R2VF: A Two-Step Regularization Algorithm to Cluster Categories in GLMs

Yuval Ben Dror
Data Science Researcher, Earnix
June 2025

Intro – Evolution of GLM

We begin with a basic encoding of our features, and fit a **standard GLM**.

City
A
B
C
B
A
A

→

City_is_A	City_is_B	City_is_C
1	0	0
0	1	0
0	0	1
0	1	0
1	0	0
1	0	0

Age
18
29
22
47
68
81

→

<19	[19,24)	…	80<=
1	0	…	0
0	0	…	0
0	1	…	0
0	0	…	0
0	0	…	0
0	0	…	1

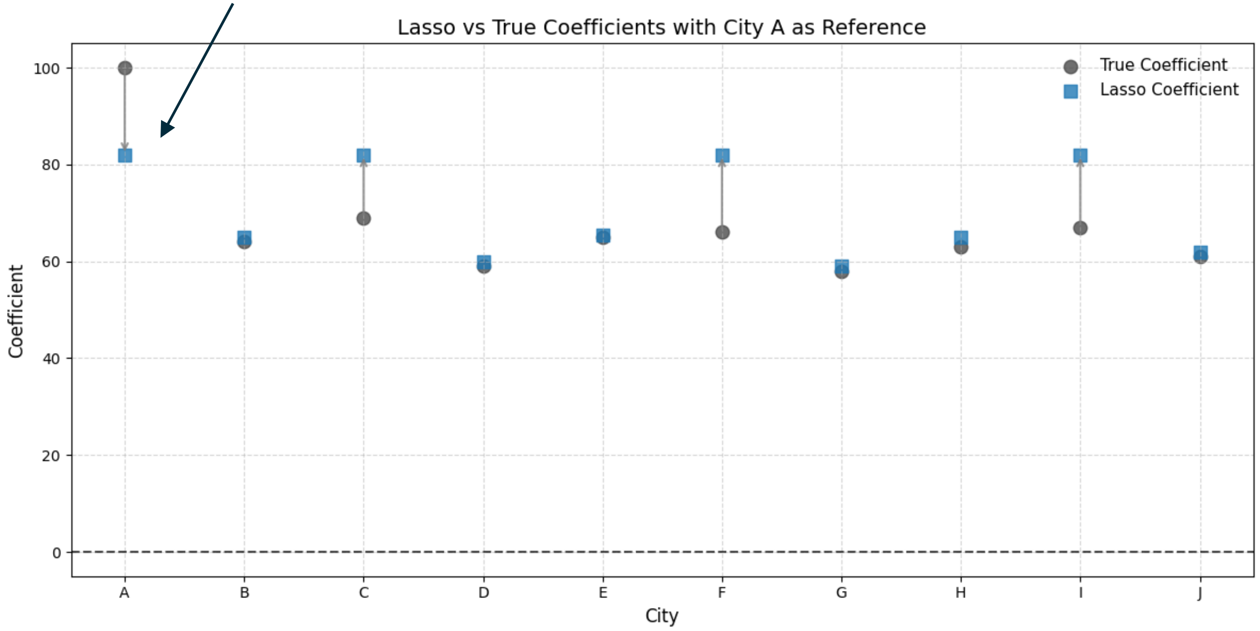
Initial fit: **overfit**.

earnix

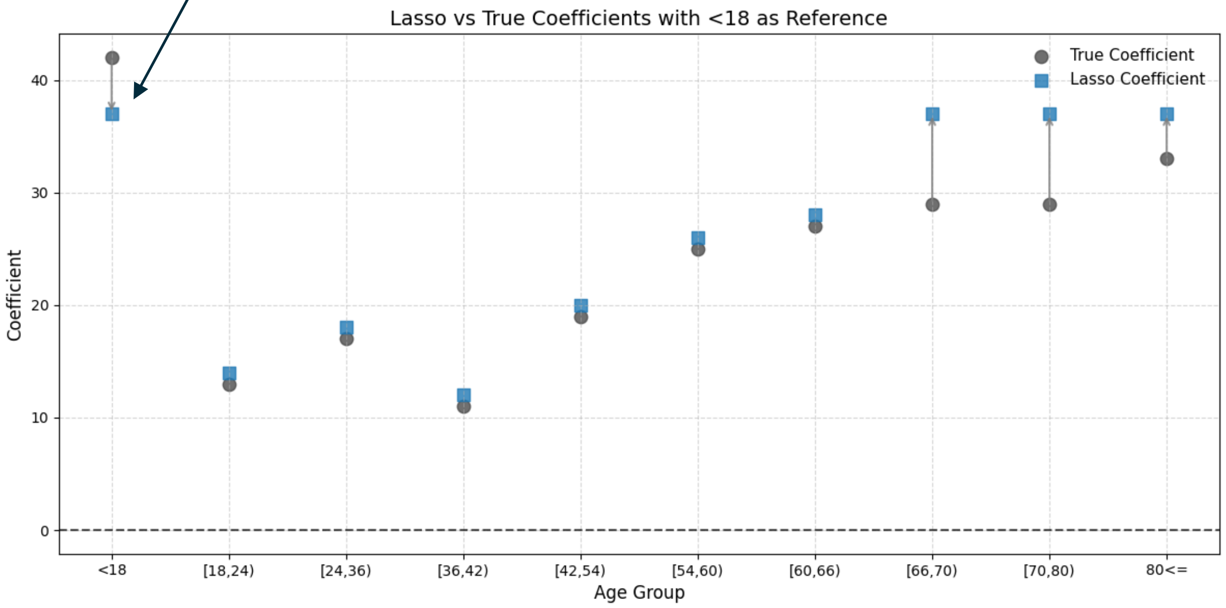
Intro – Evolution of GLM

Let’s use **Lasso**.

Reference Level



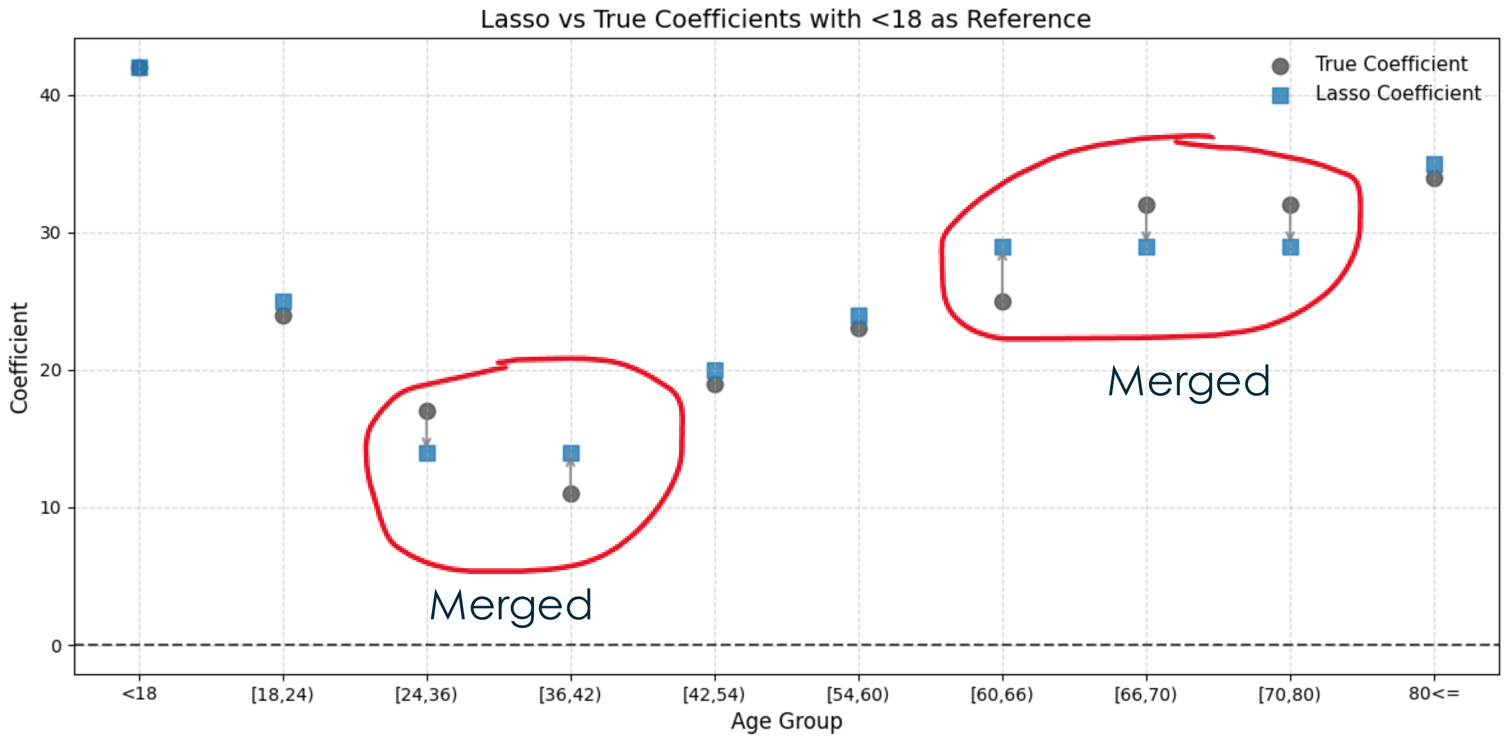
Reference Level



Standard Lasso fit: Shrinks coefficients only towards the **reference level**.

Intro – Evolution of GLM

For numeric and ordinal features – we use **fused lasso** instead.



Fused Lasso fit: works well for ordinal bins, but doesn't solve the issue with nominals.

Intro – Evolution of GLM

Bondel and Reich proposition for **nominal lasso**:

$$J_\lambda(\beta_{\text{nom}}) = \lambda \sum_{i=1}^{|NOM|} \sum_{j<k\in|NO_i|} w_{ij}\,|\beta_{ij} - \beta_{ik}|$$

However, for a **practical implementation**, this would require augmenting the design matrix **quadratically**.

1	0	1	0	1	0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	1	1	0	1	0	0
1	0	0	0	0	1	0	1	0	0	0	1	1
0	1	0	0	0	1	1	1	0	0	1	0	0
1	0	0	0	1	0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0	1	0	1	0	0
0	1	1	0	1	0	0	1	0	0	1	0	1
0	0	0	1	0	0	1	1	1	0	0	1	0
0	1	0	1	0	0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	1	1	0	0	1	0
1	1	0	0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	1	1	0	1	0	0	1	0

Naïve Approach: Target Encoding

Simple solution: use target encoding to rank the categories, and penalize only adjacent bins.

City	Target
A	1
B	7
C	5
B	6
A	3
A	2

→

City_numeric
2
6.5
5
6.5
2
2

Penalize: $|\beta_C - \beta_A|, |\beta_B - \beta_C|$

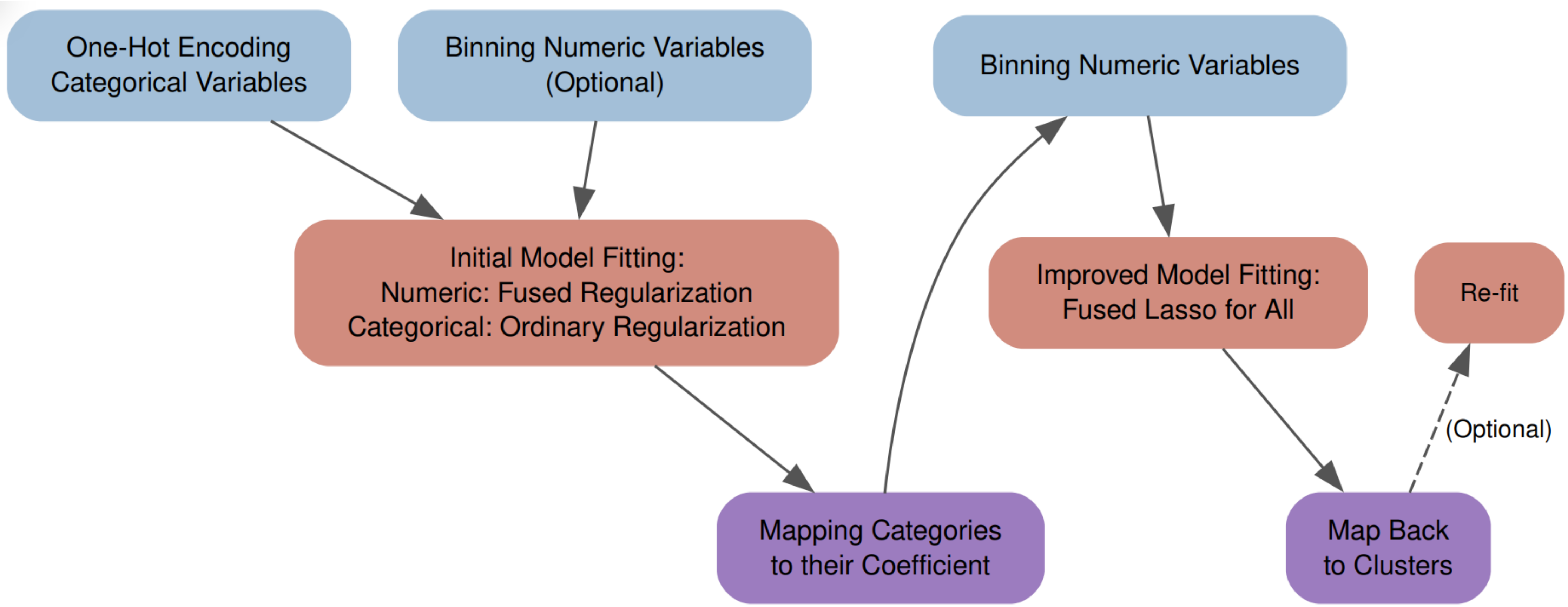
Two main issues:

1. **Target Leakage** (overfitting)
2. Relying on a **marginal effect**

earnix

Our Solution: R2VF (Ranking to Variable Fusion)

We propose the following mechanism:



R2VF: Benefits

Handling Overfitting: uses a regularized ranking for the categories.
Multivariate compatibility: uses the coefficients of the categories fitted with other predictors.
Avoiding leakage: uses a similarly structured model rather than the target itself.

earnix

Computational Approach

We use "**Split coding**" for ordinal features (which is ultimately all the features, after applying the initial steps).

Car Brand	Car Brand Ranked	>=1	>2	>3	>4
Suzuki	0	0	0	0	0
Mazda	1	1	0	0	0
Renault	2	1	1	0	0
Volkswagen	3	1	1	1	0
BMW	4	1	1	1	1

Initial bins: (Suzuki, Mazda, Renault, Volkswagen, BMW)

earnix

Computational Approach

Apply **standard lasso**, and merge accordingly.

Car Brand	Car Brand Ranked	>=1	>2	>3 Beta=0	>4
Suzuki	0	0	0	0	0
Mazda	1	1	0	0	0
Renault	2	1	1	0	0
Volkswagen	3	1	1	1	0
BMW	4	1	1	1	1

Final bins: (Suzuki, Mazda, **Renault && Volkswagen**, BMW)

Simulation

City: 26 cities labeled A to Z, randomly generated such that the number of observations per city roughly forms a linear scale (meaning, the frequency of each city varies),

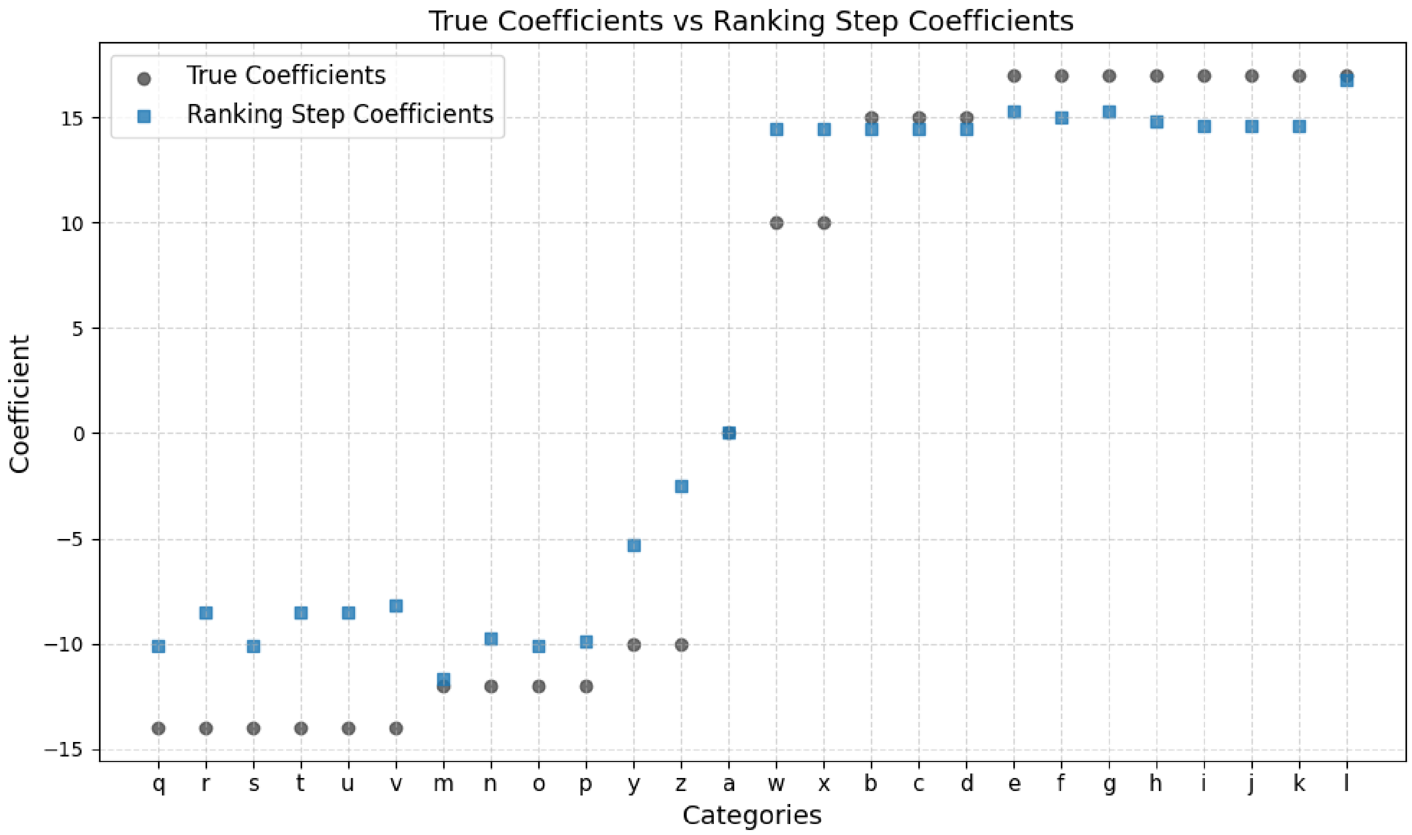
Age: An average age is randomly selected per city (varies from 34 to 46), and generated with a variability of 13.

Profession: Marked P_i (where i is a number from 0 to 99), and distributed such that it has a minor correlation with both city and age. The distribution makes some professions relatively prevalent, some very rare, and others completely absent.

```
target = 0 \
    + 15 * (row['city'] in ['b', 'c', 'd']) \
    + 17 * (row['city'] in ['e', 'f', 'g', 'h', 'i', 'j', 'k', 'l']) \
    - 12 * (row['city'] in ['m', 'n', 'o', 'p']) \
    - 14 * (row['city'] in ['q', 'r', 's', 't', 'u', 'v']) \
    + 10 * (row['city'] in ['w', 'x']) \
    - 10 * (row['city'] in ['y', 'z']) \
    - 2 * np.sqrt((row['age'] - 45))**2 \
    - 19 * (row['profession'][-1] == '1') \
    - 17 * (row['profession'][-1] == '2') \
    - 9 * (row['profession'][-1] == '3') \
    - 8 * (row['profession'][-1] == '4') \
    + 1 * (row['profession'][-1] == '5') \
    + 2 * (row['profession'][-1] == '6') \
    + 8 * (row['profession'][-1] == '7') \
    + 9 * (row['profession'][-1] == '8') \
    + 19 * (row['profession'][-1] == '9')
```

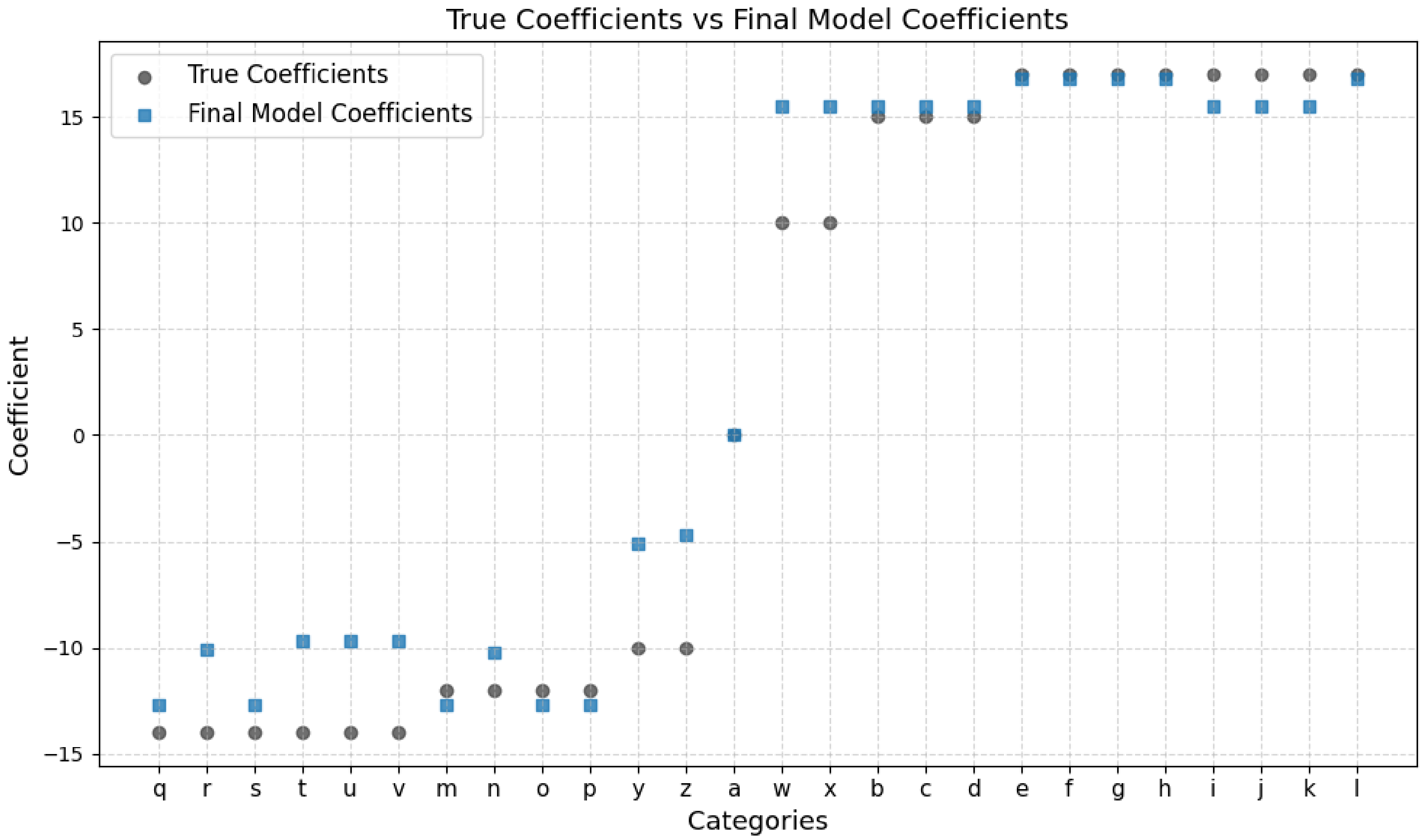
Simulation

Results after the **ranking step** (standard Lasso for nominals, fused lasso for ordinals):



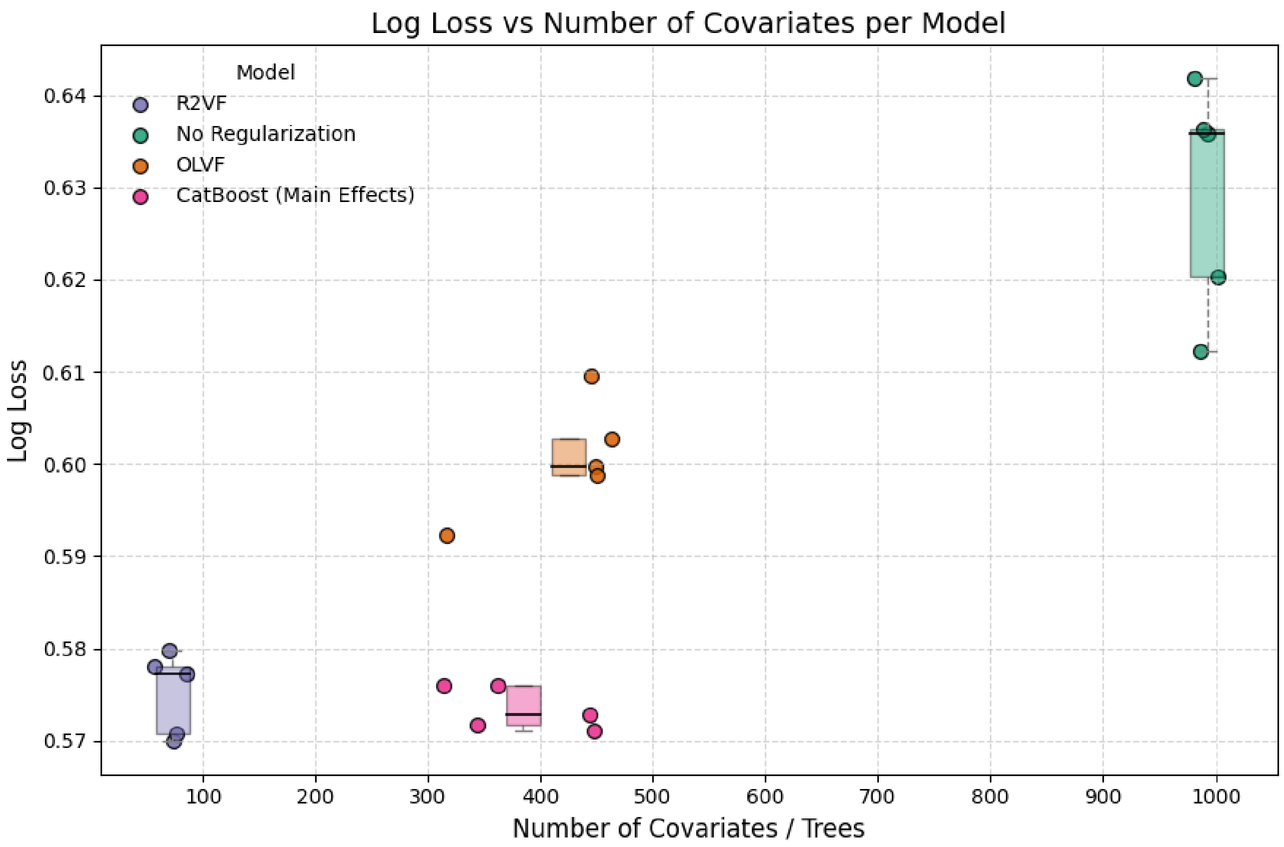
Simulation

Results after the **final step** (full R2VF):



Real Data Comparison

Dataset: FARS (Fatality Analysis Reporting System) 2022 – predicting **car accident** deaths by vehicle properties (**binary** prediction – death yes/no).



R2VF: Our two-step regularization model

OLVF: A Fused Lasso Model with standard handling of nominals

No Regularization: A non-regularized model of the initial bins

Catboost (Main Effects): A Catboost model with max depth of 1 (for a fair comparison of pure categorical treatment)

Paper + AGLM Feature

For more technical details, see our paper.

R2VF: A Two-Step Regularization Algorithm to Cluster Categories in GLMs

Yuval Ben Dror*

May 16, 2025

Abstract

Over recent decades, extensive research has aimed to overcome the restrictive underlying assumptions required for a Generalized Linear Model to generate accurate and meaningful predictions. These efforts include regularizing coefficients, selecting features, and clustering ordinal categories, among other approaches. Despite these advances, efficiently clustering nominal categories in GLMs without incurring high computational costs remains a challenge. This paper introduces Ranking to Variable Fusion (R2VF), a two-step method designed to efficiently fuse nominal and ordinal categories in GLMs. By first transforming nominal features into an ordinal framework via regularized regression and then applying variable fusion, R2VF strikes a balance between model complexity and interpretability. We demonstrate the effectiveness of R2VF through comparisons with other methods, highlighting its performance in addressing overfitting and identifying an appropriate set of covariates.

For information about Model Accelerator, an Earnix extension that includes Auto-GLM, see blog.

Thank you!

Yuval Ben Dror
Data Science Researcher, Earnix
June 2025

earnix