Team Members and Protect Title

Project Title: Project Using SHAP Post-Model Explainability as Feature Selection Technique to Reduce Dimensionality and Retrain Classifiers on More Explainable Data

Team: Joshua Gottlieb, Martin Sichali Chunsen, Alex Yoo

Advisor: Dr. Bathula Krishna

Problem Statement

High dimensionality datasets pose many issues, including high data complexity, low model interpretability, and high expense in gathering, storing and using data. There are many feature selection and dimensionality reduction techniques; however, most of these algorithms are not natively tied to the model algorithms that are used in the final classification, or they transform the features in a way that destroys all data interpretability. We propose a method of using post-model explainability algorithms, namely SHapley Additive eXplanation (SHAP) [reference: 2a] values, to capture the important features learned by a model to use for feature selection for further retraining of the same or new models on the same datasets.

Goal

Our goal is to test a variety of sampling methods to generate approximate global SHAP explanations, as well as to test various selection strategies to extract feature importances from SHAP explanations which will then be used to produce feature-reduced datasets for further retraining. We aim to demonstrate the feasibility and effectiveness of SHAP as a feature selection technique across various machine learning models and datasets.

Objectives

For details on implementation, see Assumptions, Risks, and Obstacles, as well as Methodology.

1. Test the feasibility of SHAP global explanations as a feature selection technique using small, benchmarked datasets. A variety of classifiers will be trained and tuned on these datasets, and then global SHAP explanations will be generated using the PermutationExplainer SHAP [reference: 2c] variant for each classifier. Several feature selection strategies will be tested, and then the classifiers will be retrained using the same hyperparameters on the reduced datasets. Performance of the classifiers pre- and post-feature selection will be compared.

2. Once the feasibility of SHAP for feature selection is established, the same experiments will be performed on more complex datasets with larger numbers of features and samples. It is more computationally expensive to produce global SHAP explanations on larger datasets, so the next hypothesis to test is how well approximate global SHAP

explanations can be generated using various sampling strategies. This objective demonstrates the potential scalability of this technique.

3. Should the prior two objectives be met, the last hypothesis to test, time permitting, is the use of SHAP interactions to create more nuanced explanations, and therefore, under the goals of this project, more robust feature selection. Typical SHAP features do not consider interactions, and thus incorporating interactions is a further refinement that could be tested.

Success Criteria

1. Because this is an investigation into using SHAP for feature selection, the success criteria is simply being able to determine the effectiveness of the technique. In an ideal world, the technique proves to be useful; however, if the technique is shown to be unviable, this is also a meaningful discovery.
2. Using appropriate model-specific evaluation metrics (Accuracy / Area Under Precision Recall Curve / F1-Score / Matthews Correlation Coefficient), the models trained on the SHAP feature-selected reduced datasets should show consistent or improved performance compared to models trained on the original full datasets. This would show the viability of using SHAP explanations for feature selection.

Assumptions, Risks, and Obstacles

1. There are three general classes of feature selection techniques: filter, wrapper, and embedded methods. [references: 4c, 4d, 4e, 4f, 4h]
    a. Filter methods use statistical analyses to select features based on some significance and are model agnostic. Common filter methods include correlation analysis or Mutual Information Gain. Filter methods typically require fewer computing resources but tend to lead to worse classification performance than wrapper or embedded methods.
    b. Wrapper methods involve training a model on different subsets of the features until the best subset is found using some criteria such as accuracy. Wrapper methods typically use a classic machine learning algorithm such as Decision Trees, K-Nearest Neighbors, or Support Vector Machines. Wrapper methods tend to perform better than filter methods but are computationally expensive to compute, as it is infeasible to search all possible combinations of feature space on even smaller feature spaces, as the number of features is of the order 2^N. Wrapper methods must be combined with a search strategy to explore the possible feature space and may not find optimal parameters. Wrapper methods also suffer from worse generalization performance compared to filter methods, as wrapper methods that utilize different machine learning algorithms than the algorithm used for the eventual classification task tend to perform poorly.
    c. Embedded methods encompass any techniques used during or after the training of a model to perform feature selection. Common examples include L_1 Lasso

regularization within linear classifiers or tree pruning and raising in the C4.5 algorithm for decision trees. These methods have the benefit of natively performing feature selection through the process of training the model. An issue with embedded methods is that they are typically specific to each machine learning algorithm.

2. SHAP values are typically used to explain the decisions produced by a black-box model using an additive model of Shapley values. [references: 2a, 3a, 3b, 3c] They are model agnostic and provide a way to explain how features contribute to differences from the average classification decision of the model. These feature explanations can be ordered in terms of magnitude of average impact to construct a global meta-explanation of the most influential features of a trained model. Our assumption is that these explanations can be used as a way to select features.
    a. Using SHAP in this way would count as an embedded method feature selection method, as each SHAP explanation is specific to the model-dataset combination which produced it. An individual SHAP explanation is not expected to generalize to different model types (e.g. an explanation derived from a Decision Tree is unlikely to generalize to an XGBoost or SVM model). However, unlike many embedded methods, because SHAP explanations are model-agnostic, the same process can theoretically be used regardless of model choice.

3. The choice of how many features to use from SHAP analysis determines the feature selection of the final dataset. However, we were unable to find a definitive answer in the literature for how many features to use from the SHAP feature importances for selection. We propose 3 possible strategies, in all cases, the features are ordered by magnitude of absolute mean SHAP value:
    a. Manual K-selection where the top-K features are selected. This is the least robust process; however, it has potential in that it guarantees the size of the resultant feature set. For example, if a user wished to have only 10 features, then exactly 10 features would be selected. This may come at the expense of future model performance.
    b. Percentage of total SHAP contribution. As all of the features in a SHAP explanation are positive, due to being absolute values, one method would be to sum the entire explanation to get a total SHAP contribution. Then, using this total SHAP contribution, features can be added in descending order of contribution until the desired percentage of SHAP contribution is attained. This technique is similar to the percentage of explained variance method when using techniques such as Principal Component Analysis or Linear Discriminant Analysis. A benefit of this strategy is that a certain amount of explanation should theoretically be preserved with the number of features selected; however, if the percentage chosen is high (e.g. 90%), there may be little to no feature selection performed.
    c. Relative strength compared to strongest SHAP feature. This strategy would use the strongest feature as the main comparison point. The user would specify a percentage threshold and each feature's SHAP magnitude would be compared to

the SHAP magnitude of the strongest feature. If the feature is above the user-specified threshold compared to the strongest feature, it is accepted; otherwise, it is rejected and the process terminates, as the features are sorted by descending magnitude. For example, if the strongest SHAP feature had a strength of 1 and the threshold was 10%, features would continue to be added to the reduced feature-set until the feature had a SHAP strength below 0.1. A benefit of this method is that weak features are discarded. A potential downside is that if the SHAP magnitudes are similar across many values (indicating that no one feature individually contributes strongly), then little feature selection would be performed, and if the strongest SHAP feature is much greater than the rest of the features and the threshold is set too high, the feature selection may be too strict.

4. SHAP values are computed locally for each data point and are then aggregated to produce global explanations. There is a great deal of literature covering how to sample feature coalitions to estimate SHAP values at each local data point in an efficient manner; however, we were unable to find any literature considering the difficulty in creating global explanations as the number of samples in a dataset grows. SHAP values are considered to be costly to compute, and thus, computing SHAP values for an entire dataset becomes infeasible as the size of the dataset grows. However, if a properly representative sample of the data is chosen, the approximate global SHAP explanation from that sample should be similar to the true global SHAP explanation. There are three sampling techniques that we wish to test - note that for both the second and third techniques where cross-validation is used, stratified K-folds should be utilized to preserve class balances across folds:
    a. Full dataset sampling, where every data point in the training dataset is used to produce the global SHAP explanation. This is the standard method utilized in the literature and the least scalable; however, it provides a ground truth global explanation that can be used for comparison with other sub-sampling techniques.
    b. Uniform K-fold P-sampling, where P sample data points are drawn from each of the n classes in the dataset uniformly in each of K stratified folds. The results from the cross-validation can be aggregated to produce an estimated global SHAP explanation using the mean rankings generated across each fold. The selection of K and P are likely dependent upon the size of the dataset and available compute power; however, a possible test selection would be P = sqrt(N), where N is the total number of data points in the training dataset, as the square root grows very slowly. The quality of the explanations may suffer from low sample sizes, and thus larger values of P may need to be tested.
    c. Stratified K-Fold P-sampling, where P sample data points are drawn from each of the n classes in accordance to their class balances. This method is equivalent to the prior method except that more explanations are drawn from more common classes. For example, if a dataset has an 80/20 split between class labels 0 and 1, then 0.8P samples would be drawn from samples with label 0 and 0.2P samples would be drawn from samples with label 1 for each of the K folds.

    d. A possible hypothesis to test when using sub-sampling and cross-validation is to change the ranking of features. Typically, features are ranked according to their mean absolute SHAP importance in the global explanation. However, since the global explanation is approximated, each feature has an average and standard deviation of mean SHAP importance and rank across folds. It may be worthwhile to penalize features that have high standard deviation across folds and promote features with low standard deviation across folds through the use of confidence intervals, as this is a measure of stability among the fold explanations in the approximated global explanation.

5. SHAP values are inherently structured on the assumption of an additive linear model with independent features. However, when features are dependent and interact in a non-linear fashion, the Shapley assumptions are violated and the explanation is misleading. There exist methods to estimate the effects of at least first order SHAP interactions; however, the interpretation of the SHAP importances changes. At the time of writing this project overview, there are no plans to address this extension of SHAP values to cover features that are dependent. If there is time, we may try to incorporate SHAP interactions to see if the feature selections created are more robust to noise and more complex data. [reference: 2d]

6. We will test with a variety of classifiers. Models will include:
    a. Decision Trees, Random Forest, XGBoost, and Support Vector Machines, as implemented through the scikit-learn and xgboost Python packages. Each model will be hyperparameter tuned using cross-validation through the GridSearchCV class of scikit-learn. Each of these models are "white-box" or at least partially transparent.
    b. Neural networks are true black-box models; however, their complexity and compute requirements are likely beyond the scope of this project.
    c. Although there exist multiple SHAP algorithms specific to different model types, such as TreeSHAP for tree-based models, we will utilize the Permutation SHAP (PermSHAP) method, as it is model agnostic. [references: 2b, 2c]

Preliminary Literature Review

For convenience, the literature is organized into sections based on the topics and contributions given by the literature. A brief summary of the topic or contribution is included for each source, as well as the number of citations on Google Scholar, if available. Within each section, sources are organized in chronological order.

1. **Previous Experiments using SHAP for feature selection:**
    a. S. B. Cohen, G. Dror, and E. Ruppin, “Feature selection based on the Shapley value,” in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2005, pp. 1–6.

i. Predates the creation of SHapley Additive eXplanations and proposes a filter-wrapper hybrid method for selecting features using pure Shapley coalitions.

ii. 226 citations on Google Scholar

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

i. Experiment using SHAP values for feature selection. Utilized the TreeSHAP algorithm for computing SHAP values and only worked on relatively simple datasets (most datasets had <600 samples and <20 features). Only used XGBoost models, although the authors did test on both classification and regression tasks. Brute forced the number of features to keep by simply keeping a percent of all features.

ii. 487 citations on Google Scholar

c. 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.

i. A SHAP feature selection algorithm which combines the Boruta feature selection algorithm (ref: literature review 4a) with SHAP as a feature importance scoring mechanism.

ii. Does not have an associated paper and is presented as a package on Github.

d. M. Calzolari, *shapicant*, version 0.1.0, 2020. [Online]. Available: https://github.com/manuel-calzolari/shapicant

i. A SHAP feature selection algorithm inspired by PIMP (ref: literature review 4b), using permutation of the response vector and both positive and negative feature importances from SHAP instead of the usual mean absolute SHAP value.

ii. Does not have an associated paper and is presented as a package on Github.

e. J. Verhaeghe, J. Van Der Donckt, F. Ongenae, and S. Van Hoecke, “Powershap: A power-full Shapley feature selection method,” in *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2022)*, M. R. Amini, S. Canu, A. Fischer, T. Guns, P. K. Novak, and G. Tsoumakas, Eds. Cham: Springer, 2023, vol. 13713, pp. 49–65, doi: 10.1007/978-3-031-26387-3_5.

i. Proposes the Powershap algorithm, which is similar to BorutaShap in that random variables are used to iteratively assess the importance of each real feature. Unlike BorutaShap, the entire feature set is not duplicated, and instead only one random feature is tested per iteration and is used to construct a set of distributions of SHAP feature importances for statistical comparison.

ii. 41 citations on Google Scholar

f. 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.

    i. Used SHAP feature explanations to perform feature selection in regression tasks using Random Forest and XGBoost models and TreeSHAP. Utilized the top 10 most important features for feature selection.

    ii. 95 citations on Google Scholar

g. 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.

    i. A fairly new study where the authors used Ordinary Least Squares and Logistic Regression to create statistical t-tests on each of the features generated by SHAP explanations in order to select only features which were considered statistically relevant. Only used XGBoost models. Used a Kaggle credit card fraud detection dataset with >200,000 samples and 30 features, but only tested on a single dataset. Performed no hyper-parameter tuning prior to selection of SHAP features.

    ii. 6 citations on Google Scholar

h. C. Sebastián and C. E. González-Guillén, “A feature selection method based on Shapley values robust for concept shift in regression,” *Neural Comput. Appl.*, vol. 36, pp. 14575–14597, 2024, doi: 10.1007/s00521-024-09745-4.

    i. Uses SHAP in the context of regression tasks to perform backwards selective feature elimination via user-defined prediction quantiles and iteratively adding and removing random variables to test the shift in feature importance under the assumption of data shift.

    ii. 13 citations on Google Scholar

i. 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.

    i. Experiment using SHAP for feature selection on a Kaggle Credit Card Fraud Detection dataset with >280,000 samples and 30 features, 28 of which are PCA-engineered features. Used only tree-based models, including Decision Trees, Random Forests, XGBoost, Cat Boost, and Extremely Randomized Trees. Performed hyperparameter tuning. The analysis between SHAP feature selection and feature selection based on innate model feature importances was relatively inconclusive.

    ii. 211 citations on Google Scholar

j. M. S. H. Shaon, T. Karim, M. S. Shakil, and M. Z. Hasan, “A comparative study of machine learning models with LASSO and SHAP feature selection for breast cancer prediction,” *Healthcare Anal.*, vol. 6, p. 100353, 2024, doi: 10.1016/j.health.2024.100353.

i. Tests SHAP feature explanations for feature selection on a wide variety of models, but only uses a singular dataset: a Breast Cancer dataset with 569 cases and 39 features. Used SHAP to select the top 20 features, but also somehow ended up with only 16 features with seemingly no explanation for how these features were chosen.

ii. 24 citations on Google Scholar

2. **Resources on SHAP and explainability algorithms:**
   a. S. M. Lundberg and S. I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
      i. Original SHapley Additive eXplanations paper.
      ii. 40,957 citations on Google Scholar

   b. S. M. Lundberg, G. G. Erion, and S. I. Lee, “Consistent individualized feature attribution for tree ensembles,” *arXiv preprint*, arXiv:1802.03888, 2018.
      i. TreeSHAP algorithm paper for estimating Shapley values optimized for tree-based algorithms.
      ii. 2,714 citations on Google Scholar

   c. R. Mitchell, J. Cooper, E. Frank, and G. Holmes, “Sampling permutations for Shapley value estimation,” *J. Mach. Learn. Res.*, vol. 23, no. 43, pp. 1–46, 2022.
      i. PermSHAP algorithm paper for estimating Shapley values for model-agnostic estimation.
      ii. 166 citations on Google Scholar

   d. D. Singhvi, A. Erkelens, R. Jain, D. Misra, and N. Saphra, “Knowing your nonlinearities: Shapley interactions reveal the underlying structure of data,” *arXiv preprint*, arXiv:2403.13106, 2024.
      i. Paper discussing methods for estimating and calculating Shapley interactions.
      ii. 0 citations on Google Scholar

   e. C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 3rd ed. 2025. [Online]. Available: https://christophm.github.io/interpretable-ml-book/
      i. A comprehensive book covering many common explainability algorithms, including SHAP.

3. **Explainable AI (xAI) Interpretability Methods:**
   a. L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, “Explaining explanations: An overview of interpretability of machine learning,” in *Proc. 2018 IEEE 5th Int. Conf. on Data Science and Advanced Analytics (DSAA)*, Oct. 2018, pp. 80–89.
      i. A survey on explainable AI methods and a discussion on the motivating purposes for explainable AI.
      ii. 3608 citations on Google Scholar

b. D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, “Machine learning interpretability: A survey on methods and metrics,” *Electronics*, vol. 8, no. 8, p. 832, 2019.

i. Another survey on explainable AI methods and motivating factors for explainable AI.

ii. 2262 citations on Google Scholar

c. P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable AI: A review of machine learning interpretability methods,” *Entropy*, vol. 23, no. 1, p. 18, 2020.

i. A survey of explainable AI methods, including taxonomy.

ii. 3300 citations on Google Scholar

4. **Feature Selection and Dimensionality Reduction Techniques:**

a. M. B. Kursa, A. Jankowski, and W. R. Rudnicki, “Boruta–a system for feature selection,” *Fundamenta Informaticae*, vol. 101, no. 4, pp. 271–285, 2010.

i. Introduces the Boruta algorithm for performing feature selection using random forests by duplicating the feature set to create "shadow" attributes, permuting the shadow features to destroy correlations, and then training random forests on the extended set to extract feature importances. Features are deemed important if their importance value is higher than the maximum importance of the shadow features. This process is completed iteratively until all original features have been accepted or rejected.

ii. 953 citations on Google Scholar

b. A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, “Permutation importance: A corrected feature importance measure,” *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, May 2010, doi: 10.1093/bioinformatics/btq134.

i. Introduces Permutation IMPortance (PIMP) as an algorithm to correct for the bias of the Gini Information in random forest models or of Mutual Information criterion for feature selection by permuting the response vectors, fitting a probability distribution to the null importances, and computing a maximum likelihood estimator.

ii. 2985 citations on Google Scholar

c. G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

i. A short survey of feature selection methods including a brief comparison of the effectiveness of various methods on a number of datasets.

ii. 6549 citations on Google Scholar

d. S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in *Proc. 2014 Science and Information Conf.*, Aug. 2014, pp. 372–378.

i. Discusses many feature subset selection methods, including search strategies and compares them to feature extraction techniques such as Principal Component Analysis (PCA).

ii. 1812 citations on Google Scholar

e. A. Jović, K. Brkić, and N. Bogunović, “A review of feature selection methods with applications,” in *Proc. 38th Int. Conv. on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2015, pp. 1200–1205.
    i. An overview of feature selection methods including taxonomy and applications.
    ii. 1555 citations on Google Scholar

f. B. Xue, M. Zhang, and W. N. Browne, “A comprehensive comparison on evolutionary feature selection approaches to classification,” *Int. J. Comput. Intell. Appl.*, vol. 14, no. 02, p. 1550008, 2015.
    i. A comparison between the time complexity and effectiveness of various filter and wrapper based feature selection approaches.
    ii. 93 citations on Google Scholar

g. N. El Aboudi and L. Benhlima, “Review on wrapper feature selection approaches,” in *Proc. 2016 Int. Conf. on Engineering & MIS (ICEMIS)*, Sept. 2016, pp. 1–5.
    i. A brief overview on wrapper feature selection methods, including search strategies used to explore feature space.
    ii. 260 citations on Google Scholar

h. Y. B. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, “Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy,” *Pertanika J. Sci. & Technol.*, vol. 26, no. 1, 2018.
    i. Another paper which compares the effectiveness of different filter and wrapper based feature selection methods.
    ii. 251 citations on Google Scholar

i. L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint*, arXiv:1802.03426, 2018.
    i. Paper introducing Uniform Manifold Approximation and Projection (UMAP) algorithm for dimensionality reduction
    ii. 19,039 citations on Google Scholar

j. G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, “Analysis of dimensionality reduction techniques on big data,” *IEEE Access*, vol. 8, pp. 54776–54788, 2020.
    i. Discussion dimensionality reduction techniques on big data, primarily Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA)
    ii. 1016 citations on Google Scholar

k. W. Jia, M. Sun, J. Lian, and S. Hou, “Feature dimensionality reduction: A review,” *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2663–2693, 2022.
    i. A survey of various feature selection search strategies, as well as a review of several feature extraction techniques, including PCA, LDA, Independent Component Analysis (ICA) and more.
    ii. 711 citations on Google Scholar

Python Packages

Along with the standard Python packages for machine learning (NumPy, Pandas), the following packages will be used. This list is not comprehensive and more packages may be used as the project unfolds.

1. SHAP Python package for SHapley Additive eXplanations: https://github.com/shap/shap
2. sci-kit Learn Python package for various machine learning algorithms and utility wrappers and methods: https://scikit-learn.org/stable/index.html

Methodology

To satisfy Objective 1:

1. Inspect the simple datasets, perform EDA and data cleaning as needed. However, these datasets should require little cleaning, if any. Perform train-test split according to the appropriate methods outlined in prior benchmarking procedures on these datasets.
2. Train each model type on each dataset and perform hyperparameter tuning.
3. Save models and best hyperparameters for each model.
4. Apply SHAP explainability to trained models for candidate feature selections.
    a. For the objective 1 datasets, use only full-dataset sampling
    b. Test different total features selected for SHAP
5. Create SHAP-reduced datasets using results from step 4, retrain each model on those reduced datasets, using the same hyperparameters determined in step 3. Save these models.
6. Compare the results of all models on test data to test the viability of SHAP feature selection.

To satisfy Objective 2:

1. Inspect the datasets, perform EDA and data cleaning as needed. We should strive to find ultra-clean datasets to reduce the workload needed. Perform train-test split.
2. Train each model type on each dataset and perform hyperparameter tuning.
3. Save models and best hyperparameters for each model.
4. Apply SHAP explainability to trained models for candidate feature selections.
    a. Use full-dataset sampling and test different possible sub-sampling strategies.
    b. Test different total features selected.
5. Create SHAP-reduced datasets using results from step 4, retrain each model on those reduced datasets, using the same hyperparameters determined in step 3. Save these models.
6. Compare the results of all models on test data to test the effectiveness of SHAP feature selection and compare the effectiveness of different sub-sampling strategies.

Potential Datasets

The following datasets are potential choices to use for completing Objectives 1 and 2. The experiment process should not change significantly between datasets and multiple datasets are necessary to ensure the technique is generalizable across domains. Datasets for Objective 1 are small, clean datasets that are commonly used as so-called "Toy" datasets for machine learning. Datasets for Objective 2 are larger, with more samples and more features, and may not be as clean. For each dataset, a brief description of the dataset, the target, the number of samples and the number of features are included. Datasets are organized in ascending order by number of features and then by number of instances.

Simple Datasets for Objective 1:

- B. Ramana and N. Venkateswarlu. "ILPD (Indian Liver Patient Dataset)," UCI Machine Learning Repository, 2022. [Online]. Available: https://doi.org/10.24432/C5D02C.
    - A dataset about liver cirrhosis in patient records from India. The goal is to predict the presence of liver disease. 584 instances, 10 features.
- A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano. "Heart Disease," UCI Machine Learning Repository, 1989. [Online]. Available: https://doi.org/10.24432/C52P4X.
    - Heart disease dataset with the goal of predicting the presence or absence of heart disease. The most common set is the Cleveland database. 303 instances, 13 features.
- "Mushroom," UCI Machine Learning Repository, 1981. [Online]. Available: https://doi.org/10.24432/C5959T.
    - A dataset about mushrooms with the goal of predicting whether a mushroom is edible. 8,124 instances, 22 features.
- K. Cios, L. Kurgan, and L. Goodenday. "SPECT Heart," UCI Machine Learning Repository, 2001. [Online]. Available: https://doi.org/10.24432/C5P304.
    - A dataset describing the diagnosing of cardiac SPECT images. The target is to predict normal and abnormal cardiac behavior. 267 instances, 23 features.
- Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1993). Breast Cancer Wisconsin (Diagnostic) [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.
    - Dataset with the goal to detect the presence of breast cancer. 569 instances, 30 features.

More Complex Datasets for Objective 2:

- D. Wagner, D. Heider, and G. Hattab. "Secondary Mushroom," UCI Machine Learning Repository, 2021. [Online]. Available: https://doi.org/10.24432/C5FP5Q.
    - An extension of the Mushroom dataset with the goal of predicting whether a mushroom is edible. 61,068 instances, 20 features.
- Credit Card Fraud Detection Dataset, version 1, *Zenodo*, dataset published December 4, 2022. doi: 10.5281/zenodo.7395559. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
    - A dataset using PCA transformed features regarding credit card transactions from European cardholders with the goal of detecting fraudulent transactions. 284,807 instances, 30 features.
- A. Prasad and S. Chandra. "PhiUSIIL Phishing URL (Website)," UCI Machine Learning Repository, 2024. [Online]. Available: https://doi.org/10.1016/j.cose.2023.103545.

    - Dataset containing URLs and various features with the goal to predict whether the URL is a phishing attempt or a real website. 235,795, 54 features.
- M. Agarwal, *Patient Survival Prediction [Dataset]*, 2021. [Online]. Available: https://www.kaggle.com/datasets/mitishaagarwal/patient
    - Hospitalization dataset where the target is to determine if a patient died during their time in the hospital. 91,700 instances, 84 features.
- A. Mathur. "NATICUSdroid (Android Permissions)," UCI Machine Learning Repository, 2021. [Online]. Available: https://doi.org/10.24432/C5FS64.
    - Dataset with the goal of determining if an Android application is malware or a legitimate application. 29,333 instances, 86 features.