HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

POSTGRADUATE PROGRAM

WATER CONSUMPTION PREDICTION USING MACHINE LEARNING THE CASE OF HAWASSA CITY WATER SUPPLY AND SEWAGE SERVICE ENTERPRISE

A MASTER'S THESIS

BY **MUSE KEBEDE MULATU**

OCTOBER 2024

HAWASSA, ETHIOPIA

HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

POSTGRADUATE PROGRAM

WATER CONSUMPTION PREDICTION USING MACHINE LEARNING: THE CASE OF HAWASSA CITY WATER SUPPLY AND SEWAGE SERVICE ENTERPRISE

A MASTER'S THESIS

BY MUSE KEBEDE MULATU

OCTOBER 2024

HAWASSA, ETHIOPIA

WATER CONSUMPTION PREDICTION USING MACHINE LEARNING: THE CASE OF HAWASSA CITY WATER SUPPLY AND SEWAGE SERVICE ENTERPRISE

MUSE KEBEDE MULATU

A THESIS SUBMITTED TO HAWASSA UNIVERSITY,
INSTITUTE OF TECHNOLOGY,
DEPARTMENT OF COMPUTER SCIENCE,
SCHOOL OF GRADUATE STUDIES,
HAWASSA, ETHIOPIA

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

MAIN ADVISOR: DEGIF TEKA (PH.D.)
CO-ADVISOR: MR. EPHREM ALAMEREW

OCTOBER 2024

DECLARATION

I hereby declare that this thesis's work entitled "Water Consumption Prediction using Machine Learning: The case of Hawassa City Water Supply and Sewage Service Enterprise" has been carried out by Muse Kebede Mulatu in the Computer Science Department/School of Informatics, and prepared under the guidance and supervision of my advisor Degif Teka (Ph.D.), all sources of materials used for the thesis have been duly acknowledged. I further confirm that the thesis is my original work and has not been submitted in part or full to any other higher-learning institution to earn any degree.

Name: Muse Kebede Mulatu Signature: _____ Date: Dec 04. 2024

I have given my approval as the thesis advisor for this MSc thesis to be submitted for examination.

Name: Degif Teka (Phd)

Signature: _____

Place and Date of Submission: Dec 04/2024

ADVISOR APPROVAL SHEET

This is to certify that the thesis entitled "Water Consumption Prediction using Machine Learning: The case of Hawassa City Water Supply and Sewage Service Enterprise" submitted in partial fulfillment of the requirements for the degree of Master's with specialization in Computer Science, the Graduate Program of the Faculty of Informatics and has been carried out by Muse Kebede Mulatu, ID. No "GPCoScw/0029/12", under our supervision. Therefore, we recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the department.

Dereje Teka (PhD)		Dec 04/2024
Name of Main Advisor:	Signature	Date
Ephrem A.		DC-04-2024
Name of Co-Advisor:	Signature	Date

EXAMINORS' APPROVAL SHEET

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY EXAMINORS' APPROVAL SHEET

We, the undersigned, members of the Board of Examiners of the final open defense by "Muse Kebede Mulatu", have read and evaluated his thesis entitled "Water Consumption Prediction using Machine Learning: The case of Hawassa City Water Supply and Sewage Service Enterprise", and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirements for the master's degree in computer science.

Defif T. (PhD)		Dec 4, 2024
Name of Major Advisor	Signature	Date
Ephrem A.		DEC-04-2024
Name of Co-advisor	Signature	Date
Andargachew Mekonnen (PhD)		Dec. 4. 2024
Name of Internal Examiner-I	Signature	Date
Mekonnen Kassie		DEC 4, 2024
Name of Internal Examiner-II	Signature	Date
Asrat Mulatu (Ph.D.)		25/11/2024
Name of External Examiner	Signature	Date
SGS Approval	Signature	Date

Final approval and acceptance of the thesis is contingent upon the submission of the final copy of the thesis to the School of Graduate Studies (SGS) through the Department/School Graduate Committee (DGC/SGC) of the candidate's department.

Stamp of SGS Date: _____

Department of Computer Science

ACKNOWLEDGMENT

First and foremost, I would like to thank my almighty GOD for giving me the strength, and ability to learn, understand, and conclude my thesis report. I would not have been here without His grace. My faith has been a source of comfort and resilience, and I am truly blessed to have His presence in my life.

I would like to express heartfelt gratitude to my advisor, Degif Teka (Ph.D.), and co-advisor, Mr. Ephrem Alamerew (Informatics Faculty), for their generous support, guidance, comments, and valuable feedback on my work. Their insight and experience have been invaluable in shaping my research and academic growth. My utmost gratitude also goes to the community of the Department of Computer Science of Hawassa University Institute of Technology for the staff who have enriched my experience and inspired me to strive for excellence.

It is also a great pleasure to acknowledge my deepest thanks and gratitude to Sidama regional government Hawassa City water supply and sewage service Enterprise. IT officer Mr. Thomas Agaro for facilitating and providing the raw data used in the study.

Finally, I would like to thank my dad (Dr. Kebede Mulatu), mom (Workitu Gudisa), and the rest of my family and friends for their endless support especially Cherinet and Edemealem Desalegn for the encouragement they have given me in both good and bad times during my master's study. Their belief in me has been a constant motivator and I could not have accomplished this without them.

Table of Contents

List of Figures

List of Tables

List of Equations

LIST OF ABBREVIATIONS AND ACRONYMS

ACF	Autocorrelation Function
AMRs	Automatic meter readers
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
CSV	Comma Separated Values
DMA	District Metered Areas
EDA	Exploratory data analysis
E.C	Ethiopian Calendar
GTP	Growth Transformation Plan
GDP	Growth Domestic Product
HCWSSSE	Hawassa City Water Supply and Sewerage Service Enterprise
LR	Linear Regression
MAE	Mean absolute error
MSE	Mean Squared Error
ML	Machine Learning
MLP	Multilayer Perceptron
R^2	R-squared
RF	Random Forest
SVM	Support Vector Machine
SVR	Support Vector Regressor
RMSE	Root Mean Square Error
SRS-HCWSSSE	Sidama Regional State Hawassa City water supply and sewage service
WDN	Water Distribution Network
WEKA	Waikato Environment for Knowledge Analysis

ABSTRACT

Proper management of water consumption ensures a better clean and healthy community. Therefore, predicting water consumption gives time to prepare and protect the community from unseen natural or unknown disasters. Previous studies have implemented many prediction models in specific areas that showed promise but were not applicable in developing countries. The study was conducted to develop a prediction model for water consumption for the Hawassa City Water Supply and Sewerage Service Enterprise (HCWSSSE), a city in the Sidama region, Ethiopia. The enterprise experienced water shortages due to its way of prediction solely based on the previous month's consumption rate and needed to consider seasonal changes. The models developed in the study use machine learning techniques on five-year Monthly Consumption data from 2009-2015 E.C of the Ethiopian budget year, with around 16012 data points, and modeled by training 80%, validating 10%, and testing 10%. This study explores the application of various machine learning algorithms including Random Forest (RF), Support Vector Regressor (SVR), Linear Regression (LR), and XGBoost for predicting. The performance of models was evaluated using key error evaluation metrics Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). For the Models, their R^2 rates for training, validation, and testing were Random Forest (RF) 97.23%, 97.24%, and 97.22%, Linear Regression (LR) 78.18%, 78.38%, and 77.98%, Support Vector Regressor (SVR) 79.37%, 79.92%, and 78.81% and XGBoost 97.08%, 97.07%, and 97.08% respectively. The Random Forest (RF) and XGBoost showed promise in prediction, they demonstrated effectiveness in handling complex datasets. Specifically, Random Forest (RF) offered better predictions with reduced risk of overfitting. The successful application of RF and XGBoost highlights the importance of leveraging machine learning for sustainable water management in an era of growing demand and climate variability.

Keywords: *Water Consumption, Prediction, Monthly, Machine learning (ML), Regression, Hawassa City Water Supply and Sewerage Service Enterprise (HCWSSSE)*

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Water is one of the basic consumable products essential in our daily lives. Consumption of Clean water heavily influences public health and living standards. A general rule is that a person can only survive about 3 days without water[1]. Public utilities and private water companies are under increasing pressure to address water availability. Every day, institutions like the Sidama National Regional State Hawassa City water supply and sewage service enterprise (SRS-HCWSSSE) are tasked with allocating water to the community. Water must be guaranteed 24/7 [2]. Natural causes, pollution, population growth, droughts, expansion of different businesses, unplanned developments, and human factors increase water scarcity in many areas of the world[3]. To manage this, there has to be a balance between the demand for water and the supply. Prediction of water demand helps in managing water supply [4]. It is a costly feat as the operation and allocation are considerably high in the city's management procedures.

Ethiopia for the last thirty years [5] faced many droughts and challenges being a landlocked country, and has been affected by a lack of clean water in some areas. It's estimated a fifth of the world's population lives in areas of physical water scarcity, where there is not enough water to meet all demands[6]. Ethiopia consists of nine major rivers and twelve big lakes with a potential renewable freshwater of about 122 billion m^3 but only 3% stays in the country. Currently, 5% is estimated to be used for consumption purposes even though it's estimated that 54.5 billion m^3 of surface runoff and 2.6 billion m^3 of groundwater can be developed for utilization. The average consumption per capita:

- In rural areas it is 15 liters/per capita/a day in a service radius of 1.5km
- In urban areas it is 25 liters/ per capita/a day in a service radius of around 0.5km

While the country has abundant water there is a concern about water stress as the population growth over the last decade has increased in number since 2010 which was around 87 million and in 2023, it's estimated to be approximately 123 million and still increasing [7] (accessed date 20th May 2023).

Drinking water is essential for life, and having adequate and constant access to water is necessary. The Medindia Medical Review[8] stated humans consist of 60-70% water and must hydrate themselves to avoid the signals of thirst. Thirst is a warning signal that you are dehydrated and must consume water. Your body would have already lost about 1% by then. To put it into context 22-30% of water loss would lead you to death or into a coma.

On a short-term basis, water consumption depends on the water end user behaviors, socio-demographic, property characteristics, and psychosocial constructs habits of the consumer, business, institutions, and weather conditions[6]. The introduction of machine learning (ML) advanced modeling tools has made them suitable for predicting water consumption more accurately. It is preferred as it can handle complex relationships within large datasets[9]. By employing algorithms like regression analysis, decision trees, and neural networks, researchers can enhance the predictability of water consumption. However, despite the potential benefits of these approaches, implementation in real-world scenarios opens new gaps in the selection of relevant features, model interpretability, and integration into existing water management systems and water distribution centers [10].

The integration of machine learning (ML) in water consumption sectors allows for the analysis of real-time water consumption data and leads to a robust and dynamic prediction model[11]. For instance, models like the Random Forest (RF) and Support Vector Regressor (SVR) have outperformed classical approaches in predicting residential water use[10]. This not only improves the ability to build models but enables water utility companies to improve their methods of water conservation and management efforts. A reliable water consumption model depends on the water utilities center allowing efficient and manageable water supply, recording stored water, and related equipment.

Let's take the case of the COVID-19 pandemic, caused by the SARS-COV-2 coronavirus. It was estimated that around 367,857+ cases are publicly known based on the Ethiopian Health Minister Bureau[7] (accessed date 20th Jan 2024). The literature on "Impact of the COVID-19 pandemic on water consumption behavior" [12] explains the impacts on water consumption in the case of Brazil while COVID-19 hit the country. During such a contagious disease, constant water distribution was mandatory and it was highly challenging to meet the necessary demands. Hygiene habits have been influenced by washing hands with soap and water, cleaning and sanitizing floors, and

cleaning while handling food with the utmost care to reduce the number of positive cases. Even though not a major part of this study, it's a natural disaster that can cause a major change in water consumption as the need for sanitation becomes high in contagious diseases. Prediction of domestic water importance is vital for water utility companies [4]. With the rising prevalence of such cases, collecting high-resolution data from individual users provides a large corpus of data on which predictive models can be based.

While expressing water consumption and water availability many factors can be the cause. In the study, the researchers aim to predict water consumption within an area (the case of Hawassa City). Hawassa city was known in the time of emperor Menilik II (1889-1913 E.C), but historians state the founder was Ras Mengesha Seyoum as his impact in petitioning Emperor Haile Selassie to change the capital from Yirgalem to Hawasaa for the formerly known as the south region but now as the Sidama region in 1954 [13].

In Ethiopia, water consumption is recorded by water meters and are locally recorded on automatic meter readers (AMRs) locally called "Kotari (Amharic: ቆጣሪ)" or" Wuha Kotari (Amharic: ውሃ ቆጣሪ)". These AMRs are mechanical counters that count forward and unless the history data of the last record is known the current data is the recorded data on the meter which might be years in the making. Water meters in different countries are not digitized, even though AMRs are not digitized, for the sake of billing customers' water utilities collect data from AMRs every month near real-time water consumption.

Consumable water scarcity is one factor that increases or decreases water usage in an area. The seasonal aspects also influence water availability. There are 4 main seasons in Ethiopia: The spring season known as "Tsedey (Amharic: ጸደይ)" (months from September to November), the dry period or winter also known as "Bega (Amharic: በጋ)" (months from December to February), the partial rainfall or fall also known as "Belg (Amharic: በልግ)" (months are from March to May), and the long rainy season summer also called "Keremt (Amharic: ክረምት)" (months from June to September)[13][14]. Here we can observe in Ethiopia unlike the Western world the estimation in the rainy seasons like Keremt and Belg has an annual rainfall of 50-70% and 20-30% rainfall annually[15]. The highest amount of rainfall occurs in the month of June-September while the highest is recorded in August.

Water scarcity is a pressing concern in urban areas, necessitating efficient water supply and sewage management strategies. Predicting water consumption plays a crucial role in addressing this challenge, particularly in cities like Hawassa. By utilizing machine learning techniques, we can analyze the factors to accurately predict water consumption trends in the study city Hawassa City. This Study aims to conduct a comparative analysis of various machine learning techniques for water consumption prediction and evaluate their effectiveness and potential use in water utility companies. By examining existing literature and use-case studies, we aim to identify methods and models suitable for best practices and future research direction. The relationship between demographic factors and water usage patterns ultimately contributes to sustainable water supply and sewage management practices in urban environments.

1.2 Statement of the Problem

Managing urban city resources is difficult, as rapidly urbanizing populations increase pressure on water resources and demand consistent flow. Water consumption is overlooked though necessary plans should be developed as it is a critical challenge for effective water management. Accurate prediction of water is essential for knowing what is currently available and the possible outcomes for the city administrator (municipality)[16].

Making our water distribution network (WDN) truly smart is not only a challenging task, but also shows great opportunity for innovations and new business of smart metering, monitoring, data acquisition, data management, advanced analysis, communication, and automation control[17]. Traditional prediction methods rely on simple models, that fail to capture the complex and non-linear relationships among different influencing factors, such as population growth, seasonal variations, economic activity, and climatic conditions[3]. Consumable water prediction is based on experience, through the challenges water utilities faced. Even though it has been practiced in the past there are significant inefficiencies in water distribution, increased operational costs, and heightened vulnerability to shortages during peak demand periods.

To the best of the researcher's knowledge water consumption in the case of Hawassa City has not been attempted for prediction modeling with machine learning (ML) methods, even though there has been research works conducted on water consumption elsewhere having an accurate model for predicting both short-term and long-term for both the operational and planning aspects which is

essential for the city's modernization and development of water management facilities. To address this the study aims to develop a machine learning (ML) model for predicting water consumption and enhancing the accuracy of water consumption prediction.

To calculate the water consumed in a household, we can consider variables like the total number of people, bathrooms, laundry, outdoor water use, washing dishes, and many more[18]. Such variables can be overwhelming and, in some cases impossible to comprehend. Listing all the variables is one too many in this study, we are not going to be focusing on all the individual variables but on the overall consumption.

Machine Learning (ML) offers a promising approach by reducing the error rate and increasing the accuracy of reliable water consumption predictions. Despite their potential, there are several challenges like feature complexity, as relevant factors such as population density, temperature, and seasonal variation cause fluctuation in water usage[19]. Additionally, the quality of the dataset is essential for effective model training, but real-world data exhibits containing outliers and hinder performance [20]. Furthermore, Machine learning algorithms show varying strengths; for instance, Random Forest (RF), Linear Regression (LR), Support Vector Regressor (SVR), and XGBoost handle non-linear relationships well, have the potential to improve predictive accuracy by incorporating a diverse range of features and adapting to changing conditions [21].

In an age of increasing resource demand and environmental uncertainty, this research seeks to provide valuable insights for water utility enterprises like Hawassa City Water Supply and Sewerage Service Enterprise (HCWSSSE) and policymakers to enhance their prediction abilities, manage resource allocation, and implement efficient conservation strategy for better water management strategies.

1.3 Research Questions

The study aims to answer the following research questions, provide a useful tool, and investigate the research questions.

1. How to build a machine learning model for water consumption prediction?
2. Which features are important for a better prediction model of water consumption in urban areas?
3. Which machine learning algorithms perform better for water consumption prediction?

1.4 Objectives

1.4.1 General Objective

The general objective of this study is to build monthly predictive models for water consumption using a machine-learning method for the HCWSSSE dataset and propose a better model.

1.4.2 Specific Objectives

The specific objectives of this study are as follows:

- To review related literature for a better understanding of the research problem and research design
- Perform data preparation tasks on the data for the correctness of the model-building process.
- Identify variables and features that are important to predict water consumption.
- To design and develop a suitable model based on identified techniques for the research questions.
- Train and test the model and select the best-performing result.
- Evaluating the model using the testing options.
- Comparison of the best-performing models.

1.5 Scope and Limitation of the Thesis

1.5.1 Scope of the Thesis

The scope of this study focuses on water consumption from the data source of the HCWSSSE dataset, and the city of Hawassa is used as the case study. The data that is used is clean consumable water provided by the enterprise. The data is gathered monthly in the Ethiopian budget year, and the data acquisition is constrained by the presented numerical value. In investigating the data, primary data is time-consuming as the collection is every month so the study used secondary data provided by the water agency (HCWSSSE).

1.5.2 Limitation of the Thesis

In the study, the data that is used is from the HCWSSSE the data that is collected is from one source, and the findings might not directly be applied to other regions with different geographics and climate conditions. Data either from leakages and damage to pipelines or mismanagement are

prevalent. Also mentioned in the scope of the thesis the data collected is every month and the information of daily water consumption is rounded up into a monthly aspect. The day-to-day consumption record from the current setup of AMRs is nearly impossible and very costly to achieve. Therefore, the amount of consumption of water daily is not studied.

1.6 Significance of the Study

The result of this study is believed to have a benefit in urban city development and be an input for the municipality in Hawassa in their water prediction plans. It will also benefit other researchers in different fields in the use of prediction modeling especially in machine learning, and how it was tackled in this study. The possible application of the model will allow the HCWSSSE to allow the prediction of consumable water and give an output for the next month's consumption. It will also give insight to other developing cities or countries into the importance of water consumption prediction and the need for machine learning and other methods to be implemented in other future work projects.

1.7 Organization of the Rest of the Thesis

The study is organized as follows:

Chapter One describes the introductory part, as well as the background for the study, statement of the problem, objectives, research questions, significance of the study, scope, and limitations.

Chapter two reports relevant literature that sheds more light on water consumption and the need for prediction using machine learning techniques. It reviews related literature.

Chapter three describes the methodological approach, data collection and analysis, and the tools used.

Chapter four explores the experimentation and briefly explains the results that were attained in the study.

Chapter five shows the results and gives a brief discussion of how the results were attained.

The last chapter, chapter 6 deals with the conclusion and the future works that this thesis could continue on and for other researchers to explore as well.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter reports a review of literature that briefly describes water consumption and the benefits of having a prediction model in urban city planning. Understanding consumption patterns is crucial as the total amount of water an individual uses is referred to as water consumption. Machine learning (ML) techniques used for prediction have demonstrated promising results, compared to other traditional statistical methods offering better accuracy and flexibility. Researchers have studied various prediction methodologies, and in this chapter, some of the findings related to machine learning prediction techniques are reviewed. The use of machine learning (ML) models like the Random Forest (RF), Linear Regression (LR), Support Vector Regressor (SVR), and XGBoost in the area of predicting water consumption, highlight their methodologies, advantages, and findings.

2.2 Overview of Water Consumption

Water consumption is easily affected by many factors such as population growth, water price, contamination of pipelines, economic conditions, climate changes, contagious diseases as in the case of COVID-19, and many more. Scholars have mainly divided the research methods into 3 categories (a) time series method; (b) structural analysis method; and (c) system method [22]. Hawassa is a city located in the recently reformed Sidama Region in Ethiopia. Since its establishment, the source of water has been abundant. Lake Hawassa has been a primary source providing water for agriculture, drinking, and recreational activities. The city also has easy access to groundwater. However, the city administration (municipality) faced challenges in addressing the issue of access to water in individuals' homes. Nowadays the city relies on piped water where water access is inconsistent. The consumption patterns vary depending on the location, family size, and usage. The relationship between water consumption and the factors is the beginning of the structural analysis method. The system method also includes regression which has a good effect on long-term forecasting.

2.2.1 Factors Influencing Water Consumption Prediction

Water consumption prediction is a multifaceted task influenced by various factors that must be carefully considered for accurate forecasting. Much research has been done on water consumption and related prediction models[6], [10], [17], [23]. Some factors that influence water consumption are Demographic variables, Climate Variables, Socioeconomic factors, technological advancement, and Cultural Factors. Let us discuss them.

2.2.1.1 Demographic Variable

Demographic factors play a crucial role in shaping water usage patterns and access to clean water resources. One of the primary methods used to analyze water demand is the geometric approach of population forecasting, often with an annual growth rate of 3% to estimate water requirements in specific regions like Yergalem, and Tula Kebele [24]. Population density, household size, and income levels play key roles. As the population number increases so does the demand. Also, larger households generally consume more water. Lorena et al in the literature [25] stated, water consumption prediction was categorized into two: short-term and long-term. Water demand helps in urgent situations in planning maintenance and information on the water levels in reservoirs. This gives great insight into managing areas as to when to restrict in times of drought or water shortages.

2.2.1.2 Climate Variable

Variables like temperature, precipitation, and seasonal change fall in the climate variable category. Machine learning plays an important role in identifying water patterns and analysis is made on different climate conditions. Deep learning-based models, particularly Long Short-Term Memory (LSTM) models, have shown promising results in predicting water consumption by incorporating various variables to capture nonlinear patterns effectively [19]. These models consider a range of explanatory variables, including historical water consumption data, weather conditions, and day-to-day information, to enhance the accuracy of predictions by learning from past patterns and external influences.

2.2.1.3 Socio-Economic Factors

A municipality's prediction of water consumption helps in forecasting the need for water for the city and also faithful budgeting without blind or basic values per capita consumption calculation. The behavior of water consumption can be heavily influenced by public policies and water pricing(tariff). This gives a glimpse of consumption patterns in policy changes, seasonal variation,

and various rates of price. To develop a predictive model understanding the dependent and independent variables is necessary [22]. Prediction is one of the main goals of data analytics it has the art of telling what will happen in the future and what the causes are. With this information city administrators and policy makers can give a better forecast. Factors such as population growth rates, per capita water demand near water delivery points, and residential water demand calculations play pivotal roles in determining urban water consumption levels [24]. Furthermore, when assessing water distribution for urban areas, a per capita water demand of 25 liters within a 0.5 km radius from the water source is considered, taking into account demographic factors as outlined in the GTP-II minimum service level [24].

2.2.1.4 Technological Advances

One critical factor affecting water consumption prediction is the type of usage, such as detached houses, apartments, restaurants, and elementary schools, which can significantly impact the accuracy of the predictions made by the model. Advancements in technology, such as Smart Meters and IoT devices enable real-time data which is used in enhancing predictive capabilities through machine learning for efficient use of water. Urbanization and industrialization exacerbate poor wastewater management practices, significantly impacting water quality [26]. Urban wastewaters, including those from industrial and agricultural sources, further exacerbate fecal contamination and directly impacts surface water quality in urban regions [27]

2.2.1.5 Cultural Factors

Local practice, cultural perception, and Attitudes of water use can shape the consumption patterns. Firat et al [21] stated that accurately predicting water consumption is difficult as many factors affect the consumption rate. To get better efficiency and prediction effect the literature recommends considering the application of different calculation models and is necessary to study the issue further as mentioned above the prediction of water can be affected in many ways.

2.3 Overview of Machine Learning (ML) in Water Consumption Prediction

Machine learning is a part of AI (Artificial Intelligence) and is considered to be self-teaching using algorithms and methods that include rules and programs to train the data in order when faced with new data it will be taught. Machine Learning (ML) is considered to act like human intelligence by learning the environment without being explicitly programmed. The idea was introduced by a man named Arthur Samuel, in 1952, in the new era of big data. In his words “Machine learning is the

field of study that gives computers the ability to learn without being explicitly programmed" [28]. Machine learning can be categorized based on the desired output. These categories are Supervised Learning, Unsupervised Learning, and Reinforcement Learning. In Machine learning algorithms model building is based on sample data, known as the training, validation, and test data, used to make predictions or decisions

In recent years, Machine learning (ML) has been increasingly applied in developing models for water consumption. To tackle the linear and non-linear nature of the data. Many models have shown varying degrees of success in addressing the complexity of water usage. Models used in this study (Random Forest (RF), Support Vector Regression (SVR), Linear Regression (LR), and XGBoost) have also attained varying degrees of success.

2.3.1 Supervised Learning

The supervised learning method uses the training set to teach models to produce a desired output. This machine-learning method is known for its use of labeled datasets to train algorithms that classify data or which will predict the outcomes accurately[29]. Supervised learning is separated into 2 types of problems in machine learning: Classification and Regression

- Classification uses the task of assigning test or training data to specific categories based on the characteristics that the points have with their label. It recognizes entities within the dataset and draws some conclusions as those entities should be labeled or defined. Some of the most common types of classification algorithms are Support vector machines (SVM), K-nearest Neighbor (KNN), naïve Bayes, and random forest (RF).
- Regression is used to understand the relationship between dependent and independent variables. Mostly used in making projections. Some of the most popular algorithms are linear regression, Logistic regression, and polynomial regression.

Building a supervised learning model has 7 stages:

1. **Loading the data used for making predictions:** here is to gather the necessary data to begin
2. **Investigation of the data:** to understand the features of the data before starting. Visually inspecting the statistical data and viewing what operations must be performed.

3. **Preprocess the data:** for the data to fit in the model the data must undergo preprocessing and cleaning. It is even possible to go a step further and preprocess numerical data to create an even more accurate model.
4. **Choose the model's features:** to help in prediction, the necessary features are to be chosen to reduce computational cost and improve performance.
5. **Split the data to train and validate sets:** once we have inspected the data preparation steps, we split the data into training, validating, and testing sets.
6. **Train the model:** here we train the model.
7. **Test the model using the validation set:** finally, after all the steps have been done and the model is trained. We test the model using MAE, RMSE, or a common loss function. The lower the MAE and RMSE the better the prediction is on validation of the data.

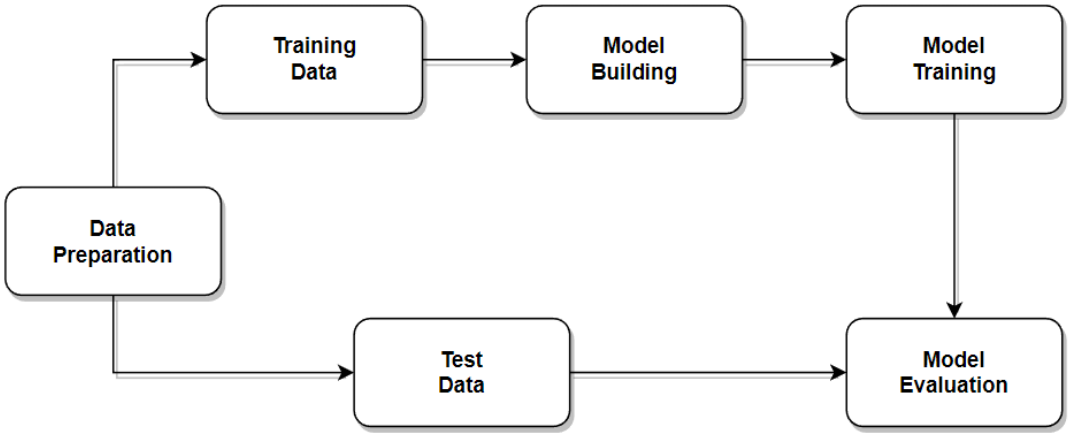

Figure 2.1 The figure for the supervised ML process

2.3.2 Random Forest (RF)

Random Forest (RF) is an ensemble learning method that constructs multiple decision trees and combines their predictions to improve accuracy and robustness [30]. RF decision tree splits the data set recursively using the decision nodes unless we are left with pure leaf nodes. It is particularly effective for handling non-linear relationships and interactions among features. RF is a model developed by Leo Breiman (2001) a substantial modification of bagging that builds a large collection of de-correlated trees [31]. Decision trees are highly sensitive to the training data which could result in high variance. They can be used in supervised learning for both classification and

regression. RF is a version of ensemble learning that takes multiple algorithms or the same algorithm and puts them together to make something much more powerful than the original.

Research done on RF models shows that the prediction accuracy can outperform traditional methods of predicting water consumption. According to the study by Adam Kulaczkowski (2024), the RF emerges as a robust and effective algorithm[32], with the ability to handle large datasets, making it the desired choice in water consumption prediction models.

We can see in Fig 2.2 that each tree is created from a different sample row and at each of the nodes a different sample feature is selected for splitting. Saying this each tree makes its prediction, and these predictions are then averaged to produce a single result[33].

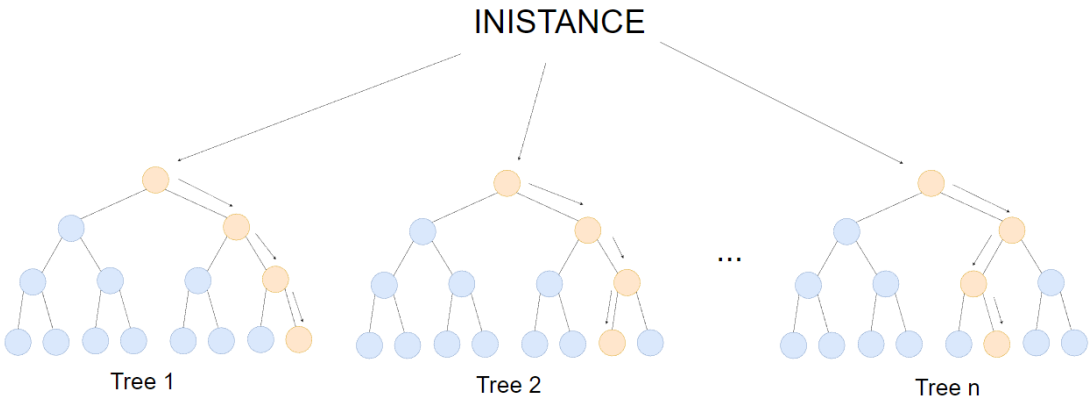

Figure 2.2 Random Forest Diagram

2.3.3 Linear Regression (LR)

Also known as simple linear regression is one of the simplest deterministic mathematical relationships between two variables X and Y. Regression models describe the relationship between variables by fitting a line to the observed data [34]. These variables are dependent and independent. One of the strengths of LR is its interpretability however, it starts to show limitations when the data set consists of significant non-linear characteristics, where models like RF and XGBoost tend to excel. When there is only one independent variable the term is a simple linear regression model when there are multiple independent variables considered it is known as a multiple linear regression model[35]. The equation for linear regression is:

$$y = \beta_0 + \beta_1 + \varepsilon$$

Equation 2.1 Linear Regression Equation

The ε is the error from the model equation. The y is the dependent variable that we are to get or study. The $\beta_0 \; and \; \beta_1$ are the parameters of the model. The parameters are known as regression coefficients. The independent variables are considered non-stochastic or the ones that are in control of the experiment.

2.3.4 Support Vector Regressor (SVR)

The Support Vector Regressor (SVR) is a powerful supervised learning algorithm commonly used for regression tasks, particularly in high-dimensional spaces. By transforming input data into a higher-dimensional space using kernel functions, SVRs effectively handle complex, non-linear relationships between the features and target variables. Unlike the Support Vector Machine (SVM), which seeks to find a hyperplane that maximizes the margin between classes, the goal of SVR is to find a hyperplane (or decision boundary) that minimizes the prediction error. SVR does this by ensuring that most data points fall within a specified margin, called the epsilon-insensitive tube, and penalizing points that fall outside this margin.

In SVR, the hyperplane is located in an N-dimensional space (where N is the number of features) to minimize the regression error. Unlike classification, where data points are assigned to distinct classes, SVR focuses on predicting continuous values and aims to produce the most accurate regression function. A graphical representation of this concept can be seen in Fig 2.3.

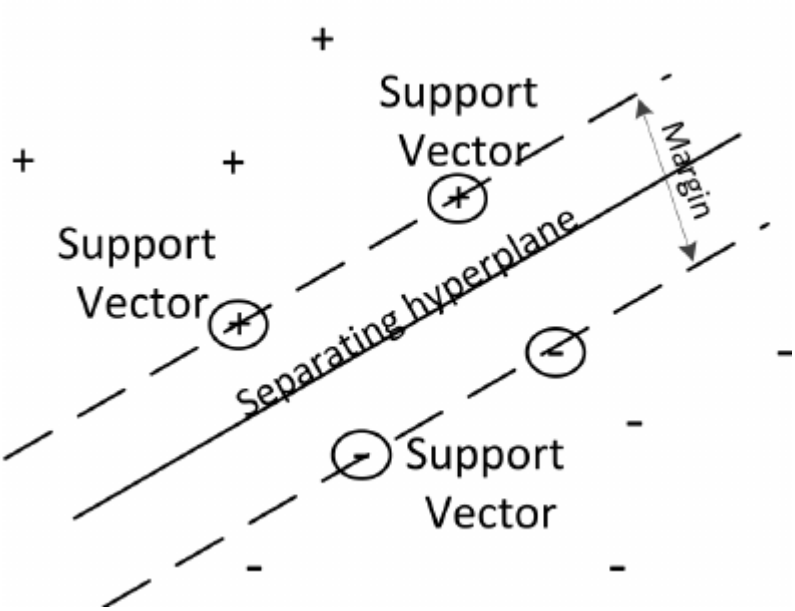

Figure 2.3 Support vector Regressor (SVR)

SVR's ability to handle non-linear relationships and its robustness to outliers make it particularly useful in diverse fields, including finance, bioinformatics, and time series forecasting. Research has shown that proper hyperparameter tuning is critical for optimizing SVR performance [37], highlighting the importance of selecting appropriate kernels (e.g., linear, polynomial, or radial basis function) and adjusting key parameters like C (regularization), epsilon (tolerance), and gamma (kernel parameter). These decisions significantly influence the model's ability to generalize and its predictive accuracy.

2.3.5 Extreme Gradient Boosting (XGBoost)

XGBoost (Extreme Gradient Boosting) parallel tree boosting (also known as Gradient-boosted decision trees (GBDT), Gradient Boosting Machines (GBM)) is a prominent machine learning algorithm known for its high efficiency and performance in predictive modeling tasks, particularly in structured data scenarios. Leveraging the principles of gradient boosting, XGBoost constructs an ensemble of decision trees, where each subsequent tree is trained to minimize the errors of the previous ones, effectively reducing bias and improving accuracy[38]. Its advanced features, including regularization techniques (L1 and L2), parallel processing, and a novel tree-pruning algorithm, allow it to handle overfitting and optimize resource usage, making it suitable for large datasets. Furthermore, XGBoost supports a variety of objective functions, enabling its application across different domains such as finance, healthcare, and e-commerce. Due to its versatility and robust performance in numerous machine learning competitions, XGBoost has established itself as a critical tool in the data scientist's toolkit, prompting ongoing research into its applications and enhancements.

2.4 Related Works

In this section, literature that was studied on water consumption, algorithms, and related topics are covered. Water consumption has become an increasingly critical area of study, particularly in light of growing concerns about water scarcity and sustainable resource management. Researchers have explored various methodologies to predict, analyze, and optimize water usage, often employing machine learning algorithms due to their capability to handle complex datasets and extract meaningful patterns. This section discusses the work directly related to the problems that are raised in this area.

Adam Kulaczkowski et al [32] conducted a compelling exploration using Random Forest (RF) on short-term water demand forecasting. The Algorithm was tested across ten district-metered areas (DMAs) in northeast Italy, from January 1st, 2021 to March 5th, 2023. The model was compared with traditional time series methods (ARIMA and SARIMA) and other ML algorithms. RF demonstrated a high accuracy rate in water demand compared to classical approaches, with training dataset R^2 of 0.95 to 0.99, while the testing was 0.58 to 0.98. In model training, a 7-day demand lag followed by 14 and 21-day demand lag was the most important explanatory variable. The literature provides a good insight into key variables that influence water consumption, which enables better decision-making. It's effective for near-to-real-time forecasting. The study presents some limitations like the model performance may vary from district metered areas (DMA) because specific variables are not accounted for, and a case-by-case evaluation was needed as no single model would work in all situations. Despite the limitations, the results show that the potential of RF for water demand forecasting is more effective. The RMSE for the training set ranged from 0.23 to 1.24 while the test was 0.58 to 3.15.

Kunwar P. Singh et al [36] they analyzed around 1500 water samples from 10 sites over 15 years. The literature achieved a Support Vector Classifier (SVC) classification accuracy of 92.5% data reduction for monitoring and predicted a Biochemical Oxygen Demand (BOD) with a correlation R^2 of 0.952. Linear models (Discriminant Analysis (DA) and partial Least Squares (PLS)) were outperformed by non-linear models (Support Vector Machines (SVM), Kernel Discriminant Analysis (KDA), Kernel Partial Least Squares (KPLS)) in both classification and regression tasks showing the effectiveness in water quality management. The performance was compared against traditional linear methods to improve the accuracy and predictive ability. The literature states a potential issue with overfitting in the neural networks, dependency on the dataset quality, and irrelevant data were required to be removed.

Bedassa Dessalegn Kitessa et al [39] studied the long-term water and energy demand in Addis Ababa, Ethiopia in 2018 using linear regression models. The model is used to express the dependence of water and energy demand on independent variables like population growth and Gross domestic product (GDP). By implementing the WEKA tool environment on the Historical consumption data of water and energy of the city, data preprocessing and filtering techniques gave way for regression models (Linear Regression (LR), Support Vector Regression (SVR)) and

classifiers (Naïve Bayes, Multi-layer Perceptron (MLP)) to be tested. The Literature estimated water demand of 520 million cubic meters (MCM) by 2030 and 1600 in 2050, and for energy 14,000 GWh by 2030 and 53,000 GWh by 2050, with a Relative Absolute Error (RAE) of less than 10% showing a high performance in linear regression models. The study outlines how population increase and per capita income (PCI) influence energy and water consumption rates, with a 5% increase in population there is a 4% increase in Kerosene. This provided insight for effective policy-making and enabled the selection of the best model for prediction by comparing different machine learning algorithms. The literature's potential issues are the accuracy of the regression model as well as the quality and completeness of data.

Antonio Candelieria et al [40] used Automatic meter readers (AMRs) in a water distribution network (WDN) in the Italian pilot project of ICeWater, from September – to December 2014 to propose a data-driven, self-learning algorithm for short-term water demand. The study was conducted by breaking it down into 2 approaches. The first is a time-series clustering method to evaluate the daily water demand patterns followed by Support Vector Machine (SVM) regression, the dataset is separately analyzed to obtain a specific daily water demand forecast model. The study employed a smart water management system that is completely data-driven. The data tested on real-world data of the ICeWater project collected a total of 26 AMRs which consist of around 100 valid inputs. From the 26 AMRs, the average Mean Absolute Percentage Error (MAPE) is less than 66.52%. The error was associated with the usage behavior in the time window of 9:00 to 13:00 which moved from 8:00 to 9:00 and in the evening. 50% of the AMRs resulted in less than 30% of MAPE in the forecast model and 15% of AMRs had a MAPE between 30% and 50%, while the smaller percentage error exceeded 50%. The short-term approach proved to be reliable on individual data, wider data is needed to make results more concrete and statistically sound. The complexity of this approach takes a significant number of resources if attempted for longer periods. However, the accuracy of the SVM has shown better performance on other models as it can adapt to aggregated and individual customers with the approach of continuously updating new data improving the accuracy over time.

Shan et al [6] classified the three major aspects of information in the collection of household water consumption under the ISS-EWATUS project. They are the: End user behavior, Socio-demographic, and earthy characteristics, and psychosocial constructs. The literature analyzed

domestic water consumption behaviors in Greece and Poland. The study used 43 questions as a questionnaire and they were distributed by the University of Thessaly (Greece) and the Institute of Ecology of Industrial Areas (Poland). There was a total of 77 cases in Greece and 41 in Poland from November through December 2014. The dataset in total is 118 values. The study provides valuable insight into household water consumption behavior and motivation for conservation. Household sizes were classified into two groups by their mean values and the shower length was indexed using an arithmetic average of the shower length of household members. This was replaced with an ordinal scale from 0 to 4 for their calculation. The mean number in Greece and Poland is 3.60 and 2.81 respectively. As a survey study, a comparison of water consumption behavior between different areas was implemented which caused inconsistent responses as the variability in the use of water differed from country to country. Also, the data relied on self-reporting which leads to inaccurate water usage estimates. The result suggested three major motivations for efficient water use, they are to save water, helping the environment, and saving money.

Shihao Shan et al[38] proposed the use of Attention-BiLSTM integrated with XGBoost residual correction. The study was conducted in southern China on a municipal water distribution system. The dataset was collected over 6 months from March 5 to August 26, 2018, with a 1-hour interval. The training, Validation, and test datasets were 105 days (15 weeks), 35 days (almost 5 weeks), and 5 weeks respectively. The maximal information coefficient (MIC) correlates historical data with the current values. The study utilized the Attention-BILSTM network as it leverages bidirectional information in historical sequence, this was corrected using the residual correction module of the XGBoost algorithm to refine and predict results in an error simulation. The results offered a fresh perspective on short-term water consumption in smart water management and showed that the model performed better than the benchmark models like the LSTM and BiLSTM. The model's performance metric was evaluated using the MAE of 544 m^3/h, RMSE of 925 m^3/h, Mean Absolute Percentage Error (MAPE) of 1.00%, and Nash-Sutcliffe Model Efficiency (NSE) of 0.99. This showed a 48.7% reduction in MAPE for the proposed Attention-BiLSTM model without XGBoost residual correction and a 41.9% RMSE reduction compared to the LSTM model. The relatively short time in the study might pose a limitation as the model's performance must work in various seasons in the year.

Zonghan Li et al [41] studied household water consumption and the factors that influence it. The study was conducted in Beijing, China in 2020. The data was collected from the Haidian District and the Tongzhou District having 21 and 4 subdistricts with 4 towns respectively. A self-design questionnaire survey was implemented resulting in a total sample data of 1320, valid 1257 after preprocessing. The collected data underwent 3-step validation (cleaning, verifying the validity of questionnaires, and calculating water consumption based on water use) and a 3-sigma principle to remove outliers. A total of 24 features were observed after the questionnaire ranging from basic housing information to usage behaviors. The study used the Least Absolute Shrinkage and Selection Operator (LASSO) and selected 4 important features (Household information (HI), Water use (WU), Energy use (EU), and Electric consumption (EC)) to build 4 models with each combination. The study conducted models on Ordinary Least squares (OLS), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) and evaluated the 4 models. Model 1: OLS, RF, and XGBoost models with HI and WU, Model 2: OLS, RF, and XGBoost models with Hi and EU, Model 3: OLS, RF, and XGBoost models with HI, WU, and EU, Model 4: OLS, RF, and XGBoost models with HI, WU, EU and EC resulting a R^2 of 0.25, 0.26,0.29, and 0.32 respectively. The study was conducted during the COVID-19 pandemic which altered the usage behavior, with a small sample size but the study shows a potential for broad application.

Table 2.1 Summary of the related papers

No.	Authors	Title	Data set	Algorithms	Metrics and Result	Limitations and Gap
1	Adam Kulaczkowski , Juneseok Lee	Harnessing the Power of Random Forest for Precise Short-Term Water Demand Forecasting in Italian Water Districts	10 Ten district metered areas (DMAs) from Jan 1, 2021 to Mar 5, 2023	-Random Forest (RF) -Comparative Analysis	Training: range from 0.95–0.99 Testing: 0.58–0.98 RMSE ranges from 0.23-1.24 and 0.58-3.15	-Model Performance Variability dropping from 0.96 to 0.74 -Case-by-case evaluation is needed as potential overfitting is seen in the model as it didn't generalize well to unseen data.
2	Kunwar P. Singha, Nikita Basantb, Shikha Guptaa	Enhancing Water Quality Monitoring and Prediction Using Support Vector Machines	1500 water samples were collected from 10 sites over 15 years.	-Linear Methods (DA, PLS) -Nonlinear Methods (SVC, SVR, KDA, KPLS)	SVC: Training 12.39%, Validation 17.70% and Test 14.86% SVR: R = 0.952 for training, 0.909 for validation, 0.907 for test) and low RMSE (1.53, 1.44, and 1.32 respectively).	-Model complexity as potential overfitting in neural networks and SVM require extensive computational resources. -Data Quality: the presence of outliers and contaminated data lead to unreliable results -Feature Relevance
3	Bedassa Dessalegn, Semu Moges,	Long-term water-energy demand prediction using a regression model: a	Historical consumption data of water	-Regression models (LR, SVR, MLR)	LR: RAE Less than or equal to 10% MAE and RMSE values were minimized	- Accuracy concern in the regression model needs to be further tested as the quality and availability of historical data

	Geremew Sahilu and Solomon Tesfamariam	case study of Addis Ababa city	and energy in Addis Ababa	- Classifiers (Naïve Bayes, MLP) -AI (ANN, ANFIS)	Energy Loss (2017 vs. 2034): Technical: 13% to 8% Nontechnical: 3% to 1% Total: 16% to 9%	can lead to uncertainties in prediction - Dependent on the quality and completeness of data and the reliance on socio-economic indicators like GDP as it may not capture all the factors influencing water demand.
4	Antonio Candelieri, Davide Soldi, Francesco Archetti	Short-term forecasting of hourly water consumption by using automatic metering readers' data	26 AMRs with data 100 valid inputs. 2600 values	-SVM, RVR, ANN - time-series clustering - Multiple Kernel Learning	The average MAPE for different values of m ranged from 65.12% to 67.33%, indicating variability in forecasting performance across AMRs.	- Limited dataset: Shortage of data availability -Anomalous Data Removal and Lack of Seasonality Analysis - Variability in consumption patterns: the complexity of the approach as it works on individual behavior and requires wider data
5	Dimitris Koutsoyiannis , Maria Koutsoyiannis	Household Water Consumption: Insight from a Survey in Greece and Poland	77 cases from Poland Greece and 41 from Poland	-Descriptive statistics: Survey and statistical method.	Water Consumption Estimates: 82% Greek and 63% Polish Motivations for Water Efficiency: having	- Sample size: Relatively small Poland (41 Respondents) - Conducted in a university while the title implies household

	, Katarzyna Kaczmarek, Michał Kaczmarek		A total of 118 values	- Ordinal Regression analysis and Factor Analysis	water, helping the environment, and saving money. Intervention Responsiveness: mandatory restrictions, and adjusted pricing.	-Self- Reported Data: Inconsistency of the data sets - Cultural differences: did not fully account for cultural differences in water usage behaviors and attitudes between the two countries
6	Shihao Shan, Hongzhen Ni, Genfa Chen, Xichen Lin, Jinyue Li	A Machine Learning Framework for Enhancing Short-Term Water Demand Forecasting Using Attention-BiLSTM Networks Integrated with XGBoost Residual Correction	March 5 - August 26, 2018, with 105 days for training, 35 days for validation, and 5 weeks for testing.	-Attention-BiLSTM network -XGBoost on residual correction - Random search for hyper-parameter optimization of BiLSTM	Mean Absolute Error (MAE): 544 m^3/h Root Mean Square Error (RMSE): 915 m^3/h Mean Absolute Percentage Error (MAPE): 1.00% Nash-Sutcliffe Efficiency (NSE): 0.99	- Relatively short time the study should address seasonal changes in the year for unforeseen events or external factors. -The gap lies in the need for more adaptive models that can dynamically adjust to real-time data and changing consumption patterns for enhanced forecasting accuracy.

7	Zonghan Li, Chunyan Wang, Yi Liu, Jiangshan Wang	Enhancing the explanation of household water consumption through the water-energy nexus concept	The data size is 1320, and valid 1257 values	-Ordinary Least Squares (OLS) - RF -XGBoost -LASSO	Results indicate that Model (4), which incorporates energy-related features, achieved an average R^2 of 0.52, a decrease in RMSE by 5.1%, and a reduction in MAPE by 3.8% compared to Model (1).	-Time of the study COVID-19 altered usage patterns -small sample size and challenges in data collection at the household level, while it has the potential for broad application. -The need for further exploration of energy use as a significant factor in household water consumption estimation models.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

In Chapter 2, several studies have been evaluated to identify the issues and gain a deep understanding of the problem, to provide a solution. This chapter covers the overall design techniques, approach, and instruments for the execution of the model. The implementation of research is to find a fact or research of scientific investigations through a means of studying a given subject matter in a particular area of interest [42]. The research approach followed in the study and the methods are based on the problem statement and objectives stated in chapter one.

This study uses an experimentation approach to create a machine learning model; the study is tested on a real-world dataset and then performance is evaluated. The study's experimental part is to look at and identify new ideas with insights into the possibility of better approaches using water consumption forecasting for the community. The study is used to give the most reliable proof of causation, it aims to solve societal or business problems. Therefore, to implement this having a test group and a training group is necessary to have a better outcome[43].

3.2 Research Design

In the study of J. Kamiri and G. Mariga [44], many machine learning types of research are conducted using an experimental approach. This approach seeks to identify new hidden factors that might affect the water consumption rate differentiation. The study also followed that and was conducted using an experimental research method to articulate the outcome, using more than one machine learning method so that the model is tested and the performance is evaluated on numerical data. An extensive literature review was necessary to formulate the research questions. Once the idea of the research was understood data collection was performed to solve the proposed problem. Finally, by evaluation of the data, the consumption rate was predicted and the results of the proposed system were evaluated and analyzed which can be used for future predictions as well. The diagram (Fig 3.1) below shows the research process based on the experimentation approach.

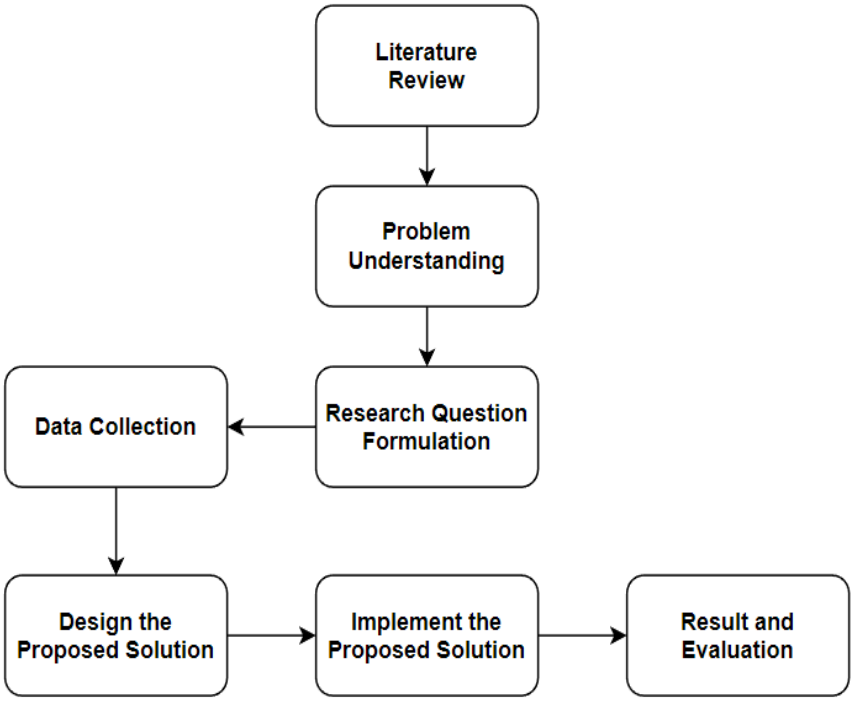

Figure 3.1 Experimentation Approach

3.3 Research Approach

The approach used in this study is an experimental approach using machine learning (ML) techniques. Research approaches used in ML play a pivotal role as the accuracy and reliability are influenced by the methods used [44]. The target variable is used in regression models, as the dependent variable is the study focuses on modeling. It attempts to present new insight into the water prediction arena using the independent variables to give results on the dependent variable.

3.3.1 Description of the Study Area

The study was conducted in the city of Hawassa, Ethiopia. The data originated from the Sidama Regional State Hawassa City Water Supply and Sewage Service Enterprise (SRS-HCWSSSE) database. The necessary data was requested by official channels by submitting a letter written by the department head of Computer Science at the Institute of Technology of Hawassa University. For the data to be approved for use, the Hawassa City Water Supply and Sewage Service Enterprise (HCWSSSE) senior management was consulted to approve the request to attain official data from their database. Once this was approved by the department's managing head, the data was collected from their IT department. The data provided was historical data on the last five (5) years of the

consumable water from the HCWSSSE database. This does not include groundwater or the lake in the city. We can see in Fig 3.2 a map view of Hawassa city.

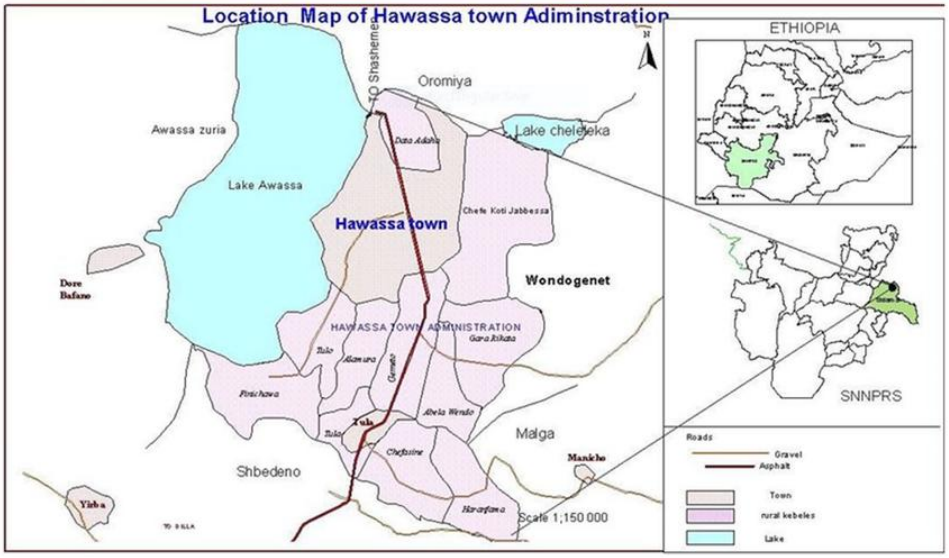

Figure 3.2 Administrative map of Hawassa city. Source: Hawassa-City-Health-Department-23

3.3.2 Data Collection

The study required numerical historical water consumption data, to design and evaluate the model. We looked into the historical consumption data from the Hawassa City Water Supply and Sewerage Service Enterprise (HCWSSSE) database. The HCWSSSE dataset is used, which contains the overall consumption rate of a given month and the annual consumption. The data collected contains information on the number of customers and monthly consumption in a given kebele for a specific month concerning its Customer Type. This information is useful to predict the next month's consumption prediction rate of water. The raw data collected from the HCWSSSE before preprocessing for the models is shown in Fig 3.3. Characteristics found in the database are both categorical and numerical values.

There are two types of data collection methods primary and secondary, primary data is data that is collected firsthand hand while secondary data is data that is already been published [45]. The data provided here is secondary data collected from 2010 - 2015 E.C of the Budget year of the Ethiopian Government. The HCWSSSE data is collected monthly, eight variables were initially noted. They are the following: Kebele, CustomerType, Consumption, Number of Customers (Numb_Cust), Month, Year, ConsAnnualTotal, and CustAnnualTotal. Encoding was done on the CustomerType, Kebele, and Month as they were all categorical variables.

Consumption Summery in Cubic Meter - By Kebelle - Year BETWEEN 2014 AND 2014

	2014																				Grand Total	
	Hamle/ሐምለ 2013		Nehase/ነሐሴ 2013		Meskerem/መስከረም 2014		Tikimt/ጥቅምት 2014		Hidar/ህዳር 2014		Tahsas/ታህሳስ 2014		Tir/ጥር 2014		Yekatit/የካቲት 2014		Megabit/መጋቢት 2014		Miazia/ሚያዝያ 2014		Ginbot/ግንቦት 2014		Sene/ሰኔ 2014			
	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust	Cons.	#Cust.
24	12	1	14	1	10	1	5	1	6	1	7	1	6	1	7	1	5	1	14	1	20	1	10	1	**116**	**12**
27A	8	1	12	1	19	1	32	1	44	1	41	1	9	1	70	1	137	1	100	1	33	1	7	1	**512**	**12**
27B	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	**0**	**8**
Adare	9,185	617	7,629	616	10,616	615	7,884	613	8,666	615	9,883	608	10,124	605	9,284	602	9,957	604	11,620	604	7,753	604	8,836	604	**111,437**	**7,307**
Addis Ababa	4,820	564	5,000	562	5,223	564	4,726	561	7,611	561	6,157	563	5,946	563	5,772	568	6,442	568	6,141	569	5,254	570	5,446	569	**68,538**	**6,782**
Andinet	2,840	240	1,905	242	2,561	241	2,394	240	3,190	245	2,513	246	3,501	245	2,673	246	2,995	247	2,939	247	2,576	245	2,572	243	**32,659**	**2,927**
Bono Hawassa	848	27	802	23	656	23	491	22	436	22	497	22	661	22	1,182	22	466	22	740	22	549	22	127	22	**7,455**	**271**
Daka	16,598	1,585	17,935	1,577	18,417	1,579	18,644	1,570	20,553	1,570	17,813	1,579	22,299	1,589	19,525	1,592	16,588	1,594	17,623	1,594	17,020	1,602	18,523	1,602	**221,538**	**19,033**
Dato Odahe	13,057	2,156	11,470	2,108	12,737	2,110	13,339	2,086	15,137	2,091	17,009	2,091	15,433	2,097	14,083	2,098	12,131	2,090	14,165	2,096	13,908	2,155	23,288	2,223	**175,757**	**25,401**
Dato Odahe 1	7,216	1,168	6,087	1,191	4,829	1,220	6,887	1,218	7,411	1,253	8,892	1,391	8,598	1,456	8,742	1,526	8,646	1,605	8,348	1,645	9,750	1,657	8,535	1,706	**93,941**	**17,036**
Dume	12,791	1,086	8,604	1,086	10,954	1,088	12,741	1,089	11,573	1,088	11,396	1,085	12,491	1,083	12,952	1,087	12,481	1,089	13,047	1,091	10,341	1,092	10,838	1,090	**140,209**	**13,054**
Dume B	17,389	1,281	10,257	1,282	11,908	1,279	12,598	1,277	13,237	1,275	14,233	1,273	14,768	1,275	12,698	1,275	12,710	1,277	14,551	1,278	9,887	1,278	14,519	1,281	**158,755**	**15,331**
Fara 1 Gudumale	11,556	729	8,607	725	9,423	726	12,451	737	12,730	747	12,780	759	14,886	795	14,102	814	14,840	830	10,902	834	11,407	837	9,597	844	**143,281**	**9,377**
Fara 2 Gudumale	11,433	1,238	9,701	1,237	10,797	1,233	10,756	1,245	11,465	1,260	11,596	1,263	13,248	1,278	12,996	1,294	13,881	1,340	12,078	1,354	10,921	1,342	9,994	1,362	**138,866**	**15,446**
Fara Alamura	31	26	131	24	29	24	67	24	74	24	26	24	18	24	37	24	4	25	2	25	4	25	23	25	**446**	**294**
Fara Alamura A	19,738	1,473	14,815	1,471	10,907	1,482	22,279	1,497	19,166	1,507	24,691	1,497	17,456	1,490	21,044	1,506	18,665	1,514	17,597	1,514	14,569	1,515	19,206	1,505	**220,133**	**17,971**
Fara Alamura B	4,610	938	12,914	938	8,543	951	9,375	961	9,100	974	7,402	962	8,584	963	10,184	984	9,256	985	8,170	987	6,070	986	8,412	989	**102,620**	**11,618**
Fara Alamura C	12,805	1,253	19,339	1,256	12,439	1,283	15,606	1,313	11,957	1,332	12,828	1,329	14,421	1,331	16,628	1,370	13,066	1,384	11,944	1,387	12,676	1,437	107,120	1,632	**260,829**	**16,307**
Fara Gudumale	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	**0**	**12**
Feladelfia	19,483	2,548	16,282	2,507	19,249	2,496	21,541	2,470	23,616	2,472	21,334	2,476	22,803	2,488	22,383	2,489	17,691	2,499	22,651	2,503	17,173	2,512	19,948	2,509	**244,154**	**29,969**
Gebeya Dar	12,992	835	10,472	836	12,476	835	11,545	838	12,538	844	16,623	845	15,045	848	14,350	846	13,153	849	15,816	849	9,606	852	11,785	862	**156,401**	**10,139**

Figure 3.3 Sample Data (source: Sidama Regional State Hawassa City Water Supply and Sewage Service Enterprise (SRS-HCWSSSE))

Let us understand the variables in detail:

A. Kebele

The Kebele type is a categorical variable in the data set. A kebele is the smallest administrative unit in Ethiopia. There are 54 unique values (kebeles) in Hawassa city that are specified under the HCWSSSE data set shown in Text Box 3.1. The enterprise classified the data according to the precise area known as kebele, which are location points in the city. Inspecting the data multiple kebeles have more than one set and are alphabetically and numerically assigned. This was done intentionally by the HCWSSSE.

```
array(['24', 'Adare', 'Addis_Ababa', 'Andinet', 'Daka', 'Dato_Odahe',
       'Dume', 'Dume B', 'Fara 1 Guduma', 'Fara 2 Guduma',
       'Fara Alamura','Fara Alamura A', 'Fara Alamura B',
       'Fara Alamura C', 'Fara Alamura D', 'Fara Gudumale',
       'Feladelfia', 'Gebeya Dar', 'Gudumale', 'Guye Stadium A',
       'Guye Stadium B','Guye Stadium C','Harer', 'Hiteta','Hiteta A',
       'Hiteta B', 'Hiteta C', 'Hiteta D', 'Hogane Wacho', 'Leku',
       'Millennium', 'Millennium A','Millennium B', 'Nigat Kokob',
       'Piassa', 'Tesso', 'Tilte','Tilte A', 'Tilte B', 'Tilte C',
       'Tilte_D', 'Tilte_E', 'Woukro','27B', 'Bono_Hawassa',
       'Dato_Odahe_1', 'Tilte_F', '27A', 'CBEBirr_Adjustm',
       'Dato_Odahe_2', 'Dato Odahe 3', 'Dato Odahe 4','Dato Odahe 5',
       'Dato Odahe 6', 'Dato Odahe 7', 'Dato Odahe 8', 'Dato Odahe 9',
       'Dato_Odahe_10', 'Dato_Odahe_11', 'Dato_Odahe_12',
       'Dato_Odahe_13'], dtype=object)
```

Text Box 3.1 Unique values of kebele

B. Customer Type

The Customer type is another categorical variable in the HCWSSSE dataset. There are 11 unique values which are represented below (Text Box 3.2). The city of Hawassa categorizes these customer types to identify which industry is used for residential or other aspects of a kebele.

```
array(['Commercial Customers', 'Domestic', 'Governmental Customers',
       'Har_Commerc', 'Har_Domestic', 'Har_Industrial', 'Har_Public',
       'Har_Standpipes(Bono)', 'Industrial Customers',
       'Public Enterprises', 'Standpipes(Bono)'], dtype=object)
```

Text Box 3.2 Unique values of CustomerType

C. Month and Year

The Month column is a categorical variable in the data set, while the year is an integer. In the Ethiopian calendar, there are 13 months. The starting month of the Ethiopian year is `'Meskerem/መስከረም'` (which is in the months of 'September' and 'October' in the Gregorian calendar) and ends with `'Pagume/ጳጉሜ'` (which is at the beginning of the month of 'September' in the Gregorian calendar). The beginning of counting for the Hawassa City water supply and sewage service Enterprise is from `'Hamle/ሐምለ'` ('July' in the Gregorian calendar). This is due to the budget year of the Ethiopian Government which starts from the month `'Hamle/ሐምለ'` and ends in the month `'Sene/ሰኔ'` ('June' in the Gregorian calendar). The study used the budget year shown in Text Box 3.3 to avoid confusion about the dates.

```
array(['Hamle/ሐምለ', 'Nehase/ነሐሴ', 'Meskerem/መስከረም', 'Tikimt/ጥቅምት',
       'Hidar/ህዳር', 'Tahsas/ታህሳስ', 'Tir/ጥር', 'Yekatit/የካቲት',
       'Megabit/መጋቢት', 'Miazia/ሚያዝያ', 'Ginbot/ግንቦት', 'Sene/ሰኔ'],
       'Pagume/ጳጉሜ', dtype=object)
```

Text Box 3.3 Unique values of the Months

The month and year were combined in a value called 'CurrentDate' and parsed to get a 'date64time' data type (dtype). The date was set to a day value of 20, this is because the day when the enterprise updates its database is on the 20th day of the month. The above (Text Box 3.3) shows the unique values of the month. In the year 2012, there was a dip in both consumption and, customers this was due to the month of `'Pagume/ጳጉሜ'` which is not a 30-day value in the Ethiopian Calander and it was only present in the year 2012. This resulted in the month of `'Pagume/ጳጉሜ'` being removed while visually inspecting the dataset as it was only recorded in 2012 E.C, and was considered an outlier in the dataset.

```
#Convert the month to numeric values:

month_numbers = {'Hamle/ሐምለ': 1, 'Nehase/ነሐሴ': 2, 'Meskerem/መስከረም': 3, 'Tikimt/ጥቅምት': 4,
                 'Hidar/ህዳር': 5, 'Tahsas/ታህሳስ': 6, 'Tir/ጥር': 7, 'Yekatit/የካቲት': 8,
                 'Megabit/መጋቢት': 9, 'Miazia/ሚያዝያ': 10, 'Ginbot/ግንቦት': 11, 'Sene/ሰኔ': 12, 'Pagume/ጳጉሜ': 13}

# Create another month column
df = df.assign(TheMonth=df.Month)
df.sample(5)
```

Figure 3.4 Code snippet of Encoding of the months

D. Number of Customer

As the name suggests, the number of customers (Numb_Cust or #Cust) stores the total number of customers monthly, it keeps track of all monthly customers, as the name would imply. The Numb_Cust is a variable that impacts the monthly consumption rate, as the number of customers rises the number of consumptions also increases.

E. Month Consumption

The monthly consumption (Cons.) is the monthly consumption record, this variable is the study variable, the study focuses on the monthly consumption prediction from the dataset and its effects on what makes a better prediction as the main objective of the Study. For machine learning to occur, the consumption variable was converted from an "object" type format to an integer. Month consumption is a dependent variable that depends on other variables to determine its value.

3.3.3 Data Preparation

Data preparation includes all the necessary steps in developing a final dataset used in the model to predict an outcome. It also includes selecting the best way to extract data for the model and dividing it. Every learning system has specific requirements about how data must be presented for analysis and hence data must be transformed to fulfill those requirements[46]. Data preparation plays a big role in building quality models because algorithms are dependent on the quality of the data that they operate on. If the data is insufficient and inappropriate, machine learning algorithms might result in less accurate and less understandable results[47]. It includes data handling, removing or modifying outlier observations, data transformation (often normalization and standardization), and feature selection. The first two steps are useful for a more accurate and complete dataset so that the data will be consistent and properly organized, and the third one is typically used to have more uniformly distributed data and to minimize data variability because transformed data can be easily used by both people and computers. Finally, the fourth step is used to obtain the required independent variables which have more relation with the dependent feature and help in building a good model.

Distributing data means dividing it into datasets for training, validation, and testing. The training set is the bulk of your sample data. It’s where the machine learning model learns the patterns in the data. In the study, eighty percent (80%) of the sample data was used for training, and the remaining 10% for validation and 10% for testing.

Here's how the data was broken down:

- Training set = 80% of the total sample size
- Validation = 10% of the total
- Test set = 10% of your total

3.3.4 Handling Missing Values

Many machine learning algorithms do not support values that have null values or that are not present, so it is important to handle them accordingly. Raw data oftentimes include missing values, this might happen for a variety of reasons. For example, if the customer doesn't pay on time, or when the system is down and the next record is on the next month, the data encoder is presented with an empty value and needs to represent the absence of that value. So, to address this issue, it's important to understand the nature of the data and fill in the missing values accordingly. In the study, the missing data were replaced with Zero (0) in the dataset either by the encoder or when the absence is noticed the dataset automatically records 0.

3.3.5 Encoding

The HCWSSSE dataset includes categorical variables. Before performing feature engineering such variables need to be addressed. The machine Learning (ML) model uses mathematical formulas that can accept numerical digits as inputs in some cases categorical also. To use this dataset, we have to convert the mixture of data (Numerical + String), one of the most effective ways is performing encoding. It is the process of converting strings into their numerical equivalent without losing significance. The label Encoder method is used to convert categorical variables to numeric values when needed. A total of 3 variables in the dataset were categorical variables. The values were converted using sklearn's LabelEncoder Function. The label encoding method was selected in the study as it showed a better handle for the months and the ability to sequence the month using the budget year of the Ethiopian government than other encoding methods.

3.3.6 Data Normalization

The data was converted initially by normalizing the input and output variables to reduce the noise. Data normalization is the process of converting data to a manageable form to reduce runtime data was then mapped to understand the underlying relationship. The need for data normalization is evident as it's used when the data needs to match the destination system. [48]. The data in the study

is scaled using a min-max scalar on the data which is manageable and easier to read between 0 and 1. This ensures that the data is easily understandable. Transformation is given by:

$$X_{std} = \frac{(X - X.min)}{(X.\max - X.\min)}$$

$$X_{scaled} = X_{std} * (max - min) + min$$

Equation 3.1 Normalization (Minmax Scaling)

3.4 Proposed Solution Design

Once we understand the problem after concluding that the current system of forecasting by the previous month manually can be determined not to be adequate, a new solution had to be designed to address the problem that is raised in the study. The necessary activities that are important in water consumption prediction are data collection, data preprocessing, model building, model comparison, and finally suggesting a better prediction model to estimate the monthly consumption rate.

Once data collection has been done, the selection of features continues to identify what will be best suited for the model based on their importance and weight. After Feature selection, in the study preprocessing is necessary as the data collected might contain missing data and NaN values in the data set. The preprocessed data can then be separated into training and testing data and modeling can occur. The best-suited model is then selected from the models that have been proposed.

The outcome of this study is to model water consumption for HCWSSSE dataset. Building this model helps in decision-making for municipal city planning in water consumption for the coming months to years. The prediction model would help in assessing the rate of water consumption in the city in the future. The study proposes models for water consumption prediction. In Fig 3.5, the flow of the methodology used in the study is shown.

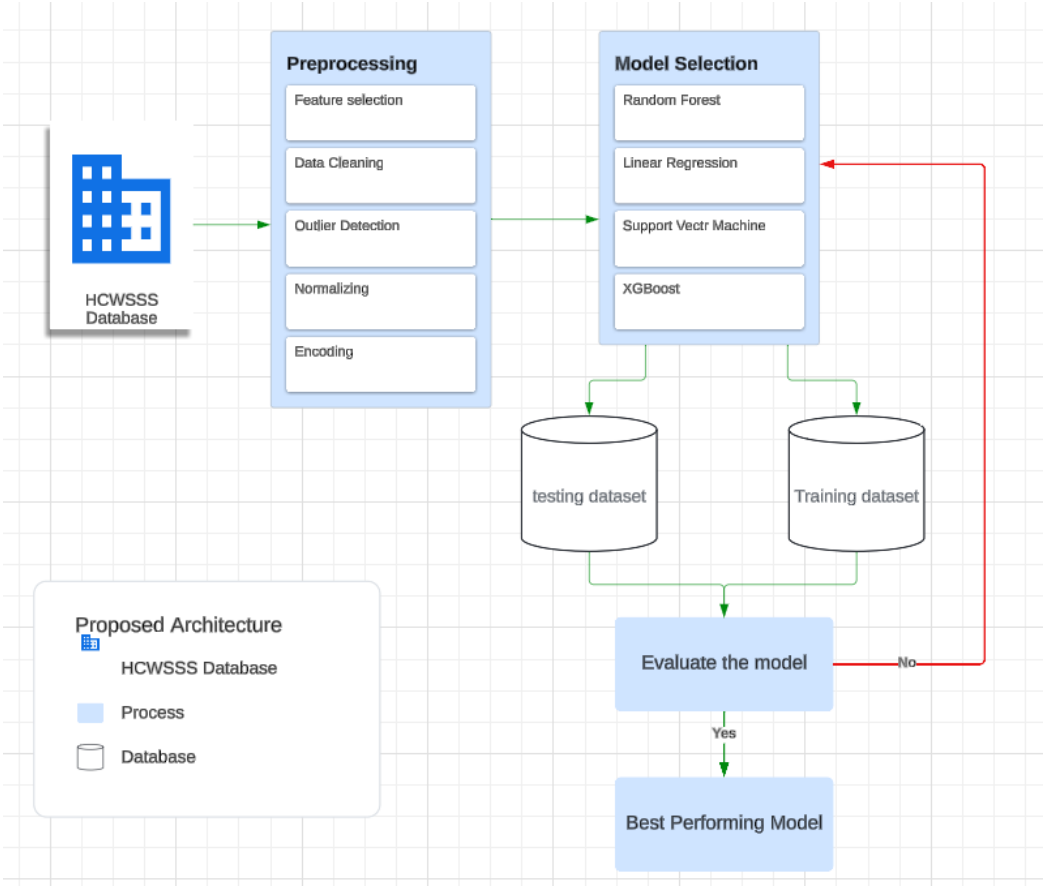

Figure 3.5 The Proposed Work Architecture

3.4.1 Design tools

It is easier to understand visual data than in textual formatted concepts, and ideas are more visually appealing to the eye. As we human beings are visual beings, we can understand better pictorial depictions. To make the solution more appealing and understandable, the solutions to the problem are depicted as much as possible in a pictorial form. The tools used to attain this are draw.io, lucid Chart, and offline designing tools like Adobe Photoshop and Illustrator platform.

3.4.2 Proposed Solution Implementation

After performing the above tasks, the next step is to implement the experimentation with the selected algorithms Random Forest (RF), Linear Regression (LR), Support Vector Regressor (SVR), and XGBoost. While conducting this study, machine learning and environments were looked at, to list out some WEKA and Anaconda-based Python were viewed and the Anaconda-based Python environment was selected. The environment showed vast resources and packages

used for machine learning. To list a few the TensorFlow, Keras, pandas, Matilab, and, pdarima libraries are used for machine learning.

3.5 Model Selection Criteria

Many methods can be used in water consumption as mentioned by Zhaocai Wang [49]. Algorithms from classical ML models can be used as well as deep learning methods. Some differences are data size (deep learning requires a large amount of data while classical performs well on smaller datasets), Interpretability (classic models are often more interpretable and easier to explain), Computational Resources, Training time (classic models generally train faster), Problem Type (for structured/tabular data, classic models often perform better or equally well as deep learning), overfitting (deep learning models are more prone to overfitting, especially with smaller datasets), and pre-trained models (deep learning excels when pre-trained models are available, but for many tasks that's not always true).

Deep learning is powerful for tasks involving large and unstructured data while classic ML is better suited for smaller, structured datasets, offering faster training, easier interpretability, and lower computational costs. The algorithms selected were considered to have good results in water consumption and prediction models. The Random Forest (RF), Linear Regression (LR), Support Vector Regressor (SVR), and XGBoost are trained and compared against each other. Let us discuss the reason for the choice.

3.5.1 Random Forest (RF)

Random Forest is a classification-based decision tree, a collection of trees used for prediction. Generally, known to perform well with large datasets, handles non-linear relations well, and reduces overfitting through averaging. It is ideal for exploratory data analysis[33]. Widely used in classification and regression models and provides functionality with training, validation, and testing data sets.

3.5.2 Linear Regression (LR)

Linear Regression is best for simple interpretation in modeling[34]. Scales well with large datasets and is very fast in prediction once the model is trained. It assumes linear relationships but needs significant feature engineering and transformation to capture non-linear relationships.

3.5.3 Support Vector Regressor (SVR)

Support vector Regressor is one of the algorithms used in regression models. It is suitable for complex decision boundaries but sensitive to feature scaling and kernel choice[36]. It may struggle with very large datasets with overfitting problems but it's manageable with parameter selection.

3.5.4 Extreme Gradient Boosting (XGBoost)

XGBoost from the gradient boosting library is excellent in capturing non-linear patterns due to its boosting techniques. Often seen as a "black box" as it is highly efficient and optimized in large datasets[38]. Generally, fast in prediction and includes parameters to reduce overfitting while handling missing values, making it a strong choice.

3.6 Working Environment

Knowing the pros and cons of the development environment results in successful and productive work, reasonably choosing a development environment is important to the success of the study. Different hardware tools, software tools, and programming languages were used in this study. The environments used in this thesis work are as follows.

3.6.1 Software tools

The software that is installed on a computer to perform the tasks that are required to solve the proposed problem is as follows

- Operating system Windows (11) Home Version (23H2) Build 22631.4317
- AMD Ryzen 7 5800H with Radeon Graphics 16GB RAM with 4GB GPU
- Anaconda Navigator 2.6.0
- Python 3.12.3
- Draw.io, LucidChart, Photoshop, and Illustrator 2023
- Jupyter Notebook 6.5.4
- NumPy, Matplotlib, pdarima, Keras library with the Tensorflow as a back end

NumPy, an open-source Python library was used as it is used for various mathematical and scientific tasks. It is used for working with arrays and has the function for working in the domain of linear algebra. To visually understand the data other than the statistical description using **Matplotlib** a plotting library is used for creating figures, and plotting the area in a figure. Also,

for data use and data wrangling the **Pandas** tool was used. **Keras** and **TensorFlow** is an open-source Python library, mainly used for building and training machine learning models.

3.6.2 Programing language

There are many different programming languages and frameworks to code machine learning models but in this study, the Python programming language has been preferred. As mentioned above python has been used to build prediction and training machine learning models in different scenarios. Python is one of the most popular programming languages for machine learning as it is open source and easy to use, with much information included in the libraries.

3.7 Evaluation of the Models

There are many ways to evaluate the performance of machine learning in regression models some are the R-squared (R^2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Evaluating models using these methods helps to quantify the model's performance in a measurable term, it enables the comparison between the models and hyperparameter tuning and provides insight into how well the model aligns with the business goals and specific performance needs. The study employed the measurements R^2, MAE, MSE, and RMSE for evaluation comparison to assess the water consumption prediction models. Each metric shows different results because they measure different aspects of the model performance, and are based on different mathematical formulations. Finally, the best models are selected according to the evaluation techniques.

3.7.1 Mean Absolute Error (MAE)

Mean absolute error is one of the many metrics for summarizing and assessing the quality of a machine learning model. Mean Absolute Error (MAE) is the simplest measure of accuracy. It is simply the mean of absolute errors. Absolute error is the difference between forecasted and actual value. It evaluates the error of the continuous data by averaging all absolute errors. Initially, it subtracts the predicted value from the actual value and then calculates the mean for all recorded absolute errors (Average sum of all absolute errors). It tells you how big an error you can expect from an average forecast. The mean of the errors is the average of all the errors in the set of predictions.

$$mae = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Equation 3.2 Mean Absolute Error (MAE) Equation

Where:

- n is the number of observations,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

3.7.2 Mean Squared Error (MSE)

Mean Squared error is a widely used metric for assessing the accuracy of a predictive model. It calculates the average squared difference between the predicted values and the actual values in a given dataset. The formula for calculating MSE is as follows:

$$mse = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Equation 3.3 Mean Squared Error (MSE) Equation

Where:

- n is the number of observations,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

3.7.3 Root Mean Square Error (RMSE)

The RMSE calculates the difference between prediction and truth for each data point. It is the standard deviation of prediction errors. RMSE measures how far out the residuals are from the line. RMSE is expressed as:

$$RMSE = \sqrt{mse} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Equation 3.4 Root Mean Square Error (RMSE)

3.7.4 R-Squared (R2)

R-squared (R^2) is a useful metric for assessing how well a model fits in a regression model. It uses a statistical measure to find the "percent of variance" of the dependent variable. In a simple regression model, it is just the square of the correlation between the dependent and independent variables, which is commonly denoted by "r". An R^2 value ranges from 0 to 1 which can be interpreted in percentage, where 0% means that the model does not explain the variability and 100% indicates that the model perfectly explains all the variability.

In this study, MAPE was not used as the evaluation metric because the Scikit-learn library handles errors differently than the common "percentage" definition of MAPE. Specifically, Scikit-learn converts the percentage error (typically from 0 to 100) into a relative error between 0 and 1 by dividing the percentage by 100. Due to how Scikit-learn scales the error values, the traditional MAPE formula did not apply, so to avoid confusion it was not used as one of the evaluation metrics.

3.8 Summary

In this chapter, methodologies to meet the research question were raised and discussed. Using hardware and software to meet the research design is also explained. This methodology underscores a systematic approach to leveraging machine learning techniques for analyzing water supply and sewage service data, contributing to improved management strategies in the Sidama region specifically the Hawassa City Water Supply and Sewerage Service Enterprise (HCWSSSE). Data collection, preprocessing, execution modeling, and evaluation and implementation of the proposed method, have been discussed.

CHAPTER FOUR

EXPERIMENTATION AND RESULT

4.1 Overview

This section deals with how the experiment is conducted in the study based on the steps mentioned in Chapter 3. This chapter will emphasize the experimental aspects of the study by comparing different Machine Learning models together, starting with data collection, pre-processing, data cleaning, normalizing, splitting the data, modeling, and evaluation analysis.

4.2 Statistics of Consumption Data

Descriptive statistics data summarizes the dataset by showing count, STD, means, min and, max. It also enables a researcher to quantify and describe the basic properties of a data set. A summary of the data is presented in Table 4.1, analytical metrics like count, mean, STD, and the calculated measure of central tendency.

Table 4.1 Descriptive Statistics of the HCWSSSE Dataset

	CustomerType	Kebele	Year	Month	MonthConsumption	Numb_Cust
count	16012.000000	16012.000000	16012.000000	16012.000000	16012.000000	16012.000000
mean	4.049588	27.938359	2012.583250	6.497689	1866.871409	174.021109
std	4.008895	14.241868	1.715657	3.451303	3357.637401	384.358266
min	0.000000	0.000000	2010.000000	1.000000	0.000000	0.000000
25%	1.000000	16.000000	2011.000000	3.000000	37.000000	2.000000
50%	2.000000	28.000000	2013.000000	6.000000	301.000000	10.000000
75%	9.000000	40.000000	2014.000000	9.000000	1929.250000	81.000000
max	10.000000	53.000000	2015.000000	12.000000	28317.000000	2488.000000

According to Table 4.1 above, there are 16012 values in total, which indicates 16012 values are in the dataset as a whole. It is presumed that if a missing value exists in the dataset the count and the number of data frames will not be equal, however, in this case, it is not seen. The data considered missing is either set to 0 (zero) when data is missing or the encoder records with a value.

4.3 Data Pre-Processing

The data was provided in a Portable Document Format (PDF) format from the Hawassa City Water Supply and Sewerage Service Enterprise (HCWSSSE) and categorized into years. To attain a usable final dataset for implementation several steps were carried out. The data was converted to an Excel spreadsheet and saved to a '.csv' file format. The data could have been stored in any number of forms or formats but in the study, the Comma-Separated Values (CSV) was more applicable as it's mostly used in data science so it was selected. The power query function was essential to merge the individual Excel files into one final dataset, which was used to arrange the data in a continuous order. The data preprocessing section includes data cleaning, Encoding, Normalizing the data (scaling), Feature selection, Sampling, and, finally data splitting.

4.3.1 Data Cleaning

The cleaning of data helps us to correct the messy data that can be found and rectify the problems beforehand so that the model can use the data and give us an output. Data cleaning is a key part of data science, it is mostly used in error prevention strategies before they occur [50]. In the study, the presence of 'NaN' or 'Na' values in the dataset was not seen. This implies that the HCWSSSE was either manually filling in the missing data or automatically converting it to zero '0' in the dataset. However this case, removing the 'NaN' values was implemented using the 'panda.drop' function in case there were hidden missing values.

```
# Drop any row or column with fully missing values

print (df.shape)
# drop  any column with 100% nan (if any)
df.dropna(axis=1, how='all', inplace=True)
# drop  any row with 100% nan (if any)
df.dropna(axis=0, how='all', inplace=True)
# Check for shape changes
print (df.shape)
```

Figure 4.1 Code snippet of Dropping NaN values

4.3.2 Detecting Outliers

Finding abnormally large data values is expected when dealing with real-world data in data cleaning. These data do not align with the existing data, they are considered outliers. Outliers in the dataset are due to the data being collected from different kebeles (the smallest administrative unit in Ethiopia), each containing diverse customer types. As a result, the consumption behavior from one kebele to another may vary leading to discrepancies in water usage, with some kebeles

exhibiting unusually high or low consumption compared to others, thereby producing outliers in the data. These outliers were identified and addressed using the z-score method mentioned in Fig 4.2. Subsequently, they were removed from the final dataset. A total of 433 values were removed. Fig 4.3 illustrates the spread in data for every month and visual explanations of its distribution. The box plot in Fig 4.4 and Fig 4.5 shows the dataset distribution using the box plot before and after outliers have been removed, allowing for a more refined analysis of the underlying patterns.

```
#Z-score method
# find the limits for the consumption
upper_limit = df['MonthConsumption'].mean() + 3*df['MonthConsumption'].std()
lower_limit = df['MonthConsumption'].mean() - 3*df['MonthConsumption'].std()
print('upper limit:', upper_limit)
print('lower limit:', lower_limit)

# trimming - delete the outlier data
new_df = df.loc[(df['MonthConsumption'] <= upper_limit) & (df['MonthConsumption'] >= lower_limit)]
print('before removing outliers:', len(df))
print('after removing outliers:',len(new_df))
print('outliers:', len(df)-len(new_df))
```

Figure 4.2 Code snippet of outlier removing using z-score method

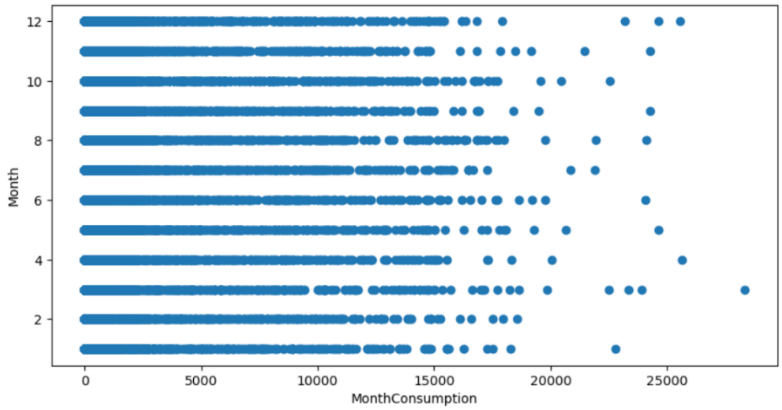

Figure 4.3 Scatter Plot of the month Consumption of the HCWSSSE dataset

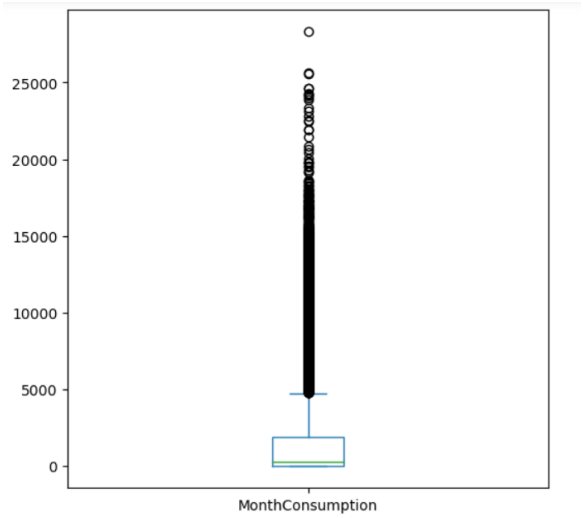

Figure 4.4 Detecting the Outliers using a box plot

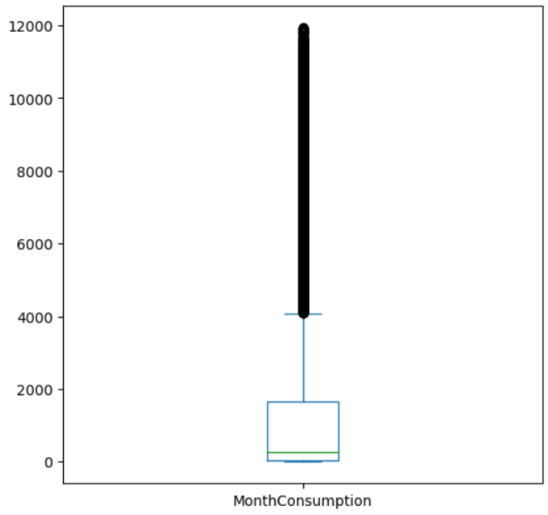

Figure 4.5 After Removing Outliers depiction using a box plot

4.3.3 Parsing Dates

Initially, the date was inputted as 2 variables (month and year) and the data set had no ‘day’ value. To address this the days were set to the 20th day (the reason was that the day that HCWSSSE usually updated the monthly data on the 20th day) and this was used for the month and year to be combined into another column formed as ‘CurrentDate’. The data type of the current date, "dtype: object," was changed to "datetime64."

As mentioned above the date displayed in the database is in the Ethiopian Calander; to avoid confusion the date was used in the budget year method the month `'Hamle/ሐምለ'` to `'Sene/ሰኔ'` than the months `'Meskerem/መስከረም'` to `'Nehase/ነሐሴ'` which is the beginning and end of the Ethiopian Calander.

```
# To impute this time series, we have to create a CurrentDate column, assuming the day is 20

df['CurrentDate'] = pd.to_datetime(dict(year=df.Year, month=df.Month, day=20))
df.info(all)
```

Figure 4.6 Code snippet Imputing the date

4.3.4 Encoding

It is used to transform Categorical variables into numerical forms of data. In the HCWSSSE dataset, Categorical variables are mentioned in Chapter 3. Machine learning algorithms work better on numerical data as they are based on mathematical equations. So, having categorical values results in the system crashing before it starts, the need to convert them to numeric form is necessary.

The study employed sklearn.preprocessing import LabelEncoder method to encode the categorical variables into numerical data. In this method, each categorical value was mapped to a numeric value. Among other encoding methods, the label encoding worked best in our case.

```
#labeling the Categorical values

label = df.select_dtypes(exclude='number').columns
for encode in label:
  df[encode] = df[encode].astype('category')
  df[encode] = df[encode].cat.codes
```

Figure 4.7 Label Encoder code snippet

A manual labeling method is implemented on the month, which converts the months to numeric values to get the ‘CurrentDate’ values for the month variable. This was done to get the budget year

calendar. With this method attaining the date was possible as the techniques did not increase the number of columns like using one-hot encoding and kept the Ethiopian budget year Calander.

```
#Convert the month to numeric values:

month_numbers = {'Hamle/ሐምሌ': 1, 'Nehase/ነሐሴ': 2, 'Meskerem/መስከረም': 3, 'Tikimt/ጥቅምት': 4,
                 'Hidar/ህዳር': 5, 'Tahsas/ታህሳስ': 6, 'Tir/ጥር': 7, 'Yekatit/የካቲት': 8,
                 'Megabit/መጋቢት': 9, 'Miazia/ሚያዝያ': 10, 'Ginbot/ግንቦት': 11, 'Sene/ሰኔ': 12, 'Pagume/ጳጉሜ': 13}

# Create another month column
df = df.assign(TheMonth=df.Month)
df.sample(5)
```

Figure 4.8 Code snippet on Conversation of the Months to Numbers

4.4 Feature selection

In the real-world dataset, we have a large number of variables that rise from the data set creating a diverse array of measurements. Feature selection, as a dimensionality reduction technique, aims to choose a small subset of the relevant features from the original ones by removing irrelevant, redundant, or noisy features [51].

The Study focuses on water consumption prediction. The correlation technique was used to identify the relevant features to water consumption, which determined the strength of the relationship between each feature. Features with higher correlation coefficients were considered more relevant. Features that were not needed for the study were manually removed. For relative feature importance, the "MonthConsumption" variable is the most important attribute in prediction modeling. To weigh the importance of the attributes ExtraTreesRegressor is used, which is part of the sklearn.ensemble package and used to assess the significance of features in the dataset. ExtraTreesRegressor can effectively identify the most impactful variables.

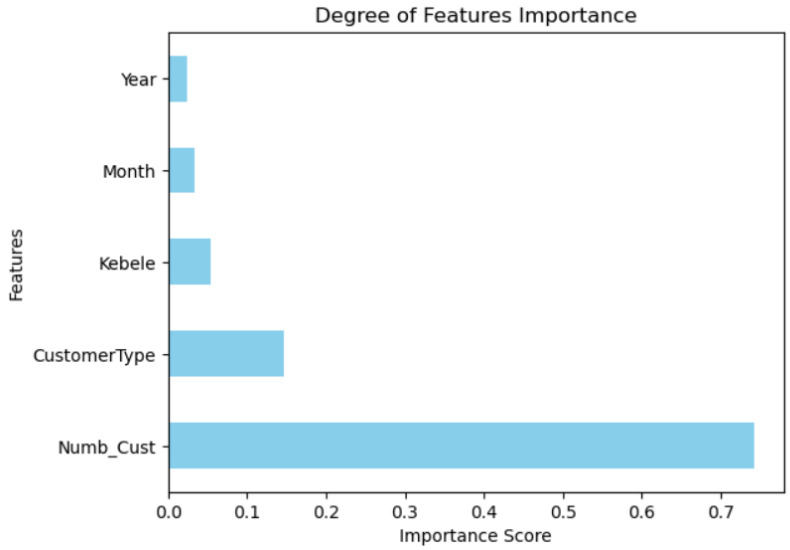

Figure 4.9 Weight of Feature Importance

Fig 4.9 demonstrates the ranking of features in terms of their importance, arranged from the least significant to the most significant. The number of customers heavily impacts the monthly consumption, as the number of customers increases so does the monthly consumption rate.

4.4.1 Correlation Analysis

Correlation quantifies the extent to which two quantitative variables, X and, Y "go together"[52]. According to Ya Lun Chow, "Correlation analysis attempts to determine the degree of relationship between variables." The Spearman Correlation method was used to identify the correlation between the continuous and categorical variables. Based on the correlation analysis and feature importance ranking, the most relevant features were selected. The heatmap shown in Fig 4.10 was created using all the variables in the dataset.

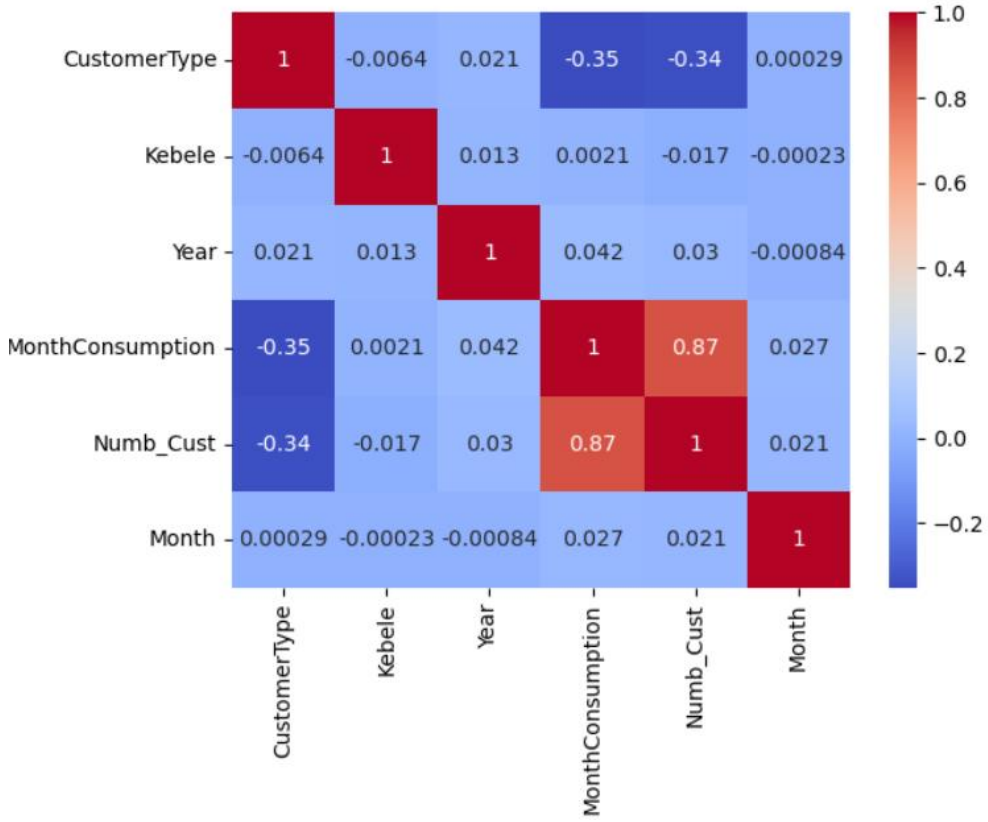

Figure 4.10 Correlation heatmap of the variables

The heat map shows the relationship between the variables in the data. The 'monthly consumption' is what the study is focused on, we can see the correlation between the consumption and the Numb_Cust. When the number of customers increases so does the consumption this is what the positive correlation implies which we can see in Fig 4.10. The CustomerType and the Kebele have

a negative correlation to the Consumption. This doesn't mean there is no effect, they have an indirect relationship with the Monthly consumption. One kebele might not have the same Numb_Cust in a specific CustomerType, this shows that even though they aren't directly impacting the monthly consumption their presence as a variable is assured.

The ConsAnnualReference and the CustAnnualReference are not visible in the correlation metrics. Since these columns corresponded to the total monthly consumption and the number of customers, they were removed during the experiment. Since they depend on the monthly consumption and number of customers respectively, they were determined that they would not be useful in forecasting the monthly water consumption.

4.5 Hyperparameter

Machine learning uses many parameters to provide results based on the user's needs to optimize the performance of each model. However, the process of choosing parameters to fit the model is a complex task. The way to reduce the difficulty of choosing parameters that provide better results or reduce learning error is to use hyperparameter tuning. Popular tuning techniques are:

- Manual search: hyperparameters are selected based on user experience.
- Random Search: Random search is a type of grid search but its parameter selected to search is based on random selection.
- Grid Search: is a traditional hyperparameter searching technique that checks all combinations of list parameters.
- Bayesian Search is a searching technique that considers the previous parameter to select a new parameter.

In this study, the Random search optimization technique was implemented for the hyperparameter tuning process. It has the advantage of reducing time and better generalization performance which became ideal for our case. In addition to time, Bayesian optimization is mostly chosen when the dataset is ≥ 100,000 with more than 100 features. But in our case, the Random search was chosen because very few hyperparameters were tuned.

4.6 Data Visualization

Data visualization is essential in today's data-driven world it enables better understanding and communication of complex information. The visual illustration from the final dataset from 2010 - 2015 E.C. is shown in Fig 4.11, the matplotlib library was used to attain the line plot figure. Fig 4.12 shows the visual illustration of the data when the data is grouped to get a time series continuous data. Plotting helps represent the visual variation and show relationships between variables and their variation over the years. The plots include all the data of every kebele and customer type in tandem and gives the line plot figures below.

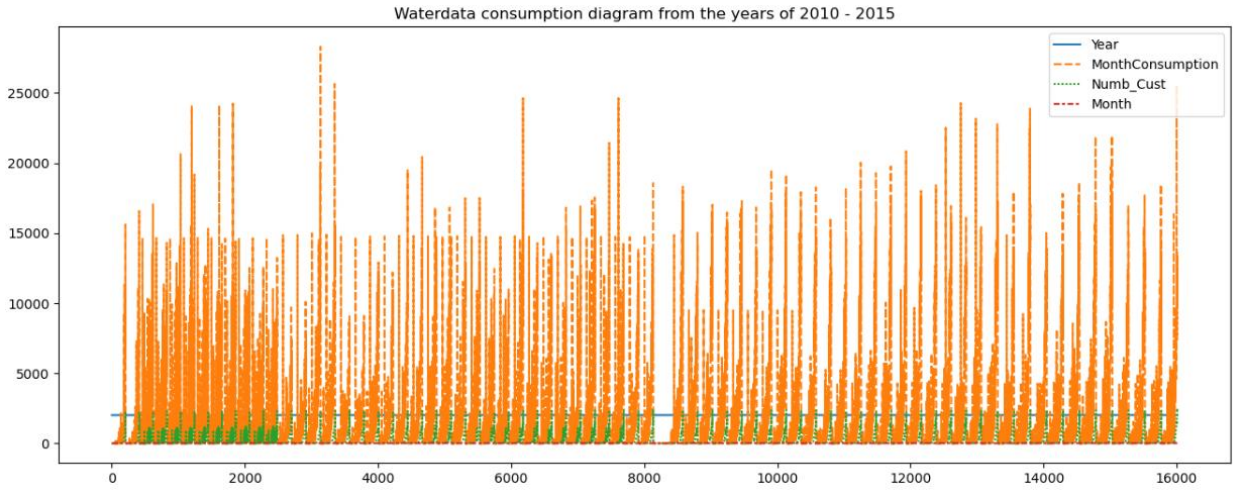

Figure 4.11 Line plot without index of the date

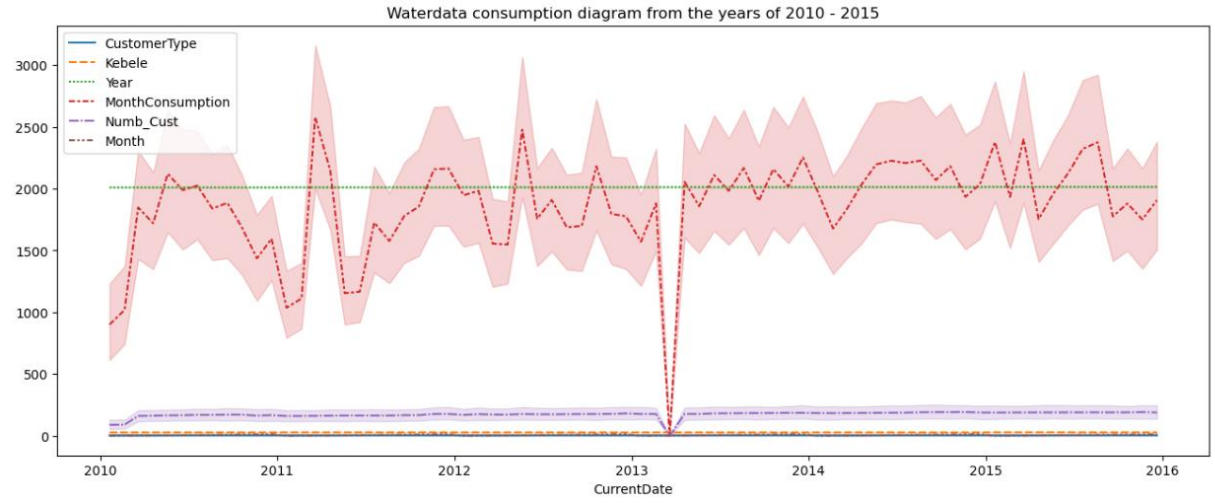

Figure 4.12 Line Chart to visually inspect the data from 2010-2015 E.C

Fig 4.12 shows the overall sequence of consumption that occurred, the 'dotted Red line' represents the Month's Consumption, the 'dotted purple' represents the number of customers, the 'Blue' shows the customer types encoded, the 'dotted green' shows the year while the 'double dotted

purple' shows the encoded month. This tells us the general consumption trends that have passed through the years, in the water consumption records of the last 5 years for all 54 kebeles and 11 customer types.

We have seen the visual representation of the dataset; Table 4.2 depicts the sample data for the final dataset. The sample data presented here is from the final data set used for modeling for machine learning. The total count is (16012 rows × 10 columns) before removing the outliers and columns not used in the machine learning models.

Table 4.2 A sample of data records on the final dataset

	CustomerType	Kebele	Year	ConsAnnualReference	CustAnnualReference	MonthConsumption	Numb_Cust	Month	TheMonth	CurrentDate
0	Domestic	Fara Alamura D	2010	511	53	0	0	1	Hamle/ሐምለ	1/20/2010
1	Domestic	Fara Gudumale	2010	13	10	0	0	1	Hamle/ሐምለ	1/20/2010
2	Public Enterprises	Dume	2010	2532	50	0	0	1	Hamle/ሐምለ	1/20/2010
3	Public Enterprises	Fara 1 Guduma	2010	126	26	0	0	1	Hamle/ሐምለ	1/20/2010
4	Public Enterprises	Fara 2 Guduma	2010	2087	74	0	0	1	Hamle/ሐምለ	1/20/2010
...	...	...	...	...	...	...	...	...	...	...
16007	Domestic	Dato Odahe 1	2015	102596	19501	7397	1603	12	Sene/ሰኔ	12/20/2015
16008	Domestic	Hiteta D	2015	154168	21395	11198	1801	12	Sene/ሰኔ	12/20/2015
16009	Domestic	Guye Stadium C	2015	166429	25006	10811	2179	12	Sene/ሰኔ	12/20/2015
16010	Domestic	Dato Odahe	2015	151228	27635	12787	2488	12	Sene/ሰኔ	12/20/2015
16011	Domestic	Feladelfia	2015	176784	28207	13312	2366	12	Sene/ሰኔ	12/20/2015

16012 rows × 10 columns

4.7 Data Normalization

As mentioned above the data is converted from a PDF to Excel and saved as a Comma-separated File (CSV). Normalization is one of the most common approaches to scale the data to a common scale. Scaling or normalization is important in machine learning algorithms like Support Vector Regressor (SVR) from the Support Vector Regressor (SVR) which rely on calculating distances between data points and are sensitive to the scale of input features. Scaling techniques were employed specifically on the individual column. The data is normalized by applying the sklearn MinMaxScaler() function and in the SVR modeling the StandardScaler() was implemented on the particular variables of the HCWSSSE dataset.

```
#Min max scaling for better experimentation
from sklearn.preprocessing import MinMaxScaler
minmax_scale = MinMaxScaler()
coll = ["Numb_Cust"]
df[coll] = minmax_scale.fit_transform(df[coll])
df.head()
```

Figure 4.13 min-max scalar code snippet

MinMax is a data normalization technique[53] uses the scales between 0 and 1 which reduces runtime and helps us understand the data easily, using each feature's minimum and maximum value. Normalizing the data ensures that the necessary features contribute equally by improving model performance and optimization.

4.8 Data Splitting

The final data set was obtained to split the data into training, validating, and testing datasets. In this step splitting the data was conducted using sklearn.model_selection train_test_split function, into a training 80%, validation 10%, and testing 10% dataset from the final data set. This dataset distribution ensures the model has enough to learn from, while still being able to evaluate and tune it effectively. How data separations are prepared varies greatly depending upon the analytic objectives for which they are required and the specific learning techniques and software by which they are to be analyzed.[46]

```
# splitting the dataset to X and Y
X = new_df.drop('MonthConsumption',axis=1)
y = new_df['MonthConsumption']

# First split: 80% train and 20% test (validation + test)
X_train,X_test,y_train,y_test  = train_test_split(X,y, test_size=0.20, random_state=0)

# Second split: 50% of test for validation and 50% for testing
X_val,X_test,y_val,y_test = train_test_split(X_train, y_train, test_size=0.50, random_state=0)
```

Figure 4.14 train_test_split Code snippet

The dataset used in this study consists of five (5) years of monthly water consumption data from Hawassa City (HCWSSSE dataset). To ensure consistency and eliminate potential bias in comparing the results, the same dataset was employed across all algorithms. In contrast, other studies have typically allocated their data into training, validation, and testing sets, using a distribution of 70%, 15%, and 15%, or 60%, 20%, and 20% respectively [23]. While this approach can be advantageous when a larger validation or testing set is required, it can also lead to insufficient data for training, which may negatively impact model performance, particularly in complex deep-learning models. It is important to note that the dataset in this study is limited to monthly data, which may further constrain the model's ability to learn effectively from the available information.

4.9 Modeling

4.9.1 Machine Learning Algorithms

In the Study, 4 methods of Machine learning algorithms were used to test which better predicts water consumption. They are the Random Forest regressor (RF), Linear Regression (LR), Support Vector Regressor (SVR), and the XGBoost were used in the modeling process. The study investigates the performance of a base model (default parameters) without implementing the hyperparameters initially, serving as a benchmark for our analysis. Hyperparameter implementation is then used to get a better-performing model. By comparing the outcomes of these two approaches, the study will show the impact of hyperparameter adjustments on model accuracy and generalization capabilities. The variable represented in monthly consumption serves as the dependent variable and is identified as the target variable within the scope of this research.

4.9.1.1 Random Forest (RF)

The random forest regression is from the sklearn.ensemble which we call the Random Forest Regressor method. Once the final dataset was prepared, the RF algorithm was imported and tested. The study implements the base model, mentioned above using the default parameters to see the difference between the base model and the hyperparameter selection. To achieve the desired values, the 'Month Consumption' is used. Following this, hyperparameter tuning was applied to the random forest (RF) to reduce the risks of underfitting or overfitting.

Finally, we will evaluate the models using the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) evaluation Metrics. Table 4.3 shows the evaluation metrics of the base model using the default parameters.

Table 4.3 Evaluation metric of RF on the base model

Evaluation of RF on the Base Model	
MSE	65337.34
MAE	117.13
RMSE	255.61

The predictive ability was evaluated using the R^2, the training dataset has an evaluation metric of 99.00%, the Validation 98.99%, and the testing 99.01% using the default parameters.

The metrics of the base model RF training, Validation, and test datasets:

Feature	Dataset	The r2
MonthConsumption	Training	99.00%
	Validation	98.99%
	Testing	99.01%

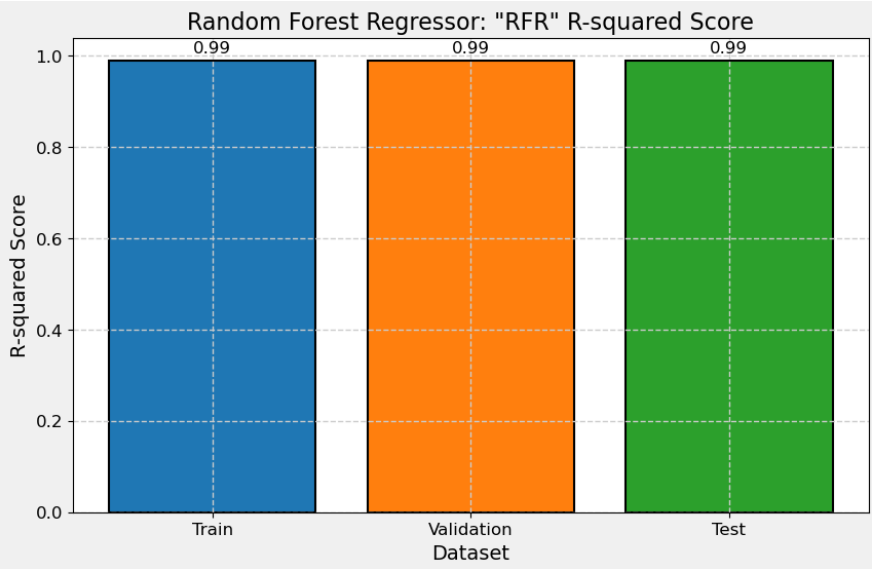

Figure 4.15 Bar plot of base model RF using default parameters

The random search cross-validation (CV) approach was employed to optimize the hyperparameters for the Random Forest model. This method was chosen to minimize the computational time required for model training. The code snippet search for the best parameters is shown in Fig 4.16.

```
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [5, 10, 15, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

rf = RandomForestRegressor(random_state=42)
randomized_search = RandomizedSearchCV(rf, param_grid, n_iter=10,
                                       cv=5, scoring='neg_mean_squared_error',
                                       random_state=42)
randomized_search.fit(X_train, y_train)
```

Figure 4.16 Code snippet of the randomized search for RF

In Table 4.4, the randomized search implementation recommended the best potential outcomes for the RF model.

Table 4.4 The best parameters used for the RF

No.	Parameters	Data/Technique used
1	n_estimators	200
2	min_samples_split	5
3	min_samples_leaf	2
4	max_depth	15

The evaluation metrics of the RF model using the hyperparameters are shown in Table 4.5.

Table 4.5 Evaluation metrics using the best selected parameter in RF

Evaluation of RF using the best parameters	
MSE	182584.97
MAE	199.78
RMSE	427.29

The Train test and validation data sets have a rate that is closely related to each other this is shown in Fig 4.17. They are closely related around 97% of the Prediction rate.

```
The metrics of the model RF using the hyperparameters:

Feature              Dataset       The r2
=========================================
MonthConsumption     Training      97.23%
                     Validation    97.24%
                     Testing       97.22%
=========================================
```

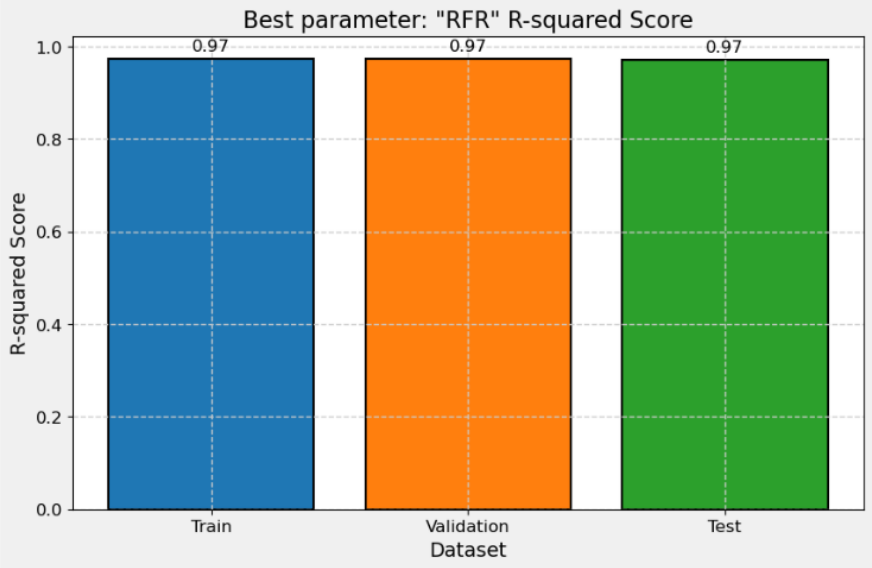

Figure 4.17 The best parameter used in RF

4.9.1.2 Linear regression (LR)

The linear regression (LR) is imported from the sklearn.linear_model. The HCWSSE dataset was called and executed for the linear regression. The base model with default hyperparameters was initially observed, then the hyperparameters were implemented to find the best result. Table 4.6 shows the evaluation done on the LR base model.

Table 4.6 Evaluation metrics of LR on the base model

Evaluation of LR for the base model	
MSE	1442539.27
MAE	716.56
RMSE	1201.05

The Train, Validation, and Test datasets have a Prediction rate 78% that is closely related, this is shown in Fig 4.18 in the bar chart.

```
The metrics of the base model LR training, Validation, and test datasets:

Feature              Dataset       The r2
=========================================
MonthConsumption     Training      78.26%
                     Validation    78.46%
                     Testing       78.05%
=========================================
```

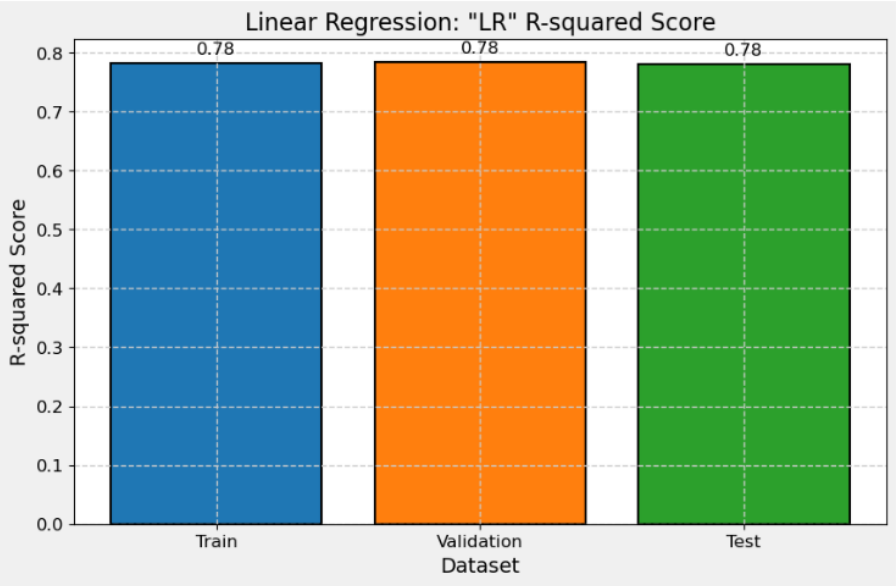

Figure 4.18 R^2 score of the base model LR

A random search cross-validation (CV) was implemented, and the Ridge () function was called unlike the RF to find the best hyperparameters of the LR model. The hyperparameters utilized for the model selected from the random search are presented in Table 4.7 below, along with their corresponding values.

Table 4.7 Best parameters used for LR

No.	Parameters	Data/Technique used
1	alpha	2.059428845085949
2	copy_X	False
3	fit_intercept	False
4	max_iter	10000
5	solver	svd

Using the hyperparameters to model the LR, suggested the best possible model outcome which resulted in a seemingly better outcome as we can see in Table 4.8 below.

Table 4.8 Evaluation metrics using the best parameters in LR

Evaluation of LR the best parameters	
MSE	1430340.24
MAE	720.02
RMSE	119.96

```
The metrics of the LR model using the hyperparameters:

    Feature             Dataset       The r2
    ========================================
    MonthConsumption    Training      78.18%
                        Validation    78.38%
                        Testing       77.98%
    ========================================
```

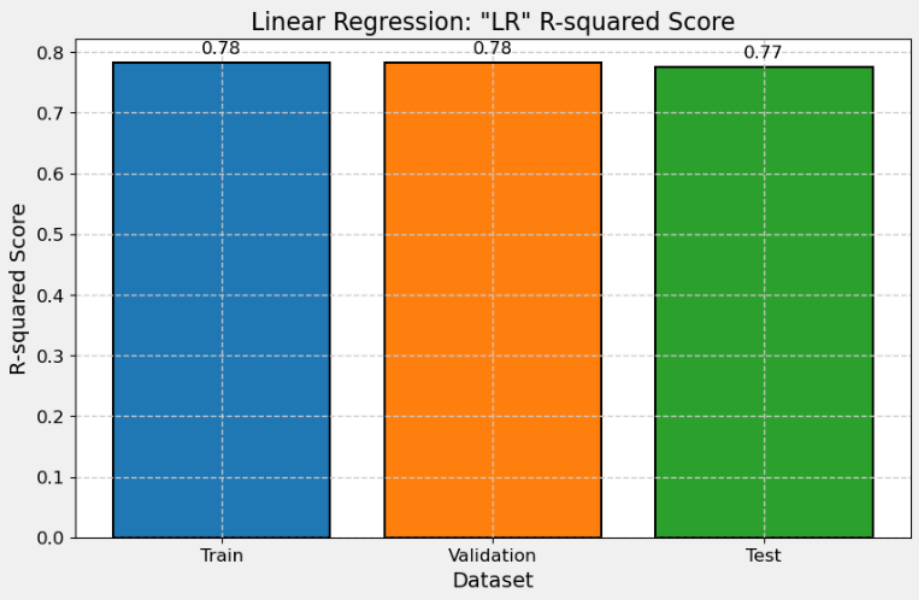

Figure 4.19 R^2 score for the best parameters in LR

4.9.1.3 Support Vector Regressor (SVR)

The Support Vector regressor (SVR) is imported from the sklearn.svm library. Just like the RF and LR the "MonthConsumption" is targeted. The Standard Scalar () function was used to scale the values before training the model. The base modeling is initially done on SVR, the best parameters are then selected using the random search CV. Let us see the base model without using the parameters in Table 4.9.

Table 4.9 Evaluation Metrics of SVR on the Base Model

Evaluation of SVR on the base model	
MSE	2076.54
MAE	1295.05
RMSE	45.56

```
The metrics of the SVR model on the training, Validation, and test d
atasets base model:

          Feature              Dataset       The r2
          ========================================
          MonthConsumption     Training      75.21%
                               Validation    75.17%
                               Testing       75.25%
          ========================================
```

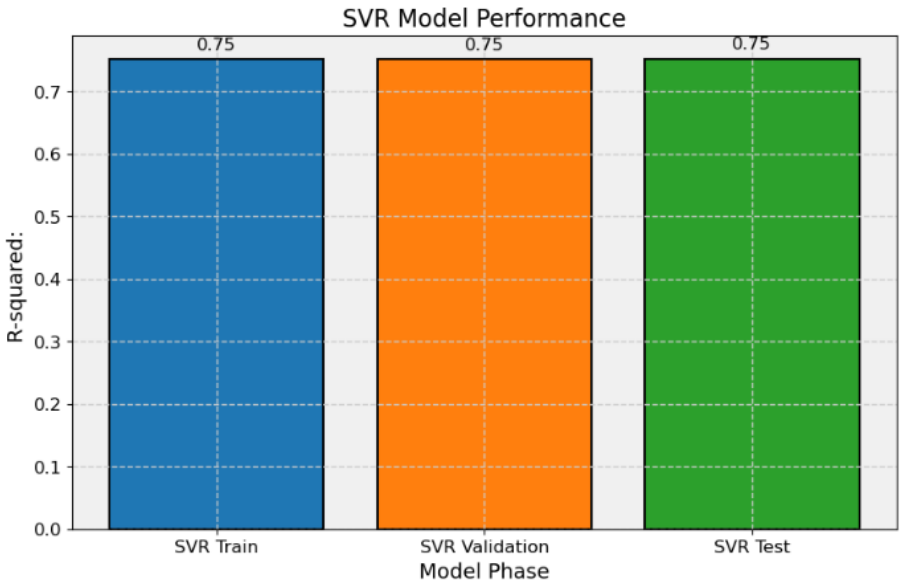

Figure 4.20 R^2 Score for the SVR on the base model

Selected parameters used for modeling the SVR model and Evaluation can be viewed in Table 4. 10 which is used in modeling.

Table 4.10 Best Parameters Used for SVR

No.	Parameters	Data/Technique used
1	kernel	rbf
2	Gamma	scale
3	Epsion	0.01
4	C	100

Table 4.11 shows the evaluation metrics on SVR using the best hyperparameters selected.

Table 4.11 Evaluation Metrics using the parameters on SVR

Evaluation of SVR using the best parameters	
MSE	617.41
MAE	574.64
RMSE	24.84

```
The metrics of the SVR model using the hyperparameters:

Feature              Dataset       The r2
=========================================
MonthConsumption     Training      79.37%
                     Validation    79.92%
                     Testing       78.81%
=========================================
```

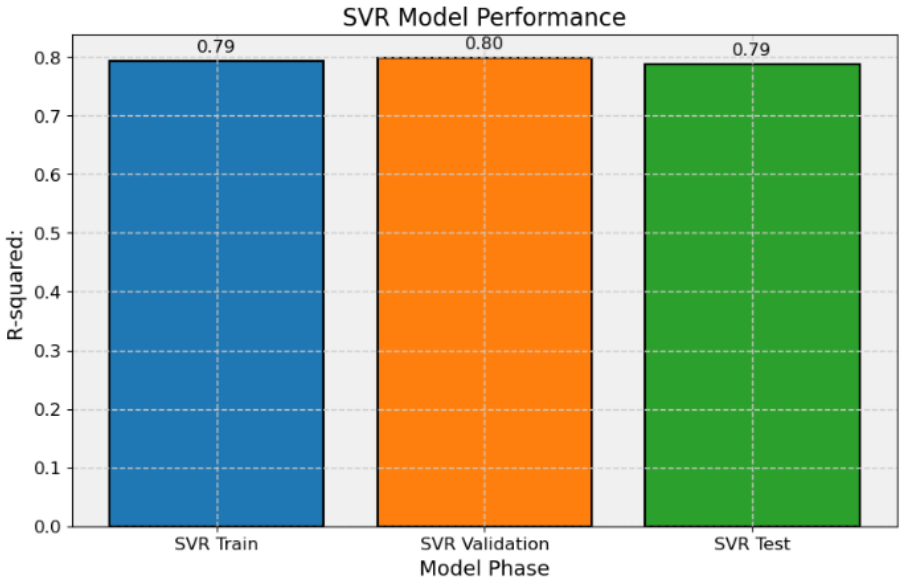

Figure 4.21 R^2 Score on the hyperparameter for the SVR

4.9.1.4 Extreme Gradient Boosting (XGBoost)

Within the Gradient Boosting framework, XGBoost is a machine learning algorithm that uses the radiant and uniform from the scipy.stats library for the parameters is imported from the XGBoost library. Modeling using the default parameters resulted in an evaluation shown in Table 4.12.

Table 4.12 Evaluation Metrics of XGBoost on the base model

Evaluation of XGBoost for the base model	
MSE	167928.35
MAE	218.26
RMSE	409.79

```
The metrics of the XGBoost model using the base model:

Feature             Dataset       The r2
=========================================
MonthConsumption    Training      97.47%
                    Validation    97.50%
                    Testing       97.44%
=========================================
```

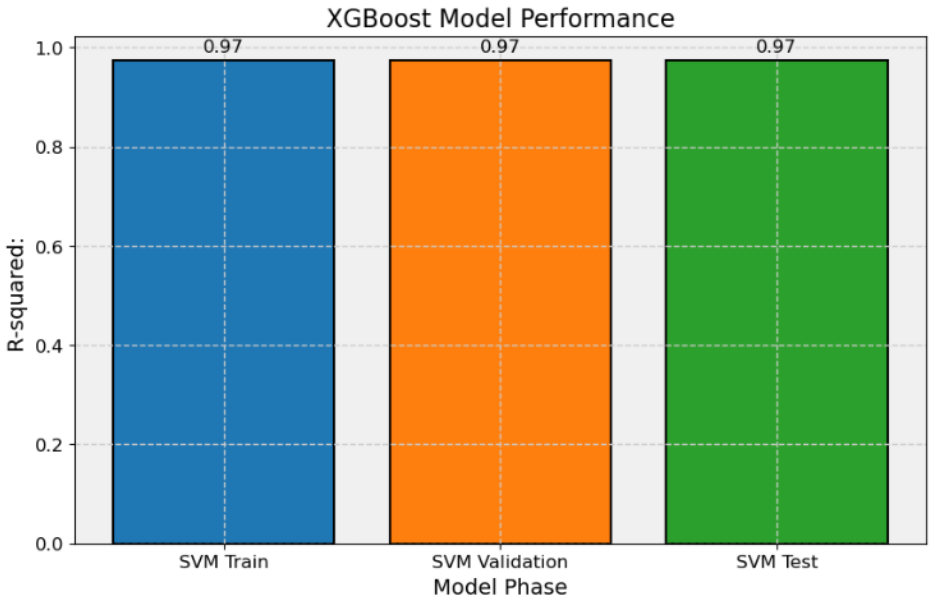

Figure 4.22 R^2 score for the XGBoost base model

The random search cv is used on XGBoost to choose the best hyperparameters. Table 4.13 shows the parameters used with its corresponding value, and Table 4.14 shows the evaluation metrics.

Table 4.13 Best parameter used for XGBoost

No.	Parameters	Data/Technique used
1	colsample_bytree	0.984
2	gamma	0.775
3	learning_rate	0.291
4	max_depth	7
5	min_child_weight	4
6	n_estimators	63
7	subsample	0.863

Table 4.14 Evaluation Metrics for the XGBoost using best parameters

Evaluation of XGBoost using the best parameters	
MSE	191890.52
MAE	239.07
RMSE	438.05

The metrics of the XGB model using the hyperparameters:

Feature	Dataset	The r2
MonthConsumption	Training	97.08%
	Validation	97.07%
	Testing	97.08%

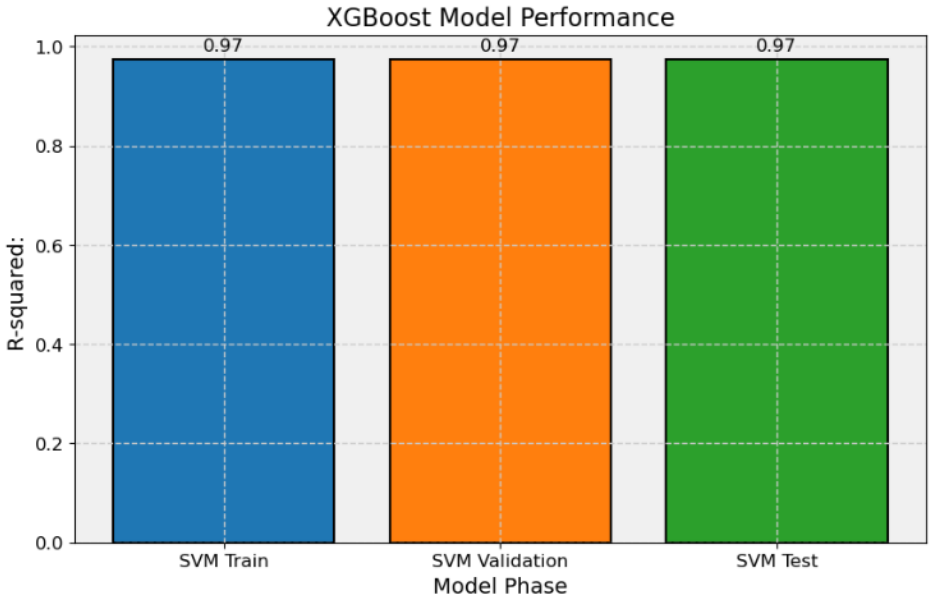

Figure 4.23 R^2 score for the XGBoost with hyperparameters

CHAPTER FIVE

EVALUATION AND DISCUSSION

This section presents the experimental results and comparative analysis of the machine learning models used for machine consumption prediction. Using the selected evaluation metrics, the performance of each model is evaluated. The research questions raised above in Chapter One are then discussed.

5.1 Experimental Result of Predictive Algorithms

The performance of the 4 machine learning models (RF, LR, SVR, and XGBoost) were evaluated using the evaluation metrics R-squared (R^2), Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). These models were trained by the HCWSSSE database from 2009 – 2015 E.C. Using machine learning methods, we would make an economic decision on what steps to take for the HCWSSSE dataset.

5.2 Results

5.2.1 R-Squared (R^2) Results

The performance of the models in the tables below shows the evaluation metrics on the training, validation, and testing dataset. The results were obtained from the models used on the final dataset. The predictability rate is measured using the R-square (R^2) on the dataset. Table 5.1 shows the base model performance using the default parameters on the dataset while Table 5.2 shows the use of hyperparameters.

Table 5.1 Evaluation Metric R^2 on all the base models

Evaluation Metric R^2	RF	LR	SVR	XGBoost
Training	99.00%	78.26%	75.21	97.47%
Validation	98.99%	78.46%	75.17%	97.50%
Testing	99.01%	78.05%	75.25%	97.44%

Table 5.1 presents the R^2 score of the models tested to show how well the models performed in the model using default parameters. RF performed the best with 99% on the training datasets 99.01% on the testing, indicating an excellent predictive performance that is well-suited for the problem as it can explain the variance in the data. The XGBoost also performed well at 97.47% which reflects a good model accuracy though with fine-tuning could perform even better. The LR showed some performance around 78.26% but was less effective than the RF and XGBoost models. The SVR performed moderately lower on the base model across all the datasets, suggesting that it needs adjustment by trying different parameters as it's not as effective for this problem.

Overall, the RF and XGBoost were effective in modeling by providing better predictive capability than the others. While the LR had moderate predictive capability, the SVR required some hyperparameter tuning.

Table 5.2 Evaluation Metric R^2 on all models using hyperparameters

Evaluation Metric R^2	RF Hyperparameter	LR Hyperparameter	SVR Hyperparameter	XGBoost Hyperparameter
Training	97.23%	78.18%	79.37%	97.08%
Validation	97.24%	78.38%	79.92%	97.07%
Testing	97.22%	77.98%	78.81%	97.08%

The results shown in Table 5.2 shows the evaluation metrics measuring the R^2 on models using hyperparameters, the higher the percentage indicated better performance. Randomized search cross-validation (CV) was used in finding the hyperparameters. Just like the performance of the base models the RF and XGBoost performed better, with values around 97.23% and 97.08% respectively on the training datasets. The RF has slightly better performance than XGBoost while the LR and SVR being lower than the other models had around 78.18% and 79.37% respectively. SVR showed a significant increase in performance after hyperparameter tuning, which indicates that hyperparameters were needed while SVR modeling. Overall, the RF and XGBoost performed exceptionally well in explaining the variance of data after hyperparameter tunning.

Comparing the models in the tables above, the RF and XGBoost are better performing in prediction capability, but the base model raises concerns of potential overfitting, especially in the RF. Table 5.2 indicates that the use of hyperparameters enhances the model generalization, and RF still performs well among the rest and reduces overfitting. The SVR improved very much, and the LR maintains moderate effectiveness. The XGBoost is a reliable choice showing strong predictive capability next to RF.

5.2.2 Error Evaluation Metric

The following results were obtained from the models tested on the final dataset; it contains the error evaluation metric done on the monthly consumption. Using the machine learning algorithms the study shows the error rate there is using these models. Table 5.3 shows the error evaluation using MSE, MAE, and RMSE for the base model, and Table 5.4 shows the models using hyperparameters. It is important to note that both tables present the overall performance metrics.

Table 5.3 Evaluation Metrics on all the base models

Error Evaluation Metric	RF	LR	SVR	XGBoost
MSE	65337.34	1442539.27	2076.54	167928.35
MAE	117.13	716.56	1295.05	218.26
RMSE	255.61	1201.05	45.56	409.79

The lower the value in the Error Evaluation metrics, the better the performance. The MAE shows the average error of the overall dataset. Let's discuss error evaluation metrics.

MSE: The SVR has the lowest MSE (2076.54), showing that it's more accurate among the models based on the MSE. RF performed moderately (65337.34), while XGBoost (167928.35) and LR (1442539.27) had the highest error rate suggesting a poor performance in squared error.

MAE: RF followed by XGBoost resulted in a lower error rate, and performed a better MAE prediction of 117.13 and 218.26 respectively, compared to the other models. LR (716.56) is

moderately high while, SVR (1295.05) had the highest error rate for the base model, though having the lowest MSE, suggesting that it has a higher average error in prediction performing poorly.

RMSE: The SVR showed an unexpectedly low RMSE value (45.56) despite being the highest in MAE, this suggests that the prediction is concentrated around the mean and predicts a narrow range of values poorly. The RF (255.61) still showed a lower error rate performing better than XGBoost (409.79) and LR (1201.05).

The evaluation of the base model suggests the RF has the best overall performance. The XGBoost follows after the RF, while the LR continues. The SVR model shows that it needs more exploration and further tuning to improve its predictive power.

Table 5.4 Evaluation Metrics of all the models using hyperparameters

Evaluation Metric	RF Hyperparameter	LR Hyperparameter	SVR Hyperparameter	XGBoost Hyperparameter
MSE	182584.97	1430340.24	617.41	191890.52
MAE	199.78	720.02	574.64	239.07
RMSE	427.29	119.96	24.84	438.05

The findings in Table 5.4 show the evaluation metrics derived from the use of hyperparameters on the models, which enhanced the algorithm's performance in some cases. Let us discuss the Error Evaluation.

MSE: The LR has the highest values (1430340.24) of MSE. The RF and XGBoost have a relatively high MSE being 182584.97 and 191890.52 respectively, indicating that squaring showed a discrepancy between the predicted and actual values but lower than LR. SVR showed the lowest suggesting better predictions in terms of squared errors.

MAE: The RF (199.78) performed the best in MAE followed by the XGBoost (239.07) with lower error values. The LR (720.02) has the highest MAE showing a larger average error than the other models, despite the low MSE. SVR (574.64) has a moderately high value among the models but is slightly better than LR.

RMSE: SVR again has the Lowest RMSE (24.84) indicating a smaller prediction error when taking into account the target variable. The RF (427.29) and XGBoost (438.05) showed a moderate RMSE, while the LR (119.96) had a surprisingly low RMSE despite its high MSE, indicating variability issues in error.

Based on the evaluation the SVR showed better performance, particularly in minimizing the errors, but this doesn't translate into prediction. The RF shows a better stable prediction with a lower mean average despite its higher MSE. The XGBoost has a higher error metric than SVR but is still better than LR. The LR performed the worst overall, indicating it was not suitable for the task.

In terms of MAE, comparing the two tables the RF has a better average error and demonstrates a slight increase from Table 5.3 to Table 5.4, moving from 117.13 to 199.78 after hyperparameter tunning. The SVR significantly improved its MAE from 1295.05 to 574.64 indicating that the hyperparameter tunning was better in SVR. The LR showed higher evaluation metrics showing that it is less reliable. Overall, the RF appeared to be more accurate in terms of absolute error in this case.

In other cases of deep learning models like long short-term memory (LSTM), the performance is evaluated as having a value of MAE of 1295.05, MSE 6540448.84, and RMSE 2939.98. We can observe that comparing it to the other models above (RF, LR, SVR, and XGBoost) LSTM has exhibited a higher value of MAE, which suggests that the model may be experiencing greater overfitting using LSTM. This indicates that, while deep learning models like LSTM can be applied to the dataset, their performance might not be optimal in this particular instance. Although not part of the current study, future work could explore additional features or modifications to the model to improve the loss rate and mitigate overfitting. Below we can see the Epoch loss reduction done on LSTM using the dataset.

```
Epoch 1/10
390/390 [==============================] - 5s 5ms/step - loss: 8850537.000
0 - val_loss: 8758454.0000
Epoch 2/10
390/390 [==============================] - 1s 3ms/step - loss: 8811693.000
0 - val_loss: 8726964.0000
Epoch 3/10
390/390 [==============================] - 1s 3ms/step - loss: 8781347.000
0 - val_loss: 8697900.0000
Epoch 4/10
```

```
390/390 [==============================] - 1s 3ms/step - loss: 8752465.000
0 - val_loss: 8669981.0000
Epoch 5/10
390/390 [==============================] - 1s 3ms/step - loss: 8724574.000
0 - val_loss: 8642439.0000
Epoch 6/10
390/390 [==============================] - 1s 3ms/step - loss: 8696437.000
0 - val_loss: 8615423.0000
Epoch 7/10
390/390 [==============================] - 1s 3ms/step - loss: 8669964.000
0 - val_loss: 8588752.0000
Epoch 8/10
390/390 [==============================] - 1s 3ms/step - loss: 8642948.000
0 - val_loss: 8562566.0000
Epoch 9/10
390/390 [==============================] - 1s 3ms/step - loss: 8616173.000
0 - val_loss: 8536307.0000
Epoch 10/10
390/390 [==============================] - 1s 3ms/step - loss: 8589760.000
0 - val_loss: 8510441.0000
195/195 [==============================] - 1s 1ms/step

Best Hyperparameters:
{'lstm_units': [64, 128, 256], 'dropout_rate': [0.2, 0.3, 0.4], 'lstm_unit
s_2': [32, 64, 128], 'dropout_rate_2': [0.1, 0.2, 0.3], 'learning_rate': [
0.001, 0.0005, 0.0001], 'optimizer': ['adam', 'rmsprop', 'sgd'], 'epochs':
[10], 'batch_size': [32]}
```

5.3 Research Question Discussion

Research questions raised in the first chapter were answered below.

1. How to develop a machine learning model for water consumption prediction?
2. Which features are important for a better prediction model of water consumption in urban areas?
3. Which machine learning algorithms perform best for water consumption prediction?

Answer for Q1: In a time-series approach data used is in a sequence of time, or where you have data that is indexed or graphed concerning time, the 'time' acts like an index variable to the model to get a result. In this study, that was not the case as time could not act like an index in the model. This resulted in the modeling for the prediction of water consumption to use historical data to forecast the next data. Regression Algorithms work best in continuous data as they are kept promptly due to the continuity of the data. The nature of the data in the study uses the regression algorithm to predict water consumption. Understanding the problem will enable a deep dive into

what to attain from the situation and what type of data is needed to tackle this problem. Once we attained such data, we employed exploratory analysis and visualized what the data might be as we can observe from the visualization of the HCWSSSE dataset. Now that we have the data and can select the feature on which data should be used to understand what to do with the data, we split the data into train test splits as mentioned in previous Chapters concerning data splitting.

Model selection based on the complexity of the data can be implemented after selection modeling can be done. The algorithms used are regression algorithms like the LR, RF, SVR, and XGBoost. The historical data played an important role in visualizing and figuring out what methods to implement in data cleaning and normalization. After base model training was evaluated, hyperparameter tuning was implemented to ensure they performed to the best of their ability. Once this is done the model evaluation metrics were evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) before deployment. This approach enhances the safeguarding of real-time prediction monitoring, as it necessitates the continuous assessment and updating of the model's performance. One can construct a machine learning model to predict water usage by following the steps above. Each stage is essential for ensuring the results are precise, dependable, and beneficial for informed decision-making for better water management.

Key points: Data Collection, Data Preprocessing, Exploratory Data Analysis, Splitting the Data, Model Selection, Model Training, Model Evaluation, Model Testing, Model Deployment.

Answer for Q2: Water consumption is influenced by many factors from the extent of the user's behavior to physical location. Identifying important features for predicting water consumption in urban areas involves considering various factors that convey usage patterns. Feature selection can show which feature is important to the dependent variable. The study's demographic variables and Socioeconomic Indicators consist of different kebeles and customer types. In each kebele, there are the individual customer types as we can see in the HCWSSSE dataset, there are 54 kebele, and inside each kebele, there are 11 customer types. The usage pattern in these sectors indicates how much water is consumed in a given time; historical data provides context in predicting the consumption rate for the future.

Feature selection was done using the ExtraTreesRegressor. The important feature found in the study was the Number of customers "Numb_Cust", when the number of customers in the city

increases unforeseen causes impact water usage, as mentioned in the introduction part of the study. The behavior of the customers could vary in different climates and external factors. In the study's case, the location is in Hawassa, Ethiopia and the weather pattern differs from the west, the winter (Keremt "Amharic: ክረምት") is the rainy season, customers tend to use rain and groundwater than the one provided to them by the HCWSSSE, though recently this is gradually changing. The Monthly consumption was the target variable to predict monthly water consumption this implies that consumption is the dependent variable. The data presented is continuous and organized in months and years, showing water usage at that specific time. Furthermore, historical data can be leveraged for feature engineering, where lagged variables such as previous months or years' usage are created to enhance the model's predictive capability.

Answer for Q3: Exploratory data analysis (EDA) would give better insight into selecting which model would work best in machine learning, it delves into understanding the data and relationships before selecting. By applying multiple models it's possible to test and compare their performance. In machine learning the choice depends on the dataset. Machine learning algorithms, which thrive on large datasets, benefit significantly from the richness of historical data, leading to more robust and generalizable models. The larger the data set the better, as it will give much information to the machine learning models to find a robust prediction ability. Among the machine learning models in the study, we see that the RF and the XGBoost have shown better results in predicting capability in the dataset (training, validating, and testing) and the error evaluation metrics. By comparing the models in both base and models that use hyperparameters, the RF gave a strong performance evaluation which suited the problem very well. However, XGBoost is also worth considering as its prediction ability was slightly less than RF allowing for more complex modeling.

Models like the LR despite not performing well like the RF and XGBoost, are good for establishing a baseline, as in this study LR had a moderate predictive capability. In other forms of modeling SVR is effective for smaller datasets, Neural networks (Artificial Neural network (ANN)) for deep learning, and Time series models (Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARMIA)) for temporal data can be effective.

CHAPTER SIX

CONCLUSION AND FUTURE WORKS

6.1 Conclusions

The main importance of water consumption prediction is that it helps the municipality plan for future water consumption. This represents better utilization in the administration of water resources. For developed and developing countries attaining a good foresight of what to expect protects them from what's to come, to be sustainable and ready for the challenges. Even though there are other factors concerning water availability (financial and others) it is a major step forward. The result of this study is believed to benefit the initiation of machine learning in developing urban cities and be an input for the municipality of Hawassa in their water management efforts.

Human beings cannot live without water and having clean consumable water readily available 24/7 should be one of the priorities of the HCWSSSE. The design of this study followed the experimental design. Hence, this study aimed to build a model that predicts the water consumption for the HCWSSSE database. From the models tested in the research, RF and the XGBoost models demonstrated better performance compared to the other models tested in the study. The ensemble nature of Random Forest allows it to manage overfitting while providing robust predictions, whereas XGBoost's gradient-boosting framework excels in accuracy and efficiency, which can be seen.

As the demand for reliable water management solutions grows, the continued adoption of RF and XGBoost in water consumption prediction would be instrumental in addressing the challenges posed by climate change and population growth, paving the way for more resilient and efficient water systems. The success of machine learning models in predicting water consumption, especially in urban cities, does not only rely on the prediction accuracy but its adaptability to changing conditions over time. It is a multifaced process that requires careful consideration and key steps that help the models capture the complexities of water usage behavior. Urban cities face many challenges related to water scarcity and management, leveraging the use of machine learning would increase the chance of better decision-making and promote sustainable water practices.

6.2 Future Work and Recommendations

While experimenting in the study some future works were identified and noted. One of them is that the data that was collected was every month of water consumption and it would give a better insight if the data were collected from a daily record of water consumption. Areas with no water (lakes, rivers, and the like) can be explored and water consumption can be evaluated using machine learning. The models are also recommended for training on a large amount of data with complex configurations. Exploring deep learning approaches for improved accuracy in water consumption predictions, incorporating additional data sources like social or economic indicators.

While population size is an important factor, incorporating socioeconomic status (SES) provides a better understanding of water utilization. SES reflects not only the number of people but also the underlying economic conditions, infrastructure availability, and individual consumption behaviors, offering a clearer picture of water demand at both the household and kebele levels. Therefore, using SES alongside population size, rather than population size alone, is likely to give a more accurate prediction of water utilization.

The incorporation of non-revenue water (NRW) in water consumption prediction. This would help us understand the reduction in water loss due to leaks, theft, or meter inaccuracies. Including NRW is a critical concept in water utility facilities and for improving efficiency and sustainability in water distribution. These models could combine advanced techniques like time series analysis, machine learning, and deep learning to predict consumption patterns and estimate NRW in real time, using data from smart meters, IoT sensors, and flow monitoring devices. The goal of NRW is to predict or understand both the actual water usage (billed water) and the amount of unaccounted water that is lost in the system. Modeling could be done using these techniques combining water demand prediction with the estimation of NRW, which allows utility facilities to optimize their strategy for reduction loss and ensure a much more sustainable and reliable water use. Accurate prediction models can help utilities manage resources, reduce waste, and improve the accuracy of billing. This could ultimately lead to proactive maintenance, better operational strategies, and more efficient water distribution systems.

Water consumption prediction and forecasting is a key factor, therefore having a better handle on the information and examining the natural cause and human behavior may introduce other research areas in universities, newly developed urban cities, and rural areas. Future studies should consider

applying these models to datasets from diverse regions to validate their generalizability. Altitude and other natural factors could be tested for different models. The city municipality could also benefit from such prediction models as resource management is their main task and other similar tasks, using Machine learning Algorithms to reduce the wastage of water being pumped to the community would economically benefit the city administration.

In Ethiopia, the forecasting of electric city power consumption from the grid can be assessed, and predictions can be researched, as the data collection is monthly. Incorporating real-time data using an IOT sensor recording water consumption can enhance accuracy and responsiveness. Models can adapt to new data in real-time, improving predictions as patterns change. This will address the data gaps to be handled, and the user behavior analysis could be determined. Analyzing the trends in water consumption patterns would give informed decision-making for better practices.

Finally, further research is needed to determine how long trained models in RF, LR, SVR, and XGBoost systems remain valid and effective as many aspects of prediction will vary. The system might miss essential variables and outcomes might be missed and the system needs to be retrained. There are many other Machine learning models and techniques that can be explored. By leveraging water consumption predictions, organizations can make informed decisions that promote sustainability, efficiency, and resilience in water resource management.

REFERENCE

[1] M. D. Elaine K. Luo, “How long can you live without water? Facts and effects,” Medical News Today. Accessed: May 18, 2023. [Online]. Available: https://www.medicalnewstoday.com/articles/325174

[2] C. Kühnert, N. M. Gonuguntla, H. Krieg, D. Nowak, and J. A. Thomas, “Application of LSTM networks for water demand prediction in optimal pump control,” *Water (Switzerland)*, vol. 13, no. 5, pp. 1–19, 2021, doi: 10.3390/w13050644.

[3] H. Heidari, M. Arabi, T. Warziniack, and S. Sharvelle, “Effects of Urban Development Patterns on Municipal Water Shortage,” *Frontiers in Water*, vol. 3, p. 77, Jul. 2021, doi: 10.3389/FRWA.2021.694817/BIBTEX.

[4] G. De Souza Groppo, M. A. Costa, and M. Libânio, “Predicting water demand: A review of the methods employed and future possibilities,” *Water Sci Technol Water Supply*, vol. 19, no. 8, pp. 2179–2198, 2019, doi: 10.2166/ws.2019.122.

[5] M. Hendrix, “Water in Ethiopia: Drought, Disease and Death,” *Global Majority E-Journal*, vol. 3, no. 2, pp. 110–120, 2012.

[6] Y. Shan, L. Yang, K. Perren, and Y. Zhang, “Household water consumption: Insight from a survey in Greece and Poland,” *Procedia Eng*, vol. 119, no. 1, pp. 1409–1418, 2015, doi: 10.1016/j.proeng.2015.08.1001.

[7] “Ethiopia population (2023) live — Countrymeters,” Country Meters. Accessed: May 22, 2023. [Online]. Available: https://countrymeters.info/en/Ethiopia

[8] A. S. S. S. R. A. Dr. Sunil Shroff, “Daily Water Intake Calculator.” Accessed: Nov. 29, 2021. [Online]. Available: https://www.medindia.net/patients/calculators/daily-water-requirement.asp

[9] T. M. (Tom M. Mitchell, *Machine Learning*. 1997.

[10] D. Lee and S. Derrible, “Predicting Residential Water Demand with Machine-Based Statistical Learning,” *J Water Resour Plan Manag*, vol. 146, no. 1, p. 04019067, 2020, doi: 10.1061/(asce)wr.1943-5452.0001119.

[11] Z. Zarrin, O. Hamidi, P. Amini, and Z. Maryanaji, “Predicting the pulse of urban water demand: a machine learning approach to deciphering meteorological influences,” *BMC Res Notes*, vol. 17, no. 1, Dec. 2024, doi: 10.1186/s13104-024-06878-6.

[12] M. A. S. Campos *et al.*, “Impact of the COVID-19 pandemic on water consumption behaviour,” *Water Supply*, vol. 00, no. 0, pp. 1–10, 2021, doi: 10.2166/ws.2021.160.

[13] M. Mengistu, “SOCIAL SCIENCES AND HUMANITIES A HISTORY OF HAWASSA TOWN UPTO 1991,” Hawassa, 2017.

[14] Hawassa City Administration, “ENVIRONMENTAL PROTECTION and FOREST DEVELOPMENT AROUND HAWASSA CITY AND THE LAKE,” Hawassa City , 2012.

[15] A. W. Worako, “International Journal of Water Resources and Environmental Engineering Evaluation of the water quality status of Lake Hawassa by using water quality index, Southern Ethiopia,” vol. 7, no. 4, pp. 58–65, 2015, doi: 10.5897/IJWREE2014.

[16] X. Fang, J. Liu, M. Zhou, H. Zhang, and J. Zhao, “Review of the Mechanism and Methodology of Water Demand Forecasting in the Socio-Economic System,” Jun. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/w16111631.

[17] Z. Y. Wu, M. El-Maghraby, and S. Pathak, “Applications of deep learning for smart water networks,” *Procedia Eng*, vol. 119, no. 1, pp. 479–485, 2015, doi: 10.1016/j.proeng.2015.08.870.

[18] R. Q. Grafton, M. B. Ward, H. To, and T. Kompas, “Determinants of residential water consumption: Evidence and analysis from a 10-country household survey,” *Water Resour Res*, vol. 47, no. 8, 2011, doi: 10.1029/2010WR009685.

[19] S. Nair and S. Nair, “Challenges in urban water management in a changing environment case study from a growing tropical city.” [Online]. Available: https://hal.science/hal-03296160v1

[20] F. Ghobadi and D. Kang, “Application of Machine Learning in Water Resources Management: A Systematic Literature Review,” Feb. 01, 2023, *MDPI*. doi: 10.3390/w15040620.

[21] M. Firat, M. E. Turan, and M. A. Yurdusev, “Comparative analysis of neural network techniques for predicting water consumption time series,” *J Hydrol (Amst)*, vol. 384, no. 1–2, pp. 46–51, Apr. 2010, doi: 10.1016/J.JHYDROL.2010.01.005.

[22] Z. Wang, X. Wu, H. Wang, and T. Wu, “Prediction and analysis of domestic water consumption based on optimized grey and Markov model,” *Water Supply*, pp. 3887–3899, 2021, doi: 10.2166/ws.2021.146.

[23] D. Walker, E. Creaco, L. Vamvakeridou-Lyroudia, R. Farmani, Z. Kapelan, and D. Savić, “Forecasting domestic water consumption from smart meter readings using statistical methods and artificial neural networks,” *Procedia Eng*, vol. 119, no. 1, pp. 1419–1428, 2015, doi: 10.1016/j.proeng.2015.08.1002.

[24] D. S. Vijayan, H. T. Tadesse, Y. Yokamo, R. Divahar, T. Bezabih Bashe, and J. Jebasingh Daniel, “A Brief Data on Water Demand Assessment for Sustainable Potable Water Supply in Yergalem Tula Kebele, Ethiopia,” *J Environ Public Health*, vol. 2022, 2022, doi: 10.1155/2022/1606590.

[25] L. Saletti-cuesta *et al.*, “Municipal Water Demand Forecasting in the Short and Long Term with ANN and SD Models,” *Sustainability (Switzerland)*, vol. 4, no. 1, pp. 1–9, 2020, [Online]. Available: https://pesquisa.bvsalud.org/portal/resource/en/mdl-20203177951%0Ahttp://dx.doi.org/10.1038/s41562-020-0887-9%0Ahttp://dx.doi.org/10.1038/s41562-020-0884-z%0Ahttps://doi.org/10.1080/13669877.2020.1758193%0Ahttp://sersc.org/journals/index.php/IJAST/article

[26] S. M. Lencha, J. Tränckner, and M. Dananto, “Assessing the water quality of lake hawassa Ethiopia—Trophic state and suitability for anthropogenic uses—applying common water quality indices,” *Int J Environ Res Public Health*, vol. 18, no. 17, 2021, doi: 10.3390/ijerph18178904.

[27] S. M. Lencha, M. D. Ulsido, and A. Muluneh, “Evaluation of seasonal and spatial variations in water quality and identification of potential sources of pollution using multivariate statistical techniques for Lake Hawassa Watershed, Ethiopia,” *Applied Sciences (Switzerland)*, vol. 11, no. 19, 2021, doi: 10.3390/app11198991.

[28] S. Raschka, “What are Machine Learning and Deep Learning? An Overview,” 2019. [Online]. Available: http://stat.wisc.edu/~sraschka/teaching/stat479-ss2019/

[29] T. Jiang, J. L. Gradus, and A. J. Rosellini, “Supervised Machine Learning: A Brief Primer,” *Behav Ther*, vol. 51, no. 5, pp. 675–687, Sep. 2020, doi: 10.1016/j.beth.2020.05.002.

[30] Leo Breiman, “RANDOM FORESTS,” Jan. 2001.

[31] “Algorithm 15.1 Random Forest for Regression or Classification.” [Online]. Available: http://www.math.usu.edu/

[32] A. Kulaczkowski and J. Lee, “Harnessing the Power of Random Forest for Precise Short-Term Water Demand Forecasting in Italian Water Districts,” MDPI AG, Sep. 2024, p. 81. doi: 10.3390/engproc2024069081.

[33] Derrick Mwiti, “Random Forest Regression: When Does It Fail and Why?,” neptune.ai. Accessed: May 22, 2023. [Online]. Available: https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why

[34] Rebecca Bevans, “Simple Linear Regression | An Easy Introduction & Examples,” Sctibbr. Accessed: May 22, 2023. [Online]. Available: https://www.scribbr.com/statistics/simple-linear-regression/

[35] “Chapter 2 Simple Linear Regression Analysis The simple linear regression model.”

[36] K. P. Singh, N. Basant, and S. Gupta, “Support vector machines in water quality management,” *Anal Chim Acta*, vol. 703, no. 2, pp. 152–162, Oct. 2011, doi: 10.1016/j.aca.2011.07.027.

[37] A. Candelieri *et al.*, “Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization,” *Comput Oper Res*, vol. 106, pp. 202–209, Jun. 2019, doi: 10.1016/j.cor.2018.01.013.

[38] S. Shan, H. Ni, G. Chen, X. Lin, and J. Li, “A Machine Learning Framework for Enhancing Short-Term Water Demand Forecasting Using Attention-BiLSTM Networks

Integrated with XGBoost Residual Correction," *Water (Switzerland)*, vol. 15, no. 20, Oct. 2023, doi: 10.3390/w15203605.

[39] B. D. Kitessa, S. M. Ayalew, G. S. Gebrie, and S. T. Teferi, "Long-term water-energy demand prediction using a regression model: A case study of addis ababa city," *Journal of Water and Climate Change*, vol. 12, no. 6, pp. 2555–2578, Sep. 2021, doi: 10.2166/wcc.2021.012.

[40] A. Candelieri, D. Soldi, and F. Archetti, "Short-term forecasting of hourly water consumption by using automatic metering readers data," *Procedia Eng*, vol. 119, no. 1, pp. 844–853, 2015, doi: 10.1016/j.proeng.2015.08.948.

[41] Z. Li, C. Wang, Y. Liu, and J. Wang, "Enhancing the explanation of household water consumption through the water-energy nexus concept," *NPJ Clean Water*, vol. 7, no. 1, Dec. 2024, doi: 10.1038/s41545-024-00298-6.

[42] A. Mackey and S. M. Gass, "Research Methods in Second Language Acquisition: A Practical Guide," 2012. [Online]. Available: www.cyandesign.co.uk

[43] Getu Degu Tegbar Yigzaw, "Research Methodology for Health Science," 2006.

[44] J. Kamiri and G. Mariga, "Research Methods in Machine Learning: A Content Analysis," *International Journal of Computer and Information Technology(2279-0764)*, vol. 10, no. 2, Mar. 2021, doi: 10.24203/ijcit.v10i2.79.

[45] S. Muhammad and S. Kabir, "METHODS OF DATA COLLECTION," 2016. [Online]. Available: https://www.researchgate.net/publication/325846997

[46] Z. S. Abdallah, L. Du, and G. I. Webb, "Data Preparation," in *Encyclopedia of Machine Learning and Data Mining*, Boston, MA: Springer US, 2017, pp. 318–327. doi: 10.1007/978-1-4899-7687-1_62.

[47] D. Kanellopoulos and P. E. Pintelas, "Data Preprocessing for Supervised Learning," 2006. [Online]. Available: https://www.researchgate.net/publication/228084519

[48] B. Mayabi and T. Wamalwa, "An Artificial Neural Network Model for Predicting Retail Maize Prices In Kenya," 2019.

[49] Y. ; C. X. ; W. H. Zhang, “Urban Water Demand Prediction Using Random Forest: A Case Study,” *Water Resources Management*, vol. 35, no. 4, pp. 1235–1249, 2021.

[50] A. C. S. Eeckels. (ACAPS) Van den Broeck J, “DATA CLEANING Dealing with messy data,” 2016.

[51] S. Wang, J. Tang, and H. Liu, “Feature Selection,” in *Encyclopedia of Machine Learning and Data Mining*, Boston, MA: Springer US, 2016, pp. 1–9. doi: 10.1007/978-1-4899-7502-7_101-1.

[52] S Jez, “Correlation,” 2018. [Online]. Available: http://www.tufts.edu/~gdallal/out.htm.

[53] Fazal Rehman Shamil, “Min Max normalization,” 2022.

APPENDIX

1. Water Tariff

Hawassa Town Water Supply & Sewage Service Enterprise					
New Water Tariff					
Connection Type					
Band	1<5m^3	6-10m^3	11-15m3	16-20m^3	>20m^3
Domestic	12	22.248	25.812	28.32	30.72
Public & Gov. Customers	25.81	28.212	31.104	34.704	39.504
Commercial & Industrial Customers	27.18	29.58	33.42	37.50	43.50
Standpipes (Bono)	9.6	9.6	9.6	9.6	9.6

Table 1 Hawassa town water supply & sewage service enterprise (Water tariff 2014) Source: SRSHC-WSSSE 2014 E.C

2. Codes used for Modeling on Jupyter NoteBook

```
#importing the neccesary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
#Pandas settings
pd.set_option('display.max_rows', 100)
pd.set_option('display.max_columns', 30)
#Importing the dataframe
df = pd.read_csv('Customer_Kebele_Cons2010_2015DBConsFinal.CSV')
df.sample(10)
#checking the unique values of the categorical columns
columns_to_explore = ['CustomerType', 'Kebele']
for col in columns_to_explore:
    display(f'For the column [{col}]; {df[col].dtype}, the unique values are:',
df[col].unique())
print(f'''
The unique count for the Kebele and customerType are:
For the "customerType":   {len(pd.unique(df['CustomerType']))}
For the "Kebele"      :   {len(pd.unique(df['Kebele']))}''')
```

```
#Drop any row column with fully missing values
print (df.shape)
df.dropna(axis=1, how='all', inplace=True)
df.dropna(axis=0, how='all', inplace=True)
#check for shape changes
print (df.shape)
# To impute this time series, we have to create a CurrentDate column, assuming the day is 20
df['CurrentDate'] = pd.to_datetime(dict(year=df.Year, month=df.Month, day=20))
# setting the index to the date column
df = df.set_index('CurrentDate')
df.head(5)
#labeling the Categorical values
label = df.select_dtypes(exclude='number').columns
for encode in label:
  df[encode] = df[encode].astype('category')
  df[encode] = df[encode].cat.codes
#Full data set correleation matrics
cor = df.corr()
cor
#Visualizing the Correlation metrics on the heatmap
plt.figure(figsize=(10,9z))
sns.heatmap(cor, annot=True, cmap='coolwarm')
#droping the unnecessary values to visualize
df = df.drop(['ConsAnnualReference','CustAnnualReference', 'TheMonth'],axis=1)
#set the width and heigth of the figure
plt.figure(figsize = (16,6))
#Adding a title to the chart
plt.title("Waterdata consumption diagram from the years of 2010 - 2015")
#Line chart showing how the water_data consumption took over time
sns.lineplot(data = df)
#reseting the index from date time and removing the unnecessary columns
df = df.reset_index()
df = df.drop(['CurrentDate'],axis=1)
#Min max scaling for better experimentation
from sklearn.preprocessing import MinMaxScaler
```

```
minmax_scale = MinMaxScaler()

# coll = ["Numb_Cust","MonthConsumption","Year"]

#coll = ["Numb_Cust", "Kebele", "Year"]

coll = ["Numb_Cust"]

df[coll] = minmax_scale.fit_transform(df[coll])

df.head()
```

#Z-score method

 # find the limits for the consumption

```
upper_limit = df['MonthConsumption'].mean() + 3*df['MonthConsumption'].std()

lower_limit = df['MonthConsumption'].mean() - 3*df['MonthConsumption'].std()

print('upper limit:', upper_limit)

print('lower limit:', lower_limit)
```

 # trimming - delete the outlier data

```
new_df = df.loc[(df['MonthConsumption'] <= upper_limit) & (df['MonthConsumption'] >=
lower_limit)]

print('before removing outliers:', len(df))

print('after removing outliers:',len(new_df))

print('outliers:', len(df)-len(new_df))
```

 # find the outliers

```
df.shape

df.loc[(df['MonthConsumption'] > upper_limit) | (df['MonthConsumption'] <
lower_limit)]

new_df["MonthConsumption"].plot(kind="box", figsize=(7,7))
```

 # trimming - delete the outlier data

```
new_df = df.loc[(df['MonthConsumption'] <= upper_limit) & (df['MonthConsumption'] >=
lower_limit)]

print('before removing outliers:', len(df))

print('after removing outliers:',len(new_df))

print('outliers:', len(df)-len(new_df))
```

#Model Building

 # importing the libraries and Algorithims

```
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import GridSearchCV,RandomizedSearchCV

from sklearn.metrics import mean_absolute_error

from sklearn.metrics import mean_squared_error
```

```
from sklearn.metrics import r2_score

from sklearn.metrics import mean_absolute_percentage_error

import numpy as np

# splitting the dataset to X and Y

X = new_df.drop('MonthConsumption',axis=1)

y = new_df['MonthConsumption']

# using splitting the data X and Y to Train, Val and Test dataset

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=0)

X_val,X_test,y_val,y_test =
train_test_split(X_train,y_train,test_size=0.50,random_state=0)

#Feature selection weight

from sklearn.ensemble import ExtraTreesRegressor

model = ExtraTreesRegressor(random_state=42)

model.fit(X_train, y_train)

feat_importances = pd.Series(model.feature_importances_, index=X.columns)

feat_importances.nlargest(10).plot(kind='barh', color='skyblue')

plt.title('Degree of Features Importance')

plt.xlabel('Importance Score')

plt.ylabel('Features')

# Show the plot

plt.show()

#Base Model for RF: Training the model and implementing on the validation dataset

rf = RandomForestRegressor()

rf.fit(X_train,y_train)

y_pred_test = rf.predict(X_test)

y_pred_train = rf.predict(X_train)

y_pred_val = rf.predict(X_val)

rfr_test = r2_score(y_test,y_pred_test)

rfr_train = r2_score(y_train,y_pred_train)

rfr_val = r2_score(y_val,y_pred_val)

#Evaluation Metrics

min_error = mean_squared_error(y_val, y_pred_val)

min_abs = mean_absolute_error(y_val, y_pred_val)

min_percent = mean_absolute_percentage_error(y_val,y_pred_val)

min_rmse = mean_squared_error(y_val, y_pred_val)

RMSE = np.sqrt(min_rmse)
```

```
print(f'''
The metrics of the base model RF training and Val datasets are:
Feature             Dataset       The r2
========================================
MonthConsumption    Training      {rfr_train*100:05.2f}%
                    Validation    {rfr_val*100:05.2f}%
                    Testing       {rfr_test*100:05.2f}%
========================================
For the "MSE":   {min_error}
For the "MAE":   {min_abs}
For the "MAPE":  {min_percent}
For the "RMSE":  {RMSE}''')
#plotting the base model
import matplotlib.pyplot as plt
x = ["Train", "Validation", "Test"]
y = [rfr_train, rfr_val, rfr_test]
color = ['#1f77b4', '#ff7f0e', '#2ca02c']
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x, y, color=color, edgecolor='black', linewidth=1.5)
ax.set_xlabel('Dataset', fontsize=14)
ax.set_ylabel('R-squared Score', fontsize=14)
ax.set_title('Random Forest Regressor: "RFR" R-squared Score', fontsize=16)
fig.set_facecolor('#f0f0f0')
ax.grid(color='#cccccc', linestyle='--', linewidth=1)
ax.tick_params(axis='both', which='major', labelsize=12)
for i, v in enumerate(y):
    ax.text(i, v + 0.01, f"{v:.2f}", ha='center', va='bottom', fontsize=12)
plt.show()
# Parameter Tuning using RandomizedSearchCV for Base model Random Forest
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [5, 10, 15, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
rf = RandomForestRegressor(random_state=42)
```

```
randomized_search = RandomizedSearchCV(rf, param_grid, n_iter=10,
                                       cv=5, scoring='neg_mean_squared_error',
                                       random_state=42)
randomized_search.fit(X_train, y_train)
#print out the best parmeters from the Randomized search
best_params = randomized_search.best_params_
print(best_params)
# Model for RF on the validation dataset with parameter
rf = RandomForestRegressor(n_estimators = best_params['n_estimators'],
                           min_samples_leaf = best_params['min_samples_leaf'],
                           min_samples_split = best_params['min_samples_split'],
                           max_depth = best_params['max_depth'])
rf.fit(X_train,y_train)
y_pred_test = rf.predict(X_test)
y_pred_train = rf.predict(X_train)
y_pred_val = rf.predict(X_val)
rfr_test = r2_score(y_test,y_pred_test)
rfr_train = r2_score(y_train,y_pred_train)
rfr_val = r2_score(y_val,y_pred_val)
#Evaluation Metrics
min_error = mean_squared_error(y_val, y_pred_val)
min_abs = mean_absolute_error(y_val, y_pred_val)
min_percent = mean_absolute_percentage_error(y_val,y_pred_val)
min_rmse = mean_squared_error(y_val, y_pred_val)
RMSE = np.sqrt(min_rmse)
print(f'''
The metrics of the model RF using the hyperparameters:
Feature             Dataset       The r2
=========================================
MonthConsumption    Training      {rfr_train*100:05.2f}%
                    Validation    {rfr_val*100:05.2f}%
                    Testing       {rfr_test*100:05.2f}%
=========================================
For the "MSE":   {min_error}
For the "MAE":   {min_abs}
For the "MAPE":  {min_percent}
```

```
For the "RMSE":  {RMSE}
''')
import matplotlib.pyplot as plt
x = ["Train", "Validation", "Test"]
y = [rfr_train, rfr_val, rfr_test]
color = ['#1f77b4', '#ff7f0e', '#2ca02c']
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x, y, color=color, edgecolor='black', linewidth=1.5)
ax.set_xlabel('Dataset', fontsize=14)
ax.set_ylabel('R-squared Score', fontsize=14)
ax.set_title('Best parameter: "RFR" R-squared Score', fontsize=16)
fig.set_facecolor('#f0f0f0')
ax.grid(color='#cccccc', linestyle='--', linewidth=1)
ax.tick_params(axis='both', which='major', labelsize=12)
for i, v in enumerate(y):
    ax.text(i, v + 0.01, f"{v:.2f}", ha='center', va='bottom', fontsize=12)
plt.show()
# Linear Regression Model
from sklearn.linear_model import Ridge
#calling the Algorithim
lr = LinearRegression()
#fitting the model
lr.fit(X_train,y_train)
y_pred_test = lr.predict(X_test)
y_pred_train = lr.predict(X_train)
y_pred_val = lr.predict(X_val)
lr_test = r2_score(y_test,y_pred_test)
lr_train = r2_score(y_train,y_pred_train)
lr_val = r2_score(y_val, y_pred_val)
#Evaluation Metrics
min_error = mean_squared_error(y_val, y_pred_val)
min_abs = mean_absolute_error(y_val, y_pred_val)
min_percent = mean_absolute_percentage_error(y_val,y_pred_val)
min_rmse = mean_squared_error(y_val, y_pred_val)
RMSE = np.sqrt(min_rmse)
print(f'''
```

```
The metrics of the LR Model:
Feature             Dataset       The r2
========================================
MonthConsumption    Training      {lr_train*100:05.2f}%
                    Validation    {lr_val*100:05.2f}%
                    Testing       {lr_test*100:05.2f}%
========================================
For the "MSE":   {min_error}
For the "MAE":   {min_abs}
For the "MAPE":  {min_percent}
For the "RMSE":  {RMSE}''')
import matplotlib.pyplot as plt
x = ["Train", "Validation", "Test"]
y = [lr_train, lr_val, lr_test]
color = ['#1f77b4', '#ff7f0e', '#2ca02c']
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x, y, color=color, edgecolor='black', linewidth=1.5)
ax.set_xlabel('Dataset', fontsize=14)
ax.set_ylabel('R-squared Score', fontsize=14)
ax.set_title('Linear Regression: "LR" R-squared Score', fontsize=16)
fig.set_facecolor('#f0f0f0')
ax.grid(color='#cccccc', linestyle='--', linewidth=1)
ax.tick_params(axis='both', which='major', labelsize=12)
for i, v in enumerate(y):
    ax.text(i, v + 0.01, f"{v:.2f}", ha='center', va='bottom', fontsize=12)
plt.show()
# Parameter Tuning using RandomizedSearchCV
#Linear Regression
from scipy.stats import uniform
from sklearn.linear_model import Ridge
param_distributions = {
    'alpha': uniform(0.001, 99.999),
    'fit_intercept': [True, False],
    'copy_X': [True, False],
    'max_iter': [100, 500, 1000,10000],
    'solver': ['auto', 'svd', 'cholesky', 'lsqr', 'sag']}
```

```
ridge = Ridge()

random_search = RandomizedSearchCV(ridge, param_distributions, n_iter=10, cv=5,
scoring='neg_mean_squared_error', random_state=42)

random_search.fit(X_train, y_train)

#print out the best parmeters from the Randomized search

best_params = random_search.best_params_

print(best_params)

lr = Ridge(

    alpha=best_params['alpha'],

    fit_intercept=best_params['fit_intercept'],

    copy_X=best_params['copy_X'],

    max_iter=best_params['max_iter'],

    solver=best_params['solver'])

lr.fit(X_train,y_train)

y_pred_test = lr.predict(X_test)

y_pred_val = lr.predict(X_val)

y_pred_train = lr.predict(X_train)

lr_test = r2_score(y_test,y_pred_test)

lr_val = r2_score(y_val,y_pred_val)

lr_train = r2_score(y_train,y_pred_train

#Evaluation Metrics

min_error = mean_squared_error(y_val, y_pred_val)

min_abs = mean_absolute_error(y_val, y_pred_val)

min_percent = mean_absolute_percentage_error(y_val,y_pred_val)

min_rmse = mean_squared_error(y_val, y_pred_val)

RMSE = np.sqrt(min_rmse)

print(f'''

The metrics of the LR model using the hyperparameters:

Feature             Dataset       The r2

========================================

MonthConsumption    Training      {lr_train*100:05.2f}%

                    Validation    {lr_val*100:05.2f}%

                    Testing       {lr_test*100:05.2f}%

========================================

For the "MSE":   {min_error}

For the "MAE":   {min_abs}
```

```
For the "MAPE":  {min_percent}
For the "RMSE":  {RMSE}''')
import matplotlib.pyplot as plt
x = ["Train", "Validation", "Test"]
y = [lr_train, lr_val, lr_test]
color = ['#1f77b4', '#ff7f0e', '#2ca02c']
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x, y, color=color, edgecolor='black', linewidth=1.5)
ax.set_xlabel('Dataset', fontsize=14)
ax.set_ylabel('R-squared Score', fontsize=14)
ax.set_title('Linear Regression: "LR" R-squared Score', fontsize=16)
fig.set_facecolor('#f0f0f0')
ax.grid(color='#cccccc', linestyle='--', linewidth=1)
ax.tick_params(axis='both', which='major', labelsize=12)
for i, v in enumerate(y):
    ax.text(i, v + 0.01, f"{v:.2f}", ha='center', va='bottom', fontsize=12)
plt.show()
# SVR Regressor
from sklearn.svm import SVR
from sklearn.model_selection import RandomizedSearchCV
# Scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test
# Expand the hyperparameter grid
param_grid = {
    "kernel": ["linear", "rbf", "poly", "sigmoid"],
    "C": [0.01, 0.1, 1, 10, 100],
    "epsilon": [0.01, 0.1, 1],
    "gamma": ["scale", "auto"]}
sv = SVR()
# RandomizedSearchCV with expanded grid
svr_search = RandomizedSearchCV(sv, param_grid, cv=5,
scoring='neg_mean_squared_error', n_jobs=-1)
svr_search.fit(X_val_scaled, y_val)
```

```
# Get the best parameters and fit the model
svr_best_params = svr_search.best_params_
sv.set_params(**svr_best_params)
sv.fit(X_train_scaled, y_train)
# Predictions
y_pred_test = sv.predict(X_test_scaled)
y_pred_val = sv.predict(X_val_scaled)
y_pred_train = sv.predict(X_train_scaled)
# Evaluate the model on the test, validation, and training sets
svr_test = sv.score(X_test_scaled, y_test)
svr_val = sv.score(X_val_scaled, y_val)
svr_train = sv.score(X_train_scaled, y_train)
# Evaluate the model on the test, validation, and training sets
print("R-squared (test):", svr_test)
print("R-squared (validation):", svr_val)
print("R-squared (training):", svr_train)
# Evaluate other metrics
sv_r2 = sv.score(X_test_scaled, y_test)
svr_mae = np.mean(abs(y_test - y_pred_test))
svr_mse = np.mean((y_test - y_pred_test) * 2)
svr_rmse = np.sqrt(svm_mse)
#best_params = svr_best_params.best_params_
print(svr_best_params)
print(f'''
The metrics of the SVR model using the hyperparameters:
Feature             Dataset       The r2
========================================
MonthConsumption    Training      {svr_train*100:05.2f}%
                    Validation    {svr_val*100:05.2f}%
                    Testing       {svr_test*100:05.2f}%
========================================
For the "MSE":   {svr_mse}
For the "MAE":   {svr_mae}
For the "MAPE":  {svr_min_percent}
For the "RMSE":  {svr_rmse}''')
X = ['SVR Train','SVR Validation','SVR Test']
```

```
y = [svr_train, svr_val, svr_test]
colors = ['#1f77b4', '#ff7f0e', '#2ca02c']
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(X, y, color=colors, edgecolor='black', linewidth=1.5)
#labels
ax.set_xlabel('Model Phase', fontsize=14)
ax.set_ylabel('R-squared: ', fontsize=14)
ax.set_title('SVR Model Performance', fontsize=16)
# plots
ax.set_facecolor('#f0f0f0')
ax.grid(color='#cccccc', linestyle='--', linewidth=1)
ax.tick_params(axis='both', which='major', labelsize=12)
# add label on top of graph
for i, v in enumerate(y):
    ax.text(i, v + 0.01, f"{v:.2f}", ha='center', va='bottom', fontsize=12)
plt.show();
```

 # XGBoost

```
import xgboost as xgb
from scipy.stats import randint, uniform
pip install xgboost
param_dist = {
    'max_depth': randint(2, 10),
    'n_estimators': randint(50, 200),
    'learning_rate': uniform(0.01, 0.3),
    'min_child_weight': randint(1, 6),
    'gamma': uniform(0, 1),
    'subsample': uniform(0.5, 0.5),
    'colsample_bytree': uniform(0.5, 0.5)}
xgb_model = xgb.XGBRegressor()
xgb_model.fit(X_train, y_train)
random_search = RandomizedSearchCV(xgb_model, param_distributions=param_dist,
n_iter=10, cv=5, scoring='neg_mean_squared_error', random_state=42)
random_search.fit(X_train, y_train)
random_best = random_search.best_params_
random_best
xgb_model_last = xgb.XGBRegressor(
```

```
    max_depth=random_best['max_depth'],
    n_estimators=random_best['n_estimators'],
    learning_rate=random_best['learning_rate'],
    min_child_weight=random_best['min_child_weight'],
    gamma=random_best['gamma'],
    subsample=random_best['subsample'],
    colsample_bytree=random_best['colsample_bytree'])
xgb_model_last.fit(X_train, y_train)
xgb_model_last = xgb.XGBRegressor(
    max_depth=random_best['max_depth'],
    n_estimators=random_best['n_estimators'],
    learning_rate=random_best['learning_rate'],
    min_child_weight=random_best['min_child_weight'],
    gamma=random_best['gamma'],
    subsample=random_best['subsample'],
    colsample_bytree=random_best['colsample_bytree'])
xgb_model_last.fit(X_train, y_train
y_pred_test = xgb_model_last.predict(X_test)
y_pred_val = xgb_model_last.predict(X_val)
y_pred_train = xgb_model_last.predict(X_train)
# Evaluate the model on the test, validation, and training sets
xgb_test = xgb_model_last.score(X_test, y_test)
xgb_val = xgb_model_last.score(X_val, y_val)
xgb_train = xgb_model_last.score(X_train, y_train)
print("R-squared (test):", xgb_test)
print("R-squared (validation):", xgb_val)
print("R-squared (training):", xgb_train)
# Evaluate other metrics
xgb_r2 = xgb_model_last.score(X_test, y_test)
xgb_mse = mean_squared_error(y_test, y_pred_test)
xgb_mae = mean_absolute_error(y_test, y_pred_test)
#xgb_mae = np.mean(abs(y_test - y_pred_test))
#xgb_mse = np.mean((y_pred_test - y_test) * 2)
xgb_rmse = np.sqrt(xgb_mse)
xgb_min_percent = mean_absolute_percentage_error(y_val,y_pred_val)
print("R-squared:", xgb_r2)
```

```
print("MAE:", xgb_mae)
print("MSE:", xgb_mse)
print("MAPE:", xgb_min_percent)
print("RMSE:", xgb_rmse)
pred_data = {"Actual" : y_test,"Predicted" : y_pred_test}
df_result = pd.DataFrame(pred_data)
df_result.sample(5)
print(f'''
The metrics of the XGB model using the hyperparaemters:
Feature             Dataset       The r2
========================================
MonthConsumption    Training      {xgb_train*100:05.2f}%
                    Validation    {xgb_val*100:05.2f}%
                    Testing       {xgb_test*100:05.2f}%
========================================
For the "MSE":   {xgb_mse}
For the "MAE":   {xgb_mae}
For the "MAPE":  {xgb_min_percent}
For the "RMSE":  {xgb_rmse}''')
X = ['SVM Train','SVM Validation','SVM Test']
y = [xgb_train, xgb_val, xgb_test]
colors = ['#1f77b4', '#ff7f0e', '#2ca02c'] fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(X, y, color=colors, edgecolor='black', linewidth=1.5)
#labels
ax.set_xlabel('Model Phase', fontsize=14) ax.set_ylabel('R-squared: ', fontsize=14)
ax.set_title('XGBoost Model Performance', fontsize=16)
# plots
ax.set_facecolor('#f0f0f0') ax.grid(color='#cccccc', linestyle='--', linewidth=1)
ax.tick_params(axis='both', which='major', labelsize=12)
# add label on top of graph
for i, v in enumerate(y):
    ax.text(i, v + 0.01, f"{v:.2f}", ha='center', va='bottom', fontsize=12)
plt.show();
pred_data = {"Actual" : y_test,"Predicted" : y_pred_test}
df_result = pd.DataFrame(pred_data)
df_result.head()
```

HAWASSA UNIVERSITY