Identifying and Overcoming Transformation Bias in Forecasting Models

Sushant More
morsusha@amazon.com
Amazon

ABSTRACT

Log and square root transformations of target variable are routinely used in forecasting models to predict future sales. These transformations often lead to better performing models. However, they also introduce a systematic negative bias (under-forecasting). In this paper, we demonstrate the existence of this bias, dive deep into its root cause and introduce two methods to correct for the bias. We conclude that the proposed bias correction methods improve model performance (by up to 50%) and make a case for incorporating bias correction in modeling workflow.

We also experiment with 'Tweedie' family of cost functions which circumvents the transformation bias issue by modeling directly on sales. We conclude that Tweedie regression gives the best performance so far when modeling on sales making it a strong alternative to working with a transformed target variable.

ACM Reference Format:
. 2022. Identifying and Overcoming Transformation Bias in Forecasting Models. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

One of the science goals in Amazon Devices is to determine which device each customer wants to buy, and ensure that every customer's choice is available for them to purchase at the time and place they desire. To deliver on this goal, the science team has invested in developing device demand forecasting models, which we call Versioned Demand Planning (VDP) models. The performance of VDP models is usually monitored at a region and device type level.[1].

An investigation of these VDP models indicated that many of the models (built using $log(sales)$ as target variable) suffered from an issue of negative bias or under-forecasting. This is demonstrated in Fig. 1. To maintain business confidentiality, we anonymize the regions by roman numerals (I, II, III, . . .) and device types by uppercase English letters (A, B, C). Furthermore the metrics – WMAPE (weighted mean-absolute percentage error) and WBias (weighted bias) are reported relative to the performance of Linear Regression (LR) model for region I and device type A throughout the paper. For a given model for a region r, device d, and time horizon h:

$$\text{reported WMAPE}_{\text{r, d, h, model}} = \frac{\text{WMAPE}_{\text{r, d, h, model}}}{\text{WMAPE}_{\text{r=I, d=A, h, model=LR}}}$$

$$\text{reported WBias}_{\text{r, d, h, model}} = \frac{\text{WBias}_{\text{r, d, h, model}}}{|\text{WBias}_{\text{r=I, d=A, h, model=LR}}|} \quad (1)$$

Please refer to Appendix A (Eq. (9) regarding how we define the MAPE and Bias metrics. As seen in Fig. 1, a systematic negative bias is present regardless of the model choice (XGBoost, Random Forest, Linear Regression, Neural networks, ARIMAX) and is seen irrespective of the region/device type combination.

The above observation will form the basis of this paper. It is arranged as follows. In the next section, we dive deep into the scientific basis of the observed under-forecasting. After establishing the root cause for bias in Sec. 2, we look into various methods of bias correction in Sec. 3. In Sec. 4, we look at the methods that circumvent the transformation bias altogether by building models directly on sales without invoking any transformation. We summarize our findings and give glimpse of the ongoing work in Sec. 5.

Region	device_type	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
II	B	XGB_linear	-0.74	0.69	-1.16	0.61	-0.72	0.33
		XGB_tree	-0.50	0.64	-0.87	0.59	-0.47	0.53
	C	XGB_linear	-0.46	0.90	-0.63	0.94	-0.92	0.66
		XGB_tree	-0.58	0.58	-0.89	0.58	-1.25	0.43

*we use weighted average of forecast versions from VDP_20190812 to VDP_20200302. Target variable: log(sales)

Region	device_type	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
II	B	Random Forest	-0.88	0.67	-1.42	0.74	-1.50	0.41
	C	Random Forest	-0.85	0.63	-1.26	0.64	-1.83	0.50

*we use weighted average of forecast versions from VDP_20190812 to VDP_20200302. Target variable: log(sales)

Region	device_type	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
I	A	Neural Network	-0.62	0.69	-0.42	0.70	-0.17	0.59

*we use weighted average of forecast versions from VDP_20191104 to VDP_20200525. Target variable: square_root(sales)

Region	device type	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
I	A	ARIMAX	-0.81	0.54	-1.16	0.52	-2.17	0.59

*we use weighted average of forecast versions from VDP_20190121 to VDP_20190812. Target variable: log(sales)

Figure 1: Negative Bias across various models when modeled on a transformed target variable. Persistent negative bias is seen irrespective of the model type (XGBoost, Random Forest, Linear Regression, Neural Networks, and ARIMAX) and region/ device type combination. We report relative metrics (Eq. (1)) for 4 different time horizons (6, 12, and 24 weeks).

[1]Examples of device types are Alexa Echo, Kindle/ eReader), Tablet, Fire TV etc.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

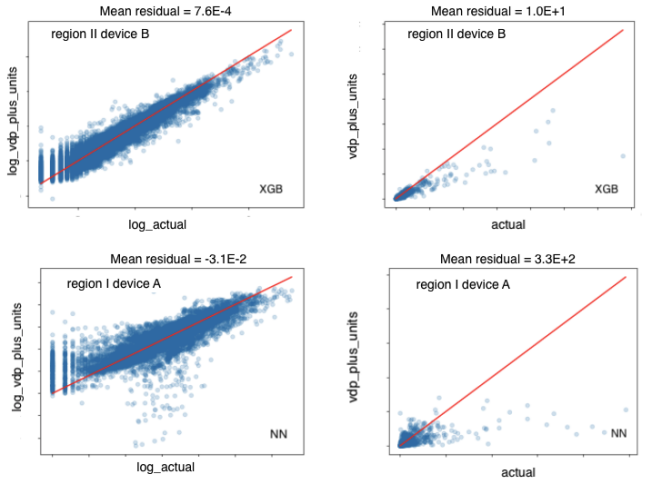

Figure 2: In-sample model fit for two machine learning models for two different device types. The plots on the left are the fit in the modeling units. The X and Y axes are target variable (Y) and the prediction output from the ML model ($\hat{Y}$) respectively. The plots on the right are obtained by back-transforming the values from plots on the left. The red line is y = x line. Points below the red line are under-forecasts.

Note that even though we use device forecasting as our test ground, the problem of transformation bias that we tackle and the methods we develop are applicable to any regression setting where the target variable is transformed before fitting a Machine Learning model.

2 ROOT CAUSE OF NEGATIVE BIAS

Our goal is to predict daily sales for each individual item given the relevant features such as selling price, seasonality, holidays, in-stock status, and product attributes. We build models on past 2-3 years of sales history and predict daily sales at individual item grain for 6-12 months in future.

Historically, we found it useful (see Sec. 2.2) to build machine learning (ML) models with $log(sales)$ as target variable and exponentiate the model prediction. To root cause the chronic issue of negative backtesting bias (refer Fig. 1), we focus on the simplest ML problem of in-sample model fit by comparing model prediction ($\hat{Y}$) with target variable (Y).

In Fig. 2, the graphs on the left are in-sample model fits in the modeling (i.e, logarithmic) units. Note that we refer to model prediction as 'vdp_plus_units'. The plots on the right is when the same model output is back-transformed (exponentiated) to the original units (without any new model fit). Residual = $Y - \hat{Y}$ is the difference between the original value and the prediction. Ideally, we want the residuals to have a mean of zero. Large positive mean for residuals implies a negative bias (or under-forecasting). Fig. 2 shows that: 1) Models do not show a bias in the modeling units. The mean of residuals is close to zero (refer plots' title). 2) A large negative bias is introduced when the prediction is back-transformed to original units. This is seen by a large positive mean of residuals (plots on right in Fig. 2).

Even though, we only demonstrate the effect of target transformation in XGBoost (XGB) and Neural Network (NN) models in Fig. 2, the behavior is qualitatively similar for other model types as well. Also, it is similar for both logarithm or square-root transformation of the target variable.

So far, we provided an empirical evidence of how modeling on a transformed target variable and back-transforming leads to bias. In the next subsection, we will dive deep into the mathematical basis of this transformation bias.

2.1 Mathematical basis of transformation bias

Jensen's inequality states that the convex transformation of mean is less than or equal to the mean applied after convex transformation [1]. The opposite is true of concave transformations. In context of probability theory, for a random variable X and a concave function f: $E[f(X)] <= f(E[X])$. In our case, X is *sales* and f is typically the logarithm which translates to

$$E(log(sales)) \leq log(E(sales)) \tag{2}$$

The expectation value, E, in Eq. (2) is the mean value of forecast distribution for given item on a given day.

$E(log(sales))$ is what we obtain as output from ML model built on a log transformed target variable and optimized on mean-squared error. Typically, the business is interested in $E(sales)$. To obtain $E(sales)$, we exponentiate the model output which is $E(log(sales))$. From Eq. (2), we obtain

$$exp\left(E(log(sales))\right) \leq E(sales)\ . \tag{3}$$

Eq. (3) is the mathematical demonstration of our claim that working with a transformed target variable and back-transforming leads to *Transformation Bias*. Note that even though, we worked with the special case of logarithm transformation, our conclusions hold for any concave transformation.

Note that in general transformation bias could be one of the several sources of bias (e.g., inaccurate trend, selling price etc.) in the model. The different sources of bias could add up in such a way that no net bias is observed in the back-transformed units. Nonetheless, as demonstrated in Eqs. (2) and (3), transformation bias will always be present when working with transformed target variable.

2.2 Modeling directly on sales

One of the straightforward ways to avoid transformation bias is to model directly on sales. Empirically, we find that the performance deteriorates when a model is built directly on sales (Fig. 3).

One of the reasons for the poor performance on untransformed target variable is the skewed nature of sales data as demonstrated in Fig. 4. The business reason for the right-skewed sales distribution is the spiky sales seasonality. We have days such as Black Friday, Prime Day, and Christmas Holidays when we observe significantly higher sales than most days. Moreover, some items are more popular than others adding to the skewness of distribution.

This skewed distribution makes it difficult to preserve homoscedasticity and normality of residuals. Additionally, minimizing mean squared-error (MSE) is the maximum likelihood estimator when

Region	device type	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
II	B	XGB_linear	0.38	0.94	0.42	0.70	0.67	1.14
		XGB_tree	1.12	0.92	1.58	1.00	6.58	1.82
	C	XGB_linear	-0.12	1.04	-0.11	0.94	0.25	0.48
		XGB_tree	0.62	0.69	1.00	0.70	1.83	0.50

*we use the weighted average of forecast versions from VDP_20190812 to VDP_20200302. **Target variable: sales**

Figure 3: Performance metrics when a forecasting model is built directly on sales as a target variable. XGB_linear and XGB_tree are XGBoost models using linear and tree base learners respectively. We are typically interested in how the model performs 6, 12, and 24 weeks in future. The horizon length is based on business requirements.

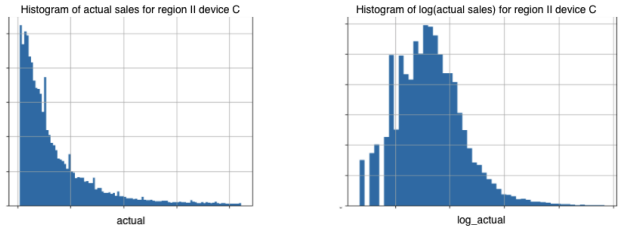

Figure 4: Histogram of daily individual item-level sales data (left) and distribution of data after log transformation (right). In the left plot, we exclude data points over 95^{th} quantile for plotability.

the noise term is Gaussian. This assumption of a Gaussian noise is typically ill-suited in presence of a highly skewed distribution (left plot in Fig. 4).

On the other hand, working with $log(sales)$ makes the distribution look more normal (right plot in Fig. 4) and makes the data more amenable to modeling. Logarithm also allows us to better treat the outliers. Note that in our case, outliers (e.g., Black Friday) are of prime business interest and can't just be discarded. Another advantage of using logarithm is that because of exponentiation during back-transformation, the prediction is always positive which makes sense for forecasting of sales.

We will return to modeling on sales again in Sec. 4, but our conclusion based on the discussion so far is that 1) Modeling directly on sales leads to poor performance 2) Modeling on $log(sales)$ has obvious advantages, but introduces a systematic transformation bias. One of the way ahead is to keep using the log transformation, but explicitly correct for the bias introduced. This will be the basis of our discussion in the next section.

3 BIAS CORRECTION TECHNIQUES

Modeling with a transformed variable was acknowledged as problematic since the 1930s [2]. Formal bias correction for logarithm was computed in 1941 [3]. In 1960, a general bias correction procedure for all transformations was developed [4]. In the next two subsections, we will look at two different methods of bias correction. The first method (Sec. 3.1) is author's original work and the second (Sec. 3.2) is an improvement over an existing method in literature [5, 6].

Region	device type	weight	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
II	C	uniform	XGB_linear	-0.46	0.90	-0.63	0.94	-0.92	0.66
		log_sales	XGB_linear	-0.23	0.65	-0.26	0.68	-0.17	0.80
		sqrt_sales	XGB_linear	0.42	0.65	0.68	0.68	1.17	0.68
		sales	XGB_linear	1.27	0.77	2.00	0.84	3.33	0.91
		uniform	XGB_tree	-0.58	0.58	-0.89	0.58	-1.25	0.43
		log_sales	XGB_tree	-0.42	0.52	-0.53	0.50	0.08	0.34
		sqrt_sales	XGB_tree	-0.38	0.54	-0.58	0.52	-0.25	0.32
		sales	XGB_tree	-0.15	0.52	-0.21	0.46	-0.08	0.20

*we use the weighted average of forecast versions from VDP_20190729 to VDP_20200302. Target variable: log(sales)

Figure 5: Effect of sales weighting on model performance across different time horizons (6, 12, and 24 weeks).

3.1 Addition of weights in cost function

The standard cost function in regression analysis is the mean-squared error.

$$J = \sum_{i \in \text{samples}} w_i (y_i - f(X_i))^2 \tag{4}$$

In the default case, $w_i = 1$ in Eq. (4). That is all the points are equally weighted. We find that making w_i in Eq. (4) a function of sales helps mitigate the negative bias. Our prescription is as follows:

(1) Start with the default uniform weight ($w_i = 1$) and successively increase the weighting, making it more aggressive (from log_sales weighting to sqrt_sales to sales and so on).
(2) With successive increases bias increases (from large negative towards zero) and MAPE (mean-absolute-percentage-error) decreases. At a certain point bias gets too large and MAPE deteriorates. Stop when the lowest MAPE and bias numbers are obtained.

The effect of successive sales weighting on model performance is demonstrated in Fig. 5. Note that given Eqs. (1), we want relative metrics to be as close to zero as possible. We find that for region II device C log_sales ($w_i = log(y_i)$ in Eq. (4)) is the best weight for linear learners and sales ($w_i = y_i$) is the best weight for tree learners. We use XGB for the demonstration here as we have seen it's one of the top-performing model for our use case. Note, that behavior is qualitatively similar regardless of the model choice though.

When we use sales weighting in cost function, we introduce a positive bias in logarithmic units. Due to Jensen's inequality, a positive bias in log units translates to less negative bias in original units. As we progressively, make the weights more aggressive, at some point, introduced bias in log units is just right enough to obtain zero bias in original units. (This is illustrated in Fig. 10.)

Addition of weights in cost function is easy to implement in most ML libraries and is seen to give improved performance in most cases. However, it suffers from the following limitations: 1) The process is iterative. We don't know which is the best weight to use apriori. 2) Performance can widely differ depending on the weight chosen 3) On a scientific front, it doesn't tackle the bias problem head on. This motivates us to explore methods that correct for transformation bias directly.

3.2 Prediction-based bias correction

In the simplest case of explicit bias transformation, the 'true' expected value of the back-transformed distribution is related to the back-transformed value obtained from model by a multiplicative

Table 1: Prediction-based BC function for region II device B

log_vdp_plus_units (or prediction)	XGB_linear	XGB_tree
$[0-2)$	1.503	1.015
$[2-4)$	1.142	1.093
$[4-6)$	1.493	1.099
$[6-8)$	2.112	1.351
≥ 8	2.527	1.687

bias correction (BC) factor.

$$E[Y] = BC * f^{-1}(E[f(Y)]) \tag{5}$$

To relate Eq. (5) to Eq. (3), note that $Y = sales$ and $f = log$. Also, based again on Eq. (3), we know that $BC \geq 1$.

According to [5, 6], the bias correction factor is related to the residuals by

$$BC \equiv e^{\epsilon} = \frac{\sum_{i=1}^{N} e^{\epsilon_i}}{N} . \tag{6}$$

On incorporating the bias correction from Eq. 6, we find that it helps mitigate the bias (by up to 50%), but doesn't eliminate it (Please refer to Fig. 11 for details). We will refine this bias correction factor next.

Fig. 2 informs us that points with higher sales need a stronger bias correction. However, the BC multiplier from Eq. (6) is independent of the sales. This motivates us to make bias correction a function of sales. Obviously, such a correction would have limited applicability, because we don't know the sales when we are making out of sample predictions. Therefore, we use model prediction, $\hat{Y}$, as a proxy for sales. Revisiting Eq. (5)

$$BC(\hat{Y}) = \frac{E[Y]}{E[f^{-1}(\widehat{f(\hat{Y})})]} \tag{7}$$

The numerator $E[Y]$ in Eq. (7) is the mean of the original sales data and the denominator in Eq. (7) is the mean of uncorrected predicted value. Instead of calculating the BC factor over whole sample (as in Eq. (6)), we evaluate the right hand side of Eq. (7) over specific prediction windows. Essentially, we learn a piece-wise linear/step function for bias correction.

We divide the prediction region in multiples of 2 (in logarithmic units) and evaluate the right hand side of Eq. (7). For the test case of region II device B, the function we learn for bias correction is shown in Table 1.

We see that the prediction based bias correction gets rid of bias in the back-transformed units. This is evident from the mean of residuals being very close to zero in the plot in right panel in Fig. 6. In applying the PB-BC technique to out-of-sample data, we make the assumption that the correction function learnt (e.g., in Table 1) carries over to the unseen data as well. There are some obvious improvements that can be done to the prediction-based correction function learnt above. For example, the step width is not uniformly spaced in the original units and the number of data points aren't evenly distributed in each prediction window.

In Fig. 7 we look at the out-of-the-sample performance in region I device B after incorporating prediction-based bias correction. Takeaways from Fig. 7 are: 1) the prediction-based bias correction immensely helps the performance. As expected the largest

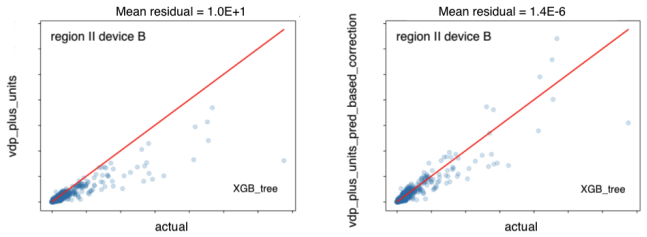

Figure 6: In-sample model fit for XGB model. The plot on the left is obtained by back-transforming the output from ML model without any bias correction. Plot on right is obtained by multiplying the model prediction by prediction-based bias correction (Table 1)

Region	device type	weight	Bias correction method	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
I	B	uniform	None	XGB_linear	-1.15	0.75	-1.47	0.72	-1.67	0.48
I	B	uniform	Prediction based	XGB_linear	-0.38	0.65	-0.21	0.62	0.67	0.30
I	B	uniform	None	XGB_tree	-1.23	0.75	-1.79	0.78	-2.33	0.64
I	B	uniform	Prediction based	XGB_tree	-0.04	0.56	-0.21	0.50	0.25	0.36

*we use the weighted average of forecast versions from VDP_20190729 to VDP_20200302. Target variable: log(sales)

Figure 7: Improvements from prediction-based bias correction for region I device type B.

improvement is in the bias metrics. 2) the benefit from bias correction transfers into MAPE improvement as well. The largest MAPE improvement of 56% (39% → 25%) is in 12W MAPE.

We acknowledge that MAPE values from PB-BC may not always perform well especially when compared to addition of weights in cost function. An example of this is show in Fig. 12. A possible reason is that the window-based approach used for calculating PB-BC may not be optimal due to data sparsity for some of the prediction windows. Checking alternative approaches of estimating PB-BC function is an ongoing effort.

4 CIRCUMVENTING TRANSFORMATION BIAS

Performance of any bias correction method will depend on how well it is able to estimate the induced transformation bias. An alternate approach to tackle this issue head-on will be to not introduce the transformation bias in the first place by working directly with actual sales as target variable.

4.1 Introduction to Tweedie Regression

We established in Sec. 2.2 that building a model on sales with MSE as a cost function led to poor performance due to skewed nature of sales data. One of the main ways in which we will generalize our regression approach will be to replace MSE by 'deviance' of a distribution in the Tweedie family [8–10].

Tweedie distribution are a family of probability distribution that include Normal, Poisson, and Gamma distributions as special cases. This is motivated in Eqs. (8). Tweedie distributions are characterized by the 'tweedie variance power', p. The mathematical formulas for 'tweedie deviance' (analogous to the cost function) are shown in

Eq. (8).

$$\bullet \text{ Poisson}: \quad 2\sum_i \left(y_i log\frac{y_i}{\hat{y}_i} - y_i + \hat{y}_i \right)$$

$$\bullet \text{ Gamma}: \quad 2\sum_i \left(-log\frac{y_i}{\hat{y}_i} + \frac{y_i - \hat{y}_i}{\hat{y}_i} \right)$$

$$\bullet \text{ Tweedie}: \quad 2\sum_i \left(y_i \frac{y_i^{1-p} - \hat{y}_i^{1-p}}{1-p} - \frac{y_i^{2-p} - \hat{y}_i^{2-p}}{2-p} \right)$$

$$\bullet \text{ Normal}: \quad \sum_i (y_i - \hat{y}_i)^2 \tag{8}$$

Its straightforward to see that:

- setting $p = 0$, recovers MSE; $p \to 1$, recovers Poisson; and $p \to 2$, recovers Gamma
- $1 < p < 2$ is referred to as Compound Poisson Gamma or just Tweedie. In rest of the discussion, we us the term Tweedie to refer to this case.

y_i and $\hat{y}_i$ in Eq. (8) are the data points and fitted values respectively. Just as while optimizing for MSE, we assume that the conditional distribution of target variable is normally distributed around the point estimate, while optimizing for Tweedie deviance, we assume that the distribution is Tweedie distributed around the point estimate output from the model.

Literature applications of Tweedie regression is typically for long-tailed positive data such as number of insurance claims/ rainfall events per year [11]. An advantage of using Tweedie for sales forecasting is that the output is always positively valued. (When using MSE, the output can be negative as well.)

4.2 Experimental set up

We will next look at results with using Tweedie cost function. The goal of these experiments is to test the effect of model design choice (specifically target transformation and cost function) on performance. To achieve this, we choose the same algorithm (XGB), feature set, and hyper-parameters.

The design choices that we evaluate are the following:

(1) Using actual sales as target variable and MSE as the cost function
(2) Using actual sales as target variable and pseudoHuber loss as the cost function. We use pseudoHuberloss as the twice differentiable alternative to mean absolute error (MAE).
  (a) A motivation for using pseudoHuberloss is that it is closely aligned with MAPE which is our model evaluation metric.
  (b) We also note that optimizing on MAE/ MAPE outputs median sales distribution for a given item for a given day. Medians aren't additive and therefore summing up daily sales estimate to obtain for example estimate for weekly sales is incorrect. This limits the practical usability of models optimized on MAE.
(3) Using the actual sales as target variable and tweedie deviance as cost function. Experiments were done using the tweedie variance power 1.1, 1.3, 1.5, 1.7, and 1.9.
(4) Using log(sales) as target variable and not doing any bias correction. We know from Sec. 2 that this is suboptimal and these results are only for reference.

Region	device type	Backcast period	target variable	cost function	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
I	E	06/22/2020 to 02/22/2021	Actual sales	MSE	1.82	1.39	2.81	1.22	4.58	1.25
			Actual sales	Huber	-0.88	1.13	-1.76	0.98	-3.08	0.84
			Actual sales	Tweedie. var_power = 1.1	0.15	0.88	-0.22	0.60	-0.34	0.17
			Actual sales	Tweedie. var_power = 1.3	0.06	0.86	-0.39	0.55	-0.56	0.18
			Actual sales	Tweedie. var_power = 1.5	0.16	0.90	-0.43	0.65	-0.98	0.27
			Actual sales	Tweedie. var_power = 1.7	-0.17	0.88	-0.87	0.68	-1.76	0.48
			Actual sales	Tweedie. var_power = 1.9	-0.40	0.86	-1.15	0.74	-2.28	0.62
			Log(sales)	MSE no bias correction	-0.70	0.88	-1.50	0.79	-2.27	0.62
			Log(sales)	MSE with sqrt_sales weights	0.07	0.94	-0.39	0.63	-0.27	0.12

Figure 8: Performance of various Tweedie models for region I device type G

(5) Using log(sales) as target variable and sqrt(sales) as weight. The weight is added to correct for transformation bias. This is the dominant setting used in production and we use this as a benchmark.

4.3 Tweedie regression results

In Fig. 8 we present results for a specific backtesting period for one of the region device type combinations. Performance for rest of the combinations is in Appendix C. Also note that in the results presented below, we report the performance of tree-based learner in the XGB algorithm as its almost always better than linear learner. While evaluating MSE and Huber cost function we set the negative predictions if any to zero.

Observations from the Tweedie regression results are as follows:

(1) The model built on actual sales with MSE as cost functions performs very poorly consistently.
(2) The model built using pseduoHuber loss performs much better than MSE, but still has high MAPE/bias values.
(3) For few of the regions (e.g, I and IV), the Tweedie regression (TR) models perform better than XGB models with bias correction (BC) — current benchmark. For many other countries, TR performance is comparable to the current benchmark. A detailed survey of TR performance can be found in Appendix C.
(4) Variance power of 1.1–1.3 gives the best results for majority of the cases.
(5) Bias values in TR are sensitive to the variance power. Specifically, as the variance power is increased the bias decreases (become more negative if it is around zero at var_power = 1.1)

4.4 Making sense of TR results

An unmistakable pattern in the Tweedie results (Fig. 8) is that model using MSE as cost function has a very high bias and hence a high MAPE. For Tweedie, we see a pattern of decreasing bias (not always a good thing because bias can have high negative values) with increase in the variance power — a hyperparameter in the Tweedie regression.

To make sense of this, we plot the deviance/ cost function (refer Eqs. (8)) for various values of variance power, p. Specifically, we set actual = 100 (an arbitrary choice) and plot the tweedie deviance as prediction varies between 10 and 190. For reference, we also plot the MSE and MAE cost functions. We see that the deviance for 1.1 is more convex than 1.5 which in turn is more convex than

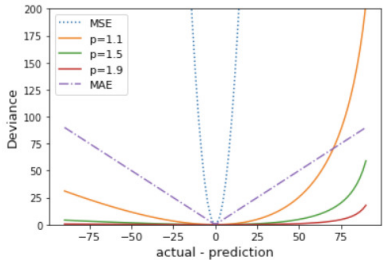

Figure 9: Tweedie deviance for variance power 1.1, 1.5, and 1.9. MSE and MAE are plotted for reference.

1.9. This implies that p=1.1 is more likely to give higher prediction values than p=1.9. Compared to these, MSE is much more convex and naturally gives much higher bias.

On an unrelated note, tweedie deviance functions are asymmetric around zero, penalizing under-prediction more than over-prediction. This might be desirable given that business cost of lost sales is higher than having an overhang.

5 CONCLUSION AND NEXT STEPS

In this work, we demonstrated that a concave transformation of target variable will introduce a negative bias in regression models which is a manifestation of Jensen's inequality. We proposed two methods to correct the transformation bias 1) addition of weights in cost function 2) direct prediction-based bias correction. These bias correction methods significantly improve both MAPE (up to 50%) and bias (up to 20X, e.g., refer Fig. 7) over different forecasting horizons.

The optimal method for bias correction is best chosen by experimentation. Sales weighting is easy to implement in most ML libraries but requires more experimentation. On the other hand, PB-BC has a more involved implementation, but involves less experimentation (once the piece-wise linear correction is determined).

We also found that leveraging Tweedie family of cost functions is a great alternative when constructing model directly on sales without any transformation. This allows us to circumvent the issue of transformation bias by avoiding target transformation altogether. We observed that tweedie variance power has a marked effect on the bias metrics and we were able to explain this based on the cost function behavior as a function of variance power.

In our work, we treated variance power as a hyper-parameter and chose it depending on the values which gave the best backtest metrics. A related question, we ask next is if we can estimate the best tweedie variance power without backtesting. This involves looking at the distribution of *deviance residuals*. Deviance residuals are generalization of residuals when using generalized cost functions (refer Eq. (8)). We expect deviance residuals to be normally distributed if the model chosen closely reflects the underlying data generation process. This forms the basis of our ongoing work.

6 SCIENCE AND BUSINESS OUTLOOK

Even though we exclusively focused on forecasting models in this paper, the issues and methods discussed are relevant anytime we have a right-skewed distribution. Log transformation is a standard practice in many regression settings and this work highlights the risk associated with it. Moreover, the methods presented in this paper such as changing the cost function to a weighted MSE or a Tweedie deviance are relatively straightforward and easy to implement in most business settings.

REFERENCES

[1] Jensen, J. L. W. V. (1906). "Sur les fonctions convexes et les inégalités entre les valeurs moyennes". Acta Mathematica. 30 (1): 175–193. doi:10.1007/BF02418571

[2] Bartlett, M.S. 1936. The square root transformation in analysis of variance. *Suppl. J. R. Stat. Soc.* **3**, 68–78

[3] Finney, D.J. 1941. On the distribution of a variate whose logarithm is normally distributed. *J. R. Stat. Soc. B* **7**, 155–161.

[4] Neyman, J. and Scott, E.L. 1960. Correction for bias introduced by a transformation of variables. *Ann. Math. Stat.* **31**, 643–655

[5] Strimbu, B M, Amarioarei A, McTague J P, Paun MM. 2018. A posteriori bias correction of three models used for environmental reporting *Forestry* 2018 **91**, 49-62

[6] Newman, M C. 1992. Regression Analysis of log-transformed data: statistical bias and its correction. *Environmental Toxicology and Chemistry* **12**, 1129-1133

[7] Zhou H, Qian W, and Yi Yang, 2019. Tweedie Gradient Boosting for Extremely Unbalanced Zero-inflated Data. arXiv.1811.10192

[8] Tweedie, M.C.K. (1984). "An index which distinguishes between some important exponential families". In Ghosh, J.K.; Roy, J (eds.). Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference. Calcutta: Indian Statistical Institute. pp. 579–604. MR 0786162

[9] Smith, C.A.B. (1997). "Obituary: Maurice Charles Kenneth Tweedie, 1919–96". Journal of the Royal Statistical Society, Series A. 160 (1): 151–154. doi:10.1111/1467-985X.00052

[10] Jørgensen, Bent (1997). The theory of dispersion models. Chapman & Hall. ISBN 978-0412997112

[11] Noll, Alexander and Salzmann, Robert and Wuthrich, Mario V., Case Study: French Motor Third-Party Liability Claims (March 4, 2020). Available at SSRN: https://ssrn.com/abstract=3164764 or http://dx.doi.org/10.2139/ssrn.3164764

A VDP MODEL METRICS

Performance of various VDP models built on transformed target variable is shown in Fig. 1. The performance is measured across different horizons (6 week, 12 week, and 24 week). We generate a new forecast every week labeled by VDP_YYYYMMDD. The MAPE (mean absolute percentage error) and bias metrics are generated for each forecast version. The reported metrics for a given model are sales-weighted average over multiple forecast versions. The details of the WMAPE (weighted MAPE) and wbias (weighted bias) calculation are as follows:

$F_{i,V,d}$: Forecasted units for item i for forecast version V on day d

$A_{i,V,d}$: Actual units for item i for forecast version V on day d

$A_{i,V}$: Actual units for item i for forecast version V across time horizon h : set of all days in time horizon of metric

$$PE_{i,V} = \frac{\sum_{i \in h} F_{i,V,d} - \sum_{i \in h} A_{i,V,d}}{\sum_{i \in h} A_{i,V,d}}$$

$$WMAPE_V = \frac{\sum_i A_{i,V} * |PE_{i,V}|}{\sum_i A_{i,V}} \quad WBias_V = \frac{\sum_i A_{i,V} * PE_{i,V}}{\sum_i A_{i,V}} \tag{9}$$

B BIAS CORRECTION DETAILS

B.1 Sales weighting effect

When we use sales weighting in cost function, we introduce a positive bias in logarithmic units. Due to Jensen's inequality, a positive bias in log units translates to less negative bias in original units. As we progressively, make the weights more aggressive, at some point,

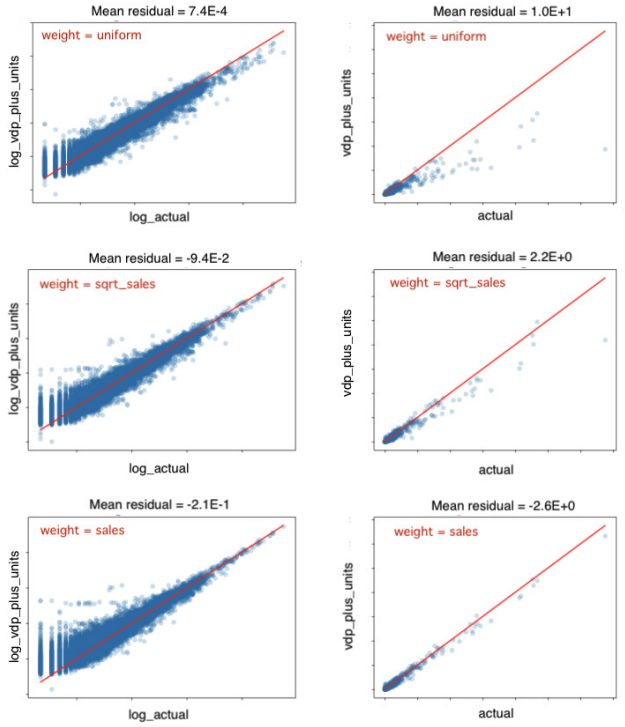

Figure 10: In-sample model fit for region II device B for XGB_tree using different weights in the cost function. The plots on the left are the fit in the modeling units (logarithm in this case). The Y axis is the prediction output from the ML model. The plots on the right are obtained by back-transforming (exponentiating in this case) the values from plots on the left. In particular, we don't do any model refit while going from plot on left to the plots on right.

introduced bias in log units is just right enough to obtain zero bias in original units. This is illustrated in Fig. 10. Note that residual = actual − prediction. So, negative bias ⇒ mean(residual) > 0

B.2 Direct bias correction

According to [5, 6], the bias correction factor is related to the variance of residuals as

$$BC = e^{\sigma^2/2} . \tag{10}$$

Eq. (10) holds when the residuals are normally distributed. If residuals are not normally distributed, 'a smearing estimate of bias' is recommended:

$$BC \equiv e^{\epsilon} = \frac{\sum_{i=1}^{N} e^{\epsilon_i}}{N} \tag{11}$$

where ϵ_i is the i^{th} regression residual. We calculate BC multiplier from Eqs. (10) and (11) for test case of region II device B. The multipliers obtained are shown in Table 2.

BC multipliers from Eq. (10) and (11) are similar as shown in Table 2 because for these models the residuals are normally distributed to a good approximation.

In Fig. 11, we plot the raw uncorrected back-transformed values on left (these are similar to plots on right in Fig. 2). The plots on right in Fig. 11 are obtained by correcting for bias transformation using Eq. (11). Specifically, we multiply the model predictions by second row in Table 2. Few observations, we make from Fig. 11:

Table 2: Bias correction multipliers for region II device B

	XGB_linear	XGB_tree
Variance-based multiplier [Eq. (10)]	1.29	1.07
Mean-based multiplier [Eq. (11)]	1.27	1.07

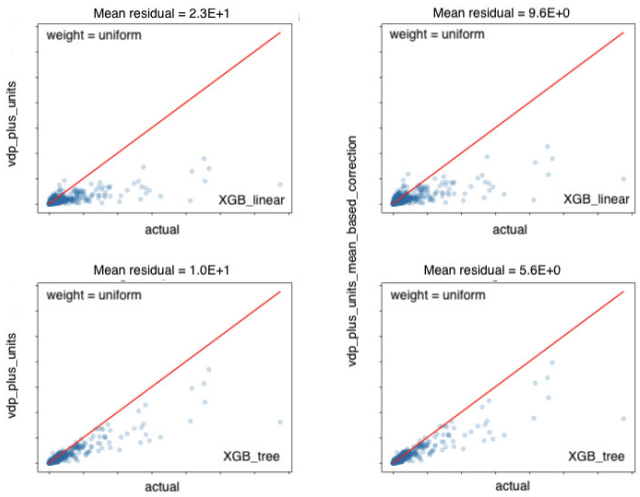

Figure 11: In-sample model fit for region II device B for XGB model. The plots on the left are obtained by back-transforming the output from ML model without any bias correction. Plots on right are obtained by multiplying the model prediction by mean-based bias correction (Eq. (11)).

(1) The negative bias definitely decreases after correction. This can be seen from the mean of residuals moving closer to zero. In fact, for both XGB_tree and XGB_linear, the bias decreased by about 50%.
(2) Nonetheless, we are still far from completely removing bias. For instance, the mean of residuals in log units (left panel in Fig. 2) is much closer to zero than what we have in the original units even after bias correction
(3) We notice that points with higher sales show a more prominent bias than lower sales. This motivated us to derive a 'prediction-based' bias correction.

B.3 Bias correction derivation

Here we derive the BC factor in Eq. (10). Following the approach in [6], in case of linear regression we have:

$$log(Y) = b_0 + b_1 X + \epsilon \tag{12}$$

ϵ has a mean 0 in logarithmic units but not in the original arithmetic units. Because the mean will not be zero after back-transformation, the error term must be retained during back transformation:

$$Y = b_0 e^{b_1 X} e^{\epsilon} \tag{13}$$

Region	device type	Backcast period	target variable	cost function	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
II	D	06/22/2020 to 02/22/2021	Actual sales	Huber	-1.11	0.97	-2.25	1.04	-4.10	1.12
			Actual sales	Tweedie. var_power = 1.1	0.25	0.74	0.04	0.71	-1.11	0.38
			Actual sales	Tweedie. var_power = 1.3	0.15	0.70	-0.17	0.68	-1.20	0.36
			Actual sales	Tweedie. var_power = 1.5	-0.08	0.70	-0.40	0.70	-1.52	0.41
			Actual sales	Tweedie. var_power = 1.7	-0.20	0.71	-0.71	0.72	-1.90	0.52
			Actual sales	Tweedie. var_power = 1.9	-0.51	0.70	-1.17	0.74	-2.48	0.68
			Log(sales)	MSE no bias correction	-0.54	0.61	-1.11	0.61	-2.31	0.63
			Log(sales)	MSE with sqrt_sales weights	-0.21	0.63	-0.60	0.60	-1.72	0.47

Figure 15: Performance of various Tweedie models for region II device type D

Region	device type	Backcast period	target variable	cost function	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
II	E	06/22/2020 to 02/22/2021	Actual sales	Huber	-0.46	0.76	-1.05	0.82	-1.75	0.48
			Actual sales	Tweedie. var_power = 1.1	0.49	0.86	0.41	0.75	0.33	0.25
			Actual sales	Tweedie. var_power = 1.3	0.4	0.86	0.28	0.75	-0.03	0.3
			Actual sales	Tweedie. var_power = 1.5	0.35	0.86	0.26	0.73	0.18	0.25
			Actual sales	Tweedie. var_power = 1.7	0.25	0.82	0.01	0.73	-0.17	0.29
			Actual sales	Tweedie. var_power = 1.9	0.12	0.81	-0.19	0.74	-0.52	0.32
			Log(sales)	MSE no bias correction	-0.11	0.81	-0.46	0.73	-0.87	0.33
			Log(sales)	MSE with sqrt_sales weights	0.16	0.76	-0.04	0.68	-0.36	0.3

Figure 16: Performance of various Tweedie models for region II device type E

Region	device type	Backcast period	target variable	cost function	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
III	D	06/22/2020 to 02/22/2021	Actual sales	Huber	-0.03	1.27	-0.84	1.34	-2.09	0.57
			Actual sales	Tweedie. var_power = 1.1	0.33	0.84	0.08	0.86	-1.18	0.43
			Actual sales	Tweedie. var_power = 1.3	0.22	0.92	-0.01	1.07	-1.28	0.44
			Actual sales	Tweedie. var_power = 1.5	-0.01	0.91	-0.46	1.04	-1.79	0.49
			Actual sales	Tweedie. var_power = 1.7	-0.15	0.91	-0.74	1.06	-2.43	0.66
			Actual sales	Tweedie. var_power = 1.9	-0.39	0.9	-1.23	1.1	-3.17	0.86
			Log(sales)	MSE no bias correction	-1.08	0.78	-1.93	0.92	-3.8	1.04
			Log(sales)	MSE with sqrt_sales weights	-0.19	0.75	-0.8	0.8	-2.42	0.66

Figure 17: Performance of various Tweedie models for region III device type D

Region	device type	Backcast period	target variable	cost function	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
IV	G	06/22/2020 to 02/22/2021	Actual sales	Tweedie. var_power = 1.1	-0.46	0.98	-1.12	0.86	-2.01	0.60
			Actual sales	Tweedie. var_power = 1.3	-0.71	1.00	-1.40	0.89	-2.09	0.70
			Actual sales	Tweedie. var_power = 1.5	-0.74	1.02	-1.42	0.92	-2.24	0.60
			Actual sales	Tweedie. var_power = 1.7	-0.87	1.05	-1.71	0.96	-2.75	0.80
			Actual sales	Tweedie. var_power = 1.9	-1.04	1.07	-1.94	1.04	-3.03	0.80
			Log(sales)	MSE with sqrt_sales weights	-0.73	1.00	-1.44	0.92	-2.35	0.70

Figure 18: Performance of various Tweedie models for region IV device type G

Region	device_type	weight	Bias correction method	model	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
II	B	uniform	None	XGB_linear	-0.50	0.58	-0.53	0.48	1.08	0.39
II	B	uniform	Prediction based	XGB_linear	0.69	0.65	1.42	0.72	4.83	1.39
II	B	log_sales	Weight based	XGB_linear	-0.19	0.54	-0.16	0.40	2.42	0.68
II	B	uniform	None	XGB_tree	-0.46	0.58	-0.84	0.58	-0.17	0.39
II	B	uniform	Prediction based	XGB_tree	0.00	0.60	-0.21	0.52	1.00	0.45

*we use the weighted average of forecast versions from VDP_20190729 to VDP_20200302. Target variable: log(sales)

Figure 12: Improvements from prediction-based bias correction for region II device type B

Region	device type	Backcast period	target variable	cost function	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
I	D	06/22/2020 to 02/22/2021	Actual sales	MSE	3.9	2.2	5.8	2.2	8.96	2.44
			Actual sales	Huber	-0.3	1.1	-1.4	1.0	-3.19	0.87
			Actual sales	Tweedie. var_power = 1.1	0.2	0.9	-0.4	0.58	-1.4	0.39
			Actual sales	Tweedie. var_power = 1.3	0.2	0.9	-0.4	0.55	-1.02	0.28
			Actual sales	Tweedie. var_power = 1.5	0.2	0.9	-0.5	0.57	-1.19	0.32
			Actual sales	Tweedie. var_power = 1.7	0.1	0.9	-0.6	0.61	-1.47	0.4
			Actual sales	Tweedie. var_power = 1.9	0.0	0.9	-0.7	0.63	-1.76	0.48
			Log(sales)	MSE no bias correction	-0.4	1.0	-1.3	0.84	-2.37	0.65
			Log(sales)	MSE with sqrt_sales weights	0.1	0.9	-0.5	0.55	-1.02	0.28

Figure 13: Performance of various Tweedie models for region I device type B

Region	device type	Backcast period	target variable	cost function	wbias6W	wmape6W	wbias12W	wmape12W	wbias24W	wmape24W
I	F	06/22/2020 to 02/22/2021	Actual sales	MSE	0.23	1.55	1.28	0.98	4.02	1.10
			Actual sales	Huber	-0.38	1.18	-0.99	0.90	-1.96	0.53
			Actual sales	Tweedie. var_power = 1.1	-0.04	1.10	-0.42	0.61	-1.28	0.35
			Actual sales	Tweedie. var_power = 1.3	-0.14	1.10	-0.64	0.61	-1.42	0.39
			Actual sales	Tweedie. var_power = 1.5	-0.07	1.10	-0.61	0.63	-1.53	0.42
			Actual sales	Tweedie. var_power = 1.7	-0.07	1.11	-0.58	0.67	-1.46	0.40
			Actual sales	Tweedie. var_power = 1.9	-0.11	1.08	-0.68	0.67	-1.83	0.50
			Log(sales)	MSE no bias correction	-1.08	1.13	-2.01	0.93	-3.71	1.01
			Log(sales)	MSE with sqrt_sales weights	-0.30	1.09	-0.93	0.69	-2.04	0.56

Figure 14: Performance of various Tweedie models for region I device type F

e^{ϵ} is the bias correction factor (Eq. (11)). If we assume that ϵ follows a normal distribution with mean 0 and variance σ^2,

$$E[e^{\epsilon}] = \int_{-\infty}^{\infty} e^{\epsilon} \mathcal{N}(0; \sigma^2)\, d\epsilon = e^{\sigma^2/2} \tag{14}$$

Eq. (14) is the bias correction factor we had in Eq. (10).

Another point to note is that

$$E[e^{\epsilon}] \geq e^{E(\epsilon)} \tag{15}$$

$$\Rightarrow E[e^{\epsilon}] \geq 1\,. \tag{16}$$

In Eq. (16), we use the property that $E(\epsilon) = 0$ or that mean of residuals is zero in the modeling units. Note that we don't make any assumptions on the distribution of error term, ϵ, in the derivation of Eq. (16).

Equation (16) tells us that the Bias correction factor will always be greater than equal to 1. Implying that our back-transformed forecasts will always under-forecast on average.

C BIAS CORRECTION AND TWEEDIE RESULTS

We provide a survey of Tweedie regression performance in Fig. 13 through Fig. 18. The takeaways from these results are presented in Sec. 4.3.