LIME ein vielseitiges Erklärermodell - auch für Machine-Learning-Laien

Simon Hoffmann

Seminar KI: gestern, heute, morgen
Computing in the Humanities, Universität Bamberg

Zusammenfassung. Das Erklärermodell LIME *(Local Interpretable Modelagnostic Explanations)* hat an sich selbst den holistischen Anspruch gesetzt, jeden maschinellen Klassifikator und dessen Prognose erklären zu können. Die Erklärung durch LIME soll dabei für eine erweiterte Gruppe von fachfremden Endanwender*innen verständlich sein und die Anpassung bzw. Verbesserung eines Modells in einem untersuchungsrelevanten, semantischen Kontext erleichtern. In dieser Arbeit wird im Rahmen von Experimenten und eines praktischen Selbstversuchs geprüft, ob sich LIME als Erklärermodell für Laien eignet und somit das Vertrauen in maschinelle Prognosen bestärkt werden kann. Zusammengefasst zeigen die Experimente, dass LIME bei der Erklärung von maschinellen Prognosen primär im Bereich der Text-Klassifikation nützlich sein kann. Die Verbesserung von Modellen setzt jedoch explizites Vorwissen zu maschinellem Lernen und Python voraus. Die Modellinterpretation aus LIME lässt sich von Laien durchführen, die Implementierung des Erklärermodells ist mit der Anleitung eines zugehörigen GITHUB-Projektes theoretisch simpel, konnte in einem Selbstversuch durch einen Paketfehler in LIME jedoch abschließend nicht nachvollzogen werden. Ob LIME letztendlich mehr Vertrauen in maschinelle Prognosen generieren kann, lässt sich an dieser Stelle nur durch die bessere Erklärung erahnen, jedoch nicht beweisen, dies impliziert weiteren Forschungsbedarf.

Schlüsselwörter: Maschinelles Lernen, Erklärer

1 Motivation

Machine Learning oder zu Deutsch maschinelles Lernen dringt in immer mehr Lebens- und Arbeitsbereiche vor und ist inzwischen ein essentieller Helfer bei Prognosen und Vorhersagen. Wie die Prognosen zustande kommen, ist gerade bei komplexeren Anwendungsfällen nicht mehr transparent, es wird in diesem Fall von Black-Box-Systemen gesprochen, bei denen der Mensch die Ergebnisse nur mit sehr viel Aufwand nachvollziehen kann. Bei sicherheitsrelevanten Themen oder Prognosen bei denen es beispielweise um Menschenleben geht, können falsche Entscheidungen die durch ihren intransparenten Entscheidungsbaum nicht erkannt wurden, weitreichende Folgen haben. (Ribeiro, Singh, & Guestrin, 2016, S.1)

Man Stelle sich folgendes Szenario vor: Ein Model für die Bildklassifizierung wurde korrekt aufgesetzt und ein neuronales Netz versucht nun erkrankte Menschen zu klassifizieren. Maschinen generieren anhand der vorliegenden Daten Wissen über die kranken Menschen. Nun besteht jedoch der Fall, obwohl die Funktionsweise des Klassifikators bekannt ist, dass durch suboptimale Trainingsdaten letztendlich falsche Schlussfolgerungen über einen zu klassifizierenden Menschen getroffen werden. So wird er beispielsweise als gesund eingestuft, weil der Hintergrund des Bildes mit dem Hintergrund von gesunden Menschen übereinstimmt. Eine transparente Entscheidungsfindung ist für derartige Einsatzbereiche unerlässlich, um wie in diesem Falle zu erkennen, dass der Hintergrund nicht ein entscheidendes Kriterium sein sollte.

Transparente Entscheidungen sind somit entscheidend für die Akzeptanz und Vertrauen in von Maschinen getroffene Entscheidungen. Das Ziel, mehr Vertrauen in maschinelle Prognosen zu schaffen hat sich das Erklärer-System LIME gesetzt. Es hat den Anspruch Entscheidungsfindungen von Black-Back-Systemen transparent darzustellen, so dass auch nicht Machine-Learning-Expert*innen Prognosen nachvollziehen können und so der Einsatz von derartigen Hilfsprogrammen mehr Akzeptanz erfährt. Draus lässt sich auch die in dieser Arbeit behandelte Forschungsfrage ableiten: Eignet sich LIME als Erklärungsmodell - auch für Machine-Learning-Laien - um die Entscheidungsfindung von Machine-Learning-Algorithmen besser zu verstehen und somit das Vertrauen in deren Prognosen zu erhöhen.

2 Hintergrund

Das Erklärer-Modell LIME *(Local Interpretable Model-agnostic Explanations)* ist nur ein Ansatz im Forschungsgebiet der *Ëxplainarity"*, das wiederum eng verbunden ist mit dem Bereich des maschinellen Lernens. Um auf Eigenschaften von LIME und anderen Erklärer-Ansätzen eingehen zu können müssen zuerst einige Hintergrundinformationen vermittelt werden: (Ribeiro et al., 2016, S.1)

2.1 Erklärungen

Machine-Learning: ist ein Teilgebiet der künstlichen Intelligenz und beschäftigt sich „[...], mit der computergestützten Modellierung und Realisierung von Lernphänomenen [...].“ (Görz, Schmid, & Wachsmuth, 2013, p. 405). Dabei geht es primär um aus bestehenden Datenbeständen mittels überwachten oder unüberwachten Lernen, Wissen und Regeln abzuleiten, entdecken bzw. zu lernen. (Görz et al., 2013, S. 405)

Instanz: Ist eine Ausprägung aus einem Datenbestand, zum Beispiel eine Zeile aus einer Tabelle, die bestimmte Attribute besitzt.

Model: Kann als Muster oder Vorlage gesehen werden. Im Kontext des maschinellen Lernens ist es eine Vorlage, die mit Trainingsdaten gefüttert wird und somit für die Prognosen verwendet werden kann.

Klassifikator: Teilt Instanzen in bestimmte Klassen ein.

Decisiontree: zu Deutsch Entscheidungsbaum, ist ein Lernverfahren, dass aus Trainingsinstanzen und deren Attributen einen Baum bildet. Dass wichtigste Attribut stellt die Wurzel dar und mit absteigender Wichtigkeit der Attribute für die Klassifikation werden so Äste und Blätter für einen Baum gebildet. Sie sind einfach zu bedienen, haben nur relativ kurze Laufzeiten und sind für Benutzer relativ einfach zu verstehen. (Görz et al., 2013, S. 413)

Features: Sind die Merkmale die eine Klasse ausmachen, dass können bestimmte Attribute aber auch ganze Instanzen sein.(Görz et al., 2013, S. 411 ff.)

Prognose: Ist eine Voraussage einer neuen zu klassifizierenden Instanz. Die Prognose wird anhand des Models, dass mit den Trainingsdaten gefüttert wurde und dem Klassifikator durchgeführt.

White- und Black-Box: Blackbox beschreibt den Zustand, dass keine Informationen über die Verarbeitung zwischen In-und Output bestehen. Whitebox beschreibt den gegenteiligen Zustand, dass alle Informationen über Verarbeitung ersichtlich sind.(Büchi & Weck, 1999, S.2)

Super Pixel: Sind kompakte Bildteilstücke, die einheitlich in der Größe sind und gut an Gebietsgrenzen haften. (Achanta et al., 2012, S.2275)

2.2 Forschungsgebiet: Explainability

Mit der vermehrten Anwendung von Black-Box-Klassifikatoren und Machine-Learning-Prognosen für kritische Aufgaben steigt das Interesse an Erklärungsansätzen. Die steigende Anzahl an Publikationen [1] zu diesem Thema zeigt, dass *Explainability* als eigenes Forschungsfeld mehr und mehr an Bedeutung gewinnt. Die sogenannten *Model Explanier Systems“*, die die Vorhersage eines Models für den Menschen verständlich machen sollen, bauen auf den Ergebnissen der Vorhersage auf und verdeutlichen die ausschlaggebenden Features, Zwischenergebnisse oder Lösungswege, wie beispielsweise ein Klassifikator auf sein Resultat gekommen ist. Hierbei besteht die Gefahr eines Paradoxons, da eine transparente Gestaltung für den Menschen wiederum in einem Blackbox-Verfahren geschieht und letztendlich die entscheidenden Daten aufgezeigt werden aber nicht

[1] https://scholar.google.de/

sichergestellt werden kann, dass die Erklärung korrekt ist, da sie wieder in einem Blackbox-System passiert.

Im Folgenden werden ein paar Ansätze an Erklärern für Blackbox-Modelle aufgeführt, die in aktuellen Ausarbeitungen verfolgt werden, um einen kleinen Überblick über das Thema *Explainability* zu schaffen:

- Verständnis von Prognosemodellen mittels *Decisionsets*: Ziel dieses Frameworks ist es die ausschlaggebenden Variablen eines Entscheidungssystems zu extrahieren und mittels *if-then-else* Regeln abzubilden. Thematisch lässt sich dieser Ansatz in das induktive Schließen einordnen, da mit logischen Regeln gearbeitet wird und so auf Regeln basierende Klassifikatoren wie Decisiontrees erklärt werden können. Entscheidend ist jedoch, dass jede Regel für sich unabhängig interpretiert werden kann und so ein echter Benefit für das menschliche Verstehen entsteht. (Lakkaraju, Bach, & Leskovec, 2016, S. 1)
- Erklärung von Black-Box-Klassifizierung basierend auf einer *Matrix Faktorisierung*: Eine analytische Methode die auf generelle Klassifizierungsmodels angewendet werden kann und so Model-Erklärungen liefert. Das Konzept dahinter ist, mittels Kontribution-Matrix Erklärungen in einem eingebetteter Beschränkungsraum mittels Matrix-Faktorisierung zu bekommen. Es werden ähnlich wie bei den *Decisionsets* regelbasierte Erklärungen aus der Kontribution-Matrix abgeleitet (Kim & Seo, 2017)
- - Interpretierbare Erklärungen mittels *vielsagenden Störungen*: Eine Technik die sich überwiegend im Bereich der Bilderkennung und Klassifizierung einsetzen lässt. Im wesentlich werden die Bereiche eines Bildes, die für eine Klassifizierung durch ein komplexes neuronales Netz entscheidend sind ,durch eine Störungsmaske kenntlich gemacht. Dadurch wird dem Nutzer eine visuelle Erklärung der Entscheidungsfindung gegeben. (Fong & Vedaldi, 2017)
- *LIME*, dass den Anspruch verfolgt, jeden Klassifikator oder Regressor mit Hilfe einer lokalen Approximation in einem interpretierbaren Model darzustellen. Dieser holistische Erklärungsansatz wird nun im folgenden Kapitel ausführlich erläutert. (Ribeiro et al., 2016, S. 2)

3 Erklärermodell LIME

LIME hat den Anspruch jede Prognose eines Klassifikators oder Regressors interpretierbar und transparent zu machen. Mit der draus resultierenden Erklärung kann das Vertrauen in das dahinterliegende Model und seiner Prognose erhöht werden. Das von Marco Tulio Ribeiro, Sameer Singh und Carlos Guestrin der University of Washington entwickelte Framework wurde 2016 veröffentlicht und befindet sich seither im stetigen Ausbau (Ribeiro et al., 2016, S.1). Es existiert eine eigene GitHub-Seite[2], in der kurz die Funktionsweise beschrieben wird und eine knappe Anleitung und Codebeispiele enthalten sind.

[2] https://github.com/marcotcr/lime

3.1 Grundidee und Maxime von LIME

Wie jede Erklärungstechnik hat auch LIME die Maxime die Ergebnisse aus Machine-Learning-Prognosen für den Menschen transparent zu machen und somit das Vertrauen in das dahinter liegende Model zu erhöhen. LIME bietet im Gegensatz zu den aufgeführten Erklärsystemen in 2.2 nicht nur die Möglichkeit jedes Model zu erklären, sondern lässt sich auch grob in zeichenbasierte und bildbasierte Erklär-Modelle unterteilen:

1. Die textorientierte Erklärung zielt auf die Sichtbarmachung der für eine bestimmte Prognose ausschlaggebenden Begriffe ab und stellt diese graphisch dar.
2. Die Erklärung eines Bildklassifikators wird durch das graphische Hervorheben von den für die Klassifizierung relevanten Superpixel umgesetzt.

Die durch LIME erstellten zeichen- und bildbasierten Erklärungen helfen den Nutzer*innen, die dahinter liegenden Entscheidungen besser zu verstehen und ihnen zu vertrauen. Generell akzeptieren Menschen Systeme besser, wenn sie darüber mehr Informationen besitzen (Ribeiro et al., 2016, S. 2)

3.2 Formaler Ansatz

The overall goal of LIME is to identify an interpretable model over the interpretable representation that is locally faithful to the classifier" (Ribeiro et al., 2016, S. 3), im Wesentlichen lässt sich diese Definition in folgende Begriffe differenzieren:

Local: Das interpretierbare Model muss lokalgetreu zum originalen Model und dessen Ergebnissen sein, um eine valide Erklärung zu gewährleisten. Ein komplett getreues Model wäre sinnlos, da es in diesem Fall denselben Umfang wie das originale Model hätte.

Interpretable: Die Ergebnisse müssen interpretierbar sein – für den Menschen nachvollziehbar.

Model-agnostic: Der Erklärer sollte mit jedem Model arbeiten können und somit modelunabhänig sein.

Explanations: Es sollte eine repräsentative Auswahl an Features präsentiert werden, die für eine Klassifikation verantwortlich sind. (Ribeiro et al., 2016, S. 3)

3.3 LIME als Erklärer für eine Instanz

Aus diesen vier Eigenschaften lassen sich zwei Kernelemente ableiten, bei denen es darum geht einen Kompromiss zu finden. Zum einen muss das Erklärer-Model interpretierbar und für den Menschen verständlich sein; zum andern sollte es nicht signifikant von der originalen Prognose abweichen (lokalgetreu). Eine interpretierbare Repräsentation muss für den Menschen verständlich sein, so wird

bei Text-Klassifikationen ein Binärvektor genutzt, der die Präsenz eines Wortes V $\in$ {0,1} in Abhängigkeit einer Klassifikation mit einem boolean Wert verdeutlicht. Bei komplexeren Klassifikatoren, wie die von Bildern, wird mit einem Binärvektor der die Abwesenheit oder Präsenz eines Superpixels in Abhängigkeit für die Klassifikation des gesamten Bildes angezeigt. (Ribeiro et al., 2016, S. 2)

Somit gilt:
x ist originale Instanz $\in \mathbb{R}$
x' ist interpretierbare Instanz $\in$ {0,1}

Nachdem für die klassifizierende Instanz eine interpretierbare Repräsentation in Form einer Binärdarstellung gewählt wurde gilt es dennoch nahe der originalen Prognose zu liegen. Dieser Kompromiss lässt sich mittels Funktion abbilden, die möglichst lokalgetreu aber dennoch interpretierbar ist:

$$\xi(x) = argmin\ \Lambda(f, g, \pi) + \Omega(g)$$

Das Kernstück der Funktion ist der Umkreis π, der bestimmt welche Instanzen für die Erklärung x‘ benutzt werden. Der Erklärer LIME nutzt dabei für Texte bzw. Zeichenketten eine Cosinusdistanz und für Bilder eine L2 Instanz. Die entscheidenden Features für die Erklärung, in einer unbekannten komplexen Funktion zu finden, erinnert an instanzbasiertes Lernen mittels K-Nearest-Neighbour (KNN).

Nachdem die Funktion von LIME beschrieben wurde, gilt es nun einen geeigneten Algorithmus für die Implementierung des Instanzen-Erklärers zu definieren 1 Um eine Instanz zu erklären zu können, muss LIME mit dem Algorithmus 1 zunächst das lokale Verhalten von f erlernen; dazu werden Beispielinstanzen um x gezeichnet‘, die im Anschluss im Originalraum wiedergefunden werden, draus ergibt sich $f(z)$. Eine Funktion, welche die Eigenschaften von x‘ wiederspiegelt und auch modelunabhängig funktioniert. Der Algorithmus iteriert über N Samples und sammelt sie in Z. Mittels K-Lasso werden K-Features ausgewählt, die der Prognose des Klassifikators am ehesten entsprechen. (Ribeiro et al., 2016, S. 5) Der Algorithmus erlaubt eine mit K definierte Anzahl an Features, die für die Erklärung der Instanz entscheidend sind, welche wiederum manuell von dem bzw. der Benutzer*in individuell festgelegt werden muss. Ebenfalls manuell wird die Anzahl der Samples (N) festgelegt, die für die Erklärung der Instanz x‘ zu Rate gezogen werden sollen. In der Praxis wird für K ein Wert zwischen fünf und zehn gewählt und N variiert je Anwendungsfall. Die anderen Parameter, wie π, lassen sich über die Standardimplementierung nicht ändern.[3]

Der Algorithmus gibt letztendlich w als Menge aller Features, die für die Erklärung der Instanz x‘ die höchste Relevanz haben, zurück. Textbasierten Features werden standardmäßig auf der Konsole ausgegeben, können jedoch auch in

[3] https://github.com/marcotcr/lime

eine Webseite eingebunden werden. Bei Bildern werden die Superpixel mit der jeweiligen Wahrscheinlichkeit hervorgehoben.

Nun ist es natürlich nicht nur von Interesse eine Instanz x zu erklären, sondern das Modell im Ganzen. Um ein holistisches Verständnis für den Klassifikator bzw. das Model zu bekommen, wird dem bzw. der Nutzer*in eine ausgewählte Menge an Instanz- Feature-Kombinationen in einer Matrix n x d‘ präsentiert.

Die Erklärungsmatrix zeigt alle Features mit dem höchsten Einfluss über alle Instanzen für ein Model. Da der nutzenden Person nur eine Auswahl von allen Kombinationen gezeigt werden soll, wird mit B ein Budget in dem Algorithmus festgelegt, dass der bzw. die Nutzer*in überschauen kann. Konkret ist B eine festgelegte Anzahl an Erklärungen für ein Model.

Der Algorithmus zu *SP LIME* erklärt, wie der 1, nur für alle Instanzen X und sammelt sie in W. Über alle Features als Erklärungen wird die Wichtigkeit für das Model als Gesamtes berechnet und letztendlich mit einem Greedy-Verfahren sortiert, sodass die wichtigste Feature- Instanzen-Kombinationen zuerst kommen und über B eingeschränkt werden können. Daraus resultiert eine Matrix mit einer Dimension von b x b, die das Model anhand von Features und Instanzen erklärt bzw. beschreibt.(Ribeiro et al., 2016, S. 5)

4 Benutzer-Experimente

Nach der Beschreibung von dem Instanz-Erklärer und *SP-LIME*, dem Model-Erklärer, werden die Ergebnisse aus den simulierten Experimenten und Benutzer*inen-Experimenten vorgestellt. Die simulierten Experimente lassen eine Evaluation ohne Testprobanten*innen zu und geben so einen guten Überblick über die Resultate aus LIME und sind mitentscheidend für die Beantwortung der Leitfrage. (Ribeiro et al., 2016, S. 6)

4.1 Simulierte Experimente

Bei den Experimenten wird folgenden Fragen nachgegangen:

- Ist die Erklärung lokalgetreu zur originalen Prognose?
- Kann die Erklärung von LIME mehr Vertrauen in die Prognose wecken?
- Sind die Erklärungen nützlich bei der Evaluierung von Models als Ganzes?

In einem aufgesetzten Experiment müssen jeweils 2000 Bücher und Film Instanzen und deren Reviews als Positiv oder Negativ klassifiziert werden. Es werden dafür folgende Klassifikatoren genutzt:

- Decisiontree
- Nearest Neighbors
- Support Vector Machines

– Logistic Regression

Bei der Evaluierung wird LIME mit parzen Window, einem Greedy-Ansatz und Random verglichen. Das Gütekriterium für die Beantwortung der ersten Frage sind die jeweiligen Recall-Werte.

Recall: TP / (TP + FN); TP = relevante Treffer mit true positive in einer Konfusionmatrix, FN = Falsch negativ in einer Konfusionmatrix. Recall beschreibt wie komplett die Ergebnisse sind.

Parzen windows: Ist eine Kerndichteschätzung, die mit Hilfe von verschiedenen Kernels und statistischen Gesetzmäßigkeiten eine Schätzung von nicht bekannten Verteilungen ermöglicht.(Parzen, 1962)

Bei einem Decisiontree erreicht LIME mit 97 Prozent den höchsten Recall-Wert und zeigt somit, dass es 3 Prozent Diskrepanz zwischen dem Erklärer und dem originalen Decisiontree besteht. Bei dem Experiment mit einer Logistik Regression erreicht LIME noch einen Wert von 90 Prozent . Um den Recall berechnen zu können, wurde ein gold set von 10 Features für jede Instanz festgelegt. Im Anschluss wurden Erklärungen mit Features generiert und diese Features mit denen des gold sets verglichen. Es wurde das Mittel über alle 400 Test-Instanzen genommen. (Ribeiro et al., 2016, S. 7)

Aufbauend auf den Recall-Werten kann nun überprüft werden, ob der Prognose vertraut werden kann. Dafür wurden 25 Prozent der Features als nicht vertrauenswürdig markiert. Es wird davon ausgegangen, dass der bzw. die Benutzer*in die Features erkennen kann, die als nicht vertrauenswürdig markiert wurden. Der berechnete F1 Score liegt für LIME zwischen 90 und 96 von 100, nachdem in den Prognosen die nicht vertrauenswürdigen Features entfernt wurden. Im Vergleich erreicht Parzen Werte zwischen 84 und 94 von 100. Die letzte Frage, ob dem Model vertraut werden kann, simuliert eine Benutzer*innenentscheidung zwischen zwei Modellen, welche Vorhersage valider erscheint. Dafür wurde die Datentabelle um 10 Prozent Störungsdaten ergänzt, mit denen im Anschluss zwei Decisiontrees trainiert wurden. Der bzw. die simulierte Nutzer*in entschied sich bereits nach 10 gesehenen Instanzen für den Decisiontree mit der höheren Genauigkeit. Jedoch liegt das Maximum für eine richtige Entscheidung mit Hilfe von LIME bei ca. 70 Prozent, ein Greedy-Ansatz erreicht einen ähnlichen Wert. Die Ergebnisse aus den simulierten Experimenten geben einen ersten positiven Anhaltspunkt, wie die Unterstützung von LIME bei der Erklärung von Models beitragen kann. Jedoch sagen sie bisher wenig über den Nutzen von LIME für reale Benutzer*innen aus, was ein Kernpunkt der Leitfrage ist. (Ribeiro et al., 2016, S. 7)

4.2 Experimente mit Menschen

Der Fokus bei LIME ist die Unterstützung von Menschen bei komplexen Black-Box-Klassifikationen. Deshalb wird nun an die simulierten Experimente mit

menschlichen Proband*innen angeknüpft. Im Gegensatz zum Abschnitt zuvor bewegen sich die Fragestellungen nun ausschließlich im Bereich der Model-Erklärer. Zum einen wird untersucht, ob ein*e Benutzer*in sich für den besten Klassifikator entscheiden kann und ob auch nicht Machine-Learning-Expert*innen ein Model verbessern können. (Ribeiro et al., 2016, S. 8)

Zum ersten Untersuchungsgegenstand: Um die Frage beantworten zu können, wurde ein Experiment mit 100 Proband*innen aufgezogen, die keine Machine-Learning-Expert*innen sind dafür aber Basiswissen über Religion haben. Passend dazu müssen die Testpersonen anhand von Stichwörtern (Features) entscheiden, welche Klasse diese zugeordnet werden sollen. Die Stichwörter kommen aus E-Mails und werden dann in 20 verschiedenen Klassen eingeordnet; darunter Klassen, wie Atheismus und Christentum. Es gilt nun zu entscheiden, ob der Klassifikator für die Klasse „Atheismus“ die richtigen Stichwörter für seine Entscheidung zu Rate zieht.

Mit der LIME-Darstellung konnten sich 89 Prozent für den korrekten Klassifikator entscheiden, während bei einer Greedy-Darstellung sich nur 80 Prozent der Befragten für den richtigen Klassifizier entschieden haben. Mit diesem Ergebnis konnte gezeigt werden, dass LIME durchaus Benutzer*innen bei der Auswahl von Klassifikatoren bei einer konkreten Problemstellung unterstützen kann. Nun stellt sich abschließend die Frage, ob LIME Testpersonen dabei helfen kann, den Klassifikator zu verbessern, dass sogenannte Feature-Engineering. Ziel hinter der Fragestellung ist, möglichst viele Wörter, die nicht im semantischen Sinne zur Klasse „Atheismus“ passen, zu entfernen. Letztendlich konnten innerhalb von zwei Runden 68 Wörter über alle Teilnehmer*innen identifiziert werden, die bei der Klassifizierung nicht beachtet werden sollen.

Im Bereich der Bild-Klassifizierung wurde ein weiteres Experiment aufgesetzt, bei dem es darum geht, durch die Erklärung Rückschlüsse auf einen nicht vertrauenswürdigen Klassifikator zu ziehen. Dies wurde im Gegensatz zu den vorherigen Experimenten mit Student*innen, die Grundwissen im Machine Learning Bereich haben, durchgeführt. Es geht im Speziellen um die korrekte Klassifizierung eines Canoidea als Hund oder Wolf. Die Deutlichmachung der für die Klassifizierung entscheidenden Superpixel durch LIME konnten bis auf 3 von 27 Student*innen erkennen, dass der Schnee im Hintergrund nicht für die Klassifizierung eines Canoidea benutzt werden sollte und man deshalb dem Klassifikator nicht vertrauen sollte.

Die Experimente mit realen Menschen sowie simulierten Benutzer*innen konnten durchwegs positive Resultate in Bezug auf die Erklärung bzw. das Verstehen für Models und Klassifikatoren und dem daraus entstehenden Vertrauen aufweisen.

5 Praktische Anwendung

LIME bietet, wie bereits im Vorfeld erwähnt, eine kleine Dokumentation sowie Codebeispiele auf GITHUB [4], die im Laufe die letzten Monate immer weiter ausgebaut wurden. LIME ermöglicht eine simple Implementierung in Python und unterstützt Models aus scikit-learn, ein Framework für maschinelle Datenanalyse. Es unterstützt von Decisiontrees bis hin zu Clustering alle gängigen Regressoren und Klassifikatoren. Über die Unterstützung von Keras und Deeplearning gibt es bisher keine Informationen. Für die Umsetzung einer Erklärung implementiert in Python müssen zuerst die entsprechenden Pakete von Scit-learnin (Pedregosa et al., 2011) und LIME importiert werden.

Listing 1.1. Import LIME-Package

```
from lime import lime_text
from lime.lime_text import LimeTextExplainer
explainer = LimeTextExplainer(class_names=class_names)
```

Sobald die Pakete und Daten importiert wurden, kann über den „LimeTextExplainer“ z.B. eine Instanz erklärt werden. Die Anzahl der zu nutzenden Features lässt sich als Parameter setzen:

Listing 1.2. Def. LIME-Explainer

```
exp = explainer.explain_instance(newsgroups_test.data[idx],
 c.predict_proba, 6)
```

Ist der Erklärer festgelegt hat der bzw. die Benutzer*in verschiedene Möglichkeiten sich die Feature-Zusammensetzung anzeigen zu lassen. Diese lassen sich als Liste oder als Anzeige in Balkenform eingebettet in eine HTML-Seite darstellen.

Die Implementierung einer Bild-Erklärung mit Python und LIME erfordert im Gegensatz zur einer simplen Instanz-Erklärung Vorarbeit mit Inception v3 und Tensorflow. Im Anschluss kann wie beim Text auch über „explaininstance“ und einer Masken-Funktion im Erklärer die Instanz erklärt werden

Aus der praktischen Anwendung lässt sich folgern, dass die Implementierung Vorwissen in der Programmierung und der Datenaufbereitung benötigt. Die Nutzung des Erklärers scheint in der Theorie simpel, ließ sich in der Praxis jedoch nicht nachvollziehen.

[4] https://github.com/marcotcr/lime

6 Kritik und Fazit

LIME hat den Anspruch an sich selbst, jeden Klassifikator erklären zu können. Dieses ambitionierte Ziel ist nach der einjährigen Entwicklungszeit noch nicht erreicht. Bei Bildern lassen sich bisher nur Instanzen erklären keine ganzen Models. Weiterhin ist LIME nicht in der Lage Bilder zu erklären, welche sich nicht in Superpixel unterteilen lassen oder sich nur durch bestimmte Farben definieren. Die Ergebnisse der simulierten Benutzer*innenexperimente sind überwiegend positiv, jedoch konnte nur bewiesen werden, dass LIME bei Decisiontrees und Logistik Regression gute Werte hat; zu weiteren gar komplexeren Klassifikatoren gibt es keine Auskünfte. In den Experimenten konnten sich 89 Prozent der Testpersonen für das richtige Model entscheiden, ein gutes Ergebnis, ein einfacher Greedy-Ansatz schafft mit 80 Prozent ein ähnliches Ergebnis. LIME erreicht beim Decisiontree 97 Prozent Recall bei der Logistik Regression im Vergleich nur ca. 90 Prozent. Ob diese Werte des Decisiontrees für Industrie-Standards oder sicherheitsrelevante Anwendungen ausreichen, ist fraglich. Die Implementierung des Explainers in Python gestaltet sich theoretisch einfach. In einem Selbstversuch war es leider nicht möglich dieses Theorem zu bestätigen, da die Nutzung von LIME durch ein fehlerhaftes Paket nicht möglich war. Generell ist eine Implementierung oder auch Anpassung eines Models mit dem LIME-Erklärer in Python ohne umfangreiches Wissen zu Datenvorbereitung und Model-Implementierung nicht durchführbar.

Zusammengefasst stellt LIME nichtsdestotrotz - gerade im Vergleich mit anderen Ansätzen im Gebiet der Machine-Learning-Erklärer – einen Vorreiter bezüglich der Flexibilität und des Umfangs der Anwendungsfälle dar. Es können sowohl Text, Bilder, einzelne Instanzen wie auch ein ganzes Model erklären. Damit vereint LIME viele Eigenschaften, welche andere Erklärungsansätze nur punktuell bedienen. Im Kontext der Leitfrage kann resümiert werden, dass die Erklärung der maschinellen Prognosen auch für nicht Machine-Learning-Expert*innen ersichtlich ist. Somit werden fachliche Diskussionen mit Expert*innen auf dem entsprechenden Gebiet des Machine-Learning-Programms unterstützt; beispielsweise behandelnde Personen bei der Diagnose von Krankheiten. LIME ermöglicht allerdings keine Anpassung von Models oder Klassifikatoren von Nicht-Machine-Learning-Expert*innen. Ob letztendlich mehr Vertrauen oder Akzeptanz gegenüber maschinellen Prognosen durch LIME erwirkt wurde, lässt sich nur durch die Erklärungen implizieren, weitere Aussagen lassen sich an dieser Stelle nicht dazu machen. Ebenfalls lässt sich für Laien nicht erkennen, wie die Erklärung zu Stande kommt, was in gewisser Weise aus LIME einen Black-Box-Erklärer macht. Wie von Entwickler*innen beschrieben, ist das Projekt im Aufbau und soll künftig weitere Klassifikatoren und GPU-Verarbeitung unterstützen. Weitere Forschungen werden von einem Teil des Projektteams unter dem Codenamen „Anchors“ weitergeführt, dass laut dem 2018 veröffentlicht Paper „Anchors: High-Precision Model-Agnostsic Explanations“ sich auf die Erklärung von komplexen Models mit Hilfe von Regeln konzentriert. (Ribeiro, Singh, & Guestrin, 2018)

Literatur

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, *34*(11), 2274–2282.

Büchi, M., & Weck, W. (1999). *The greybox approach: When blackbox specifications hide too much* (Tech. Rep.). Citeseer.

Fong, R. C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*.

Görz, G., Schmid, U., & Wachsmuth, I. (2013). *Handbuch der künstlichen intelligenz*.

Kim, J., & Seo, J. (2017). Human understandable explanation extraction for black-box classification models based on matrix factorization. *arXiv preprint arXiv:1709.06201*.

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1675–1684).

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, *33*(3), 1065–1076.

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.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. , 1-10.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations.

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

Abb. 1. Algorithm 1 Sparse Linear Explanations using LIME