

Laboratorio 3. Deep Learning

Data Science

Links

<https://github.com/FabianJuarez182/DS-LAB3.git>
<https://docs.google.com/doc/edit?usp=sharing>

LSTM para Regular

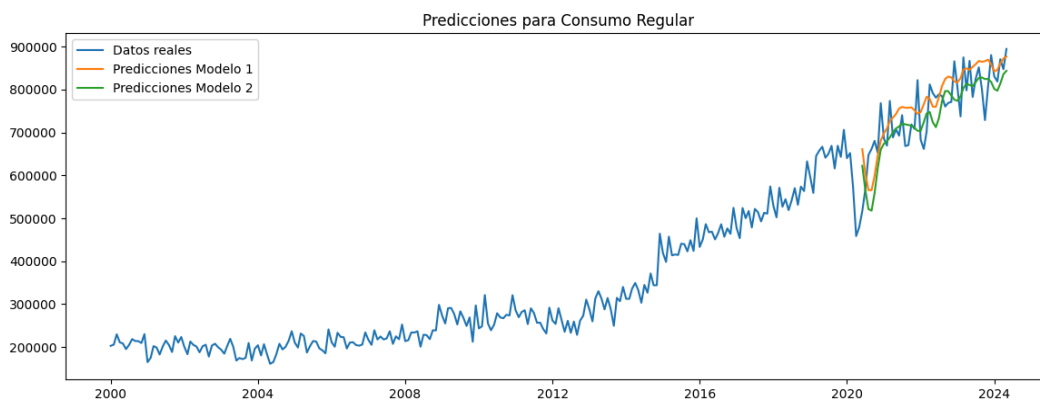


Figura 1. Predicciones para Consumo Regular.

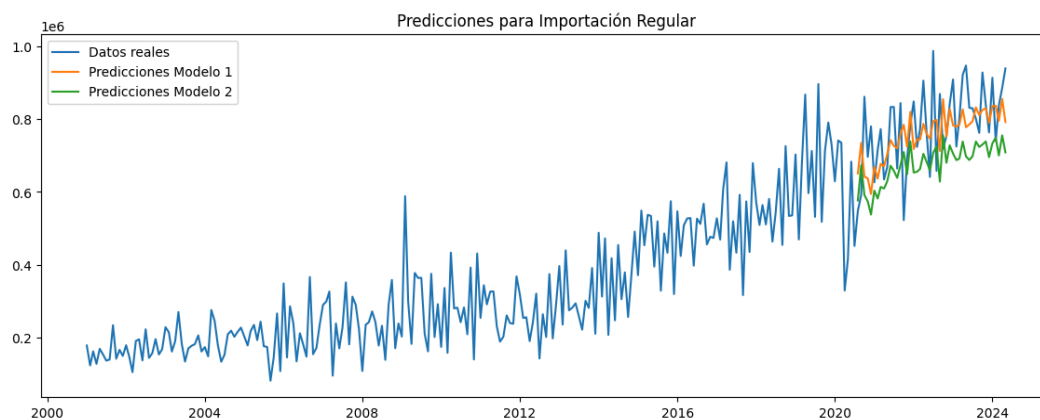


Figura 2. Predicciones para Importaciones Regular.

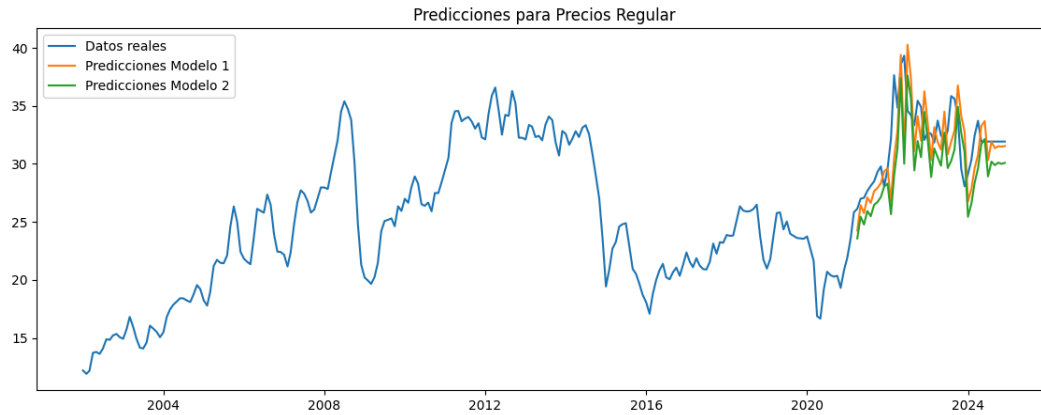


Figura 3. Predicciones para Precios Regular.

Resultados de los modelos usando el combustible Regular

Consumo Regular - MSE Modelo 1: 4128430054.0149274

Consumo Regular - MSE Modelo 2: 3359885438.75914

Importación Regular - MSE Modelo 1: 19219346114.64861

Importación Regular - MSE Modelo 2: 19197134846.465763

Precios Regular - MSE Modelo 1: 4.720558508249839

Precios Regular - MSE Modelo 2: 4.706652907621188

1. Consumo Regular:

- El MSE del Modelo 2 es considerablemente menor que el del Modelo 1 (aproximadamente 18% menor). Esto sugiere que el Modelo 2 tiene un mejor rendimiento en la predicción del consumo regular.

2. Importación Regular:

- Las diferencias en MSE entre ambos modelos son mínimas, con el Modelo 2 mostrando una ligera mejora. Esto indica que incrementar el número de unidades LSTM en el segundo modelo no tiene un impacto significativo en esta variable.

3. Precios Regular:

- Similarmente al caso de la importación, la diferencia en MSE entre los dos modelos es muy pequeña, con el Modelo 2 mostrando una mejora marginal. Esto sugiere que ambos modelos están funcionando de manera similar para la predicción de precios.

Comparación de los Modelos:

- El Modelo (v2), que utiliza 100 unidades en las capas LSTM en lugar de 50 como en el Modelo (v1), parece mejorar ligeramente las predicciones en todas las métricas evaluadas, especialmente en el caso del consumo regular.

LSTM para Super

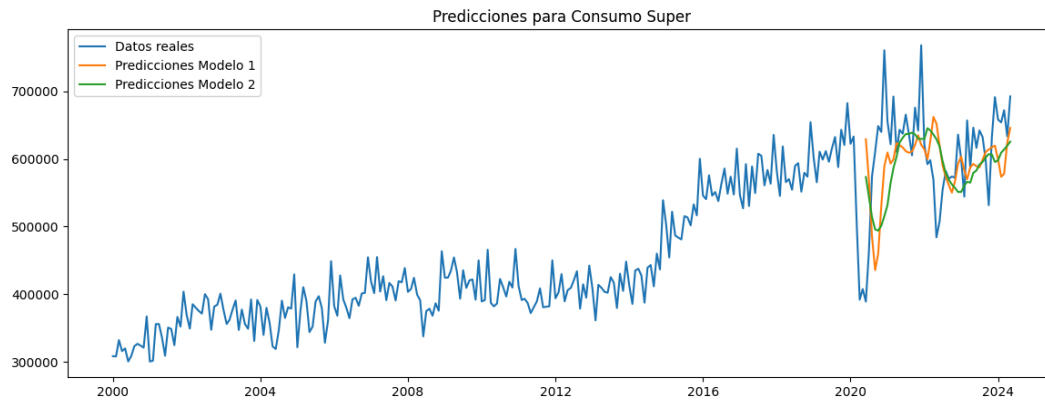


Figura 4. Predicciones para Consumo Super.

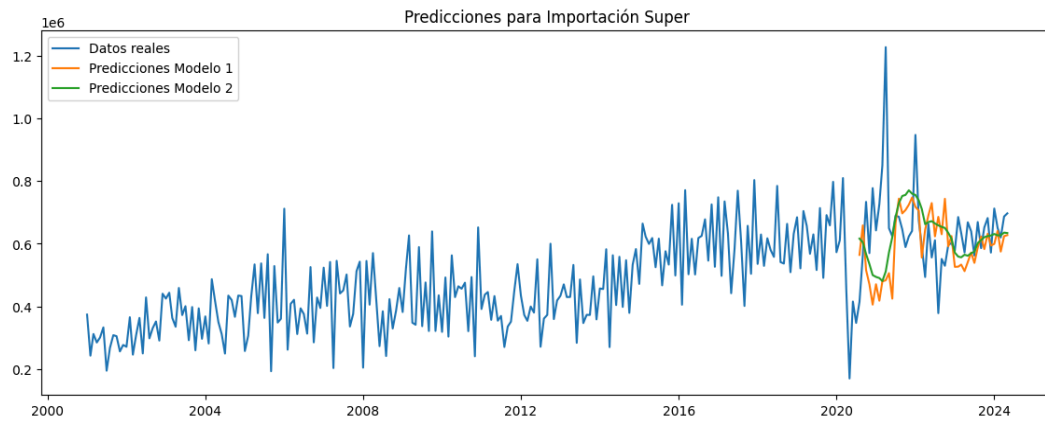


Figura 5. Predicciones para Importaciones Super.

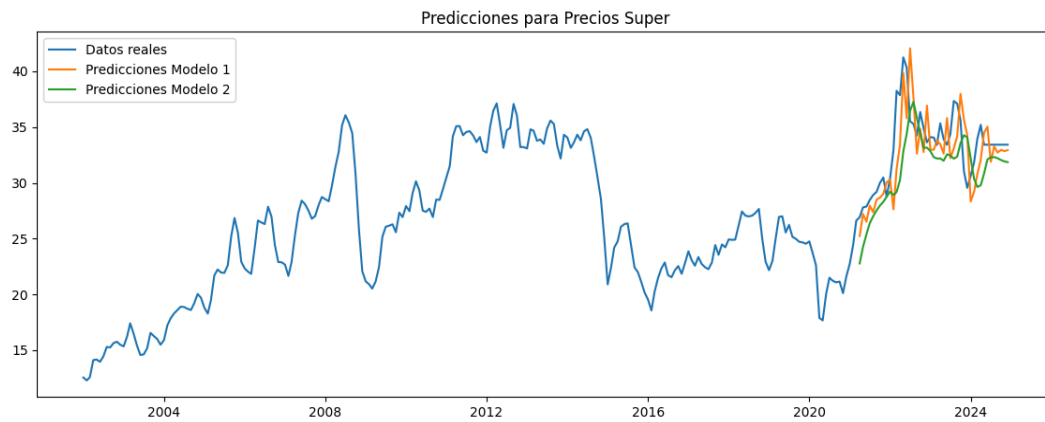


Figura 6. Predicciones para Precios Super.

Resultados de los modelos usando el combustible Super

Consumo Super - MSE Modelo 1: 6010472383.084897
Consumo Super - MSE Modelo 2: 6015801762.3087015
Importación Super - MSE Modelo 1: 32366120915.281803
Importación Super - MSE Modelo 2: 26088167026.96068
Precios Super - MSE Modelo 1: 3.570162377045806
Precios Super - MSE Modelo 2: 8.060372675458458

4. Consumo Super:

- La diferencia en MSE entre los dos modelos es mínima, lo que indica que ambos modelos tienen un rendimiento muy similar para predecir el "Consumo Super". No parece haber una mejora significativa al cambiar de la arquitectura v2 a la v3 en este caso.

5. Importación Super:

- Aquí, el Modelo 2 muestra una mejora significativa con un MSE considerablemente menor que el Modelo 1, lo que sugiere que la arquitectura de v3 es más efectiva para predecir "Importación Super". La incorporación de capas bidireccionales y dropout parece haber ayudado a mejorar la capacidad del modelo para capturar patrones relevantes en estos datos.

6. Precios Super:

- En este caso, el Modelo 1 tiene un rendimiento significativamente mejor que el Modelo 2. El MSE más alto del Modelo 2 sugiere que las mejoras que se añadieron en v3 no ayudaron y posiblemente complicaron la tarea de predecir "Precios Super".

Comparación de los Modelos:

- Modelo (v2): Es un modelo más simple y menos propenso a sobreajuste. Es adecuado cuando los datos no requieren capturar patrones extremadamente complejos, como se ve en "Consumo Super" y "Precios Super".
- Modelo (v3): Aunque más complejo, muestra su valor en tareas como "Importación Super," donde se necesita capturar relaciones más detalladas y secuenciales en los datos. Sin embargo, esta complejidad puede ser contraproducente en otros contextos, como en la predicción de "Precios Super."

LSTM para Diesel

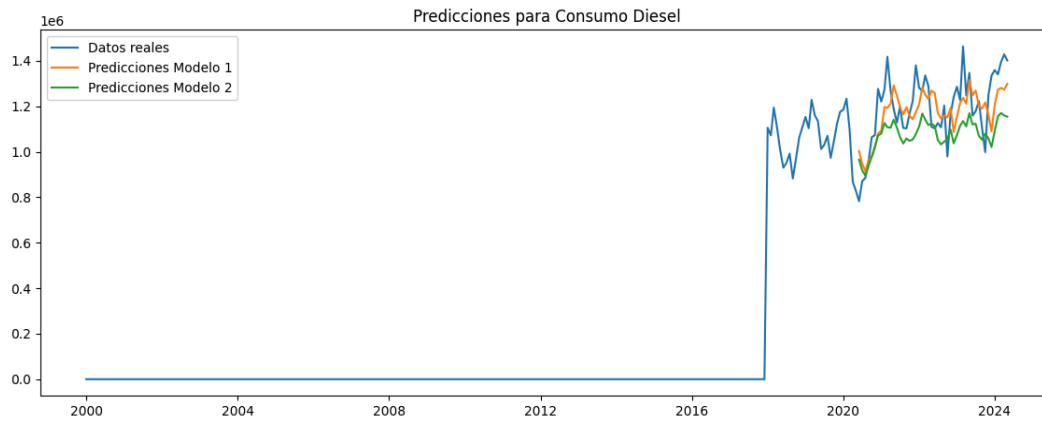


Figura 7. Predicciones para Consumo Diesel.

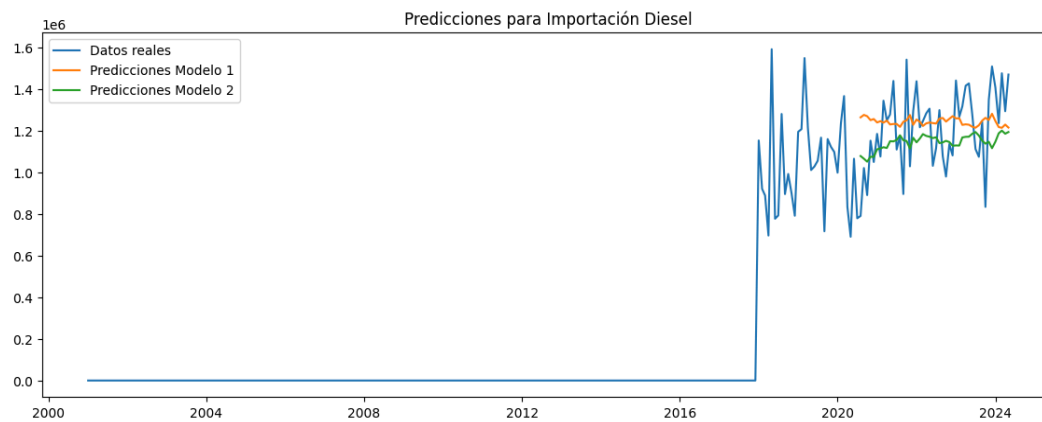


Figura 8. Predicciones para Importaciones Diesel.

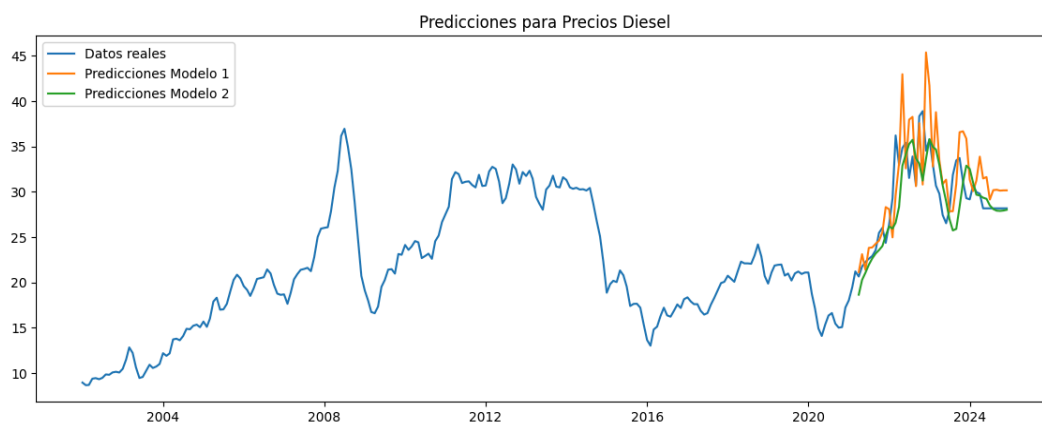


Figura 9. Predicciones para Precios Diesel.

Resultados de los modelos usando el combustible Diesel

Consumo Diesel - MSE Modelo 1: 10999539731.492296
Consumo Diesel - MSE Modelo 2: 24034593659.562145
Importación Diesel - MSE Modelo 1: 39802234593.91359
Importación Diesel - MSE Modelo 2: 34032278107.916706
Precios Diesel - MSE Modelo 1: 12.780979415367458
Precios Diesel - MSE Modelo 2: 6.480447288699467

7. Consumo Diesel:

- El Modelo 1 tiene un MSE significativamente menor que el Modelo 2, lo que indica que el Modelo 1 es más efectivo para predecir el "Consumo Diesel". La mayor complejidad del Modelo v3 (Modelo 2) parece haber resultado en un sobreajuste o una ineficacia en capturar los patrones relevantes para este caso.

8. Importación Diesel:

- El Modelo 2 muestra un rendimiento mejor con un MSE menor que el Modelo 1. Esto sugiere que la mayor complejidad del Modelo v3 (Modelo 2) es beneficiosa para capturar los patrones en los datos de "Importación Diesel".

9. Precios Diesel:

- De manera similar, el Modelo 2 supera significativamente al Modelo 1 en la predicción de "Precios Diesel", con un MSE mucho menor. Esto refuerza la idea de que, para este tipo de predicción, la complejidad adicional del Modelo v3 es ventajosa.

Comparación de los Modelos:

- Modelo 1 (v2): Es más óptimo para tareas donde los datos no requieren capturar patrones extremadamente complejos, como en la predicción de "Consumo Diesel". Su simplicidad evita el sobreajuste y permite un rendimiento confiable y eficiente.
- Modelo 2 (v3): Resulta más óptimo para tareas donde se necesita capturar relaciones más complejas en los datos, como en "Importación Diesel" y "Precios Diesel". Aunque más complejo, este modelo demuestra un rendimiento superior en estas tareas específicas.

Resultados Finales

```
1
2 def train_lstm_model(X_train, y_train, units=50, epochs=50, batch_size=1):
3     model = Sequential()
4     model.add(LSTM(units, return_sequences=True, input_shape=(X_train.shape[1], 1)))
5     model.add(LSTM(units, return_sequences=False))
6     model.add(Dense(1))
7     model.compile(optimizer='adam', loss='mean_squared_error')
8     model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size, verbose=1)
9     return model
10
```

```
1 def train_lstm_model_v2(X_train, y_train, units=100, epochs=50, batch_size=1):
2     model = Sequential()
3     model.add(LSTM(units, return_sequences=True, input_shape=(X_train.shape[1], 1)))
4     model.add(LSTM(units, return_sequences=False))
5     model.add(Dense(1))
6     model.compile(optimizer='adam', loss='mean_squared_error')
7     model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size, verbose=1)
8     return model
9
```

```
5 def train_lstm_model_v3(X_train, y_train, units=100, epochs=50, batch_size=32, dropout_rate=0.2, learning_rate=0.001):
6     model = Sequential()
7
8     model.add(Bidirectional(LSTM(units, return_sequences=True, input_shape=(X_train.shape[1], 1))))
9     model.add(Dropout(dropout_rate))
10
11     model.add(Bidirectional(LSTM(units, return_sequences=False)))
12     model.add(Dropout(dropout_rate))
13
14     model.add(Dense(25, activation='relu'))
15     model.add(Dense(1))
16
17     optimizer = Adam(learning_rate=learning_rate)
18     model.compile(optimizer=optimizer, loss='mean_squared_error')
19
20     model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size, verbose=1)
21
22     return model
```

Se hicieron un total de tres modelos LSTM, nombrados como v1, v2, y v3. Tras realizar las pruebas para cada tipo de combustible, al inicio se descartó el modelo v1 que era el más simple y dio los peores resultados, por lo que a posteriori se usaron los modelos v2 y v3, nombrados como modelos 1 y 2, respectivamente. La diferencia entre la estructura de estos modelos se hizo para tener un tuneo de parámetros lo suficientemente variado que al mismo tiempo involucra el factor de la complejidad del modelo como otro detalle a tomar en cuenta, además de los resultados, para la elección del más óptimo.

- **Consumo:** El **Modelo 1** es generalmente preferible, especialmente para Diesel y Super, debido a su menor MSE, aunque el **Modelo 2** es mejor para Regular.
- **Importación:** El **Modelo 2** es consistentemente mejor para todos los combustibles.
- **Precios:** El **Modelo 2** es superior para Diesel y Regular, mientras que el **Modelo 1** es mejor para Super.

Elección General del Modelo por Categoría:

- **Consumo: Modelo 1** en general (considerando Diesel y Super).
- **Importación: Modelo 2** para todos los combustibles.
- **Precios: Modelo 2** para Diesel y Regular, **Modelo 1** para Super.

Comparación ARIMA y LSTM

1. Precios Diesel:

- **RMSE ARIMA:** 0.9630
- **RMSE LSTM:** 2.55
- **Conclusión:** El modelo ARIMA tiene un RMSE significativamente menor que el modelo LSTM, lo que sugiere que ARIMA es el mejor modelo para predecir los precios del Diesel. La diferencia es considerable, indicando que ARIMA es más preciso en este caso.

2. Precios Super:

- **RMSE ARIMA:** 1.0711
- **RMSE LSTM:** 2.84
- **Conclusión:** Nuevamente, el modelo ARIMA supera al modelo LSTM, con un RMSE mucho más bajo. Esto indica que ARIMA es más adecuado para modelar y predecir los precios de Super.

3. Precios Regular:

- **RMSE ARIMA:** 1.0721
- **RMSE LSTM:** 2.17
- **Conclusión:** El modelo ARIMA también ofrece un mejor rendimiento en la predicción de los precios de Regular, con un RMSE casi la mitad del obtenido por el modelo LSTM.

Conclusiones

En todas las series de tiempo (Precios Diesel, Precios Super, y Precios Regular), el modelo ARIMA tiene un RMSE más bajo que el modelo LSTM, lo que indica que ARIMA es el modelo más preciso para estas predicciones. Esto sugiere que, a pesar de la capacidad de LSTM para capturar relaciones complejas en los datos, en este caso particular, la simplicidad y efectividad de ARIMA ofrecen mejores resultados.

Esto podría deberse a que las series de tiempo en cuestión no presentan patrones extremadamente complejos o no lineales que requieran la complejidad de un modelo LSTM, y los métodos tradicionales como ARIMA son suficientemente robustos y precisos para capturar las tendencias subyacentes. Por lo tanto, para estas series de tiempo, ARIMA es la mejor opción en términos de precisión y simplicidad.