An Empirical Study of Downstream Analysis Effects of Model Pre-Processing Choices

Jessica M. Rudd, Herman “Gene” Ray

Analytics and Data Science Institute, College of Software and Computer Engineering, Kennesaw State University, Kennesaw, GA, USA
Email: jessirudd.com

Abstract

This study uses an empirical analysis to quantify the downstream analysis effects of data pre-processing choices. Bootstrap data simulation is used to measure the bias-variance decomposition of an empirical risk function, mean square error (MSE). Results of the risk function decomposition are used to measure the effects of model development choices on model bias, variance, and irreducible error. Measurements of bias and variance are then applied as diagnostic procedures for model pre-processing and development. Best performing model-normalization-data structure combinations were found to illustrate the downstream analysis effects of these model development choices. In additions, results found from simulations were verified and expanded to include additional data characteristics (imbalanced, sparse) by testing on benchmark datasets available from the UCI Machine Learning Library. Normalization results on benchmark data were consistent with those found using simulations, while also illustrating that more complex and/or non-linear models provide better performance on datasets with additional complexities. Finally, applying the findings from simulation experiments to previously tested applications led to equivalent or improved results with less model development overhead and processing time.

Keywords
Empirical Analysis, Bias-Variance Decomposition, Mean Squared Error, Downstream Analysis Effects, Empirical Risk

1. Introduction

Introduction Popularized in the work by David Holpert and William Macready, the No Free Lunch (NFL) Theorem states that no single machine learning algorithm is better than all the others on all problems [1]. Other researchers have
tried multiple models to find one that works best for a problem. In fact, studies by Carp [2] [3] illustrate effects on research findings in functional MRI (fMRI) studies due to variations in analytic strategy, with increased model flexibility leading to higher rates of false positive results. Wagenmakers, et al. [4] point out that many studies in psychology do not commit to an analysis method before seeing the data, with some researchers fine-tuning their analysis to the data, proposing that researchers “preregister their studies and indicate in advance the analyses they intend to conduct” in order to be considered as “confirmatory” research, rather than as “exploratory”. A study published in May 2020 expands on Carp’s findings, noting that fMRI analyses conducted on the same data by seventy different laboratories produced a wide range of results [5]. This particular study highlighted the fact that fMRI analysis requires several stages of pre-processing and analysis to determine which areas of the brain show activity. They found that the choice of pre-processing pipeline led to widely varied results. Among the seventy study teams, no two teams selected the same pipeline. Figure 1 illustrates the potential implications of varying pipeline choices in neuroimaging.

Perhaps the most illustrative lack of research consistency is the study by Silberzahn, et al. [6] which recruited 29 independent research teams with 61 analysts to address the question, “Are soccer referees more likely to give red cards to dark-skin-toned players than to light-skin-toned players?” The research teams represented 13 countries, a variety of disciplines, and a range of expertise and academic degrees. Using the same dataset and research question, the 29 teams utilized 29 unique analytical modeling approaches resulting in 21 unique combinations of covariates, 20 teams with significant positive results, and odds ratios ranging from 0.89 to 2.93, as in Table 1. To say the least, analytic choices, even if justifiable and statistically valid, have a downstream effect on model results.

The statistical model development framework can be generally divided into

![Figure 1](image-url)

**Figure 1.** Researchers process neuroimaging data using a wide variety of pipelines, which can produce varying results. Making different choices for each step leads to a different end point—the red dots represent how activation moves throughout the brain depending on which pipeline is used [5].
Table 1. From “Many Analysts, One Data Set: Making Transparent How Variations in Analytic Choices Affect Results” [6].

| Team | Distribution | Treatment of nonindependence | Number of covariates | Analytics Approach | OR     |
|------|--------------|------------------------------|----------------------|--------------------|--------|
| 1    | Linear       | Clustered standard errors    | 7                    | Ordinary least squares regression with robust standard errors, logistic regression | 1.18 [0.95, 1.41] |
| 6    | Linear       | Clustered standard errors    | 6                    | Linear probability model | 1.28 [0.77, 2.13] |
| 14   | Linear       | Clustered standard errors    | 6                    | Weighted least squares regression with clustered standard errors | 1.21 [0.97, 1.46] |
| 4    | Linear       | None                         | 3                    | Spearman correlation | 1.21 [1.20, 1.21] |
| 11   | Linear       | None                         | 4                    | Multiple linear regression | 1.25 [1.05, 1.49] |
| 10   | Linear       | Variance component           | 3                    | Multilevel regression and logistic regression | 1.03 [1.01, 1.05] |
| 2    | Logistic     | Clustered standard errors    | 6                    | Linear probability model, logistic regression | 1.34 [1.10, 1.63] |
| 30   | Logistic     | Clustered standard errors    | 3                    | Clustered robust binomial logistic regression | 1.28 [1.04, 1.57] |
| 31   | Logistic     | Clustered standard errors    | 6                    | Logistic regression | 1.12 [0.88, 1.43] |
| 32   | Logistic     | Clustered standard errors    | 1                    | Generalized linear models for binary data | 1.39 [1.10, 1.75] |
| 8    | Logistic     | None                         | 0                    | Negative binomial regression with a log link | 1.39 [1.17, 1.65] |
| 15   | Logistic     | None                         | 1                    | Hierarchical log-linear modeling | 1.02 [1.00, 1.03] |
| 3    | Logistic     | Variance component           | 2                    | Multilevel logistic regression using Bayesian inference | 1.31 [1.09, 1.57] |
| 5    | Logistic     | Variance component           | 0                    | Generalized linear mixed models | 1.38 [1.10, 1.75] |
| 9    | Logistic     | Variance component           | 2                    | Generalized linear mixed-effects models with logit link | 1.48 [1.20, 1.84] |
| 17   | Logistic     | Variance component           | 2                    | Bayesian logistic regression | 0.96 [0.77, 1.18] |
| 18   | Logistic     | Variance component           | 2                    | Hierarchical Bayes model | 1.10 [0.98, 1.27] |
| 23   | Logistic     | Variance component           | 2                    | Mixed-model logistic regression | 1.31 [1.10, 1.56] |
| 24   | Logistic     | Variance component           | 3                    | Multilevel logistic regression | 1.38 [1.11, 1.72] |
| 25   | Logistic     | Variance component           | 4                    | Multilevel logistic binomial regression | 1.42 [1.19, 1.71] |
| 28   | Logistic     | Variance component           | 2                    | Mixed-effects logistic regression | 1.38 [1.12, 1.71] |
| 21   | Miscellaneous| Clustered standard errors    | 3                    | Tobit regression | 2.88 [1.03, 11.47] |
| 7    | Miscellaneous| None                         | 0                    | Dirichlet-process Bayesian clustering | 1.71 [1.70, 1.72] |
| 12   | Poisson      | Fixed effect                 | 2                    | Zero-inflated Poisson regression | 0.89 [0.49, 1.60] |
| 27   | Poisson      | None                         | 1                    | Poisson regression | 2.93 [0.11, 78.66] |
| 13   | Poisson      | Variance component           | 1                    | Poisson multilevel modeling | 1.41 [1.13, 1.75] |
| 16   | Poisson      | Variance component           | 2                    | Hierarchical Poisson regression | 1.32 [1.06, 1.63] |
| 20   | Poisson      | Variance component           | 1                    | Cross-classified multilevel negative binomial model | 1.40 [1.15, 1.71] |
| 26   | Poisson      | Variance component           | 6                    | Hierarchical generalized linear modeling with Poisson sampling | 1.30 [1.08, 1.56] |

three phases: data discovery, variable preparation, and modeling. Within each of these phases there are steps in the model development that encompass a wide range of data management, data mining, and data analysis techniques, including data ingestion, sample selection, data cleaning and imputation, feature reduction, feature engineering, normalization, model development, and model validation. Analyzing the downstream effects of modeling approaches within each of these
steps will allow for statistically motivated modeling choices in the future. Quantifying the analysis effects of these strategies provides a diagnostic illustration of where researchers can expect to find improvements in their model results.

This study quantifies the downstream analysis effects of data pre-processing choices by utilizing a decomposition of the loss functions, measuring effects on model bias, variance, and irreducible error/random noise. In this way, measurements of bias and variance can be efficiently applied as diagnostic procedures for model pre-processing and development. Applying bias-variance decomposition to a variety of data distributions and model types can lead towards an improved understanding of quantitative variations within model development methods as well as comparing results consistently between methods. Understanding of statistical bias and variance can be used to diagnose problems with machine learning bias and develop methods for reducing bias and variance in algorithms. For example, this bias-variance trade-off does not always behave as expected under distributional assumptions. Even with the availability of more advanced models, such as neural networks, simple models still often perform well, or even better than more complex models, in experiments [7]. Generally, while more complex models result in decreased bias, they tend to increase variance and, therefore, do not generalize well to new data [8]. However, it has been found that ensemble models, although complex, often outperform single models and this seems contradictory to the trade-off between simplicity and accuracy. In this case, decomposition of bias-variance for ensembles led to the understanding that while increased complexity for a single model often increases variance, averaging multiple models will often (but not always) lead to decreased variance [9]. The goal, therefore, of understanding the effects on bias-variance decomposition is to quantify the downstream analysis effects of a selection of model development choices. Using these results as diagnostic procedures can lead to improved model development performance and consistent, reproducible results across data types and domains.

2. Methods
2.1. Quantifying Bias-Variance Trade-Off

We measure the effects of various normalization methods on the bias-variance decomposition of the risk function by directly simulating the definition of the decomposition under varying conditions. We use information found from these simulations to quantify the downstream analysis implications for the predictive models of interest. Since “an important goal in algorithm design is to minimize statistical bias and variance and thereby minimize error [10],” we use our findings to propose pre-processing and algorithm design choices that best minimize common design effects on bias and variance. For example, “any change that increases the representational power of an algorithm can reduce its statistical bias. Any change that expands the set of available alternatives for an algorithm or makes them depend on a smaller fraction of the training data can increase the
variance of the algorithm [10].” The result of such a study is to formulate a theory of bias and variance reduction and predict when either or both will succeed in practice.

2.2. Data Normalization

In this context we consider normalization to include data scaling techniques such as normalization and standardization. Data can be scaled so that features measured on different scales can be compared. Probability distributions of features can also be adjusted to be in alignment with each other. Another method is to shift and scale the features (standardization), which removes the units of measure. The primary goal of normalization is to scale each data point in a way that gives equal weight to the features to be used in developing a model. We consider four within-feature normalization methods including standardization, min-max, max absolute value (maxAbs), and quantile transformation, and one between-feature normalization, quantile normalization.

2.2.1. Z-Score, “Standardization”

Standardization is a method that shifts and scales the data to be centered around 0 with a standard deviation of 1:

$$x_i - \frac{\text{mean}(x)}{\text{stdev}(x)}$$  

Characteristics of this method include:
• Assumes data is normally distributed within each feature.
• Centers distribution around 0, with standard deviation 1.
• If data has outliers, scales most of the data to a small interval.
• Does not produce normalized features with the exact same scale.

Even if the data has outliers, Z-score normalization will scale most of the non-outlier data to be in a similar range between all features, assuming the data is normally distributed, as in Figure 2.

2.2.2. Min-Max Normalization

For each feature, the minimum value of that feature is transformed to a 0, the maximum value is transformed to a 1, and every other value lies between 0 and 1:

$$\frac{x_i - \text{min}(x)}{\text{max}(x) - \text{min}(x)}$$  

Advantages of this method include:
• Scales data between 0 and 1; guarantees all features have exact same scale.
• Preserves shape of original distribution.
• Preserves 0 entries in sparse data.
• Least disruptive to information in original data.

However, this method does not reduce the importance of outliers, so skewed results can still exist after normalizing if outliers exist, as in Figure 3.
Figure 2. The data is squished due to outlier but most of the data lies within similar range for both features ([11]).

Figure 3. Min-Max Normalization fixes the distribution on the Y-axis but is still problematic on the X-axis due to the outlier ([11]).

2.2.3. Max Absolute Value Normalization
This method scales feature by its maximum absolute value so that the maximum absolute value of each feature will be 1. This sets the distribution of each feature
between −1 and 1.

\[ \frac{x_i - \text{mean}(x)}{\max(\text{abs}(x_i - \text{mean}(x)))} \]  

(3)

Characteristics of this method include:

- Good for data with positive and negative values.
- Preserves 0 entries in sparse data.
- Similar sensitivity to outliers as in min-max normalization.

2.2.4. Quantile Transformation

Quantile transformation transforms each feature independently to follow a uniform or normal distribution. This is a non-linear transformation that uses the estimated value of the cumulative distribution function (CDF) to map a feature original values to a uniform or normal distribution:

1) Calculate empirical ranks, using percentile function.
2) Modify the ranking through interpolation.
3) Map to a Normal distribution by inverting the CDF, and clipping bounds at the extreme values so they don’t go to infinity.

Characteristics of this method include:

- Tends to spread out the most frequent values of a given feature.
- Smooths out unusual distributions.
- Less sensitive to outliers as other scaling methods.
- Distorts the linear correlations between variables measured at the same scale, but variables measured at different scales are more directly comparable.
- For a Normal transformation, the median of the feature becomes the mean, centered at 0.

2.2.5. Quantile Normalization

Quantile normalization is a method most notably used in genetics to normalize within samples, rather than within features as in the previously described methods. In genetic sequencing, data is often normalized based on the assumption of consistent within and between sample distributions, with observed variation around these distributions assumed to be the result of technical noise. Samples are normalized to the same distribution as each other or to a reference gene sample ([12]).

1) Given \( n \) arrays of length \( p \), form \( X \) of dimension \( p \times n \) where each array is a column;
2) Sort each column of \( X \) to give \( X_{\text{sort}} \);
3) Take the means across rows of \( X_{\text{sort}} \) and assign this mean to each element in the row to get \( X'_{\text{sort}} \);
4) Get \( X_{\text{normalized}} \) by rearranging each column of \( X'_{\text{sort}} \) to have the same ordering as original \( X \).

Characteristics of this method include:

- Makes 2 or more distributions identical in statistical properties.
- Does not preserve original data distributions.
2.3. Model Characteristics

In order to test the effects of various pre-processing methods on select data structures, the data structures and normalizations are considered under a selection of commonly used modeling techniques. The considered models include generalized linear models (GLM), decision tree, random forest, support vector machine (SVM), gradient boosting, and neural network. The selected models represent a range of simple to complex, parametric and non-parametric, global and local, stochastic methods. Each method has characteristics that may require normalization for optimal results or lead to unintended effects if incorrect normalization is used. For example, while GLM are fit using maximum likelihood estimation (MLE) which provides statistically optimal properties of the estimators, scaling data by normalization and standardization is still important because variables with a large difference in ranges can result in an ill-conditioned design matrix and difficulty reaching model convergence, resulting in slower processing times and unstable parameter estimates.

2.3.1. Generalized Linear Models

Historically, Generalized Linear Models (GLM) are an extension of simple linear regression models with continuous targets and continuous and/or categorical features. The form of such a model is expressed as

\[ y_i \sim N(x_i \beta, \sigma^2), \]

where \( x_i \) is the data in feature \( i \), and \( \beta \) are the coefficient parameters to be estimated as part of the linear function. In simple linear regression the assumption is that \( y \) is normally distributed, and the errors are normally distributed as \( e_i \sim N(0, \sigma^2) \) and independent, the data is fixed, and there is constant variance \( \sigma^2 \). The GLM extends this simple linear model concept by assuming the target variable, \( y_i \), follows a distribution within the exponential family (e.g. normal, binomial, poisson, etc.) with mean \( \mu_i \). The target then follows some linear or nonlinear function of \( x_i \beta \), the linear combination of data and estimated coefficient parameters [13]. A summary of common GLM is found in Table 2.

Generalized Linear Models are comprised of three main components: Random,

| Model               | Random   | Link  | Systematic |
|---------------------|----------|-------|------------|
| Linear Regression   | Normal   | Identity | Continuous |
| ANOVA               | Normal   | Identity | Categorical |
| ANCOVA              | Normal   | Identity | Mixed      |
| Logistic Regression | Binomial | Logit   | Mixed      |
| Loglinear           | Poisson  | Log    | Categorical |
| Poisson Regression  | Poisson  | Log    | Mixed      |
| Multinomial response| Multinomial | Generalized Logit | Mixed      |
Systematic, and Link Function. The random component refers to the distribution of the target variable \( Y \), e.g. normal distribution in linear regression, or binomial distribution in logistic regression. The systematic component specifies the explanatory features \( (X_1, X_2, \ldots, X_k) \) and their linear combination. The Link Function specifies the link between the random distribution of the target variable and the systematic features. Assumptions of GLMs include:

- Data are independently distributed.
- Errors are independent, but do not need to be normally distributed (i.e. Logistic Regression).
- Dependent variable does not need to be normally distributed (expect in linear regression) but are distributed within the exponential family.
- Assumes a linear relationship between the link function transformed target and the explanatory features.
- Uses Maximum Likelihood Estimation (MLE) to estimate the parameters, so it relies on large sample properties and regularity conditions (1st and 2nd derivatives must exist).

GLMs use Maximum Likelihood Estimation (MLE) to estimate the model parameters. In each of the distributions considered above (i.e. Linear, Logistic, Poisson, etc.), the distribution depends on one or more unknown parameters, \( \theta \). The value of these parameters, \( \theta \), is estimated using observed data \( x \). The function of \( \theta \) that results from plugging in observed data \( x \) is known as the Likelihood Function:

\[
L(\theta; x) = \prod_{i=1}^{n} f(x_i; \theta)
\]  

This function is the product of the values of the parameters, given each sample of data, and is denoted simply as \( L(\theta) \). The log-likelihood is often used for computational convenience. The goal in GLMs is to maximize the likelihood of a parameter estimate given the observed data. The value of \( \theta \) that maximizes this function is known as \( \hat{\theta} \), the maximum-likelihood estimate (MLE). The maximum of the function is found by taking the derivatives with respect to the parameter(s) \( \theta \).

In this work, three generalized linear models are considered for three distinct target data types: linear regression, logistic regression, and Poisson regression.

### 2.3.2. Linear Regression

Linear regression is used for data with a continuous target which is a linear combination of the explanatory features, as in

\[
y_i = \beta_0 + \beta x_i + \epsilon_i
\]  

where index \( i \) represents each data point. This models the mean expected value of \( Y \). The random component of linear regression, \( Y \), has a normal distribution and normally distributed errors, \( \epsilon_i \sim N(0, \sigma^2) \). The systematic component, the explanatory features \( X \), can be continuous, categorical, or a combination of both, and is linear in the parameters \( \beta_0 + \beta \). In multiple linear regression with mul-
Multiple explanatory features, there is still a linear combination of the features in terms of their coefficient parameters \( \beta \)'s but the features themselves can have transformations, *i.e.* \( X^2 \) or \( \log(X) \). The link function is the identity link, \( \eta = E(Y) \) since linear regression is modeling the mean response directly.

### 2.3.3. Logistic Regression

When there is a binary target (*i.e.* 0 and 1) binary logistic regression models the log odds of probability of "success" (target = 1). The random component, \( Y \), has a binomial distribution, Binomial \((n, \pi)\), where \( \pi \) is the probability of success. The systematic component, \( X \), can be continuous, categorical, or a combination of both, and is also linear in the parameters as in linear regression. However, in this case, the link function is the Logit link, \( \eta = \logit(\pi) = \log\left(\frac{\pi}{1-\pi}\right) \). Specifically, the logit link models the log odds of the mean response, \( \pi \).

### 2.3.4. Poisson Regression

When the target of interest is an expected count (*i.e.* counts of disease, number of homes sold in a day, etc.), we extend the generalized linear model to use a log-linear or Poisson regression model. This models the expected count as a function of the explanatory predictors, \( X = (X_1, X_2, \ldots, X_k) \), where the predictors can be continuous, categorical, or a combination of both. When all the predictors are categorical this is known as a log-linear model. The random component of the Poisson model is the response \( Y \) with Poisson distribution, \( y_i \sim \text{Poisson}(\mu) \) for \( i = 1, \ldots, N \) where expected count of \( y_i \) is \( E(Y) = \mu \). The systematic component is, as in the other GLM models, the linear combination of explanatory features \( X \). Finally, the link function for the Poisson regression model is the natural log link, \( \log(\mu) = \beta_0 + \beta_i x_i \).

**Advantages of GLM**

- Do not need to transform target variable to have normal distribution.
- Models fit using MLE which provides statistically optimal properties of the estimators.
- Model can be easily explained and parameters can be interpreted in the context of the prediction problem.
- Easily implemented in most software.

**Disadvantages of GLM**

- Still has to be a linear function of the parameters; the link function serves only to connect the nonlinear target distribution to a linear function.
- Target responses must be independent.

### 2.3.5. Decision Tree

A decision tree is a non-parametric classification technique that learns decision rules from features, using locally optimized, recursive partitioning. The algorithm assigns each sample in a dataset into a predicted class based on each samples’ feature attributes. The algorithm uses information gain (7) to find the best features for classifying the data, where \( p \) and \( n \) are the proportion of 0 and 1
values of a binary outcomes for the $i$-th target class. Then, for each value defined for the decision values of the best feature (the feature and splitting value that best splits the predicted 0 and 1 outcomes), the algorithm repeats the process with additional, next-best predictive features. This process continues until the leaves of the tree are pure (samples at each node belong to the same class) or a pre-defined stopping criteria is reached [14]. In this way, decision tree is also a feature importance algorithm, where the data will be split on the most important, predictive features first.

$$G(A) = I(p, n) - \sum_{i=1}^{p+n} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

where

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

**Advantages**

- Since the decision tree algorithm is based on ordering and splitting the values within each feature, rather than a scale-dependent maximum likelihood optimization, scaling and normalizing features is not required.
- Robust to missing data.
- This model provides visual splits of the data and ordered feature importance that is easy to understand and interpret.
- Implicit variable screening and selection, the top nodes of the tree are the most important variables in the dataset.
- Non-parametric model does not assume linearity or any other distribution of the data. Model is built only based on observed data.

**Disadvantages**

- Since this is a locally optimized, greedy algorithm, it is not guaranteed that a global optimum will be reached.
- Decision tree is very sensitive to changes in data. Small changes in data (i.e. adding samples) can lead to large structural changes in the tree, i.e. high variance.
- This is a more complex model and often requires more training time.
- Without regularization (early stopping, pruning, max nodes, etc.), there is high risk of overfitting.

2.3.6. Random Forest

Random forest is a method that uses ensemble learning to address some of the disadvantages of the decision tree model. Ensemble learning combines results from multiple models to make more accurate predictions than any one single model, by reducing variance. Random forest uses an ensemble learning technique known as bootstrap aggregation, aka bagging. Bagging uses random sampling with replacement to build individual models on subsets of the available data and then aggregate the results into one prediction. The repeated sampling leads to an algorithm that is known to reduce variance, as in one of the main
disadvantages of the decision tree model. Random forest combines many decision trees into one model by running the individual decision tree models in parallel and then outputting the prediction that is the mode of target classes for a classification problem or the mean prediction for a regression problem [15]. The structure of a random forest model is shown in Figure 4.

**Advantages**
- Much like decision tree, gives estimates of most important features.
- Known for high accuracy, low bias.
- Decreased variance in comparison to decision tree.
- Can handle large datasets with high dimensionality.
- Since it identifies most important features, can be used as a feature reduction method.
- Robust to missing data.
- Use of bootstrap sampling allows for successful application when data is limited.

**Disadvantages**
- When classifying categorical data, biased in favor of features with more levels.
- Will overfit data if regularization not used, such as limiting number of features that can be split at each node.
- More difficult to interpret than single decision tree model.

### 2.3.7. Support Vector Machines (SVM)

SVM with Gaussian kernel is a parametric model that represents instances of data as points in space and then builds a model to assign new instances to one category or another. Each data point is represented as an n-dimensional vector, then SVM constructs an n-1-dimensional separating hyperplane to discriminate 2 classes, with maximized distance between the hyperplane and data points on each side. SVM aims to find the best hyperplane for separation of both classes [16]. Data are represented as:

$$ (\bar{x}_i, y_i), \cdots, (\bar{x}_n, y_n) $$

where $y_i$ is either 1 or $-1$, indicating to which class $x_i$ belongs. Each $x_i$ is p-dimensional vector representing all of the characteristic values (features) of $x_i$. The hyperplane that best separates the group of $x_i$ vectors where $y_i = 1$ from the group of vectors where $y_i = -1$ is:

$$ \bar{\omega} \cdot \bar{x} - b = 0 $$

where $\bar{\omega}$ is the normal vector to the hyperplane and $b$ is the offset of the hyperplane from the origin. If the data points are linearly separable, the hard margin can be represented as

$$ \bar{\omega} \cdot \bar{x} - b = 1 $$

and

$$ \bar{\omega} \cdot \bar{x} - b = -1 $$

**Figure 5** shows a maximum margin separation for linearly separable data. The samples that fall on the margin are known as the support vectors.
The SVM algorithm assumes that data is in a standard range (usually between 0 to 1, or −1 to 1), so it is recommended to scale features before using the algorithm. In fact, when using the Gaussian kernel, if data is normalized between 0 and 1, then the dot product between the feature vectors and the separating hyperplane is the cosine similarity [18].

**Advantages**
• If there is clear separation of the data classes, SVM works very well.
• Effective in high-dimensional data, especially when the number of features is similar or greater than the number of samples.
• Since the samples that make up the support vectors are the only training data used to define the model, SVM is memory efficient.

Disadvantages
• Since this model has to calculate the distance between every training point to create a separating hyperplane, it is computationally expensive as the size of the data set increases.
• Noisy data with overlapping target classes are difficult to separate; Kernel functions can be added to transform the data into higher level feature space for improved separation but this adds model complexity.
• Does not directly provide parameter coefficients so it is difficult to interpret.

2.3.8. Gradient Boosting
Gradient boosting is another form of ensemble learning, this time utilizing a technique known as boosting. In a boosting algorithm, predictions are not made in parallel as in the bagging method of random forest. In this case, subsequent prediction models learn from the mistakes of previous models. Observations have an unequal probability of appearing in the subsequent models, with high error observations appearing in the most models. This is contrary to the random forest model where observations are selected for each model via bootstrapping (random selection with replacement) and have equal probability of appearing in each model. Visual comparison of single, bagging, and boosting models is show in Figure 6.

In gradient boosting, an ensemble of weak models, often decision trees, are used to improve the model based off of hard to predict samples. The algorithm leverages patterns in model residuals, such as those from using MSE loss, to build subsequent models from the weak predictions. For example, in a simple linear regression there is the assumption that the sum of the residuals is 0, i.e. spread randomly with no pattern around zero. However, assuming there is some pattern in the residuals for a base model, such as a decision tree, gradient boosting builds sequential models off of these residual patterns until there is no longer a pattern, i.e. average residual is zero or constant. The sequential model predictions are then weighted into a combined prediction. The intuitive idea behind gradient boosting is to combine several weak models, with each additional weak model improving the MSE of the overall model. Advantages of bagging and boosting ensemble techniques are illustrated in Figure 7.

Advantages
• Focus on difficult to classify cases makes it robust to imbalanced datasets.
• MSE is commonly used loss function, but gradient boosting can be optimized on many objective functions so it can be extended to many different problem spaces.

Disadvantages
Figure 6. Bagging (independent models) and boosting (sequential models) [19].

Figure 7. Ensembling [19].

- Requires more hyperparameter tuning, and training time to avoid overfitting compared with random forest.
- Sensitive to overfitting if data is noisy, i.e. many hard to classify cases to use in the sequential models.
- Longer training requirements due to sequential nature of algorithm (as opposed to parallel model development in random forest).

2.3.9. Neural Network

In this study, the effects of normalization on various data types are also tested using a multi-layer perception (MLP), also known as the simple form of a neural network. Neural networks are models that learn non-linear function approximations by feeding a set of input features into an output. Although the input and output layers are similar to the linear approximations of generalized linear models, neural networks differ in that there is one or more non-linear hidden layers, as in Figure 8 with one hidden layer.

The first layer, the input layer, contains a set of neurons $x_1, x_2, \ldots, x_m$ representing the $m$ input features. The inputs are fed into the hidden layer first with a weighted linear combination, similar to the linear combination of features
and $\beta$s in a GLM. The combined inputs are then transformed by a non-linear activation, such as a $\tan$ function. From the last hidden layer, the input layer then applies an activation function to transform the values into outputs, such as the $sigmoid$ function for a binary classification problem. The weights within each layer of the neural network are learned through a process of backpropagation and gradient descent. This process uses derivatives with respect to each parameter to find the optimal value of the selected loss function. Even though neural network uses non-linear transformations in the hidden layers, the network still uses linear combinations of the features and weights to learn the optimal parameters, as in the GLM, linear-based methods.

**Advantages**
- Can learn complex, non-linear models.
- Works well with “big data”; feeding neural networks more data leads to improved training and results.
- Ability to detect all possible interactions between predictor variable.

**Disadvantages**
- MLP with hidden layers have a non-convex loss function where there exists more than one local minimum; different random weight initializations can lead to different validation accuracy.
- Sensitive to feature scaling, due to above disadvantage.
- Requires a lot of tuning (number of hidden neurons, layers, iterations), and regularization to prevent overfitting.
- Requires a lot of data for best training and results.
- Difficult, computationally expensive to train.
- “Black box” algorithm is difficult or not possible to interpret.

**Figure 8.** One hidden layer neural network.
2.4. Model Summary

A global model is one in which there is a single predictive formula for the entire data space. It is expected that a linear transformation of data in a linear-based global model (linear regression, logistic regression, Poisson regression, linear SVM) will result in the model parameters (i.e. weights in a neural network, coefficients in regression) adjusting to reach the optimal value of the risk function, such as using the MLE in the GLM class of models. As a result, we expect that choice of normalization method should not affect the risk function value as long as the feature space is a convex function, but it can affect the values and stability of the feature coefficients. In this case, even though normalization may not affect the estimated total average error, it may have effects on the estimated average bias and variance due to model instability.

For non-linear, locally recursive models such as decision tree, random forest, and gradient boosting regression, it is also expected that within-feature global normalization will have little effect on risk function value. These tree-based models optimize by finding the best split-point within each individual feature by the percentage of labels correctly classified using that feature. Since these models are local, recursive models, as long as the ordering within the features is preserved, normalization of the data should not affect the loss function value. However, although we’re using a decision tree-based learning model for the gradient boosting regression, this type of sequential boosting model relies on minimizing the MSE for the global model through subsequent predictions on the individual model residuals. Because of this, it is suspected that the gradient boosting model will exhibit patterns in bias-variance decomposition similar to the linear models. However, since we are using the default hyper-parameters in the gradient boosting model for consistent simulation conditions, it is possible that the bias-variance decomposition results will suffer from over-fitting and have a longer training time.

2.5. Simulation Methods

In order to approximate the bias-variance decomposition we need to approximate the expected value $\mathbb{E}_x \left[ \hat{f}_t (X) \right]$ by simulating many variants of the training data sets. We can do this via bootstrap sampling. We take a synthetic input dataset $D$ and create variants of $D$ from $D_1, \cdots, D_T$ of size $n$ (Algorithm 1).

Now for each $(x,y)$ example we have many predictions $\hat{f}_1(x), \hat{f}_2(x), \cdots, \hat{f}_B(x)$ and can estimate:

- variance: variance of $\hat{f}_1(x), \hat{f}_2(x), \cdots, \hat{f}_B(x)$
- bias: average $\left( \hat{f}_1(x), \hat{f}_2(x), \cdots, \hat{f}_B(x) \right) - y$

$B = 1000$ bootstrap replicate datasets with 70% training samples and 30% out-of-bag testing samples were selected from simulated bivariate normal data with $n = 1000$ samples. The simulated features have different means and standard deviations, and an identity covariance matrix. The true target value, $Y$, was created as a simple linear function of the simulated features plus a random error.
Algorithm 1. Bootstrap Sampling.

```plaintext
for \( t = 1, \ldots, T \) do
    \( D_t = \emptyset \)
    for \( i = 1, \ldots, n \) do
        Pick \((x, y)\) uniformly at random from \( D \) (i.e., with replacement)
        and add it to \( D_t \)
    end for
end for
Create \( B \) bootstrap variants of \( D \)
for each bootstrap dataset \((b)\) do
    \( T_b \) is the dataset and \( U_b \) are the “out of bag” examples
    Train a hypothesis \( \hat{f}_b \) on \( T_b \)
    Test \( \hat{f}_b \) on each \( x \) in \( U_b \)
end for
```

term. In addition, datasets with binary (logistic) target and continuous target were created for each simulated data distribution. The models described previously, representing varying complexity, were applied using the training data and the bias-variance decomposition of the MSE risk computed on the test set, with total average risk, bias, and variance calculated over all 1000 bootstrap replicates. This process was repeated on the simulated data using the previously described normalization techniques. Initial dataset simulations were completed in R and risk function decompositions with bootstrapping completed in Python. This process was then repeated on additional simulated datasets including rank-based data, categorical data, mixed data, and Poisson data. Default hyperparameters were used for all tested models and data structures, with no additional hyperparameter tuning to allow for consistent comparison between methods. MSE risk decomposition was then performed on several benchmark datasets to assess results on various data characteristics including sparse data, wide data (more features than samples), and imbalanced data. These benchmark datasets were from the UCI Machine Learning Library [20] and are listed in Table 3. The results of bias-variance decomposition are then used to inform model development on several existing study applications. Selected applications include the historical data from the NCAA Men’s Basketball Tournament (historical data used due to cancellation of NCAA tournament in 2020; comparing to model results from last years competition), and a credit risk model [21].

3. Simulation Results

Results are divided into three sections describing results from 1) bias-variance decomposition simulation, 2) bias-variance decomposition on benchmark datasets, and 3) application of findings from Sections 1 and 2 to existing NCAA data and credit risk data. The first section on simulation results is divided by target data type (binary, continuous, Poisson). Results across data structures and models is relatively consistent so summaries were provided to avoid repetition. Complete results of bias-variance decomposition under various data structures,
Table 3. Benchmark datasets and characteristics.

| Dataset                      | Target Type | Attribute Type | Dataset Characteristics                                                                 | # Features | # Instances |
|------------------------------|-------------|----------------|-----------------------------------------------------------------------------------------|------------|-------------|
| Wine Quality                 | Binary      | Numeric        | Imbalanced                                                                              | 10         | 4898        |
| Breast Cancer Wisconsin      | Binary      | Numeric        | features have very dissimilar ranges, with half of the features near unary at 0          | 30         | 569         |
| Congressional Voting Records | Binary      | Categorical    | Missing data                                                                            | 16         | 435         |
| Abalone                      | Binary      | Mixed          | Imbalanced                                                                              | 8          | 4177        |
| Arrhythmia                   | Binary      | Mixed          | Imbalanced; small dataset; # features more than 1/2 # of instances                      | 279        | 452         |
| Forest Fires                 | Continuous  | Numeric        | No missings                                                                             | 13         | 517         |
| Solar Flare                  | Continuous  | Categorical    | # of common solar flares within 24 h; distribution of target is highly skewed towards 1 | 10         | 1066        |
| Auto MPG                     | Continuous  | Mixed          | No missings                                                                             | 8          | 398         |

normalization strategies, and models are shown in Appendices A and B. Performance measures for simulated model results are found in Table 4.

3.1. Binary Target

Over all models applied to bivariate normal data with binary target, SVM using within-feature normalization methods (risk = 0.290), and logistic regression with raw data or quantile normalization (risk = 0.290) have best, similar risk function results (Table A1). While SVM has the consistently best results and is normalization-agnostic (Figure B1(d)), logistic regression (Figure B1(a)) with raw data or quantile normalization has similar performance with a faster run time (approximately 5 seconds vs 18 seconds as in Table 4). If an analyst would like to use decision tree, random forest, or neural network models instead, it is recommended to use raw data or quantile normalization for best risk function results.

For models applied to rank-based data with binary target, logistic regression using raw data or quantile normalization (risk = 0.444), and SVM with raw data or quantile normalization (risk = 0.437) have best, similar risk function results (Table A4). While SVM has the consistently best results (14d), logistic regression with raw data or quantile normalization (Figure B4(a)) has similar performance with a faster run time (approximately 6 seconds vs 39 seconds as in Table 4). If an analyst would like to use decision tree (best risk = 0.489), random forest (best risk = 0.445), gradient boosting (best risk = 0.465), or neural network (best risk = 0.475) models instead, it is recommended to use raw data or quantile normalization for best risk function results.

Over all normalization methods and models applied to categorical data with binary target, risk function estimates ranged from 0.473 to 0.502, indicating that this data is somewhat model- and normalization-agnostic (Table A7). SVM
Table 4. Average model performance (in seconds of processing time) for bias-variance decomposition of simulated data structures.

| Features     | Bivariate Normal | Ranked | Categorical | Mixed |
|--------------|------------------|--------|-------------|-------|
| **Target**   |                  |        |             |       |
| **Binary**   |                  |        |             |       |
| Logistic Regression | 5    | 6      | 7           | 14.2  |
| Decision Tree  | 2    | 3      | 15          | 3.3   |
| Random Forest | 139  | 209    | 183         | 208.6 |
| SVM           | 18   | 39     | 21          | 37.3  |
| Gradient Boosting | 14   | 25     | 14          | 22.7  |
| Neural Network | 223  | 356    | 262         | 287   |
| **Continuous**|                  |        |             |       |
| Linear Regression | 1.2  | 0.8    | 8.8         | 1.6   |
| Decision Tree  | 3    | 2.3    | 1           | 2.8   |
| Random Forest | 206  | 167    | 156.9       | 170.9 |
| SVM           | 30   | 42     | 10          | 30.4  |
| Gradient Boosting | 17   | 13     | 8.7         | 10.2  |
| Neural Network | 26   | 14     | 6.8         | 66.4  |
| **Poisson**   |                  |        |             |       |
| Poisson Regression | 0.8  | 6      | 2.4         | 8     |
| Decision Tree  | 2.3  | 2.7    | 1           | 8     |
| Random Forest | 167  | 173.7  | 158.4       | 3.4   |
| SVM           | 42   | 36.6   | 22.2        | 176.5 |
| Gradient Boosting | 13   | 12.9   | 7.5         | 9.5   |
| Neural Network | 14   | 24     | 6.6         | 43.1  |

using z-standardized data, and logistic regression with all methods except z-standardization resulted in best risk function value of 0.473 (Figure B7(d)). However, while SVM and logistic regression have similar results, logistic regression (Figure B7(a)) is more than 3 times faster (7 seconds vs. 21 seconds average processing time as in Table 4). Since the simulated dataset consists of all categorical data, the features are first converted to [0, 1] coded dummy features, effectively “normalizing” the data between 0 and 1, so additional normalization methods are not expected to have an effect on the downstream analysis.

For mixed data types with a binary target, although there are slight deviations between normalization and model performance, risk function values do not vary much between all methods, with a range between 0.479 and 0.507 (Table A10). A decision tree model using raw data leads to the best results (Figure B10(b)), and gradient boosting using raw or quantile normalized data leads to the worst results (Figure B10(e)). However, considering the consistency of performance across normalization methods and models, it is recommended to make selections
based on additional criteria, such as processing resources, model interpretation, or another performance measure such as specificity and sensitivity.

### 3.2. Continuous Target

Over all models applied to bivariate normal data with continuous target, neural network using raw data (risk = 0.342), and linear regression with raw data (risk = 0.338) have best, similar risk function results (Table A2). Quantile normalization method for both models has similar results with MSE risk of 0.344 for linear regression (Figure B2(a)) and 0.356 for neural network (Figure B2(f)). However, while neural network and linear regression have similar results, linear regression is approximately 20 times faster (1.2 seconds vs. 26 seconds average processing time as in Table 4). SVM has the most consistent results; even though the within-feature normalization methods all perform worse than raw or quantile normalized data, SVM within-feature normalization results perform better than the same normalization in all other tested models. If normalization and scaling of data is required, as with features measured on highly divergent scales, it is recommended for an analyst to test the SVM model, keeping in mind increased processing requirements.

When applied to rank-based data with continuous target, neural network using raw and quantile normalized data (risk = 0.175), and linear regression with raw and quantile normalized data (risk = 0.174) have best, similar risk function results (Table A5). However, while neural network and linear regression have similar results, linear regression is more than 17 times faster (0.8 seconds vs. 14 seconds average processing time as in Table 4). SVM has the most consistent results (Figure B5(d)); even though the within-feature normalization methods all perform worse than raw or quantile normalized data, SVM within-feature normalization results perform better than the same normalization in all other tested models. If normalization and scaling of data is required, as with features measured on highly divergent scales, it is recommended for an analyst to test the SVM model, keeping in mind increased processing requirements. Note, however, that outside of the best-performing linear regression and neural network models, all other methods and models perform significantly worse due to exploding estimates of average bias.

While normalization generally does not improve or worsen results as compared with raw data, it is not recommended to use z-standardization for categorical data with a continuous target, as the simulation results indicate increased risk function values (Table A8). In particular, if using linear regression (Figure B8(a)), z-standardization and quantile transformation should be avoided as these methods used with this model lead to significant explosion in the risk function value. Outside of z-standardization for any tested model, and quantile transformation for linear regression, simulation results indicate that this type of data is both normalization- and model-agnostic. In this case, normalization and model selection can be based off of additional criteria, such as processing requirements or model transparency.
Over all models applied to mixed data with continuous target, neural network using raw data (risk = 0.366) and quantile normalized data (risk = 0.425), and linear regression using raw data (risk = 0.367) and quantile normalized data (risk = 0.363) have best, similar risk function results (Table A11). However, while neural network (Figure B11(f)) and linear regression (Figure B11(a)) have similar results, linear regression is approximately 41 times faster (1.6 seconds vs. 66 seconds average processing time as in Table 4). SVM has the most consistent results; even though the within-feature normalization methods all perform worse than raw or quantile normalized data, SVM within-feature normalization results perform better than the same normalization in all other tested models. If normalization and scaling of data is required, as with features measured on highly divergent scales, it is recommended for an analyst to test the SVM model, keeping in mind increased processing requirements and potential for increased bias.

### 3.3. Poisson Target

Over all models applied to bivariate normal data with Poisson target, random forest using within-feature normalization methods (risk = 1.125), and gradient boosting using within-feature normalization methods (risk = 1.167) have best, similar risk function results (Table A3). Poisson regression using raw or quantile normalized data also has strong results with risk function value of 1.5. Although Poisson regression (Figure B3(a)) results are not as strong as those found using random forest (Figure B3(c)) and gradient boosting (Figure B3(e)), Poisson regression has processing time more than 200 times faster than random forest (0.8 seconds vs. 167 seconds average processing time) and 13 times faster than gradient boosting (0.8 vs. 167 seconds average processing time) as seen in Table 4.

For rank-based data with Poisson target, SVM using raw and quantile normalized data (risk = 1.281), and Poisson regression with raw and quantile normalized data (risk = 1.280) have best, similar risk function results (Table A6). However, while SVM and Poisson regression have similar results, Poisson regression is more than 6 times faster (6 seconds vs. 36.6 seconds average processing time), as seen in Table 4. Although Poisson regression has the best results for this data, use of the within-feature normalization methods result in unstable, exploding risk function values and should be avoided (Figure B6(a)). Within-feature normalizations should also be avoided if using a neural network on this data for the same reason. If normalization and scaling of data is required, random forest model has the best results for the within-feature methods, although it has the longest processing time.

Although there are slight deviations between normalization and model performance, risk function values do not vary much between all methods when considering categorical data with Poisson target, with a range between 1.499 and 1.783 (Table A9). A decision tree model using min-max, maxAbs, quantile
transformation, or quantile normalization lead to the best results (Figure B9(b)), and SVM using z-standardization leads to the worst result (Figure B9(d)). However, considering the consistency of performance across normalization methods and models, it is recommended to make selections based on additional criteria, such as client requested models.

Over all models applied to mixed data with Poisson target, gradient boosting (Figure B12(e)) using raw and quantile normalized data (risk = 1.746), and random forest (Figure B12(c)) using raw data (risk = 1.820) have best results (Table A12). Generally, the best performing risk function values do not vary much between all tested models, with a range between 1.476 for the gradient boosting model and 2.087 for the decision tree model (Table 4). The similarity in best performing model results indicates that this data type is somewhat model-agnostic, although normalization methods should be selected carefully if required for analysis. For example, it is not recommended to use any of the tested within-feature normalizations if a neural network is used due to significant increases in bias and variance found in the simulation results.

4. Benchmark Data Results

Benchmark datasets were selected from the UCI Machine Learning Library to cover data types similar to those covered in the simulations. Complete tables of risk function decomposition results are found in Appendix Section A.2, and figures are found in Appendix Section B.2. Binary target datasets with numeric features (wine quality, breast cancer), and categorical features (congressional voting records), have bootstrapped bias-variance decomposition results consistent with those found in the simulated datasets with the same data structure characteristics (see Figures B13-B15). The traditional within-feature normalization methods (z-standardization, min-max, maxAbs, quantile transformation) result in risk function values that are the same or worse than using raw data or quantile normalization. For the wine quality data, using raw or quantile normalized data in logistic regression, linear SVM, or neural network results in best risk function performance, while quantile normalization with logistic regression was best for the breast cancer data. For the congressional voting records data, logistic regression and neural network with raw and quantile normalized data were also found to be the best method-model combinations, consistent with simulated data results. However, it is interesting to note that z-standardization in both of these models resulted in the worst risk function performance among all other method-model combinations applied to this dataset, due to both increased bias and variance.

For the binary target data with mixed data type features (arrhythmia, abalone), raw data and quantile normalization also lead to the best risk function performance. However, the arrhythmia dataset is a relatively more complex dataset in comparison to the others tested. It has missing data, many features in comparison to few instances (i.e. 279 features vs. 452 instances), and imbalanced target
data. As a result, logistic regression is not well suited for describing these complex relationships, and has worse risk function performance; decision tree and gradient boosting regression with raw data or quantile normalization have best results (Figure B17). In contrast, while the abalone dataset is also an imbalanced dataset, it has less complex data structures with only 8 features and over 4000 instances. In this case, logistic regression, linear support vector machine, and neural network are well-suited for the less complex data, and result in improved risk function values due to decreased variance in comparison to the more complex models (Figure B16).

For assessing results on continuous target data, the forest fires dataset (numeric features), solar flare dataset (categorical features), and auto MPG dataset (mixed type features) were considered. Once again, in all cases, the within-feature normalization performed the same or worse than using raw data, with the between-feature quantile normalization process being the only method that resulted in same or some improvement to risk function values. For both numeric and categorical only datasets (forest fires and solar flare, respectively), the linear-based logistic regression and neural network models with raw data or quantile normalization resulted in best performance (see Figure B18 and Figure B19), with all other normalization-model combinations resulting in both increased bias and variance. In the case of the more complex mixed feature type auto MPG dataset, gradient boosting regression also has improved performance, but logistic regression has similar performance and is a faster algorithm (Figure B20).

5. Applications

Findings of the empirical study and benchmark data were applied to existing studies, using best-performing normalization-model combinations in the model development process.

5.1. NCAA Tournament Data

For the 2019 NCAA Men’s Basketball Tournament Bracket prediction problem, data was used from over 100,000 NCAA regular season games, with the goal to take information about two teams as input, and output a probability of team 1 winning a game. Motivated by the popular Kaggle competition, models were developed to minimize log-loss between predicted win probabilities and actual game outcomes, as in:

$$\text{log Loss} = -\frac{1}{n} \sum_{i \in S} [y_i \log_{10}(y_i) + (1 - y_i) \log_{10}(1 - y_i)]$$

(12)

This loss function has high penalty for models that are both confident and wrong ([7]). Model development involved:

- Readily available game statistics, provided by Kaggle.
- Commonly used external ratings systems (Massey Ratings).
- No additional feature engineering.
• No domain knowledge.

The analysis value-added in comparison to previous research and public models found on Kaggle was to focus on the comparison of various normalization techniques for model development. In particular, the use of outside domain knowledge (public health, genetics) to apply a technique from one domain (genetic research) to an unrelated domain (sports data) proved advantageous but was not, initially, statistically motivated. However, through simulation of bias-variance decomposition and findings from application to benchmark data, it is expected that improved loss function performance for this type of data (ranked data, balanced target, non-missing data) can be achieved by using a linear-based model with raw or quantile normalized data. Rather than iterating through many normalization-model combinations (which took over 12 hours of computation time when building the model for the 2019 tournament), logistic and linear SVM models with raw and quantile normalized data were trained on NCAA regular season data from 2014-2017 and tested on the 2018 tournament. The same features were used as in the 2019 model, with only the model and normalization selection process updated based on the model development framework findings. From the simulation results, logistic regression and SVM for both raw and quantile normalization provided similar results, although SVM outperformed logistic somewhat due to decreased variance, although it has increased bias. In the updated NCAA application on 2018 data, logistic regression with raw data outperformed the other tested models, with a log-loss score of 0.569 (Figure 9). This log-loss score, in comparison to other Kaggle submissions in the 2018 tournament, would have ranked 23rd out of 933 teams (98 percentile) and required only the original data supplied by Kaggle and no additional feature engineering or model tuning. In addition, the entire model development process and testing took less than 30 minutes. Three out of the four models developed correctly predicted the Final Four including the tournament Champion, Villanova. This is compared with a 2019 bracket that, while it correctly predicted the tournament winner, only predicted two of the Final Four teams, and scored in the 90th percentile of Kaggle Log-loss scores.

5.2. Credit Risk Data

Previous research by Rudd and Priestley (rudd2017comparison) compare the use of logistic regression and decision trees for prediction of commercial credit risk. The dataset, provided by Equifax, included over 11 million records and over 300 features, and involved extensive data preprocessing including imputation, feature reduction, and transformation. The effects of normalization were not considered at the time. Based on findings from simulations and benchmark results, it was found that gradient boosting regression with raw and quantile normalization should also be considered for this type of data. Running the analysis again, this time including gradient boosting regression, found best results for gradient boosting with raw data (AUC = 0.96). A drawback, however, of this result in the context of credit risk analysis is that gradient boosting is much more
Figure 9. 2018 men's basketball march madness bracket developed using winning model, logistic regression with raw data.

difficult to explain than the logistic regression and decision tree models, and can be problematic in a heavily regulated industry where model interpretability is required (Figure 10).

6. Conclusions and Suggestions

In this study, simulation was used to investigate the effects of normalization on downstream analysis results. Normalization methods were investigated by utilizing a decomposition of the empirical risk functions, measuring effects on model bias, variance, and irreducible error. Estimates of bias and variance were then used as diagnostic procedures for data pre-processing and model development. We used our findings to propose model development and algorithm design choices that best minimize common design effects on bias and variance.
Mean square error (MSE) was considered as the evaluation metric, and the effects of a selection of normalization methods were measured on the empirical risk function. Normalization techniques were selected that represent both data invariant as well as data variant normalization strategies. For example, techniques such as z-score standardization (transforms data to have a mean of zero and a standard deviation of 1) and feature scaling (rescaling data to have values between 0 and 1) change the spread and position of data points (all by consistent factors) but do not change the distribution shape of the data, whereas techniques such as quantile normalization, commonly used in genetic differential expression analysis, affect the measures of spread, position, and shape. Through simulation of various data structures and bootstrap sampling of the bias-variance decomposition, best performing model-normalization-data structure combinations were found to illustrate the downstream analysis effects of these model development choices. For example, it was found that for rank-based data with binary target, quantile normalization performed better than the data invariant methods with similar or improved performance over raw data due to decreased variance in the loss function value. In addition, results found from simulations were verified and expanded to include additional data characteristics (imbalanced, sparse) by testing on benchmark datasets available from the UCI Machine Learning Library. Normalization results on benchmark data were consistent with those

Figure 10. AUC curves for selected model-normalization combinations tested based on model framework results.
found using simulations, while also illustrating that more complex and/or non-linear models provide better performance on datasets with additional complexities, such as wide data (large feature to instance ratio) as in the arrhythmia dataset. Finally, applying the findings from simulation experiments to previous applications led to equivalent or improved results with less model development overhead and processing time. Applying the model framework to the 2018 NCAA Men’s Basketball data resulted in a log-loss score that would have been ranked 23 out of 933 teams (98th percentile) and only required 30 minutes of model overhead, as opposed to a 2019 model that required over 12 hours of processing and resulted in a 90th percentile log-loss score.

6.1. Limitations

While this work provides a statistical illustration of the downstream effects of model development choices, it represents a baseline for further research in this area. For example, while the bias-variance decomposition simulations described in this study illustrate that model and normalization method selection do affect downstream results, they are only suggestive of theoretical properties of these specific methods that should be further explored. Also, a researcher’s primary modeling goal (i.e., predictive accuracy vs. explanatory model) will determine both appropriate model and pre-processing technique selection. In addition, the main goal of normalization is to put features on comparable scale for improved model fitting, performance, and interpretability. Considering normalization as a model selection procedure and selecting based on minimized risk function value can potentially lead to overfitting. Finally, this study considers a limited selection of models and model performance measures, while assuming all other proposed aspects of the model development framework are held constant. A more exhaustive study of performance assessments should be considered to better establish the downstream analysis effects of statistical procedures, including coverage probabilities, misclassification rates, sensitivity/specificity, etc. In this study, we selected MSE due to the ability to generalize the risk functions across multiple data types and model applications. In addition, these assessments need to include additional consideration on various combinations of model development strategies within the model development process, i.e., sample selection, feature engineering, model validation, etc.

6.2. Future Research

In this research, an empirical study was used to illustrate the downstream analysis effects of model development choices. Further theoretical work is recommended to connect the empirical findings theoretical properties. For example, since the generalized linear models were most often found to have best risk function results regardless of data structure or selected normalization, additional theoretical study can enhance the explanation of these results. If we assume that MLE is unbiased in GLM then estimates of bias should resolve towards zero. In
this case, bias-variance decomposition of the loss will be completely defined by variance. As an extension, the Cramer-Rao lower bound property of MLE (the lower bound of the variance of the estimator) suggests that no other model will achieve a better result of the bias-variance loss decomposition. This potential explanation requires theoretical understanding of at least two questions: 1) Are generalized linear models, in fact, unbiased? and 2) Does the Cramer-Rao lower bound theorem apply to variance of the prediction and not just the parameters? In addition, estimates of predictive risk are only one way to assess performance of a statistical procedure and downstream analysis effects. In order to provide further justification connecting the empirical evidence with the proposed framework, it is recommended to consider additional measures of performance such as coverage of predictive intervals, class probabilities cutoff points, gain and lift charts, etc. The diversity of analytic choices and resulting modeling pipelines leads to a wide range of potential future research required to truly quantify the complete effects of a unified model development framework.

**Conflicts of Interest**

The authors declare no conflicts of interest regarding the publication of this paper.

**References**

[1] Wolpert, D.H., Macready, W.G., et al. (1997) No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation, 1*, 67-82. [https://doi.org/10.1109/4235.585893](https://doi.org/10.1109/4235.585893)

[2] Carp, J. (2012) On the Plurality of (Methodological) Worlds: Estimating the Analytic Flexibility of fMRI Experiments. *Frontiers in Neuroscience, 6*, 149. [https://doi.org/10.3389/fnins.2012.00149](https://doi.org/10.3389/fnins.2012.00149)

[3] Carp, J. (2012) The Secret Lives of Experiments: Methods Reporting in the fMRI Literature. *Neuroimage, 63*, 289-300. [https://doi.org/10.1016/j.neuroimage.2012.07.004](https://doi.org/10.1016/j.neuroimage.2012.07.004)

[4] Wagenmakers, E.-J., et al. (2012) An Agenda for Purely Confirmatory Research. *Perspectives on Psychological Science, 7*, 632-638. [https://doi.org/10.1177/1745691612463078](https://doi.org/10.1177/1745691612463078)

[5] Botvinik-Nezer, R., et al. (2020) Variability in the Analysis of a Single Neuroimaging Dataset by Many Teams. *Nature*, 1-7.

[6] Silberzahn, R., et al. (2018) Many Analysts, One Data Set: Making Transparent How Variations in Analytic Choices Affect Results. *Advances in Methods and Practices in Psychological Science, 1*, 337-356.

[7] Yuan, L.H., et al. (2015) A Mixture-of-Modelers Approach to Forecasting NCAA Tournament Outcomes. *Journal of Quantitative Analysis in Sports, 11*, 13-27. [https://doi.org/10.1515/jqas-2014-0056](https://doi.org/10.1515/jqas-2014-0056)

[8] Singh, S. (2018) Understanding the Bias-Variance Tradeoff. [https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229](https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229)

[9] Domingos, P. (2000) A Unified Bias-Variance Decomposition. *Proceedings of 17th
International Conference on Machine Learning, 231-238.

[10] Dietterich, T.G. and Kong, E.B. (1995) Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms. Technical Report, Department of Computer Science, Oregon State University, Corvallis.

[11] Normalization. https://www.codecademy.com/articles/normalization

[12] Evans, C., Hardin, J. and Stoebel, D.M. (2017) Selecting Between-Sample RNA-Seq Normalization Methods from the Perspective of Their Assumptions. Briefings in Bioinformatics, 19, 776-792. https://doi.org/10.1093/bib/bbx008

[13] Agresti, A. (2003) Categorical Data Analysis. Vol. 482, John Wiley & Sons, Hoboken. https://doi.org/10.1002/0471249688

[14] Owen, S., et al. (2015) Advanced Analytics with Apache Spark.

[15] Chakure, A. (2020) Random Forest and Its Implementation. https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f

[16] Rudd, J.M. (2018) Application of Support Vector Machine Modeling and Graph Theory Metrics for Disease Classification. Model Assisted Statistics and Applications, 13, 341-349. https://doi.org/10.3233/MAS-180444

[17] Kumari, V.A. and Chitra, R. (2013) Classification of Diabetes Disease Using Support Vector Machine. International Journal of Engineering Research and Applications, 3, 1797-1801.

[18] Forman, G., Scholz, M. and Rajaram, S. (2009) Feature Shaping for Linear SVM Classifiers. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 299-308. https://doi.org/10.1145/1557019.1557057

[19] Grover, P. (2019) Gradient Boosting from Scratch. https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

[20] Dua, D. and Graff, C. (2017) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[21] Rudd, M.P.H., GStat, J.M. and Priestley, J.L. (2017) A Comparison of Decision Tree with Logistic Regression Model for Prediction of Worst Non-Financial Payment Status in Commercial Credit.
## Appendix A

### A.1. Simulations

#### A.1.1. Bivariate Normal Data with Binary Target

**Table A1.** Bias-variance decomposition results for bivariate normal data with binary target.

| Data               | Model    | Normalization | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|--------------------|----------|---------------|------|------------|---------|----------------|-------------------|-------------------|
| Bivariate Normal - Binary Target | Logistic | Total Risk    | 0.290 | 0.462      | 0.474   | 0.485          | 0.482             | 0.290             |
|                    |          | Bias          | 0.290 | 0.220      | 0.228   | 0.237          | 0.234             | 0.290             |
|                    |          | Variance      | 0.000 | 0.242      | 0.246   | 0.438          | 0.248             | 0.000             |
|                    |          | Noise         | 0.000 | 0.000      | 0.000   | 0.190          | 0.000             | 0.000             |
|                    |          | Variance-Bias Ratio | 0.000 | 1.098      | 1.080   | 1.850          | 1.061             | 0.000             |
|                    |          | Percent Change from Raw | –     | 159.319 | 163.488 | 167.408 | 166.279 | 100.007 |
| Decision Tree      |          | Total Risk    | 0.408 | 0.564      | 0.564   | 0.564          | 0.564             | 0.393             |
|                    |          | Bias          | 0.250 | 0.337      | 0.337   | 0.337          | 0.337             | 0.225             |
|                    |          | Variance      | 0.158 | 0.227      | 0.227   | 0.348          | 0.227             | 0.168             |
|                    |          | Noise         | 0.000 | 0.000      | 0.000   | 0.121          | 0.000             | 0.000             |
|                    |          | Variance-Bias Ratio | 0.631 | 0.673      | 0.673   | 1.033          | 0.673             | 0.748             |
|                    |          | Percent Change from Raw | –     | 138.303 | 138.303 | 138.303 | 138.303 | 96.434 |
| Random Forest      |          | Total Risk    | 0.295 | 0.396      | 0.396   | 0.396          | 0.396             | 0.295             |
|                    |          | Bias          | 0.286 | 0.207      | 0.207   | 0.207          | 0.207             | 0.286             |
|                    |          | Variance      | 0.009 | 0.188      | 0.188   | 0.252          | 0.188             | 0.009             |
|                    |          | Noise         | 0.000 | 0.000      | 0.000   | 0.064          | 0.000             | 0.000             |
|                    |          | Variance-Bias Ratio | 0.032 | 0.909      | 0.909   | 1.215          | 0.909             | 0.033             |
|                    |          | Percent Change from Raw | –     | 134.241 | 134.241 | 134.241 | 134.241 | 100.002 |
| SVM                |          | Total Risk    | 0.292 | 0.290      | 0.290   | 0.290          | 0.290             | 0.290             |
|                    |          | Bias          | 0.287 | 0.290      | 0.290   | 0.290          | 0.290             | 0.288             |
|                    |          | Variance      | 0.005 | 0.000      | 0.000   | 0.000          | 0.000             | 0.005             |
|                    |          | Noise         | 0.000 | 0.000      | 0.000   | 0.000          | 0.000             | 0.000             |
|                    |          | Variance-Bias Ratio | 0.016 | 0.000      | 0.000   | 0.000          | 0.000             | 0.016             |
|                    |          | Percent Change from Raw | –     | 99.440   | 99.440   | 99.440 | 99.440 | 100.218 |
| Gradient Boosting  |          | Total Risk    | 0.332 | 0.655      | 0.655   | 0.655          | 0.655             | 0.332             |
|                    |          | Bias          | 0.249 | 0.540      | 0.540   | 0.541          | 0.540             | 0.249             |
|                    |          | Variance      | 0.082 | 0.115      | 0.115   | 0.132          | 0.115             | 0.083             |
|                    |          | Noise         | 0.000 | 0.000      | 0.000   | 0.018          | 0.000             | 0.000             |
A.1.2. Bivariate Normal Data with Continuous Target

Table A2. Bias-Variance decomposition results for bivariate normal data with continuous target.

| Data                  | Type of Loss | MSE          | MSE          | MSE          | MSE          | MSE          | MSE          |
|-----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                       |              | Total Loss   | Bias         | Variance     | Noise        | Variance-Bias Ratio | Percent Change from Raw |
| Bivariate Normal - Continuous Target | Linear       |              | None         | Z-standard   | Min-Max      | MaxAbs (−1, 1)      |                          |
|                       |              | Total Loss   | Bias         | Variance     | Noise        | Variance-Bias Ratio | Percent Change from Raw |
|                       |              | 0.338        | 0.335        | 0.003        | 0.000        | 0.008          | –                        |
|                       |              | 200,151.804  | 200,151.449  | 0.355        | 0.000        | 0.000          | 59,281,297.280          |
|                       |              | 8,999,347.482| 8,999,335.213| 12.269       | 0.000        | 0.000          | 2,665,441,848.182       |
|                       |              | 2,146,481,764.484| 2,146,478,729.725| 12.269       | 3022.490     | 0.000          | 635,748,573,177.009     |
|                       |              | 2,247,609,346| 2,247,069,765| 539,581      | 0.000        | 0.000          | 665,700,710.123         |
|                       |              | 0.344        | 0.338        | 0.006        | 0.000        | 0.019          | 101.929                  |

| Decision Tree         |              | Total Loss   | Bias         | Variance     | Noise        | Variance-Bias Ratio | Percent Change from Raw |
|                       |              | 0.748        | 0.444        | 0.304        | 0.000        | 0.685          | –                        |
|                       |              | 114.588      | 111.856      | 2.731        | 0.000        | 0.024          | 15,311,228              |
|                       |              | 114.588      | 111.856      | 2.731        | 0.000        | 0.024          | 2,665,441,848.182       |
|                       |              | 114.588      | 111.856      | 2.731        | 0.000        | 0.024          | 635,748,573,177.009     |
|                       |              | 114.588      | 111.856      | 2.731        | 0.000        | 0.024          | 665,700,710.123         |
|                       |              | 0.589        | 0.338        | 0.251        | 0.000        | 0.742          | 78.694                   |

| Random Forest         |              | Total Loss   | Bias         | Variance     | Noise        | Variance-Bias Ratio | Percent Change from Raw |
|                       |              | 2.809        | 2.425        | 0.385        | 0.000        | 0.159          | –                        |
|                       |              | 22.970       | 22.501       | 0.469        | 0.000        | 0.021          | 15,311,228              |
|                       |              | 22.970       | 22.501       | 0.469        | 0.000        | 0.021          | 2,665,441,848.182       |
|                       |              | 22.970       | 22.501       | 0.469        | 0.000        | 0.021          | 635,748,573,177.009     |
|                       |              | 22.970       | 22.501       | 0.469        | 0.000        | 0.021          | 665,700,710.123         |
|                       |              | 2.812        | 2.427        | 0.385        | 0.000        | 0.158          | 78.694                   |
## A.1.3. Bivariate Normal Data with Poisson Target

### Table A3. Bias-variance decomposition results for bivariate normal data with poisson target.

| Data                      | Model             | Normalization | Type of Loss | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|---------------------------|-------------------|---------------|--------------|------|------------|---------|----------------|--------------------|--------------------|
| Bivariate Normal - Poisson Target | Poisson Regression |               | Total Loss   | MSE  | MSE        | MSE     | MSE            | MSE                | MSE                |
|                           |                   |               | Bias         | 1.502| 191.901    | 1.874   | 1.764          | 3.004              | 1.502              |
|                           |                   |               | Variance     | 1.280| 1.561      | 1.609   | 1.600          | 1.459              | 1.275              |
|                           |                   |               | Noise        | 0.222| 190.340    | 0.264   | 0.264          | 1.546              | 0.227              |
|                           |                   |               | Variance-Bias Ratio | 0.000| 0.000      | 0.000   | 0.101          | 0.000              | 0.000              |
| Percent Change from Raw   | SVM               |               |              | 0.173| 121.919    | 0.164   | 0.165          | 1.060              | 0.178              |
|                           |                   |               |              | 0.020| 0.000      | 0.000   | 0.000          | 0.000              | 0.000              |
|                           | SVM               |               |              | 0.470| 0.009      | 0.013   | 0.013          | 0.013              | 0.052              |
|                           | SVM               |               |              | 1.336| 1336.235   | 1362.522| 1355.909       | 1361.662           | 101.476            |
|                           | SVM               |               |              | 0.550| 151.982    | 151.982 | 152.078        | 151.982            | 0.552              |
|                           | SVM               |               |              | 0.289| 150.938    | 150.938 | 151.034        | 150.938            | 0.290              |
|                           | SVM               |               |              | 0.261| 1.043      | 1.043   | 1.043          | 1.043              | 0.262              |
|                           | SVM               |               |              | 0.000| 0.000      | 0.000   | 0.001          | 0.000              | 0.000              |
|                           | SVM               |               |              | 0.047| 0.009      | 0.013   | 0.013          | 0.013              | 0.052              |
|                           | SVM               |               |              | 1336.235| 1336.235  | 1362.522| 1355.909       | 1361.662           | 101.476            |
|                           | Gradient Boosting |               | Total Loss   | MSE  | MSE        | MSE     | MSE            | MSE                | MSE                |
|                           |                   |               | Bias         | 0.341| 200,151.851| 8,999,382.281| 2,145,420,994.195| 2,247,611.004       | 0.356              |
|                           |                   |               | Variance     | 0.335| 200,151.497| 8,999,370.017| 2,145,297,375.914| 2,247,071.518       | 0.315              |
|                           |                   |               | Noise        | 0.007| 0.355      | 12.264  | 12.264         | 539.487            | 0.041              |
|                           |                   |               | Variance-Bias Ratio | 0.000| 0.000      | 0.000   | 123606.017     | 0.000              | 0.132              |
| Percent Change from Raw   |                   |               |              | 0.020| 0.000      | 0.000   | 0.000          | 0.000              | 0.132              |
|                           |                   |               |              | 27,615.537| 27,615.537| 27,633.076| 27,615.537     | 100.363            | 104.238            |
|                           | Neural Network    |               | Total Loss   | MSE  | MSE        | MSE     | MSE            | MSE                | MSE                |
|                           |                   |               | Bias         | 0.341| 200,151.851| 8,999,382.281| 2,145,420,994.195| 2,247,611.004       | 0.356              |
|                           |                   |               | Variance     | 0.335| 200,151.497| 8,999,370.017| 2,145,297,375.914| 2,247,071.518       | 0.315              |
|                           |                   |               | Noise        | 0.007| 0.355      | 12.264  | 12.264         | 539.487            | 0.041              |
|                           |                   |               | Variance-Bias Ratio | 0.000| 0.000      | 0.000   | 123606.017     | 0.000              | 0.132              |
| Percent Change from Raw   |                   |               |              | 0.020| 0.000      | 0.000   | 0.000          | 0.000              | 0.132              |
|                           |                   |               |              | 58,619,129.637| 2,635,678,625.092| 628,336,487,970.484| 658,265,211.621     | 104.238            |
### Decision Tree

|             | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-------------|------------|------|----------|-------|---------------------|-------------------------|
| **Total Loss** | 1.869      | 1.801| 1.801    | 1.802 | 1.801               | 2.065                   |
| Bias        | 1.126      | 1.428| 1.428    | 1.429 | 1.428               | 1.173                   |
| Variance    | 0.743      | 0.373| 0.373    | 0.373 | 0.373               | 0.892                   |
| Noise       | 0.000      | 0.000| 0.000    | 0.000 | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.660   | 0.261| 0.261    | 0.261 | 0.261               | 0.760                   |
| Percent Change from Raw | –       | 96.319| 96.319   | 96.371| 96.319              | 110.450                 |

### Random Forest

|             | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-------------|------------|------|----------|-------|---------------------|-------------------------|
| **Total Loss** | 1.564      | 1.125| 1.125    | 1.125 | 1.125               | 1.564                   |
| Bias        | 1.369      | 0.980| 0.980    | 0.980 | 0.980               | 1.370                   |
| Variance    | 0.195      | 0.145| 0.145    | 0.145 | 0.145               | 0.194                   |
| Noise       | 0.000      | 0.000| 0.000    | 0.000 | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.142   | 0.148| 0.148    | 0.148 | 0.148               | 0.142                   |
| Percent Change from Raw | –       | 71.964| 71.964   | 71.964| 71.964              | 100.059                 |

### SVM

|             | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-------------|------------|------|----------|-------|---------------------|-------------------------|
| **Total Loss** | 1.676      | 1.324| 1.443    | 1.886 | 1.341               | 1.700                   |
| Bias        | 1.528      | 1.091| 1.192    | 1.805 | 1.104               | 1.556                   |
| Variance    | 0.148      | 0.233| 0.252    | 0.252 | 0.237               | 0.143                   |
| Noise       | 0.000      | 0.000| 0.000    | 0.171 | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.097   | 0.213| 0.211    | 0.139 | 0.214               | 0.092                   |
| Percent Change from Raw | –       | 78.974| 86.122   | 112.515| 80.023              | 101.403                 |

### Gradient Boosting

|             | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-------------|------------|------|----------|-------|---------------------|-------------------------|
| **Total Loss** | 1.561      | 1.167| 1.167    | 1.167 | 1.167               | 1.550                   |
| Bias        | 1.325      | 0.943| 0.943    | 0.943 | 0.943               | 1.319                   |
| Variance    | 0.235      | 0.224| 0.224    | 0.224 | 0.224               | 0.231                   |
| Noise       | 0.000      | 0.000| 0.000    | 0.000 | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.178   | 0.237| 0.237    | 0.237 | 0.237               | 0.175                   |
| Percent Change from Raw | –       | 74.795| 74.795   | 74.791| 74.795              | 99.310                  |

### Neural Network

|             | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-------------|------------|------|----------|-------|---------------------|-------------------------|
| **Total Loss** | 1.501      | 50.236| 2611.864 | 553,224.671| 740.072           | 1.496                   |
| Bias        | 1.253      | 25.275| 1286.970 | 257,374.377| 412.057           | 1.267                   |
| Variance    | 0.248      | 24.961| 1324.894 | 1324.894 | 328.015           | 0.229                   |
| Noise       | 0.000      | 0.000| 0.000    | 294,525.400| 0.000            | 0.000                   |
| Variance-Bias Ratio | 0.198   | 0.988| 1.029    | 0.005 | 0.796               | 0.181                   |
| Percent Change from Raw | –       | 3346.523| 173,990.881| 36,853,390.488| 49,300.337     | 99.642                  |
### A.1.4. Ranked Data with Binary Target

**Table A4.** Bias-variance decomposition results for ranked data with binary target.

| Data               | Model       | Normalization | None | Z-standard | Min-Max | MaxAbs (-1, 1) | Quantile Transform | Quantile Normalize |
|--------------------|-------------|---------------|------|------------|---------|----------------|--------------------|--------------------|
| Ranked - Binary Target | Logistic    | Total Loss    | 0.444 | 0.505      | 0.506   | 0.506          | 0.506              | 0.444              |
|                    |             | Bias          | 0.369 | 0.343      | 0.344   | 0.344          | 0.343              | 0.369              |
|                    |             | Variance      | 0.075 | 0.163      | 0.162   | 0.240          | 0.163              | 0.075              |
|                    |             | Noise         | 0.000 | 0.000      | 0.000   | 0.078          | 0.000              | 0.000              |
|                    |             | Variance-Bias Ratio | 0.202 | 0.474      | 0.473   | 0.698          | 0.476              | 0.202              |
|                    |             | Percent Change from Raw | –     | 113.941 | 114.128 | 114.127 | 114.080 | 100.000 |
| Decision Tree      |             | Total Loss    | 0.489 | 0.494      | 0.494   | 0.494          | 0.494              | 0.493              |
|                    |             | Bias          | 0.292 | 0.246      | 0.246   | 0.246          | 0.246              | 0.291              |
|                    |             | Variance      | 0.197 | 0.248      | 0.248   | 0.454          | 0.248              | 0.202              |
|                    |             | Noise         | 0.000 | 0.000      | 0.000   | 0.206          | 0.000              | 0.000              |
|                    |             | Variance-Bias Ratio | 0.672 | 1.009      | 1.006   | 1.843          | 1.006              | 0.692              |
|                    |             | Percent Change from Raw | –     | 100.905 | 101.041 | 101.041 | 101.041 | 100.767 |
| Random Forest      |             | Total Loss    | 0.445 | 0.488      | 0.487   | 0.487          | 0.487              | 0.445              |
|                    |             | Bias          | 0.354 | 0.248      | 0.247   | 0.247          | 0.247              | 0.354              |
|                    |             | Variance      | 0.091 | 0.240      | 0.240   | 0.399          | 0.240              | 0.091              |
|                    |             | Noise         | 0.000 | 0.000      | 0.000   | 0.159          | 0.000              | 0.000              |
|                    |             | Variance-Bias Ratio | 0.257 | 0.965      | 0.969   | 1.613          | 0.969              | 0.257              |
|                    |             | Percent Change from Raw | –     | 109.658 | 109.524 | 109.524 | 109.524 | 100.000 |
| SVM                |             | Total Loss    | 0.437 | 0.441      | 0.460   | 0.460          | 0.459              | 0.437              |
|                    |             | Bias          | 0.437 | 0.407      | 0.308   | 0.309          | 0.313              | 0.437              |
|                    |             | Variance      | 0.000 | 0.034      | 0.152   | 0.187          | 0.146              | 0.000              |
|                    |             | Noise         | 0.000 | 0.000      | 0.000   | 0.036          | 0.000              | 0.000              |
|                    |             | Variance-Bias Ratio | 0.000 | 0.083      | 0.493   | 0.606          | 0.468              | 0.000              |
|                    |             | Percent Change from Raw | –     | 101.023 | 105.432 | 105.403 | 105.171 | 100.000 |
| Gradient Boosting  |             | Total Loss    | 0.465 | 0.521      | 0.521   | 0.521          | 0.521              | 0.465              |
|                    |             | Bias          | 0.303 | 0.298      | 0.299   | 0.299          | 0.299              | 0.303              |
|                    |             | Variance      | 0.162 | 0.222      | 0.222   | 0.334          | 0.222              | 0.162              |
|                    |             | Noise         | 0.000 | 0.000      | 0.000   | 0.112          | 0.000              | 0.000              |
|                    |             | Variance-Bias Ratio | 0.535 | 0.745      | 0.745   | 1.119          | 0.745              | 0.535              |
|                    |             | Percent Change from Raw | –     | 112.059 | 112.123 | 112.123 | 112.123 | 100.000 |
### A.1.5. Ranked Data with Continuous Target

**Table A5.** Bias-variance decomposition results for ranked data with continuous target.

| Data                  | Model        | Normalization | None          | Z-standard     | Min-Max       | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|-----------------------|--------------|---------------|---------------|---------------|--------------|---------------|-------------------|-------------------|
|                       | Linear       | Type of Loss  | Total Loss    | MSE           | MSE          | MSE           | MSE               | MSE               |
|                       |              |               | Neural Network | 0.478         | 0.492        | 0.487         | 0.486             | 0.484             | 0.476             |
|                       |              |               | Bias          | 0.253         | 0.248        | 0.249         | 0.247             | 0.250             | 0.258             |
|                       |              |               | Variance      | 0.225         | 0.245        | 0.238         | 0.383             | 0.234             | 0.218             |
|                       |              |               | Noise         | 0.000         | 0.000        | 0.000         | 0.144             | 0.000             | 0.000             |
|                       |              |               | Variance-Bias Ratio | 0.889        | 0.988        | 0.955         | 1.547             | 0.937             | 0.844             |
|                       |              |               | Percent Change from Raw | –            | 103.006      | 101.848       | 101.619            | 101.338            | 99.489            |
|                       | Decision Tree |               | Total Loss    | 6464.023      | 3,058,180.695 | 3,058,180.695 | 3,058,180.695     | 3,058,180.695      | 6421.793           |
|                       |              |               | Bias          | 959.928       | 3,055,675.410 | 3,055,675.410 | 3,055,675.410     | 3,055,675.410      | 1113.417           |
|                       |              |               | Variance      | 5504.094      | 2505.285     | 2505.285     | 2505.285          | 2505.285          | 5308.376           |
|                       |              |               | Noise         | 0.000         | 0.000        | 0.000        | 0.000             | 0.000             | 0.000             |
|                       |              |               | Variance-Bias Ratio | 5.734        | 0.001        | 0.001        | 0.001             | 0.001             | 4.768             |
|                       |              |               | Percent Change from Raw | –            | 47,310.799  | 47,310.799  | 47,310.799        | 47,310.799        | 99.347             |
|                       | Random Forest |               | Total Loss    | 102,252.318   | 1,283,608.530 | 1,283,608.530 | 1,283,608.530     | 1,283,608.530     | 102,252.318        |
|                       |              |               | Bias          | 84,950.814    | 1,277,680.008 | 1,277,680.008 | 1,277,680.008     | 1,277,680.008     | 84,950.814         |
|                       |              |               | Variance      | 17,301.504    | 5928.522     | 5928.522     | 5928.522          | 5928.522          | 17,301.504         |
|                       |              |               | Noise         | 0.000         | 0.000        | 0.000        | 0.000             | 0.000             | 0.000             |
|                       |              |               | Variance-Bias Ratio | 0.204        | 0.005        | 0.005        | 0.005             | 0.005             | 0.204             |
A.1.6. Ranked Data with Poisson Target

Table A6. Bias-variance decomposition results for ranked data with poisson target.

| Data     | Model     | Normalization | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|----------|-----------|---------------|------|------------|---------|----------------|-------------------|-------------------|
| Ranked - Poisson Target | Poisson Regression | Type of Loss | MSE   | MSE        | MSE     | MSE            | MSE               | MSE               |
|          |           | Total Loss    | 1.280 | 2.000E+18  | 8.871E+17 | 8.850E+17       | 9.325E+17         | 1.280             |
|          |           | Bias         | 1.280 | 1.009E+18  | 2.303E+17 | 2.294E+17       | 2.537E+17         | 1.280             |
|          |           | Variance     | 0.000 | 9.907E+17  | 6.568E+17 | 6.568E+17       | 6.788E+17         | 0.000             |
|          |           | Noise        | 0.000 | 4.096E+03  | 0.000E+00 | 1.261E+15       | 1.024E+03         | 0.000             |
Continued

|                     | Variance-Bias Ratio | Percent Change from Raw |
|---------------------|---------------------|-------------------------|
| Decision Tree       | 0.000               | 9.816E-01               |
|                     | 2.852E+00           | 2.863E+00               |
|                     | 2.675E+00           | 0.000                   |
| Total Loss          | 2.418               | 2.276                   |
| Bias                | 1.318               | 1.325                   |
| Variance            | 1.100               | 0.950                   |
| Noise               | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.834               | 0.717                   |
| Percent Change from Raw | 94.094           | 93.868                   |
| Random Forest       | 1.505               | 1.307                   |
| Total Loss          | 1.505               | 1.307                   |
| Bias                | 1.396               | 1.279                   |
| Variance            | 0.108               | 0.028                   |
| Noise               | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.077               | 0.022                   |
| Percent Change from Raw | 86.878           | 86.880                   |
| SVM                 | 1.281               | 1.434                   |
| Total Loss          | 1.281               | 1.353                   |
| Bias                | 1.280               | 1.339                   |
| Variance            | 0.001               | 0.096                   |
| Noise               | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.001               | 0.071                   |
| Percent Change from Raw | 111.913          | 121.171                  |
| Gradient Boosting   | 1.804               | 1.999                   |
| Total Loss          | 1.804               | 2.000                   |
| Bias                | 1.594               | 1.330                   |
| Variance            | 0.210               | 0.670                   |
| Noise               | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.132               | 0.504                   |
| Percent Change from Raw | 110.831          | 110.855                  |
| Neural Network      | 1.345               | 3224.484                |
| Total Loss          | 1.345               | 39,115.685              |
| Bias                | 1.302               | 2119.686                |
| Variance            | 0.043               | 1104.797                |
| Noise               | 0.000               | 0.000                   |
| Variance-Bias Ratio | 0.033               | 0.521                   |
| Percent Change from Raw | 239,694.606       | 2,907,696.079            |
### A.1.7. Categorical Data with Binary Target

**Table A7.** Bias-variance decomposition results for categorical data with binary target.

|          | Data               | Model     | Normalization | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|----------|--------------------|-----------|---------------|------|------------|---------|----------------|-------------------|-------------------|
| **Categorical - Binary Target** | **Logistic** | **Total Loss** |               | 0.473 | 0.476      | 0.473   | 0.473          | 0.473             | 0.473             |
|          | Bias               | 0.281     | 0.266         | 0.281 | 0.281      | 0.281   | 0.281          | 0.281             | 0.281             |
|          | Variance           | 0.192     | 0.209         | 0.192 | 0.282      | 0.192   | 0.192          | 0.192             | 0.192             |
|          | Noise              | 0.000     | 0.000         | 0.000 | 0.090      | 0.000   | 0.000          | 0.000             | 0.000             |
|          | Variance-Bias Ratio| 0.683     | 0.786         | 0.683 | 1.004      | 0.683   | 0.683          | 0.683             | 0.683             |
|          | Percent Change from Raw | –       | 100.573     | 100.000 | 100.000   | 100.000 | 100.000        | 100.000           | 100.000           |
| **Decision Tree** | **Total Loss** |               |               | 0.502 | 0.480      | 0.485   | 0.485          | 0.485             | 0.485             |
|          | Bias               | 0.338     | 0.306         | 0.306 | 0.306      | 0.306   | 0.306          | 0.306             | 0.306             |
|          | Variance           | 0.164     | 0.178         | 0.178 | 0.284      | 0.178   | 0.178          | 0.178             | 0.178             |
|          | Noise              | 0.000     | 0.000         | 0.000 | 0.105      | 0.000   | 0.000          | 0.000             | 0.000             |
|          | Variance-Bias Ratio| 0.484     | 0.710         | 0.582 | 0.926      | 0.582   | 0.582          | 0.582             | 0.582             |
|          | Percent Change from Raw | –     | 95.636       | 96.508 | 96.508   | 96.508   | 96.508         | 96.508             | 96.508             |
| **Random Forest** | **Total Loss** |               |               | 0.476 | 0.484      | 0.476   | 0.476          | 0.476             | 0.476             |
|          | Bias               | 0.288     | 0.306         | 0.288 | 0.288      | 0.288   | 0.288          | 0.288             | 0.288             |
|          | Variance           | 0.187     | 0.187         | 0.187 | 0.275      | 0.187   | 0.187          | 0.187             | 0.187             |
|          | Noise              | 0.000     | 0.000         | 0.000 | 0.085      | 0.000   | 0.000          | 0.000             | 0.000             |
|          | Variance-Bias Ratio| 0.649     | 0.714         | 0.649 | 0.944      | 0.649   | 0.649          | 0.649             | 0.649             |
|          | Percent Change from Raw | –     | 101.709     | 100.000 | 100.000   | 100.000 | 100.000        | 100.000           | 100.000           |
| **SVM**  | **Total Loss** |               |               | 0.476 | 0.473      | 0.476   | 0.476          | 0.476             | 0.476             |
|          | Bias               | 0.288     | 0.285         | 0.288 | 0.288      | 0.288   | 0.288          | 0.288             | 0.288             |
|          | Variance           | 0.188     | 0.187         | 0.188 | 0.275      | 0.188   | 0.188          | 0.188             | 0.188             |
|          | Noise              | 0.000     | 0.000         | 0.000 | 0.087      | 0.000   | 0.000          | 0.000             | 0.000             |
|          | Variance-Bias Ratio| 0.653     | 0.656         | 0.653 | 0.957      | 0.653   | 0.653          | 0.653             | 0.653             |
|          | Percent Change from Raw | –     | 99.398       | 100.000 | 100.000   | 100.000 | 100.000        | 100.000           | 100.000           |
| **Gradient Boosting** | **Total Loss** |               |               | 0.483 | 0.496      | 0.483   | 0.483          | 0.483             | 0.483             |
|          | Bias               | 0.305     | 0.326         | 0.305 | 0.305      | 0.305   | 0.305          | 0.305             | 0.305             |
|          | Variance           | 0.178     | 0.170         | 0.178 | 0.284      | 0.178   | 0.178          | 0.178             | 0.178             |
|          | Noise              | 0.000     | 0.000         | 0.000 | 0.106      | 0.000   | 0.000          | 0.000             | 0.000             |
|          | Variance-Bias Ratio| 0.585     | 0.521         | 0.585 | 0.932      | 0.585   | 0.585          | 0.585             | 0.585             |
|          | Percent Change from Raw | –     | 102.711     | 100.000 | 100.000   | 100.000 | 100.000        | 100.000           | 100.000           |
| **Neural Network** | **Total Loss** |               |               | 0.491 | 0.496      | 0.491   | 0.491          | 0.491             | 0.491             |
A.1.8. Categorical Data with Continuous Target

Table A8. Bias-variance decomposition results for categorical data with continuous target.

| Data       | Model        | Normalization | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|------------|--------------|---------------|------|------------|---------|----------------|-------------------|-------------------|
| Categorical-Continuous Target | Linear | Type of Loss | MSE | MSE | MSE | MSE | MSE |
|            |              | Total Loss    | 0.243 | 3,414,987,786,139,060,000,000.000 | 0.243 | 0.243 | 12,972,148,828,537 | 0.243 |
|            |              | Bias          | 0.241 | 2,518,113,273,062,070,000,000.000 | 0.241 | 0.241 | 27,556,466,022 | 0.241 |
|            |              | Variance      | 0.003 | 896,874,513,076,987,000,000.000 | 0.003 | 0.003 | 12,944,592,362,515 | 0.003 |
|            |              | Noise         | 0.000 | 3584.000  | 0.000 | 0.000 | 0.000 | 0.000 |
|            |              | Variance-Bias Ratio | 0.011 | 0.356 | 0.011 | 0.011 | 469.748 | 0.011 |
|            |              | Percent Change from Raw | – | 1,403,381,296,822,710,000,000,000 | 100.000 | 100.000 | 5,330,874,423,463,480 | 100.020 |
|            | Decision Tree | Total Loss    | 0.245 | 1.325 | 0.243 | 0.243 | 0.243 | 0.243 |
|            |              | Bias          | 0.239 | 1.198 | 0.240 | 0.240 | 0.240 | 0.240 |
|            |              | Variance      | 0.006 | 0.126 | 0.003 | 0.003 | 0.003 | 0.003 |
|            |              | Noise         | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
|            |              | Variance-Bias Ratio | 0.025 | 0.105 | 0.013 | 0.013 | 0.013 | 0.013 |
|            |              | Percent Change from Raw | – | 540.783 | 99.133 | 99.133 | 99.133 | 99.133 |
|            | Random Forest | Total Loss    | 0.364 | 1.605 | 0.364 | 0.364 | 0.364 | 0.364 |
|            |              | Bias          | 0.350 | 1.489 | 0.350 | 0.350 | 0.350 | 0.350 |
|            |              | Variance      | 0.013 | 0.116 | 0.013 | 0.013 | 0.013 | 0.013 |
|            |              | Noise         | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
|            |              | Variance-Bias Ratio | 0.038 | 0.078 | 0.038 | 0.038 | 0.038 | 0.038 |
|            |              | Percent Change from Raw | – | 441.086 | 100.000 | 100.000 | 100.000 | 100.000 |
|            | SVM          | Total Loss    | 0.242 | 0.689 | 0.242 | 0.242 | 0.242 | 0.242 |
|            |              | Bias          | 0.238 | 0.689 | 0.238 | 0.238 | 0.238 | 0.238 |
|            |              | Variance      | 0.004 | 0.000 | 0.004 | 0.004 | 0.004 | 0.004 |
|            |              | Noise         | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
A.1.9. Categorical Data with Poisson Target

Table A9. Bias-variance decomposition results for categorical data with poisson target.

| Data                  | Model                  | Normalization | Type of Loss | None | Z-standard | Min-Max | MaxAbs (-1, 1) | Quantile Transform | Quantile Normalize |
|-----------------------|------------------------|----------------|--------------|------|------------|---------|----------------|---------------------|------------------|
| Categorical - Poisson Target | Poisson Regression     |                | Total Loss   | MSE  | MSE        | MSE     | MSE            | MSE                 | MSE              |
|                       |                        |                | Bias         | 1.682| 1.673      | 1.682   | 1.682          | 1.682               | 1.682            |
|                       |                        |                | Variance     | 0.071| 0.074      | 0.071   | 0.071          | 0.071               | 0.071            |
|                       |                        |                | Noise        | 0.000| 0.000      | 0.000   | 0.000          | 0.000               | 0.000            |
|                       |                        |                | Variance-Bias Ratio | 0.042| 0.044      | 0.042   | 0.042          | 0.042               | 0.042            |
|                       |                        |                | Percent Change from Raw | –     | 99.698   | 100.000 | 100.000        | 100.000             | 100.000          |
| Decision Tree         |                        |                | Total Loss   | 1.559| 1.591      | 1.499   | 1.499          | 1.499               | 1.499            |
|                       |                        |                | Bias         | 1.351| 1.435      | 1.314   | 1.314          | 1.314               | 1.314            |
|                       |                        |                | Variance     | 0.208| 0.157      | 0.185   | 0.185          | 0.185               | 0.185            |
|                       |                        |                | Noise        | 0.000| 0.000      | 0.000   | 0.000          | 0.000               | 0.000            |
|                       |                        |                | Variance-Bias Ratio | 0.154| 0.109      | 0.141   | 0.141          | 0.141               | 0.141            |
|                       |                        |                | Percent Change from Raw | –     | 102.089  | 96.164  | 96.164         | 96.164              | 96.164           |
| Random Forest         |                        |                | Total Loss   | 1.581| 1.661      | 1.581   | 1.581          | 1.581               | 1.581            |
|                       |                        |                | Bias         | 1.434| 1.559      | 1.434   | 1.434          | 1.434               | 1.434            |
Continued

|                | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|----------------|----------|-------|---------------------|-------------------------|
| **Variance**   | 0.147    | 0.102 | 0.147               | 105.064                 |
| **Noise**      | 0.000    | 0.000 | 0.000               | 100.000                 |
| **Variance-Bias Ratio** | 0.103    | 0.066 | 0.103               | 100.000                 |
| **Percent Change from Raw** | –       | 105.064 | 100.000 | 100.000 |
| **Total Loss** | SVM      |       |                     |                         |
| **Bias**       | 1.754    | 1.783 | 1.754               | 1.754                   |
| **Variance**   | 0.069    | 0.043 | 0.069               | 0.069                   |
| **Noise**      | 0.000    | 0.000 | 0.000               | 0.000                   |
| **Variance-Bias Ratio** | 0.041    | 0.025 | 0.041               | 0.041                   |
| **Percent Change from Raw** | –       | 101.675 | 100.000 | 100.000 |
| **Total Loss** | Gradient Boosting |       |                     |                         |
| **Bias**       | 1.521    | 1.544 | 1.521               | 1.521                   |
| **Variance**   | 0.178    | 0.166 | 0.178               | 0.178                   |
| **Noise**      | 0.000    | 0.000 | 0.000               | 0.000                   |
| **Variance-Bias Ratio** | 0.133    | 0.121 | 0.133               | 0.133                   |
| **Percent Change from Raw** | –       | 101.547 | 100.000 | 100.000 |
| **Total Loss** | Neural Network |       |                     |                         |
| **Bias**       | 1.554    | 1.525 | 1.554               | 1.554                   |
| **Variance**   | 0.149    | 0.224 | 0.149               | 0.149                   |
| **Noise**      | 0.000    | 0.000 | 0.000               | 0.000                   |
| **Variance-Bias Ratio** | 0.106    | 0.172 | 0.106               | 0.106                   |
| **Percent Change from Raw** | –       | 98.134 | 100.007 | 99.993 |

A.1.10. Mixed Data with Binary Target

Table A10. Bias-variance decomposition results for mixed data with binary target.

| None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|------|------------|---------|----------------|--------------------|--------------------|
| 0.499| 0.499      | 0.499   | 0.497          | 0.499              | 0.496              |
| 0.311| 0.250      | 0.250   | 0.253          | 0.250              | 0.303              |
| 0.188| 0.249      | 0.250   | 0.479          | 0.249              | 0.193              |
| 0.000| 0.000      | 0.000   | 0.235          | 0.000              | 0.000              |
| 0.605| 0.999      | 1.000   | 1.891          | 0.998              | 0.638              |
| –    | 99.962     | 99.988  | 99.521         | 99.931             | 99.254             |
| 0.479| 0.504      | 0.503   | 0.503          | 0.503              | 0.492              |
| 0.267| 0.268      | 0.267   | 0.267          | 0.267              | 0.287              |
| 0.212| 0.236      | 0.236   | 0.389          | 0.236              | 0.204              |
| 0.000| 0.000      | 0.000   | 0.153          | 0.000              | 0.000              |
A.1.11. Mixed Data with Continuous Target

Table A11. Bias-variance decomposition results for mixed data with continuous target.

| Data          | Model | Normalization | Type of Loss | Linear | MSE     | MSE     | MSE     | MSE     | MSE     | MSE     | Quantile Transform | Quantile Normalize |
|---------------|-------|---------------|--------------|--------|---------|---------|---------|---------|---------|---------|--------------------|--------------------|
| Mixed Data -  |       |               |              |        | Total   | Loss    | Total   | Bias    | Variance | MSE     |                    |                    |
| Continuous    |       |               |              |        | 0.367   | 3,086,898,153,055,320,000,000 | 5,568,986,309 | 2121588770,884 | 1,895,990,182,853 | 0.363 |
| Target        |       |               |              |        | 0.362   | 2,060,070,195,326,140,000,000 | 5,568,970,874 | 2121584456,477 | 886,312  | 0.355 |
|               |       |               |              |        | 0.005   | 1,026,827,957,729,170,000,000 | 15,435 | 15,435 | 1,895,989,296,541 | 0.008 |

A.1.12. Mixed Data with Discrete Target

Table A12. Bias-variance decomposition results for mixed data with discrete target.

| Data          | Model | Normalization | Type of Loss | Linear | MSE     | MSE     | MSE     | MSE     | MSE     | MSE     | Quantile Transform | Quantile Normalize |
|---------------|-------|---------------|--------------|--------|---------|---------|---------|---------|---------|---------|--------------------|--------------------|
| Mixed Data -  |       |               |              |        | Total   | Loss    | Total   | Bias    | Variance | MSE     |                    |                    |
| Discrete      |       |               |              |        | 0.367   | 3,086,898,153,055,320,000,000 | 5,568,986,309 | 2121588770,884 | 1,895,990,182,853 | 0.363 |
| Target        |       |               |              |        | 0.362   | 2,060,070,195,326,140,000,000 | 5,568,970,874 | 2121584456,477 | 886,312  | 0.355 |
|               |       |               |              |        | 0.005   | 1,026,827,957,729,170,000,000 | 15,435 | 15,435 | 1,895,989,296,541 | 0.008 |

A.1.13. Mixed Data with Both Targets

Table A13. Bias-variance decomposition results for mixed data with both targets.

| Data          | Model | Normalization | Type of Loss | Linear | MSE     | MSE     | MSE     | MSE     | MSE     | MSE     | Quantile Transform | Quantile Normalize |
|---------------|-------|---------------|--------------|--------|---------|---------|---------|---------|---------|---------|--------------------|--------------------|
| Mixed Data -  |       |               |              |        | Total   | Loss    | Total   | Bias    | Variance | MSE     |                    |                    |
| Both Targets  |       |               |              |        | 0.367   | 3,086,898,153,055,320,000,000 | 5,568,986,309 | 2121588770,884 | 1,895,990,182,853 | 0.363 |
|               |       |               |              |        | 0.362   | 2,060,070,195,326,140,000,000 | 5,568,970,874 | 2121584456,477 | 886,312  | 0.355 |
|               |       |               |              |        | 0.005   | 1,026,827,957,729,170,000,000 | 15,435 | 15,435 | 1,895,989,296,541 | 0.008 |
### Decision Tree

|                      | Noise  | 10240.000 | 0.000 | 4298.972 | 0.000 | 0.000 |
|----------------------|--------|-----------|-------|----------|-------|-------|
| Bias                 | 0.014  | 0.498     | 0.000 | 0.000    | 2,139,189.427 | 0.022 |
| Variance             | 0.000  | 0.000     | 0.000 | 0.000    | 0.000  | 0.000 |
| Percent Change from Raw | 841,116,717,750,002,000,000 | 1,517,435,060.458 | 578,089,621,005.122 | 516,618,612,087.6 | 59 | 98.992 |

### Random Forest

|                      | Noise  | 0.000 | 23.205 | 23.205 | 23.205 | 3.921 |
|----------------------|--------|-------|--------|--------|--------|-------|
| Bias                 | 0.472  | 73.880 | 73.821 | 73.821 | 73.821 | 0.717 |
| Variance             | 0.483  | 0.398  | 0.401  | 0.401  | 0.401  | 0.476 |
| Noise                | 0.000  | 0.000  | 0.000  | 0.004  | 0.000  | 0.000 |
| Bias-Variance Ratio  | 1.023  | 0.005  | 0.005  | 0.005  | 0.005  | 0.664 |
| Percent Change from Raw | 7777.318 | 7771.442 | 7771.406 | 7771.442 | 124.898 |

### SVM

|                      | Noise  | 0.000 | 11.521 | 11.528 | 12.201 | 0.831 |
|----------------------|--------|-------|--------|--------|--------|-------|
| Bias                 | 0.783  | 11.519 | 11.528 | 11.521 | 11.952 | 0.801 |
| Variance             | 0.032  | 0.021  | 0.021  | 0.250  | 0.031  |       |
| Noise                | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000 |
| Bias-Variance Ratio  | 0.041  | 0.012  | 0.012  | 0.012  | 0.012  | 0.120 |
| Percent Change from Raw | 1412.698 | 1416.026 | 1413.574 | 1496.065 | 101.955 |

### Gradient Boosting

|                      | Noise  | 0.000 | 59.055 | 62.797 | 62.806 | 1.694 |
|----------------------|--------|-------|--------|--------|--------|-------|
| Bias                 | 1.154  | 58.342 | 62.004 | 62.013 | 62.004 | 1.151 |
| Variance             | 0.541  | 0.712  | 0.792  | 0.792  | 0.792  | 0.543 |
| Noise                | 0.000  | 0.000  | 0.000  | 0.001  | 0.000  | 0.000 |
| Bias-Variance Ratio  | 0.469  | 0.012  | 0.013  | 0.013  | 0.013  | 0.472 |
| Percent Change from Raw | 3483.593 | 3704.328 | 3704.895 | 3704.328 | 99.940 |

### Neural Network

|                      | Noise  | 0.425 | 172,936.383 | 5,567,736.822 | 2,092,341,169.038 | 1,980,462.195 |
|----------------------|--------|-------|-------------|---------------|-------------------|--------------|
| Bias                 | 0.359  | 172,879.121 | 5,567,709.809 | 2,077,346,984.312 | 1,980,266.673 | 0.355 |
| Variance             | 0.007  | 57.262  | 27.013      | 27.013        | 195.522         | 0.070 |
| Noise                | 0.000  | 0.000   | 0.000       | 14994157.713  | 0.000            | 0.000 |
| Bias-Variance Ratio  | 0.020  | 0.000   | 0.000       | 0.000         | 0.000            | 0.198 |
| Percent Change from Raw | 47,277,162.892 | 1,522,101,918.052 | 572,001,983,668.337 | 541,416,629.719 | 116.261 |
## A.1.12. Mixed Data with Poisson Target

Table A12. Bias-variance decomposition results for mixed data with poisson target.

| Data | Model | Normalization | Type of Loss | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|------|-------|---------------|--------------|------|------------|---------|----------------|--------------------|--------------------|
| Categorical - Continuous Target | Poisson Regression | Total Loss | MSE | 1.929 | 631.554 | 2.208 | 2.070 | 6.928 | 1.929 |
| | | Bias | MSE | 1.743 | 6.792 | 1.801 | 1.928 | 1.271 | 1.740 |
| | | Variance | MSE | 0.186 | 624.762 | 0.407 | 0.407 | 5.657 | 0.188 |
| | | Noise | MSE | 0.000 | 0.000 | 0.000 | 0.264 | 0.000 | 0.000 |
| | | Variance-Bias Ratio | MSE | 0.107 | 91.989 | 0.226 | 0.211 | 4.452 | 0.108 |
| | | Percent Change from Raw | MSE | – | 32.734.401 | 114.423 | 107.289 | 359.076 | 99.957 |
| | Decision Tree | Total Loss | MSE | 2.087 | 2.127 | 2.137 | 2.140 | 2.137 | 2.119 |
| | | Bias | MSE | 1.123 | 1.199 | 1.213 | 1.212 | 1.213 | 1.264 |
| | | Variance | MSE | 0.964 | 0.927 | 0.924 | 0.924 | 0.924 | 0.855 |
| | | Noise | MSE | 0.000 | 0.000 | 0.000 | 0.004 | 0.000 | 0.000 |
| | | Variance-Bias Ratio | MSE | 0.859 | 0.773 | 0.762 | 0.762 | 0.762 | 0.677 |
| | | Percent Change from Raw | MSE | – | 101.899 | 102.382 | 102.521 | 102.382 | 101.557 |
| | Random Forest | Total Loss | MSE | 1.820 | 2.129 | 2.113 | 2.113 | 2.113 | 1.821 |
| | | Bias | MSE | 1.608 | 2.029 | 1.997 | 1.997 | 1.997 | 1.609 |
| | | Variance | MSE | 0.212 | 0.100 | 0.116 | 0.116 | 0.116 | 0.212 |
| | | Noise | MSE | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | | Variance-Bias Ratio | MSE | 0.132 | 0.049 | 0.058 | 0.058 | 0.058 | 0.132 |
| | | Percent Change from Raw | MSE | – | 116.956 | 116.100 | 116.100 | 116.100 | 100.040 |
| | SVM | Total Loss | MSE | 1.940 | 2.029 | 2.201 | 2.247 | 2.188 | 1.938 |
| | | Bias | MSE | 1.775 | 1.865 | 2.157 | 2.244 | 2.133 | 1.771 |
| | | Variance | MSE | 0.165 | 0.164 | 0.044 | 0.044 | 0.055 | 0.166 |
| | | Noise | MSE | 0.000 | 0.000 | 0.000 | 0.041 | 0.000 | 0.000 |
| | | Variance-Bias Ratio | MSE | 0.093 | 0.088 | 0.020 | 0.020 | 0.026 | 0.094 |
| | | Percent Change from Raw | MSE | – | 104.577 | 113.429 | 115.807 | 112.765 | 99.867 |
| | Gradient Boosting | Total Loss | MSE | 1.746 | 2.042 | 2.000 | 2.001 | 2.000 | 1.746 |
| | | Bias | MSE | 1.531 | 1.876 | 1.787 | 1.787 | 1.787 | 1.530 |
| | | Variance | MSE | 0.216 | 0.166 | 0.213 | 0.213 | 0.213 | 0.216 |
| | | Noise | MSE | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | | Variance-Bias Ratio | MSE | 0.141 | 0.089 | 0.119 | 0.119 | 0.119 | 0.141 |
### A.2. Benchmark Results

#### A.2.1. Wine Quality Data with Binary Target

Table A13. Bias-variance decomposition results for wine quality data with binary target.

| Data                     | Model      | Normalization | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|--------------------------|------------|---------------|------|------------|---------|----------------|--------------------|--------------------|
| Wine Quality with binary target | Logistic   | Total Loss    | 0.038 | 0.058      | 0.100   | 0.166          | 0.062              | 0.038              |
|                          |            | Bias          | 0.037 | 0.036      | 0.037   | 0.047          | 0.034              | 0.037              |
|                          |            | Variance      | 0.000 | 0.022      | 0.063   | 0.144          | 0.028              | 0.001              |
|                          |            | Noise         | 0.000 | 0.000      | 0.000   | 0.024          | 0.000              | 0.000              |
|                          |            | Variance-Bias Ratio | 0.010 | 0.609   | 1.710   | 3.086          | 0.809              | 0.022              |
|                          |            | Percent Change from Raw | – | 153.636 | 265.145 | 440.891 | 165.508 | 100.728 |
| Decision Tree            |            | Total Loss    | 0.061 | 0.772      | 0.696   | 0.684          | 0.688              | 0.061              |
|                          |            | Bias          | 0.030 | 0.608      | 0.491   | 0.475          | 0.481              | 0.030              |
|                          |            | Variance      | 0.031 | 0.164      | 0.205   | 0.046          | 0.207              | 0.031              |
|                          |            | Noise         | 0.000 | 0.000      | 0.000   | 0.162          | 0.000              | 0.000              |
|                          |            | Variance-Bias Ratio | 1.033 | 0.269   | 0.417   | 0.098          | 0.430              | 1.034              |
|                          |            | Percent Change from Raw | – | 1259.483 | 1135.652 | 1116.508 | 1123.095 | 100.079 |
| Random Forest            |            | Total Loss    | 0.039 | 0.822      | 0.762   | 0.754          | 0.749              | 0.039              |
|                          |            | Bias          | 0.031 | 0.692      | 0.593   | 0.581          | 0.572              | 0.031              |
|                          |            | Variance      | 0.008 | 0.129      | 0.169   | 0.127          | 0.177              | 0.008              |
|                          |            | Noise         | 0.000 | 0.000      | 0.000   | 0.047          | 0.000              | 0.000              |
|                          |            | Variance-Bias Ratio | 0.263 | 0.187   | 0.285   | 0.218          | 0.310              | 0.263              |
|                          |            | Percent Change from Raw | – | 2087.923 | 1936.937 | 1916.597 | 1902.651 | 100.093 |
| SVM                      |            | Total Loss    | 0.038 | 0.037      | 0.037   | 0.037          | 0.037              | 0.038              |
|                          |            | Bias          | 0.035 | 0.037      | 0.037   | 0.037          | 0.037              | 0.035              |
|                          |            | Variance      | 0.003 | 0.000      | 0.000   | 0.000          | 0.000              | 0.003              |
### A.2.2. Breast Cancer Data with Binary Target

#### Table A14. Bias-variance decomposition results for breast cancer data with binary target.

| Data                        | Model   | Normalization | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|-----------------------------|---------|---------------|------|------------|---------|----------------|--------------------|--------------------|
| Breast cancer data with binary target | Logistic | Total Loss    | 0.043 | 0.626 | 0.626 | 0.626 | 0.626 | 0.042 |
|                             | Bias    | 0.029 | 0.626 | 0.626 | 0.626 | 0.626 | 0.028 |
|                             | Variance| 0.013 | 0.000 | 0.000 | 0.000 | 0.000 | 0.014 |
|                             | Noise   | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
|                             | Variance-Bias Ratio | 0.451 | 0.000 | 0.000 | 0.000 | 0.000 | 0.487 |
|                             | Percent Change from Raw | 1465.753 | 1465.753 | 1465.753 | 1465.753 | 1465.753 | 98.205 |
| Decision Tree               | Total Loss | 0.091 | 0.601 | 0.514 | 0.488 | 0.536 | 0.090 |
|                             | Bias    | 0.047 | 0.566 | 0.284 | 0.239 | 0.322 | 0.046 |
|                             | Variance| 0.043 | 0.035 | 0.231 | 0.370 | 0.214 | 0.044 |
|                             | Noise   | 0.000 | 0.000 | 0.000 | 0.121 | 0.000 | 0.000 |
|                             | Variance-Bias Ratio | 0.911 | 0.062 | 0.814 | 1.544 | 0.667 | 0.951 |
|                             | Percent Change from Raw | 663.908 | 568.172 | 539.083 | 592.106 | 99.025 |
| Random Forest               | Total Loss | 0.059 | 0.626 | 0.551 | 0.498 | 0.590 | 0.059 |
|                             | Bias    | 0.048 | 0.626 | 0.449 | 0.289 | 0.512 | 0.048 |
### A.2.3. Voting Data with Binary Target

**Table A15.** Bias-variance decomposition results for congressional voting data with binary.

| Data                | Model       | Normalization     | None | Z-standard | Min-Max | MaxAbs (−1, 1) | Quantile Transform | Quantile Normalize |
|---------------------|-------------|-------------------|------|-----------|---------|----------------|--------------------|--------------------|
| Voting data with binary target | Logistic    | Total Loss        | 0.058 | 0.112 | 0.058 | 0.058 | 0.058 | 0.059 |
|                     |             |                   |      |          |         |                 |                    |                    |
|                     | Bias        |                   | 0.046 | 0.078 | 0.046 | 0.046 | 0.046 | 0.048 |
|                     | Variance    |                   | 0.011 | 0.034 | 0.011 | 0.017 | 0.011 | 0.011 |
|                     | Noise       |                   | 0.000 | 0.000 | 0.000 | 0.005 | 0.000 | 0.000 |
|                     | Variance-Bias Ratio |       | 0.247 | 0.435 | 0.247 | 0.365 | 0.247 | 0.227 |
|                     | Percent Change from Raw |     | 193.123 | 100.000 | 100.000 | 100.000 | 101.162 |
### Decision Tree

|           | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-----------|------------|------|----------|-------|--------------------|-------------------------|
|           | 0.066      | 0.036| 0.030    | 0.000 | 0.853              | –                       |
|           | 0.063      | 0.040| 0.023    | 0.000 | 0.567              | 96.100                  |
|           | 0.066      | 0.036| 0.030    | 0.000 | 0.816              | 100.475                 |
|           | 0.066      | 0.036| 0.043    | 0.013 | 1.182              | 100.475                 |
|           | 0.066      | 0.036| 0.030    | 0.000 | 0.816              | 100.475                 |
|           | 0.066      | 0.036| 0.013    | 0.000 | 0.816              | 100.475                 |

### Random Forest

|           | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-----------|------------|------|----------|-------|--------------------|-------------------------|
|           | 0.044      | 0.031| 0.012    | 0.000 | 0.388              | –                       |
|           | 0.053      | 0.041| 0.012    | 0.000 | 0.297              | 121.723                 |
|           | 0.044      | 0.031| 0.018    | 0.005 | 0.561              | 100.000                 |
|           | 0.044      | 0.031| 0.012    | 0.000 | 0.388              | 100.000                 |
|           | 0.044      | 0.031| 0.012    | 0.000 | 0.388              | 100.000                 |
|           | 0.044      | 0.031| 0.012    | 0.000 | 0.388              | 100.000                 |

### SVM

|           | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-----------|------------|------|----------|-------|--------------------|-------------------------|
|           | 0.052      | 0.045| 0.007    | 0.000 | 0.162              | –                       |
|           | 0.054      | 0.038| 0.016    | 0.000 | 0.427              | 102.883                 |
|           | 0.052      | 0.045| 0.011    | 0.004 | 0.162              | 100.000                 |
|           | 0.052      | 0.045| 0.007    | 0.000 | 0.162              | 100.000                 |
|           | 0.052      | 0.045| 0.007    | 0.000 | 0.162              | 100.000                 |
|           | 0.052      | 0.045| 0.007    | 0.000 | 0.162              | 100.000                 |

### Gradient Boosting

|           | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-----------|------------|------|----------|-------|--------------------|-------------------------|
|           | 0.056      | 0.033| 0.022    | 0.000 | 0.668              | –                       |
|           | 0.058      | 0.039| 0.022    | 0.000 | 0.472              | 103.490                 |
|           | 0.056      | 0.033| 0.032    | 0.009 | 0.949              | 100.000                 |
|           | 0.056      | 0.033| 0.022    | 0.000 | 0.668              | 100.000                 |
|           | 0.056      | 0.033| 0.022    | 0.000 | 0.668              | 100.000                 |
|           | 0.056      | 0.033| 0.022    | 0.000 | 0.668              | 100.000                 |

### Neural Network

|           | Total Loss | Bias | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|-----------|------------|------|----------|-------|--------------------|-------------------------|
|           | 0.059      | 0.045| 0.014    | 0.000 | 0.312              | –                       |
|           | 0.079      | 0.059| 0.021    | 0.000 | 0.353              | 134.324                 |
|           | 0.059      | 0.045| 0.020    | 0.006 | 0.442              | 100.000                 |
|           | 0.059      | 0.045| 0.014    | 0.000 | 0.312              | 100.000                 |
|           | 0.059      | 0.045| 0.014    | 0.000 | 0.312              | 100.000                 |
|           | 0.059      | 0.045| 0.014    | 0.000 | 0.312              | 100.000                 |

### Percent Change from Raw

- Decision Tree: ~96.100 100.475 100.475 100.475 100.475
- Random Forest: 121.723 100.000 100.000 100.000 100.000
- SVM: 102.883 100.000 100.000 100.000 99.649
- Gradient Boosting: 103.490 100.000 100.000 100.000
- Neural Network: 134.324 100.000 100.000 100.000 98.049
### A.2.4. Abalone Data with Binary Target

**Table A16. Bias-variance decomposition results for abalone data with binary target.**

| Data                        | Model     | Normalization | None | Z-standard | Min-Max | MaxAbs (-1, 1) | Quantile Transform | Quantile Normalize |
|-----------------------------|-----------|---------------|------|------------|---------|---------------|--------------------|--------------------|
| **Abalone data with binary target** | **Logistic** | **Total Loss** | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 |
|                            |           | Bias          | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 |
|                            |           | Variance      | 0.000 | 0.000 | 0.000 | 0.144 | 0.000 | 0.000 |
|                            |           | Noise         | 0.000 | 0.000 | 0.000 | 0.144 | 0.000 | 0.000 |
|                            |           | Variance-Bias Ratio | 0.000 | 0.000 | 0.000 | 1.542 | 0.000 | 0.000 |
|                            |           | Percent Change from Raw | – | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 |
| **Decision Tree**          |           | **Total Loss** | 0.145 | 0.102 | 0.230 | 0.252 | 0.391 | 0.145 |
|                            |           | Bias          | 0.079 | 0.091 | 0.098 | 0.111 | 0.174 | 0.079 |
|                            |           | Variance      | 0.066 | 0.011 | 0.131 | 0.046 | 0.217 | 0.066 |
|                            |           | Noise         | 0.000 | 0.000 | 0.000 | 0.095 | 0.000 | 0.000 |
|                            |           | Variance-Bias Ratio | 0.847 | 0.124 | 1.340 | 0.419 | 1.248 | 0.847 |
|                            |           | Percent Change from Raw | – | 70.193 | 158.349 | 173.704 | 269.903 | 100.000 |
| **Random Forest**          |           | **Total Loss** | 0.108 | 0.093 | 0.122 | 0.127 | 0.149 | 0.108 |
|                            |           | Bias          | 0.081 | 0.093 | 0.077 | 0.082 | 0.079 | 0.081 |
|                            |           | Variance      | 0.027 | 0.000 | 0.045 | 0.127 | 0.070 | 0.027 |
|                            |           | Noise         | 0.000 | 0.000 | 0.000 | 0.081 | 0.000 | 0.000 |
|                            |           | Variance-Bias Ratio | 0.337 | 0.000 | 0.587 | 1.554 | 0.881 | 0.337 |
|                            |           | Percent Change from Raw | – | 86.254 | 112.596 | 117.182 | 138.182 | 100.000 |
| **SVM**                    |           | **Total Loss** | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 |
|                            |           | Bias          | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 |
|                            |           | Variance      | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
|                            |           | Noise         | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
|                            |           | Variance-Bias Ratio | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
|                            |           | Percent Change from Raw | – | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 |
| **Gradient Boosting**      |           | **Total Loss** | 0.103 | 0.094 | 0.171 | 0.213 | 0.237 | 0.103 |
|                            |           | Bias          | 0.081 | 0.093 | 0.086 | 0.109 | 0.105 | 0.081 |
|                            |           | Variance      | 0.021 | 0.001 | 0.085 | 0.003 | 0.132 | 0.021 |
|                            |           | Noise         | 0.000 | 0.000 | 0.000 | 0.102 | 0.000 | 0.000 |
|                            |           | Variance-Bias Ratio | 0.261 | 0.015 | 0.985 | 0.025 | 1.263 | 0.261 |
|                            |           | Percent Change from Raw | – | 92.110 | 166.500 | 207.969 | 230.980 | 100.000 |
| **Neural Network**         |           | **Total Loss** | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 | 0.093 |
Continued

| Data                          | Model | Normalization | None  | Z-standard | Min-Max | MaxAbs (-1, 1) | Quantile Transform | Quantile Normalize |
|-------------------------------|-------|---------------|-------|------------|---------|----------------|-------------------|-------------------|
| Arrhythmia data with binary target | Logistic | Total Loss    | 0.065 | 0.875      | 0.755   | 0.636          | 0.905             | 0.064             |
|                               |       | Bias          | 0.046 | 0.828      | 0.650   | 0.476          | 0.877             | 0.046             |
|                               |       | Variance      | 0.018 | 0.047      | 0.105   | 0.144          | 0.027             | 0.018             |
|                               |       | Noise         | 0.000 | 0.000      | 0.000   | 0.016          | 0.000             | 0.000             |
|                               |       | Variance-Bias Ratio | 0.398 | 0.057      | 0.161   | 0.302          | 0.031             | 0.395             |
|                               |       | Percent Change from Raw | –     | 1350.982   | 1165.815| 983.032        | 1397.399          | 98.683            |
| Decision Tree                |       | Total Loss    | 0.034 | 0.094      | 0.099   | 0.098          | 0.100             | 0.034             |
|                               |       | Bias          | 0.018 | 0.056      | 0.055   | 0.056          | 0.055             | 0.018             |
|                               |       | Variance      | 0.016 | 0.038      | 0.044   | 0.046          | 0.044             | 0.016             |
|                               |       | Noise         | 0.000 | 0.000      | 0.000   | 0.004          | 0.000             | 0.000             |
|                               |       | Variance-Bias Ratio | 0.907 | 0.691      | 0.796   | 0.834          | 0.798             | 0.907             |
|                               |       | Percent Change from Raw | –     | 274.608    | 290.841 | 286.111        | 291.056           | 100.000           |
| Random Forest                |       | Total Loss    | 0.059 | 0.111      | 0.172   | 0.153          | 0.160             | 0.059             |
|                               |       | Bias          | 0.059 | 0.057      | 0.063   | 0.062          | 0.062             | 0.059             |
|                               |       | Variance      | 0.001 | 0.054      | 0.108   | 0.127          | 0.098             | 0.001             |
|                               |       | Noise         | 0.000 | 0.000      | 0.000   | 0.035          | 0.000             | 0.000             |
|                               |       | Variance-Bias Ratio | 0.013 | 0.937      | 1.708   | 2.057          | 1.594             | 0.013             |
|                               |       | Percent Change from Raw | –     | 186.609    | 289.307 | 258.218        | 268.824           | 100.000           |
| SVM                          |       | Total Loss    | 0.059 | 0.059      | 0.059   | 0.059          | 0.059             | 0.059             |
|                               |       | Bias          | 0.059 | 0.059      | 0.059   | 0.059          | 0.059             | 0.059             |
|                               |       | Variance      | 0.000 | 0.000      | 0.000   | 0.000          | 0.000             | 0.000             |
|                               |       | Noise         | 0.000 | 0.000      | 0.000   | 0.000          | 0.000             | 0.000             |
|                               |       | Variance-Bias Ratio | 0.000 | 0.000      | 0.000   | 0.000          | 0.000             | 0.000             |
|                               |       | Percent Change from Raw | –     | 100.000    | 100.000 | 100.000        | 100.000           | 100.000           |
Continued

|                          | Total Loss | Bias     | Variance | Noise | Variance-Bias Ratio | Percent Change from Raw |
|--------------------------|------------|----------|----------|-------|---------------------|-------------------------|
| Gradient Boosting        |            | 0.033    | 0.019    | 0.014 | 0.768               | ~ 184.039               |
|                          |            | 0.061    | 0.059    | 0.002 | 0.036               | 185.689                 |
|                          |            | 0.061    | 0.059    | 0.000 | 0.046               | 185.823                 |
|                          |            | 0.033    | 0.019    | 0.002 | 0.046               | 185.800                 |
|                          |            | 0.061    | 0.014    | 0.000 | 0.000               | 100.000                 |
| Neural Network           |            | 0.071    | 0.057    | 0.014 | 0.251               | ~ 547.100               |
|                          |            | 0.387    | 0.206    | 0.181 | 0.880               | 83.160                  |
|                          |            | 0.059    | 0.059    | 0.000 | 0.000               | 86.268                  |
|                          |            | 0.061    | 0.058    | 0.000 | 0.000               | 104.927                 |
|                          |            | 0.074    | 0.057    | 0.000 | 0.309               | 102.682                 |
|                          |            | 0.073    | 0.057    | 0.000 | 0.268               |                        |

A.2.6. Forest Fires Data with Continuous Target

Table A18. Bias-variance decomposition results for forest fires data with continuous target.

| Min-Max | MaxAbs (~1, 1) | Quantile Transform | Quantile Normalize |
|---------|----------------|--------------------|--------------------|
| MSE     | MSE            | MSE                | MSE                |
| 54,418,873.320 | 61,656,927.105 | 77,782,033.117 | 8254.563           |
| 38,494,821.256 | 43,230,677.674 | 12,450,153.080 | 8188.901           |
| 15,924,052.064 | 15,924,052.064 | 65,331,880.037 | 65.662             |
| 0.000   | 2,502,197.367  | 0.000              | 0.000              |
| 0.414   | 0.368          | 5.247              | 0.008              |
| 659,258.104 | 746,943.595   | 942,291.388       | 100.000            |
| 191,218.064 | 191,189.902   | 191,069.504       | 14,564.197         |
| 82,269.473  | 82,240.074    | 82,166.683        | 9705.018           |
| 108,948.592 | 108,902.820   | 108,802.232       | 4859.179           |
| 0.000   | 1.237          | 0.000              | 0.000              |
| 1.324   | 1.325          | 1.325              | 0.501              |
| 1312.932 | 1312.739      | 1311.912          | 100.000            |
| 83,921.663 | 83,911.156    | 83,802.232        | 9973.155           |
| 60,221.253 | 60,210.717    | 60,139.045        | 9118.659           |
| 23,700.410 | 23,700.410    | 23,663.187        | 854.497            |
| 0.000   | 0.029          | 0.000              | 0.000              |
| 0.394   | 0.394          | 0.393              | 0.094              |
| 841.476 | 841.370        | 840.278           | 100.000            |
| 8540.229 | 8542.123      | 8535.867          | 8527.384           |
A.2.7. Solar Flares Data with Continuous Target

Table A19. Bias-variance decomposition results for solar flares data with continuous target.

| Data                | Model | Normalization | None | Z-standard | Min-Max | MaxAbs (-1, 1) | Quantile Transform | Quantile Normalize | Type of Loss | MSE | MSE | MSE | MSE | MSE | MSE |
|---------------------|-------|---------------|------|------------|---------|---------------|-------------------|-------------------|---------------|-----|-----|-----|-----|-----|-----|
| Solar Flare Data   | Linear |               |      |            |         |               |                   |                   | Total Loss    | 0.376 | 0.457 | 0.376 | 0.376 | 0.376 | 0.376 |
| - Continuous Target|       |               |      |            |         |               |                   |                   | Bias          | 0.346 | 0.454 | 0.345 | 0.346 | 0.346 | 0.345 |
|                     |       |               |      |            |         |               |                   |                   | Variance      | 0.030 | 0.002 | 0.031 | 0.031 | 0.030 | 0.031 |
|                     |       |               |      |            |         |               |                   |                   | Noise         | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 |
|                     |       |               |      |            |         |               |                   |                   | Variance-Bias Ratio | 0.087 | 0.005 | 0.089 | 0.089 | 0.087 | 0.089 |
|                     |       |               |      |            |         |               |                   |                   | Percent Change from Raw | –          | 121.569 | 100.043 | 100.000 | 100.000 | 100.127 |
|                     |       |               |      |            |         |               |                   |                   | Decision Tree |               |                   |                   |                   |                   |                   |
|                     |       |               |      |            |         |               |                   |                   | Total Loss    | 0.769 | 0.734 | 0.769 | 0.769 | 0.769 | 0.772 |
|                     |       |               |      |            |         |               |                   |                   | Bias          | 0.534 | 0.408 | 0.534 | 0.534 | 0.534 | 0.534 |
|                     |       |               |      |            |         |               |                   |                   | Variance      | 0.235 | 0.326 | 0.235 | 0.235 | 0.235 | 0.237 |
|                     |       |               |      |            |         |               |                   |                   | Noise         | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
|                     |       |               |      |            |         |               |                   |                   | Bias-Variance Ratio | 0.440 | 0.799 | 0.440 | 0.440 | 0.440 | 0.444 |
|                     |       |               |      |            |         |               |                   |                   | Percent Change from Raw | –          | 95.427 | 100.000 | 100.000 | 100.000 | 100.413 |
### A.2.8. Auto MPG Data with Continuous Target

**Table A20.** Bias-variance decomposition results for auto mpg data with continuous target.

| Data | Model | Normalization | None | Z-standard | Min-Max | MaxAbs ($-1,1$) | Quantile Transform | Quantile Normalize |
|------|-------|----------------|------|------------|---------|-----------------|--------------------|-------------------|
| Auto mpg Data - Continuous Target | Linear | Type of Loss | MSE | MSE | MSE | MSE | MSE | MSE |
| **Total Loss** | 12.98046102 | 294,549,287 | 21.07621525 | 10,679,343,815 | 1,330,291,933 | 12.98015593 | 1,162.369 | 82,272,453,970.192 | 10,248,418,231.961 | 99.998 |
| Bias | 12.26544827 | 28,970,4291.2 | 15.90522097 | 10,498,618,797 | 1,271,499,879 | 12.26513305 | 82,272,453,970.192 | 10,248,418,231.961 | 99.998 |
| Variance | 0.715012746 | 4,844,995.802 | 0.343 | 0.346 | 0.346 | 0.343 | 0.346 | 0.343 | 0.343 |
| Noise | 0.000 | 0.000 | 0.000 | 180,725,013.172 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Bias-Variance Ratio | 0.058 | 0.017 | 0.325 | 0.000 | 0.046 | 0.058 | 0.046 | 0.058 | 0.058 |
| Percent Change from Raw | – | 2,269,174,312.476 | 162.369 | 82,272,453,970.192 | 10,248,418,231.961 | 99.998 | 1,162.369 | 82,272,453,970.192 | 10,248,418,231.961 | 99.998 |
### Decision Tree

|               | Total Loss | Bias | Variance | Noise | Bias-Variance Ratio | Percent Change from Raw |
|---------------|------------|------|----------|-------|---------------------|-------------------------|
| **Random Forest** | Total Loss | Bias | Variance | Noise | Bias-Variance Ratio | Percent Change from Raw |
| **SVM**       | Total Loss | Bias | Variance | Noise | Bias-Variance Ratio | Percent Change from Raw |
| **Gradient Boosting** | Total Loss | Bias | Variance | Noise | Bias-Variance Ratio | Percent Change from Raw |
| **Neural Network** | Total Loss | Bias | Variance | Noise | Bias-Variance Ratio | Percent Change from Raw |
Appendix B

B.1. Simulations

B.1.1. Bivariate Normal Data with Binary Target

Figure B1. Bias-variance decomposition for bivariate normal data with binary target.
B.1.2. Bivariate Normal Data with Continuous Target

Figure B2. Bias-variance decomposition for bivariate normal data with continuous target.
B.1.3. Bivariate Normal Data with Poisson Target

**Figure B3.** Bias-variance decomposition for bivariate normal data with poisson target.
B.1.4. Ranked Data with Binary Target

Figure B4. Bias-variance decomposition for ranked data with binary target.
B.1.5. Ranked Data with Continuous Target

![Bias-variance decomposition for ranked data with continuous target.](image)

Figure B5. Bias-variance decomposition for ranked data with continuous target.
B.1.6. Ranked Data with Poisson Target

Figure B6. Bias-variance decomposition for ranked data with poisson target.
B.1.7. Categorical Data with Binary Target

Figure B7. Bias-variance decomposition for categorical data with binary target.
B.1.8. Categorical Data with Continuous Target

**Figure B8.** Bias-variance decomposition for categorical data with continuous target.
B.1.9. Categorical Data with Poisson Target

![Bias-variance decomposition for categorical data with poisson target.](image)

*Figure B9.* Bias-variance decomposition for categorical data with poisson target.
B.1.10. Mixed Data with Binary Target

Figure B10. Bias-variance decomposition for mixed data with binary target.
B.1.11. Mixed Data with Continuous Target

![Bias-variance decomposition for mixed data with continuous target.](image)

**Figure B11.** Bias-variance decomposition for mixed data with continuous target.
B.1.12. Mixed Data with Poisson Target

Figure B12. Bias-variance decomposition for mixed data with poisson target.
B.2. Benchmark Data Results

B.2.1. Wine Quality Data

Figure B13. Bias-variance decomposition for wine quality data with binary target.
B.2.2. Breast Cancer Data

Figure B14. Bias-variance decomposition for breast cancer data with binary target.
B.2.3. Voting Data

**Figure B15.** Bias-variance decomposition for congressional voting data with binary target.
B.2.4. Abalone Data

Figure B16. Bias-variance decomposition for abalone data with binary target.
B.2.5. Arrhythmia Data

Figure B17. Bias-variance decomposition for arrhythmia data with binary target.
B.2.6. Forest Fires Data

Figure B18. Bias-variance decomposition for forest fires data with continuous target.
B.2.7. Solar Flares Data

Figure B19. Bias-variance decomposition for solar flares data with continuous target.
B.2.8. Auto MPG Data

Figure B20. Bias-variance decomposition for auto MPG data with continuous target.