Scientific Team Project KMD
Design and Implementation of Prototypical GUI for Model Agnostic Explanation Technique for Ruptured Status Classification

Akanksha Saxena (223681), Sachin Nandakumar (223685), Vritika Kalra (223674), Yash Shah (223740)
Supervisors: Prof. Myra Spiliopoulou, Uli Niemann

September 16, 2019

1 Introduction

Machine learning (ML) is being used in a variety of tasks these days. ML algorithms are categorised into supervised and unsupervised. Classification and regression are supervised techniques since target values associated with instances are available during learning phase. When the target values are not available and the goal is to explore the data to find meaningful relations in it, is called unsupervised learning. Clustering is unsupervised technique where the learning is done without any target variable. Association rule mining is another example. Then, there is reinforcement learning where an agent learns based on the feedback it receives.

These techniques are used particularly for datasets which are generally encountered in domains such as medicine,marketing, big data analysis and so on. Generally, complex models are built to predict the outcome of the model. While the models are scientifically sound, they are not easy to understand and explain. Hence, the question of why should the user trust the prediction of the model remains.

1.1 Motivation

Model explanation is especially required in critical domains like medical mining or aviation to enable an expert user understand the prediction. It serves many purposes. First, with the explanations provided, the user would be able to trust the outcome. Second, model explanations uncover hidden biases that the model might have. Both of the above reasons would help an expert user take better decisions. Additionally and legally, it will be required in nations where new set of comprehensive regulations such as the General Data Protection Regulation

(GDPR) are being adopted [10] [5]. These regulations clearly define the need for "Non-Discrimination" and "Right to explanation" under these new set of rules.

1.2 Fundamentals of Interpretable Machine Learning

Interpretable machine learning comprises of several algorithms and statistical methods which tries to understand the reasoning behind the prediction given by a model. For our study, we are restricting ourselves to model-agnostic methods and example-based explanations [6]. Therefore our definition of Interpretable Machine Learning is more concise, where, with the help of model's output a human can understand the reason behind a certain prediction. In other words, the higher the interpretability of the model, the easier it would be to justify the reason to a human for selecting a certain prediction over other as it also provides the grounds for selection. Selection is also based on the criticality of decision's impact in the real world. If the interpretation is successful, to what extent the model can be trusted. Counter-intuitively, if the model has failed, a human should know about the justification for the same.

1.3 Outline of the project

Our task is to explore four methods of model interpretation for a medical dataset of brain aneurysm. Below are the four tasks that have been implemented and integrated in a GUI. Furthermore, the detailed explanation of these tasks have been provided in the respective sections.

- Model reliance technique for determining Global Feature Importance
- Individual Conditional Expectation (ICE) plots for feature homogeneity
- Density plots to represent Counterfactual Instances
- Decision set for each class label(ruptured and unruptured)

We start by pre-processing of the given data, test ensemble models for a good accuracy and apply our model interpretation techniques on them. We combine all four methodologies in an independent GUI based application so the subject expert can view the findings and derive inferences from it.

2 Application Study

2.1 A brief summary of the data

The dataset provided to us for this project consists of 100 records. These records correspond to 100 intracranial aneurysms recorded at the university hospital of Magdeburg, Germany. Analysing the dataset, the records correspond to 77 female and 16 male patients where the age ranges from 33-85 years and consists

of 2 types of aneurysms, BF: Bifurcated and SW: Sidewall, with a distribution of 62 to 24. The dataset is heterogeneous since it consists of demographics of patients such as age and gender along with their morphological features. There are few missing values in the otherwise clean data.

2.2 Pre-processing

In the preprocessing stage of the dataset, we tried imputation of the values using various common strategies such as imputing with zeroes, mean as well as most frequent strategies, nearest feature strategies (we tried with all features, 4 features and 2 features). To feed the categorical variables to the model, those values were one-hot encoded. Since the data across all features were on a different scale, we also tried normalizing the dataset and also applied z-score to each of the features as a separate case.

2.3 Model design

Since the intent of the project was not to come up with an improved model, we chose 2 models which were motivated from [8]. We chose Gradient Boosted Trees and Support Vector Machines. We chose GradientBoostingClassifier of sklearn.ensemble and XGBoost for Gradient Boosted Trees and chose XGBoost among them based on the performance. Similarly, for SVMs, we used LinearSVC of the sklearn.svm class.

2.4 Model performance and parameter tuning

The performance was measured for every possible combination of models - 3 different imputation techniques, one-hot encoding of the categorical variables and 2 data transformation techniques along with the plain dataset without applying any transformation - with GradientBoostingClassifier, XGBoost and LinearSVC. Since the dataset is considerably small, we divided the whole process of evaluating the model into 2. We used nested cross validation with 5 inner splits and 10 outer splits to evaluate the generalization performance of the model. Hyperparameter tuning is achieved through gridsearch with 10 split repeated stratified cross validation and hence the model is chosen.

Out of all the possible combinations, standardizing the dataset using z-scores showed significant improvement in the overall performance of the model. The below set of figures shows the various combination of imputation techniques chosen along with the z-score for 3 sets of classification techniques.

	Imputation Technique	Categorical Encoding	Standardization	Model
Option 1	Fill with zeroes	One-hot encoding	z-score	GradientBoostingClassifier / XGBoost / LinearSVC
Option 2	Mean			
Option 3	Most frequent			
Option 3	2 nearest neighbours			
Option 4	4 nearest neighbours			
Option 5	All features as neighbours			

Figure 1: Different types of imputation techniques applied to a z-score one-hot encoded aneurysm dataset

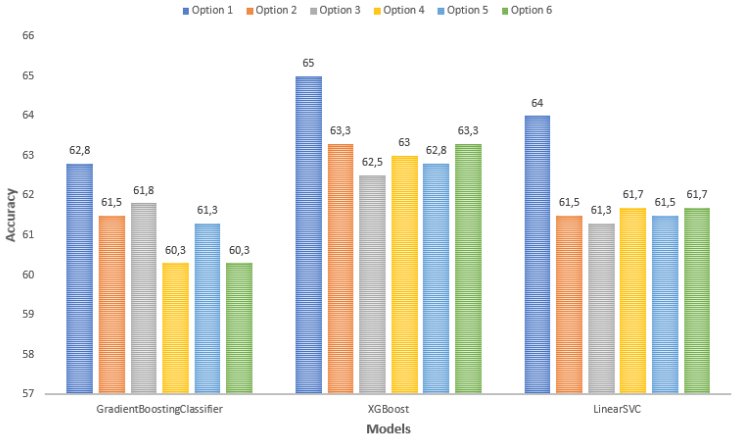

Figure 2: Performance chart of all models. Each option in the figure corresponds to the model used in Figure 1. Imputation with 0s always performed better than the rest.

From the above bar chart it is visible that in all the three cases, imputing the missing values with zero always stood out. Comparing between all the three, XGBoost and LinearSVC came out on top with an accuracy score of 0.65 and 0.64 respectively.

3 Implementation of Interpretable Machine Learning methods

3.1 Model Reliance technique for determining global feature importance

The idea behind this task is to find the most important features that contribute to the prediction of the aneurysm classes. Since the dataset consists of demographics as well as morphological features, it is important to understand which

feature contributes most to the prediction especially in a medical domain. Finding feature importance also contributes to feature selection which is one of the core and influential concept in machine learning that affects the performance of the model.

Feature importance in itself has got its distinctions. First distinction is between model-agnostic and model-specific importance. Model-agnostic interpretation methods provide great deal of flexibility. This means, various interpretation methods can anyway be applied irrespective of which machine learning model is used. On the other hand, Model-specific interpretation methods restricts the explanation to that model alone. For example, algorithms like RandomForest and XGBoost automatically calculates important features on a trained predictive model. The second distinction is between global and local feature importance. Local feature importance evaluates the importance of features in case of prediction of a sub-problem. Whereas, global feature importance considers the overall predictions into account. Thus, to define the most important global features irrespective of the model used, we went for a model agnostic approach called model reliance. Model reliance or MR as cited in [6] [2] is used as a core measure in determining the model class reliance (MCR). MCR is defined as a range of values (that summarizes many prediction models) that takes into account the Variable Interest (VI) of model's prediction accuracy for a given class. MR is a permutation feature importance algorithm which was first introduced in [2]. Using MR, we initially find the original estimated model error. Then for each variable, we perturb all its values and estimates the corresponding model error. MR measure for that variable is calculated by dividing the perturbed estimated model error by the original estimated model error. The operation of division or subtraction works well in this case. Here, instead of finding any covariates and hence relating them to the importance of a feature, we shuffle the values of each variable against the whole dataset and estimates the model error. The algorithm is explained well in [2].

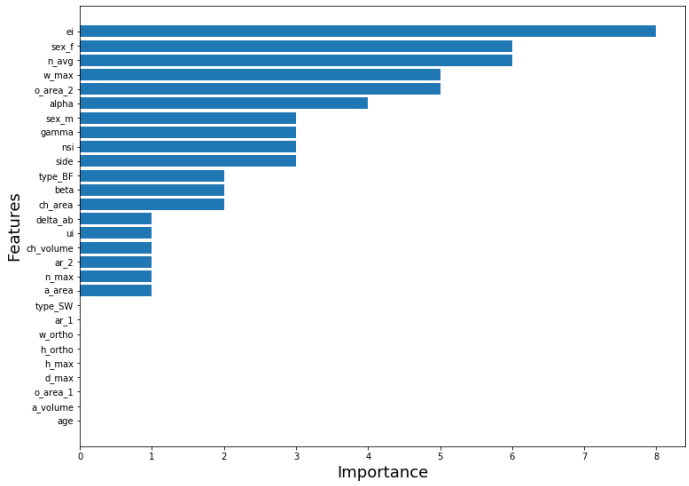

Figure 3: The above bar chart shows the Model Reliance measures for each of the features of the dataset and are sorted in the descending order. This implies that the feature with highest measure, **ei**, is the most important feature for the particular model under consideration. This particular MR values are that when permutation importance is run on Gradient Boosted algorithm.

We implemented model reliance using permutation feature importance algorithm on top of both the models, Gradient Boosted Trees and Support Vector Machines, and have provided the results in terms of a horizontal bar chart. The MR measure for each instance was the subtraction between the two errors. This could possibly be a workaround when the original estimated error is zero.

3.2 Individual Conditional Expectations (ICE) plots

A black-box model can have complicated parameters which can range from a few thousand to millions for a real-world dataset. A key question about model interpretation is "How does the model inputs work?" or to rephrase "How does change in a particular variable v affect the model's prediction?". One of the ways as discussed in the previous section, feature importance shows the strength of the relationship between a variable and model's predictions. But it lacks in providing any functional relationship between model inputs and predictions.

Friedman's PDP or Partial dependence plots addresses this issue by visualizing the mean change in prediction value for a particular parameter. But as the original paper states [3], PDPs are only useful for summarising chosen subset of parameters when the interrelation between remaining parameters is weak. Also for a given hypothesis space X, PDPs are not effective in revealing extrapolations and outliers.

3.2.1 ICE

Individual conditional expectations (ICE) is disintegrated form of PDP from a visual perspective. It plots one individual line graph for every instance which shows the final output change when values of feature change [4] [6]. Mathematically, consider observed values for an instance, $x(v)$ and its predicted output, $x(p)$. For n observations $\{x(v), x(p)\}_{i=1}^{n}$, a line is plotted against every observed value of $x(v)$. The x-axis remains fixed values of $x(v)$ and $x(p)$ is varied on y-axis for n observations. Each line establishes a homogeneous (if any) relationship between each line and $x(v)$ thereby giving end-user several inferences of conditional relationships modeled by the black-box algorithm. In terms of actual implementation, the ICE line plot for a single observation is observed by creating a dataset from the given data where each unique observation from the selected feature is replicated for all the other instances of the given data and prediction is observed.

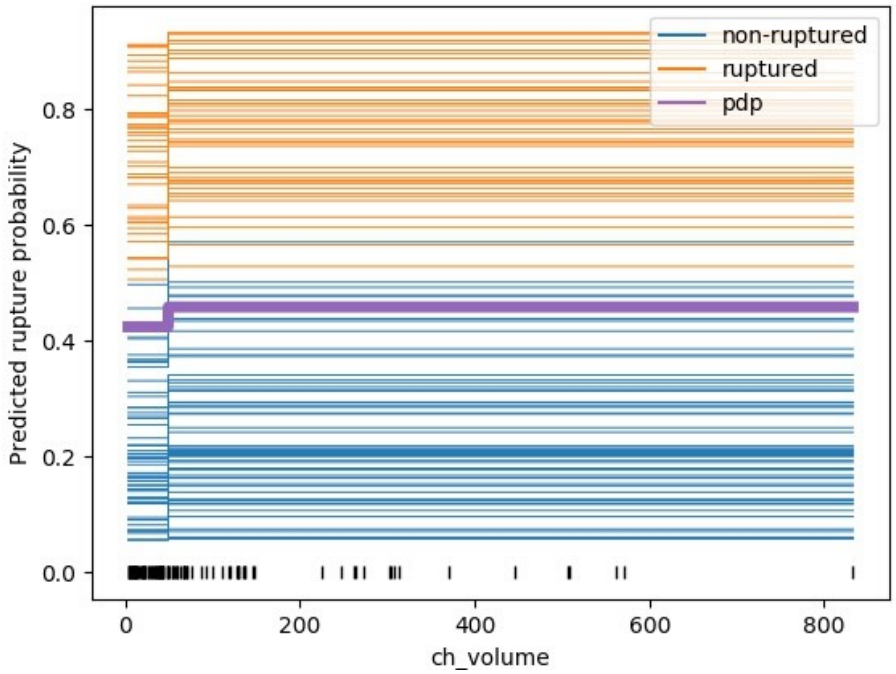

Figure 4: ICE plot of the feature 'ch_volume' where x-axis shows range of feature value and y-axis shows values ranging from 0 to 1 showing probability of rupture classification. Each line also shows rupture status and a Partial dependency plot shows overall trend.

A rug plot is used here to visualise the distribution of the data. Mostly, it is used with either a histogram or density estimation plots. Rug values for each feature is placed along the x-axis. The term "rug" loosely translates to

perpendicular markers that look like tassels along the edges of the given plot.

3.2.2 Centered ICE

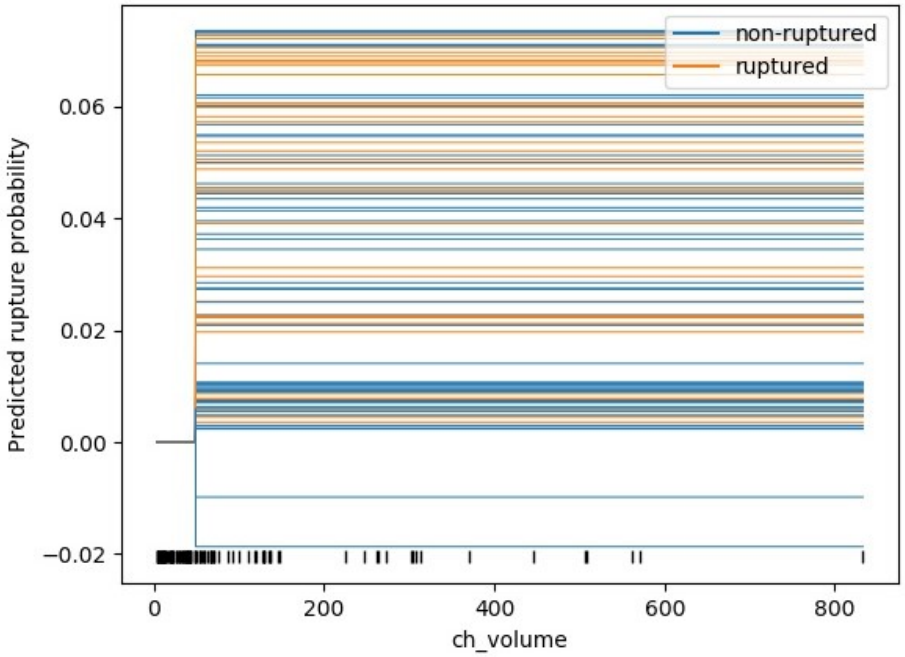

Figure 5: "centered" ICE plot of the feature 'ch_volume'. We selected the smallest value of any given feature and it's corresponding prediction value for centering all further values with respect to it.

For each plotted line for a given feature, it is difficult to find any real difference or change in the prediction value (here in case - change in classification probability) as all of them start at different predictions. Centered ICE (c-ICE) provides a simple solution to center the values at a particular fixed point in the feature and center all the lines with respect to chosen point [4] [6]. According to [4], given equation defines c-ICE.

$$\hat{f}_{cent}^{(i)} = \hat{f}^{(i)} - 1\hat{f}(x^*, x_{Ci})$$

where the unadorned $\hat{f}$ denotes the fitted model and 1 is a vector of 1's of the appropriate dimension. Hence the point $(x^*, (x^*, x_{Ci}))$ acts as a "base case" for each curve.

3.3 Density plots to represent Counterfactual Instances

The motivation for this task comes from example based explanation methods. We use counterfactual explanations [6] [10] [7] to explain the model. Counterfactual explanations are in form of "if x had not occurred, y would not have occurred". Counterfactual examples are hypothetical instances which flips the prediction of the original instance [6]. Since we work with only input and output, counterfactual explanation is a model agnostic approach. Due to the large number of instances, it is usually challenging to represent the higher number of counterfactuals in a more meaningful format. For this task, we show the nearest counterfactual for every feature using density plot.

We select one instance from the data in order to explain the learned model. We use range of each feature (min value of the feature in the dataset and max value of the feature in the dataset) to create example counterfactual value. For each instance, we change only one feature value at a time and replace it with the calculated feature value to create our example counterfactual instance. We create 100 such example counterfactual instances. We now look for counterfactual instances that can flip the target value from 'ruptured' to 'unruptured' or vice versa. We then predict the outcome of example instances using our learned model. For learning our model we use Support Vector Machine and Gradient Boosted Trees as mentioned previously. We then return the first example instance that changes the target value of the instance as our counterfactual instance.

Figure. 6 represents the density plot of the feature 'ch_volume' for the instance number 77. The original value of the instance is represented by orange dashed line and the counterfactual value is represented by blue dashed line. As visible from the plot, the target value of the instance 77 changes from ruptured to non ruptured if the value of ch_volume changes from 48.07 to 52.83.

3.4 Decision set for each class label

Decision rules are one of the most interpretable methods since they are easier to explain in natural language. The structure of a decision rule is in terms of if-then structure. If the condition is true then the prediction is made [6]. Generally, decision rules are easier to explain if they are short in length [6], i.e., few feature value combined with an AND to give a certain prediction. A set of decision rule is called decision set.

For this task, we have used decision rules to explain the prediction of the learned model for each ruptured and non ruptured classes. The model is learned using support vector machine and gradient boosted trees. To learn the rules, we use sequential covering approach that learn rules and create a decision set covering the whole dataset. We use Ripper algorithm [1] [6] [9] to implement decision rules which is one of the variants of sequential covering decision rule technique.

Sequential covering technique first learn a single rule that applies to most of the data in the data set. It then adds the rule to the decision set and then re-

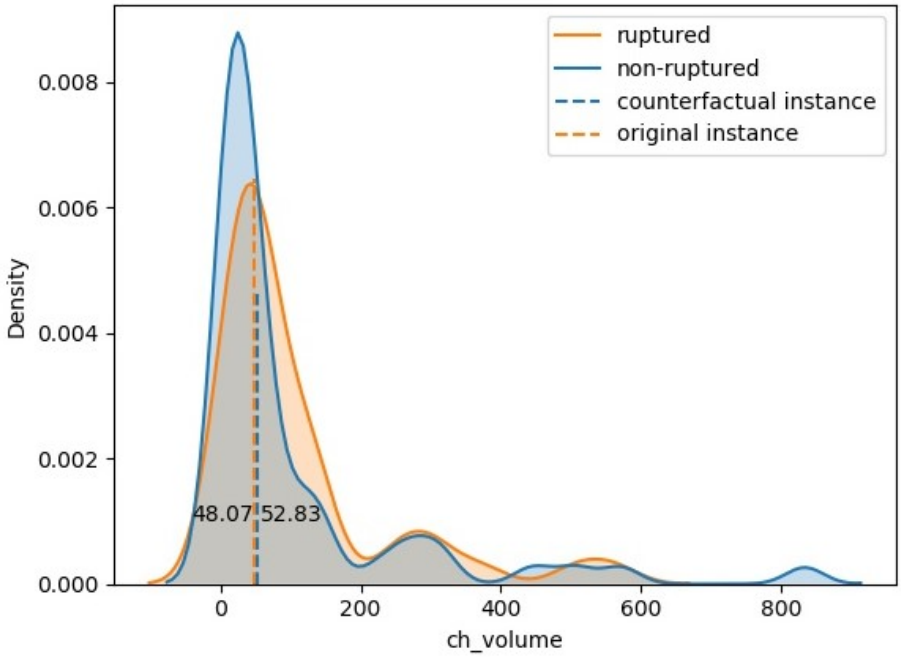

Figure 6: Density plot of the feature 'ch_volume' representing original and counterfactual feature value

moves the data it used to learn the rule. Consecutively, the next rule is learned on the remaining data set. The process is repeated till all the data are covered [6]. At the end a decision set is created. The Ripper algorithm is presented in [1] [6].

RIPPER(Repeated Incremental Pruning to Produce Error Reduction) Algorithm is a rule-based classifier which has a direct rule extraction method from the data.

Four Important phases of this algorithm continue with

i) Growth: until a rule reaches its stopping conditions, multiple attributes are added to the rules continuously

ii) Pruning: each rule is pruned incrementally until we have achieved a pruning metric by conceding pruning of attribute's final sequence.

iii) Optimization: generated rules are further optimized by using 2 types of approach. One by adding an attribute to the original rule or by considering a new rule growing it independently.

iv) Selection: Most beneficial rules are retained and rest are discarded.

RIPPER algorithm is capable of balancing precision and recall of the class which helps in minimizing misclassification cost of test data. For RIPPER's pruning and optimization stages, suitable weights are assigned to false positive errors and false negative errors respectively.

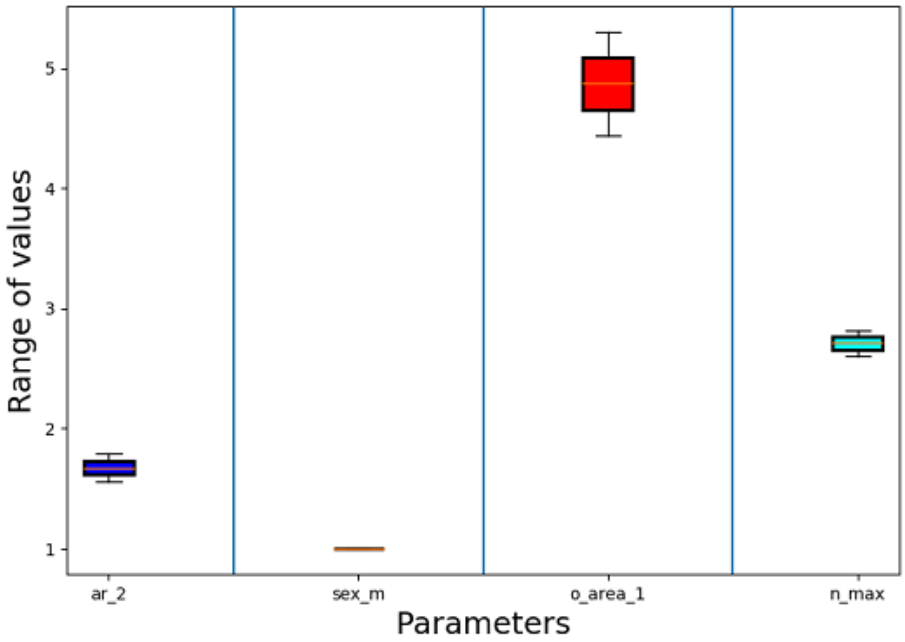

Figure 7: Decision set for 'ruptured' for model trained using GBT

For the project, we use python 'wittgenstein' package that helps us find the decision set of both ruptured and unruptured examples. We train the model using GBT and SVM. We then predict the outcome of all the examples using the learned model. Then, using the predicted outcomes, we use Ripper algorithm [1] [6] to find out decision rules by providing instance feature and predicted outcome as input for each class label.

Figure 7 and Figure 8 show box plots of decision sets created by RIPPER [1] for ruptured and unruptured class labels. In Figure 8, the range of the first rule of the decision set 'sex_m $\wedge$ a_area' is [0.0] for 'sex_m' and [16.53, 28.74] for 'a_area' and this rule is satisfied by 11 instances. The range of each parameter is represented by the box plot whereas the number of instances that are covered by each decision rules in the set are mentioned in the GUI. Different box plots (separated by vertical lines) in both the figures give the value of different decision rules in the decision set. In Figure 8, we can see the rules of the form feature-value pair connected by AND ($\wedge$), i.e. disjunction of conjunctions.

3.5 GUI

The objective of the whole project is to design and implement a prototypical GUI that explains the above mentioned four model agnostic approaches. Hence, we came up with a GUI based solution called G-MARC (GUI for Model Agnostic

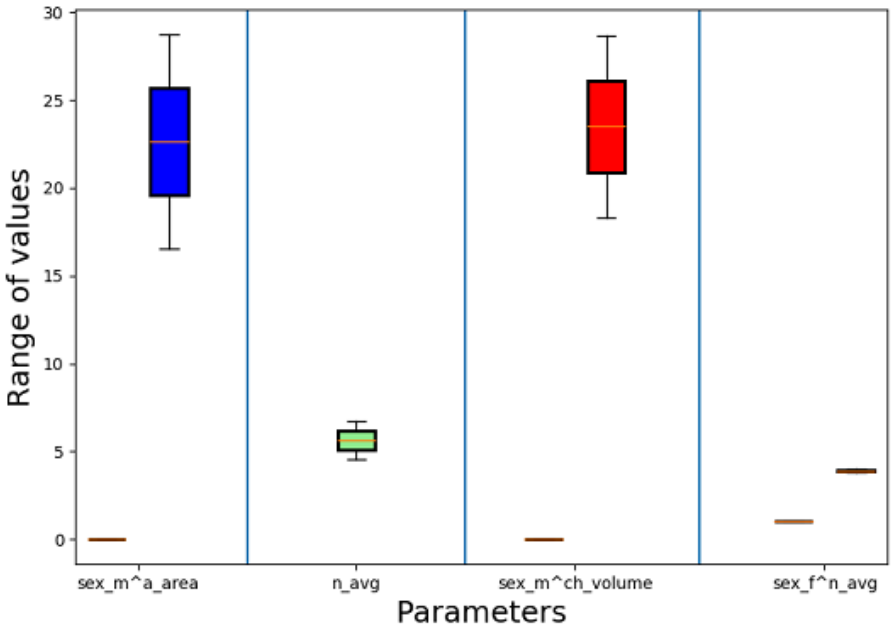

Figure 8: Decision set for 'unruptured' for model trained using GBT

explanations for Rupture status Classification). It is implemented using the Qt software which is a FOSS widget toolkit for creating GUIs.

The GUI is divided into 4 tabs for each of the model agnostic techniques. Each tab provides the option for the user to select the model for which he/she wants the result. In addition to that, various other options such as selecting the features, type of plot, choosing instances and selecting a particular aneurysm class are provided based on the requirement of the technique. The GUI is also made user-friendly by providing a toolbar for the plots where the user can stretch or adjust the plot, zoom values, modify the layout and axes and also save each plot as an image to his/her local system.

Additional work: In addition to the tasks, we came up with an idea where the user is provided with an option to download the report with the necessary information and plots of each task. Since we don't save all plots (other than some very few necessary ones to run the GUI) in the directory, saving and writing the plots (as images) to the report is quite tedious in terms of time complexity. So, this process has been implemented as a separate thread to make sure that the GUI does not become non-responsive. We also provide a Help option in the menu bar which is moreover a guide for the user regarding the GUI.

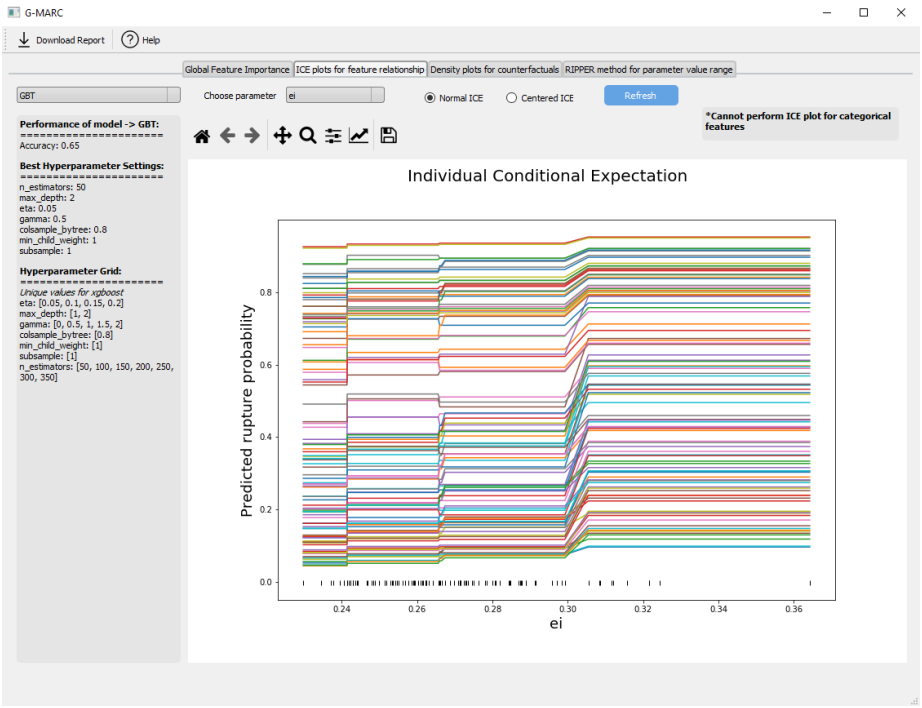

Figure 9: Screenshot of GUI for 4 model-agnostic techniques

The whole application has been converted into a single standalone executable. The GUI with all the features are compiled into a Windows (platform-specific) executable file with the help of cx_freeze package of python. This helps the user to install the application without manual installation of python or any of its dependencies by running the msi file by following the INSTALL.txt provided along with it. But sometimes there might be some dll dependencies that aren't sorted out automatically by cx_freeze. All the features with their respective description is given in Table 1

4 Results and Discussion

4.1 Findings on the dataset

The model reliance points out that in case of both the predictive models, the **ei** feature is the most globally significant one. Other features such as **w_max**, **o_area_2** and **beta** are also found to contribute to the predictability of both the models with fair consistency. 9 features are found to be irrelevant in case of Gradient Boosted Trees and 16 in case of Support Vector Machine models. But this does not mean retraining the model by removing the insignificant features will increase the performance of the model. This might or might not increase

Table 1: Features (Morphological and Non-Morphological) used for classification.

Feature	**Description**
age	age of the patient
side	side of the Aneurysm
A_A	Area of the aneurysm (without the ostium)
V_A	Volume of the aneurysm $[mm^3]$
A_{O1}	Area of the ostium (variant 1) $[mm^2]$
A_{O2}	Area of the ostium (variant 2) $[mm^2]$
D_{max}	Max. diameter of the aneurysm $[mm]$
H_{max}	Max. height of the aneurysm $[mm]$
W_{max}	Max. width of the aneurysm perpendicular
H_{ortho}	Height of the aneurysm (approximated)
W_{ortho}	Max. width parallel to the projected ostium
N_{max}	Max. NC diameter, i.e., the max. possible
N_{avg}	Avg. NC diameter, i.e., the mean distance
AR_1	Aspect ratio: H_{ortho}/N_{max}
AR_2	Aspect ratio: H_{ortho}/N_{avg}
V_{CH}	Volume of the convex hull of the aneurysm
A_{CH}	Area of the convex hull of the aneurysm
EI	Ellipticity index: $1 - (18\pi)^{\frac{1}{3}} V_{CH}^{\frac{2}{3}}/A_{CH}$
NSI	Non-sphericity index: $1 - (18\pi)^{\frac{1}{3}} V^{\frac{2}{3}}/A$
UI	Undulation index: $1 - \frac{V}{V_{CH}}$
α	Min. of DB_1B_2 and DB_2B_1 $[deg]$
β	Max. of DB_1B_2 and DB_2B_1 $[deg]$
γ	Angle at D, i.e. B_1DB_2 $[deg]$
$\Delta_{\alpha\beta}$	Abs. difference between α and β $[deg]$

the generalization performance but the whole logic of permutation feature importance is based on a fixed model where it is not required to retrain the model. Retraining by reduced set of features changes the whole model and hence the feature importance.

It's necessary to follow up with more testing to actually understand the importance of each feature. The ICE plot shown in Figure. 4 and Figure. 5 for feature **ch_volume** shows its interaction with rest of the feature values. Here we see that the prediction value change is minor for most of the values and it doesn't change the rupture status. But a few values of the given particular feature when interacting with some instances shows the change in rupture status probability. And to investigate further, We see counterfactual plot in Figure. 6 where for one particular instance, we can see even a minor change in the value results in change of prediction. We deliberately selected **ch_volume** for our analysis in this report to form basis for our findings that features which does not show up on top of model reliance are also significant and retraining the

model with reduced set will result in change of model and hence, predictions.

4.2 Limitations

Although, our project's final aim is achieved but there are several limitations, some of which can be addressed in future work. First of all, the models we selected after running tests for ensemble machine learning models for our model agnostic methods haven't taken into account deep learning models like Artificial Neural Networks (ANNs). Secondly, the ICE plot functionality for our project doesn't take categorical features and is limited to continuous valued (morphological features) attributes and interval-valued attributes. Additionally, in terms of GUI, it is limited to our selected models and data. Any additional modification in them would require recompiling source-code and rebuilding GUI. The GUI could possibly become non-responsive when queried for plots in case of the task mentioned in section 3.3. This is because of the overhead of predicting the counterfactuals by permuting the dataset for each feature.

4.3 Future Work

For future work, we suggest the below tasks that can help us further with the objectives of the project:

- Since the data set is very small, we feel that having more data would help us get more insights even with the similar implementation. Hence, we suggest collecting more data for both ruptured and unruptured classes is required.
- More approaches to understand model agnostic methods can be explored like *SHapley Additive exPlanations* to understand individual predictions [6].
- The GUI can be made more responsive by running several time consuming piece of codes in separate threads. This might increase the cost of optimization overhead which can then be taken as a separate work.

5 Summary

For this project we first analysed the data provided and did the required preprocessing. We then used GBT and SVM to learn the model. To explain the prediction of these two models, we used four different Model Agnostic approaches to understand the learned model. We used Feature Importance, ICE plots, Counterfactual Explanations and Decision Rules to explain the learned model. At the end we integrated the implementation of the four tasks into a single GUI.

References

[1] Mountain Avenue, Murray Hill, William W Cohen, Complexity Of, and Rule Pruning. Fast Effective Rule Induction. In *Machine Learning: Procedings of the Twelfth International Conference(ML95)*, 1994.

[2] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *arXiv e-prints*, page arXiv:1801.01489.

[3] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001.

[4] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24:44–65, 2015.

[5] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38:50–57, 2017.

[6] Christoph Molnar. *Interpretable Machine Learning*. leanpub, 2019.

[7] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. *arXiv preprint arXiv:1905.07697*, 2019.

[8] Uli Niemann, Philipp Berg, Annika Niemann, Oliver Beuing, Bernhard Preim, Myra Spiliopoulou, and Sylvia Saalfeld. Rupture Status Classification of Intracranial Aneurysms Using Morphological Parameters. In *Proc. of IEEE Int. Symposium on Computer-Based Medical Systems (CBMS)*, pages 48–53, 2018.

[9] Achilleas Tziatzios. Data Mining of Range-Based Classification Rules for Data Characterization. (March), 2014.

[10] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard journal of law technology*, 31:841–887, 04 2017.