Case Study: Probabilistic Expense Prediction

Ashutosh Tewari

February 21, 2023

1 Overview

This case study involves analyzing consumer expenditure behavior in the United States and devising models for predicting monthly household expenses (given some other attributes). A key requirement is to have the ability to quantify uncertainty in the predictions and compute quantiles for the same. Also, the models should be rooted in modern Machine Learning methods for better performance. This short document is intended to serve as supplementary material to go along with the submitted code. The purpose here is to elucidate the rationale behind different choices made during this modeling exercise.

1.1 Dataset

The dataset comes from a Consumer Expenditure survey spanning multiple years, with good coverage, regionally and demographically, across the US. This dataset is further augmented with the monthly inflation and unemployment rates, over the same time frame, procured from the US Bureau of Labor Statistics database. The combined dataset comprises 6 numeric (*age, family-size, inflation, unemployment, income, expense*), 6 categorical (*urban, race, marital, occupation, state, region*) and 1 ordinal (*education*) variables, for ~900k unique households. The dataset is mostly complete with less than 0.01% missing values. As a preprocessing step, households with non-positive incomes and expenses were purged (~ 5% of data).

2 Methods

This is a classic probabilistic regression setting, where the goal is to obtain the conditional density $P(y|X)$ of the target y (i.e. *expense*) given the rest of the attributes as covariates, X. When viewed as a table, the rows $[y, X]$ can be seen as i.i.d samples from some unknown joint distribution $P(y, X)$. If a dataset is complete (as the case here), it is often beneficial and more tractable to directly model the conditional density $P(y|X)$ than learning the joint $P(y, X)$ (well-known *discriminative* vs. *generative* trade-off). For this reason, the conditional modeling route is taken in this case study. Additionally, if the goal is to get the quantiles at some prescribed levels (p10, p50, p90, etc.), conditional modeling can further be performed *explicitly* or *implicitly*. In the former, one "explicitly" learns a parametric conditional density, while in the latter only the desired quantiles are learned. Both these modeling approaches were employed in this case study and their performances are compared. Despite modeling the predictive distribution differently, a common theme in both approaches is the use of deep neural networks for model parameterization.

2.1 Feature Selection and Transformations

Due to the heavy-tailed nature of the *expense* and *income* variables, log-transformations are employed, followed by further standardizations (zero-mean, unit-variance). Essentially, the conditional density $P(\tilde{y}|\tilde{X})$ is sought, where $\tilde{\square}$ denotes the transformed variables. A desirable property of monotonic transformations (such as log and standardization) is that the quantile computation (in the original space) is straightforward and simply requires inversion of the transformed space quantiles. This follows from the invariance of cumulative density values under monotonic transformations. PyTorch's *Distributions* class allows one to specify such "transformed distributions" that makes it really convenient to toggle between different spaces connected via such bijective mappings. Hence, this class is used liberally in coding part of this case study. Additionally, the categorical covariates are transformed via *one-hot* encoding, and the only ordinal covariate, *education*, is kept as is.

A quick check on the information content between the covariates and the target variable is performed by analyzing the pair-wise *Mutual Information* (MI). All the continuous variables were discretized (to a sufficient degree) to make the MI computation tractable. Figure 1 shows the computed MI values along with an arbitrarily chosen threshold line. Clearly, not all covariates are informative about our target

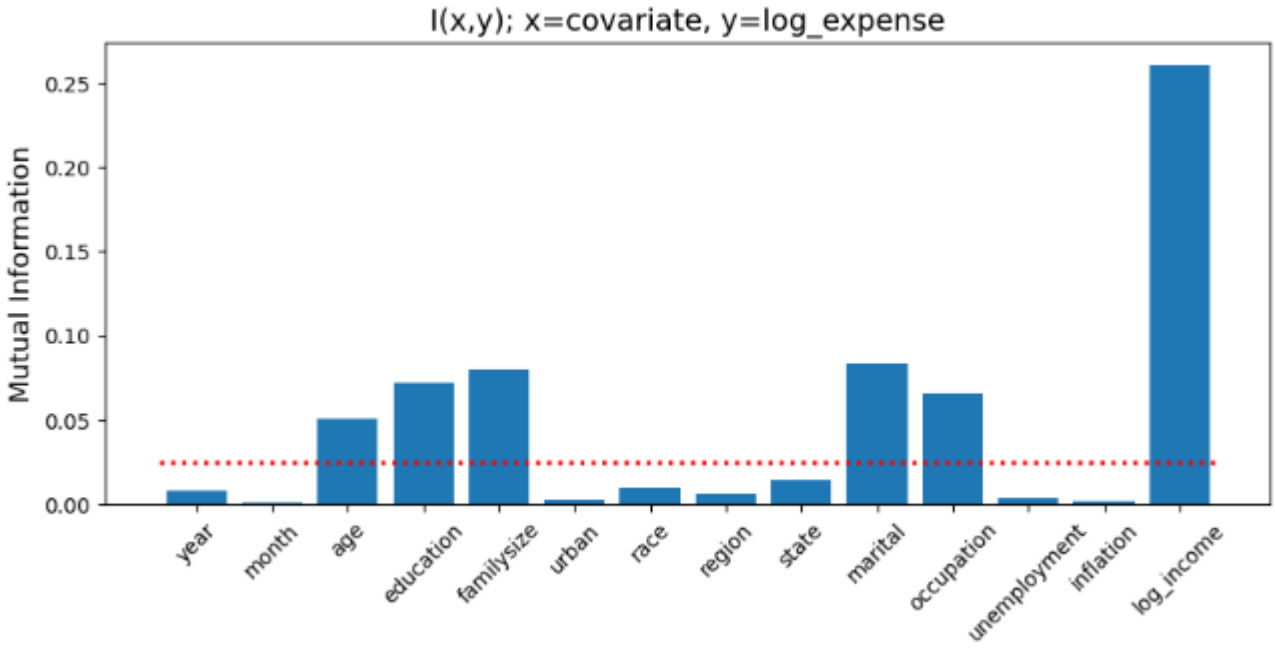


Figure 1: Mutual Information of the target (log expense) variable with respect to all the other covariates. The dotted-red line shows a threshold for feature selection.

variable. Therefore, selecting only those covariates (*age, education, family-size, marital, occupation, and log-income*) with MI above the shown threshold should be okay for the task at hand. Nevertheless, the impact of macroeconomic indicators (*inflation* and *unemployment* rates) is also assessed despite seemingly low MI.

2.2 Conditional Gaussian Modeling

In this approach, the target density is explicitly modeled as a conditional Gaussian, where the mean and the scale values are derived from parametric functions of X as shown in Equation (1). Both these functions ($\mu(\cdot)$ and $\sigma(\cdot)$) are encoded as separate DNNs with learnable parameters θ and ϕ, respectively. The

flexibility endowed by the DNNs allows learning really complex conditionals (non-linear, heteroskedastic etc.), with minimal assumptions, provided there is sufficient data to tune the parameters. In this case study, DNNs with multiple dense hidden layers (with ReLU activation) were used to construct these functions.

$$P(y|X) = \mathcal{N}\left(y|\mu(X;\theta), \sigma(X;\phi)\right) \tag{1}$$

For parameter tuning in such settings, a commonly used loss function is the negative log-likelihood over the data. For the case of conditional Gaussian density, this loss is akin to the penalized ℓ_2-norm of the residual, r. It is also straightforward to incorporate observation weights in this formulation. An alternate loss function is the Pinball loss given as $\max(q \cdot r, (q-1) \cdot r))$ where r denotes the residual between an observation and the quantile at probability q. The Pinball loss computed over multiple probability levels $\sum_i \max\left(q_i \cdot r_i, (q_i - 1) \cdot r_i\right)$ can be viewed as a weighted ℓ_1-norm and can be a more robust alternative to the ℓ_2-loss. Both these losses were employed in this case study and the results are compared. Using Pinball loss for Gaussian conditionals is straightforward becasue the quantiles are easily computable and differentiable.

2.3 Quantile Regression

Here, rather than modeling the conditional density, the aim is to directly learn the quantiles at some prescribed levels, says $q = [{}^{1}/_{200}, {}^{1}/_{40}, {}^{33}/_{200}, {}^{1}/_{4}, {}^{1}/_{2}, {}^{3}/_{4}, {}^{167}/_{200}, {}^{195}/_{200}, {}^{199}/_{200}]$, by minimizing the aforementioned Pinball loss. This approach can be seen as an attempt to "implicitly" learn the underlying conditional density, by piece-wise approximating its distribution function (inverse quantile). As a result, there is no assumption of Gaussian density (or any other parametric family) which imparts additional flexibility to the model. One caveat is that the quantiles other than the prescribed ones can only be approximated by interpolation. In this case study, the quantile regression is implemented using DNNs. The network architecture is very similar to the one used for conditional Gaussian, except that the last layer outputs a vector of size 9 (i.e. the number of quantiles desired).

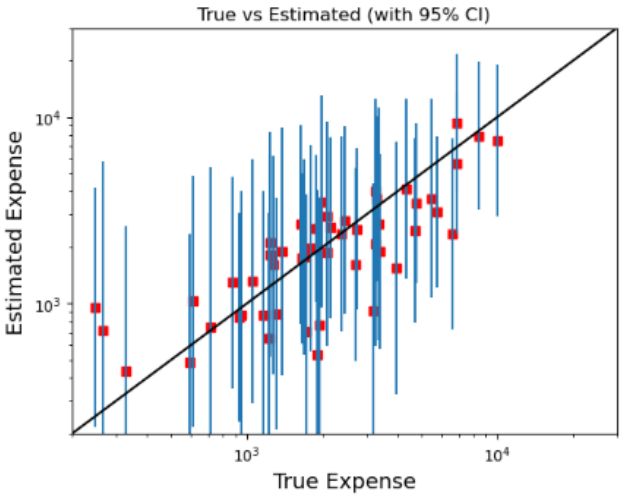
3 Summary of Results

Table 1 summarizes the results of probabilistic regression with different models and configurations. The *Baseline* model is a linear-Gaussian conditional model, while the rest use nonlinear DNN for modeling either the conditional densities or the quantiles. The loss value is the weighted-Pinball loss computed on the validation dataset. The nonlinear models clearly outperform the baseline model, nevertheless, the difference between them is marginal. Including macroeconomic indicators (inflation and unemployment rates) seem to help a bit. The quantiles submitted for the test dataset is based on the Model ID 3.

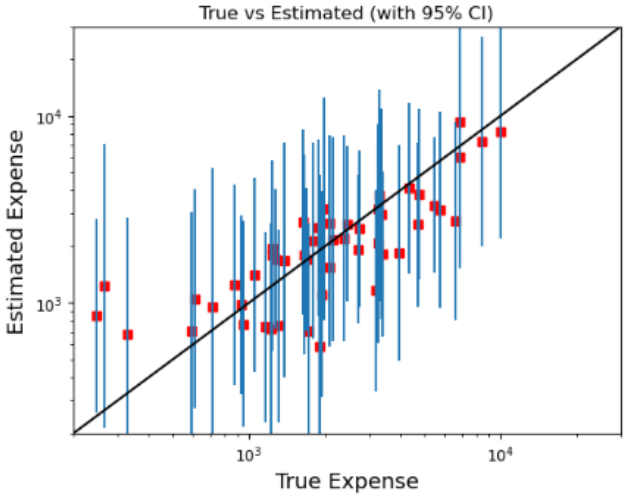
Lastly, for eyeballing purposes, Figure 2 shows the scatter plot for 4 models from Table 1. The error bars represent p90 prediction interval on 50 randomly selected data points from the validation set.

Table 1: Summary of performance by different models in different settings. Baseline is a linear regression model. The second column indicates the loss function was used for model training. The third indicates if the macroeconomic indicators were included in the regression. The last column shows the weighted Pinball loss on the validation dataset.

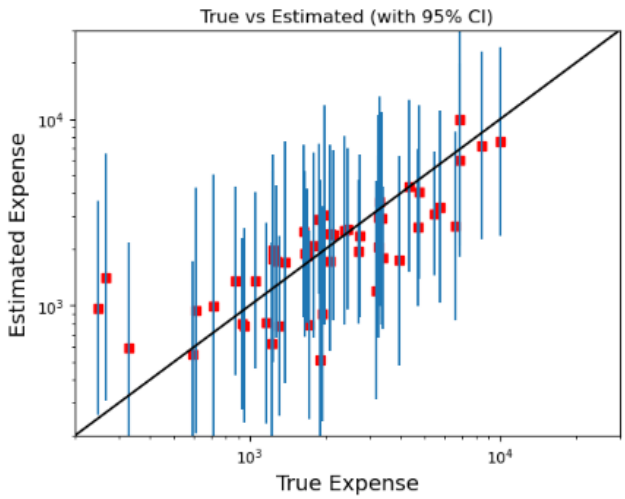
ID	**Models**	**Loss**	**Macro Used**	**Pinball Loss (validation)**
1	Baseline	MLE	Yes	702577
2	Conditional Gaussian	MLE	No	662378
3	Conditional Gaussian	MLE	Yes	659325
4	Conditional Gaussian	Pinball	No	664568
5	Conditional Gaussian	Pinball	Yes	662171
6	Quantile Regression	Pinball	No	660545
7	Quantile Regression	Pinball	Yes	659912



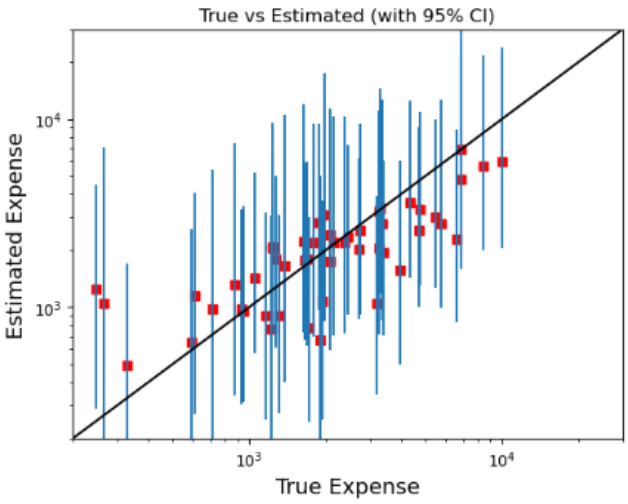
(a) Model ID-1



(b) Model ID -3



(c) Model ID -5



(d) Model ID-7

Figure 2: Scatter plots of True vs Estimated expense for four models referenced in Table 1. Fifty data points are randomly picked from the validation set for these plots.