Train Test Split

Neil Liberman

Introduction

When building a predictive model we want our model to do two main things. **Predict well on data we have** and **consistently predict well in other samples**.

While it may appear being good at predicting the data we have will translate to predicting well in another sample, this is not always the case. In fact, the more specifically we try to cater our model to data we have, the more we risk not being effective at predicting other samples. This is called overfitting.

...Why does this occur?

Overfitting

When we build a predictive model, we use data we have to build the model. However, our sample is not a perfect representation of the population as a whole.

Thus:

> We may get overly specific to explain connections in our data that don't truly exist in the real world and won't predict as well on new samples. This is called overfitting.

Understanding Error

We have two major types of error. Error due to bias and error due to variance.

> Error due to bias: How far off our predictions are in a given sample from their actual value.
>
> Error due to variance: The difference in our model's effectiveness from one sample to another.

These two are inversely related. When one goes up, the other will likely go down. However, they don't move completely synchronously. Thus, finding the right balance will help us achieve our goal in minimizing the sum of those two errors.

Finding Balance

We need to describe our data thoroughly in order to build a useful model which limits error due to bias, but in being too specific we run the risk of increasing error due to variance.

How do we handle this issue?

What is Train Test Split?

A Methodology used in building predictive models that allows us to build a model on part of the data (“training data”) and test the model’s effectiveness on the remaining data (“testing data”).

Our goal is to predict well on data we don’t have not on data we do have. Thus, it makes sense to test on data we don’t have in the model. That is why we never use testing data to build our model.

How should we split the data? Most of the time, a simple random sample is sufficient.

What are the Implications of TTS

Train Test Split sacrifices our model's ability to predict the data we have (error due to bias). This is because we won't score the model's effectiveness on the data used to construct the model and we have less data to build our model on.

However, it should improve our model's ability to predict effectively and more consistently on new samples.

Problems with Train Test Split

The more data we have the better and train test split implies we aren't going to use all of our data.

K Folds Cross Validation

With train test split we lose part of our data. However, K folds CV is a type of train test split model that prevents us from losing any data.

We do this by splitting our data evenly into groups, called folds. The number of folds we use is our own decision and denoted by K.

If we have a dataset of 100 observations and use 5 folds, each fold would contain 20 observations. No data can be used in more than one fold, and as to which fold the data falls into should be random.

How Many Folds Should I Use?

Low Number of Folds:

> Low error due to variance and higher error due to bias.

Large Number of Folds:

> Low error due to bias and higher error due to variance.

The tradeoff is your choice but for large datasets K=3 is a reasonable number. For smaller datasets, you will want to use a larger number of folds.

How Does K Folds CV Work?

We train our model on every fold but one which we holdout as our testing set. We build the best model we can and test how well it performs on the testing set.

We do this K times, each time creating a new model and testing on a different fold. By the end of the process we will have built K models and tested on every fold exactly one time.

This allows us to train on all of our data while still having a reliable way of measuring our model's performance.

Real World Implementation

Working for the CDC we are tasked with predicting the possibility of a West Nile Virus outbreak in a given area. We will need features to build our model on. I chose to look at the bird population, the mosquito population, the proportion of developed land, and temperature.

I will look at past data, build a model on that data, and test how well the model did on that data. However, that will overstate how predictive my model is.

Real World Implementation

Instead I opt to use Train Test Split on my model so as to improve the effectiveness and consistency of my model on new samples. I would randomly split my sample into a number of folds of my choosing.

Running TTS

Suppose the percentage of developed land was very strongly correlated with incidence of WNV while the other features were much less correlated.

We may want to safeguard from the possibility of our training data having a large percentage of observations where % of developed land was low and our testing data having a small percentage of observations where % of developed land was low. This may lead us to believe our model wasn't performing well when we applied it to the test data.

Stratified Random Sample

A random sample will most likely prevent this clustering from happening but we may choose to use a stratified sample just in case.

A stratified sample is where we divide our data into groups and then randomly select a certain number of observations from each group.

Real World Shortcomings

Train Test Split Model works under the assumption that future data is exchangeable with past data. Thus, if this is not the case, we will likely have a deficient model despite our splitting our data into training and testing sets.

An example of this is trading on the stock market. We can implement a train test split model but this isn't going to make our model a useful one.

Conclusion

Train Test Split is a very useful way to lower our model's error due to variance and have a better representation of what our model's performance will be in new samples.

We should be cognizant of the possibility that certain observations can cluster disproportionately between our training and testing set.