

# KNN PRODUCTOS MENSUALES

Miguel Angel Villegas

2025-03-09

```
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 mensual. Puesto que es un método no-parámétrico no existe información de AIC y de los residuales del modelo resultante. Por lo que el criterio a considerar para evaluar el modelo seran las metricas 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)

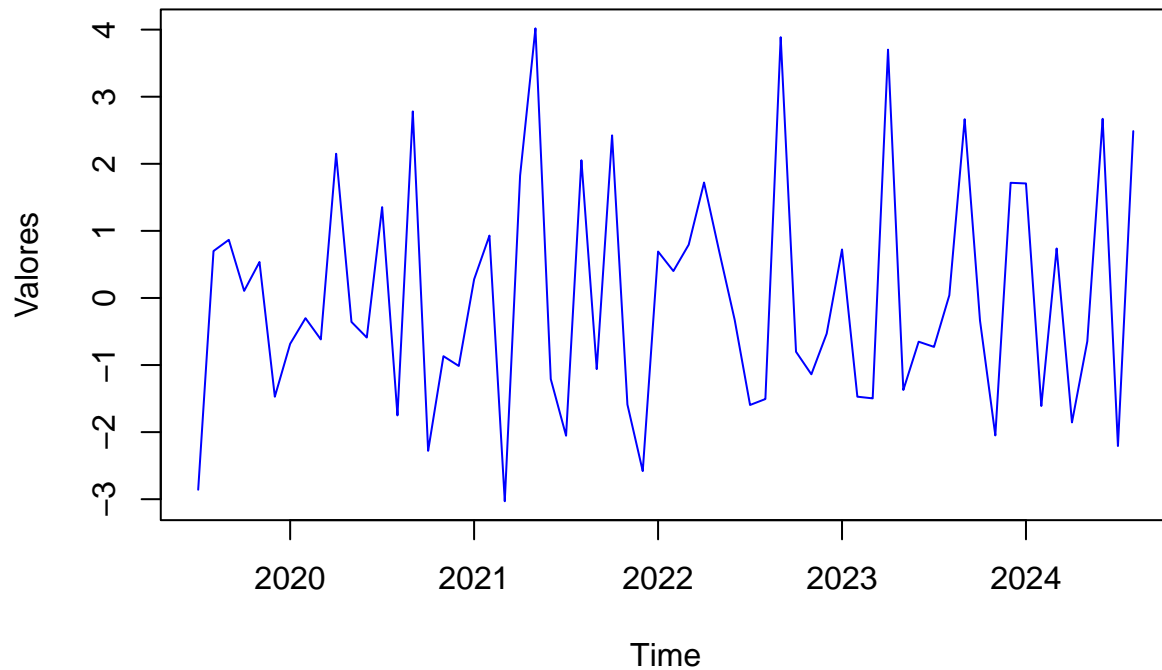
##           dif      Fecha
## 1 -2.8586186 2019-07-03
## 2  0.6996184 2019-07-04
## 3  0.8669633 2019-07-05
## 4  0.1055037 2019-07-06
## 5  0.5360249 2019-07-08
## 6 -1.4721819 2019-07-09
```

Serie de tiempo

```
serie_KNN_PROD_mes <- ts(data = df_dif$dif,
                        start = c(2019,07,03), end = c(2024,08,05),
                        frequency = 12)
```

Gráficas de serie de tiempo

## Serie mensual de productos



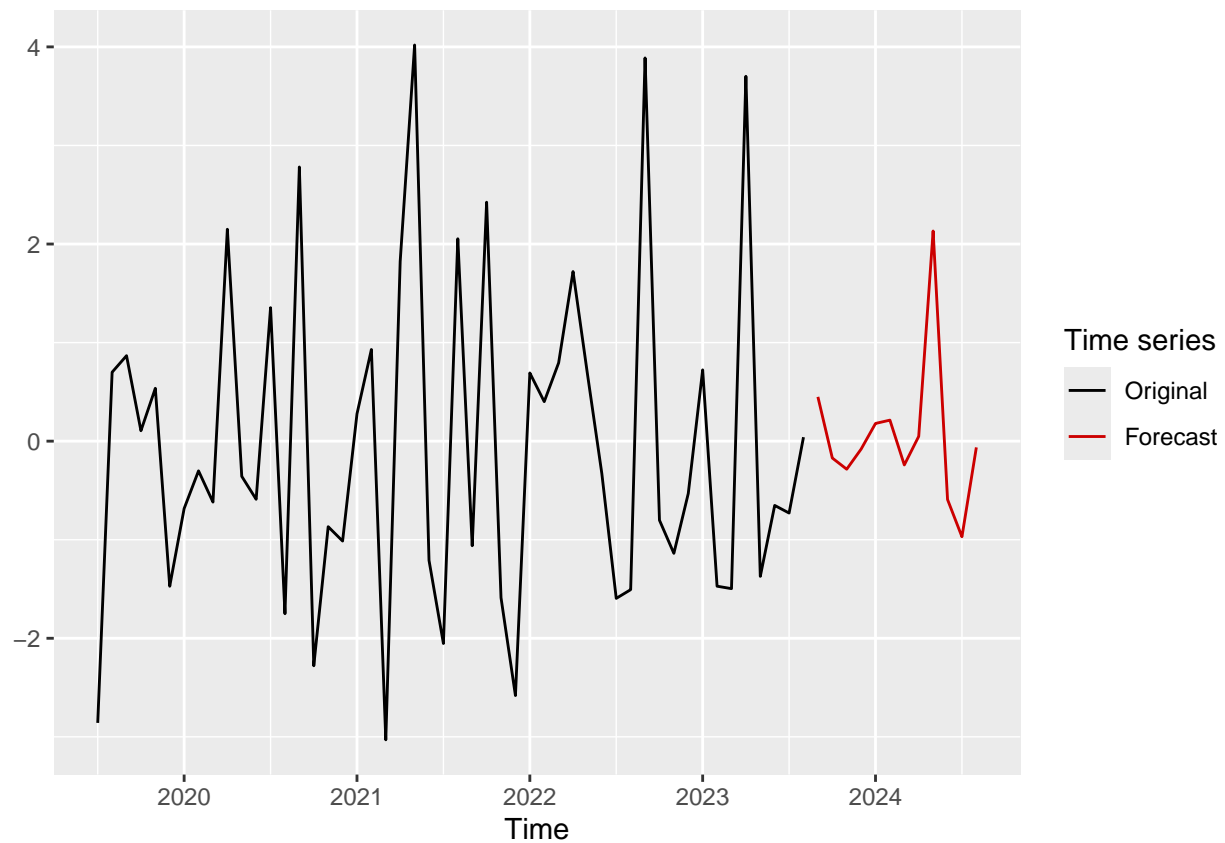
Serie de entrenamiento y prueba

```
dividida_knn_PROD_mes <- ts_split(serie_KNN_PROD_mes,  
                                  sample.out = round(length(serie_KNN_PROD_mes)*0.2))  
  
train_knn_PROD_mes <- dividida_knn_PROD_mes$train  
  
test_knn_PROD_mes <- dividida_knn_PROD_mes$test
```

## Modelo KNN con la estrategia de multiples entradas y multiples salidas

```
pronosticos_knn_PROD_mes <- knn_forecasting(train_knn_PROD_mes,  
                                             h = length(test_knn_PROD_mes),  
                                             lags = c(1:12),  
                                             k = c(2:12), msas = "MIMO")
```

Gráfica pronóstico

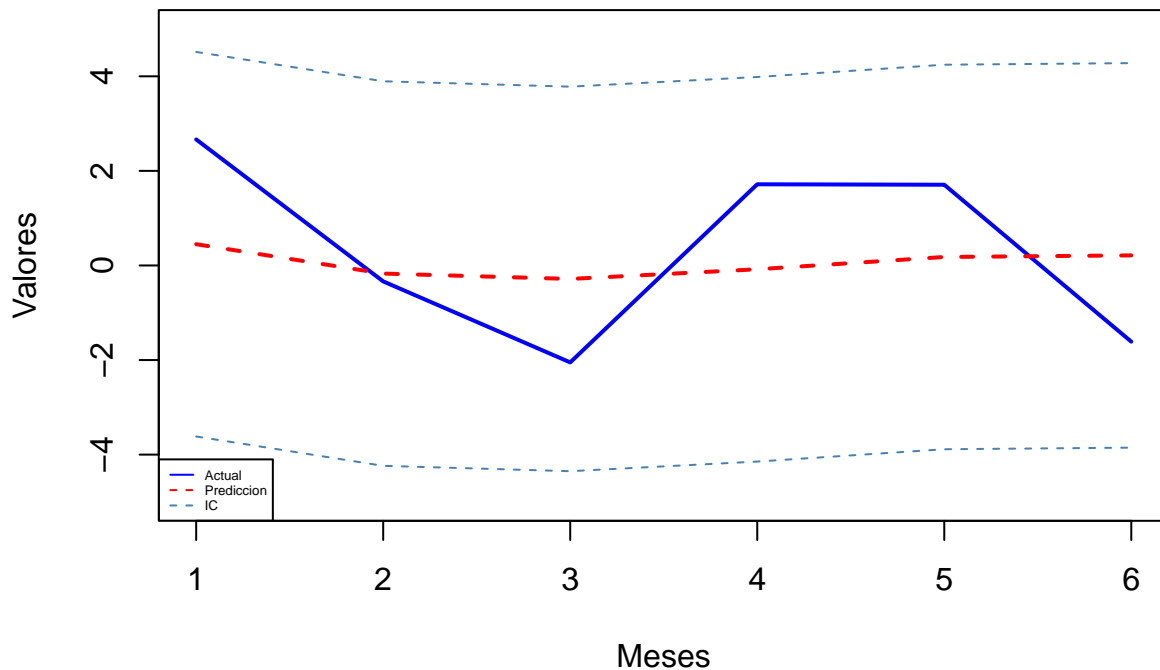


Intervalo de confianza

```
erro_PROD_knn_mes <- ( test_knn_PROD_mes - pronosticos_knn_PROD_mes$prediction)
sd_errores_PROD_knn_mes <- sd(erro_PROD_knn_mes, na.rm = T)
margen_error_PROD_knn_mes <- sd_errores_PROD_knn_mes * 1.96 # qnorm(0.975)
# Limites
pronosticos_knn_PROD_mes$inferior <- pronosticos_knn_PROD_mes$prediction - margen_error_PROD_knn_mes
pronosticos_knn_PROD_mes$superior <- pronosticos_knn_PROD_mes$prediction + margen_error_PROD_knn_mes
```

Gráfica con los valores de prueba y pronóstico.

## Serie de pronósticos mensuales de productos



Medidas de exactitud

```
accuracy(pronosticos_knn_PROD_mes$prediction,x = test_knn_PROD_mes)
```

```
##           ME      RMSE      MAE      MPE      MAPE      ACF1 Theil's U
## Test set 0.2216569 1.998473 1.833902 122.6632 122.6632 -0.3694361 0.8929086
```

```
#           ME      RMSE      MAE      MPE      MAPE      ACF1 Theil's U
# Test set 0.1654994 1.787765 1.597842 237.3555 237.3555 -0.2981265 0.8896603
```

```
accuracy(pronosticos_knn_PROD_mes$prediction[1:6],x = test_knn_PROD_mes[1:6])
```

```
##           ME      RMSE      MAE      MPE      MAPE
## Test set 0.2977485 1.680412 1.549156 87.63299 87.63299
```

```
#           ME      RMSE      MAE      MPE      MAPE
# Test set 0.7376441 1.574764 1.415285 330.2587 330.2587
```

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

```
prono_knn_PROD_roll_mes <- knn_forecasting(serie_KNN_PROD_mes,
                                          h = length(test_knn_PROD_mes),
                                          lags = c(1:12),
                                          k = c(2:12), msas = "MIMO")
roll_PROD_mes <- rolling_origin(knnf = prono_knn_PROD_roll_mes,
                               h = length(test_knn_PROD_mes), rolling = T)
```

Evitando el cero absoluto

```
prueba_numeric_mes <- as.numeric(test_knn_PROD_mes)
prueba_numeric_mes[prueba_numeric_mes == 0] <- 1e-6
```

```
accuracy(prono_knn_PROD_roll_mes$prediction,x = prueba_numeric_mes)
```

```
##           ME      RMSE      MAE      MPE      MAPE
## Test set 0.1089979 1.909413 1.677114 94.28222 94.28222
```

```
#           ME      RMSE      MAE      MPE      MAPE
# Test set -0.1697463 1.931223 1.689193 -191.7965 379.705
```

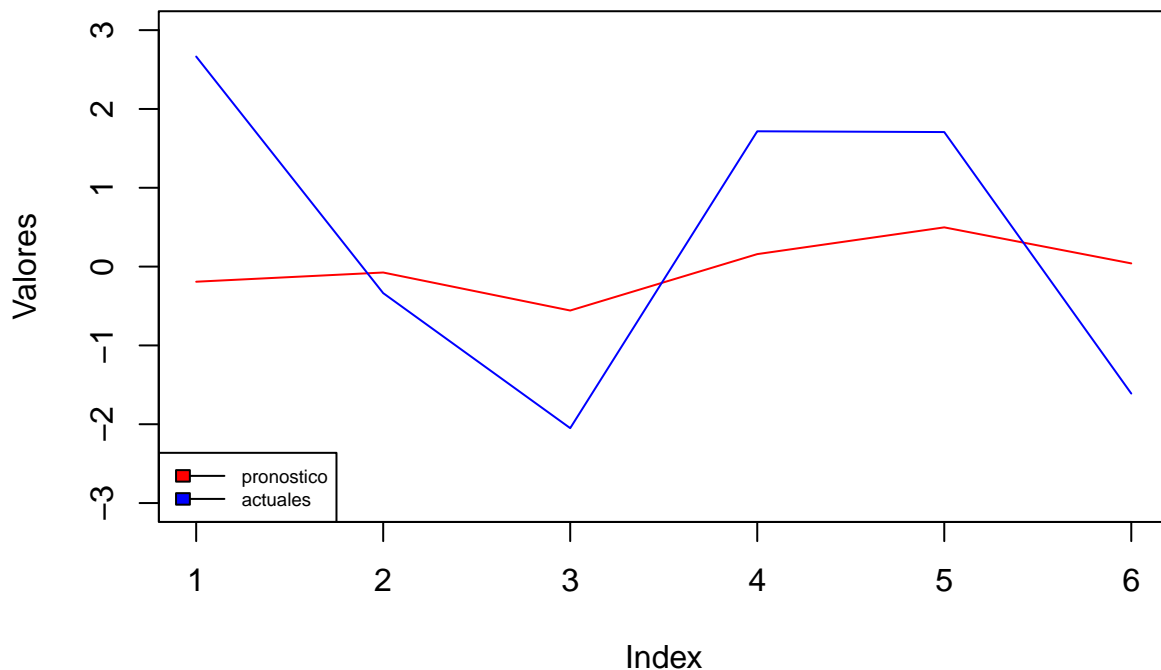
Metricas con una prevision de 6 meses

```
accuracy(prono_knn_PROD_roll_mes$prediction[1:6],x = prueba_numeric_mes[1:6])
```

```
##           ME      RMSE      MAE      MPE      MAPE
## Test set 0.3698512 1.687008 1.504995 86.97358 86.97358
```

```
#           ME      RMSE      MAE      MPE      MAPE
# Test set 0.4364375 1.682402 1.46633 82.00908 82.00908
```

## Serie de pronósticos mensuales de productos



## Conclusiones

El modelo resultante por el método sin rolling ofrece un mejor resultado.