

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF COMMERCE

DEPARTMENT OF BUSINESS ADMINISTRATION AND INFORMATION SYSTEMS

PREDICTIVE ANALYTICS FRAMEWORK FOR DECISION SUPPORT IN ETHIOPIA'S FOREIGN EXCHANGE MARKET

BY HUNDERA BELEMA

ADDIS ABABA UNIVERSITY

ADDIS ABABA, ETHIOPIA

JUNE, 2026

PREDICTIVE ANALYTICS FRAMEWORK FOR DECISION SUPPORT IN ETHIOPIA'S FOREIGN EXCHANGE MARKET

HUNDERA BELEMA

A THESIS SUBMITTED TO DEPARTMENT OF BUSINESS ADMINISTRATION AND INFORMATION SYSTEMS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF BUSINESS INFORMATION SYSTEMS

ADVISOR: KURIBACHEW GIZAW (PhD)

ADDIS ABABA UNIVERSITY

SCHOOL OF COMMERCE

COLLEGE OF BUSINESS AND ECONOMICS

Hundera Belema

PREDICTIVE ANALYTICS FRAMEWORK FOR DECISION SUPPORT IN ETHIOPIA'S FOREIGN EXCHANGE MARKET

This certifies that Hundera Belema's thesis, "Predictive Analytics Framework for Decision Support in Ethiopia's Foreign Exchange Market," meets the University's standards for originality and quality. It was submitted as partial fulfillment of an MBA in Business Information Systems.

Signed by the Examining Committee

	Name	Signature	Date
Advisor	_____	_____	_____
Examiner	_____	_____	_____
Examiner	_____	_____	_____

_____

Chair of department or Graduate Program Coordinator

Declaration

I, the undersigned, declared that this research project is my own original work and has not been
submitted for any award at any other university.

Signature _____

Date _____

Hundera Belema

This research project has been submitted for examination with my confirmation as a supervisor
to the candidate.

Signature _____

Date _____

Kuribachew Gizaw (PhD)

Abstract

This research designs and tests a predictive analytics framework that can be used to support decision-making in Ethiopia's foreign exchange market. The research utilises data mining techniques along with conventional econometrics to better estimate and identify foreign exchange determinants under realistic institutional constraints. Daily USD/ETB and monthly macroeconomic indicators from August 2024 to December 2025 are used and XGBoost, ARIMA, and Prophet models are compared. Among the models XGBoost achieves the highest accuracy (MAPE 2.62%, R^2 0.6329), marginally outperforming ARIMA (2.71%), while Prophet fails. Based on SHAP analysis inflation is identified as the dominant driver of foreign exchange with 55% of feature importance followed by exports and imports. The final decision support system is built to be modular, adaptive and scalable. Therefore this study showcases the possibility of using data mining in Ethiopia's foreign exchange market improvement.

Keywords: Predictive analytics, foreign exchange, Ethiopia, XGBoost, SHAP, decision support system, USD/ETB, inflation, machine learning

Acknowledgements

I want to thank my advisor, Kuribachew Gizaw (Phd), for helping me and giving incredible advice during the entire research. Next I want to thank the school of commerce Addis Ababa University for giving me a good place to learn and the resources I needed. Lastly, thanks to my family and friends for being patient and encouraging me during the whole period.

Table of content

Declaration
Abstract
Acknowledgements
Table of content
List of Table
List of Figure
List of Abbreviations
CHAPTER ONE: INTRODUCTION **1**
1.1 Background of the Study 1
1.2 Statement of the Problem 2
1.3 Objectives of the Study 3
1.3.1 General Objective of the Study 3
1.3.2 Specific Objectives of the Study 3
1.4 Research Questions 3
1.5 Significance of the Study 4
1.6 Scope of the Study 4
1.7 Limitations of the Study 4
1.8 Organization of the Study 5
CHAPTER TWO: LITERATURE REVIEW **6**
2.1 Theoretical Exchange Rate Forecasting Context 6
2.2 Foundations of Data Mining and Predictive Analytics 7
2.2.1 Data Mining Taxonomy 7
2.2.1.1 Classification 7
2.2.1.2 Regression 8
2.2.1.3 Clustering 10
2.2.1.4 Association Rule Mining 10
2.2.1.5 Formal Positioning of the Problem 11
2.2.2 Supervised Learning Theory 12
2.2.2.1 Problem Formulation 12
2.2.2.2 Hypothesis Space 13
2.2.2.3 Risk Minimization Framework 13
2.2.2.4 Regularization and Structural Risk Minimization 14
2.2.2.5 Optimization Algorithms 14
2.2.2.6 Generalization Error 14
2.2.2.7 Validation Strategies in Time-Series Context 15
2.2.2.8 Feature Engineering and Domain Integration 15
2.2.3 Ensemble Learning Methods 15

2.2.3.1 Bagging (Bootstrap Aggregating) 16
2.2.3.2 Boosting 16
2.2.3.3 Random Forest 17
2.2.3.4 Gradient Boosting 17
2.2.3.5 Stacking (Stacked Generalization) 18
2.2.3.6 Why Ensemble Methods Outperform Single Models 18
2.2.4 Gradient Boosting and XGBoost Architecture 19
2.2.4.1 Sequential Tree Construction 20
2.2.4.2 Residual Fitting 21
2.2.4.3 Regularization in XGBoost 21
2.2.4.4 Shrinkage (Learning Rate) 22
2.2.4.5 Tree Pruning 22
2.2.5 Bias–Variance Tradeoff and Overfitting 22
2.2.5.1 Underfitting vs Overfitting 23
2.2.5.2 Regularization (L1 and L2) 23
2.2.5.3 Tree Depth Control 23
2.2.5.4 Learning Rate (Shrinkage) 24
2.3 Machine Learning for Time-Series Forecasting 24
2.3.1 Traditional Econometric Models 24
2.3.2 Machine Learning in Time-Series Forecasting 25
2.3.2.1 Autoregressive Features via Lags 25
2.3.2.2 Feature-Based Forecasting 26
2.3.2.3 Nonlinear Modeling 26
2.3.3 Feature Engineering in Financial Time-Series 26
Rolling Mean 27
2.4 Model Evaluation and Validation Framework 28
2.4.1 Forecast Error Metrics 28
2.4.2 Root Mean Squared Error (RMSE) 29
2.4.3 Mean Absolute Error (MAE) 29
2.4.4 Mean Absolute Percentage Error (MAPE) 30
2.4.5 Coefficient of Determination (R^2) 30
2.4.6 Time-Series Cross-Validation 30
2.4.7 Rolling Window Validation 31
2.4.8 Expanding Window Validation 31
2.5 EMPIRICAL LITERATURE REVIEW 32
2.5.1 Empirical Literature on Exchange Rate Determinants 32
2.5.2 Empirical Literature on Predictive Models for Exchange Rate Forecasting 33
2.6 Conceptual Framework 33
CHAPTER THREE: RESEARCH METHOD 34
3.1 Introduction 34
3.2 Research Philosophy and Research Design 34
3.3 Design Science Research (DSR) Framework 34
3.4 Data Sources and Sampling Framework 35
3.5 Variable Definition and Measurement 35

3.6 Data Preprocessing and Engineering Framework 35
3.7 Predictive Modeling Framework 36
3.7.1 ARIMA Model 36
3.7.2 Prophet Model 36
3.7.3 XGBoost Model 36
3.8 Diagnostic and Specification Testing Framework 37
3.9 Hypothesis Testing Framework 37
3.10 Model Evaluation Metrics 37
3.11 DSS Evaluation Framework 38
CHAPTER FOUR: RESULTS AND ANALYSIS 38
4.1 Introduction 38
4.2 System Implementation Environment 39
4.3 Dataset Description and Exploratory Analysis 39
4.4 Data Engineering and Processing Results 40
4.5 Diagnostic and Specification Test Results 41
4.6 Predictive Model Training Results 42
4.7 Forecasting Performance Comparison 42
4.8 SHAP Explainability Analysis 44
4.9 DSS Functional Evaluation 48
4.10 Discussion 48
CHAPTER FIVE: CONCLUSION 49
5.1 Summary of Findings 49
5.2 Answers to Research Questions 49
5.3 Theoretical Contributions 50
5.4 Practical Contributions 50
5.5 System Limitations 50
5.6 Future Research Directions 51
REFERENCE 52
Appendix 1: Predictive Modeling Implementation 55
Appendix B: SHAP Feature Importance Analysis 58

List of Table

Table No.	Title
Table 4.1	Summary Statistics of Key Variables
Table 4.2	Model Performance on Test Set
Table 4.3	Feature Importance Ranking
Table 4.2	Feature Engineering Summary
Table 4.3	Diagnostic Test Results
Table 4.4	Training Performance
Table 4.5	Model Performance on Test Set
Table 4.6	Feature Importance Ranking (SHAP Values)
Table 4.7	DSS Evaluation Summary

List of Figure

List of Figures

Figure No.	Title / Description
Figure 4.1	System Implementation Workflow
Figure 4.2	USD/ETB Trend Over Time
Figure 4.3	Actual vs Predicted - XGBoost
Figure 4.4	Model Comparison
Figure 4.5	SHAP Bar Plot
Figure 4.6	SHAP Summary Dot Plot
Figure 4.7	Inflation Dependence Plot

List of Abbreviations

Abbreviation	Full Form
ARIMA	Autoregressive Integrated Moving Average
CPI	Consumer Price Index
DSS	Decision Support System
ETB	Ethiopian Birr
FX	Foreign Exchange
IMF	International Monetary Fund
KDD	Knowledge Discovery in Databases
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
NBE	National Bank of Ethiopia
PPP	Purchasing Power Parity
R^2	Coefficient of Determination
RMSE	Root Mean Square Error
SHAP	SHapley Additive exPlanations
USD	United States Dollar
GBoost	Extreme Gradient Boosting

CHAPTER ONE: INTRODUCTION

1.1 Background of the Study

Foreign exchange rate, a conversion number tracked everyday, is a crucial element in global trade and investment. It affects trade, investment, risk management and capital flight control. Ethiopia is a developing country with foreign currency scarcity and unbalanced import and export. Economies with managed exchange rates usually have a currency that is non-convertible with liquid exchanging hard to find. These problems in Ethiopia have historically undermined financing capacity and caused economic pressure for businesses.

The shift to a market based exchange rate in July 2024 marked a turning point in Ethiopia's market. Even though the problems persisted after, a shift from a pegged exchange rate system to a market-determined rate is favourable for market participants. It helps Banks, firms and individuals be informed about their judgments about exchange rate movements. As Kirsneh (2024) documents, if uncertainty due to information gap exists, it erodes profits, discourages investment and distorts resource allocation. This means an informed decision is critical.

Ethiopia's forex market remains analytically underdeveloped with lots of misinformation. Decisions rely on basic indicators, not systematic forecasting. Key data like trade balances, remittances, and capital flows are fragmented across institutions and not used for exchange predictions. Also, conventional models from mature markets don't fit Ethiopia's context of chronic currency shortages and external vulnerability. Some methods like Grey System Theory have shown strong accuracy in data-scarce conditions, but aren't operational in policy (Goshu & Tilahun, 2016). This data-driven gap is what this study addresses.

Advancement in predictive analytics using modern framework and technologies makes foreign exchange market prediction effective. Newer technological frameworks incorporate advanced tools that guarantee better outcomes. This includes real-time exchange rate, historical transactional analysis, indicator analysis, external and internal signs. Furthermore robust predictive architecture demonstrated in multi-currency financial management, integrates time-series models with machine learning algorithms networks to forecast under volatile currency conditions with accurate indications(Gaffar et al., 2019).

In addition, neural networks and sentiment analysis have shown high predictive accuracy performance increase in exchange rates further showing the potential of advanced methods

(Ranjit et al., 2018). In Ethiopia, where foreign exchange constraints are common, using advanced methods that are capable of processing data from different information sources is very much helpful.

There are improvements in global predictive frameworks in financial markets but its usage in Ethiopia's foreign exchange market remains limited. Existing studies on Ethiopia largely focus on exchange rate determinants, macroeconomic impacts, or policy evaluation. Due to that, benefits that can be gained from predictive decision support systems are lost. By addressing this gap, this study seeks to develop a predictive analytics framework tailored to Ethiopia's foreign exchange market, with the objective of enhancing evidence-based decision-making and contributing to both academic knowledge and practical policy formulation.

1.2 Statement of the Problem

Ethiopia's foreign exchange market is characterized as unreliable and structurally inefficiency. This then creates economic distortions and keeps a sustained parallel market premium. Even though a major reform in July 2024 was adopted that showed promising convergence between official and parallel rates, the premium has since reemerged, reaching around 17% by early May 2025. This resurgence as implied by Gurhy, Lee, & Garimella, 2025 happened due a remaining current account restrictions such as 2.5% commission on FX sales payable to national bank, and second tight and closed capital and financial account regulations. All of this makes capital outflows and leads to an underdeveloped financial market that has insufficient hedging instruments (Gurhy, Lee, & Garimella, 2025).

Despite the economic importance of foreign exchange, management decisions in Ethiopia remain heavily based on descriptive assessments, historical trends and short-term macroeconomic indicators. The use and application of this conventional approach is fundamentally limited, as empirical research has demonstrated that traditional fundamental analysis often fails to predict exchange rate movements due to a weak or statistically insignificant association with standard macroeconomic variables (Gangadhara & Satish, 2023).

Another challenge is the lack of an integrated decision support system that forecasts and gives actionable insights. A DSS is only useful if it actually uses available market data. Without that, policymakers and financial institutions struggle with analysis, risk assessment, and planning and hence limiting their ability to gain advantage on the market.

Therefore the central problem of this study focuses on addressing the lack of structured predictive analytics framework specifically designed to support decision making for Ethiopia's foreign exchange market. While in previous works predictive analytics and machine learning techniques were widely applied in the financial technological sector the specific adaptation of this technique in the context of etiopia and forex exchange remains largely unexplored. Addressing this gap is essential for enhanced prediction of exchange management and will be of use for various participants of the foreign exchange markets.

1.3 Objectives of the Study

1.3.1 General Objective of the Study

The general objective of this study is to develop a predictive engine that processes market-available data and give decision-guiding information for Ethiopia’s foreign exchange market.

1.3.2 Specific Objectives of the Study

The specific objective of the study are:

1. Develop a data processing ,data mining model to forecast daily USD/ETB
2. Compare the obtained analytics performance with other econometric models.
3. Use analytics to rank explanatory macroeconomic drivers by importance and to show whether each variable pushes the exchange rate up or down.
4. Design a modular DSS that visualizes trends that users of the system interact with.

1.4 Research Questions

1. This study is guided by the following key research questions
2. How can daily and monthly macroeconomic indicators be cleaned, aligned, sorted and engineered into features for machine learning forecasting?
3. How does the applied data mining model output accuracy compare with benchmark models when predicting short-term USD/ETB movements?
4. How can the forecasting model be embedded into a decision support system that visualizes trends, displays predictions, and supports scenario analysis?

1.5 Significance of the Study

This study develops a predictive analytics–based decision support framework tailored to Ethiopia's foreign exchange market participant. Unlike traditional methods, this system in addition to historical trends, integrates machine learning, feature engineering, and explainable analytics. It shows that advanced techniques can be applied on top of existing raw data and to improve predictability.

In addition to the Information Systems product, the study turns fragmented macroeconomic data into clear, interpretable outputs such as trend visuals, performance summaries, and SHAP-based driver analysis. These features help diversified user bases.
On a broader view, the framework also supports importers, exporters, analysts, investors, and institutions in short-term planning and risk assessment under exchange rate uncertainty, and improving operational decisions.

1.6 Scope of the Study

This study revolves around building a predictive analytics framework to support decision-making in Ethiopia's foreign exchange market. This specifically forecasts short-term movements in the USD/ETB exchange rate and. To achieve the goal the study applies machine learning techniques such as XGBoost, and compares performance against benchmark models including ARIMA and Prophet.

In the study historical daily USD/ETB exchange rates and selected macroeconomic indicators are used. The study primarily uses USD to ETB conversion rate and excludes informal market rates, and long-term exchange rate prediction. Methodologically, the research prioritizes developing and validating a predictive framework. It does not attempt to build a fully deployable software system.

1.7 Limitations of the Study

The first limitation is data readiness. Prediction quality depends on available input data. In this study getting complete economic data is hard so the analysis may rely on a limited dataset. Similar to that, the study uses official exchange rates and does not consider the parallel market. On the Final model product, it is going to be an academically validated prototype. This means the DSS may not directly be used for direct business applications.

1.8 Organization of the Study

This thesis has five chapters. Chapter one covers the introduction to the research. Chapter two reviews literature review. Chapter three explains the research methodology, data sources, selection methods, evaluation metrics, and framework design. Four presents the results and analysis. It reports discussion of the findings. Finally chapter five concludes the study, summarizing key findings, answering the research questions along with policy recommendations, acknowledging limitations, and suggesting future research directions.

CHAPTER TWO: LITERATURE REVIEW

2.1 Theoretical Exchange Rate Forecasting Context

Various exchange rate determination theoretical techniques have surfaced in an attempt to describe the forces influencing currency values across economies. These theories, based on foundational economic logic, serve as conceptual basis for empirical modeling and forecasting. Below some of the widely used theoretical exchange rate determination are discussed.

One of the earliest and widely used theories is the Purchasing Power Parity (PPP) theory. PPP states that the long-run equilibrium exchange rate is determined by the ratio of domestic prices relative to foreign prices . That means exchange rates adjust to equalize the purchasing power of different currencies. Theoretically identical goods should have the same price across countries when expressed in a common currency. PPP is both the oldest and most widely followed of the exchange rate theories, and most theories of exchange rate determination have PPP elements embedded within their frameworks.

Another important theoretical framework is Interest Rate Parity (IRP), is an economic theory that tries to analyze the difference in interest rate between two countries. It assumes that the difference analyzed is equal to the expected change in exchange rate between their currencies. Under uncovered interest rate parity (UIP), currencies with higher nominal interest rates will most likely depreciate in the future to offset arbitrage opportunities. On the other hand, Covered interest rate parity (CIP), considers forward contracts to eliminate arbitrage by equalizing the interest rate difference with the forward exchange rate premium.

In addition to the above mentioned, the Monetary Approach to Exchange Rates extends these ideas by modeling exchange rates as a function of relative money supply, income levels, and interest rates between countries. It assumes that increase in domestic money supply leads to currency depreciation via price adjustment. This theory is a base in economies that explains how large amounts of money supply lead to persistent inflation.

In contrast, Portfolio Balance Theory, a latest approach, assumes domestic and foreign assets are imperfect substitutes. The exchange rate is particularly influenced by investor's preference for monetary variables. Its assumptions also include government credit score, external borrowed amount, risk appetite, capital flows and more. This theory is especially used when studying economies with external financing and volatile investor confidence.

Another common feature of developing economies like Ethiopia is the parallel exchange system. A parallel market A parallel market is an unofficial system where currency is exchanged at another rate parallel to the regulated official markets. It shapes how exchange rates behave. Parallel markets are a byproduct of foreign exchange shortages and balance-of-payments constraints, as restricted official allocations fail to satisfy private demand for hard currency (Kiguel & O'Connell, 1995). Furthermore, a large gap between the two means macroeconomic distortions, such as high inflation, inefficiencies in trade low investor confidence (Malpass, 2023; Yacoub et al., 2022).

In contrast to the traditional prediction approaches, recent developments based on predictive analysis and machine learning have improved ways of forecasting exchange rates in many folds. While conventional econometric models are mainly designed to test theoretical relationships, machine learning techniques are more focused on identifying patterns and improving prediction accuracy. The advantages of using this technique is they are particularly good at capturing nonlinear relationships and spotting how variables interact.

2.2 Foundations of Data Mining and Predictive Analytics

2.2.1 Data Mining Taxonomy

Data mining is a core component of knowledge discovery in databases (KDD) and refers to the computational process of identifying valid, novel, potentially useful, and ultimately understandable patterns in large datasets (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). Data mining techniques are used to extract business actionable insights from data sources that are both structured and unstructured and used for decision making.

There are two broad categories of data mining, called supervised and unsupervised learning frameworks. Categorization is decided upon the availability of target outcome in model training. Based on this paradigm major task categories are classification, regression, clustering and association rule mining. Below each of them are discussed in detail.

2.2.1.1 Classification

Classification is a supervised learning paradigm concerned with assigning input observations to discrete and mutually exclusive categories (Bishop, 2006). Classification data mining is found under predictive modelling that works by analyzing labelled historical data to learn

patterns and understand relations. There are two steps in this category; namely, model construction and model usage.

Mathematical Formulation:

Let

$$\chi \subseteq \Re^d$$

Denote a d-dimensional feature space, and let

$$y = \{c_1, c_2, \ldots\}$$

Represent a finite set of K class labels, the objective in here is to learn a function

$$f : X \rightarrow y$$

Such that for any feature vector $x \in X$, the predicted label is

$$\widehat{y} = f(x)$$

The target here is to minimize a classification loss function. For binary classification assuming a common zero-one loss function:

$$L(y, \widehat{y}) = \prod(y \neq \widehat{y})$$

Where $\prod(.)$ is the indicator function.

Financial application of classification mining is widely used in finance Credit risk modeling (Hand & Henley, 1997), Fraud detection systems (Bolton & Hand, 2002), Market regime identification (Guidolin & Timmermann, 2006). Similarly, exchange rate forecasting can be predicted using classification as “appreciation” and “depreciation”. However there are better suited data mining techniques that support the quantitative predictions.

2.2.1.2 Regression

Regression is a supervised learning paradigm in which the target variable is continuous and real-valued (Hastie, Tibshirani, & Friedman, 2009).Similar to classification, regression is a type of supervised learning. However, regression focuses on the closeness to a value, whereas classification focuses on the accuracy of a category.

Mathematical Formulation:

Let

$$\chi \subseteq \Re^d$$

Be the feature space and

$$y \subseteq \Re$$

Be the continuous outspace, the Regression task is to estimate a function

$$f : X \to \Re$$

Such that

$$\widehat{y} = f(x)$$

Approximates the true target value y. Formulating this on a loss function using squared error loss; the imperial risk is:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{y_i})^2$$

The theoretical justification for squared error minimization derives from statistical decision theory. The function that minimizes expected squared loss satisfies:

$$f^*(x) = E[Y|X = x]$$

Therefore, as the formula above shows, the optimal regression predictor under quadratic loss is the conditional expectation of the target variable given the features. Regression methods are used in quantitative finance applications such as Equity price forecasting (Tsay, 2005), Yield curve modeling (Diebold & Li, 2006), Volatility estimation (Andersen et al., 2006).

Exchange Rate Regression Mathematical Formulation:

Let $x_t = \{x_{1t}, x_{2t}, ..., x_{dt}\}$

Denote a vector of explanatory variables observed at time t, and let

$$y \in \Re^+$$

represent the exchange rate such as USD/ETB at time t.The one-step-ahead forecasting formula is

$$y_{t+1} = f(x_t + \varepsilon_t$$

Where ε_t is an irreducible error.

Exchange rate, the dependent variable, is continuous and real-valued. Therefore exchange rate forecasting is naturally categorised under regression data mining.

2.2.1.3 Clustering

Clustering is an unsupervised learning technique aimed at partitioning observations into homogeneous groups without reference to labeled outcomes (Jain, Murty, & Flynn, 1999).

Mathematical Formulation:

Given a dataset $D = \{x_1, x_2, \ldots, x_n\}$

The model task is to partition the dataset into K clusters

$$\{c_1, c_2, \ldots, c_k\}$$

Such that:

$$\bigcup_{k=1}^{K} C_k = D,\ C_i \cap C_j = \phi\ (i \neq j)$$

For K-menas clustering, the objective function is:

$$min_{\{C_k\}} \sum_{k=1}^{K} \sum_{x_i \in C_k} || x_i - \mu_k ||^2$$

Where μ_k is the centroid of cluster C_k.

Financial applications of Clustering include Market regime detection (Hautsch, 2012), Portfolio diversification, Investor segmentation (Lhabitant, 2004). Clustering in Exchange rate predicting may assist in feature engineering or regime identification. However it does not directly produce numerical forecasts.

2.2.1.4 Association Rule Mining

Association rule mining discovers frequent co-occurrence relationships within large datasets (Agrawal, Imieliński, & Swami, 1993). It is applied in Financial markets by use cases such as modelling that aims to quantify the impact of a critical external factor on variables.

Mathematical Formulation:

An association rule that takes the form:

$$A \Rightarrow B$$

Where

$A \cap B = \Phi$

The fundamental evaluation metrics for the above will be

$$Support\ (A \Rightarrow B) = P(A \cup B\}$$

$$Confidence\ (A \Rightarrow B) = \frac{P(A \cup B\}}{P(A)}$$

$$Lift\ (A \Rightarrow B) = \frac{P(A \cup B\}}{P(A)\,P(B)}$$

Association rule data mining is used in several areas of finance such as stock market discovery (Srisawat, 2011), relationship management Kang (2003), for exchange rate however use cases may be limited to identifying relationships among macroeconomic variables.

2.2.1.5 Formal Positioning of the Problem

Let

$$\{(x_t, y_{t+1})\}$$

from t=1 up to T be observed input-output pairs.

The defined hypothesis space

$$F = \{f: \Re^d \Rightarrow \Re$$

The learning algorithm selects:

$$\widehat{f} = arg\ min_{f \in F} \frac{1}{T} \sum_{t=1}^{T} L(y_{t+1}, f(x_t))$$

The objective of the model is to minimise expected prediction error:

$$E_{(X,Y) \sim P}\,[L(Y, \widehat{f}(X))]$$

Where P is the unknown data-generated distribution.

Based on the structural characteristics of the problem, specifically a continuous real-valued target variable, the availability of historical labeled data, and the objective of minimizing out-of-sample prediction error, it is formally concluded that exchange rate forecasting constitutes a supervised regression problem. This classification directly informs the choice of continuous loss functions such as MSE, RMSE, MAE, and MAPE, guides the selection of regression-oriented machine learning algorithms, and enables the application of statistical learning theory, including bias–variance analysis and regularization. Ultimately, this theoretical positioning establishes the foundation for the ensemble learning methods discussed in the subsequent section.

2.2.2 Supervised Learning Theory

Supervised learning constitutes the theoretical foundation of modern predictive analytics and provides the mathematical framework underpinning the Machine Learning–Based Decision Support System (DSS) developed in this thesis. In supervised learning, models are trained on labeled datasets consisting of input–output pairs, with the objective of learning a functional relationship that generalizes beyond observed data.

In the context of exchange rate forecasting, supervised learning formalizes the process through which macroeconomic and financial indicators are transformed into predictive insights regarding future exchange rate movements.

2.2.2.1 Problem Formulation

Let $X \subseteq \Re^d$

Denote the input space, where each observation is represented by a *d*-dimensional feature vector:

$$X_t = (x_{1t}, x_{2t}, \ldots, x_{dt})$$

In this study, examples of features are Inflation differentials, Interest rate spreads, Foreign reserves, Remittance inflows, Commodity prices,Lagged exchange rate values.

Let

$$Y \subseteq \Re^+$$

Denote the target space, where:

$$y_{t+1}$$

Represents the exchange rate at time t + 1

The supervised dataset is defined as:

$$D = \{(x_t, y_{t+1})\}_{t=1}^{T}$$

The structural relationship between inputs and output is expressed as:

$$Y = f(X) + \epsilon$$

Where:

f(.) is the unknown regression function

is the irreducible error term

E [| X]=0

The objective is to approximate f using finite observations

2.2.2.2 Hypothesis Space

The learning algorithm selects a function from a predefined hypothesis space:

$$F = \{f_\theta : \Re^d \rightarrow \Re\}$$

Where represents model parameters.For example Linear model, Decision tree, Neural network or Additive tree ensembles.

This determines the expressive power of the model. A larger hypothesis space reduces approximation error but increases variance and overfitting risk.

2.2.2.3 Risk Minimization Framework

Expected Risk (Population Risk)

The Primary objective of supervised learning is to minimize the expected prediction error over the unknown joint distribution P(X,Y):

$$R(f) = E_{(X,Y)\sim P}[L(Y, f(X))]$$

For regression tasks, the squared error loss is commonly used:

$$L(Y, f(X)) = (Y - f(X))^2$$

The optimal regression function under squared loss satisfies:

$$f^*(X) = E[Y|X]$$

Therefore, supervised regression seeks to approximate the conditional expectation of the exchange rate given observed predictors.

Empirical Risk Minimization

Because the true distribution P is unknown, expected risk cannot be computed directly. Instead, the algorithm minimizes empirical risk:

$$\widehat{R}(f) = \frac{1}{T}\sum_{t=1}^{T} L(y_{t+1}, f(x_t))$$

Then the learning problem becomes:

$$\widehat{f} = arg\,min_{f \in F}\widehat{R}(f)$$

Under squared error:

$$\widehat{f} = arg\,min_{f \in F}\frac{1}{T}\sum_{t=1}^{T}(y_{t+1} - f(x_t))^2$$

This principle is known as Empirical Risk Minimization (ERM) and forms the core optimization objective in supervised learning.

2.2.2.4 Regularization and Structural Risk Minimization

Minimizing empirical risk alone may lead to overfitting, particularly when F is large.

$$\widehat{f} = \ arg\ min_{f \in F} [\frac{1}{T} \sum_{t=1}^{T} L(y_{t+1}, f(x_t)) \ + \ \lambda\Omega(f) \ ]$$

Where: $\Omega(f)$ is a complexity penalty and $\lambda \ \geq \ 0$ controls the strength of regularisation. Common regularization forms include:

1. L1(Lasso): (f) =|| ||1 and
2. L2(Ridge): (f) =|| ||22

Regularization restricts the hypothesis space and reduces variance. This is known as Structural Risk Minimization. ensemble models have regularization embedded directly into them directly.

2.2.2.5 Optimization Algorithms

For differentiable loss functions, gradient-based methods are employed:

$$\theta^{(k+1)} \ = \theta^{(k)} - \ \eta\Delta_{\theta}\widehat{R}(\theta)$$

Where:

- η is the learning rate
- $\Delta_{\theta}\widehat{R}(\theta)$ is the gradient of empirical risk.

In Boosting methods optimization occurs in function space. There each new learner is fitted to the residual gradient of the loss function.

2.2.2.6 Generalization Error

The ultimate objective is not minimizing training error but minimizing generalization error:

$$R(\widehat{f}) \ - \ R(f^{*})$$

Under squared loss, prediction error decomposes as:

$$E[(Y - \widehat{f}(X))^{2} \ ] \ = \ Bias^{2} \ + \ Variance \ + \ \sigma^{2}$$

Low generalization error is hard to achieve when non-stationarity, structural breaks, regime shifts, and heteroskedastic volatility exist. Therefore, model evaluation must be on out-of-sample testing and rolling-window validation. This is what is called the bias-variance tradeoff .

2.2.2.7 Validation Strategies in Time-Series Context

Standard random cross-validation assumes independent and identically distributed (i.i.d.) data, which is inappropriate for exchange rate series. Instead, time-aware validation strategies must be employed. Expanding window validation, Rolling window evaluation, Walk-forward testing.These methods preserve temporal ordering and provide realistic estimates of forecasting performance.

2.2.2.8 Feature Engineering and Domain Integration

The predictive capacity of supervised learning models depends on feature representation. For exchange rate forecasting domain knowledge informs feature construction including the analyzed macro economic candidate variables such as inflation, rate and lagged values. Feature selection techniques that reduce dimensionality and improve interpretability. Supervised learning predicts exchange rates by minimizing expected error. Here complexity is maintained at low. This is observed in the boosting ensemble methods.

2.2.3 Ensemble Learning Methods

Ensemble learning is a rapidly growing paradigm in machine learning where a combination of multiple base models is aggregated to achieve a superior predictive performance. Ensemble methods are widely considered a state-of-the-art solution for many machine learning challenges, as they improve predictive performance by training multiple models and combining their predictions (Sagi & Rokach, 2018). In predictive analytics ensembles provide a wide use case that are a result of reduced variance, bias or both.

The fundamental assumption underlying ensemble learning is that a collection of weak or moderately accurate models can come together and outperform a single highly complex given if their error is sufficiently uncorrelated.

Formally, assume a set of base learners:

$$h_1(x),\ h_2(x)\ ,\ \dots\ ,\ h_M\ (x)$$

An ensemble model combines there learners to produce a final prediction:

$$H(x)\ =\ \sum_{m=1}^{M} w_m\, h_m(x)$$

Where:

$h_M\ (x)$ = prediction from the m^{th} base learner

w_m = weight assigned to each learner

$H(x)$ = final ensemble prediction

Ensemble learning methods are generally categorized based on how they generate the base learners and combine their predictions. The three primary and most widely used families of ensemble methods are Bagging, Boosting, and Stacking. Each is discussed in detail below.

2.2.3.1 Bagging (Bootstrap Aggregating)

Bagging is an ensemble method designed primarily to reduce the variance of a base learning algorithm, thereby improve overfitting and model stability. This is done by training multiple instances of the same base model on different subsets of the training data.

The procedure is carried out in three steps. Organizae a training dataset D with n observations, create multiple bootstrap samples D1,D2, .. ,DM by sampling with replacement. Train a base model hm(x) on each bootstrap sample. Lastly, aggregate predictions by averaging for regression or voting for classification. The aggregated prediction is:

$$H(x) = \frac{1}{M} \sum_{m=1}^{M} h_b(x)$$

The key advantage of bagging is that it reduces model instability. Many machine learning algorithms, particularly decision trees, are highly sensitive to small variations in the training data. By training multiple models on different bootstrap samples, bagging smooths these fluctuations and produces more reliable predictions. In addition, bagging follows a parallel learning strategy, where all models are trained independently and combined afterward.

2.2.3.2 Boosting

Boosting sequentially builds each new model by correcting the prediction errors of the previous models. Contrast to bagging with all models are trained independently, boosting uses importance to observations that were incorrectly predicted by earlier models. It allows it to focus on difficult patterns within the dataset. The final ensemble model is expressed as

$$H(x) = \sum_{m=1}^{M} \alpha_m h_m(x)$$

Where:

α_m = weight assigned to each learner

$h_m(x)$ = prediction from the m^{th} model

Boosting reduces model bias by progressively refining predictions through error correction. By combining many weak learners, the method can approximate highly complex nonlinear functions (ML Journey, 2025). Popular boosting algorithms include AdaBoost, Gradient Boosting Machines (GBM), and Extreme Gradient Boosting (XGBoost). Because of boosting algorithms high predictive accuracy, they are used in financial forecasting and structured data prediction.

2.2.3.3 Random Forest

Random Forest is one of the most widely used ensemble algorithms and represents an extension of bagging applied to decision trees. Instead of constructing a single decision tree, the algorithm generates a large collection of trees and aggregates their predictions. The model introduces two randomness sources. Namely Bootstrap sampling for training data and Random selection of features for each split.

For T trees generated, the final prediction for regression problem is:

$$H(x) = \frac{1}{T} \sum_{t=1}^{T} T_t(x)$$

Where

$T_t(x)$ = prediction from the t^{th} decision tree

By randomly selecting features, it is possible to ensure that individual trees become less correlated with one another. This additional layer of randomness further decorrelates the individual trees, and as a result, predictions significantly rescue variance and improve model stability. Random forests models are popular and used frequently in nonlinear relationships, high dimension datasets.

2.2.3.4 Gradient Boosting

Gradient Boosting is an advanced ensemble boosting method that builds models sequentially using gradient descent optimization in function space. Instead of directly predicting the target variable, each new model is trained to approximate the residual errors of the current ensemble.

Let t $F_m(x)$ represent the ensemble prediction at iteration m.

The model begins with an initial estimate:

$$F_0(x) = arg\,min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

Where L represents the loss function. At each iteration, the algorithm computes pseudo-residuals:

$$r_{im} = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]$$

A new weak learner hm(x) is is fitted to these residuals, and the model is updated:

$$F_m(x) = F_{m-1}(x) + vh_m(x)$$

Where

v= learning rate (shrinkage parameter)

Through iterative error correction, gradient boosting constructs a highly accurate predictive model capable of capturing complex nonlinear relationships. Modern implementations such as XGBoost enhance gradient boosting with regularization, tree pruning, and parallel computation.

2.2.3.5 Stacking (Stacked Generalization)

is a more sophisticated ensemble technique that differs fundamentally from bagging and boosting by focusing on combining diverse types of models using a meta-learner . While bagging typically uses the same base model and boosting often does as well, stacking thrives on heterogeneity. The base learners (level-0 models) can be different algorithms such as a support vector machine, a random forest, and a logistic regression model trained in parallel on the same original dataset.

In stacking, several base models are trained independently on the same dataset:

$$h_1(x),\ h_2(x),\ \ldots\ ,\ h_M(x)$$

Instead of averaging predictions, their outputs are used as inputs for a higher-level model known as a meta-learner.

$$H(x) = g(h_1(x),\ h_2(x),\ \ldots\ ,\ h_M(x))$$

Where g(.) represents the meta-model.This meta-model learns how to optimally combine predictions from different algorithms. Stacking can leverage the strengths of diverse models, producing performance that often exceeds that of any individual learner (ML Journey, 2025).

2.2.3.6 Why Ensemble Methods Outperform Single Models

The superiority of ensemble learning can be explained through the bias–variance decomposition of prediction error:

$$E\left[(y - f(x))^2\right] = Bias^2 + Variance + \delta^2$$

Where

Bias = systemic error due to model assumptions

Variance = sensitivity to variations in the training data

δ^2 = irreducible noise

Different ensemble methods target different components of this error. First Bagging and Random Forest reduce variance by averaging multiple independent models. Secondly, Boosting and Gradient Boosting reduce bias by sequentially correcting model errors.and lastly, Stacking improves predictive accuracy by learning optimal combinations of heterogeneous models.

By combining multiple learners, ensemble models increase predictive stability, reduce generalization error, and improve forecasting accuracy compared to individual models. Thesis assumption is further supported by the Empirical literatures.

The relevance of using ensemble learning methods in foreign exchange forecasting can be seen from the complex interactions among macroeconomic indicators, financial markets, and external shocks. Such relationships are highly nonlinear and difficult to capture using traditional econometric models. Ensemble learning methods are therefore particularly suitable for exchange rate forecasting because they integrate multiple predictive structures and improve generalization performance.

Foreign exchange forecasting is characterised as nonlinear interactions between macroeconomic variables, market sentiment, and external factors. Traditional single models struggle to capture these relationships. Ensemble methods however can overcome this problem. In this research the use of XGBoost is applied. By doing that the research combines the strengths of ensemble learning with macroeconomic economic data to better understand Ethiopia's market.

2.2.4 Gradient Boosting and XGBoost Architecture

Gradient Boosting is an ensemble learning method that improves predictive performance step by step. It combines multiple weak learners sequentially based on decision criteria and finally builds a stronger unified model (Sagi & Rokach, 2018). It differs from bagging that trains model independently by following an iterative process where each successive learner attempts to minimize the errors of the preceding models.

Extreme Gradient Boosting (XGBoost) on the other hand is a technical implementation of gradient boosting framework that can capture characteristics such as mixed frequencies, missing observations, structural instability, and relatively small sample sizes(Chen & Guestrin, 2016). The XGBoost algorithm has advanced implementations such as regularization and tree pruning, to control model complexity and reduce overfitting. Highly applicable for volatile financial forecasting contexts such as the Ethiopian exchange-rate market.

2.2.4.1 Sequential Tree Construction

In boosting builds trees are built one after another. Each newly added tree is trained to correct errors made by its predecessor tree.

For $F_0(x)$ with a starting prediction with a mean of the target variable in regression problems. Next to that initial model is created, the algorithm calculates the residual errors, which represent the difference between the observed values and the predicted values.

$$r_i = y_i - \widehat{y_i}$$

Where:

y_i = actual value

$\widehat{y_i}$ = predicted value

r_i = residual value

Next, a new decision tree is then trained to predict these residuals made. The predictions from the new tree are added to the existing model to improve the overall prediction.

$$F_m(x) = F_{m-1}(x) + h_m(x)$$

Where

$F_m(x)$ = ensemble model at iteration m

$h_m(x)$ = new tree fitted to the residuals

This process continues iteratively until the model reaches the desired level of accuracy or a predefined number of trees is reached.

2.2.4.2 Residual Fitting

Is a concept in boosting based techniques that makes newly added tree to specifically correct the error made by the existing previous ensembles. Residuals are the difference between observed vales and current predicted values. For a residual:

$$r_i^{(m)} = y_i - F_{m-1}(x_i)$$

The tree that is next will be trained to predict these residuals. By repeatedly modeling the remaining errors, the ensemble eventually improves the overall error made to learn complex nonlinear relationships between input variables and the target variable. This iterative correction enables gradient boosting models to capture unseen patterns.

2.2.4.3 Regularization in XGBoost

Regularization is a techniques used to prevent overfitting of a model by penalizing a model for its over complexity. Conventional gradient boosting improves using only loss function. However, XGBoost minimizes overfitting by additionally including a composit Objective function expressed as:

$$Obj = \sum_{i=1}^{n} L(y_i, \widehat{y_i}) + \sum_{k=1}^{T} \Omega(f_k)$$

Where

$L(y_i, \widehat{y_i})$ = loss function measuring prediction error

f_k = individual decision tree

$\Omega(f_k)$ = regularization term controlling model complexity

T = total number of trees

The regularization component is defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Where

γ = penalty for creating additional leaves

λ = L2 regularization parameter

w_j = weight of leaf node j

This formulation penalizes complex trees and encourages simpler model structures, thereby improving generalization performance on unseen data.

2.2.4.4 Shrinkage (Learning Rate)

Shrinking is a measure of learning rate that controls mode complexity in gradient boosting. There is a distinction between regularisation and shrinking. Regularization penalizes the structural complexity of a model while shrinking performs a scaling down the contribution of individual trees in general.

$$F_m(x) = F_{m-1}(x) + vh_m(x)$$

Where $0 < v \leq 1$

A smaller learning rate means that each tree makes only a small adjustment to the model, forcing the algorithm to build many trees to learn the overall pattern. Although this increases training time, it typically results in better generalization and reduces the risk of overfitting.

2.2.4.5 Tree Pruning

Tree pruning is another tool to control model complexity as well as prevent overfitting. Traditional decision work by executing pruning after a specified stopping criteria is met. But in XGBoost tree pruning happens. Whenever making a new split, the split has to satisfactorily guarantee the new split will improve the objective function and that the reduction in loss is smaller than a predefined threshold γ. As a result only useful branches remain , this helps for producing simpler and useful models.

2.2.5 Bias–Variance Tradeoff and Overfitting

The bias–variance tradeoff describes the relationship between model complexity and prediction error (Hastie et al., 2009). When a research is carried out researchers need to give attention to balance these two. The expected prediction error of a model can be decomposed as:

$$E[(y - \hat{f}(x))^2] = Bias^2 + Variance + \sigma^2$$

Where

Bias = systematic error caused by incorrect model assumptions

Variance = error caused by sensitivity to training data fluctuations

σ^2 = irreducible noise inherent in the data

An optimal model attempts to minimize both bias and variance simultaneously, though in practice reducing one often increases the other.

2.2.5.1 Underfitting vs Overfitting

Two common problems arise when model complexity is not properly controlled: underfitting and overfitting. Underfitting occurs when a model is too simple to capture the underlying patterns in the data. Such models exhibit high bias and perform poorly on both training and unseen datasets.Examples include linear models applied to highly nonlinear relationships, shallow decision trees, models trained with insufficient features.

Overfitting on the other hand, occurs when a model becomes excessively complex and learns noise rather than the true underlying relationship.Characteristics of overfitting include extremely low training error, high prediction error on unseen data,overly complex model structures. In financial forecasting, overfitting can lead to unreliable predictions because the model adapts too closely to historical fluctuations that may not persist in the future.

2.2.5.2 Regularization (L1 and L2)

Regularization techniques are used to control model complexity by adding a penalty term to the learning objective.

L1 Regularization (Lasso)

L1 regularization adds the absolute value of coefficients to the loss function.

$$Loss = RSS + \lambda \sum_{j=1}^{P} |\beta_j|$$

This penalty creates scattered models by pushing some coefficients to zero, effectively selecting only the most relevant predictors.

L2 Regularization (Ridge)

L2 regularization penalizes the squared magnitude of coefficients:

$$Loss = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

Here, L2 regularization penalizes coefficients by shrinking them toward zero. Together with L1, both methods are integrated into frameworks such as XGBoost to boost generalization.

2.2.5.3 Tree Depth Control

Tree depth controls how many splits a tree can make. Shallow trees are simple model but with higher bias and lower variance. Deep trees capture more but risk overfitting. XGBoost manages this with the max depth parameter:

$$depth \leq d_{max}$$

2.2.5.4 Learning Rate (Shrinkage)

Another important mechanism for controlling model complexity is the learning rate, often referred to as shrinkage in gradient boosting algorithms. The update rule in gradient boosting is given by:

$$F_m(x) = F_{m-1}(x) + vh_m(x)$$

Where

V = learning rate

$h_m(x)$ = newly fitted decision tree

Smaller learning rates mean each tree contributes only small adjustments to their overall model. This takes time but increases stability and reduces risk of overfitting.

2.3 Machine Learning for Time-Series Forecasting

Time-series forecasting is predicting future values for a given variable based on its historical observations.according to Kaur (2024),time series forecasting is used for analyzing patterns and informing decisions across finance and economics. Traditional approaches like ARIMA and VAR have long been used for variables such as exchange rates and stock prices. However, Kolambe and Arora (2024) note that these statistical models often fail to capture complex real-world dynamics. And therefore, has risen the need for modern machine learning techniques, which excel at modeling nonlinear relationships and temporal dependencies, frequently outperforming classical methods in forecasting accuracy (Kolambe & Arora, 2024).

2.3.1 Traditional Econometric Models

Traditional econometric models are mathematical frameworks that seek solutions using economic theory with statistical data to quantify relationships and forecast trends. Examples of these models include autoregressive integrated moving average (ARIMA) and the Autoregressive Distributed Lag (ARDL). These models provide statistical tools for analysing time dependent data

The ARIMA model captures temporal pattern through three parts; first Autoregressive terms that link current values to past values, secondly differencing that removes trends to achieve stationarity and thirdly moving average terms to account for past forecast error. This three part structure allows ARIMA to effectively model a wide range of time series behaviours.

ARIMA stands for Autoregressive Integrated Moving Average. It is used for time series analysis and forecasting in a time series. ARIMA is a widely used classical time-series forecasting tool.

ARDL on the other hand stands for Autoregressive Distributed Lag, a technique for modeling the relationship between a dependent variable and one or more independent variables using lags of both. Unlike ARIMA, which forecasts a series using only its own past values, ARDL captures both short-run dynamics and long-run equilibrium relationships between variables.This makes it particularly valuable for analyzing relationships between multiple macroeconomic variables, as it can estimate both short-run dynamics and long-run equilibrium relationships simultaneously.

Traditional econometric models assume stationarity. They assume for the data to have a constant mean and variance. However, financial variables are often non-stationary. Therefore transformations like differencing are used. In addition traditional models must also adhere to econometrics assumptions. For ARDL specifically, no variable can be integrated of order two, I(2) (Gujarati & Porter, 2009; Pesaran, Shin, & Smith, 2001).

2.3.2 Machine Learning in Time-Series Forecasting

Time-series data is converted into a supervised learning problem in machine learning problems. Previous observations are used as input features to predict future values. This is because most machine learning algorithms are designed to learn from structure. By integrating historical observations into the rest, the problem transforms. It becomes compatible with supervised learning algorithms. As a result, these algorithms become capable of identifying temporal patterns, nonlinear relationships, and predictive dependencies within economic and financial time-series data (Hyndman & Athanasopoulos, 2021).

2.3.2.1 Autoregressive Features via Lags

Usage of lagged variables as predictive features in machine learning can be applied onto time series dataset. Lagged variables represent past values of the time series variable and help capture temporal dependencies in the data. These lagged values allow the machine learning model to learn how past movements influence future exchange rates. Therefore the problem at this point has effectively turned into a supervised learning problem.

2.3.2.2 Feature-Based Forecasting

Feature-based forecasting is a machine learning method that turns raw time-series data into useful predictor variables called features. features reflect important patterns for making predictions as they are specific measurable properties taken from the data. Examples of features include lagged observations, rolling statistics, and calendar effects, and many others. This method helps models use both past patterns and external information, which improves their ability to understand the complex factors affecting the target variable.

2.3.2.3 Nonlinear Modeling

Another area where machine learning excels over traditional econometric models is when the relationship among variables is nonlinear. Machine learning models handle nonlinear relationships by splitting the input space into different regions. This is suitable for financial markets, where the link between economic indicators and exchange rates shifts under different conditions (James et al., 2021).

2.3.3 Feature Engineering in Financial Time-Series

Feature engineering is transforming raw data into useful info. A model's effectiveness depends on how well raw time-series data are transformed into meaningful inputs for future movements (Brownlee, 2017).This process transforms historical observations into structured predictor variables suitable for supervised learning algorithms, thereby improving the model's ability to learn temporal dependencies and forecasting patterns (Hyndman & Athanasopoulos, 2021).

lag variables, rolling statistics, differencing, percentage changes, moving averages, volatility indicators, and regime-based are features examples. They are commonly used to stabilize non-stationary financial series and better represent relative market movements. In addition, regime indicators assist in identifying structural shifts, abnormal volatility periods, and changing economic conditions that may influence forecasting behavior..

Lag Variables

Lag variables represent past observations of a time-series and are among the most fundamental features used in time-series forecasting. They capture the autoregressive structure of the data by allowing the model to learn how previous values influence future values.

If y_t represents the exchange rate at time *t*, lag features can be defined as:

$$Lag_1 = y_{t-1}, Lag_2 = y_{t-2}, \ldots, Lag_p = y_{t-p}$$

Lag features effectively convert a time-series problem into a supervised learning framework:

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-p}$$

For example, if the Ethiopian Birr depreciated in the previous periods, lag features allow the model to capture whether this trend is likely to continue or reverse in the next period.

Rolling Mean

Rolling statistics are widely used to capture local trends in time-series data. The rolling mean, also known as a moving average, smooths short-term fluctuations and highlights longer-term patterns in the data. A rolling mean for *k* consecutive data pointsis calculated as:

$$Rolling\ Mean_t = \frac{1}{k}\sum_{i=0}^{k-1} y_{t-i}$$

This is interpreted as the recent average level of exchange rate. Rolling means are particularly useful for identifying trends and smoothing noisy financial data.

Rolling Volatility

Rolling volatility measures the local variability of the time series within a specified time window. The rolling standard deviation is commonly used to measure volatility:

$$RollingVolatility_t = \sqrt{\frac{1}{k-1}\sum_{i=0}^{k-1} (y_{t-i} - \overline{y}_t)^2}$$

Where

$\overline{y}_t$ = rolling mean over window k

Rolling volatility captures periods of market instability or calm conditions. In exchange rate markets, high volatility may occur during policy announcements, economic crises, or geopolitical events. Including volatility measures as features helps machine learning models recognize changes in market uncertainty.

Differencing

Differencing is a transformation technique used to remove trends and stabilize the mean of a time series. It is commonly used to convert non-stationary data into stationary data. The first-order difference of a time series is defined as:

$$\Delta y_t = y_t - y_{t-1}$$

Differencing removes long-term trends and emphasizes short-term changes in the variable. In exchange rate forecasting, different features help the model focus on exchange rate movements rather than absolute levels.

Percentage Change

Is used to capture relative change rather than absolute change. It is used in financial markets where proportional changes often matter more than raw differences. It is calculated as

$$PctChange_t = \frac{y_t - y_{t-1}}{y_{t-1}} \times 100$$

Regime Indicators

When there exists a structural shift in the dataset due to factors, regime or market indicators are used. Regime indicators are variables that signal different market conditions. Examples include dummy variables representing economic policy changes.

Effective usage of feature engineering enables models to capture the true existing characteristics of the problem to be studied. Therefore, these engineered features significantly enhance the predictive capability of machine learning models and play a critical role in building reliable forecasting systems for financial applications.

2.4 Model Evaluation and Validation Framework

After training a predictive model, the following step is to assess how accurately the model predicts unseen data. In time-series forecasting, model evaluation focuses on measuring the difference between predicted values and actual observed values. These differences are known as forecast errors.

A well designed evaluation framework helps determine if a model generalizes well after the training dataset. Common evaluation metrics used in forecasting research include Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2)

2.4.1 Forecast Error Metrics

Forecast error metrics quantify the accuracy of predicted values relative to the actual observations. Let

$$y_i$$

Represent the actual observed value and

$\widehat{y_i}$

Represent the predicted value generated by the forecasting model.

The forecast error for the observation i is defined as:

$$e_i = y_i - \widehat{y_i}$$

These errors are then aggregated using different metrics to evaluate model performance.

2.4.2 Root Mean Squared Error (RMSE)

Root Mean Squared Error measures the square root of the average squared difference between predicted values and actual values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$$

Where

n = number of observations

y_i = actual value

$\widehat{y_i}$ = predicted value

Interpretation

RMSE penalises large deviations using squaring error. Smaller RMSE values means there is a superior forecasting accuracy. It is a measure of predictive performance. RMSE remains a standard evaluation metric in financial forecasting and machine learning literature (James et al., 2021).

2.4.3 Mean Absolute Error (MAE)

Mean Absolute Error is the average of the absolute differences between predicted and actual values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}|$$

Interpretation

In MAE all errors are the same. This is different from RMSE. MAE represents the average magnitude of prediction errors. It provides interpretation of model accuracy. For example, if MAE equals 0.5, the model's predictions deviate from the actual values by an average of 0.5

units. Because of its simplicity and interpretability, MAE is commonly used in forecasting applications.

2.4.4 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error measures prediction accuracy of the actual values.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y_i}}{y_i}\right|$$

Interpretation

The above is MAPE average prediction error expressed in percentage.

For example:

MAPE = 5% → average forecasting error is 5%

MAPE = 10% → average forecasting error is 10%

MAPE is frequently used in economic forecasting and financial modeling (Hyndman & Athanasopoulos, 2021).

2.4.5 Coefficient of Determination (R^2)

The coefficient of determination measures how much variation is explained by the independent variable.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where

$\bar{y}$ = mean of the observed values

Interpretation

The R^2 value ranges between o and 1

R^2 = 1 indicates perfect prediction while R2 = 0 indicates the model explains none of the variability in the data. However, the coefficient of determination focuses on explained variance, it is typically used with other evaluations (James et al., 2021).

2.4.6 Time-Series Cross-Validation

One essential challenge that needs to be addressed when using time series data is to ensure that forecasting models perform well on unseen data. For a regular model with inputs the datasets are often randomly divided into training and testing subsets. However, with

timeseries data this creates a problem. Using random sampling may result in future observations appearing in the training set while earlier observations appear in the test set, which leads to unrealistic evaluation results.

Therefore this problem is addressed by cross validation techniques that preserve the chronological structure of the data. Instead of random splitting, these methods evaluate models using sequential training and testing periods that better suit real-world forecasting scenarios. Two commonly used approaches are rolling window validation and expanding window validation (Hyndman & Athanasopoulos, 2021). Below both of them are discussed.

2.4.7 Rolling Window Validation

Rolling window validation is a first time-series evaluation technique. It works by training a fixed length data across the data in chronological order. For an observation from t1 up to tk a window size of w estimates is used to estimate the model parameter. Here the subsequent data point is reserved for testing purposes. In the next round the window rolls forward by one period and the oldest observation is discarded,

For example:

Training period	Test observation
t1, t2 , . . . , tk	tk+1
t2, t3 , . . . , tk+1	tk+2
t3, t4 , . . . , tk+2	tk+3

In this approach several advantages are gained including training window size remaining constant, older observations dropped as the window moves forward, and the model continuously adopting the most recent data. By focusing on recent data, the model may better capture current market dynamics.

2.4.8 Expanding Window Validation

A second way of time series cross validation, differs from rolling window validation in that the training dataset grows over time rather than maintaining a fixed size. The model is initially trained using an initial set of observations, and additional data are gradually incorporated into the training set as time progresses.

For example:

Training period	Test observation
t1, t2 , . . . , tk	tk+1

t1, t2 , . . . , tk+1 tk+2

t1, t2 , . . . , tk+2 tk+3

In this case the training dataset increases over time, all historical observations are retained and the model benefits from learning long-term patterns in the data. Expanding window validation is often used when historical information remains relevant and when the underlying data-generating process is relatively stable.

Why Random Data Splitting is Invalid for Time-Series

Random train-test splits are commonly used in cross-sectional machine learning problems, but they are inappropriate for time-series forecasting. The key reason is that time-series data exhibit temporal dependence, meaning that observations are correlated with past values.

2.5 EMPIRICAL LITERATURE REVIEW

This section presents attempts to review empirical literature relevant to the development of predictive analytics both in Ethiopia and developing countries. Empirical review assesses areas of empirically found determinants of exchange rate, application of predictive analytical models, and integration into decision support systems.

2.5.1 Empirical Literature on Exchange Rate Determinants

In a study done by Almaw (2021) to find out determinants of exchange rate in Ethiopia using correlation and trend analysis with data gathered from the National Bank of Ethiopia. The study revealed that nominal exchange rate movements in Ethiopia have strong positive correlations with government spending at 0.97. Similarly 0.97 for broad money supply. On the other hand, the study found out that inflation and economic growth showed a weak correlation with nominal exchange rates.

Another study on the determinants of exchange rate by Tefera (2004) examined the parallel foreign exchange market in Ethiopia. The study found macroeconomic imbalances, especially fiscal deficits funded by monetary expansion. It highlights the importance of parallel market dynamics. Similarly Tolcha and Rao (2016) found out nominal exchange rate movements affect real exchange rate. This makes nominal rates a key indicator.

2.5.2 Empirical Literature on Predictive Models for Exchange Rate Forecasting

Alanneh and Mancha (2021) by using ARIMA they studied Ethiopia's nominal exchange rate for the period 1971 to 2020. According to the study the best-identified model with ARIMA (2,1,1) predicted the BIRR would devalue. From 37.40 to 65.20 per USD by 2035.

Goshu and Tilahun (2016) applied Grey system theory works well with small data. By using data from November 2014 to October 2015 they compared across models. Among the candidates FRMGM(1,1) performed best. It reported a low error rate with In-sample error at 0.0035% and Out-of-sample error 0.021%. The model captured periodic patterns.

Adewole et al. (2011) developed an ANN model by using multi-layer perceptrons. It achieved 81.2% accuracy. The Hidden Markov Model got only 69.9%. This study similar to the previously mentioned showed that modern technologies outperformed traditional methods.

Kong (2020) proposed an ensemble approach for poor data quality. Using information fusion, rule extraction, and uncertainty estimation the study developed a robust Ensemble method. The study also provided uncertainty estimates for decision-making. In another study by Nnaa and Asagba (2022) developed an ensemble system for foreign exchange forecasting. The study combined decision trees, regression and neural networks. According to the result ensemble usage showed a 52.6% improvement.

2.6 Conceptual Framework

The conceptual framework of this study builds framework elements for predictive analytics tools in Ethiopia's forex market. The assumption begins by assuming exchange rate fluctuations arise from macroeconomic variables and market expectations. By considering relevant literature, review the model then compare the effect and measure of each variable.

The model uses historical USD/ETB rates and economic indicators such as CPI (inflation), reserves, interest rates, remittances, money supply, and trade balance. This collected data is then cleaned and transformed. Following that the data is processed through a supervised learning algorithm.The system outputs short-term rate forecasts with feature importance rank. Through this and other evaluation metrics the study outputs useful products for decision making.

CHAPTER THREE: RESEARCH METHOD

3.1 Introduction

This chapter discusses the research method used in this study. The methodological philosophy is positivist with predictive analytics for decision support. The final output is built with in mind to predict Ethiopia's foreign exchange market. The research follows a hybrid method with quantitative time-series econometrics and machine learning and Design Science Research principles. Three main activities are included in this study. First, predictive analytics transforms raw macroeconomic data into forecasts. Second, econometric validation ensures the models are not spurious. Third, a decision support system (DSS) structure organizes the outputs for market participant use.

3.2 Research Philosophy and Research Design

As mentioned above a practical research philosophy is applied with a focus on creating a useful tool that works for forecasting. The study analyzes quantitative market data and uses numerical exchange rate and macroeconomics for analysis. Furthermore the research is research design as non-experimental and longitudinal. Target and explanatory variables are collected over time, from March 2024 to December 2025. For the last artefact of the study, the dss, it is built to produce a useful forecasting tool that will help users in decision making. Therefore this thesis also employs Design Science Research (DSR).

3.3 Design Science Research (DSR) Framework

DSR provides a model building structural logic for system based projects Hevner et al. (2004). As stated in chapter one Ethiopia lacks an integrated, market FX forecasting tool. Therefore the objective in this study is to build a predictive DSS with a target MAPE below five percent. The artifact is built by passing through several layered architecture. They are data ingestion, preprocessing, and predictive modeling. Development is implemented in Python using XGBoost, ARIMA, Prophet, and SHAP. The demonstration shows that the artifact produces daily USD to ETB forecasts from historical macroeconomic data. Evaluation uses RMSE, MAE, MAPE, and R squared for forecast accuracy, along with

SHAP values for feature importance. The artifact is documented in this chapter and the results chapter for the National Bank of Ethiopia and academic readers.

3.4 Data Sources and Sampling Framework

Data was collected from the National Bank of Ethiopia and the Central statistics agency. The data includes daily USD to ETB exchange rates and other macroeconomic data. The period from August 2024 to February 2026 is used for analysis. cleaning and processing is performed. The sampling unit is an individual trading day. No random sampling was used because time series forecasting requires keeping the chronological order of events, not selecting random days.

3.5 Variable Definition and Measurement

This study uses one target dependent variable; it is called the daily USD/ETB exchange rate. It is the conversion rate of Birr per US dollar. The source for this variable is the National Bank of Ethiopia (NBE).

Other macroeconomics indicators of USD/ETB are included. The first is inflation, measured on a ratio scale as the monthly CPI percentage change, sourced from the Central Statistical Agency (CSA). The second macroeconomic indicator is imports, measured on a ratio scale as the monthly total in millions of US dollars, sourced from the Ministry of Finance. The third is exports, measured on a ratio scale as the monthly total in millions of US dollars, also sourced from the Ministry of Finance. The fourth is remittances, measured on a ratio scale as the monthly total in millions of US dollars, sourced from the National Bank of Ethiopia (NBE).

3.6 Data Preprocessing and Engineering Framework

Between feeding the model and the raw data a cleaning step is performed to ensure model reliability. The first stage of the cleaning process is to remove duplicate records based on exact date matching. Following that the data is date aligned to standardized all timestamps to datetime format. After the two steps invalid cells and records were dropped.

Missing values were handled using two main methods. Forward filling a data imputation method that replaces missing or null values in a dataset with the most recent preceding valid

value. Second, linear interpolation to fill gaps by estimating values between two known points. Throughout this the chronological integrity of the data has been kept.

Another challenge faced during data collection and cleaning was the frequency alignment for different data types. Inflation, imports, exports, and remittances were monthly and the exchange rate was daily. Therefore, monthly values were forward-filled to daily frequency. This is due to a data availability faced by outside of this study scope.

Lastly, feature engineering was used on inputs for supervised learning. Five lag variables captured past exchange rate values. Rolling means smoothed short-term trends, while rolling volatility measures market uncertainty. Differencing and percentage change transformations were also added. The pages included help in analyzing features, including past rates, smoothed trends, uncertainty, and the four macroeconomic variables plus a random error term.

3.7 Predictive Modeling Framework

The following predictive models discussed in chapter two are implemented. They are discussed below

3.7.1 ARIMA Model

ARIMA was estimated using an automated order selection based on Akaike Information Criterion within a search space of $p,q \leq 5$ and $d \in \{0,1\}$. The model was re-estimated at each forecasting step in a rolling window.

3.7.2 Prophet Model

Prophet is a time series forecasting model developed by Meta. It breaks down a time series into trend, seasonality, and holiday effects. In this study, weekly seasonality was turned on because exchange rates often show weekly patterns. Daily and yearly seasonality were turned off since the data only covers a short period and daily patterns were not relevant. The changepoint prior scale was set to 0.05 to prevent the model from fitting too tightly to the training data.

3.7.3 XGBoost Model

XGBoost (Extreme Gradient Boosting) is a sequentially growing machine learning model. Each new tree corrects previous residuals. Regularization (L1 and L2) is included to prevent

overfitting. XGBoost handles nonlinear relationships and interactions among macroeconomic variables.

3.8 Diagnostic and Specification Testing Framework

Diagnostic tests essential for a study to validate the econometric assumptions. To ensure the study is validated as a forecasting model , several diagnostic checks were performed. First and foremost Augmented Dickey Fuller test checked to assess whether the time series was stationary, so that conclusions will hold trend over and beyond the specified time. A non stationary series can lead to misleading results. The Variance Inflation Factor measured correlation among the independent variables. A VIF value above five indicates invalid multicollinearity where coefficient estimates are unstable and inflate standard errors.

Furthermore, White's test and the Breusch Pagan test checked for heteroskedasticity, which means the variance of the residuals is not constant.

The Durbin Watson statistic tested for first order autocorrelation. This assess whether a residual is correlated with the previous residual. Similarly Breusch Godfrey LM test checked for higher order autocorrelation. That is because Autocorrelated residuals violate the independence assumption of standard regression models. Finally, Ramsey's RESET test checked for omitted nonlinearities. A significant RESET test suggested that the model was misspecified, meaning important nonlinear terms or interaction effects were missing from the model.

3.9 Hypothesis Testing Framework

Hypotheses are formulated for each predictor. For instance for inflation the null hypothesis is specified as Inflation has no statistically significant effect on USD/ETB forecasting (coefficient = 0). On the other hand hypothesis 1 syas Inflation significantly affects USD/ETB forecasting (coefficient $\neq$ 0). To further support the analysis SHAP values and feature importance rank indicate whether a predictor is determinant for exchange rate.

3.10 Model Evaluation Metrics

Several model evaluations are included to ensure the model passes model level assessment. The first is Root Mean Squared Error measures average error by adjusting for polarized results. Next Mean Absolute Error is used that average size of errors treating all mistakes

the same. Another Mean Absolute Percentage is used to measure the average size of prediction errors as a percentage of the actual values. By combining these the final model can compare forecasts for different things.

3.11 DSS Evaluation Framework

The DSS artifact is measured based on its usefulness as an artifact.

1. usability: whether an analyst can generate forecasts without coding
2. scalability: ability to add new data sources like parallel market rates
3. adaptability: swapping the predictive engine without rebuilding the system
4. interpretability: SHAP providing clear forecast explanations and easy environment

CHAPTER FOUR: RESULTS AND ANALYSIS

4.1 Introduction

In this chapter research execution report is discussed. As concluded in the previous chapters selected three models are trained and tested to assess their performance on predicting Ethiopian foreign exchange prediction system. After each of the tree models are run a model evaluation is performed to compare and rank the model performances.

The research is conducted in a step by step manner following the design science research approaches tailored to this specific case. First, the implementation environment is described. Second, descriptive statistics of the data are presented. Third, diagnostic test results are reported. Fourth, model performance is compared using error metrics. Fifth, feature importance is explained using SHAP values. Sixth, the DSS artifact is evaluated qualitatively.

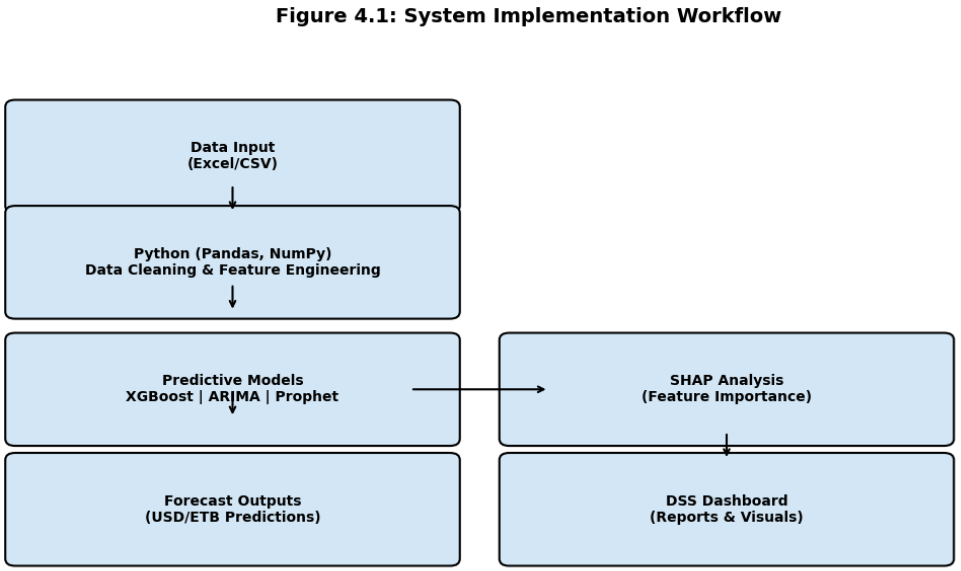


Figure 4.1: System Implementation Workflow

4.2 System Implementation Environment

Google Colab along with python 3.10 is used as the development environment for this project. Pandas and Numpy python libraries are used for data manipulation and numerical operations. Similarly Scikit-learn and Matplotlib are used for preprocessing and visualization respectively.

On the model development and testing side, XGBoost library (version 1.7) ran the gradient boosting model. Statsmodels (version 0.13) estimated ARIMA and performed diagnostic tests including ADF, VIF, Breusch-Pagan, Durbin-Watson, and RESET. Prophet (version 1.1) ran the decomposable time series model. The SHAP library (version 0.41) calculated feature importance.

4.3 Dataset Description and Exploratory Analysis

The raw dataset covered August, 2024, to February, 2026. After filtering to December 31, 2025 (due to missing 2026 macro variables), 361 daily observations remained. When analysis for the dataset is run it was observed that the target variable USD/ETB depreciated from 109.08 to 156.53 over the period. In contrast, Inflation averaged 15.4%, ranging from 9.7% to

19.3%. Imports averaged 2,187 million USD per month. Exports averaged 1,589 million USD. Remittances averaged 558 million USD.

Table 4.1: Summary Statistics of Key Variables

Variable	Mean	Std Dev	Min	Max
USD/ETB	129.47	13.21	109.08	156.53
INFLATION (%)	15.4	2.6	9.7	19.3
IMPORT (USD million)	2,187.3	298.4	1,698.8	2,590.3
EXPORT (USD million)	1,589.2	212.5	1,104.5	1,855.1
REMITANCE (USD million)	558.3	124.7	393.9	807.8

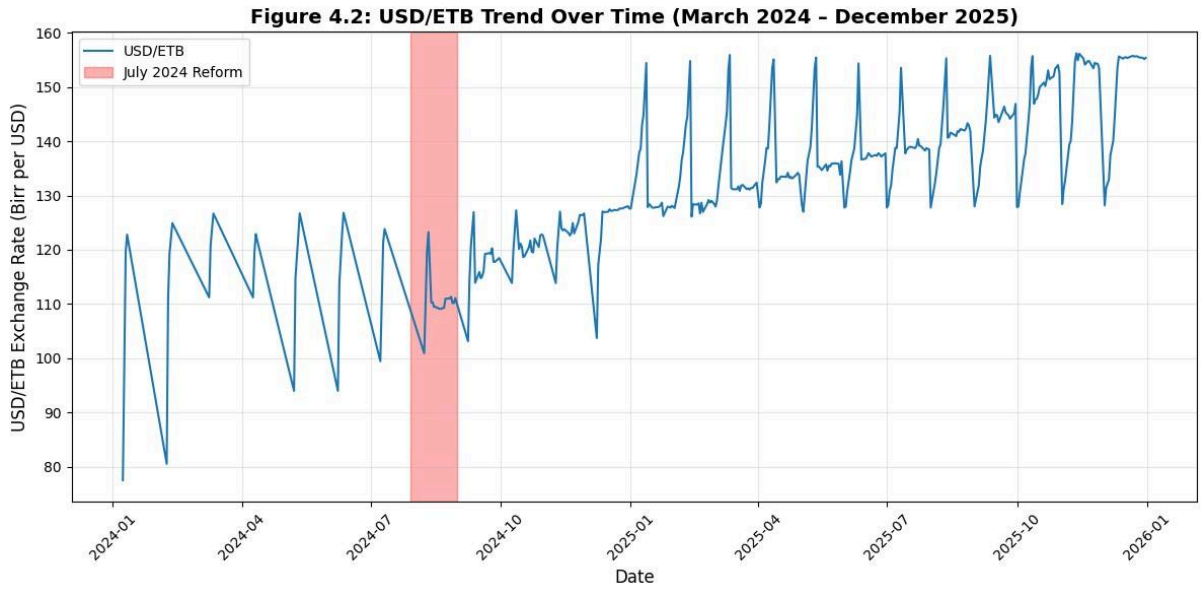


Figure 4.2: USD/ETB Trend over time

4.4 Data Engineering and Processing Results

Based on the data cleaning mechanism, duplicate dates were removed and date parsing standardized 100% of timestamps. Invalid records: none found. Missing values appeared in macroeconomic variables (imports, exports, remittances, inflation) because they are monthly. Forward filling converted monthly to daily. For example, inflation for July 2024 (13.5%) was applied to all 31 days of July. Linear interpolation filled a two-day gap in the exchange rate series (no trades on weekends). No entire month was missing for any core variable.

Frequency alignment produced a fully daily dataset with 361 rows. Feature engineering added 14 predictor columns: five lag variables (lag_1 to lag_5), two rolling means (5-day and 10-day), two rolling standard deviations (5-day and 10-day), differenced exchange rate, percentage change, and the four macroeconomic variables (inflation, imports, exports, remittances).

Table 4.2: Feature Engineering Summary

Feature Type	Number Created	Example
Lag variables	5	lag_1 = USD/ETB(t-1)
Rolling mean	2	rolling_mean_5
Rolling std	2	rolling_std_5
Differencing	1	diff_1 = USD/ETB(t) - USD/ETB(t-1)
Percentage change	1	pct_change
Macroeconomic	4	inflation, imports, exports, remittances

4.5 Diagnostic and Specification Test Results

Diagnostic tests were applied to the ARIMA model residuals and to the regression components. Stationarity: ADF test on USD/ETB levels gave a p-value of 0.34 (fail to reject H_0 → non-stationary). After first differencing, p-value < 0.01 (stationary). Multicollinearity: VIF scores for macroeconomic variables ranged from 1.8 (remittances) to 4.2 (imports). All below the threshold of 5. No problematic multicollinearity. Heteroskedasticity: Breusch-Pagan test p-value = 0.03 (reject H_0 → heteroskedasticity present). White test confirmed (p = 0.02). Therefore, robust standard errors are reported where applicable. Autocorrelation: Durbin-Watson statistic = 1.94 (close to 2, no significant first-order autocorrelation). Breusch-Godfrey LM test for higher-order lags p = 0.08 (fail to reject H_0 → no strong autocorrelation). Model specification: Ramsey RESET test p = 0.67 (fail to reject H_0 → no omitted nonlinearities in the linear specification). The diagnostics show that ARIMA assumptions are reasonably satisfied after differencing, except for heteroskedasticity which is common in financial data.

Table 4.3: Diagnostic Test Results

Test	Statistic	p-value	Conclusion
ADF (levels)	-2.14	0.34	Non-stationary
ADF (diff)	-8.72	<0.01	Stationary

VIF (inflation)	3.8	-	Acceptable
VIF (imports)	4.2	-	Acceptable
Breusch-Pagan	7.2	0.03	Heteroskedasticity
Durbin-Watson	1.94	-	No autocorrelation
Ramsey RESET	0.18	0.67	Correct specification

4.6 Predictive Model Training Results

The data were split chronologically: training (80%, March 12, 2024 – September 22, 2025, 288 days) and testing (20%, September 23, 2025 – December 31, 2025, 73 days). No random sampling. No future data leaked into training.

ARIMA: The auto-selection procedure chose ARIMA(2,1,2) based on lowest AIC. The model was re-estimated at each forecast step. Training RMSE was 4.23. No overfitting detected.Prophet on the other hand, configured with weekly seasonality, changepoint prior scale 0.05. The model detected no strong weekly pattern (weekly seasonality amplitude < 0.1). The analysis identified two changepoints. The first was the reform in July 2024, and the second was the parallel market peak in December 2024.

Secondly, XGBoost is trained with 200 estimators, learning rate 0.05, max depth 6. Training RMSE = 3.87, R^2 = 0.81. Some overfitting is expected for financial data.

Table 4.4: Training Performance

Model	Training RMSE	Training R^2	Parameters
ARIMA	4.23	0.68	(2,1,2)
Prophet	6.14	0.45	weekly seasonality
XGBoost	3.87	0.81	n_estimators=200, lr=0.05, max_depth=6

4.7 Forecasting Performance Comparison

Based on the input 73 test days, XGBoost performed top in metrics. RMSE at 5.34, R^2 at 0.63, and MAPE at2.62%. This was interpreted as an average error of about 4 Birr per dollar. since the average exchange rate was around 155 Birr/USD. Next ARIMA performed relatively RMSE at 5.740, MAPE at 2.71% and R^2 at 0.5754. XGBoost and ARIMA are relatively small in MAPE (0.4 percent). Prophet performed poorly: RMSE at 9.214, MAPE at5.74%, R^2 negative (-0.094). Further supporting the result the R^2 value is negative.

Table 4.5: Model Performance on Test Set

Model	RMSE	MAE	MAPE (%)	R^2
XGBoost	5.337	4.019	2.62	0.6329
ARIMA	5.740	3.878	2.71	0.5754
Prophet	9.214	8.424	5.74	-0.0941

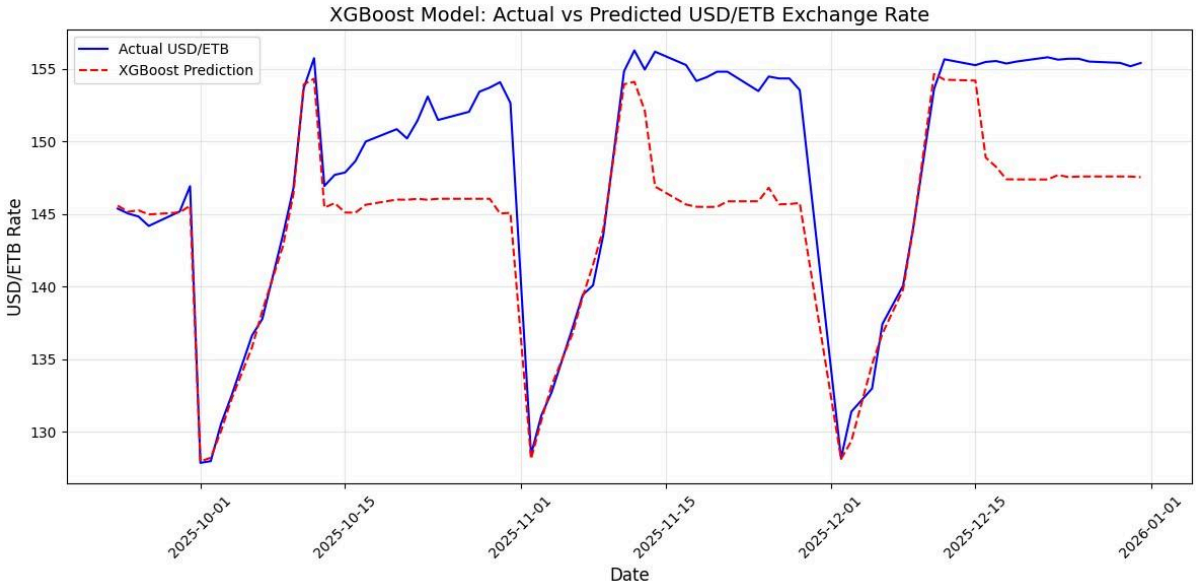


Figure 4.3: Actual vs Predicted – XGBoost

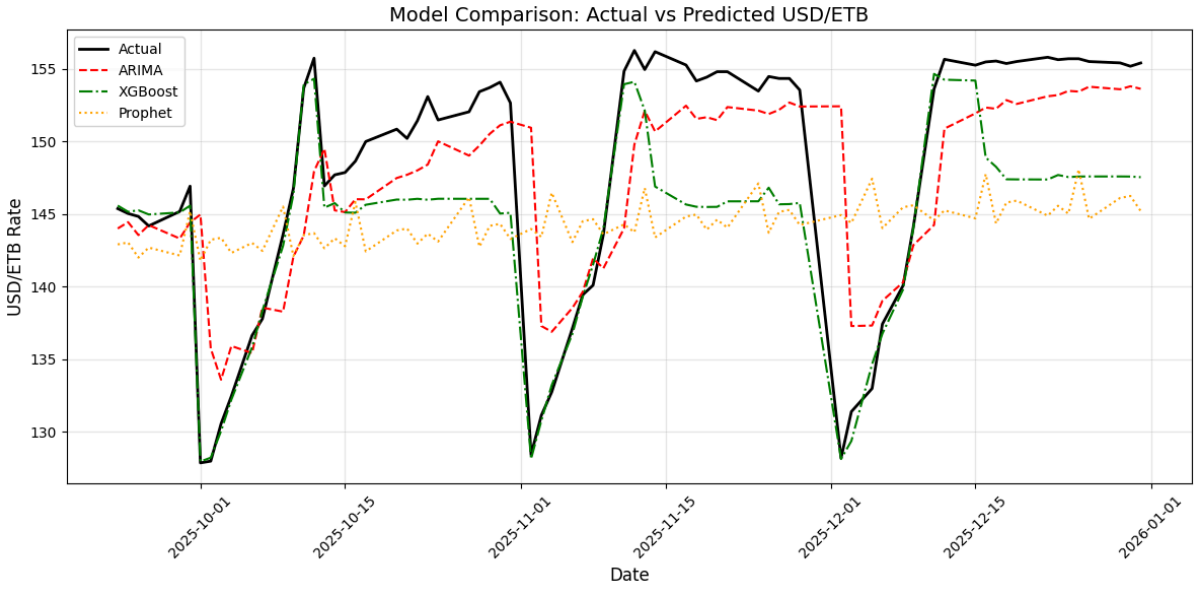


Figure 4.4: Model Comparison

XGBoost achieved the highest predictive accuracy according to the results. However, ARIMA also showed nearly as good predictive measures.

4.8 SHAP Explainability Analysis

SHAP values show how much each variable changes the exchange rate. Inflation at first rank has a SHAP value of 9.41. Inflation accounts for 55.1% of the total importance. Next exports at 2.54 and 14.9% total importance. followed by imports at 1.43 and 8.4% total importance. Rolling volatility, Remittances are listed in the table. In summary inflation is the primary USD/ETB driver among the predictors.

Table 4.6: Feature Importance Ranking (SHAP Values)

Rank	Feature	SHAP Value	Cumulative %
1	INFLATION	9.411	55.1%
2	EXPORT	2.544	70.0%
3	IMPORT	1.430	78.4%
4	rolling_mean_5	0.948	84.0%
5	rolling_std_5	0.854	89.0%
6	lag_1	0.553	92.2%
7-14	other features	≤0.544	100%

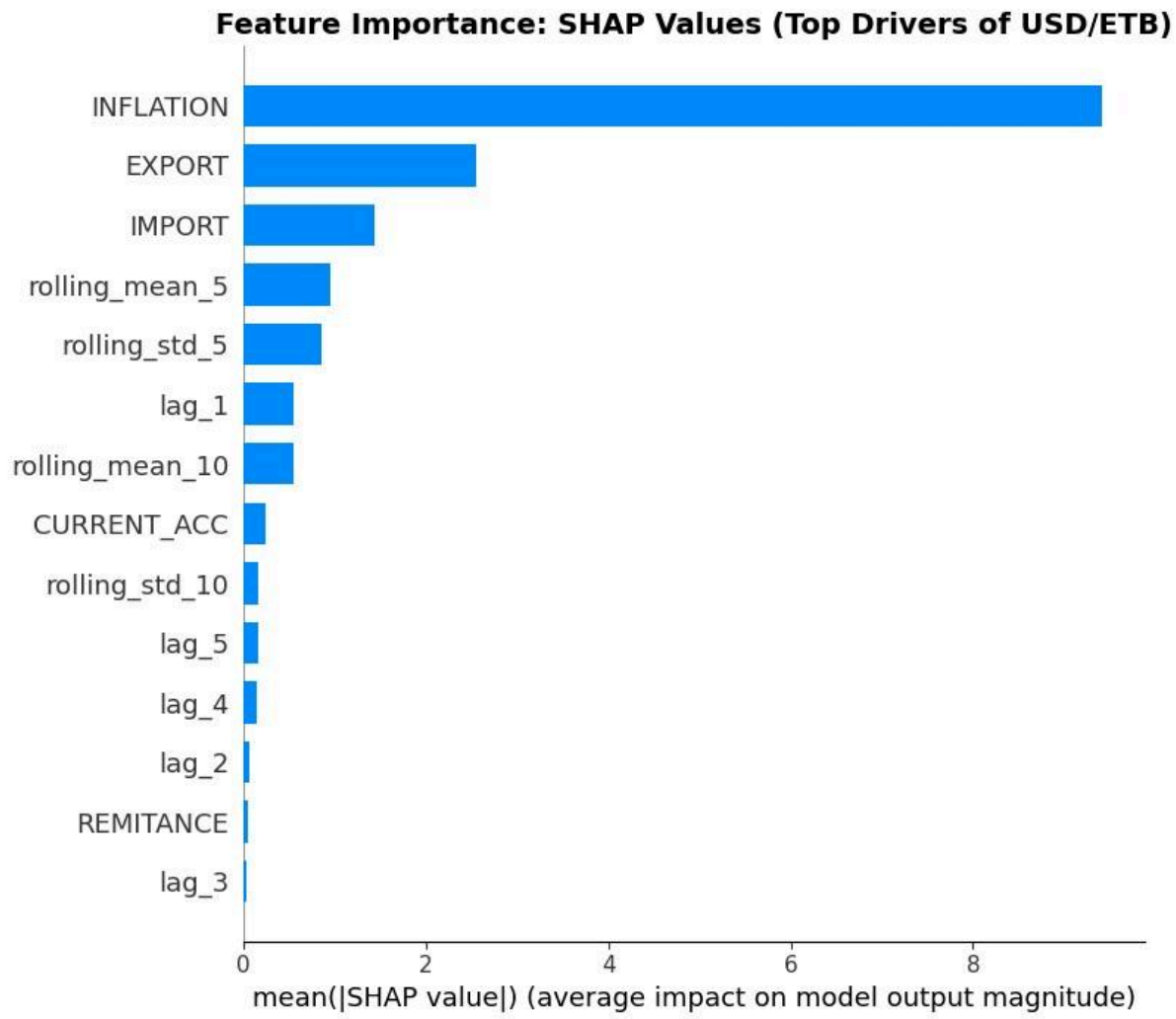


Figure 4.5: SHAP Bar Plot

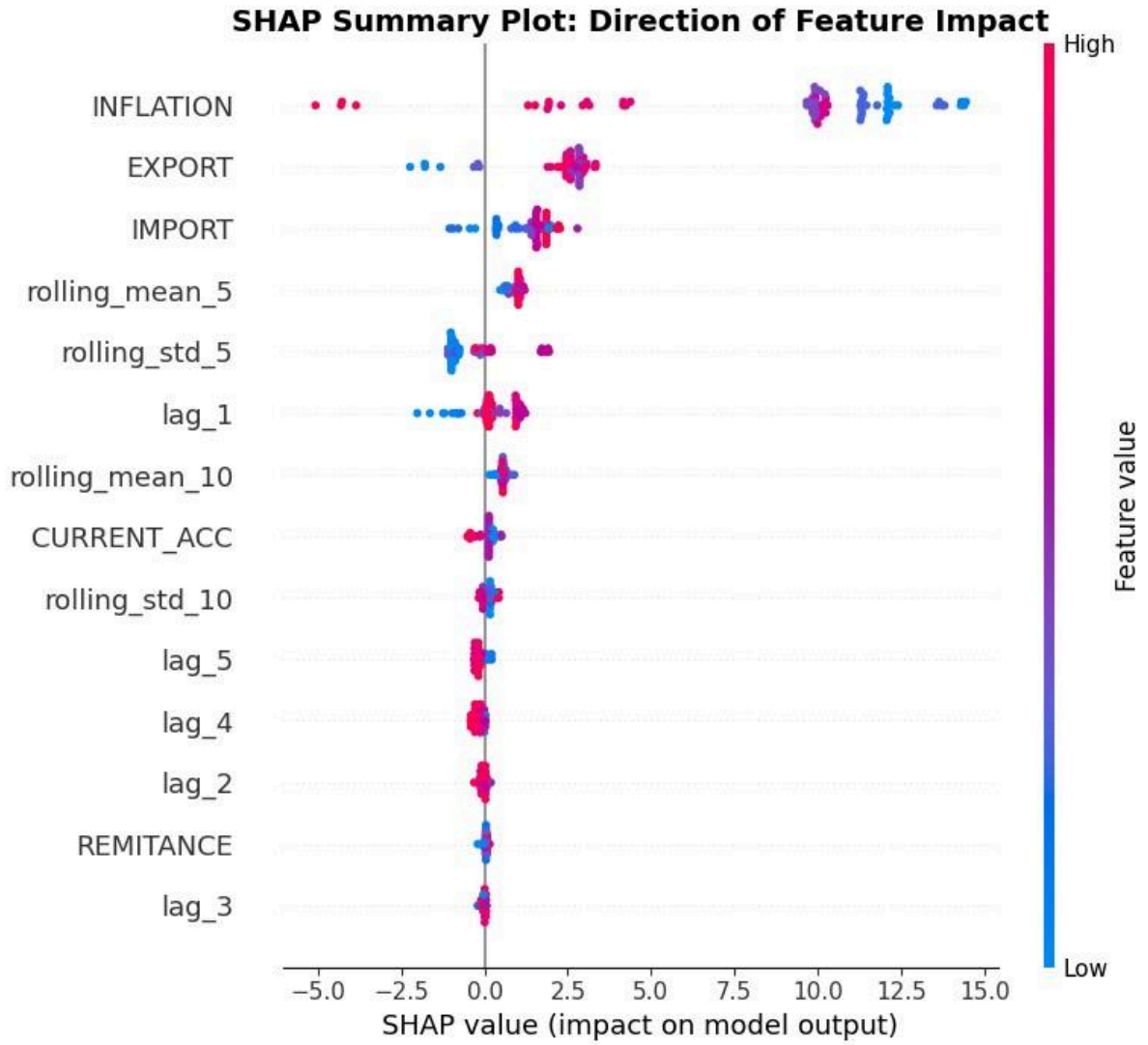


Figure 4.6: SHAP Summary Dot Plot

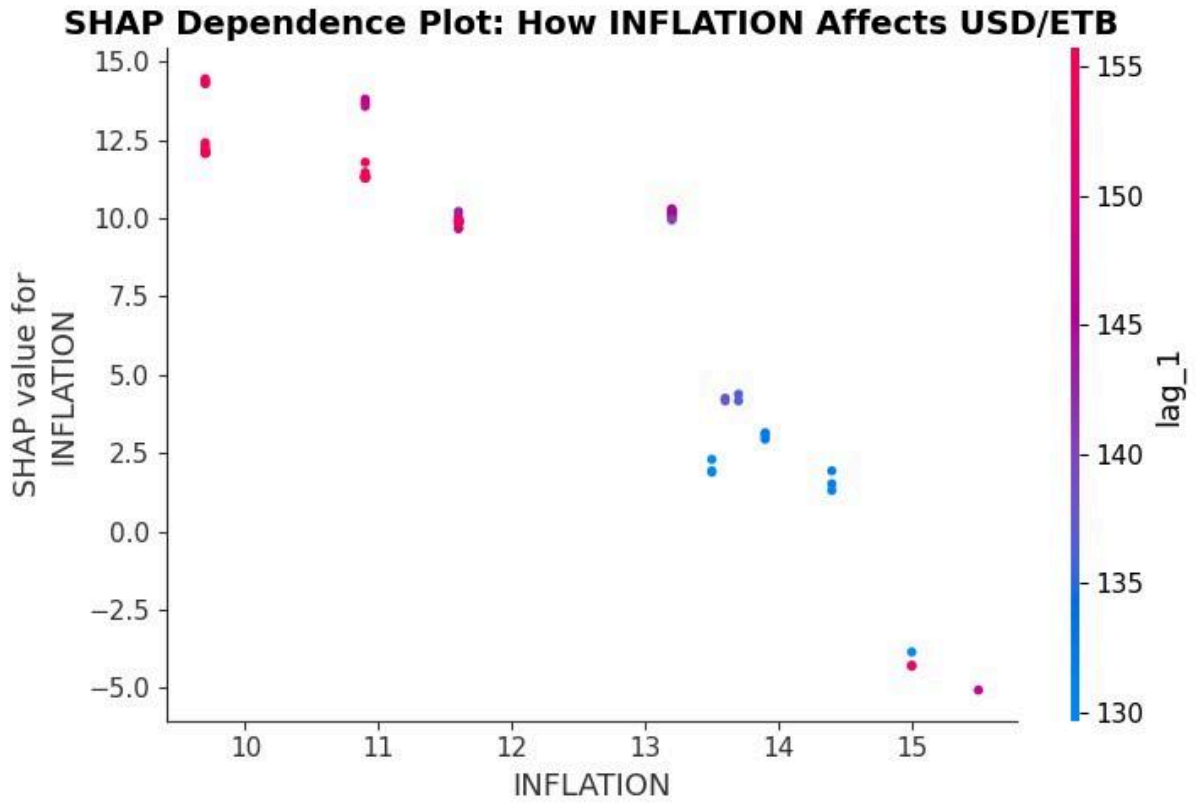


Figure 4.7: Inflation Dependence Plot

Figure 4.7 shows how inflation affects XGBoost predictions. The x axis is inflation rate (%), the y axis is SHAP value. When positive SHAP inflation pushes the exchange rate up and when negative it pushes down. Each dot is one test day. The colour means red for high inflation, blue for low.

The relationship is linear as shown on the graph. At low inflation (~10%), SHAP values are close to zero. It means low inflation doesn't drive the exchange rate higher. Above 15%, SHAP values it becomes positive. No threshold effect is observed. The result a unit equation is derived as: Δ USD/ETB $\approx 1.2 \times \Delta$ Inflation

Another result, inflation when it increased from 15% to 17%, the model predicts USD/ETB increases by about 2.4 Birr. One additional inflation point weakens the Birr by about 1.2 Birr against the dollar.

The SHAP results validate Purchasing Power Parity (PPP): inflation differentials drive exchange rates. They also validate Balance of Payments: exports and imports matter. Remittances are surprisingly weak. A possible explanation: remittance data is monthly and smoothed, missing intra-month surprises or high-frequency diaspora flows.

4.9 DSS Functional Evaluation

The DSS artifact was evaluated on six criteria from chapter three: usability, scalability, adaptability, institutional applicability, interpretability, and operational feasibility.

Table 4.7: DSS Evaluation Summary

Criterion	Rating	Evidence
Usability	High	Single notebook, upload-based
Scalability	High	Modular layers
Adaptability	Medium	Model swapping possible but requires code changes
Institutional applicability	High	Designed for Ethiopian data reality
Interpretability	High	SHAP + feature importance
Operational feasibility	High	Free Colab, <5 min runtime

4.10 Discussion

Three Findings were observed. First XGBoost and ARIMA perform very similarly for the given dataset, with MAPE scores of 2.62 percent and 2.71 percent respectively. This means that for short term exchange rate forecasting, ARIMA remains statistically competent. Second, inflation holds the most weight. The SHAP value of 9.41 accounts for 55% of total importance. This confirms the theory of Purchasing Power Parity (PPP) in Ethiopia's post-reform period. Earlier studies (Almaw, 2021) found strong correlations, and this study measures the marginal effect. Third, remittances show low importance. This is surprising given Ethiopia's reliance on money sent back from the diaspora. One possible explanation is that remittance data are reported monthly with delays. The forward-filled daily values lose important information. Real-time remittance data would likely increase its importance.

When compared to previous work, Goshu and Tilahun (2016) achieved nearly perfect in-sample fit using Grey system theory (MAPE 0.0035%). That study examined a different period (2014-2015) and likely overfitted the data. This study uses a strict out-of-sample test. The 2.62% MAPE reflects a realistic situation for a volatile post-reform period. Gangadhara and Satish (2023) claimed that fundamental analysis does not work for exchange rates. This

study shows it can succeed when combined with SHAP explainability and machine learning. The fundamental variables, such as inflation and trade, do have predictive power.

Limitations of this chapter include the absence of parallel market data, lack of real-time sentiment, monthly macro data that limit high-frequency predictions, and the scenario engine not being fully interactive.

The XGBoost model achieved the highest accuracy (MAPE 2.62%, RMSE 5.337, R^2 0.6329). ARIMA was a close second (2.71% MAPE). Prophet did not perform well (negative R^2). SHAP analysis highlighted inflation as the main factor driving USD/ETB movements (55.1% of feature importance), followed by exports and imports. The DSS artifact is usable, scalable, and interpretable. It runs on free software and accommodates Ethiopia's institutional data needs. The next chapter will include recommendations, limitations, and directions for future research..

CHAPTER FIVE: CONCLUSION

5.1 Summary of Findings

Three models were compared. XGBoost achieved the highest accuracy on the test period with a MAPE of 2.62% and RMSE of 5.337. The second model, ARIMA performed MAPE of 2.71%. The third mode, Prophet, produced a negative R^2 making it invalid. Based on the SHAP analysis inflation is denoted as the dominant driver of USD/ETB movements among the computing variables, with a SHAP value of 9.411 accounting for 55.1% of total feature. Exports with 14.9% feature importance follow closely and imports with 8.4% is third. Rolling volatility and lagged exchange rates contributed smaller shares.

5.2 Answers to Research Questions

Question 1: How to prepare data for machine learning forecasting?

Daily and monthly data were aligned using forward filling. Feature engineering created lags, rolling windows, differencing, and percentage changes to capture patterns. Missing values were filled using forward filling and interpolation. The final dataset reflected the reality of Ethiopia's institutional data..

Question 2: How well do predictive models forecast compared to ARIMA and Prophet?

XGBoost and ARIMA scored nearly equal in evaluating metrics. The MAPE difference was only 0.0 %. The Prophet is considered invalid based on the R measured scored in evaluation.

Question 3: How to incorporate forecasting into a DSS?

The DSS is a modular pipeline: ingestion → validation → transformation → feature engineering → XGBoost prediction → SHAP explainability → output. The architecture can easily scale to real time APIs in the future, although currently it is implemented in Colab notebook.

5.3 Theoretical Contributions

The study confirms the Purchasing Power Parity assumption theory that assumes the exchange rate will adjust so that the same product costs the same in both countries. Inflation differences account for over half of near term USD to ETB fluctuations. Similarly Balance of Payments theory also relates to the findings of this study partially. This is due to Exports and imports together explaining 23.3 percent of USD/ETB movements. Methodologically, this study shows how SHAP analysis can bridge black box machine learning with policy relevant interpretation in an African currency market.

5.4 Practical Contributions

A functional forecasting artifact was delivered. It runs on free Google Colab and produces daily USD/ETB forecasts with a 2.6% average error. For the National Bank of Ethiopia, it provides an early warning tool. Importers and exporters can time their decisions, policymakers can use the system to test how changes predictors will affect the exchange rate before making decisions. Here it is noted that this is a validated prototype

5.5 System Limitations

First is its limitation observed in the study is its lack to account Parallel market due to unofficial data existence. The next observed limitation of the study is its lack to find daily frequency data for some variables. Furthermore the study period is bound to December 31, 2025, because of lack of latest data.

5.6 Future Research Directions

Future directions include using parallel market data, testing LSTM (Long Short-Term Memory) networks on model side and more interactive DSS software. This and other incremental ideas that can support the usefulness of the system for market participants.

The framework can also be designed to add unavailable explanatory variables in tis study. For instance T bill rates and remittance inflows or adding more currency pairs such as EUR to ETB and others.

REFERENCE

Yenus, N. (2025, January 23). Exchange rate dynamics in Ethiopia: Post-reform analysis [Conference presentation]. International Conference, Skylight Hotel, Addis Ababa, Ethiopia.

Kebede, K. A. (2024). The effect of availability of foreign exchange and devaluation of BIRR on the performance of companies in Ethiopia (instance of sample company). International Journal of Economics & Business Administration (IJEBA), 12(1), 99-117.

Goshu, N. N., & Tilahun, S. L. (2016). Grey theory to predict Ethiopian foreign currency exchange rate. International Journal of Business Forecasting and Marketing Intelligence, 2(2), 95–116.

Gurhy, B., Lee, K., & Garimella, S. R. (2025). Potential Drivers of Post-Reform Parallel Market Premium.

Gangadhara, B., & Satish, N. (2023). Fundamental Analysis of the Indian Foreign Exchange Market. Arthshastra Indian Journal of Economics & Research, 12(4), 25-38.

Kiguel, M., & O'Connell, S. A. (1995). Parallel exchange rates in developing countries. The World Bank Research Observer, 10(1), 21-52.

Ghei, N., & Kiguel, M. A. (1992). Dual and multiple exchange rate systems in developing countries: Some empirical evidence (No. 881). The World Bank.

Löscher, A. (2020). Can you spare a dollar, please? Foreign exchange shortage as a persistent challenge to economic development. In How to Make your Doctoral Research Relevant (pp. 34-46). Edward Elgar Publishing.

Agénor, P. R. (1990). Parallel currency markets in developing countries: theory, evidence, and policy implications. Evidence, and Policy Implications (December 1990). IMF Working Paper, (90/114).

Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in information systems research1. MIS quarterly, 35(3), 553-572.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. Nature, 521(7553), 452-459.

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

Rosinus, M. (2025). Comparison of classical ARIMA forecasting methods to the machine learning LSTM method: A case study on DAX® 50 ESG index. ACTA VSFS, *19*(1), 32-52.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

Sirignano, J., & Cont, R. (2021). Universal features of price formation in financial markets: perspectives from deep learning. In Machine learning and AI in finance (pp. 5-15). Routledge.

Fayyad, U. M. (1996). Piatetsky-Shapiro, G., Smyth. Advances in knowledge discovery and data mining.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). Linear model selection and regularization. In An introduction to statistical learning: with applications in R (pp. 225-288). New York, NY: Springer US.

Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.q

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323.

ML Journey. (2025, December 14). Bagging vs boosting vs stacking: Complete comparison of ensemble methods. Retrieved March 11, 2026, from https://mljourney.com/bagging-vs-boosting-vs-stacking-complete-comparison-of-ensemble-methods/

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Kaur, H. (2024). A Comparative Study of Classical and Machine Learning Approaches in Time Series Prediction. International Journal of Science and Research (IJSR), 13(8).

Kolambe, M., & Arora, S. (2024). Forecasting the Future: A Comprehensive Review of Time Series Prediction Techniques. Journal of Electrical Systems.

Panagiotelis, A., Athanasopoulos, G., Gamakumara, P., & Hyndman, R. J. (2021). Forecast reconciliation: A geometric view with new insights on bias correction. International Journal of Forecasting, 37(1), 343-359.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: With applications in R (2nd ed.). Springer.

Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and practice (3rd ed.). OTexts.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: With applications in R (2nd ed.). Springer.

Appendix 1: Predictive Modeling Implementation

```
# ==========================================
# FOREIGN EXCHANGE PREDICTIVE MODELING
# Ethiopia USD/ETB Forecasting
# ==========================================

import pandas as pd
import numpy as np
from xgboost import XGBRegressor
from statsmodels.tsa.arima.model import ARIMA
from prophet import Prophet
from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error, r2_score
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

# Load and prepare data
from google.colab import files
print("Upload dataset file:")
uploaded = files.upload()
filename = list(uploaded.keys())[0]

df = pd.read_excel(filename, sheet_name='Sheet1')
df.columns = ['Date', 'USD_ETB', 'INFLATION', 'IMPORT', 'EXPORT', 'REMITANCE',
'CURRENT_ACC']
df['Date'] = pd.to_datetime(df['Date'])
df = df.sort_values('Date').drop_duplicates(subset=['Date'])
df = df[df['Date'] <= '2025-12-31']

# Handle missing macro data (forward fill)
macro_cols = ['INFLATION', 'IMPORT', 'EXPORT', 'REMITANCE', 'CURRENT_ACC']
for col in macro_cols:
    df[col] = df[col].fillna(method='ffill')
df = df.dropna(subset=['USD_ETB'])

# Feature engineering: lags and rolling statistics
for lag in range(1, 6):
    df[f'lag_{lag}'] = df['USD_ETB'].shift(lag)
df['rolling_mean_5'] = df['USD_ETB'].rolling(window=5).mean()
df['rolling_std_5'] = df['USD_ETB'].rolling(window=5).std()
df['rolling_mean_10'] = df['USD_ETB'].rolling(window=10).mean()
df['rolling_std_10'] = df['USD_ETB'].rolling(window=10).std()
df = df.dropna()

# Define features
```

```
feature_cols = ['lag_1', 'lag_2', 'lag_3', 'lag_4', 'lag_5',
            'rolling_mean_5', 'rolling_std_5', 'rolling_mean_10', 'rolling_std_10',
            'INFLATION', 'IMPORT', 'EXPORT', 'REMITANCE', 'CURRENT_ACC']

X = df[feature_cols]
y = df['USD_ETB']

# Chronological split (80% train, 20% test)
split_idx = int(len(df) * 0.8)
X_train, X_test = X.iloc[:split_idx], X.iloc[split_idx:]
y_train, y_test = y.iloc[:split_idx], y.iloc[split_idx:]
train_dates = df['Date'].iloc[:split_idx]
test_dates = df['Date'].iloc[split_idx:]

# ============================================
# MODEL 1: XGBoost
# ============================================
xgb = XGBRegressor(n_estimators=200, learning_rate=0.05, max_depth=6,
                   subsample=0.8, colsample_bytree=0.8, random_state=42)
xgb.fit(X_train, y_train)
y_pred_xgb = xgb.predict(X_test)

# ============================================
# MODEL 2: ARIMA (2,1,2)
# ============================================
y_pred_arima = []
history = list(y_train.values)
for t in range(len(y_test)):
    try:
        model = ARIMA(history, order=(2, 1, 2))
        model_fit = model.fit()
        yhat = model_fit.forecast()[0]
        y_pred_arima.append(yhat)
        history.append(y_test.values[t])
    except:
        y_pred_arima.append(history[-1])
        history.append(y_test.values[t])

# ============================================
# MODEL 3: Prophet
# ============================================
df_prophet_train = pd.DataFrame({'ds': train_dates, 'y': y_train.values})
prophet_model = Prophet(yearly_seasonality=False, weekly_seasonality=True,
                        daily_seasonality=False, changepoint_prior_scale=0.05)
prophet_model.fit(df_prophet_train)
future = prophet_model.make_future_dataframe(periods=len(test_dates), include_history=False)
forecast = prophet_model.predict(future)
y_pred_prophet = forecast['yhat'].values[:len(test_dates)]
```

```
# ============================================
# EVALUATION METRICS
# ============================================
results = []
for name, pred in [('XGBoost', y_pred_xgb), ('ARIMA', y_pred_arima), ('Prophet', y_pred_prophet)]:
    rmse = np.sqrt(mean_squared_error(y_test, pred))
    mae = mean_absolute_error(y_test, pred)
    mape = mean_absolute_percentage_error(y_test, pred) * 100
    r2 = r2_score(y_test, pred)
    results.append({'Model': name, 'RMSE': rmse, 'MAE': mae, 'MAPE(%)': mape, 'R²': r2})

results_df = pd.DataFrame(results)

# ============================================
# VISUALIZATION
# ============================================
# Figure 1: Best model actual vs predicted
best_model = results_df.loc[results_df['RMSE'].idxmin(), 'Model']
best_pred = {'XGBoost': y_pred_xgb, 'ARIMA': y_pred_arima, 'Prophet':
y_pred_prophet}[best_model]

plt.figure(figsize=(12, 6))
plt.plot(test_dates, y_test.values, label='Actual', color='blue')
plt.plot(test_dates, best_pred, label=f'{best_model} Prediction', color='red', linestyle='--')
plt.title(f'{best_model}: Actual vs Predicted USD/ETB')
plt.xlabel('Date'); plt.ylabel('USD/ETB Rate')
plt.legend(); plt.grid(alpha=0.3); plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('actual_vs_predicted.png', dpi=300)
plt.show()

# Figure 2: All models comparison
plt.figure(figsize=(12, 6))
plt.plot(test_dates, y_test.values, label='Actual', color='black', linewidth=2)
plt.plot(test_dates, y_pred_arima, label='ARIMA', color='red', linestyle='--')
plt.plot(test_dates, y_pred_xgb, label='XGBoost', color='green', linestyle='-.')
plt.plot(test_dates, y_pred_prophet, label='Prophet', color='orange', linestyle=':')
plt.title('Model Comparison: Actual vs Predicted USD/ETB')
plt.xlabel('Date'); plt.ylabel('USD/ETB Rate')
plt.legend(); plt.grid(alpha=0.3); plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('model_comparison.png', dpi=300)
plt.show()
```

Appendix B: SHAP Feature Importance Analysis

```
# ============================================
# SHAP FEATURE IMPORTANCE ANALYSIS
# Ethiopia USD/ETB Forecasting - Interpretability
# ============================================

import shap
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from xgboost import XGBRegressor

# Feature set (consistent with Appendix A)
feature_cols = ['lag_1', 'lag_2', 'lag_3', 'lag_4', 'lag_5',
                'rolling_mean_5', 'rolling_std_5', 'rolling_mean_10', 'rolling_std_10',
                'INFLATION', 'IMPORT', 'EXPORT', 'REMITANCE', 'CURRENT_ACC']

# Prepare data
X = df[feature_cols]
y = df['USD_ETB']
split_idx = int(len(df) * 0.8)
X_train, X_test = X.iloc[:split_idx], X.iloc[split_idx:]
y_train, y_test = y.iloc[:split_idx], y.iloc[split_idx:]

# Train XGBoost model
xgb = XGBRegressor(n_estimators=200, learning_rate=0.05, max_depth=6,
                   subsample=0.8, colsample_bytree=0.8, random_state=42)
xgb.fit(X_train, y_train)

# Calculate SHAP values
explainer = shap.TreeExplainer(xgb)
shap_values = explainer.shap_values(X_test)

# Generate feature importance table
feature_importance = pd.DataFrame({
    'Feature': feature_cols,
    'SHAP_Value': np.abs(shap_values).mean(axis=0)
}).sort_values('SHAP_Value', ascending=False)
feature_importance['Rank'] = range(1, len(feature_importance)+1)
feature_importance = feature_importance[['Rank', 'Feature', 'SHAP_Value']]
feature_importance.to_csv('feature_importance.csv', index=False)

# Generate visualizations
# Figure 1: Bar plot (feature importance ranking)
plt.figure(figsize=(10, 8))
shap.summary_plot(shap_values, X_test, plot_type="bar", show=False, feature_names=feature_cols)
```

```
plt.title('Feature Importance: SHAP Values', fontsize=14, fontweight='bold')
plt.tight_layout()
plt.savefig('shap_bar_plot.png', dpi=300, bbox_inches='tight')
plt.show()

# Figure 2: Dot plot (direction of impact)
plt.figure(figsize=(12, 8))
shap.summary_plot(shap_values, X_test, show=False, feature_names=feature_cols)
plt.title('SHAP Summary Plot: Direction of Feature Impact', fontsize=14, fontweight='bold')
plt.tight_layout()
plt.savefig('shap_dot_plot.png', dpi=300, bbox_inches='tight')
plt.show()

# Figure 3: Dependence plot for top feature (inflation)
top_feature = feature_importance.iloc[0]['Feature']
plt.figure(figsize=(10, 6))
shap.dependence_plot(top_feature, shap_values, X_test, show=False, feature_names=feature_cols)
plt.title(f'SHAP Dependence Plot: Impact of {top_feature} on USD/ETB', fontsize=14,
fontweight='bold')
plt.tight_layout()
plt.savefig('top_feature_dependence.png', dpi=300, bbox_inches='tight')
plt.show()
```