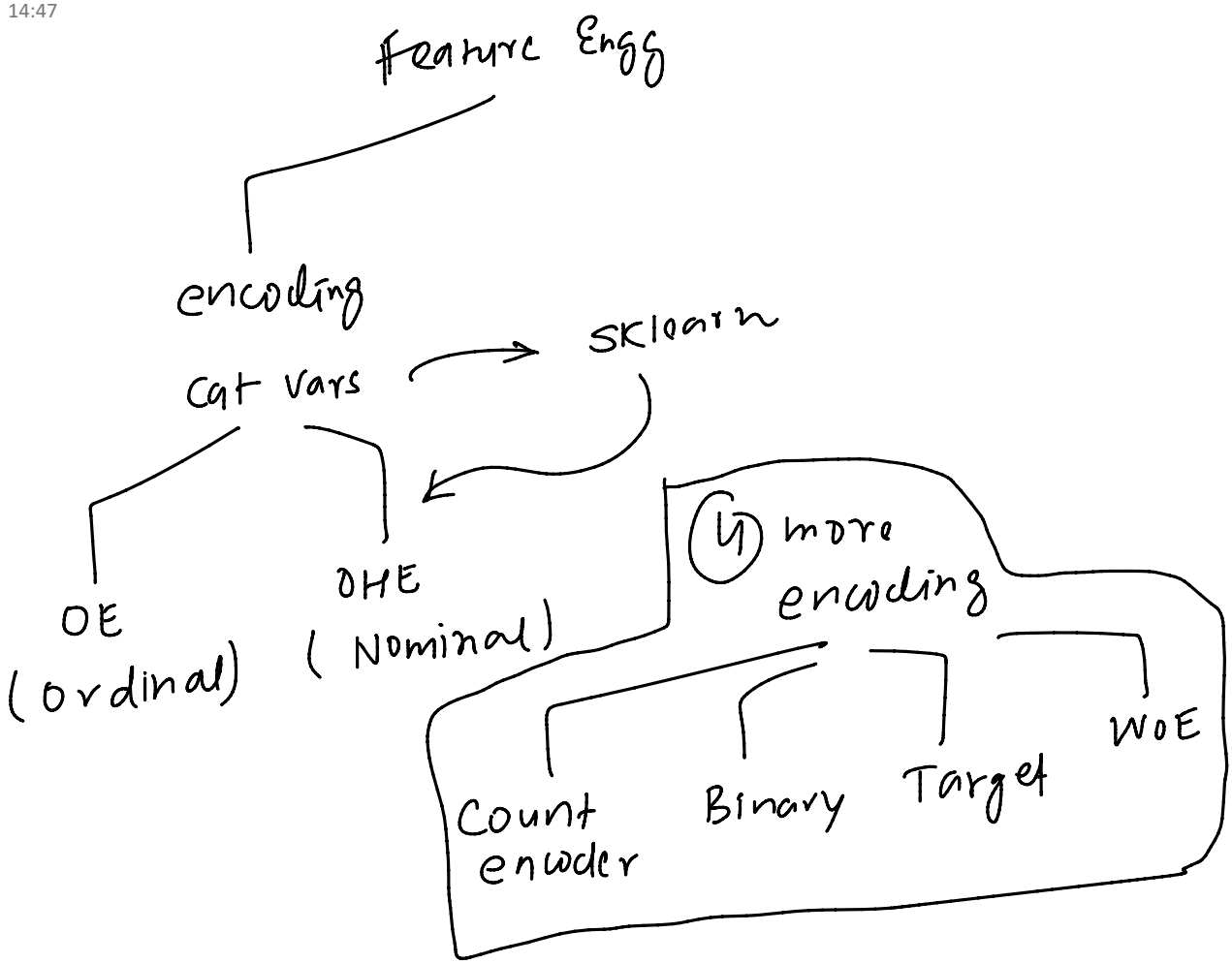
Recap

08 February 2024 14:47



Count and Frequency Encoder

08 February 2024 14:48

A count encoder is a technique used in the pre-processing of categorical data for machine learning models. It transforms each categorical value into a number representing the frequency of that value in the dataset. This method is particularly useful for converting categorical variables into a format that can be more easily used by various machine learning algorithms, which typically require numerical input.

Pet ID	Breed
1	Labrador
2	Beagle
3	Beagle
4	Labrador
5	Siamese

Pet ID	Breed (Count Encoded)
1	2
2	2
3	2
4	2
5	1

A frequency encoder is a technique similar to count encoding, used in the pre-processing of categorical data for machine learning models. While both methods transform categorical values based on their occurrence in the dataset, frequency encoding specifically converts each categorical value into a number representing the relative frequency or proportion of that value in the dataset, rather than the absolute count.

Pet ID	Breed
1	Labrador
2	Beagle
3	Beagle
4	Labrador
5	Siamese

Pet ID	Breed (Frequency Encoded)
1	0.4
2	0.4
3	0.4
4	0.4
5	0.2

Code

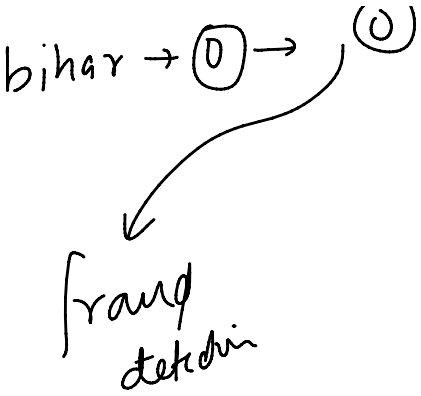
Advantages

1. Simple and Efficient
2. Reduces dimensionality
3. Captures frequency info

Disadvantages

1. Collision
2. No info on categorical relationship
3. Handling of unseen categories
4. Data with skewed distributions(some categories having much more frequency) -> underappreciates the contribution of rare categories

2. No info on categorical relationship
3. Handling of unseen categories
4. Data with skewed distributions(some categories having much more frequency) -> underappreciates the contribution of rare categories



Potential Use-cases

1. High Cardinality
2. Imbalanced Dataset
3. Good for tree-based algo

Binary Encoder →

08 February 2024 14:48

maha
0 →
1 →

Binary encoding is a technique used for converting categorical data into a numerical format that machine learning algorithms can work with. Unlike one-hot encoding, which creates a new binary column for each level of a categorical variable, binary encoding first converts the categories into ordinal numbers, then transforms those ordinal numbers into binary code, and finally splits the bits of the binary code into separate columns. This method is particularly useful for dealing with high cardinality categorical data, as it significantly reduces the dimensionality compared to one-hot encoding.

OHE ← 8 cates → 4 cols

Item	Fruit	Bit1	Bit2	Bit3	Bit4
Item1	Apple	0	0	0	1
Item2	Banana	0	0	1	0
Item3	Cherry	0	0	1	1
Item4	Date	0	1	0	0
Item5	Elderberry	0	1	0	1
Item6	Fig	0	1	1	0
Item7	Grape	0	1	1	1
Item8	Honeydew	1	0	0	0

8 cols

OHE
high cardinal
Sector
1
108 → 120 category
↓
[1 col → 120 cols]
↓
feature space expand

Advantages

1. Reduced Dimensionality (Memory Efficient)
2. Handling of new category

Disadvantages

1. Loss of interpretation

Use-Cases

1. High Cardinality features
2. Tree-based models

OHE → 4 cols
8 cate

State
Maha → 1 → 0001
Guj → 2 → 0010
Goa → 3 → 0011
Sikkim → 4 → 0100

bit_0	bit_1	bit_2	bit_3
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0

8 states

Bihar → 9 →

Target Encoder / mean encoder

08 February 2024 14:48

Target encoding, also known as mean encoding, is a technique for encoding categorical variables where each category is replaced with the mean value of the target variable for that category. It's a form of feature engineering that can help machine learning models understand and leverage the relationship between categorical features and the target variable more effectively.

Color	Encoded Color	Target
Red	0.5	0
Red	0.5	1
Green	0	0
Blue	1	1
Blue	1	1

The Main Problem - Overfitting

Solution - Don't just rely on the category mean, create a blend of category mean and global mean.

$$\text{Target encoding} = \lambda \times \text{Category Mean} + (1-\lambda) \times \text{Global Mean}$$

$$\lambda = \frac{1}{1+\exp\left(-\left(\frac{n-\text{min\_samples\_leaf}}{\text{smoothing}}\right)\right)}$$

1. Objective: The goal of target encoding, particularly when incorporating smoothing, is to calculate a target mean for each category that takes into account both the global mean of the target variable across all categories and the specific mean of the target variable for that category. This approach aims to encode categorical variables in a way that reflects their relationship with the target variable more accurately.

2. Balancing with λ: To balance the influence of the global mean and the category-specific mean, a smoothing factor λ is used. The value of λ determines how much weight is given to the category mean versus the global mean for each category's encoded value.

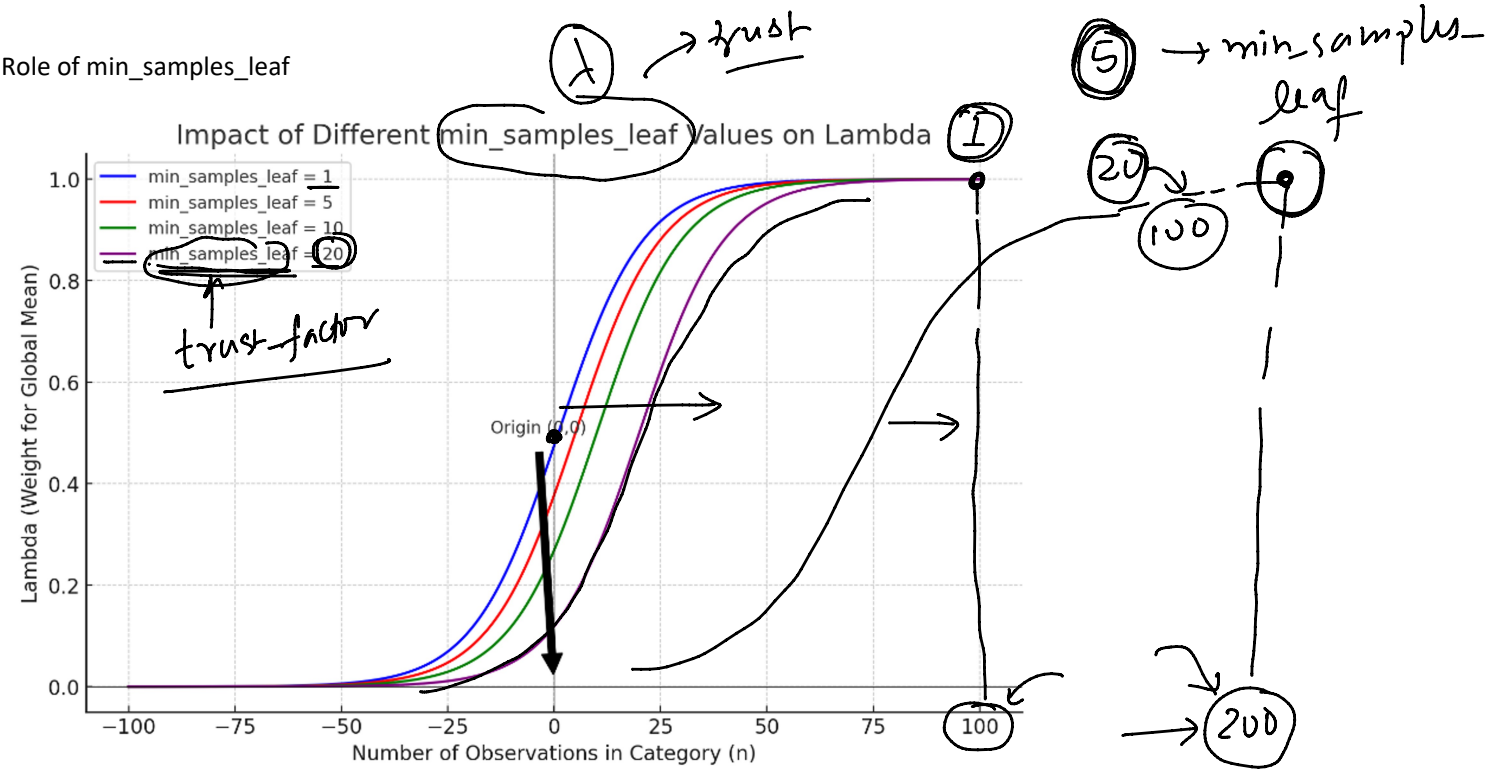
3. Dependency on n: λ is dependent on n, the number of observations (or rows) in that category. The idea is that categories with a larger number of observations can rely more on their own mean (category-specific mean) because it's considered more reliable due to the larger sample size. Conversely, categories with fewer observations should rely more on the global mean to avoid overfitting to potentially noisy estimates.

4. Sigmoid Function for Probabilistic Interpretation: The use of a sigmoid function to represent the relationship between λ and n introduces a smooth, probabilistic transition from reliance on the global mean to reliance on the category mean. The sigmoid function ensures that as n increases, λ adjusts in a non-linear and bounded manner, providing a controlled way to shift

the weighting from the global mean towards the category-specific mean.

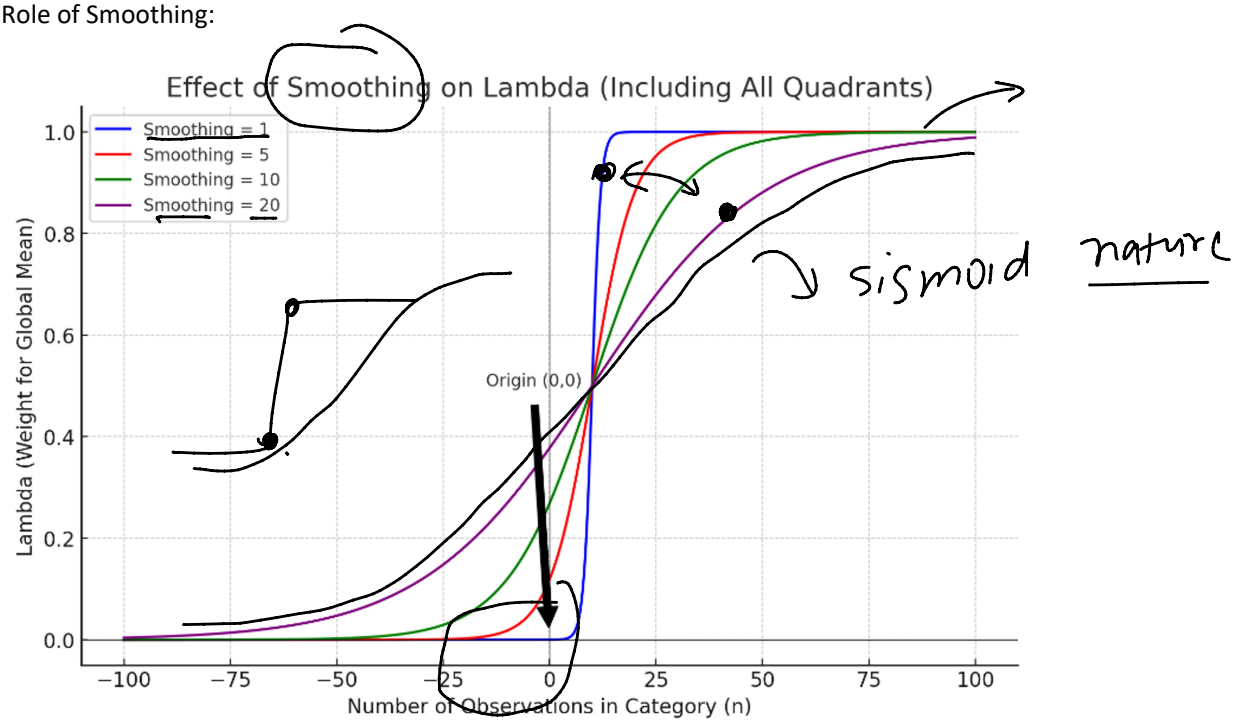
The sigmoid function's characteristics make it an ideal choice for this purpose, as it smoothly transitions between 0 and 1, allowing λ to adjust in a way that reflects the increasing reliability of the category mean with more observations, while also ensuring that the transition is gradual and bounded, reducing the risk of sudden shifts in encoding based on the sample size.

Role of min_samples_leaf



This geometrically means that the function becomes more conservative, requiring more evidence (in the form of a larger number of observations) before moving away from the global mean.

Role of Smoothing:



Geometrically, the smoothing factor in target encoding can be seen as controlling the "sharpness" or "smoothness" of the transition in the sigmoid curve that adjusts λ. It fine-tunes how responsive the encoding is to the number of observations in each category, allowing for a more nuanced balance between capturing category-specific effects and maintaining generalizability to avoid overfitting.

Advantages:

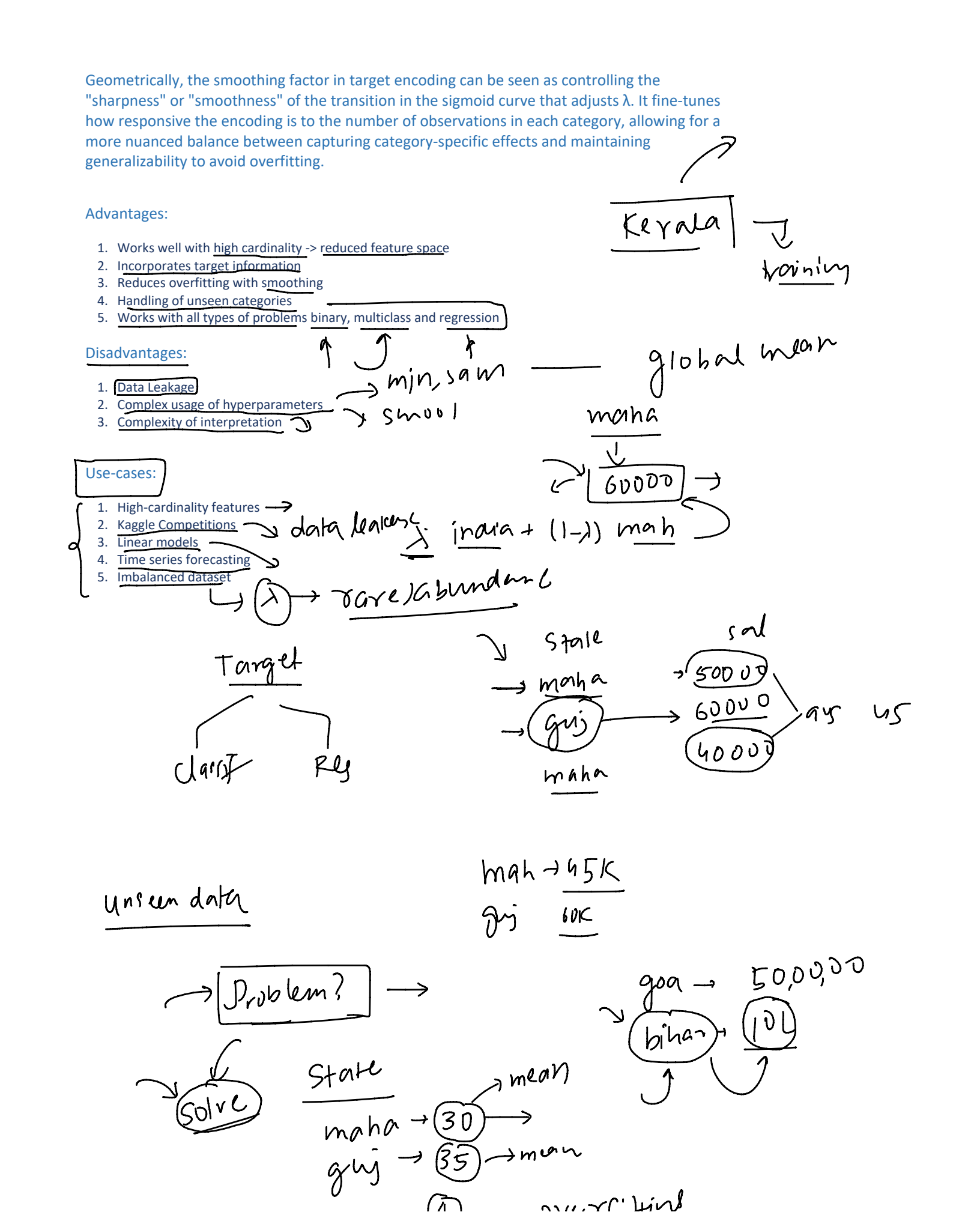
1. Works well with high cardinality -> reduced feature space
2. Incorporates target information
3. Reduces overfitting with smoothing
4. Handling of unseen categories
5. Works with all types of problems binary, multiclass and regression

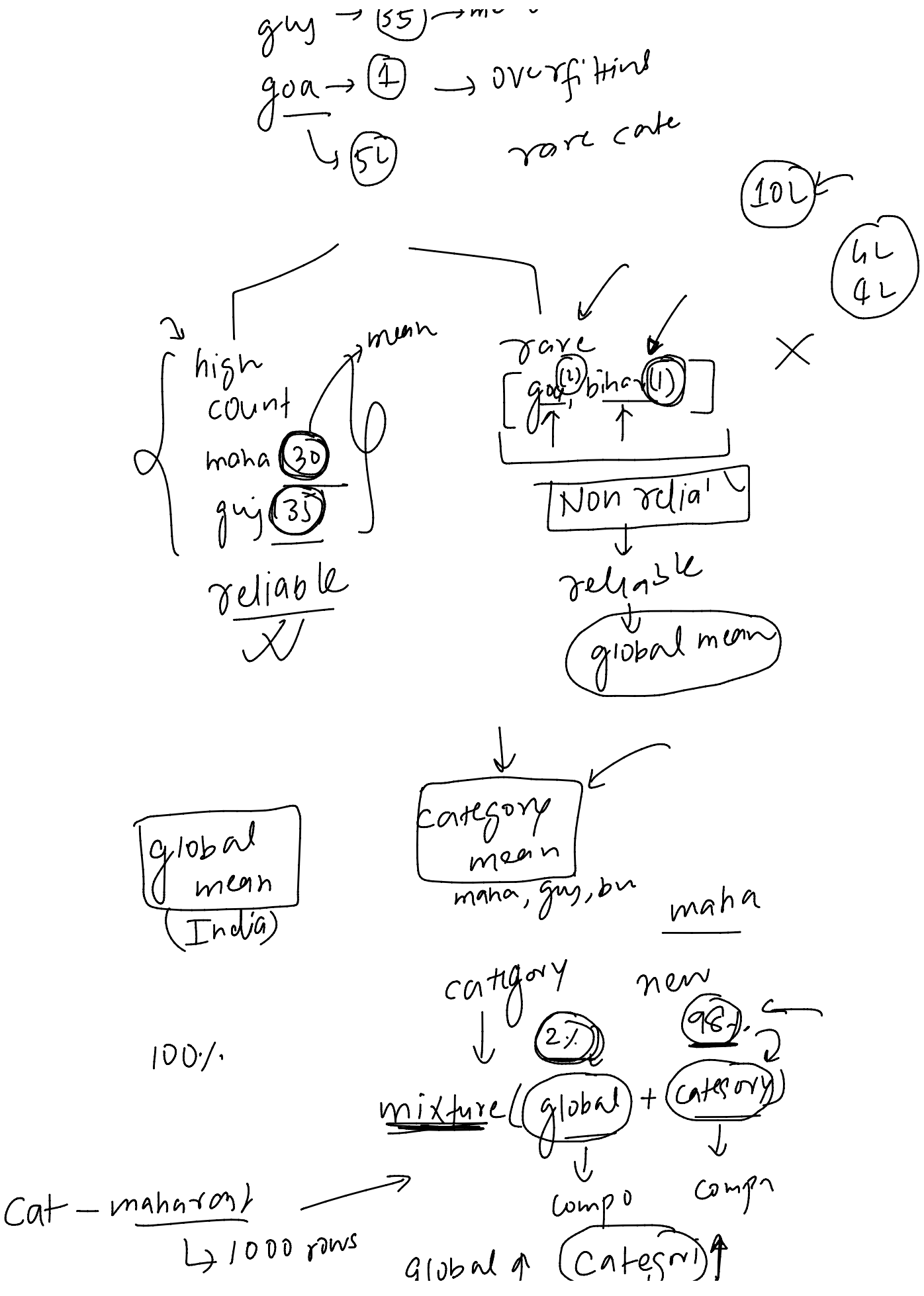
Disadvantages:

1. Data Leakage
2. Complex usage of hyperparameters
3. Complexity of interpretation

Use-cases:

1. High-cardinality features
2. Kaggle Competitions
3. Linear models
4. Time series forecasting
5. Imbalanced dataset




guj → 55 → mi
goa → 1 → overfitting
51
rare cate
10L
4L
4L
high count
mean
maha 30
guj 35
reliable
rare
goa 1
biha 1
Non relia'
reliable
global mean
global mean (India)
category mean
maha, guj, bu
maha
category
new
96
2%
100%
mixture
global
+
category
compo
compo
Cat – maharas
1000 rows
Global
Categ

↳ 1000 rows

global ↑ Categ ↑

98%
global ↑

2%
Cater ↑

rare bihar
↳ (5)

(λ) → (λ) ↑ global

exam → score (λ)
↑
hours, iq

(λ) → f(x)

y → f(x)

λ = y = f(hours, iq)

state
man → λ
bih → λ
guj → λ
goa → λ

(x) → ?

(λ) ↑ cat mean ↑
(λ) ↓ global mean

num of occurs → maha → (λ)
↓
5000

bihar → λ
↓
(5)

→ num of times

$\lambda = f(n)$ ← num of times the cat has occured in data

$\boxed{\lambda = f(n)}$

mathematical $-\infty \longleftarrow n \longrightarrow \infty$

$0-1$

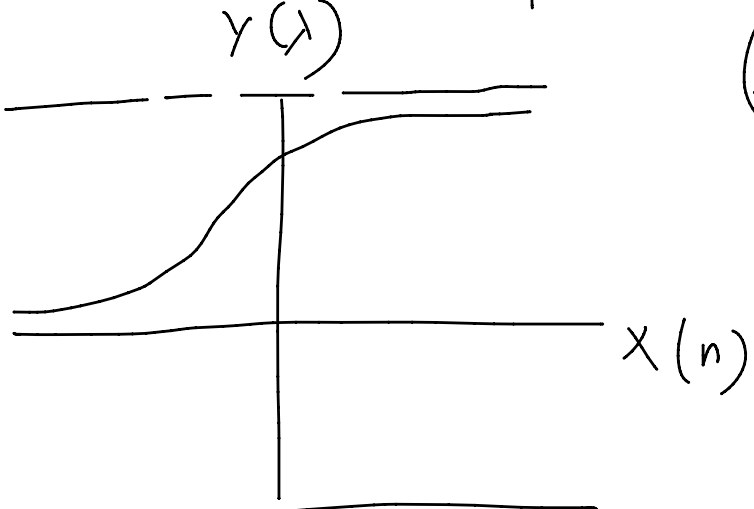
$\lambda = \boxed{0-1}$

$n \uparrow \quad \lambda \uparrow$

$n \downarrow \quad \lambda \downarrow$

λ probabistic

$\times \quad \boxed{Y \to \begin{matrix} 0 \\ 1 \\ \frac{1}{4} \end{matrix}}$



(λ)

sig(n

$$\boxed{\lambda = \frac{1}{1+e^{-n}}}$$

(λ) λ_{mh}

$$(\lambda) = \frac{1}{1+e^{-5000}}$$

state

mh

gj

mh

sala

—

—

—

$\lambda = 1$

$(1-\lambda) = 0$

bi

S –

(λ)

$$\lambda_{bh} = \frac{1}{1+e^{-5}} = 0.99$$

$\lambda = 0.99$ $\quad (1-\lambda) = (0.01)$

↙ ↘

cow bihar → cat-

Weight of Evidence

08 February 2024 14:48

binary classification

encoding → finance → cat → number

Weight of Evidence (WoE) encoding is a powerful technique used primarily in the domain of credit risk modeling and other financial analytics, but its utility extends across various fields where predictive modeling is employed. WoE encoding transforms categorical variables into a continuous scale, representing the logarithmic ratio of the distribution of "good" outcomes to the distribution of "bad" outcomes within each category. This method is particularly useful for binary classification problems.

cat → nums

Formula for Weight of Evidence

The Weight of Evidence for a category is calculated using the formula:

$$\text{WoE} = \ln\left(\frac{\text{Distribution of Good}}{\text{Distribution of Bad}}\right)$$

Where:

- **Distribution of Good** refers to the proportion of positive outcomes within the category.
- **Distribution of Bad** refers to the proportion of negative outcomes within the category.
- ln denotes the natural logarithm.

banking → repay / default

WoE → target ↑↓ monotonic ↓↑

ApplicationID	EmploymentStatus	LoanOutcome
1	Employed	Good
2	Unemployed	Bad
3	Employed	Bad
4	Student	Good
5	Employed	Good
6	Unemployed	Bad
7	Student	Bad
8	Employed	Good

nums

1.09 / -7.60 / 1.09 / 0 / 1.09 / -7.60 / 0 / 1.09

kid / senior citizen → 0.0

good → 4

bad → 4

Employed: $\ln\left(\frac{3/4}{1/4}\right) = \ln(3) = 1.09$ (good (3), bad (1))

Unemployed: $\ln\left(\frac{0.001/4}{2/4}\right) = \ln(0) = \text{undefined} \rightarrow -7.60$ (good (1), bad (2))

student: $\ln\left(\frac{1/4}{1/4}\right) = \ln(1) = 0$ (good (1), bad (1))

Advantages

1. Captures the predictive power -> monotonic relationship with the target
2. Can handle high cardinality
3. Handles rare categories well
4. Good interpretation

Disadvantages

1. Not suitable for multiclass classification and regression task.
2. Division by 0
3. Handling unseen categories → 0.0
4. Data Leakage

Use-cases

1. Credit risk modelling/fraud detection/churn prediction/customer segmentation/insurance claim prediction

Words of Wisdom

14 February 2024 19:22

One-Hot Encoder (OHE)

- Advice: Use for low cardinality features where the number of unique categories is small. Ideal for linear models and neural networks where preserving the distinction between categories without implying order is crucial. Watch out for dimensionality explosion in datasets with many categories.

Ordinal Encoder (OE)

- Advice: Best when the categorical variable has a natural, meaningful order (e.g., rating levels, educational status). Ensure your model can appropriately handle the imposed ordinality. Not suitable for nominal data where no such order exists.

Count Encoder

- Advice: Effective for capturing the frequency signal of categories, which can be particularly informative in large datasets. However, be cautious with rare categories; their low counts might lead to misleading interpretations. Pair with other encodings or use in ensemble models to mitigate this.

Binary Encoder

- Advice: A go-to for medium to high cardinality features where one-hot encoding would be impractical. It efficiently reduces dimensionality while preserving more information than simple ordinal encoding. Ensure binary patterns are meaningful for your model.

Target Encoder

- Advice: Highly useful when the relationship between the category and the target is strong and direct. Essential for complex models where capturing nuanced patterns is key. Always use smoothing and cross-validation to prevent overfitting and leakage.

Weight of Evidence (WoE)

- Advice: Primarily shines in binary classification tasks, especially in risk and financial domains. It transforms categories into a measure of predictive power, making it invaluable for interpretability and model performance. Ensure the target is binary and handle categories with no events carefully to avoid infinite values.

General Wisdom:

- Understand Your Data: The effectiveness of each encoding technique is heavily dependent on the nature of your categorical data and the specific problem you're solving.
- Experimentation is Key: There's rarely a one-size-fits-all solution. Experiment with different encoders and validate their impact on your model's performance.
- Guard Against Leakage: Techniques that use target information (like Target Encoder and WoE) require careful cross-validation strategies to avoid leaking target information into your training process.
- Balance Complexity and Interpretability: More complex encodings can sometimes improve model performance but at the cost of making your model harder to interpret. Consider your project's goals and choose accordingly.

Summary

14 February 2024 18:53

1. Ordinal Encoder

- How it works: Converts each category into a unique integer based on the order of appearance or alphabetical order.
- Advantages:
  - Simple to implement and understand.
  - Preserves order where it might be meaningful.
- Disadvantages:
  - Imposes an ordinal relationship that may not exist, potentially misleading the model.
  - Not suitable for non-ordinal data or models sensitive to numerical relationships.
- Use Cases:
  - Tree-based models where ordinal relationships can be useful.
  - Situations where the natural order of categories carries meaningful information.

2. One-Hot Encoder

- How it works: Creates a separate binary column for each category level, with a 1 indicating the presence of the category.
- Advantages:
  - Prevents the introduction of artificial ordinal relationships.
  - Easy to interpret and implement.
- Disadvantages:
  - Can lead to a large increase in dataset dimensionality, especially with high-cardinality features.
  - Not efficient for models that can inherently handle categorical data.
- Use Cases:
  - Linear models or neural networks where categorical variables need to be explicitly converted into numerical format.
  - Datasets with a relatively small number of unique categories.

3. Binary Encoder

- How it works: First converts categories to ordinal numbers, then encodes those numbers as binary, and finally splits the digits of each binary number into separate columns.

- Advantages:
  - More space-efficient than one-hot encoding for high-cardinality features.
  - Avoids the curse of dimensionality to some extent.
- Disadvantages:
  - Still introduces a form of ordinality that might not be inherent to the data.
  - Binary representation can be less intuitive for interpretation.
- Use Cases:
  - Situations with high-cardinality categorical features where one-hot encoding is impractical.
  - Models that benefit from reduced dimensionality but can't exploit categorical nature directly.

4. BaseN Encoder

- How it works: A generalization of binary encoding, allowing encoding to a base-N representation, thus providing a flexible compromise between one-hot and binary encoding.
- Advantages:
  - Customizable to balance between dimensionality and information retention.
  - Useful for controlling the expansion of feature space.
- Disadvantages:
  - Selection of base requires tuning and experimentation.
  - May introduce complexity in interpretation depending on the chosen base.
- Use Cases:
  - Datasets where neither binary nor one-hot encoding offers a satisfactory compromise between information retention and feature space expansion.
  - Scenarios requiring a tunable approach to categorical encoding.

5. Target Encoder

- How it works: Replaces a categorical value with the mean of the target variable for that category.
- Advantages:
  - Captures information about the target, potentially improving model performance.
  - Reduces dimensionality without losing important information.
- Disadvantages:

    - Risk of overfitting and data leakage if not properly regularized or if used without cross-validation.
    - Target leakage can occur if mean calculation includes the validation/test set.
- Use Cases:
    - Models where capturing the relationship between features and the target is crucial.
    - Situations with categorical variables that have a direct relationship with the target variable.

6. James-Stein Encoder

- How it works: Employs the James-Stein estimator to shrink estimates of means towards the overall mean, a form of regularization that can be particularly effective when dealing with small sample sizes.
- Advantages:
    - Can improve estimates for categories with few observations.
    - Helps to prevent overfitting by shrinking extreme values towards the global mean.
- Disadvantages:
    - The approach is somewhat complex and less intuitive than simpler methods.
    - Performance gains are context-dependent and may not always justify the additional complexity.
- Use Cases:
    - Regression tasks where the target variable is continuous, and the dataset contains many small categories.

7. M-estimate Encoder

- How it works: A simplified version of target encoding which adds a smoothing parameter to balance the category mean and the overall mean, reducing the impact of categories with few samples.
- Advantages:
    - Provides a balance between the category mean and the global mean, reducing the risk of overfitting.
    - Simple to implement with a single additional smoothing parameter.
- Disadvantages:
    - Requires careful tuning of the smoothing parameter to avoid underfitting or overfitting.
    - May still be susceptible to data leakage if not used with proper cross-validation techniques.
- Use Cases:

    - Situations where target encoding is desirable but with additional control over the influence of small sample sizes.
    - Regression and binary classification problems with categorical variables.

8. Weight of Evidence Encoder

- How it works: Transforms categories based on the log of odds of the target being 1 within each category, traditionally used in credit scoring.
- Advantages:
    - Directly interpretable in terms of the odds ratio, offering insights into the predictive power of categories.
    - Naturally handles binary target variables, making it ideal for binary classification.
- Disadvantages:
    - Primarily suitable for binary classification tasks.
    - Handling of categories with zero occurrences of the target variable requires careful treatment to avoid infinite values.
- Use Cases:
    - Credit risk assessment and other binary classification tasks where understanding the odds is important.
    - Financial and medical domains where interpretability of model inputs is crucial.

9. Leave One Out Encoder

- How it works: Similar to target encoding but leaves out the current row's target when calculating the mean target for a category to reduce overfitting.
- Advantages:
    - Reduces the likelihood of overfitting compared to standard target encoding.
    - Maintains the relationship between categorical variables and the target.
- Disadvantages:
    - Computationally more intensive than standard target encoding.
    - Still may require additional regularization techniques in practice.
- Use Cases:
    - Any supervised learning task where target leakage and overfitting are major concerns.
    - Projects where model interpretability is less critical than predictive accuracy.

10. CatBoost Encoder

- How it works: Similar to leave-one-out encoding but uses more advanced

smoothing techniques, inspired by the CatBoost algorithm, to reduce overfitting further.
- Advantages:
  - Offers strong performance with categorical variables, particularly with tree-based models.
  - Incorporates sophisticated smoothing to combat overfitting effectively.
- Disadvantages:
  - Complexity can make it harder to interpret than simpler encoders.
  - May not always offer performance improvements over simpler methods depending on the dataset.
- Use Cases:
  - Classification and regression tasks where categorical variables play a significant role.
  - Projects benefiting from tree-based models, especially when using CatBoost.

11. Generalized Linear Mixed Model Encoder (GLMM)'

- How it works: Uses a generalized linear mixed model to estimate the effect of each category on the target, blending categorical encoding with statistical modeling techniques.
- Advantages:
  - Can capture complex relationships between categorical features and the target.
  - Provides a statistically rigorous way to encode categorical variables.
- Disadvantages:
  - Requires more statistical knowledge to implement and interpret.
  - Computationally intensive compared to simpler encoding methods.
- Use Cases:
  - Advanced statistical analysis and predictive modeling in fields like biostatistics or social sciences.
  - Situations where the relationship between categorical variables and the target is complex and potentially non-linear.

12. Sum Encoder (Effect Encoder)

- How it works: Similar to one-hot encoding but uses -1 for the reference category instead of dropping it, allowing for the representation of effects relative to a baseline.
- Advantages:
  - Captures relative effects of categories against a baseline, useful for linear models.
  - Can be more informative than standard one-hot encoding in certain

analyses.

- Disadvantages:
  - Can introduce multicollinearity in linear models without regularization.
  - Interpretation can be less straightforward than with one-hot encoding.
- Use Cases:
  - Linear regression models where understanding the relative effect of categories is important.
  - Situations requiring a baseline or reference category for comparison.

13. Polynomial Encoder

- How it works: Encodes categorical variables as orthogonal polynomials, capturing non-linear relationships between the categorical variable and the target.
- Advantages:
  - Can model complex, non-linear relationships within the categorical data.
  - Useful for trend analysis in ordered categories.
- Disadvantages:
  - More difficult to interpret than linear encodings.
  - Best suited for ordered categorical variables, limiting its applicability

  Use Cases:
  - Analysis where the order of categories matters and the relationship between categories and the target is suspected to be non-linear.
  - Regression models that benefit from capturing polynomial relationships.

14. Backward Difference Encoder

- How it works: Encodes categories by calculating the difference between each category and the preceding category, emphasizing the change between adjacent categories.
- Advantages:
  - Highlights the difference in the target variable between adjacent categories, useful for ordered categories.
  - Reduces multicollinearity compared to one-hot encoding.
- Disadvantages:
  - The encoding assumes an ordinal relationship, which may not always be appropriate.
  - Interpretation can be challenging, especially with non-ordinal data.
- Use Cases:
  - Situations where the focus is on the change or difference in the target variable between categories.
  - Ordered categorical data where the sequence of categories has

significance.

15. Helmert Encoder

- How it works: Compares each level of a categorical variable to the mean of the subsequent levels, offering a contrast coding system that is useful for hypothesis testing.
- Advantages:
  - Can be useful in statistical analysis, particularly in design of experiments and ANOVA.
  - Provides a systematic way to compare each category against the mean of subsequent categories.
- Disadvantages:
  - Interpretation can be less intuitive than other encoding methods.
  - Assumes an order in the categories, which may not exist.
- Use Cases:
  - Detailed statistical analyses where contrasts between categories are of interest.
  - Regression modeling in experimental designs.

16. Hashing Encoder

- How it works: Uses the hash function to encode categories into a fixed size of dimensions, reducing dimensionality and handling new categories dynamically.
- Advantages:
  - Efficient with high-cardinality features and large datasets.
  - Can handle unseen categories during training.
- Disadvantages:
  - Loss of information and possible collisions due to hashing to a smaller dimension.
  - Encoded features are not interpretable.
- Use Cases:
  - Text classification and natural language processing tasks.
  - Datasets with a large number of categories or dynamic features where new categories can appear.

17. Quantile Encoder

- How it works: Similar to target encoding but encodes categories based on the quantile of the target distribution within each category, rather than the mean.
- Advantages:

    - Captures the distribution of the target within categories, which can be more informative than the mean alone.
    - Useful when the target distribution is skewed or non-normal.
- Disadvantages:
    - Requires careful handling to avoid overfitting, similar to target encoding.
    - Interpretation of encoded values can be more complex than mean-based encodings.
- Use Cases:
    - Regression tasks where understanding the distribution of the target is crucial.
    - Projects where the target variable's mean does not fully capture the relationship with categorical features.