

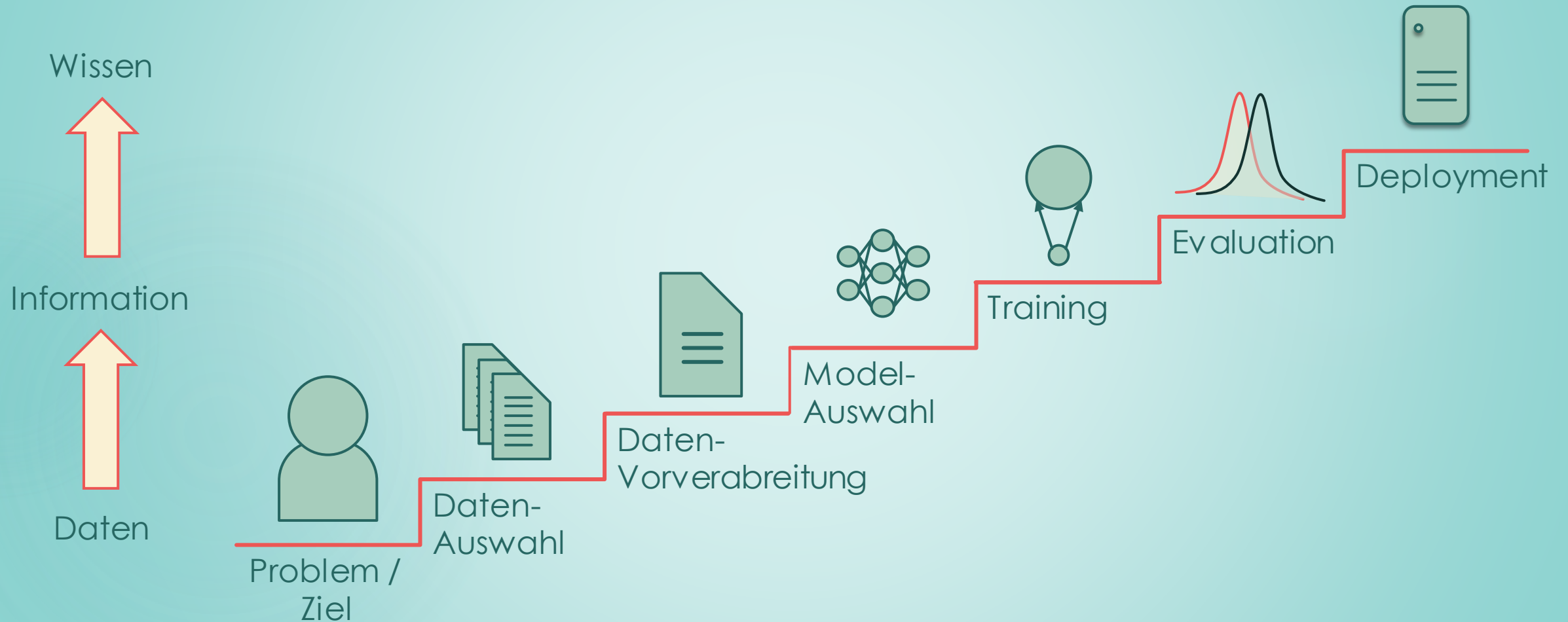
Anwendungen KI/ML

ROBERT-
ALEXANDER
WINDBERGER

Projekt 1: Netflix Recommendation System – Datenvorverarbeitung und -erkundung

ANWENDUNGEN
KI/ML

Der Weg



1. Data Reader

1. Erstellen Sie in Python eine Klasse `DataReader`. Verwenden Sie dazu die Pandas-Bibliothek.
 1. Implementieren Sie eine Methode `read_netflix_data(file_path: str)`, die den rohen `pandas.DataFrame` zurückgibt.
 2. Implementieren Sie eine Methode `preprocess_netflix_data(data: pandas.DataFrame)`, welche den `pandas.DataFrame` aus `read_netflix_data` vorverarbeitet (siehe Folie).
 3. Implementieren Sie eine Methode `write_netflix_data(data_preprocessed: pandas.DataFrame)`, die den vorverarbeiteten Netflix Datensatz in eine Pickle-Datei speichert

1.1 read_netflix_data (0,5 Pkt.)

► pandas.DataFrame nach einlesen:

	index	id	title	type	release_year	age_certification	runtime	genres	production_countries	seasons	imdb_id	imdb_score	imdb_votes
0	0	ts300399	Five Came Back: The Reference Films	SHOW	1945	TV-MA	48	['documentation']	['US']	1.0	NaN	NaN	NaN
1	1	tm84618	Taxi Driver	MOVIE	1976	R	113	['crime', 'drama']	['US']	NaN	tt0075314	8.3	795222.0
2	2	tm127384	Monty Python and the Holy Grail	MOVIE	1975	PG	91	['comedy', 'fantasy']	['GB']	NaN	tt0071853	8.2	530877.0
3	3	tm70993	Life of Brian	MOVIE	1979	R	94	['comedy']	['GB']	NaN	tt0079470	8.0	392419.0
4	4	tm190788	The Exorcist	MOVIE	1973	R	133	['horror']	['US']	NaN	tt0070047	8.1	391942.0
...
5801	5801	tm1014599	Fine Wine	MOVIE	2021	NaN	100	['romance', 'drama']	['NG']	NaN	tt13857480	6.9	39.0
5802	5802	tm1108171	Edis Starlight	MOVIE	2021	NaN	74	['music', 'documentation']	[]	NaN	NaN	NaN	NaN
5803	5803	tm1045018	Clash	MOVIE	2021	NaN	88	['family', 'drama']	['NG', 'CA']	NaN	tt14620732	6.5	32.0
5804	5804	tm1098060	Shadow Parties	MOVIE	2021	NaN	116	['action', 'thriller']	[]	NaN	tt10168094	6.2	9.0
5805	5805	ts271048	Mighty Little Bheem: Kite Festival	SHOW	2021	NaN	0	['family', 'comedy', 'animation']	[]	1.0	tt13711094	8.8	16.0

1.2 preprocess_netflix_data

```
def preprocess(self):
    self.netflix_data_raw.drop(columns="seasons", inplace=True)
    self.netflix_data_raw.drop(columns="age_certification", inplace=True)
    self._drop_missing_values()
    self._set_types()
    self.netflix_data.drop(columns="production_countries", inplace=True)
    self._convert_list_to_bool("genres")
    self.netflix_data.drop(columns="genres", inplace=True)
    self._split_data()
    self.data_leakage_warning = self._is_data_leakage()
```


1.2 _drop_missing_values (0,5 Pkt.)


```
def preprocess(self):  
    self.netflix_data_raw.drop(columns="seasons", inplace=True)  
    self.netflix_data_raw.drop(columns="age_certification", inplace=True)  
    self._drop_missing_values()  
    self._set_types()  
    self.netflix_data.drop(columns="production_countries", inplace=True)  
    self._convert_list_to_bool("genres")  
    self.netflix_data.drop(columns="genres", inplace=True)  
    self._split_data()  
    self.data_leakage_warning = self._is_data_leakage()
```

- ▶ Nachdem Sie die Spalten "seasons" und "age_certification" gelöscht haben, löschen Sie alle Zeilen mit ungültigen Werten
- ▶ [5806 rows x 13 columns] -> [5264 rows x 11 columns]

1.2 _set_types (0,5 Pkt.)

```
def preprocess(self):  
    self.netflix_data_raw.drop(columns="seasons", inplace=True)  
    self.netflix_data_raw.drop(columns="age_certification", inplace=True)  
    self._drop_missing_values()  
    self._set_types()  
    self.netflix_data.drop(columns="production_countries", inplace=True)  
    self._convert_list_to_bool("genres")  
    self.netflix_data.drop(columns="genres", inplace=True)  
    self._split_data()  
    self.data_leakage_warning = self._is_data_leakage()
```

- Setzen Sie die Typen aller Spalten entsprechend ihres Inhalts



index	int32
id	string
title	string
type	category
release_year	object
runtime	int32
genres	string
production_countries	string
imdb_id	string
imdb_score	float64
imdb_votes	int32
dtype:	object

1.2 _convert_list_to_bool (1 Pkt.)

```
def preprocess(self):  
    self.netflix_data_raw.drop(columns="seasons", inplace=True)  
    self.netflix_data_raw.drop(columns="age_certification", inplace=True)  
    self._drop_missing_values()  
    self._set_types()  
    self.netflix_data.drop(columns="production_countries", inplace=True)  
    self._convert_list_to_bool("genres")  
    self.netflix_data.drop(columns="genres", inplace=True)  
    self._split_data()  
    self.data_leakage_warning = self._is_data_leakage()
```

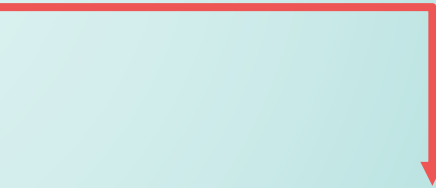
- Erzeugen Sie eine neue Bool-Merkmalsspalte für jedes Genre und setzen sie die entsprechenden Werte

```
expanded_columns: Index(['index', 'id', 'title', 'type', 'release_year', 'runtime', 'imdb_id',  
    'imdb_score', 'imdb_votes', 'crime', 'drama', 'comedy', 'fantasy',  
    'horror', 'european', 'thriller', 'action', 'music', 'romance',  
    'family', 'western', 'war', 'animation', 'documentation', 'history',  
    'scifi', 'reality', 'sport'],  
    dtype='object')
```

1.2 _split_data (1 Pkt.)

```
def preprocess(self):  
    self.netflix_data_raw.drop(columns="seasons", inplace=True)  
    self.netflix_data_raw.drop(columns="age_certification", inplace=True)  
    self._drop_missing_values()  
    self._set_types()  
    self.netflix_data.drop(columns="production_countries", inplace=True)  
    self._convert_list_to_bool("genres")  
    self.netflix_data.drop(columns="genres", inplace=True)  
    self._split_data()  
    self.data_leakage_warning = self._is_data_leakage()
```

- ▶ Teilen Sie den Datensatz in einen Train-, Validations-, und Testsplit auf. Die Mengenverhältnisse können als Klassenvariable gesetzt werden



```
Anzahl Samples im Train Set: 4211  
Anzahl Samples im Validation Set: 1053  
Anzahl Samples im Test Set: 0
```

1.2 _is_data_leakage (1 Pkt.)

```
def preprocess(self):  
    self.netflix_data_raw.drop(columns="seasons", inplace=True)  
    self.netflix_data_raw.drop(columns="age_certification", inplace=True)  
    self._drop_missing_values()  
    self._set_types()  
    self.netflix_data.drop(columns="production_countries", inplace=True)  
    self._convert_list_to_bool("genres")  
    self.netflix_data.drop(columns="genres", inplace=True)  
    self._split_data()  
    self.data_leakage_warning = self._is_data_leakage()
```

- Implementieren Sie eine Prüfung, um auszuschließen, dass Dateneinträge in mehreren Teilen des Splits vorkommen (sogenannte Data Leakage)

Datenleck vorhanden? False

1.3 write_netflix_data (0,5 Pkt.)

test.pickle
train.pickle
val.pickle

- Speichern Sie Train-, Validations- und Testsplit als .pickle-Dateien ab.

	index	id	title	type	release_year	runtime	imdb_id	imdb_score	imdb_votes	crime	drama	comedy	fantasy	horror	european	thriller
	4405	ts256297	The Devil Punisher	SHOW	2020	70	tt13317376	5.9	157	False	False	False	True	False	False	False
	2889	tm430090	The Last Runaway	MOVIE	2018	107	tt7606620	5.9	719	True	True	False	False	False	False	False
	2286	tm369179	Irreplaceable You	MOVIE	2018	96	tt6119856	6.4	9442	False	True	True	False	False	False	False
	254	tm33545	Forgetting Sarah Marshall	MOVIE	2008	111	tt0800039	7.1	280121	False	True	True	False	False	False	False
	5378	tm876608	Sounds Like Love	MOVIE	2021	110	tt11698662	5.4	2023	False	True	True	False	False	False	False

	4315	tm945435	Octonauts and the Great Barrier Reef	MOVIE	2020	47	tt13150606	6.6	131	False	False	True	False	False	False	False
	4502	tm842869	Fadily Camara: La plus drôle de tes copines	MOVIE	2019	54	tt11168150	5.7	46	False	False	True	False	False	False	False
	3868	ts90095	Jinn	SHOW	2019	32	tt10751504	3.5	10043	False	True	False	True	True	False	True
	525	tm44634	Kevin James: Sweat the Small Stuff	MOVIE	2001	42	tt0305727	7.4	1083	False	False	True	False	False	False	False
	806	tm52062	Mosquita y Mari	MOVIE	2012	85	tt1978480	6.3	918	False	True	False	False	False	False	False

[4211 rows x 28 columns]

2. Datenerkundung und -visualisierung

2. Legen Sie ein Jupyter Lab Projekt an und bearbeiten folgende Aufgaben in einzelnen Zellen. Verwenden Sie dafür Ihren vorverarbeiteten Trainingsdatensatz.
 1. Geben Sie Mittelwert und Standardabweichung der Spalten „imdb_score“ und „imdb_votes“ aus (0,5 Pkt.)
 2. Erzeugen Sie je ein Histogramm aus den Daten „imdb_score“ und „imdb_votes“ (0,5 Pkt.)
 3. Tragen Sie in je einem Scatter-Plot „imdb_score“ und „imdb_votes“ gegen „release_year“ auf (0,5 Pkt.)
 4. Verwenden Sie ein Balkendiagramm, um die Beliebtheit von Genres nach der Häufigkeit Ihres Vorkommens im Datensatz zu visualisieren (1 Pkt.)
 5. Tragen Sie in einem Box-Plot IMDB Scores nach Filmgenres auf (1 Pkt.)

Kriterien

- ▶ Vollständigkeit, Funktion und Qualität der Abgabe (10 Punkte)
- ▶ Ausführliche Kommentare auf Deutsch
 - ▶ Bei unzureichenden oder schlechten (unklar, grammatisch falsch, schlechte Rechtschreibung) 1-2 Punkte Abzug
- ▶ Alle Plots mit Achsenbeschriftung und gegebenenfalls Legende
 - ▶ Bis zu 1 Punkt Abzug, bei fehlenden oder unzureichender Darstellung

Organisatorisches

- ▶ Gruppen mit maximal 5 Leuten
- ▶ Abgabe besteht aus einem Ordner mit Ihrem Gruppennamen mit
 - ▶ Einer Datei `netflix_data_reader.py`, die die Klasse aus Teil 1 enthält
 - ▶ Einem Jupyter notebook `data_exploration.ipynb` mit der Lösung aus Teil 2
- ▶ Abgabe am 21.4.
- ▶ Zwei weitere Abgaben am 19.5. und am 16.6. mit je 10 Punkten, 30 Punkte insgesamt
- ▶ Evtl. Bonus Aufgaben gegen Ende des Semesters

Notenschlüssel

Note	Punktzahl
1	30
1,3	28 - 29,5
1,7	26 - 27,5
2	24 - 25,5
2,3	22 - 23,5
2,7	20 - 21,5
3	18 - 19,5
3,3	16 - 17,5
3,7	14 - 15,5
4	12 - 13,5
5	<12