Paper 1862-2018

How Good is That Forecast?

The Nuances of Prediction Evaluation Across Time

Aric LaBarr, Elder Research, Inc.

ABSTRACT

When predicting across time, typical methodologies of prediction evaluation no longer hold true. It is not practical to take a hold-out sample randomly from observations in the data set or even use a typical k-fold cross-validation structure. Even newer methods of prediction evaluation in cross-sectional data like target shuffling should not just be applied to data where a temporal structure is inherent. How then can we determine if we have a good forecast? This talk highlights advantages and disadvantages to techniques evaluating predictions when forecasting future observations. It also discusses possible biases arising from time structures of data that should be considered.

INTRODUCTION

How can you tell if your model has produced a good forecast? Proper model evaluation is key to understanding the validity of any forecasting model technique. Typically, data falls into one of two categories – cross-sectional or time series. Cross-sectional data is the data that most people work with. It is data that is measured either at one single point in time or without consideration of any temporal effect. A classic example of this would be to study students in a statistics class and try to relate the number of hours studied on their upcoming statistics test and their subsequent grade on that test. Time series data on the other hand typically measures values across time to understand the evolution of those values as time progresses. To juxtapose the previous example, we might be interested in a particular student's grades on tests across a series of statistics classes to see if there is a discernable, temporal pattern.

All modeling techniques need validation regardless of whether they are cross-sectional or time series based. There is a difference between model accuracy and model validation. Accuracy deals with a models ability to predict data it was built on. Model validity reveals how a model predicts data which it has not seen before. Model validation should reveal some notion of how believable the results of a model process are. Without proper model validation, future effectiveness of a developed model is unknown which poses a problem when trying to sell your model to a decision maker for implementation. Perhaps even worse, if an improper technique is used for model validation, decision makers might no longer trust further models you produce.

Models built with cross-sectional data or time series data require different model validation techniques because of the inherent temporal structure of time series data. First, this paper will go through common model validation measures that can be used in both cross-sectional and time series modeling. Next it highlights techniques for models developed with cross-sectional data followed by the problems of using these techniques with time series models and laying out better validation techniques for time series based models. Lastly, the paper will cover inherent biases from time series data that needs to be considered when building a model for any of the validation techniques to be sound.

MODEL EVALUATION

Before you apply validation techniques, you need to look over measures of validation and accuracy to compare and evaluate your models. There are a variety of different ways of measuring model validity. This paper does not assume that is creates an extensive list of validation measures, but here are a few common measures based on a continuous response target since we compare against time series models where targets are most often continuous. We will discuss two different types of validation measures – scale dependent and scale independent.

SCALE DEPENDENT

The first two validation (or accuracy) measures depend on the scale of the data – the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). The MAE is the average of the absolute differences between your prediction and the true value:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|Y_t - \hat{Y}_t\right|$$

The RMSE is the estimate of the standard deviation of your model because it is the square root of the average of the square deviations from the truth your predictions are:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2}$$

These measures should be used when you compare models base don data with the same units because they are scale dependent. The MAE is much easier to interpret as it just how far off your estimates were. Being the standard deviation of the model, the RMSE is harder to interpret, but still widely used.

SCALE INDEPENDENT

The problems with scale dependent measures are exactly that – they depend on the scale of the data. With that restriction you cannot compare models across data sets measured on different scales. Two common scale independent measures are the Mean Absolute Percentage Error (MAPE) and the Symmetric Mean Absolute Percentage Error (sMAPE). The MAPE is a variation on the MAE with the scale removed:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right|$$

Unlike the MAE though, the MAPE is not a symmetric measure of validation. It overweights overpredictions in your model. This can be seen best through the example in Table 1. To adjust for this, albeit not completely, there is the sMAPE:

$$\text{sMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|Y_t - \hat{Y}_t\right|}{\left(|Y_t| + |\hat{Y}_t|\right)}$$

The sMAPE is more symmetric than the MAPE, but not completely symmetric.

	$Y_t = 1$, $\hat{Y}_t = 3$	$Y_t = 2$, $\hat{Y}_t = 3$	$Y_t = 3$, $\hat{Y}_t = 3$	$Y_t = 4$, $\hat{Y}_t = 3$	$Y_t = 15$, $\hat{Y}_t = 3$	MEAN
AE	2	1	0	1	12	3.2
SE	4	1	0	1	144	30
APE	200%	50%	0%	25%	80%	71%
Sym. APE	50%	20%	0%	14.3%	66.7%	30.2%

Table 1: Examples of Validation Measures

Any of these validation measures can be used with the following model validation techniques to understand how your model would react in a real deployment setting.

CROSS-SECTIONAL PREDICTION EVALUATION

TRAIN VS. TEST DATA

When choosing models, it is common practice to split data into multiple pieces. This allows validation of a model on data which the model was not built on. Without this validation, models can be built to maximize accuracy with no regard to usefulness in a new setting – typically called overfitting. You randomly sample the whole data set to remove certain observations to set aside. These set aside observations are called the test set or a hold-out sample and no analysis is performed on them. Figure 1 below is an example of this.

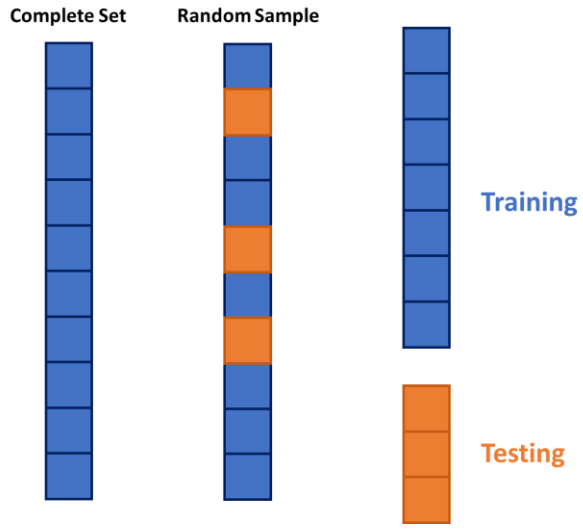

Figure 1: Training vs. Testing for Cross-sectional Data

You can always develop a model with more accuracy by adding more parameters or variables to your model. In fact, the perfect model can be obtained by using a variable for every combination of input values to a model. Take Table 2 as an example.

In Table 2, the perfect model is developed by creating parameters for all combinations of inputs – Males who are 35, Males who are 41, Females who are 28, and Females who are 35. What if a new observation needed to be predicted for a Female who is 31? The model described couldn't predict them.

Training data is used to develop the modeling technique as well as the parameters and variables used in the model. Test data is used for validation of the model developed on the training set. Just because a model is accurate on the training data, doesn't mean that it predicts new data well.

When evaluating a model using the test data, one consideration is to not repeatedly test a model, go back to build again on the training and then retest on the test data. Repeating this approach multiple times leads to overfitting the test data set, which still doesn't allow an accurate measure of how good a model does on new data. To protect against this, another approach is to split data into three sections – training, validation, and testing. In this case, the extra validation set allows for better model selection and training.

Observation	Gender	Age
1	Male	35
2	Male	35
3	Male	41
4	Female	28
5	Female	35
6	Female	28

Table 2: Perfect Model Creation Data Set

Common percentages of your total data to split among the two or three samples varies. Some common percentages in order of training, validation, testing are 60-20-20, 70-20-10, 40-40-20, 40-30-30, etc. In all the different recommendations that exist, a strong majority of the data resides in the training and validation splits, while leaving a smaller piece for testing at the end.

CROSS VALIDATION & BOOTSTRAPPING

What if the piece of validation or test data that you set aside contained important information that you would have preferred to build the model with or you don't have enough observations that you feel you

could split out into multiple data sets easily? In either of these scenarios, k-fold cross validation or bootstrapping is a possibility.

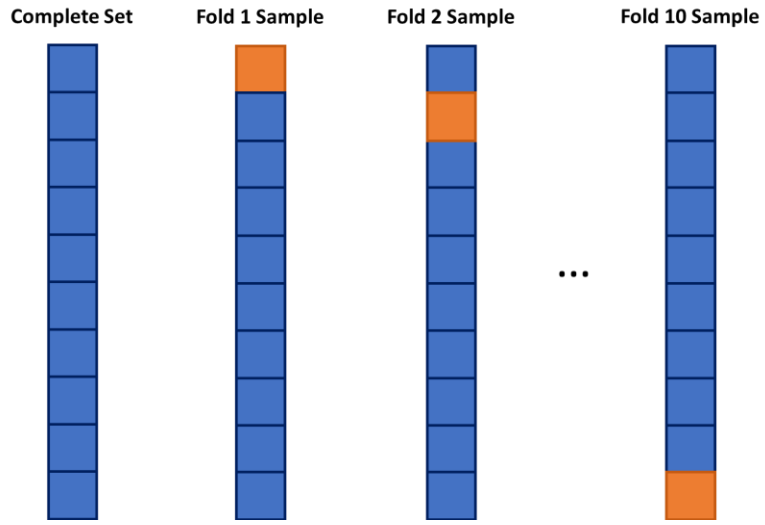

Figure 2: 10-Fold Cross Validation Structure

K-fold cross validation occurs when you split your data set into k equal pieces and use k-1 pieces of the data to build your model and the k^{th} piece to validate your model. However, once this process is complete, you switch out which of the k pieces you have for validation and repeat the process with the other k-1 pieces until you have gone through the process k times and every equal piece was used for training and validation. For example, the common 10-fold cross-validation is diagramed in Figure 2. In this example, you split your data into 10 pieces and rotate out 1 piece (10% of your data) for validation and use the remaining 90% of your data for training until the you rotate through the whole process. You can select k to be as large as the sample size itself in a special case of cross validation called leave-one-out cross validation where every observation is held out to form the validation data set. This process is much more computationally burdensome, which is why people settle for 5 or 10 fold cross validation.

K-fold cross validation is perfectly used to determine the validity of a model more than to select a model. A model can be built on the entire data set, and then the process is repeated across all the samples (or folds). From each fold, you estimate the model's validity based on the validation data set. Now you have a collection of k measures of accuracy. Average these k measures of accuracy together and you obtain an estimate of the validity of the modeling process.

You could always test multiple modeling methods across the folds and use their average validity to select which modeling process is better for your data. However, this potentially overfits your data since you used the validation data sets to select the modeling process. This wouldn't be as bad if you combined the training vs. testing approach with the k-fold cross validation. You could essentially build a k-fold cross validation within the training sample to select the best type of model and then test that model's final validity against the test sample set aside at the beginning.

The previous techniques force independent samples. Bootstrapping is a technique that does not. Bootstrapping has you draw a sample the same size as your data set, but with replacement. Essentially, you could have some observations repeated as well as some observations left out completely. The goal is still to evaluate a "new" sample that is similar to your data set, but not the exact same. From there the process is the same as k-fold cross validation. You average the measures of validity across your bootstrapped samples to get an estimate of the validity of your modeling approach.

TARGET SHUFFLING

A more recent approach to evaluating the validity of a model is the process of target shuffling. Typically performed after model selection is complete, target shuffling tries to simulate what statistical tests were designed to do when they were first invented – identify how likely results from a model occurred due to random chance. Here is the process:

1. Randomly shuffle the values of the target variable, while leaving the input variable values in the same location. This removes any possible relationship between the target variables and the inputs.

2. Repeat the model building process (preferably in an automated way) to identify any possible relationships between the input variables and the newly shuffled target variable.
3. Save the “best” model’s measure of validity – RMSE, MAPE, AIC, etc.
4. Repeat the process thousands of times.
5. Look at the distribution of the collection of validity measures from each iteration.
6. Evaluate where your original model’s validity measure falls on this distribution of validity measures.

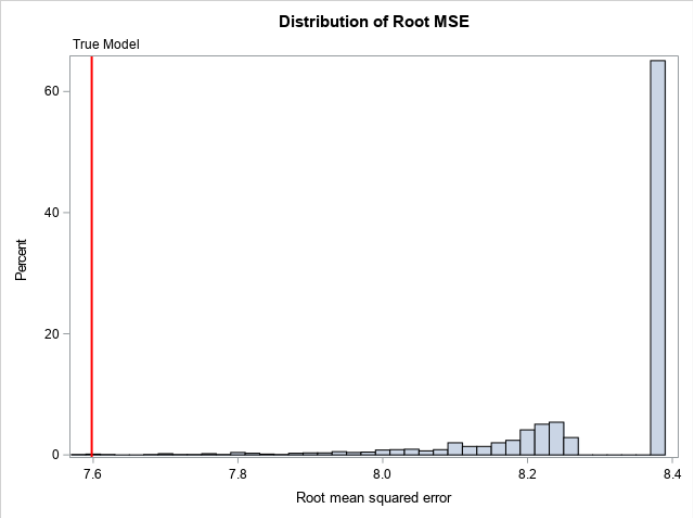

Figure 3: Target Shuffling on RMSE

Figure 3 is an example of the target shuffled distribution.

As you can see from this distribution, our initial model is far to the left of the distribution, but not the “best” model. In fact. two models performed better than the original model in the realistic data situation. How is this possible? Random chance. All of the other models were built off of data that was randomly ordered. There should have been on relationships, but some appeared by shear luck. However, we can now say that our results have a 0.13% chance of occurring randomly (2/1500) – exactly what statistical p-values try to summarize. Essentially, target shuffling is a simulated version of permutation testing or exact p-values. In permutation testing, you would have shuffled the target variable values until every permutation of possible models would have been created. Through simulations we can do this approximately with thousands of iterations.

TIME SERIES PREDICTION EVALUATION

END OF DATA HOLD-OUT SAMPLE

In time series, the inherent structure of the data changes how we can validate models. Typical methods of randomly sampling training vs. testing as well as k-fold cross validation don’t adhere to time series data. Cross-sectional models attempt to predict values of data which the model has seen before, irrespective of time. Time series models attempt forecasting of future values of data, typically using past values of data.

Figure 4: Time Series Hold Out Sample

Therefore, they cannot randomly sample from different points in time as the evolutionary structure of the data will be broken. Instead of randomly sampling observations in the data, we sample the values at the end of the data set temporally as seen in Figure 4 below.

Much like with cross-sectional data, there are no set percentages of data that we hold out. The minimum amount of data that we hold out is typically the minimum we need to forecast. If we want to forecast next month’s daily sales, then we should have at least a month of daily sales in the test sample. Like the cross-sectional data validation, a measure of validity is calculated for the test data set.

ROLLING HOLD OUT SAMPLES

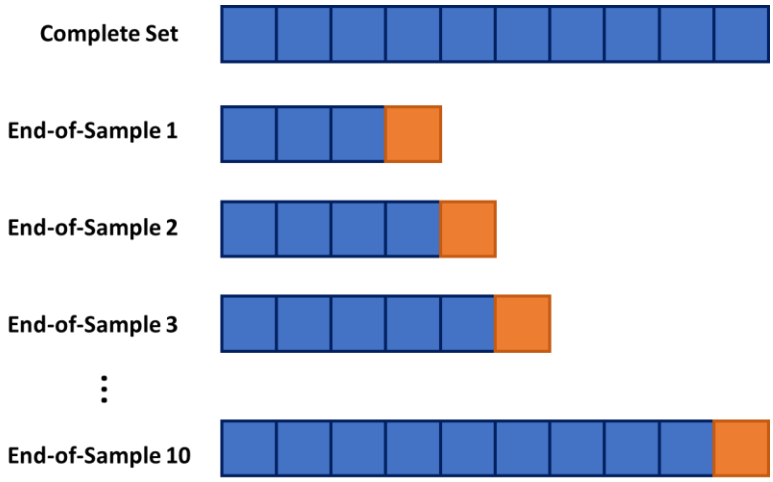

Figure 5:Time Series 10-Rolling Window Example

Time series data has its own version of k-fold cross validation – a rolling window validation. Again, time series data depends on its temporal structure, so randomly sampling percentages must be at the end of the series. If the data series is long enough, you can build many models while continually adding more observations at the end of the series and always predicting further down the series. Figure 5 below displays this approach.

You can have rolling hold-out samples as small as one observation, but ideally your desired forecast length should be the length of the rolling window. Using the same example as before, if you desire to forecast a month of daily sales data, your rolling window should ideally be at least a month wide.

TARGET SHUFFLING IN TIME SERIES?

The process of target shuffling in time series is a much more difficult and even controversial topic. Is target shuffling even possible with time series? The value and process of target shuffling is to remove the structure between the target and the inputs that predict it by shuffling the target and leaving the input values alone. However, since time series models typically use the previous values of the target variable, shuffling the target involves shuffling the inputs as well.

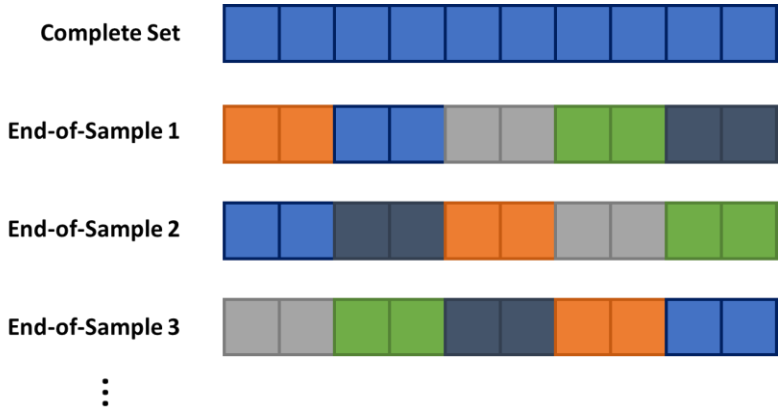

Figure 6: Bootstrapping a Time Series

Instead of shuffling individual values of the target variable, those who have proposed target shuffling-like approaches suggest shuffling windows of the target variable. Essentially, you shuffle around pieces of the series to form a new series and still try to build time series models on. Some even refer to this as bootstrapping the time series data since you are creating a new data set similar to, but not the same as, your original data set. This process is repeated many times over as seen in Figure 6.

From these many “samples” you calculate the validation measure to build a distribution of possible validations and compare where your original distribution measure falls on this distribution. As you can see, the temporal structure of time series adds complication to any of the validation processes that are common in cross-sectional data.

INHERENT TIME SERIES BIASES

All of the validation techniques above help you avoid problems and overfitting in your model to some extent. However, when measuring things across time you must also consider some possible inherent biases that easily occur due to the temporal structure of the data. The biggest possible bias in time series modeling is look ahead bias.

Look ahead bias occurs when you are using information to build a model that you wouldn't have known at the time your model was actually being used. For example, you are forecasting energy usage for a power plant. You have historical, hourly energy readings along with historical, hourly temperature readings since you believe that energy usage is highly related to temperature. You build a time series model that uses past values of energy usage, but also current temperature. Therefore, in the forecast you would need future values of temperature as seen in Figure 7. If you build a model on historical data and in your validation, you have actual temperatures you use to build your forecasts you have committed look ahead bias. In reality, you do not have future values of temperature and only estimates. In your validation you must also use forecasted values of temperature to get an accurate forecast. None of the previous techniques discussed in the paper would have accounted for the look-ahead bias in this example. In fact, your historical temperature data would probably be best substituted with historical forecasted temperatures to ensure that you are not biasing your model. However, this may not be possible to find so easily.

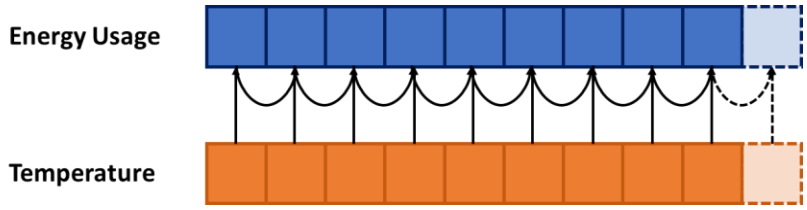

Figure 7: Look-Ahead Bias Example

Validation techniques protect you, but you must actively think about how your model is deployed and used when you validate it. Validation is trying to help you understand how your model works in deployment, so all aspects of deployment must be considered.

CONCLUSION

Validating models is not only important but expected. To accurately explain how your model performs in deployment, validation is needed. There are many different techniques to validate data in both cross-sectional and time series based modeling. All of these techniques are trying to better help protect you against yourself in the modeling process.

Time series data's temporal structure adds complication to all of the validation techniques mentioned above. The inherent interdependency of time series data means you need to consider different approaches and techniques to accurately validate. The best way to always ensure that you are validating your model is to try and create the best placebo for actual deployment that you can. How is your model used? Build your validation to repeat that process as close as possible!

RECOMMENDED READING

- *Handbook of Statistical Analysis and Data Mining Applications*
- *Forecasting: Principles and Practice*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Aric LaBarr, Ph.D.
Elder Research, Inc.
aric.labarr@elderresearch.com
https://www.elderresearch.com/company/our-team/aric-labarr