

Deep Learning

Time series

Uni-Variant / multi variant Season/trend/noise Working with time windows

Time Series

סדרת זמן היא רצף של נתונים שנאספים לאורך זמן במרווחים קבועים.

המאפיין המרכזי של סדרות זמן: הנתונים תלויים בזמן, כלומר הערכים מושפעים מהערכים הקודמים.

שימושים נפוצים:

- **תחזיות עסקיות** – מכירות חודשיות, ביקוש למוצרים.
- **כלכלה ופיננסים** – מחירי מניות, אינפלציה.
- **אקלימולוגיה** – טמפרטורה, גשם, זיהום אוויר.
- **תחבורה** – עומסי תנועה, זמני הגעה של רכבות.
- **רשתות חברתיות** – מגמות חיפושים, פוסטים פופולריים.



Time Series

1. סדרת זמן חד-משתנית (Univariate Time Series)

סדרה שיש בה רק משתנה אחד לאורך זמן.
המטרה היא לחזות את הערך הבא של אותו משתנה.

דוגמאות:

- טמפרטורה יומית $(\text{temperature}[t] \rightarrow \text{temperature}[t+1])$.
- מחירי מניות $(\text{stock_price}[t] \rightarrow \text{stock_price}[t+1])$.
- מספר הנוסעים בטיסות $(\text{passengers}[t] \rightarrow \text{passengers}[t+1])$.

2. סדרת זמן רב-משתנית (Multivariate Time Series)

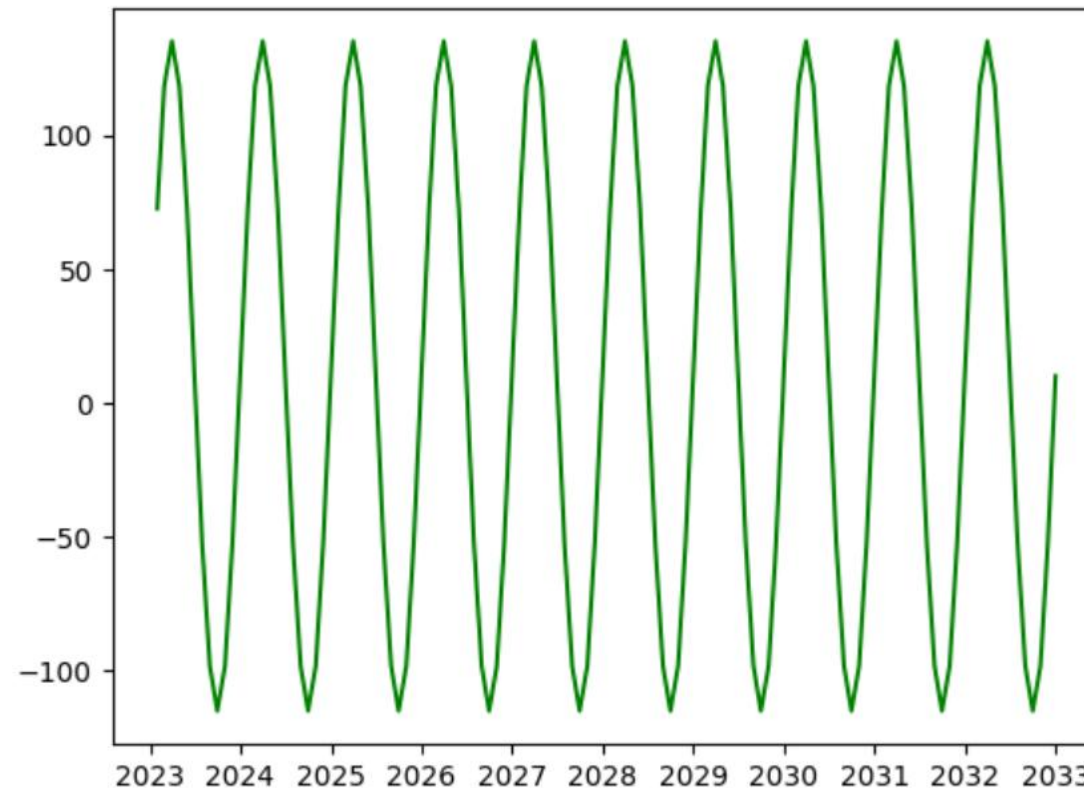
סדרה שיש בה כמה משתנים שמשפיעים על התחזית.
המטרה היא לחזות משתנה אחד או יותר בהתבסס על כמה משתנים.

דוגמאות:

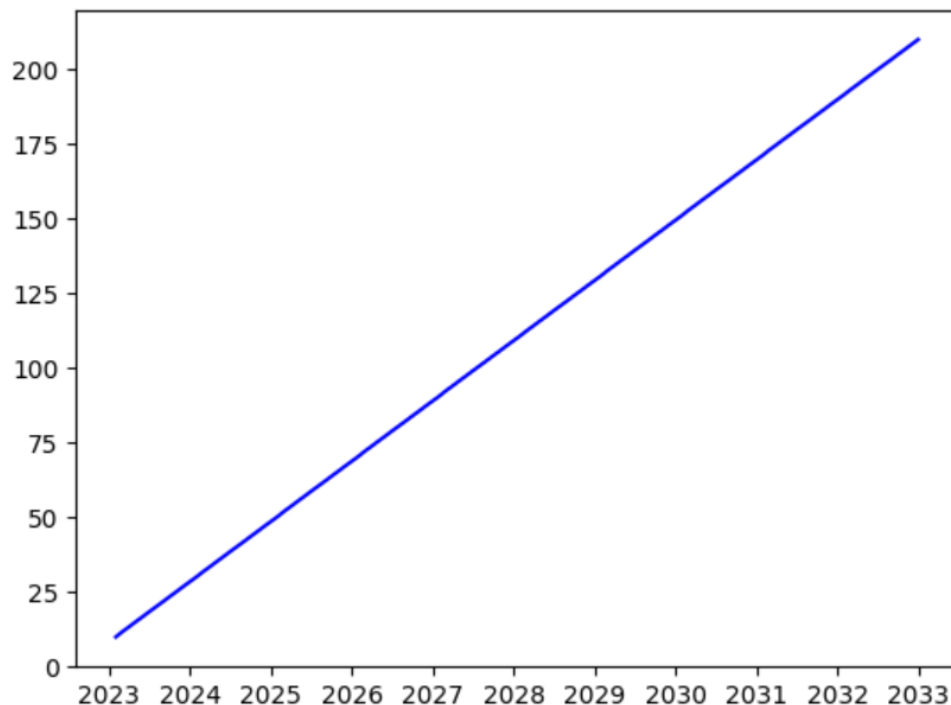
- חיזוי טמפרטורה לפי לחות ומהירות הרוח.
- חיזוי מחירי מניות לפי מחירי נפט וריבית בנקים.
- חיזוי עומס תחבורה לפי שעה, מזג אוויר, ואירועים מיוחדים.

עונתיות (Seasonality)

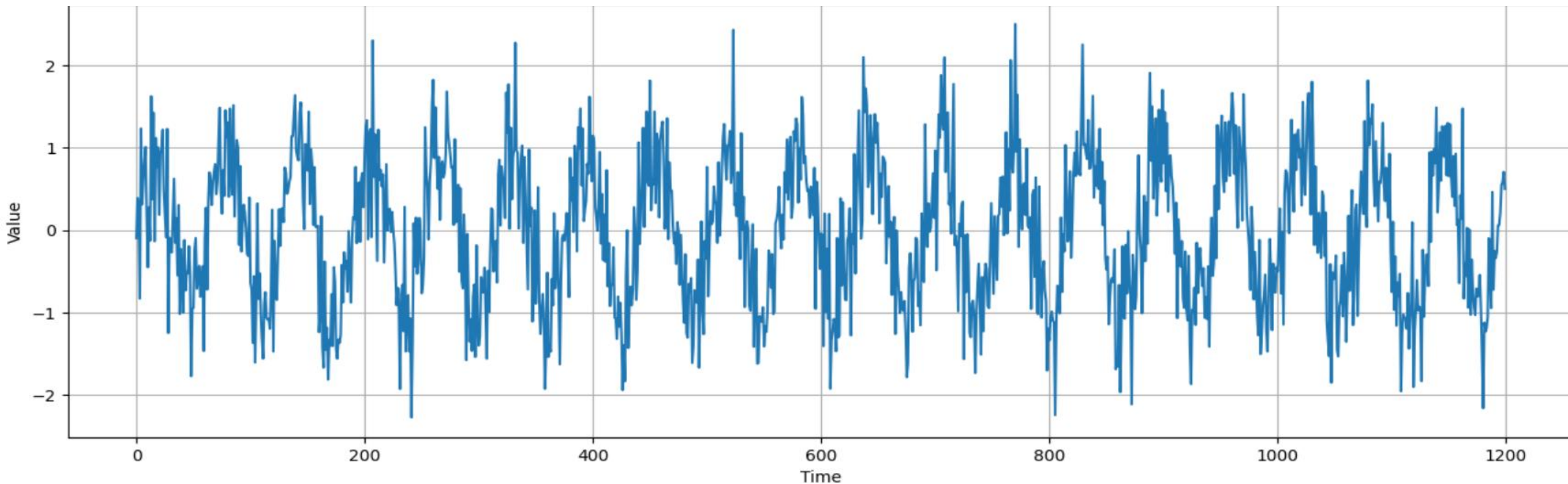
עונתיות היא תבנית מחזורית שחוזרת על עצמה בפרקי זמן קבועים. היא נגרמת מגורמים חיצוניים צפויים כמו עונות השנה, חגים, או ימים בשבוע. סדרות זמן עונתיות מאופיינות בכך שהתנהגות הנתונים בתקופה מסוימת דומה להתנהגות באותה תקופה בשנים או מחזורים קודמים.



טרנד הוא הכיוון הכללי של הנתונים לאורך זמן. הוא מתאר מגמת עלייה או ירידה מתמשכת, ללא קשר לשינויים מחזוריים קצרים. טרנד יכול להיות ליניארי, לוגאריטמי או בעל שיפוע משתנה, והוא משקף שינויים ארוכי טווח כמו צמיחה כלכלית, התחממות גלובלית, או ירידה באוכלוסייה.



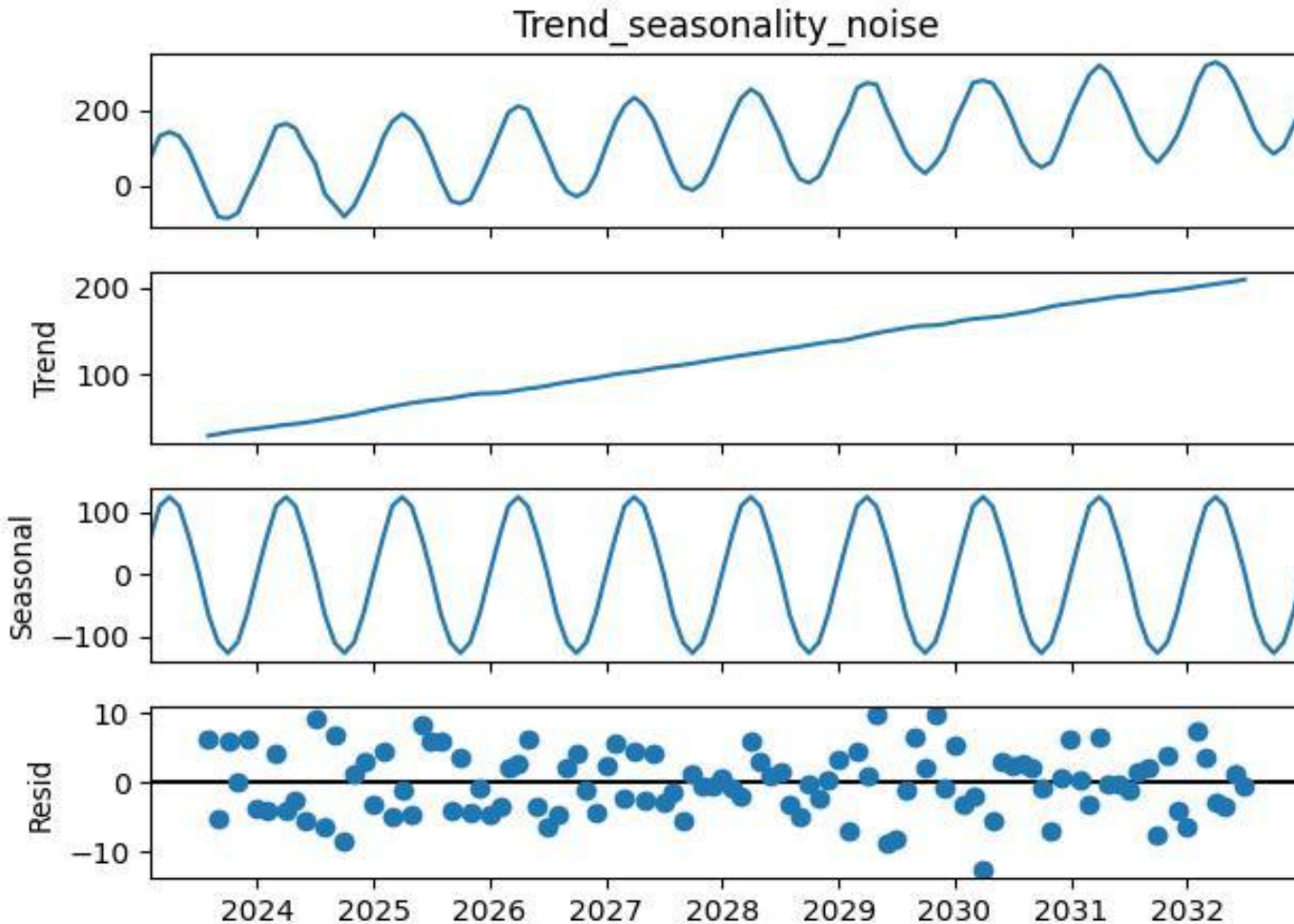
רעש הוא וריאציה אקראית בנתונים שאינה מוסברת על ידי עונתיות או טרנד. הוא נובע מגורמים חיצוניים לא צפויים או משגיאות במדידה, ויכול לגרום לסטיות פתאומיות בערכים. רעש מקשה על זיהוי תבניות משמעותיות בסדרת הזמן, ולעיתים מסננים אותו כדי לשפר את איכות התחזיות.



עונתיות (Seasonality)

```
from statsmodels.tsa.seasonal import seasonal_decompose

result = seasonal_decompose(df['Trend_seasonality_noise'],
                             model='additive', #we used +
                             period=12)#we simulate 12 months
```



פירוק סידרה למרכיביה:

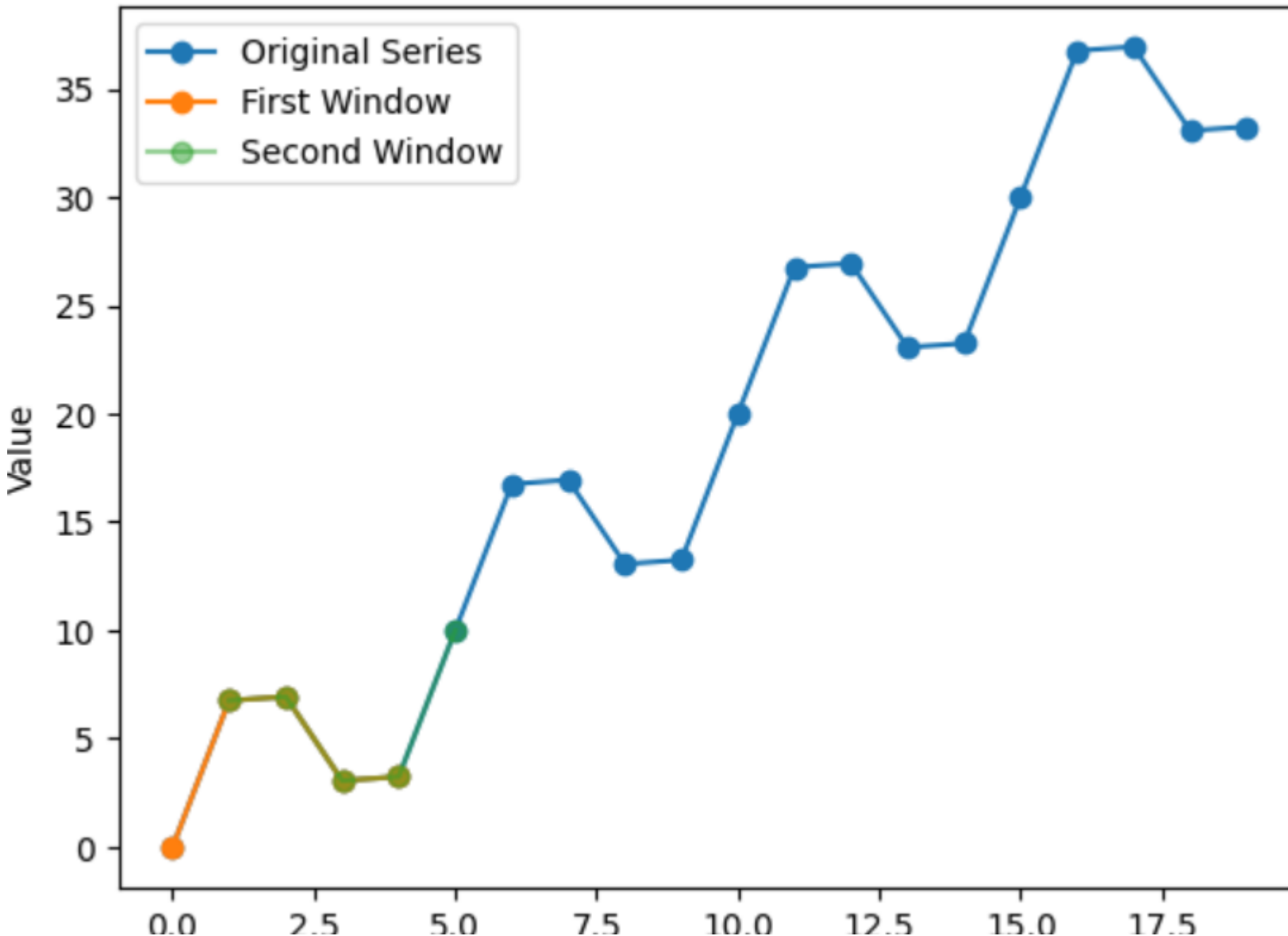
עונתיות

מגמה

ורעש

windows

Sliding Windows with Overlap



מחזיר את כל השלבים ברצף

- כאשר מוגדר

`return_sequences=False` (ברירת

המחדל), השכבה מחזירה רק את
השלב האחרון של ה-RNN, כלומר
וקטור יחיד לכל הרצף.

- כאשר `return_sequences=True`,

השכבה מחזירה את כל השלבים לאורך
הרצף, כלומר מטריצה שבה כל צעד
בזמן (timesteps) נשמר.

- שימושי כאשר מעבירים את הפלט

לשכבת RNN נוספת, למשל ב -

Stacked RNN / LSTM / GRU

windows

יצירה באמצעות

numpy

פשוט קל ונוח

מתאים לנתונים קטנים

```
# פונקציה לחלוקה לחלונות
def create_windows(series, window_size):
    X, y = [], []
    for i in range(len(series) - window_size):
        X.append(series[i:i+window_size]) # צעדים window_size של
        y.append(series[i+window_size]) # הערך הבא אחרי החלון
    return np.array(X), np.array(y)

# יצירת חלונות עם גודל 5
window_size = 5
X, y = create_windows(series, window_size)

# הצגת שני חלונות ראשונים
print(f"First Window: {X[0]} label: {y[0]}")
print(f"Second Window: {X[1]} label: {y[1]}")
```

windows

```
def windowed_dataset(series, window_size, batch_size, shuffle_buffer=1000):  
    dataset = tf.data.Dataset.from_tensor_slices(series)  
    dataset = dataset.window(window_size + 1, shift=1,  
                             drop_remainder=True) # יצירת חלונות עם חפיפה  
    dataset = dataset.flat_map(lambda window: window.batch(window_size + 1)) # הפיכת החלונות לרשימה אחת  
    dataset = dataset.map(lambda window: (window[:-1],  
                                         window[-1])) # (הערך הבא) y - ו (כל החלון) X הפרדת  
    dataset = dataset.shuffle(shuffle_buffer).batch(batch_size).prefetch(1) # עירבוב ושיפור ביצועים  
    return dataset  
  
# הגדרת גודל חלון ובאטץ'  
window_size = 5  
batch_size = 2 # אפשר לשנות כדי לראות איך זה משפיע  
  
# יצירת החלונות  
dataset = windowed_dataset(series, window_size, batch_size)
```

יצירה באמצעות tensorflow.data.Dataset מייצר תזרים (pipeline) נוח, מתאים לעבודה עם נתונים גדולים (big data) ומכין את הנתונים כטנסורים עבור המודל