Report Draft

Andrew H., Grace C., Nick H., Emily P.

2024-12-02

Introduction

While pondering a topic, we discovered that our group shared a mutual distaste for the random scattering of E-Scooters and how seemingly popular the pay-to-ride service was becoming in Fort Collins. This idea sparked our interest - we wanted to know just how popular these scooters have become over the last few years. Our team reached to reach out of Fort Collins Transportation Planner, Rachel Ruhlen, whom gave us the data that SPIN had been gathering since the service's launch in 2021.

Abstract

Fort Collins is widely considered one of most commuter friendly cities. As the push for Eco-friendly transportation methods rise, local E-Scooter/Bike companies like SPIN, are starting to become more widely available. This report aims to use statistical machine learning methods - Autoregressive integrated moving average (ARIMA), K Nearest Neighbors (KNN), Seasonal-Trend Decomposition (STLM), Boosting, etc… - and spatial geographical data to understand and forecast the longevity of ridership.

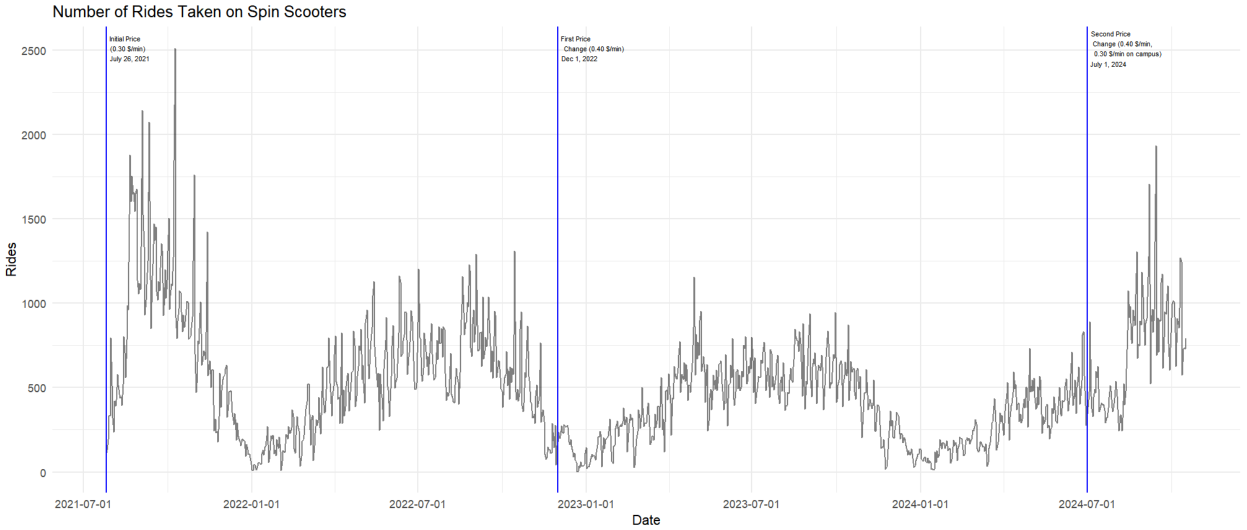
Data

The models and analyses was conducted on a collection of .csv files provided to this project by the City of Fort Collins Website and Rachel Ruhlen. For the purpose of describing the data files, the last portion of each file will be the reference name. The 'deployment' CSV consists of the latitude and longitude of multiple scooter deployments - there are many discrepancies in this CSV where either negative lat/long causes a deployment outside of the Fort Collins service area. A deployment is described as the location where SPIN decides to drop a scooter off to the public. The "trip ends" and "trip starts" CSV also provides latitude and longitude data based on where a scooter trip is initiated or ended by a customer. The "metrics-data" tracks metrics on a quarterly basis since Q3-2021 (the initial launch date) - variables included within this CSV are: Median Trip Duration (minutes) and Distance (miles), Average Trip Duration (minutes) and Distance (miles), Average Trips/Day, and Total Trips.

The "Routes-Data" contains variables "Segment ID", Percent and Count Matched Trips, and Segment Geometric ID. The "Quarterly Match Trip" folder consist of the same "Routes-Data" CSV files split up by quarter. These CSV's had multiple inconsistencies on the "Segment ID" variable - the worst collection of "NA" names totaled to 9800 making it difficult to understand precise streets where trips start.

Methods

Using the "analyze trips by date" CSV and "spin prices" a visualization was performed in order to obtain a general understanding of the data we had - this served as a starting point for the rest of our project. We noticed a gradual downward trend in ridership from the initial launch of SPIN with spikes in usage when prices decreased and towards the start of the fall semester.



Using the "Date" column, we created month, year, and week number columns. We used separate data frames for weekly and monthly data. In each of these data frames, we grouped by the time frame of interest to get the sum of scooter rides in each interval. We then created two new columns for the normal and college prices.

To forecast future ridership, a time series analysis was performed. In both analyses data from 2024 was used as testing data, and previous data was used as the training data in the form of a time series. The accuracy of the models in both analyses was measured using the mean absolute percentage error (MAPE). It is the average magnitude of error produced by a model; in other words, it measures "the average absolute percentage difference between the predictions and the actuals" (Roberts). This is a commonly used measurement for accuracy in time series analyses because of its simplicity and interpretability (Roberts). For our analysis, it allowed us to compare the errors in the weekly and monthly forecasts on a comparable scale. Only the most accurate models (meaning those with the lowest MAPE values) are included in this paper.

The forecast package in R has many functions for time series forecasting, including those for STLF and ARIMA methods. The forecastHybrid package in R allows several kinds of time series forecasting methods to be combined. The options are ARIMA, STLM, TBATS, NNETAR, ETS, and THETAM. Models created using this package were created in both the weekly and monthly forecasts, and they are referred to as hybrid models and the methods used are specified.

In both analyses, several models generated from time series forecasting methods were tested. One of these models was an STLF model. STLF (Seasonal and Trend Decomposition using LOESS Forecasting) decomposes the time series into trend, seasonal, and residual components using STL (Seasonal and Trend Decomposition using LOESS) (Andrés & Andrés). It generates a forecast by deseasonalizing the time series data by subtracting the seasonal component, applying a non-seasonal forecasting method (e.g., ARIMA or ETS) to the deseasonalized data, and reseasonalizing the forecasts by adding back the last year of the estimated seasonal component (Andrés & Andrés). This method is resistant to outliers, so these observations don't significantly affect estimates (Andrés & Andrés).

Autoregressive integrated moving average (ARIMA) is a type of time series analysis used for understanding historical data and predicting future values (Wikipedia contributors). Autoregressive (AR) means "that the evolving variable of interest is regressed on its prior values" (Wikipedia contributors). Moving average (MA) means "that the regression error is a linear combination of error terms whose values occurred contemporaneously and at various times in the past" (Wikipedia contributors). Integrated (I) means "that the data values have been replaced with the difference between each value and the previous value" (Wikipedia contributors). ARIMA models are good at capturing seasonality, cycles, or trends (Pathan & M.), which was fitting for our data since there was clear seasonality in ridership.

Like STLF, the STLM model is a part of the STL (Seasonal and Trend decomposition using LOESS) framework. STLM (Seasonal and Trend Decomposition using LOESS Modeling) uses the same STL decomposition as the

STLF method (R: Forecasting using stl objects). However, the STLM method focused on decomposition and modeling of the seasonally adjusted data while STLF combines decomposition, modeling, and forecasting into one step. This means the STLM method provides more control over the modeling process (R: Forecasting using stl objects).

TBATS (Trigonometric seasonality, Box-Cox transformation, ARIMA errors, Trend, and Seasonal components) can accommodate non-linear trends in data, which makes it suitable for data with complex trend patterns and means the model is highly flexible. This is fitting for our data since we believed the price changes affected trend patterns. (Ellis).

Neural Network AutoRegression (NNETAR) can also capture complex non-linear trends. It typically trains multiple networks and combines their forecasts, which can improve prediction accuracy and stability (12.4 Neural network models). It handles seasonality by automatically selecting optimal seasonal and non-seasonal lags based on AIC (12.4 Neural network models).

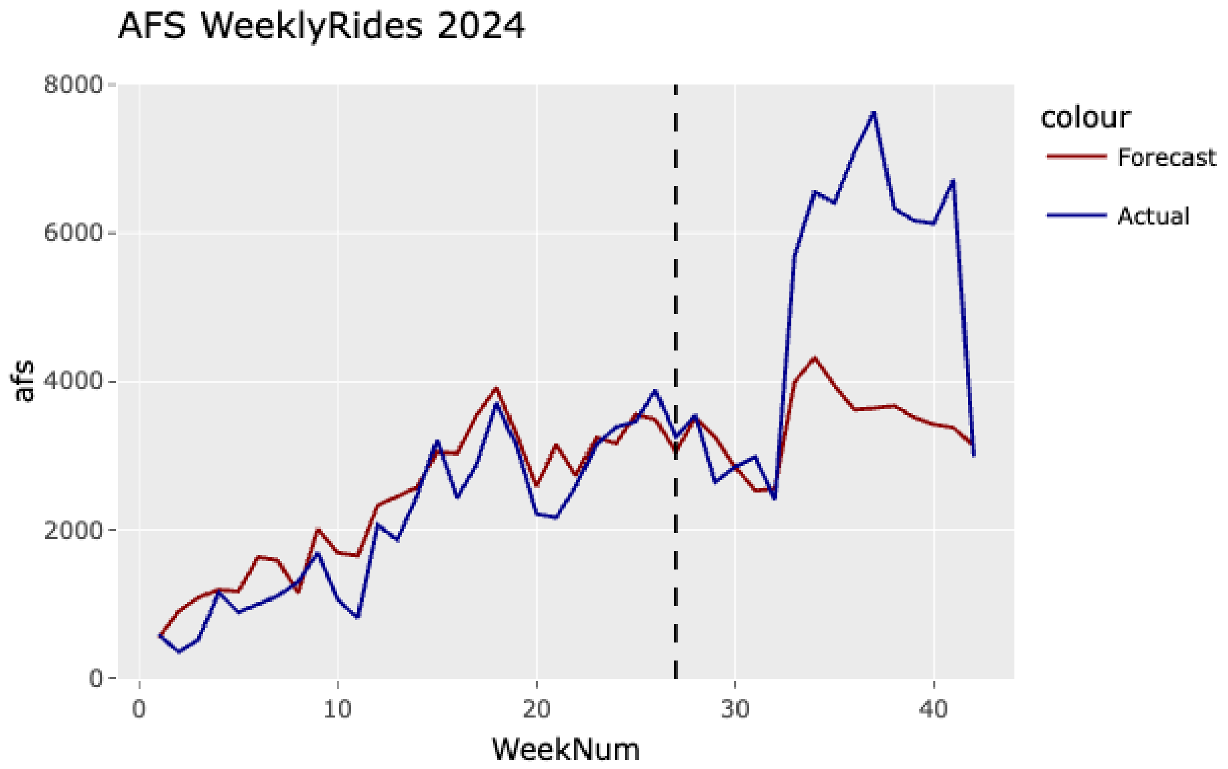
ETS (Error, Trend, Seasonality) models decompose timer series into three key components. Error can be Additive (A) or Multiplicative (M). Trend can be None (N), Additive (A), Additive dampend (Ad), Multiplicative (M) or Multiplicative dampend (Md). Seasonality can be None (N), Additive (A), or Multiplicative (M). This framework allows for flexibility in the combination of these components (Svetunkov). ETS generates forecasts by decomposing the time series into its components, extrapolating each component into the future, and recombining the components to produce the final forecast (7.7 Forecasting with ETS models). ETS can capture complex seasonal patterns, including seasonality and trends that change over time.

THETAM, a variation of the Theta method, is another flexible model that is good at capturing complex seasonalities and changes in trends. It is “based on the decomposition of the time series into three components: trend, seasonality and noise. The model then forecasts the long-term trend and seasonality, and uses the noise to adjust the short-term forecasts” (Castellon). It is often more accurate than other methods, particularly with complex trends and seasonality (Castellon).

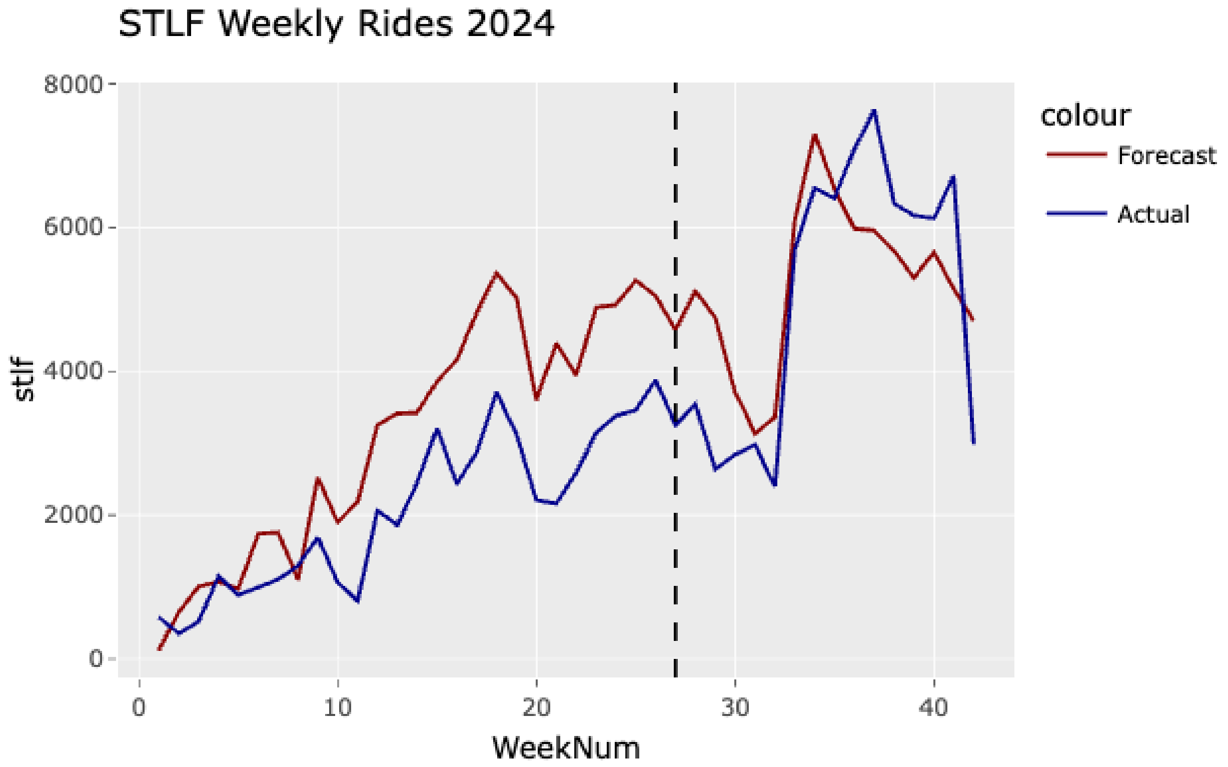
In both analyses, several models were tested, but only the most accurate models will be included in this paper. The college price decrease initiated in July 2024 caused a huge spike in ridership that was not captured by most models. Thus, in both analyses, the final model is a combination of two models: one that was accurate before the price change and one that was accurate after the price change. Both forecasts use the latter since the forecasts occur after the price change.

Weekly Rides

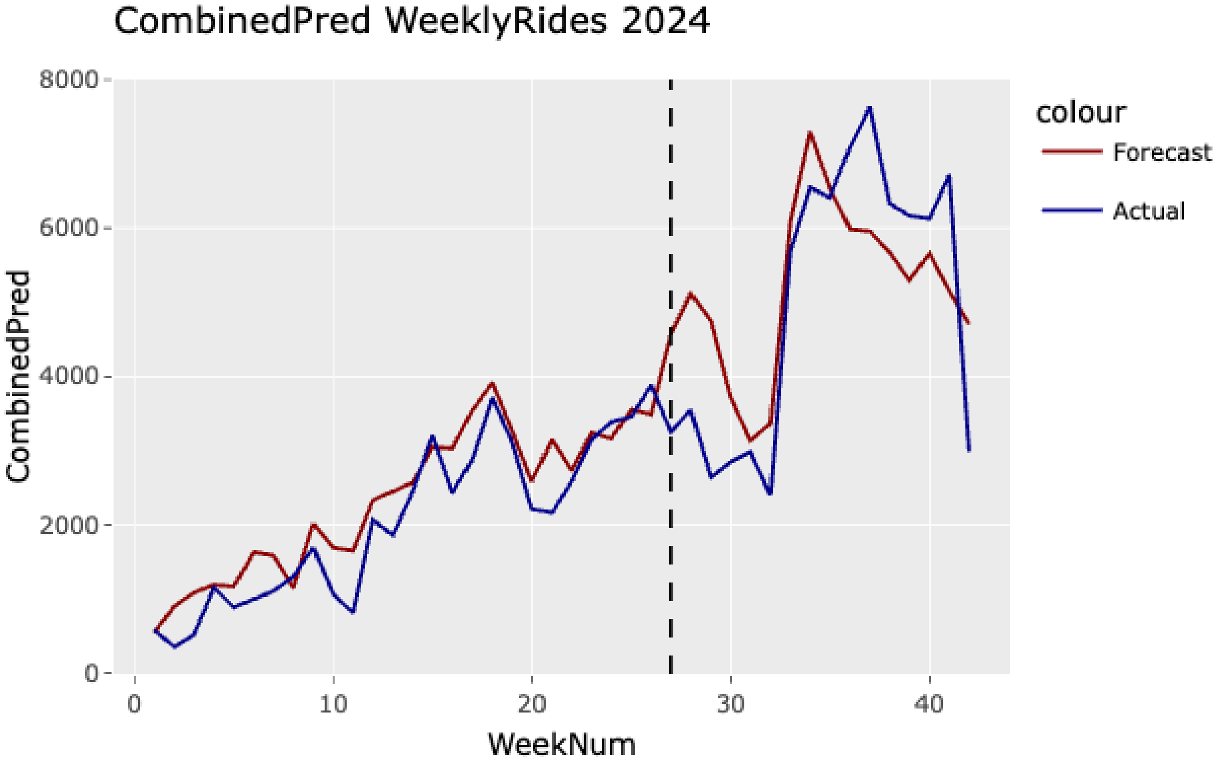
Before the price change, a hybrid model composed of an ARIMA, STLM, and THETAM model (weighted equally) most accurately predicted ridership when fit on the testing data, with a MAPE of 29.69. The plot below compares the actual ridership values to the forecasted values; the date of the price change is marked by a dotted black line.



An STLF model most accurately captured the spike (based on visually comparing the forecast graphs) in ridership when fit on the testing data, with an overall MAPE of 47.12.

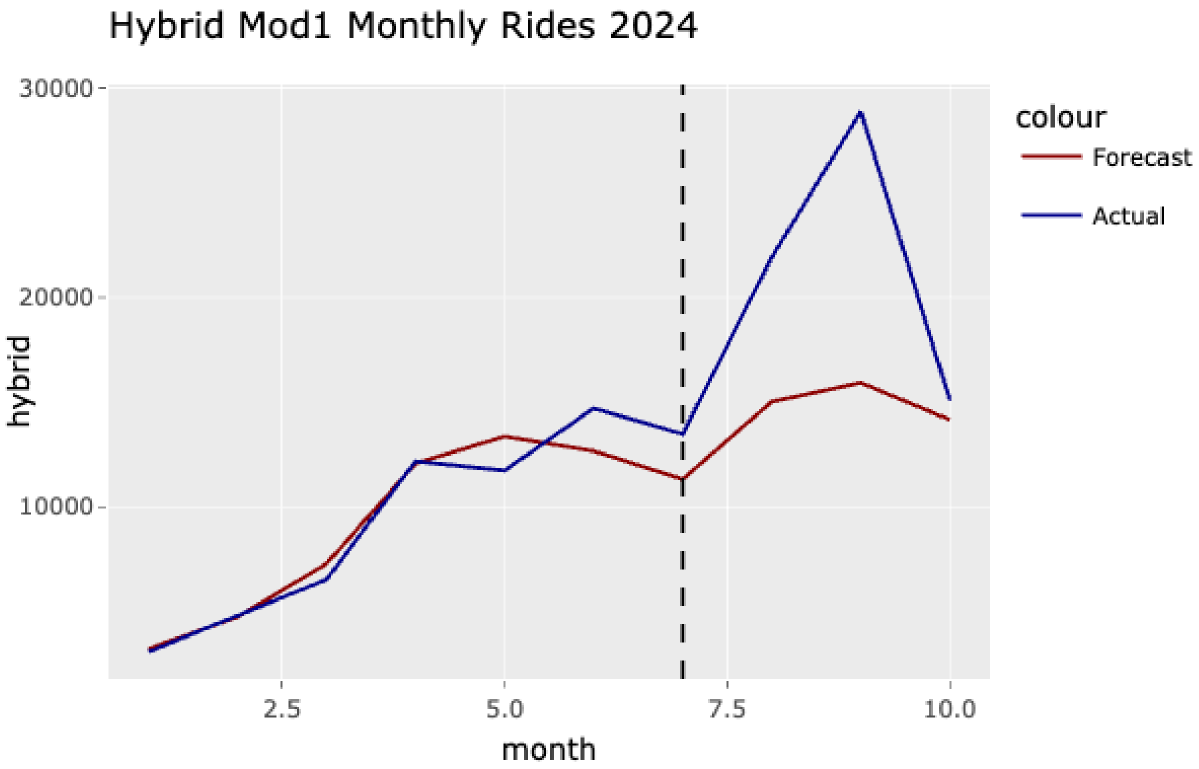


The final model for predicting weekly ridership uses the hybrid model before the price change and the STLF model after the price change. This model accurately captures the trend before the price change and the spike after the price change and has a MAPE of 29.04 on testing data.

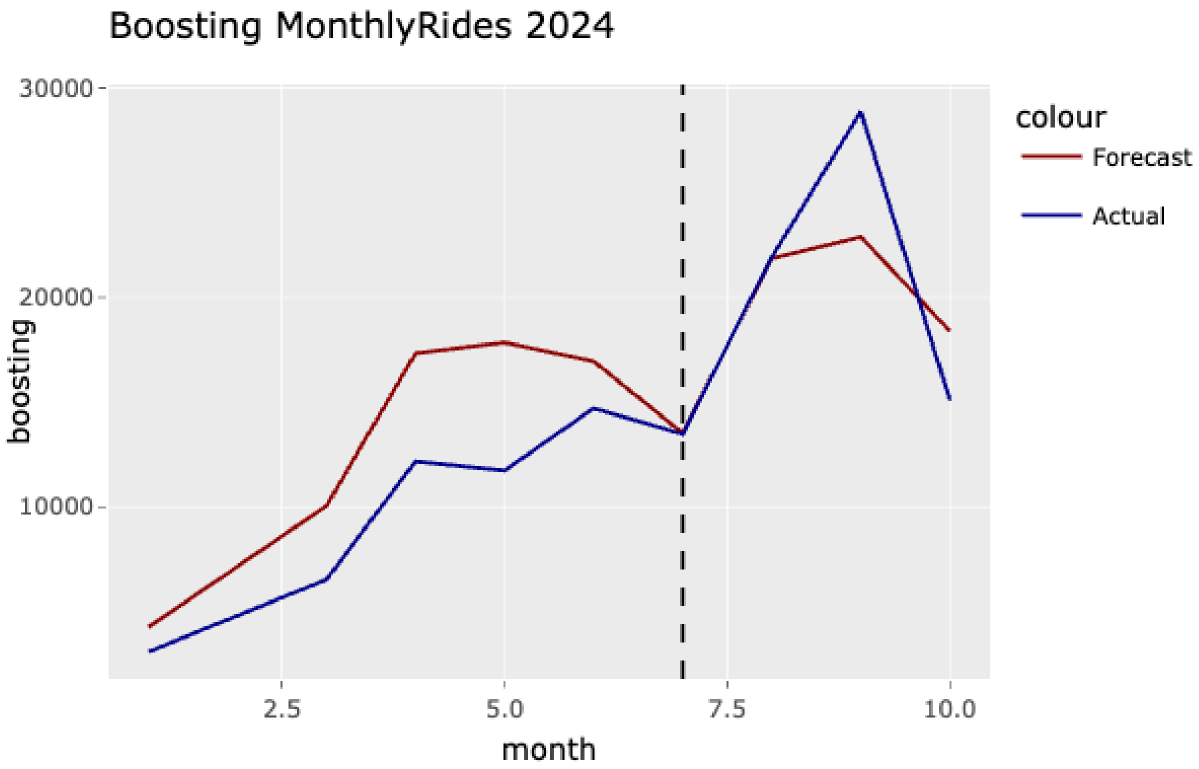


Monthly Rides

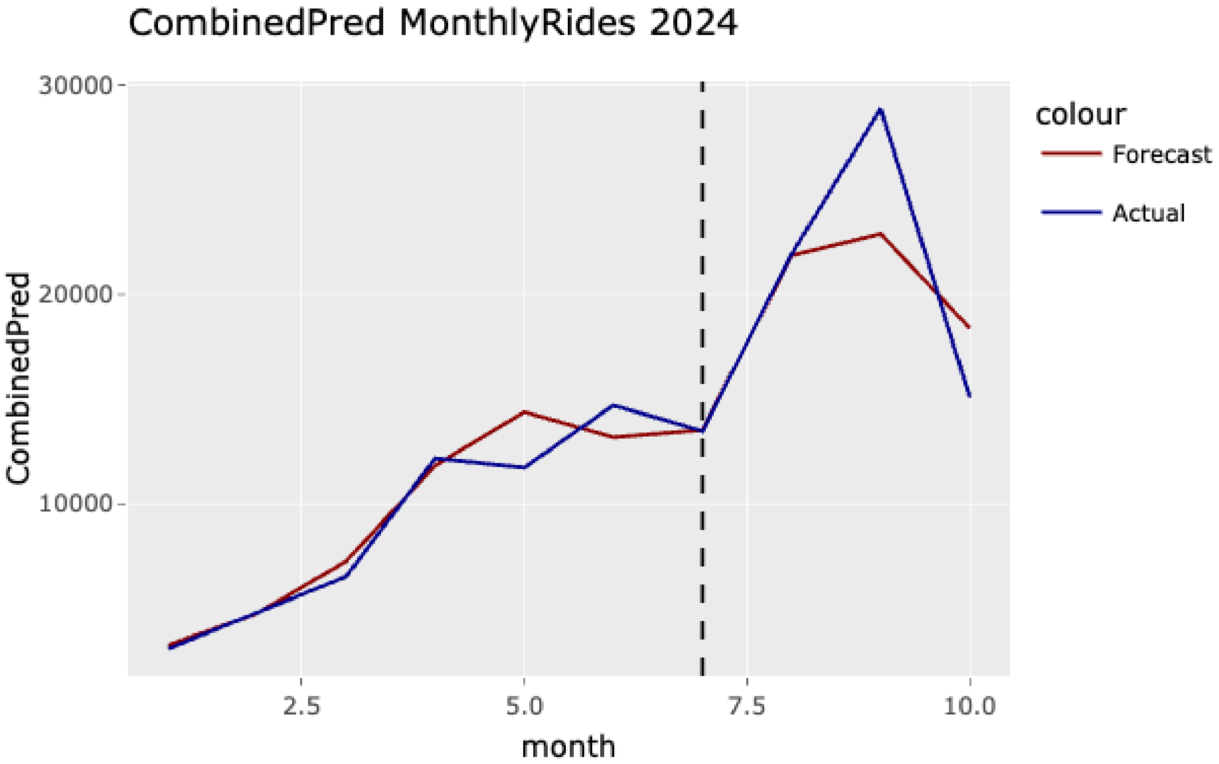
Before the price change, a hybrid model composed of an ARIMA, STLM, THETAM, NNETAR, ETS, and TBATS model (weighted equally) most accurately predicted ridership when fit on the testing data, with a MAPE of 14.39.



A boosting model most accurately captured the spike in ridership when fit on the testing data, with an overall MAPE of 29.39.



The final model for predicting weekly ridership uses the hybrid model before the price change and the boosting model after the price change. This model accurately captures the trend before the price change and the spike after the price change and has a MAPE of 8.86 on testing data.



KNN - Geospacial

SPIN provided our group with three separate "curb event" files containing geographic coordinates for the deployment, start, and end locations for spin scooters across Fort Collins. The rows included purely spatial data (latitude, longitude) that were unconnected to each other file-wise, as-in there was no way to tell which scooter starts at the pinged location. The data also included miscalculations like missing negatives that created large outliers in the data where certain scooters reported their start in places like Shanghai.

We mapped the data and compared the density score of the predicted results on a prediction grid of Fort Collins to determine if deployments are being distributed in an optimal matter. The model should help us determine if there are too many scooters being deployed in an area, and find areas where scooters are not being deployed enough to fit scooter demand.

A 20-fold cross validation was used to divide the map Fort Collins into 20 grid squares to compare our predictions. 20 was chosen as a tradeoff between complexity and accuracy. Both models selected an epanechnikov kernel and best k = 15. The deployment KNN model reported a means squared error of 0.04891901 and the start KNN model reported a means squared error of 0.04380498.

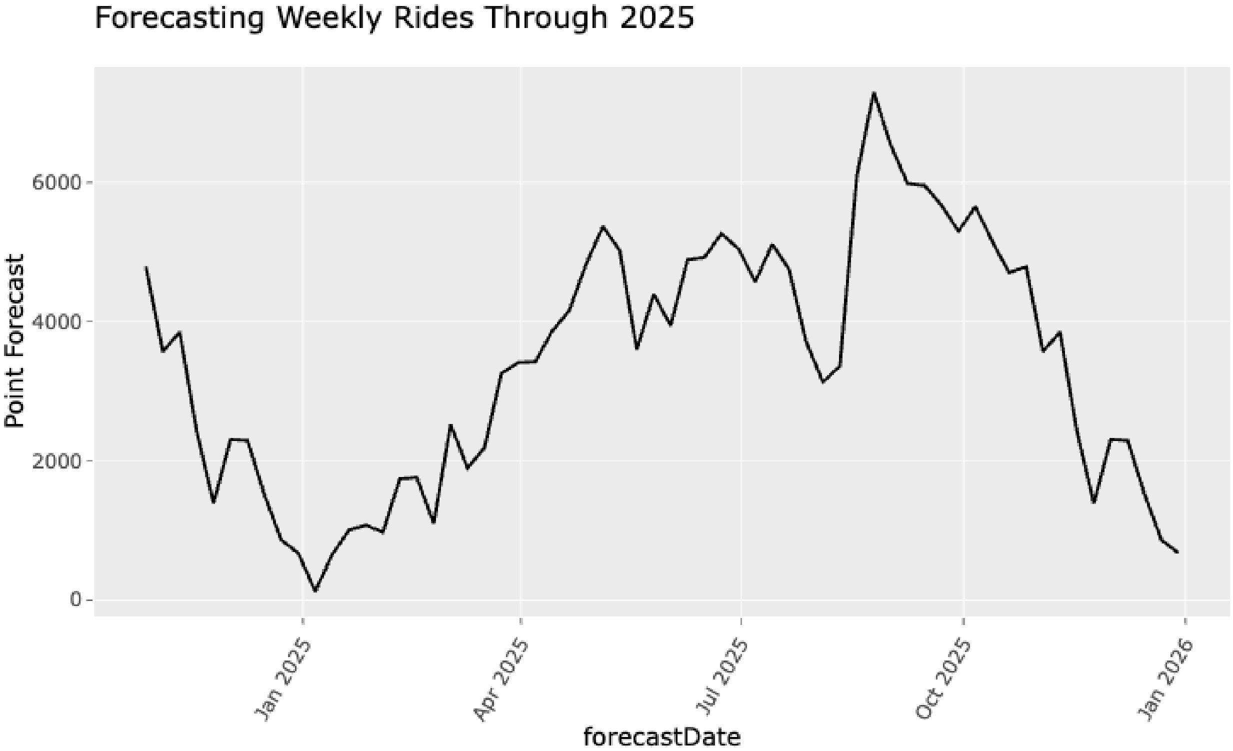
We made comparisons of the model predictions from our KNN regression to create a mesh of points across the map of Fort Collins. The deployment model predicts what deployment density should be at that location, and the start model predicts what the usage density should be. These represent the "expected" levels of deployment vs. usage, smoothed out to account for day – to – day variations.

Results

The predictions for future weekly and monthly ridership will be made assuming that the normal and college prices remain the same.

Weekly

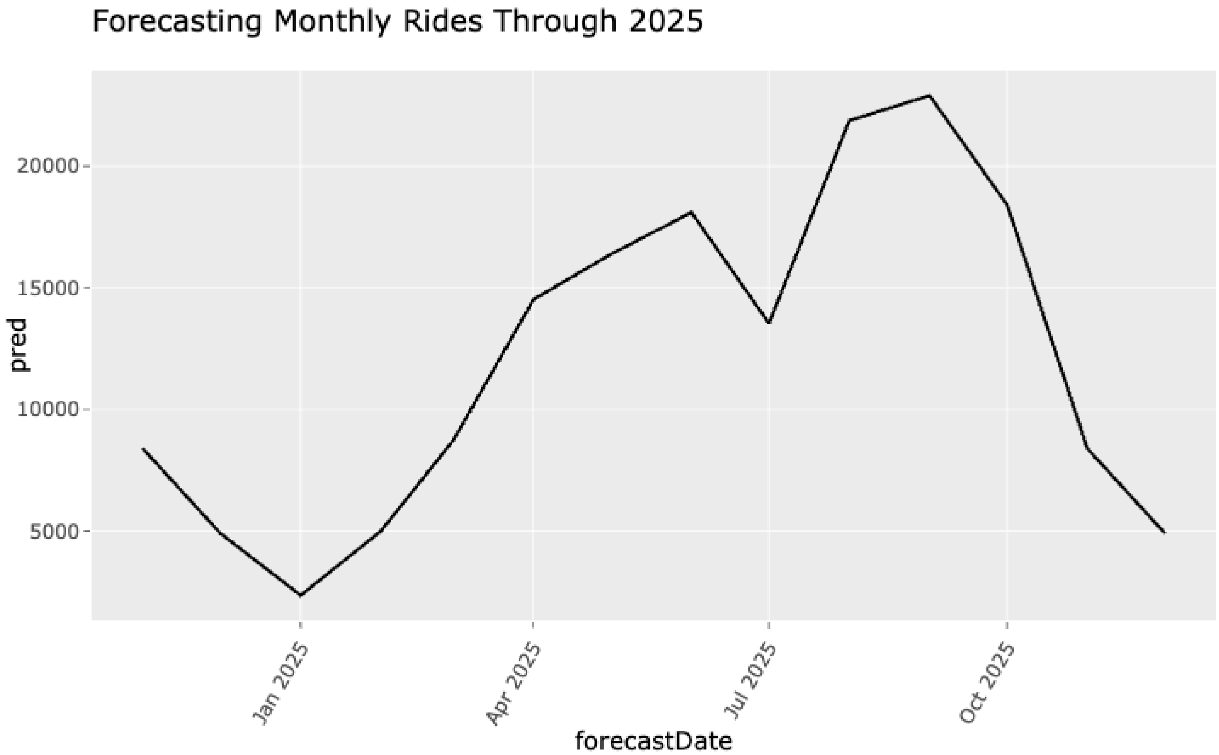
Since the STLF model is used for predicting after the price change, this model was used to predict ridership through the end of 2025. The plot is shown below, beginning at the end of the testing data.



In 2025, the week with the lowest ridership (around 122) will be the first week of January 2025 and the week with the highest ridership (around 7300) will be the week of August 25. The January prediction seems slightly low. The August prediction seems accurate, since it is slightly higher than that week in the previous two years.

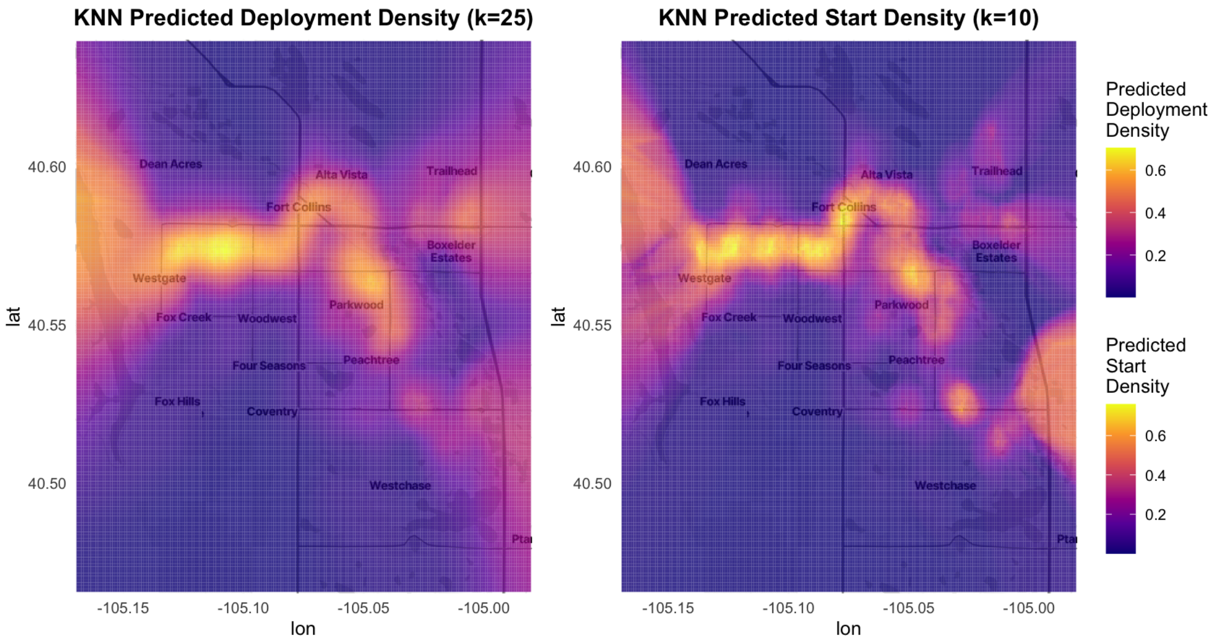
Monthly

Since the boosting model is used for predicting after the price change, this model was used to predict ridership through the end of 2025. The plot is shown below, beginning at the end of the testing data. However, since this model does not incorporate a way to track a trend, it will make the same predictions for each individual month in the future (i.e., all January predictions will be the same).



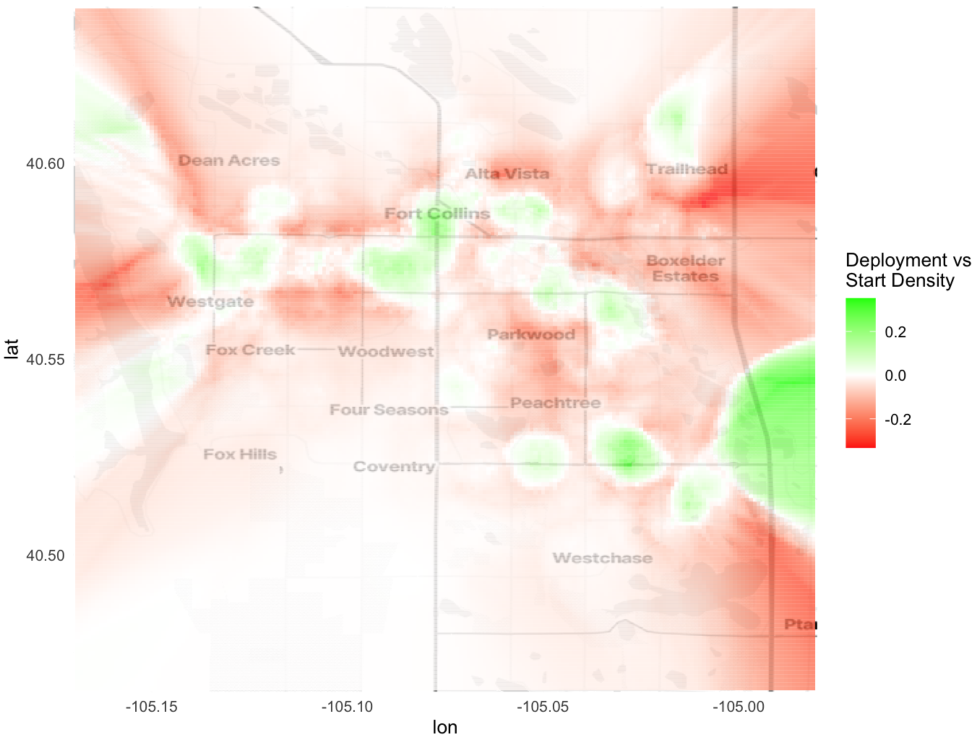
In 2025, SPIN can expect January to be the month with the lowest ridership (around 2350) and September to be the month with the highest ridership (around 22900). The January prediction is slightly lower than all previous January values. The September prediction is about the same as the September 2022 value.

KNN Deployment & Start Heatmaps



Compared side by side, the two models bear a strong resemblance to each other with the deployment model showing a more gradual gradient compared to the start model. This matches with the observations when comparing the raw data. There are notable edge effects in areas where the model is predicting past the SPIN service area in Fort Collins.

KNN Deployment Optimization Map



This map highlights where SPIN can improve their deployment strategies. Smooth transitions between colors on the map represent gradual changes in the relationship between deployment and usage patterns, which helps understand not only which areas to make changes, but just how significant those changes should be.

The model is less effective reaching toward the outer bounds of the service area in the data. As it reaches the outer bounds an edge effect takes hold and spreads the last noticed pattern past the service area of the data into spaces where no scooters are used.

Works Cited

7.7 Forecasting with ETS models | Forecasting: Principles and Practice (2nd ed). (n.d.). https://otexts.com/fpp2/ets-forecasting.html 12.4 Neural network models | Forecasting: Principles and Practice (3rd ed). (n.d.). https://otexts.com/fpp3/nnetar.html Andrés, D., & Andrés, D. (2024, January 5). Time Series Forecasting with STL - ML Pills. ML Pills - Machine Learning Pills. https://mlpills.dev/time-series/time-series-forecasting-with-stl/ Castellon, N. (n.d.). Optimized Theta Model. NIXTLA. https://nixtlaverse.nixtla.io/statsforecast/docs/models/optimizedtheta.html Ellis, C. (2023, July 2). When to use TBATS - Crunching the Data. Crunching the Data. https://crunchingthedata.com/when-to-use-tbats/ Pathan, A., & M., N. (2023, September 1). What are the advantages and disadvantages of using ARIMA models for forecasting? https://www.linkedin.com/advice/0/what-advantages-disadvantages-using-arima R: Forecasting using stl objects. (n.d.). https://search.r-project.org/CRAN/refmans/forecast/html/forecast.stl.html Roberts, A. (2023, February 10). Mean Absolute percentage error (MAPE): what you need to know. Arize AI. https://arize.com/blog-course/mean-absolute-percentage-error-mape-what-you-need-to-know/ Svetunkov, I. (2024, November 29). 4.1 ETS taxonomy | Forecasting and Analytics with the Augmented Dynamic Adaptive Model (ADAM). https://openforecast.org/adam/ETSTaxonomy.html Wikipedia contributors. (2024, October

9). Autoregressive integrated moving average. Wikipedia. https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average