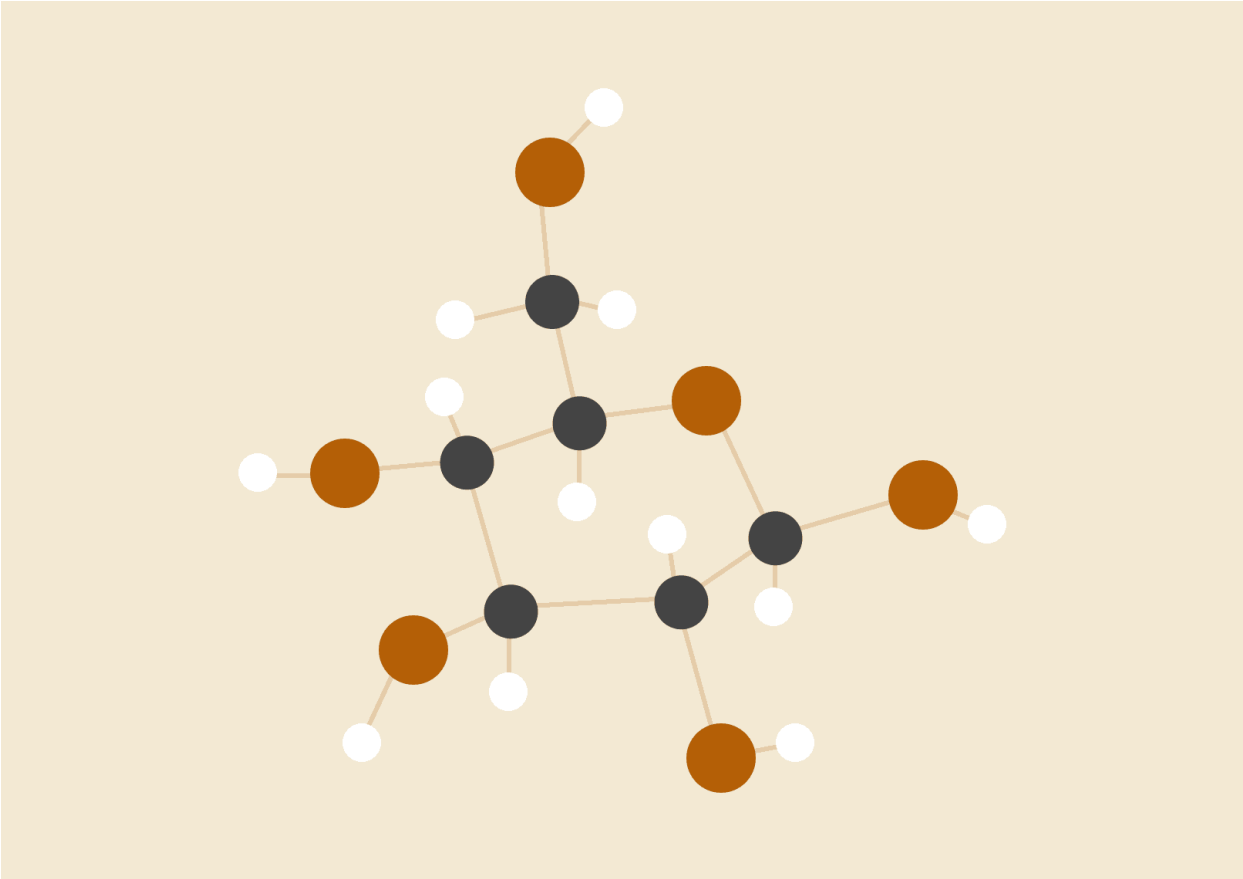
Forecasting Job Openings in the United States



By Qihan Liu, Kaiting Liang, Jingyi Guo, Yina Luo

03/04/2023
MAS640

Table of content

Introduction

The purpose of this document is to provide a walkthrough of the time series analysis conducted to forecast job openings in the United States. The analysis includes fitting different models to historical data, assessing their performance, and selecting the final model. This report does not include code but will present relevant output, interpretation, and discussion of the analysis process.

Data Description

The dataset used in this analysis contains monthly job opening data for the United States from January 2002 to December 2021, which is a total of 240 months. This data contains 5 variables: date, openings, layoffs, spend, and employees. We are going to forecast job openings in the U.S. in the future 12 months.

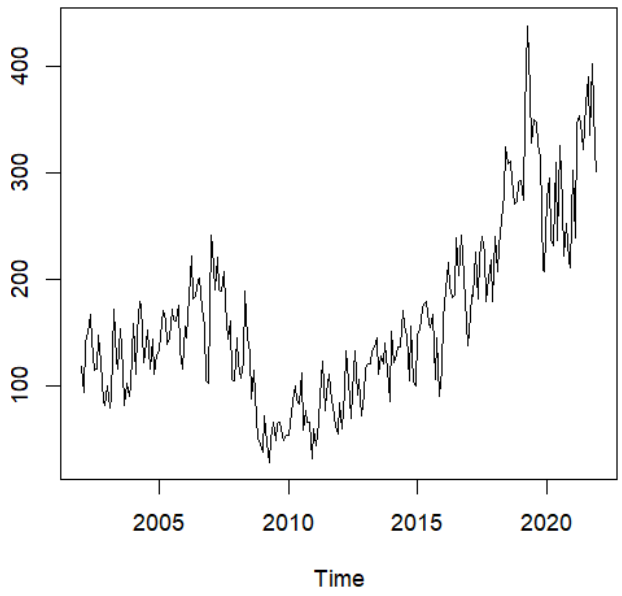
Data Preprocessing

A. Data cleaning and transformation

   The dataset used in this analysis does not contain any missing values, and the data appears to be continuous. Therefore, we convert these 4 variables into a time series format for further analysis and modeling.

B. Time series visualization

   To better understand the underlying patterns and trends in the data, we plot every month's job openings to see if there are any trends and seasonality.



From the time series plot, it can be observed that job openings in the U.S. exhibited a relatively volatile pattern before the year 2010, with frequent ups and downs, and doesn't have a specific trend. Job openings went down drastically in 2008 and 2009, maybe because of the Financial crisis in 2008. However, from 2010 onwards, there is a clear and consistent upward trend in the number of openings. And the number of job openings fluctuated abnormally and peaked in 2019.

There appear to be some less obvious seasonal patterns in the time series, we'll find out whether there's seasonality in our further work.

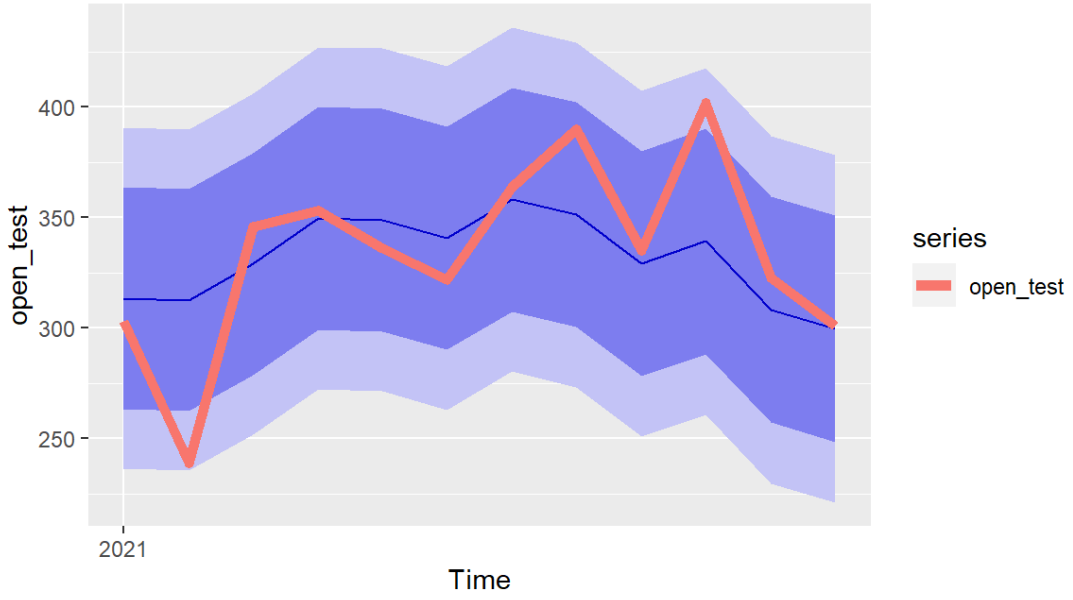
Model Fitting

As the trend is totally different between data before 2010 and data after it, we decide to only use data after 2010 for our analysis. We choose data from January 2010 to December 2020 as our training data and data from January 2021 to December 2010 as our testing data.

A. Linear regression

In the linear model, the dependent variable is open_train, which is regressed on two

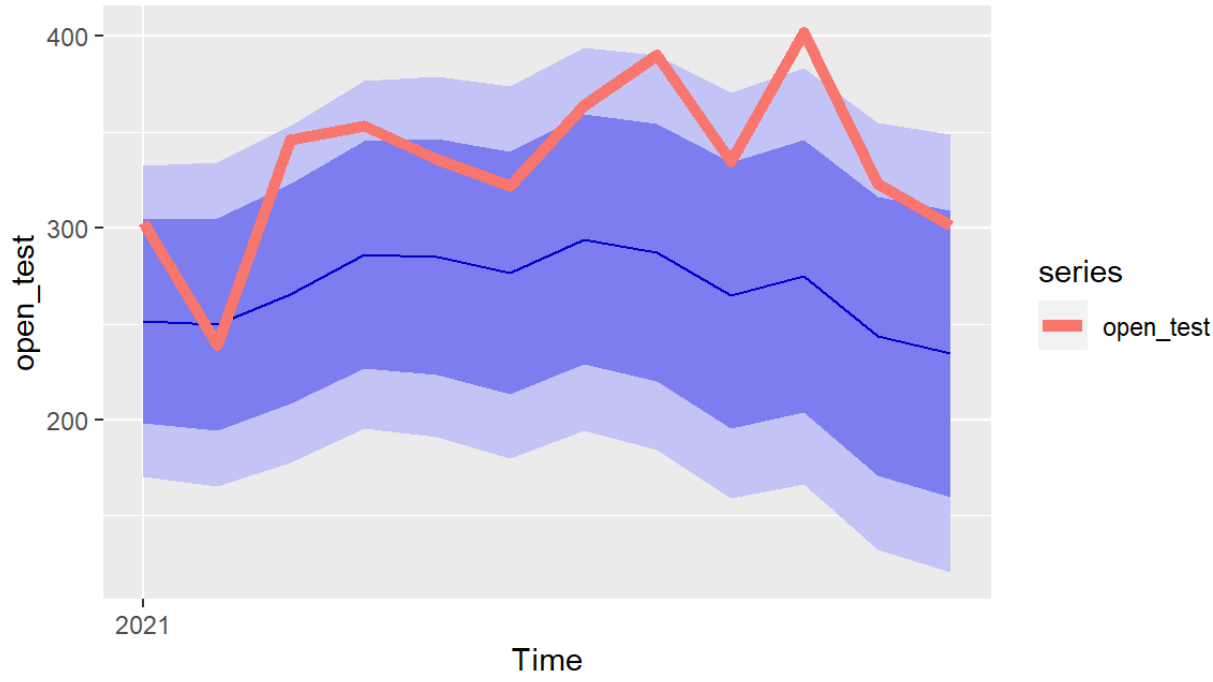
variables: a quadratic trend and a seasonal factor. The coefficients of the model are rounded to one decimal point and printed. We generated a forecast for open_train for the next 12 periods, using the estimated regression model. And then plot the fitted values of the linear regression model for the original time series data as well as the predicted values for the test period and the test data. Finally, calculate the accuracy of the predicted values versus the actual test data.



The RMSE we calculated by linear regression is 31.68383. That means the average difference between the predicted and actual values of the model was 31.68 which means the model fit was not bad.

B. STL(Seasonal Trend Decomp with Loess)

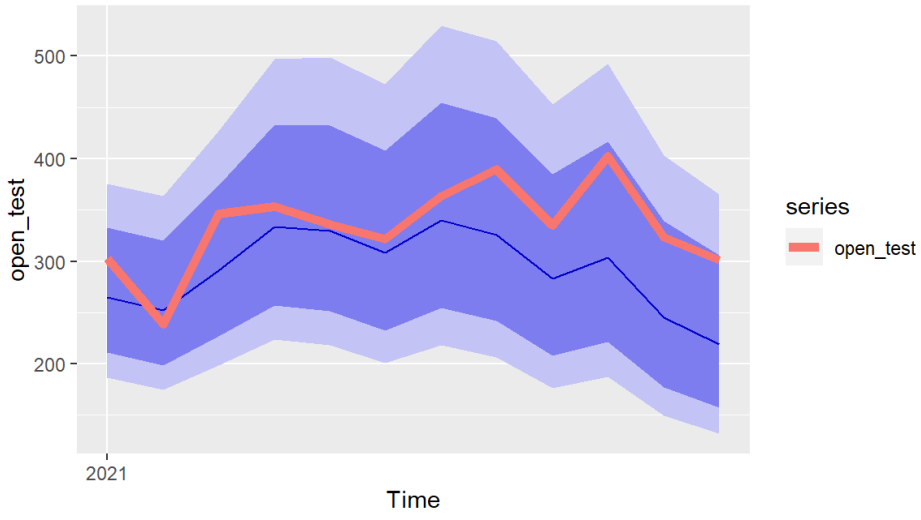
In the STL model, the parameter uses a periodic filter to estimate the seasonal component. Same as linear regression we forecast for the next 12 periods. However, the STL model has some differences, decomposing the seasonal, trend, and remainder components obtained from the model to calculate the fitted model for the time series data.



The RMSE we calculated by STL is 73.96735. This is already over 50, so we think that it indicates a large error between the predicted and actual values.

C. ETS(Exponential Smoothing)

ETS, same with STL to capture trend, and seasonality, in the time series. The difference is it uses exponential smoothing instead of LOESS. Here we log the open_train to help ETS better estimate the trend and seasonality.



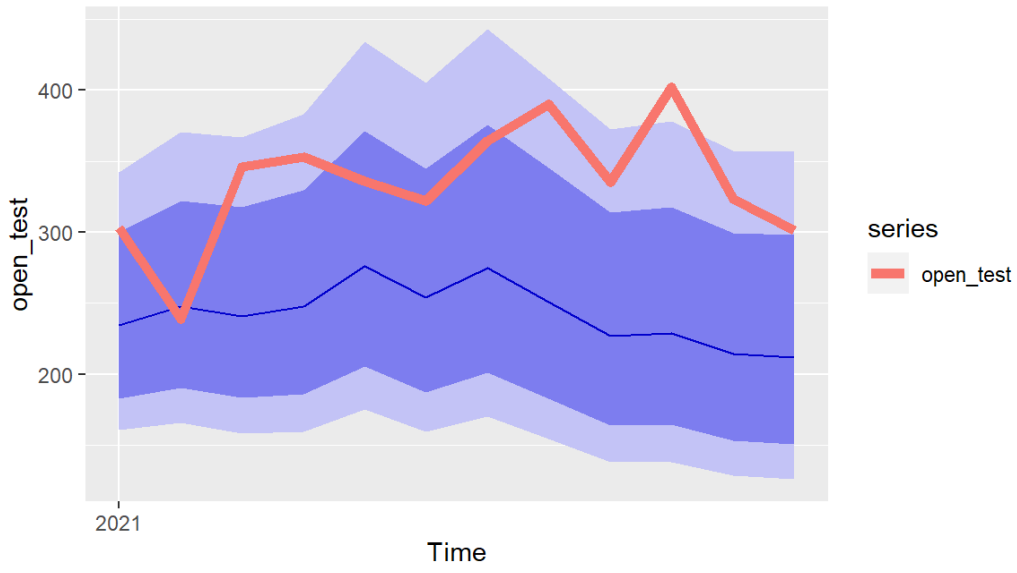
The RMSE we calculated by ETS is 53 95512.The fit of this model is better than STL.

D. ARIMA

Upon examining the time series plot of job openings, it is evident that higher values exhibit a greater degree of variability as compared to lower values. So we use BOX-COX transformation to test how to transform data for stationarity and normality. The λ we have strongly suggests a logarithmic transformation (λ = 0) for these data.

And to fit an ARIMA model, the data should be stationary, meaning that the statistical properties of the series remain constant over time. We achieve stationarity by removing trends and seasonality through differencing. After differencing, we observe that the data is stable and does not show any visible trends or seasonal patterns.

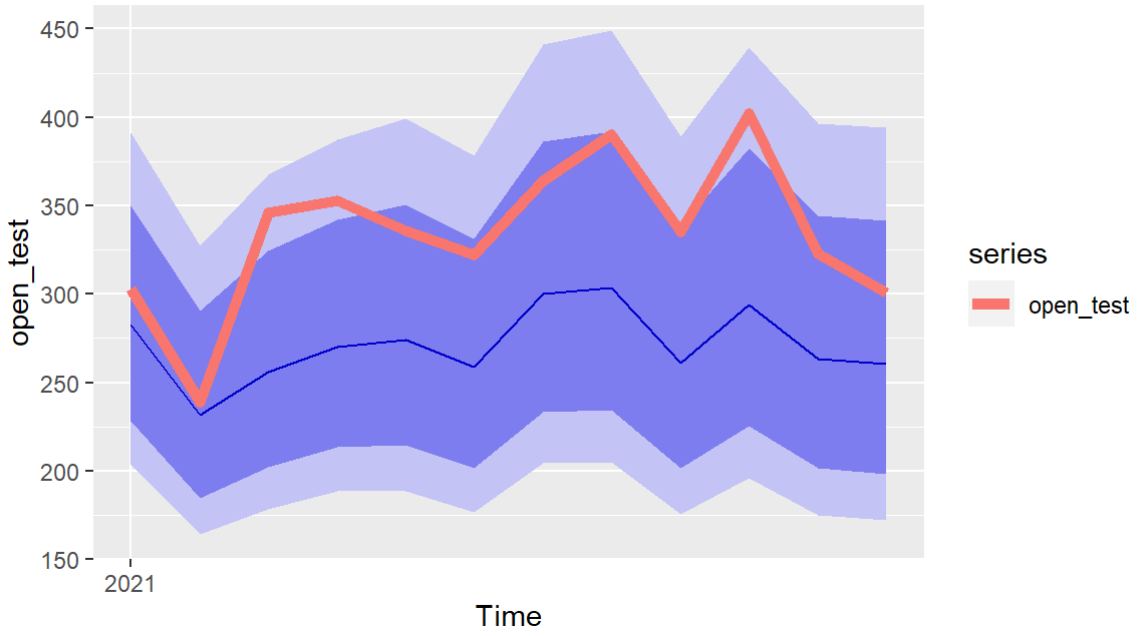
To determine the optimal ARIMA parameters, we use autocorrelation and partial autocorrelation plots. The autocorrelation plot reveals a slow decay, while the partial autocorrelation plot shows spikes at the first three lags and one seasonal lag. After overfitting, we settle on an ARIMA(3,1,3)(1,0,0) model.



The MRSE we calculated using ARIMA is 101.62319. It's the highest one of all the models.

E. ARIMAX(Dynamic Regression)

In the ARIMAX model, we introduce one variable to enhance the predictive power of the model based on the ARIMA model. We pick employees because from the plot before, we consider it shares some same trends and patterns with the opening.



The MRSE we calculated using ARIMAX is 69.01051. The fit of this model is better than the fit of ARIMA.

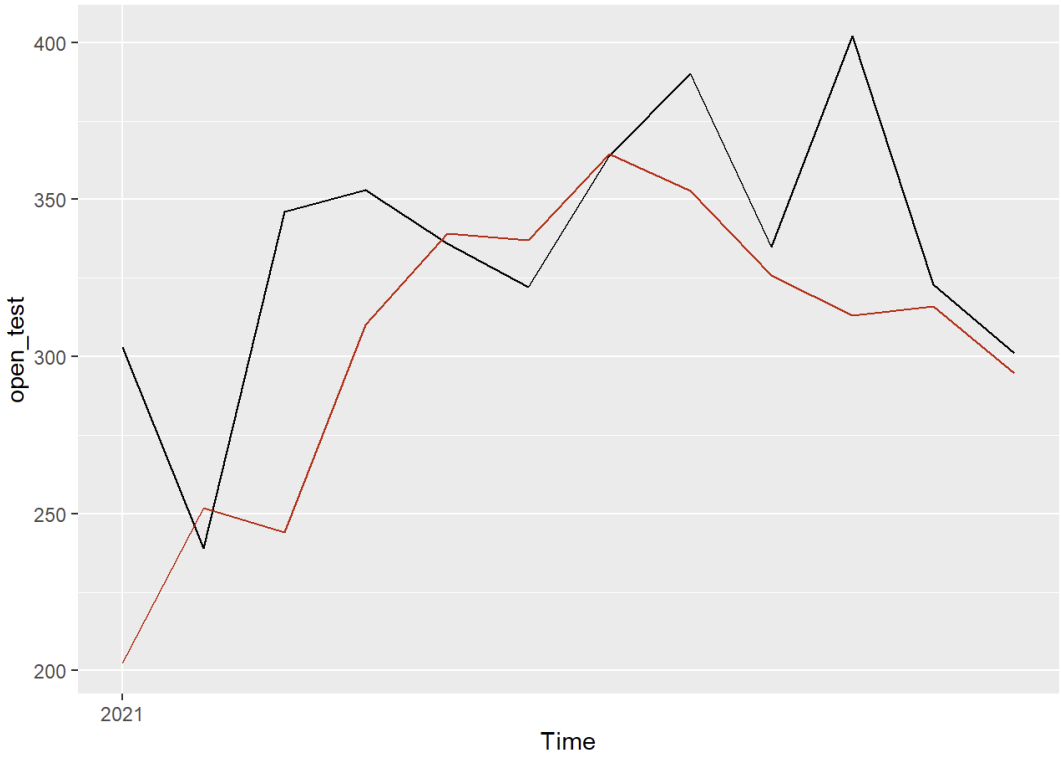
F. VAR(Vector Autoregression)

This model assumes the predictors influence the outcome. It is the situation, in this case, we have multiple variables and they may influence each other. For example, the layoff may relate to employees, or the opening may influence spending.

The performance of our VAR model was not satisfactory, as it resulted in an RMSE of 80.2, which was higher compared to the other models we tested.

G. NNET(Neural Net Autoregression)

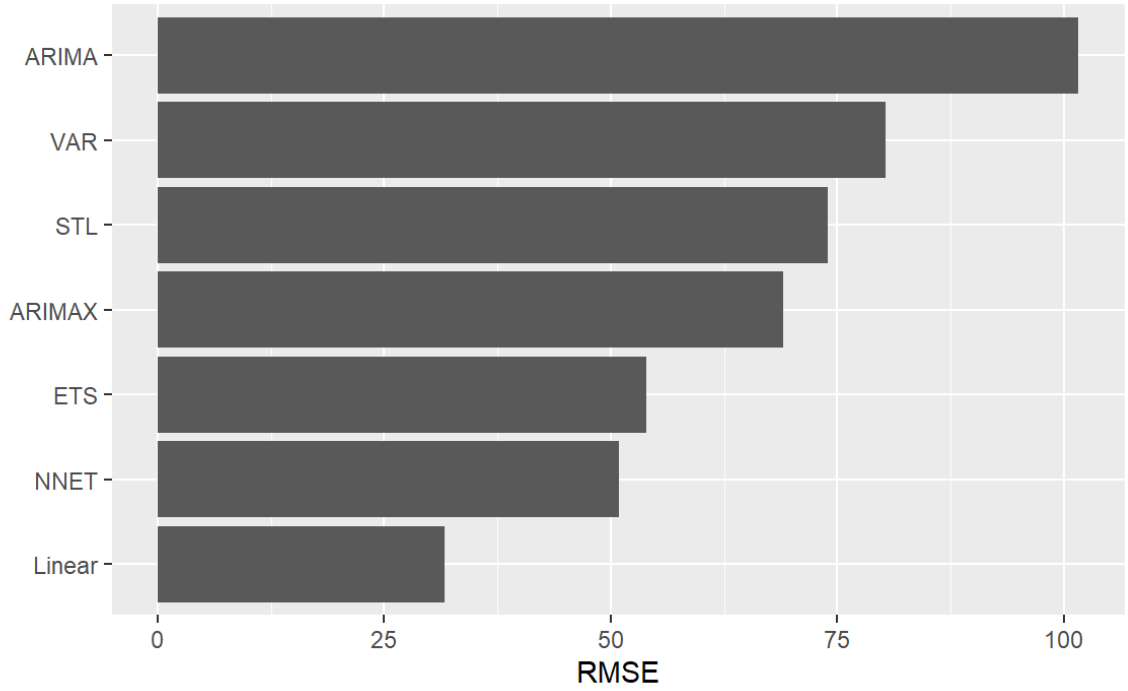
We use the Neural Net Autoregression here to fit the log of open_train. We set the parameters p and P to 10, which means that the model uses the previous 10 observations of the time series and one lag of the first seasonal period as inputs. The size parameter is set to 3, which is the number of hidden neurons in the neural network. We also change decay to 0.05 to penalize a small amount during training.



The performance of the NNET model is not good. The RMSE is 54.05138. But the fit of this model is in the top three of all models.

Model Comparison

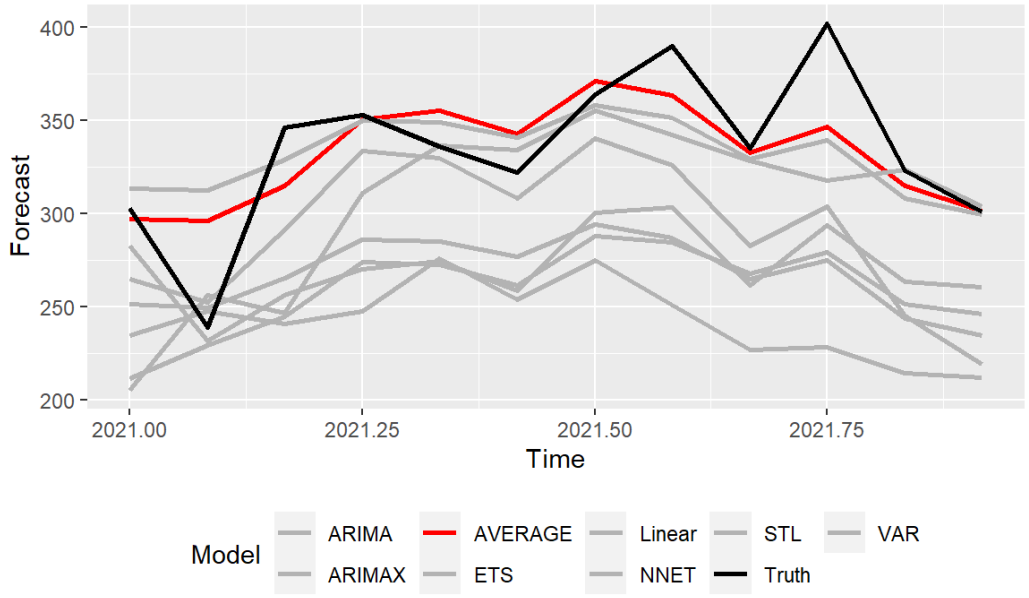
After assessing the performance of various models, we plotted the root mean square error of each model. The results showed that the linear model outperformed the other models, with the lowest RMSE. The neural network model and exponential smoothing method performed moderately well, ranking second and third, respectively. However, it is worth noting that the ARIMA model surprisingly performed the worst among all the models.



Final Model - Combining the Models

After evaluating the performance of different models and their underlying patterns, we selected five models - linear regression, ETS, ARIMAX, NNET, and VAR - to combine. We excluded ARIMA since it didn't fit well in the test data, and STL since it shows a similar pattern as ETS. We then weighted the models considering their respective RMSE values, resulting in assigning a

weight of 2 to linear regression and 0.5 to VAR, which produced the best result.



The combined model outperformed the individual models, resulting in the best performance overall.