Inhaltsverzeichnis

[Einleitung 1](#_Toc524103095)

[Anleitung zum Ausführen der Programme 1](#_Toc524103096)

[Voraussetzungen 1](#_Toc524103097)

[Random Forest 1](#_Toc524103098)

[Multilayer Perceptron 1](#_Toc524103099)

[Konzeption eines symbolischen Lernverfahrens 1](#_Toc524103100)

[Gründe für Random Forest 1](#_Toc524103101)

[Architektur des Modells 1](#_Toc524103102)

[Ergebnisse des Trainings 1](#_Toc524103103)

[Konzeption eines subsymbolischen Lernverfahrens 1](#_Toc524103104)

[Gründe für Multilayer Perceptron und Tensorflow 1](#_Toc524103105)

[Architektur des Modells 1](#_Toc524103106)

[Ergebnisse des Trainings 1](#_Toc524103107)

# Einleitung

# Anleitung zum Ausführen der Programme

## Voraussetzungen

## Random Forest

Für die Ausführung der Random Forest Implementierung müssen die folgenden Python Packages installiert werden:

* scipy
* pandas
* scikit-learn

Um das Programm auszuführen, muss die Datei **main.py** per Kommandozeile, mittels **python main.py**, ausgeführt werden.

## Multilayer Perceptron

# Konzeption eines symbolischen Lernverfahrens

## Gründe für Random Forest

Ein Random Forest ist ein Klassifizierer, der aus mehreren Entscheidungbäumen besteht. Jeder dieser Entscheidungsbäume wird randomisiert erzeugt. Für die eigentliche Klassifizierung wird jeder Entscheidungsbaum befragt. Die Klasse mit den meisten Stimmen bildet das Endergebnis.

Dabei sprechen die folgenden Gründe für die Verwendung eines Random Forest:

* Die Input-Daten müssen kaum aufbereitet werden, da ein Random Forest mit binären, kategorischen und numerischen Attributen umgehen kann.
* Ein Random Forest sützt sich implizit auf die wichtigsten Attribute eines Datensatzes.
* Der Klassifizierer kann einfach und vorallem schnell angelernt werden. Im Gegensatz zu einem Neuronalen Netz ist die Genauigkeit eventuell etwas geringer, jedoch ist der Aufwand um einen Random Forest zu erzeugen deutlich geringer.
* Der Klassifizierungs-Prozess kann parallelisiert werden.
* Es gibt viele gute open-source Implementierungen, für diese Ausarbeitung wurde der RandomForestClassifier aus Scikit-Learn (http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html) verwendet.

## Ergebnisse des Trainings

Nach der Vorbereitung des Datensatzes, musste zunächst die optimale Anzahl der Entscheidungsbäume evaluiert werden. Beginnend mit einem Baum, wurde Prediction-Vorgang mit bis zu 1000 Bäumen ausgeführt. Mit dem Ergebnis, dass 827 Bäume eine Genauigkeit von 99,3% erreichen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Präzision** | **Recall** | **F1-Score** | **Support** |
| **0** | 99% | 100% | 100% | 886 |
| **1** | 100% | 94% | 97% | 115 |
| Avg/ Total | 99% | 99% | 99% | 1001 |

# Konzeption eines subsymbolischen Lernverfahrens

## Gründe für Multilayer Perceptron und Tensorflow

## Architektur des Modells

## Ergebnisse des Trainings