

Institut für Informatik, Goethe-Universität Frankfurt am Main

BENUTZERHANDBUCH - PROTOTYP

Automatisierte Überprüfung von Machine Learning Verfahren hinsichtlich Erklärbarkeit

Anil Yelin

Frankfurt am Main
10. Oktober 2022

Inhaltsverzeichnis

1	Hintergrund	2
2	Installation	2
3	User Interface	3
3.1	Dataset Overview	3
3.2	Model Training	4
3.3	Explainability Section	5
3.4	Feature Importance based on SHAP values	5
3.5	Explainability Checker Framework	6
3.5.1	Tab - Component Consistency	7
3.5.2	Tab - Component Robustness	7
3.5.3	Tab - Component Stability	7
3.5.4	Tab - Component Simplicity	7
3.5.5	Tab - Component Feature Importance	7

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

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 Enviroment 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 Enviroment 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 User Interface

Die vorliegende Streamlit App ist eine Single Page Anwendung, d.h. alle Funktionalitäten finden sich auf einer einzigen Seite.

3.1 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

3.2 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.

3.3 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.

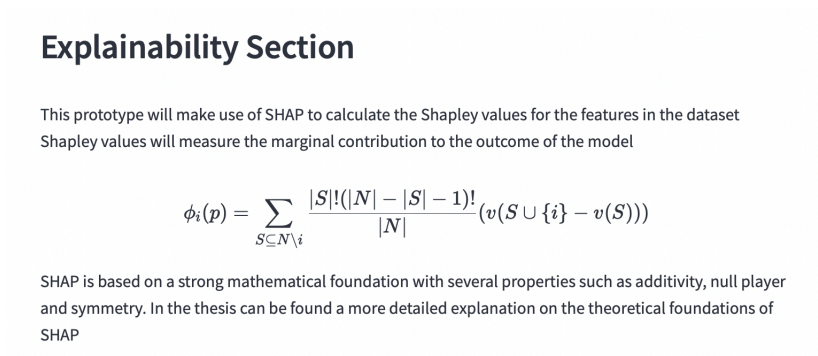


Abbildung 3: Screenshot zur Explainability Section

3.4 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.

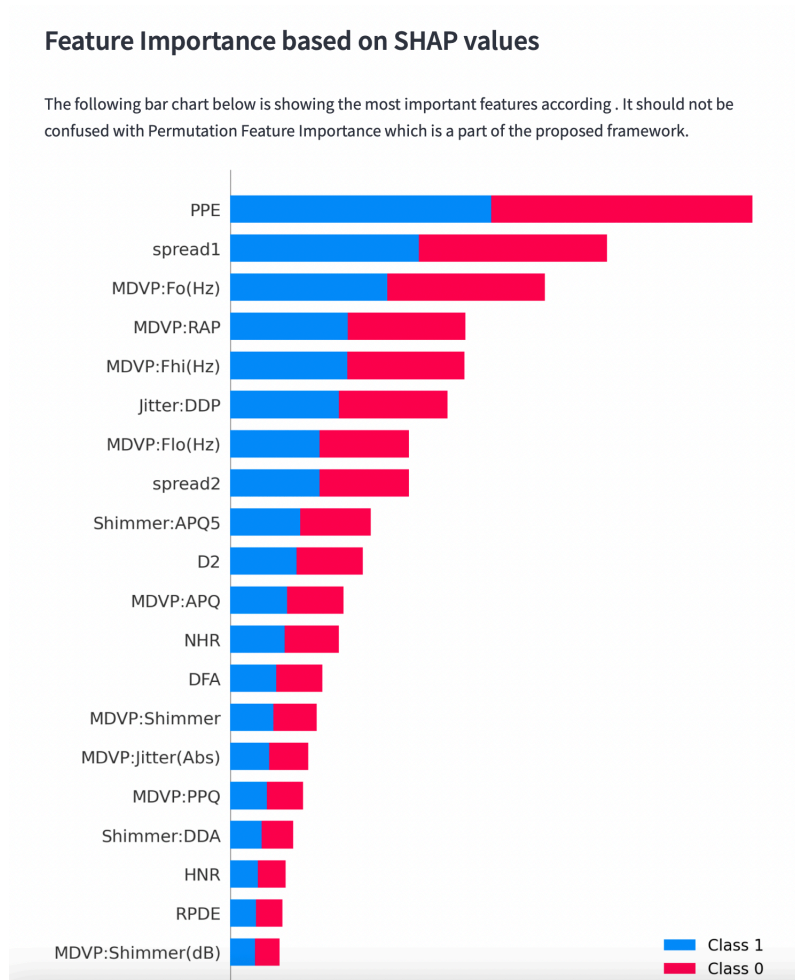


Abbildung 4: Screenshot zur Feature Importance based on SHAP values

3.5 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.

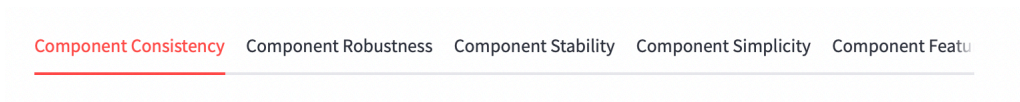
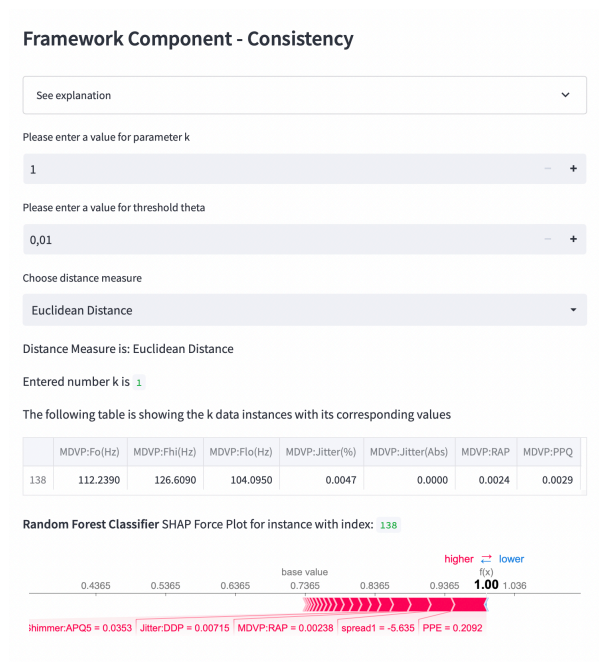


Abbildung 5: Screenshots Tabs

3.5.1 Tab - Component Consistency



3.5.2 Tab - Component Robustness

3.5.3 Tab - Component Stability

3.5.4 Tab - Component Simplicity

3.5.5 Tab - Component Feature Importance

Literatur