MachineLearnAthon – Microlecture Data Preparation (for tabular data)

Recorded by Lara Kuhlmann

MachineLearnAthon
A project Co-funded by the Erasmus+ programme of the European Union

Learning outcomes of today

After successfully completing this micro-lecture, you are able to….

- Identify the basic steps involved in data preparation
- Handle missing data in a dataset
- Understand different data types
- Perform data transformations

Agenda for today

- Missing data
- Data types
- Data transformation

Missing Data (I)

Product	Price	Country_Tax	Payment_Type	Country	Continent
1	1200	23.0	Diners	Ireland	NaN
1	1200	20.0	Mastercard	United Kingdom	Europe
2	3600	20.0	Visa	United Kingdom	Europe
NaN	NaN	NaN	NaN	Algeria	Africa

- Many data contain missing values
- In Python, missing values are represented with „NaN“, which stands for „Not a Number“
- It is important to check for missing values before applying ML methods to a dataframe
- Before deciding how to handle missing data, it is recommend to start counting the missing values

Count nan per column/row:

```
df_nan.isnull().apply(np.sum, axis=0/1)
```

	axis=0	axis=1	
Product	1	0	1
Price	1	1	0
Country_Tax	1	2	0
Payment_Type	1	3	4
Country	0		
Continent	1		

Bramer, M. (2007). *Principles of data mining*. Springer.

Co-funded by the Erasmus+ programme of the European Union

Missing Data (II)

Product	Price	Country_Tax	Payment_Type	Country	Continent
1	1200	23.0	Diners	Ireland	NaN
1	1200	20.0	Mastercard	United Kingdom	Europe
2	3600	20.0	Visa	United Kingdom	Europe
NaN	NaN	NaN	NaN	Algeria	Africa

- Missing values can bias results or reduce the accuracy of data models
- If the proportion of missing values in a column or row is very high, it might be advisable to drop it
- Before dropping columns or rows, you should consider the entailed information loss
- Also consider the sparsity of your data
- Instead of dropping missing values, you could impute the data (replace the missing data with plausible data)

Bramer, M. (2007). *Principles of data mining*. Springer.

- Drop **rows with > 0 nan** values:

```
df.dropna(how='any', axis=0)
```

- Drop **all nan rows**:

```
df.dropna(how='all', axis=0)
```

- Use `axis=1` for dropping **columns** analogously
- **Replace nan** with value

```
df.fillna(value)
```

Co-funded by the Erasmus+ programme of the European Union

Data Imputation

- Data imputation is „a class of procedures that aims to fill in the missing values with estimated ones”[1]
- Imputation techniques allow you to preserve as much data as possible, improving data integrity without discarding observations
- If possible, use other information available to fill missing values (e.g. in the data example below, the value of Continent in the first row is Europe)
- Common imputation techniques are the mean/ median/ mode imputation, which replace the missing values with either of these values
- More advanced methods include model-based imputation, which train a regressor to predict the missing value

Product	Price	Country_Tax	Payment_Type	Country	Continent
1	1200	23.0	Diners	Ireland	NaN
1	1200	20.0	Mastercard	United Kingdom	Europe
2	3600	20.0	Visa	United Kingdom	Europe
NaN	NaN	NaN	NaN	Algeria	Africa

[1] García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing, p.60.

Data types

Tabular data may contain different data types. The most common data types are:

- Numerical data
  - Integer: Whole numbers (e.g. number of products)
  - Float: Decimal numbers (e.g. stock prices)
- Categorical data
  - Nominal: Categories without any order (e.g. countries)
  - Ordinal: Categories with an order (e.g. rating scales)
- Boolean data: Binary values (True/ False or 0/1)
- Text (string) data: Free-form text (e.g. names, addresses)
- Datetime data

Bramer, M. (2007). *Principles of data mining*. Springer.

Co-funded by the Erasmus+ programme of the European Union

Importance of correct data types

The correct formatting of data…

- Ensures correct interpretation of the data and its meaning
- Impacts how data can be manipulated, filtered, or analyzed (e.g., mathematical operations for numerical data, grouping for categorical data)
- Data type errors can lead to inaccurate analysis, coding bugs, or model failures

Bramer, M. (2007). *Principles of data mining*. Springer.

Co-funded by the Erasmus+ programme of the European Union

Datetime values

- There are different valid datestring formats (see on the right hand side)
- If formatted correctly, each element (day, month and year) of a date can be extracted and for example used to create another feature
- Mathematical operations can be performed on datatime values, such as adding a certain number of days to a date

Valid datestring formats:

- '27-11-2018'
- '2018-11-27'
- '27/11/2018,
- '27 11 2018,
- '27.11.2018
- '20181127,
- etc

Invalid datestring formats:

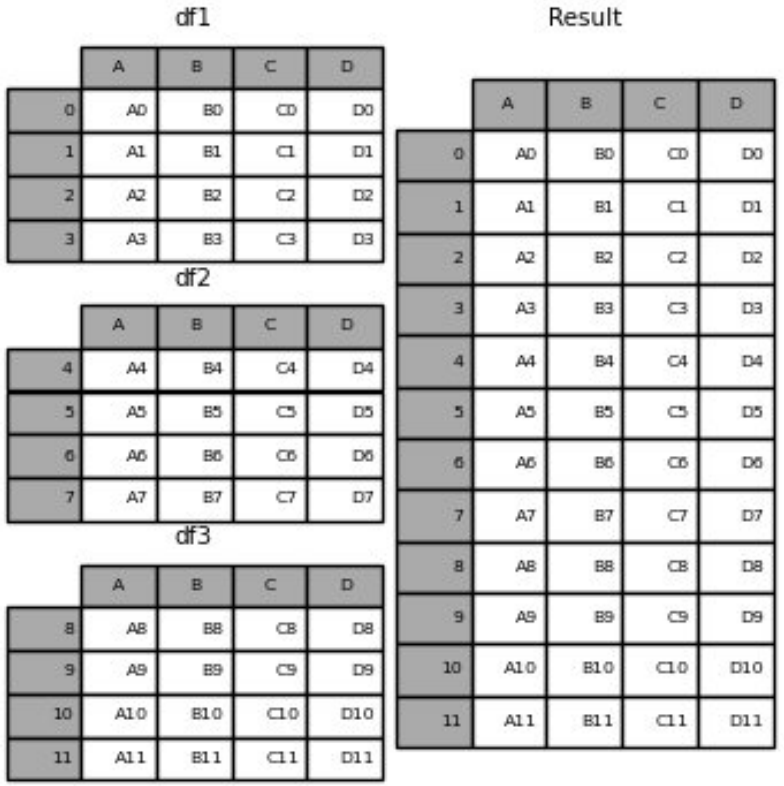
- '27112018
- '27\11\2018,
- etc

Source: https://docs.python.org/3/library/datetime.html

Co-funded by the Erasmus+ programme of the European Union

Combining datasets - concat

- There are different methods for combining multiple datasets
- If the datasets have the same variables, you can concat the data (see on the right hand side)
- The resulting dataframe will have the same numbers of columns but more rows



	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

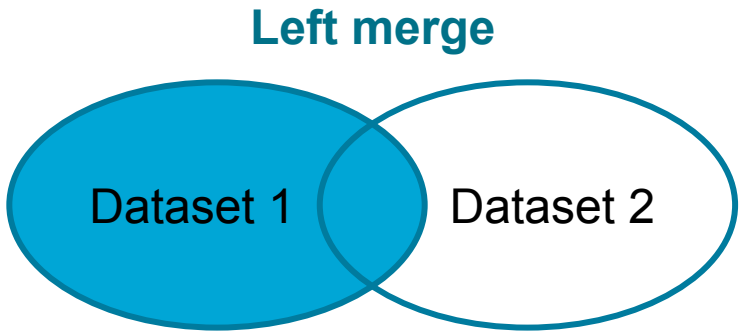
	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Source: https://pandas.pydata.org/docs/user_guide/merging.html

Combining datasets – merge (I)

- If the datasets have at least variable in column, they can be used as keys to join the datasets
- The merge can either be left, right, inner or outer
- In case of a left merge, only the keys from the left dataset are used
- This may result in missing values in the column from the right dataset



left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

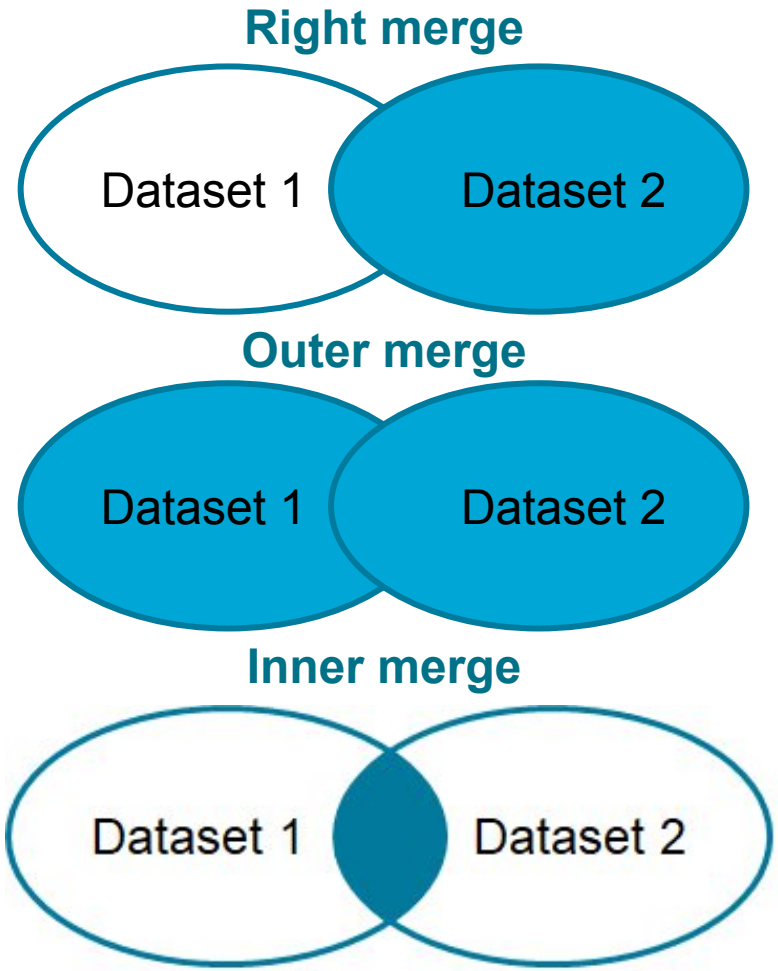
	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN

Source: https://pandas.pydata.org/docs/user_guide/merging.html

Combining datasets – merge (II)



Source: https://pandas.pydata.org/docs/user_guide/merging.html

left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2
3	K2	K0	NaN	NaN	C3	D3

left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K0	NaN	NaN	C3	D3
5	K2	K1	A3	B3	NaN	NaN

left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2

technische universität dortmund
TU Delft
Co-funded by the Erasmus+ programme of the European Union
Machine LearnAthon

Data normalization

- Data normalization is a technique used to rescale numerical data into a standardized range, typically between 0 and 1 or -1 and 1
- This ensures that all features contribute equally to analysis or modeling, regardless of their original scale
- The most commonly used normalization technique is the Min-Max Scaling

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

García et al. (2015): Data Preprocessing in Data Mining, p.46 f.

Further measures

- Delete data
- Delete duplicates
- One-hot encoding (Transforming categorical data into Boolean data)

Picture_id	Subject
1	Muffin
2	Chihuahua
3	Muffin
4	Chihuahua
5	Chihuahua

Picture_id	Muffin	Chihuahua
1	True	False
2	False	True
3	True	False
4	False	True
5	False	True

Source: https://www.freecodecamp.org/news/chihuahua-or-muffin-my-search-for-the-best-computer-vision-api-cbda4d6b425d/

Recap this lecture

After successfully completing this micro-lecture, you are able to….

- Identify the basic steps involved in data preparation
- Handle missing data in a dataset
- Understand different data types
- Perform data transformations

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This material was created with the assistance of an AI language model.

technische universität dortmund
TU Delft
Co-funded by the Erasmus+ programme of the European Union
Machine LearnAthon