ML Concepts - Representation

*Feature Engineering

1. **Feature Engineering:** It involves transforming raw data into a feature vector, which is a set of floating-point values used in machine learning models. This process is crucial and can be time-consuming.
2. **Mapping Numeric Values:** Integer and floating-point data don't require special encoding since they can be directly multiplied by model weights. For example, converting the raw integer value 6 to the feature value 6.0 is straightforward.
3. **Mapping Categorical Values:** Categorical features have discrete, non-numeric values. These values need to be converted to numeric form for modeling purposes. This is done by assigning a unique numeric value to each category. An "out-of-vocabulary" (OOV) category is used for values not present in the dataset.
4. **One-Hot Encoding:** To address the constraints of direct indexing, a binary vector is created for each categorical feature. The vector has a length equal to the number of elements in the vocabulary. This representation is called one-hot encoding when a single value is 1, and multi-hot encoding when multiple values are 1. It allows for flexibility in learning different weights for each category.
5. **Sparse Representation:** When dealing with a large number of unique values (e.g., 1,000,000 street names), explicitly creating a binary vector for each element can be inefficient in terms of storage and computation. Instead, a sparse representation is used, where only nonzero values are stored. Despite this, an independent model weight is still learned for each feature value.

What are the qualities of Good Features?

1. **Avoid Rarely Used Discrete Feature Values:**
   - Good feature values should appear more than 5 times in a dataset for effective learning.
   - Many examples with the same discrete value help the model understand its relationship with the label.
   - Rarely occurring feature values do not contribute effectively to predictions.
2. **Prefer Clear and Obvious Meanings:**
   - Features should have clear and easily understandable meanings.
   - Descriptive feature names are important for clarity.
   - Unclear or cryptic feature values can hinder understanding.
3. **Avoid Mixing "Magic" Values with Actual Data:**
   - Floating-point features should not contain peculiar or out-of-range values.
   - Using specific values (e.g., -1) to represent missing or undefined data is discouraged.
   - Instead, create a Boolean feature to indicate the presence or absence of the original feature.
4. **Account for Upstream Instability:**

- Feature definitions should remain consistent over time.
- Values that may change due to external factors or upstream processes should be handled carefully.
- Avoid using values that are subject to frequent changes as features.