PROFINIT
An Amdocs Company

Data Preparation

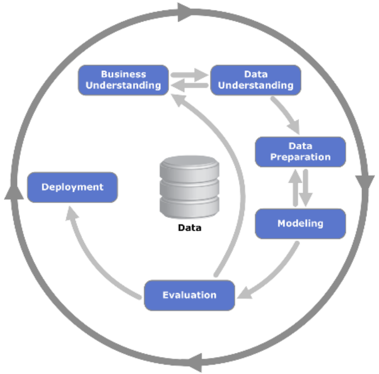
Eva Blažková, Jan Hučín

3. listopadu 2025

CRISP-DM a Data Preparation

PROFINIT
An Amdocs Company

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives *Background* *Business Objectives* *Business Success Criteria* **Assess Situation** *Inventory of Resources* *Requirements, Assumptions, and Constraints* *Risks and Contingencies* *Terminology* *Costs and Benefits* **Determine Data Mining Goals** *Data Mining Goals* *Data Mining Success Criteria* **Produce Project Plan** *Project Plan* *Initial Assessment of Tools and Techniques*	**Collect Initial Data** *Initial Data Collection Report* **Describe Data** *Data Description Report* **Explore Data** *Data Exploration Report* **Verify Data Quality** *Data Quality Report*	**Select Data** *Rationale for Inclusion/ Exclusion* **Clean Data** *Data Cleaning Report* **Construct Data** *Derived Attributes* *Generated Records* **Integrate Data** *Merged Data* **Format Data** *Reformatted Data* *Dataset* *Dataset Description*	**Select Modeling Techniques** *Modeling Technique* *Modeling Assumptions* **Generate Test Design** *Test Design* **Build Model** *Parameter Settings* *Models* *Model Descriptions* **Assess Model** *Model Assessment* *Revised Parameter Settings*	**Evaluate Results** *Assessment of Data Mining Results w.r.t. Business Success Criteria* *Approved Models* **Review Process** *Review of Process* **Determine Next Steps** *List of Possible Actions* *Decision*	**Plan Deployment** *Deployment Plan* **Plan Monitoring and Maintenance** *Monitoring and Maintenance Plan* **Produce Final Report** *Final Report* *Final Presentation* **Review Project** *Experience Documentation*



https://www.mygreatlearning.com/blog/why-using-crisp-dm-will-make-you-a-better-data-scientist/

Data preparation phases

- **Select data**
  - Which (portions of) data sets will (not) be used and why?
  - Collect additional data (internal, external)
- **Clean data**:
  - Correct, replace, remove, ignore noise
  - Deal with special values, missing values and outliers
  - Aggregation level
- **Construct data (feature eng)**
  - Extract new attributes or re-construct missing (BMI, day of the week)
- **Integrate data**
  - combinine data from multiple sources
- **Format data (transform data)**
  - Re-arrange, re-order, re-format (string -> numeric, date formatting, re-ordering categrories)

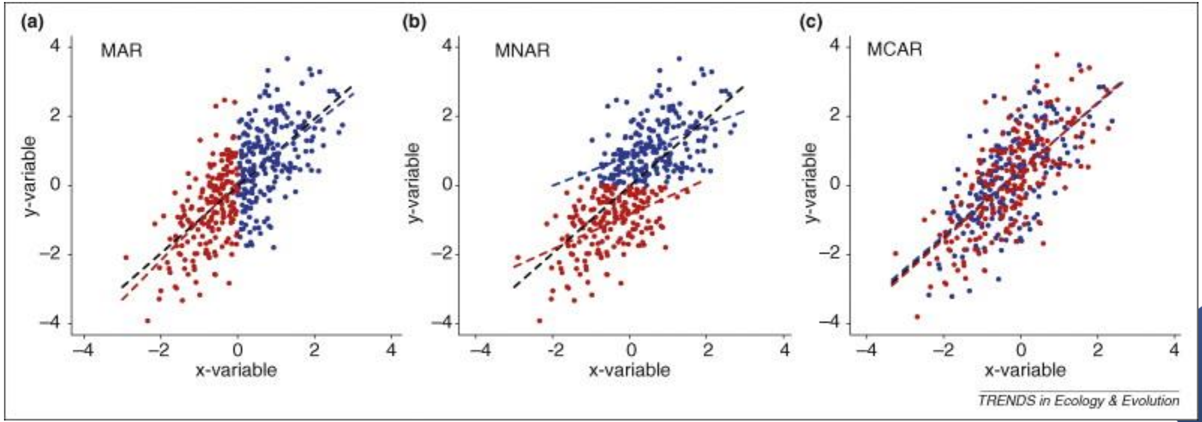
Data cleaning

Real-world data are dirty

Incomplete	
• Types of missingness: • MCaR, MaR, MnaR	• Not available when collected • Different criteria when collected • Human/computer errors
Noisy	Income = -10 000
• Errors / outliers • Various sources: spelling, typos, word transpositions, multiple values in a single field	• Failures during data collection • Human/Computer errors • Data transmission errors
Inconsistent	Rating 1-5 vs. A-C
• Synonyms, prefix/suffix, variations, abbreviations, truncation and initials • Inconsistencies among datasets/subgroups	• Different data sources • Different stages of collection

Missing values

- **Missing completely at random (MCAR)**
  - No difference between our primary variable of interest and the missing and non-missing values
- **Missing at random (MAR)**
  - No significant difference between our primary variable of interest and the missing and non-missing values
  - Not a realistic assumption for many real-time data
- **Missing not at random (MNAR)**
  - Missing values depend on our primary variable of interest or on other unobserved variables
  - Not ignorable



Missing values

- **Missing completely at random (MCAR)**
  -
- **Missing at random (MAR)**
  -
  -
- **Missing not at random (MNAR)**
  -

- X = income
- Y = credit risk

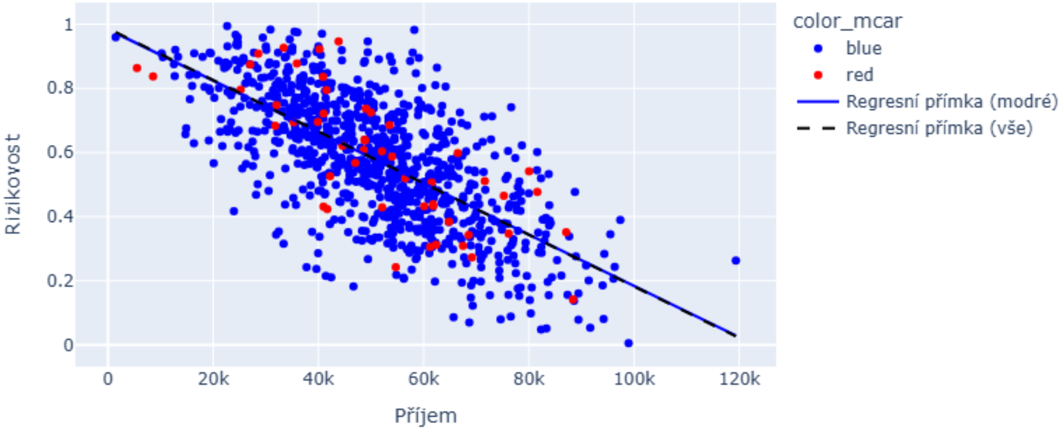
  (loan approval)

Missing values

- **Missing completely at random (MCAR)**
  - During data migration from system A to system B, some data were randomly lost.
- **Missing at random (MAR)**
  -
  -
- **Missing not at random (MNAR)**
  -

- X = income
- Y = credit risk

  (loan approval)



Missing values

- **Missing completely at random (MCAR)**
  - During data migration from system A to system B, some data were randomly lost.
- **Missing at random (MAR)**
  -
  -
- **Missing not at random (MNAR)**
  -

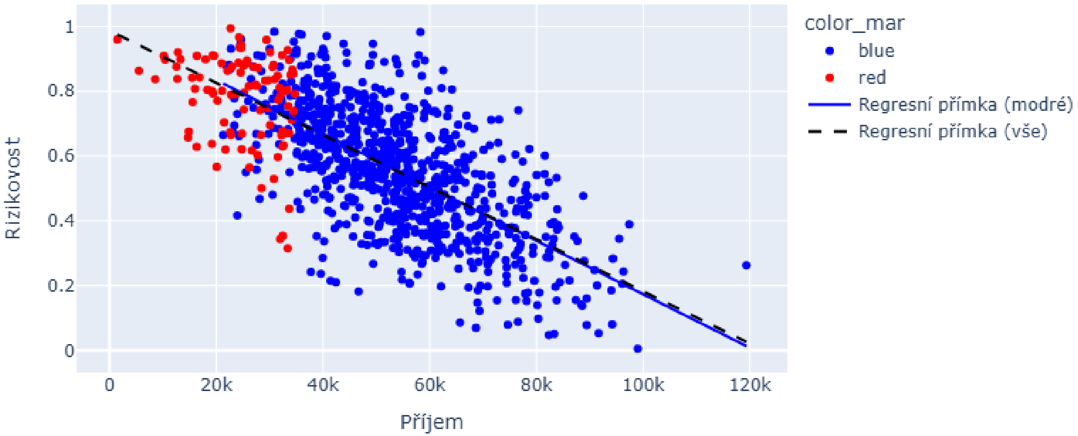
- X = income
- Y = credit risk

  (loan approval)

Missing values

PROFINIT
An Amdocs Company

- **Missing completely at random (MCAR)**
  - During data migration from system A to system B, some data were randomly lost.
- **Missing at random (MAR)**
  - Clients with low income avoid filling it in, fearing loan rejection
  - Self-employed clients often omit income as it's harder to document
- **Missing not at random (MNAR)**
  -



- X = income
- Y = credit risk

  (loan approval)

Missing values

PROFINIT
An Amdocs Company

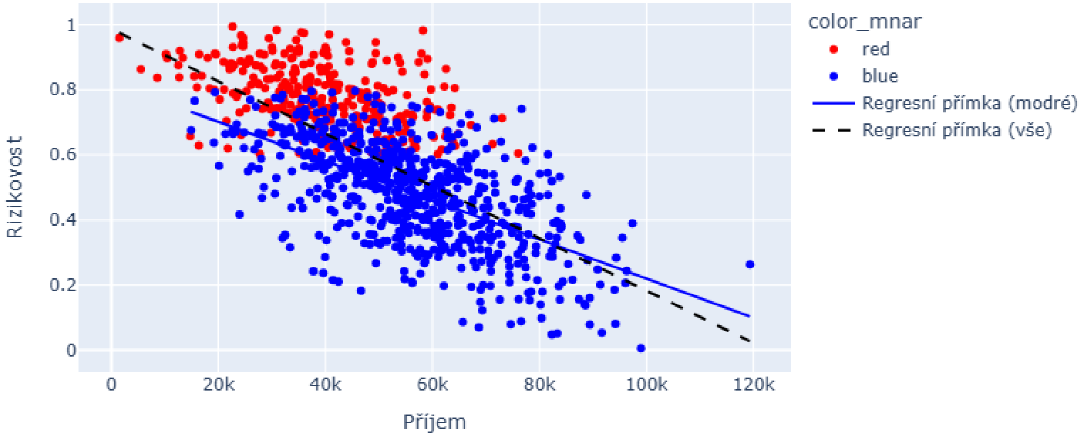
- **Missing completely at random (MCAR)**
  - During data migration from system A to system B, some data were randomly lost.
- **Missing at random (MAR)**
  - Clients with low income avoid filling it in, fearing loan rejection
  - Self-employed clients often omit income as it's harder to document
- **Missing not at random (MNAR)**
  -

- X = income
- Y = credit risk

  (loan approval)

Missing values

- **Missing completely at random (MCAR)**
  - During data migration from system A to system B, some data were randomly lost.
- **Missing at random (MAR)**
  - Clients with low income avoid filling it in, fearing loan rejection
  - Self-employed clients often omit income as it's harder to document
- **Missing not at random (MNAR)**
  - For high-risk clients, income is missing because they didn't reach the part of the application where income is entered



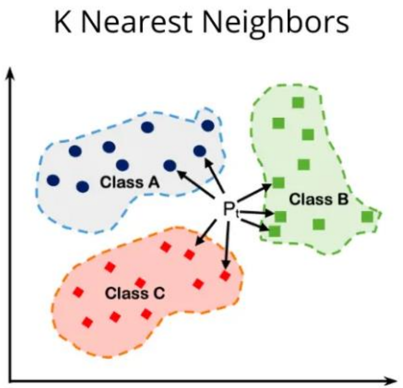
- X = income
- Y = credit risk

  (loan approval)

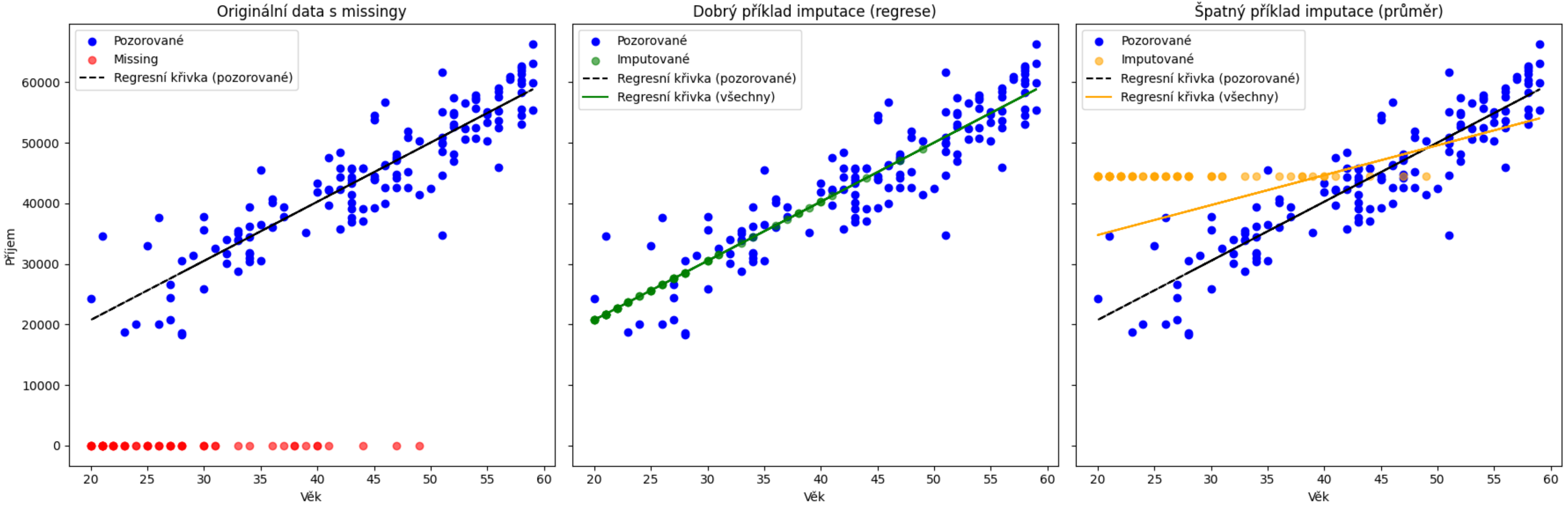
Dealing with missingness

- Delete coumns/rows
- Ignore
- Fill-in the holes
  - Reconstruct
  - Global constant (use as information)
  - Impute
    - Typical value – mean, median, modus
    - Random (observed) value
    - Use submodel
    - K Nearest Neighbors
    - EM algoritmus

- Consistently
- Document the process

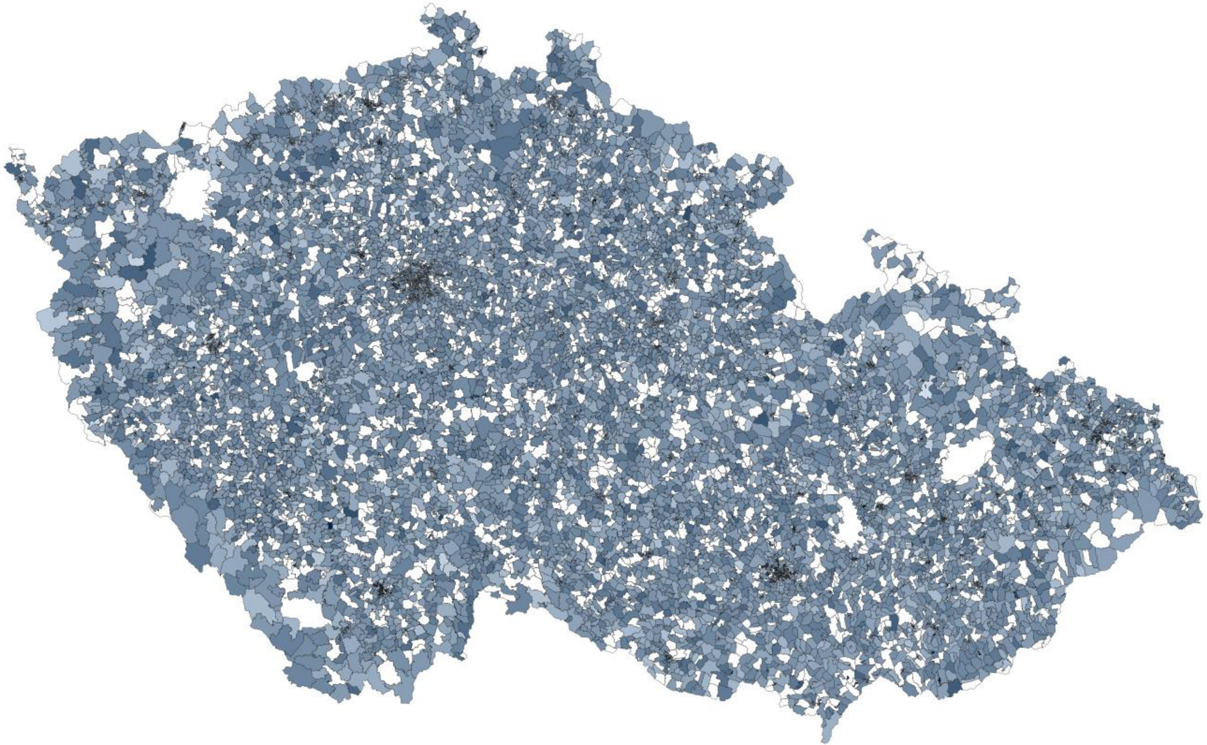


Imputation - Example



Geoscore

- Electoral districts change slightly with each election
- Missing data are filled in using a submodel



Příklady

- Reviews
- Limnigraphs
- Precipitation
- Vehicule age
- Personal data (GDPR)

Noise

- Random error/variance added to measured variables
- Not necessary gaussian
- Weakens target ~ feature relationship
- Domain knowledge sometimes necessary

Reduce the noise

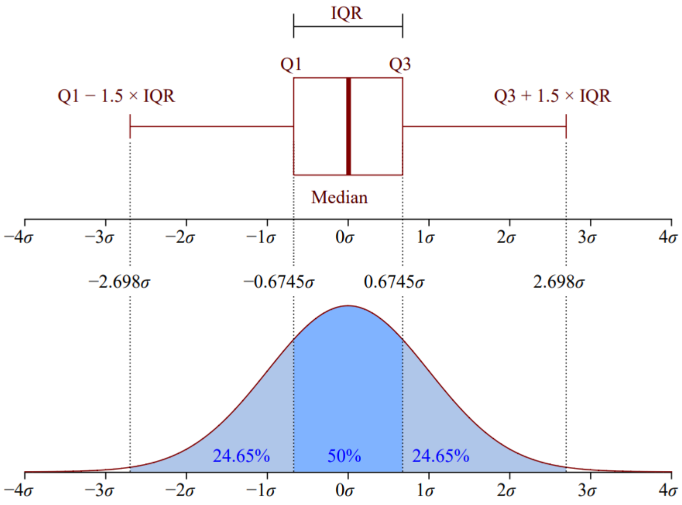
- Binning
  - Sort values and than cut into bins
- Smoothing
  - Fit subomodel and use fitted values instead

Outliers

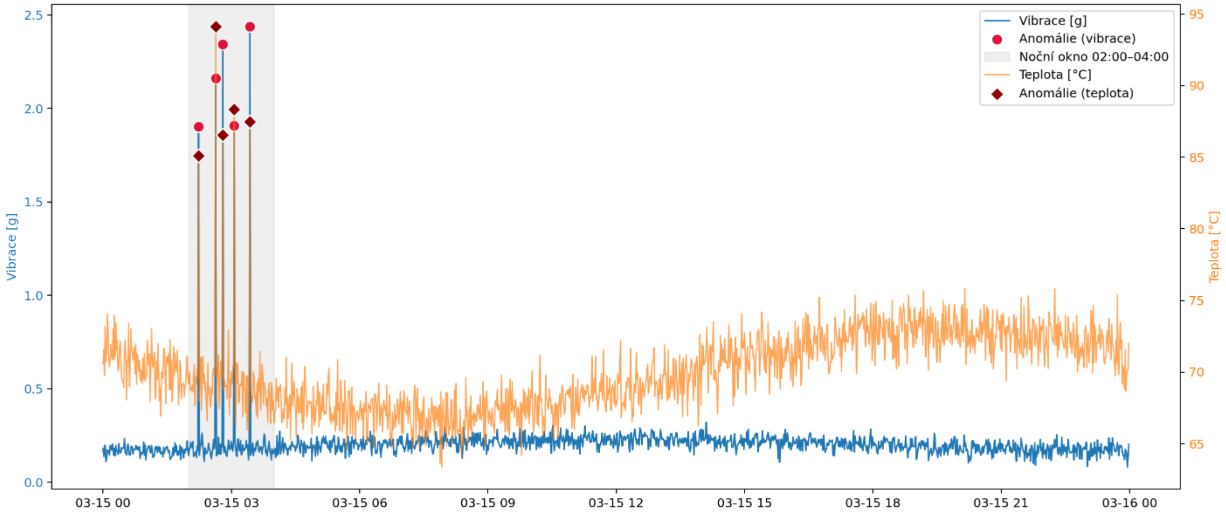
- Errors might introduce too influential observations
  - Very large/small/otherwise strange values
  - Might be a multidim. problem (e.g. short basketball player)

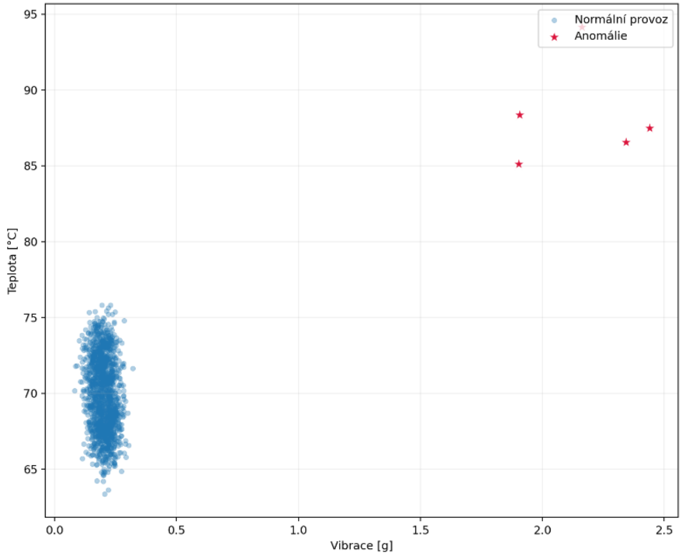
How to detect outliers

- Visually (boxplot, scatterplot)
- Statistics
  - Z-score $Z = \dfrac{X - \mathrm{E}[X]}{\sigma(X)}$
  - $x < Q1 - 1.5 * IQR$ *and*
    $x > Q3 + 1.5 * IQR$



Anomaly detection – IoT sensors



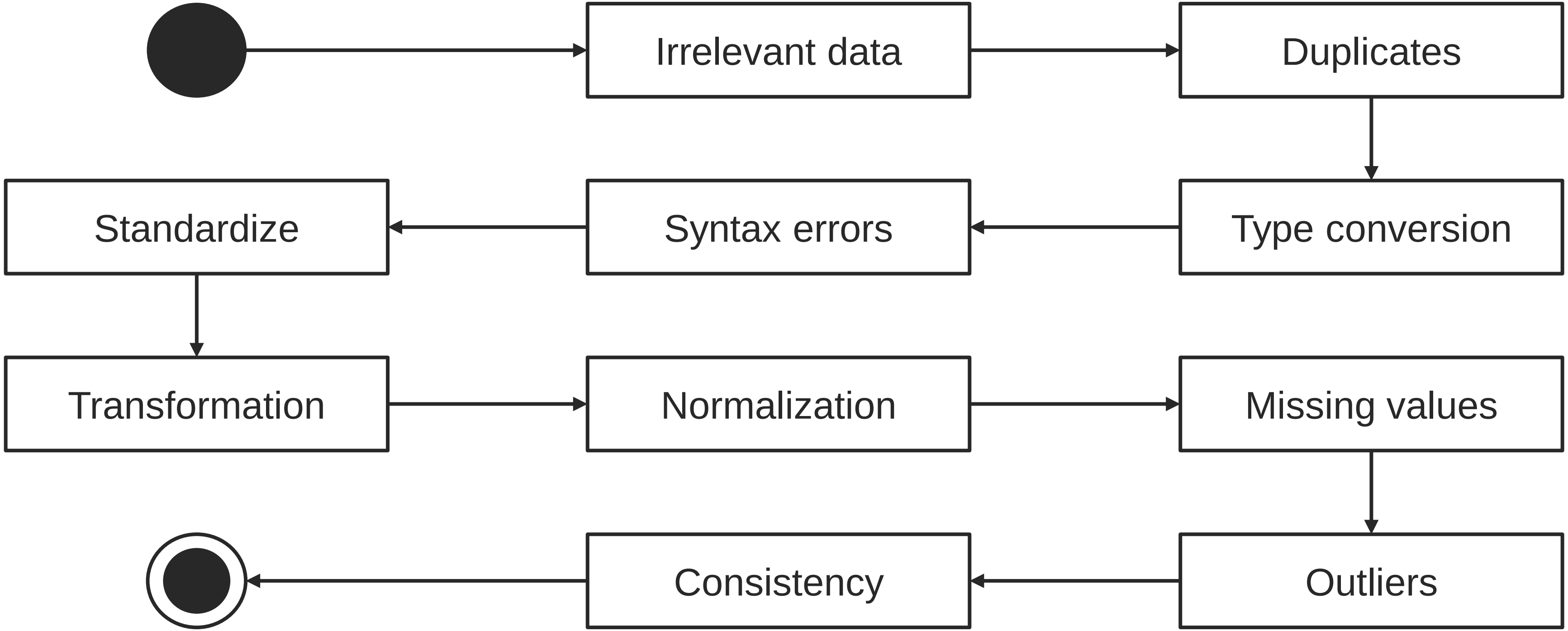


Inconsistent data

- Different data sources claim different things
- Domain knowledge might be necessery

- Replace by N/A
- Repair
  - Pick the ground truth dataset
  - String distances (Levenshtein)
- Clustering
- Ignore/ Remove

Data cleaning



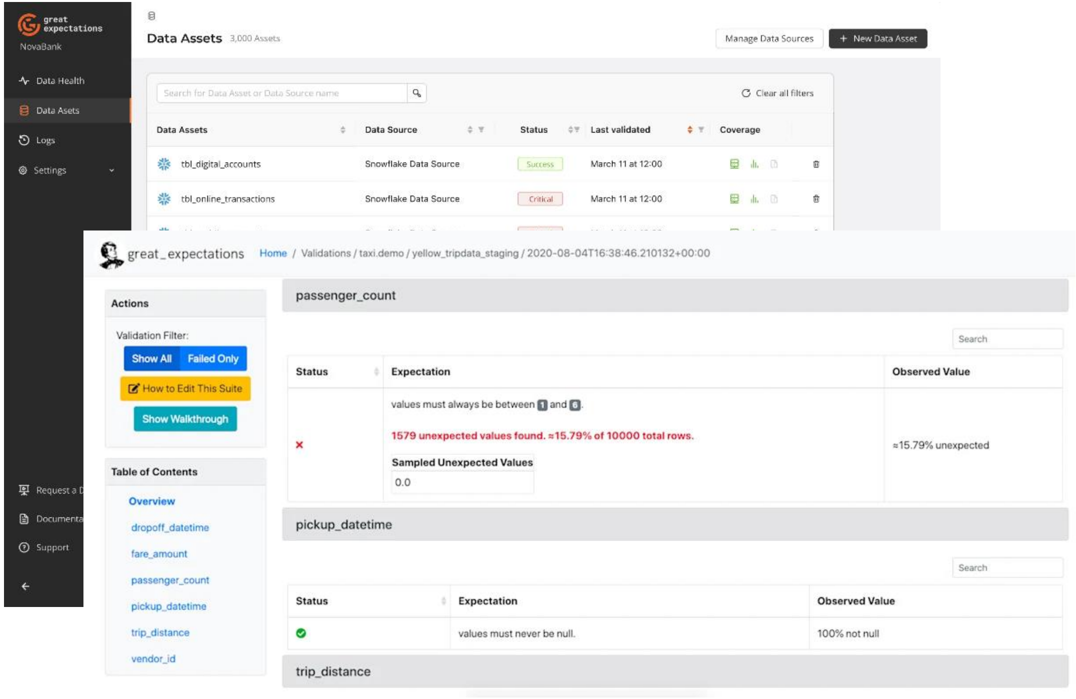
Data cleaning

Data quality

great expectations

PROFINIT
An Amdocs Company

- Automatic check
  - before and/or after data cleaning



Data transformation

Data type issues

String

- Standardize casing
- Remove whitespaces, new lines
- Correcting typos
- Standardize encoding
- Map to type / categorical variables
- Remove stop words

Date and time

- Format
- Time zones

Data transformation

PROFINIT
An Amdocs Company

	FROM Categorical	Numeric
TO Numeric	One-hot encoding Ordinal encoding Label encoding Target mean encoding Frequency encoding Scoring	Mathematic transformations • log, Box-Cox • Standardization (z-score)
Categorical	Ordering Segmentation, clustering	Binning • Equal width • Equal frequency (quantile) • Entropy (decision tree)

One-Hot encoding

- Most common method
- **Binary Column** is created for each **Unique Category**
- If a category is present, the corresponding column is set to 1, and all other columns are set to 0.
- One column can be omitted

Binary encoding

- the columns are treated as digits of binary number

City
Praha
Plzeň
Písek
Polička

Praha	Plzeň	Písek	Polička	Binary encoding
1	0	0	0	1000
0	1	0	0	100
0	0	1	0	10
0	0	0	1	1

Label and ordinal encoding

Label encoding

- Each unique category → unique integer
- Number may be misinterpreted by ML

Ordinal encoding

- Special case when the categories can be naturally ordered

City
Praha
Plzeň
Písek
Polička

City	Value
Praha	1
Plzeň	2
Písek	3
Polička	4

City	Value
Praha	1
Plzeň	2
Polička	4
Praha	1

Label and ordinal encoding

PROFINIT
An Amdocs Company

Label encoding

- Each unique category → unique integer
- Number may be misinterpreted by ML

Ordinal encoding

- Special case when the categories can be naturally ordered

Size
S
M
L
XL

⇨

Size	Value
M	2
XL	4
S	1
S	1

Count and target encoding

Count encoding

- Category is encoded by its count (frequency)

Target encoding

- Category is encoded by its target mean

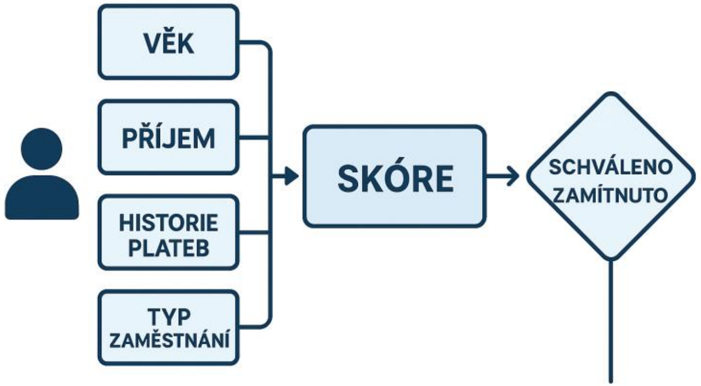
Size	Target
S	1
XL	1
S	1
S	0
M	0
L	1
L	0

Size	count	Target mean
S	3	2/3
M	1	0
L	2	½
XL	1	1

Size	Count enc.	Target enc.
S	3	2/3
XL	1	1
S	3	2/3
S	3	2/3
M	1	0
L	2	½
L	2	½

Scoring

- Simplifies complex data
- feature vector → single value
- Result of a submodel or dimensionality reduction
- Give possibility of automation
- Examples
  - Scorecards in risk departments
  - Socio-demographic score
  - Behavioral score
  - Transactional score
  - Credit score



Equal width an depth binning

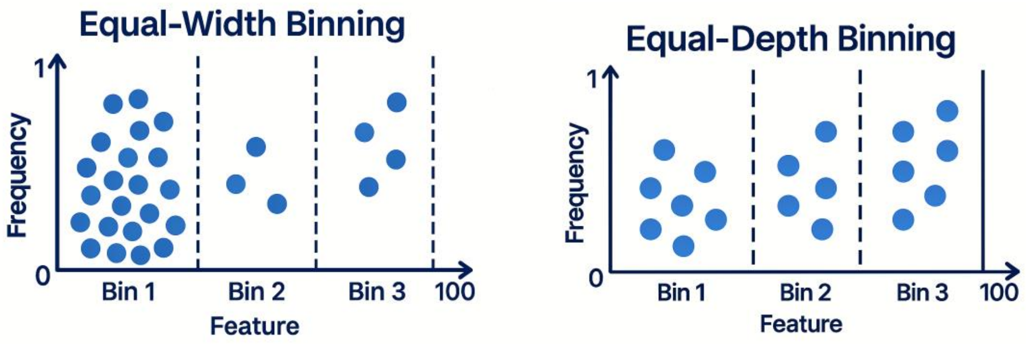
- Dividing the range into N intervals

Equal width binning

- Uniform grid
- Pros
  - Simple
  - Explainable
- Cons
  - Sensitive to outliers

Equal depth binning

- Same number of samples
- Pros
  - Suitable for skewed distributions
  - Better model stability



Entropy Binning

PROFINIT
An Amdocs Company

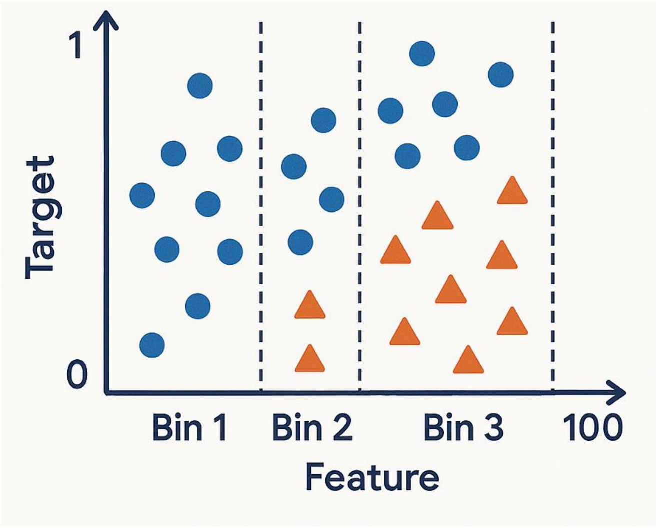
- Based on **Information Gain** from decision tree algorithms.
- Measures how well a split separates the target classes.
- Chooses bin edges that **maximize reduction in entropy**

Use

- Captures **non-linear relationships** between feature and target.
- Produces bins that are **informative for classification**.
- Often used in **credit scoring**, **churn prediction**, and **risk modeling**.

How It Works

1. Sort feature values.
2. Evaluate all possible split points.
3. Calculate entropy before and after each split.
4. Select split with **highest information gain**.
5. Repeat recursively until stopping criteria.



Mathematic transformations

- **Unit conversion**
- °F→°C,
  mpg → liters/100 km,
  €→CZK
- **Scale standardization**
- **Change of distribution shape** (normalization)
- Q-Q plot, skewness, kurtosis

Z-score

$$Z = \frac{X - \mathrm{E}[X]}{\sigma(X)}$$

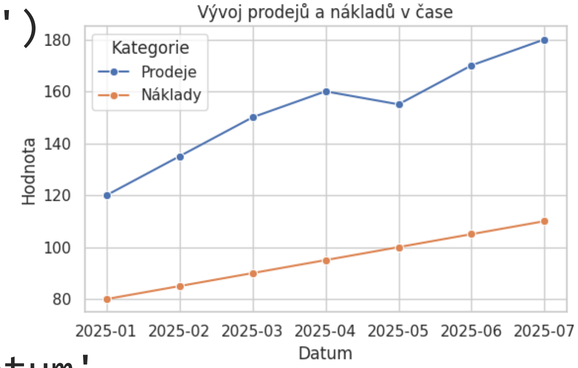
Box-Cox transformation

$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln y_i & \text{if } \lambda = 0, \end{cases}$$

Long format

- Each observation is a row
- Useful for seaborn, plotnine, plotly

```
sns.lineplot(data=long_df,
             x='Datum',
             y='Hodnota',
             hue='Kategorie',
             marker='o')
```



- To wide format

```
long_df.pivot(index='Datum',
columns='Kategorie',
values='Hodnota')
```

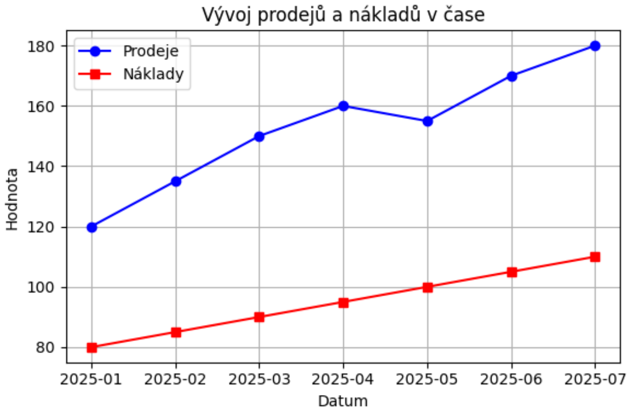
Datum	Kategorie	Hodnota
2025-01	Prodeje	120
2025-02	Prodeje	135
2025-03	Prodeje	150
2025-04	Prodeje	160
2025-05	Prodeje	155
2025-06	Prodeje	170
2025-07	Prodeje	180
2025-01	Náklady	80
2025-02	Náklady	85
2025-03	Náklady	90
2025-04	Náklady	95
2025-05	Náklady	100
2025-06	Náklady	105
2025-07	Náklady	110

Wide format

- Each variable forms a separate column
- Example: time series data, pivot tables
- Useful for mathplotlib

```
plt.plot(df_wide['Datum‘], df_wide['Sales‘],
         label='Sales‘, color='blue')
plt.plot(df_wide['Datum‘], df_wide['Náklady‘],
         label='Náklady‘, color='red')
```

Datum	Náklady	Prodeje
2025-01	80	120
2025-02	85	135
2025-03	90	150
2025-04	95	160
2025-05	100	155
2025-06	105	170
2025-07	110	180



- To long format

```
pd.melt(df_wide,
        id_vars=['Datum‘],
        value_vars=['Sales', 'Náklady‘],
        var_name='Kategorie‘,
        value_name='Hodnota')
```

Feature engineering

Feature engineering

= Using domain knowledge to extract the most relevant information in a raw data into variables to be used in an ML model.

- **Creation**
  - Combining variables to get new ones
- **Transformations**
  - To be able to use desired ML tool (e.g. numeric input only)
  - To improve model performance (therefore, depends on the ML in use)
- **Extraction**
  - text mining, clustering
- **Selection**
  - Judge which features worth to keep in model

Atomic data

- Logs (events, sessions)
- Products of client
- Relations (of people, events)

Statistical unit

- Type of behavior
- Client
- Clusters

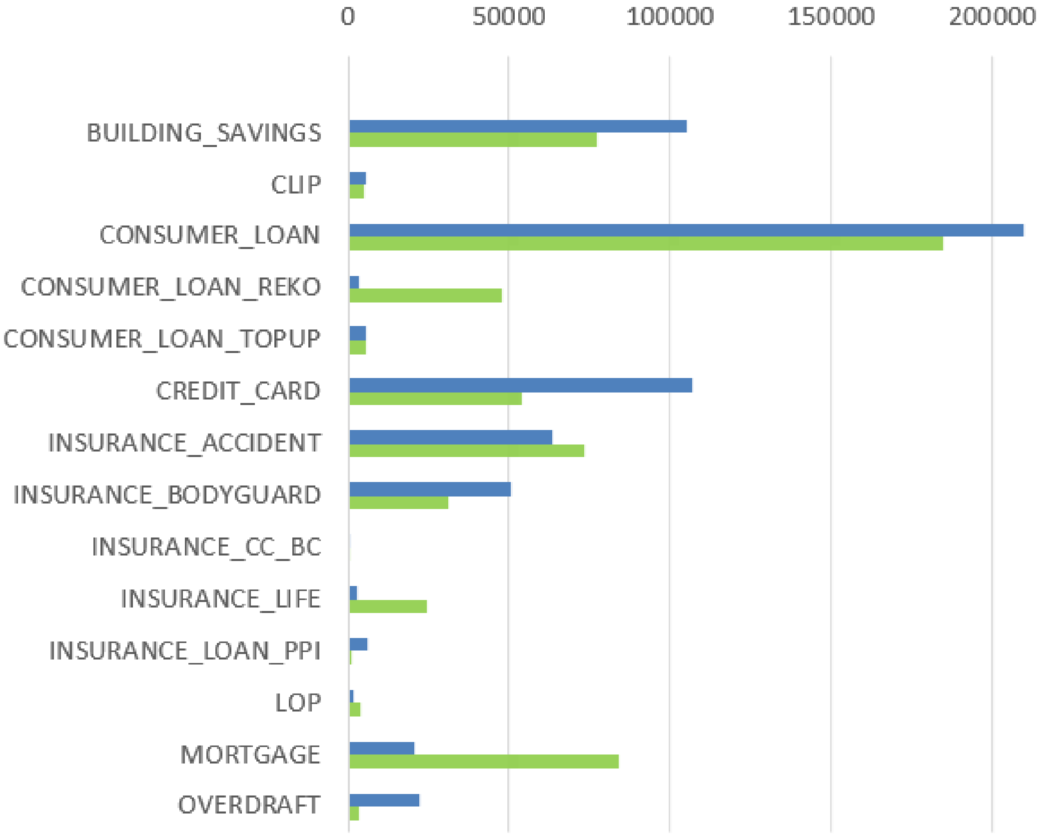
PROFINIT
An Amdocs Company

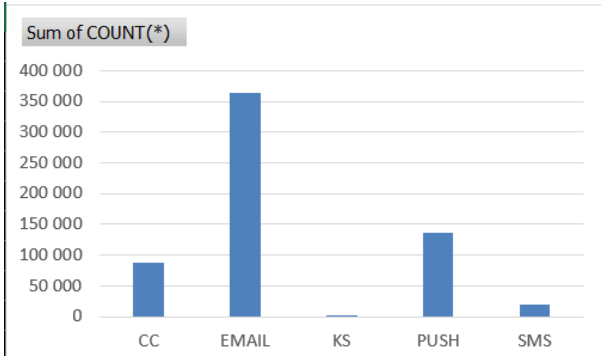
Transformations

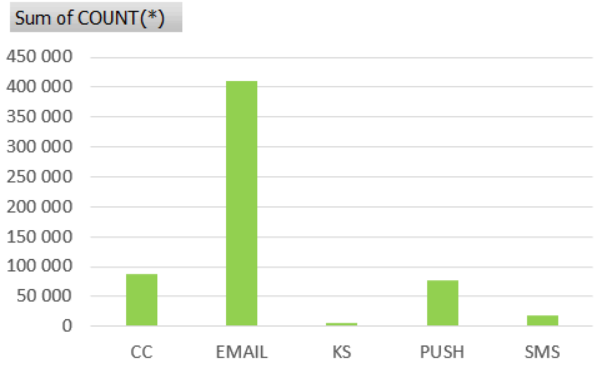
- Aggregations (min, max, count, avg, median)
- Detection of interesting points
- Clustering, dimension reduction

Campaign optimization

Expected revenue (client, product, channel) =
propensity to buy (client, product) *
product revenue (client, product) *
channel efficiency (product, channel)
\- **channel costs** (channel)







Campaign optimization

PROFINIT
An Amdocs Company

Campaigns
Campaign_id
Product_id
Customer_id
Channel_id
datetime

Sales
Product_id
Customer_id
datetime

Campaigns

- One client was contacted multiple times
- One client was contacted via multiple channels
- One client was contacted about multiple products

Sales

- One client bought multiple products
- How far from the outreach did the outreach have an effect?
- What if the client purchased a different product than the one targeted by the campaign

Channel

Campaigns
Campaign_id
Product_id
Customer_id
Channel_id
datetime

Sales
Product_id
Customer_id
datetime

Expected revenue (client, product, channel) =
propensity to buy (client, product) *
product revenue (client, product) *
channel efficiency (product, channel)
- **channel costs** (channel)

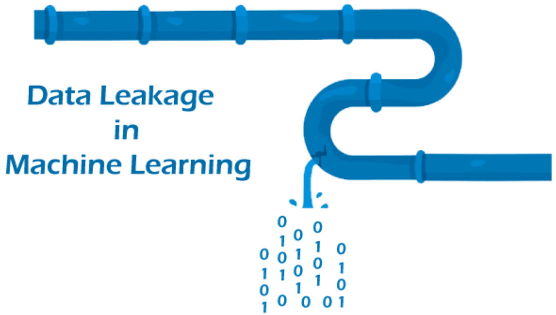
Promising features

- Last_channel_before_sell
- Last_channel_product_before_sell
- Channel_sale_contribution_per_product
- Channel_sale_contribution_per_customer

Data leakage

Data leak

- Train data are not consistent with prediction data
  - Something is unvailable
  - Something is added later
  - Something is removed
  - The policy has changed
  - Even information quality matters



- Validation data have to be shifted in time
  - Customer behavior after application, post-approval transactions
  - Loan status or outcome fields, derived features from future data
  - Internal notes or manual overrides
  - Credit bureau updates after application

Missing values

- **Missing completely at random (MCAR)**
  - During data migration from system A to system B, some data were randomly lost.
- **Missing at random (MAR)**
  - Clients with low income avoid filling it in, fearing loan rejection
  - Self-employed clients often omit income as it's harder to document
- **Missing not at random (MNAR)**
  - For high-risk clients, income is missing because they didn’t reach the part of the application where income is entered
  - Use of information about missing is data leak !
- X = income
- Y = credit risk (loan approval)