



SERIES TEMPORALES

HW FINAL: Viajeros Baleares: TIB

Predicciones diarias y semanales

Vega Ramiro, José María

Contenido

Contenido	1
INDICE DE ILUSTRACIONES	1
1. REPORT	3
1. Tratamiento de los datos.....	3
1. ETS 4	
1. Predicción diaria.....	4
1. Predicción semanal.....	5
2. ARIMA 6	
1. Predicción diaria.....	7
2. Predicción semanal.....	8
3. THETA y 4THETA.....	10
1. Predicción diaria.....	10
2. Predicción semanal.....	13
3. TBATS 15	
1. Predicción diaria.....	16
2. Predicción semanal.....	18
4. PROPHET (diaria).....	20
5. NEURALPROPHET (semanal)	22
6. CONCLUSIONES	25

INDICE DE ILUSTRACIONES

<u>Ilustración 1: Tablas mape diario y semanal</u>	3
<u>Ilustración 2: Parmetros modelo ets diario</u>	4
<u>Ilustración 3: Prediccion ets diario</u>	4
<u>Ilustración 4: Parámetros modelo ets emanal</u>	4
<u>Ilustración 5: Predicción ets semanal</u>	5
<u>Ilustración 6: métricas ets semanal</u>	5
<u>Ilustración 7: Predicción mes enero semanal</u>	6
<u>Ilustración 8: Parámetros arma diario</u>	7
<u>Ilustración 9: Resumen modelo ets diario</u>	7
<u>Ilustración 10: Predicciones arima diario</u>	7
<u>Ilustración 11: Métricas arma diario</u>	8
<u>Ilustración 12: Parámetros arima semanal</u>	8

<u>Ilustración 13: resumen arima semanal</u>	8
<u>Ilustración 14: predicciones arima semanal</u>	9
<u>Ilustración 15: Métricas arima semanal</u>	9
<u>Ilustración 16: Parámetros theta y 4theta diario</u>	10
<u>Ilustración 17: Predicciones theta y 4theta diario</u>	11
<u>Ilustración 18: Métricas theta y 4theta</u>	12
<u>Ilustración 19: Parámetros theta semanal</u>	12
<u>Ilustración 20: Parámetros 4theta semanal</u>	13
<u>Ilustración 21: Predicciones theta y 4theta semanal</u>	14
<u>Ilustración 22: Parámetros TBATS diario</u>	15
<u>Ilustración 23: Predicciones diario tbats</u>	16
<u>Ilustración 24: métricas Tbats diario</u>	16
<u>Ilustración 25: predicciones enero TBATS diario</u>	17
<u>Ilustración 26: Parámetros tbats semanal</u>	17
<u>Ilustración 27: predicciones tbats semanal</u>	18
<u>Ilustración 28: Métricas tbats semanal</u>	19
<u>Ilustración 29: predicción enero tbats semanal</u>	19
<u>Ilustración 30: Parámetros prophet diario</u>	20
<u>Ilustración 31: predicciones prophet diario</u>	21
<u>Ilustración 32: Parámetros prophet diario</u>	21
<u>Ilustración 33: Parámetros NeuralProphet semanal</u>	22
<u>Ilustración 34: Predicciones NeuralProphet semanal</u>	24
<u>Ilustración 35: Métricas NeuralProphet semanal</u>	24

1. REPORT

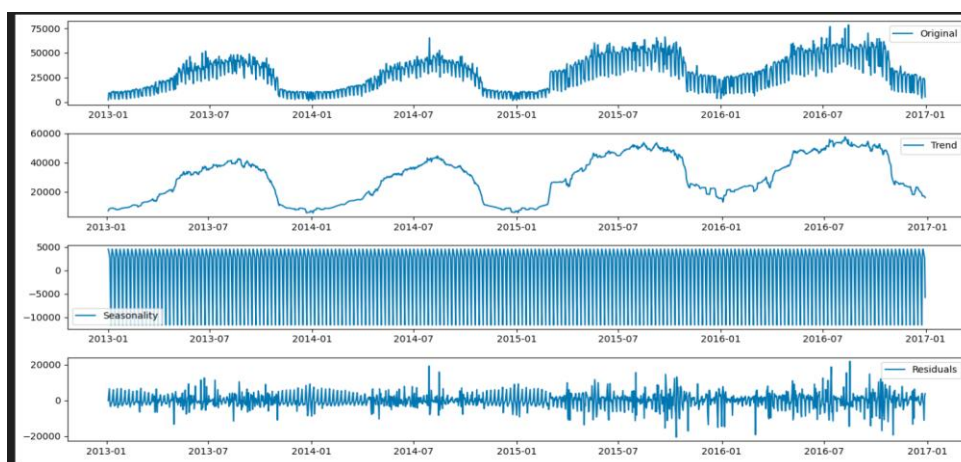
En este report, se explica los diferentes modelos que se utilizaron para hacer las predicciones, los resultados que se obtuvieron y el porque de que no hayan sido escogidos

1. Tratamiento de los datos.

En cuanto a los datos que se han utilizado para elaborar el trabajo, contamos con el día, el mes y el año junto a los respectivos clientes de una línea de autobuses. Se ha pedido realizar la predicción tanto diaria como semanal del mes de 2017.

Para la elaboración de la predicción diaria, hemos tenido que realizar transformaciones de los datos que fueron facilitados. En primer lugar, se unificaron tanto el día con el mes y la fecha, recogiendo todo en una misma columna, y dándole su respectivo valor. Posteriormente a esta nueva columna se le asignó formato fecha, y se añadió como index.

Para la elaboración de la predicción semanal, se partió a partir de el dataset de la predicción diaria, lo que se hizo fue agrupar a los visitantes por semanas con la función resample dentro de Python (se puede ver todo el proceso en el notebook "tratamiento datos").



La descomposición se ha realizado usando un modelo aditivo, que es el más común para este tipo de análisis. Los componentes descompuestos son:

Observado (Original): La serie temporal original de los visitantes totales.

Tendencia (Trend): Muestra la tendencia a largo plazo en tus datos, ayudando a identificar si hay un aumento o disminución general en el número de visitantes con el tiempo.

Estacionalidad (Seasonality): Indica patrones que se repiten en intervalos regulares, como variaciones diarias, semanales, mensuales o anuales.

Residuos (Residuals): La variabilidad en los datos que no puede ser explicada por la tendencia o la estacionalidad. Estos pueden incluir eventos aleatorios o anomalías.

Antes de empezar a explicar cada modelo, hemos recogido las diferentes métricas que obtuvimos en cada uno de los modelos a modo de resumen, para tener una idea de porque fueron seleccionados los modelos definitivos.

MAPE	DIARIO	SEMANAL
ETS	39,44%	9,46%
ARIMA	55,56%	8,08%
THETA	42.05%	14.54%
4THETA	44.68%	13.61%
TBATS	14.60%	8,69%
PROPHET	15.84%	6.72%
NeuralProphet	13,65%	12.63%

Ilustración 1: Tablas mape diario y semanal

1. ETS

El modelo ETS (Error, Trend, Seasonality) divide la serie temporal en tres componentes: error aleatorio, tendencia y estacionalidad. Se enfoca en descomponer la serie en tres componentes clave: el error, que representa la variabilidad no sistemática; la tendencia, que refleja la dirección a largo plazo; y la estacionalidad, que captura patrones cíclicos. ETS ofrece distintas especificaciones según la presencia o ausencia de estos componentes.

1. Predicción diaria

Parámetros

```
ets_diario=AutoETS(auto=True, sp=7,seasonal=True, n_jobs=-1)
ets_diario.fit(y_train_diario)

#Predicciones
# Obtener el índice de tiempo de y_test_diario y usarlo como horizonte de predicción
fh = y_test_diario.index

# Realizar la predicción usando el índice de tiempo de y_test_diario
ets_pred_diario = ets_diario.predict(fh)
```

Ilustración 2:Parmaetros modelo ets diario

En cuanto a los parámetros solo hemos puesto el seasonal period =7 refiriéndonos a semanal.

Predicciones

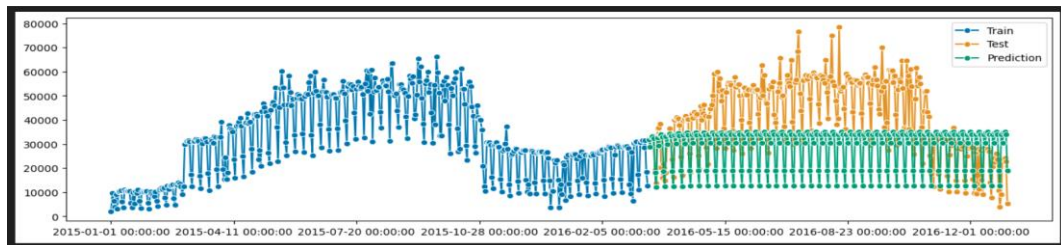


Ilustración 3: Prediccion ets diario

En cuanto a las predicciones se observó que en la serie diaria no es capaz de encontrar ningún tipo de tendencia o estacionalidad, si no que sigue completamente una línea recta dándonos muy malos resultados.

Métricas

1. Predicción semanal

Parámetros

```
ets_semanal=AutoETS(
    error='add',
    trend=None,
    seasonal='mul',
    sp=52,
    n_jobs=-1
)
ets_semanal.fit(y_train_semanal)
```

Ilustración 4:Parámetros modelo ets emanal

1. Sesional period de 52:

Indica que hay una estacionalidad en la serie que se repite cada 52 semanas. Esto sugiere una periodicidad anual, común cuando se trabajan con datos semanales.

2. Error aditivo:

Significa que las variaciones en la serie temporal se suman constantemente, indicando que el impacto de los errores es constante a lo largo del tiempo.

3. Sin tendencia:

La ausencia de una tendencia implica que la serie no muestra un cambio sistemático o dirección a largo plazo en el tiempo.

4. Estacionalidad multiplicativa:

Indica que la amplitud de la estacionalidad varía proporcionalmente con el nivel de la serie. Esto es común cuando la magnitud de la estacionalidad es proporcional al nivel de la serie en sí.

Predicciones

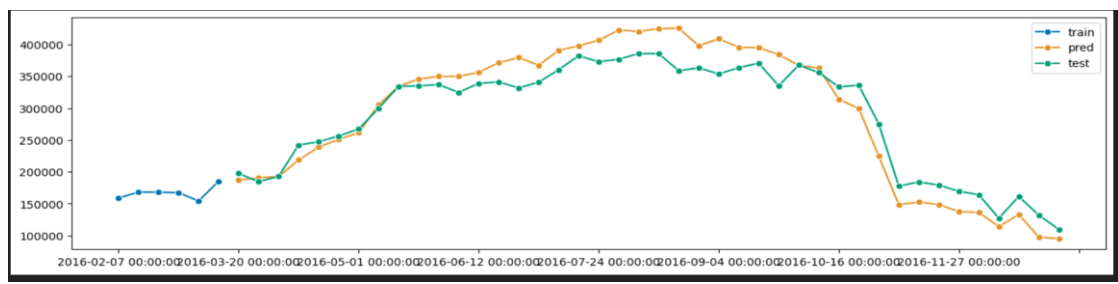


Ilustración 5: Predicción ets semanal

La discrepancia entre las predicciones y los datos reales, especialmente cuando se observan valores más altos en la parte posterior de la predicción, podría indicar varias cosas en el contexto de un modelo ETS, podría indicar que el modelo no está capturando adecuadamente cambios o patrones emergentes en la serie que se manifiestan más adelante en el tiempo, también destacar que el modelo ETS tenga limitaciones al prever eventos inesperados o cambios abruptos en la serie, lo cual es común en situaciones dinámicas.

Métricas

MAPE: 0.09466984318890426
MSE: 886280233.02375
R2 score: 0.8791750608392199

Ilustración 6: métricas ets semanal

En general, aunque el R^2 es alto, es importante tener en cuenta el MAPE y el MSE. Un MAPE del 9.4% es aceptable en muchos casos, pero el MSE elevado indica que hay casos en los que las predicciones pueden alejarse significativamente de los valores reales.

Predicción Enero 2017

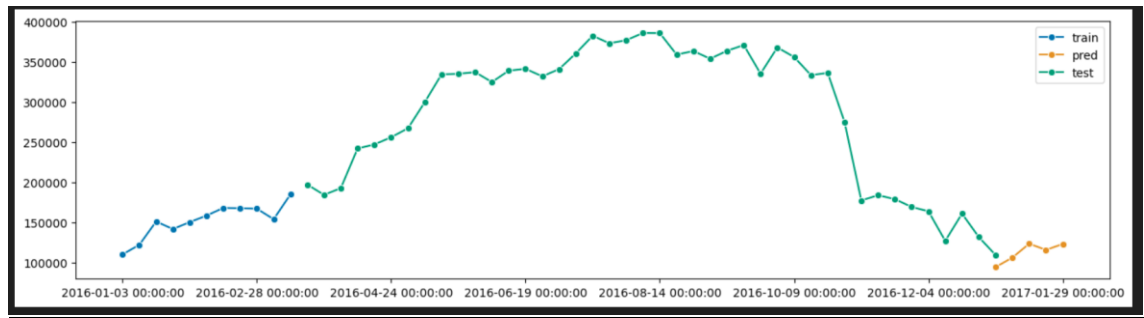


Ilustración 7: Predicción mes enero semanal

Se ha elaborado la predicción para el mes de enero ya que se consideró que las métricas obtenidas eran relativamente buenas, y como en la predicción se estima que los valores aumentarían para el mes de enero al igual que se estima en los modelos definitivos-

2. ARIMA

El modelo ARIMA, que significa Modelo Autoregresivo Integrado de Media Móvil, es una herramienta popular en el análisis de series temporales para pronosticar datos futuros basándose en observaciones pasadas. Combina tres componentes principales: autoregresión (AR), diferenciación (I) para lograr estacionariedad, y media móvil (MA).

El componente AR explica la correlación entre un valor y sus retrasos, el componente I se utiliza para hacer la serie estacionaria diferenciándola, y el componente MA modela el error de la predicción como una combinación lineal de errores pasados.

Este modelo es altamente configurable y puede adaptarse a una gran variedad de estructuras de datos temporales.

1. Predicción diaria

Parámetros y modelo

```
#Entrenamos el modelo
from sktime.forecasting.base import ForecastingHorizon
from sktime.forecasting.arima import AutoARIMA

Arima_diario = AutoARIMA(sp=7, suppress_warnings=True, trace=True)
Arima_diario.fit(y_train_diario_log)
```

Ilustración 8: Parámetros arma diario

SARIMAX Results					
=====					
Dep. Variable:	y	No. Observations:	1168		
Model:	SARIMAX(0, 1, 3)x(2, 0, [1], 7)	Log Likelihood	63.445		
Date:	Sat, 13 Jan 2024	AIC	-112.889		
Time:	12:01:01	BIC	-77.454		
Sample:	01-01-2013	HQIC	-99.523		
	- 03-13-2016				
Covariance Type:	opg				
=====					
	coef	std err	z	P> z	[0.025 0.975]

Ilustración 9: Resumen modelo ets diario

SARIMAX revelan un análisis de una serie temporal compuesta por 1168 observaciones. Este modelo, SARIMAX(0, 1, 3)x(2, 0, [1], 7), incorpora componentes estacionales con un patrón que se repite semanalmente. La salida del modelo proporciona un desglose detallado de cómo cada componente, como los términos autoregresivos y de media móvil, contribuye a las predicciones.

Predicciones

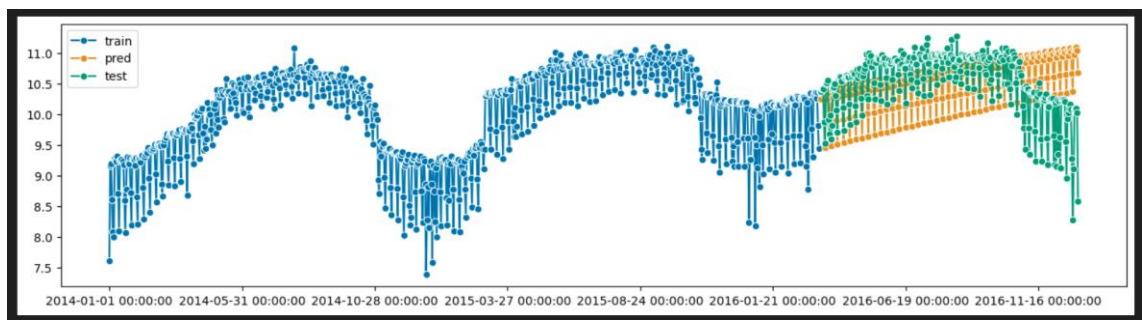


Ilustración 10: Predicciones arima diario

El modelo SARIMAX muestra habilidad para identificar una tendencia general positiva, su limitación en la captura de la estacionalidad y en la respuesta a cambios abruptos puede afectar su capacidad para proporcionar pronósticos precisos en ciertos escenarios. Se podrían explorar ajustes en la especificación del modelo o considerar enfoques más avanzados para mejorar su capacidad predictiva.

Métricas

MAPE: 0.5556613085101777
MSE: 328734806.75579613
R2 score: -0.37189538219619167

Ilustración 11: Métricas arma diario

Estos resultados pueden ser considerados "malos" por varias razones. El MAPE alto sugiere que las predicciones están desviándose considerablemente de los valores reales en términos porcentuales. El MSE elevado indica

que hay grandes errores cuadráticos entre las predicciones y los datos reales, señalando una falta de precisión general. El R^2 negativo sugiere que el modelo no está logrando explicar ni siquiera la variabilidad promedio de los datos, indicando un ajuste deficiente.

2. Predicción semanal

Parámetros y modelo

```
#Entrenamos el modelo
Arima_semanal =AutoARIMA(sp=52,suppress_warnings=True, trace = True)
Arima_semanal.fit(y_train_semanal_log)
```

Ilustración 12: Parámetros arima semanal

SARIMAX Results					
=====					
Dep. Variable:	y		No. Observations:	188	
Model:	SARIMAX(1, 1, 1)x(1, 0, [1, 2], 52)		Log Likelihood	154.073	
Date:	Sat, 13 Jan 2024		AIC	-296.146	
Time:	12:14:49		BIC	-276.759	
Sample:	01-06-2013		HQIC	-288.291	
	- 08-07-2016				
Covariance Type:	opg				
=====					
	coef	std err	z	P> z	[0.025 0.975]

Ilustración 13: resumen arima semanal

1. No. Observations (Número de Observaciones):

Hay 188 observaciones en la serie temporal utilizada para ajustar el modelo.

2. Model (Modelo):

El modelo SARIMAX(1, 1, 1)x(1, 0, [1, 2], 52) especifica la presencia de componentes autoregresivos, de media móvil y estacionales. La estacionalidad se observa cada 52 semanas.

3. Log Likelihood (Verosimilitud Logarítmica):

La verosimilitud logarítmica positiva de 154.073 indica que el modelo tiene un buen ajuste a los datos observados.

Predicciones

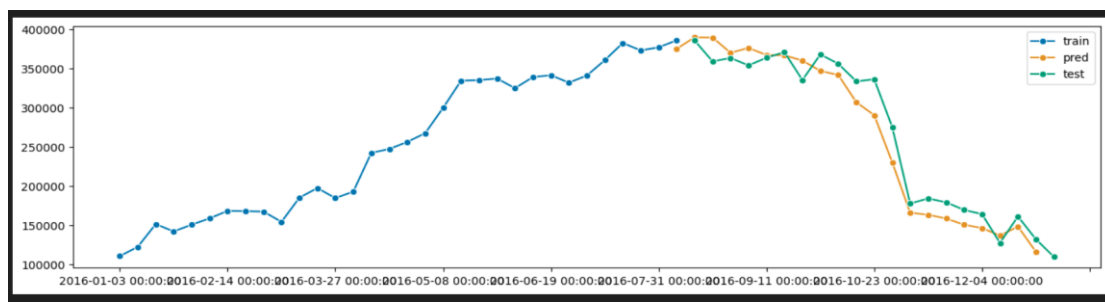


Ilustración 14: predicciones arima semanal

1. Tendencia General:

Ambas series muestran una tendencia descendente en la cantidad de visitantes totales a lo largo del tiempo, particularmente hacia finales de 2016.

2. Estacionalidad:

Algunos patrones estacionales podrían estar presentes en ambas series, donde hay fluctuaciones recurrentes en los datos. Por ejemplo, se observa un descenso pronunciado en las semanas alrededor del 30 de octubre de 2016.

3. Diferencias Específicas:

En los datos de test (a partir de 2016-08-14), hay una mayor variabilidad y fluctuaciones en comparación con la predicción (hasta 2016-12-25).

4. Valores Específicos:

Al comparar valores específicos en fechas comunes, se aprecia que los números en la pred suelen ser ligeramente inferiores a los del test, aunque la diferencia no es extrema.

Métricas

```
MAPE: 0.08088483490649195
MSE: 447082077.13361466
R2 score: 0.9548980515683306
```

Ilustración 15: Métricas arima semanal

El MAPE bajo sugiere que el modelo está produciendo predicciones precisas en términos porcentuales. El MSE alto puede señalar la presencia de errores significativos en algunas predicciones. Sin embargo, el R^2 alto indica que el modelo tiene un buen poder explicativo y se ajusta bien a la variabilidad general de los datos. Es importante considerar estas métricas en conjunto para obtener una evaluación integral del rendimiento del modelo.

3. THETA y 4THETA

Los modelos Theta y 4-Theta son técnicas de pronóstico para series temporales que se han destacado por su simplicidad y efectividad, especialmente en la competición de pronóstico M3. La principal idea detrás del modelo Theta es descomponer la serie temporal en dos o más componentes 'Theta' que son esencialmente líneas rectas obtenidas al modificar la curvatura de la serie original.

El **modelo 4-Theta**, por otro lado, es una extensión del modelo Theta original. Introduce la idea de usar cuatro líneas Theta en lugar de dos. Estas cuatro componentes están diseñadas para capturar diferentes aspectos de la serie temporal

1. Predicción diaria

Parámetros

```
#Entrenamos y predecimos con los mejores parámetros
modelo_4theta_diario=FourTheta(theta=best_grid_model[1]['theta'],
                                model_mode=ModelMode.ADDITIVE,
                                season_mode=SeasonalityMode.MULTIPLICATIVE,
                                trend_mode=TrendMode.EXPONENTIAL
                                )
modelo_4theta_diario.fit(train_diario)
forecast_4theta_diario_bueno = modelo_4theta_diario.predict(len(val_diario))

train_diario.plot(label="train")
val_diario.plot(label="true")
forecast_4theta_diario_bueno.plot(label="prediction")
```

Ilustración 16: Parámetros theta y 4theta diario

- **Theta:** El valor de theta se toma del mejor modelo en una búsqueda en cuadrícula (`best_grid_model[1]['theta']`). El valor específico no se proporciona en tu fragmento de código, pero theta generalmente representa un parámetro de suavizado en modelos de series temporales. Un valor más alto de theta generalmente indica un mayor nivel de suavizado.
- **model_mode:** Este parámetro se establece en `ModelMode.ADDITIVE`, lo que sugiere que el modelo está asumiendo que la relación entre las componentes (tendencia, estacionalidad, y error) es aditiva.
- **season_mode:** Se establece en `SeasonalityMode.MULTIPLICATIVE`, indicando que la estacionalidad se multiplica por las otras componentes del modelo. Esto es común cuando la variación estacional es proporcional al nivel de la serie temporal.

- `trend_mode`: Se establece en `TrendMode.EXPONENTIAL`, lo que sugiere que el modelo asume una tendencia exponencial.

Predicciones

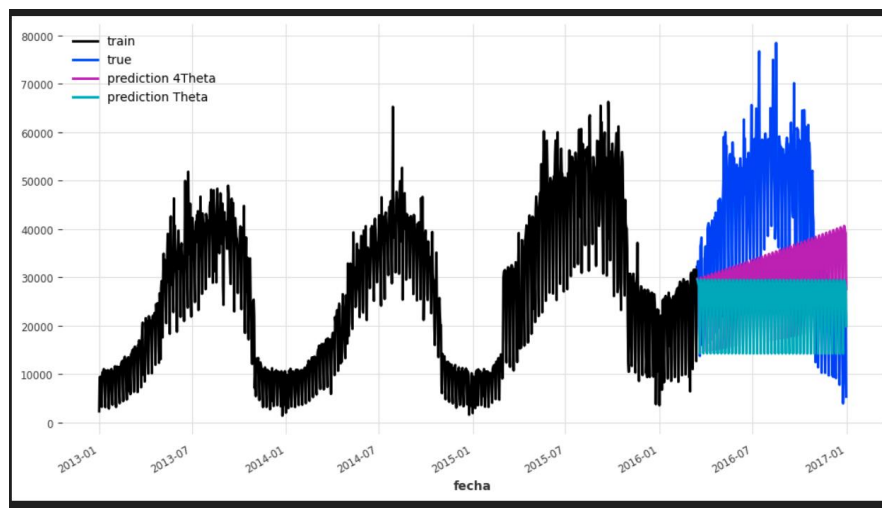


Ilustración 17: Predicciones theta y 4theta diario

En cuanto a las predicciones el modelo theta y 3theta para las predicciones diarias son muy malas, ya que no son capaces observar ningún componente de estacionalidad como la tendencia de la serie, por lo que ha sido descartado rápidamente

Métricas

```
The MAPE is: 44.68, with theta = 0.9795918367346932.  
model FourTheta(seasonality_period=52) obtains MAPE: 42.05%
```

Ilustración 18: Métricas theta y 4theta

1. MAPE: 44.68% con $\theta = 0.9795918367346932$: Esto sugiere que, en promedio, las predicciones realizadas por el modelo tienen un error porcentual absoluto de aproximadamente 44.68%. El valor de θ (que a menudo está asociado con los parámetros de suavizado en modelos de series temporales) se proporciona como 0.9795918367346932.
2. FourTheta(seasonality_period=52) obtiene MAPE: 42.05%: Esto indica que otro modelo llamado FourTheta, con un período de estacionalidad de 52, tiene un MAPE promedio del 42.05%. Comparativamente, este modelo podría ser percibido como mejor ajustado a los datos en comparación con el primer modelo, ya que tiene un MAPE más bajo.

2. Predicción semanal

Parámetros

```
best_theta_model.model_params
```

```
OrderedDict([('theta', 0.9795918367346932),
             ('seasonality_period', None),
             ('season_mode',
              <SeasonalityMode.MULTIPLICATIVE: 'multiplicative'>)])
```

Ilustración 19: Parámetros theta semanal

THETA

- θ : El valor de θ es 0.9795918367346932. Este parámetro suele estar asociado con el suavizado en modelos de series temporales. En particular, en modelos de suavizado exponencial, θ a menudo representa la tasa de suavizado, y un valor más alto de θ indica un mayor nivel de suavizado. En este caso, un valor de 0.9795918367346932 sugiere que el modelo está aplicando un suavizado sustancial a los datos.
- seasonality_period: Este parámetro está establecido en None, lo que significa que el período de estacionalidad no está explícitamente especificado. Podría implicar que el modelo está configurado para detectar automáticamente el período de estacionalidad en los datos.

- `season_mode`: La estacionalidad está configurada en `SeasonalityMode.MULTIPLICATIVE`, lo que indica que el modelo asume que la componente estacional se multiplica por las otras componentes del modelo. En términos simples, la variación estacional se percibe como proporcional al nivel general de la serie temporal.

```
#Elaboramos modelo con mejores hiperparámetros
modelo_4theta=FourTheta(theta=best_grid_model[1]['theta'],
                        model_mode=ModelMode.ADDITIVE,
                        season_mode=SeasonalityMode.ADDITIVE,
                        trend_mode=TrendMode.EXPONENTIAL
                        )
modelo_4theta.fit(train)
forecast_4theta = model.predict(n=43)

train.plot(label="train")
val.plot(label="true")
forecast_4theta.plot(label="prediction")
```

Ilustración 20: Parámetros 4theta semanal

4THETA

`theta`: El valor de `theta` se establece en 1. Un valor de 1 para `theta` sugiere que no se está aplicando suavizado exponencial, ya que `theta` es el parámetro de suavizado y, en este caso, se ha establecido en su máximo valor. Esto podría indicar que el modelo confía más en los datos recientes sin aplicar un suavizado significativo.

`seasonality_period`: El parámetro `seasonality_period` se establece en `None`, indicando que el modelo está configurado para detectar automáticamente el período de estacionalidad en los datos.

`season_mode`: `SeasonMode.ADDITIVE` sugiere que la estacionalidad se trata de manera aditiva. Esto implica que la variación estacional se suma a las otras componentes del modelo.

`model_mode`: `ModelMode.ADDITIVE` indica que el modelo asume que la relación entre las componentes (tendencia, estacionalidad y error) es aditiva.

`trend_mode`: `TrendMode.EXPONENTIAL` sugiere que la tendencia se modela de manera exponencial.

normalization: True indica que se está aplicando normalización. La normalización puede ayudar a que el modelo se comporte de manera más efectiva al ajustar las escalas de las variables.

Predicciones

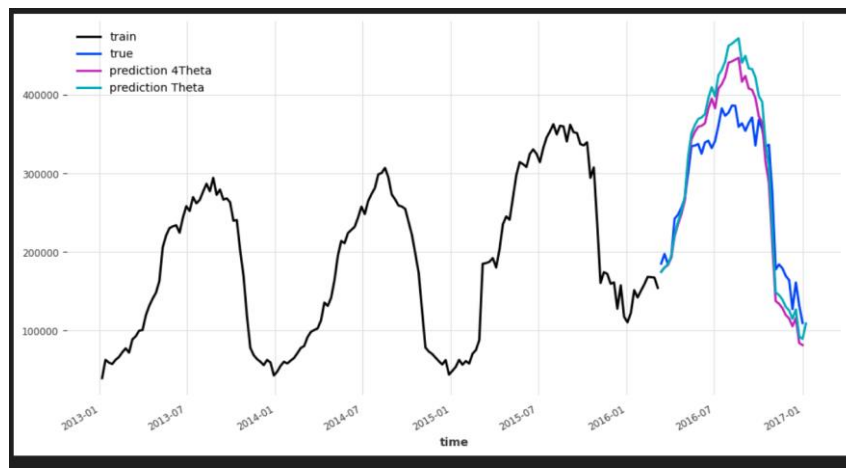


Ilustración 21: Predicciones theta y 4theta semanal

En cuanto a las predicciones comparando ambos modelos theta y 4theta, se observa como los valores de 4theta se encuentran un poco mas bajos que los de Theta acercándose mas a la predicción por arriba pero perdiendo exactitud por debajo.

Métricas

En el modelo Theta el MAPE es del 14.54%: Este valor indica que, en promedio, las predicciones del modelo tienen un error porcentual absoluto del 14.54%. En términos prácticos, esto sugiere que, al utilizar este modelo de series temporales con el valor específico de theta (0.9795918367346932), las predicciones tienden a desviarse del valor real en alrededor del 14.54%.

El modelo 4Theta parece tener un buen rendimiento según el MAPE reportado (14.54%), y el valor de theta proporciona información sobre el nivel de suavizado aplicado en el modelo para adaptarse a los patrones temporales en los datos.

3. TBATS

TBATS es un modelo avanzado para series temporales que puede manejar múltiples estacionalidades, tendencias y transformaciones automáticas de Box-Cox. Aquí están los parámetros que has utilizado:

1. Predicción diaria

Parámetros

```
#Elaboramos el modelo
model_diario =TBATS(
    use_box_cox=None,
    box_cox_bounds=(0, 1),
    use_trend=None,
    use_damped_trend=None,
    seasonal_periods=(7,365), # Estacionalidad diaria y anual (se pueden poner más)
    use_arma_errors=True,
    show_warnings=False,
    multiprocessing_start_method='spawn',
    random_state=0)

#Lo entrenamos
model_diario.fit(train)
```

Ilustración 22: Parámetros TBATS diario

Los parámetros clave incluyen la posible transformación de Box-Cox para estabilizar la varianza, la consideración automática de tendencias (amortiguadas o no), y la inclusión de estacionalidades semanales y anuales a través de `seasonal_periods=(7,365)`. Además, utiliza errores tipo ARMA para capturar dependencias en los errores de la serie, y permite el control sobre las advertencias y la reproducibilidad con los parámetros `show_warnings` y `random_state`. Todo esto contribuye a un modelo flexible y robusto para predecir patrones complejos en datos temporales.

Predicciones

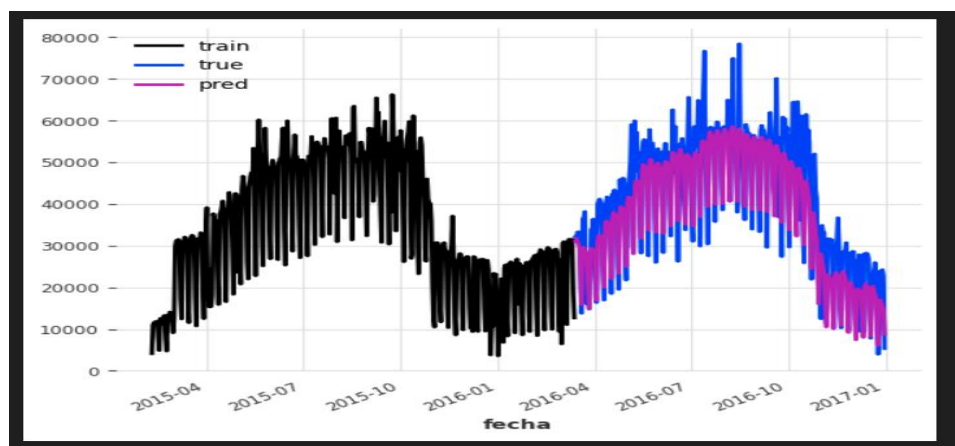


Ilustración 23: Predicciones diario tbats

Se puede observar cómo los resultados de las predicciones al comparlo con test, son buenos porque son capaces de seguir la tendencia, pero nunca llegan a acertar del todo. Se puede sacar en claro que estos datos son bastante conservadores porque no se pasan nunca del máximo o mínimo, si no que se encuentran en la media.

Métricas

```
model TBATS(seasonal_periods=(7, 365)) obtains MAPE: 14.60%
```

Ilustración 24: métricas TbatS diario

El MAPE del modelo es muy bueno, ya que falla un 14% pero como explicamos anteriormente este error, siempre se encuentra por debajo del máximo o por encima del mínimo, nunca se pasa.

Predicciones de enero

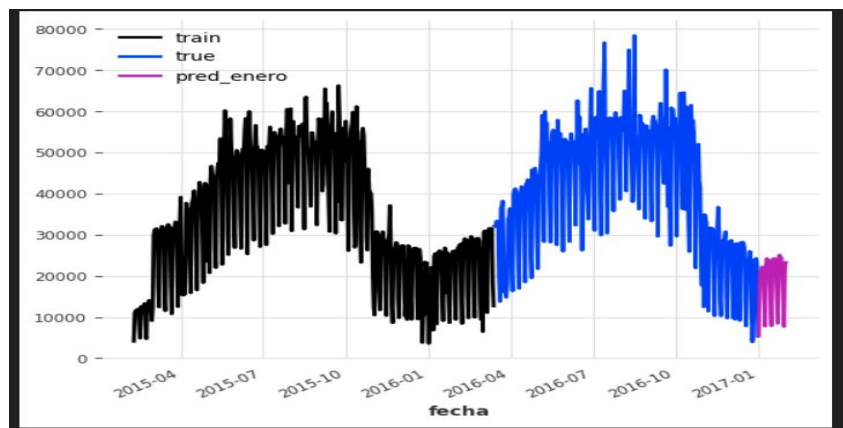


Ilustración 25: predicciones enero TBATS diario

Hemos elaborado las predicciones diarias con este modelo, ya que creemos que es un buen modelo, donde las predicciones siempre se encuentran dentro de los valores, sin tener fallos exageradamente altos

2. Predicción semanal

Parámetros

```
#Modelamos y entrenamos
model_semanal = TBATS(
    use_box_cox=None, #Si pones None, el modelo lo calcula
    use_trend=None,
    use_damped_trend=None,
    sp=52,
    use_arma_errors=True,
    n_jobs=1)
model_semanal.fit(y_train_semanal)
# TBATS(...)
```

Ilustración 26: Parámetros tbats semanal

use_box_cox: Controla si se aplica una transformación de Box-Cox a los datos.

use_trend: Controla si se incluye una componente de tendencia en el modelo.

use_damped_trend: Controla si se incluye una tendencia amortiguada en el modelo.

sp: Especifica la frecuencia estacional (en este caso, semanal, con un valor de 52).

use_arma_errors: Controla si se incluyen errores autoregresivos de media móvil (ARMA) en el modelo.

n_jobs: Controla el número de trabajadores paralelos utilizados para ajustar el modelo (1 en este caso).

Predicciones

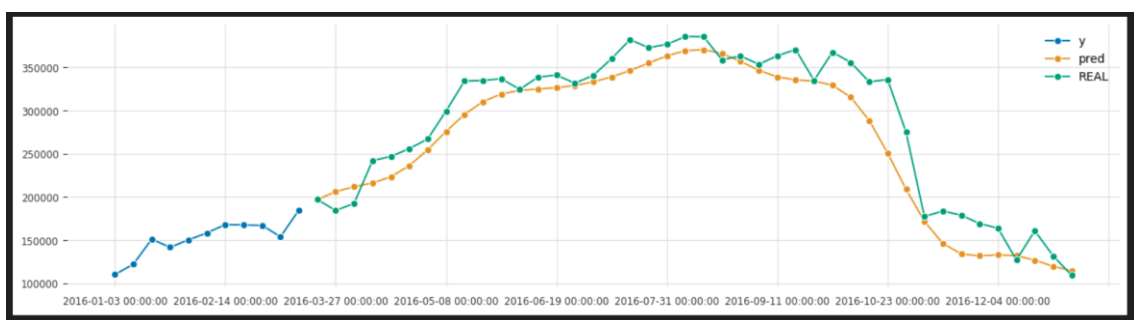


Ilustración 27: predicciones tbats semanal

Las tres líneas siguen patrones similares, lo que sugiere que el modelo que generó las predicciones ('pred') tuvo un desempeño razonablemente bueno al seguir la tendencia de los datos reales ('REAL'). Sin embargo, hay desviaciones entre 'pred' y 'REAL', lo que indica que, aunque las predicciones están en la dirección correcta, no son perfectas.

El pico de todas las series se encuentra alrededor de septiembre de 2016, después del cual hay un descenso notable. Este patrón sugiere un comportamiento estacional o un evento específico que causó el aumento seguido de una caída.

Para un análisis más detallado, sería necesario entender el contexto de los datos (como la naturaleza de la variable medida), la metodología de predicción, y considerar medidas de precisión como el error cuadrático medio o el coeficiente de determinación (R^2).

Métricas

```
MAPE: 0.08699405969294809
MSE: 823804828.7123388
R2: 0.8876922167496226
```

Ilustración 28: Métricas tbts semanal

Un valor de 0.086 un bajo porcentaje promedio de error, lo que sugiere que las predicciones son generalmente precisas en un 91.3%, Un valor alto como 8238048278 en el MSE indica que los errores individuales son en promedio grandes en magnitud. En el R^2 score nn valor de 0.8876922167496226 sugiere que el modelo es capaz de explicar aproximadamente el 88.8% de la variabilidad en los datos,

Resultados predicciones enero 2017

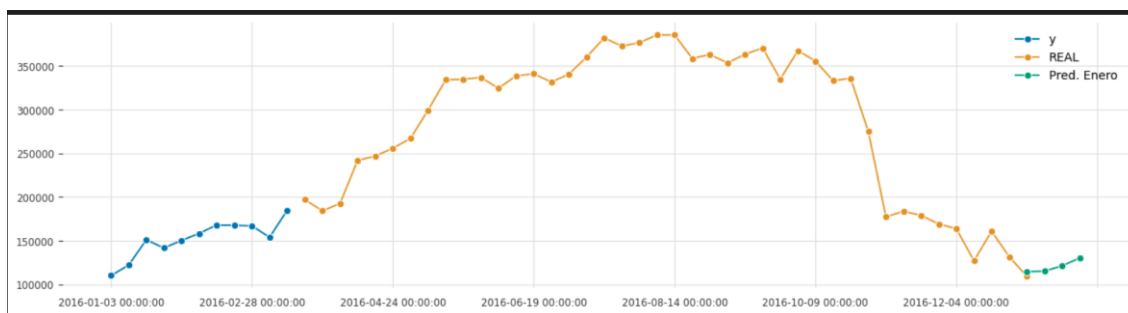


Ilustración 29: predicción enero tbats semanal

Se han realizado las predicciones para el mes de enero de 2017 con este dataset, por el mismo propósito que las predicciones diarias, se obtuvo unas buenas métricas y se estima que las predicciones pueden tener un gran valor a la hora de comparar con el modelo definitivo.

4. PROPHET (diaria)

El modelo Prophet es una herramienta de pronóstico desarrollada por Facebook para series temporales que destaca por su facilidad de uso y capacidad para manejar datos con tendencias y patrones estacionales. Es especialmente robusto frente a datos faltantes y cambios en la tendencia, y es capaz de incorporar efectos de días festivos. Utiliza un modelo aditivo donde las tendencias no lineales se ajustan con la estacionalidad anual, semanal y diaria, más los efectos de los días festivos. Es particularmente útil en contextos empresariales con series temporales que tienen patrones estacionales fuertes y varias temporadas de datos históricos.

El modelo prophet semanal ha sido escogido como el modelo definitivo por lo que ya ha sido comentado en el resumen ejecutivo, por lo que ahora comentaremos solo la predicción diaria

Parámetros

```
model = Prophet(  
    seasonality_mode='additive',  
    changepoint_prior_scale=0.1,  
    holidays_prior_scale=0.01,  
    changepoint_range=0.9,  
    yearly_seasonality=20  
)  
model.fit(train_diario)
```

Ilustración 30: Parámetros prophet diario

1. `seasonality_mode='additive'`: Establece que la estacionalidad del modelo es aditiva, lo cual es apropiado cuando las fluctuaciones estacionales no cambian proporcionalmente con el nivel de la serie temporal.

2. `changepoint_prior_scale=0.1`: Este parámetro controla la flexibilidad del modelo para ajustar las tendencias. Un valor más bajo hace que el modelo sea menos flexible, posiblemente evitando el sobreajuste.

3. `holidays_prior_scale=0.01`: Define la fuerza de la influencia de los días festivos en el modelo. Un valor más bajo significa que el modelo es menos flexible en el ajuste de los días festivos.

4. `changepoint_range=0.9`: Esto indica que el 90% de los datos se utilizarán para detectar los puntos de cambio. Los puntos de cambio son las fechas donde Prophet puede cambiar la tendencia de la serie temporal.

5. `yearly_seasonality=20`: Configura el modelo para tener 20 componentes estacionales para capturar la estacionalidad anual.

Predicciones

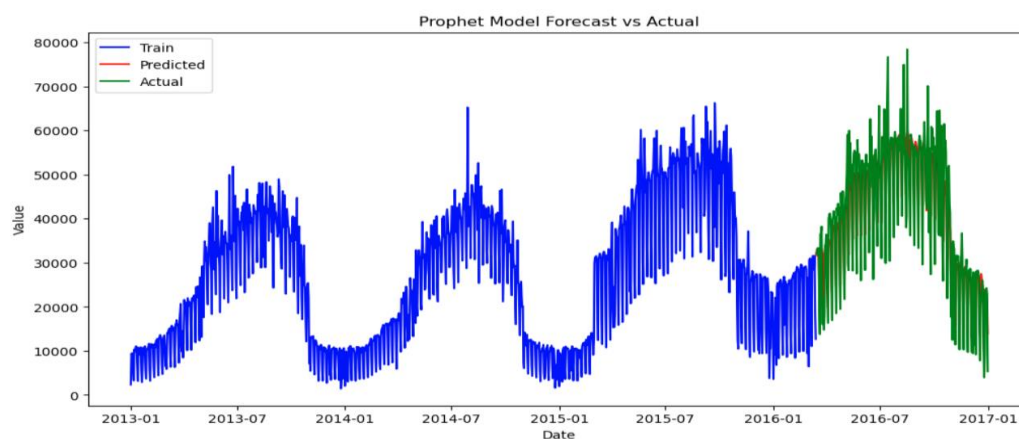


Ilustración 31: predicciones prophet diario

La gráfica muestra la comparación entre los datos reales ('Actual') y los valores predichos por un modelo Prophet ('Predicted'), junto con el conjunto de entrenamiento ('Train'). El modelo parece capturar la estacionalidad y la tendencia de los datos de entrenamiento con bastante precisión. Las predicciones siguen de cerca a los datos reales, reflejando los patrones estacionales evidentes en el conjunto de datos. Sin embargo, hay algunos picos, especialmente notorios en los datos reales, que el modelo no predice completamente. Esto puede deberse a eventos atípicos o a la falta de datos externos que podrían explicar estos picos. En general, el modelo proporciona una base sólida para pronosticar, aunque puede necesitar ajustes para manejar mejor las anomalías o incorporar información adicional que pueda afectar a la serie temporal.

Métricas

MAPE: 15.84%
MSE: 38547890.77
R2 score: 0.84

Ilustración 32: Parámetros prophet diario

Las métricas indican un rendimiento razonablemente bueno del modelo Prophet con un MAPE (Error Porcentual Absoluto Medio) del 15.84%, lo que sugiere que las predicciones tienen, en promedio, un 15.84% de error respecto a los valores reales. El MSE (Error Cuadrático Medio) es relativamente alto, lo que puede ser influenciado por los picos y variaciones extremas en los datos. Un R2 (coeficiente de determinación) de 0.84 implica que el modelo puede explicar el

84% de la variabilidad en los datos, lo que es bastante alto y señala una buena adecuación del modelo a los datos observados, como se refleja en la gráfica.

5. NEURALPROPHET (semanal)

Parámetros

```
model =model = NeuralProphet(  
    growth="linear", # Determine trend types: 'linear', 'discontinuous', 'off'  
    changepoints=None, # list of dates that may include change points (None->automatic)  
    n_changepoints=15,  
    changepoints_range=0.8,  
    trend_reg=0,  
    trend_reg_threshold=False,  
    yearly_seasonality="auto",  
    weekly_seasonality="auto",  
    daily_seasonality="auto",  
    seasonality_mode="additive",  
    seasonality_reg=0,  
    n_forecasts=1,  
    n_lags=0, # modificar  
    epochs=50, # 50  
    loss_func="Huber",  
    normalize="auto", # Type of normalization ('minmax', 'standardize', 'soft', 'off')  
    impute_missing=True  
)
```

Ilustración 33: Parámetros NeuralProphet semanal

1. **growth:** Este parámetro determina el tipo de tendencia que se aplicará al modelo. Puede ser "linear" (lineal), "discontinuous" (discontinua) u "off" (apagado). El tipo de tendencia define cómo se modela la evolución de la serie temporal a lo largo del tiempo.
2. **changepoints:** Una lista de fechas que pueden incluir puntos de cambio en la serie temporal. Si se establece en None, el modelo intentará detectar automáticamente los puntos de cambio.
3. **n_changepoints:** El número de puntos de cambio a utilizar si no se especifican explícitamente en changepoints.
4. **changepoints_range:** Controla la fracción de la serie temporal donde se permiten puntos de cambio. Un valor más alto permitirá más puntos de cambio en la serie temporal.
5. **trend_reg:** Parámetro de regularización aplicado a la tendencia. Puede ayudar a controlar la complejidad del modelo ajustando la tendencia.
6. **trend_reg_threshold:** Si es True, el parámetro trend_reg se aplica solo a los componentes de tendencia que superan un umbral. Si es False, se aplica a todos los componentes de tendencia.
7. **yearly_seasonality, weekly_seasonality y daily_seasonality:** Estos parámetros controlan la detección y modelado automático de estacionalidades anuales, semanales y diarias, respectivamente. Pueden ser "auto" para habilitar la

detección automática o un número entero para forzar el número de estacionalidades.

8. `seasonality_mode`: Define si las estacionalidades se modelan de forma "additive" (aditiva) o "multiplicative" (multiplicativa) con respecto a la tendencia.
9. `seasonality_reg`: Parámetro de regularización aplicado a las estacionalidades. Al igual que `trend_reg`, ayuda a controlar la complejidad del modelo.
10. `n_forecasts`: Número de pasos de pronóstico adelante en el tiempo.
11. `n_lags`: Número de pasos de retardo para utilizar como entrada en el modelo. Si se establece en 0, significa que no se utilizarán lags.
12. `epochs`: Número de épocas (iteraciones) durante el entrenamiento del modelo.
13. `loss_func`: Función de pérdida a utilizar durante el entrenamiento del modelo. En este caso, se utiliza la función de pérdida "Huber".
14. `normalize`: Tipo de normalización aplicada a la serie temporal antes del entrenamiento del modelo. Puede ser "auto", "minmax" (escala min-max), "standardize" (normalización estándar), "soft" o "off".
15. `impute_missing`: Si es True, el modelo intentará imputar valores faltantes en la serie temporal antes de entrenar.

Predicciones

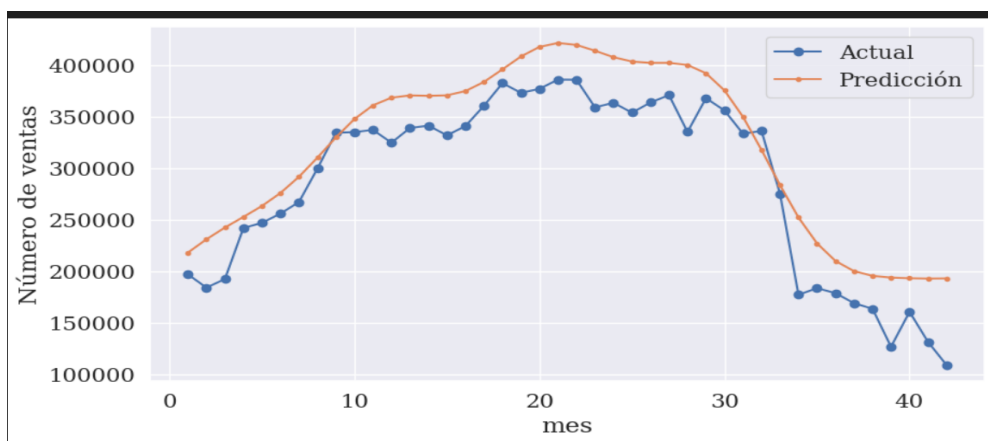


Ilustración 34: Predicciones NeuralProphet semanal

El gráfico muestra la comparación entre los valores de ventas reales y las predicciones a lo largo de aproximadamente 42 meses. Ambas líneas, la azul para los valores reales y la naranja para las predicciones, siguen una tendencia similar con las predicciones siguiendo de cerca a los valores reales, indicando

un modelo predictivo bastante acertado. Sin embargo, hay desviaciones notables, especialmente hacia el final donde las predicciones no capturan el aumento repentino y posteriormente el descenso en las ventas. Esto sugiere que el modelo podría no estar capturando bien ciertas tendencias o eventos inesperados.

Métricas

```
MAE: 33932.65 ventas  
MAPE: 0.1497%  
R2 Score: 0.80
```

Ilustración 35: Parámetros NeuralProphet semanal

1. MAE (Mean Absolute Error): El MAE de 33,932.65 ventas indica que, en promedio, las predicciones del modelo se desvían aproximadamente 33,932 unidades de las ventas reales. Este es un número absoluto que da una idea de la magnitud del error sin considerar la dirección (sobrestimación o subestimación).

2. MAPE (Mean Absolute Percentage Error): Un MAPE de 0.1497% es extremadamente bajo, lo que sugiere que el modelo es muy preciso en términos porcentuales. Es una forma de medir el error en términos relativos y es particularmente útil para comparar la precisión a través de diferentes escalas de datos.

3. R2 Score: Un puntaje R cuadrado de 0.80 indica que el modelo puede explicar el 80% de la variabilidad en los datos de ventas. Esto es generalmente considerado como una buena puntuación, significando que el modelo tiene un buen ajuste a los datos.

Combinando estas métricas, se puede concluir que el modelo tiene un rendimiento general sólido, con predicciones cercanas a los valores reales la mayoría del tiempo. Sin embargo, el MAE sugiere que puede haber algunos errores absolutos significativos.

6. CONCLUSIONES

Tras analizar el reporte desde la perspectiva de una empresa de autobuses, se identifican varias conclusiones clave:

Diversidad de Modelos para Predicciones: Se utilizó una variedad de modelos (ETS, ARIMA, Theta, 4Theta, TBATS, Prophet, NeuralProphet) para predecir la

demanda diaria y semanal de servicios de autobús. Esta diversidad metodológica es crucial para identificar el modelo más adecuado.

Importancia del Tratamiento de Datos: Se enfatiza el tratamiento de los datos como paso fundamental, incluyendo la unificación de fechas y la transformación de datos para predicciones diarias y semanales. Este proceso es vital para garantizar la precisión de las predicciones.

Desempeño Variable de los Modelos: Los modelos mostraron resultados mixtos. Algunos, como el ETS y ARIMA, tuvieron limitaciones en capturar tendencias o estacionalidades, mientras que otros como TBATS y NeuralProphet mostraron mejor capacidad predictiva, especialmente en las predicciones semanales.

Elección de Modelos según Métricas: La selección de los modelos definitivos se basó en métricas como el MAPE, MSE y R^2 . En general, se prefirieron modelos con bajos MAPE y R^2 altos, indicando predicciones precisas y un buen ajuste a los datos.

Prophet y NeuralProphet como Modelos Definitivos: El modelo Prophet fue seleccionado para la predicción semanal y NeuralProphet para la diaria, debido a su robustez y precisión, reflejada en las métricas y su capacidad para manejar datos con tendencias y patrones estacionales.

Relevancia para la Planificación Empresarial: Para una empresa de autobuses, estas predicciones son cruciales para la planificación de recursos, la gestión de flotas y la optimización de horarios. La capacidad de anticipar la demanda permite una operación más eficiente y una mejor experiencia para los usuarios.

En resumen, el reporte destaca la importancia de un análisis exhaustivo y la elección de modelos estadísticos adecuados para predecir la demanda en el sector del transporte. Esto no solo mejora la eficiencia operativa, sino que también contribuye a una mejor toma de decisiones estratégicas en la empresa.