A Quantitative Analysis of Explainable Artificial Intelligence Techniques as Applied to Machine Learning Models for Breast Cancer Classification

Author: Stephen Newman

Supervisor: Dr Shuaib Memon

Submitted to the University of York in partial fulfilment of the requirements

for the degree of MSc Computer Science with Artificial Intelligence

December 2022

Word count: 8,676

Executive Summary

In this research we apply two different binary classification approaches to a dataset as provided by [1] describes a collection of features obtained from digitised images of breast mass samples extracted via fine needle aspiration (FNA) [2]. The overarching purpose is to develop a pair of machine learning models capable of classifying if a given sample is benign or malignant in nature. The two approaches employed are that of the Optimal Sparse Decision Trees (OSDT) [3] and Random Forest Classifier [4]. These models were then compared against one another in terms of both accuracy and interpretability.

We shall show that, although the OSDT algorithm produced good result, the Random Forest Classifier outperformed in terms of accuracy but was far less interpretable when both were trained against the same data under a 10-fold cross validation process. The data was sourced from the UCI Machine Learning Repository [1] and has been utilised in accordance with the CC BY-NC-SA 4.0 [5] creative commons licence which permits the building upon the published material if the use is credited and for non-commercial purposes. Though the data pertains to medical samples acquired from human beings, care must be taken to avoid exposure of a patient's identity as part of this research. The dataset contains no Personally Identifiable Information (PII), as such, no mitigation is required.

I have received no funding from any party in relation to this research project. As such, in line with the University of York's Code of Ethics [6] I submit that this work meets the ethical requirements required as part of the MSc Computer Science with Artificial Intelligence programme.

Acknowledgements

During this assignment, my supervisor Dr Shuaib Memon has provided me with valuable insight and challenged me to develop as a student. I thank him for his advice and time, without which this would not have been realised. Additionally, this work would not have been possible without the support, encouragement, and understanding I received from my family and friends.

Contents

Figures

Equations

Tables

Introduction

Artificial Intelligence (AI) is being applied to an increasing number of facets of our lives [7]. From scenarios such as recommendation engines through to complex systems designed to drive vehicles on public roads, there seems to be no end to the scope and variety of the tasks that AI techniques are being applied to. It is natural that these scenarios exist on a spectrum between what are referred to as low stakes towards those that would be described as high stakes. For example, a private individual may have less interest in YouTube's recommendation engine providing appropriate and interesting content than they would be a system charged with detecting the presence of malignant cancers within tissue samples.

AI is a field of Computer Science and includes the sub-field of Machine Learning (ML) [8]. In many typical programming scenarios, the programmer is in possession of two elements, namely the rules to apply and the data over which to apply those rules. The concept of ML is that the computer is provided with the data and attempts to discover the rules through several mechanisms [9]. The varying approaches that can be applied to ML align with different sets of problems, the approach of supervised learning is applicable in scenarios where the computer has access to both the data and the correct outcome [10]. From these two inputs the goal is to train an ML model which can be used to predict the outcome for previously unseen, or novel, data. The more successful the model is at correctly predicting unseen data, the more generalised that model is said to be [11].

ML models can produce undesirable results or raise worrying questions. Recently, Twitter made use of an ML model which, when applied, exhibited a racial bias during the cropping of images [12]. An individual’s life chances could be extremely negatively impacted by the misclassification of a tissue sample as benign instead of malignant. In instances where an individual has been harmed or exposed to potential harm it is reasonable for that individual and applicable regulatory bodies to ask the question as to why this happened such that suitable action can be taken. Viewed from a different perspective, content recommendation engines should promote content in an ethical fashion. It would not be reasonable if content produced by white men in their early 20s would be recommended over other options based on the ethnic group of the creator. Developers of and researchers in AI systems may be able to build better systems and refine approaches if those systems have an ability to describe why model outputs have been determined. This need for explainability in systems has been known for many years, was acknowledged during the development of expert systems during the late 1970s and early 1980s [13], and within the field of AI is the focus of the sub-field of eXplainable Artificial Intelligence (XAI). These stakeholder classifications, described more fully by Preece, Harborne, Braines, Tomsett, and Chakraborty [13], possess differing priorities and therefore different requirements and expectations of XAI. This complexity shows a clear need for further research in this area.

Rudin [14] asserts that although much attention has been paid to the explainability of black box ML models that these efforts may ultimately harm the use of such models in society. For the widespread use of AI within our lives to be acceptable, it is necessary for society to largely trust those deployed models. The more critical a role the model performs, the higher the need for that model to earn and retain the trust of society. To this end Rudin urges, rather than continue to leverage black box techniques and then layer on additional mechanisms to extract explanations, that interpretable techniques be preferred and especially so in high stakes arenas. While there are multiple key issues listed regarding Explainable ML [14], the core issue to be focussed on is the first from their paper. Specifically, the assertion that “It is a myth that there is necessarily a trade-off between accuracy and interpretability”. The use of the word myth here is particularly interesting, the Oxford English Dictionary includes this definition [15]: “A widespread but untrue or erroneous story or belief; a widely held misconception; a misrepresentation of the truth.”. By asserting that the concept of a trade-off existing between accuracy and interpretability is widely held but ultimately false is a strong statement and one that appears to warrant some investigation.

Investigating such a question will not be simple, nor will any result obtained be wholly conclusive. A sensible starting point seems to be applying the algorithm devised by Rudin et. al. during their work on Optimal Sparse Decision Trees (OSDT) and comparing the ML model acquired against more established, but potentially less interpretable alternative approaches. Measuring each ML model against each other in terms of both accuracy and interpretability may aid in judging if a trade-off between those two metrics is, indeed, a myth. As such, this research will attempt to determine the following:

1) Can the OSDT computation technique developed by Hu, Rudin, and Seltzer be applied to build interpretable binary classification ML models?
2) How do such ML models compare in terms of accuracy versus competing ML models developed using an alternative classification technique?

3) How do such ML models compare in terms of interpretability versus competing ML models developed using an alternative classification technique?

As the OSDT technique developed in [16] can only be used to solve for binary classification problems, this work will necessitate the use of a dataset compatible with binary classification. This is to say that the model will be able to predict which one of two possible outcomes is most appropriate given the input data. There are several options available to develop a competing model as binary classification is such a deeply researched scenario and we shall discuss a few of these options within this paper.

Literature Review

While much research continues to be performed in the arena of XAI, there are some fundamentals problems which need to be met and overcome. Within [17] the point is made that interpretability and explainability are at times used interchangeably, but that explainability extends from interpretability. Suggesting that the goals of interpretability satisfy the Developer and Theorist stakeholder communities identified by [13], and explainability is for the benefit of the Ethicist and User communities defined in the same paper. For any reasonable conversation to take place, we need to agree upon what is meant by interpretable and explainable. This is acknowledged in [13] [14], and [18]. For the purposes of this research, we shall utilise the definition as presented in [14]. Specifically considering that an interpretable ML model is one that is constrained to a domain language such that an expert in that domain could find the description of the model understandable. The description provided by such a model would be clear to a domain expert and they would be able to determine if the mechanisms employed seem reasonable, valid, and ethical. Explainability will therefore be described as a post-hoc analysis of a decision to determine how such a decision was determined.

Some models are inherently more interpretable than others, largely this correlates with whether the model is a white box or a black box. A white box model, also described as being a transparent model, can have its inner workings inspected and reasoned about [19]. The definition of interpretability we have aligned with from [14] aligns with this property well and should satisfy the Ethicists [13] particularly as a given model can be inspected to determine if non-ethical features are being leveraged during a decision making process.

For example, race should not be considering during a bank loan application process, an interpretable model would expose this prejudicial feature consideration quickly. Conversely a black box or opaque model has inner workings which are not made available or are so complex as to be impractical for a human being to reason about. Deep neural networks can often make it difficult to understand why a particular decision has been made as the number and type of layers, the activation functions and weights being applied results in a large and complex function. A human being attempting to determine the “why” of a decision would find it hard to operate with this level of complexity. Significant effort has been spent developing mechanisms by which black box models can be made explainable.

An example of one of these techniques is permutation feature importance [20] where, during training time, the model's error rate is checked as a feature is permuted. If a given variable is found to be highly indicative of the model's decision, then it is likely that that variable is of more importance within the domain being modelled. However, as this determination is made during training time, when the correct decision is made, it is not possible to predict for all degrees and combinations of possible permutation. As such, this technique may lead to incorrect explanations when presented with novel data, as is likely to occur once the model is deployed and in active use.

Given a trained black box model defined as some function $f(x)$ and a mechanism is utilised to determine an interpretable (or at least more interpretable) companion explainer function $e(x)$ such that an explanation of the black box model can be made available. It is desirable that any explainer exhibits high fidelity [21], but if an explainer has anything less than 100% fidelity then it must, for some values of x yield a different output than the black box model function. This is referred to as the "two model" problem [22], once trust is lost in an explainer then the interested stakeholder or stakeholders would be less likely to rely upon it. It seems reasonable that the degree of tolerated divergence between a model and its explainer is related to the seriousness of the domain within which the model has been created.

The fidelity required of an explainer within a low stakes domain such as a content recommendation engine could be vastly different to that required within a high stakes domain i.e. that of a cancer diagnosis model. If a content consumer is exposed to recommendations from a poorly performing engine, then the consumer is likely to find said recommendations of little use. The danger of a cancer diagnosis model incorrectly labelling malignant masses as benign could have real lasting damage to the associated patient, conversely incorrectly identifying benign masses as malignant increases risks and costs associated with unnecessary medical treatment. These high stakes domains would likely require much higher fidelity in any explainer associated with a deployed model.

Given the problems associated with attempting to derive explainers with a suitable degree of fidelity, it seems likely that the industry should proceed as Rudin directs [14] and leverage naturally interpretable techniques rather than continue to leverage black box models especially if the domain is deemed to be high stakes in nature. However, there are several challenges against this point of view, firstly some organisations are in the business of the creation of and charging for the use of models they develop. If such an organisation were to provide a product which was trivial to reverse engineer, it may have a negative effect on their ability to succeed within their chosen market. Tooling such as that being developed and sold by IBM [23] is targeted at allowing organisations to leverage common patterns to provide explainers, thus continuing to realise the benefits provided by an opaque model. Additionally, some problems may not lend themselves to being solved in an interpretable manner, for example, reinforcement learning methods receive their data in steps, developing and later applying policies to determine what the next best action should be [24]. This typically yields a new set of data which goes through the same policy execution loop. While it is possible to leverage a rule list or decision tree for each grouping of policies, the number of these policies and complexity in navigating and selecting the appropriate policy could become problematic. If such a mechanism could be developed it could benefit not only the users and ethicists associated with a domain, but developers of such models could reduce the search space by removing options which lead to undesirable outcomes [25]. Considering the upcoming legal requirements being developed with AI in mind such as the Artificial Intelligence Act [26], it is likely the interest in and need for continued research into XAI will only increase.

A further point of contention within the research is that increasing interpretability lowers accuracy. Pintela et al, state that black box ML models which are less interpretable are often more accurate [19]. As discussed in the introduction, this claim is directly refuted by Rudin [14] and the research contained within this paper aims to determine if the downward pressure on accuracy as a feature of interpretable models is, as Rudin describes, a myth [14]. To make progress in this area it is necessary to determine how accuracy and interpretability are to be compared.

When a binary classifier is tested, there are four possible outcomes:

- The target was label α and the prediction was label α (True Positive)
- The target was label β and the prediction was label β (True Negative)
- The target was label β and the prediction was label α (False Positive)
- The target was label α and the prediction was label β (False Negative)

The testing outcomes can be visualised by means of a confusion matrix, as seen in Figure 1, this gives a quick at-a-glance mechanism by which an individual working with, or considering such an ML model to understand that ML model’s performance.

		Predicted Classification	
	Total Population = P + N	**Positive**	**Negative**
Actual Classification	**Positive**	**True Positive**	**False Negative**
	Negative	**False Positive**	**True Negative**

Figure 1 - Confusion Matrix

The term accuracy in the second research question is well defined in relation to the analysis of binary classifiers. Put simply, the accuracy of such a classifier is the number of times the classifier correctly classifies the input divided by the total number of classifications made, as shown in Equation 1. The two correct classifications are represented as the True Positive and the True Negative segments in the Figure 1. Classifiers which yield an accuracy value closer to 1 (100%) are preferable over those closer to 0 (0%).

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

Equation 1 - Accuracy

Making a comparison based on interpretability is less well defined, especially as there is a reliance upon the interpreter being an expert in the classifier's domain. In the mid-1950s, George A. Miller published a paper [27] which presented the idea that a human being could typically process 7±2 items of information at once. This rough heuristic, while not considering the capabilities of a domain expert may be enough to provide some notion as to how complex a classifier may or may not be.

As Hu, Rudin, and Seltzer have made the implementation of their OSDT [16] publicly available [3] that will be the implementation used to address the first research question. Binary classifiers have been subject to much analysis and their relative performance characteristics have been a key element of that research [28].

Paper	Author(s)	Pros	Cons
Optimal Sparse Decision Trees (2019)	X. Hu, C. Rudin and M. Seltzer	• Introduces and uses OSDT	• Simple datasets • Analysis of interpretability is lacking • Compares only with a single other algorithm (CART)
A Comparative Study on Various Binary Classification Algorithms and their Improved Variant for Optimal Performance (2020)	V. Bahel, S. Pillai and M. Malhotra	• Complex datasets • Numerous classification processes are considered and compared	• OSDT is not included • Analysis of interpretability is lacking

Table 1 - Comparison of Papers

During Rudin et al's production and proving of their OSDT algorithm [16], the datasets used were of reasonably low complexity, for example the Compas dataset used to predict recidivism is comprised of 12 attributes when prepared for use by the OSDT algorithm. It may follow that constructing interpretable ML models from a small selection of attributes provides a natural advantage to a decision tree attempting to make accurate predictions with as few nodes as possible. Similarly, as Bahel, Pillai, and Malhotra [28] compared binary classification algorithms they did not make use of any of the datasets used by Rudin et al. As [28] noted that the Breast Cancer Wisconsin Diagnostic [1] was found to be the best performing dataset and as it has a higher attribute count than any of the datasets used in [25] we shall be leveraging the Breast Cancer Wisconsin Diagnostic dataset in order to build upon both research papers. The merits and demerits of each paper are summarised in Table 1.

Motivation

The motivation behind this research is to gain greater understanding as to the tools and techniques available to AI practitioners in connection with XAI. If we consider the recent Payment Protection Insurance (PPI) mis-selling scandal when the regulator, the Financial Services Authority (FSA), found that self-regulation of the market had failed especially regarding suitability checks [29]. As the use of AI continues to expand, decisions models make may well form part of a future scandal. This indicates that the need to be able to document explanations captured at the point of decision making will increase accordingly. Some of the arenas of our lives that AI will be utilised will be of little consequence to us, others will be more important, and when an AI makes a mistake, it is reasonable as a society to seek to understand why that mistake was made. This mistake could result in a poorly managed investment portfolio, a misdiagnosis of a disease, or an immediately catastrophic car accident with loss of life and life-altering injuries being a definite possibility. When this happens, how we do answer the question – why? Due to the variety of applications and the complexity of the decisions being made, applicable techniques and mechanism are likely to also be varied in terms of both approach and complexity.

Methodology

To compare the performance of different ML techniques it is feasible to use those techniques and the same set of data to train and test competing ML models, by contrasting the performance characteristics of the produced ML models we can begin to understand which approach to employ. This is similar to the research done by Bahel, Pillai, and Malhotra [28] where a number of different algorithms were used, and the Random Forest Classifier was found to perform well in this context. To answer the first research question, we shall first attempt to build a model using the OSDT implementation that is available on GitHub [3]. This classifier has been selected as it has previously been shown to perform well against both Classification and Regression Trees (CART) and BinOCT classifiers [30]. The appeal of OSDT in relation with this work is that traditional decision trees can be prone to overfitting their training data [31] and generating large trees. Most decision trees operate in a greedy fashion, attempting to make the split at each node which provides the best split based on the calculation being employed. Popular choices are Gini Impurity and Entropy. Regardless of the choice made, if a split is made early in tree formation that later turns out to be a bad split, then additional splits must be performed in order to undo this “bad” split. Without backtracking, there isn’t a mechanism to detect and undo this costly split and the end result is that the tree becomes larger than it needs to be, in turn making it harder to reason about i.e. interpret.

Gini Impurity is a number between 0 and 0.5 which shows the likelihood that new data would be considered incorrectly classified based on the distribution in the training dataset. For a dataset *D* containing *k* classes with the probability of samples belonging to the class *i* being denoted as p_i then Gini Impurity is defined in Equation 2.

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

Equation 2 - Gini Impurity

Conversely Entropy is a measure of disorder, which for the same dataset *D* containing *k* classes with the probability of samples belonging to class *i* being denoted as p_i then Entropy is defined in Equation 3.

$$Entropy(D) = \sum_{i=1}^{k} -p_i \log_2 p_i$$

Equation 3 - Entropy

As the tree is being constructed, at each potential split and according to the metric being employed, the sub-set of data under consideration is interrogated and the best split that leads to the lowest Gini Impurity or lowest Entropy depending on the metric being used. As there is typically no provision for backtracking, traditional Decision Tree algorithms operating in this greedy fashion can miss the realisation the split *q* followed by split *r* leads to a higher combined reduction in Gini Impurity/Entropy, i.e. information gain, than split *r* followed by split *q*. To undo this suboptimal split, further splits must be made at lower levels to reverse this decision.

The mechanism described in [16] makes available several hyperparameters which can be leveraged to tune the mechanisms employed. The property *lamb* of the *bbound* function is used as part of the mechanism which causes larger trees to be penalised. As described in section 3.1 Objective Function of [16], as a tree with H_d leaves is multiplied by this factor by selecting smaller values, trees of more complexity are allowed to form. The experiments included in the original paper make use of several different vectors of lamb values, the collection that is most common is given as *lambs1 = [0.1, 0.05, 0.025, 0.01, 0.005, 0.0025]*, we shall apply this same set of values while executing our experiments. The other hyperparameter of interest is the *prior_metric* which determines the scheduling policy applied to the priority queue controlling the exploration of the search space. The *curiosity* scheduling policy was found to provide the best results within the original paper so we shall apply that same policy for our initial experiments. This hyperparameters have been selected as, in the context of the lambda value collection it is the most common set of values that were applied in the original research, and the curiosity scheduling policy yielded the best results. By attempting to keep in line with the original research, it is anticipated that the results will be more directly comparable.

In order to have a competition between different models a second model shall be required. During the work contained within [28] several techniques suitable for binary classification workloads where investigated and it was found that the Random Forest classifier won out in terms of performance. As such, our second model will be constructed applying this technique. Leo Breiman introduced [4] Random Forests in 2001, instead of training a single decision tree against the entire dataset the approach within this technique is to train many smaller decision trees on different subsets of the data. Each of the smaller decision trees are then expected to vote for a class and the class with the most votes, in the case of classification problems, is returned as the prediction. During formation of the decision trees within the forest several steps take place to manipulate the training data seen by each tree. Through a combination of sampling (selecting a subset of records) and feature bagging (selecting a subset of the features) the likelihood of the decisions trees within the forest overfitting is greatly reduced as each individual tree has experienced a sub-set of a sub-set of the full training data. This reduction in available data is offset by training multiple decision trees within the forest, and the implementation that shall be used during this work is Random Tree Classifier [32] which has been included as part of the Scikit-Learn platform. By default, the Random Forest Classifier makes use of a forest of 100 trees and it shall be this default that is applied, in future research it may be interesting to perform the same experiments with forests of differing sizes, but for this first step we shall abide by the default values.

The datasets being used during the testing of the various classifiers during the production of [28] is available on the UCI Machine Learning Repository, specifically the Breast Cancer Wisconsin (Diagnostic) Data Set [33]. We shall be using the same dataset for several reasons. It qualifies as being related to a high stakes domain and as such merits being supported by either an interpretable ML model or an opaque model in combination with a high-fidelity explainer. The dataset contains features obtained from a digitised image for a fine needle aspirate (FNA) of a breast mass [2]. Each record contains an id number, a diagnosis (either M – Malignant or B – Benign) as there are two classes this data is suitable for use with a binary classifier. The next thirty attributes then describe several characteristics of cell nuclei, these attributes being real values such as radius, symmetry, and smoothness. The values stored are the mean, standard error, and worst or largest as computed for each image. The groups of ten such attributes are listed in Table 2 and the mechanisms by which are they are computed are given in [34].

Name	Description
Radius	Mean of distances from center to points on the perimeter
Texture	Standard deviation of gray-scale values
Perimeter	
Area	
Smoothness	Local variation in radius lengths
Compactness	Perimeter^2 / Area – 1.0
Concavity	Severity of concave portions of the contour
Concave Points	Number of concave portions of the contour
Symmetry	
Fractal Dimension	“coastline approximation” – 1

Table 2 - Real-Valued Features from Dataset

To prepare the data for use by the algorithm, the ID number attribute shall be removed as this is metadata about the record and should not be presented to the supervised learning mechanism. The implementation of the OSDT algorithm has a limitation in that all features are constrained to be either a 0 (zero) or a 1 (one). The dataset being used is made up of 30 data elements, each of which is represented as continuous data. Continuous data is data which falls in a continuous sequence, including any value within the applicable range. While the class identifiers in the dataset ‘B’ and ‘M’ can simply be replaced with a 0 or 1 respectively, in order to reduce a continuous range into membership of such a constrained set {0, 1} a procedure of unsupervised discretization will be applied utilising a fixed width strategy.

The data will be broken into 10 folds for the purposes of *k*-fold cross validation. *k*-fold cross validation is a useful technique which can be used to help test the performance of a given modelling technique. By taking the original dataset and generating *k* sets of random training and test data from it, it is possible to embark upon *k* training and analysis runs without needing *k* sets of complete data. Upon splitting the previous vector of lamb values will be iterated over and the best performing tree for each lamb value and fold will be stored along with its evaluation score for later analysis. This will yield 10 examples of OSDT, answering the first research question. This also mirrors an aspect of the research by Bahel, Pillai, and Malhotra [28] where 10-fold was the highest fold degree utilised. Additional research could be performed to determine if and how the number of folds utilised also affects the performance of trained models.

The same 10-fold cross validation technique will be applied during the training and testing of the Random Forest Classifier algorithm, after which the results can be compared in terms of accuracy as shown previously. This will allow us to answer the second research question.

To answer the third research question, it will be necessary to analyse the resultant decision trees yielded from the OSDT approach and select the tree with the best accuracy. Once the tree has been identified, the tree shall be diagrammed to allow for visual inspection. Special attention will be paid to the number of features upon which the tree operates, and this shall be compared against the 7±2 heuristic developed by Miller in [27]. Determining the interpretability of the Random Forest is likely to be more challenging, while the most accurate forest may be obtainable, the tasks of visualising the 100 trees that make up that forest will not lead to an interpretable result. This is because 100 is certainly above the 7±2 threshold discussed. Instead, the model shall be inspected for permutation feature importance in order to provide an explainer, this capability is made possible by the Scikit-Learn package [35]. The goal of this will be to attempt to describe which of the features are most likely to affect accuracy as they are changed. The features which offer the greatest accuracy shift will be identified as those features which have the most importance as determined by the Random Forest Classifier.

Results

These experiments have been run utilising Docker Desktop 4.15.0 on an AMD Ryzen Threadripper 3970X 32-Core Processor @3.70 GHz with 128GB RAM running Windows 11 Pro 22H2 with 24 CPUs and 32GB of RAM allocated to the Docker platform.

Leveraging the OSDT algorithm proved to be significantly more challenging than originally expected. The process of training the ML models took a significant amount of time and the data structures used were more complex to understand and work with than those presented by the Scikit-Learn provided mechanisms. The OSDT algorithm has a time limit after which the improvement searching process is cut short and the algorithm returns. This is set to 1,800 seconds in the example test_accuracy.py code file [3], which equates to 30 minutes. During the model training process with the lambda value of 0.025, this time limit started to be encountered so we adjusted the lambda collection to not extend beyond that value to prevent that timeout from potentially affecting the results. As such we have access to the ten folds being used for fitting and training purposes across each of the three largest lambda values {0.1, 0.05, 0.025}.

The accuracy for both the training and testing phases are shown in Table 3.

Fold	Lambda	Training Accuracy	Testing Accuracy
0	0.1	0.91015625	0.666666667
1	0.1	0.892578125	0.824561404
2	0.1	0.884765625	0.807017544
3	0.1	0.892578125	0.824561404
4	0.1	0.880859375	0.929824561
5	0.1	0.876953125	0.964912281
6	0.1	0.880859375	0.929824561
7	0.1	0.87890625	0.947368421
8	0.1	0.888671875	0.771929825
9	0.1	0.8791423	0.946428571
0	0.05	0.91015625	0.666666667
1	0.05	0.892578125	0.824561404
2	0.05	0.884765625	0.807017544
3	0.05	0.892578125	0.824561404
4	0.05	0.880859375	0.929824561
5	0.05	0.876953125	0.964912281
6	0.05	0.880859375	0.929824561
7	0.05	0.87890625	0.947368421
8	0.05	0.888671875	0.771929825
9	0.05	0.8791423	0.946428571
0	0.025	0.935546875	0.754385965
1	0.025	0.923828125	0.859649123
2	0.025	0.923828125	0.859649123
3	0.025	0.921875	0.859649123
4	0.025	0.919921875	0.894736842
5	0.025	0.91796875	0.929824561
6	0.025	0.919921875	0.912280702
7	0.025	0.916015625	0.947368421
8	0.025	0.923828125	0.877192982
9	0.025	0.914230019	0.964285714

Table 3 - Optimal Sparse Decision Tree Classifier Accuracy by Fold

Applying the Random Forest Classification algorithm as made available by the Scikit-Learn environment was a much easier experience than making use of the OSDT algorithm by Rudin et al. This is to be expected as Scikit-Learn is an established suite of ML tools for the Python environment.

The accuracy for both the training and testing phases are shown in Table 4.

Fold	Training Accuracy	Testing Accuracy
0	0.97265625	0.859649123
1	0.96875	0.894736842
2	0.966796875	0.912280702
3	0.966796875	0.929824561
4	0.966796875	0.98245614
5	0.96875	0.98245614
6	0.96875	0.929824561
7	0.962890625	0.964912281
8	0.966796875	0.964912281
9	0.968810916	0.982142857

Table 4 - Random Forest Classifier Accuracy by Fold

The resultant output from executing the experiments along with the directions as to how to perform such executions is include within the accompanying artefact, the layout of which is described in Appendix A – Contents of artefact Directory.

Analysis

During the data preparation stage, to make the input acceptable to the OSDT algorithm it was necessary to discretize the data, this has had two main effects upon the data being presented to both algorithms. Firstly, the number of features has increased by a factor of five, this is because the arity variable, which is used to control the number of bins into which the data is discretized, is set to five in the Data Preparation notebook. The second impact is that the nature of the discretization applied was fixed width rather than equal frequency. While fixed width was selected to prevent identical values from being allocated to different bins, this could leave the dataset with empty bins making them irrelevant. This may present the Random Forest algorithm more of a challenge as some of the decision trees making up the forest may operate on attributes containing no data as each tree is exposed to a random subset of those attributes. However, upon analysis the prepared data the following summary can be formed, of note is that in each of the training and testing datasets there is at least one record with a 1 for each of the features, that is to say, none of the features are unrepresented in any of the test files.

Fold	Training			Testing		
	Unused Features	Benign Instances	Malignant Instances	Unused Features	Benign Instances	Malignant Instances
0	0	346 (68%)	166 (32%)	0	11 (19%)	46 (81%)
1	0	322 (63%)	190 (37%)	0	35 (61%)	22 (39%)
2	0	321 (63%)	191 (37%)	0	36 (63%)	21 (37%)
3	0	328 (64%)	184 (36%)	0	29 (51%)	28 (49%)
4	0	328 (64%)	184 (36%)	0	29 (51%)	28 (49%)
5	0	312 (61%)	200 (39%)	0	45 (79%)	12 (21%)
6	0	316 (62%)	196 (38%)	0	41 (72%)	16 (28%)
7	0	313 (61%)	199 (39%)	0	44 (77%)	13 (23%)
8	0	313 (61%)	199 (39%)	0	44 (77%)	13 (23%)
9	0	314 (61%)	199 (39%)	0	43 (77%)	13 (23%)

Table 5 - Prepared Data Summary

This has meant that there is an imbalance between the classes present between the training and testing datasets for each fold. In the example of fold 0, this imbalance is quite marked, the training dataset contains a heavy skew towards benign records, whereas the testing dataset exhibits an even stronger skew in the opposite direction. While this is the most extreme example, each of the folds show the same potential issue to varying degrees, as shown in the table above.

The results in Table 3 are split according to their respective lambda value. The lambda property controls the curiosity of the algorithm, the lower the value the more likely a sub-optimal (at first glance) split is allowed to be investigated. To understand the effect this has on the accuracy we can plot the box chart in Figure 2.

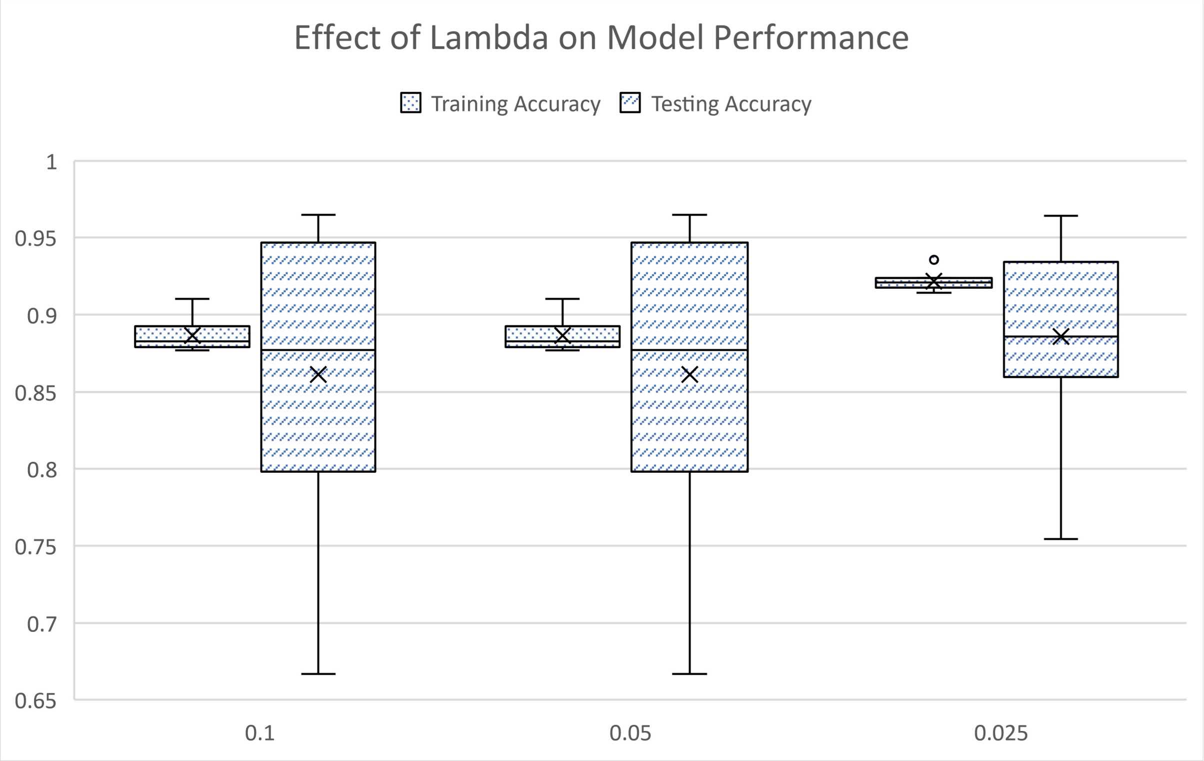

Figure 2 - Effect of Lambda on Model Performance

There are a few interesting points in this data, the first is that the training and testing performance for both 0.1 and 0.05 lambda values are identical. We would need to perform more analysis to understand why but it may be that the curiosity approach is not sufficiently activated until lower values are utilised. It would be interesting to extend the algorithm time limit to see if this trend continues as the lambda value is reduced further.

Once the lambda value is lowered to 0.025, we do see the performance of the generated models increase in the aggregate though the best performing model is from the 0.1/0.05 lambda set. The model trained on fold 5 achieved a testing accuracy of 96.49%, the best performing model from the 0.025 lambda set achieved 96.43%. The mean performance of the 0.025 lambda set is over 2 percentage points above the 0.1/0.05 lambda set at 88.59, and, as is made clear by the whiskers in the previous figure, the standard deviation is much improved, down to 0.0566 (to four decimal places) from 0.0929 (to four decimal places).

The results in Table 4 show us that the Random Forest Classifier is consistent at fitting a set of decision trees at training time, this is demonstrated by the training accuracy across all folds having a mean of 0.9678 (to four decimal places) and standard deviation of 0.0024 (to four decimal places). Such a low standard deviation shows that the population is very close to the mean i.e., the whole population of data points is closely packed around the mean. This can also be visualised from Figure 3 showing the line showing the training accuracy in per fold being reasonably straight between 0.96 and 0.98 with little fluctuation.

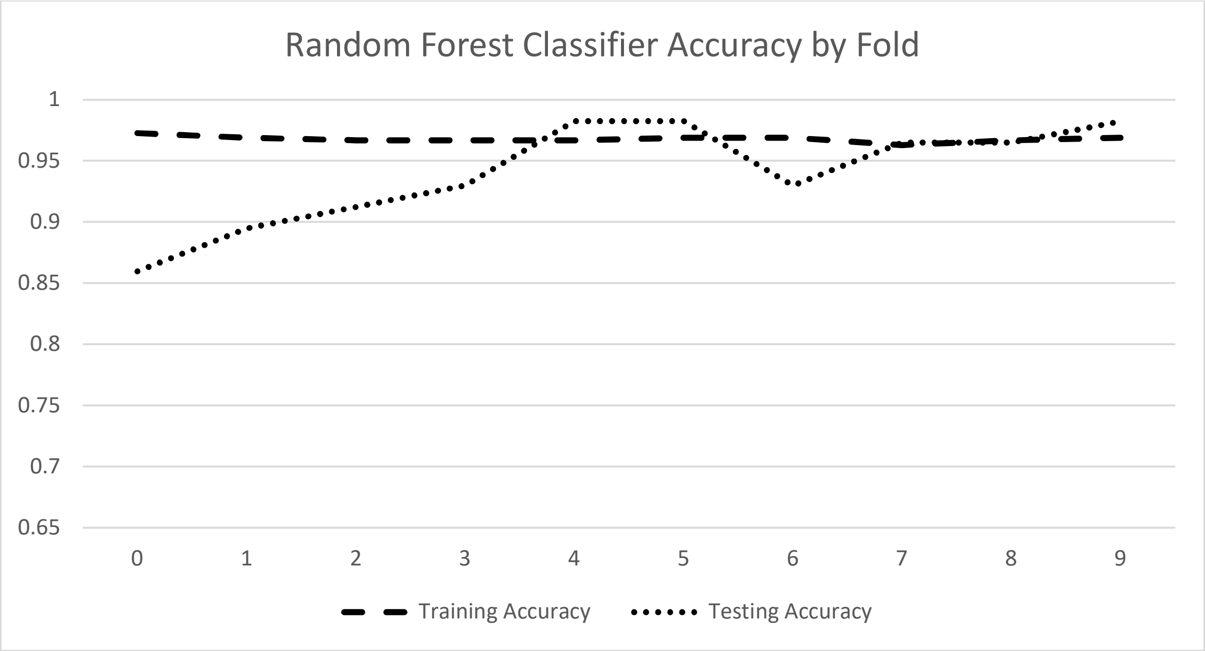

Figure 3 - Random Forest Classifier Accuracy by Fold

The testing accuracy varies to a much greater extent with only two classifiers operating at similar accuracies during both training and testing, these being those produced in association with folds 7 and 8. Three classifiers demonstration better performance, those trained on folds 4, 5, and 9. With the other classifiers performing between 2 and 10 percentage points less well during testing than during training.

Comparing the accuracy by fold of the Random Forest Classifier against the OSDT classifiers obtained we can see the difference between the candidate classifiers.

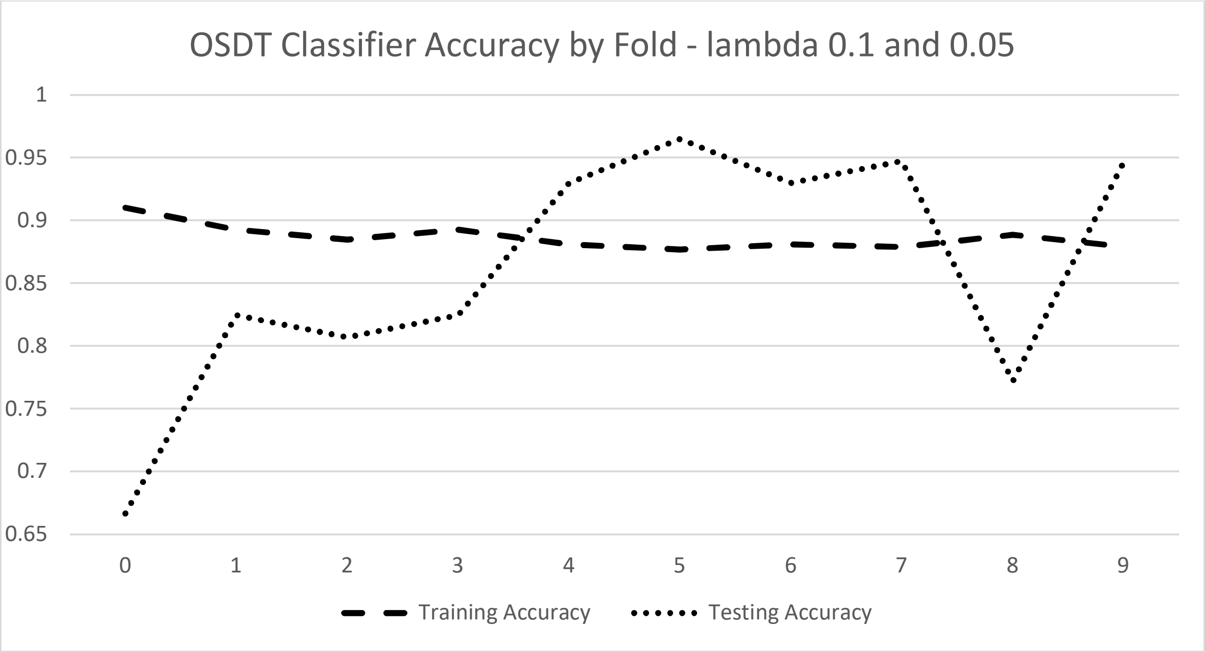

Figure 4 - OSDT Classifier Accuracy by Fold - lambda 0.1 and 0.05

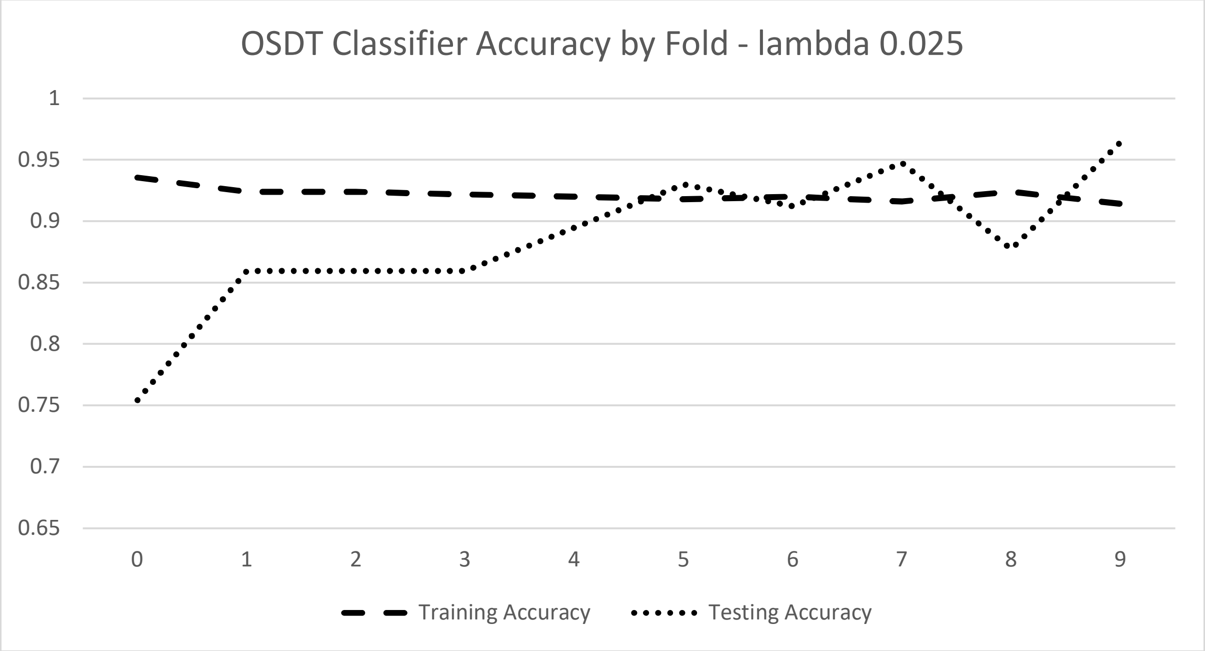

Figure 5 - OSDT Classifier Accuracy by Fold - lambda 0.025

By comparing Figure 3, Figure 4, and Figure 5, it is visually apparent that the OSDT classifiers underperform when compared to the Random Forest classifier. This is apparent in less accurate performances at both the training and testing phases and with the testing accuracy especially being more erratic than that produced by the

Random Forest classifier. This is further brought home by Table 6 where we examine the mean of each classifier and identify the accuracy of the best performer (in terms of accuracy) between them at both the training and testing phases.

Classifier	Training Mean	Training Max	Testing Mean	Testing Max
OSDT @ 0.1 / 0.05	0.886547043	0.91015625	0.861309524	0.964912281
OSDT @ 0.025	0.921696439	0.935546875	0.885902256	0.964285714
Random Forest	0.967779529	0.97265625	0.940319549	0.98245614

Table 6 - Accuracy Comparison

This data shows that the best performing Random Forest classifier is over 3.5 percentage points more accurate that the closest OSDT classifier at training time, but this advantage narrows to just under 2 percentage points during testing.

An interesting phenomenon is that all the classifiers exhibit poor generalisation when trained and tested with the folds {0, 1, 2, 3}, it is possible that the data within these particular datasets is split in such a way that may prove more difficult to detect a suitable classifier. It would be interesting to see if the results could be improved with a lower lambda value being made available to the OSDT algorithm, this would necessitate increasing the timelimit to allow the algorithm to complete. The Random Forest Classifier has a maximum depth of 5 which may well be restricting the ability for the data to be fully interrogated by each forest member.

One of the key benefits espoused by the researchers developing the OSDT Classifier is that it results in accurate and optimal, or near-optimal decision trees. The algorithm itself outputs details of the decision tree generated but the output isn’t as readily accessible as output from more established ML algorithms.

The output produced by the algorithm, in the case where a decision tree has been produced which is more accurate than the baseline decision tree classifier, includes the internal order the attributes are presented in, a set of leaf decision point paths, and a prediction for each leaf node. As we consider the output for the best performing, in terms of accuracy, model produced through this algorithm when the lambda value is set to 0.025, the leaf node paths are described in Table 7. This model has been selected as a demonstration as to how interpretable models produced by the OSDT algorithm ad models produced at 0.1 and 0.5 lambda were solvable with only two leaf nodes. The extra complexity caused by the decision tree have a depth greater than one allows us to discuss the mechanism more fully and as such provide more value to researcher choosing to build upon this research.

Leaf Node Path	Prediction
(1,)	0 (Benign)
(-17, -1)	1 (Malignant)
(-1, 17)	0 (Benign)

Table 7 - Leaf Node Paths

Each of these leaf node paths describe an unordered description of the decision points made between the root node and a given leaf node in the decision tree. The attribute under consideration in the node can be located by utilising this number in connection with the output labelled as ‘the order of x's columns’. For this model, this collection is given as:

[35, 115, 30, 15, 130, 116, 110, 102, 12, 138, 112, 2, 37, 25, 125, 100, 136, 31, 17, 50, 85, 1, 132, 7, 51, 32, 60, 11, 131, 27, 135, 16, 107, 86, 61, 3, 10, 13, 5, 117, 145, 0, 127, 103, 126, 139, 65, 142, 105, 75, 137, 123, 66, 121, 140, 113, 38, 22, 76, 146, 36, 21, 106, 122, 120, 143, 42, 26, 128, 28, 40, 41, 20, 33, 147, 18, 104, 108, 118, 39, 43, 87, 91, 23, 4, 114, 62, 14, 101, 52, 29, 133, 93, 129, 109, 72, 34, 19, 6, 77, 44, 90, 64, 144, 67, 69, 54, 47, 81, 124, 149, 74, 79, 119, 94, 80, 92, 46, 57, 98, 58, 9, 24, 88, 49, 111, 84, 89, 83, 99, 59, 56, 8, 71, 95, 97, 78, 48, 148, 134, 141, 96, 55, 70, 73, 45, 82, 53, 68, 63]

It should be noted that the lookup to perform here should be done in a manner that treats the collection as be 1-based, not 0-based. For example, if the node path contained the value 5, it would refer to a node containing a test being performed against the 130th attribute in the record, not the 116th. To then understand which attribute that is, a 0-based lookup should be performed against the data's attribute set. Specifically in this case, this would resolve to the 'worst concavity 0.0-0.2504' attribute. For the highest performing OSDT model, the leaf node paths involved the 1st and 17th indexed values, these resolve to the 35th and 136th attribute in the data set respectively. For clarity these are the attributes labelled 'mean concavity 0.34144-0.4268' and 'worst concave points 0.0-0.05819'.

Once the attributes being considered in each of the nodes, the paths need to be arranged such that a binary tree is formed. The sign identifier indicates that the path to the leaf is based on the indicated attribute for the record under test being set to 0 when the sign identifier is negative and 1 when the sign identifier is positive. Following this process, we can describe the leaf nodes output by the best performing OSDT classifier with lambda set to 0.025 shown in Table 8.

Leaf Node Path	**Readable Decision Tree Path**	**Prediction**
(1,)	mean concavity 0.34144-0.4268 is 1	0 (Benign)
(-17, -1)	worst concave points 0.0-0.05819 is 0 and mean concavity 0.34144-0.4268 is 0	1 (Malignant)
(-1, 17)	mean concavity 0.34144-0.4268 is 0 and worst concave points 0.0-0.05819 is 1	0 (Benign)

Table 8 - Readable Decision Tree Paths

By analysing the leaf node paths it is possible for us to construct a decision tree in diagrammatic form.

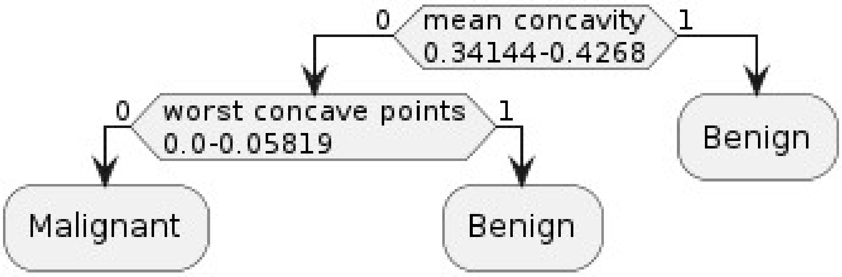

Figure 6 - Decision Tree for Highest OSDT Performer with lambda set to 0.025

Once the algorithm’s output has been deciphered and its findings presented in the pictorial format shown in Figure 6 it is possible to codify it within an Excel spreadsheet into which the test data for fold 9 has been loaded and then analyse that output presented in the confusion matrix in Figure 7.

		Predicted Classification	
	Total Population = P + N	**Malignant (Positive)**	**Benign (Negative)**
Actual Classification	**Malignant (Positive)**	**12**	**1**
	Benign (Negative)	**1**	**42**

Figure 7 - Confusion Matrix for Highest OSDT Performer with lambda set to 0.025

This confusion matrix further explains the performance of this model, there are two incorrect classifications. Both of which are problematic in this domain but for different reasons and to different extremes. In the case of a false positive a benign sample is classified as malignant; this could cause the patient to undergo additional procedures and at the very least cause an elevated level of stress to that individual. However, a false negative, where a malignant sample has been classified as benign could allow the disease to worsen with potentially disastrous outcomes for the patient.

Underpinning these results is a model which can be easily interpreted, of the 150 attributes presented to the algorithm a very successful model has been produced which extracts information from just two of those attributes. This model can be easily understood by domain experts as to the data points being examined and could also serve as a starting point for model developers, employing a different discretization strategy or generating a larger number of bins could help improve the sensitivity of the model. Sensitivity is the measure as to how well a model identifies instances of the positive class i.e. sample which are malignant, it can be calculated by the Equation 4.

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Equation 4 - Sensitivity

While subjecting patients to the additional trauma and uncertainty a false positive classification would bring is not ideal, it is possible, in this domain, that a false negative is deemed more problematic for the patient. This model's sensitivity at 92.31% (to two decimal places) is commendable but given the seriousness of the disease, would benefit from more research being undertaking to derive better performing models.

The Random Forest Classifier is comprised of 100 decisions trees, and those decision trees each have a maximum depth of 5, rather than the depth of 2 shown in the previous decision tree. The depth of a node in a *k*-ary tree is the number of edges between the tree's root node and the node in question. Figure 8 shows the 16th (note this is 0-based) tree from the 100 decision trees from the best performing Random Forest Classifier model, note that the ranges described by the data points in the decision nodes have been truncated for legibility reasons.

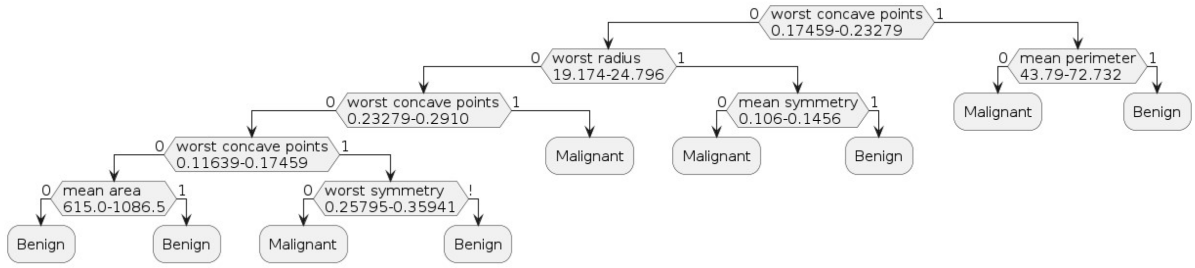

Figure 8 - 16th Decision Tree from the best performing Random Forest Classifier

This decision tree is clearly more complex than the tree determined by the OSDT algorithm, comprising of nine leaf nodes rather than three and considering eight different attributes rather than the two previously discussed. While this decision tree is still interpretable, we should remember that the Random Forest is comprised of one hundred such trees, each of which effectively votes for a classification and then the most voted for classification is then output by the forest as the eventual prediction. It is this combination of decision tree outputs that makes this algorithm less naturally interpretable than the OSDT algorithm. The Scikit-Learn package makes it possible to derive the permutation importance of the attributes within the dataset, after applying this mechanism and outputting the resultant diagram, it is shown that shuffling the values within the attribute 'mean perimeter 72.732-101.674' has the greatest effect on the model's error rate. But it is important to note that it is not clear as to why this is the case and if the permutation importance for each of the Random Forest is investigated, there are a great many attributes which appear to not move the model's error rate as they are shuffled. This may allow model designers to consider removing those attributes from the dataset, but this would require further research to determine the effect of such a change.

Conclusion

It has been shown that it is indeed possible to leverage the work by Hu, Rudin, and Seltzer presented in [16] to produce an interpretable binary classification ML model with this particular dataset. There are challenges with using this algorithm in its current form, these challenges affect the preparation required of the data being analysed and the effect this could have on the model performance.

The dataset utilised in this research was comprised of an identifier, which was excluded, a classification target, and 30 attributes containing continuous data. These attributes were aggregations across 10 attributes covering the mean, standard error, and "worst" of those attributes. The OSDT algorithm is limited to being able to operate on attributes containing only values from the set {0, 1}. While the classification target could easily be reduced to this set, the 30 continuous data points were more problematic, and the handling of those data points could very well affect the algorithm's performance in terms of accuracy and in terms of processing time.

Accuracy could be affected by the incorrect placement of bin terminators within the dataset. While performing discretization with a consistent bin width does prevent the issue of instances with the same value being placed into different bins, it does potentially ignore the underlying distribution of each attribute. However, care should be taken when aligning bins to the training dataset as it may promote overfitting to that dataset.

Processing time was found to be problematic while executing the OSDT algorithm leading to the smaller lambda values being removed from the execution plan. This may have been caused, in part, by the increased attribute count from 30 to 150 following the discretization process. To understand the impact on execution time that this number of attributes present will require additional research.

It has also been possible to compare the accuracy of OSDT ML models against those generated by the Random Forest algorithm. While one of the OSDT ML models did come extremely close to the performance, in terms of accuracy, as the best performing Random Forest classification model it has not been possible to comprehensively prove or disprove Rudin's assertion that "It is a myth that there is necessarily a trade-off between accuracy and interpretability" it has been impressive how well such a simple model as obtained from the OSDT algorithm performs against Random Forest ML models. This could mean that this particular binary classification problem, that being the classification of samples performed via fine needle aspiration, is fairly simple to solve and that applying the Random Forest approach is not strictly necessary in this case. Of course, when operating in such a high stakes domain as cancer diagnosis, perhaps skewing towards the most accurate model available is the reasonable course of action to take. This will depend on the risk appetite of misclassifications being weighed against any increased ability to provide reasoning behind any classification made.

As the OSDT algorithm is not an approach which is developed to the same level of polish as the Random Forest algorithm included as part of Scikit-Learn, it is understandable that extracting the details of the model is more involved than simply rendering a decision tree to a png file. Once the output of the learning process is interpreted and presented in a diagrammatic form, the simplicity offered by the models generated by the OSDT algorithm make them for more interpretable than those produced by the Random Forest approach.

If the OSDT algorithm could be further enhanced to output data structures natively supported by Scikit-Learn it would go a long way to removing this impediment and further increase OSDT's interpretability over Random Forest. Specifically, if it would be possible to have decision trees be automatically describable as they are for the included Decision Tree Classifier, that would allow a much quicker production of an interpretable view of the model's processing.

Limitations and Future Work

The results obtained during this research have highlighted the need to perform further experiments focussing on the effects different data preparation techniques have upon the performance, in terms of accuracy, of the OSDT algorithm. This could include different discretization techniques, data segregation techniques, and a comparative study as to how the application of these techniques affect the algorithm's performance in terms of accuracy. It was also not possible to complete the experiments using the smaller values for the lambda hyperparameter, it is possible that by allowing more time for the experiments to complete the resultant decision tree could exhibit greater performance. With the results obtained herein, it is not possible for us to conclude that the accuracy vs interpretability trade-off is indeed a myth or not.

References

[1] UC Irvine Machine Learning Repository, “UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set,” UC Irvine Machine Learning Repository, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29. [Accessed 25 06 2022].

[2] American Cancer Society, “Fine Needle Aspiration (FNA) of the Breast,” American Cancer Society, [Online]. Available: https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/breast-biopsy/fine-needle-aspiration-biopsy-of-the-breast.html. [Accessed 25 06 2022].

[3] X. Hu and M. Seltzer, “xiyanghu/OSDT: Optimal Sparse Decision Trees (OSDT),” [Online]. Available: https://github.com/xiyanghu/OSDT. [Accessed 25 06 2022].

[4] L. Breiman, “Random Forests,” *Machine Learning,* vol. 45, pp. 5-32, 2021.

[5] Creative Commons, “Creative Commons — Attribution-NonCommercial-ShareAlike 4.0 International — CC BY-NC-SA 4.0,” Creative Commons, [Online]. Available: https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode. [Accessed 25 06 2022].

[6] University of York, “Code of practice on ethics - Staff home, University of York,” University of York, [Online]. Available: https://www.york.ac.uk/staff/research/governance/research-policies/ethics-code/. [Accessed 25 06 2022].

[7] F. Doshi-Velez and K. Been, “Towards a rigorous science of interpretable machine learning,” 02 03 2017. [Online]. Available: https://arxiv.org/abs/1702.08608. [Accessed 11 12 2022].

[8] S. Russell and P. Norvig, “What is AI?,” in *Artificial Intelligence A Modern Approach Third Edition*, Harlow, Pearson, 2016, pp. 1-3.

[9] L. Moroney, “From Programming to Learning,” in *AI and Machine Learning for Coders - A Programmer's Guide to Artificial Intelligence*, Sebastopol, O'Reilly, 2020, p. 5.

[10] S. Russell and P. Norvig, “Supervised Learning,” in *Artificial Intelligence A Modern Approach Third Edition*, Harlow, Pearson, 2016, pp. 695-697.

[11] A. Géron, “Overfitting the Training Data,” in *Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow*, Sebastopol, O'Reilly, 2019, pp. 27-29.

[12] BBC, “Twitter finds racial bias in image-cropping AI,” 20 05 2021. [Online]. Available: https://www.bbc.co.uk/news/technology-57192898. [Accessed 22 10 2022].

[13] A. Preece, D. Harborne, D. Braines, R. Tomsett and S. Chakraborty, “Stakeholders in Explainable AI,” 29 09 2018. [Online]. Available: https://arxiv.org/pdf/1810.00184.pdf. [Accessed 26 02 2022].

[14] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence,* vol. 1, no. 5, p. 206–215, 2019.

[15] Oxford University Press, “myth, n. : Oxford English Dictionary,” Oxford University Press, 2022. [Online]. Available: https://www.oed.com/view/Entry/124670. [Accessed 03 12 2022].

[16] X. Hu, C. Rudin and M. Seltzer, “Optimal Sparse Decision Trees,” in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, 2019.

[17] A. Thampi, “Interpretability vs. Explainability,” in *Interpretable AI*, Shelter Island, New York, Manning Publications Co., 2022, pp. 14-15.

[18] C. Molnar, Interpretable Machine Learning - A Guide for Making Black Box Models Explainable, 2022.

[19] E. Pintelas, I. E. Livieris and P. Pintelas, “A Grey-Box Ensemble Model Exploiting Black-Box Accuracy and White-Box Intrinsic Interpretability,” *Algorithms,* vol. 13, no. 17, 2020.

[20] A. Fisher, C. Rudin and F. Dominici, “All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously,” *Journal of Machine Learning Research,* vol. 20, no. 177, pp. 1-81, 2019.

[21] G. Ras, M. van Gerven and P. Haselager, “Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges,” in *Explainable and Interpretable Models in Computer Vision and Machine Learning*, Switzerland, Springer, 2018, pp. 19-36.

[22] C. Rudin, “Please Stop Doing "Explainable" ML,” The Berkman Klein Center for Internet & Society, 19 08 2019. [Online]. Available: https://www.youtube.com/watch?v=I0yrJz8uc5Q. [Accessed 25 06 2022].

[23] IBM Research, “AI Explainability 360,” [Online]. Available: https://aix360.mybluemix.net/. [Accessed 25 06 2022].

[24] S. Russell and P. Norvig, “Reinforcement Learning,” in *Artificial Intelligence A Modern Approach Third Edition*, Harlow, Pearson, 2016, pp. 830-831.

[25] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova and C. Zhong, “Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges,” *Statistics Surveys,* vol. 16, pp. 1-85, 2022.

[26] Future of Life Institute, “The Artificial Intelligence Act,” [Online]. Available: https://artificialintelligenceact.eu/. [Accessed 25 06 2022].

[27] G. A. Miller, “The magical number seven, plus or minus two: Some limits on our capacity for processing information.,” *Psychological Review,* vol. 63, no. 2, pp. 81-97, 1956.

[28] V. Bahel, S. Pillai and M. Malhotra, “A Comparative Study on Various Binary Classification Algorithms and their Improved Variant for Optimal Performance,” in *IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 2020.

[29] E. Ferran, “Regulatory Lessons from the Payment Protection Insurance Mis-selling Scandal in the UK,” *European Business Organization Law Review (EBOR),* vol. 13, no. 2, pp. 247-270, 06 2012.

[30] S. Verwer and Y. Zhang, “Learning optimal classification trees using a binary linear program formulation,” in *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[31] S. Russel and P. Norvig, “Generalization and overfitting,” in *Aritifical Intelligence A Modern Approach Third Edition*, Harlow, Pearson Education Limited, 2016, p. 705.

[32] scikit-learn developers, “sklearn.ensemble.RandomForestClassifier — scikit-learn 1.2.0 documentation,” scikit-learn developers, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed 11 12 2022].

[33] D. Dua and C. Graff, “UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set,” UCI Machine Learning Repository

[http://archive.ics.uci.edu/ml], 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29. [Accessed 23 10 2022].

[34] N. Street, W. H. Wolberg and O. L. Mangasarian, “Nuclear Feature Extraction For Breast Tumor Diagnosis,” in *SPIE - The International Society for Optical Engineering*, San Jose, California, 1993.

[35] scikit-learn developers, “Permutation Importance vs Random Forest Feature Importance (MDI) — scikit-learn 1.1.2 documentation,” scikit-learn developers, 2022. [Online]. Available: https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html. [Accessed 24 10 2022].

Appendix A – Contents of artefact Directory

The artefact directory, includes the following files and structure:

- mounted (directory)
    - accuracy (directory)

      Contains the output accuracy calculation obtained during executing the experiments
    - diagrams (directory)

      Contains the output rendered diagrams (decision trees and permutation importance) obtained during the experiments
        - dtc_tree_{fold}_{lambda}.png

          The Decision Tree Classifier formed during the beginning of the OSDT process, in instances where the OSDT algorithm could not improve this is useful in understanding the tree structure. Named 00 through 09 in place of the {fold} marker and the appropriate lambda value is substituted into the {lambda} marker
        - rfc_permutation_importance_{fold}.png

          The Permutation Importance diagram for the Random Forest Classifier. Named 00 through 09 in place of the {fold} marker
        - rfc_tree_{fold}_{tree}.png

          The Decision Tree Classifier that votes as part of a Random Forest. Named 00 through 09 in place of the {fold} marker and 000 through 099 for the {tree} marker as the tree index within the forest
    - Data Preparation.ipynb

- The Jupyter notebook containing the pre-processing steps required to manipulate the original data into the form expected by OSDT.ipynb and Random Forest Classifier.ipynb
- OSDT.ipynb

  The Jupyter notebook containing the experiments performed that leverage the OSDT algorithm
- osdt.py

  The OSDT algorithm from GitHub
- Random Forest Classifier.ipynb

  The Jupyter notebook containing the experiments performed that leverage the Random Forest Classifier
- rule.py

  Support file for the OSDT algorithm from GitHub
- wdbc.data

  The original data as downloaded from Breast Cancer Wisconsin (Diagnostic) Data Set
- wdbc.names

  Describes the original data as downloaded from Breast Cancer Wisconsin (Diagnostic) Data Set
- wdbc-test-{fold}.csv

  Named 00 through 09 in the place of the {fold} marker and related to the testing data for the fold numbered (zero based)
- wdbc-train-{fold}.csv

  Named 00 through 09 in the place of the {fold} marker and related to the training data for the fold numbered (zero based)

- Analysis.xlsx

  A workbook into which the training and test data has been loaded, along with the accuracy results for the purposes of investigation and chart production
- docker-compose.yml

  A means by which a predictable working environment can be provisioned including the relevant packages used
- Fast-Track Ethics Application Approval - Newman20220915.pdf

  The received Fast-Track Ethics Approval
- README.md

  A document describing the means by which the experiments can be executed
- Stephen - Fast Track Ethics Form v3 Signed.pdf

  The submitted Fast-Track Ethics Form