UQ for Credit Risk Management: A deep evidence regression approach

Ashish Dhiman, ashish1610dhiman@gmail.com

Abstract. Machine Learning has invariantly found its way into various Credit Risk applications. Due to the intrinsic nature of Credit Risk, quantifying the uncertainty of the predicted risk metrics is essential, and applying uncertainty-aware deep learning models to credit risk settings can be very helpful. In this work, we have explored the application of a scalable UQ-aware deep learning technique, Deep Evidence Regression and applied it to predicting Loss Given Default. We contribute to the literature by extending the Deep Evidence Regression methodology to learning target variables generated by a Weibull process and provide the relevant learning framework. We demonstrate the application of our approach to both simulated and real-world data.

1. Introduction.

1.1. Credit Risk Management.

Credit risk management is assessing and managing the potential losses that may arise from the failure of borrowers or counterparties to fulfil their financial obligations. In other words, it identifies, measures, and mitigates the risks associated with lending money or extending credit to individuals, businesses, or other organizations.

Credit risk's anticipated loss (EL) comprises three components: Probability of Default (PD), Loss Given Default (LGD), and Exposure at Default (EAD). PD is the likelihood that a borrower will fail to fulfill their financial commitments in the future. LGD refers to the proportion of the outstanding amount that is lost in the event of default. Lastly, EAD refers to the outstanding amount at the time of default.[6]

LGD prediction is important as accurate prediction of LGD not only supports a healthier and risk-less allocation of capital, but is also vital for pricing the security properly. [6] & [7]. There is a large body of literature using advanced statistical and machine learning methods for prediction of LGD [6]. However the machine learning literature on LGD has yet to address an essential aspect, which is the uncertainty surrounding the estimates and predictions.[4].

UQ techniques like Bayesian Neural Network, Monte Carlo Dropout and ensemble methods as outlined in [1] present a natural first step towards quantifying uncertainty. However, almost all these methods are computationally and memory intensive, and require sampling on test data after fitting the network, making them difficult to adapt for complex neural network architectures that involve a large number of parameters.

1.2. Deep Evidence Regression.

The primary inspiration of this work is taken from the work done by Amini et al in [2]. The paper develops a unique approach, **Deep Evidence Regression** as a scalable and accurate UQ aware deeo learning technique for regression problems. This approach predicts the types of uncertainty directly within the neural network structure, by learning prior distributions over the parameters of the target distribution, referred to as evidential distributions. Thus this method is able to quantify uncertainty without extra computations after training, since the estimated parameters of the evidential distribution can be plugged into analytical formulas for epistemic and aleatoric uncertainty, and target predictions.

The setup of the problem is to assume that the observations from the target variable, y_i

are drawn i.i.d. from a **Normal distribution** with unknown mean and variance parameters $\theta = \mu, \sigma^2$. With this we can write the log likelihood of the observation as:

$$Lik(\mu, \sigma^2) = log(p(y_i|\mu, \sigma^2) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y_i - \mu)^2}{2\sigma^2}$$

Learning θ that maximises the above likelihood successfully models the uncertainty in the data, also known as the aleatoric uncertainty. However, this model is oblivious to its predictive epistemic uncertainty. [2]. Epistemic uncertainty, is incorporated by placing higher-order prior distributions over the parameters θ. In particular a Gaussian prior is placed on the unknown mean and an Inverse-Gamma prior on the unknown variance.

$$\mu \sim \mathcal{N}(\gamma, \sigma^2\nu^{-1}) \quad \sigma^2 \sim \Gamma^{-1}(\alpha, \beta)$$

Following from above the posterior $p(\mu, \sigma^2|\gamma, \nu, \alpha, \beta)$ can be approximated as $p(\mu|\gamma, \nu) * p(\sigma^2|\alpha, \beta)$. Hence:

$$p(\mu, \sigma^2|\gamma, \nu, \alpha, \beta) = \frac{\beta^\alpha\sqrt{\nu}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}(1/\sigma^2)^{\alpha+1}\exp\left(-\frac{2\beta + \nu(\gamma - \mu)^2}{2\sigma^2}\right)$$

Amini et al [2] thus find the likelihood of target variable given evidential parameters, as:

$$p(y_i|\gamma, \nu, \alpha, \beta) = \int_\theta p(y_i|\theta)p(\theta|\gamma, \nu, \alpha, \beta)$$

where $\theta = \{\mu, \sigma^2\}$. Then a Neural Network is trained t infer, the parameters $m = \{\gamma, \nu, \alpha, \beta\}$, of this higher-order, evidential distribution.

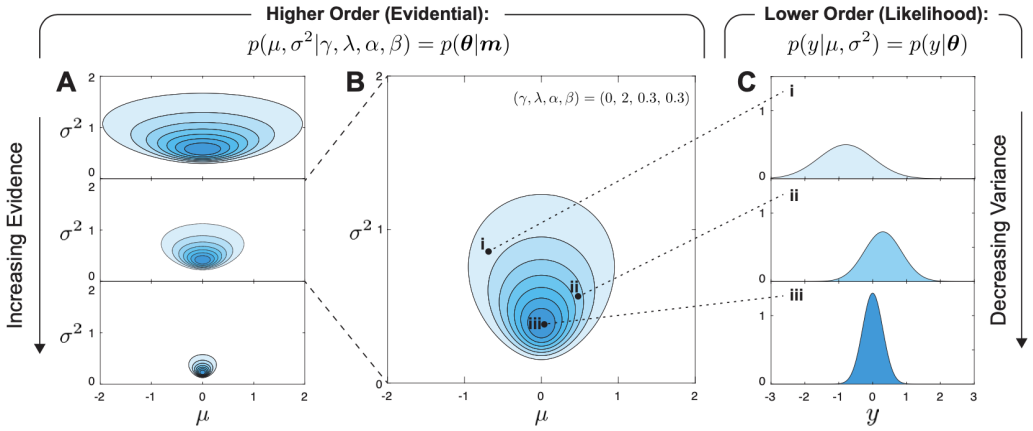

Figure 1.1. *Deep Evidence Regression(image from [2])*

1.3. Weibull distribution. The Weibull distribution is a continuous probability distribution commonly used in reliability analysis to model the failure time of a system or component. The probability density function (PDF) of the Weibull distribution is given by:

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & \text{if } x \geq 0, \\ 0 & \text{if } x < 0, \end{cases} \tag{1.1}$$

where $\lambda > 0$ is the scale parameter and $k > 0$ is the shape parameter. The scale parameter determines the location of the distribution, while the shape parameter controls the rate at which the failure rate changes over time. The Weibull distribution can be used to model a variety of phenomena, including the Loss given default values.

The work by [2] assumes a normal distribution on LGD values. While this assumption might be true in a lot of settings, however it does not follow in the context of Loss Given Default. While normal distribution is symmetric and has a support over entire real line, however the LGD values are restricted to a range of $[0, 1]$ and might not necessarily be symmetric.

Hence in the section below we provide a novel theoretical framework to learn target variables which follow Weibull distribution. We provide the following theoretical results, in the setting of target variables following a Weibull dataset.

- Log Likelihood
- Mean Prediction
- Prediction Uncertainty

We also provide results testing our approach on both simulated and real world dataset.

2. Deep Evidence Regression for Weibull Data.

2.1. Problem setup. We consider the problem where the observed targets, y_i, are drawn iid from a Weibull distribution, with a known shape or rate parameter k and an unknown scale λ. Although ideally we would want to keep both the parameters unknown, however with both λ and k there are no priors with which likelihood can be computed analytically [3]. Hence we have decided to simplify the problem setup by assuming known shape k.

$$y_i \sim Weibull(k, \lambda) \tag{2.1}$$

$$\text{where } k \in \mathrm{R}^+, \lambda \in \mathrm{R}^+ \tag{2.2}$$

$$\text{Hence pdf of } y_i \text{ is} \tag{2.3}$$

$$\implies p(y_i;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{y_i}{\lambda}\right)^{k-1} e^{-(y_i/\lambda)^k}, & \text{if } y_i \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

For the above setting we want to place priors on the unknown parameter, λ, such that we are able to get solve for the likelihood of y_i given the parameters of the prior distribution. Hence similar to work in [8] and [5], we define the following prior.

$$\theta = \lambda^k \tag{2.5}$$

Hence the pdf of y_i becomes: (2.6)

$$p(y_i|\theta, k) = \frac{k}{\theta} {y_i}^{k-1} \exp\left(-y_i^k/\theta\right) \tag{2.7}$$

And we place a Inverse Gamma Prior on θ (2.8)

$$\theta \sim \Gamma(\alpha, \beta) \quad (\alpha > 0) \tag{2.9}$$

Hence pdf of θ is (2.10)

$$p(\theta|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{1}{\theta^{\alpha+1}} \exp\left(-\frac{\beta}{\theta}\right) \tag{2.11}$$

2.2. Learning Log-Likelihood. Hence we can define likelihood of y_i given the higher order evidential parameters α, β can be defined as :

$$Lik = p(y_i|\alpha, \beta) = \int_\theta p(y_i|\theta, k) p(\theta|\alpha, \beta) d\theta \tag{2.12}$$

$$\text{Now given } \lambda, k > 0 \implies \theta > 0 \tag{2.13}$$

$$p(y_i|\alpha, \beta) = \int_{\theta=0}^{\infty} \Big(\frac{k}{\theta} {y_i}^{k-1} \exp\left(-y_i^k/\theta\right)\Big) \Big(\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{1}{\theta^{\alpha+1}} \exp\left(-\frac{\beta}{\theta}\right)\Big) d\theta \tag{2.14}$$

$$= k y_i^{k-1} \frac{\beta^\alpha}{\Gamma(\alpha)} \int_{\theta=0}^{\infty} \frac{1}{\theta^{\alpha+2}} \exp(-\frac{y_i^k + \beta}{\theta}) \tag{2.15}$$

(2.16)

Now let (2.17)

$$\Omega = k y_i^{k-1} \frac{\beta^\alpha}{\Gamma(\alpha)} \quad \text{and,} \tag{2.18}$$

$$a = y_i^k + \beta \tag{2.19}$$

$$\implies p(y_i|\alpha, \beta) = \Omega \int_{\theta=0}^{\infty} \frac{1}{\theta^{\alpha+2}} \exp(-\frac{a}{\theta}) d\theta \tag{2.20}$$

Substituting $t = 1/\theta$ (2.21)

$$\implies p(y_i|\alpha, \beta) = \Omega \int_{\theta=\infty}^{0} t^{\alpha+2} \exp(-at)(-dt * t^{-2}) \tag{2.22}$$

$$= \Omega \int_{\theta=0}^{\infty} t^\alpha \exp(-at) dt \tag{2.23}$$

(2.24)

(2.25) Now we know from [10]

$$\int_0^\infty x^n \exp(-ax)dx = \frac{\Gamma(1+n)}{a^{1+n}} \tag{2.26}$$

$$\implies p(y_i|\alpha,\beta) = \Omega\frac{\Gamma(1+\alpha)}{a^{1+\alpha}} \tag{2.27}$$

(2.28) substituting back Ω and a

$$\implies Lik = p(y_i|\alpha,\beta) = ky_i^{k-1}\frac{\beta^\alpha}{\Gamma(\alpha)}\frac{\Gamma(1+\alpha)}{(y_i^k+\beta)^{1+\alpha}} \tag{2.29}$$

$$= \frac{\alpha k y_i^{k-1}\beta^\alpha}{(y_i^k+\beta)^{\alpha+1}} \tag{2.30}$$

Hence the log-likelihood for i'th observation is defined as:

$$Log - Lik_i = L_i^{lik} = \log\alpha_i + \log k + (k-1)\log y_i + \alpha_i \log\beta_i - (\alpha_i+1)({y_i}^k + \beta_i) \tag{2.31}$$

We set up our neural network to minimise the negative Log-Likelihood plus some regularisation cost, discussed in section below.

2.3. Mean Prediction and UQ of prediction. Given the main advanatge of Deep Evidence Regression over other UQ aware deep learning methods like Bayesian NN, esembling etc, is due to existence of analytical solution for both predictions and unceratinty from NN output, without the need for sampling. Hence this section details the derivation of mean prediction and epistemic uncertainty.

2.3.1. Mean Prediction.

(2.32) We define the mean prediction as

$$Z = E[y_i|\alpha,\beta] \tag{2.33}$$

(2.34) Now given $y_i \sim Weibull(k,\lambda)$

$$Z = E[\lambda * \Gamma(1+\frac{1}{k})] = E(\lambda) * \Gamma(1+\frac{1}{k}) \quad \text{(k is known)} \tag{2.35}$$

$$E[\lambda] = \int_\lambda \lambda p(\lambda)d\lambda \tag{2.36}$$

(2.37)

Hence to solve for mean prediction we need to find pdf $p(\lambda)$. Because we know $\theta = \lambda^k \sim \Gamma^{-1}(\alpha,\beta)$, we can use change of variable to find pdf of λ [9].

$$p(\lambda|\alpha,\beta) = p_\theta(\lambda^k) * |\frac{d\lambda^k}{d\lambda}| \tag{2.38}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\frac{1}{\lambda^k})^{\alpha+1} \exp(-\frac{\beta}{\lambda^k}) * |k\lambda^{k-1}| \tag{2.39}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\frac{1}{\lambda^k})^{\alpha+1} \exp(-\frac{\beta}{\lambda^k}) * k\lambda^{k-1} \quad (\text{given } \lambda, k > 0) \tag{2.40}$$

Hence,

$$E[\lambda] = \int_\lambda \lambda \frac{\beta^\alpha}{\Gamma(\alpha)}(\frac{1}{\lambda^k})^{\alpha+1} \exp(-\frac{\beta}{\lambda^k}) * k\lambda^{k-1} d\lambda \tag{2.41}$$

$$= \frac{k\beta^\alpha}{\Gamma(\alpha)} \int_{\lambda=0}^{\infty} \frac{1}{\lambda^{k\alpha+k-k}} \exp(\frac{\beta}{\lambda^2}) d\lambda \tag{2.42}$$

Substituting $t = 1/\lambda$, we get: (2.43)

$$dt = -1/\lambda^2 d\lambda, \tag{2.44}$$

$$E[\lambda] = \frac{k\beta^\alpha}{\Gamma(\alpha)} \int_{t=0}^{\infty} t^{k\alpha-2} \exp(-\beta t^k) dt \tag{2.45}$$

By table of integrals at [10] (2.46)

$$\int_0^\infty y^m e^{-by^k}, dy = \frac{\Gamma\left(\frac{m+1}{k}\right)}{kb^{(m+1)/k}} \tag{2.47}$$

$$\implies E[\lambda] = \frac{k\beta^\alpha}{\Gamma(\alpha)} * \Gamma(\frac{k\alpha-1}{k}) * \frac{1}{k} * \frac{1}{\beta^{\frac{k\alpha-1}{k}}} \tag{2.48}$$

Hence we get the mean prediction as:

$$Z = E[y_i|\alpha,\beta] = E(\lambda) * \Gamma(1+\frac{1}{k}) \tag{2.49}$$

$$\frac{k\beta^\alpha}{\Gamma(\alpha)} * \Gamma(\frac{k\alpha-1}{k}) * \frac{1}{k} * \frac{1}{\beta^{\frac{k\alpha-1}{k}}} * \Gamma(1+\frac{1}{k}) \tag{2.50}$$

$$= \Gamma(1+\frac{1}{k})\frac{1}{\Gamma(\alpha)}\Gamma(\alpha-\frac{1}{k}) * \beta^{1/k} \tag{2.51}$$

2.3.2. UQ of Prediction. We quantify the epistemic uncertainty as $Var(Z)$ with defined as above, i.e. $Z = E[y_i|\alpha,\beta]$

$$Var(Z) = E(Z^2) - [E(Z)]^2 \tag{2.52}$$

With $E(Z)$ defined as in 2.49, we only need $E(Z^2)$ (2.53)

$$E[Z^2] = E[\lambda^2 * \Gamma^2(1 + \frac{1}{k})] \tag{2.54}$$

$$= E[\lambda^2] * \Gamma^2(1 + \frac{1}{k})] \tag{2.55}$$

Similar to approach outlined in 2.3.1, we get: (2.56)

$$E[\lambda^2] = \Gamma(1 + \frac{2}{k})\frac{1}{\Gamma(\alpha)}\Gamma(\alpha - \frac{2}{k}) * \beta^{2/k} \tag{2.57}$$

Hence we can write

$$Var(Z) \propto \frac{\beta}{\Gamma(\alpha)}\big(\Gamma(\alpha - \frac{2}{k}) - \Gamma(\alpha - \frac{1}{k})\big)$$

2.4. Regularisation Cost. As followed in [2], the intuition behind the regularisation cost is to increases the variance of prediction in cases where it's unsure. Hence we define the Regularisation cost for the i'th observation as

$$L_i^{reg} = |error_i| * (\alpha_i)$$

where $error_i = y_i - Z_i$ and Z is defined in 2.49.

Note the regularisation cost defined above is what was found to work best in experiments, we need more theoretical analysis to decide the exact coefficients of α and β in the regularisation cost.

2.5. Deep NN training. The learning process is then set up with a deep neural network with two output neurons to predict the parameters of the prior/evidential distribution, α and β. The neural network is trained using the cost function:

$$L_i^{NN} = -L_i^{lik} + c * L_i^{reg}$$

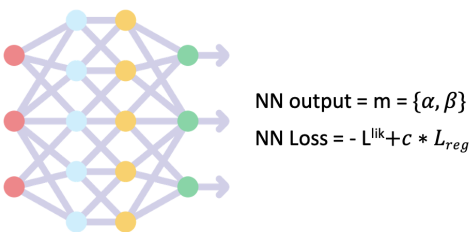

Figure 2.1. *expected timeline*

where c is the hyper-parameter governing the strength of regularisation.

3. Results and Experiments. In this section, we present the results of experiments conducted on both simulated and real data. The aim was to evaluate the performance of our proposed method and compare it with existing methods. The simulated data was generated based on example data given in [2], while the real dataset was obtained from a peer to peer lending company.

3.1. Simulated Data. Here generate a target variable following a Weibull distribution. The target variable is generated as:

$y_i = x_i^2 + \epsilon, \epsilon \sim Weibull(k = 1.2, \lambda = 0.2)$

The train set is comprised of $x \in [0, 3]$ while test set is $x \in [0, 4]$.

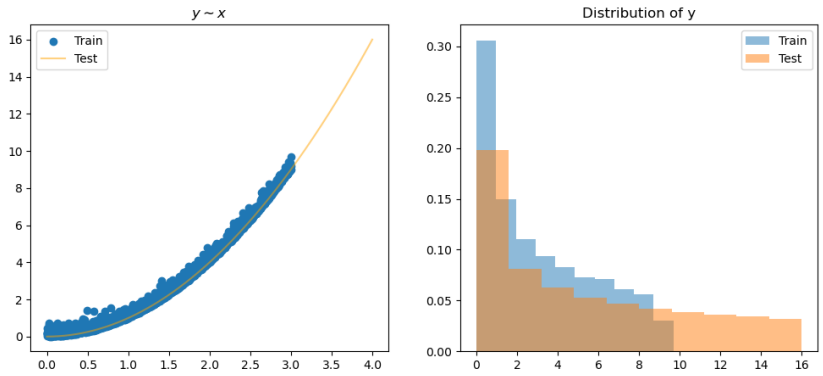

Figure 3.1. *y vs x and Distribution of y(right) for synthetic data*

Since our approach assumes known k, k is estimated from the training set.

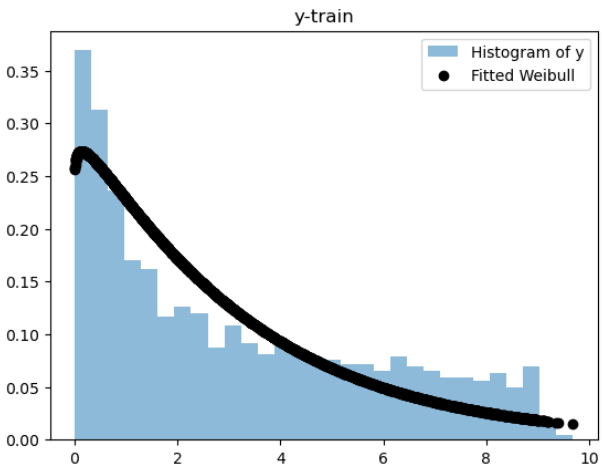

Figure 3.2. *Weibull Fit on Train data*

Next we fit both the original deep evidence regression and proposed weibull version of deep evidence regression. We find that original version predicts the same uncertainty irrespective of Out-of-Distribution data or not. On the other hand the prediction interval starts growing

towards the end of train data i.e. beyond x¿3. Having said that the Deep evidence regression is better able to capture the signal outside the train window.

	original	**proposed**
MSE (overall)	0.1475	0.4380
MSE (train range)	0.0242	0.0257
var(overall)	0.1187	8.3923
var(in test)	0.1518	18.3143

Table 3.1

Deep Evidence regression original vs proposed results on simulated data. We see MSE-train is similar for both, while var or epistemic uncertainty is better captured by proposed.

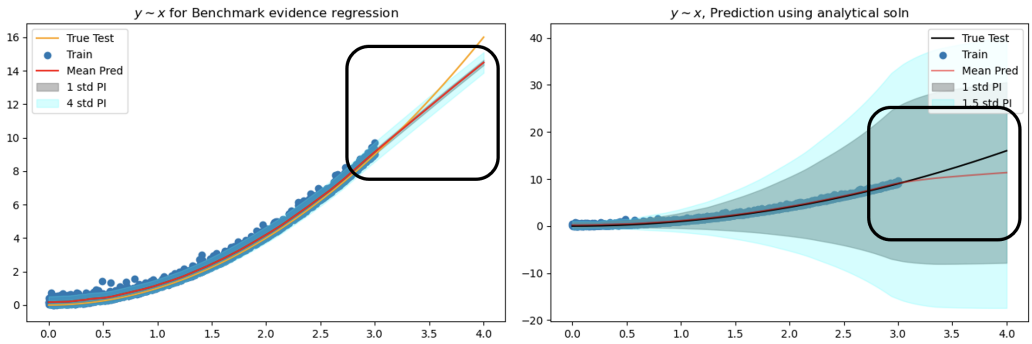

Figure 3.3. *Deep evidence regression (left) vs Weibull evidence Regression (right). We see that while both the models are very close to true prediction in the train range, however, uncertainty is much better captured by proposed version in case of OOD test data*

To confirm that our analytical calculations make sense, we have also created the mean prediction after sampling from the NN outputs. Firstly we sample θ from $\Gamma(\alpha, \beta)$. λ is then calculated as the k-th root of θ or $\lambda = \theta^{1/k}$. Finally, the response variable y_i can be sampled as $Weibull(\lambda, k)$. The consistency between the results from sampling and analytical calculations, supports the analytical calculations in 2.49.

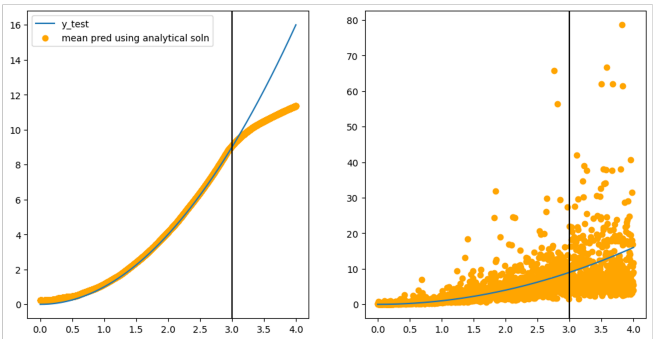

Figure 3.4. *Mean prediction using analytical equation(left) vs from sampling (right)*

3.2. Real Data: Loss Given Default for peer to peer lending. The dataset under consideration pertains to mortgage data sourced from Peer to Peer lending during the period of 2007 to 2011. Peer-to-peer lending, which is an emerging form of credit aimed at funding borrowers from small lenders and individuals seeking to earn interest on their investments. Through an online platform, borrowers can apply for personal loans, which are typically unsecured and funded by one or more peer investors. The P2P lender acts as a facilitator of the lending process and provides the platform, rather than acting as an actual lender. While these platforms have become popular in the US, UK, Australia, and other financial markets, the US is currently leading the way in the peer-to-peer lending arena.

However, the data does not include the loss given default values. Instead, the recovery rate has been used as a proxy, which is calculated as the ratio of recoveries made to the origination amount. The dataset contains approximately 46 variables denoted as 'x,' which include features such as the time since the loan was issued, debt-to-income ratio (DTI), joint applicant status, and delinquency status, among others. In total, the dataset comprises around 23,000 rows.

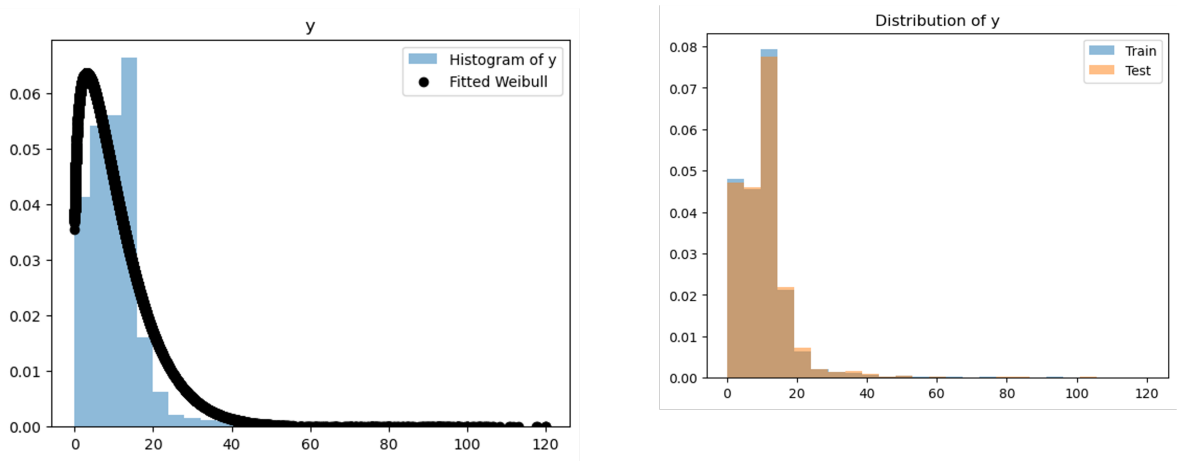

Figure 3.5. *Distribution and Weibull fit of the recovery rate (left). Histogram of recovery rate for train/test split (right). It is evident that Weibull might not be a good fit to this data.*

As described in the approach the shape parameter was found as $k = 1.2386$ by fitting a Weibull distribution on the train dataset. Also given the sensitivity of both the approaches to regression cost, cross validation was done to arrive at the best regularisation cost.

The model summary for Benchmark model is:

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_210 (Dense)            (None, 1)                 46

 dense_211 (Dense)            (None, 100)               200
```

```
 dense_212 (Dense)           (None, 64)                6464

 dense_213 (Dense)           (None, 32)                2080

 dense_214 (Dense)           (None, 20)                660

 dense_normal_gamma_35 (Dens  (None, 4)                84
 eNormalGamma)

=================================================================
Total params: 9,534
Trainable params: 9,534
Non-trainable params: 0
_________________________________________________________________
```

The model summary for Weibull model is:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_616 (Dense)           (None, 1)                 46

 dense_617 (Dense)           (None, 350)               700

 dense_618 (Dense)           (None, 300)               105300

 dense_619 (Dense)           (None, 300)               90300

 dense_620 (Dense)           (None, 250)               75250

 dense_621 (Dense)           (None, 250)               62750

 dense_622 (Dense)           (None, 200)               50200

 dense_623 (Dense)           (None, 200)               40200

 dense_624 (Dense)           (None, 200)               40200

 dense_weibull_gamma_35 (Den  (None, 2)                402
 seWeibullGamma)

=================================================================
Total params: 465,348
Trainable params: 465,348
Non-trainable params: 0
_________________________________________________________________
```

	original	**proposed**
Regularisation cost (c)	0.0450	0.0023
MSE-train	22.3345	**17.4018**
MSE-test	19.3115	**16.7881**
var-train	np.nan	**252.1040**
var-test	np.nan	**268.8399**

Table 3.2

Results for original vs proposed model for recovery rate. proposed version does not only have lower RMSE, but also is able to capture uncertainty more aptly.

The results from the model are given below:

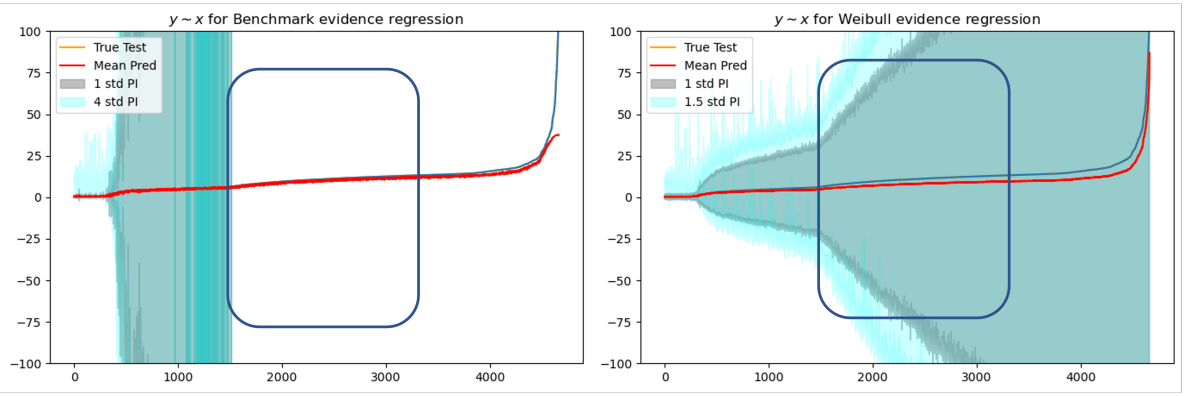

Figure 3.6. *Predicted UQ for Benchmark Regression (left) vs proposed model(right). Again we see that the updated model is much better able to capture the UQ. With Uncertainty increasing after recovery rate > 30, which is the less dense region and has fewer observations.*

4. Conclusion and Discussion. We propose an improvement over Deep Evidence Regression, specifically targeted to usecases where the target might follow Weibull distribution. We then test the proposed method both on simulated and real world dataset in the context of Credit risk management. In our experiments the proposed model appears to be more effective in dealing with Weibull targets.

However, we are not sure if the proposed approach would generalise to other distributions apart from Weibull. Additionally, the proposed model has only two outputs, which could limit its flexibility when compared to the benchmark model, which had four outputs from the neural network. In consequence the proposed model requires a deeper network architecture compared to the benchmark model. Furthermore we find that both the models exhibit a high sensitivity to regularization cost, which means that changes in the regularization coefficient can significantly impact the model's performance.

Overall, these points suggest that both models have their strengths and weaknesses, and selecting the most appropriate model depends on the specific task requirements and considerations.

REFERENCES

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14927–14937. Curran Associates, Inc., 2020.

[3] Pasquale Erto and Massimiliano Giorgio. A note on using bayes priors for weibull distribution, 2013.

[4] Paolo Gambetti, Geneviève Gauthier, and Frédéric Vrins. Recovery rates: Uncertainty certainly matters. *Journal of Banking & Finance*, 106:371–383, 2019.

[5] Pseudodifferential Operators (https://stats.stackexchange.com/users/320438/pseudodifferential operators). Find a conjugate prior for the weibull distribution under reparametrization. Cross Validated. URL:https://stats.stackexchange.com/q/594050 (version: 2022-10-30).

[6] Matthias Nagl, Maximilian Nagl, and Daniel Rösch. Quantifying uncertainty of machine learning methods for loss given default. *Frontiers in Applied Mathematics and Statistics*, 8, 2022.

[7] Luc Severeijns. Challenging lgd models with machine learning, 2018.

[8] tomka (https://math.stackexchange.com/users/118706/tomka). Conjugate prior for the weibull distribution. Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/3529658 (version: 2020-02-04).

[9] Wikipedia contributors. Change of variables — Wikipedia, the free encyclopedia, 2023. [Online; accessed 1-May-2023].

[10] Wikipedia contributors. List of integrals of exponential functions — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_integrals_of_exponential_functions&oldid=1142840420, 2023. [Online; accessed 1-May-2023].