

05_forecasting_demanda

November 11, 2025

1 Forecasting de Demanda

1.1 Librerías

```
[2]: !pip install prophet --quiet
```

```
[36]: import pandas as pd
import numpy as np
from datetime import datetime
from statsmodels.tsa.arima.model import ARIMA
import warnings
warnings.filterwarnings("ignore")
```

```
[37]: # =====
#   Generar series temporales sintéticas desde métricas de venta
#   =====
import pandas as pd
import numpy as np
from datetime import datetime, timedelta

# Cargar datos base
df = pd.read_csv("metricas_venta_integradas.csv")

# Crear rango de fechas semanales
fechas = pd.date_range(start="2024-01-01", periods=52, freq="W")

# Crear dataframe vacío
datos_sinteticos = []

# Simular ventas semanales
for _, row in df.iterrows():
    for fecha in fechas:
        # Generar demanda aleatoria en torno a la media y desviación
        cantidad = max(0, np.random.normal(row["demanda_promedio"],
        ↪row["desviacion_demanda"]))
        datos_sinteticos.append({
            "fecha": fecha,
            "sku": row["sku"],
```

```

        "tienda": row["tienda"],
        "cantidad_vendida": round(cantidad, 2)
    })

# Convertir a DataFrame
ventas_sinteticas = pd.DataFrame(datos_sinteticos)
ventas_sinteticas.to_csv("ventas.csv", index=False)

print(" Archivo 'ventas.csv' generado con series sintéticas.")
print(ventas_sinteticas.head(5))

```

```

    Archivo 'ventas.csv' generado con series sintéticas.
      fecha      sku      tienda  cantidad_vendida
0 2024-01-07  HM000001  TIENDA001             1.86
1 2024-01-14  HM000001  TIENDA001             2.25
2 2024-01-21  HM000001  TIENDA001             0.00
3 2024-01-28  HM000001  TIENDA001             1.55
4 2024-02-04  HM000001  TIENDA001             0.31

```

1.2 Generar series temporales sintéticas

```

[41]: # =====
#   Generar series temporales sintéticas desde métricas de venta
#   =====
import pandas as pd
import numpy as np
from datetime import datetime

# Cargar métricas base
df = pd.read_csv("metricas_venta_integradas.csv")

# Crear rango de fechas semanales (52 semanas)
fechas = pd.date_range(start="2024-01-01", periods=52, freq="W")

# Generar dataset sintético
datos_sinteticos = []
for _, row in df.iterrows():
    for fecha in fechas:
        cantidad = max(0, np.random.normal(row["demanda_promedio"],
        ↪row["desviacion_demanda"]))
        datos_sinteticos.append({
            "fecha": fecha,
            "sku": row["sku"],
            "tienda": row["tienda"],
            "cantidad_vendida": round(cantidad, 2)
        })

```

```

ventas_sinteticas = pd.DataFrame(datos_sinteticos)
ventas_sinteticas.to_csv("ventas.csv", index=False)

print(" Archivo 'ventas.csv' generado correctamente con",
      len(ventas_sinteticas), "registros.")
print(ventas_sinteticas.head(10))

```

Archivo 'ventas.csv' generado correctamente con 31200 registros.

	fecha	sku	tienda	cantidad_vendida
0	2024-01-07	HM000001	TIENDA001	2.25
1	2024-01-14	HM000001	TIENDA001	0.00
2	2024-01-21	HM000001	TIENDA001	0.91
3	2024-01-28	HM000001	TIENDA001	3.78
4	2024-02-04	HM000001	TIENDA001	2.45
5	2024-02-11	HM000001	TIENDA001	1.81
6	2024-02-18	HM000001	TIENDA001	2.79
7	2024-02-25	HM000001	TIENDA001	2.52
8	2024-03-03	HM000001	TIENDA001	3.20
9	2024-03-10	HM000001	TIENDA001	1.18

1.3 Forecasting ARIMA (resultados_forecast.csv)

```

[43]: # =====
#   Pronóstico de demanda con ARIMA
#   =====

import warnings
warnings.filterwarnings('ignore')

from statsmodels.tsa.arima.model import ARIMA
import pandas as pd

# Leer las ventas sintéticas
df = pd.read_csv("ventas.csv")
df["fecha"] = pd.to_datetime(df["fecha"])

resultados = []

# Generar forecast por SKU-Tienda
for (sku, tienda), grupo in df.groupby(["sku", "tienda"]):
    grupo = grupo.sort_values("fecha")
    y = grupo["cantidad_vendida"].values

    # Validar longitud mínima
    if len(y) < 10 or y.sum() == 0:
        print(f" Sin datos suficientes para {sku} - {tienda}")
        continue

```

```

try:
    modelo = ARIMA(y, order=(1,1,1))
    modelo_fit = modelo.fit()
    pred = modelo_fit.forecast(steps=8)
    fechas_futuras = pd.date_range(start=grupo["fecha"].iloc[-1],
    ↪periods=8, freq="W")

    for i, f in enumerate(fechas_futuras):
        resultados.append({
            "sku": sku,
            "tienda": tienda,
            "fecha_predicha": f,
            "semana_a_futuro": i+1,
            "prediccion_ARIMA": round(float(pred[i]), 2)
        })
    print(f" Predicción generada para {sku} - {tienda}")
except Exception as e:
    print(f" Error en {sku}-{tienda}: {e}")

# Guardar resultados
df_forecast = pd.DataFrame(resultados)
df_forecast.to_csv("resultados_forecast.csv", index=False)
print("\n Archivo 'resultados_forecast.csv' generado correctamente.")
print("Filas:", len(df_forecast))
print(df_forecast.head(10))

```

```

Predicción generada para HM000001 - TIENDA001
Predicción generada para HM000001 - TIENDA002
Predicción generada para HM000001 - TIENDA003
Predicción generada para HM000001 - TIENDA004
Predicción generada para HM000001 - TIENDA005
Predicción generada para HM000001 - TIENDA006
Predicción generada para HM000002 - TIENDA001
Predicción generada para HM000002 - TIENDA002
Predicción generada para HM000002 - TIENDA003
Predicción generada para HM000002 - TIENDA004
Predicción generada para HM000002 - TIENDA005
Predicción generada para HM000002 - TIENDA006
Predicción generada para HM000003 - TIENDA001
Predicción generada para HM000003 - TIENDA002
Predicción generada para HM000003 - TIENDA003
Predicción generada para HM000003 - TIENDA004
Predicción generada para HM000003 - TIENDA005
Predicción generada para HM000003 - TIENDA006
Predicción generada para HM000004 - TIENDA001
Predicción generada para HM000004 - TIENDA002
Predicción generada para HM000004 - TIENDA003
Predicción generada para HM000004 - TIENDA004

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Predicción generada para HM000100 - TIENDA005

Predicción generada para HM000100 - TIENDA006

Archivo 'resultados_forecast.csv' generado correctamente.

Filas: 4800

	sku	tienda	fecha_predicha	semana_a_futuro	prediccion_ARIMA
0	HM0000001	TIENDA001	2024-12-29	1	1.60
1	HM0000001	TIENDA001	2025-01-05	2	1.77
2	HM0000001	TIENDA001	2025-01-12	3	1.75
3	HM0000001	TIENDA001	2025-01-19	4	1.75
4	HM0000001	TIENDA001	2025-01-26	5	1.75
5	HM0000001	TIENDA001	2025-02-02	6	1.75
6	HM0000001	TIENDA001	2025-02-09	7	1.75
7	HM0000001	TIENDA001	2025-02-16	8	1.75
8	HM0000001	TIENDA002	2024-12-29	1	1.39
9	HM0000001	TIENDA002	2025-01-05	2	1.50

1.4 Validación automática

```
[44]: # =====  
# Validar resultados del forecasting  
# =====  
import pandas as pd  
  
try:  
    df_forecast = pd.read_csv("resultados_forecast.csv")  
    print(f" Total filas: {len(df_forecast)}")  
    print(df_forecast.head(10))  
except Exception as e:  
    print(f" Error al leer el archivo: {e}")  
  
# Diagnóstico adicional (ventas totales)  
df_ventas = pd.read_csv("ventas.csv")  
resumen = df_ventas.groupby("sku")["cantidad_vendida"].sum().  
    ↪sort_values(ascending=False).head(10)  
print("\n Ventas totales (top 10 SKU):\n", resumen)
```

Total filas: 4800

	sku	tienda	fecha_predicha	semana_a_futuro	prediccion_ARIMA
0	HM0000001	TIENDA001	2024-12-29	1	1.60
1	HM0000001	TIENDA001	2025-01-05	2	1.77
2	HM0000001	TIENDA001	2025-01-12	3	1.75
3	HM0000001	TIENDA001	2025-01-19	4	1.75
4	HM0000001	TIENDA001	2025-01-26	5	1.75
5	HM0000001	TIENDA001	2025-02-02	6	1.75
6	HM0000001	TIENDA001	2025-02-09	7	1.75
7	HM0000001	TIENDA001	2025-02-16	8	1.75
8	HM0000001	TIENDA002	2024-12-29	1	1.39

```
9  HM000001  TIENDA002      2025-01-05      2      1.50
```

```
Ventas totales (top 10 SKU):
```

```
sku
HM000092    595.49
HM000035    592.27
HM000051    591.48
HM000100    578.14
HM000003    574.75
HM000006    574.66
HM000040    557.63
HM000029    554.03
HM000074    551.21
HM000009    549.94
```

```
Name: cantidad_vendida, dtype: float64
```

```
[ ]:
```