Understanding the Perils of Black Box Explanations

Anurag Mishra
Computer Science, NYU Tandon School of Engineering
aa9279@nyu.edu

Advised by: Professor Claudio Silva

ABSTRACT

As machine learning black boxes become more widely used in fields like healthcare and criminal justice, more focus is being placed on developing tools and approaches for explaining these black boxes in a comprehensible manner. Domain experts use these explanations to diagnose systemic errors and underlying biases in black boxes. I show in this study that post hoc explanation approaches like LIME and SHAP, which rely on input perturbations, are unreliable. I built an app on streamlit to present the results and show that not only synthetic but real world data just by defining a function can how effectively hide any given classifier's biases by allowing an adversarial entity to build any arbitrary desired explanation. My method may be used to scaffold any biased classifier so that its predictions on the input data distribution remain skewed but the scaffolded classifier's post hoc explanations appear harmless. The Project has both a normal explorations of explanation libraries like lime and shap and then pointing out the peril of these methods by the exploration under synthetic dataset I show how the framework's severely biased (racist) classifiers can easily trick popular explanation approaches like LIME and SHAP into producing harmless explanations that do not reflect the underlying biases, using extensive testing real-world datasets like Census KDD. The results were really intriguing the we could clearly see that lime and shap neither performs well with imbalanced dataset but also correlated arrtibutes in the dataset. Also Lime completely fails to give detect a unrelated column to the attribute under the influence of biased classifier where as Shap tries to spread out the result because of its local accuracy in existence [2]

KEYWORDS

Data Exploration, Synthetic Data,Correlation,Machine Learning,Streamlit, Lime,Shapely Values, Post Hoc Explanations Methods,CatbootClasifier

Capstone Final Project, Spring 2022, New York, USA
© 2022

Reference Format:
Anurag Mishra. 2022. Understanding the Perils of Black Box Explanations. In *NYUAD Capstone Final Project Reports, Spring 2022, New York, USA.* 7 pages.

1 GITHUB REPOSITORY

This Github Repository [1] Consists of all the code required to test all results mentioned in the paper. Note the data was not uploaded because of the size and thus please use the reference of dataset [2] to download it.

2 INTRODUCTION

Because of the success of machine learning (ML) models, there is a growing interest in using them to assist decision-makers in vital domains including healthcare and criminal justice. The ability of decision makers to grasp and trust the operation of these models is critical to their acceptance in domain-specific applications. Decision makers can only discover flaws and potential biases in these models and decide when and how much to depend on them if they have a comprehensive grasp of their behavior. The proprietary nature and rising complexity of machine learning models, on the other hand, make it difficult for domain experts to comprehend these complicated black boxes, necessitating the development of tools that can faithfully describe them.

As a result, post-hoc strategies for explaining black box models in a human-interpretable manner have recently exploded. One of the most common applications for such explanations is to assist domain experts in detecting discriminating biases in black box models. Local, model-agnostic methods that focus on explaining individual predictions of a given black box classifier, such as LIME and SHAP, are the most prominent of these techniques. These methods calculate the contribution of individual features to a specific prediction by perturbing a specific instance in the data and analyzing the effect of the perturbations on the black-box classifier's output. These methods have been used to explain a variety of classifiers, including neural networks and sophisticated ensemble models, as well as in a variety of fields, including law, medicine, finance, and science However, little research has been done on the reliability and robustness of these explanation strategies, particularly in the adversarial scenario, leaving their utility for important applications in doubt. In this paper, I show how an attacker can exploit fundamental flaws in post-hoc explanation techniques to construct classifiers whose post-hoc explanations can be manipulated arbitrarily. More particular, I devise a unique approach for effectively masking any black box classifier's discriminatory biases. Our method takes advantage of the fact that post hoc explanation techniques like LIME and SHAP are perturbation-based.

This paper will describe the following using visuals from the web tool created using Streamlit:

1. Introduce the data to be explored using some visualisation to understand it properly
2. Post that we will apply Catboost Classfier on our ready to train dataset
3. Provide the necessary background information on the problem.
4. Explore the results on the test dataset and use Lime and Shap to see regular predictions [7]
5. Introduce Synthetics Experiment and analyse results
6. Apply the experiment on real world dataset i.e. Census KDD [2]
7. Explain what we infer from it and whether these post hoc explanation methods are really realiable or not.

3 RELATED WORK

A lot has been explored and already been found in the field of both supervised and unsupervised Machine learning and progressing through the semester and through out the course I really gained amazing insights about explanations for these machine learning models and why is it so important. The most important of all the reasons I chose this topic as it is so sensitive and it was so well explained by Professor Claudio Silva and also by one of the guest Professor Carlos Scheidegger and his paper was really insightfull [3]

While doing my research I read quite a few blogs and papers for refreshing the concepts and few of them gave me insights to shape up this project.

Understanding about correlations was very important as that was one of the most important part of experiment for this paper and this was really well explained in this blog [10]

Understading the Lime [9] and Shap [4] would not have been possible without these original paper and a simple explanation by this blog [7]

Without using Streamlit documentation this tool was not possible [11]

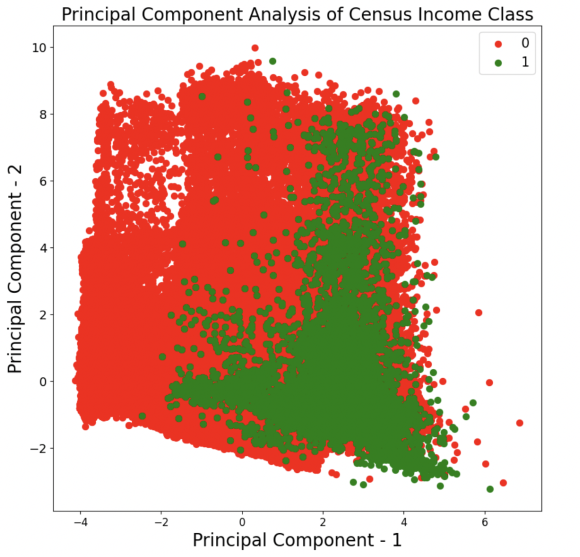


Figure 1: Shows PCA components and variance in data and also how imbalanced these class labels are in the dataset [8]

4 METHODOLOGY

4.1 Streamlit App Description

To analyse and visualise the project I created a tool using streamlit [11] it is a multi page web app with sections like Home,Data, Model Predictions,Lime, Shap and synthetic tab. Each page has its own functions. For a better picture See Figure to explore how these interactive visualisation look like.

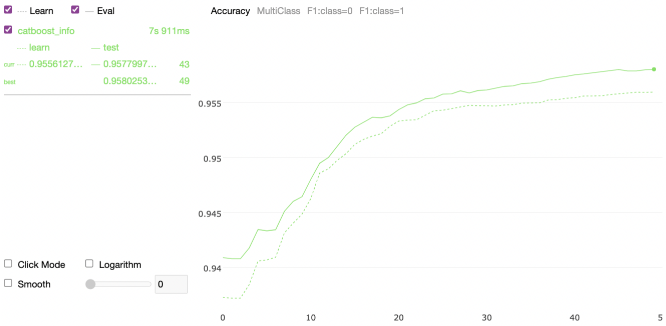
- Home Page is just a description of Project and the Github link where the repository for this project is.
- The Data tab helps to understand the data distribution i.e. whether it is imbalanced, need scaling,and is one hot encoding or label encoding required? It also shows visualisations to see a better picture about the data
- The Model Prediction tab just prints the prediction of CatBoost Model.
- Explore on the Lime tab what predictions it gives and also see the fooled explanations.
- Explore on the Shap tab what predictions it gives and also see the fooled explanations.

4.2 Data Exploration

The data provided by the Professor was Census KDD [2] and it required a lot of features engineering and cleaning.

Firstly the .data file and test data both consist of 40 features to which I added two uncorrelated and correlated columns.

The classification label were highly imbalanced which had to be balanced before training, also there were a lot if



☑ ---- Learn ☑ — Eval

☑ catboost_info 7s 911ms

	---- learn	— test	
curr	0.9556127...	0.9577997...	43
best		0.9580253...	49

□ Click Mode □ Logarithm

□ Smooth 0

Figure 2: Model Predictions Using CatBoostClassifier

categorical features so to better train with RandomForest it needed to be scaled but with predictions being almost the same I chose to use CatBoost Model [8] which works very well with the categorical features.

To visualise the data I used PCA and it can be clearly seen that there is imbalance in the income class where label 0 is class -50000 and label 1 is +50000. (see Figure 1)

4.3 Model Predictions

Post perfecting the dataset - cleaning and applying feature engineering - It became obvious to use CatBoostClassfier Model for training [8] The predictions made by the model on the dataset were amazing. The results can be seen in the Figure 4. The model worked with 95-97 percent accuracy on both the train and testset.

4.4 Hypothesis

After all the trainig testing and predicitions of the model it was time to explore post hoc explanations but before we had to hypothesize few things which could help us understand the perils and with all the research done I had created these hypothesis listed below:

1. **Lime and shap fails to sense imbalance in data and give random explanations when fed through same imbalanced and balanced classed in dataset**
2. **Lime and Shap doesnot perform well when there is correlation in the dataset**
3. **These Post hoc explanations method are not reliable under biased conditions**
4. **Shap performs better when there is correlated attributes present in the dataset than Lime**

4.4.1 ***Experiments on Synthetic Dataset.*** To perform experiments to accept or reject these hypothesis I started by generating a sensitive synthetic dataset (loan acceptance or rejection) which has correlated attributes.Just to give an idea Figure 3 shows the correlation matrix. Also a function that performs in a way described in the equation below where is e(x) is our adversarial classifier, f(x) is the biased classifier and g(x) is our unbiased classifier (e.g., makes predictions based on innocuous features that are correlated based on situations and uncorrelated with sensitive attributes)

$$e\left(x\right) = \begin{cases} f(x)\ ;\ if\ race == \ 'african\ american' \\ g(x)\ ;\ \ otherwise \end{cases}$$

4.4.2 Results of Experiment on Synthetic Data. The results that we got from synthetic data were astonishing and helped me accept the null hypothesis that these methods [4, 9] not only fails when there is correlated columns but can also be easily influenced by biased classifiers. Please see Figure 4 for the results. The Lime and Shap explanations both got fooled by the biased classifier and gave results where they clearly predicted loan denied and the features responsible were race and sex.

5 RESULTS

5.1 Implementing these Hypothesis on Real World Dataset

Now we take our real worl dataset and introduce two new attritbutes 'NoRelationtodataColumn' and 'NoRelationtodataColumn2'.Former of these column is correlated withthe attribute weeks of work per year and the latter column is an uncorrelated column introduced and based on this the claassifier has been trained to choose class label to make it a biased attribute and fool these explanations.

5.1.1 Experimental Setup. Remember the 4 hypothesis I mentioned:

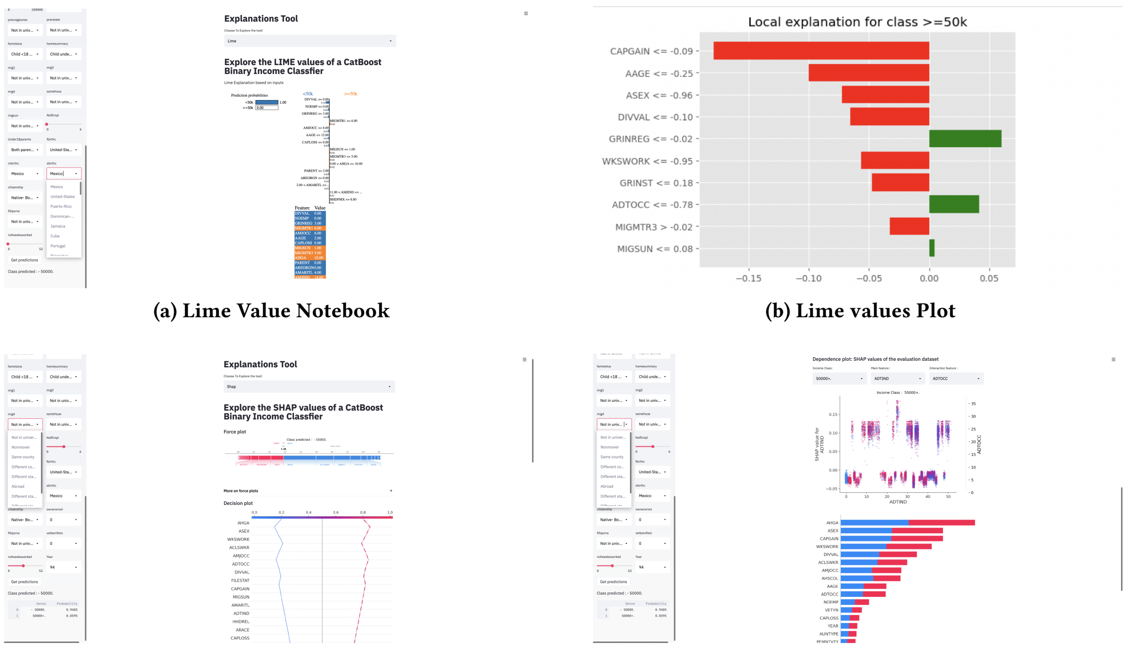
1. Fail to screen out the Imbalanced class label
2. Unreliable when there is Biased attribute/classifier
3. Fails when there is correlated attribute(s)
4. Shap explains better when there is correlated attribute

5.2 Final Explanations after Introducing Anomalies

Now Lets start by implementing these assumptions one by one of the data.

5.2.1 ***Imbalanced Class Results.*** This section we will discuss how these lime and shap explanations performs when implemented on both original dataset with imbalance in class label present and post processing and balancing these class label.

1. We See that these results were totally different and one can question the explanability of these methods. The lime values not only wrongly predicted unrelated



(a) Lime Value Notebook

(b) Lime values Plot

(c) Shapely Values Dashboard

(d) Shapely Values

Figure 3: Lime and Shap Explanations on Original Dataset

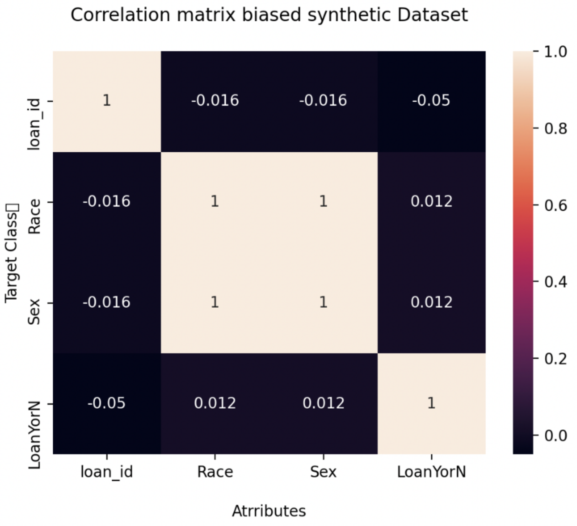


Figure 4: Correlation Matrix for Experiment Data

column the main feature but also gave random explanation and it concluded that it was unstable. The shap values retained its top feature but failed to understand correlation and gave different results for imbalanced and balanced case.

5.2.2 *Biased Classifier and Correlated Attributes.*

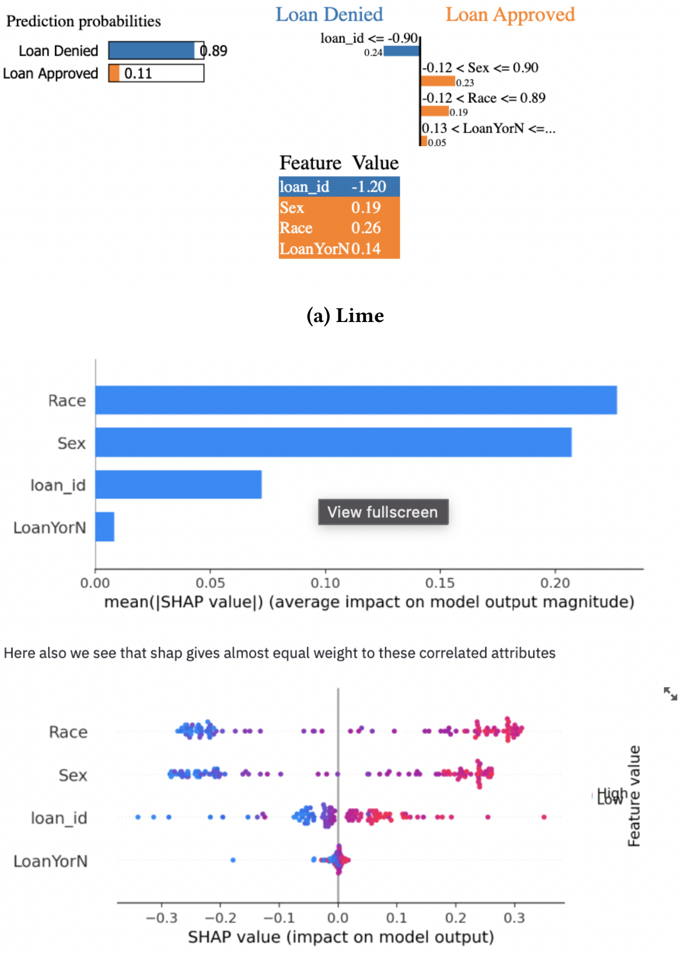
The Results for this case was amazing and lead me to conclude that these methods have their way to react to these anomalies.

The result for lime values were worse as it was completely fooled by the correlated attribute and gave maximum weight to it for the explanation

It was also fooled by the Biased classifier and the column which had nothing to do with the data could been seen in the attribute which contributed in the results of explanation.

In case of LIME, when a single feature is used for the attack i.e., g(x) uses a single feature for making predictions, the adversarial classifier e successfully shifts the feature importance in LIME from CAPGAIN to Uncorrelated Feature under the influence of classifier f. These results demonstrate that the LIME explanation technique has been effectively fooled by the adversarial classifier e for all three datasets.

For SHAP, when a single uncorrelated feature is used for the attack, the adversarial classifier e successfully shifts the feature importance from the sensitive feature. When one correlated and other uncorrelated features are used in the attack , the adversarial classifier is less successful in removing the top feature but still tops the original feature WKSWORK.This is due to SHAP's local accuracy property, which requires feature attributions to equal the difference between a particular prediction and the background distribution's average



(a) Lime

Here also we see that shap gives almost equal weight to these correlated attributes

(b) Shapely Values

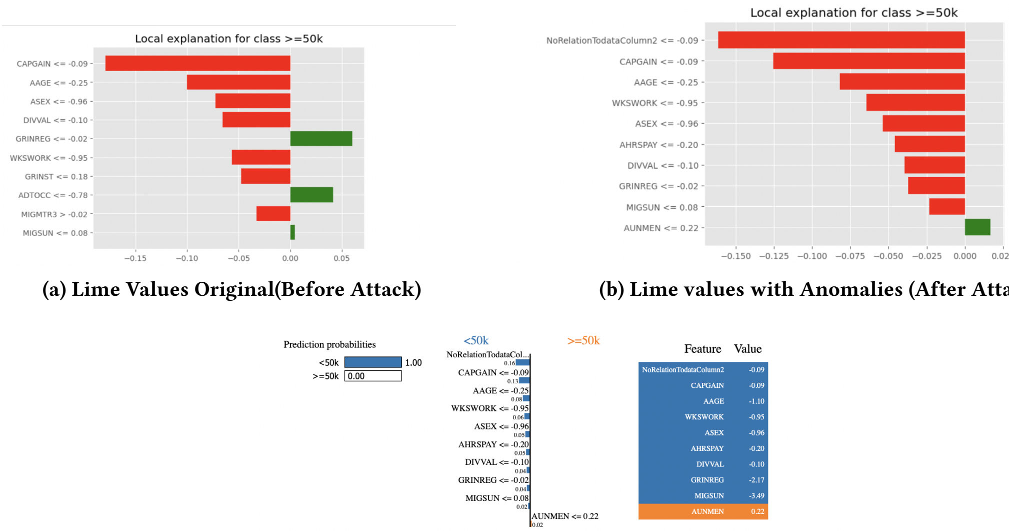
Figure 5: Experiment Results on Synthetic Data For Both Lime and Shap

prediction. When a single most informative feature cannot be identified, this property will disperse feature attributions among numerous features.

6 PERILS OF LIME AND SHAP

(1) **LIME [5]**
  (a) The correct definition of the neighborhood is a very big, unsolved problem when using LIME with tabular data. In my opinion it is the biggest problem with LIME and the reason why I would recommend to use LIME only with great care. For each application you have to try different kernel settings and see for yourself if the explanations make sense. Unfortunately, this is the best advice I can give to find good kernel widths.
  (b) Sampling could be improved in the current implementation of LIME. Data points are sampled from a Gaussian distribution, ignoring the correlation between features. This can lead to unlikely data points which can then be used to learn local explanation models.
  (c) The complexity of the explanation model has to be defined in advance. This is just a small complaint, because in the end the user always has to define the compromise between fidelity and sparsity.
  (d) Another really big problem is the instability of the explanations. Figure 6 shows that the explanations of imbalanced and balanced also with correlations and without were so unstable. Also, in my experience, if you repeat the sampling process, then the explanations that come out can be different. Instability means that it is difficult to trust the explanations, and you should be very critical.
  (e) LIME explanations can be manipulated by the data scientist to hide biases. The possibility of manipulation makes it more difficult to trust explanations generated with LIME. Local surrogate models, with LIME as a concrete implementation, are very promising. But the method is still in development phase and many problems need to be solved before it can be safely applied.

(2) **Shapely Values [6]** Athough it performs better because of its local accuracy property it still has shortcomings.
  (a) KernelSHAP is slow. This makes KernelSHAP impractical to use when you want to compute Shapley values for many instances. Also all global SHAP methods such as SHAP feature importance require computing Shapley values for a lot of instances.
  (b) KernelSHAP ignores feature dependence. Most other permutation based interpretation methods have this problem. By replacing feature values with values from random instances, it is usually easier to randomly sample from the marginal distribution. However, if features are dependent, e.g. correlated, this leads to putting too much weight on unlikely data points. TreeSHAP solves this problem by explicitly modeling the conditional expected prediction.
  (c) TreeSHAP can produce unintuitive feature attributions. While TreeSHAP solves the problem of extrapolating to unlikely data points, it does so by changing the value function and therefore slightly changes the game. TreeSHAP changes the value function by relying on the conditional expected prediction. With the change in the value function, features that have



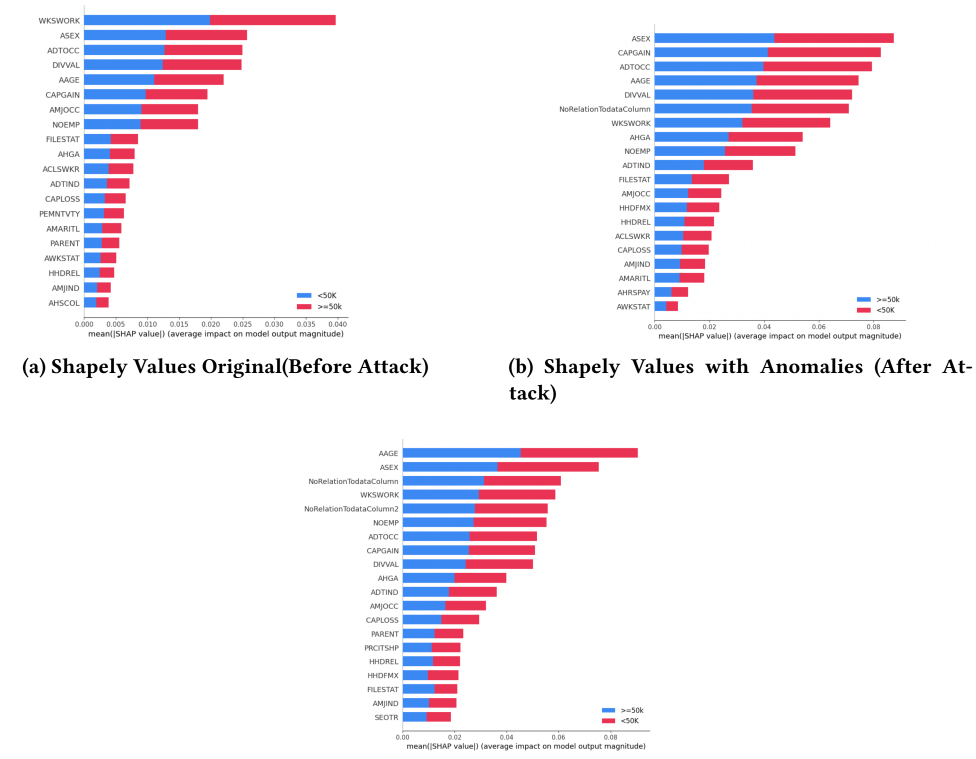
(a) Lime Values Original(Before Attack)

(b) Lime values with Anomalies (After Attack)

Feature	Value
NoRelationTodataColumn2	-0.09
CAPGAIN	-0.09
AAGE	-1.10
WKSWORK	-0.95
ASEX	-0.96
AHRSPAY	-0.20
DIVVAL	-0.10
GRINREG	-2.17
MIGSUN	-3.49
AUNMEN	0.22

(c) Lime values with Anomalies (After Attack) Notebook

Figure 6: Before and After Attack Lime Fooled Explanations



(a) Shapely Values Original(Before Attack)

(b) Shapely Values with Anomalies (After Attack)

(c) Shapely Values with Anomalies (After Attack 2)

Figure 7: Shap Fooled Explanations before and after

no influence on the prediction can get a TreeSHAP value different from zero.

(d) The disadvantages of Shapley values also apply to SHAP: Shapley values can be misinterpreted and access to data is needed to compute them for new data (except for TreeSHAP).

7 CONCLUSION

I suggested a novel framework for properly hiding any black box classifier's discriminatory biases. Our method takes advantage of the fact that post hoc explanation techniques like LIME and SHAP are perturbation-based to build a scaffolding around the biased classifier, so that its predictions on the input data distribution remain biased but its behavior on the perturbed data points is controlled, making the post hoc explanations appear completely harmless. Extensive testing with real-world data Census Data [2] domains shows that our method is effective at developing adversarial classifiers that can mislead post-hoc explanation procedures, with LIME proving to be more vulnerable than SHAP. As a result, my findings show that current post hoc explanation methodologies are insufficient for determining discriminating behavior of classifier in sensitive data environment. This work sets the way for a number of promising future research avenues in machine learning explainability. To begin, it would be interesting to investigate whether other types of post hoc explanation techniques (for example, gradient-based approaches) are equally subject to adversarial attacks. Second, developing new approaches for generating adversarially resilient explanations that can withstand attacks like those described in this paper would be intriguing. Also exploring various kinds of data like image and understanding how these methods wrongly interpret image data.

8 ACKNOWLEDGEMENTS

I would like to express my gratitude to my Primary Advisor,Prof Claudio Silva, who guided me throughout this project, gave inspirations and lessons on how to visualise machine learning and also gave motivation at every step of the process. I wish to acknowledge the help provided by the TA's of the Course - Visualisation for Machine Learning. I would also like to show my deep appreciation for specifically one of the teaching assistant Peter Xenopoulous who helped me finalize my project and guided me throughout the process and helped me with many obstacles during the process.

REFERENCES

[1] AnuragMishra1712. [n.d.]. ANURAGMISHRA1712/visualisation-for-machine-learning. https://github.com/AnuragMishra1712/Visualisation-For-Machine-Learning

[2] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[3] Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. 2021. Shapley Residuals: Quantifying the limits of the Shapley value for explanations. *Advances in Neural Information Processing Systems* 34 (2021).

[4] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[5] Christoph Molnar. 2022. Interpretable machine learning. https://christophm.github.io/interpretable-ml-book/lime.html

[6] Christoph Molnar. 2022. Interpretable machine learning. https://christophm.github.io/interpretable-ml-book/shap.html

[7] Ashutosh Nayak. 2019. Idea Behind LIME and SHAP. https://towardsdatascience.com/idea-behind-lime-and-shap-b603d35d34eb

[8] Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features.. In *NeurIPS*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 6639–6649. http://dblp.uni-trier.de/db/conf/nips/nips2018.html#ProkhorenkovaGV18

[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " 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*. 1135–1144.

[10] Aishwarya V Srinivasan. 2019. Why exclude highly correlated features when building regression model. https://towardsdatascience.com/why-exclude-highly-correlated-features-when-building-regression-model-34d77a90ea8e

[11] Adrien Treuille. 2018. Streamlit Documentation. https://docs.streamlit.io/