Carlos Enrique Jaramillo Aros

MCD – FAII – Parcial Practico

Objetivo

En días recientes trabajamos con una compañía de comestibles que estaba interesada en predecir el comportamiento de las ventas (en unidades) de sus dos productos estrella. El equipo fue contratado para generar un modelo que permita pronosticar las ventas del siguiente mes de cada uno de esos dos productos. La base de datos disponible en el archivo Examen.csv tiene la información de cada uno de los productos desde enero de 2008.

El objetivo es encontrar el mejor modelo para pronosticar cada una de las series, de igual manera se deberán explicar de manera clara los modelos utilizados enfocado en los dos productos.

Análisis de contexto de los datos.

Con el fin de llevar a cabo el uso adecuado de modelos de series de tiempo, se realizaron diferentes validaciones previas para la preparación de los datos, de igual manera se realizó un análisis exploratorio de los datos que permitió descomponer las series de tiempo y esto a su vez permitió tomar decisiones sobre los modelos a utilizar.

Dentro de la exploración de los datos se logró encontrar lo siguiente:

1. El conjunto de datos está compuesto por (127 registros y 2 columnas (para cada producto).
2. El índice del conjunto de datos es un iterador de 1 a 127, por esta razón es importante realizar un ajuste al mismo, para que inicie en enero de 2008.
3. No se cuenta con valores duplicados o faltantes en el conjunto de datos.
4. El producto 1 cuenta con una distribución aproximadamente normal, cosa que no ocurre completamente con el producto 2, el cual es un poco mas variable (la distribución es mas ancha).

```
data.shape
```

```
(127, 2)
```

```
print(data.dtypes)
```

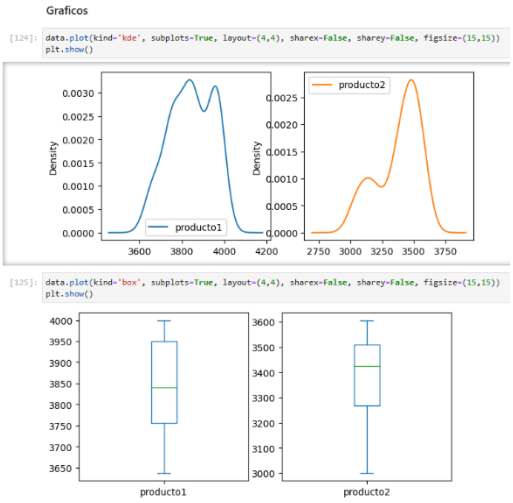
```
producto1    float64
producto2    float64
dtype: object
```

```
data.duplicated().sum()
```

```
0
```

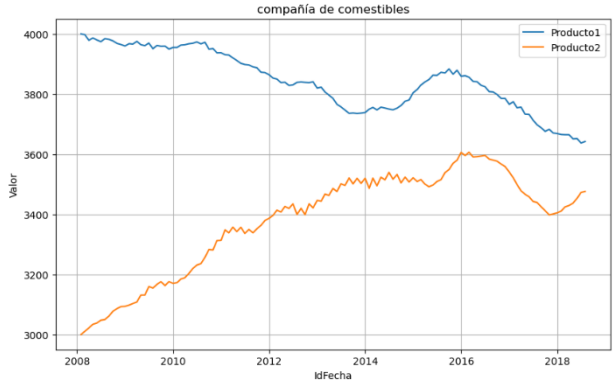
```
data.isna().sum()
```

```
producto1    0
producto2    0
dtype: int64
```



5. De forma general el producto1 tiene una tendencia aproximada decreciente y el producto 2 tiene una tendencia aproximada creciente.

```
plt.figure(figsize=(10, 6))
plt.plot(data.index, data['producto1'], label='Producto1')
plt.plot(data.index, data['producto2'], label='Producto2')
plt.title('compañía de comestibles')
plt.xlabel('IdFecha')
plt.ylabel('Valor')
plt.legend()
plt.grid()
plt.show()
```



6. Al realizar la descomposición en Error (Error), Tendencia (Trend), Estacionalidad (Seasonal). Podemos corroborar (pero no concluir) el comportamiento de ambas series de tiempo (producto 1 y producto 2).

```
td_componentes = seasonal_decompose(data[["producto1"]],model="additive",period=1)
fig = td_componentes.plot()
fig.set_size_inches((16, 9))
fig.tight_layout()
plt.show()
```

```
td_componentes = seasonal_decompose(data[["producto2"]],model="additive",period=1)
fig = td_componentes.plot()
fig.set_size_inches((16, 9))
fig.tight_layout()
plt.show()
```

Con esta revisión y entendimiento (numérica y no funcional de los datos) podemos determinar que los modelos que podrían aplicar para esta revisión que requiere el cliente (predicción mes a mes).

Se plantea desarrollar el ejercicio (Usando diferentes librerías de Python) ejecutando las diferentes configuraciones y protocolo de evaluación **ventana móvil** . Esta metodología se utiliza para evaluar el rendimiento de modelos predictivos al dividir el conjunto de datos en segmentos secuenciales o "ventanas" y entrenar el modelo en cada una de ellas. La ventana se desplaza a lo largo del tiempo, lo que permite evaluar la capacidad del modelo para generalizar patrones a lo largo de diferentes períodos. Esta aproximación es crucial para garantizar que el modelo sea robusto y capaz de hacer predicciones precisas en datos futuros desconocidos, esto permite ver de manera dinámica el desempeño de los modelos, de igual manera ayuda a mitigar el sobre ajuste (overfitting) evitando “patrones locales” en cierto segmento de datos.

Para poder iterar en los diferentes hiperparametros de los modelos a revisar, se planteará utilizar **optimización bayesiana** con 100 iteraciones con cada conjunto de datos (esto genera 100 modelos) para poder selección un subconjunto pequeño (top 5) de configuraciones que permitan comparar entre modelos.

El objetivo es encontrar la combinación óptima de parámetros que maximice o minimice una métrica de rendimiento. En lugar de explorar sistemáticamente todo el espacio de búsqueda, la optimización bayesiana utiliza un modelo de probabilidad para estimar la función objetivo y guiar la búsqueda hacia las regiones más prometedoras.

se basa en el teorema de Bayes para actualizar de manera iterativa la distribución de probabilidad de la función objetivo a medida que se realizan más evaluaciones, para poder ejecutarla se usa el paquete BayesianOptimization de Python (!pip install fastai wwf bayesian-optimization -q –upgrade)

Teniendo en cuenta que el conjunto de datos (producto1 y producto2) no cuentan con estacionalidad, esto será generará una restricción en uno de los grupos de modelos que serán usados.

Los siguientes, son un grupo de modelos usados para evaluar y realizar la predicción para poder dar al cliente.

1. ARIMA: El modelo ARIMA combina componentes autoregresivas (AR), de integración (I) y de media móvil (MA) para modelar patrones en una serie temporal. La notación ARIMA(p, d, q) representa los órdenes de estas componentes. La parte autoregresiva (AR) utiliza las observaciones pasadas, la parte de integración (I) se refiere al grado de diferenciación necesario para hacer estacionaria la serie, y la parte de media móvil (MA) se basa en los errores pasados.

Top 5 Resultados ARIMA

Producto 1

	Producto	Modelo	Configuración	RMSE	Coef R2
0	producto1	ARIMA 1	P=1, I=2, Q=2	188.632619	-0.664830
1	producto1	ARIMA 2	P=1, I=2, Q=2	188.632619	-0.664830
2	producto1	ARIMA 3	P=2, I=2, Q=1	188.802853	-0.667836
3	producto1	ARIMA 4	P=1, I=2, Q=2	188.632619	-0.664830
4	producto1	ARIMA 5	P=2, I=2, Q=2	188.830730	-0.668328

Producto 2

	Producto	Modelo	Configuración	RMSE	Coef R2
11	producto2	ARIMA 1	P=2, I=2, Q=1	237.015996	-0.608427
12	producto2	ARIMA 2	P=1, I=2, Q=2	238.091874	-0.623063
13	producto2	ARIMA 3	P=1, I=2, Q=2	238.091874	-0.623063
14	producto2	ARIMA 4	P=2, I=2, Q=2	238.052745	-0.622529
15	producto2	ARIMA 5	P=2, I=2, Q=1	237.015996	-0.608427

2. ETS : El modelo ETS representa un enfoque que descompone la serie temporal en componentes de error (E), tendencia (T) y estacionalidad (S), cada parámetros controla la presencia y naturaleza de cada componente y los parámetros Alpha, Beta y Gamma correspondientemente, controla la suavización de cada componente.

Producto 1

	Producto	Modelo	Configuración	RMSE	Coef R2
5	producto1	Suavizacion ETS 1	(add, mul,None), a=0.748, B=0.799, g=0.632	36.695973	0.820644
6	producto1	Suavizacion ETS 2	(add, add,None), a=0.135, B=0.893, g=0.797	36.791229	0.819711
7	producto1	Suavizacion ETS 3	(add, add,None), a=0.571, B=0.958, g=0.858	36.469331	0.822852
8	producto1	Suavizacion ETS 4	(add, add,None), a=0.704, B=0.844, g=0.501	36.625857	0.821328
9	producto1	Suavizacion ETS 5	(add, add,None), a=0.806, B=0.572, g=0.99	36.973388	0.817921

Producto 2

	Producto	Modelo	Configuración	RMSE	Coef R2
16	producto2	Suavizacion ETS 1	(mul, add,None), a=0.01, B=0.01, g=0.01	63.238126	0.598541
17	producto2	Suavizacion ETS 2	(mul, mul,None), a=0.99, B=0.559, g=0.01	47.083514	0.777453
18	producto2	Suavizacion ETS 3	(add, mul,None), a=0.01, B=0.99, g=0.99	47.501632	0.773483
19	producto2	Suavizacion ETS 4	(add, mul,None), a=0.99, B=0.99, g=0.01	48.159261	0.767168
20	producto2	Suavizacion ETS 5	(mul, mul,None), a=0.01, B=0.99, g=0.533	47.490425	0.773590

3. Modelo Polinómico de Grado 2: El modelo polinómico de segundo grado es una regresión polinómica que utiliza una función cuadrática para ajustarse a la serie temporal y su formula general es :

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t$$

Los B son los coeficientes del polinomio,

t Tiempo

e , termino del error

Producto 1

	Producto	Modelo	Configuración	RMSE	Coef R2
10	producto1	Polinomica	Grado 2	73.807227	-144.175998

Producto 2

	Producto	Modelo	Configuración	RMSE	Coef R2
21	producto2	Polinomica	Grado 2	23.793228	0.956262

Con estos resultados y generando un resumen de las métricas de evaluación utilizadas (RMSE y Coef. R2) podemos concluir que el mejor modelo es.

Pruebas de supuesto

Una vez ejecutada la ejecución de los modelos, se plantea ejecutar las pruebas sobre supuestos (**Autocorrelacion**, **Homocedasticidad**, **Normalidad**) con el fin de garantizar la validez de los pronósticos generados con estos modelos, se encontró que este modelo cumple con todas las pruebas de supuestos realizadas.

1. Autocorrelación [Box-Pierce, Ljung-Box]:
   p-value = 0
2. Homoscedasticidad [Ljung-Box] (restando media y elevando al cuadrado:
   p-value = 0
3. Normalidad [shapiro - JarqueBera]:
   p-value > 0.05

   *Nota para el producto 2, el mejor modelo cuenta con ciertas ambigüedades en el resultado, entonces se toma la decisión mayoritaria.

Limitaciones

Limitaciones de los pronósticos están ligadas a la presunción de continuidad de los datos basado en los patrones del pasado, en caso de presentarse cambios abruptos, esto podría generar un pronostico no acertado.

Conclusión

Este informe corto muestra la evaluación de los modelos con diferentes configuraciones sobre los hiperparemtros para las series de tiempo de los dos productos estrella de esta compañía de comestibles.

Basado en el cálculo de los errores encontrados en cada uno de los modelos, se puede determinar que

Para el producto 1, el modelo con menor error es **Suavizacion ETS (add,add,None)** según **RMSE**, **Coef. R2**.

Para el producto 2, el modelo con menor error es **Polinomica de grado 2** según **RMSE**, **Coef. R2**.