Explore the Stability of SHAP Explained Feature Importance

Yanyu, Chen[*]

New York University

Abstract

This project mainly focuses on probing the stability of feature importance explanation made by the Shapley values when the inherent correlation and class balance of training data change. 3 different ways to generate synthetic data sets from original data given by users are designed. By using different types synthetic data set to train the XGBoost classifier, and comparing change in features' shapley values corresponding to each type, we can verify that explaining features' contribution to model prediction results is not perfectly stable. Hence, the interpretability of black box machine learning classification models, explained by SHAP, is harmed. Bias in models can generate easily as a consequence. An interactive tool is built to visualize this change in shapley values. Yet, this project also illustrates that stacking the features' shapley values calculated given different training sets, and use the stacked bar plot to explain importance, might help increase stability. So the SHAP explainer is still useful to interpret black box models' results. The interactive tool's Git-hub repository is *Git-hub YC*.

1 Introduction

In recent years, we have noticed many novel applications of machine learning algorithms to help develop self-driving cars, influence police decision-making [1], and predict individuals' future criminal probability to make trial sentencing. However, advanced technology promotion usually accompanied with unexpected social problems. The article "Machine Bias", written by Angwin et.al [7], indicates the software generated by black-box machine learning model can give biased prediction towards individuals from different race. And such bias is hidden inside the model as well as original train data. If the bias can not be abandoned, the society will be hurt in whole. Hence, it's important to have a global method that can help scientists debug the train data bias by explicitly and visually interpreting features attribution to model prediction.

SHAP(SHapley Additive exPlanations) explainer by Lundberg and Lee, 2017 [8] is a popular model-agnostic method applied to globally interpret the feature attribution to prediction results of black-box models [9]. And shapley values is a mathematical concept that used in game theory. Although numerically, since marginal distribution of each data instance (x) are included, the SHAP method to explain feature attribution costs greatly, it still has been applied in many fields including finance to debug models.

For instance, Jabeur et.al [4] used the SHAP explainer to attribute the proportion each features contribute to the gold price forecasting by XGBoost Regression model. In particular, they used the SHAP explainer API to draw the static plot of feature importance and then interpret the model, as Figure 1 describes. However, Kumar et.al [5] suggested that the explanation of feature importance calculated by shapley values is mathematically unreliable. In general, he explained the SHAP explainer is not globally applicable even though it attribute each individual data.

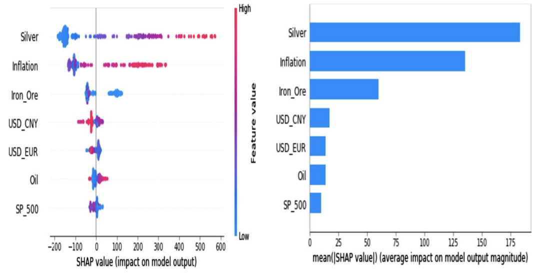

Figure 1: Quoted from Jabeur et.al ”Forecasting gold price with the XGBoost algorithm and SHAP interaction values”

Insipred by Jabeur's and Kumar's works, in order to find the level of SHAP method's reliability in explaining feature attribution. I plan to verify whether the feature importance results explained by shapley values are stable when input train data change. Specifically, when the interventional correlation of features change, fully redundancy are imported, class imbalance happen in train data. I will write codes to generate synthetic data with different characteristics. The written code is expected to function like Barr's package [2]. Then, depending on the type of synthetic train datasets I test, this project has three hypotheses to probe — How much does the shap explained feature attribution change:

- H1: when interventional correlation between features change
- H2: when fully redundancy being added to train data
- H3: when the balance structure of train data change

After the interactive tool is constructed by the test data of gold price forecast, which is collected by myself from various sources, we can use the UCI-Consensus Income Dataset [6] to conduct case study. Interpreting the plots results, we will see that SHAP explainer is not perfectly stable, but still can assist model interpretation if we use the stacked shapley values to explain.

[*] e-mail: yc3823@nyu.edu

2 Related Works

This section described in-depth about the works I mentioned in introduction. Kumar's article, particularly section 3.1.1 – Issue with conditional distribution, explained: "...consider adding feature C to a datasets with two features A and B, so that $P(X_c = X_B) = 1$, and the model f is trained on all there features... B and C should be equally informative and have the same shapley values... so this means $v_{f,x}(B) = v_{f,x}(C) = v_{f,x}(BC)$ and $v_{f,x}(AB) = v_{f,x}(AC) = v_{f,x}(ABC)$...". In view of this description, for any data instance x, we have attribution of feature (ϕ_v):

$$\phi_v(A) = \frac{1}{3}\Delta_v(A,\emptyset) + \frac{2}{3}\Delta_v(A,BC) \tag{1}$$

$$\phi_v(B) = \phi_v(C) = \frac{1}{3}\Delta_v(B,\emptyset) + \frac{2}{3}\Delta_v(B,A) \tag{2}$$

But when the model explanation is limited to only features A and B, the attribution changed to below. Equations (1)(2)(3)(4) are all quoted directly from Kumar.

$$\phi_{v'}(A) = \frac{1}{2}\Delta_v(A,\emptyset) + \frac{1}{2}\Delta_v(A,BC) \tag{3}$$

$$\phi_{v'}(B) = \frac{1}{2}\Delta_v(B,\emptyset) + \frac{1}{2}\Delta_v(B,A) \tag{4}$$

Then in section 3.1.2 – Issue with Interventional Distribution, kumar illustrates another situation: "... a model trained on a data set with three features: X_1 and X_2, both $N(0,1)$, and an engineered feature $X_3 = X_1 \cdot X_2$...the model f is forced to $extrapolate$ to an unseen part of the feature space." Therefore, unlike section 3.1.1, here the features used to make predictions are interventional correlated, and such correlation will influence the calculated shapley value attribute feature importance. In conclusion, kumar wrote:"...By manipulating the model's behavior on unfamiliar parts of the feature space, they can twist the explanations on the familiar part to their will."

Generally, Kumar indicates change in features selected will give rise to change in feature importance explanation by SHAP. However, the total amount of change cannot be seen directly from the equations. Hence, my interactive tool is designed to visually compare the mean shapley value of features of original dataset (i.e only features A and B) and of the synthetic dataset with redundancy (i.e include feature C) in bar plot, line plot, stacked bar plot by the means of Altair tool and Streamlit. To generate the synthetic data sets that probe conditional distribution (kumar, sec.3.1.1) and interventional distribution (kumar, sec.3.1.2), this project largely quoted codes from Barr's [2] package for generating synthetic tabular data out of covariance matrix and by adding redundant features. Yet, instead of generating redundant features by sklearn packages (Barr, 2020), I generate fully redundant features by directly copying $n_{redundant}$ features from original data. Detailed explanation of this method is written in section 3.2 below. Meantime, Barr generated synthetic data directly by input random covariance matrix given by users. However, in order to compare and find change in shapely values explaining feature attribution, we need original data to reference. This project will first ask the users to clean the raw data into clean CSV files, i.e the original data, then ask the users to upload it to the interactive tool and continue synthetic data generation.

3 Methods

This section explains the detailed methods used to generate different types of synthetic data sets. Although this report only discusses the change of SHAP explained feature importance in XGBoost Classifier, the final interactive visualization tool will provide more classification models choice to test, including XGBoost Classifier, Random Forest, and Logistic regression model.

The tool is initially designed and tested by the "Original Data of Gold Price" I collected and cleaned online. Because the features' attribution to gold price prediction has been probed thoroughly nowadays, the testing results can be seen more directly and I specifically referenced Chainani's works [3]. A case study of census income will then be completed by this tool, and its results will be illustrated explicitly in section 4. Both test files of original data and the tool written by Streamlit are uploaded in *my Git-hub repository*.

Generally, three types of synthetic data sets are designed. The first type is by change covariance matrix of original data to generate new tabular data set, the second type is to add fully copied redundant features to train data, and the third type is created by same input covariance matrix and then be sliced into train data of different class balance ratios.

3.1 H1: Change interventional features correlation

$$X_3 = X_1 \cdot X_2 \tag{5}$$

Above relation is directly quoted from Kumar, in which the X_3 is the feature resulted from the intervention between X_1 and X_2. From this equation, a synthetic data set shoul be a tabular data generated by the covariance matrix of $df = df(X_1,X_2) + df(X_3 = X_1 \cdot X_2)$. But in my tool, I decide to try something different. The purpose is still probing how the change in interventional correlation between features will affect the final prediction of class label and how SHAP explainer can disclose this change in shapley values. But the process works as:

- User observe the correlation matrix of the original data.
- Select one base feature (X_{base}) and two features (X_1,X_2) who have the most strong correlation coefficient with X_{base}.
- The tool will change the correlation between X_{base} and X_1,X_2 to near zero, forming new correlation matrix
- Then the correlation matrix will be converted into covariance matrix and "make-tabular-data" function will be used to generate the synthetic data.

Then using this new data set, we train the XGBoost classifier and observing the difference in features' mean shapley value between new data set and original data set.

3.2 H2: Adding fully redundant features

$$X_3 = \left[X_1, \ldots, X_{n_{redundant}} \right] \quad n_{redundant} \leq n_{original\ features} \qquad (6)$$

The second type synthetic data probes the feature redundancy's effect on shapley values and explanation. It references kumar article's section 3.1.1: "...feature C is added directly to data of feature A, B..." In equation (6), X_3 is the redundant features added and it is equal to direct copy of existing features. $n_{redundant}$ determines the number of features copied to form X_3. In our tool, for easier visualization, we let $n_{redundant} = [0, 4, 6]$ and requires the uploaded original data to have at least 6 features. The resulted synthetic data will be: $df = df(X_1, \ldots) + df(X_3)$.

3.3 H3: Change class imbalance

The third type synthetic data will not change the features in train data, but will change the proportion of data instance $(X, y_{class\ 0})$ and $(X, y_{class\ 1})$, then output $df = (X, y_{class\ 0} \cdot ratio_0) + (X, y_{class\ 1} \cdot ratio_1)$. The ratio combo $([ratio_0, ratio_1])$ is set to fixed selection for easier calculation in the visualization tool:

- Balance: [0.5, 0.5]
- Slightly Imbalance: [0.4, 0.6]
- Very Imbalance: [0.1, 0.9]

4 SHAP Explanation Change Results

This section illustrate results of two cases. Sec 4.1 is the results of "Original Data of gold price", and sec 4.2 is the results of "Original Data of census income". Each sub-part will explain results of original correlation matrix and mean shapley values feature attribution of corresponding type of synthetic data. Also, a stacked feature importance bar plot will be drew to show the SHAP explainer might still be reliable if accumulated shapley values are observed. For both cases the generated synthetic data set size is set to $n_{samples} = 800$

4.1 Test data: Original gold price

4.1.1 H1 - gold

From Figure 2, we can see that feature "Dollar index" is strongly correlated with features "Crude Oil" and "Platinum". Hence, we can set $X_{base} = Dollarindex$, and X_1, X_2 = Crude Oil, Platinum in the interactive tool to generate synthetic data by changing original covariance matirx. The y_{label} predicted by the XGBoost classifier is 0 or 1, where class 0 = Gold price will go down tomorrow, and class 1 = Gold price will go up tomorrow.

Observing the line plot in Figure 3, we can see the shapley values of Dollar index increase, while the crude-oil and platinum decreases. This explanation is reasonable and indicates the SHAP explainer is accurate in feature attribution. Previously the prediction results is influenced by the combined efforts of Dollar index, crude oil, and platinum, so an accurate model interpreter should address the importance of all 3 features in final prediction. But when the prediction is influenced by one single feature, which other features can't influence, the interpreter should address more importance on this single feature. Therefore, the SHAP explainer actually works well to capture this interventional correlation change and help users to debug data.

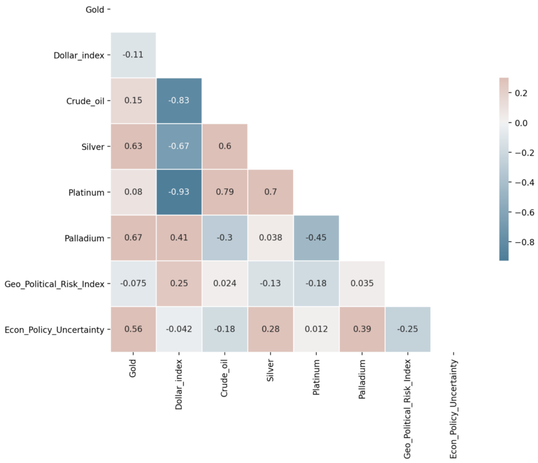

Figure 2: Correlation Matrix of gold price original data

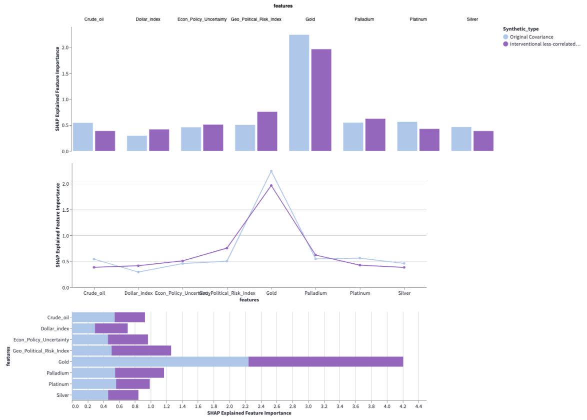

Figure 3: H1 Altair Plots of gold price original data

4.1.2 H2 - gold

Observing the line plot in Figure 4, we can see that importing redundant features change the feature importance attribution by SHAP, and for the feature "Platinum", its feature importance increases gradually as more redundant features are imported. This observation matched the mathematical issue that Kumar elucidated, and exhibit the SHAP interpreter of model can be unstable when the train data include redundancy. In detail, the added features are direct copy of existing features, so the feature attribution shall not varies significantly. However, the shapley values of the copied features change largely enough to see a difference, and the feature "platinum", which is strong correlated to "Dollar index" fluctuates largely, even though it's not the copied features.

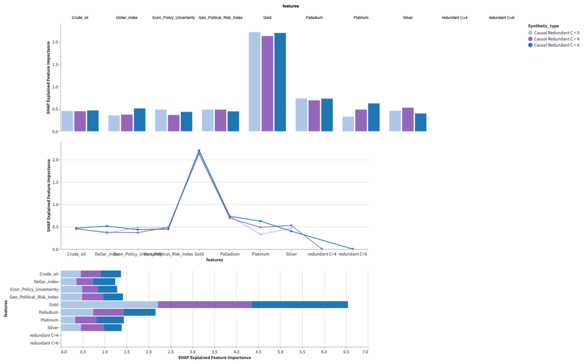

Figure 4: H2 Altair Plots of gold price original data

However, the SHAP interpreter are not said to be useless because of this result. By staking the mean shapley values of all selection of $n_{redundant}$ synthetic data set, the accumulative feature importance bar plot still explain the same portion of feature importance as the one generated singly by original data (the light blue bars).

4.1.3 H3 - gold

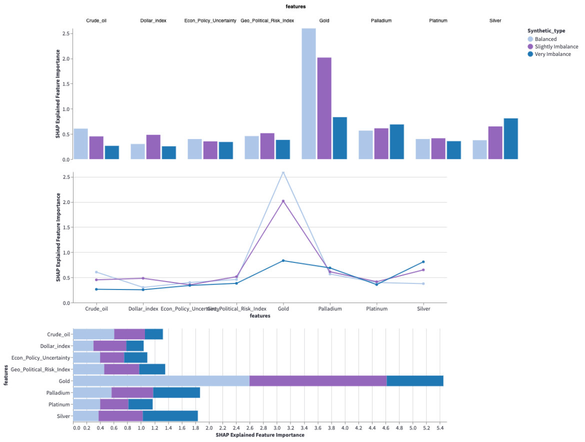

Figure 5: H3 Altair Plots of gold price original data

Then let observe Figure 5, in which we can observe that the class imbalance will change the prediction and SHAP feature importance attribution significantly. See the line plot, we can tell when the importance attribution of feature “Gold” significantly decreases as the classes in the train data getting more and more imbalance. This observed change actually reveals SHAP interpreter is sensitive to the balance of train data and may lead the user to wrongly address the most deterministic features in model prediction. Meawhile, unlike previous two hypothesis, the stacked bar plot of mean shapley values is also not stable when the class balance change.

4.2 Case study: Original consensus income

Regarding the proper function of the interactive tool with gold price data, we can do a real world analysis to interpret black box XGBoost classifier in predicting the individual’s income will be under 20k (class 0), or over and equal to 20k (class 1).

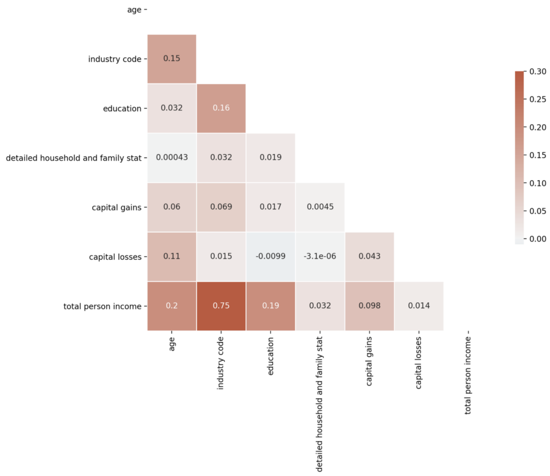

Figure 6: Correlation Matrix of census income original data

4.2.1 H1 - census

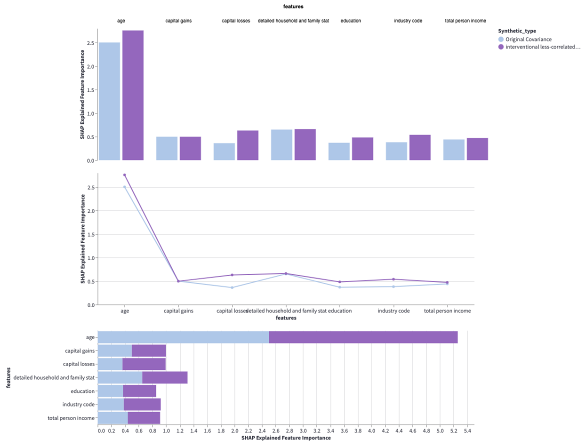

Figure 7: H1 Altair Plots of census income original data

From Figure 6, we can see that the correlation between features are not high as the one of gold price test data, but still, based on the color depth, we can see that feature “total person income” is relative strongly correlated with features “industry code”(0.75) and “age”(0.2). Hence, we can set $X_{base} = age$, and $X_1,\ X_2$= industry code, age in the interactive tool to generate synthetic data by changing original covariance matrix.

Observing Figure 7 line plot, the SHAP explained feature importance of all features don’t change much when the covariance matrix of original data change. It could be the result of initial independent relationship between feature in original train data. Hence, we may conclude that the interpretation by shapley values can be quite stable when the input features are less correlated with each other. Also, the independent original correlation coefficients might be resulted from the large size of input tabular data of census income.

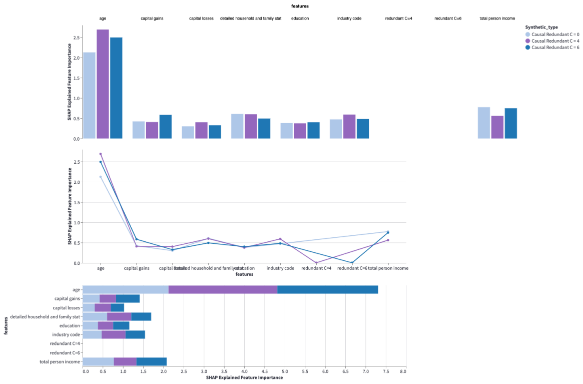

Figure 8: H2 Altair Plots of census income original data

4.2.2 H2 - census

Observing Figure 8, we can observe similar change in shapley values and the feature importance attribution. After fully redundant features are added to the input train data, the SHAP explained feature attributions are still in the same trend. In other words, the SHAP interpreter is stable in this case. Hence, probably, we may say, when the features in data used to train the black box model are less depend of each other, the SHAP can help interpret models more accurately. And the stacked bar plot exhibits consistent feature importance attribution.

4.2.3 H3 - census

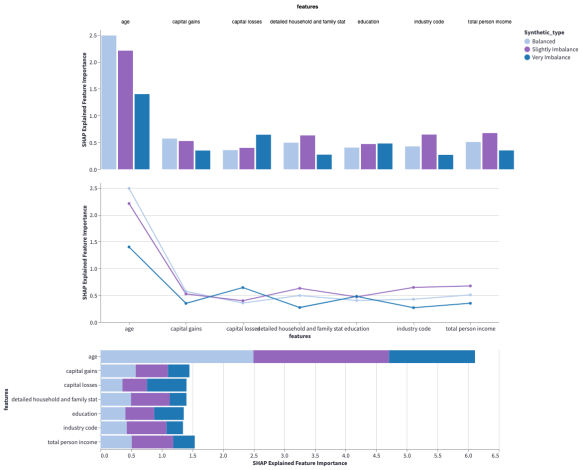

Figure 9: H3 Altair Plots of census income original data

Observing Figure 9, the imbalance in train data actually influences the shapley value result greatly. For instance, the shapley value of feature "household and family stat", "industry code", and "total person income" decreases significantly as the imbalance increases. In this case, even though the input data features are not high dependent of each other, the model explanation give by SHAP interpreter is still unstable. Yet, see the stacked plot, we can still obtain some confidence because it illustrates the accumulative importance of each feature, and this description is consistent with the original data set.

5 Discussion

The interactive tool in this project can be used to help users find accumulative SHAP feature importance bar plot, which may give more stable feature attribution explanation that help detect and reduce bias in black box model. But this tool include preset features of $n_{redundant}$ and class balance ratio, and only world with classification models. Hence, a future work to be done could be increase the operation freedom for users.

6 Conclusion

Generally, for both cases, the main importance rank of features in all 3 hypotheses are the same, only individual features importance attribution changes. As a result, if the user who use SHAP to explain model only need to know the ranking, SHAP is effective and stable. But if they want to do minor debug work, then they might try to find the accumulative SHAP feature importance results rather than the unstable original one.

References

[1] A. Babuta, M. Oswald, and C. Rinik. Machine learning algorithms and police decision-making: legal, ethical and regulatory challenges. 2018.

[2] B. Barr, K. Xu, C. Silva, E. Bertini, R. Reilly, C. B. Bruss, and J. D. Wittenbach. Towards Ground Truth Explainability on Tabular Data. In *2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020)*, pp. 362–367, 2020.

[3] R. Chainani et al. Factors influencing gold prices. *Voice Res*, 5:42–45, 2016.

[4] S. B. Jabeur, S. Mefteh-Wali, and J.-L. Viviani. Forecasting gold price with the xgboost algorithm and shap interaction values. *Annals of Operations Research*, pp. 1–21, 2021.

[5] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pp. 5491–5500. PMLR, 2020.

[6] T. Lane and R. Kohavi. Uci machine learning repository census-income (kdd) data set. 2000.

[7] J. Larson and J. Angwin. Machine bias. *Voice Res*, 2016.

[8] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[9] C. Molnar. *Interpretable machine learning*. Lulu.com, 2020.