**Arthritis Analysis**

Table of Contents

[1. Data preprocessing: 2](#_Toc106616648)

[1.1 Data Scaling 2](#_Toc106616649)

[1.2 Missing Values Replacing 2](#_Toc106616650)

[2. Initial dataset splitting 2](#_Toc106616651)

[2.1 Training set Oversampling 3](#_Toc106616652)

[3.Attributes and classifier algorithms selection: 4](#_Toc106616653)

[3.1 Using the CorrelationAttributeEval method with the five classifier algorithms 5](#_Toc106616654)

[3.1.1 Attribute selection method: ‘CorrelationAttributeEval’ 5](#_Toc106616655)

[3.1.2 Applying the five classifier algorithms 6](#_Toc106616656)

[3.2 Using the ‘GainRatioAttributeEval’ method with the five classifier algorithms 9](#_Toc106616657)

[3.2.1 Attribute selection method: ‘GainRatioAttributeEval’ 9](#_Toc106616658)

[3.2.2 Applying the five classifier algorithms 10](#_Toc106616659)

[3.3 Using the ‘InfoGainAttributeEval’ method with the five classifier algorithms 12](#_Toc106616660)

[3.3.1 Attribute selection method: ‘InfoGainAttributeEval’ 12](#_Toc106616661)

[3.3.2 Applying the five classifier algorithms 14](#_Toc106616662)

[3.4 Using the ‘CfsSubsetEval’ method with the five classifier algorithms 17](#_Toc106616663)

[3.4.1 Attribute selection method: ‘CfsSubsetEval’ 17](#_Toc106616664)

[3.4.2 Applying the five classifier algorithms 18](#_Toc106616665)

[3.5 Using the ‘OneRAttributeEval’ method with the five classifier algorithms 20](#_Toc106616666)

[3.5.1 Attribute selection method: ‘OneRAttributeEval’ 20](#_Toc106616667)

[3.5.2 Applying the five classifier algorithms 22](#_Toc106616668)

[4. Conclusion 25](#_Toc106616669)

[4.1 The Best Test Result 25](#_Toc106616670)

[4.2 Most relevant attributes 25](#_Toc106616671)

[4.3 What I learned from this project 26](#_Toc106616672)

[4.4 Other Observations from this project 26](#_Toc106616673)

[5 References 26](#_Toc106616674)

# Data preprocessing:

## Data Scaling

I first used ‘Standardize’ to scale the numerical values in each attribute to make sure that the algorithms (such as KNN) would not be biased towards the values with higher magnitude.

Graphical user interface, text, application, table, Excel

Description automatically generated

## 1.2 Missing Values Replacing

Then I used ‘ReplaceMissingValues’ to replace all the missing values. The detailed screenshot is below:

Graphical user interface

Description automatically generated

# Initial dataset splitting

I split the preprocessed dataset (stratified) into 66-34 – meaning the training set is 66% (saved it as ‘project-train.arff.’) and the testing set is 34% (saved as ‘project-test.arff’). The detailed screenshots are below:

The testing set:

Graphical user interface

Description automatically generated

The traning set:

Graphical user interface, application

Description automatically generated

## 2.1 Training set Oversampling

I used SMOTE to balance the class labels in the training set. This is because the original class labels were unbalanced (5218 for label 2 and 2658 for label 1). To produce a high TPR rate for label 1, the class label attribute should be balanced first.

Graphical user interface, application

Description automatically generated

# 3.Attributes and classifier algorithms selection:

I used five attribute selection methods:

1. **CorrelationAttributeEval** - This method measures the Pearson correlation coefficient between each feature and the class attribute. It provides a descending list of the correlation values of each attribute (weka.sourceforge.io,n.d.).
2. **GainRatioAttributeEval** – This method evaluates the attributes by measuring the gain ratio of each attribute. Gain ratio uses the split entropy to reduce the bias caused by the information gain measure.
3. **InfoGainAttributeEval** – This method evaluates the attributes by measuring the information gain of each attribute. Information gain is based on calculating how pure the dataset is for the class attribute. Attributes with the highest information gain can be selected (Wikipedia, n.d.).
4. **CfsSubsetEval** – This method provides a subset of features that best fit the criterion that each selected feature should have high correlation with the class label but also have low correlation with each other.
5. **OneRAttributeEval** – This method evaluates the attributes by using the 1R classifier. It constructs rules and an error rate for each attribute and shows a ranking list of the features providing the best performance. (Jae Young Lee, June 2022)

The classifier algorithms I used are:

1. **Decision Tree** – This algorithm is constructed in a top-down tree model and in an iterative manner. A test attribute is selected by certain statistical measures (such as information gain), then leaf nodes are created based on the selected attributes. (Wikipedia, n.d.)
2. **Naïve Bayes** – This algorithm is constructed based on Bayes theorem with naïve independence assumption between each feature. The probability of each class label can be calculated based on this theorem. (Jae Young Lee, June 2022)
3. **KNN** – This algorithm relies on calculating the distances (such as Euclidean distance) between the new data and its k nearest neighbor tuples and determines its class label by majority voting.
4. **Random Forest** – This algorithm is an ensemble learning method that splits a dataset into multiple trees by bootstrap aggregating. The final classification is determined by the majority voting of those multiple tree classifications. (Wikipedia, n.d.).
5. **Logistic Regression** – This is a supervised algorithm that uses the logistic function to calculate the coefficients from a training dataset. Then the probability can be calculated, and the class label can be classified based on a threshold. (IBM, n.d.)

## 3.1 Using the CorrelationAttributeEval method with the five classifier algorithms

### 3.1.1 Attribute selection method: ‘CorrelationAttributeEval’

Graphical user interface, text, application, email

Description automatically generated

The attributes I chose are the first 10 attributes due to their correlation coefficient rankings. Then I reduced the attributes from both the initial training and testing sets and saved them respectively.

The attributes I chose are shown below:

Graphical user interface, text, application

Description automatically generated

### 3.1.2 Applying the five classifier algorithms

I performed the five algorithms from the reduced training dataset and tested them on the reduced test dataset.

#### Decision Tree:

Graphical user interface, text

Description automatically generated

#### Naïve Bayes:

Graphical user interface, text

Description automatically generated

#### KNN:

Graphical user interface, text

Description automatically generated

#### Random Forest:

Graphical user interface, text

Description automatically generated

#### Logistic regression:

Graphical user interface, text

Description automatically generated

## 3.2 Using the ‘GainRatioAttributeEval’ method with the five classifier algorithms

### 3.2.1 Attribute selection method: ‘GainRatioAttributeEval’

I chose ‘GainRatioAttributeEval’ as my second attribute selection option to reduce the attributes.

Graphical user interface, text, application, email

Description automatically generated

The attributes I chose are the first 10 attributes due to their gain ratio rankings. Then I reduced the attributes from both the initial training and testing sets and saved them respectively.

The attributes I chose are shown below:

Graphical user interface, application

Description automatically generated

### 3.2.2 Applying the five classifier algorithms

I performed the five algorithms from the reduced training dataset and tested them on the reduced test dataset.

#### Decision Tree:

Text

Description automatically generated

#### Naïve Bayes:

Graphical user interface, text

Description automatically generated

#### KNN:

Graphical user interface, text

Description automatically generated

#### Random Forest:

Graphical user interface, text

Description automatically generated

#### Logistic regression:

Text

Description automatically generated

## 3.3 Using the ‘InfoGainAttributeEval’ method with the five classifier algorithms

### 3.3.1 Attribute selection method: ‘InfoGainAttributeEval’

I chose ‘InfoGainAttributeEval’ as my third attribute selection option to reduce the attributes.

Graphical user interface, text, application, email

Description automatically generated

The attributes I chose are the first 10 attributes due to their information gain rankings. Then I reduced the attributes from both the initial training and testing sets and saved them respectively.

The attributes I chose are shown below:

Graphical user interface, text, application

Description automatically generated

### 3.3.2 Applying the five classifier algorithms

I performed the five algorithms from the reduced training dataset and tested them on the reduced test dataset.

#### Decision Tree:

Graphical user interface, text

Description automatically generated

#### Naïve Bayes:

Graphical user interface, text

Description automatically generated

#### KNN:

Graphical user interface, text

Description automatically generated

#### Random Forest:

Graphical user interface, text

Description automatically generated

#### Logistic Regression:

Graphical user interface, text

Description automatically generated

## 3.4 Using the ‘CfsSubsetEval’ method with the five classifier algorithms

### 3.4.1 Attribute selection method: ‘CfsSubsetEval’

I chose ‘CfsSubsetEval’ as my fourth attribute selection option to reduce the attributes.

Graphical user interface, text, application

Description automatically generated

I chose the selected 15 attributes from the initial training and testing sets and saved them respectively. The selected attributes are:

Graphical user interface, text, application

Description automatically generated

### 3.4.2 Applying the five classifier algorithms

I performed the five algorithms from the reduced training dataset and tested them on the reduced test dataset.

#### Decision Tree:

Graphical user interface, text

Description automatically generated

#### Naïve Bayes:

Graphical user interface, text

Description automatically generated

#### KNN:

Graphical user interface, text

Description automatically generated

#### Random Forest:

Graphical user interface, text

Description automatically generated

#### Logistic Regression:

Graphical user interface, text, application

Description automatically generated

## 3.5 Using the ‘OneRAttributeEval’ method with the five classifier algorithms

### 3.5.1 Attribute selection method: ‘OneRAttributeEval’

I chose ‘OneRAttributeEval’ as my third attribute selection option to reduce the attributes.

Graphical user interface, text, application, email

Description automatically generated

I chose the first 10 attributes due to the 1R rankings from the initial training and testing datasets and saved them respectively. The selected attributes are:

Graphical user interface, text, application

Description automatically generated

### 3.5.2 Applying the five classifier algorithms

I performed the five algorithms from the reduced training dataset and tested them on the reduced test dataset.

#### Decision Tree:

Graphical user interface, text

Description automatically generated

#### Naïve Bayes:

Graphical user interface, text, application, email

Description automatically generated

#### KNN(K=4)

Graphical user interface, text

Description automatically generated

#### Random Forest:

Graphical user interface, text

Description automatically generated

#### Logistic regression:

Graphical user interface, text

Description automatically generated

# Conclusion

## 4.1 The Best Test Result

In our dataset, the class label ‘1’ means that the person has/had some form of arthritis, and the class label ‘2’ means that the person does not have such conditions. In practice, especially for medical research, I believe it is more important to predict if patients have arthritis than if patients do not have arthritis. In other words, predicting the class label ‘1’ is more important than predicting the class label ‘2’. Therefore, my main criterion for choosing the best model is that without losing too much TPR rate for class ‘2’, the TPR rate for class ‘1’ has to be as high as possible.

The best test result I obtained for our dataset was from the **‘CfsSubsetEval’ attribute selection method with Logistic Regression classifier algorithm**. This is because both the TPR rates for class 1 and class 2 were above 70% (70.3% for class 2, and 72.8% for class 1). And the overall ROC and PRC values were relatively high as well. Especially when the data points for the class label are unbalanced, this is a better model of classification for both class labels.

The performance matrix of this model is shown below:

Graphical user interface, text, application

Description automatically generated

## 4.2 Most relevant attributes

Since correlation coefficient is usually used to measure the strength of relationship between the features and the class attribute, based on the Pearson correlation values for each variable in our dataset, the five attributes that I think are most relevant to the class attribute are shown below:

|  |  |
| --- | --- |
| Correlation coefficient | Attribute |
| 0.41202 | x.age80 |
| 0.3916 | x.ageg5yr |
| 0.34337 | employ1 |
| 0.28712 | x.age65yr |
| 0.26354 | diffwalk |

## 4.3 What I learned from this project

First, preprocessing of the initial data is important for machine learning models to produce an accurate result for classification or prediction. Data preprocessing can be done in many ways. In the project, replacing the missing values, standardizing the values of the feature attributes and oversampling of the class attribute were used to create a suitable dataset for the above five machine learning algorithms.

Second, stratified sampling is also important especially when class label is unbalanced. Performing stratified sampling when splitting the dataset for machine learning models can help reduce sampling bias in the test set and produce a better outcome for classification.

Third, multiple attribute selection methods and machine learning algorithms can be used for one dataset. Different approaches often produce different outcomes. A better model(s) can be determined based on the expected outcome and certain pre-set criteria.

## 4.4 Other Observations from this project

From this project, I observed that the age-related attributes have relatively high correlation with the class attribute, which means age can be an important factor to decide if a person has arthritis or not. This is because the age related attributes such as x.age80, x.age65yr and x.ageg5yr were often listed in the top 10 attributes when the above attribute selection methods were used. For example, when the CorrelationAttributeEval method was used to measure the correlation coefficients between each attribute and the class label, there were 3 age-related attributes listed in the top 10 features with high correlation values.

Generally speaking, even though the overall accuracy rates for both classes are not very high, I still believe the model that produced the best performance could be an effective tool for arthritis prevention and risk factor identification, especially in combination with other professional medical techniques and methods.

# References

1. *weka.sourceforge.io. (n.d.).* [CorrelationAttributeEval (sourceforge.io)](https://weka.sourceforge.io/doc.dev/weka/attributeSelection/CorrelationAttributeEval.html)
2. *IBM. (n.d.).*[*What is Logistic regression? | IBM*](https://www.ibm.com/topics/logistic-regression)
3. *Wikipedia. (n.d.).* [*Random forest - Wikipedia*](https://en.wikipedia.org/wiki/Random_forest#:~:text=Random%20forests%20or%20random%20decision%20forests%20are%20an,forests%20correct%20for%20decision%20trees%27%20habit%20of%20)
4. *Wikipedia. (n.d.).* [*Decision tree - Wikipedia*](https://en.wikipedia.org/wiki/Decision_tree)
5. *Jae Young Lee (June 2022). Blackboard@BU.* [*Content (bu.edu)*](https://learn.bu.edu/ultra/courses/_85840_1/cl/outline)