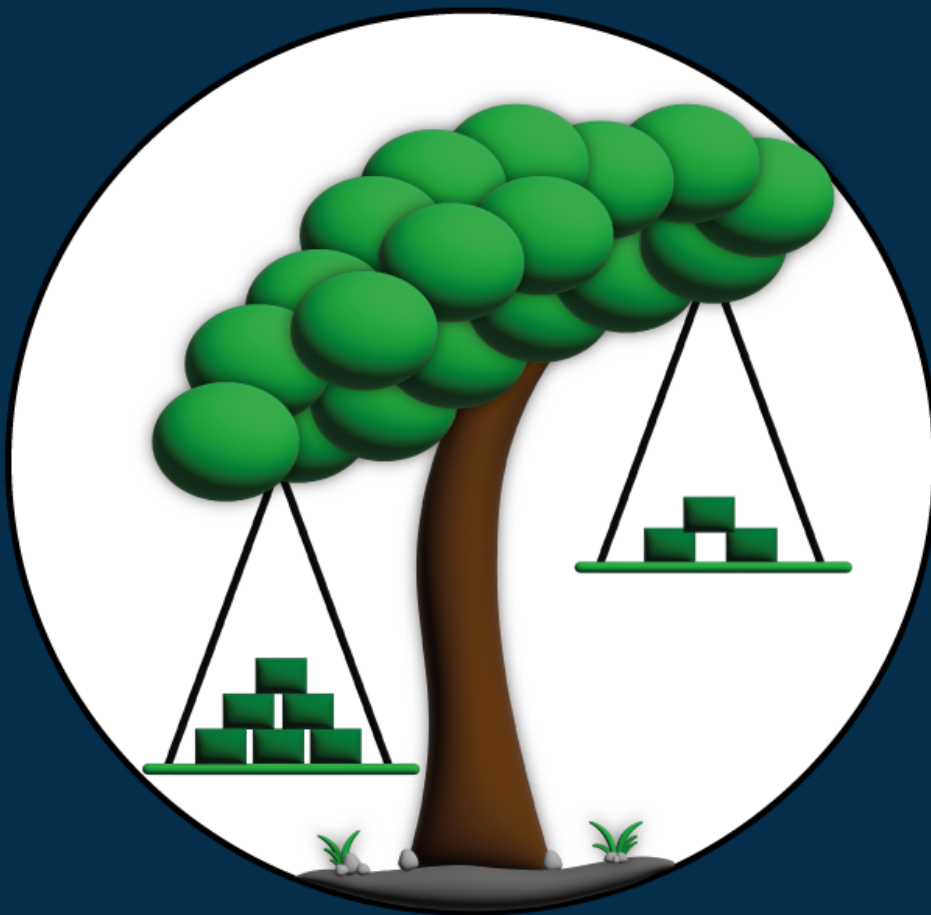


Handbuch

Simulationssoftware Entscheidungsbaum



Konzeption des Handbuchs:

Das Handbuch dient zur Erklärung der für die Lehrerfortbildung „KI@Informatik 11 – was, wozu, wie, womit unterrichten“ der Didaktik der Informatik der Universität Passau zum Thema „Künstliche Intelligenz“ in der 11. Jahrgangsstufe entwickelten Software zur Simulation und didaktischen Visualisierung von Entscheidungsbäumen.

Dr. Wolfgang Pfeffer

Dominicus-von-Linprun-Gymnasium Viechtach

E-Mail: schule@pfeffer-wolfgang.de

Tobias Fuchs

Universität Passau

E-Mail: fuchs_unipa@outlook.de

Weiteres Material findet sich im Mebis-Kurs „Material zu den KI-Fortbildungen der Universität Passau“ mit der ID-Nummer 1324361. Den Einschreibeschlüssel zum Kurs erhalten Sie auf Anfrage.



Inhaltsverzeichnis

1. Verwendung des Simulators	5
1.1 Herstellen der Voraussetzungen: Python und das Paket Graphviz, sowie die Graphviz-Software .	5
1.2 Ausführen des Entscheidungsbaum Simulators	6
1.2.1 Herunterladen der Software <i>Entscheidungsbaum Simulator</i>	7
1.2.2 Ausführen des <i>Entscheidungsbaum Simulators</i> (Windows):	7
1.2.3 Ausführen des <i>Entscheidungsbaum Simulators</i> (MacOS):	10
2. Die Simulationssoftware	12
2.1 Trainingsphase	12
2.1.1 Auswahl der Trainingsdaten	12
2.1.2 Auswahl der Trainingsparameter und Training des Entscheidungsbaums	13
2.1.3 Automatische Aufteilung in Trainings- und Testdaten	15
2.1.4 Baumdetails anzeigen	16
2.1.5 Ansicht Trainingsdaten	16
2.1.6 Ansicht Entscheidungsbaum	17
2.2 Testphase	17
2.2.1 Testmodus aktivieren	17
2.2.2 Ansicht Testphase	18
2.2.3 Ansicht Konfusionsmatrix und Gütemaß	19
3. Weitere Informationen	21



Überblick

Die Simulationssoftware *Entscheidungsbaum-Simulator* ist im Rahmen der Fortbildungsinitiative KI für die Lehrerfortbildung „KI@Informatik 11 – was, wozu, wie, womit unterrichten“ der Didaktik der Informatik der Universität Passau als didaktische Software entwickelt worden. Der Simulator dient als Hilfsmittel für die Umsetzung des Themas *Entscheidungsbaum* des LehrplanPLUS-Kapitels 11.4 Künstliche Intelligenz des bayerischen Lehrplans. Mit Hilfe der Software kann sowohl die Trainings- und Testphase des *Algorithmus maschinellen Lernens am Beispiel des Entscheidungsbaum-Algorithmus* simuliert, als auch der *Einfluss von Trainingsdaten und Parametern auf die Zuverlässigkeit der Ergebnisse eines Verfahrens maschinellen Lernens* untersucht werden.

Die Software bietet sowohl die Möglichkeit einen Entscheidungsbaum unter Verwendung verschiedener Split-Kriterien (u.a. *Fehlklassifikationsrate*, *Gini-Impurity*, ...) unter Einbezug verschiedener Hyperparameter zu erstellen und graphisch darzustellen, als auch die Güte des erstellten Baums anhand von Testdaten zu testen, die Ergebnisse in Form einer Konfusionsmatrix anzuzeigen und das Modell anhand des Gütemaßes *Genauigkeit* zu bewerten. Der Simulator ermöglicht außerdem die genauere Untersuchung eines erstellten Entscheidungsbaums indem man sich die verwendeten (Teil-) Datensätze in den einzelnen Knoten, sowie die dort berechneten Informationsgewinne anzeigen lassen kann. Des Weiteren kann ein geladener Datensatz automatisch in Trainings- und Testdaten geteilt werden um den Einfluss verschiedener Trainingsdaten auf die Modellerstellung zu simulieren. Für das Training und Testen des Entscheidungsbaums können dabei mitgelieferte Trainings- und Testdaten in Form von .csv-Dateien verwendet werden. Alternativ können auch eigene Trainings- und Testdaten in den Simulator geladen werden.

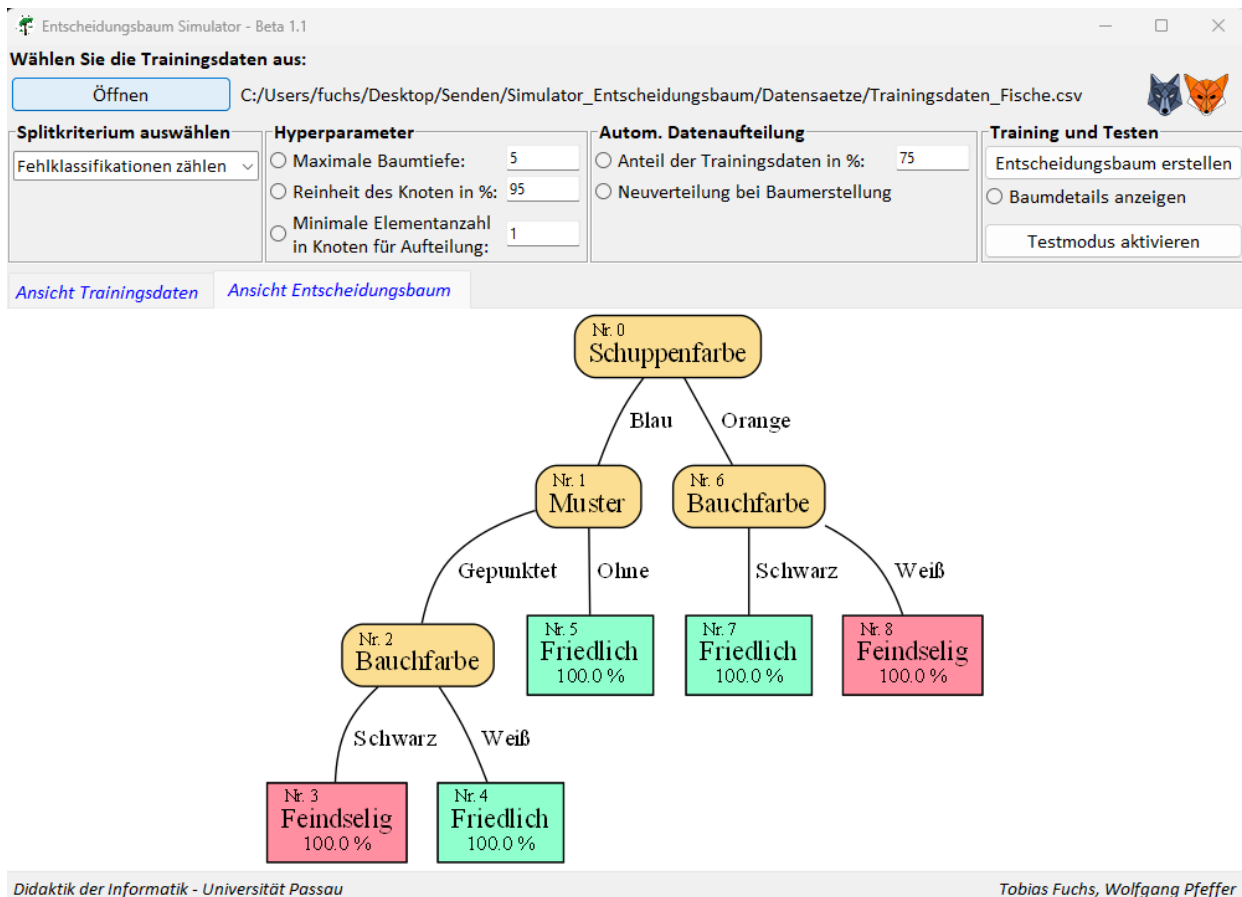


Abbildung 1: Simulator mit dem für den Datensatz *Trainingdaten_Fische.csv* erstellten Entscheidungsbaum



1. Verwendung des Simulators

Die Simulationssoftware *Entscheidungsbaum Simulator* wird in Form von Python-Skripten zur Verfügung gestellt und ist deshalb problemlos unter Windows nutzbar. Eine Nutzung unter MacOS und Linux ist prinzipiell auch möglich, allerdings wurde die Software noch nicht für diese Systeme getestet und optimiert. Damit Sie die Simulationssoftware *Entscheidungsbaum Simulator* ausführen können, muss auf Ihrem System lediglich **Python** mit dem Paket **Graphviz**, sowie die Graphviz-Software installiert sein.

Hinweis:

Es wird später auch kurz erläutert, wie Sie sich für Windows und MacOS eine ausführbare Datei des Simulators erstellen können, welche dann auf einem Rechner ohne vorherige Installation von Python und dem oben genannten Paket ausführbar ist. Lediglich die Graphviz-Software muss separat über den entsprechenden Installer installiert werden.

1.1 Herstellen der Voraussetzungen: Python und das Paket Graphviz, sowie die Graphviz-Software

Installation der Software Graphviz

Der Simulator benötigt für die Darstellung des Entscheidungsbaums in graphischer Form im Hintergrund eine Installation der Software Graphviz bzw. deren Bibliotheken. Diese lässt sich einfach und einmalig mit Hilfe des von Graphviz zur Verfügung gestellten Installers installieren.

1. Laden Sie sich deshalb auf der Graphviz-Homepage <https://graphviz.org/download/> den Installer für Ihr System herunter (Mitte der Seite).
2. **!Wichtig! Aktivieren** Sie bei Ausführung des Installers **die Option** *Add Graphviz to the system PATH* (s. Abb. 2, rote Markierung). Den restlichen Installer können Sie so durchklicken.

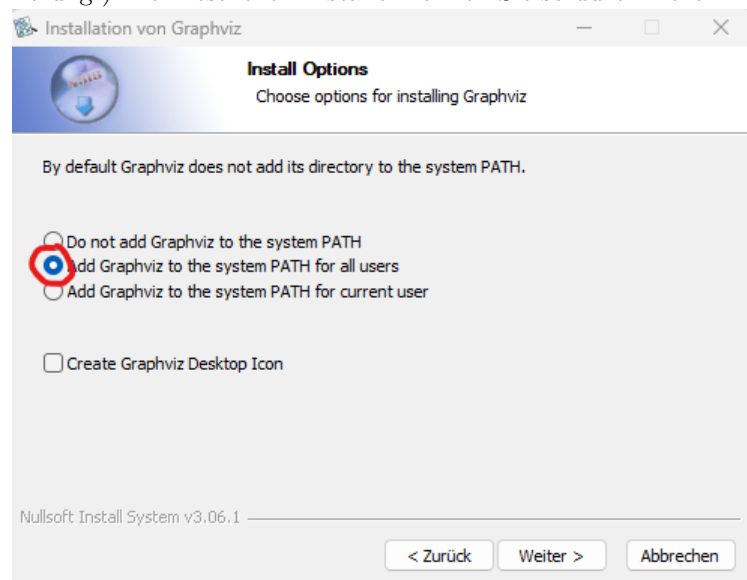


Abbildung 2: Screenshot des Graphviz-Installers mit aktivierter Option zum Hinzufügen zu PATH

Weitere Informationen und eine ausführliche Installationsanleitung finden Sie unter

<https://forum.graphviz.org/t/new-simplified-installation-procedure-on-windows/224>



Installation von Python

Nutzen Sie für die Installation von Python am besten den auf der Python-Homepage zur Verfügung gestellten Installer. Dieser richtet, neben der Installation, Python auf Ihrem System ein.

1. Laden Sie sich den Python-Installer für Ihr System unter <https://www.python.org/downloads/> herunter.
2. **!Wichtig! Setzen** Sie im ersten Fenster des Installers **den Haken** bei „Add python.exe to PATH“ (s. Abb. 3 Stelle 1.).
3. Klicken Sie dann auf „Install Now“ (s. Abb. 3 Stelle 2.). Dadurch wird Python installiert, die notwendigen Tools eingerichtet und Python für die Verwendung vorbereitet.

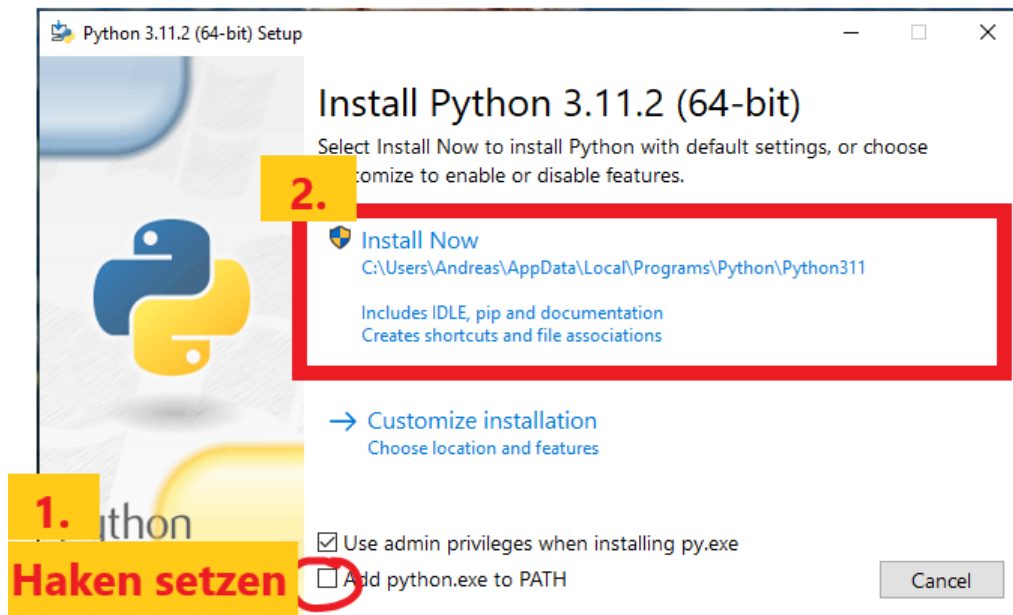


Abbildung 3: Screenshot des Python-Installers für Microsoft Windows

Installation des Pakets (Graphviz)

Am einfachsten lassen sich Pakete für Python mit dem **Paket-Installer pip** installieren. Dieser wird bei Nutzung des Installers zur Installation von Python automatisch mitinstalliert.

1. Öffnen Sie die Kommandozeile/Eingabeaufforderung (Windows).
2. Geben Sie dort den Befehl

```
pip install graphviz
```

ein und bestätigen Sie den Befehl mit der Eingabetaste. Das Paket Graphviz wird nun automatisch heruntergeladen und installiert.

1.2 Ausführen des Entscheidungsbaum Simulators

Im Folgenden werden unterschiedliche Möglichkeiten aufgezeigt, wie der Simulator gestartet werden kann. Sie können darunter diejenige Variante wählen, welche Ihnen am liebsten ist und zu Ihrem System passt.

Hinweis:

Die aufgeführten Möglichkeiten stellen keine abschließende Liste an Startmöglichkeiten dar. Es gibt darüber hinaus noch viele weitere Möglichkeiten ein Python-Skript auszuführen. Wir haben uns in der folgenden Auflistung



allerdings auf einzelne Möglichkeiten beschränkt, die für den Einsatz im Unterricht praktikabel und einfach umsetzbar sind.

1.2.1 Herunterladen der Software *Entscheidungsbaum Simulator*

- (a) Laden Sie sich die aktuellste Version des *Entscheidungsbaum Simulators* unter https://github.com/D3rFuch5/Simulator_Entscheidungsbaum.git durch Klick auf *Code* -> *Download ZIP* herunter. Alternativ finden Sie die ZIP-Datei auch unter <https://www.ddi.fim.uni-passau.de/ki-gym>
- (b) Entpacken Sie die .zip-Datei. Der entpackte Ordner sollte folgende Struktur besitzen.

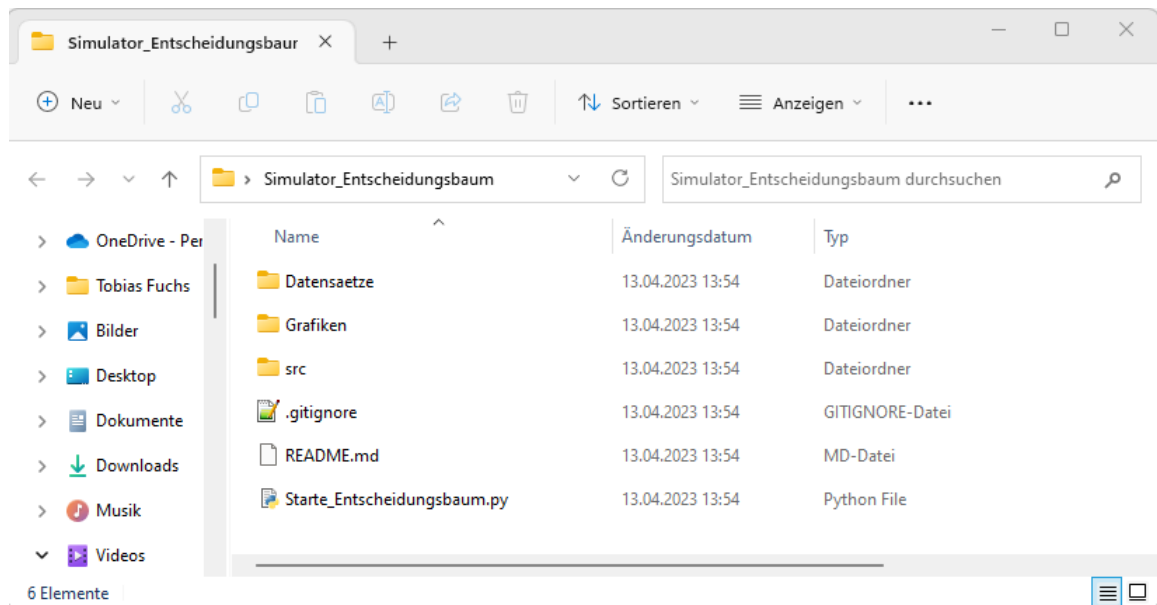


Abbildung 4: Screenshot des entpackten ZIP-Ordners des *Entscheidungsbaum Simulators*

1.2.2 Ausführen des *Entscheidungsbaum Simulators* (Windows):

Möglichkeit 1: Doppelklick auf *Starte_Entscheidungsbaum.py*

Doppelklicken Sie auf die Datei *Starte_Entscheidungsbaum.py*. Es öffnet sich die Oberfläche des *Entscheidungsbaum Simulators*.

Hinweis 1:

Durch Installation von Python mit dem Installer sollte Python als Standardprogramm für die Ausführung von .py-Dateien festgelegt sein.

Falls nicht, können Sie durch Rechtsklick auf die Datei *Starte_Entscheidungsbaum.py* unter *Öffnen mit* -> *Python* Python für die Ausführung der Datei auswählen.

Hinweis 2:

Wollen Sie verhindern, dass sich neben der Oberfläche des *Entscheidungsbaum Simulators* auch eine Konsole öffnet, ändern Sie die Dateiendung der Datei *Starte_Entscheidungsbaum.py* zu *Starte_Entscheidungsbaum.pyw*.

Dies verhindert, dass beim Start, neben der Oberfläche des Simulators auch eine Konsole geöffnet wird.

**Möglichkeit 2: Eingabe von `python Starte_Entscheidungsbaum.py` in die Konsole**

1. Öffnen Sie den entpackten Ordner.

2. Öffnen Sie eine Konsole für den **geöffneten** Ordner.

Rechtsklicken Sie hierfür in den Ordner und wählen *In Terminal öffnen* aus. Es öffnet sich eine Konsole, welche bereits den Pfad des Ordners besitzt.

3. Geben Sie in der Konsole den Befehl

`python Starte_Entscheidungsbaum.py`

ein und bestätigen Sie den Befehl mit der Eingabetaste. Es öffnet sich die Oberfläche des *Entscheidungsbaum Simulators*.

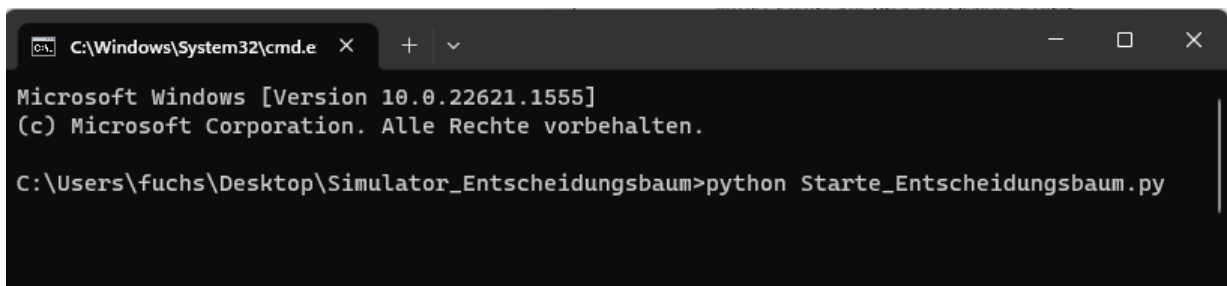


Abbildung 5: Starten des *Entscheidungsbaum Simulators* mittels einer Konsole für den aktuellen Ordner

Hinweis:

Bei älteren Windows-Versionen ist die Auswahl *Im Terminal öffnen* nicht verfügbar.

Klicken Sie in einem solchen Fall in die Pfadzeile des geöffneten Ordners (s. Abb. 6 Teil 1.).

Geben Sie in die Pfadzeile `cmd` ein und bestätigen Sie den Befehl mit der Eingabetaste (s. Abb. 6 Teil 2.).

Es öffnet sich nun auch eine Konsole mit dem Dateipfad des geöffneten Ordners (s. Abb. 6 Teil 3.).

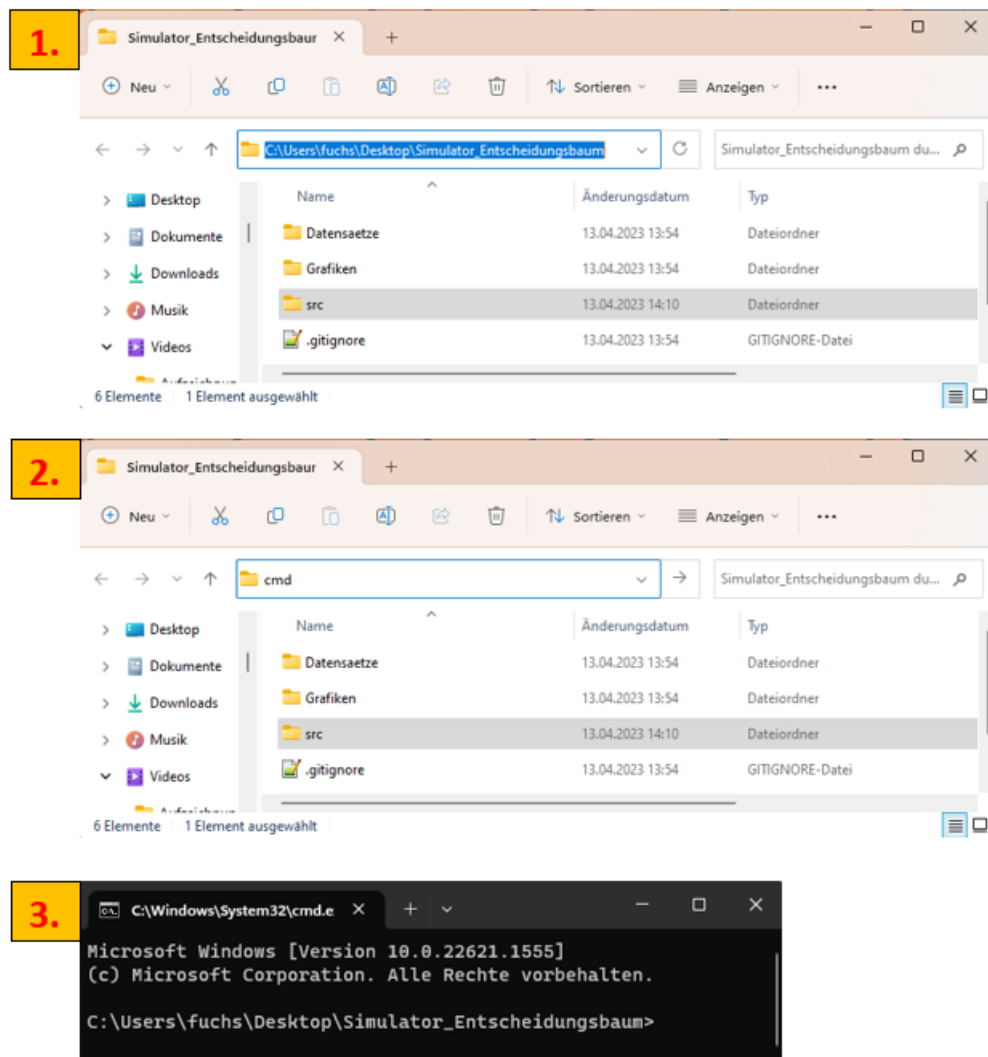


Abbildung 6: Öffnen einer Konsole für den aktuellen Ordner

Möglichkeit 3: Start mit IDLE

Die Abkürzung IDLE steht für „*Integrated Development and Learning Environment*“ und bezeichnet eine Entwicklungsumgebung, welche bei einer normalen Python-Installation mit an Bord ist und nicht extra installiert werden muss.

1. Öffnen Sie IDLE aus den installierten Programmen.
2. Öffnen Sie die Datei `Starte_Entscheidungsbaum.py` über „*File -> Open*“.
3. Drücken Sie `F5` um den Simulator zu starten.

Alternativ klicken Sie im Fenster der geöffneten Datei auf „*Run -> Run Module*“.

Möglichkeit 4: Erstellen einer ausführbaren .exe-Datei mit dem Paket PyInstaller

Mit Hilfe des Pakets *PyInstaller* (<https://pyinstaller.org/en/v4.2/index.html>) ist es möglich aus Python-Skripten ausführbare Dateien für Windows und MacOS zu erstellen. Eine solche ausführbare Datei kann auf einem beliebigen Rechner des jeweiligen Systems ausgeführt werden, ohne dass dafür eine Installation von Python oder zusätzlicher Pakete notwendig ist, da diese in der ausführbaren Datei „gebündelt“ werden. Die Installation der Graphvi-Software (siehe 1.1) muss dennoch auf dem Rechner vorhanden sein, da diese nicht in der ausführbaren Datei „gebündelt“ wird.



Auf dem Rechner, auf dem die ausführbare Datei erstellt wird, muss Python mit den notwendigen Paketen installiert sein.

- *Installation des PyInstallers:*

Öffnen Sie eine Konsole und geben dort den Befehl

```
pip install pyinstaller
```

ein und bestätigen Sie den Befehl mit der Eingabetaste. Das Paket Pyinstaller wird nun automatisch heruntergeladen und installiert.

- *Erstellen der ausführbaren .exe-Datei:*

1. Öffnen Sie die Konsole **im Ordner** des Entscheidungsbaum-Simulators, d.h. in dem sich die Datei `Starte_Entscheidungsbaum.py` befindet.

2. Geben Sie den Befehl

```
pyinstaller --noconfirm --windowed --onefile --name Simulator_Entscheidungsbaum --icon  
Grafiken/Logo_Entscheidungsbaum.png Starte_Entscheidungsbaum.py
```

ein und bestätigen Sie den Befehl mit der Eingabetaste.

Die ausführbare Datei wird nun erstellt. Das kann einige Minuten dauern.

3. Im Ordner des Entscheidungsbaum-Simulators wurde automatisch ein Ordner `dist` erstellt, welcher die ausführbare Datei enthält.

Kopieren Sie den Ordner *Grafiken* **in** den erstellten `dist`-Ordner.

4. Doppelklick auf die `.exe`-Datei startet nun den *Entscheidungsbaum Simulator*.

5. Wenn Sie die ausführbare Datei auf einem anderen Rechner nutzen wollen, kopieren Sie den `dist`-Ordner (mit enthaltenem Ordner *Grafiken*) auf den anderen Rechner. Auf diesem muss kein Python installiert sein. Lediglich die Graphviz-Software (siehe [1.1](#)) muss installiert sein.

Hinweis:

Die Lizenzen der ggf. gebündelten Dateien sind zu beachten.

1.2.3 Ausführen des *Entscheidungsbaum Simulators* (MacOS):

Unter MacOS kann es, je nach verwendetem System, vereinzelt zu Darstellungsfehlern (u.a. Abschneiden eines Teils der Oberfläche) kommen. Der Fehler ist bekannt und es wird an einer zeitnahen Lösung gearbeitet. Sobald der Fehler behoben ist, wird eine neue Version des Simulators zur Verfügung gestellt und dieser Hinweis entfernt.

Möglichkeit 1: Verwenden des Python Launchers

Mit Python 3 wurde der *Python-Launcher*, ein Programm zum Ausführen von Python-Skripten über den Finder hinzugefügt.

1. Suchen Sie dazu die Python-Skriptdatei `Starte_Entscheidungsbaum.py` im Finder.
2. Klicken Sie anschließend mit der rechten Maustaste auf die Datei und wählen Sie im Kontextmenü „*Öffnen mit -> Python Launcher*“.

Alternativ können Sie das Python-Skript `Starte_Entscheidungsbaum.py` auch auf das Python Launcher-Symbol ziehen, sei es im Dock oder im Anwendungsordner.



Möglichkeit 2: Verwenden des Terminals

1. Öffnen Sie ein Terminal **für den Ordner**, welcher die Datei `Starte_Entscheidungsbaum.py` enthält.
2. Geben Sie im Terminal den Befehl

```
python3 Starte_Entscheidungsbaum.py
```

ein und bestätigten Sie den Befehl mit der Eingabetaste. Es öffnet sich die Oberfläche des *Entscheidungsbaum Simulators*.

Hinweis: Wie öffnet man ein Terminal im gewünschten Ordner aus dem Finder?

1. Öffnen Sie ein Finder-Fenster und navigieren Sie dann zu dem Ordner, den Sie verwenden möchten.
2. Wenn Sie die Pfadleiste nicht am unteren Rand des Finder-Fensters sehen, wählen Sie „*Darstellung*“ -> „*Pfadleiste einblenden*“.
3. Klicken Sie bei gedrückter Taste „*ctrl*“ auf den Ordner in der Pfadleiste und wählen Sie dann „*In Terminal öffnen*“.

Möglichkeit 3: Erstellen einer .app-Datei mit dem Paket PyInstaller

Die Einrichtung des *PyInstallers* und die Erstellung einer ausführbaren .app-Datei erfolgt analog zu der Beschreibung der Erstellung einer .exe-Datei für Windows.

Es muss lediglich eine andere Icon-Datei im Format .icns verwendet werden. Alternativ kann auch auf ein Icon verzichtet werden. Dafür muss einfach der Befehlsteil `--icon Grafiken/Logo_Entscheidungsbaum.png` aus dem obigen Befehl weggelassen werden.

Hinweis:

Die Lizenzen der ggf. gebündelten Dateien sind zu beachten.



2. Die Simulationssoftware

Der *Entscheidungsbaum-Simulator* ermöglicht die Simulation der Trainingsphase (siehe 2.1) und Testphase (siehe 2.2) des maschinellen Lernprozesses des Entscheidungsbaum-Algorithmus.

2.1 Trainingsphase

Beim Starten des Simulators ist die Oberfläche für die Trainingsphase vorbereitet. Im Folgenden wird beschrieben, welche Schritte nötig sind um ein trainiertes Modell, also einen Entscheidungsbaum, zu erhalten.

2.1.1 Auswahl der Trainingsdaten

Durch Klick auf den Button *Öffnen* im Bereich *Wählen Sie die Trainingsdaten aus:* (siehe Abb. 7) öffnet sich ein Kontextmenü über das Sie Ihre Trainingsdaten auswählen können.

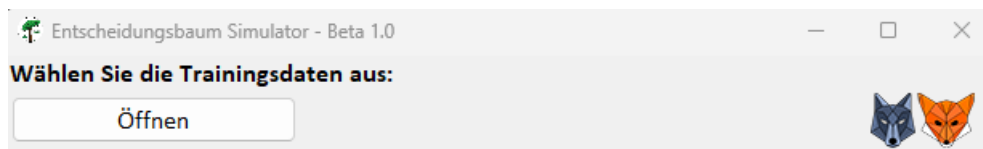


Abbildung 7: Auswahlmöglichkeit für die Trainingsdaten

Entsprechen die Daten dem notwendigen Format, werden diese automatisch geladen und im Tab *Ansicht Trainingsdaten* angezeigt.

Die Trainingsdaten müssen in Form einer *.csv*-Datei vorliegen. In der ersten Zeile stehen die Benennungen der betrachteten Merkmale(hier: *Muster*, *Flossenfarbe*,...) und die Benennung der Labels palte (hier: *Label*). In den folgenden Zeilen stehen nun die Werte der einzelnen Trainingsdatenpunkte entsprechend der festgelegten Reihenfolge der Merkmale.

1	Muster;Flossenfarbe;Bauchfarbe;Schuppenfarbe;Label
2	Gepunktet;Rot;Schwarz;Blau;Feindselig
3	Gepunktet;Gelb;Schwarz;Blau;Feindselig
4	Gepunktet;Rot;Weiß;Blau;Friedlich
5	Gepunktet;Rot;Schwarz;Orange;Friedlich
6	Ohne;Rot;Weiß;Orange;Feindselig
7	Gepunktet;Gelb;Weiß;Orange;Feindselig
8	Ohne;Gelb;Weiß;Orange;Feindselig
9	Ohne;Gelb;Schwarz;Blau;Friedlich
10	Ohne;Gelb;Weiß;Blau;Friedlich

Abbildung 8: Mitgelieferter Trainingsdatensatz *Trainingsdaten_Fische.csv*

Der letzte Eintrag jeder Zeile stellt dabei immer das *erwartete Label* (hier: *Feindselig* oder *Friedlich*) des aktuellen Trainingsdatenpunkts dar.

Beachten Sie bitte, dass die einzelnen Einträge durch ein ; („Strichpunkt“) voneinander getrennt sein müssen.

Hinweis:

Im Ordner *Datensaetze* werden bereits einzelne Beispieldatensätze mitgeliefert. Diese können Sie direkt verwenden oder sich an diesen bei der Erstellung eigener Datensätze orientieren.



2.1.2 Auswahl der Trainingsparameter und Training des Entscheidungsbaums

Nachdem die Trainingsdaten geladen sind, können noch die Parameter des Entscheidungsbaum-Algorithmus festgelegt werden.

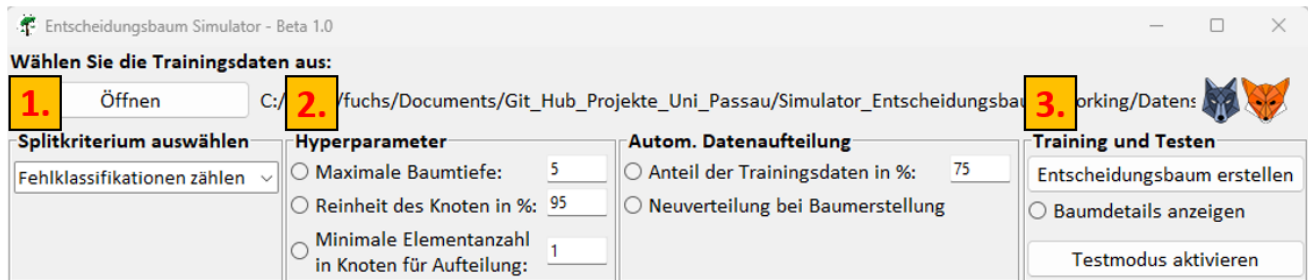


Abbildung 9: Parameterauswahl für den Entscheidungsbaum-Algorithmus

Auswahl des Splitkriteriums: (siehe Abb. 9 Markierung 1.)

Im Bereich *Splitkriterium auswählen* können Sie das Kriterium auswählen, welches für die Bestimmung des **besten Attributs**, also die Berechnung des Informationsgewinns, verwendet wird.

Zur Auswahl stehen folgende Splitkriterien:

- *Fehlklassifikationen zählen*

Hierbei handelt es sich um das in der *Handreichung Künstliche Intelligenz 11* des ISB bzw. in den Fortbildungen *KI@Informatik 11 – was, wozu, wie, womit unterrichten* vorgestellte Verfahren, bei dem für die Bestimmung des Informationsgewinns die Anzahl an fehlerhaften Klassifikationen vor und nach der potentiellen Aufteilung des Datensatzes betrachtet werden. Da dieses Verfahren für den Unterricht am geeignetsten erscheint, ist es als Standard vorausgewählt.

- *Fehlklassifikationsrate*
- *Gini-Impurity*
- *Entropie*

Da in diesem Handbuch der Fokus auf der Anwendung des Simulators liegt, wird für die fachliche Erklärung der Splitkriterien *Fehlklassifikationsrate*, *Gini-Impurity* und *Entropie* auf die ISB-Handreichung oder auf die Fortbildungen *KI@Informatik 11 – was, wozu, wie, womit unterrichten* der Universität Passau verwiesen.

(Optional) Festlegen der Hyperparameter: (siehe Abb. 9 Markierung 2.)

Im Bereich *Hyperparameter* können Sie unterschiedliche Hyperparameter festlegen, welche bei der Erstellung des Entscheidungsbaums berücksichtigt werden sollen.

Verfügbare Hyperparameter:

- *Maximale Baumtiefe:*

Dieser Hyperparameter bestimmt die maximale Tiefe des Entscheidungsbaums.

Ist der Hyperparameter aktiv mit eingetragenen Wert 5, hat der erstellte Baum eine maximale Baumtiefe von 5. Ein Baum mit Tiefe 0 besteht nur aus der Wurzel.

Als Eingabe sind nur nicht-negative, ganze Zahlen zugelassen.

- *Reinheit des Knoten in %:*

Dieser Hyperparameter bestimmt die minimale Reinheit eines Knoten in Prozent, welche dieser mindestens



haben muss, damit er nicht weiter aufgeteilt wird.

Ist der Hyperparameter aktiv mit eingetragenem Wert 60, wird ein Knoten nicht weiter aufgeteilt, wenn mindestens 60 % seiner Trainingsdaten zu einer Klasse gehören. Andernfalls wird weiter aufgeteilt.

Als Eingabe sind nur nicht-negative, ganze Zahlen im Bereich von 0 bis 100 zugelassen.

- *Minimale Elementzahl in Knoten für Aufteilung:*

Dieser Hyperparameter bestimmt die minimale Anzahl an Trainingsdatenpunkten, welche der zu einem Knoten gehörende Teildatensatz haben muss für eine weitere Aufteilung. Ist diese minimale Anzahl unterschritten, findet keine weitere Aufteilung des Knoten statt.

Ist der Hyperparameter aktiv mit eingetragenem Wert 5, wird ein Knoten nur aufgeteilt, wenn der zum Knoten gehörende Trainingsdatensatz aus 5 oder mehr Elementen besteht.

Als Eingabe sind nur positive, ganze Zahlen größer gleich 1 zugelassen.

Damit ein Hyperparameter bei der Baumerstellung berücksichtigt wird, muss er über den entsprechenden Auswahlbutton aktiviert und der gewünschte Wert muss in dem jeweiligen Eingabefeld eingetragen werden. Wurde ein zulässiger Wert eingetragen, wird dieser bei der *nächsten* Baumerstellung berücksichtigt. Andernfalls wird dann eine Fehlermeldung ausgegeben.

Abbildung 10: Aktivierter Hyperparameter *Maximale Baumtiefe* mit Wert 5

Training und Testen: (siehe Abb. 9 Markierung 3.)

Im Bereich *Training und Testen* können Sie den Entscheidungsbaum trainieren. Durch Klick auf den Button *Entscheidungsbaum erstellen* wird der Baum trainiert, sofern Trainingsdaten geladen wurden und ggf. aktivierte Hyperparameter zulässige Eingaben besitzen.

Der erstellte Entscheidungsbaum wird im Tab *Ansicht Entscheidungsbaum* angezeigt.

Auf die Auswahl *Baumdetails anzeigen* (siehe 2.1.4) und den Button *Testmodus aktivieren* (siehe 2.2.1) wird an späterer Stelle eingegangen.

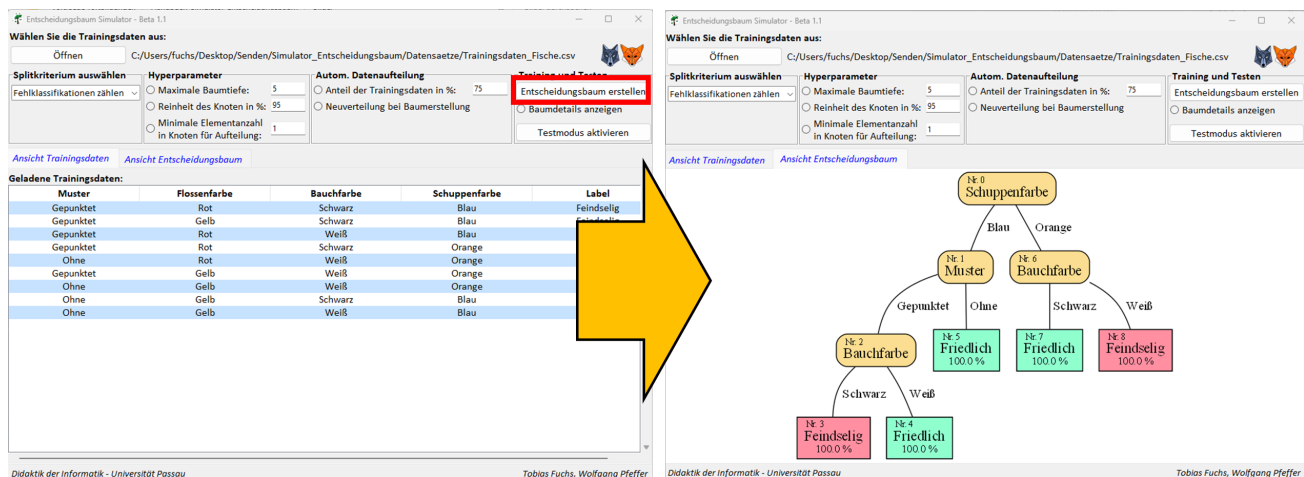


Abbildung 11: *Links:* Ansicht mit geladenen Trainingsdaten *Rechts:* Anzeige des durch Klick auf *Entscheidungsbaum erstellen* trainierten Entscheidungsbaums



2.1.3 Automatische Aufteilung in Trainings- und Testdaten

Der *Entscheidungsbaum Simulator* bietet die Möglichkeit einen Datensatz zu laden, welcher dann automatisch nach einem eingegebenen Prozentsatz in Trainings- und Testdaten aufgeteilt wird. Es ist somit nicht notwendig die Daten vorab manuell in Trainings- und Testdaten zu unterteilen.

Diese Funktion muss im Bereich *Autom. Datenaufteilung* durch Aktivierung des Auswahlbuttons *Anteil der Trainingsdaten in %* aktiviert werden.

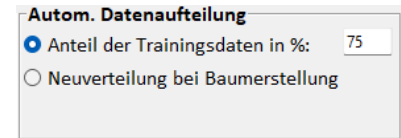


Abbildung 12: Bereich *Autom. Datenaufteilung* mit aktivierter Aufteilungsfunktion

Aktivierung/Deaktivierung der automatischen Datenaufteilung:

Ist bereits ein Datensatz als Trainingsdatensatz geladen, wird dieser bei Aktivierung der Funktion als allgemeiner Datensatz behandelt und in Trainings- und Testdaten aufgeteilt.

Falls bei Aktivierung noch keine Daten geladen sind, wird der unter *Wählen Sie die Daten aus* (identisch mit dem Bereich zum Laden der Trainingsdaten bei nicht aktivierter automatischer Aufteilung) geladene Datensatz für die Aufteilung verwendet. Der als Trainingsdatensatz verwendete Teildatensatz wird automatisch im Tab *Ansicht Trainingsdaten* angezeigt.

Bei Deaktivierung der automatischen Aufteilung wird der geladene Datensatz komplett als Trainingsdatensatz verwendet.

Der Wert des Eingabefelds neben dem Auswahlbutton *Anteil der Trainingsdaten in %*: legt fest, wie viele Datenobjekte (Anzahl aufgerundet) des geladenen Datensatzes entsprechend dem eingegebenen Prozentsatz als Trainingsdaten verwendet werden.

Der gewünschte Prozentsatz muss *vor* Aktivierung des Auswahlbuttons eingegeben werden, damit er verwendet wird.

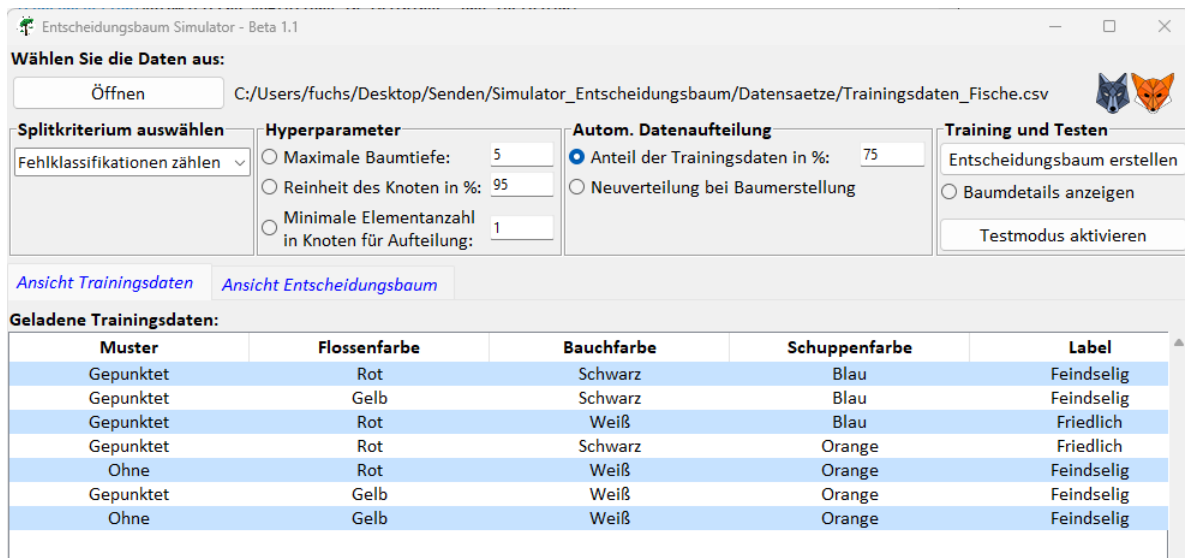


Abbildung 13: Aufteilung des mitgelieferten Datensatzes *Trainingsdaten_Fische.csv* in 75% Trainingsdaten(aufgerundet) und Testdaten (restl. Daten)

Hinweis:

Eine Aktivierung/Deaktivierung der automatischen Datenaufteilung und die damit ggf. verbundene Änderung der Trainingsdaten führt in jedem Fall dazu, dass ein ggf. bereits trainierter Entscheidungsbaum verworfen wird. Der Baum muss also durch Klick auch *Entscheidungsbaum erstellen* in jedem Fall neu trainiert werden.



Option *Neuverteilung bei Baumerstellung*:

Eine Aktivierung des Auswahlbuttons *Neuverteilung bei Baumerstellung* (siehe Abb. 13) führt dazu, dass bei Erstellung eines Entscheidungsbaums. d.h. Klick auf *Entscheidungsbaum erstellen* die Aufteilung in Trainings- und Testdaten neu „ausgewürfelt“ wird. Der eingegebene Prozentsatz bleibt dabei erhalten.

2.1.4 Baumdetails anzeigen

Im Bereich *Training und Testen* kann der Auswahlbutton *Baumdetails anzeigen* aktiviert werden. Sofern ein Entscheidungsbaum trainiert ist, öffnet sich ein Fenster (siehe Abb. 15) in welchem man sich durch Eingabe der Nummer des Knotens, welchen man genauer betrachten möchte und anschließendem Klick auf *Details anzeigen*, Details zu diesem Knoten anzeigen lassen kann. Dadurch ist es möglich Informationen des Knotens, wie den zugehörigen (Teil-) Trainingsdatensatz, ggf. das beste Attribute sowie die Informationsgewinne für die verbleibenden Attribute des zugehörigen (Teil-) Trainingsdatensatzes einzusehen.

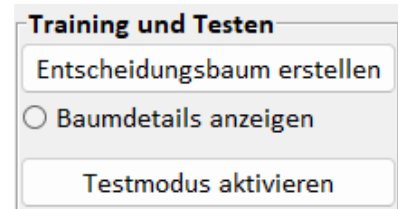


Abbildung 14: Button zur Anzeige der Baumdetails

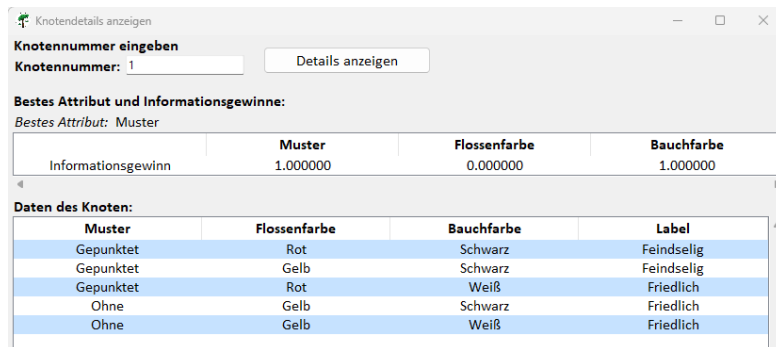


Abbildung 15: Fenster *Knotendetails anzeigen* für Knoten 1 des in Abb. 11 trainierten Baums

2.1.5 Ansicht Trainingsdaten

Im Tab *Ansicht Trainingsdaten* werden die Daten angezeigt, welche momentan als Trainingsdaten dienen. Nach erfolgreichem Laden einer Trainingsdatendatei werden diese automatisch angezeigt.

Bei Verwendung der Funktion *Automatische Datenaufteilung* wird hier der Anteil der geladenen Daten angezeigt, welche als Trainingsdaten verwendet werden.

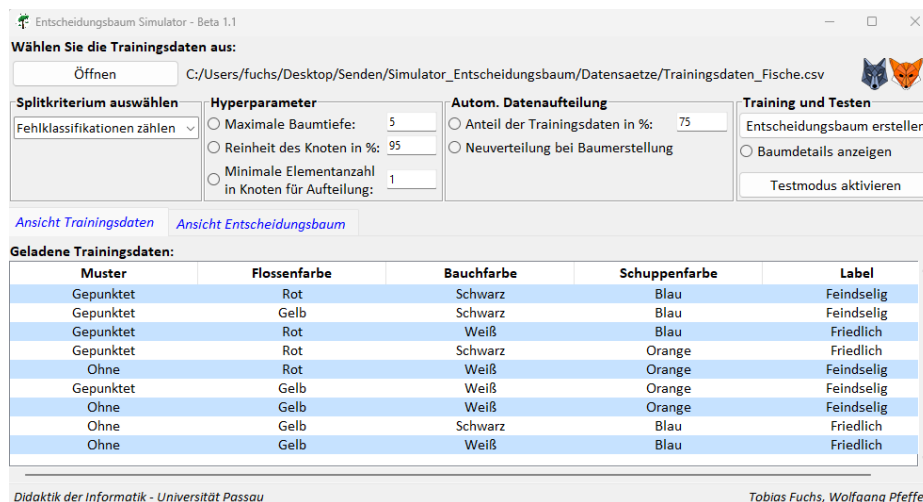


Abbildung 16: *Ansicht Trainingsdaten* mit geladenen Trainingsdatensatz *Trainingsdaten_Fische.csv*



2.1.6 Ansicht Entscheidungsbaum

Im Tab *Ansicht Entscheidungsbaum* wird der durch Klick auf *Entscheidungsbaum erstellen* trainierte Entscheidungsbaum angezeigt. Falls aktuell kein Baum trainiert ist oder ein Fehler beim Training auftrat, wird ein Cartoon-Baum angezeigt.

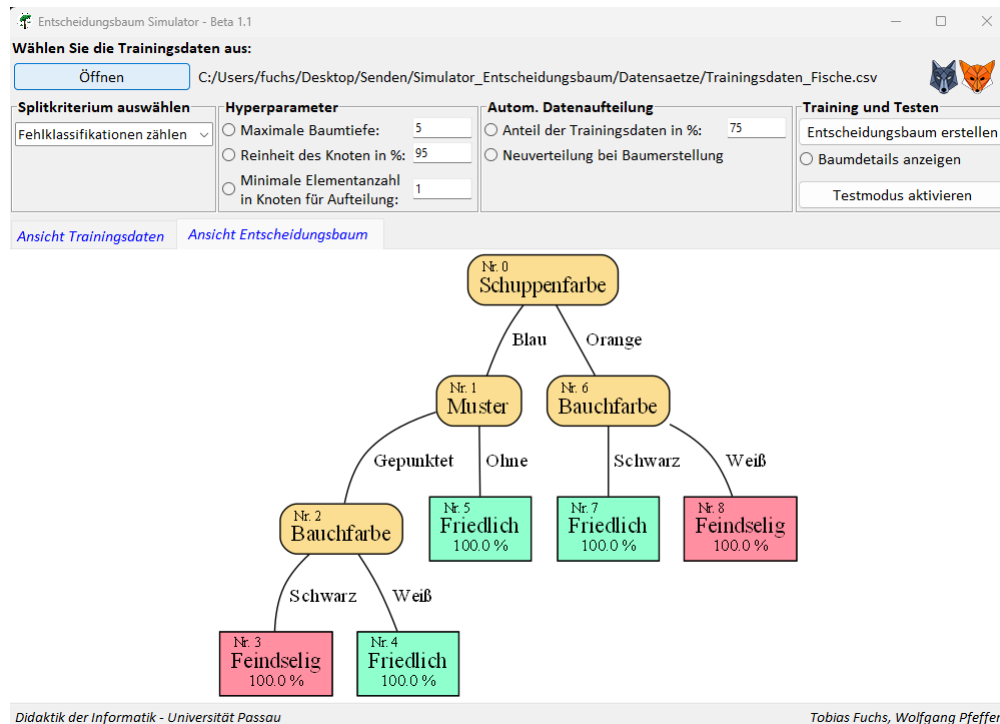


Abbildung 17: *Ansicht Entscheidungsbaum* mit dem trainiertem Entscheidungsbaum

2.2 Testphase

Neben der Trainingsphase (siehe 2.1) lässt sich mit dem *Entscheidungsbaum Simulator* auch die Testphase des maschinellen Lernprozesses simulieren.

2.2.1 Testmodus aktivieren

Um von der Trainingsphase in die Testphase zu wechseln, muss der Testmodus des Simulators durch Klick auf den Button *Testmodus aktivieren* aktiviert werden. Dadurch wird einerseits der Tab *Ansicht Testphase* eingeblendet, welcher die Testphase realisiert. Andererseits werden die für das Training notwendigen Eingabemöglichkeiten gesperrt, da während des Testens das Modell nicht weiter verändert werden soll. Ein weiteres Training ist erst wieder nach Deaktivierung des Testmodus durch Klick auf den Button *Testmodus deaktivieren* (vorher: *Testmodus aktivieren*) möglich.

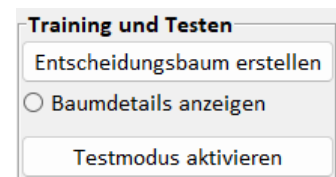


Abbildung 18: Button zur Aktivierung des Testmodus



2.2.2 Ansicht Testphase

Der Tab *Ansicht Testphase* ist für die Realisierung der Testphase zuständig.

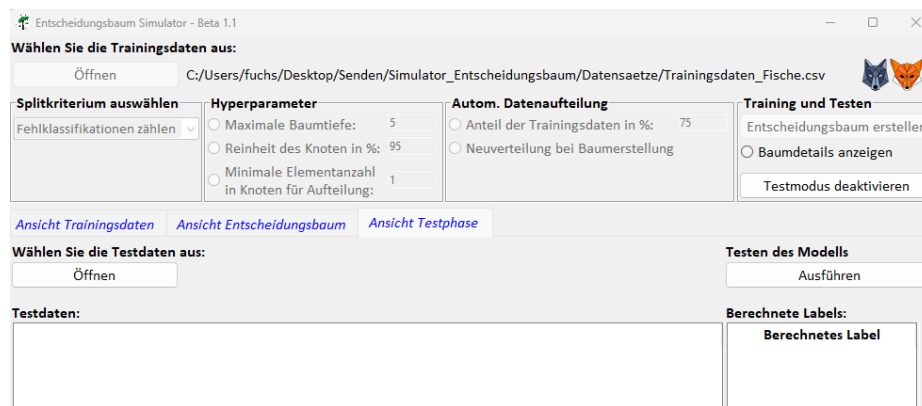


Abbildung 19: Initiale *Ansicht Testphase*

Laden der Testdaten:

Durch Klick auf den Button *Öffnen* im Bereich *Wählen Sie die Testdaten aus:* können Sie die von Ihnen gewünschten Testdaten laden.

Die Testdaten müssen in Form einer *.csv*-Datei vorliegen. In der ersten Zeile stehen die Benennungen der betrachteten Merkmale (hier: *Muster*, *Flossenfarbe*,...) und die Benennung der Labelspalte (hier: *Label*).

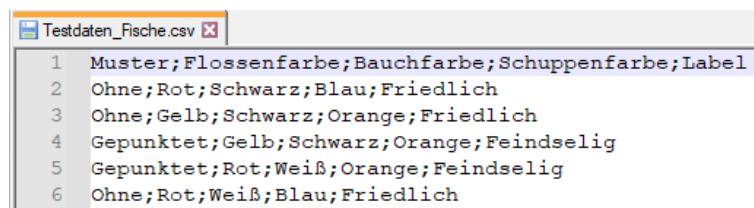


Abbildung 20: Mitgelieferter Testdatensatz *Testdaten_Fische.csv*

In den folgenden Zeilen stehen nun die Werte der einzelnen Testdatenpunkte entsprechend der festgelegten Reihenfolge der Merkmale. Der letzte Eintrag jeder Zeile stellt dabei immer das *erwartete Label* (hier: *Feindselig* oder *Friedlich*) des aktuellen Testdatenpunkts dar.

Beachten Sie bitte, dass die einzelnen Einträge durch ein *;* („Strichpunkt“) voneinander getrennt sein müssen.

Entsprechen die Daten dem notwendigen Format, werden diese automatisch geladen und im Bereich *Testdaten* angezeigt.

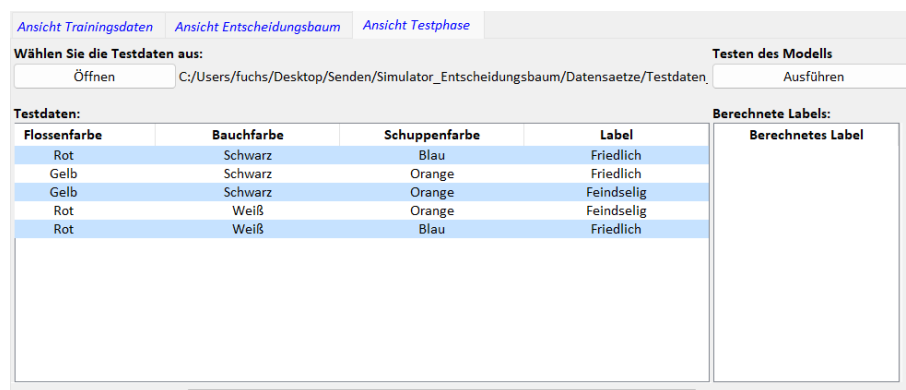


Abbildung 21: *Ansicht Testmodus* mit geladenen Testdaten

Hinweis 1:

Im Ordner *Datensaetze* werden bereits einzelne Beispieldatensätze mitgeliefert. Diese können Sie direkt verwenden oder sich an diesen bei der Erstellung eigener Datensätze orientieren.



Hinweis 2:

Falls Sie in der Trainingsphase die *Automatische Datenaufteilung* in Trainings- und Testdaten aktiviert haben, ist hier die Möglichkeit Testdaten zu laden gesperrt und es werden die Testdaten aus der automatischen Datenaufteilung angezeigt.

Durchführung des Testens:

Durch Klick auf den Button *Ausführen* im Bereich *Testen des Modells* werden die geladenen Testdaten mit dem trainierten Entscheidungsbaum ausgewertet. Die berechneten Labels werden direkt in der Reihenfolge der Testdaten im Bereich *Berechnete Labels* angezeigt.

Stimmt das **berechnete Label** nicht mit dem **erwarteten Label** des Testdatenpunkts überein, wird der entsprechende Eintrag im Bereich *Berechnete Labels* rot hervorgehoben.

Ansicht Trainingsdaten

Ansicht Entscheidungsbaum

Ansicht Testphase

Ansicht Konfusionsmatrix und Gütemaß

Wählen Sie die Testdaten aus:

Öffnen

C:/Users/fuchs/Desktop/Senden/Simulator_Entscheidungsbaum/Datensaeetze/Testdaten

Ausführen

Testen des Modells

Testdaten:

Flossenfarbe	Bauchfarbe	Schuppenfarbe	Label	Berechnetes Label
Rot	Schwarz	Blau	Friedlich	Friedlich
Gelb	Schwarz	Orange	Friedlich	Friedlich
Gelb	Schwarz	Orange	Feindselig	Feindselig
Rot	Weiß	Orange	Feindselig	Feindselig
Rot	Weiß	Blau	Friedlich	Friedlich

Berechnete Labels:

Abbildung 22: *Ansicht Testmodus* nach durchgeführtem Testen der geladenen Testdaten. Der dritte Testdatenpunkt(*rot*) ergab bei der Berechnung nicht das erwartete Label.

Nach erfolgreicher Durchführung des Testvorgangs wird außerdem der Tab *Ansicht Konfusionsmatrix und Gütemaß* eingeblendet.

2.2.3 Ansicht Konfusionsmatrix und Gütemaß

Der Tab *Ansicht Konfusionsmatrix und Gütemaß* dient der Anzeige weiterer Details des Testens. Nach erfolgreichem Ausführen des Testvorgangs nach Klick auf den Button *Ausführen* im Tab *Ansicht Testphase* wird dieser Tab angezeigt. Er enthält einerseits die Berechnung des Gütemaßes **Genauigkeit** des aktuellen Entscheidungsbaums für die betrachteten Testdaten. Die Genauigkeit ergibt sich als Anteil der korrekt klassifizierten Testdatenobjekte an allen Testdatenobjekten. Andererseits werden die Ergebnisse des Testvorgangs in Form einer Konfusionsmatrix dargestellt (siehe Abb. 23).

Ansicht Trainingsdaten

Ansicht Entscheidungsbaum

Ansicht Testphase

Ansicht Konfusionsmatrix und Gütemaß

Gütemaß:

Gütemaß Genauigkeit:

Genauigkeit = Anzahl korrekt klassifizierter Datenobjekte / Anzahl aller Datenobjekte

= 4 / 5

= 0.8 = 80.0%

Konfusionsmatrix:

		Berechnet	
		Feindselig	Friedlich
Erwartet	Feindselig	1	1
	Friedlich	0	3

Abbildung 23: *Ansicht Konfusionsmatrix und Gütemaß* für den Testdatensatz *Testdaten_Fische.csv* für den mit *Trainingsdaten_Fische.csv* trainierten Entscheidungsbaum



Falls (einzelne) Testdatenobjekte des Testdatensatzes nicht mit dem aktuell trainierten Entscheidungsbaum klassifiziert werden können, weil sie Label erwarten, für die der Baum nicht trainiert wurde, wird ein entsprechender Hinweis im Bereich *Gütemaß* angezeigt inklusive der Information, wie viele Testdatenobjekte deshalb nicht klassifiziert werden konnten.

[Ansicht Trainingsdaten](#) [Ansicht Entscheidungsbaum](#) [Ansicht Testphase](#) [Ansicht Konfusionsmatrix und Gütemaß](#)

Wählen Sie die Testdaten aus:
 C:/Users/fuchs/Documents/Git_Hub_Projekte_Uni_Passau/Simulator_Entscheidungsbaum

Testen des Modells

Testdaten:

Flossenfarbe	Bauchfarbe	Schuppenfarbe	Label
Rot	Schwarz	Blau	Aggressiv
Gelb	Schwarz	Orange	Friedlich
Gelb	Schwarz	Orange	Feindselig
Rot	Weiß	Orange	Feindselig
Rot	Weiß	Blau	Aggressiv

Berechnete Labels:

Berechnetes Label
Friedlich
Friedlich
Friedlich
Feindselig
Friedlich

[Ansicht Trainingsdaten](#) [Ansicht Entscheidungsbaum](#) [Ansicht Testphase](#) [Ansicht Konfusionsmatrix und Gütemaß](#)

Gütemaß:
Gütemaß Genauigkeit:
$$\text{Genauigkeit} = \frac{\text{Anzahl korrekt klassifizierter Datenobjekte}}{\text{Anzahl aller Datenobjekte}}$$
$$= \frac{2}{5}$$
$$= 0.4 = 40.0\%$$

Hinweis: 2 Testdatenobjekte konnten nicht zugeordnet werden.
Mögl. Gründe: Nicht zum Training passende Testdaten, nicht trainierte Label/Merkmalwerte,...

Konfusionsmatrix:

	Berechnet		
Erwartet		Feindselig	Friedlich
	Feindselig	1	1
	Friedlich	0	1

Abbildung 24: *Oben:* Zwei Testdaten besitzen das nicht trainierte Label *Aggressiv* *Unten:* Hinweis im Bereich *Gütemaß*, dass zwei Testdaten nicht klassifiziert werden konnten



3. Weitere Informationen

Die Simulationssoftware *Entscheidungsbaum Simulator* ist im Rahmen der Fortbildungsinitiative KI für die Lehrerfortbildung „KI@Informatik 11 – was, wozu, wie, womit unterrichten“ der Didaktik der Informatik der Universität Passau entwickelt worden. Der Simulator dient als ein Hilfsmittel für die Umsetzung des Themas *Entscheidungsbaum* des LehrplanPLUS-Kapitels 11.4 Künstliche Intelligenz des bayerischen Lehrplans.

Die aktuellste Version des *Entscheidungsbaum Simulators* sowie dieses Handbuchs finden Sie unter https://github.com/D3rFuch5/Simulator_Entscheidungsbaum.git sowie auf der Seite der Didaktik der Informatik der Universität Passau unter <https://www.ddi.fim.uni-passau.de/ki-gym>

Der Simulator wurde in der Programmiersprache Python entwickelt und verwendet neben der Python-Bibliothek (Version 3.11.3) das Paket Graphviz (Version 0.20.1) und die Installation der Graphviz-Software.

Die Simulationssoftware *Entscheidungsbaum Simulator* an sich und dieses zugehörige Handbuch sind unter der *APACHE LICENSE, VERSION 2.0* veröffentlicht. Die Python-Bibliothek steht unter PSF-Lizenz, das Paket Graphviz unter MIT-Lizenz. Weitere von diesen Paketen verwendete Pakete/Bibliotheken können unter anderer Lizenz stehen und in diesen Fällen gelten diese Lizenzen.

- APACHE LICENSE, VERSION 2.0:

Link: <https://www.apache.org/licenses/LICENSE-2.0>

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

“License“ shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor“ shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity“ shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “control“ means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You“ (or “Your“) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source“ form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object“ form shall mean any form resulting from mechanical transformation or translation of a Source



form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work“ shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works“ shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution“ shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted“ means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.“

“Contributor“ shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and



- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

(5.) Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

(6.) Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

(7.) Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

(8.) Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,



or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

(8.) Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

(9.) Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets “{}“ replaced with your own identifying information. (Don’t include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same “printed page“ as the copyright notice for easier identification within third-party archives.

Copyright yyyy name of copyright owner

Licensed under the Apache License, Version 2.0 (the “License“); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS“ BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

- PSF LICENSE AGREEMENT FOR PYTHON 3.11.3 (*Stand: 07.04.2023*):

Link: <https://docs.python.org/3/license.html#psf-license>



1. This LICENSE AGREEMENT is between the Python Software Foundation (“PSF”), and the Individual or Organization (“Licensee”) accessing and otherwise using Python 3.11.3 software in source or binary form and its associated documentation.
 2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 3.11.3 alone or in any derivative version, provided, however, that PSF’s License Agreement and PSF’s notice of copyright, i.e., “Copyright © 2001-2023 Python Software Foundation; All Rights Reserved“ are retained in Python 3.11.3 alone or in any derivative version prepared by Licensee.
 3. In the event Licensee prepares a derivative work that is based on or incorporates Python 3.11.3 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 3.11.3.
 4. PSF is making Python 3.11.3 available to Licensee on an “AS IS“ basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 3.11.3 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
 5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 3.11.3 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 3.11.3, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
 6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
 7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
 8. By copying, installing or otherwise using Python 3.11.3, Licensee agrees to be bound by the terms and conditions of this License Agreement.
- Paket Graphviz (*Stand: 13.04.2023*):
Link: <https://github.com/xflr6/graphviz/blob/master/LICENSE.txt>

The MIT License (MIT)

Copyright (c) 2013-2022 Sebastian Bank

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following



conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.