Machine Learning

CSI 5155

Explainable AI (XAI), Interpretability, and Trustworthiness

Assignment 3

Authors
Alexandra Sklokin (300010511)

Professor
Dr. Herna Viktor

November 24, 2021

Contents

1 Methodology

For this assignment I used *jupyter notebooks*, Excel spreadsheets, and the *sklearn Python* library. Please find all of the code for this assignment in my git repository[1].

For Part A, I performed preprocessing, hyperparameter tuning, training and evaluation.

I decided not to transform the categorical or numerical values, since the purpose of this assignment is Explainable AI (XAI), thus I choose to prioritize interpretability rather than high scores. By keeping the original data intact, we can better explain, interpret, and trust the results obtained by the Decision Tree Classifier.

I used Lasso and filter-based feature selection methods, with and without oversampling, to create different datasets. From six possible datasets, I selected that which resulted in the highest F1-score with the Decision Tree. The best scores were obtained from oversampling the whole dataset (without any feature selection).

Using *GridSearchCV* and this dataset, I obtained these hyperparameters: *criterion: 'gini', max_depth: 9, min_samples_leaf: 2, min_samples_split: 3*.

Following training, I obtained a weighted F1-score of 92.92% and accuracy of 92.93%.

[1] https://github.com/alexandrasklokin/CSI5155/tree/main/Assignment3

2 Experimental Results

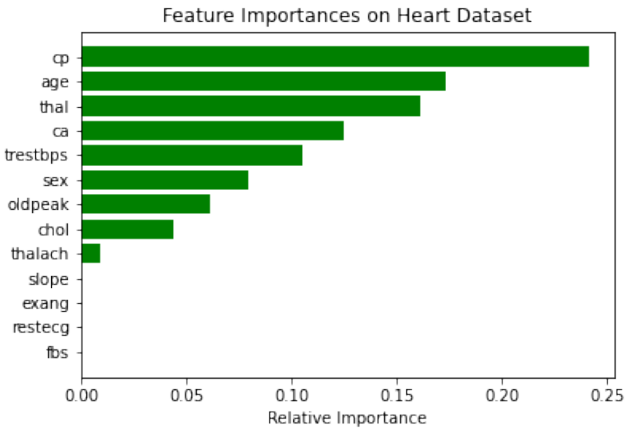

Figure 1: Graph of Feature Importance for Decision Tree Predictions over Heart Dataset

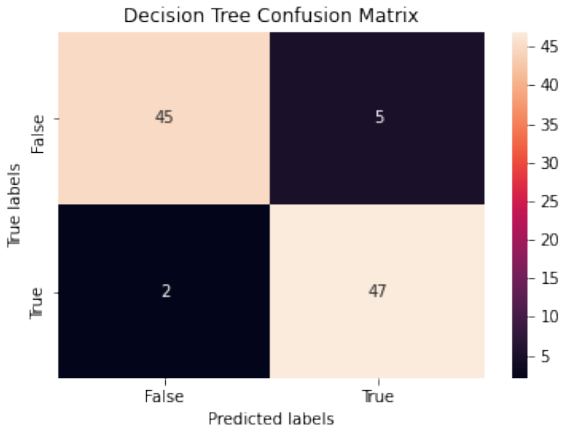

Figure 2: Decision Tree Confusion Matrix

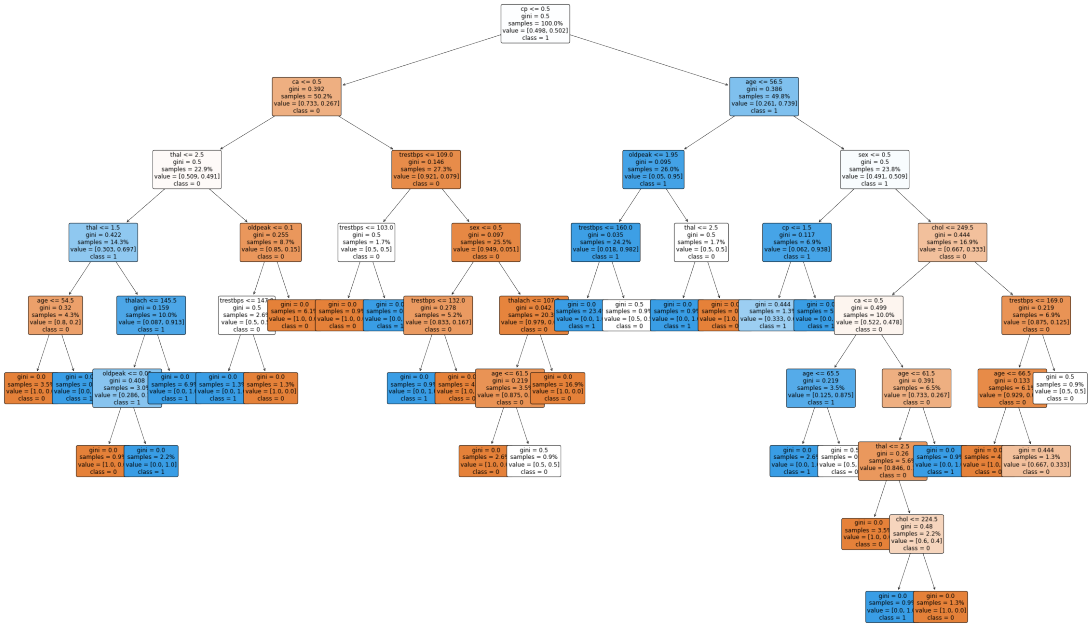

Figure 3: Visualization of Best Decision Tree for Heart Dataset, with Proportions

3 Discussion

3.1 Part A

1) Explain how, and why, the algorithm made a specific decision.

To construct a Decision Tree, we build a binary tree by recursively selecting the best feature to split on. For each leaf and subset (of features and data), we continue to split if both classes are present, and stop otherwise (*Appendix A*).

Feature importance is calculated as the reduction in the criterion used to select split points. On the graph of feature importance (*Figure 1*), we see a gradual decrease in feature importance, with the following being the 3 most important features:

- **cp** chest pain type
- **age** age in years
- **thal** 3 = normal; 6 = fixed defect; 7 = reversable defect

We can only explain this using statistics, since the Decision Tree does not know any scientific justifications, but simply makes predictions based on what was learned from the training data. Intuitively speaking, this hierarchy of features seems trustworthy, since we know that chest pain, age and defects are linked to the probability of having a heart disease.

In the visualization of our Decision Tree (*Figure 3*), we see that important features are located at the top, clearly because they lead to the most *Gini index* reduction.

We can also visualize the paths followed to make predictions (*Figure 4*). It is clear that we begin at the top of the tree, and using the feature values, traverse until we reach a leaf with a class value, whether it be correct or not.

2) Explain why the algorithm did not do something else.

Using hyperparameter tuning, we obtained the *Gini index* as the optimal criterion for splitting. This metric measures the expected error if we were to label examples in the leaf randomly. In the resultant Decision Tree (*Figure 3*), we note the *Gini indices* at each node range between 0.0 and 0.5. For each node, the feature with the smallest *Gini index* was chosen, therefore the algorithm could not have chosen any other feature, since their *Gini indices* would have been larger. Any other split would not improve the purity at the nodes.

When making predictions, it is always very clear which path should be followed, as explained in the previous question. There is no way to obtain a different prediction or follow a different path, for the same feature values. In

our example (*Figure: 4*), we see that for each instance there is exactly one path which can be followed for each example. We can also use contrastive explanations, by taking comparing the paths followed by two different instances, resulting in two different terminal leaves (and possibly two different predictions).

3) Discuss when the algorithm succeeded and when it failed.

The model successfully made predictions for both the negative and positive classes. Using a confusion matrix (*Figure 2*), we can clearly see that the model was very accurate, and rarely made any wrong predictions.

The model failed to make correct predictions for seven test instances (*Table 1*). We can visualize two such examples where the values in the features form a path through the Decision Tree, resulting in a wrong prediction (*Figure 4*). We cannot know the exact reason that these errors occur, since the model does not 'know' any a-priori information, but rather learns from patterns in the training data. However, errors can be due to overfitting or underfitting the training data. After all, no model is perfect (and if it is, then we should be suspicious)!

4) Explain how you would decide if the resultant model can be trusted.

We can trust our model if it is accurate, interpretable, ethical and fair.

We can use statistical metrics to evaluate the trustworthiness of the resultant model. In this case, we achieved a very high accuracy (92.93%), which represents the number of observations which were correctly classified. Therefore, we can trust our model to make correct predictions most of the time. We also observed a good F1-score (92.92%), which is the statistic which leverages Precision and Recall. We note that our model predicted 0 or 1 in about equal proportions, which tells us that it is not biased to one class over the other (*Figure 2*).

Trustworthiness and interpretability go hand-in-hand. A reason to 'trust' a model is if we can easily understand it. As explained above, our Decision Tree was constructed using statistical methods which have been explored and taught for many years. In fact, Decision Trees are considered part of Transparent ML, because the resultant model is not a 'black-box' which takes input and using complex methods, finds an output. Rather this model can be easily visualized (*Figure 3*), and even used to make predictions (*Figure 4*).

Finally, we would like to trust our model to make ethical, fair, and lawful decisions. A common counter-example is Amazon's scrapped hiring system, which was biased towards male applicants. In our case, the domain of the data should not lead to any discrimination. Additionally, we could check if the assumptions made by the Decision Tree match scientific research. For example, it is well known that the probability of heart disease increases with age. We can refer to our visualization (*Figure 3*) and decide that this trend is generally upheld. We may also employ an expert in the field (ex. cardiologist) to audit our results and model.

In general, I would trust the resultant model, for the reasons mentioned above- the high statistical scores, transparency of Decision Tree Model, and fair decision making.

5) Explain how the algorithm could potentially improve its predictions.

There are several ways we could potentially improve the predictions of the Decision Tree Classifier. Firstly, Decision Trees tend to have high variance, and are very sensitive to noisy data. To combat noisy data we can use some/all of the following strategies:

- **More Data** can help our model to better identify the true behaviour of the target. We hope that by collecting more data, the proportion of noisy data is lower, and therefore the effect is smaller. This dataset had just over 300 instances, which is quite small. As we know, in machine learning 'garbage in, garbage out'! This method will not affect the interpretability of results.
- **Principle Component Analysis (PCA)** is commonly used to reduce noise from data by 'omitting' the features that contain noisy data. However, this method will not be as interpretable of XAI, since we are creating new features which may not have any 'meaning'.
- **Regularization** aims to reduce model flexibility, to avoid overfitting, and handle noisy data. This method would result in interpretable results, but the explainability, fairness and trustworthiness may be affected. We would not be able to easily explain why a model made a certain decision.
- **Tree Pruning** is another method to reduce overfitting and handle noisy data. This can make our model even more interpretable, and therefore more trustworthy, since we are reducing the size of the Decision Tree Model.

Finally, we can use ensembles to improve our Decision Tree results. Boosting and bagging are two ensemble methods which can be applied to the Decision Tree Classifier, and can generally improve prediction scores. Boosting algorithms will result in harder to interpret models, but there are visualization tools which can help.

3.2 Part B

1) Write a summary, of 400 to 500 words, reflecting on XAI, interpretability, fairness, and trustworthiness, when considering the machine learning models you have developed in this course and the datasets we used.

XAI aims to explain the reasoning behind AI predictions, increasing their trustworthiness, interpretability, and fairness. As AI decisions become more critical to our lives, industries (ex. law, medicine, finance, etc.) may become reluctant to trust AI if they cannot understand the building blocks of these algorithms. Miller uses social science to explain the following four characteristics of explainability for AI models: "(1) why–questions are contrastive; (2) explanations are selected; (3) explanations are social; and (4) probabilities are not as important as causal links" [3].

A ML algorithm is interpretable if we are able to explain how a model is trained, and how decisions are made. Trustworthiness means we can justify why an algorithm made its predictions, and that its decision-making will be accurate, ethical, unbiased, fair, etc. The definition of a fair model is controversial, but generally refers to a model where predictions are independent of 'sensitive' features (race, gender, etc.).

We tend to consider complex models as trustworthy and simple models interpretable. Contrary to popular belief, there is not necessarily a trade-off between prediction accuracy and interpretability. In the domain of criminal justice, Angelino has proposed a simple two-feature rule-based model, just as accurate as the popular black-box COMPAS model [1]. In the field of computer vision, adding interpretability constraints to deep networks leads to more trustworthy results, with no damage to accuracy [2].

A popular approach for interpretability is visualization, for example with the DT. If we can visualize it, we can trust and interpret the results, and even evaluate decision fairness. By contract, RFs and GBEs, comprised of many trees, may be too large to interpret. There do exist post-hoc methods of visualization, where results are summarized using trees. KNNs and DT are human-friendly models which can be visualized and explained with Miller's principles [3]. Predictions made by SVMs can be visualized using white-box methods, but training relies on hard to interpret mathematics [5]. The same applied to MLPs, a class of neural networks, which, although visualize, can not be easily trusted.

Rudin from Duke University says, "Trusting a black box model means that you trust not only the model's equations, but also the entire database that it was built from" [4]. The *Online_Shopping_Intention* dataset was highly unbalanced, but after rebalancing, we can no longer trust that the training data reflects the true distribution of shopping intent among online users. Models trained on the unbalanced dataset may not be fair. The *Heart* dataset was small and well documented, making it highly interpretable; we could select specific examples, and form contrastive explanations. However, this dataset was quite small, meaning we may not be able to trust our models to make accurate predictions on out-

side data. *Marketing_Campaign* is not well documented and has missing entries, meaning ML engineers would have to make (possibly non-trustworthy) assumptions. We would also question whether the predictions made on this dataset are fair, since it is unbalanced, and some of the features might be controversial towards making predictions (ex. sex).

4 References

[1] Elaine Angelino et al. *Learning Certifiably Optimal Rule Lists for Categorical Data*. 2018. arXiv: 1704.01701 [stat.ML].

[2] Chaofan Chen et al. *This Looks Like That: Deep Learning for Interpretable Image Recognition*. 2019. arXiv: 1806.10574 [cs.LG].

[3] Tim Miller. *Explanation in Artificial Intelligence: Insights from the Social Sciences*. 2018. arXiv: 1706.07269 [cs.AI].

[4] Cynthia Rudin and Joanna Radin. "Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From An Explainable AI Competition". In: *Harvard Data Science Review* 1.2 (Nov. 22, 2019). DOI: 10.1162/99608f92.5a8a3a3d. URL: https://hdsr.mitpress.mit.edu/pub/f9kuryi8.

[5] Farhad Shakerin and Gopal Gupta. *White-box Induction From SVM Models: Explainable AI with Logic Programming*. 2020. arXiv: 2008.03301 [cs.AI].

[6] Herna Viktor. *Topic4 Trees (CSI5155 Lecture Notes)*. Nov. 2021.

A Decision Tree Algorithms [6]

Algorithm 1 GrowTree (D, F)

```
# Input: data D; set of features F
if Homogeneous(D) then
  return Label(D)
end if
S ← BestSplit(D, F)
split D into subsets D_i according to the literals in S
for each i do
  if D_i ≠ ∅ then
    T_i ← GrowTree(D_i, F)
  else
    T_i is a leaf labelled with Label(D)
  end if
end for
return a tree whose root is labelled with S and whose children are T_i
```

Algorithm 2 BestSplit (D, F)

```
# Input: data D; set of features F
I_min ← 1
for each fϵF do
  split D into subsets D_1, ..., D_l according to the values v_j ≤ f
  if Gini(D_1, ..., D_j) ≤ I_min then
    I_min ← Gini(D_1, ..., D_j)
    f_best ← f
  end if
end for
return f_best
```

B Manual Path Examples

index	y_test	y_pred	x												
			age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
104	0	0	57	1	2	128	229	0	0	150	0	0.4	1	1	3
258	1	1	54	0	1	132	288	1	0	159	1	0	2	1	2
97	0	1	67	1	0	120	237	0	1	71	0	1	1	0	2
196	1	0	65	1	0	120	177	0	1	140	0	0.4	2	0	3

```
|--- cp <= 0.50
|   |--- ca <= 0.50
|   |   |--- thal <= 2.50
|   |   |   |--- thal <= 1.50
|   |   |   |   |--- age <= 54.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- age >  54.50
|   |   |   |   |   |--- class: 1
|   |   |   |--- thal >  1.50
|   |   |   |   |--- thalach <= 145.50
|   |   |   |   |   |--- oldpeak <= 0.05
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- oldpeak >  0.05
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- thalach >  145.50
|   |   |   |   |   |--- class: 1
|   |   |--- thal >  2.50
|   |   |   |--- oldpeak <= 0.10
|   |   |   |   |--- trestbps <= 147.00
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- trestbps >  147.00
|   |   |   |   |   |--- class: 0
|   |   |   |--- oldpeak >  0.10
|   |   |   |   |--- class: 0
|   |--- ca >  0.50
|   |   |--- trestbps <= 109.00
|   |   |   |--- trestbps <= 103.00
|   |   |   |   |--- class: 0
|   |   |   |--- trestbps >  103.00
|   |   |   |   |--- class: 1
|   |   |--- trestbps >  109.00
|   |   |   |--- sex <= 0.50
|   |   |   |   |--- trestbps <= 132.00
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- trestbps >  132.00
|   |   |   |   |   |--- class: 0
|   |   |   |--- sex >  0.50
|   |   |   |   |--- thalach <= 107.00
|   |   |   |   |   |--- age <= 61.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- age >  61.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- thalach >  107.00
|   |   |   |   |   |--- class: 0
```

```
|--- cp >  0.50
|   |--- age <= 56.50
|   |   |--- oldpeak <= 1.95
|   |   |   |--- trestbps <= 160.00
|   |   |   |   |--- class: 1
|   |   |   |--- trestbps >  160.00
|   |   |   |   |--- class: 0
|   |   |--- oldpeak >  1.95
|   |   |   |--- thal <= 2.50
|   |   |   |   |--- class: 1
|   |   |   |--- thal >  2.50
|   |   |   |   |--- class: 0
|   |--- age >  56.50
|   |   |--- sex <= 0.50
|   |   |   |--- cp <= 1.50
|   |   |   |   |--- class: 1
|   |   |   |--- cp >  1.50
|   |   |   |   |--- class: 1
|   |   |--- sex >  0.50
|   |   |   |--- chol <= 249.50
|   |   |   |   |--- ca <= 0.50
|   |   |   |   |   |--- age <= 65.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- age >  65.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- ca >  0.50
|   |   |   |   |   |--- age <= 61.50
|   |   |   |   |   |   |--- thal <= 2.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- thal >  2.50
|   |   |   |   |   |   |   |--- chol <= 224.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- chol >  224.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- age >  61.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |--- chol >  249.50
|   |   |   |   |--- trestbps <= 169.00
|   |   |   |   |   |--- age <= 66.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- age >  66.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- trestbps >  169.00
|   |   |   |   |   |--- class: 0
```

Figure 4: Decision Tree Path Examples from Test Set

C Test Set Errors

index	y_test	y_pred	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	
44	0	1	50	1	2	140	233	0	1	163	0	0.6	1	1	3	
29	0	1	62	0	0	150	244	0	1	154	1	1.4	1	0	2	
196	1	0	65	1	0	120	177	0	1	140	0	0.4	2	0	3	
185	1	0	59	1	0	135	234	0	1	161	0	0.5	1	0	3	
37	0	1	46	1	2	150	231	0	1	147	0	3.6	1	0	2	
97	0	1	67	1	0	120	237	0	1	71	0	1	1	0	2	
50	0	1	35	1	0	126	282	0	0	156	1	0	2	0	3	

Table 1: Incorrect Predictions Made by Decision Tree in Test Set