Diplomado_DS_UNAM_proyecto_final_v2

October 7, 2023

Diplomado en Ciencia de datos UNAM

Proyecto Final

Alumno: Ibarra Ramírez Sergio

07 Octubre 2023

0.1 1. Título: Aplicación de modelos de series de tiempo y redes neurnales para el pronóstico de Demanda de gas natural en México

0.2 2.Resumen:

El presente trabajo explora la efectividad de los modelos de series de tiempo, específicamente ARIMA y SARIMA, y los modelos de redes neuronales, como LSTM, en el pronóstico de la demanda de gas natural en los sectores eléctrico y petrolero en México. El estudio aprovecha los datos de demanda mensual que abarcan de 12 a 15 años para predecir la demanda de los próximos 12 meses. El objetivo principal es determinar qué enfoque de modelado supera a los demás en términos de precisión de pronóstico, según lo medido por métricas como el Error Porcentual Absoluto Medio (MAPE) y el Error Cuadrático Medio (RMSE), así como criterios de información como el Criterio de Información de Akaike (AIC). Además, esta investigación investiga las fortalezas y debilidades de cada modelo, destacando su uso idoneo en función de las características únicas de la serie de tiempo de la demanda.

El sector energético, particularmente las industrias eléctrica y petrolera, juega un papel fundamental en la economía global. El pronóstico preciso de la demanda en estos sectores es crucial para la asignación eficiente de recursos, la gestión de costos y la planificación de infraestructura. Con la creciente importancia del gas natural como fuente de energía más limpia, los modelos de pronóstico robustos se convierten en herramientas esenciales para los tomadores de decisiones en estos sectores.

El estudio utiliza datos históricos extensos de demanda, que abarcan más de una década, para proporcionar una evaluación completa de los modelos de pronóstico seleccionados. Primero, se aplican al conjunto de datos modelos de series de tiempo tradicionales, ARIMA (AutoRegressive Integrated Moving Average) y SARIMA (Seasonal ARIMA). Estos modelos están bien establecidos y son ampliamente utilizados en el pronóstico de series de tiempo debido a su simplicidad e interpretabilidad. La investigación evalúa su desempeño frente a los modelos de redes neuronales.

Posteriormente, se emplean modelos de aprendizaje profundo, específicamente Redes Neuronales Recurrentes de Memoria a Largo Plazo (LSTM), para capturar dependencias temporales complejas dentro de los datos. Estos modelos han ganado popularidad por su capacidad para manejar datos

secuenciales de manera efectiva. La investigación evalúa su precisión de pronóstico y evalúa si su complejidad ofrece ventajas en el modelado de la demanda de gas natural sobre los enfoques tradicionales.

Los resultados de este estudio ayudarán a los profesionales de la industria a tomar decisiones mejor informadas sobre la asignación de recursos de gas natural y la planificación de infraestructura. Al identificar el modelo de pronóstico más preciso para cada sector, esta investigación contribuye a optimizar la utilización de recursos y las estrategias de reducción de costos. Además, proporciona información valiosa sobre las fortalezas y debilidades de cada enfoque de modelado, lo que permite una selección de modelo informada basada en las características específicas de los datos de la serie de tiempo de la demanda.

0.3 3. Introducción

Dada la relevancia del gas natural en México en los sectores productivos ha sido de interés pronosticar su demanda y dicho trabajo se ha desarrollado desde hace casi 30 años. En esta dirección, varios estudios han sido publicados, entre los que destacan los siguientes.

En 1995, ANAYA en su tesis de maestría propuso un modelo de ecuación lineal para predecir la demanda de gas natural en los años 90's y menciona "La tarea de hacer una demanda del gas natural en México es compleja en virtud de la gran cantidad de interrelaciones y fenómenos qué se circunscriben en el consumo de este energético".

Los estudios publicados por la Secretaría de Energía (SENER) anualmente desde 2002 hasta 2018 con el título "Prospectiva de Gas Natural en México", en los cuales se presentan estimaciones de la demanda de gas natural en México por sector usuario.

En 2016, Hinojosa en su tesis de maestría utiliza series de tiempo y redes neuronales para el pronóstico de la demanda de gas natural en México para el año de 2017

0.4 4. El problema (contexto)

En respuesta al panorama mundial cambiante y los cambios internos en México, se propone una actualización del enfoque con que anteriormente el gobierno mexicano ha pronosticado la demanda de gas natural. Esta recomendación surge del reconocimiento de que los métodos y marcos de tiempo de pronóstico actuales no están equipados adecuadamente para abordar las complejidades planteadas por los eventos globales recientes, los cambios de regímenes gubernamentales y la necesidad de una planificación energética precisa y prospectiva.

El contexto global ha experimentado transformaciones significativas en la última década, con eventos como la pandemia de COVID-19, la guerra de Ucrania y las crisis ambientales y alimentarias que ejercen un impacto profundo en los mercados energéticos. Estos factores, que tienen efectos dominó en las naciones, deben integrarse en el modelo de pronóstico de gas natural de México como variables cuantificables. Descuidarlos resultaría en predicciones incompletas y potencialmente poco confiables.

A nivel interno, México ha experimentado cambios sustanciales, incluido el advenimiento de un nuevo régimen gubernamental y la introducción de nuevas políticas públicas que han dado forma al panorama económico y social del país. Para adaptarse a estos cambios, el enfoque de pronóstico del gobierno debe adaptarse para reflejar estas realidades cambiantes. La suposición obsoleta de un entorno coherente e inmutable ya no es válida.

Se recomienda comparar varios métodos de pronóstico tanto para la demanda de gas natural, evaluando su precisión en varios escenarios. La selección del modelo más apropiado debe guiarse por la minimización de errores y el aseguramiento de la precisión, incluso hasta 1-2 años más allá del período de pronóstico inicial. La estrategia existente de pronosticar 15 años por delante en función de datos históricos que abarcan 10-15 años puede no ser ideal dada la dinámica actual de los factores globales y nacionales.

Para garantizar la relevancia y actualidad del estudio, proponemos un pronóstico enfocado para el período 2022-2023, basado en datos históricos de 2005 a 2021. Este período de tiempo más corto se alinea con la dinámica cambiante de la última década, ofreciendo una perspectiva más realista para los tomadores de decisiones en la planificación y política energética.

De manera crucial, el estudio debe describir los modelos de pronóstico empleados, acompañados de una presentación completa de los resultados. Estos modelos deben tener en cuenta las variables y supuestos actuales, ofreciendo una visión detallada y matizada de la demanda y producción de gas natural pronosticada en México. Lograr un margen de error inferior al 10% es esencial para mejorar la confiabilidad y la utilidad práctica del estudio.

Esta iniciativa de pronóstico actualizada proporcionará información valiosa para los tomadores de decisiones clave, incluidos la Secretaría de Energía (SENER) y la Comisión Nacional de Hidrocarburos (CNH), etc.

0.5 5. Propósito del estudio

Objetivos generales:

a) Evaluación del estado actual: Evaluar de manera integral el estado actual de la demanda de gas natural en México mediante la agregación y síntesis de datos e información relevantes.
b) Validación de pronósticos: Validar rigurosamente la precisión y confiabilidad de los pronósticos de demanda de gas natural en México, asegurando que los tomadores de decisiones puedan confiar en estas proyecciones para la planificación e implementación de políticas.
c) Apoyo a la transición energética: Ofrecer un apoyo analítico que permita tomar decisiones informadas sobre la transición de México hacia un futuro energético sostenible, considerando las dimensiones ambientales, económicas y sociales.

Objetivos específicos:

- Selección del modelo: Elegir uno o dos modelos adecuados tanto del análisis de series de tiempo como de las redes neuronales artificiales, en función de su capacidad para proporcionar pronósticos precisos en el panorama energético único de México.
- Análisis de resultados: Analizar rigurosamente los resultados generados por los modelos estadísticos elegidos bajo diversos escenarios, lo que permitirá una comprensión integral de los resultados potenciales.

0.6 6. Descripcion del proyecto

Este proyecto se embarca en un esfuerzo para mejorar la precisión y confiabilidad del pronóstico de la demanda de gas natural en México, con un enfoque principal en los sectores eléctrico y petrolero. Aprovechando la riqueza de datos disponibles de fuentes oficiales como la Secretaría de Energía (SENER) y el Instituto Nacional de Estadística y Geografía (INEGI), esta investigación busca

dotar a los tomadores de decisiones de información valiosa para la planificación energética efectiva y la formulación de políticas.

Recopilación de datos: El proyecto comenzará con la recopilación diligente de datos históricos de los sitios web oficiales de la SENER y el INEGI, asegurando la autenticidad y confiabilidad de la información. Este amplio conjunto de datos servirá de base para nuestros modelos de pronóstico.

Preparación y preprocesamiento de datos: Para garantizar la calidad y la idoneidad de los datos, se ejecutará una rigurosa fase de preprocesamiento. Esto implica limpiar los datos para eliminar inconsistencias, completar cualquier valor faltante y abordar los posibles valores atípicos. El objetivo es obtener un conjunto de datos limpio y robusto para su análisis.

Enfoque de modelado: El núcleo del proyecto radica en el despliegue de dos metodologías de pronóstico distintas pero poderosas: ARIMA (AutoRegressive Integrated Moving Average) y RNN (Recurrent Neural Network). ARIMA, un modelo de series de tiempo bien establecido, ofrece interpretabilidad y simplicidad. Por el contrario, RNN, un modelo de aprendizaje profundo, sobresale en capturar dependencias temporales complejas dentro de los datos. Ambos modelos se utilizarán para pronosticar la demanda de gas natural para los próximos 12 meses en los sectores eléctrico y petrolero.

Análisis comparativo: Un componente crítico de este proyecto implica realizar una comparación exhaustiva de los modelos ARIMA y LSTM. Esta comparación abarcará varias facetas, incluyendo la precisión del pronóstico, la robustez en el manejo de diferentes escenarios, la eficiencia computacional y la capacidad de adaptarse a datos en evolución. Los pros y los contras de cada modelo se evaluarán a fondo para determinar su idoneidad para tareas de pronóstico específicas.

0.7 7. Hipotesis

Este proyecto de investigación explora varias hipótesis clave que sustentan la exploración del pronóstico de la demanda de gas natural en México.

En primer lugar, se espera que los modelos de series de tiempo, en particular ARIMA y SARIMA, sobresalgan en la captura y el "aprendizaje" de patrones estacionales y tendencias a largo plazo en los datos. Sin embargo, es probable que estos modelos muestren sensibilidad a los valores atípicos o las tendencias locales inusuales, lo que podría provocar una reducción de la precisión en presencia de tales anomalías de datos.

En segundo lugar, se prevé que la utilización de redes neuronales recurrentes (RNN) y redes de memoria a largo plazo (LSTM) proporcionará un mecanismo más robusto para "aprender" patrones locales dentro de los datos. Estos modelos de aprendizaje profundo están diseñados para capturar dependencias temporales complejas, lo que los hace muy adecuados para manejar variaciones locales y desviaciones de la norma.

En tercer lugar, se hipotetiza que la investigación logrará un error porcentual absoluto medio (MAPE) dentro del rango del 5-8% al pronosticar la demanda de gas natural para los próximos 12-18 meses. Este rango refleja la aspiración a un alto nivel de precisión en el pronóstico, asegurando que los modelos produzcan predicciones confiables.

En cuarto lugar, se espera que tanto los modelos de series de tiempo como los modelos RNN/LSTM estén influenciados por los valores atípicos observados en 2020 y 2021, atribuidos principalmente a la pandemia de COVID-19. Es probable que estas circunstancias excepcionales desafíen la adaptabilidad de los modelos y puedan resultar en inexactitudes de pronóstico durante estos períodos

específicos.

Por último, con respecto a la inclusión de variables externas como el Producto Interno Bruto (PIB), la dinámica del tipo de cambio y los tipos de cambio monetarios, se hipotetiza que estas variables pueden introducir más ruido que información adicional útil en el proceso de pronóstico. Dado esto, un enfoque potencialmente superior puede implicar utilizar solo valores de demanda pasados como variables independientes, asegurando que los modelos permanezcan enfocados y no perturbados por factores externos.

0.8 8. Flujo de trabajo y 9. Mapeo del sistema

Se presenta a continuación el flujo de trabajo general que se seguirá así como los resultados esperados que consta básicamente de 3 etapas: 1. Recolección y limpieza de datos 2. Desarrollo y evaluación de modelos 3. Evaluación de modelos y conclusiones generales

[5]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_8_Flujo_de_trabajo.png',␣
↪height=500, width=700)
```

[5]:

0.9 10. Definición de métricas adecuadas

En el contexto del pronóstico de la demanda de gas natural utilizando series de tiempo y modelos de redes neuronales recurrentes (RNN), la selección y comprensión de métricas de evaluación precisas es fundamental. En primer lugar, empleamos dos métricas de error cruciales: el error porcentual absoluto medio (MAPE) y el error cuadrático medio (RMSE).

MAPE: -Ventaja(s) Es muy útil cuando se quieren evaluar dos modelos en términos de porcentaje promedio /medio de error. -Desventaja(s): Tiende a dispararse si la media de los datos tiende a cero.

RMSE: -Ventaja(s) Esta en las mismas unidades que la variable estudiada, lo que permite tener una clara idea de cuanto puede costar el error promedio en los mismos términos de la variable de interes. -Desventaja(s): Tiende a castigar mucho los valores atípicos.

Además, utilizamos el criterio de información de Akaike (AIC) en nuestro marco de evaluación. El AIC combina la verosimilitud del modelo con el número de parámetros utilizados, priorizando los modelos que ofrecen un equilibrio entre el ajuste a los datos y la simplicidad. El principio de parsimonia subraya este enfoque, abogando por modelos que logren una precisión comparable empleando menos parámetros. Esto es particularmente relevante en el contexto de nuestra investigación, ya que ayuda a seleccionar modelos que alcanzan un equilibrio óptimo entre complejidad y eficacia de pronóstico.

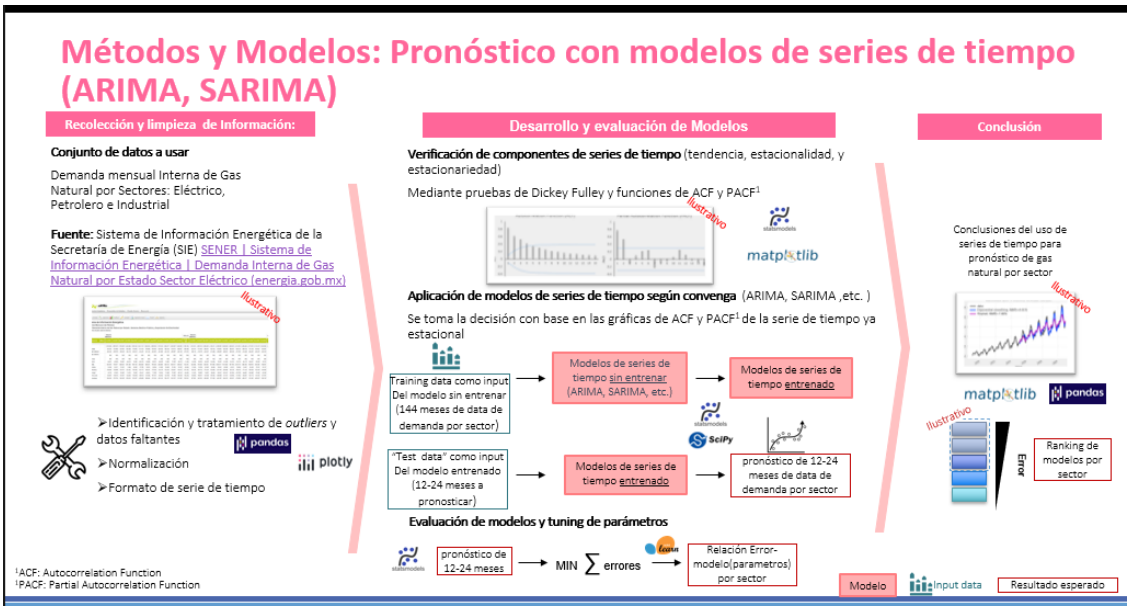
Al definir y utilizar meticulosamente estas métricas, nuestra investigación tiene como objetivo proporcionar un marco robusto para la evaluación de modelos de series de tiempo y LSTM, avanzando en última instancia la precisión y confiabilidad de los pronósticos de demanda de gas natural.

0.10 11. Métodos y modelos

- Para el caso de los modelos de series de tiempo

[1]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_11_Metodos_modelos_series_tiempo.
↪png', height=500, width=700)
```

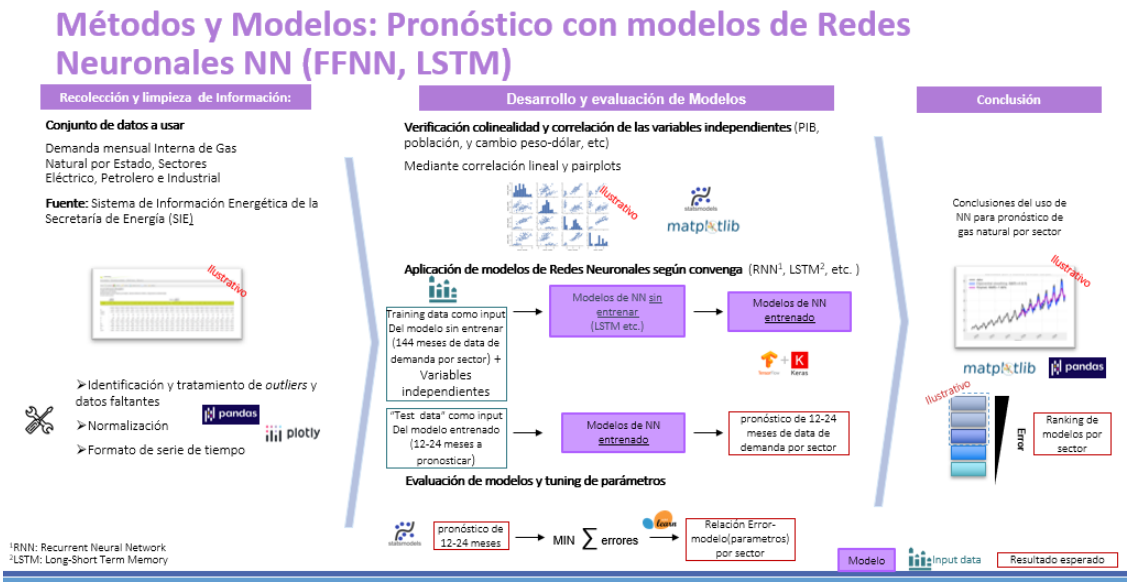
[1]:



- Para el caso de los modelos de Redes Neuronales LSTM

[2]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_11_Metodos_modelos_LSTM.png',␣
↪height=500, width=700)
```

[2]:



0.11 12. Evaluación de modelos

Se presenta un ejemplo de evaluación de modelo de serie de tiempo (todos los casos se encuentran en archivos ipynb identificados por nombre en la carpeta compartida)

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import register_matplotlib_converters
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.arima.model import ARIMA
from sympy import true
from datetime import datetime, timedelta
register_matplotlib_converters()
from time import time
```

Se lee la data "original" de Demanda en sector eléctrico

[2]:
```
csv_demanda_electrico_original = pd.read_csv('Demanda_electrico_2022_full1.
  ↪csv',index_col='Date',parse_dates=True)
csv_demanda_electrico_original
```

[2]:

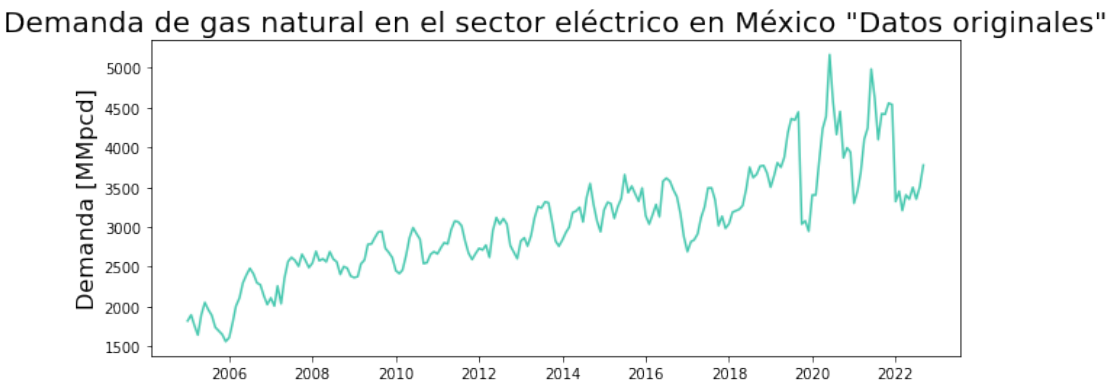
Date	Demanded_Gas
2005-01-01	1819.58
2005-02-01	1895.33
2005-03-01	1765.86

```
2005-04-01        1642.70
2005-05-01        1895.54
…                   …
2022-05-01        3350.03
2022-06-01        3498.70
2022-07-01        3350.97
2022-08-01        3506.42
2022-09-01        3778.37

[213 rows x 1 columns]
```

Se grafica la data “original” de Demanda en sector eléctrico

[3]:
```
plt.figure(figsize=(10,4))
plt.plot(csv_demanda_electrico_original, color='#48C9B0')
plt.title('Demanda de gas natural en el sector eléctrico en México "Datos␣
 ↪originales"', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize =16)
```

[3]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



Se lleva a cabo la prueba de estacionariedad de Dickey-Fulley a la data “original” de Demanda en sector eléctrico

[6]:
```
import pandas as pd
from statsmodels.tsa.stattools import adfuller

# Perform ADF test for stationarity
adf_test_electrico_original_sin_diferenciar =␣
 ↪adfuller(csv_demanda_electrico_original)


adf_test_electrico_original_sin_diferenciar
```

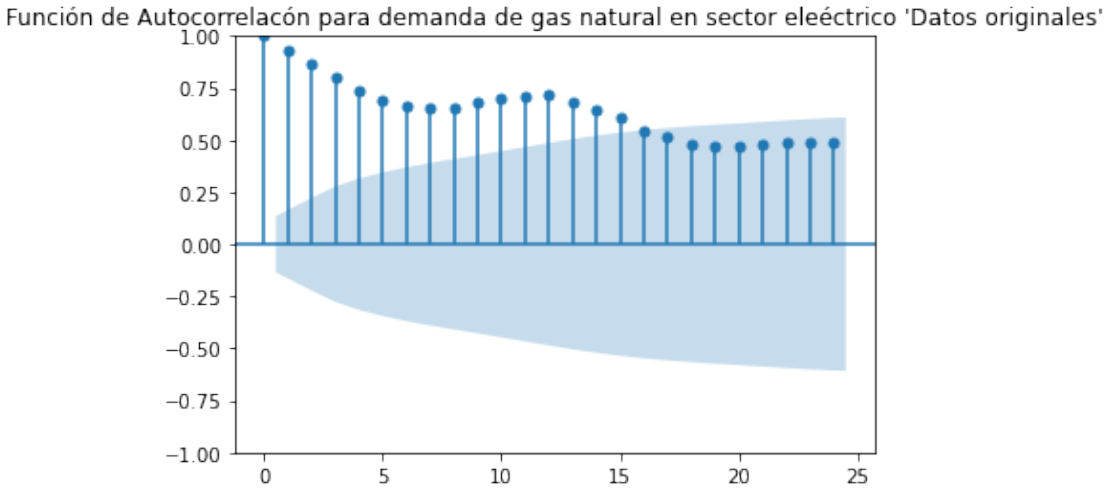
[6]:
```
(-1.9362234591018295,
 0.3152169397511435,
 15,
 197,
 {'1%': -3.463987334463603,
  '5%': -2.8763259091636213,
  '10%': -2.5746515171738515},
 2667.963876967698)
```

[8]:
```
print(f"The ADF statistic value f is:␣
 ↪{adf_test_electrico_original_sin_diferenciar[0]}")

print(f"The ADF p value p is: {adf_test_electrico_original_sin_diferenciar[1]}")

if adf_test_electrico_original_sin_diferenciar[0] <␣
 ↪adf_test_electrico_original_sin_diferenciar[4]['5%']:
    print("Se rechaza H0: SI existe suficiente evidencia para rechazar H0, por␣
 ↪lo tanto SI existe estacionariedad")
else:
    print("Se acepta H0: NO existe suficiente evidencia para rechazar H0, por␣
 ↪lo tanto NO existe estacionariedad")
```

```
The ADF statistic value f is: -1.9362234591018295
The ADF p value p is: 0.3152169397511435
Se acepta H0: NO existe suficiente evidencia para rechazar H0, por lo tanto NO
existe estacionariedad
```

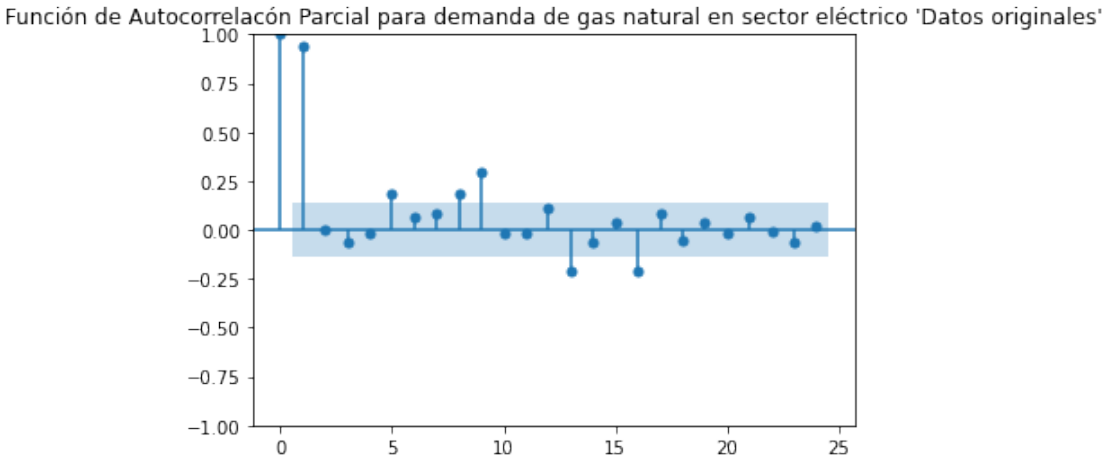
Se elaboran las gráficas de ACF y PACF de la data “original” de Demanda en sector eléctrico

[4]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()

# Plot the ACF
tsaplot.plot_acf(csv_demanda_electrico_original.dropna(), ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón para demanda de gas natural en sector␣
 ↪eleéctrico 'Datos originales'")

# Show the plot
plt.show()
```



[5]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()

# Plot the ACF
tsaplot.plot_pacf(csv_demanda_electrico_original.dropna(), ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón Parcial para demanda de gas natural en␣
 ↪sector eléctrico 'Datos originales'")

# Show the plot
plt.show()
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\graphics\tsaplots.py:348: FutureWarning: The default method
'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the
default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
```



Se procede entonces a diferenciar la serie de data "original" de Demanda en sector eléctrico para lograr estacionariedad

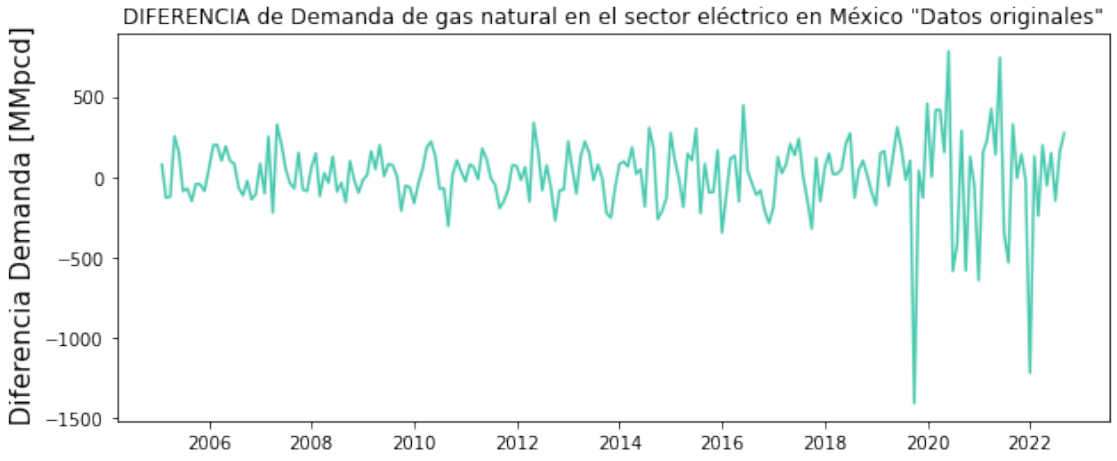
[9]:
```
demanda_electrico_original_diff1 = csv_demanda_electrico_original.diff()
demanda_electrico_original_diff1
```

[9]:
```
            Demanded_Gas
Date
2005-01-01           NaN
2005-02-01         75.75
2005-03-01       -129.47
2005-04-01       -123.16
2005-05-01        252.84
…                    …
2022-05-01        -53.41
2022-06-01        148.67
2022-07-01       -147.73
2022-08-01        155.45
2022-09-01        271.95

[213 rows x 1 columns]
```

Se grafica la diferencia de la la serie de data "original" de Demanda en sector eléctrico

[10]:
```
plt.figure(figsize=(10,4))
plt.plot(demanda_electrico_original_diff1, color='#48C9B0')
plt.title('DIFERENCIA de Demanda de gas natural en el sector eléctrico en␣
 ↪México "Datos originales"')
plt.ylabel(' Diferencia Demanda [MMpcd]', fontsize =15)
```

[10]:
```
Text(0, 0.5, ' Diferencia Demanda [MMpcd]')
```



Se lleva a cabo la prueba de estacionariedad de Dickey-Fulley a la DIFERENCIA1 Demanda de gas natural en el sector eléctrico en México “Datos originales”

[11]:
```
import pandas as pd
from statsmodels.tsa.stattools import adfuller

# Check for infinite or NaN values
demanda_electrico_original_diff1.dropna(inplace=True)

# Perform ADF test
adf_test_electrico_original_diferencia1 =␣
 ↪adfuller(demanda_electrico_original_diff1)
```

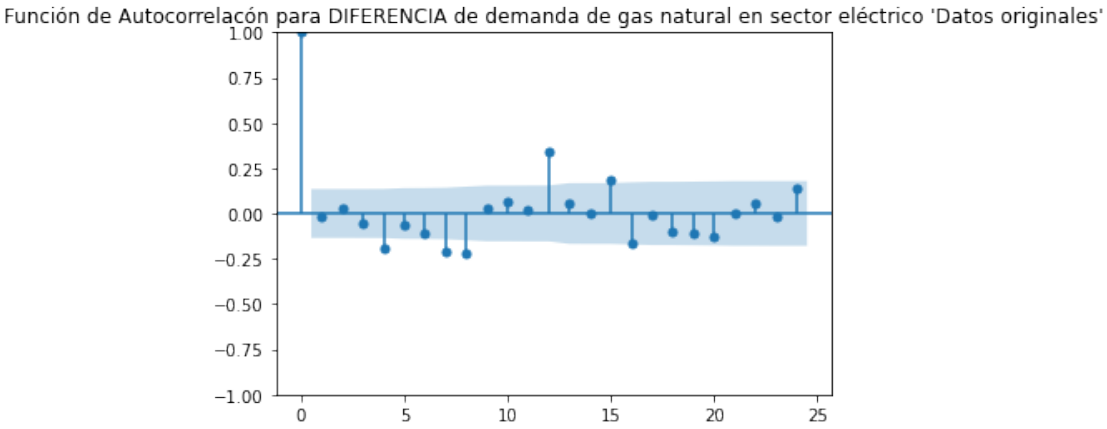
[12]:
```
print(f"The ADF statistic value f is:␣
 ↪{adf_test_electrico_original_diferencia1[0]}")

print(f"The ADF p value p is: {adf_test_electrico_original_diferencia1[1]}")

if adf_test_electrico_original_diferencia1[0] <␣
 ↪adf_test_electrico_original_diferencia1[4]['5%']:
    print("Se rechaza H0: SI existe suficiente evidencia para rechazar H0, por␣
 ↪lo tanto SI existe estacionariedad")
else:
    print("Se acepta H0: NO existe suficiente evidencia para rechazar H0, por␣
 ↪lo tanto NO existe estacionariedad")
```

```
The ADF statistic value f is: -4.063276407512036
The ADF p value p is: 0.0011131147894365412
Se rechaza H0: SI existe suficiente evidencia para rechazar H0, por lo tanto SI
existe estacionariedad
```

Se elaboran las gráficas de ACF y PACF de la DIFERENCIA de Demanda de gas natural en el

sector eléctrico en México

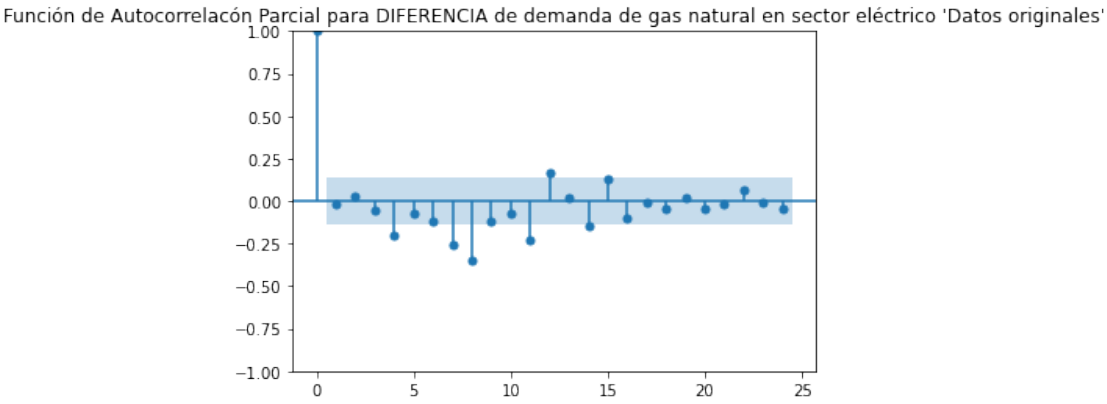
[13]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()

# Plot the ACF
tsaplot.plot_acf(demanda_electrico_original_diff1.dropna(), ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón para DIFERENCIA de demanda de gas␣
 ↪natural en sector eléctrico 'Datos originales'")

# Show the plot
plt.show()
```



[14]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()

# Plot the ACF
tsaplot.plot_pacf(demanda_electrico_original_diff1.dropna(), ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón Parcial para DIFERENCIA de demanda de␣
 ↪gas natural en sector eléctrico 'Datos originales'")

# Show the plot
plt.show()
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\graphics\tsaplots.py:348: FutureWarning: The default method
```

```
'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the
default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
```

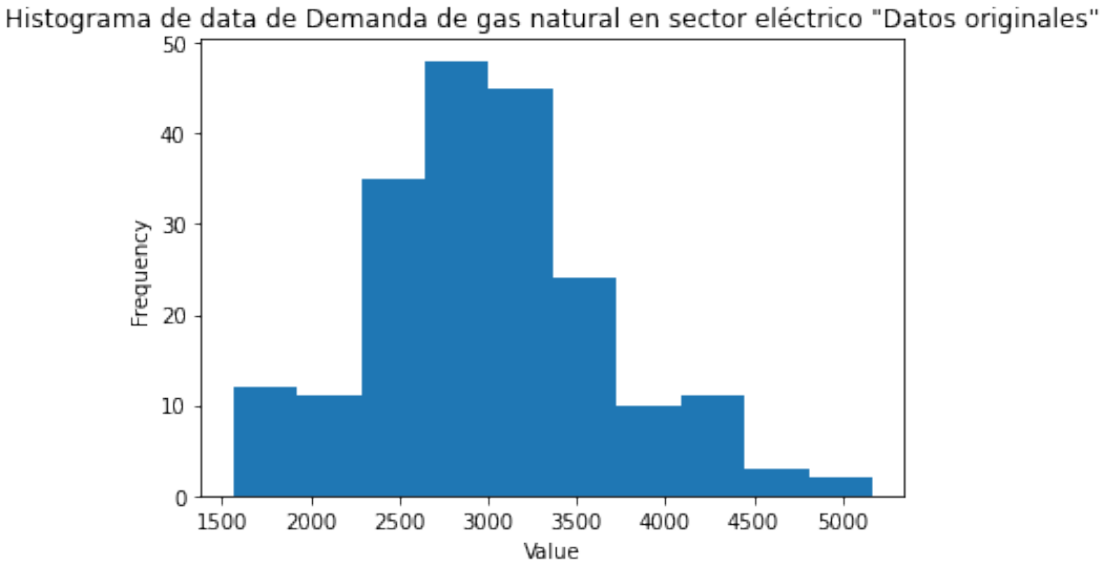


Separamos la data original de Demanda de gas natural en el sector eléctrico en data de train y test

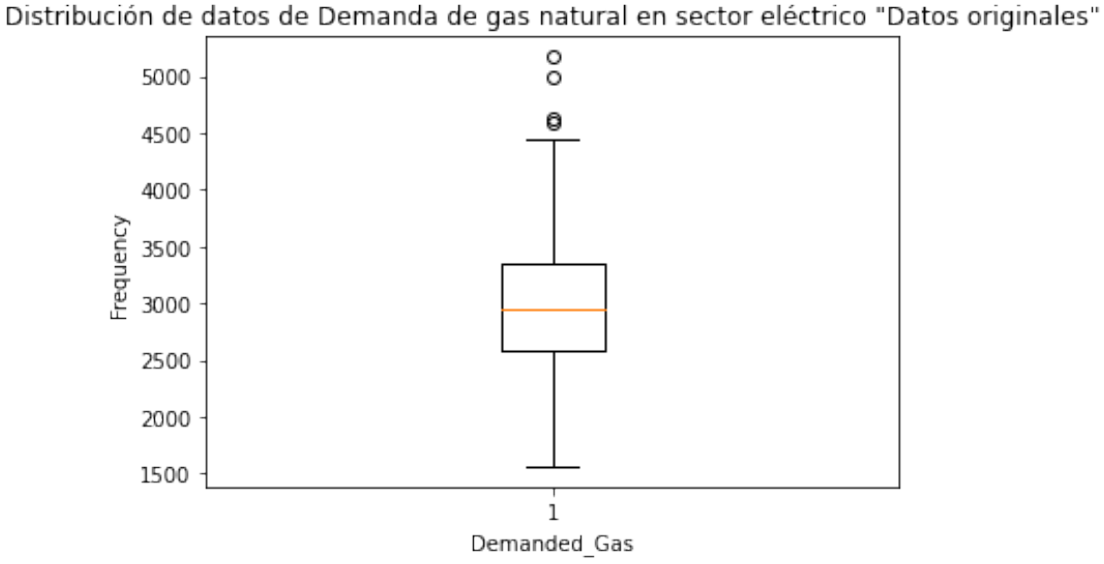
[15]:
```
# Number of data points to keep for testing (in this case, the last 12)
num_test_points = 12

# Split the data into training and testing sets
demanda_electrico_original_train_data = csv_demanda_electrico_original[:
 ↪-num_test_points]
demanda_electrico_original_test_data =␣
 ↪csv_demanda_electrico_original[-num_test_points:]
```

Se grafica la distribución de la data “original” de Demanda en sector eléctrico

[16]:
```
# Generate the histogram
plt.hist(demanda_electrico_original_train_data, bins=10)  # Adjust the number␣
 ↪of bins as per your data

# Add labels and title
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histograma de data de Demanda de gas natural en sector eléctrico␣
 ↪"Datos originales"')

# Display the plot
plt.show()
```



[17]:
```
# Create a boxplot of the Demanded_Gas column
plt.boxplot(demanda_electrico_original_train_data)

# Add labels and title
plt.xlabel('Demanded_Gas')
plt.ylabel('Frequency')
plt.title('Distribución de datos de Demanda de gas natural en sector eléctrico␣
 ↪"Datos originales"')

# Display the plot
plt.show()
```



0.11.1 Se define y entrena modelo ARIMA para el caso de la data original de Demanda en el sector eléctrico

[18]:

```
##Create the model
model_ARIMA_electrico_original = ARIMA (demanda_electrico_original_train_data,␣
 ↪order=(4,1,4))

##Fit the model
start = time()
model_ARIMA_electrico_original_fit = model_ARIMA_electrico_original.fit()
end = time()
print('Model fitting time', end-start)

##Summary of the model
print(model_ARIMA_electrico_original_fit.summary())
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\base\model.py:604: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
Model fitting time 1.5495128631591797
                               SARIMAX Results
==============================================================================
Dep. Variable:           Demanded_Gas   No. Observations:                  201
Model:                 ARIMA(4, 1, 4)   Log Likelihood               -1333.392
Date:                Fri, 06 Oct 2023   AIC                           2684.783
Time:                        14:14:17   BIC                           2714.468
Sample:                    01-01-2005   HQIC                          2696.796
                         - 09-01-2021
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
```

```
ar.L1          0.8308      0.074     11.229      0.000       0.686       0.976
ar.L2         -0.4263      0.067     -6.332      0.000      -0.558      -0.294
ar.L3          0.8686      0.086     10.046      0.000       0.699       1.038
ar.L4         -0.8465      0.054    -15.783      0.000      -0.952      -0.741
ma.L1         -0.9958      0.087    -11.435      0.000      -1.166      -0.825
ma.L2          0.4966      0.072      6.891      0.000       0.355       0.638
ma.L3         -1.0714      0.094    -11.400      0.000      -1.256      -0.887
ma.L4          0.7984      0.102      7.846      0.000       0.599       0.998
sigma2       4.11e+04   4100.493     10.022      0.000    3.31e+04    4.91e+04
===================================================================================
===
Ljung-Box (L1) (Q):                   0.42   Jarque-Bera (JB):
513.46
Prob(Q):                              0.52   Prob(JB):
0.00
Heteroskedasticity (H):               4.57   Skew:
-1.07
Prob(H) (two-sided):                  0.00   Kurtosis:
10.55
===================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

Hagamos predicciones con el modelo ARIMA para la Demanda de gas natural en sector eleéctrico 'Datos originales'

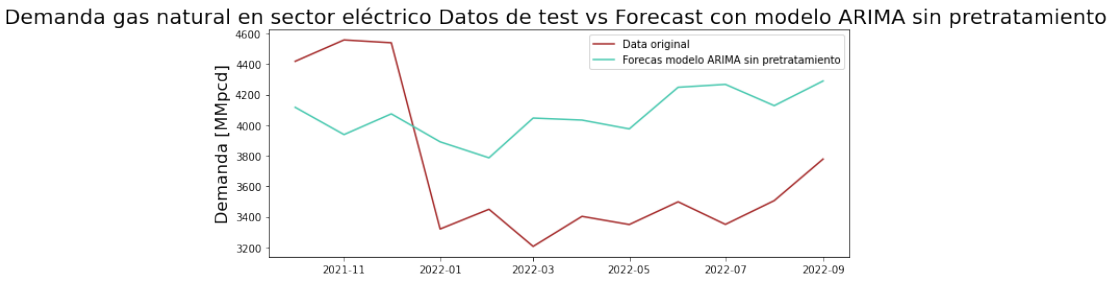
[19]:
```
##get prediction start and end dates
pred_model_ARIMA_electrico_start_date = demanda_electrico_original_test_data.
 ↪index[0]
pred_model_ARIMA_electrico_end_date = demanda_electrico_original_test_data.
 ↪index[-1]

##get the predictors and residuals
predictions_model_ARIMA_electrico_original = model_ARIMA_electrico_original_fit.
 ↪predict(start=pred_model_ARIMA_electrico_start_date, end=␣
 ↪pred_model_ARIMA_electrico_end_date)
print(predictions_model_ARIMA_electrico_original)
```

```
2021-10-01    4116.760353
2021-11-01    3938.306019
2021-12-01    4073.956534
2022-01-01    3890.889753
2022-02-01    3786.363557
2022-03-01    4046.456309
2022-04-01    4033.246030
```

```
2022-05-01    3975.558960
2022-06-01    4247.672609
2022-07-01    4266.689838
2022-08-01    4127.559924
2022-09-01    4289.069773
Freq: MS, Name: predicted_mean, dtype: float64
```

[20]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data, color='brown')
plt.plot(predictions_model_ARIMA_electrico_original, color='#48C9B0')
plt.legend(('Data original', 'Forecas modelo ARIMA sin pretratamiento'),␣
 ↪fontsize=10)

plt.title('Demanda gas natural en sector eléctrico Datos de test vs Forecast␣
 ↪con modelo ARIMA sin pretratamiento ', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[20]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



Cuantificando el error de las predicciones con el modelo ARIMA para la Demanda de gas natural en sector eleéctrico 'Datos originales'

[21]:
```
from sklearn.metrics import mean_absolute_percentage_error, mean_squared_error
import numpy as np

# Compute errors
MAPE_predictions_model_ARIMA_electrico_original =␣
 ↪mean_absolute_percentage_error(demanda_electrico_original_test_data,␣
 ↪predictions_model_ARIMA_electrico_original)
RMSE_predictions_model_ARIMA_electrico_original = np.
 ↪sqrt(mean_squared_error(demanda_electrico_original_test_data,␣
 ↪predictions_model_ARIMA_electrico_original))

print('MAPE:', MAPE_predictions_model_ARIMA_electrico_original)
print('RMSE:', RMSE_predictions_model_ARIMA_electrico_original)
```

```
MAPE: 0.16739974297359148
RMSE: 623.6320827738078
```

0.11.2 Proceso de Pretratamiento (Ajuste de valores atípicos por media aritmetica y/o mediana de serie) como una estrategia para mejorar los resultados de forecast de modelos ARIMA

0.11.3 Proceso de Pretratamiento Ajuste de valores atípicos por media aritmetica

[22]:
```
outlier_threshold = 1.2

import pandas as pd

def remove_replace_outliers_media(data):
    # Calculate Q1 and Q3
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1

    # Identify outliers
    outliers_mask = (data < (Q1 - outlier_threshold * IQR)) | (data > (Q3 +␣
 ↪outlier_threshold * IQR))

    # Calculate the historical mean excluding outliers
    historical_mean = data[~outliers_mask].mean()

    # Replace outliers with historical mean
    data.loc[outliers_mask] = historical_mean

    return data
```

Se genera una copia del demanda_electrico_original para conservar los datos originales en un array y éstos no se vean afectados por el tratamiento de valores atípicos

[23]:
```
# Create a copy of the original DataFrame
demanda_electrico_original_train_data_para_tratamiento =␣
 ↪demanda_electrico_original_train_data.copy()
demanda_electrico_original_train_data_para_tratamiento
```

[23]:

Date	Demanded_Gas
2005-01-01	1819.58
2005-02-01	1895.33
2005-03-01	1765.86
2005-04-01	1642.70
2005-05-01	1895.54
…	…
2021-05-01	4243.93

```
2021-06-01       4985.53
2021-07-01       4631.85
2021-08-01       4098.81
2021-09-01       4424.39

[201 rows x 1 columns]
```

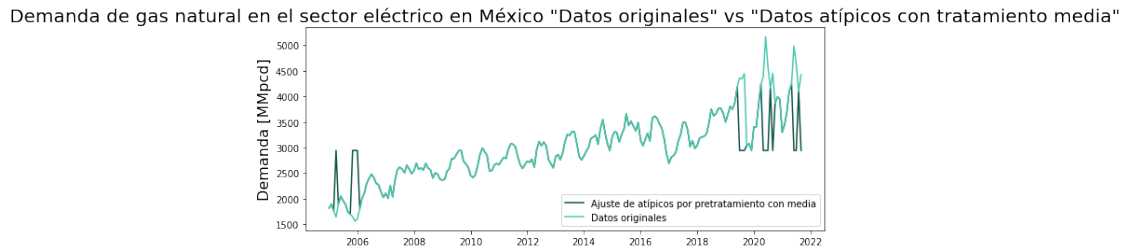
[24]:
```
demanda_electrico_train_pretratamiento_media =␣
 ↪remove_replace_outliers_media(demanda_electrico_original_train_data_para_tratamiento['Deman
demanda_electrico_train_pretratamiento_media.tail(10)
```

[24]:
```
Date
2020-12-01    3941.120000
2021-01-01    3298.430000
2021-02-01    3454.210000
2021-03-01    3681.300000
2021-04-01    4104.820000
2021-05-01    4243.930000
2021-06-01    2944.216952
2021-07-01    2944.216952
2021-08-01    4098.810000
2021-09-01    2944.216952
Name: Demanded_Gas, dtype: float64
```

Se grafica de los datos de entrenamiento de la Demanda de Gas Natural en sector eléctrico sin valores atípicos (tras haber aplicado la función remove_replace_outliers_media)

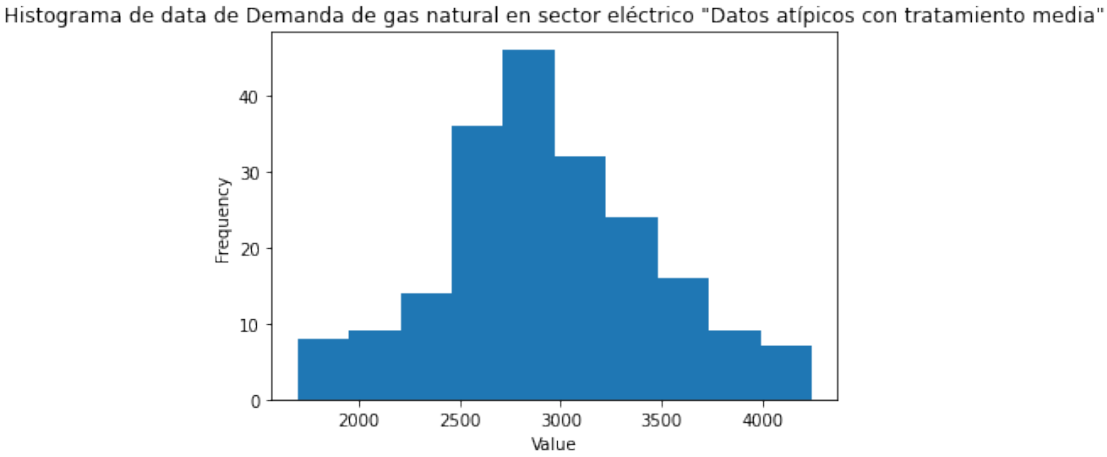
[25]:
```
plt.figure(figsize=(10,4))
plt.plot(demanda_electrico_train_pretratamiento_media, color='#0B5345')
plt.plot(demanda_electrico_original_train_data, color='#48C9B0')
plt.title('Demanda de gas natural en el sector eléctrico en México "Datos␣
 ↪originales" vs "Datos atípicos con tratamiento media"', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize =16)
# Add a legend to the left bottom corner
plt.legend(['Ajuste de atípicos por pretratamiento con media', 'Datos␣
 ↪originales'], loc='lower right', fontsize=10)
```

[25]:
```
<matplotlib.legend.Legend at 0x211b7e75990>
```



Se grafica la distribución de Demanda de gas natural en el sector eléctrico “Datos atípicos con tratamiento media”

[26]:
```
# Generate the histogram
plt.hist(demanda_electrico_train_pretratamiento_media, bins=10)  # Adjust the␣
 ↪number of bins as per your data

# Add labels and title
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histograma de data de Demanda de gas natural en sector eléctrico␣
 ↪"Datos atípicos con tratamiento media"')

# Display the plot
plt.show()
```



Se lleva a cabo la prueba Dickey Fulley para la serie de demanda electrico train data datos atípicos con tratamiento media

[27]:
```
import pandas as pd
from statsmodels.tsa.stattools import adfuller

# Perform ADF test for stationarity
adf_test_demanda_electrico_train_pretratamiento_media =␣
 ↪adfuller(demanda_electrico_train_pretratamiento_media)


adf_test_demanda_electrico_train_pretratamiento_media
```

[27]:
```
(-1.1674638525703216,
 0.687433065014806,
```

```
 11,
 189,
 {'1%': -3.4654311561944873,
  '5%': -2.8769570530458792,
  '10%': -2.574988319755886},
 2587.340568073859)
```

[28]:
```
print(f"The ADF statistic value f is:␣
 ↪{adf_test_demanda_electrico_train_pretratamiento_media[0]}")

print(f"The ADF p value p is:␣
 ↪{adf_test_demanda_electrico_train_pretratamiento_media[1]}")

if adf_test_electrico_original_diferencia1[0] <␣
 ↪adf_test_electrico_original_diferencia1[4]['5%']:
    print("Se rechaza H0: SI existe suficiente evidencia para rechazar H0, por␣
 ↪lo tanto SI existe estacionariedad")
else:
    print("Se acepta H0: NO existe suficiente evidencia para rechazar H0, por␣
 ↪lo tanto NO existe estacionariedad")
```

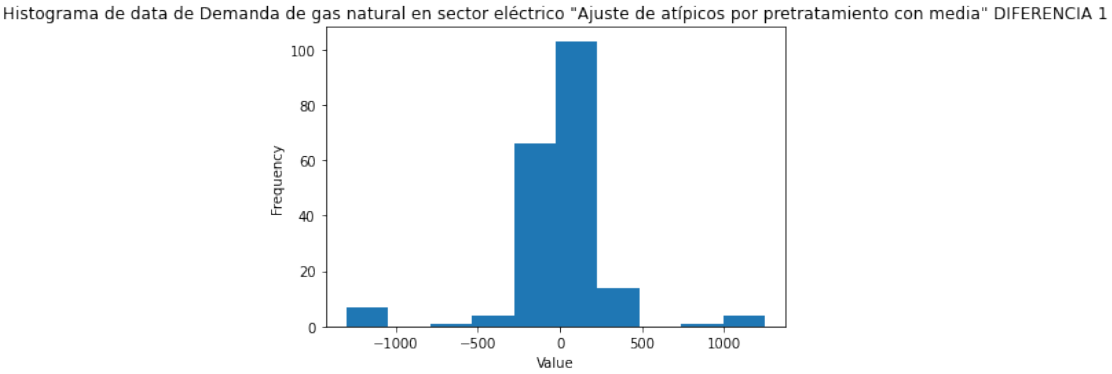
```
The ADF statistic value f is: -1.1674638525703216
The ADF p value p is: 0.687433065014806
Se rechaza H0: SI existe suficiente evidencia para rechazar H0, por lo tanto SI
existe estacionariedad
```

También para este caso se muestran las gráficas de ACF y PACF de la DIFERENCIA de Demanda de gas natural en el sector eléctrico en México

[29]:
```
demanda_electrico_train_tratamiento_media_diff1 =␣
 ↪demanda_electrico_train_pretratamiento_media.diff()
demanda_electrico_train_tratamiento_media_diff1
```

[29]:
```
Date
2005-01-01            NaN
2005-02-01      75.750000
2005-03-01    -129.470000
2005-04-01    1178.356952
2005-05-01   -1048.676952
                  …
2021-05-01     139.110000
2021-06-01   -1299.713048
2021-07-01       0.000000
2021-08-01    1154.593048
2021-09-01   -1154.593048
Name: Demanded_Gas, Length: 201, dtype: float64
```

[30]:
```
# Generate the histogram
plt.hist(demanda_electrico_train_tratamiento_media_diff1.dropna(), bins=10)  #␣
 ↪Adjust the number of bins as per your data

# Add labels and title
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histograma de data de Demanda de gas natural en sector eléctrico␣
 ↪"Ajuste de atípicos por pretratamiento con media" DIFERENCIA 1')

# Display the plot
plt.show()
```



Se lleva a cabo la prueba Dickey Fulley para la serie de demanda electrico train data con pretratamiento de media Diferencia 1

[31]:
```
import pandas as pd
from statsmodels.tsa.stattools import adfuller

# Perform ADF test for stationarity
adf_test_demanda_electrico_train_tratamiento_media_diff1 =␣
 ↪adfuller(demanda_electrico_train_tratamiento_media_diff1.dropna())


adf_test_demanda_electrico_train_tratamiento_media_diff1
```

[31]:
```
(-6.518615852472467,
 1.055865744285777e-08,
 11,
 188,
 {'1%': -3.465620397124192,
  '5%': -2.8770397560752436,
  '10%': -2.5750324547306476},
```

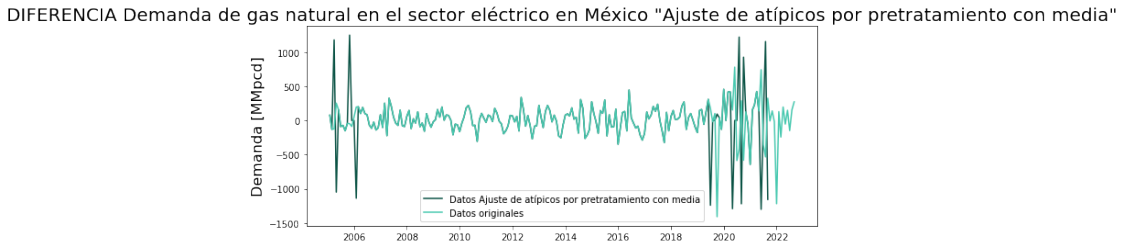
2573.4576689312335)

[32]:
```
print(f"The ADF statistic value f is:␣
 ↪{adf_test_demanda_electrico_train_tratamiento_media_diff1[0]}")

print(f"The ADF p value p is:␣
 ↪{adf_test_demanda_electrico_train_tratamiento_media_diff1[1]}")

if adf_test_demanda_electrico_train_tratamiento_media_diff1[0] <␣
 ↪adf_test_demanda_electrico_train_tratamiento_media_diff1[4]['5%']:
     print("Se rechaza H0: SI existe suficiente evidencia para rechazar H0,␣
 ↪por lo tanto SI existe estacionariedad")
else:
    print("Se acepta H0: NO existe suficiente evidencia para rechazar H0, por␣
 ↪lo tanto NO existe estacionariedad")
```

```
The ADF statistic value f is: -6.518615852472467
The ADF p value p is: 1.055865744285777e-08
Se rechaza H0: SI existe suficiente evidencia para rechazar H0, por lo tanto SI
existe estacionariedad
```

[33]:
```
plt.figure(figsize=(10,4))
plt.plot(demanda_electrico_train_tratamiento_media_diff1, color='#0B5345')
plt.plot(demanda_electrico_original_diff1, color='#48C9B0')
plt.title('DIFERENCIA Demanda de gas natural en el sector eléctrico en México␣
 ↪"Ajuste de atípicos por pretratamiento con media"', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize =16)
plt.legend(['Datos Ajuste de atípicos por pretratamiento con media', 'Datos␣
 ↪originales'], loc='lower center', fontsize=10)
```

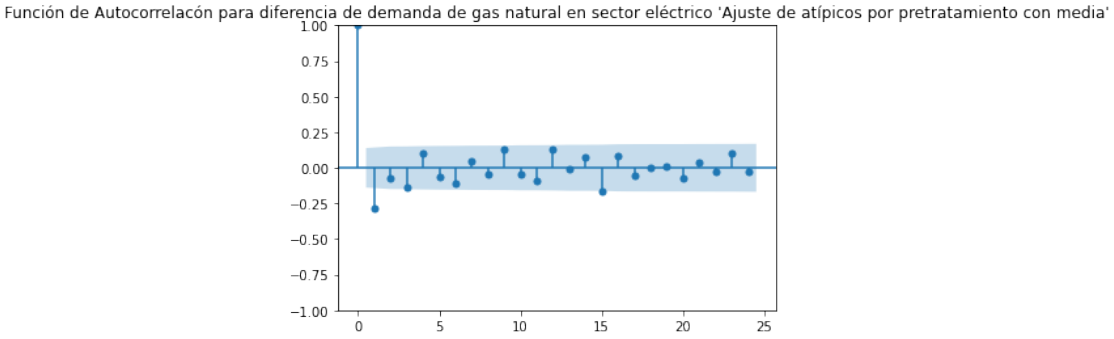
[33]: <matplotlib.legend.Legend at 0x211b823d6f0>



Se muestran las gráficas de ACF y PACF de la DIFERENCIA de Demanda de gas natural en el sector eléctrico en México con los “Datos outliers con tratamiento media”

[34]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()
```

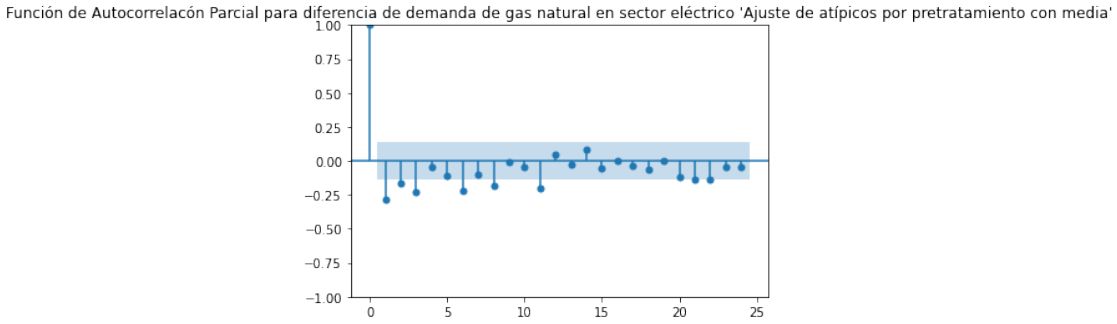
```
# Plot the ACF
tsaplot.plot_acf(demanda_electrico_train_tratamiento_media_diff1.dropna(),␣
 ↪ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón para diferencia de demanda de gas␣
 ↪natural en sector eléctrico 'Ajuste de atípicos por pretratamiento con␣
 ↪media'")

# Show the plot
plt.show()
```



[35]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()

# Plot the ACF
tsaplot.plot_pacf(demanda_electrico_train_tratamiento_media_diff1.dropna(),␣
 ↪ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón Parcial para diferencia de demanda de␣
 ↪gas natural en sector eléctrico 'Ajuste de atípicos por pretratamiento con␣
 ↪media'")

# Show the plot
plt.show()
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\graphics\tsaplots.py:348: FutureWarning: The default method
'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the
default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
```

```
  warnings.warn(
```



Apliquemos un modelo ARIMA a nuestra data de Demanda de gas natural en sector eléctrico 'Datos atipicos con tratamiento media'

[36]:
```
##Create the model
model_ARIMA_electrico_tratamiento_media = ARIMA␣
 ↪(demanda_electrico_train_pretratamiento_media, order=(3,1,1))

##Fit the model
start = time()
model_ARIMA_electrico_tratamiento_media_fit =␣
 ↪model_ARIMA_electrico_tratamiento_media.fit()
end = time()
print('Model fitting time', end-start)

##Summary of the model
print(model_ARIMA_electrico_tratamiento_media_fit.summary())
```

```
Model fitting time 0.1433091163635254
                               SARIMAX Results
==============================================================================
Dep. Variable:           Demanded_Gas   No. Observations:                  201
Model:                 ARIMA(3, 1, 1)   Log Likelihood               -1417.527
Date:                Fri, 06 Oct 2023   AIC                           2845.055
Time:                        14:26:47   BIC                           2861.546
Sample:                  01-01-2005   HQIC                          2851.729
                       - 09-01-2021
Covariance Type:                 opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3898      0.072      5.433      0.000       0.249       0.530
ar.L2          0.0396      0.057      0.698      0.485      -0.072       0.151
ar.L3         -0.0930      0.062     -1.500      0.134      -0.215       0.028
ma.L1         -0.8942      0.050    -17.988      0.000      -0.992      -0.797
```

```
sigma2      8.461e+04   7189.782     11.769      0.000     7.05e+04    9.87e+04
================================================================================
===
Ljung-Box (L1) (Q):                   0.17   Jarque-Bera (JB):
130.40
Prob(Q):                              0.68   Prob(JB):
0.00
Heteroskedasticity (H):               2.12   Skew:
-0.13
Prob(H) (two-sided):                  0.00   Kurtosis:
6.95
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
```

[37]:

```
##get prediction start and end dates
pred_model_ARIMA_electrico_start_date = demanda_electrico_original_test_data.
  ↪index[0]
pred_model_ARIMA_electrico_end_date = demanda_electrico_original_test_data.
  ↪index[-1]
```

[38]:

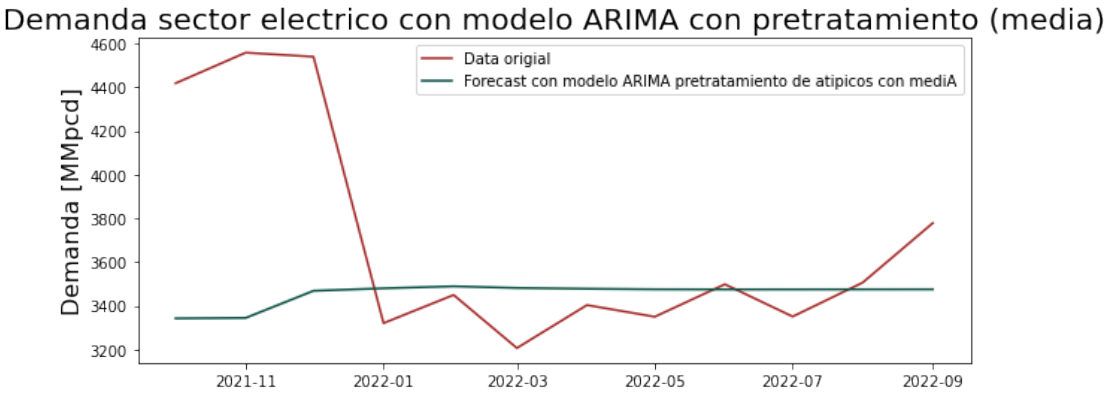
```
##get the predictors and residuals
predictions_model_ARIMA_electrico_tratamiento_media =␣
  ↪model_ARIMA_electrico_tratamiento_media_fit.
  ↪predict(start=pred_model_ARIMA_electrico_start_date, end=␣
  ↪pred_model_ARIMA_electrico_end_date)
print(predictions_model_ARIMA_electrico_tratamiento_media)
```

```
2021-10-01    3342.584843
2021-11-01    3344.701259
2021-12-01    3468.712160
2022-01-01    3480.073521
```

```
2022-02-01    3489.218532
2022-03-01    3481.697420
2022-04-01    3478.071526
2022-05-01    3475.509621
2022-06-01    3475.067053
2022-07-01    3475.130333
2022-08-01    3475.375771
2022-09-01    3475.515108
Freq: MS, Name: predicted_mean, dtype: float64
```

[39]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data['Demanded_Gas'], color='brown')
plt.plot(predictions_model_ARIMA_electrico_tratamiento_media,color='#0B5345')

plt.legend(('Data origial', 'Forecast con modelo ARIMA pretratamiento de␣
 ↪atipicos con mediA'), fontsize=10)


plt.title('Demanda sector electrico con modelo ARIMA con pretratamiento␣
 ↪(media)', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[39]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



Observamos el MAPEy RMSE de demanda_electrico_original_test_data[‘Demanded_Gas’] vs predictions_model_ARIMA_electrico_tratamiento_media

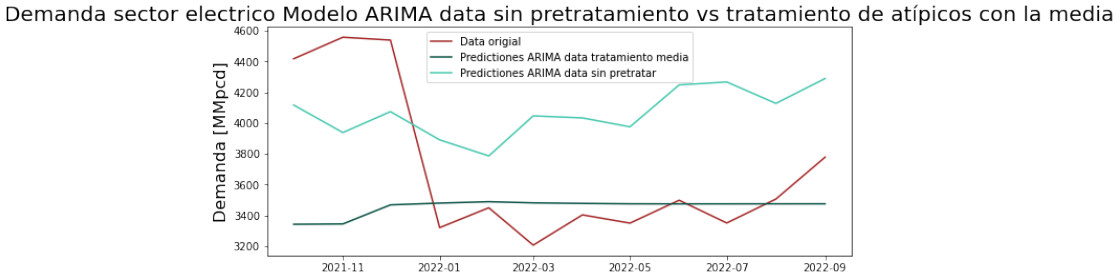
[40]:
```
from sklearn.metrics import mean_absolute_percentage_error, mean_squared_error
import numpy as np

# Compute errors
```

```
MAPE_predictions_model_ARIMA_electrico_tratamiento_media =␣
 ↪mean_absolute_percentage_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_ARIMA_electrico_tratamiento_media)
RMSE_predictions_model_ARIMA_electrico_tratamiento_media = np.
 ↪sqrt(mean_squared_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_ARIMA_electrico_tratamiento_media))

print('MAPE:', MAPE_predictions_model_ARIMA_electrico_tratamiento_media)
print('RMSE:', RMSE_predictions_model_ARIMA_electrico_tratamiento_media)
```

```
MAPE: 0.09020615543992543
RMSE: 577.615681899461
```

[41]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data['Demanded_Gas'], color='brown')
plt.plot(predictions_model_ARIMA_electrico_tratamiento_media, color='#0B5345')
plt.plot(predictions_model_ARIMA_electrico_original, color='#48C9B0')
plt.legend(('Data origial', 'Predictiones ARIMA data tratamiento media',␣
 ↪'Predictiones ARIMA data sin pretratar'), fontsize=10)

plt.title('Demanda sector electrico Modelo ARIMA data sin pretratamiento vs␣
 ↪tratamiento de atípicos con la media', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[41]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



0.11.4 Vamos a entrenar un segundo modelo SARIMA a nuestra data de Demanda de gas natural en sector eléctrico 'Datos atipicos con tratamiento media'

[42]:
```
from statsmodels.tsa.statespace.sarimax import SARIMAX
# Create the SARIMA model
model_SARIMA_electrico_tratamiento_media =␣
 ↪SARIMAX(demanda_electrico_train_pretratamiento_media, order=(3, 1, 1),␣
 ↪seasonal_order=(0, 1, 0, 12))
```

```
# Fit the SARIMA model
model_SARIMA_electrico_tratamiento_media_fit =␣
 ↪model_SARIMA_electrico_tratamiento_media.fit()

# Print the summary of the model
print(model_SARIMA_electrico_tratamiento_media_fit.summary())
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
                                     SARIMAX Results
==========================================================================
===========
Dep. Variable:                          Demanded_Gas   No. Observations:
201
Model:             SARIMAX(3, 1, 1)x(0, 1, [], 12)   Log Likelihood
-1360.903
Date:                             Fri, 06 Oct 2023   AIC
2731.806
Time:                                     14:32:11   BIC
2747.988
Sample:                                 01-01-2005   HQIC
2738.362
                                      - 09-01-2021
Covariance Type:                              opg
==========================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------
ar.L1          0.4934      0.048     10.248      0.000       0.399       0.588
ar.L2          0.0440      0.054      0.815      0.415      -0.062       0.150
ar.L3         -0.0574      0.059     -0.974      0.330      -0.173       0.058
ma.L1         -1.0000      4.729     -0.211      0.833     -10.268       8.268
sigma2      1.102e+05    5.2e+05      0.212      0.832   -9.08e+05    1.13e+06
==========================================================================
===
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):
171.44
Prob(Q):                              0.88   Prob(JB):
0.00
Heteroskedasticity (H):               1.96   Skew:
-0.49
Prob(H) (two-sided):                  0.01   Kurtosis:
```

```
7.57
==============================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

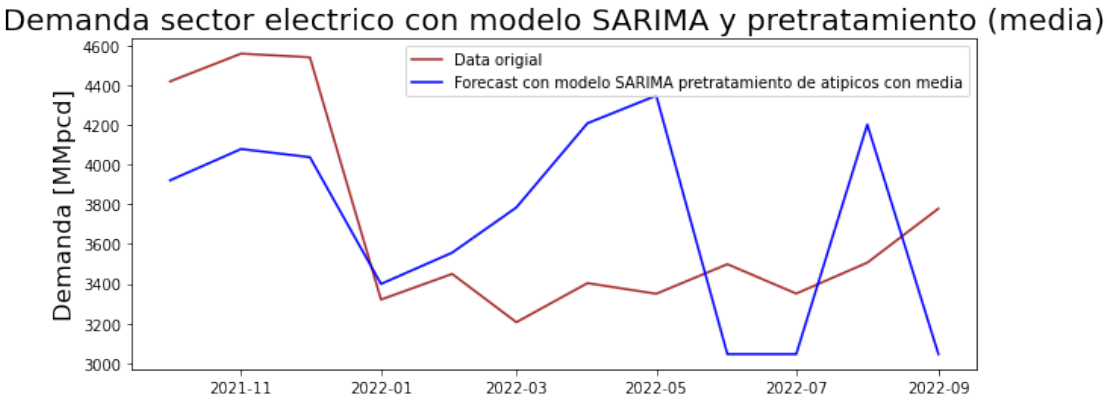
Se obtiene el forecast del model SARIMA electrico tratamiento media

[43]:
```
##get the predictors and residuals
predictions_model_SARIMA_electrico_tratamiento_media =␣
 ↪model_SARIMA_electrico_tratamiento_media_fit.
 ↪predict(start=pred_model_ARIMA_electrico_start_date, end=␣
 ↪pred_model_ARIMA_electrico_end_date)
print(predictions_model_SARIMA_electrico_tratamiento_media)
```

```
2021-10-01    3920.013325
2021-11-01    4077.759914
2021-12-01    4036.702363
2022-01-01    3399.397399
2022-02-01    3556.647381
2022-03-01    3783.898023
2022-04-01    4207.252812
2022-05-01    4346.203977
2022-06-01    3046.396069
2022-07-01    3046.351763
2022-08-01    4200.927895
2022-09-01    3046.329997
Freq: MS, Name: predicted_mean, dtype: float64
```

[44]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data['Demanded_Gas'], color='brown')
plt.plot(predictions_model_SARIMA_electrico_tratamiento_media, color='blue')

plt.legend(('Data origial', 'Forecast con modelo SARIMA pretratamiento de␣
 ↪atipicos con media'))


plt.title('Demanda sector electrico con modelo SARIMA y pretratamiento␣
 ↪(media)', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[44]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



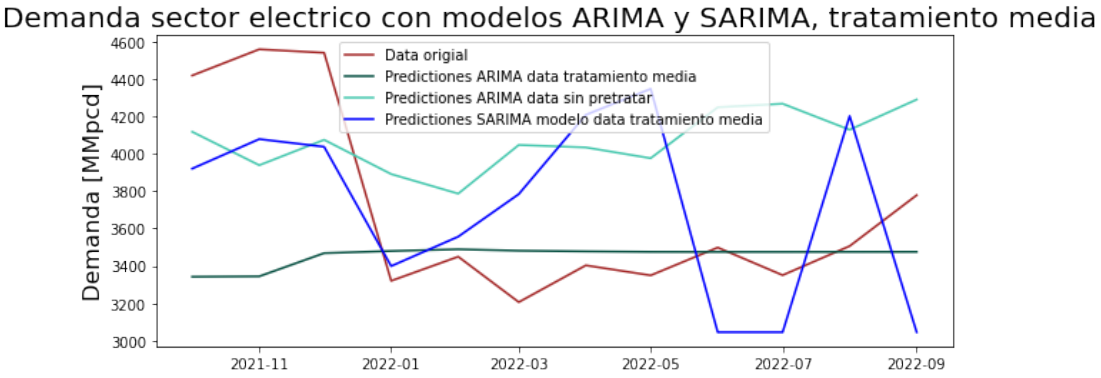
Observamos el MAPEy RMSE de demanda_electrico_original_test_data['Demanded_Gas'] vs predictions_model_SARIMA_electrico_tratamiento_media

[45]:
```
from sklearn.metrics import mean_absolute_percentage_error, mean_squared_error
import numpy as np

# Compute errors
MAPE_predictions_model_SARIMA_electrico_tratamiento_media =␣
 ↪mean_absolute_percentage_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_SARIMA_electrico_tratamiento_media)
RMSE_predictions_model_SARIMA_electrico_tratamiento_media = np.
 ↪sqrt(mean_squared_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_SARIMA_electrico_tratamiento_media))

print('MAPE:', MAPE_predictions_model_SARIMA_electrico_tratamiento_media)
print('RMSE:', RMSE_predictions_model_SARIMA_electrico_tratamiento_media)
```

```
MAPE: 0.14238234601319957
RMSE: 579.8163066140048
```

[46]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data['Demanded_Gas'], color='brown')
plt.plot(predictions_model_ARIMA_electrico_tratamiento_media, color='#0B5345')
plt.plot(predictions_model_ARIMA_electrico_original, color='#48C9B0')
plt.plot(predictions_model_SARIMA_electrico_tratamiento_media, color='blue')
plt.legend(('Data origial', 'Predictiones ARIMA data tratamiento media',␣
 ↪'Predictiones ARIMA data sin pretratar', 'Predictiones SARIMA modelo data␣
 ↪tratamiento media'), fontsize=10)

plt.title('Demanda sector electrico con modelos ARIMA y SARIMA, tratamiento␣
 ↪media ', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[46]: Text(0, 0.5, 'Demanda [MMpcd]')



0.11.5 Proceso de Pretratamiento Ajuste de valores atípicos por mediana

[47]:
```
import numpy as np
import tensorflow as tf

# Set a seed for numpy random number generation
np.random.seed(0)

# Set a seed for TensorFlo
```

[50]:
```
outlier_threshold = 1.2

import pandas as pd

def remove_replace_outliers_mediana(data):
    # Calculate Q1 and Q3
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1

    # Identify outliers
    outliers_mask = (data < (Q1 - outlier_threshold * IQR)) | (data > (Q3 +␣
 ↪outlier_threshold * IQR))

    # Calculate the historical meadian excluding outliers
    historical_median = np.median(data[~outliers_mask])

    # Replace outliers with historical mean
    data.loc[outliers_mask] = historical_median

    return data
```

Se genera nuevamente una copia del demanda_electrico_original para conservar los datos originales en un array y éstos no se vean afectados por el tratamiento de valores atípicos

[48]:
```
# Create a copy of the original DataFrame
demanda_electrico_original_train_data_para_tratamiento2 =␣
 ↪demanda_electrico_original_train_data.copy()
demanda_electrico_original_train_data_para_tratamiento2
```

[48]:
```
            Demanded_Gas
Date
2005-01-01       1819.58
2005-02-01       1895.33
2005-03-01       1765.86
2005-04-01       1642.70
2005-05-01       1895.54
…                  …
2021-05-01       4243.93
2021-06-01       4985.53
2021-07-01       4631.85
2021-08-01       4098.81
2021-09-01       4424.39

[201 rows x 1 columns]
```

[51]:
```
demanda_electrico_train_pretratamiento_mediana =␣
 ↪remove_replace_outliers_mediana(demanda_electrico_original_train_data_para_tratamiento2['De
demanda_electrico_train_pretratamiento_mediana.tail(10)
```

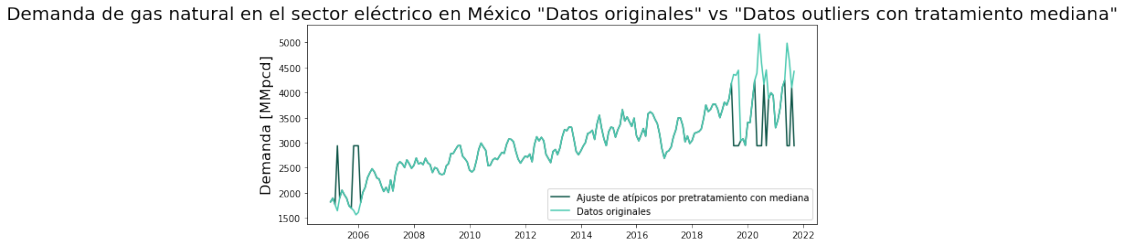
[51]:
```
Date
2020-12-01    3941.12
2021-01-01    3298.43
2021-02-01    3454.21
2021-03-01    3681.30
2021-04-01    4104.82
2021-05-01    4243.93
2021-06-01    2939.05
2021-07-01    2939.05
2021-08-01    4098.81
2021-09-01    2939.05
Name: Demanded_Gas, dtype: float64
```

Se grafica de los datos de entrenamiento de la Demanda de Gas Natural en sector eléctrico sin valores atípicos (tras haber aplicado la función remove_replace_outliers_mediana)

[52]:
```
plt.figure(figsize=(10,4))
plt.plot(demanda_electrico_train_pretratamiento_mediana, color='#0B5345')
plt.plot(demanda_electrico_original_train_data, color='#48C9B0')
```

```
plt.title('Demanda de gas natural en el sector eléctrico en México "Datos␣
 ↪originales" vs "Datos outliers con tratamiento mediana"', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize =16)
# Add a legend to the left bottom corner
plt.legend(['Ajuste de atípicos por pretratamiento con mediana', 'Datos␣
 ↪originales'], loc='lower right', fontsize=10)
```

[52]: <matplotlib.legend.Legend at 0x211b823cf10>



También para este caso se muestran las gráficas de ACF y PACF de la DIFERENCIA de Demanda de gas natural en el sector eléctrico en México

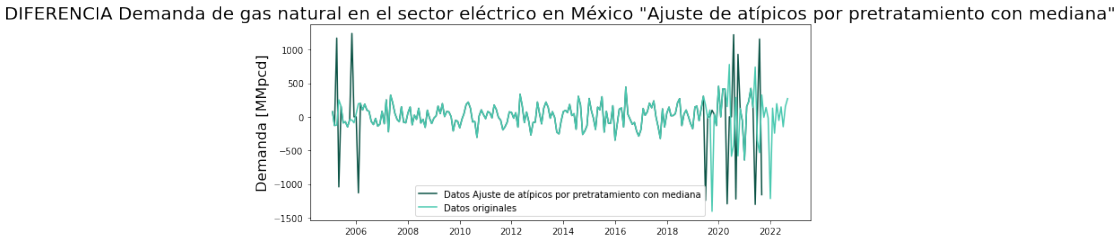
[53]:
```
demanda_electrico_train_tratamiento_mediana_diff1 =␣
 ↪demanda_electrico_train_pretratamiento_mediana.diff()
demanda_electrico_train_tratamiento_mediana_diff1
```

[53]:
```
Date
2005-01-01        NaN
2005-02-01      75.75
2005-03-01    -129.47
2005-04-01    1173.19
2005-05-01   -1043.51
                 …
2021-05-01     139.11
2021-06-01   -1304.88
2021-07-01       0.00
2021-08-01    1159.76
2021-09-01   -1159.76
Name: Demanded_Gas, Length: 201, dtype: float64
```

[54]:
```
plt.figure(figsize=(10,4))
plt.plot(demanda_electrico_train_tratamiento_mediana_diff1, color='#0B5345')
plt.plot(demanda_electrico_original_diff1, color='#48C9B0')
plt.title('DIFERENCIA Demanda de gas natural en el sector eléctrico en México␣
 ↪"Ajuste de atípicos por pretratamiento con mediana"', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize =16)
```

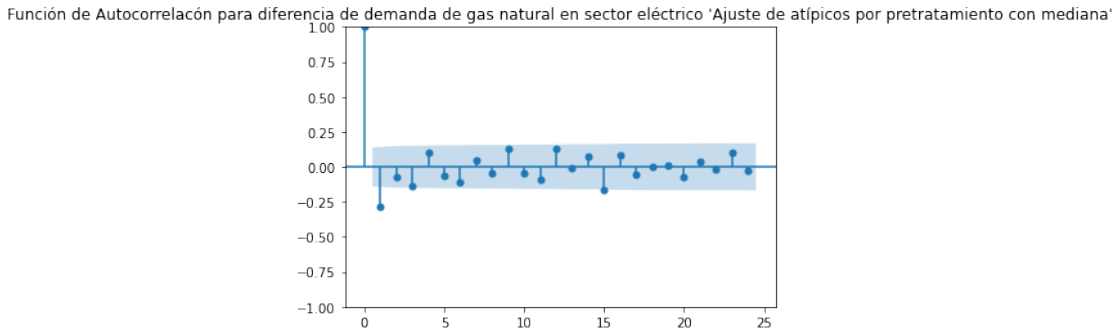
```
plt.legend(['Datos Ajuste de atípicos por pretratamiento con mediana', 'Datos␣
 ↪originales'], loc='lower center', fontsize=10)
```

[54]: <matplotlib.legend.Legend at 0x211c1f371c0>



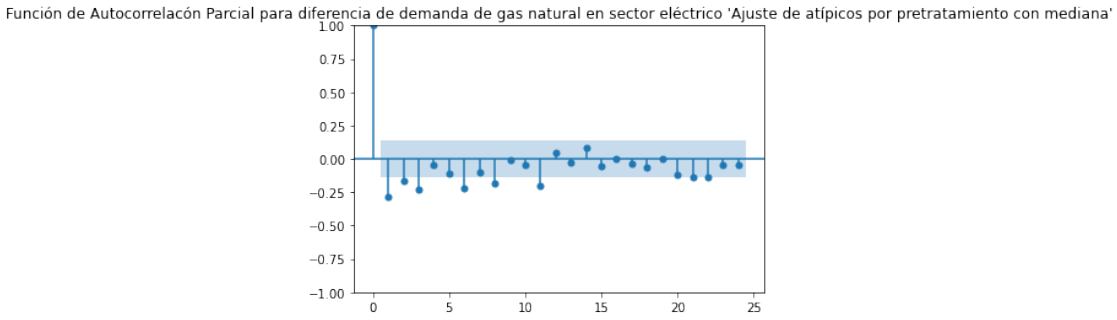
Se muestran las gráficas de ACF y PACF de la DIFERENCIA de Demanda de gas natural en el sector eléctrico en México con los "Datos outliers con tratamiento mediana"

[55]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()

# Plot the ACF
tsaplot.plot_acf(demanda_electrico_train_tratamiento_mediana_diff1.dropna(),␣
 ↪ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón para diferencia de demanda de gas␣
 ↪natural en sector eléctrico 'Ajuste de atípicos por pretratamiento con␣
 ↪mediana'")

# Show the plot
plt.show()
```



[56]:
```
import statsmodels.graphics.tsaplots as tsaplot
# Create the Matplotlib axes object
fig, ax = plt.subplots()

# Plot the ACF
tsaplot.plot_pacf(demanda_electrico_train_tratamiento_mediana_diff1.dropna(),␣
 ↪ax=ax)

# Set the title
ax.set_title("Función de Autocorrelacón Parcial para diferencia de demanda de␣
 ↪gas natural en sector eléctrico 'Ajuste de atípicos por pretratamiento con␣
 ↪mediana'")

# Show the plot
plt.show()
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\graphics\tsaplots.py:348: FutureWarning: The default method
'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the
default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
```



Apliquemos un modelo ARIMA a nuestra data de Demanda de gas natural en sector eléctrico 'Datos atipicos con tratamiento media'

[57]:
```
##Create the model
model_ARIMA_electrico_tratamiento_mediana = ARIMA␣
 ↪(demanda_electrico_train_pretratamiento_mediana, order=(3,1,1))

##Fit the model
start = time()
model_ARIMA_electrico_tratamiento_mediana_fit =␣
 ↪model_ARIMA_electrico_tratamiento_mediana.fit()
end = time()
```

```python
print('Model fitting time', end-start)

##Summary of the model
print(model_ARIMA_electrico_tratamiento_mediana_fit.summary())
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
```

```
Model fitting time 3.7543869018554688
                               SARIMAX Results
==============================================================================
Dep. Variable:           Demanded_Gas   No. Observations:                  201
Model:                 ARIMA(3, 1, 1)   Log Likelihood               -1417.741
Date:                Sat, 07 Oct 2023   AIC                           2845.483
Time:                        09:59:38   BIC                           2861.974
Sample:                    01-01-2005   HQIC                          2852.157
                         - 09-01-2021
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3891      0.072      5.421      0.000       0.248       0.530
ar.L2          0.0397      0.057      0.702      0.483      -0.071       0.151
ar.L3         -0.0922      0.062     -1.488      0.137      -0.214       0.029
ma.L1         -0.8944      0.050    -17.992      0.000      -0.992      -0.797
sigma2      8.488e+04   7251.759     11.705      0.000    7.07e+04    9.91e+04
================================================================================
===
Ljung-Box (L1) (Q):                   0.17   Jarque-Bera (JB):
130.53
Prob(Q):                              0.68   Prob(JB):
0.00
Heteroskedasticity (H):               2.15   Skew:
-0.14
Prob(H) (two-sided):                  0.00   Kurtosis:
6.95
================================================================================
===
```

```
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

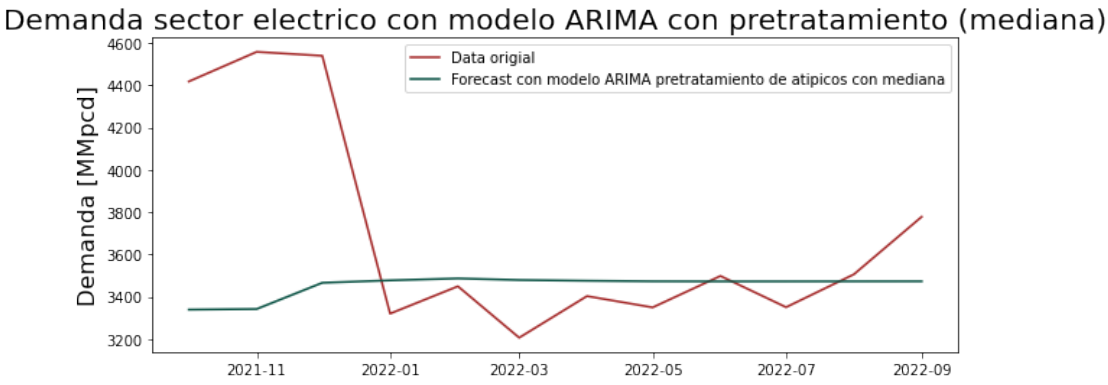
[58]:
```
##get prediction start and end dates
pred_model_ARIMA_electrico_start_date = demanda_electrico_original_test_data.
 ↪index[0]
pred_model_ARIMA_electrico_end_date = demanda_electrico_original_test_data.
 ↪index[-1]
```

[59]:
```
##get the predictors and residuals
predictions_model_ARIMA_electrico_tratamiento_mediana =␣
 ↪model_ARIMA_electrico_tratamiento_mediana_fit.
 ↪predict(start=pred_model_ARIMA_electrico_start_date, end=␣
 ↪pred_model_ARIMA_electrico_end_date)
print(predictions_model_ARIMA_electrico_tratamiento_mediana)
```

```
2021-10-01    3339.624784
2021-11-01    3342.497383
2021-12-01    3466.452952
2022-01-01    3477.875154
2022-02-01    3486.981553
2022-03-01    3479.551807
2022-04-01    3475.969451
2022-05-01    3473.440596
2022-06-01    3472.999075
2022-07-01    3473.057011
2022-08-01    3473.295144
2022-09-01    3473.430819
Freq: MS, Name: predicted_mean, dtype: float64
```

[60]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data['Demanded_Gas'], color='brown')
plt.plot(predictions_model_ARIMA_electrico_tratamiento_mediana,color='#0B5345')

plt.legend(('Data origial', 'Forecast con modelo ARIMA pretratamiento de␣
 ↪atipicos con mediana'), fontsize=10)


plt.title('Demanda sector electrico con modelo ARIMA con pretratamiento␣
 ↪(mediana)', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[60]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



[61]:
```
from sklearn.metrics import mean_absolute_percentage_error, mean_squared_error
import numpy as np

# Compute errors
MAPE_predictions_model_ARIMA_electrico_tratamiento_mediana =␣
 ↪mean_absolute_percentage_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_ARIMA_electrico_tratamiento_mediana)
RMSE_predictions_model_ARIMA_electrico_tratamiento_mediana = np.
 ↪sqrt(mean_squared_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_ARIMA_electrico_tratamiento_mediana))

print('MAPE:', MAPE_predictions_model_ARIMA_electrico_tratamiento_mediana)
print('RMSE:', RMSE_predictions_model_ARIMA_electrico_tratamiento_mediana)
```

```
MAPE: 0.0901689908270974
RMSE: 578.6743217702743
```

Vamos a entrenar un segundo modelo SARIMA a nuestra data de Demanda de gas natural en sector eléctrico 'Datos atipicos con tratamiento mediana'

[62]:
```
from statsmodels.tsa.statespace.sarimax import SARIMAX
# Create the SARIMA model
model_SARIMA_electrico_tratamiento_mediana =␣
 ↪SARIMAX(demanda_electrico_train_pretratamiento_mediana, order=(3, 1, 1),␣
 ↪seasonal_order=(0, 1, 0, 12))

# Fit the SARIMA model
model_SARIMA_electrico_tratamiento_mediana_fit =␣
 ↪model_SARIMA_electrico_tratamiento_mediana.fit()

# Print the summary of the model
print(model_SARIMA_electrico_tratamiento_mediana_fit.summary())
```

```
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
```

```
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Sergio\AppData\Local\Programs\Python\Python310\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency
information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
                                     SARIMAX Results
==========================================================================
==========
Dep. Variable:                          Demanded_Gas   No. Observations:
201
Model:             SARIMAX(3, 1, 1)x(0, 1, [], 12)   Log Likelihood
-1361.009
Date:                             Sat, 07 Oct 2023   AIC
2732.018
Time:                                    10:19:38   BIC
2748.200
Sample:                                01-01-2005   HQIC
2738.575
                                     - 09-01-2021
Covariance Type:                              opg
==========================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------
ar.L1          0.4933      0.048     10.255      0.000       0.399       0.588
ar.L2          0.0429      0.054      0.796      0.426      -0.063       0.149
ar.L3         -0.0560      0.059     -0.949      0.343      -0.172       0.060
ma.L1         -1.0000      3.533     -0.283      0.777      -7.924       5.924
sigma2      1.103e+05   3.88e+05      0.284      0.776   -6.51e+05    8.72e+05
==========================================================================
===
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):
172.53
Prob(Q):                              0.88   Prob(JB):
0.00
Heteroskedasticity (H):               1.98   Skew:
-0.48
Prob(H) (two-sided):                  0.01   Kurtosis:
7.59
==========================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

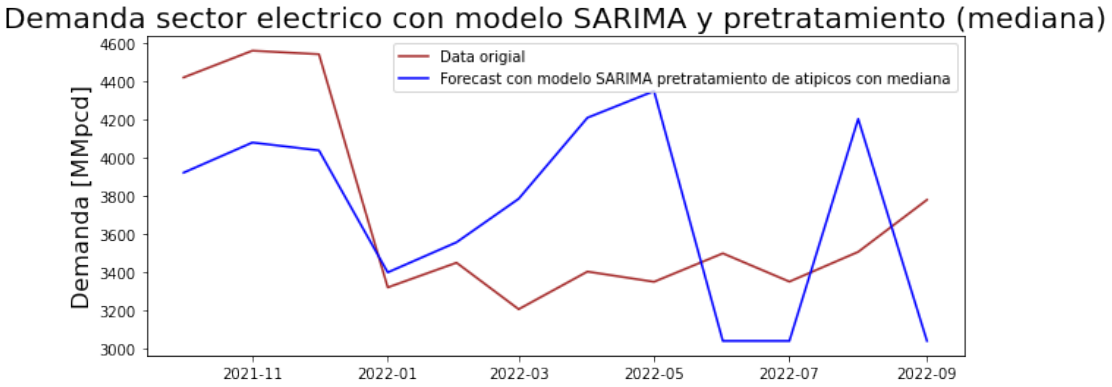
Se obtiene el forecas del model SARIMA electrico tratamiento mediana

[63]:
```
##get the predictors and residuals
predictions_model_SARIMA_electrico_tratamiento_mediana =␣
 ↪model_SARIMA_electrico_tratamiento_mediana_fit.
 ↪predict(start=pred_model_ARIMA_electrico_start_date, end=␣
 ↪pred_model_ARIMA_electrico_end_date)
print(predictions_model_SARIMA_electrico_tratamiento_mediana)
```

```
2021-10-01    3920.040946
2021-11-01    4077.635583
2021-12-01    4036.541825
2022-01-01    3399.248794
2022-02-01    3556.541453
2022-03-01    3783.829918
2022-04-01    4207.210693
2022-05-01    4346.175865
2022-06-01    3041.207332
2022-07-01    3041.165231
2022-08-01    4200.908767
2022-09-01    3041.143793
Freq: MS, Name: predicted_mean, dtype: float64
```

[64]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data['Demanded_Gas'], color='brown')
plt.plot(predictions_model_SARIMA_electrico_tratamiento_mediana, color='blue')

plt.legend(('Data origial', 'Forecast con modelo SARIMA pretratamiento de␣
 ↪atipicos con mediana'))


plt.title('Demanda sector electrico con modelo SARIMA y pretratamiento␣
 ↪(mediana)', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[64]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



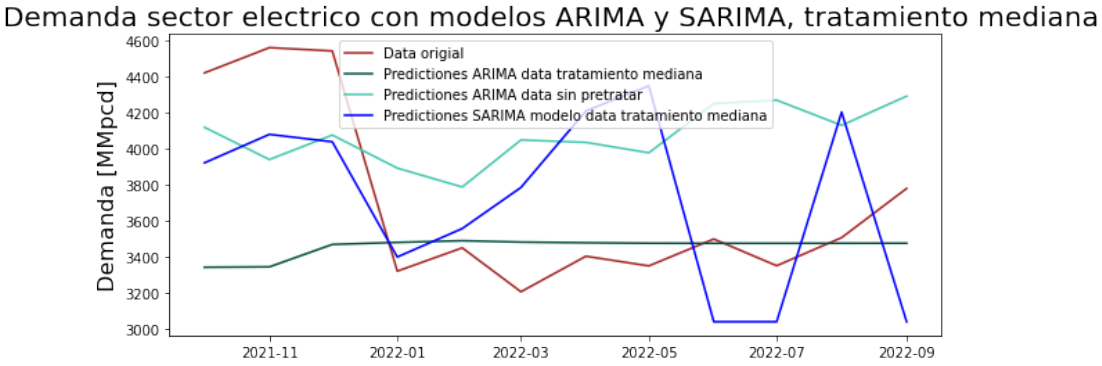
Observamos el MAPEy RMSE de demanda_electrico_original_test_data['Demanded_Gas'] vs predictions_model_SARIMA_electrico_tratamiento_mediana

[65]:
```
from sklearn.metrics import mean_absolute_percentage_error, mean_squared_error
import numpy as np

# Compute errors
MAPE_predictions_model_SARIMA_electrico_tratamiento_mediana =␣
 ↪mean_absolute_percentage_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_SARIMA_electrico_tratamiento_mediana)
RMSE_predictions_model_SARIMA_electrico_tratamiento_mediana = np.
 ↪sqrt(mean_squared_error(demanda_electrico_original_test_data['Demanded_Gas'],␣
 ↪predictions_model_SARIMA_electrico_tratamiento_mediana))

print('MAPE:', MAPE_predictions_model_SARIMA_electrico_tratamiento_mediana)
print('RMSE:', RMSE_predictions_model_SARIMA_electrico_tratamiento_mediana)
```

```
MAPE: 0.14274375472181014
RMSE: 580.9294952263588
```

[67]:
```
##Ploting the predicitons vs the test_data
plt.figure(figsize =(10,4))
plt.plot(demanda_electrico_original_test_data['Demanded_Gas'], color='brown')
plt.plot(predictions_model_ARIMA_electrico_tratamiento_media, color='#0B5345')
plt.plot(predictions_model_ARIMA_electrico_original, color='#48C9B0')
plt.plot(predictions_model_SARIMA_electrico_tratamiento_mediana, color='blue')
plt.legend(('Data origial', 'Predictiones ARIMA data tratamiento mediana',␣
 ↪'Predictiones ARIMA data sin pretratar', 'Predictiones SARIMA modelo data␣
 ↪tratamiento mediana'), fontsize=10)

plt.title('Demanda sector electrico con modelos ARIMA y SARIMA, tratamiento␣
 ↪mediana ', fontsize=20)
plt.ylabel('Demanda [MMpcd]', fontsize=16)
```

[67]:
```
Text(0, 0.5, 'Demanda [MMpcd]')
```



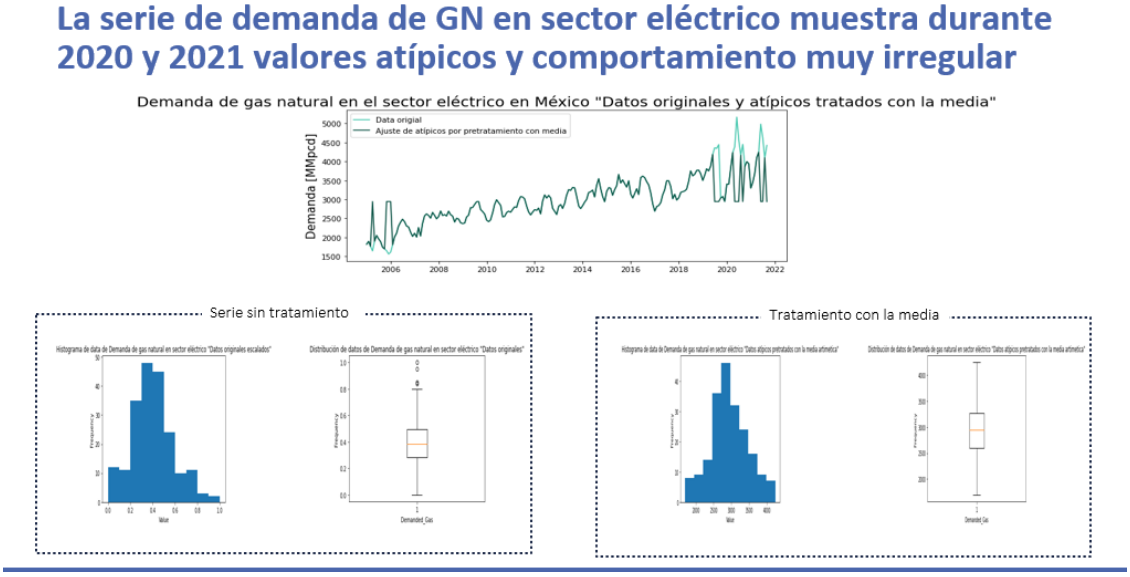
0.12 13. Análisis y ## 14. Resultados

0.12.1 - Para el caso del pronóstico de demanda de Gas Natural (GN) en el sector eléctrico en México

Se muestra la serie de datos de demanda de gas natural en sector eléctrico y el efecto que tiene tratar los outliers con la media aritmetica. En este caso la data pasa de ser no normal a normal tras el tratamiento

[68]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_14_Resultados_electrico1_de_3.
 ↪png', height=500, width=700)
```

[68]:

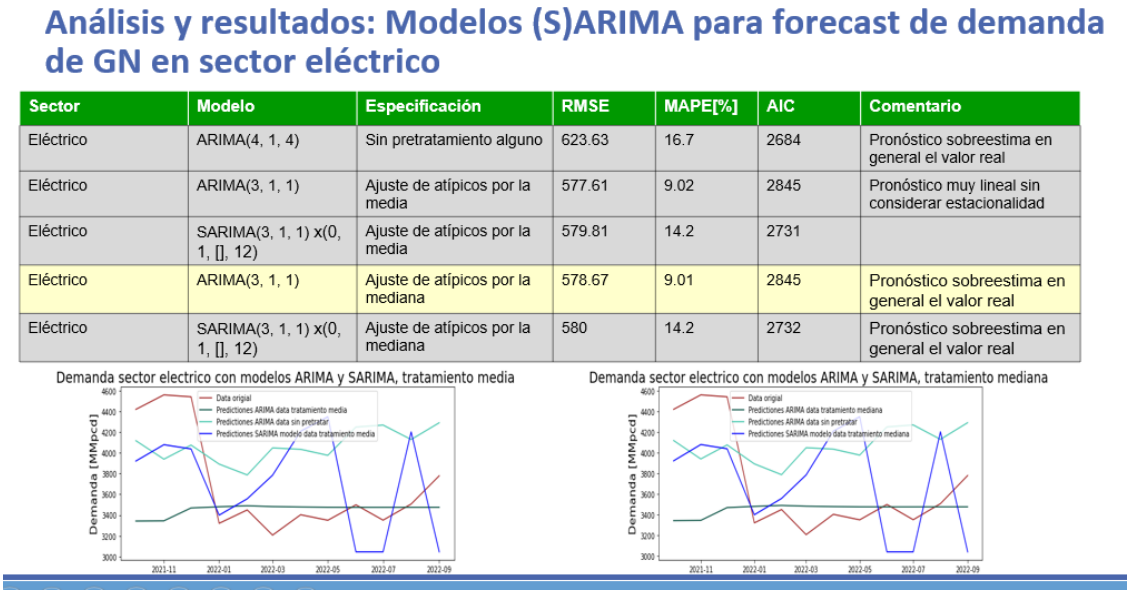


Se muestran los resultados de los modelos (S)ARIMA para el forecast de de demanda de Gas Natural en el sector eléctrico. En donde el modelo con el menor error resulto ser ARIMA(3,1,1)

con tratamiento mediana. Aunque no es un gran modelo en términos de reproducibilidad de estacionalidad

[69]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_14_Resultados_electrico2_de_3.
 ↪png', height=500, width=700)
```

[69]:



Sector	Modelo	Especificación	RMSE	MAPE[%]	AIC	Comentario
Eléctrico	ARIMA(4, 1, 4)	Sin pretratamiento alguno	623.63	16.7	2684	Pronóstico sobreestima en general el valor real
Eléctrico	ARIMA(3, 1, 1)	Ajuste de atípicos por la media	577.61	9.02	2845	Pronóstico muy lineal sin considerar estacionalidad
Eléctrico	SARIMA(3, 1, 1) x(0, 1, [], 12)	Ajuste de atípicos por la media	579.81	14.2	2731	
Eléctrico	ARIMA(3, 1, 1)	Ajuste de atípicos por la mediana	578.67	9.01	2845	Pronóstico sobreestima en general el valor real
Eléctrico	SARIMA(3, 1, 1) x(0, 1, [], 12)	Ajuste de atípicos por la mediana	580	14.2	2732	Pronóstico sobreestima en general el valor real

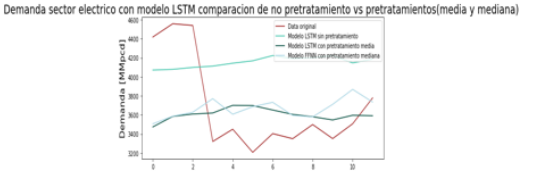
El modelo LSTM con datos atípicos ajustados por la media resultó ser un buen balance entre reproducibilidad de estacionalidad y NO sobreestimación de pronósticos para el caso de la demada de gas natural en sector eléctrico

[70]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_14_Resultados_electrico3_de_3.
 ↪png', height=500, width=700)
```

[70]:

Análisis y resultados: Modelos LSTM para forecast de demanda de GN en sector eléctrico

Sector	Modelo	Especificación	RMSE	MAPE[%]	AIC	Comentario
Eléctrico	LSTM (100, 1, relu, Adam)	Sin pretratamiento alguno	1055	28	NA	Pronóstico sobreestima demasiado el valor real
Eléctrico	LSTM (100, 1, relu, Adam)	Ajuste de atípicos por la media	514.2	10.7	NA	Pronóstico mejora ajustando atípicos
Eléctrico	LSTM (100, 1, relu, Adam)	Ajuste de atípicos por la mediana	539.8	11.3	NA	
Eléctrico	LSTM (100, 1, relu, Adam)	Se normaliza la data de entrenamiento	618	12	NA	Pronóstico no toma mucho en cuenta la estacionalidad
Eléctrico	LSTM (100, 1, relu, Adam)	Se normaliza la data de entrenamiento + Ajuste de atípicos por la media	600	12	NA	Pronóstico no toma en cuenta estacionalidad



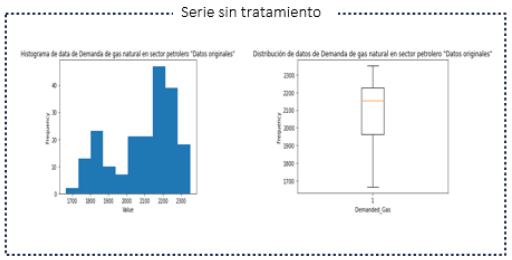
0.12.2 - Para el caso del pronóstico de demanda de Gas Natural (GN) en el sector petrolero en México

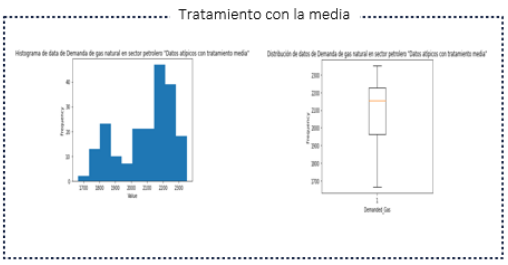
Se muestra la serie de datos de demanda de gas natural en sector petroler y el efecto que tiene tratar los outliers con la media aritmetica. En este el asjute de atípicos NO ayudó a corregir la no normalidad de los datos

[71]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_14_Resultados_petrolero1_de_3.
 ↪png', height=500, width=700)
```

[71]:

La serie de demanda de GN en sector petrolero es No estacionaria y No normal inclusive si se tratan los atípicos





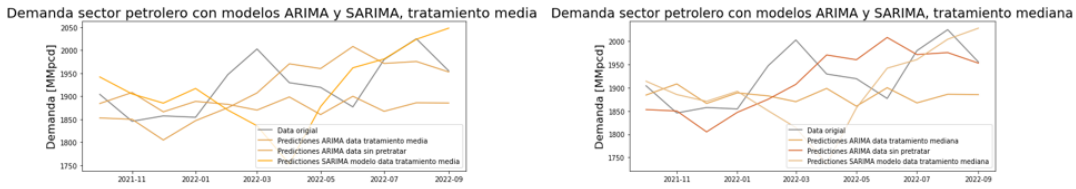
Se muestran los resultados de los modelos (S)ARIMA para el forecast de de demanda de Gas Natural en el sector petrolero. En donde el modelo con el menor error resulto ser ARIMA(4,1,4) sin pretratamiento alguno pues resultó ser un buen balance entre reproducibilidad de estacionalidad y NO sobreestimación de pronósticos

[72]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_14_Resultados_petrolero2_de_3.
 ↪png', height=500, width=700)
```

[72]:

Análisis y resultados: Modelos (S)ARIMA para forecast de demanda de GN en sector petrolero

Sector	Modelo	Especificación	RMSE	MAPE[%]	AIC	Comentario
Petrolero	ARIMA(4, 1, 4)	Sin pretratamiento alguno	60	2.5	2227	
Petrolero	ARIMA(4, 1, 3)	Ajuste de atípicos por la media	75	3.2	2226	
Petrolero	SARIMA(4, 1, 1) x(0, 1, [], 12)	Ajuste de atípicos por la media	87	3.5	2176	
Petrolero	ARIMA(4, 1, 3)	Ajuste de atípicos por la mediana	75	3.2	2226	
Petrolero	SARIMA(3, 1, 1) x(0, 1, [], 12)	Ajuste de atípicos por la mediana	91	3.5	2179	



El modelo LSTM (100, 1, relu, Adam) con datos atípicos ajustados por la media es un buen modelo aunque subestima en general el pronóstico

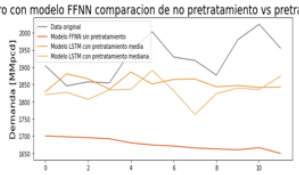
[73]:
```
from IPython.display import Image
Image(filename='Diplomado_DS_UNAM_proyecto_final_14_Resultados_petrolero3_de_3.
 ↪png', height=500, width=700)
```

[73]:

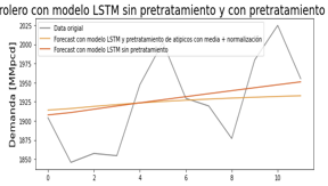
Análisis y resultados: Modelos LSTM para forecast de demanda de GN en sector petrolero

Sector	Modelo	Especificación	RMSE	MAPE[%]	AIC	Comentario
Petrolero	LSTM (100, 1, relu, Adam)	Sin pretratamiento alguno	250	12	NA	Pronóstico subestima en general el valor real
Petrolero	LSTM (100, 1, relu, Adam)	Ajuste de atípicos por la media	90	3.9	NA	Pronóstico subestima en general el valor real
Petrolero	LSTM (100, 1, relu, Adam)	Ajuste de atípicos por la mediana	106	4.7	NA	
Petrolero	LSTM (100, 1, relu, Adam)	Se normaliza la data de entrenamiento	60	2.4	NA	Pronóstico no toma mucho en cuenta la estacionalidad
Petrolero	LSTM (100, 1, relu, Adam)	Se normaliza la data de entrenamiento + Ajuste de atípicos por la media	60	2.5	NA	Pronóstico no toma en cuenta estacionalidad

emanda sector petrolero con modelo FFNN comparacion de no pretratamiento vs pretratamientos(media y mediana)



Demanda sector petrolero con modelo LSTM sin pretratamiento y con pretratamiento (media + normalización)



0.13 15. Conclusiones

En este caso el "mejor modelo" en términos de error para el pronóstico de demanda de gas natural en sector eléctrico fue el LSTM (100, 1, relu, Adam) con datos atípicos ajustados por la media pues resultó ser un buen balance entre reproducibilidad de estacionalidad y NO sobreestimación de pronósticos

En este caso el "mejor modelo" en términos de error y criterio AIC para el pronóstico de demanda de gas natural en sector petrolero fue ARIMA (4,1,4) sin pretratamiento alguno pues resultó ser un buen balance entre reproducibilidad de estacionalidad y NO sobreestimación de pronósticos seguido de LSTM (100, 1, relu, Adam) con datos atípicos ajustados por la media

Algunas conclusiones generales sobre el uso de modelos (S)ARIMA y LSTM para el pronóstico de gas natural en méxico son: -Los modelos de series de tiempo y LSTM son bastante sensibles a valores atípicos y tiene dificultad en captar cambios de tendencia repentinos muy marcados.

-Los modelos de series de tiempo son bastante buenos captando estacionalidad pero pueden exagerar en los pronósticos al sobrestimar o subestimar el forecast.

-Los modelos de LSTM suelen no sobre estimar o subestimar tanto en comparación con los (S)ARIMA NO siempre el normalizar los datos de entramiento ayuda al pronóstico de los modelos LSTM, pues el pronóstico tiende a perder "elasticidad" y capacidad de reflejar estacionalidad

-En este caso si la data es No estacionarie inclusive si se tratan los atípicos (caso de demanda petrolero) los modelos (S)ARIMA fueron mejores que los LSTM para estimar la demanda del Gas Natural pues tuvieron un mejor balance entre la capacidad para reproducir estacionalidad y tendencia