Feature Importance for Predictive Accuracy: An Euler Decomposition

Ludger Hentschel

April 30, 2026

Abstract

We develop the Euler Decomposition of Explained Fit (EDEF), an exact, additive, and model-conditional attribution of realized predictive accuracy to the features of a given prediction model. Existing feature-importance measures primarily address two distinct questions: which features are informative for model development, and why a model produces a particular prediction. EDEF addresses a third: how much does each feature contribute to the model's realized predictive accuracy?

Existing methods target different objects, like performance under refitting, prediction-level attribution, or performance under counterfactual inputs. They do not decompose realized predictive accuracy of a fixed fitted model. Once model fit is defined as the reduction in expected loss relative to a baseline, the attribution problem admits an exact solution.

We derive closed-form contributions using Euler's theorem for linear regressions and a path-integral extension for binary classification models. The attribution conditions on the fitted model and realized inputs, requires no refitting or counterfactual evaluation, and has negligible computational cost once predictions are available.

Because EDEF expresses feature importance as a sample average, it also admits standard errors that quantify sampling variability in the evaluation data. This enables formal statistical inference and monitoring of feature contributions across samples.

www.ludgerhentschel.com ludger@ludgerhentschel.com

Contents

Acknowledgements
For helpful comments, I am grateful to Nishant Gurnani, Ingemar Hentschel, and Shubham Jaiswal.

1 Introduction

Machine learning models are increasingly deployed in settings where predictive performance must be monitored, explained, and defended over time. For an established model, a central question is: how much does each feature contribute to the model's realized predictive accuracy? The large literature on feature importance targets different objects: model performance under refitting, prediction-level attribution, or performance under counterfactual inputs. It does not decompose the realized predictive accuracy of a fixed fitted model. We provide an exact attribution via the Euler Decomposition of Explained Fit (EDEF).

Feature importance measures are designed to answer different questions. We distinguish three types. The first concerns *model exploration*: which features are informative, and how does model performance change when features are added or removed? Methods such as partial R^2, dominance analysis (Budescu, 1993), and Shapley decompositions of explained variance (Lindeman et al., 1980; Kruskal, 1987) address this question by refitting models or evaluating counterfactual specifications.

The second concerns *prediction explanation*: why does a fixed, fitted produce a particular prediction? Methods such as SHAP (Lundberg and Lee, 2017) and integrated gradients (IG) (Sundararajan et al., 2017) attribute predictions to inputs relative to a baseline. These methods explain predictions, not predictive accuracy, and can assign large importance to features that do not improve and may degrade out-of-sample fit. This distinction is consequential: a feature may be highly informative or influential in prediction, yet contribute little to realized predictive accuracy once redundancy and interactions within the fitted model are taken into account.

The third question, which we address, concerns *model-fit attribution*: how much does each feature contribute to the realized predictive accuracy of a fixed, fitted model? This question is model-conditional and evaluates the model only at the realized inputs. It arises naturally whenever model performance is assessed: at the end of model development, during validation and comparison, in production monitoring, and when diagnosing performance changes or proposed model modifications.

No existing method provides an exact, additive decomposition of realized predictive accuracy for a fixed fitted model. Among existing approaches, SAGE (Covert et al., 2020) is closest in spirit: it is global, accuracy-oriented, and additive. We show that SAGE and EDEF coincide under quadratic loss with an additive signal, but differ in general. SAGE evaluates model performance under counterfactual inputs generated by feature removal and measures how

performance would change if features were unavailable. EDEF instead conditions on the realized inputs and attributes how observed feature variation contributed to realized predictive accuracy.

Our approach is direct. We define model fit as the reduction in expected loss relative to a baseline predictor. This definition applies in and out of sample and aligns with how predictive performance is evaluated in practice. When the fitted signal admits an additive decomposition, attribution follows from Euler's theorem for homogeneous functions, or from its path-integral extension when homogeneity fails. The resulting decomposition is exact, additive, global, and model-conditional, and is computationally trivial once fitted predictions are available.

Regression

For regression models, we measure fit as the reduction in mean squared error relative to an intercept-only baseline. When the fitted signal admits an additive representation, Euler's theorem yields an exact additive decomposition across prediction components. The result applies broadly, including to ordinary least squares and regularized linear models, and remains valid out of sample.

Classification

For binary classification, we measure fit as the improvement in expected log loss relative to a constant-probability baseline. Because log loss is not homogeneous in the fitted score, we use a path-integral representation to obtain an exact additive decomposition. The resulting contributions provide a model-conditional attribution of predictive accuracy under a strictly proper scoring rule.

Standard errors

EDEF contributions are sample averages of observation-level quantities and therefore admit standard errors that quantify sampling variability in the evaluation data. This enables formal inference and monitoring of feature contributions over time. In contrast, variability reported by other methods typically reflects approximation error rather than sampling uncertainty.

Empirical illustrations

We illustrate EDEF using the Ames Housing regression dataset (De Cock, 2011) and the UCI Adult income classification dataset (Dua and Graff, 2019; Kohavi, 1996). In both cases, we evaluate feature contributions on held-out samples. The examples show that EDEF and SAGE produce similar results under quadratic loss, while other methods differ because they target different objects. They also show that features commonly identified as important may

contribute little to realized predictive accuracy once the full model is taken into account.

Speed

EDEF is computationally trivial once fitted predictions are available, requiring a single pass over the evaluation data. In contrast, methods such as SAGE rely on Monte Carlo approximation and repeated model evaluation, making them substantially more expensive.

Organization

The remainder of the paper proceeds as follows. Section 2 introduces the general framework and derives the decomposition and standard errors. Section 3 specializes to regression and establishes the connection to R^2 and the Pratt decomposition. Section 4 develops the classification case under log loss. Section 5 compares EDEF to existing methods. Section 6 presents the empirical examples, and Section 7 concludes.

2 Euler Decomposition of Explained Fit

This section develops the general EDEF framework. We define a loss-based measure of predictive accuracy relative to a baseline predictor and derive an exact, additive decomposition together with its standard errors. Sections 3 and 4 specialize this framework to regression and classification, respectively.

2.1 Setup

Let $\widetilde{y} \in \mathbb{R}^N$ denote a vector of observed outcomes with finite, nonzero variance and define the centered outcome

$$y = \widetilde{y} - \mathbb{E}[\widetilde{y}]. \tag{1}$$

We interpret the intercept-only predictor $\widehat{y}_0 = \mathbb{E}[\widetilde{y}]$ as the baseline model and evaluate predictive performance relative to this baseline.

Throughout, expectations, variances, and covariances are sample averages. Since y is centered, $\mathbb{E}[y] = 0$ and $\mathrm{Var}(y) = \mathbb{E}[y^2]$.

Let $\widehat{y} \in \mathbb{R}^N$ denote the predictions of a regression or forecasting model, and assume that $\widehat{y}$ is centered so that $\mathbb{E}[\widehat{y}] = 0$.[1]

For all regression models, we measure predictive accuracy using mean squared error and define the improvement in fit relative to the baseline

[1] If the fitted model includes an intercept and is evaluated on centered regressors, this condition holds automatically. In regularized regressions, the intercept is typically excluded from regularization to preserve this property.

predictor as[2]

$$\Delta\mathcal{L} = \mathrm{Var}(y) - \mathrm{Var}(y - \widehat{y}). \tag{2}$$

With perfect predictions $\widehat{y} = y$, predictive accuracy attains a maximum value of $\mathrm{Var}(y)$. With poor predictions, predictive accuracy can be close to 0 or even negative, especially out of sample for noisy predictions.

Scaling by $\mathrm{Var}(y)$ yields the standard coefficient of determination

$$R^2 = \frac{\Delta\mathcal{L}}{\mathrm{Var}(y)} = 1 - \frac{\mathrm{Var}(y - \widehat{y})}{\mathrm{Var}(y)}. \tag{3}$$

The alternative expression $R^2 = \mathrm{Var}(\widehat{y})/\mathrm{Var}(y)$ holds only when predictions $\widehat{y}$ are orthogonal to prediction errors $y - \widehat{y}$, which is true in-sample for ordinary least squares but not in general. Without this orthogonality, $\Delta\mathcal{L}$ remains a measure of predictive accuracy, while $\mathrm{Var}(\widehat{y})/\mathrm{Var}(y)$ captures only the scale of predictions relative to outcomes.

Expanding the squared error yields

$$\Delta\mathcal{L}(\widehat{y}) = 2\ \mathrm{Cov}(y, \widehat{y}) - \mathrm{Var}(\widehat{y}). \tag{4}$$

This representation makes clear that predictive accuracy depends on both alignment with the outcome and the variance of the fitted signal. It is well defined both in and out of sample and does not rely on orthogonality or optimality conditions specific to any estimation procedure.

2.2 EDEF Decomposition

Equation (4) expresses predictive accuracy as the difference of two homogeneous functions,

$$\Delta\mathcal{L}(\widehat{y}) = g_1(\widehat{y}) - g_2(\widehat{y}), \tag{5}$$

where $g_1(\widehat{y}) = 2\,\mathrm{Cov}(y, \widehat{y})$ is homogeneous of degree one in $\widehat{y}$ and $g_2(\widehat{y}) = \mathrm{Var}(\widehat{y})$ is homogeneous of degree two.

This structure is not incidental. Because predictive accuracy is expressed as a function of the fitted signal that is homogeneous term-by-term, it admits

[2] Defining explained fit relative to an intercept-only baseline model is partly conceptual and partly a matter of convenience. The intercept is not a feature, but a normalization that centers predictions and defines the baseline level of predictive accuracy. EDEF allocates only the incremental explained fit arising from additive prediction components beyond the mean. When the contribution of the unconditional mean is of interest, it can be tracked separately as a baseline component of total fit.

an exact additive decomposition via Euler's theorem. In this sense, the attribution problem is solved directly by the functional form of the evaluation metric: once predictive accuracy is written in this form, the decomposition follows without approximation or additional assumptions.

For a function homogeneous of degree k, Euler's theorem states $g(x) = \frac{1}{k} x^\top \nabla g(x)$. Applying Euler's theorem to each term yields

$$g_1(\widehat{y}) = \frac{2}{N}\, \widehat{y}^\top y, \tag{6}$$

and

$$g_2(\widehat{y}) = \frac{1}{N}\, \widehat{y}^\top \widehat{y}. \tag{7}$$

Euler's theorem is an exact identity for homogeneous functions, not a local approximation.

In many prediction models, the fitted signal admits an additive decomposition

$$\widehat{y} = \sum\nolimits_j \widehat{y}_j, \tag{8}$$

where $\widehat{y}_j$ denotes a component associated with a feature, regressor, or model element.

To obtain a well-defined attribution, we impose the normalization

$$\mathbb{E}[\widehat{y}_j] = 0 \qquad \text{for all } j, \tag{9}$$

which assigns all level effects to the intercept-only baseline. This normalization does not affect fitted values or predictive accuracy, but ensures that contributions are uniquely defined.

Substituting the additive decomposition into the Euler representation yields the EDEF result: an exact additive decomposition of predictive accuracy,

$$\Delta\mathcal{L}(\widehat{y}) = \sum\nolimits_j C_j, \tag{10}$$

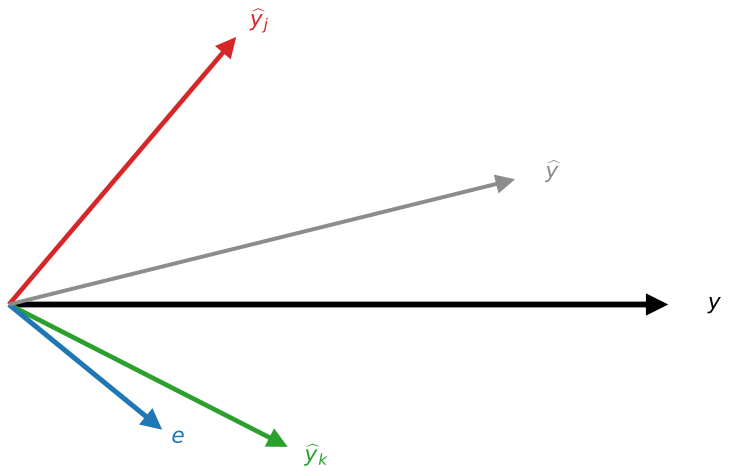
$$C_j = 2\,\mathrm{Cov}(y, \widehat{y}_j) - \mathrm{Cov}(\widehat{y}, \widehat{y}_j) \tag{11}$$

$$= \mathrm{Cov}(y, \widehat{y}_j) + \mathrm{Cov}(e, \widehat{y}_j), \tag{12}$$

where $e = y - \widehat{y}$ is the realized prediction error.[3]

[3] This is correct even if $\Delta\mathcal{L}(\widehat{y}) < 0$. Less negative contributions C_j indicate stronger contributions to fit.

Figure 1: Geometry of Euler Contributions to Model Fit



The figure illustrates the Euler decomposition of realized regression fit for an outcome vector y (black) and a fitted value $\widehat{y} = \widehat{y}_j + \widehat{y}_k$ (gray). Two additive prediction components, $\widehat{y}_j$ (red) and $\widehat{y}_k$ (green), appear with different alignment relative to the prediction error e (blue).

In the Euler decomposition, the contribution of component $\widehat{y}_\ell$ depends on both its covariance with y and its covariance with $e = y - \widehat{y}$. These covariances determine whether the component contributes new explanatory direction or primarily overlaps with the existing fitted value.

In the diagram, both $\widehat{y}_j$ and $\widehat{y}_k$ align positively with y, but they exhibit different alignment with the realized residual e. Here, $\widehat{y}_j$ makes a smaller Euler contribution to model fit because its negative alignment with e contributes to a larger distance between $\widehat{y}$ and y. Conversely, $\widehat{y}_k$ makes a larger Euler contribution because its positive alignment with e contributes to a smaller distance between $\widehat{y}$ and y.

The Euler decomposition and the figure do not rely on orthogonality assumptions; angles represent empirical covariances.

Figure 1 shows that the contribution C_j admits a geometric interpretation. View y, $\widehat{y}$, and the components $\widehat{y}_j$ as vectors in $\mathbb{R}^N$ with inner product $\langle a, b \rangle = \text{Cov}(a, b)$. All quantities are fixed once the model is fitted.

The first term in (12) measures alignment with the outcome, while the second captures alignment with the realized error. A component improves predictive accuracy to the extent that it aligns with both. Components that primarily reinforce existing fitted structure without reducing residual error receive smaller contributions and may be negative.

Negative contributions arise naturally from interaction among components rather than from any intrinsic deficiency of a feature. Predictive accuracy depends on how components combine, not on their magnitudes in isolation. When monitoring model performance across samples, persistent negative contributions may reflect stable redundancy, while changes in sign indicate shifts in how components interact within the fitted signal.

When the change in loss $\Delta\mathcal{L}(\widehat{y})$ is negative, some of the contributions must be negative. As our simulations in **??** confirm, this is most likely to occur

outside the training sample for models with low explanatory power. In these settings, any additive attribution must assign negative contributions to some components.

More generally, even when $\Delta\mathcal{L}(\widehat{y})$ is positive, individual contributions may be negative. This occurs when a component is misaligned with the observed outcome, $\mathrm{Cov}(y, \widehat{y}_j) < 0$, or when it reinforces the realized prediction error, $\mathrm{Cov}(e, \widehat{y}_j) < 0$. In the latter case, the component tends to move predictions in the opposite direction of the residual, pushing the fitted value further away from the outcome. Because the residual e is defined using the full fitted model, this term reflects alignment with realized prediction errors rather than the effect of removing the component.

The EDEF decomposition depends only on realized fitted values and their additive components. It conditions on the fitted model and the observed inputs, and requires no refitting, perturbation, or counterfactual evaluation.

2.3 Standard Errors

Each contribution C_j is a sample average of observation-level quantities c_{ij} and therefore inherits sampling variability. Feature importance is thus an estimable quantity rather than a purely descriptive label.

Appendix B shows that

$$SE(C_j) = \sqrt{\frac{1}{N}\,\mathbb{E}\big[(c_{ij} - C_j)^2\big]}. \tag{13}$$

These standard errors reflect sampling variability in the evaluation data, conditional on the fitted model. They apply both in-sample and out-of-sample and provide a basis for assessing whether contributions differ reliably from zero, materially exceed a threshold, or whether changes in contributions across samples reflect noise or genuine shifts in predictive relevance.

We often express contributions to fit in proportional terms. Appendix B also derives standard errors for the proportional contributions. However, statistical tests for contributions are often sharper than statistical tests for proportional contributions. The proportional contributions introduce additional estimation noise through the denominator. In particular, when testing whether a contribution is different from zero, testing C_j is logically equivalent to testing shares but always has smaller standard errors.

The standard errors measure sampling variability in the observations used to evaluate the fitted model, conditional on the model itself. This differs from the variability measures reported by feature importance methods that rely on Monte Carlo approximation, such as SHAP and SAGE. In those

approaches, reported variability reflects approximation error arising from sampling feature coalitions or background observations, rather than sampling uncertainty in the evaluation data. As a result, increasing the number of Monte Carlo samples reduces this variability without changing the underlying estimand. By contrast, EDEF requires no simulation and its standard errors decrease with the size of the evaluation sample, providing a direct basis for statistical inference about feature contributions. These two notions of variability are conceptually distinct and not directly comparable.

2.4 Grouped EDEF

Because EDEF contributions sum exactly to total predictive accuracy, they can be aggregated across groups of components. This is particularly useful when components are numerous or highly collinear, in which case individual contributions may be unstable. It is also helpful for categorical features with many categories. We commonly expand such features into several dummy variables but may be interested in the importance of a group of dummies corresponding to a single categorical feature.

For a partition of components into groups G, define

$$C_G = \sum_{j \in G} C_j. \tag{14}$$

Then

$$\Delta\mathcal{L} = \sum_G C_G, \tag{15}$$

so predictive accuracy is allocated exactly across groups.

Standard errors follow from the same logic:

$$SE(C_G) = \sqrt{\frac{1}{N}\,\mathbb{E}[(c_{iG} - C_G)^2]}, \qquad c_{iG} = \sum_{j \in G} c_{ij}. \tag{16}$$

Grouped EDEF attribution aggregates interchangeable components into more stable or more interpretable units while preserving exact additivity. This parallels Owen values (Owen, 1977), but avoids combinatorial averaging and requires no additional computation once the fitted model is available.

3 Linear Regression

Linear regression is the canonical setting for EDEF. The fitted signal decomposes naturally into regressor-specific components, the EDEF contributions take a transparent closed form, and there is an exact connection to R^2 and the

in-sample Pratt decomposition. This section develops these results and clarifies when and why variance-based measures of feature importance coincide with or diverge from EDEF.

3.1 Features as Prediction Components

For linear models of the form $\widehat{y} = X\widehat{\beta}$ with centered regressors X, the fitted signal decomposes naturally into regressor-specific components

$$\widehat{y}_j = X_j \widehat{\beta}_j. \tag{17}$$

Substituting into the EDEF contribution yields

$$C_j = 2\,\widehat{\beta}_j \operatorname{Cov}(y, X_j) - \widehat{\beta}_j \operatorname{Cov}(\widehat{y}, X_j) \tag{18}$$

$$= \widehat{\beta}_j \operatorname{Cov}(y, X_j) + \widehat{\beta}_j \operatorname{Cov}(e, X_j), \qquad e = y - \widehat{y}. \tag{19}$$

The first term reflects the regressor's marginal association with the outcome, while the second captures how the regressor-specific fitted component aligns with the realized prediction error after accounting for the full model.[4]

This decomposition makes clear that marginal association alone does not determine predictive contribution. A regressor may have a large marginal correlation with y yet contribute little to predictive accuracy if its fitted component primarily reinforces other components without reducing residual error. Conversely, a regressor with modest marginal explanatory power may materially improve predictive accuracy by correcting systematic prediction errors.

3.2 In-Sample OLS and the Connection to R^2

Ordinary least squares constitutes a special case in which predictive accuracy, explained variance, and correlation-based measures coincide in sample. Under OLS with centered variables, fitted values $\widehat{y}$ are orthogonal to residuals $e = y - \widehat{y}$, implying

$$\operatorname{Cov}(e, X_j) = 0 \qquad \text{for all } j, \tag{20}$$

and therefore

$$C_j = \widehat{\beta}_j \operatorname{Cov}(y, X_j). \tag{21}$$

[4] Because the fitted signal is linear in the coefficients, EDEF contributions can equivalently be computed by differentiating with respect to $\widehat{\beta}_j$ rather than $\widehat{y}_j$, treating X as fixed.

Summing across features gives

$$\Delta\mathcal{L} = \sum_j \widehat{\beta}_j \operatorname{Cov}(y, X_j) = \operatorname{Cov}(y, \widehat{y}) = \operatorname{Var}(\widehat{y}), \tag{22}$$

where the final equality follows from OLS orthogonality. Normalizing by $\operatorname{Var}(y)$ yields

$$R^2 = \frac{\Delta\mathcal{L}}{\operatorname{Var}(y)} = \frac{\operatorname{Var}(\widehat{y})}{\operatorname{Var}(y)}. \tag{23}$$

In this knife-edge setting, decomposing predictive accuracy, predicted variance, and R^2 are equivalent up to scale. This equivalence is a special property of in-sample OLS and does not extend beyond this case.

The algorithm in Appendix D summarizes the computation for a general predictive model. Compared to many competing approaches, these computations are inexpensive: they require only means and covariances of realized predictions and their components, computed once on the evaluation sample.

3.3 Pratt Decomposition

Pratt (1987) proposes a decomposition of explained variance for linear regression based on marginal correlations. For standardized regressors estimated by ordinary least squares,

$$R^2 = \sum_j \widehat{\beta}_j \operatorname{Corr}(y, X_j), \tag{24}$$

with components

$$P_j = \widehat{\beta}_j \operatorname{Corr}(y, X_j) \tag{25}$$

interpreted as measures of variable importance.

The Pratt decomposition allocates *explained variance* in the estimation sample and relies on the orthogonality of fitted values and residuals.[5] By contrast, EDEF attributes *realized predictive accuracy* of a fixed fitted model, measured as the reduction in mean squared error relative to a baseline.

For in-sample OLS with standardized regressors, proportional Pratt and EDEF attributions coincide exactly. This equivalence is driven entirely by OLS orthogonality. Outside this setting — including out-of-sample evaluation, weighted or generalized least squares, and regularized linear models such as Ridge, Lasso, and Elastic Net — residuals generally correlate with fitted

[5] Thomas, Hughes, and Zumbo (1998) provide a geometric interpretation of the Pratt decomposition.

components. In these cases, predicted variance no longer coincides with predictive accuracy, and the ratio $\mathrm{Var}(\widehat{y})/\mathrm{Var}(y)$ reflects only the scale of predictions rather than their alignment with the outcome.

EDEF continues to apply without modification in all of these settings, providing an exact and additive decomposition of predictive accuracy precisely where variance-based measures lose their interpretation.

3.4 Relation to Shapley Decomposition of Model Fit

Under squared-error loss with an additive signal, the Shapley (1953) decomposition of explained fit collapses algebraically to the EDEF contributions. This equivalence follows directly from the Shapley marginal contribution formula.

Let the value function

$$v(S) = \Delta\mathcal{L}\left(\sum_{k\in S}\widehat{y}_k\right) \tag{26}$$

denote the explained fit of the partial prediction using components in S. The marginal contribution of component j to coalition S is

$$v(S\cup j) - v(S) = 2\,\mathrm{Cov}(y,\widehat{y}_j) - 2\,\mathrm{Cov}(\widehat{y}_S,\widehat{y}_j) - \mathrm{Var}(\widehat{y}_j), \tag{27}$$

where $\widehat{y}_S = \sum_{k\in S}\widehat{y}_k$.

The Shapley value is the expectation of this marginal contribution over all permutations. For a uniformly random permutation, each component $k \neq j$ precedes j with probability one half. Writing

$$\widehat{y}_S = \sum_{k\neq j}\mathbf{1}k \prec j\widehat{y}_k, \tag{28}$$

linearity of covariance implies

$$\mathbb{E}[\mathrm{Cov}(\widehat{y}_S,\widehat{y}_j)] = \frac{1}{2}\,\mathrm{Cov}(\widehat{y}-\widehat{y}_j,\widehat{y}_j). \tag{29}$$

Taking expectations yields the Shapley value

$$\phi_j = \mathbb{E}!\left[v(S\cup j) - v(S)\right] \qquad = 2\,\mathrm{Cov}(y,\widehat{y}_j) - \mathrm{Cov}(\widehat{y},\widehat{y}_j), \tag{30}$$

which coincides exactly with the EDEF contribution.

Under quadratic loss, Shapley allocations average out overlap symmetrically, yielding the Euler allocation in closed form. This equivalence does not extend beyond quadratic loss or additive signal representations.

This equivalence has two important implications. First, in the quadratic-loss setting with additive predictions, the Shapley decomposition of model fit admits a closed-form solution and does not require combinatorial averaging. Second, methods that approximate Shapley values for model fit via Monte Carlo sampling, such as SAGE, are targeting the same object in this setting. EDEF therefore provides the exact Shapley allocation at negligible computational cost. Section 5 discusses the relationship between these approaches in more detail.

3.5 Scope of EDEF

The results above do not depend on least squares or on orthogonality conditions. The essential requirement is that the fitted signal admits an additive representation

$$\widehat{y} = \sum_j \widehat{y}_j. \tag{31}$$

EDEF assigns importance directly to these additive components.

The same structure extends immediately to weighted and generalized least squares. After applying the implied transformation to outcomes and regressors, the fitted signal remains additive and EDEF applies without modification.

Penalized linear models, including Ridge (Hoerl and Kennard, 1970), Lasso (Tibshirani, 1996; Zou, 2006), and Elastic Net (Zou and Hastie, 2005), also satisfy this structure. These models violate the OLS orthogonality conditions, so the equivalence with the Pratt decomposition breaks down, but EDEF continues to measure predictive accuracy without modification. The Ames Housing illustration in section 6.2 demonstrates this point empirically.

Generalized linear models (GLMs) admit EDEF when the fitted signal is taken to be the linear predictor $\widehat{\eta} = X\widehat{\beta}$, rather than the conditional mean $g^{-1}(\widehat{\eta})$. On this scale, the signal decomposes additively into components $\widehat{\eta}_j = X_j\widehat{\beta}_j$.[6] The choice of signal representation and the choice of loss function are distinct: the present section concerns additive structure, while Section 4 addresses the role of nonlinear loss.

More broadly, any model with an explicitly additive predictor supports EDEF on that scale. Generalized additive models provide such a decomposition by construction. Many machine learning methods produce predictions that are additive in meaningful internal components (Hastie et al., 2009), including

[6] Attribution on the mean scale generally requires nonlinear transformations and does not admit a simple additive decomposition.

ensemble methods such as boosting and random forests, and kernel methods expressed in terms of training examples.

We treat prediction components as primitive and do not require them to correspond to original input features. In nonlinear architectures, predictions are linear in collections of constructed features or internal activations, and EDEF assigns predictive accuracy directly to these components.

When the object of interest is attribution to the original input variables, a closed form additive decomposition is generally unavailable outside linear models. In such cases, attribution requires aggregating marginal effects along paths in input space. Hentschel (2026) develops this path-integral attribution for nonlinear models using numerical integration.

4 Binary Classification

The regression setting in section 3 admits a direct application of Euler's theorem because the improvement in fit is homogeneous in the fitted signal. In binary classification, this property fails. Log loss is not homogeneous in the fitted score, so endpoint derivatives alone do not yield an exact decomposition of global fit.

We address this by applying the fundamental theorem of calculus along the straight-line path from the baseline score to the fitted score. This path-integral construction, detailed in appendix A, is the natural generalization of Euler's theorem when homogeneity fails. For log loss, the resulting contributions admit a closed-form expression and require no numerical integration. EDEF therefore applies with the same exactness and additivity as in the regression case.

4.1 Log Loss as a Measure of Classification Fit

For classification, we measure model fit using improvement in expected log loss rather than threshold-based classification accuracy. Log loss plays the same role in classification as explained variance in regression: it evaluates how well a model explains outcomes relative to a baseline, rather than how often it produces correct classifications at an arbitrary threshold.

Log loss is a strictly proper scoring rule for binary outcomes: expected loss is uniquely minimized when predicted probabilities equal true conditional probabilities (Gneiting and Raftery, 2007). It therefore evaluates probabilistic forecasts directly, rewards calibration, and admits well-defined population expectations.[7]

[7] These properties distinguish log loss from threshold-dependent metrics such as accuracy or F1, and from rank-based measures such as AUC. Such metrics do not evaluate probabilistic

We evaluate model fit using

$$\ell(y, \widehat{p}) = -y \log \widehat{p} - (1 - y) \log(1 - \widehat{p}). \tag{32}$$

Although log loss coincides with the negative log likelihood for Bernoulli models, we use it purely as an evaluation metric. The decomposition applies to any classification model that produces probabilistic predictions, regardless of how those predictions were obtained.

4.2 Baseline Model and Explained Log Loss

We define explained fit relative to the constant baseline $\overline{p} = \mathbb{E}[y]$, corresponding to an intercept-only model. This is directly analogous to centering the outcome in regression.

Explained log loss is

$$\Delta\mathcal{L} = \mathbb{E}[\ell(y, \overline{p})] - \mathbb{E}[\ell(y, \widehat{p})]. \tag{33}$$

This quantity measures how much predictive information the fitted model captures beyond unconditional class frequencies.

4.3 EDEF Decomposition for Log Loss

Define the log score

$$f(y, \widehat{\eta}) = y\widehat{\eta} - \log(1 + e^{\widehat{\eta}}), \tag{34}$$

where $\widehat{\eta}$ is the fitted score and $\widehat{p} = \sigma(\widehat{\eta})$. Let $\overline{\eta} = \log(\overline{p}/(1 - \overline{p}))$ denote the baseline score. Then

$$\Delta\mathcal{L} = \mathbb{E}[f(y, \widehat{\eta}) - f(y, \overline{\eta})]. \tag{35}$$

The decomposition depends only on the scoring rule and the realized scores, not on the estimation procedure that produced them.

Because f is not homogeneous in $\widehat{\eta}$, Euler's theorem does not apply directly. This is not a limitation of the approach but a property of the loss function. When homogeneity fails, the exact analogue of Euler's theorem is obtained by averaging the directional derivative along the path from the baseline to the fitted score.

---

forecasts directly and do not support additive decompositions of model fit. By contrast, log loss aggregates additively across observations and is therefore well suited to the decomposition developed here.

Applying the fundamental theorem of calculus along this path, for each observation,

$$f(y, \widehat{\eta}) - f(y, \overline{\eta}) = \int_0^1 \frac{d}{dt} f\left(y,\, \overline{\eta} + t(\widehat{\eta} - \overline{\eta})\right) dt \tag{36}$$

$$= (\widehat{\eta} - \overline{\eta}) \int_0^1 \left(y - \sigma\left(\overline{\eta} + t(\widehat{\eta} - \overline{\eta})\right)\right) dt. \tag{37}$$

Assuming an additive representation

$$\widehat{\eta} = \overline{\eta} + \sum_{j=1}^{K} \widehat{\eta}_j, \tag{38}$$

we obtain an exact additive decomposition

$$\Delta\mathcal{L} = \sum_{j=1}^{K} C_j, \tag{39}$$

with

$$C_j = \mathbb{E}\left[\widehat{\eta}_j \int_0^1 \left(y - \sigma\left(\overline{\eta} + t(\widehat{\eta} - \overline{\eta})\right)\right) dt\right]. \tag{40}$$

For log loss, the integral admits a closed form,

$$C_j = \mathbb{E}\left[\widehat{\eta}_j \left(y - \frac{\log(1 + e^{\widehat{\eta}}) - \log(1 + e^{\overline{\eta}})}{\widehat{\eta} - \overline{\eta}}\right)\right], \tag{41}$$

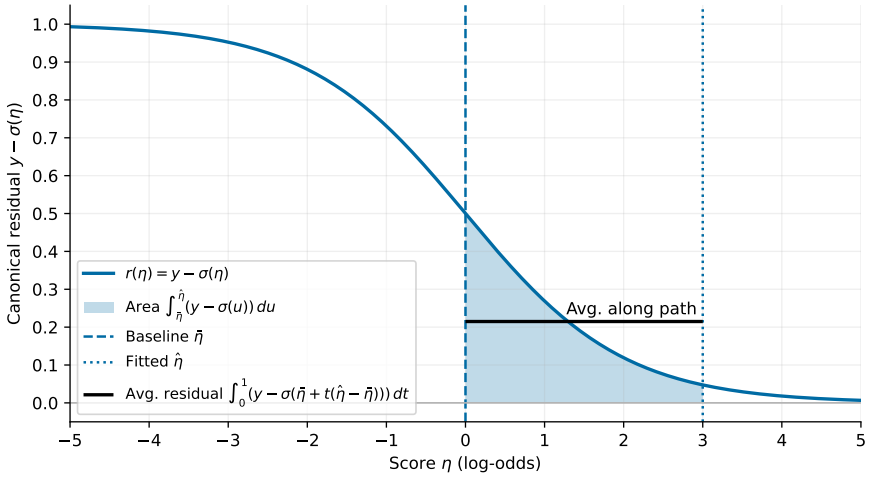
with the ratio defined by continuity when $\widehat{\eta} = \overline{\eta}$.

The contribution C_j measures how strongly the score component $\widehat{\eta}_j$ aligns with residual predictive uncertainty along the path from baseline to fitted score. Score movements near extreme probabilities contribute little, while movements near the decision boundary contribute more, reflecting diminishing marginal value of confidence.

Figure 2 illustrates this geometry. The canonical residual $y - \sigma(\eta)$ varies nonlinearly with the score, and contributions correspond to the area under this curve along the path from baseline to fitted score.

The decomposition is exact and attributes model fit at the level of score components. Additivity holds on the score scale: while probabilities are nonlinear in η, the score itself is linear and admits an additive representation. Many classification models produce additive score components. For nonlinear

Figure 2: Logistic Residual vs. Score and the Path Integral



The figure illustrates the path-integral calculation used to attribute model fit to a single score component. The plot shows the canonical residual $y - \sigma(\eta)$ as a function of the score η for a single observation with $y = 1$, a baseline score $\overline{\eta} = 0$, and a fitted score $\widehat{\eta} = 3$. Here, $\sigma(\cdot)$ is the logistic function. The shaded region corresponds to the path integral

$$\int_0^1 \left(y - \sigma\left(\overline{\eta} + t(\widehat{\eta} - \overline{\eta})\right)\right)\, dt,$$

taken along the straight-line path from the baseline score $\overline{\eta}$ to the fitted score $\widehat{\eta}$. The horizontal line indicates the average residual over this path. Multiplying this average residual by the score contribution $\widehat{\eta}_j$ of a component yields that component's contribution to the improvement in log loss for the observation.

models like neural networks, however, the score components may be nonlinear in the input features. If we wish to attribute model fit to the input features in such nonlinear models, Hentschel (2026) derives a numerical solution to the line integral that looks through to the features.

4.4 Specialization to Logistic Regression

In logistic regression, the score is additive and the score components are linear in the features, so that

$$\widehat{p} = \sigma(\widehat{\eta}), \qquad \widehat{\eta} = \widehat{\beta}_0 + \sum_{j=1}^{K} \widehat{\beta}_j x_j. \tag{42}$$

Writing

$$\widehat{\eta} = \overline{\eta} + \sum_{j=1}^{K} \widehat{\eta}_j, \tag{43}$$

$$\widehat{\eta}_j = \widehat{\beta}_j x_j, \tag{44}$$

and substituting into equation (41) yields the feature-level contributions.

The baseline score $\overline{\eta}$ corresponds to the unconditional class probability and differs from the fitted intercept when regressors are not centered. This distinction has no effect on the decomposition: constant shifts cancel, so attribution is carried entirely by the components $\widehat{\eta}_j$.

As in linear regression, contributions depend on how components interact with the full fitted signal. Here, importance is determined by alignment with residual uncertainty rather than covariance with the outcome.

4.5 Weighted Evaluation

The decomposition extends directly to weighted evaluation. Let $w_i \geq 0$ define weights and $\mathbb{E}_w[g] = \sum_i \widetilde{w}_i g_i$. Replacing expectations by $\mathbb{E}_w$ yields an exact additive decomposition of weighted log-loss improvement.

Weights define the metric in which predictive fit is evaluated, not corrections to the model. This interpretation preserves model-conditional attribution.

4.6 Standard Errors

Each contribution C_j is a sample average of observation-level terms $c_{ij} = \widehat{\eta}_{ij} w_i$, where w_i is the path weight implied by equation (41). Standard errors follow directly,

$$SE(C_j) = \sqrt{\frac{1}{N}\,\mathbb{E}\big[(c_{ij} - C_j)^2\big]}. \tag{45}$$

As in regression, these standard errors reflect sampling variability in the evaluation data, conditional on the fitted model, and apply both in-sample and out-of-sample.

5 Relation to Existing Measures

Feature-importance measures differ fundamentally in the question they are designed to answer. In the introduction, we organized these questions into three categories: model exploration, prediction explanation, and model-fit attribution. Table 1 maps existing measures onto these categories. The table makes clear that EDEF occupies a distinct position: it is global, targets predictive accuracy, and evaluates the model exclusively at realized inputs. The discussion below focuses on the differences that matter most in practice, since the literature often treats feature-importance methods as largely interchangeable.

Table 1: Conceptual Classification of Feature-Importance Measures

Method	Object	Model fixed	Realized inputs	Scope	SEs avail.	Process
Coefficients	Pred/Score	Y	Y	Glo	N	Local sens
IG	Pred	Y	N	Loc	N	Cont path
SHAP	Pred	N	N	Loc agg	N	Discrt avg
Perm./Pert.	Acc/Pred	N	N	Glo	N	Discrt rem
SAGE	Acc	N	N	Glo	N	Discrt avg
EDEF	Acc	Y	Y	Glo	Y	Cont path

The table summarizes characteristics of feature-importance measures. *Object*: what the method attributes – predictions (Pred), predictive accuracy (Acc), or the fitted score (Score). *Model fixed*: whether the method conditions on the actual deployed model evaluated at the realized inputs, without counterfactual feature removal, marginalization, or perturbation. *Realized inputs only*: whether the method evaluates the model exclusively at realized inputs, requiring no additional model evaluations. *Scope*: whether attribution is local, at the level of individual predictions (Loc); aggregated across observations (Loc agg); or global, at the level of the fitted model (Glo). *SE avail.*: whether standard errors for the importance estimates are naturally available without resampling. These reflect variation in the sample and are different from Monte Carlo approximation errors. *Process*: how importance is allocated – local sensitivity (Local sens), discrete feature removal or averaging (Discrt rem/avg), or a continuous path integral (Cont path).

5.1 SAGE

Covert, Lundberg, and Lee (2020) introduce SAGE, which applies the Shapley (1953) framework to global model performance. Like EDEF, SAGE is global, accuracy-oriented, and additive.

The two methods differ fundamentally in how they define a feature's contribution. SAGE is a feature-removal method: it measures how much predictive performance would degrade if a feature were replaced by draws from its marginal distribution. This is a counterfactual question about feature availability. EDEF is a decomposition method: it measures how much each component of the fitted signal contributed to the model's realized predictive accuracy on the evaluation data. This is a factual question about the deployed model.

This distinction collapses in the special case of quadratic loss and linear predictions, where the two methods coincide exactly, as shown in Section 3.4. In general, however, feature removal changes the model. When SAGE marginalizes over a feature, it evaluates the fitted model at inputs outside its training support and disrupts the joint distribution of the inputs. The resulting performance gap reflects both the feature's importance and the model's inability to substitute other features for it. EDEF never does this. The fitted model is always evaluated at inputs it was actually given, so the attribution

is model-conditional in a strict sense: it describes how the model generates its performance, not how performance would change under hypothetical modifications.

The divergence between the two methods is not bounded outside the special case of quadratic loss and linear predictions. For nonlinear models, SAGE evaluates the fitted model at counterfactual inputs that may lie far from the training support, where model behavior is unconstrained. The classification example in Section 6.3 illustrates meaningful divergence even for a simple logistic model; for genuinely nonlinear models the gap can be arbitrarily large. The extension of EDEF to nonlinear models is developed in Hentschel (2026).

SAGE also inherits the computational cost of Shapley methods and typically requires Monte Carlo approximation. Standard errors are not naturally available.

5.2 Prediction Explanation Methods

SHAP (Lundberg and Lee, 2017), integrated gradients (Sundararajan et al., 2017), and related local methods explain individual predictions by attributing them to input features relative to a baseline. They do not attribute predictive accuracy. This distinction is consequential. A feature can make large contributions to predictions while simultaneously degrading predictive fit – particularly out of sample, where the model may be overconfident or miscalibrated. Methods that decompose predictions cannot detect this, because they never evaluate alignment between predictions and outcomes.

SHAP and integrated gradients also differ from EDEF in scope. Both are local methods that attribute individual predictions; aggregating them across observations yields summaries of prediction contributions but not an exact decomposition of a global fit measure such as mean squared error or expected log loss.[8] Integrated gradients use a path-integral construction along input paths, as does EDEF, but integrate the gradient of the prediction function rather than the gradient of the loss. Applied to the loss instead of predictions, integrated gradients become a numerical approximation to EDEF; in the settings considered here, EDEF provides the exact closed-form solution.

Perturbation and permutation methods (Breiman, 2001; Fisher et al., 2019) measure the decline in performance when a feature is disrupted. Unlike SHAP and integrated gradients, they do target predictive accuracy rather than predictions. But they do so by evaluating the model at modified inputs

[8] It is possible to apply SHAP to loss functions directly. However, this yields a decomposition of loss at individual observations rather than a decomposition of global expected loss, which is what SAGE and EDEF target.

and therefore measure sensitivity to disruption rather than contribution to realized fit. They are not additive and depend on the perturbation scheme and feature correlations.

5.3 Model Exploration Measures

Measures in the model exploration tradition – including partial R^2, dominance analysis (Budescu, 1993), and Shapley-value decompositions of explained variance – assess the value of features by refitting or removing them. Because they do not condition on a fixed model, they do not attribute realized predictive accuracy.

The Pratt decomposition (Pratt, 1987) is a partial exception: it allocates explained variance in the estimation sample without refitting. It coincides with EDEF for in-sample OLS, where orthogonality holds, but diverges outside this setting. Pratt attributes variance, not predictive accuracy, and does not apply out of sample or under regularization.

5.4 Summary

The measures discussed above answer different questions by construction. The table makes this explicit. Exploration methods vary the model; prediction explanation methods vary the inputs or attribute predictions rather than accuracy; perturbation methods measure sensitivity to disruption. None attributes the realized predictive accuracy of a fixed model evaluated on actual data.

EDEF addresses this specific problem. It is exact, additive, global, and model-conditional, and it admits standard errors that enable formal monitoring and statistical inference – a capability that none of the existing methods provides directly. When the objective is to attribute realized predictive accuracy of a fixed model, the problem admits a simple and exact solution.

6 Illustrations

We illustrate EDEF with a numerical verification and empirical examples. The simulations verify the accounting identities, the divergence from variance-based attributions, and the accuracy of the standard errors in a controlled setting. The empirical examples demonstrate the model evaluation use case: fit a model once, evaluate EDEF contributions on held-out data, and use standard errors to assess whether the contributions are statistically reliable.

6.1 Numerical Verification

Table 2 verifies the properties of EDEF in a controlled linear data-generating process. We simulate samples of 500 observations for fitting and 500 for

Table 2: Monte Carlo Simulations

Panel A. Simulation characteristics

R^2	$\mathbb{E}[R^2_{oos}]$	$\mathbb{E}[\lvert \sum_j C_j - \Delta\mathcal{L}\rvert]$	$\Pr(\Delta\mathcal{L} < 0)$
0.60	0.599	0	0
0.30	0.299	0	0
0.10	0.099	0	0.000
0.02	0.020	0	0.056

Panel B. Comparison to Pratt

	In Sample OLS	Out of Sample OLS	Out of Sample Elastic Net
R^2	$\mathbb{E}[\max_j \lvert C_j - P_j\rvert]$	Med $\left(\max_j \lvert C_j/\Delta\mathcal{L} - P_j/\sum_j P_j\rvert\right)$	
0.60	0	0.015	0.006
0.30	0	0.029	0.010
0.10	0	0.075	0.022
0.02	0	0.591	0.124

Panel C. Variability of Contributions (Target $R^2 = 0.30$)

ρ	$\Pr(C_j < 0)$	Negative mass $\mathbb{E}[\sum_{C_j<0} \lvert C_j\rvert/\Delta\mathcal{L}]$	Standard Errors 95% coverage
0.00	0.029	0.002	0.949
0.30	0.078	0.006	0.949
0.60	0.183	0.038	0.949
0.90	0.200	0.203	0.948

The table reports Monte Carlo simulations with sample size 500 in the separate training and evaluation samples across 100,000 replications. The data-generating process is a linear model with $K = 5$ features and coefficients $\{1.0, 0.6, 0.0, -0.4, 0.2\}$. Regressors are jointly normal with mean zero, unit variance, and pairwise correlation $\rho_{ij} = \rho^{|i-j|}$. In Panels A and B, $\rho = 0.7$; panel C varies ρ. We set noise variance to target the listed population R^2 values. Entries reported as 0 are numerically zero within machine precision.

Euler contributions are $C_j = 2\,\mathrm{Cov}(y, \widehat{y}_j) - \mathrm{Cov}(\widehat{y}, \widehat{y}_j)$ and sum to $\Delta\mathcal{L} = \mathrm{Var}(y) - MSE(y - \widehat{y})$ on each evaluation sample, up to floating-point error. Pratt contributions are $P_j = \mathrm{Cov}(y, \widehat{y}_j)$. (We use the covariance-form for Pratt components, which coincide with Pratt's correlation-based formulation for standardized variables.) The corresponding proportional attributions are $C_j/\Delta\mathcal{L}$ and $P_j/\sum_j P_j$.

Panel A reports average out-of-sample R^2, the absolute add-up error $\lvert \sum_j C_j - \Delta\mathcal{L}\rvert$, and the frequency with which $\Delta\mathcal{L} < 0$. Panel B reports medians of the indicated discrepancies between Euler and Pratt attributions under OLS and Elastic Net estimation. Panel C reports the frequency of negative Euler contributions, the share of total contribution mass attributable to negative C_j, and empirical coverage of analytical 95% confidence intervals.

evaluation, with results aggregated over 100,000 replications. Outcomes follow a linear model with correlated Gaussian features and known coefficients, allowing direct comparison between sample estimates and population quantities.

Panel A verifies the accounting identities. By construction, EDEF contributions sum exactly to the improvement in mean squared error relative to the baseline in every sample. The simulations confirm that this identity holds numerically and that predictive accuracy can be negative in finite samples, particularly when signal strength is low.

Panel B illustrates the relationship between EDEF and the Pratt decomposition. In the training sample under ordinary least squares, the two coincide exactly. Out of sample, they diverge because orthogonality no longer holds: proportional attributions based on predicted variance differ from those based on predictive accuracy, with discrepancies increasing as signal strength declines. Regularization reduces estimation noise but does not restore equivalence.

Panel C examines the role of feature correlation and the behavior of standard errors. Negative contributions arise from interaction among correlated features but are typically small unless correlations are high. The reported coverage confirms that the standard errors derived in appendix B accurately reflect sampling variability in this setting.

The simulations serve as a numerical check of the theory: they verify exact additivity, illustrate divergence from variance-based measures outside the OLS setting, and confirm the validity of the standard errors under a known data-generating process.

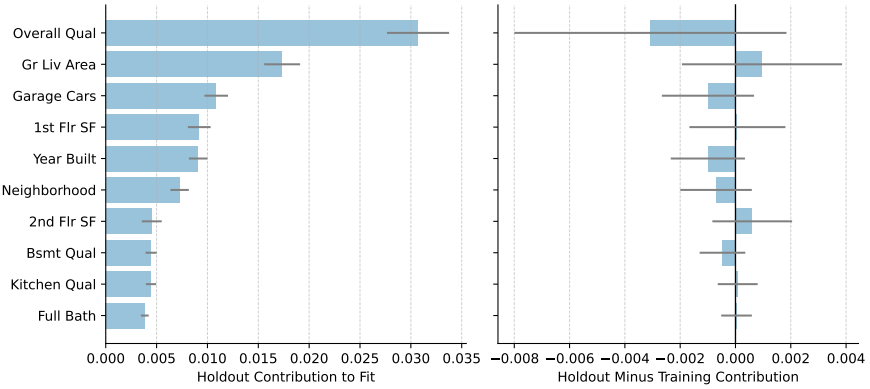
6.2 Ames Housing: Regression

We use the Ames housing dataset, a standard benchmark for tabular regression. The outcome is the log sale price. There are 2,930 total observations and 80 base features, including a mix of numerical variables and categorical attributes (e.g., neighborhood, quality ratings). We preprocess all variables using median imputation, standardization, and one-hot encoding. After one-hot encoding the categorical features, there are about 290 features.[9] When reporting feature importance, we sum the contributions from all the dummies corresponding to a single categorical feature.

We estimate a ridge regression model with the regularization parameter selected via cross-validation. We evaluate feature importance for the training

[9] The number of expanded features varies slightly across samples because we apply one-hot encoding after the train-test split, and rare categorical levels may be absent in a given training sample.

Figure 3: Comparing Feature Importance for Regressions



The figure shows EDEF feature importance for the most important features in a Ridge regression model for the Ames housing data.

The left panel reports contributions to fit in levels for the hold-out sample, with error bars indicating plus and minus two standard errors. The standard errors reflect sampling variability in the evaluation data, conditional on the fitted model, and therefore support statistical inference.

The right panel reports differences in contributions to fit between the hold-out and training samples. The corresponding standard errors combine variability from both samples under the assumption that estimation errors are uncorrelated across the non-overlapping samples.

sample and for a holdout sample. The samples are randomly chosen, non-overlapping halves of the full data set.

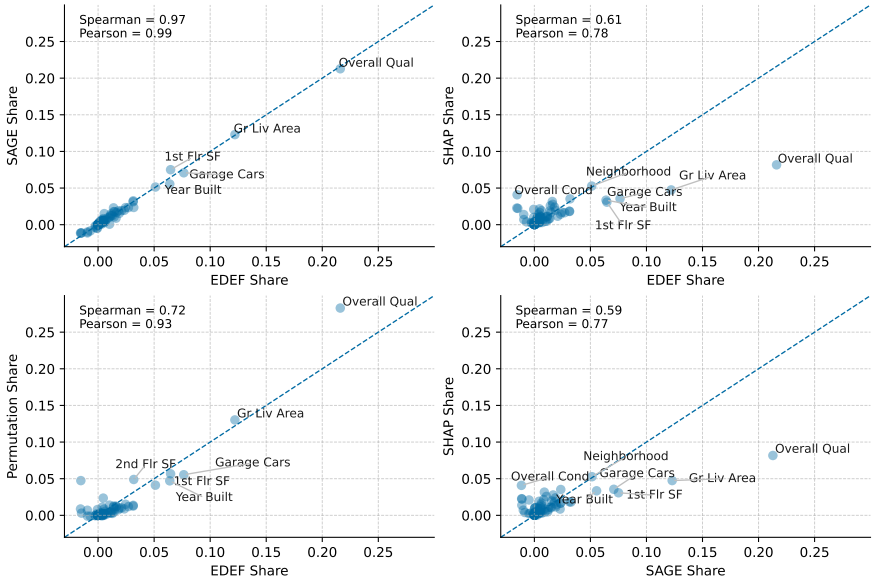
The left bar chart in figure 3 reports EDEF contributions in levels together with two-standard-error bands for the top 10 features based on the hold out sample. The top features make statistically significant contributions to model fit but the difference in importance of many adjacent features is statistically insignificant. The right bar chart reports the changes in contributions relative to the training sample, also with two-standard-error bands. The standard errors for the changes use the variability for both samples and assume that the estimation errors are uncorrelated across the samples.[10]

This illustrates a key advantage of the EDEF framework: contributions can be interpreted as sample averages, allowing us to construct standard errors and statistical inference in a straightforward manner.

Figure 4 reports pairwise comparisons of proportional contributions to model fit across methods. The scatter plots show strong agreement between

[10] When feature contributions are evaluated for many components, testing for changes in importance across samples raises a multiple testing problem. Simple feature-by-feature inference can therefore be misleading. Because EDEF contributions have standard errors that reflect sampling variability in the evaluation data, they can be combined with standard multiple-testing procedures, such as false discovery rate control (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001), to support valid inference across large numbers of features. Related issues arise in model selection and iterative refinement, where data-driven specification search can invalidate naive inference; these concerns are addressed by model-search procedures such as Hansen (2005).

Figure 4: Comparing Feature Importance for Regressions



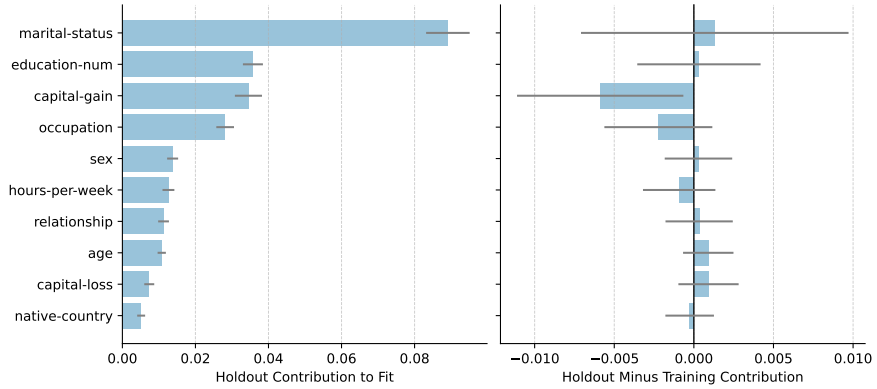
The figure compares feature importance shares for a hedonic Ridge regression model using the Ames housing data. The EDEF and SAGE methods target predictive accuracy, with EDEF decomposing realized fit and SAGE evaluating counterfactual performance under feature removal. Permutation importance measures the change in performance under feature perturbations, while SHAP values reflect average contributions to predictions.

EDEF and SAGE, with points lying close to the 45-degree line. This is expected in the regression setting with quadratic loss. In this case, EDEF coincides with the Shapley decomposition of model fit, and SAGE provides a Monte Carlo approximation to the same object.

SHAP and permutation importance exhibit systematic deviations from EDEF. Both methods tend to attribute higher importance to features that are correlated with other predictors, reflecting their reliance on perturbations or conditional expectations that break the correlation structure of the data. These differences are economically meaningful and not attributable to sampling variation.

The apparent discrepancy reflects a difference in the question being answered. Overall Condition is indeed informative about house prices, but in this dataset it is strongly correlated with other structural features such as overall quality, age, and size. The fitted model therefore distributes this information across multiple components. Once the full fitted signal is taken into account, Overall Condition contributes little additional alignment with the outcome. EDEF and SAGE attribute realized predictive accuracy to the components that actually generate it in the fitted model, and therefore assign less credit to partly redundant features.

Figure 5: Feature Importance for Classification



The figure shows EDEF feature importance for the most important features in a regularized logistic regression model for the UCI adult income data.

The left panel reports contributions to fit in levels for the hold-out sample, with error bars indicating plus and minus two standard errors. The standard errors reflect sampling variability in the evaluation data, conditional on the fitted model, and therefore support statistical inference.

The right panel reports differences in contributions to fit between the hold-out and training samples. The corresponding standard errors combine variability from both samples under the assumption that estimation errors are uncorrelated across the non-overlapping samples.

Overall, the regression example shows that EDEF and SAGE deliver nearly identical decompositions of model fit, while SHAP and permutation importance answer different questions.

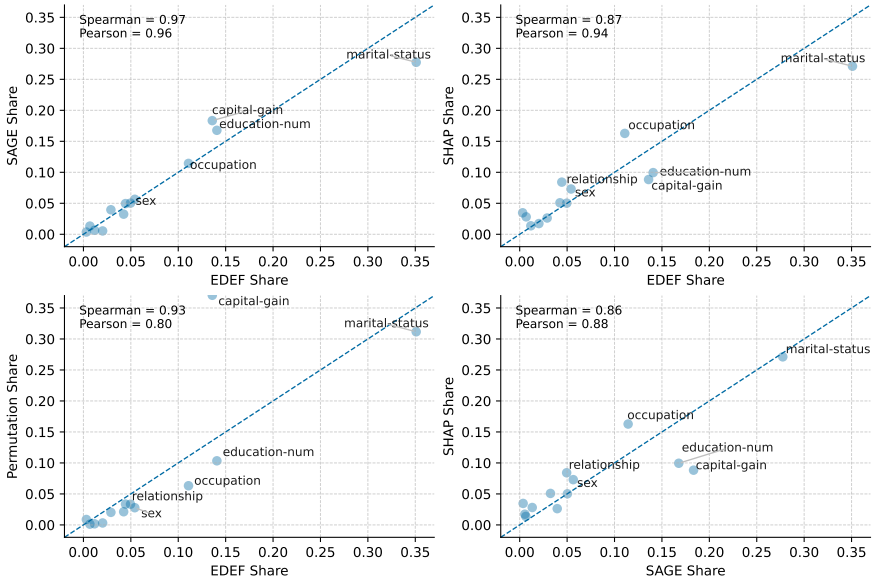
6.3 UCI Adult Income: Classification

We next consider a binary classification problem based on the UCI Adult income dataset. The dataset is a standard benchmark for binary classification. The outcome indicates whether annual income exceeds $50,000. There are 48,842 total observations and 13 base features. Features include demographic and employment characteristics, preprocessed using median or mode imputation and one-hot encoding. After one-hot encoding the categorical features, there are about 100 features. We estimate an L2-regularized logistic regression model with the regularization parameter selected via cross-validation. Feature importance is evaluated out of sample on a random holdout set under log loss. The holdout sample is half of the total sample.

Figure 5 shows EDEF feature importance for the top 10 features. As in the regression case, we report the EDEF contributions and their changes across samples with standard errors. This again highlights the ability to perform inference on feature contributions, which is not directly available for the alternative methods.

In the classification setting, the decomposition includes a baseline adjustment component. We normalize the reported proportional contributions

Figure 6: Comparing Feature Importance for Classification



The figure compares the feature importance shares for a logistic regression classification model using the UCI adult income data. The EDEF and SAGE methods target predictive accuracy, with EDEF decomposing realized fit and SAGE evaluating counterfactual performance under feature removal. Permutation importance measures the change in performance under feature perturbations, while SHAP values reflect average contributions to predictions.

over the displayed features to ensure comparability across methods.

Figure 6 compares feature importance across methods. In contrast to the linear regression case, where EDEF and SAGE should converge, this is no longer true for classification models. Although the results from the two methods are broadly similar, the scatter plot also reveal noticeable deviations from the 45-degree line for several features.

This difference reflects the nonlinear structure of the log loss. EDEF attributes contributions along the score path of the fitted model, while SAGE averages marginal contributions over feature coalitions. Under quadratic loss these approaches coincide, but under log loss they need not agree exactly.

The largest discrepancies arise for features with strong effects on predicted probabilities, particularly in regions where the log loss is highly nonlinear. In these cases, the path-based attribution used by EDEF and the coalition-based averaging used by SAGE capture different aspects of model behavior. Despite these differences, the two methods remain qualitatively aligned, identifying a similar set of important features.

In contrast, SHAP and permutation importance continue to display systematic deviations, consistent with their reliance on perturbations that alter the joint distribution of the inputs.

Table 3: Runtime Comparison for Feature Importance Methods

Method	Runtime (sec)	Relative to EDEF
Panel A: Ames Housing Ridge Regression		
EDEF	0.0027	1
SHAP	0.013	4.8
Permutation	12	4,500
SAGE	280	100,000
Panel B: UCI Adult Income Logistic Regression		
EDEF	0.025	1
SHAP	0.19	7.8
Permutation	2.7	110
SAGE	860	34,000

The table reports relative timings of the feature importance calculations for the Ames housing regressions. All values are rounded to two significant figures and are intended to convey approximate computational cost rather than precise measurements.

The Ames housing Ridge regressions use 1,465 observations and 80 raw features. The UCI adult income logistic regressions use 24,421 observations and 13 base features. In each case, the number of observations represents a random hold out sample containing half of all available observations.

After one-hot encoding all categorical features, there are approximately 290 features in the Ames housing regressions and approximately 100 features in the UCI adult income logistic regressions. The number of expanded features varies slightly across samples because we apply one-hot encoding after the train-test split, and rare categorical levels may be absent in a given training sample.

To speed up computations, SAGE uses estimation samples of 250 observations for the Ames Ridge regressions and 1,000 observations for the UCI logistic regressions. All other methods use the full set of provided observations.

Permutations permute the base features. EDEF, SHAP, and SAGE see the expanded features with one-hot dummies for all categorical variables.

6.4 Computational Cost

Table 3 reports runtimes for the different methods in both examples.

We compute SHAP values using the `SHAP` Python package (Lundberg and Lee, 2017), SAGE values using the `SAGE-Importance` package (Covert et al., 2020), and permutation importance using the implementation in `Scikit-learn` (Breiman, 2001; Pedregosa et al., 2011). We generally use default settings. For SAGE, however, we use estimation samples of 250 observations for the Ames Ridge regressions and 1,000 observations for the UCI logistic regressions to speed up computations. All other methods use the full set of provided observations.

EDEF is effectively instantaneous, requiring only a single pass through the evaluation data. SHAP is also fast due to analytic shortcuts available for linear models. Permutation importance is slower, reflecting repeated model evaluations for each feature.

SAGE is substantially more computationally intensive. Even when evaluated on a reduced background and evaluation sample, SAGE requires several minutes to complete, compared with fractions of a second for EDEF. This reflects SAGE's need to average marginal contributions over feature coalitions via Monte Carlo sampling.

The large difference in runtime is not due to implementation details but to the underlying structure of the methods. EDEF exploits the additive structure of the fitted model to compute contributions in closed form, while SAGE approximates the same object through repeated model evaluation.

In practical settings, this difference can be decisive. EDEF provides exact contributions at negligible computational cost, whereas SAGE may be infeasible to compute at scale without substantial approximation.

Because EDEF contributions are sample averages of per-observation terms, their computation scales linearly in the sample size for a fixed model. This structure also allows straightforward subsampling for extremely large datasets: contributions computed on a subsample remain unbiased estimates of the full-sample quantities, with standard errors decreasing at the usual rate. In contrast, methods such as SAGE require sampling over both observations and feature coalitions, so subsampling affects both statistical precision and approximation error.

7 Conclusion

Feature-importance methods differ in the questions they are designed to answer. Much of the existing literature focuses on model exploration or on explaining individual predictions. This paper addresses a different question: how much did each feature contribute to the realized predictive accuracy of a fixed fitted model? We call the resulting method EDEF – the Euler Decomposition of Explained Fit.

The answer follows directly once model fit is defined as the reduction in expected loss relative to a baseline predictor. In regression, this quantity is a difference of homogeneous functions of the fitted signal, and Euler's theorem yields an exact additive decomposition. In classification under log loss, homogeneity fails, but the same logic extends through a path-integral representation based on the fundamental theorem of calculus. In both cases, the decomposition is exact, additive, and model-conditional: it attributes realized predictive accuracy to the components of the fitted model evaluated at the realized inputs, without refitting or counterfactual evaluation.

The simplicity of this result reflects the structure of the problem. EDEF requires only realized predictions and their additive components, is compu-

tationally trivial once predictions are available, and applies to any model with an additive signal representation, including linear and regularized regressions, generalized linear models, generalized additive models, and ensemble methods.

Because EDEF expresses feature importance as a sample average, it also admits standard errors that quantify sampling variability in the evaluation data. These standard errors apply both in-sample and out-of-sample and require no additional computation. They enable formal monitoring of feature contributions over time or across samples, allowing changes in model behavior to be distinguished from noise. For established or deployed models, this capability is central.

Finally, computing EDEF contributions to fit is materially faster than other mainstream feature importance measures. Some methods are orders of magnitude more expensive to compute than EDEF.

The companion paper Hentschel (2026) extends this framework to nonlinear models, where attribution is carried out in input space using a path-integral construction. Together, the two papers establish a unified and tractable framework for attributing predictive accuracy across a broad class of regression and classification models.

8 References

Benjamini, Yoav, and Yosef Hochberg, 1995, Controlling the false discovery rate: A practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society, Series B* 57 (1), 289–300.

Benjamini, Yoav, and Daniel Yekutieli, 2001, The control of the false discovery rate in multiple testing under dependence, *Annals of Statistics* 29 (4), 1165–1188.

Breiman, Leo, 2001, Random forests, *Machine Learning* 45, 5–32.

Budescu, David V., 1993, Dominance analysis: A new approach to the problem of relative importance of predictors in multiple regression, *Psychological Bulletin* 114 (3), 542–551.

Covert, Ian, Scott M. Lundberg, and Su-In Lee, 2020, Understanding global feature contributions with additive importance measures, in *Advances in Neural Information Processing Systems*, volume 33, 17212–17223.

De Cock, Dean, 2011, Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project, *Journal of Statistics Education* 19 (3), 1–15.

Dua, Dheeru, and Casey Graff, 2019, UCI machine learning repository.

Fisher, Aaron, Cynthia Rudin, and Francesca Dominici, 2019, All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, *Journal of Machine Learning Research* 20 (1), 1–81.

Gneiting, Tilmann, and Adrian E. Raftery, 2007, Strictly proper scoring rules, prediction, and estimation, *Journal of the American Statistical Association* 102 (477), 359–378.

Hansen, Peter R., 2005, A test for superior predictive ability, *Journal of Business & Economic Statistics* 23 (4), 365–380.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman, 2009, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, second edition (Springer, New York).

Hentschel, Ludger, 2026, Feature importance: Decomposing model fit in nonlinear regressions, Working paper, Versor Investments, New York, NY.

Hoerl, Arthur E., and Robert W. Kennard, 1970, Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* 12 (1), 55–67.

Kohavi, Ron, 1996, Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 202–207.

Kruskal, William, 1987, Relative importance by averaging over orderings, *The American Statistician* 41 (1), 6–10.

Lindeman, Richard H., Peter F. Merenda, and Ruth Z. Gold, 1980, *Introduction to Bivariate and Multivariate Analysis* (Scott, Foresman, Glenview, IL).

Lundberg, Scott M., and Su-In Lee, 2017, A unified approach to interpreting model predictions, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4768–4777 (Curran Associates Inc., Red Hook, NY).

Owen, Guillermo, 1977, Values of games with a priori unions, in Rudolf Henn, and Otto Moeschlin, eds., *Mathematical Economics and Game Theory*, volume 141 of *Lecture Notes in Economics and Mathematical Systems*, 76–88 (Springer, Berlin, Heidelberg).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, 2011, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12, 2825–2830.

Pratt, John W., 1987, Dividing the indivisible: Using simple symmetry to partition variance explained, in Timo Pukkila, and Simo Puntanen, eds., *Proceedings of the Second International Conference in Statistics*, 245–260 (University of Tampere, Tampere, Finland).

Shapley, Lloyd S., 1953, A value for n-person games, *Contributions to the Theory of Games* 2, 307–317.

Silberberg, Eugene, 1978, *The Structure of Economics: A Mathematical Analysis* (McGraw–Hill, New York, NY).

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan, 2017, Axiomatic attribution for deep networks, in Doina Precup, and Yee Whye Teh, eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3319–3328 (PMLR).

Tasche, Dirk, 2008, Capital allocation to business units and sub-portfolios: The Euler principle, in Andrea Resti, ed., *Pillar II in the New Basel Accord: The Challenge of Economic Capital*, 423–453 (Risk Books, London).

Thomas, D. Roland, Edward Hughes, and Bruno D. Zumbo, 1998, On variable importance in linear regression, *Social Indicators Research* 45, 253–275.

Tibshirani, Robert, 1996, Regression shrinkage and selection via the Lasso, *Journal of the Royal Statistical Society, Series B* 58 (1), 267–288.

Zou, Hui, 2006, The adaptive Lasso and its oracle properties, *Journal of the American Statistical Association* 101 (476), 1418–1429.

Zou, Hui, and Trevor Hastie, 2005, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67 (2), 301–320.

A Euler Decompositions and Path Integrals

This appendix clarifies the relationship between Euler's theorem for homogeneous functions and the path-integral decomposition used in the main text. The latter is the natural generalization of the former when homogeneity does not hold.

The decomposition is expressed in terms of gradients and integrals, but it is not an approximation. The path-integral identity is an exact consequence of the fundamental theorem of calculus and does not rely on a Taylor expansion or on evaluating derivatives at a single point.

A.1 A Path-Integral Identity for Differentiable Functions

Let $f : \mathbb{R}^K \to \mathbb{R}$ be continuously differentiable, and let $x, x_0 \in \mathbb{R}^K$. Consider the straight-line path

$$x(t) = x_0 + t(x - x_0), \qquad t \in [0, 1]. \tag{46}$$

By the chain rule,

$$\frac{d}{dt} f(x(t)) = \nabla f(x(t))^\top (x - x_0). \tag{47}$$

Applying the fundamental theorem of calculus yields the exact identity

$$f(x) - f(x_0) = \int_0^1 \nabla f(x(t))^\top (x - x_0)\, dt. \tag{48}$$

Rewriting the inner product gives an additive decomposition across coordinates,

$$f(x) - f(x_0) = \sum_{j=1}^{K} (x_j - x_{0j}) \int_0^1 \frac{\partial f}{\partial x_j}(x(t))\, dt. \tag{49}$$

This identity expresses the change in f as the sum, over coordinates, of each increment $(x_j - x_{0j})$ multiplied by its average marginal effect along the path from x_0 to x.

A.2 Euler's Theorem as a Special Case

Now suppose that f is positively homogeneous of degree one,

$$f(\lambda x) = \lambda f(x) \qquad \text{for all } \lambda > 0. \tag{50}$$

Euler's theorem states

$$f(x) = \sum_{j=1}^{K} x_j \frac{\partial f}{\partial x_j}(x). \tag{51}$$

This follows directly from the path-integral identity. Take $x_0 = 0$ and $x(t) = tx$. Then

$$f(x) - f(0) = \int_0^1 \nabla f(tx)^\top x \, dt. \tag{52}$$

For a function homogeneous of degree one, the gradient $\nabla f(tx)$ is homogeneous of degree zero and therefore constant along rays, so $\nabla f(tx) = \nabla f(x)$ for all $t > 0$. The integrand is therefore constant in t, and the integral reduces to

$$\nabla f(x)^\top x, \tag{53}$$

which is Euler's theorem. See Silberberg (1978) or Tasche (2008).

A.3 Interpretation

Euler's theorem recovers the function value from derivatives evaluated at a single point because homogeneity eliminates variation along the path from the origin. When homogeneity does not hold, derivatives vary along this path, and the correct generalization is to average them.

The path-integral identity expresses the change in a function as the integral of its directional derivative along a path connecting a baseline to an evaluation point. The straight-line path used here treats all components symmetrically, preserves additivity aligned with the fitted model, and reduces exactly to Euler's theorem when homogeneity holds. Alternative paths introduce ordering or counterfactual structure that is not implied by the fitted model.

The decomposition also relies on the existence of a meaningful baseline and evaluation point. In predictive modeling, these arise naturally from the baseline and fitted models that define explained fit. In this setting, the decomposition has a direct interpretation as an attribution of model performance.

The constructions used in the main text follow this principle. In regression, explained fit is homogeneous in the fitted signal, so Euler's theorem yields an endpoint decomposition. In classification, log loss is not homogeneous in the fitted score, so an exact decomposition requires averaging derivatives along the score path. Both cases are exact and depend only on the structure

of the evaluation metric, not on the estimation procedure that produced the fitted model.

B Standard Errors

This appendix derives standard errors for the Euler contributions to regression fit,

$$C_j = 2\,\mathrm{Cov}(y, \widehat{y}_j) - \mathrm{Cov}(\widehat{y}, \widehat{y}_j), \tag{54}$$

for a fixed model $\widehat{y} = \sum_{k=1}^{K} \widehat{y}_k$ with additive structure.

The key observation is that each contribution C_j can be written as a sample average of observation-level quantities. Standard errors therefore follow from elementary arguments for sample means. Throughout, we condition on the fitted prediction function $\widehat{y}(\cdot)$ and treat the model as fixed. Inference reflects sampling variability in the evaluation data, not uncertainty from re-estimation.

B.1 Euler Contributions as Sample Averages

It is convenient to express each contribution as a single covariance. Define

$$a_i = 2y_i - \widehat{y}_i, \qquad b_{ij} = \widehat{y}_{ij}, \tag{55}$$

and center both variables by subtracting their sample means. Then

$$C_j = \mathrm{Cov}(a, b_j) = \mathbb{E}[a_i\, b_{ij}]. \tag{56}$$

Define observation-level contributions

$$c_{ij} = a_i\, b_{ij}, \qquad c_i = (c_{i1}, \ldots, c_{iK})^\top. \tag{57}$$

Then the vector of Euler contributions is simply

$$C = (C_1, \ldots, C_K)^\top = \mathbb{E}[c_i], \tag{58}$$

that is, each C_j is a sample mean.

B.2 Variance and Standard Errors

Under i.i.d. sampling, define the population covariance of c_i,

$$\Sigma = \mathrm{Cov}(c_i) = \mathbb{E}\left[(c_i - \mathbb{E}[c_i])(c_i - \mathbb{E}[c_i])^\top\right]. \tag{59}$$

We estimate Σ using the sample covariance

$$\widehat{\Sigma} = \mathbb{E}\left[(c_i - C)(c_i - C)^\top\right]. \tag{60}$$

Because C is a sample average over N observations, $\mathrm{Cov}(C) = \Sigma/N$. The standard error for C_j is therefore

$$SE(C_j) = \sqrt{\frac{1}{N}\,\widehat{\Sigma}_{jj}} = \sqrt{\frac{1}{N}\,\mathbb{E}\left[(c_{ij} - C_j)^2\right]}. \tag{61}$$

If the evaluation sample exhibits heteroskedasticity or serial dependence, $\widehat{\Sigma}$ can be replaced by a HAC estimator applied to $\{c_i\}$.

B.3 Grouped Contributions

For a group of features $G \subseteq \{1, \ldots, K\}$, define $C_G = \sum_{j \in G} C_j$. Then

$$C_G = \mathbb{E}[c_{iG}], \qquad c_{iG} = \sum_{j \in G} c_{ij}. \tag{62}$$

The corresponding standard error is

$$SE(C_G) = \sqrt{\frac{1}{N}\,\mathbb{E}[(c_{iG} - C_G)^2]}. \tag{63}$$

This univariate calculation automatically accounts for dependence across features within the group and avoids estimating the full covariance matrix.

B.4 Standard Errors for Contribution Shares

In applications, it is often useful to report proportional contributions rather than level contributions. Define total explained fit as

$$\Delta\mathcal{L} = \sum_{k=1}^{K} C_k = \mathbf{1}^\top C, \tag{64}$$

and define the share of component j as

$$S_j = \frac{C_j}{\Delta\mathcal{L}}. \tag{65}$$

The share is a smooth function of the vector C whenever $\Delta\mathcal{L} \neq 0$, so its standard error follows from the delta method.

Let e_j denote the jth unit vector. The gradient of S_j with respect to C is

$$\nabla_C S_j = \frac{1}{\Delta\mathcal{L}} e_j - \frac{C_j}{(\Delta\mathcal{L})^2}\mathbf{1}. \tag{66}$$

Since $\widehat{\mathrm{Cov}}(C) = \widehat{\Sigma}/N$, the delta-method standard error is

$$SE(S_j) = \sqrt{\frac{1}{N}\left(\frac{1}{\Delta\mathcal{L}} e_j - \frac{C_j}{(\Delta\mathcal{L})^2}\mathbf{1}\right)^{\top} \widehat{\Sigma}\left(\frac{1}{\Delta\mathcal{L}} e_j - \frac{C_j}{(\Delta\mathcal{L})^2}\mathbf{1}\right)}. \tag{67}$$

Equivalently, define the observation-level total contribution

$$d_i = \sum_{k=1}^{K} c_{ik}, \qquad \Delta\mathcal{L} = \mathbb{E}[d_i]. \tag{68}$$

Then the influence-function representation of the share is

$$s_{ij} = \frac{c_{ij} - C_j}{\Delta\mathcal{L}} - \frac{C_j}{(\Delta\mathcal{L})^2}(d_i - \Delta\mathcal{L}), \tag{69}$$

and the same standard error can be written as

$$SE(S_j) = \sqrt{\frac{1}{N}\,\mathbb{E}[s_{ij}^2]}. \tag{70}$$

This form is often easiest to implement because it avoids explicit matrix multiplication.

For a group G, define

$$S_G = \frac{C_G}{\Delta\mathcal{L}}, \qquad C_G = \sum_{j\in G} C_j. \tag{71}$$

The same formulas apply after replacing C_j by C_G, c_{ij} by c_{iG}, and e_j by the group indicator vector $\mathbf{1}_G$.

Share standard errors should be interpreted with the usual caution for ratios. They are well behaved when total explained fit is far from zero, but can be unstable when $\Delta\mathcal{L}$ is small. For this reason, level contributions remain the primary object for inference, while shares provide a convenient scale for reporting and comparing importance across methods.

B.5 Extension to Classification

The same standard error calculation applies to the classification setting. In that case, each contribution C_j can likewise be written as a sample average

$$C_j = \mathbb{E}[c_{ij}], \tag{72}$$

where $c_{ij} = \widehat{\eta}_{ij} w_i$ and w_i is the path-integral weight defined in equation (41).

Because the contributions remain sample means of observation-level quantities, the standard errors follow from the same argument as in the regression case. No modification is required.

B.6 Implementation Remarks

In-sample versus out-of-sample.

Because we condition on the fitted model, the same formulas apply in-sample and out-of-sample. Only the evaluation sample used to compute c_i and its size N differ.

OLS as a special case.

Under in-sample ordinary least squares, $\mathrm{Cov}(e, \widehat{y}_j) = 0$ exactly. The contribution reduces to

$$C_j = \mathrm{Cov}(y, \widehat{y}_j), \tag{73}$$

and the standard error formula applies directly to this simplified expression.

Auxiliary regression interpretation.

Each contribution C_j can be viewed as the intercept in a regression

$$c_{ij} = \alpha_j + \varepsilon_{ij}. \tag{74}$$

The OLS estimator $\widehat{\alpha}_j$ equals C_j, and the corresponding standard error coincides with equation (61).

C Multinomial Classification

This appendix extends the EDEF framework from binary to multinomial classification. The extension introduces no new attribution principle. It applies the same path-integral logic used in the main text to vector-valued scores in softmax models.

C.1 Setup

Consider a classification problem with $K > 2$ classes. Let $\widehat{p}_k(x)$ denote the predicted probability of class k, and choose class K as a reference category. Define the $(K-1)$-dimensional logit vector

$$s(x) = \begin{pmatrix} s_1(x) \\ \vdots \\ s_{K-1}(x) \end{pmatrix}, \qquad s_k(x) = \log \frac{\widehat{p}_k(x)}{\widehat{p}_K(x)}. \tag{75}$$

Let s_0 denote the baseline logit vector corresponding to the baseline class probabilities $\overline{p} = \mathbb{E}[y]$.

C.2 Explained Log Loss

For an observation $y \in \{1, \ldots, K\}$, define the multinomial log score

$$f(y, s) = s_y - \psi(s), \qquad \psi(s) = \log\left(1 + \sum_{k=1}^{K-1} e^{s_k}\right). \tag{76}$$

Explained fit is defined as

$$\Delta\mathcal{L} = \mathbb{E}\big[f(y, s) - f(y, s_0)\big] = \psi(s) - \psi(s_0) - \nabla\psi(s_0)^\top (s - s_0). \tag{77}$$

C.3 Path-Integral Decomposition

Define the straight-line path in logit space

$$s(t) = s_0 + t(s - s_0), \qquad t \in [0, 1]. \tag{78}$$

Applying the path-integral identity yields

$$\Delta\mathcal{L} = \int_0^1 (s - s_0)^\top \big(y - p(s(t))\big)\, dt, \tag{79}$$

where $p(s)$ denotes the vector of predicted probabilities implied by s.

Assuming an additive decomposition of the fitted score,

$$s(x) - s_0 = \sum_{j=1}^{p} s^{(j)}(x), \tag{80}$$

we obtain the multinomial EDEF contributions

$$C_j = \mathbb{E}\left[s^{(j)}(x)^\top \int_0^1 \big(y - p(s(t))\big)\, dt\right]. \tag{81}$$

This is the direct analogue of the binary classification result: each component is weighted by its alignment with the path-averaged residual $y - p(\cdot)$.

C.4 Quadratic Representation and Geometry

The improvement in log loss also admits an exact quadratic representation. Applying Taylor's theorem with integral remainder,

$$\Delta\mathcal{L} = \int_0^1 (1-t)\,(s - s_0)^\top \nabla^2\psi(s(t))(s - s_0)\, dt. \tag{82}$$

The Hessian

$$\nabla^2\psi(s) = \mathrm{diag}(p(s)) - p(s)p(s)^\top \tag{83}$$

is the Fisher information matrix of the multinomial model.

Defining the path-integrated metric

$$W_{path} = 2\int_0^1 (1-t)\,\nabla^2\psi(s(t))\, dt, \tag{84}$$

we obtain

$$\Delta\mathcal{L} = \frac{1}{2}(s - s_0)^\top W_{path}(s - s_0). \tag{85}$$

This representation shows that explained fit corresponds to a quadratic form in the displacement of the fitted signal, with curvature determined by a path-averaged Fisher metric. It provides a geometric interpretation but is not required for computing the EDEF contributions.

C.5 Standard Errors

As in regression and binary classification, each contribution can be written as a sample average

$$C_j = \mathbb{E}[c_{ij}], \tag{86}$$

where c_{ij} is the observation-level contribution.

Standard errors therefore follow directly:

$$SE(C_j) = \sqrt{\frac{1}{N}\,\mathbb{E}[(c_{ij} - C_j)^2]}. \tag{87}$$

Grouped contributions are handled identically by summing c_{ij} across features.

C.6 Summary

Multinomial classification introduces no new attribution principle. It replaces scalar quantities in the binary case with vector-valued analogues. The decomposition remains exact, additive, and model-conditional, and standard errors follow from the same sample-average structure.

D Algorithms

This appendix presents pseudocode for computing EDEF contributions and their standard errors in the regression and classification settings. Both algorithms follow the same structure: center the outcome and prediction components, compute observation-level contributions as products of centered components with a residual-like term, average across observations to obtain the EDEF contributions, and compute standard errors as the square root of the sample variance of the observation-level contributions divided by the sample size.

The two algorithms differ only in the residual-like term. For regression under squared error loss, this term is $2y_i - \widehat{y}_i$, which is linear in the outcome and the fitted value and admits direct computation from realized predictions. For classification under log loss, this term is the closed-form path weight w_i defined in equation (41), which averages the canonical residual $y_i - \sigma(\cdot)$ along the straight-line path from the baseline score to the fitted score.

In both cases, the algorithms require only a single pass through the evaluation data. Computational cost is negligible relative to model estimation. The regression algorithm accommodates weighted and generalized least squares by supplying pre-transformed outcomes and components. The classification algorithm accommodates observation weights through the weighted expectation $\mathbb{E}_w[\cdot]$ described in section 4.5.

D.1 Regression

Algorithm 1: EDEF for Regression

```
# Inputs:
```

```
# y      : (N,) vector of realized outcomes,
#            centered internally by the algorithm
# Y_hat  : (N, K) matrix of fitted signal components
#            with y_hat = sum_j Y_hat[:, j]
#
# Baseline:
# The intercept-only baseline is E[tilde y].
# All computations are performed relative to this baseline.
#
# Notes:
# For linear models with an intercept and centered regressors,
# Y_hat[:, j] = X[:, j] * beta[j] is already mean zero.
# Component centering ensures a unique attribution by assigning
# all level effects to the intercept-only baseline.
# Components are centered internally by the algorithm.
#
# For WLS or GLS, supply y and Y_hat already transformed
# by the appropriate weighting or whitening matrix.

# Center outcome (defines intercept-only baseline)
y_c = y - mean(y)

# Aggregate fitted signal and center
y_hat = Y_hat.sum(axis=1)
y_hat_c = y_hat - mean(y_hat)

# Center fitted components (normalization for attribution)
Yc = Y_hat - mean(Y_hat, axis=0)

# Euler contributions to improvement in MSE
for j in range(K):       # Can be vectorized
    C[j] = ( 2 * mean(y_c * Yc[:, j]) - mean(y_hat_c * Yc[:, j]) )

DeltaL = sum(C)  # Reduction in MSE relative to baseline

# Plain (i.i.d.) standard errors for contributions
N = len(y)
a = 2 * y_c - y_hat_c    # Observation-level term shared across
    features
for j in range(K):       # Can be vectorized
    c_ij = a * Yc[:, j]  # Observation-level contributions
    SE[j] = sqrt( mean((c_ij - C[j])**2) / N )

# Outputs:
# C          : Contributions to model fit
# SE         : Standard errors for C
# DeltaL     : Reduction in MSE relative to intercept-only baseline
# C / DeltaL : Proportional contributions
#                May be unstable if abs(DeltaL) is near 0
```

D.2 Binary Classification

Algorithm 2: EDEF for Binary Classification

```
# This algorithm computes exact Euler-style contributions under log loss,
# along with vanilla (i.i.d.) standard errors under the same evaluation
    weights.
#
# Inputs:
# y      : (N,) vector with y in {0,1}
# Eta_hat : (N, K) matrix of fitted score components
#          with eta = eta_bar + sum_j Eta_hat[:, j]
# w      : optional (N,) nonnegative observation weights defining the
#          evaluation metric (default: uniform weights)
#
# Notes:
# - For logistic regression, Eta_hat[:,j] = X[:,j] * beta[j].
# - Weights define the empirical measure used to evaluate fit (analogous to
#   WLS/GLS metrics), not ad hoc class rebalancing. All expectations below are
#   computed under these weights.
# - The baseline p_bar is the best constant predictor under the same weights.
# - For non-uniform weights, standard errors use the effective sample size
#   N_eff = 1 / sum_i wtil_i^2, where wtil are normalized weights.

# Helper functions
sigmoid(z) = 1 / (1 + exp(-z))
softplus(z) = log(1 + exp(z)) # implement stably if needed
logloss(y,p)= - y*log(p) - (1-y)*log(1-p)

# Dimensions
N, K = Eta_hat.shape

# Weighted mean helper (normalizes weights)
if w is None:
   w = ones(N)
w_sum = sum(w)
wtil = w / w_sum
wmean(a) = sum(wtil * a)

# Effective sample size under normalized weights
N_eff = 1.0 / sum(wtil**2)

# Baseline probability and baseline score (intercept-only, weighted)
p_bar  = wmean(y)
eta_bar = log(p_bar / (1 - p_bar))

# Aggregate fitted score and fitted probabilities
eta = eta_bar + Eta_hat.sum(axis=1)
p   = sigmoid(eta)
```

```
# Baseline and fitted log loss (fit improvement, weighted)
L_bar = wmean(logloss(y, p_bar))
L_hat = wmean(logloss(y, p))
DeltaL = L_bar - L_hat

# Closed-form path weight under log loss:
#   w_i^{path} = y_i - ( softplus(eta_i) - softplus(eta_bar) ) / delta_i,
# with the limit w_i^{path} -> y_i - sigmoid(eta_bar) as delta_i -> 0.
delta     = eta - eta_bar
sp_eta    = softplus(eta)     # (N,)
sp_eta_bar = softplus(eta_bar) # scalar

eps = 1e-12
w_path = zeros(N)
mask = abs(delta) > eps
w_path[mask] = y[mask] - (sp_eta[mask] - sp_eta_bar) / delta[mask]
w_path[~mask]= y[~mask] - sigmoid(eta_bar)

# Observation-level contributions and component contributions (weighted)
# c_ij = Eta_hat[i,j] * w_path[i], C[j] = E_w[c_ij]
c = Eta_hat * w_path[:,None]                 # (N, K)
C = sum(wtil[:,None] * c, axis=0)            # (K,)

# Plain (i.i.d.) standard errors for contributions under the same weights
# Var_w(c_j) = E_w[(c_ij - C[j])^2], SE(C_j) = sqrt( Var_w(c_j) / N_eff )
var_w = sum(wtil[:,None] * (c - C[None,:])**2, axis=0) # (K,)
SE    = sqrt(var_w / N_eff)                      # (K,)

# Output:
# C satisfies sum_j C[j] = DeltaL exactly (up to floating-point error)
# SE are vanilla standard errors under the evaluation weights
# Proportional importance: C / sum(C) = C / DeltaL
```