# TIME SERIES FORECASTING

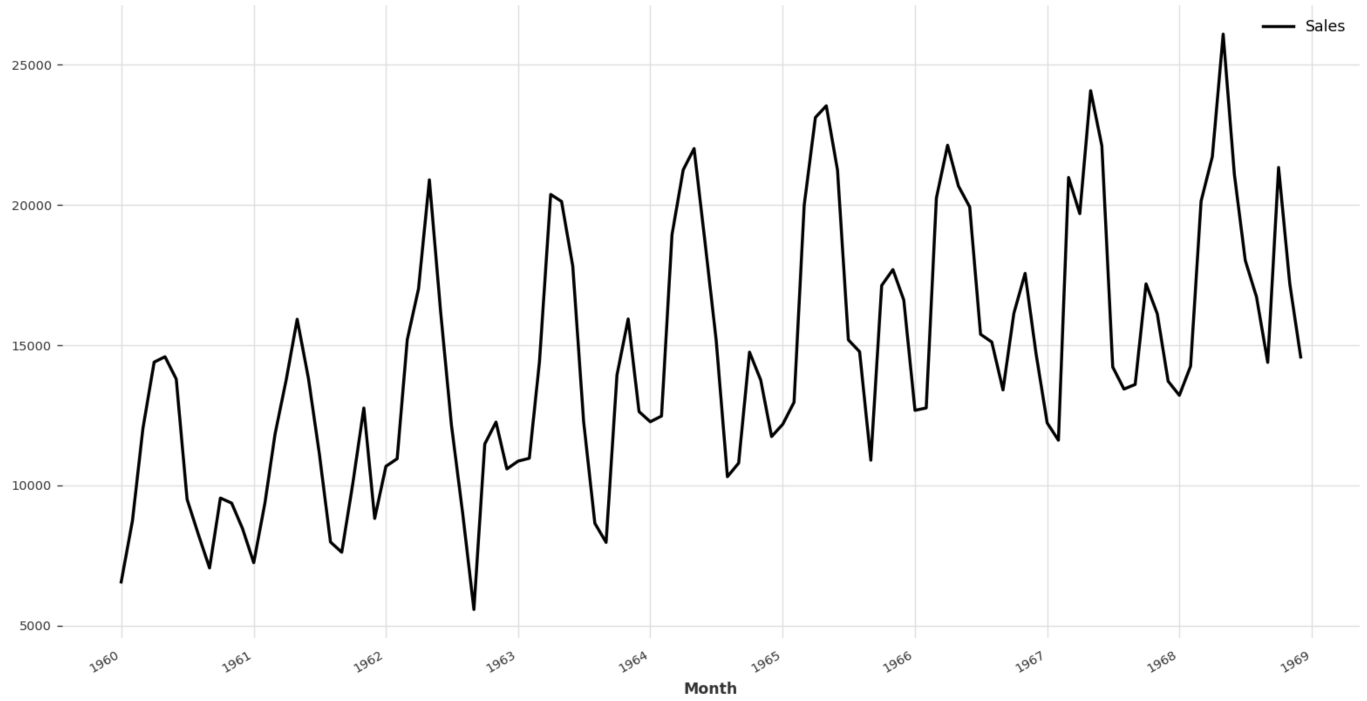
MY CS395 PROJECT- AARON MULLEN

# WHAT IS TIME SERIES FORECASTING?

- Looking at time-oriented data and making predictions about future observations
- There are different types of time series:
  - Some have covariates, some are multivariate
- There are also different types of models used to analyze time series:
  - Some were built specifically for time series forecasting
  - Others are applications of more general machine learning methods
- My goal in this project was to explore these models and their uses on different types of time series

# THE MODELS

| Model | Multivariate? | Covariates? |
|---|---|---|
| ARIMA | | Yes |
| Exponential Smoothing | | |
| Four Theta | | |
| Fast Fourier Transform | | |
| Kalman Forecaster | Yes | Yes |
| TBATS | | |
| Regression | Yes | Yes |
| Random Forest | Yes | Yes |
| Recurrent Neural Network | Yes | Yes |
| NBEATS | Yes | |
| NHiTS | Yes | |
| Temporal Fusion Transformer | Yes | Yes |
| VARIMA | Yes | Yes |



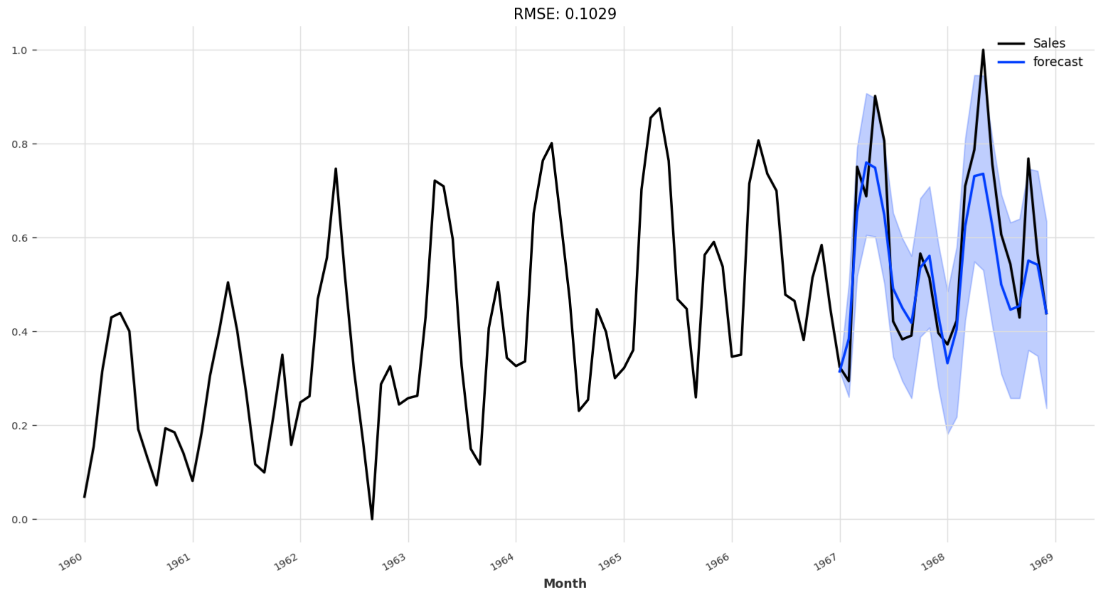

# DATASET 1- CAR SALES

The first dataset tested on kept track of monthly car sales over a ten-year period

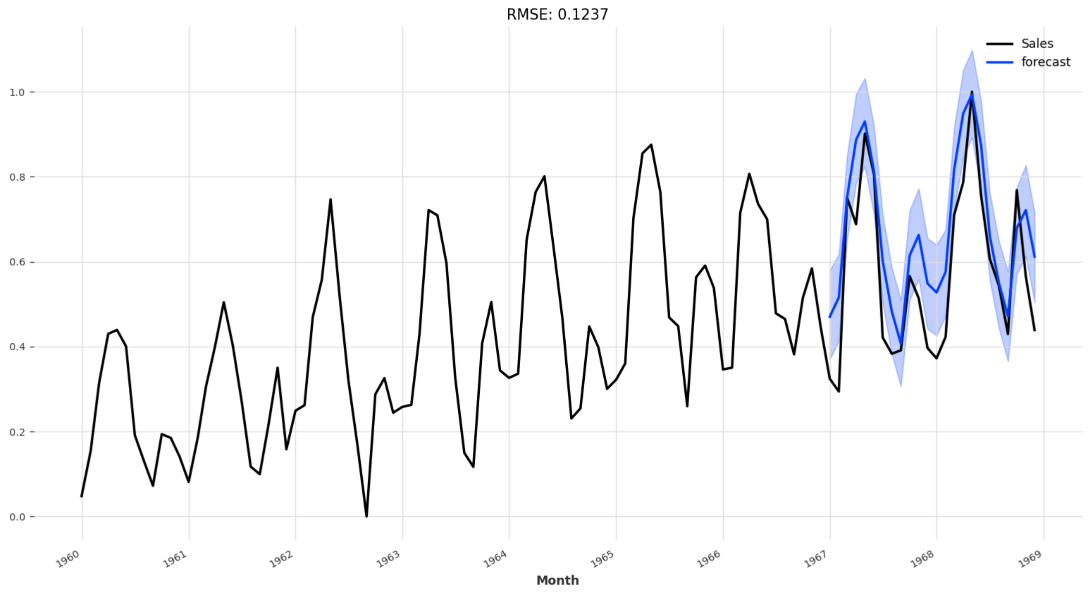
Only 120 rows, no extra variables or conditions

# ARIMA

- One of the most popular and oldest models for time series forecasting
- Uses past observations of the data (lags) and past errors of the model
- Error metric used for all of these is Root Mean Square Error
  - Improvement over Mean Absolute Percentage Error used in the midterm
  - Dependent on scale of the data, so all data was normalized first



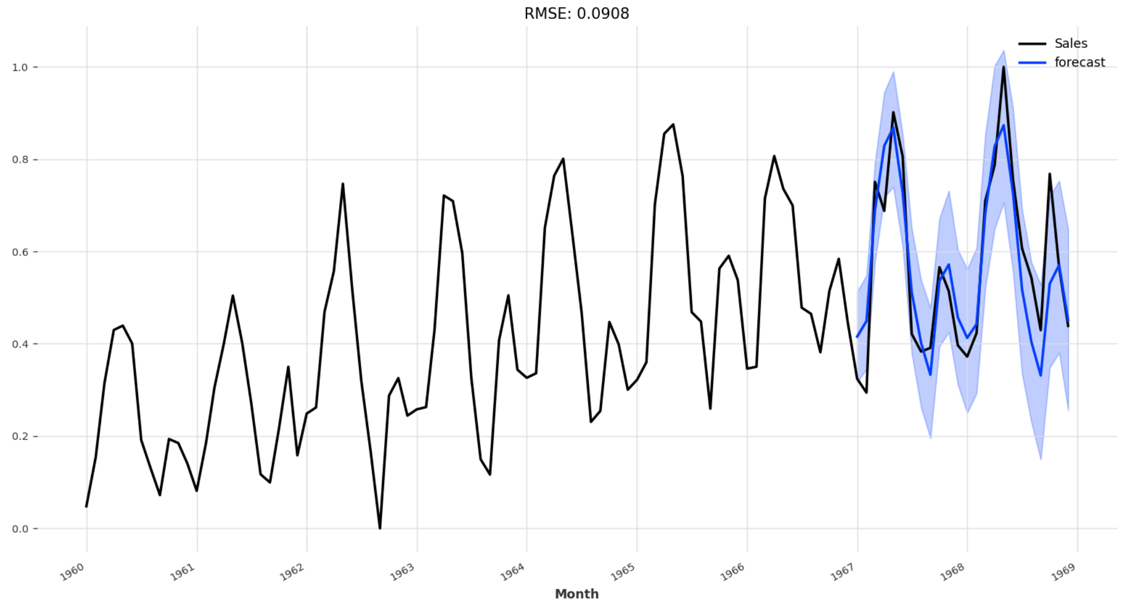
# EXPONENTIAL SMOOTHING



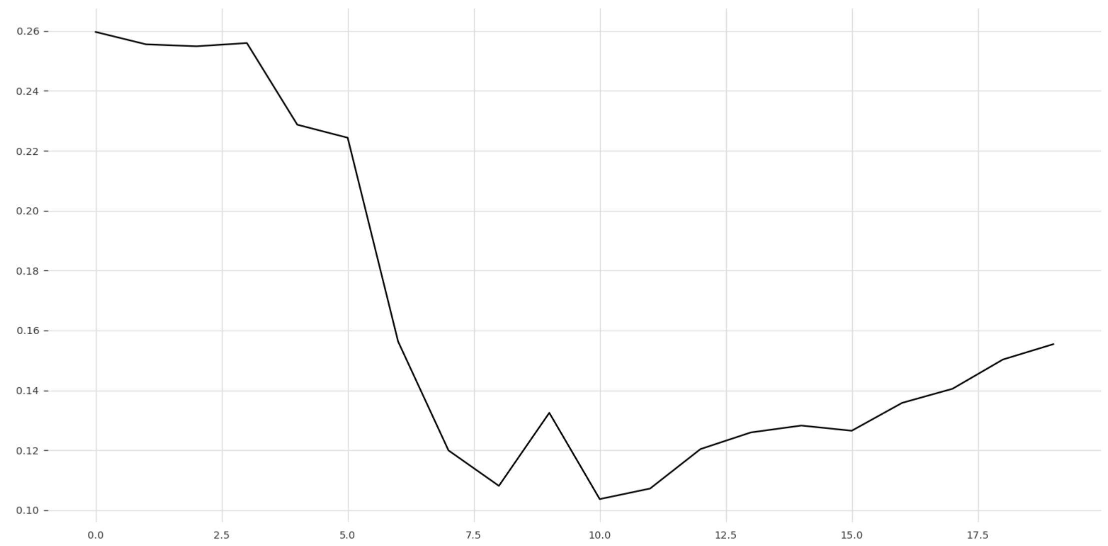

- Similar to ARIMA, but it weighs lags exponentially
  - Older observations are less important to the prediction than more recent ones

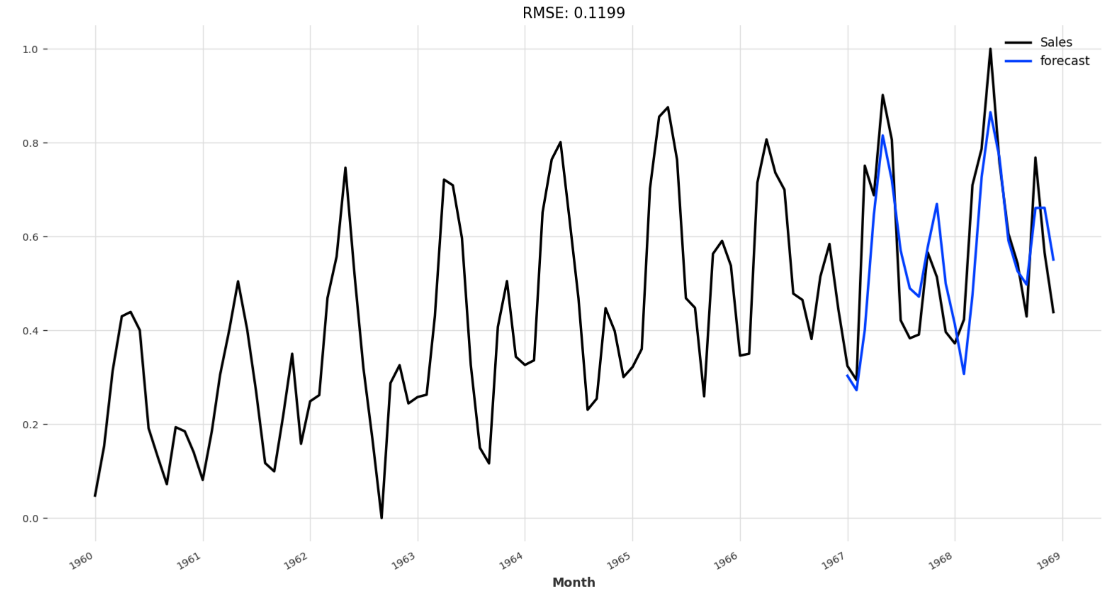
# TBATS

- Trigonometric, Box-Cox, ARIMA, Trend, Seasonal
- Uses all of these components to build many different models, each with different parameters and strategies
- After each model creates its predictions, they are analyzed to determine which performs the best
- Performs extremely well, but takes longer to train



# REGRESSION

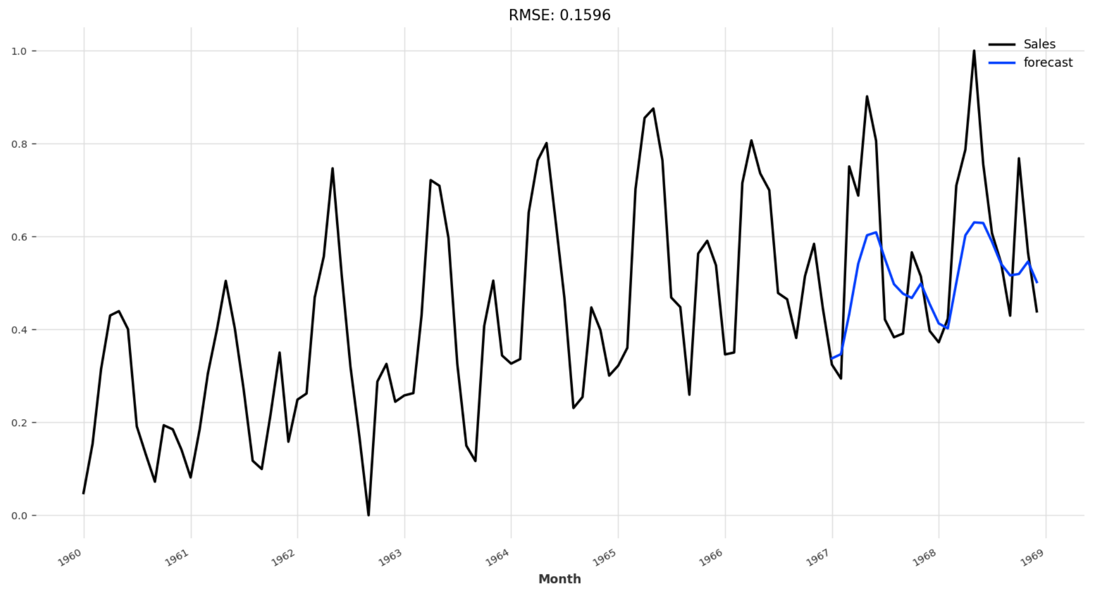


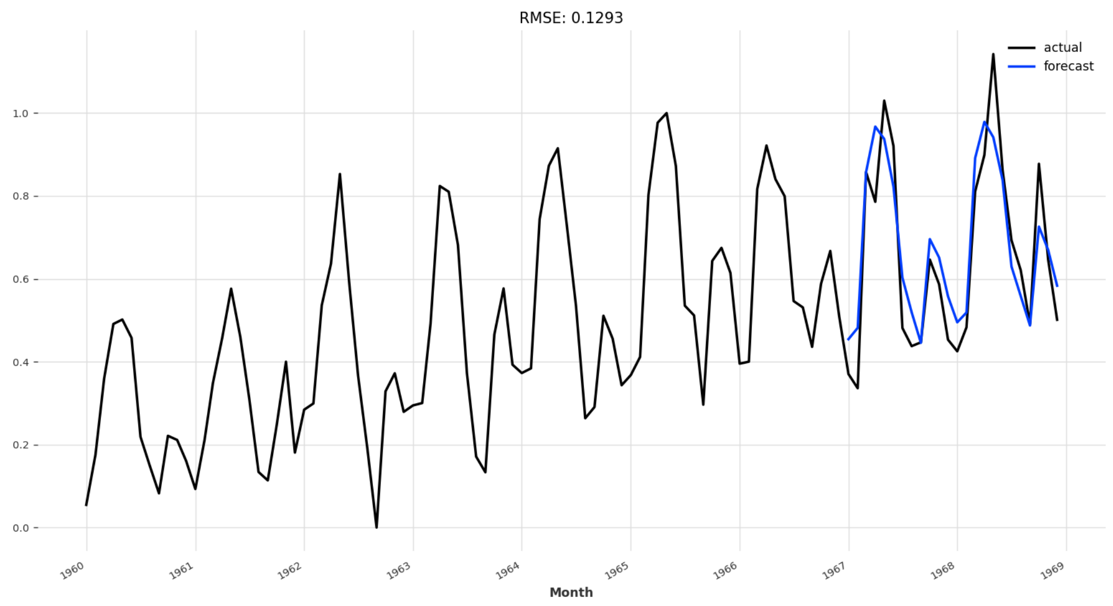

- Builds a feature matrix using current and past observations
- Estimates a linear relationship between the matrix and the prediction variable
- Can use both time-step features for more time-dependent data and lagged features for more serial-dependent data
- Requires definition of number of lags
  - Top graph shows the RMSE as it changes depending on the number of previous lags the model uses

# RANDOM FOREST

- Randomly samples input into many separate, different datasets
- Builds a decision tree for each dataset, and in the end, averages them together
- Does not seem to work well for time series forecasting
  - I believe the random sampling works against the order-dependent time series



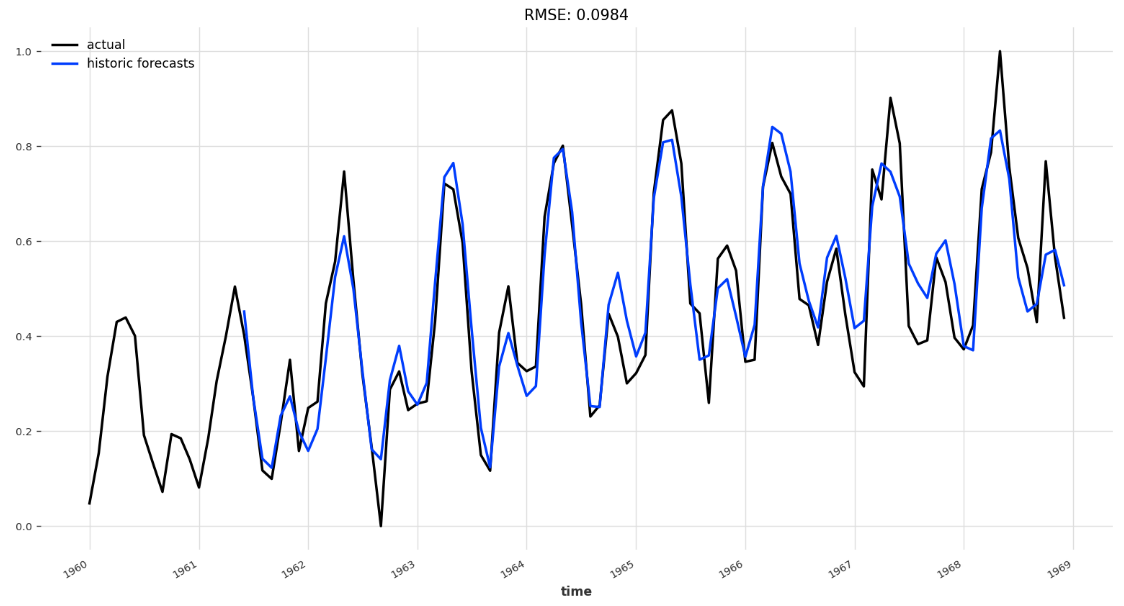
# RECURRENT NEURAL NETWORK



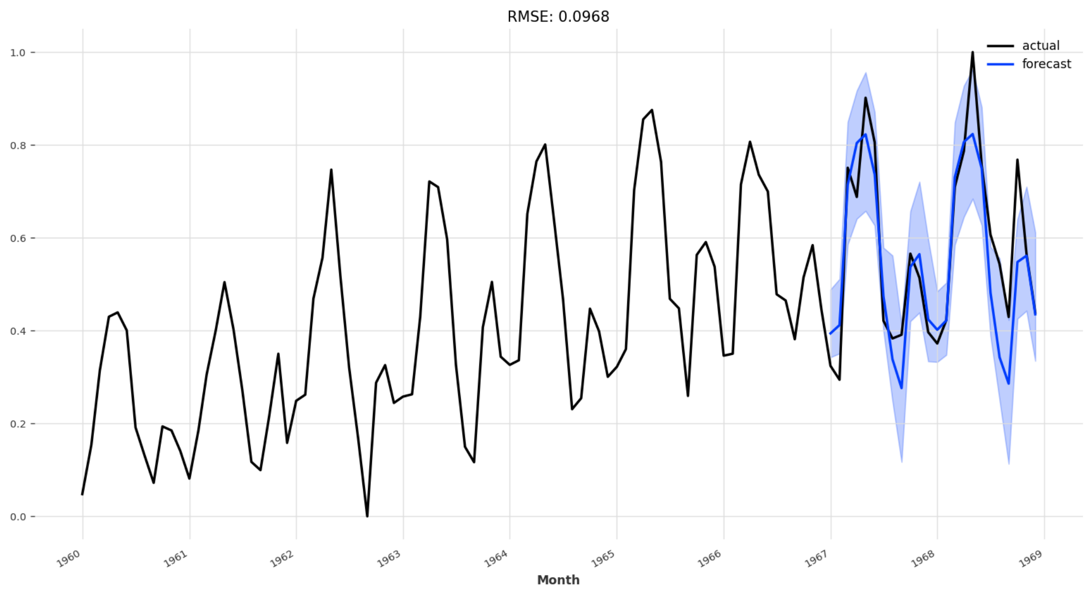

- Uses internal memory to remember previous inputs, making RNNs a popular choice for time series
  - Hidden layer produces output and loops it back in as input for the next run
- Specifically, this is a Long Short-Term Memory (LSTM) network
  - Uses gated cells to keep track of previous inputs for longer, allowing them to store inputs for longer and manage that information better
  - Input gate, output gate, and forget gate can categorize information

# NBEATS

- Requires a different definition of the training and validation sets
- Begins with a small training set, tries to predict the next block of time, observes the actual values, and remembers its error
- Incorporates that block into the training set and runs again, predicting the next block
- This continues until it has mapped the entire series, at which point it is equipped well to make future predictions
- Depending on series length and forecast horizon, this method could take a long time to train, but the extra predictions and expanding training window allow it to be very accurate



# TEMPORAL FUSION TRANSFORMER

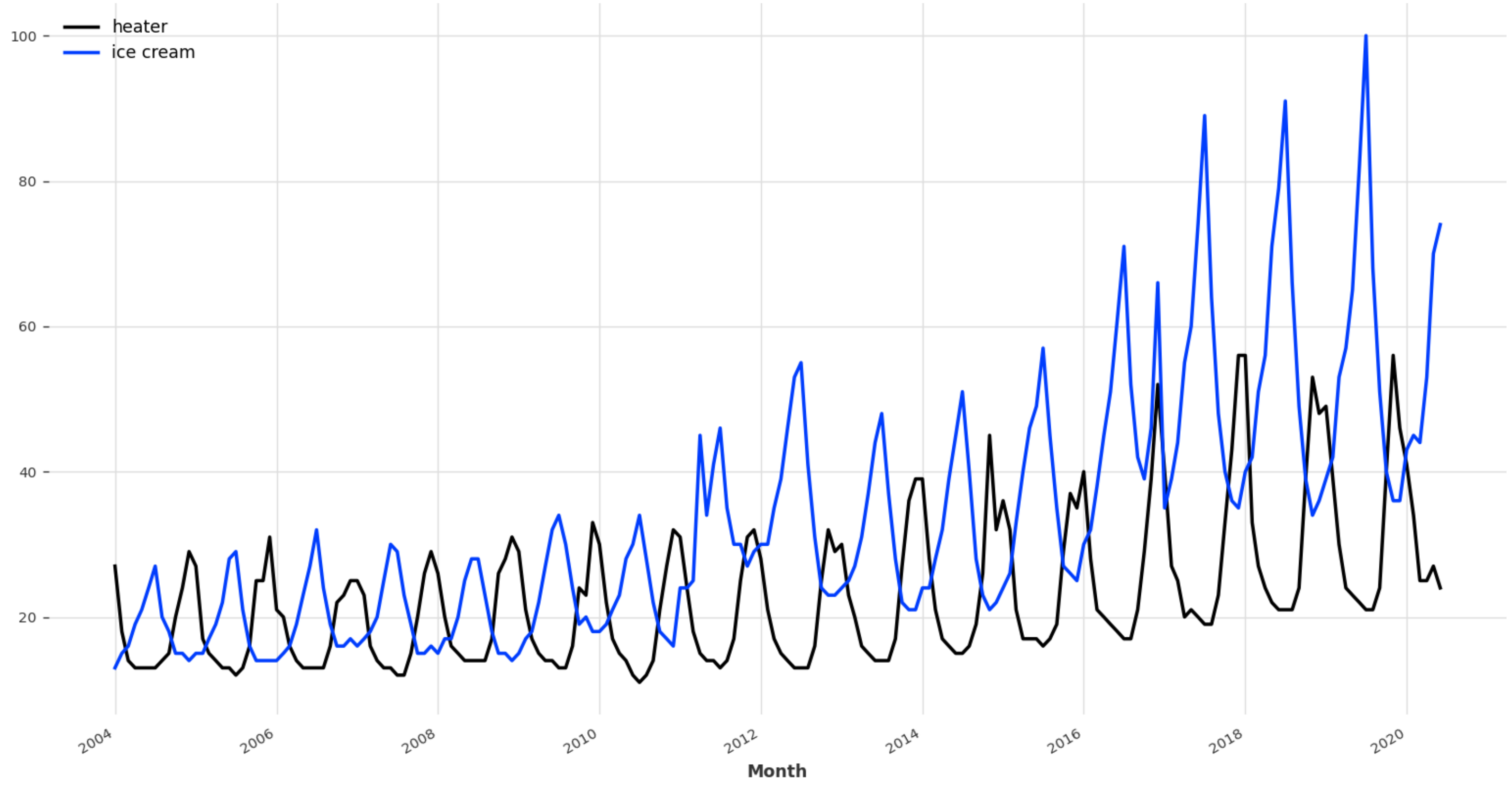



- Deep learning neural network architecture from Torch
- Integrated mechanisms of several different network architectures
  - Integrates some from LSTM, like gated memory blocks
  - Uses attention block to identify long range patterns and prioritize the most relevant
- By far the slowest training time of any model, but performs very well

# OVERALL RESULTS- DATASET 1

| Model | Root Mean Square Error | Time to Train (seconds) |
|---|---|---|
| ARIMA | 0.1029 | 5 |
| Exponential Smoothing | 0.1237 | 5 |
| Four Theta | 0.2880 | 4 |
| Fast Fourier Transform | 0.2957 | 4 |
| Kalman Forecaster | 0.1932 | 4 |
| TBATS | 0.0908 | 75 |
| Regression | 0.1199 | 4 |
| Random Forest | 0.1596 | 4 |
| Recurrent Neural Network | 0.1293 | 15 |
| NBEATS | 0.0984 | 9 |
| NHiTS | 0.1208 | 9 |
| Temporal Fusion Transformer | 0.0968 | 113 |

In this case, the simpler models tended to perform the best. This makes sense, given the simple and small dataset. The more complex models may simply not have enough data to produce as good of predictions.




# DATASET 2- MULTIVARIATE

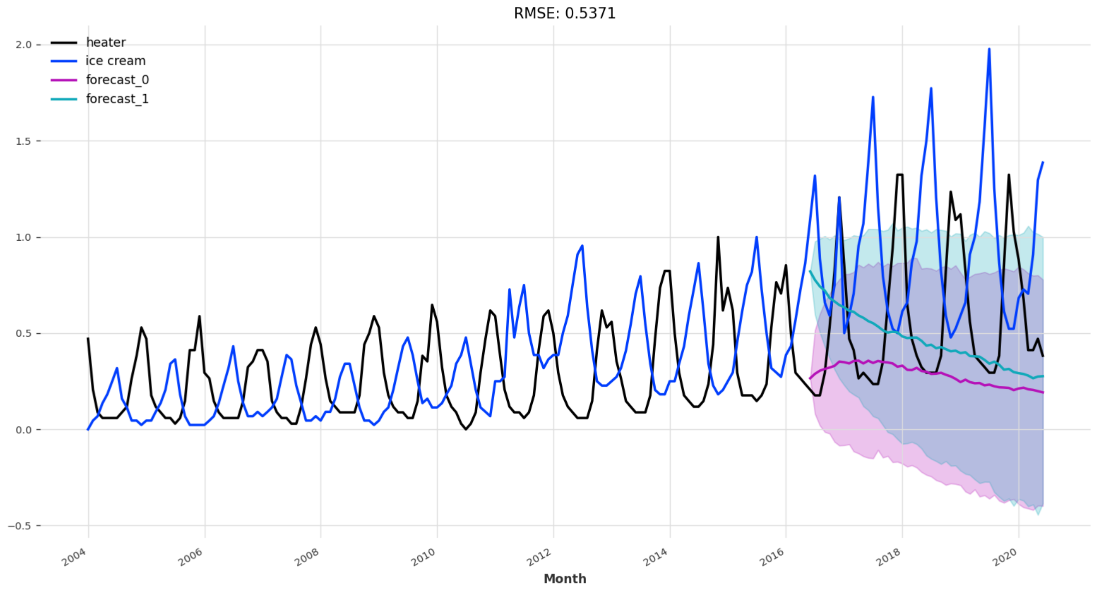
Monthly measurements of ice cream sales and heater sales over sixteen years

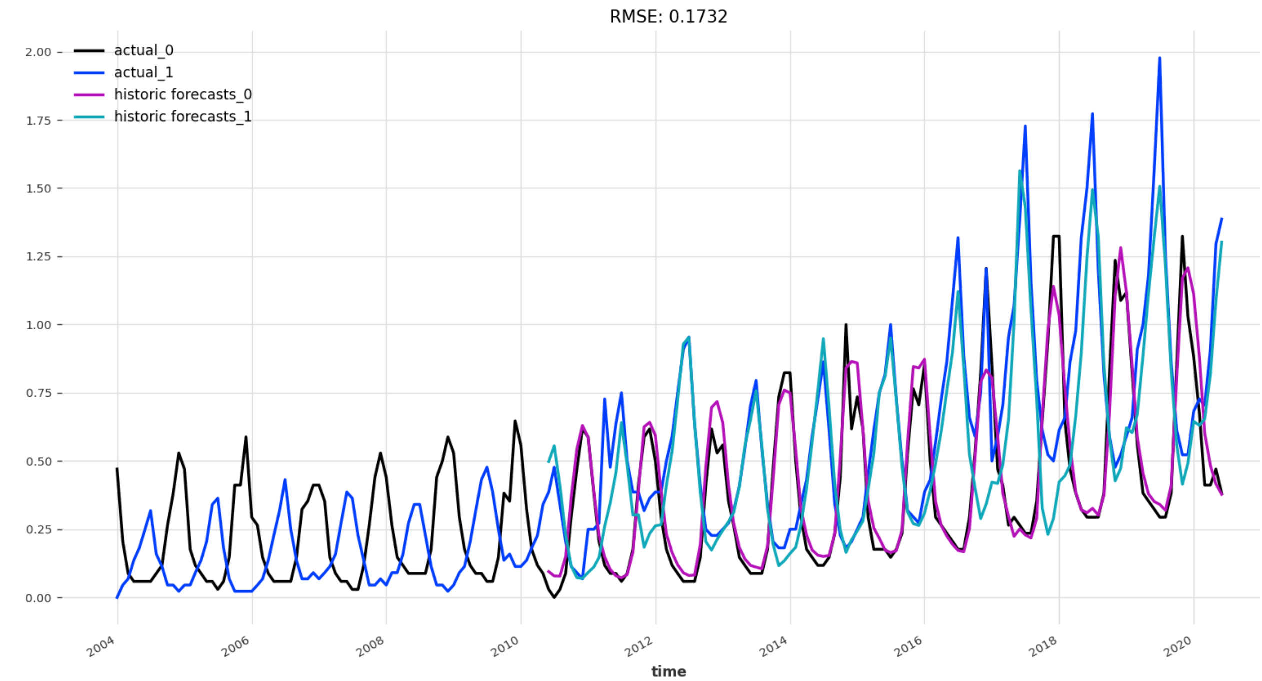
Use of models is limited- not every model supports multivariate series

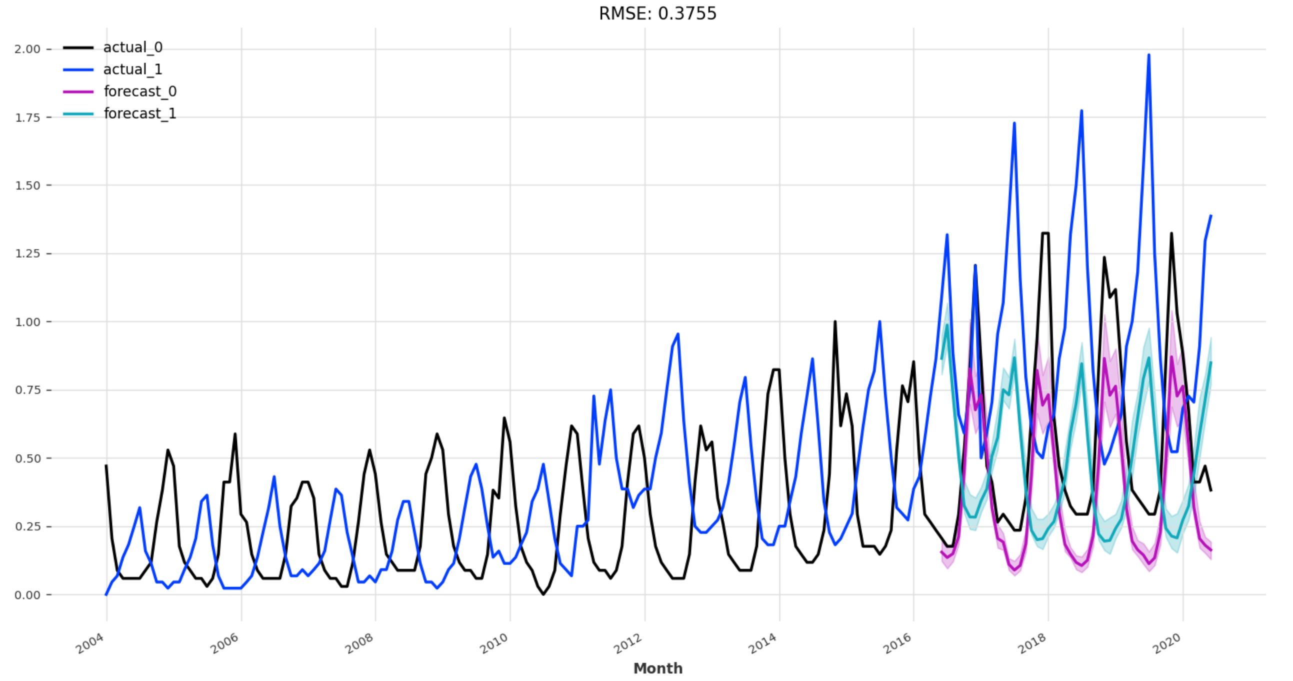
Goal is to predict both series with one model

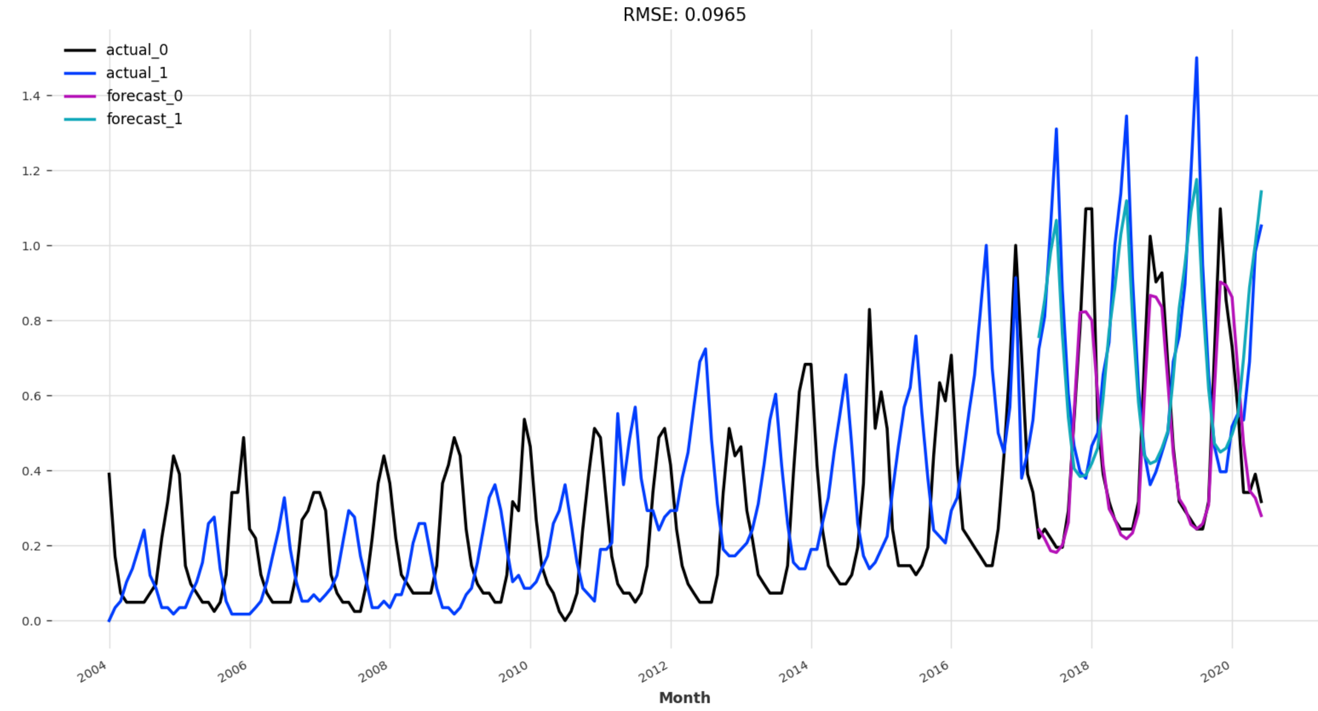
# VARIMA

- Uses ARIMA model, but generalized to multivariate series
- Unable to predict series at all





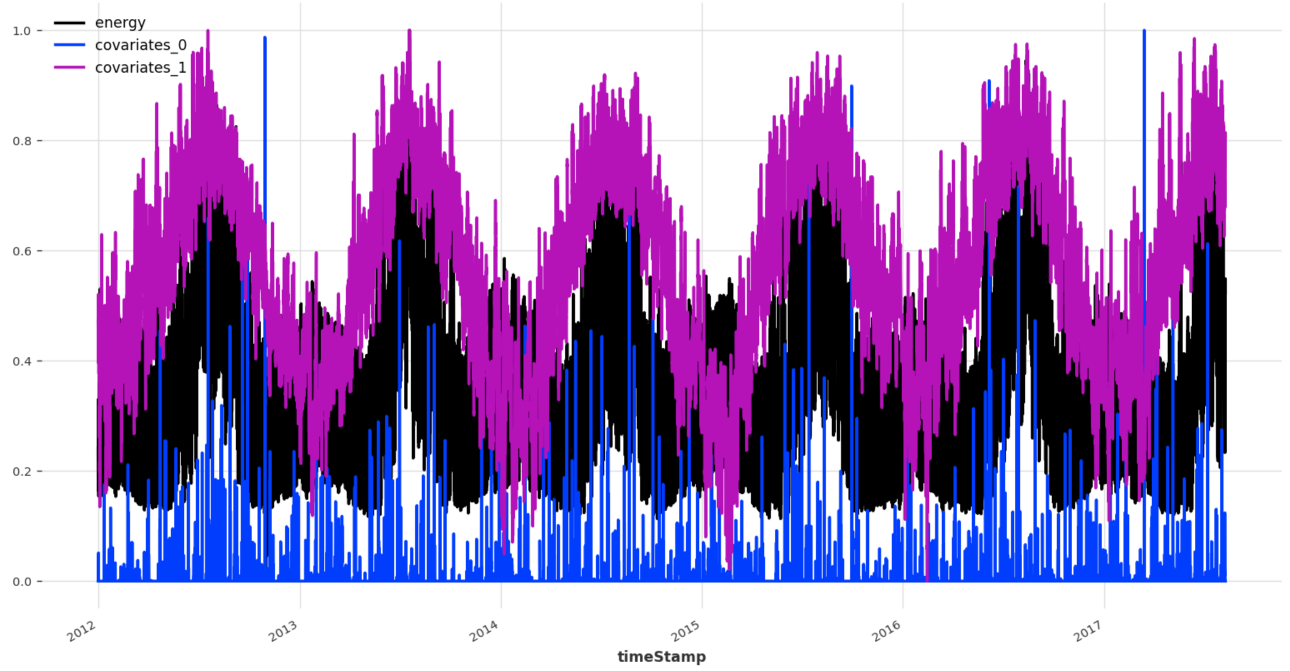

Top left: NBEATS
Top right: TFT
Bottom right: RNN

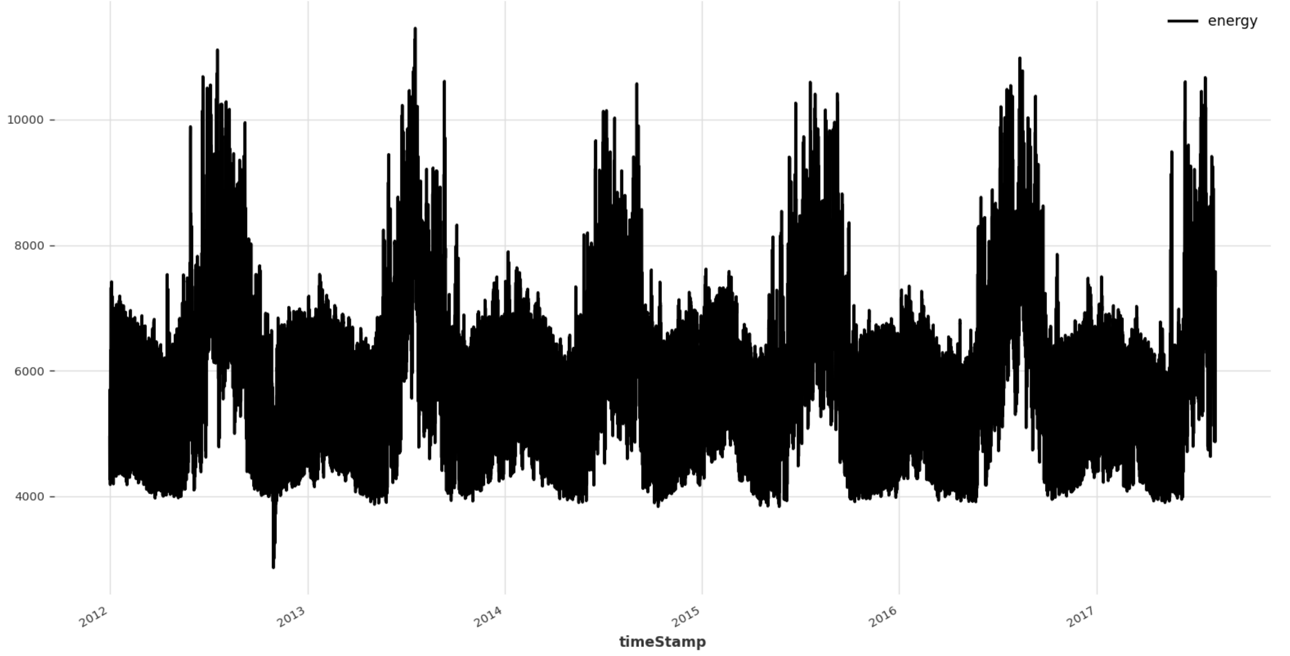
# OVERALL RESULTS- DATASET 2

| Model | Root Mean Square Error | Time to Train (seconds) |
|---|---|---|
| VARIMA | 0.5398 | 5 |
| Kalman Forecaster | 0.3968 | 4 |
| Regression | 0.2118 | 4 |
| Random Forest | 0.4519 | 5 |
| Recurrent Neural Network | 0.0965 | 26 |
| NBEATS | 0.1732 | 11 |
| NHiTS | 0.1806 | 11 |
| Temporal Fusion Transformer | 0.3707 | 135 |

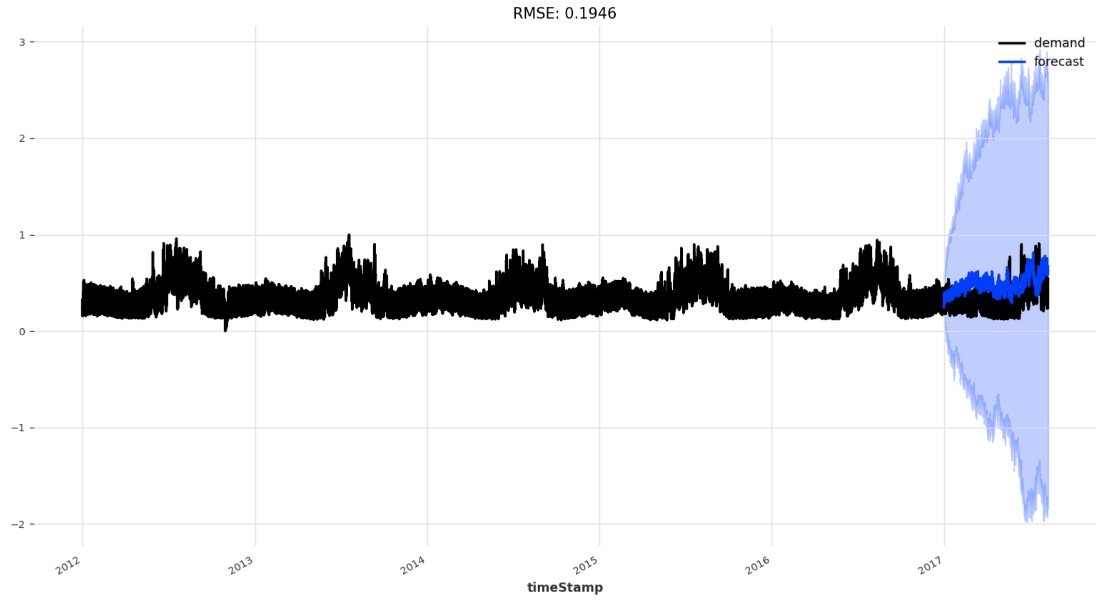
# DATASET 3- COVARIATES

- The third dataset is a large increase in size and complexity from the first two
- Features hourly readings of energy usage covering several years, for around 50,000 rows
- In addition to time and energy, temperature and precipitation are also recorded as covariates
  - Covariates are additional variables known throughout the entire time series that aid in the prediction



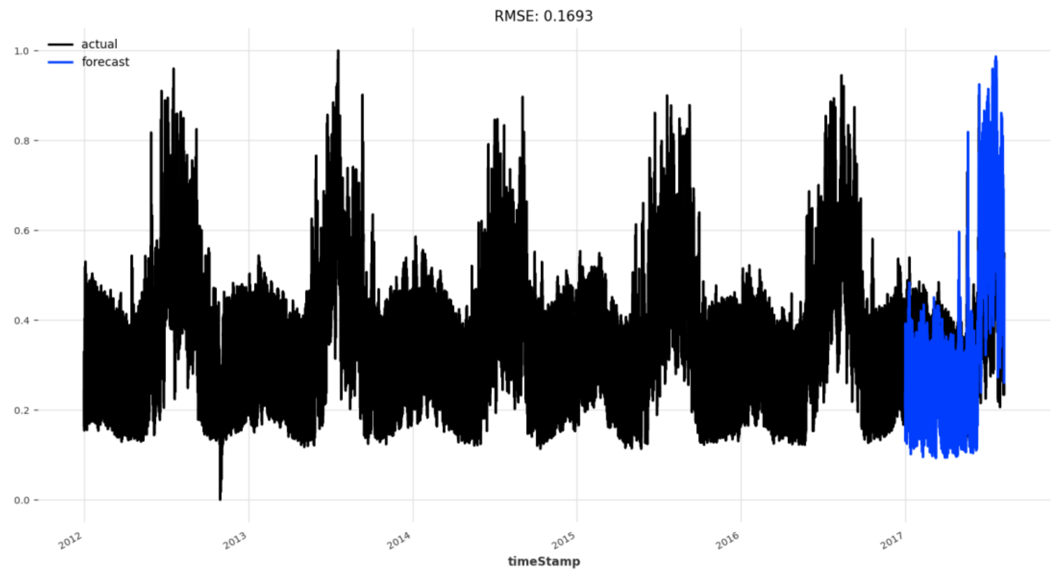



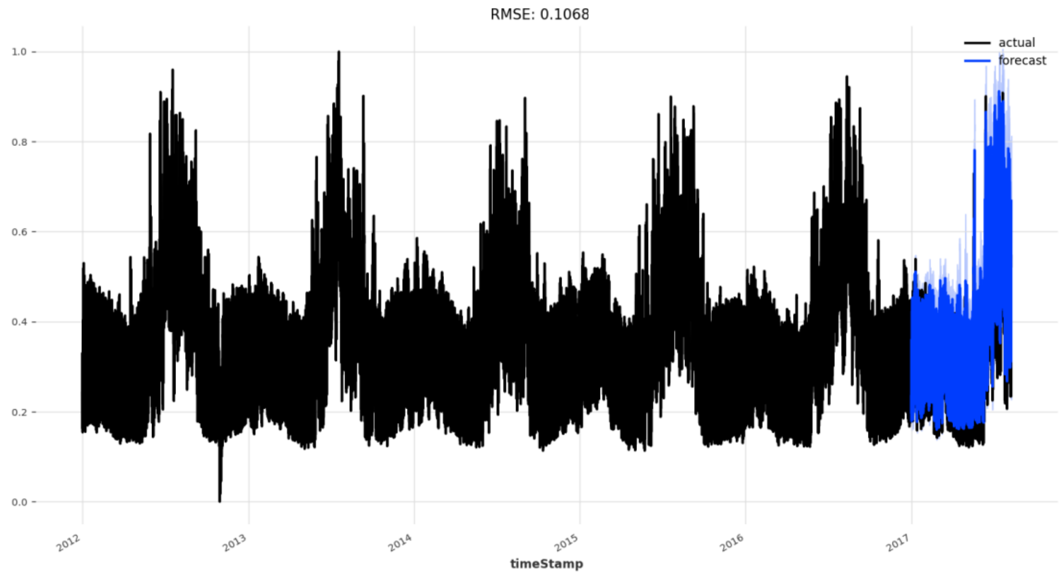
# ARIMA




- Does not perform very well
- This dataset may be too big/complex for ARIMA's relatively simple approach

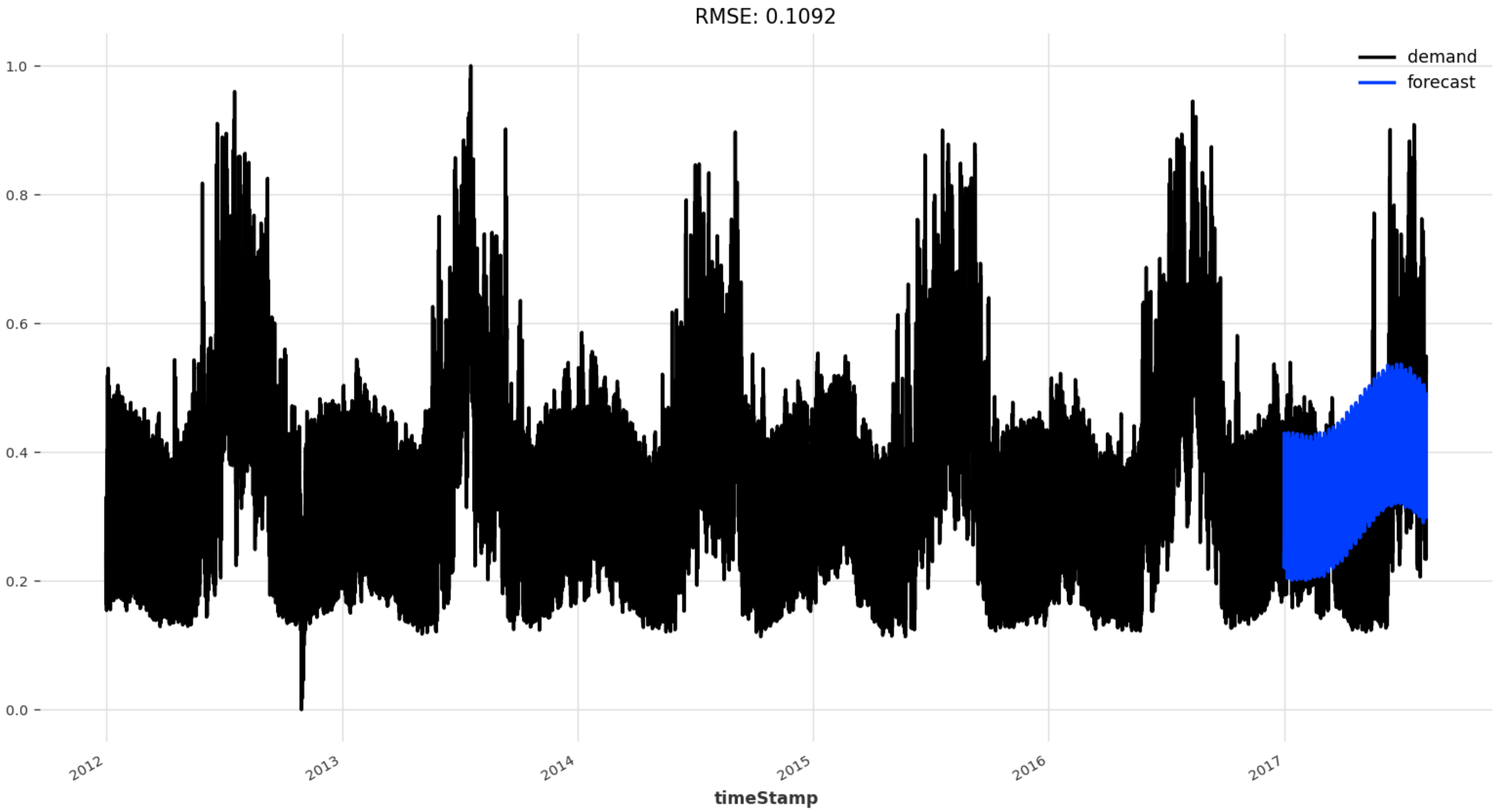
# RNN (LEFT) AND TFT (RIGHT)



# REGRESSION

- Achieves very low RMSE by simply modeling the trend of the data rather than specific data points



# OVERALL RESULTS- DATASET 3

| Model | Root Mean Square Error | Time to Train (minutes) |
|---|---|---|
| ARIMA | 0.1946 | 2 |
| Regression | 0.1092 | 0.5 |
| Recurrent Neural Network | 0.1693 | 8.95 |
| Temporal Fusion Transformer | 0.1068 | 65.37 |

## Finding good datasets

- It is hard to find publicly available datasets, especially of a particular size or complexity
- Most of the datasets used were provided by the Darts library, which provided the framework for the models as well

## Finding good accuracy metrics

Mean Absolute Percentage Error was used at first, but had a noticeable bias in favor of predictions that are too low versus predictions that are too high

- Other metrics used were Mean Absolute Scaled Error, R2 Score, Coefficient of Correlation, and Root Mean Square Error
- RMSE was chosen for consistency and ease of use, good as long as the data is normalized

# CHALLENGES FACED

# CONCLUSIONS

After testing the first dataset, I believed that the simpler, time series-specific models would be a better choice than the more complex, general machine learning methods

After testing with multivariate series and covariates, this proved not to be true

ARIMA and VARIMA produced significantly worse results than the RNN or TFT

Many of the time series-specific models were also less open to customization and did not allow for covariates or multivariate datasets

# SO, WHICH IS THE BEST?

- If required to pick one model that works the best under a variety of circumstances, it would be:

  Recurrent Neural Network

- Runner Ups:
  - NBEATS: always produces great predictions with its unique expanding training window, but for long series, it can have extensive training and prediction times
  - Temporal Fusion Transformer: worked the best on the energy dataset and worked well with the others. Very flexible but longest training times
  - For a simple, time series-specific model with no covariates or extra predictions, TBATS performs the best

# QUESTIONS?