Locally Interpretable Predictions of Parkinson's Disease Progression

Qiaomei Li[*] Rachel Cummings[*] Yonatan Mintz[*]

March 24, 2020

Abstract

In precision medicine, machine learning techniques have been commonly proposed to aid physicians in early screening of chronic diseases such as Parkinson's Disease. These automated screening procedures should be interpretable by a clinician who must explain the decision-making process to patients for informed consent. However, the methods which typically achieve the highest level of accuracy given early screening data are complex black box models. In this paper, we provide a novel approach for explaining black box model predictions of Parkinson's Disease progression that can give high fidelity explanations with lower model complexity. Specifically, we use the Parkinson's Progression Marker Initiative (PPMI) data set to cluster patients based on the trajectory of their disease progression. This can be used to predict how a patient's symptoms are likely to develop based on initial screening data. We then develop a black box (random forest) model for predicting which cluster a patient belongs in, along with a method for generating local explainers for these predictions. Our local explainer methodology uses a computationally efficient information filter to include only the most relevant features. We also develop a global explainer methodology and empirically validate its performance on the PPMI data set, showing that our approach may Pareto-dominate existing techniques on the trade-off between fidelity and coverage. Such tools should prove useful for implementing medical screening tools in practice by providing explainer models with high fidelity and significantly less functional complexity.

1 Introduction

In precision medicine, machine learning techniques have been commonly proposed to aid physicians in early screening of chronic diseases. Many of these diseases become more difficult to treat as they progress, so accurate early screening is critical to ensure resources are directed towards the most effective treatment plan [Pagan, 2012]. Since the final treatment decision must inevitably be made by a doctor, these screening procedures should be interpretable such that a clinician can explain the decision-making process to patients for informed consent. However, the types of models that achieve the highest level of accuracy given early screening data tend to be extremely complex, meaning that even machine learning experts have difficulties explaining why certain predictions are made, leading many to describe them as "black box" [Breiman, 2001]. In this paper, we bridge this gap by providing a novel approach for explaining black box model predictions which can give high fidelity explanations with lower model complexity.

In particular we will focus on early screening of Parkinson's Disease (PD). PD is a complicated neurodegenerative disorder that affects the central nervous system and specifically the motor control of individuals [mjf, 2019]. This disorder is estimated to affect 930,000 individuals in the US by 2020, and is more prevalent in the geriatric population affecting more then 1% of the population over the age of 60 and 5% of the population over age 85 [Findley, 2007, Kowal et al., 2013, Rossi et al., 2018]. These statistics and other recent studies on Parkinson's epidemiology indicate that as the population ages, the prevalence of PD is expected to grow to over 1.2 million by 2030 in the US alone, increasing the total economic burden of the disorder to approximately $26 billion USD [Kowal et al., 2013, Rossi et al., 2018]. One of the most challenging

[*]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology. Emails: {qli374, rachelc,ymintz}@gatech.edu. Part of this work was completed while R.C. was a Google Research Fellow at the Simons Institute for the Theory of Computing.

aspects of PD research is that there is still no clear consensus on the root cause, and whether it is a single disease or a group of diseases characterized by similar symptoms known as Parkinsonism [Rao et al., 2006]. Since the disorder manifests differently between individuals (with different primary symptoms expressed across different patients) [Rao et al., 2006, Fereshtehnejad et al., 2017, Fereshtehnejad and Postuma, 2017], studying sub-categorization of PD disease progression has been of great interest in the medical community, particularly using novel advances in data-driven and statistical methods.

In this paper, we will use data from the Parkinson's Progression Marker Initiative (PPMI) [PPM] to develop and analyze a method for classifying patients based on their disease progression, and to provide data-driven PD sub-types. We develop a model that can be used with screening measurements to predict how a potential patient's symptoms are likely to develop over the course of the following two years. Our resulting five sub-types correlate well with known primary PD symptoms and have clear medical interpretations. We then develop a random forest model which can accurately predict which of these sub-types a patient should be classified into, given common screening data. Since this model is a black box, we additionally develop a method for generating *local explainers* for each prediction. Our local explainer methodology uses a computationally efficient information filter to include only the most relevant features to explain a given prediction, resulting in a methodology we believe useful for implementing such screening tools in practice by providing explainer models with high fidelity and significantly less functional complexity. We then develop a global explainer methodology by aggregating local explainers. We use an Integer Programming based approach to determine which local explainers to include in our global explainer. Our global explainer must trade off between coverage, fidelity of predictions, and interpretability. We show that it is on the Pareto frontier of this trade-off space, relative to existing methods. Additionally, many other global explainers are constrained to have perfect coverage, while our method has an additional degree of freedom, which allows for improvements in fidelity and interpretability.

The remainder of the paper will proceed as follows. A discussion of related literature and previous work connected to interpretable machine learning and PD diagnosis is given next in Section 1.1. We will discuss our data driven cluster analysis for determining PD progression sub-types in Section 2. Then in Section 3 we will discuss our local explainer methodology and provide with numerical validation of this methodology in Section 4. In Section 5 we show how to extend this local explainer framework into a global explainer using an Integer Programming (IP) framework, and in Section 6, we provide empirical validation and compare the performance of our IP-based approach with other local and global explainer methods.

1.1 Related Work

Due to the prevalence and complexity of PD, there has been a significant amount of literature focused on using data-driven methods and machine learning to assist with diagnoses. Several diagnosis methods have been proposed including those that use classical ML models such as kernel SVMs [Prashanth et al., 2016], ensemble models [Latourelle et al., 2017, Castillo-Barnes et al., 2018], and both supervised and unsupervised deep learning methods [Hirschauer et al., 2015, Adeli et al., 2016, Peng et al., 2017, Singh et al., 2018]. The classical and ensemble methods have typically focused on lab tests and genetic markers, while the deep learning methods were developed to incorporate MRI imaging into these predictions. The majority of this work focuses on binary diagnoses, labeling individuals as either healthy or having PD, but do not give information on disease progression or disease sub-types. Also, most of the proposed methods—particularly the ensemble and deep learning methods—are difficult to interpret. For example, they may identify a region of interest in an MRI image or highlight certain genetic markers, but it is difficult to explain to clinicians or patients why these regions are important for the model's final decision. In contrast, our model is meant to predict the disease progression of individuals based on early screening data. To ensure interpretability, we introduce a local and global explainer techniques so that proper and clear rationale can be given to classifications.

In addition to the work on diagnoses, there has also been significant research into the use of data-driven methods for PD sub-type identification [Graham and Sagar, 1999, Erro et al., 2013, Fereshtehnejad et al., 2015, Fereshtehnejad and Postuma, 2017]. The majority of analyses that fall in this stream of literature focuses on using unsupervised methods such as k-means clustering to create patient sub-types based on screening data, and then track the importance of the clustering based on longitudinal progression observations. In contrast, our model will first cluster patient types based on the dynamic behavior of disease progression, and then attempt to predict these clusters using screening data. We believe this approach will be useful in identifying the most effective course of treatment by directly treating the primary symptoms that develop in each progression cluster.

Our paper also draws on previous work in the broader field of interpretable machine learning. The two primary types of interpretable learning include models that are interpretable by design [Aswani et al., 2019], and black box models that can be explained using global explainer [Wang and Rudin, 2015, Lakkaraju et al., 2016, Ustun and Rudin, 2016, Bastani et al., 2018] and local explainer [Ribeiro et al., 2016, 2018] methods. Models that are interpretable by design are perhaps the gold standard for interpretable ML; however, these models often require significant domain knowledge to formulate and train, and are therefore not suited for exploratory tasks such as PD diagnosis. Global explainer methodology attempts to train an explainable model (such as a decision tree with minimal branching) in order to match the predictions of a black box model across the entirety of its feature space. While these models may provide some understanding on the general behavior of the black box model, if the relationship between features and black-box predictions is too complex, then the global explainer may remove many subtleties that are vital for validation and explanation. In contrast, local explainer methods attempt to train simpler models centered around a the prediction of a single data point. The most commonly used local explainer methods are Local Interpretable Model-Agnostic Explanations (LIME) [Ribeiro et al., 2016] and anchors [Ribeiro et al., 2018]. While local methods cannot validate the full black box model, they are useful for understanding the subtleties and justification for particular predictions. The method we propose in this paper follows from the idea of local explanations. We then aggregate these local explainers into a global explainer, trading off between coverage of the global model and fidelity of the local explainers that comprise our global model. We believe this method is most appropriate for the problem of PD diagnosis, where the relationship between different screening measures and the diagnosis is quite complex, and the model should incorporate the richness of this relationship in its predictions.

2 Clustering Methodology and PPMI Dataset

PD is a complex disorder, and is often expressed differently by different patients, which has motivated the need to create PD sub-types to better direct treatment. While many existing data-driven methods focus on clustering patients based on their baseline measurements [Fereshtehnejad and Postuma, 2017], we propose clustering patients using the trajectory of how their symptoms progress.

We will use data collected in the PPMI study [PPM], which is a long run observational clinical study designed to verify progression markers for PD. To achieve this aim, the study collected data from multiple sites and includes lab test data, imaging data, genetic data, among other potentially relevant features for tracking PD progression. The study includes measurements of all these various values for the participants across 8 years at regularly scheduled follow up appointments. The complete data set contains information on 779 patients, and included 548 patients diagnosed with PD or some other kind of Parkinsonism and 231 healthy individuals as a control group.

2.1 Determination of Criterion and Cluster Analysis

Since there is significant heterogeneity in how PD symptoms are expressed, there also is no agreement on a single severity score or measurement that can be used as a surrogate for PD progression. Thus instead of considering a single score, we will model the severity of the disease as a multivariate vector, and the disease progression as the trajectory of this vector through a multidimensional space. Using the PPMI data [PPM] and other previous literature on PD progression [Rao et al., 2006, Martinez-Martin et al., 2017, Bhat et al., 2018], we considered the following measures of severity to model disease progression:

- Unified Parkinson's Disease Rating Scale (UPDRS) II & III [Martínez-Martín et al., 1994]: The UPDRS is a questionnaire assessment that is commonly used to track symptoms of PD by an observer. It consists of four major sections, each meant to measure a different aspect of the disease. These sections are: (I) Mentation Behavior and Mood, which includes questions related to depression and cognitive impairment; (II) Activities for Daily Living, which includes questions related to simple daily actions such as hygiene and using tools; (III) Motor Examination, which includes questions related to tremors and other physical ticks; and (IV) Complications of Therapy, which attempts to assess any adverse affects of receiving treatment. For our analysis we focused on the aggregate scores of sections II and III of the UPDRS to track physical symptoms of the disease.

- Montreal Cognitive Assessment (MoCA) [Nasreddine et al., 2005]: Although not exclusively used for PD, the MoCA is a commonly used assessment for determining cognitive impairment and includes sections related to attention, executive functions, visual reasoning, and language. For our analysis, we used the MoCA scores of the individual patients as surrogates for their cognitive symptoms.
- Modified Schwab and England Activities of Daily Living Scale (MSES) [Siderowf, 2010]: The MSES is a metric used to measure the difficulties that individuals face when trying to complete daily chores due to motor deficiencies. This assessment is generally administered at the same time as the UPDRS and is often appended as a section V or VI. We used this score as a measure of how much autonomy the patients experience based on their symptoms.

We formed the empirical trajectory of these scores for each patient using the values measured during the patients' participation in the PPMI study [PPM]. For our cluster analysis we used longitudinal measurements that were taken across the first seven visits of the study corresponding to a period of 21 months, where the first measurement formed the patient's baseline, and the next five measurements were taken at follow up visits at regular three month intervals; the final measurements were taken after six months. We chose this timeline for our analysis because participation was high among all participants in the study during this period, so we did not have to exclude any patients, and visits were more frequent to better capture disease progression over time. After these seven measurements, follow-up visits were scheduled too infrequently to provide useful trajectory modeling information.

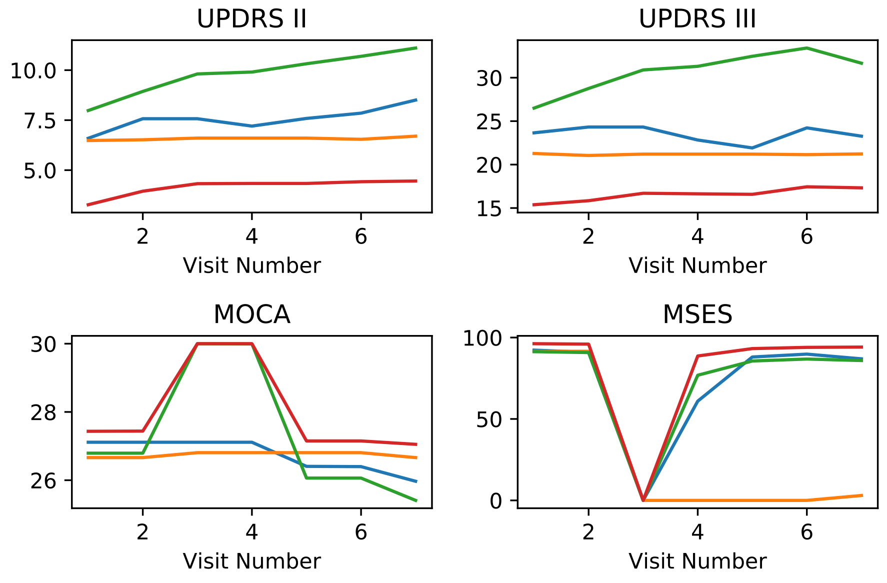
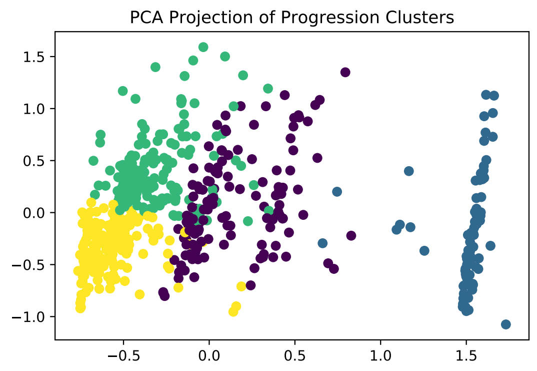


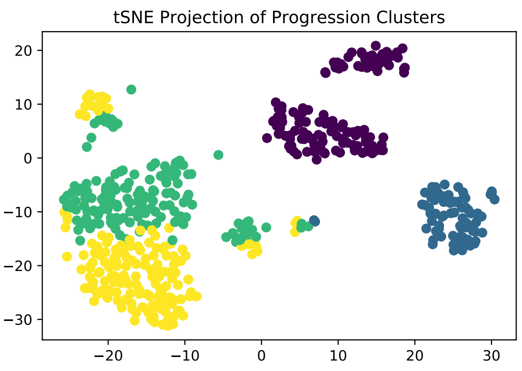
Figure 1: Mean trajectory progression for given score by cluster. Blue corresponds to Group 0, orange corresponds to Group 1, green corresponds to Group 2, and red corresponds to Group 3. The y-axis of each plot the is numerical value of the corresponding disease severity measure.

We used these trajectories to cluster the patients together into progression sub-types. The main motivation for this approach is that if patients' severity scores progress in a similar way, then it may identify a useful sub-type for treatment design. Only patients diagnosed with PD were included in the cluster analysis, since we are interested in finding useful sub-types of disease progression. Each trajectory was then flattened out as a 28 dimensional vector, with the first four entries corresponding the measurements at baseline, the next four for the 3 month follow up, and so on. Using scikit-learn and Python 3.7, we performed k-means clustering on these trajectories to define our sub-types [Pedregosa et al., 2011, Friedman et al., 2001]. Using cross validation and the elbow method (as seen in Figure 8 in the appendix), we determined that there are four potential sub-types of disease progressions for the PPMI participants. We label these as: moderate physical symptoms cognitive decline cluster (Group 0), stagnant motor symptoms autonomy decline cluster (Group 1), motor symptom dominant cluster (Group 2), and moderate symptoms cluster (Group 3). The names we assigned to each individual cluster were given by the observed mean trajectories of the relevant scores for individuals that were classified into a particular cluster as shown in Figure 1.

In Figure 2 we show two 2-dimensional projections of the different cluster groups. Figure 2a shows the projection onto the first two principal components of the data using PCA [Friedman et al., 2001]; this projection method is meant to preserve linear relationships among data points as well as distances between data points that are far apart. The projection shown in Figure 2b corresponds to the tSNE projection of the data onto a two-dimensional space [Maaten and Hinton, 2008], this projection method was designed with manifolds in mind and is meant to preserve close distances (i.e., data points close in the tSNE projection should be also close in the higher dimensional space). Note that in both projections our resulting clusters are distinct and do not significantly overlap.



(a) PCA Projection



(b) tSNE Projection

Figure 2: Two different 2-dimensional projections for visualizing trajectory clusters. Purple corresponds to Group 0, blue corresponds to Group 1, green corresponds to Group 2, and yellow corresponds to Group 3.

2.2 Validation of Clusters

To test whether these clustered sub-types provide additional insight into the health of the patients, we performed several statistical comparisons of each patients' characteristics at baseline across all four sub-types plus healthy patients, to determine if there were any statistically significant differences. The results and values of these comparisons are presented in Table 1 below.

As seen in Table 1, many of the key screening measurements of the populations from the different clusters are significantly different, implying our clusters are informative about the health of individuals. In particular, we note that Group 0—which corresponds to moderate physical symptoms with cognitive decline—tends to be younger on average then the other groups, indicating this group may contain many more individuals with early onset PD. Moreover, the sub-types vary substantially in their sleep score and olfactory evaluation,

		Group 0	Group 1	Group 2	Group 3	Healthy	p value
Lymphocytes		1.643^{m}	1.749	1.642^{n}	1.704^{p}	1.850^{mnp}	0.01
REM Sleep Score		5.549^{de}	1.892^{dfgh}	5.969^{fij}	5.087^{gik}	3.247^{ehjk}	<0.001
UPDRS part II		6.594	6.482	7.981	3.272	N/A	<0.001
UPDRS part III		23.654	21.277	26.503	15.382	N/A	<0.001
Schwab & England Score		92.256	91.506	91.321	96.214	N/A	<0.001
Age		58.925^{a}	60.446	62.912^{abc}	58.387^{b}	59.571^{c}	0.02
	Anosmia	46	10	57	41	6	<0.001
Olfactory	Hyposmia	68	11	91	98	68	
	Normosmia	19	5	11	34	122	
Race White		95.49%	93.98%	94.34%	94.22%	94.37%	0.99
Gender Male		67.67%	57.83%	65.41%	63.01%	65.80%	0.63
Geriatric Depression Score		5.391	5.069	5.270	5.231	5.168	0.68

Table 1: Comparison of baseline and screening measurements between clusters. p-values labeled in the table represent difference between all groups, and significant pairwise comparisons using a two sample T-test are marked by superscripts with p-values a-0.008; b-0.001; c-0.02;d,e,f,g,h,i,j,k-<0.001, m-0.003;n-0.004;p-0.04

which are both measures that have previously been shown to be strong indicators of PD [Rao et al., 2006] indicating that these progression sub-types are sensitive to these important predictors.

Overall, the comparisons shown in Table 1 show that our data driven clusters are not only informative when comparing different forms of disease progression, but also correspond to variations in screening measurements. Based on this analysis, we believe that using screening data to predict these clusters could lead to clinically significant insights that can help with treatment.

3 Local Explainer Algorithm

After identifying the four disease progression sub-types, we would like to predict which kind of disease progression an individual might experience, given measurements collected during a screening visit. As we will show in our experiments in Section 4, this task is best performed by complex black box models such as artificial neural networks (ANN) and bagged forests. This means that while the prediction may be accurate, it will not be easily explained, which make such models difficult to use for diagnosis recommendations. Our goal is to instead develop a method that trains simple auxiliary explainer models, and can still accurately describe the relationship between the data and the model output within a small region of a given prediction.

This methodology is known as *training local explainer models* and has been shown to be useful in understanding black box predictions [Ribeiro et al., 2016, 2018]. One of the key tradeoffs in generating model explanations is that of *fidelity*—how well the explainer approximates the black box model—and *interpretability*—how easy it is for a practitioner to trace the predictions of the model. In contrast to previous literature which has proposed the use of regularization to achieve this goal, we propose directly computing locally significant features using an information filter. Generally, computing such filters can be computationally expensive and requires the use of numerical integration; however, one of our main contributions in this paper is to introduce an efficient algorithm for filtering out less significant features. This methodology will allow us to train local explainers that are significantly less complex then those that use regularization, with better fidelity.

3.1 Notation

Before proceeding to our discussion on the local explainer method, we will first establish some technical notation. We assume that for each patient $i = 1, ..., n$ we have an ordered pair (x_i, y_i), where $x_i \in \mathcal{X} \subseteq \mathbb{R}^m$ are the features values of the patient and $y_i \in \mathcal{L} \subseteq \mathbb{Z}$ is the corresponding class label generated by a black box model f. Through our analysis we will also refer to this set of points through matrix notation where $X \in \mathcal{X}^n \subseteq \mathbb{R}^{m\times n}$ is the feature value matrix and $y \in \mathcal{L}^n \subseteq \mathbb{Z}^n$ is the vector of class labels, where each row in these matrices corresponds to a single patient's data. For our analysis we assume that $\mathcal{X}$ is a compact set. Let $\Phi = \{1, ..., m\}$ be the set of features, and it may also be used to denote the index set of the features. This set can be partitioned into two sets $\Phi_c, \Phi_b \subseteq \Phi$ that represent the set of continuous and binary features respectively.

Furthermore we define the set-valued function $\Phi^* : \mathcal{X} \to \Phi$ as the function which extracts the minimum set of necessary features to accurately predict the class of a point x. Namely,

$$\Phi^*(x) = \underset{\varphi\subseteq\Phi}{\arg\min}\{|\varphi| : p(y|x) = p(y|x[\varphi])\}, \tag{1}$$

where $x[\varphi]$ is an indexing operation that maintains the values of x but only for the features in φ, and p is the conditional probability mass function of the labels y given the observation of some features. Specifically, if a feature index is not included $\Phi^*(x)$, then it is not required to understand the particular label of x. In addition, we will denote the ball around a point x of radius r with respect to a metric d as $\mathcal{B}(x, r, d)$.

Finally, a key feature of the explainer training method we propose includes the use of *mutual information*. In information theory, mutual information is a quantity that measures how correlated two random variables are with one another. If X, Y are two random variables with joint density p and marginal densities p_x, p_y, then the mutual information between X and Y is denoted $I(X; Y)$ and calculated as:

$$I(x; y) = \mathbb{E} \log \frac{p(X, Y)}{p_x(X), p_y(Y)} = \int_x \int_y p(x, y) \log \frac{p(x, y)}{p_x(x), p_y(y)} dxdy. \tag{2}$$

If X and Y are independent then $I(X; Y) = 0$; otherwise $I(X; Y) > 0$, meaning that X contains some information about Y. A similar quantity can be computed using a conditional distribution on another random variable Z, known as the *conditional mutual information* and denoted $I(X; Y|Z)$.

3.2 Local Explainer Algorithm Description

Our main local explainer algorithm extends previous local explainer methods such as LIME [Ribeiro et al., 2016] by restricting the sampling region around the prediction, and including an information filter to ensure that fewer features are included in the final explainer mode.

Our general local explainer is formally presented in Algorithm 1, but we will give a brief overview of its operations here. The algorithm takes in hyper-parameters including number of points N to be sampled for training the explainer, a distance metric d, and a radius r around the point $\bar{x}$ being explained. First the algorithm samples N points uniformly from within a r radius of $\bar{x}$; we call this set of points $T(\bar{x})$. Depending on the distance metric being used this can often be done quite efficiently, especially if the features are binary valued or an ℓ^p metric is used [Barthe et al., 2005]. Then using the sampled points, the algorithm uses the Fast Forward Feature Selection (FFFS) algorithm as a subroutine (formally presented in Section 3.3 and Appendix A), which uses an information filter to remove unnecessary features and reduce the complexity of the explainer model. The FFFS algorithm uses an estimate of the joint empirical distribution of $(T(\bar{x}), f(T(\bar{x}))$ to select the most important features for explaining the model's predictions in the given neighborhood. We denote this set of features $\hat{\Phi}$. Then, using these features and the selected points, the explainer model g is trained by minimizing an appropriate loss function that attempts to match its predictions to those of the black box model. In principle a regularization term can be added to the training loss of explainer g. However, through our empirical experiments in Section 4 we found that FFFS typically selected at most five features, so even the unregularized models where not overly complex.

Algorithm 1 Local Explainer Training Algorithm

```
Require: sampling radius $r$, number of sample points $N$, black box model $f$, data point to be explained $\bar{x}$,
    and loss function $L$ for the explainer model $(\bar{x}, \bar{y})$
1: Initialize $T(\bar{x}) = \emptyset$
2: for $j = \{1, ..., N\}$ do
3:     Sample $x \sim U(\mathcal{B}(\bar{x}, r, d))$
4:     $T(\bar{x}) \leftarrow T(\bar{x}) \cup x$
5: end for
6: Obtain $\hat{\Phi}(\bar{x}) = \text{FFFS}(T(\bar{x}), \Phi, f)$
7: Train $g = \arg\min_{\hat{g} \in \mathcal{G}} \{\sum_{x \in T(\bar{x})} L(f(x) - \hat{g}(x[\hat{\Phi}]))\}$
8: return g
```

3.3 Fast Forward Selection Information Filter

A key step in our algorithm is the use of a mutual information filter to reduce the number of features that will be included in the training of the local explainer. Mutual information filters are commonly used in various signal processing and machine learning applications to assist in feature selection [Brown et al., 2012]. However, these filters can be quite challenging to compute depending on the structure of the joint density function of the features and labels, and can require the use of (computationally expensive) numerical integration. We counteract this by considering an approximation of the density function, using histograms to calculate continuous features. When multiple combinations of features need to be considered as in our setting, the problem of finding the maximum-information minimum-sized feature set is known to be computationally infeasible [Brown et al., 2012]. As such, our proposed method for computing the filter includes a common heuristic known as *forward selection*, which essentially chooses the next best feature to be included in the selected feature pool in a greedy manner. Using this method alone would still require recomputing the conditional distribution of the data based on previously selected features, which can result in long run times for large N. However, using some prepossessing techniques, we can show that these quantities can be stored efficiently using a tree structure, which allows quick computation of the filter.

The general idea of the FFFS algorithm is to consider the feature selection process as a tree construction. Part of this construction relies on an estimate of the empirical density of the features as a histogram with at most B bins and preprocessed summary tensor $M \in \{0,1\}^{B \times |\Phi| \times N}$ which indicates which bin of the histogram a feature value for a particular data point lays in. For each entry, $M[b, \varphi, x] = 1$ if the value of feature φ at point x falls in the bin b. Otherwise, $M[b, \varphi, x] = 0$. The depth of the tree represents the number of selected features and each node of the tree is a subset of $T(\bar{x})$. For instance, at the beginning of the selection process, we have a tree with exactly one node R where $R = T(\bar{x})$. Assume binary feature φ_1 is selected in the first round. Then two nodes a, b are added under R, where $a = \{x_j : M[1, \varphi_1, j] = 1\}$ and $b = \{x_j : M(2, \varphi_1, j) = 1\}$. In the second round, we use the partition sets a, b to compute the mutual information instead of the complete set R. The set a is used for computing $\hat{p}(\varphi|\varphi_1 = 1)$, $\hat{p}(y|\varphi_1 = 1)$, and $\hat{p}(\varphi; y|\varphi_1 = 1)$, while b is used when the condition is $\varphi_1 = 2$. In each round the leaves $\mathcal{L}$ of the current tree represent the set of partition sets corresponding to all random permutation of selected features information. Therefore, $\mathcal{L}$ provides us sufficient information for calculating the desired mutual information. As shown in Algorithm 4, the algorithm only outputs the leaves $\mathcal{L}$, not the entire tree. The main algorithmic challenge is to efficiently calculate the marginal distributions $(\hat{p}(\varphi|S), \hat{p}(y|S)$ and joint distribution $\hat{p}(\varphi; y|S)$, which we are able to do using the tree structure.

The detailed structure of the FFFS algorithm used to compute the filtered feature set $\hat{\Phi}$ requires several subroutines, and the formal algorithmic construction for computing the filter is presented across Algorithms 2, 3, 4, and 5. The main FFFS algorithm is Algorithm 2, and it calls the subroutines for recursion (Algorithm 3), selecting features (Algorithm 4), and partitions (Algorithms 5). Formal presentation of these algorithms, as well as detailed descriptions, are given in Appendix A.

4 Experimental Validation of Local Explainer

In this section we empirically evaluate the quality of our local explainer methodology by first showing that accurate sub-type predictions of our PD sub-type clusters (as described in Section 2) can be achieved using black-box methods applied to the data of individuals measured during the screening visit. We then apply our local explainer methodology developed in Section 3 to explain the predictions given by these black-box models.

Our clusters were derived from longitudinal measurements of the four metrics of disease severity described in Section 2.1, measured across the first seven visits in the study over a period of 21 months. Treating these cluster (and the healthy patients) as our ground truth class labels, we first train black box machine learning models to predict which of these progression sub-types an individual will most likely experience given her screening data. This is meant to model the data available to a physician when she must make treatment decisions for a new patient. From screening data in the PPMI data set, we included the following 31 features: PTT, Lymphocytes, Hematocrit, Eyes, Psychiatric, Head-Neck-Lymphatic, Musculoskeletal, Sleep Score, Education Years, Geriatric Depression Score, Left Handed, Right Handed, Gender Male, Female Childbearing, Race White, Race Hispanic, Race American Indian, Race Asian, Race Black, Race PI, Anosmia, Hyponosmia, Normosmia, MRI Normal, MRI Abnormal Insignificant, MRI Abnormal Significant, BL/SC UPDRSII, BL/SC UPDRSIII, BL/SC MOCA, BL/SC MSES, and BL/SC Age. Among these 31 features, 20 features are binary variables and 11 features are continuous variables.

For accurate sub-type predictions using this data, in Section 4.1 we trained three machine learning prediction models: one interpretable model (logistic regression) and two complex black box models (a feed forward ANN and a bagged forest). Our results indicate that the black box models outperform the simpler model, which necessitates the use of a local explainer method for this application to achieve both accurate classification and explainability.

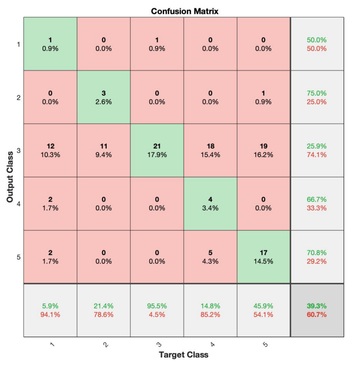
In Section 4.2 we computed local explanations based on the random forest model predictions (which was the model with the highest accuracy) using our proposed FFFS method with the information filter and a local explainer method. This is analogous to LIME [Ribeiro et al., 2016] which does not contain an information filter. Our results show that given a requirement of high explainer fidelity, the use of the information filter will result in less complex explainer models. All experiments described in this section were run on a laptop computer with a 1.2GHz Intel Core m3-7Y32 processor and MATLAB version R2019a with the machine learning and deep learning tool kits [MATLAB, 2010].

4.1 Machine Learning Models for Cluster Prediction

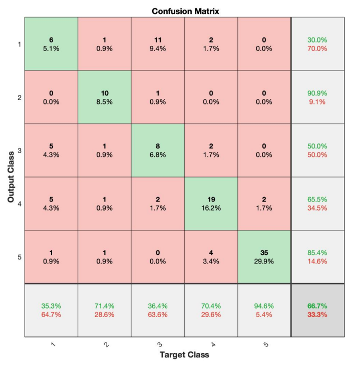
We considered three different kinds of machine learning models for the task of predicting the progression cluster: logistic regression, feed forward ANN, and a bagged forest model. The patient data was split into training, validation, and testing sets with 70% of the data used for training, 15% for validation, and 15% for testing. Among 779 patients, 545 patients were selected for training, and 117 patients were selected for validation and testing.

Since bagged forests and ANNs are sensitive to hyperparamter settings, we used cross-validation to set their respective hyperparamters. Using cross validation and MATLAB's hyperparemeter optimization methods we found that the most effective ANN architecture for our task was with a single hidden layer containing one hundred hidden ReLu units. For the random forest model, we found that an ensemble of 50 bagged trees gave the best results compared to other forest sizes.

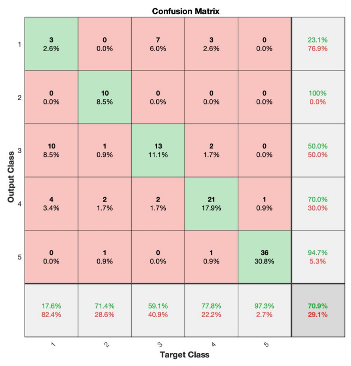
Figures 3 and 4 show the performance of the models on the same training, validation, and testing sets. In both figures, the classes 1-4 correspond to Groups 0-3, and class 5 corresponds to healthy patients (which



(a) Logistic Regression

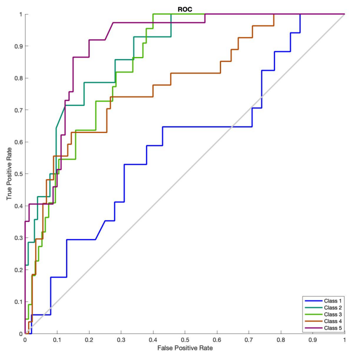


(b) Neural Network

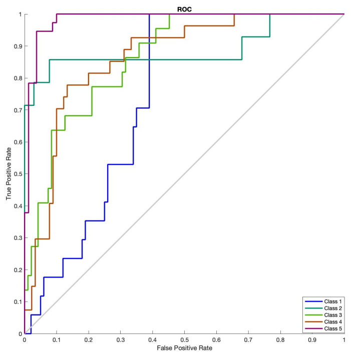


(c) Random Forest

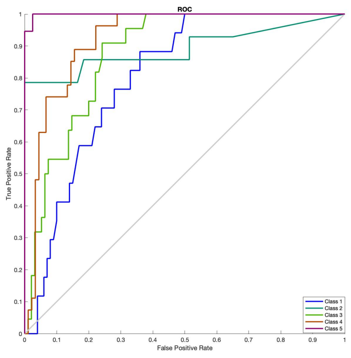
Figure 3: Confusion Matrices



(a) Logistic Regression



(b) Neural Network



(c) Random Forest

Figure 4: ROC Curves

we will also call Group 4). Figure 3 contains the confusion matrix for each model. The rows of the matrix are the *output class*, which represents the predicted class, and the columns of the matrix are the *target class*, which is the true class. The cells on the diagonal of the matrix count accurate predictions. Each cell in the rightmost column has two values: the top number is the percentage of patients that are correctly predicted to each class, and the bottom number is the percentage of patients that are incorrectly predicted to each class. For each cell on the bottom row, the top number is the percentage of patients that belong to each class and is correctly predicted, and the bottom number is the percentage of patients that are incorrectly predicted. For the rest of cells in the matrix, the number in each cell counts for the number of patients that fall in this observation. The cell at the bottom right corner of each matrix shows the total percentage of patients that were correctly and incorrectly predicted.

As shown in Figures 3 and 4, the logistic regression model under-performs relative to the ANN and bagged forest models. Even though the bagged forest model has a lower prediction rate for Group 0 compared to the ANN, it has equal or higher rates of accurate prediction for the other classes. Additionally, the bagged forest model consistently performed better than the ANN and logistic regression models in our experiments. We concluded from these results that the bagged forest classification model is the most effective for our prediction task, and we chose to consider its predictions when evaluating our local explainer method.

4.2 Local Explainer Validation

Since the main difference between our local explainer training algorithm and those in the literature is our use of the FFFS information filter, our experiments on the local explainer are focused on validating the effectiveness of using this information filter. We compare the performance of our local explainer training algorithm to a similar algorithm without a filtering step. We then compare the performance of these methods in terms of explainer complexity and fidelity, across different sampling radii and across all patients.

For the sampling parameters of our algorithm, we sampled $N = 10,000$ points centered around each patient within a radius r of either 3, 7, 11, or 15. The distance metric for computing this radius was a combination of the ℓ_∞ norm for the continuous features and the ℓ_1 norm for the binary features. The continuous value feature of each of the points was sampled uniformly using standard techniques (Barthe et al. [2005]). For binary valued features, we randomly chose at most r binary features and flipped their values. We first randomly generated an integer k between 0 and r, and randomly selected k binary features which we then flipped from their current value (that is, values of 1 were set to 0 and vice versa). To compute probability density estimates, we found that the method performed well with histograms with only three bins for continuous features and two bins for binary features. Intuitively three bins allows us to categorize feature values as low, medium, or high relative to their range.

For both training methods, we chose to train decision trees as our the local explainer class because these have been shown to be ergonomically suitable for explaining black box models in healthcare contexts [Bastani et al., 2018]. Then we computed the corresponding *fidelity score*, defined as the percentage of data where the prediction of the decision tree matched the prediction of the random forest model. We used the number of leaves on the decision tree as a measure of the explainer complexity.

In Figure 5, we compare the explainer complexity and fidelity level of the explainers generated by the two different training methodology across the four different tested sampling radii. Unsurprisingly, when the sampling radius is small (i.e., $r = 3$), there is not much advantage to using the information filter in terms of reducing model complexity for a given fidelity level. Since all points are sampled so closely together, the relevant features are easily learned in explainer training. Conversely, when the sampling radius is large ($r = 15$), the addition of the information filter only helps slightly. With such a large radius, sampling feature values that are far from the point that is meant to be explained may not give useful information for that prediction. However, when considering the medium radius ranges, for high levels of fidelity, the inclusion of the information filter provides simpler models across the board. In particular, consider the plots corresponding for local explainer radius of $r = 7$ and $r = 11$ in Figure 5. Note that in both of these figures, when considering high fidelity explainers generated by both methods (fidelity ≥ 0.6), the explainers generated by the information filter method are less complex then those generated without the filter. This would indicate that using our information filter, we can obtain high fidelity local explainers that are on average less complex then those generated without this filter. When considering low fidelity explainers, the no filter method creates less complex models then the filter method. This is because our filter method is better equipped to find relevant features even in more complex regions of the black box model, while the no filter method is unable to learn these regions effectively with a fixed sample size. This is significant since this would indicate that our proposed methodology is able to explain a larger portion of the feature space using less complex models while still finding meaningful features for explanations, relative to existing methodologies.

Overall, the plots in Figure 5 show that incorporating an information filter into local explainer training can be more effective in extracting relevant features then using regularization, and can generate less complex models with high fidelity. In addition, these results indicate that using an information filter allows for local explainers with information filters to obtain higher fidelity over a larger radius with relatively less complex models. This is particularly significant since less complex models can be me more easily interpreted by domain experts, making it easier for them to translate the clinical significance of the black box model outputs. while larger explanation radii are useful for model validation and generalization of explanations. Moreover, even in complex decision regions generated by the black box model, using an information filter in conjunction with local explainers is better at extracting relevant features for predictions which again can be useful for model validation and providing clinical insights.

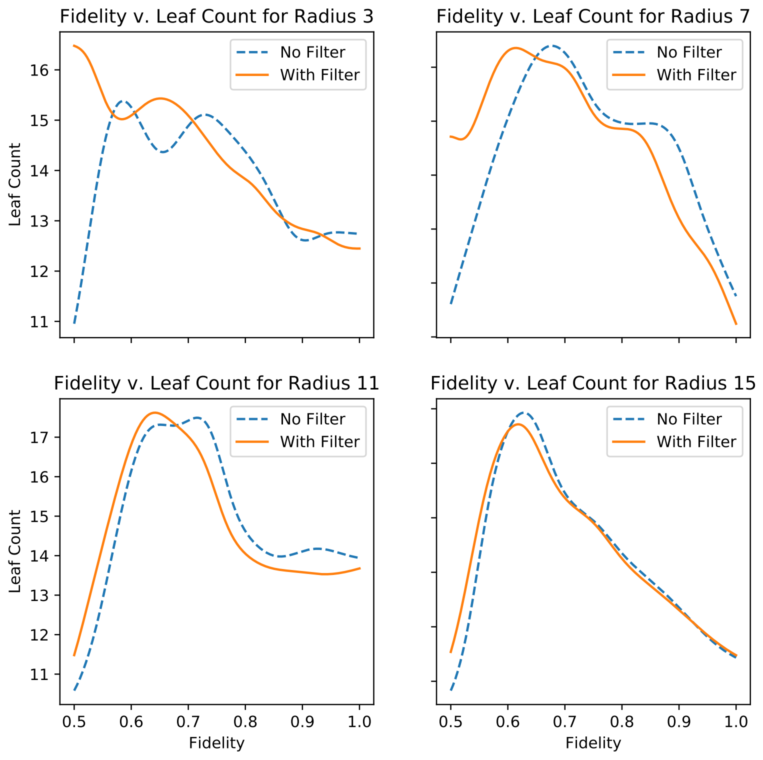


Figure 5: Comparison of local explainer algorithm with the information filter (solid line) and without the the information filter (dashed line) for various different radius settings for the algorithms. The x-axis corresponds to the given fidelity score of the model and the y-axis measures the complexity of the decision tree explainer by the number of leaves. For a small radius ($r = 3$) and large radius ($r = 15$), the addition of an information filter does not lead to a significant difference in model complexity across all levels of fidelity. However, using the information filter in explainer training for moderate sized radii ($r = 7$ and $r = 11$) results in less complex models at higher levels of fidelity (> 0.6).

5 Global Explainer Methodology

While our proposed local explainer method is useful for providing insight into the behaviour of the black box model in a restricted region of the feature space it cannot be used for total model validation. For this task we require a global explainer that could provide insights into the behavior of the model across the entirety of the feature space. However, an explainer that is constrained to explain all of the feature space is likely have low fidelity since the explainer model is less complex then the black box. This introduces a trade-off between two qualities of an explainer model: its *coverage* and *fidelity*. One way to address this challenge is to create a global explainer model by aggregating several local explainer models. Several existing approaches use this idea [Ribeiro et al., 2016] by formulating the construction of the global explainer as an optimization problem. Generally, this problem is framed as maximizing the total coverage of the explainer subject to a constraint on the total number of local explainers included in the aggregate. Solving this problem is conjectured to computationally intractable [Ribeiro et al., 2016], and therefore in existing work it is solved using heuristics. In this section, we formulate the problem of constructing the aggregate explainer explicitly as an integer linear program that allows us to directly trade off coverage and fidelity.

5.1 Mathematical Programming Formulation of Aggregation Problem

In order to formulate the optimization problem for the global explainer, we first need to formally define the concepts of coverage and fidelity. Building upon the notation from Section 3.1, let $g_{i,r} : \mathcal{X} \to \mathcal{L}$ be the local explainer generated by using Algorithm 1 on patient i with radius r. Furthermore let $\mathcal{X}_{i,r} \subset \mathcal{X}$ be defined as the set of points explained by explainer $g_{i,r}$; that is $\mathcal{X}_{i,r} := \{x \in \mathcal{X} : \|x - x_i\| \leq r\}$. Define γ as the aggregate set of all explainers generated in this process: $\gamma = \{g_{1,r_1}, g_{2,r_2}, ..., g_{n,r_n}\}$ for some potential local explainers $g_{1,r_1}, g_{2,r_2}, ..., g_{n,r_n}$. Using these quantities we define the *coverage of aggregate exmplainer* γ on the data set $\mathcal{X}$ as the total number of points that are covered by the explanation radius of at least one explainer contained in γ. We denote the coverage as:

$$\text{Cov}(\gamma, \mathcal{X}) = \sum_{x \in \mathcal{X}} \max_{i \in \{i : g_{i,r} \in \gamma\}} \mathbb{1}[x \in \mathcal{X}_{i,r}]. \tag{3}$$

Next recall that the fidelity of a single local explainer can be defined as the accuracy of that explainer using the predicted labels of the black box model as the ground truth. We will define the *fidelity of aggregate explainer* γ on the data set $\mathcal{X}$ as the minimum of the fidelity obtained by each individual local explainer. We denote this as:

$$\text{Fidelity}(\gamma, \mathcal{X}) = \min_{\{i : g_{i,r_i} \in \gamma\}} \frac{1}{|\mathcal{X}_{i,r}|} \sum \mathbb{1}[g_{i,r}(x) = f(x)] \mathbb{1}[x \in \mathcal{X}_{i,r}]. \tag{4}$$

Although the fidelity of γ can be defined as the average of the fidelities of its component explainers, we believe using the minimum fidelity gives a stricter measure on how well the global explainer captures the behavior of the black box model.

Let K denote the budget of the maximum number of explainers that can be contained in γ, and let φ be the minimum fidelity we would like the aggregate explainer to obtain. Then the optimization problem to be solved can be written as:

$$\max_{\gamma} \{\text{Cov}(\gamma, \mathcal{X}) : \text{Fidelity}(\gamma, \mathcal{X}) \geq \varphi, |\gamma| \leq K\}. \tag{5}$$

5.2 Reformulation as Integer Program (IP)

Note that as written, optimization problem (5) is not trivial to solve, and in particular could require enumerating all possible subsets γ of local explainers. To address this challenge, we propose reformulating problem (5) as an Integer Program (IP) that can be solved using current commercial software. To do this, we first define the three different sets of binary variables that we will call w_i, y_j, z_{ij}. Let w_i be a binary variable that is equal to 1 if explainer $g_{i,r_i} \in \gamma$. That is, $w_i = \mathbb{1}[g_{i,r_i} \in \gamma]$. Let y_j be defined as a binary variable that is equal to 1 if point j is covered by the global explainer γ. That is $y_j = \mathbb{1}[x_j \in \cup_{\{i : g_{i,r_i} \in \gamma\}} \mathcal{X}_{i,r_i}]$. Finally, let z_{ij} be a binary variable that is equal to 1 if explainer $g_{i,r_i} \in \gamma$ covers point x_j. That is, $z_{ij} = \mathbb{1}[x_j \in \mathcal{X}_{i,r_i}]$. We can now define the coverage and fidelity of aggregate explainer γ in terms of these three sets of variables.

Proposition 1. *Cov*$(\gamma, \mathcal{X})$*, the coverage of local explainer set* γ *on data set* $\mathcal{X}$*, can be expressed with the following set of integer variables and constraints:*

$$\begin{aligned} Cov(\gamma, \mathcal{X}) = & \sum_{j \in \mathcal{X}} y_j, \\ s.t. \quad & z_{ij} \leq w_i, \quad i, j \in \mathcal{X}, \\ & y_j \geq z_{ij}, \quad i, j \in \mathcal{X}, \\ & y_j \leq \sum_{i \in \mathcal{X}} z_{ij}, \quad j \in \mathcal{X}, \\ & \|x_i - x_j\| z_{ij} \leq r_i, \quad i, j \in \mathcal{X}. \end{aligned} \tag{6}$$

Proof. Recall the definition of $\mathrm{Cov}(\gamma, \mathcal{X})$ as given in Equation (3). We will directly reconstruct this definition using the binary variables defined above. First note that through a simple direct substitution we obtain $\mathrm{Cov}(\gamma, \mathcal{X}) = \sum_{x \in \mathcal{X}} \max_{i \in \{i : g_{i,r} \in \gamma\}} z_{ij}$. Since taking the maximum of binary variables is equivalent to the Boolean OR operator, we see that $y_j = \max_{i \in \{i : g_{i,r} \in \gamma\}} z_{ij}$, which provides us with the first equality. The next two inequalities directly models that explainer g_{i,r_i} can only explain point x_j if it is included in γ, which is a standard way of modeling conditional logic in IP [Wolsey and Nemhauser, 1999]. The next two constraints come from modeling the Boolean OR operator using integer constraints [Wolsey and Nemhauser, 1999]. The final constraint ensures that a point x_j can be covered by an explainer g_{i,r_i} if $x_j \in \mathcal{X}_{i,r_i}$, thus preserving the local region of the local explainer. □

Next we consider the average fidelity constraint.

Proposition 2. *The constraint Fidelity*$(\gamma, \mathcal{X}) \geq \varphi$ *can be modeled using the following set of integer linear constraints:*

$$\begin{aligned}
&\|x_i - x_j\| z_{ij} \leq r_i, \quad i, j \in \mathcal{X}, \\
&z_{ij} \leq w_i, \quad i, j \in \mathcal{X}, \\
&\sum_{j \in \mathcal{X}} \left(\mathbb{1}_{\{f(x_j) = g_{i,r_i}(x_j)\}} - \varphi\right) z_{ij} \geq 0, \quad i \in \mathcal{X}.
\end{aligned} \tag{7}$$

Proof. The first two constraints ensure proper local behavior of the local explainer as in Proposition 1. Thus we will focus the derivation of the final constraint. Using the definition of $\mathrm{Fidelity}(\gamma, \mathcal{X})$ in Equation (4) and directly substituting variables for indicators, we can express the lower bound constraint as,

$$\min_{\{i : g_{i,r_i} \in \gamma\}} \frac{1}{|\mathcal{X}_{i,r}|} \sum_{j \in \mathcal{X}} \mathbb{1}[g_{i,r_i}(x_j) = f(x_j)] z_{ij} \geq \varphi.$$

Note that if the minimum over all explainers g_{i,r_i} must have fidelity of at least φ, then every local explainer must also have fidelity at least φ. This allows us to disaggregate this constraint across all $i \in \mathcal{X}$. Let us consider the constraint for a single local explainer g_{i,r_i}. Using the definition of z_{ij}, we note that $|\mathcal{X}_{i,r_i}| = \sum_{j \in \mathcal{X}} z_{ij}$. Thus the new lower bound fidelity constraint for a single explainer can be written as:

$$\frac{\sum_{j \in \mathcal{X}} \mathbb{1}[g_{i,r_i}(x_j) = f(x_j)] z_{ij}}{\sum_{j \in \mathcal{X}} z_{ij}} \geq \varphi. \tag{8}$$

Note that the denominator of the left hand side can only be zero when the numerator is also zero because $\sum_{j \in \mathcal{X}} z_{ij} \geq \sum_{j \in \mathcal{X}} \mathbb{1}[g_{i,r_i}(x_j) = f(x_j)] z_{ij}$. This means that we can multiply both sides of the inequality by the sum $\sum_{j \in \mathcal{X}} z_{ij}$ while still maintining its validity. Distributing φ and combining like terms gives us with the form of the constraint presented in the proposition statement. □

We can then use these expressions to for coverage and fidelity to re-write our optimization problem as an integer program that can then be solved using commercial solvers.

Proposition 3. *The optimization problem in* (5) *can be written as the following integer program:*

$$\begin{aligned}
\max \sum_{j\in\mathcal{X}} & y_j, \\
s.t.\quad & z_{ij} \le w_i, \quad i,j\in\mathcal{X}, \\
& y_j \ge z_{ij}, \quad i,j\in\mathcal{X}, \\
& y_j \le \sum_{i\in\mathcal{X}} z_{ij}, \quad j\in\mathcal{X}, \\
& \|x_i - x_j\| z_{ij} \le r_i, \quad i,j\in\mathcal{X}, \\
& \sum_{j\in\mathcal{X}} \big(\mathbb{1}_{\{f(x_j)=g_{i,r_i}(x_j)\}} - \varphi\big) z_{ij} \ge 0, \quad i\in\mathcal{X}, \\
& \sum_{i\in\mathcal{X}} w_i \le K, \\
& y_j, w_i, z_{ij} \in \{0,1\} \quad i,j\in\mathcal{X}.
\end{aligned} \tag{9}$$

Proof. The objective function and first five constraints come directly from Propositions 1 and 2. The next constraint comes using the definition of w_i and direct substitution to obtain that $|\gamma| = \sum_{i\in\mathcal{X}} w_i$, which is then used to rewrite the budget constraint from (5). The final constraint ensures that our new variables are binary integers. □

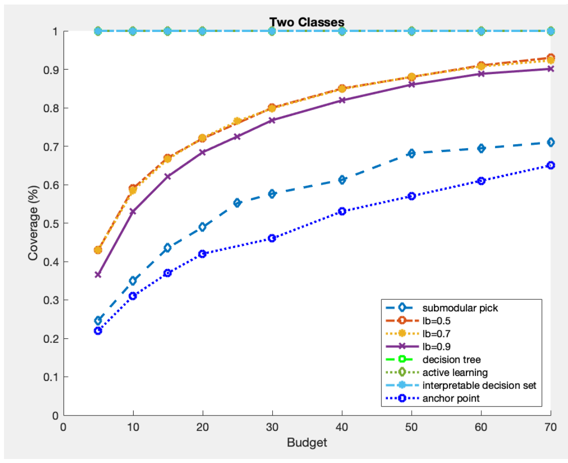
6 Experimental Validation of Global Explainer

In this section we empirically validate the quality of our global explainer methodology using the PPMI dataset described in Sections 2 and 4. In our experiments, we first use the clustering algorithm of Section 2 to assign labels and then apply our local explainer as described in Section 3. We then use the IP-based global explainer described in Section 5 to choose the local explainers that will comprise our global explainer.

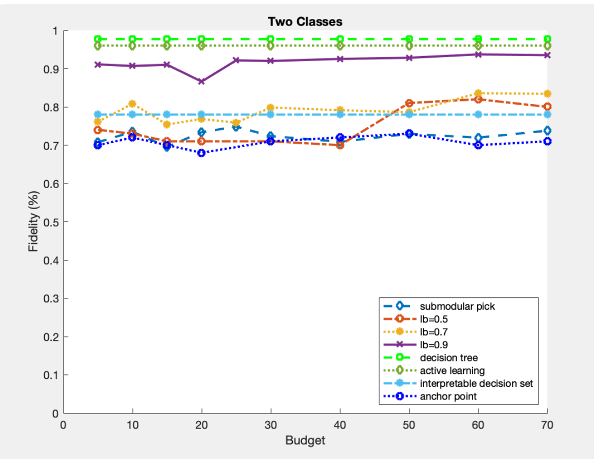
We ran the optimization algorithm for binary classification (2 class) between healthy individuals and patients diagnosed with PD, and for multi-class (5 class) classification among healthy individuals and the four PD sub-types. Figure 6 shows the coverage and average fidelity score of the global explainer generated by our IP-based approach, compared with other existing local and global explainers methods as a function of the budgeted number of local explainers that are allowed in the global explainer. We tested performance for budgets $K = 5, 10, 15, 20, 30, 40, 50, 60, 70$. The lines labeled lb=0.5,0.7 or 0.9 in Figure 6 represent the result of the IP with the lower bound of the fidelity score set to 50%, 70% or 90%, respectively. Based on testing different sampling radii (shown above in Figure 5), we found that when the sampling radius was in a medium range (i.e., $r = 7$ or 11) our method produced a simpler model with same precision than those produced by existing methodologies. Therefore we used sampling radius $r = 11$ in our experiments for all local explainers.

We compared our IP with the following prior methods: the submodular pick algorithm from [Ribeiro et al., 2016], a simple decision tree trained on the labels of the black box model Friedman et al. [2001], an extracted decision tree method method [Bastani et al., 2018], an interpretable decision set [Lakkaraju et al., 2016], and an anchor points method [Ribeiro et al., 2018]. The interpretable decision set, decision tree, and extracted approaches are all global explainer methods, so they are always guaranteed have perfect coverage. The submodular pick and anchor point methods, along with our IP method must trade off between coverage and fidelity.

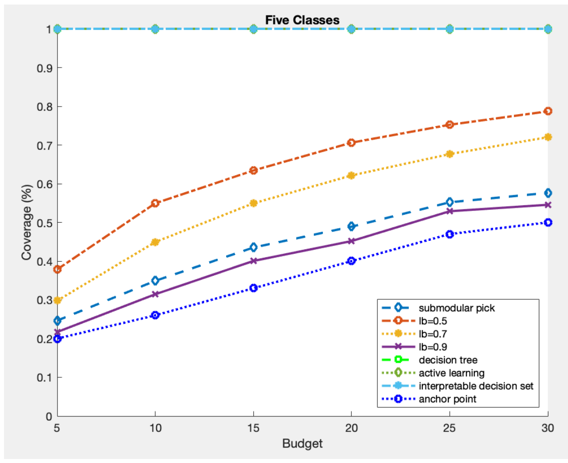
For the simpler classification problem with only two classes, our IP approach, extracted decision tree, and regular decision tree methods all produced a model with significantly higher fidelity score and higher coverage than other methods. Since a large percentage of the patients are healthy individuals, it is reasonable that the decision tree and extracted tree methods achieve higher precision than our IP. However, our IP outperforms the other existing methodologies when the classification problem is more complex (5 class). Even though



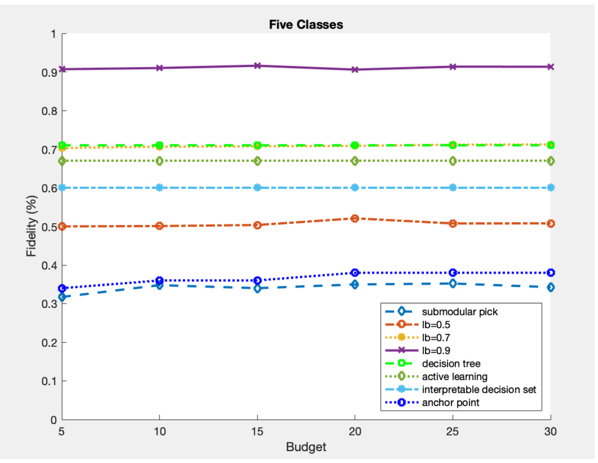
(a) 2 Class Coverage



(b) 2 Class Fidelity Score



(c) 5 Class Coverage

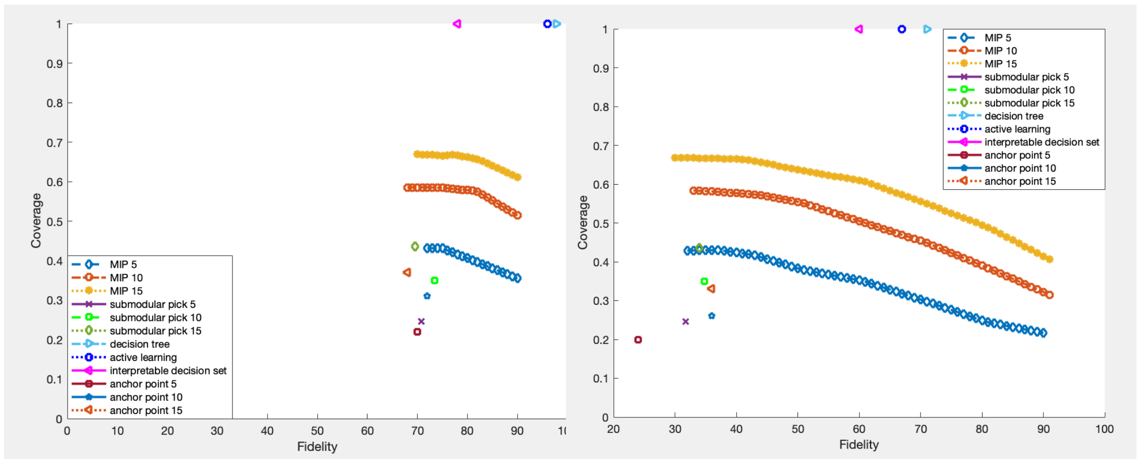


(d) 5 Class Fidelity Score

Figure 6: Performance of IP and other local and global explainer methods on PPMI dataset.

the coverage of IP is lower than the other global explainer methods, the model produced by IP has much higher coverage and precision than the existing local explainers methods in both problems.

Figure 7 shows the relationship between the coverage and fidelity of each method. The lines labeled MIP 5, MIP 10, and MIP 15 respectively show the smooth transition of coverage and fidelity score of our IP approach with fixed budgets of 5, 10, and 15 as the lower bound of the fidelity score varies from 0% to 90% in increments of 10%. Since our IP approach is the only one that has a lower bound constraint on fidelity, we are able to observe this trade-off between the coverage and fidelity for MIP, while the other existing methods lack such freedom.



(a) 2 Classes Fidelity vs. Coverage

(b) 5 Classes Fidelity vs. Coverage

Figure 7: Trade-off between fidelity and coverage for IP and other local and global explainer methods on PPMI dataset. For our IP-based approach, the lower bound on fidelity is varied from 0% to 90%, while the other methods do not exhibit this degree of freedom.

References

Parkinson's Progression Markers Initiative. URL `http://www.ppmi-info.org`.

The Michael J. Fox Foundation for Parkinson's Research, 2019. URL `http://www.michaeljfox.org`.

E. Adeli, F. Shi, L. An, C.-Y. Wee, G. Wu, T. Wang, and D. Shen. Joint feature-sample selection and robust diagnosis of Parkinson's disease from MRI data. *NeuroImage*, 141:206–219, 2016.

A. Aswani, P. Kaminsky, Y. Mintz, E. Flowers, and Y. Fukuoka. Behavioral modeling in weight loss interventions. *European Journal of Operational Research*, 272(3):1058–1072, 2019.

F. Barthe, O. Guédon, S. Mendelson, and A. Naor. A probabilistic approach to the geometry of the ℓ_p^n-ball. *The Annals of Probability*, 33(2):480–513, 2005.

H. Bastani, O. Bastani, and C. Kim. Interpreting predictive models for human-in-the-loop analytics. *arXiv preprint 1705.08504*, 2018.

S. Bhat, U. R. Acharya, Y. Hagiwara, N. Dadmehr, and H. Adeli. Parkinson's disease: Cause factors, measurable indicators, and early diagnosis. *Computers in Biology and Medicine*, 102:234–241, 2018.

L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, 2001.

G. Brown, A. Pocock, M.-J. Zhao, and M. Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13(Jan):27–66, 2012.

D. Castillo-Barnes, J. Ramírez, F. Segovia, F. J. Martínez-Murcia, D. Salas-Gonzalez, and J. M. Gorriz. Robust ensemble classification methodology for I123-ioflupane SPECT images and multiple heterogeneous biomarkers in the diagnosis of Parkinson's disease. *Frontiers in Neuroinformatics*, 12:53, 2018.

R. Erro, C. Vitale, M. Amboni, M. Picillo, M. Moccia, K. Longo, G. Santangelo, A. De Rosa, R. Allocca, and F. Giordano. The heterogeneity of early Parkinson's disease: A cluster analysis on newly diagnosed untreated patients. *PLoS ONE*, 8(8):e70244, 2013.

S.-M. Fereshtehnejad and R. B. Postuma. Subtypes of Parkinson's disease: What do they tell us about disease progression? *Current neurology and neuroscience reports*, 17(4):34, 2017.

S.-M. Fereshtehnejad, S. R. Romenets, J. B. Anang, V. Latreille, J.-F. Gagnon, and R. B. Postuma. New clinical subtypes of Parkinson disease and their longitudinal progression: A prospective cohort comparison with other phenotypes. *JAMA Neurology*, 72(8):863–873, 2015.

S.-M. Fereshtehnejad, Y. Zeighami, A. Dagher, and R. B. Postuma. Clinical criteria for subtyping Parkinson's disease: Biomarkers and longitudinal progression. *Brain*, 140(7):1959–1976, 2017.

L. J. Findley. The economic impact of Parkinson's disease. *Parkinsonism & Related Disorders*, 13:S8–S12, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2001.

J. M. Graham and H. J. Sagar. A data-driven approach to the study of heterogeneity in idiopathic Parkinson's disease: Identification of three distinct subtypes. *Movement Disorders*, 14(1):10–20, 1999.

T. J. Hirschauer, H. Adeli, and J. A. Buford. Computer-aided diagnosis of Parkinson's disease using enhanced probabilistic neural network. *Journal of Medical Systems*, 39:179, 2015.

S. L. Kowal, T. M. Dall, R. Chakrabarti, M. V. Storm, and A. Jain. The current and projected economic burden of Parkinson's disease in the United States. *Movement Disorders*, 28(3):311–318, 2013.

H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1675–1684, 2016.

J. C. Latourelle, M. T. Beste, T. C. Hadzi, R. E. Miller, J. N. Oppenheim, M. P. Valko, D. M. Wuest, B. W. Church, I. G. Khalil, and B. Hayete. Large-scale identification of clinical and genetic predictors of motor progression in patients with newly diagnosed Parkinson's disease: A longitudinal cohort study and validation. *The Lancet Neurology*, 16(11):908–916, 2017.

L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (Nov):2579–2605, 2008.

P. Martínez-Martín, A. Gil-Nagel, L. M. Gracia, J. B. Gómez, J. Martinez-Sarries, F. Bermejo, and C. M. Group. Unified Parkinson's disease rating scale characteristics and structure. *Movement Disorders*, 9(1): 76–83, 1994.

P. Martinez-Martin, C. Rodriguez-Blazquez, and M. J. Forjaz. *Rating scales in movement disorders*. Elsevier, 2017.

MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

Z. S. Nasreddine, N. A. Phillips, V. Bédirian, S. Charbonneau, V. Whitehead, I. Collin, J. L. Cummings, and H. Chertkow. The montreal cognitive assessment, MoCA: A brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699, 2005.

F. L. Pagan. Improving outcomes through early diagnosis of Parkinson's disease. *The American Journal of Managed Care*, 18(7 Suppl):S176–82, 2012.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

B. Peng, S. Wang, Z. Zhou, Y. Liu, B. Tong, T. Zhang, and Y. Dai. A multilevel-ROI-features-based machine learning method for detection of morphometric biomarkers in Parkinson's disease. *Neuroscience Letters*, 651:88–94, 2017.

R. Prashanth, S. D. Roy, P. K. Mandal, and S. Ghosh. High-accuracy detection of early Parkinson's disease through multimodal features and machine learning. *International Journal of Medical Informatics*, 90: 13–21, 2016.

S. S. Rao, L. A. Hofmann, and A. Shakil. Parkinson's disease: Diagnosis and treatment. *American Family Physician*, 74(12):2046–2054, 2006.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, AAAI '18, pages 1527–1535, 2018.

A. Rossi, K. Berger, H. Chen, D. Leslie, R. B. Mailman, and X. Huang. Projection of the prevalence of Parkinson's disease in the coming decades: Revisited. *Movement Disorders*, 33(1):156–159, 2018.

A. Siderowf. *Schwab and England activities of daily living scale*, pages 99–100. Elsevier, 12 2010.

G. Singh, L. Samavedham, E. C.-H. Lim, A. D. N. Initiative, and P. P. M. Initiative. Determination of imaging biomarkers to decipher disease trajectories and differential diagnosis of neurodegenerative diseases (DIsease TreND). *Journal of Neuroscience Methods*, 305:105–116, 2018.

B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

F. Wang and C. Rudin. Falling rule lists. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, AISTATS '15, pages 1013–1022, 2015.

L. A. Wolsey and G. L. Nemhauser. *Integer and Combinatorial Optimization.* Wiley, 1999.

A FFFS Algorithmic Details

In this appendix, we present and discuss the FFFS algorithm used in our local explainer method. The main algorithm is presented in Algorithm 2, and the required subroutines are presented in Algorithms 3, 4, and 5.

Since the main structure of the algorithm requires a recursive tree traversal, Algorithm 2 includes a general prepossessing wrapper algorithm that starts the recursion. In this part of the algorithm, the sampled data points are used to compute the empirical densities of their feature values. These densities are approximated using histograms which can vary in the number of bins. For simplicity of presentation, we assume each histogram has the same bin size, but of course this detail can be modified in implementation. The key addition here is the computation of tensor M, which tracks the inclusion of each data point's features into their respective histogram bin.

Algorithm 3 contains the main recursion of the filter computation, and it outputs the selected features when it terminates. The recursion of Algorithm 3 requires a set of selected features S, a set of unselected features U, the binary tensor M, the black box model predictions Y, and $\mathcal{L}$, which is a set of partition sets of points in $T(\bar{x})$. Since no features are selected prior to the first call to Algorithm 3, we initialize the inputs $S = \emptyset$, $U = \Phi$, $Y = f(T(\bar{x}))$ and $\mathcal{L} = T(x_i)$ when it is first called in Algorithm 2. The recursion terminates and outputs the current set of selected features when either all features are selected or $\mathcal{L}$ becomes empty. If the termination condition is not met, Algorithm 3 calls Algorithm 4, which updates $S, U,$ and $\mathcal{L}$ using a bin expansion. Then Algorithm 3 makes a recursive call with updated inputs and repeat the previous steps.

Algorithm 4 is used to select one feature from the set of unselected features that maximizes the mutual information $I(\varphi; Y|U)$, and to update $\mathcal{L}$ given the current selected feature. We apply forward selection in Algorithm 4. In order to find $\varphi^* = \arg\max_{\varphi \in U} I(\varphi; Y|U)$, we compute $I(\varphi; y|S)$ for each unselected feature φ. The approximated mutual information $I(\varphi; y|S)$ is computed using the following equation [Brown et al., 2012]:

$$I(\varphi; y|S) \approx \hat{I}(\varphi; y|S) = \frac{1}{|T(\bar{x})|} \sum_{i=1}^{N} \log \frac{\hat{p}(\varphi; y|S)}{\hat{p}(\varphi|S)\hat{p}(\varphi|S)}.$$

If $I(\varphi^*; y|S)$ is not positive, then we do not select any new features. If no new feature is selected, we terminate the process by setting $U = \emptyset$, which satisfies the termination condition of Algorithm 3, and the feature selection process will be complete. If $I(\varphi^*; y|S) > 0$, then we can obtain additional information on the prediction by adding φ^* to the set of selected features S and removing it from the set of unselected features U. Algorithm 4 then calls Algorithm 5 to update $\mathcal{L}$ to $\mathcal{L}'$. Algorithm 5 is used to partition each set in $\mathcal{L}$ given current selected feature φ^*. Using the binary tensor M, we can collect the set of bins for φ^*. As an illustrative example of this process, let $B_{\varphi^*} = \{b_1, b_2\}$ and $\mathcal{L} = T(\bar{x}) = \{x_1, x_2, ..., x_p\}$. Assume $x_i^{\varphi^*} \in b_1$ for $i < 5$ and $x_i^{\varphi^*} \in b_2$ otherwise. Then we can partition the set $\{x_1, x_2, ..., x_p\}$ into 2 sets ℓ_1, ℓ_2 s.t. $\ell_1 = \{x_1, ..., x_4\}$ and $\ell_2 = \{x_5, ..., x_p\}$. Next we add sets ℓ_1, ℓ_2 to $\mathcal{L}'$. Since $\mathcal{L}$ contains exactly one set, we finish the partition process, and Algorithm 5 outputs $\mathcal{L}' = \{\{x_1, ..., x_4\}, \{x_5, ..., x_p\}\}$.

Algorithm 2 Fast Forward Feature Selection (FFFS)

```
Require: $T(\bar{x}), \Phi, f$
 1: for $\varphi \in \Phi_c$ do
 2:     Form histogram with bin set $B_\varphi$ and frequencies $\hat{p}_\varphi$
 3: end for
 4: set $M \in |B_\varphi| \times |\Phi| \times N$ as a zero tensor
 5: for $x \in T(\bar{x})$ do
 6:     for $\varphi \in \Phi$ do
 7:         for $b \in B_\varphi$ do
 8:             if $x[\varphi] \in b$ then
 9:                 Set $M[b, \varphi, x] = 1$
10:             end if
11:         end for
12:     end for
13: end for
14: return RecursionFFS($\emptyset, \Phi, M, f(T(\bar{x})), T(\bar{x})$)
```

Algorithm 3 Recursion FFS

```
Require: $S, U, M, Y, \mathcal{L}$
 1: if $U = \emptyset$ or $\mathcal{L} = \emptyset$ then
 2:     return $S$
 3: else
 4:     $[S', U', \mathcal{L}'] =$ SelectFeature($S, U, M, Y, \mathcal{L}$)
 5:     return RecursionFFS($S', U', M, Y, \mathcal{L}'$)
 6: end if
```

Proposition 4. *The time complexity of the FFFS algorithm for a fixed maximum discretization bin size is* $\mathcal{O}(N|\Phi|)$.

Proof. Note that the size of the generated points is given by the input parameter N, and the set of all features is denoted by Φ. First, since the bin sized is fixed as a constant, and the preprocessing step requires

Algorithm 4 Select Feature

```
Require: $S, U, M, Y, \mathcal{L}$
1: $f^* = \arg\max_{f \in U} I(f; Y|U)$
2: if $I(f^*; Y|U) > 0$ then
3:     $U = U \setminus f^*$
4:     $S = S \cup f^*$
5:     $\mathcal{L}'$= BinPartition $(M, \mathcal{L}, f^*)$
6:     return $S, U, \mathcal{L}'$
7: else
8:     $U = \emptyset$
9:     return $S, U, \mathcal{L}$
10: end if
```

Algorithm 5 Bin Partition

```
Require: $M, \mathcal{L}, f^*$
1: Use $M$ to find $B_{f^*}$ s.t. $B_{f^*} = \{b_1, b_2, ..., b_k\}$ is the set of bins for feature $f^*$
2: $\mathcal{L}' = \emptyset$
3: for $\ell \in \mathcal{L}$ do
4:     Partition $\ell$ into smaller sets $\{\ell_1, \ell_2, ...\ell_k\}$ w.r.t $B_{f^*}$: $\ell_i = \{t \in l : t^{f^*} \in b_i\} \forall i \in \{1, ..., k\}$
5:     $\mathcal{L}' = \mathcal{L}' \cup \{l_1, ..., l_k\}$
6: end for
7: return $\mathcal{L}'$
```

a nested **for** loop, the total time complexity of the preprocessing is $\mathcal{O}(N|\Phi|)$. The FFFS algorithm operates as a tree traversal, where the depth of the tree at the final stage corresponds to the number of selected features. In each level of the tree, the mutual information of all points is evaluated using Algorithm 4 and the sets of generated points are partitioned into smaller sets using Algorithm 5, which combined require $\mathcal{O}(N)$ operations. Next, since in the worst case, all features contain positive mutual information on the prediction value of the black box model, the maximum possible tree depth is given by $|\Phi|$. Combining these two facts gives the desired result. $\square$

B Additional figures

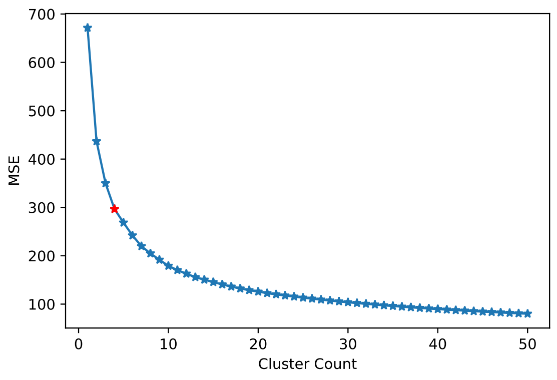


Figure 8: Elbow plot for determining number of clusters to use for k-means clustering. Red marked value is located at 4 clusters and roughly corresponds to the bend in the elbow. The x-axis describes the total number of clusters used in k-means clustering, and the y-axis represents the MSE loss associated with the resulting clusters.