Cubic Ridge Regression for Scaling Machine Learning Credit Risk Scores

Marick S. Sinay, PhD, Peng Jiang, PhD, Andrew Jennings, PhD

Abstract

Machine learning (ML) models for bespoke credit risk analytics is well trodden ground. Both FinTech's and larger financial institutions use ML methods for in-house credit risk models. Model development initiatives vary by institution. Smaller teams typically adopt open-source software. Whereas larger institutions often leverage licensed software that can be more "point and click". However, most ML binary classification algorithms provide a score value from zero to one with high decimal precision. Scaling of this raw score to match the score range and odds of an existing score, or a bureau score, is usually a business requirement. Oftentimes, this scaling process entails rounding to the nearest integer value to generate a user-friendly score. Whether writing custom code to scale or a licensed solution, one common way is a two-step linear transformation from unscaled scores, first to implied log odds, then finally to scaled values. However, the literature on scaling of credit scores is rather sparse, especially for high decimal precision ML analytics. To that end, the authors outline a novel approach to better scale a bespoke score to match the range and odds of an existing score. This involves leveraging a third order polynomial Ridge regression. We combine this technique with a known, albeit less documented, approach to match both the odds at a base score value and the Points to Double the Odds (PDO). On an empirical data set the authors observe at least four main benefits to this novel scaling framework. First, we can achieve greater risk differentiation in terms of unique score values, even after rounding the scaled score to the nearest integer. On our data set our cubic fit garners a roughly 53% increase in unique score values over the traditional linear method. This is an obvious benefit to business end users for setting credit strategies and policies given the increase in fidelity in risk rank ordering and risk differentiation. Second, the scaled score is linear in the log odds space, whereas the linear transformed scaled score demonstrates non-linearity in the log odds space. Third, the odds at a

Preprint submitted to Credit Scoring and Credit Control Conference XIX July 21, 2025

base score value are better matched. Fourth, the points to double the odds are preserved better under the third order polynomial transformation. We will outline in detail our novel approach and share empirical results on a proprietary data set.

Keywords:
Machine learning credit score, scaling, XGBoost, Ridge regression, Cubic splines

---

1. Introduction

Machine learning (ML) techniques for binary classification have been widely used in the consumer lending space to develop bespoke advanced predictive analytics to assess the credit worthiness of a consumer when applying for various different credit products for many years now. Typical binary target variables include first payment default, ever charged off, ever 60+ days past due over the first "n" months on book, to name just a few. Smaller teams typically leverage open source software, such as Python and all the various libraries for data science. This comes at the benefit of almost complete flexibility on choices of feature engineering, model techniques, hyper-parameter tuning, etc.... Larger institutions often use more "point and click" software to develop credit scorecards, that are sometimes informed by ML techniques for feature selection and discretization of continuous variables.

One typical aspect of such model development projects is the scaling of a "raw" score value to a more traditional credit score range. Furthermore, it is oftentimes the case that business end users, such as credit strategy teams, require that the new scaled score roughly match the odds of either an existing in-house score or the odds of bureau score, at least over some relevant range of score values. This requires model development teams to use an appropriate transformation from the raw score range to the desired scaled score range, while preserving rank order of course, and matching the odds.

Various different approaches have been adopted for scaling raw score values to a preexisting score range, while attempting to match the good to bad odds over at least some relevant range of score values. However, one potential shortcoming of such score scaling processes is range compression. This is when the underlying original raw score value has greater granularity, and thus greater risk differentiation power, than the resulting scaled score. Additionally, matching the odds of an existing score can sometimes prove to be

challenging.

We will demonstrate through use of a cubic transformation that we can alleviate the range compression and better match the odds of an existing credit bureau score relative to the commonly adopted linear transformation approach that is often used. In this way, this work is novel and practitioners may want to explore the cubic transformation as an option when conducting their score scaling process.

2. Description of the Data

The data used here is a proprietary data asset based on 136,683 total number of loans. Within this 117,022 records were used to train a bespoke machine learning (ML) binary classification predictive model. Additionally, 19,895 records were not utilized for model training and will be used in the score scaling process.

The definition of the target variable used was the following.

- **Positive Class:** At the time of model development, all positive labeled loans must have had 18+ months from loan origination, i.e. 18+ months on book (MOB), and never went more that 30 days past due (DPD) at any point over the life of the loan.
- **Negative Class:** Negative labeled loans either went 60+ days past due over the first 18 months on book OR were charged off at any point over the life of the loan.

The features used for model development were a wide set of credit attributes from the credit bureau. Given this high level description of the data set we move our attention to some summary of analysis of two credit bureau scores that were included on the data set.

3. Industry Standard Credit Bureau Scores

Included in the data set were two industry standard credit bureau scores. We will refer to these credit bureau scores as Credit Bureau Score 1 and Credit Bureau Score 2. In Figures 1 and 2 below we present histograms of the two credit bureau scores, bifurcated by good and bad labeled loans. By viewing the data in this way we can get a sense of how well the credit bureau scores separate good from bad loans. By no means is the only way

to understand the risk differentiation power of these two bureau scores, but it is one reasonable way to get a visual sense for the two scores' separation power.

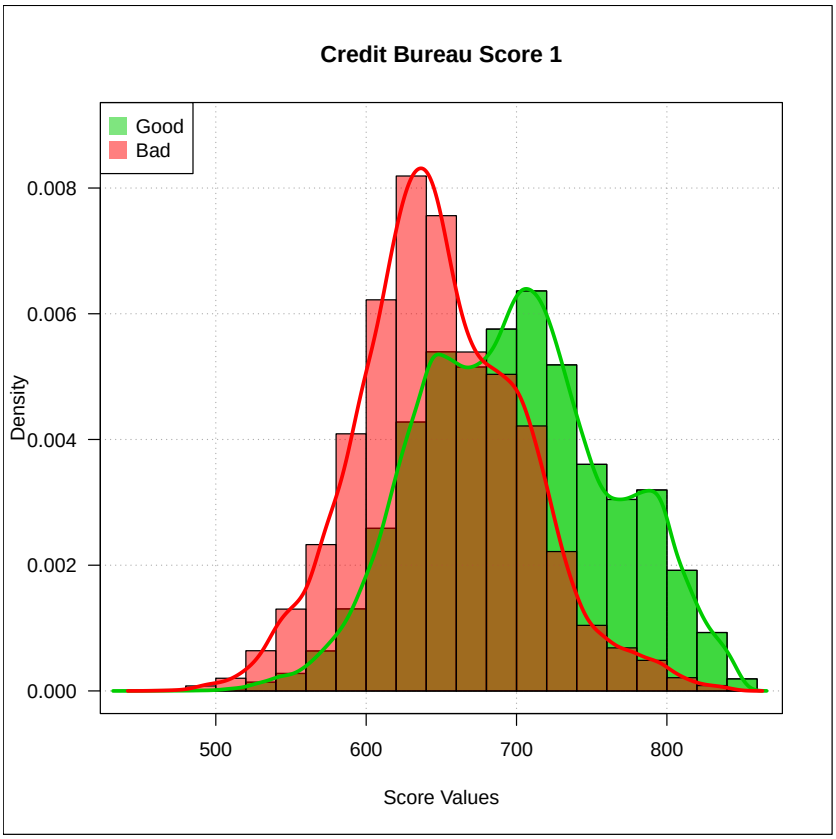

Figure 1: Bifurcated histogram for credit bureau score 1.

As mentioned above it is very often the case that business end users will require a bespoke scorecard or bespoke ML model score be scaled to match some existing score that is already in use. Additionally, it is common that business end users will also want a new score to match the odds of an existing score, as best as possible. By matching the odds we can minimize the business disruption and strategy impacts from a new score being introduced. This also tends to help with interpretation and strategy setting. In the next section we will turn attention to the bespoke ML score that was developed from this data set.

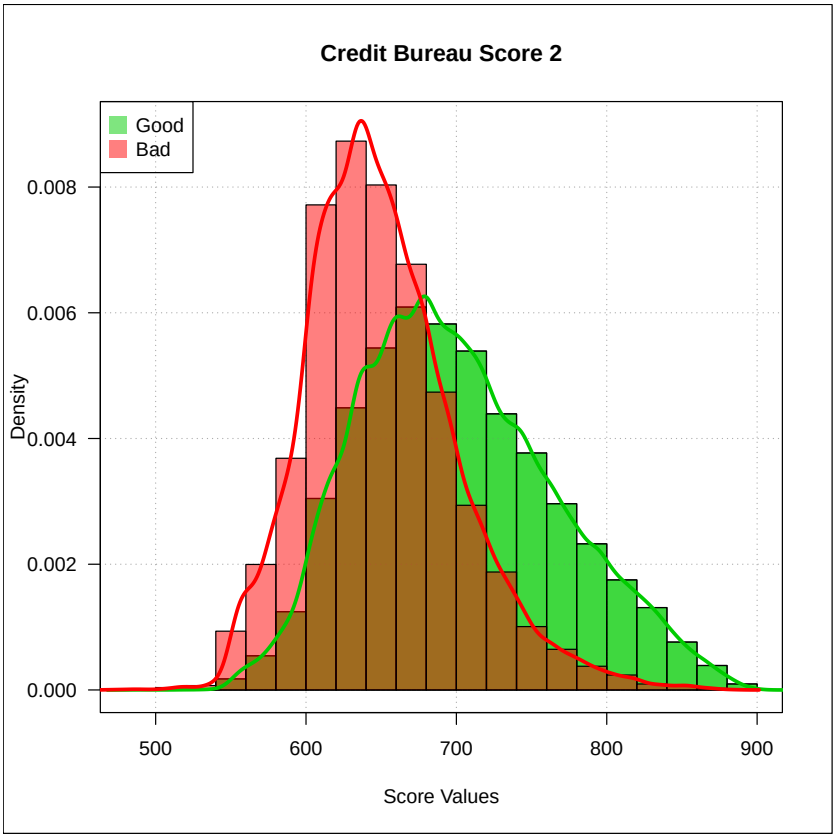

Figure 2: Bifurcated histogram for credit bureau score 2.

4. Unscaled Machine Learning Score

We trained a machine learning (ML) binary classification model on the target variable defined above. In this particular case, we used XGBoost, which is a very popular ML technique that is widely used. The raw unscaled output from the ML model, i.e. the fitted values, are on the interval from zero to one. Because we defined our good loan outcome to correspond to a value of 1 in our training data set, and a bad loan outcome to correspond to a value of 0, fitted values from the ML model closer to zero are indicative of higher risk. Conversely, records with score values closer to 1 are lower risk.

In Figure 3 we present the bifurcated histogram for the raw score on the test data set. We see a significant separation and this will ultimately motivate the cubic transformation we detail below.

Having provided some data visualization of the credit bureau scores and

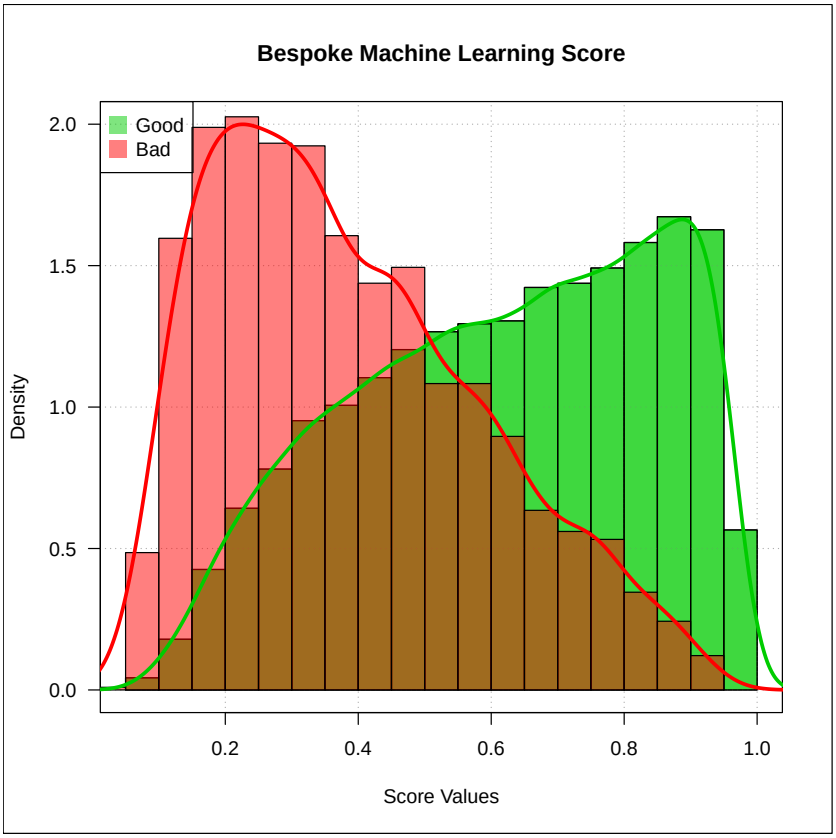

Figure 3: Bifurcated histogram for unscaled machine learning score.

the underlying raw unscaled ML fitted values, we turn now to the scaling process itself.

5. Scaling Process

In this section will outline the following key steps for the scaling process and then demonstrate how it is applied to our specific ML score and data set. At a high level, the scaling process begins with calculating intervals on the unscaled score range. From these intervals we can then calculate the observed log odds rate within each score interval as well as the mean score value within the interval. From these 2 vectors of data we can then fit various different functional relationships between the mean values and the observed log odds. In particular, we will explore a linear and cubic functional form.

From either the linear or cubic fit we can estimate the statistical relationship, a mapping if you will, between the unscaled scores and the *fitted* log odds. The classic choice for this functional relationship is linear, however, we will demonstrate that a cubic relationship performs better in many respects. The precise details of the scaling steps are outlined directly below.

1. Starting from the roughly 20k test set records, we calculate 20 score intervals, or *bins*, based on the percentiles of the unscaled ML score. Test records are used here to avoid any potential skewness or bias in using fitted score values from records used in model training. In our case, this results in roughly 1000 records in each of the 20 bins. In practice, each practitioner should decide on an appropriate number of bins in order to have enough sample size in each bin to reliable estimate the good to bad rate in each bin. Five, ten, twenty or twenty five are common choices for the number of bins. In practice, if the data supports more bins, this tends to be better. These score bins will as serve the backbone of the remaining steps of the scaling process.
2. Step 2 is to calculate the good to bad log odds within each of the 20 bins. This will serve as the target variable in Step 4 below.
3. Step 3 is to calculate the mean score value within each of the 20 bins. This will serve as the predictor variable in Step 4 below.
4. Step 4 has 2 variants, which we will elaborate upon.
   (a) The traditional approach is to fit a linear ordinary least square (OLS) model using the mean score value as the predictor and the log odds as the response variable. In our case we have 20 observations from which to fit this OLS, and we will denote the resulting fitted OLS model as follows:

$$\hat{y}_1(x) = \hat{\beta}_0 + \hat{\beta}_1 x \quad \forall\, x \in [0,1]$$

   (b) We will show that for ML models, it may very well be the case that a cubic transformation from unscaled scores to the log odds space, more accurately fits the data and provides optimal scaling properties to be detailed below.

$$\hat{y}_2(x) = \hat{\alpha}_0 + \hat{\alpha}_1 x + \hat{\alpha}_2 x^2 + \hat{\alpha}_3 x^3 \quad \forall\, x \in [0,1]$$

5. Step 5 is to then use the fitted transformation, from 4.a and 4.b, to map raw unscaled ML score values to the fitted log odds space.

6. Finally, Step 6 is to use the **Fundamental Scaling Equation** to map fitted log odds values to the scaled score space.

$$S'(x) = S_b + \frac{PDO}{\log(2)} \left[\hat{y}_i(x) - \log(O_b)\right] \quad \forall\, x \in [0,1] \text{ and } i = 1,2 \quad (1)$$

In Equation 1 we define the following terms.

- $S'(x)$ is the final scaled score value, written as a function of the unscaled score value.
- x is the raw unscaled score value from the ML model output.
- S_b denotes a *base* score value, which is entirely chosen by the practitioner. We will provide more details on this below.
- PDO denotes the Points to Double the Odds, which is also chosen by the practitioner, but is usually informed based on the observed data available. We will provide more details on this below.
- $\hat{y}_i(x)$ is the fitted log odds value from Step 4 above. Because we are comparing two different models in steps 4.a and 4.b, we will make it clear which one we are using where. More details to follow on this below.
- O_b is the good to bad odds ratio in a relatively small interval around the base score S_b.

In Table 1 we highlight the key relationships that are imposed on the resulting scaled score, which are by design and a result of utilizing the Fundamental Scaling Equation.

Table 1: Implied odds of the fundamental scaling equation.

Unscaled Score: x	**Fitted Log Odds: $\hat{y}_i(x)$**	**Scaled Score: $S'(x)$**
x_0	$\log\left(\frac{O_b}{2}\right) = \log(O_b) - \log(2)$	$S_b - PDO$
x_1	$\log(O_b)$	S_b
x_2	$\log(2O_b) = \log(O_b) + \log(2)$	$S_b + PDO$

For any continuous real valued selection for $\hat{y}(x)$ and reasonable choices of S_b, O_b and PDO, there exists values x_0, x_1 and x_2 on the unscaled score

range such that the fitted odds will be equal to the corresponding values in Table 1. For example, there exists an unscaled score value of x_1 such that the fitted log odds are $\hat{y}(x_1) = \log(O_b)$, which means the associated fitted odds are in fact O_b and the scaled score value will be in fact S_b, which is exactly what is desired. Similarly, there exists an unscaled score value of x_2 such that the fitted log odds are $\hat{y}(x_2) = \log(2O_b)$, which means the associated fitted odds are $2O_b$ and the the scaled score is $S_b + PDO$, which again is exactly what is desired, meaning that at a score value of $S_b + PDO$ the fitted odds are double the odds at S_b.. Analogous comments can be made regarding x_0. In this way, the Fundamental Scaling Equation necessarily imposes the desired odds matching by design.

Note that throughout it should be understood that we are using the natural logarithm with base e. Furthermore, note that Equation 1 above imposes a linear relationship between scaled score values and fitted log odd values, but not necessarily a linear relationship between scaled score values and raw unscaled score values, which involves the functional form of $\hat{y}_i(x)$. Obviously, depending on the functional form of $\hat{y}_i(x)$, the relationship between unscaled fitted values from the ML model to the scaled score values, may or may not be linear. It is important however, the estimated function $\hat{y}_i(x)$ be a real valued continuous strictly monotonically increasing function in order to preserve the risk rank order of the original unscaled ML fitted values.

Typically, scaled score values are rounded to the nearest integer in order to match the traditional score values of standard credit bureau scores. This also has an impact on the risk differentiation power of a high decimal precision unscaled score. The cubic fit can help to alleviate this by yielding more unique integer value risk scores. We will highlight this fact in the next section below.

5.1. Linear and Cubic Fitted Models

In Tables 2 and 3 we present the estimated regression coefficients, the standard errors, the t values and the associated p values for the OLS and cubic fitted models.

Table 2: OLS estimated model.

	Estimate	Std. Error	t value	Pr(>|t|)
$\hat{\beta}_0$	-0.5481	0.2806	-1.95	0.0665
$\hat{\beta}_1$	5.3866	0.4393	12.26	0.0000

For the OLS model the estimated slope coefficient is statistically significant at the 0.000 level of confidence.

Table 3: Cubic estimated model.

	Estimate	Std. Error	t value	Pr(>|t|)
$\hat{\alpha}_0$	-2.2611	0.7224	-3.13	0.0065
$\hat{\alpha}_1$	21.2892	4.7246	4.51	0.0004
$\hat{\alpha}_2$	-37.8596	9.1962	-4.12	0.0008
$\hat{\alpha}_3$	25.2955	5.4497	4.64	0.0003

For the cubic fitted model all the estimated regression coefficients are statistically significant at the 0.007 level or lower. In the next section we will assess these relationships more.

5.2. Log Odds and ML Relationship

In Figure 4 we present the observed log odds and the mean unscaled score scatter plot for the chosen 20 bins, along with the OLS best fit line and the cubic fit. We can see that the cubic fit captures the relationship between the log odds rate and the mean score values much better than the OLS fit. In point of fact, the adjusted R squared value of the OLS fit is 88.72%, versus 96.09% for the cubic fit.

Furthermore, in terms of the fitted log odds space on the vertical axis, the OLS line results in much more range compression, relative to the cubic fit. The greater range in the cubic fit is not only more reflective of the true underlying pattern in the data, but also the cubic fit will also result in greater range in the final scaled score values. This is optimal for credit underwriting decisions where greater risk differentiation is preferred.

In fact, for raw unscaled fitted values on the zero to one interval, we can analytically calculate the possible minimum and maximum scaled score

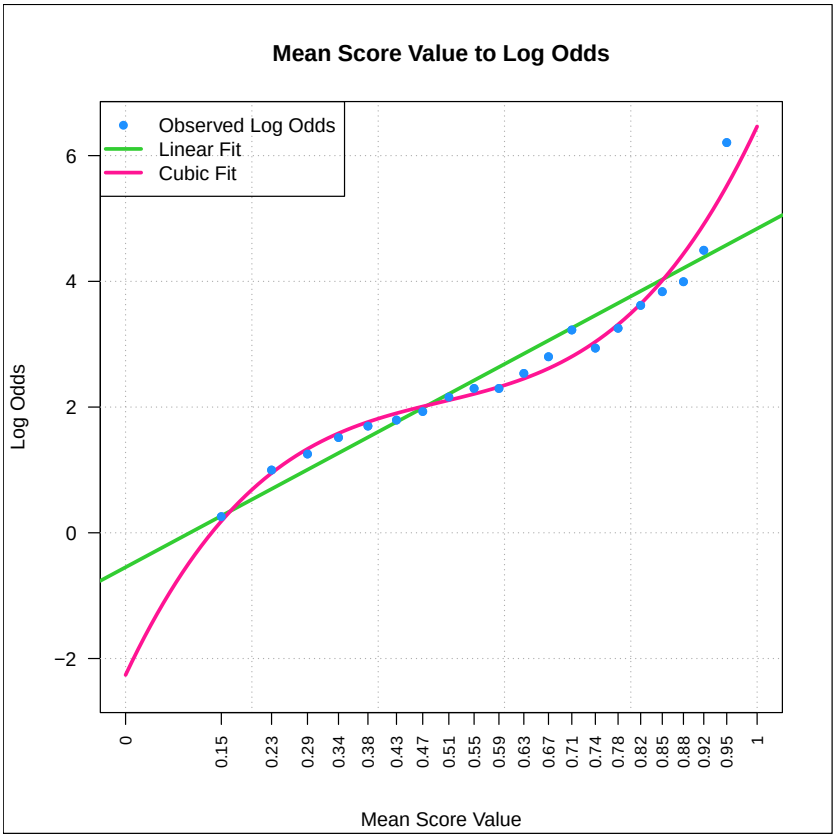

Figure 4: Log odds and machine learning score.

values. For the linear transformation we have the following.

$$\begin{aligned} S'(0) &= S_b + \frac{PDO}{\log(2)}\left[\hat{y}_1(0) - \log(O_b)\right] \\ &= S_b + \frac{PDO}{\log(2)}\left[\hat{\beta}_0 - \log(O_b)\right] \\ S'(1) &= S_b + \frac{PDO}{\log(2)}\left[\hat{y}_1(1) - \log(O_b)\right] \\ &= S_b + \frac{PDO}{\log(2)}\left[\hat{\beta}_0 + \hat{\beta}_1 - \log(O_b)\right] \end{aligned}$$

For the cubic transformation we have the following.

$$\begin{aligned} S'(0) &= S_b + \frac{PDO}{\log(2)}\left[\hat{y}_2(0) - \log(O_b)\right] \\ &= S_b + \frac{PDO}{\log(2)}\left[\hat{\alpha}_0 - \log(O_b)\right] \\ S'(1) &= S_b + \frac{PDO}{\log(2)}\left[\hat{y}_2(1) - \log(O_b)\right] \\ &= S_b + \frac{PDO}{\log(2)}\left[\hat{\alpha}_0 + \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3 - \log(O_b)\right] \end{aligned}$$

This leads to the conditions that if $\hat{\beta}_0 > \hat{\alpha}_0$ *and* $\hat{\beta}_0 + \hat{\beta}_1 < \hat{\alpha}_0 + \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3$, then we are guaranteed less range compression from the cubic transformation relative to the linear transformation. In looking back at Tables 2 and 3 we can see that both of these conditions are satisfied.

$$\begin{aligned} \hat{\beta}_0 &= -0.5481 > -2.2611 = \hat{\alpha}_0 \\ \hat{\beta}_0 + \hat{\beta}_1 &= 4.8384 < 6.4639 = \hat{\alpha}_0 + \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3 \end{aligned}$$

Therefore, we are guaranteed less range compression, and thus more unique scaled score values, from the cubic transformation relative to the linear transformation. We turn now to explore the relationship between the credit bureau scores and the log odds.

5.3. Log Odds and Credit Bureau Score Relationships

For the purposes of comparison, let us examine the log odds to mean score value for both of the credit bureau scores. Notice in Figures 5 and 6 that the observed log odds to mean score value relationship is inherently much more linear. In fact, the cubic fit which is included in the figures is nearly linear over the relevant range of score values. This again illustrates how the bespoke ML score can provide much greater risk differentiation.

5.4. Scaling Parameter Selections

For our case, we opted to use Credit Bureau Score 1 to select the three scaling parameters.

1. We chose to center our base score value at $S_b \approx 650$.
2. In the score interval $(645, 654]$, which is a close neighborhood of our base score value, the observed good to bad odds are $O_b \approx 4.7$.

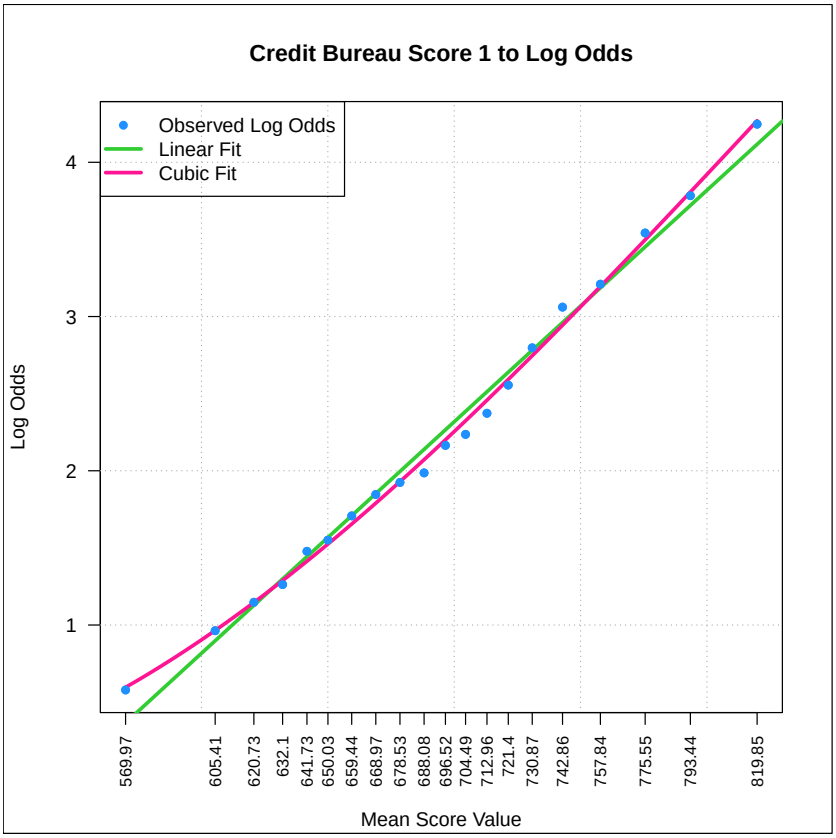

Figure 5: Log odds and Credit Bureau Score 1.

3. The selected $PDO \approx 30$.

Based on these selected values, in the next section we will provide a comparison between the resulting OLS and cubic scaled scores.

5.5. Empirical Results Comparing OLS to Cubic Scaling

With these three scaling values selected, along with the fitted OLS and fitted cubic function, we can now use Equation 1 to calculate two competing scaled scores to compare and contrast how the scaling from the OLS versus the fitted cubic function perform. In Table 4 we can observe the key summary statistics for the two competing scaled scores.

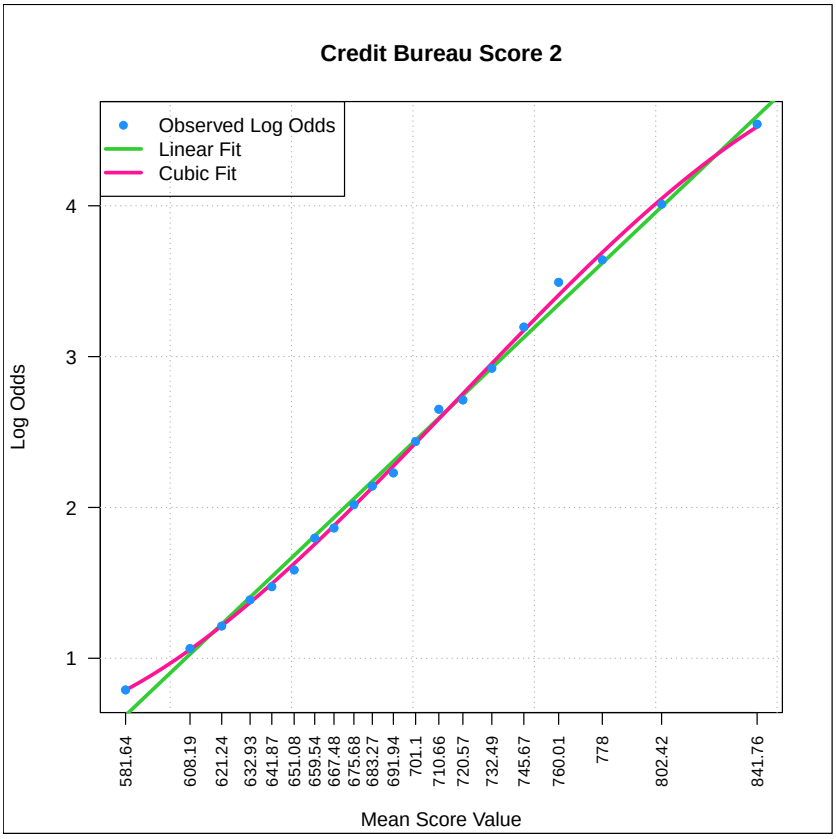

Figure 6: Log odds and Credit Bureau Score 2.

Table 4: OLS and Cubic scaled score summary statistics.

	Min	**1st Qrt**	**Median**	**Mean**	**3rd Qrt**	**Max**	**# Unique**
OLS	564	618	666	671	723	790	227
Cubic	503	630	669	669	704	850	347

The score range for the OLS transformation is from 564 to 790, with a total of 227 unique values. Compare this with the cubic transformation with a score rage from 503 to 850, for a total of 347 unique values. Thus the cubic transformation results in roughly 53% more unique score values, which results in greater risk differentiation. This has substantial positive benefits for credit strategies to provide more granular threshold selection for risk tiering, which can contribute to risk based pricing decisions.

In Figures 7 and 8 we present the histograms of the OLS and cubic

transformation scaled scores. Note that while not exactly bell shaped, the cubic transformation scaled score more closely resembles that of a Normal bell shape distribution.

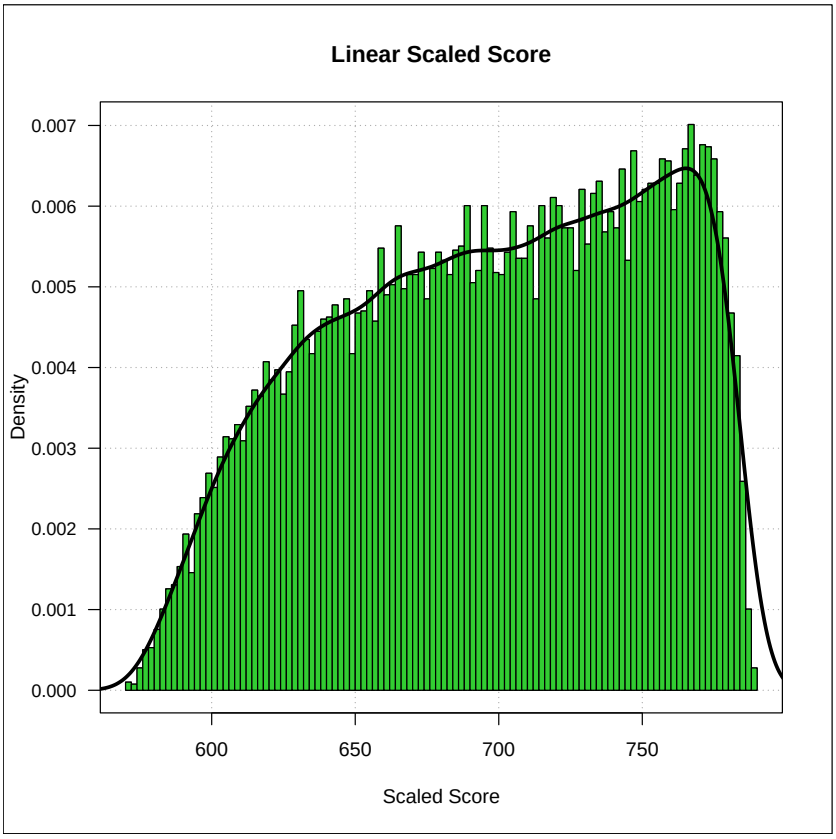

Figure 7: Linear scaled score histogram.

As additional check, we can compare the desired good to bad odds and the observed odds for the OLS and cubic transformation scaled scores in a small interval around three specific score values. As we recall, at the base score of $S_b \approx= 650$ we wanted achieve a good to bad odds ratio or roughly $O_b \approx 4.7$ with Points to Double the Odds of roughly 30. In Table 5 we present the observed odds in small intervals around 620, 650 and 680 for the two competing scaled scores. We can clearly see that the cubic transformation is able to better match the desired odds.

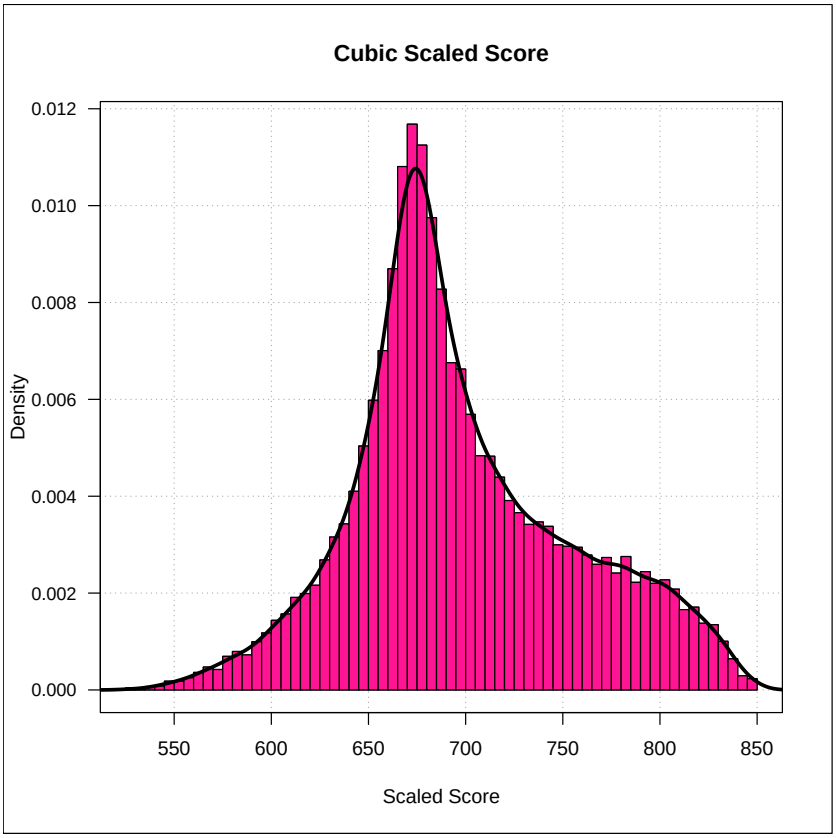

Figure 8: Cubic scaled score histogram.

Table 5: OLS and Cubic scaled score comparison.

	$S_b - PDO$	S_b	$S_b + PDO$
Desired Odds	2.4	4.7	9.4
OLS Transformation Odds	3.2	6.6	9.1
Cubic Transformation Odds	2.3	4.3	9.9

As a final check, we can look at the plots of the observed odds rate to versus the two scaled scores. Please refer to Figures 9 and 10. Notice that even after the OLS transformation, the resulting scaled score still presents a cubic relationship with the log odds. Compared with the cubic transformation scaled score, which now has a linear relationship with the log odds. A linear relationship between the resulting scaled score and the log odds is typically preferred by business end users.

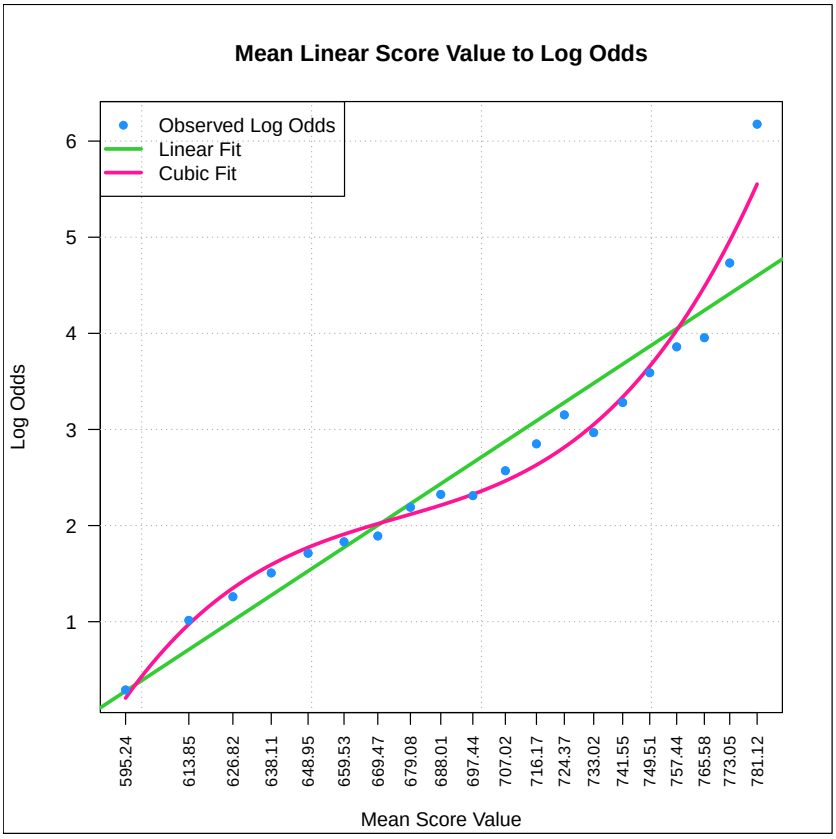

Figure 9: Log odds and OLS scaled scores.

6. Conclusion

We have provided a novel approach to accomplish scaling of a credit score. The key difference in our approach is to utilize a cubic transformation when mapping raw unscaled score values to the fitted log odds space, which is then mapped to the scaled score space via the Fundamental Scaling Equation. The key benefits of our approach are the following.

- Relative to the traditional linear transformation function, the cubic transformation provides a wider range of values. This is important since our intended application is risk differentiation. Range compression is typically compounded by the business requirement to generate integer valued credit risk scores.

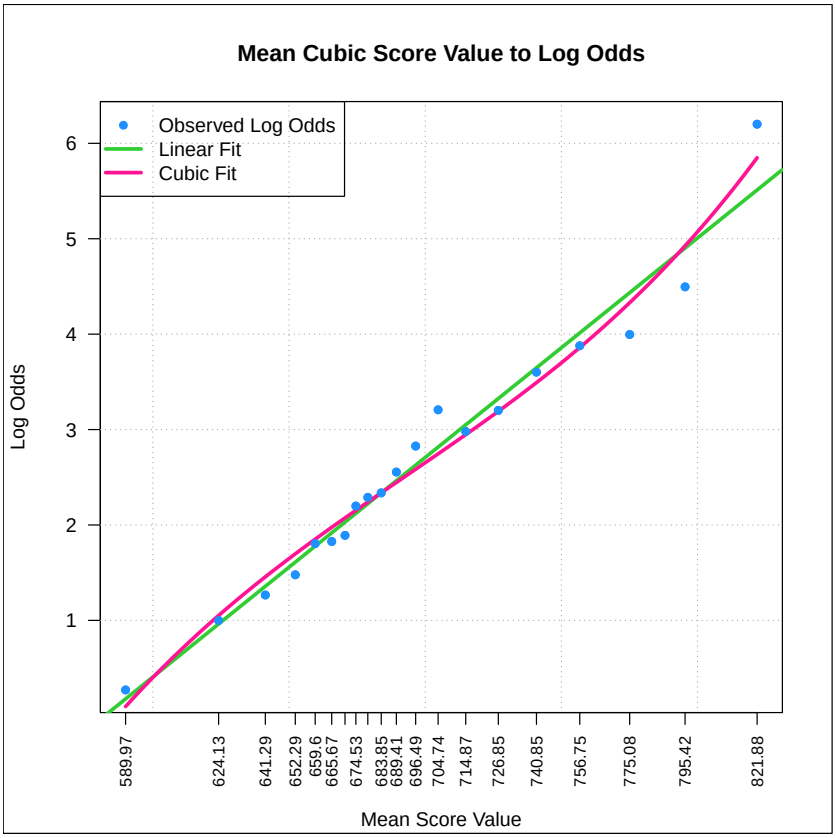

Figure 10: Log odds and Cubic scaled scores.

- The cubic transformation is better able to match the desired good to bad odds ratio at a selected base score value. In addition the Points to Double the Odds (PDO) is also matched more accurately.
- The distribution of the cubic scaled score is more bell shaped, versus the linear transformation.
- Lastly, the cubic scaled score is linear in the log odds space, whereas the linear scaled score still exhibits a third order polynomial relationship with log odds.

For all of the above reasons, practitioners who need to scale a bespoke score, especially an ML score, to a known score range for an existing score, and also match the odds and PDO, should consider exploring both the classic

linear transformation as well as cubic transformation. Further research work that is motivated may be to consider other more flexible transformations such as the Box–Cox power transformation, smoothing splines, LOESS, to name just a few.