Feature Engineering Part 1: Comprehensive Review

Based on DSC 208R – Data Management for Analytics by Arun Kumar, UC San Diego

June 2025

Motivation and What is Feature Engineering?

The Importance of Features in ML

The performance of Machine Learning (ML) models heavily depends on the quality of the features used. Bad features can lead to poor model performance, regardless of the sophistication of the ML algorithm. As famously stated by Andrew Ng, "coming up with features is difficult, time-consuming, requires expert knowledge. 'Applied machine learning' is basically feature engineering."

What is Feature Engineering?

Feature engineering is the process of transforming raw data into features that are suitable for building effective ML models. It involves:

- Selecting, transforming, or creating variables from raw data.
- Making data ready for ML algorithms.
- Improving model performance by providing better inputs.

It is distinct from Feature Selection, which is the process of selecting a subset of relevant features from an existing set.

Feature Types and Techniques

Features can generally be categorized into numerical, categorical, and textual, each requiring specific engineering techniques.

Numerical Features

These are quantitative variables that can be measured numerically.

1. **Standardization (Z-score Normalization)**: Transforms data to have a mean of 0 and a standard deviation of 1.
$$X' = \frac{X - \mu}{\sigma}$$
2. **Min-Max Scaling (Normalization to a range)**: Scales data to a fixed range, typically [0, 1].
$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$
3. **Binning (Discretization)**: Converts numerical features into categorical ones by grouping values into "bins."
    - *Fixed-width binning*: Uses predefined boundaries.
    - *Quantile binning*: Creates bins with an equal number of data points.
4. **Log Transform**: Used for skewed distributions to make them more Gaussian-like, especially useful for highly skewed positive data.

5. **Square Root Transform**: Similar to log transform, useful for count data or positive skewed data.
6. **Power Transforms (e.g., Box-Cox, Yeo-Johnson)**: More general transformations to handle various skewnesses.

Categorical Features

These are qualitative variables representing categories or labels.

1. **One-Hot Encoding**: Converts categorical variables into a numerical format where each category becomes a new binary (0/1) feature. This is suitable for nominal (unordered) categories.
2. **Label Encoding**: Assigns a unique integer to each category. This is suitable for ordinal (ordered) categories but can imply a false sense of order for nominal data if not handled carefully.
3. **Feature Hashing**: A technique to convert categorical features into numerical ones, often used for high-cardinality categorical variables. It hashes categories into a fixed-size vector. This avoids increasing dimensionality significantly and uses less memory than one-hot encoding but can suffer from hash collisions.

Textual Features

These are features derived from raw text data.

1. **Bag of Words (BoW)**: Represents a document as a collection of words, ignoring grammar and word order, but keeping multiplicity.
2. **TF-IDF (Term Frequency-Inverse Document Frequency)**: A numerical statistic that reflects how important a word is to a document in a collection or corpus. It weights words based on their frequency in a document (TF) and their rarity across the entire corpus (IDF).
3. **Word Embeddings (e.g., Word2Vec, GloVe)**: Dense vector representations of words that capture semantic relationships and context. These are learned from large text corpora and represent words in a continuous vector space.

Advanced Feature Engineering

Beyond basic transformations, feature engineering can involve more complex techniques:

- **Interaction Features**: Creating new features by combining existing ones (e.g., multiplying two numerical features, or concatenating two categorical features). This can capture relationships that single features might miss.
- **Time-Based Features**: Extracting features from datetime columns, such as year, month, day of week, hour, or deriving features like "time since last event."
- **Aggregation Features**: Summarizing information from related rows or tables (e.g., average purchase amount for a customer).
- **Geospatial Features**: Extracting meaningful information from location data, such as distance to a landmark or density of points in an area.

The choice of feature engineering techniques is highly dependent on the domain, data characteristics, and the specific ML problem being solved. It often involves a deep understanding of the data and iterative experimentation.