Journal of Scientific and Technological Research, Vol.5 (1), 2023
https://doi.org/10.59738/jstr.v5i1.23(17-26).eaqr5800

Print ISSN: 2309-7167
e- ISSN: 2959-7943

বাংলাদেশ উন্মুক্ত বিশ্ববিদ্যালয়
BANGLADESH OPEN UNIVERSITY

Journal of Scientific and Technological Research

Understanding Model Predictions: A Comparative Analysis of SHAP and LIME on Various ML Algorithms

Md. Mahmudul Hasan*
School of Science and Technology, Bangladesh Open University, Gazipur-1705, Bangladesh.

Abstract

To guarantee the openness and dependability of prediction systems across multiple domains, machine learning model interpretation is essential. In this study, a variety of machine learning algorithms are subjected to a thorough comparative examination of two model-agnostic explainability methodologies, SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations). The study focuses on the performance of the algorithms on a dataset in order to offer subtle insights on the interpretability of models when faced with various algorithms. Intriguing new information on the relative performance of SHAP and LIME is provided by the findings. While both methods adequately explain model predictions, they behave differently when applied to other algorithms and datasets. The findings made in this paper add to the continuing discussion on model interpretability and provide useful advice for utilizing SHAP and LIME to increase transparency in machine learning applications.

Keywords: Machine Learning, SHAP, LIME.

1. Introduction

Model interpretability has become a crucial issue in the field of machine learning, especially when using predictive algorithms in industries with high stakes, including healthcare, banking, and customer churn prediction. By ensuring that these models' inner workings are understandable, interpretability promotes confidence and empowers stakeholders to make wise decisions based on model predictions. There has never been a greater demand for efficient and clear model explanations due to the complexity of machine learning techniques. By conducting a thorough comparative analysis of two well-known model-agnostic explainability techniques, SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), across a wide range of machine learning algorithms, this research aims to address the imperative of model interpretability. We compare six different algorithms-Naive Bayes (Webb et al., 2010), Logistic Regression (LaValley, 2008), Decision Tree (Myles et al., 2004), Random Forest (Breiman, 2001), Gradient Boosted Tree (Natekin & Knoll, 2013), and Multilayer Perceptron (Ramchoun et al., 2016)-to clarify how well these strategies perform in comparison.

We make use of a real-world dataset that relates to Telco Customer Churn prediction to evaluate the efficacy of SHAP and LIME. Customer churn, a pervasive issue in the telecom sector, highlights the need for interpretable models that can support targeted retention initiatives. This dataset acts as a useful testbed, allowing us to assess the explainability methodologies in a setting of complexity and unpredictability found in the real world. There will be more detail about the methodology, experimental design, outcomes, and discussion of our findings in the parts that follow. Insights into the relative

*Corresponding author: Md. Mahmudul Hasan (mahmudulhasan@bou.ac.bd)

Journal homepage: jstr.bousst.edu.bd

strengths and limits of SHAP and LIME on various machine learning algorithms will result from this comparison investigation, providing practitioners and academics with invaluable help in choosing the best explainability strategies for certain modeling scenarios. This research ultimately strives to increase the usability and transparency of machine learning models across a variety of applications.

2. Related Works

Due to its crucial role in increasing the transparency, accountability, and accessibility of machine learning models for decision-makers, model interpretability has attracted a lot of attention recently. These papers examine several approaches to analyzing machine learning models, concentrating on SHAP and LIME in particular. Shapley values and the LIME paradigm are combined in a proposed approach, named LIMASE, to produce locally accurate and understandable justifications for any model (Aditya & Pal, 2022). When SHAP, LIME and MDA are compared for feature selection stability, finds that MDA is not as stable as LIME and SHAAP, with LIME being better suited for human understanding (Man & Chan, 2021). In structure-activity relationship investigations, 2020 offers SHAP as a technique for explaining the activity predictions made by sophisticated machine learning algorithms. As an alternative to the KLIME method suggests LIME-SUP, a locally interpretable model built on supervised partitioning (Hu et al., 2018). Overall, these publications demonstrate how SHAP and LIME are helpful in generating comprehensible justifications for machine learning models. However, thorough comparison of various ML algorithms is needed, as demonstrated in this study.

3. Methodology

The Telco Customer Churn (Momin et al., 2020), a real-world dataset that depicts a common business challenge in the telecommunications sector, is used in this research. Data on customers, including demographics, services used, and previous usage patterns, are included in the dataset. To ascertain whether a client will churn or not is the goal of churn prediction. This issue is presented as a problem of binary categorization. It is frequently helpful to visually investigate the link between input attributes and output labels in order to identify the general trends in the dataset. This provides some insight into the problem's complexity and may influence the model that will be investigated. Six different machine learning algorithms are used in this research, each chosen for its unique properties, in order to thoroughly analyze the interpretability techniques:

- **Naive Bayes:** This is a probabilistic classifier well-known for its efficiency in categorical data analysis and text categorization.
- **Logistic Regression:** A linear model that is frequently employed for binary classification applications, logistic regression offers interpretability through coefficient analysis.
- **Decision Tree:** A non-linear model that provides clearly understandable decision rules by segmenting the feature space depending on attribute requirements.
- **Random Forest:** An ensemble model made up of several decision trees that offers both interpretability and predictive power through feature significance.
- **Gradient Boosted Tree:** A boosting technique noted for its prediction accuracy and model complexity that combines the benefits of decision trees.
- **Multilayer perceptron (MLP):** A deep learning architecture called a multilayer perceptron (MLP) has numerous layers of synthetic neurons and represents a complicated, non-linear model.

In this research two cutting-edge, model-independent explainability techniques are used:

- **SHAP**: SHAP values are produced for each method to give a general knowledge of feature significance and contribution to model predictions. SHAP stands for SHapley Additive Explanations. In order to provide a consistent approach to interpretability, SHAP values are derived using cooperative game theory concepts (Lundberg & Lee, 2017).

- **LIME**: By fitting interpretable models to disturbed portions of the data, LIME (Local Interpretable Model-Agnostic Explanations) provides local explanations. This method offers insights into local decision limits and facilitates the interpretation of forecasts for specific situations (Ribeiro et al., 2016).

On the churn dataset a number of models are trained and assess how well they perform. We will make use of the model and the sklearn package to make this procedure simpler. To train the model while tracking train and test accuracy the model. Fit () API is used. Figure 1 shows the subsequent model for explainability of ML Algorithms.

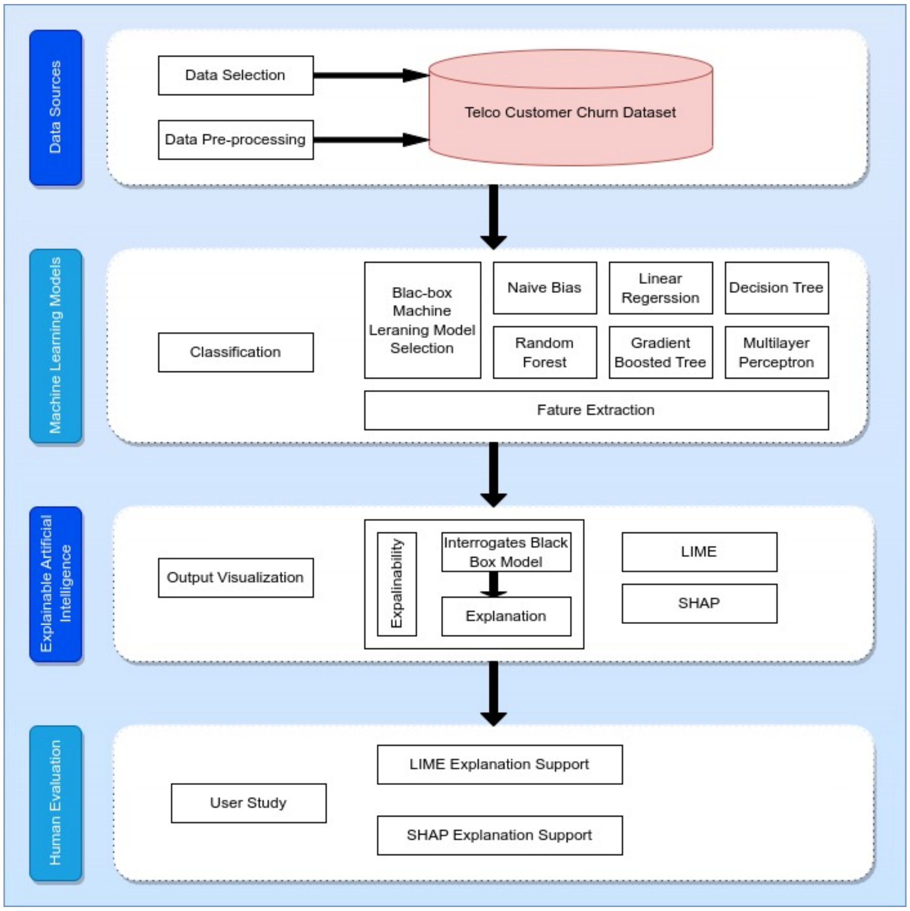


Figure 1: Proposed Model for Explainability of ML Algorithms

Pair plot visualizations in Figure 2 of continuous variables in the dataset are used for constructing correlation plots. Some characteristics in these graphs are binned and replaced with their mean values to decrease the computational complexity. A brief assessment of the results here indicates that consumers that left seem to be paying more each month. Both linear (PCA) and non-linear (UMAP) dimensionality reduction approaches are used in the dataset's features as a first step in dimensionality reduction. Below are the PCA and UMAP results.

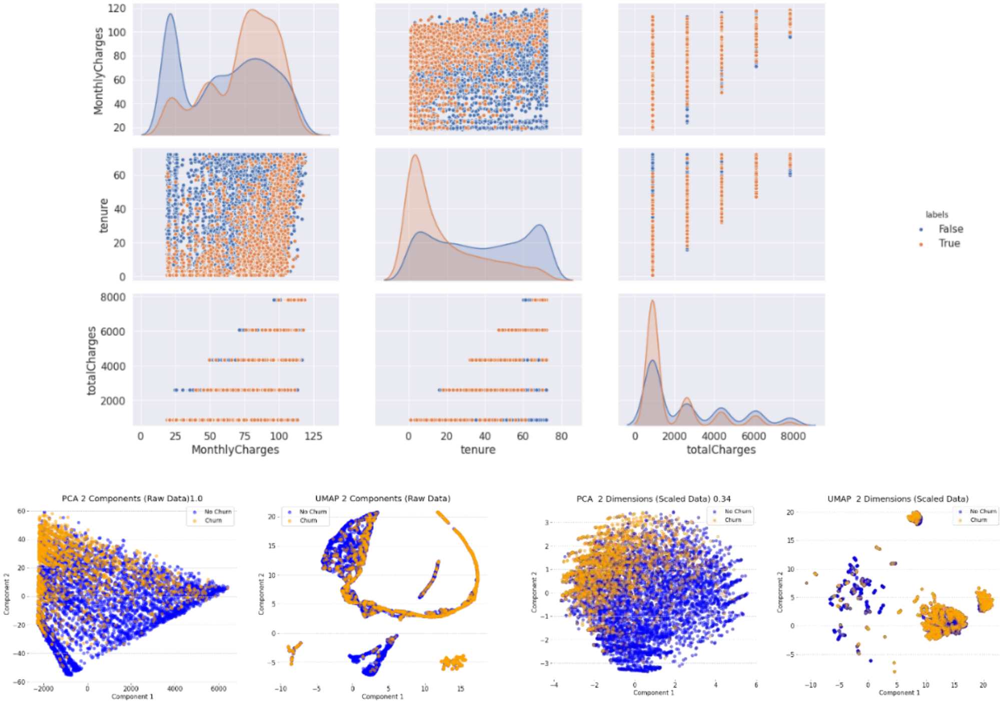
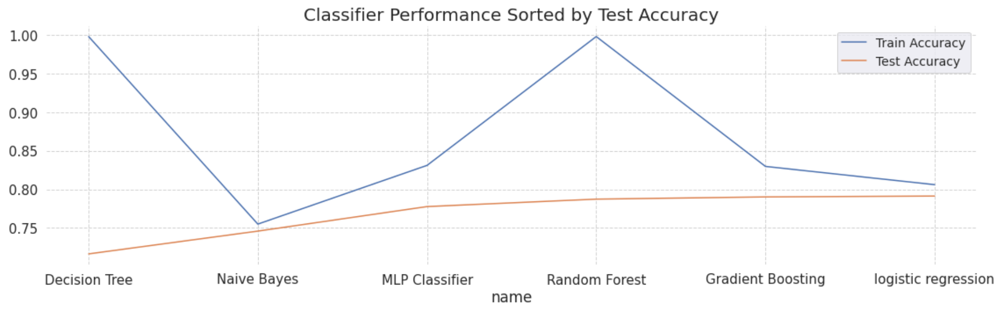


Figure 2: Data Visualization

Both techniques are used to decrease the input feature's dimension to two dimensions, which is then shown as a scatter plot. For the given amount of variation explained PCA findings do not clearly distinguish between data points of churning customers and non-churning customers. This study shows that models that can simulate non-linear patterns are more suited for this job.

4. Results and Discussion

Figure 3 displays the performance of six classifiers and their training time. The Decision Tree and Random Forest models perform poorly on the test set while having excellent train accuracy, which suggests overfitting for this issue. For this dataset, gradient boosted trees and logistic regression provide a decent balance of train/test accuracy.



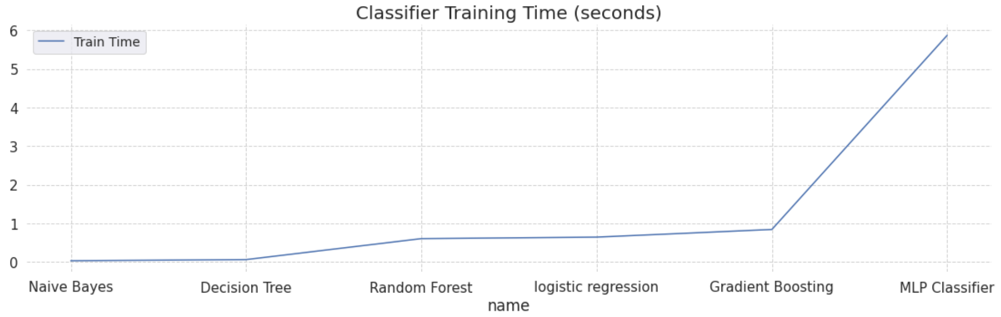


Figure 3: Train and Test Accuracy

Now there are a few trained models that can be utilized to make predictions. This predictions forecast the likelihood that a cable client would churn based on the data for each individual customer. What is not quite evident, though, is how each of these characteristics influences the anticipated churn likelihood. These explanations can be seen from a global perspective (how does each feature affect results on average for the whole dataset?) or a local one (how does each feature affect forecasts for a specific customer?). Some models come with built-in characteristics that offer these kinds of justifications. Examples of these, often known as white box models, include Decision Trees (feature importance), logistic regression (model coefficients), and linear regression (model coefficients).

A. *Global Explanation*

For models such as linear and Logistic Regression in Figure 4 the model coefficients to infer feature importance (note that coefficients need to be interpreted with care for each model type). This gives some idea of how an increase/change in each feature might result in a change in the log odds that the customer will churn and get a general understanding of how important a feature is for the entire dataset.

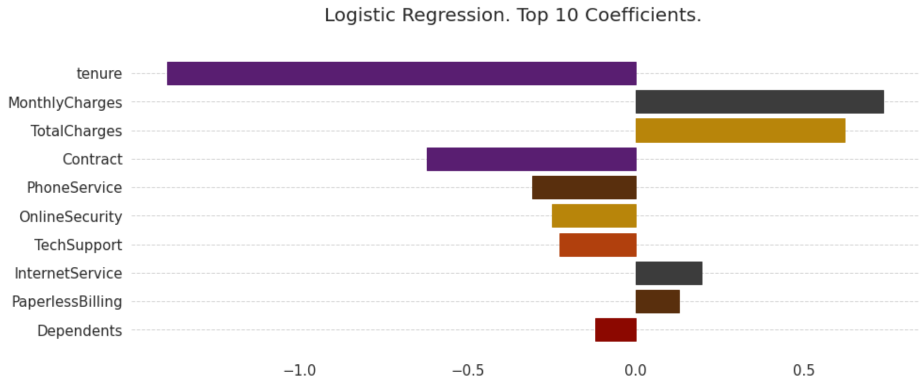


Figure 4: Train and Test Accuracy

B. *Explanations via Feature Importance Scores [Tree Based Models]*

The qualities of tree-based models shown in Figure 5 can be used to infer the significance of a feature. The average reduction in impurity for each feature for each decision tree, i.e., the significance of the feature in terms of lowering the uncertainty (classifiers) or variance (regressors) of the decision tree prediction. The gini significance score is another name for this number. In the list of trained models, we can see the average relevance of each variable for each tree-based model.

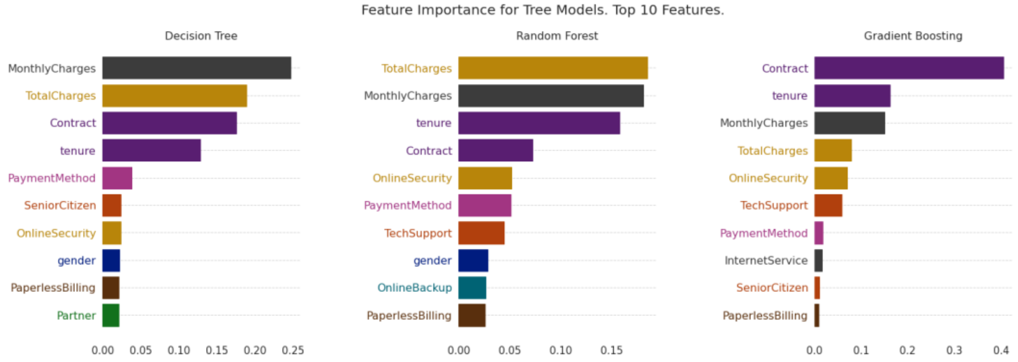
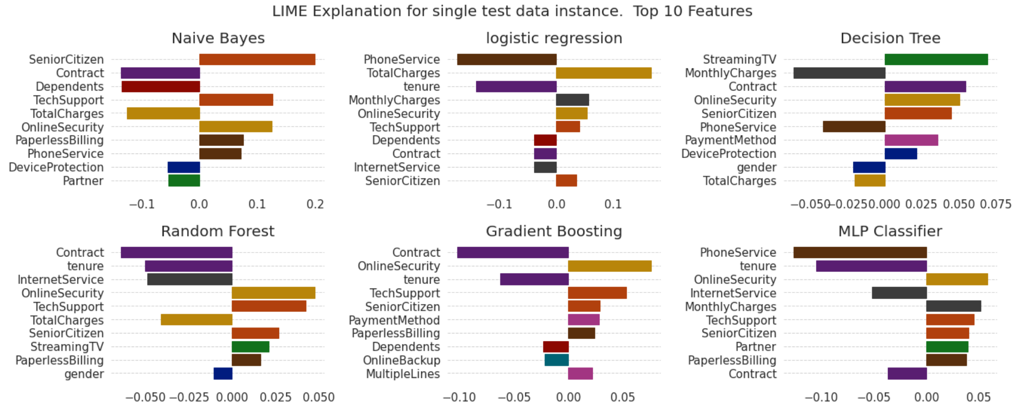


Figure 5: Tree Based Models

These numbers for feature significance seem intriguing. Monthly Charges, Total Charges, Contract, and Tenure consistently score in the top 5 salient aspects according to all 3 models, indicating their importance but with some drawbacks. Because the feature significance ratings are relative, it is challenging to evaluate them in light of the expected result. Although they state that under the Decision Tree model, Total Charges is comparatively more relevant than Contract, they do not state how much a $1 USD rise in Total Charges affects the likelihood of a customer leaving. Global estimations over the full training dataset are used to calculate feature significance metrics. Within the same model, the order and amount of feature significance may vary for a subset of consumers.

C. *LIME Tabular Explainer*

Explain a test data instance for all models. In the following section there are visualizations LIME explanations for a given data point in the test set. Figure 6 shows the LIME explanation for single data instance with runtime.



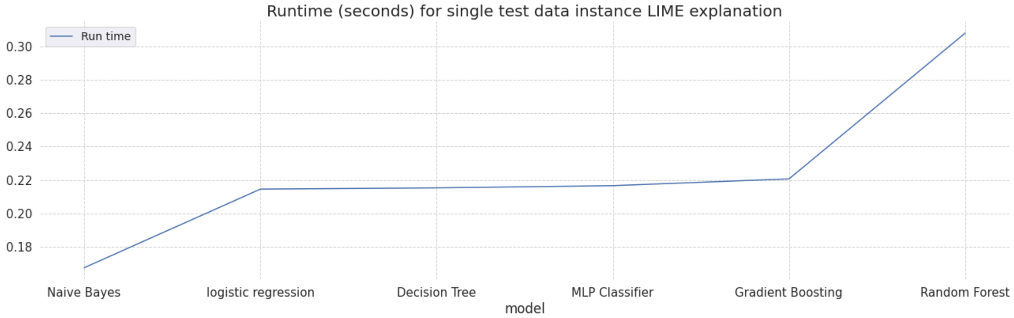


Figure 6: LIME Explanation of Test Data

In order to obtain local explanations, the LIME method employs an approximative linear model. This explanation model might have flaws just like any other ML model. So to increase trust in an explanation's quality, verify that the local model is indeed a decent approximator of the original model as a first step. Figure 7 shows the comparison of LIME actual and local prediction.

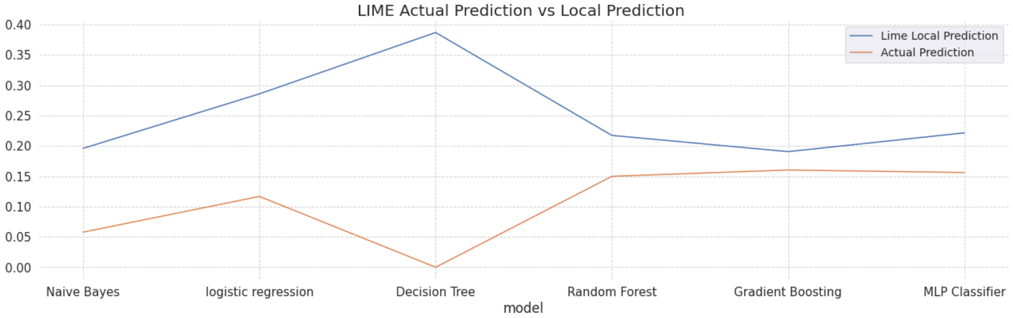


Figure 7: LIME Local Prediction vs Actual Prediction

The predictions provided by the LIME local model and the original model for the stated data instance are depicted in the plot above. Both figures ought to be close. If they are not, it may be difficult to believe the explanation when this happens. There are only a few options available. LIME's settings might be changed to produce a clearer explanation. For instance, increasing the kernel width or LIME perturbations, enhance the initial model as in this instance, the Decision Tree has overfitting symptoms.

The following technique of explanation (SHAP) seeks to deal with these discrepancies.

D. SHAP

Implementations of a variety of Shapley value-based explanations are available in the SHAP library. Outcome is shown in Figure 8. These comprise the TreeExplainer, the DeepExplainer and GradientExplainer, and the KernelExplainer.

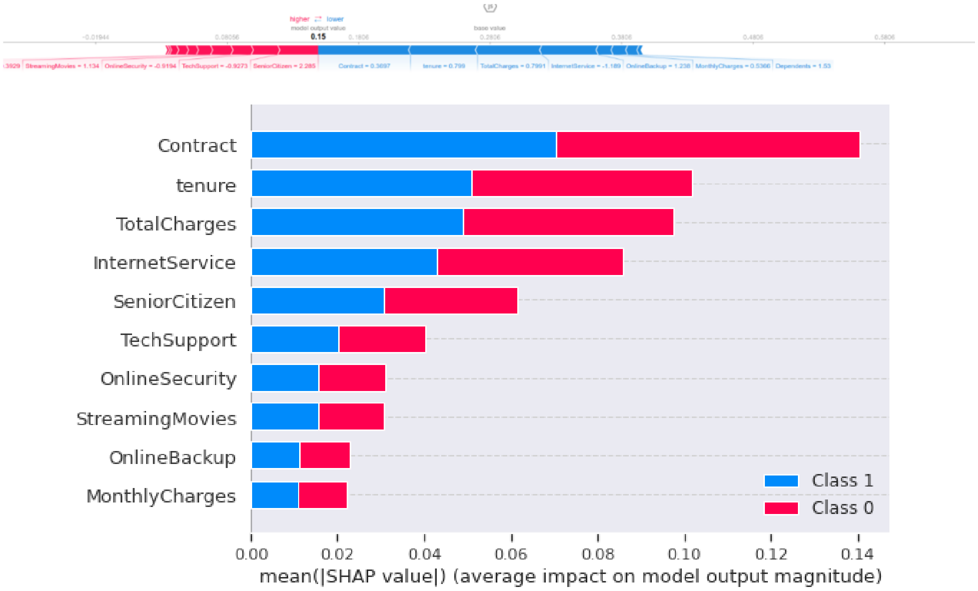


Figure 8: SHAP Library Implementation

On Random Forests, TreeExplainer takes 0.091 seconds. In contrast, KernelExplainer only needs > 12.15 seconds (which is > 100 times quicker). Figure 9 displays the Kernel SHAP explanation for test data.

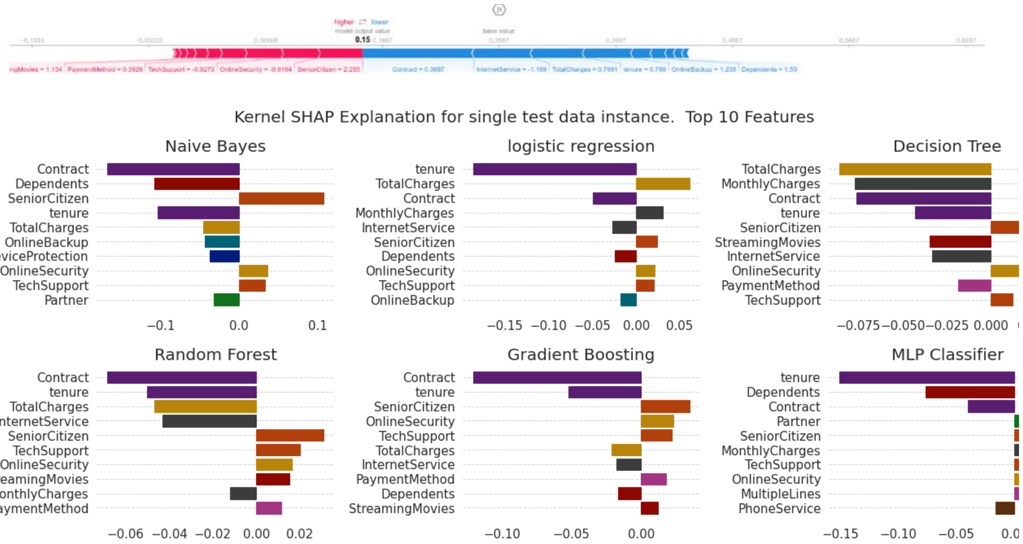


Figure 9: Kernel SHAP Explanation for Test Data

Figure 10 shows the comparison of run time between Kernel SHAP and LIME for a single instance of the dataset.

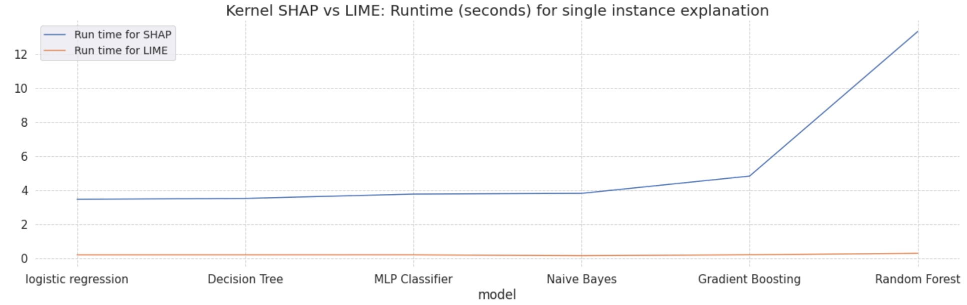


Figure 10: Kernel SHAP vs LIME Runtime

The results mentioned earlier call into doubt the efficacy of explanations when a subsample is employed as background information for SHAP. Figure 11 shows the comparison of Kernel SHAP ran time vs background data size.

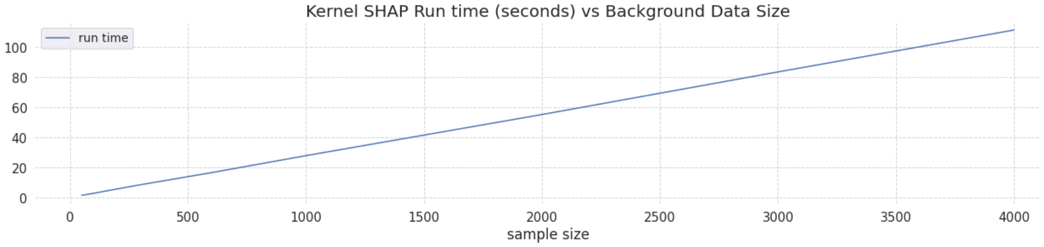


Figure 11: Kernel SHAP Ran Time vs Background Data Size

This problem is avoided by the TreeExplainer since, in accordance with the SHAP documentation, it does not call for a backdrop dataset. Figure 12 displays the expected values vs mean predicon.

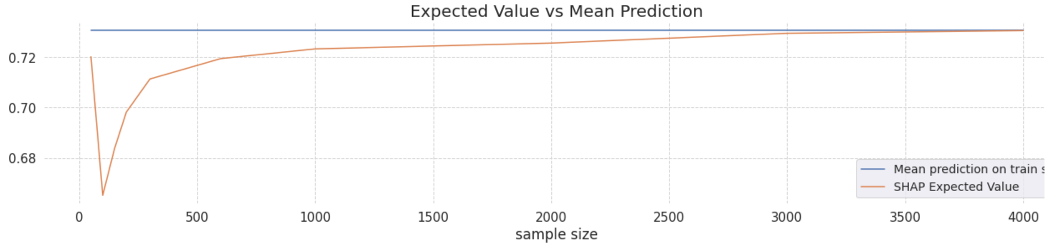


Figure 12: Mean Prediction on Train Set vs SHAP Expected Value

SHAP explains the deviation of a prediction from the expected/baseline value which is estimated using the training dataset. Depending on the specific use case, it may be more meaningful to compute the expected value using a specific subset of the training set as opposed to the entire training set. For example, it may be more meaningful to explain a churn prediction with respect to how it deviates from customers who did not churn. LIME works with models that output probabilities for classification problems. Models like SVMs are not particularly designed to output probabilities (though they can be coerced into this with some issues.). This may introduce some bias into the explanations.

5. Conclusion

Both LIME and SHAP are effective tools for model explanation. Theoretically, SHAP is the superior strategy because it offers mathematical assurances for the precision and coherence of justifications. Even with approximations, the model-neutral SHAP implementation (KernelExplainer) is sluggish in reality. Tree-based models can be benefited from improvements included into SHAP TreeExplainer (up to 100x faster than KernelExplainer), this performance issue is considerably less of an issue. Finally, this research offers important perspectives on the comparison of SHAP and LIME across various machine learning techniques. Algorithmic context and application-specific interpretability requirements should guide the selection of interpretation approach. These findings add to the body of information on model interpretability and help practitioners choose explainability strategies in a well-informed manner.

References

Aditya, P. S. R., & Pal, M. (2022). *Local Interpretable Model Agnostic Shap Explanations for machine learning models*. https://doi.org/10.48550/ARXIV.2210.04533

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Hu, L., Chen, J., Nair, V. N., & Sudjianto, A. (2018). *Locally Interpretable Models and Effects based on Supervised Partitioning (LIME-SUP)*. https://doi.org/10.48550/ARXIV.1806.00663

LaValley, M. P. (2008). Logistic regression. *Circulation*, *117*(18), 2395–2399.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *30*.

Man, X., & Chan, E. P. (2021). The Best Way to Select Features? Comparing MDA, LIME, and SHAP. *The Journal of Financial Data Science*, *3*(1), 127–139. https://doi.org/10.3905/jfds.2020.1.047

Momin, S., Bohra, T., & Raut, P. (2020). Prediction of customer churn using machine learning. *EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing: BDCC 2018*, 203–212.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *18*(6), 275–285.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*, 21.

Ramchoun, H., Ghanou, Y., Ettaouil, M., & Janati Idrissi, M. A. (2016). *Multilayer perceptron: Architecture optimization and training*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. *Encyclopedia of Machine Learning*, *15*(1), 713–714.