Recursive forecasting with trend features

Trend features

Recursive forecasting: features built with target

Time	y	Lag 1 y	Rolling mean y
2020-02-12	23	NaN	NaN
2020-02-13	30	23	NaN
2020-02-14	35	30	NaN
2020-02-15	30	35	29.3
2020-02-16		30	31.7
2020-02-17			
2020-02-18			

Recursive forecasting: features built with target

Time	y	Lag 1 y	Rolling mean y
2020-02-12	23	NaN	NaN
2020-02-13	30	23	NaN
2020-02-14	35	30	NaN
2020-02-15	30	35	29.3
2020-02-16		30	31.7
2020-02-17			
2020-02-18			

Impute or **drop** the missing values at the start of the time series.

Recursive forecasting: features built with target

Time	y	Lag 1 y	Rolling mean y
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16		30	31.7
2020-02-17			
2020-02-18			

Impute or **drop** the missing values at the start of the time series.

Recursive forecasting: features built with target

	Target y	Features X	
Time	y	Lag 1 y	Rolling mean y
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16		30	31.7
2020-02-17			
2020-02-18			

model.fit(X_train, y_train)

model.predict(X_pred)

Recursive forecasting: features built with target

Target **y** | Features **X**

Time	y	Lag 1 y	Rolling mean y
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16		30	31.7
2020-02-17			
2020-02-18			

model.fit(X_train, y_train)

model.predict([30, 31.7])

$\hat{y}_{T+1}$

Recursive forecasting: features built with target

Target **y** | Features **X**

Time	y	Lag 1 y	Rolling mean y
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17			
2020-02-18			

model.fit(X_train, y_train)

model.predict([30, 31.7])

Recursive forecasting: features built with target

	Target **y**	Features **X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17		$\hat{y}_{T+1}$	
2020-02-18			

model.fit(X_train, y_train)

Recursive forecasting: features built with target

	Target **y**	Features **X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17		$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1} + 30 + 35}{3}$
2020-02-18			

model.fit(X_train, y_train)

Recursive forecasting: features built with target

	Target **y**	Features **X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17		$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1} + 30 + 35}{3}$
2020-02-18			

model.fit(X_train, y_train)

model.predict($[\hat{y}_{T+1}, \ldots]$)

$\hat{y}_{T+2}$

Recursive forecasting: features built with target

	Target	Features	
	y	**X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17	$\hat{y}_{T+2}$	$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1} + 30 + 35}{3}$
2020-02-18			

model.fit(X_train, y_train)

Recursive forecasting: features built with target

	Target **y**	Features **X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17	$\hat{y}_{T+2}$	$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1} + 30 + 35}{3}$
2020-02-18		$\hat{y}_{T+2}$	

model.fit(X_train, y_train)

Recursive forecasting: features built with target

	Target **y**	Features **X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17	$\hat{y}_{T+2}$	$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1} + 30 + 35}{3}$
2020-02-18		$\hat{y}_{T+2}$	$\frac{\hat{y}_{T+1} + \hat{y}_{T+2} + 35}{3}$

model.fit(X_train, y_train)

Recursive forecasting: features built with target

	Target **y**	Features **X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**
2020-02-12	23	23	29.3
2020-02-13	30	23	29.3
2020-02-14	35	30	29.3
2020-02-15	30	35	29.3
2020-02-16	$\hat{y}_{T+1}$	30	31.7
2020-02-17	$\hat{y}_{T+2}$	$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1}+30+35}{3}$
2020-02-18		$\hat{y}_{T+2}$	$\frac{\hat{y}_{T+1}+\hat{y}_{T+2}+35}{3}$

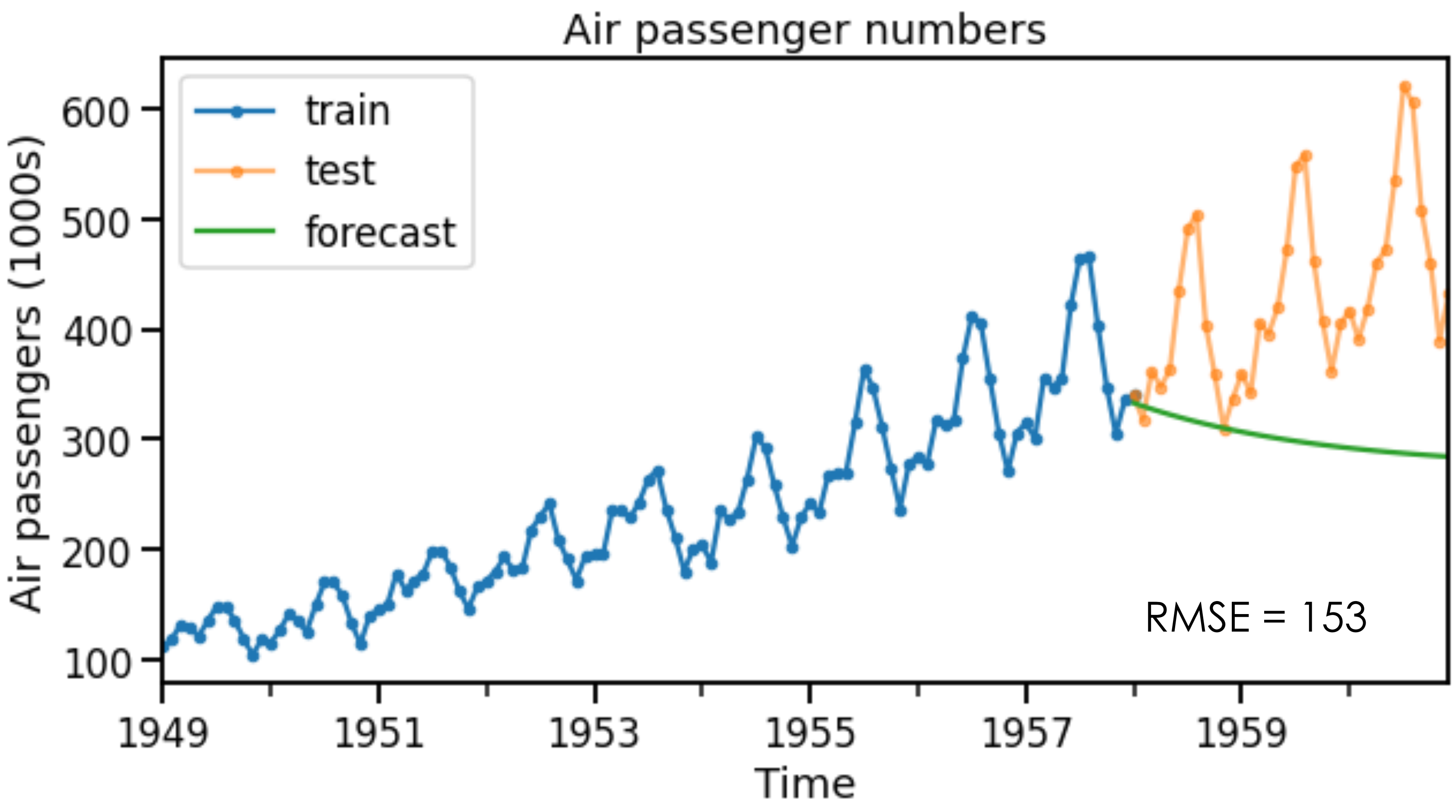
model.fit(X_train, y_train)

model.predict([$\hat{y}_{T+1}$, …])

$$\Downarrow$$

$\hat{y}_{T+3}$

Example: Linear regression



Features:

- Lag y of 1

Why does it decay?

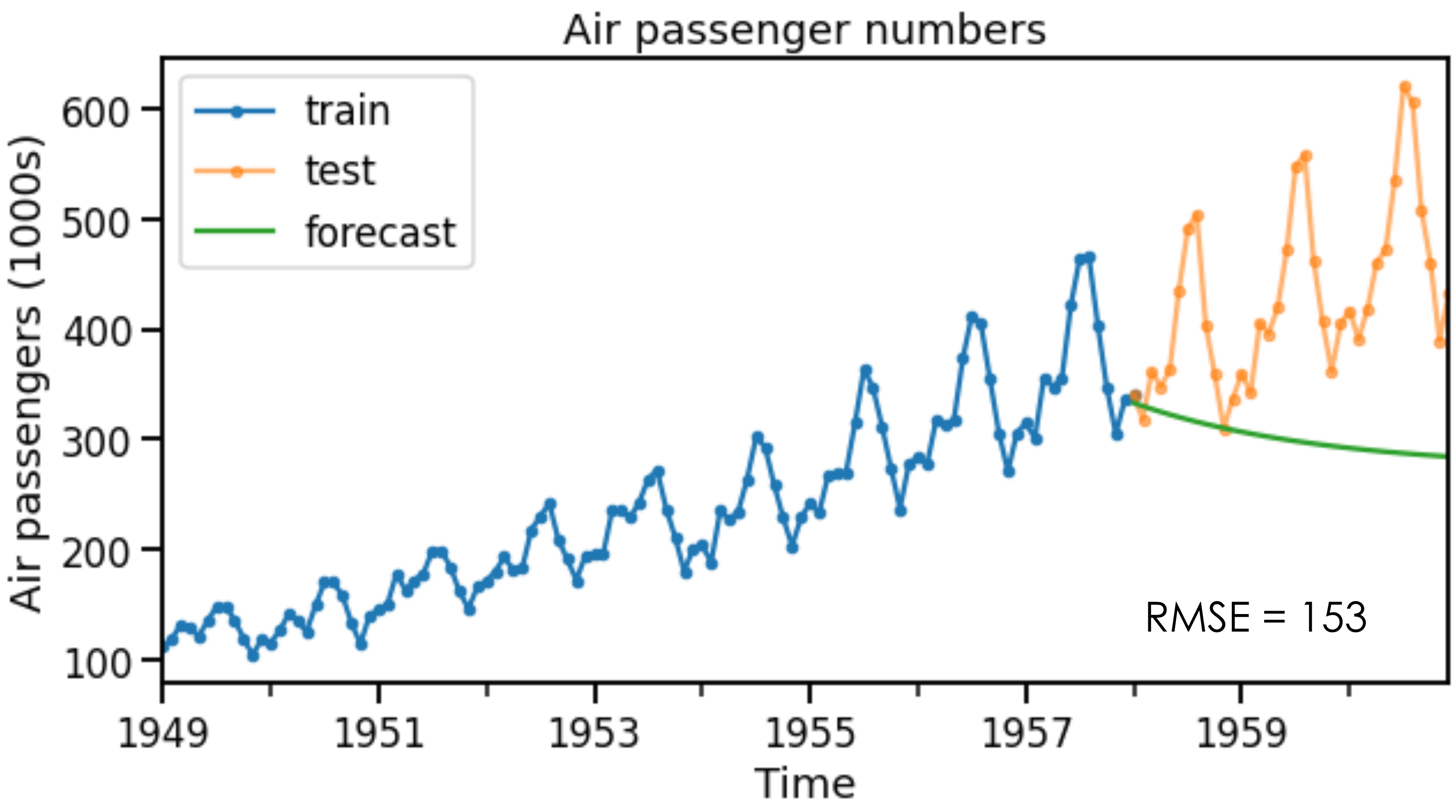
$$y_{t+1} = \beta_0 + \beta_1 y_t$$

Consider a simple case

$$y_{t+1} = \beta_1 y_t$$

If β_1< 1 then y_{t+1} will decay exponentially as we recursively iterate forward in time.

Example: Linear regression



Features:
- Lag y of 1

Why does it decay?

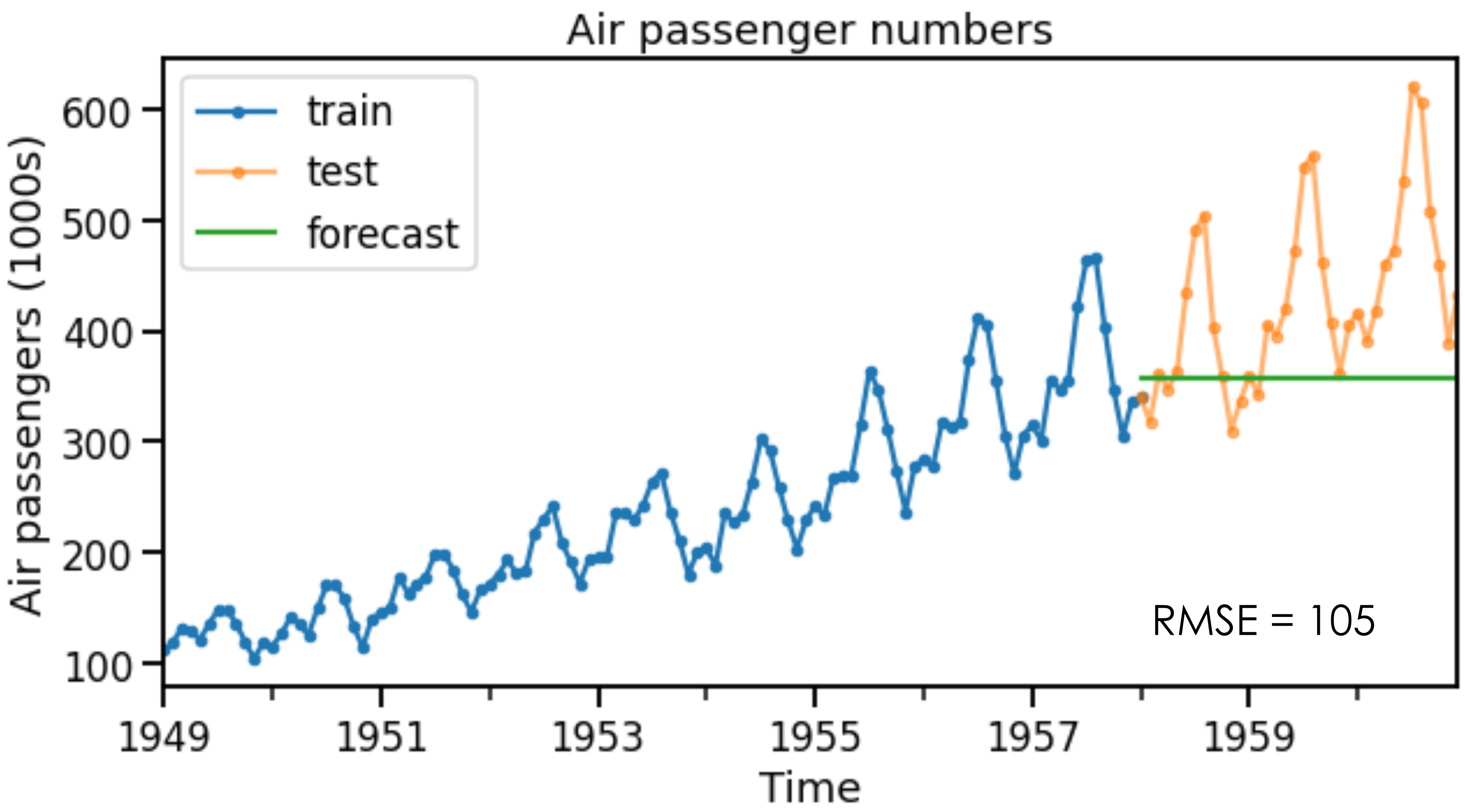
$$y_{t+1} = \beta_0 + \beta_1 y_t$$

Consider a simple case

$$y_{t+1} = \beta_1 y_t$$

If $\beta_1 > 1$ then y_{t+1} will grow exponentially as we recursively iterate forward in time.

Example: Gradient Boosted Trees



Features:

- Lag y of 1

Recursive forecasting: lags & window & time

Time	y	Lag 1 y	Rolling mean y	t (days)
2020-02-12	23	23	29.3	0
2020-02-13	30	23	29.3	1
2020-02-14	35	30	29.3	2
2020-02-15	30	35	29.3	3
2020-02-16		30	31.7	4
2020-02-17				5
2020-02-18				6

Recursive forecasting: lags & window & time

	Target **y**		Features **X**	
Time	**y**	**Lag 1 y**	**Rolling mean y**	***t* (days)**
2020-02-12	23	23	29.3	0
2020-02-13	30	23	29.3	1
2020-02-14	35	30	29.3	2
2020-02-15	30	35	29.3	3
2020-02-16		30	31.7	4
2020-02-17				5
2020-02-18				6

model.fit(X_train, y_train)

model.predict(X_pred)

Recursive forecasting: lags & window & time

	Target **y**	Features **X**		
Time	**y**	**Lag 1 y**	**Rolling mean y**	***t* (days)**
2020-02-12	23	23	29.3	0
2020-02-13	30	23	29.3	1
2020-02-14	35	30	29.3	2
2020-02-15	30	35	29.3	3
2020-02-16		30	31.7	4
2020-02-17				5
2020-02-18				6

model.fit(X_train, y_train)

model.predict(X_pred)

$$\hat{y}_{T+1}$$

Recursive forecasting: lags & window & time

	Target **y**	Features **X**		
Time	**y**	**Lag 1 y**	**Rolling mean y**	***t* (days)**
2020-02-12	23	23	29.3	0
2020-02-13	30	23	29.3	1
2020-02-14	35	30	29.3	2
2020-02-15	30	35	29.3	3
2020-02-16	$\hat{y}_{T+1}$	30	31.7	4
2020-02-17		$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1} + 30 + 35}{3}$	5
2020-02-18				6

model.fit(X_train, y_train)

model.predict(X_pred)

Recursive forecasting: lags & window & time

Target **y** | Features **X**

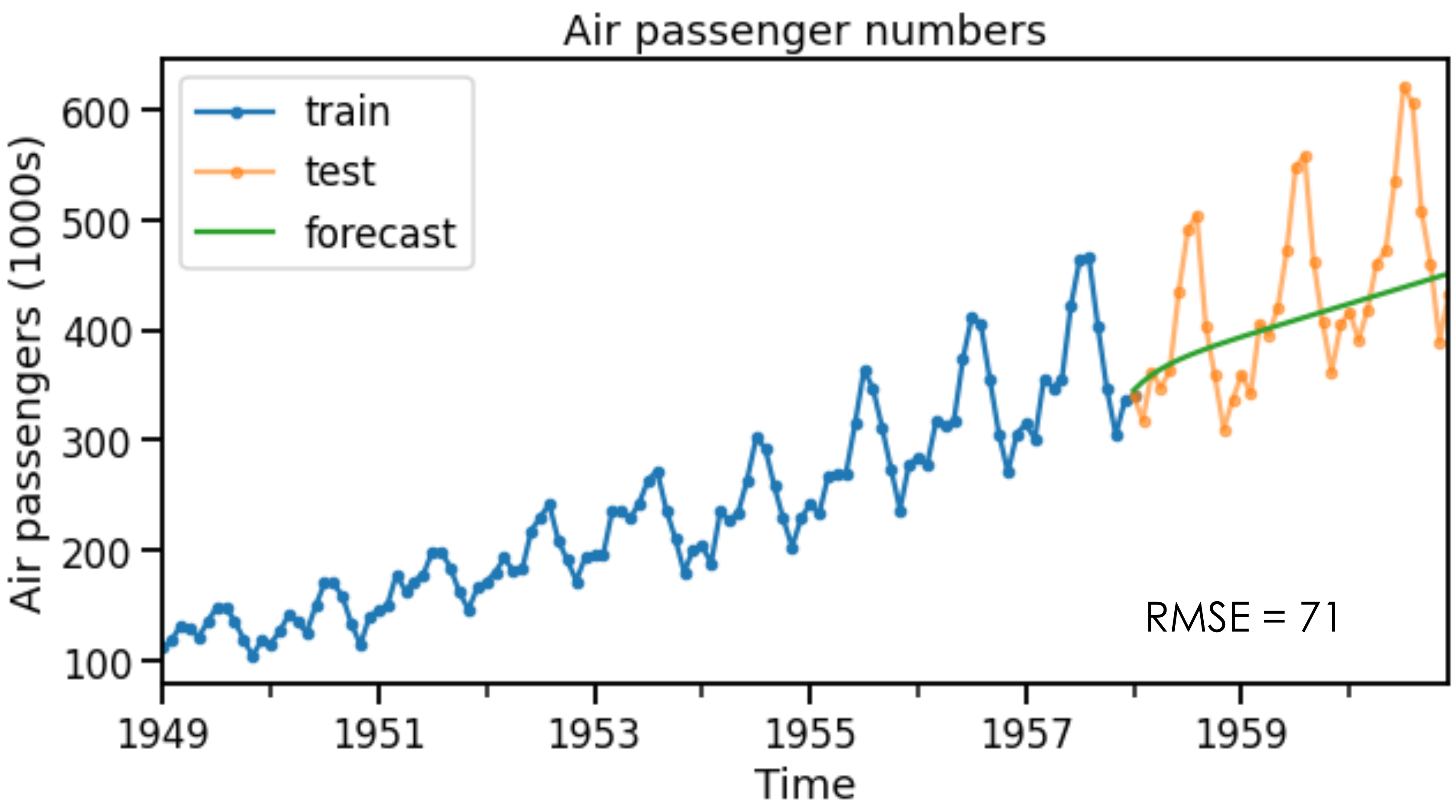
Time	y	Lag 1 y	Rolling mean y	t (days)
2020-02-12	23	23	29.3	0
2020-02-13	30	23	29.3	1
2020-02-14	35	30	29.3	2
2020-02-15	30	35	29.3	3
2020-02-16	$\hat{y}_{T+1}$	30	31.7	4
2020-02-17		$\hat{y}_{T+1}$	$\frac{\hat{y}_{T+1} + 30 + 35}{3}$	5
2020-02-18				6

model.fit(X_train, y_train)

model.predict(X_pred)

$\hat{y}_{T+2}$

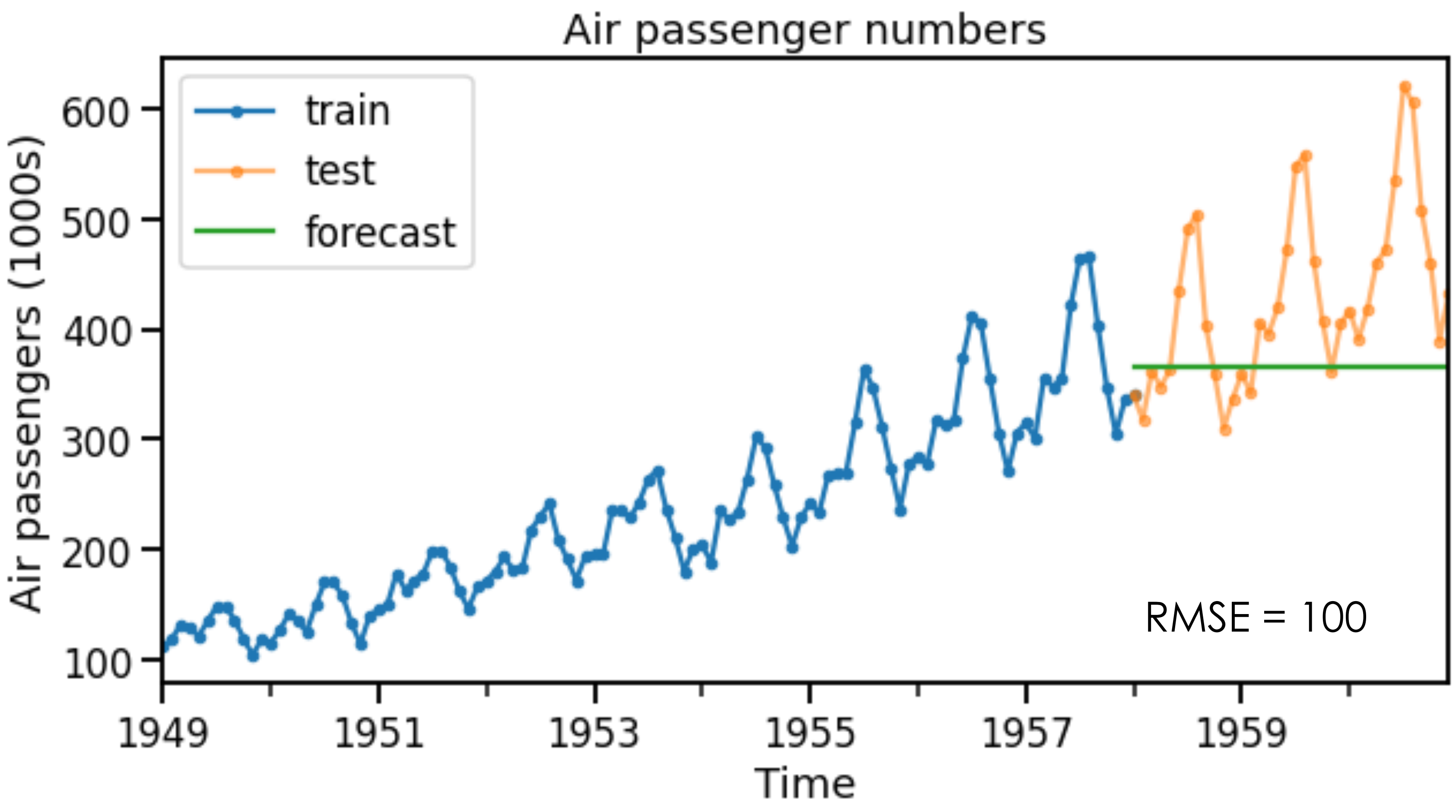
Example: Linear regression



Features:

- Lag y of 1
- Time since start (t)

Example: Gradient Boosted Trees



Features:

- Lag y of 1
- Time since start (t)