Time Series Forecasting and Dynamic Pricing for Cloud Usage

by

Donald Inyene Ekanem
B.Eng., Mechanical Engineering, University of Nigeria Nsukka, 2007

SUBMITTED TO THE PROGRAM IN SUPPLY CHAIN MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE IN SUPPLY CHAIN MANAGEMENT
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2023

Signature of Author: ____
Department of Supply Chain Management
May 12, 2023

Certified by: ____
Dr. Ilya Jackson
Postdoctoral Associate
Capstone Advisor

Accepted by: ____
Prof. Yossi Sheffi
Director, Center for Transportation and Logistics
Elisha Gray II Professor of Engineering Systems
Professor, Civil and Environmental Engineering

Time Series Forecasting and Dynamic Pricing for Cloud Usage

by

Donald Inyene Ekanem

Submitted to the Program in Supply Chain Management
on May 12, 2023 in Partial Fulfillment of the
Requirements for the Degree of Master of Applied Science in Supply Chain Management

ABSTRACT

This capstone explores different classical time series forecasting models to forecast cloud usage for an Infrastructure as a Service (IaaS) provider. The objective is to provide forecast information to help with capacity planning and propose a pricing model to optimize the capacity and manage revenue. The Mean Absolute Percentage Error (MAPE) performance criteria was compared for all candidate forecasting models to select the most suitable one. Analysis of the data showed a high linear trend in most of the zones, as well as a weekly seasonality. An elastic pricing model was proposed to influence customer demand behaviors to smoothen out capacity during the week. The conclusion is that the demand can be forecasted using a linear model with weekly seasonality. The determination of the most suitable forecasting model and prescribed elastic pricing model will help the sponsor company plan and manage capacity and revenue more optimally.

Advisor: Ilya Jackson

Title: Postdoctoral Associate

ACKNOWLEDGMENTS

I would like to express my utmost gratitude to my advisor, Dr. Ilya Jackson, for his patience, guidance and support throughout this project. He provided useful tips and access to necessary resources. His probing questions exposed insights and drove rigor in this work.

I would also like to appreciate my project sponsor, Pranav Prakash, for always being available to provide context, clarification and motivation to keep the project on track.

My sincere appreciation must also go to my loving wife, Chineze, and my kids, Jasmine, Jayden, and Janaya, whom I am yet to meet as at the time of writing. I owe you all greatly for being my emotional support despite the remarkable sacrifice.

To my writing coach, Toby Gooley. Your patience, empathetic communication, encouragement and open feedback made this writing experience a priceless one.

Finally, my classmates; the SCM blended class of 2023. You all made it worthwhile.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

In recent times, companies and organizations have elected to move away from making significant investments in computer hardware, software, and databases. The sheer volume and dynamic nature of data today, with the associated risks, necessitate moving this requirement to competent service providers. Cloud computing has afforded the opportunity to rent these services from infrastructure providers and pay only as they are used. Online access is offered to a wide range of computing resources and tools such as business applications, development tools, data storage, compute services, and networking solutions. These services are provided by a software vendor and can be hosted either at the vendor's data center or the customer's data center, and managed by the cloud services provider (Oracle Nigeria, 2022).

The provision to pay only as the service is used has enabled companies to free up capital funds which would otherwise have been tied up in Information Technology (IT) infrastructure, and focus on their core competencies, be it manufacturing, education, healthcare or consulting. It also saves them from the risk of obsolescence, as the IT space is a rapidly evolving one, enabling them to be flexible and scalable in supporting their IT solutions.

There are three main types of cloud services: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

- SaaS is an offering in which the service provider hosts the client’s software applications at their location, and the client accesses them over the internet.
- PaaS gives the client access to developer tools which they can use to build their own applications. They have ready-to-use programming units allowing client developers to build new features into their systems.

- IaaS allows customers to access infrastructure service as required, over the internet. The installation, configuration and maintenance of the software on the cloud is the responsibility of the client (Oracle Nigeria, 2022).

The Company is a cloud infrastructure provider furnishing IaaS offerings to numerous clients. When a client decides to “move to the cloud,” they are moving their IT infrastructure offsite to be managed by the company, which manages this at an offsite data center. Their data engineers provide the specifications of the hardware required for the workload. The company then assists the client to integrate data and stay abreast of market and industry demands by developing functionality and new capabilities, and providing regular system updates.

1.2 PROBLEM STATEMENT AND RESEARCH QUESTIONS

The capstone determines and recommends an appropriate forecasting method which will provide the basis for capacity investment. As well, the capstone proposes a demand elastic pricing model to smooth out demand based on seasonality observed. Historical data on usage of a particular product offering from the last three years in certain regions, has been obtained for this exercise.

Based on this, the following questions were put forward:

1. What properties of the data and business requirements needed to be considered in determining the appropriate forecasting technique to use?
2. How could the forecast outputs be properly interpreted to make the right investment recommendations to management?
3. How could a pricing model be developed to minimize fluctuation in demand and optimize capacity utilization?

1.3 SCOPE: PROJECT GOALS AND OUTCOMES

The project sought to develop a forecasting model for demand of a cloud product offering in selected zones. This robust and verifiable forecast is used in capacity planning.

The time series forecasting and machine learning capabilities of Python were used to determine and adopt the right forecast methodology to use. The Mean Absolute Percentage Error (MAPE) performance evaluation parameter was used to assess the accuracy of the model. A suitable demand elastic pricing model was proposed to help the company manage and smooth out capacity utilization.

The key deliverables to the company are:

1. A forecasting methodology to use for future capacity planning requirements.
2. A demand elastic pricing model to smooth out demand and optimize capacity.

To achieve the project objectives and deliverables, the project plan included the following steps. First, existing literature on IaaS cloud computing, forecasting in cloud computing, and dynamic pricing models, were explored. Next, the historical demand data was analyzed to identify features of the demand pattern, such as trend and seasonality. The outcome of these first two steps lead to the determination from several candidate methodologies, of the most suitable to use for the forecasting exercise. The selected methodology was determined the Mean Absolute Percentage Error (MAPE) performance criteria. Finally, the recommendation for the use of this standardized, verifiable, and repeatable methodology was provided to the product managers for use in capacity planning, as well as a suitable pricing model to handle demand seasonality.

The expected benefits to the company include having robust information to make the right level of investment, and getting the best out of their working capital. As well, it gives them a competitive advantage, as they will continue to proactively plan and invest to meet the required capacity demands and service levels with their customers. Finally, the elastic pricing model will enable them influence consumer behavior to smoothen out capacity utilization and better manage revenue generation.

2. STATE OF THE ART

To address the focus of the capstone project — to determine and implement the most suitable methodology for the forecasting of demand for cloud computing services to facilitate the capacity planning process, as well as to propose a suitable pricing model — literature was reviewed in several areas: (1) cloud computing, (2) Infrastructure as a Service (IaaS) (3) times series forecasting in cloud systems, and (4) demand elastic pricing models.

2.1 CLOUD COMPUTING

Cloud computing is a phrase utilized to refer to a specific category of application and platform. The dynamic allocation, configuration, reconfiguration, and deallocation of servers as required, characterizes a cloud computing platform. Cloud servers may be either physical or virtual machines.

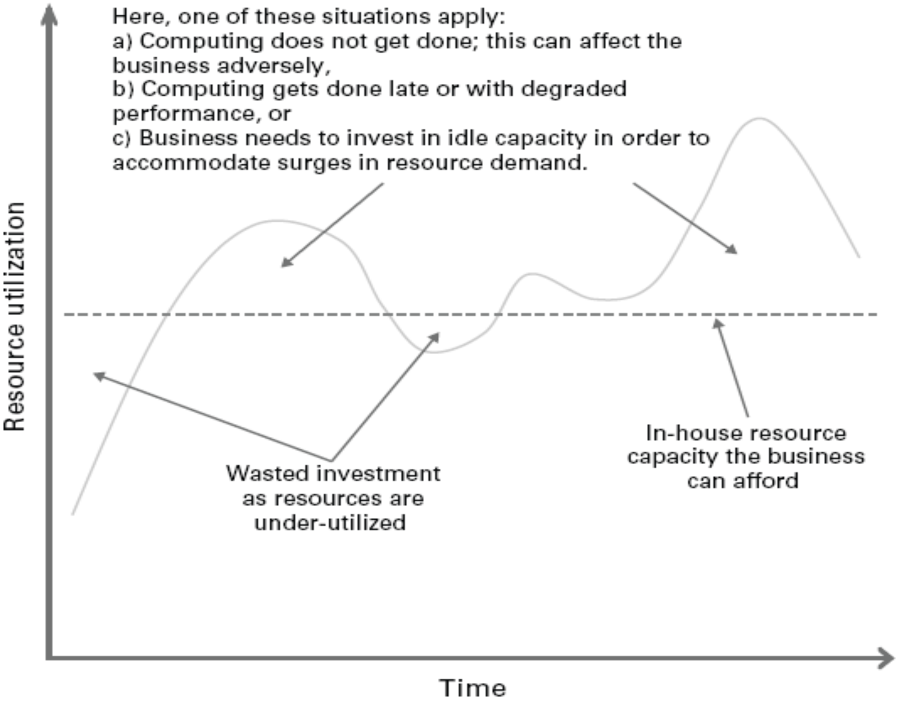
A Virtual Machine (VM) is a form of computing that employs software instead of a physical device to execute software programs and launch applications. A single physical "host" machine can accommodate one or more virtual "guest" machines. Each VM operates its own operating system and works independently from other VMs, even if they are running on the same host. This implies that a virtual MacOS machine can be run on a physical PC, for instance (VMware, 2023).

Sophisticated cloud systems generally involve other computing resources like storage area networks (SANs), network hardware, firewalls, and additional security equipment. These applications are designed to be available on the internet. They use large data centers and powerful servers which host web applications and services. Cloud applications can be accessed by anyone with an internet connection and a browser (Boss et al., 2007).

Ruparelia (2016) notes that cloud computing helps to solve the dilemma of underinvesting or overinvesting in computer resources due to the variable demand. Businesses are able to invest in computing resources on an as-needed basis as cloud computing converts the capital expenditure on computing resources into an operating one. Figure 1 shows different levels of usage for a fixed level of investment and the deficits or idle capacities on either side of the investment line.

Figure 1

Investment Problem Solved by Cloud Computing



Note. From *Cloud Computing*, by N. B. Ruparelia, “Introduction” (p. 2), 2016, MIT Press Scholarship Online

Mell and Grance (2011) explain that the cloud computing model promotes availability and is composed of five essential characteristics, three service models, and four deployment models. These are discussed in sections 2.11, 2.12 and 2.13, respectively.

2.1.1 CHARACTERISTICS OF CLOUD COMPUTING

On demand self-service: Consumers can provision computing capabilities as required, without interaction with each service provider.

Broad network access: Heterogeneous client platforms such as mobile phones, laptops, and tablets serve as standard platforms through which available capabilities are accessed over a network.

Resource pooling: Using a multi-tenant model, consumers are served by the provider's pooled computing resources. Consumers are only able to specify location at a high level of abstraction, such as, country, state or data center. They have no knowledge of the exact location of services provided.

Rapid elasticity: In some cases, capabilities can be elastically employed and released to scale rapidly up or down as demand requires. The capabilities appear unlimited to the customer.

Measured service: Using a metering capability, resource usage is optimized to the type of service. The monitoring, controlling, and reporting of resource usage, provides transparency for both the provider and consumer (Mell & Grance, 2011).

2.1.2 CLOUD COMPUTING SERVICE MODELS

Software as a Service (Saas): The consumer is provided the capability to use the provider's applications running on the cloud. Client interfaces such as a web browser or program interface on client devices, present avenues through which the client can access the applications. With the exception of limited user specific configurations, the underlying cloud infrastructure, are not managed by the consumer.

Platform as a Service (Paas): The consumer can deploy unto the cloud infrastructure, provider supported applications using programming languages, tools, services and libraries created or acquired by the client. The consumer has control over deployed applications and configuration settings for the application hosting environment but not the underlying cloud infrastructure.

Infrastructure as a Service (Iaas): The consumer can provision fundamental computing resources. They are also able to deploy and run arbitrary software such as operating systems and applications. The consumer has control over operating systems, storage, deployed applications, and limited control over select networking components, but not the underlying cloud infrastructure (Mell & Grance, 2011).

2.1.3 CLOUD COMPUTING DEPLOYMENT MODELS

Private Cloud: A single organization comprising multiple consumers (e.g., business segments or units) has exclusive access to the cloud infrastructure. The infrastructure may exist on or off premises and be owned, managed, and operated by the organization, a third party, or some combination of both.

Community Cloud: A specific community of consumers from an organization with shared concerns (e.g., security requirements, policy, and compliance considerations) has exclusive access to the cloud infrastructure. The infrastructure may exist on or off premises and be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of both.

Public Cloud: The general public has access to the cloud infrastructure. The infrastructure exists on the premises of the provider and may be owned, managed, and operated by a business, academic, or government organization, or some combination of them.

Hybrid Cloud: When two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology, are combined together, a hybrid cloud is obtained. The technology enables data and application portability (Mell & Grance, 2011).

As the service provider of the capstone is an IaaS provider, IaaS cloud computing will be further explored. This is to enable us to understand the structure and features and to determine the best methodology to employ in forecasting.

2.2 INFRASTRUCTURE AS A SERVICE

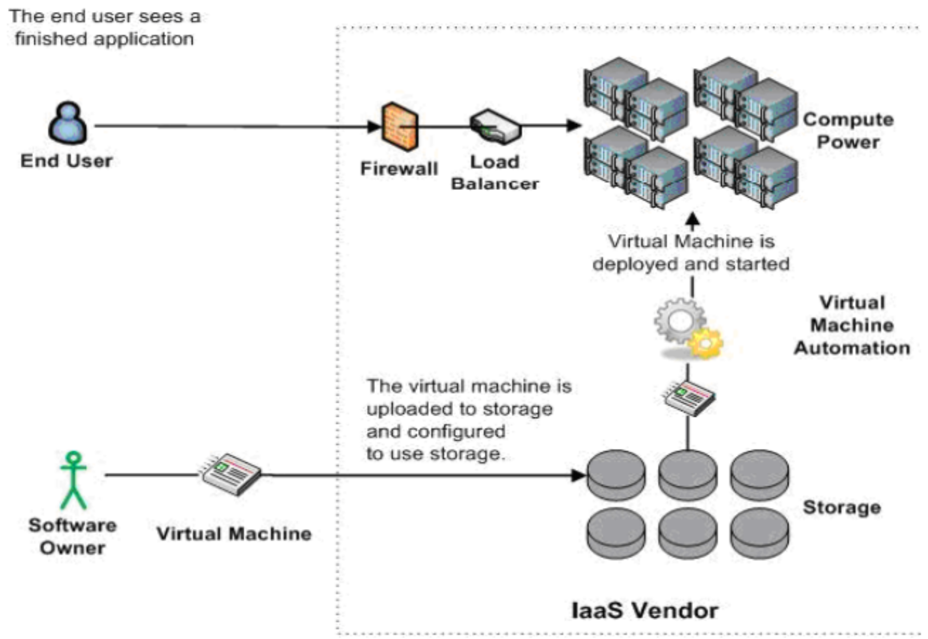
Bhardwaj et al. (2010) define Infrastructure as a Service (IaaS) as the delivery of hardware and related software as an evolution of traditional hosting that does not require any long-term commitment and allows users to provision resources on demand. The provider just keeps the data center operational while the users deploy and manage the software services like they would at their own data center.

IaaS is a form of hosting which includes network access, routing services and storage. The service provider provides the hardware and administrative services required to store applications and a platform for running them. Bandwidth scaling, memory and storage are generally included, and vendors compete on the performance and pricing offered for their dynamic services. The service is responsible for housing, running and maintaining the equipment, which they own. IaaS can be bought either on a pay-as-you-go basis or as a contract. Most buyers, however, consider the major benefit of IaaS to be the flexibility of the pricing, as they only need to pay for the resources which the delivery of their applications require (Bhardwaj et al., 2010).

Figure 2 illustrates how IaaS provides an environment for running user-built, virtualized systems in the cloud.

Figure 2

Infrastructure as a Service


The end user sees a finished application
End User
Firewall
Load Balancer
Compute Power
Virtual Machine is deployed and started
Virtual Machine Automation
The virtual machine is uploaded to storage and configured to use storage.
Software Owner
Virtual Machine
Storage
IaaS Vendor

Note. From *Cloud Computing: A study of infrastructure as a service (IAAS)*, by S. Bhardwaj, L. Jain, & S. Jain, “Understanding Infrastructures as a Service (IaaS)” (p. 62), 2010, International Journal of Engineering and Information Technology

The views of IaaS for the Consumer and Provider are listed below.

Consumer’s view on IaaS

- Applications accessed from anywhere
- A flexible, scalable, virtualized and automated modular system
- Adaptable and obtainable

- Applications and data on platform provisioning and maintenance by provider
- Own the hardware and control things like space and power

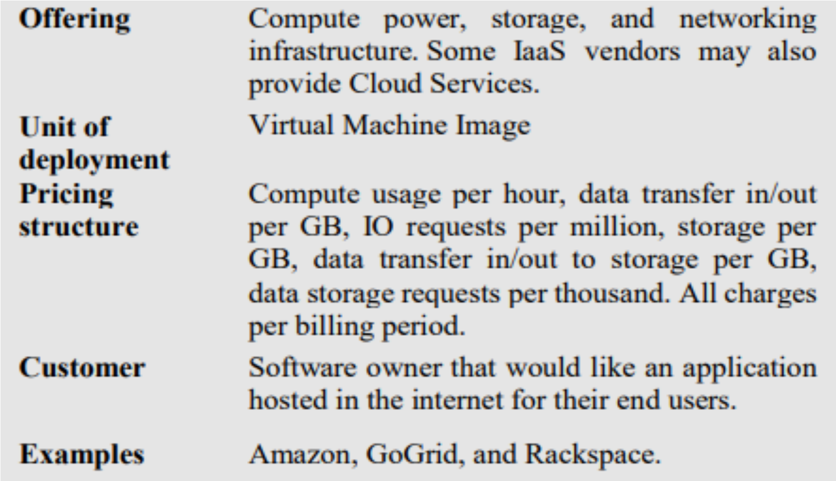
Provider’s view on IaaS

- Provide virtual infrastructure (server, storage and Network virtualization).
- Provisioning of power and space.
- Provide load balancing services.
- Deployment of web-based applications to provision infrastructure for customer as demanded
- Provisioning and account management (Bhardwaj et al., 2010).

To sum up, an IaaS offering provides cost savings as the associated infrastructure and networking do not need to be purchased and maintained by the customer. These assets are the responsibility of the IaaS vendor and customers are only charged as they use it. Figure 3 summarizes the main features of IaaS. IaaS often appeals to infrastructure architects because it provides an infrastructure-based approach to outsourcing datacenter workloads to the Cloud. If an application can be virtualized it can be uploaded to an IaaS environment and run (Bhardwaj et al., 2010).

Figure 3

IaaS Summary



Offering	Compute power, storage, and networking infrastructure. Some IaaS vendors may also provide Cloud Services.
Unit of deployment	Virtual Machine Image
Pricing structure	Compute usage per hour, data transfer in/out per GB, IO requests per million, storage per GB, data transfer in/out to storage per GB, data storage requests per thousand. All charges per billing period.
Customer	Software owner that would like an application hosted in the internet for their end users.
Examples	Amazon, GoGrid, and Rackspace.

Note. From *Cloud computing: A study of infrastructure as a service (IAAS)*, by S. Bhardwaj, L. Jain, & S. Jain, “Conclusion” (p. 63), 2010, International Journal of engineering and information Technology

2.3 FORECASTING IN CLOUD SYSTEMS

The ability to efficiently adapt available resources to workloads over time highlights the elasticity property of cloud computing as an important feature of computing services. To achieve a truly elastic system, the problem of workload forecasting has to be tackled (Baldan et al., 2018).

Buyya et al. (2009) note that cloud computing grants the same utility properties as those of electricity, telephony, water and gas services, to computing. It is important that the customers are able to have unlimited access to the services while only paying for what they use. Two of the underlying property of this paradigm are “self-scaling” and “elasticity.” The system has to be able to modify available resources to match different workloads over time, avoiding inefficient use of these resources. As cloud systems are often unable to react immediately to changes of variables, a predictive system is essential for the intelligent management of resources (Baldan et al., 2018).

2.3.1 SCALABILITY AND ELASTICITY IN CLOUD

Users of a cloud platform usually only require a variable amount of resources to be dynamically released by the service provider as demanded. Resources are thus provided in an *elastic* manner (Konstanteli et al., 2014).

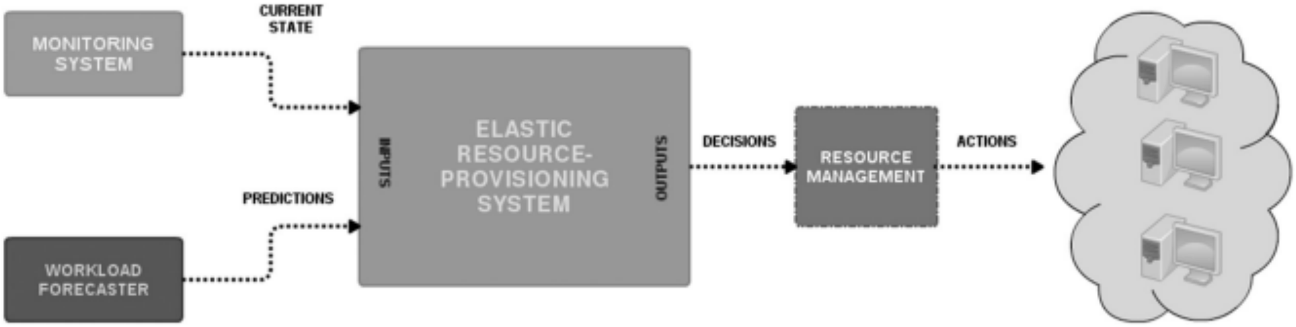
Scalability, which is the ability to increase workload size within existing infrastructure without impacting performance, differs from elasticity in that manual installation and deployment of resources are carried out, and is dependent on early detection of issues. Elasticity, on the other hand, involves the automatic and instantaneous response to issues. It is synonymous with “self-scaling”; there is no human intervention required. This situation of dynamism is more relevant for systems which have to face peak demands. It is preferred to have a cloud platform which can react to these eventualities than a mostly underutilized infrastructure (Baldan et al., 2018).

There are several advantages of elasticity. These include: smaller provisioning cycles, more efficient maintenance of server applications, faster adoption of new web-based models, higher security and integrity levels, better user experience, and lower levels of complexity (Baldan et al., 2018).

Knowing that cloud clients sign up for a service-level agreement with the cloud service provider for minimum levels of service, usually with attached penalties, Roy et al. (2011) note that capacity and demand have to be planned for average and peak loads. The provider incurs lower cost when provisioning cloud services for average load, as less hardware is required. This, however, negatively impacts performance during peak demand and could lead to penalties and diminished user experience. On the other hand, it costs the provider more to provision cloud services for peak load. The resources are underused a lot of the time and unnecessary expenses will be incurred in procuring and maintaining the infrastructure. These two issues are the crux of the problems with cloud workload forecasting (Buyya et al., 2011). Figure 4 shows the general arrangement of an elastic cloud system.

Figure 4

Elastic Cloud System



Note. From *A Forecasting Methodology for Workload Forecasting in Cloud Systems*, by Baldan et al., “A Forecasting Case Study: Workload in Cloud Systems” (p. 932), 2018, IEEE Transactions on Cloud Computing

The current platform state which is obtained from real-time process monitoring is one of the inputs to the elastic cloud system. The other is the forecast for the following states, made by the workload forecasting module. This module sends forecasts obtained from historical data to the resource provisioning system. The module then makes decisions which are sent to the management component. The resource management component finally takes actions (switching on/off virtualized resources) on the pool depending on received instructions.

The main parameters to consider in an elastic cloud system are those related to the use of the main resources offered by the platform – central processing unit (CPU) and memory utilization, disk space, and bandwidth. These parameters must be studied, forecasted and managed properly for the system to make appropriate provisioning and management of resources at any time. CPU load and memory use are the most critical and scarce resources in a computer system. They are the main bottlenecks in cloud platforms (Baldan et al., 2018).

As the capstone seeks to forecast the demand, as well as develop a pricing model to smooth out demand, candidate models capable of testing for trend and seasonality were assessed. The selected methodology was employed to analyze and forecast this demand for capacity planning.

2.3.2 FORECASTING MODELS

Literature was explored on several suitable time series forecasting models which were relevant to the data set obtained.

2.3.2.1 NAÏVE

The naïve method of forecasting is the simplest technique as it uses the most recent observation as the forecast for the next period. Naïve forecasts are optimal when data follows a random walk pattern. Random walk data are usually non-stationary like financial and economic data (Hyndman & Athanasopoulos, 2021).

2.3.2.2 MOVING AVERAGE

Using the moving average is usually the first step in a classical time series decomposition to estimate the trend. To estimate the trend-cycle at a specific time, *t,* the time series values are averaged over certain time periods preceding *t*. Since values that are close in time tend to be similar in value, averaging helps remove some of the random fluctuations and creates a smoother trend-cycle component. This is referred to as an *m*-MA, which stands for a moving average of order *m* (Hyndman & Athanasopoulos, 2021).

2.3.2.3 LINEAR AUTO-REGRESSION

Also referred to as the *AR(p)* model, this model is frequently used as a benchmark model. It is considered as a special case of ARIMA and is based on considering each variable value as a linear combination of the previous ones (Box et al., 2015). For the capstone, this will be used by considering the days and weekends as variables. The weekends will be modelled as binary variables; 1 for weekdays and 0 for weekends.

2.3.2.4 ARIMA

The combination of an Autoregressive Moving Average (ARMA) model and differencing gives the non-seasonal Autoregressive Integrated Moving Average (ARIMA) model. By taking the difference between consecutive values of a time series, the mean of the series can be made more stable. This is because differencing removes variations in the level of the series, which in turn removes or decreases trend and seasonality (Hyndman & Athanasopoulos, 2021). ARIMA models are denoted as ARIMA (p, d, q), where *p* is the number of lagged values considered in the autoregressive part, *q* is the number of lagged values considered for the moving average part, and *d* is the number of differences considered (Baldan et al., 2018).

ARMA is a forecasting model that utilizes both auto-regression (AR) analysis and moving average (MA) methods on well-behaved time-series data. The model assumes that the time series is stationary and that any fluctuations occur uniformly around a specific time (Gordon, 2022).

2.3.2.5 SARIMA

A seasonal ARIMA (SARIMA) model is formed by including additional seasonal terms in the ARIMA model. It includes the denotation (*P, D, Q)m* for the seasonal part of the model, where *m* represents the number of observations per year. The seasonal part comprises terms identical to the non-seasonal part but incorporates backshifts of the seasonal period (Hyndman & Athanasopoulos, 2021). A SARIMAX model is formed by feeding the SARIMA model with exogenous variables.

2.3.2.6 EXPONENTIAL SMOOTHING (ETS)

ETS takes a weighted average of past values to predict future values. The idea behind ETS is that recent values are more important than older values when predicting the future. They are thus assigned higher weights. This model can then be extended to include trend and seasonal components (Peixeiro, 2022).

2.3.2.7 PROPHET MODEL

Prophet is an additive regression model consisting of four parts:

- A piecewise linear trend. Prophet selects changepoints from the data to detect changes in the trend.
- A yearly seasonal component modeled using Fourier series.
- A weekly seasonal component using dummy variables.
- A user-provided list of important holidays.

Two main benefits of using Prophet are:

- The package makes it much more straightforward to create a forecast by including different forecasting techniques, such as ARIMA and exponential smoothing, each with their own strengths, weaknesses, and tuning parameters.
- The forecasts are customizable and easily understandable. There are smoothing parameters for trends and seasonality, upper limits of the growth curves can be set, and

irregular holidays like the dates of the Super Bowl, Thanksgiving, and Black Friday, can be modeled (Taylor & Letham, 2017).

2.3.2.8 HOLT-WINTERS

The Holt-Winters model requires the user to select among an additive, a multiplicative, and a non-seasonal version based on the nature of the data. Starting values must then be selected for the seasonal factors, the local mean (level), and trend. These three constants, commonly denoted as α, β and γ, respectively are continuously updated by exponential smoothing as new observations become available (Chatfield, 1978).

2.3.3 TIME SERIES DECOMPOSITION

In order to understand the seasonality on a weekly level at each zone, the time series would have to be decomposed.

To select forecasting methods, a useful approach is to divide a time series into two components: systematic and unsystematic. The systematic components are those that display consistency or repetition and can be modeled and described, while the unsystematic components cannot be directly modeled. A typical time series consists of three systematic components - level, trend, and seasonality - and one unsystematic component, referred to as noise. The level is the series' average value, the trend indicates the increasing or decreasing value of the series, the seasonality shows the repeating short-term cycle, and noise represents the random variation in the series (Brownlee, 2020).

A series is considered to be an aggregate of these four components which can be combined additively or multiplicatively. There are several methods to decompose a time series.

2.3.3.1 AUTOMATIC TIME SERIES DECOMPOSITION

The Python statsmodels library provides an execution of the classical decomposition in a function called *seasonal_decompose ()*, which requires specification of an additive or multiplication model. The result contains the four components of the time series which can be plotted.

2.3.3.2 PROPHET TIME SERIES DECOMPOSITION

As discussed in section **2.3.2.7**, the Prophet forecasting model can also be used to decompose a time series into its constituent elements. The model contains a feature which can be used to remove the trend and noise from the data to visually reveal the seasonality. This is done by using the *plot_components ()* function (Clarke, 2021).

This is useful for the capstone as the identified seasonality is important for the objective of developing a pricing model to handle the weekly fluctuations to optimize capacity.

2.4 DEMAND ELASTIC PRICING MODELS

To smoothen out the seasonality observed in demand, the capstone aims to develop a demand elastic pricing model to incentivize customers to take up more capacity during the low cycles.

Gallo (2021) notes that the common belief is that customers in most markets tend to be price-sensitive, meaning that they are likely to buy a product or service if its price is lower and are less likely to purchase it if its price is higher. However, this idea can be expressed more precisely through the concept of price elasticity, which measures the degree of customer demand responsiveness to changes in the price of a product or service.

The formula for this sensitivity or elasticity is:

Price elasticity of demand = $\frac{\text{Percentage change in quantity demanded}}{\text{Percentage change in price}}$

or

$$\varepsilon = \frac{dQ/Q}{dp/p} \tag{1}$$

where ε = Elasticity of demand

Q = Quantity demanded

dQ = change in Quantity demanded

p = Price

dp = change in Price

The elasticity constant, ε, reveals by how much the quantity demanded changes when there is a change in price. For our purpose, it will be used to calculate by what magnitude the price should change when the quantity demanded changes.

2.4.1 SPOT PRICING IN THE CLOUD

An important concept in cloud systems is spot pricing. This is used to manage idle capacity and offer alternative pricing models other outside standard contract arrangements.

Microsoft Azure, a cloud service provider, uses this functionality. Spot virtual machines (VMs) are a cost-effective option for customers to acquire VMs from a pool of spare capacity that has not been utilized, with a price reduction of up to 90% compared to pay-as-you-go. The tradeoff, however, is that these spot VMs can be taken away with little notice, mainly when the demand for capacity increases or the VMs are required to serve reserved instances or pay-as-you-go customers. Although spot VMs may not be suitable for critical production workloads that cannot afford any interruption in service, substantial cost savings can be realized by operating various types of workloads, including stateless, non-production applications, and big data, on spot VMs (Spot, 2023).

3. DATA AND METHODOLOGY

The dataset obtained from the client shows historical daily sales of one product family from May 2019 until February 2022. It contains information about sales in 5 regions. These regions have zones under them, and there are a total of 13 zones. Table 1 shows the fields in the dataset.

Table 1

Data Fields

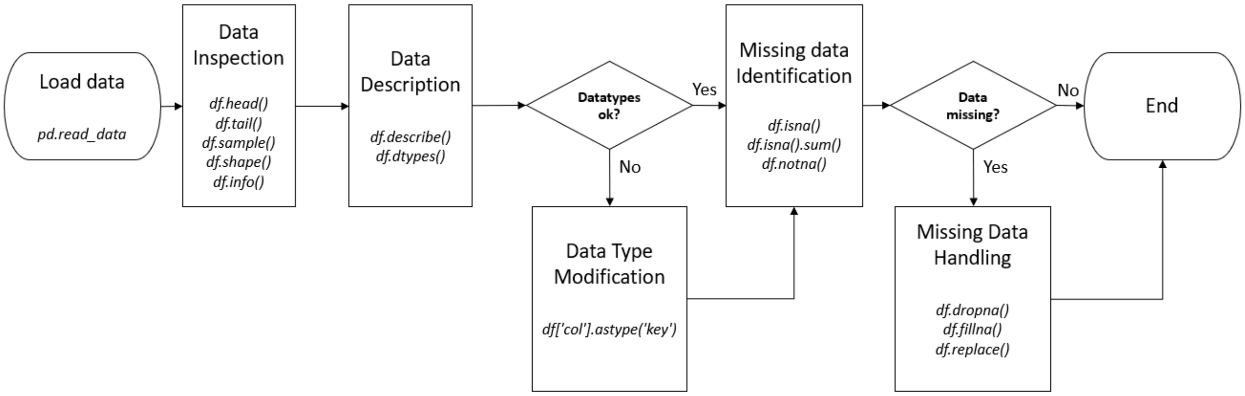
Data Field	Description
Usage_date	Date of consumption
Product Family	Virtual Machines calss with unique specifications
Sold QTY	Compute power
Zone	Building in location where servers and storage devices are kept
Region	Location where servers and storage devices are kept
Region-zone	Specific building in a location
Year	Date field showing year of consumption
Month	Date field showing month of consumption
Weeknum	Date field showing week of consumption
Year-Month	Date field showing year and month of consumption
Year-Weeknum	Date field showing year and week of consumption
Serial_Monthnum	Serial month number
Serial_weeknum	Serial week number

3.1 DATA PREPROCESSING

Before analyzing the data, several Python operations were carried out to check for missing or inconsistent values, and clean the dataset, if necessary. The workflow process in Figure 5 was followed to carry out the data preprocessing.

Figure 5

Data Preprocessing Workflow



The flow chart in Figure 5 shows the different processes and the Python commands used in preprocessing the dataset. First, the data was loaded into Python and converted into a Pandas data frame. It was then inspected to confirm the appearance of the data in the data frame, and check the number of rows and columns. Next, the names and data types of all the columns were examined. A check for missing data followed to conclude the process.

The data set obtained had no missing data or inconsistencies and was thus ready for analysis.

3.2 DATA VISUALIZATION

To help with the understanding of the data, identify any patterns or trends, and help with the selection of the analysis methods, the demand was plotted against time in days. For better visibility, the data was plotted by the five regions, with the individual zones under them. Figures 6, 7, 8, 9, and 10, show the demand trends for regions Alpha, Beta, Delta, Epsilon, and Gamma, respectively.

Figure 6

Region Alpha Demand Visualization

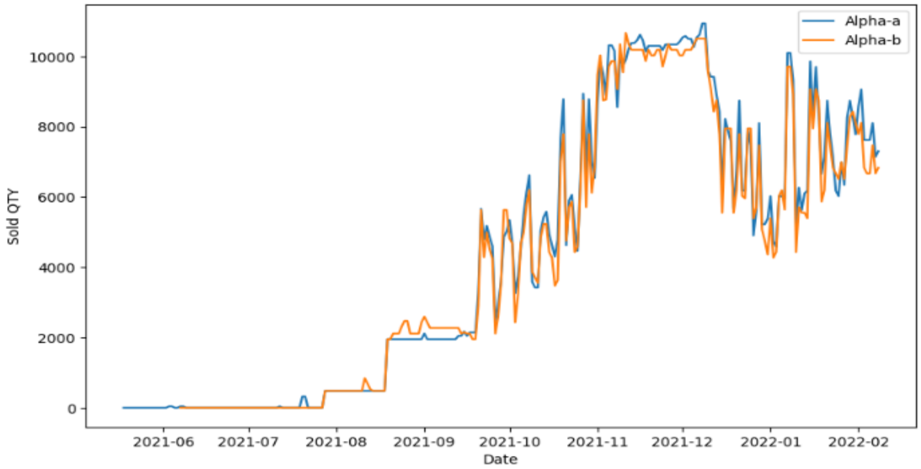


Figure 6 shows the demand for zones Alpha-a and Alpha-b. An initial startup phase with no demand is observed, which ramps up gradually. Though only limited data is available for this region, some positive trend and seasonality, are observed, and both regions follow a very similar demand pattern.

Figure 7

Region Beta Demand Visualization

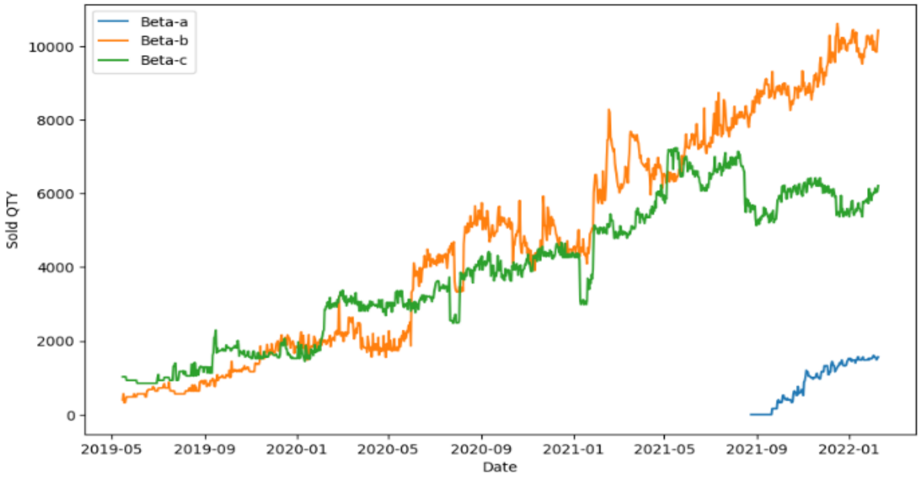


Figure 7 shows the demand for zones Beta-a, Beta-b and Beta-c. The demand pattern in this region shows a strong uniform positive linear trend, and some seasonality.

Figure 8

Region Delta Demand Visualization

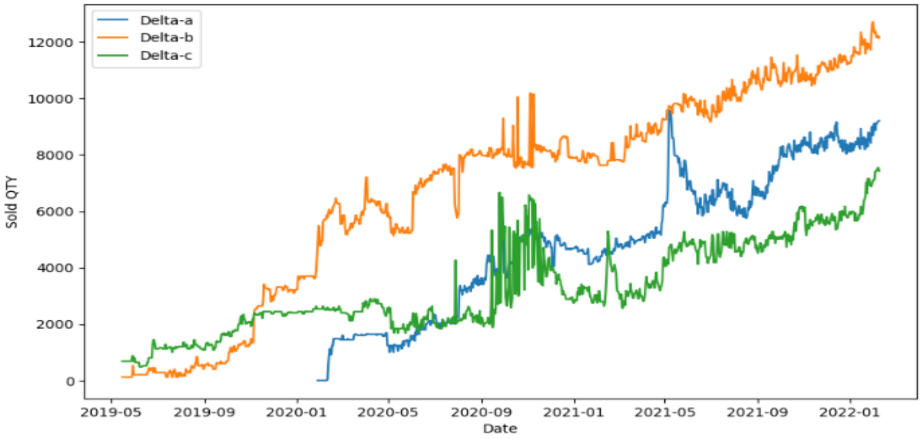


Figure 8 shows the demand for zones Delta-a, Delta-b and Delta-c. The demand pattern in this region follows an identical pattern in all zones and shows some positive trend and seasonality.

Figure 9

Region Epsilon Demand Visualization

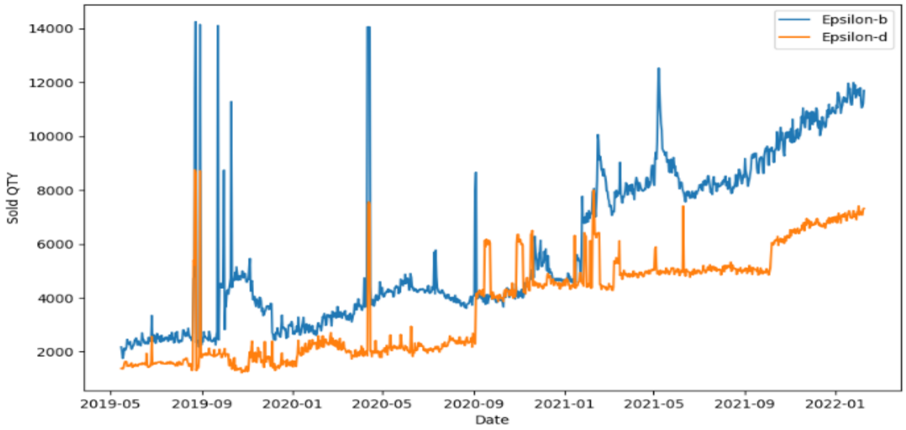


Figure 9 shows the demand for zones Epsilon-b and Epsilon-d. The demand pattern in this region follows an identical pattern in both zones and shows a slower moving positive trend when compared with the previous zones. There is some seasonality and obvious spikes in some months, which are explained as large bursts of demand by customers for very short periods.

Figure 10

Region Gamma Demand Visualization

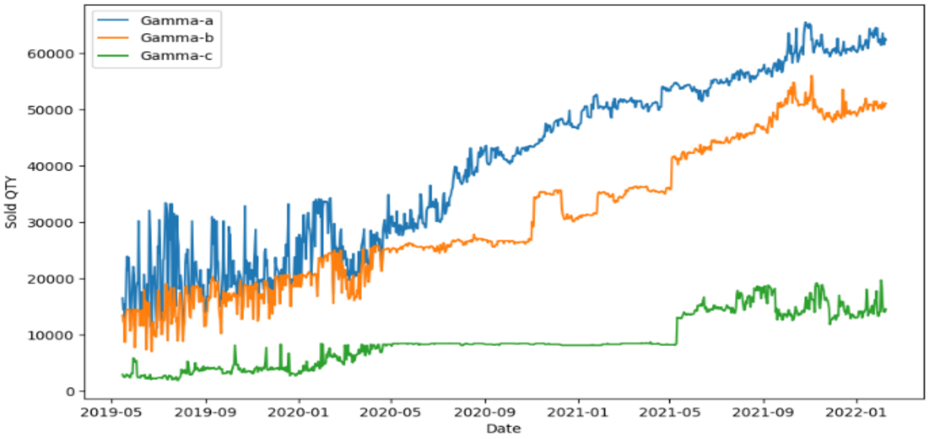


Figure 10 shows the demand for zones Gamma-a, Gamma-b and Gamma-c. The demand in this region follows an identical positive trend pattern in zones Gamma-a and Gamma-b. Initially, demand in both zones shows very erratic behavior but becomes less noisy subsequently. Demand in Gamma-c plateaus between May 2020 and May 2021, and while initially assumed to be a capacity limit, is explained by a steady state demand in that zone before a new customer is adopted.

On average, all the regions show positive linear trends which indicates that linear models will be good candidate models to identify trends. Relevant candidate models will also be adopted to test for and identify any seasonality, in order to adopt the best forecasting model.

3.3 DATA ANALYSIS

A summary analysis of the data was done to identify the main statistical parameters of every zone. Table 2 shows the summary statistics of the unique zones in the dataset.

Table 2

Statistical Summary of Dataset

Region-zone	Count	Mean	Std	Min	Median	Max
Alpha-a	267	4266	3873	0	3744	10,944
Alpha-b	247	4478	3691	0	4432	10,672
Beta-a	169	878	570	0	1040	1,607
Beta-b	1000	4706	3054	320	4532	10,613
Beta-c	1000	3867	1886	848	3898	7,238
Delta-a	742	4932	2498	0	4903	9,567
Delta-b	1000	6756	3614	120	7812	12,716
Delta-c	1000	3272	1615	480	2789	7,552
Epsilon-b	1000	5856	2935	1752	4595	14,248
Epsilon-d	1000	3736	1883	1216	4206	8,752
Gamma-a	1000	40062	15518	10048	42301	65,520
Gamma-b	1000	30687	12401	6960	26659	56,023
Gamma-c	1000	8735	4419	1823	8250	19,654

Table 2 shows that some of these zones have come online later than others, and after the earliest date in our dataset. Discussions with the company revealed that this is expected as new regions open due to customer demand. Also, management and capacity decisions are made at the zone level and forecasting is required at a daily level for rigorous analysis.

3.4 METHODOLOGY

For analysis, a test zone was picked and assessed to determine the most suitable approach and subsequently, this will be scaled to the remaining twelve zones. The zone *Beta-b* was used as the test model. The models are evaluated using Mean Absolute Percentage Error (MAPE), as the errors were expected to be proportional to the actual values and assessed on a percentage basis.

3.4.1 MODEL EVALUATION

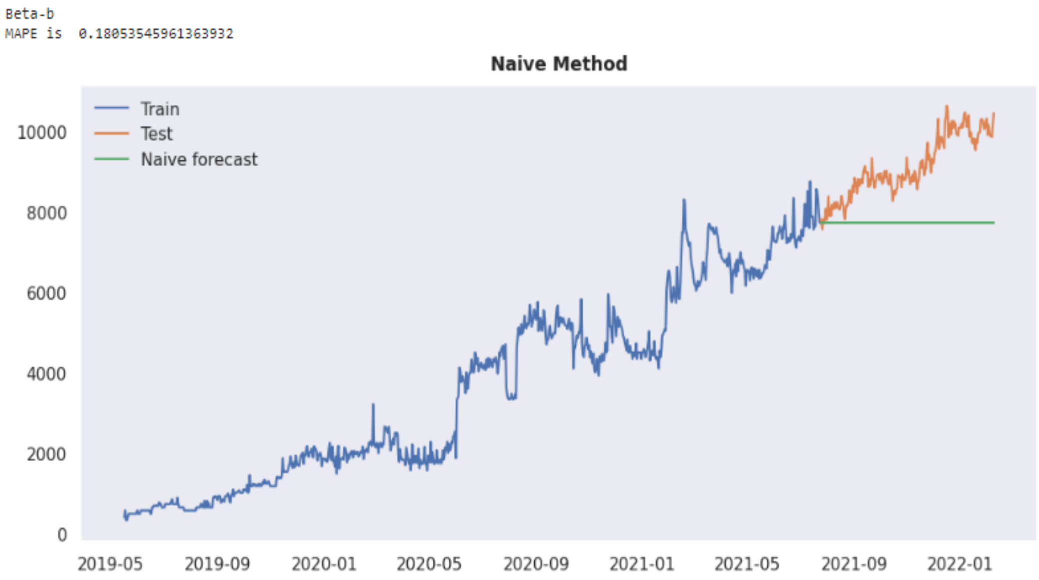
For the model assessments, the data was first assessed using the simplest forecasting method, which is the Naïve forecast. Here the actual demand for the previous day is used as the forecast for the next day. The insights from the model evaluations helped determine the next best suitable candidate model to evaluate for more insights.

3.4.1.1 NAÏVE FORECAST

Figure 11 shows the naïve test plot for Beta-b using a train-test split of 80:20.

Figure 11

Naïve Method



```
Beta-b
MAPE is  0.18053545961363932
```

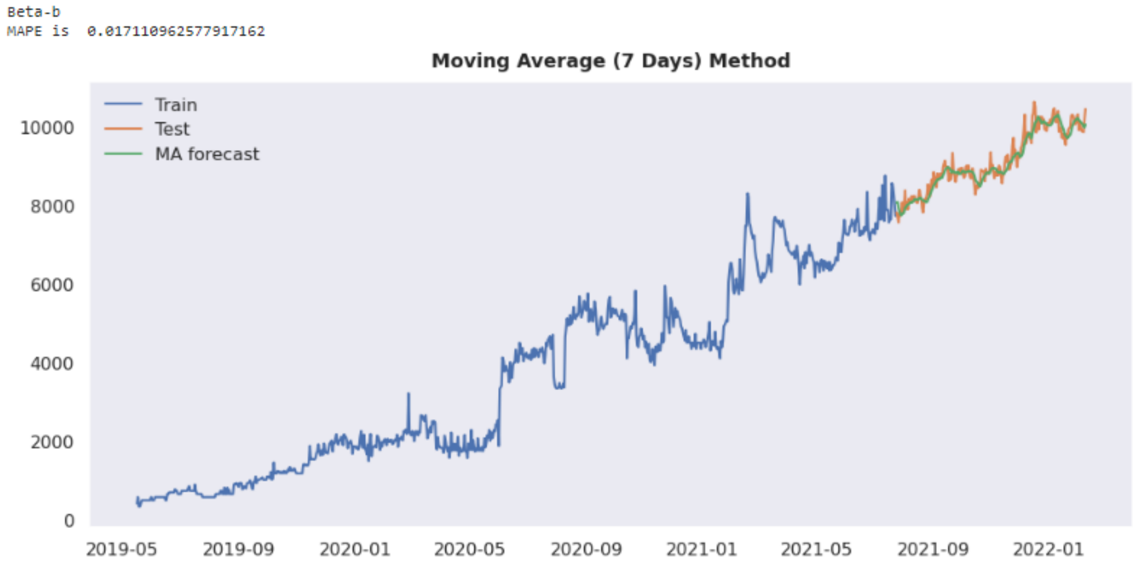
The model, which simply uses the most recent demand as the forecast, has a MAPE of 18%. Other models were assessed, starting with those models which test for linearity, the Moving Average and Linear Regression models, in 3.4.1.2 and 3.4.1.3, respectively.

3.4.1.2 MOVING AVERAGE

Figure 12 shows the 7 days moving average plot using an 80:20 train-test split. The 7-days parameter was selected to get the average demand for one week as a prediction.

Figure 12

7 Days Moving Average



```
Beta-b
MAPE is  0.017110962577917162
```

The 7 days moving average has a MAPE of 1.7%, which shows a high level of linearity.

3.4.1.3 REGRESSION ANALYSIS

The plot of Beta-b demand in Figure 7 shows an obvious positive linear trend. A suitable model to test this was a linear regression model using the days and weekend flags. The model is of the form

$$y = \beta_o + \beta_1.t + \beta_2.wf \tag{2}$$

where y = predicted sold quantity

β_o = Coefficient of the intercept

β_1 = Coefficient of the time in days

t = time count

β_2 = Coefficient of the weekend flag

wf = weekend flag which is 1 for weekdays and 0 for weekends

Figure 13 shows the linear regression plot, and Figure 14 shows the regression summary statistics.

Figure 13

Beta-b Regression Plot

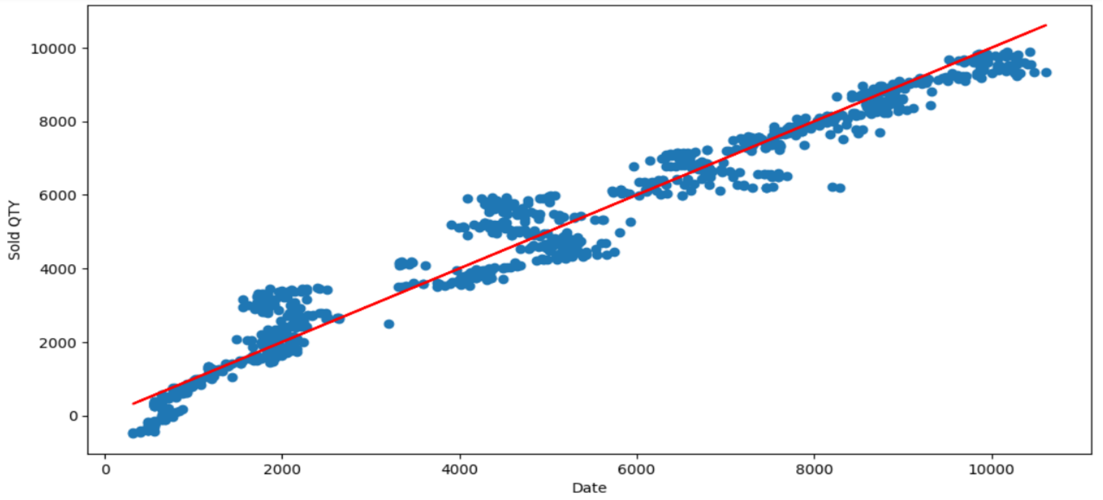
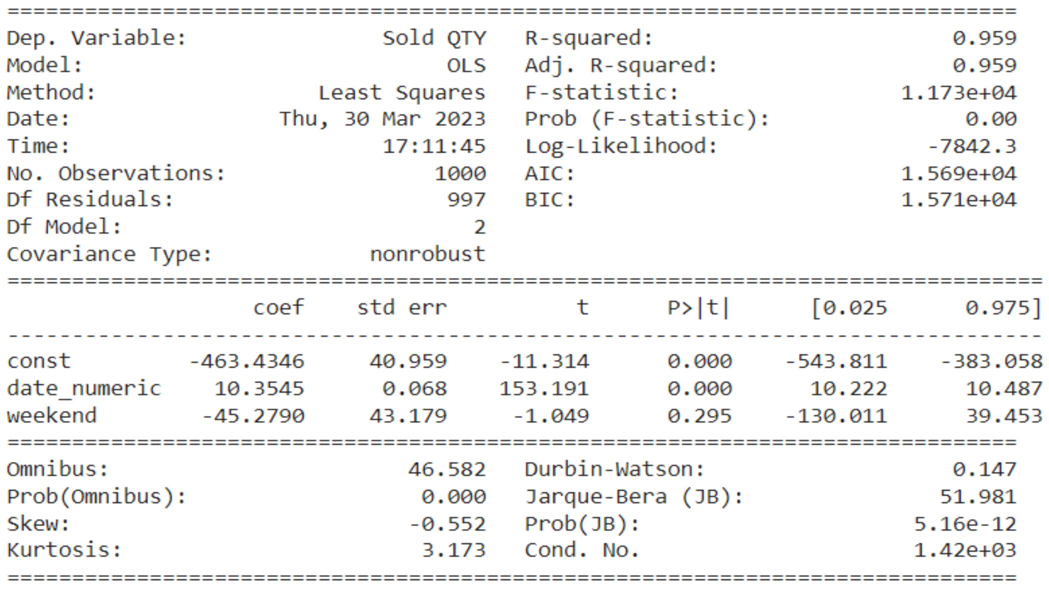


Figure 14

Beta-b Regression Summary Statistics



```
==============================================================================
Dep. Variable:               Sold QTY   R-squared:                       0.959
Model:                            OLS   Adj. R-squared:                  0.959
Method:                 Least Squares   F-statistic:                 1.173e+04
Date:                Thu, 30 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:11:45   Log-Likelihood:                -7842.3
No. Observations:                1000   AIC:                         1.569e+04
Df Residuals:                     997   BIC:                         1.571e+04
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const         -463.4346     40.959    -11.314      0.000    -543.811    -383.058
date_numeric    10.3545      0.068    153.191      0.000      10.222      10.487
weekend        -45.2790     43.179     -1.049      0.295    -130.011      39.453
==============================================================================
Omnibus:                       46.582   Durbin-Watson:                   0.147
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               51.981
Skew:                          -0.552   Prob(JB):                     5.16e-12
Kurtosis:                       3.173   Cond. No.                     1.42e+03
==============================================================================
```

The model has a high R-squared value of 96% which shows that the regression line highly fits the data. However, the coefficient of the constant or intercept, β_o, is unusually high at -463.43. Also, it is observed that the p-value of the weekend flag is insignificant at 0.295. The model is refit without the intercept. This is done to match reality better, as in this case, the regression line passes through the origin, 0.

3.4.1.4 REGRESSION ANALYSIS WITHOUT INTERCEPT

The new model is of the form

$$y = \beta_1.t + \beta_2.wf \qquad (3)$$

Figure 15 shows the linear regression plot and Figure 16 shows the regression summary statistics without the intercept.

Figure 15

Beta-b Regression Plot without Intercept

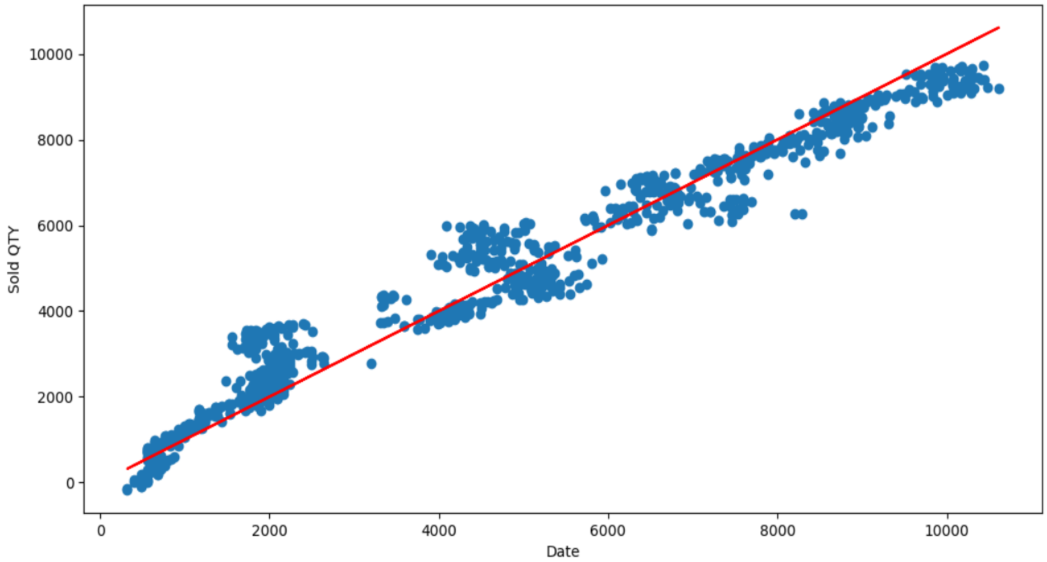
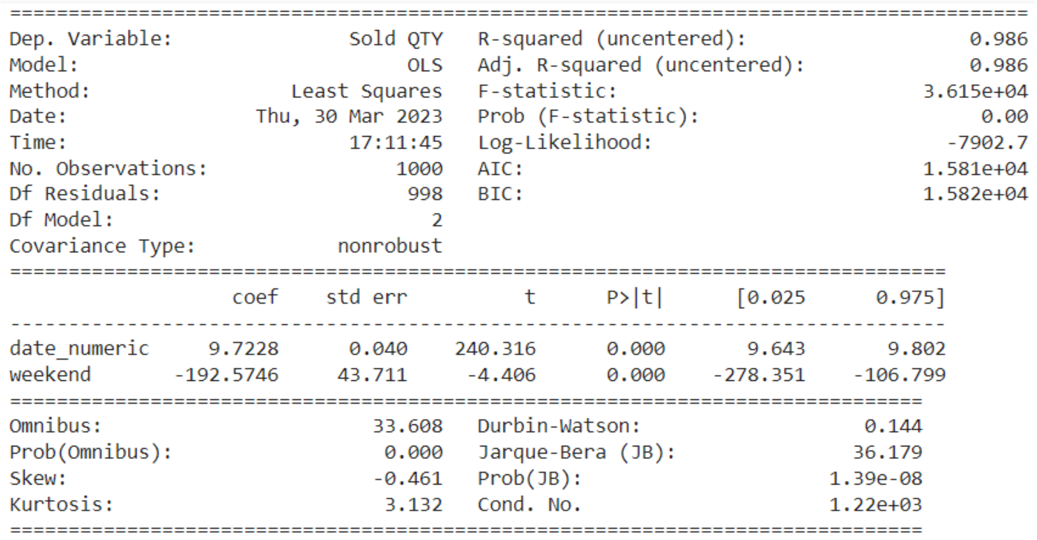


Figure 16

Beta-b Regression without Intercept Summary Statistics



```
==============================================================================
Dep. Variable:               Sold QTY   R-squared (uncentered):                   0.986
Model:                            OLS   Adj. R-squared (uncentered):              0.986
Method:                 Least Squares   F-statistic:                          3.615e+04
Date:                Thu, 30 Mar 2023   Prob (F-statistic):                        0.00
Time:                        17:11:45   Log-Likelihood:                         -7902.7
No. Observations:                1000   AIC:                                  1.581e+04
Df Residuals:                     998   BIC:                                  1.582e+04
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
date_numeric     9.7228      0.040    240.316      0.000       9.643       9.802
weekend       -192.5746     43.711     -4.406      0.000    -278.351    -106.799
==============================================================================
Omnibus:                       33.608   Durbin-Watson:                   0.144
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               36.179
Skew:                          -0.461   Prob(JB):                     1.39e-08
Kurtosis:                       3.132   Cond. No.                     1.22e+03
==============================================================================
```

The model has a higher R-squared value of 99% which shows that the regression line better fits the data. It is also observed that the p-value of the weekend flag is significant in this model. This shows a strong relationship between the variables, and indicates a good fit.

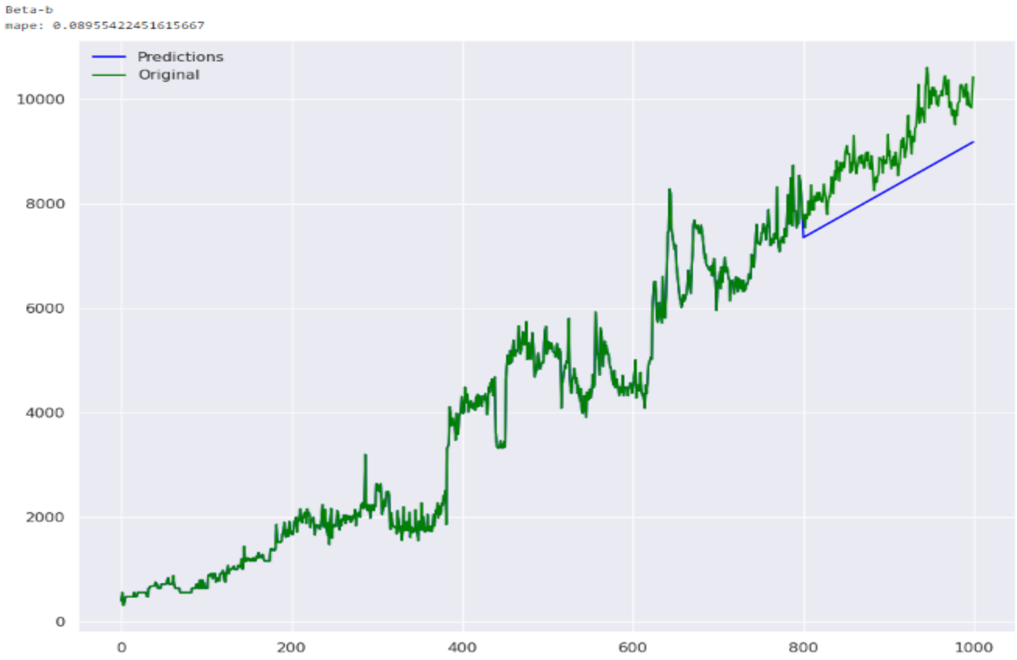
This model was tested using an 80:20 train-test split to assess the MAPE.

3.4.1.5 LINEAR REGRESSION MODEL

Figure 17 shows the plot of the Linear Regression model.

Figure 17

Linear Regression Model



```
Beta-b
mape: 0.08955422451615667
```

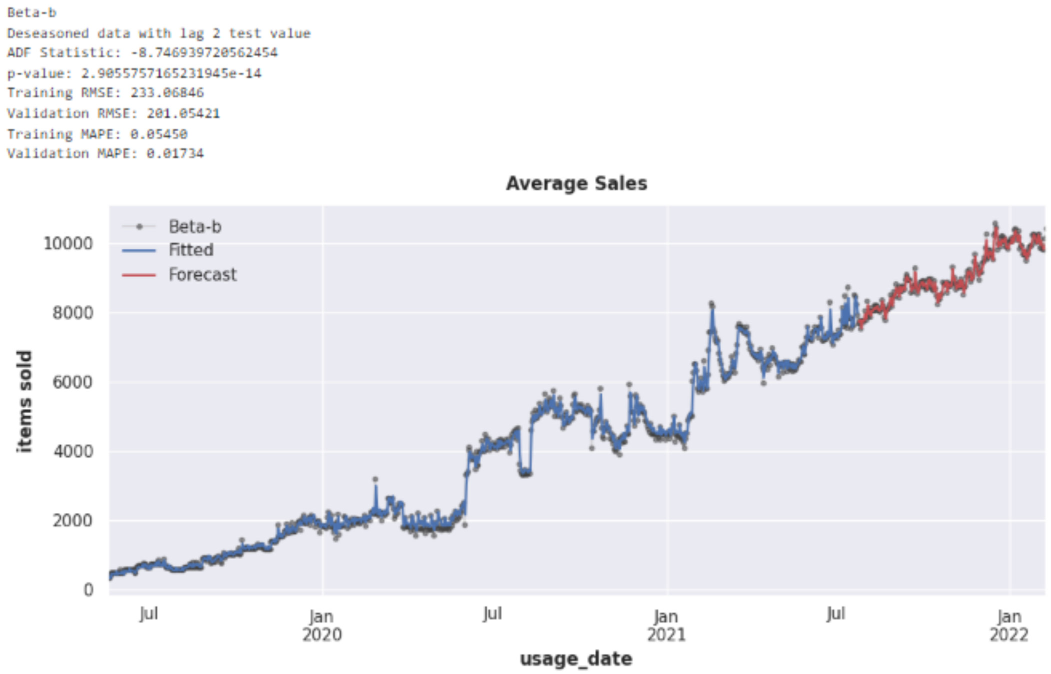
The model in Figure 17 has a MAPE of 9%. Using the machine learning functionalities in Python, to develop a more robust model, an improvement was done on the linear regression model using a Lag (actual data from the past as a feature to forecast), and a deterministic process, to model the weekly seasonality.

3.4.1.6 LINEAR REGRESSION WITH WEEKLY SEASONALITY AND LAG

Figure 18 shows the linear regression with a daily frequency of order 7 to represent daily data with weekly seasonality, and lag of 1.

Figure 18

Linear Regression Model with Weekly Seasonality and Lag



```
Beta-b
Deseasoned data with lag 2 test value
ADF Statistic: -8.746939720562454
p-value: 2.9055757165231945e-14
Training RMSE: 233.06846
Validation RMSE: 201.05421
Training MAPE: 0.05450
Validation MAPE: 0.01734
```

The model in Figure 18 has a MAPE of 1.7% which is a significant improvement on the base linear model in 3.4.1.5. The p-value is also very significant at 0. Again, indicating a strong relationship among the selected variables.

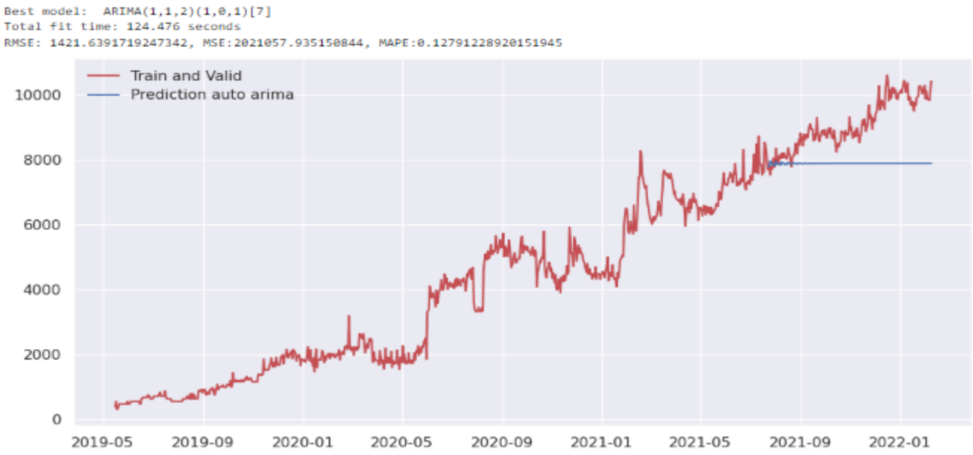
The data was next tested with the ARIMA model to evaluate the dependence of current values on the errors or residuals of past predictions. The SARIMA model incorporates seasonal differences as the previous linear models showed a good fit and high relationship of the variables, with weekly seasonality.

3.4.1.7 ARIMA AND SARIMA MODELS

The model determines the best parameters before fitting. Figure 19 shows the plot of the ARIMA model and Figure 20 shows the SARIMA evaluation.

Figure 19

ARIMA Model

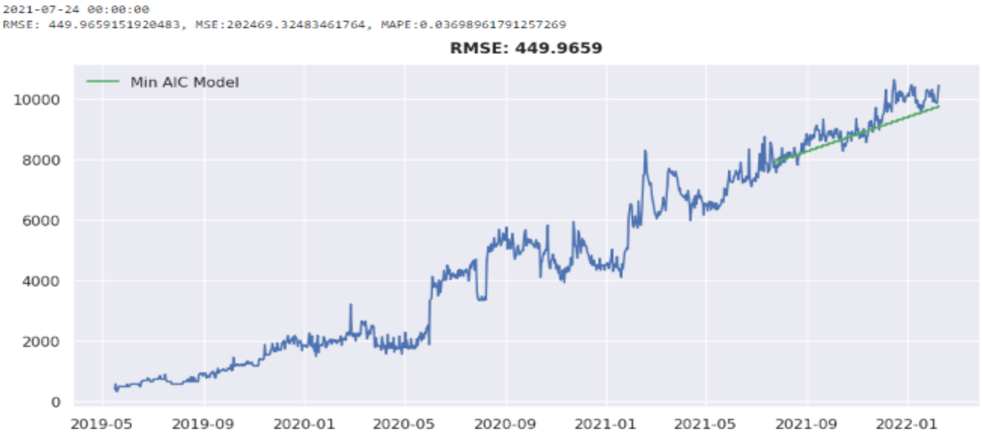


```
Best model:  ARIMA(1,1,2)(1,0,1)[7]
Total fit time: 124.476 seconds
RMSE: 1421.6391719247342, MSE:2021057.935150844, MAPE:0.12791228920151945
```

The ARIMA model has a MAPE of 13% and is not an improvement on the linear model assessed in 3.4.1.5.

Figure 20

SARIMA Model



```
2021-07-24 00:00:00
RMSE: 449.9659151920483, MSE:202469.32483461764, MAPE:0.03698961791257269
```

The SARIMA model has a MAPE of 37% which is worse than previous models. It indicates that the residuals or errors of previous predictions are not a good forecast indicator.

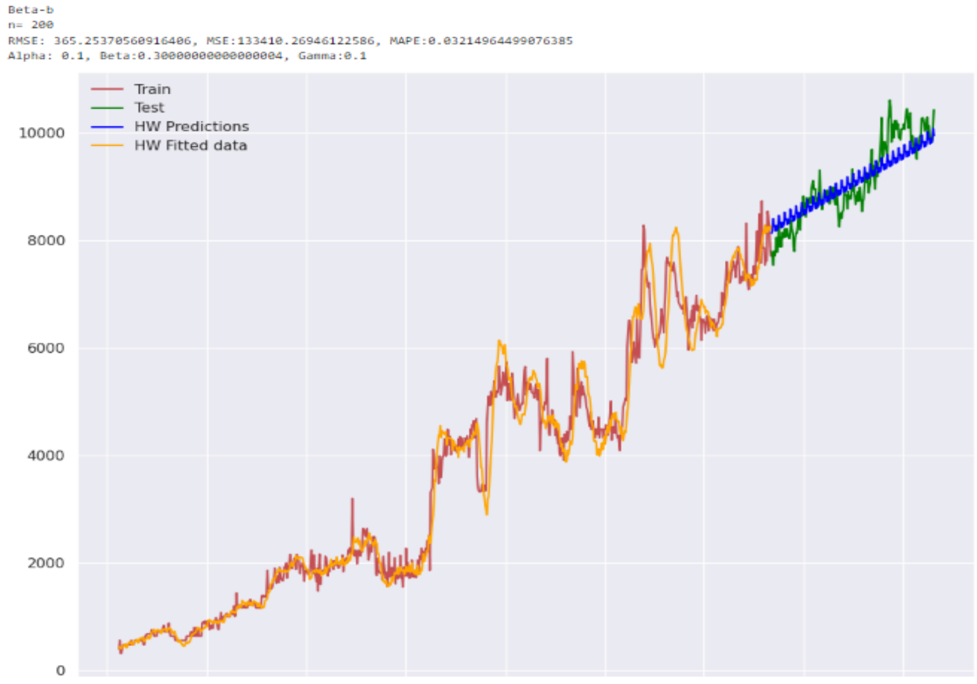
Another model which handles seasonality and trend, and was tested to improve on the performance of the SARIMA model is the Holt-Winters model. This is because the model, also known as the triple exponential smoothing model, is adaptive. It continuously updates the smoothing parameters, *α, β,* and *γ* which generally leads to higher accuracy.

3.4.1.8 HOLT-WINTERS MODEL

The Holt-Winters model determines the best values of *α, β,* and *γ* before fitting. Figure 21 shows the plot with *α, β,* and *γ* values of 0.1, 0.3 and 0.1, respectively.

Figure 21

Holt-Winters Model



```
Beta-b
n= 200
RMSE: 365.25370560916406, MSE:133410.26946122586, MAPE:0.03214964499076385
Alpha: 0.1, Beta:0.30000000000000004, Gamma:0.1
```

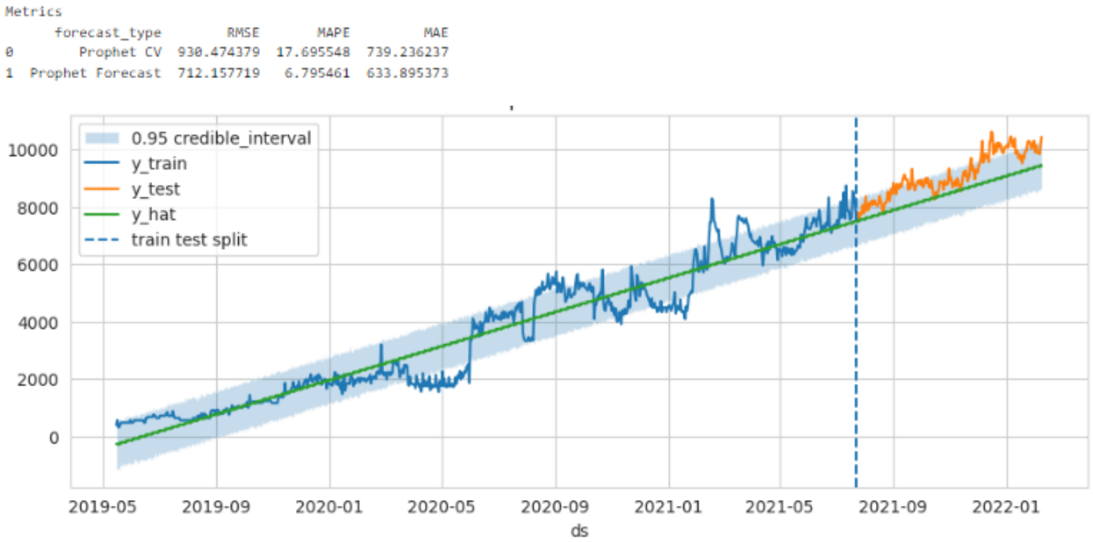
The model has a MAPE of 3.2% which is better than the ARIMA and SARIMA models in 3.4.7 but not as good at the linear model in 3.4.1.5. To use a model which combines the tests for trend, seasonality and exogenous factors such as holidays, the data was tested using the Prophet model.

3.4.1.9 PROPHET MODEL

Figure 22 shows the Prophet model plot of demand showing standard Prophet forecast and Prophet cross validation.

Figure 22

Prophet Model



```
Metrics
     forecast_type        RMSE       MAPE         MAE
0       Prophet CV  930.474379  17.695548  739.236237
1  Prophet Forecast  712.157719   6.795461  633.895373
```

Figure 17 shows the Prophet model with Standard Prophet forecast and Prophet cross-validation with MAPE values of 6.8% and 17.7% respectively.

The linear model with weekly seasonality outperforms all the models evaluated and confirms that addition of other factors does not model the data better than the trend and weekly seasonality. To better understand this seasonality which will be useful in price modeling, the time series was decomposed.

3.4.2 TIME SERIES DECOMPOSITION

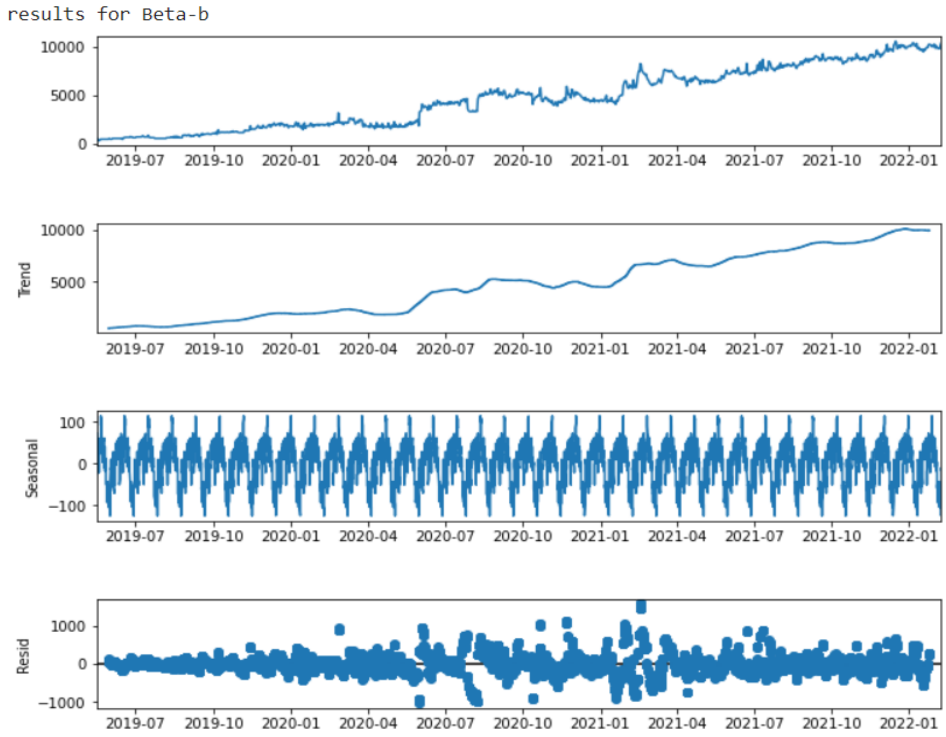
Python tools such as the statsmodels module and Prophet decomposition were used to decompose the data into the constituent elements of trend, seasonality and residuals.

3.4.2.1 STATSMODELS TIME SERIES DECOMPOSITION

The Python statsmodels module is an open-source module used for the estimation of different statistical models (Seabold et al., 2010). Figure 23 shows the output of the decomposition for the trend, seasonality and residuals.

Figure 23

Beta-b Time Series Decomposition



```
results for Beta-b
```

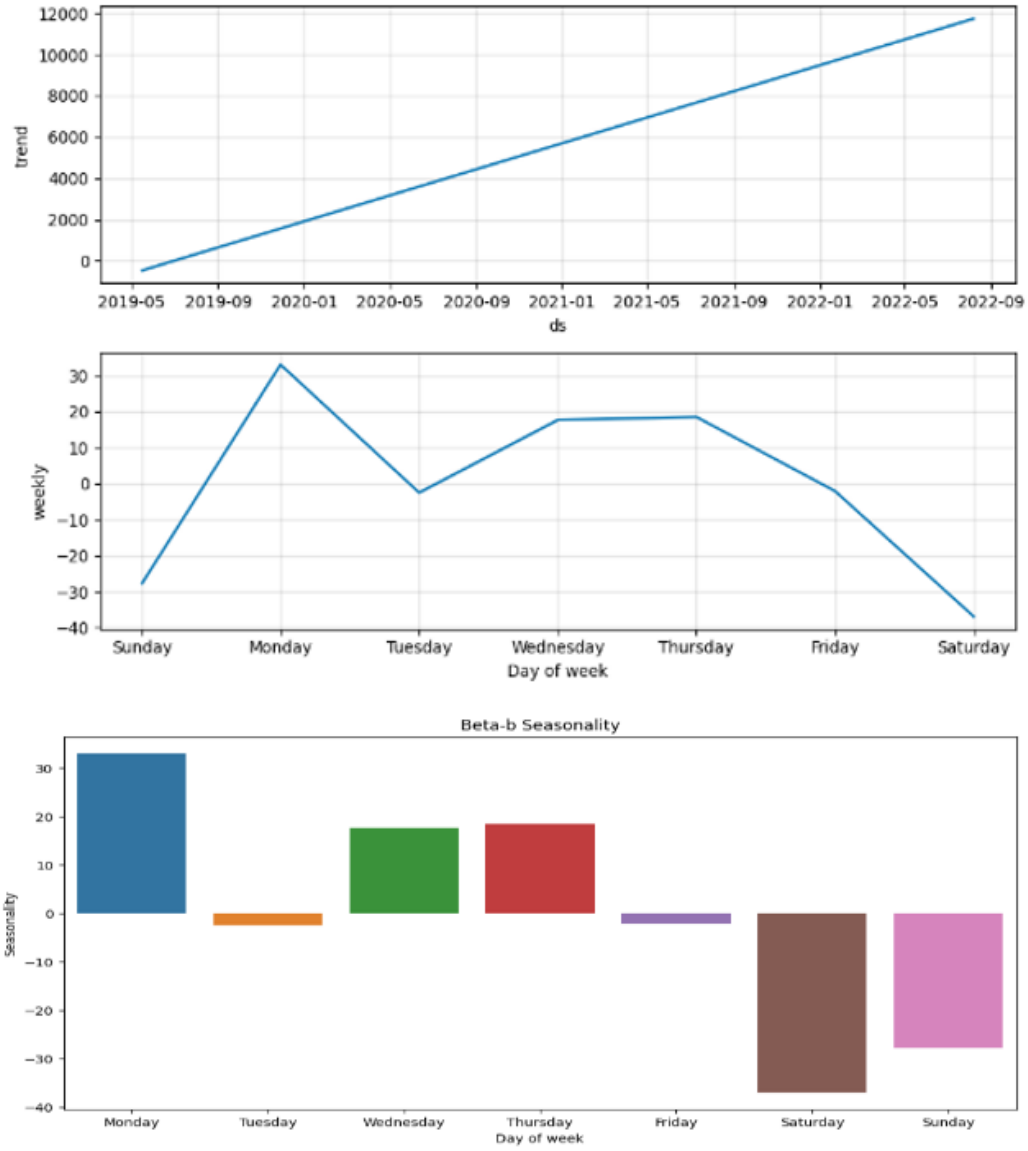
The decomposition output in Figure 23 shows that the trend is not linear. This is because statsmodels uses moving average to model the trend. The regression analysis carried out earlier in 3.4.1.5, however, showed high linearity. A better model for the decomposition is the Prophet model which was set up to identify the trend and weekly seasonality.

3.4.2.2 PROPHET TIME SERIES DECOMPOSITION

The Prophet time series decomposition of sold quantity in Figure 24 shows a linear trend which matches that of the regression analyses carried out earlier. The weekly seasonality is also evident.

Figure 24

Beta-b Prophet Time Series Decomposition of Sold Quantity



The weekly seasonality plot in Figure 24 shows that the demand spikes at the beginning of the week on Mondays and drops over the weekend, and this matches the expectation of usage for the cloud service. The Prophet model decomposition takes into consideration the linear trend and weekly seasonality and is thus a suitable model for the seasonality determination for all the zones.

Table 3 shows the weekly seasonality values for Beta-b.

Table 3

Beta-b Weekly Seasonality Values

		Weely Seasonality						
Region-Zone	Mean Demand	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Beta-b	4706	33	(3)	18	19	(2)	(37)	(28)

The seasonality values in Table 3 show demand reductions over the weekend. The demand picks up again at the beginning of the week. Values in parentheses indicate negative values.

3.4.3 DEMAND ELASTIC PRICING MODEL

The demand elastic pricing model discussed in 2.4 was tested on the Beta-b seasonality for Saturday. Using the equation in (1), the modified pricing is obtained;

$$\varepsilon = \frac{dQ/Q}{dp/p}$$

$$dp = p_2 – p_1 = \frac{\frac{dQ}{Q} * P1}{\varepsilon}$$

$$p_2 = p_1 – \frac{\frac{dQ}{Q} * P1}{\varepsilon} \qquad (4)$$

Where p_2 = Modified price due to seasonal change in demand

p_1 = Existing standard contract rate per unit time

dQ = Change in demand on specific day

Q = Mean daily demand

ε = Elasticity constant

For this illustration, the standard contract rate, P_1, is obtained by using the average rate of a particular Virtual Machine (VM) specification for some cloud service providers; Microsoft Azure, Amazon Web Services (AWS), Oracle Cloud, and Google. The VM specifications and pricing are shown in Appendix A. The average rate, P_1, is $0.159/hr.

The weekly seasonality for Saturday, *dQ*, is obtained from Table 3 as 37 units.

The mean daily demand, *Q*, for the focus region is obtained from Table 2. The value is 4,706 units.

An elasticity constant, ε, of 0.1, is assumed.

Substituting these values into (4),

$$p_2 = 0.159 - \frac{\frac{37}{4706} * 0.159}{0.1}$$

This gives a modified price of $0.146, which is an 8% decrease in the price for Saturdays.

4. RESULTS AND ANALYSIS

All the model evaluations carried out on the test zone, Beta-b, were done for the remaining 12 zones.

4.1 LINEAR REGRESSION

The complete linear regression analysis of all the zones is shown in Table 4.

Table 4

Linear Regression Summary

Region-Zone	Alpha-a	Alpha-b	Beta-a	Beta-b	Beta-c	Delta-a	Delta-b	Delta-c	Epsilon-b	Epsilon-d	Gamma-a	Gamma-b	Gamma-c
R Squared Value	0.87	0.87	0.98	0.99	0.97	0.98	0.98	0.95	0.94	0.95	0.97	0.97	0.96
Coefficient of Day	35.53	38.54	10.71	9.72	7.25	12.63	13.04	6.08	10.82	6.97	71.81	55.51	16.48
Coefficient of Weekend	(557.20)	(375.70)	(57.35)	(192.57)	214.54	227.21	207.77	188.40	346.91	185.75	2,825.90	2,174.04	237.36

The linear regression summary in Table 4 shows that all the zones have high R squared values of between 87% and 99%. This shows that the linear model fits the observed data to a high degree. Figures in parentheses represent negative values.

4.2 MODEL PERFORMANCE EVALUATION

A complete assessment was done for all 13 regions using all the candidate models, and the MAPE values are as shown in Table 5.

Table 5

Model Evaluation

		Linear Regression(Lag 1+Weekly Ind)	SARIMA	Auto ARIMA	(Manual Opt) Holt Winters	Prophet	Linear regression	Naïve	7D MA
S/No	Product Region	MAPE	MAPE	MAPE	MAPE	MAPE	MAPE	MAPE	MAPE
1	Alpha-a	16.69%	14.95%	30.06%	41.83%	89.90%	46.48%	29.64%	15.94%
2	Alpha-b	14.65%	25.69%	25.46%	17.16%	23.10%	78.80%	47.14%	16.54%
3	Beta-a	3.95%	12.32%	9.84%	2.10%	11.20%	30.50%	2.44%	1.78%
4	Beta-b	1.82%	3.70%	12.79%	3.21%	4.10%	8.96%	18.05%	1.72%
5	Beta-c	2.48%	27.44%	25.98%	13.20%	38.90%	24.50%	12.18%	2.12%
6	Delta-a	2.21%	8.83%	7.73%	6.10%	28.10%	6.80%	20.10%	2.10%
7	Delta-b	1.56%	2.87%	3.94%	6.60%	5.70%	14.20%	9.06%	1.48%
8	Delta-c	3.72%	5.63%	11.28%	8.10%	19.50%	5.60%	15.89%	2.67%
9	Epsilon-b	4.35%	9.03%	16.32%	3.00%	9.80%	4.90%	20.71%	2.10%
10	Epsilon-d	2.66%	10.66%	10.25%	5.10%	16.30%	9.30%	20.04%	1.51%
11	Gamma-a	1.74%	1.90%	3.39%	1.90%	4.50%	19.50%	6.49%	1.18%
12	Gamma-b	2.05%	5.58%	3.66%	3.20%	5.90%	6.90%	9.41%	1.16%
13	Gamma-c	4.86%	20.07%	17.22%	9.70%	26.23%	13.20%	11.05%	5.90%
	Avg MAPE	4.83%	11.44%	13.69%	9.32%	21.79%	20.74%	17.09%	4.32%

Table 5 shows that the Linear Regression and 7-Day Moving Average models perform best. This infers a very high linear trend in the demand data for all zones. The regions Alpha-a and Alpha-b perform worst across all the models, and this is easily attributable to the fact that they have the least amount of data available for the analysis.

The shortcoming of the best performing models, however, is their forecasting limitations. Due to the fitting properties of the Lag 1 and 7-day moving average models, they cannot be used to forecast further than one day of unseen data. The linear model with the weekly seasonality was rerun without the lag and, even though the overall MAPE performance reduced, it will be used for forecasting as it is capable of handling forecasts over a longer horizon.

Table 6 shows the model evaluation comparing the MAPE values for the model with and without lags.

Table 6

Linear Regression without Lags Evaluation

		Linear Regression(Lag 1+Weekly Ind)	Linear Regression Without Lag	
S/No	Product Region	MAPE	MAPE	SMAPE
1	Alpha-a	16.69%	8864.75%	94.54%
2	Alpha-b	14.65%	13635.60%	97.04%
3	Beta-a	3.95%	14.02%	6.28%
4	Beta-b	1.82%	6.05%	3.15%
5	Beta-c	2.48%	19.63%	8.80%
6	Delta-a	2.21%	3.12%	1.56%
7	Delta-b	1.56%	11.56%	5.43%
8	Delta-c	3.72%	8.40%	4.57%
9	Epsilon-b	4.35%	14.63%	8.05%
10	Epsilon-d	2.66%	8.36%	4.12%
11	Gamma-a	1.74%	4.72%	2.30%
12	Gamma-b	2.05%	9.11%	4.84%
13	Gamma-c	4.86%	13.24%	7.39%
	Avg MAPE	4.83%	1739.48%	19.08%
	Average MAPE without Alpha-a and Alpha-b	2.85%	10.26%	5.13%

As expected, the average MAPE value of the model without lags is worse than the model with lags. It is also observed that the MAPE values for the zones Alpha-a and Alpha-b perform very poorly. A performance evaluation metric more suitable to these kinds of abnormal results, called the Symmetrical Mean Absolute Percentage Error (SMAPE), is introduced.

One shortcoming of MAPE is that it is asymmetric. It penalizes negative errors more heavily than positive ones. This is because the percentage error cannot exceed 100% for forecasts that are too low, while there is no upper limit for the forecasts that are too high. Due to this, models that under-forecast are favored by MAPE over models that over-forecast. SMAPE fixes this shortcoming by equally treating over and under forecasting. It has lower and upper bounds of 0% and 200% respectively. The non-intuitiveness of this range is a flaw of this metric (Lewinson, 2020). It is observed that discounting the two outlying regions, Alpha-a and Alpha-b, the MAPE and SMAPE values show very good performance.

A valuable insight obtained from the analysis, was the weekly seasonality observed during the time series decomposition. This is relevant for the elastic pricing model to be developed.

4.3 DEMAND SEASONALITY

Table 7 shows the seasonality in all the zones. Figures in parentheses represent negative values.

Table 7

Weekly Seasonality

		Weekly Seasonality						
Region-Zone	Mean Demand	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Alpha-a	5807	(355)	156	107	(37)	265	146	(283)
Alpha-b	5643	(278)	182	83	(22)	198	144	(306)
Beta-a	1053	(32)	(1)	37	26	22	(14)	(38)
Beta-b	4706	33	(3)	18	19	(2)	(37)	(28)
Beta-c	3867	4	(0)	15	17	10	(16)	(30)
Delta-a	5026	(8)	26	7	24	(24)	0	(26)
Delta-b	6756	9	(19)	45	36	(25)	(10)	(34)
Delta-c	3272	6	33	3	29	15	(26)	(61)
Epsilon-b	5856	71	65	(47)	(23)	119	(70)	(115)
Epsilon-d	3736	(26)	33	11	18	88	(59)	(65)
Gamma-a	40062	130	450	782	681	553	(1090)	(1506)
Gamma-b	30687	153	394	345	338	230	(619)	(840)
Gamma-c	8735	23	102	75	149	146	(231)	(264)

It is observed that across most zones, the demand drops over the weekend and picks up at the start of the week on Mondays. To optimize the capacity better throughout the week, a dynamic pricing model is proposed to incentivize customers to move some workloads to the weekends.

5. DISCUSSION

Time series forecasting is a widely used statistical approach for predicting future values based on historical data. In this study, a time series analysis was carried out on a dataset to forecast demand for a particular cloud product, revealing linear trends and seasonality on a weekly basis. In this discussion, the focus is on how this information can be utilized for capacity planning, capacity optimization, and revenue management to maximize profits and enhance the efficiency of the supply chain.

5.1 CAPACITY PLANNING

Capacity planning refers to the process of determining the optimal production or service capacity that an organization should maintain to meet anticipated demand. Given the linear trend and weekly seasonality observed in the dataset, it is crucial to incorporate these patterns into capacity planning to ensure the proper allocation of resources.

Linear Trend: The linear trend indicates that demand is gradually increasing over time, suggesting the need for a scalable service delivery plan that can accommodate this growth. Companies should consider investing in infrastructure and technology that allow for efficient expansion of service capacity. Regularly reviewing and updating capacity plans can also help organizations respond to changes in demand patterns and minimize the risk of over- or under-investment in capacity.

Weekly Seasonality: Since demand exhibits a weekly pattern, capacity planning should account for fluctuations in demand throughout the week. For example, additional resources may be allocated during peak demand periods, while capacity may be reduced during off-peak periods. This can help companies improve customer satisfaction and reduce carrying costs.

5.2 CAPACITY OPTIMIZATION

Capacity optimization refers to the process of using existing resources more efficiently to meet demand. By leveraging the linear trend and seasonality information, organizations can make informed decisions on resource allocation and scheduling to improve overall efficiency.

Linear Trend: As demand grows steadily, organizations should seek ways to increase the efficiency of their service processes. This can be achieved through continuous improvement programs, such as DevOps practices. These practices, tools, and processes enhance an organization's ability to provide applications and services quickly and efficiently, surpassing traditional software development and infrastructure management processes. The practices focus on eliminating waste and reducing variability in processes. Additionally, investing in research and development to create more efficient service methods can help companies stay competitive in the market.

Weekly Seasonality: By understanding the weekly fluctuations in demand, organizations can adjust their service schedules and resource allocation accordingly. This may involve scheduling maintenance and downtime during off-peak periods, or assigning extra resources during periods of high demand. By aligning service schedules with demand patterns, companies can reduce costs associated with idle time and infrastructure maintenance, while ensuring a steady supply of resources for customers.

5.3 REVENUE MANAGEMENT

Revenue management involves the strategic use of pricing, promotions, and inventory control to maximize revenue. The linear trend and seasonality observed in the dataset provide valuable insights for creating effective revenue management strategies.

Linear Trend: The steady increase in demand suggests that the market may be willing to accept higher prices over time, allowing organizations to explore price increases or the introduction of premium products. Implementing dynamic pricing strategies, such as adjusting prices based on real-time demand information, can help companies maximize revenues while maintaining customer satisfaction.

Weekly Seasonality: Seasonal fluctuations in demand provide opportunities for targeted promotions and pricing strategies. For instance, companies can offer discounts or promotional bundles during off-peak periods to stimulate demand and reduce excess inventory. Alternatively, premium pricing can be implemented during peak demand periods to capitalize on higher customer willingness to pay. By aligning pricing and promotion strategies with demand patterns, companies can optimize revenue generation and better manage their inventory levels.

Future work on revenue management could adopt the model developed by Püschel and Neumann, 2009, to utilize and allocate idle capacity for non-contract jobs. They recommend using a dynamic pricing policy based on different utilization levels. They propose a mathematical function to represent the policy:

$$\max_{x} \sum_{1}^{j}\left(\frac{1}{2} * x_j\right)^{j} \; \forall j \in J \tag{5}$$

Subject to:

$$\sum_{1}^{j} c_{jr}(t) * x_j \leq c_r(t) \; \forall t \in T, \forall r \in R \tag{6}$$

$$\left(1 - H_1\left(p_j - p_{p1}\right)\right) * \frac{\sum_{1}^{j} c_{jr}(t) * x_j}{c_r(t)} \leq l_{p1} \forall t \in T, \forall r \in R \tag{7}$$

Where *T* = set of all regarded timeslots

J = set of available jobs

R = set of all resource types

x_j = binary allocation for acceptance or rejection of job j

$c_{jr}(t)$ = capacity required by job j in timeslot t

$c_r(t)$ = capacity available for resource type r during timeslot t

The objective function in (5) is a generating function representing the sequential nature of the policy. Equation (6) is the capacity constraint ensuring that the model does not allocate more capacity of a resource type than is available. Equation (7) is the utilization-based pricing based on a certain reservation price p_{p1} to be achieved when utilization surpasses threshold l_{p1}. The variable p_j represents the unit price and timeslot for job j. $H_1(n)$ is the Heaviside step function. $H_1(n)$ is 0 for $n<0$ and 1 for $n\geq 0$.

The model is designed to maximize the revenue with respect to the sequence of workload order requests, requested workload, and utilization period.

The time series forecasting of demand, revealing a linear trend and weekly seasonality, provides valuable insights for capacity planning, capacity optimization, and revenue management. By incorporating this information into their strategic decision-making processes, organizations can better allocate resources and improve the efficiency of their supply chains, leading to increased profitability and customer satisfaction. It is crucial for companies to regularly review and update their capacity plans and service schedules to respond to changing demand patterns and market conditions. By leveraging advanced statistical techniques, such as time series forecasting, companies can gain a competitive advantage in their industries and achieve long-term success. Further research can explore other variables and factors, such as, industry trends, regulation, cost, and integration, which may affect demand, and how they can be incorporated into capacity planning and revenue management strategies. Overall, organizations should strive to continuously improve their operations and stay adaptable to changes in the market to ensure sustained growth and profitability.

6. CONCLUSION

In this project, several classical time series forecasting models were analyzed to determine the best model for forecasting demand for cloud usage by an Infrastructure as a Service (Iaas) provider. The project set out to meet the sponsor company's objective of developing a robust, verifiable and repeatable forecasting methodology to forecast demand to help with capacity planning, capacity optimization and revenue management. The company aims to maintain a competitive edge in the market by always meeting customer service level requirements. They also seek to maximize revenue by maintaining capacity to meet existing demand and future growth.

The Linear model with weekly seasonality performed best and gave the highest average forecast accuracy with a Mean Absolute Percent Error (MAPE) of 19%. From the results, the positive linear trend indicates growth projections in the medium to long term. On a weekly basis, however, seasonality is observed. The management will have to make several decisions to manage and optimize this observed trend and seasonality.

As the data analysis shows that the demand for cloud services is increasing steadily over time, the company should consider adjusting their pricing strategy to capture the increasing value of their service. This could involve raising prices or introducing new pricing tiers. They should also plan for capacity expansion to ensure that they have the necessary infrastructure to handle the growing demand in the future. As well, they should improve customer support, as more customers are using the cloud service, which could lead to an increase in support requests. The company should ensure that they have adequate customer support resources to handle the growing customer base.

On the weekly seasonality observed, the company should use this information to optimize the allocation of their resources, such as server capacity, bandwidth, and other infrastructure components, to match the weekly pattern of demand. As some days of the week have lower demand than others, the company could offer promotions or discounts to incentivize customers to use the service on these slower days.

The company should review their Service Level Agreements (SLAs) to ensure they align with the linear trend and weekly seasonality in demand. This would help to set realistic expectations with customers and ensure that they receive the level of service they require during peak times.

For future research, more distinguishing features of the zones can be obtained to enable more advanced machine learning models to be tested on the data. This would provide some more specific insights to improve the forecasting and drive more informed managerial decisions. As well, more usage data should be obtained for the late adopted zones which did not perform as well, and the analysis revisited. Further analysis done with more data will provide more robust and reliable forecast results for these zones.

Price sensitivity analyses would also be required to determine the elasticity. This is a useful input into the dynamic pricing model developed to manage capacity and revenue. Slight variations should be done on the existing standard prices and the impact on demand measured and used to compute the elasticity.

The project helps in assessing and identifying a suitable forecasting model which is useful for capacity planning. An elastic pricing model is also recommended for capacity optimization and revenue management. In general, the findings and recommendations are applicable in planning for capacity to optimize company assets, manage revenue generation and maintain customer satisfaction by meeting the service levels.

7. REFERENCES

Baldan, F. J., Ramirez-Gallego, S., Bergmeir, C., Benitez-Sanchez, J. M., & Herrera, F. (2018). A Forecasting Methodology for Workload Forecasting in Cloud Systems. *IEEE Transactions on Cloud Computing*, 6(4), 929-941. https://doi.org/10.1109/tcc.2016.2586064

Bergmeir C., & Benítez, J. M. (2012). Neural networks in R using the Stuttgart neural network simulator: RSNNS. *Journal of Statistical Software*, *46*(7). https://doi.org/10.18637/jss.v046.i07

Bhardwaj, S., Jain, L., & Jain, S. (2010). Cloud computing: A study of infrastructure as a service (IAAS). *International Journal of engineering and information Technology*, 2(1), 60-63. https://www.academia.edu/1181740/Cloud_computing_A_study_of_infrastructure_as_a_service_IAAS_?auto=citations&from=cover_page

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brownlee J. (2021, October 12). A Gentle Introduction to the BFGS Optimization Algorithm. *Machine Learning Mastery Blog*. https://machinelearningmastery.com/bfgs-optimization-in-python/

Brownlee J. (2020, December 10). How to Decompose Time Series Data into Trend and Seasonality. *Machine Learning Mastery Blog*. https://machinelearningmastery.com/decompose-time-series-data-trend-seasonality/

Buyya, R., Broberg, J., & Goscinski, A. (2011). Cloud Computing: Principles and Paradigms. John Wiley & Sons, Inc. https://onlinelibrary-wiley-com.libproxy.mit.edu/doi/pdf/10.1002/9780470940105

Buyya, R., Yeo C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud Computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*(6). https://ieeexplore-ieee-org.libproxy.mit.edu/stamp/stamp.jsp?tp=&arnumber=7501816

Chatfield, C. (1978). The Holt-Winters Forecasting Procedure. *Journal of the Royal Statistical Society*. Series C (Applied Statistics), *27*(3), 264-279. https://doi.org/10.2307/2347162

Clarke, M. (2021, March 13). *How to perform time series decomposition*. Practical Data Science https://practicaldatascience.co.uk/machine-learning/how-to-perform-time-series-decomposition

Gallo, A. (2015, August 21). *A refresher on price elasticity*. Harvard Business Review. https://hbr.org/2015/08/a-refresher-on-price-elasticity

Gordon, J. (2022). *What is an Autoregressive Moving Average?* The Business Professor, https://thebusinessprofessor.com/en_US/research-analysis-decision-science/autoregressive-moving-average-arma-definition

Hyndman, R. J., Athanasopoulos G. (2021). *Forecasting: Principles and Practice* (3rd ed.). Otexts Publishing. https://otexts.com/fpp3/

Konstanteli, K., Cucinotta, T., Psychas, K., & Varvarigou, T. A. (2014). Elastic admission control for federated cloud services. *IEEE Transactions on Cloud Computing*, *2*(3). https://ieeexplore-ieee-org.libproxy.mit.edu/stamp/stamp.jsp?tp=&arnumber=6847184

Lewinson, E. (2020, November 1). *Choosing the correct error metric: MAPE vs. sMAPE*. Towards Data Science. https://towardsdatascience.com/choosing-the-correct-error-metric-mape-vs-smape-5328dec53fac

Lin, G. (2023, March 16). *Cloud Pricing Comparison: AWS vs. Azure vs. Google Cloud Platform in 2023*. Cast AI. https://cast.ai/blog/cloud-pricing-comparison-aws-vs-azure-vs-google-cloud-platform/

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. *National Institute of Standards and Technology*, *152*, 29–41. https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf

Oracle Nigeria. (2022). *Cloud Computing*. https://www.oracle.com/ng/cloud/what-is-cloud-computing/

Peixeiro M. (2022). *Time series forecasting in Python*. Manning Publications. https://www.manning.com/books/time-series-forecasting-in-python-book

Roy, N., Dubey, A., & Gokhale, A. (2011, July). *Efficient autoscaling in the cloud using predictive models for workload forecasting* [Conference Session]. IEEE 4th International Conference on Cloud Computing, Washington D.C., United States. https://ieeexplore-ieee-org.libproxy.mit.edu/abstract/document/6008748

Ruparelia, Nayan B., 'Introduction', Cloud Computing (Cambridge, MA, 2016; online edn, MIT Press Scholarship Online, 22 Sept. 2016), https://doi.org/10.7551/mitpress/9780262529099.003.0001

Seabold, J., Skipper, C., & Perktold, J. (2010) “Statsmodels: Econometric and Statistical Modelling with Python” *In* Proceedings of the 9th Python in Science Conference. https://efaidnbmnnnibpcajpcglclefindmkaj/https://conference.scipy.org/proceedings/scipy2010/pdfs/seabold.pdf

Spot. (n.d.). *What are Azure Spot Virtual Machines?* Spot. Retrieved April 7, 2023, from https://spot.io/resources/azure-pricing/what-are-azure-spot-virtual-machines/

Taylor, S. J., & Letham, B. (2017, February 23). Prophet: forecasting at scale. *Meta Blog*. https://research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/

VMWare.Inc. (2009). *Virtualization Overview* [White Paper]. VMWare.Inc. https://www.vmware.com/pdf/virtualization.pdf

VMware. (n.d.). *What is a Virtual Machine?* VMware. Retrieved April 10, 2023, from https://www.vmware.com/topics/glossary/content/virtual-machine.html

8. APPENDIX

Appendix A

Virtual Machine Pricing

Table A

Pricing for Cloud General Purpose Specifications

Cloud Provider	Instance type	vCPU	RAM (GB)	Price per hour $
AWS	t4g.xlarge	4	16	0.134
Azure	B4ms	4	16	0.166
Google Cloud	e2-standard-4	4	16	0.151
Oracle	VM.Standard3.Flex	4	16	0.184
Average Price				0.159

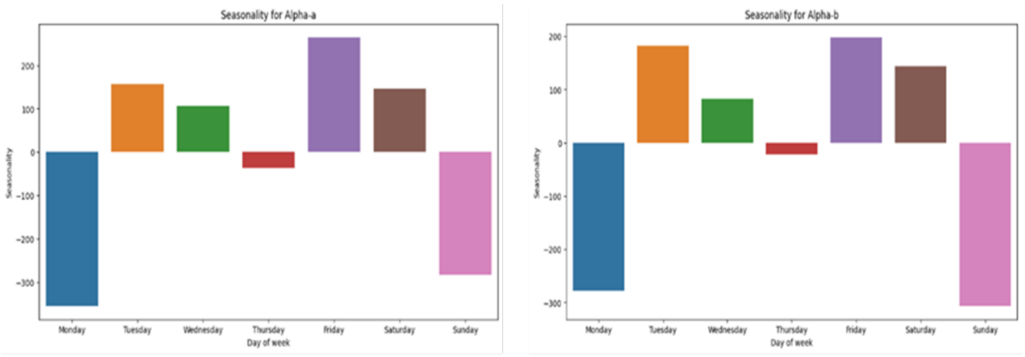
Note. Adapted from *Cloud pricing based on On-Demand rates* by L. Gil, 2023, Cast AI, https://cast.ai/blog/cloud-pricing-comparison-aws-vs-azure-vs-google-cloud-platform/

Appendix B

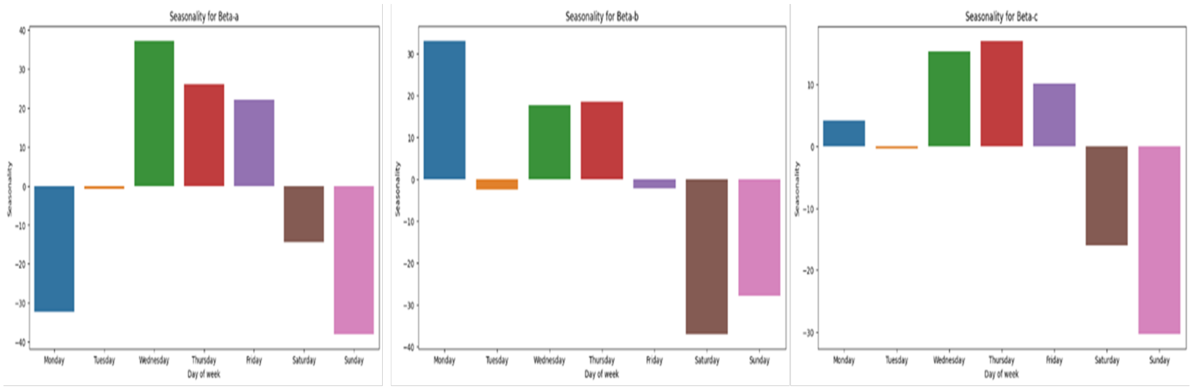
Zone Seasonality

Figure B

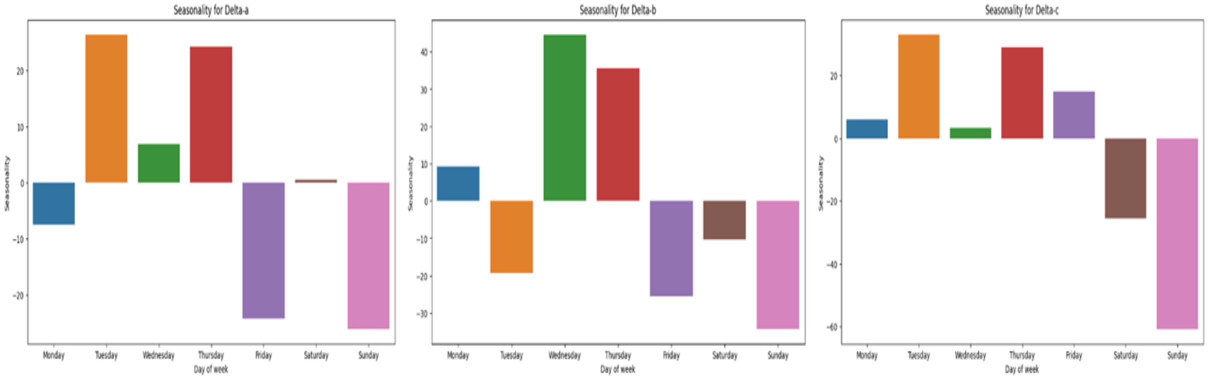
Zones Weekly Seasonality Plots



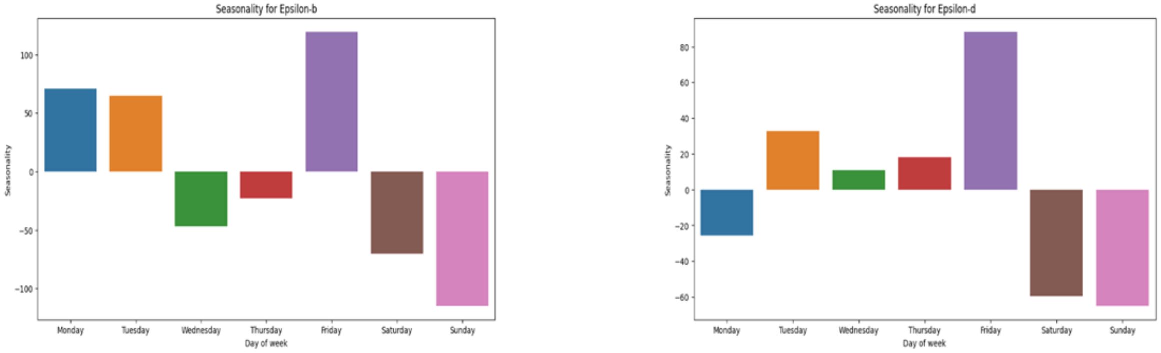
Alpha Zones a & b



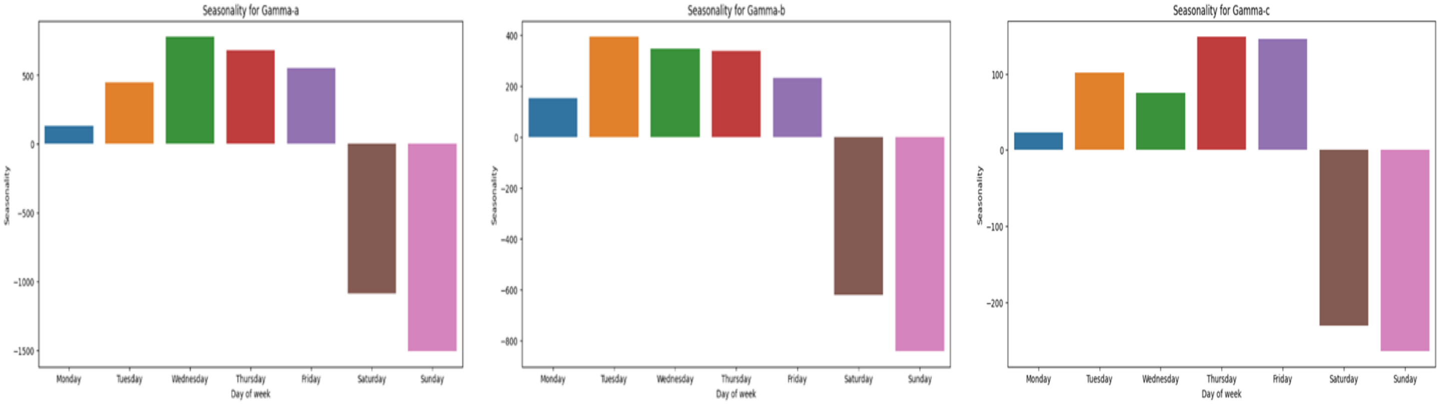
Beta Zones a, b & c



Delta Zones a, b & C



Epsilon Zones b & d



Gamma Zones a, b & c