Zoish: A Novel Feature Selection Approach Leveraging Shapley Additive Values for Machine Learning Applications in Healthcare

Hossein Javedani Sadaei, Salvatore Loguercio, Mahdi Shafiei Neyestanak, and Ali Torkamani [†]

Scripps Research Translational Institute, and
Department of Integrative Structural and Computational Biology
Scripps Research, La Jolla, CA 92037, USA
E-mail: hjavedani@scripps.edu
E-mail: loguerci@scripps.edu
E-mail: mshafiei@scripps.edu
[†]*E-mail: atorkama@scripps.edu*
www.scripps.edu

Daria Prilutsky

Takeda Development Center Americas, Inc., Cambridge, MA, 02139, USA
E-mail: daria.prilutsky@takeda.com
www.takeda.com

In the intricate landscape of healthcare analytics, effective feature selection is a prerequisite for generating robust predictive models, especially given the common challenges of sample sizes and potential biases. Zoish uniquely addresses these issues by employing Shapley additive values—an idea rooted in cooperative game theory—to enable both transparent and automated feature selection. Unlike existing tools, Zoish is versatile, designed to seamlessly integrate with an array of machine learning libraries including scikit-learn, XGBoost, CatBoost, and imbalanced-learn.

The distinct advantage of Zoish lies in its dual algorithmic approach for calculating Shapley values, allowing it to efficiently manage both large and small datasets. This adaptability renders it exceptionally suitable for a wide spectrum of healthcare-related tasks. The tool also places a strong emphasis on interpretability, providing comprehensive visualizations for analyzed features. Its customizable settings offer users fine-grained control over feature selection, thus optimizing for specific predictive objectives.

This manuscript elucidates the mathematical framework underpinning Zoish and how it uniquely combines local and global feature selection into a single, streamlined process. To validate Zoish's efficiency and adaptability, we present case studies in breast cancer prediction and Montreal Cognitive Assessment (MoCA) prediction in Parkinson's disease, along with evaluations on 300 synthetic datasets. These applications underscore Zoish's unparalleled performance in diverse healthcare contexts and against its counterparts.

Keywords: Feature Selectors, Zoish, SHapley Additive exPlanations

1. Introduction

Healthcare datasets, despite being typically sparse and heterogeneous, are a treasure trove of rich information. However, their high-dimensionality, often paired with smaller sizes, presents obstacles to building predictive models, with overfitting and extensive training time being common concerns.[1–3] Feature selection becomes an essential strategy in this context, aimed at pruning redundant or less important features. This helps to minimize information loss, enhance model interpretability, and curtail computational demands.

Although traditional feature selection methods, grounded in statistical concepts like correlation analysis or chi-square tests, are widely used, they tend to fall short in offering detailed insights into feature importance. This shortcoming, along with the manual effort required, can lead to a time-intensive cycle of feature selection and performance evaluation, thus requiring expert intervention.[4,5]

Our proposed feature selection tool, Zoish, aims to overcome these limitations by utilizing the mathematical framework of additive Shapley values. Originating from cooperative game theory, Shapley values offer detailed understanding of feature importance,[6] thereby enhancing both local (instance-level) and global interpretability. Moreover, the integration of Zoish with our scalable hyperparameter optimization package, Lohrasb,[7] facilitates building models with optimal feature sets, all the while maintaining an industry-ready, user-friendly design.

The structure of the paper is as follows. The first section sheds light on the core concepts of additive Shapley values and delves into the mathematical principles vital to Zoish. Subsequent to this, a section introducing a user guide for Zoish is presented. Lastly, we demonstrate the adaptability of Zoish through a variety of use-cases, experiments on large synthetic datasets, and a closing discussion.

2. Theoretical Foundations of Zoish

Our exploration into Zoish begins with the foundation of its theoretical structure, built upon Shapley additive values. First proposed in the field of cooperative game theory, Shapley additive values have proven to be a potent tool for understanding the contribution of each feature to a prediction made by a machine learning model. Simply put, the Shapley value of a feature represents the average marginal contribution of that feature, factored across all possible combinations of features.

2.1. *Shapley Additive Values and Feature Selection: A Game Theoretical Approach*

In the realm of cooperative game theory, the Shapley value denotes each player's payoff based on their marginal contribution across all possible coalitions. In machine learning, 'players' correspond to the features and 'game' to the prediction task.[6]

An additive cooperative game assumes that the value of any coalition equals the sum of its members' independent values. This idea leads us to Shapley additive values, where the Shapley value of a feature equals its average marginal contribution across all feature subsets.

Mathematically, the Shapley value for a feature i in an additive game is:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)) = v(\{i\})$$

In this formula, $|N|$ denotes the total number of features, S a subset of features excluding feature i, and $|S|$ the number of features in subset S. The terms $|S|!(|N| - |S| - 1)!$ and $|N|!$ calculate the number of possible permutations of features. The expression $v(S \cup \{i\}) - v(S)$ computes the marginal contribution of feature i when added to subset S.

In feature selection, the Shapley value presents a way to distribute the model's prediction among the features, based on their marginal contribution.[8,9] Shapley Additive exPlanations (SHAP) values, maintaining additivity, provide a unified measure of feature importance, attributing the difference between the model's actual and expected output to each influencing feature.[10] High Shapley or SHAP values indicate significant feature importance, while values near zero suggest negligible predictive power.[11] This correlation aids in reducing data dimensionality and enhances model interpretability, marking a significant stride in feature selection.

2.2. *Properties of Shapley Additive Values*

The Shapley additive values satisfy a number of properties that make them particularly useful for interpreting machine learning models:

- **Efficiency**: The sum of the Shapley values of all features is equal to the difference between the prediction for an instance and the average prediction over all instances.
- **Symmetry**: If two features contribute equally to all possible combinations of features, they have the same Shapley value.
- **Additivity**: Given two games (or in our context, two models), the Shapley value of the combined game is the sum of the Shapley values of the individual games.
- **Nullity (Dummy)**: If a feature does not improve the prediction for any combination of features, its Shapley value is zero.

2.2.1. *Proof of Nullity (Dummy)*

The Nullity (Dummy) property states that if a feature does not change the prediction model, i.e., its contribution is always zero, then its Shapley value is also zero. Let f be the prediction model and d be such a dummy feature.

According to the definition of Shapley values, the Shapley value of a feature is the average of its marginal contributions across all possible subsets of features. Therefore, the Shapley value $s_f(d)$ for the dummy feature d is:

$$s_f(d) = \frac{1}{M} \sum_{S \subseteq N \setminus \{d\}} |S|!(|N| - |S| - 1)!(f(S \cup \{d\}) - f(S))$$

In the above formula, M is the total number of possible subsets of N that can be formed when the dummy feature d is excluded. The term $|S|!(|N| - |S| - 1)!$ is the number of permutations of N in which the dummy feature d and the features in subset S appear together, and

$f(S \cup \{d\})$ and $f(S)$ are the values of the game f when the dummy feature d is added to subset S and when it is not, respectively.

Since d is a dummy feature, adding it to any subset S does not change the value of the game f. Thus, we have $f(S \cup \{d\}) = f(S)$ for all $S \subseteq N \setminus \{d\}$, which simplifies the above formula to:

$$s_f(d) = \frac{1}{M} \sum_{S \subseteq N \setminus \{d\}} |S|!(|N| - |S| - 1)!(0)$$

$$s_f(d) = 0$$

This confirms that the Shapley value of a feature that does not contribute to the prediction model is indeed zero, thus proving the Nullity (Dummy) property.[9]

2.3. *Leveraging Shapley Additive Values for Efficient Feature Selection*

Shapley additive values have become increasingly prominent in feature selection for machine learning due to their robustness, efficiency, and power. Verdinelli et al. examined the explainability of machine learning models, focusing on methods such as LOCO and Shapley Values for assessing feature importance. Although their research indicated that Shapley Values do not eliminate feature correlation, they proposed new, statistically sound axioms for measuring feature importance.[12] In a separate study, Karczmarz et al. compared Shapley and Banzhaf values in the context of explaining tree ensemble models. They found Banzhaf values to be more intuitive, efficient, and numerically robust, and introduced faster algorithms for both methods to improve computational efficiency.[13] The SHAP (SHapley Additive exPlanation) library serves as a prime example of effectively leveraging the additivity and efficiency inherent in Shapley values.[14] A fundamental advantage of Shapley values lies in their additivity, which enables fast and efficient computation, especially in the context of tree-based models. The SHAP and FastTreeShap libraries employ Tree SHAP, a highly efficient and accurate algorithm designed for tree ensembles.[14,15] Given the innate additivity of ensemble tree models, which amalgamate multiple decision trees, this characteristic ensures swift and precise computation of Shapley values.[6] The efficacy of Shapley values is further underscored by their intrinsic efficiency. This is manifested in the fact that the sum of the Shapley values for all features equals the difference between the prediction for a specific instance and the average prediction across all instances. This aspect permits a meaningful distribution of the ”credit” for a prediction across features, hence illuminating their relative importance. The Zoish package,[16] designed to optimize feature selection, taps into the beneficial properties of Shapley values. It employs the Nullity (or Dummy) property to eliminate features with Shapley values close to or exactly zero, indicating their minimal predictive relevance. To assist users in setting the cut-off level, two methods are offered: one involves setting an internal parameter called *threshold*, while the other entails defining the number of desired features to retain in the model. By removing these non-influential features, the package enables dimensionality reduction of the model without sacrificing prediction quality. The symmetry property of Shapley values is also exploited by Zoish. This property mandates that if two features contribute equally to all possible subsets of other features, they must have identical Shapley values. By identifying and discarding these redundant features, Zoish facilitates the construction of models that are simpler and more interpretable, with no compromise on predictive power. By utilizing the SHAP and FastTreeShap libraries to incorporate Shapley additive values, Zoish implicitly benefits from its advantages, including the mathematical robustness and beneficial properties of Shapley values. Therefore, these libraries and the Zoish package present themselves as potent instruments for feature selection, spanning a wide range of machine learning tasks.

3. Feature Selection Approaches

Zoish is a versatile package designed to enhance the evaluation of feature importance and improve the overall performance of machine learning models.[16] While Zoish can function effectively as a standalone tool for feature selection, it is engineered to be highly extensible and can seamlessly integrate with hyperparameter optimization packages to further refine its capabilities. One such potent integration is with the Lohrasb package,[7] which provides advanced tuning methods to optimize the feature selection process. However, it's worth noting that users are not confined to using Lohrasb; Zoish's flexible architecture allows for easy integration with other hyperparameter optimization tools as well.

3.1. *Optimization and Flexibility in Zoish*

Zoish's integration with Lohrasb serves a dual purpose: it not only optimizes the tree-based estimator used for feature selection but also offers a choice of hyperparameter tuning methods, including Optuna, GridSearchCV,[17] RandomizedSearchCV,[18] OptunaSearchCV, tune-sklearn,[19] and Ray's Tune.[20] This optimization is crucial for enhancing Zoish's feature selection capabilities, as represented in Fig 1. However, the use of Lohrasb is optional, giving users the freedom to employ other tree-based estimators or hyperparameter tuning engines. Even without hyperparameter optimization, Zoish maintains its core functionality, allowing for a balance between efficiency and interpretability. The importance of hyperparameter optimization for feature selection is further elaborated in Section 6. Therefore, while Lohrasb's role is significant for optimal performance, users have the flexibility to choose an approach that best suits their specific needs.

3.2. *Workflow explanation*

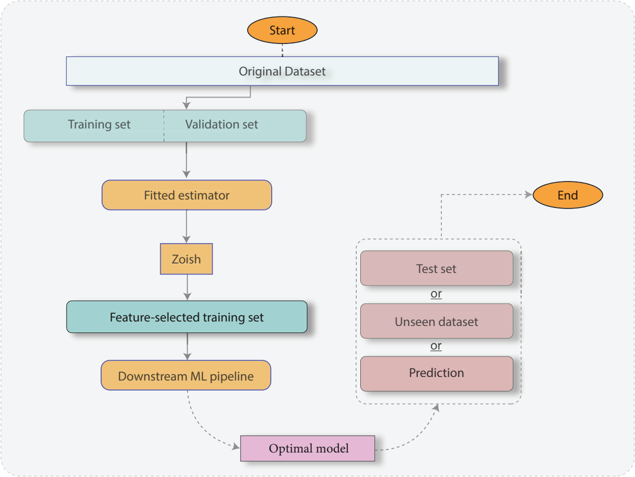
Within a machine learning pipeline, Zoish functions as a feature selection component. The pipeline commences by cleaning and splitting the original dataset into training and validation subsets. A tree-based estimator, which is compatible with Zoish, is trained on the training subset. If hyperparameter tuning is applied, tools such as Lohrasb optimize the estimator against a specific metric, as shown in Fig 1.

Once the estimator is optimized, it becomes an input to Zoish along with a set of parameters, such as cross-validation settings, Shapley value calculation algorithms, and feature importance thresholds. Zoish computes Shapley values via either the SHAP library for smaller datasets, due to its exhaustive computational approach, or FastTreeShap for larger datasets, owing to its computational efficiency.

Based on the calculated Shapley values, Zoish automatically selects the highest-ranking features. The training set is then narrowed down to these selected features. Subsequently, these refined training and validation sets are channeled to the next steps in the pipeline, which usually involve fitting another predictive model.

To ensure robustness in feature selection, Zoish employs multiple rounds of cross-validation on the same training set, regulated by a parameter named *n_iter*.

Fig. 1. Zoish workflow



Documentation and code examples elucidating these operational details can be found in the Zoish repository.

4. Source Code, installation and usage example

The public repository of Zoish is available on GitHub alongside examples for end users is `https://github.com/TorkamaniLab/zoish`. Zoish package is available on PyPI and can be installed with pip:

```
pip install zoish
```

A straightforward example demonstrates how Zoish can be effectively combined with hyperparameter optimizers. Both this example and the comprehensive documentation in the repository highlight the package's flexibility and adaptability across various scenarios.

```
import xgboost as xgb
from sklearn.model_selection import KFold, GridSearchCV
from zoish.feature_selectors.shap_selectors import ShapFeatureSelector
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
```

```
X_train, X_test, y_train, y_test = ...  # Your dataset here

grid = GridSearchCV(xgb.XGBClassifier(), {'n_estimators': [100, 150], 'max_depth':\
[6, 10], 'gamma': [0.5, 1.0]}, cv=5, n_jobs=-1, scoring='accuracy').\
fit(X_train, y_train)

shap_selector = ShapFeatureSelector(grid.best_estimator_, \
num_features=15, cv=KFold(10), n_iter=5, direction="maximum", \
scoring="accuracy", algorithm='auto', use_faster_algorithm=True)

pipeline = Pipeline(steps=[("s", shap_selector), \
("m", LogisticRegression())]).fit(X_train, y_train)
```

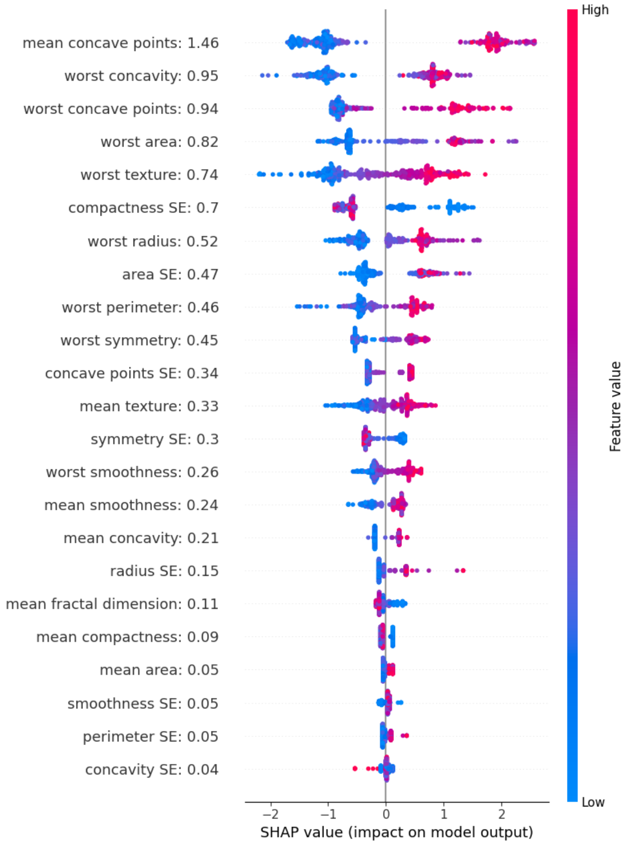
5. Use cases and applications

5.1. *Use case 1: Application to UCI breast cancer dataset - comparison with related XAI work*

To demonstrate the value of the Zoish feature selector in a real use case from the biomedical domain, we applied it to the openly available breast cancer dataset from the UCI Archive,[21] and compared the results with a recent study evaluating different feature importance measures for the same dataset.[22]

The UCI dataset includes benign and malignant samples from 569 patients, 212 with cancer and 157 with fibrocystic breast masses. Each sample includes thirty features - ten real valued features for each cell nucleus (*radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension*) each reported as Mean, Standard Error (SE) and Worst.[23] As the classes in this dataset are almost linearly separable, classification per se is not a difficult task; however, the most important features generally differ depending on the technique used.[24] To further investigate this aspect, Saarela et al.[22] compared different feature importance measures using both linear (logistic regression) and non-linear (random forest) models and local interpretable model-agnostic explanations for the same dataset. In Fig. 2 we show the top 20 important features for the UCI Breast Cancer dataset computed with Zoish by training a XGboost classifier over ten folds of cross validation. The AUC for the trained classifier was 0.96, similar to the mean AUC reported in[22] (0.99+-002). Overall, the most important features in Zoish/XGboost agree well with the set of nine statistically significant features for both RF and LR reported in[22] – where, for each method, significance was computed through a procedure based on permutation tests – i.e. by shuffling class labels in the training data over hundreds of runs. Seven out of nine features deemed statistically significant in[22] were found in Fig. 2 (*Mean concave points, Worst concave points, Worst Area, Worst Radius, Worst Perimeter, Mean Concavity, Mean Area*), with five of the most significant features near the top of the list (Table 1). Only one feature was labeled as not significant by both RF and LR (*Worst Compactness*) and such feature has consistently a zero Shap value in Zoish/XGboost (not shown).

Fig. 2. Shap summary plot of the top 20 important features for the UCI Breast Cancer dataset - computed with Zoish by training a XGboost classifier.



It is worth noting that certain features which rank very high in Zoish/XGboost (*Worst concavity, Worst texture, Compactness SE*) appeared not to be significant in RF classification, hence not reported in the set of nine common, statistically significant features for breast cancer classification in the UCI dataset. Why did Zoish / XGboost select them up then? A very interesting hint comes from the analysis of local importance measures for a specific set of observations in the UCI dataset. Again in,[22] LIME (local interpretable model-agnostic explanations,[25]) was used to estimate local importances for the four most interesting observations, (i.e. correctly classified as benign with highest probability, correctly classified as malignant with highest probability, misclassified as benign with highest probability, and misclassified as malignant with highest probability). Strikingly, the features ranking high in Zoish/XGboost but absent in RF were also important features in LIME, especially for the observations misclassified as benign (false negatives), which are critical for medical purposes (Table 1). The final recommendation in[22] was to combine several explanation techniques in order to provide more reliable and trustworthy results, but this advice can often be impractical. Conversely, The Zoish/XGboost feature selector appears to select relevant features both at the global and local level, adding more detailed explanations of feature importance (i.e., not just the magnitude but also the direction of change), while being fast and straightforward to use.

Table 1. Comparison of top features in the UCI Breast Cancer dataset

Zoish features	**Significant features in LR and RF**	**LIME - Correctly classified benign (RF)**	**LIME - Misclassified benign (RF)**
concave points 1	concave points 3	area 3	perimeter 3
concavity 3	area 3	perimeter 3	area 3
concave points 3	concave points 1	radius 3	radius 3
area 3	area 1	concave points 3	texture 3
texture 3	perimeter 3	texture 3	concavity 3
compactness 2	radius 3	concave points 1	area 2
radius 3	concavity 1	concavity 3	smoothness 3
area 2	perimeter 1	area 2	area 1
perimeter 3	radius 1	texture 1	concave points 1
symmetry 3	area 2		
concave points 2	concavity 3		
texture 1	texture 3		
symmetry 2	texture 1		
smoothness 3	compactness 3		
smoothness 1	radius 2		
concavity 1	perimeter 2		
radius 2	compactness 1		
fractal dimension 1	smoothness 3		
compactness 1	symmetry 3		
area 1	fractal dimension 3		

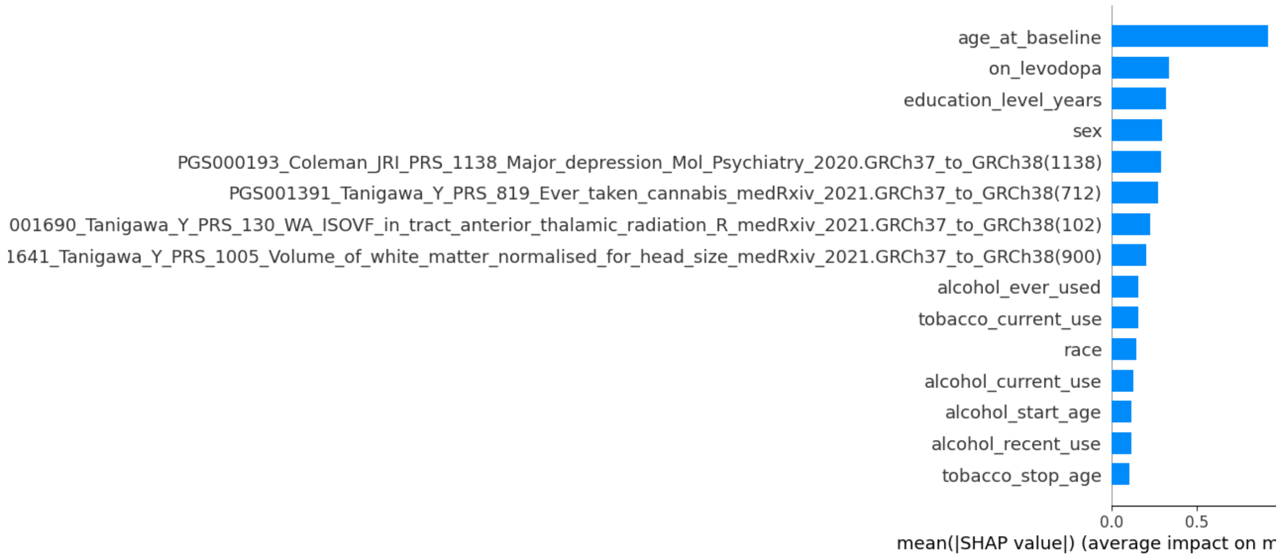
This Table is about comparison of top features in the UCI Breast Cancer dataset computed by Zoish/XGboost vs. Random Forest / Logistic Regression / LIME. For all features, 1=Mean, 2=Standard Error, 3=Worst. Top 20 features in Column 2 are ranked based on permutation p-value for RF. In red are features found in LIME but not considered significant in RF/LR.

5.2. *Use case 2: Predict short-term PD progression status using the Montreal Cognitive Assessment (MoCA)*

Our model, Zoish, was put to another practical test where we aimed to predict short-term PD progression status using the Montreal Cognitive Assessment (MoCA) total scores for patients in baseline. MoCA was developed as a tool to screen patients who present with mild cognitive complaints and usually perform in the normal range on the MMSE (Mini-Mental State Examination).[26] For this prediction task, we utilized the AMP-PD dataset, which is a comprehensive collection of data from various sources, including clinical information, genetic data, imaging data, and other biomarkers from individuals with Parkinson's disease. The dataset consists of eight cohorts, making it a large and harmonized resource. Access to the data was obtained under the AMP-PD Data Use Agreement, and the information was retrieved from the website: `https://amp-pd.org/`. Our prediction model incorporated several essential features from the datasets, such as "family history," genetic information (PRS), "medical history," "smoking and alcohol history," and demographic information of the participants from the eight cohorts. After fitting the model, we evaluated its performance using the coefficient of determination, commonly known as R-squared, and achieved an R-squared value of 23.6 percent on the test dataset. Additionally, we calculated the Mean Squared Error (MSE) of the model to be 2.49 for the test dataset. Furthermore, the Mean Absolute Error (MAE) was found to be 13.04. The MAE represents the average absolute difference between the predicted values and the actual total score values in the test dataset. List of selected features by Zoish can be seen in Fig. 3

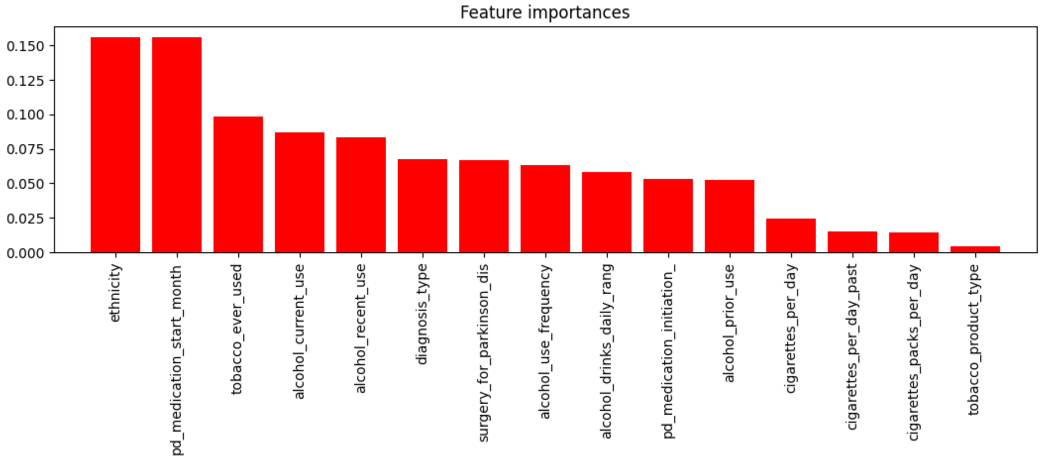
In order to draw a comparison with another prevalent feature selector, specifically, **SelectFromModel** from the sklearn library, we applied it to the same dataset under identical

Fig. 3. List of selected features and their importance by Zoish



conditions. This application yielded a Mean Absolute Error (MAE) of 15.91, a Mean Squared Error (MSE) of 2.69, and an R-squared value of 0.12. The features selected by this approach are depicted in Fig. 4.

Fig. 4. List of selected features and their importance by SelectFromModel



As observed, Zoish not only outperforms its counterparts in terms of prediction accuracy, but it also excels in the selection of meaningful features. Notably, the Polygenic Risk Scores (PRSs) selected by Zoish have demonstrated substantial relevance to the Montreal Cognitive Assessment (MoCA). A prime example among these is *PGS001641*, which is renowned for its strong correlation with the volume of white matter, normalised for head size. This particular PRS underscores the genetic predisposition towards the volume of white matter, a crucial neuroimaging measurement that relates directly to cognitive functions evaluated in the MoCA test. Therefore, the selection of this PRS by Zoish validates its capability in discerning features with profound implications for cognitive assessment.

6. Evaluations and Performance Analysis

To offer a comprehensive evaluation of Zoish, we performed an array of tests ranging from comparative analyses to hyperparameter optimization and scalability assessments.

Comparative Analysis: We initiated our evaluation with a rigorous comparison involving 300 synthetic datasets tailored to mirror the complexities of healthcare data. These datasets span regression, binary classification, and multi-label classification tasks. Zoish was compared against six established feature selection techniques from Scikit-learn under identical conditions.Our findings suggested that Zoish surpassed other selectors in 77% of regression problems, while in multi-label classification and binary classification tasks, Zoish outperformed in 53% and 57% of the cases, respectively (refer to Table 2).

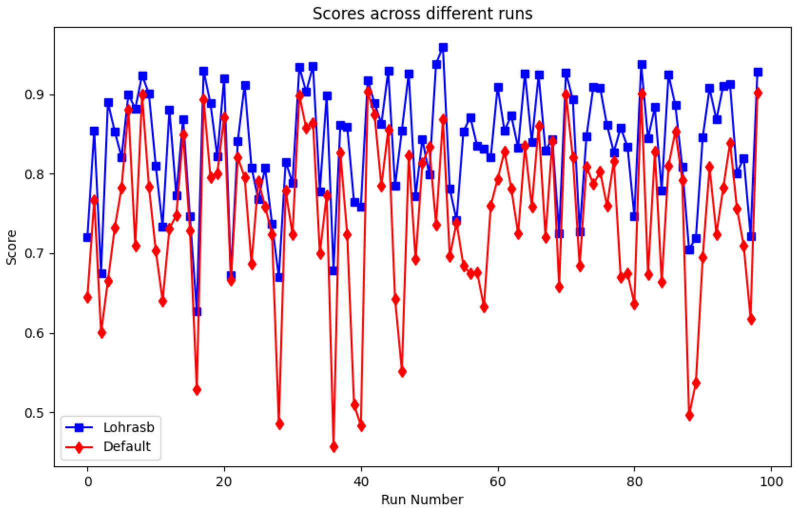
Table 2. Performance comparison of Zoish with other feature selectors

Selector	Regression	Binary Classification	Multi-label Classification
Zoish	**77%**	**57%**	**53%**
VarianceThreshold	2%	3%	6%
SelectKBest	2%	3%	6%
SelectPercentile	2%	2%	2%
RFE	5%	10%	6%
RFECV	7%	10%	11%
SelectFromModel	5%	15%	16%

Hyperparameter Optimization: While Zoish itself is powerful, coupling it with a hyperparameter optimization tool like Lohrasb significantly improves performance. We performed 100 runs comparing Zoish's efficacy with and without Lohrasb, and found marked improvements when paired with Lohrasb (see Fig. 5).

Scalability: Our most recent update introduces a faster algorithm for Shapley value computation, making Zoish efficient on large datasets. In our trials, Zoish selected 500 features from a dataset with 10,000 samples in under 2 minutes on a machine with a 2.3 GHz Quad-Core Intel Core i7 processor and 32 GB RAM.

Fig. 5. The importance of Hyperparameter Optimization for Better Feature Selection



All the code for our tests is available in the public repository, allowing for independent verification and further exploration of Zoish's capabilities.

7. Discussion and Limitations

This paper introduces Zoish, a feature selection tool built on cooperative game theory principles.[16] Zoish has gained traction in the community, as evidenced by a significant number of downloads from pip-trends (`https://piptrends.com/package/zoish`). The tool specializes in optimizing predictive models, particularly in the healthcare sector, and leverages Shapley additive values for a comprehensive view of feature importance at both local and global scales.[10] Through its Nullity property, Zoish effectively minimizes model complexity by omitting features with negligible Shapley values, thereby retaining model performance.[3] The tool is further enriched by integration with the Lohrasb package, which aids in achieving optimal estimators and hyperparameter settings.[7]

While Zoish's capabilities are robust, some limitations are noteworthy. Firstly, its computational efficiency may be compromised when dealing with exceptionally large datasets. Secondly, the Shapley values employed assume feature independence and local linearity—assumptions that may not be fully met in complex applications like healthcare. These limitations are partially mitigated by Zoish's tree-based modeling approach, which is robust to feature correlation and can capture non-linear relationships.[6]

The flexibility and interpretability of Zoish make it a promising tool for future applications in other high-dimensional data fields, including finance and e-commerce. Additional functionalities could be incorporated to broaden its applicability further.

Future work will focus on extending Zoish's utility to various high-dimensional domains and incorporating more algorithms and tools for an even more robust feature selection process. We have conducted extensive tests on synthetic datasets mimicking real-world complexities in healthcare, which are detailed in Section 6. These tests demonstrate Zoish's reliability and adaptability, even under challenging conditions.

8. Author contributions statement

Author contributions include; conceptualization (HJS, AT), methodology and software (HJS), validation and data curation (HJS, SL, MSN, AT, and DP), writing (HJS, SL, MSN, and AT), and funding (AT).

9. Acknowledgments

This work is supported by UL1RR025774 and R01HG010881. Also We would like to express our gratitude to Dua, D. and Graff, C.[21] for generously providing us with the Breast Cancer Data used in this study. Without their contribution, this research would not have been possible.

References

1. I. M. Johnstone and D. M. Titterington, Statistical challenges of high-dimensional data (2009).
2. M. Muja and D. G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, *IEEE transactions on pattern analysis and machine intelligence* **36**, 2227 (2014).
3. H. J. Sadaei, A. Cordova-Palomera, J. Lee, J. Padmanabhan, S.-F. Chen, N. E. Wineinger, R. Dias, D. Prilutsky, S. Szalma and A. Torkamani, Genetically-informed prediction of short-term parkinson's disease progression, *npj Parkinson's Disease* **8**, p. 143 (2022).
4. J. Tang, S. Alelyani and H. Liu, Feature selection for classification: A review, *Data classification: Algorithms and applications* , p. 37 (2014).
5. M. A. Hall, Correlation-based feature selection for machine learning (1999).
6. D. Fryer, I. Strümke and H. Nguyen, Shapley values for feature selection: The good, the bad, and the axioms, *Ieee Access* **9**, 144352 (2021).
7. H. Javedani Sadaei and A. Torkamani, Lohrasb: A scalable estimator optimization tool compatible with scikit-learn APIs (April 2023).
8. T. P. Michalak, K. V. Aadithya, P. L. Szczepański, N. R. Jennings and R. Narayanam, Efficient computation of the shapley value for game-theoretic network centrality, *Journal of Artificial Intelligence Research* **46**, 607 (2013).
9. L. S. Shapley, A value for n-person games, *Contributions to the Theory of Games* **2**, 307 (1953).
10. S. M. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions, *Advances in neural information processing systems* **30** (2017).
11. C. Molnar, *Interpretable machine learning* (Lulu.com, 2020).
12. I. Verdinelli and L. Wasserman, Feature importance: A closer look at shapley values and loco, *arXiv preprint arXiv:2303.05981* (2023).
13. A. Karczmarz, A. Mukherjee, P. Sankowski and P. Wygocki, Improved feature importance computations for tree models: Shapley vs. banzhaf, *arXiv preprint arXiv:2108.04126* (2021).
14. S. Lundberg, S.-I. Lee *et al.*, SHAP (SHapley Additive exPlanations) `https://github.com/slundberg/shap`, (2023), Accessed: 2023-09-25.
15. J. Yang, Fast treeshap: Accelerating shap value computation for trees, *arXiv preprint arXiv:2109.09847* (2021).
16. H. Javedani Sadaei and A. Torkamani, Zoish: Automated feature selectoion tools (July 2023), If you use this software, please cite it as below.
17. S. Varma and R. Simon, Parameter estimation using grid search with cross-validation, *XGBoost: A Scalable Tree Boosting System* **12**, p. 48 (2012).
18. J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, *Journal of Machine Learning Research* **13**, 281 (2012).
19. R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez and I. Stoica, Tune: A research platform for distributed model selection and training in pytorch, *arXiv preprint arXiv:1807.05118* (2018).
20. R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez and I. Stoica, Tune: A research platform for distributed model selection and training in pytorch, *arXiv preprint arXiv:2002.02613* (2020).
21. D. Dua and C. Graff, UCI machine learning repository (2017).
22. S. J. Mirka Saarela, Comparison of feature importance measures as explanations for classification models, *SN Applied Sciences* **3** (2021).
23. O. M. WH Wolberg, WN Street, Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates, *Cancer Letters* **77**, p. 163–171 (1994).
24. A. S. E Aličković, Breast cancer diagnosis using ga feature selection and rotation forest, *Neural Computation Applications* **28**, p. 753–763 (2017).

25. M. T. Ribeiro, S. Singh and C. Guestrin, ”why should i trust you?”: Explaining the predictions of any classifier (2016).
26. 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**, 695 (2005).