

KNN PRODUCTOS SEMANALES

Miguel Angel Villegas

2025-03-11

```
library(readxl)
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method              from
##   as.zoo.data.frame zoo

library(TSstudio)
library(tsfknn)
```

Introducción

Se descargan los valores de los productos ,luego se crea la serie temporal con una frecuencia semanal. Puesto que es un método no-paramétrico no existe información de AIC y de los residuales del modelo resultante. Por lo que el criterio a considerar para evaluar el modelo serán las métricas de exactitud, además el mejor modelo se determina en base al número de “k” vecinos en un vector de valores, que dependiendo el número de retrasos se puede maximizar pues el modelo tolera un número máximo de ambos combinados, en caso de sobre pasar el numero máximo de retrasos o vecinos próximos se despliega una advertencia la cual indica la imposibilidad de realizar el cálculo.

Se obtienen los datos.

```
ruta_prod <- "/cloud/project/df_dif.xlsx"
excel_sheets(ruta_prod)

## [1] "Sheet 1"

df_dif <- as.data.frame(read_xlsx(path = ruta_prod, col_names = T,
                                sheet = "Sheet 1"))
head(df_dif)

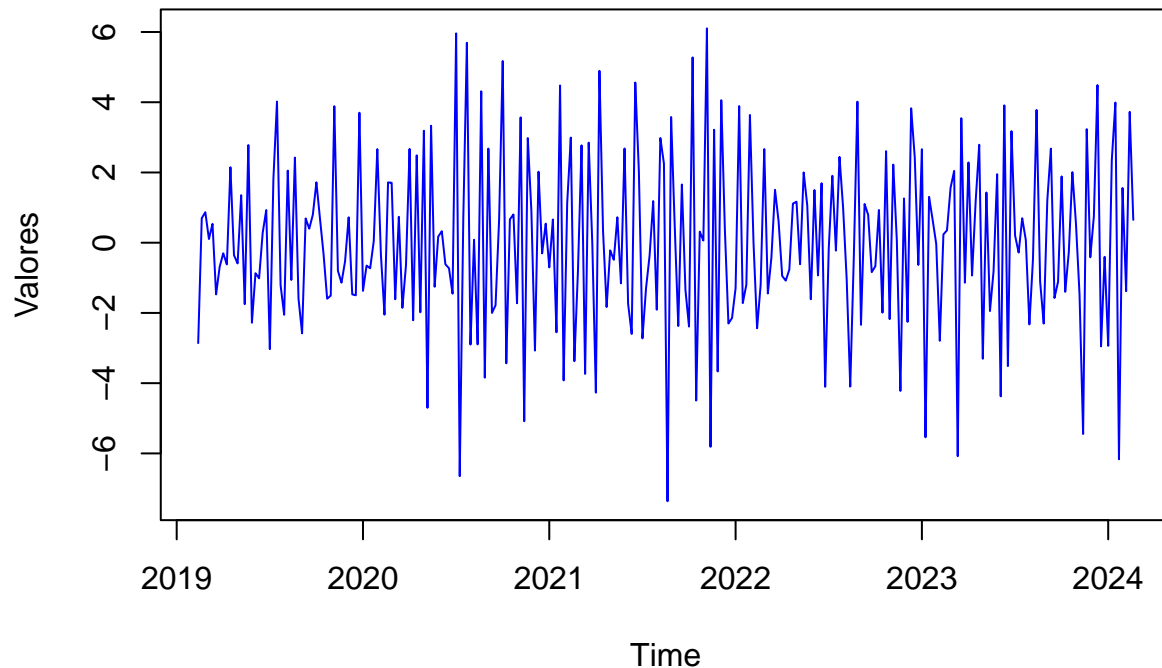
##           dif      Fecha
## 1 -2.8586186 2019-07-03
## 2  0.6996184 2019-07-04
## 3  0.8669633 2019-07-05
## 4  0.1055037 2019-07-06
## 5  0.5360249 2019-07-08
## 6 -1.4721819 2019-07-09
```

Serie de tiempo

```
serie_KNN_PROD_sem <- ts(data = df_dif$dif,
                        start = c(2019,07,03), end = c(2024,08,05),
                        frequency = 52)
```

Gráficas de serie de tiempo

Serie mensual de productos



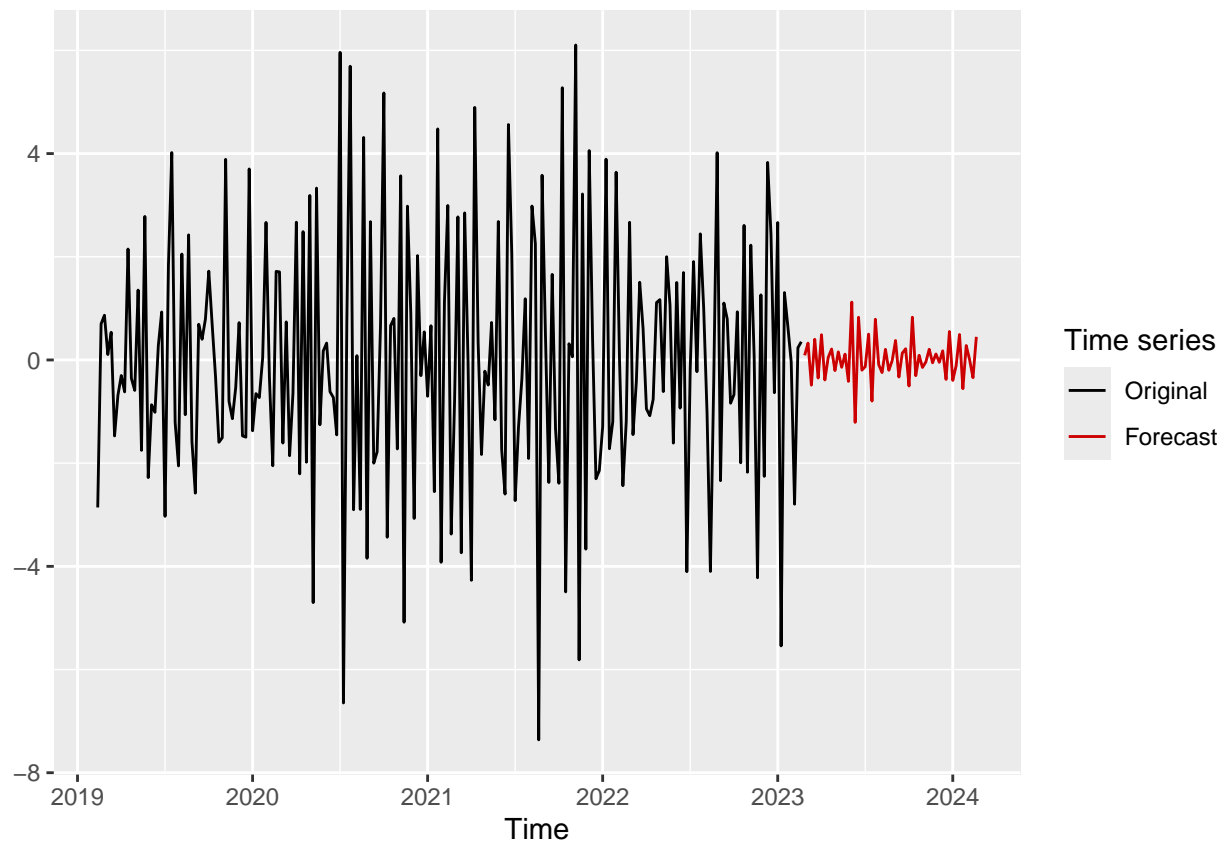
Serie de entrenamiento y prueba

```
dividida_knn_PROD_sem <- ts_split(serie_KNN_PROD_sem,  
                                  sample.out = round(length(serie_KNN_PROD_sem)*0.2))  
  
train_knn_PROD_sem <- dividida_knn_PROD_sem$train  
  
test_knn_PROD_sem <- dividida_knn_PROD_sem$test
```

Modelo KNN con la estrategia de múltiples entradas y múltiples salidas.

```
pronosticos_knn_PROD_sem <- knn_forecasting(train_knn_PROD_sem,  
                                            h = length(test_knn_PROD_sem),  
                                            lags = c(1:52),  
                                            k = c(2:52), msas = "MIMO")
```

Gráfica pronóstico.

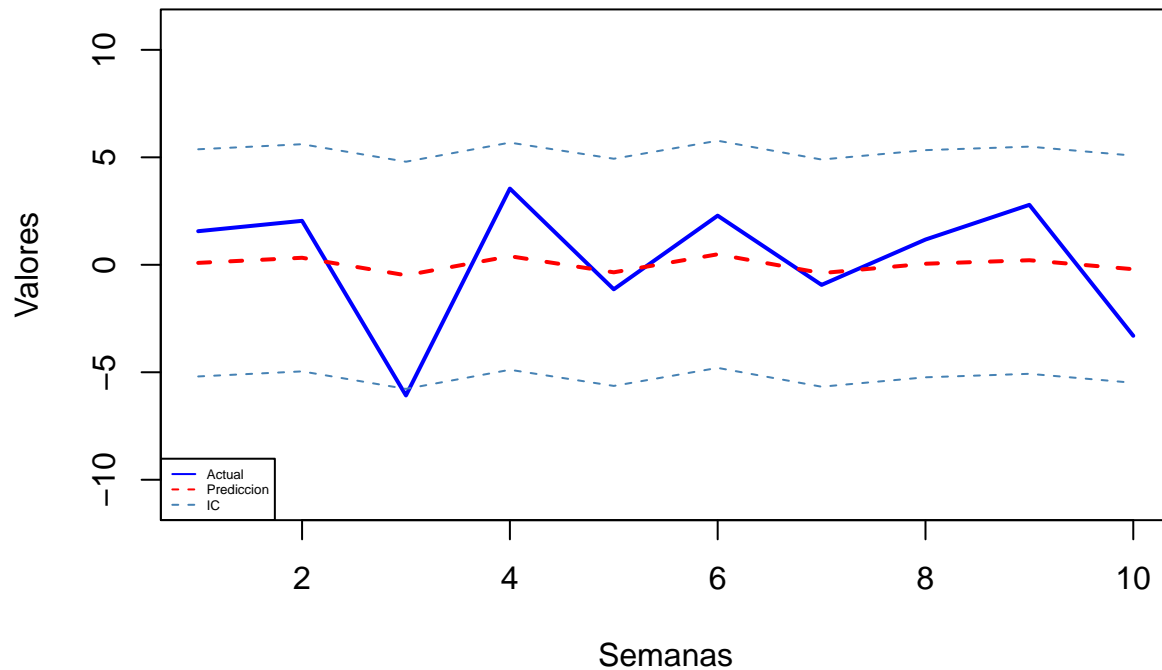


Intervalo de confianza.

```
erro_PROD_knn_sem <- ( test_knn_PROD_sem - pronosticos_knn_PROD_sem$prediction)
sd_errores_PROD_knn_sem <- sd(erro_PROD_knn_sem, na.rm = T)
margen_error_PROD_knn_sem <- sd_errores_PROD_knn_sem * 1.959964 # qnorm(0.975)
# Limites
pronosticos_knn_PROD_sem$inferior <- pronosticos_knn_PROD_sem$prediction - margen_error_PROD_knn_sem
pronosticos_knn_PROD_sem$superior <- pronosticos_knn_PROD_sem$prediction + margen_error_PROD_knn_sem
```

Gráfica con los valores de prueba y pronóstico.

Serie Prueba – Pronostico KKN semanal total



Medidas de exactitud.

```
accuracy(pronosticos_knn_PROD_sem$prediction,x = test_knn_PROD_sem)
```

```
##           ME      RMSE      MAE      MPE      MAPE      ACF1 Theil's U
## Test set 0.04682072 2.670496 2.189752 85.6034 121.8601 -0.4807777 0.9189619
```

```
#           ME      RMSE      MAE      MPE      MAPE      ACF1 Theil's U
# Test set 0.04682072 2.670496 2.189752 85.6034 121.8601 -0.4807777 0.9189619
```

```
accuracy(pronosticos_knn_PROD_sem$prediction[1:10],x = test_knn_PROD_sem[1:10])
```

```
##           ME      RMSE      MAE      MPE      MAPE
## Test set 0.1810232 2.606608 2.186588 84.75038 84.75038
```

```
#           ME      RMSE      MAE      MPE      MAPE
# Test set 0.1810232 2.606608 2.186588 84.75038 84.75038
```

Se realiza el método rolling del paquete “tsfknn” que evaluó la precisión de origen rotativo de la predicción de un modelo.

```
prono_knn_PROD_roll_sem <- knn_forecasting(serie_KNN_PROD_sem,
                                          h = length(test_knn_PROD_sem),
                                          lags = c(1:52),
                                          k = c(1:52), msas = "MIMO")
roll_PROD_sem <- rolling_origin(knnf = prono_knn_PROD_roll_sem,
                               h = length(test_knn_PROD_sem), rolling = T)
```

Evitando el cero absoluto.

```
prueba_numeric_sem <- as.numeric(test_knn_PROD_sem)
prueba_numeric_sem[prueba_numeric_sem == 0] <- 1e-6
```

Métricas con una previsión de 10 semanas.

```
accuracy(prono_knn_PROD_roll_sem$prediction,x = prueba_numeric_sem)
```

```
##           ME      RMSE      MAE      MPE      MAPE
## Test set 0.01632381 2.769352 2.244057 113.7758 122.1823
```

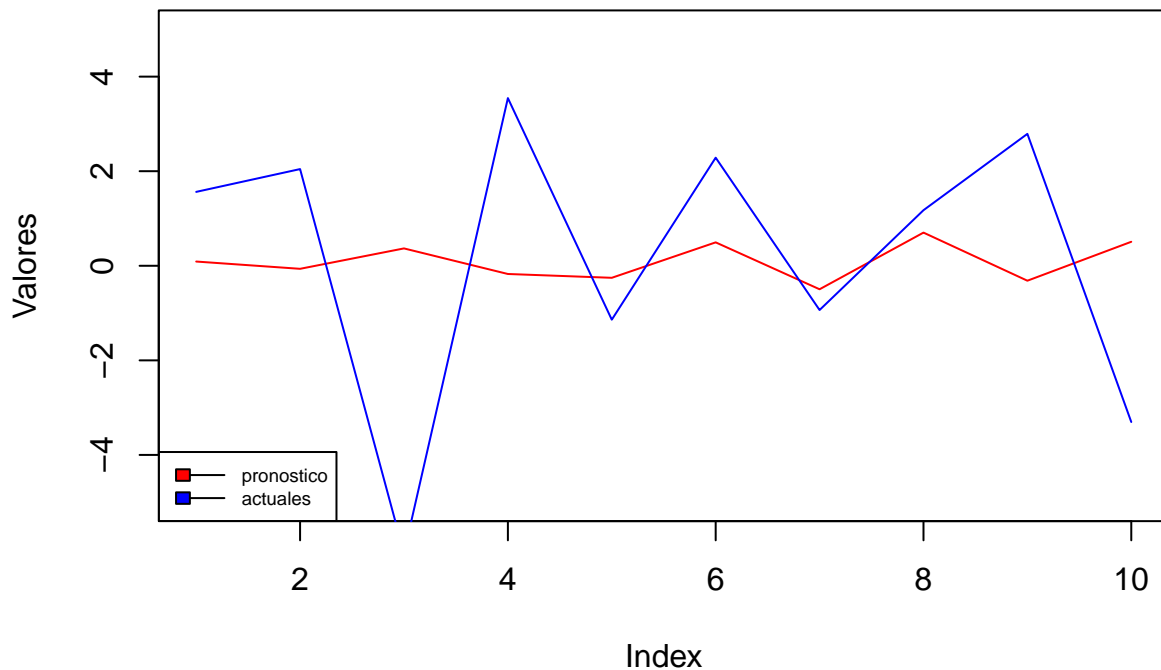
```
#           ME      RMSE      MAE      MPE      MAPE
# Test set 0.01632381 2.769352 2.244057 113.7758 122.1823
```

```
accuracy(prono_knn_PROD_roll_sem$prediction[1:10],x = prueba_numeric_sem[1:10])
```

```
##           ME      RMSE      MAE      MPE      MAPE
## Test set 0.1089595 3.009593 2.42531 87.82648 87.82648
```

```
#           ME      RMSE      MAE      MPE      MAPE
# Test set 0.1089595 3.009593 2.42531 87.82648 87.82648
```

Serie de pronósticos semanales de productos



Conclusiones

El modelo resultante por el método sin rolling ofrece un mejor resultado en las medidas de exactitud.