# Time Series Forecasting

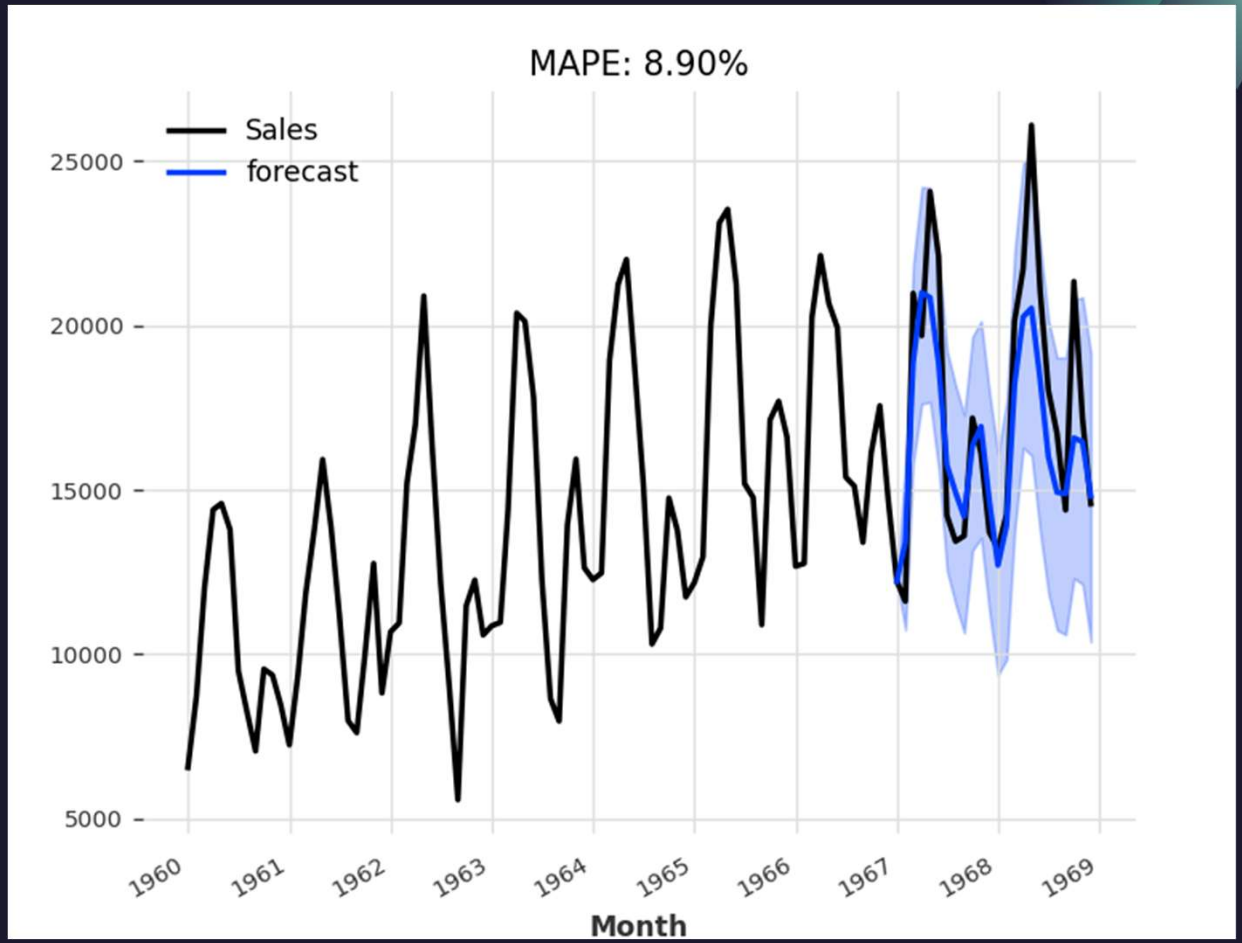
Aaron Mullen

# My Project

- Explore existing methods of time series forecasting
- See how they perform on different types of datasets
- So far, I have only used one dataset, but I have tried almost every major method
  - Car Sales dataset (simple and small)

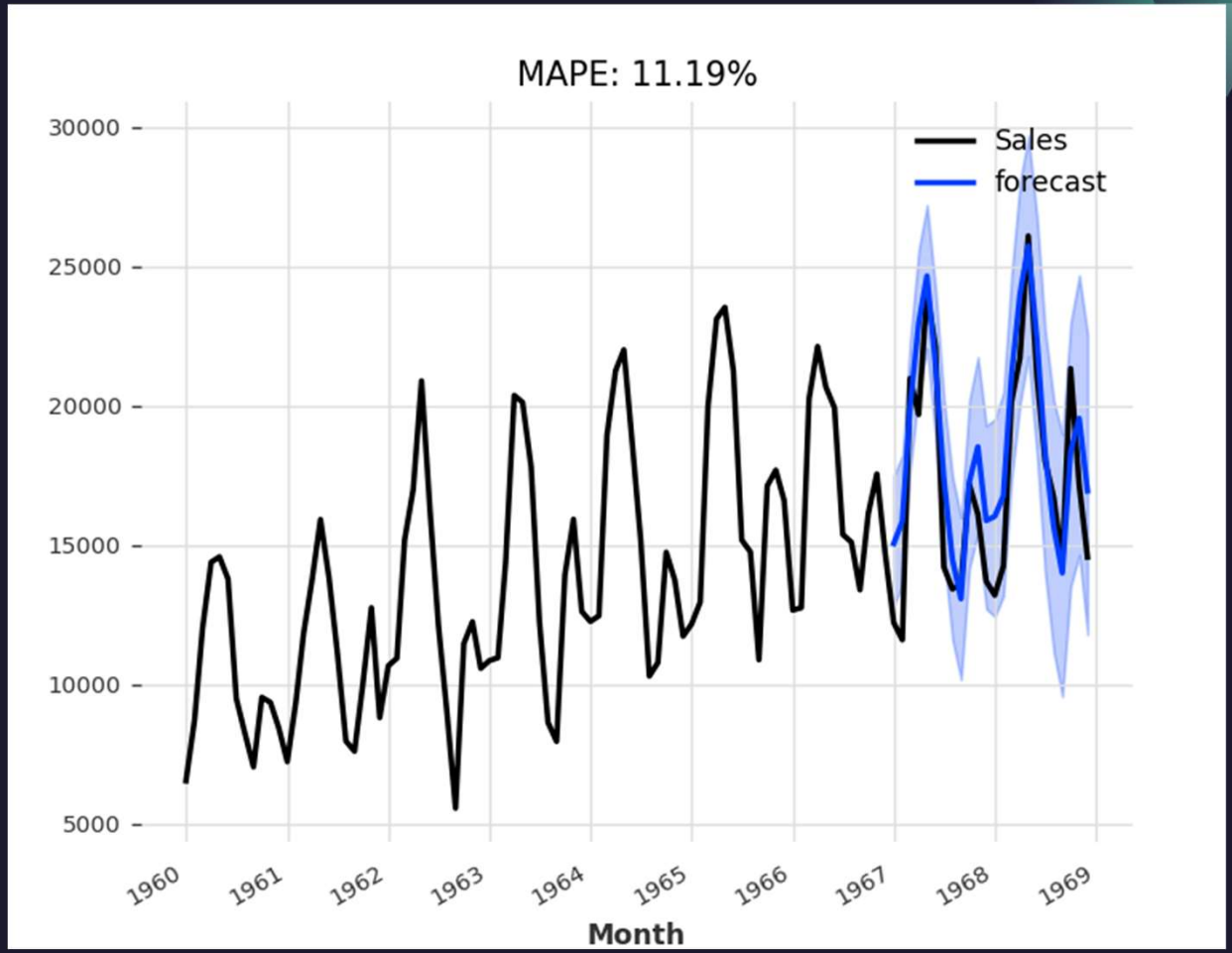
# ARIMA

- Easy to implement, works well
- Auto Regressive Integrated Moving Average
- Uses previous measured values as well as previous error values from past time steps
- Mean Absolute Percentage Error: error measurement of how close the prediction is with the actual values
  - Lower = better
  - Has problems (will discuss later)



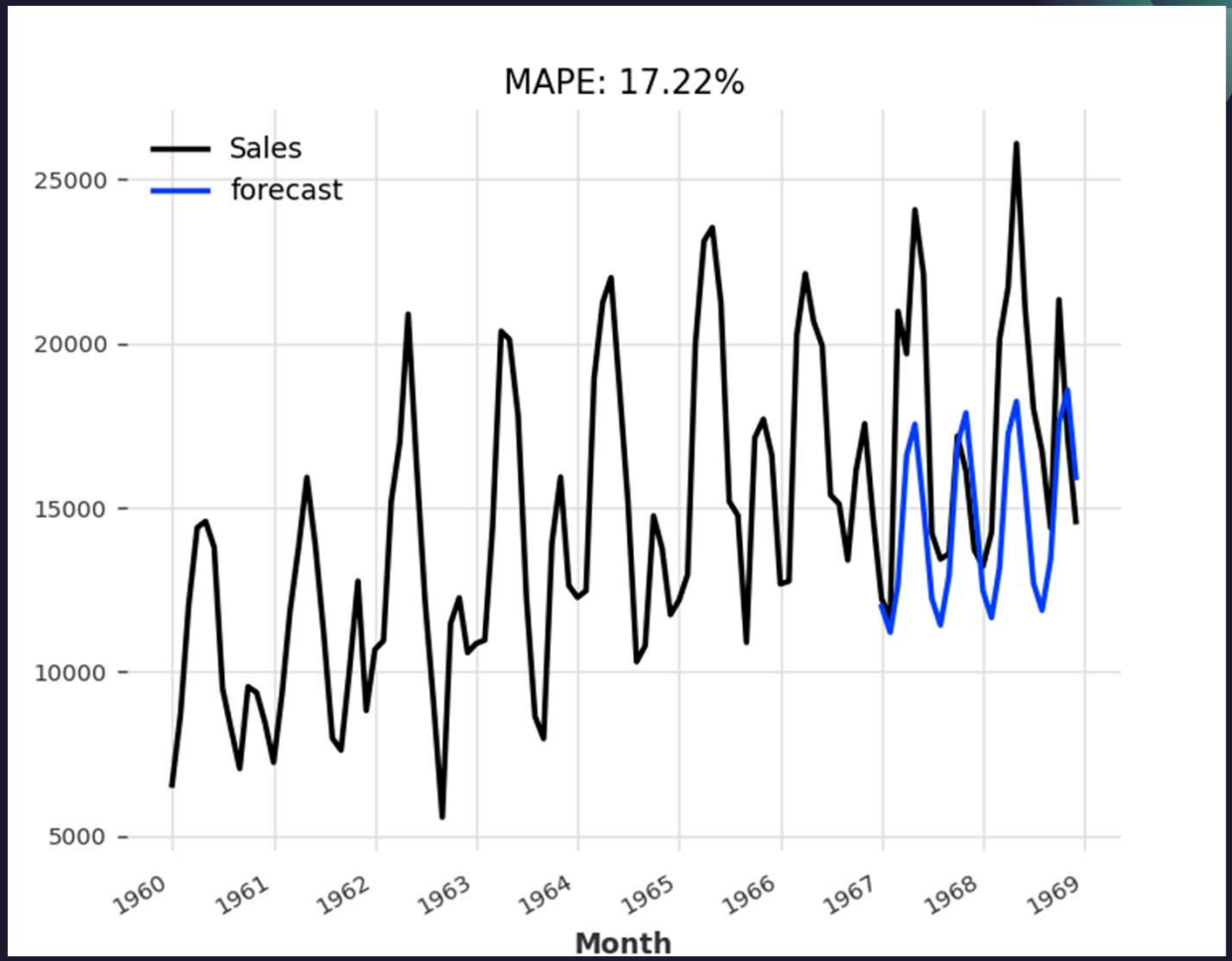
# Exponential Smoothing

- Very similar to ARIMA: main difference is that weights of older values exponentially decrease
- So older observations are less important comparatively
- Versus ARIMA, where all observations are weighted equally



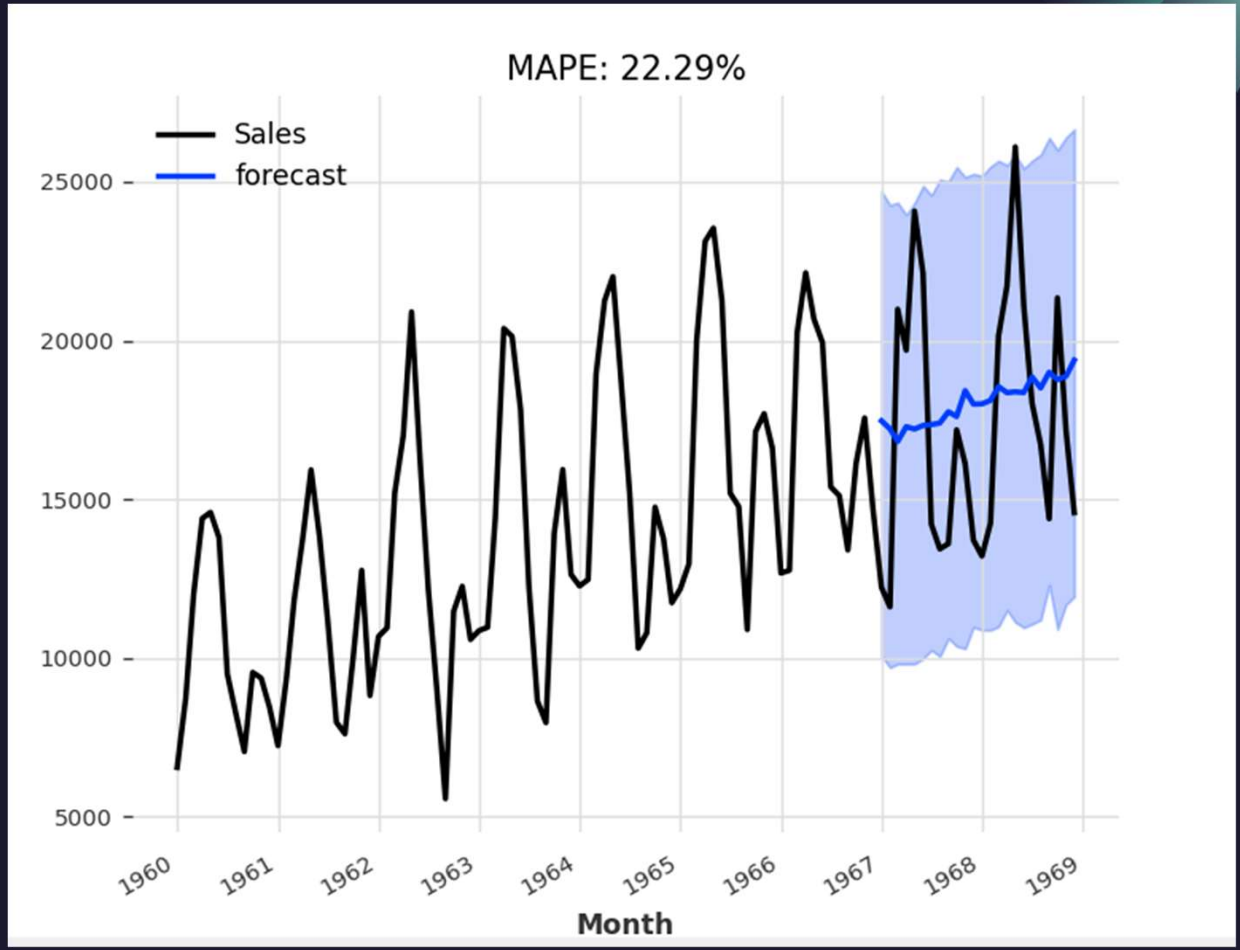
# Theta

- Uses two "theta" lines to make prediction: one to represent its linear trend and one to represent its curvature
- Decomposes seasonal data before predicting



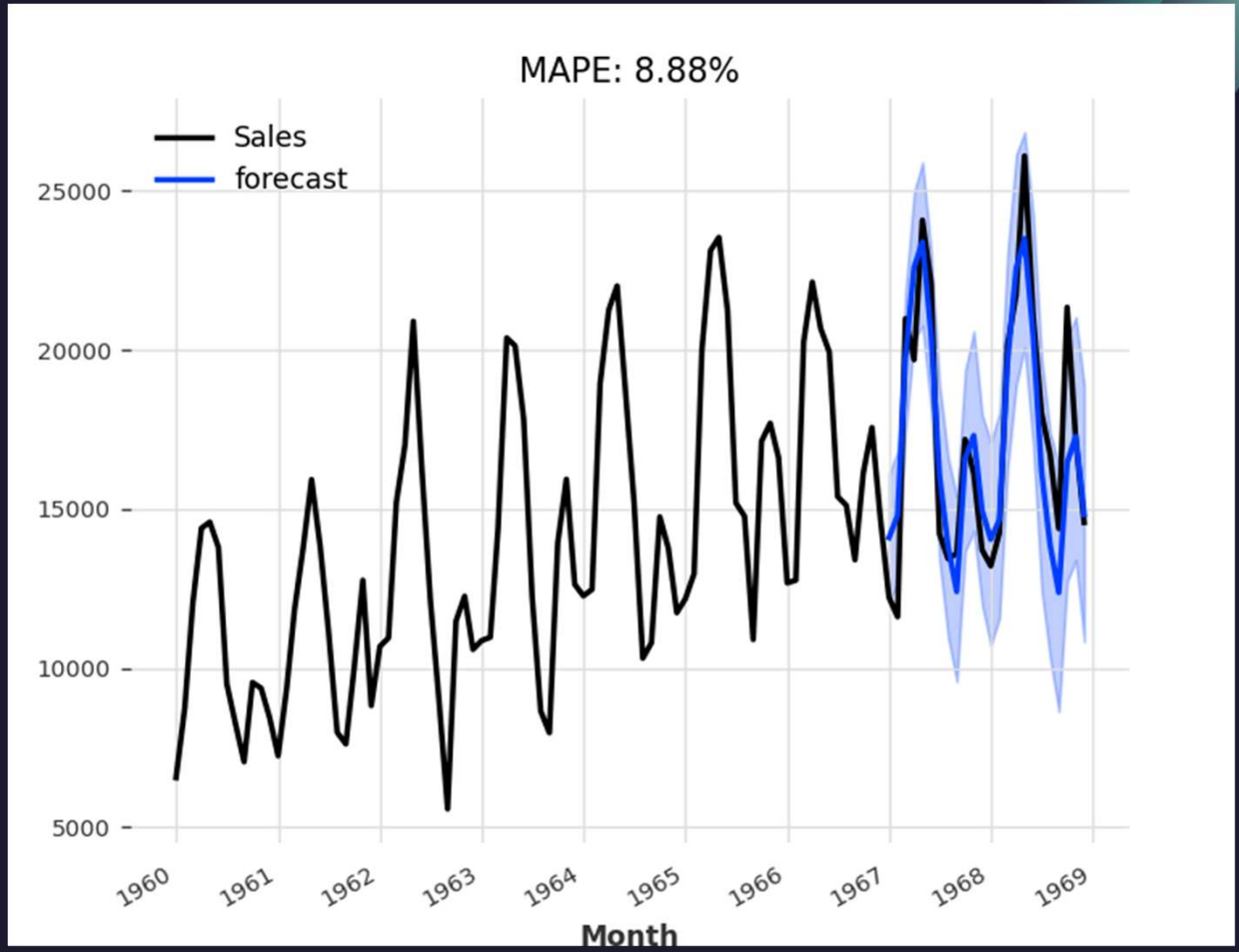
# Kalman Forecaster

- Uses Kalman filter to produce estimates of variables and their uncertainties
- Recursive algorithm, updating previous estimates with new averages
- Better at creating a range of possible future values than predicting specific ones



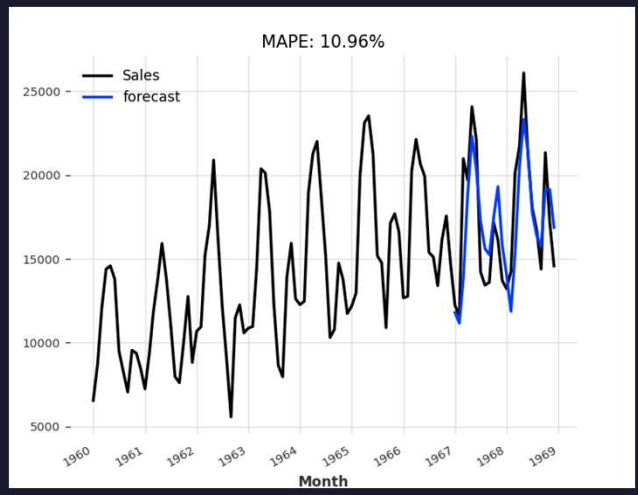
# TBATS

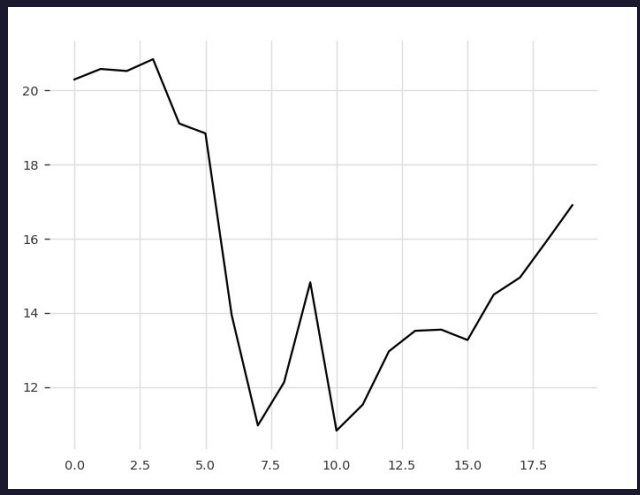
- (Trigonometric, Box-Cox, ARMA, Trend, Seasonal)
- Clearly very good performance, but takes significantly longer to train than other methods so far
- Specifically built to account for multiple seasonality patterns



# Regression

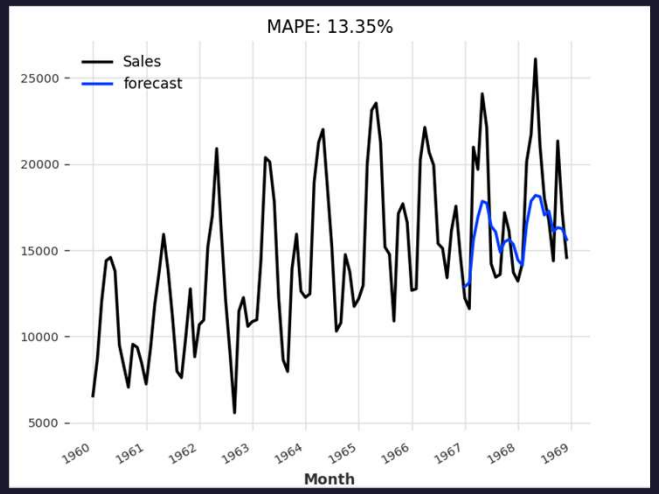
- Builds feature matrix of current and past observations
- Uses linear regression model to get estimate of linear relationship of response to feature matrix
- Requires definition of how many lags (previous observations) it should look at for each time step
- Bottom graph displays performance on y-axis and number of lags on x-axis
- Works well when parameters are optimized

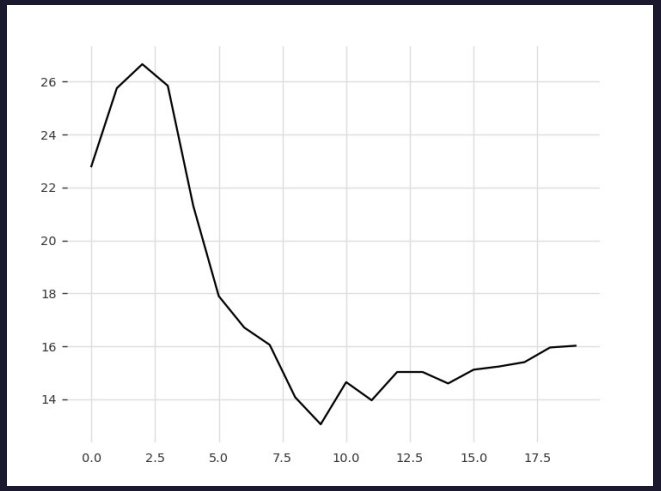


# Random Forest

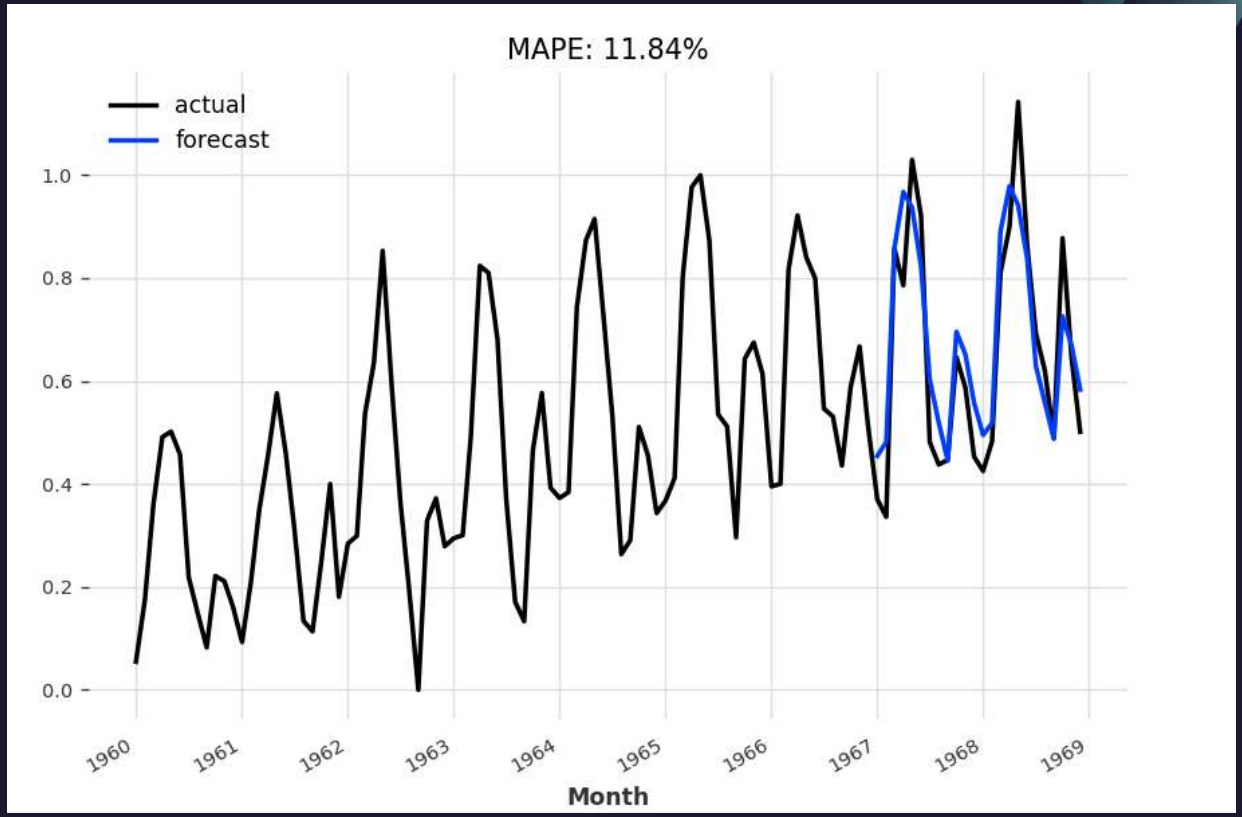
- Uses ensemble of different random decision trees built on slightly different training sets, averaged together
- Requires data to be reformed into supervised learning problem (replacing time column with lagged values)
- Validated using walk-forward validation (predict first step in validation, add it to model, refit, and predict next validation step, and so on)
- Suspicion: random sampling does not work well for very time and order-oriented data like time series



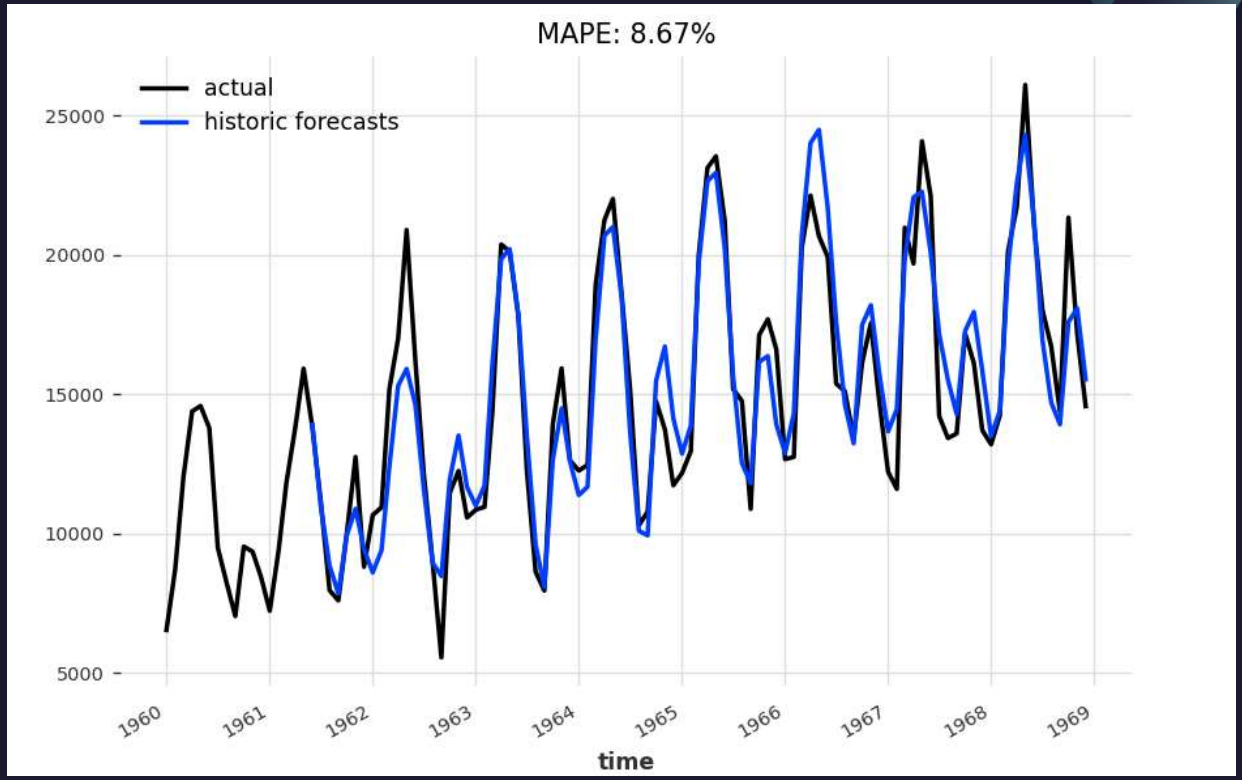
# Recurrent Neural Network

- Uses Long-Term Short Memory (LTSM): instead of neurons, memory blocks are connected through layers
- Each block has gates that manage the block's inputs and outputs
- Requires transformation of data, like the regression problems
- Performs well, but takes a long time to train and requires a lot of parameter tuning



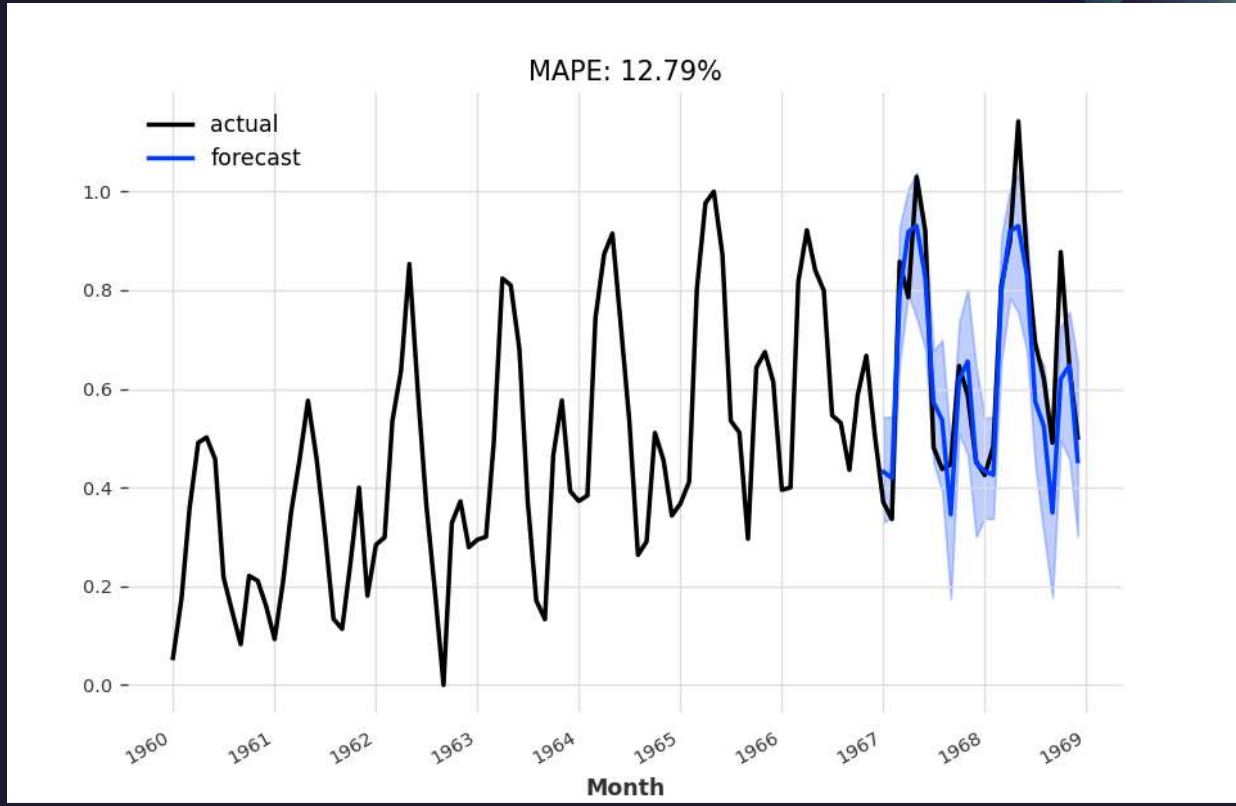
# NBEATS

- Neural Basis Expansion Analysis for interpretable Time Series forecasting
- Deep stack of fully-connected layers that look at windows of past values for each pass
- Consists in several blocks connected in a residual way: first block tries to model past window and future the best it can, then the second block tries to model the residual error of the past reconstruction done by the previous block, and so on
- Uses expanding window to repeatedly build new training sets, create forecast, and then move on
- Faster training, more configuration, and better performance than regular RNN



# Temporal Fusion Transformer

- Deep neural network architecture from torch
- Similar to RNN: complex architecture, many parameters, long training time, but more general than other, more effective time series specific methods



# Lessons I have learned

- Mean Absolute Percentage Error is not the best accuracy metric
  - Based on my observations, it seems to favor predictions that are too low more than predictions that are too high
  - Research has supported this idea
- What other metrics to try?
  - Mean Absolute Scaled Error- penalizes positive and negative forecast errors equally
  - Root Mean Squared Error- commonly used, but very sensitive to outliers
  - Correlation between actual and predicted values
- Another lesson: simpler models designed specifically for time series forecasting will likely always work better than more complex, but general, models

# What's next?

- A few more methods that I have not tried (Prophet, Croston, TCN)
- Try these methods on different types of datasets (univariate vs. multivariate, seasonal vs. non-seasonal, bigger and more complex data sets)
- Discover more about accuracy metrics and their strengths and weaknesses