Time Series Analysis
&
Forecasting

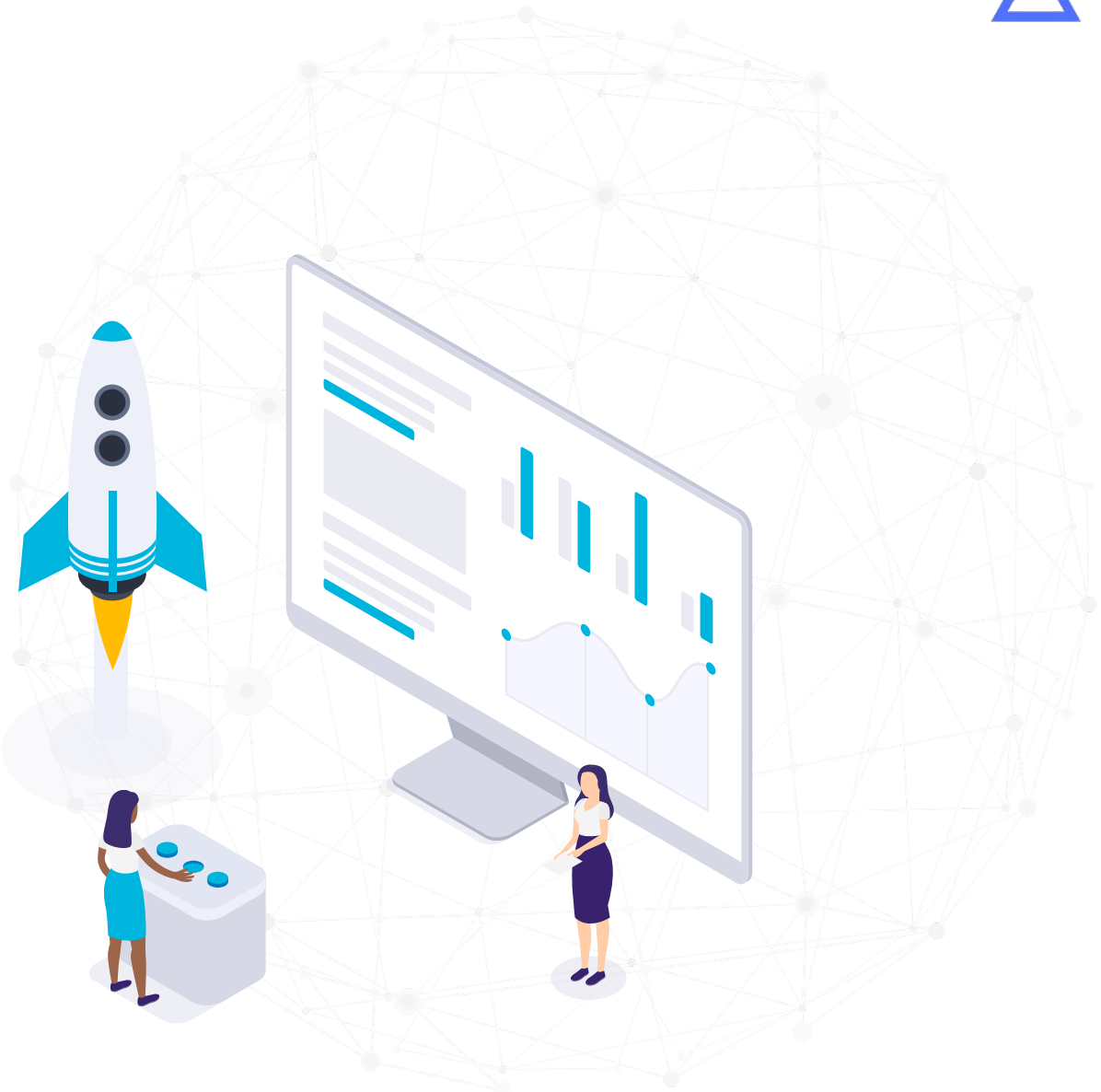
Agenda

1. Introduction to Time Series Forecasting
2. Understanding Time Series Data
3. Pre-processing and Data Cleaning
4. Exploratory Data Analysis for Time Series
5. Time Series Forecasting Models: A Comprehensive Overview
6. Univariate Time Series Forecasting Methods
7. Multivariate Time Series Forecasting Methods
8. Evaluating Forecasting Performance
9. Advanced Time Series Forecasting Techniques
10. Time Series Forecasting in Practice: Case Studies
11. Overcoming Challenges in Time Series Forecasting
12. Future Trends in Time Series Forecasting

Introduction

Time Series Analysis

- Time series forecasting is an important area of machine learning that is often neglected.
- It is important because there are so many prediction problems that involve a time component and these problems are often neglected because it is this time component that makes time series problems more difficult to handle.
- Before getting started with Time Series Analysis, let's get our basics clear on Anomaly Detection



What is Time Series?

It is a series of observations taken at specified times basically at equal intervals. It is used to predict future values based on past observed values.

Components of Time Series

Increasing or Decreasing value in the series

A general systematic linear or (most often) non-linear component that changes over time and does repeat.

The data in the time series follows a temporal sequence, but the measurements might not happen at a regular time interval.

Pattern exists when data exhibit rises & falls that are not of fixed period.

Testing TS Stationarity

There are many methods to check whether a time series is stationary or non-stationary.

1. **Look at Plots:** You can review a time series plot of your data and visually check if there are any obvious trends or seasonality.
2. **Summary Statistics:** You can review the summary statistics for your data for seasons or random partitions and check for obvious or significant differences.
3. **Statistical Tests:** You can use statistical tests to check if the expectations of stationarity are met or have been violated.

You can split your time series into two (or more) partitions and compare the mean and variance of each group. If they differ and the difference is statistically significant, the time series is likely non-stationary.

Testing TS Stationarity

Statistical tests make strong assumptions about your data. They can only be used to inform the degree to which a null hypothesis can be rejected or fail to be reject. The result must be interpreted for a given problem to be meaningful.

Nevertheless, they can provide a quick check and confirmatory evidence that your time series is stationary or non-stationary.

ADF Test is otherwise known as **unit root test.**

Testing TS Stationarity

The null hypothesis of the test is that the time series can be represented by a unit root, that it is not stationary (has some time-dependent structure). The alternate hypothesis (rejecting the null hypothesis) is that the time series is stationary.

- **Null Hypothesis (H0):** If failed to be rejected, it suggests the time series has a unit root, meaning it is non-stationary. It has some time dependent structure.
- **Alternate Hypothesis (H1):** The null hypothesis is rejected; it suggests the time series does not have a unit root, meaning it is stationary. It does not have time-dependent structure.

We interpret this result using the p-value from the test. A p-value below a threshold (such as 5% or 1%) suggests we reject the null hypothesis (stationary), otherwise a p-value above the threshold suggests we fail to reject the null hypothesis (non-stationary).

- **p-value > 0.05:** Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.
- **p-value <= 0.05:** Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

Pre-Processing

What is Pre-Processing and why is it Necessary?

- **Definition:** Data pre-processing is the initial phase of data preparation that involves transforming raw data into a structured and cleaned format suitable for analysis.

Common Data Pre-Processing Techniques

a. Data Cleaning:

- Handle missing values: Imputation or removal of missing data.
- Remove duplicates: Eliminate identical records from the dataset.
- Deal with outliers: Treat or remove extreme data points.

b. Data Transformation:

- Feature scaling: Standardize or normalize numerical features.
- Encode categorical data: Convert categorical variables into numerical representations.
- Feature engineering: Create new informative features.

Handle Missing Values

Delete Rows/Columns

This method we commonly used to handle missing values. Rows can be deleted if it has insignificant number of missing value Columns can be delete if it has more than 75% of missing value

Replacing with mean/ median/mode

This method can be used on independent variable when it has numerical variables. On categorical feature we apply mode method to fill the missing value.

Algorithm Imputation

Some machine learning algorithm supports to handle missing value in the datasets. Like KNN, Naïve Bayes, Random forest.

Predicting the missing values

Prediction model is one of the advanced method to handle missing values. In this method dataset with no missing value become training set and dataset with missing value become the test set and the missing values is treated as target variable.

Example

For Numerical Data

Airlines	Ticket Price
Indigo	3887
Air Asia	7662
Jet Airways	-
Air India	5221
SpiceJet	4321

Suppose we have Missing values in the categorical data:

Then we take the mode of the dataset a to fill the missing values: Here :

Mode = Indigo

We substitute the Indigo in place of missing value in Airline column

The following are some steps involve in Data Cleaning

Suppose we have Airlines ticket price data in which there is missing value.

Steps to fill the numeric missing value:-

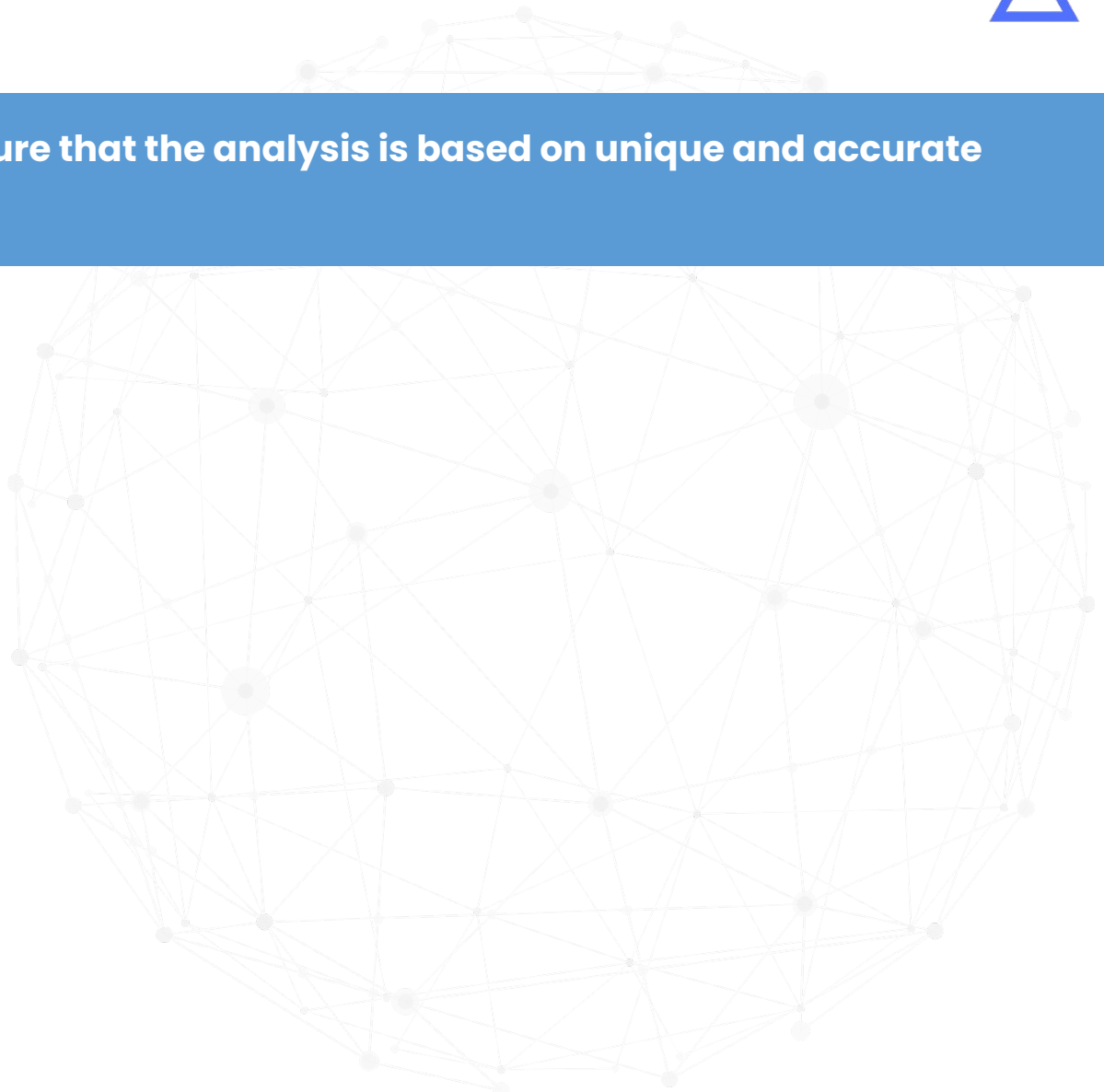
1. Compute the mean/median of the data
   (3887+7662+5221+4321)/4 = **5272.75**
2. Substitute the Mean of the value in missing place.

For categorical Data

Airlines	Ticket Price
Indigo	3887
Indigo	7675
Air Asia	4236
-	6524
Jet Airways	4321

Remove Duplicates

Removing duplicates from Time Series data is important to ensure that the analysis is based on unique and accurate values. In Python, this can be done using the Pandas library



Outlier Treatment

Outliers are the most extremes values in the data. It is an abnormal observations that deviate from the norm. Outliers do not fit in the normal behavior of the data.

Detect Outliers using following methods

1. Boxplot
2. Histogram
3. Scatter plot
4. Z-score
5. Inter quartile range(values out of 1.5 time of IQR)

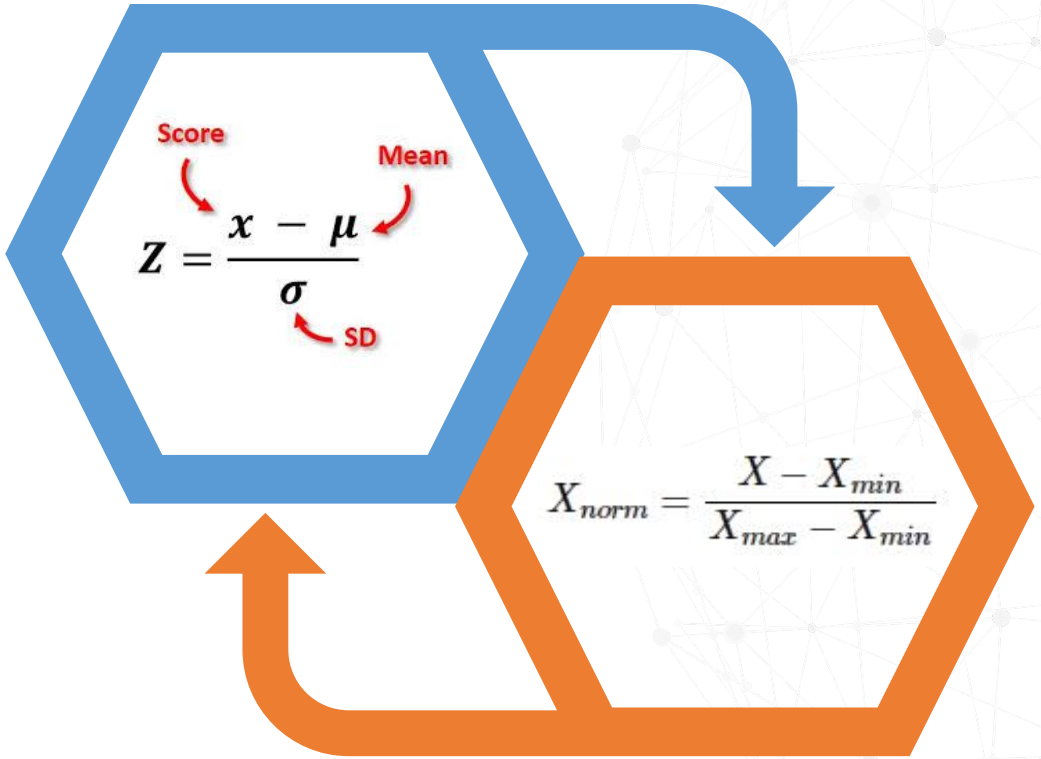
Handle Outlier using following methods

1. Remove the outliers.
2. Replace outlier with suitable values by using following methods:-
   - Quantile method
   - Inter quartile range
3. Use that ML model which are not sensitive to outliers
4. Like:-KNN, Decision Tree, SVM, Naïve Bayes, Ensemble methods

Feature Scaling Method.

Standard Scaler

Standard scaler ensures that for each feature, the mean is zero and the standard deviation is 1, bringing all feature to the same magnitude. In simple words Standardization helps you to scale down your feature based on the standard normal distribution



Min-Max Scaler

Normalization helps you to scale down your features between a range 0 to 1

Example

Normalization

Age	Income (£)	New value
24	15000	(15000 – 12000)/18000 = 0.16667
30	12000	(12000 – 12000)/18000 =0
28	30000	(30000 – 12000)/18000 =1

Income Minimum = 12000
Income Maximum = 30000
(Max – min) = (30000 – 12000) = 18000

Hence, we have converted the income values between 0 and 1

Please note, the new values have
Minimum = 0
Maximum = 1

Example

Standardization

Age	Income (£)	New value
24	15000	(15000 - 19000)/9643.65 = -0.4147
30	12000	(12000 - 19000)/9643.65 = -0.7258
28	30000	(30000-19000)/9643.65 = 1.1406

Average = (15000 + 12000 + 30000)/3 = 19000
Standard deviation = 9643.65

Hence, we have converted the income values to lower values using the z-score method.

x = c(-0.4147, -0.7258, 1.1406)
mean(x) = -0.000003 ~ 0
var(x) = 0.999 ~1

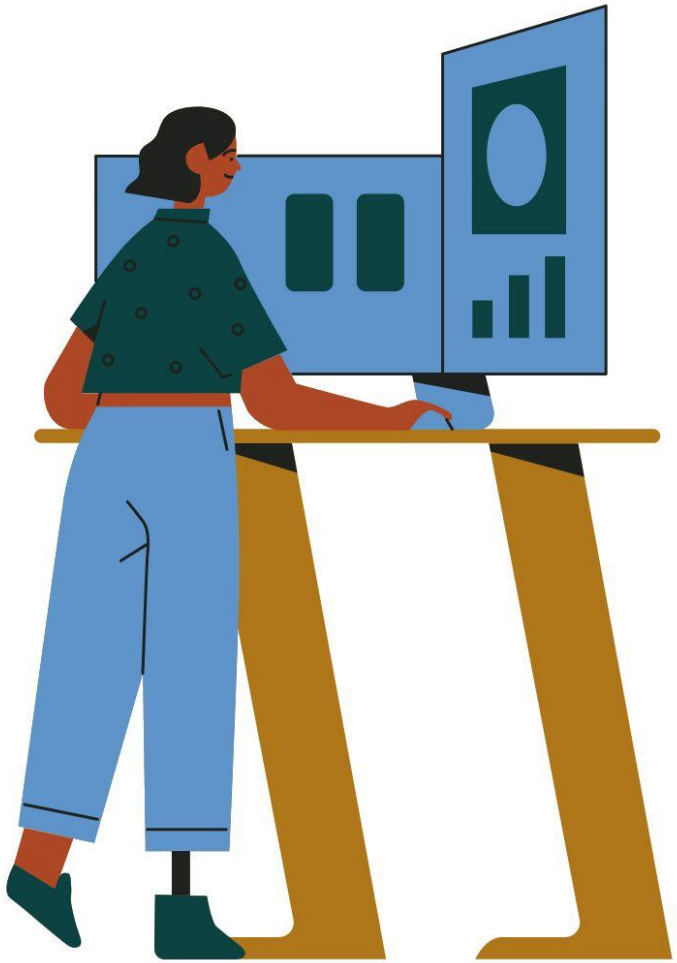
Feature Encoding

Feature encoding help us to transform categorical data into numeric data.

Label encoding	Label encoding is technique to transform categorical variables into numerical variables by assigning a numerical value to each of the categories.
One-Hot encoding	This technique is used when independent variables are nominal. It creates k different columns each for a category and replaces one column with 1 rest of the columns is 0. Here, 0 represents the absence, and 1 represents the presence of that category.

Best Practices

- Always start with data exploration to understand your data thoroughly.
- Handle missing data strategically, considering the nature of the problem and the data.
- Use appropriate scaling methods based on the algorithms you intend to use.
- Keep track of the steps applied to the data and document them properly.
- Validate the pre-processed data with domain experts if possible.



Exploratory Data Analysis

What is EDA?

- **Definition:** Exploratory Data Analysis (EDA) is a preliminary approach to data analysis that involves visually and statistically summarizing the main characteristics of a dataset.
- EDA plays a significant role in data cleaning, feature selection, and hypothesis generation before proceeding with more advanced modeling or analysis techniques.

Why is EDA Important?

- EDA helps us understand the data's distribution, relationships, and potential outliers.
- Identify and address missing values, duplicates, and anomalies during EDA.
- EDA aids in selecting relevant features for modeling.
- EDA can suggest potential hypotheses for further investigation.

Common EDA Techniques

Univariate Analysis: Examining individual variables to understand their distributions and characteristics using histograms, box plots, etc.

Bivariate Analysis: Exploring relationships between two variables with scatter plots, correlation matrices, etc.

Multivariate Analysis: Analyzing multiple variables simultaneously to uncover complex patterns and interactions.

EDA Workflow

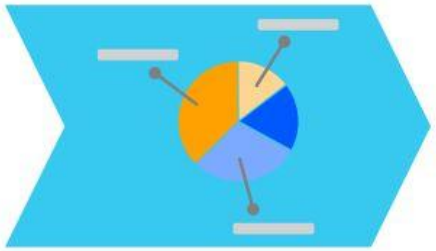
1 - Data Collection

Gather the dataset from reliable sources.

2 - Data Cleaning

Handle missing values, duplicates, and outliers.

3 - Data Visualization



Plot charts, histograms, scatter plots, etc., to explore the data visually.

4 - Data Summarization

Calculate statistical measures like mean, median, variance, etc.

5 - Insight Generation

Derive insights from the patterns observed during the analysis.

Time Series Forecasting Models

Algorithms

ARIMA (AR, MA, ARMA, ARIMA)
Facebook Prophet
LSTMs
Holt's Winter Exponential Smoothing
GARCH
SARIMA/SARIMAX
VAR
VARMA etc. etc.

Quick Link:
https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/

ARIMA

Auto regressive

Moving Average

Autoregressive Moving Average (No differencing)

Autoregressive Integrated Moving Average

ARIMA

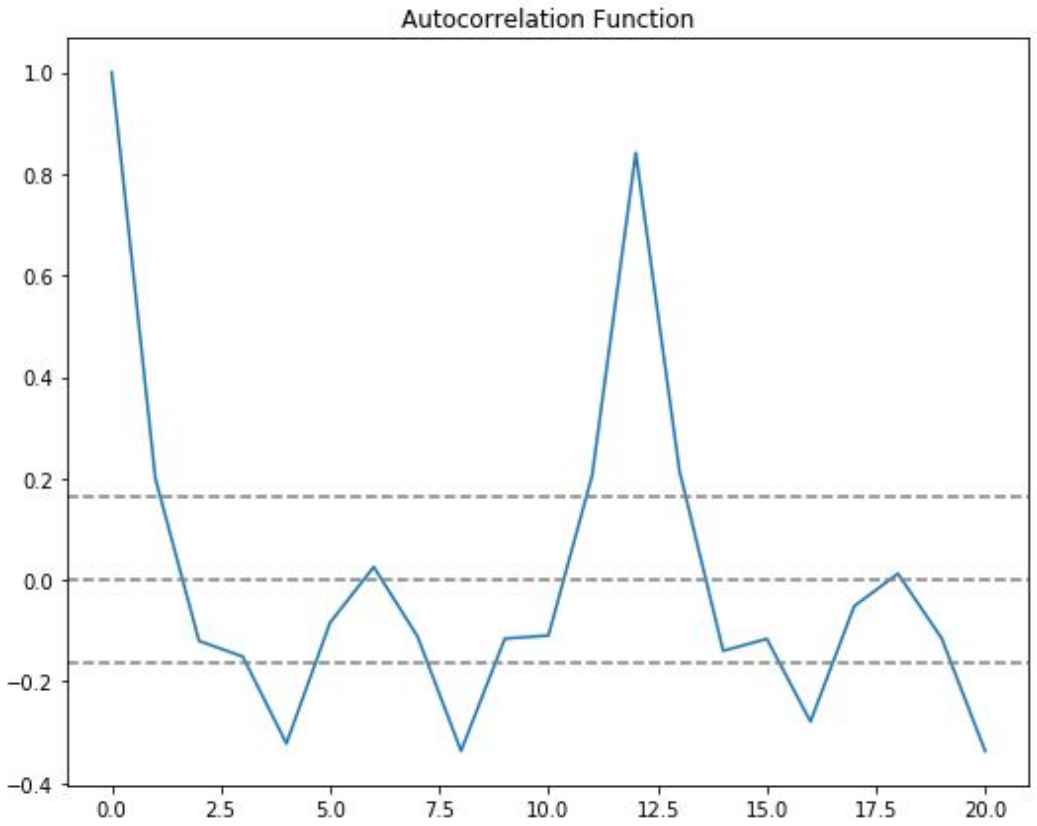
There are different techniques to find the right parameters for ARIMA(p,d,q)

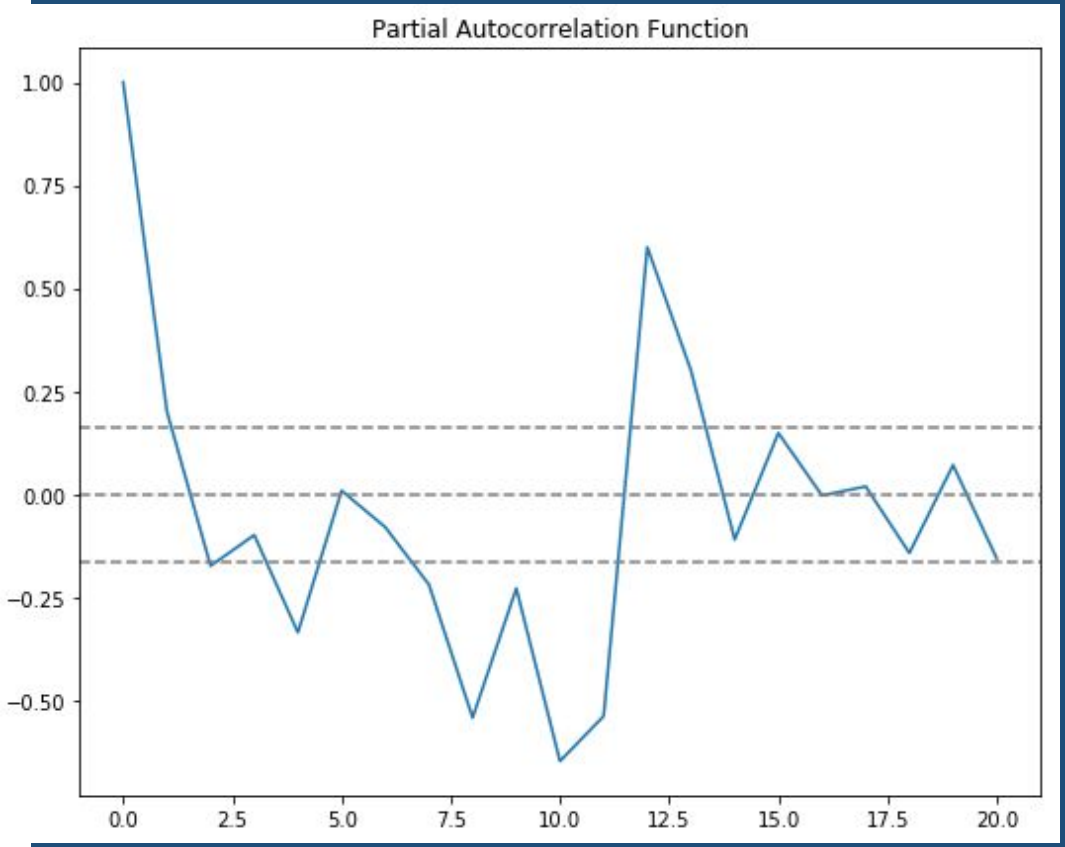
- ACF/PACF Plots
- Grid Search
- Auto Arima

Let's learn about these techniques in the next slides.

Auto Correlation

Partial Auto Correlation





1. p – The lag value where the PACF chart crosses the upper confidence interval for the first time. If you notice closely, in this case p=2.
2. q – The lag value where the ACF chart crosses the upper confidence interval for the first time. If you notice closely, in this case q=2.

Grid Search

ACF/PACF plots are some traditional methods of obtaining p & q values, and are sometimes misleading, hence we need to perform a hyper parameter optimization step in Time Series Analysis to get the optimum p,d & q values

Auto Arima

Grid Search techniques are manual ways, the same task can be achieved in few lines of coding and with a better efficiency using Auto Arima

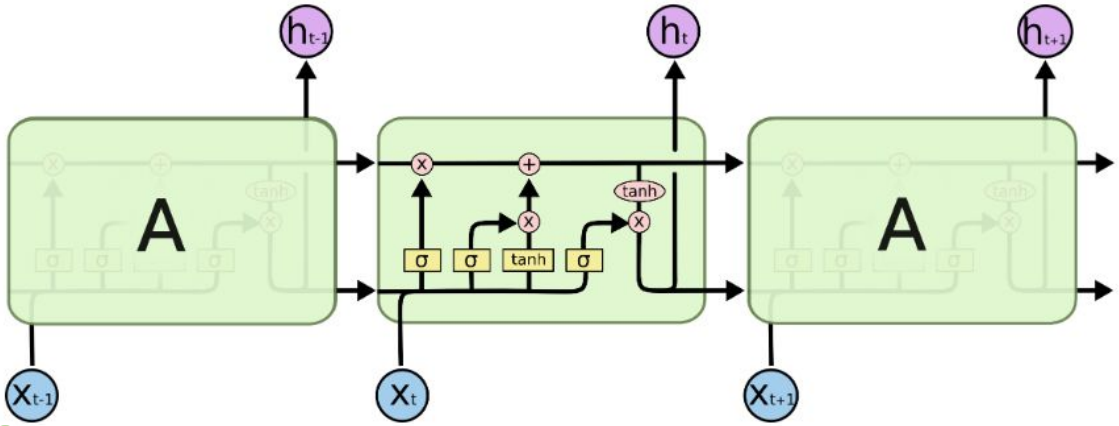
Facebook Prophet

Quick Link:
https://facebook.github.io/prophet/docs/quick_start.html#python-api

Features:

- Very fast
- An additive regression model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects
- Robust to missing data & shifts in trend, and handles outliers automatically.
- Easy procedure to tweak & adjust forecast while adding domain knowledge or business insights.

LSTMs



Quick Link:
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Features:

LSTM cell in place of standard neural network layers:

1. Input gate
2. Forget gate
3. Output gate

Multi Variate TS Analysis

Univariate vs Multivariate

Univariate

Series with a single time-dependent variable

Key Characteristics:

- **Single Variable**: Univariate analysis deals with a solitary time series, examining its historical values and detecting trends, seasonality, and anomalies.
- **Simplicity**: Univariate analysis is relatively simple to perform as it involves analyzing one-dimensional data.
- **Forecasting**: In univariate analysis, the primary goal is often to forecast future values of the same variable based on its past observations.

Multivariate

Series having more than one time series variable

Key Characteristics:

- **Multiple Variables**: Multivariate analysis considers the interactions and relationships between multiple time series variables simultaneously.
- **Complexity**: Analyzing multiple variables adds complexity as it involves understanding the interdependencies between these variables.
- **Causality**: Multivariate analysis can help identify causal relationships between different variables, if they exist.

Importance of Multivariate Analysis in Time Series

Capturing

Multivariate analysis allows us to understand how multiple variables influence each other over time, enabling a comprehensive view of the system under study.

Improved

By considering relationships between variables, we can enhance the accuracy of time series forecasts.

Identifying

Multivariate analysis uncovers hidden patterns and trends that may not be evident when considering individual variables in isolation.

Challenges in Multivariate Time Series Analysis

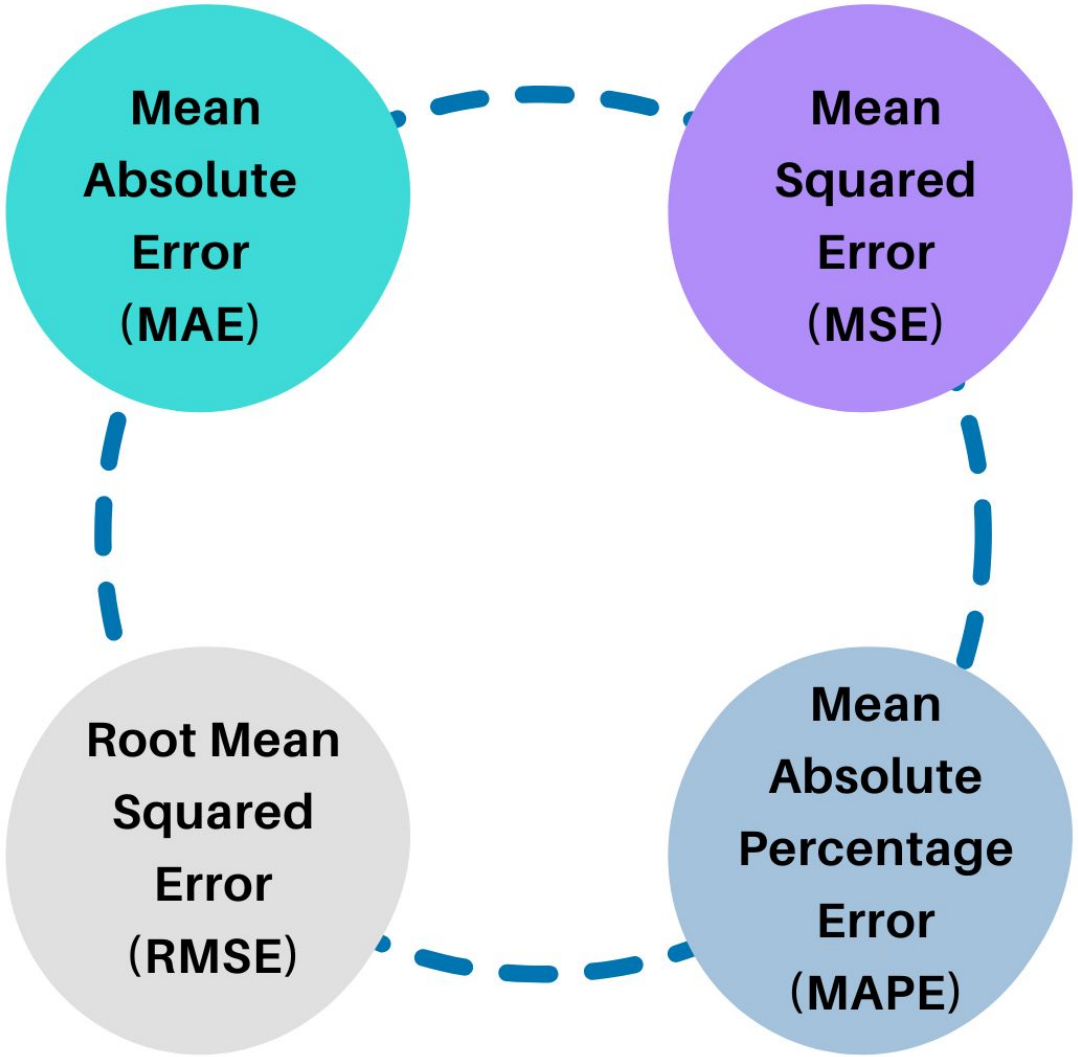
- **High Dimensionality:** Analyzing multiple variables requires handling high-dimensional data.
- **Model Complexity:** Multivariate analysis may lead to more complex models that require careful interpretation.
- **Data Pre-processing:** Data cleaning and preparation become crucial to handle multiple variables effectively.

Evaluating Forecasting Performance

Importance of Forecasting Evaluation

- **Reliable Decisions:** Accurate forecasts aid in making well-informed decisions, reducing risks, and optimizing resources.
- **Model Selection:** Evaluating multiple forecasting models helps identify the most suitable and accurate approach for a given dataset.
- **Continuous Improvement:** Regular evaluation enables model refinement and improvement over time.

Forecasting Evaluation Metrics



Mean Absolute Percentage Error

Advanced Time Series Forecasting Techniques

Seasonal Decomposition of Time Series (STL)

- STL is a time series decomposition method that separates a time series into three components: Seasonal, Trend, and Residual.

- It is a non-parametric approach that handles time series with irregular and complex seasonality.

Advantages of STL

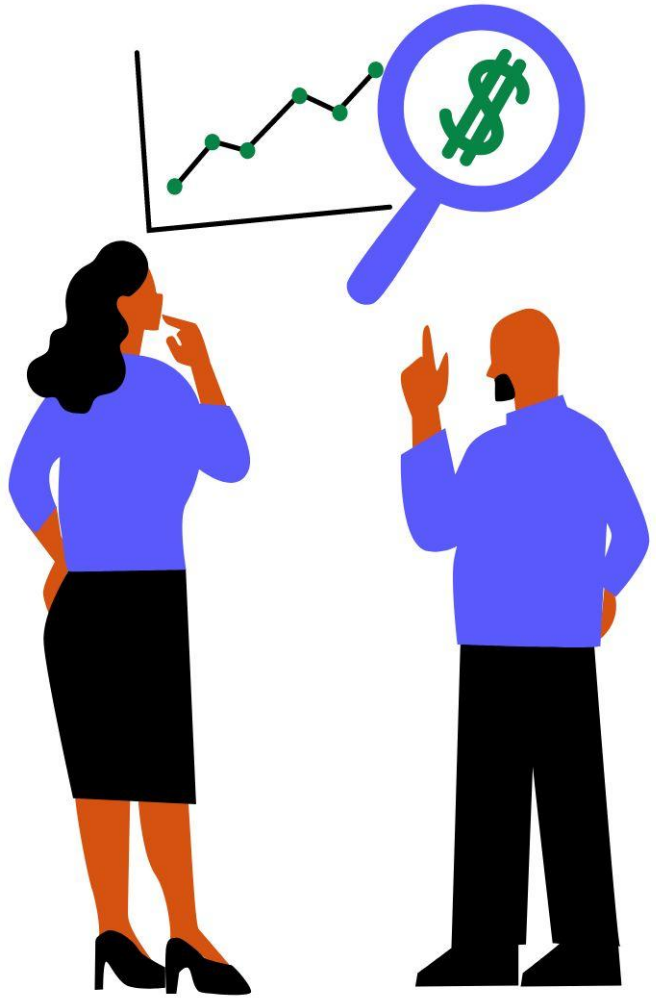
- **Robustness:** STL can handle outliers and irregular seasonality well.

- **Flexibility**: It adapts to different time series patterns without making strict assumptions.

- **Interpretability:** STL provides clear and interpretable components, aiding better understanding of time series behavior.

Gaussian Processes

- Non-parametric Bayesian approach for time series forecasting.
- Provides a probabilistic framework, useful for capturing uncertainties in predictions.
- Suitable for small to medium-sized datasets.



DeepAR

- DeepAR is a probabilistic forecasting algorithm based on Recurrent Neural Networks (RNNs).
- It was introduced by Amazon for handling complex time series data with uncertainty.
- DeepAR can generate point forecasts and prediction intervals, providing a measure of uncertainty for each forecast.

Ensemble Methods

- Combining multiple forecasting models to get more accurate and robust predictions.

- Techniques like weighted averaging, stacking, and boosting can enhance forecasting performance.

Case Studies

Energy Demand Forecasting

- **Objective:** A utility company wanted to predict the electricity demand accurately to optimize power generation and distribution.

- **Approach:** Time series data from past years, including historical demand, weather data, and holidays, was collected. SARIMA and Prophet models were used to capture seasonality and trends. DeepAR was applied for probabilistic forecasting with prediction intervals.

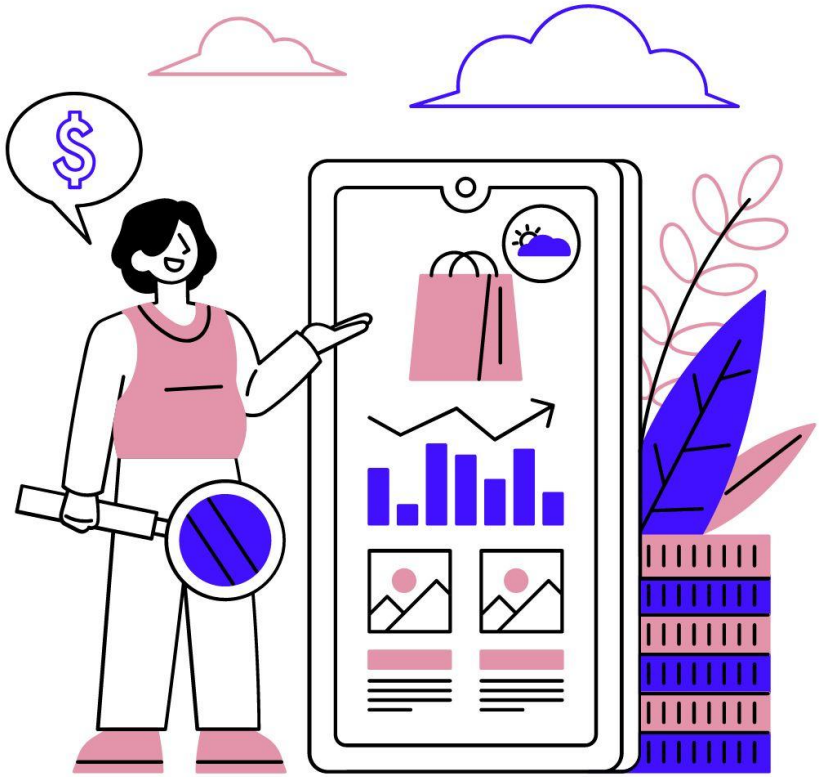
- **Outcome:** The accurate forecasts enabled the utility company to optimize power generation, reduce operational costs, and prevent energy shortages during peak demand periods.

Stock Market Prediction

- **Objective:** A financial institution aimed to predict stock prices for various companies to optimize trading strategies.
- **Approach:** Historical stock price data, trading volumes, and financial indicators were used as input. FB Prophet was applied in order to forecast the stock market values
- **Outcome:** The accurate stock price predictions helped the financial institution make better investment decisions and improve trading strategies.

Demand Forecasting for E-commerce

- **Objective:** An e-commerce platform aimed to predict customer demand to optimize pricing and inventory management.
- **Approach:** Time series data of customer browsing behavior, product views, and historical sales were collected. Multiple models were used to capture seasonality and long-term dependencies.
- **Outcome:** Accurate demand forecasts helped the e-commerce platform optimize pricing, improve product recommendations, and efficiently manage inventory, resulting in increased sales and customer retention.



Thank you