Multi-objective Semi-supervised Clustering using PCA, Agglomerative, KMeans and Explanation using Decision Trees

Kaushik Jadhav
kajadhav@ncsu.edu

Ajith Kumar Vinayakamoorthy Patchaimayil
avinaya@ncsu.edu

Sunandini Medisetti
smedise@ncsu.edu

I. Introduction

In today's world, machine learning models are widely used in various fields, ranging from healthcare to finance, and from social media to self-driving cars. Everyone is focused on building models that have high ”accuracy” and can accurately make predictions or classifications based on given data. Machine Learning is a complex process that involves processing large datasets and many interrelated tasks that need to be performed in a specific order within certain time. However, there are few issues that make Machine Learning as well as Software Development inefficient.

Firstly, in most software development projects, there are tight resource constraints as well as tight schedules to adhere to. Maybe we don't have enough data or maybe not enough time. So the challenge here is, ”Fishing”, that is the process of prioritizing many things, without knowing too much about each thing [2]. We seek a ”big picture overview” rather than lots of details. In other words, if X and Y are sets of decisions and goals, then sampling X is cheap, but Y is expensive. Secondly, the problem with solely focusing on ”accuracy” is that it leads to models that are often considered as ”black boxes”, meaning that the reasoning behind the model's decisions is not transparent or easily understandable [3]. This lack of transparency and interpretability can result in mistrust and skepticism towards these models, especially in sensitive domains such as healthcare or finance.

To solve the Fishing problem, we perform Multi-objective semi-supervised clustering (MSSC). Multi-objective semi-supervised clustering involves clustering data points using multiple objectives and utilizing a small amount of labeled data to guide the clustering process [4]. MSSC aims to simultaneously optimize multiple objectives while incorporating both labeled and unlabeled data. By doing so, MSSC can produce more meaningful and interpretable clustering results that satisfy multiple criteria. Secondly, to solve the transparency problem, researchers have developed a range of Explanation techniques that aim to provide interpretable and meaningful explanations for the model's decision-making processes. We use one such Explanation Algorithm along with MSSC to make a Multi-objective semi-supervised Explanantion system (MSSE).

In this study, we compare a previously implemented baseline Multi-objective Explanantion system, which was based on Recursive Fastmap (sway1) and Bins Contrast Set Learning (xpln1), with our own system that uses PCA (sway2), Agglomerative (sway3) and KMeans (sway4) for clustering and Decision Trees for explanantion (xpln2,3,4). We run the 4 models on 20 iterations of 11 different datasets and evaluate results based on statistics as well as Kruskal-Wallis and Mann-Whitney U tests.

A. Structure of this Paper

1) List of Research Questions:

a) How does MSSC solve the Fishing problem?: Large amounts of data can be approximated by the centroids of a few clusters [5]. Given X decision attributes and Y goal attributes, clusters are typically groupings in the X space

b) How can we optimize clustering while simultaneosuly balancing multiple objectives in MSSC?: At each recursive clustering iteration, we use the Zitzler continuous domination predicate, which aims to find a set of non-dominated solutions that provide a trade-off between different objectives [5].

c) Why Explanation is needed?: MSSC just gives us clusters. But to get meaningful explanations for the model's decision-making process & to learn about ”Rules”, we perform Explanation. MSSE provides models that are not only accurate but also informative & easy to understand for users [3].

d) How can we evaluate and compare different MSSEs in a fair and comprehensive manner?: We can use non-parametric tests like Mann-Whitney U and Kruskal Wallis to determine if the differences in results of the two systems are significant or not. Then to determine which of the two is better, we can use metrics like median or ranks.

2) List of overall contributions:

a) Data Pre-processing: Many of the datasets are unclean and have issues like missing values. The original code of the baseline Fastmap was failing for many datasets and so we had to do things like impute NaNs, '?' by means and label encode text data. After performing data pre-processing, the models started working for all 11 datasets.

b) Comparison of 3 clustering algorithms: We have done a broad comparison of 3 clustering algorithms of PCA (sway2), Agglomerative (sway3) and KMeans (sway4) against our Recursive Fastmap baseline (sway1). After running 20 iterations on 11 different datasets with a total of 33 Y columns, it was observed that sway3, sway4, sway2 were chosen as the BEST algorithm for 17/33, 8/33 and 1/33 while Fastmap (sway1) was only chosen for 6/33 columns.

c) Explanation using Decision Trees: Decision Trees are faster, more flexible and more interpretable than CSL[1]. Instead of using the Bins based Contrast Set Learning (CSL) xpln1 of baseline, we have applied Decision Tree to sway2,3,4 to get xpln2,3,4 of which xpln3 was chosen BEST for 15/33 Y columns. Also, we got much lower sampling tax and explanation tax figures for xpln3 than xpln1.

d) Hyperparameter Optimization (HPO): We did Hyperparameter Optimization (HPO) using 2 methods: 1. Minimal Sampling method and 2. the established optimization method of HyperOpt. Post HPO, not just sway and xpln 2,3,4 but also 1 gave improved results. Also, we ran Hyperopt for 100 evals and noticed that Minimal Sampling method, with just 50 evals gave really good results. Although not as good as Hyperopt, but we were able to get good results with relatively fewer peeks with Minimal Sampling.

e) Replacing Zitzler with Boolean Domination: We attempted to check if Boolean Domination (BDOM) gave better results compared to Zitzler continuous domination predicate. However, during HPO, we found Zitzler still gave better results compared to BDOM and so we continued with Zitzler only for rest of the project.

f) Kruskal-Wallis and Mann-Whitney U: Along with evaluating the models based on means, we have also used Kruskal-Wallis and Mann-Whitney U tests to check that there is significant difference in the model results.

3) Caveats: For this study, we did not explore algorithms like mini-batch KMeans as they would require millions of more rows of data. Also, we did not explore replacing Zitzler or BDOM with aggregation functions. Lastly, we didn't check if sway2,3,4 are taking longer time to run than sway1.

II. Related Work

Fishing is the process of prioritizing many things, without knowing too much about each thing [2]. In the past researchers have attempted to use many techniques to solve the fishing problem. Analytic Hierarchy Process (AHP) is a multi-criteria decision-making technique that involves breaking down a problem into a hierarchy of criteria and sub-criteria, and then using pairwise comparisons to determine the relative importance of each criterion. However, AHP can be time-consuming and requires a lot of input from domain experts, making it difficult to use in practice [8]. In another technique called Weighted Criteria, criteria are assigned different weights based on their importance, and items are scored based on the weighted sum of their criteria scores. However, this technique can be subjective, as the weights assigned to each criterion are based on the opinion of the person or team doing the prioritizing [14]. In one study, KD tree was used to rank data, but had problems with curse of dimensionality [9].

From past work, using a Recursive Fastmap based on Random Projections to perform Multi-objective Semi-supervised Clustering seems to be "state-of-the-art" [6][7]. Fastmap is a near linear time approximation of PCA. In Recursive Fastmap, at each recursive division, we reduce multiple dimensions to one by projecting every point to a line connecting two most distant items. The distance calculation for these Random Projections is done using cosine distance. Random Projection is based on Heuristics, but it still works as we are taking a not very good, but fast algorithm and applying it several times so that down below it is better [10]. Random Projection can be done in two ways: 1. Gaussian Random Projection and 2. Locality Sensitivity Hashing (LSH) [5]. However, Gaussian projection consumes a lot of RAM and works only for numeric data. So we use Recursive Fastmap with LSH as the "baseline" against which we will compare our own custom models of sway2,3,4. The intuition behind picking PCA (sway2), Agglomerative (sway3) and KMeans (sway4) as our improved models is described below.

There have also been works where other clustering algorithms like PCA, Agglomerative and KMeans were used, not for Fishing, but for other similar use cases. PCA has been used for unsupervised anomaly detection in various domains, such as finance, healthcare, and manufacturing [11]. In another study, Agglomerative clustering was used to group software modules based on their structural and behavioral characteristics [12]. K-means clustering has been used for test case prioritization to group test cases based on their coverage and execution time [13]. This is why we attempt to use PCA (sway2), Agglomerative (sway3) and KMeans (sway4) for MSSC and check if they do better than the Fastmap (sway1) baseline. And with very little work, a clusterer can become an optimizer. So, at each recursive clustering iteration, we use the Zitzler continuous domination predicate, which aims to find a set of non-dominated solutions that provide a trade-off between different objectives [5].

Further, Explaining the MSSC results with a Bin based Contrast Set Learning (CSL) (xpln1) improves transparency and interoperability of the model because it provides insights into why certain instances are grouped together and others are not, which can be useful in understanding the clustering results [15]. However, CSL can be computationally expensive for large datasets and may not generalize well to new instances [16]. Decision Trees resolve these issues and are faster, more flexible and more interpretable than CSL[1]. So, we have compared Bins based Contrast Set Learning (CSL) xpln1 of baseline, with the Decision Tree we applied to sway2,3,4 to get xpln2,3,4.

III. Methods

A. Algorithms

1) Data Pre-processing: The 11 datasets are raw and unclean and the baseline sway1 as well as the improved sway 2,3,4 fail for many of the datasets. So we clean the data using data pre-processing. Firstly, many columns have missing values. We impute these by means. Secondly, many of the proposed algorithms can't work on raw string data. So we use label encoding to convert string columns to numbers. Along with these, a few minor modifications to the algorithm codes make sway1,2,3,4 pass for all 11 datasets.

2) Principal Component Analysis (sway2): Principal Component Analysis (PCA) is a dimensionality reduction technique that can be used to guide MSSC. PCA finds the linear combinations of the original features that capture the maximum variance in the data. The resulting principal components (PCs) are orthogonal and ordered by the amount of variance they explain. First we pre-process the data by imputing missing values with means and then doing label encoding. Then keeping the number of prinicpal components as 1, we apply PCA to the preprocessed rows and then recursively split the results, half in left and half in right clusters. For the distant points we take 10 random choices from each half cluster. Finally, we apply Zitzler predicate to find the set of non-dominated solutions that provide a trade-off between different objectives.

3) Agglomerative Clustering (sway3): Agglomerative clustering is a hierarchical clustering algorithm that can be adapted for Multi-objective semi-supervised clustering (MSSC). The algorithm starts by treating each object as a singleton cluster and iteratively merges the closest pairs of clusters until a stopping criterion is met. First we preprocess the data using imputation and label encoding. Then we choose a distance metric that can capture the similarity or dissimilarity between data points for each objective or label. In the baseline, AHA distance metric was used. But now, as label encoding has already been done, we used Eucledian distance as the distance metric. Next, we choose a linkage criterion that can determine how clusters are formed based on the distance metric. We choose 'ward', that minimizes the variance of the clusters being merged. Keeping the number of clusters as 2, we apply Agglomerative clustering. This gives clusters with labels 0 and 1. Clusters with label 0 are added to the left cluster and those with 1 are added to the right cluster. The distant points again are random samples taken from each half cluster. These are then sent to Zitzler to get best clusters.

4) KMeans Clustering (sway4): KMeans clustering is a Machine Learning algorithm that group 'n' observations into 'K' clusters based on the distance. The objective of K-means algorithm is to minimize the sum of squared distances between each data point and its assigned centroid. After the data has been preprocessed, we apply KMeans keeping number of clusters as 2 for recursive biclustering. Then after we get the left and right clusters, we get the distant points based on whether the distance of the point from the centroid is more than that of it from the original row. Finally, we apply Zitzler to get the best clusters.

5) Boolean Domination (BDOM): We attempted to use Boolean Domination to provide a trade-off between the different objectives. The standard Boolean Domination predicate says one thing dominates another if it is better for at least one and worse for neither. For each row, we compared the product of weights and normalized row values and returned domination true if row1 was better at least one of row2 and worst at neither. However, during Hyperparameter Optimization, it was observed that Zitzler predicate still gave better results for sway1,2,3,4. So we continued with Zitzler predicate only for the rest of the project.

6) Explanation using Decision Trees (xpln2,3,4): Decision Trees can be used as one of the components of an MSSE algorithm to provide interpretable explanations for the model's predictions. We apply Decision Trees to the clusters given by sway2,3,4 to generate xpln2,3,4. We train a simple DecisionTreeClassifier by using the best and rest clusters given by sway as X_train set and adding string labels 'best' and 'rest' to them and using the labels as y_train set. After training, we use our original data rows as the X_test set. For each row in X_test, we generate a prediction. The prediction is again in the form of string labels 'best' or 'rest'. We map these labels to the original data rows based on index and use the row entries as our generated explanation rules.

7) Hyperparameter Optimization (HPO): We performed Hyperparameter Optimization (HPO) using 2 methods: 1. Minimal Sampling method and 2. the established optimization method of HyperOpt. For Minimal Sampling based HPO, for 50 hyperparameter samples, we ran sway and xpln and used Zitzler to compare which hyperparameters gave best results. For Hyperopt, we used a weighted sum objective function and fmin measure to find best params. Post HPO, not just sway and xpln 2,3,4 but also sway and xpln1 gave better results.

B. Data

Name	#X	#Y	#Rows
auto2.csv	21	4	93
auto93.csv	4	3	398
china.csv	16	1	499
coc1000.csv	17	5	1000
coc10000.csv	22	3	10000
healthCloseIsses12mths0001-hard.csv	5	3	10000
healthCloseIsses12mths0011-easy.csv	5	3	10000
nasa93dem.csv	22	4	93
pom.csv	10	3	10000
SSM.csv	13	2	239360
SSN.csv	17	2	53662

TABLE I
DATASET ATTRIBUTES

1) auto2.csv: This is an automobile dataset with car data. The intention is to use the X attributes of a car like maker, type, RPM, etc to predict speed related Y attributes like MPG, weight, class, etc.

2) auto93.csv: This is an automobile dataset with car data. The intention is to use the pure numerics like Model, origin, Volume, etc to predict speed related Y attributes like MPG, weight, Acceleration, etc.

3) china.csv: The China dataset consists of 499 records with 19 attributes. The attributes are PID, AFP, Input, Output, Enquiry, File, Interface, Added, Changed, Deleted, PDR-AFP, PDR-UFP, NPDR-AFP, NPDU-UFP, Resource, Dev.Type, Duration,N-Effort and Effort.

4) coc1000.csv: This is a dataset for software project estimation with 1000 rows and 25 columns. The actual effort in the dataset is measured by person-month which represents the number of. months that one person needs to develop a given project.

5) coc10000.csv: This is a dataset for software project estimation with 10000 rows and 25 columns. The actual effort in the dataset is measured by person-month which represents the number of. months that one person needs to develop a given project.

6) healthCloseIsses12mths0001-hard.csv: This dataset contains different combinations of hyperparameters of sklearn.ensemble.RandomForestClassifier. The goal is to use N_estimators, criterion Min_sample_leaves, Min_impurity_decrease, Max_depth to maximize Accuracy+ and PRED40+ and minimize Mean Error-.

7) healthCloseIsses12mths0011-easy.csv: This dataset contains different combinations of hyperparameters of sklearn.ensemble.RandomForestClassifier. The goal is to use N_estimators, criterion Min_sample_leaves, Min_impurity_decrease, Max_depth to maximize Accuracy+ and PRED40+ and minimize Mean Error-.

8) nasa93dem.csv: This is a COCOMO NASA dataset for software project estimation with 93 rows and 26 columns. The actual effort in the dataset is measured by person-month which represents the number of. months that one person needs to develop a given project.

9) pom.csv: The dataset is of POM3 model, which is a tool for exploring the management challenge of agile development balancing idle rates, completion rates and overall cost.

10) SSM.csv: This dataset contains 239360 features of configuration space of Trimesh, a library to manipulate triangle meshes for FLASH: A Fast Optimizer for SBSE Tasks.

11) SSN.csv: This dataset contains 53662 features of configuration space of Trimesh, a library to manipulate triangle meshes for FLASH: A Fast Optimizer for SBSE Tasks.

C. Performance Measures

For each of the 11 datasets, we rank the data uzing Zitzler predicate and normalize the ranks between 1-100 and show the ranked data means in the .out files as ”top” along with the results of all, sway1, sway2, sway3, sway4. We fix a budget of 10 evaluations maximum and run each algorithm 20 times on the 11 datasets. The evaluations taken by sway1, sway2, sway3, sway4 for each of the datasets given a maximum of 10 budget is shown in Table II.

D. Summarization Methods

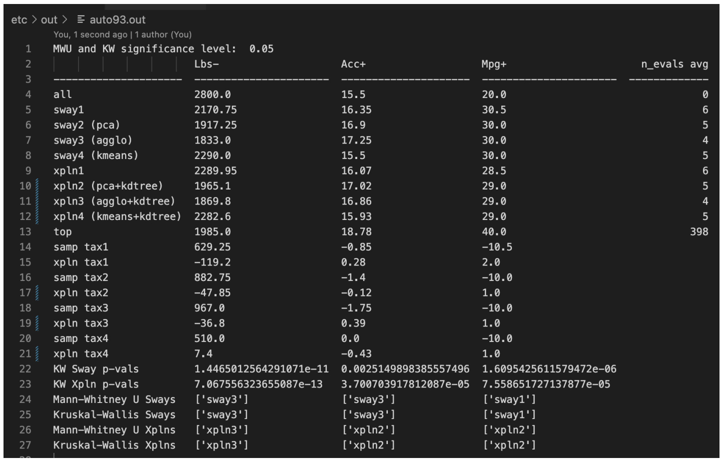
As shown in Table II, sway2,3,4 take less evaluations than sway1. Other than the number of evaluations, we also analyze statistical methods and tests to determine which of the 4 sways and xplns is BEST. For the best rows that we get from sways and xplns, first we use measures of central tendencies (median for numerics, mode for symbolics) to condense multiple rows to one and then take mean of all such medians. So we will have 4 averages for sway1, sway2, sway3, sway4. In statistics, effect size refers to the magnitude of the difference between two groups. To determine if the effect size between each of the pairs of sways is significant, we use two significance tests, namely Kruskal-Wallis and Mann-Whitney U. These tests are used to determine if the difference between two groups is statistically significant, meaning it is unlikely to have occurred by chance. Other than these, we also use cliff’s delta to generate a bottom table for each dataset with the unequal rows. The results of the tests and central tendency measures are discussed below in the Results section.

Name	#Rows	#sway1	#sway2	#sway3	#swa
auto2.csv	93	5	4	4	3
auto93.csv	398	6	5	4	5
china.csv	499	6	5	7	7
coc1000.csv	1000	6	5	4	4
coc10000.csv	10000	8	7	7	5
healthCloseIsses12mths0001-hard.csv	10000	8	7	7	7
healthCloseIsses12mths0011-easy.csv	10000	8	7	7	6
nasa93dem.csv	93	5	4	6	7
pom.csv	10000	8	7	6	8
SSM.csv	239360	10	9	9	10
SSN.csv	53662	9	8	8	8
Total	**32123**	**79**	**68**	**69**	**70**

TABLE II
DATASET SWAY #EVALUATIONS

IV. RESULTS

We ran the 4 sways and xplns for 20 iterations and for 11 datasets. In each iteration, for each column, we get means of medians/modes of the best clusters generated by the 4 sways and xplns. There are a total of 33 Y columns and each column is either ”-”, which we want to minimize or ”+” which we want to maximize. So if the means of central tendency for an algorithm is minimum for a ”-” column or maximum for a ”+” column, then that algorithm is BEST. However, while comparing we also do Kruskal-Wallis and Mann-Whitney U to determine if the differences in means are significant.



```
MWU and KW significance level:  0.05
                       Lbs-                    Acc+                    Mpg+                     n_evals avg
---------------------  ----------------------  ----------------------  ----------------------  -------------
all                    2800.0                  15.5                    20.0                                0
sway1                  2170.75                 16.35                   30.5                                6
sway2 (pca)            1917.25                 16.9                    30.0                                5
sway3 (agglo)          1833.0                  17.25                   30.0                                4
sway4 (kmeans)         2290.0                  15.5                    30.0                                5
xpln1                  2289.95                 16.07                   28.5                                6
xpln2 (pca+kdtree)     1965.1                  17.02                   29.0                                5
xpln3 (agglo+kdtree)   1869.8                  16.86                   29.0                                4
xpln4 (kmeans+kdtree)  2282.6                  15.93                   29.0                                5
top                    1985.0                  18.78                   40.0                              398
samp tax1              629.25                  -0.85                   -10.5
xpln tax1              -119.2                  0.28                    2.0
samp tax2              882.75                  -1.4                    -10.0
xpln tax2              -47.85                  -0.12                   1.0
samp tax3              967.0                   -1.75                   -10.0
xpln tax3              -36.8                   0.39                    1.0
samp tax4              510.0                   0.0                     -10.0
xpln tax4              7.4                     -0.43                   1.0
KW Sway p-vals         1.4465012564291071e-11  0.0025149898385557496   1.6095425611579472e-06
KW Xpln p-vals         7.067556323655087e-13   3.700703917812087e-05   7.558651727137877e-05
Mann-Whitney U Sways   ['sway3']               ['sway3']               ['sway1']
Kruskal-Wallis Sways   ['sway3']               ['sway3']               ['sway1']
Mann-Whitney U Xplns   ['xpln3']               ['xpln2']               ['xpln2']
Kruskal-Wallis Xplns   ['xpln3']               ['xpln2']               ['xpln2']
```

Fig. 1. ”Top Table” of auto93.csv

Name	#Y	#sway1	#sway2	#sway3	#sway4	#xpln1	#xpln2	#xpln3	#xpln4
auto2.csv	4			4				3	1
auto93.csv	3	1		2			2	1	
china.csv	1				1			1	
coc1000.csv	5	1		1	2	3	1		1
coc10000.csv	3	2			1	2			1
healthCloseIsses12mths0001-hard.csv	3		1	1	1			1	2
healthCloseIsses12mths0011-easy.csv	3			3				3	
nasa93dem.csv	4	1		3		1		3	
pom.csv	3	1		1	1	2		1	
SSM.csv	2			1	1			1	1
SSN.csv	2			1	1			1	1
Total	**33**	**6**	**1**	**17**	**8**	**8**	**3**	**15**	**7**

TABLE III
BEST SWAYS AND XPLNS FREQUENCY

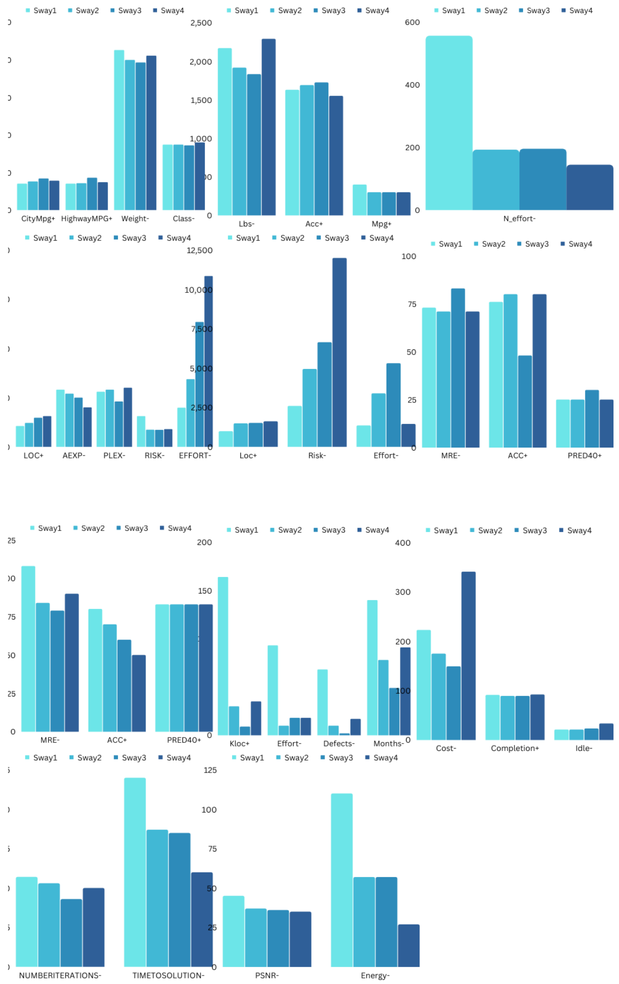


Fig. 2. High Level Overview of Sway1,2,3,4 means for all 11 datasets

Fig. 1. shows the top table we are getting for auto93.csv. We are generating 11 such tables for 11 datasets. Notice the p-values in the table, all are below the 0.05 significance threshold indicating that the mean differences are significant, which is the case in all our 11 datasets too. So we print the best sways and xplns by each of Kruskal-Wallis and Mann-Whitney U in the last 4 rows of the table. As shown in the figure, we can see that for the 3 Y columns of auto93.csv, sway3 was selected for 2 and sway 1 for 1. Also, xpln3 was selected for 1 column and xpln2 for the other two. This was just for the 3 Y columns of auto93.csv. There are a total of 33 Y columns in the 11 datasets. For these 33 Y columns, the frequency of significant BEST sways and xplns are shown in above Table III. Fig 2. also shows the distributions of sway1,2,3,4 means for all 11 datasets.

As shown in Table III, sway3 and xpln3 were chosen BEST more frequently for 17/33 and 15/33 columns followed by sway and xpln 4,1 and lastly 2. Also, from Table II we see that sway3 took less evaluations than sway1. We also observed another trend that as sampling size increased, the results got better. So to summarize, the means of central tendencies of sway and xpln 3 were better than sway and xpln 1 and the differences were significant too. So we can comment that the Agglomerative Clustering + Decision Tree Explanation outperformed Fastmp Clustering + CSL Explanation for most of the datasets.

V. Discussion

A. Threats to Validity

1. Overfitting: There is a risk of overfitting the Agglomerative Clustering and Decision Tree models to the training data, which can lead to poor generalization performance and inaccurate explanations.

2. Lack of transparency: While Decision Trees are generally considered interpretable, they can become complex and difficult to understand as the number of features and classes increases.

3. Limited scope: Decision Trees are limited to generating local explanations for individual predictions, and may not provide a global understanding of the model.

4. Limited expressiveness: Decision Trees may not be able to capture certain types of interactions or patterns in the data, such as nonlinear or high-order interactions.

B. Other Discussions

Considering the above limitations, it would be okay to say that even the baseline sway1 and the improved sway3 may not be 100% accurate all of the time. It only gives us a probablistic estimate of solving the Fishing problem and generating corresponding explanantions.

C. Future Work

It would be interesting to research models like mini-batch K-means that require millions of rows of data. Also, it would be interesting to see if we can get any better alternative for Zitzler predicate. We tried with Boolean Domination in this study, but it turned out to be worse. Exploring aggregation functions to replace Zitzler would be interesting. Lastly, it would be interesting to document and try to optimize the run times of sway1,2,3,4 as this was not covered in this study.

VI. Conclusion

In this study, we have also explained how Multi-objective semi-supervised Explanation can solve both fishing and interoperability problems. And we compared the performances of the 4 Mutli-objective semi-supervised clusterers of Fastmap, PCA, Agglomerative and KMeans along with comparing CSL with Decision Tree for explanation. Agglomerative Clustering + Decision Tree Explanation was found to be the best option and did much better than the baseline of Fastmap Clustering + Bins CSL Explanation. We have also stated the limitations of each of these algorithms and have suggested future scope for the same.

B2. February Study

For each of the datasets, we have calculated sampling tax and explanation tax as per sway and xpln1,2,3,4. The sampling tax is all - swayX and explanation tax is swayX - xplnX. The exact tax figures for auto93.csv are shown in Fig 1 for a simple view. For a holistic view, we compare taxes of xpln2,3,4 with xpln1 and write a P if the sampling and explanantion taxes of xpln2,3,4 are less than xpln1, as shown in Table IV and V. Explanation variance is whether explanations generated from a few random probes of a complex multi-dimensional can be widely variable. To demonstrate this, we plot the frequency of the most frequent rule given by each learner in Table VI. As seen in the table, xpln3 has most probable combination of more frequent rules and therefore has the least variance. So for the Agglomerative Clustering + Decision Tree Explanation xpln3, it is safe to conclude that the same task can be solved with less budget in February using what was learned in January.

Name	xpln2	xpln3	xpln4
auto2.csv		P	P
auto93.csv	P	P	
china.csv		P	
coc1000.csv	P		P
coc10000.csv			P
healthCloseIsses12mthsOOO1-hard.csv			P
healthCloseIsses12mthsOO11-easy.csv		P	
nasa93dem.csv		P	
pom.csv		P	
SSM.csv		P	P
SSN.csv		P	P

TABLE IV
Sampling Tax

Name	xpln2	xpln3	xpln4
auto2.csv	P	P	P
auto93.csv	P	P	P
china.csv	P	P	
coc1000.csv	P	P	P
coc10000.csv			
healthCloseIsses12mthsOOO1-hard.csv		P	P
healthCloseIsses12mthsOO11-easy.csv		P	
nasa93dem.csv	P	P	P
pom.csv		P	P
SSM.csv	P		
SSN.csv		P	P

TABLE V
Explanation Tax

Name	xpln1	xpln2	xpln3	xpln4
auto2.csv	8/20	4/20	**11/20**	8/20
auto93.csv	12/20	8/20	**15/20**	9/20
china.csv	6/20	1/20	6/20	**7/20**
coc1000.csv	7/20	5/20	**10/20**	9/20
coc10000.csv	5/20	**6/20**	5/20	3/20
healthCloseIsses12mthsOOO1-hard.csv	5/20	3/20	**10/20**	7/20
healthCloseIsses12mthsOO11-easy.csv	5/20	8/20	5/20	**10/20**
nasa93dem.csv	**10/20**	4/20	9/20	6/20
pom.csv	**4/20**	3/20	3/20	4/20
SSM.csv	3/20	2/20	**4/20**	4/20
SSN.csv	**5/20**	2/20	4/20	3/20

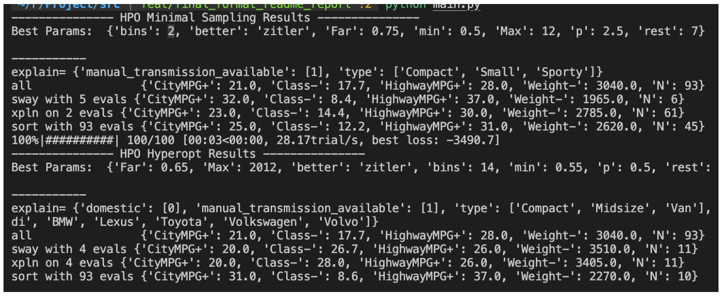
TABLE VI
Explanation Variance by Most frequent rule count

B3. Ablation Study

For Ablation Study, we use the same crux of our project that is around clustering. As a part of ablation study, we compare the 4 algorithms of Fastmap (sway1), PCA (sway2), Agglomerative (sway3) and KMeans (sway4), running one and disabling the others for each of the 20 iterations of the 11 datasets. Refer to Fig. 2 and Table III. We observe that sway3 gave the best performance and also, although sway2 was PCA and we expected it to be better, it gave worse performance than the baseline sway1. We also do one more thing that during hyperparameter tuning, first we restricted the bin size param to 2 to 10, but later made it 2 to 14. We observed during hyperparameter tuning that as bin size increases above 12, we start getting worse figures for all 4 algorithms.

B4. HPO Study

In HPO study, we compare our Hyperparamter Optimization using minimal sampling with the established hyperparameter optimizer of Hyperopt. For Minimal Sampling based HPO, we first generate all possible combinations of the param grid using itertools and randomly sample 50 of them. Then for the 50 hyperparameter samples, we ran sway and xpln and used Zitzler to compare which hyperparameters gave best results. For Hyperopt, we used a weighted sum objective function and fmin measure to find the most optimal hyperparams and ran Hyperopt for 100 evals.



```
~/T/Project/src | feat/final_format_readme_report !2  python main.py
--------------- HPO Minimal Sampling Results ---------------
Best Params:  {'bins': 2, 'better': 'zitler', 'Far': 0.75, 'min': 0.5, 'Max': 12, 'p': 2.5, 'rest': 7}

-----------
explain= {'manual_transmission_available': [1], 'type': ['Compact', 'Small', 'Sporty']}
all                {'CityMPG+': 21.0, 'Class-': 17.7, 'HighwayMPG+': 28.0, 'Weight-': 3040.0, 'N': 93}
sway with 5 evals {'CityMPG+': 32.0, 'Class-': 8.4, 'HighwayMPG+': 37.0, 'Weight-': 1965.0, 'N': 6}
xpln on 2 evals {'CityMPG+': 23.0, 'Class-': 14.4, 'HighwayMPG+': 30.0, 'Weight-': 2785.0, 'N': 61}
sort with 93 evals {'CityMPG+': 25.0, 'Class-': 12.2, 'HighwayMPG+': 31.0, 'Weight-': 2620.0, 'N': 45}
100%|##########| 100/100 [00:03<00:00, 28.17trial/s, best loss: -3490.7]
--------------- HPO Hyperopt Results ---------------
Best Params:  {'Far': 0.65, 'Max': 2012, 'better': 'zitler', 'bins': 14, 'min': 0.55, 'p': 0.5, 'rest':

-----------
explain= {'domestic': [0], 'manual_transmission_available': [1], 'type': ['Compact', 'Midsize', 'Van'],
di', 'BMW', 'Lexus', 'Toyota', 'Volkswagen', 'Volvo']}
all                {'CityMPG+': 21.0, 'Class-': 17.7, 'HighwayMPG+': 28.0, 'Weight-': 3040.0, 'N': 93}
sway with 4 evals {'CityMPG+': 20.0, 'Class-': 26.7, 'HighwayMPG+': 26.0, 'Weight-': 3510.0, 'N': 11}
xpln on 4 evals {'CityMPG+': 20.0, 'Class-': 28.0, 'HighwayMPG+': 26.0, 'Weight-': 3405.0, 'N': 11}
sort with 93 evals {'CityMPG+': 31.0, 'Class-': 8.6, 'HighwayMPG+': 37.0, 'Weight-': 2270.0, 'N': 10}
```

Fig. 3. Minimal Sampling HPO and Hyperopt outputs of auto2.csv

Fig. 2. above shows the output of Minimal Sampling HPO vs Hyperopt ran for a single dataset. We observed that Minimal Sampling HPO generates nearly as good results as Hyperopt with just 50 evaluations. So we are able to get not the best but still good results with relatively fewer peeks with Minimal Sampling.

References

[1] Victor Feitosa Souza, Ferdinando Cicalese, Eduardo Sany Laber, & Marco Molinaro (2022). Decision Trees with Short Explainable Rules. In Advances in Neural Information Processing Systems.

[2] https://github.com/timm/tested/blob/main/docs/onFishing.md

[3] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter and L. Kagal, ”Explaining Explanations: An Overview of Interpretability of Machine Learning,” 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 2018, pp. 80-89, doi: 10.1109/DSAA.2018.00018.

[4] Ghasemi, Zahra & Khorshidi, Hadi & Aickelin, Uwe. (2022). Multi-objective Semi-supervised Clustering for Finding Predictive Clusters. Expert Systems with Applications. 195. 116551. 10.1016/j.eswa.2022.116551.

[5] https://github.com/timm/tested/blob/main/docs/onCluster.md

[6] https://www.molgen.mpg.de/3659531/MITPress–SemiSupervised-Learning.pdf

[7] https://en.wikipedia.org/wiki/Random_projection

[8] Triantaphyllou, Evangelos & Mann, Stuart. (1995). Using the analytic hierarchy process for decision making in engineering applications: Some challenges. The International Journal of Industrial Engineering: Theory, Applications and Practice. 2. 35-44.

[9] .M. Muja and D. G. Lowe, ”Scalable Nearest Neighbor Algorithms for High Dimensional Data,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 11, pp. 2227-2240, 1 Nov. 2014, doi: 10.1109/TPAMI.2014.2321376.

[10] ttp://library.mpib-berlin.mpg.de/ft/gg/gg_why_2008.pdf

[11] https://www.atmosera.com/blog/pca-based-anomaly-detection/

[12] Sarhan, Qusay & Ahmed, Bestoun & Bures, Miroslav & Zamli, Kamal. (2020). Software Module Clustering: An In-Depth Literature Analysis. IEEE Transactions on Software Engineering. 10.1109/TSE.2020.3042553.

[13] Xia, C., Zhang, Y., & Hui, Z. (2021). Test Suite Reduction via Evolutionary Clustering. IEEE Access, 9, 28111-28121.

[14] https://airfocus.com/blog/weighted-decision-matrix-prioritization/

[15] https://github.com/timm/tested/blob/main/docs/onExplain.md

[16] Gardner, M., Artzi, Y., Basmova, V., Berant, J., Bogin, Evaluating models’ local decision boundaries via contrast sets.