Deep Neural Network

AIML Module 9

Seetha Parameswaran

BITS Pilani

The author of this deck, Prof. Seetha Parameswaran, is gratefully acknowledging the authors who made their course materials freely available online.

Time Series Modelling and Forecasting

What we Learn….

9.1 Univariate, Multivariate and Multi-step CNN Models

9.2 Univariate, Multivariate and Multi-step LSTM Models

Time Series Data

Time series Data

- General definition: “A time series is a collection of observations made
- sequentially through time, whose dynamics is often characterized by short/long period fluctuations (seasonality and cycles) and/or long period direction (trend)”.
- Such observations may be denoted by $X_1, X_2, X_3, \ldots X_t, \ldots, X_T$ since data are usually collected at discrete points in time.
  - The interval between observations can be any time interval (seconds, minute, hours, days, weeks, months, quarters, years, etc.) and we assume that these time **periods are equally spaced**.
  - One of the most distinctive characteristics of a time series is the mutual dependence between the observations, generally called SERIAL CORRELATION OR **AUTOCORRELATION.**

Time series example 1

Numbers of Doctorates Awarded in US, annual data – Engineering Vs. Education

At a glance

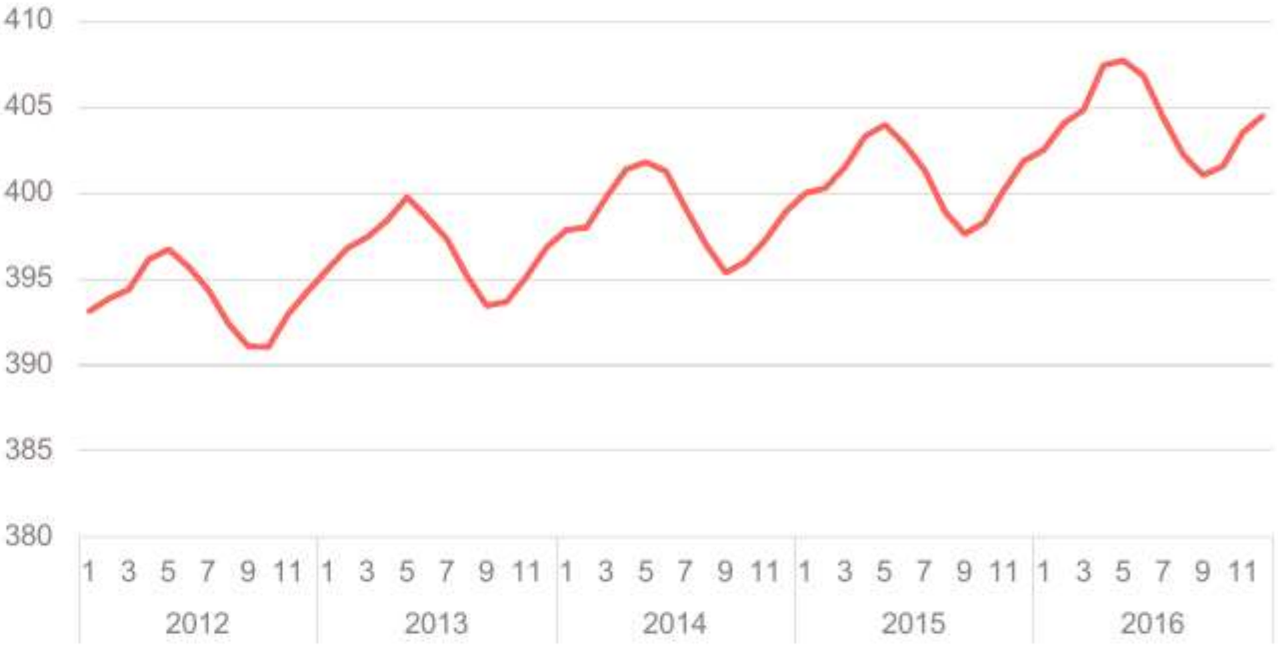
Annual data

Different «directions»

No big fluctuations

Time series example 2

Monthly carbon dioxide concentration (globally averaged over marine surface sites)



At a glance

Monthly basis data

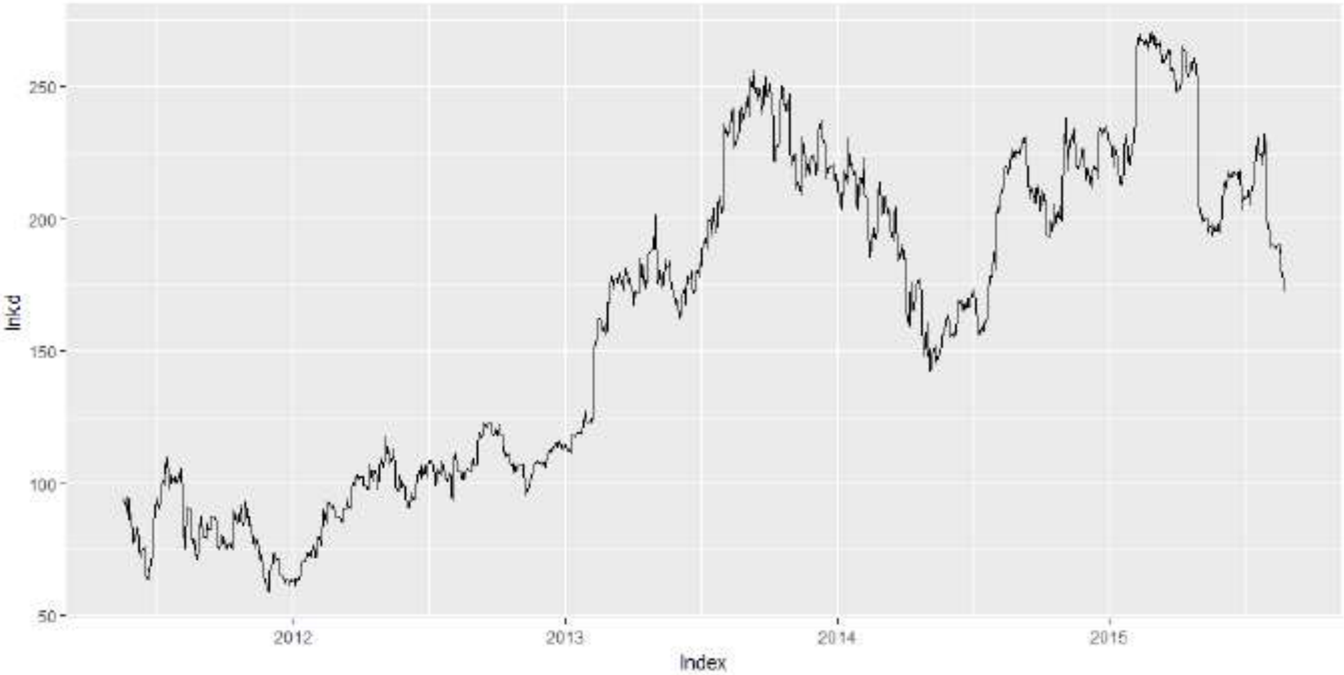
Regular pattern

Constant fluctuations

Average value increases year by year

Time series example 3

LinkedIn daily stock market closing price



At a glance

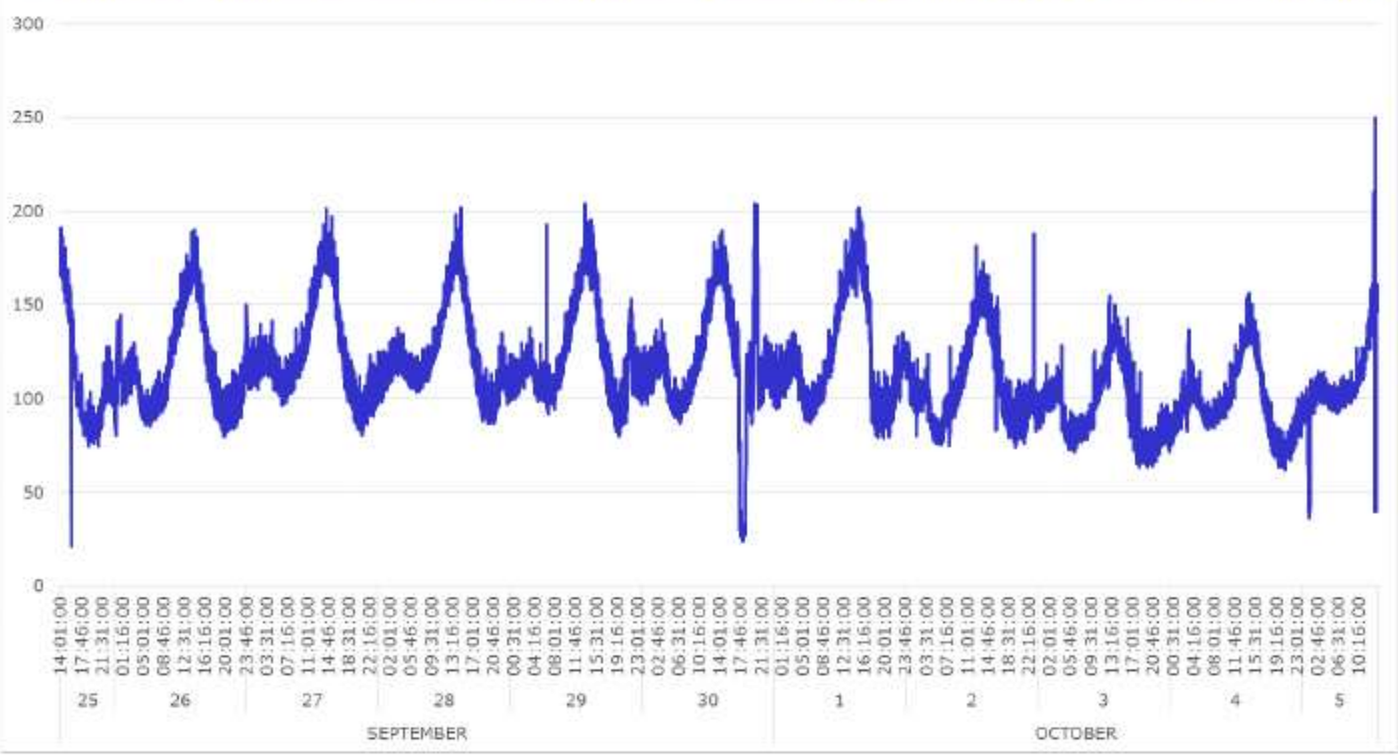
Daily basis data

Very irregular dynamic

Many sudden changes

Time series example 4

Number of photos uploaded on the Instagram every minute (regional sub-sample)



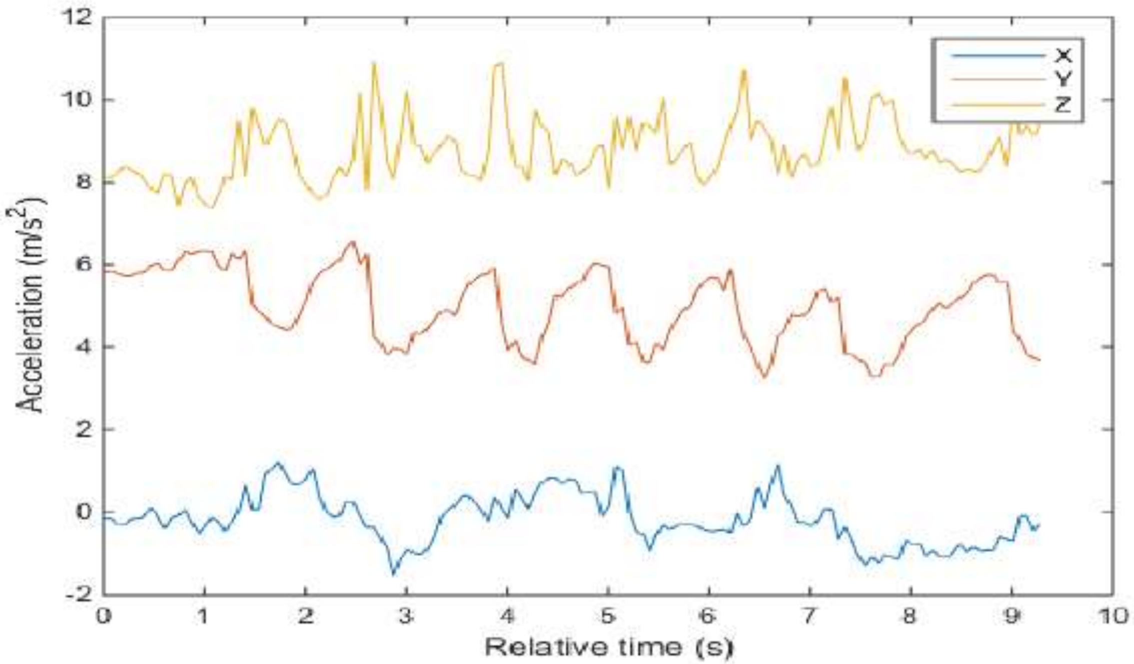
At a glance

Minute basis data

Almost regular daily pattern but with some anomalies and spikes

Time series example 5

Acceleration detected by a smartphone sensors during a workout session (10 seconds)



At a glance

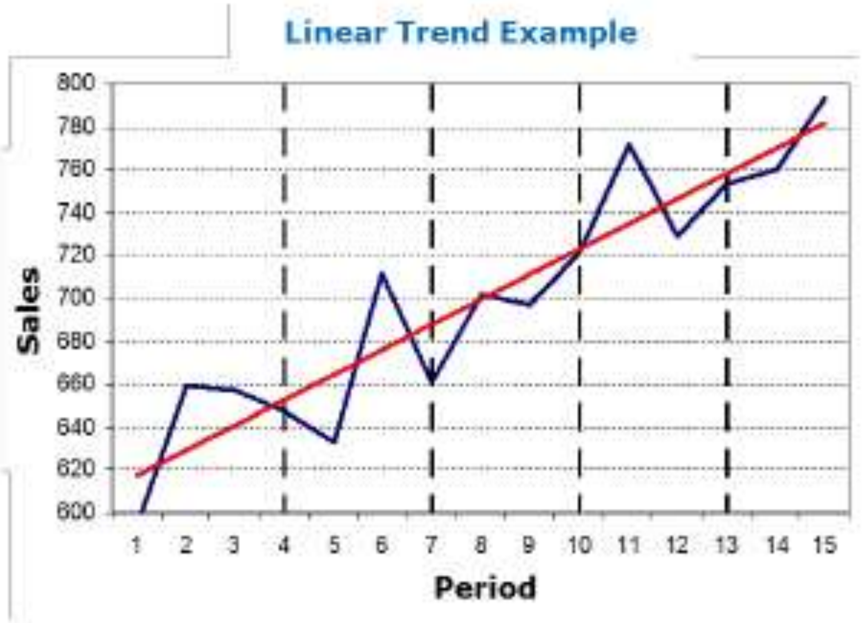
Milliseconds basis data

Each sensor has its own dynamics

Time Series Data Properties: Main Elements

1. TREND

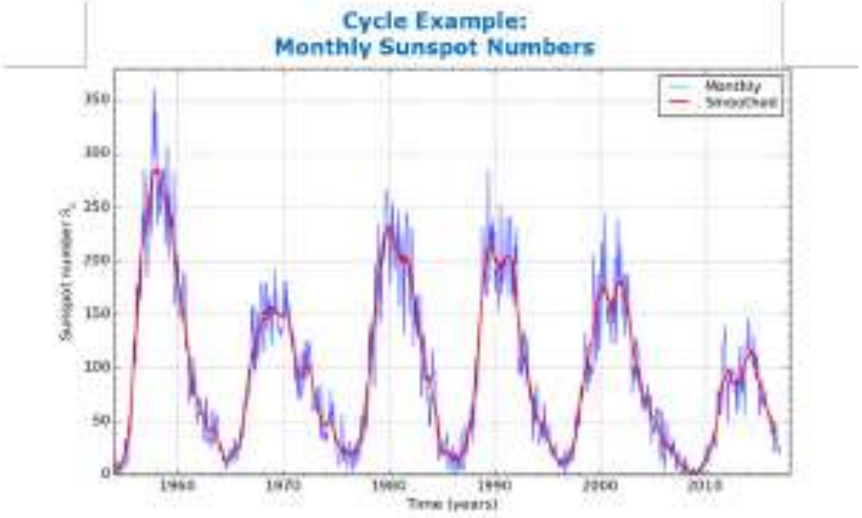
- The general direction in which the series is running during a long period.
- A TREND exists when there is a long-term increase or decrease in the data.
- It does not have to be necessarily linear (could be exponential or others functional form).



Time Series Data Properties: Main Elements

2. CYCLE

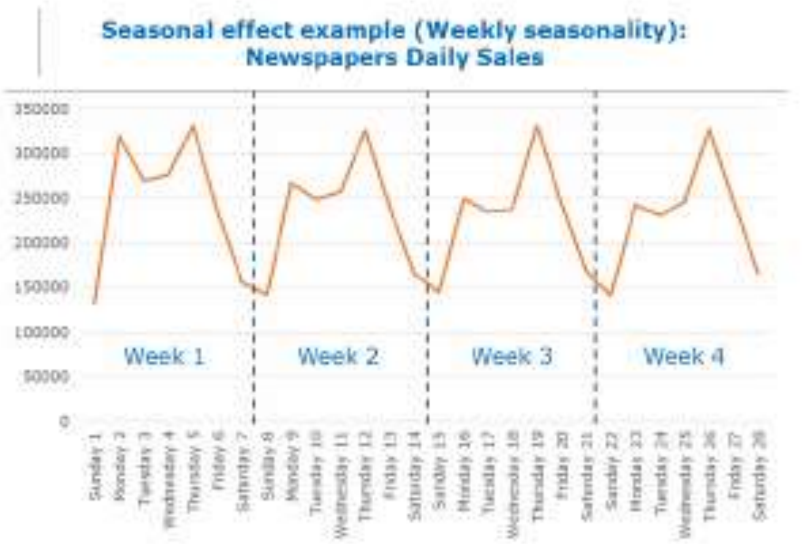
- Long-term fluctuations that occur regularly in the series.
- A CYCLE is an oscillatory component (i.e. Upward or Downward swings) which is repeated after a certain number of years, so:
    - May vary in length and usually lasts several years (from 2 up to 20/30)
    - Difficult to detect, because it is often confused with the trend component.



Time Series Data Properties: Main Elements

3. SEASONAL EFFECTS

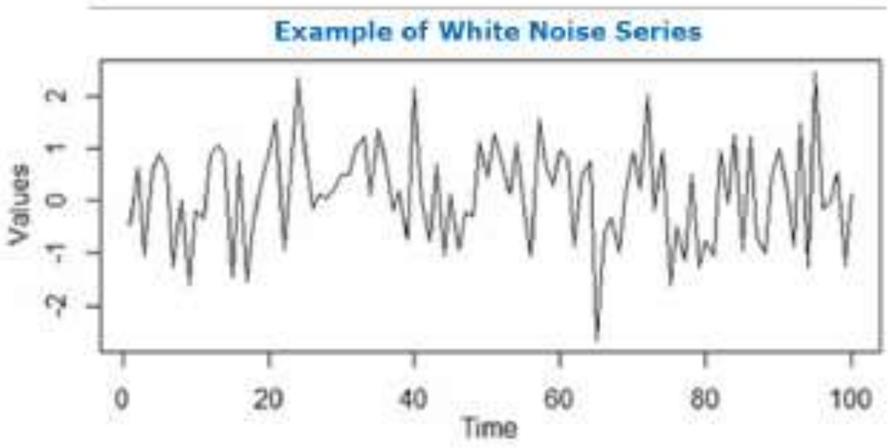
- Short-term fluctuations that occur regularly - often associated with months or quarters.
- A SEASONAL PATTERN exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, day of the week).
- Seasonality is always of a fixed and known period.



Time Series Data Properties: Main Elements

4. RESIDUAL

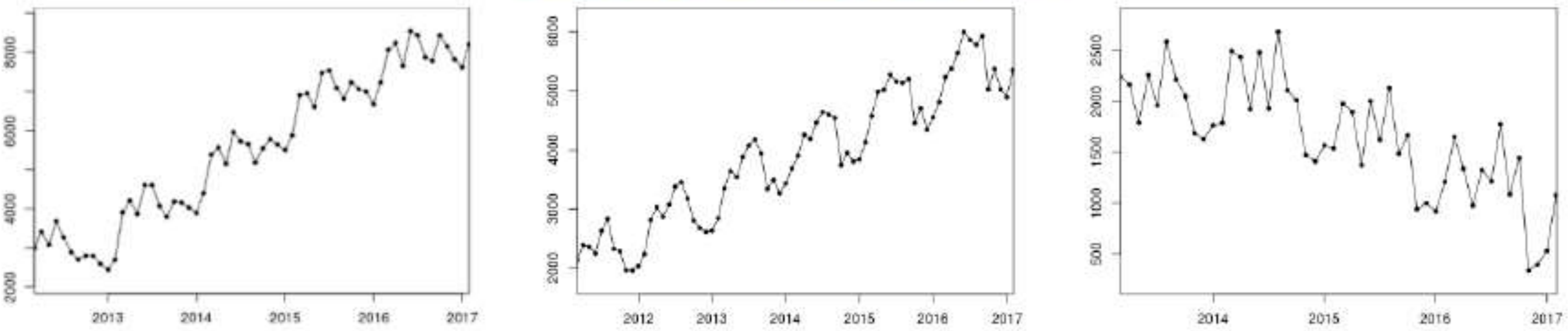
- Whatever remains after the other components have been taken into account.
- The residual/error component is everything that is not considered in previous components.
- Typically, it is assumed to be the sum of a set of random factors (e.g. a white noise series) not relevant for describing the dynamics of the series.



Seasonal effect: Additive Seasonality

- When the seasonality in **Additive**, the dynamics of the components are independents from each other; for instance, an increase in the trend-cycle will not cause an increase in the magnitude of seasonal dips.
- The difference of the trend and the raw data is **roughly constant in similar periods of time** (months, quarters) irrespectively of the tendency of the trend.

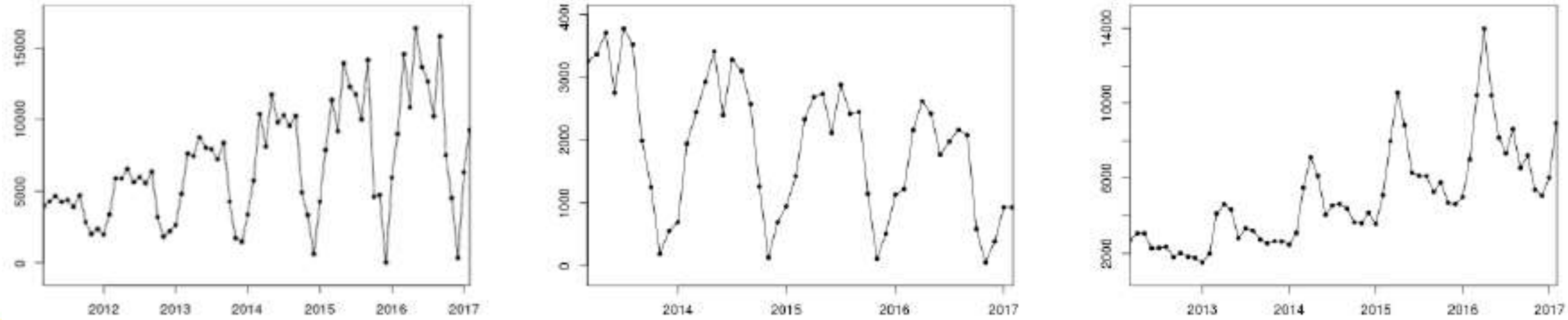
EXAMPLES OF ADDITIVE SEASONALITY



Seasonal effect: Multiplicative Seasonality

- In the multiplicative model the amplitude of the seasonality increase (decrease) with an increasing (decreasing) trend, therefore, on the contrary to the additive case, the **components are not independent from each other.**
- When the variation in the seasonal pattern (or the variation around the trend-cycle) **appears to be proportional** to the level of the time series, then a multiplicative model is more appropriate.

EXAMPLES OF MULTIPLICATIVE SEASONALITY



Seasonal effect: Frequency

- According to the data granularity and to the type of seasonality you want to model, it is important to consider the right seasonal frequency (i.e. how many observations you have for every seasonal cycle)
  - For annual seasonality, the data points are years, quarters, months or weeks (in this case you will face only annual seasonality), but if the frequency of observations is smaller than a week, things get more comp

Frequency		Cycle type			
		Hour	Day	Week	Year
Data granularity	Annual				1
	Quarterly				4
	Monthly				12
	Weekly			1	52.18
	Daily		1	7	365.25
	Hourly	1	24	168	8766
	Minutes	60	1440	10080	525960

*Every year, on average, is made up of 365 days and 6 hours → so 365.25 days and 365.25/7=52.18 weeks

Numerical and Graphical description of Time Series

- The first step in Time Series Analysis is to produce a detailed exploratory analysis of the data to get some insights about the distribution of the series over time.

Graphical descriptive analyses

- Time plot
- Seasonal plot
- Box plot analysis
- Scatterplots (Lag plots)
- Plotting auto-correlation and cross-correlation functions

Numerical descriptive analyses

- Sampling period evaluation (start, end, data points features)
- Number of data available
- Missing value and outlier evaluation
- Frequency distribution analysis
- Summary descriptive statistics (overall and by season)

Time Series Data and Analysis

Time Series Analysis

- **Time series analysis** is a specific way of analyzing a sequence of data points collected over an interval of time.
- **Time** is a crucial variable because it shows how the data adjusts over the course of the data points as well as the final results. It provides an additional source of information and a set order of dependencies between the data.
- **Time series analysis** involves developing models to gain an understanding of the data to understand the underlying causes. Analysis can provide the “why” behind the outcomes you are seeing.
- **Forecasting** then takes the next step of what to do with that knowledge and the predictable extrapolations of what might happen in the future.

Main Objectives of Time Series Analysis

- Summary description (graphical and numerical) of data point vs. time
- Interpretation of specific series features (e.g. seasonality, trend, relationship with other series)
- Forecasting (e.g. predict the series values in (t + 1), (t + 2), ... , (t + k))
- Hypothesis testing and Simulation (comparing different scenarios)

Time Series Forecasting

Time Series Forecasting

- **Time series forecasting** is the process of analyzing time series data using statistics and modeling to make predictions and inform strategic decision-making.
- Time series forecasting is part of **predictive** analytics.
- It can show likely changes in the data, like **seasonality or cyclic behavior**, which provides a better understanding of data variables and helps forecast better.

Challenges in Time Series Forecasting

- At the time of the work, the future outcome is completely unavailable and can only be estimated through careful analysis and evidence-based priors.
- Likelihood of forecasts can vary wildly, especially when dealing with the commonly fluctuating variables in time series data as well as factors outside our control.

Applications of Time Series Forecasting

- Weather forecasting
- Climate forecasting
- Economic forecasting
- Healthcare forecasting
- Engineering forecasting
- Finance forecasting
- Retail forecasting
- Business forecasting
- Environmental studies forecasting
- Social studies forecasting

Time series forecasting considerations

1. Time horizons
   - The time frame of your forecast is known as a time horizon.
   - Time horizon is a fixed point in time where the forecast ends.
   - It's much easier to forecast a shorter time horizon with fewer variables than it is a longer time horizon.
2. Dynamic and static states
   - If the forecast is static, it is set in stone once it is made, so make sure your data is adequate for a forecast.
   - Dynamic forecasts can be constantly updated with new information as it comes in. This means you can have less data at the time the forecast is made, and then get more accurate predictions as data is added.

Time series forecasting considerations

3. Data quality
    - Typical guidelines for data quality apply.
        - Make sure data is complete,
        - Data is not duplicated or redundant,
        - Data was collected in a timely and consistent manner,
        - Data is in a standard and valid format,
        - Data is accurate for what it is measuring,
        - Data is uniform across sets.
        - Data is collected at consistent intervals over the period of time being tracked. This helps account for trends in the data, cyclic behavior, and seasonality. It also can help identify if an outlier is truly an outlier or if it is part of a larger cycle. Gaps in the data can hide cycles or seasonal variation, skewing the forecast as a result.

Limitations of Classical Methods for TS Forecasting

1. **Focus on complete data**: missing or corrupt data is generally unsupported.
2. **Focus on linear relationships**: assuming a linear relationship excludes more complex joint distributions.
3. **Focus on fixed temporal dependence**: the relationship between observations at different times, and the number of lag observations provided as input, must be diagnosed and specified.
4. **Focus on univariate data**: many real-world problems have multiple input variables.
5. **Focus on one-step forecasts**: many real-world problems require forecasts with a long time horizon.

Strengths of DNN for TS Forecasting

1. **Robust to Noise**. Neural networks are robust to noise in input data and in the mapping function and can support learning and prediction in the presence of missing values.
2. **Nonlinear**. Neural networks do not make strong assumptions about the mapping function and readily learn linear and nonlinear relationships.
3. **Multivariate Inputs**. An arbitrary number of input features can be specified, providing direct support for multivariate forecasting.
4. **Multi-step Forecasts**. An arbitrary number of output values can be specified, providing direct support for multi-step and even multivariate forecasting.

Strengths of DNN for TS Forecasting

5. **Feature Learning in CNN**. Automatic identification, extraction and distillation of salient features from raw input data that pertain directly to the prediction problem that is being modeled.
6. **Native Support for Sequences.** Recurrent neural networks directly add support for input sequence data.
7. **Learned Temporal Dependence.** The most relevant context of input observations to the expected output is learned and can change dynamically.

Limitations of DNN for TS Forecasting

1. The mapping function is fixed or static.
2. Fixed Inputs. The number of lag input variables is fixed, in the same way as traditional time series forecasting methods.
3. Fixed Outputs. The number of output variables is also fixed; although a more subtle issue, it means that for each input pattern, one output must be produced.

Framework for Time Series Forecasting

Framework

Time series forecasting involves **developing and using a predictive model** on data where there is an **ordered relationship between observations.**

Framework

1. What are the inputs and outputs for a forecast?
    a. Inputs: Historical data provided to the model in order to make a single forecast.
    b. Outputs: Prediction or forecast for a future time step beyond the data provided as input.
2. What are the endogenous and exogenous variables?
    a. Endogenous: Input variables that are influenced by other variables in the system and on which the output variable depends.
    b. Exogenous: Input variables that are not influenced by other variables in the system and on which the output variable depends.
3. Are you working on a regression or classification predictive modeling problem?
    a. ˆRegression: Forecast a numerical quantity.
    b. ˆClassification: Classify as one of two or more labels.

Framework

4. Are the time series variables unstructured or structured?
    a. Unstructured: No obvious systematic time-dependent pattern in a time series variable.
    b. Structured: Systematic time-dependent patterns in a time series variable (e.g. trend and/or seasonality).
5. Are you working on a univariate or multivariate time series problem?
    a. Univariate: One variable measured over time.
    b. Multivariate: Multiple variables measured over time.
    c. Univariate and Multivariate Inputs: One or multiple input variables measured over time.
    d. Univariate and Multivariate Outputs: One or multiple output variables to be predicted.

Framework

6. Do you require a single-step or a multi-step forecast?
    a. One-step: Forecast the next time step.
    b. ˆMulti-step: Forecast more than one future time steps.
7. Do you require a static or a dynamically updated model?
    a. ˆStatic. A forecast model is fit once and used to make predictions.
    b. ˆDynamic. A forecast model is fit on newly available data prior to each prediction.
8. Are your observations contiguous or discontiguous?
    a. ˆContiguous. Observations are made uniform over time.
    b. ˆDiscontiguous. Observations are not uniform over time.

Steps to Develop a Forecasting Model

1. Define Problem.
   a. Inputs vs. Outputs: What are the inputs and outputs for a forecast?
   b. Endogenous vs. Exogenous: What are the endogenous and exogenous variables?
   c. Unstructured vs. Structured: Are the time series variables unstructured or structured?
   d. Regression vs. Classification: Are you working on a regression or classification predictive modeling problem? What are some alternate ways to frame your time series forecasting problem?
   e. Univariate vs. Multivariate: Are you working on a univariate or multivariate time series problem?
   f. Single-step vs. Multi-step: Do you require a single-step or a multi-step forecast?
   g. Static vs. Dynamic: Do you require a static or a dynamically updated model?
   h. Contiguous vs. Discontiguous: Are your observations contiguous or discontiguous?

   Some useful tools to help get answers include:

   a. ˆData visualizations (e.g. line plots, etc.).
   b. ˆStatistical analysis (e.g. ACF/PACF plots, etc.).

Steps to Develop a Forecasting Model

2. Design Test Harness
    a. Split the dataset into a train and test set.
    b. Fit a candidate approach on the training dataset.
    c. Make predictions on the test set.
    d. Calculate a metric that compares the predictions to the expected values.

Steps to Develop a Forecasting Model

3. Test Models.: Test many models using your test harness.
    a. Baseline. Simple forecasting methods such as persistence and averages.
    b. Autoregression. The Box-Jenkins process and methods such as SARIMA.
    c. Exponential Smoothing. Single, double and triple exponential smoothing methods.
    d. Linear Machine Learning. Linear regression methods and variants such as regularization.
    e. Nonlinear Machine Learning. kNN, decision trees, support vector regression and more.
    f. Ensemble Machine Learning. Random forest, gradient boosting, stacking and more.
    g. Deep Learning. MLPs, CNNs, LSTMs, and Hybrid models.

Steps to Develop a Forecasting Model

4. Finalize Model.
   a. List of the top 5 to 10 candidate models that are skillful on the problem.
   b. Pick one or multiple models and finalize them. This involves training a new final model on all available historical data (train and test). The model is ready for use.

Prepare Time Series Data for DNNs

Prepare Time Series Data for DNNs

Objectives

- Transform a time series dataset into a two-dimensional supervised learning format.
- Transform a two-dimensional time series dataset into a three-dimensional structure suitable for CNNs and LSTMs.
- Split a very long time series into subsequences ready for training a CNN or LSTM model.

Time Series Data to 2D Supervised Learning Data

Univariate Time Series Data

time,	measure
1,	100
2,	110
3,	108
4,	115
5,	120

Univariate Time Series Data for supervised learning

X,	y
?,	100
100,	110
110,	108
108,	115
115,	120
120,	?

Using sliding window method.

Time Series Data to 2D Supervised Learning Data

Multivariate Time Series Data

time,	measure1,	measure2
1,	0.2,	88
2,	0.5,	89
3,	0.7,	87
4,	0.4,	88
5,	1.0,	90

Multivariate Time Series Data for supervised learning

X1,	X2,	X3,	y
?,	?,	0.2,	88
0.2,	88,	0.5,	89
0.5,	89,	0.7,	87
0.7,	87,	0.4,	88
0.4,	88,	1.0,	90
1.0,	90,	?,	?

Using sliding window method.

Time Series Data to 2D Supervised Learning Data

Multivariate Time Series Data

time,	measure1,	measure2
1,	0.2,	88
2,	0.5,	89
3,	0.7,	87
4,	0.4,	88
5,	1.0,	90

Multivariate Time Series Data for sequence prediction (multi-step)

X1,	X2,	y1,	y2
?,	?,	0.2,	88
0.2,	88,	0.5,	89
0.5,	89,	0.7,	87
0.7,	87,	0.4,	88
0.4,	88,	1.0,	90
1.0,	90,	?,	?

Using sliding window method.

Time Series Data to 3D Supervised Learning Data

- First convert time series data to 2D data for supervised learning.
- Then convert 2D data to 3D data.
    - The input layer for CNN and LSTM models is specified by the input shape argument on the first hidden layer of the network.
    - The input to every CNN and LSTM layer must be three-dimensional.
        - The three dimensions of this input are [samples, timesteps, features]
            - **Samples**. One sequence is one sample.
            - **Time Steps**. One time step is one point of observation in the sample. One sample is comprised of multiple time steps.
            - **Features**. One feature is one observation at a time step. One time step is comprised of one or more features.

Example: Helper function

```
# transform univariate time series to supervised learning problem
from numpy import array

# split a univariate sequence into samples
def split_sequence(sequence, n_steps):
  X, y = list(), list()
  for i in range(len(sequence)):
    # find the end of this pattern
    end_ix = i + n_steps
    # check if we are beyond the sequence
    if end_ix > len(sequence)-1:
      break
    # gather input and output parts of the pattern
    seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
    X.append(seq_x)
    y.append(seq_y)

  return array(X), array(y)
```

Example: Convert TS data to 2D data

```
# define univariate time series
series = array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print(series.shape)
# transform to a supervised learning problem
X, y = split_sequence(series, 3)
print(X.shape, y.shape)
# show each sample
for i in range(len(X)):
  print(X[i], y[i])
```

```
(10,)

(7, 3) (7,)

[1 2 3] 4
[2 3 4] 5
[3 4 5] 6
[4 5 6] 7
[5 6 7] 8
[6 7 8] 9
[7 8 9] 10
```

Example

```
(10,)
(7, 3) (7,)
(7, 3, 1)
```

```python
# define univariate time series
series = array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print(series.shape)
# transform to a supervised learning problem
X, y = split_sequence(series, 3)
print(X.shape, y.shape)
# show each sample
for i in range(len(X)):
  print(X[i], y[i])
```

CNNs for Time Series Forecasting

Univariate CNN Models

- Univariate time series are datasets comprised of a single series of observations with a temporal ordering and a model is required to learn from the series of past observations to predict the next value in the sequence.
- The CNN model will learn a function that maps a sequence of past observations as input to an output observation.
- A one-dimensional CNN is a CNN model that has a convolutional hidden layer that operates over a 1D sequence.

Demo Code for CNN

- Univariate CNN
  - https://colab.research.google.com/drive/1VXCnwxtRdFkDYlxGqhnEuQeXTjQNc0uM#scrollTo=Ifms-u8BBraF
- Multivariate CNN
  - https://colab.research.google.com/drive/1aADi0xKTExjZSReSFmwTke_Ws4yAwp4k#scrollTo=8kDtsv_uJygm

PS: Use BITS email ID to access files.

RNNs for Time Series Forecasting

Univariate RNN Models

- The LSTM model will learn a function that maps a sequence of past observations as input to an output observation.
- The sequence of observations must be transformed into multiple examples from which the LSTM can learn.

Demo Code for RNN

- Univariate RNN
  - https://colab.research.google.com/drive/1b-ORe5WeYHpTHgcpFKPIoWRQcWNLXBq_#scrollTo=vfZ36VoM9ZIh
- Multivariate RNN
  - https://colab.research.google.com/drive/1s5bsuW-wmxyuPkIK_C4mOwc8JPqlMwwS

PS: Use BITS email ID to access files.

Ref

- Deep Learning for Time Series Forecasting by Jason Brownlee (R4)
- https://www.tableau.com/learn/articles/time-series-forecasting

Next Session:
CNN