

# Assessing the Degree of Feature Interactions that Determine a Model Prediction

Krishna Khadka\*, Sunny Shree\*, Yu Lei\*, Raghu N. Kacker†, D. Richard Kuhn†

\*Department of Computer Science and Engineering,

The University of Texas at Arlington,

Arlington, USA

{krishna.khadka, sunny.shree}@mavs.uta.edu, ylei@cse.uta.edu

†Information Technology Laboratory,

National Institute of Standards and Technology,

Gaithersburg, USA

{raghu.kacker, d.kuhn}@nist.gov

**Abstract**—Machine Learning (ML) models rely on capturing important feature interactions to generate predictions. This study is focused on validating the hypothesis that model predictions often depend on interactions involving only a few features. This hypothesis is inspired by  $t$ -way combinatorial testing for software systems. In our study, we utilize the notion of Shapley Additive Explanations (SHAP) values to quantify each feature’s contribution to model prediction. We then use a greedy approach to identify a minimal subset of features ( $t$ ) required to determine a model prediction. Our empirical evaluation is performed on three datasets: Adult Income, Mushroom, and Breast Cancer, and three classification models: Logistic Regression, XGBoost, and SVM. Through our experiments, we find that the majority of predictions are determined by interactions involving only a subset of features.

**Index Terms**—Feature Interaction, Model Prediction, Combinatorial Testing

## I. INTRODUCTION

ML model predictions are made by recognizing interactions among various features. It is important to understand how models’ predictions are made in high-stake fields such as healthcare [28], finance [29], and autonomous driving [30]. Additionally, testing and debugging these predictions are important to ensure their reliability. The identification of a minimal subset of features for each model’s predictions help us to both interpret and fix issues with the model’s predictions.

Many studies have been conducted to study feature importance in ML models, shedding light on the influence of individual features on model predictions. Feature importance can be categorized into two types: model-level and instance-level. Model-level methods, such as Gini importance [9], permutation importance [8], as well as global surrogate models like linear models [25], and decision trees [33], provide insights into a feature’s overall impact on the model decision process for all possible instances. These methods help identify important features and potential biases at the model level but may not explain how those features affect individual predictions. Instance-level methods, like Local Interpretable Model-agnostic Explanations (LIME) [17] and Shapley Additive Explanations (SHAP) [23], focus on explaining individual model decisions.

LIME utilizes a simpler, local model to explain the rationale behind the decision process for an individual instance, while SHAP measures the contribution of individual features to model predictions for a specific instance.

While previous works mostly focus on quantifying feature importance, they do not explicitly identify a minimum subset of features that determines a model’s prediction. In our study, we hypothesize that the model prediction is often determined by the interaction of only a few features, i.e.  $t$  out of  $n$  total features, where  $t$  is typically a small number, while the features that are important for different predictions may be different. This hypothesis is a motivation from  $t$ -way combinatorial testing in software testing, where system failures can be identified with  $t$  feature interaction, where  $t$  is typically a small number. Validating this hypothesis will give confidence and thus enable future work in applying  $t$ -way testing in the domain of machine learning such as model interpretation, testing and debugging.

To empirically validate our hypothesis, we develop an approach to identify a minimal subset of features whose interactions are important for preserving a model prediction. In our approach we use SHAP values [23] to measure the contribution of individual features and their interactions. SHAP values quantify the contribution of each feature to model predictions at an instance-level. These contributions are additive, meaning that their sum captures the total effect of feature interactions on the prediction [24]. Each SHAP value for a feature inherently considers all the possible interactions with other features. Moreover, features can have either positive or negative influences on the model’s classification decision, directing it towards class 1 or class 0, respectively.

To identify a minimal feature subset, we begin with an empty set  $R$  and iteratively add features based on their SHAP values. This process includes supporting and opposing features—those that positively or negatively affect the model’s classification, respectively. The addition of supporting features to  $R$  continues until the cumulative SHAP values in  $R$  are sufficient to maintain the model’s prediction (e.g., maintains the same class prediction or keeps the prediction probability above a certain level), despite potential negative contributions from the opposing

features. This approach is optimal as it iteratively identifies a minimal subset of features, starting with most influential supporting feature.

We conducted experiments on three datasets, Adult Income [21], Mushroom [22], and Breast Cancer [31], using three classification models, Logistic Regression [25], XGBoost [26], and Support Vector Machine [32]. Our results provide strong support for our hypothesis that only a small subset of features are required to preserve model predictions. We observed that the majority of instances achieved accurate predictions with fewer than five interacting features. On average, two or three features were enough to maintain accurate predictions. To verify that the subset of features we identified using our approach are truly determining a model prediction, we retrained the model using the features we identified and then checked if the retrained model produces the same prediction.

The structure of this paper is organized as follows: Section II provides the foundational background. Section III presents our approach to identification of feature subset. Section IV describes our experimental design and presents the results. In section V, we review the related studies. Finally, Section VI concludes the paper, summarizing our findings and suggesting future research directions.

## II. BACKGROUND

### A. Machine Learning Tasks

Machine Learning models are algorithms that generally perform two key tasks: regression and classification. The models are trained with input features and corresponding outcomes, allowing them to learn the relationships between them. Once trained, these models can be used for performing inference tasks. The regression model predicts continuous outcomes, e.g., predicting real estate prices [1] or forecasting weather [2]. On the other hand, classification model predicts discrete labels, e.g., credit card fraud detection [5], spam email detection [3], or diagnosing medical diseases [4].

### B. Decision Making in Classification Task

Consider a dataset  $D$  with  $N$  instances, where  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , where  $x$  is the input feature vector, and  $y$  is the class label. The dataset  $D$  is split into a training set and a test set.  $M$  is a classification model, which maps input features to class labels:  $M : x \rightarrow y$ , parameterized by weight  $\theta$ . Classification task consist of the following two phases:

1) *Training Phase*: During the training phase,  $M$  learns to map input features  $x$  to their corresponding class labels  $y$ . The model  $M$  learns by updating model parameters  $\theta$  during training. The process begins by defining a loss function  $\mathcal{L}((x_i; \theta), y_i, \hat{y}_i)$ , which measures the error between the model's predictions  $\hat{y}_i$  and the actual labels  $y_i$ . The objective is to minimize the loss function  $\mathcal{L}$ . The model parameters are iteratively updated using an optimization algorithm such as gradient descent to minimize the total loss on the training dataset.

2) *Inference Phase*: After the training phase, model  $M$  is applied to unseen test data. The model  $M$  utilizes its learned parameters  $\theta$  to make predictions on test instances. The model is said to be an effective model if it can generalize well on the test set. The performance of the model is evaluated using different metrics such as accuracy, precision, recall, and F1 scores.

*Feature Contribution for Decision Making*: A model's decision is the contribution of individual features and their interactions. Different features may have varying levels of influence on the model's predictions. Some features are determining features, while others may have minimal or no impact. This requires identifying the important features and quantifying their contributions for the final decision. Understanding how different features influence the model prediction helps interpret the model prediction.

### C. SHAP Values

SHAP values are developed based on game theory, and are used to quantify the contribution of each feature to a model prediction. Each feature is treated as a "player" in a game and the prediction is the total "payout" [24]. SHAP values considers all possible combinations of features and ensure a fair distribution of the prediction among all features based on their individual contributions.

These values can be positive or negative, indicating the direction of influence a feature has on the model's prediction. In binary classification, a positive SHAP value adds to the final prediction, indicating a push towards the positive class, while a negative value subtracts from the final prediction, indicating a push towards the negative class. This is important for interpreting the model's behavior due to different feature contributions. In addition, SHAP values provide local interpretability allowing the examination of feature contribution on an individual level.

1) *Calculating SHAP Value*: SHAP Value is calculated with the formula:

$$\phi_j = \sum_{S \subseteq \{1, \dots, p\} \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} (v(S \cup \{j\}) - v(S)) \quad (1)$$

Here,  $\phi_j$  is the SHAP value for feature  $j$ .  $S$  is a subset of features excluding  $j$ ,  $p$  is the total number of features, and  $v(S)$  is the prediction value for the feature set  $S$ . This equation distributes the prediction output fairly among all contributing features based on their individual impact, which is the core principle of SHAP value.

2) *Explaining Predictions Using SHAP Values*: To explain a model prediction with SHAP values involves combining the base value with the SHAP values associated with each feature for the prediction. The base value is the average prediction of the model over the training set, serving as a starting point. SHAP values are then added to the base value (BV), which either increases or decreases the final prediction based on the feature's contribution towards the prediction. Mathematically, a final prediction (FP) can be expressed as.

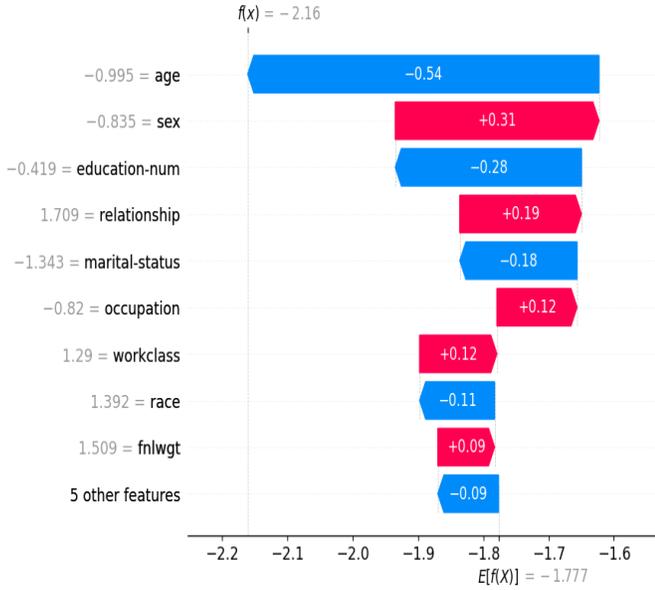


Fig. 1. Waterfall Plot Illustrating SHAP Values For a Class 0 Standardized Sample of Adult Income Predicted by Logistic Regression. The Blue and Red Bars Represent SHAP Values, with Positive Values Favoring Class 1 and Negative Values Favoring Class 0, respectively. Normalized Feature Values Are Displayed to the Left of Respective Feature Names.

$$FP = BV + \sum_{i=1}^N \phi_i \quad (2)$$

where  $N$  is the number of features. The sum of base value and cumulative sum of SHAP values is equal to the final prediction. In binary classification, the final prediction is transformed into a prediction probability through a logistic function. The prediction probability is assigned a class label based on a classification threshold. A common threshold for binary classification is 0.5, which means if the probability is greater than 0.5 then the assigned class is 1; otherwise it is class 0.

3) *Additive Nature of SHAP Values*: The additive nature of SHAP values refers to the fact that the amount of contribution for a feature set is equal to the sum of the amounts of contribution of individual features in the set. In another word, SHAP values inherently captures the effects of feature interactions by considering all possible combinations of features during computation. These SHAP values can be added to the base value to obtain a final prediction.

For instance, Figure 1, presents a waterfall plot displaying SHAP values for an instance in Adult Income Dataset. In this context, the base value is -1.777. To calculate the final prediction for the instance, we sum up the individual SHAP values with the base value, resulting in a sum of -2.16. This summation illustrates the additive nature of SHAP values, showing how they are added together to produce the final prediction. Subsequently, a logistic function is applied to the final prediction to derive the class label.

### Algorithm 1 Feature Subset Identification Using SHAP Values

**Input:** Dataset  $D$ , Machine Learning Model  $M$ , Classification Threshold  $T$

**Output:** List of minimum subsets of features required for each test instance to preserve the model prediction.

- 1: Split  $D$  into training and test sets.
- 2: Train  $M$  on the training set.
- 3: Compute Base Prediction Value, BV, using  $M$  and the training set.
- 4: Initialize a list  $All\_R$  to store subsets for all test instances.
- 5: **for** each instance in the test set **do**
- 6:     Initialize the instance-specific subset  $R$  as empty.
- 7:     Compute SHAP values,  $\phi$ , for all the features.
- 8:     Predict the class label for the instance using model  $M$ .
- 9:     **if** Class label is 1 **then**
- 10:         Identify the supporting set of features,  $S$ , i.e., features with positive SHAP values.
- 11:         Sort  $S$  in descending order based on their SHAP values.
- 12:         Identify the opposing set of features,  $O$ , i.e., features with negative SHAP values.
- 13:         **else**
- 14:             Identify the supporting set of features,  $S$ , i.e., features with negative SHAP values.
- 15:             Sort  $S$  in descending order based on the absolute values of their SHAP values.
- 16:             Identify the opposing set of features,  $O$ , i.e., features with positive SHAP values.
- 17:             **end if**
- 18:             Calculate  $\sum_{i=1}^n |\phi_{O_i}|$ , where  $n$  is the number of features in  $O$ .
- 19:             **for**  $t = 1$  to the number of features in  $S$  **do**
- 20:                 Append feature  $S_t$  to  $R$ .
- 21:                 Compute Margin as  $BV + \sum_{i=1}^t \phi_{S_i} - T$ .
- 22:                 **if**  $|\text{Margin}| > \sum_{i=1}^n |\phi_{O_i}|$  **then**
- 23:                     **Break** the loop,  $R$  now contains a sufficient subset.
- 24:                 **end if**
- 25:             **end for**
- 26:             Append the subset  $R$  for this instance to  $All\_R$ .
- 27:     **end for**
- 28: **Return**  $All\_R$

### III. APPROACH

This section presents our approach that uses SHAP values to empirically validate the hypothesis that only a subset of features are sufficient for a model prediction in a binary classification task. We aim to identify the minimal subset of features required for model predictions, and we provide a detailed description of our approach in Algorithm 1.

#### A. Step 1: Compute and Sort SHAP Values

1) *Computation of SHAP Values (Lines 1-7 of Algorithm 1)*: We begin by training model  $M$  with a training dataset. Next we compute SHAP values,  $\phi$ , for each instance in

TABLE I  
SUMMARY OF DATASETS USED IN THE STUDY

Dataset	Total Instances	Class 0 Instances	Class 1 Instances	Training Instances	Test Instances	Categorical Features	Continuous Features	Classes
Adult Income	48842	37155	11687	39074	9768	8	6	2
Mushroom	8124	3916	4208	6499	1625	22	0	2
Breast Cancer	569	357	212	455	114	0	30	2

the test set. SHAP values can be computed using existing SHAP explainers, e.g., shap.Explainer, shap.TreeExplainer, or shap.KernelExplainer [23]. For our purpose, we use the most commonly used shap.Explainer.

2) *Sorting SHAP Values Based on Predicted Class Label (Lines 8-17 of Algorithm 1)*: Given the model  $M$ , we predict a class label for the test instance under consideration. We then identify the SHAP values based to their contribution towards this prediction. For instances classified as 1, we select features with positive SHAP values, and sort them in a non-increasing order by their magnitude. Likewise, for instances labeled as 0, we choose features with negative SHAP values and sort them similarly in non-increasing order based on their absolute values. These sorted features are supporting features ( $S$ ) for the prediction, while the remaining features are considered as opposing features ( $O$ ). For class 1, the opposing features contain with negative SHAP values, whereas for class 0, they contain features with positive SHAP values.

### B. Step 2: Margin-based Iterative Top $t$ Feature Identification (Lines 18-28 of Algorithm 1)

A minimal set of features sufficient for preserving the model prediction is a subset of features from the supporting set. This subset is determined iteratively by adding features from the supporting set and recalculating the margin after each addition. The margin  $M$  is defined as the difference between the Base Prediction Value (BV) using model  $M$  and the Classification Threshold  $T$ , adjusted by the cumulative SHAP values of the added features. The objective is to find the smallest number of features whose absolute margin exceeds the sum of the absolute SHAP values of the opposing features. We use margin to determine whether the contribution of the top  $t$  features, based on their SHAP values, is sufficient to maintain the original classification of a test instance.

1) *Margin Calculation (Line 14 of Algorithm 1)*: The margin is calculated by the equation:

$$\text{Margin} = \text{BV} + \sum_{i=1}^t \phi_i - T \quad (3)$$

where BV is the base value, which is the average prediction made by the model across all instances. The expression  $\sum_{i=1}^t \phi_i$  represents the aggregate sum of the SHAP values for the top  $t$  features. The term  $T$  is classification threshold.

2) *Iterative Identification of Top  $t$  Features (Lines 15-21 of Algorithm 1)*: We begin with  $t = 1$  and calculate margin using the SHAP value(s) of the top  $t$  features. We then compare the absolute margin against the cumulative sum of absolute SHAP values of all the opposing-direction features.

If the initial set of  $t$  features is insufficient to maintain the classification, i.e. absolute margin is less than the sum of absolute value of the opposing SHAP values, we increment  $t$  to include an additional feature in the margin calculation. We then repeat the evaluation process for identifying the top  $t$  features as described above. This iterative process continues until we find a minimum number of  $t$  features whose cumulative contribution, including the base prediction, preserves the original prediction. Note that this process must terminate since the cumulative contribution of subset of supporting features must match the original prediction.

Through this systematic approach, we leverage SHAP values to identify top  $t$  feature set sufficient for preserving model predictions.

### C. Example

Here, we explain a working example with Figure 1, which displays a waterfall plot of SHAP values for a standardized instance from the Adult Income dataset trained with LR. The horizontal axis represents the log-odds values. The base value log-odds for is  $BV = -1.777$ , the log-odds classification threshold is 0, and the log-odds value of the final prediction is  $-2.16$ . After applying logistic function, the model’s predicted class label for this particular example is 0.

Our initial step starts with sorting the features with negative SHAP values as the class label is 0. The sorted features comprise [age, education-num, marital-status, race, and 5 other features]. For  $t = 1$ , we calculate the margin using the ‘age’ feature’s SHAP value, utilizing Equation 3. The resulting margin is calculated as  $\text{Margin} = -2.317$ .

Next, we calculate the sum of all opposing SHAP values, which is 0.89. To check the classification’s preservation, we compare the absolute value of the margin to the sum of opposing SHAP values. Here, the absolute value of the margin, 2.317, is greater than the sum of opposing SHAP values, 0.89, confirming the preservation of the classification. Thus, for this instance,  $t = 1$  preserves the classification.

## IV. EXPERIMENT AND RESULTS

We aim to validate our hypothesis by answering the following research question:

- How many features determine the prediction of an instance?

#### A. Datasets:

In our experiment, we choose three commonly used real-world datasets for binary classification tasks. Each dataset consists of categorical or continuous features or both types of features. We also have balanced and imbalanced datasets. These variations allow us to test our hypothesis across different scenarios. The datasets include:

- **Adult Income** – The dataset is used to predict whether an individual income exceeds more than \$50,000 per year [21]. It represents an imbalanced dataset with uneven class distribution and includes both categorical and continuous features. It has been widely used in binary classification tasks.
- **Mushroom Dataset** - The Mushroom dataset is used for classification of mushrooms as either edible or poisonous, based on 23 species within the Agaricus and Lepiota families [22]. It consists of categorical features describing physical attributes of mushrooms. It is a balanced dataset, with a nearly equal distribution of classes.
- **Breast Cancer** - The Breast Cancer dataset is for the classification of breast cancer tumors into malignant or benign categories [31]. It consists of 569 instances described by 30 continuous features each. This dataset is utilized for health and medicine research, specifically in tasks related to classification.

For experimental preparation, we perform 80:20 split for training and testing purpose. Data preprocessing includes standard normalization. Table 1 provides a summary overview of these datasets.

#### B. Classification Models:

We use three types of classification models. They are commonly used for tabular data. We train each of the three model types with each of the three datasets and thus obtain a total of nine models:

- **Logistic Regression (LR)** – LR is a simple model in machine learning for classification task [25]. Its linear nature is useful in scenarios where relationship between features and outcomes are expected to be linear.
- **XGBoost** – XGBoost is an implementation of gradient boosted trees [26]. It has high utility in handling large and complex dataset. In addition, it also handles non-linear interactions between features and thus helps understand complex non-linear feature interactions.
- **Support Vector Machine (SVM)** – SVM is a kernel based classification model that uses support vectors to identify an optimal hyperplane to maximize the margin between the classes [31]. SVM is suited for high dimensional features.

The models are trained using scikit-learn library [27] on the training set of each dataset, using standard hyper-parameter values.

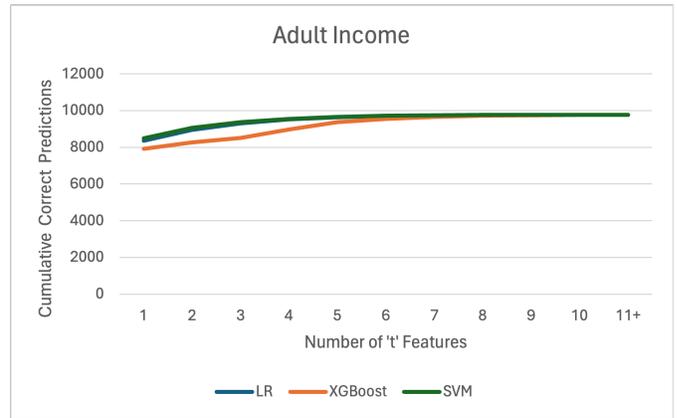


Fig. 2. Adult Income Dataset: Cumulative Predictions Determined by  $t$  Feature

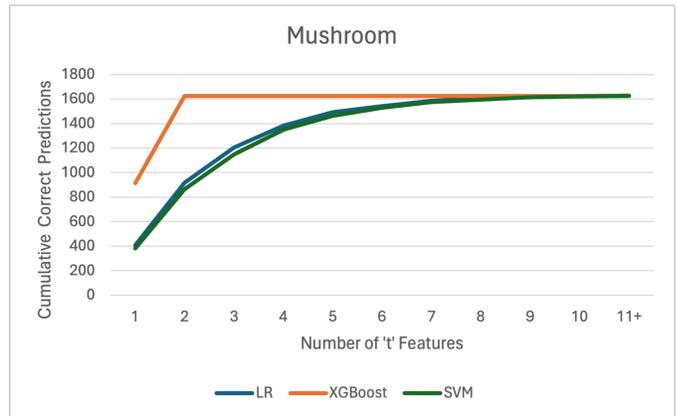


Fig. 3. Mushroom Dataset: Cumulative Predictions Determined by  $t$  Feature

#### C. SHAP Value Computation

SHAP values are computed using the SHAP library [23]. In particular, we use shap.Explainer from the SHAP library.

#### D. Results and Discussion

We begin our results and discussion with an instance-level examination, followed by an aggregated dataset-level analysis.

1) *Instance-Level Analysis:* We identify the number of features that are sufficient to maintain the original prediction for each test instance. Figure 2 presents the results for the Adult Income dataset. As we increase  $t$  from  $t = 1$ , we can see the cumulative number of correct predictions increasing for all models. However, beyond  $t = 6$ , the graph levels off. This suggests that while additional features initially were required for some instances, there comes a point where further additions are not needed. This leveling off is indicative of the diminishing benefit of adding more features. For all models, more than 90% of the instances achieved correct predictions using fewer than 5 interacting features identified using our approach.

Figure 3 shows the results for the Mushroom dataset. Here, we again initiate  $t$  at 1 for each instance. For LR and SVM, the cumulative number of correct predictions gradually increases as we include more features, reaching a plateau after  $t = 7$ .

TABLE II  
STATISTICAL SUMMARY OF  $t$  VALUES

Datasets	Models	Mean	Median	Maximum	Minimum	Total Features
Adult Income	LR	1.32	1	10	1	14
	XGBoost	1.62	1	11	1	
	SVM	1.17	1	10	1	
Mushroom	LR	2.76	2	11	1	22
	XGBoost	1.01	1	3	1	
	SVM	2.96	2	14	1	
Breast Cancer	LR	1.6	1	7	1	30
	XGBoost	1.82	2	7	1	
	SVM	4.05	4	20	1	

TABLE III  
MODEL PERFORMANCE COMPARISON BASED ON RANDOMLY PICKED SAMPLES

Datasets	Models	1		2		3		4		5		Model Accuracy
		t	Acc	t	Acc	t	Acc	t	Acc	t	Acc	
Adult Income	LR	1	0.76	1	0.77	3	<b>0.79</b>	1	0.76	2	0.78	0.82
	XGBoost	4	<b>0.82</b>	5	0.81	1	0.76	1	0.81	1	0.81	0.85
	SVM	1	<b>0.8</b>	5	0.78	2	0.76	1	0.79	1	0.8	0.81
Mushroom	LR	1	0.9	4	<b>0.94</b>	5	0.87	3	0.91	4	0.81	0.93
	XGBoost	1	<b>0.98</b>	1	0.98	1	0.98	1	0.98	1	0.98	0.98
	SVM	2	0.89	4	<b>0.94</b>	4	0.93	2	0.89	3	0.9	0.92
Breast Cancer	LR	3	0.9	2	<b>0.97</b>	2	0.89	1	0.88	2	0.9	0.97
	XGBoost	1	0.84	2	<b>0.95</b>	1	0.82	2	0.93	3	0.92	0.95
	SVM	4	0.82	6	<b>0.98</b>	4	0.93	3	0.86	2	0.84	0.97

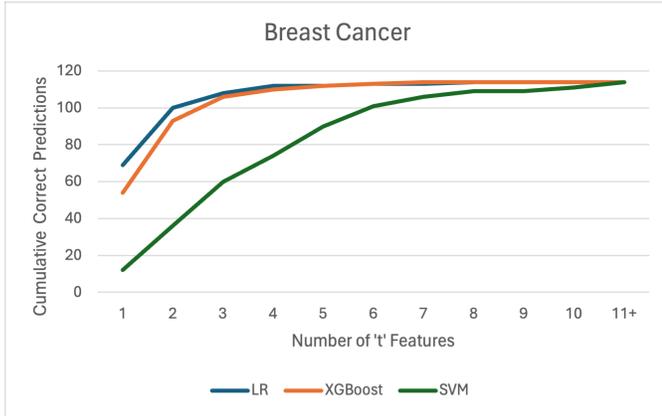


Fig. 4. Breast Cancer Dataset: Cumulative Predictions Determined by  $t$  Feature

This means that additional features beyond this point are not needed to preserve correct predictions. In contrast, we observe a striking trend with XGBoost. The majority of instances attained correct predictions with just  $t = 2$  features. This implies that, two features sufficed to achieve accurate predictions for nearly all instances. This can be attributed to XGBoost’s strong feature selection capabilities.

Figure 4 presents the results for the Breast Cancer dataset.

We start with  $t = 1$  for each instance. For all the three models, the cumulative number of correct predictions increases as we increase more features. The curves reach a plateau after  $t = 7$  which suggests that at most seven features are enough to get most predictions correct.

Many machine learning models may contain redundant or irrelevant features that do not contribute meaningfully to predictions. Our results demonstrate that with just a small subset of features, predictions can be preserved. It’s worth noting that while some instances may require the same number of features to maintain predictions, the specific features that are needed may vary.

2) *Dataset-Level Analysis:* The dataset-level analysis provides insights across various datasets and models. For each model and the corresponding test set, we calculated four key statistical properties: mean value, median value, maximum value, and minimum value of  $t$ . The numbers are shown in Table II.

**Mean Value:** For the Adult Income dataset, all three models had small mean values. It indicates that, on average, only a small subset of features was needed to preserve predictions. SVM had a mean  $t$  of 1.17; LR had a mean  $t$  of 1.32; while XGBoost had a slightly higher mean  $t$  of 1.62. For the Mushroom dataset, the mean value ranges from 1.01 to

2.96. For the Breast Cancer dataset, the mean value ranges from 1.6 to 4.05. Overall, a maximum average of 4 features are sufficient for preserving model predictions across the datasets.

**Median Value:** In both the Adult Income and Mushroom datasets, the median  $t$  value for the majority of instances is only one or two features needed to maintain accurate predictions. For breast cancer, LR has a median of 1, XGBoost has a median of 2, and SVM has a higher median value of 4.

**Maximum Value:** In case of the Adult Income dataset, LR and SVM had a maximum of 10 features for a single instance, while XGBoost needed up to 11 features. Similarly, for the Mushroom dataset, LR required a maximum of 11 features, SVM a maximum of 14 features, while XGBoost needed a maximum of only 3 features. For Breast cancer, LR and XGBoost had a maximum of 7 features, while SVM had a maximum of 20 features. It is interesting to note that these maximum values are significantly lower than the total numbers of features and instances requiring these maximum number of features are very few.

**Minimum Value:** For all instances across all datasets, the minimum  $t$  was consistently 1, indicating that there were instances where only a single feature was sufficient to preserve the original prediction.

In summary, our dataset-level analysis shows that, on average, a small number of features are needed to maintain accurate predictions across various instances.

3) *Validation of Model Predictions with Reduced Feature Set:* To validate that the subset of features we identified using our approach truly determine model predictions, we retrained the model using only the identified features and subsequently examined whether the retrained model produced the same predictions as the original. For this evaluation, we randomly selected five instances from the test set for each combination of dataset and model. The original model was retrained using the features identified as most important for these selected instances. These newly trained models, which used the reduced feature set, were tested on the test set, and their accuracy was compared to models trained with the entire feature set. All the models trained on reduced feature subset have an accuracy within 6% of the original model accuracy. For some instances, such as instance 2 in Mushroom dataset trained on LR, achieved even higher accuracy than the original accuracy with  $t = 4$ . This observation suggests that utilizing only a few features can sometimes produce better results compared to using all features. Detailed results are presented in Table III.

## V. RELATED WORK

Several studies have explored the feature importance in the context of model predictions [10] [7] [6]. Feature importance study can be split into two parts: Model-level and instance-level approaches.

Model-level approaches provide feature importance at the entire dataset level, including feature interaction techniques such as Gini importance [9], permutation importance [8], and Principal Component Analysis (PCA) [15]. Gini importance is based on entropy that measures a feature’s contribution by

checking how often a feature is used to split the data, whereas permutation importance measures the decrease in a model’s accuracy when the feature values are randomly shuffled. Both approaches rank features based on their contribution to model predictions. Principal Component Analysis (PCA) is a statistical analysis for feature reduction, identifying a few features out of the total feature set at the dataset level [15] [16].

Unlike model-level approaches, our approach works at the instance level. In particular, we try to identify a set of features that are sufficient to preserve an individual prediction. As observed in our experiments, the feature sets that preserve individual predictions differ from instance to instance.

Instance-level approaches, e.g. LIME [17] and SHAP [19], provide feature importance at the instance level. LIME constructs an interpretable surrogate model around individual predictions to approximate the original model predictions. These surrogate models closely mimic the behavior of original models, explaining the contributions of individual features to a particular instance. SHAP values are developed based on game theory. They capture the contributions of individual features towards a model prediction by considering all possible combinations of features [19] [20].

Instance-level approaches are developed to explain individual model decisions. They do not identify a minimum set of features for preserving predictions. Note that our approach uses SHAP values to measure the contributions of individual features. In this respect, our approach is complementary to approaches that measure feature importance at the instance level.

Our work is inspired by  $t$ -way combinatorial testing, a software testing approach to analyze the effects of interactions among  $t$  parameters on a system’s output [14]. In particular, our work is inspired by a similar study that investigates the degree of parameter interactions for software failures [34].  $T$ -way testing has also been used in different machine learning tasks for detecting model biases [11], generating synthetic data [12], and testing deep neural networks [13], etc. We hope that our current work provides additional support for the effectiveness of applying  $t$ -way testing to machine learning systems.

## VI. CONCLUSION AND FUTURE WORK

We developed an approach that uses SHAP values to identify a minimal subset of features that preserves a model prediction. We conducted an empirical study to assess the degree of feature interactions that determine individual model predictions, using three datasets, Adult Income, Mushroom and Breast cancer and three classification models, Logistic Regression, XGBoost, and SVM. Our experimental results demonstrate that model predictions could be preserved using fewer than five features for the majority of instances. Furthermore, we verified the identified reduced feature sets by retraining the models using only the features we identified and subsequently checked whether the retrained models produced consistent predictions with the original ones.

In the future, we aim to broaden the scope of our research by considering a variety of machine learning models and diverse datasets to gain a more comprehensive understanding of the

underlying dynamics. We also plan to explore feature subset in multi-class classification tasks. Our long term goal is to pave the way for the application of combinatorial testing in the field of machine learning.

#### ACKNOWLEDGEMENT

This work is supported by a research grant (70NANB21H092) from Information Technology Lab of National Institute of Standards and Technology (NIST).

Disclaimer: Certain equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

#### REFERENCES

- [1] Manjula, R., Jain, S., Srivastava, S., & Kher, P. R. (2017, November). Real estate value prediction using multivariate regression models. In *IOP Conference Series: Materials Science and Engineering* (Vol. 263, No. 4, p. 042098). IOP Publishing.
- [2] Holmstrom, M., Liu, D., & Vo, C. (2016). Machine learning applied to weather forecasting. *Meteorol. Appl.*, 10, 1-5.
- [3] Shams, R., & Mercer, R. E. (2013, December). Classifying spam emails using text and readability features. In *2013 IEEE 13th international conference on data mining* (pp. 657-666). IEEE.
- [4] Singh, P., Singh, N., Singh, K. K., & Singh, A. (2021). Diagnosing of disease using machine learning. In *Machine learning and the internet of medical things in healthcare* (pp. 89-111). Academic Press.
- [5] Ghosh, S., & Reilly, D. L. (1994, January). Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on* (Vol. 3, pp. 621-630). IEEE.
- [6] Mi, X., Zou, B., Zou, F., & Hu, J. (2021). Permutation-based identification of important biomarkers for complex diseases via machine learning models. *Nature communications*, 12(1), 3008.
- [7] Fisher, A., Rudin, C., & Dominici, F. (2019). All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.*, 20(177), 1-81.
- [8] Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.
- [9] Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance?. *Bioinformatics*, 34(21), 3711-3718.
- [10] Rajbahadur, G. K., Wang, S., Oliva, G. A., Kamei, Y., & Hassan, A. E. (2021). The impact of feature importance methods on the interpretation of defect classifiers. *IEEE Transactions on Software Engineering*, 48(7), 2245-2261.
- [11] Patel, A. R., Chandrasekaran, J., Lei, Y., Kacker, R. N., & Kuhn, D. R. (2022, April). A combinatorial approach to fairness testing of machine learning models. In *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 94-101). IEEE.
- [12] Khadka, K., Chandrasekaran, J., Lei, Y., Kacker, R. N., & Kuhn, D. R. (2023, April). Synthetic Data Generation Using Combinatorial Testing and Variational Autoencoder. In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 228-236). IEEE.
- [13] Chandrasekaran, J., Lei, Y., Kacker, R., & Kuhn, D. R. (2021, April). A combinatorial approach to testing deep neural network-based autonomous driving systems. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 57-66). IEEE.
- [14] Bryce, R. C., Lei, Y., Kuhn, D. R., & Kacker, R. (2010). Combinatorial testing. In *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization* (pp. 196-208). IGI Global.
- [15] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3), 37-52.
- [16] Song, F., Guo, Z., & Mei, D. (2010, November). Feature selection using principal component analysis. In *2010 international conference on system science, engineering design and manufacturing informatization* (Vol. 1, pp. 27-30). IEEE.
- [17] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).
- [18] Marcílio, W. E., & Eler, D. M. (2020, November). From explanations to feature selection: assessing SHAP values as feature selection mechanism. In *2020 33rd SIBGRAPI conference on Graphics, Patterns and Images (SIBGRAPI)* (pp. 340-347). Ieee.
- [19] Lee, Y. G., Oh, J. Y., Kim, D., & Kim, G. (2023). Shap value-based feature importance analysis for short-term load forecasting. *Journal of Electrical Engineering & Technology*, 18(1), 579-588.
- [20] Liu, Y., Liu, Z., Luo, X., & Zhao, H. (2022). Diagnosis of Parkinson's disease based on SHAP value feature selection. *Biocybernetics and Biomedical Engineering*, 42(3), 856-869.
- [21] Becker, Barry and Kohavi, Ronny. (1996). Adult. *UCI Machine Learning Repository*. <https://doi.org/10.24432/C5XW20>.
- [22] Mushroom. (1987). *UCI Machine Learning Repository*. <https://doi.org/10.24432/C5959T>.
- [23] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- [24] Shapley, Lloyd S. "A value for n-person games." *Contributions to the Theory of Games* 2.28 (1953): 307-317
- [25] Cox, D. R. (1958). The regression analysis of binary sequences (with discussion). *Journal of the Royal Statistical Society, Series B (Methodological)*, 20(2), 215-242.
- [26] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM.
- [27] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- [28] Ahmad, M. A., Eckert, C., & Teredesai, A. (2018, August). Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics* (pp. 559-560).
- [29] Kovalerchuk, B., Vityaev, E., Demin, A., & Wilinski, A. (2023). Interpretable Machine Learning for Financial Applications. In *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook* (pp. 721-749). Cham: Springer International Publishing.
- [30] Chen, J., Li, S. E., & Tomizuka, M. (2021). Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 5068-5078.
- [31] Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1995). Breast Cancer Wisconsin (Diagnostic). *UCI Machine Learning Repository*. <https://doi.org/10.24432/C5DW2B>.
- [32] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4), 18-28.
- [33] Quinlan, J. R. (1986). *Induction of Decision Trees*. *Machine Learning*, 1(1), 81-106.
- [34] Kuhn, D. R., Wallace, D. R., & Gallo, A. M. (2004). Software fault interactions and implications for software testing. *IEEE transactions on software engineering*, 30(6), 418-421.