

Institut für Informatik, Goethe-Universität Frankfurt am Main

BENUTZERHANDBUCH - PROTOTYP

Automatisierte Überprüfung von  
Machine Learning Verfahren  
hinsichtlich Erklärbarkeit

*Anil Yelin*

Frankfurt am Main  
25. Oktober 2022

# Inhaltsverzeichnis

<b>1</b>	<b>Hintergrund</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>2</b>
<b>3</b>	<b>Main Version - User Interface</b>	<b>3</b>
3.1	Schnelle Navigation . . . . .	3
3.2	Dataset Overview . . . . .	4
3.3	Model Training . . . . .	5
3.4	Explainability Section . . . . .	5
3.5	Feature Importance based on SHAP values . . . . .	6
3.6	Explainability Checker Framework . . . . .	7
3.6.1	Tab - Component Consistency . . . . .	8
3.6.2	Tab - Component Robustness . . . . .	10
3.6.3	Tab - Component Stability . . . . .	12
3.6.4	Tab - Component Simplicity . . . . .	13
3.6.5	Tab - Component Feature Importance . . . . .	15
3.6.6	Summary Section . . . . .	16
<b>4</b>	<b>Custom Version - User Interface</b>	<b>17</b>

# 1 Hintergrund

Der Prototyp wurde im Kontext der Masterarbeit mit dem Titel *Automatisierte Überprüfung von Machine Learning Verfahren hinsichtlich Erklärbarkeit* entwickelt.

Im Grundsatz werden die in der Masterarbeit entwickelten algorithmischen Bausteine exemplarisch implementiert. Der Prototyp ist in Python entwickelt worden und verwendet im Wesentlichen Bibliotheken:

- **Streamlit** (Erstellung einer webbasierten Programms für den Browser)
- **Pandas** (Datenverarbeitung, -manipulation)
- **Scikit Learn** (Verwendung der Machine Learning Algorithmen)
- **SHAP** (Bibliothek zur Anwendung des XAI Verfahrens SHAP)
- **eli5** (Bibliothek zur Durchführung von Permutation Feature Importance)
- **Matplotlib** (Bibliothek zur Erstellung von Diagrammen, Grafiken)
- **Numpy** (Bibliothek für effiziente wissenschaftliche Berechnungen)
- und die zugehörigen Abhängigkeiten bzgl. der obigen Bibliotheken

# 2 Installation

Der gesamte Quellcode des Prototypen findet sich auf Github unter folgendem Link:

[github.com/anilyelin/AutomatedXAI](https://github.com/anilyelin/AutomatedXAI)

Die wesentlichen Installationsschritte sind wie folgt:

1. Es empfiehlt sich die Python Paketverwaltungssoftware Anaconda zu installieren. Mit dieser Software ist es möglich verschiedene Environments (Umgebungen) mit verschiedenen Modulen zu generieren.
2. Das Projekt auf den lokalen Rechner klonen

`git clone https://github.com/anilyelin/AutomatedXAI.git`

3. Nachdem das Projekt geklont wurde, muss ein Environment generiert werden mit den relevanten Bibliotheken, die für den Prototypen notwendig sind. Diese finden sich in der Datei *requirements.txt*. Im Kommandozeileninterpreter/Terminal ist folgender Befehl einzugeben, wenn man ein Environment basierend auf einem requirements.txt erstellen will.

```
[  
conda create --name <env_name> --file requirements.txt
```

4. Navigiere in das Verzeichnis

```
cd AutomatedXAI/src/main
```

5. Starten der Streamlit App

```
streamlit run streamlit-test.py
```

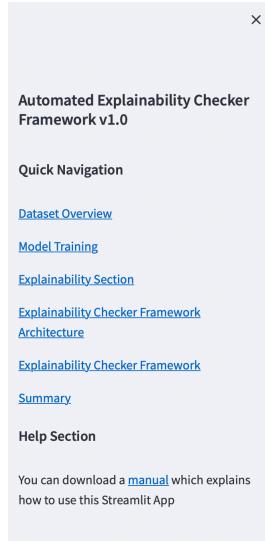
6. Im Regelfall öffnet sich dann automatisch der Browser mit der besagten Streamlit Applikation, falls dies nicht der Fall ist, kann man die URL aus dem Terminal in den Browser kopieren

## 3 Main Version - User Interface

Dieser Abschnitt behandelt die Main Version der insgesamt zwei Streamlit Applikationen, welcher auf dem Parkinson Datensatz basiert. Die zweite Version namens **automated-xai-custom**, womit eigene Datensätze verwendet werden können, wird im nächsten Abschnitt behandelt. Die vorliegende Streamlit App ist eine Single Page Anwendung, d.h. alle Funktionalitäten finden sich auf einer einzigen Seite.

### 3.1 Schnelle Navigation

Auf der linken Seite befindet sich eine Sidebar, welche als schnelle Navigation zwischen den einzelnen Sektionen der Applikation fungiert. Beim Klicken auf den entsprechend Abschnitt, gelangt automatisch zu der jeweiligen Stelle.



## 3.2 Dataset Overview

In diesem Abschnitt wird der verwendete Datensatz tabellarisch aufgezeigt, es sind alle Spalten und die ersten fünf Datensätze zu sehen. Bei Bedarf kann der gesamte Datensatz über den Download Button runtergeladen werden.

### Dataset Overview

Parkinson Dataset

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP
0	phon_R01_S01_1	119.9920	157.3020	74.9970	0.0078	0.0001	0.0
1	phon_R01_S01_2	122.4000	148.6500	113.8190	0.0097	0.0001	0.0
2	phon_R01_S01_3	116.6820	131.1110	111.5550	0.0105	0.0001	0.0
3	phon_R01_S01_4	116.6760	137.8710	111.3660	0.0100	0.0001	0.0
4	phon_R01_S01_5	116.0140	141.7810	110.6550	0.0128	0.0001	0.0

[Download as csv file](#)

Abbildung 1: Screenshot zum Abschnitt Dataset Overview

### 3.3 Model Training

In diesem Abschnitt werden die beiden Black Box Machine Learning Verfahren Random Forest Classifier sowie Extra Trees Classifier mittels diversen Parametern konfiguriert. Es sind bereits optimale Parameter als Default Werte eingetragen, jedoch lassen sich alle Werte ändern.

**Model Training**

[Random Forest Classifier](#)   [Extra Trees Classifier](#)

Please choose the first black box model for training

Random Forest Classifier

**Random Forest Classifier**

Please enter the number of estimators

300

Please enter a random state number

42

Please enter the size of the test dataset

0,20

Please enter the max depth for a tree

30

Hyperparameter Summary

#Estimators	Random State	Tree Max Depth	Test Size	Training Size	Model Score
300	42	30	0.2	0.8	0.9487

Abbildung 2: Screenshot zum Abschnitt Model Training

Es ist anzumerken, dass die Parameter die selben für beide Modelle sind, um eine gewisse Vergleichbarkeit herzustellen, d.h. wenn man auf den Tab Extra Trees Classifier klickt, sieht man, dass die Parameter aus dem ersten Tab übernommen wurden und sich nicht ändern lassen.

### 3.4 Explainability Section

In diesem Abschnitt wird kurz das XAI Verfahren vorgestellt, womit in diesem Prototypen die Erklärungen generiert werden. Es handelt sich um das Verfahren SHAP. Die ausführliche theoretische Erläuterung findet sich in der Thesis.

## Explainability Section

This prototype will make use of SHAP to calculate the Shapley values for the features in the dataset  
Shapley values will measure the marginal contribution to the outcome of the model

$$\phi_i(p) = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S))$$

SHAP is based on a strong mathematical foundation with several properties such as additivity, null player and symmetry. In the thesis can be found a more detailed explanation on the theoretical foundations of SHAP

Abbildung 3: Screenshot zur Explainability Section

### 3.5 Feature Importance based on SHAP values

In diesem Abschnitt wird eine globale Sicht auf die Erklärbarkeit des Modells geworfen. Es wird eine Grafik präsentiert, die die wichtigsten Features hinsichtlich der Shapley Werte aufzeigt.

### Feature Importance based on SHAP values

The following bar chart below is showing the most important features according . It should not be confused with Permutation Feature Importance which is a part of the proposed framework.

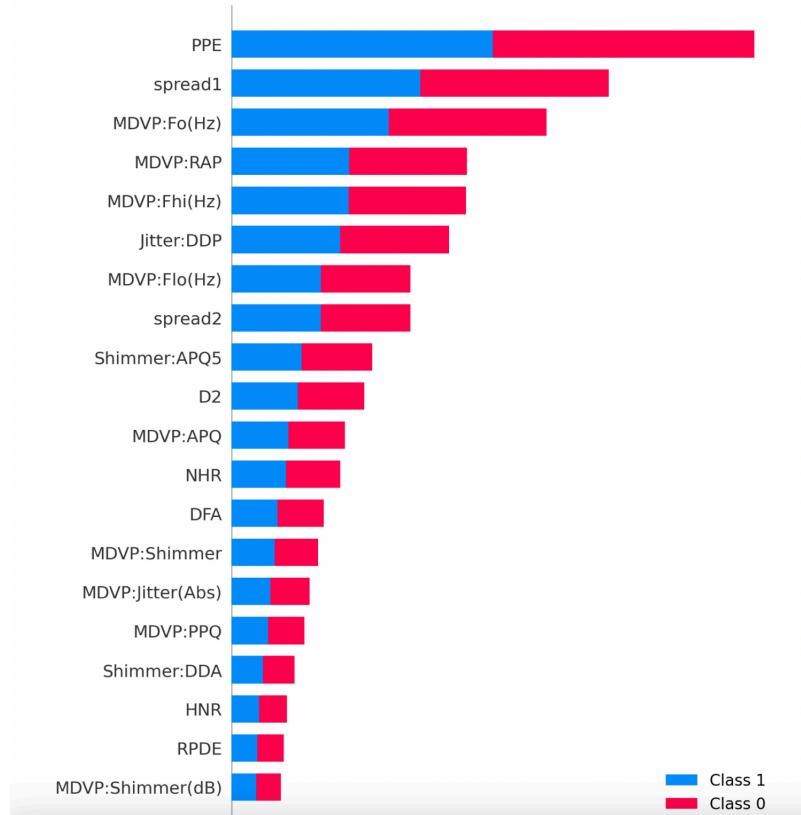


Abbildung 4: Screenshot zur Feature Importance based on SHAP values

### 3.6 Explainability Checker Framework

In diesem Abschnitt wird das eigentliche Framework vorgestellt, vorher wird noch die High Level Architektur mittels einer Grafik präsentiert. Das konzeptionelle Framework besitzt insgesamt fünf Komponenten und diese sind mittels fünf Tabs implementiert worden.

Abbildung 5: Screenshots Tabs

### 3.6.1 Tab - Component Consistency

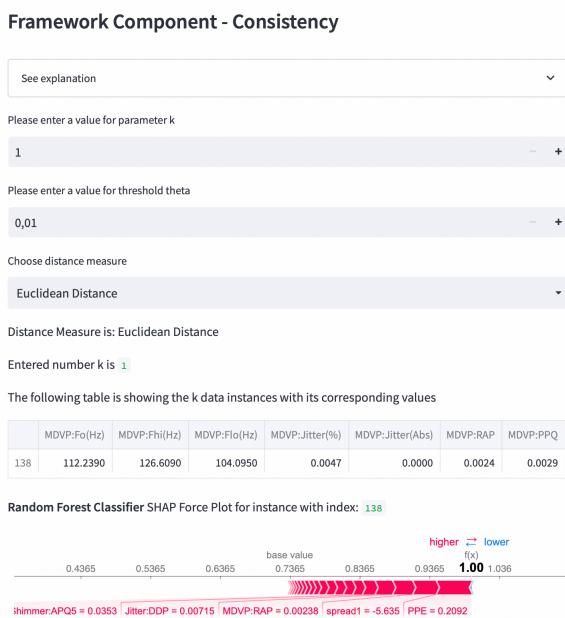


Abbildung 6: Screenshot Komponente Konsistenz

In dem Tab **Component Consistency** können verschiedene Parameter eingegeben werden. Zum einen der Parameter  $k$ , welcher die Anzahl an zu untersuchenden Dateninstanzen angibt sowie der Parameter  $\theta$ , der die obere Schranke definiert. Es können manuell Zahlen eingegeben werden oder auf der rechten Seite mittels + und – können die Werte inkrementiert und dekrementiert werden. Weiterhin ist das zugrundeliegende Distanzmaß die euklidische Norm, womit die Abstände zwischen den SHAP Vektoren bestimmt werden. Gemäß dem gewählten Parameter  $k$  wird anschließend eine Tabelle mit den  $k$  Dateninstanzen gezeigt wie auf der Abbildung 6 zu sehen ist. Als visuelle Komponente für die generierte Erklärung mittels der Python Bibliothek SHAP, dient der sogenannte SHAP Forceplot, welcher die Einflüsse

(positiv und negativ) der einzelnen Features anhand der berechneten SHAP Scores visualisiert. Dieser Forceplot wird jeweils für beide Verfahren (RFC und ETC) pro Instanz ausgegeben.

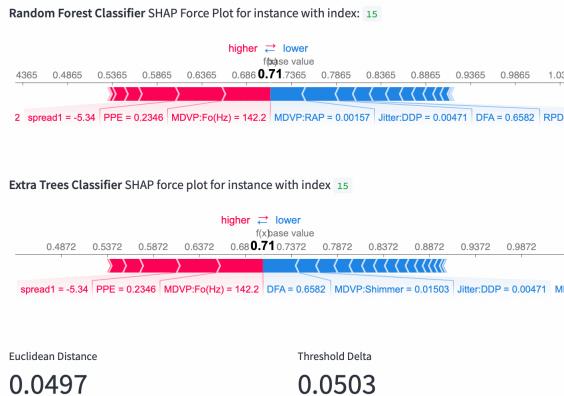


Abbildung 7: Euklidische Distanz und Schwellwert Delta

Außerdem wird für jede Dateninstanz  $i$  der berechnete euklidischen Abstand ausgegeben und sowie die Differenz der gewählten oberen Schranke  $\theta$  und der eukl. Distanz. Ist dieser Delta Werte negativ, bedeutet dies, dass die Schranke nicht eingehalten wird. Abbildung 7 zeigt exemplarisch wie die Werte für die Dateninstanz 15 aussehen.

Als letzter Teil in diesem Tab dient eine Tabelle zur Zusammenfassung der erzielten Werte wie in Abbildung 8 zu sehen ist.

## [Consistency] Summary Table

Below you can see a table with all results with respect to your parameters.

	Data Instance (index)	Euclidean Distance SHAP Vectors: RFC <-> ETC	Threshold Delta 0.1
1	138	0.0414	0.0586
2	16	0.0426	0.0574
3	155	0.0400	0.0600
4	96	0.0443	0.0557
5	68	0.0501	0.0499
6	153	0.0459	0.0541
7	55	0.0340	0.0660
8	15	0.0497	0.0503
9	112	0.0605	0.0395
10	111	0.0351	0.0649

[Download results as csv file](#)

Threshold of: `0.1` is not maintained for `9` instances of `10` instances (in total)

Consistency Score: `100.0 %`

Abbildung 8: Summary Table

### 3.6.2 Tab - Component Robustness

Die Eingabeoptionen sind im nahezu identisch zum Tab **Component Consistency** mit dem wesentlichen Unterschied, dass ein Button zur automatisierten Ausführung von marginalen Unterschieden gedrückt werden muss (Abbildung 9). Um brauchbare Ergebnisse zu erhalten, muss dieser Button von der Endanwenderin oder dem Endanwender gedrückt werden. Es gibt außerdem die Option bei Bedarf manuelle Änderungen an den Eingabedaten durchzuführen.

Component Consistency Component Robustness Component Stability Component Simplicity Component Feature

### Framework Component - Robustness

See explanation ▾

Please enter a value for k

 - +

Abbildung 9: Screenshot vom Tab Robustness

Die nachfolgende Abbildung zeigt wie anhand von Eingabemasken, die Werte der Features manuell geändert werden können.

Manual changes

### Manual Marginal Changes

Please enter an index to modify the data manually

 - +

Abbildung 10: Screenshot manuelle Änderungen

Am Ende des Tabs gibt wie bei allen anderen Komponenten eine Summary Table zur Übersicht der Ergebnisse. Da die automatisierte marginale Än-

derung auf einer Normalverteilung basiert, die auf die einzelnen Werte der Features hinzugaddiert werden, kann dementsprechend zu unterschiedlichen Ergebnissen kommen. Dies liegt an der stochastischen Natur des Verfahrens.

### 3.6.3 Tab - Component Stability

Die Eingabemaske unterscheidet sich marginal zu den anderen Tabs. Hier wird automatisiert nach  $k$  benachbarten Dateninstanzen mit dem gleichen Target gesucht, um anschließend die generierten Erklärungen zu vergleichen. Wie in Abbildung 11 zu sehen ist, kann man die benachbarten Dateninstanzen mit den Indizes 111 und 112 sehen sowie das Delta (die Differenz der Featurewerte) in einer tabellarischen Übersicht.

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ
111	208.5190	220.3150	199.0200	0.0061	0.0000	0.0037	0.0034
112	204.6640	221.3000	189.6210	0.0084	0.0000	0.0050	0.0049

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ
112	-3.8550	0.9850	-9.3990	0.0023	0.0000	0.0013	0.0015

Abbildung 11: Screenshot vom Tab Stability

Es werden die Ergebnisse zum einen für das RFC Verfahren sowie das ETC Verfahren ausgegeben. D.h. der euklidische Abstand der benachbarten Da-

teninstanzen 111 und 112 jeweils für beide Verfahren und ob die Schranke  $\theta$  eingehalten wird.

RFC Euclidean Distance <b>0.0789</b>	Threshold Value <b>-0.0689</b>
RFC: Threshold is not maintained	
ETC Euclidean Distance <b>0.0871</b>	Theta Delta <b>-0.0771</b>
ETC threshold is not maintained	

Abbildung 12: Ausgabe der Ergebnisse für die Stabilitätskomponente

### 3.6.4 Tab - Component Simplicity

Für diese Komponente ist Eingabemaske identisch zu den vorherigen Komponenten, jedoch unterscheidet sich die Ausgabe der Ergebnisse. Für diese Komponente wird analysiert, wie viele SHAP Werte verbleiben, wenn man sehr kleine und negative SHAP Werte basierend auf einer Cut Off Threshold abschneidet.

Component Consistency Component Robustness Component Stability **Component Simplicity** Component Feature

### Simplicity

[Simplicity Cutoff] Please enter a value for parameter k

The entered number for parameter k is: **1**

SHAP values below the cut off threshold will be ignored since they don't contribute much to the final result

Please enter a cut off threshold

The cutoff threshold is: **0.01**

The following table is showing the k data instances

	MDVP:Fo[Hz]	MDVP:Fhi[Hz]	MDVP:Flo[Hz]	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ
138	112.2390	126.6090	104.0950	0.0047	0.0000	0.0024	0.0029

Abbildung 13: Screenshot des Tabs Simplicity

Es können weiterhin die einzelnen SHAP Werte für beide Verfahren tabellarisch aufgelistet werden.

## RFC SHAP Values

For instance: [111](#)

[RFC] Show particular shap values for given instance	
	0
0	-0.0634
1	0.0090
2	-0.0744
3	-0.0076
4	0.0037
5	0.0281
6	-0.0029
7	0.0225
8	-0.0168
9	-0.0075

RFC Score for given instance: [18.18 %](#)

Abbildung 14: SHAP Werte für eine Instanz

Nachfolgend ist die Übersichtstabelle sowie die finalen Scores zu sehen.

## [Simplicity Summary Table]

Below one can find the results for the k instances

	Index	RFC Simpl. Score	ETC Simpl. Score
1	138	45.4500	54.5500
2	16	36.3600	36.3600
3	155	50.0000	45.4500
4	96	36.3600	36.3600
5	68	45.4500	68.1800
6	153	45.4500	31.8200
7	55	40.9100	36.3600
8	15	22.7300	22.7300
9	112	31.8200	27.2700
10	111	18.1800	18.1800

[RFC] Final Score (Average): 37.27 %

[ETC] Final Score (Average): 37.73 %

ETC has a better final score for simplicity component

Abbildung 15: Screenshot der Summary Table für die Komponente Simplicity

### 3.6.5 Tab - Component Feature Importance

Diese Komponente läuft vollständig automatisiert ab, daher gibt es keine Eingabemaske wie bei den vorherigen Komponenten.

	feature	weight	std
0	spread1	0.0256	0.0000
1	PPE	0.0205	0.0103
2	MDVP:Fhi(Hz)	0.0205	0.0103
3	DFA	0.0103	0.0126
4	Jitter:DDP	0.0051	0.0103
5	spread2	0.0000	0.0000
6	RPDE	0.0000	0.0000
7	HNR	0.0000	0.0000
8	NHR	0.0000	0.0000
9	Shimmer:DDA	0.0000	0.0000

[Download results as csv file](#)

Total Weight: 0.0821

### 3.6.6 Summary Section

Dieser Abschnitt in der Single Page Applikation ist dazu gedacht, alle Tabellen aus den jeweiligen Komponenten zusammenfassend darzustellen. Auf diese Weise hat man alle Ergebnisse im Überblick.

## Summary

Below you can find all relevant tables and scores at one place

### Consistency Check

	Data Instance (index)	Euclidean Distance SHAP Vectors: RFC <--> ETC	Threshold Delta 0.1
1	138	0.0414	0.0586
2	16	0.0426	0.0574
3	155	0.0400	0.0600
4	96	0.0443	0.0557
5	68	0.0501	0.0499
6	153	0.0459	0.0541
7	55	0.0340	0.0660
8	15	0.0497	0.0503
9	112	0.0605	0.0395
10	111	0.0351	0.0649

Consistency Score: 100.0 %

## 4 Custom Version - User Interface

Diese Version des Prototypen ermöglicht das Hochladen von eigenen Datensätzen in dem .csv Format. Es gibt jedoch einige wesentliche Änderungen im Vergleich zur Main Version, welche maßgeschneidert auf den Parkinson Datensatz ist. Der hochgeladene Datensatz muss i.d.R. einem Preprocessing unterzogen werden. Diese beinhalten folgende Schritte

- Auswahl des Target für das Klassifikationsproblems

Please select the column which is the target for the model training

Select the target

sepal.length
 

- sepal.length
- sepal.width
- petal.length
- petal.width
- variety

- Label Encoding für Spalten ohne numerische Daten

There is/are non numeric column(s) in the dataset

Please press the button below to apply Label Encoding for columns without numeric data

**Apply Label Encoding**

Calling function for label encoding...

variety has no numeric values

Dataset after label encoding

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1000	3.5000	1.4000	0.2000	0
1	4.9000	3.0000	1.4000	0.2000	0
2	4.7000	3.2000	1.3000	0.2000	0
3	4.6000	3.1000	1.5000	0.2000	0
4	5.0000	3.6000	1.4000	0.2000	0
5	5.4000	3.9000	1.7000	0.4000	0
6	4.6000	3.4000	1.4000	0.3000	0
7	5.0000	3.4000	1.5000	0.2000	0
8	4.4000	2.9000	1.4000	0.2000	0
9	4.9000	3.1000	1.5000	0.1000	0

- gegebenenfalls das Löschen von nicht relevanten Spalten

## Drop columns

If you want to drop specific columns of the dataset, use the button below

Column

sepal.length ×variety ×⋮

Drop Column

Die restlichen Komponenten (Erklärbarkeitsüberprüfung und -bewertung) sind identisch zur Main Version.