Call Center Daily Forecast - HarvardX Capstone Project

Yuko Hayakawa

2024-05-03

Introduction

This project is a daily order forecasting project for a company's call center. In order to determine the staffing of the call center, a daily forecast of telephone orders received is required. The order data is recorded in US time, so must be converted to Japan time, taking into account business days and Japanese holidays.

Method and Analysis

The orders received by phone is extracted from all orders received in the past three years, and the future is predicted based on various influencing factors. Since different customer types are affected differently, we will forecast the number of orders by customer types before the overall order forecast is made. Also, unpredictable system bugs and natural disasters can cause the number of orders to fluctuate widely back and forth, so we will like to proceed with a comprehensive look at MAE, MAPE, MASE, SMAPE, RMSE and RSQ as measure of accuracy.

Data Preparation

Install and Require Packages

```
# Load Libraries
if(!require(webshot2)) install.packages("webshot2", repos = "http://cran.us.r-project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(gridExtra)) install.packages("gridExtra", repos = "http://cran.us.r-project.org")
if(!require(timetk)) install.packages("timetk", repos = "http://cran.us.r-project.org")
if(!require(reshape2)) install.packages("reshape2", repos = "https://cran.us.r-project.org")
if(!require(Boruta)) install.packages("Boruta", repos = "https://cran.us.r-project.org")
if(!require(imputeTS)) install.packages("imputeTS", repos = "https://cran.us.r-project.org")
if(!require(modeltime)) install.packages("modeltime", repos = "https://cran.us.r-project.org")
if(!require(parsnip)) install.packages("parsnip", repos = "https://cran.us.r-project.org")
if(!require(recipes)) install.packages("recipes", repos = "https://cran.us.r-project.org")
if(!require(workflows)) install.packages("workflows", repos = "https://cran.us.r-project.org")
if(!require(readxl)) install.packages("readxl", repos = "https://cran.us.r-project.org")
if(!require(rsample)) install.packages("rsample", repos = "https://cran.us.r-project.org")
if(!require(glmnet)) install.packages("glmnet", repos = "https://cran.us.r-project.org")
```

Since purchase trends differ by membership type, forecasts shall be made for each type and finally synthesized to produce an overall purchase forecast.

Type-D Forecast

First of all, we forecast a Type-D projection

Load Data and Data Preparation

Load Date Data

This Date data includes dates from 1/1/2021 ~ 12/31/2024, as well as Japanese days of the week and national holidays.(The holiday is marked 7)

```
Date <- read.csv("data/Date.csv") %>%
  mutate(Date = ymd(Date))
head(Date)
```

```
##         Date dayofweek
## 1 2021-01-01         7
## 2 2021-01-02         7
## 3 2021-01-03         7
## 4 2021-01-04         7
## 5 2021-01-05         2
## 6 2021-01-06         3
```

Load Order Data and Explor

```
# Load Customer Order Data
order <- read.csv("data/Orders.csv") %>%
  mutate(Date = ymd(Date))

head(order)
```

```
##         Date     time order_number order_type init      id customer_type pv
## 1 2021-02-01 17:00:54     88776961   Internet  *WB 6256345             D  0
## 2 2021-02-02 17:12:37     88817751   Internet  *WB 9380762             D  0
## 3 2021-02-02 17:40:06     88818264   Internet  *WB 5338244             D  0
## 4 2021-02-02 17:34:38     88818138   Internet  *WB   84947             D  0
## 5 2021-02-02 17:57:30     88818574   Internet  *WB    2762             D  0
## 6 2021-02-03 00:04:22     88826785   Internet  *WB 3732015             D  0
##   sales
## 1  2000
## 2  2000
## 3  2000
## 4  2000
```

```
## 5  2000
## 6  2000
```

See Customer Type

```
unique(order$customer_type)
```

```
## [1] "D" "W"
```

See Order Type

```
unique(order$order_type)
```

```
##  [1] "Internet"        "Autoship"        "Manual Autoship" "standard"
##  [5] "App Order"       "RMA"             "Online"          "Convention"
##  [9] "Professional"    "Promotion"
```

See Order Initial

```
unique(order$init)
```

```
##   [1] "*WB" "*AS" "CK9" "JTZ" "J$K" "THM" "TYO" "NOG" "SOA" "S4M" "NTO" "SON"
##  [13] "JTY" "NZS" "IIS" "YTO" "JYR" "NK"  "JM$" "MHR" "S3M" "AKI" "MK7" "M1A"
##  [25] "MO7" "GKI" "MO8" "HMD" "HIO" "AKB" "MA8" "JKO" "OTJ" "M5S" "ROY" "SAO"
##  [37] "CEO" "K2S" "ES7" "YKT" "TK6" "YKD" "M2T" "S5H" "NKO" "TK5" "AKH" "TKY"
##  [49] "CYO" "MFO" "AK9" "YH7" "KKA" "KSA" "M8C" "AOB" "MHT" "J5P" "AYG" "K6M"
##  [61] "OYS" "RUO" "JT!" "EHI" "JAO" "JMZ" "YYO" "M3T" "3IA" "TBB" "6MH" "RTS"
##  [73] "JYS" "YUU" "OCK" "YKA" "ED"  "KZW" "RK!" "EAO" "E1S" "JTK" "COA" "RK3"
##  [85] "ET5" "KOM" "JRC" "KY1" "H2M" "INO" "WOS" "JYK" "UTO" "SAK" "RP3" "TMM"
##  [97] "SM"  "YUF" "QUB" "LUB" "CUW" "DYA" "KOY" "SKO" "MOH" "JTH" "ERI" "IWA"
## [109] "KR"  "M4K" "AOJ" "ORI" "AY2" "MTO" "SOT" "FUJ" "MTA" "AKA" "JTE" "NFO"
## [121] "PJO" "SHI" "G3N" "AY3" "E1K" "MSK" "RI4" "Z89" ""    "TAT" "M4W" "K1Z"
## [133] "YBO" "SSQ" "SKG" "H3R" "YUK" "S1Z" "MXX" "CDK" "YUB" "WIA" "OMO" "ADY"
## [145] "AP!" "OLN" "SLC" "K3S" "NAO" "WSA" "TDO" "AI8" "SKT" "FBT" "MWY" "MNT"
## [157] "JNU" "HRF" "VIP" "JDH" "YKM" "MSY" "AII" "KNK" "M9D" "OTR" "7AE" "KI9"
## [169] "KYO" "KKO" "TKS" "MWR" "MEY" "OTA" "MOE" "K3N" "RGS" "CI5" "TO3" "KHD"
## [181] "LKB" "M8D" "GBM" "CKT" "KI1" "SPH" "CUM" "K3Y" "NA8" "SKI" "ITK" "DWN"
## [193] "KZ1" "RF$"
```

See PV and Sales by customer type

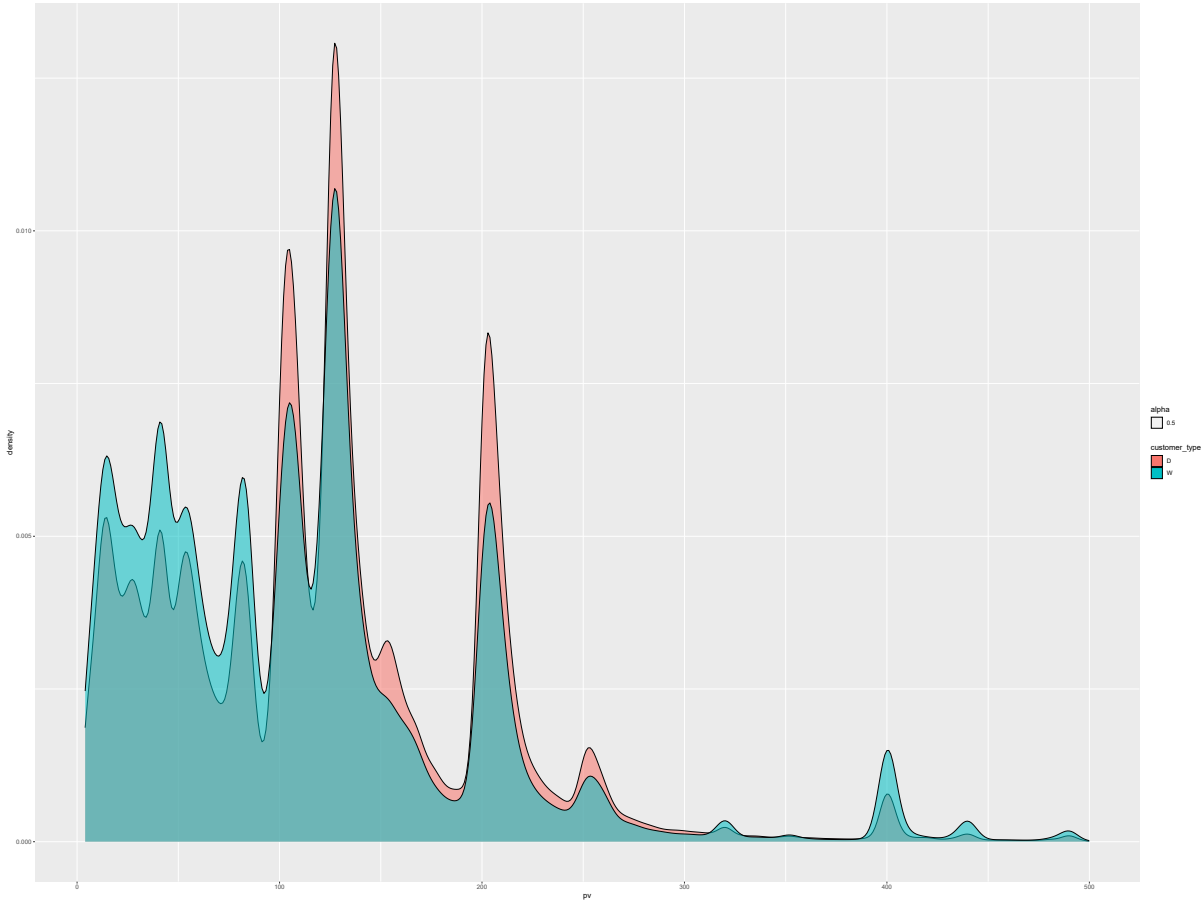
```
order %>%
  group_by(customer_type) %>%
```

```
  summarise(avg_pv = mean(pv),
            avg_sales = mean(sales),
            med_pv = median(pv),
            med_sales = median(sales),
            sd_pv = sd(pv),
            sd_sales = sd(sales))
```

```
## # A tibble: 2 x 7
##   customer_type avg_pv avg_sales med_pv med_sales sd_pv sd_sales
##   <chr>          <dbl>     <dbl>  <dbl>     <int> <dbl>    <dbl>
## 1 D               121.    17459.    114     16100  108.   15557.
## 2 W               114.    16197.    103     14400  115.   16172.
```

See Distribution of Purchase

```
order %>%
  filter(pv > 0, pv < 500) %>%
  ggplot(aes(pv, fill = customer_type, alpha = 0.5)) +
  geom_density()
```



It is clear that different membership types have different purchase price zones and different order types. We see

here that forecasts need to be made for each membership type.

See Number of Orders by Order and Customer Type

```
order %>%
  group_by(customer_type, order_type) %>%
  summarise(order = n()) %>%
  pivot_wider(names_from = customer_type,
              values_from = order )
```

```
## `summarise()` has grouped output by 'customer_type'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 10 x 3
##    order_type            D      W
##    <chr>             <int>  <int>
##  1 App Order         53808  59331
##  2 Autoship        1988774 587391
##  3 Convention           27      2
##  4 Internet         112341  52413
##  5 Manual Autoship  143438  37110
##  6 Online              407     13
##  7 Professional          2      3
##  8 Promotion             1      1
##  9 RMA               30687  11717
## 10 standard          61708  19488
```

Adjustment of Time Difference

```
# convert to Japanese time
order <- order %>%
  mutate(Date_us = as_datetime(paste(Date, time))) %>%
  mutate(Date_jpn = as.Date(Date_us + hours(15)))

order <- order %>%
  select(Date_jpn, order_type, init, customer_type)

names(order) <- c("Date", "order_type", "init", "customer_type")
```

Create Type = "D" Data

```
# Create Order Data of "D" and summarize
order_D <- order %>%
  filter(customer_type == "D") %>%
  group_by(Date) %>%
```

```r
  summarise(all = n())
order_D$all <- as.numeric(order_D$all)

# Create Telephone Order Data and summarize
order_D_tel <- order %>%
  filter(customer_type == "D",
         !(init %in% c("*AS", "*WB"))) %>%
  group_by(Date) %>%
  summarise(tel = n())
order_D_tel$tel <- as.numeric(order_D_tel$tel)

# Left join
order_D_tel <- left_join(order_D, order_D_tel, by = "Date")
order_D_tel[is.na(order_D_tel)] <- 0

head(order_D_tel)
```

```
## # A tibble: 6 x 3
##   Date         all   tel
##   <date>     <dbl> <dbl>
## 1 2021-01-01   240     0
## 2 2021-01-02   715     0
## 3 2021-01-03   853     0
## 4 2021-01-04  2284   250
## 5 2021-01-05  2233   245
## 6 2021-01-06  1791   321
```

Weighting the days of the week

Weight Telephone Orders only on business days, as they are heavily influenced by days of the week and holidays.

```r
order_D_tel <- Date %>%
  filter(Date <= "2023-12-31") %>%
  left_join(order_D_tel, by = "Date")

# Weighting
order_D_weight <- order_D_tel %>%
  group_by(dayofweek) %>%
  summarise(tel = sum(tel))

all <- sum(order_D_tel$all)

order_D_weight <- order_D_weight %>%
  mutate(percent = tel / all,
         weight = round(percent * 100, 3))

order_D_weight <- order_D_weight %>%
```

```
  select(dayofweek, weight)

order_D_weight
```

```
## # A tibble: 7 x 2
##   dayofweek weight
##       <int>  <dbl>
## 1         1  1.97
## 2         2  1.90
## 3         3  1.72
## 4         4  1.74
## 5         5  1.90
## 6         6  0.21
## 7         7  0.127
```

Load Promotion Data

Promotions and product launches also affect orders, so we will use this information as a predictor variable.

```
# Load Promotion Data and convert to integeres
Promotion_D <- read.csv("data/Promotion.csv") %>%
  mutate(Date = ymd(Date),
         Project_A = as.integer(Project_A),
         Project_B = as.integer(Project_B),
         Promotion_1 = as.integer(Promotion_1),
         Promotion_2 = as.integer(Promotion_2),
         Promotion_3 = as.integer(Promotion_3),
         Promotion_4 = as.integer(Promotion_4),
         Promotion_5 = as.integer(Promotion_5),
         Promotion_6 = as.integer(Promotion_6),
         New_Product = as.integer(New_Product),
         Reform_Product = as.integer(Reform_Product),
         LTO_Launch = as.integer(LTO_Launch),
         Event = as.integer(Event_Ticket)
         ) %>%
  select(-Event_Ticket)

head(Promotion_D)
```

```
##         Date dayofweek Project_A Project_B Promotion_1 Promotion_2 Promotion_3
## 1 2021-01-01         7         0         0           1           0           0
## 2 2021-01-02         7         0         0           1           0           0
## 3 2021-01-03         7         0         0           1           0           0
## 4 2021-01-04         7         0         0           1           0           0
## 5 2021-01-05         2         0         0           1           0           0
## 6 2021-01-06         3         0         0           1           0           0
##   Promotion_4 Promotion_5 Promotion_6 New_Product Reform_Product LTO_Launch
```

```
## 1           0           0           0           0           0           0
## 2           0           0           0           0           0           0
## 3           0           0           0           0           0           0
## 4           0           0           0           0           0           0
## 5           0           0           0           0           0           0
## 6           0           0           0           0           0           0
##   Event
## 1     0
## 2     0
## 3     0
## 4     0
## 5     0
## 6     0
```

Add Weight to Promotion Data

```
Promotion_D <- Promotion_D %>%
  left_join(order_D_weight, by = "dayofweek")
```

Left_join Promotion Data to Order Data

```
order_D_tel <- order_D_tel %>%
  select(-dayofweek) %>%
  left_join(Promotion_D, by = "Date")
```

Exploratory Analysis Type-D

See the Head of Data

```
head(order_D_tel)
```

```
##         Date  all tel dayofweek Project_A Project_B Promotion_1 Promotion_2
## 1 2021-01-01  240   0         7         0         0           1           0
## 2 2021-01-02  715   0         7         0         0           1           0
## 3 2021-01-03  853   0         7         0         0           1           0
## 4 2021-01-04 2284 250         7         0         0           1           0
## 5 2021-01-05 2233 245         2         0         0           1           0
## 6 2021-01-06 1791 321         3         0         0           1           0
##   Promotion_3 Promotion_4 Promotion_5 Promotion_6 New_Product Reform_Product
## 1           0           0           0           0           0              0
## 2           0           0           0           0           0              0
## 3           0           0           0           0           0              0
## 4           0           0           0           0           0              0
## 5           0           0           0           0           0              0
```

```
## 6           0           0           0           0           0              0
##   LTO_Launch Event weight
## 1          0     0  0.127
## 2          0     0  0.127
## 3          0     0  0.127
## 4          0     0  0.127
## 5          0     0  1.904
## 6          0     0  1.724
```

See Numbers of Order by 1 person

```
order_by_init <- order %>%
  filter(!(init %in% c("*AS", "*WB", ""))) %>%
  group_by(Date, init) %>%
  summarise(total_cases = n()) %>%
  ungroup() %>%
  group_by(init) %>%
  summarise(
    total_cases_per_day = mean(total_cases),
   total_cases_per_init = sum(total_cases)
  )
order_by_init %>%
  arrange(desc(total_cases_per_day))
```

```
## # A tibble: 191 x 3
##    init  total_cases_per_day total_cases_per_init
##    <chr>               <dbl>                <int>
##  1 JYK                  53.5                34857
##  2 K3Y                  43                     43
##  3 YUK                  25.0                 1199
##  4 RI4                  18.5                 1957
##  5 KI1                  17.5                   35
##  6 JTZ                  17.4                  522
##  7 KI9                  15.5                 1175
##  8 SSQ                  15.0                 3656
##  9 KYO                  14.8                  901
## 10 SKT                  14.6                 2106
## # i 181 more rows
```

```
mean(order_by_init$total_cases_per_day)
```

```
## [1] 7.046614
```

```
median(order_by_init$total_cases_per_day)
```

```
## [1] 5.862903
```

Average 7 cases, median 10 cases. Since the employee's company history is also a factor, we would like to use the top 10 employees as a guideline for forecasting accuracy. We think it would be reasonable if we could add 2-3 more employees to the shift, so we would like to have 15 x 3 = 45 employees –> Final RMSE would be less than 50.

Confirm the Correlation

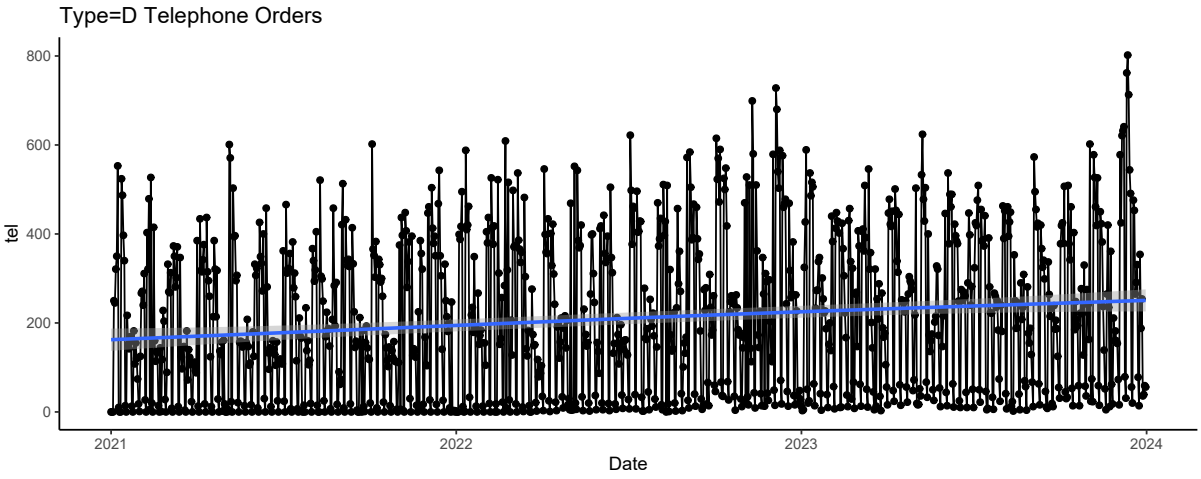
```
order_D_tel %>%
  select(-Date) %>%
  cor() %>%
  round(2)
```

```
##                   all   tel dayofweek Project_A Project_B Promotion_1
## all              1.00  0.56     -0.19      0.09     -0.02        0.05
## tel              0.56  1.00     -0.65      0.06     -0.03        0.05
## dayofweek       -0.19 -0.65      1.00      0.02      0.00        0.05
## Project_A        0.09  0.06      0.02      1.00     -0.32        0.17
## Project_B       -0.02 -0.03      0.00     -0.32      1.00       -0.16
## Promotion_1      0.05  0.05      0.05      0.17     -0.16        1.00
## Promotion_2      0.19  0.08     -0.02     -0.15     -0.08       -0.15
## Promotion_3      0.04  0.06     -0.05     -0.13      0.08       -0.13
## Promotion_4      0.08  0.04     -0.01     -0.01     -0.20       -0.09
## Promotion_5     -0.08 -0.04     -0.02     -0.05     -0.17       -0.24
## Promotion_6      0.09  0.04      0.02      0.09      0.55       -0.30
## New_Product      0.06 -0.01     -0.02     -0.02      0.01       -0.06
## Reform_Product   0.03 -0.01     -0.07     -0.06      0.06       -0.04
## LTO_Launch       0.03  0.04     -0.10      0.03      0.02        0.03
## Event            0.08  0.13     -0.10      0.01     -0.04       -0.04
## weight           0.20  0.76     -0.84     -0.02     -0.01       -0.03
##                 Promotion_2 Promotion_3 Promotion_4 Promotion_5 Promotion_6
## all                    0.19        0.04        0.08       -0.08        0.09
## tel                    0.08        0.06        0.04       -0.04        0.04
## dayofweek             -0.02       -0.05       -0.01       -0.02        0.02
## Project_A             -0.15       -0.13       -0.01       -0.05        0.09
## Project_B             -0.08        0.08       -0.20       -0.17        0.55
## Promotion_1           -0.15       -0.13       -0.09       -0.24       -0.30
## Promotion_2            1.00        0.12        0.09       -0.10       -0.10
## Promotion_3            0.12        1.00       -0.07       -0.01       -0.02
## Promotion_4            0.09       -0.07        1.00       -0.12       -0.23
## Promotion_5           -0.10       -0.01       -0.12        1.00       -0.24
## Promotion_6           -0.10       -0.02       -0.23       -0.24        1.00
## New_Product            0.04        0.05       -0.01        0.07        0.00
## Reform_Product         0.02        0.03       -0.04        0.07        0.02
## LTO_Launch            -0.01       -0.03       -0.08        0.08       -0.01
## Event                  0.11        0.05        0.02        0.04       -0.01
## weight                 0.02        0.05        0.01        0.01       -0.02
##                 New_Product Reform_Product LTO_Launch Event weight
## all                    0.06           0.03       0.03  0.08   0.20
```

```
## tel                -0.01         -0.01       0.04  0.13   0.76
## dayofweek          -0.02         -0.07      -0.10 -0.10  -0.84
## Project_A          -0.02         -0.06       0.03  0.01  -0.02
## Project_B           0.01          0.06       0.02 -0.04  -0.01
## Promotion_1        -0.06         -0.04       0.03 -0.04  -0.03
## Promotion_2         0.04          0.02      -0.01  0.11   0.02
## Promotion_3         0.05          0.03      -0.03  0.05   0.05
## Promotion_4        -0.01         -0.04      -0.08  0.02   0.01
## Promotion_5         0.07          0.07       0.08  0.04   0.01
## Promotion_6         0.00          0.02      -0.01 -0.01  -0.02
## New_Product         1.00          0.68       0.22 -0.02   0.04
## Reform_Product      0.68          1.00       0.28 -0.02   0.08
## LTO_Launch          0.22          0.28       1.00 -0.02   0.10
## Event              -0.02         -0.02      -0.02  1.00   0.10
## weight              0.04          0.08       0.10  0.10   1.00
```

Create Graph

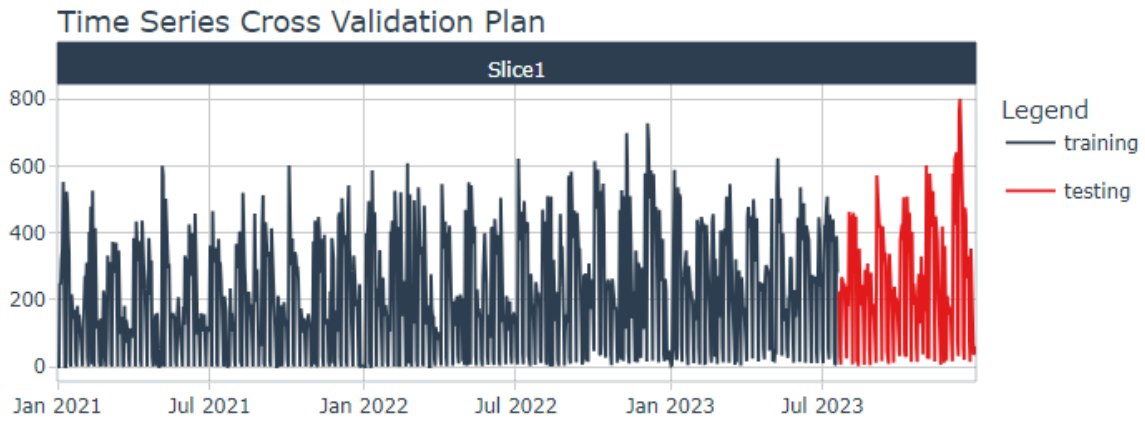
```
order_D_tel %>%
  ggplot(aes(Date, tel)) +
  geom_line() +
  geom_point() +
  geom_smooth() +
  labs(title = "Type=D Telephone Orders") +
  theme_classic()
```



Data Preparation for Forecast

Create test and training data while preserving the temporal order of the data. Create and plot time-series re-sampling specifications (rset) for cross-validation purposes.

```
Splits_D <- initial_time_split(order_D_tel, prop = 0.85)

Splits_D %>%
  tk_time_series_cv_plan() %>%
  plot_time_series_cv_plan(Date, tel,.interactive = TRUE)
```



Model Pre-Processing

Create a recipe specification for the training data using the recipe function. The target variable is the telephone order (tel), and the predictor variables used in the recipe are the explanatory variables that have an impact on telephone orders (tel) from the correlation coefficient.

The following 7 patterns of Models are created to find the most accurate model.

Machine Learning (ML) Models

- Elastic
- Random Forest
- Gradient Boost
- Prophet Boost Hybrid Model
- Arima Boost

NON ML Models

- Exponential Smoothing
- Prophet

Create Recipe to make a Linear Model

- **receipe()** : Create a recipe
- **step_timeseries_signature(Date)** : Adds time series signature variables (e.g., year, month, day, week, quarter) based on the Date column.
- **step_rm(matches("(hour)|(minute)|(second)|(am.pm)|(xts)|(iso)|(lbl)"))**

: Removes variables that match the specified regular expressions. (unnecessary time-related variables)

- **step_zv(all_predictors())** : (Enable this function as needed) Remove zero-variance predictors (variables with constant values) from the recipe.
- **step_dummy(all_nominal_predictors(), one_hot = TRUE)** : Creates dummy variables (one-hot encoding) for all nominal categorical predictors in the recipe.

```
recipe_D <- recipe(tel ~ Date + dayofweek + Project_A + Promotion_1 +
                     Promotion_2 + Promotion_3 + Promotion_4 + Promotion_5 +
                     Promotion_6 + LTO_Launch + Event + weight,
                   training(Splits_D)) %>%
  step_timeseries_signature(Date) %>%
  step_rm(matches("(hour)|(minute)|(second)|(am.pm)|(xts)|(iso)|(lbl)")) %>%
  # step_zv(all_predictors()) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE)
```

Workflow to match those 7 Models

ML Models

```
# Elastic Net
fit_glmnet_D <- linear_reg(penalty = 0.1, mixture = 0.5) %>%
  set_engine("glmnet")

# Create workflow to access metadata about the currently running flow in Power Automate.
# It is enabling you to build more robust and informative flows.
fit_glmnet_D <- workflow() %>%
  add_model(fit_glmnet_D) %>%
  add_recipe(recipe_D %>% step_rm(Date)) %>%
  fit(training(Splits_D))

# Random Forest
fit_rf_D<- workflow() %>%
  add_model(rand_forest("regression") %>%
              set_engine("ranger")) %>%
  add_recipe(recipe_D %>% step_rm(Date)) %>%
  fit(training(Splits_D))

# Gradient Boost
fit_xgb_D <- workflow() %>%
  add_model(boost_tree("regression") %>% set_engine("xgboost")) %>%
  add_recipe(recipe_D %>% step_rm(Date)) %>%
  fit(training(Splits_D))

# Prophet Boost Hybrid Model
fit_prophet_boost_D <- workflow() %>%
  add_model(prophet_boost("regression", seasonality_yearly = TRUE) %>% set_engine("prophet_xgboost")) %>%
```

```
  add_recipe(recipe_D) %>%
  fit(training(Splits_D))

# Arima Boost
fit_arima_boosted_D <- workflow() %>%
  add_model(arima_boost("regression") %>% set_engine("auto_arima_xgboost")) %>%
    add_recipe(recipe_D) %>%
  fit(training(Splits_D))
```

Non ML Models

```
# Exponential Smoothing
fit_ets_D <- workflow() %>%
  add_model(exp_smoothing() %>% set_engine("ets")) %>%
  add_recipe(recipe_D) %>%
  fit(training(Splits_D))
# Prophet
fit_prophet_D <- workflow() %>%
  add_model(prophet_reg() %>% set_engine("prophet")) %>%
  add_recipe(recipe_D) %>%
  fit(training(Splits_D))
```

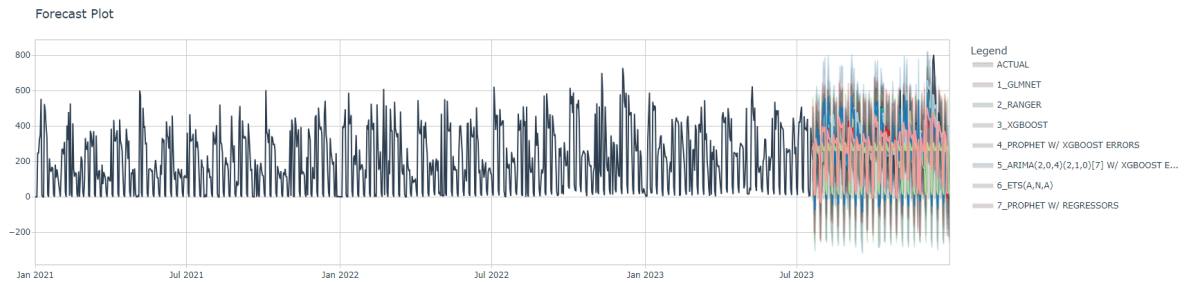
Add fitted model to a Model Table

```
models_tbl_D <-
  modeltime_table(
    fit_glmnet_D,
    fit_rf_D,
    fit_xgb_D,
    fit_prophet_boost_D,
    fit_arima_boosted_D,
    fit_ets_D,
    fit_prophet_D
  )

rm(
    fit_glmnet_D,
    fit_rf_D,
    fit_xgb_D,
    fit_prophet_boost_D,
    fit_arima_boosted_D,
    fit_ets_D,
    fit_prophet_D,
    recipe_D
)

### Calibration
```

```
calibration_table_D <- models_tbl_D %>%
  modeltime_calibrate(new_data = testing(Splits_D))
```

Plot those Models to Test Data

```
calibration_table_D %>%
  modeltime_forecast(
    new_data = testing(Splits_D),
    actual_data = order_D_tel
    ) %>%
  plot_modeltime_forecast(.interactive = TRUE)
```



Compare Accuracy

```
calibration_table_D %>%
  modeltime_accuracy() %>%
  table_modeltime_accuracy(
    .interactive = FALSE
  )
```

Accuracy Table

.model_id	.model_desc	.type	mae	mape	mase	smape	rmse	rsq
1	GLMNET	Test	82.99	129.80	0.59	56.82	108.79	0.68
2	RANGER	Test	45.18	37.36	0.32	26.15	68.49	0.87
3	XGBOOST	Test	47.23	36.57	0.34	27.16	72.65	0.85
4	PROPHET W/ XGBOOST ERRORS	Test	61.05	99.01	0.44	43.22	82.68	0.85
5	ARIMA(2,0,4)(2,1,0)[7] W/ XGBOOST ERRORS	Test	86.34	129.91	0.62	69.34	118.56	0.65
6	ETS(A,N,A)	Test	97.68	110.50	0.70	46.30	140.62	0.47
7	PROPHET W/ REGRESSORS	Test	82.94	118.15	0.59	52.00	106.98	0.69

Create a Recipe Specification for the Promotion Data

```
dates <- unique(order_D_tel$Date)
`%ni%` <- Negate(`%in%`)

Promotion_D <- Promotion_D %>%
  select(Date, Project_A, Promotion_1, Promotion_2, Promotion_3, Promotion_4,
         Promotion_5,Promotion_6, LTO_Launch, Event, weight, dayofweek) %>%
  filter(Date %ni% dates)

Promotion_D <- recipe(Project_A ~ Date + Promotion_1 + Promotion_2 +
                        Promotion_3 + Promotion_4 +Promotion_5 + Promotion_6 +
                        LTO_Launch + Event + weight + dayofweek, Promotion_D) %>%
  step_timeseries_signature(Date) %>%
  step_rm(matches("(hour)|(minute)|(second)|(am.pm)|(xts)|(iso)|(lbl)")) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  prep() %>%
  juice()

head(Promotion_D)
```

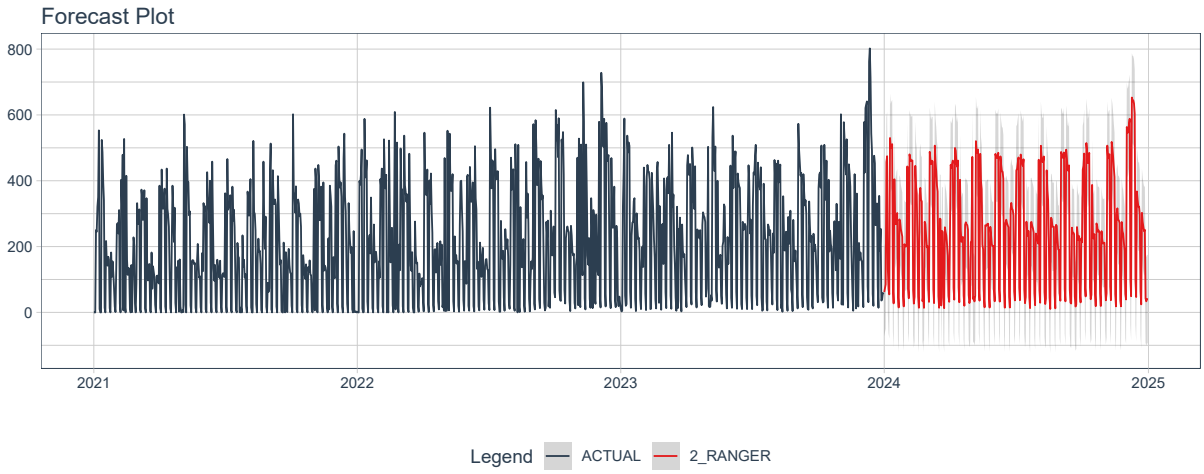
```
## # A tibble: 6 x 28
##   Date       Promotion_1 Promotion_2 Promotion_3 Promotion_4 Promotion_5
##   <date>           <int>       <int>       <int>       <int>       <int>
## 1 2024-01-01           1           0           0           0           0
## 2 2024-01-02           1           0           0           0           0
## 3 2024-01-03           1           0           0           0           0
## 4 2024-01-04           1           0           0           0           0
## 5 2024-01-05           1           0           0           0           0
## 6 2024-01-06           1           0           0           0           0
## # i 22 more variables: Promotion_6 <int>, LTO_Launch <int>, Event <int>,
## #   weight <dbl>, dayofweek <int>, Project_A <int>, Date_index.num <dbl>,
## #   Date_year <int>, Date_half <int>, Date_quarter <int>, Date_month <int>,
## #   Date_day <int>, Date_wday <int>, Date_mday <int>, Date_qday <int>,
## #   Date_yday <int>, Date_mweek <int>, Date_week <int>, Date_week2 <int>,
## #   Date_week3 <int>, Date_week4 <int>, Date_mday7 <int>
```

Choose the most Accuracy and Forecast

If we are unsure about some accuracy, select multiple choices to choose the better result from the output data.

```
Forecast_D <- calibration_table_D %>%
  filter(.model_id %in% c(2)) %>%
  modeltime_refit(order_D_tel) %>%
  modeltime_forecast(
    # h = "5 months",
```

```
    actual_data = order_D_tel,
    new_data = Promotion_D
  )

Export_D <- Forecast_D %>%
  filter(.model_desc != "ACTUAL") %>%
  select(-.model_id, -.key) %>%
  spread(key = .model_desc, value = .value) %>%
  mutate(.index = ymd(.index))

write_csv(Export_D, "forecast/type_D_forecast.csv")

Forecast_D %>%
  plot_modeltime_forecast(.interactive = FALSE)
```



Type-W Forecast

Next, we forecast a Type-W projection with same steps as Type-D.

Load Data and Data Preparation

Load Order Data and Create type ="W" Data

```
# Create Order Data of "W" and summarize
order_W <- order %>%
  filter(customer_type == "W") %>%
  group_by(Date) %>%
  summarise(all = n())
order_W$all <- as.numeric(order_W$all)


# Create Telephone Order Data and summarize
```

```r
order_W_tel <- order %>%
  filter(customer_type == "W",
         !(init %in% c("*AS", "*WB"))) %>%
  group_by(Date) %>%
  summarise(tel = n())
order_W_tel$tel <- as.numeric(order_W_tel$tel)

# Left join
order_W_tel <- left_join(order_W, order_W_tel, by = "Date")
order_W_tel[is.na(order_W_tel)] <- 0
```

Weighting the days of the week

```r
order_W_tel <- Date %>%
  filter(Date <= "2023-12-31") %>%
  left_join(order_W_tel, by = "Date")

# Holiday
order_W_weight <- order_W_tel %>%
  group_by(dayofweek) %>%
  summarise(tel = sum(tel))

all <- sum(order_W_tel$all)

order_W_weight <- order_W_weight %>%
  mutate(percent = tel / all,
         weight = round(percent * 100, 3))

order_W_weight <- order_W_weight %>%
  select(dayofweek, weight)

order_W_weight
```

```
## # A tibble: 7 x 2
##   dayofweek weight
##       <int>  <dbl>
## 1         1  1.83
## 2         2  1.74
## 3         3  1.59
## 4         4  1.55
## 5         5  1.71
## 6         6  0.096
## 7         7  0.045
```

Load Promotion Data

```
# Promotion Data
Promotion_W <- read.csv("data/Promotion.csv") %>%
  mutate(Date = ymd(Date),
         Project_A = as.integer(Project_A),
         Project_B = as.integer(Project_B),
         Promotion_1 = as.integer(Promotion_1),
         Promotion_2 = as.integer(Promotion_2),
         Promotion_3 = as.integer(Promotion_3),
         Promotion_4 = as.integer(Promotion_4),
         Promotion_5 = as.integer(Promotion_5),
         Promotion_6 = as.integer(Promotion_6),
         New_Product = as.integer(New_Product),
         Reform_Product = as.integer(Reform_Product),
         LTO_Launch = as.integer(LTO_Launch),
         Event = as.integer(Event_Ticket)
         ) %>%
  select(-Event_Ticket)
```

Add Weight to Promotion Data

```
Promotion_W <- Promotion_W %>%
  left_join(order_W_weight, by = "dayofweek")
```

Left_join Promotion data to Order Data

```
order_W_tel <- order_W_tel %>%
  select(-dayofweek) %>%
  left_join(Promotion_W, by = "Date")
```

Exploratory Analysis Type-W

See the Head of Data

```
head(order_W_tel)
```

```
##         Date all tel dayofweek Project_A Project_B Promotion_1 Promotion_2
## 1 2021-01-01  42   0         7         0         0           1           0
## 2 2021-01-02 154   0         7         0         0           1           0
## 3 2021-01-03 159   0         7         0         0           1           0
## 4 2021-01-04 591  64         7         0         0           1           0
```

```
## 5 2021-01-05 599  73         2         0         0           1           0
## 6 2021-01-06 374  88         3         0         0           1           0
##   Promotion_3 Promotion_4 Promotion_5 Promotion_6 New_Product Reform_Product
## 1           0           0           0           0           0              0
## 2           0           0           0           0           0              0
## 3           0           0           0           0           0              0
## 4           0           0           0           0           0              0
## 5           0           0           0           0           0              0
## 6           0           0           0           0           0              0
##   LTO_Launch Event weight
## 1          0     0  0.045
## 2          0     0  0.045
## 3          0     0  0.045
## 4          0     0  0.045
## 5          0     0  1.738
## 6          0     0  1.589
```

Confirm the Correlation

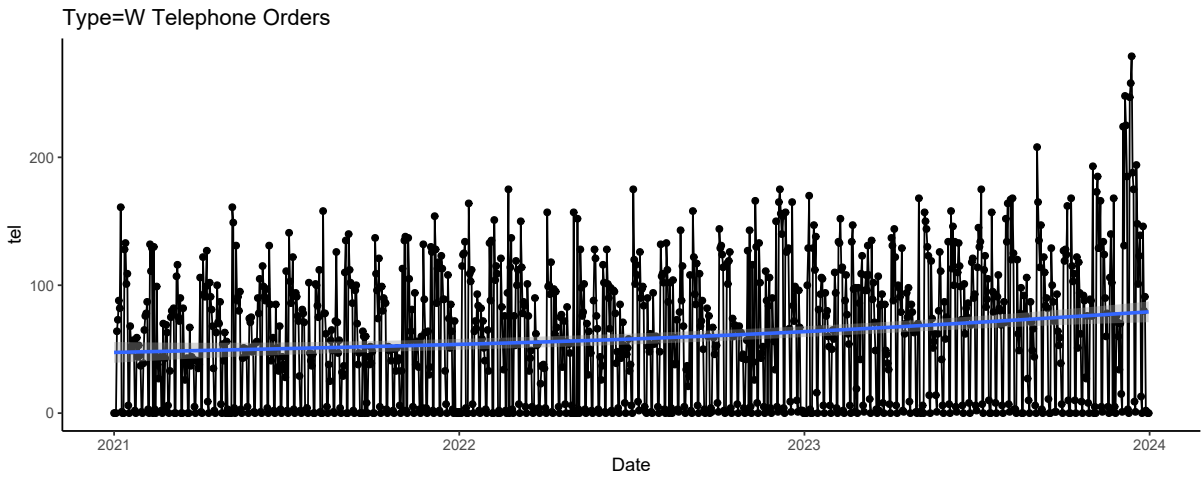
```
order_W_tel %>%
  select(-Date) %>%
  cor() %>%
  round(2)
```

```
##                  all   tel dayofweek Project_A Project_B Promotion_1
## all             1.00  0.49     -0.17      0.17     -0.04        0.00
## tel             0.49  1.00     -0.68      0.06     -0.04        0.04
## dayofweek      -0.17 -0.68      1.00      0.02      0.00        0.05
## Project_A       0.17  0.06      0.02      1.00     -0.32        0.17
## Project_B      -0.04 -0.04      0.00     -0.32      1.00       -0.16
## Promotion_1     0.00  0.04      0.05      0.17     -0.16        1.00
## Promotion_2     0.24  0.07     -0.02     -0.15     -0.08       -0.15
## Promotion_3     0.06  0.06     -0.05     -0.13      0.08       -0.13
## Promotion_4     0.11  0.04     -0.01     -0.01     -0.20       -0.09
## Promotion_5    -0.12 -0.03     -0.02     -0.05     -0.17       -0.24
## Promotion_6     0.10  0.00      0.02      0.09      0.55       -0.30
## New_Product     0.04  0.01     -0.02     -0.02      0.01       -0.06
## Reform_Product  0.00  0.01     -0.07     -0.06      0.06       -0.04
## LTO_Launch      0.01  0.05     -0.10      0.03      0.02        0.03
## Event           0.08  0.14     -0.10      0.01     -0.04       -0.04
## weight          0.18  0.79     -0.85     -0.02     -0.01       -0.03
##                Promotion_2 Promotion_3 Promotion_4 Promotion_5 Promotion_6
## all                   0.24        0.06        0.11       -0.12        0.10
## tel                   0.07        0.06        0.04       -0.03        0.00
## dayofweek            -0.02       -0.05       -0.01       -0.02        0.02
## Project_A            -0.15       -0.13       -0.01       -0.05        0.09
```

```
## Project_B            -0.08        0.08       -0.20       -0.17        0.55
## Promotion_1          -0.15       -0.13       -0.09       -0.24       -0.30
## Promotion_2           1.00        0.12        0.09       -0.10       -0.10
## Promotion_3           0.12        1.00       -0.07       -0.01       -0.02
## Promotion_4           0.09       -0.07        1.00       -0.12       -0.23
## Promotion_5          -0.10       -0.01       -0.12        1.00       -0.24
## Promotion_6          -0.10       -0.02       -0.23       -0.24        1.00
## New_Product           0.04        0.05       -0.01        0.07        0.00
## Reform_Product        0.02        0.03       -0.04        0.07        0.02
## LTO_Launch           -0.01       -0.03       -0.08        0.08       -0.01
## Event                 0.11        0.05        0.02        0.04       -0.01
## weight                0.02        0.05        0.01        0.01       -0.02
##                New_Product Reform_Product LTO_Launch Event weight
## all                   0.04           0.00       0.01  0.08   0.18
## tel                   0.01           0.01       0.05  0.14   0.79
## dayofweek            -0.02          -0.07      -0.10 -0.10  -0.85
## Project_A            -0.02          -0.06       0.03  0.01  -0.02
## Project_B             0.01           0.06       0.02 -0.04  -0.01
## Promotion_1          -0.06          -0.04       0.03 -0.04  -0.03
## Promotion_2           0.04           0.02      -0.01  0.11   0.02
## Promotion_3           0.05           0.03      -0.03  0.05   0.05
## Promotion_4          -0.01          -0.04      -0.08  0.02   0.01
## Promotion_5           0.07           0.07       0.08  0.04   0.01
## Promotion_6           0.00           0.02      -0.01 -0.01  -0.02
## New_Product           1.00           0.68       0.22 -0.02   0.04
## Reform_Product        0.68           1.00       0.28 -0.02   0.08
## LTO_Launch            0.22           0.28       1.00 -0.02   0.10
## Event                -0.02          -0.02      -0.02  1.00   0.10
## weight                0.04           0.08       0.10  0.10   1.00
```

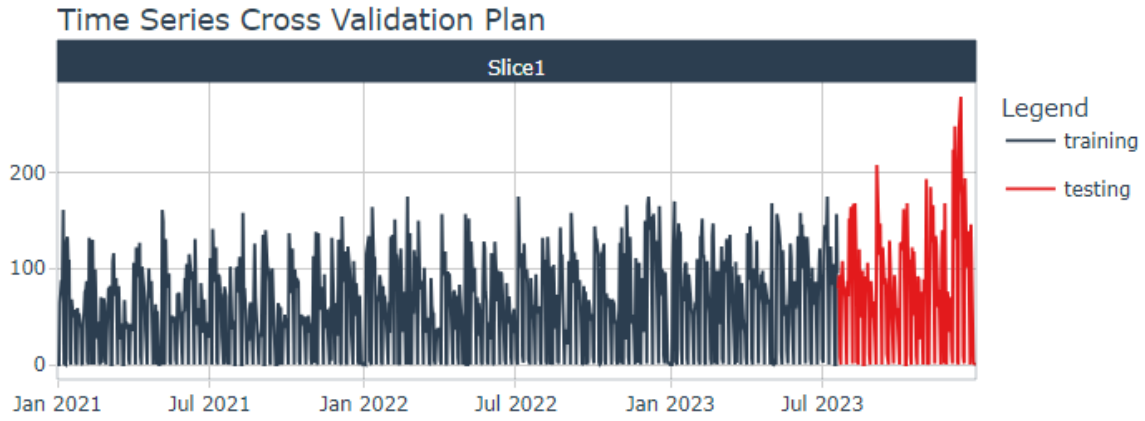
Create Fraph

```
order_W_tel%>%
  ggplot(aes(Date, tel)) +
  geom_line() +
  geom_point() +
  geom_smooth() +
  labs(title = "Type=W Telephone Orders") +
  theme_classic()
```



Data Preparation for Forecast

```
Splits_W <- initial_time_split(order_W_tel, prop = 0.85)

Splits_W %>%
  tk_time_series_cv_plan() %>%
  plot_time_series_cv_plan(Date, tel,.interactive = TRUE)
```



Model Pre-Processing

Create Recipe to make a Linear Model

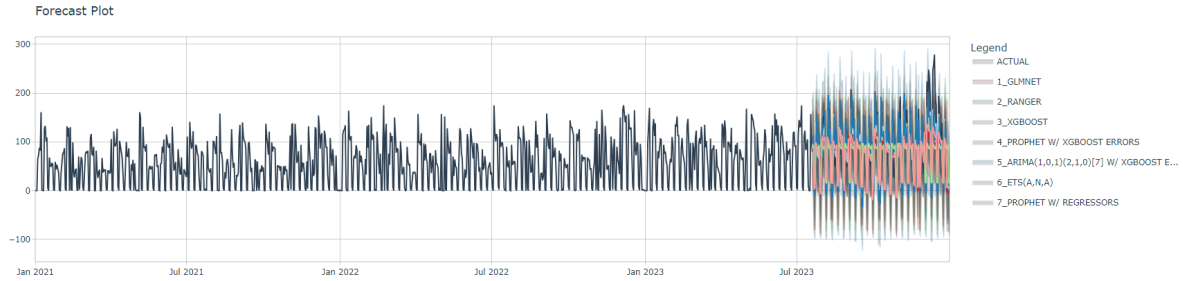
```
recipe_W <- recipe(tel ~ Date + dayofweek + Project_A + Project_B + Promotion_1 +
                     Promotion_2 + Promotion_3 + Promotion_4 + LTO_Launch +
                     Event + weight, training(Splits_W)) %>%
  step_timeseries_signature(Date) %>%
  step_rm(matches("(hour)|(minute)|(second)|(am.pm)|(xts)|(iso)|(lbl)")) %>%
```

```
  # step_zv(all_predictors()) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE)

############################
### ML Models
# Elastic Net
fit_glmnet_W <- linear_reg(penalty = 0.1, mixture = 0.5) %>%
  set_engine("glmnet")
fit_glmnet_W <- workflow() %>%
  add_model(fit_glmnet_W) %>%
  add_recipe(recipe_W %>% step_rm(Date)) %>%
  fit(training(Splits_W))

# Random Forest
fit_rf_W<- workflow() %>%
  add_model(rand_forest("regression") %>% set_engine("ranger")) %>%
  add_recipe(recipe_W %>% step_rm(Date)) %>%
  fit(training(Splits_W))

# Gradient Boost
fit_xgb_W <- workflow() %>%
  add_model(boost_tree("regression") %>% set_engine("xgboost")) %>%
  add_recipe(recipe_W %>% step_rm(Date)) %>%
  fit(training(Splits_W))

# Prophet Boost Hybrid Model
fit_prophet_boost_W <- workflow() %>%
  add_model(prophet_boost("regression", seasonality_yearly = TRUE) %>%
              set_engine("prophet_xgboost")) %>%
  add_recipe(recipe_W) %>%
  fit(training(Splits_W))

# Arima Boost
fit_arima_boosted_W <- workflow() %>%
  add_model(arima_boost("regression") %>% set_engine("auto_arima_xgboost")) %>%
  add_recipe(recipe_W) %>%
  fit(training(Splits_W))

############################
### Non ML Models
# Exponential Smoothing
fit_ets_W <- workflow() %>%
  add_model(exp_smoothing() %>% set_engine("ets")) %>%
  add_recipe(recipe_W) %>%
  fit(training(Splits_W))
# Prophet
fit_prophet_W <- workflow() %>%
```

```
  add_model(prophet_reg() %>% set_engine("prophet")) %>%
  add_recipe(recipe_W) %>%
  fit(training(Splits_W))

############################
### Add fitted model to a Model Table
models_tbl_W <-
  modeltime_table(
    fit_glmnet_W,
    fit_rf_W,
    fit_xgb_W,
    fit_prophet_boost_W,
    fit_arima_boosted_W,
    fit_ets_W,
    fit_prophet_W
  )
############################

rm(
    fit_glmnet_W,
    fit_rf_W,
    fit_xgb_W,
    fit_prophet_boost_W,
    fit_arima_boosted_W,
    fit_ets_W,
    fit_prophet_W,
    recipe_W
)

### Calibration
calibration_table_W <- models_tbl_W %>%
  modeltime_calibrate(new_data = testing(Splits_W))
```

Plot those Models to Test Data

```
calibration_table_W %>%
  modeltime_forecast(
    new_data = testing(Splits_W),
    actual_data = order_W_tel
    ) %>%
  plot_modeltime_forecast(.interactive = TRUE)
```



Compare Accuracy

```
calibration_table_W %>%
  modeltime_accuracy() %>%
  table_modeltime_accuracy(
    .interactive = FALSE
  )
```

Accuracy Table

.model_id	.model_desc	.type	mae	mape	mase	smape	rmse	rsq
1	GLMNET	Test	27.12	Inf	0.52	74.87	40.67	0.69
2	RANGER	Test	19.06	Inf	0.37	56.57	31.83	0.81
3	XGBOOST	Test	22.21	Inf	0.43	62.66	34.52	0.74
4	PROPHET W/ XGBOOST ERRORS	Test	21.73	Inf	0.42	65.13	31.36	0.80
5	ARIMA(1,0,1)(2,1,0)[7] W/ XGBOOST ERRORS	Test	31.96	Inf	0.62	82.83	45.22	0.58
6	ETS(A,N,A)	Test	34.86	Inf	0.67	75.40	49.86	0.49
7	PROPHET W/ REGRESSORS	Test	27.08	Inf	0.52	75.09	38.98	0.72

Create a Recipe Specification for the Promotion Data

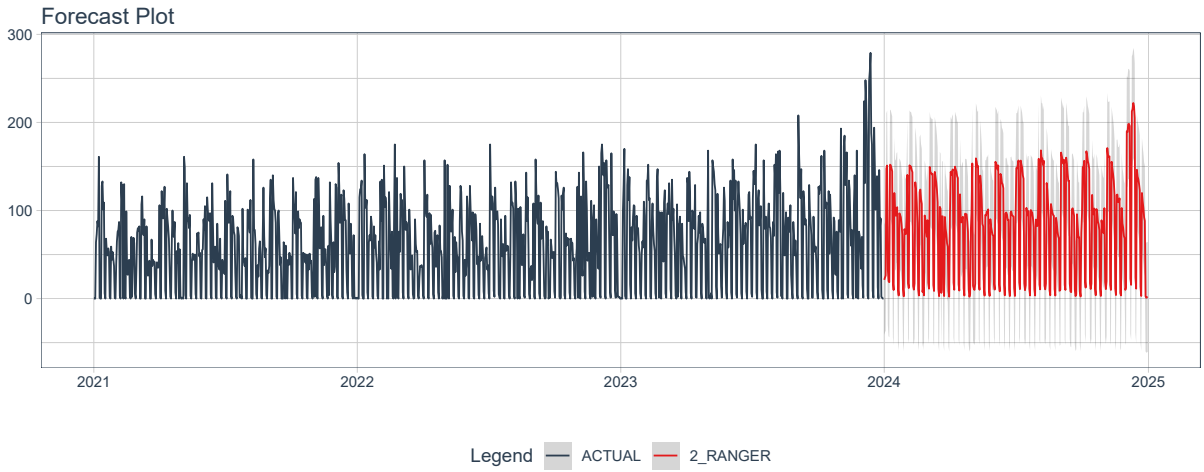
```
dates <- unique(order_W_tel$Date)
`%ni%` <- Negate(`%in%`)

Promotion_W <- Promotion_W %>%
  select(Date, Project_A, Project_B, Promotion_1, Promotion_2, Promotion_3,
         Promotion_4,
         LTO_Launch, Event, weight, dayofweek) %>%
  filter(Date %ni% dates)

Promotion_W <- recipe(Project_A ~ Date + Project_B + Promotion_1 + Promotion_2 +
                        Promotion_3 +
                        Promotion_4 +  LTO_Launch + Event + weight + dayofweek,
                      Promotion_W) %>%
  step_timeseries_signature(Date) %>%
```

```
  step_rm(matches("(hour)|(minute)|(second)|(am.pm)|(xts)|(iso)|(lbl)")) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  prep() %>%
  juice()

head(Promotion_W)
```

```
## # A tibble: 6 x 27
##   Date       Project_B Promotion_1 Promotion_2 Promotion_3 Promotion_4
##   <date>         <int>       <int>       <int>       <int>       <int>
## 1 2024-01-01         0           1           0           0           0
## 2 2024-01-02         0           1           0           0           0
## 3 2024-01-03         0           1           0           0           0
## 4 2024-01-04         0           1           0           0           0
## 5 2024-01-05         0           1           0           0           0
## 6 2024-01-06         0           1           0           0           0
## # i 21 more variables: LTO_Launch <int>, Event <int>, weight <dbl>,
## #   dayofweek <int>, Project_A <int>, Date_index.num <dbl>, Date_year <int>,
## #   Date_half <int>, Date_quarter <int>, Date_month <int>, Date_day <int>,
## #   Date_wday <int>, Date_mday <int>, Date_qday <int>, Date_yday <int>,
## #   Date_mweek <int>, Date_week <int>, Date_week2 <int>, Date_week3 <int>,
## #   Date_week4 <int>, Date_mday7 <int>
```

Choose the most Accuracy and Forecast

```
Forecast_W <- calibration_table_W %>%
  filter(.model_id %in% c(2)) %>%
  modeltime_refit(order_W_tel) %>%
  modeltime_forecast(
    # h = "5 months",
    actual_data = order_W_tel,
    new_data = Promotion_W
  )

Export_W <- Forecast_W %>%
  filter(.model_desc != "ACTUAL") %>%
  select(-.model_id, -.key) %>%
  spread(key = .model_desc, value = .value) %>%
  mutate(.index = ymd(.index))

write_csv(Export_W, "forecast/type_W_forecast.csv")

Forecast_W %>%
  plot_modeltime_forecast(.interactive = FALSE)
```



All Types Forecast

Finally, include Type-D and Type-W Forecasts in the predictor variables to Forecast the over all number of phone orders.

Load Data and Data Preparation

Load Forecast Data of Type-D and Type-W and Preparation

If multiple accuracies are selected in the table, review the exported data and work to keep the more accurate data. Import the processed data again and use it for the overall forecast.

```
# Create Type-W Data using Telephone Orders of Type-D forecast
Forecast_D <- read_csv("forecast/type_D_forecast.csv")

Forecast_D <- Forecast_D[,c(1,4)]

names(Forecast_D) <- c("Date", "tel")

order_D_tel <- order_D_tel %>%
  select(Date, tel)
order_D_tel <- rbind(order_D_tel, Forecast_D)

rm(Forecast_D)


# Create Type-W Data using Telephone Orders of Type-W forecast
Forecast_W <- read_csv("forecast/type_W_forecast.csv")

Forecast_W <- Forecast_W[,c(1,4)]

names(Forecast_W) <- c("Date", "tel")
```

```r
order_W_tel <- order_W_tel %>%
  select(Date, tel)
order_W <- rbind(order_W_tel, Forecast_W)

rm(Forecast_W)
```

Create Data for All Telephone Orders

```r
# All Telephone Orders
order_tel <- order %>%
  select(Date, init) %>%
  filter(!(init %in% c("*AS", "*WB")))

order_tel <- order_tel %>%
  group_by(Date) %>%
  summarise(tel = n())

order_tel$tel <- as.numeric(order_tel$tel)

order_tel <- Date %>%
  filter(Date <= "2023-12-31") %>%
  select(Date) %>%
  left_join(order_tel, by = "Date")
order_tel[is.na(order_tel)] <- 0


# All orders
order_all <- order %>%
  group_by(Date) %>%
  summarise(all = n())

order_tel <- order_all %>%
  left_join(order_tel, by = "Date")
```

Weighting the days of the week

```r
order_tel <- Date %>%
  filter(Date <= "2023-12-31") %>%
  left_join(order_tel, by = "Date")

# Holiday
order_weight <- order_tel %>%
  group_by(dayofweek) %>%
  summarise(tel = sum(tel))
```

```r
all <- sum(order_tel$all)

order_weight <- order_weight %>%
  mutate(percent = tel / all,
         weight = round(percent * 100, 3))

order_weight <- order_weight %>%
  select(dayofweek, weight)

order_weight
```

```
## # A tibble: 7 x 2
##   dayofweek weight
##       <int>  <dbl>
## 1         1  1.93
## 2         2  1.86
## 3         3  1.69
## 4         4  1.69
## 5         5  1.85
## 6         6  0.182
## 7         7  0.107
```

Load Promotion Data

```r
# Promotion Data
Promotion <- read.csv("data/Promotion.csv") %>%
  mutate(Date = ymd(Date),
         Project_A = as.integer(Project_A),
         Project_B = as.integer(Project_B),
         Promotion_1 = as.integer(Promotion_1),
         Promotion_2 = as.integer(Promotion_2),
         Promotion_3 = as.integer(Promotion_3),
         Promotion_4 = as.integer(Promotion_4),
         Promotion_5 = as.integer(Promotion_5),
         Promotion_6 = as.integer(Promotion_6),
         New_Product = as.integer(New_Product),
         Reform_Product = as.integer(Reform_Product),
         LTO_Launch = as.integer(LTO_Launch),
         Event = as.integer(Event_Ticket)
         ) %>%
  select(-Event_Ticket)
```

Add Weight to Promotion Data

```
Promotion <- Promotion %>%
  left_join(order_weight, by = "dayofweek")
```

Left_join Promotion Data to Order Data

```
order_tel <- order_tel %>%
  select(-dayofweek) %>%
  left_join(Promotion, by = "Date")
```

Exploratory Analysis All

See the Head of Data

```
head(order_tel)
```

```
##         Date  all tel dayofweek Project_A Project_B Promotion_1 Promotion_2
## 1 2021-01-01  282   0         7         0         0           1           0
## 2 2021-01-02  869   0         7         0         0           1           0
## 3 2021-01-03 1012   0         7         0         0           1           0
## 4 2021-01-04 2875 314         7         0         0           1           0
## 5 2021-01-05 2832 318         2         0         0           1           0
## 6 2021-01-06 2165 409         3         0         0           1           0
##   Promotion_3 Promotion_4 Promotion_5 Promotion_6 New_Product Reform_Product
## 1           0           0           0           0           0              0
## 2           0           0           0           0           0              0
## 3           0           0           0           0           0              0
## 4           0           0           0           0           0              0
## 5           0           0           0           0           0              0
## 6           0           0           0           0           0              0
##   LTO_Launch Event weight
## 1          0     0  0.107
## 2          0     0  0.107
## 3          0     0  0.107
## 4          0     0  0.107
## 5          0     0  1.864
## 6          0     0  1.691
```

Confirm the Correlation

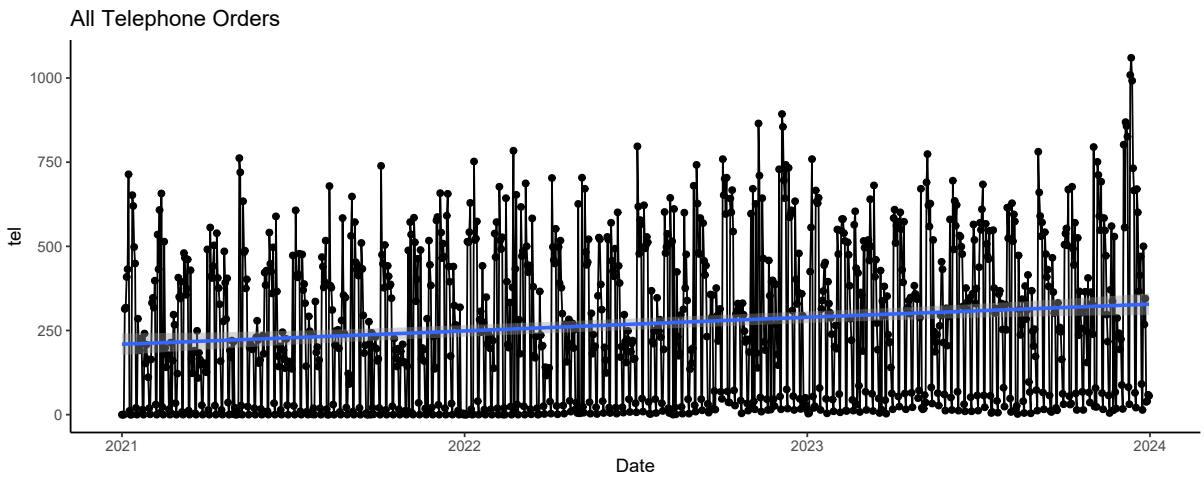
```r
order_tel %>%
  select(-Date) %>%
  cor() %>%
  round(2)
```

```
##                  all   tel dayofweek Project_A Project_B Promotion_1
## all             1.00  0.55     -0.19      0.11     -0.03        0.04
## tel             0.55  1.00     -0.66      0.06     -0.03        0.05
## dayofweek      -0.19 -0.66      1.00      0.02      0.00        0.05
## Project_A       0.11  0.06      0.02      1.00     -0.32        0.17
## Project_B      -0.03 -0.03      0.00     -0.32      1.00       -0.16
## Promotion_1     0.04  0.05      0.05      0.17     -0.16        1.00
## Promotion_2     0.20  0.08     -0.02     -0.15     -0.08       -0.15
## Promotion_3     0.04  0.06     -0.05     -0.13      0.08       -0.13
## Promotion_4     0.09  0.04     -0.01     -0.01     -0.20       -0.09
## Promotion_5    -0.09 -0.04     -0.02     -0.05     -0.17       -0.24
## Promotion_6     0.09  0.03      0.02      0.09      0.55       -0.30
## New_Product     0.05  0.00     -0.02     -0.02      0.01       -0.06
## Reform_Product  0.02  0.00     -0.07     -0.06      0.06       -0.04
## LTO_Launch      0.03  0.04     -0.10      0.03      0.02        0.03
## Event           0.08  0.13     -0.10      0.01     -0.04       -0.04
## weight          0.20  0.77     -0.85     -0.02     -0.01       -0.03
##                Promotion_2 Promotion_3 Promotion_4 Promotion_5 Promotion_6
## all                   0.20        0.04        0.09       -0.09        0.09
## tel                   0.08        0.06        0.04       -0.04        0.03
## dayofweek            -0.02       -0.05       -0.01       -0.02        0.02
## Project_A            -0.15       -0.13       -0.01       -0.05        0.09
## Project_B            -0.08        0.08       -0.20       -0.17        0.55
## Promotion_1          -0.15       -0.13       -0.09       -0.24       -0.30
## Promotion_2           1.00        0.12        0.09       -0.10       -0.10
## Promotion_3           0.12        1.00       -0.07       -0.01       -0.02
## Promotion_4           0.09       -0.07        1.00       -0.12       -0.23
## Promotion_5          -0.10       -0.01       -0.12        1.00       -0.24
## Promotion_6          -0.10       -0.02       -0.23       -0.24        1.00
## New_Product           0.04        0.05       -0.01        0.07        0.00
## Reform_Product        0.02        0.03       -0.04        0.07        0.02
## LTO_Launch           -0.01       -0.03       -0.08        0.08       -0.01
## Event                 0.11        0.05        0.02        0.04       -0.01
## weight                0.02        0.05        0.01        0.01       -0.02
##                New_Product Reform_Product LTO_Launch Event weight
## all                   0.05           0.02       0.03  0.08   0.20
## tel                   0.00           0.00       0.04  0.13   0.77
## dayofweek            -0.02          -0.07      -0.10 -0.10  -0.85
## Project_A            -0.02          -0.06       0.03  0.01  -0.02
## Project_B             0.01           0.06       0.02 -0.04  -0.01
## Promotion_1          -0.06          -0.04       0.03 -0.04  -0.03
## Promotion_2           0.04           0.02      -0.01  0.11   0.02
```

```
## Promotion_3          0.05          0.03      -0.03  0.05   0.05
## Promotion_4         -0.01         -0.04      -0.08  0.02   0.01
## Promotion_5          0.07          0.07       0.08  0.04   0.01
## Promotion_6          0.00          0.02      -0.01 -0.01  -0.02
## New_Product          1.00          0.68       0.22 -0.02   0.04
## Reform_Product       0.68          1.00       0.28 -0.02   0.08
## LTO_Launch           0.22          0.28       1.00 -0.02   0.10
## Event               -0.02         -0.02      -0.02  1.00   0.10
## weight               0.04          0.08       0.10  0.10   1.00
```

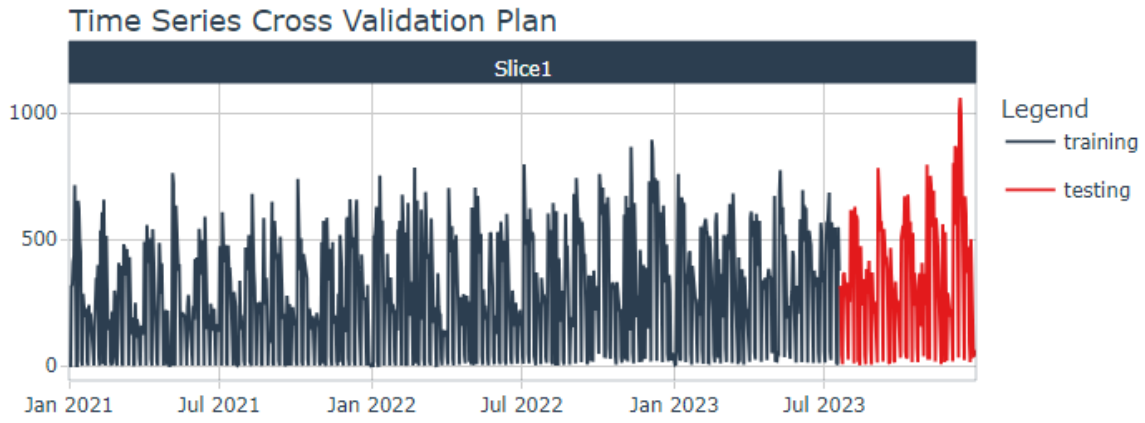
Create Fraph

```
order_tel %>%
  ggplot(aes(Date, tel)) +
  geom_line() +
  geom_point() +
  geom_smooth() +
  labs(title = "All Telephone Orders") +
  theme_classic()
```



Data Preparation for Forecast

```
Splits <- initial_time_split(order_tel, prop = 0.85)

Splits %>%
  tk_time_series_cv_plan() %>%
  plot_time_series_cv_plan(Date, tel,.interactive = TRUE)
```



Model Pre-Processing

Create Recipe to make a Linear Model

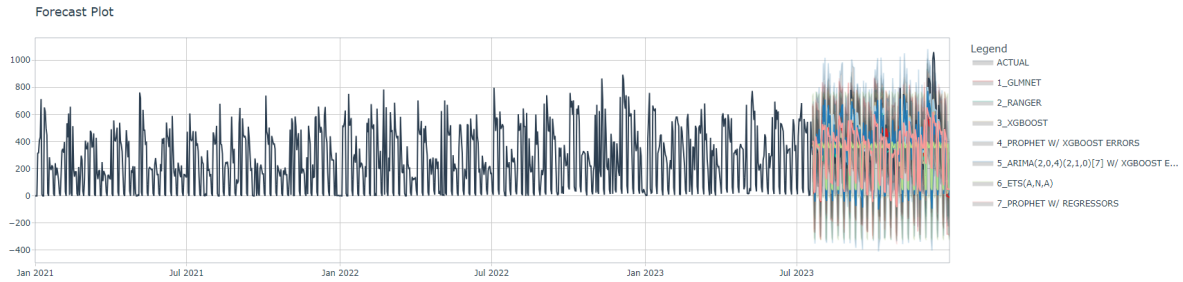
```
recipe_spec <- recipe(tel ~ Date + dayofweek + Project_A + Promotion_1 +
                        Promotion_2 + Promotion_3 + Promotion_4 + Promotion_5 +
                        LTO_Launch + Event + weight, training(Splits)) %>%
  step_timeseries_signature(Date) %>%
  step_rm(matches("(hour)|(minute)|(second)|(am.pm)|(xts)|(iso)|(lbl)")) %>%
  # step_zv(all_predictors()) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE)

############################
### ML Models
# Elastic Net
fit_glmnet <- linear_reg(penalty = 0.1, mixture = 0.5) %>%
  set_engine("glmnet")

fit_glmnet <- workflow() %>%
  add_model(fit_glmnet) %>%
  add_recipe(recipe_spec %>% step_rm(Date)) %>%
  fit(training(Splits))

# Random Forest
fit_rf<- workflow() %>%
  add_model(rand_forest("regression") %>% set_engine("ranger")) %>%
  add_recipe(recipe_spec %>% step_rm(Date)) %>%
  fit(training(Splits))

# Gradient Boost
fit_xgb <- workflow() %>%
  add_model(boost_tree("regression") %>% set_engine("xgboost")) %>%
```

```
  add_recipe(recipe_spec %>% step_rm(Date)) %>%
  fit(training(Splits))

# Prophet Boost Hybrid Model
fit_prophet_boost <- workflow() %>%
  add_model(prophet_boost("regression", seasonality_yearly = TRUE) %>% set_engine("prophet_xgboost")) %>%
  add_recipe(recipe_spec) %>%
  fit(training(Splits))

# Arima Boost
fit_arima_boosted <- workflow() %>%
  add_model(arima_boost("regression") %>% set_engine("auto_arima_xgboost")) %>%
  add_recipe(recipe_spec) %>%
  fit(training(Splits))

############################
### Non ML Models
# Exponential Smoothing
fit_ets <- workflow() %>%
  add_model(exp_smoothing() %>% set_engine("ets")) %>%
  add_recipe(recipe_spec) %>%
  fit(training(Splits))
# Prophet
fit_prophet <- workflow() %>%
  add_model(prophet_reg() %>% set_engine("prophet")) %>%
  add_recipe(recipe_spec) %>%
  fit(training(Splits))

############################
### Add fitted model to a Model Table
models_tbl <-
  modeltime_table(
    fit_glmnet,
    fit_rf,
    fit_xgb,
    fit_prophet_boost,
    fit_arima_boosted,
    fit_ets,
    fit_prophet
  )
############################

rm(
    fit_glmnet,
    fit_rf,
    fit_xgb,
    fit_prophet_boost,
```

```
    fit_arima_boosted,
    fit_ets,
    fit_prophet,
    recipe_spec
)

### Calibration
calibration_table <- models_tbl %>%
  modeltime_calibrate(new_data = testing(Splits))
```

Plot those Models to Test Data

```
calibration_table %>%
  modeltime_forecast(
    new_data = testing(Splits),
    actual_data = order_tel
    ) %>%
  plot_modeltime_forecast(.interactive = TRUE)
```



Compare Accuracy

```
calibration_table %>%
  modeltime_accuracy() %>%
  table_modeltime_accuracy(
    .interactive = FALSE
  )
```

Accuracy Table

.model_id	.model_desc	.type	mae	mape	mase	smape	rmse	rsq
1	GLMNET	Test	105.35	151.85	0.56	57.73	142.49	0.70
2	RANGER	Test	61.10	37.10	0.32	25.80	95.16	0.86
3	XGBOOST	Test	66.78	33.66	0.35	28.46	100.47	0.84
4	PROPHET W/ XGBOOST ERRORS	Test	78.24	126.39	0.41	44.69	107.58	0.85
5	ARIMA(2,0,4)(2,1,0)[7] W/ XGBOOST ERRORS	Test	110.42	117.31	0.58	65.13	157.23	0.64

6	ETS(A,N,A)	Test	129.13	164.94	0.68	49.17	188.16	0.48
7	PROPHET W/ REGRESSORS	Test	106.32	154.99	0.56	55.75	139.54	0.71

Create a Recipe specification for the Promotion Data

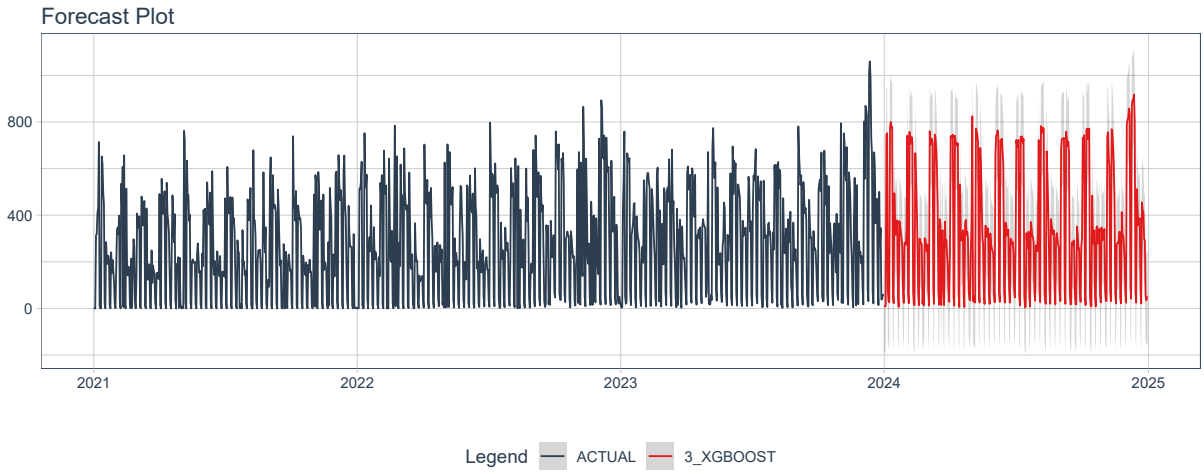
```
dates <- unique(order_tel$Date)
`%ni%` <- Negate(`%in%`)

Promotion <- Promotion %>%
  select(Date, Project_A, Promotion_1, Promotion_2, Promotion_3, Promotion_4,
         Promotion_5, LTO_Launch, Event, weight, dayofweek) %>%
  filter(Date %ni% dates)

Promotion <- recipe(Project_A ~ Date + Promotion_1 + Promotion_2 + Promotion_3 +
                      Promotion_4 + Promotion_5 + LTO_Launch + Event + weight +
                      dayofweek, Promotion) %>%
  step_timeseries_signature(Date) %>%
  step_rm(matches("(hour)|(minute)|(second)|(am.pm)|(xts)|(iso)|(lbl)")) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  prep() %>%
  juice()

head(Promotion)
```

```
## # A tibble: 6 x 27
##   Date       Promotion_1 Promotion_2 Promotion_3 Promotion_4 Promotion_5
##   <date>           <int>       <int>       <int>       <int>       <int>
## 1 2024-01-01           1           0           0           0           0
## 2 2024-01-02           1           0           0           0           0
## 3 2024-01-03           1           0           0           0           0
## 4 2024-01-04           1           0           0           0           0
## 5 2024-01-05           1           0           0           0           0
## 6 2024-01-06           1           0           0           0           0
## # i 21 more variables: LTO_Launch <int>, Event <int>, weight <dbl>,
## #   dayofweek <int>, Project_A <int>, Date_index.num <dbl>, Date_year <int>,
## #   Date_half <int>, Date_quarter <int>, Date_month <int>, Date_day <int>,
## #   Date_wday <int>, Date_mday <int>, Date_qday <int>, Date_yday <int>,
## #   Date_mweek <int>, Date_week <int>, Date_week2 <int>, Date_week3 <int>,
## #   Date_week4 <int>, Date_mday7 <int>
```

Choose the most Accuracy and Forecast

```
Forecast <- calibration_table %>%
  filter(.model_id %in% c(3)) %>%
  modeltime_refit(order_tel) %>%
```

```r
  modeltime_forecast(
    # h = "5 months",
    actual_data = order_tel,
    new_data = Promotion
  )

Export <- Forecast %>%
  filter(.model_desc != "ACTUAL") %>%
  select(-.model_id, -.key) %>%
  spread(key = .model_desc, value = .value) %>%
  mutate(.index = ymd(.index))

write_csv(Export, "forecast/All Telephone Order_forecast.csv")

Forecast %>%
  plot_modeltime_forecast(.interactive = FALSE)
```



Conclusion

The market for this industry has been favorable for the Japanese market since Corona in 2020, and sales have been on the rise. Although online advertising has greatly promoted the market and increased the number of buyers, it seems that the influence of promotions alone is no longer highly accurate in predicting orders due to the manner in which promotions are launched. Unpredictable system errors may reduce accuracy because of the extreme increase in phone orders when online orders cannot be placed. In addition, due to the time difference with the U.S., the impact of promotion deadlines and system outages is significant, especially when switching to daylight saving time, which causes confusion, and including data on these factors may improve accuracy. One way to do this would be to also consider the impact of rising prices, falling stock prices, natural disasters, and a weakening yen. Ideally, we would like to list MAPE at 10% or less, RMSE at 10 for each type, and overall accuracy at 20 or less.