Dominique da Silva u21629944

ASSIGNMENT 3

Table of Contents

Assignment Objective

Machine Learning Models for Breast Cancer Classification.

This report presents the implementation and evaluation of 3 learning models for breast cancer classification. The models include an Artificial Neural Network (ANN), a Genetic Programming Classification Algorithm (GP), and a C4.5 Decision Tree. The goal is to analyze and compare the performance of these models in predicting the recurrence-events of breast cancer using the provided data set from the UCI Machine Learning Repository.

The breast cancer dataset consists of 286 instances, with each instance described by nine attributes, including age, menopause status, tumor size, number of involved lymph nodes, presence of node-caps, degree of malignancy, breast location, breast quadrant, and irradiation. The class attribute indicates whether the patient experienced a recurrence event or not, with two classes: "no-recurrence-events" and "recurrence-events." The dataset was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.
The data set is split into a training set (80% => 229 instances) and a testing set (20% => 57 instances).

Data Preprocessing/Wrangling

Categorical mappings or the attributes are stored as arrays. Each instance is evaluated, and each attribute of an instance is individually evaluated. This algorithm takes this single instance attribute values and finds the index of the value in the mapping array. If the value has been denoted by “?”, i.e. we have a missing value we opted to rather use the average of the data set for that particular data set, rather than deleting the entry or instance.
This was deemed to be a reasonable approach for the following reasons:
Data retention. We do not want to lose valuable information, especially with respect to the other attributes of the instance.

Preservation of distribution. It helps to maintain the statistical properties of the data. Deleting the instance may potentially affect the distribution and lead to biased results.

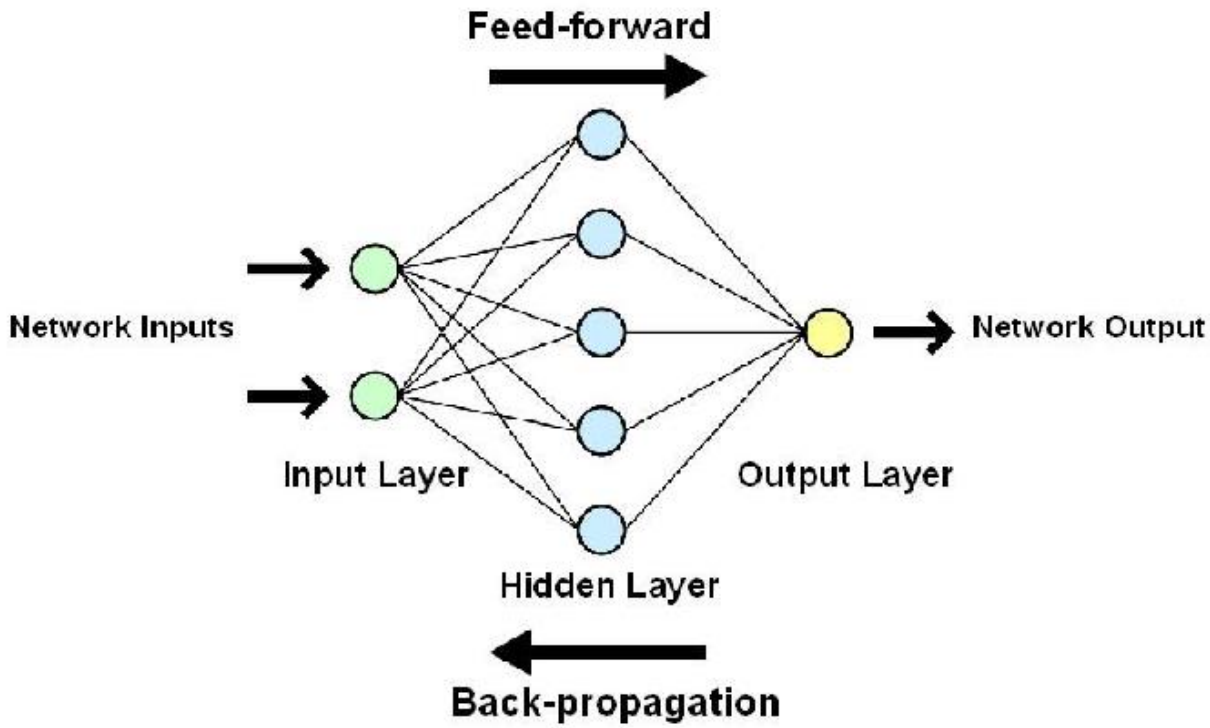
Impact of model performance. Deleting the instance reduces the size of the data set. This could affect our model’s ability to learn and generalize. But imputing missing values with the average, we retain the overall structure of the data set, potentially improving the model’s performance.

Dominique da Silva u21629944

Artificial Neural Network

Model Description

ANN (Artificial Neural Network) is used for classification tasks. We have an input layer, one or more hidden layers, and an output layer. The number of input neurons is the number of features of the dataset.



Parameters (indicating seed value for randomization as well):

- Input Size: The number of features in the data set, i.e. the 9 attributes of columns of the given data set. This is the number of input neurons in the network.
- Hidden Size: The number of neurons in the hidden layer of the network. This is a user defined value. This program uses 10 neurons.
- Output Size: The number of output neurons. We are using an activation function and doing classification with 2 classes, therefore the size would be set to 1.
- Hidden Weights, Hidden Biases, Output Weights: They are randomly initialized to be between the values -0.5 and 0.5
- Learning Rate: Rate at which the weights are updated during the learning process. It controls the magnitude of weight adjustments in each iteration of back-propagation. Selection process of this values keeps in mind that the network should not overshoot, or get stuck in a local optima, ensuring that the network converges to an optimal solution. Training used a value of 0.1 as an initial value, and yielded satisfactory results. After many changes, it was found that 0.12 increased the accuracy of the data set from the original 78% to 85%. 0.2 yielded a low accuracy of 66%, thus it was too large.
- Random: Random(1234)

Stopping Condition

This is defined by the number of epochs. An epoch refers to a complete iteration over the entire training dataset. A fixed number of epochs was used, 10, it was a simple approach yet very effective. 20 Epochs was tested, yet didn't yield any improvements in results.

Feed-forward learning

Hidden Layer: ReLU (Rectified Linear Unit) activation was used. It is commonly used in hidden layers within a neural network to introduce non-linearity and allows the network to learn complex patterns. Also note that only one hidden layer was used for the ANN implementation. This was due to being efficient enough and not adding any additional complexity to the code.

Output Layer: Sigmoid activation function was used, since it is commonly used in binary classification, i.e. the output should be between 0 and 1. Our input value is then altered to a probability like interpretation, where the values close to 0 is interpreted as one class and the values in the other class is represented as values closer to 1.

Back-propagation

Back-propagation is used for training for the training the model. We iteratively adjust the weights and biases in the network to minimize the error between the predicted output and the actual output.
In this implementation, the backpropagation algorithm is applied by updating the weights and biases in two steps: updating the weights and bias between the hidden layer and the output layer, and updating the weights and biases between the input layer and the hidden layer.

Results

From given information about the data set we know that the data set is imbalances, having 201 instances of class no-recurrence-events and 85 instance of recurrence-events. This would also negatively impact the results of the model since it will be heavily biased towards our majority class, which is evident in our table. This could cause unrealistic classifications of a recurrence-event being classified as a no-recurrence-event, which could potentially have an impact on the health of a patient.

Accuracy

Total instances: 57

Correct predictions: 49

Accuracy = (Correct predictions / Total instances) * 100

Accuracy = (49 / 57) * 100

Accuracy ≈ 85.96%

The accuracy of the ANN on the given testing set is approximately 85.96%.

F-Measure

True positives (TP): The number of instances where the actual classification is recurrence-events and the prediction is also recurrence-events. In this case, TP = 2.

False positives (FP): The number of instances where the actual classification is no-recurrence-events but the prediction is recurrence-events. In this case, FP = 1.

False negatives (FN): The number of instances where the actual classification is recurrence-events but the prediction is no-recurrence-events. In this case, FN = 3.

Precision = TP / (TP + FP)

Precision = 2 / (2 + 1)

Precision = 2/3

Recall = TP / (TP + FN)

Recall = 2 / (2 + 3)

Recall = 2/5

F-measure = 2 * (Precision * Recall) / (Precision + Recall)

F-measure = 2 * (2/3 * 2/5) / (2/3 + 2/5)

F-measure ≈ 0.571

Since our F-measure value is around 0.5, this suggests a balanced trade-off between precision and recall. This indicates that the model is moderately successful in an accurate identification, but there is room for improvement. Ideally we would have wanted a value closer to 1 or even the value 1, since we want to ideally achieve perfect balance between precision and recall.

Dominique da Silva u21629944

Genetic Programming Classification Algorithm

Model Description

A machine learning algorithm that is inspired by the principles of evolution and natural selection. The algorithm normally starts with an initial population of randomly generated programs, typically represented as trees. This implementation GP model is for binary classification.

Parameters:

- Population size: Given in the specification that it had to be set to 100.
- Maximum Number of Generations: Given that it had to be set to 50 generations in the given Assignment specification.
- Fitness Function: It measures the performance of a decision tree by calculating the accuracy of its predictions on the given dataset. The accuracy is the ratio of correctly predicted instances to the total number of instances in the dataset. The higher the accuracy, the better the fitness of the decision tree.
- Selection Method: Using Tournament Selection style with a size of
- Genetic Operators: Crossover and Mutation. We have the parameters set to 0.6 and 0.02 respectively.
- Termination Conditions: We iterate for the maximum number of generations, and then terminate our training process.
- Tree Depth: Specified that care should be taken such that the tree size grows exponentially.
- Node Constraints: The maximum depth of the decision tree. It is controlled by the maxDepth parameter in the generateRandomTree method. The maxDepth specifies the maximum number of levels or depth allowed in the decision tree. If the current depth exceeds the maxDepth or a random probability condition is met, a leaf node (classification node) is generated. This constraint helps to limit the complexity of the decision tree and prevent overfitting.

Flow Control

1. Initialization:
   - A random seed is set for reproducibility.
   - The initial population of decision trees is generated using the generateRandomTree function.
2. Evolution:
   - For each generation (controlled by generations_num):
   - Tournament selection (tournamentSelection) is applied to select individuals from the current population based on their fitness (accuracy) values. A tournament size of 4 is used.
   - Crossover (crossover) is performed with a probability determined by crossover_rate. Random pairs of selected decision trees undergo subtree crossover to create offspring.
   - Mutation (mutate) is applied to each selected decision tree with a probability determined by mutation_rate. Mutation introduces random changes to individual decision trees.
   - The selected individuals, including the offspring, form the population for the next generation.
   - The fitness (accuracy) of each decision tree in the new population is evaluated using evaluateAccuracy.
   - The best decision tree and the average accuracy of the population are printed for monitoring progress.
3. Termination:

- After the specified number of generations, the evolution process ends.
- The best decision tree found throughout the evolution is stored in bestTree and returned as the final result.
- Overall, the algorithm iteratively evolves the population by selecting, crossing over, and mutating decision trees to improve their fitness. The process continues until the specified number of generations is reached, producing the best decision tree that accurately predicts the class labels for the given dataset.

Results:

Accuracy

Total instances: 57

Correct predictions: 47

Accuracy = (Correct predictions / Total instances) * 100

Accuracy = (47 / 57) * 100

Accuracy ≈ 82.46%

The accuracy of the GP on the given testing set is approximately 82.46%.

F-Measure

True positives (TP): The number of instances where the actual classification is recurrence-events and the prediction is also recurrence-events. In this case, TP = 4.

False positives (FP): The number of instances where the actual classification is no-recurrence-events but the prediction is recurrence-events. In this case, FP = 3.

False negatives (FN): The number of instances where the actual classification is recurrence-events but the prediction is no-recurrence-events. In this case, FN = 8.

Precision = TP / (TP + FP)

Precision = 4 / (4 +3)

Precision = 4/7

Recall = TP / (TP + FN)

Recall = 4 / (4 + 8)

Recall = 4/12 = 1/3

F-measure = 2 * (Precision * Recall) / (Precision + Recall)

F-measure = 2 * (4/7* 1/3) / (4/7+ 1/3)

F-measure ≈ 0.344

We did see an improvement in f-measure when the maximum depth of a tree was set from 7 to 5. The model could have been overfitting, seeing as we have 9 features, and just trying to map all 9 features into the tree instead of doing some feature reduction.

The F-measure that was calculated suggests that the model's performance in terms of the precision and recall is low. This indicates room for improvement in the model's ability to correctly classify instance of recurrence-events, the model is only identifying 33.3% of the recurrence-events correctly.

Dominique da Silva u21629944

C4.5 Decision Tree

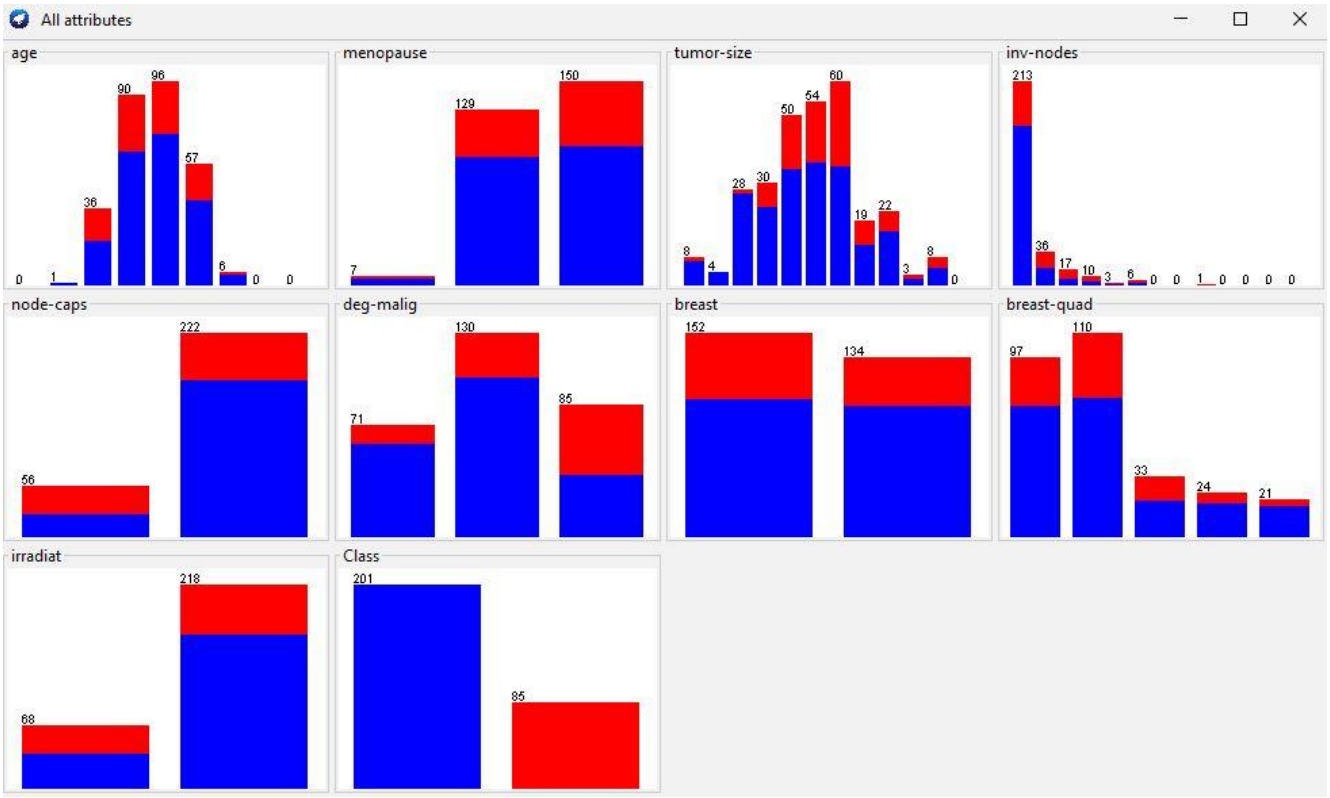
Model description:

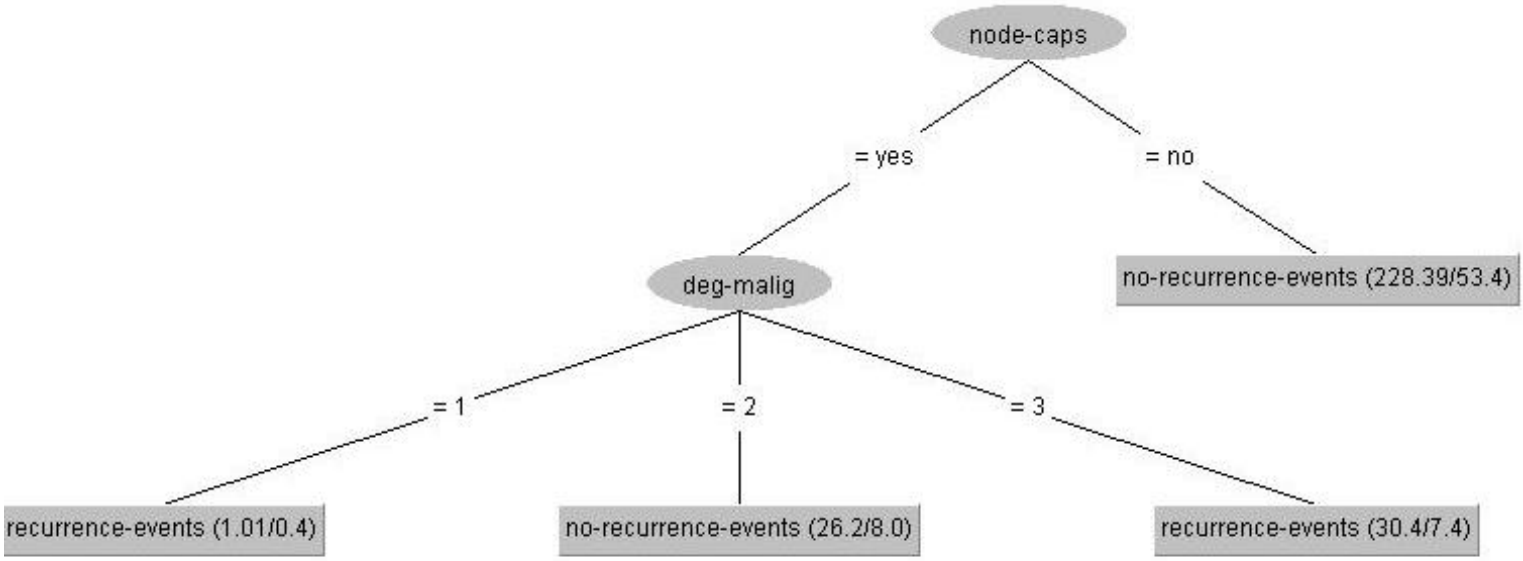
A decision tree is a model with a tree-like structure that aims to model probabilistic outcomes and all their consequences which is deemed necessary to model. Each branch of the decision tree represents a decision that could be made.

This was much easier to implement compared to the other two classification algorithms, however, decision trees are deemed to be instable since tiny changes in the data may lead to the entire structure of the tree being adapted, potentially reducing key features in our data set.

ReadMe file

Classifier Output file




node-caps
= yes
= no
deg-malig
no-recurrence-events (228.39/53.4)
= 1
= 2
= 3
recurrence-events (1.01/0.4)
no-recurrence-events (26.2/8.0)
recurrence-events (30.4/7.4)

Dominique da Silva u21629944

Comparative Analysis between Models

ANN (Artificial Neural Network) and GP (Genetic Programming), and we will compare their performance in terms of accuracy and F-measure.

Accuracy:

ANN: The accuracy of the ANN model on the given testing set is approximately 85.96%. This means that out of 57 instances, the model made 49 correct predictions.

GP: The accuracy of the GP model on the given testing set is approximately 82.46%. This means that out of 57 instances, the model made 47 correct predictions.

From the accuracy comparison, we can see that the ANN model performs slightly better than the GP model in terms of overall accuracy. The ANN model achieved a higher percentage of correct predictions on the testing set.

F-Measure:

ANN: The F-measure of the ANN model is approximately 0.571. This metric takes into account both precision and recall, providing a balanced measure of the model's performance. The ANN model achieved a precision of 2/3 and a recall of 2/5.

GP: The F-measure of the GP model is approximately 0.344. The GP model achieved a precision of 4/7 and a recall of 1/3.

In terms of F-measure, which considers both precision and recall, the ANN model again outperforms the GP model. The higher F-measure of the ANN model indicates a better balance between precision and recall compared to the GP model.

Overall, based on the provided accuracy and F-measure values, the ANN model demonstrates better performance than the GP model. It achieves higher accuracy and F-measure, indicating that it has a higher ability to make correct predictions and maintain a balance between precision and recall.