1 Einleitung

Die Programmiersprache Python ist eine einfach zu erlernende Sprache, die es dem Anwender erlaubt, komplexe Programme für verschiedene Anwendungsgebiete zu entwickeln (Theis, 2017, S.17).

Die Grundlage dieser Hausarbeit bilden drei CSV-Dateien, die vier Trainingsfunktionen („train.csv“), 50 Idealfunktionen („ideal.csv“) sowie 100 Testpunkte („test.csv“).

Ziel dieser Hausarbeit ist es, ein Python Programm zu entwickeln, das die drei CSV-Dateien importiert, den Inhalt verarbeitet und abschließend die Daten in eine SQLite-Datenbank speichert.

Die folgenden zwei Kapitel geben die Aufgabenstellung, unterteilt in zwei Aufgabenteile, wieder. Im Hauptteil werden die verwendeten Bibliotheken vorgestellt sowie die Struktur des Programms erklärt. Im Schlussteil wird ein Resümee gezogen, wobei auch darauf eingegangen wird, welche Vor- sowie Nachteile das erzeugte Python Programm besitzt.

* 1. Aufgabenteil 1

In einem ersten Abschnitt soll zu den vier Trainingsfunktionen jeweils eine bestmögliche Idealfunktion ermittelt werden. Eine Idealfunktion ist dann die bestmögliche Passung, wenn die Summe aller quadratischen y-Abweichungen minimal ist. Dieses Kriterium kann mit folgender Formel (Least-Square) berechnet werden:

= Y-Wert der Trainingsfunktion

= Y-Wert der Idealfunktion

Sind die besten Passungen aus Trainings- und Idealfunktion bestimmt, sollen die jeweiligen Ergebnisse logisch visualisiert werden.

2.2 Aufgabenteil 2

Die vier bestmöglichen Idealfunktionen aus Aufgabenteil 1 werden nun zur Analyse der 100 Testpunkte verwendet. Es wird geprüft, ob die Testpunkte einer oder mehreren Idealfunktionen zugeordnet werden können. Um zu entscheiden, ob ein Testpunkt zu einer Idealfunktion passt, wird zuerst die Abweichung zwischen dem Y-Wert des Testpunktes und dem Y-Wert der Idealfunktion berechnet. Ist die Abweichung kleiner als das Testkriterium (maximale Y-Abweichung zwischen Ideal – und Trainingsfunktion, multipliziert mit ), gilt der Testpunkt als anpassbar. Auch die Ergebnisse aus Aufgabenteil 2 sollen logisch visualisiert werden.

2. Python-Bibliotheken und -Module

Dieses Kapitel beschreibt die in dieser Hausarbeit verwendeten Python-Bibliotheken sowie Python-Module und erläutert, welche Abläufe im Programm sie vereinfachen.

2.1 Pandas-Bibliothek

Die Pandas-Bibliothek ist spezialisiert für das umfangreiche Arbeiten mit strukturierten oder tabellarischen Daten. Sie eignet sich besonders gut für die Manipulation, Vorbereitung und Bereinigung von Daten (McKinney, 2023, S.23).

Für die Bearbeitung der Aufgabe wird das Pandas-Dataframe inklusive der anwendbaren Funktionen verwendet. Ein Pandas-Dataframe ist eine tabellenartige, geordnete Datenstruktur, wobei jede Zeile sowie Spalte einen Index besitzt (McKinney, 2023, S.148).

In dieser Hausarbeit wird die Pandas-Bibliothek verwendet, um die Daten aus den CSV-Dateien zu importieren und zu speichern. Mit Hilfe der zur Pandas-Bibliothek gehörenden Methoden werden die erzeugten Pandas-Dataframes bearbeitet und so manipuliert, dass sie die Weiterverarbeitung zum Erreichen des Ziels der Aufgabenstellung ermöglichen.

2.2 Matplotlib-Bibliothek

Matplotlib ist eine der populärsten Bibliotheken zur Visualisierung von Daten in Python. Sie ermöglicht es dem Anwender, verschiedene Arten von Diagrammen und Grafiken zu erstellen. Hierzu zählen unter anderem Scatterplots, Liniendiagramme und Balkendiagramme (Matthes, 2019, S.306). Die Matplotlib-Bibliothek besitzt die Fähigkeit, mit einer Vielzahl von Betriebssystemen und grafischen Ausgabegeräten zu interagieren (VanderPlas, 2018, S.245).

Alle in dieser Hausarbeit erstellten Grafiken zur Visualisierung der Ergebnisse werden auf Basis der Matplotlib-Bibliothek erstellt. Für Aufgabenteil 1 werden ausschließlich Liniendiagramme erstellt. Die Ergebnisse aus Aufgabenteil 2 werden in einer Grafik visualisiert, die eine Liniendiagramm und einen Scatterplot kombiniert.

Für eine bessere Lesbarkeit der Diagramme/Grafiken werden Titel, Legende und Achsenbeschriftung hinzugefügt (Matthes, 2019, S.308).

2.3 SQL-Alchemy-Bibliothek

Die SQL-Alchemy-Bibliothek dient als Verbindungselement zwischen dem Python-Code und einer Datenbank. SQL-Alchemy konvertiert Python-Funktionsaufrufe in die entsprechenden SQL-Anweisungen. Es werden mehrere Datenbanken, bspw. MySQL oder SQLite, unterstützt (Campesato, 2023, S.253).

In dieser Hausarbeit sollen die Ergebnisse in einer SQLite-Datenbank gespeichert werden. SQLite ist eine prozessinterne Bibliothek, die eine in sich geschlossene, serverlose, konfigurationsfreie SQL-Datenbank-Engine beinhaltet. Der Code für SQLite ist öffentlich zugänglich, wobei die SQLite-Datenbank die am weitesten verbreitete Datenbank ist (Campesato, 2023, S.260+261).

Mit Hilfe der SQL-Alchemy-Bibliothek wird im Python-Programm die Verbindung zur SQLite-Datenbank hergestellt, eine Tabelle angelegt und die Daten in die Datenbank überschrieben. Das Ergebnis ist abschließend jeweils eine lokale SQLite-Datei für den Trainings-, Ideal- und Testdatensatz.

2.5 UnitTest-Bibliothek

Das Testen des Python-Programms ist ein wichtiger Schritt bei der Entwicklung eines Python-Codes. Es wird sichergestellt, dass die Funktionalität des Programms wie erwartet gegeben ist. Fehler können aufgedeckt und frühzeitig behoben werden. Eine Möglichkeit, das Python-Programm wie beschrieben zu testen, bietet die UnitTest-Bibliothek. Sie ist einer der beliebtesten Test-Bibliotheken und gehört zu den Standardbibliotheken von Python (Kapil, 2019, S.238-240).

Mit der UnitTest-Bibliothek wird im vorliegenden Python-Programm die Funktionalität der Module der einzelnen Klassen überprüft. Es wird geprüft, ob Berechnung korrekt durchgeführt werden und ob Argumente der Klassen richtig zurückgegeben werden. Außerdem wird das Anlegen von Instanzen der erzeugten Klassen überprüft.

2.4 Tkinter-Modul

Das Tkinter-Modul ist eine Schnittstelle zur Tk-Bibliothek, die zu den Standardbibliotheken von Python gehört. Durch das Tkinter-Modul ist es möglich, grafische Oberflächen für kleinere Anwendungen zu erstellen (Theis, 2017, S.371).

In dem Python-Programm wird das Modul „filedialog“ des Pakets Tkinter verwendet, das vorgefertigte Dateidialoge bereitstellt. Die Dateidialogen fordern den Nutzer des Python-Programms dazu auf, Dateien oder Ordner auszuwählen, um diese im Python-Programm hineinzuladen (Ernesti, 2020, S.869).

Das Modul filedialog kommt in dieser Hausarbeit jedoch nur dann zum Einsatz, wenn die als Default eingebetteten Dateipfade nicht existieren. Um einen Abbruch des Programms zu verhindern, müssen schließlich die drei CSV-Dateien manuell ausgewählt werden.

2.3 OS-Modul

Das os-Modul ermöglicht es, betriebssystemabhängige Funktionalitäten zu nutzen (OS-DOKU). Im erstellten Python-Programm werden mit Hilfe des os-Moduls die Datei-Pfade der zu importierenden CSV-Dateien überprüft. Existieren die als Default implementierten Dateipfade nicht, wird der Benutzer aufgefordert, die drei CSV-Dateien manuell auszuwählen.

2.4 Math-Modul

Durch das math-Modul erlangt der Anwender Zugang zu mathematischen Funktionen. Hierzu zählen unter anderem die trigonometrischen Funktionen (Theis, 2017, S.89-91). Für die Berechnung des Testkriteriums in Aufgabenteil 2 wird die Berechnung der Quadratwurzel benötigt, die ebenfalls eine Funktion des math-Moduls ist.

3 Struktur des Python-Programms

Dieses Kapitel behandelt die Struktur des entwickelten Python-Programms. Die Aufgabenstellung gibt vor, dass eine objektorientierte Programmierung angewendet werden soll. Außerdem sollen sowohl Standard- als auch user-definierte Exception Handlings sinnvoll verwendet werden und Unit-Tests, wo immer es sich anbietet, implementiert werden. In den folgenden Unterkapiteln werden die genannten Anforderungen vertieft und die Umsetzung im Python-Code erläutert.

3.1 Objektorientierte Programmierung

Unter einer objektorientierten Programmierung ist zu verstehen, dass alle Elemente, mit denen ein Python-Programm erstellt wird, eine Instanz einer Klasse ist. Dies betrifft einzelne Zahlen genauso wie Listen, Dictionaries usw. (Ernesti, 2020, S.381).

Grundlage der objektorientierten Programmierung sind sogenannte Klassen, die mit Eigenschaften (Attribute) und Methoden (Funktionen) ausgestattet werden. Es können nun im Programm-Code mehrere Objekte (Instanzen) dieser Klassen erzeugt werden und die entsprechenden Methoden auf die Objekte angewendet werden. Weiterhin besteht die Möglichkeit, dass einzelne Klassen von einer anderen Klasse erben. Es wird eine Basisklasse (Eltern-Klasse) definiert, die ihre Eigenschaften und Methoden an die abgeleitete Klasse (Kind-Klasse) vererbt. Ähnliche Objekte mit identischen Eigenschaften und Methoden können so vereinfacht erzeugt werden (Theis, 2017, S.203).

Python ermöglicht es dem Anwender, bei der Definition der Attribute und Methoden einer Klasse aus drei Sichtbarkeitstufen zu wählen. Auf Attribute und Methoden, die als public deklariert werden, kann von außerhalb der Klasse ohne Einschränkungen zugegriffen werden. Ähnlich verhält es sich bei der Sichtbarkeitsstufe protected, die häufig in Vererbungshierachien Anwendung findet. Prinzipiell ist der Zugriff auf die Attribute und Methoden von außen uneingeschränkt möglich. Per Konvention wird jedoch festgelegt, dass die Sichtbarkeitsstufe protected behandelt wird wie die Sichtbarkeitsstufe privat. Hierbei ist kein Zugriff auf Attribute und Methoden von außen möglich (Steyer, 2018, S.161).

In dieser Hausarbeit werden zur Bearbeitung der Aufgabenstellung drei Basisklassen erzeugt, wobei eine Basisklasse drei abgeleitete Klassen besitzt. Zusätzlich sind in dem Python-Programm vier Klassen zum Abfangen von Fehlern (Exception Handling) implementiert. Im Folgenden werden die drei zur Lösung der Aufgabenstellung verwendeten Klassen sowie ihre Funktionalitäten erläutert.

Die erstellten Klassen des Python-Programms besitzen Attribute der Sichtbarkeitsstufe privat oder protected. Um auf diese Attribute zugreifen zu können, gibt es für jedes Attribut eine Getter- und Setter-Methode (Steyer, 2018, S.162+163). Diese Methoden beginnen im Python-Code jeweils mit hole (get) und setze (set). Um die folgenden Grafiken nicht zu überladen, werden sie nicht aufgeführt.

3.1.1 Klasse Datensatz

Die Klasse Datensatz bildet die Grundlage für die Verarbeitung der Daten aus den CSV-Dateien. Abbildung 1 zeigt die Attribute und Methoden der Klasse Datensatz.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Die Grafik zeigt, dass die Klasse Datensatz nur ein Attribut besitzt (dataframe). Bei der Erstellung der Klasse muss jedoch nicht zwingend eine Variable für das Attribut übergeben werden. In diesem Fall bekommt das Attribut dataframe den Wert einer leeren Zeichenkette. Somit kann eine Instanz der Klasse Datensatz entweder mit einer leeren Zeichenkette oder einem Pandas-Dataframe (siehe Kapitel \ref{sec:Pandas}) erzeugt werden. Wird einer Instanz der Klasse Datensatz bei der Erzeugung kein Pandas-Dataframe zugewiesen, bietet die Methode „import\_dataframe()“ die Möglichkeit, das Attribut dataframe durch den Import einer CSV-Datei zu belegen. Hierbei wird aus den Daten der CSV-Datei ein Pandas-Dataframe erzeugt und gespeichert. Die im Folgenden beschriebenen Methoden können jeweils nur ausgeführt werden, wenn dem Attribut dataframe ein Pandas-Dataframe zugewiesen ist. Daher wird vor jeder Ausführung der Methoden unter Verwendung eines Exception Handlings überprüft, ob es sich um ein Pandas-Dataframe oder eine leere Zeichenkette handelt.

Bei dem Großteil der in Tabelle ?? beschriebenen Methoden der Klasse Datensatz wird lediglich auf Funktionen des Datentyps Pandas-Dataframe zurückgegriffen. Die Klasse Datensatz dient nicht dazu, Berechnungen, Vergleiche oder Ähnliches durchzuführen. Sie soll das Arbeiten mit einem Pandas-Dataframe erleichtern und anwenderfreundlicher gestalten. Die Bezeichnungen der Methoden sind auf die zu verarbeitenden Daten dieser Hausarbeit abgestimmt. Somit kann es bei der Anwendung der Klasse Datensatz auf abweichende Datenstrukturen zu Verständnisproblemen kommen. Es ist z.B. zwingend notwendig, dass das Pandas-Dataframe in der ersten Spalte die X-Werte der Funktionen beinhaltet, um bei der Anwendung der Methode hole\_x\_spalte() ein korrekten Rückgabewert zu erhalten.

Zum einen wird die Klasse Datensatz verwendet, um die Daten aus den gegebenen CSV-Dateien zu importieren. Durch die implementierten Methoden werden die Daten gespeichert und für weitere Berechnungen aufbereitet.

Zum anderen werden die Funktionalitäten des Pandas-Dataframes, und damit auch der Klasse Datensatz, genutzt, um die Daten mit geringem Aufwand in eine SQLite-Datenbank zu exportieren.

3.1.2 Klasse Funktion

Die Klasse Funktion ist eine Elternklasse und besitzt zwei Kindklassen. Mit Hilfe dieser Klassen werden die zur Lösung der Aufgabenstellung benötigten Berechnungen durchgeführt. Abbildung X zeigt den Aufbau der Klasse Funktion inklusive der abgeleiteten Klassen.

Die Elternklasse Funktion besitzt nur ein Attribut (funktion). Jedoch ist es bei der Erstellung einer Instanz der Klasse Funktion zwingend erforderlich, ein Pandas-Dataframe zu übergeben. Wird kein Pandas-Dataframe übergeben, wird das Programm mit Hilfe des Exception Handlings abgebrochen. Die Klasse Funktion verarbeitet nur die ersten beiden Spalten des Pandas-Dataframes, da hier per Definition die X- sowie Y-Werte der Funktion zu finden sind. Tabelle X zeigt die zur Klasse Funktion gehörenden Methoden und beschreibt ihre Funktionalität.

Ähnlich wie bei der Klasse Datensatz (siehe Kapitel) ist die Grundlage der Klasse Funktion ein Pandas-Dataframe (siehe Grafik sowieso). Die Funktionalität aller in Tabelle gezeigten Methoden basiert auf Funktionen eines Pandas-Dataframes. Auch in diesem Fall stellen die Bezeichnung der Methoden der Klasse Funktion eine Beziehung zu der Datenstruktur der vorgegebenen CSV-Dateien her.

Die Klasse Trainingsfunktion beinhaltet mit der Methode suche\\_idealfunktion(datensatz) die eigentliche Funktion zur Bearbeitung von Aufgabenteil 1 (siehe Kapitel \ref{sec:Aufgabenteil1}). Einer Trainingsfunktion wird bei der Anwendung dieser Methode ein Datensatz, also ein Pandas-Dataframe, zur weiteren Verarbeitung zugewiesen. Im Regelfall ist dies der Datensatz der Idealfunktionen. Es wird über alle Funktionen des Pandas-Dataframes der Idealfunktionen iteriert. Pro Spalte wird eine Vergleichsfunktion angelegt, die eine Instanz der Elternklasse Funktion ist. Sie dient lediglich als Vergleichsobjekt. Hierzu werden Methoden der Klasse Funktion, bspw. hole\\_y\\_werte(), benötigt. Nun wird über die Y-Werte der Vergleichsfunktion iteriert und jeweils die Y-Abweichung zum entsprechenden Y-Wert der Trainingsfunktion berechnet. Die Y-Abweichungen werden quadriert und summiert, um den Least-Square-Wert zu berechnen (siehe Formel \ref{eq:leastsquare}). Die Vergleichsfunktion mit dem geringsten Least-Square-Wert wird schließlich als Instanz der Klasse Idealfunktion unter dem Attribut ideal\\_funktion gespeichert. Zusätzlich wird zur Anwendung des Testkriteriums für Aufgabenteil 2 (siehe Kapitel \ref{sec:Aufgabenteil2}) die maximale Abweichung zwischen einem Y-Wert der Trainingsfunktion und dem entsprechenden Y-Wert der Idealfunktion bestimmt.

Die Klasse Idealfunktion beinhaltet neben dem Attribut der Elternklasse (funktion) vier weitere Attribute. Instanzen der Klasse Idealfunktion dienen nur als Speicherort der bestimmten Funktionen zur entsprechenden Trainingsfunktion. Bei der Erzeugung einer Instanz der Klasse Idealfunktion müssen neben einem Pandas-Dataframe der Idealfunktion zusätzlich folgende Attribute zugewiesen werden (siehe Grafik xxx):

* Least-Square-Wert
* maximale Y-Abweichung
* Liste aller Y-Abweichungen zur Trainingsfunktion
* Trainingsfunktion

Die Klasse Idealfunktion besitzt lediglich weitere Methoden zum Setzen bzw. Ausgeben der Attribute.

Klasse Punkt

Die Klasse Punkt ist definiert durch einen X- und Y-Wert. Mit Hilfe der Klasse Punkt werden die XY-Paare der einzelnen Funktion verarbeitet und entsprechende Berechnungen durchgeführt. Abbildung xx zeigt die Attribute und Methoden der Klasse Punkt.

**\subsection{Klasse Datensatz}**

**\label{sub:Datensatz}**

Die Klasse Datensatz bildet die Grundlage für die Verarbeitung der Daten aus den CSV-Dateien. Sie ist zudem eine Elternklasse der Klasse Funktion, die in Kapitel XX erläutert wird. \abbildung{fig:Datensatz} zeigt die Attribute und Methoden der Klasse Datensatz sowie die entsprechenden Kindklassen.

\begin{figure}[htbp]

\centering

**\includegraphics**[scale=0.7]{Grafik\_Klasse\_Datensatz}

\caption{Attribute und Methoden der Klasse Datensatz}

**\label{fig:Datensatz}**

\end{figure}

Die Grafik zeigt, dass die Klasse Datensatz nur ein Attribut besitzt (dataframe). Bei der Erstellung der Klasse muss jedoch nicht zwingend eine Variable für das Attribut übergeben werden. In diesem Fall bekommt das Attribut dataframe den Wert einer leeren Zeichenkette. Somit kann eine Instanz der Klasse Datensatz entweder mit einer leeren Zeichenkette oder einem Pandas-Dataframe (siehe Kapitel \ref{sec:Pandas}) erzeugt werden. Wird einer Instanz der Klasse Datensatz bei der Erzeugung kein Pandas-Dataframe zugewiesen, bietet die Methode import\\_dataframe() die Möglichkeit, das Attribut dataframe durch den Import einer CSV-Datei zu belegen. Hierbei wird aus den Daten der CSV-Datei ein Pandas-Dataframe erzeugt und gespeichert. Die in der folgenden Tabelle \ref{tab:Klasse\_Datensatz} beschriebenen Methoden können jeweils nur ausgeführt werden, wenn dem Attribut dataframe ein Pandas-Dataframe zugewiesen ist. Daher wird vor jeder Ausführung der Methoden unter Verwendung eines Exception Handlings überprüft, ob es sich um ein Pandas-Dataframe oder eine leere Zeichenkette handelt.

\begin{table}[htbp]

\centering

**\includegraphics**[width=0.8\textwidth]{Tabelle\_Datensatz}

\caption{Methoden der Klasse Datensatz}

**\label{tab:Klasse\_Datensatz}**

\end{table}

Bei dem Großteil der in Tabelle \ref{tab:Klasse\_Datensatz} beschriebenen Methoden der Klasse Datensatz wird lediglich auf Funktionen des Datentyps Pandas-Dataframe zurückgegriffen. Die Klasse Datensatz dient nicht dazu, Berechnungen, Vergleiche oder Ähnliches durchzuführen. Sie soll das Arbeiten mit einem Pandas-Dataframe erleichtern und anwenderfreundlicher gestalten. Die Bezeichnungen der Methoden sind auf die zu verarbeitenden Daten dieser Hausarbeit abgestimmt. Somit kann es bei der Anwendung der Klasse Datensatz auf abweichende Datenstrukturen zu Verständnisproblemen kommen. Es ist z.B. zwingend notwendig, dass das Pandas-Dataframe in der ersten Spalte die X-Werte der Funktionen beinhaltet, um bei der Anwendung der Methode hole\\_x\\_spalte() ein korrekten Rückgabewert zu erhalten.\\

Zum einen wird die Klasse Datensatz verwendet, um die Daten aus den gegebenen CSV-Dateien zu importieren. Durch die implementierten Methoden werden die Daten gespeichert und für weitere Berechnungen aufbereitet. \\

Zum anderen werden die Funktionalitäten des Pandas-Dataframes, und damit auch der Klasse Datensatz, genutzt, um die Daten mit geringem Aufwand in eine SQLite-Datenbank zu exportieren.

**\subsection{Klasse Funktion}**

Die Klasse Funktion ist eine Elternklasse und besitzt zwei Kindklassen. Mit Hilfe dieser Klassen werden die zur Lösung der Aufgabenstellung benötigten Berechnungen durchgeführt. \abbildung{fig:Funktion} zeigt den Aufbau der Klasse Funktion inklusive der abgeleiteten Klassen.

\begin{figure}[htbp]

\centering

**\includegraphics**[scale=0.7]{Grafik\_Klasse\_Funktion}

\caption{Attribute und Methoden der Klasse Funktion}

**\label{fig:Funktion}**

\end{figure}

Die Elternklasse Funktion besitzt nur ein Attribut (funktion). Jedoch ist es bei der Erstellung einer Instanz der Klasse Funktion zwingend erforderlich, ein Pandas-Dataframe zu übergeben. Wird kein Pandas-Dataframe übergeben, wird das Programm mit Hilfe des Exception Handlings abgebrochen. Die Klasse Funktion verarbeitet nur die ersten beiden Spalten des Pandas-Dataframes, da hier per Definition die X- sowie Y-Werte der Funktion zu finden sind. Tabelle \ref{tab:Klasse\_Funktion} zeigt die zur Klasse Funktion gehörenden Methoden und beschreibt ihre Funktionalität.

\begin{table}[htbp]

\centering

**\includegraphics**[width=0.8\textwidth]{Tabelle\_Funktion}

\caption{Methoden der Klasse Funktion}

**\label{tab:Klasse\_Funktion}**

\end{table}

Ähnlich wie bei der Klasse Datensatz (siehe Kapitel \ref{sub:Datensatz}) ist die Grundlage der Klasse Funktion ein Pandas-Dataframe (siehe \abbildung{fig:Funktion}). Die Funktionalität aller in Tabelle \ref{tab:Klasse\_Funktion} gezeigten Methoden basiert auf Funktionen eines Pandas-Dataframes. Auch in diesem Fall stellen die Bezeichnungen der Methoden der Klasse Funktion eine Beziehung zu der Datenstruktur der vorgegebenen CSV-Dateien her. \vspace{6pt}\\

\vspace{6pt}\textbf{Klasse Trainingsfunktion}\\

Die Klasse Trainingsfunktion beinhaltet mit der Methode suche\\_idealfunktion(datensatz) die eigentliche Funktion zur Bearbeitung von Aufgabenteil 1 (siehe Kapitel \ref{sec:Aufgabenteil1}). Einer Trainingsfunktion wird bei der Anwendung dieser Methode ein Datensatz, also ein Pandas-Dataframe, zur weiteren Verarbeitung zugewiesen. Im Regelfall ist dies der Datensatz der Idealfunktionen. Es wird über alle Funktionen des Pandas-Dataframes der Idealfunktionen iteriert. Pro Spalte wird eine Vergleichsfunktion angelegt, die eine Instanz der Elternklasse Funktion ist. Sie dient lediglich als Vergleichsobjekt. Hierzu werden Methoden der Klasse Funktion, bspw. hole\\_y\\_werte(), benötigt. Nun wird über die Y-Werte der Vergleichsfunktion iteriert und jeweils die Y-Abweichung zum entsprechenden Y-Wert der Trainingsfunktion berechnet. Die Y-Abweichungen werden quadriert und summiert, um den Least-Square-Wert zu berechnen (siehe Formel \ref{eq:leastsquare}). Die Vergleichsfunktion mit dem geringsten Least-Square-Wert wird schließlich als Instanz der Klasse Idealfunktion unter dem Attribut ideal\\_funktion gespeichert. Zusätzlich wird zur Anwendung des Testkriteriums für Aufgabenteil 2 (siehe Kapitel \ref{sec:Aufgabenteil2}) die maximale Abweichung zwischen einem Y-Wert der Trainingsfunktion und dem entsprechenden Y-Wert der Idealfunktion bestimmt.\vspace{6pt}\\

\vspace{6pt}\textbf{Klasse Trainingsfunktion}\\

Instanzen der Klasse Idealfunktion dienen nur als Speicherort der bestimmten Funktionen zur entsprechenden Trainingsfunktion. Die Klasse Idealfunktion beinhaltet neben dem Attribut der Elternklasse (funktion) vier weitere Attribute.Bei der Erzeugung einer Instanz der Klasse Idealfunktion müssen neben einem Pandas-Dataframe der Idealfunktion zusätzlich folgende Attribute zugewiesen werden (siehe \abbildung{fig:Funktion}):

\setlist{noitemsep}

\begin{itemize}

\item Least-Square-Wert

\item maximale Y-Abweichung

\item Liste alle Y-Abweichungen zur Trainingsfunktion

\item Trainingsfunktion

\end{itemize}

Die Klasse Idealfunktion besitzt lediglich weitere Methoden zum Setzen bzw. Ausgeben der Attribute.

**\subsection{Klasse Punkt}**

Die Klasse Punkt ist definiert durch einen X- und Y-Wert. Mit Hilfe der Klasse Punkt werden die XY-Paare der einzelnen Funktion verarbeitet und entsprechende Berechnungen durchgeführt. \abbildung{fig:Punkt} zeigt die Attribute und Methoden der Klasse Punkt.

\begin{figure}[htbp]

\centering

**\includegraphics**[scale=0.7]{Grafik\_Klasse\_Punkt}

\caption{Attribute und Methoden der Klasse Punkt}

**\label{fig:Punkt}**

\end{figure}