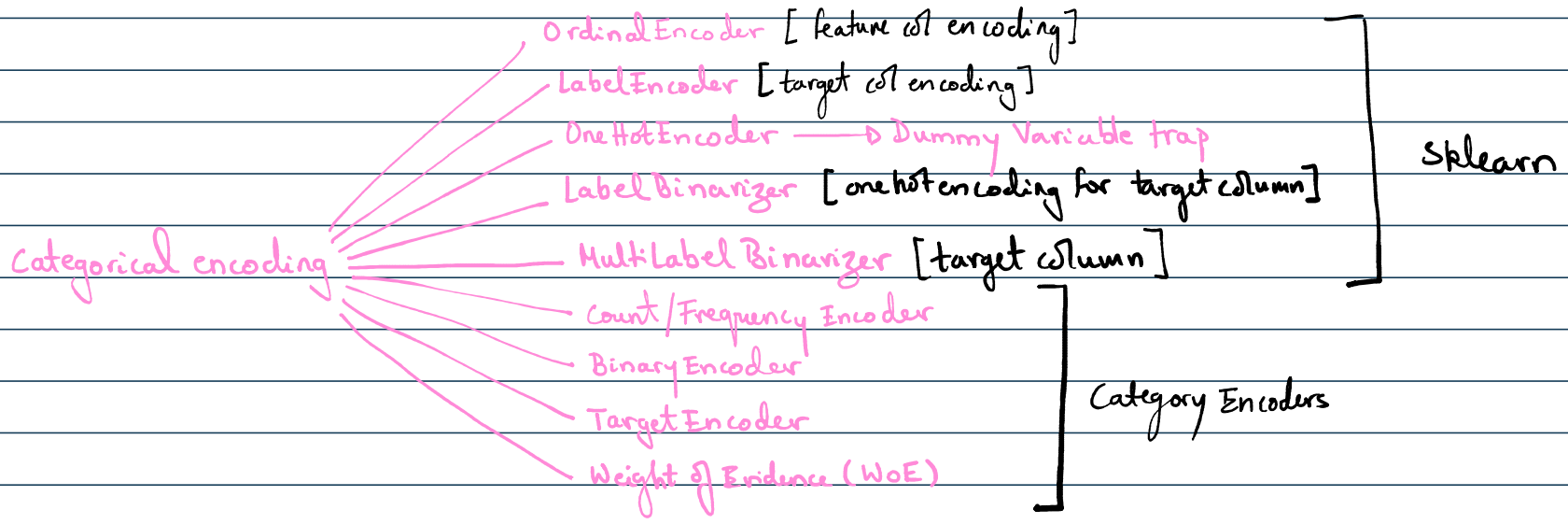
Feature Encoding

Sunday 12 May 2024 9:42 PM



Discretization → Information loss is a common problem for all binning techniques

- Custom Binning [Using Domain knowledge]
- Uniform/equal width binning

$$\text{Bin Width} = \frac{\text{Max} - \text{Min}}{\#\text{bins}}$$ → Tricky

KBinsDiscretizer [sklearn] — n_bins, encode, strategy → uniform

→ Simple

→ Uniform coverage

→ Sensitive to outliers → Resulting in many empty bins

→ Not adaptive i.e. Doesn't study the distribution of the data

→ same algorithm regardless of underlying distribution

→ ↑bins → noise ↓bins → Loss of info

use when — evenly distributed data; can act as a baseline

- Quantile/equal frequency binning

#bins → (#bin - 1) quantiles

e.g. 4bins → 100/4 = 25 → 0 - 25% - 50% - 75% - 100%

e.g. 4 bins → 100/4 = 25 → 0 - 25% - 50% - 75% - 100%

→ Bin width may differ but frequency in each bin is effectively same.

KBinsDiscretizer [sklearn]
- n_bins → Tricky
- encode
- strategy → quantile

→ able to handle outliers
→ Handles skewed distributions

→ Difficulty in bins interpretation
→ True info about the data distribution is lost
→ Computationally expensive → sorting

K-Means Binning

→ Binning with the help of clustering
→ Works just like k-means clustering
→ Midpoints of cluster centroids act as the bin boundaries

KBinsDiscretizer [sklearn]
- n_bins / clusters
  - elbow
  - silhouette
  → help in finding the # of bins
- encode
- strategy → kmeans

→ Adaptive
→ Minimizes within-bin variance

→ sensitive to outliers
→ Computationally expensive.
→ Assumption of similar size & density of clusters
→ Interpretability

Binarization / Threshold Binning

Num → 0 or 1 based on a threshold

Binarizer
- NLP
- Image processing

→ losing a lot of info

→ Losing a lot of info

→ Limited use

- Decision Tree Based Binning

→ Supervised Binning

→ feature vs target → DT → Threshold

(-2) → leaf node

↳ plot tree

→ No need to specify # of bins

→ Data leakage ?

Feature-engine library