# Using SHAP Post-Model Explainability as a Model-Agnostic Feature Selection Technique

Joshua Gottlieb, Martin Sichali Chunsen and Alexander Yoo / Dr. Krishna Bathula

Seidenberg School of Computer Science & Information Systems

## ABSTRACT

Feature selection techniques in data science help reduce dimensionality of datasets, remove sparsity, and eliminate noisy patterns, thereby improving generalization of models to future data [1]. Many feature selection techniques fail to optimize for the predictive power of the model, require iterating through the extensive feature space, or apply only to certain model types. SHAP values focus on capturing the predictive power of each feature within the context of the trained model in order to provide explainability. Because SHAP treats the model as a black box, the SHAP technique is model agnostic. In this paper, we employ SHAP values to capture and rank the important features learned by a model and to use these rankings for feature selection. We build upon prior work using SHAP as a feature selection technique by creating the MAX and SUM selection algorithms, which are designed to align with intuitive feature selection goals without the requirement for costly iterative processes. We validate the effectiveness of our SUM and MAX SHAP selection algorithms by testing their performance with five machine learning models using ten diverse datasets, demonstrating that these SHAP selection strategies produce equivalent or superior model performance and greater feature reduction compared to other state-of-the-art techniques.

## BACKGROUND AND LITERATURE REVIEW

Traditional feature selection techniques are classified into three groups: filter, wrapper, and embedded methods [2]. Filter methods are fast but rely on statistical measures and are unable to use patterns learned by models. Wrapper methods require iteratively retraining models on different feature subsets, making them computationally expensive. Embedded methods operate during the model training process but are model-specific.

SHAP was originally designed for machine learning explainability to explain black-box model predictions [3]. SHAP assigns values to features that are consistent and locally accurate with model predictions. This means that SHAP is model-agnostic and captures patterns learned by the model, while being a universal technique that can be applied to any model type. These SHAP values can be used as feature rankings for feature selection.

SHAP values provide feature rankings, and there has been a lack of work in deciding how many features to use without resorting to computationally expensive iterative methods.

- Marcilio and Eler naively selected the top 10%/…/100% of features. [4]
- Gebreyesus et. al. combined SHAP with sequential forward selection. This iterative method was expensive and lacked the benefits of a one-pass technique. [5]
- Kraev et. al. filtered features by running significance tests using recursive logistic/linear regression tests on the SHAP features. Their results showed minimal improvement with high computational overhead. [6]
- Wang et. al. naively chose the top 3/5/7/10/15 features without explanation. [7]
- Keany implemented the BorutaShap algorithm, which combines the Boruta feature selection method with SHAP values. This process requires many iterations. [8]
- Verhaeghe et al. introduced Powershap. Like the BorutaShap algorithm, Powershap creates an expanded feature set and requires many iterations to select features. [9]

Our study addresses the gap in current literature by introducing two new SHAP selection algorithms (SUM and MAX) which require no iterative processes and are grounded in intuitive human selection goals. We test the robustness of our techniques across five different machine learning models using ten binary classification datasets of varying size and complexity. We use the model-agnostic Permutation SHAP explainer, to showcase the model-agnostic strengths of SHAP as a feature selection technique.

## DATA PREPARATION

| Dataset Name | Source | Instances | Features |
|---|---|---|---|
| Indian Liver Patient | UCI | 584 | 10 |
| Heart Disease | UCI | 303 | 13 |
| Mushroom | UCI | 8,124 | 22 |
| SPECT Heart | UCI | 267 | 23 |
| Breast Cancer | UCI | 569 | 30 |
| Secondary Mushroom | UCI | 61,068 | 20 |
| Credit Card Fraud | Kaggle | 284,807 | 30 |
| Phishing URL | UCI | 235,795 | 54 |
| Patient Survival | Kaggle | 91,700 | 84 |
| Android Permissions | UCI | 29,333 | 86 |

Each of our ten datasets was sent through basic preprocessing. Duplicate rows and columns with greater than 50% missingness were dropped. Remaining missing values were imputed using the median for numeric columns and with the mode for categorical columns. Categorical features were encoded based on cardinality:

- Cardinality 2: binary flags
- Cardinality 3-10: one-hot encoding
- Cardinality 10+: target encoding

For each dataset, the data was split 80%/20% for training and testing.

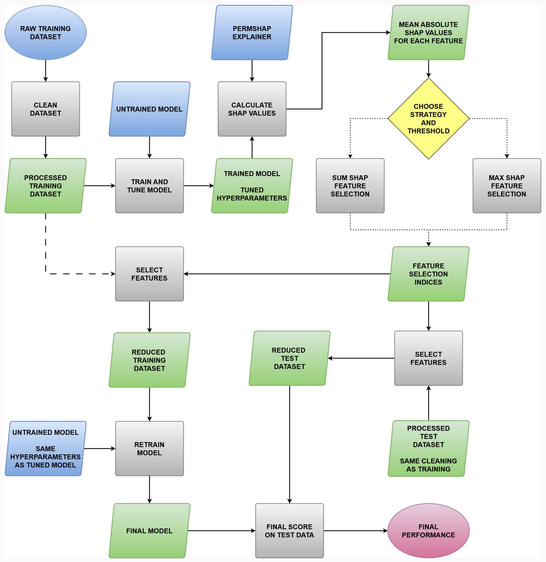
## METHODOLOGY

**Algorithm 1** MAX SHAP Psuedocode

Sort the feature strengths $I_j$ in descending order: $I_j \geq I_{j+1}$
Choose $M = \max(I_j) = I_0$
Define a proportion constant $\rho$ in [0, 1].
Initialize an index $j = 0$.
**while** $I_j \geq \rho \cdot M$ **do**
  Increment $j$
**end while**
**return** $[f_0, \ldots, f_j]$, the selected sorted features.

**Algorithm 2** SUM SHAP Pseudocode

Sort the feature strengths $I_j$ in descending order: $I_j \geq I_{j+1}$
Calculate the total sum $S = \sum I_j$
Define a proportion constant $r$ in [0, 1].
Initialize a running sum $s$ and an index $j = 0$.
**while** $s < r \cdot S$ **do**
  Add $I_j$ to $s$ and increment $j$
**end while**
**return** $[f_0, \ldots, f_j]$, the selected sorted features.




1. Each dataset is processed for use in modeling.
2. Five models are trained on each dataset (Logistic Regression, Decision Trees, Random Forests, XGBoost, and Support Vector Classifiers). Hyperparameter tuning is performed using 5-fold CV, with PR AUC as the optimization metric.
3. The trained models are used with the Permutation explainer to calculate the SHAP values, which are aggregated and sorted into a global ranked feature list.
4. Feature selection is performed using one of our two SHAP selection algorithms: SUM and MAX. The reduced feature training and testing sets are created, and the model is re-trained with the same hyperparameters on the reduced feature training set.
5. Model performance is scored using the reduced feature test data for later evaluation.

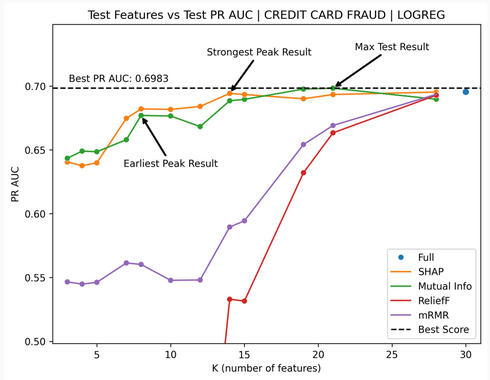
The MAX strategy selects all features that are at least as strong as $\rho \cdot M$, where M is the strongest feature importance. The MAX strategy frames selection in terms of relative feature strength. The user selects an intuitive threshold (ρ) for feature strength with respect to the strongest feature. Values tested for ρ: 0.01, 0.05, 0.1, 0.15, 0.25, 0.5.

The SUM strategy selects features until the sum of their importances is at least as strong as $r \cdot S$, where S is the sum of all of the feature importances. The SUM strategy frames selection in terms of total explanation. The user selects an intuitive threshold ($r$) for total explanation. Values tested for $r$: 0.5, 0.6, 0.7, 0.8, 0.9, 0.95.

The SHAP strategies were compared with three filter-based methods: Mutual Information Gain (MI), ReliefF, and Minimum-Redundancy-Maximum-Relevance (mRMR).

Across all datasets, models, selection thresholds, and selection strategies, the workflow produced more than 2,410 experiments, enabling a comprehensive comparison between SHAP-based and traditional feature selection methods.

## RESULTS AND ANALYSIS




Choosing the best number of features ($k$) is challenging, so we evaluate SHAP selection using four frameworks: minimum overfitting, maximum test performance, the strongest peak, and the earliest peak in test PR AUC. Minimum overfitting selects the $k$ that minimizes the difference between cross-validation and test PR AUC, though it often yields very small feature sets. Maximum test performance selects the $k$ that maximizes test PR AUC and usually keeps the most features. Peak-based methods identify local maxima (after removing peaks below the 0.33 quantile) where the strongest peak finds the highest local maximum and the earliest peak selects the first rising local maximum. A visual interpretation of the latter three frameworks is shown above.

**Counts and Mean Features Kept by Framework, Global Counts**

| Selection Type | Min Overfit Count | Min Overfit Mean K% | Max Test Count | Max Test Mean K% | Strongest Peak Count | Strongest Peak Mean K% | Earliest Peak Count | Earliest Peak Mean K% |
|---|---|---|---|---|---|---|---|---|
| mRMR | 5 | 21.36 | 5 | 75.76 | 7 | 74.85 | 4 | 72.46 |
| Mutual Information | 9 | 37.51 | 10 | 54.59 | 11 | 54.05 | 12 | 46.28 |
| ReliefF | 14 | 20.18 | 2 | 90.00 | 0 | 0.00 | 2 | 49.85 |
| SHAP | 10 | 16.42 | 24 | 46.21 | 25 | 43.60 | 26 | 39.38 |

Using these frameworks, SHAP-based MAX and SUM strategies are compared against MI, ReliefF, and mRMR. The counts above indicate the number of dataset-model combinations that the technique ranked best under each framework.

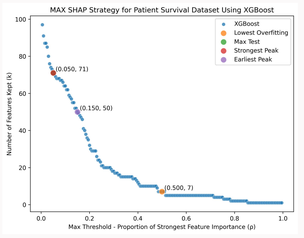
SHAP outperforms filter methods in three of the four frameworks: maximum test, strongest peak, and earliest peak. ReliefF outperforms SHAP in the minimum overfitting framework, because it consistently underfits, lowering both CV and test scores.

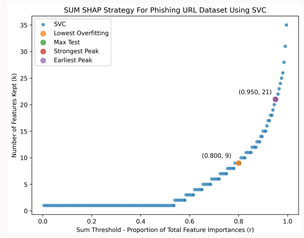
SHAP consistently selects fewer features while maintaining or improving performance, demonstrating stronger feature rankings. Across models, SHAP performs best for Decision Trees, Logistic Regression, and often outperforms MI for Random Forests, while mRMR works best with SVC, due to redundancy reduction. MI and SHAP have roughly equivalent performance for XGBoost models.

**SHAP MAX and SUM Threshold Counts by Framework**

| Selection Type | Min Overfit Count | Max Test Count | Strongest Peak Count | Earliest Peak Count |
|---|---|---|---|---|
| Max = 0.01 | 2 | 15 | 10 | 10 |
| Max = 0.05 | 4 | 10 | 8 | 3 |
| Max = 0.1 | 7 | 6 | 6 | 5 |
| Max = 0.15 | 3 | 2 | 5 | 7 |
| Max = 0.25 | 4 | 1 | 1 | 2 |
| Max = 0.5 | 14 | 4 | 3 | 3 |
| Sum = 0.5 | 6 | 0 | 0 | 0 |
| Sum = 0.6 | 4 | 0 | 1 | 4 |
| Sum = 0.7 | 1 | 2 | 2 | 2 |
| Sum = 0.8 | 2 | 4 | 7 | 8 |
| Sum = 0.9 | 1 | 1 | 2 | 1 |
| Sum = 0.95 | 2 | 5 | 5 | 5 |

Comparing the SUM and MAX SHAP selection strategies, the MAX strategy tends to perform best across frameworks. Using thresholds ρ between 0.01 and 0.1 grants the best test performances, while using ρ = 0.5 best minimized overfitting by aggressively removing weak features. The SUM strategy performed poorly at all thresholds, in comparison.




The MAX strategy produces stable threshold curves with gradual drops in feature count, whereas the SUM strategy produces curves with long plateaus and sudden jumps, as feature are only added under the SUM strategy if the cumulative importance fails to reach $r \cdot S$. This makes the SUM strategy more stochastic when adding features. The SUM strategy often underperforms as $r \to 1$, since many weak features are added to cover the required cumulative importance, degrading performance.

**Time Performance Comparisons**

| Selection Type | Average Time (s) |
|---|---|
| mRMR | 972.58 |
| Mutual Information | 17.43 |
| ReliefF | 495.78 |
| SHAP | 12,606.50 |

Although SHAP delivers superior feature reduction and strong predictive performance, it has the highest computational complexity, being 10x to 55x slower than the other selection methods. SHAP computation scales with both the number of features and number of instances, resulting in poor time performance. MI is the fastest and second most useful method, while SHAP provides the most effective and model-informed feature rankings, offering the best performance when its computational cost is acceptable.

## CONCLUSIONS

Our experiments show that SHAP is a highly effective feature selection technique. While originally created for the purpose of machine learning explainability, SHAP values can be repurposed into a technique for model-agnostic feature ranking that is able to leverage the patterns discovered by models to choose high performance features. SHAP feature subsets exhibit equivalent or superior performance to feature subsets selected by common filter-based feature selection techniques, while resulting in greater feature reduction.

The MAX and SUM algorithms created for these experiments proved effective for choosing the number of features to retain in a way that is simple, intuitive, and forgoes expensive iterative processes. In particular, the MAX strategy provides a natural way to filter out weak features by enforcing features to be relevant with respect to the strongest feature. Our experiments showed that ρ between 0.01 and 0.1 are ideal for most datasets.

Despite the performance benefits of using SHAP as a feature selection technique, our experiments have shown that SHAP is a computationally expensive technique. SHAP scales poorly with dataset size and the size of the feature space, especially when using the model-agnostic Permutation explainer. Time and resources permitting, SHAP produces superior results to the filter methods that were tested, but SHAP may not be worthwhile to implement for sufficiently large and complex datasets.

Future improvements include: modifying the SUM strategy to filter out weak features, creating an automatic or semi-automatic mode for selecting $k$ which requires less human input, and estimating the global SHAP importances to reduce time complexity.

## REFERENCES

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol. 3, pp. 1157–1182, 2003.

[2] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers & Electrical Engineering, vol. 40, no. 1, pp. 16–28, 2014, 40th-year commemorative issue.

[3] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[4] W. E. Marcílio and D. M. Eler, "From explanations to feature selection: Assessing SHAP values as feature selection mechanism," in Proc. 33rd SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI), Porto de Galinhas, Brazil, 2020, pp. 340–347, doi: 10.1109/SIBGRAPI51738.2020.00053.

[5] Y. Gebreyesus, D. Dalton, S. Nixon, D. De Chiara, and M. Chinnici, "Machine learning for data center optimizations: Feature selection using Shapley additive explanation (SHAP)," Future Internet, vol. 15, no. 3, p. 88, 2023, doi: 10.3390/fi15030088.

[6] E. Kraev, B. Koseoglu, L. Traverso, and M. Topiwalla, "Shap-Select: Lightweight feature selection using SHAP values and regression," arXiv preprint, arXiv:2410.06815, 2024, doi: 10.48550/arXiv.2410.06815.

[7] H. Wang, Q. Liang, J. T. Hancock, et al., "Feature selection strategies: A comparative analysis of SHAP-value and importance-based methods," J. Big Data, vol. 11, no. 44, 2024, doi: 10.1186/s40537-024-00905-w.

[8] E. Keany, "BorutaShap: A wrapper feature selection method which combines the Boruta feature selection algorithm with Shapley values," Zenodo, 2020, doi: 10.5281/zenodo.4247610.

[9] J. Verhaeghe, J. Van Der Donckt, F. Ongenae, and S. Van Hoecke, "Powershap: A power-full Shapley feature selection method," in Mach. Learn. Knowl. Discov. Databases (ECML PKDD 2022), M. R. Amini et al., Eds. Cham: Springer, 2023, vol. 13713, pp. 49–65, doi: 10.1007/978-3-031-26387-3_5.