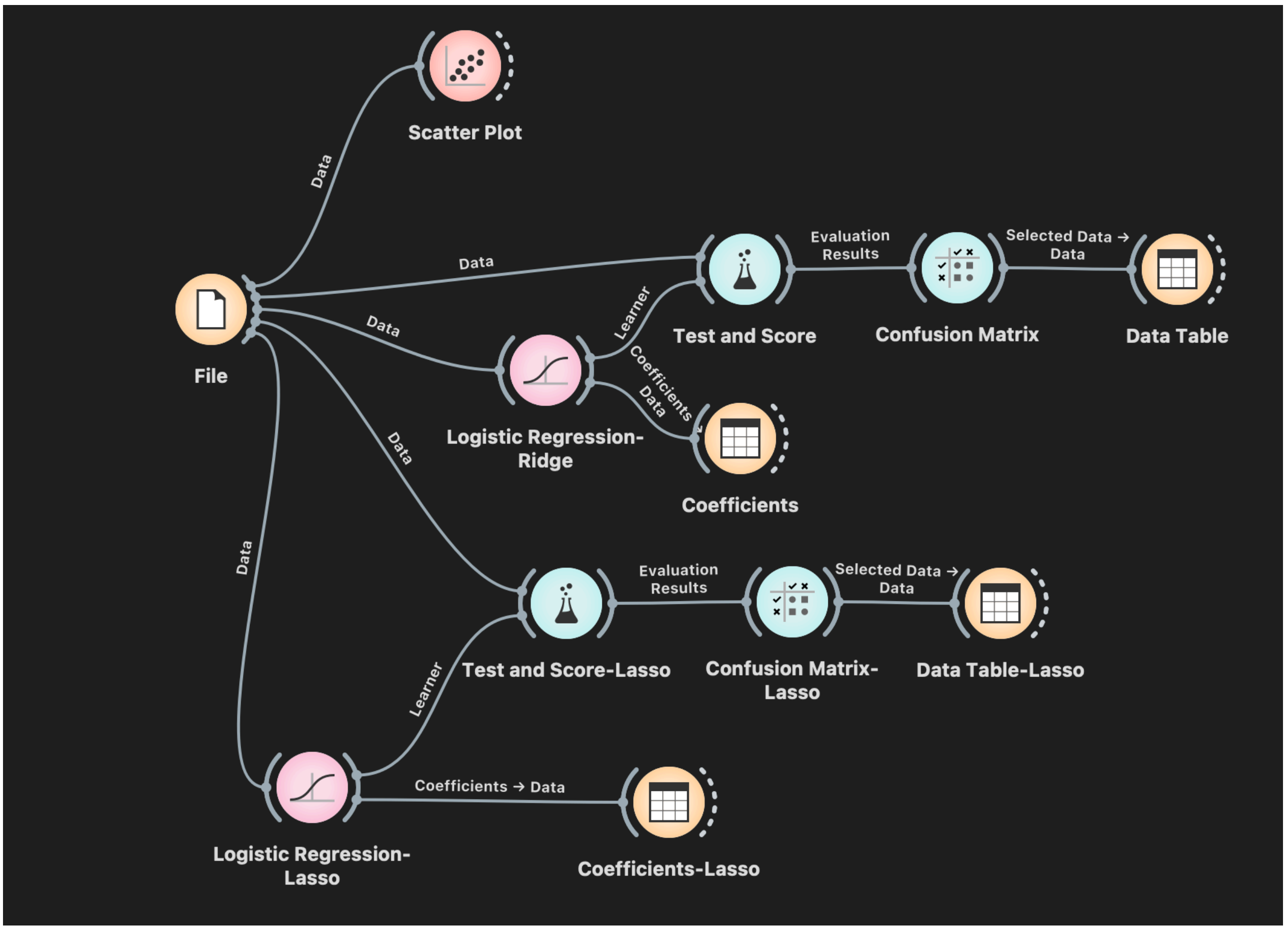
E. Number of patients in each target class.

```
# Group by 'Category' and count the occurrences
grouped_count = df.groupby('Fracture').size().reset_index(name='Count')

print(grouped_count)

```

```
     Fracture  Count
0    fracture     50
1 no fracture    119
```

3. Apply **Logistic Regression** using **Ridge Regulation** and explain the following (5pts):

Try to use orange in the following questions



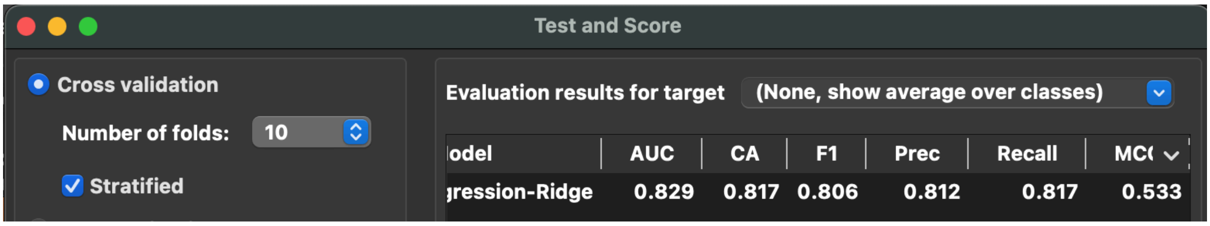
A. **Feature(s)** considered important based on the coefficient values.

| | name | no fracture |
|---|---|---|
| 1 | intercept | 5.39456 |
| 2 | Age | -0.0547052 |
| 3 | Weight_kg | 0.0637684 |
| 4 | Height_cm | -0.0457223 |
| 5 | BMD | 3.09401 |

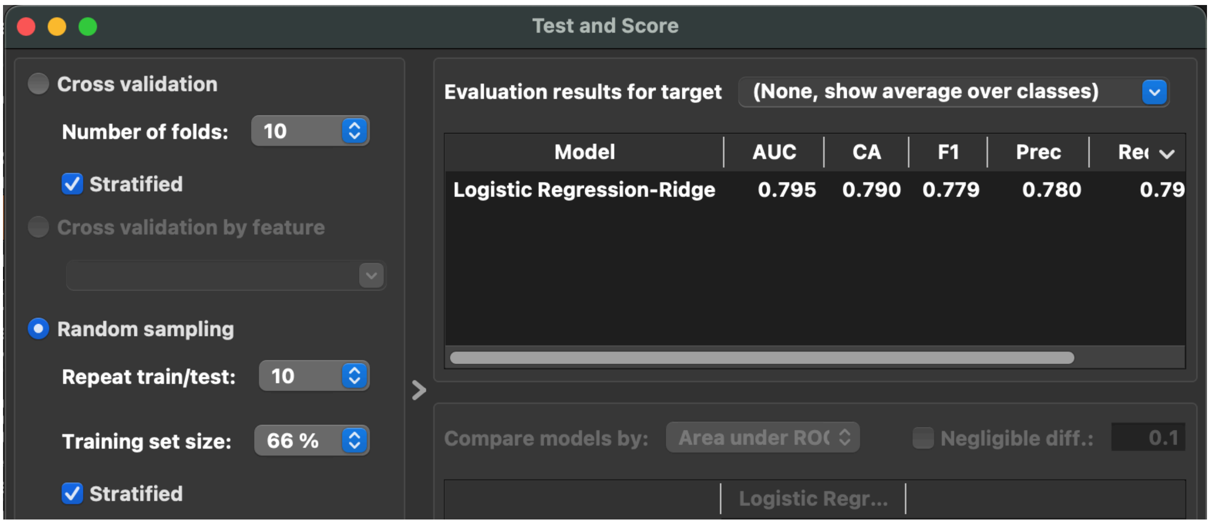
Based on the coefficients, BMD has a high value, indicating its importance in classifying the targets. In contrast, the other three features have lower coefficient values, meaning they have less impact.

B. **Classification accuracy.**

Achieved a classification accuracy (CA) of 0.817 through 10-fold cross-validation.




Achieved a classification accuracy (CA) of 0.790 using random sampling with 10 repeated train/test splits and a 66% training set size.



C. **Number of patients misclassified for each target class.**

23 fracture patients were misclassified as having no fracture, and 8 patients without fractures were misclassified as having a fracture, making a total of 31 misclassified patients.

Show: Number of instances

| | | Predicted | | |
|---|---|---|---|---|
| | | fracture | no fracture | Σ |
| Actual | fracture | 27 | 23 | 50 |
| | no fracture | 8 | 111 | 119 |
| | Σ | 35 | 134 | 169 |

4. Apply **Logistic Regression** using **Lasso Regulation** and explain the following (5pts):

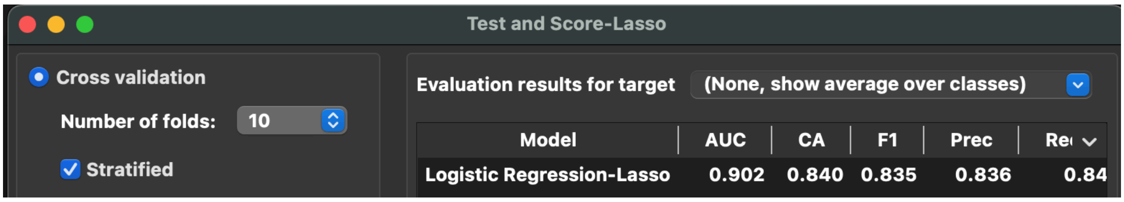
A. **Feature(s)** considered important based on the coefficient values.

| | name | no fracture |
|---|---|---|
| 1 | intercept | 0 |
| 2 | Age | -0.0300864 |
| 3 | Weight_kg | 0.028482 |
| 4 | Height_cm | -0.0397827 |
| 5 | BMD | 10.0272 |

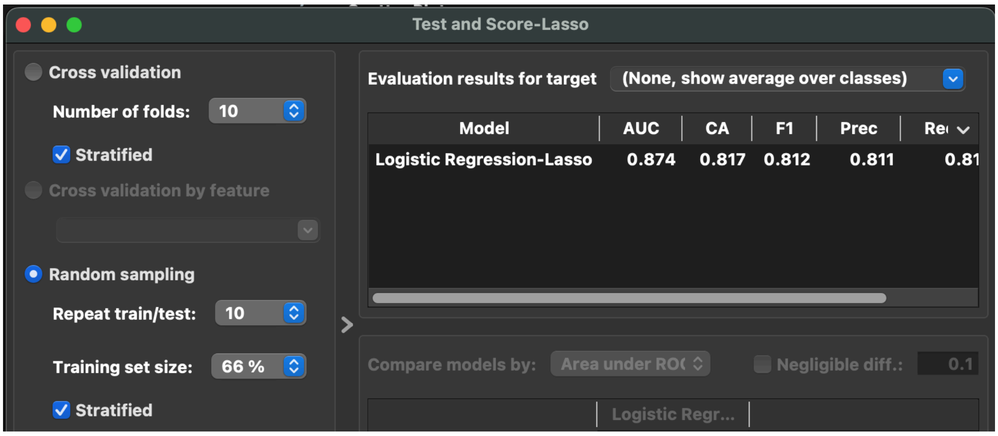
Based on the coefficients, BMD has a high value, indicating its importance in classifying the targets. In contrast, the other three features have lower coefficient values, meaning they have less impact.

B. **Classification accuracy.**

Achieved a classification accuracy (CA) of 0.840 through 10-fold cross-validation.

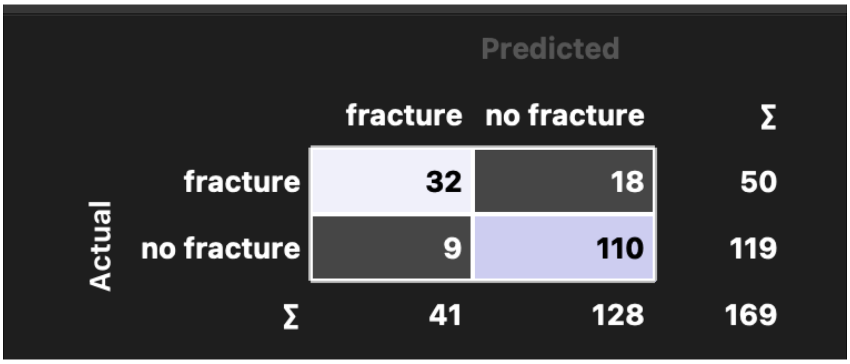



Achieved a classification accuracy (CA) of 0.817 using random sampling with 10 repeated train/test splits and a 66% training set size.




C. **Number of patients misclassified for each target class.**

18 fracture patients were misclassified as having no fracture, and 9 patients without fractures were misclassified as having a fracture, making a total of 27 misclassified patients.

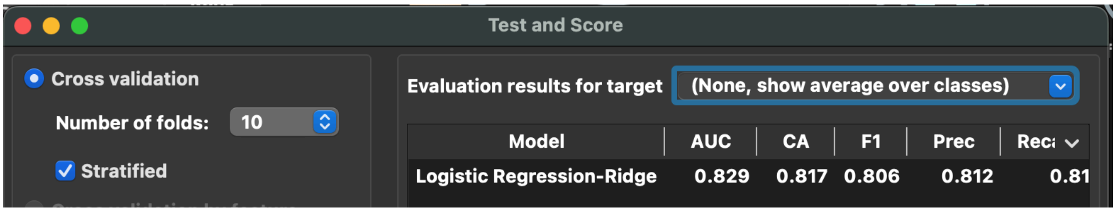


| Actual \ Predicted | fracture | no fracture | Σ |
|---|---|---|---|
| fracture | 32 | 18 | 50 |
| no fracture | 9 | 110 | 119 |
| Σ | 41 | 128 | 169 |

D. **Comparison** of classification accuracies among the regulation methods.

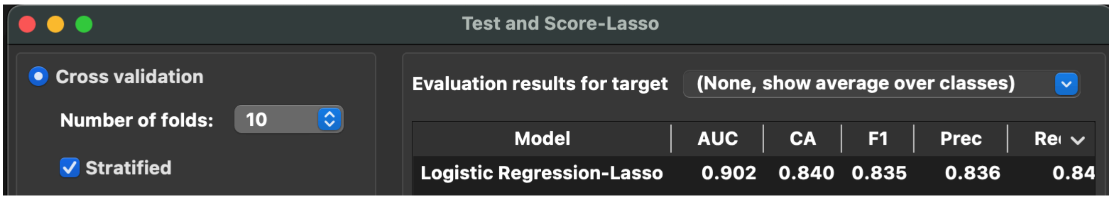
Using 10-fold stratified cross-validation, Lasso achieved a higher classification accuracy (0.840) compared to Ridge (0.817).

**Ridge**




| Model | AUC | CA | F1 | Prec | Reca |
|---|---|---|---|---|---|
| Logistic Regression-Ridge | 0.829 | 0.817 | 0.806 | 0.812 | 0.81 |

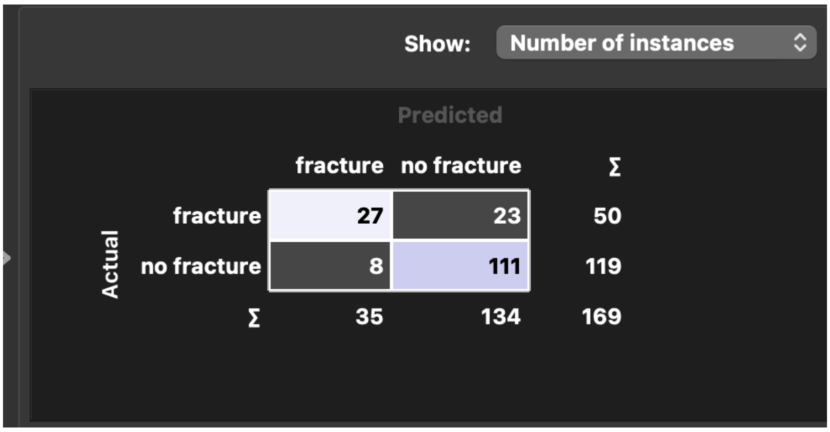
**Lasso**




| Model | AUC | CA | F1 | Prec | Re |
|---|---|---|---|---|---|
| Logistic Regression-Lasso | 0.902 | 0.840 | 0.835 | 0.836 | 0.84 |

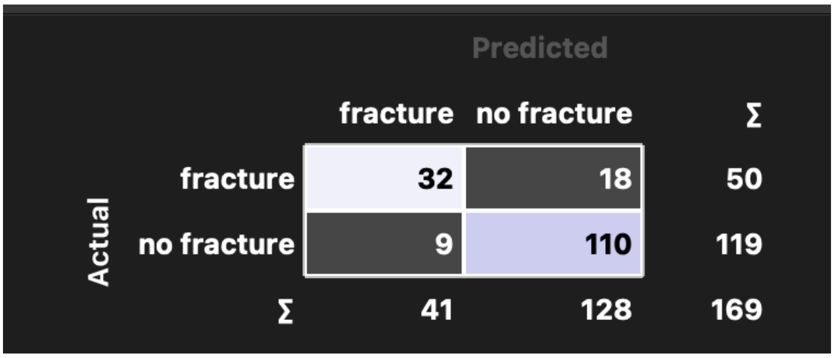
In the confusion matrix, out of 169 instances, Ridge misclassified 31 patients (8 + 23), while Lasso misclassified 27 patients (9 + 18). Therefore, Lasso has a higher classification accuracy than Ridge in this case.

**Ridge**




| | fracture | no fracture | Σ |
|---|---|---|---|
| fracture | 27 | 23 | 50 |
| no fracture | 8 | 111 | 119 |
| Σ | 35 | 134 | 169 |

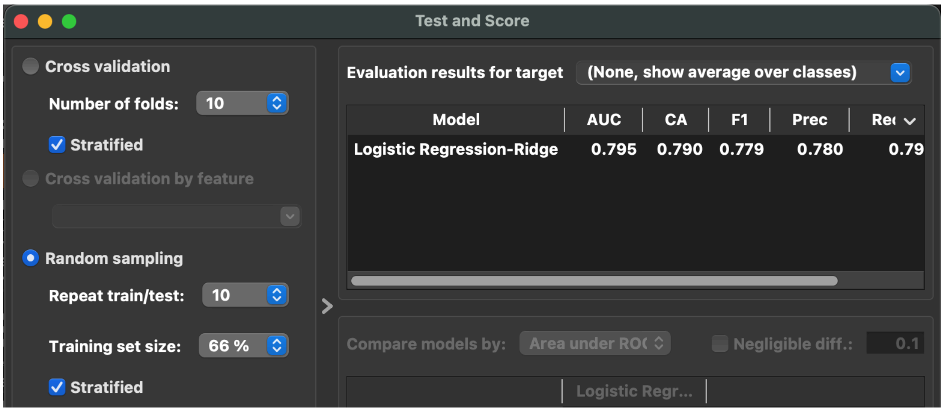
**Lasso**



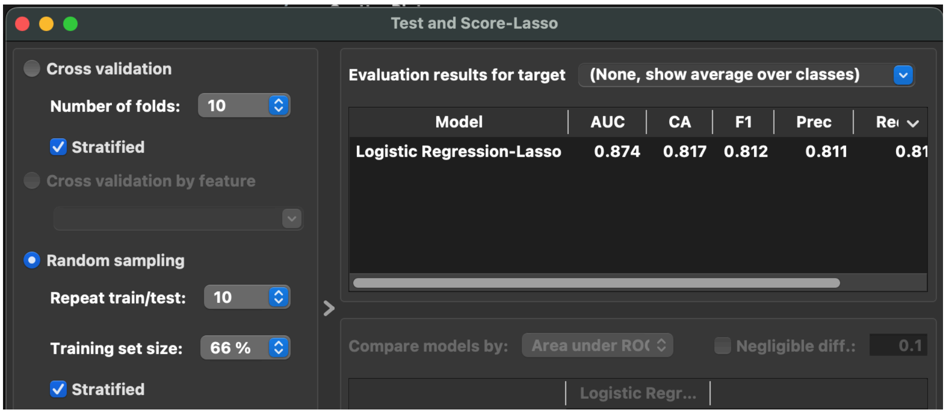

| | fracture | no fracture | Σ |
|---|---|---|---|
| fracture | 32 | 18 | 50 |
| no fracture | 9 | 110 | 119 |
| Σ | 41 | 128 | 169 |

I also used random sampling in the **Test & Score** widget with 10 repeated train/test splits and a 66% training set size. In this setup, Lasso achieved a higher classification accuracy (0.817) compared to Ridge (0.790).

**Ridge**

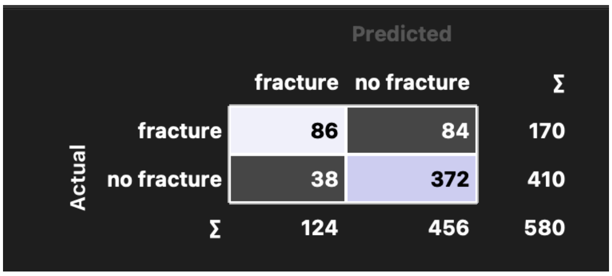



**Lasso**




I used random sampling with 10 repeated train/test splits and a 66% training set size. This means we get 580 test instances. (169 original instances x 34% test size x 10 repeats = 580 instances (169 x 34% = 57.46, system get 58)). In the confusion matrix, Ridge misclassified 122 patients (38 + 84) out of these 580 instances, while Lasso misclassified 116 patients (39 + 67). Therefore, Lasso has a higher classification accuracy than Ridge in this case.

**Ridge**

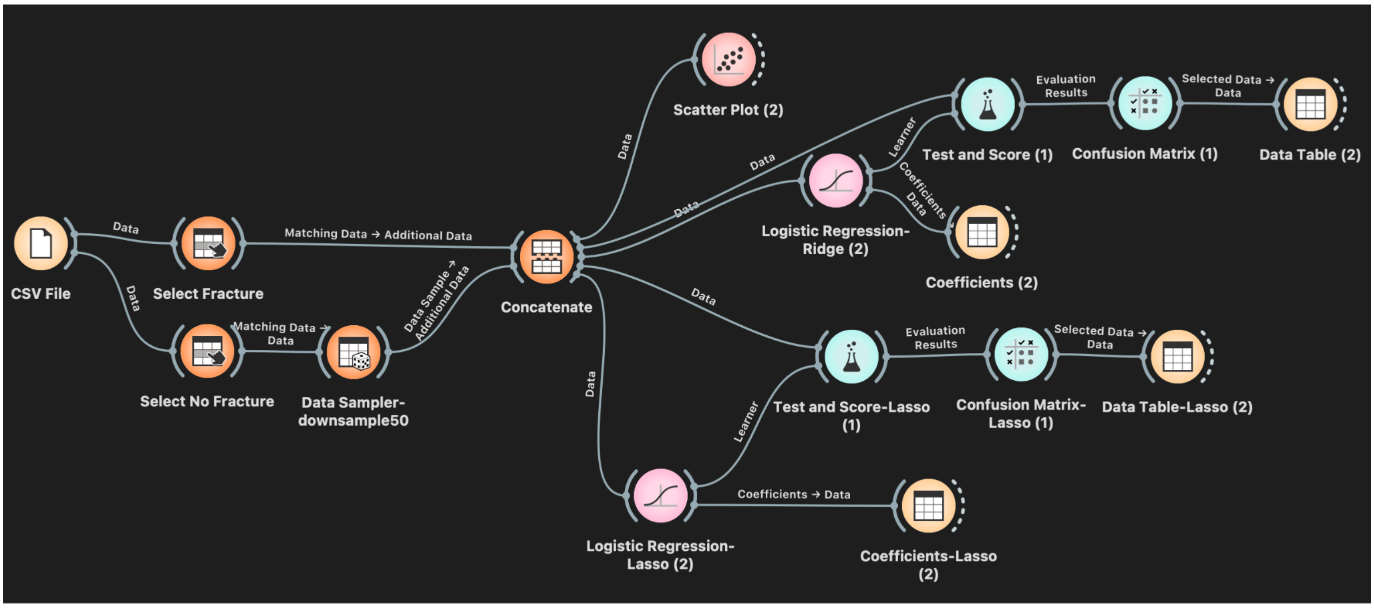


| Actual \ Predicted | fracture | no fracture | Σ |
|---|---|---|---|
| fracture | 86 | 84 | 170 |
| no fracture | 38 | 372 | 410 |
| Σ | 124 | 456 | 580 |

## Lasso

| | | Predicted | | |
|---|---|---|---|---|
| | | fracture | no fracture | Σ |
| Actual | fracture | 103 | 67 | 170 |
| | no fracture | 39 | 371 | 410 |
| | Σ | 142 | 438 | 580 |

In addition, I resampled the data to balance the classes, reducing the number of 'no fracture' instances to 50 to match the number of 'fracture' instances. I then applied the same logistic regression as described above.




The results are as follows.

## Ridge

Coefficients (2)

| | name | no fracture |
|---|---|---|
| 1 | intercept | 6.30171 |
| 2 | Age | -0.0409785 |
| 3 | Weight_kg | 0.0789787 |
| 4 | Height_cm | -0.0657073 |
| 5 | BMD | 2.5931 |

Test and Score (1)

Cross validation

Number of folds: 10

Stratified

Evaluation results for target (None, show average over classes)

| Model | AUC | CA | F1 | Prec |
|---|---|---|---|---|
| Logistic Regression-Ridge (2) | 0.800 | 0.720 | 0.720 | 0.721 |

Predicted

| Actual | fracture | no fracture | Σ |
|---|---|---|---|
| fracture | 34 | 16 | 50 |
| no fracture | 12 | 38 | 50 |
| Σ | 46 | 54 | 100 |

## Lasso

Coefficients-Lasso (2)

| | name | no fracture |
|---|---|---|
| 1 | intercept | 0 |
| 2 | Age | -0.0183849 |
| 3 | Weight_kg | 0.0528584 |
| 4 | Height_cm | -0.0536482 |
| 5 | BMD | 8.82503 |

Test and Score-Lasso (1)

Cross validation

Number of folds: 10

Stratified

Evaluation results for target (None, show average over classes)

| Model | AUC | CA | F1 | Prec |
|---|---|---|---|---|
| Logistic Regression-Lasso (2) | 0.890 | 0.810 | 0.810 | 0.810 |

Predicted

| Actual | fracture | no fracture | Σ |
|---|---|---|---|
| fracture | 41 | 9 | 50 |
| no fracture | 10 | 40 | 50 |
| Σ | 51 | 49 | 100 |