James Bloor

Boston University Metropolitan College

MET CS 699

Project Assignment

Summer 1 2022

BU ID: U45448564

Preprocessing

After loading the data into Weka, I split the data into a testing and training dataset as wanted. 66% for training and the rest went to the testing. When this was done the class distribution did remain consistent to what was seen in the original data, the ratio for the entire data set was about 2-1 (2’s were more frequent than 1’s). While I was looking at the training set, I knew that it would be valuable to balance the data so that there were as many 1 instances as 2. To make the data balanced I used the SMOTE feature in Weka to sample the 1 data with replacement. Now the training dataset is ready for our next steps.

Attribute Selection

Below are the five attribute selectors that I used for my project.

CfsSubsetEval:

This attribute selector looks at a subset of attributes and the predictability each feature must select attributes. This method also looks at redundancy between features to limit the attributes selected. [employ1, income2, height3, children, alcday5, diffwalk, physhlth, menthlth, fmonth, persdoc2, checkupl, sleptim1, x.age.g, x.ageg5yr, x.rawrake, x.mracel]

ClassifierAttributeEval:

For the Classifier Attribute Evaluator uses a given classifier to measure the worth of attributes and ranks them from there. The classifier given here was ZeroR. I selected the top 10 attributes here once they were ranked. [x.state, genhlth, chckdnyl, iyear, fmonth, dispcode, hlthplnl, rmvteth4, menthlth, diabete3]

CorrelationAttributeEval:

Similar to the Classifier Attribute Evaluator, the correlation Attribute Evaluator evaluates the worth of each attribute vs the class. It measures this by Pearson’s correltation. I selected the top 10 attributes here once they were ranked. [x.age80, x.age5yr, employ1, x.chldcnt, x.age65yr, checkupl, x.age.g, persdoc2, x.hcvu651, difwalk]

GainRatioAttributeEval:

The Gain Ration Attribute Evaluator looks at the gain ratio with respect to a class for each attribute. I selected the top 10 attributes here once they were ranked. [employ1, children, x.age.g, x.ageg5yr, physhlth, sleptim1, income2, x.chldcnt, alcday5, persdoc2]

InfoGainAttributeEval:

Lastly, I used the Information Gain Attribute Evaluator to rank attributes. Here, each attribute is measured by the info gain with respect to the class. I selected the top 10 attributes here once they were ranked. [x.ageg5yr, employ1, htin4, income2, htm4, x.age80, fmonth, sleptim1, height3, physhlth]

Classifier Algorithms

Naïve Bayes:

Naïve Bayes uses Bayes rule and assumes that each attribute is independent. From here, it estimates a precision value numerically, based on information from the training data.

Simple Logistic:

The Simple Logistic regression in Weka is a linear classifier. It used regression-based method to determine the classification.

J48:

J48 is a machine leaning decision tree that dopes well with both categorical and continuous data.

Random Forest:

A Random Forest is built of many decision trees, all constructed differently. Form there, this method takes the majority of a class predicted to determine the class of a given tuple.

Random Tree:

The Random Tree algorithm is a supervised machine learning classifier that randomly chooses attributes at each node, attempting to best determine a given class.

Test Results

Below are my 25 test results with the corresponding screenshots and wanted test results.

CFS SubsetEval + Naïve Bayes

Graphical user interface, text, application

Description automatically generated

CFS SubsetEval + Simple Logistic

Graphical user interface, text, application

Description automatically generated

CFS SubsetEval + J48

Graphical user interface, text

Description automatically generated

CFS SubsetEval + Random Forest

Graphical user interface, text

Description automatically generated

CFS SubsetEval + Random tree

Graphical user interface, text, application

Description automatically generated

ClassifierAttributeEval + Naïve Bayes

Graphical user interface, text, application

Description automatically generated

ClassifierAttributeEval + Simple Logistic

Graphical user interface, text, application

Description automatically generated

ClassifierAttributeEval + J48

Graphical user interface, text, application

Description automatically generated

ClassifierAttributeEval + Random Forest

Graphical user interface, text

Description automatically generated

ClassifierAttributeEval + Random tree

Graphical user interface, text, application, email

Description automatically generated

CorrelationAttributeEval + Naïve Bayes

Graphical user interface, text

Description automatically generated

CorrelationAttributeEval + Simple Logistic

Graphical user interface, text

Description automatically generated

CorrelationAttributeEval + J48

Graphical user interface, text

Description automatically generated

CorrelationAttributeEval + Random Forest

Graphical user interface, text

Description automatically generated

CorrelationAttributeEval + Random tree

Graphical user interface, text

Description automatically generated

GainRatioAttributeEval + Naïve Bayes

Graphical user interface, text, application, email

Description automatically generated

GainRatioAttributeEval + Simple Logistic

Graphical user interface, text, application

Description automatically generated

GainRatioAttributeEval + J48

Graphical user interface, text, application

Description automatically generated

GainRatioAttributeEval + Random Forest

Graphical user interface, text

Description automatically generated

GainRatioAttributeEval + Random tree

Graphical user interface, text, application

Description automatically generated

InfoRatioAttributeEval + Naïve Bayes

Graphical user interface, text, application

Description automatically generated

InfoRatioAttributeEval + Simple Logistic

Graphical user interface, text

Description automatically generated

InfoRatioAttributeEval + J48

Graphical user interface, text, application

Description automatically generated

InfoRatioAttributeEval + Random Forest

Graphical user interface, text, email

Description automatically generated

InfoRatioAttributeEval + Random tree

Graphical user interface, text, application, email

Description automatically generated

Attribute Selection and Classifier’s Best Performance

CFS SubsetEval attribute selection method with a random forest algorithm gave me the best performance).

List of Attributes for the Best Performance

The attribuetes in the CFS SubserEval + Random Forest were employ1, income2, height3, children, alcday5, diffwalk, physhlth, menthlth, fmonth, persdoc2, checkupl, sleptiml, x.age.g, x.ageg5yr, x.rawrake, x.mracel.

Best Test Results

Graphical user interface, text

Description automatically generated

Discussion

Criteria for Best Model:

In the data that we were given looks at if a person was told then have a form of arthritis, rheumatoid arthritis, gout, lupus, or fibromyalgia. In the data these tuples are noted as a 1, everyone else is a 2. This means that the “1” accuracy is what is more important here than the 2 because we would rather bucket people in the 1 group then leave them out and suffer the consequences someone might have if they ever are in pain because of this missed diagnosis. Highest overall accuracy was done in the random forest model with the CFS SubsetEval attribute selector, 73.6%. However, I do not want to solely select the “best” model on accuracy. We need to focus on what the precision is on class 1. This will tell us how well the model did on identifying the true positives and creating false positives. A higher precision score will reward us for true positives and punish us for false positives. The random forest model with the CFS SubsetEval attribute selector also had the highest precision score out of all our 25 models.

Five most Relevant Attributes:

The five most relevant attributes in my eyes are employ1, income2, physhlth, sleptim1, and persdoc2. I find these five to be the most relevant because, not only were they used in the “best” model, but they were also determined to be valuable attributes when attribute selectors ranked attributes. All five of these attributes were used in 15 of the 25 models above. All five also make sense in the diagnosis of arthritis, rheumatoid arthritis, gout, lupus, or fibromyalgia. Employment shows if a person is working (who employs them) or not. The older you are, the more likely one of these diagnoses have been given to you. Income has a ton to do with what people are diagnosed with, most of the time the less money one makes, the more likely they are to have poor health. Physhlth asks how many of the last 30 days someone had a physical illness/injury. If a person has one of these ailments, it might be caused by arthritis, rheumatoid arthritis, gout, lupus, or fibromyalgia. Persdoc2 notes how many care providers one considers they have. With more health-related concerns, the more providers one can have. Sleptim1 is how much sleep one gets in a 24-hour period, on average. This can help us with determine if someone is having trouble sleeping with one of these diagnoses. While these attributes also cannot also create a viable model, they all play an important role in the process.

Learnings:

After running these 25 models, I learned some good lessons that I can apply to real world mining. Before taking this class, I did not see the value in trying many different models to find the best fit. From class and this project, I now see the value in doing so and how different models can be when they are measured. Even looking at my results above, I believe that some of my models can be improved. Whether that is including more attributes when training the data or using a different model that I did not utilize. The other lesson I took away from this project was not to always focus on the accuracy of a model. As we spoke about in class, looking at other measures is valuable to the overall model and human knowledge is needed to decide what model is best for a specific situation. IN the situation we have here, we want to make sure we capture as many people as possible who have been diagnosed with arthritis, rheumatoid arthritis, gout, lupus, or fibromyalgia. While my best model did have the highest accuracy, its F-measure and precision for the 1 class was some of and the best in all my models. Thinking critically about whenever situation we are plac3ed in when designing a model is always going to be needed.

Observations:

Overall, I am pleased with my results but still wanting a bit more from my models. While my accuracy/precision/F-measures are good in my opinion, I believe that some improvements can be made by digging in deep to the attributes to find which can best improve my models. This project did help reinforce many teachings from class that I was not expecting. Also, I had never heard of Weka before taking this class. While I will most likely stick to R for future models, for someone that does not know how to build models in python or R, Weka is extremely user friendly. With only a few clicks a user can create model with no limits. In the end, I am very pleased with my learning in this class and through this project, I can immediately bring the tools and learns from this class to my workplace and utilize them to wherever life takes me in the future.