<k/>

TIME SERIES FORECASTING, EXPLAINED

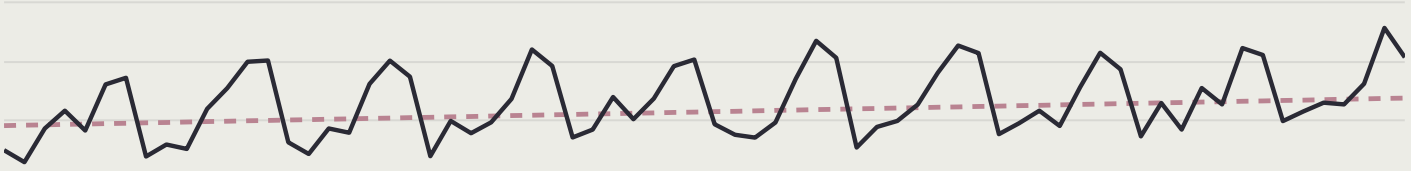
THEKHANSA.COM

Learn it with one example: the daily sales of a small cafe. Page 1: understand it. Page 2: do it.

1 What is a time series?

A list of numbers measured again and again over time, in order. Because the order matters, yesterday helps you predict today. Examples: daily sales, hourly temperature, weekly website visits, monthly rent.

Forecasting means using the past to estimate the next days, weeks or months, and saying how sure you are.



Daily cafe sales — The slow climb (trend)

2 Trend



The slow direction.

Example: the cafe gets popular, so sales climb all year.

3 Season



A pattern that repeats on the calendar.

Example: weekends are busy. Repeats every 7 days, so the period is 7.

4 Noise



Random ups and downs.

Example: a rainy Tuesday. You cannot forecast it, only measure how big it usually is.

5 Stationary: a steady series

A stationary series has no trend and the same wobble all the time, like a heart rate at rest. Models such as ARIMA need this, so we first remove the trend.



Before: sales climb, so it is not stationary.

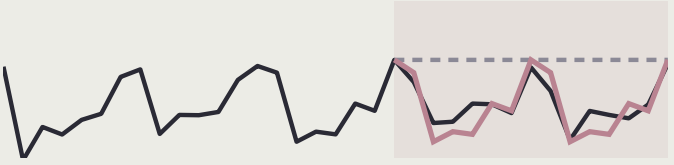


After `diff(7)`: this Monday minus last Monday. Steady.

Check it: ADF p-value below 0.05 and KPSS p-value above 0.05 both say "stationary". If they disagree, difference once more.

6 Beat the baseline first

Before any clever model, try a very simple guess. If your model cannot beat it, it is not worth using.



Real sales — Naive — Seasonal naive

Naive: tomorrow = today. **Seasonal naive:** this Saturday = last Saturday. The shaded part is the test: days the model never saw.

Example: on our cafe, seasonal naive misses by 5 sales a day. Naive misses by 15. A good model must beat 5.

7 Which model should you try?

Your situation	Try	Why, in plain words
Short history, no clear pattern	**Naive**, then **exponential smoothing**	Simple models are hard to beat when data is small.
One series with a weekly or yearly pattern	**Seasonal naive**, **ETS**, **SARIMAX**	They learn the repeating pattern directly.
You have extra clues: holidays, promos, price, weather	**Prophet** or a **machine learning** model	They accept extra columns, not just the past.
Hundreds of products at once	One shared **machine learning** model, or **StatsForecast**	One model learns from all series together.
Huge data, long forecasts	**Deep learning** (TFT, N-BEATS)	Only pays off with lots of data. Start simpler.

<k/>

NOW DO IT: MODELS, TESTS AND SCORES

In the code, `train` is the past and `test` is the last 28 days of the cafe series.

THEKHANSA.COM

8 Exponential smoothing (ETS)

Recent days count more than old days. Holt-Winters adds a trend and a repeating season. Fast and often very good.

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
m = ExponentialSmoothing(train, trend="add",
        seasonal="add", seasonal_periods=7).fit()
m.forecast(28)
```

9 ARIMA / SARIMAX

p = how many past days it looks at. **d** = how many times we subtract yesterday (removes trend). **q** = how many past mistakes it learns from. The seasonal set repeats this for the weekly pattern.

```
from statsmodels.tsa.statespace.sarimax import SARIMAX
m = SARIMAX(train, order=(1, 1, 1),
        seasonal_order=(1, 0, 1, 7)).fit()
m.get_forecast(28).summary_frame()  # mean + range
```

10 Prophet

Finds trend, weekly and yearly patterns by itself, and can add holidays. Needs two columns: **ds** (date) and **y** (value).

```
from prophet import Prophet
m = Prophet().fit(df)         # df: ds, y
m.predict(m.make_future_dataframe(28))
```

11 Machine learning

Give the model clues from the past, then let it learn. Always **shift first**, so a clue never contains the day you predict.

```
df["lag7"]  = df["y"].shift(7)
df["roll7"] = df["y"].shift(1).rolling(7).mean()
df["dow"]   = df.index.dayofweek
HistGradientBoostingRegressor().fit(X, y)
```

12 Test it honestly: backtest

One test is one roll of the dice. A backtest repeats it on several past windows: train on earlier days, test on later days.




```
from sklearn.model_selection import TimeSeriesSplit
for tr, te in TimeSeriesSplit(4, test_size=28).split(s):
    ...  # fit on s.iloc[tr], score on s.iloc[te]
```

13 Score it: a worked example

Day	Real	Forecast	Miss
Mon	100	95	5
Tue	120	125	5
Wed	90	100	10
Thu	110	105	5

MAE = average miss = (5+5+10+5) / 4 = **6.25** sales. Easy to explain.

RMSE punishes big misses more. **MAPE** = miss as a percent (here about 6%) but breaks when real values are near 0.

> **Best score for comparing models: MASE.** It is your error divided by the error of seasonal naive. Under 1 means you beat the baseline. 0.75 means 25% better.

14 Mistakes that fool you

Leakage	A clue that contains the day you predict. Example: a 7-day average that includes today. Scores look great, real forecasts fail.
Shuffling	A random split lets the model peek at the future. Split by time.
Tuning on test	Pick settings on the backtest, then touch the last window once.
One lucky score	Report several windows and the baseline next to your model.

15 Read the result

A forecast is a range, not a line. Say "about 140, probably between 130 and 150".

Check how often real values fall inside your 95% range. It should be close to 95 out of 100. Far lower means your range is too narrow. Leftover mistakes (residuals) should look like random noise.

```
from statsmodels.stats.diagnostic import acorr_ljungbox
acorr_ljungbox(m.resid, lags=[10])  # p > 0.05: ok
```

16 Your recipe, every time

Plot	Clean	Split	Baseline	Model	Backtest	Explain
See the series. Describe it in two sentences.	Dates, gaps, outliers.	By time. Keep a final test.	Naive and seasonal naive.	ETS, SARIMAX, ML or Prophet.	Several windows, score with MASE.	Show the range in plain words.

Full lessons, runnable code and 4 practice projects. Free.

thekhansa.com/learn/time-series