DEPARTMENT OF BIOMEDICAL ENGINEERING

24AI671-FUNDAMENTALS OF DATA SCIENCE

UNIT V- DATA SCIENCE FOR REAL-WORLD ENGINEERING SOLUTIONS

5.2. PRESCRIPTIVE ANALYTICS

Prescriptive analytics is the most advanced form of analytics. It takes the output of predictive models and combines it with **optimisation, simulation and business rules** to recommend the **best action** to take. It answers the question **"What should we do, and how can we make it happen?"**

Predictive analytics may tell us that demand will rise by 20% next month. Prescriptive analytics goes further and tells us **how much to produce, when to buy raw material and which machines to schedule** so that profit is maximum and cost is minimum. It also shows the likely result of each possible decision, so managers can compare options before choosing.

Main Elements of a Prescriptive Model

- **Objective function** – the goal to be maximised or minimised, such as profit, cost, time or energy use.
- **Decision variables** – the quantities that can be controlled, such as production quantity, route or schedule.
- **Constraints** – the limits within which the decision must be made, such as budget, machine capacity, time, raw material and safety rules.
- **Solver / algorithm** – the method that searches all possible choices and finds the best feasible solution.
- **Feedback loop** – the results of the decision are monitored and used to improve the next decision.

Maximise (or Minimise) $Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$ subject to $Ax \leq b$, $x \geq 0$

In this standard linear programming form, **Z** is the objective, **x** are the decision variables, **c** are their profit or cost values, and **Ax ≤ b** represents the resource constraints.

Techniques Used

Technique	Description	Example
Linear / Integer programming	Finds the best value of a linear objective when constraints are given; integer programming allows only whole-number answers	Production planning, resource allocation
Simulation (Monte Carlo)	Runs the system many times with random inputs to test 'what-if' situations and measure risk	Project risk analysis, supply-chain testing
Genetic algorithm, PSO	Nature-inspired search methods that find good solutions to very hard problems	Vehicle routing, job scheduling
Decision analysis	Decision trees and expected value are used to compare choices under uncertainty	Supplier or site selection
Reinforcement learning	A software agent learns the best actions by trial, reward and penalty	Robot control, energy management
Rule-based / expert systems	If–then rules created by domain experts	Alarm and fault-handling response

Predictive vs Prescriptive Analytics

Point	Predictive	Prescriptive
Main question	What will happen?	What should be done?
Output	Forecast or probability	Recommended action or decision
Methods	Regression, classification, time series	Optimisation, simulation, reinforcement learning
Complexity	Moderate	High – needs predictions plus optimisation
Example	Machine will fail in 7 days	Repair on day 5 at the lowest cost and downtime

Applications

- **Manufacturing** – best job-shop schedule and inventory level that keep machines busy at minimum cost.
- **Smart grid** – deciding which power plants to run and how much power to supply (economic dispatch), and when to ask consumers to reduce load (demand response).
- **Logistics** – finding the shortest and cheapest delivery routes for many vehicles.
- **Maintenance** – choosing the best time to repair a machine so that downtime and repair cost are lowest.

- **Construction** – allocating workers, machines and materials so that the project finishes on time within budget.
- **Healthcare** – scheduling doctors, beds and operation theatres.

Advantages and Limitations

Advantages	Limitations
Gives clear, optimal and fast decisions	Depends on accurate predictions and data
Saves cost and time	Complex models and high computing power
Handles many constraints together	Users may not trust or understand the recommendation
Allows 'what-if' analysis before action	Needs expert knowledge to model the problem correctly

OBSERVE OPTIMIZE OUTSPREAD

5.3. TIME SERIES ANALYSIS

A **time series** is a set of observations recorded one after another at **regular time intervals** – every second, hour, day, month or year. Examples are hourly electricity load, daily rainfall, monthly sales and the vibration signal of a machine. The order of the data is important because the present value usually depends on past values.

Time series analysis studies this data to understand its pattern (trend, seasonality, etc.) and to **forecast future values**. It is one of the most widely used methods in predictive analytics.

Components of a Time Series

A time series can be broken into four parts. Separating them helps us understand the data and select a suitable model.

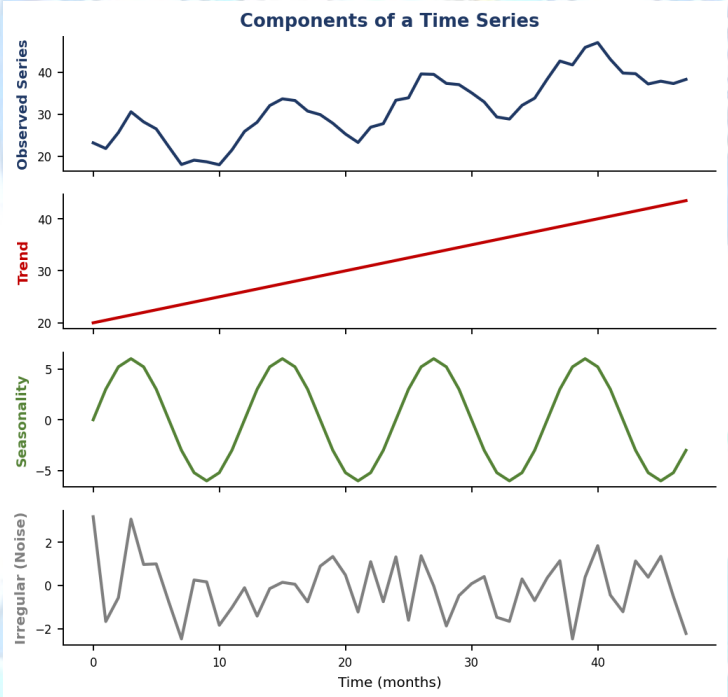


Fig 3.1 Observed series split into trend, seasonality and noise

Component	Meaning	Example
Trend (T)	Long-term upward or downward movement of the data	Electricity demand rising over the years
Seasonality (S)	Regular pattern that repeats at a fixed interval	Higher AC load every summer; festival-time sales
Cyclic (C)	Ups and downs that repeat but with no fixed period (usually longer than a year)	Economic or business cycles
Irregular (I)	Random, unpredictable changes that cannot be explained	Sudden power failure, sensor error

The components are combined in two ways. If the seasonal swings stay the same size, the **additive model** is used. If they grow along with the trend, the **multiplicative model** is used.

Additive model: $Y = T + S + C + I$

Multiplicative model: $Y = T \times S \times C \times I$

Stationarity

A time series is **stationary** when its **mean, variance and autocorrelation do not change with time**. It has no trend and no seasonality. Most forecasting models such as AR, MA and ARIMA work correctly only on stationary data.

- A non-stationary series is made stationary by **differencing** (subtracting the previous value: $Y_t - Y_{t-1}$), log transformation or removing trend and seasonality.
- Stationarity is checked by plotting the data, looking at rolling mean and variance, and by the **ADF (Augmented Dickey–Fuller) test**. If the p-value is less than 0.05 the series is stationary.

Steps in Time Series Forecasting

- **Plot the data** and study the trend, seasonality and unusual points.
- **Check stationarity** and apply differencing or transformation if needed.
- **Select the model order** using the **ACF** (autocorrelation function – correlation of the series with its own past values) and **PACF** (partial autocorrelation – direct correlation after removing the effect of in-between lags). PACF helps choose p and ACF helps choose q.
- **Fit the model** and estimate its parameters.
- **Check the residuals** – the leftover errors should look like random noise with no pattern. If not, the model is improved.
- **Forecast** future values and measure the error using MAE, RMSE or MAPE.

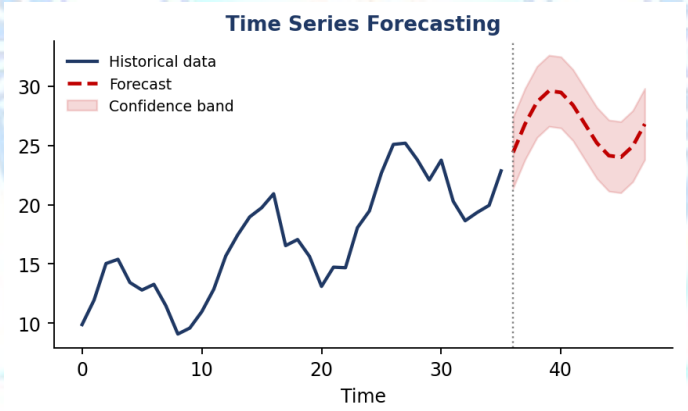


Fig 3.2 Forecast with confidence band

Time Series Models

Smoothing Methods

- **Moving average** – the average of the last k values is used as the forecast. It removes random noise but reacts slowly to change.
- **Simple exponential smoothing** – recent values are given more weight than old values: $F_{t+1} = \alpha Y_t + (1 - \alpha)F_t$, where α (0 to 1) is the smoothing constant. **Holt's method** adds trend and **Holt–Winters** adds trend and seasonality.

AR, MA and ARIMA Models

- **AR(p) – Autoregressive model:** the present value is a weighted sum of the p previous values plus an error.

$$Y_t = c + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \ldots + \varphi_p Y_{t-p} + \varepsilon_t$$

- **MA(q) – Moving average model:** the present value depends on the q previous forecast errors.

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta q \varepsilon_{t-q}$$

- **ARIMA(p, d, q):** combines AR, differencing and MA. **p** = order of the AR part, **d** = number of times the data is differenced, **q** = order of the MA part. **SARIMA** adds seasonal terms and is used for seasonal data.

Machine Learning Models

- **LSTM / GRU:** special neural networks that remember long-term patterns and handle non-linear data.
- **Prophet:** a model that handles trend, seasonality and holidays and is easy to use.

Forecast Error Measures

- **MAE** – average of absolute errors; **RMSE** – square root of average squared error (punishes large errors); **MAPE** – average percentage error, easy to understand.

$$\text{MAPE} = (100 / n) \Sigma \mid (\text{Actual} - \text{Forecast}) / \text{Actual} \mid$$

Applications

- Short-term and long-term electricity load forecasting, and solar and wind power prediction.
- Monitoring vibration, temperature and current of machines to detect faults early.
- Rainfall, river-flow and water-demand prediction in civil and environmental engineering.
- Network traffic prediction and anomaly detection in IoT sensor data.
- Sales, stock and inventory forecasting in business.