**AAQuickML**

AAQuickML is a Python ML app built and used by Advanced Analytics team with a unified interface for multiple learning tasks. The interface is built on Streamlit. The goal is to improve the model selection efficiency and take full advantage of AutoML. It currently supports:

* Time Series data Exploratory Data Analysis (EDA)
* Time Series regression (Arima, SARIMAX, Prophet, LSTM, GRU)
* Time Series validation (Train-test, multiple train-test, work-forward)
* Time Series auto piloting

AAQuickML provides a fast way to pick the modeling algorithm that is the most suitable for your dataset. It includes traditional statistical Time Series algorithm ARIMA, as well as some more recent algorithms such as Prophet by FB and Recurrent Neural Networks such as LSTM and GRU.

For Streamlit, see:

<https://www.streamlit.io/>

For ARIMA models, see:

<https://sktime.org/>

<http://alkaline-ml.com/pmdarima/0.9.0/index.html>

For Prophet by FB, see:

<https://facebook.github.io/prophet/>

For Recurrent Neural Networks, see:

<https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM>

<https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU>

<https://www.tensorflow.org/guide/keras/rnn>

1. **Initialization**

The app can be initialized by the following command. The host can access the app by the first URL, and people in the same Network can access the app by the second URL.

streamlit run “P:\Python\PythonCode\AAQuickML\Mike\Streamlit\app\_v1.0.py”

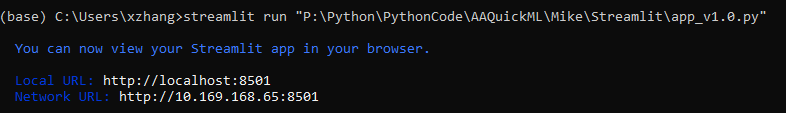
If you wish to be the host, you will need to install streamlit, fbprophet, sktime, tensorflow, and other required Python packages. Below are some commands for quick anaconda set-up.

pip install sktime

pip install fbprophet

conda install -c conda-forge tensorflow

pip install streamlit

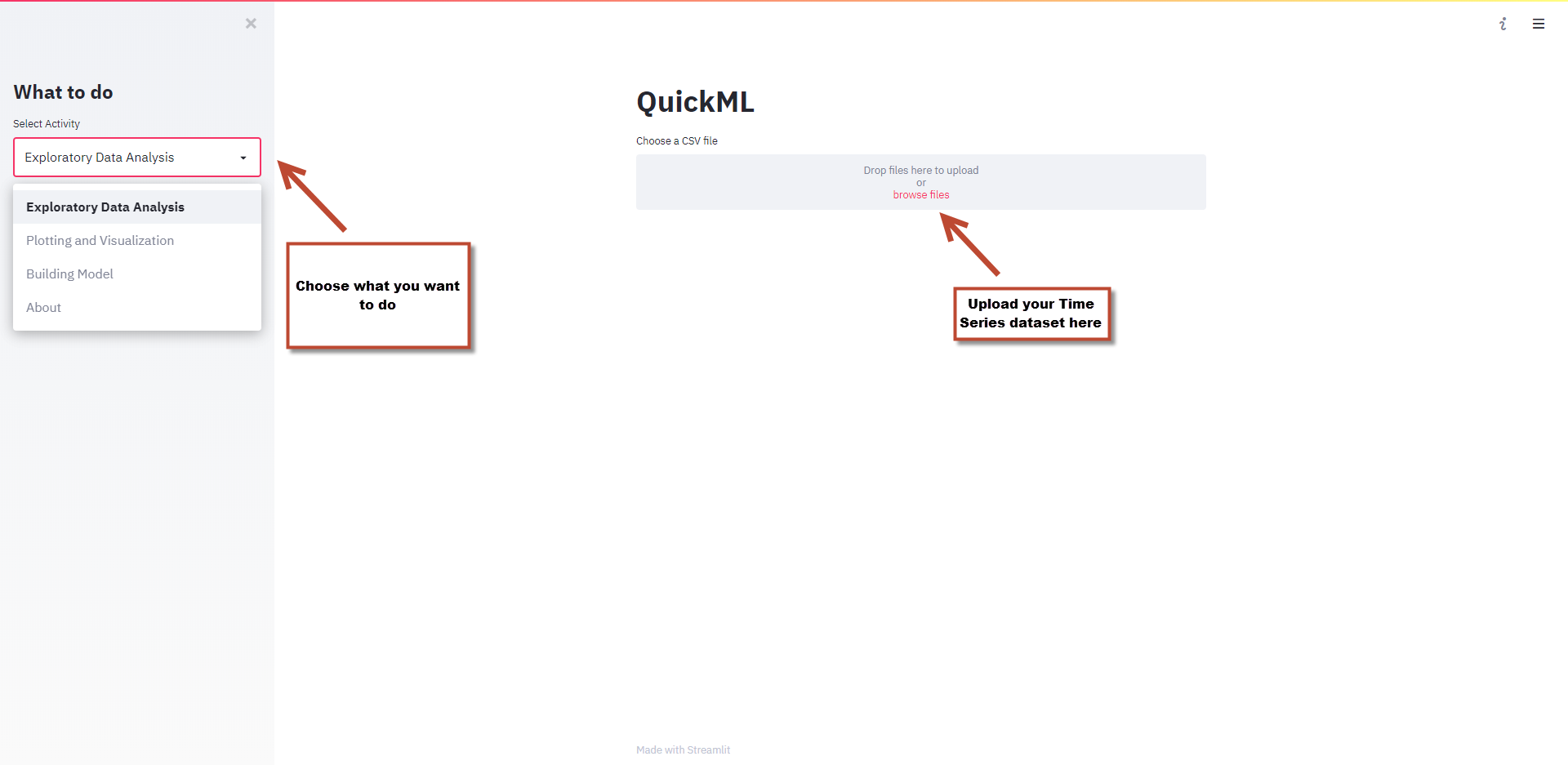


1. **Interface**

Three main functions so far are **EDA, Visualization, and Modeling**. EDA and Visualization was mostly developed on <https://github.com/Alro10/streamlit-time-series>. The modeling part is the core of this app.

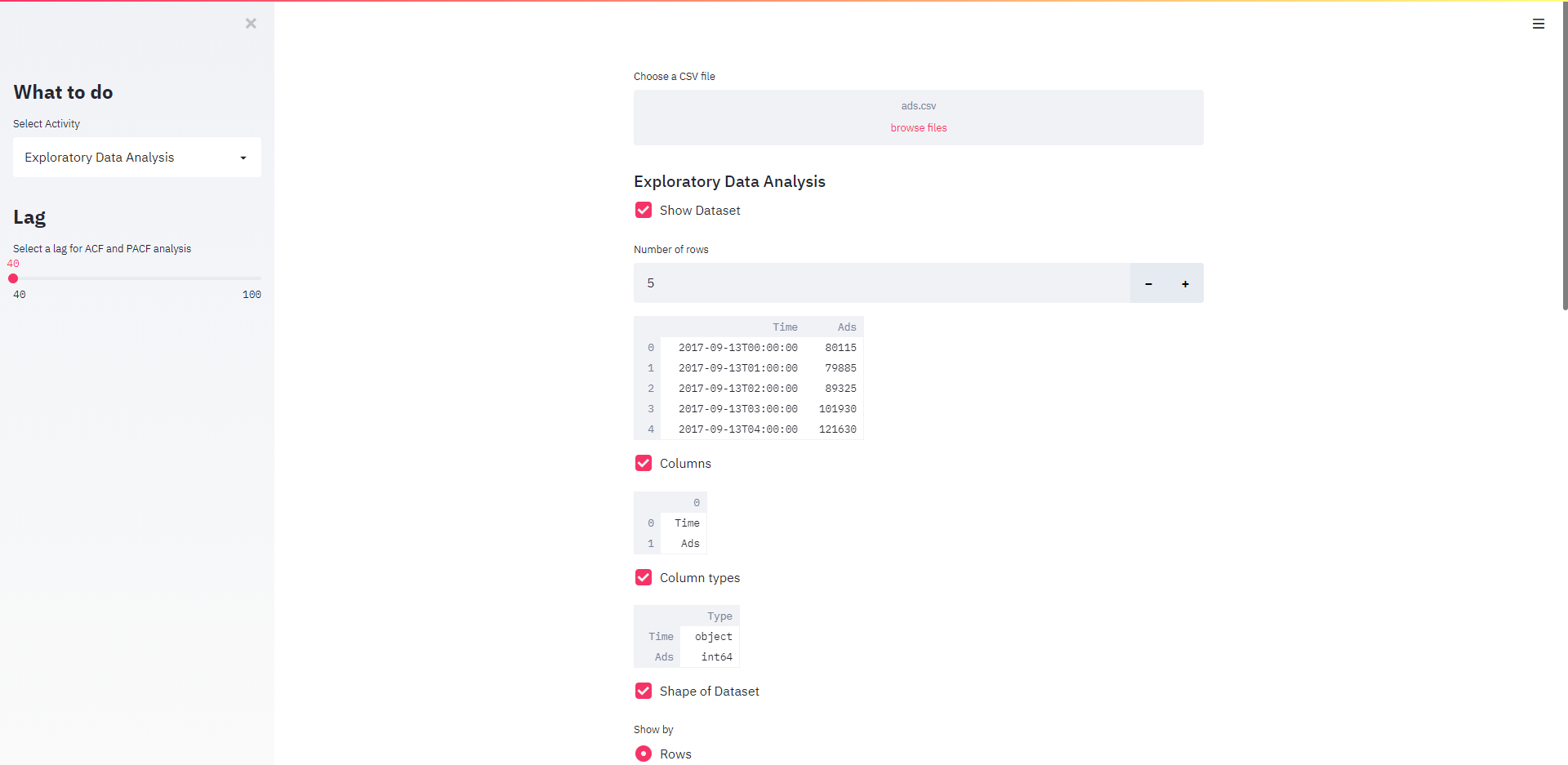
The central of the app is where the most of work is going to take place, while the side of the app is where most of essential parameters need to be manually set.

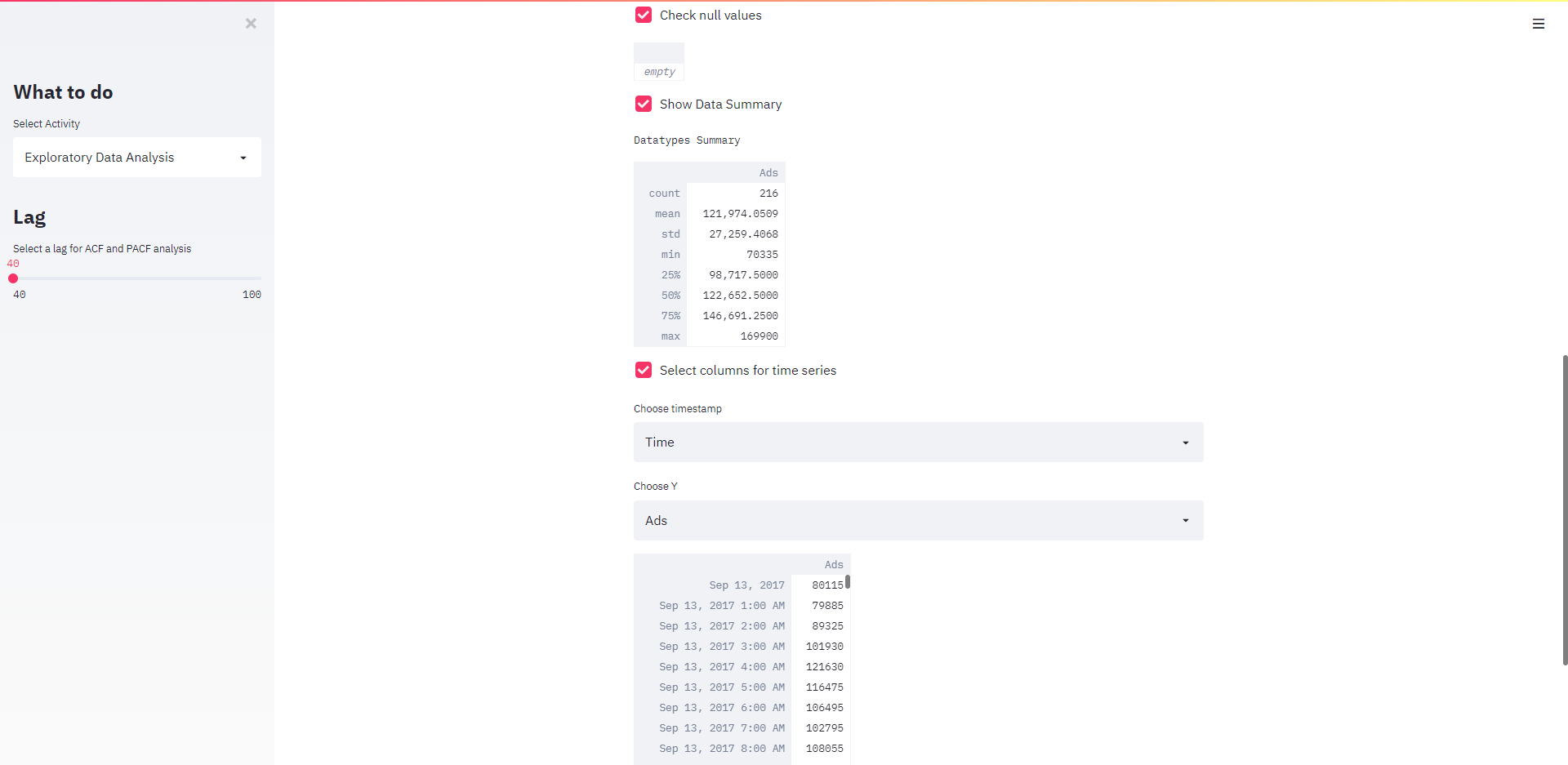
On the welcome page,

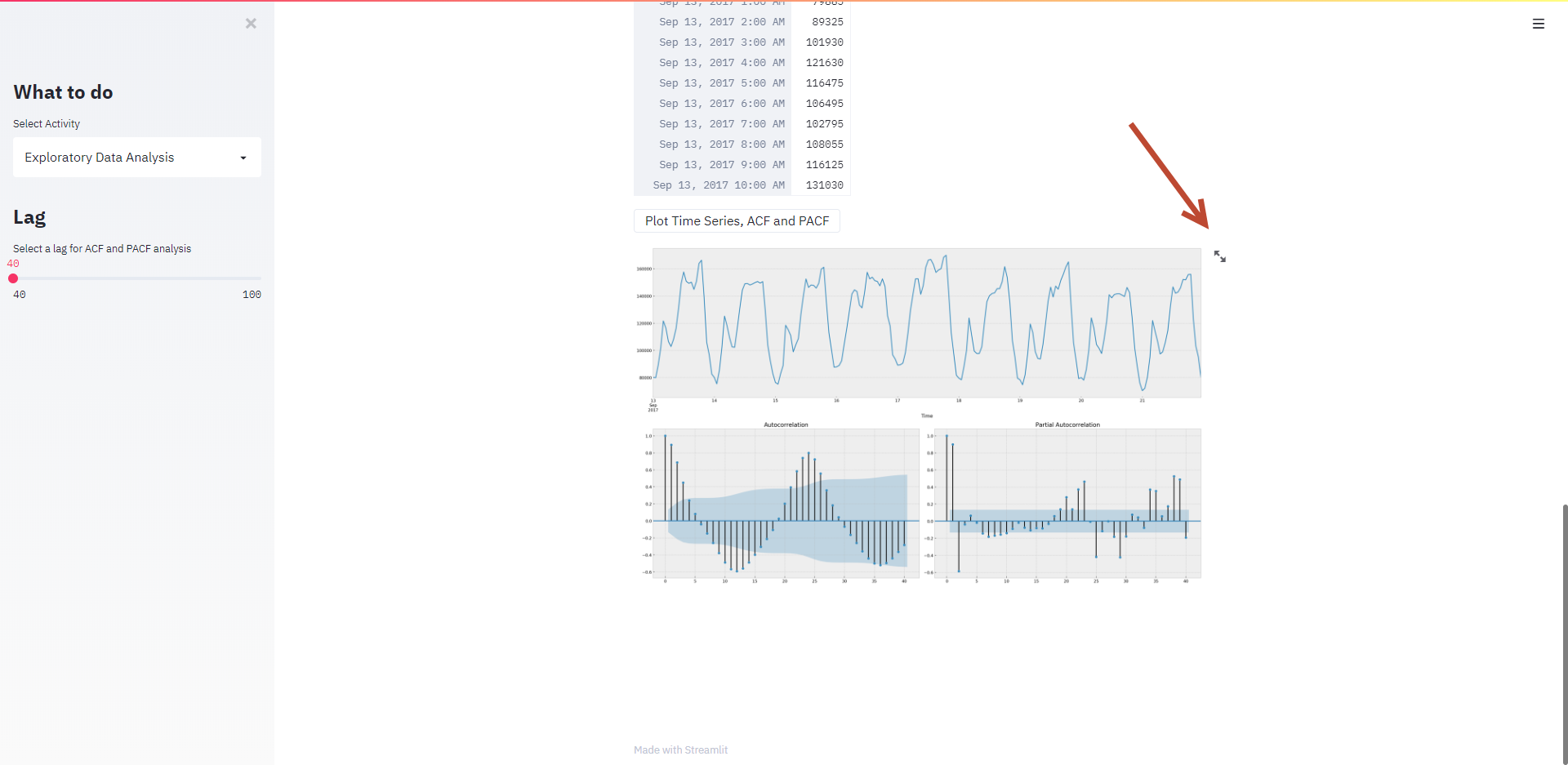


1. **EDA & Visualization**

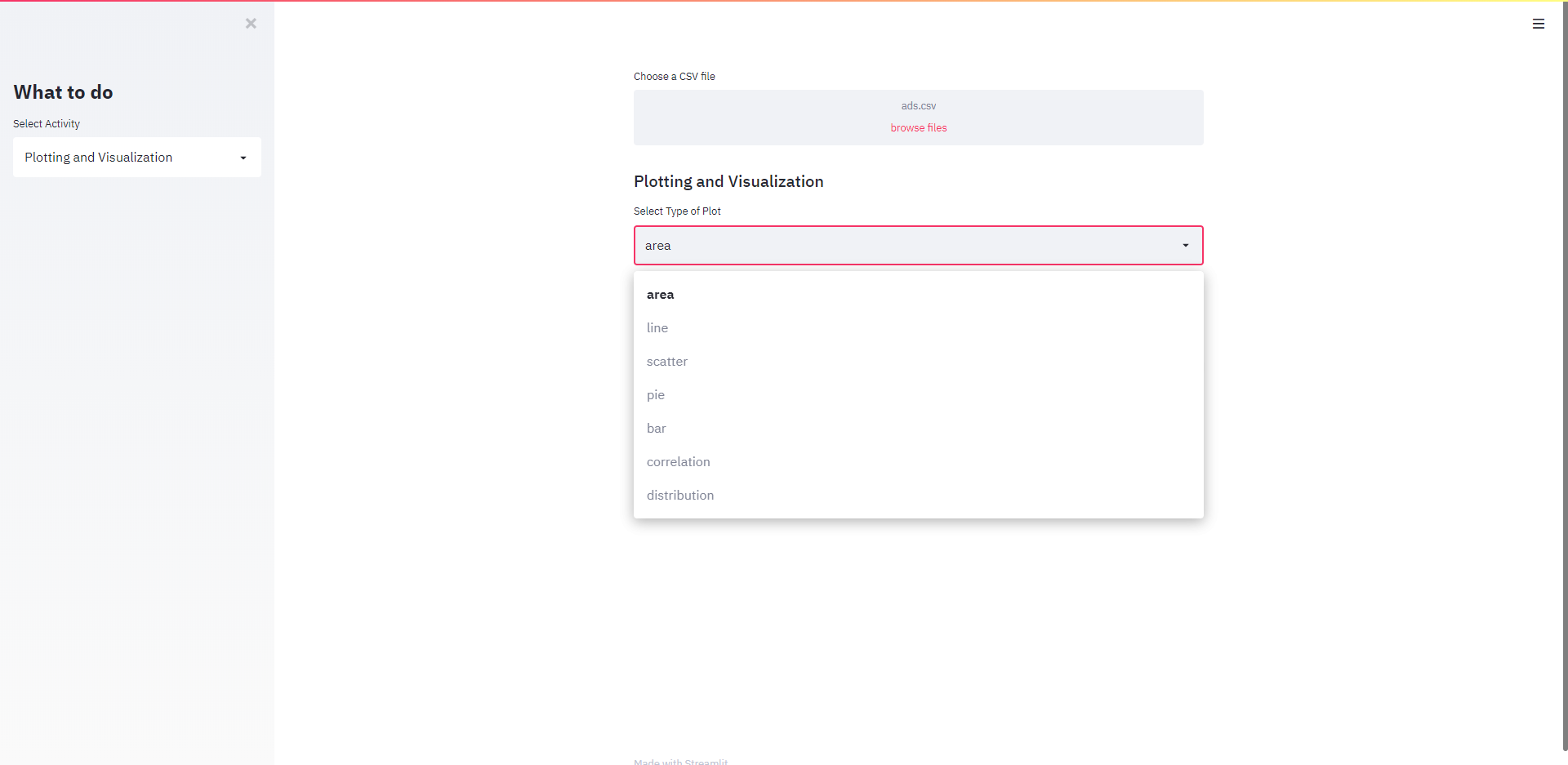
If you choose EDA or Plotting and Visualization, you will be able to explore and visualize your data. When you are in EDA, you will be able to choose number of lags on the side, which basically controls the length of horizontal axis for your ACF and PACF analysis. ACF and PACF will help you determine parameters for ARIMA model. However, the parameter tuning process has been automated in this app, so you don’t need worry too much about it. If you wish to make your charts bigger, you can let your mouse pointer stay on the chart for a sec and a full screen button should appear on the top right corner of the chart.

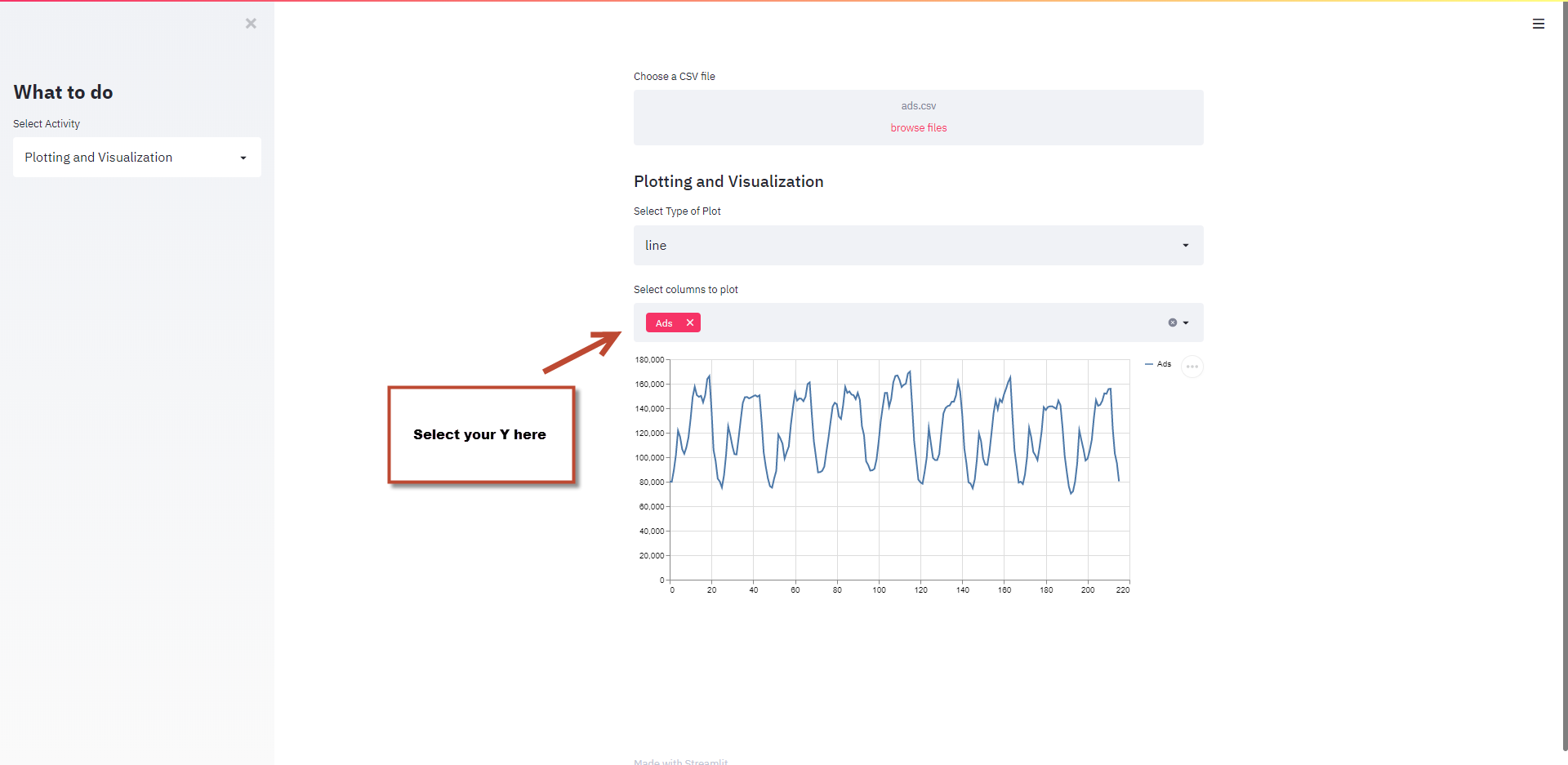






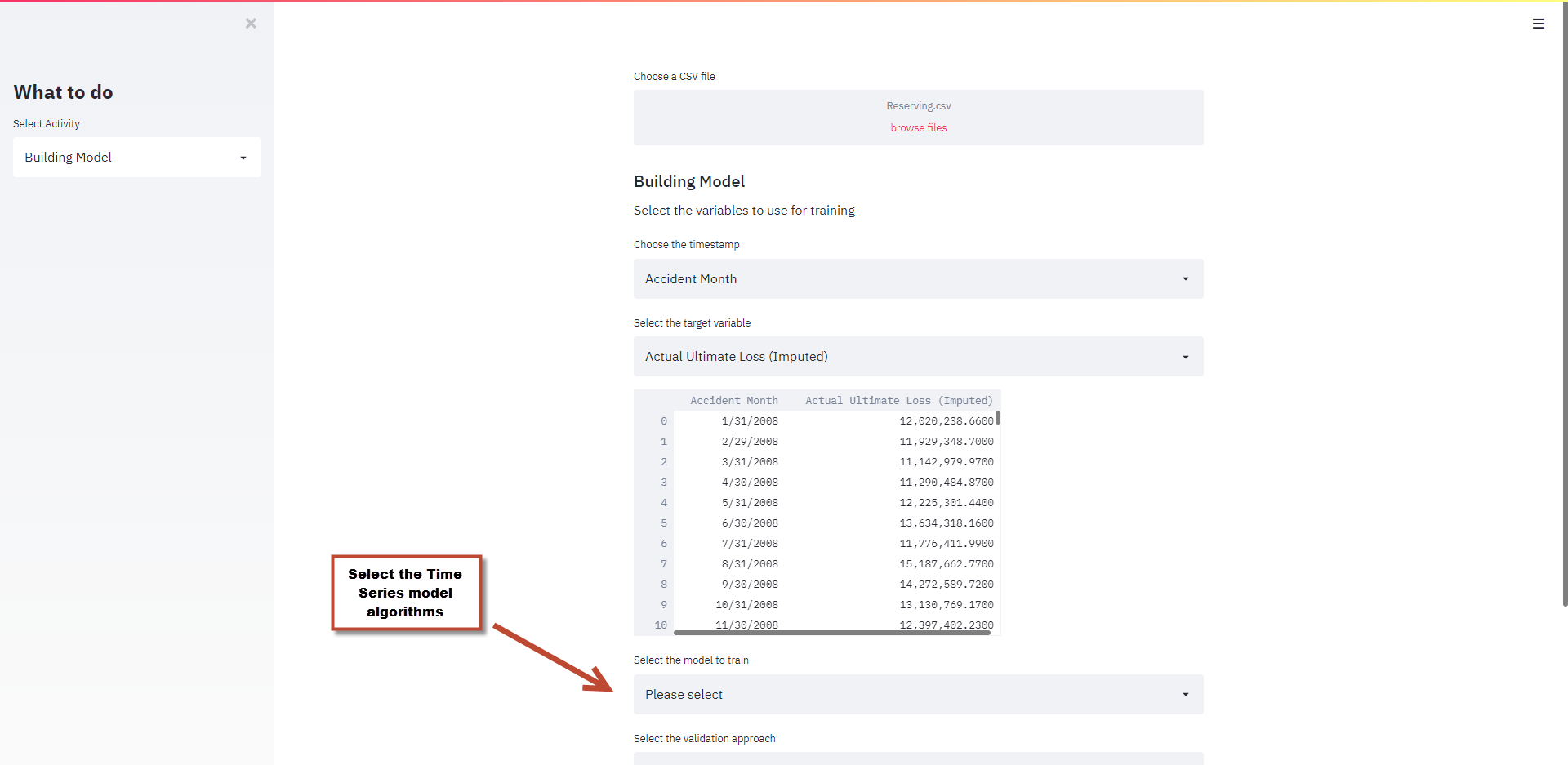
In the “Plotting and Visualization” section, many plot types are available. It should be quite straightforward to plot your dataset here.



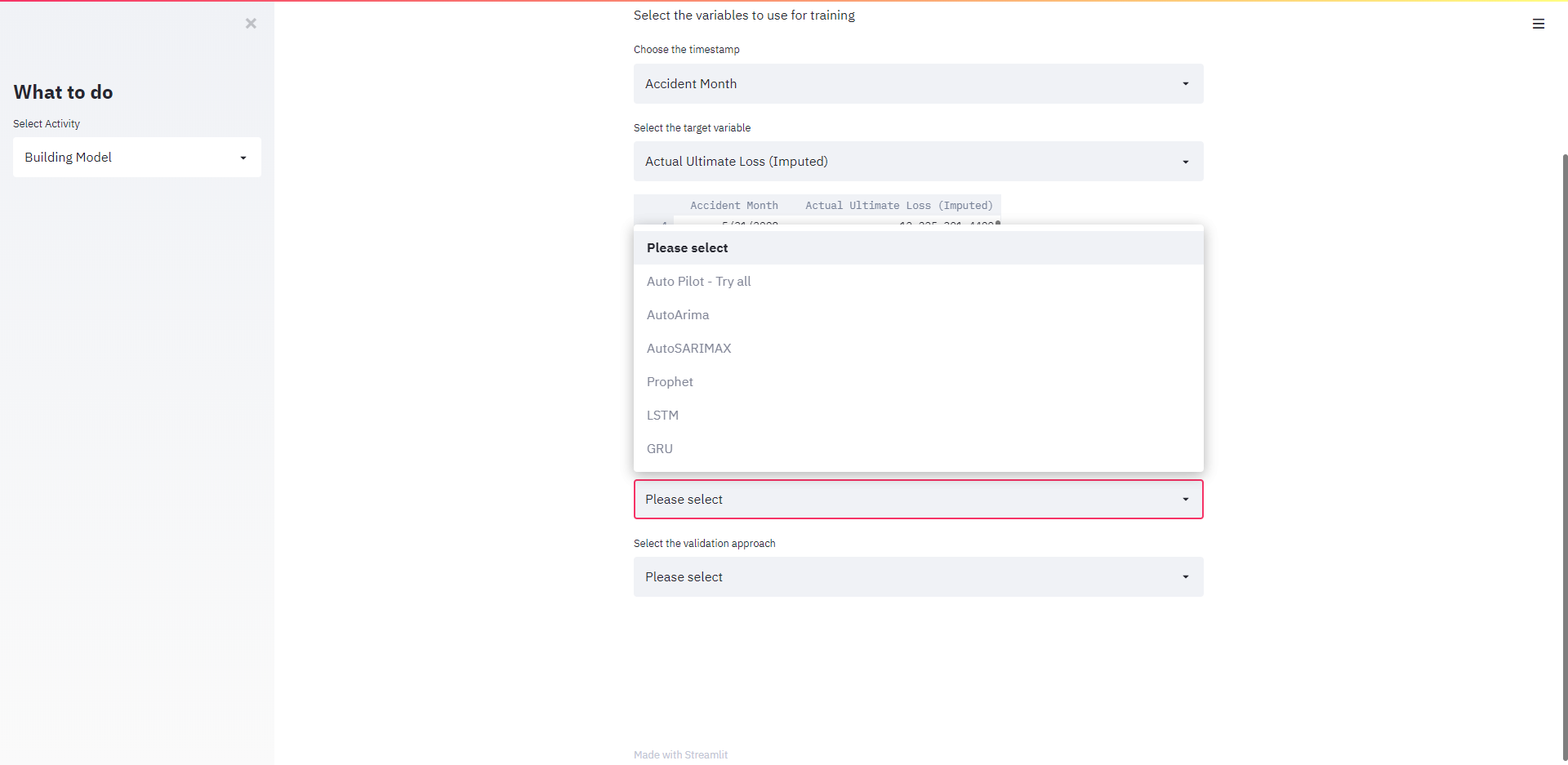


1. **Building Model**

This section is where we build models, try different validation methods, and even auto pilot all implemented algorithms. There are two major components, modeling algorithm and validation method. First, let’s talk about what modeling algorithms are there.

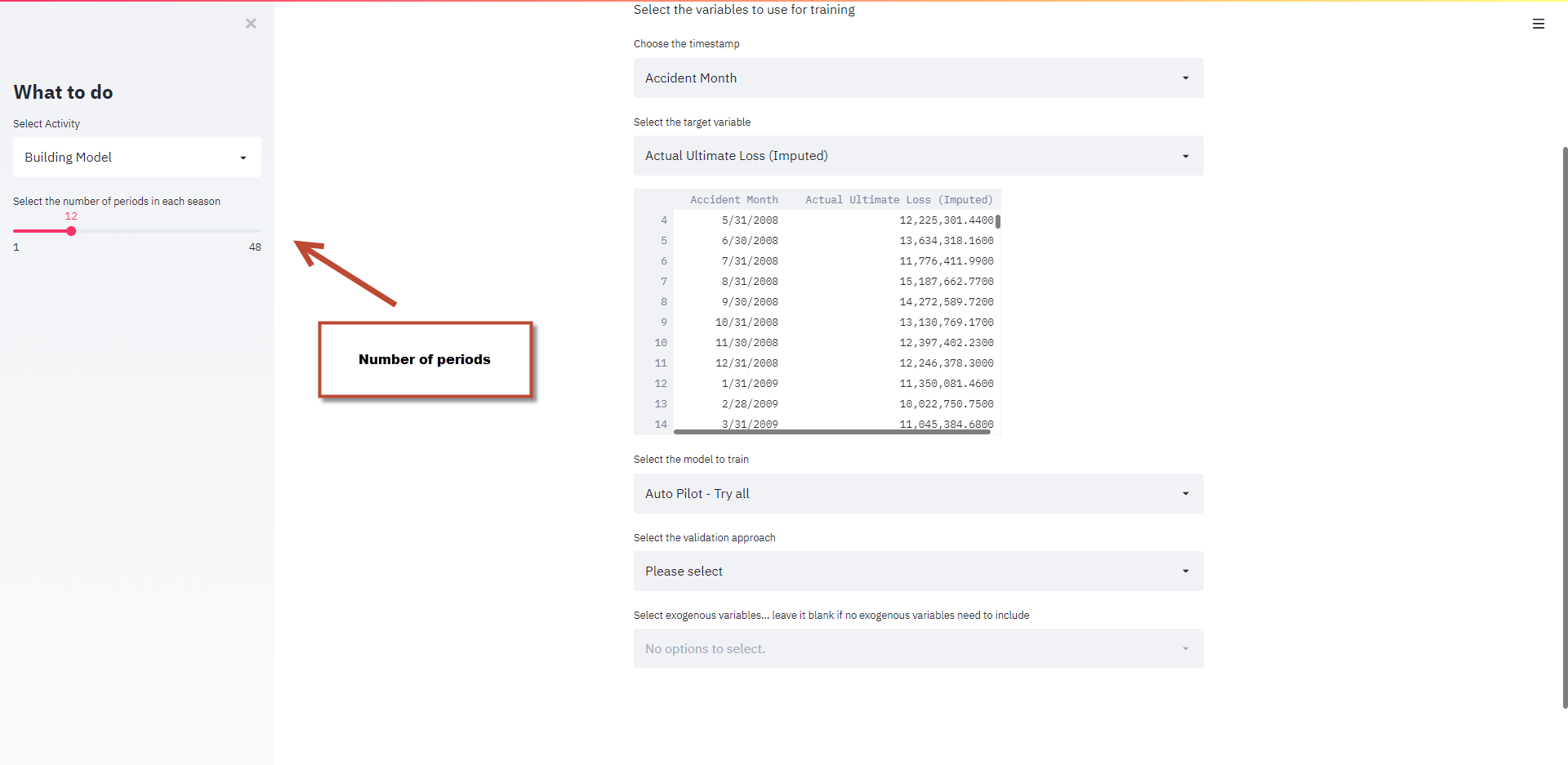


When you click on the field to choose model, you will see 6 options so far as of 8/28/2020. Namely they are **Auto Pilot, AutoArima, AutoSArimaX, Prophet, LSTM, and GRU**.



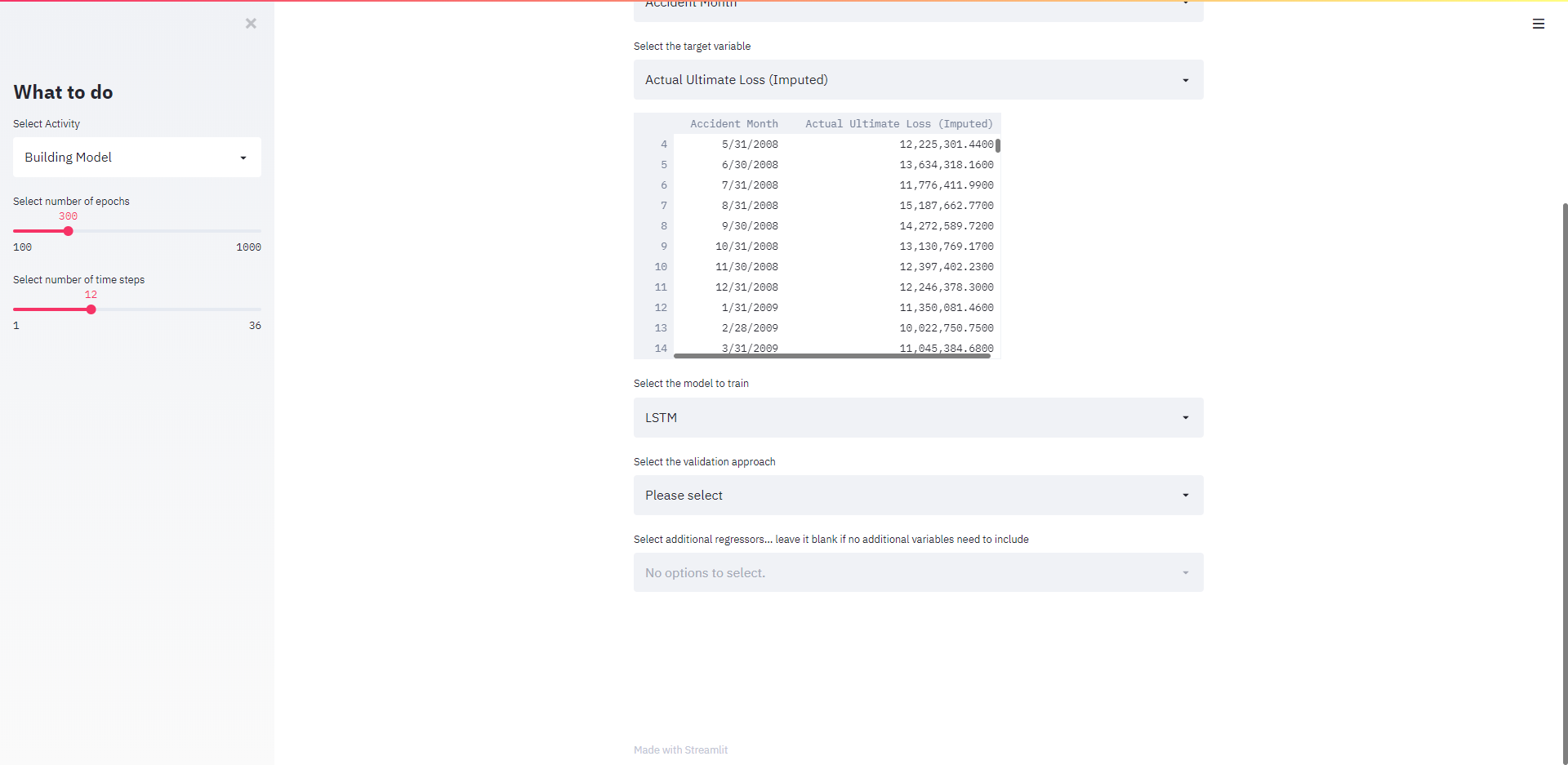
* 1. **Building Model - Auto Pilot – Try all, AutoArima, AutoSARIMAX**

If you select Auto Pilot, AutoArima, or AutoSAMRIMAX, you will need to set number of periods with your knowledge about your dataset. Basically, if your data is monthly and you think each season should be 1 year, then you should try 12; if your data is daily and you think each season should be 1 week, then you should try 7. But it’s merely a number based on your belief, so you might want to try different numbers here manually and it’s the only parameter need to be set manually for ARIMA models.



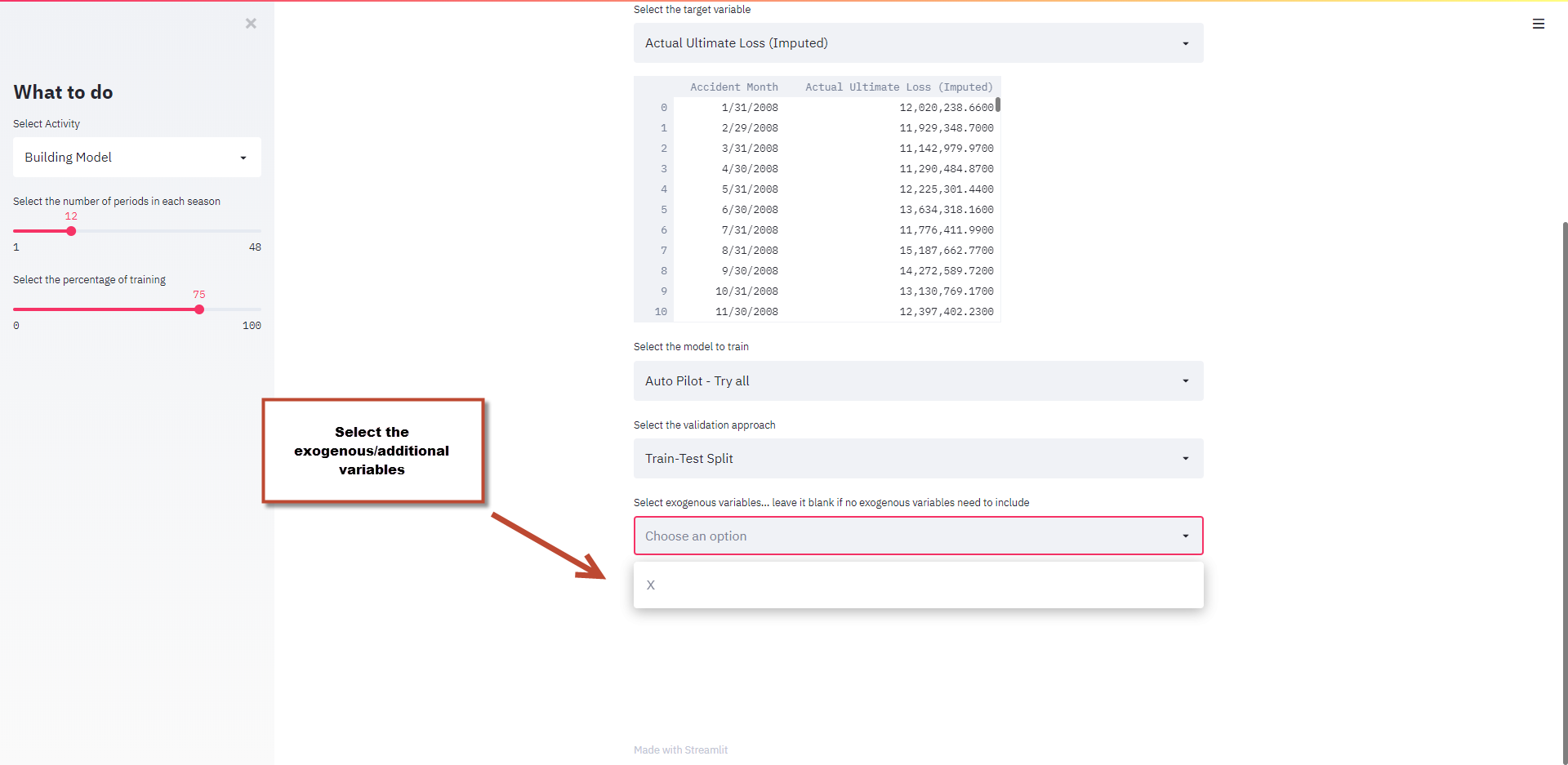
* 1. **Building Model – LSTM, GRU**

When you want to use LSTM or GRU, you will need to choose maximum number of Epochs and number of time steps. Although there is early stopping in place, you are still able to set the maximum number of epochs in case it takes too long to train.



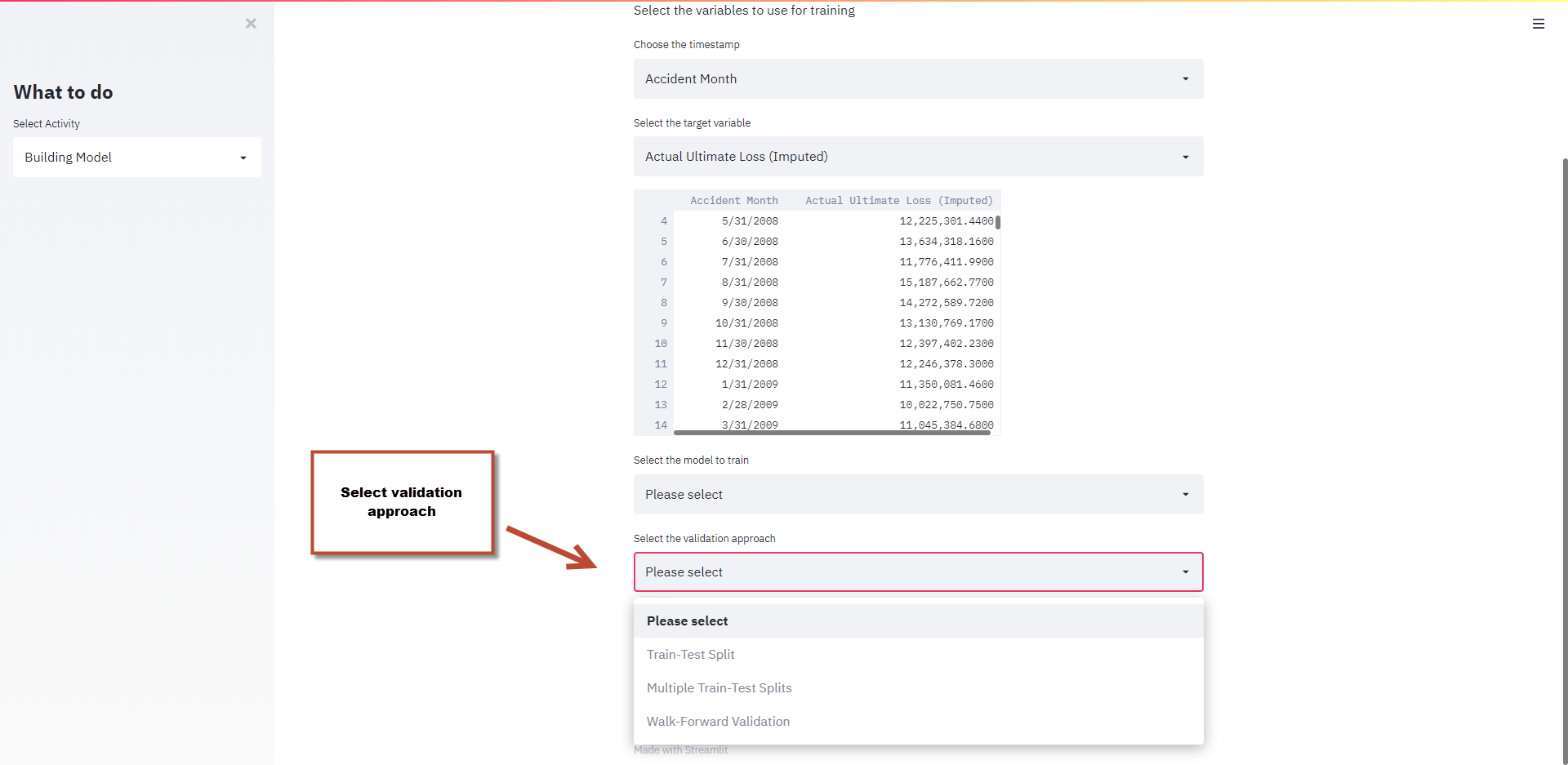
* 1. **Building Model – Exogenous variables**

Other than AutoArima, you should be able to select any additional variables you want to include in your model. If not, just leave it blank. By default, the model takes only numerical values and you need to transform any categorical variables before importing the dataset. This will be something to work on in the future.



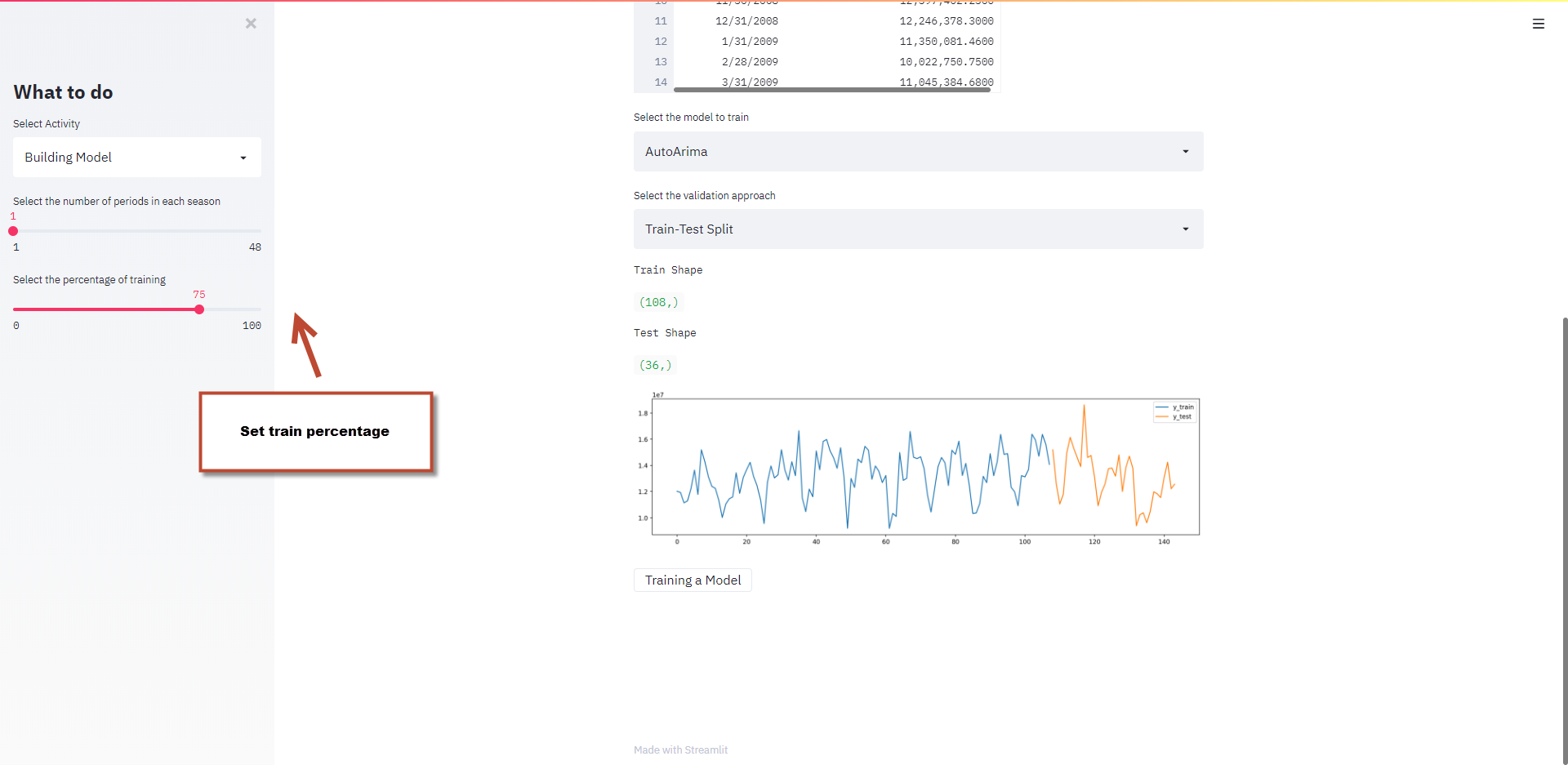
* 1. **Building Model – Validation**

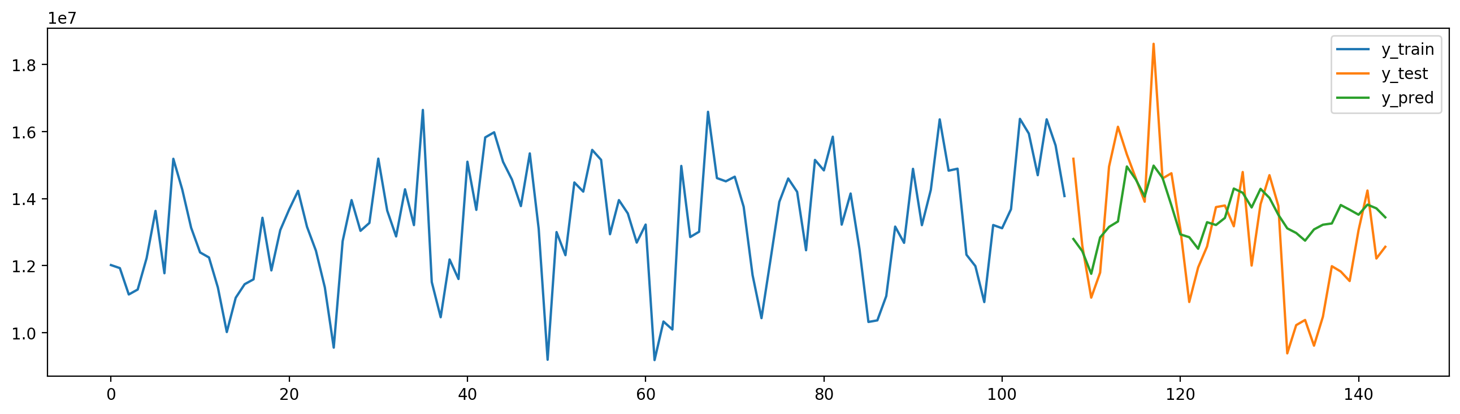
Three validation approaches have been implemented. They are traditional train-test split, multiple train-test splits (k-fold cross validation for time series) and walk forward validation.



* 1. **Building Model – Validation – train-test split**

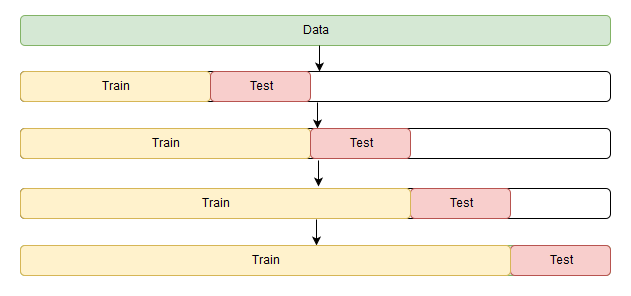
Train-test split will split your data by your provided time stamp column, and you can choose the split percentage on the left side. After training your model, you will see a graph like below. The thing worth being aware of is that the prediction is made based on previous predicted values other than actual values. For example, assume our model used 10 records (t1 to t10) in the past to predict t11. During the train-test split validation, the model acts like not knowing anything as of the split point, so the prediction at t140 in the chart is based on predictions from t130 to t139.

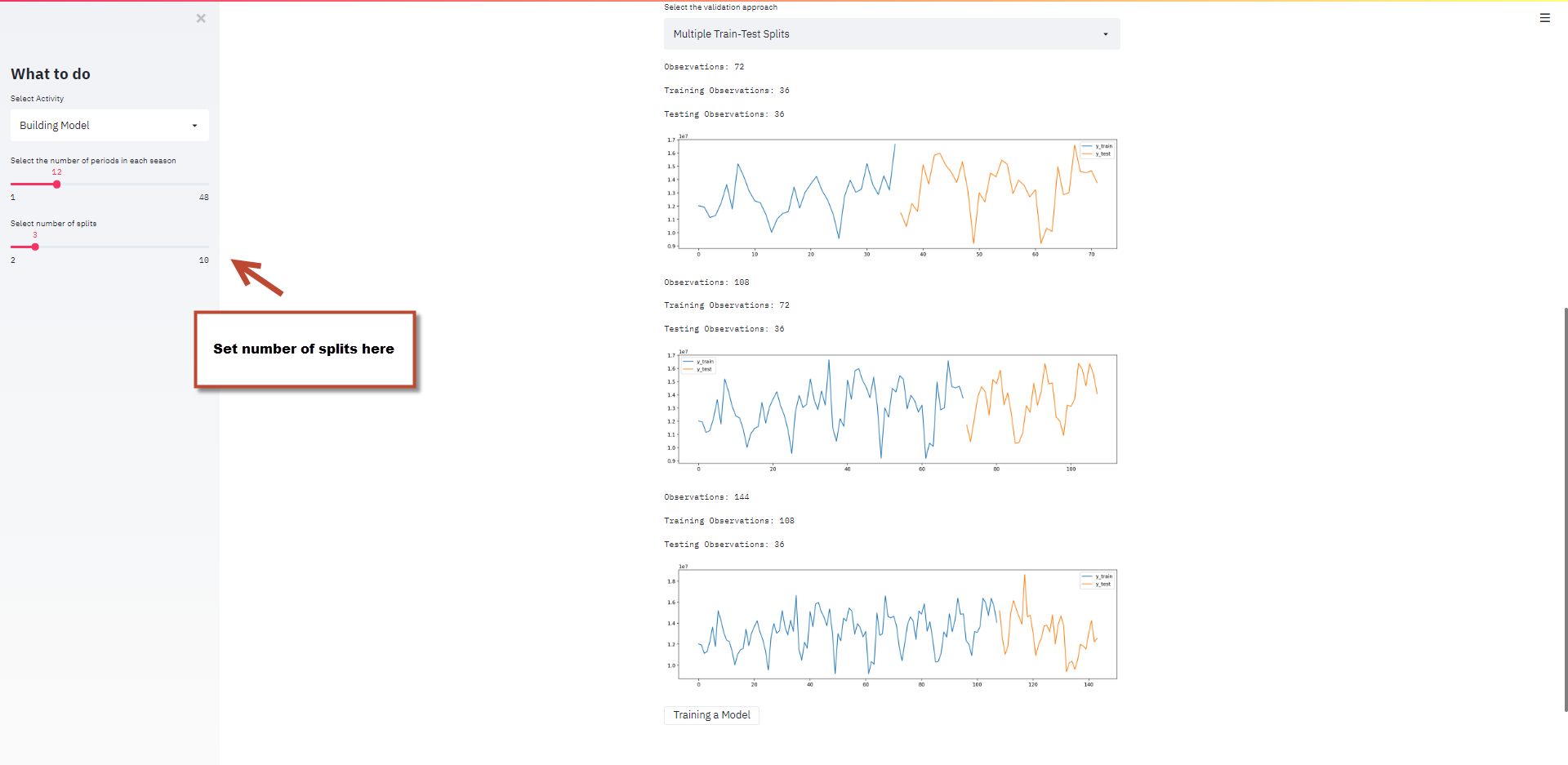




* 1. **Building Model – Validation – multiple train test split**

Multiple train-test split will split your data by selected times on the left side. After N splits, the time series will create N new series in the following fashion, and each series will be treated as a single train-test split. Because the test datasets are of the same size, we compare the evaluation metrics between N new series. This is a more robust approach than single train-test split to test the fitness of the model.

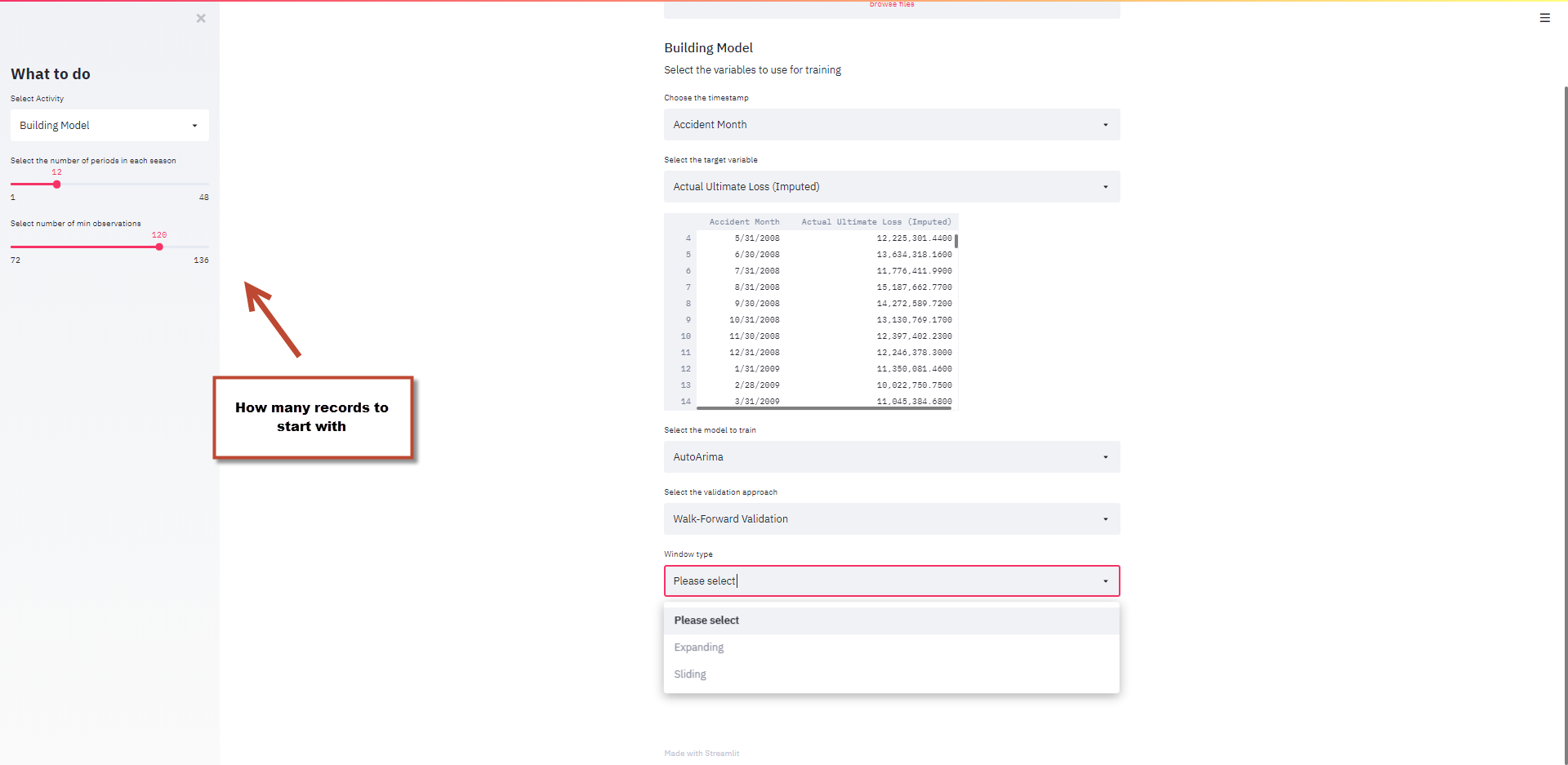


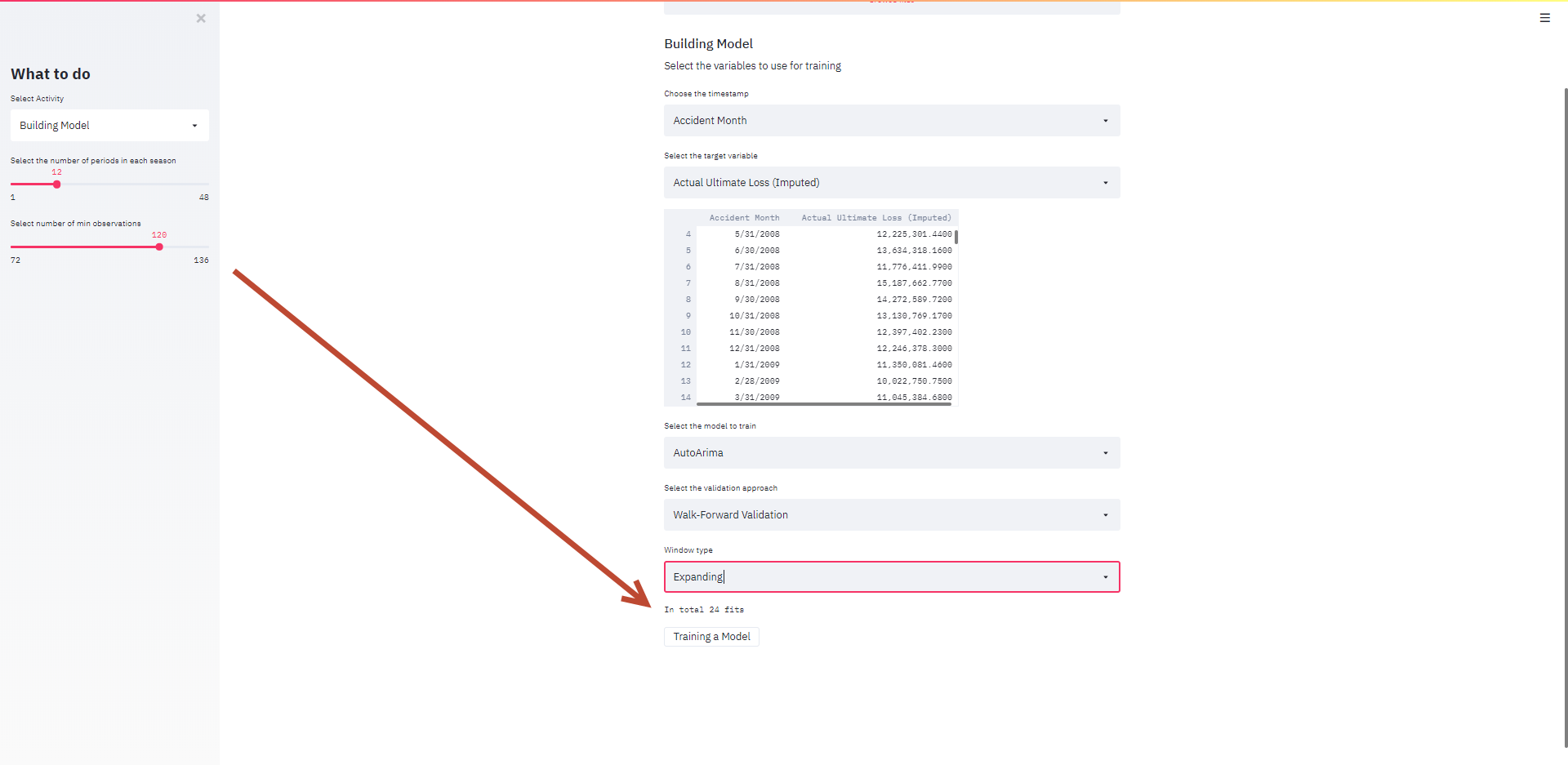


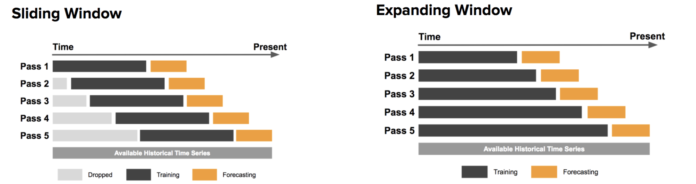
* 1. **Building Model – Validation – Walk-Forward validation**

Walk-Forward validation makes one step prediction at a time. On the left side, we need to choose the minimum number of observations to fit the first model. Since this process will train many models, a line showing how many models will be trained in total will appear after the number of minimum observations is set.

Two types of walk-forward validation styles, sliding and expanding, are shown below. In sliding setting, the oldest record is dropped, and a new record is added at every step; while in expanding setting, the oldest record will not be dropped and everything else is the same as sliding.







* 1. **Building Model – Train and output results**

After the model and validation have been set, next step is to train the model. There is a button at the bottom and the model will starting training once you click on it. After the training is done, you will see a link that will let you download the prediction in csv format with which you can do further analysis.

