**Model Diagnostics: Prediction Uncertainty**

[[PiML Tutorials](https://piml.medium.com/?source=post_page-----990ff0bd2c4e--------------------------------)](https://piml.medium.com/?source=post_page-----990ff0bd2c4e--------------------------------)

[PiML Tutorials](https://piml.medium.com/?source=post_page-----990ff0bd2c4e--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F75c69de4b20b&operation=register&redirect=https%3A%2F%2Fpiml.medium.com%2Fmodel-diagnostics-prediction-uncertainty-990ff0bd2c4e&user=PiML+Tutorials&userId=75c69de4b20b&source=post_page-75c69de4b20b----990ff0bd2c4e---------------------post_header-----------)

10 min read

·

Oct 31

102

*Today’s PiML tutorial is about quantification of prediction uncertainty for a pre-trained machine learning model. We take a deep dive into conformal prediction for regression models, introduce several ways of running split conformal prediction, and demonstrate how to detect unreliable regions via PiML reliability testing.*

**⚒️Machine Learning Model Uncertainty**

Uncertainty quantification (UQ) of machine learning is critical for real-world decision making, especially in high-stake scenarios where uncertain outcomes can have profound consequences. UQ not only adds a layer of model transparency, but also assesses reliability and level of confidence.

**📌Requirement of Risk Management**

UQ has always been a key requirement in effective risk management. It is also know as “expected ranges” or “reliability”. Among other sources, we collect quotes from[**SR11–7 Model Risk Management Framework (2011)**](https://www.federalreserve.gov/supervisionreg/srletters/sr1107a1.pdf)**:**

* “Outcome analysis …. establishing expected ranges for those actual outcomes in relation to the intended objectives and assessing the reasons for observed variation.”
* “Back-testing involves the comparison of actual outcomes with model forecasts not used in model development. The comparison is generally done using expected ranges or statistical confidence intervals around model forecasts.”

as well as the quote from [**NIST’s AI Risk Management Framework (Initial Draft, 2022)**](https://www.nist.gov/system/files/documents/2022/03/17/AI-RMF-1stdraft.pdf)**:**

* “Reliability indicates whether a model consistently generates the same results, within the bounds of acceptable statistical error. [It] can be a factor in determining thresholds for acceptable risk.”

**📌Sources of Uncertainty**

There are two main sources of uncertainty: the data side and the model side. The first side includes

* Data with noise: low or high, constant or heterogenous, outliers, distribution shift, etc.
* Data sparsity in regions, with limited representation or learning capacity
* Randomness in data partitioning (e.g., train-test-split)

and the second side includes

* Modeling framework, feature selection, and hyperparameter tuning;
* Stochastic optimization: random state, initialization, early stopping, non-guaranteed optimum.

We can re-group them into two types of uncertainty for a pre-trained model:

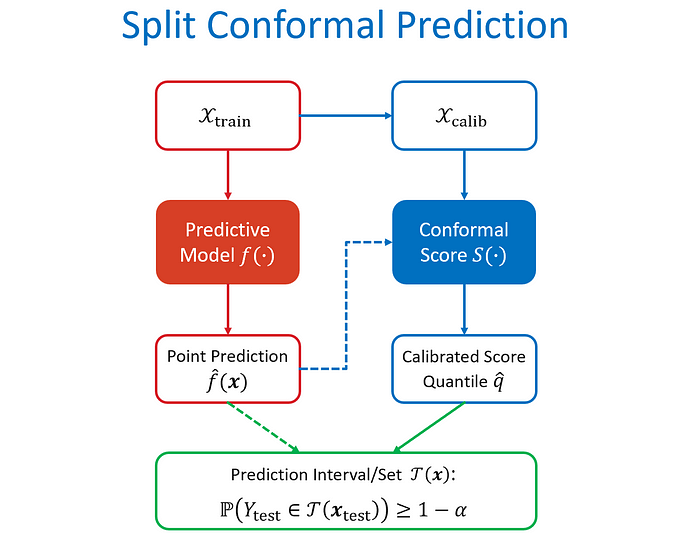
1. **prediction uncertainty** when the pre-trained model is fixed, where the uncertainty may come from data noise, data sparsity, and lack of model fit.
2. **retraining uncertainty** when the pre-trained model can be retrained, where the uncertainty may be due to a) random data splitting, b) hyperparameter tuning, and c) stochastic optimization.

We focus on **prediction uncertainty**in this tutorial, and leave the topic of **retraining uncertainty** to a future tutorial.

**🚀Split Conformal Prediction**

**Conformal prediction**is a distribution-free UQ framework pioneered by Vladimir Vovk since 1990s; see the monograph “Algorithmic learning in a random world” by Vovk, Gammerman and Shafer (Springer 2005; 2022)or[*alrw.net*](http://alrw.net/). There is also a gentle introduction by Angelopoulos and Bates (2021) that first appeared in [*arXiv*](https://arxiv.org/abs/2107.07511)and later was publishedin *Foundations and Trends in Machine Learning (2023).*

Among different approaches of conformal prediction, the **split conformal prediction (SCP)**is the most appealing framework that is*simple and easy* to implement. It is *model-agnostic*, which means it is applicable to arbitrary ML models. As illustrated below, for a pre-trained model with point prediction, it employs a hold-out calibration dataset to generate the prediction interval or prediction set with statistically *guaranteed coverage* of the true response.



Schematic Diagram of Split Conformal Regression

**Naïve SCP:**For a given pre-trained regression model, a vanilla choice of the conformal score is the absolute residual that measures non-conformality or dissimilarity between the true response and the predicted outcome:

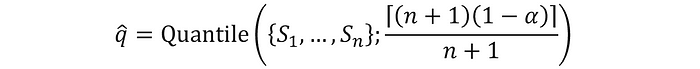


Conformal Score Function for Naïve SCP

Then given a proper hold-out calibration dataset of n samples, a test sample, and a pre-specified error rate α (say α = 0.1), SCP generally takes the following three-step procedure:

1. Compute the conformal score S\_i, i=1,…,n for each sample in the calibration dataset;

2. Calculate the calibrated score quantile with probability ⌈(n+1)(1-α)⌉/(n+1), where ⌈x⌉ is the ceiling function rounding x up to the nearest integer (e.g. ⌈90.9⌉ = 91)



Formula of Calibrated Score Quantile

3. Construct the prediction set/interval for the test sample. This can be done by solving the inequality to get the bounds for the unknown y.



Prediction Set by Split Conformal Prediction

Using the vanilla score, the naïve SCP constructs the prediction interval of the form:



Prediction Interval by Naïve SCP

**📌Coverage Guarantee**

Let’s explain a little bit the choice of probability ⌈(n+1)(1-α)⌉/(n+1) when calculating the score quantile in Step 2 above. This leads to the beautiful result of guaranteed coverage for the prediction interval.

First of all, let’s make an assumption of exchangeability among conformal scores.

**Exchangeability Assumption:**The random variables S\_1, S\_2, …, S\_n+1 (conformal scores for calibration and test samples) are exchangeable, i.e., their joint distribution are permutation invariant.

This is a weaker condition than i.i.d. (refer to [Wikipedia](https://en.wikipedia.org/wiki/Exchangeable_random_variables) for details). In most of time, let’s take it simply as the i.i.d. assumption. For machine learning prediction, when calibration and test samples (***x****, y*) are independently drawn from the same distribution, their conformal scores S(***x****, y, f^*) are i.i.d., where the third component *f^*is trained by a separate training data.

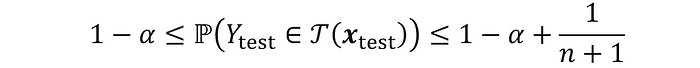
* Under the i.i.d. or exchangeability condition, the ranks of S\_1, S\_2, …, S\_n+1 are uniformly distributed over 1,2, …, n+1.
* The probability that S\_n+1 is among the ⌈(n+1)(1-α)⌉-smallest of the sequence is simply given by



* Let *q^ =*Quantile({S\_1, …, S\_n}; ⌈(n+1)(1-α)⌉/(n+1)) same as in SCP.
* By converting the rank to quantile, we can derive that



Thus, it’s justified that when the exchangeability assumption is valid, the prediction set constructed by SCP has statistically guaranteed coverage:



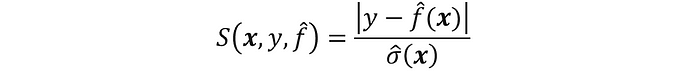
Coverage Guarantee for Conformal Prediction

When α = 0.1 and n=100, this probability bounds mean that the prediction set T(***x***\_test) covers the true response *Y*\_test with probability at least 90% and at most 90.99%. What an amazing result!!!

**📌Locally Adaptive Prediction Interval**

The prediction interval by naïve SCP has constant bandwidth 2*q^*everywhere, which provides an overall measure of prediction uncertainty. It is desirable to quantify the uncertainty with **local adaptivity**, so that less uncertain regions (prediction is relatively easy) have narrower bandwidth, while more uncertain regions (prediction is relatively hard) have wider bandwidth.

**Sigma SCP:** One natural way is to modify the vanilla conformal score by normalizing the absolute residual per sample, which leads to the so-called studentized residual in magnitude:



Locally Adaptive Conformal Score Function

The denominator σ(**x**) is a pointwise dispersion measure, e.g. mean absolute deviation (MAD) or standard deviation. An auxiliary model is needed to estimate σ(**x**) based on a hold-out dataset that is disjoint from training or calibration datasets. An effective approach to estimate MAD σ(**x**) is to train a Gradient Boosting Machine (GBM) with MSE loss on the absolute residuals obtained through the hold-out dataset:



Hold-out Dataset for Fitting MAD

Plugging the conformal score of studentized residual form into the SCP procedure, we obtain the locally adaptive prediction interval:



Locally Adaptive Prediction Interval by Sigma SCP

**📌A New Residual-Quantile Approach**

**CRQR SCP:** Quantile regression provides an effective way to construct the prediction interval. For a pre-trained regression model, we may construct data with residuals as responses



through a hold-out dataset, use it to fit a GBM with quantile loss for the lower and upper residual quantiles



Next, apply SCP to the residual quantiles through a disjoint hold-out calibration data, for which we suggest to use the following score based on the conformalized quantile regression (CQR) by Romano et al. (2020).



Conformal Score for Conformalized Quantile Regression (Romano, et al., 2020)

The calibrated score quantile *q^ =*Quantile({S\_1, …, S\_n}; ⌈(n+1)(1-α)⌉/(n+1)) can be calculated as usual. Plugging it into S\_n+1 ≤ *q^*, we obtain the prediction interval of the testing sample,



This is an elegant form of prediction interval as it consists of three interpretable components:

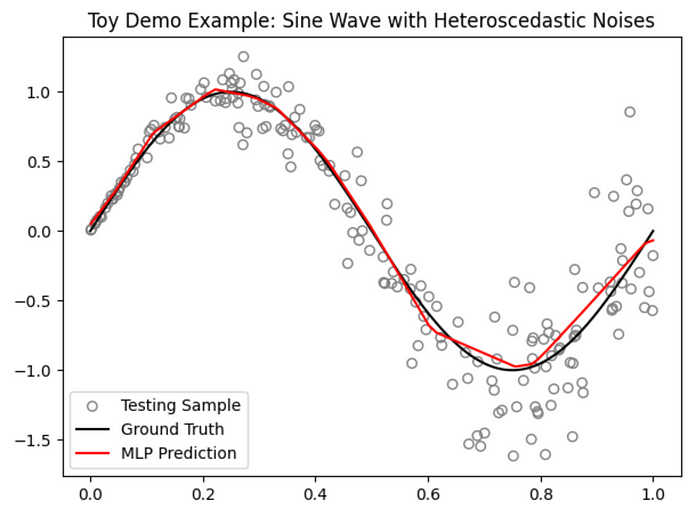
1. the original point prediction,
2. the fitted residual quantiles, and
3. the calibrated conformal adjustment.

Such **conformalized residual quantile regression** (CRQR) method is the one implemented by PiML toolbox since its first launch in May 2022.

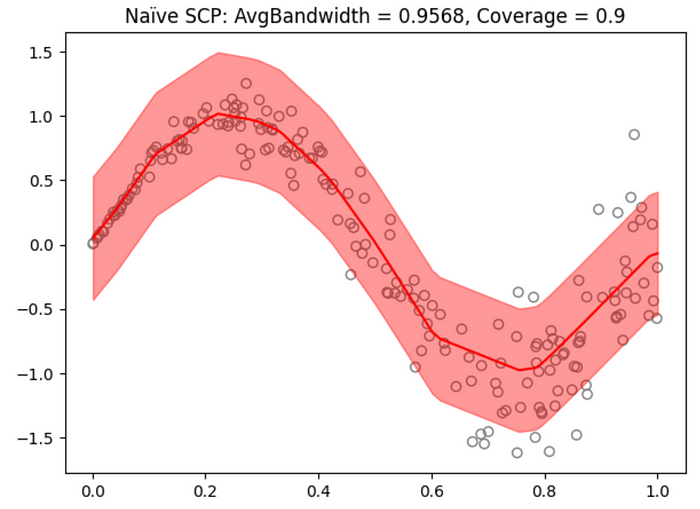
**🎯Toy Demo Example**

Let’s compare three different SCP methods by a simple toy example taken from Lei, et al. (JASA, 2018). It is a simulated sine function with heteroscedastic noises: y = sin(2πx) +|x|⋅N(0,1)⋅π²/10, with 1000 equal-spaced samples generated between 0 and 1.

Let’s randomly split the data into three sets: 60% training, 20% calibration, and 20% testing. Based on the training dataset, a neural network model (a.k.a. multi-layer perceptron (MLP)) is trained with 4 hidden layers each having 20 ReLU activation nodes. The point prediction on the testing dataset is shown below.

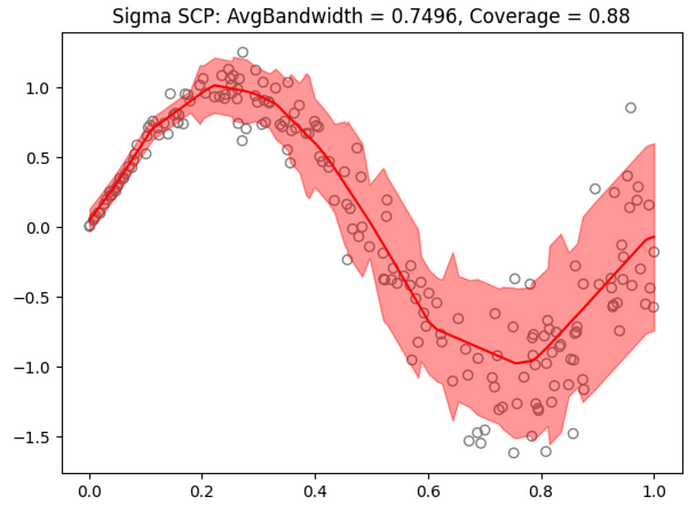


Based on the calibration dataset, the **Naïve SCP** method generates the constant-bandwidth prediction intervals on the testing dataset, with the average bandwidth 0.9568 and the coverage ratio 0.90.



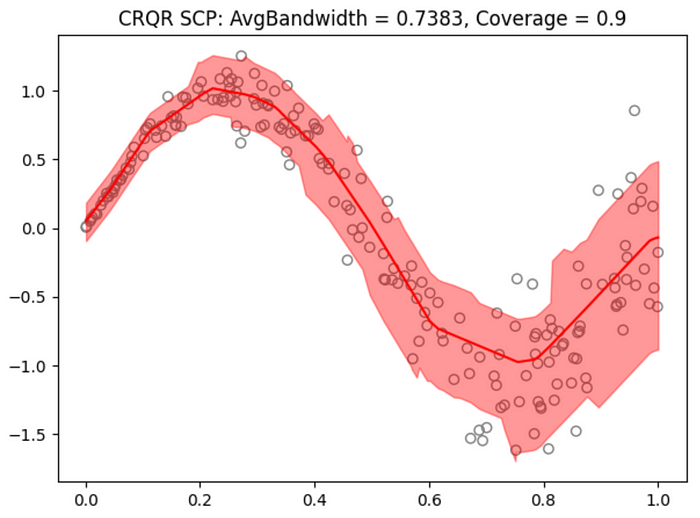
Prediction Band by Naïve SCP

To run the other two SCP methods, we need further split the calibration dataset to get a subset for fitting the auxiliary models. The **Sigma SCP** method generates the varying-bandwidth prediction intervals, with the average bandwidth 0.7496 and the coverage ratio 0.88.



Prediction Band by Sigma SCP

Finally, the **CRQR SCP** method also generates the varying-bandwidth prediction intervals, with the better average bandwidth 0.7383 and the coverage ratio 0.90.



Prediction Band by CRQR SCP

In this toy example, we used [*MAPIE*](https://github.com/scikit-learn-contrib/MAPIE) to run Naïve SCP and tweaked it to run Sigma SCP, and used [*PiML*](https://github.com/SelfExplainML/PiML-Toolbox) to run CRQR SCP. We share the Python codes by the end of this tutorial, plus some other useful resources for applied conformal prediction.

**🔬Unreliable Region Detection**

The aforementioned CRQR SCP method is implemented by PiML toolbox for reliability testing of regression models. PiML also supports unreliable region detection over the testing samples, by utilizing the slicing technique from [*a previous tutorial*](https://medium.com/@piml/model-diagnostics-trilogy-part-1-499371898b6a).

A practical user guide for unreliable region detection:

*1. Identify the features sensitive to prediction uncertainty (details below).*

*2. Perform segmented bandwidth analysis (i.e., slicing) according to the identified features.*

*3. Verify the diagnostic result jointly with weak spot and other PiML tests.*

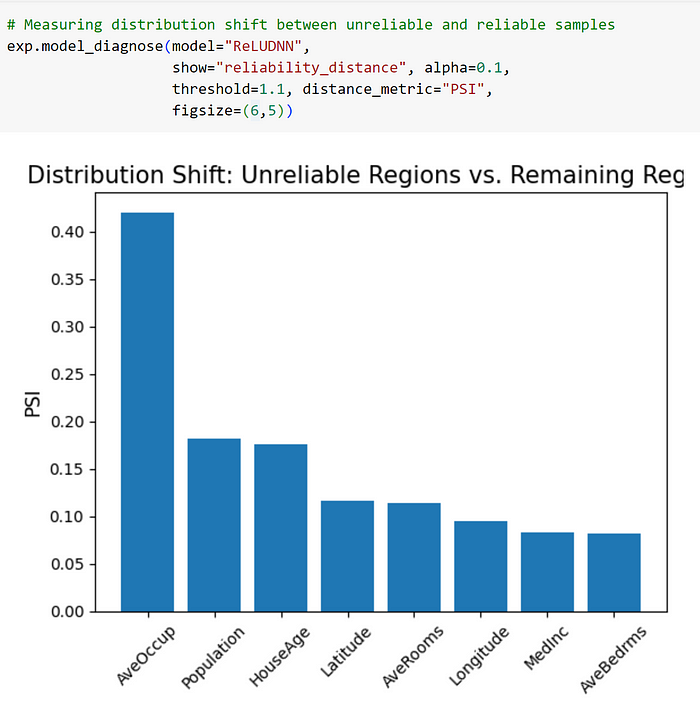
There are different approaches to identify the sensitive features to prediction uncertainty. PiML currently implements the distribution shift analysis between unreliable and reliable test samples, upon thresholding on the sample-wise bandwidths.

Let’s walk through PiML reliability testing through the CaliforniaHousing case study. See [*the previous tutorial*](https://medium.com/@piml/model-diagnostics-trilogy-part-1-499371898b6a)about data and model pipelines.

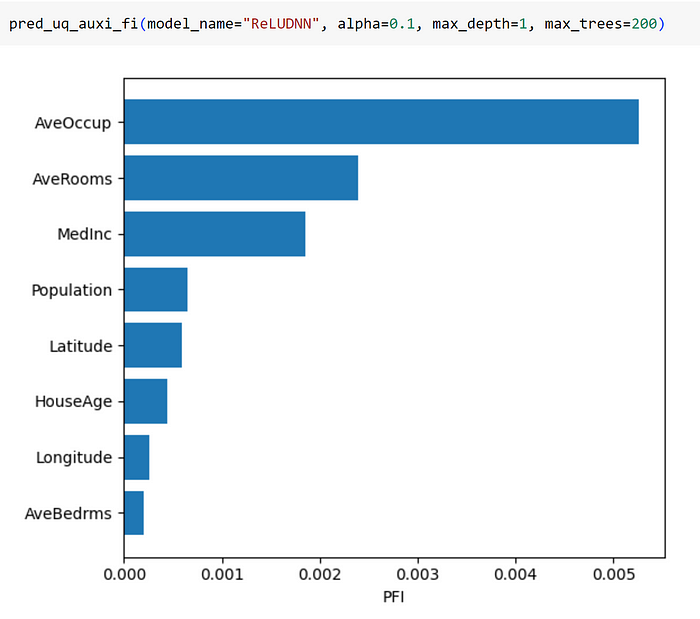
For the pre-trained ReLU-DNN model on the training data, PiML **exp.model\_diagnose** can be executed in the following way to show the reliability table of testing response coverage and average bandwidth:



Run **exp.model\_diagnose**with option “reliability\_distance” and a user-specified bandwidth threshold to divide the testing data into unreliable and reliable samples. The distance metric “PSI” would then measure the marginal distribution shift; revisit the definition of “PSI” in the [*previous tutorial*](https://piml.medium.com/model-diagnostics-trilogy-part-1-499371898b6a). According to the rank-ordered PSI values, the variable “AveOccup” is found to be most sensitive.

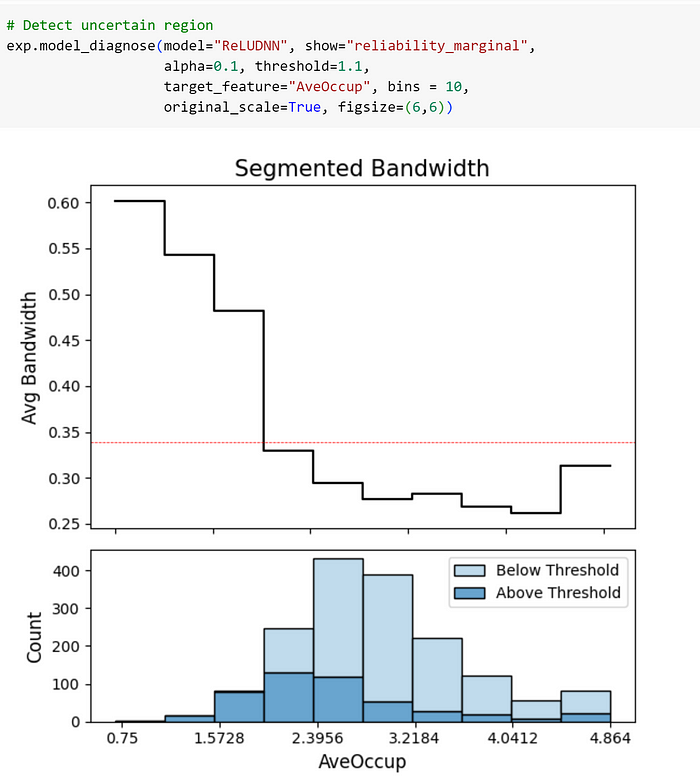


Alternatively, one can fit an auxiliary model (e.g. interpretable XGB1 or XGB2) on the bandwidth responses {(***x***, Bandwidth(***x***))} for each sample ***x*** in the testing dataset, then identify the important features by model interpretability or explainability methods. Below is an example result of using GBM of depth 1 and its variable importance plot.



It’s also found the “AveOccup” variable is most sensitive, but the orders of other variables do not match the distribution shift analysis. This is because the auxiliary modeling approach takes the continuous responses of bandwidths, and it models all the variables jointly. In contrast, the distribution shift is a dichotomization approach and the PSI is measured one variable at a time.

Next, run again **exp.model\_diagnose**with option “reliability\_marginal” to perform segmented bandwidth analysis. By specifying target\_feature = “AveOccup”, we can divide the marginal into ten buckets and evaluate the average bandwidth of prediction intervals per bucket. Thus, the unreliable regions can be detected if some buckets have significantly higher average bandwidth.



To verify the findings, we can cross-check the WeakSpot and resilience testing results again in [*the previous tutorial*](https://medium.com/@piml/model-diagnostics-trilogy-part-1-499371898b6a). The low “AveOccup” regions are diagnosed to have poor prediction with high uncertainty.

**🍭Python Notebooks**

We share the Python codes in Jupyter notebook formats that are linked through Google Colab:

* [*Toy Demo Example: Three Split Conform Prediction Methods*](https://colab.research.google.com/github/SelfExplainML/PiML-Toolbox/blob/main/docs/Workshop/202310-UNCC/Toy%20Demo%20Examples%20-%20Three%20SPC%20Methods.ipynb)
* [*CaliforniaHousing Case Study: PiML Reliability Testing*](https://colab.research.google.com/github/SelfExplainML/PiML-Toolbox/blob/main/docs/Workshop/202310-UNCC/ModelDiagPart2_PredictionUncertainty_CaliforniaHousing.ipynb)

There are other useful resources for applied conformal prediction. Check out the first link below for an updated and comprehensive list:

* <https://github.com/valeman/awesome-conformal-prediction>
* <https://github.com/scikit-learn-contrib/MAPIE>
* <http://alrw.net/>

**⚖️Caveats:**

We have introduced the benefits of using conformal prediction for machine learning uncertainty quantification. It also has some known limitations, including

* Prediction set can be effectively generated for regression and multi-class problems, but less informative for binary classification;
* The guaranteed coverage of true response is only in the unconditional or marginal sense, but impossible in the conditional sense.

**Thank you for reading!**

If you are, like us, passionate about data science, interpretable machine learning, and/or model risk management, please feel free to contact us on [**LinkedIn**](https://www.linkedin.com/in/ajzhang/).

*All images without a source credit were created by the author or generated by PiML.*

**About PiML***PiML was born as an educational toolbox for interpretable machine learning in Python community. It is designed for both model development and model diagnostics. Since its first launch on May 4th, 2022, PiML has continuously updated with new features and capabilities, together with a complete*[***user guide***](https://selfexplainml.github.io/PiML-Toolbox)*.*