Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods

Jamal Ahmed Bhatti
Alireza Bayat Makou

Winter semester 2021/2022

Outline

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

4 PDP analysis

5 Best Classifier

6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

Introduction

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

4 PDP analysis

5 Best Classifier

6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods
Jamal Ahmed Bhatti
Alireza Bayat Makou
Introduction
Experimental Setup
LIME and SHAP results
PDP analysis
Best Classifier
Hyperparameter sensitivity of LIME and SHAPE parameters
References
References

Introduction

- Post-hoc explanations are needed for black box models

Introduction

Introduction

Experimental Setup

LIME and SHAP results

PDP analysis

Best Classifier

Hyperparameter sensitivity of LIME and SHAPE parameters

References

References

- Post-hoc explanations are needed for black box models
- Black box models are common because of the propriety rights or because of complex models

Introduction

- Post-hoc explanations are needed for black box models
- Black box models are common because of the propriety rights or because of complex models
- LIME(Local interpretable model-agnostic explanations) based on local surrogate model and SHAP (Shapley Additive Explanations) can be global interpretation methods are used for explanations for the black box models

Introduction

- Post-hoc explanations are needed for black box models
- Black box models are common because of the propriety rights or because of complex models
- LIME(Local interpretable model-agnostic explanations) based on local surrogate model and SHAP (Shapley Additive Explanations) can be global interpretation methods are used for explanations for the black box models

⇝ These techniques are not fool-proof

Introduction

Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods

Jamal Ahmed Bhatti
Alireza Bayat Makou

Introduction

Experimental Setup

LIME and SHAP results

PDP analysis

Best Classifier

Hyperparameter sensitivity of LIME and SHAPE parameters

References

References

- Post-hoc explanations are needed for black box models
- Black box models are common because of the propriety rights or because of complex models
- LIME(Local interpretable model-agnostic explanations) based on local surrogate model and SHAP (Shapley Additive Explanations) can be global interpretation methods are used for explanations for the black box models

⇝ These techniques are not fool-proof

  - Both rely on input perturbations

Introduction

- Post-hoc explanations are needed for black box models
- Black box models are common because of the propriety rights or because of complex models
- LIME(Local interpretable model-agnostic explanations) based on local surrogate model and SHAP (Shapley Additive Explanations) can be global interpretation methods are used for explanations for the black box models

⇝ These techniques are not fool-proof
  - Both rely on input perturbations
  - By guessing the input perturbations a classifier can hide the model biases and send the perturb input to the innocuous model [6]

Applications

Applications

- One can not act without trusting the model if the application is like a medical diagnosis or terrorism detection

Applications

Introduction

Experimental Setup

LIME and SHAP results

PDP analysis

Best Classifier

Hyperparameter sensitivity of LIME and SHAPE parameters

References

References

- One can not act without trusting the model if the application is like a medical diagnosis or terrorism detection
- Two major problems:

Applications

- One can not act without trusting the model if the application is like a medical diagnosis or terrorism detection
- Two major problems:
    - Trusting a model

Applications

- One can not act without trusting the model if the application is like a medical diagnosis or terrorism detection
- Two major problems:
  - Trusting a model
  - Trusting an individual prediction

Claims of the author

- "A novel scaffolding technique that effectively hides the biases of any given classifier by allowing an adversarial entity to craft an arbitrary desired explanation"

Problem Statement I

- Let $\mathcal{D}$ denote the input *dataset* for N points$(x_1, y_1) \dots (x_N, y_N)$. where x_i(vector) denotes the feature value at the dataset point i and y_i is the corresponding class label

Problem Statement I

- Let $\mathcal{D}$ denote the input *dataset* for N points$(x_1, y_1) \ldots (x_N, y_N)$. where x_i(vector) denotes the feature value at the dataset point i and y_i is the corresponding class label
- Let there M features in dataset $\mathcal{D}$ and let $y_i \in \mathcal{C}$ denote the *class label*

Problem Statement I

- Let $\mathcal{D}$ denote the input *dataset* for N points$(x_1, y_1) \ldots (x_N, y_N)$. where x_i(vector) denotes the feature value at the dataset point i and y_i is the corresponding class label
- Let there M features in dataset $\mathcal{D}$ and let $y_i \in \mathcal{C}$ denote the *class label*
- Let f denote the *black box classifier* such that $f(x_i) \in \mathcal{C}$

Problem Statement I

- Let $\mathcal{D}$ denote the input *dataset* for N points$(x_1, y_1) \dots (x_N, y_N)$. where x_i(vector) denotes the feature value at the dataset point i and y_i is the corresponding class label
- Let there M features in dataset $\mathcal{D}$ and let $y_i \in \mathcal{C}$ denote the *class label*
- Let f denote the *black box classifier* such that $f(x_i) \in \mathcal{C}$
- Let g denote the *explanation model* that is intended to explain $f, g \in G$ where G is the *class of linear models*

Problem Statement I

- Let $\mathcal{D}$ denote the input *dataset* for N points$(x_1, y_1) \dots (x_N, y_N)$. where x_i(vector) denotes the feature value at the dataset point i and y_i is the corresponding class label
- Let there M features in dataset $\mathcal{D}$ and let $y_i \in \mathcal{C}$ denote the *class label*
- Let f denote the *black box classifier* such that $f(x_i) \in \mathcal{C}$
- Let g denote the *explanation model* that is intended to explain $f, g \in G$ where G is the *class of linear models*
- Let the *complexity* of the explanation g denoted as $\Omega(g)$; $\Omega(g)$ will penalize the objective function (regularization term)

Problem Statement I

- Let $\mathcal{D}$ denote the input *dataset* for N points$(x_1, y_1)\ldots(x_N, y_N)$. where x_i(vector) denotes the feature value at the dataset point i and y_i is the corresponding class label
- Let there M features in dataset $\mathcal{D}$ and let $y_i \in \mathcal{C}$ denote the *class label*
- Let f denote the *black box classifier* such that $f(x_i) \in \mathcal{C}$
- Let g denote the *explanation model* that is intended to explain $f, g \in G$ where G is the *class of linear models*
- Let the *complexity* of the explanation g denoted as $\Omega(g)$; $\Omega(g)$ will penalize the objective function (regularization term)
- Let $\pi_x(x')$ denote the *proximity measure/local kernel* between inputs x and x'

Problem Statement II

$$\underset{g \in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$$\mathcal{L}(f, g, \pi_x) = \sum_{x' \in X'} [f(x') - g(x')]^2 \pi_x(x')$$

Problem Statement II

$$\arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$$\mathcal{L}(f, g, \pi_x) = \sum_{x' \in X'} [f(x') - g(x')]^2 \pi_x(x')$$

- where $\mathcal{L}$ is the loss function, and X' is the set of inputs constituting the neighborhood of x

Problem Statement II

$$\arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$$\mathcal{L}(f, g, \pi_x) = \sum_{x' \in X'} [f(x') - g(x')]^2 \pi_x(x')$$

- where $\mathcal{L}$ is the loss function, and X' is the set of inputs constituting the neighborhood of x
- The objective function approximates f in the vicinity of x

Major differences between LIME and SHAP

Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods

Jamal Ahmed Bhatti
Alireza Bayat Makou

Introduction

Experimental Setup

LIME and SHAP results

PDP analysis

Best Classifier

Hyperparameter sensitivity of LIME and SHAPE parameters

References

References

The difference lies in the choice of $\Omega(g)$ and $\pi_x(x')$

- $\Omega(g)$ is the number of non-zero weights in the linear model
- $\pi_x(x')$ is defined using L_2 or cosine similarity

- (Kernel) SHAP grounds on game theoretic principles so that it satisfies certain properties

Main ideas

- Adversary wants to fool the post hoc explanation techniues and hide the underlying biases of f
- Adversary provides only f
- The f is trained X_{dist} which neither adversary nor the framework has access
- The X_{dist} is biased but LIME and SHAP will not reveal
- The idea is to train OOD(Out-of-distribution) classifier which can identify if the instance is asked from the perturbed space or from the dataset

Methodology

- Let's asssume ψ is an unbiased classifier and e is the adversarial classifier

$$e(x) = \begin{cases} f(x), & \text{if } x \in X_{dist} \\ \psi(x), & \text{otherwise} \end{cases}$$

- Two kinds of datasets are created X and X_p where X_p is the perturbed dataset with labels if OOD is true or not
- Classifier is trained on $X \bigcup X_p$ with their labels

Datasets

- COMPAS [3]
- Communities and Crime (CC) [5]
- German credit [4]
- Boston Housing [1]
- Student Prediction [2]

Experimental Setup

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

4 PDP analysis

5 Best Classifier

6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

Experimental Setup I

- Random split (90-10)
- Biased Classifier f :
  - Sensitive Feature
  - For instance for COMPAS decision purely based on race
- OOD-Classifier
  - For LIME X was added to $\mathcal{N}(0, 1)$
  - For SHAP, a random subset of features for $x_i \in X$ and replace them by the background distribution
    - Background distribution for SHAP: cluster centers from kmeans with 10 clusters.
  - Classifier: Random forest with 100 tree depth

Experimental Setup II

- Unbiased Classifier ψ:
  - Based on uncorrelated feature with sensitive feature
    - Two Co-related feature: Based on XOR for two features
- Generating Explanation:
  - LIME tabular implementation (default)
  - Kernel SHAP implementation with *kmeans* with 10 clusters as the background distribution (default)

LIME and SHAP results

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

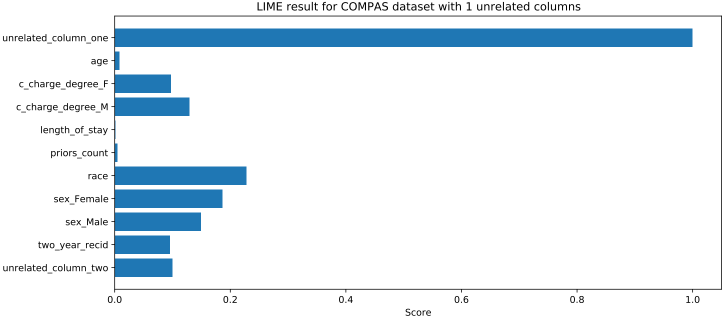
4 PDP analysis

5 Best Classifier

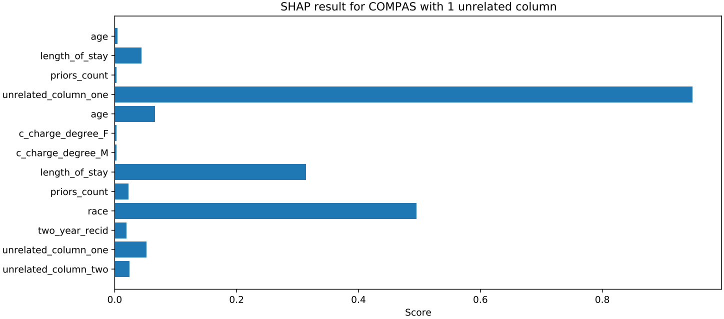
6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

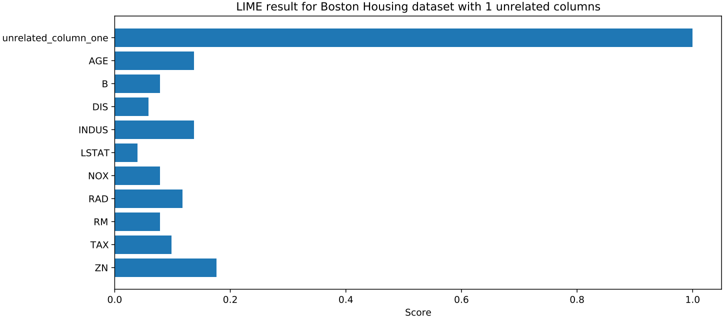
LIME results for COMPAS



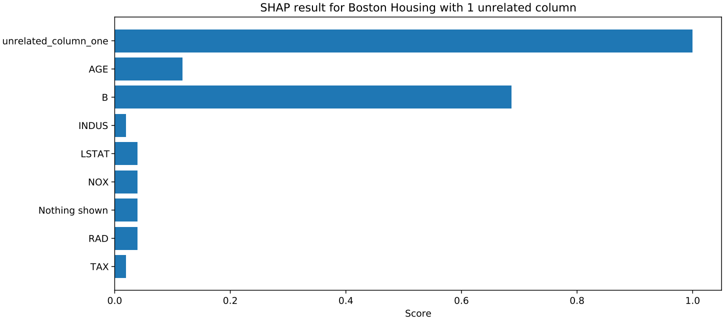
SHAP results for COMPAS



LIME results for Boston Housing



SHAP results for Boston Housing



PDP analysis

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

4 PDP analysis

5 Best Classifier

6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

PDP results

- LIME model Boston Housing

PDP results

- LIME model Boston Housing
- SHAP model Boston Housing

Introduction

Experimental Setup

LIME and SHAP results

PDP analysis

Best Classifier

Hyperparameter sensitivity of LIME and SHAPE parameters

References

References

PDP results

- LIME model Boston Housing
- SHAP model Boston Housing
- LIME model COMPAS

PDP results

- LIME model Boston Housing
- SHAP model Boston Housing
- LIME model COMPAS
- SHAP model COMPAS

Introduction
Experimental Setup
LIME and SHAP results
PDP analysis
Best Classifier
Hyperparameter sensitivity of LIME and SHAPE parameters
References
References

Best Classifier

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

4 PDP analysis

5 Best Classifier

6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

Best Classifier I

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:

Best Classifier I

Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods

Jamal Ahmed Bhatti
Alireza Bayat Makou

Introduction

Experimental Setup

LIME and SHAP results

PDP analysis

Best Classifier

Hyperparameter sensitivity of LIME and SHAPE parameters

References

References

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:
  - Classifiers: LogisticRegression, SVC, KNeighborsClassifier, GaussianNB, MultinomialNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:
  - Classifiers: LogisticRegression, SVC, KNeighborsClassifier, GaussianNB, MultinomialNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier
  - Metric: F1-Score (same as paper)

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:
  - Classifiers: LogisticRegression, SVC, KNeighborsClassifier, GaussianNB, MultinomialNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier
  - Metric: F1-Score (same as paper)
  - Total hyperparameters: 125

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:
  - Classifiers: LogisticRegression, SVC, KNeighborsClassifier, GaussianNB, MultinomialNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier
  - Metric: F1-Score (same as paper)
  - Total hyperparameters: 125
  - GridSearch

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:
  - Classifiers: LogisticRegression, SVC, KNeighborsClassifier, GaussianNB, MultinomialNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier
  - Metric: F1-Score (same as paper)
  - Total hyperparameters: 125
  - GridSearch
  - With available computation power only could do for LIME for SHAP it would take months!

Best Classifier I

Introduction

Experimental Setup

LIME and SHAP results

PDP analysis

Best Classifier

Hyperparameter sensitivity of LIME and SHAPE parameters

References

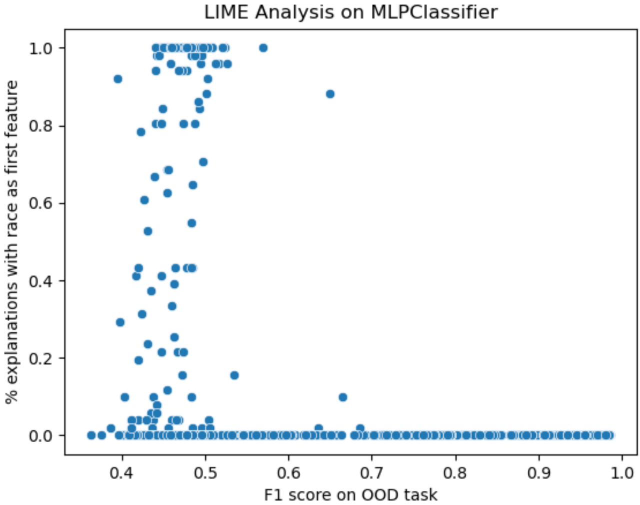
References

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:
  - Classifiers: LogisticRegression, SVC, KNeighborsClassifier, GaussianNB, MultinomialNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier
  - Metric: F1-Score (same as paper)
  - Total hyperparameters: 125
  - GridSearch
  - With available computation power only could do for LIME for SHAP it would take months!
- Best Classifier:

Best Classifier I

- Misclassification of OOD leads to correct explanation for both LIME and SHAP
  - Certain tweaking of the hyperparameters for the classifier is needed
  - The paper suggested the Random Forest with tree depth 100
- Search space we considered:
  - Classifiers: LogisticRegression, SVC, KNeighborsClassifier, GaussianNB, MultinomialNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier
  - Metric: F1-Score (same as paper)
  - Total hyperparameters: 125
  - GridSearch
  - With available computation power only could do for LIME for SHAP it would take months!
- Best Classifier:
  - DecisionTreeClassifier($max_depth = 10, random_state =$ 123454321) (not same as the paper)

Best Classifier II



Best Classifier III

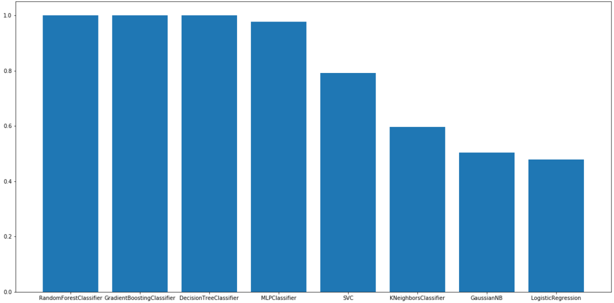


Figure: Best Classifer results, y-axis is the $F-1$ score

Hyperparameter sensitivity of LIME and SHAPE parameters

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

4 PDP analysis

5 Best Classifier

6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

References

1 Introduction

2 Experimental Setup

3 LIME and SHAP results

4 PDP analysis

5 Best Classifier

6 Hyperparameter sensitivity of LIME and SHAPE parameters

7 References

References I

- Ali Al Bataineh and Devinder Kaur. "A comparative study of different curve fitting algorithms in artificial neural network using housing dataset". In: *NAECON 2018-IEEE National Aerospace and Electronics Conference*. IEEE. 2018, pp. 174–178.
- Huda Al-Shehri et al. "Student performance prediction using support vector machine and k-nearest neighbor". In: *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*. IEEE. 2017, pp. 1–4.
- Julia Angwin et al. "Machine bias". In: *ProPublica, May 23.2016* (2016), pp. 139–159.
- C Blake, E Koegh, and CJ Mertz. "Repository of Machine Learning". In: *University of California at Irvine* (1999), p. 75.

References II

Michael Redmond and Alok Baveja. "A data-driven software tool for enabling cooperative information sharing among police departments". In: *European Journal of Operational Research* 141.3 (2002), pp. 660–678.

Dylan Slack et al. "Fooling lime and shap: Adversarial attacks on post hoc explanation methods". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 180–186.