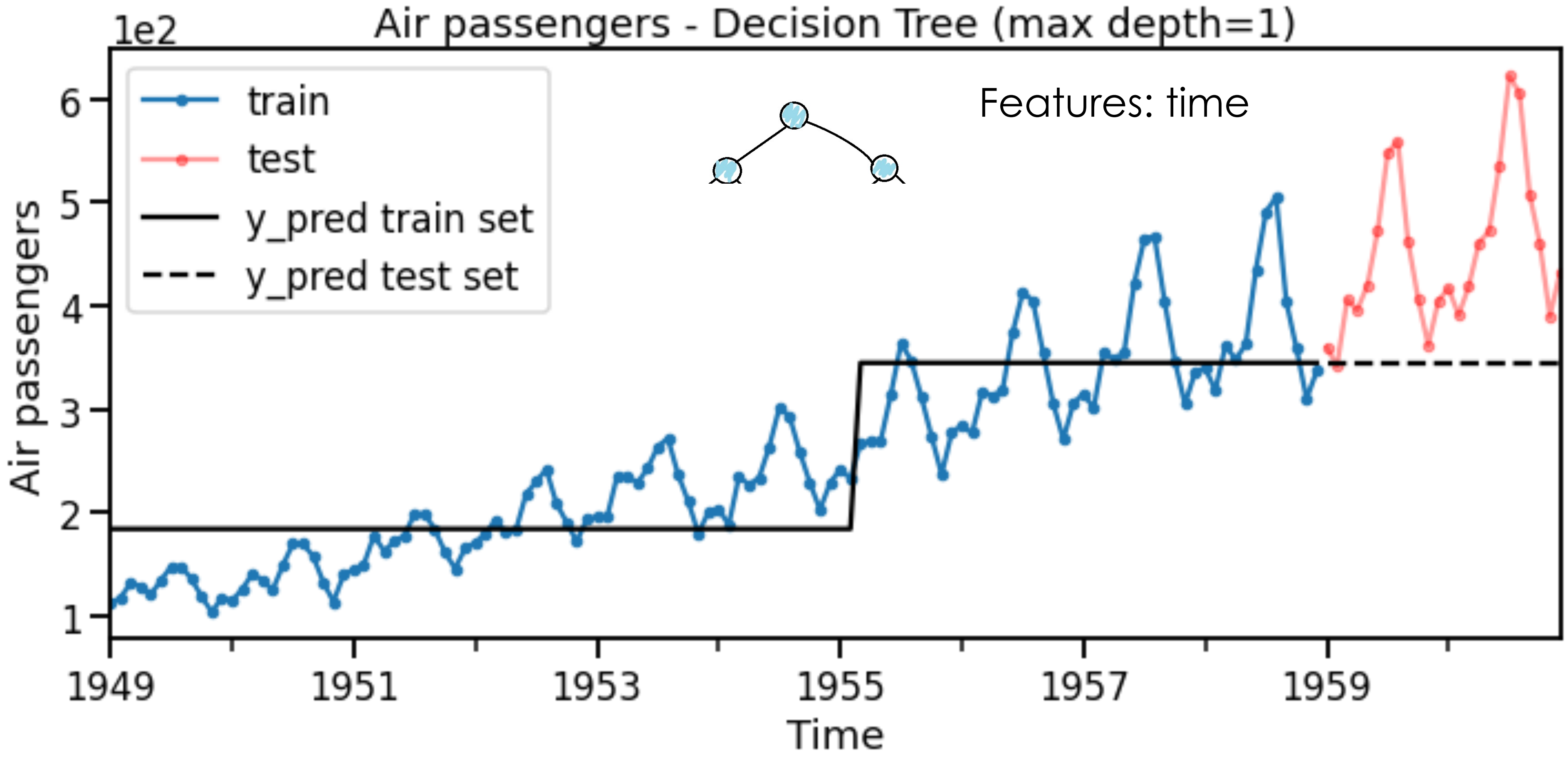
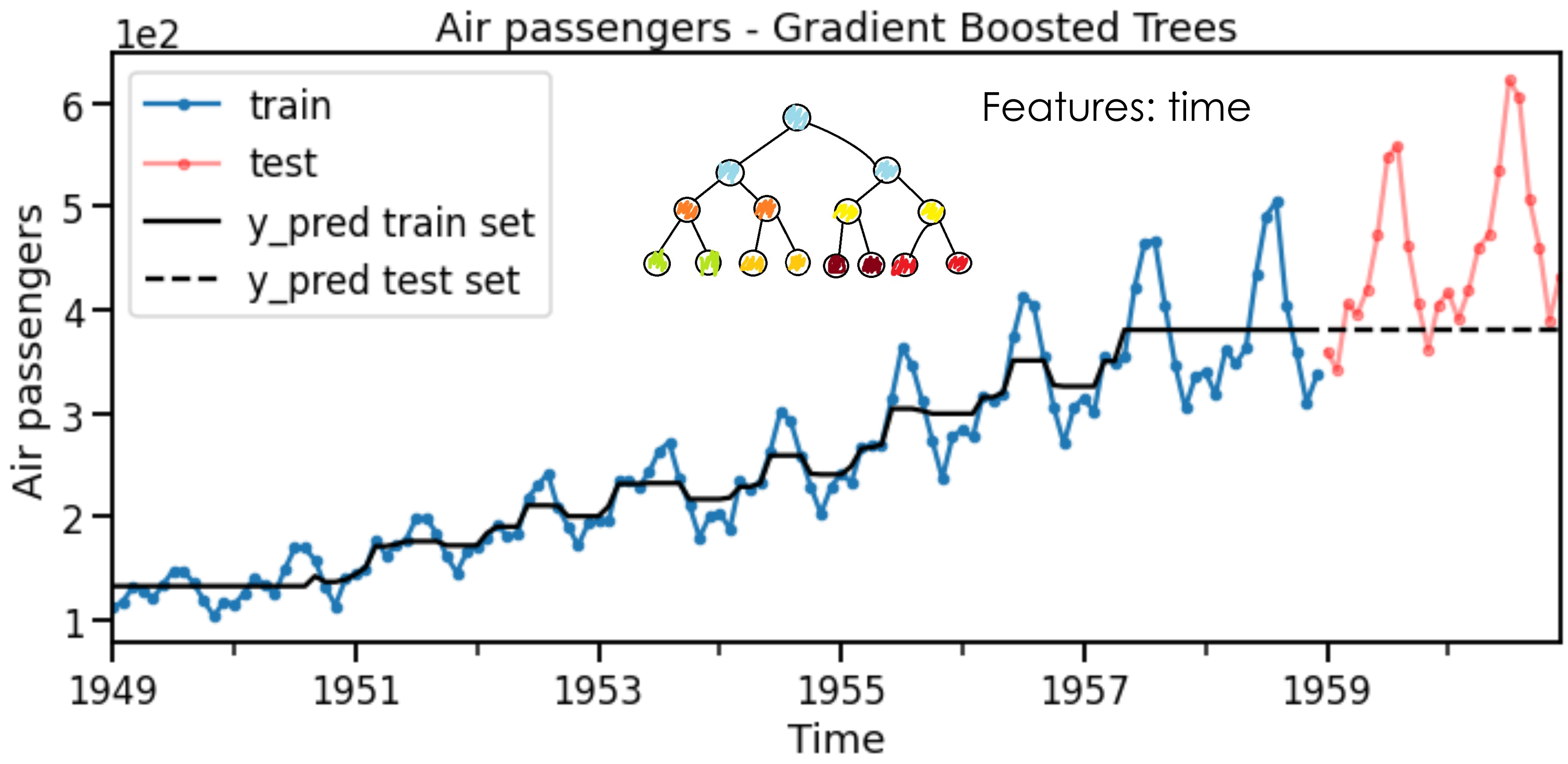
Tree-based models and trend

Trend features

Tree-based models cannot extrapolate



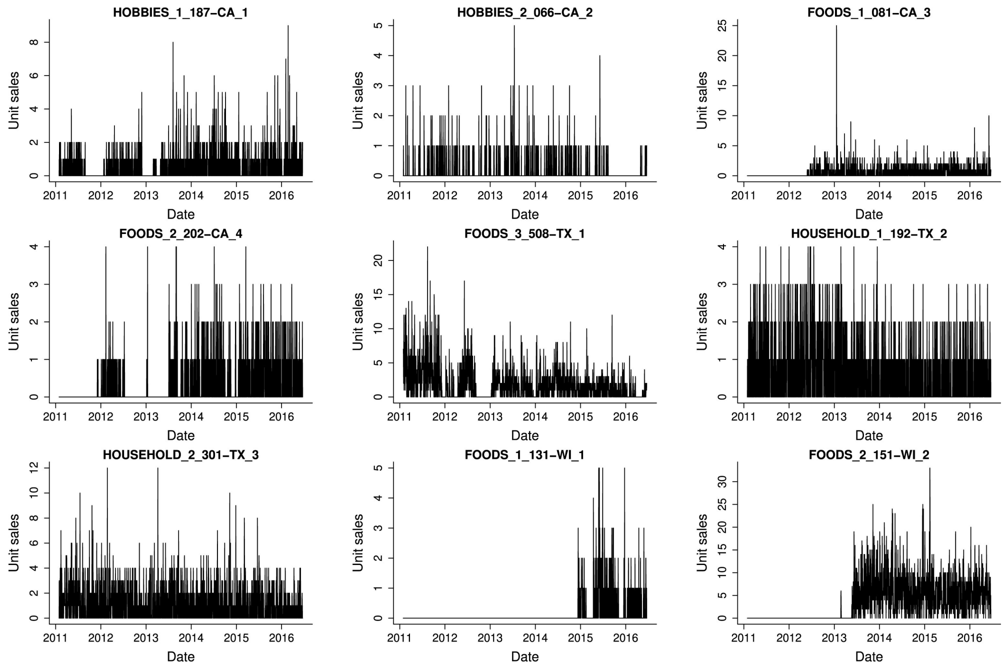
Tree-based models cannot extrapolate



How did they perform so well in forecasting competitions?

These datasets:

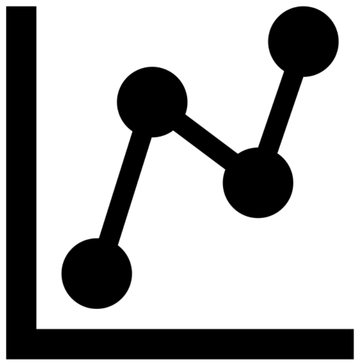
- Not much trend.
- Had multiple time series.
- Many categorical features (e.g., product category, country).
- Exogenous variables (price, promos, etc.).
- Trees are great at learning across these multiple data types.



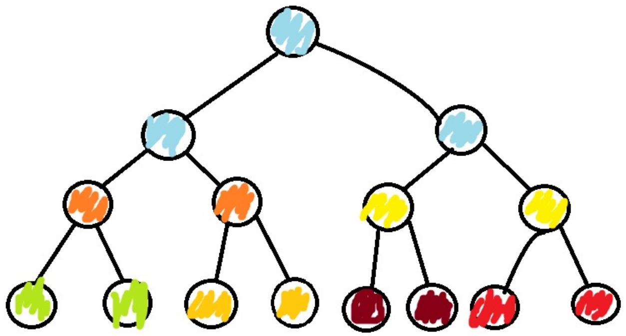
[1] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, organization, and implementation." *International Journal of Forecasting* (2021).

How to use Tree-based models if there is trend?

De-trend the time series first



Use more advanced tree algorithms

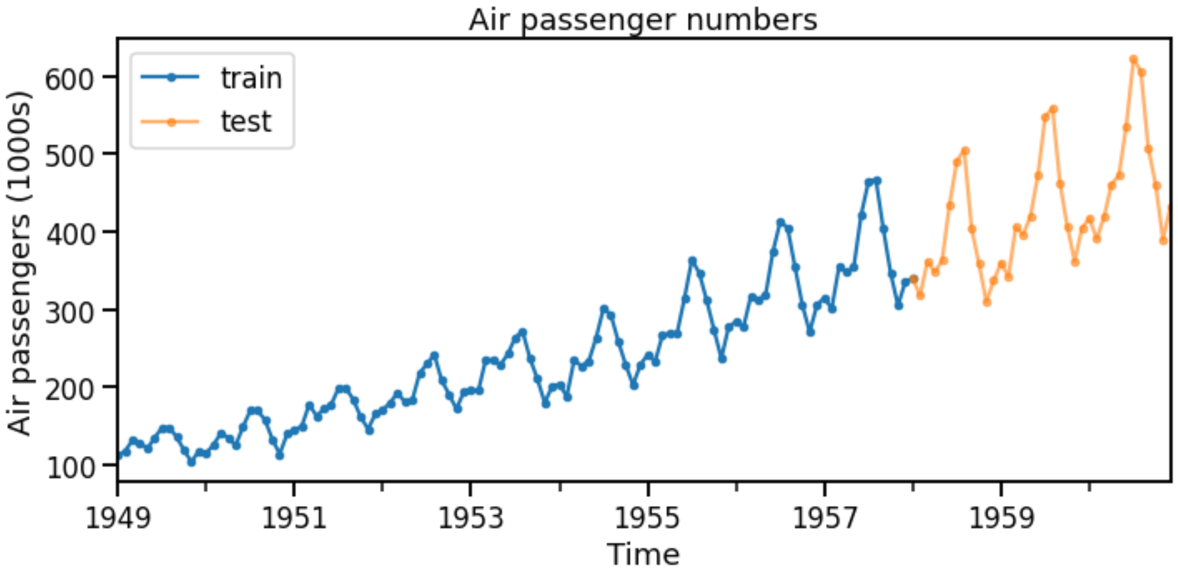


$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$ $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$

De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

E.g., $T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$



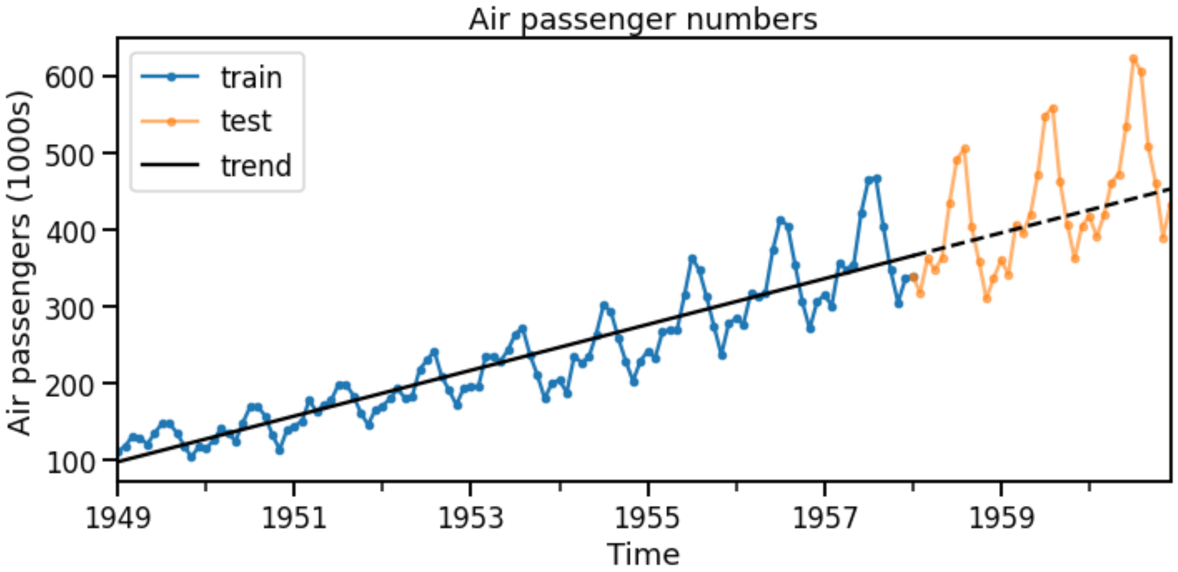
De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

$$\text{E.g., } T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$

2) De-trend the target variable:

$z_t = y_t - T_t$ **or** $z_t = y_t \div T_t$



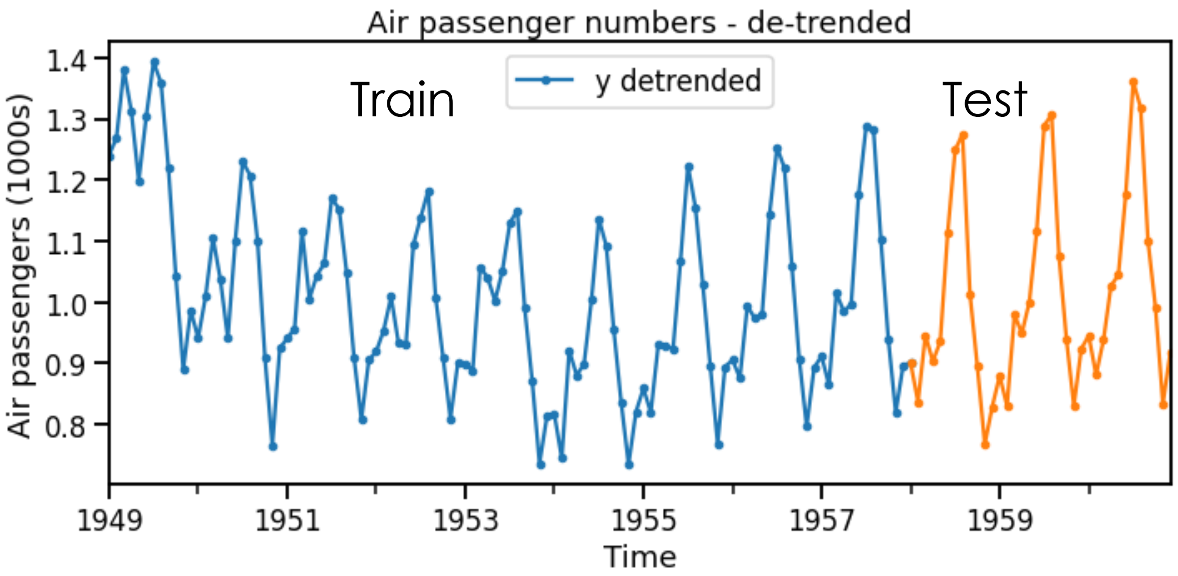
De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

$$\text{E.g., } T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$

2) De-trend the target variable:

$z_t = y_t - T_t$ **or** $z_t = y_t \div T_t$



De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

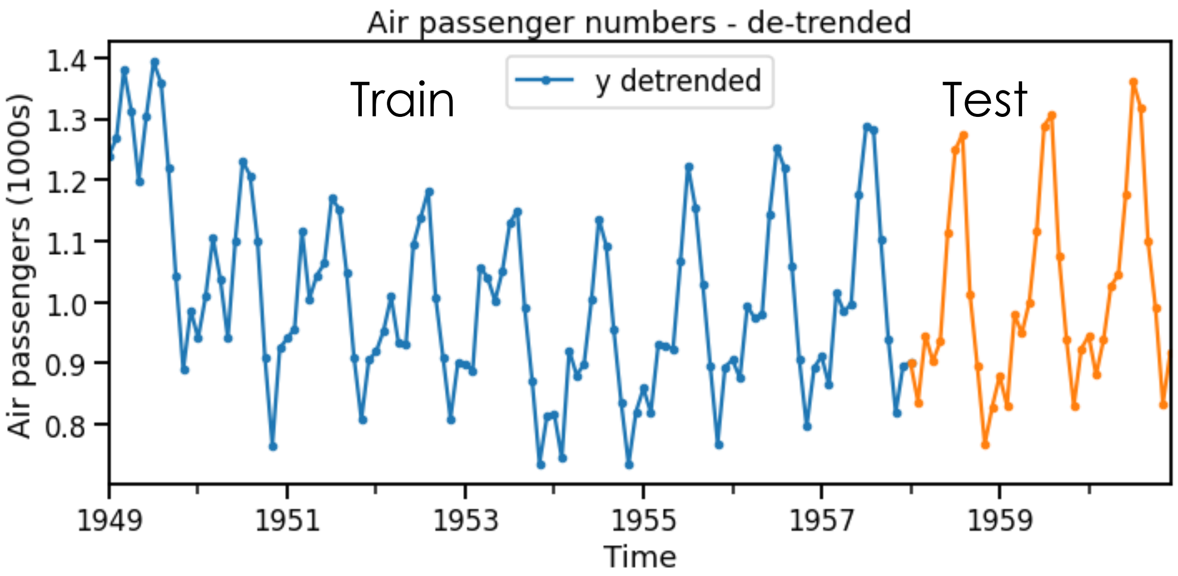
$$\text{E.g., } T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$

2) De-trend the target variable:

$$z_t = y_t - T_t \textbf{ or } z_t = y_t \div T_t$$

3) Build a forecast on z_t:

$$\hat{z}_{t+h} = Tree(z_t, X_{features})$$



De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

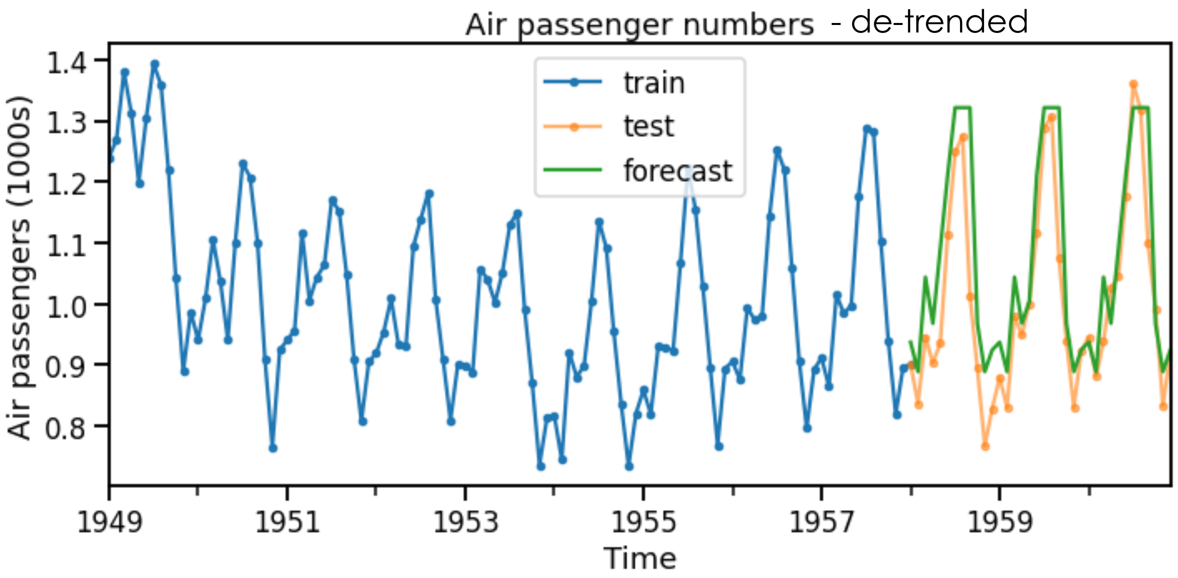
$$\text{E.g., } T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$

2) De-trend the target variable:

$$z_t = y_t - T_t \textbf{ or } z_t = y_t \div T_t$$

3) Build a forecast on z_t:

$$\hat{z}_{t+h} = Tree\left(z_t, X_{features}\right)$$



Features: time, lag 1 & 12

De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

$$\text{E.g., } T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$

2) De-trend the target variable:

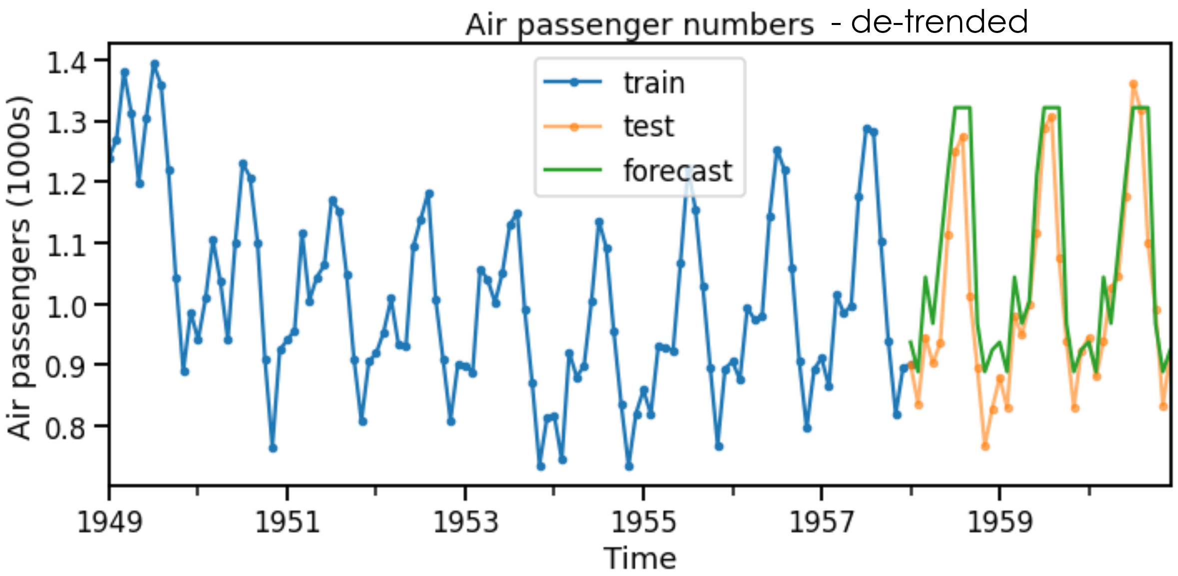
$$z_t = y_t - T_t \textbf{ or } z_t = y_t \div T_t$$

3) Build a forecast on z_t:

$$\hat{z}_{t+h} = Tree(z_t, X_{features})$$

4) Forecast the trend using any method:

$$\text{E.g., } \hat{T}_{t+h} = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$



Features: time, lag 1 & 12

De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

$$\text{E.g., } T_t = \beta_0 + \beta_1 t + \beta_2 t_2 + \cdots$$

2) De-trend the target variable:

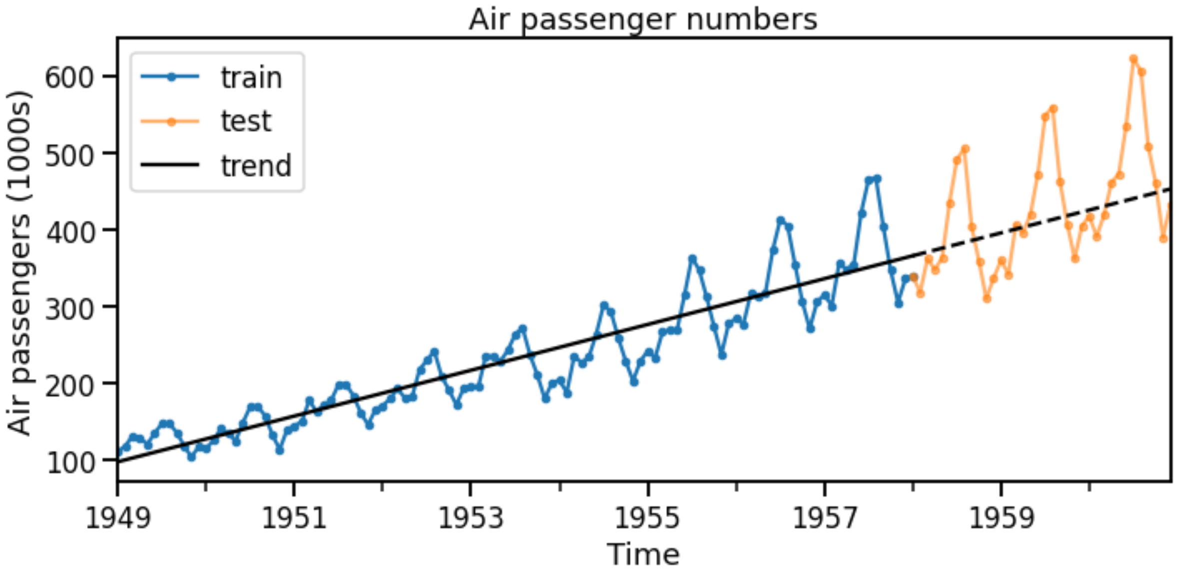
$$z_t = y_t - T_t \textbf{ or } z_t = y_t \div T_t$$

3) Build a forecast on z_t:

$$\hat{z}_{t+h} = Tree(z_t, X_{features})$$

4) Forecast the trend using any method:

$$\text{E.g., } \hat{T}_{t+h} = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$



Features: time, lag 1 & 12

De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

$$\text{E.g., } T_t = \beta_0 + \beta_1 t + \beta_2 t_2 + \cdots$$

2) De-trend the target variable:

$$z_t = y_t - T_t \textbf{ or } z_t = y_t \div T_t$$

3) Build a forecast on z_t:

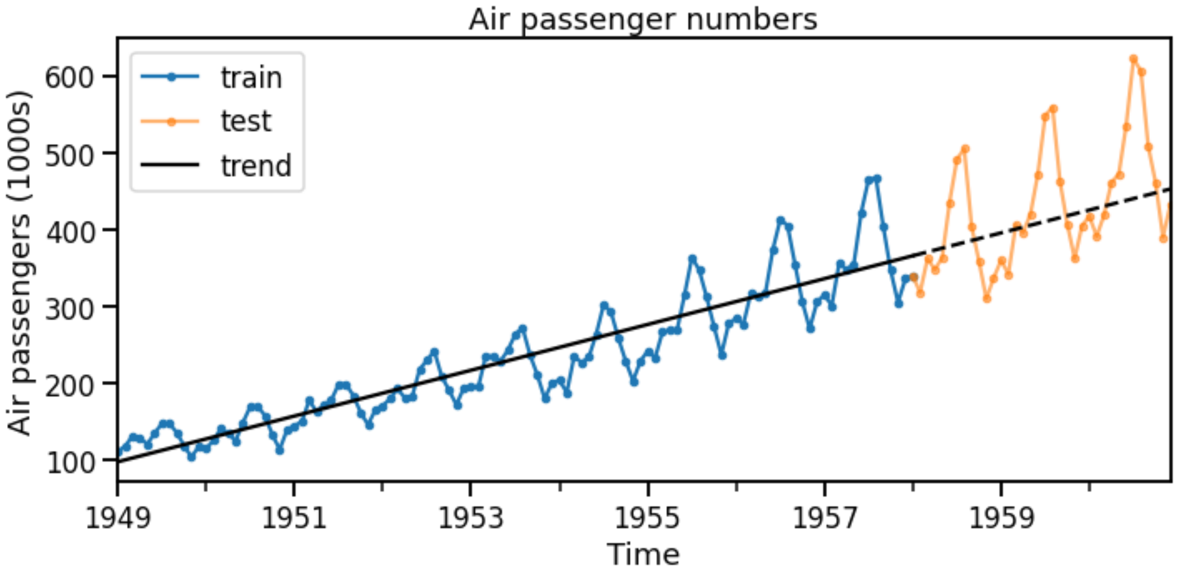
$$\hat{z}_{t+h} = Tree\left(z_t, X_{features}\right)$$

4) Forecast the trend using any method:

$$\text{E.g., } \hat{T}_{t+h} = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$$

5) Add the trend forecast to the de-trended forecast:

$$\hat{y}_{t+h} = \hat{z}_{t+h} + \hat{T}_{t+h} \textbf{ or } \hat{y}_{t+h} = \hat{z}_{t+h} \times \hat{T}_{t+h}$$



Features: time, lag 1 & 12

De-trending the time series for tree based models

1) Estimate the trend of y_t using any method:

E.g., $T_t = \beta_0 + \beta_1 t + \beta_2 t_2 + \cdots$

2) De-trend the target variable:

$z_t = y_t - T_t$ **or** $z_t = y_t \div T_t$

3) Build a forecast on z_t:

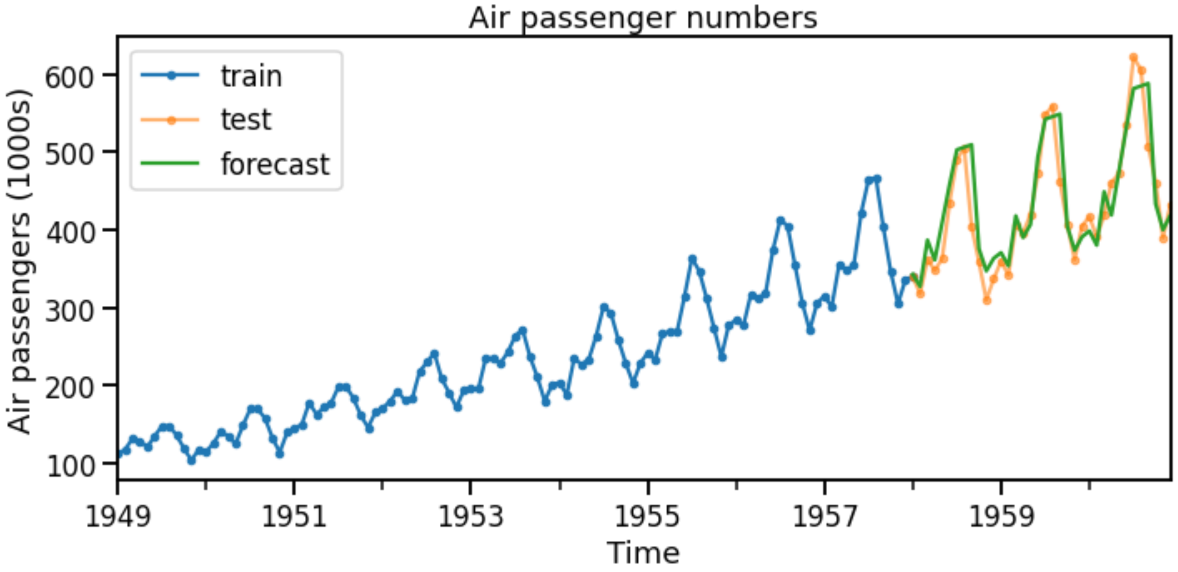
$\hat{z}_{t+h} = Tree(z_t, X_{features})$

4) Forecast the trend using any method:

E.g., $\hat{T}_{t+h} = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots$

5) Add the trend forecast to the de-trended forecast:

$\hat{y}_{t+h} = \hat{z}_{t+h} - \hat{T}_{t+h}$ **or** $\hat{y}_{t+h} = \hat{z}_{t+h} \times \hat{T}_{t+h}$



Features: time, lag 1 & 12

Pros and cons

Pros

After adjusting the target we use the same forecasting workflow.

The additional trend forecast is a new source for error.

Modelling a non-linear trend is harder.

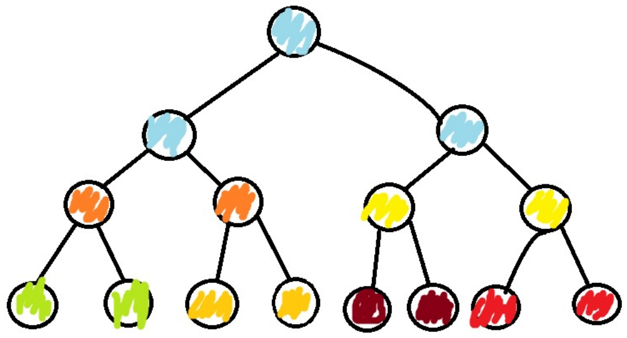
Cons

More advanced tree algorithms

Papers & blogs:
- https://towardsdatascience.com/xgboost-for-timeseries-lightgbm-is-a-bigger-boat-197864013e88
- https://arxiv.org/pdf/2009.09110.pdf
- https://arxiv.org/pdf/2211.08661.pdf

Code:
- https://lightgbm.readthedocs.io/en/latest/Parameters.html#linear_tree
- https://github.com/cerlymarco/linear-tree (warning: small/no community, no unit tests)

Use more advanced tree algorithms

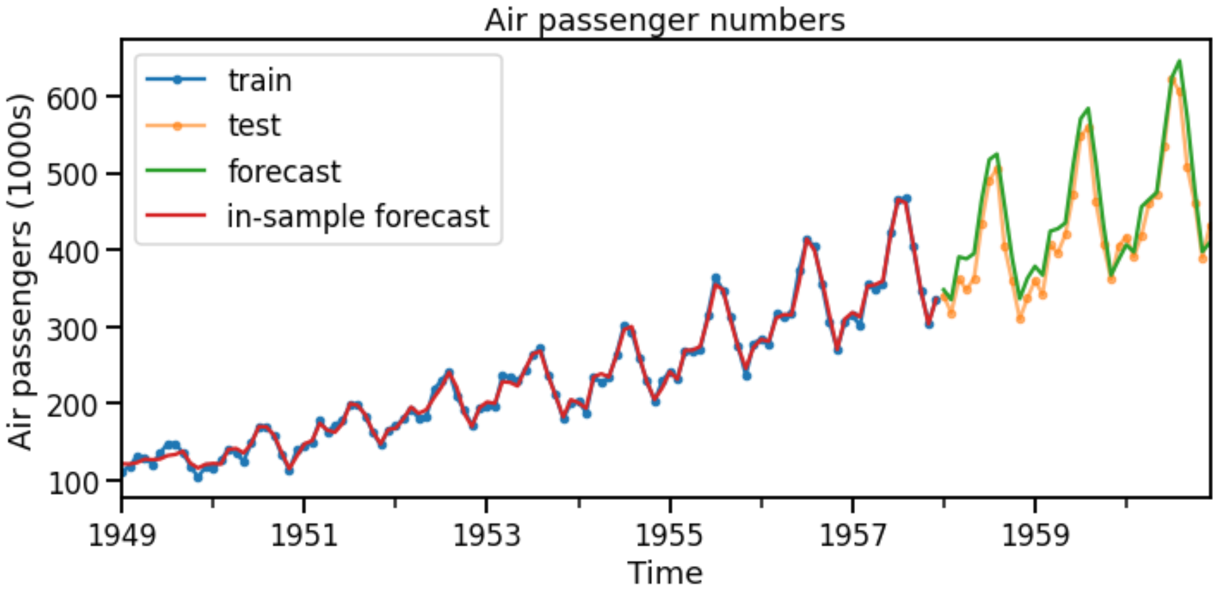


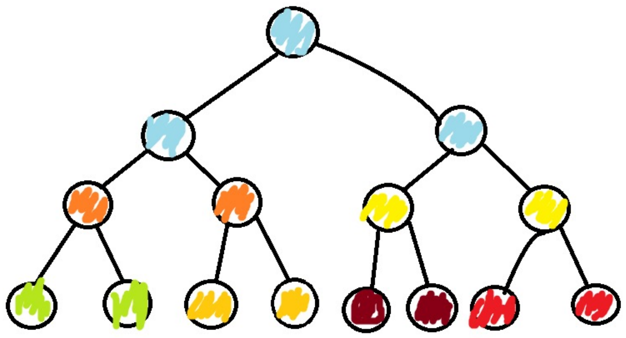
$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$ $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$

LightGBM with linear trees

```
from lightgbm import LGBMRegressor
```

```
# Define the model.
model = LGBMRegressor(linear_tree=True)
```





$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$ $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$

Features:

- Time
- lag 1, 2, 3, & 12
- Window mean of size 12

Summary

Tree-based models cannot extrapolate and so will struggle with trend.

De-trending the time series is one option to overcome this.

Linear trees fit a linear model at the leaves and can handle trend.