Data Preprocessing Workflow for Startup Dataset

HINIMDOU MORSIA GUITDAM

August 4, 2025

Contents

Workflow Diagram

Here is the workflow diagram representing the preprocessing steps:

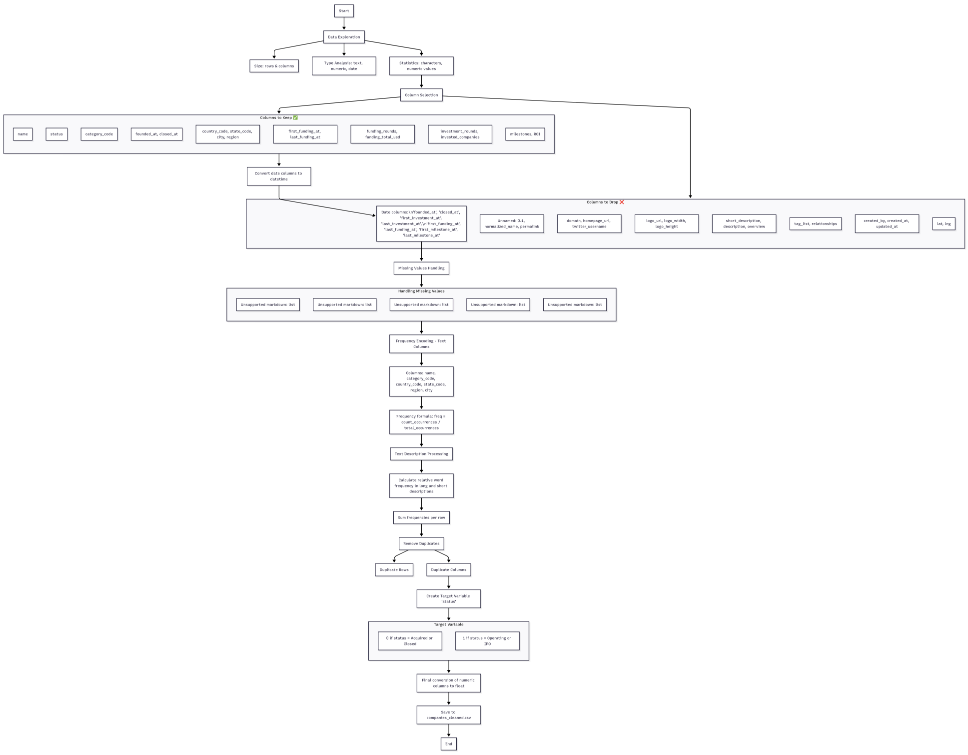

Figure 1: Data preprocessing workflow diagram

1 Data Exploration

- Inspect dataset size: number of rows and columns.
- Analyze column data types: text, numeric, date.
- Compute statistics: number of characters in text columns, numeric value distributions.

These initial exploratory steps are crucial to understand the structure and quality of the dataset. Inspecting the size provides an overview of data volume, which impacts computational resources and processing time. Analyzing column data types ensures proper handling of variables, guiding the choice of preprocessing methods. Computing basic statistics, such as text length distributions and numeric value summaries, helps detect anomalies, outliers, or inconsistencies that could affect model performance. This foundational analysis sets the stage for informed and effective data cleaning and transformation.

2 Column Selection

2.1 Columns to Keep

Column	Justification
`name`	Main identifier for each startup, essential for deduplication and tracking.
`status`	Target variable (Operating, IPO, Acquired, Closed) for supervised learning.
`category_code`	Business sector (e.g., fintech, edtech); useful categorical feature.
`founded_at`	Founding date; allows calculation of startup age and trend analysis.
`closed_at`	Provides info on non-active startups.
`country_code`, `state_code`, `city`, `region`	Geographic context potentially influencing success.
`first_funding_at`, `last_funding_at`	Funding timeline and history.
`funding_rounds`	Number of funding rounds; indicator of investor interest.
`funding_total_usd`	Total funding amount; key financial metric.
`investment_rounds`, `invested_companies`	Measures investment activity and investor behavior.
`milestones`	Milestones reached, indicating development stages.
`ROI`	Return on Investment, useful as feature or secondary target if available.

2.2 Columns to Drop

Column	**Justification**	**Example**
id, entity_id, parent_id, entity_type	Internal technical IDs, irrelevant for modeling.	id = 123456, no business meaning
Unnamed: 0.1	Redundant index column, often auto-generated.	Values like 0, 1, 2, 3 . . . duplicate DataFrame index
normalized_name, permalink	Duplicate info of name column.	normalized_name = "startup-x" duplicates name
domain, homepage_url, twitter_username	Web/social media fields, not useful for analysis.	homepage_url = "http://www.example.com"
logo_url, logo_width, logo_height	Visual/media info not exploited in this project.	logo_url = "http://logo.example.com/
short_description, description, overview, tag_list	Long text fields, noisy for tabular analysis.	description = "A fast-growing tech startup..."
relationships	Nested/complex data needing advanced parsing.	JSON-like data with investors and partners details
created_by, created_at, updated_at	Dataset metadata, not business-relevant.	created_at = "2021-05-10T12:34:56Z"
lat, lng	Often incomplete/imprecise coordinates, requiring external enrichment.	lat = NaN, lng = NaN for many startups

3 Datetime Conversion

Convert the following columns to datetime format:

- founded_at, closed_at
- first_investment_at, last_investment_at
- first_funding_at, last_funding_at
- first_milestone_at, last_milestone_at

For each pair of dates, sort and fill missing values by median date, then compute difference in days.

4 Missing Values Treatment

- For categorical/text columns with few missing values, fill missing entries with the mode.
- For numeric columns with few missing values, fill with median or mean depending on normality.
- For description and short_description, fill missing with "Not provided".
- For country_code, state_code, city, and region, fill missing with "Unknown".
- For date columns, sort and replace missing values with median date, then calculate difference in days.

5 Frequency Encoding for Categorical Columns

Apply frequency encoding to the following columns:

`name`, `category_code`, `country_code`, `state_code`, `region`, `city`

The formula for frequency encoding is:

$$\text{freq} = \frac{\text{count of the category}}{\text{total number of rows}}$$

6 Text Description Processing

Let

$$D = \{d_1, d_2, \ldots, d_N\}$$

be the set of long descriptions and

$$S = \{s_1, s_2, \ldots, s_N\}$$

the set of short descriptions.

Define vocabularies:

$$V_D = \{w_1, w_2, \ldots, w_{M_D}\}, \quad V_S = \{u_1, u_2, \ldots, u_{M_S}\}$$

Relative frequency of word $w \in V_D$:

$$f_D(w) = \frac{\text{number of occurrences of } w \text{ in } D}{\sum_{w' \in V_D} \text{number of occurrences of } w'}$$

Relative frequency of word $u \in V_S$:

$$f_S(u) = \frac{\text{number of occurrences of } u \text{ in } S}{\sum_{u' \in V_S} \text{number of occurrences of } u'}$$

For each description d_i containing words $\{w_{i1}, \ldots, w_{iK_i}\}$:

$$\text{desc\_freq\_sum}_i = \sum_{k=1}^{K_i} f_D(w_{ik})$$

For each short description s_i containing words $\{u_{i1}, \ldots, u_{iL_i}\}$:

$$\text{short\_desc\_freq\_sum}_i = \sum_{l=1}^{L_i} f_S(u_{il})$$

These sums are used as compact numerical features.

This approach is chosen to avoid increasing the dimensionality of the dataset as would happen with techniques such as one-hot encoding or other high-dimensional encoding methods, thus keeping the feature space manageable and reducing computational complexity.

7 Duplicate Removal

Remove duplicate rows and columns to ensure data integrity.

8 Target Variable Creation

Define a binary target variable from `status` column:

$$\text{target} = \begin{cases} 0 & \text{if status} = \text{Acquired or Closed} \\ 1 & \text{if status} = \text{Operating or IPO} \end{cases}$$

9 Final Data Conversion

Convert all numeric columns to `float` type to maintain homogeneity.

10 Saving Cleaned Dataset

Save the cleaned and processed dataset to `companies_cleaned.csv`.

Code snippet for final conversion and saving:

```
df_cleaned = df_cleaned.astype(float)
df_cleaned.to_csv('companies_cleaned.csv', index=False)
```

Conclusion

In this report, we presented a comprehensive data preprocessing workflow for the startup dataset. The process involved cleaning and preparing the data by selecting relevant columns, handling missing values, converting date fields, and applying suitable encoding techniques. The use of compact numerical representations for textual descriptions helped to limit the dataset's dimensionality, avoiding issues commonly associated with traditional methods like one-hot encoding.

These preprocessing steps are crucial to ensure high-quality input data and to improve the performance of machine learning models that will be developed in subsequent phases. The cleaned, consistent, and ready-to-use dataset will be a key asset for the next stage of predictive analysis of startup acquisition status.

Thank you for your attention !