# Autoregressive Models

## Concept Module 15

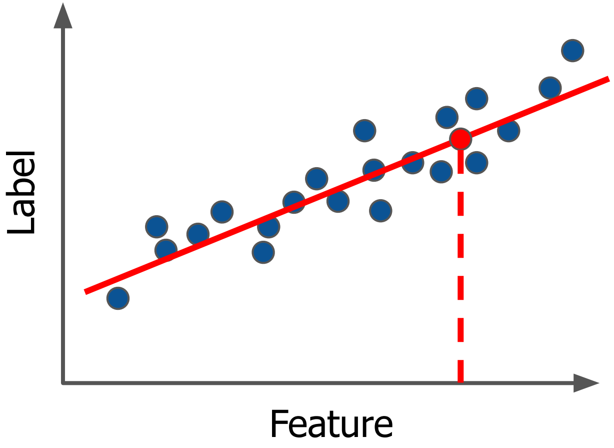
Template by: Laurent Lessard

# Data science with time series

- We saw some basic unsupervised learning:
  - Visualizing time series at different timescales
  - Aggregating data over different timescales
- What about supervised learning?
  - What sorts of models make sense?
  - How do we do test/train splitting?
  - How do we do model selection?

# Supervised learning review: Regression

Ordinary data (no order):



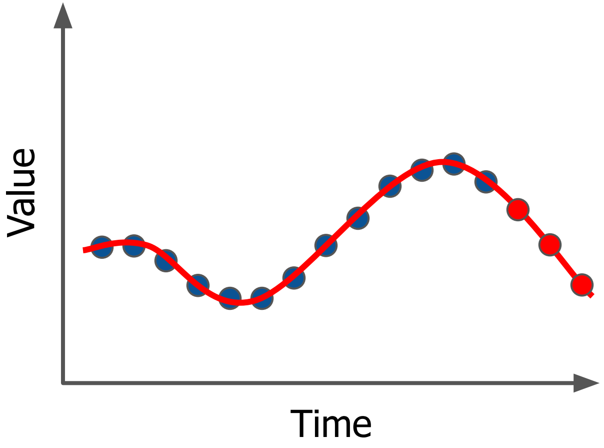

Predict label from feature using linear regression

Key features:

- Use a random subset as the test/train split
- Treat all data equally

# Supervised learning with time series

Time series data (sequential):




Predict future values from past values

What we want:

- Ability to extrapolate (not just interpolate)
- Assign more importance to more recent data

# Autoregressive (AR) model

**Main idea:** Use past k values to predict next value!

Equation of model: for each index t, we have:

$x[t] \approx \beta_0 + \beta_1 x[t-1] + \beta_2 x[t-2] + \ldots + \beta_k x[t-k]$

What is the set of $\beta_i$ coefficients that does the best job of predicting future x values?

# Autoregression is regression!

Start with time series: `(x[0],x[1],x[2],...,x[99])`
and then assemble into a table with lagged values:

$$X = \begin{pmatrix} x[2] & x[1] & x[0] \\ x[3] & x[2] & x[1] \\ x[4] & x[3] & x[2] \\ \vdots & \vdots & \vdots \\ x[97] & x[96] & x[95] \\ x[98] & x[97] & x[96] \end{pmatrix} \qquad Y = \begin{pmatrix} x[3] \\ x[4] \\ x[5] \\ \vdots \\ x[98] \\ x[99] \end{pmatrix}$$

Features Labels

# Autoregression is regression!

Then perform multiple regression:

$$\begin{pmatrix} x[3] \\ x[4] \\ x[5] \\ \vdots \\ x[98] \\ x[99] \end{pmatrix} \approx \beta_0 + \begin{pmatrix} x[2] & x[1] & x[0] \\ x[3] & x[2] & x[1] \\ x[4] & x[3] & x[2] \\ \vdots & \vdots & \vdots \\ x[97] & x[96] & x[95] \\ x[98] & x[97] & x[96] \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}$$

to solve for the β coefficients.

# Autoregression forecasting

Using the coefficients { $\beta_0, \beta_1, \beta_2, \beta_3$ }, predict future values:

Data: { x[0], x[1], … ,x[98], x[99] }

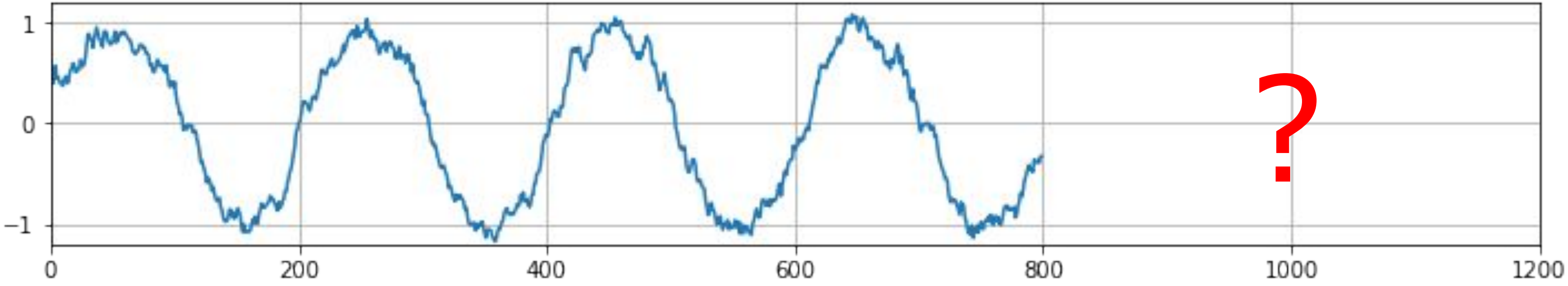
Predictions:

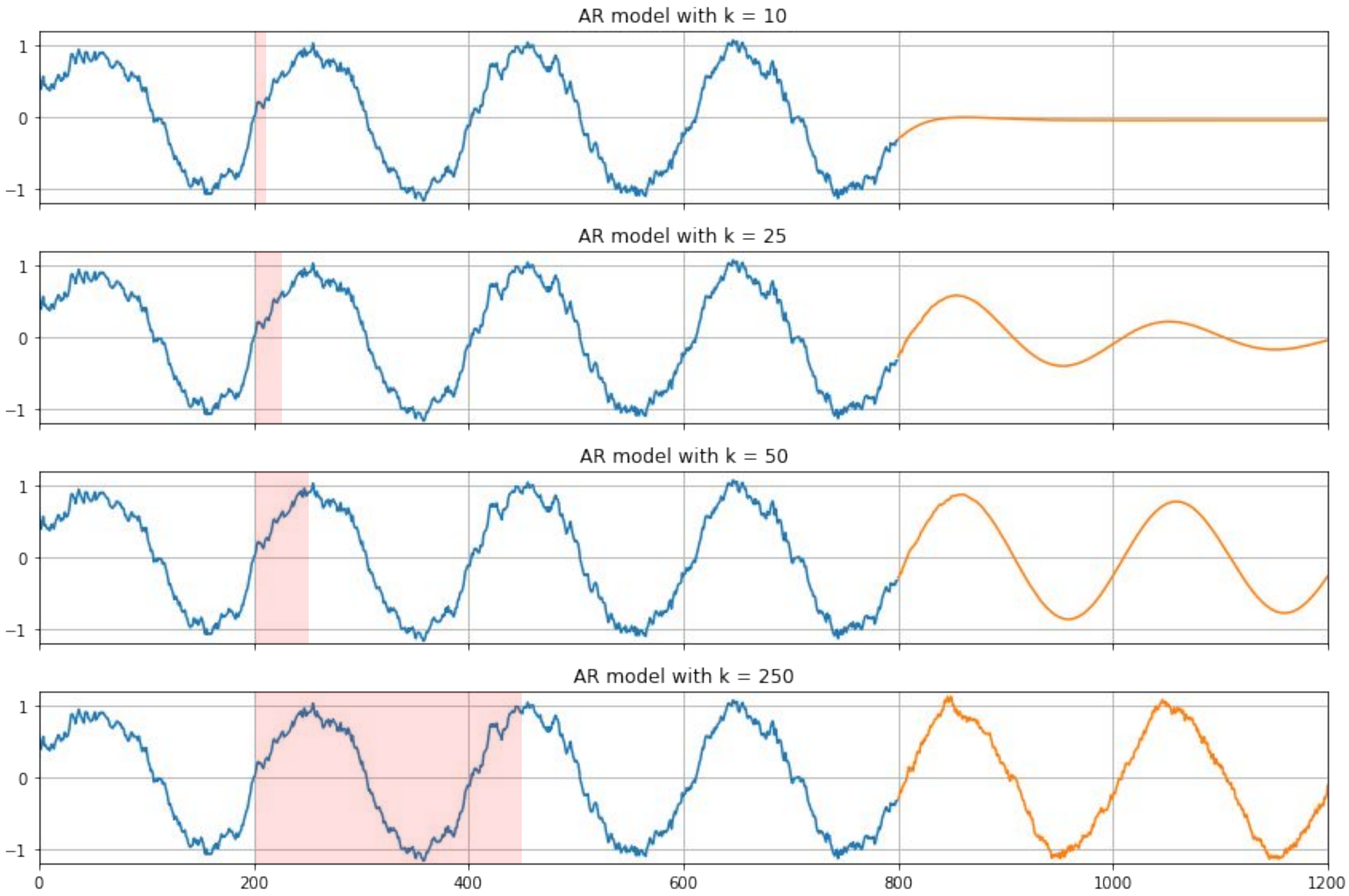
- $x[100] = \beta_0 + \beta_1 x[99] + \beta_2 x[98] + \beta_3 x[97]$
- $x[101] = \beta_0 + \beta_1 x[100] + \beta_2 x[99] + \beta_3 x[98]$
- $x[102] = \beta_0 + \beta_1 x[101] + \beta_2 x[100] + \beta_3 x[99]$
- $x[103] = \beta_0 + \beta_1 x[102] + \beta_2 x[101] + \beta_3 x[100]$

Can continue forecasting based on prior forecasts, but errors will accumulate!

# Example: Noisy sine wave

- Noisy sine wave observed for 800 timesteps
- Can we forecast the next 400 timesteps?






**Underfit:** model just uses the mean

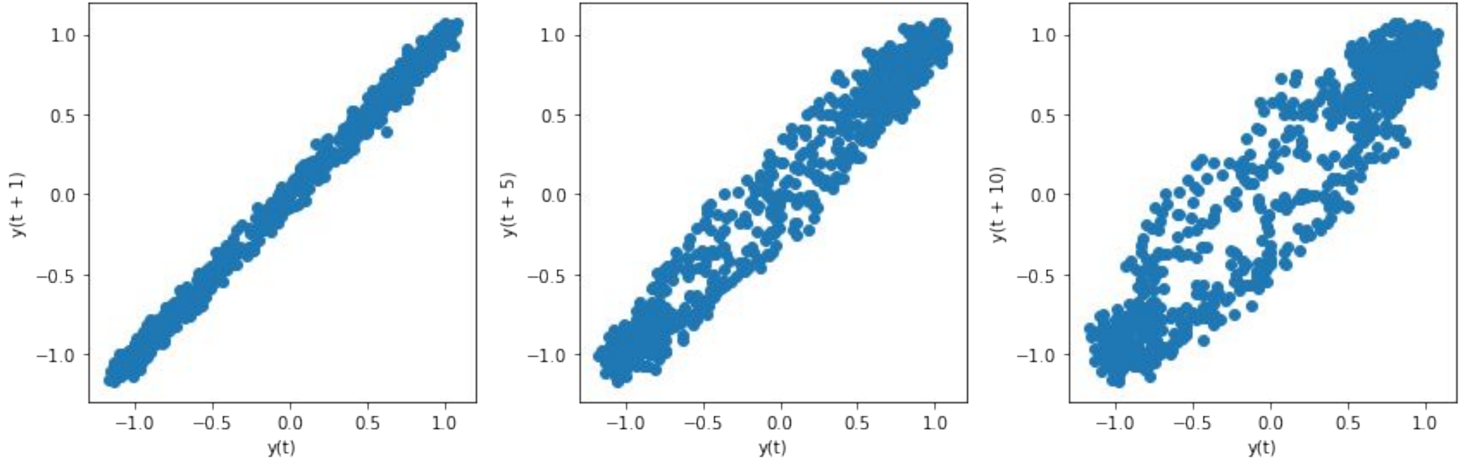
**Not bad:** model captures the short-term trend

**Good:** model captures longer-term trend

**Overfit:** model is "learning the noise"

# Lag plot

To know whether $x_{t+1}$ can be predicted by $x_t$, we can make a lag plot: a scatter plot of $x_t$ versus $x_{t+k}$ for some $k > 0$.




For the noisy sine wave, nearby x's are highly correlated!

# Autoregressive models in Python

```
from pandas.plotting import lag_plot

lag_plot(x, lag=1)  # make a lag plot (will AR work?)
```

```
from statsmodels.tsa.ar_model import AR

ar = AR(x)                     # input the time series
arfit = ar.fit(maxlag=5)       # set number of lags
```

Extract β parameters

```
# obtain parameters
arfit.params
```

Forecast future values

```
# Predict future values
pred = arfit.predict(start=100, end=200)
```

# How to choose the lag k?

- Use a test/train split as we did with other supervised learning model selection problems.
  **Note:** split after creating features, not before!
- Other popular methods from statistics includes the Akaike Information Criterion (AIC) and other methods.
  (beyond the scope of this class)

# Summary

- Autoregressive (AR) models use the past k values of the time series to predict the next value.
- Lag plots are a useful way to see if an AR model will work.
- Forecasting errors will always accumulate over time.
- Choosing the lag k can be done via model selection methods. Too small leads to a poor fit, too large leads to overfitting the data.