Time Series Forecasting Product

Diogo Resende

€

The anatomy of a Forecasting product

Traditional Forecasting Approach

Model → Outcome

Forecasting Model → Forecast

Modern Forecasting Product

Model 1, Model 2, Model n → Ensemble → Forecast

Forecasting components

- Seasonalities
- Outliers
- Regressors
- Non-linearity
- Trend changes

Why Ensemble

Deep dives

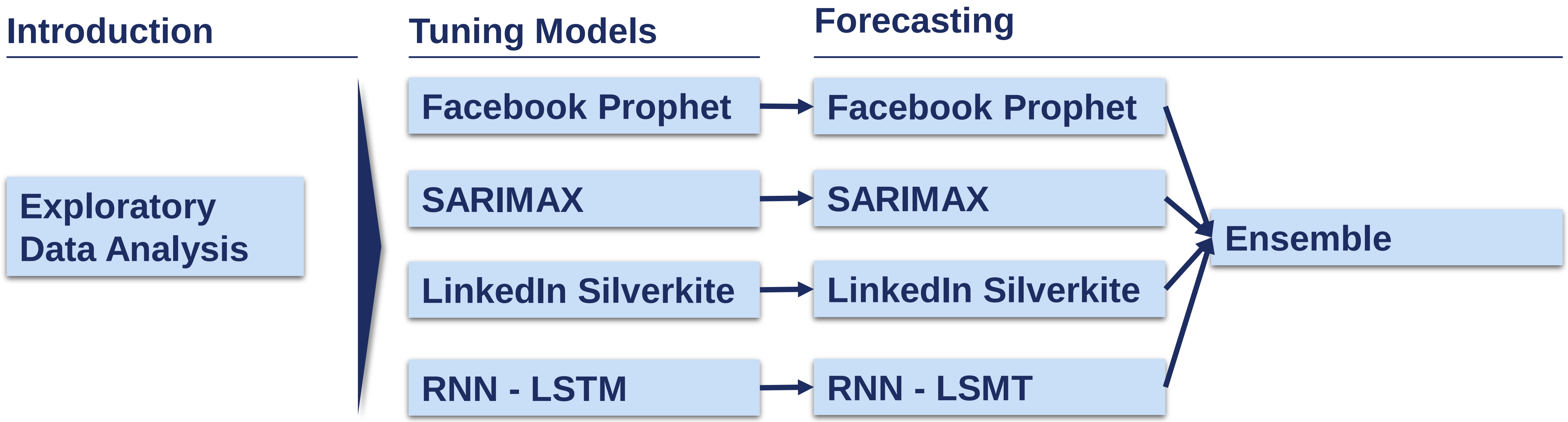
The research on combining forecasts to achieve better accuracy is extensive, persuasive, and consistent.

Essam Mahmoud,
"Accuracy in Forecasting: A Survey," *Journal of Forecasting*, April–June 1984, p. 139;

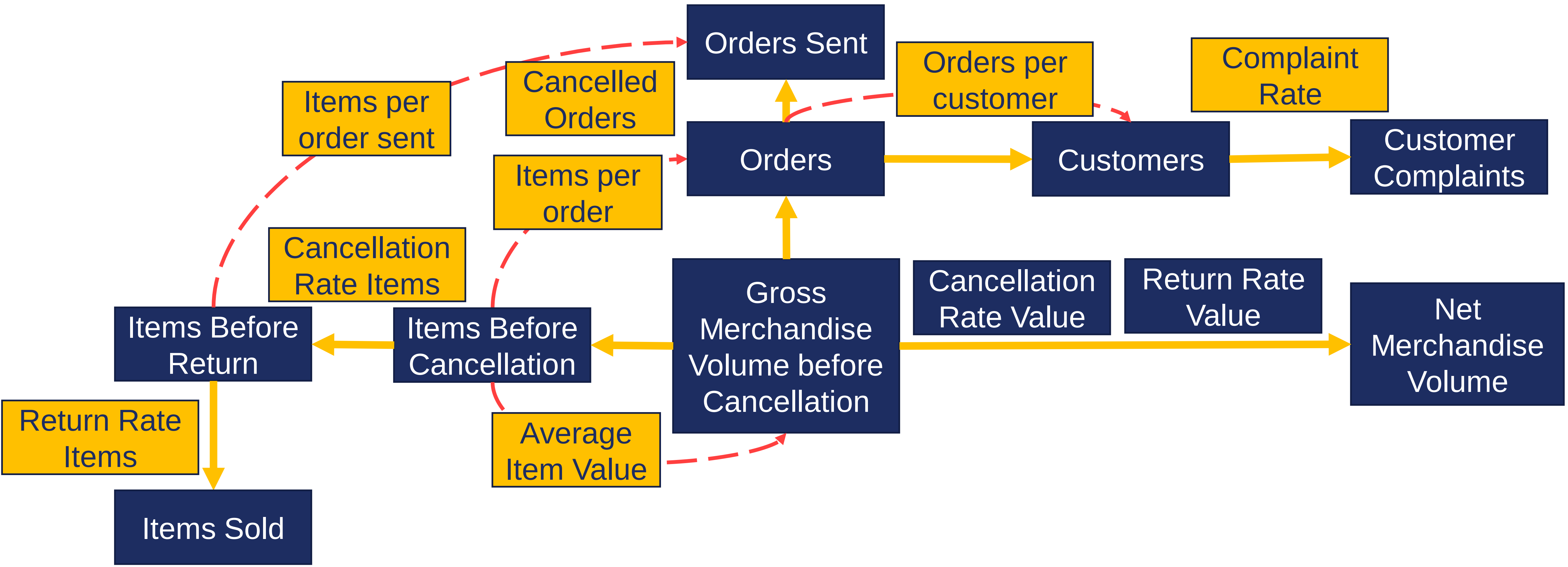
Spyros Makridakis and Robert L. Winkler,
"Averages of Forecasts: Some Empirical Results," Management Science, September 1983, p. 987

Victor Zarnowitz,
"The Accuracy of Individual and Group Forecasts from Business Outlook Surveys," Journal of Forecasting, January–March 1984, p. 10.

The Project


Introduction
Tuning Models
Forecasting
Exploratory Data Analysis
Facebook Prophet
SARIMAX
LinkedIn Silverkite
RNN - LSTM
Facebook Prophet
SARIMAX
LinkedIn Silverkite
RNN - LSMT
Ensemble

Why Forecasting matters



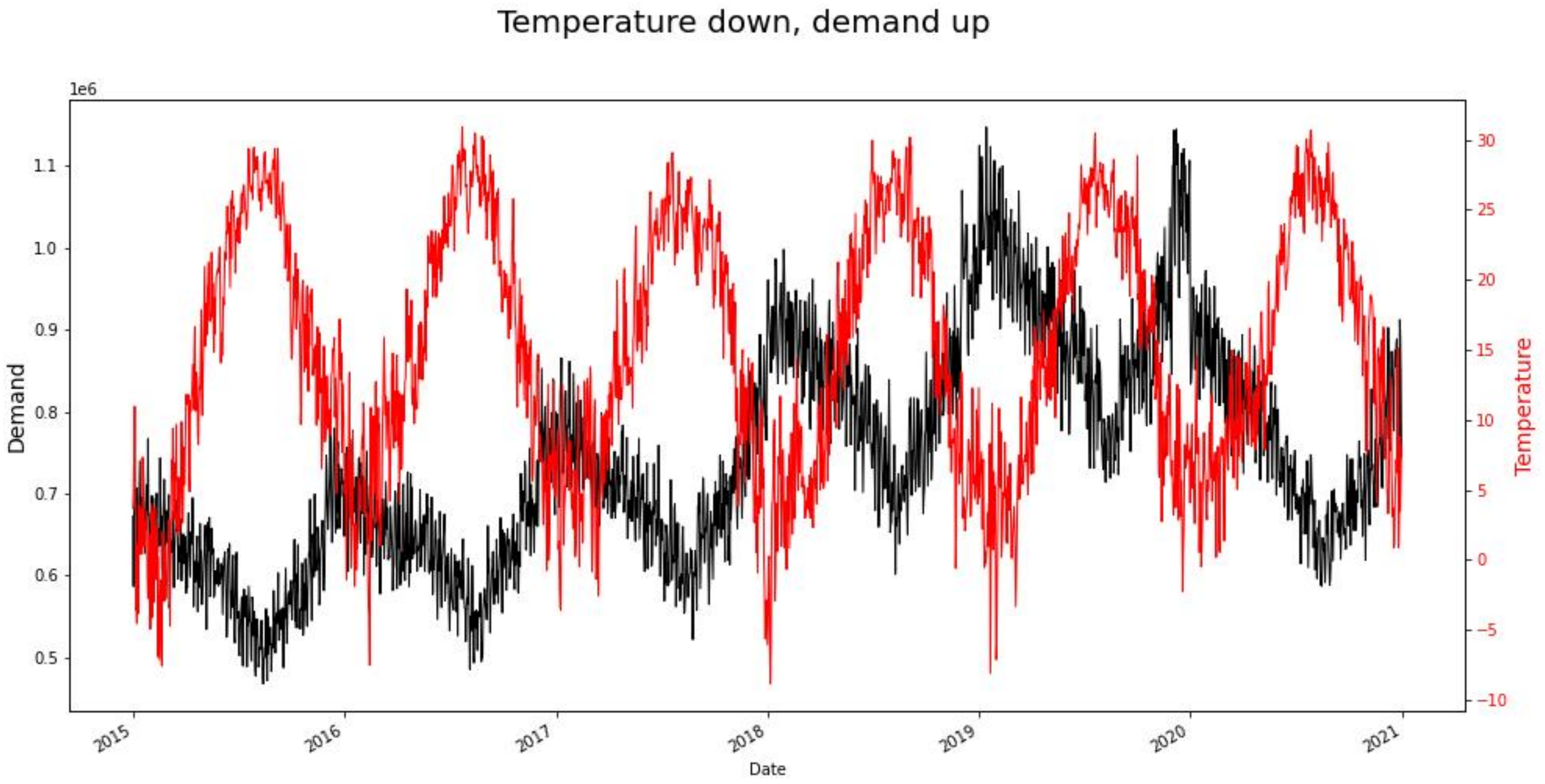
Exploratory Data Analysis

Section Overview

What will be achieved

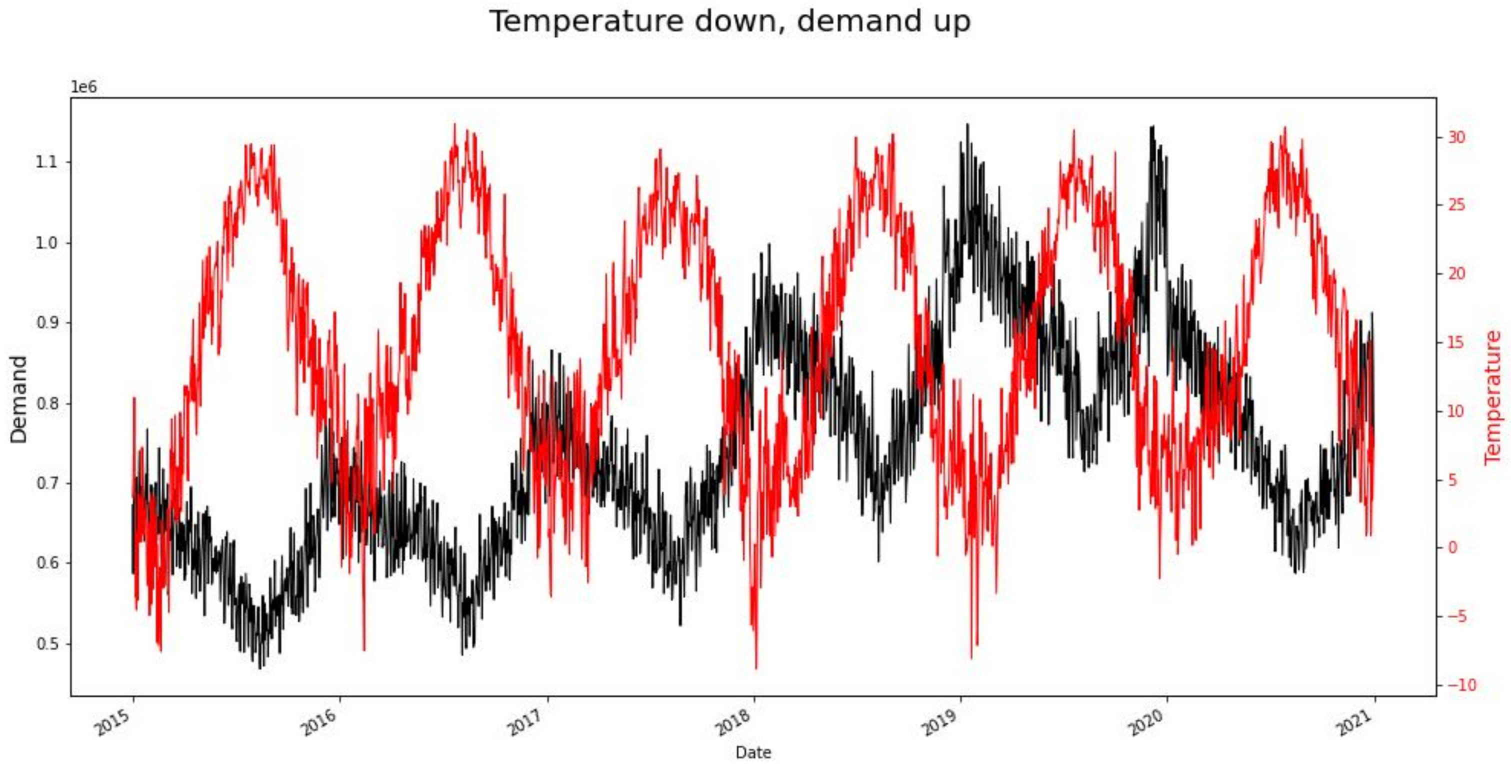
1. Time Series Concepts
2. Seasonality Types
3. Auto-Correlation
4. Summary Statistics
5. Correlation
6. Cool Visualizations

Our data


Temperature down, demand up
1e6
1.1
1.0
0.9
0.8
0.7
0.6
0.5
Demand
30
25
20
15
10
5
0
−5
−10
Temperature
2015
2016
2017
2018
2019
2020
2021
Date

What is Time Series Data?

Visualization



Key ideas

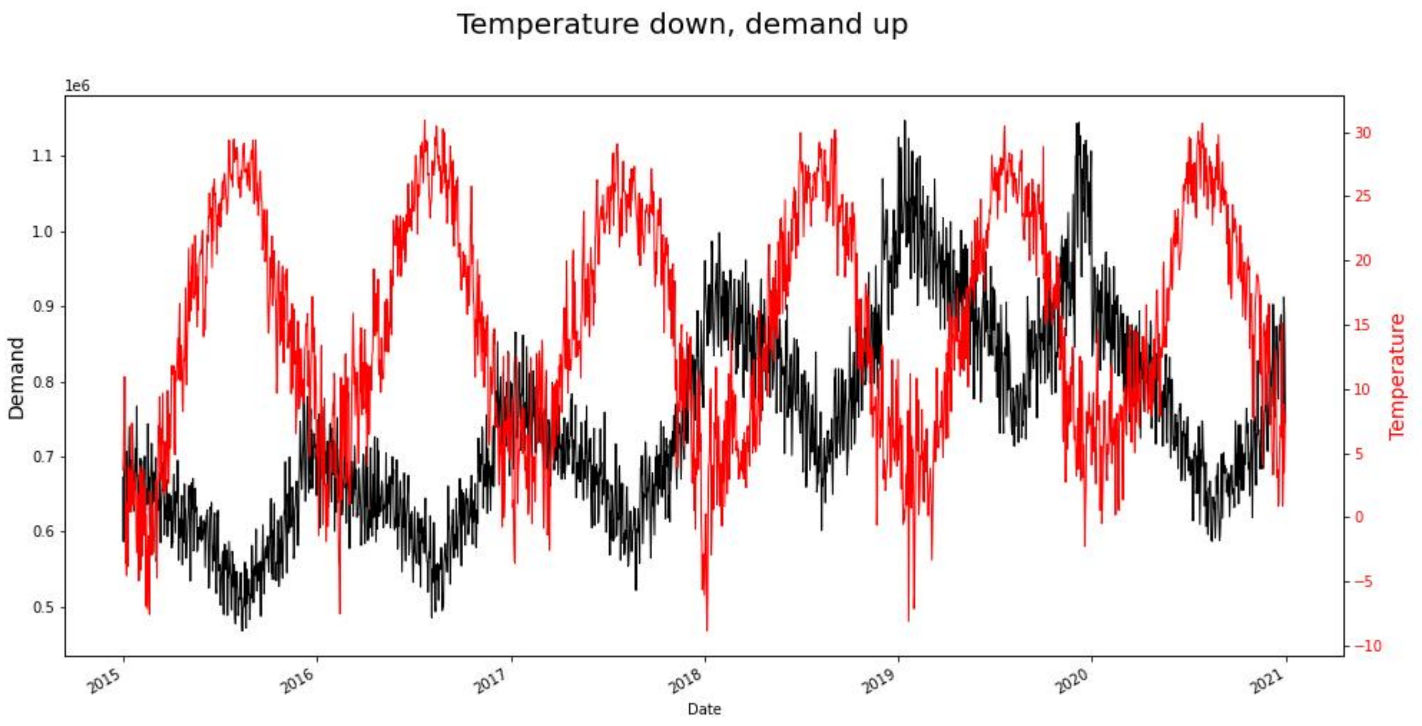
Sequence of data points in time order (oldest to newest)

Most commonly, it is data recorded in equally distanced time periods

Type of Panel Data (multidimensional dataset)

Time Series are usually decomposed into 3 parts

Visualization



Key ideas

A seasonal Time Series can be decomposed into:

- **Trend**
- **Seasonality**
- **Error**

We try to use external regressors to model the remaining error term.

Case Study Briefing – Demand Forecasting

Scenario

Airbnb missed the earning expectations

The market where the company is struggling is the US

Forecasting Product

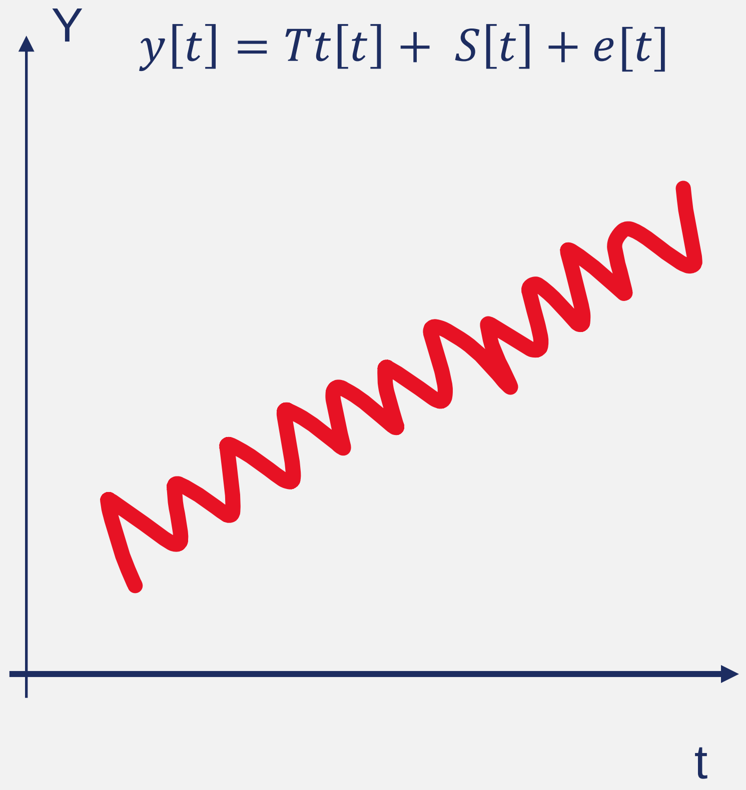
Demand in New York

1. Holidays, Temperature and Marketing Investment
2. Daily Demand
3. Historical Data to find patterns
4. Predict demand for the incoming month

Additive vs. Multiplicative

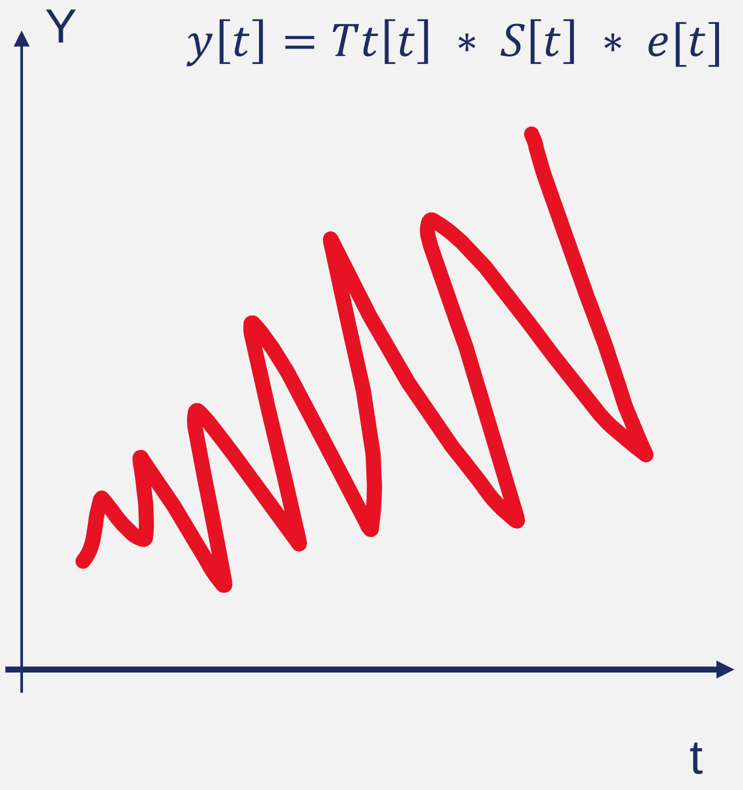
Additive

$$y[t] = Tt[t] + S[t] + e[t]$$



Multiplicative

$$y[t] = Tt[t] * S[t] * e[t]$$



Key ideas

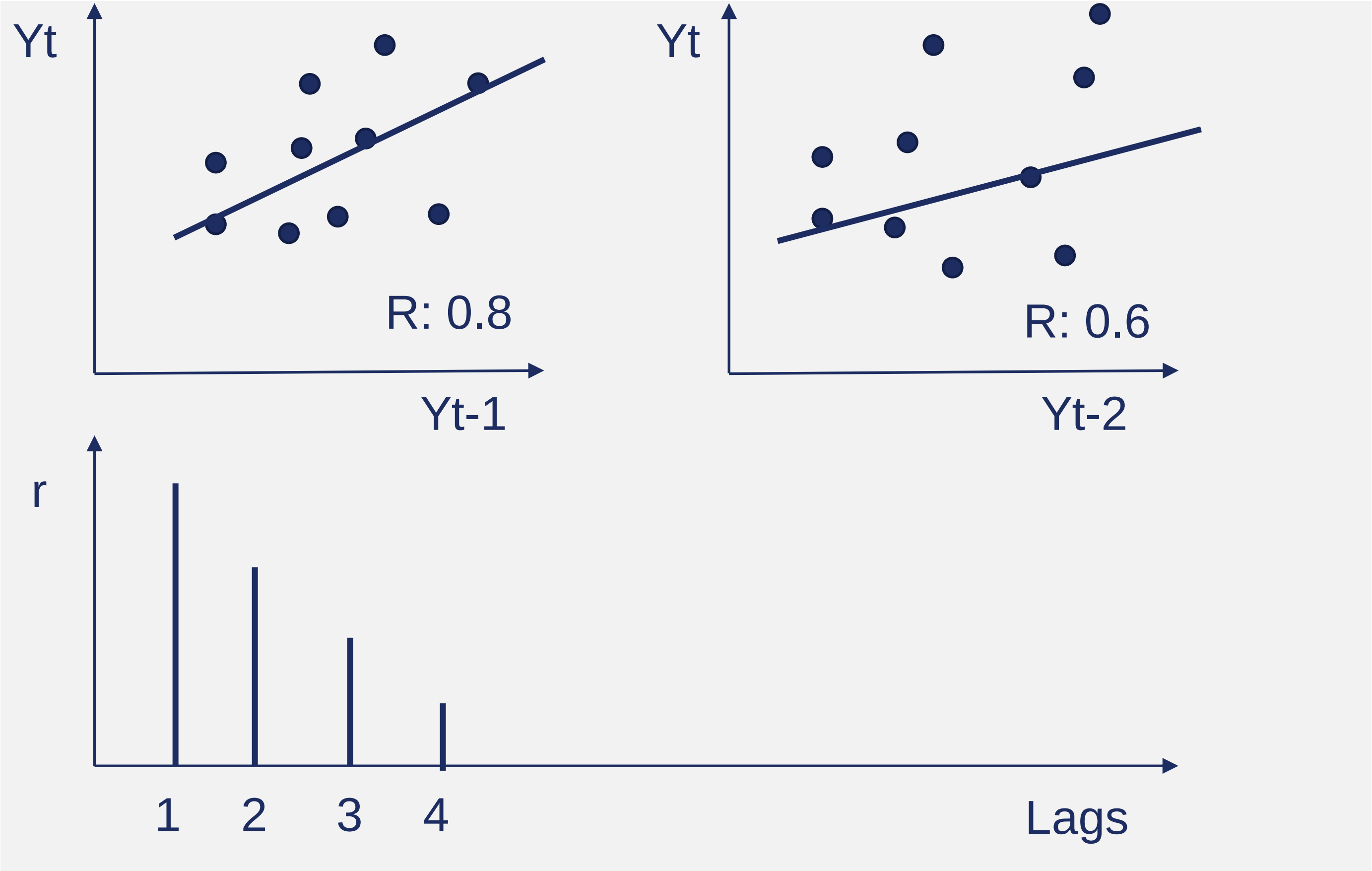
If we talk about seasonality in terms of percentage, then we should consider a multiplicative seasonality.

If it is in adding absolute values, then it is additive.

If trend is exponential, then it is multiplicative

Auto-correlation plots (ACF)

Visualization



Description

There is information in the past

You correlate the time series with its lagged values

The correlation will decrease with higher lags

Facebook Prophet

Section Overview

What will be achieved

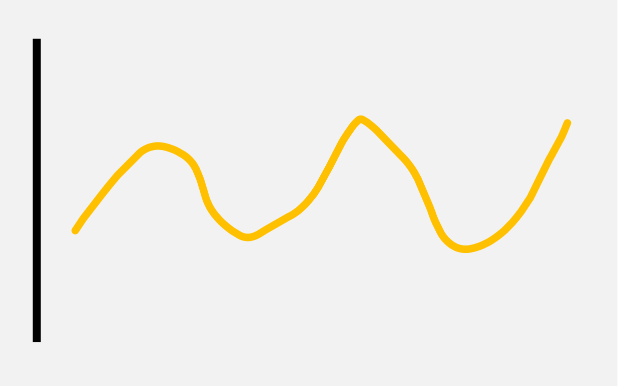
1. Facebook Prophet key concepts
2. Impact of events
3. Cross-Validation
4. Parameter Tuning
5. Measuring errors
6. Cool Visualizations

Structural Time Series

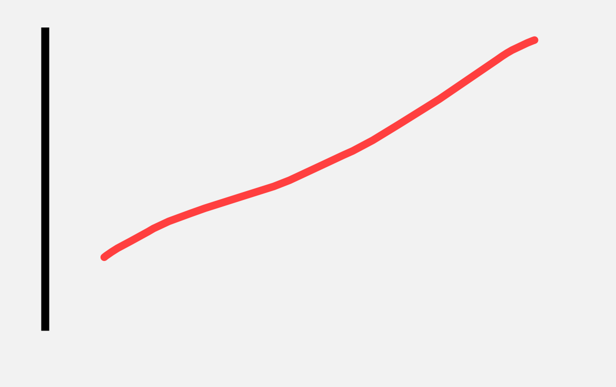
Visualization

Data

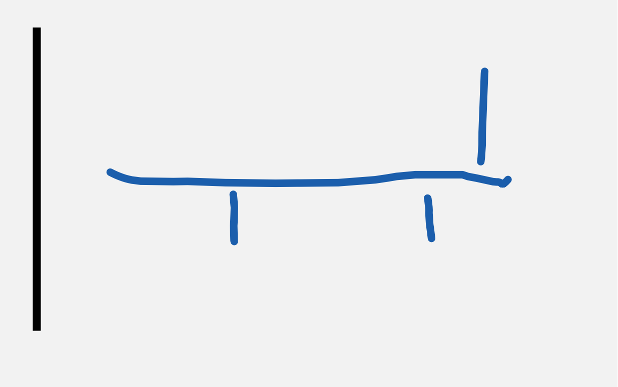
Seasonality



Trend



Exogenous impacts



Description

Structural Time Series is the decomposition of the data in at least:

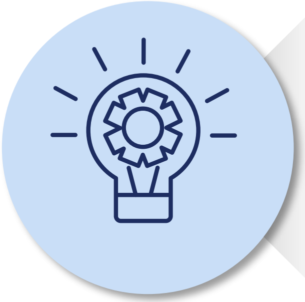
- Trend
- Seasonality
- Exogenous impacts
- Error Term

Methodological framework

$$y(t) = c(t) + s(t) + x(t) + \epsilon$$

Facebook Prophet quick facts

Which?



Description

1. Built by facebook
2. Stan background - probabilistic programming language for statistical inference
3. Dynamic Holidays
4. Prophet is customizable in ways that are intuitive to non-experts
5. Built-in Cross Validation

Prophet Mechanics

Methodological framework

$$y(t) = c(t) + s(t) + h(t) + x(t) + \epsilon$$

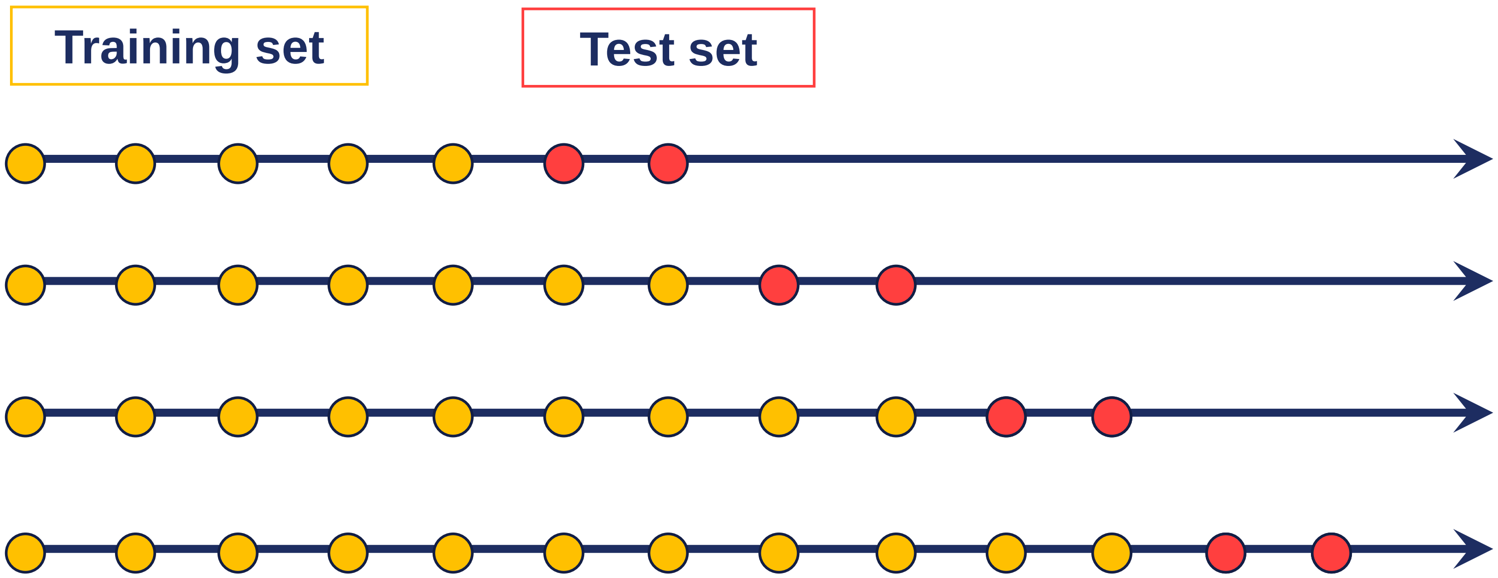
Where:

c(t)	Trend +
s(t)	Seasonality +
h(t)	Holiday effects +
x(t)	External regressors +
e	error

Facebook Prophet Model

Component	Description
Holidays	Dataframe that we prepared
Seasonality_mode	Multiplicative or additive
Seasonality_prior_scale	Strength of the seasonality
Holiday_prior_scale	Larger values allow the model to fit larger seasonal fluctuations
Changepoint_prior_scale	Does the Trend change easily?

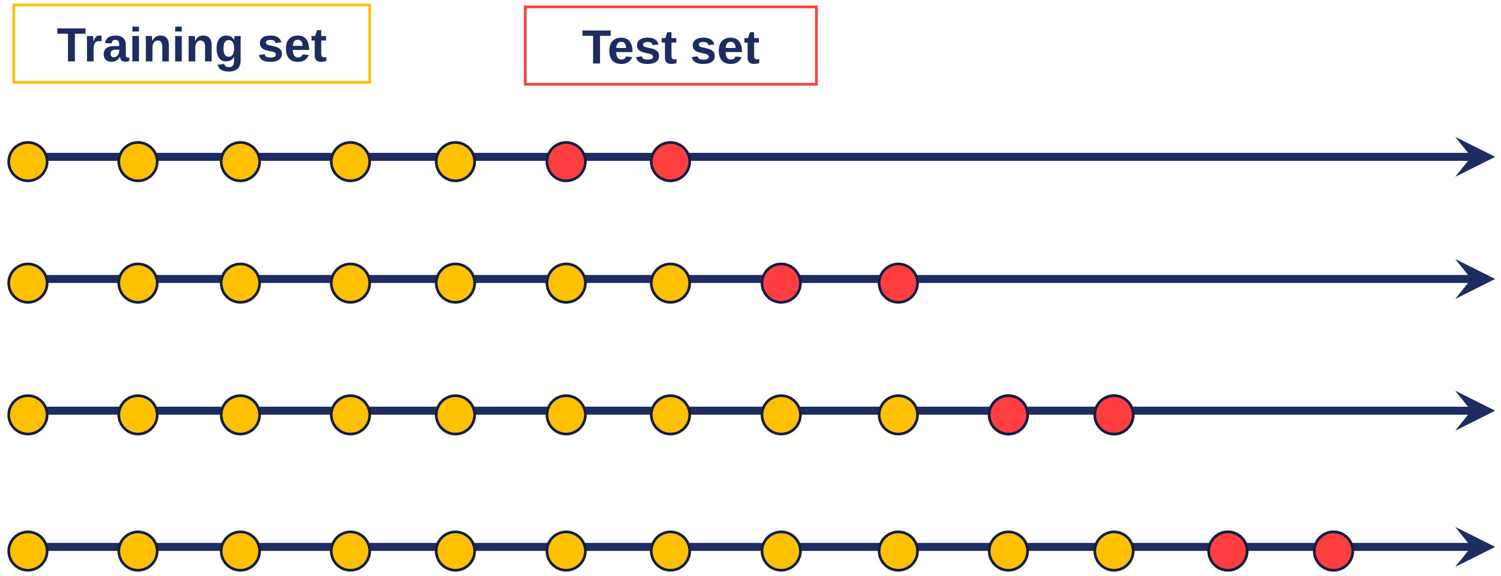
Cross Validation – Rolling Forecast



Key Idea

Repeating the assessment of our model reinforces its evaluation

Cross Validation – Sliding Forecast

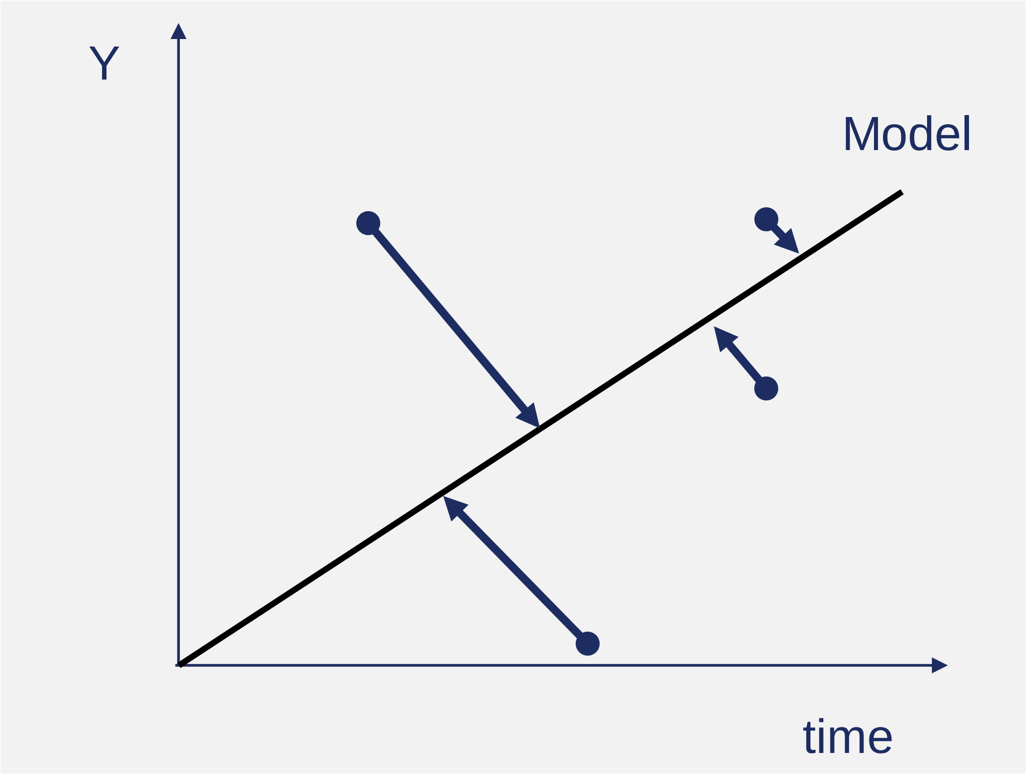


Key Idea

A rolling forecast adds training data as it performs Cross-Validation.
A sliding forecast always keeps the same size for the training data

Mean Absolut Error (MAE) vs Root Squared Mean Error (RSME)

Visualization



Key ideas

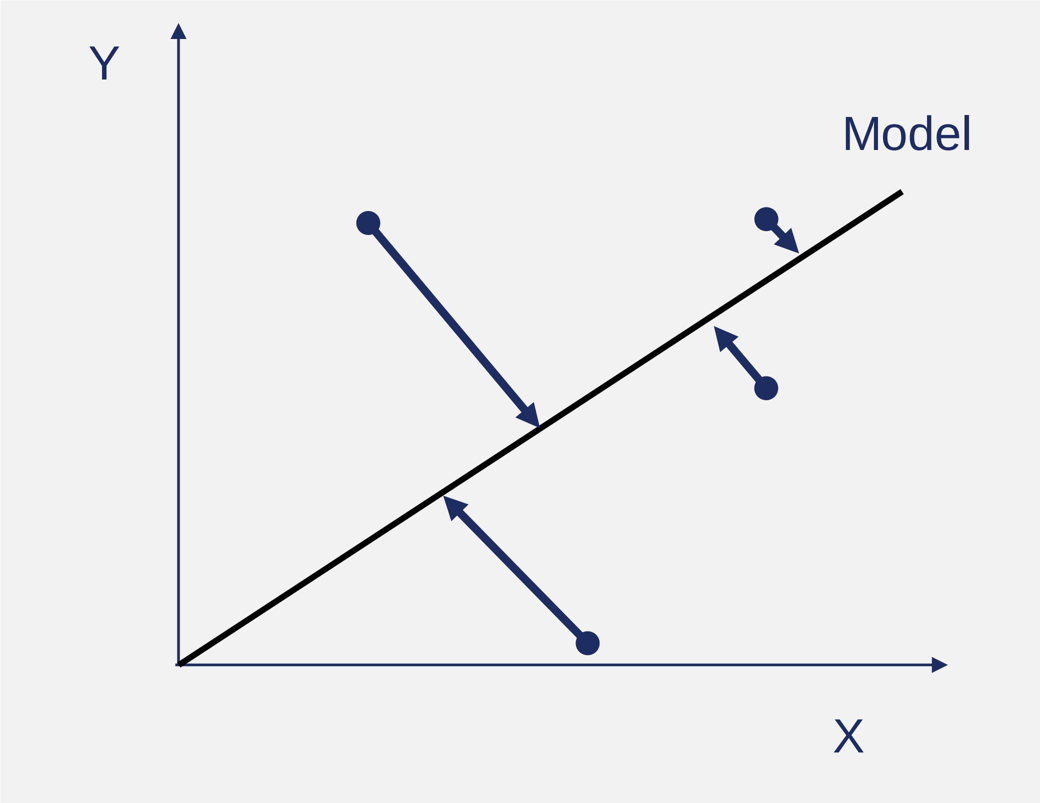
- MAE and RSME are performance indicators for Regression models with continuous dependent variables

$$MAE = \frac{\sum|y - \hat{y}|}{n} \qquad RSME = \sqrt{\frac{\sum(\hat{y} - y)^2}{n}}$$

- RSME is quite useful for models with extremes / outliers
- MAE is more interpretable.

Mean Absolut Percent Error (MAPE)

Visualization



Key ideas

- MAPE represents a very interpretable way of measuring errors

$$MAPE = \frac{\sum \frac{|y - \hat{y}|}{y}}{n}$$

- Clear downside is that all error has the same relevance, regardless of the magnitude, if the percent error is the same

- There is no universal good accuracy measure. It will depend on your problem and business need!

Parameter Tuning

Context

Advanced models have parameters to tune to optimize accuracy

Description

Parameter → Run Model → Measure error → Save error

Parameter	Save error
Holiday_prior_scale: 5	5000
Holiday_prior_scale: 10	6000
Holiday_prior_scale: 20	6100

Parameters to tune

Component	Description
Seasonality_prior_scale	Strength of the seasonality
Holiday_prior_scale	Larger values allow the model to fit larger seasonal fluctuations
Changepoint_prior_scale	flexibility of the automatic changepoint selection
Seasonality.mode	Multiplicative or additive

Pros and Cons

+ Pros	− Cons
1 Flexible	1 Requires optimization
2 Built-in Cross Validation	2 Not good with short-term dynamics
3 Dynamics Events	
4 Great with Regressors	

ARIMA, SARIMA & SARIMAX

Section Overview

About ARIMA

1. SARIMAX comes from ARIMA
2. Auto-Regressive Integrated Moving Average
3. Auto-Regressive is around 100 years old
4. Part of most modern Forecasting models
5. Another model, GARCH is used in Finance

What does it all mean?

Acronym	Description
ARIMA	AutoRegregressive Integrated Moving Average
SARIMA	Seasonal + ARIMA
SARIMAX	SARIMA + Exogenous variables

What is ARIMA?

Component	Description
AutoRegressive	The output is regressed on its own lagged values
Integrated	Number of times we need to do differencing to make our time series stationary
Moving Average	Instead of using the past values, the MA model uses past forecast errors.

AutoRegressive components

Key Idea

Past values, the lags, contain information that help predict future values

Visualization

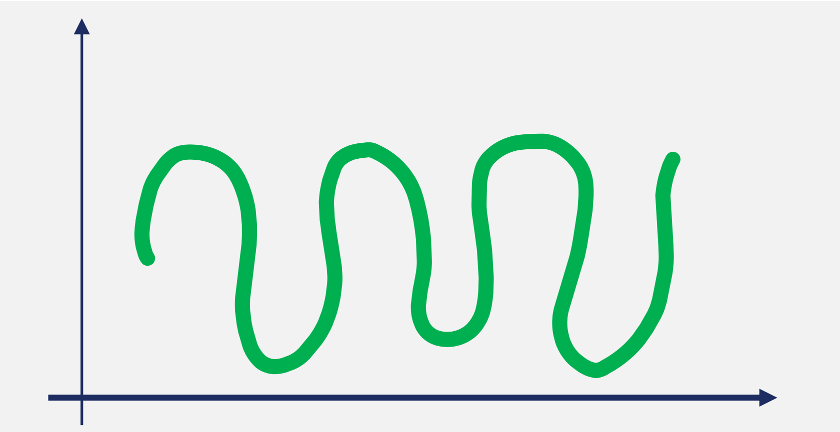
$$Y_t = c + \alpha_1 * Y_{t-1} + \alpha_2 * Y_{t-2} + \ldots + \alpha_n * Y_{t-n}$$

How to determine how many lags

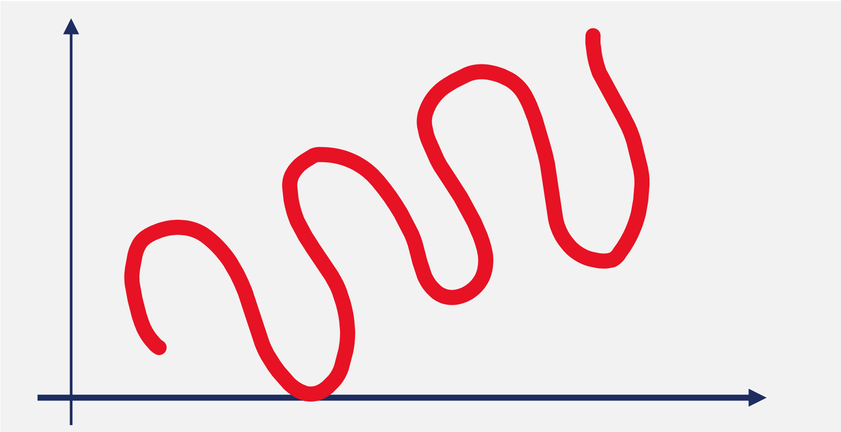
We will perform parameter tuning

Stationarity

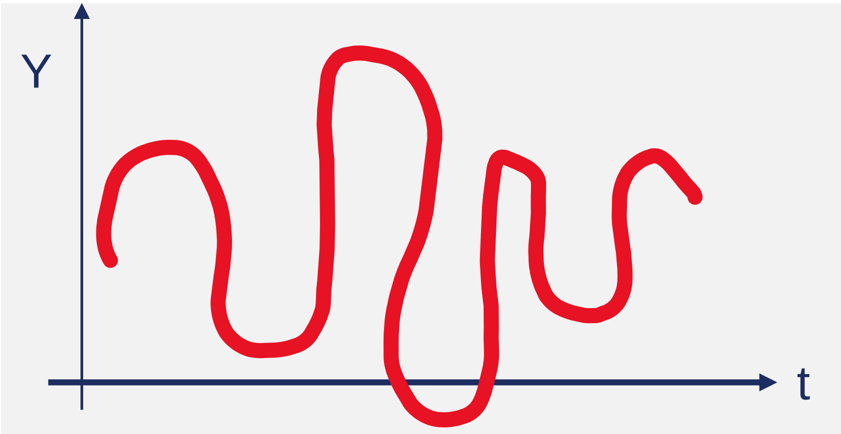
Stationary Time Series



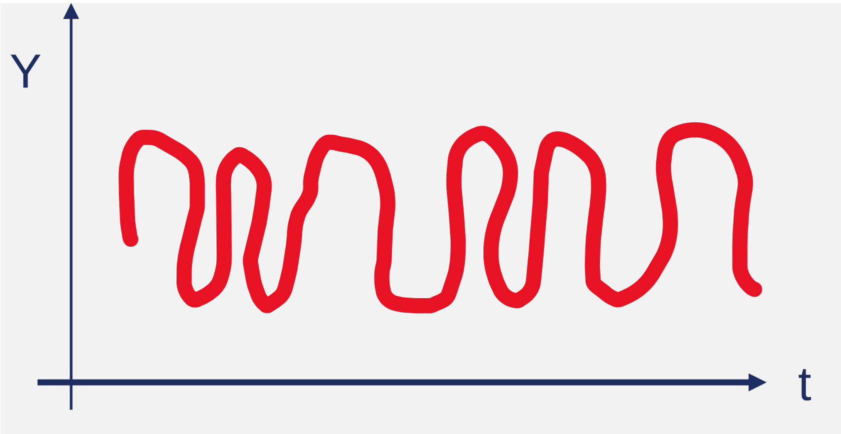
Time dependent mean



Time dependent variance



Time dependent covariance



Key idea

Mean, variance and covariance are not time dependent

Stationary Time Series have a clearly defined pattern

Statistical test:

Dickey-Fuller test. If p-value is less than 0.05, time series is considered stationary

Making Data Stationary

Time Series	1st differencing	2nd differencing
5	NA	NA
9	4	NA
1	-8	-12
7	6	14
3	-4	-10
7	4	8
4	-3	-7

Key idea

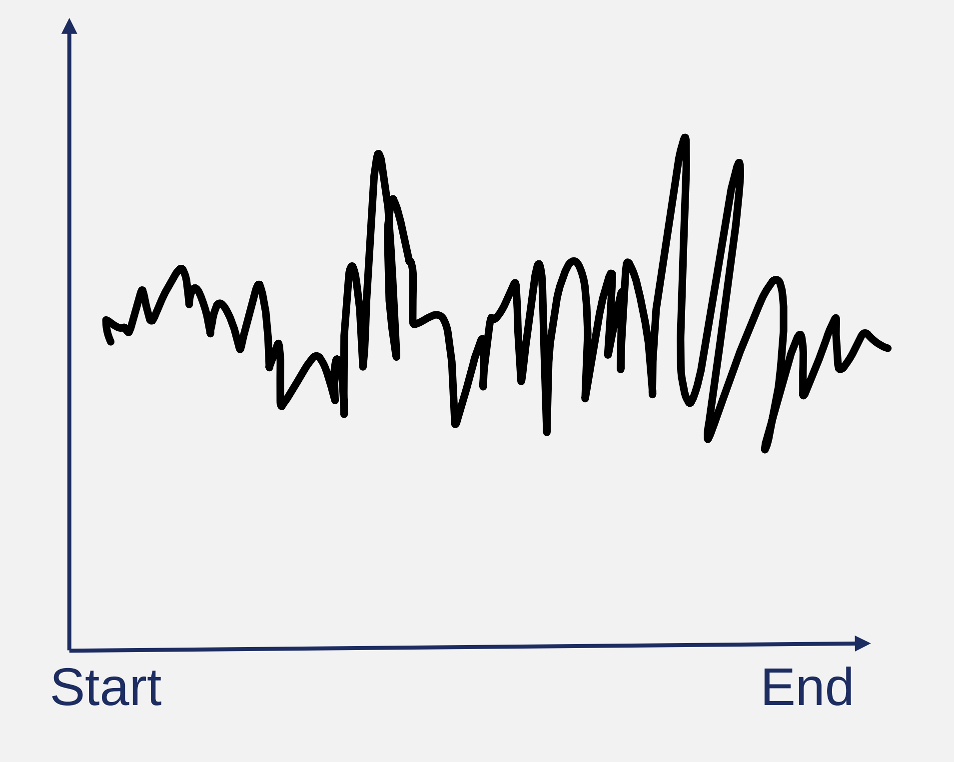
Making data stationary is simple, yet the concept is confusing.

From a practical perspective, it is a check that we need to do

We will find the optimal number of differencing

Moving Average components

Visualization of the errors



Methodological Framework

$$y_t = c + \alpha_1{}^* \, \varepsilon_{t-1} + \ldots + \alpha_n{}^* \, \varepsilon_{t-n}$$

What it is?

Past error lags, contain information that help predict future values

How to do it?

We will perform parameter tuning

3 factors to optimize in ARIMA/ARIMAX(p,d,q)

Order	Description	Explanation
p	Order of the Autoregressive	Number of time series lags used
d	Degree of first Differencing involved	Number of differences to make time series stationary
q	Order of the Moving Average part	Number of forecasting errors lags used

Key Idea

P, d, and q are non-negative integers.

6 factors to optimize in SARIMA/SARIMAX

	Data Type		Acronym		Factors
Time Series	Seasonal Data	→	S	→	P, D, Q
	Non-seasonal Data	→	ARIMA	→	p, d, q

Key Idea

Despite having 3 more factors to optimize, they mirror the classic ARIMA (p, d, q)

Pros and Cons

+

Easy Implementation 1

Great results 2

−

1 Better with low amount of time periods or frequency

2 Poor at dealing with non-linearity

3 Does not handle complex seasonalities

LinkedIn Silverkite

Section Overview

About LinkedIn Silverkite

1. Silverkite Process
2. How it differs from Facebook Prophet
3. Trend and Fitting Algorithms
4. Ridge and Gradient Boosting

Silverkite Overview

Data inputs

- Time Series
- Regressors
- Events
- Holidays

Also provided internally

Function inputs

- Growth terms
- Seasonalities
- Changepoints
- Lagged Regressors
- Auto-regression

Automated or customized

Model Magic

- Machine Learning

Output

- Forecast
- Accuracy
- Vizualization

Silverkite vs Prophet

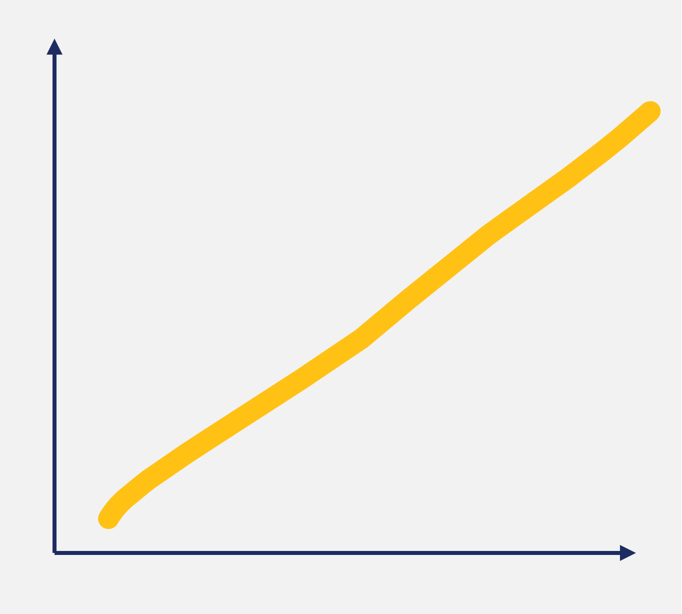
	LinkedIn Silverkite	Facebook Prophet
Speed	Faster	Slower
Forecast accuracy (default)	Good	Good
Forecast accuracy (customized)	High	Limited (medium / high)
Ease of use	Good (ok)	Good
Autoregressive	Yes	No
Fit	Bayesian	Ridge, Gradient Boosting…

Model Components

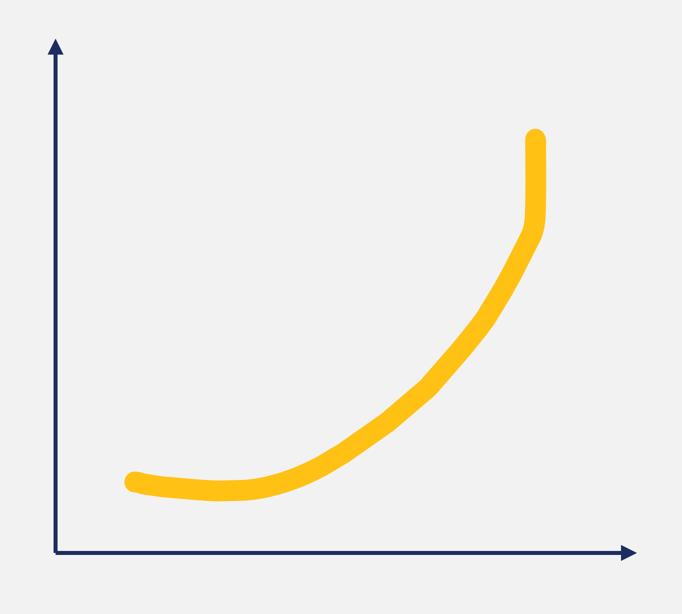
Name	Possibilities	Type
Growth terms	Linear, quadratic, square root	Tune
Seasonalities	Yearly, Quarterly, Monthly, etc..	Auto
Holidays / events	Country holidays/ other events	Input
Changepoints	When should the trend change	Auto
Regressors	Other factors influencing	Input
Lagged Regressors	Lagged effect of the regressors	Auto
Auto-regression	Using the Time Series itself	Auto

Growth terms

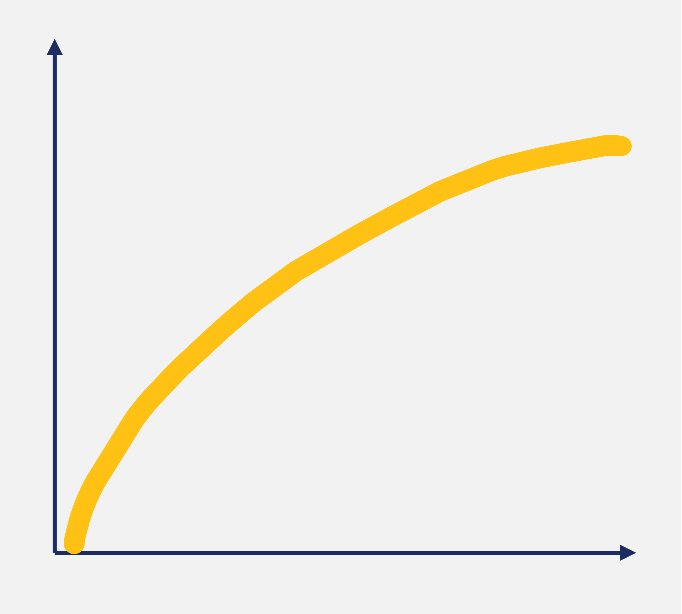
Linear



Quadratic

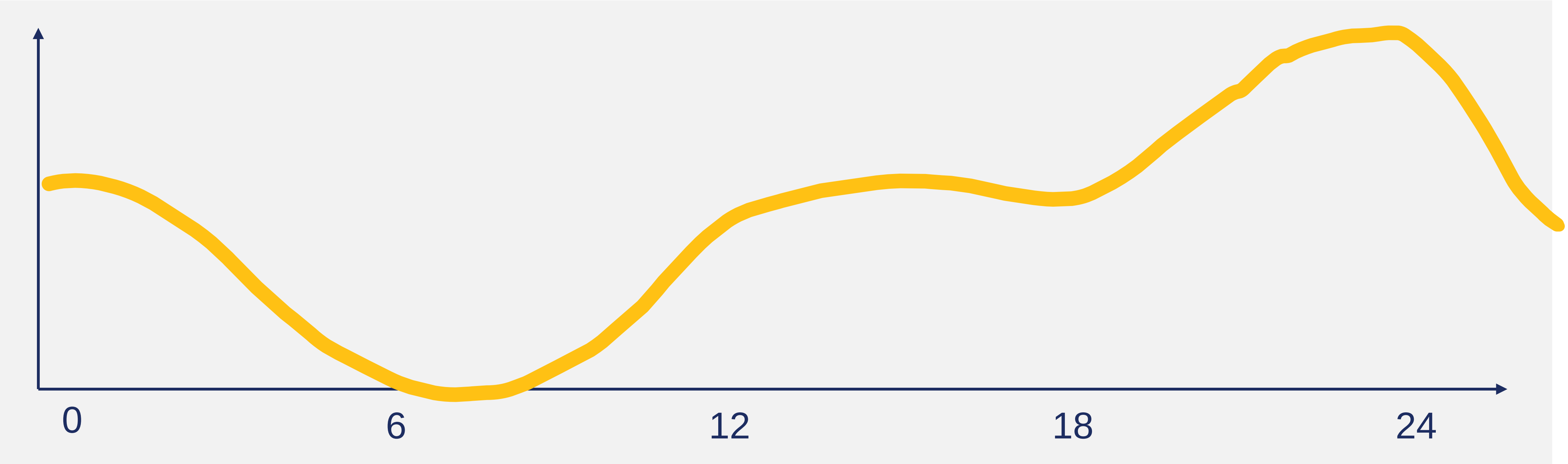


Square Root



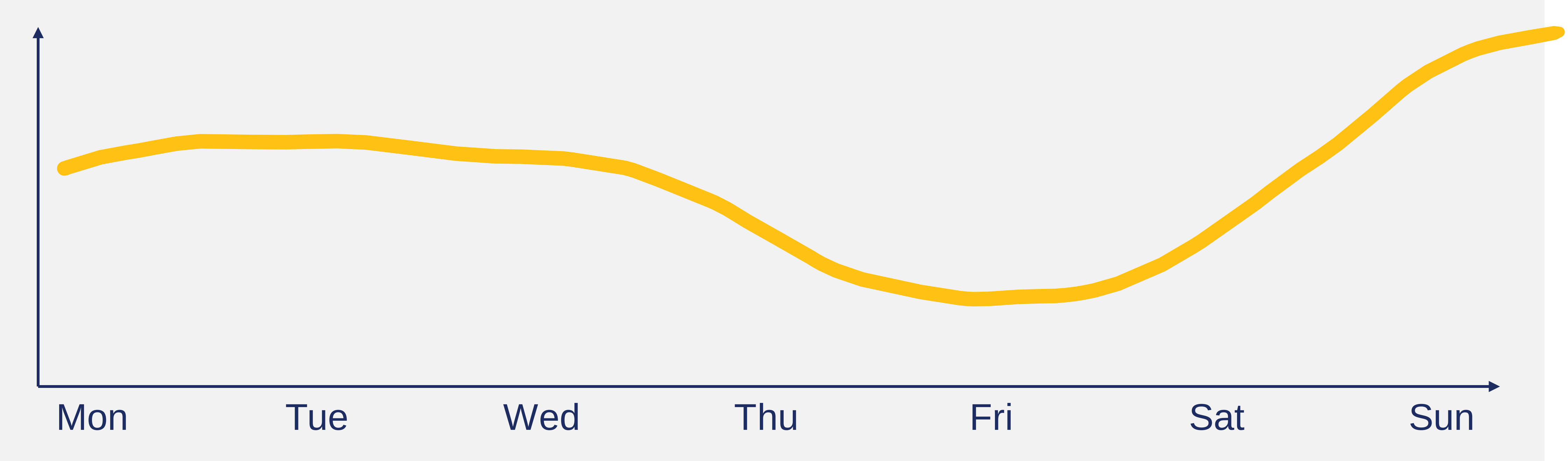
Netflix daily seasonality

Visualization



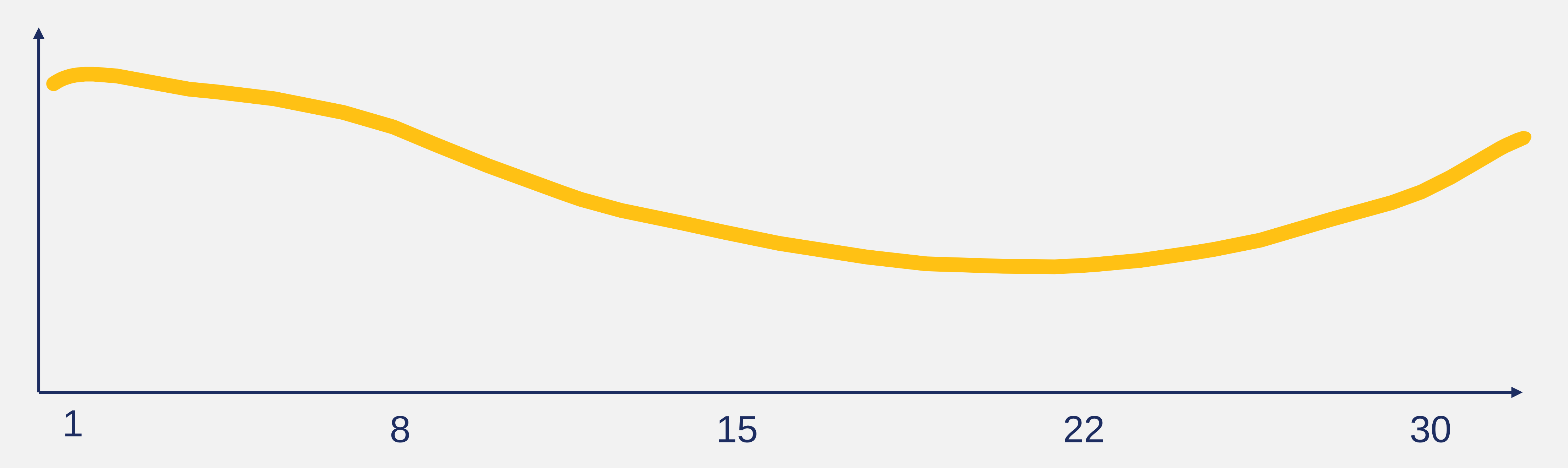
Netflix weekly seasonality

Visualization



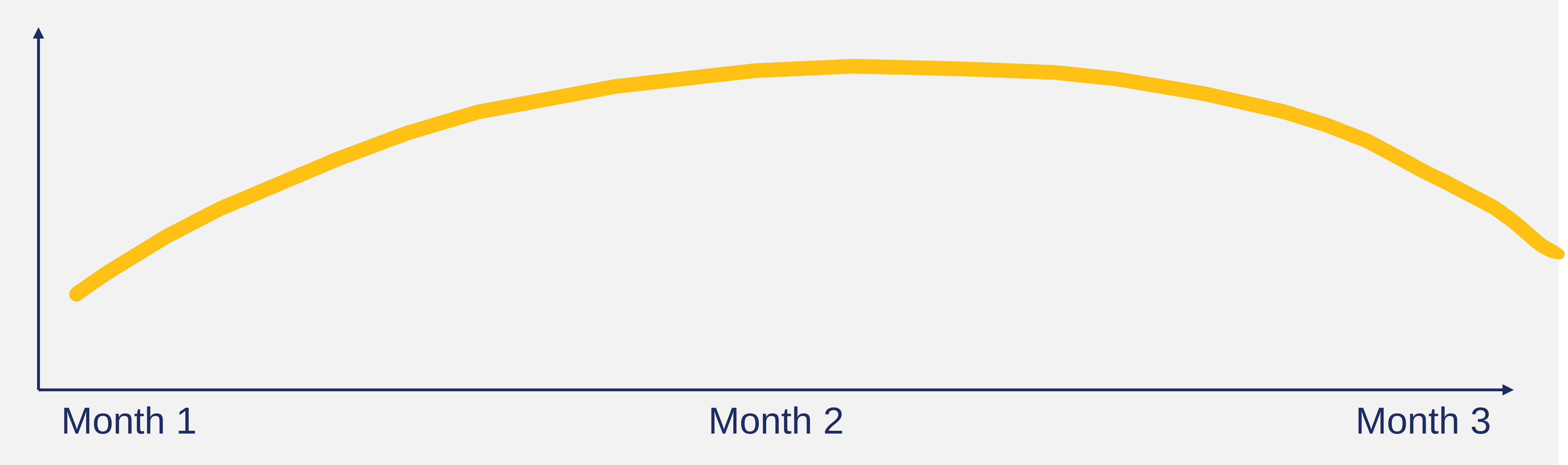
Netflix monthly seasonality

Visualization



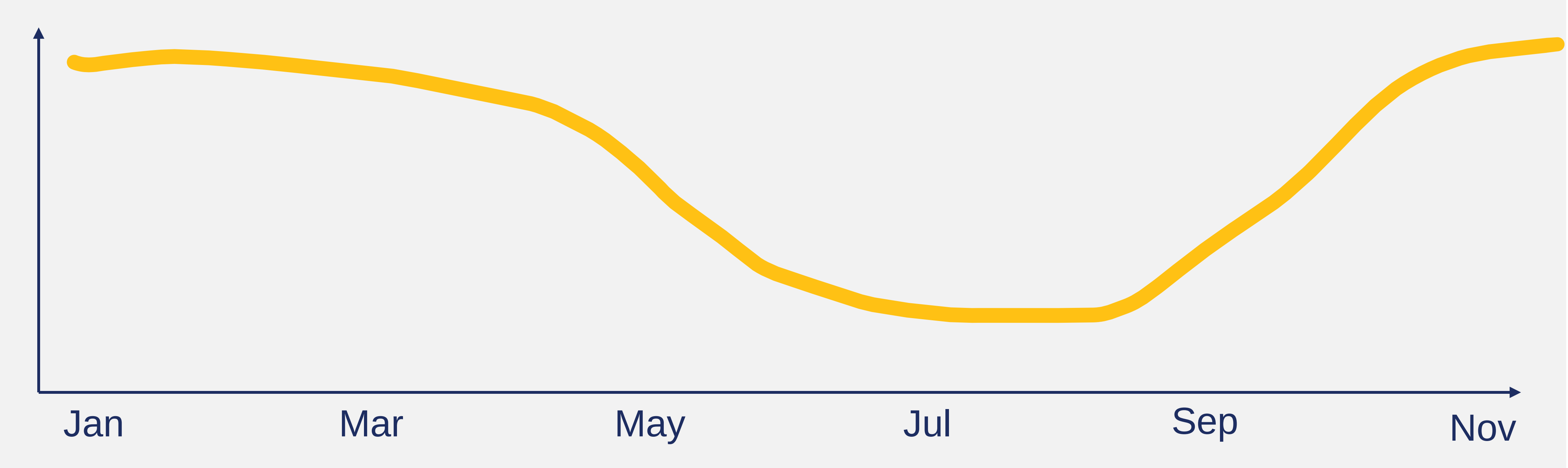
Netflix quarterly seasonality

Visualization



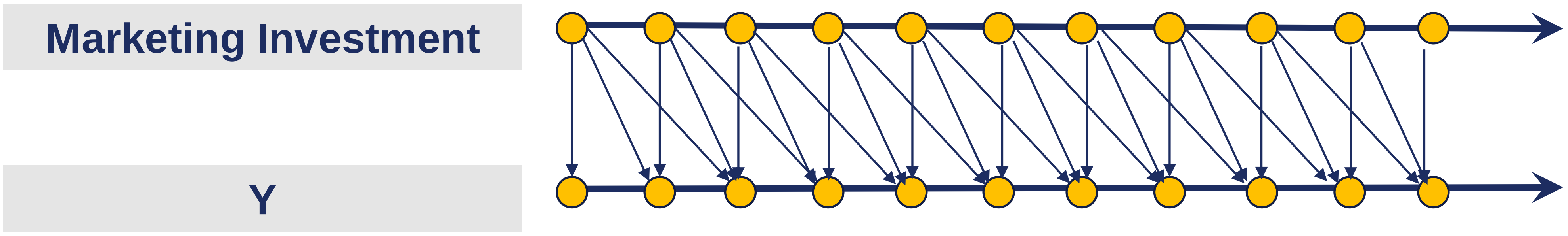
Netflix yearly seasonality

Visualization



Lagged Regressors

Visualization



Key Idea

If the regressors impact the days after the event happened, we use lagged regressors. We will set it on auto-pilot. The lags will depend on the forecasting horizon.

Fitting algorithm logic

Visualization

Time Series

- Growth terms
- Seasonalities
- Holidays/events
- Changepoints
- Regressors
- Lagged Regressors
- Auto-regression

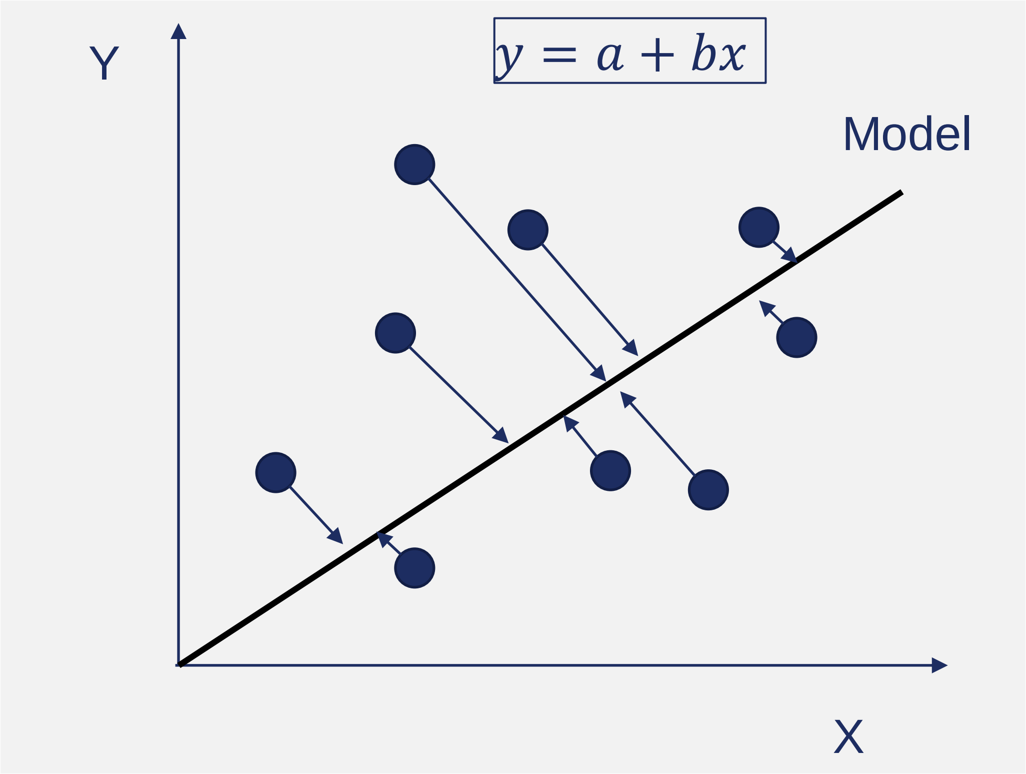
?

Fitting Algorithms

Name	Note
Linear Regression	Poor with collinearity
Elastic Net	
Ridge	
Lasso	
Stochastic Gradient Descent	Unstable
Lars	Outlier/noise sensitivity
Lasso Lars	
Random Forest	Tree Models don’t model growth well
Gradient Boosting	

From Linear to Ridge Regression

Visualization



Key ideas

Linear regression works by minimizing the residuals squared aka sum of least squares

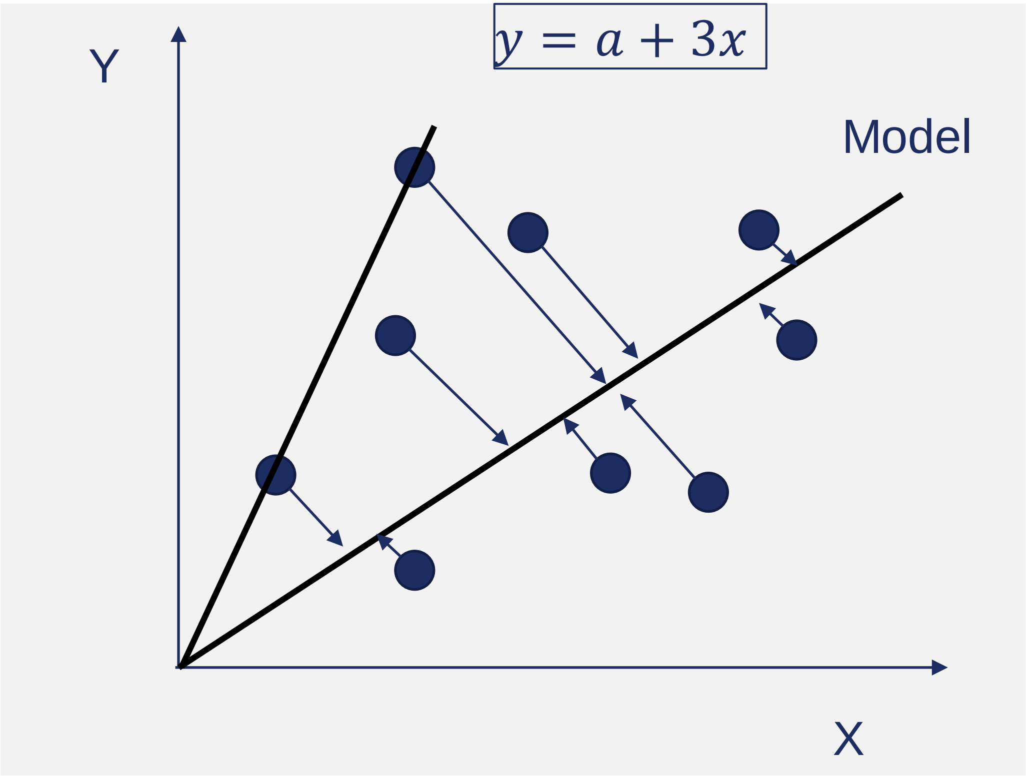
Ridge Regression Works by minimizing:

Residuals / least squares **+**

Lambda (Bias) * Slope ^{2}

From Linear to Ridge Regression

Visualization



Key ideas

Linear: minimizes the residuals squared

Ridge: minimizes the residuals squared + bias coefficient * slope^2

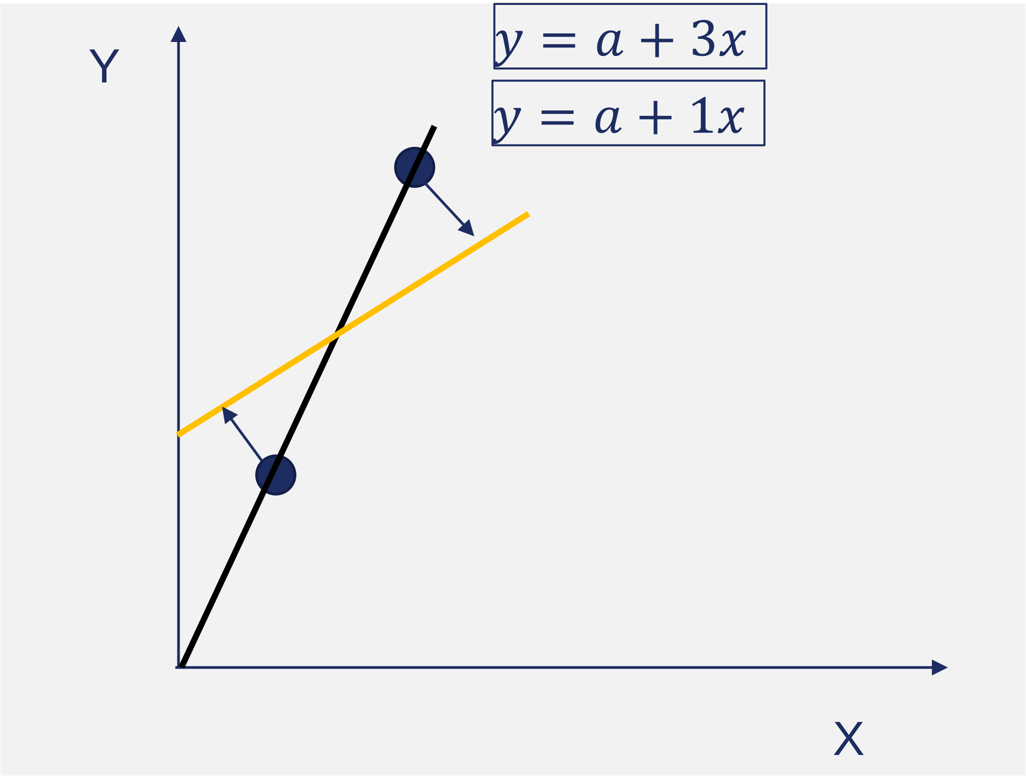
Scenario 1:

Linear: 0^2

Ridge: $0^2 + 1 * 3^2 = 9$

From Linear to Ridge Regression

Visualization



Key ideas

Linear: minimizes the residuals squared

Ridge: minimizes the residuals squared + bias coefficient * slope^2

Scenario 1:

Linear: 0^2

Ridge: $0^2 + 1 * 3^2 = 9$

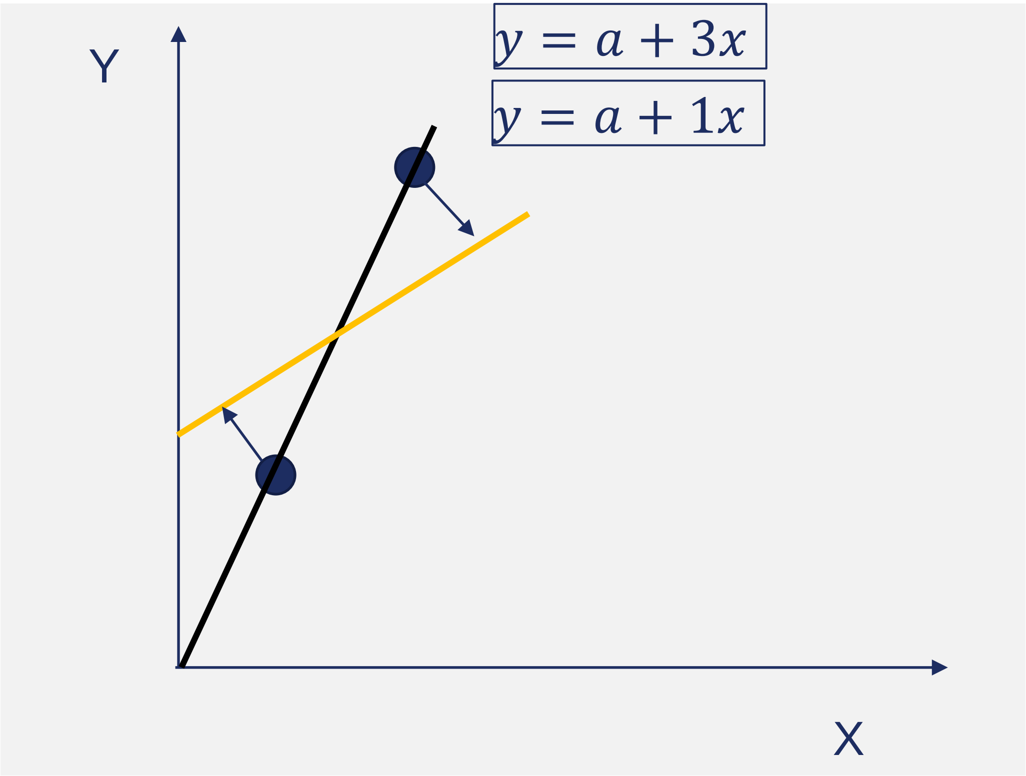
Scenario 2:

Linear: $1^2 + 1^2 = 2$

Ridge: $(1^2+1^2) + 1 * 1^2 = 3$

Ridge - Conclusion

Visualization



Key ideas

Linear Regression finds the best fit

Ridge Regression penalizes extreme coefficients

How? Introduces Bias to decrease volatility

Ridge Regression penalizes overfitting

Ridge Regression is useful when you don‘t have a lot of data points

Bias Coefficient: value between 0 and infinite that you can tune. The default is 0

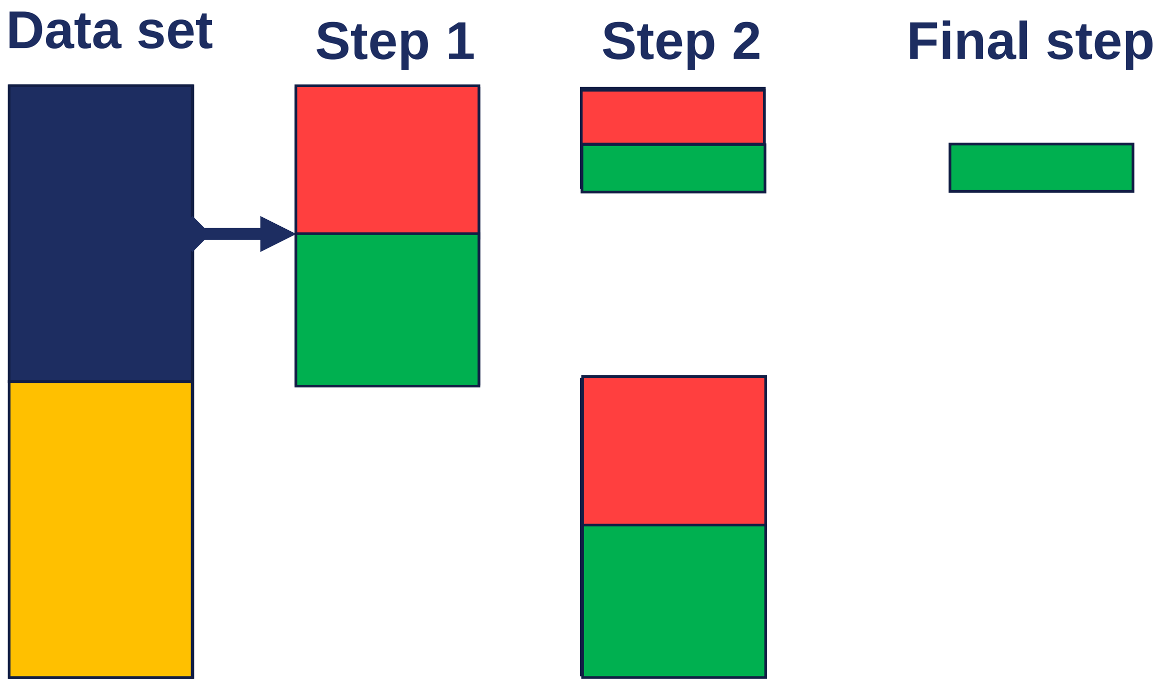
XGBoost is a state-of-art Machine Learning Algorithm

Description

1. Stands for Extreme Gradient Boosting
2. It is an Ensemble Algorithm
3. Has Boosting and Feature Sampling features
4. Can be used for both Regression and Classification
5. XGBoost treats NA‘s as information
6. Poor at dealing with time/growth
7. Excellent dealing with non-linear relationships

Boosting is the secret sauce of XGBoost

Visualization



Description

Step 1: Take random sample without replacement to create model 1

Step 2: take random sample without replacement and add some of the wrongly predicted data in step 1

The wrongly predicted data will have a greater weight than the regular data

Final Step: Focus on the observations that are getting wrong and right predictions

The final prediction will be with majority vote

Boosting: XGBoost gives different weights depending on how difficult it is to predict

First Iteration / Learner

Outcome		Predictor	Weight
✓ 1	←	X	25%
✓ 0	←	X	25%
✗ 0	←	X	25%
✗ 1	←	X	25%

Second Iteration / Learner

Outcome		Predictor	Weight
✗ 0	←	X	40%
✓ 0	←	X	20%
✗ 0	←	X	20%
✓ 1	←	X	20%

Third Iteration / Learner

Outcome		Predictor	Weight
✓ 1	←	X	45%
✓ 1	←	X	35%
✗ 1	←	X	20%

Key Idea

XGBoost only looks at a fraction of the observation at the time
Observations that are more difficult to predict are given a bigger weight

Feature Sampling: XGBoost also gives different weights to different predictors

First Iteration / Learner

Error	Outcome	X1	X2	X3
✗	1	50%	50%	
✗	0			
✗	1			
✓	1			

Second Iteration / Learner

Error	Outcome	X1	X2	X3
✗	1	50%		50%
✗	0			
✓	0			
✓	1			

Third Iteration / Learner

Error	Outcome	X1	X2	X3
✗	1		40%	60%
✓	1			
✗	0			
✓	0			

Key Idea

Predictors also have different weights if they yield different model results

Feature Sampling: XGBoost also gives different weights to different predictors

First Iteration / Learner

Error	Outcome	X1	X2	X3	Weight
Yes	1	50%	50%		25%
Yes	0				25%
Yes	1				25%
No	1				25%

Second Iteration / Learner

Error	Outcome	X1	X2	X3	Weight
Yes	1	50%		50%	30%
Yes	0				30%
No	0				30%
No	0				10%

Third Iteration / Learner

Error	Outcome	X1	X2	X3	Weight
Yes	1		40%	60%	35%
No	1				35%
No	0				25%
No	0				5%

Key Idea

Predictors also have different weights if they yield different model results

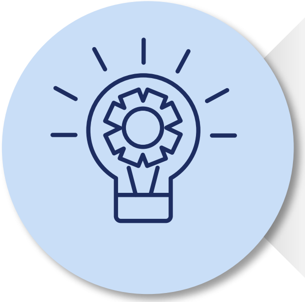
Pros and Cons - Silverkite

Pros		Cons	
Great Accuracy	1	1	Not beginner friendly
Parameter Tuning does not take long	2	2	Customization is complex
Seasonalities and Fitting algorithms	2		

RNN LSTM

Neural Networks quick facts

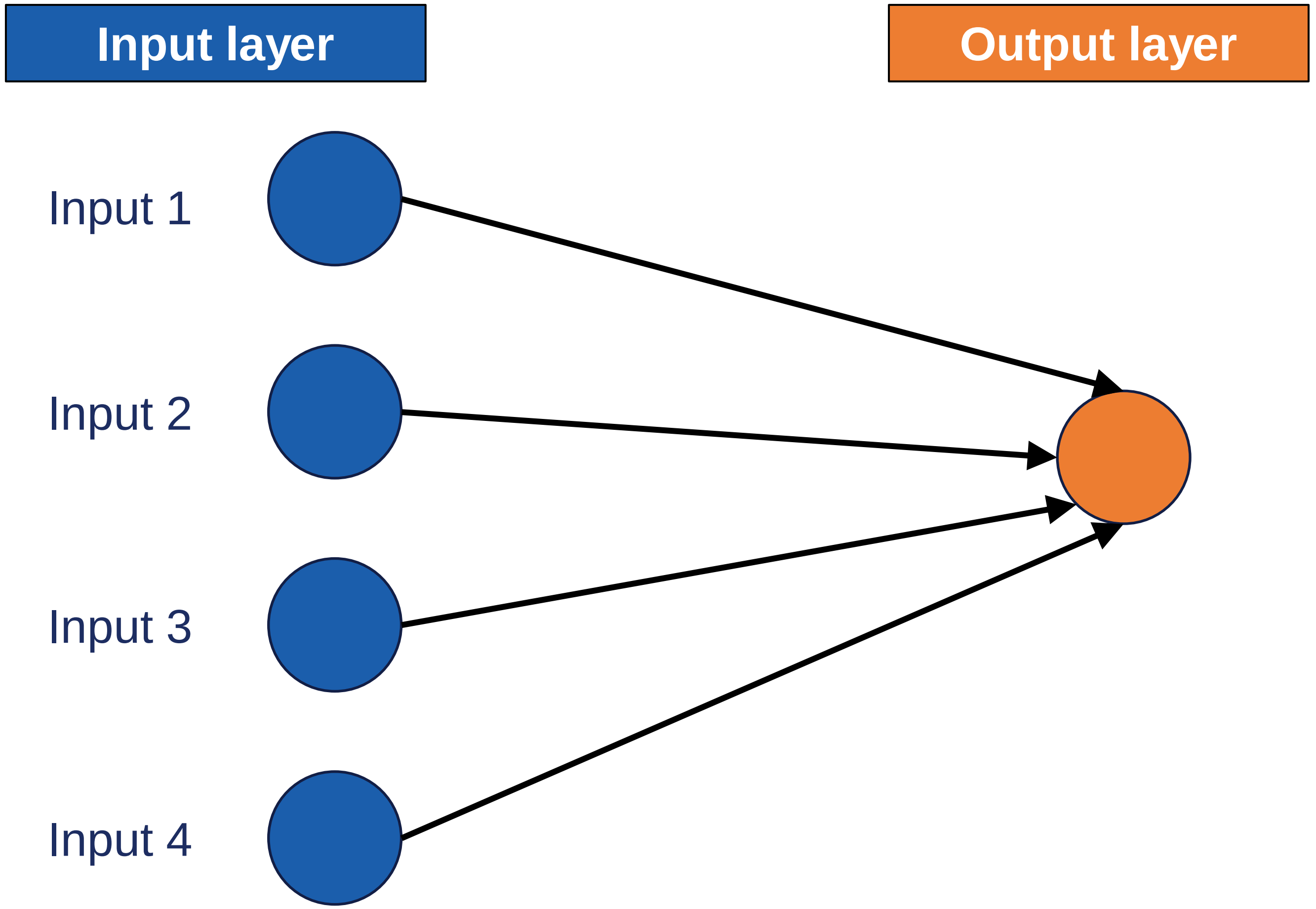
Which?



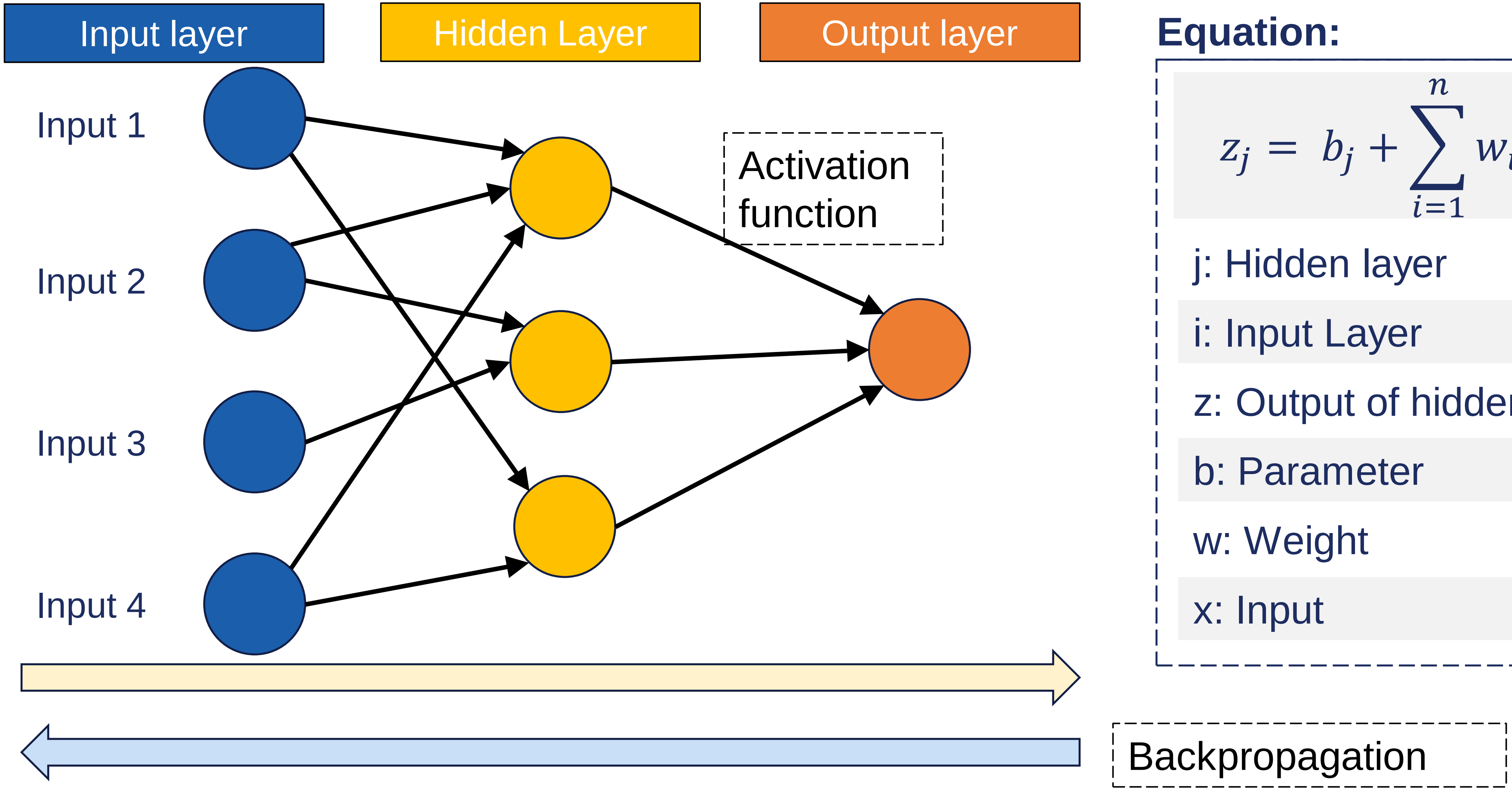
Description

1. Idea comes from the 1940‘s
2. Name comes from working like the synapses in our brain
3. Neurons or nodes have weights that get adjusted as the learning proceeds
4. There is an element of randomness. We would always get different results
5. Recurrent Neural Networks – Advanced form of Neural Networks

Multilinear Regression architecture



Simple Neural Network architecture



Equation:

$$z_j = b_j + \sum_{i=1}^{n} w_{i,j} * x_i$$

j: Hidden layer

i: Input Layer

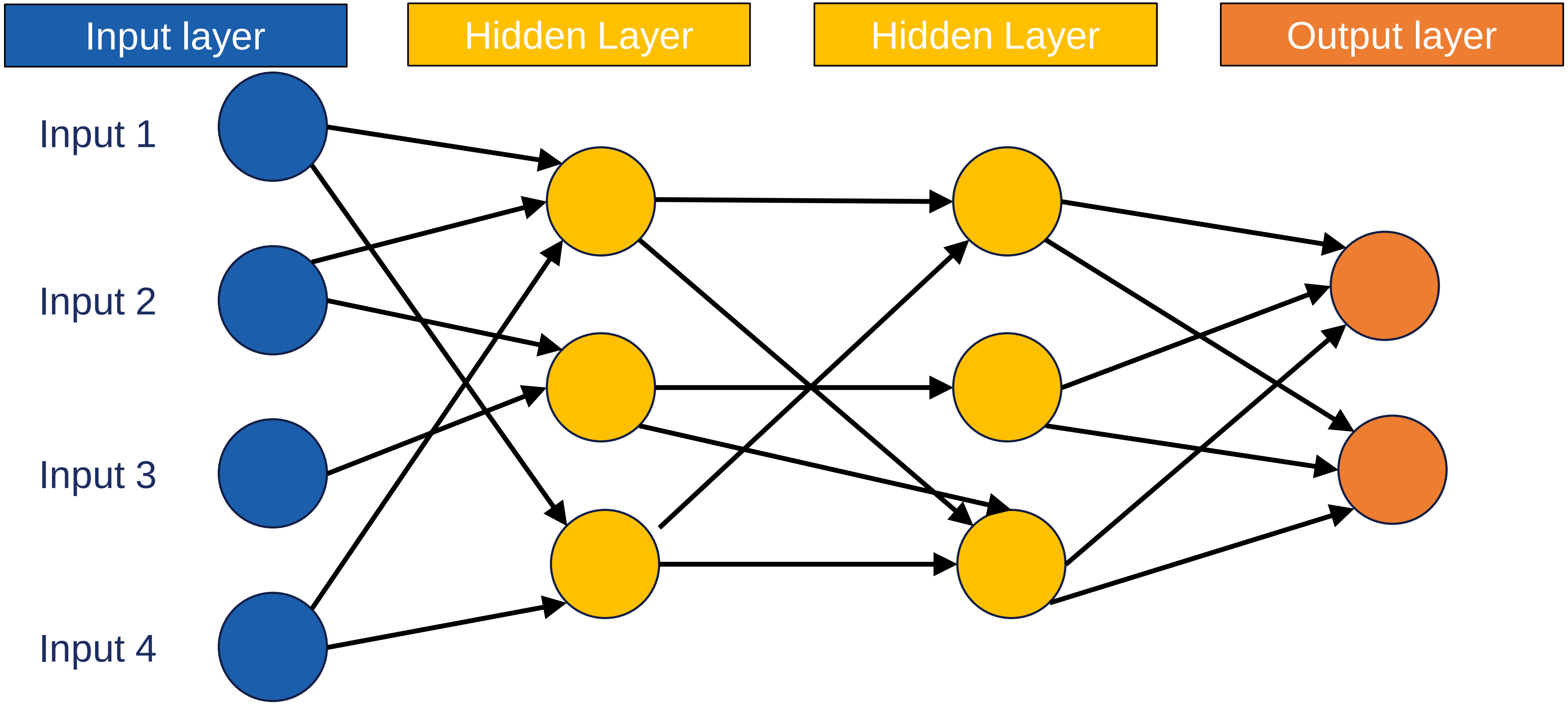
z: Output of hidden layer

b: Parameter

w: Weight

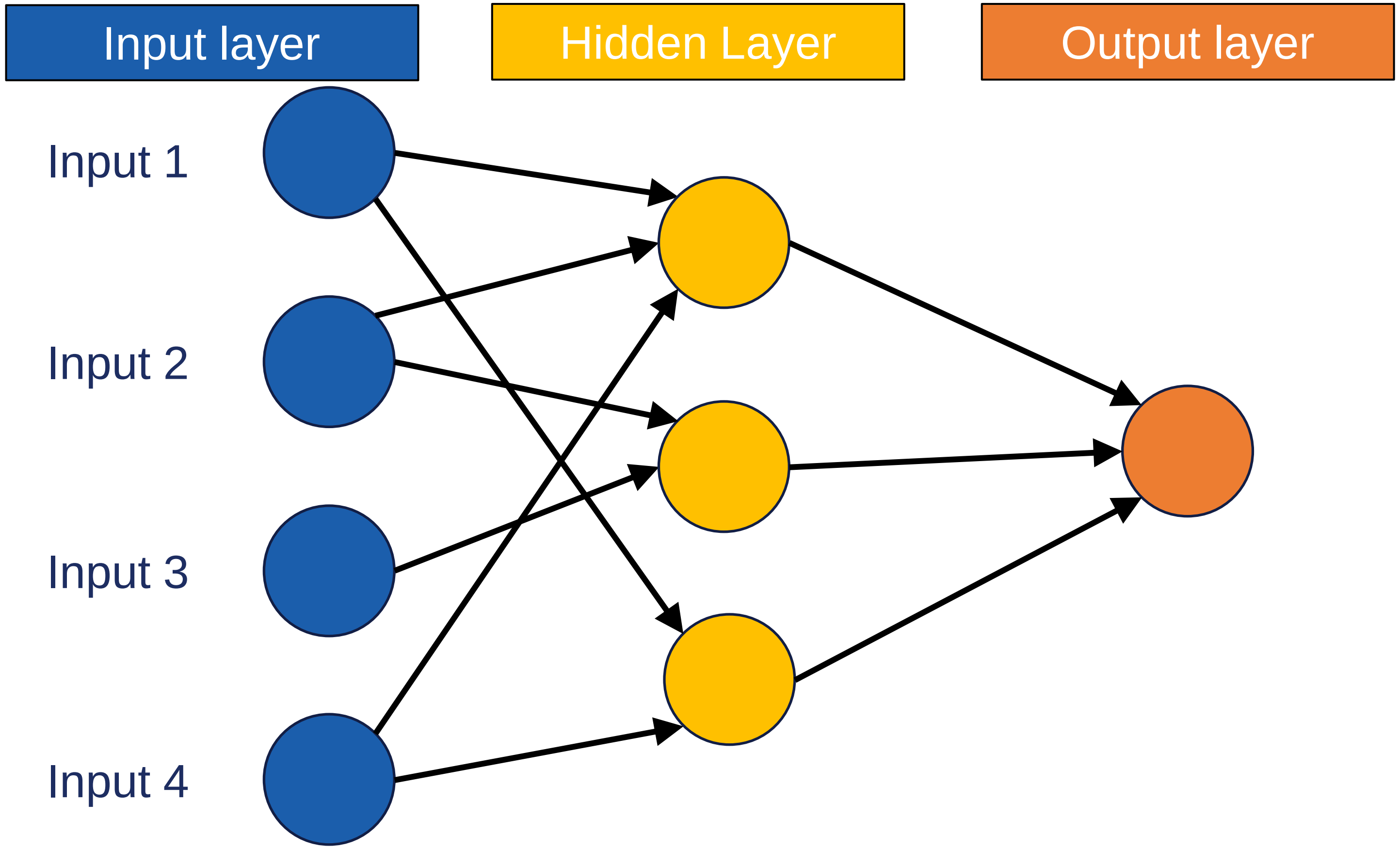
x: Input

Neural Networks can have multiple Hidden Layers and outputs

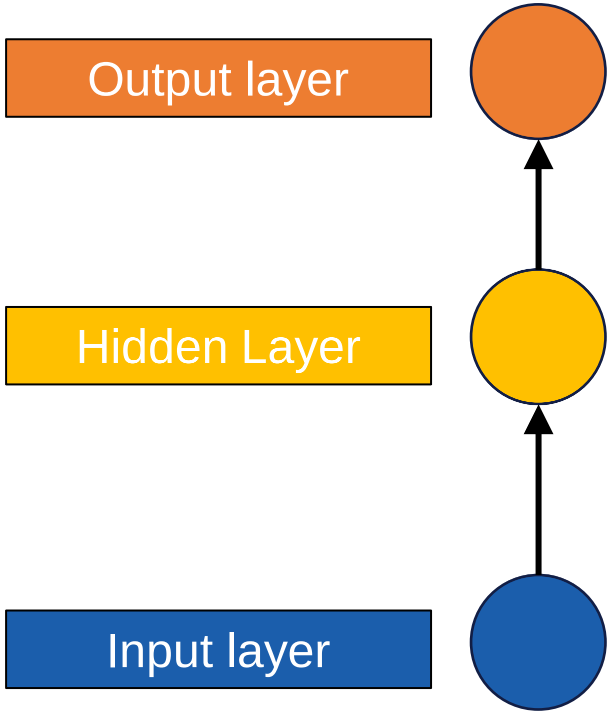


Simplified Neural Network Visualization

Artificial Neural Network Visualization

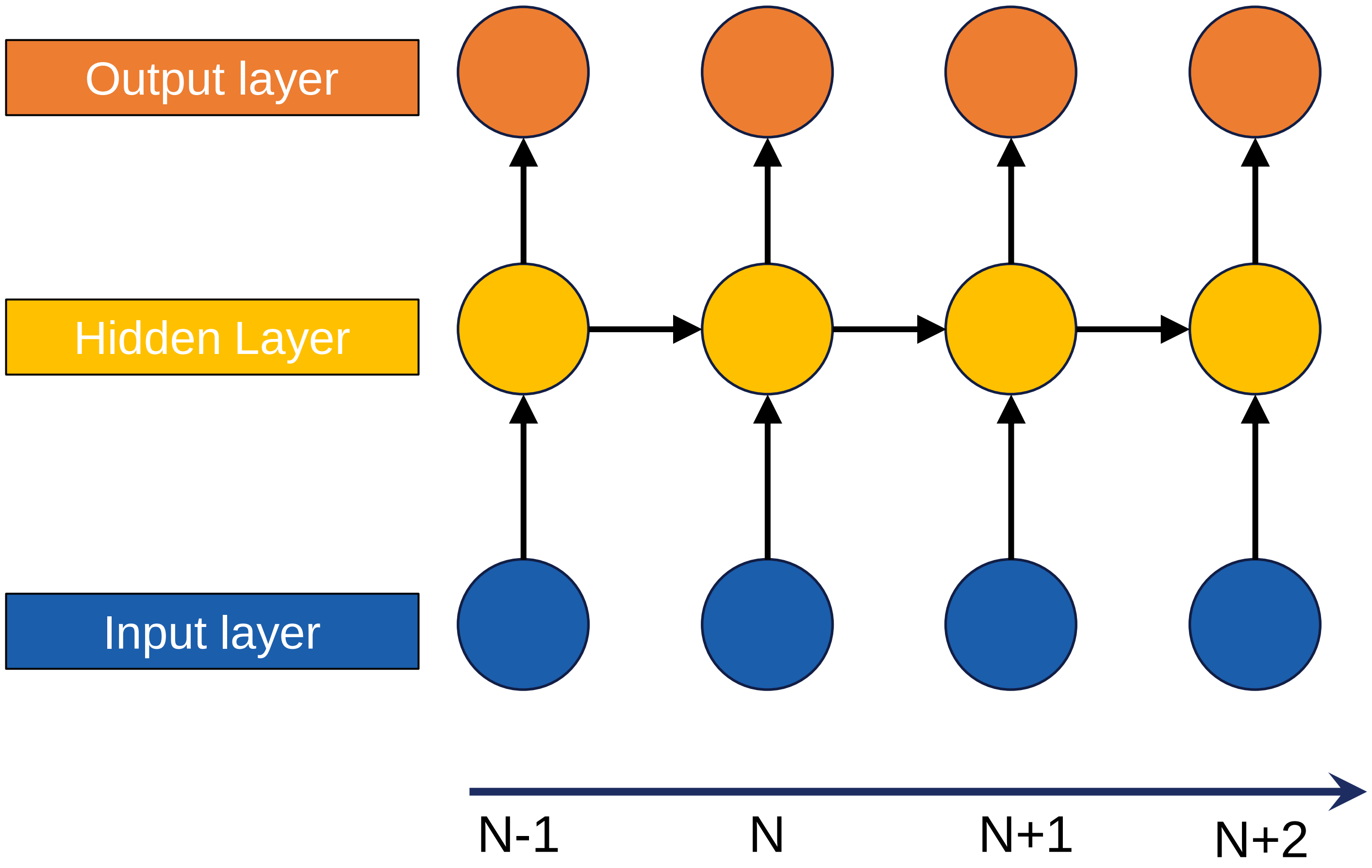


Simplified Visualization



Recurrent Neural Networks architecture

Recurrent Neural Network



Key ideas

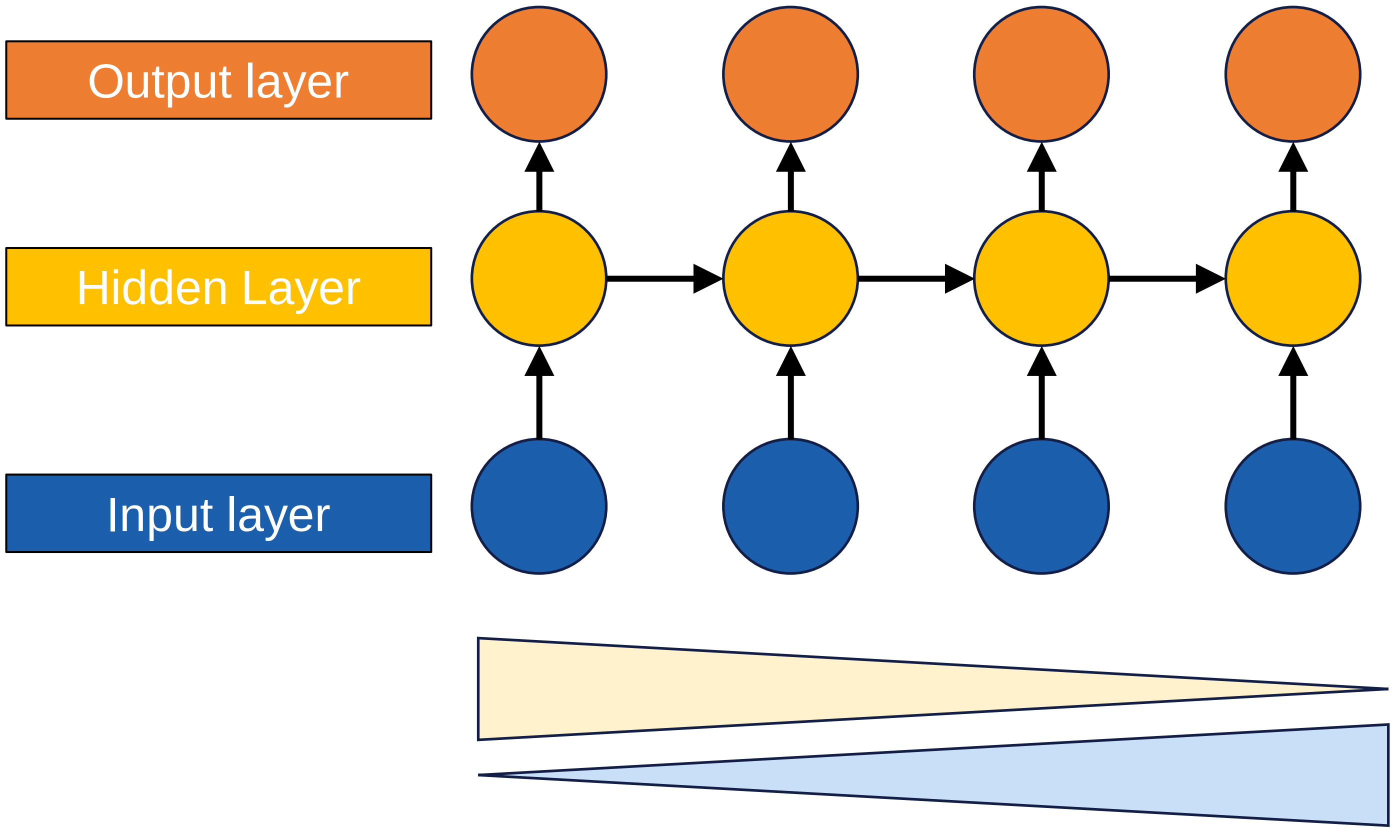
The output at time N is influenced by the inputs at time N and the outputs of N-1.

RNN logic is similar to the other models we have seen.

RNN can be used to create Music or Books.

The issue with RNN

Recurrent Neural Network



Key ideas

The output at time N is influenced by the inputs at time N and the outputs of N-1

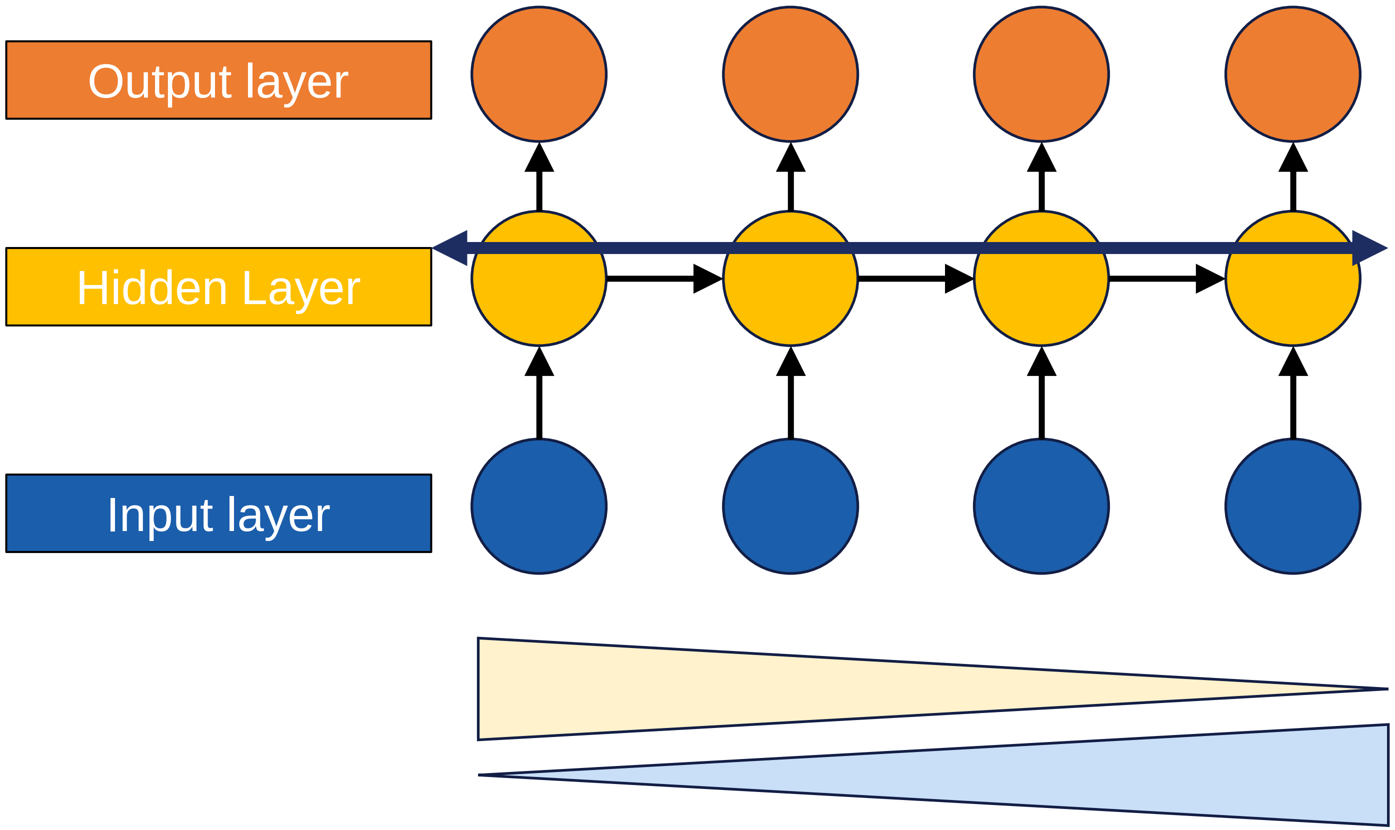
The impact of immediate data is more relevant

The backpropagation also updates more the weights of the last few elements of the series than the initial ones

The initial weights of the series barely get trained

Long Short-Term Memory

Recurrent Neural Network



Key ideas

The output at time N is influenced by the inputs at time N and the outputs of N-1

LSTM has a memory channel that freely flows

It allows the algorithm to have this Long-Memory that has been trained and is updated with every epoch

LONG SHORT-TERM MEMORY
Sepp Hochreiter
Jürgen Schmidhuber

LSTM Model

Component	Description
Dropout	Fraction of neurons ignored
N_rnn_layers	Number of hidden layers
Hidden_dim	Size for feature maps for each hidden RNN layer
N_epochs	Number of complete iterations through the training set
Lr	How much the model learn with the error?
Training_length	Duration of past and future during training. Must be > than ICL
Input_chunk_length	Number of past time steps that are fed to the model

Pros and Cons

Pros	Cons
1. Robust to outliers	1. Low insights
2. Simple to use	2. Requires tuning
3. Great with non-linearity	3. Poor with dealing with trend

Ensemble

Ensemble Introduction

Which?

Description

1. Ensemble is an average of forecasts
2. Forecasting models have advantages and disadvantages
3. Seasonality, trend, regressors, short-term changes..
4. Combining models is a solution to overcome flaws
5. The Last Mile starts now. Are you ready

Ensemble mechanism

Example

Date	Prophet	SARIMAX	Silverkite	LSTM	Ensemble
t	750	850	825	775	800

Key Idea

Ensemble is an average of models. The goal models have flaws, but if you group all of them, then some models will average out the error

Date	Prophet	SARIMAX	Silverkite	LSTM	Average
Historic RMSE	48.1	60	47.8	83.4	59.8

$$Weight = \frac{0.25}{\frac{error}{avg\ error}}$$

Penalizing Models with higher average error

Example

Date	Prophet	SARIMAX	Silverkite	LSTM	Ensemble
FC t	750	850	825	725	800
Weights FC t	187.5	212.5	206.3	193.2	800
New FC t	223.6	201	253.4	132.1	810.1

Date	Prophet	SARIMAX	Silverkite	LSTM	Average
Historic RMSE	48.1	60	47.8	83.4	59.8
Weights	31.1%	24.9%	31.3%	17.9%	25%

$$Weight = \frac{0.25}{\frac{error}{avg\ error}}$$

$$\boxed{Weight = \frac{0.25}{\frac{error}{avg\ error}} / excess}$$

31.3% + 24.9% + 31.3% + 17.9% = 1.05

Pros and Cons

+

1. Accuracy

−

1. Preparation
2. No explanatory power