|  |
| --- |
| IU |
| Unit 1:Newsvendor Model |
| DLMAIEECMDF02  Mojtaba Nabipour |

# Newsvendor model

**Study Goals**

On completion of this unit, you will be able to …

…define optimal inventory stocking decision.

…explain the classical newsvendor model.

…understand various demand models.

…identify important distributions.

…understand some important extensions of the classical newsvendor model.

## Introduction

The newsvendor problem, a well-known inventory management dilemma, entails making decisions to determine the most advantageous quantity to order for a time-sensitive item with uncertain demand. The problem is named after the analogy of a newsvendor who must decide how many newspapers to order for sale the next day. The newsvendor model is used to find the optimal level of stocking in order to minimize the total cost in the presence of demand uncertainty. The newsvendor problem is common in real life, including following examples: Perishable goods such as blood products that spoil quickly. Certain consumables or goods that are subject to fashion and become obsolete after a certain period of time. The term can also be applied to products that rely on technology, such as cell phones, as well as similar examples. Considering the uncertainty of the demand level, this problem is not a trivial problem. In the following, we begin with a simple form of this model, while using demand modeling methods to address demand uncertainty. By the end, we will explore the complex model of multi-period newsvendors.

## 1.1 Single Period Newsvendor Model

These models operate under the constraint that the inventory of goods can only be accessed for a limited duration, and it is impossible to retain stock between consecutive periods. These systems are inherently linked to one-time decision models, which pertain to decision-making processes involving a singular choice that is not anticipated to be replicated in subsequent instances. The goal of a one-time decision model is to make the best possible decision given the available information and resources at the time. Here, the objective is to determine the order quantity that maximizes a product's profit when the demand is probabilistic (Chopra et al., 2007). If the order level is lower than the demand level, some profit will be lost (understocking situation). Meanwhile, if the anticipated demand is not met, overstocking costs will reduce profits. Thus, the problem is how to choose the order level to maximize the profit or minimizing the cost. To formulate the problem, we need some notations:

|  |  |  |
| --- | --- | --- |
| Single-Period Model Notations | | |
| Notation | **Explanation** | **Simply speaking** |
| s | Unit price | The price at which a unit is sold |
| w | Wholesale cost per unit | Purchasing + transporting+… |
| Cu | Understocking cost per unit | If one unit is short the cost (losing the profit) will be Cu |
| Co | Overstocking cost per unit | If one unit is unsold the cost will be Co. usually it contains the original cost added by the inventory cost |
| f ( ) | Probability density function of demand | Likelihood of a certain demand |
| F ( ) | Cumulative density function of demand | Probability of realization of less than a certain demand level |
| Q | Order level | Number of stocked units with Q\* as optimum |
| x | Demand level |  |

Source: [Mojtaba Nabipour], (2023).

Given the order level Q, the total cost is calculated as:

Cu(x-Q) x>Q (Eq.1)

C(Q) =

Co(Q-x) x<Q

Since x is a random variable, the expected value of cost must be calculated as:

(Eq.2)

It can be proved that the e(Q) is concave (Leibniz’s rule). Therefore, we can use the optimality condition to achieve the minimum(Reid & Sanders, 2023):

(Eq.3)

In literature this fraction is known as the critical ratio and represents the relative importance of overstocking against understocking cost. To achieve the Q\*, one can take the inverse of the cumulative function:

(Eq.4)

 . (Eq.5)

The critical ratio indicates the level of uncertainty in stocking decision. For example, stocking of very perishable goods is associated with a very large uncertainty, forcing to reduce the Q and vice versa, see Figure.

Probability Distribution Of Demand For Two Perishable Goods



Newspaper

Fashion goods

Order level

Order level

Source: [Mojtaba Nabipour], (2023).

**Example 1**: according to given tables

1. Calculate the expected profit if the order quantity is 3
2. Calculate the expected profit if the order quantity is 4
3. Given that the probability distribution of the demands is normal with mean of 7 and standard deviation of 1, calculate the optimal order quantity or Q\*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Demand quantity** | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| **probability** | 0.04 | 0.06 | 0.13 | 0.17 | 0.19 | 0.16 | 0.13 | 0.07 | 0.05 |

|  |  |  |  |
| --- | --- | --- | --- |
| **w** | 120$ | **Cu** | 150-120=30$ |
| **s** | 150$ | **Co** | 120-100=20$ |

**Answer**:

1. The profit can be calculated by subtracting the cost from the raw revenue but in the case of probabilistic demand we calculate the expected profit:

Since *F(3) = 1* we have

*E[Profit] = 3×150 -3×120 = 90*

1. If *demand = 3* then *revenue = 3×150* with the probability of *0.04*

If *demand ≥ 4* then *revenue = 4×150* with the probability of *0.96 i.e. F(4)=0.96*; So the expected profit will be

*E[Profit] = 3×150×0.04 + 4×150×0.96 – 4×120 = 114*

1. F(Q\*) = *30 / (30+20) = 0.6*

NORM.INV *(0.6, 7, 1) = 7.25*

It is possible to use the NORM.INV (x, mean, std) from excel.

### Self-check Questions

1. What is the objective of the classical newsvendor problem?

*The objective of the classical newsvendor problem is to determine the optimal order quantity that maximizes profit or minimizes cost by balancing the trade-off between overstocking and understocking inventory levels.*

2. How does the classical newsvendor problem differ from the standard inventory management problem?

*The classical newsvendor problem differs from the standard inventory management problem in that demand for the product is uncertain and cannot be forecasted accurately. Additionally, the cost of overstocking and understocking may not be equal.*

## 1.2 Demand as a Stochastic Quantity

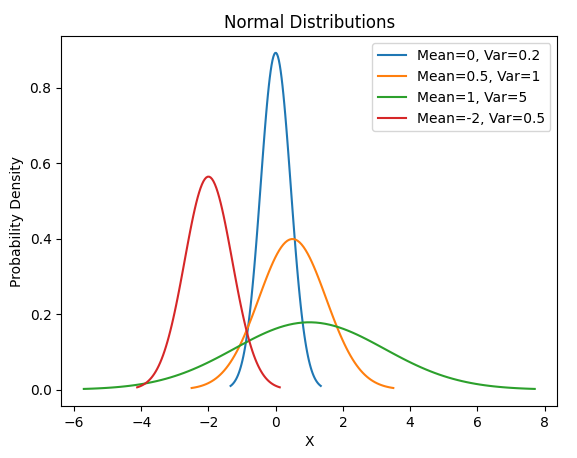
So far, we introduced the demand quantity as an uncertain variable, and implicitly used statistics to handle it in our cost function. In this section we assume the demand as a stochastic quantity that follows a statistical distribution. Firstly, we will briefly review probability distributions and the relevant statistical concepts, then these concepts will be used to describe the demand variable.

In light of deterministic demand in inventory management, it refers to a situation where the demand for a product or item is known with certainty. In other words, the quantity of items needed or sold is predetermined and does not vary or fluctuate. As it is obvious if the received demands are deterministic, the inventory management would be a straightforward job. However, in reality the demands are random variables and have uncertain nature. Practically the probability distribution helps us to handle the associated uncertainty. Here we first concentrate on continuous distributions then we discuss discrete ones. The normal distribution is very popular to describe the events when the probability of the occurrence is higher around a specific mean value and with an almost constant average dispersion. Thus, this distribution is symmetric around the mean and can be defined by 2 parameters: Mean (µ) and standard deviation (σ)

. (Eq.6)

Figure demonstrates 4 different normal distribution diagrams. Normal distribution appears as a bell-shaped form. Notice the impact of parameters on the shape of the distribution.

Normal Distribution



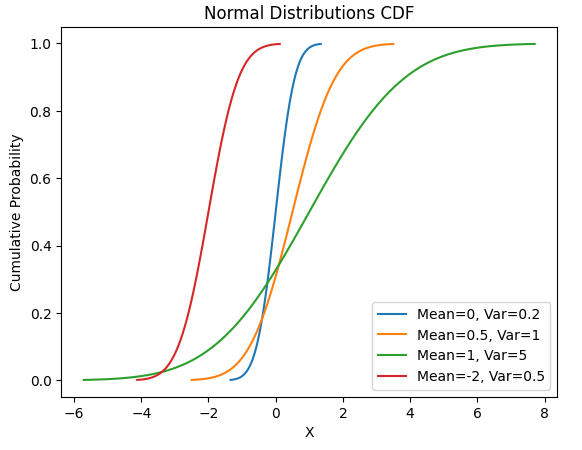
Source: [Mojtaba Nabipour], (2023).

The Central Limit Theorem (CLT) is based on the normal distribution model. As a result of this theory, averages derived from independent, identically distributed random variables have approximately normal distributions, no matter what method is being used to sample them (as long as the distribution has a finite variance).(Ross, 2017). A cumulative distribution function (CDF) is an estimate of the probability that a random distribution will occur less than or equal to a certain value. Thus, the CDF is the integration of all probability values of the random variable occurred between 0 and the threshold value Z. This means:

(Eq.7)

The CDF of the normal distribution is shown in Figure.

CDF of Normal Distribution



Source: [Mojtaba Nabipour], (2023).

**Example 2:**

There is a normal distribution of weekly product demand with a mean of 1500 and a standard deviation of 300. In the next two weeks, there will be no deliveries and the current on-hand inventory is 3200. Considering the independent nature of the demands in different weeks:

a) In each of the next two weeks, what is the probability that demand will be less than *1600*?

b) What is the probability that the total demand in the next two weeks exceeds *3200*?

**Answer:**

a) let’s define x1 as the demands of 1st next week and x2 as the demands of the 2nd next week

according to the assumptions the answer will be:

*a is the demanded probability in section a* and *p(a)* is it’s probability

Since x1 and x2 are independent we have:

**P(a) = 0.3976**

b) Any linear combination of normal variables has normal distribution with the same linear combination of means and variances.

If we define *x = x1+x2* then:

i.e. we have

So, the requested probability will be:

b is the demanded probability in section b.

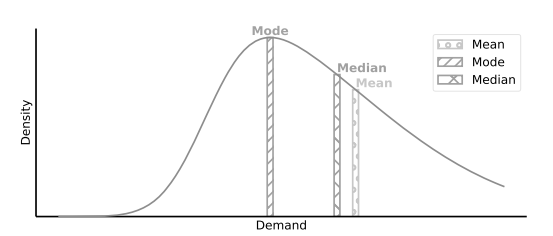
**P(b) = 0.068.**

Current on-hand inventory is often referred to as safety stock. It helps the vendor to cover demands during supply shortages.

So far we assumed that the demands follow a normal distribution. As the name suggests, in normal situations the demands for a certain stock often fluctuate around an average with a probability that is high enough, so it makes sense to use the normal distribution to describe the stochasticity of such demands.

Sometimes small variations in situations like exceptional weather, product improvement, competitors understocking, etc. generate unexpected demands for usual low demand items. For example, the demand for toilet paper experienced a dramatic increase during the Covid pandemic. So, using the normal distribution with a symmetrical expansion over a mean value may not be desirable. Normal distributions have the same mode (the value that occurs the most often), median (the value in the middle of a dataset), and mean, and are symmetrically distributed around them. However, there are often higher exceptions (values other than mode value) in real supply chains than the normal distribution predicts. As the figure below illustrates, this means the real distribution is right skewed and the demand fluctuates around the mode value and not the mean (Vandeput, 2020). Further in normal distribution the negative part of the horizontal real axis which indicates the demand values does not cope with the reality and is only applicable in the case of high values of mean that makes the negative part negligible. As a rule of thumb the mean value must be five times of standard deviation away from the origin (Snyder, 1984). Considering these difficulties, based on some studies like the one carried out by Burgin and Wild (Burgin & Wild, 1967), the demand characteristics can be better described by the gamma probability distribution.

**Demand Distribution in Real Supply Chain**



Source: [Mojtaba Nabipour], (2023).

The gamma distribution can be defined by two parameters α and β. The first parameter is called **shape** and the second is called **scale**.

These parameters can be derived from the mean and standard deviation of data as (Vandeput, 2020):

(Eq.8)

(Eq.9)

The general form of the density function of the gamma distribution is:

(Eq.10)

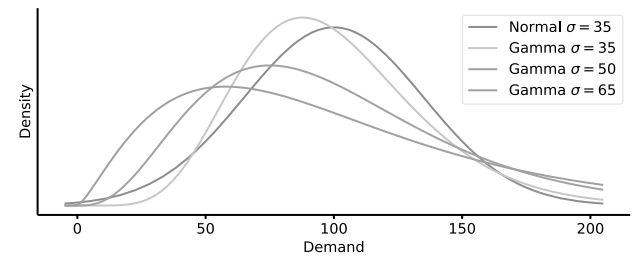
As Figure 5 demonstrates, by decreasing the standard deviation, the gamma distribution gets closer to the normal distribution and the skew is diminished. A further characteristic of the gamma distribution is that it is strictly positive. Figure shows that its PDF around 0 is usually very low.

As we said before the skewness of the distribution is directly related to the standard deviation. The skewness of the distribution is defined as below:

(Eq.11)

The skewness gives us a reliable and straightforward tool to assess whether a gamma or normal distribution is more suited to a particular set of demand data.

**Gamma Distribution With Different Standard Deviation**



Source: [Mojtaba Nabipour], (2023).

The skewness of an observed set of demand is calculated according to (Excel function SKEW (range of data) can compute the value):

. (Eq.12)

So if the skewness is close to 0 the distribution would be normal and if it is close to 2σ/μ the distribution would be gamma. Actually there is a rule of thumb that we can use to choose the right distribution (Vandeput, 2020).

**Example 3.1**

The 94 days’ demand history has been gathered in the following table. Which distribution is a best fit to it.

Table Title

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2.15259 | 3.78 | 1.84023 | 2.32288 | 2.70754 | 3.27141 | 3.80136 | 4.38438 | 5.40975 | 5.9123 |
| 0.82956 | 1.25938 | 1.99574 | 2.4638 | 2.81085 | 3.26863 | 3.8726 | 4.41528 | 5.72096 | 5.532 |
| 0.89594 | 3.8 | 2.0645 | 2.45825 | 2.89762 | 3.33974 | 3.95324 | 4.69969 | 5.82157 | 5.6732 |
| 1.92093 | 1.35397 | 2.01937 | 2.46988 | 2.86425 | 3.35425 | 4.0478 | 4.77041 | 6.17035 | 1.2346 |
| 0.98322 | 4.01 | 2.12213 | 2.5635 | 2.93938 | 3.43649 | 4.23609 | 4.79774 | 6.26586 |  |
| 1.08231 | 1.36602 | 2.13441 | 2.58187 | 2.95057 | 3.5293 | 4.2524 | 4.80418 | 6.35507 |  |
| 1.07008 | 1.45362 | 2.19559 | 2.5467 | 3.10426 | 3.54447 | 4.30449 | 4.99675 | 6.75722 |  |
| 1.15259 | 1.63576 | 2.20841 | 2.6403 | 3.10062 | 3.54565 | 4.31569 | 5.03056 | 7.0478 |  |
| 1.22441 | 3.48 | 2.28921 | 2.6036 | 3.13624 | 3.60547 | 4.32002 | 5.11493 | 8.05613 |  |
| 3.45 | 1.75526 | 2.30481 | 2.66168 | 3.25755 | 3.82151 | 4.37879 | 5.35323 | 8.09159 |  |

**(Table Source)**

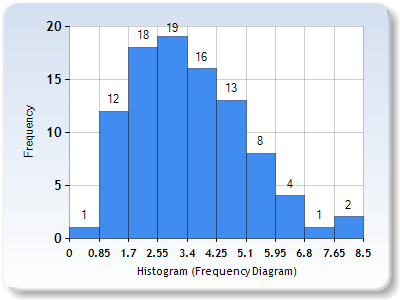
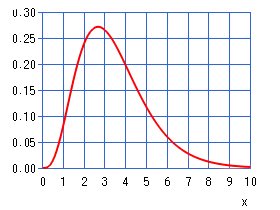
**Solution 3.1:**

Sd = 0.6468 , µd = 3.4417 , σd = 1.6277

)

The histogram of the table (blue bins) and the fitted gamma distribution (red line) is depicted in Figure.

**The Fitted Gamma Distribution on the Demand Data**



Source: [Mojtaba Nabipour], (2023).

In the above example we use the histogram method to identify the distribution. This method is simple, but has some limitations such as limited accuracy in capturing fine-grained variations or irregularities in the data. Indeed, the prominent one is histograms only provide a visual representation of the data and do not provide additional information about the underlying distribution. Students interested in detailed explanations can turn to Vandeput (2020).

**Discrete probability distributions**

A type of probability distribution that models the probability of specific outcomes in a discrete set of values.

**Discrete probability distributions** are used to model situations where there are a finite number of possible outcomes, each with a specific probability of occurring.

In demand modeling, discrete probability distributions are commonly used to model the uncertainty in customer demand for a product. The probability distribution is used to estimate the likelihood of different levels of demand, which can be used to make decisions about inventory levels, pricing, and other aspects of supply chain management.

There are several types of discrete probability distributions that are commonly used in demand modeling, each with their own characteristics and applications. Some of the most commonly used distributions are introduced in the following paragraphs.

#### Poisson Distribution

The Poisson distribution is commonly used to model demand for products or services that occur randomly over time, such as customer arrivals or machine breakdowns. The Poisson distribution is characterized by a single parameter, lambda, which represents the average number of occurrences per unit of time. The Poisson distribution is often used in inventory management and service operations, where demand is unpredictable and sporadic.

#### Binomial Distribution

The binomial distribution is commonly used to model demand for products or services that have a fixed number of possible outcomes, such as the success or failure of a marketing campaign. The binomial distribution is characterized by two parameters, n and p, which represent the number of trials and the probability of success, respectively. The binomial distribution is often used in pricing and promotion decisions, where the goal is to estimate the probability of different levels of demand under different marketing scenarios.

#### Geometric Distribution

The geometric distribution is commonly used to model demand for products or services that have a fixed probability of success on each trial, such as the probability of a customer purchasing a product on each visit. The geometric distribution is characterized by a single parameter, p, which represents the probability of success. The geometric distribution is often used in customer behavior modeling, where the goal is to estimate the probability of different levels of demand based on the probability of a customer purchasing a product on each visit or interaction.

#### Negative Binomial Distribution

The negative binomial distribution is similar to the geometric distribution, but allows for a variable number of trials before a success occurs. The negative binomial distribution is characterized by two parameters, r and p, which represent the number of failures before the first success and the probability of success, respectively. The negative binomial distribution is often used in inventory management and service operations, where demand can be unpredictable and sporadic.

#### Hypergeometric Distribution

The hypergeometric distribution is commonly used to model demand for products or services that have a fixed number of possible outcomes and a limited population to draw from, such as the demand for a specific product in a particular market segment. The hypergeometric distribution is characterized by three parameters, N, n, and k, which represent the total population, the sample size, and the number of successes in the population, respectively. The hypergeometric distribution is often used in market research and product development, where the goal is to estimate the demand for a product in a specific market segment.

Here we give an introduction into Poisson distribution.

The Poisson distribution is a probability distribution that is used to model the number of rare events that occur in a fixed interval of time or space. It is named after the French mathematician Siméon Denis Poisson, who first introduced it in 1837.

The Poisson distribution is particularly useful in situations where the occurrence of events is random and independent, and where the average rate of occurrence is known or can be estimated. The distribution is characterized by a single parameter, λ (lambda), which represents the average number of occurrences in the interval of interest.

The probability mass function (PMF) of the Poisson distribution is given by (Walpole et al., 2011):

where X is the random variable representing the number of occurrences, k is a non-negative integer, e is the base of the natural logarithm (approximately 2.718), and k! represents the factorial of k.

Suppose a customer service center receives an average of 12 calls per hour. We can use the Poisson distribution to calculate the probability of receiving a certain number of calls in a given hour. For example, the probability of receiving exactly 10 calls in an hour can be calculated as:

This means that there is a 9.4% chance of receiving exactly 10 calls in an hour, assuming that the number of calls follows a Poisson distribution with a mean of 12 calls per hour.

In a retail store, demand can be modeled using the Poisson distribution by treating each purchase as an occurrence that happens randomly over time. The Poisson parameter, lambda, represents the average number of purchases per unit of time, such as per hour or per day.

For example, suppose a retailer wants to forecast the number of purchases during a one-hour period at a particular store location. The retailer can collect historical data on the number of purchases made during one-hour periods at that location on previous days. The data might look something like this:

Table Title

|  |  |  |
| --- | --- | --- |
| Day of Week | Hour of Day | Number of Purchases |
| Monday | 12:00 PM | 45 |
| Tuesday | 1:00 PM | 50 |
| Wednesday | 2:00 PM | 35 |
| Thursday | 3:00 PM | 60 |
| Friday | 4:00 PM | 75 |
| Saturday | 5:00 PM | 90 |
| Sunday | 6:00 PM | 80 |

Source: [Mojtaba Nabipour], (2023).

Using this data, the retailer can estimate the Poisson parameter, lambda, for purchases during a one-hour period at the store location. The Poisson parameter, lambda, can be estimated by calculating the average number of purchases per one-hour period across all days of the week. In this case, the average number of purchases per one-hour period is:

lambda = (45 + 50 + 35 + 60 + 75 + 90 + 80) / 7 = 63.6

The retailer can then use the Poisson distribution to estimate the probability of different numbers of purchases during a one-hour period at the store location. For example, the retailer might want to estimate the probability of having more than 70 purchases during a one-hour period, which would require the store to be prepared to handle a higher volume of customers than usual.

Using the Poisson distribution with lambda = 63.6, the probability of having more than 70 purchases during a one-hour period can be calculated as follows:

P(X>70)=1-P(X<=70)= 1 - P(X = 0) - P(X = 1) - ... - P(X = 70) ≈ 0.004

This calculation shows that the probability of having more than 70 purchases during a one-hour period is quite low, at approximately 0.4%. This information can help the retailer make decisions about staffing levels, inventory management, and other aspects of store operations.

### Self-check questions

1. why is Gamma distribution being more suitable for demand description than normal distribution?

*The gamma distribution allows for a wider range of shapes and has a longer tail than the normal distribution, which makes it more suitable for describing demand patterns that may have skewness.*

## 1.3 Demand Models

As we discussed in the previous section, demand was considered as a stochastic quantity. An inventory often experiences a broad range of demand patterns for its items. Without identifying and distinguishing these patterns, planning a successful inventory management policy would probably fail. In many published papers, inventory models are classified rather than demand models. The figure below presents a proposed classification of inventory demand patterns based on the study of the nature of demand for inventory items proposed by A.H. Kobbacy and Liang in (Kobbacy & Liang, 1999).

**Types of Demand**

Source: [Mojtaba Nabipour] (2023), based on [Khairy A.H. Kobbacy] (1999, pg 358).

Demand patterns that are statistically predictable have relatively smooth and repetitive fluctuations and are analyzed using statistical forecasting techniques. Demand patterns can vary over time and are classified into two categories: time-dependent and time-independent demand patterns. While in the case of the former, the type of quantity that is required for replenishment is determined by the time when the decision to replenish is taken, whereas this is not the case with the latter. A time independent demand can be assumed to be constant if it is reasonably stable and is able to be accurately forecasted. Within the category of constant demand, there are two subtypes: absolutely constant demand (with minimal variation) and quasi-constant demand (with variation below a specified threshold). Whenever there is substantial uncertainty associated with a demand, it can be referred to as stochastic, which can be expressed by theoretical or empirical probability distributions. Demands that change with time are categorized based on the change factor. A seasonal variation, trend, or both may be responsible for time dependency of demand. Due to lumpy or uneven demand, statistically unpredictable demand patterns experience sudden fluctuations. It is possible to categorize them into approximative and non-approximative patterns. With an approximative pattern, an item's demand has severe, but regular spikes and dips that do not recur annually but at predictable intervals. Whereas non-approximative demand is characterized by severe and unpredictable highs and lows of demand that do not recur predictably. Demand patterns such as these cannot be forecasted using conventional forecasting techniques, including statistical techniques. Within a tolerable error range, they can be approximated by a statistically predictable pattern. As real-world examples illustrate, demand patterns that cannot be forecasted using conventional statistical techniques, including those with sudden fluctuations and severe, unpredictable highs and lows, fall into the category of statistically unpredictable demand patterns. These patterns are further divided into two subtypes: approximative and non-approximative patterns. An approximative pattern refers to demand that experiences regular spikes and dips at predictable intervals, although they do not recur annually. On the other hand, non-approximative demand patterns exhibit severe and unpredictable variations in demand that do not follow a predictable pattern. However, it is possible to approximate these demand patterns within an acceptable margin of error by using statistically predictable patterns.

It is rarely feasible to determine the demand rate in a purely objective manner. Instead, one may take advantage of the knowledge of inventory managers. By closely monitoring inventory levels and sales trends, they can identify patterns in customer behavior and adjust inventory levels accordingly. The knowledge and expertise of inventory managers play a crucial role in effectively managing demand rates. By closely monitoring inventory levels and sales trends, inventory managers can gather valuable insights into customer behavior and purchasing patterns. This information allows them to make informed decisions regarding inventory management and adjust stock levels accordingly.

Inventory managers' knowledge can be useful in several ways. By closely monitoring inventory levels, they can identify patterns and trends in customer demand. They can analyze sales data to determine which products are popular, which ones are declining in demand, and any seasonal or cyclical patterns that may influence customer buying behavior. This understanding enables them to anticipate fluctuations in demand and adjust inventory levels to meet customer needs efficiently.

Moreover, inventory managers can use their knowledge to optimize inventory replenishment strategies. By analyzing sales trends and monitoring inventory turnover rates, they can determine the appropriate timing and quantities for replenishing stock. This helps to avoid stock-outs and excess inventory, both of which can lead to financial losses and customer dissatisfaction. In summary, the knowledge of inventory managers is invaluable in understanding customer behavior, optimizing inventory replenishment, improving forecasting accuracy, and implementing effective inventory control strategies.

### Test Flowcharts

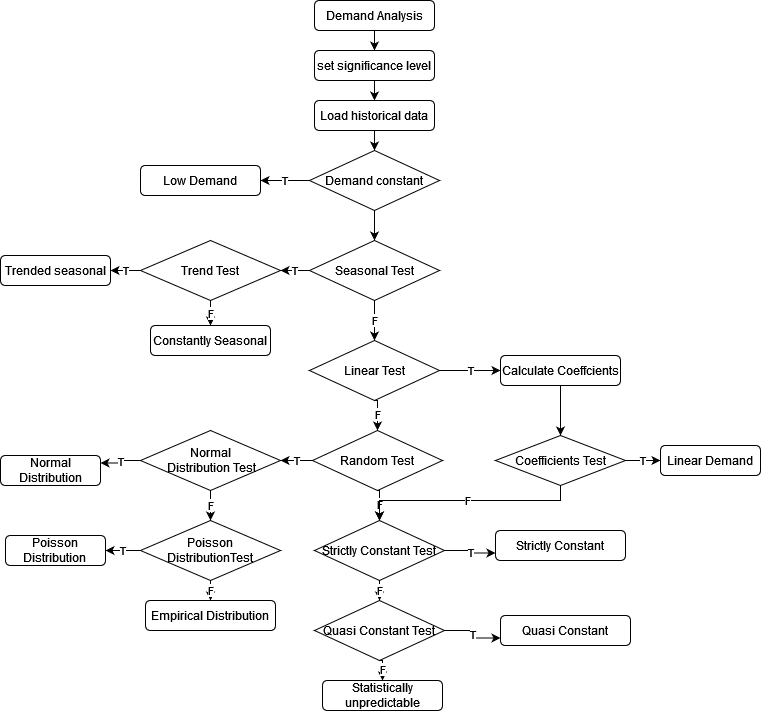
To determine the pattern of the historical demands, we can perform some statistical tests to separate the different demand classes that was shown in Figure. In the first step we should be able to detect any seasonal movement in historical demands and separate them from basic demands. Then the stationary, linear and probabilistic demand models should be identified. This process is demonstrated, as a flowchart, in Figure 8.

Some statistical tests can be used to detect seasonal movements. The Kruskal-Wallis test is a non-parametric rank-based test that is essentially employed to compare two or more samples to see if they belong to the same distribution or not(Ostertagova et al., 2014). The test is as follows:

The test is as follows:

(Eq.13)

The Identification Flowchart



Source: [Mojtaba Nabipour] (2023), based on [Eva Ostertagová] (2014, 3).

Where:

n = sum of sample sizes for all samples

g = number of samples

Ti = sum of ranks in the ith sample

ni = size of the ith sample

The steps through which, on can perform a test on the historical demands are

1. Sort data from different samples into ascending order
2. Compute the summation of rank for each sample separately
3. Compute the H test (Test-statistic from eq. 13)
4. Find the [critical chi-square (α) value](https://www.statisticshowto.com/how-to-find-a-critical-chi-square-value/) with L (L=g-1, g is the number of seasons) degrees of freedom
5. If H > α then demand is seasonal

There are also some statistical tests for detecting trends in data collected over time. The Mann-Kendall test is used to determine whether a trend in data values is statistically significant or if they are increasing or decreasing. Another similar test is Spearman-rank correlation which is used interchangeable with the Mann-Kendall test, while the Mann-Kendall test shows a more robust performance (Libiseller & Grimvall, 2002).

### Self-check Questions

1. explain a stationary demand pattern

*A stationary demand pattern is a type of demand pattern that remains constant over time. In other words, the mean and variance of the demand remain the same over time. This means that there is no trend or seasonality in the demand pattern.*

2. what Kruskal-Wallis test do exactly?

*The Kruskal-Wallis test is a statistical test that is used to compare two or more samples to see if they belong to the same distribution or not. It is a non-parametric rank-based test, which means that it does not assume that the data follows a normal distribution.*

3. how one can define the seasonality of a demand pattern?

*To define the seasonality of a demand pattern, one can look for patterns that repeat over a fixed period of time. Seasonality refers to the regular and predictable fluctuations in demand that occur at fixed intervals, such as daily, weekly, monthly, or yearly. In order to identify seasonality in a demand pattern, one can use statistical methods such as autocorrelation analysis or spectral analysis. These methods can help identify periodic patterns in the data and determine whether they are statistically significant.*

## 1.4 Multi-Period Newsvendor Model

A multi-period newsvendor model is an extension of the classical newsvendor problem, in which unsold items or unfulfilled demands are carried over to the next period to be dealt with. Thus, the inventory level of each period, instead of the first, does not equal the ordering level. So the problem here is to find the optimal inventory level for each period in order to maximize profit (Matsuyama, 2006). To achieve this, several models and assumptions have been proposed. In this section we used the approach introduced and proposed by Matsuyama in (Matsuyama, 2006). Instead of using the original notation, for the sake of reader’s convenience, we use previously listed one. We also ignore some discussed theorems and their proofs. Ones who would like more details may read the original paper. The extra notations will be introduced in the following.

### Model Consideration

With the model, we define how we handle unsold commodities or unfulfilled demands for the next period. Quantity ordering for period ***i*** is planned according to the following rules:

* Buying items is always accomplished so that the inventory level (li) becomes (lmin ≤ li ≤ lmax).
* If the received demands during the current period is smaller than the inventory level, li – xi indicates the unsold stocks. Some portions of the unsold items will take over to the next period according to α(li – xi) where 0 ≤ α ≤ 1 . The holding cost of this portion would be hiα(li – xi). hi is the holding cost of unsold items in period i. Carrying over this portion to the next cycle, the ordering quantity for period i+1 would be li+1 - α(li – xi).
* If there are some unsatisfied demands at the end of the current period it will cost a penalty equal to Cu(xi – li). If this is not the last period, some portions of the demands equal to β(xi – li) would be fulfilled in the next period (0 ≤ β ≤ 1). In that case li+1 + β(xi – li) units are complied with at the beginning of the next period with the selling price δsi + (1-δ)si+1  for each unit (0 ≤ δ ≤ 1).

### Profit function

To define the profit function for multi-period newsvendor model we need to introduce 4 auxiliary functions.

(Eq.14)

(Eq.15)

(Eq.16)

(Eq.17)

Function **gi** represents the profit of selling items during the period **i** and **di** function is what we consider doing with the unsold items or unfulfilled demands at the end of period **i** and start of period **i+1**. It is obvious that the d function is 0 for the terminal period.

Function **gi** represents the profit of selling items during the period **i** and the **di** function is what we consider doing with the unsold items or unfulfilled demands at the end of period **i** and start of period **i+1**. It is obvious that the d function is 0 for the terminal period. Therefore, the d function for a nonterminal period I will be

(Eq.18)

Assuming no setup cost if we have just one period, the profit would be

. (Eq.19)

In case of two periods the profit would be

(Eq.20)

The general recursive definition of the profit function for an N period newsvendor model is

. (Eq.21)

Defining the profit of an inventory planning during an interval consists of N periods in a none recursive form and leads to

(Eq.22)

Since x is a random variable, the profit becomes a random variable and the expected value of p is

(Eq.23)

where f(xi) is the density function of the random variable xi.

The problem of determining the optimal inventory level for period i can be done by taking the derivative of the expected value of the profit and solving the equation:

(Eq.24)

**Example:** compute the optimal inventory level for a retailer during a period consisting of 20 cycles with following assumption:

**α** = 0.8, **β** = 0.7, **δ** = 0.5, **Cu** = 20

1/20 50 ≥ xt ≥ 30

ft(xt) =

0 else

st = 200 + 0.8t

wt = 80 + 0.6t

ht = 2.5 + 0.03t

**Answer**

With random generation of demand level (xt) the following table is the 20 cycles simulation of the inventory.

The inverse of uniform cumulative probability calculated as:

*I*(*F(xt)*) = *30* + *F(xt)*(*50* − *20*)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t | s | w | h | x | F(L\*) | L\* |
| 1 | 200.8 | 80.6 | 2.53 | 39.4 | 0.760858 | 45.21716 |
| 2 | 201.6 | 81.2 | 2.56 | 34 | 0.759572 | 45.19145 |
| 3 | 202.4 | 81.8 | 2.59 | 34.9 | 0.758294 | 45.16588 |
| 4 | 203.2 | 82.4 | 2.62 | 39.6 | 0.757022 | 45.14045 |
| 5 | 204 | 83 | 2.65 | 48.6 | 0.755758 | 45.11516 |
| 6 | 204.8 | 83.6 | 2.68 | 47 | 0.7545 | 45.09001 |
| 7 | 205.6 | 84.2 | 2.71 | 42.4 | 0.75325 | 45.06499 |
| 8 | 206.4 | 84.8 | 2.74 | 32.8 | 0.752006 | 45.04011 |
| 9 | 207.2 | 85.4 | 2.77 | 35 | 0.750768 | 45.01537 |
| 10 | 208 | 86 | 2.8 | 48.8 | 0.749538 | 44.99075 |
| 11 | 208.8 | 86.6 | 2.83 | 35.4 | 0.748314 | 44.96627 |
| 12 | 209.6 | 87.2 | 2.86 | 35.3 | 0.747096 | 44.94192 |
| 13 | 210.4 | 87.8 | 2.89 | 35.5 | 0.745885 | 44.91771 |
| 14 | 211.2 | 88.4 | 2.92 | 49.1 | 0.744681 | 44.89362 |
| 15 | 212 | 89 | 2.95 | 33.9 | 0.743483 | 44.86966 |
| 16 | 212.8 | 89.6 | 2.98 | 48.6 | 0.742291 | 44.84582 |
| 17 | 213.6 | 90.2 | 3.01 | 48.3 | 0.741106 | 44.82211 |
| 18 | 214.4 | 90.8 | 3.04 | 32.7 | 0.739927 | 44.79853 |
| 19 | 215.2 | 91.4 | 3.07 | 34 | 0.738754 | 44.77507 |
| 20 | 216 | 92 | 3.1 | 43 | 0.419941 | 38.39882 |

### Self-check Questions

1. Explain the difference between classical newsvendor and multi period model

*The classical newsvendor model assumes that there is only one period of sales and any unsold items cannot be carried over to the next period. The goal of this model is to find the optimal order quantity that maximizes profit, given a fixed selling price and a fixed cost per unit. This model assumes that demand is known with certainty and follows a specific probability distribution. On the other hand, the multi-period newsvendor model is an extension of the classical newsvendor problem, in which unsold items or unfulfilled demands are carried over to the next period to be dealt with. Thus, the inventory level of each period, instead of the first, does not equal the ordering level. The goal of this model is to find the optimal inventory level for each period in order to maximize profit. This model takes into account demand uncertainty and allows for more flexibility in managing inventory levels over time.*

## 1.5 Extensions

In section 1.4 the classical newsvendor model was extended to a multi-period problem. It is possible to extend different variables of the classical newsvendor model to adapt different scenarios of the real world. In this section, we review two important extensions to the newsvendor model.

## 1.6 Multiproduct Newsvendor Model

Suppose a retailer who has expanded his/her business from selling only one product to a set of various products, try to define an optimal stocking quantity for every product. The problem can be formulated as follows (Choi et al., 2011). To determine the stocking policy or order quantity of each product, the same approach can be used as in the classical version. The only difference is that we calculate the profit for each product (πi) separately. Additionally, to provide more diversification, left over stocks were salvaged for a per unit price *g*. Thus the profit probability function can be written as:

(Eq.25)

The equation πi represents a mathematical expression that calculates the profit for a specific inventory item based on different conditions. It incorporates several variables and parameters to capture different scenarios in inventory management. by integrating of the above equation, the expectation of profit for each product will be achieved. The optimal order quantity (Q\*) maximizes the profit that can be computed by taking the derivative of the expectation equation.

. (Eq.26)

### Stocking Constraints

In the case of limited space of the inventory, the newsvendor model must be reformulated under the subject of space constraints. The space constraint is stated as follows.

,

where ai is the considered space for unit of product I and A is the inventory space. To solve the optimal stocking problem under the mentioned constrained, if the optimal solution violates the constraint, Hadley et al. (Hadley & Whitin, 1963) 10.1016/j.ejor.2005.11.002proposed a dynamic programming solution based on Lagrange multiplier (Abdel-Malek & Areeratchakul, 2007). The Lagrangian will be:

(Eq.27)

Since the Lagrangian is concave, the first order derivative is enough to determine the optimal Q.

(Eq.28)

In (Yao et al., 2006) the authors examine the stocking policy under the assumption of influence of pricing strategies on demand levels. Considering risk and disruptions of the supply chains, makes inventory policies robust and effective. In (Garvey & Carnovale, 2020) the authors proposed a novel inventory model that takes into account the propagation of disruptions in a supply chain. It is an updated version of the single-period newsvendor model, based on the use of Bayesian Networks, that takes into account the ripple effect of risk propagation throughout the supply chain. This model allows supply chain managers to minimize supply chain risk severity by directing their attention to the factors that have the greatest impact on overall risk severity in the network.

In many existing supply chains and inventory optimization models, the newsvendor model is the foundation. For their own needs and conditions, managers, retailers, producers, etc., extend the basic model.

### Self-check Questions:

1. Explain about some various flavor of classical newsvendor problem.

*1) Multi-period newsvendor model: This extension considers a scenario where unsold items or unfulfilled demands can be carried over to the next period. The goal is to find the optimal inventory level for each period in order to maximize profit.*

*2) Newsvendor problem with price-dependent demand: In this extension, demand is assumed to be a function of price, and the goal is to find the optimal price and inventory level that maximize profit.*

*3) Newsvendor problem with lead time: This extension takes into account the time it takes for an order to be fulfilled and delivered, and aims to find the optimal order quantity that minimizes expected cost while meeting demand requirements.*

*4) Newsvendor problem with quality-dependent demand: This extension assumes that demand is influenced by product quality, and aims to find the optimal inventory level and quality level that maximize profit.*

*5) Newsvendor problem with multiple products: This extension considers a scenario where there are multiple products with different demand patterns and production costs, and aims to find the optimal inventory levels for each product that maximize overall profit.*

*These extensions allow for more realistic modeling of real-world scenarios and can provide insights into how different factors affect inventory management decisions.*

Summary

In the economic literature, the newsvendor problem has been studied as a fundamental problem in stochastic inventory control since the eighteenth century, but there is still much work to be done. Supply chains for trendy and perishable products have been extensively analyzed using it. In recent decades, operational research has extensively studied the newsvendor problem, and its models have been extended to model a variety of problems that occur in the real world.

Unit 2 – Traditional Methods of Demand Forecasting

**Study Goals**

On completion of this unit, you will be able to…

…define demand forecasting.

…identify the traditional methods for demand forecasting.

…determine which method is suitable for a specific demand type.

…apply the methods discussed to predict future demands .

Basic Readings

Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and practice.

Peixeiro, M. (2022). Time series forecasting in Python. Manning Publications.

Shumway, R. H., & Stoffer, D. S. (2017). Time series analysis and its applications: With R examples (3rd ed.). Springer.

Bisgaard, S., & Kulahci, M. (2011). Time series analysis and forecasting by example (1st ed.). John Wiley & Sons.

Further Reading

Lindsey, J. K. (2004). Statistical analysis of stochastic processes in time (Vol. 14). Cambridge University Press.

Capasso, V., & Capasso, V. (2021). Introduction to Continuous-Time Stochastic Processes. Springer.

Bartlett, M. S. (1978). An introduction to stochastic processes: With special reference to methods and applications. Cambridge University Press.

Sepehri, A. (2021). Inventory management under carbon emission policies: A systematic literature review. In Decision Making in Inventory Management (pp. 187-218).

2. Traditional methods of Demand Forecasting

Introduction

Every process dealing with predicting the future of uncertain events can be called forecasting. Demand is an uncertain quantity that may follow a probabilistic distribution and can be impacted by certain factors such as seasonality and trends. Manufacturing managers must forecast future demands for their products to ensure they have the materials, labor, and capacity to meet them. Resource planning and scheduling takes place well before the firm is faced with product demands. Since retailers and even producers need to minimize their stocking costs, inventory control relies heavily on demand forecasting.

Any supply chain system needs to be resilient to change to be effective. Thus, managing complex supply chains requires sophisticated forecasting techniques. A historical flow of demand is used to make the forecasts. The assumption behind quantitative forecasting is that historical data adequately describes the scenario. These data are gathered in a successive measurement process from the same source. This collection is referred to as time series data. Demand for an item over time can be collected and analyzed as a time series to predict future demands, helping with better planning. Following is an explanation of traditional methods of demand forecasting. Throughout this unit, historical demand data analysis will be explained using traditional time series analysis methods. These methods rely on statistical concepts, which are briefly introduced as needed.

Indeed, the use of traditional demand forecasting methods spans across diverse industries and sectors, offering wide-ranging applications. These methods find relevance in key areas such as production planning, supply chain management, pricing and promotion strategies, financial planning and budgeting, and capacity planning. By employing traditional demand forecasting techniques, organizations are empowered to make well-informed decisions, optimize resource allocation, and effectively navigate dynamic market conditions.

2.1 Exponential Smoothing

Exponential smoothing is a popular method for forecasting time series in a wide range of industries. It is a statistical technique that uses a weighted average of past observations to predict future values. The method assumes that recent data points are more relevant than older ones. It assigns exponentially decreasing weights to past observations. This approach allows for the smoothing of irregularities in the data and provides a more accurate forecast.

Since exponential smoothing belongs to the category of average-based algorithms, it makes sense to begin with a brief description of these algorithms. Generally, they use some sort of averaging of historical data to forecast future data. Average-based methods have some advantages and disadvantages. The advantages of these methods include simplicity, speed, and low computational costs. However, they may not work well for time series with complex patterns or irregular fluctuations.

Exponential smoothing has proven to be particularly valuable in the retail industry, where precise demand forecasting plays a vital role in inventory optimization and efficient supply chain management. This technique enables retailers to anticipate customer demand accurately, align their inventory levels accordingly, and streamline their operations to meet market requirements effectively. Similarly, the financial sector leverages exponential smoothing techniques for market analysis and forecasting stock prices. By applying these algorithms to historical stock data, the inherent fluctuations are smoothed out, allowing investors to generate forecasts that provide valuable insights for making well-informed decisions regarding stock trading and investment strategies.

Cumulative average

A time series is a sequence of data points collected over time, typically at regular intervals. It is a set of observations that are ordered chronologically and can be used to analyze trends, patterns, and behavior over time. So the time series A can be defined as a set of observations Ai so that .

In the simplest form, the forecasting of next data *Ak+1* is considered as the average of all past data. This can be written as:

(Eq.1)

If the pattern of data is horizontal when data fluctuates randomly over time around a constant mean, creating a horizontal pattern**,** the mean of historical data can be used for the next forecast.

Average-based methods are applicable on stationary time series data where the statistical properties of the generating process are not changed over time.

To adapt better to abrupt changes in time series, we can just include the last ***M*** observations when taking an average. This method is called moving average in which a window of size ***M*** always slides forward to take an average of the last ***M*** observations (Figure). While this is an effective way to smooth out short-term fluctuations, it consumes less memory space than the full average method.

**Moving Average Method**

Ein Bild, das Reihe, Text, Diagramm, Screenshot enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Simple moving average

The second equation describes the method

(Eq.2)

The moving average method treats all data within a window equally, disregarding the fact that more recent observations hold greater significance compared to those further in the past. This limitation is what motivates the use of exponential smoothing, which assigns exponentially decreasing weights to the observations.

Exponential Smoothing

The recursive equation of the method is:

. (Eq.3)

Where:

α is the smoothing constant, yk is the last observation, Ak is the last forecast, and *Ak+1 is the next forecast.*

Expanding the equation 3 will reveal the naming implication which exponentially decays weights.

If we continue this substitution process, we eventually reach Eq. 4.

(Eq.4)

Accordingly, Ak+1 is the exponentially weighted average of all past observations. Putting α equal to 1/k gives the full average equation (Eq.1). The α coefficient must be between 0 and 1 and not equal to 0 or 1. A lower α value provides stable predictions with smoothed random variation. On the other hand, higher values provide the most rapid response to real changes in the time series. Beyond that, there are no guidelines for choosing a proper value for alpha in a specific scenario. One of the straightforward methods to estimate the α, is to compute the forecast for nine consecutive values of α from 0.1 to 0.9 and to choose the one with the highest level of accuracy based on minimum forecasting error (et). Forecasting error is referring to Yt and Ft as the actual and forecasted values of the time series for period t, we can determine the forecasting error: et = Yt - Ft

One of the major error measuring techniques is the root-mean-square error (RMSE), a technique based on forecasting error.

Equation 5 describes RMSE.

(Eq.5)

Where ek represents the difference between the predicted and observed values for each individual observation or time period.

Example 1:

Demand quantities of one product for 2 years is available in the table below. What is the optimal value of α?

|  |  |  |  |
| --- | --- | --- | --- |
| Demand history for two years | | | |
| DATE | DEMAND | DATE | DEMAND |
| Jan(Year1) | 97.6 | **Jan(Year2)** | 89.3 |
| Feb(Year1) | 95.1 | **Feb(Year2)** | 88.5 |
| Mar(Year1) | 90.3 | **Mar(Year2)** | 93.7 |
| Apr(Year1) | 92.5 | **Apr(Year2)** | 92.7 |
| May(Year1) | 89.8 | **May(Year2)** | 94.7 |
| Jun(Year1) | 92.7 | **Jun(Year2)** | 95.3 |
| Jul(Year1) | 94.4 | **Jul(Year2)** | 94.7 |
| Aug(Year1) | 96.2 | **Aug(Year2)** | 95.3 |
| Sep(Year1) | 88.9 | **Sep(Year2)** | 94.7 |
| Oct(Year1) | 90.2 | **Oct(Year2)** | 96.5 |
| Nov(Year1) | 88.2 | **Nov(Year2)** | 99.2 |
| Dec(Year1) | 91 | **Dec(Year2)** | 96.9 |

Source: [Mojtaba Nabipour], (2023).

The forecasting value for α=0.1, 0.3, 0.6 and 0.9 have been computed and listed in table 2 (Y^ denotes the forecasted value). As can be inferred from Figure 2, the forecasting series with α=0.6 are more consistent with original data. The computed RMSE confirms this guess.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Forecasted value with different values of α | | | | | |
| Date | Demand | Y^(α=0.1) | Y^(α=0.3) | Y^(α=0.6) | Y^(α=0.9) |
| Jan(Year1) | 97.6 | 97.6 | 97.6 | 97.6 | 97.6 |
| Feb(Year1) | 95.1 | 97.6 | 97.6 | 97.6 | 97.6 |
| Mar(Year1) | 90.3 | 97.35 | 96.85 | 96.1 | 95.35 |
| Apr(Year1) | 92.5 | 96.645 | 94.885 | 92.62 | 90.805 |
| May(Year1) | 89.8 | 96.2305 | 94.1695 | 92.548 | 92.3305 |
| Jun(Year1) | 92.7 | 95.58745 | 92.85865 | 90.8992 | 90.05305 |
| Jul(Year1) | 94.4 | 95.29871 | 92.81106 | 91.97968 | 92.43531 |
| Aug(Year1) | 96.2 | 95.20883 | 93.28774 | 93.43187 | 94.20353 |
| Sep(Year1) | 88.9 | 95.30795 | 94.16142 | 95.09275 | 96.00035 |
| Oct(Year1) | 90.2 | 94.66716 | 92.58299 | 91.3771 | 89.61004 |
| Nov(Year1) | 88.2 | 94.22044 | 91.86809 | 90.67084 | 90.141 |
| Dec(Year1) | 91 | 93.6184 | 90.76767 | 89.18834 | 88.3941 |
| Jan(Year2) | 89.3 | 93.35656 | 90.83737 | 90.27533 | 90.73941 |
| Feb(Year2) | 88.5 | 92.9509 | 90.37616 | 89.69013 | 89.44394 |
| Mar(Year2) | 93.7 | 92.50581 | 89.81331 | 88.97605 | 88.59439 |
| Apr(Year2) | 92.7 | 92.62523 | 90.97932 | 91.81042 | 93.18944 |
| May(Year2) | 94.7 | 92.63271 | 91.49552 | 92.34417 | 92.74894 |
| Jun(Year2) | 95.3 | 92.83944 | 92.45687 | 93.75767 | 94.50489 |
| Jul(Year2) | 94.7 | 93.08549 | 93.30981 | 94.68307 | 95.22049 |
| Aug(Year2) | 95.3 | 93.24694 | 93.72686 | 94.69323 | 94.75205 |
| Sep(Year2) | 94.7 | 93.45225 | 94.1988 | 95.05729 | 95.2452 |
| Oct(Year2) | 96.5 | 93.57702 | 94.34916 | 94.84292 | 94.75452 |
| Nov(Year2) | 99.2 | 93.86932 | 94.99441 | 95.83717 | 96.32545 |
| Dec(Year2) | 96.9 | 94.40239 | 96.25609 | 97.85487 | 98.91255 |

|  |  |  |  |
| --- | --- | --- | --- |
| RMSE(α=0.1) | RMSE(α=0.3) | RMSE(α=0.6) | RMSE(α=0.9) |
| 3.698793311 | 2.939379223 | 2.642176214 | 2.656275036 |

Source: [Mojtaba Nabipour], (2023).

Despite the computational simplicity and memory efficiency of the exponential smoothing, this method is not able to detect the trend or seasonality patterns of time series. The figure below is shown for better understanding.

**Chart of original data and forecasted values**

Source: [Mojtaba Nabipour], (2023).

Holt’s Exponential smoothing

To augment trend pattern detection capability to the simple exponential smoothing, a new parameter called trend factor is added to the original method to extend the exponential smoothing. The resulting method is called Holt’s exponential smoothing (Hyndman & Athanasopoulos, 2018). Here the assumption is the series has additive trend pattern where the trend is linear and the time series is the sum of its components, and therefore has level (L) and trend (T). This technique utilizes 3 equations and 2 smoothing parameters α and β to build a model to forecast.

In this model, the next value level in the series is estimated by equation 6. It is like simple exponential smoothing but now the trend factor has been accounted:

(Eq.6)

Then we try to estimate the trend factor:

(Eq.7)

The initial values for L and T are essential for starting the forecasting process. The specific values for L0 and T0 depend on the approach used and the available historical data. The setup permits the trend to fluctuate and alter its shape as time progresses. The pace at which the trend adjusts is regulated by the smoothing constant (β), hence if the trend undergoes rapid changes throughout the series, increasing the β makes it possible to catch up with the series. With this particular arrangement, the trend can fluctuate and alter its shape as time progresses. The rate at which the trend adjusts is determined by the smoothing constant (β), allowing for quick adaptation to sudden changes in the series. For instance, if the trend changes rapidly, a higher smoothing constant may be necessary to learn the new trend more quickly.

And finally we add the level function and trend factor to reach the final estimated value as:

(Eq.8)

m means how many steps towards the future we are trying to estimate.

**Example 2**

Table 3 demonstrates the application of Holt’s exponential smoothing method to forecasting the annual ticket demand for an airline between 1989 and 2016 (Hyndman & Athanasopoulos, 2018). The smoothing parameters and initial values for leave and are estimated by minimizing sum of squared error. the parameter values are α=0.8321 and β=0.0001. The very small value of β indicates that the slope is roughly constant over the course of time. From the 2016 we are out sampled and continue with the last row values and changing steps (m) to predict the future. Figure 3 indicates the method can capture the trend of demand.

**Holt's method for estimating the trend pattern**

Source: [Mojtaba Nabipour], (2023).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| The historical data of ticket demand and forecasting values | | | | | |
| Year | Time Step K | Observation(1000) | Level | Trend | Forecast |
| 1989 | 0 |  | 15.57 | 2.102 |  |
| 1990 | 1 | 17.55 | 17.57 | 2.102 | 17.67 |
| 1991 | 2 | 21.86 | 21.49 | 2.102 | 19.68 |
| 1992 | 3 | 23.89 | 23.84 | 2.102 | 23.59 |
| 1993 | 4 | 26.93 | 26.76 | 2.102 | 25.94 |
| 1994 | 5 | 26.89 | 27.22 | 2.102 | 28.86 |
| 1995 | 6 | 28.83 | 28.92 | 2.102 | 29.33 |
| 1996 | 7 | 30.08 | 30.24 | 2.102 | 31.02 |
| 1997 | 8 | 30.95 | 31.19 | 2.102 | 32.34 |
| 1998 | 9 | 30.19 | 30.71 | 2.101 | 33.29 |
| 1999 | 10 | 31.58 | 31.79 | 2.101 | 32.81 |
| 2000 | 11 | 32.58 | 32.8 | 2.101 | 33.89 |
| 2001 | 12 | 33.48 | 33.72 | 2.101 | 34.9 |
| 2002 | 13 | 39.02 | 38.48 | 2.101 | 35.82 |
| 2003 | 14 | 41.39 | 41.25 | 2.101 | 40.58 |
| 2004 | 15 | 41.6 | 41.89 | 2.101 | 43.35 |
| 2005 | 16 | 44.66 | 44.54 | 2.101 | 44 |
| 2006 | 17 | 46.95 | 46.9 | 2.101 | 46.65 |
| 2007 | 18 | 48.73 | 48.78 | 2.101 | 49 |
| 2008 | 19 | 51.49 | 51.38 | 2.101 | 50.88 |
| 2009 | 20 | 50.03 | 50.61 | 2.101 | 53.49 |
| 2010 | 21 | 60.64 | 59.3 | 2.102 | 52.72 |
| 2011 | 22 | 63.36 | 63.03 | 2.102 | 61.4 |
| 2012 | 23 | 66.36 | 66.15 | 2.102 | 65.13 |
| 2013 | 24 | 68.2 | 68.21 | 2.102 | 68.25 |
| 2014 | 25 | 68.12 | 68.49 | 2.102 | 70.31 |
| 2015 | 26 | 69.78 | 69.92 | 2.102 | 70.6 |
| 2016 | 27 | 72.6 | 72.5 | 2.102 | 72.02 |
|  | m |  |  |  |  |
|  | 1 |  |  |  | 74.6 |
|  | 2 |  |  |  | 76.7 |
|  | 3 |  |  |  | 78.8 |
|  | 4 |  |  |  | 80.91 |
|  | 5 |  |  |  | 83.01 |

Source: [Mojtaba Nabipour], (2023).

Holt-Winters smoothing method

Thanks to the previous method, one can capture the trend in time series forecasting. However, this method does not reveal the seasonality patterns, this is where the Holt-Winters smoothing method comes in. Holt (1957) worked with Winters (1960) to extend Holt's method to include seasonality as well (Hyndman & Athanasopoulos, 2018). The Holt-Winters method can be applied in two different ways depending on the type of seasonality. If the seasonality is additive the additive version must be used and if the seasonality is multiplicative (when the time series is the product of its components), the multiplicative version must be used. The three necessary components to describe the model are the level, trend and season.

Combining all together, the final equations can be expressed as follows. The additive model equations are:

(Eq.9)

Eq.10)

(Eq.11)

(Eq.12)

Eq. 9 to 11 express the level, trend and seasonality (with h seasons) update respectively. γ is the seasonal smoothing factor and acts just like α and β. Notice that in additive series we use subtraction to decompose the different components whereas in multiplicative series division will be used for decomposition purpose. Here, for the seasonal component update (Eq.11) the level component is removed using subtraction. Eq.12 performs the forecast considering the additive seasonality and additive trend.

If the series has the multiplicative seasonal component, the equation 9, 11 and 12 changes to (Makridakis et al., 2008):

(Eq.13)

(Eq.14)

(Eq.15)

Note that the only difference is, how seasonality composes (multiplication instead of addition) and how seasonality decomposes (division instead of subtraction). Moreover, in equation 14 to update the seasonality, the level component has been removed by division. The adjustments made to the composition (multiplication) and decomposition (division) of seasonality in the Holt-Winters smoothing method are valuable for capturing and accommodating seasonal fluctuations, resulting in a more inclusive and precise forecasting technique. By excluding the level component during the seasonality update, the method ensures that the estimation of seasonality is independent and accurately reflects the inherent seasonal patterns present in the data.

To initialize the level, seasonal and trend indices, the data of one complete season is needed. then, using following equations the mentioned indices can be initialized.

(Eq.16)

(Eq.17)

(Eq.18)

**Example 3:**

The seasonal demands of an item in a hypothetical hardware tools shop are recorded on a quarterly basis, as shown in Table 4. The chosen model assumes a combination of multiplicative seasonality and additive trend. Therefore, equations 13 to 15 are utilized to estimate the trend and make forecasts for future demands. The forecasting process begins from season 2, as it requires the collection of data for one complete season before it can commence.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Holt-Winters method to capture the seasonality of the quarterly demand data | | | | | | | |
| Year | Quarter | t | Demand | Lt | Tt | St | At+h |
| 1994 | 1 | 1 | 500 |  |  | 1.333333 |  |
|  | 2 | 2 | 350 |  |  | 0.933333 |  |
|  | 3 | 3 | 250 |  |  | 0.666667 |  |
|  | 4 | 4 | 400 | 375 | -12.5 | 1.066667 |  |
| 1995 | 1 | 5 | 450 | 396.9667 | -9.05333 | 1.273412 | 483.3333 |
|  | 2 | 6 | 350 | 372.3747 | -10.6072 | 0.935307 | 362.0524 |
|  | 3 | 7 | 200 | 296.7938 | -17.1046 | 0.668827 | 241.1783 |
|  | 4 | 8 | 300 | 287.3869 | -16.3348 | 1.059833 | 298.3352 |
| 1996 | 1 | 9 | 350 | 302.1219 | -13.2278 | 1.23893 | 345.161 |
|  | 2 | 10 | 200 | 252.9623 | -16.821 | 0.891905 | 270.2048 |
|  | 3 | 11 | 150 | 201.4173 | -20.2934 | 0.691596 | 157.9377 |
|  | 4 | 12 | 400 | 268.2504 | -11.5807 | 1.189227 | 191.9611 |
| 1997 | 1 | 13 | 550 | 373.5062 | 0.102908 | 1.309011 | 317.9958 |
|  | 2 | 14 | 350 | 363.8087 | -0.87713 | 0.912946 | 333.2237 |
|  | 3 | 15 | 250 | 317.4823 | -5.42206 | 0.720351 | 251.002 |
|  | 4 | 16 | 550 | 406.7605 | 4.047961 | 1.238103 | 371.1103 |
| 1998 | 1 | 17 | 550 | 465.9614 | 9.563264 | 1.270414 | 537.7528 |
|  | 2 | 18 | 400 | 444.9496 | 6.505758 | 0.908756 | 434.1286 |
|  | 3 | 19 | 350 | 410.5851 | 2.418728 | 0.759978 | 325.2062 |
|  | 4 | 20 | 600 | 487.3071 | 9.84905 | 1.236049 | 511.3412 |
| 1999 | 1 | 21 | 750 | 597.7855 | 19.91199 | 1.265679 | 631.5942 |
|  | 2 | 22 | 500 | 570.255 | 15.16774 | 0.899169 | 561.3363 |

Source: [Mojtaba Nabipour], (2023).

In this scenario α, β and γ are set to 0.4, 0.1 and 0.3, respectively. Figure 4 depicts that the method has captured the seasonality of the demand data reasonably.

**Capturing Seasonality Component of the Time Series Using Holt-Winters Technique**

Source: [Mojtaba Nabipour], (2023).

Self-check Questions:

1- Considering following table as a 6-month demand history, calculate RMSE for an exponential smoothing model with α=0.54.

|  |  |
| --- | --- |
| Date | Demand |
| Jan | 89.5 |
| Feb | 95.6 |
| Mar | 94.3 |
| Apr | 92.1 |
| May | 95.8 |
| Jul | 88.7 |

2- In example 3 use the multiplicative trend with additive seasonality model. Then compare the RMSE of resulted model by the original one.

2.2 Autoregressive Integrated Moving Average (ARIMA)

A different approach to time series forecasting is provided by autoregressive integrated moving average (ARIMA) models (Box et al., 2015). ARIMA (Autoregressive Integrated Moving Average) was developed in the 1970s. It is a statistical technique that is widely used to forecast future values of a time-series variable based on its past values. The ARIMA method is based on the assumption that the time-series data can be decomposed into three components: trend, seasonality, and randomness. The trend component represents the long-term direction of the time-series data, while the seasonality component represents the recurring patterns in the data. The randomness component represents the unpredictable fluctuations in the data.In time series forecasting, exponential smoothing and ARIMA models provide complementary approaches. Exponential smoothing makes use of trend and seasonality to describe data, whereas ARIMA uses autocorrelations to describe data that obtained from (Hyndman & Athanasopoulos, 2018). Mathematically, autocorrelation represents the statistical relationship or correlation between a variable and its lagged values. ARIMA as the name suggests, uses autoregressive and moving average processes to model the time series. It is therefore reasonable to begin by explaining these two processes, but let's first consider the concept of stationarity.

Stationarity is simply held when the statistical properties of a stochastic process like the one that generates a time series are not changed by time. Therefore, time series with trends can’t be stationary because the mean of the data is not constant over time. Or a seasonal time series also can’t be stationary because of lack of constant variance.

Autoregressive process

In an autoregressive model, the variable of interest can be expressed as a linear combination of its lagged values when the values of variable that occurred at previous time points. The formal definition of an autoregressive process of order *p* can be written as:

(Eq.19)

AR stands for autoregressive process and εt is white noise or error term with mean of zero and variance of σ2. A wide range of different time series patterns can be handled by autoregressive models (Hyndman & Athanasopoulos, 2018). Φ coefficients are the process parameters which determines the dependency degree of current value to the lagged values. The C parameter is the constant value of the process. It can be considered as the noise average too. In some literature it doesn’t appear separately.

**Example:**

Figure 5 shows the generated series from an AR(2) process that is expressed by equation 20.

(Eq.20)

**An Autoregressive Time Series Generated by Equation 20**

Source: [Mojtaba Nabipour], (2023).

The variance of white noise doesn’t change the pattern only changes the scale (Figure 6)

**Increasing the White Noise Variance to 1.5**

Source: [Mojtaba Nabipour], (2023).

The series generated by autoregressive processes are stationary, by construction. But there are some constraints for coefficients φ in order to make the process stationary (Hyndman & Athanasopoulos, 2018).

For an AR(1) process:

For an AR(2) process:

In higher orders the constraints become more complicated

Moving average Process

Instead of using past values to estimate the current value, a moving average (MA) process utilizes the linear combinations of past forecast error terms to forecast the future value. It is called the moving average process (the name sounds similar to what we saw before but the concept is totally different) since it is assumed that a stationary process is a random fluctuation around a forward moving average of the series (white noise process).

. (Eq.21)

Equation 21 describes the general form of a MA process. The depth of the considered lagged errors is determined by *q*. Like AR processes, moving average processes can only model stationary time series. To model nonstationary time series with these two methods, we need to transform the original series to obtain a stationary time series. To do that, one of the major transformation techniques is differencing, which essentially calculates the difference of consecutive observations, to remove the level change between them.

Differencing can be done multiple times, which indicates its order.

For example, the first order differencing is:

(Eq.22)

We can continue to reach second order and third order differencing via:

(Eq.23)

(Eq.24)

For example, the figure 6 depicts the Dow Jones closing averages during . Since it has an obvious upward trend, it can’t be stationary. First order differencing of the mentioned time series has been demonstrated in figure 8.

**Dow Jones Time Series**

Source: [Mojtaba Nabipour], (2023).

This transformation eliminates the trend of the time series and this time it exhibits a random pattern with constant mean and variance which convinces us to consider it as a stationary time series.

**First Order Differencing**

Source: [Mojtaba Nabipour], (2023).

Non-seasonal ARIMA Process

As we saw in the previous sections, moving average and autoregressive processes are capable of modeling stationary time series. With the aid of differencing, we can make nonstationary time series with trend components, stationary. It would be reasonable to integrate these methods to produce a pervasive time-series model.

We can get a non-seasonal ARIMA model by combining differencing with autoregression and moving averages. The model is described fully by equation 25.

(Eq.25)

Like before *p* and *q* indicate the order of lags in autoregression and moving average processes, respectively (Hyndman & Athanasopoulos, 2018). The order of differencing is indicated by *d*. ARIMA(*p,0,0)* is an autoregressive process (AR(*p*)) and ARIMA (*0, 0, q*) is a moving average process (MA(*q*)).

Table 5 listed some examples of ARIMA processes.

|  |  |
| --- | --- |
| Process | Equation |
| ARIMA(1,1,2) |  |
| ARIMA(1,2,1) |  |
| ARIMA(2,2,0) |  |
| ARIMA(1,0,2) |  |

It can be concluded that the differencing order *d* is the necessary steps of differencing to transform a nonstationary time series into a stationary time series. To determine the appropriate order of differencing, we typically assess the stationarity of the time series at different levels of differencing. If the initial time series is highly nonstationary, we start with a first-order differencing (d = 1) and check if the resulting differenced series exhibits stationarity.

In the following, the main stages of an ARIMA modeling will be discussed according to (Peixeiro, 2022).

* Identification test to determine if the time series is stationary or not
* Choosing appropriate model’s parameters i.e. *p, d* and *q*
* Fitting the best model to the observations
* Estimating the coefficients φ and ϴ
* Validating the model

Checking if the residuals (Forecasting errors) have a random pattern if they show any correlation, the model has to be changed (Ramos & Oliveira, 2016). Forecasting thorough the model. Moreover, we can use some accuracy metrics like RMSE. In order to make the concepts ready to be practical we avoid complicated statistical formulas and, the above stages will be explained by *python* codes.

In the beginning, we need to determine if a time series is stationary or if not, how many differencing steps we need to apply to obtain stationarity. By looking at the time series plot we might see easily if there is any strong upward or downward trend. However, if the time series is difficult to judge by just looking at the plot, we need more sophisticated tools.

There are some statistical tests that can be utilized to examine the stationarity of time series. One of them is called ADF test (Augmented Dicky-Fuller) in which the null hypothesis is that the series is none stationary and if the *p-value* is not lower the significant value (0.05) it is failed to reject the null hypothesis and the time series is not stationary (Dickey, 2015).

In the python programming language, we can use the adfuller(time-series) function from the statsmodels package to get the *p-value.*

For example, lets determine the stationarity of the time series of figure 6.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

p-value= 0.36133347874971455

Since the p-value is greater than 0.05 the we can’t reject the null hypothesis and the time series is not stationary. Subsequently, an examination shall be conducted to ascertain whether the first-order differencing process engenders stationarity within the time series under consideration. To difference we use a method called **diff** that takes the data and number of steps as parameters. The following code snippet shows Ein Bild, das Text, Schrift, weiß, Screenshot enthält.

Automatisch generierte Beschreibunghow to use it.

p-value= 0.0034683081217674982

The p-value allows us to reject the null hypothesis so the first order differencing would be enough. Actually, through differencing and ADF test we can determine the value of *d* in the ARIMA model. The next step is to define the value of *p* and *q*. Parameter *p* is the order of the autoregressive model. Basically, it tells how many lags the current value is correlated with. Thus, we need to analyze the autocorrelation of the time series.

Autocorrelation analysis provides us with Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) that can be great help not only for stationarity determination but also in model identification (*p, q*). The ACF measures the correlation of the time series with its past values.

(Eq.26)

But when the lag extends beyond one, it raises a question:

How much information can we gain from yt-k about yt that we can’t obtain from in between lags. This question guides us to another function called the Partial Autocorrelation function (PACF). This function does the same thing as ACF after removing the effects of in between observations. For example, PACF (yt , yt-3) is the correlation that is not explained by yt-1 and yt-2. PACF makes it possible to determine the number of lags that are needed to describe the current value, so no surprise that one uses it in ARIMA models to determine the value of *p.* Therefore, after the most significant lags in a partial autocorrelation plot we face a drop in correlation the index of the last lag is used to determine *p*.

Likewise, the ACF plot can tell us similar things about MA process of the ARIMA model where, after the significant contributing past error terms the plot values drop below the confidence region which is typically represented as a band around the predicted values in a time series plot. The width of the band depends on the level of confidence chosen by the analyst, which is often set at 95% or 99%. So, we can consider the last significant lag index as *q*.

**Example:**

We are given a data set that records the usage amount of a web server over several minutes. What is the appropriate ARIMA model to describe the data?

We start by plotting the time-series

**Time Series Plot of the Example**

Ein Bild, das Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Obviously, it is not a stationary time series. Let’s start with first order differencing

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

The test shows that we can’t reject the null hypothesis. Thus, we continue with the second order differencing. Figure 10 shows the p-value of ADF test and the plot of the time series after second order differencing.

**The Plot of the Second Order Differenced Time Series**

Ein Bild, das Text, Schrift, Reihe, Diagramm enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

The plot looks like a stationary time series and the ADF test suggests that *d* is 2.

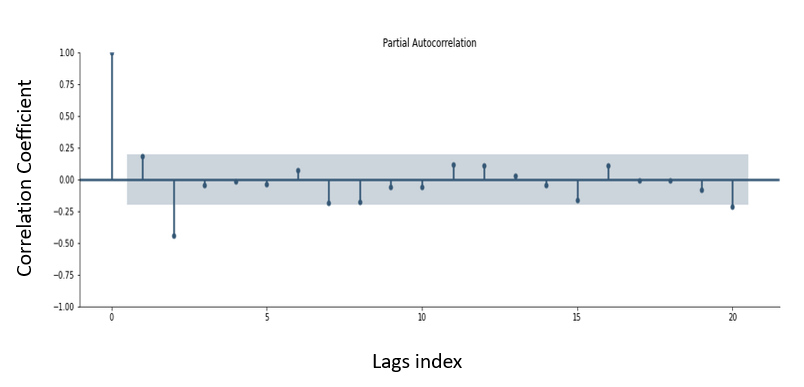
Now to reveal the value of *p*, the PACF plot will be utilized.

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

As the PACF plot indicates (Figure 11) after lag 2 there is no significant (shaded area) autocorrelation. Thus, we can decide not to go further than lag 2 for the autoregressive part (*p=2*)

**Partial Autocorrelation of the Time Series**



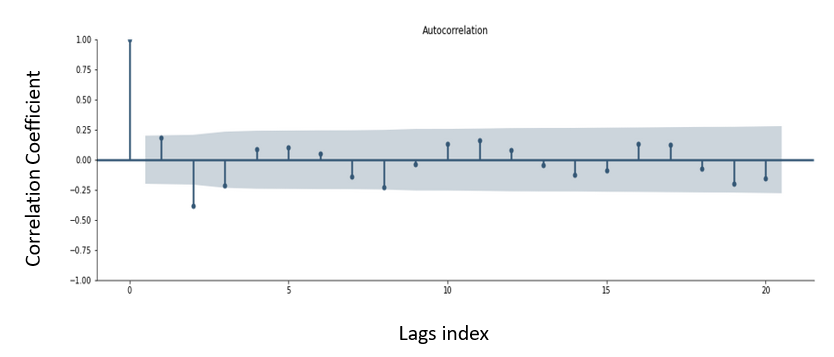
Source: [Mojtaba Nabipour], (2023).

Finally, by looking at the last lag crossings the threshold in ACF plot (Figure 12) *q* can be estimated.

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

**Autocorrelation of the Time Series**



Source: [Mojtaba Nabipour], (2023).

From the plot above the *q* parameter should be 2. Thus now the final ARIMA model can be defined as ARIMA (2,2,2).

To determine the coefficients, we need to fit the model to the data. In order to do that ARIMA class from the **statsmodels.tsa.arima.model** module will be used.

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

The summary of the model fitting is depicted in figure 13

**Summary of ARIMA Fitting Results**

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Eventually we can see how well the model has been fitted by plotting the actual series alongside with the predicted one (Figure 14).

Aiming to predict future values using autocorrelation measures within a time series, Auto Regressive Integrated Moving Average (ARIMA) incorporates autocorrelation measures into the model. It measures the dependence of a particular sample on a few previous observations through autoregression. By measuring and integrating these differences, data patterns become stationary, or the correlation with past data is minimized. (Linearity and no collinearity are fundamental assumptions of the linear regression model). Data is then condensed, and significant features are brought out through a moving average.

Self-check Questions:

1. Compute the average value of an autoregressive process described by following equation for 50 iterations.
2. Explain the procedure of ARIMA modeling briefly.

* *Stationarity: Check if the time series is stationary, meaning that its statistical properties such as mean and variance do not change over time. If not, transform the data to achieve stationarity.*
* *Differencing: If the data is not stationary, take differences of the data until it becomes stationary.*
* *Autocorrelation and Partial Autocorrelation Analysis: Analyze the autocorrelation and partial autocorrelation functions of the differenced data to determine appropriate values for p, d, and q in the ARIMA model.*
* *Model Selection: Select an appropriate ARIMA model based on the results of step 3.*
* *Model Fitting: Fit the selected ARIMA model to the data.*
* *Model Validation: Validate the fitted model by checking its residuals for* randomness and independence.

1. What does the AIC criterion tell us about an ARIMA model?

*The AIC (Akaike Information Criterion) is a statistical measure that evaluates the goodness of fit of a model while taking into account the number of parameters used in the model. In ARIMA modeling, the AIC criteria can be used to compare different models and select the one with the lowest AIC value. The lower the AIC value, the better the model fits the data while using fewer parameters. Therefore, AIC criteria can help us determine which ARIMA model is best suited for a given time series data.*

**Predicting Time Series Using the Fitted ARIMA Model**

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Observations

Time Steps

Source: [Mojtaba Nabipour], (2023).

2.3 State-Space Models

Any time series we like to model and predict is the result of consecutive observations of a system’s output. A system can be realized as a finite set of states, some of which are available to measure and some are not So called, "hidden states". As a transition between different states, the system produces observations and outputs. The state-space model is like a statistical model, but with two fundamental differences. Observations in state-space models may be noisy or difficult to measure. So, the state-space model assumptions are more realistic because they explicitly model the underlying system dynamics that generate the observed data . The other difference is the model updating characteristic, so that as the model is running, it incorporates prior knowledge to improve the prediction. The more interesting advantage that the model offers, especially for time series analysis, is that the focus is on the process that generates noisy data, instead of noisy data itself. It means these kind of models try to understand the dynamics behind the process rather than describing the relationships between process’s outcomes.

The general form of output generation (The linear form with additive noise) and state evolution can be written as in equation 26 and 27.

(Eq.26)

(Eq.27)

Zt describes the linear model of the system that generates the observations. α is the state vector of the system through which the system evolves. T is the transition function that controls the state evolution. It describes the system dynamics. *d* and *c* are the intercepts. ε is the measurement noise and η is the disturbance noise of the system. If the system is Gaussian, then we have:

(Eq.28)

(Eq.29)

**Time series example**

Consider an additive time series where the observations consist seasonal and trend components so we may write it as:

(Eq.30)

μ, γ and ε are the trend, seasonality and error components respectively. Suppose we can model the trend component as a simple random-walk process as in equation 31

(Eq.31)

If μt = αt and γt = 0 and the error is a white noise process, we can write the time series as:

(Eq.32)

. (Eq.33)

As it can be seen, the time series has been developed into a state-space model where equation 32 is the measurement model that relates the observation (y) to the state (α). Equation 33 describes the dynamics of the state evolution.

In the following, we explore the Kalman filter, which is the most used method for state-space models.

The Kalman Filter

The Kalman filtering method is the most developed and deployed method for state-space models, in which in an iterative process tries to minimize the error of prediction by incorporating new measurements. Thus, the Kalman filtering process contains two iterative steps that are depicted in Figure 15.

**Kalman Iterative Process**

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Regarding the previous state-space model equations, the prediction step contains following equations.

(Eq.34)

(Eq.35)

The ‘ notation indicates the predicted term prior to observation or measurement. And the superscript T is the transposition operation and defines the linear relationship between the current state vector and the previous state vector. The P is the error covariance which expresses the uncertainty about the state. In the first phase next state and error covariance are predicted. Receiving a new observation, the correction or filtering phase will begin according to the following three equations. R is the measurement error variance.

(Eq.36)

Equation 36 computes the Kalman gain that is used to correct the estimation of the next state by:

(Eq.37)

Intuitively speaking, the Kalman gain regulates the impact of the prediction error on the state correction part of the estimation. By time advances the Kalman gain is expected to decrease. The last equation updates the error covariance.

. (Eq.38)

An example of Kalman filtering, in demand forecasting, can be observed in (Kandananond, 2014).

To explain stationary demand processes, the first order autoregressive model is commonly used

(Eq.39)

Equation 39 is linear with white noise, so we can use Kalman filtering to predict the time series.

The output can be written as

(Eq.40)

where v is the measurement noise. The Kalman filter equations, considering the given system, are:

**The time update equations**

(Eq.41)

(Eq.42)

**Measurement update equations**

(Eq.43)

(Eq.44)

(Eq.45)

As an illustration, the result for φ = 0.9 is shown in (Kandananond, 2014).

**Self-check Questions:**

1. What are the differences between state-space modeling and statistical modeling?

* *Observations: In state-space models, observations may be noisy or difficult to measure, whereas statistical models assume that observations are precise and error-free.*
* *Model Updating: State-space models have a characteristic of model updating, meaning that as new data becomes available, the model is updated to incorporate the new information.*

*Statistical models do not have this characteristic. State-space models also offer several advantages over classical difference-in-differences schemes, including inferring the temporal evolution of attributable impacts, incorporating empirical priors on parameters into fully Bayesian models, and accommodating multiple sources of variation.*

2. Explain the two main stages of Kalman filtering.

* *Prediction: In the prediction step, the Kalman filter uses the previous state estimate and its covariance to predict the current state and its covariance. This prediction is based on the system dynamics model, which describes how the state evolves over time.*
* *Update: In the update step, the Kalman filter incorporates new measurements to improve its estimate of the current state. The update is based on a comparison between the predicted measurement and the actual measurement, as well as their respective uncertainties.*

*These two steps are repeated iteratively as new measurements become available, with each iteration improving the accuracy of the state estimate. The Kalman filter is widely used in various fields such as engineering, economics, and finance for its ability to provide accurate estimates even in noisy environments.*

2.4 Bayesian Structural Time Series Analysis

Analysis of time series data presents some of the most difficult analytical challenges. As a result of a limited amount of data but needing to inform some of the most important decisions, for example, corporate planning, commonly uses time series analysis to forecast demand, which requires understanding seasonality, trends, and quantification of the impact of business. In most cases, however, the lack of historical data makes it difficult to estimate these components accurately. Uncertainty can be handled transparently by Bayesian structural time series models. The Time series forecasting, feature selection, and other applications are all done using the Bayesian structural time series model (BSTS). The BSTS model is a statistical method that is utilized for various applications, including feature selection and time series prediction. It is specifically designed to be used with data that is in the form of a time series.Time series analysis using the Bayesian method has several advantages that can be listed as follows.

* Versatility: Bayesian methods offer the flexibility to model complex systems by incorporating prior knowledge and assumptions, enabling more precise and reliable predictions.
* Quantifying Uncertainty: Bayesian methods provide a means to quantify uncertainty in both model parameters and predictions. This is particularly valuable in time series analysis where data can be noisy and subject to uncertainty.
* Model Comparison: Bayesian methods provide a framework for comparing different models and selecting the most suitable one based on the available data.
* Integration of External Information: Bayesian methods facilitate the integration of external information, such as expert knowledge or additional data sources, into the modeling process.
* Real-time Adaptation: Bayesian methods can be updated in real-time as new data becomes available, allowing for more accurate and timely predictions.

With time series models, overfitting is a problem, especially when estimating models with many parameters over a short period. Despite not being a problem in this case, when dealing with multiple variables, such as in economic forecasting, it can be problematic. In order to solve the overfitting problem, we need to take a Bayesian approach. This allows us to prioritize our variables. As a result, it can be used to assess the extent to which different marketing campaigns have contributed to the change in product sales, brand popularity, and other relevant indicators that are related to product sales and brand popularity. There are several factors that can be considered in state-space models as opposed to classical difference-in-differences schemes, including inferring the temporal evolution of attributable impacts, the incorporation of empirical priors on parameters into fully Bayesian models, and the ability to accommodate multiple sources of variation, including contemporaneous covariates that change over time.

A Bayesian structural time series model with unobserved components would be used. In comparison to ARIMA models, this technique is more transparent and elegantly deals with uncertainty. Due to its lack of differencing, lags, and moving averages, it is more transparent. Model components can be visually inspected. By quantifying posterior uncertainty, controlling variance, and imposing prior beliefs on the model, it handles uncertainty better. In addition, any ARIMA model can be recast as a structural model. The Bayesian structural model can be summarized as follows:

(Eq.46)

Here xt denotes a set of regressors, St represents seasonality, and μt is the local level term. Latent state evolution over time is referred to as the unobserved trend at the local level. For instance, it can represent underlying growth in brand value or external factors that are hard to track, but it can also absorb short-term fluctuations that should be explicitly controlled. As a result, strange coefficient estimates due to spurious relationships can be avoided. In case the historical data cannot yield meaningful estimates, the outside priors for the means can be specified by the Bayesian nature of the model.

BSTS uses linear Gaussian models that are similar to Kalman filtering to provide more insight into the constituent components of series like seasonality, trend, and regression (Fildes, 1991).

The Bayesian approach always tries to estimate the distribution of future states with the help of previous observations and previous state distribution. This approach is based on the Bayes theorem.

. (Eq.47)

Here, θ is the hidden state, p(θ|y) is the posterior distribution given observations y. p(y) is the marginal probability of y and is given by:

. (Eq.48)

The state evolution can be stated as:

(Eq.49)

Where T is the transition function and can include all the mentioned components. Under the normal distribution of the error terms (η and ɛ) it can be proven that the posterior distribution of the states is always less uncertain (Mathys et al., 2011).

The state transition usually modeled by a Markov Chain. Using Markov Chain Monte Carlo sampling, one can simulate complicated distribution function. Using the general model (Eq.46) and Bayesian inference we can forecast the time series.

For better understanding, there is an example of how Bayesian structural time series (BSTS) method can be applied in forecasting time series. Suppose we want to forecast the monthly sales of a retail store for the next year. We have historical sales data for the past 5 years and we want to use this data to build a forecasting model. We can use the BSTS method to model the time series data. The BSTS method decomposes the time series into several components, including trend, seasonality, and any other relevant factors. The method then estimates the parameters of each component using Bayesian inference. Here's how we can apply the BSTS method to our example:

* Data preparation: We start by preparing the data by converting it into a time series format. We can then visualize the data to identify any trends, seasonality, or other patterns.
* Model specification: Next, we specify the model by defining the components that will be used to model the time series. For example, we can include a linear trend component, a seasonal component, and any other relevant factors such as promotions or holidays.
* Prior specification: We then specify the prior distributions for each parameter in the model. This allows us to incorporate any prior knowledge or assumptions about the underlying system.
* Bayesian inference: We use Bayesian inference to estimate the posterior distribution of each parameter in the model. This involves updating the prior distributions with the observed data to obtain the posterior distributions.
* Forecasting: Finally, we use the estimated model parameters to make forecasts for future time periods. The forecasts will include uncertainty intervals that reflect the uncertainty in the model parameters.

Self-check Questions:

1. How does Bayesian inference reduce the uncertainty of the forecasting procedure?

*Bayesian inference is a statistical approach that uses prior knowledge and data to update the probability of a hypothesis. In the context of time series forecasting, Bayesian inference can be used to reduce uncertainty by incorporating prior knowledge about the system being modeled. By using Bayesian methods, we can incorporate prior beliefs about the parameters of the model into our analysis. This allows us to make more informed predictions and reduce uncertainty in our forecasts. Additionally, Bayesian methods allow us to update our beliefs as new data becomes available, which can further improve the accuracy of our forecasts. Overall, Bayesian inference provides a powerful tool for reducing uncertainty in time series forecasting by allowing us to incorporate prior knowledge and update our beliefs as new data becomes available.*

**Summary**

Traditional methods of demand forecasting heavily rely on statistical analysis of time series. Such methods assume that the underlying demand pattern follows a certain pattern, and that future demand is similar to past demands. Statistical analysis allows for the identification of seasonality, trends and other patterns in the data which can provide insight into future demand. Methods such as exponential smoothing, moving average and ARIMA models assume that the time series are stationary with fixed variance and the observations are available. In contrast, Kalman filtering and BSTS models try to estimate hidden mechanism that generates the observations. Thus, they are capable of forecasting variables that may not necessarily be observable. This, however, requires much more information and computation than the first category of methods.

Unit 3 – Data-Driven Methods for Demand Forecasting

**Study Goals**

On completion of this unit, you will be able to…

… define data-driven methods.

… identify the data-driven methods for demand forecasting.

…determine which method is suitable for a specific demand type.

…apply the methods discussed to predict future demands.

Basic Reading

Blokdyk, G. (Year). Recurrent neural network: real life actions: practical tools for self-assessment.

Joseph, M. (2022). Modern Time Series Forecasting with Python: Explore industry-ready time series forecasting using modern machine learning and deep learning. Packt Publishing.

Further Reading

Dunis, C., Zhou, B., & Robinson, A. (Year). Applied Time Series Analysis with Neural Networks.

Akhundov, S., & Veys, Y. (Year). Time Series Forecasting with Deep Learning: A Survey.

Brownlee, J. (Year). Time Series Forecasting with Python and Deep Learning.

Smith, M. (Year). Time Series Forecasting: A Machine Learning Approach.

3. Data-Driven methods of Demand Forecasting

Introduction

Data-driven methods for demand forecasting use statistical and machine learning techniques to analyze historical data and predict future demand for a product or service. These approaches depend on having access to substantial quantities of data, which can be gathered from diverse sources like sales logs, customer actions, and market patterns. By examining this data, companies can obtain valuable information about consumer behavior and use it to make well-informed choices regarding inventory control, production scheduling, and marketing tactics. Data-driven demand forecasting can help businesses optimize their operations, reduce costs, and improve customer satisfaction by ensuring that the right products are available at the right time and in the right quantities. Traditionally, demand forecasting is modeled as a time series consisting of linear components. It is often the case that the real world involves models that are more complex. With the aid of artificial intelligence techniques such as supervised learning and neural networks, we can handle this complexity and do more accurate modeling. in this unit we get familiarize with the concept and applications of these techniques in demand forecasting.

3.1 Recurrent Neural Networks

Recurrent neural networks (RNNs) belong to a specific category of artificial neural networks intended for handling sequential data. Unlike traditional feedforward neural networks that process input data in a sequential manner, RNNs possess the capability to retain information from past inputs, enabling them to effectively process input sequences. RNNs possess a memory that enables them to capture temporal relationships within data. This quality makes them highly suitable for tasks like speech recognition, language translation, and time series prediction. The fundamental component of an RNN is the recurrent neuron, which takes an input and a hidden state, and generates an output and a new hidden state. The hidden state functions as the network's memory, enabling it to retain information from previous inputs. The output from the recurrent neuron can be reintroduced into the network as input for the subsequent time step, allowing the processing of input sequences. However, a major challenge in training RNNs is the vanishing gradient problem, where the gradients utilized to update network weights diminish considerably as they propagate backward in time. (Hewamalage et al., 2019). This can make it difficult for the network to learn long-term dependencies in the data. Numerous versions of recurrent neural networks (RNNs) have been created to tackle this issue. Among them are long short-term memory (LSTM) networks and gated recurrent units (GRUs), both of which incorporate distinct gating mechanisms that regulate the information flow throughout the network. Overall, RNNs are a powerful tool for processing sequential data and have been used successfully in a wide range of applications (Joseph, 2022). However, they can be computationally expensive to train and may require careful tuning of hyper parameters to achieve good performance. The general structure of a feedforward neural network is shown in Figure. One can provide a feedback path for neurons in the hidden layer to add the memory capability to the network to reach RNNs (Figure).

**General Architecture of a Feedforward Neural Network**

Ein Bild, das Screenshot, Text, Kreis enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

To fully understand how feedback structure works, look at the hidden layers unfolding by time in in the figure below.

**Recurrent Neural Network**

Ein Bild, das Diagramm, Screenshot, Reihe, Electric Blue (Farbe) enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

In the figure below, 3 time steps of hidden layers’ evolution has been depicted. As it is obvious, the temporal input data (x) change the behavior of the hidden layers and the output respectively.

**Unfolding the hidden layers for 3 time steps**

Ein Bild, das Text, Schrift, Reihe, Diagramm enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

a(t) and y(t) can be expressed as the following equations.

|  |  |
| --- | --- |
|  | (Eq.1) |
|  | (Eq.2) |

W’s are the weights that will be determined by training. g1 and g2 are the activation functions.

Depending on the number of x and y there are 5 types of RNN that are listed in table

|  |  |
| --- | --- |
| Number of X and Y | Type of RNN |
| N(X)=N(Y)=1 | One-to-One |
| N(X)=1 , N(Y)>1 | One-to-Many |
| N(X)>1 , N(Y)=1 | Many-to-One |
| N(X)=N(Y)>1 | Many-to-Many |
| N(X)<>N(Y) | Many-to-Many |

Source: [Mojtaba Nabipour], (2023).

The loss function of an RNN network is the summation of all time step loss functions.

|  |  |
| --- | --- |
|  | (Eq.3) |

The back propagation is done at each point in time. The derivative of the loss function respect to the weight matrix is:

|  |  |
| --- | --- |
|  | (Eq.4) |

Ordinary recurrent neural networks have two shortcomings of

* Vanishing/exploding gradient problems
* Gradient clipping

The vanishing gradient issue in recurrent neural networks (RNNs) refers to the problem of the gradients becoming extremely small as they propagate through the network during backpropagation. The reason behind this phenomenon is that, in every time step, the gradients are subject to the weights of the network, and when these weights are of small value, the gradients will also reduce to a small scale. Consequently, the network can encounter challenges in acquiring long-term dependencies since the insufficient gradients cannot efficiently update the weights. This drawback can cause inadequate execution and a delayed convergence rate in RNNs (Hebbar et al., 2022).

Similarly, if the gradient update is too much it leads to unstable and inaccurate predictions. This issue is particularly common in RNNs because they have feedback loops that can amplify the gradients over time, leading to an exponential increase in their magnitude. The exploding gradient problem can be addressed by using gradient clipping, which limits the maximum size of the gradients, or by using more stable RNN architectures such as the LSTM or GRU.

Gradient Clipping

Gradient clipping is a technique to overcome the exploding gradient problem by preventing the gradient from exceeding a specified limit. Figure illustrates the technique using a graph.

**Gradient Clip Illustration**

Ein Bild, das Text, Reihe, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Some types of RNNs are designed to remedy the vanishing gradient problem in a specific way by using specific gates

These gate is used to build LSTM networks cells and will be described in the following subsections in details.

Long-Short Term Memory (LSTM) networks

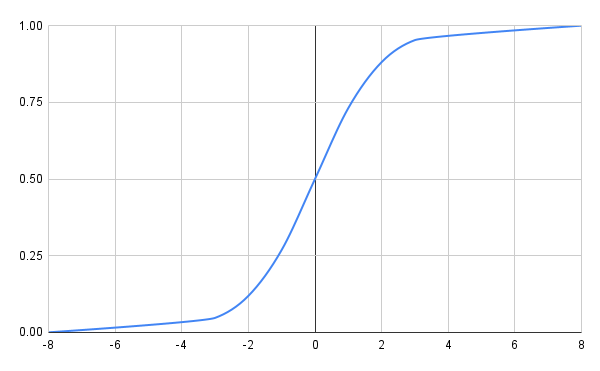
LSTM neural networks are a type of recurrent neural networks designed to handle the challenge of handling long dependencies in sequential data. They prove to be highly beneficial in various applications, including time-series forecasting, speech recognition, language translation, and image captioning. These tasks involve input data that consists of a sequence of words or images (Nithin et al., 2022). LSTM networks possess a crucial characteristic which enables them to selectively retain or discard information from previous time steps. This capability is achieved by utilizing "memory cells" that store information over time, along with "gates" that regulate the flow of information in and out of these cells. These gates are implemented as sigmoid functions, which consider the current input and the previous hidden state as input and produce a value between 0 and 1. This value determines the extent to which the input and previous state should be transmitted to the subsequent time step. The forget gate decides the amount of the previous memory cell to retain, while the input gate determines how much of the current input should be added to the memory cell. Lastly, the output gate determines the portion of the current memory cell that should be passed on to the next time step (Abbasimehr et al., 2020). LSTM networks possess the ability to preserve important information from past time steps while discarding irrelevant details, allowing them to handle long-term dependencies effectively and overcome the "vanishing gradient" issue commonly encountered in conventional RNNs. Consequently, LSTM networks are highly advantageous for sequential data modeling and consistently achieve cutting-edge performance across various tasks. Prior to delving into the cell structure, it is beneficial to revisit the sigmoid and TanH functions. These mathematical functions, namely the sigmoid and hyperbolic tangent functions, are extensively employed in machine learning and artificial intelligence algorithms.

The sigmoid function is a type of activation function that is used in neural networks to introduce non-linearity into the output of a neuron. It is defined as:

|  |  |
| --- | --- |
|  | (Eq.5) |

The value of x serves as the input for the function. The sigmoid function transforms any input value into a range between 0 and 1, making it beneficial for binary classification tasks. As the input to the function increases, the output gets closer to 1, while for smaller inputs, the output approaches 0. A visual representation of the function can be seen in Figure..

**Sigmoid Graph**



Source: [Mojtaba Nabipour], (2023).

The tangent hyperbolic function, on the other hand, is another type of activation function that is used in neural networks. It is defined as:

|  |  |
| --- | --- |
|  | (Eq.6) |

The hyperbolic tangent function transforms input values into a range of -1 to 1, serving as a valuable tool for regression tasks. As the input increases, the output gradually reaches 1, while for small inputs, the output gradually approaches -1. Both the sigmoid and hyperbolic tangent functions are frequently employed in machine learning algorithms to introduce non-linear behavior in neuron outputs. This non-linearity enables neural networks to grasp intricate patterns within the data. The figure showcases the graph of the hyperbolic tangent function.

**Graph of TANH**

Ein Bild, das Reihe, Diagramm, parallel enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

LSTM Cell Structure

It has the ability to learn patterns over long sequences of data by selectively retaining or discarding information. The functioning of an LSTM cell can be divided into four key components: the input gate, the forget gate, the output gate, and the cell state.

1. **Input Gate:** The input gate is responsible for identifying the relevant information from the current input and previous hidden state that should be added to the cell state. By utilizing a sigmoid activation function, the input gate calculates a value between 0 and 1 for each element of the input, determining the significance of incorporating the corresponding information into the cell state.
2. **Forget Gate:** The forget gate determines which information from the previous cell state should be retained or discarded. It takes the current input and previous hidden state as inputs and applies a sigmoid activation function, resulting in a value ranging from 0 to 1 for each element in the cell state. This value signifies the degree to which each piece of information should be preserved or forgotten.
3. **Cell State:** The cell state serves as the memory of the LSTM cell. It stores information from previous time steps and updates it based on the input gate and forget gate. The input gate determines the information to be added to the cell state, while the forget gate determines the information to be retained or forgotten. The cell state is updated by multiplying the previous cell state by the forget gate and adding the input gate multiplied by the current input.
4. **Output Gate:** The output gate determines which information within the current cell state is suitable for use as the hidden state. It takes inputs from the current input and previous hidden state and applies a sigmoid activation function to each element in the cell state, resulting in a value between 0 and 1. This value indicates the suitability of the data for use as the hidden state. Then, the cell state goes through a tanh activation function to generate the actual hidden state, which is subsequently multiplied by the output gate to produce the final output.

**Structure of LSTM Cell**

Ein Bild, das Text, Diagramm, Screenshot, Plan enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Overall, the LSTM cell selectively remembers or forgets information based on the input and previous state, and updates its memory accordingly (Brownlee, 2018). This allows it to handle long-term dependencies in sequential data and avoid the vanishing gradient problem in traditional RNNs. Figure demonstrates the details of an LSTM cell consisting mentioned gates where C is the cell state and h is the hidden state.

**Implementation**:

To implement an LSTM network we use python with Keras library. The steps can be summarized as follows:

**Step 1: Import the necessary libraries**

mport numpy as np

*import pandas as pd*

*from keras.models import Sequential*

*from keras.layers import Dense, LSTM*

**Step 2: Load the dataset**

*data = pd.read\_csv('your\_dataset.csv')*

**Step 3: Preprocess the data**

*Split the data into training and testing sets*

*train\_data = data[:800]*

*test\_data = data[800:]*

*# Normalize the data*

*from sklearn.preprocessing import MinMaxScaler*

*scaler = MinMaxScaler(feature\_range=(0, 1))*

*train\_data\_scaled = scaler.fit\_transform(train\_data)*

*test\_data\_scaled = scaler.transform(test\_data)*

*# Create the input and output sequences*

*def create\_sequences(data, seq\_length):*

*X = []*

*y = []*

*for i in range(len(data)-seq\_length-1):*

*X.append(data[i:(i+seq\_length), 0])*

*y.append(data[i+seq\_length, 0])*

*return np.array(X), np.array(y)*

*seq\_length = 10*

*X\_train, y\_train = create\_sequences(train\_data\_scaled, seq\_length)*

*X\_test, y\_test = create\_sequences(test\_data\_scaled, seq\_length)*

*# Reshape the input sequences*

*X\_train = np.reshape(X\_train, (X\_train.shape[0], X\_train.shape[1], 1))*

*X\_test = np.reshape(X\_test, (X\_test.shape[0], X\_test.shape[1], 1))*

**Step 4: Build the LSTM model**

*model = Sequential ()*

*model.add(LSTM(units=50, return\_sequences=True, input\_shape=(X\_train.shape[1], 1)))  
model.add(LSTM(units=50))  
model.add(Dense(units=1))  
model.compile(optimizer='adam', loss='mean\_squared\_error')*

**example 1:**

We have a monthly demand history data saved in a CSV file. An LSTM model has been implemented using Python. The steps with coding in COLAB will be described in the following.

The most difficult part is basically how to reformat the data correctly to feed the model.

First we read the dataset and store it in an array.

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

The plot of the data is demonstrated in Figure.

**The Plot of Monthly Demand Dataset**

Ein Bild, das Text, Screenshot, Reihe, Diagramm enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

It looks like the time series has the seasonality and trend pattern. We can decompose them by following code.

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

The decomposition results include seasonality, trend and noise patterns are shown in Figure.

**Results of Demand Dataset Decomposition**

Ein Bild, das Text, Reihe, Schrift, Diagramm enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Splitting the data set into train and test sets is the next step, as shown in the code below.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

The range of numbers should be limited to get better results. To do that, we use MinMaxScaler function form preprocessing library of sklearn package.

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Schrift, weiß, Screenshot enthält.

Automatisch generierte Beschreibung

Now the data value will be set between zero and one.

The next following code lines reformat the training set data into batches ready to feed the model.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

We use batches of 12 months’ demand data to train the model. The output of the model is the prediction of the next following month.

The next step is defining the model and is relatively straightforward.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

The created network model can be visualized using following code

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

The network diagram is depicted in Figure

**The Network Created by the Model**

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

Training of the model is done using *fit* method in 50 epochs

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

The loss graph illustrates how well the training has been gone

**The Loss Graph of the Training Procedure**

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

**The Predicted vs Realized Demand**

Ein Bild, das Diagramm, Reihe, Steigung enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

To test the model, the last 12 values of the training dataset as a starting batch is selected to predict the test dataset.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

After making the predictions the scaled values must be back to their originals

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung



The prediction plot is illustrated in Figure.

LSTM is better suited for complex data that has non-linear relationships and long-term dependencies. Considering computational resources, statistical methods are better suited for simpler data that has linear relationships and short-term dependencies.

Self-check questions:

1. How do RNNs handle sequential data, and what advantages does this give them over other models?

***Answer****:* *by sequentially analyzing individual data points and retaining an internal state that preserves previous information, RNNs effectively generate predictions for the present data point. Updating the internal state at each time step enables RNNs to grasp both temporal and long-term dependencies within the input sequence. In contrast, conventional feedforward neural networks and other machine learning models lack the capacity to capture these dependencies, rendering them inappropriate for sequential data.*

2. What is the main advantage of LSTM networks over traditional RNNs?

***Answer:*** *The main advantage of Long Short-Term Memory (LSTM) networks over traditional Recurrent Neural Networks (RNNs) is their ability to handle the vanishing gradient problem which is a common issue in training RNNs.*

3. How does an LSTM network handle vanishing gradients, which can be a challenge in training RNNs?

***Answer:*** *LSTM networks solve this problem by introducing a memory cell and three gating mechanisms: the input gate, forget gate, and output gate. The memory cell is designed to store information about the previous inputs and can control the flow of information through the gates.*

4. What are the main components of an LSTM cell, and how do they work together to process sequential data?

***Answer:***

1. *Memory Cell: The memory cell is the main component of the LSTM cell and is responsible for storing and updating information about the input sequence over time. It acts as a conveyor belt that passes information from one time step to the next.*
2. *Input Gate: The input gate controls the flow of information into the memory cell. It is a sigmoid function that takes the current input and the previous hidden state as input and decides how much new information should be added to the memory cell.*
3. *Forget Gate: The forget gate controls the flow of information out of the memory cell. It is a sigmoid function that takes the current input and the previous hidden state as input and decides how much old information should be removed from the memory cell.*
4. *Output Gate: The output gate controls the flow of information from the memory cell to the next hidden state. It is a sigmoid function that takes the current input and the previous hidden state as input and decides how much information from the memory cell should be output to the next hidden state.*

*The LSTM cell processes sequential data as follows:*

1. *At each time step, the current input and the previous hidden state are passed to the input gate and forget gate, respectively.*
2. *The input gate decides how much new information should be added to the memory cell, while the forget gate decides how much old information should be removed from the memory cell.*
3. *The memory cell updates its state based on the input from the input gate and the forget gate, and stores the new information.*
4. *The output gate decides how much information from the memory cell should be output to the next hidden state.*
5. *The updated hidden state is returned as the output of the LSTM cell, and is also used as input to the next time step.*

5. How can the forget gate in an LSTM network be adjusted to improve its performance on a given task?

***Answer:*** *The forget gate in an LSTM (Long Short-Term Memory) network controls the flow of information out of the memory cell, deciding how much old information should be removed from the cell. By adjusting the forget gate, we can improve the network's performance on a given task. Here are some ways to adjust the forget gate:*

1. *Change the Activation Function: The forget gate is typically implemented as a sigmoid function that takes the input, current hidden state, and previous cell state as input. By changing the activation function, we can modify the behavior of the forget gate. For example, using a ReLU (Rectified Linear Unit) activation function can improve the performance of the network on tasks that require non-linear transformations.*
2. *Use Different Initialization Methods: The forget gate's weights are typically initialized randomly, which can sometimes lead to poor performance. Changing the initialization method to something like Xavier initialization or He initialization can improve the network's performance by ensuring that the weights are initialized near the optimal value.*
3. *Add Regularization: Regularization techniques like dropout or weight decay can be used to prevent overfitting and improve the network's performance. By applying regularization to the forget gate, we can prevent it from overfitting to the training data and improve its generalization ability.*
4. *Adjust the Learning Rate: The learning rate controls the step size of the gradient descent algorithm during training. By adjusting the learning rate, we can control how much the weights of the network are updated during each iteration of training. If the forget gate is not learning properly, we can adjust the learning rate to encourage more or less learning.*
5. *Change the Number of Units: The number of units in the forget gate can be adjusted to change the size of the network. By increasing or decreasing the number of units, we can control the complexity of the network and improve its performance on a given task.*

3.2 Supervised Learning

Supervised learning is a type of machine learning algorithm where the computer is trained on a labeled dataset, meaning that the data is already categorized or classified. The algorithm learns to recognize patterns in the input data and maps those patterns to the correct output or label. The goal of supervised learning is to train a model that can accurately predict the correct output for new, unseen data. In supervised learning, the input data is often referred to as the "features" or "predictors," while the output data is referred to as the "target" or "label." The labeled dataset is split into a training set, which is used to train the model, and a testing set, which is used to evaluate the model's performance.

There are two main types of supervised learning algorithms: regression and classification. Regression algorithms are used when the output is a continuous variable, such as predicting the price of a house based on its features. Classification algorithms are used when the output is a categorical variable, such as predicting whether an email is spam or not based on its content.

Supervised learning has many applications, including image recognition, speech recognition, natural language processing, and fraud detection. It is a powerful tool for solving complex problems and making accurate predictions based on data. Supervised learning algorithms can be used for time series forecasting as well. One frequently employed method for time series forecasting with supervised learning involves utilizing regression algorithms, like linear regression or neural networks (discussed in the preceding section). The algorithm learns from a labeled dataset comprising past variable values and their corresponding future values. The input data consists of a sequence of past values, while the output data represents the predicted future value.

Another approach entails employing classification algorithms such as decision trees or random forests. In this scenario, the output data is a categorical variable that signifies whether the future value will surpass or fall below a specific threshold. The algorithm is trained on a labeled dataset of past values and their corresponding binary labels, and it learns to classify new data based on the input sequence. Time series forecasting using supervised learning has many applications, including predicting stock prices, weather forecasting, and energy demand forecasting. It can help businesses make informed decisions based on future trends and improve their operations by optimizing resource allocation and inventory management.

To do demand forecasting using supervised learning, we need to follow these steps:

* **Collect and preprocess historical demand data:** We need to collect and preprocess historical demand data, which includes the number of units sold or the revenue generated for a product or service over a period of time. Preprocessing may involve cleaning the data, handling missing values, and transforming the data if necessary.
* **Split the data into training and validation sets:** We need to split the historical demand data into training and validation sets. The training set will be used to train the model, while the validation set will be used to evaluate the model's performance.
* **Choose a suitable model:** We need to choose a suitable model for demand forecasting. Recurrent neural networks (RNNs) such as long short-term memory (LSTM) networks are commonly used for time series forecasting tasks.
* Train the model: We need to train the model on the training set using an appropriate loss function such as mean squared error (MSE) or mean absolute error (MAE). The model should learn to map past demand data to future demand data.
* **Evaluate the model:** To assess the effectiveness of the model, it is necessary to analyze its performance on the validation set using evaluation measures like MSE or MAE. If the model's performance falls short of expectations, it may be necessary to modify the model's hyperparameters or experiment with alternative architectural designs.
* **Make predictions:** Once the model is trained and validated, we can use it to make predictions on new, unseen demand data. Overall, demand forecasting using supervised learning involves collecting and preprocessing historical demand data, splitting the data into training and validation sets, choosing a suitable model, training the model, evaluating the model's performance, and making predictions on new data.

In the following sections two major regression-based methods will be explained.

Support Vector Regression

Support vector regression (SVR) is a supervised learning technique employed for regression tasks. Its objective is to determine a function that can estimate the continuous value of a target variable based on one or more input features. The underlying concept of SVR involves identifying a hyperplane within the feature space that optimizes the distance between predicted and actual values. Essentially, SVR aims to discover a line that maximizes the separation from data points while still capturing the overall data trend. To accomplish this, SVR employs a kernel function to transform input features into a higher-dimensional space, facilitating the identification of a hyperplane that can effectively separate data points. The algorithm then strives to locate the hyperplane that maximizes the margin between predicted and actual values.

In support vector regression (SVR), a kernel is a function that maps the input features into a higher-dimensional space, where it becomes easier to find a hyperplane that separates the predicted values from the actual values. The kernel function essentially transforms the input data into a new representation that may be more suitable for linear separation. SVR employs several kernel functions, including the linear kernel, polynomial kernel, and radial basis function (RBF) kernel, which are frequently utilized. The selection of a kernel function is influenced by the characteristics of the data and the particular problem being addressed. The kernel trick is a powerful technique that allows the computation of the hyperplane in the high-dimensional feature space without actually computing the coordinates of the data points in that space. This makes SVR computationally efficient even for large datasets.

The SVR distinguishes itself from other regression models by aiming to find the optimal line within a predefined threshold, rather than minimizing the discrepancy between actual and predicted values. (Zhang & O'Donnell, 2020). Hyperplanes and boundary lines are separated by threshold values. The complexity of SVR fits increases more than quadratically with sample number. Scaling to datasets with more than 10000 samples is challenging due to this limitation.

The Python implementation steps of SVR are listed below.

First, we import the necessary libraries:

*import pandas as pd*

*import numpy as np*

*from sklearn.svm import SVR*

*from sklearn.metrics import mean\_squared\_error*

Next, we load the time series data into a Pandas DataFrame and split it into training and testing sets:

*# Load data*

*data = pd.read\_csv('time\_series\_data.csv', index\_col='date')*

*# Split data into train and test sets*

*train\_data = data.iloc[:100]*

*test\_data = data.iloc[100:]*

Then, we prepare the data by creating lag features. We create a function that takes a time series and a lag value as inputs and returns a DataFrame with the original time series and its lagged values:

*def create\_lag\_features(data, lag):*

*df = pd.DataFrame(index=data.index)*

*df['y'] = data['y']*

*for i in range(1, lag+1):*

*df[f'lag\_(Hewamalage et al.)'] = data['y'].shift(i)*

*df.dropna(inplace=True)*

*return df*

We apply this function to our training and testing sets to create lag features:

*# Create lag features for training set*

*train\_df = create\_lag\_features(train\_data, lag=3)*

*# Create lag features for testing set*

*test\_df = create\_lag\_features(test\_data, lag=3)*

Next, we split the lagged time series into X and y variables for training and testing:

*# Split lagged time series into X and y variables for training set*

*X\_train = train\_df.drop('y', axis=1) y\_train = train\_df['y']*

*# Split lagged time series into X and y variables for testing set*

*X\_test = test\_df.drop('y', axis=1) y\_test = test\_df['y']*

We then fit an SVR model to the training data:

*# Fit SVR model to training data*

*svr = SVR(kernel='linear', C=1e3, epsilon=0.1)*

[*svr.fit*](https://svr.fit)*(X\_train, y\_train)*

Finally, we use the trained SVR model to make predictions on the testing data and calculate the mean squared error:

*# Make predictions on testing data*

*y\_pred = svr.predict(X\_test)*

*# Calculate mean squared error*

*mse = mean\_squared\_error(y\_test, y\_pred)*

*print(f'MSE: {mse:.2f}')*

Some advantages and disadvantages of SVR are listed below.

**Pros of SVR:**

* Can handle non-linear relationships between input and output variables.
* Can handle high-dimensional data.
* Can handle outliers in the data.
* Can be used for both regression and classification tasks.
* Can be trained with different kernel functions to handle different types of data.

**Cons of SVR:**

* Requires careful selection of hyperparameters, such as the kernel function and regularization parameter.
* Can be sensitive to the choice of kernel function and hyperparameters, which can lead to overfitting or underfitting.
* Can be computationally expensive for large datasets or complex kernel functions.
* Requires a large amount of data to train effectively.
* May not perform well with noisy or sparse data.
* the accuracy of the model is not maintained for large datasets.

Self-check questions:

1. What is the purpose of supervised learning in machine learning?

***Answer****: Supervised learning is a machine learning technique that involves training a model to make predictions based on input and output pairs, where the output is a known label or target value that corresponds to the input. The primary objective of supervised learning is to learn a mapping between the input and output variables, so that the model can accurately predict the output for new, unseen input data. The labeled dataset is used to teach the algorithm how to make predictions through the development of a mapping function that can predict the output for new input data.*

2. How does the process of training a supervised learning model differ from testing it?

***Answer****: To summarize, supervised learning model training involves teaching the model to make precise predictions based on labeled examples, whereas testing the model includes assessing its performance on new and unseen data. Differences between the two processes include variations in data, objectives, evaluation metrics, procedures, and the risk of overfitting.*

3. What is the main difference between support vector regression and other regression techniques?

***Answer****: To summarize, Support Vector Regression (SVR) differs from other regression techniques in two main ways: firstly, it handles outliers by only considering the support vectors, which are the data points closest to the decision boundary. Secondly, it allows for highly non-linear decision boundaries through the use of kernel functions, which can capture complex shapes that other regression techniques may not be able to handle.*

4. How does support vector regression handle non-linear relationships and outliers in the data?

***Answer****: Support Vector Regression addresses non-linear relationships and outliers in the data through kernel functions that transform the input space into a higher dimensional feature space, by considering support vectors that are the data points closest to the decision boundary, and by tuning parameters such as the regularization parameter and kernel function to optimize the model's performance. These features make SVR a versatile and robust regression technique suitable for various applications.*

5. What techniques can be used to determine the optimal value for the regularization parameter in support vector regression?

***Answer****: finding the optimal value for the regularization parameter in Support Vector Regression (SVR) can be achieved through various techniques such as grid search, random search, cross-validation, or Bayesian optimization. The selection of the optimal approach depends on factors such as the size of the dataset, available computational resources, and the desired level of accuracy.*

3.3 Effects of Correlation and Confounding

Correlation pertains to the connection between two variables, wherein alterations in one variable are linked to changes in the other variable. When it comes to time series forecasting, correlation can aid in recognizing patterns and trends within the data, enabling more precise predictions. Nonetheless, correlation can also result in misleading associations, where two variables seem to be connected, but the relationship lacks causality. If this relationship is not comprehended accurately, it can lead to inaccurate forecasts.

**Confounding**

This refers to a third variable that affects the relationship between two other variables.

In time series forecasting, **confounding** can lead to inaccurate predictions if the effect of the confounding variable is not properly accounted for. For example, if a company's sales are forecasted based on historical data, but a major competitor enters the market during the forecast period, the competitor's impact on sales must be considered to make accurate predictions. To mitigate the impact of correlation and confounding on data-driven forecasting of time series, it is important to carefully analyze the data and identify any potential relationships or confounding variables. This can involve using statistical techniques such as regression analysis or time series decomposition to identify patterns and trends in the data. Additionally, it is important to consider external factors that may impact the forecast, such as changes in the market or industry. By properly accounting for correlation and confounding, data-driven forecasting of time series can be more accurate and reliable.

Statically analyzing the correlation and confounding effect in a time series before forecasting is an important step in ensuring the accuracy of the forecast. Here are the steps to follow:

* **Collect the data:** Collect the time series data that you want to analyze. This data should include the dependent variable (the variable you want to forecast) and any independent variables that may have an effect on the dependent variable.
* **Plot the data:** Plot the time series data to visualize the trends and patterns in the data. This will help you identify any outliers or anomalies in the data.
* **Calculate the correlation:** Calculate the correlation between the dependent variable and each independent variable. This will help you identify any variables that are strongly correlated with the dependent variable.
* **Check for confounding effects:** Ensure there are no confounding factors among the independent variables. Confounding factors arise when two independent variables are associated with each other, creating challenges in identifying the variable responsible for the alteration in the dependent variable.
* **Use regression analysis:** Use regression analysis to determine the relationship between the dependent variable and each independent variable. This will help you identify any significant predictors of the dependent variable.

For example, you want to forecast the sales of a particular product over the next year. You have collected data on the sales of the product over the past five years, as well as data on the price of the product, the marketing spending, and the seasonality of the product.

To statically analyze the correlation and confounding effect in this time series, you would first plot the data to visualize any trends or patterns. You may notice that sales tend to increase during certain months of the year, indicating seasonality. Next, you would calculate the correlation between sales and each independent variable (price, marketing spending, and seasonality). You may find that sales are negatively correlated with price, positively correlated with marketing spend, and strongly correlated with seasonality.

You would then check for any confounding effects between the independent variables. For example, you may find that marketing spend and seasonality are strongly correlated, which could make it difficult to determine which variable is causing the change in sales.

Finally, you would use regression analysis to determine the relationship between sales and each independent variable. This would help you identify the significant predictors of sales and use them to forecast future sales.

**Example:**

We have an inventory cost for 19 months for 3 products. Determine the most significant predictors.

This could be done using regression analysis.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Month | Inventory cost | Product A | Product B | Product C |
| 1 | 44439 | 515 | 541 | 928 |
| 2 | 43936 | 929 | 692 | 711 |
| 3 | 44464 | 800 | 710 | 824 |
| 4 | 41533 | 979 | 675 | 758 |
| 5 | 46343 | 1165 | 1147 | 635 |
| 6 | 44922 | 651 | 939 | 901 |
| 7 | 43203 | 847 | 755 | 580 |
| 8 | 43000 | 942 | 908 | 589 |
| 9 | 40967 | 630 | 738 | 682 |
| 10 | 78582 | 1113 | 1175 | 1050 |
| 11 | 45003 | 1086 | 1075 | 984 |
| 12 | 44303 | 843 | 640 | 828 |
| 13 | 42070 | 500 | 752 | 708 |
| 14 | 44353 | 813 | 989 | 804 |
| 15 | 45968 | 1190 | 823 | 904 |
| 16 | 47781 | 1200 | 1108 | 1120 |
| 17 | 43202 | 731 | 590 | 1065 |
| 18 | 44074 | 1089 | 607 | 1132 |
| 19 | 44610 | 786 | 513 | 839 |

Source: [Mojtaba Nabipour], (2023).

We use MS excel to do the analysis (Data analysis toolpack in data tab)

The dependent variable or Y will be inventory cost and the independent variables or Y will be three products columns. The summary of analysis is shown in Figure.

**Linear Regression Analysis using Excel**

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Source: [Mojtaba Nabipour], (2023).

We always need to inspect the P-values to determine the most significant predictors. If the P-value of a predictor is greater than 0.15, we can exclude the variable from the regression since it has not significance impact on the prediction of the outcome. In this case the product A P-value is greater than 0.15 so we can exclude it and run the analysis again for regression-based prediction (Figure). The linear regression as the name suggests looking for a linear relationship between independent variables and the dependent variable. The formula will be

(Eq.7)

To predict the outcome or Y using the linear formula, as the regression analysis shows, the constant value can be substitutes with intercept value, the a1 with product A coefficient, the a2 with product B coefficient and a3 with product C coefficient. But as the P-value column suggests we can eliminate the x1 variable from the equation.

**Regression Analysis after refining**

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Self-check questions

1. in previous example assuming the following values for the products, what is the predicted outcome?

Product A= 1200

Product B= 800

Product C= 1000

2. What is the difference between confounding and correlation in the context of supervised learning?

***Answer****: To sum up, in supervised learning, correlation and confounding are two distinct concepts. Correlation refers to a statistical relationship between variables, whereas confounding refers to a situation where the effect of one variable is mixed with the effect of another variable, which can impact the accuracy of predictive models. To overcome confounding variables, one can use different strategies, such as controlling for confounding variables, stratifying the data, or using techniques like propensity score matching or instrumental variable analysis.*

3. Can a high degree of correlation between predictor variables lead to overfitting in a supervised learning model?

***Answer****: a high degree of correlation between predictor variables can cause overfitting in a supervised learning model by providing redundant information that can lead to unstable estimates, inflated standard errors, and reduced statistical power.*

4. What steps can be taken to address issues related to correlation and confounding in supervised learning?

***Answer***: *To mitigate this problem, techniques such as feature selection, regularization, or dimensionality reduction can be used to reduce the complexity of the model and improve its generalizability.*

5. How can you determine the significance of the relationship between the predictor variable and the response variable in a regression model?

***Answer****: In order to assess the importance of the connection between the predictor variable and the response variable in a regression model, various methods can be employed. These include calculating the correlation coefficient, conducting hypothesis testing, evaluating the coefficient of determination, and examining the confidence interval of the regression coefficient. These techniques are useful for determining both the strength and statistical significance of the relationship between the predictor and response variables.*

3.4 Big-Data Newsvendor Model

The big data newsvendor model is a statistical approach employed in supply chain management to enhance inventory management and reduce expenses. It is founded on the well-known newsvendor problem, an optimization challenge within operations research. In the traditional newsvendor problem, a newsstand proprietor must determine the appropriate daily newspaper order quantity to maximize profits. Although the actual demand for newspapers is uncertain, the owner can make an estimation using historical data. The objective of the newsvendor problem is to identify the order quantity that yields the highest expected profits, considering the uncertain demand.

The big data newsvendor model extends this problem to incorporate additional sources of data, such as weather data, social media data, and sales data from other retailers. By incorporating these additional sources of data, the model can more accurately predict demand and optimize inventory levels.

For example, a retailer might use weather data to predict how much ice cream will sell on a hot day, or use social media data to predict which products are likely to be popular. By incorporating these additional sources of data into the newsvendor model, the retailer can optimize inventory levels and minimize costs. Overall, the big data newsvendor model is a powerful tool for supply chain management that leverages statistical models and machine learning algorithms to optimize inventory levels and minimize costs. In this section we review two recent study on this matter.

In (Ban & Rudin, 2019), the researchers employ contemporary machine learning methods, including algorithmic stability theory and quantile regression, to examine the newsvendor dilemma in situations where the decision-maker possesses both historical demand data and a substantial set of demand-related features.They apply these techniques to develop a sample average approximation algorithm that can handle big data and provide practical insights for decision makers. The authors use algorithmic stability theory and quantile regression, which are both techniques from machine learning, to analyze the newsvendor problem. They apply these techniques to develop a sample average approximation algorithm that can handle big data. The authors suggest two methods for determining the best order quantity using a dataset that includes both demand and related feature observations. One method involves applying empirical risk minimization (ERM) with or without regularization, a machine-learning principle. The second method, known as kernel-weights optimization (KO), draws inspiration from the Nadaraya-Watson kernel regression technique. These approaches are discussed in more detail in Section 2 of the article on page 3. The authors apply these approaches to develop a sample average approximation algorithm that can handle big data and provide practical insights for decision makers. The past works that the authors compare and contrast their algorithms to are based on incorporating exogenous information in inventory decision making. Specifically, they compare their work to (Liyanage & Shanthikumar, 2005) who use a Bayesian approach to incorporate demand forecasts and other information into the newsvendor problem. This is discussed in more detail in Section 2.4.1 of the article on page 7.

In another study (Huber et al., 2019), the researchers utilize machine learning to establish the ideal order quantity through the application of a data-oriented technique to solve the newsvendor problem. They classify inventory management data-driven approaches into three levels, and in the second level, they employ a normal distribution, differentiating it from the data-driven approach. To optimize the process, they select a specific quantile of the corresponding demand distribution. In the third level, they utilize a parametric QR model, such as ANNs, and estimate its parameters by resolving problem (8) instead of minimizing the MSE. The outcome of this procedure is the determined order quantity.

The writers contrast their technique against established methods across three levels of data-driven approaches in inventory management. They demonstrate that data-driven methods surpass model-based ones in most instances, using their real-world dataset. They also examine the empirical impact of choices made at each level on performance. The current literature lacks clarity on how these choices influence performance.

The authors in (Clausen & Li, 2022) the researchers utilized machine learning techniques to identify the most suitable quantity of orders in a dynamic inventory model that uses the Empirical Risk Minimization (ERM) principle. They developed a machine learning algorithm to address the problem and carried out an empirical investigation using genuine business data to showcase the efficiency of their inventory model and solution algorithm driven by big data. The numerical results of the study reveal that their comprehensive big data-driven model can yield cost reductions of up to 60% compared to the most successful single-variable benchmark model and up to 6.37% cost savings compared to the most successful big data-driven benchmark model.

Overall, these papers demonstrate the potential of machine learning and data-driven approaches in improving inventory management decisions. By leveraging large amounts of data and advanced algorithms, these approaches can provide more accurate demand forecasts, optimize inventory levels, and reduce costs. However, there are still challenges to be addressed, such as the need for high-quality data and the difficulty of interpreting complex machine learning models. In conclusion, these papers highlight the importance of incorporating machine learning and data-driven approaches into inventory management research and practice. As technology continues to advance, we can expect to see more innovative solutions that leverage these approaches to improve supply chain efficiency and effectiveness.

Self-check Questions

1. How can machine learning algorithms be used to improve newsvendor decision-making?

***Answer****: machine learning algorithms can be utilized to enhance newsvendor decision-making by improving demand forecasting, inventory optimization, dynamic pricing, and customer segmentation. By leveraging these techniques, newsvendors can improve their competitiveness, profitability, and customer satisfaction.*

2. What are some limitations of using big data in newsvendor analysis?

***Answer****: while big data can offer new avenues for improving newsvendor analysis, its utilization is not without limitations. These include factors such as data quality, volume, privacy, interpretability, and cost. Addressing these limitations requires careful planning, expertise, and adherence to legal and ethical standards.*

3. What are the main challenges faced by newsvendors in big data?

***Answer****: newsvendors face several challenges when working with big data, including data integration, quality, privacy, analysis, infrastructure, and cost. Addressing these challenges requires careful planning, expertise, and investment in technology and resources.*

**Summary**

Different approaches to demand forecasting, such as recurrent neural networks, supervised learning, correlation and confounding effect analysis, and the big data newsvendor model, each have their own set of advantages and disadvantages. Recurrent neural networks are effective in capturing intricate patterns in time-series data, but they necessitate a substantial amount of data and can be computationally intensive. Supervised learning is a straightforward and efficient method for demand forecasting, but it assumes that future demand will resemble past demand. Analyzing correlation and confounding effects can help identify the factors influencing demand, but it requires a deep understanding of the business and market. The big data newsvendor model can handle vast amounts of data and generate accurate demand forecasts, but it necessitates a significant investment in data infrastructure and analytical capabilities. In general, data-driven methods for demand forecasting offer valuable insights and enhance decision-making, but their strengths and limitations must be carefully considered. Choosing the appropriate approach depends on the specific needs of the business and the available data.

Abdel-Malek, L. L., & Areeratchakul, N. (2007). A quadratic programming approach to the multi-product newsvendor problem with side constraints. *European Journal of Operational Research*, *176*(3), 1607-1619.

Burgin, T., & Wild, A. (1967). Stock control—Experience and usable theory. *Journal of the Operational Research Society*, *18*(1), 35-52.

Choi, S., Ruszczyński, A., & Zhao, Y. (2011). A multiproduct risk-averse newsvendor with law-invariant coherent measures of risk. *Operations Research*, *59*(2), 346-364.

Chopra, S., Meindl, P., & Kalra, D. V. (2007). *Supply Chain Management by Pearson*. Pearson Education India.

Garvey, M. D., & Carnovale, S. (2020). The rippled newsvendor: A new inventory framework for modeling supply chain risk severity in the presence of risk propagation. *International Journal of Production Economics*, *228*, 107752.

Hadley, G., & Whitin, T. M. (1963). *Analysis of inventory systems*.

Kobbacy, K. A., & Liang, Y. (1999). Towards the development of an intelligent inventory management system. *Integrated Manufacturing Systems*, *10*(6), 354-366.

Libiseller, C., & Grimvall, A. (2002). Performance of partial Mann–Kendall tests for trend detection in the presence of covariates. *Environmetrics: The official journal of the International Environmetrics Society*, *13*(1), 71-84.

Matsuyama, K. (2006). The multi-period newsboy problem. *European Journal of Operational Research*, *171*(1), 170-188.

Ostertagova, E., Ostertag, O., & Kováč, J. (2014). Methodology and application of the Kruskal-Wallis test. Applied mechanics and materials,

Reid, R. D., & Sanders, N. R. (2023). *Operations management: an integrated approach*. John Wiley & Sons.

Ross, S. M. (2017). *Introductory statistics*. Academic Press.

Snyder, R. D. (1984). Inventory control with the gamma probability distribution. *European Journal of Operational Research*, *17*(3), 373-381.

Vandeput, N. (2020). De Gruyter. <https://doi.org/doi:10.1515/9783110673944>

Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2011). Probability and Statistics for Engineers and Scientists, 9th. In: Pearson.

Yao, L., Chen, Y. F., & Yan, H. (2006). The newsvendor problem with pricing: Extensions. *International Journal of Management Science and Engineering Management*, *1*(1), 3-16.