Data Handling

Introduction to Data Science
2nd lecture

Prepared by: Assoc. Prof. Alan Jović, PhD

Ac. year 2023/2024

Contents

Data pipeline – data handling

Data storage model → **Data handling** → Data analysis

Data handling – process steps

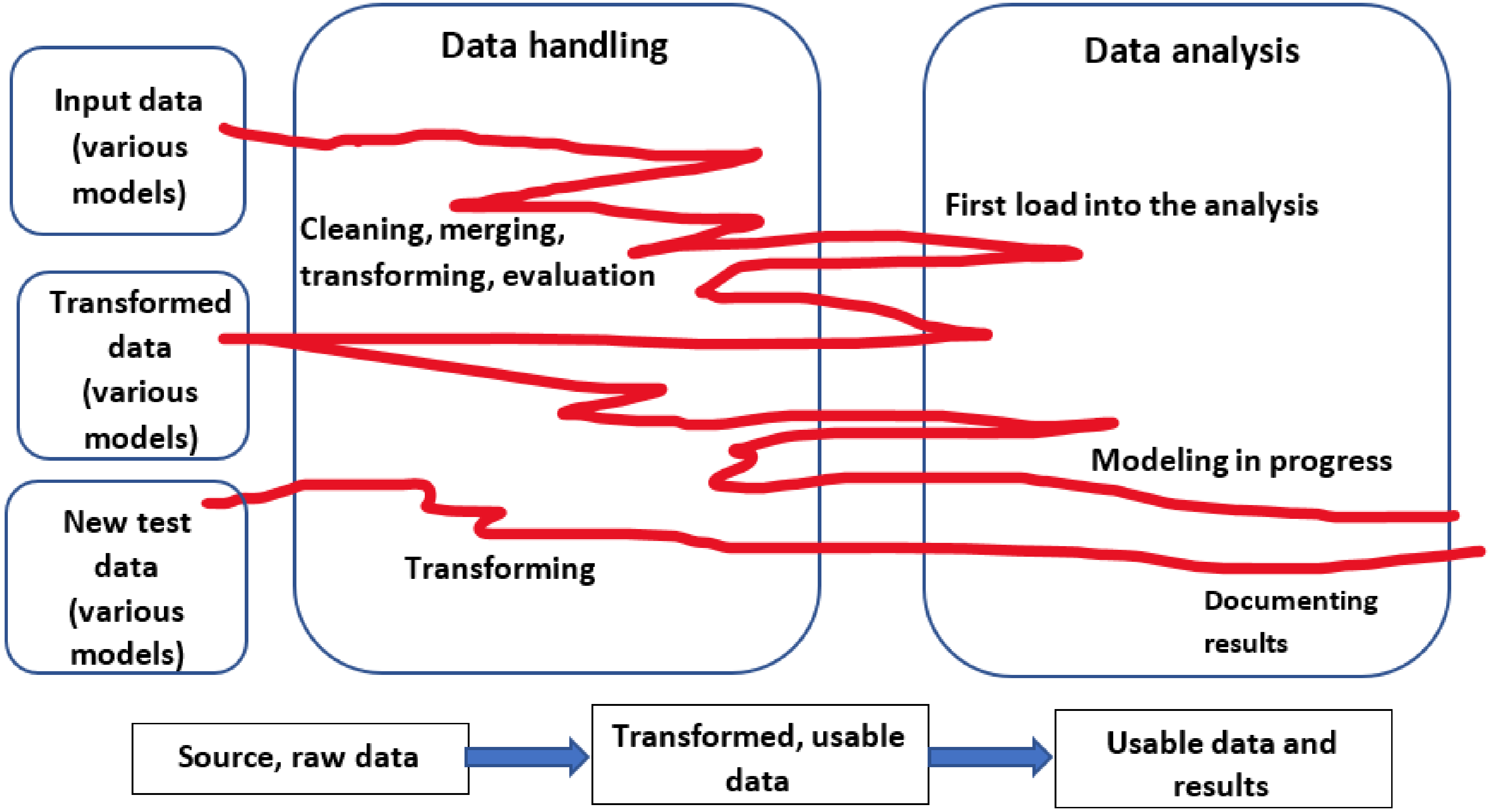
Data handling

- General name for all operations over data that follow:
  - after acquiring source data from the place of storage
  - until the start of the analysis with statistical and machine learning methods
- Alternative names (subtle differences):
  - **Data preparation** – data are being prepared for some time of analysis
  - **Data wrangling** – literally: arguing among data
  - **Data munging** – historically, mung is a term for progressive degradation of a dataset – it is a backronym of ”mash until no good”
- https://www.talend.com/resources/what-is-data-preparation/

Data handling

- Using unprocessed data in future statistical analysis without considering them first – **a recipe for disaster**
  - Can make analysis goal impossible to set properly
  - Can fail machine learning algorithms or give improbable statistics
  - Can lead to inaccurate conclusions
- **50% – 80%** of total time (and money) during data science project goes to data handling
- Data handling is, together with data storage, a basic field of work for a **data engineer**
- The goal of data handling: **prepare data to become reliable and usable**

Data handling



A graph that is difficult to explain, but it strikes directly into the core of data handling, because the process is:

- Extremely *ad hoc* in its execution
- No perfect recipe
- Such that it demands a lot of thinking and sound logics
- Most often unappreciated in companies, where only modeling and **results** are wanted

Adapted from: EPFL, ADA, 2020

Data handling process

- Data handling comprises the following important steps:
  1. **Data survey** (*data exploration, data learning*)
     - Visual and statistical diagnostics of a dataset (including a manual inspection of numbers)
     - The goal is to get to know the data and find their flaws
  2. **Data transformation** (*data organizing, data assembly*)
     - **Transformation of models, formats and data dimensions** in a form useful for analysis
     - First, a transformation to a relational, table form
     - Can including finding and merging with additional data sources (*data enrichment, data merging*)
     - In cases of small analyses and locally available data, can be skipped

Data handling process

- Data handling comprises the following important steps:

3. **Data cleaning**
    - Finding and removing errors, duplicates, synonyms, outliers, missing values and other dataset problems
4. **Data validation** (*data authentication*)
    - After previous steps, checks whether the data are now correct
    - Sometimes implicitly included in all the previous steps
    - More details about what is being checked: https://corporatefinanceinstitute.com/resources/knowledge/data-analysis/data-validation/

Data handling process

- Data handling comprises the following important steps:

5. **Data loading** – optional as a separate step
   - Data are being **loaded into a data structure** suitable for further analysis (if previously changed in another place or in another format)

6. **Data augmentation**
   - Changes size and diversiveness in dataset examples
7. **Feature engineering**
   - Work on dataset features

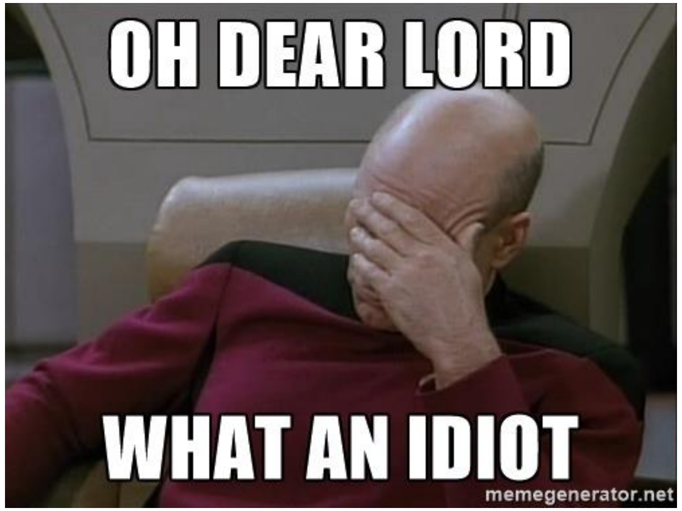
Last two steps can be a part of **data handling**, but also of **data analysis**

Data preprocessing

- ”Something” that happens with previously prepared dataset before a ”real” analysis

Dataset problems

Horror stories about “dirty data”



- “Dear Idiot” letter
- 17,000 men are pregnant
- “As the crow flies“

https://www.linkedin.com/pulse/dirty-data-horror-stories-when-michael/

Point taken: significant portion of data in companies is “bad“ (10–25%, depending on the company, different assessments)

Types of common problems in datasets

- **Missing data**
- **Incorrect data**
- **Outliers**
- **Sparse data**
- **Noisy data**
- **Monotonic features**
- **Imbalanced datasets**

About 75% of dataset problems requires a human intervention to be made (e.g., experts in a field, crowdsourcing)

Example of a dataset with several problems

ID	First name	Last name	Height	Weight	Sex	Age	Hypertension	AMI
1	Peter	Jackson	85	169	M	61	?	No
2	Humphrey	Bogart	174	68	M	123	Yes	No
3	Carrie	Fisher	140	65	F	66	No	Yes
4	Peter	Sellers	173	67	M	118	No	Yes
5	Scarlett	Johansson	160	56	F	38		No
6	Sigourney	Weaver	182	66	Ž	74	Null	No

Missing data

Two main types:

- **Missing (but known) values**
  - Exist in the real process, but were not put into the dataset
- **Empty (unknown) values**
  - A value can not assumed in a real world and is not put into the dataset

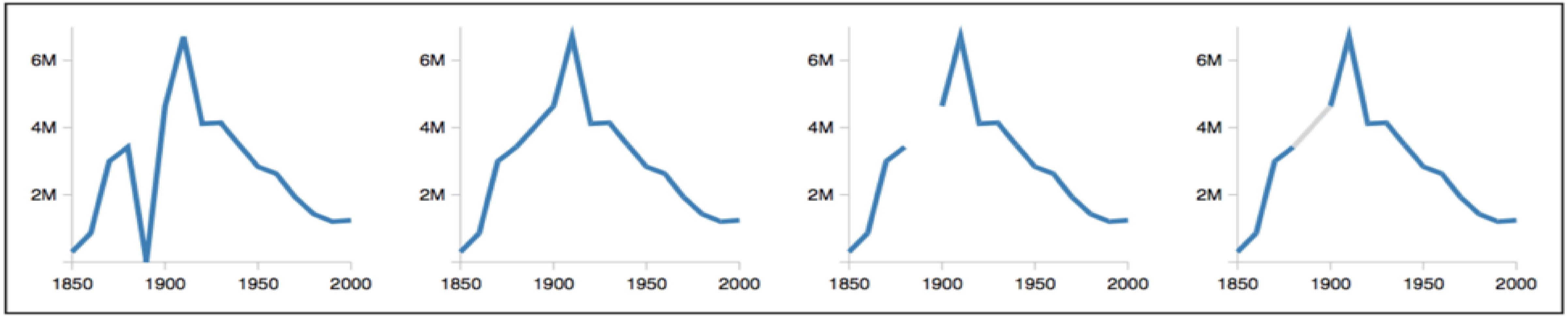
- Often, we are not certain with which type of missing data we are dealing with
  - Various specific values stored in the place of missing data
    - ‘’ – empty field, ‘-’, ‘x’, ‘NULL’, ‘N/A’, ‘BLANK’, ‘„”’ – various types of apostrophes, ‘?’, ‘???’ ...
- Problem detection with **detailed data survey** or **the use of visualization**
- Solving missing data problems in practice is most often independent of the missing data type

Missing data – solving the problem

- **Disregard all examples (objects) that contain it**
  - Sometimes not possible, e.g., when **most** examples have a value of a feature missing
- **Replace the missing value with some other value**
  - Under the assurance that dataset information content does not degrade
  - Simple methods observe one feature
    - **Preserve the measure of mean** – replace with an arithmetic mean, median ili dominant value (mod value)
    - **Preserve variability** – if needed, add noise with replacement to preserve variability
  - More complex methods consider the relation among more than one feature and select a replacement that will influence the whole data **the least**
    - E.g. regression, *k*-nearest neighbors algorithm

An example

- USA population census, people who work on a farm are shown, data from 1890 was lost due to fire (records burned down)



What to do with 1890?

- Set value to 0?
- Interpolate based on close data?
- Disregard missing data entirely?

Domain knowledge and knowledge of data acquisition should lead the choice of replacement method!

Incorrect data

- Often a result of human error during input
- Sometimes put in deliberately
  - User does not know the exact information, but does not want to leave it empty
  - User does not want someone else to know the correct information
  - User has some benefit to enter an incorrect information
- Rarely a result of technical system malfunction
- In a general case, an **insolvable problem**
- **Requires a detailed survey, visualization and thinking about the data**

Outliers

- Data that stands out as **being far outside of usual values** for a specific feature or features
- Reasons for appearance of such data: incorrect input, measurement error, data processing error, natural state
- Problem if the data are incorrect – if they are not the result of the natural state
- They need to be found and removed (if the experts agree that it does not show the natural state)

Outliers

- **Used methods of discovery**
  - **Data visualization**
  - **Statistical methods** – z-value, linear regression
  - **Algorithms of unsupervised machine learning**
    - Based on distance, density, grouping, etc.

Sparse data

- A case when for some features, only a small number of examples has a value different from 0
  - Common with text and document analysis datasets
- A majority of machine learning algorithms **work badly with sparse data**
  - Model overfitting – bad generalization on the test data, providing advantage or disregrading features with sparse data
- Approaches to solving the problem
  - Remove features with sparse data
  - Reduce dimensionality – e.g., principal component analysis method
  - Use of machine learning methods more resistent to sparse data

Noisy data

- Data noise is present to some extent in all data that are the result of **measurements**
- **Data = real signal + noise**
  - Noise is the result of natural processes
  - Noise is the result of measurement sensors imperfection
  - Present with 1D, 2D and 3D signals
- There are methods for noise reduction when signal/noise ratio is infavorable
  - **Methods are extremely dependent on the specific problem**
  - E.g., baseline wandering reduction and electric current frequence (50Hz) filtering in ECG recordings
- Sometimes, it is not possible to remove noise (partially or completely)
  - The dataset on which the model is build should have the same statistical properties as the dataset on which the model will be tested / applied

Monotonic features

- **Features that have values that rise (or decline) with no limits**
- The most common examples
  - Features connected to the passage of time, e.g., dates in various formats
  - Features of ordinal numbers of records, IDs, etc.
- Problem solutions:
  - **Disregards such a feature (the most common solution)**
  - Transform into a specific form suitable for modeling
    - E.g. A date can se transformed into a season or a day of the week, that have cycles, if there is a need for such a data, or it can be turned into a time series

Imbalanced datasets

- A dataset problem in which there is an **imbalance in the number of examples of individual target feature classes**
    - E.g., 95% of examples are from healthy persons, 5% are from patients suffering from a disorder
- Imbalance in the number of examples of individual classes makes it difficult to build a model that will classify examples of the **majority** and **minority** classes equally well
- Multiple approaches to solving the problem
    - **Acquire more data for minority class**
    - **Resampling** – oversampling and undersampling
    - Cost-sensitive learning
    - Application of classifier ensembles
    - ...

Data augmentation

- Data augmentation comes after the data handling phase and before data analysis, together (in parallel) with feature engineering
- Unlike feature engineering, here the focus is on **examples (objects)**
- **Artificial increase in the number of examples**
- Not done always, but depending on the need
    - More often if models are used that need a lot of data (e.g., deep learning models)
    - More rarely if there is enough data
    - More rarely if the data are well-balanced among classes
    - More often in computer vision and natural language processing tasks

Data augmentation

- Generating new synthetic examples
  - Direct copies of old examples
  - With added noise over old examples
  - Based on the “nearest neighbors” examples
  - Transformations of old examples
    - **In images**: rotation, translation, scaling, flipping, cutting, color improvement, contrast improvement, saturation improvement...
    - **In natural language processing**: translation to a number of foreign languages and then back again

Feature engineering

Feature engineering

- Feature: a measurable property of an example that should be taken into account

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. (	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mr	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, M	female	26	0	0	STON/O2. 31	7.925		S
5	4	1	1	Futrelle, Mrs	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. Wi	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. J	male		0	0	330877	8.4583		Q

Source: https://www.datarobot.com/wiki/feature/

Feature engineering

- Feature engineering is a **process** in which one tries to **select or transform** the most relevant variables (features) from a prepared dataset with the goal of successful modeling
- One differs:
  - **manual approach** to feature engineering (domain knowledge is very relevant)
  - **semi-automated approach** to feature engineering (domain knowledge is less important)
  - **fully-automated approach** to feature engineering (domain knowledge has no role)

Manual approach to feature engineering

- **Extracting (calculating) features** (**feature extraction, feature elicitation**)
  - Define, implement, and calculate features from **raw** data
  - Potentially **infinite space** of features
  - In signal analysis, one differs:
    - Time domain features (often statistical features)
    - Frequency domain features (features obtain from signal's frequency spectrum)
    - Nonlinear features (phase space features, entropies, ...)
  - Different image features (e.g., color histograms) and volume data features
  - Features are usually calculated **after previous preparation** (e.g., noise removal, missing values interpolation, and similar)

Manual approach to feature engineering

- Is characterized by a **review of individual features**, and then:

- **Adding new features based on the existing ones**
- **Removal of irrelevant features**

Manual approach to feature engineering

- **Adding new features based on the existing ones**
  - Usually done after feature extraction from raw data
  - Feature construction based on a single existing feature
    - Numerical values discretization (***binning***) – not very common today, tool-dependent
    - Transformation of a categorical to numerical feature (***label encoding***)
    - Transformation of a categorical to multiple binary features (***one-hot encoding***)
    - Value normalization
  - Construction based on multiple existing features
    - Manual combination of multiple features into a single one, e.g., sum, quotient, product, etc.

Transformation of one categorical to multiple binary features

- Many machine learning algorithms cannot work directly with **categorical values**, they require that all input and target variables are **numerical**
- A limitation made by an **effective implementation** of machine learning algorithms
- Transformation of a categorical feature to a numerical one – **label encoding**: category1 -> 1 ; category2 -> 2 .... category*n* -> *n* **only when ordering of categories has some sense**
- Otherwise, **each category** of a categorical feature **becomes a new binary feature** – **one-hot encoding**
    - Of *n* categories we get *n* binary features, which have value of 1 for those examples for which the corresponding category is valid, and 0 otherwise
- https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

Value normalization

- Necessary when different features are **measured on different scales**
  - Features measured on lower scales (e.g., between 1 and 10) would be less relevant to a model than those on the higher scales (e.g., between 1000 and 10000), which would lead to worse results
- **The most common normalization is to transform the values into range between 0 and 1**
- Normalization methods
  - **Decimal scaling** (divide values with the maximal value of the decimal space)
    - E.g., if all values are up to 100, and at least some is larger than 10, then divide with 100
  - **Min-Max** normalization (linear transformation of values): $x' = (x - min) / (max - min)$
  - **z-value** normalization (statistical normalization using mean and variance), also known as **standardization:** $x' = (x - mean) / stdev$

Manual approach to feature engineering

- **Removal of irrelevant features**
  - Monotonic features
  - Constant features
  - Features with very sparse data
  - Duplicates and **statistically redundant features**
    - Most commonly – correlation analysis

Removal of statistically redundant features by correlation analysis

person_name	is_male	is_female
Aman	1	0
Abhinav	1	0
Ashutosh	1	0
Dishi	0	1
Abhishek	1	0
Avantika	1	0
Ayushi	0	1

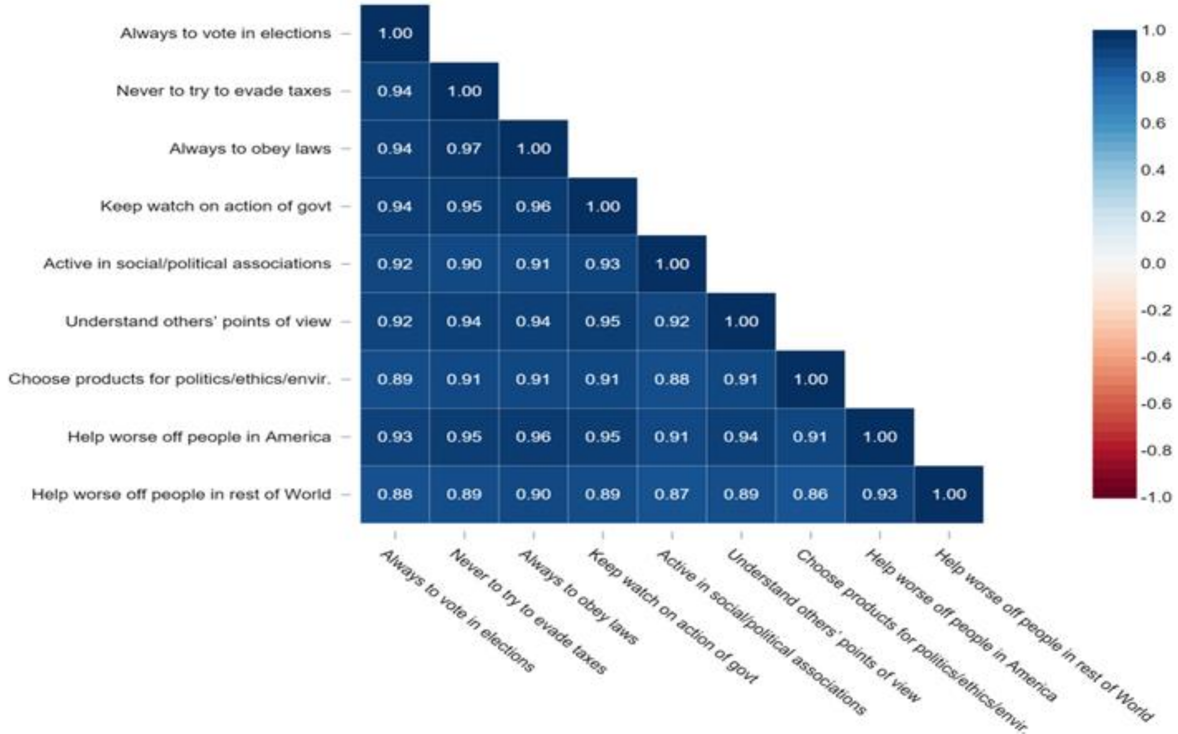
HIGHLY CORRELATED ATTRIBUTES

One attribute can be removed without any information loss. As one attribue can easily determine the other.

Source: https://www.geeksforgeeks.org/redundancy-and-correlation-in-data-mining/

Removal of statistically redundant features by correlation analysis

- Correlation is calculated between every two variables in the set and a correlation matrix is built
- For the two variables for which the correlation value is very high (ideally 1), you **select one of them for removal from the dataset** – that one is redundant
- Threshold of correlation coefficient value for removal of a feature depends on domain and goal of the analysis, but is usually higher than 0.9
- Sometimes it is better not to remove a feature if we are unsure whether that would be correct



Source: https://www.displayr.com/what-is-a-correlation-matrix/

Semi-automated approach to feature engineering

- **Feature selection**
- **Dimensionality reduction**

Feature selection

- Features are **removed** from the dataset – this reduces its **dimensionality**
- In feature selection, **the interpretation of features is kept**, because those that are kept are **not changed**
- One wants to **keep the result** of modeling of the initial feature set or to **improve the result**
- Methods:
  - **Filters**
  - **Wrappers**
  - **Embedded methods**
  - **Hybrid methods**

More common

Less common

Feature selection

- Optimal feature subset = the smallest possible number of features that gives the best results (for classification, prediction...)
- The search for the optimal feature subset is an **NP-hard problem**
  - Search 2^M feature subsets, where M is the number of features
- Existing empirical methods of search usually work in polynomial time and **do not guaranty finding the optimal subset**

Filters

- Filter methods define a **criterium** that shows how a specific feature is relevant for the description of a target variable
- Usually, the **features are ranked** with respect to the criterium
  - User can then select first *n* features
- Different filters (each with its own mathematical formulation):
  - **mutual information**
  - **chi-square, χ^2**
  - symmetrical uncertainty
  - Relief (Relief, ReliefR, ReliefC...)
  - **correlation coefficient (mostly for regression problems)**

Wrappers

- Use **a machine learning algorithm for evaluation** of a specific feature subset in order to determine whether that subset is better / the same / worse than its superset
- Machine learning algorithm is often not the one that would be used later for building a model
  - Fast algorithms are preferred, in order to evaluate as many feature subsets as possible, e.g., Naive Bayes
- Search of feature subsets space can start from the full set or from an empty set and use different search strategies (a naive approach would be random guessing)
  - Greedy strategies (e.g., best first)
  - Forward selection and backward elimination
  - Evolutionary algorithms
- **Wrapper: slower, but more accurate methods than filters**

Dimensionality reduction

- Problem: high dimensionality (number of variables) in a dataset
- **The curse of dimensionality**: data in the large number of dimensions become **sparse**
  - Learning algorithms have difficulty adjusting to sparse data, which leads to a weaker generalization
  - An exponential number of examples is needed, with respect to the number of variables, to populate the space
- The goal is to **reduce dimensionality** of the problem, while keeping the initial information in the data
- Unlike feature selection methods, dimensionality reduction methods transform initial features
- Methods:
  - Principal Component Analysis, PCA https://www.geeksforgeeks.org/principal-component-analysis-pca/
  - Multidimensional Scaling, MDS https://www.statisticshowto.com/multidimensional-scaling/
  - Autoencoders https://www.jeremyjordan.me/autoencoders/
  - ...

Fully automated approach to feature engineering

- **Feature learning, representation learning**
  - An approach with which one bypasses expert features extraction
    - The approach is independent of domain knowledge
    - Increasingly used in different application areas (biomedicine, computer vision)
  - An assumption is that one works with **raw input data** (cleaned, prepared) and most often:
    - Signals (1D time series)
    - Images – 2D signals
    - Volume data – 3D signals
  - Raw data are being transformed within the algorithm to an internal model that is described with low-level features
    - Features that have a clear mathematical formulation but unclear semantics

Fully automated approach to feature engineering

- A particular **machine learning algorithm** is used for internal learning of new features
  - The idea is that new features will be **highly discriminatory and useful** for the problem being solved
  - New features are obtained by **transformations of input data** or the initial feature set
  - New features are mostly called **representations**
  - Both supervised and unsupervised algorithms are used
- Some known feature learning algorithms
  - Traditional: ICA
  - **Deep learning: multilayer perceptron, convolutional neural network, autoencoders, and restricted Boltzmann machines**
- https://towardsdatascience.com/unsupervised-feature-learning-46a2fe399929

References

- Alice Zheng, Amanda Casari (2018), *Feature Engineering for Machine Learning*, O'Reilly Media
- Dorian Pyle (1999), *Data Preparation for Data Mining*, Morgan Kaufmann
- Alan Jović, Karla Brkić, Nikola Bogunović (2015), A review of feature selection methods with application, *MIPRO 2015*, https://ieeexplore.ieee.org/abstract/document/7160458

Conclusions

- Data handling is a **complex process** with which data is being prepared for analysis
  - It comprises of a series of steps and data transformations
- Dataset plays a big role in the process – dataset size and features are important
- Dataset can have various problems, some are easily solvable, some are not
  - There is no perfect solution for all problems
  - One needs a lot of engineering work
- Feature engineering stresses out the important that features have for the usefulness of future data analysis
  - Manual approach, semi-automated approach, fully automated approach
  - The goal is usually to find the optimal set of features for a given problem